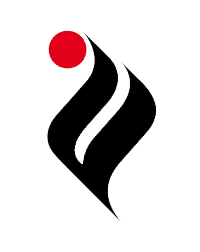
**PLACEMENT**

**RECORD AUTOMATION**

**SYSTEM**



DEVELOPERS:

RISHAB :101603276

SABHAN GOYAL:101610077

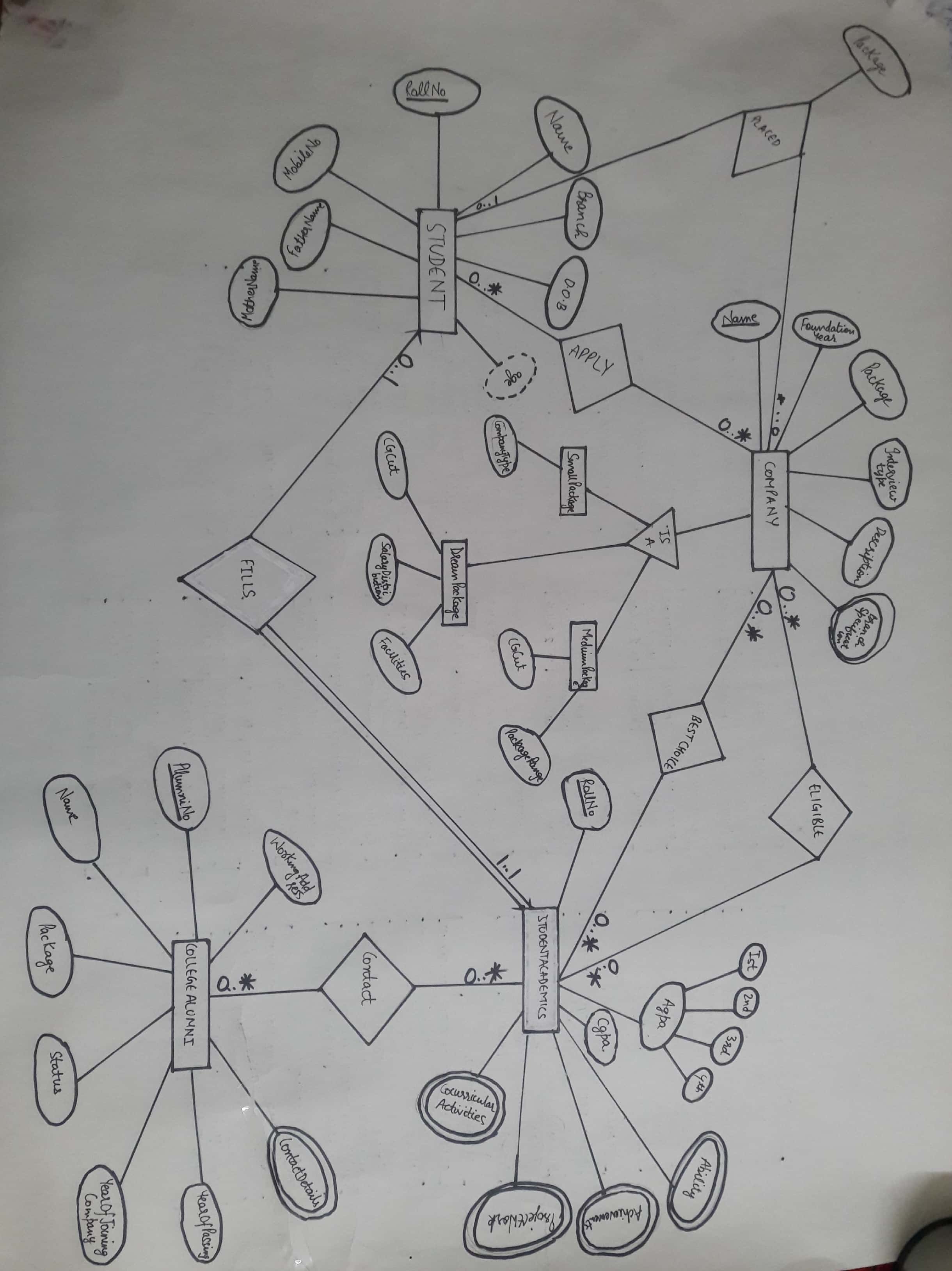
**INDEX**

1. Requirement Analysis
2. ER Diagram
3. ER to Table
4. FD Diagram
5. Finalised Table
6. PL/SQL Codes

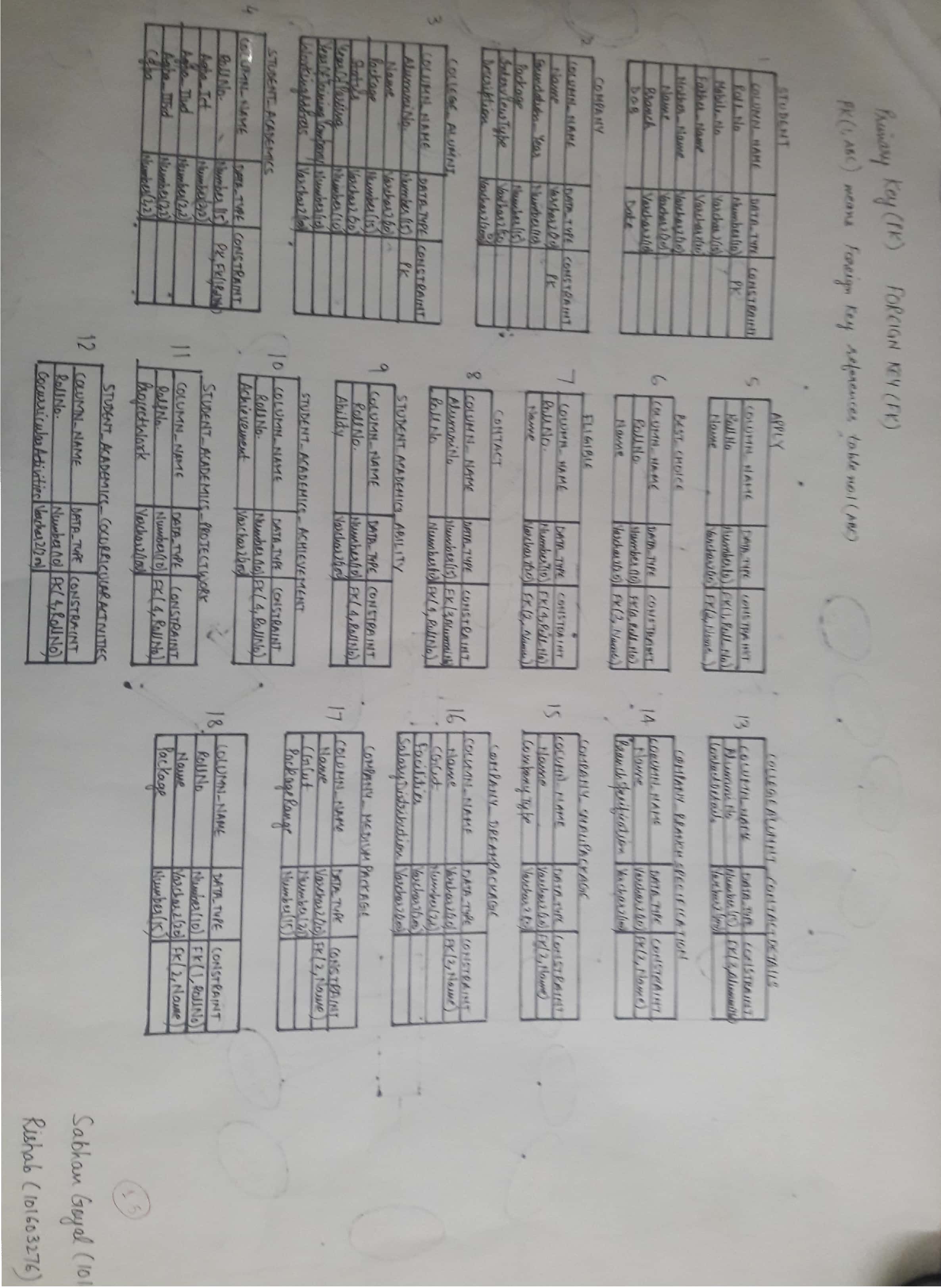
**REQUIREMENT ANALYSIS**

The project includes student registration and company registration. Each student can apply for various companies coming for recruitments. Company are categorized in 3 Categories-Small Package, Medium Package, Dream Package. Person once placed in the category cannot apply in the same category but can apply in the higher category if it exists. Once the registration happens, the software calculates who are eligible for company. Companies are shown student’s ability according to their demand. Similarly, students are shown best companies suitable according to their abilities. After that the data of college alumni are shown to them, the student can contact them and contact record is maintained. After the placement procedure ends, result of selected ones is displayed and they are removed from placement competition and are added in the list of selected ones. The whole project focuses on making the placement management easier.

