DBMS – Placement Management System

Submitted By

Name: Naren Chandrashekhar

SRN: PES2UG20CS216

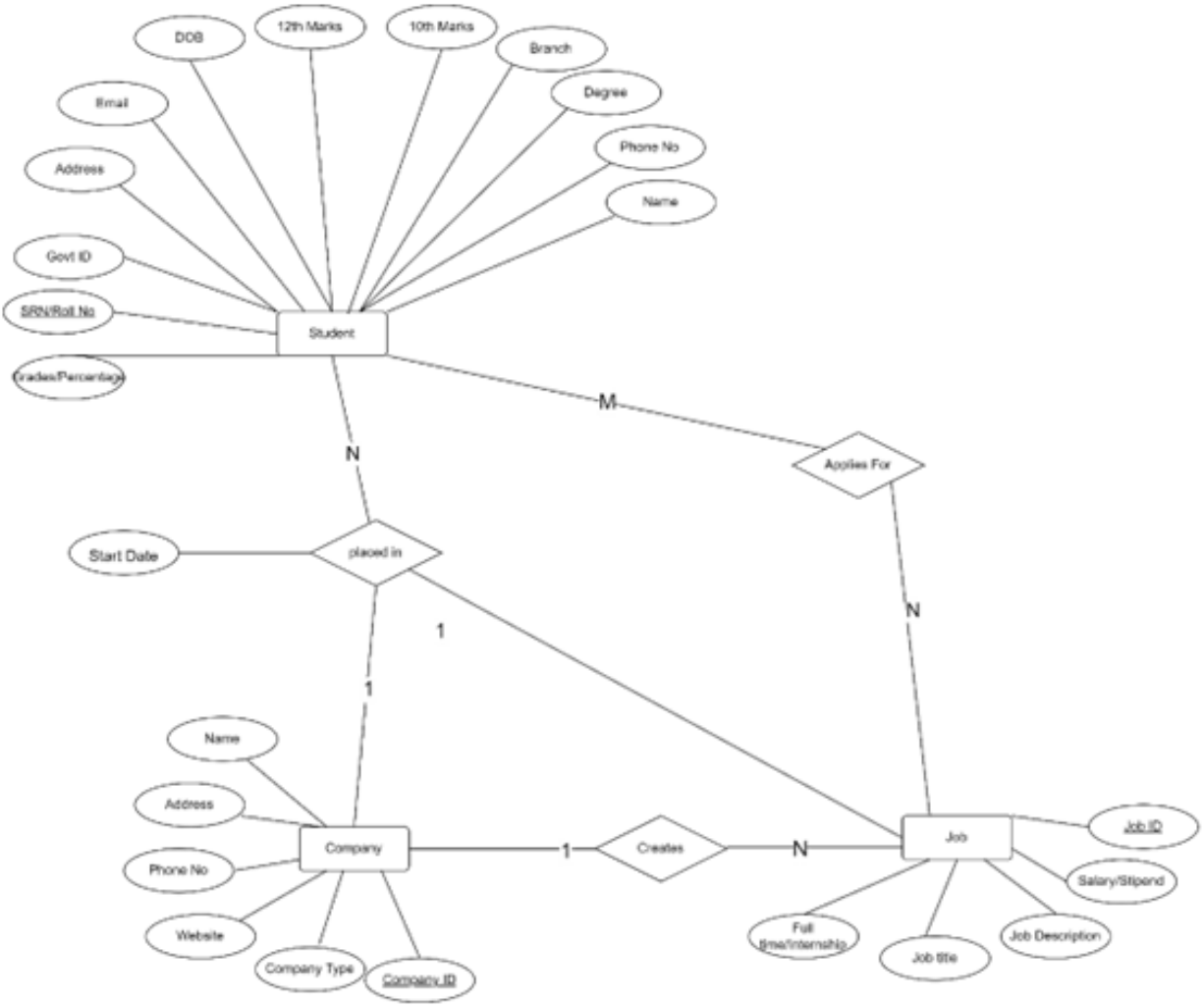
V Semester Section D

Short Description and Scope of the Project

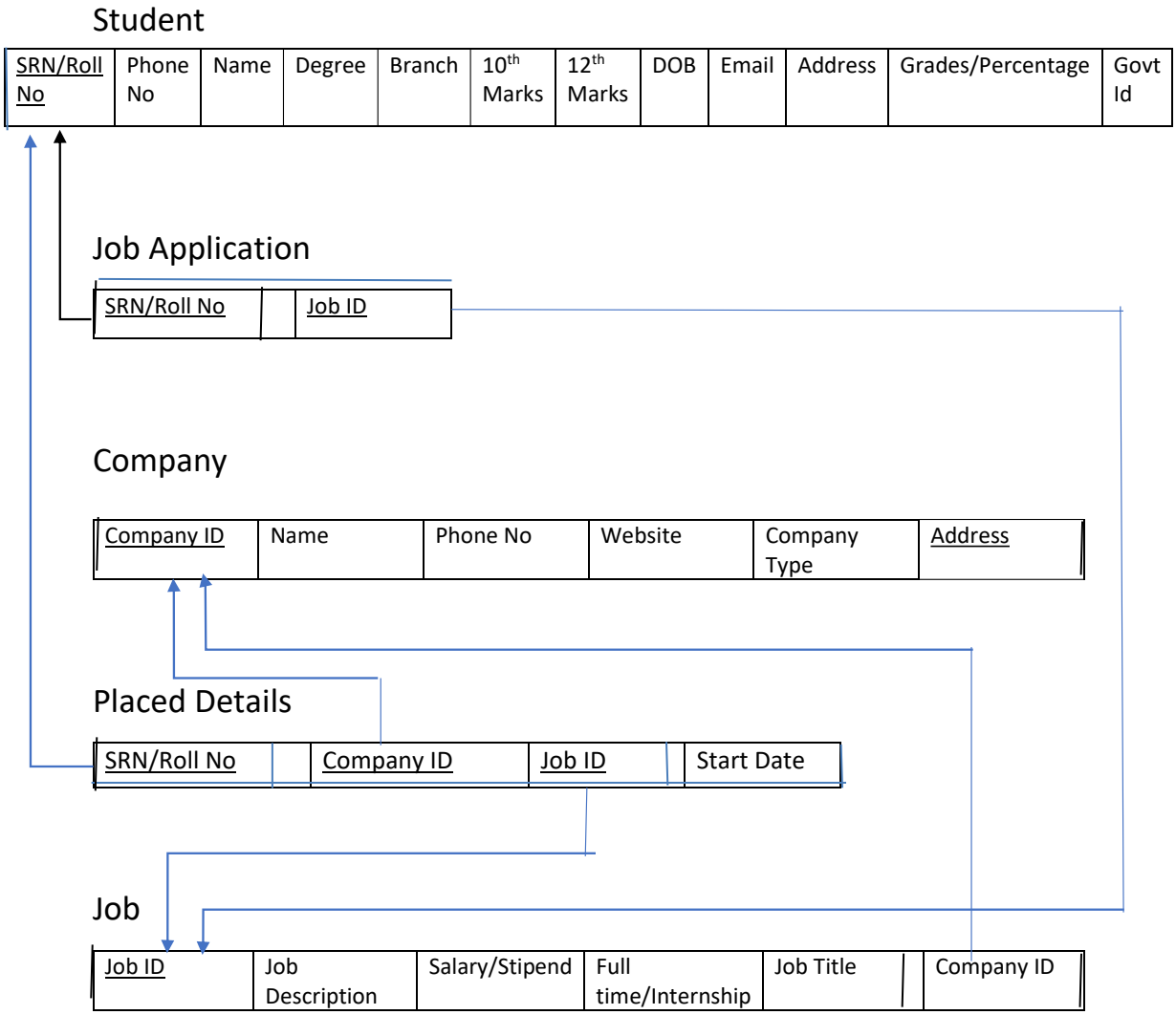
The objective of this project is to design and develop database management application integrated with a user interface to store and manage placement related data of students of a university/institution. The database has tables such as student details, company details, placement details, job details and job application details.

