Q.1)Consider the following database:
Room (room_no, room_name, room_type, charges)
Guest (Guest_code, Gname, city, no_of persons)
The relationship is as follows: Room-Guest: one-to-one. The room_type can have values as either ‘AC’ or ‘NonAC’.

A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] Execute the following queries in PostGreSQL

```
create table Room (room_no int primary key, room_name text, room_type text
check(room_type in ('AC','NONAC')), charges float);

create table Guest (Gcode int primary key, Gname text, city text, nop int,rno int
references Room unique not null);
```

i) List all guests whose name starts with “S”.

```
select * from guest where gname like 'S%';
```

ii) Increase the charges of all AC rooms by 15%.

```
update room set charges=charges+charges*0.15;
```

iii) List the minimum charges of a room.

```
select min(charges) as "Minimum Charges" from room;
```

iv) List the names of the guests in the sorted order by city name.

```
select gname from guest order by city;
```

B)Write a procedure to find sum and product of two numbers.[10 Marks]

```
create or replace function sum_product(int,int)
returns void as'
declare
sum int;
prod int;
begin
sum=$1+$2;
raise notice ''Sum of two number is:%'',sum;
prod=$1*$2;
raise notice ''Product of two number is:% '',prod;
end;'
language 'plpgsql';
```

Execution:

```
Test=# select sum_product(3,6);
NOTICE:  Sum of two number is:9
NOTICE:  Product of two number is:18
```

2

Q.1)Consider the following database:
College (cno, cname, street name, ccity)
Principal (pno, pname, experience, Salary)
The relationship is as follows: College-Principal: one-to-one. Experience must greater than 10 years.
A)Create above database in PostgreSQL and insert sufficient records. [10 Marks]

Table creation:

```
create table college(cno int primary key,cname text, sname text,ccity text);
create table principal(pno int primary key, pname text,exp int check (exp>10),salary
float, cno int references college unique not null);
```

---

Execute the following queries in PostGreSQL

i)Display all colleges whose name contains 'and'.

```
  select cname from college where cname like '%and%';
```

ii)List the average salary of a Principal.

```
  select avg(salary) from principal;
```

iii)List the names of all Principals having experience between 10 to 20 years.

```
  select pname from principal where exp between 10 and 20;
```

iv)Change the street name of college from MG Road to Nehru road.

```
  update college set sname='Nehru road' where sname='MG Road';
```

---

B)Write a stored procedure to insert a record in table College.
[10 Marks]

---

```
create or replace function insert_record(int,text,text,text)
returns void as'
declare
begin
insert into college (cno,cname,sname,ccity) values ($1,$2,$3,$4);
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select insert_record(101,'DYPatil','Pimpri','Pune');
```

---

---

3

Q.1)Consider the following database:
Employee(eno, ename, designation, salary)
Department(dno, dname, location)
The relationship is as follows: Employee-Department: many-to-one. Location should not be null.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] Execute the following queries in PostGreSQL

---

```
create table Depart2(dno int primary key, dname text,loc text);
create table Emp2(eno int primary key,ename text, desig text, salary float, dno int
references Depart2);
```

---

i)Give a 5% raise in salary to all the employees.

```
   update emp2 set salary=salary+salary*0.05;
```

ii)Display average salary of an employee.

```
   select avg(salary) from emp2;
```

iii)List the details of all the departments located at city_____.

```
   select Depart2.* from Depart2 where loc='Pune';
```

iv)Display the details of employees whose names ends with an alphabet "r".

```
   select Emp2.* from Emp2 where ename like '%r';
```

---

B)Write a stored function using cursors to display all the details of Employee whose salary is more than 80,000.[10 Marks]

```
create or replace function disp_Emp()
returns void as'
declare
rec record;
c1 cursor for select Emp2.* from Emp2 where salary > 80000;
```

```
begin
open c1;
loop
fetch c1 into rec;
exit when not found;
raise notice ''Employee Details are:% % % % '',rec.eno,rec.ename,rec.desig,rec.salary;
end loop;
close c1;
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select disp_emp();
NOTICE:  Employee Details are: 1 Ram HR 90000
```

