PROGRAM 1

Create a table customer (cust_no varchar (5), cust_name varchar (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 customer to cust
h) drop the table

Table structure:-

customer

Attribute	Datatype	Constraint
cust_no	varchar(5)	
cust_name	varchar(15)	
age	number	
phone	varchar(10)	

Cust_phone

Attribute	Datatype	Constraint
cust_name	varchar(15)	
phone	varchar(10)	

Query:-

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

A)
```
insert into customer values(1,'SHAMEEM',19,234578);
insert into customer values(2,'ASWIN',19,7865490);
insert into customer values(3,'ATHUL',19,4567321);
insert into customer values(4,'ASWANI',19,5895624);
insert into customer values(5,'ADITHYA',19,895262);
select * from customer;
```

B).
```
alter table customer add d_birth date;
```

C). create table cust_phone as select cust_name,phone from customer;
D). alter table customer drop 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;

Output

Table created

a). 5 rows inserted

cust_no	cust_name	age	phone
1	SHAMEEM	19	234578
2	ASWIN	19	7865490
3	ATHUL	19	4567321
4	ASWANI	19	5895624
5	ADITHYA	19	895262

b). Table altered
c). Table altered
d). Table altered
e). Table altered
f). 5 rows deleted
g). Table altered
h). Table dropped.

PROGRAM 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

Table structure:-

Salesman

Attribute	Datatype	Constraint
salesman_no	varchar(5)	
s_name	varchar(10)	
place	varchar(15)	
phone	varchar(15)	

Sales_order

Attribute	Datatype	Constraint
order_no	varchar(10)	
order_date	date	
salesman_no	varchar(10)	
del_type	char(5)	
order_status	char(10)	

Query:-

- create table salesman(salesman_no varchar(10) primary key,sname varchar(10) not null,place varchar(10),phone varchar(10) unique);
- create table sales_order(order_no varchar(10) primary key,order_date date not null,salesman_no varchar(10) constraint sno references salesman(salesman_no),del_type char(5) constraint check_del check(del_type in('p','f')),order_status char(10) constraint check_order check(order_status in('inprocess','fulfilled','backorder','cancelled')));

A). insert into salesman values('s1','JERIN','KOZHIKODE','456633');
insert into salesman values('s2','RAMSHAD','TMSY','67633');
insert into salesman values('s3','DIPINESH','KOYILANDI','2562333');

insert into sales_order values('p1','12-June-2018','s1','p','inprocess');
insert into sales_order values('p2','25-May-2018','s2','f','fulfilled');
insert into sales_order values('p3','29-April-2018','s3','p','cancelled');
select * from salesman;
select * from sales_order;

B). alter table salesman drop constraint salesman_pkey;

C). alter table sales_order drop constraint sno;
alter table sales_order drop constraint check_del;
alter table sales_order drop constraint check_order;

D). alter table salesman add primary key(salesman_no);

E) alter table sales_order add constraint sno foreign key(salesman_no) references salesman(salesman_no);
alter table sales_order add constraint check_del check(del_type in('p','f'));
alter table sales_order add constraint check_order check(order_status in('inprocess','fulfilled','backorder','cancelled'));

Output:-

Table created
Table created

A). 5 rows inserted, 5 rows inserted

B). Table altered
Table altered

C). Table altered

D). Table altered.

E). Table altered.

PROGRAM 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’

Table structure:-

Hospital

Attribute	Datatype	Constraints
doctorid	varchar(5)	
doctorname	char(10)	
department	char(10)	
qualification	varchar(10)	
experience	varchar(5)	

Query:-

```
create table hospital(doctorid varchar(5), doctorname char(10), department char(10), qualification
varchar(10),  experience varchar(5));
A). insert into hospital values('d001','Arun','Skin','MBBS','9');
    insert into hospital values('d002','Athira','Ortho','MD','3');
    insert into hospital values('d003','Kavya','Skin','MD','7');
    insert into hospital values('d004','Kiran','ENT','BHMS','5');
    insert into hospital values('d005','Jaya','Gynac','MBBS','8');
B). select * from hospital;
C). select * from hospital where qualification='MD';
D). select doctorname from hospital where experience>'5' and qualification!='MD';
E). select doctorname from hospital where department='Skin';
F). update hospital set experience='5' where doctorid='d003';
G). delete from hospital where doctorid='d005';
```

Output:-

Table created.

a). 5 rows inserted.

b).

