1. Create a table customer (cust_novarchar(5), cust_namevarchar(15), age number, phone varchar(10))

a) insert 5 records and display it
b) add new field d_birth with date datatype
c) create another table cust_phone with fields cust_name and phone from customer table
d) remove the field age
e) change the size of the cust_name to 25
f) delete all the records from the table
g) rename the table cutomer to cust
h) drop the table

```
Create table customer(cust_no varchar(5), cust_name varchar(15), age int, phone varchar(10) );

a)insert into customer values('c001','Raju',21,'9878675645');

 insert into customer values('c002','Rajina',32,'2356781221');

  insert into customer values('c003','Theertha',22,'8745987456');

   insert into customer values('c004','James',28,'8521456986');

    insert into customer values('c005','Sana',12,'9632589632');

    select * from customer

     b)alter table customer add column d_birth date;

     c)create table cust_phone as select cust_name, phone from customer;

     d)alter table customer drop column age;

     e)alter table customer alter column cust_name type varchar(25)

     f)delete from customer;

     g)alter table customer rename  to cust

     h)drop table cust;
```

2. Create a table sales_man (salesman_no primary key, s_name not null, place, phone unique)
Create table sales_order(order_no primary key
order_date not null
salesman_no foreign key references salesman_no in sales_man
del_type values should be either P or F (check constraints)
order_status values should be 'Inprocess','Fullfilled','Backorder', 'Cancelled' (check constraints))

a) Insert few records in both tables
b) Delete primary key from sales_man table
c) Delete Foreign key and Check constraints from sales_order table
d) Add primary key in sales_man using ALTER TABLE
e) Add foreign key and CHECK constraints in sales_order table using ALTER TABLE

```
create table salesman(salesman_no varchar(5)primary key,s_name varchar(10) not null,place
varchar(15),phone varchar(15) unique);

create table sales_order(order_no varchar(5) primary key,order_date date not null,salesman_no
varchar(5) constraint sales_order_salesman_no_fkey  references salesman(salesman_no),del_type char
constraint sales_order_del_type_check check(del_type in ('p','f')),order_status varchar(15) constraint
sales_order_order_status_check check(order_status in ('inprocess','fulfilled','backorder','cancelled')));
```

```
 a)   insert into salesman values('S01','Rijas','Mumbai','9784563214');
insert into salesman values('S08','Pranoy','Pune','6589632145');
insert into sales_order values('OD01','12-02-2019','S01','p','inprocess');
 insert into sales_order values('OD02','25-03-2019','S08','f','cancelled');

 b)   alter table sales_order drop constraint sales_order_salesman_no_fkey
alter table sales_order drop constraint sales_order_del_type_check
 alter table sales_order drop constraint sales_order_order_status_check

c)   alter table salesman drop constraint salesman_pkey
d)   alter table salesman add primary key (salesman_no)
e)  alter table sales_order add foreign key (salesman_no) references salesman(salesman_no);
 alter table sales_order add constraint del_type check(del_type in('p','f'));
 alter table sales_order add constraint order_status check(order_status in
('inprocess','fulfilled','backorder','cancelled'));
```

3. Create a table Hospital with the fields (doctorid,doctorname,department,qualification,experience). Write the queries to perform the following.
    a) Insert 5 records
    b) Display the details of Doctors
    c) Display the details of doctors who have the qualification ‘MD’
    d) Display all doctors who have more than 5 years experience but do not have the qualification ‘MD’
    e) Display the doctors in ‘Skin’ department
    f) update the experience of doctor with doctored=’D003’ to 5
    g) Delete the doctor with DoctorID=’D005’

```
Create table Hospital (doctorid varchar(5),doctorname varchar(10),department varchar(10),qualification
varchar(10),experience integer);

 a)insert into hospital values('D001','Gopal','Dental','MBBS',3);

 insert into hospital values('D002','Jaya','Cardio','MD',3.5);

 insert into hospital values('D003','Nithya','skin','MBBS',7);

insert into hospital values('D004','Alex','ENT','MD',4);
```

```
Insert into hospital values('D005','Sunaina','General','MBBS',2);
```

```
b)select * from hospital;
```

```
c)select doctorname,qualification from hospital where qualification='MD';
```

```
d)select doctorname,qualification from hospital where experience>5 and qualification!='MD';
```

```
e)select doctorname,department from hospital where department='skin';
```

```
f)update hospital set experience=5 where doctorid='D005';
```

```
g)delete from hospital where doctorid='D005';
```

4. Create the following tables

    Bank_customer (accno primary key, cust_name,place)
    Deposit (accno foreignkey, deposit_no, damount)
    Loan (accno foreign key loan_no, Lamount)

