* 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 & dele 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 & Application
- 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 Catalog of Dictionary It is Called as Metadata.

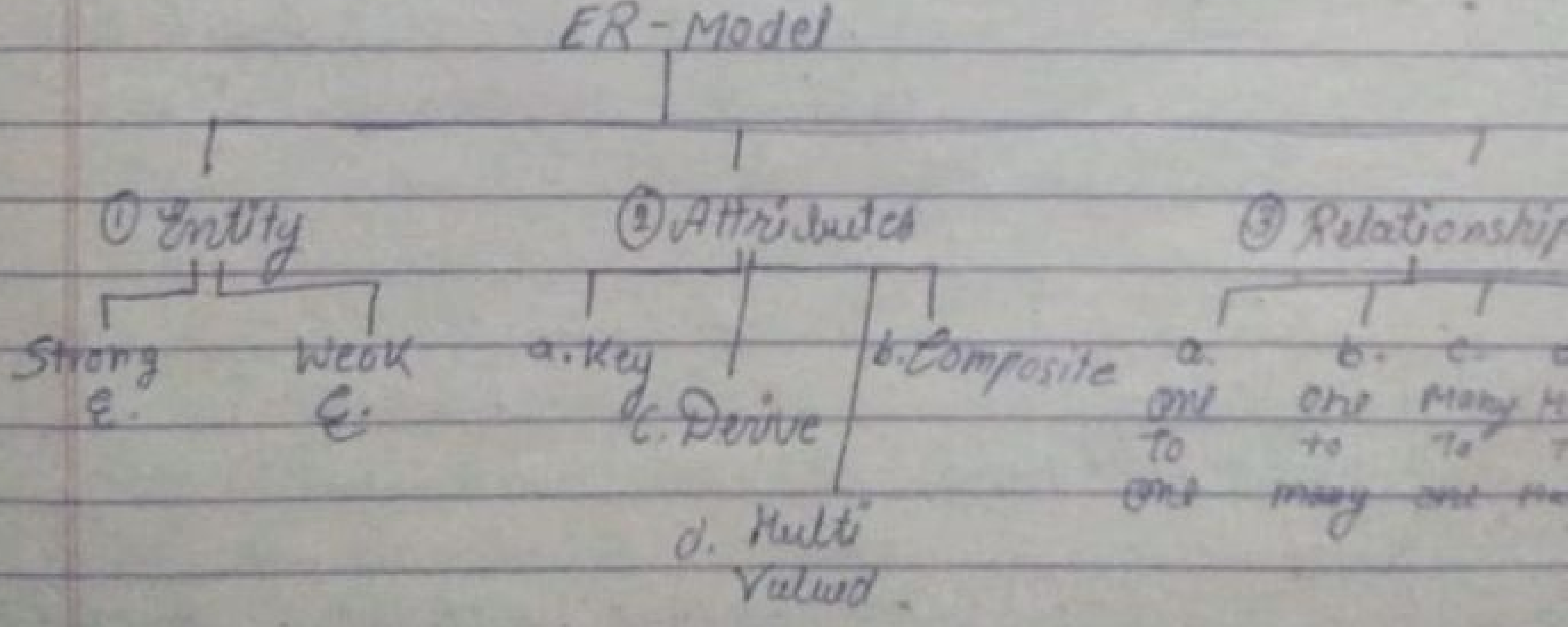
* Application.

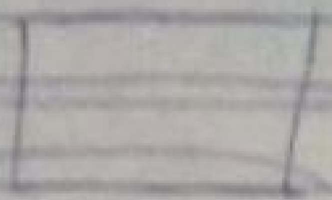
1. Enterprise Information
2. Airlines
3. Telecommunication
4. University
5. Banking & finance Sector
6. Social Media Sites
7. Manufacturing.

