```
CREATE TABLE Employee(
Id int,
Name varchar(20),
Address varchar(50),
City varchar(50),
Department varchar(20),
Salary int
);

INSERT INTO Employee(Id,Name,Address,City,Department,Salary) VALUES (1,'Sreedev','Attithara kidangara','Alappuzha','Computer',10000);
SELECT * FROM Employee;
INSERT INTO Employee(Id,Name,Address,City,Department,Salary)
VALUES (1,'yadhu','vietnam colony','Alappuzha','Computer',30000);
INSERT INTO Employee(Id,Name,Address,City,Department,Salary)
VALUES (1,'unnikrishnan','Kaavalam','Alappuzha','Computer',5000);

SELECT * FROM Employee where Name='Sreedev';

DROP TABLE IF EXISTS employee;

CREATE TABLE employee( emp_id int primary key, emp_name varchar(20), designation varchar(30), department varchar(20), salary int);
```

```
SELECT * FROM employee;

INSERT INTO employee (emp_id, emp_name, designation, department, salary) VALUES
(101, 'Aarav', 'Software Engineer', 'Development', 25000),
(102, 'Isha', 'Data Analyst', 'Data Science', 22000),
(103, 'Vikram', 'Web Developer', 'Development', 27000),
(104, 'Sita', 'UI/UX Designer', 'Design', 23000),
(105, 'Ravi', 'System Administrator', 'IT Support', 20000),
(106, 'Anaya', 'Project Coordinator', 'Project Management', 24000),
(107, 'Rahul', 'Project Administrator', 'Project Management', 26000),
(108, 'Priya', 'Business Analyst', 'Business Analysis', 25000),
(109, 'Karthik', 'DevOps Engineer', 'Development', 28000),
(110, 'Meera', 'Quality Assurance', 'Project Management', 21000),
(111, 'Sneha', 'Technical Writer', 'Documentation', 19000);

delete from employee where emp_name= 'Priya';

--order by:

select emp_name , salary from employee order by salary asc;

--second highest salary

select emp_name ,salary from employee where
salary = (select max(salary) from employee
```

```
where salary < (select max(salary) from employee));

--using distict keyword

SELECT DISTINCT department FROM employee;

--count and distinct

SELECT COUNT(DISTINCT department) AS dept_details FROM employee;

SELECT COUNT(*) AS dept_details FROM (
   SELECT DISTINCT department FROM employee
) AS unique_dept;
--above alias is used

SELECT emp_id FROM employee AS EMP_ID;

ALTER TABLE employee ADD age int;

select * from employee;

update employee SET age=20 where emp_name='Aarav';

select * from employee where salary>25000;
select * from employee where salary= (select max(salary) from employee);

select avg(salary) as avg_salary from employee;
```

```
select distinct(count(*)) as no_of_records from employee;



create table customer(
cust_id int primary key,
name varchar(15),
city varchar(20)
);

insert into customer(cust_id,name,city)
values(1,'sreedev','alappuzha');
insert into customer values(
2,'yadhu','mariyampally'),
(3,'jinson','pooppally'),
(4,'dano','champakkulam');



select * from customer;

create table products(
pro_id int primary key,
name varchar(20),
place varchar(20),
cust_id int
foreign key(cust_id) references customer(cust_id)
);

insert into products(pro_id,name,place,cust_id)
values(1,'samsung','alappuzha',1);
```

```
insert into products
values(2,'apple','thodupuzha',2),
(3,'watch','kottayam',3),
(4,'redmi','changanacherry',2);
insert into products values(
5,'apple','kidangara',3);

select * from products;

--joins

select customer.name as cust_name,
products.name as product_name from customer
inner join products on customer.cust_id=products.pro_id;

ALTER TABLE customer
ADD phone_number VARCHAR(15);

ALTER TABLE customer
DROP COLUMN phone_number;

EXEC sp_rename 'customer.name', 'cust_name', 'COLUMN';
EXEC sp_rename 'products.name', 'product_name','COLUMN';

select * from customer join products on customer.cust_id=products.pro_id;

select * from customer join products on customer.cust_id=products.pro_id where customer.cust_id=2;
select distinct(customer.city) as city_list,product_name from customer join products on
customer.cust_id=products.pro_id;
```

```
select cust_name,city from customer join products on customer.cust_id=products.cust_id

where products.product_name='apple';

--left join

select * from customer left join products on customer.cust_id=products.cust_id;

--right join

select * from customer right join products on customer.cust_id=products.cust_id;

--full join

select * from customer full join products on customer.cust_id=products.cust_id;--shows the both tables
.dano is included

