* Introduction.
- Data Base.
Data Base Management System is a Software That is use To Manage The Database.
• Hospital -
Data Base

- The Database is a Collection of Inter related Data which is used To retrieve, Insert & dete The Data.
- DBMS is 1st Created by Charles Bachman, a DBMS is a Collection of Programmes That Enables Users To Create & Maintain Database.
- The DBMS is a General Purpose Software System That facilitated, The Process of Manipulating & Sharing databases among Several Users & Applications
- for Ex. The Company database Organizes The Data about The Admin, Employee, Manager & Clients, Cleaner.
- Data is a Group of Measurements, observation & Discription Than Can Be Used To Convey Info.
* DBMS Stands for Database Management System.
- The Data base Defi or Descriptive Info. is also Stored By The DBMS In The form of Database Catlog & Dictionary It is Called as Metadata.

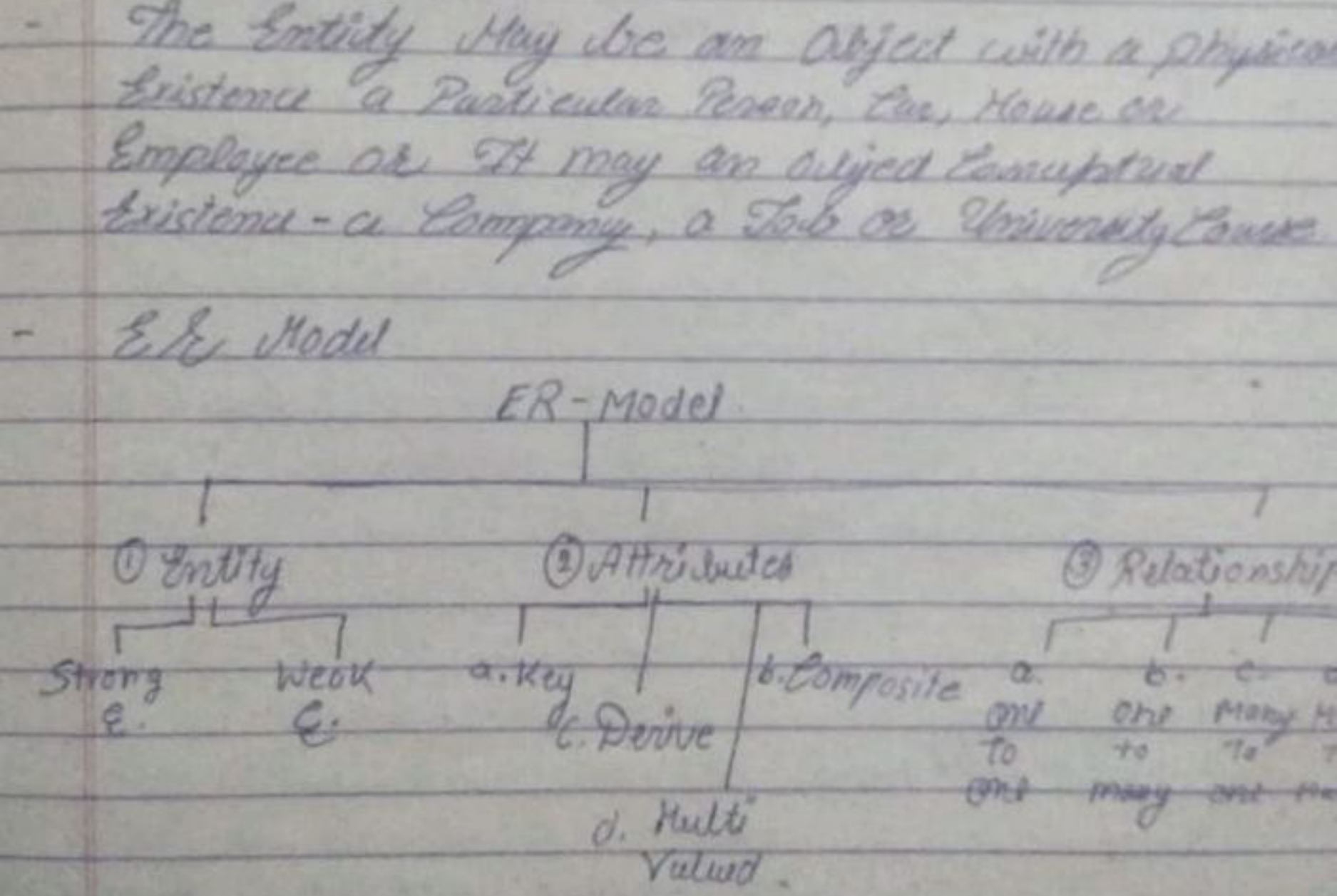
* Application.

1. Enterprise Information
2. Airlines
3. Telecommunication
4. University
5. Banking & Finance Sector
6. Social Media Sites
7. Manufacturing.

} Application of DBMS.

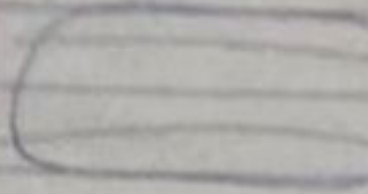
- The Entity May be an Object with a Physical Existence a Particular Person, Car, House or Employee or It may an Object Conceptual Existence - a Company, a Job or University Course

- E-R Model



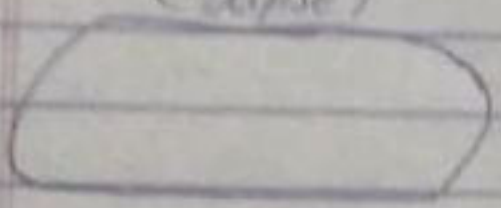
- ER Model is used to model the logical view of the system from a data perspective which consists of these basic symbols:

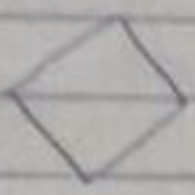
(rectangle)

1.  - It represents Entity in the ER Model.

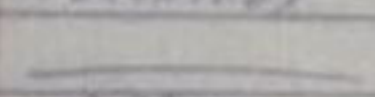
(rectangle).

(Ellipse)

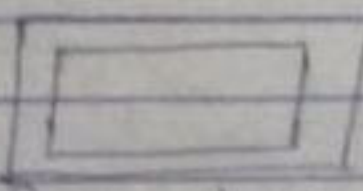
2.  - It represent attribute in ER Model

3.  - It represents relationship among Entities

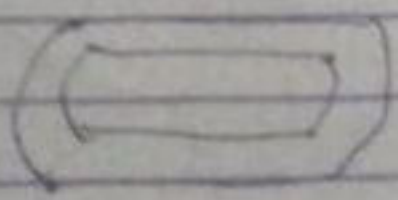
(diamond)

4.  - Attribute's To Entities & Entity sets with other relationship Types.

