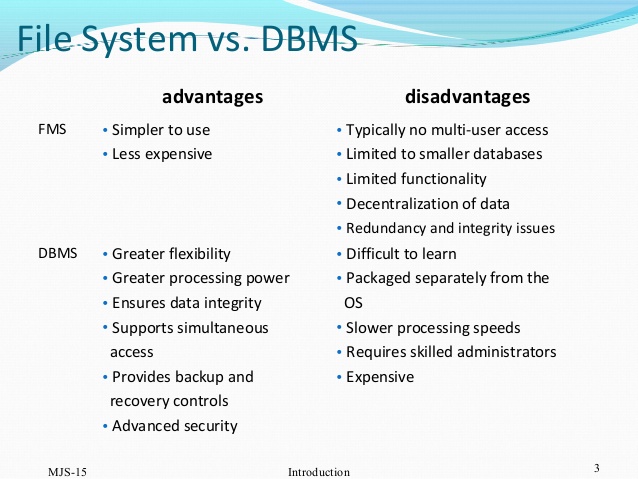
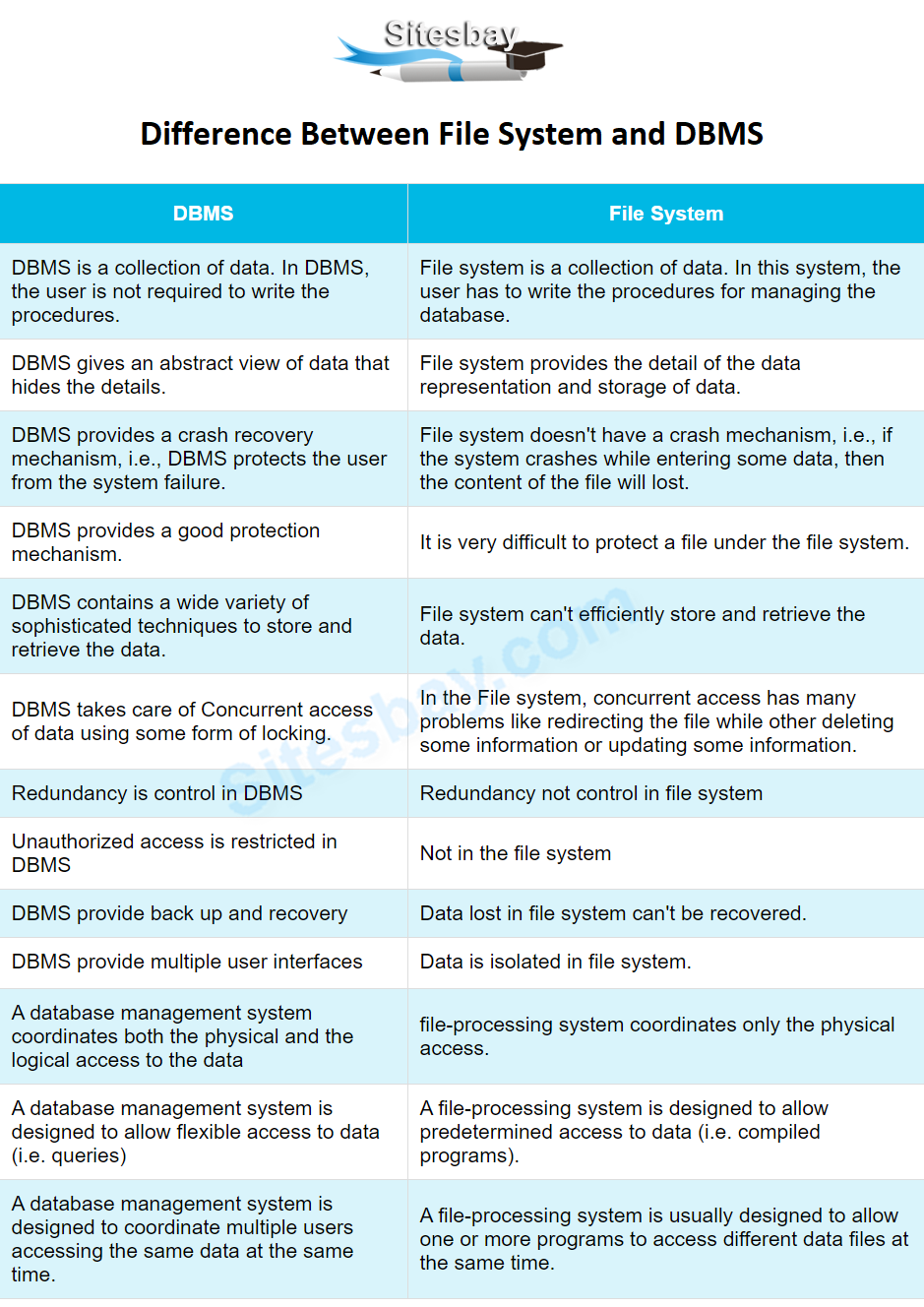