This project can be used by users such as placement officers, students, college management, department heads to store, manage and generate reports related to placement activities in a college. This application can be used by universities to keep track of its students and their records. It can be used to gain meaningful insights about any university and details of student placements.

ER Diagram


DOB
12th Marks
10th Marks
Branch
Degree
Email
Phone No
Address
Name
Govt ID
SRN/Roll No
Student
Grades/Percentage
M
N
Applies For
Start Date
placed in
1
1
N
Name
Address
Company
1
Creates
N
Job
Job ID
Phone No
Salary/Stipend
Website
Company Type
Company ID
Full time/Internship
Job title
Job Description

Relational Schema



DDL statements - Building the database

Creating tables

Student Table

```
create table student(SRN int NOT NULL PRIMARY KEY, Name VARCHAR(100), Phone_no int, Degree VARCHAR(20), Branch VARCHAR(30), 10th int,12th int, DOB date, Email VARCHAR(50), Address VARCHAR(100), CGPA int, Govt_ID VARCHAR(100));
```

Company Table

```
create table company(Company_ID int NOT NULL PRIMARY KEY, Name VARCHAR(50), Phone_No int, Website VARCHAR(50), Company_Type VARCHAR(30), Address VARCHAR(100) NOT NULL);
```

Job Table

```
create table job(Job_ID int NOT NULL PRIMARY KEY, Job_Description VARCHAR(100), Salary_or_Stipend int, Full_time_or_Internship VARCHAR(30), Job_Title VARCHAR(40), Company_ID int, FOREIGN KEY (Company_ID) REFERENCES company(Company_ID));
```

Job_Application Table

```
create table job_application(SRN int, FOREIGN KEY(SRN) REFERENCES student(SRN), Job_ID int, FOREIGN KEY(Job_ID) REFERENCES job(Job_ID));
```

Placed_Details Table

```
create table placed_details(SRN int, FOREIGN KEY(SRN) REFERENCES student(SRN), Company_ID int, FOREIGN KEY(Company_ID) REFERENCES company(Company_ID), Job_ID int, FOREIGN KEY(Job_ID) REFERENCES job(Job_ID), Start_Date date);
```

Populating the Database

```
select * from student;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	SRN	Name	Phone_no	Degree	Branch	10th	12th	DOB	Email	Address	CGPA	Govt_ID
Edit Copy Delete	PES001	Axe	1111	B-Tech	Computer Science	90	92	2000-01-01	axe@gmail.com	Mumbai	9	GI005
Edit Copy Delete	PES002	Ave	1112	B-Tech	Computer Science	95	91	2000-02-03	ave@gmail.com	Mumbai	8	GI015
Edit Copy Delete	PES003	Apple	1131	B-Tech	Computer Science	80	82	2001-01-01	apple@gmail.com	Bangalore	10	GI025

```
SELECT* from company;
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	Company_ID	Name	Phone_No	Website	Company_Type	Address
Edit Copy Delete	1	Amazon	9999	amazon.in	Product	Bangalore
Edit Copy Delete	2	Intel	9998	intel.com	Product	Bangalore
Edit Copy Delete	3	Infosys	9988	infosys.com	Services	Bangalore
Edit Copy Delete	4	Facebook	9991	facebook.com	Product	Hyderabad

```
SELECT* from job;
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	Job_ID	Job_Description	Salary_or_Stipend	Full_time_or_Internship	Job_Title	Company_ID
Edit Copy Delete	101	Data Analyst	110000	Full time	Software Engineer	4
Edit Copy Delete	1000	Front End	200000	Full time	Software Engineer	1
Edit Copy Delete	2000	UI/UX	150000	Full time	Software Engineer	3

