Data-base Mangement System

Data

-> Data is a collection of a distinct small unit of information

-> It can be used in a variety of forms like text, numbers, media etc.

-> Word data is originated from word 'datum' that means "single piece of information".

Database

-> A database is an organized collection of data, so that it can be easily accessed & manged.

-> We can organized data into tables, rows, columns & index it to make it easier to find relevent information

-> The main purpose of the database is to operate a large amount of information by storing, retrieving & managing data

-> There are many database available like MySQL, Sybase Oracle, MongoDB, SQL server etc.

Database mangement system

-> Database mangement system is a software which is used to manage the database

ex. MySQL, Oracle

-> This are very popular commercial database which is used in different applications.

-> DBMS provides an interface to perform various operation like database creation, storing data in it, updating data, creating a table in the database & a lot more.

-> It provides protection & security to the database

★ DBMS allows users for the following tasks.

1) Data definition :- It is used for creation, modification & removal of definition that defines the organization of data in the database

2) Data updation :- used for insertion, modification & deletion of actual data in the database.

3) Data Retrieval :- retrieve the data from database which can be used by application for various purposes.

★ characteristics of DBMS

→ ~~DBMS con~~ Data organization :- DBMS helps organize & store data in structured manner

→ Data integrity :- It ensures data accuracy & Consistency

→ Data security :- Provides user authentication & authorization to control who can access & modify the data

→ Data Retrieval :- user can easily retrieve data using queries, Searching & filtering mechanism

→ Backup & recovery :- Create backups of the data & recover it in case of system failures or data loss.

→ Multi-user Support :- DBMS allows multiple users to access & modify data simultaneously

→ Data Compression :- Compress data to reduce Storage Space

★ Data Abstraction

→ Data abstraction is a process of hiding unwanted or irrelevant details from the end user.

→ It allows users to interact with database without needing to understand the underlying data structure & storage mechanism.

→ The database system consist of complicated data structure & relation.

→ For users to access the data easily, these complication are kept hidden

→ Only the relevent Part of the database is made accessible to the users through data abstraction.

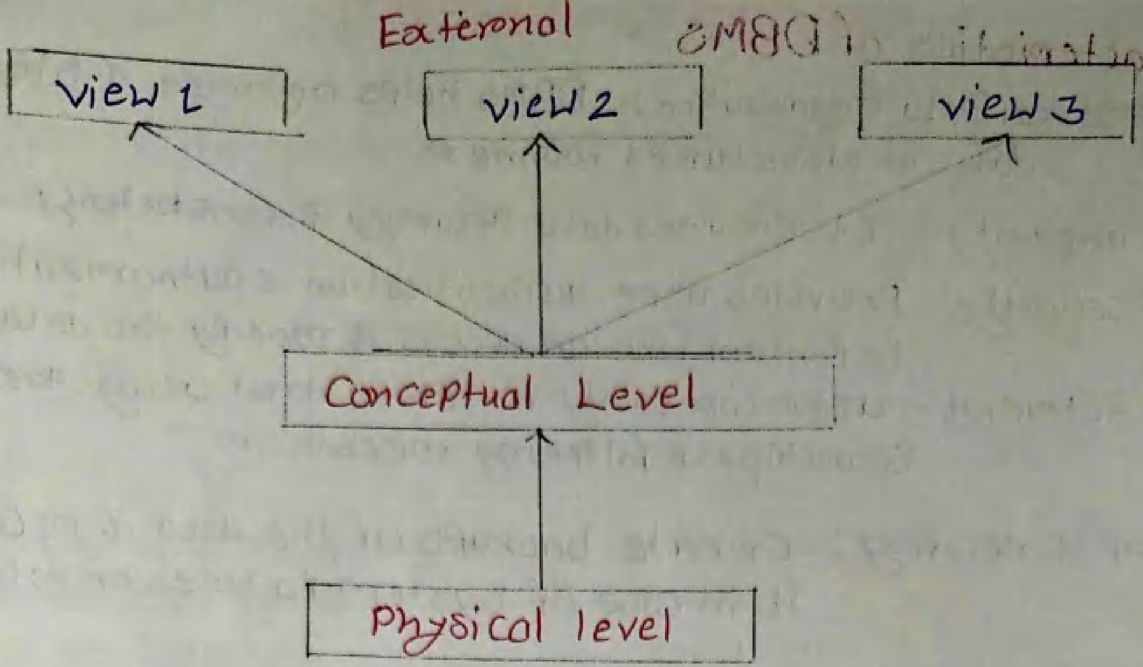
→ The developers use level of abstraction that hide irrelevant details from the users. to reduce complexity & make the System efficient.

