

```
create database my1;
use my1;

create table employee(
eid int not null,
e_name varchar(20),
e_salary int,
e_age int,
e_gender varchar(20),
e_dept varchar(20),
primary key(eid)
);

insert into employee values(
1, 'Sam', 95000, 45, 'Male', 'Operation'
);
insert into employee values(
2, 'bob', 80000, 21, 'Male', 'Support'
);
insert into employee values(
3, 'kalyan', 45000, 23, 'Male', 'Data scientist'
);

select * from employee;
select e_age, e_gender from employee;

select distinct e_gender from employee;

select e_name from employee where e_gender='Male';

select e_name from employee where e_salary>45000 and e_age>22 and
e_dept='Operation';

select * from employee where e_salary>45000 and e_age>22 and e_dept='Operation';

select * from employee where e_dept='support' or e_dept='Operation';

select * from employee where not e_dept='Operation';

select * from employee where e_name like 'k%';
select * from employee where e_age like '2_';

select * from employee where e_age between 20 and 30;

select * from employee

create table dept(
did int not null,
```

```
d_name varchar(20),
e_location varchar(20),
primary key(did)
);
insert into dept values(
1, 'DS', 'Solapur'
);
insert into dept values(
2, 'Analytics', 'Pune'
);
insert into dept values(
3, 'Big data', 'Mumbai'
);
```

```
select * from dept;
```

```
select employee.e_name, employee.e_dept, dept.d_name, dept.e_location
from employee
left join dept
on employee.e_dept=dept.d_name
```

```
select employee.e_name, employee.e_dept, dept.d_name, dept.e_location
from employee
inner join dept
on employee.e_dept=dept.d_name
```

```
select employee.e_name, employee.e_dept, dept.d_name, dept.e_location
from employee
right join dept
on employee.e_dept=dept.d_name
```

```
select employee.e_name, employee.e_dept, dept.d_name, dept.e_location
from employee
full join dept
on employee.e_dept=dept.d_name
```

```
delete from employee where e_name='sam';
select * from employee
```

```
truncate table employee
```

```
update employee set e_name='kkb' where eid=2;
```

```
update employee set e_age=27 from employee
join dept on employee.e_dept=dept.d_name
where e_location='Pune';
select * from dept
```

```
delete employee from employee
join dept on employee.e_dept=dept.d_name
where e_location='Solapur'

select * from employee

merge employee_target as t
using employee_source as s
    on t.eid=s.eid
when matched
    then update set t.e_salary=s.e_salary, t.e_age=s.e_age
when not matched by target
    then insert (eid, e_name, e_salary, e_age, e_gender, e_dept)
    values(s.eid , s.e_name, s.e_salary, s.e_age, s.e_gender, s.e_dept)
when not matched by source
    then delete;

select * from employee_target

alter table employee
add e_dob date;

alter table employee
drop column e_dob;

select * from employee

create table #student(
s_id int, s_name varchar(20));

select * from #student
insert into #student values(1, 'sam')

# functions
select min(e_salary) from employee
select max(e_age) from employee
select count(*) from employee where e_gender='male'
select sum(e_salary) from employee
select avg(e_salary) from employee

select '          KALYAN budharam'
select ltrim('          kalyan')
select lower('          KALYAN budharam')
select ltrim(lower('          KALYAN budharam'))
select ltrim(upper('          KALYAN budharam'))
select ltrim(lower(reverse('          KALYAN budharam'))))
select SUBSTRING('kalyan budharam',8,8)
```

#case statements

```
select case
when 10>20 then '10 is greater'
when 10<20 then '10 is less than 20'
else '10 is =20'
end
```

```
select * from employee
select *, grade=
case
when e_salary>80000 then 'A'
when e_salary>60000 then 'B'
else 'C'
end
from employee
go
```

#if functions

```
select
IIF (10>20, '10 is greater', '20 is greater')
```

```
select eid, e_name, e_age, iif(e_age>30, 'old employee', 'young employee') as
employee_generation from employee
```

users def functions

```
create function add_five(@num as int)
returns int
as
begin
return(@num+5)
end
```

```
select dbo.add_five(100)
```

```
create function select_gender(@gender as varchar(20))
returns table as return
(select * from employee where e_gender=@gender)
```

```
select * from dbo.select_gender('male')
```

orderby clause

```
select * from employee order by e_salary;
select * from employee order by e_salary desc;
```

#top clause

```
select top 2* from employee;
```

```
select top 2* from employee order by e_salary desc;
```

```
#groupby
```

```
select AVG(e_salary), e_gender from employee group by e_gender;
```

```
select AVG(e_age), e_dept from employee group by e_dept order by AVG(e_age) desc;
```

```
#having
```

```
select e_dept, avg(e_salary) as avg_salary
```

```
from employee
```

```
group by e_dept
```

```
having AVG(e_salary)>60000;
```

```
select*from employee
```

```
#union must be columns nos/names are exact,/all =multiples also shows
```

```
select * from employee
```

```
union all
```

```
select * from dept;
```

```
#procedures
```

```
create procedure emp_age
```

```
as select e_age from employee
```

```
go
```

```
exec emp_age
```

```
create procedure emp_avg_salary
```

```
as select AVG(e_salary) from employee
```

```
go
```

```
exec emp_avg_salary
```

```
#procedures with parameters
```

```
create procedure emp_gender @gender varchar(20)
```

```
as select * from employee
```

```
where e_gender=@gender
```

```
go
```

```
exec emp_gender @gender='male'
```

```
#views
```

```
create view e_dep as
```

```
select * from employee
```

```
where e_dept='data scientist'
```

```
select * from e_dep
```