**ER DIAGRAM**

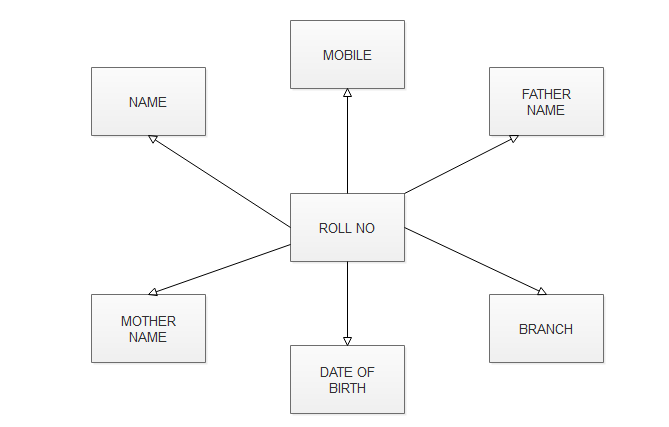
****

**ER TO TABLES**

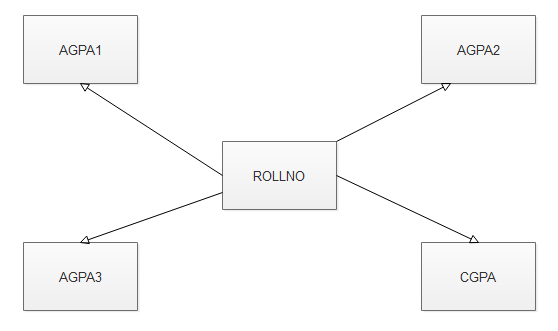
****

**FD DIAGRAMS FOR TABLES**

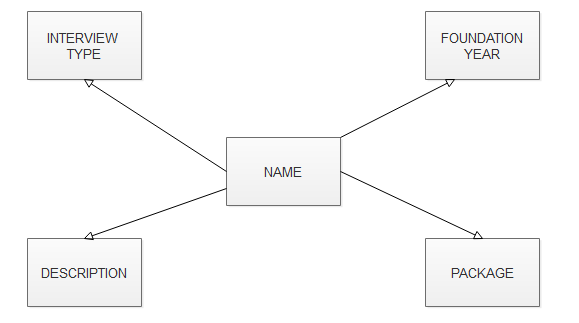
STUDENT



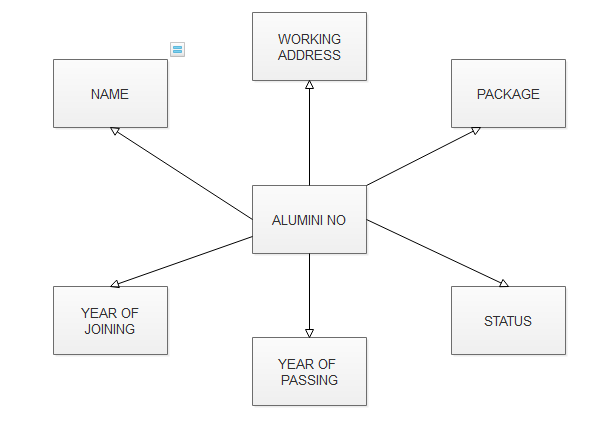
STUDENT\_ACADEMICS



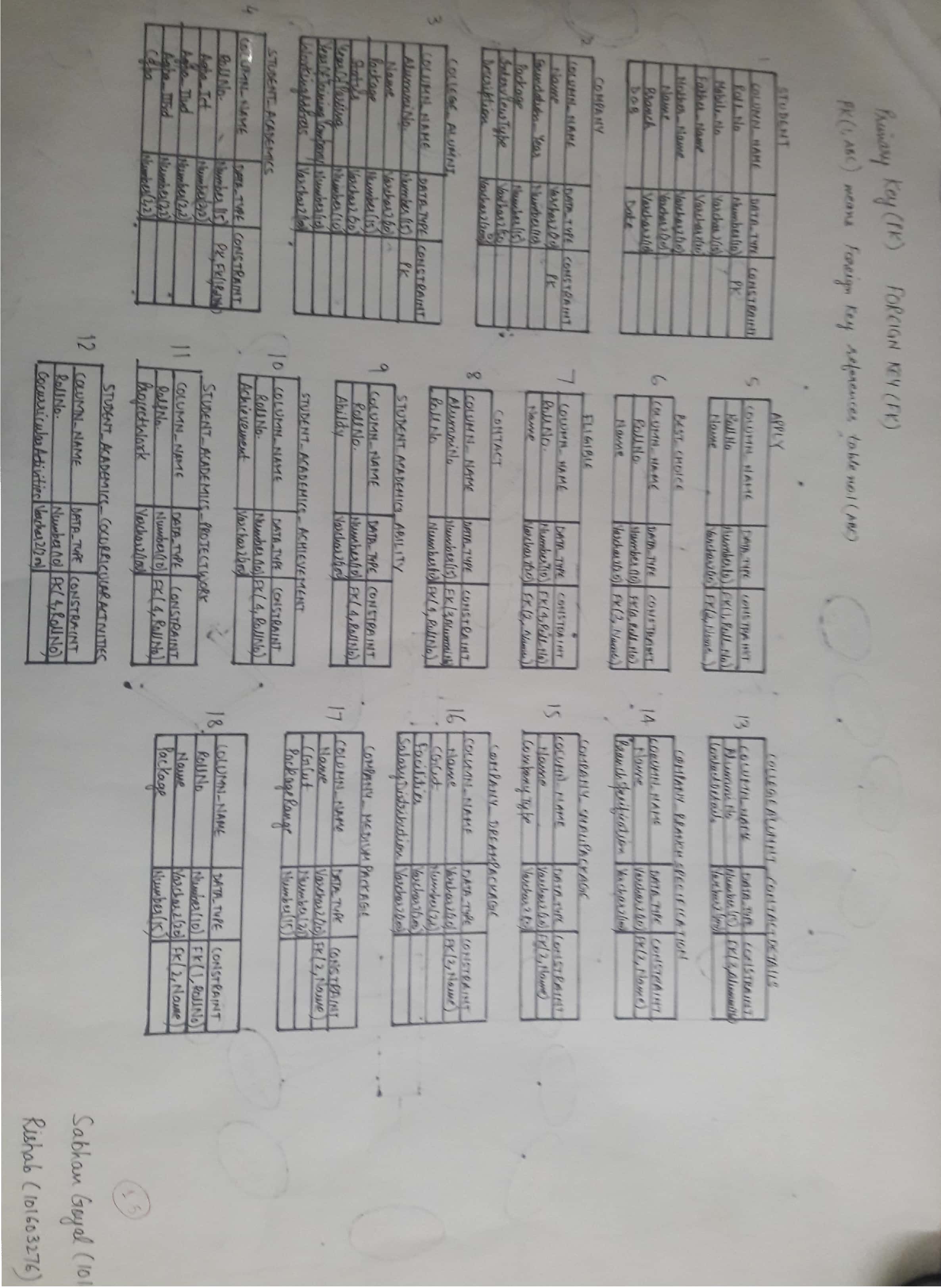
COMPANY



ALUMUNI



**FINALISED TABLES**

****

**Pl/Sql codes**

1. **INSERTION CODE AND EXCEPTION HANDELLING**

declare

rno student.rollno%type := &rollno;

sname student.name%type := '&name';

mob student.mobile\_no%type := '&mob';

fname student.father\_name%type := '&father\_name';

mname student.mother\_name%type := '&mother\_name';

branch student.branch%type := '&branch';

dob student.dob%type := '&dob';