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

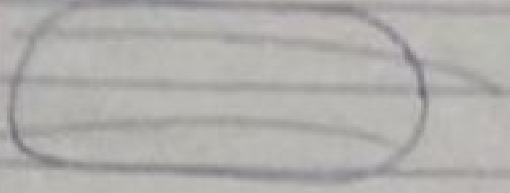
- E.R. Model



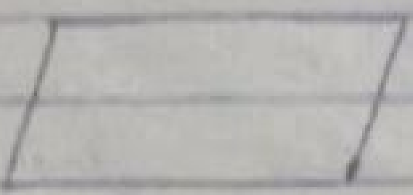


(Rectangle)

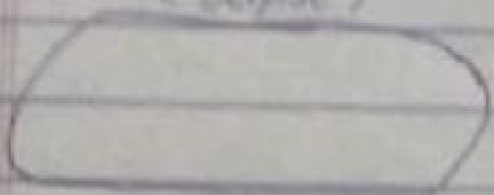
-ER Model is Used To Model The Logical View of The System from a data perspective which Consist of as follows Symbols:

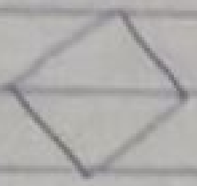


(Elipse)

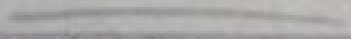
1.  - It represents Entity in The Er Model.

(Rectangle).

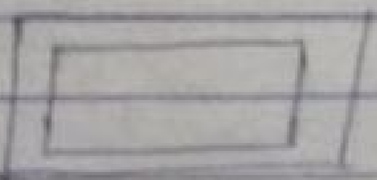
2.  - It represent attribute in Er Model

3.  - It represents Relationship among Entities.

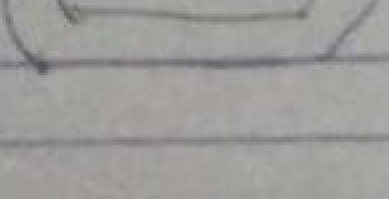
(diamond)

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

(Lines)
(Connector)
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	Disadvantages.
1. Simplicity	1. Maintainace Problem
2. Structural Independence	2. The maintaince of relational D.B. Becomes Difficult Over Time Due To The Increase In The Data.
3. Ease of use	3. Cost
4. Query Capability	4. Physical Storage.
5. few relational database have limits ~~of~~ on fields, length which cannot be Exited.	5. Complexity in Structure decrease in Performance over Time.

To maintain Data Integrity.

4. Indexes : Specific Indexing Strategies to optimize retrival.

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

6. Views - outlines Vertual 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."

Application of DBMS (Explanation)

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

2. Airlines - Client Related Data, reservation & Schedu

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

4. University - It maintain's The Information About Student, Course, Loans, Banking Transactio Enrol, Student grades, Staff ~~sal~~ roles.

. 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 & Bonds.

(Types of DBMS)

1. Relational Database Management System (RDBMS)
- Data is organised into Tables (relation) with rows & Column & 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 database store data in Various non-relational format 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 representation & relationships.

* Database Languages

DDL
(Data Defination L.)
- Create
- Alter
- Drop
- Truncate
- Comment
- Rename

DML
(Data Manipulation L.)
- Select
- Insert
- update
- Delete
- Merge
- Call
- Explain Plan
- Lock Table

DCL
(Data Control L.)
- Grant
- Revoke

TCL
(Transacti-al Contro
- Roll Bac
- Commit
- Save Poin

* Database Languages.

1) Data Defination Language.
- It deals with database Schema & descriptions. how da
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)- ~~X~~ By using alter we Can include or drop one or m
Columns from The Existing Table, also we Can inclu
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 Manupalation Language (DML)

- It Deals with data Manipulation & include 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.
- The 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 Access Data from Database.
  Syntax - SELECT * from Table-name;

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

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

  Table-name: Name of Table u want To Delete

  Condition: - The Condition That 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, Column2 = Value2, ...
  WHERE Condition;

- Merge - UPSERT operation (Insert or Update)
- Call - Call a PL/SQL or Java Subprogrammes
- 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 Transactional 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 ~~wt~~ without making any Changes Modification in Table. DQL is Very essentitial 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 Tintegrity - 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 Access - A DBMS Provide Mechanism for Controlling Concurrent Access 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 Access To The data & Encrypting Sensitive Data.

5) BackUP & 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 usefull in a Collabranative 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, especially 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 Costomer 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 Course enrolled Info., Students grades & Staffs Role.

4) Airlines Reservations & Schedules.

5) Telecommunication - Prepaid & Postpaid Bill Maintainance.

- हम इस Schema को एक logical representation है, of Data जो ये बताता है कि कैसे (how) डेटा Store होना chahiye Database में that in a logical manner.
- यह बताता है कि कैसे डेटा organize किया जाता है &
- Relation between tables & इसे 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 Emplement 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 ~~Unstructed~~ 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 Per of Time Per To Information available होती है उसे Instance कहते है।

* Instances In a Database.