---

4

Q.1)Consider the following database:
Person (pnumber, pname, birthdate, income)
Area (area_code, aname, area_type, pincode)
The relationship is as follows: Person-Area: many-to-one. The area_type can have values as either “urban” or “rural”.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks]

---

```
create table Area1 (acode int primary key, aname text, atype text check(atype in
('U','R')),pincode numeric);
create table Person1 (pno int primary key, pname text, bdate date, income float, acode
int references area1);
```

Execute the following queries in PostGreSQL

---

i)List the details of all people whose name starts with the  alphabet “R”.

```
select * from person1 where pname like 'R%';
```

ii) Display the details of people in the sorted order of their  income.

```
        select * from person1 order by income;
```

iii)Display the count of areas of “urban” type.

```
     select count(*) from area1 where atype='U';
```

iv)   Change the pincode of “kalyaninagar” to 411036.

```
        update area1 set pincode=411036 where aname='kalyaninagar';
```

---

B)Create a stored procedure named as “addrecords” for adding person records.[10 Marks]

---

```
create or replace function addrecords(int,text,date,float,int)
returns void as'
declare
begin
insert into person1 (pno, pname,bdate,income,acode) values ($1,$2,$3,$4,$5);
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select addrecords(1,'Rahul','2-2-1980',20000,101);
```

---

5

Q.1)Consider the following database:
Doctor (dno, dname, addr, phone_no, specialization)
Patient (pno, pat_name, city, disease)
The relationship is as follows: Doctor-Patient: many-to-many.

A)Create above database in PostgreSQL and insert sufficient records.[10 Marks]

```
create table Doctor (dno int primary key, dname text, addr text, phone_no numeric, spe
text);
create table Patient (pno int primary key, pname text, city text, dis text);
create table DP (dno int references doctor, pno int references patient);
```

Execute the following queries in PostGreSQL

i) Find the names of all doctors which start with “M”.

```
select dname from doctor where dname like 'M%';
```

ii) Count the number of doctors who are Neurologists.

```
select count(*) from doctor where spe='Neurologists';
```

iii)Give the list of all patients who are suffering from “Fever”.

```
select * from patient where dis='Fever';
```

iv) Find the specialization and phone numbers of all doctors from Alandi.

```
select spe,phone_no from doctor where addr='Alandi';
```

B)Write a stored function using cursors to display all the details of all Patients from Nashik city.[10 Marks]

```
create or replace function disp_details()
returns void as'
declare
rec record;
c1 cursor for select patient.* from patient where city=''Nashik'';
begin
open c1;
loop
fetch c1 into rec;
exit when not found;
raise notice ''Patient Datils are: % % % % '',rec.pno,rec.pname,rec.city,rec.dis;
end loop;
close c1;
end;'
language 'plpgsql';
```

Execution:

```
Test=# select disp_details();
NOTICE:  Patient Datils are: 101 Rahul Nashik Fever
```

6

Q.1)Consider the following database:
Student (rno, name, city)
Teacher(tno, tname, phone_no, salary)
The relationship is as follows: Student-Teacher: many-to-many with subject as a descriptive attribute.

A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

```
create table Student (rno int primary key, name text, city text);
create table Teacher(tno int primary key, tname text, phone_no numeric, salary float);
create table ST (rno int references Student, tno int references teacher, sub text);
```

i)List all student whose name start from ‘Sh’ .

```
select name from student where name like 'Sh%';
```

ii)Display the count of students from city_____.

```
select count(*) from student where city='Pune';
```

iii)Find the maximum salary of teachers.

```
select max(salary) from teacher;
```

iv)Change the phone number of “Prof. Satkar” to “9822131226”

```
upadte teacher set phone_no=9822131226 where tname='Prof.Satkar';
```

B)Create a stored procedure named as “updaterecords” to give 5% rise in salary of teacher.[10 Marks]

```
create or replace function updaterecords()
returns void as'
declare
begin
update teacher set salary=salary+salary*0.05;
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select updaterecords();
```

---

7

Q.1) Consider the following database:
Policy (pno, pname, premium_amt, policy_type)
Customer (cno, cname. city, agent_name)
The relationship is as follows: Policy-Customer: many-to-one. The “policy_type” can have values as “Yearly”, “Half-yearly” or “Monthly”
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

