FUNCTION

SET A

Q.1] Write a function to print total number of persons of a particular area. (Accept area_name as input parameter.) Display appropriate message.

```
sybca29=> create or replace function Omkar(n area.aname%type)
returns int as'
declare
ans int;
begin
select count(person.pno) into ans from area,person where area.aname=person.aname and
area.aname=n;
return ans;
end;
'language 'plpgsql';
NOTICE: type reference area.aname%TYPE converted to character varying
CREATE FUNCTION
************************************OUTPUT*********************************
sybca29=> select Omkar('pune');
 Omkar
--------
     1
(1 row)
*****************************************************************************
```

Q.2] Write a stored function to update a income of all persons living in 'urban area' by 25%.

```
sybca28=> create or replace function kalyan()
returns void as'
declare
begin
update person set income=income+(income*0.25) from area where area.aname=person.aname
AND area.a_type=''urban'';
end;
'language 'plpgsql';
CREATE FUNCTION
************************************OUTPUT*********************************
sybca28=> select kalyan();
 kalyan
--------
(1 row)
```

SET B

Q. 1] Write a stored function to count the number of teachers who are teaching to a given student.(Accept student name as input parameter)

```
sybca24=> create or replace function st(s std.s_name%type)
sybca24-> returns int as'
sybca24'> declare
sybca24'> ans int;
sybca24'> begin
sybca24'> select count(teach.t_no)into ans from teach,std,std_teach
sybca24'> where std.s_no=std_teach.s_no AND teach.t_no=std_teach.t_no AND std.s_name=s;
sybca24'> return ans;
sybca24'> end;
sybca24'> 'language 'plpgsql';
NOTICE: type reference std.s_name%TYPE converted to character varying
CREATE FUNCTION
**********************************OUTPUT*******************************
sybca24=> select st('rutu');
 st
----
  1
(1 row)
************************************************************************
```

Q. 2] Write a stored function to find the details of minimum experienced teacher.

```
sybca30=>create or replace function std_teach()
sybca30->returns int as'
sybca30'> DECLARE
sybca30'> ans int;
sybca30'> BEGIN
sybca30'>
sybca30'>
sybca30'>select tno,tname,qualification,experience into ans from teach where experience=(select
min(experience)from teach);
sybca30'>raise notice"the teacher details is ",total;
sybca30'>return ans;
sybca30'> END;
sybca30'>      'language
'plpgsql'; CREATE FUNCTION
**********************************OUTPUT*******************************
sybca30=> select std_teach();
NOTICE the teacher deatails is(1,mr.kulkarni,mba,5)
-------

(1 row)
```

****

SET C

Q. 1] Write a stored function to accept date and passenger name and display no.of berths reserved and ticket amount paid by him/her.

```
sybca22=> create or replace function rr(d ticket83.tdate%type,n passenger83.p_name%type)
returns record as'
sybca22'> declare
sybca22'> ans record;
sybca22'> begin
sybca22'> select no_of_berths,ticket_amt into ans from ticket83,passenger83 where
passenger83.p_id=ticket83.p_id AND tdate=d AND p_name=n;
sybca22'> return ans;
sybca22'> end;
sybca22'> 'language'plpgsql';
NOTICE: type reference ticket83.tdate%TYPE converted to date
NOTICE: type reference passenger83.p_name%TYPE converted to character varying
CREATE FUNCTION
***********************************OUTPUT*********************************
sybca22=> select rr('11oct2017','rutu');
   rr
-----------
 (5,50.00)
(1 row)
***************************************************************************
```

Q. 2] Write a stored function to update the status of the ticket from 'waiting' to 'confirmed' for passenger named 'Mr.nene'.

```
sybca16=> create or replace function ur() returns void as'
sybca16'> declare
sybca16'> begin
sybca16'> update ticket set status=''waiting'' from pas where pas.pid=ticket.pid and
status=''confirmed'' and pname=''mr.nene'';
sybca16'> end;
sybca16'> 'language'plpgsql';
CREATE FUNCTION

***********************************OUTPUT*********************************
sybca16=> select ur();
 ur
----
(1 row)

***************************************************************************
```

SET D

Q.1] Write a stored function to find the number of employees whose joining date is before '2019-12-02'.

```
create or replace function pro(e employee.eno%type)
returns int as'
declare
ans int;
begin
select count(project.pno) into ans from project,employee,pro_emp where
employee.eno=pro_emp.eno AND project.pno=pro_emp.pno AND employee.eno=e;
return ans;
end;'
language 'plpgsql';
CREATE FUNCTION
```

************************************OUTPUT**********************************

```
sybca21=> select pro(321);
 pro
-----
   1
(1 row)
```

****

Q.2] Write a stored function to accept eno as input parameter and count number of project on which that employee is working.

```
create or replace function proj(e emp.eno%type)
 returns int as'
 declare
 ans int;
 begin
 select count(proj.pno)into ans from proj,emp,ep where proj.pno=ep.pno AND emp.eno=ep.eno
AND emp.eno=e;
 return ans;
 end;
 'language 'plpgsql';
 NOTICE: type reference emp.eno%TYPE converted to integer
  CREATE FUNCTION
```

************************************OUTPUT**********************************

```
  sybca20=> select proj(11);
   proj
  ------
    1
 (1 row)
```

****

SET E

Q.1] Write a stored function to accept a srno and years as input parameter and returns the total no of prizes won by that student in that year.

```
sybca17=> create or replace function sete1(a student1.sregno%type,b sc.year%type)
sybca17-> returns int as'
sybca17'> declare
sybca17'> ans int;
sybca17'> begin
sybca17'> select count(rank)into ans from student1,sc where student1.sregno=sc.sregno and
student1.sregno=a and year=b;
sybca17'> return ans;
sybca17'> end;
sybca17'> 'language 'plpgsql';
NOTICE: type reference student1.sregno%TYPE converted to integer
NOTICE: type reference sc.year%TYPE converted to integer
CREATE FUNCTION
***************************************OUTPUT********************************
sybca17=> select sete1('11','2017');
 sete1
-------
     1
*****************************************************************************
```

***** Q2]. Write a stored function to find the name of student who has participated in maximum number of competitions in the year 2015.**

```
sybca25=> create or replace function sete()
returns record as'
declare
name record;
begin
select sname,count(*) into name from std,comp,std_comp where std.sregno=std_comp.sregno
AND comp.cno=std_comp.cno AND year="2015" group by sname having count(*)>1;
return name;
end;
'language 'plpgsql';
CREATE FUNCTION
***************************************OUTPUT********************************
sybca25=> select sete();
       sete
--------------------
 ("nikhil kadam",3)
(1 row)
```

****