   Write the following queries

   a) Display the details of the customers
   b) Display the customers along with deposit amount who have only deposit with the bank
   c) Display the customers along with loan amount who have only loan with the bank
   d) Display the customers they have both loan and deposit with the bank
   e) Display the customer who have neither a loan nor a deposit with the bank

```
create table bank_customer( accno integer  primary key,cust_name varchar(10),place varchar(20));
insert into bank_customer values(101,'anu','vtk');
insert into bank_customer values(102,'anup','clt');
insert into bank_customer values(103,'appu','tvm');
insert into bank_customer values(104,'anupama','vtk');
insert into bank_customer values(105,'anusree','clt');
create table deposit(accno integer  references bank_customer(accno),depositno integer primary key,depositamt integer);
insert into deposit values(101,201,10000);
insert into deposit values(102,202,50000);
insert into deposit values(103,203,10000);
select * from deposit;
create table loan(accno integer  references bank_customer(accno),loanno integer primary key,loanamt integer);
insert into loan values(101,301,1000);
insert into loan values(102,302,5000);
insert into loan values(104,303,10000);
a) select * from bank_customer;
```

```
b)  select distinct bank_customer.accno,bank_customer.cust_name,deposit.depositamt from
bank_customer,deposit,loan where bank_customer.accno=deposit.accno and deposit.accno not
in(select accno from loan);
c)  select distinct bank_customer.accno,bank_customer.cust_name,loan.loanamt from
bank_customer,deposit,loan where bank_customer.accno=loan.accno and loan.accno not in(select
accno from deposit);
e) select bank_customer.accno,bank_customer.cust_name from bank_customer where accno
IN(select loan.accno from loan,deposit where loan.accno=deposit.accno);
f)  select bank_customer.accno,bank_customer.cust_name from bank_customer where accno not
IN(select  loan.accno from loan) AND accno not IN(select  deposit.accno from deposit);
```

5. Create a table employee with fields (EmpID, EName, Salary, Department, and Age). Insert some records. Write SQL queries using aggregate functions and group by clause
    A. Display the total number of employees.
    B. Display the name and age of the oldest employee of each department.
    C. Display the average age of employees of each department
    D. Display departments and the average salaries
    E. Display the lowest salary in employee table
    F. Display the number of employees working in purchase department
    G. Display the highest salary in sales department;
    H. Display the difference between highest and lowest salary

```
Create table employee(EmpID varchar(5) primary key, EName varchar(10), Salary integer, Department
varchar(15),Age int);
insert into employee values('E001','Fasal',50000,'HR',33);
 insert into employee values('E002','Julie',12000,'purchase',25);
insert into employee values('E003','Thanu',33450,'Administrative',29);
 insert into employee values('E004','Jez',12345,'Office',28);
insert into employee values('E005','Aryan',51342,'Sales',40);
insert into employee values('E006','Meera',48000,'HR',40);
select * from employee
a) select count(*) from employee
b) select ename,age from employee where age in (select max(age) from employee group by department)
c) select department,avg(age) from employee group by department;
d) select department, avg(salary) from employee group by department
 e) select min(salary) from employee
f) select count(*) from employee where department ='purchase'
g) select max(salary) from employee where department='Sales'
h) select max(salary)-min(salary) from employee
```

6. Create a table product with the fields (Product_code primary key, Product_Name, Category, Quantity, Price).

   Insert some records Write the queries to perform the following.
    a. Display the records in the descending order of Product_Name

b. Display Product_Code, Product_Name with price between 20 and 50

c. Display the details of products which belongs to the categories of ‘bath soap’, ‘paste’, or ‘washing powder’

d. Display the products whose Quantity less than 100 or greater than 500

e. Display the products whose names starts with 's'

f. Display the products which not belongs to the category 'paste'

g. Display the products whose second letter is 'u' and belongs to the Category 'washing powder'

```
Create table product(Product_code varchar(4) primary key, Product_Name varchar(15),
Category varchar(15), Quantity int, Price numeric(6,2))
insert into product values('p001','sunlight','washing powder',300,50.75);
insert into product values('p002','surf excel','washing powder',50,48.50);
insert into product values('p003','colgate','paste',120,20);
insert into product values('p004','lux','bath soap',25,50);
insert into product values('p005','enchanteur','perfume',530,120);
select * from product
a) select * from product order by Product_Name desc
 b) select Product_Code, Product_Name from product where price between 20 and 50
c) select * from product where category in('bath soap','paste','washing powder')
d) select * from product where quantity<100 or quantity >500
e) select * from product where Product_Name like 's%';
f) select * from product where category!='paste'
g) select * from product where Product_Name like '_u%' and category='washing powder'
```

7. Consider the employee database given below. Give an expression in SQL for each of the following queries:

   EMPLOYEE (Employee-Name, City)
   WORKS (Employee-Name, Company-Name, Salary)
   COMPANY (Company-Name, City)
   MANAGES (Employee-Name, Manager-Name)