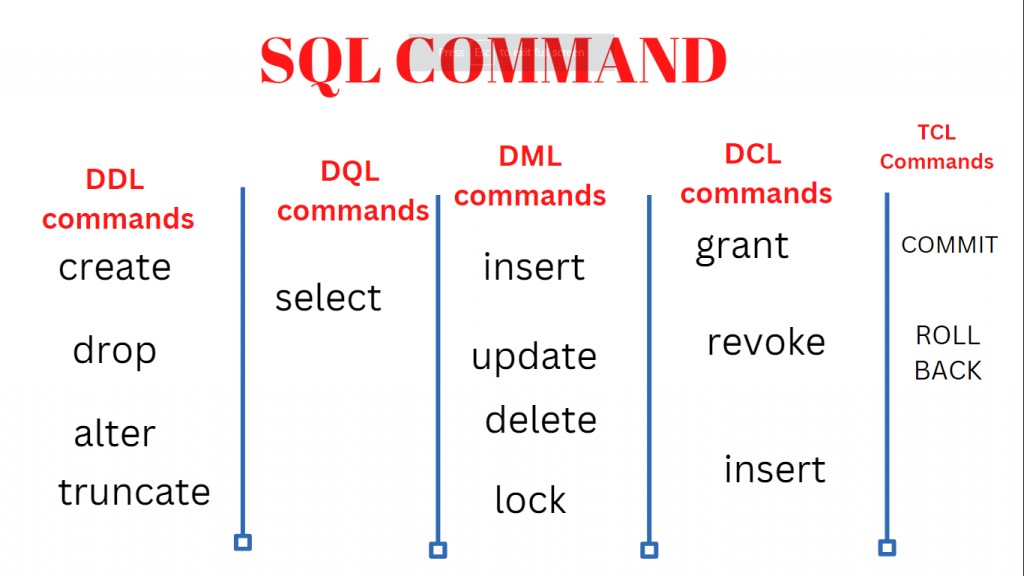
DAY 1