```
SELECT* from job_application;
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows:

Extra options

SRN	Job_ID
PES001	1000
PES002	2000

Showing rows 0 - 1 (2 total, Query took 0.0011 seconds.)

```
SELECT* from placed_details;
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 Filter rows:

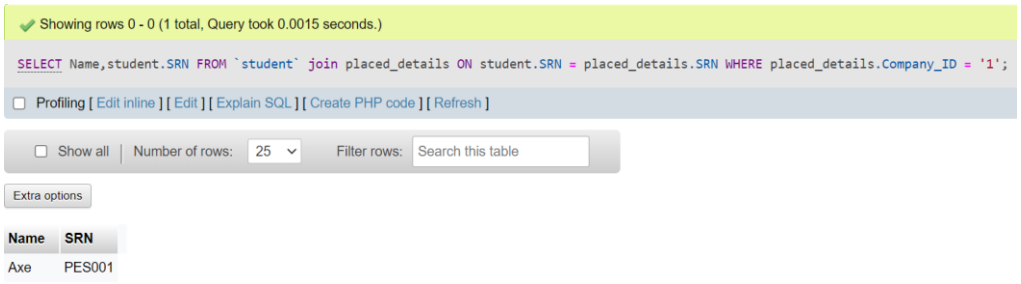
Extra options

SRN	Company_ID	Job_ID	Start_Date
PES001	1	1000	2022-12-01
PES002	2	2000	2022-09-08

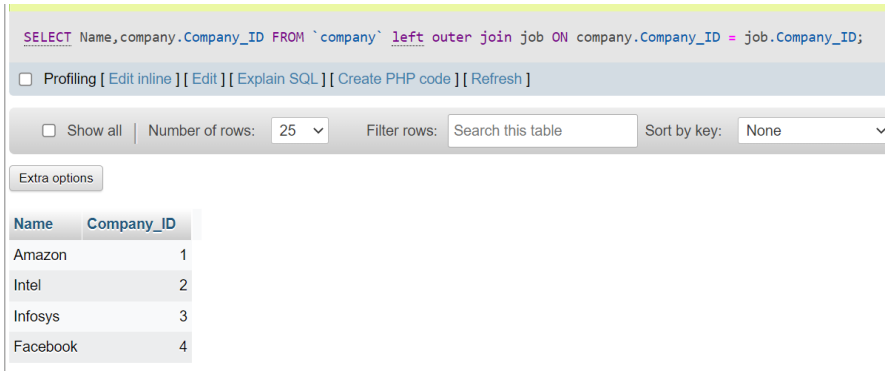
Join Queries

Showcase at least 4 join queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

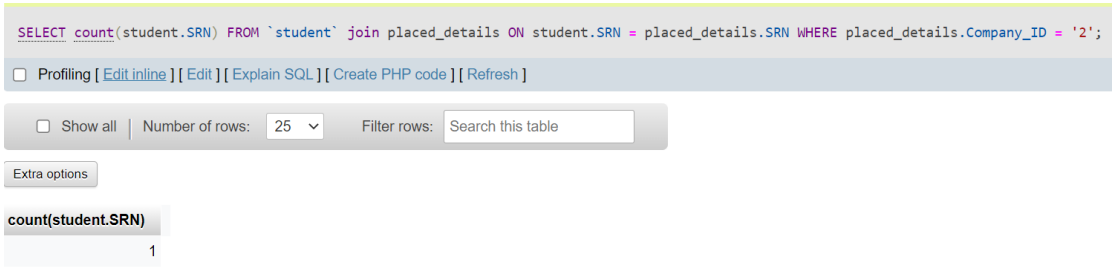
1. Retrieve Name and SRN of all students who have been placed in a company with id = 1.



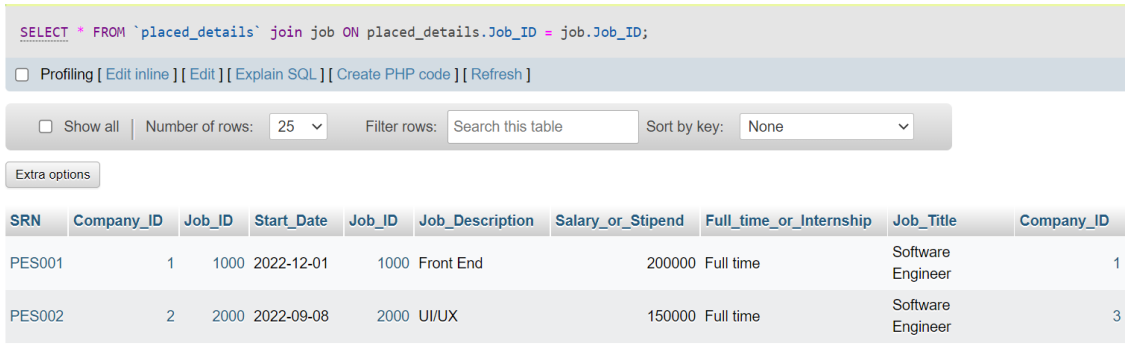
2. Retrieve the names of all company who have posted and not posted a job