doctorid	doctorname	department	qualification	experience
d001	Arun	Skin	MBBS	9
d002	Athira	Ortho	MD	3
d003	Kavya	Skin	MD	7
d004	Kiran	ENT	BHMS	5
d005	Jaya	Gynac	MBBS	8

c).

doctorid	doctorname	department	qualification	experience
D002	Athira	Ortho	MD	3
D003	Kavya	Skin	MD	7

d).

doctorname
Arun
Jaya

e).

doctorname
Arun
Kavya

f). 1 row updated
g).1 row deleted

PROGRAM 4

Create the following tables

Bank_customer (accno primary key, cust_name, place)

Deposit (accno foreign key, 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

Table structure:-

Bank_customer

Attribute	Datatype	Constraint
acc_no	number	
cust_name	varchar(10)	
place	varchar(15)	

Deposit

Attribute	Datatype	constraint
acc_no	number	
deposit_no	number	
damount	number	

Loan

Attribute	Datatype	constraint
loan_no	number	
l_amount	number	

Query:-

```
create table bank_customer(acc_no int primary key, cust_name varchar(10), place varchar(15));
create table deposit (acc_no int references bank_customer(acc_no), deposit_no int,damount int);
create table loan(acc_no int references bank_customer(acc_no), loan_no int, l_amount int);

insert into bank_customer values(12344,'Abin','Kannur');
insert into bank_customer values(12345,'Arun','Kozhikode');
insert into bank_customer values(12346,'Athira','Koyilandi');
insert into bank_customer values(12347,'Samuel','Manjeri');
insert into bank_customer values(12348,'Aparna','Aluva');
insert into bank_customer values(12349,'Kiran','Pattambi');

insert into deposit values(12345,100,25000);
insert into deposit values(12346,101,30000);
insert into deposit values(12347,102,300000);

insert into loan values(12345,200,70000);
insert into loan values(12348,201,65000);
insert into loan values(12349,202,35000);
```

A). select b.acc_no,b.cust_name,b.place,d.deposit_no,d.damount,l.loan_no,l.l_amount from bank_customer b full join deposit d on (b.acc_no=d.acc_no) full join loan l on(b.acc_no=l.acc_no);

B). select cust_name,damount from bank_customer,deposit where bank_customer.acc_no=deposit.acc_no and not exists(select * from loan where loan.acc_no=bank_customer.acc_no);

C). select cust_name,l_amount from bank_customer,loan where bank_customer.acc_no=loan.acc_no and not exists(select * from deposit where deposit.acc_no=bank_customer.acc_no);

D). select cust_name from bank_customer,deposit,loan where bank_customer.acc_no=loan.acc_no and bank_customer.acc_no=deposit.acc_no;

E). select cust_name from bank_customer where acc_no not in(select acc_no from deposit union select acc_no from loan);

Output:-

Table created.

a).

acc_no	cust_name	place	deposit_no	damount	loan_no	l_amount
12344	Abin	Kannur	-	-	-	-

12345	Arun	Kozhikode	100	25000	200	70000
12346	Athira	Koyilandi	101	30000	-	-
12347	Samuel	Manjeri	102	300000	-	-
12348	Aparna	Aluva	-	-	201	65000
12349	Kiran	Pattambi	-	-	202	35000

b).

cust_name	damount
Athira	30000
Samuel	300000

c).

cust_name	l_amount
Aparna	65000
Kiran	35000

d).

cust_name
Arun

e).

cust_name
Abin

PROGRAM 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

Table structure:-

Attribute	Datatype	Constraint
emp_id	number	
emp_name	varchar(10)	
salary	number	
department	varchar(10)	
age	number	

Query:-

```
create table employee(emp_id int, emp_name varchar(10), salary float, department varchar(10), age float);

insert into employee values(101,'JOSEPH',25000,'Sales',30);
insert into employee values(102,'STEPHEN',30000,'Sales',37);
insert into employee values(103,'DAVID',18000,'Sales',21);
insert into employee values(104,'JOHN',15000,'Marketing',25);
insert into employee values(105,'JAMES',21000,'Purchasing',32);
insert into employee values(106,'KARTHIK',25000,'Purchasing',35);

A). select count(*) from employee;
B). select emp_name,department,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='Purchasing';
G). select max(salary) from employee where department='Sales';
H). select max(salary)-min(salary) as salary_difference from employee;
```

Output:-

Table created.