→ There are three level of Data abstraction

1) Physical level

2) Logical or Conceptual level

3) View or External level



1) Physical level

-> It is the lowest level of abstraction for DBMS which defines how the data is actually stored

-> It defines data structure to store data & access method used by the ~~datable~~ database

-> It deals with how data is stored on storage devices such as hard disks & it includes like file organization, indexing & data storage formats.

-> It is very complex level to understand.

ex. customer information is stored in tables & data is stored in the form of blocks of storage such as ~~bytes~~ bytes, gigabytes.

2) conceptual level (Logical)

-> Logical level is the intermediate lever or next higher level.

-> It describes what data is stored in the database & what relationship exists among those data.

-> It describes the structure of the entire data in the form of tables.

-> The logical level or conceptual level is less complex than the physical level.

3> view level

-> It is the highest level

-> In view level, there are different levels of views &

-> An every view only defines a Part of the entire data

-> It Provides many views of same database.

-> view level can be used by all users. This level is the least complex & easy to understand

ex. students & teachers
student can see or access their marks & data hence it has own view
teachers can edit marks, give leave appl^n or other info also it has there own view.

★ Data models

-> Data models define how the logical structure of database is modeled

-> Data models are fundamental entities to introduce abstraction in a DBMS.

-> Data models define how data is connected to each other & how they are Processed & stored inside the system

-> Data models
i> Relational Data model
2) Semistructured
3> Entity Relationship
4> object-based

I> Relational

-> This type of model designs the data in the form of rows & columns within a table.

-> Relational model uses tables for representing data & in-bet^n relationships.

-> Tables are also called relations.

-> The relational model is widely often used model which is Primarily used by commercial data Processing applications.

2> Entity-Relationship Data model

→ An ER model is the logical representation of data as objects & relationships among them.

→ These objects are known as entities & relationship is an association among these entities.

→ It was widely used in database designing

→ A set of attributes describe the entities
for example :- Std_name, Std_ID describes 'Student Entity'

→ A set of same type of entities is known as Entity set & set of same type of relationships is known as 'relationship-set.

3> Object-based Data model

→ An extension of the ER model with notions of functions, encapsulation & object identity.

→ This model supports a rich type system that includes ~~structured~~ structured & collection types.

→ Here objects are nothing but the data carrying its properties.

4> Semistructured Data model

→ This type of data model is different from the other three data models

→ It allows the data specifications at places where the individual data items of same type may have diff. attributes set.

→ The Extensible Markup Language also known as XML, is widely used for representing semistructured data.

→ It gains importance because of its application in exchange of data.

★ Data model schema & Instances

→ The data which is stored in the database at a particular moment of time is called an instance of database

→ The overall design of a database is called schema

→ A database schema is the skeleton structure of the database. It represents the logical view of entire database.

→ A schema contains schema objects like table, foreign key, Primary key, views, columns, data types etc.

→ A database schema can be represented by using visual diagram That diagram shows database objects & relationship with each other.

★Entity

→ Entity in DBMS can be a real-world object with an existence,
eg. in a college database, the entities can be Professor, student, courses etc.

→ Entity has attributes which can be considered as Properties describing it.
eg. Prof. Name, Address, Salary etc

1) Strong Entity

→ It has a Primary key & weak entities are dependent on strong

→ It's existence is not dependent on any other entity.

→ It is represented by single rectangle.

eg. In the above eg. Prof. is strong entity & it's ID is Primary key

2) Weak Entity

→ The weak entity in DBMS donot have a Primary key & are dependent on Parent entity.

→ It mainly depends on other entities.

→ It is represented by double rectangle

Attributes :- Attributes are the Properties which describe an entity

eg. attributes of student entity

Roll no.
Name
Branch
Age.

1) composite :- It can be divided into smaller sub Parts, each Sub Part can form an independent attribute.

eg. Name

First Name Middle Name Last Name

2) Simple or Atomic attribute

→ Attributes that cannot be further subdivided are called atomic attributes.

eg. Phone no.
PIN ~~co~~ Code

3) Single valued Attribute

→ Attributes having a single value for a Particular item is called single valued Attribute.

eg. Age of student

4) Multi-valued attribute

→ The attribute which takes up more than a single value for each entity instances is a multi-valued attribute.

eg. Phone no. of student : Landline & mobile

5) Derived or Stored attribute

→ When one attribute value is derived from other.

eg. Age can be derived from DOB

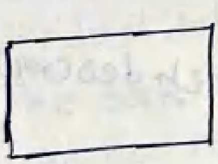
★ ER Diagram

→ ER Diagram stands for Entity Relationship Diagram or ERD

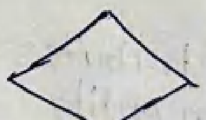
→ ER Diagram stands for Entity Relationship Diagram or ERD
→ It is a diagram that displays the relationship of entity sets stored in a database.