3. Retrieve the total number of students who got placed in Intel



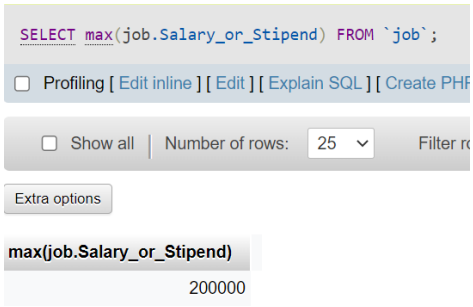
4. Retrieve all data about a placed student



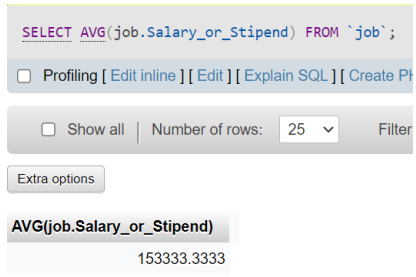
Aggregate Functions

Showcase at least 4 Aggregate function queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

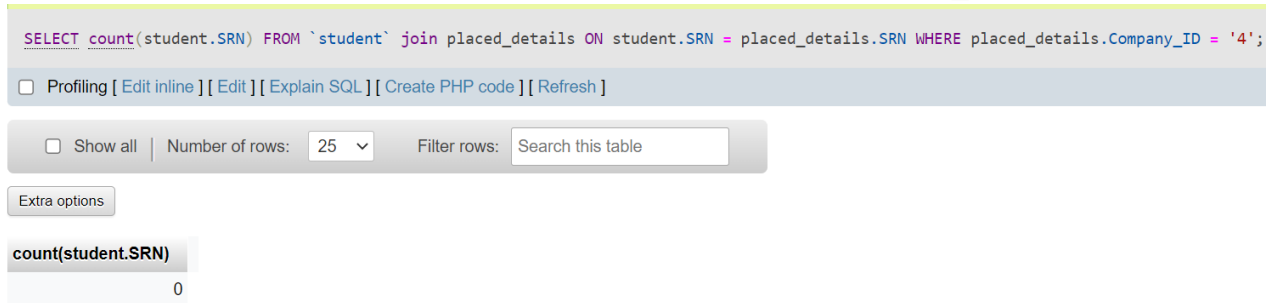
1. Retrieve the max salary of a student



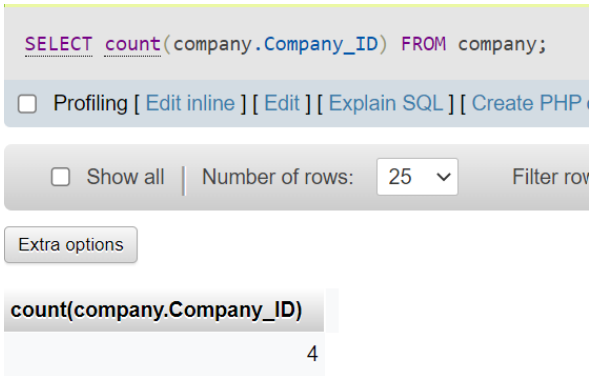
2. Retrieve avg salary of student



3. Retrieve the total number of students who got placed in Company_ID = 4



4. Retrieve the number of companies who have come to the college for placements



Set Operations

Showcase at least 4 Set Operations queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1. Select all student SRN who have been placed in any company