File System VS DBMS:







**Where, having, limit, aggregate functions, distinct, sorting,**

Day 1: sql

Plsql

Agile devops

Genai

Java basic

Python 2 days

Testing

Cicd git

Day 10 dev nd beh abt testing

Web services

12 docker bd Kubernetes

13 cloud

14,15 selenium 16 proj ppt

JOINS:

1.Inner Join

2.Left join

3.Right join

4.FULL outer join

5.cross join

6.self-join

Views

Indexing

**Without index:**

Full table scan

Select \* from employees where name=’sarah’

Data base checks every row until if finds sarah

**With index**

Creating an index

Create index idx\_name on employees(name);

Now searching becomes easier as it directly jumps to the names column

Normalization

1NF:

Removing multiple values

2NF:

Removing partial dependency(composite key)

3NF:

No transitive dependency(non key column dependending on other non key column)

BCNF:

create table orders(order\_id int,customer\_id int,city varchar(20),status varchar(20),order\_date date,amount decimal);

insert into orders(order\_id,customer\_id,city,status,order\_date,amount)

values(1,101,'Pune','Paid','2025-08-01',1200.00),

(2,102,'Mumbai','Paid','2025-08-01',1500.00),

(3,103,'Delhi','Cancelled','2025-08-01',1800.00),

(4,104,'Pune','Paid','2025-08-02',2000.00),

(5,105,'Mumbai','Paid','2025-08-02',500.00),

(1,101,'Pune','Refunded','2025-08-03',1200.00),

(6,105,null,'paid',null,null);

select \* from orders;

select \* from orders where order\_id=3;

select distinct city from orders;

select order\_id,city,amount from orders order by amount desc;

select sum(amount) as total\_amount from orders ;

select avg(amount) as avg\_order\_value from orders;

select city,sum(amount) as revenue from orders where status='paid' group by city;

select city,sum(amount) as revenue from orders where status='paid' group by city having sum(amount)>2500;

select order\_id,city,amount from orders order by amount desc limit 3;

select \* from orders where amount>1000;

select \* from orders where city='pune' or status='paid';

select \* from orders where amount between 1000 and 1500;

select \* from orders where city in ('Mumbai','Delhi');

select \* from orders where city like 'M%';

create table customers(id int,name varchar(20),mail varchar(25));

insert into customers(id,name,mail) values (101,'john','john@gmail.com'),

(102,'raj','raj@gmail.com'),

(103,'Jay','jay@gmail.com'),

(104,'Nikhil','nikhil@gmail.com'),

(105,'Nithin','nithin@gmail.com');

select \* from customers;

select o.order\_id,c.name,o.city,o.amount

from orders o

inner join customers c

on

o.customer\_id=c.id;

select o.order\_id,c.name,o.city,o.amount

from orders o

left join customers c

on

o.customer\_id=c.id;

select o.order\_id,c.name,o.city,o.amount

from orders o

right join customers c

on

o.customer\_id=c.id;

select o.order\_id,c.name,o.city,o.amount

from orders o

left join customers c

on

o.customer\_id=c.id

union

select o.order\_id,c.name,o.city,o.amount

from orders o

right join customers c

on

o.customer\_id=c.id;

select c.name,o.order\_id

from customers c

cross join orders o;

select date(order\_date) as order\_day,

count(\*) as orders,

sum(amount) as revenue

from orders

where status='paid'

group by date(order\_date)

order by order\_day;

select customer\_id,count(\*) as paid\_orders

from orders

where status='paid'

group by customer\_id

having count(\*)>=1

order by paid\_orders DESC;

select \* from orders;

select city,count(distinct customer\_id) as unique\_buyers

from orders where

status='paid'

group by city

order by unique\_buyers desc;

select coalesce(city,'Unknown') as city\_label,sum(amount) as revenue

from orders

where status='paid'

group by coalesce(city,'Unknown');

create table employees(emp\_id int,name varchar(20),dept\_id int,salary int);

insert into employees(emp\_id,name,dept\_id,salary) values

(1,'john',101,60000),

(2,'Alice',102,80000),

(3,'bob',101,50000),

(4,'Sarah',103,90000),

(5,'David',102,70000);

select \* from employees;

create table departments(dept\_id int,dept\_name varchar(20));

insert into departments(dept\_id,dept\_name)

values(101,'it'),

(102,'hr'),

(103,'finance');

select \* from employees where salary>(select avg(salary) from employees);

select name from employees where

dept\_id in(select dept\_id from departments where dept\_name in('it','finance'));

select dept\_id,max(salary) as highest\_salary

from employees

group by dept\_id;

select e.name,e.salary,d.dept\_name

from employees e

join(select dept\_id,max(salary) as highest\_salary from

employees

group by dept\_id) temp

on e.dept\_id=temp.dept\_id and e.salary=temp.highest\_salary

join departments d on e.dept\_id=d.dept\_id;

create view emp\_salaries

as select name,salary from employees;

select \* from emp\_salaries;

create view employee\_dept as

select e.name,e.salary,d.dept\_name

from employees e

join departments d on e.dept\_id=d.dept\_id;

select \* from employee\_dept;

show full tables where table\_type='view';

show create view employee\_dept;