(Lines)
(Connected)
(connect Entity)

5.  - It represents weak Entity in ER Model.

Double rectangle

6.  - It represent Multi Value attributes.

alter Table Student ADD (Pincode int);

Capital.

* Advantages & Disadvantages of DBMS.

Advantages

1. Simplicity
2. Structural Independence
3. Ease of use
4. Query Capability
5. Few relational database have limits ~~of~~ on fields, length which cannot be Exited.

Disadvantages.

1. Maintainace Problem
2. The maintaince of relational DB. Becomes Difficult over Time Due to The Increase In The Data.
3. Cost
4. Physical Storage.
5. Complexity in Structures decrease in Performance over Time.

To maintain Data Integrity.

4. Indexes : Specific Indexing Strategies to optimize retrieval.

5. Normalization - Describe The level of normalization to avoid Data redundancy & Improve data Integrity.

6. Views - outlines Vectual Tables (views) That are derived from one or More Tables for Specific Purpose

Defi -

"Blueprint refers To schema or Data model That defines how database will be structured, including The Tables, relationships, Constraints & other elements that make up The database."

* Entity is an object or Thing in The real World that is distinguishable & Can be represented In a database. Entities Typically represents objects, Concepts, Event or places That have a distinct Existance & are relevant to database's purpose.

- Ex. For database for University - Entity could include Student, Course, Professor.

2. Attribute

Attributes are characteristics or properties that describe an Entity. They Provide more detail about the Entity by Defining Its Specific Qualities.

Ex. 'Student' is an Entity

- Student ID, Name, DOB, Address are attributes.

"An Entity is Something about which data is stored, & its attributes define Specific details about that Entity."

Applications of DBMS (Explanations)

1. Enterprise Information - Sales, Accounting, Human Resources, Manufacturing, online Details.

2. Airlines - Client related Date, reservation & Schedu

3. Telecommunication - Phone, Telephone - PostPaid, Prepai Bill maintainance.

4. University - It maintain's The Information About Student, Course, Exams, Banking Transactio Email, Student grades, Staff role.

5. Banking & Finance Sector - Banks Maintain The Customers Details, Accounts, Banking Transa -ons, Credit card Transactions.

Finance - Storing The Information about Sales & Holdings Purchasing of Financial

(Types of DBMS)

1. Relational Database Management System (RDBMS)
   - Data is Organised into Tables (relation) with rows & Columns & relationship between Data is Managed Through primary & foreign Keys.
   - SQL (Structured Query Language) is used To Query & Manipulate data.

2. No-SQL DBMS
   - Designed for high Performance Scenarios & large Scale data.
   - No-SQL databases store data in Various non-relational formats Such as Key Value pairs, documents, graphs or Columns.

3. Object-Oriented DBMS
   - Stores data as object, Similar To Those used in object-oriented Programming, allowing for complex data representations & relationships.

* Database Languages

DDL (Data Definition L.)	DML (Data Manipulation L.)	DCL (Data Control L.)	TCL (Transacti-al Contro
- Create	- Select	- Grant	- Roll Back
- Alter	- Insert	- Revoke	- Commit
- Drop	- Update		- Save Point
- Truncate	- Delete		
- Comment	- Merge		
- Rename	- Call		
	- Explain Plan		
	- Lock Table		

* Database Languages.

1) Data Definition Language.
- It deals with database Schema & descriptions. how data should reside in database.

- (Create)- To Create a Database & Its objects like (Table, Index, Views, Store Producers, Functions & Triggers.)
Syntaxes :-(Create Database Database_Name;
(Create Schema Schema_Name;)
-(Create Table Table_Name((Column1, Datatype

- (Alter)- It By using alter we can include or drop one or m
Columns from the Existing Table. Also we can incl
new Columns in Existing Tables.
Syntax :- -(alter Table Table_name ADD(Column_name datatyp

- (Drop)- It is used To delete the Structure & record Stored In the Table To drop a Table Permanently from th Memory.
Syntax : -(Drop Table Table_name;)

- (Truncate)- Remove all spaces from a Table (rows) including all spaces allocated for records are removed.
Syntax :- (Truncate Table Table_name;)

- (Rename): It is used To rename the Table.
Syntax :-(Rename old Table_name To New_Table_name;)

- (Comment): Include Comments To Data Dictionary.

2) Data Manupulation Language (DML)
- It Deals with data Manipulations & includes most Common SQL Statements Such as SELECT, INSERT, UPDATE, DELETE etc.
- It is use To store, modify, retrive, delete & update data in a database.
- Data Query Language (DQL) is Subset of DML.
- Its Most Common Command ~~is SQL~~ of DQL is SELECT.
- SELECT Statement help on retriving ~~Table~~ data from Table without Changing Anything in Table.

- (SELECT)- To Acess Data from Database.
Syntax - SELECT * from Table-name;

- (Insert)- Insert Data Into Table.
Syntax - ~~For~~ INSERT_INTO Table-name (Values);
Ex. INSERT_INTO Student Values (102, 'ABC');

- (DELETE)- Delete all records from A Database Table Temporarily. It is used To remove rows from Table
Syntax - DELETE FROM Table-name WHERE Condition;

Name of Table u wont To Delete

- The Condition The Identifies which row To Delete.
- If no Cond^n Specified all rows would delete.

- (UPDATE)- updates Existing Data within a Table.
Syntax :- UPDATE table-name
SET COLUMN1 = Value1, Column 2 = Value 2;...
WHERE Condition;

- Merge - UPSERT operation (Insert or Update)
- Call - Call a PL/SQL or Java Subprogram unit
- Explain PLAN - Interpretation of the data Access Path.
- LOCK Table - Concurrency Control.

All are in Capital letters.

3) Data Control Language (DCL)
- It Acts as an access specifier to Database.
(Basically to grant & revoke Permission to users in DB.)
- GRANT - Grant Permission to user for running DML (SELECT, INSERT, DELETE..) Commands on the Table.
- REVOKE (Cancel) - revoke permissions to user for running DML (SELECT, INSERT, DELETE...) Command on specified Table.