```
SELECT student.SRN from student INTERSECT SELECT placed_details.SRN from placed_details;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key:

Extra options

SRN
PES001
PES002

2. Select all Company_ID where students have been placed

```
SELECT company.Company_ID from company INTERSECT SELECT placed_details.Company_ID from placed_details;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

Company_ID
1
2

3. Select all Company_ID where there are job openings

```
SELECT company.Company_ID from company INTERSECT SELECT job.Company_ID from job;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort

Extra options

Company_ID
1
3
4

Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

Function: To check if a student is eligible for placements. Eligibility criteria is >65% for 10th and 12th marks

```
Create Function eligibility1(SRN varchar(20)) Returns varchar(100) Deterministic Begin Declare per10 int; Declare per12 int; Declare msg varchar(100); Set per10 = (select 10th from student where student.SRN=SRN); Set per12 = (select 12th from student where student.SRN=SRN); If per10 > 65 and per12>65 Then Set msg = Concat('You are eligible for placements'); Else SET msg = Concat('You are not eligible for placements'); End if; Return msg; End;
```

[Edit inline] [Edit] [Create PHP code]

Showing rows 0 - 2 (3 total, Query took 0.0009 seconds.)

```
select SRN,Name,eligibility1(SRN) from student;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	SRN	Name	eligibility1(SRN)
Edit Copy Delete	PES001	Axe	You are eligible for placements
Edit Copy Delete	PES002	Ave	You are eligible for placements
Edit Copy Delete	PES003	Apple	You are eligible for placements

Procedure: To check if a given company is a Product type or not

```
CREATE procedure company_typ(IN Company_ID int(11) ,OUT msg varchar(100)) BEGIN DECLARE cnt varchar(100); set cnt= (select Company_Type from company where Company_ID=company.Company_ID); IF cnt = 'Product' THEN set msg= 'This company is a product comapny, hence the compensations given are higher'; ELSE set msg='This is not a product company'; END IF; END;;
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
call company_typ(1,@x);
```

[Edit inline] [Edit] [Create PHP code]

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
select @x;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

@x
This company is a product comapny, hence the compe...

Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

Triggers.

The problem statement is: Validating the SRN of a student in the student table. The format should be PES___. In the below example, we have given xXX instead of PES, and hence the message given back is illegal srn.

```
1  DELIMITER $$
2  create trigger srn_check
3  before insert
4  on student
5  for each row
6  begin
7  declare error_msg varchar(255);
8  set error_msg='illegal srn';
9  if ( new.SRN != 'PES%' ) then signal sqlstate '45000' set message_text=error_msg;
10 end if;
11 end $$
12 DELIMITER ;
```

```
insert into student values('xXX005', 'Amp','2111','B-Tech','Computer Science', 91,'92','2000-03-02','amp@gmail.com','Bangalore','7','GI012');
```

MySQL said:

```
#1644 - illegal srn
```

Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be an window to accept and run any SQL statement and display the result

I have used streamlit and pymysql to connect my python frontend to mysql backend.
I have insert and fetch function on company database.