---

```
create table Customer (cno int primary key, cname text, city text, agent_name text);
create table Policy (pno int primary key, pname text, p_amt float, p_type text
check(p_type in('Y','HY','M')),cno int references Customer);
```

i) List the details of all customers who live in ___city.

```
select * from customer where city='Pune';
```

ii) Display the average premium amount.

```
select avg(p_amt) from  customer;
```

iii) Increases the premium amount for Monthly policies by 10%.

```
update Policy set p_amt=p_amt+p_amt*0.01;
```

iv) Display the policy type wise count of policies.

```
select p_type,count(*)
from policy group by  p_type;
```

---

B)Create a stored function named as names as “max_premium” which will find max premium amount.[10 Marks]

```
create or replace function max_premium()
returns void as'
declare
amt float;
begin
select into amt max(p_amt) from policy;
raise notice ''Maximum Premium amount is:= % '',amt;
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select max_premium();
```

---

8

Q.1)Consider the following database:
Item (item_no, name, quantity)
Supplier (s_no, name, city)
The relationship is as follows: Item-Supplier: many-to-many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

---

```
create table Item (ino int primary key, iname text, quan int);
create table Supplier (sno int primary key, sname text, city text);
```

```
create table ItS (ino int references Item, sno int references Supplier);
```
---
i)Change the quantity for item “Mouse” to 800.
```
  update item set quan=800 where iname='Mouse';
```
ii)List the details of the suppliers whose name begins with the alphabet “M”.
```
   select * from supplier where sname like 'M%';
```
iii)Display the count of items.
```
   select count(*) as "ItemCount" from item;
```
iv)List the names of suppliers who do not live in city_____.
```
   select name from supplier where city <> 'Pune';
```
---
B)Write a stored function to find the minimum quantity of item.[10 Marks]
```
create or replace function min_quan()
returns void as'
declare
mq float;
begin
select into mq min(quan) from Item;
raise notice ''Minimum quantity is:= % '',mq;
end;'
language 'plpgsql';
```
---
Execution :
```
select min_quan();
```
---

9

Q.1)Consider the following database:
Student (sno , s_name, s_class)
s_class can be either "FY", "SY" or "TY"
Teacher (tno , t_name, yrs_experience )
The relationship is as follows: Student-Teacher: M-M with descriptive attribute subject.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL
---
```
create table Stud2(sno int primary key, s_name text, s_class text check(s_class
in('FY','SY','TY')));
create table Teach (tno int primary key , t_name text,exp);
create table ST1(sno int references stud2 on delete cascade, tno int references teach,
sub text);
```
---
i)Give class-wise number of students.
```
   select s_class, count(*) from stud2 group by s_class;
```
ii)List all students studying in class “TY”.
```
   select * from stud2 where s_class='TY';
```
iii)Count the number of students who have taken subject “____ ”.
```
   select count(*) from st1 where sub='C';
```
iv)Delete record of student whose sno = 101.
```
   delete from stud2 where sno=101;
```
---
B)Write a stored function to take teacher name as input and returns the years of experience of that teacher.[10 Marks]
```
create or replace function disp_teach(text)
returns void as'
declare
yexp int;
begin
select into yexp exp from teach where tname=$1;
raise notice ''Years of Experience is := % '',yexp;
end;'
language 'plpgsql';
```

```
-------------------------------------------------------
Test=# select disp_teach('Seema');
NOTICE:  Years of Experience is := 24
-------------------------------------------------------------------
```

10

Q.1)Consider the following database:
Account ( acct_no, acct_type, balance, branch_name)
Customer (cust_no, cust_name, cust_city)
Relationships: Customer-Account :1-M. acct_type can be “saving” or “current”
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

```
-------------------------------------------------------------------
create table Account (acct_no int primary key, acct_type text check(acct_type
in('saving','current')), bal float, bname text);
create table Cust2 (cust_no int primary key, cust_name text, cust_city text, acct_no int
references Account);
-------------------------------------------------------------------
```

i)Display information of all saving accounts having balance > 500,000

```
      select * from account where bal>500000;
```

ii)Count customers whose name starts with ‘r’.

```
      select count(*) from Cust2 where cust_name like 'r%';
```

iii)Find the total balance at branch “M.G.Road”.

```
      select sum(bal) from account where bname='M.G.Road';
```

iv)Delete the record whose cust_name is _____.

```
      delete from cust2 where cust_name='Satish';
-------------------------------------------------------------------
```