DUPLICATE EXCEPTION;

PRAGMA EXCEPTION\_INIT(DUPLICATE,-00001);

begin

insert into student values (rno,sname,mob,fname,mname,branch,dob);

EXCEPTION

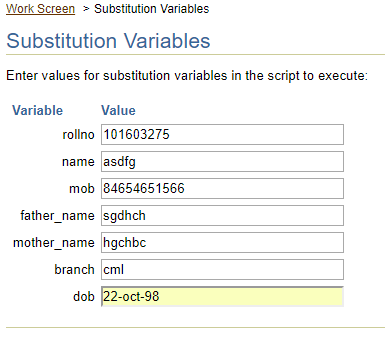
WHEN DUPLICATE THEN

DBMS\_OUTPUT.PUT\_LINE('RNO ALREADY USED');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(' ERROR OCCURED');

end;



1. **Procedure**

create or replace procedure company\_specialisation (nm in varchar2,checkt in number,ct in varchar2,cg in number,fc in varchar2) is

begin

if checkt=1 then

insert into company\_small\_package(name,company\_type) values (nm,ct);

elsif checkt = 2 then

insert into company\_medium\_package(name,cgpa\_cut) values (nm,cg);

elsif checkt = 3 then

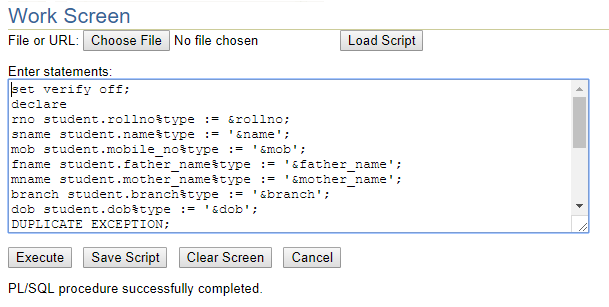
insert into company\_dream\_package(name,cgpa\_cut,facility) values (nm,cg,fc);

else

dbms\_output.put\_line('wrong input');

end if;

end;



1. **Cursor**

declare

cursor c1 is select rollno,name from placed;

begin

for r in c1 loop

dbms\_output.put\_line(r.rollno||'placed in'||r.name);

end loop;

end;

declare

cursor c1 is select \* from company;

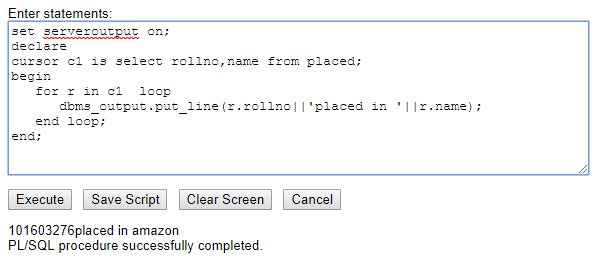
begin

for r in c1 loop

dbms\_output.put\_line(name||foundation\_year||package||interview\_type||description);

end loop;

end;



1. **Trigger**

declare

rno apply.rollno%type := &rollno;

name apply.name%type := '&name';

begin

insert into apply values (rno,name);

end;

CREATE OR REPLACE TRIGGER ELIGIBLE\_TRIG

AFTER INSERT OR UPDATE ON APPLY

FOR EACH ROW

DECLARE

cgs student\_academics.cgpa%type;

cgc company\_dream\_package.cgpa\_cut%type;

BEGIN

select cgpa into cgs from student\_academics where rollno=:new.rollno;

select cgpa\_cut into cgc from company\_dream\_package where name=:new.name;

if cgs>=cgc then

insert into eligible values(:new.rollno,:new.name);

end if;

END;

TABLE FILLED WITH TRIGGER