→ ER diagrams help to explain the logical ~~to~~ structure of database

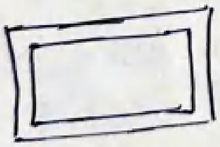
→ ER diagrams help to explain the logical ~~to~~ structure of database
→ ER diagram contain different symbols to represent relationships



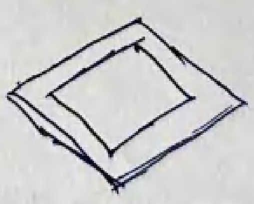
⇒ Strong Entity



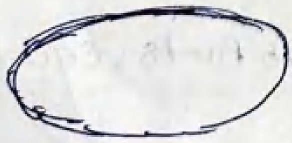
⇒ Relationship



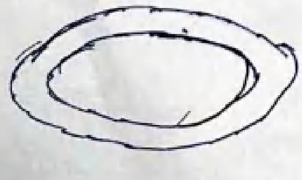
⇒ Weak entity



⇒ Weak Relation-ship



⇒ Attribute



⇒ Multi-valued attribute

Relationship :- Relationship is nothing but an association among two or more entities

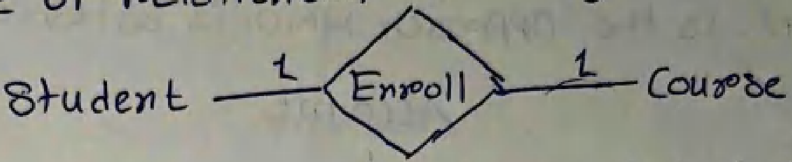
e.g. Tom works in chemistry department

★ Cardinality :-

⇒ Defines the numerical attributes of the relationship betn two entities or entity sets.

1) one-to-one Relationship

⇒ This type of relationship is rarely seen in real world



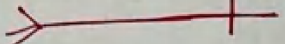
~~The~~

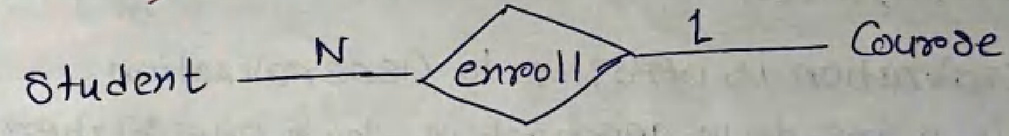
The above example describe that one student can enroll only one course & course will also have only one student

symbol :-

2) Many to one Relationship

→ Many entities can be associated with just one entity. For example students enrolls for only one course but a course can have many students.

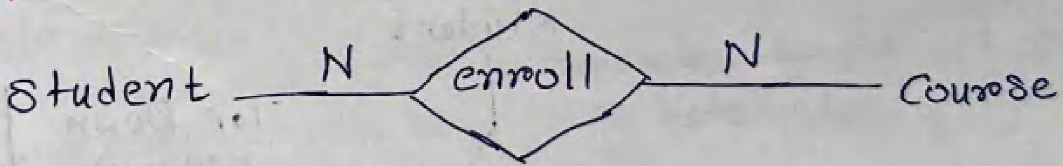
symbol :- 



3) Many to many Relationship

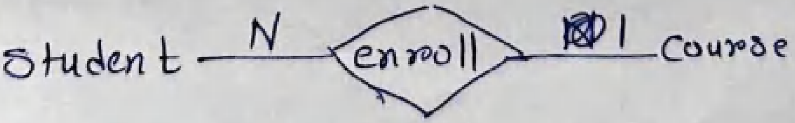
→ e.g. one student can enroll for more than one course. And a course can have more than 1 student enrolled in it.

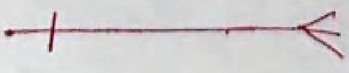
symbol



4) one to many

→ one entity can be associated with multiple entities



symbol :- 

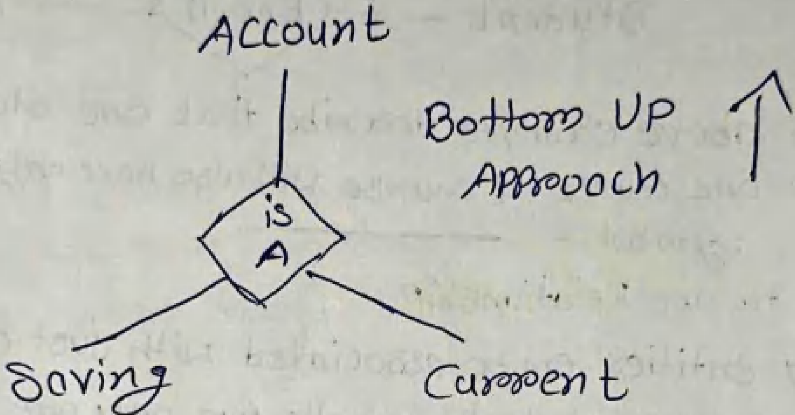
★ Thero enhonced ER model

1) Generalization

→ Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity

→ In generalization, the higher level entity can also. Combine with other lower level entities to make further higher level entity

→ It's more like Superclass & subclass system, but the only diff. is the approach which is bottom up.