6 rows inserted.

a).

count(*)
6

b).

emp_name	department	age
STEPHEN	Sales	37
JOHN	Marketing	25
KARTHIK	Purchasing	35

c).

department	avg(age)
Purchasing	33.5
Sales	29.33
Marketing	25

d).

department	avg(salary)
Purchasing	23000
Sales	24333.33
Marketing	15000

e).

min(salary)
15000

f).

count(*)
2

g).

max(salary)
30000

h).

salary_difference
15000

PROGRAM 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'

Table structure:-

Attribute	Datatype	Constraints
product_code	varchar(4)	
product_name	char(10)	

category	char(20)	
quantity	number	
price	number	

Query:-

```
create table product(product_code varchar(4) primary key,product_name char(10),category char(20), quantity int,price int);
insert into product values('p111','Dove','Bathsoap',600,38);
insert into product values('p112','Sunlight','Washing powder',200,65);
insert into product values('p113','Chandrika','Handwash',500,20);
insert into product values('p114','Santoor','Bathsoap',100,30);
insert into product values('p115','Ariel','Washing powder',60,10);
insert into product values('p116','Colgate','Paste',120,25);
insert into product values('p117','Closeup','Paste',150,35);
insert into product values('p118','Sandal','Bathsoap',650,80);
insert into product values('p119','Gathri','Washing powder',100,30);
insert into product values('p120','Kabani','Washing soap',650,70);
```

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('Bathsoap','Washing powder','Paste');
D). select * from product where (quantity<100 or quantity>500);
E). select product_name from product where product_name like 'S%';
F). select product_name from product where category <>'Paste';
G). select product_name from product where product_name like '_u%' and category='Washing powder';

Output:-

Table created.
10 rows inserted.
a).

product_code	product_name	category	quantity	price
p112	Sunlight	Washing powder	200	65
p114	Santoor	Bathsoap	100	30
p118	Sandal	Bathsoap	650	80

p120	Kabani	Washing soap	650	70
p119	Gathri	Washing powder	100	30
p111	Dove	Bathsoap	600	38
p116	Colgate	Paste	120	25
p117	Closeup	Paste	150	35
p113	Chandrika	Handwash	500	25
p115	Ariel	Washing powder	60	10

b).

product_code	product_name
p111	Dove
p113	Chandrika
p114	Santoor
p116	Colgate
p117	Closeup
p119	Gathri

c).

product_code	product_name	category	quanitity	price
p111	Dove	Bathsoap	600	38
p112	Sunlight	Washing powder	200	65
p114	Santoor	Bathsoap	100	30
p115	Ariel	Washing powder	60	10
p116	Colgate	Paste	120	25
p117	Closeup	Paste	150	35
p118	Sandal	Bathsoap	650	80
p119	Gathri	Washing powder	100	30

d).

product_code	product_name	category	quantity	price
p111	Dove	Bathsoap	600	38
p115	Ariel	Washingpowder	60	10
p118	Sandal	Bathsoap	650	80
p120	kabani	Washingsoap	650	70

e).

product_name
Sunlight
Santoor
Sandal

f).

product_name
Dove
Sunlight
Chandrika
Santoor
Ariel
Sandal
Gathri
Kabani

g).

product_name
sunlight

PROGRAM 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.

Table structure:-

Employee

Attribute	Datatype	Constraints
emp_name	varchar(20)	
city	varchar(20)	

Works

Attribute	Datatype	Constraints
emp_name	varchar(20)	
company_name	varchar(20)	
salary	number	

Company

Attribute	Datatype	constraints
company_name	varchar(20)	
city	varchar(20)	

Manages

Attributes	Datatype	Constraints
emp_name	varchar(20)	
manager_name	varchar(20)	

Query:-

```
create table employee(emp_name varchar(20),city varchar(20));
create table works(emp_name varchar(20),company_name varchar(20),salary int);
create table company(company_name varchar(20),city varchar(20));
create table manages(emp_name varchar(20),manager_name varchar(20));

insert into employee values('Ajith','Tirur');
insert into employee values('Alex','Tanur');
insert into employee values('Shahina','Manjeri');
insert into employee values('Sindhu','Calicut');
insert into employee values('Balan','Vadakara');

insert into works values('Ajith','Infosys',25000);
insert into works values('Alex','TCS',35000);
insert into works values('Shahina','Wipro',20000);
insert into works values('Sindhu','Infosys',10000);
insert into works values('Balan','Infosys',15000);

insert into company values('Infosys','Calicut');
insert into company values('Wipro','Trivandrum');
insert into company values('TCS','Kochi');
insert into manages values('Ajith','Sachin');
insert into manages values('Alex','Sayooj');
insert into manages values('Shahina','Akhila');
insert into manages values('Sindhu','Kiran');
insert into manages values('Balan','Salman');

A). select emp_name from works where company_name='Infosys';
B). select employee.emp_name,employee.city from employee,works where
employee.emp_name=works.emp_name and works.company_name='Wipro';
C). select employee.emp_name,employee.city from employee,works where
employee.emp_name=works.emp_name and works.company_name='Infosys' and salary>10000;
```

