

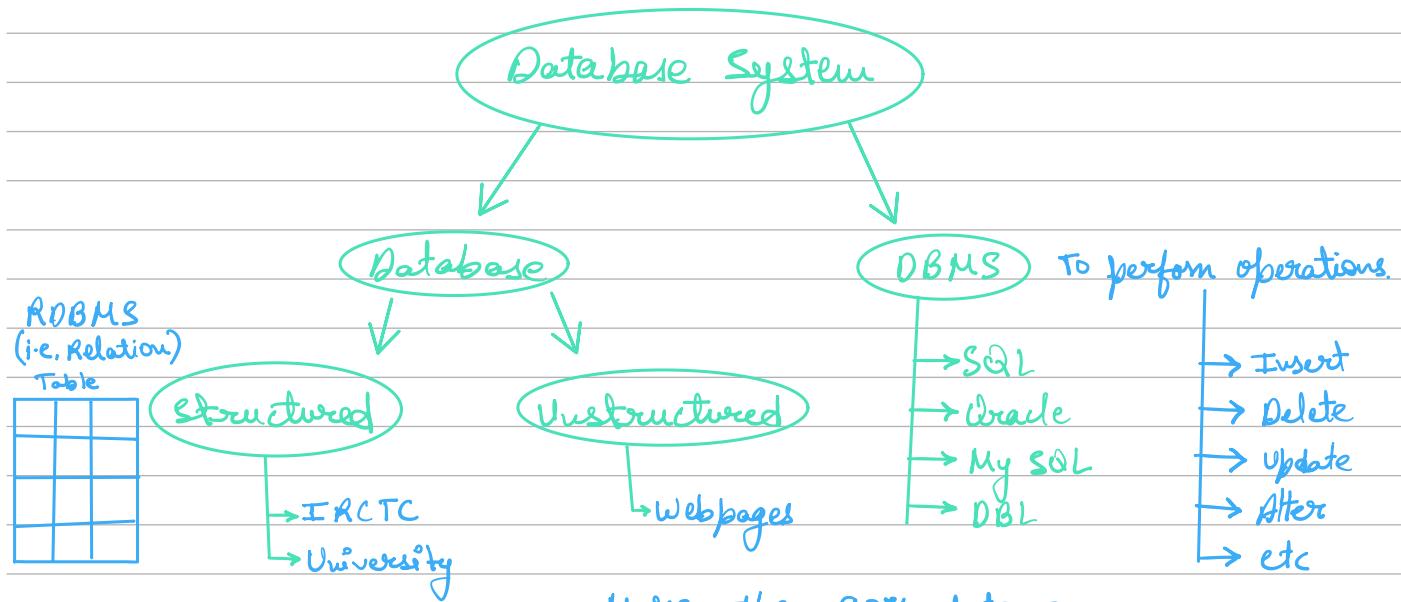
DBMS

- By Arijitesh

Unit - I

* Database System

- Database is a collection of related data.



More than 90% data on this earth is unstructured.
(Big data, ADOBE).

* File System Vs DBMS

- When it came to make the availability of the data to more than one user at a time, then file system failed. Because:-

① Searching data in file system is time consuming & also requires a larger memory.

For example:- A user searched for a small information (size of 1KB) but the file system will provide with a huge data (size of 1GB).

And it will be headache of the user only to look for the particular data that he/she searched for.

But in DBMS, data received is accurate & the actual requested one.

② In file system, we need the meta data (data about data). Like location, etc. to access it.

Attributes But in case of DBMS, we just need to write the query & we will get the desired data, even though we don't know the location path of data.

③ When a large no. of user access the file system at the same time, definitely there is gonna be a inconsistency in the data & file system will fail to handle huge no. of user at the same time.

But the DBMS will provide a proper protocol

RR	✓
RW	X
WR	X
WRX	X

④ File system does not provide any Role Based Access Control But DBMS provides so.