B)Write a stored function using cursors to print names of all customers from city ___.[10 Marks]

```
create or replace function disp_cust2(text)
returns void as'
declare
rec record;
c1 cursor for select * from cust2 where cust_city=$1;
begin
open c1;
loop
fetch c1 into rec;
exit when not found;
raise notice ''Details of Customer are: %'',rec.cust_name;
end loop;
close c1;
end;'
language 'plpgsql';
-----------------------------------------------------
Test=# select disp_cust2('Satara');
NOTICE:  Details of Customer are: Satish
-------------------------------------------------------------
```

11

Q.1)Consider the following database:
Bus ( Bus_no , capacity ,depot_name)
Route (Route_no ,source ,destination ,no_of_stations )

Relationship : Bus-Route : M-1. Bus capacity is not null
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

```
-------------------------------------------------------------------
create table Route (R_no int primary key,src text ,dest text ,nos int);
```

```
create table Bus( B_no int primary key, cap  text ,depot_name text, R_no int references
Route);
i)List all buses which belongs to depot _____.
    select * from bus where  depot_name='Kothrud';
ii)Delete Bus details whose Bus number is _____.
    delete from bus where B_no=101;
iii)List the route details having number of stations > 10.
    select * from route where nos>10;
iv)List all routes starting from Station ________.
    select * from route where src='Pune';
```

---

B)Write a stored function using cursors to accept route_no from the user and display number of stations of that route.[10 Marks]

```
create or replace function disp_nos(int)
returns void as'
declare
rno int;
c1 cursor for select nos from route where r_no=$1;
begin
open c1;
loop
fetch c1 into rno;
exit when not found;
raise notice ''Number of stations are: %'',rno;
end loop;
close c1;
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select disp_nos(101);
NOTICE:  Number of stations are: 4
```

---

12

Q.1)Consider the following database:
Game (gcode, gname, noofplayers, coachname, captain_name)
Player (pno, pname)
There exists a one-to-many relationship between Game and Player
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

---

```
create table Game (gcode int primary key, gname text, nop int, coachname text, cname
text);
create table Player (pno int primary key, pname text, gcode int references game on delete
cascade);
i)      Display all game names that ends with “ball”.
        select gname from game where gname like '%ball';
ii)    Give the average number of players.
        select avg(nop) from player;
iii)   Display all details of game “kho kho“.
        select * from game where gname='kho kho';
iv)   Update the coach name from “____” to “____” for game “hockey”.
       update game set coachname='Satish' where gname='hockey';
```

---

B)Create a stored procedure named as “deleterecords“ for deleting the Game record having coach name _____.[10 Marks]

```
create or replace function deleterecords(text)
returns void as'
declare
begin
delete from game where coachname=$1;
raise notice ''Record deleted successfully..... '';
```

```
end;'
language 'plpgsql';
-------------------------------------------------------------
Test=# select deleterecords('Satish');
NOTICE:Record deleted successfully.....
-------------------------------------------------------------------
```

13

Q.1)Consider the following database:
Item (item_no, name, quantity, rate)
Supplier (s_no, name, city, contact)
The relationship is as follows: Item-Supplier: many-to-many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

```
-------------------------------------------------------------------
create table Item (ino int primary key, iname text, quan int,rate float);
create table Supplier (sno int primary key, sname text, city text);
create table ItS (ino int references Item, sno int references Supplier);
-------------------------------------------------------------------
```

i)List the details of the suppliers whose name begins with the alphabet "P".

```
      select * from supplier where  sname like 'P%';
```

ii)Delete record of item_no _____ .

```
      delete from item where ino=101;
```

iii)Display the count of items with rate > 50Rs.

```
      select count(*) from item where rate>50;
```

iv)List the names of suppliers live in city.

```
      select sname from supplier where city='Pune';
-------------------------------------------------------------
```