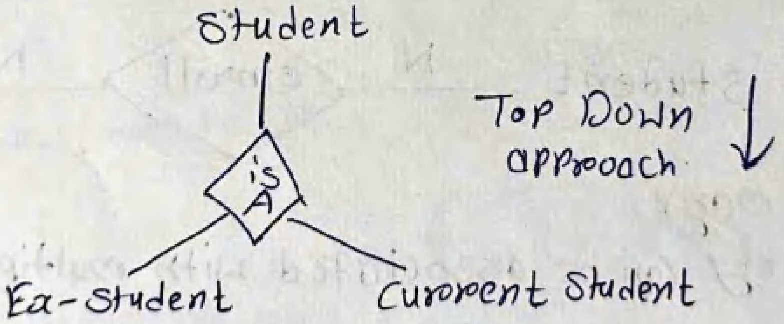


2) specialization

→ specialization is opposite to Generalization.

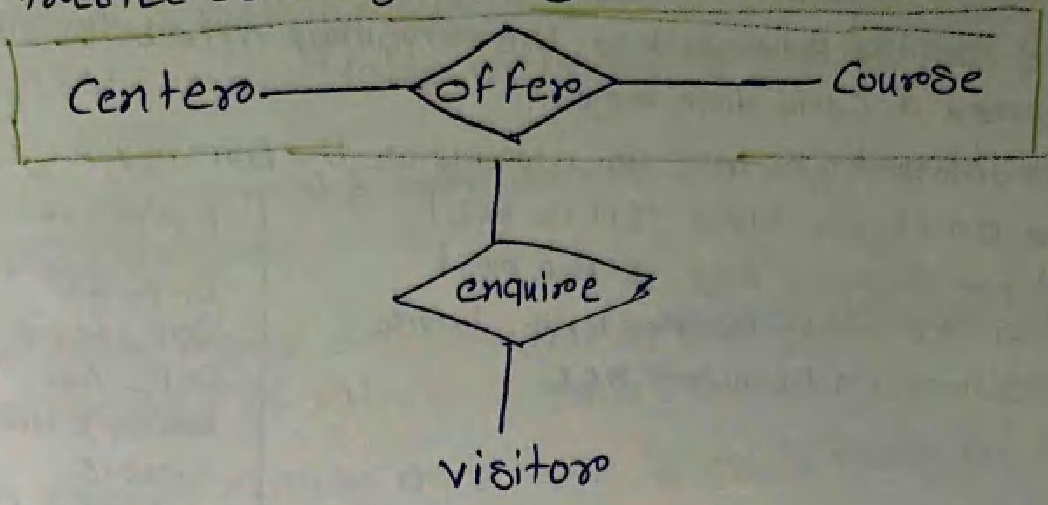
→ It is a top-down approach in which one higher level entity can be broken down into two lower level entity.

→ In specialization, a higher level entity may not have any lower level entity sets, it's Possible



3) Aggregation

→ Aggregation is a process when relation betⁿ two entities is treated as a single entity.



→ ~~In the~~

→ In the above diagram the relationship betⁿ center & course together, is acting as an Entity which is in relationship with another entity visitor

★ ~~DB~~ keys

→ keys play 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 & identify relationships betⁿ tables.

1) Primary key.

→ It is the First key used to identify one & only one instance of an entity uniquely.

→ An entity can contain multiple keys, ~~as we saw in the~~ but the key which is most suitable from those lists becomes a primary key.

→ The Primary key it is unique for the whole entity

→ for ex. Adhar no., PRN etc.

2) Candidate key

→ A Candidate key is an attribute or set of attribute that can uniquely identify a tuple.

→ Except for the Primary key, the remaining attributes are considered a candidate key.

→ The candidate keys are as strong as the Primary key.

e.g. In the employee table ID is best suited for Primary key & the rest attributes like SSN, Passport no., licence are considered a candidate key.

Employee
Emp_ID
Emp_Name
Emp_Add
Passport No.
Licence
SSN

3) Super key

→ Super key is an attribute set that can uniquely identify a tuple

→ A super key is a superset of candidate key.

→ e.g. In the above employee table for (ID, Name) the name of two employees can be the same but their ID can't be same Hence this combination can also be a key.