- The Instance of database is The Values of These Variables at any given Time. Instances are also Called Current state or Database State. The Database Schema is a Design That define The Variables in The Tables That belong To a Perticular Database. There May be many Instances That Correspond To Certain Database Schema. The new 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.

Orderid	item	Amount	Customer id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	keyboard	400	1
5	MousePad	850	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) are 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.

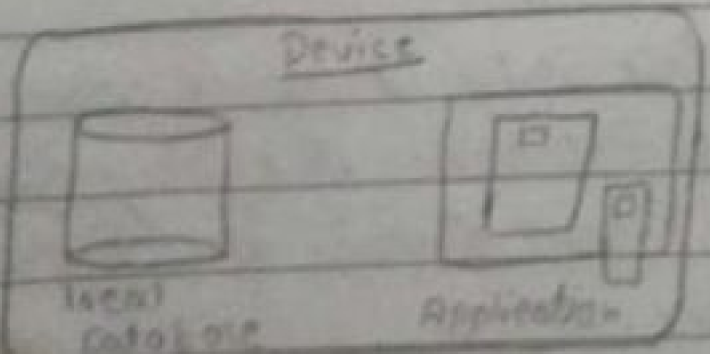
- Geeks Greeks Info!

- 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 DataBase file & Performs operation like Querying, Inserting, Updating or Deliting Records.

- There is No Sepeeation Between The Application & The Database Since Both resides on a Same Machine.



* Advantages of 1-Tier Architeture.

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 Architeture, which makes it Cost Effictive.

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

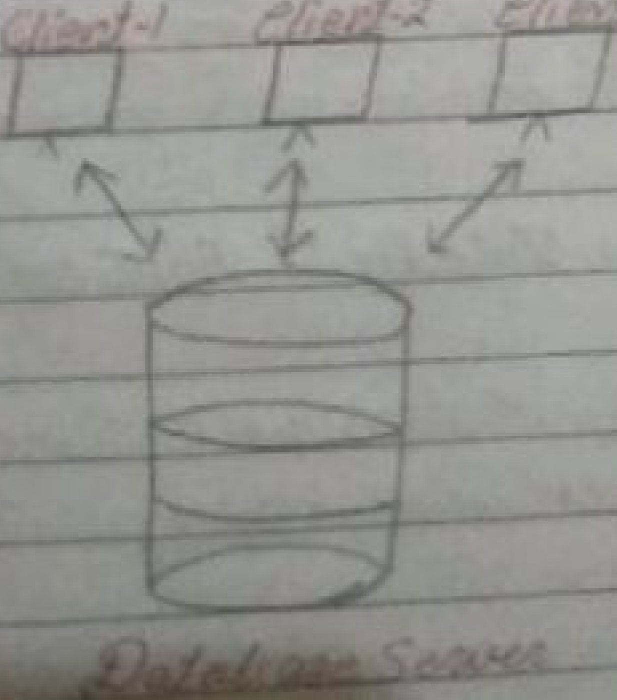
* Disadvantage

1) Sculable - only one User Can Acess 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 Architeture.



- Two Tier Mean 2 layer Here are Two Layers one Client layer & Second database Layer.
- Here, Client is a p Machine in which a Interface is running & This interf 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.

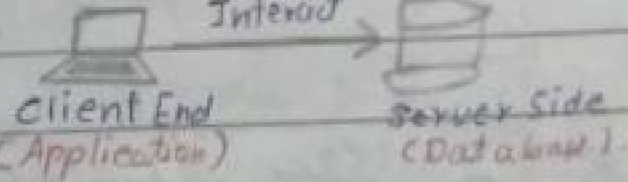
- This is how 2-Tier Architecture works.
- Limited Clients & Database hence Maintainance is Easy.

Problems

- Scalability.
- Security- client is directly Interacting with DataBase

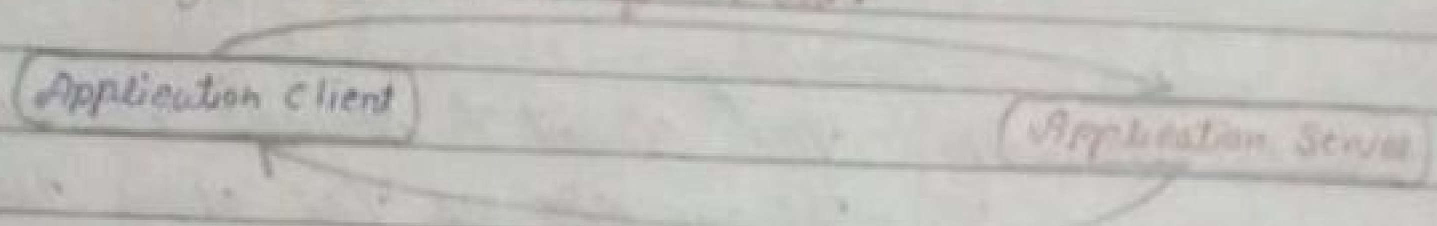
(Geeks for Geeks).

