DBMS-Mini Project

Attendance Mapping System

Submitted By:

Amruth S

PES1UG20CS038

V Semester Section ‘A’

Short Description and Scope of the Project

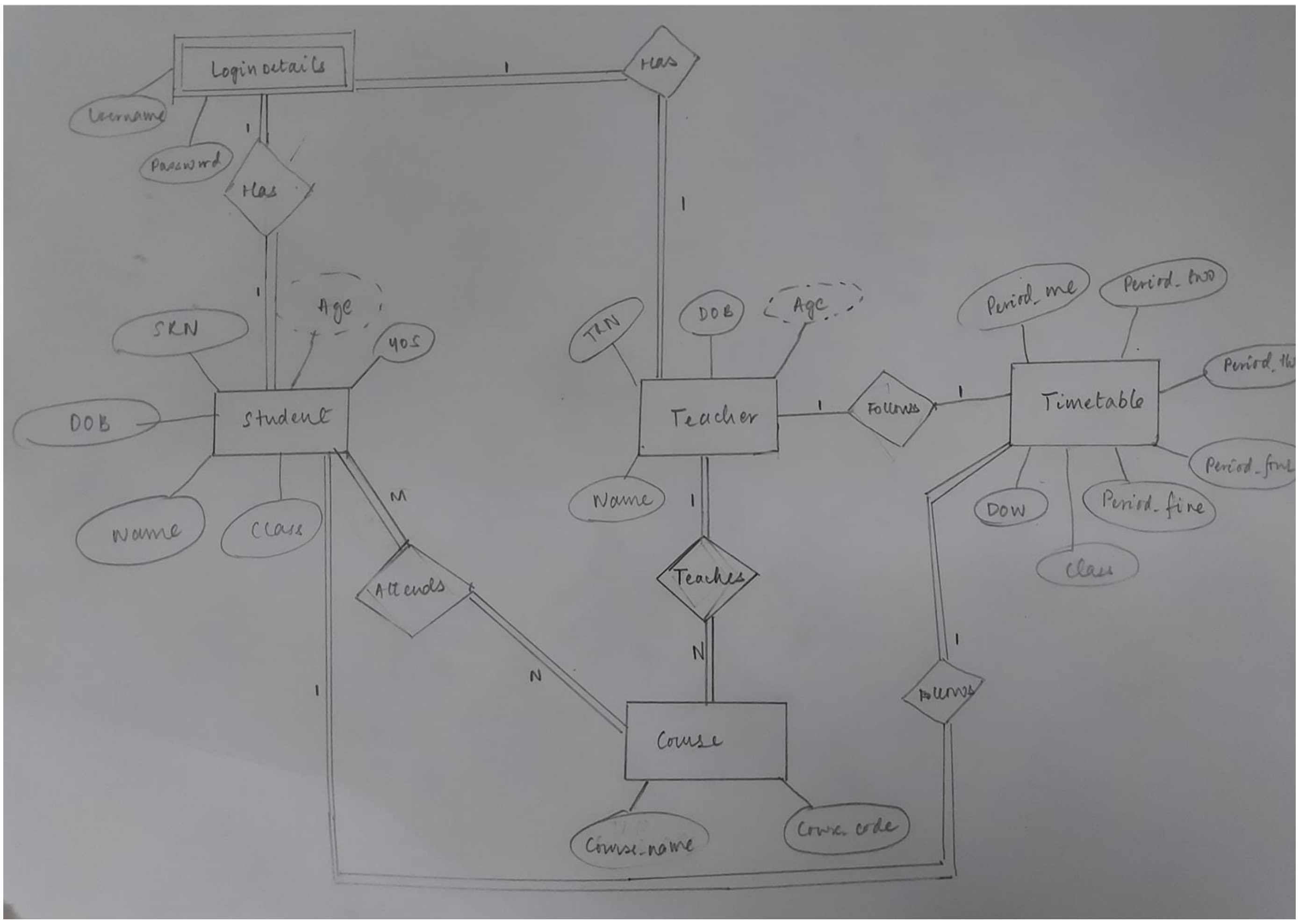
The Attendance Mapping System is an application developed to ease the routine of taking attendance in Schools, Colleges and Universities. It has an intuitive UI which makes it easy for teachers to allot attendance to students and for students to keep track of their attendance in their opted courses.