select * from customer inner join products on customer.cust_id=products.cust_id;--dano dont buy any
columns  so  he is ignored

select * from customer join products on customer.cust_id=products.cust_id;--dano is ignored

select * from employee order by  salary desc;

select count(*) from employee group by  department;

select distinct(department),count(*) as number_of_employees
```

```
from employee group by department order by department asc;


--stored procedures


create procedure emp_records
as
begin
        select * from employee;
end;

exec emp_records;

--parameterized stored procedures

create procedure emp_select
@emp_id int
as
begin
        select * from employee where emp_id=@emp_id;
end;

exec emp_select @emp_id=101;

create procedure emp_count_dept
@id int,
@salary decimal(10,2)
```

```
as
begin
	update employee set salary=@salary where emp_id=@id;
end;

exec emp_count_dept @id=102,@salary=25000.07;

select * from employee;



create procedure emp_dept_count
as
begin
	select distinct(department),count(*) as count_emp
	from employee group by department;
end;

exec emp_dept_count;



--if

create procedure age_emplo
@emp_age int=null

as
begin
	if @emp_age is null
		select * from employee where age=null;
```

```
        else
                select * from employee where age=@emp_age;
end

exec age_emplo;

--operators

select salary,salary*0.10 as bonus from employee;

select * from employee where salary>25000;

select emp_name,salary from employee where salary>18000 and salary<25000
order by salary asc;

select cust_id from customer intersect
select cust_id from products;

select * from products;

select * from customer;

--in opeartor

select * from employee where age in (20,28,10);

update employee set age=23 where emp_id=103;

update employee set age=28 where age=23;
```

```
select * from employee;

--between

select  * from employee where salary between 20000 and 22000;

--null opertor

select * from employee where age is null;

select * from employee where age is not null;

--concat 2 strings

select emp_name+' '+department as empl_dept from employee;

select emp_name+' '+department as empl_dept from employee
where age=20;



--select top

select top 3 * from employee;

select top 50 percent * from employee where salary<20000;
```

```
select top 5 * from employee where salary>20000;


--like

select emp_name from employee where emp_name like 'r%';

select emp_name from employee where emp_name like '%a';


select distinct(department),count(*) as count_emp from employee group by department;


--having


create table Orders1
(order_id int primary key,
cust_id int,
ord_amt int
);

insert into Orders1
values(1,101,500),
(2,102,300),(3,101,200),(4,103,700),(5,102,400);

select * from Orders1;

select cust_id, sum(ord_amt) as total_amount from Orders1
```

```
group by cust_id having sum(ord_amt)<500;

insert into Orders1
values (6,104,100);

drop table Orders1;

--exists

create table Buyers(
cust_id int primary key,
name varchar(20)
);

create table Orders(
ord_id int primary key,
cust_id int foreign key (cust_id) references Buyers(cust_id),
ord_amt  int
);

insert into Buyers values
(1,'julie'),(2,'jacob'),(3,'yadhu');

insert into Orders values
(1,1,500),(2,2,300);

select name from Buyers where exists
(select ord_amt from Orders where Orders.cust_id=Buyers.cust_id);
```

```
select * from employee;

select emp_name,salary from employee where exists
(select emp_name from employee where department='Design');

select emp_name,salary from employee where not exists
(select emp_name from employee where department='hr');



create table managers(
emp_id int primary key,
name varchar(20),
manage_id int
);



---self join

insert into managers values
(1,'susan',null),
(2,'alice',1),
(3,'jacob',2),
(4,'sreedev',1),
(5,'unnikrishnan',2);

select * from managers;

select a.name as Employee,b.name as Manager
from managers a join managers b on a.emp_id=b.manage_id;
```

```
select * from employee;

select * from employee where salary=
(select max(salary) from employee where
salary<(select max(salary) from employee));

select max(salary) from employee;



--stored procedures



--CREATING A TABLE USERS

CREATE TABLE Users(
User_id INT,
Username VARCHAR(25),
Password VARCHAR(25),
Email VARCHAR(25));

--SHOWING THE TABLE STRUCTURE

EXEC sp_help Users;

--CREATED STORED PROCEDURE FOR CREATING USER
```

```
CREATE PROCEDURE CreateNewUser
@Userid INT,