4) Transactional Control Language (TCL)
- It Acts as an Manager for all Types of Transactions Data and all Transactions. Some of Commands of TCL are:
- Roll Back - Used To Cancel or Undo Changes Made in DB.
- Commit - It is Use To apply or Save Changes in DB.
- Save Point - It is use To Save data on the Temporary Basis in DB.
- DQL is Subset of DML, Its Common Command SELECT use To retrive data from Table ~~it~~ without making any changes Malifeation in Table. DQL is very essential for retrival of essential data from a DB.

* (Advantages of DBMS)

1) Data Organisation - A DBMS allows for the Organisation & Storage of data in a structured manner, making it Easy To retrive & Query the data as Needed.

2) Data Integrity - A DBMS Provide Mechanism for Enforcing data Integrity Constraints. Such as Constraints on Values of data & access Controls That restrict who Can Access The data.

3) Concurrent Acess - A DBMS Provide Mechanism for Controlling Concurrent Acess To DB. To Ensure That Multiple User Can Access Data without Conflicting with Each other.

4) Data Security - A DBMS Provides Tools for Managing Security of data, Such as Controlling Acess To The data & Encrypting Sensitive Data.

5) Back UP & Recovery - DBMS Provides Mechanisms for Backing up & recovering data in Event of a System failure.

6) Data sharing - A DBMS allows Multiple Users to access & Share The Same data, which Can be useful in a Collaborative work Environment.

* Disadvantages.

1) Complexity - DBMS Can be Complex To Setup & Maintain, requiring Specialized Knowledge & Skill.

2) Performance Overhead - The use of DBMS Can add overhead to performance of an application, can specially in cases where high level of Concurrency is required

3) Scalability - The use of DBMS can limit the scalability of an application, since it requires the use of locking & other synchronization mechanism to ensure data consistency.

4) Cost - The cost of purchasing maintaining & upgrading a DBMS can be high, especially for a large & complex system.

5) Limited Use Cases - Not all use cases are suitable for a DBMS, some solutions don't need reliability, consistency or security & may be better served by another type of data storage.

* Applications Of DBMS:

1) Enterprise Information - Sales, Accounting, human resources Manufacturing, Online retailer.

2) Banking & Finance Sector - Banks Maintaining the Customer details, accounts, loans, Banking Transaction, Credit Card Transaction. Finance: storing information about Sales & holding, purchasing of financial Stocks & Bonds.

3) University - Maintaining Information about Student Courses enrolled Info., Students grades & Staffs role.

4) Airlines Reservations & Schedules.

5) Telecommunication - Prepaid & Post paid bill Maintainance.

- [illegible] Schema [illegible] logical representation है,
of Data [illegible] [illegible] [illegible] (row) [illegible] Store
Hona chahiye Database के [illegible] in a logical manner.
- [illegible] [illegible] है [illegible] organize [illegible] [illegible] है
- Relation between Tables & [illegible] Contain करता है Table, view, field, relation
between keys.

Schema

"Its The Logical Representation of The Database."

- Organised - How Data is organised in The Database.
- Relations - It Tells about relation of The Data Weather is Dependent or Independent etc.
- Constraints - All Constraints are define.
- Entities - Schema defines ~~rel~~ relation among different entities.
- Database designer designs The Schema So others Can Understand It.
- To Implement Schema we have To Use SQL.

Ex. Student

ID	Roll no	Address

← Its Diagramatical Structure.

Database Schema.

1. A Database Schema is a Logical representation of data, that shows how The data in a database Should be stored Logically. It shows how The data is organised & relation Between Tables.
2. Database Schema Contains Table, Views, fields & relation between different Keys. (Primary & foreign).
3. (Data is stored In The form of files which is ~~Unstructured~~ unstructured in nature which Make accessing data difficult Thus To resolve This issue Data is organised in a structured way with The help of database Schema.
4. Database Schema define Sets of guidelines that Control database, along with That it Provide Information about way of accessing & Modifying data.

Current Point Rec of Time Rec To Information available that is use Instance rec of.

* **Instances For a Database :**

- The Instance of database is The Values of Three Variables at any given Time. Instances are also called Current State or Database State. The Database Schema is a Design That defines The Variables in The Tables That belong To a Particular Database. There May be many Instances That Correspond To Certain Database Schema. The most Data Item Can be Inserted, Modified or Deleted at any Time. So, According To This we Can Say Data Can change from one Stage To another.

Ex.

Order id	Item	Amount	Balance id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mouse Pad	250	2

- The 5 rows in above-provided Table are called Instances Because They Provides Information of Database stored at The Current Point in a Time. So, on This Basis, We Can Say That Instances give Information of database at any Point in Time.

* **States in Database**

1. Empty ~~States~~ stage : This State occurs when New Database is created.
2. Initial Stage : This State is occurs when The data is insert into Database for Very first Time.
3. Current Stage : The Present Image of Database at a Current Time.

DATABASE ARCHITECTURE

* Database Architecture.

1. 1 Tier Architecture / Client Tier Architecture

- All The Application & Data are present on one Computer. Even presentation will be also done on The Same Computer.

Ex. Microsoft Excel, word. Even Games

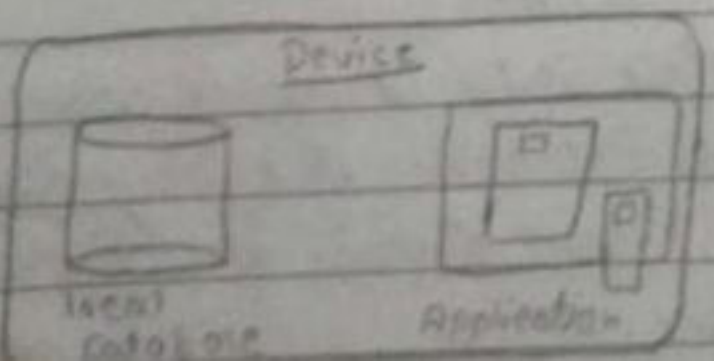
- This whole Application (consider one of them) is present on one device & all The Presentation & View is also on Same device The application is installed & All Data related To It will be stored on The Same Computer.
- There is No other layers In This Tier of Architecture Everything is present on a Single Machine.

* Geeks Geeks Info!

- In - 1 Tier Architecture The Database is Directly available To User, The user can directly sit on The DBMS & use it. That is, The client, Server & Database all are present on Same Machine.

Ex. Microsoft Access - A user open Microsoft Access on Their Computer.

- The application (Access) directly Access The Local Data Base file & Performs operation like Querying, Inserting, Updating or Deleting Records.
- There is No Seperation Between The Application & The Database Since Both resides on a Same Machine.



* Advantages of 1-Tier Architecture.

1) Simple Architecture - 1-Tier Architecture is the most Simple Architecture to set up, as only a single Machine is required to maintain it.

