STORED PROCEDURE:

```
set search_path to jobrecruitmentsystem;

create or replace function Vacancies(Pos varchar(50)) returns table(place varchar(50),vacancies int)
As $body$

begin

  return query select BranchId,vacancy from locatedat where locatedat.p_name=pos;

end;

$body$

language plpgsql;


select * from Vacancies('CA');
```

//

```
create or replace function ExpertIn(expi varchar(50)) returns table(C_Number varchar(20)) As
$body$

begin

  return query select C_No  from candidateexpertisedarea where
candidateexpertisedarea.expertisedarea=expi;

end;

$body$

language plpgsql;
```

```
//////////////////////////////////////////////////////////

set search_Path to jobrecruitingsystem;

create or replace function InterviewsTakenBy(inv varchar(20)) returns table(Candidate_No
varchar(20),Pos varchar(50), Round int) As $body$

begin

  return query select C_No,P_Name,Stage from review where interviewer_No=inv;

end;

$body$

language plpgsql;

//////////////////////////////////////////////////////////////////////

set search_path to jobrecruitmentsystem;

create or replace function knowledgeOf(tech varchar(100)) returns table(Candidate_No
varchar(20),Pos_Name varchar(50)) As $body$

begin

  --return query select BranchId,vacancy from locatedat where locatedat.p_name=pos;

  return query select Distinct C_No,P_Name from TechnicalSkill1 natural join new_job_Application;

end;

$body$

language plpgsql;

select * from knowledgeOf('ubuntu');

//////////////////////////////////////////////////////////////////////
```

```
set search_path to jobrecruitmentsystem;

create or replace function QualifiedOfDegree(dig varchar(20)) returns table(percentage bigint) As
$body$

begin


  --return Query select count(Distinct c_no) as k1 from new_job_application natural join Education
where degree=dig;

   return query  select ((select count(C_No) from new_job_Application natural join Education where
degree=dig

 and status='Qualified')*100)/(select count(c_No) as x from Education where Degree=dig)

 as t;

end;

$body$

language plpgsql;


select QualifiedOfDegree('B.Tech');



////////////////////////////////////////////////////////////


set search_Path to jobrecruitmentsystem;

create or replace function HighestAppliedPosition() returns table(Pos

varchar(50),Coun bigint) As $body$

begin

 return query select p_name, max(cot) from

(select P_Name, Count(c_no) as cot from new_job_application group by P_name ) as s GROUP BY
s.p_name order by max(cot) desc limit 1;

end;

$body$

language plpgsql;

select * from HighestAppliedPosition();
```

//

```
set search_Path to jobrecruitmentsystem;

create or replace function SelectedAll() returns table(City_Name varchar(50),Position_Name
varchar(50),No_Of_Candidate bigint) As $body$

begin

  return query select city,p_n ,n_of_candidate from Location natural join

        (select branchid,p_name as p_n,count(c_no) as n_of_candidate  from candidate natural join

              (select c_no,p_name from new_job_application where status='Qualified') as y group
by branchid,p_name ) as x;

end;

$body$

language plpgsql;

select * from SelectedAll();
```