B)Write a function to find the details of items whose quantity is greater than 30.[10 Marks]

```
create or replace function disp_details()
returns void as'
declare
rec record;
begin
for rec in select * from item where quan>30
loop
raise notice '' % % % '', rec.ino,rec.iname,rec.quan;
end loop;
end;'
language 'plpgsql';
-------------------------------------------------
Execution:
Test=# select disp_details();
NOTICE:   101 Pen 34
NOTICE:   102 Pencil 36
-------------------------------------------------
```

14

Q.1)Consider the following database:
Book (Book_no, title, author, price, year_published)
Customer (cid, cname, addr)
Relation between Book and Customer is Many to Many with quantity as descriptive attribute.
A)Create above database in PostgreSQL and insert sufficient records.[10 marks] and Execute the following queries in PostGreSQL

```
-------------------------------------------------------------------
create table Book1 (B_no int primary key, title text, author text, price float, yp int);
create table Customer (cid int primary key, cname text, addr text);
create table BC (B_no int references Book, cid int references customer, quan int);
```

i)  Display customer details staying at “Pune”.

```
     select * from customer where addr='Pune';

ii)    Display author wise details of book.
        select author,title,price,yp from book group by author,title,
        price,yp;

iii)   Display the average price of a book.
        select avg(price) from book;

iv)    Delete the record from book table with Book_no ____.
        delete from book where B_no=101;
```

---

B)Write a function, to define a cursor to print the details of the Books published in year 2024.[10 marks]

```
create or replace function disp_book()
returns void as'
declare
rec record;
c1 cursor for select * from book1 where yp=2024;
begin
open c1;
loop
fetch c1 into rec;
exit when not found;
raise notice ''Details of Book are: % % % %'',rec.B_no,rec.title, rec.author,rec.price;
end loop;
close c1;
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select disp_book();
NOTICE:  Details of Book are: 101 C Sujata 120
```

---

15

Q.1)Consider the following database:
Sales_order(s_orderno, s_order_date, order_amt)
Client(client_no, name, address)
The relationship is as follows:Client and Sales_order: one-many. order_amt should be > 0

A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

---

```
create table Client2(client_no int primary key, name text, add text);
create table Sales_order2(ono int primary key, odate date, oamt float check(oamt>0),
client_no int references Client2);
```

---

```
i)Display all sale records having order date before "____".
  select * from sales_order2 where odate < '2024-12-10';
 ii)Find maximum sales order amount.
   select max(oamt) from Sales_order2;
iii)Update the client address of all clients from "Nasik" to
    "Ahilyanagar".
   update client2 set add='Ahilyanagar' where add='Nasik';
iv)Add column order_status to the Sales_order table.
   alter table sales_order2 add ostatus text;
```

---

B)Create a stored procedure named as "addrecords" for adding new sales order records. [10 Marks]

```
create or replace function add_records(int,date,float,int,text)
returns void as'
declare
```

```
begin
insert into sales_order2(ono,odate,oamt,client_no,ostatus) values ($1,$2,$3,$4,$5);
end;'
language 'plpgsql';
------------------------------------------------------------
Execution:
Test=# select add_records(1,'2-2-2024',20000,101,'C');
------------------------------------------------------------------------------------------------
-------------------------------
```

16

Q.1)Consider the following database:
Car(car_code, c_name, c_price, color_type)
color_type can be “metallic” or “solid”
Customer (cust_code, cust_name, cust_address)
The relationship is as follows: Customer and car: one-to-many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

```
------------------------------------------------------------------
create table Cust3 (cust_code int primary key,cust_name text, cadd text);
create table Car(car_code int primary key,c_name text,c_price float, ctype text,
cust_code int references Cust3);
------------------------------------------------------------------
```

i)Find the names of all Customers whose name start with “B”.

```
   select cust_name from Cust3 where cust_name like 'B%';
```

ii)Count the number of “metallic” cars.

```
  select count(*) from car where ctype='metallic';
```

iii)Give the list of all customers staying in ShivajiNagar.

```
   select cust_name from cust3 where cadd='ShivajiNagar';
```

iv)Increase the price of all “Ferrari” cars by 15%.

```
    update car set c_price=c_price+0.15;
------------------------------------------------------
```