It not only helps the students and teachers; it also helps institution administrators to map courses to students and teachers to courses.

It also helps them keep a track total student, teachers in the institution and more.

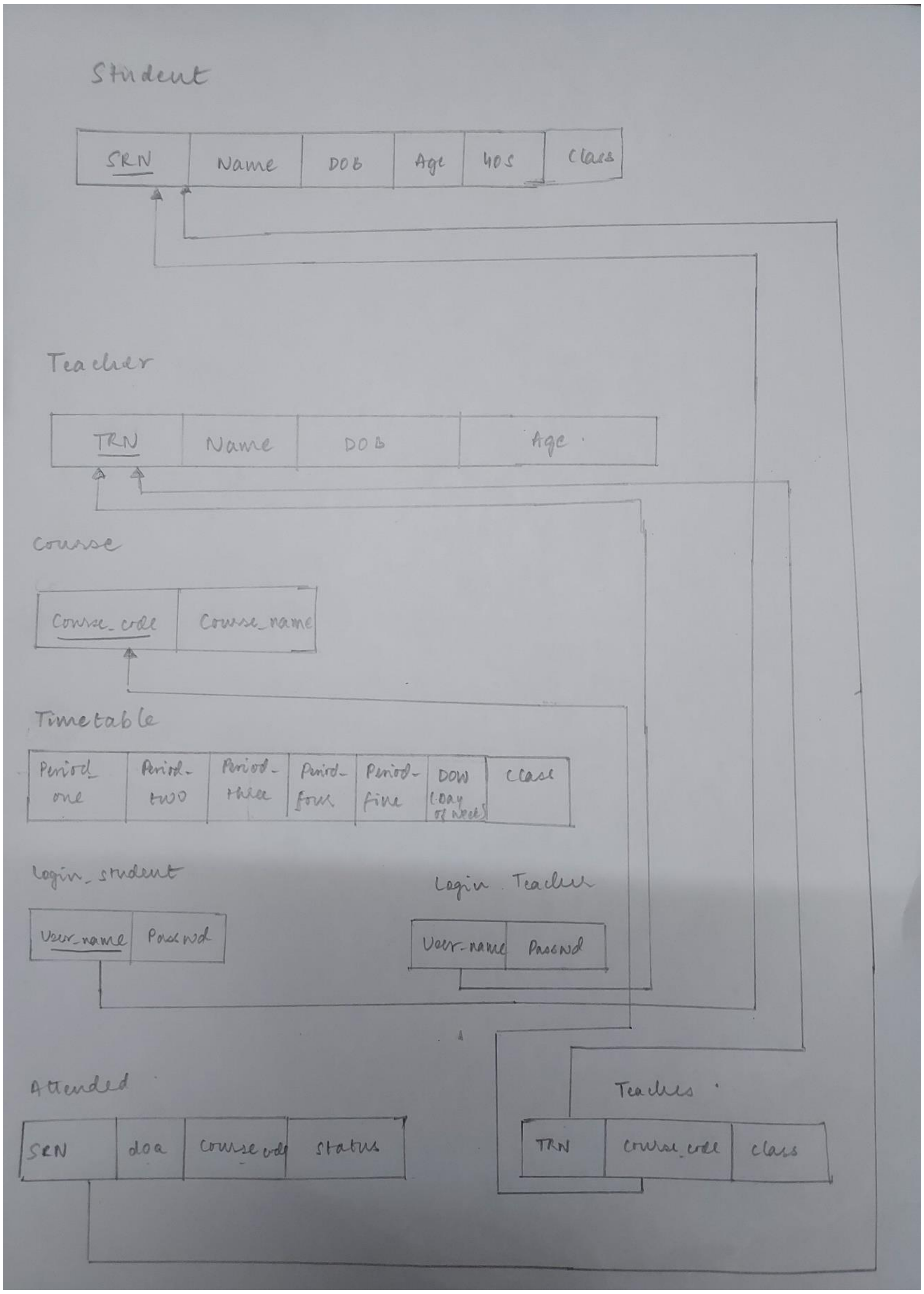
Since the developed app is a web-app it can accessed from any device, anywhere across the world.