→ The super key would be Emp_ID (ID, Name) etc.

4) Foreign key

→ Foreign keys are the column of the table used to point to the Primary key of another table.

→ Every ~~empot~~ employee works in specific department in company & ~~empb~~ employee & department are two diff. entities.

→ So we can't store the department's information in the employee table, That's why we link these two tables through the Primary key of one table.

→ We add the Primary key of Department table, department_ID as a new attribute in Employee table.

→ In the Employee table, Department_ID is the foreign key & both the tables are related.

* Relational model

→ Relational model represent the database as a collection of relations.

→ A relation is nothing but a table of values.

→ Every row in the table represents a collection of related data values

→ These rows in the table denote a real-world entity or relationship.

→ The table name & column name are helpful to interpret the meaning of values in each row.

→ Each row is known as a tuple, & each table of the column has a name or attribute

Domain :- It contains a set of atomic values that an attribute can take.

Attribute :- Each column in a table, attributes are the properties which define a relation eg. Roll no., Name etc

Tables :- In the relational model the relations are saved in the table format.
A table has two properties rows & Column. Rows represen
records & columns represent attributes.

Tuple :- It is nothing but a single row of a table, which contains a single record.

Relational schema :- A relation schema represents the name of the relation with its attributes.

Degree :- The total number of attributes which in the relation is called the degree of the relation

Cardinality :- The total no. of rows present in the table

Column :- The column represents the set of values for a specific attribute

Relational instance :- Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.

Relation key:- Every row has one, two or multiple attributes which is called relation key.

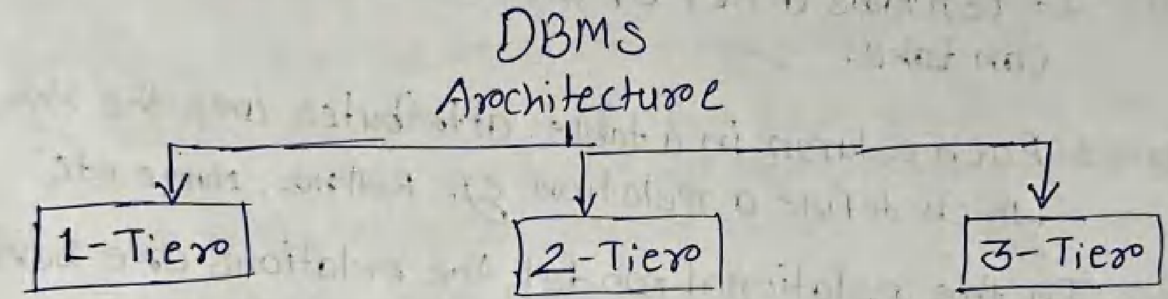
★ Architecture for dbms (client/server)

→ The DBMS design depends upon its architecture.

→ The basic client/server architecture is used to deal with a large no. of PC's, web servers & other components that are connected with networks.

→ The client/server architecture consists of many PCs and a workstation which are connected via the network.

→ DBMS architecture depends upon how users are connected to the database to get their request done.



I) 1-Tier Architecture

→ In this architecture, the database is directly available to the user.

→ It means the user can directly sit on the DBMS & uses it.

→ Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.

→ The 1-Tier architecture is used for developement of the local application.

→ where programmers can directly communicate with the database for the quick responce.

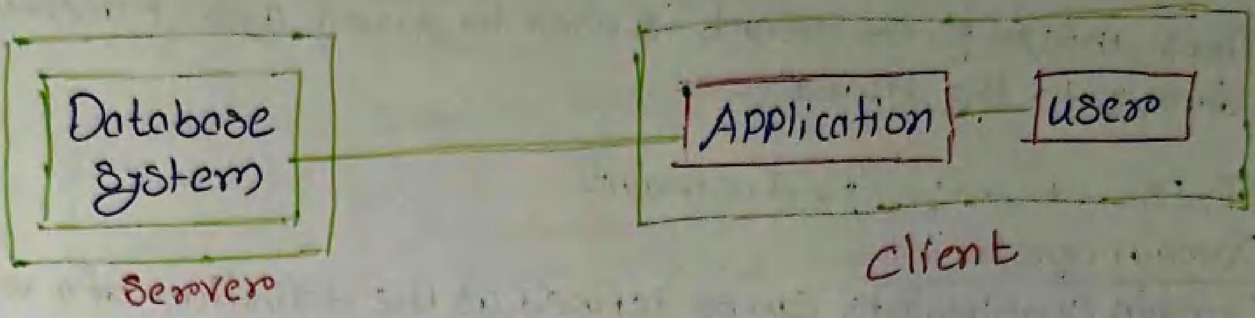
2) 2-Tier Architecture

→ The 2-Tier architecture is same as basic client-server

→ In the two tier architecture; applications on the client end can directly communicate with the databases at the server side.