2) Cost-Effective - No Extra hardware is required for Implementing 1-Tier Architecture, which makes it Cost Effective.

3) Easy To Implement - 1-Tier Architecture can be Easily Deployed (moved) & hence it is mostly used in Small Projects.

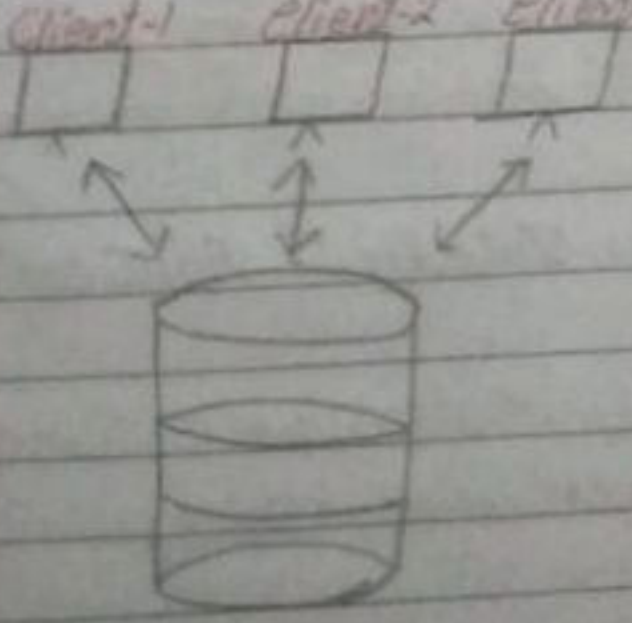
* Disadvantage

1) Security - Only one User can access System at a time.

2) Cannot share Info. - Info. cannot be shared in Client Machine

3) Application May Not Work - It May ~~not~~ not work if Changes are Made in Machine.

* 2-Tier Architecture



- Two Tier Mean 2 Layer Here are Two Layers one Client layer & second database layer.
- Here, client is a pc Machine in which a Interface is running & This interface is helping us to fetch the data from this database server.
- It form connection with database using JDBC-ODBC.

Here, 1st client & Database Server will form Connection Then a Query will be written on the Interface & Then This Query will come To Database Server & Here it will get Processed (Because our written Programme Can be in The high level Language So To Convert it into low level Language Processing is done) & after This whatever would be The demand of The client will be given back To it.

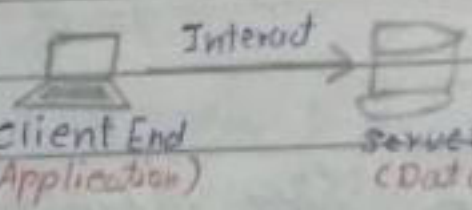
- This is how 2-Tier Architecture works.
- Limited clients & Database hence Maintainance is Easy.

Problems

- Scalability -
- Security - client is directly Interacting with Data Base

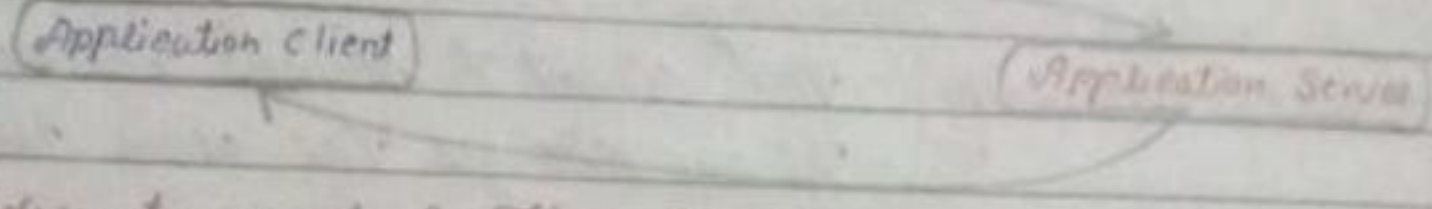
(Geeks for Geeks).

- The 2-Tier Architecture is Similar To a basic Client-Server Model.
- The Application at The client End Directly Communicates with Database on Server Side.
- API's like JDBC & ~~OR~~ ODBC are Used for This Interaction.



- The Server Side is responsible for Providing Query Processing (Solving Given Program or Question) & Transaction Management functionalities.
- On The client Side, The user Interface & Application programs are run.
- The Application on client Side Establishes a Connection with Server Side To Communicate with DBMS.

- An Advantage of This Type is That maintainance & Understanding are Easier & Compatible with Existing Systems.
- However, This Model Give Poor Performance when There are Large Number of Users.



* Advantages of 2-Tier Architecture.

1. Easy To Access - 2 Tier Architecture make Easy Access To Data-Base, which make fast retrival.
2. Scalable - We Can Scale The Database Easily, by including Clients or upgrading hardware.
3. Low Cost - 2-Tier Architecture is cheaper than 3-Tier Arch. & Multi-Tier Arch.
4. Easy Deployment - 2 Tier Arch. is Easier To Deploy Than 3-Tier Architecture.
5. Simple - 2-Tier Architecture is Easily Understandable & well as Simple Because of only 2 Components.

* Disadvantages.

1. Security - Client Directly Interacts with The Database. - which Can Expose The Sensitive Data.
   - It's harder To protect System from Security Threats.
2. Scalability - - As more users Connect, The System or - Database gets Overloaded, Slowing Things Down.
   - It's harder To Expand System to handle more users.

* 2nd Tier Architecture.

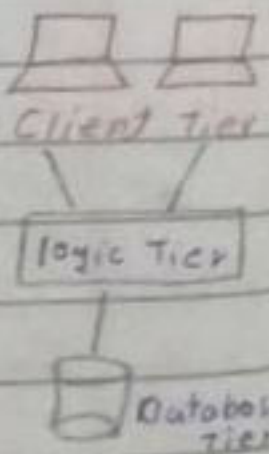
3) Single Point of Failure: If Server or Database goes Down. The whole System stops working. Because There is No Back-up Layer To Keep Things running.

4) Limited flexibility: The client & Servers are closely Connected
- If one changes The other usually Needs To change Too.
- Which Can Make Updates More Difficult

* 3rd Tier Architecture.

- Into Three-Tier Architecture for DBMS, The System is Devided in 3 Distinct Layers. - Each with Specific role.
- These structre improve ~~Scalabi~~ Scalability, maintainability & Security by ~~Spe~~ seperating The Different responsibilities

1. Presentation Tier (Client Tier):



- Role: - This is The Topmost layer That Interacts with The user. It's Consist of User Interface, where users Can input Data, view results & Interact with System.