- The 2-Tier Architeture is Similiar To a basic Client-Server Model.
- The Application at The Client End Directly Communicat 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 progra are run.
- The Application on Client Side Establishes a Connection wit 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.
~~2. Scalability~~ - which Can Expose The Sensitive Data.
- Its harder To protect System from Security Threats.

2. Scalability - - As more users Connect, The System or
- Database gets Overloaded, Slowing Things Down.
- Its harder To Expand System to handle more users.

* ~~2nd Tier Architecture.~~

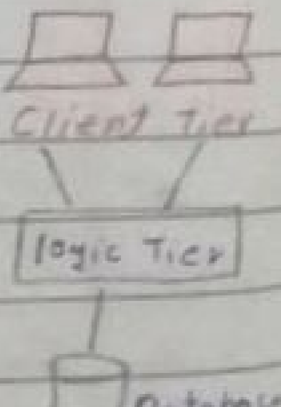
3) Single Point of failure: If Server or Database goes Down. The whole System Stops working, becau There is No Back-up Layer To Keep Thing running.

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

* 3rd Tier Architecture.

- Ino Three- Tier Architecture for DBMS, The System i Devided in 3 Distinct Layers. - Each with Specific role.
- These structre improve ~~Seatatrail~~ Scalability, maintainabilit & Security by ~~spe~~ seperating The Different responsibilitie

1. Presentation Tier (Client Tier):
- Role: - This is The Topmost Layer That Interacts with The user. Its 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 & recieve The Processed Data from Middle layer & Present The result To user in a readable form (like a web Page or a Appscreen).

2. **Logic Type Tier (Application / Business Logic Ty Tier):**

- **Role** :- This Layer Acts as an Intermediary between The Client Tier & The Database Tire.
  - This Layer handles The Business Logic, Processing of Data & Performs any Necessary Calculations or Transformat-ion Before Sending It To Database or Back To The User.
- **Ex** :- Web Servers, Application Servers or Business Logic Engines.
- Funtions :- 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.

