```sql
-- write a query to create a table employeed with column empid as primary
key , name, location as default LPU,
 salary should be greater than 20000
create table employeed(empid number(20) primary key, name varchar(30),
location varchar(20) default 'LPU', salary number(25)
check (salary>20000));
-- write a query to insert values in the table
insert into employeed values(1, 'Ankit', 'Jalandhar', 25000);
insert into employeed values(2, 'Rahul', 'Jalandhar', 75000);
insert into employeed values(3, 'Mayank', 'Phagwara', 55000);
select * from employeed;
-- write a query to select minimum salary from the table employeed
select min(salary) from employeed;
-- write a query to select maximum salary from the table employeed
select max(salary) from employeed;
-- write a query to find average  salary of employees
select avg(salary) as average from employeed;
-- write a query to find total salary paid by company to employees
select sum(salary) as totalsalary from employeed;
-- write a query to find no of tuples in the table
select count(*)as noof_tuples from employeed;
-- write a query to find totalnumber location where employees are
deputed.
select count(distinct location) from employeed;
select count(unique(location)) as Total_loactions from employeed;
select distinct location from employeed;-- show available locations with
name
select location as city, salary as basic from employeed
-- group by- statement groups rows that the same values into summary
rows, to arrange identical data into groups
-- write query  to retrieve sum of salary loctionwise
-- select coloumn1, function_name(column2) from tablename where condition
group by col1
select location, sum(salary) from employeed group by location;
--having: return rows where aggregate values meet the specified
conditions
 -- select coloumn1, function_name(column2) from tablename where
condition group by col1 having condition
-- write a query to show sum of salary for that location where 2 or more
employees are duputed
select location, sum(salary) from employeed group by location having
count(location)>=2;
--logical operators:
-- write a query to retrieve name of employees start with M
select name from employeed where name like 'M%';   Mia ,Mohit, Mi, M
-- write a query to retrieve name of employees end with k
select name from employeed where name like '%k';
-- write a query to retrieve name of employees which containing an
select name from employeed where name like '%an%';
--write a query to find the name of employee who 2nd character is a ,
first character canbe any and after 'a'
there are 4 characters
 select name from employeed where name like '_a____';
- in operator : it compares a value to that specified list value
-- select * from tablename where columnname in(value1,value2,.....);
select * from employeed where salary in (25000,4000);
-- write a query to retrieve salary of employee which are not equal to
25000 or 4000
select * from employeed where salary not in (25000,4000);
```

```sql
-- write a query to retrieve employee details which are not deputed in
'Phagwara'
select * from employeed where location not in('Phagwara');
-- write a query to find 2 last minimum salary from the employee table
select min(salary) from employeed where salary not in (select min(salary)
from employeed);
-- write a query to retrieve the details of those employees whose salary
field is null
select * from employeed where salary is NULL; run this for compare NULL
select * from employeed where salary=NULL;-- always show no data found--
problematic query
select * from employeed where salary in(NULL);always show no data found
problematic query
select * from employeed where salary is not NULL; run this for compare
not  NULL
select * from employeed where salary != NULL; -- always show no data
found problematic query

-- between operator : that select values within a given range . the value
can be number number , date etc

select * from table name where columname between value1 and value2;
-- write a query to find all employee salary between 25000 to 75000
select * from employeed where salary between 25000 and 75000;
```