B)write a stored function to display details of all metallic coloured cars having price in the range 100000 to 500000.[10 Marks]

```
create or replace function disp_car()
returns void as'
declare
rec record;
begin
for rec in select * from car where c_price between 100000 and 500000
loop
raise notice '' % % % %'', rec.car_code,rec.c_name,rec.c_price,rec.ctype;
end loop;
end;'
language 'plpgsql';
------------------------------------------------
Execution:
Test=# select disp_car();
------------------------------------------------------------------
```

17

Q.1)Consider the following database:
Property (pno, description, area, rate) rate should be > 0
Owner (owner_name, city, phno)
The relationship is as follows: owner and Property : One to Many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks]and Execute the following queries in PostGreSQL

```
------------------------------------------------------------
create table Owner (owner_name text primary key, city text, phno numeric);
```

```
create table Property (pno int primary key, descri text,area text, rate float,owner_name
text references owner);
-------------------------------------------------------------
i)List the name of owners that ends with letter ‘a’.
select owner_name from Owner where owner_name like 'a%';
ii)Display the average rate of a property.
select avg(rate) from Property;
iii)Update the phone Number of “Dr. Vikas” to 8856916175.
update Owner set phno=8856916175 where owner_name='Dr. Vikas';
iv)Display area wise property details.
select area,pno,descri,owner_name from Property group by area,pno,descri,owner_name;
-------------------------------------------------------------------------------------------
-----
B)Create a stored function named as “min_price” which will find minimum rate of
property.[10 Marks]
create or replace function min_price()
returns void as'
declare
min_rate float;
begin
select into min_rate min(rate) from property;
raise notice ''Minimum rate is % '',min_rate;
end;'
language 'plpgsql';
-----------------------------------------------------------------------
Test=# select min_price();
-----------------------------------------------------------------
```

18

```
Q.1)Consider the following database:
Employee (emp_no, emp_name, city, designation, salary)
Project (project_no, project_name, status, start_date)
The relationship is as follows: Employee and Project: many-to-one.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and
Execute the following queries in PostGreSQL
-----------------------------------------------------------------
create table Project2 (project_no int primary key, project_name text, status text, sdate
date);
create table Employee (emp_no int primary key, emp_name text, city text, desig text,
salary float,project_no int references project2);
-----------------------------------------------------------------
i)Add constraint status. The value of status should be “Complete”, “In progress”.
       alter table project2 add constraint status_check check
        (status in ('Complete','In progress'));
ii)Count the number of Projects which are “in progress”.
        select count(*) from project2 where status='In progress';
iii)Increase the salaries of all employees working on project 10 by 5%.
     update Employee set salary=salary+salary*0.05 where
     project_no=10;
iv)Display names of all completed projects.
     select project_name from project2 where status='Complete';
-------------------------------------------------------------------------------------------
---------
B)Create a stored function named as names as “max_salary” which will find maximum salary
of an employee.[10 Marks]
create or replace function max_salary()
returns void as'
declare
max_sal float;
begin
select into max_sal max(salary) from employee;
raise notice ''Maximum Salary is % '',max_sal;
end;'
language 'plpgsql';
```

```
-------------------------------------------------------------
Execution:
Test=# select max_salary();
NOTICE:  Maximum Salary is 20000
-------------------------------------------------------------------------
```

19

Q.1)Consider the following database:
Project (pno, pname, start_date, budget, status)
Project Status Constraints: C – completed, PProgressive, I-Incomplete
Department (dno, dname, HOD, no_of_staff)
The relationship is as follows: Project- Department Many to One.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

```
-----------------------------------------------------------
create table Depart1 (dno int primary key, dname text,HOD text, nos int);
create table Project1 (pno int primary key, pname text, sdate date, budget float, status
text check (status in ( 'C','I','P')), dno int references depart1);
 i)    Display the project names that have start date as 12/6/2019.
        select pname from Project1 where sdate='2019-06-12';
ii)    Display the total budget of projects.
        select sum(budget) from Project1;
iii)   Display the HOD name of Computer department.
        select hod from depart1 where dname='Computer';
iv)   all project names having budget more than 30000.
        select pname from Project1 where budget >30000;
-------------------------------------------------------------------
```