   A) Find the names of all employees who work in Infosys
   B) Find the names and cities of residence of all employees who works in Wipro
   C) Find the names, and cities of all employees who work in Infosys and earn more than Rs. 10,000.
   D) Find the employees who live in the same cities as the companies for which they work.
   E) Find all employees who do not work in Wipro Corporation.
   F) Find the company that has the most employees.

```
create table employ(ename varchar(20) primary key,city varchar(20));
insert into employ values('anu','calicut');
insert into employ values('anup','calicut');
insert into employ values('anusree','tvm');
insert into employ values('anupriya','kochi');
insert into employ values('anoop','trissur');
insert into employ values('anuvinda','kochi');
insert into employ values('appu','kochi');
select * from employ;

create table cmpny(cname varchar(20) primary key,city varchar(20));
insert into cmpny values('infosis','calicut');
insert into cmpny values('IBM','kochi');
insert into cmpny values('wipro','tvm');
select * from cmpny;

create table manages(ename varchar(20) references employ(ename),mname varchar(20));
insert into manages values('anu','anu');
insert into manages values('anup','anu');
insert into manages values('anusree','anoop');
insert into manages values('anupriya','anoop');
insert into manages values('anoop','anoop');
insert into manages values('anuvinda','anoop');

create table works(ename varchar(20) references employ(ename),cname varchar(20) references
cmpny(cname),salary int);
insert into works values('anu','infosis',25000);
insert into works values('anup','infosis',9000);
insert into works values('anusree','wipro',30000);
insert into works values('anupriya','wipro',10000);
insert into works values('anoop','wipro',22000);
insert into works values('anuvinda','wipro',25000);
insert into works values('appu','IBM',25000);
select * from works;

A) select ename from works where cname='infosis';
B)select employ.ename,employ.city from employ,works where employ.ename=works.ename and
works.cname='wipro';
C)select employ.ename,employ.city from employ,works where employ.ename=works.ename and
works.cname='infosis' and works.salary>10000;
D)select employ.ename from employ,cmpny,works where employ.city=cmpny.city and
employ.ename=works.ename and works.cname=cmpny.cname;
E)select ename from employ where ename not in( select ename from works where cname='wipro');
F)create view emplcount as select cname,count(cname)as count from works group by cname;
  select cname from emplcount where count in(select max(count) from emplcount);
```

8. Write a program code to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding value of calculated area in an empty table named areas with field's radius and area.

```
create table areas(radius int,area numeric);


  create or replace function radius() returns void as $$
```

```
declare
r int;
a numeric;
begin
for r in 3..7
loop
a:=3.14*r*r;
insert into areas  values(r,a);
end loop;
end $$ language plpgsql;

select radius();


select * from areas;
```

9. Write a program block to calculate the electricity bill by accepting cust_no and units_consumed.

```
   create table bill(cno int primary key,cname varchar(10),uc int,bamt numeric);
insert into bill values('1','smitha',250,null);
insert into bill values('2','smith',50,null);
insert into bill values('3','mitha',150,null);
insert into bill values('4','smitha',100,null);
insert into bill values('5','tha',250,null);
select * from bill;
create or replace function billcalc() returns void as $$
declare
BA numeric;
custno int;
```

```
unit int;

cur3 cursor for select cno,uc from bill;

begin

open cur3;

loop

fetch cur3 into custno,unit;

exit when not found;

BA:= unit*3;

update bill set bamt=BA where cno=custno;

end loop;

close cur3;

end $$ language plpgsql;


select billcalc();


select * from bill;
```

10. Create a procedure toprint Fibonacci number up to a limit, limit is passed as an argument

```
  Create or replace function fib(N int)returns void as $$
declare
t1 int:=0;
t2 int:=1;
i int;
t3 int:=0;
begin
raise notice '%',t1;
raise notice '%',t2;
for i in 2..N loop
t3:=t1+t2;
t1:=t2;
t2:=t3;
```

```
    raise notice '%',t3;
    end loop;
    end $$ language plpgsql;

    select fib(10);
```

11. Create a function to check whether a given number is prime or not

```
   create or replace function prime(n int)

returns void as $$

declare

i int ;

f int:=0;

begin

If n=2 then

f:=0;

else

for i in 2..(n-1)

loop

if (mod(n,i)=0)

then

f:=1;

exit;

end if;

end loop;

end if;

if(f=0)

then

raise notice'prime number';

else

raise notice 'not prime';
```

```
end if;

end $$ language plpgsql;


select prime (5);
```

12. create a table emp_salary(empno,ename,dept,salary)

Write a function to return the average salary of a particular department by accepting departmentname as argument.

```
   create table empsal(empno int primary key,ename varchar(10),dept varchar(10),salary int);

insert into empsal values(1,'anu','sales',10000);
insert into empsal values(2,'anu','HR',20000);
insert into empsal values(3,'manu','sales',30000);
insert into empsal values(4,'sanu','IT',40000);
insert into empsal values(5,'nanu','sales',30000);
insert into empsal values(6,'priya','HR',25000);
insert into empsal values(7,'vinu','IT',50000);

select * from empsal;

create or replace function salcalc(d varchar) returns numeric as $$
declare
av numeric;
cur cursor for select round(avg(salary)) from empsal where dept=d;
begin
open cur;
fetch cur into av;
return av;
close cur;
end $$ language plpgsql;

select salcalc('IT');
```

`