→ For this interaction API's like: ODBC, JDBC are used.

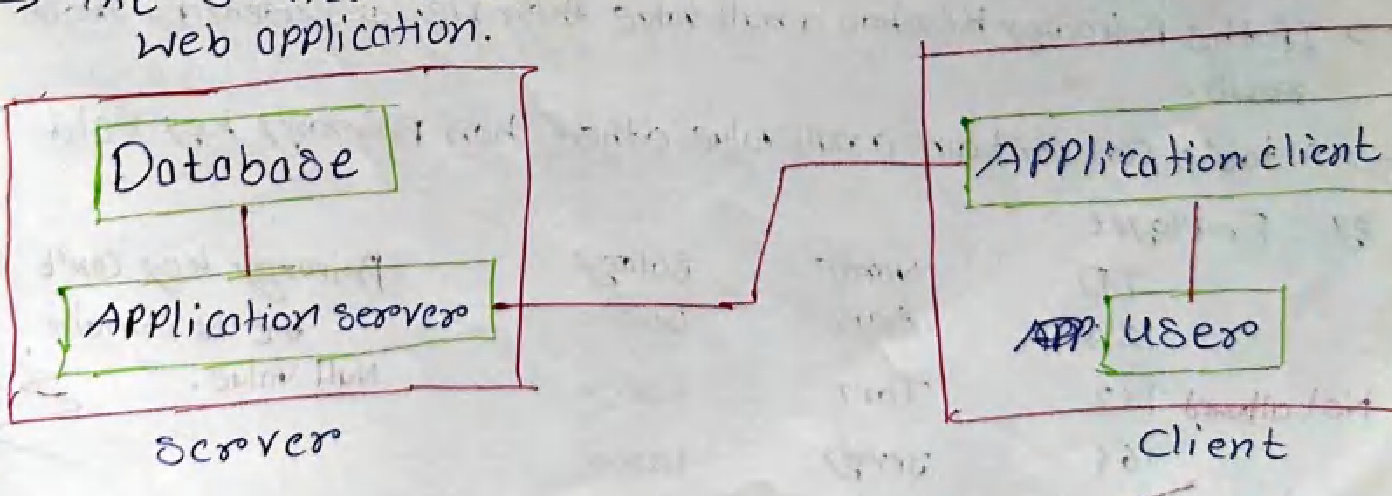
- → The user interfaces & application programs are run on the client side.
- → The server side is responsible to provide the functionalities like query processing & transaction management.
- → To communicate with the DBMS, client side application establishes a connection with the server side.



2-Tier Architecture

III) 3-Tier Architecture

- → The 3-Tier architecture contains another layer bet^{n} client & server.
- → In this architecture, client can't directly communicate with the server.
- → The application on the client-end interacts with an application server which further communicates with the database system
- → The user has no idea about the existence of the database beyond the application server
- → The database also has no idea about any other user beyond the application.
- → The 3-Tier architecture is used in case of large web application.



★ Integrity Constraints

→ Integrity constraints are a set of rules. It is used to maintain the quality of information.

→ Integrity constraints ensure that the data insertion, updating & other processes have to be performed in such a way that data integrity is not affected.

→ Thus, integrity constraint is used to guard against accidental damage to the database.

Types of Integrity constraint

1> Domain constraint

→ Domain constraints can be defined as the definition of a valid set of values for an attribute.

→ The data type of domain includes string, character, integer, time, date, currency etc.

→ The value of the attribute must be available in the corresponding domain

ex.

ID	Name	Sem	Age
1000	Tom	Ist	18
1001	John	IInd	20
1002	Leo	IVth	22
1003	Sam	VIIth	~~20~~A

→ Not allowed because Age is an integer attribute

2> Entity Integrity constraints

→ The entity integrity states that primary key value can't be NULL

→ This is because the primary key value is used to identify individual rows in relation.

→ If the primary key has a null value then we can't identify those rows.

→ A table can contain a null value other than primary key field

eg. Employee

ID	Name	Salary
123	Sam	30000
142	Tom	48000
164	Jerry	18000
	Jack	170000

Not allowed

Primary key can't ~~be N~~ contain Null value.

3) Referential Integrity Constraints

→ A referential integrity Contraints is specified bet^n two tables.

→ In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary key of Table 2.

→ Then every value of foreign key in Table 1 must be null or be available in Table 2

ex.

ID	Name	Age	D-no.
1	Jack	20	11
2	Tom	27	24
3	Sam	17	18
4	Ram	19	13

← Foreign key

Relationship

Primary key

D_no	D_location
11	Mumbai
24	Delhi
13	Noida

Not allowed as D-no. 18 not defined as Primary key of table 2 & in table 1 D-no is a foreign key defined.