B)Write a stored function using cursors to display names of all projects which are “in progress”.[10 Marks]

```
create or replace function disp_proj()
returns void as'
declare
rec record;
c1 cursor for select * from project1 where status=''P'';
begin
open c1;
loop
fetch c1 into rec;
exit when not found;
raise notice ''Details of Project are: % % % %'',rec.pno,rec.pname, rec.sdate,rec.budget;
end loop;
close c1;
end;'
language 'plpgsql';
-----------------------------------------------------
Test=# select disp_proj();
NOTICE:  Details of Project are: 1 Robot 2024-12-12 50000
-----------------------------------------------------------------------
```

20

Q.1) Consider the following database:
Bus (bus_no, capacity, depot_name)
Driver (driver_no, driver_name, license_no, address, age)
The relationship is as follows: Bus and Driver: M-M with Date_of_duty.the descriptive attribute
A) Create above database in PostgreSQL and insert sufficient records.
and Execute the following queries in PostGreSQL [10 Marks]

```
-----------------------------------------------------
create table Bus (bus_no int primary key, cap text, depot_name text);
create table Driver(driver_no int primay key, dname text, lic_no numeric, address text,
age int);
create table DB(bus_no int references Bus, driver_no int references Driver, dduty date);
```

```
-----------------------------------------------------------------
i)     Find the number of buses having capacity more than 20.
        select count(*) from bus where cap> 20;
ii)    Count number of drivers having age > 40.
        select count (*) from driver where age>40;
iii)   Give the names of all drivers starting with 'S'.
        select dname from driver where dname like 'S%';
iv)  Display all bus details of _____depot.
      select bus.* from bus where depot_name='Kothrud';
-----------------------------------------------------------------------------------------
B) Write a stored procedure to find maximum of two numbers.    [10 Marks]
create or replace function max_no(int,int)
returns void as'
declare
begin
if($1 > $2) then
raise notice ''Maximum Number is:% '',$1;
else
raise notice ''Maximum Number is: %'',$2;
end if;
end;'
language 'plpgsql';
-----------------------------------------------
Test=# select max_no(13,4);
NOTICE:  Maximum Number is:13
-----------------------------------------------------------------
```

21

```
Q.1)Consider the following database:
Customer (cust_no, cust_name, city)
Loan (loan_no, loan_amt)
loan_amt should be > 0.
Relation between Customer and Loan is Many to Many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and
Execute the following queries in PostGreSQL
-----------------------------------------------------------
create table Customer1 (cust_no int primary key,cust_name text,city text);
create table Loan (loan_no int primary key,loan_amt float check(loan_amt>0));
create table CL (cust_no int references Customer1, loan_no int references loan);
-----------------------------------------------------------
i)List all customers whose name starts with 'A'.
   select * from Customer1 where cust_name like 'A%';
ii)Display city-wise customer names.
   select city,cust_name from Customer1 order by city,cust_name;
iii)Display all loan numbers whose amount is more than 2 lakhs.
   select loan_no from loan where loan_amt >200000;
iv)Change city 'Pune' to 'Mumbai' for customer '____'.
   update Customer1 set city='Mumbai' where city='Pune' and
   cust_name='Satish';
-----------------------------------------------------------------
B)Write a stored function using cursors to display details of all customers sorted by
city names. [10 Marks]
create or replace function disp_cust()
returns void as'
declare
rec record;
c1 cursor for select * from Customer1 order by city;
begin
open c1;
raise notice '' Details are :'';
loop
fetch c1 into rec;
exit when not found;
```

```
raise notice ''% % '',rec.cust_no,rec.cust_name;
end loop;
close c1;
end;'
language 'plpgsql';
-------------------------------------
Execution:
Test=# select disp_cust();
-------------------------------------------------------------------------------------------
--------------------
```

22

Q.1)Consider the following database:
Customer (cust_no, cust_name, city)
product (product_no, pname, price) price should be > 0.
Relation between Customer and product is Many to Many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and
Execute the following queries in PostGreSQL

```
-------------------------------------------------------------------------------------------
----------
create table Cust (cno int primary key, cname text, city text);
create table prod (pno int primary key, pname text, price float check(price>0));
create table CP(cno int references cust, pno int references prod);
-----------------------------------------------------------------
```

i) List all customers whose name ends with 'A'.

```
 select * from cust where cname like '%A';
```

ii) Count number of products whose price is more than 1000.

```
 select count(*) from prod where price >1000;
```

iii) Increase price of all products by 5%.

```
  update prod set price=price+price*0.05;
```

iv) Display details of customer who are from _____city.

```
 select * from cust where city='Pune';
-------------------------------------------------------------------------------------------
```