- Ex:- A web browser, Mobile App, Dekstop Application.

- Functions:- It Sends user requests To Middle layer & receive The Processed Data from Middle layer & Present The result To user in a readable form (like a web Page or a App screen).

2. Logic Layer Tier (Application / Business Logic Layer Tier):

- Role :- This layer acts as an intermediary between the client Tier & the Database Tier.
  - This layer handles the business Logic, Processing of Data & Performs any necessary calculations or transformations on Before sending it to Database or Back to the user.
- Ex :- Web Servers, Application Servers or Business Logic Engines.
- Functions :- It Process the User Input, Interacts with the Database to retrive or Modify data, applies business rules & sends results Back to the (Presentation Tier/client Tier).

3. Database Tier :

- Role : This is where all the data is stored, managed & processed. It includes DBMS & Database Itself.
- Ex : Relational Databases like My SQL, Postgre SQL or SQL Server, No SQL Database like Mongo DB
- Functions :- It is responsible storing, retriving & updating Data.
  - The Data Tier handles Queries from Logic Tier
  - Perform operation on Database & returns the results.

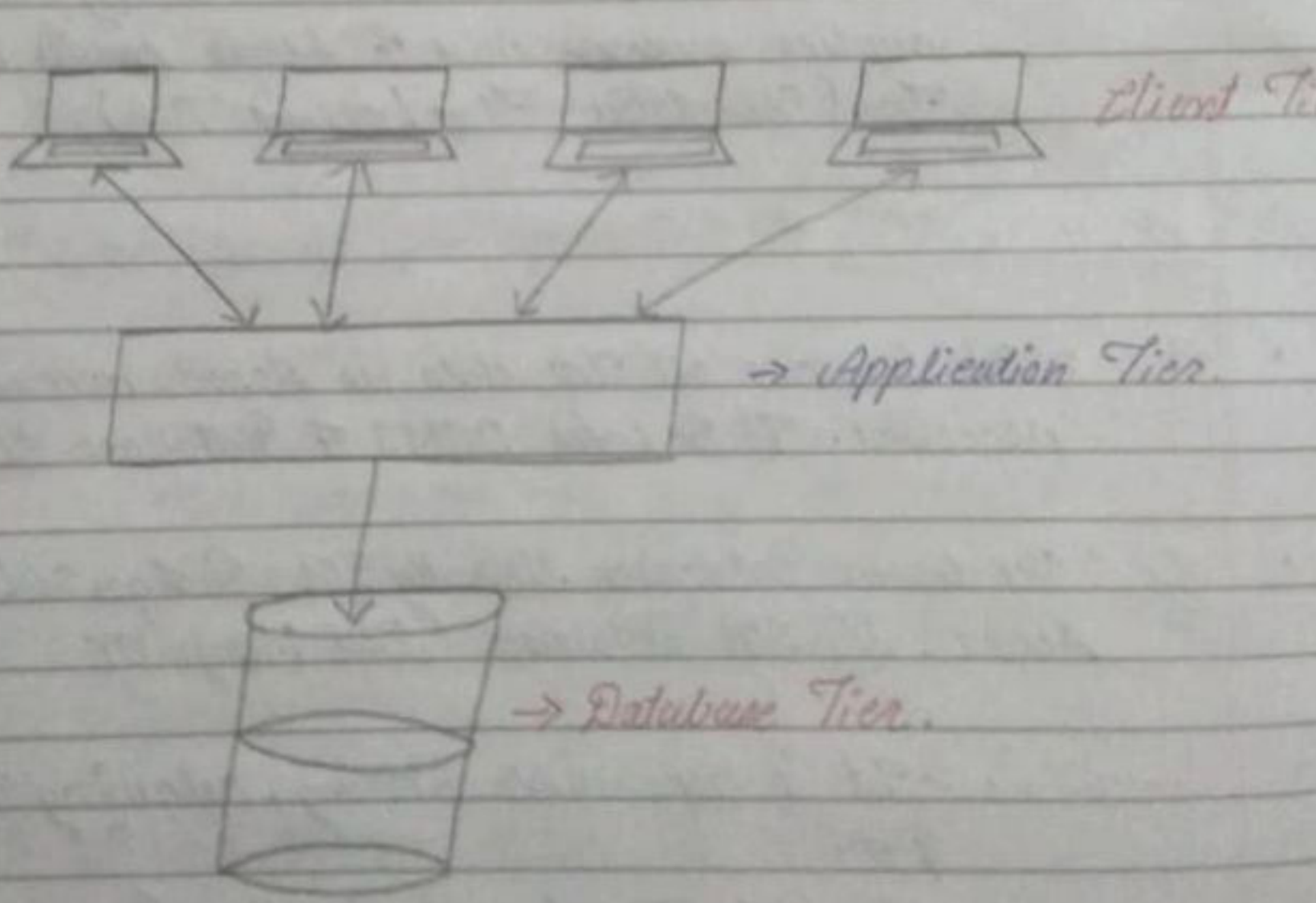
* Workflow in a 3-Tier Architecture.

1. The user interacts with the Presentation (web) Interface.

2. The request is sent to the Application Tier, which pr
the logic (request) & communicate with the Database Tier

3. The Database Tier execute queries, returns Data, & sends
Back to Application Tier.

4. The Application Tier then return the Processed Data to
Presentation Tier, which Displays it to User.



3-Tier Architecture.

Page No.
Date

Advantages

1. Each layer is seperate, which make it Easier To manage & update Each part without affecting other.
2. You can Scale Each Tier Independently. Ex you can Include More Servers To Database Tier without affecting Presentation layer.
3. By Seprating The Data & Bussiness logic layers from Presentation layer, Sensitive Data can be better Protected & Managed.

Disadvantages:

1. Complexity - Managing & Maintaining 3 different layers can be more Complicated Than a Simple Architecture.
2. Performance Overhead: Communication Between Tiers can ~~Include~~ lead To Delay or latency & reduce Overall System Performance.
3. Cost: More Resources (Servers, Infrastructure) may be needed manage Each Seperate layer, Increasing The Overall Cost.

* Data Models.

- Data Models are Mainly Useful in order To Design The DataBase.
- Data Model is a Complete Idea about how final System would look like after Its Implementation.

- A Datamodel in DBMS is Conceptual Framework That defines Structure, relationships, Constraints of data stored in Data Base.
- It Serves as Blueprint for designing database, describing how Data is Organised How It Can be accessed & Manipulated.
- Data Model provides a way To Describe The Logical Structure of data, Independent of Actual Implementation In DBMS.

