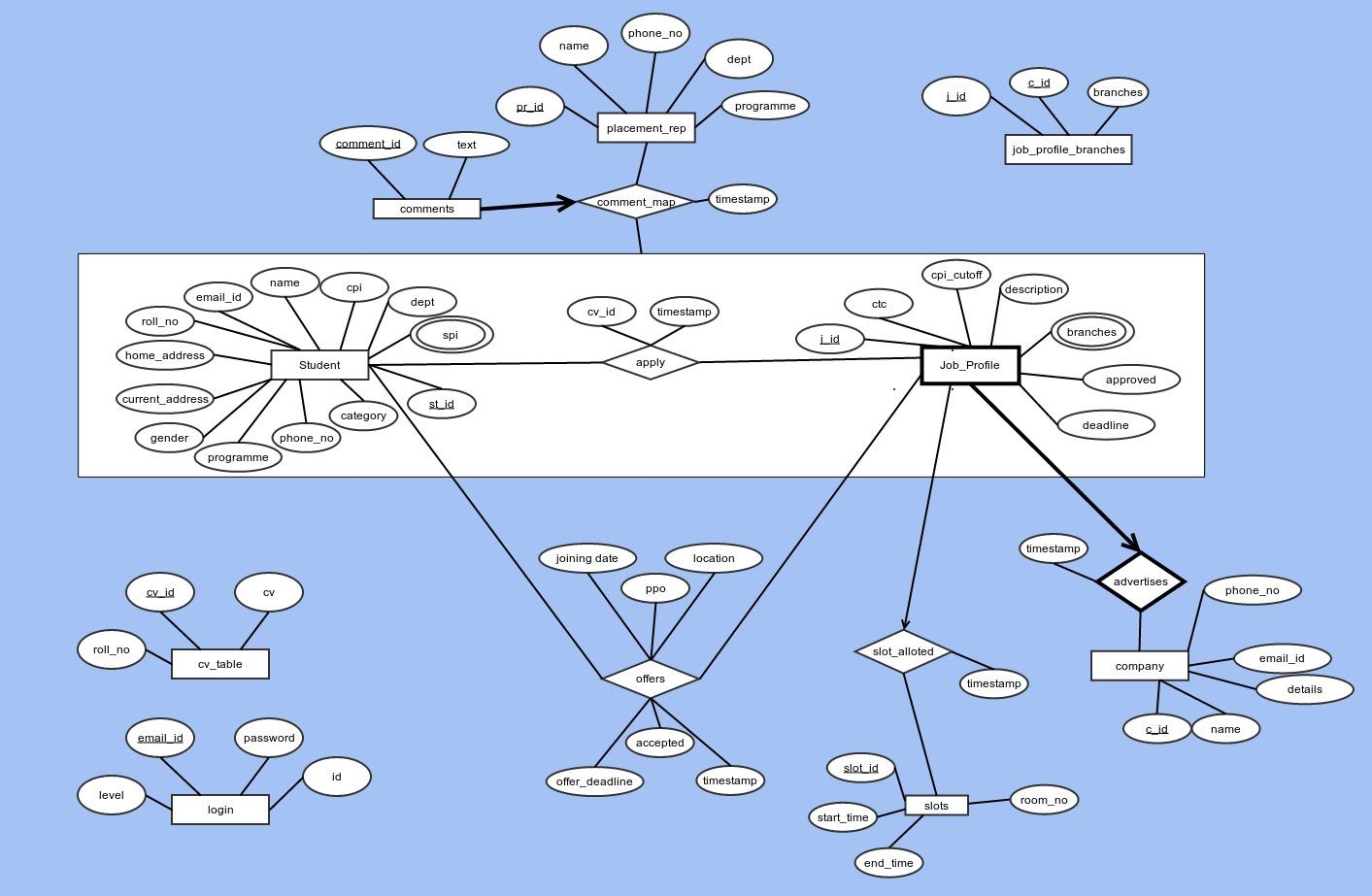
**Group 2**

(Aditya Jha - 11010102, Harshil Lodhi - 11010121, Shobhit Chaurasia - 11010179)

PLACEMENT PORTAL

ER-Diagram



### DDL for each table ….

drop table comments;

drop table offers;

drop table slot\_alloted;

drop table ppt\_slots;

drop table slots;

drop table IF EXISTS placement\_rep;

drop table IF EXISTS apply;

drop table IF EXISTS cv\_table;

drop table IF EXISTS job\_profile\_branches;

drop table IF EXISTS job\_profile;

drop table IF EXISTS company;

drop table IF EXISTS student;

drop table IF EXISTS login;

create table login(

id int unique not null auto\_increment,

email\_id varchar(50) not null,

level int not null,

password varchar(32) not null,

constraint login\_pk primary key(email\_id)

