****

Name: **Class:**

Assignment No.:4 **Batch:**

Ass.Name:Stored Function **Date:**

****

SET A Database Name :- Bank database

```
bank=# create table branch
bank-# (bid integer primary key,
bank(# brname char(30),
bank(# brcity char (10));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"branch_pkey" for table "branch"
CREATE TABLE

bank=# insert into branch values(101,'Aundh','pune');
INSERT 0 1
bank=# insert into branch values(102,'Deccan','pune');
INSERT 0 1
bank=# insert into branch values(103,'M.G. road','pune');
INSERT 0 1
bank=# insert into branch values(104,'Sadashiv Peth','pune');
INSERT 0 1

bank=# select * from branch;
 bid |             brname              |   brcity
-----+---------------------------------+------------
 101 | Aundh                           | pune
 102 | Deccan                          | pune
 103 | M.G. road                       | mumbai
 104 | Sadashiv Peth                   | pune
(4 rows)


bank=# create table customer
bank-# (cno integer primary key,
bank(# cname char(20),
bank(# caddr char(35),
bank(# city char(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"customer_pkey" for table "customer"
CREATE TABLE

bank=# insert into customer
values(201,'Vishal','pimpri','pune');
INSERT 0 1
bank=# insert into customer
values(202,'Vikas','kalyan','mumbai');
INSERT 0 1
```

```
bank=# insert into customer
values(203,'Amar','Dadar','mumbai');
INSERT 0 1
bank=# insert into customer
values(204,'Ashish','nigdi','pune');
INSERT 0 1

bank=# select * from customer;
 cno |        cname         |
-----+----------------------+--
 201 | Vishal               |
 202 | Vikas                |
 203 | Amar                 |
 204 | Ashish               |
(4 rows)
 |                caddr                |         city
+-------------------------------------+----------------------
 | pimpri                              | pune
 | kalyan                              | mumbai
 | Dadar                               | mumbai
 | nigdi                               | pune
(4 rows)


bank=# create table loan_application
bank-# (lno integer primary key,
bank(# lamtrequired integer,
bank(# lamtapproved integer,
bank(# l_date date);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"loan_application_pkey" for table "loan_application"
CREATE TABLE

bank=# insert into loan_application values
(301,500000,250000,'07/22/2013');
INSERT 0 1
bank=# insert into loan_application values
(302,30000,8000,'06/16/2014');
INSERT 0 1
bank=# insert into loan_application values
(303,400000,400000,'07/22/2014');
INSERT 0 1
bank=# insert into loan_application values
(304,50000,45000,'09/12/2013');
INSERT 0 1
bank=# insert into loan_application values
(305,60000,60000,'08/12/2013');
INSERT 0 1
bank=# insert into loan_application values
(306,500000,500000,'09/22/2013');
INSERT 0 1
```

```
bank=# insert into loan_application values
(307,500000,450000,'09/25/2013');
INSERT 0 1
bank=# select * from loan_application;
 lno | lamtrequired | lamtapproved |   l_date
-----+--------------+--------------+------------
 301 |       500000 |       250000 | 2013-07-22
 302 |        30000 |         8000 | 2014-06-16
 303 |       400000 |       400000 | 2014-07-22
 304 |        50000 |        45000 | 2013-09-12
 305 |        60000 |        60000 | 2013-08-12
 306 |       500000 |       500000 | 2013-09-22
 307 |       500000 |       450000 | 2013-09-25
(7 rows)


bank=# create table bcl
bank-# (bno integer references branch(bid),
bank(#  cno integer references customer(cno),
bank(#  lno integer references loan_application(lno));
CREATE TABLE

bank=# insert into bcl values (101,201,301);
INSERT 0 1
bank=# insert into bcl values (101,203,302);
INSERT 0 1
bank=# insert into bcl values (102,202,303);
INSERT 0 1
bank=# insert into bcl values (103,201,304);
INSERT 0 1
bank=# insert into bcl values (102,204,305);
INSERT 0 1
bank=# insert into bcl values (103,202,306);
INSERT 0 1
bank=# insert into bcl values (104,202,307);
INSERT 0 1

bank=# select * from bcl;
 bno | cno | lno
-----+-----+-----
 101 | 201 | 301
 101 | 203 | 302
 102 | 202 | 303
 103 | 201 | 304
 102 | 204 | 305
 103 | 202 | 306
 104 | 202 | 307
(7 rows)

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

Functions

a) Find the total number of customers in a particular branch. Take branch name as input from user.