Ex.

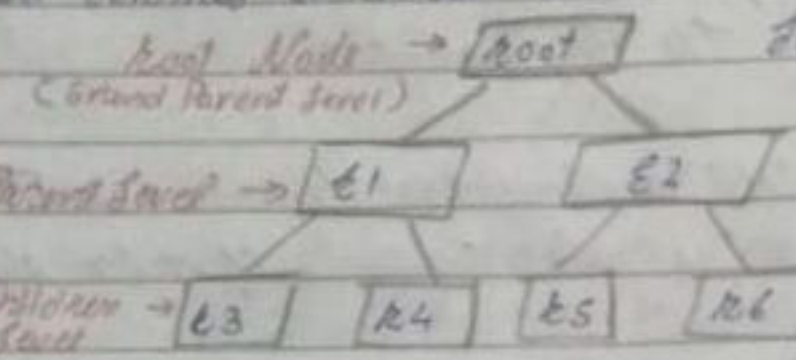
- My SQL - A widely-used open-source relational DBMS. It Stores Data in Tables & is used by Many web ~~Publications~~ Applications.

Example Use - A website storing Users Information, Names, Email, Passwords in Database.

1. Hierarchial DM
2. Network DM
3. Entity-relationship DM.
4. relational Model.
5. Object Based Data Model.

1. Hierarchical Data Model (Developed By IBM 1950's).

- It is Mainly Used To store The Information in a Hierarchy or level by level Manner.
- Grand Parent Level, Parent Level, Children Level.
- It forms a tree Tree Structure.



- It Mainly forms One To Many relationship.
(Each Node will have only One Parent Node & Many children)

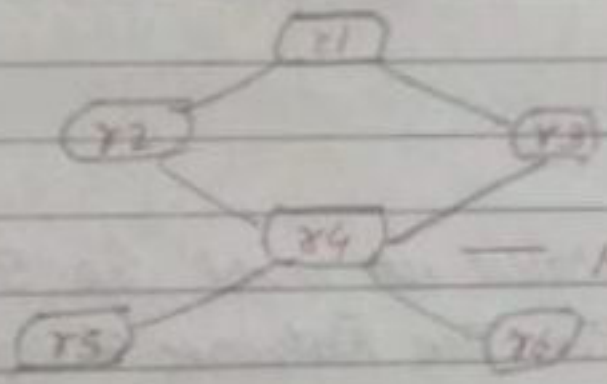
- In This Data is organised in a Tree like Structure where Each record Consist of one Parent record & Many children.
- In Hierarchial Model, Segments Pointed To by The logical association are called The child Segment & other Segment is called Parent Segment.
- If There is a Segment without Parent It will be called as Root & Segment which has No children are called leaves.

Advantages	Disadvantages
1. It has Very Simple hierarchial DB structure.	1. lacks flexibility. Deletion of one Segment Can lead To Deletion of all Seg. under it.
2. It has Data sharing, as all Data are held in Common DB.	2. It has no standard.
3. It offer Data Security.	3. It is also limited as many of Comm relationships do not Conform to 1 to N format as required by Hirarchial Model.

2. Network Data Model

- It follows the Graph Structure.
- It follows Many-To-Many relationship.
- Here Each Node can have many children & Parents.
- It is an Extension of Hierarchial Data Model.

- The Network Data Model is one of the most oldest Data Model that was designed To handle Complex Data relationships more effectively Them the Hierarchial Model.
- In Network model Data is organised in a Graph Structure which allows for more flexible & Complex relationships between different Types of Data. This model is particularly usefull for representing Many To Many Relationship. where a single record can be associated with multiple other records in Both Direct.



Advantages.

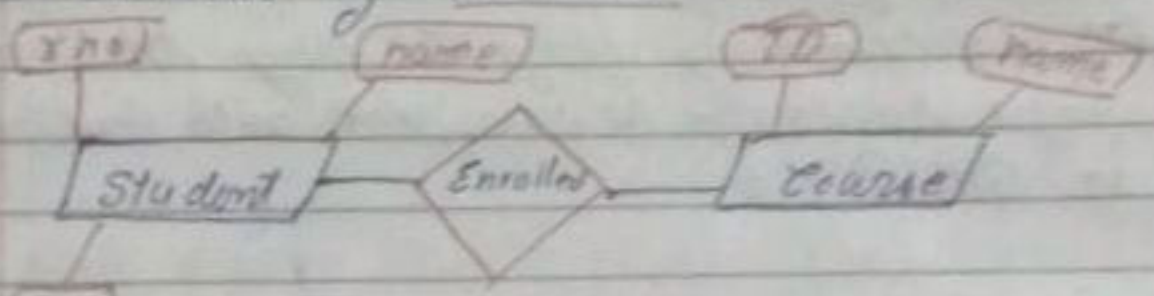
1. The Network Model is flexible
   - It allows to represent Complex real world relations.

2. The Network model allow many-to-many.
   This is usefull in cases where an Entity might have Multiple relation with others.
3. Data Integrity.
4. Supports for multiple Parent records.

3. Entity Relation Model (ER-Model)

- Contain 3 Things
  1. Entity 2. Attributes 3. Relationship.

- Entity - Anything That has an Physical Existance is Called an Entity & also It is Distinguishable.
- Attributes - Properties of Entities. It Tell more about Entity.
- Relation - It is use To Connect 2 Entities.
- Entities are represented by rectangle.
- Attributes by Ellipse
- relations By Diamond.



- The Entity relationship Model is model for Identifying entities To be represented In The Database & representation of how Those Entities are related.
- The ER Data Model Specifies (clearly) Enterprise (Project) Schema. That represent Overall Logical Structure of a DB Graphically.
- Peter Chen Developed ER Model in 1976.
- The ER Model was created To Provide a Simple & Understandable model for representing The Structure & Logic of Data Base.
- The ER Diagram Explain relationship among different Entities present in DB. ER Model are used To model real-world objects like a person, Car, Company & Relation Between This real world objects.

4. Relational Model.

- A relational Database is defined as a group of Interdependent Tables, which are linked To Each other using Some Common fields of Each Related Table.
- This Model Can be represented as Model with rows & Columns.
- Each row is Known as Tuple.
- Each ~~Table~~ Column of a ~~Column~~ Table has a name or attribute.
- It is well Known as DB Technology Because It is usually used To represent real world objects & relation Between Them.
- Ex. Oracle, Sybase, MySQL Server etc. ← relational Models.

5. Object Orient Data Model

Entity & Entity Sets.

