

Triggers:

set search_path to jobrecruitmentsystem;

create or replace function ChangeStage() returns trigger AS \$body\$

declare

x1 varchar(5);

Begin

if new.rating>6 then

if new.stage<3 then

update new_job_application set stage=new.stage+1 , status='pending' where

app_no=new.app_no and c_no=new.c_no and p_name=new.p_name;

elseif new.stage=3 then

update new_job_application set status='Qualified' where app_no=new.app_no

and c_no=new.c_no and p_name=new.p_name;

Select BranchId into x1 from candidate where C_No=new.C_No;

update LocatedAt set vacancy=vacancy-1 where BranchId=x1 and

P_Name=new.P_Name;

end if;

else

update new_job_Application set status='DisQualified' where app_no=new.app_no and

c_no=new.c_no and p_name=new.p_name;

end if;

return new;

End

\$body\$ language plpgsql;

CREATE TRIGGER SC

after insert

ON review

FOR EACH ROW

EXECUTE PROCEDURE ChangeStage();

Trigger 2:

```
set search_Path to jobrecruitmentsystem;
```

```
--select * from new_job_application;
```

```
create or replace function History() returns trigger AS $body$
```

```
declare
```

```
    x date;
```

```
    y int;
```

```
    y1 varchar(5);
```

```
    y2 int;
```

```
    t1 varchar(50);
```

```
Begin
```

```
    select Unique_Identity into t1 from candidate where candidate.C_no=new.C_no;
```

```
    if t1=null then
```

```
        return new;
```

```
    else
```

```
        select AppliedDate into x from New_Job_Application where  
New_Job_Application.C_no=t1;
```

```
    select BranchId into y1 from candidate where C_No=t1;
```

```
        select vacancy into y2 from locatedat where BranchId=y1;
```

```
    if x=null then
```

```
        x=current_date;
```

```
    end if;
```

```
    y=current_date-x;
```

```
    if y> 365 then
```

```
        if y2=0 then
```

```
            raise notice 'There Is No Vacancy For This Position!';
```

```
else
    update new_Job_Application set NextInterviewDate=Current_date+ integer '3';
end if;
--return new ;
else
    delete from candidate where C_no=new.C_no;
delete from new_job_application where C_no=new.C_no;
    raise notice 'You Are Not Allowed to Apply!';
--return new;
end if;
    return new;
end if;

End
$body$ language plpgsql;
```

```
CREATE TRIGGER CheckDuplicate
after insert
ON new_job_application
FOR EACH ROW
EXECUTE PROCEDURE History()
```