4) key constraints

→ keys are the entity set that is used to identify within its entity set uniquely

→ An entity set can have multiple keys, but out of which one key will be the Primary key.

→ A Primary key can contain a unique & null value in the relational table

eg.

ID	Name	Sem	Age
1000	Tom	1st	17
1001	Sam	3rd	24
1002	Jam	2nd	21
1003	Ram	4th	19
1002	Om	7th	22

↓

Not allowed, Because all row must be unique.

★ Relational Query langauge

→ Relational Query langauge is used by the user to communicate with database.

→ They are generally on a higher level than any other programing langauge

→ This is farther divided into two types

1) Procedural
2) Non-Procedural

1) Procedural

→ Information is retrieved from the database by specifying the sequence of operations to be performed.

For ex. Relational Algebra

→ Structured Query langauge (SQL) is based on relational algebra

→ Relational algebra consists of set of operations that take one or two relations as an input & produces a new relation as output.

eg. select, Project, Rename operation.

2). Non-Procedural langauge

→ Information is retrieved from the database without specifying the sequence operation to be performed.

→ users only specify what information is to be retrieved

eg. Relational calculus.

→ Relational calculus is non-procedural query langauge in which info. is retrieved from the database without specifying sequence of operation to be performed.

→ The Relational calculus is of two types

1) Tuple calculus
2) Domain calculus.

★ Relational algebra

→ Relational algebra is a Procedural query language.

→ Which takes instances of relations as input & Produce relations as output

→ It uses operators to Perform queries, an operator can be either unary or binary

→ The fundamental ~~as~~ operations of relational algebra are as follows

1) Select (σ)

→ The select operation is used to select a subset of the tuples from a relation that satisfy a selection condition

→ One can consider the select operaction to be a filter that keeps only those tuples that satisfy a qualifying condition

eg. I) to select the ~~Empoly~~ Employee tuples whose department is 4.

$\sigma_{Dno=4}$(Employee)

II) whose Salary is greater than 30000 in Employee tuple

$\sigma_{Salary>30000}$(Employee)

Notation :- $\sigma_p(r)$

σ => used for selection Prediction

r => is used for relation

p => It is used as Propositional logic formula which may use Connectors like AND, OR & NOT, Also these terms may make use of relational operators such as =, ≠, <, >, ≥, ≤ etc.

2) Project operation (π)

→ The select operation selects some of the rows from the table while discarding other rows.

→ The Project operation on other hand. selects certain column from the table & discards the other columns.

→ If we are interested in only certain attributes of a relation, we use the Project operation to Project the relation over these attributes only.

ex. suppose ~~@~~ Customer Relation :- ~~Name F-name, L-name, G~~ Name, Payment mode, CITY

$\pi_{Name, City}$(Customer)

→ The Project operation shows only Name & City discard Pay-mode

★ Structured Query language

Characteristics

→ It is easy to learn

→ It is used to access data from relational database management systems.

→ SQL can execute queries against the database.

→ SQL is used to describe the data

→ It is used to define the data in database & manipulate it when needed.

→ It is used to create & drop the database & table

→ SQL used to create a view, stored Procedure, function in a database.

Advantages

1> Faster query Processing

→ Large amount of data is retrieved quickly & efficiently operation like insertion, deletion, manipulation of data is also done in almost no time.

2> No Coding skills :-

→ For data retrieval, large number of lines of code is not required. All basic keyword such as SELECT, INSERT INTO, UPDATE etc are used.

(3) Standardized language

→ Due to documentation & long establishment over years, it Provides a uniform Platform worldwide to all its users.

4) Portable

→ It can be used in Programs in PCs, server, laptops independent of any Platform.

5) Interactive language

→ Easy to learn & understand, answers to complex queries can be received in seconds.

6) Multiple data views

7) Security :-

→ SQL databases have built-in security features that help Protect data from unauthorized access.

8) Backup & Recovery

→ SQL databases have built-in backup & recovery tools that help recover data in case of system failures, crashes or other disasters.

★ SQL Commands

→ SQL Commands are instructions. It is used to communicate with the database.

→ It is also used to perform specific tasks, functions & quiries of data.

→ SQL can perform various tasks like create a table, add data to tables, drop the table, modify table, set permission for users.

★ Types of SQL commands

→ There are five types of SQL commands:
DDL, DML, DCL, TCL & DQL

1> Data Definition Language (DDL)

→ DDL changes the structure of the table like creating a table, deleting a table, altering a table etc.

→ All the command of DDL are auto-commited that means it permanently save all the changes in the database

I> CREATE

→ It is used to create a new table in the database

→ The user has to give information like table name, column names & their datatypes.