- Entity - "Entity is a Thing or a object in real world which is distinguishable from other objects Present in the world, Based on the Values and attributes it Posses".
  - Entity vo cheese hain Jo world main hain Pr agar hum usse Ek Group main se Identify Kr skte hain on the basis of Its Some attributes or Qualities usse Entity kahte hain.

- Types Of Entities -
  - Tangible - Entities which Physically Exist in real world is called Tangible Entities.
    Ex. An Company, A bank, Locker etc.
  - Intangible - Entities which doesn't Exist Physicall but Exist logically are called Intangible Entities.
    Ex. An Bank Account, Mobile Application Etc

- Entity Set.
  - "Collect / Set of ~~Same~~ Same Type of Entities That share same properties or Attributes called Entity Set".
  - "The Entities of Entity-Set have Common attribute for Each Set, But Values of Those Attributes are ~~Zero~~ different
  - Example:
    The Collection of all ~~Tables~~ of Students from student Table (Here student include with Its attributes Name, id, roll etc). at a particular Instant of Time.

This Entities have some attributes

Ex. Table Name : Student.

Stud-id	Stud-Name	Age	Gender
1	Sushil	20	M
2	Tony	19	M
3	Kumar	21	M
4	Arjuni	18	F

This Entities have some attributes.

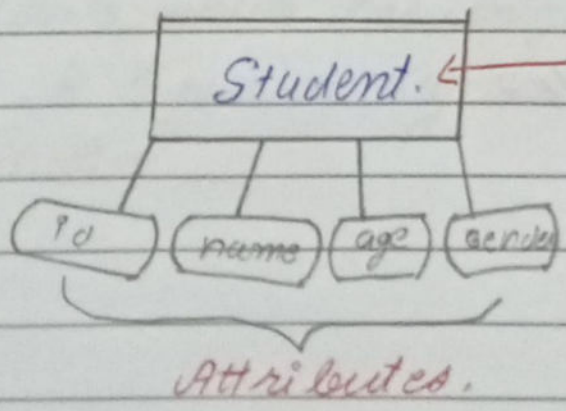
→ Here In This Table Individual Elements & data are Entities.

- Here Each row is an Entity, Because It is distinguishable from Each other.

- Here Each Entity OR row belongs To Student Hence, The Type of Entity here is Student.

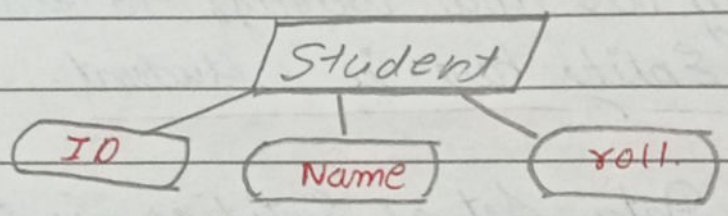
- The Complete Data Set of Entities is Called Entity set (Everything Present in Table) for above Table records with Stud_id, name, age, Yender are Entity set.

- representation is ER Diagram.
  - we Can't represent Entities in ER Diagram Beeause Entities are data or Info at Particular Instance of Time.
  - Entity Set Can be represented in rectangle.



Attributes & Its Types in DBMS.

- Attributes are Properties or characteristics of an Entity.
- Attribute is use To describe The Entity.
- Attribute is Nothing but a peiece of data That give more Information about Entities.
- Attribute is use To distinguish one Entity from other Entity.
- Attribute helps to Categorize The Entities & Entity can be Easily retrived & Manipulated.
- An Entity without attribute is of No Use in database.
- Ex.



Here, ID, Name, roll.no is The attributes of Entity Set Student.

- Types of Attributes.
  - Simple Attribute
  - Composite Attribute
  - Single Valued Attribute
  - Multi Valued Attribute
  - Key Attribute
  - Derived Attribute
  - Stored Attribute
  - Complex Attribute.

1) Simple Attribute -
"Simple Attribute are The attributes Than Cannot be divide into More Attributes." They Cant be divided in sub-attributes.
Ex. roll_no, Name, Age Etc.

2) Composite Attribute -
"When Two or More Than Two Simple attributes are Combined To make an Attribute, Than That Attribute is Called Composite Attribute".
Ex. Address - Its a attribute That is derived from 3 Simple Sub-attributes City, State & Street.

3) Single Valued Attribute.
"The attribute with only a Single Value is Known as Single Valued attribute".
Ex. DOB - Date of Birth Can be only one. roll_no Etc.

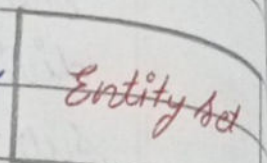
4) Multi Valued Attribute.
"An Attribute which Can have Multiple Values is Konown as Multivalued attribute".
Ex. attribute name Ph_no one Person Can have Multiple Ph_no's.

5) Key Attribute.
" The attribute which has Unique Value for Every row in The Table is Known as Key attribute.
Ex. rollno of Student

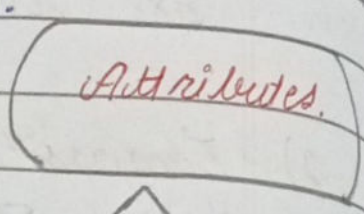
Date

- Representation of ER Diagram.

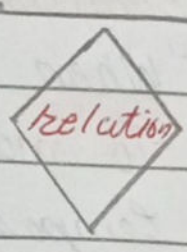
1) rectangle represent The Entity in ER Diagram.



2) Elipse is use To represent The Attributes.

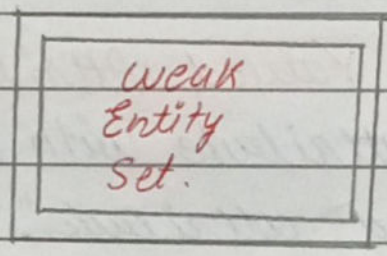


3) Diamond is use To represent relation Between Entity sets.

