

SCHEMA for CDC Database

Things to be deleted

Super_user(ID)

1. *Login(Username, Password)*

2. *Student(Roll_no, name, department, gender, semester, year, email, phone_no, address) # data added*

3. *Company(Name, Company_id, url, email, address, phone_no) # data added*

4. *Contact_person(Company_id, name, phone_no, email) # data added*

5. *Comapny_req2(job_id, **Company_id**, year, department, cutoff_CGPA) # data added*

6 . *job(job_id, Title, Intro, Description, Vacancies, Skill_req, Salary, Procedur, start, end, last_date, job_type(I or P)) # data added*

8. *sec_edu(Roll_no, 10th, 12th) # data added*

9. *GPA(Roll_no, Sem1, Sem2, Sem3, Sem4, Sem5, Sem6, Sem7, Sem8, CGPA, backlog) # data added*

10. *cv(Roll_no, Courses&Electives, Work_Exp, Project_Details, Skills, Interests, POR, Achievements, Extracurr) # data added*

11. *app_stu(Roll_no, job_id)*

12. *select_stu (Roll_no, job_id)*

13. *verify_check(Roll_no, verified)("yes"/"no")*

Users -

Student - 8 (username and password are same)

Stud_rep - 1

TPO - 1

Company_contact_person - 1

Superuser - root

101601010

101701031

111601010

111701030

121601010

121701030

131601010

131701010

c8

c12

c7

c6

c10

c5

c9

c11

sr1

USER

Student

101701031

131701010

ROLES

1. stud_role

Select,insert,update ,delete on stud_view (*Seeing his/her own details*)

Select on stud_com_view (*Seeing the company eligible for him/her*)

select on apply_comp_view (*Finding the company which she/he applied*)

insert on app_stu (*Student applying a company*) (via procedure apply_com)

View

a) stud_view

create view stud_view as select * from Student natural join sec_edu natural join GPA natural join cv where (select user()) = concat(Student.Roll_no,'@localhost');
Grant option are select , insert , update , delete on view

d) verify_view

create view verify_view as select * from verify_check where (select user()) = concat(Roll_no,'@localhost');

Grant option are select

b)stud_com_view(To see available opportunity)

create view stud_com_view as with info(j,c) as (select C.job_id,C.Company_id from Company_req2 as C,stud_view as S where S.year = C.year and S.department = C.department and S.CGPA >= C.cutoff.CGPA) select Name,Company_id,url,email,address,phone_no,job_id,title,Intro,Description,Vacancies,skill_req ,salary,Procedur,start,end,last_date,job_type from job,Company,info where info.j = job_id and info.c = Company_id

c)apply_comp_view(To see in what all companies he has applied)

create view apply_comp_view as (select * from job where job_id in (select job_id from app_stu where (select user()) = concat(Student.Roll_no,'@localhost'))

e)

Create view browse as (select * from stud_com_view as s where s.job_id not exists in (select t.job_id from apply_com_view as t))

2. studrep_role

grant select on stud_detail_view

Grant option are select , update on verify_check

- Stud_detail_view

create view stud_detail_view as(select * from Student natural join GPA natural join cv natural join sec_edu where Roll_no in(select Roll_no from verify_check where verified = 'no'));

3.comp_role

```

grant select,update,insert,delete on com_view
grant select,update,insert,delete on com_job_view
grant select on app_stu_com_view
grant select on select_stu_com_view
grant select,insert on select_stu
/*****not good*****/
Grant select on Student
Grant select on cv
Grant select on GPA
Grant select on sec_edu
/*****/
/////insert on select_stu

```

- **Com_view**

```

Create view com_view as
select (*)
from Company,Contact_person
Where (select user()) = concat(Company.Company_id,'@localhost') and
Contact_person.Company_id = Company.Company_id

```

- **Com_job_view**

```

Create view com_job_view as
Select (*)
from company_req2 natural join job
Where (select user()) = concat(Company.Company_id,'@localhost')

```

- **app_stu_com_view**

```

create view app_stu_com_view as
with j(id) as (select job_id from Company_req2 where(select user()) =
concat(Company_req2.Company_id,'@localhost')) select Roll_no,id as job_id from
Student,j where Roll_no in (select Roll_no from app_stu,j where job_id=id);

```

- **select_stu_com_view**

```

create view select_stu_com_view as
with j(id) as (select job_id from Company_req2 where(select user()) =
concat(Company_req2.Company_id,'@localhost')) select

```

```
Roll_no,name,department,semester,id as job_id from Student,j where Roll_no in (select Roll_no from select_stu,j where job_id=id);
```

tpo_role

All privileges are there(Discuss with others)

PROCEDURE

1.Student applying the company(verified)

Procedure name : apply_com

Input : Job id

```
create procedure apply_com(in job_id varchar(10))
begin
    insert into app_stu values( (select substring( (select user() ) ,1,9 ) ) ,job_id);
end!
```

2.Company selecting the applied student(verified)

Procedure name : select_stu

Input : Student Roll_no , Job id

```
create procedure sel_stu(in Roll_no varchar(10),in job_id varchar(10))
begin
    insert into select_stu values(Roll_no,job_id);
End!
```

3.Class representative can view detail of one student

Procedure : unver_stu

Input : Roll_no

```
create procedure unver_stu (in Roll_no varchar(10))
begin
```

```
select * from stud_detail_view as v where v.Roll_no = Roll_no;  
end
```

4. Class representative making the student details verified

Procedure :verify

Input : Roll_no

```
create procedure verify(in roll_no varchar(10))  
begin  
    update stud_rep_view set verified="yes" where Roll_no=roll_no;  
end!
```

5. company views details of the student

Procedure :cv_stu

Input : Roll_no

```
create procedure cv_stu(in roll varchar(10))  
Begin  
    Select * from Student natural join GPA natural join cv natural join sec_edu where  
Roll_no=roll;  
end
```

Trigger

1. Trigger to make student unverified when the student updating its details

1. updating student

create trigger make_unver_one

after update on Student

for each row

Begin

update verify_check set verified="no" where new.Roll_no = verify_check.Roll_no;

end

2. updating GPA

create trigger make_unver_two

after update on GPA

for each row

Begin

update verify_check set verified="no" where new.Roll_no = verify_check.Roll_no;

end

3.updating sec_edu

create trigger make_unver_three

after update on sec_edu

for each row

Begin

update verify_check set verified="no" where new.Roll_no = verify_check.Roll_no;

end

4.updating cv

create trigger make_unver_four

after update on cv

for each row

Begin

update verify_check set verified="no" where new.Roll_no = verify_check.Roll_no;

end

2. Setting the role when a user is logging in (Taken Care in the front end part)