);

create table student (

roll\_no int not null unique,

st\_id int,

name varchar(50) not null,

gender enum('M','F') not null default 'F',

cpi decimal(3,2) not null,

dept enum('CSE', 'MNC', 'EEE', 'ECE', 'CL', 'CE', 'EP','BT','DOD','ME','CST','HSS') not null,

programme enum('BTECH','MTECH','PhD','MSc','BSc','BA','MA') not null,

category enum('GEN','SC','ST','OBC','PD') not null,

phone\_no bigint(10),

current\_address varchar(200) not null,

home\_address varchar(200) not null,

constraint student\_pk primary key(st\_id),

constraint student\_fk foreign key(st\_id) references login(id) on delete cascade on update cascade

);

create table company (

c\_id int not null,

name varchar(50) not null,

details varchar(1000),

phone\_no bigint(10) not null,

email\_id varchar(50) not null,

constraint company\_pk primary key(c\_id),

constraint company\_fk foreign key(c\_id) references login(id) on delete cascade on update cascade

);

create table job\_profile (

j\_id int not null auto\_increment,

ctc int not null,

cpi\_cutoff decimal(3,2) default 0,

description varchar(500),

approved enum('Y','N', 'P') default 'P',

tstamp timestamp not null default current\_timestamp on update current\_timestamp,

c\_id int not null,

deadline datetime not null,

constraint job\_profile\_pk primary key(j\_id,c\_id),

constraint job\_profile\_fk foreign key(c\_id) references company(c\_id) on delete cascade on update cascade

);

create table job\_profile\_branches (

j\_id int not null,

c\_id int not null,

dept enum('CSE', 'MNC', 'EEE', 'ECE', 'CL', 'CE', 'EP','BT','DOD','ME','CST','HSS', 'ALL'),

constraint job\_profile\_branches\_pk primary key(j\_id, c\_id, dept),

constraint job\_profile\_branches\_fk1 foreign key (j\_id) references job\_profile(j\_id) on delete cascade on update cascade,

constraint job\_profile\_branches\_fk2 foreign key (c\_id) references company(c\_id) on delete cascade on update cascade

);

create table cv\_table(

st\_id int not null,

cv\_id int auto\_increment,

cv varchar(1000),

cv\_name varchar(50) not null,

constraint cv\_table\_pk primary key(cv\_id),

constraint cv\_table\_fk foreign key (st\_id) references student(st\_id) on delete cascade on update cascade

);

create table apply(

st\_id int not null,

j\_id int not null,

c\_id int not null,

cv\_id int not null,

tstamp timestamp default current\_timestamp on update current\_timestamp,

constraint apply\_pk primary key(st\_id,j\_id, c\_id),

constraint apply\_fk1 foreign key (j\_id) references job\_profile(j\_id) on delete cascade on update cascade,

constraint apply\_fk2 foreign key (c\_id) references company(c\_id) on delete cascade on update cascade,

constraint apply\_fk3 foreign key (st\_id) references student(st\_id) on delete cascade on update cascade,

constraint apply\_fk4 foreign key (cv\_id) references cv\_table(cv\_id)

);

create table placement\_rep (

name varchar(50) not null,

phone\_no bigint(10) not null,

pr\_id int not null,

dept enum('CSE', 'MNC', 'EEE', 'ECE', 'CL', 'CE', 'EP','BT','DOD','ME','CST','HSS') not null,

programme enum('BTECH','MTECH','PhD','MSc','BSc','BA','MA') not null,

constraint placement\_rep\_pk primary key(pr\_id),

constraint placement\_rep\_fk foreign key (pr\_id) references login(id) on delete cascade on update cascade

);

create table slots (

slot\_id int not null auto\_increment,

room\_no varchar(20) not null,

start\_time datetime not null,

end\_time datetime not null,

constraint slots\_pk primary key(slot\_id)

);

create table ppt\_slots(

slot\_id int not null,

c\_id int not null unique,

constraint ppt\_slots\_pk primary key(slot\_id),

constraint ppt\_slots\_fk1 foreign key (c\_id) references company(c\_id) on delete cascade on update cascade,

constraint ppt\_slots\_fk2 foreign key (slot\_id) references slots(slot\_id) on delete cascade on update cascade

);

create table slot\_alloted(

slot\_id int not null,

c\_id int not null,

j\_id int not null,

constraint slot\_alloted\_pk primary key(slot\_id),

constraint slot\_alloted\_fk1 foreign key (c\_id) references company(c\_id) on delete cascade on update cascade,

constraint slot\_alloted\_fk2 foreign key (slot\_id) references slots(slot\_id) on delete cascade on update cascade,

constraint slot\_alloted\_fk3 foreign key (j\_id) references job\_profile(j\_id) on delete cascade on update cascade