B)Create a stored procedure named as "addrecords" to add customer record.[10 Marks]

```
create or replace function addrecords(int,text,text)
returns void as'
declare
begin
insert into cust (cno,cname,city) values ($1,$2,$3);
end;'
language 'plpgsql';
-----------------------------------------------------------------
Execution:
Test=# select addrecords(1,'Suresh','Pune');

-----------------------------------------------------------------
```

23

Q.1)Consider the following database:
Student (rno, name, city)
Subject (subno, subname, teachername)
Relation between Customer and product is Many to Many with descriptive attribute mark.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and
Execute the following queries in PostGreSQL

```
-----------------------------------------------------------------
create table Stud1 (rno int primary key, name text, city text);
create table Sub (subno int primary key , subname text, tname text);
```

```
create table SS (rno int references Stud on delete cascade, Subno int references Sub on
delete cascade,mark int);
i)      List all students from city _______.
         select stud.* from stud where city='Pune';
ii)    Count number of subjects taught by ______.
        select count(*) from sub where tname='Satish';
iii)   Display name of all teachers who teaches subject "OS"
        select distinct(tname) from sub where subname='OS';
iv)   Delete record of a student named ______.
        delete from stud where name='Suresh';
```

---

B)Create a stored procedure named as "addrecords" to add student record. [10 Marks]

```
create or replace function addrecords(int,text,text)
returns void as'
declare
begin
insert into Stud1 (rno,name,city) values ($1,$2,$3);
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select addrecords(1,'Suresh','Pune');
```

---

24

Q.1)Consider the following database:
Book (bid, btitle, price, publication)
Author (aid, aname, mobile number,city)
Relation between Author and Book is one to Many
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

---

```
create table Author (aid int primary key, aname text, mobile numeric,city text);
create table Book (bid int primary key, btitle text , price float, pub text, aid int
references author);
```

---

i)display author names that starts with S.

```
   select aname from author where aname like 'S%';
```

ii)Display the total price of book published by "Prentice hall".

```
   select sum(price) as "Total Price" from book where pub='Prentice
   hall';
```

iii)Update mobile number of author named _____ to 9844567822

```
   update author set mobile=9844567822 where aname='Satish';
```

iv)Display details of books written by author_____.

```
   select btitle from book,author where author.aid=book.aid and
   aname='Satish';
```

---

B)Create a stored function named as "max_price" which will find maximum book price. [10 Marks]

```
create or replace function max_price()
returns void as'
declare
mprice float;
begin
select into mprice max(price) from book;
raise notice ''Maximum Price is : %  '',mprice;
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select max_price();
NOTICE:  Maximum Price is : 1000
```

---

25

Q.1)Consider the following database:
Professor (prof_no, prof_name, designation, salary)
Department (dno, dname, location)
The relationship is as follows: Department-Professor: one to many.
A)Create above database in PostgreSQL and insert sufficient records.[10 Marks] and Execute the following queries in PostGreSQL

---

```
create table Depart (dno int primary key, dname text, loc text);
create table Professor (pno int primary key, pname text, desig text, salary float, dno int references depart);
```

i)Display average salary of professor.

```
select avg(salary) from professor;
```

ii)List the details of all the departments located at _ .

```
select * from depart where loc='Pimpri';
```

iii)Display the details of professors whose names ends with an alphabet “r”.

```
select * from professor where pname like '%r';
```

iv)Display details of all professors working in “Computer” department.

```
select professor.* from professor,depart where professor.dno=depart.dno and dname='Computer';
```

---

B)Create a stored procedure named as “display_message” which will display the message “Welcome to RDBMS world!!!!.” [10 Marks]

```
create or replace function display_message()
returns void as'
declare
begin
raise notice '' Welcome to RDBMS world!!!!.'';
end;'
language 'plpgsql';
```

---

Execution:

```
Test=# select display_message();
NOTICE:   Welcome to RDBMS world!!!!.
```

---