```
bank=# create or replace function tot_cust(text)
bank-# returns integer as'
bank'# declare
bank'# bname alias for $1;
bank'# cnt integer;
bank'# begin
bank'#   select into cnt count(*) from bcl,branch
       where branch.bid=bcl.bno and brname=bname;
bank'#   return cnt;
bank'# end;'
bank-# language 'plpgsql';
CREATE FUNCTION
O/P:-
bank=# select tot_cust ('Deccan');
 tot_cust
----------
        2
(1 row)
```

b)Write a stored function to find maximum loan amount approved.

```
bank=# create or replace function max_lamt()
bank-# returns int as'
bank'# declare
bank'#   maxlamt integer;
bank'# begin
bank'#   select into maxlamt max(lamtapproved)from
loan_application;
bank'#   return maxlamt;
bank'# end;'
bank-# language 'plpgsql';
CREATE FUNCTION

O/P :-
bank=# select max_lamt();
 max_lamt
----------
   500000
(1 rows)
```

SET B Database Name:- Project-Employee database

```
project=# create table PROJECT
project-# (PNO INTEGER primary key,
project(# P_NAME CHAR(30),
project(# PTYPE CHAR(20),
project(# DURATION INTEGER);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"project_pkey" for table "project"
CREATE TABLE

project=# insert into project values (101,
'banking','application',3);
INSERT 0 1
project=# insert into project values (102,
'robotics','system',7);
INSERT 0 1
project=# insert into project values (103,
'medical','application',2);
INSERT 0 1

project=# select * from project;
 pno |             p_name      |        ptype         |
duration
-----+---------------------------------+----------------------+--
-------
 101 | banking                 | application          |
3
 102 | Robotics                | system               |
7
 103 | medical                 | application          |
2
(3 rows)


project=# create table EMPLOYEE
project-# (ENO INTEGER primary key,
project(#  E_NAME CHAR (20),
project(# QUALIFICATION CHAR (15),
project(# JOINDATE DATE);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"employee_pkey" for table "employee"
CREATE TABLE

project=# insert into employee
values(2000,'Amar','BE','2/10/2010');
INSERT 0 1
project=# insert into employee
values(2001,'Sandeep','MCS','4/01/2012');
INSERT 0 1
```

```
project=# insert into employee
values(2002,'Ajay','MCA','1/22/2010');
INSERT 0 1

project=# select * from employee;
 eno  |        e_name        |  qualification  |  joindate
------+----------------------+-----------------+-----------
 2000 | Amar                 | BE              | 2010-02-10
 2001 | Sandeep              | MCS             | 2012-04-01
 2002 | Ajay                 | MCA             | 2010-01-22
(3 rows)


project=# create table proj_emp
project-# (pno integer references project(pno),
project(#  eno integer references employee (eno),
project(#  start_date date,
project(#  no_of_hours_worked integer);
CREATE TABLE

project=# insert into proj_emp
values(101,2000,'05/30/2012',20);
INSERT 0 1
project=# insert into proj_emp
values(102,2002,'04/01/0201',50);
INSERT 0 1
project=# insert into proj_emp
values(102,2000,'04/01/2013',30);
INSERT 0 1
project=# insert into proj_emp
values(103,2001,'07/16/2012',25);
INSERT 0 1

project=# select * from proj_emp;
 pno | eno  | start_date | no_of_hours_worked
-----+-----+-----------+-------------------
 101 | 2000 | 2012-05-30 |                 20
 102 | 2002 | 0201-04-01 |                 50
 102 | 2000 | 2013-04-01 |                 30
 103 | 2001 | 2012-07-16 |                 25
 (4 rows)
-----------------------------------------------------------------
```

Functions

a)Write a function to count number of employees working on specific project.