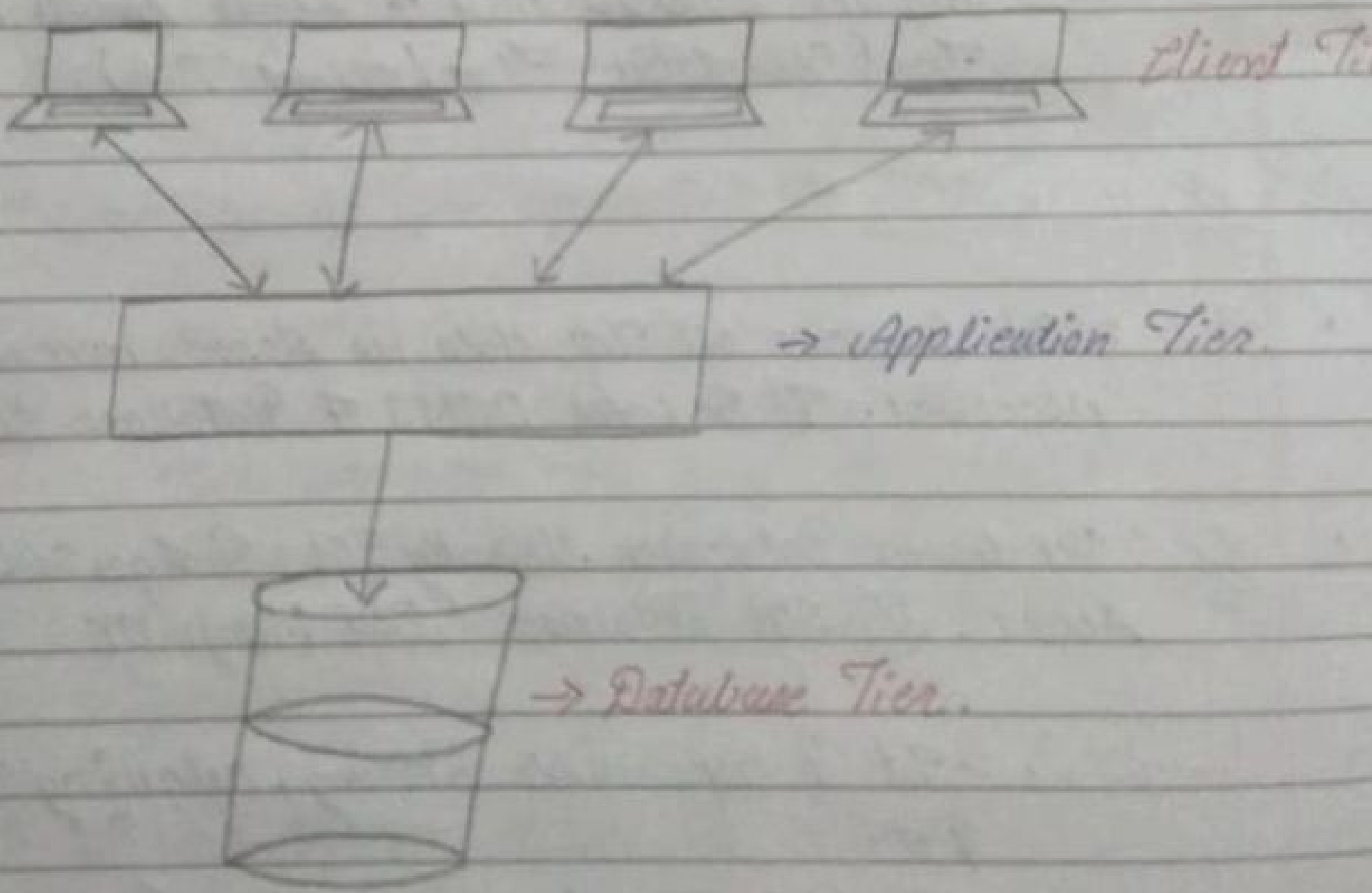
* Workflow in a 3-Tier Architecture.

1. The user Interacts with The Presentation (web Interfac

2. The request is sent To The Application Tier, which P The logic(request) & Communicate with The Database Tie

3. The Database Tier Execute Queries, returns Data. & Sen Back To Application Tier.

4. The Application Tier Than return The Processed Data to Presentation Tier which Displays It To Users.



3-Tier Architecture.

Advantages	Dis-Advantages.
1. Each layer is Seperate, which make it Easier To manage & update Each part without affecting other.	1. Complexity - Managing & Maintaini 3 different layers can be more Complicated Than a Simple Architecture.
2. You can Scale Each Tier Independently. Ex you can Emclude More servers To Database Tier without affecting Presentation layer.	2. Performance Overhead: Commun-cation Between Tiers can can ~~Include~~ lead To Delay o latency & reduce overall system Performance.
3. By Seprating The Data & Bussiness logic layers from Presentation layer, Sensitive Data can be better Protected & Managed.	3. Cost: More resources (Serves, Infrastructure) may be needed manage Each Seperate layer, Increasing The Overall Cost.

* Data Models.

- Data Models are Mainly Usefull in order To Design The DataBase.
- Data Model ~~zarkt~~ is 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.

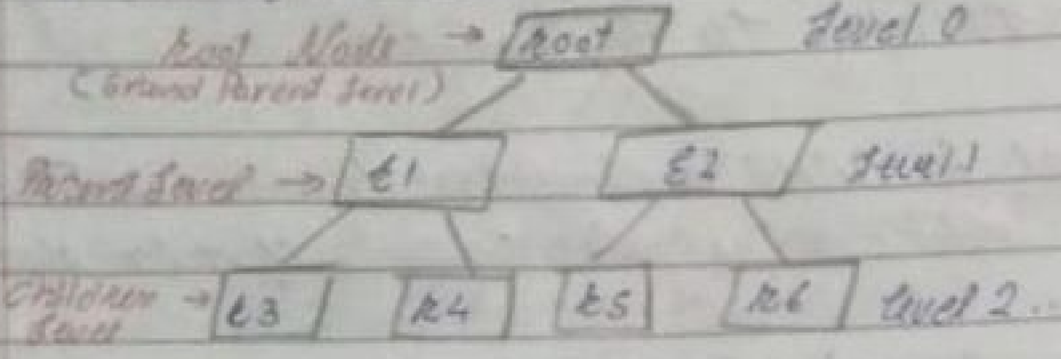
- My SQL - A widely-used open-source relational DBMS. It stores Data in Tables & is used by Many web ~~Publications~~ Applications.