Syntax:-

```
CREATE TABLE table_name (column_1 datatype, column_2 datatype);
```

eg.

```
create table Student (ID int, fname varchar(30), lname varchar(30));
```

II> ALTER

→ This command is used to add, delete or change columns in the existing table.

→ The user needs to know the existing table name & can do add, delete or modify tasks easily

1) To add a new Column in the table

ALTER TABLE table_name ADD Column_name COLUMN-definition

ex. ALTER TABLE ~~table_name~~ student ADD (Address varchar(30));

2) To modify existing Column in the table

ALTER TABLE table_name MODIFY (Column-definitions...);

ex. ALTER TABLE student MODIFY (Name varchar (20));

III) DROP

→ It is used to delete both the structure & record stored in the table.

Syntax:- DROP TABLE table_name;

ex. DROP TABLE Student;

IV) TRUNCATE

→ It is used to delete all the rows from the table & free the space containing the table.

Syntax:- TRUNCATE TABLE table_name;

eg. TRUNCATE TABLE Student;

2) DATA Manipulation Language

→ DML Commands are used to modify the database. It is responsible for all form of changes in the database.

→ The command of DML is not auto-committed that means it can't Permanently save all the changes in the database

a) INSERT

→ INSERT statement is a SQL query.

→ It is used to insert data into the row of table

Syntax:- INSERT INTO table_name (Col1, Col2, Col3.....)
VALUES (val1, val2, val3, ...);

OR

```
INSERT INTO table_name
        VALUES (val1, val2. . . N);
```

b> UPDATE

→ This command is used to update or modify the value of column in the table

Syntax:-
```
UPDATE table_name SET (col_name1 = val1, col_name2 = val2 . . . N)
        WHERE (condition);
```

eg.
```
UPDATE Students
  SET user_name = 'Jitu'
   WHERE Std_Id = '3' ;
```

c> DELETE

→ It is used to remove one or more row from a table

Syntax:-
```
DELETE FROM table_name [WHERE condition];
```

eg.
```
DELETE '
```

★ VIEW in SQL

→ views in SQL are considered as a virtual table. A view also contains rows & columns

→ To create the view, we can select the fields from one or more tables present in the database.

→ A view can either have specific rows based on certain condition or all the rows of table

→ views help to reduce the complexity

→ It increases the security by excluding the sensitive info from the view.

1> Creating view

→ A view can be created using the CREATE VIEW statement

→ we can create a view from a single table or multiple tables.

Syntax:-
```
CREATE VIEW view_name AS
   SELECT col1, col2. . . .
    FROM table_name
    WHERE condition;
```

Creating view from multiple tables

→ View from multiple tables can be created by simply include multiple tables in select statements

Eg.
```
CREATING VIEW Marksview AS
SELECT std1.name, std1.Add, std2.mark
 FROM std1, std2
 WHERE std1.name = std2.mark
```

2) Deleting view

→ A view can be deleted using the Drop view statement

Syntax:-
```
DROP VIEW view_name;
```

3) Updating view

→ There are certain conditions needed to be satisfied to update a view

1> The select statement which is used to create a view should not include GROUP BY & ORDER BY clause.

2> select statement should not have DISTINCT keyword

3> The view should have all NOT NULL values.

Syntax:-
```
CREATE OR REPLACE VIEW viewname AS
  SELECT col1, col2, ..
  FROM tablename
  WHERE cond^n;
```

★ Clauses

1> GROUP BY clause

2) HAVING clause

3) order by clause

1) GROUP BY

→ This clause is used to arrange identical data into groups.

→ The GROUP BY statement is used with the SQL SELECT statement

→ This statement is used with aggregation function

Syntax:- SELECT Column
FROM table-name
WHERE conditions
GROUP BY column
~~ORDER BY column~~

2> HAVING Clause

→ HAVING Clause is used to specify a search condition for a group or an aggregate

→ Having is used in a GROUP BY clause. If you are not using GROUP BY Clause then you can use HAVING function like a WHERE Clause.

Syntax:-
```
SELECT Col1, Col2
FROM table-name
WHERE Conditions
GROUP BY Col1, Col2
HAVING Condition;
```

eg.
```
SELECT company, COUNT(*)
FROM product_mast
GROUP BY company
HAVING COUNT(*)>2;
```

3> ORDER BY

→ The ORDER BY clause sorts the result-set in ascending or descending order.

→ It sorts the records in ascending order by default

→ DESC keyword is used to sort the records in descending order.

Syntax:-
```
SELECT Col1, Col2 . . .
FROM table-name
WHERE Condition
ORDER BY Col1, Col2 . . . ASC|DESC;
```