```
project=# create or replace function no_of_emp(text)
project-# returns integer as'
project'# declare
project'#   pname alias for $1;
project'#   cnt integer;
project'# begin
project'#   select into cnt count(*) from proj_emp,project
 where proj_emp.pno=project.pno and p_name=pname;
project'#   return cnt;
project'# end;'
project-# language 'plpgsql';
CREATE FUNCTION
```

O/P:-

```
project=# select no_of_emp('Robotics');
 no_of_emp
-----------
         2
(1 row)
```

b)

```
project=# create or replace function emp_join()
project-# returns integer as'
project'# declare
project'#   cnt integer;
project'# begin
project'#   select into cnt count(*) from employee
where joindate < ''03/10/2010'';
project'#   return cnt;
project'# end;'
project-# language 'plpgsql';
CREATE FUNCTION
```

O/P:-

```
project=# select emp_join();
 emp_join
----------
        2
(1 row)
```

SET C Database Name:- Business Trip database

```
business=# create table dept
business-# ( deptno varchar(10) primary key,
business(# dept_name char(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"dept_pkey" for table "dept"
```

```
CREATE TABLE

business=# insert into dept values (10,'computer');
INSERT 0 1

business=# select * from dept;
 deptno |      dept_name
--------+---------------------
 10     | computer
 20     | western
 30     | maths
(3 rows)

business=# create table salesman
business-# (sno integer primary key,
business(# s_name char(30),
business(# start_year integer,
business(# deptno varchar(10) references dept(deptno));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"salesman_pkey" for table "salesman"
CREATE TABLE

business=# insert into salesman values(101,'Amit',2008,10);
INSERT 0 1

business=# select *from salesman;
 sno |             s_name             | start_year | deptno
-----+--------------------------------+------------+-------
 101 | Amit                           |       2008 | 10
 102 | Vishal                         |       2010 | 20
 103 | Ajay                           |       2010 | 10
 104 | Mr. Patil                      |       2008 | 30
 105 | Raju                           |       2005 | 20
(5 rows)

business=# create table trip
business-# (tno integer primary key,
business(# from_city char(20),
business(# to_city char(20),
business(# dparture_date date,
business(# return date,
business(# sno integer references salesman(sno));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"trip_pkey" for table "trip"
CREATE TABLE

business=# insert into trip values
(1,'pune','mumbai','07/22/2008','07/25/2008',102);
INSERT 0 1

business=# select * from trip;
```

```
 tno |      from_city       |       to_city        |
dparture_date |   return   | sno
-----+--------------------+--------------------+-----------
--+-----------+----
   1 | pune                 | mumbai               |
2008-07-22    | 2008-07-25 | 102
   2 | pune                 | banglore             |
2012-09-12    | 2012-09-14 | 101
   3 | nashik               | calcutta             |
2014-06-25    | 2014-06-29 | 102
   4 | pune                 | calcutta             |
2014-07-15    | 2014-07-20 | 103
   5 | pune                 | mumbai               |
2014-07-15    | 2014-07-20 | 104
   6 | pune                 | Nashik               |
2014-08-12    | 2014-08-15 | 105
(6 rows)

business=# create table expense
business-# (eid integer primary key,
business(# amount money);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
"expense_pkey" for table "expense"
CREATE TABLE

business=# insert into expense values (1,'$12000');
INSERT 0 1

business=# select * from expense;
 eid |   amount
-----+-----------
   1 | $12,000.00
   2 | $20,000.00
   3 |  $9,000.00
   4 | $14,000.00
   5 | $16,000.00
   6 | $15,000.00
(6 rows)
```

Functions

a)

```
business=# create or replace function trip_no()
business-# returns integer as'
business'# declare
business'#    tripno integer;
business'# begin
business'#   select  into tripno tno from trip ,expense where
tno=eid and amount=(select max(amount) from expense);
business'#    return tripno;
```

```
business'# end;'
business-# language 'plpgsql';
CREATE FUNCTION

O/P:-
business=# select trip_no();
 trip_no
---------
       2
(1 row)
```

b)

```
business=# create or replace function total_trips()
business-# returns integer as'
business'# declare
business'#     tot_trip integer;
business'# begin
business'#   select into tot_trip count(*) from trip where
from_city=''pune'' and to_city=''mumbai'';
business'#    return tot_trip;
business'# end;'
business-# language 'plpgsql';
CREATE FUNCTION

O/P:-
business=# select total_trips();
 total_trips
-------------
           2
(1 row)
```