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

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

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

- It is Mainly Used To Store The Information in a Hirarchial or Level by Level Manner.
- Grand Parent Level, Parent Level, Children Level.
- It forms a ~~star~~ 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 Hirarchial 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.

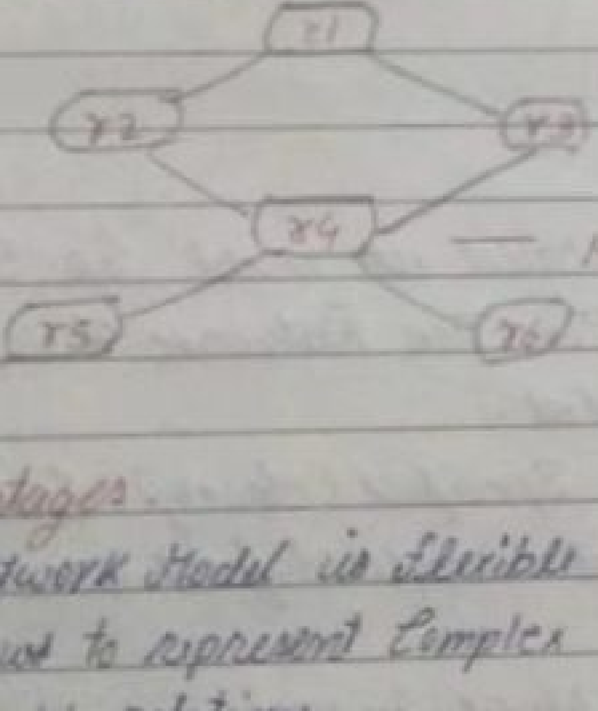
1. It has Very Simple hirarchial DB structure.
2. It has Data sharing, as all Data are held in Common DB.
3. It offer Data Security.

Disadvantages

1. lacks flexibility. Deletion of one Segment Can lead To Deletion of all Seg. under it.
2. It has no standard.
3. It is also limited as many of Common 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.
- I Here Each Node Can have many children & Parents.
- It is an Extension of Hirarchial Data Model.

- The Network Data Model is one of The Most oldest Data Model That was designed To handle Complex Data relationships more effectively Then The Hirarchial Model.
- In Network model Data is Organised in a Graph Structure which allows for a 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 Directs.



— Here R4 is associated with Multiple Record in Both Direct

Advantages.

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

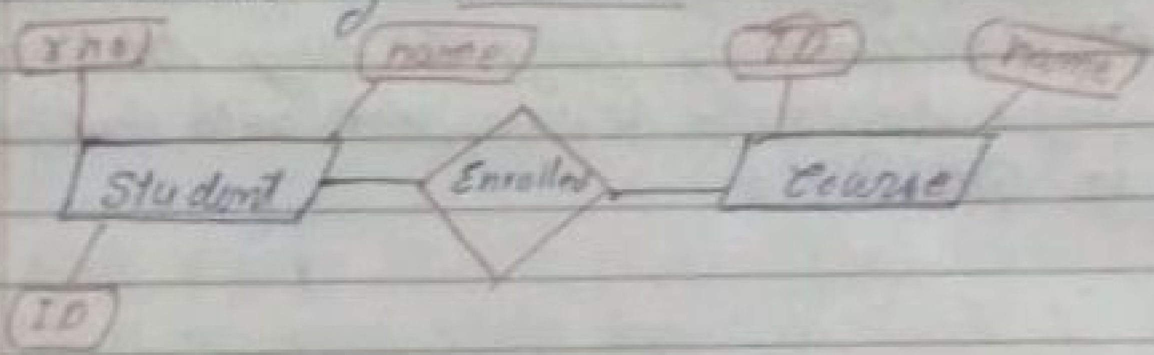
2. The Network model allow many-to-manyR. This is usefull in Cases where an Entity might have Multiple relation with others.
3. Data Integrity
4. Supports for Multiple Parent records.

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 Elipse
- relations By Diamound.



- The Entity relationship Model is model for Identifying entities To be represented In The Database & representation of how These 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 Independent 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.