D). select employee.emp_name from employee,company where employee.city=company.city;
E). select emp_name from works where company_name!='Wipro';
F). select company_name,count(*) as no_of_employees from works group by company_name having count(distinct emp_name)>=all(select count(distinct emp_name) from works group by company_name);

Output:-

Table created.
Table created.
Table created.
Table created.

a).

emp_name
Ajith
Sindhu
Balan

b).

emp_name	city
Shahina	Manjeri

c).

emp_name	city
Ajith	Tirur
Balan	Vadakara

d).

emp_name
Sindhu

e).

emp_name
Ajith
Alex
Sindhu
Balan

f).

company_name	no_of_employees
Infosys	3

PROGRAM 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.

Table structure:-

Attribute	Datatype	Constraint
radius	number	
area	float	

Query:-

```
create table areas(radius float,area float);
create or replace function calcarea() returns void as
'declare
rad float:=3;
area float;
```

```
begin
loop
area:=3.14*rad*rad;
insert into areas values (rad,area);
rad:=rad+1;
exit when rad>7;
end loop;
end;
'language 'plpgsql';

select calcarea();
select * from calcarea();
```

Output:-

Table created.

Function created.

radius	area
3	28.26
4	50.24
5	78.5
6	113.04
7	153.86

PROGRAM 9

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

Table structure:-

Attribute	Datatype	Constraint
cons_no	int	Primary key
units	int	
amount	float	

Query:-

```
create table bill(cons_no int primary key,units int,amount float);
create or replace function elecbill(int,int) returns void
language plpgsql
as $$
declare
cons_no alias for $1;
units alias for $2;
amount float;
begin
amount:=units*6.40;
insert into bill values(cons_no,units,amount);
end;
$$

select elecbill(123,216);
select * from bill;
```

Output:-

Table created.

cons_no	units	amount
123	216	1382.4

PROGRAM 10

Create a procedure to print Fibonacci number up to a limit, limit is passed as an argument

Query:-

```
create or replace function fibo(int) returns text as
'declare
a int:=0;
b int:=1;
c int;
n alias for $1;
begin
raise notice "The fibonacci series is :";
while a<=n
loop
raise  notice "%",a;
c:=a+b;
a:=b;
b:=c;
end loop;
end;
'language'plpgsql';

select fibo(15);
```

Output:-

Fibonacci series is :
0
1
1
2
3
5
8
13

PROGRAM 11

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

Query:-

```
create or replace function prime(int) returns text
language plpgsql
as $$
declare
n alias for $1;
i int:=2;
counter int:=1;
msg text;
begin
for i in 2..n/2 loop
if mod(n,i)=0 then
counter:=0;
exit;
end if;
end loop;
if counter = 1 then
msg:=n||' is a prime number';
else
msg:=n||' is not a prime number';
end if;
return msg;
end;
$$

select prime(31);
select prime(5);
select prime(4);
```

Output:-

```
Function created.
31 is a prime number.
5 is a prime number.
4 is not a prime number.
```

PROGRAM 12

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

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

Table structure:-

Attribute	Datatype	Constraint
empno	int	Primary key
ename	varchar(25)	
dept	varchar(25)	
salary	int	

Query:-

```
create table emp_salary(empno int primary key,ename varchar(25),dept varchar(25),salary int);
insert into emp_salary values(100,'Irshad','sales',10000);
insert into emp_salary values(101,'Arun','sales',13000);
insert into emp_salary values(103,'Akhil','production',15000);
insert into emp_salary values(104,'Arya','production',20000);
insert into emp_salary values(102,'Deepak','hr',18000);

create or replace function avgsal(varchar) returns int
language plpgsql
as $$
declare
dname alias for $1;
avgs int:=0;
begin
select avg(salary) into avgs from emp_salary where dept=dname;
return avgs;
end;
$$

select avgsal('sales');
```

```
select avgsal('production');
select avgsal('hr');
```

Output:-

```
Table created
Function created.
avgsal : 11500
avg sal: 17500
avgsal: 18000
```