@Username VARCHAR(25),
@Password VARCHAR(25),
@Email VARCHAR(25)
AS
BEGIN
INSERT INTO Users (User_id, Username, Password, Email)
VALUES (@Userid, @Username, @Password, @Email)
END;

EXEC CreateNewUser
@Userid=1,
@Username='sreedev',
@Password='123dss',
@Email='sreedev@gmail.com';

SELECT * FROM Users;

EXEC CreateNewUser
@Userid=101,
@Username = 'Adithya',
@Password = 'aDithya123',
@Email = 'adithya555@gmail.com';

--stored procedure for  deleting an user

CREATE PROCEDURE Del_User
@User_id int
```

```
AS
BEGIN
DELETE FROM Users WHERE User_id=@User_id;
END;

EXEC Del_User @User_id=1;

--dropping a procedure

DROP PROCEDURE IF EXISTS Del_User;



--create a procedure for update

CREATE PROCEDURE UPDATE_USER
@user_name varchar(20),
@user_id int

AS
BEGIN
        UPDATE Users SET Username=@user_name WHERE User_id=@user_id;
END;

EXEC UPDATE_USER @user_name='sreedev',@user_id=101;

select * from  Users;



--if else inside stored procedure
```

```
CREATE TABLE Students (
Student_id INT PRIMARY KEY,
first_name VARCHAR(25),
last_Name VARCHAR(25),
dob DATE,
admission_date DATE);

drop table STUDENTS;

---Create stored procedure for CRUD operations

CREATE PROCEDURE ManageStudent
@Action NVARCHAR(10),
@Studentid INT = NULL,
@FirstName NVARCHAR(50) = NULL,
@LastName NVARCHAR(50) = NULL,
@DOB DATE = NULL,
@AdmissionDate DATE = NULL
AS
BEGIN
IF @Action = 'CREATE'
BEGIN
INSERT INTO Students(Student_id,first_name, last_Name, dob, admission_date)
VALUES (@Studentid, @FirstName, @LastName, @DOB, @AdmissionDate);
END

ELSE IF @Action = 'READ'
BEGIN
```

```
SELECT * FROM Students WHERE Student_id = @Studentid;
END

ELSE IF @Action = 'UPDATE'
BEGIN
UPDATE Students SET
first_name=@FirstName,
last_Name=@LastName,
dob=@DOB,
admission_date=@AdmissionDate
WHERE Student_id=@Studentid;
END

ELSE IF  @Action = 'DELETE'
BEGIN
DELETE FROM Students WHERE Student_id=@Studentid;
END

END;


---executing stored procedures

EXEC ManageStudent
@Action = 'CREATE',
@Studentid = 101 ,
@FirstName = 'Sreedev',
@LastName = 'Dasappan',
@DOB ='2000-1-1',
```

```
@AdmissionDate ='2024-9-1';

EXEC ManageStudent
@Action='READ',
@Studentid=101;


EXEC ManageStudent
@Action = 'UPDATE',
@Studentid = 101 ,
@FirstName = 'Jane';

EXEC ManageStudent
@Action='DELETE',
@Studentid=101;

---indexing


create table Students11(
stud_id int,
name nvarchar,
age int,city nvarchar
);

alter table Students11 alter column name varchar(20);
alter table Students11 alter column city varchar(20);

create clustered index idx_Studentid on Students11(stud_id);
```

```
insert into Students11 values(
101,'sreedev',20,'china'),
(103,'yadhu',34,'africa'),
(102,'janaki',20,'kerala');

select * from Students11;

select * from Students11 where stud_id=103;

EXEC sp_help 'Students11';


---non clustered index

CREATE TABLE reg_form (
first_name VARCHAR(20),
last_name VARCHAR(20),
state VARCHAR(20),
city VARCHAR(20),
dob DATE,
user_name VARCHAR(20),
password VARCHAR(20)
);

INSERT INTO reg_form (first_name, last_name, state, city, dob, user_name, password) VALUES
('Aarav', 'vishnu', 'Kerala', 'Kochi', '2000-01-15', 'aarav_m', 'pass1'),
('Isha', 'Ganesh', 'Kerala', 'Thiruvananthapuram', '1998-05-22', 'isha_n','pass2'),
('Vikram', 'Alex', 'Kerala', 'Kozhikode', '1995-07-30', 'vikram p', 'pass3'),
```

```
('Sita', 'Kurup', 'Kerala', 'Kollam', '1997-11-05', 'sita_k', 'pass4'),
('Ravi', 'Raj', 'Kerala', 'Kochi', '1990-09-14', 'ravi_r', 'pass5'),
('Anaya', 'Shetty', 'Karnataka', 'Bangalore', '1999-02-18', 'anaya_s', 'pass6'),
('Rahul', 'Kumar', 'Karnataka', 'Mysore', '1996-04-20', 'rahul_k', 'pass7'),
('Priya', 'Rao', 'Karnataka', 'Hubli', '1992-06-25', 'priya_r', 'pass8'),
('Karthik', 'Natarajan', 'Tamil Nadu', 'Chennai', '1989-12-12', 'karthik_n', 'pass9'),
('Meera', 'Lijo', 'Tamil Nadu', 'Coimbatore', '1994-08-08', 'meera_i', 'pass10');

select * from reg_form;

update  reg_form set dob='1999-02-01' where first_name='sita';

select * from reg_form where state='Kerala' and city='Kollam';

create clustered index idx_firstname on reg_form(first_name);

select *from reg_form;

create nonclustered index idx_city on reg_form(city);

select * from reg_form;

select * from reg_form order by dob;

select distinct state from reg_form;
select distinct state ,count(*) as number from reg_form group by state;


select * from employee;
```

```
update employee set age=30 where age is null;
```