ER Diagram



Login details
Username
Password
Has
Has
1
1
1
1
Student
SRN
Age
YOS
DOB
Name
Class
M
Attends
N
Teacher
TRN
DOB
Age
Name
1
Follows
1
Teaches
N
Course
Course_name
Course code
Timetable
Period_one
Period_two
Period_thr
Period_four
Period_fine
DOW
Class
1
Owns
1

Relational Schema



DDL statements - Building the database

```
CREATE TABLE `student` (
  `SRN` varchar(20) NOT NULL,
  `Name` varchar(20) DEFAULT NULL,
  `DOB` date DEFAULT NULL,
  `Age` int(11) DEFAULT NULL,
  `YOS` int(11) DEFAULT NULL,
  `Class` varchar(3) DEFAULT NULL
)

CREATE TABLE `teacher` (
  `TRN` varchar(20) NOT NULL,
  `Name` varchar(20) DEFAULT NULL,
  `DOB` date DEFAULT NULL,
  `Age` int(11) DEFAULT NULL
)

CREATE TABLE `timetable` (
  `Period_one` varchar(20) DEFAULT NULL,
  `Period_two` varchar(20) DEFAULT NULL,
  `Period_three` varchar(20) DEFAULT NULL,
  `Period_four` varchar(20) DEFAULT NULL,
  `Period_five` varchar(20) DEFAULT NULL,
  `dow` varchar(20) DEFAULT NULL,
  `class` varchar(3) DEFAULT NULL
)
CREATE TABLE `course` (
  `course_name` varchar(20) DEFAULT NULL,
  `course_code` varchar(20) NOT NULL
)
```

```
CREATE TABLE `teaches` (
  `TRN` varchar(20) DEFAULT NULL,
  `course_code` varchar(20) DEFAULT NULL,
  `class` varchar(5) DEFAULT NULL
   FOREIGN KEY (TRN) REFERENCES teacher(TRN),
  FOREIGN KEY (course_code) REFERNCES course(course_code)
)

CREATE TABLE `login_student` (
`User_name` varchar(20),
`Passwd` varchar(20),
PRIMARY KEY (User_name),
FOREIGN KEY (User_name) REFERNCES student(SRN)
)
CREATE TABLE `login_teacher` )
`User_name` varchar(20),
`Passwd` varchar(20),
PRIMARY KEY (User_name),
FOREIGN KEY (User_name) REFERNCES teacher(TRN)
)
CREATE TABLE `attended` (
  `SRN` varchar(20) DEFAULT NULL,
  `doa` date DEFAULT NULL,
  `course_code` varchar(20) DEFAULT NULL,
  `status` varchar(1) DEFAULT NULL,
   FOREIGN KEY (SRN) REFERNCES student(SRN)
)
```

Populating the Database

Populating the student table.

```
INSERT INTO `student` (`SRN`, `Name`, `DOB`, `Age`, `YOS`, `Class`) VALUES
('011', 'Norm', '2002-11-11', 20, 3, '5A'),
('016', 'Doof', '2002-11-12', 20, 3, '5B'),
('019', 'Phineas', '2002-01-01', 20, 3, '5A'),
('021', 'Jerry', '2002-12-21', 19, 3, '5A'),
('022', 'Buford', '2003-01-31', 19, 3, '5B'),
('045', 'Carl', '2001-01-01', 21, 3, '5B'),
('056', 'Ferb', '2001-03-01', 21, 3, '5B'),
('803', 'Saurav', '2001-04-01', 21, 3, '5A'),
('PES1UG20CS018', 'Rokhade', '2002-07-25', 20, 3, '5B'),
('PES1UG20CS025', 'Akarsh', '2001-01-01', 21, 3, '5A'),
('PES1UG20CS035', 'Amogh', '2002-08-19', 20, 3, '5B'),
('PES1UG20CS038', 'Amruth S', '2002-06-04', 20, 3, '5A');
```