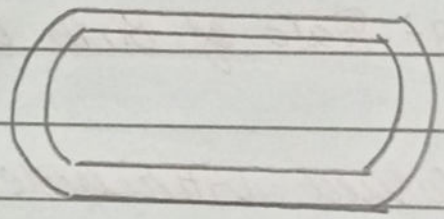


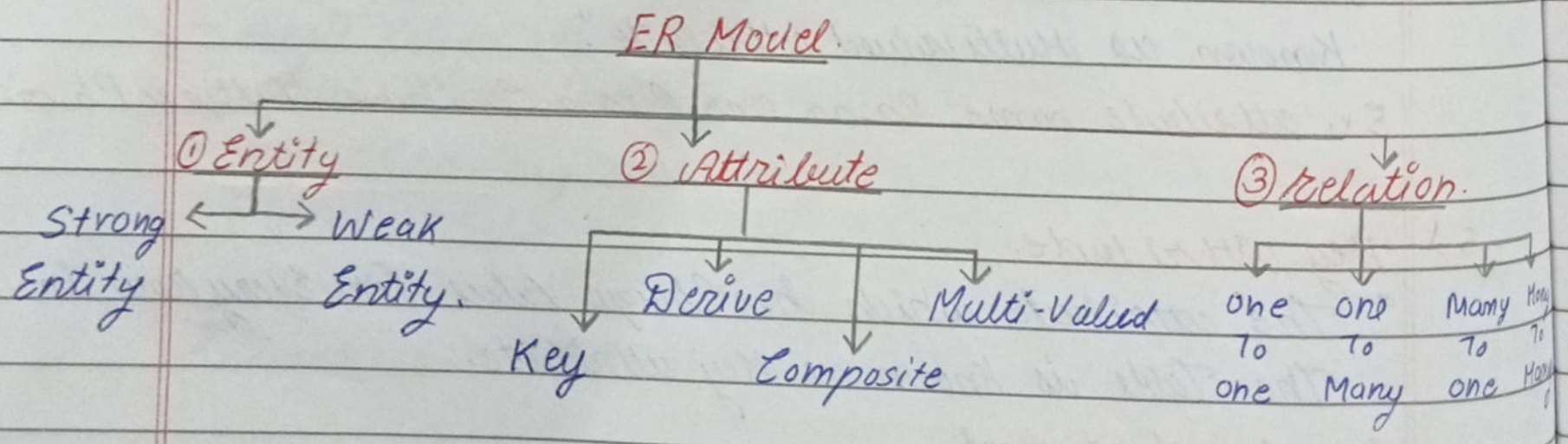
4) Lines are used To Connect Entities with Attributes and diamond.

5) Double rectangle represents weak Entity.



6) Double Elipse represent Multi Valued attributes.





Page No.
Date

Constraints In DBMS.

- SQL foreign Key.
  - A foreign key is a column or set of column in one table that reference (point) to the primary key column of another table.
  - This create relation between two tables. Ensuring that child table. (* foreign key) can only have Values that is there in Parent Tables (Primary key column.)

  - The Table containing foreign key is called foreign table & the table that is referenced by the foreign key is Primary table or Parent table.

  - The Primary Purpose of foreign key is To Maintain refrencial Intigrity, Ensuring the relation between table is consistant & invalid data dosent Enter the system.
- Syntax with Ex.

```
create Table Customers (cust_id int Primary Key,
                        cust_name Varchar (50) NOT NULL);
create Table Orders ( order_id int Primary key, order_no
                      int NOT NULL, cust_id int Foreign
                      Key (~~custon~~ cust_id) REFERENCES
                      Customers (cust_id);
```

Syntax:

```
Column name_datatype FOREIGN Key (Colum name)
REFERENCE tb-name (Column_name);
```

The Table where This foreign key is an Primary Key

- A foreign key is a field (or collection of fields) in one Table, that refers to Primary key in another Table.
- A Table can have multiple foreign key.

- Primary Key In SQL.
  - Primary key in SQL is a column (or group of columns) that Uniquely Identifies a record in a Table.
  - A Primary Key must contain Unique Values & cannot have any null values.
  - The can be only one Primary Key In a Table, But that Primary Key can ~~consist~~ have of one or more columns.
  - A Primary key has automatically Unique key constraint defined on It, and It ensures that that are No duplicate or Null Values in that Column.
  - No new rows can be Inserted with already Existing Primary Key.
  - Primary can be classified as

Syntax: Create Table order (cust_id int Primary Key
name varchar(30));

Page No.
Date

- AON NOT NULL Constraint.
  - The NOT Null Constraint is A Used To Enforce That a Column in a Table must always Contain a Value, it Cannot Contain a Null Value.
  - Columns in SQL Can hold Null Values To prevent it NOT NULL is Used.
  - Because Some fields like id's, Name's, roll-no's Cant be Null.
  - Not Null is Similar To primary Key Constraint Both Prevent Null Values.

Primary Key	NOT NULL.
- Uniquely Identifies Each Record in a Table.	- Simple Ensure Column is NOT NULL.

  - NOT NULL is use A To Enforce mandatory fields.
  - It Can be used during Table Creation or modification (with ALTER Command).

Syntax: Create Table Emp(Emp-id int NOT NULL Primary Key, Name Varchar (50));

- Unique Constraint.
  - The SQL Unique Constraint Ensures That all The Values in a Column are different from other one.
  - It Can be applied To one or more Column in Table.
  - If Unique is used It will reject any duplicate Value if Insert.
  - The Unique allows NULL Values Unless Explicitly defined Otherwise (a NULL is Considered as distinct Value by SQL).
  - Therfore, Multiple NULL Values are ~~not~~ allowed in Column with Unique Constraint, which is different from primary Key Constraint, where Null Values are Not allowed.

Page No.
Date

Syntax: Simply write Unique after Column.

Normalization.

Introduction.

- Primary Key & Foreign Key Rules.

A large DB defined as a single relation may result in data duplication. The repetation of data result in.

- Making relations very large.
- It isn't easy to maintain & Update data as it would Involve Searching many records in relation.
- wastage & Poor Utilization of disk Space & resources.
- The likelihood of Errors & inconsistencies increases.

- So to handle these Problems, we should analyze & Decompose the relations with redundant data into smaller, simpler & well-structured relations that are satisfy desirable properties. Normalization is a proces of decomposing the ~~et~~ ~~rel~~ relations into relations with fewer attributes.

What is Normalization.

- Normalization is The Process of Organizing The data in DB.
- Normalization is use To minimize redundancy from a relation or set of relations. It is also used To Eliminate Undesirable Characteristics like Insection, Update & Deletion Anomalies.
- Normalization divides larger Table Into Smaller and link Them using relationships.
- The normal form is Used To reduce redundancy From Database Table.

Why Do we need Normalization

- The main reason for normalizing The relations is removing These anomalies.
- Failure To Eliminate Anomalies leads To data redundancy & Can Cause Data Integridity & Other Problem as The DB grows. Normalization Consist of a ~~Serie~~ series of guidlines That helps To guide you in creating a good database Struc.

Types of Normal forms:-

- Normalization work Through series of Stages Called Normal forms.
- The Normal form ~~apply~~ apply To individual relation.
- The relation is said To be in Particular normal form if It Satisfies Constraints.