In DBMS, we can restrict users to access certain data and only authorised or selected users can only access it.

⑤ In file system, we can save same file only by changing its name (i.e. Data Duplication).

But in DBMS, there are some strict constraints that does not allow this to happen
And this helps in data integrity.

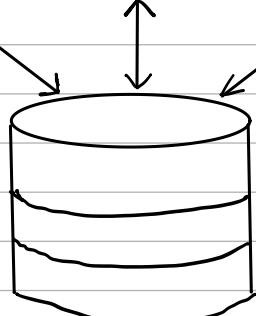
→ Primary Key
→ Foreign Key

Redundancy

★ 2-Tier & 3-Tier Architecture.

Client Layer

Client 1 Client 2 Client 3



These are the interface through user interact with the database.

Data Layer

Queries from Client side will be processed in the data layer.

Database
Server

Advantage

- Easy maintenance

Disadvantage

- Security: Because Client is directly interacting with database.
& they can create problem.
- Scalability: It is difficult for the client layer to handle multiple users

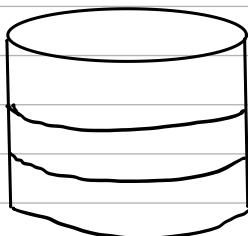
- Due to the problem of security & scalability, we have shifted to 3-Tier Architecture.
- In 2-Tier Architecture, client requests are executed at the database, which increases the load on the database server.
but in case of 3-Tier Architecture, these are executed at Application Server present in Business layer.

Client layer



Business layer

Application Server



Database
Server

Data layer

This is the interface where all the user interact with the Client Application that process all the queries with the support of Application Server present in Business layer

Instead of writing any program, we just provide the interface with some inputs. And it generates & passes that program to Application Server which processes it.

Due the query is converted into low level language / machine language, only then it is passed to the database & this reduces the load on Database server.

Unit - 2

emp

E-id	Name	Dept	Salary
1	Avitesh	HR	10000
2	Darshit	MRKT	20000
3	Abhinav	HR	30000
4	Gautam	MRKT	40000
5	Varun	IT	50000

Create table emp (

```

E-id INT Primary key,
Name Varchar(20),
Dept Varchar(10),
Salary int
);

```

DQL Data Query Language.

(P) Select.

Select * from emp;

>>>

E-id	Name	Dept	Salary
1	Avitesh	HR	10000
2	Darshit	MRKT	20000
3	Abhinav	HR	30000
4	Gautam	MRKT	40000
5	Varun	IT	50000

←

] Tuple

Select name, salary from emp;

>>>

Name	Salary
Avitesh	10000
—	—
—	—
—	—

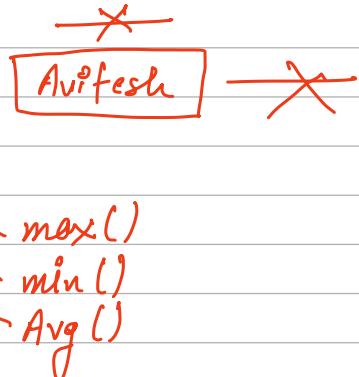
Select Name from emp
where salary > 20000;

>>>

Name	Salary
—	30000
—	40000
—	50000

emp

E-id	Name	Dept	Salary
1	Avitesh	HR	10000
2	Darshit	MRKT	20000
3	Abhinav	HR	30000
4	Gautam	MRKT	40000
5	Varun	IT	50000



① maximum salary:-

Select max(salary) from emp;

>>> 50000

② Name of person with maximum salary.

Select name from emp where salary = (Select max(salary) from emp);

>>> Varun.