);

create table offers(

st\_id int not null,

c\_id int not null,

j\_id int not null,

joining\_date datetime,

ppo enum('Y','N') not null,

location varchar(50),

accepted enum('Y','N') not null default 'N',

offer\_deadline datetime not null,

tstamp timestamp default current\_timestamp on update current\_timestamp,

constraint offers\_pk primary key(st\_id,j\_id, c\_id),

constraint offers\_fk1 foreign key (j\_id) references job\_profile(j\_id) on delete cascade on update cascade,

constraint offers\_fk2 foreign key (c\_id) references company(c\_id) on delete cascade on update cascade,

constraint offers\_fk3 foreign key (st\_id) references student(st\_id) on delete cascade on update cascade

);

create table comments(

comment\_id int not null auto\_increment,

txt varchar(1000) not null,

st\_id int not null,

c\_id int not null,

j\_id int not null,

pr\_id int not null,

tstamp timestamp default current\_timestamp on update current\_timestamp,

constraint comment\_map\_pk primary key(comment\_id),

constraint comment\_map\_fk1 foreign key (j\_id) references job\_profile(j\_id) on delete cascade on update cascade,

constraint comment\_map\_fk2 foreign key (c\_id) references company(c\_id) on delete cascade on update cascade,

constraint comment\_map\_fk3 foreign key (st\_id) references student(st\_id) on delete cascade on update cascade,

constraint comment\_map\_fk4 foreign key (pr\_id) references placement\_rep(pr\_id) on delete cascade on update cascade

);

## VIEWS

### -- views for job Profiles for each dept ….

create view cseJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "CSE";

create view eceJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "ECE";

create view eeeJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "EEE";

create view mncJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "MNC";

create view meJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "ME";

create view cstJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "CST";

create view clJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "CL";

create view ceJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "CE";

create view hssJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "HSS";

create view dodJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "DOD";

create view epJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "EP";

create view btJobProfiles as select j.c\_id, j.j\_id from job\_profile as j, job\_profile\_branches as jb where j.j\_id = jb.j\_id and j.c\_id = jb.c\_id and jb.dept = "BT";

### -- views for Offers of each dept ….

create view cseOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "CSE";

create view eceOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "ECE";

create view eeeOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "EEE";

create view mncOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "MNC";

create view meOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "ME";

create view cstOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "CST";

create view clOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "CL";

create view ceOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "CE";

create view hssOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "HSS";

create view dodOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "DOD";

create view epOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "EP";

create view btOffers as select o.c\_id, o.j\_id, o.st\_id from offers as o, student as s where o.st\_id = s.st\_id and s.dept = "BT";

### view for slots….

create view slot\_time as (select s.\*, sa.j\_id,sa.c\_id from slot\_alloted sa, slots s where s.slot\_id = sa.slot\_id);

**Some queries for displaying statisctics ….**

**DEPT WISE :-**

#registered students

select count(\*), dept from student group by dept

#total offers

select count(\*), s.dept from offers as o, student as s where o.st\_id = s.st\_id

group by s.dept

#actual no. of students placed

select count(\*), s.dept from (select distinct(oTemp.st\_id) from offers as oTemp) as o, student as s

where o.st\_id = s.st\_id

group by s.dept

#%age placed

#average salary

select s.dept, avg(j.ctc) from offers as o, student as s, job\_profile as j where j.j\_id = o.j\_id and j.c\_id = o.c\_id and o.st\_id = s.st\_id group by s.dept

#company wise no. of offers

select temp.cnt, c.name from (select count(\*) as cnt, o.c\_id from offers as o group by o.c\_id) as temp, company as c

where c.c\_id = temp.c\_id

**Some other Queries**

* Display all the applications for a company

select \* from apply as a, student as s, job\_profile as j where s.st\_id = a.st\_id and a.j\_id = j.j\_id and a.c\_id = j.c\_id and j.c\_id = 2

* Display all the application for a job profile

select \* from apply as a, student as s, job\_profile as j where s.st\_id = a.st\_id and a.j\_id = j.j\_id and a.c\_id = j.c\_id and j.c\_id = 2 and j.j\_id = 1

* Show pending application

select \* from company as c,job\_profile as j where c.c\_id = j.c\_id and j.approved='P'

* Update to make a job profile approved.

UPDATE `job\_profile` SET `approved`='Y' where j\_id = 1 and c\_id = 2

* See all the offers which a company has made, along with yes or no from student

select \* from offers as o, student as s, job\_profile as j

where o.c\_id=1 and s.st\_id = o.st\_id and o.j\_id = j.j\_id and o.c\_id = j.c\_id

* See all the students who were not offered job by a company

select \* from student as s1 where s1.st\_id in

(

select a.st\_id from apply as a, job\_profile as j1 where a.j\_id = j1.j\_id and a.c\_id = j1.c\_id and j1.c\_id = 2 and a.st\_id not in (

select o.st\_id from offers as o, job\_profile as j2

where o.c\_id=2 and o.j\_id = j2.j\_id and o.c\_id = j2.c\_id

)

)