Assignment 3 : Stored Functions

Set A

Project-Employee Database

Consider the following Entities and their Relationships for Project-Employee database.

Project `(pno integer, pname char (30), ptype char (20), duration integer)`
Employee `(eno integer, ename char (20), qualification char (15), joining_date date)`

Relationship between Project and Employee is many to many with descriptive attribute
`start_date` date, `no_of_hours_worked` integer.
Constraints: Primary Key, `duration` should be greater than zero, `pname` should not be null.

1. Write a stored function to find the number of employees whose joining date is before '01/01/2007'.

```
Create or Replace FUNCTION setA1 () Returns Int AS'
Declare
cnt int;
BEGIN
select into cnt count(eno)from project_employee where start_date < ''2007-01-01'';
return cnt;
END;
'LANGUAGE 'plpgsql';

postgres=# select setA1();
 seta1
--------
      2
(1 row)
```

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

```
Create or replace FUNCTION SetA2(int) Returns Int AS'
Declare
cnt int;
BEGIN
select into cnt count(eno) from project_employee where eno=$1;
return cnt;
END;
'LANGUAGE 'plpgsql';

postgres=# select setA2(2);
 seta2
--------
      2
(1 row)
```

3. Write a stored function to accept project name and display employee details who worked more than 2000 hours.

```
Create or replace FUNCTION SetA3(varchar) Returns void AS'
Declare
rec record;
BEGIN
for rec in select eno, ename, qualification, joining_date from employee where eno IN
 (select eno from project_employee where no_of_hours_worked > 2000 and pno=(select pn
o from project where pname=$1)) loop
  raise notice ''%  %  %  %'', rec.eno, rec.ename, rec.qualification, rec.joining_dat
e;
End loop;
END;
'LANGUAGE 'plpgsql';

postgres=# select setA3('Microsoft');
NOTICE:  2  Shoeb  BCA  2021-09-15
 seta3
--------

(1 row)
```

4. Write a stored function to display all projects started after date “01/01/2019”.

```
CREATE OR REPLACE FUNCTION SetA4() RETURNS
record AS'
DECLARE
rec record;
BEGIN
select into rec pname from Project where pno IN (select pno from Project_Employee whe
re CAST(start_date as Char(10)) > ''2019-01-01'');
RETURN rec;
END;
'LANGUAGE 'plpgsql';

postgres=# select setA4();
    seta4
--------------
 (Microsoft)
(1 row)
```

Set B

Bus Transport Database

Consider the following Entities and their Relationships for Bus Transport database.

Bus `(bus_no int ,b_capacity int , depot_name varchar(20))`
Route `(route_no int, source char (20), destination char (20), no_of_stations int)`
Driver `(driver_no int ,driver_name char(20), license_no int, address char(20), d_age int, salary float)`

Relationship between Bus and Route is many to one and relationship between Bus and Driver is many to many with descriptive attributes `date_of_duty_allotted` and shift. Constraints: Primary Key, `license_no` is unique, `b_capacity` should not be null, shift can be *Morning* or *Evening*.

1. Write a stored function to accept route no and display bus information running on that route.

```
CREATE OR REPLACE FUNCTION SetB1 (int) RETURNS
int AS'
DECLARE
rec record;
cnt Int:=10;
BEGIN
raise notice ''bus_no || b_capacity || depot_name || route_no'';
for rec in select bus_no, b_capacity, depot_name, route_no from Bus where route_no=$1
loop
  raise notice ''%        %            %           %'', rec.bus_no, rec.b_capacity, r
ec.depot_name, rec.route_no;
end loop;
RETURN null;
END;
'LANGUAGE 'plpgsql';

select setB1(1);
NOTICE:  bus_no || b_capacity || depot_name || route_no
NOTICE:  10        25           Kothrud           1
NOTICE:  25        30           Deccan           1
 setb1
-------

(1 row)
```

2. Write a stored function to accept shift and depot name and display driver details who having duty allocated after ‘01/07/2020’.

```
CREATE OR REPLACE FUNCTION SetB2 (varchar, varchar) RETURNS
int AS'
DECLARE
rec record;
BEGIN
raise notice ''driver_no | driver_name | license_no | address | d_age | salary'';
for rec in select * from Driver where driver_no IN (select driver_no from bus_driver
 where shift=$1 and date_of_duty_allotted > ''2020-07-01'' and bus_no IN(select bus_n
o from Bus where depot_name=$2)) loop
  raise notice ''%           %      %       %     %      %'', rec.driver_no, rec.driv
er_name, rec.license_no, rec.address, rec.d_age, rec.salary;
end loop;
RETURN null;
END;
'LANGUAGE 'plpgsql';


postgres=# select setB2('Evening','Kothrud');
NOTICE:  driver_no | driver_name | license_no | address | d_age | salary
NOTICE:  9           Gurpreet      452574       Patna     42      23000
 setb2
--------

(1 row)
```

3. Write a stored function to accept source name and display count of buses running from source place.

```
CREATE OR REPLACE FUNCTION SetB3 (varchar) RETURNS
INT AS'
DECLARE
cnt INT;
BEGIN
select into cnt count(bus_no) from BUS where route_no IN(select route_no from ROUTE w
here source=$1);
RETURN cnt;
```

```
END;
'LANGUAGE 'plpgsql';

postgres=# select setB3('Deccan');
 setb3
--------
      2
(1 row)
```

4. Write a stored function to accept depot name and display driver details having age more than 50.

```
CREATE OR REPLACE FUNCTION SetB4 (varchar) RETURNS void AS'
DECLARE
rec record;
BEGIN
for rec in select * from Driver where driver_no IN (select driver_no from BUS_driver
 where bus_no IN (select bus_no from Bus where depot_name=$1)) and d_age > 50 loop
  raise notice ''% % % % % %'', rec.driver_no, rec.driver_name, rec.license_no, rec.a
ddress, rec.d_age, rec.salary;
END loop;
END;
'LANGUAGE 'plpgsql';

postgres=# select SetB4('Kothrud');
NOTICE:  7 Partha 451674 Kothrud 52 25000
 setb4
--------

(1 row)
```