③ Select Second highest salary

$x = \text{select max(salary) from emp where salary} <> \text{select max(salary) from emp};$

>>> 40000

④ Name of person with second highest salary

Select name from emp where salary \rightarrow not equals to

= (Select max(salary) from emp where salary $<>$ select max(salary) from emp);

(5) No. of employee in each department.

emp

Count(*)

E-id	Name	Dept	Salary
1	Avitesh	HR	10000
2	Darshit	MRKT	20000
3	Abhinav	HR	30000
4	Gautam	MRKT	40000
5	Varun	IT	50000

Select Count(*) from emp;

<<> 5

HR	Abhinav	30000
MRKT	Gautam	40000
IT	Varun	50000

Select dept, count(*) from emp groupby dept;

HR	2
MRKT	2
IT	1

→ HR	$1+1=2$
→ MRKT	$1+1=2$
IT	1

(6) All dept name having no. of employees less than 2.

Select dept from emp groupby dept having count(*) < 2;

Explanation!

Select dept from emp;

>>> HR
MRKT
HR
MRKT
IT

Select dept from emp groupby dept;

>>> HR
MRKT
IT

Select dept, count(*) from emp groupby dept;

HR	2
MRKT	2
IT	1

Select dept, count(*) from emp groupby dept having count(*) < 2;

IT	1
----	---

Select dept from emp groupby dept having count(*) 22;

>>> IT

② Name of emp having highest salary dept wise

Select max(salary) from emp;

>>> 50000

Select ~~of~~ max(salary) from emp groupby dept;

HR	30000
Mkt	40000
IT	50000

30000
40000
50000

Select ^{dept} name from emp where salary in

(Select max(salary) from emp groupby dept);

30000

no of values.

sum in salary + 30000

→ 102000

HR	Ashish
Mkt	Gautam
IT	

Student (RollNo, Name, Father_Name, Branch)

Book (ISBN, Title, Author, Publisher)

Issue (RollNo, ISBN, Date-of-Issue)

Eid	Name	Dept	Salary
-----	------	------	--------

{ Insert into emp (Eid → Name, dept, salary)
values (1, 'Avitesh', 'HR', 50000), (2, 'Dhruv', 'SERT', 100) }

emp

Eid	Name	Dept	Salary
1	Avitesh	HR	50000
2	Dhruv	SERT	100

Select Salary, Dept, from emp where Name = "Dhruv";
Salary: 100 Dept: MRKT

Select * from emp where Name = "Avitesh"

Roll no 4±;

emp

E-id	Name	Dept	Salary
1	Avitesh	HR	10000
2	Darshit	MRKT	20000
3	Abhinav	HR	30000
4	Gautam	MRKT	40000
5	Varun	IT	50000

Select [Name] from <TABLE NAME>]

emp;

Select Name from emp where salary > 30000;

Select Name from emp where dept = "HR";

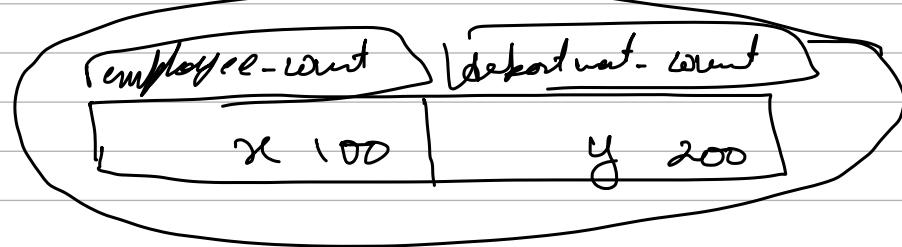
not equals
to

~~Select Name, Salary from emp where dept <> 'IT';~~

employee (emp_id, emp_name, dept_id, sal)

department (dept_id, dept_name, designation)

①



Count(*) ✓
min()
max()
Avg()

Select (Select count(*) from employee) as employee-count,
(Select count(*) from department) as department-count;

② select Avg(salary) as Average-Salary from employee.

Average Salary
5000

③ Select max(salary) as max-sal from employee;

④ Select max(sal) as max-sal, min(sal) as min-sal

from employee where dept_id =

(Select dept_id from department where dept_name = 'HR')