Populating the teacher table.

```
INSERT INTO `teacher` (`TRN`, `Name`, `DOB`, `Age`) VALUES
('TRN001', 'Saurav', '1972-01-01', 50),
('TRN002', 'Amit', '1973-08-08', 49),
('TRN003', 'John', '1973-01-08', 49),
('TRN004', 'Don', '1969-08-11', 53),
('TRN005', 'Beckett', '1980-08-08', 42),
('TRN006', 'Strange', '1981-08-08', 41),
('TRN007', 'Rogers', '1957-08-08', 65),
('TRN008', 'Stark', '1965-09-23', 57),
('TRN009', 'Steve', '1967-08-08', 55),
('TRN010', 'Tony', '1962-08-08', 60),
('TRN011', 'Bruce', '1975-08-08', 47),
('TRN012', 'Murphy', '1999-08-08', 23);
```

Populating the course table.

```
INSERT INTO `course` (`course_name`, `course_code`) VALUES
('Physics', 'PHY101'),
('Chemisrty', 'CHEM101'),
('Electronics', 'ECE101'),
('Engineering Drawing', 'DRAW101'),
('Mathematics', 'MATH101');
```

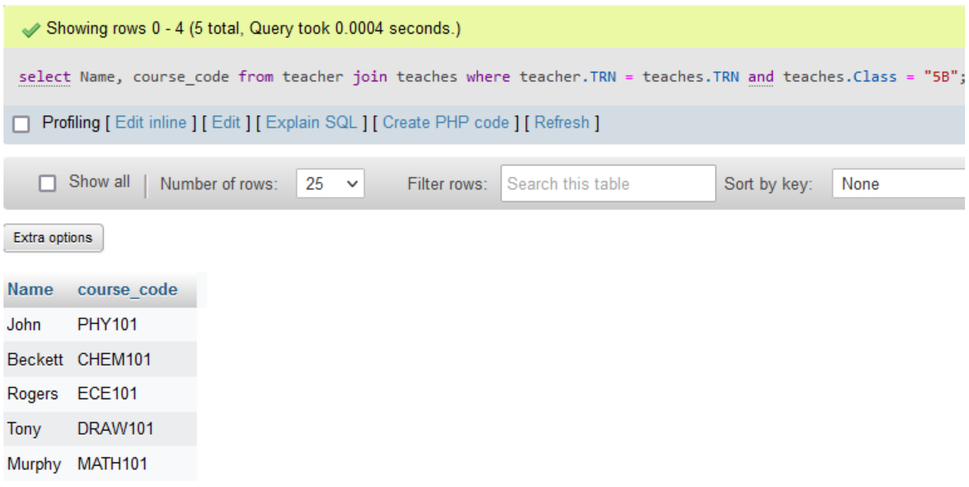
Populating the teaches table.

```
INSERT INTO `teaches` (`TRN`, `course_code`, `class`) VALUES
('TRN002', 'PHY101', '5A'),
('TRN003', 'PHY101', '5B'),
('TRN004', 'CHEM101', '5A'),
('TRN005', 'CHEM101', '5B'),
('TRN006', 'ECE101', '5A'),
('TRN007', 'ECE101', '5B'),
('TRN001', 'DRAW101', '5A'),
('TRN010', 'DRAW101', '5B'),
('TRN011', 'MATH101', '5A'),
('TRN012', 'MATH101', '5B')
```

Join Queries

1. Query to get all teachers teaching to a particular student

   Get Name, course_code from the join result of tables teacher and teaches where the join condition is teacher's TRN must be equal to the one in teaches and teaches.Class must be the same as the class of the student



Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

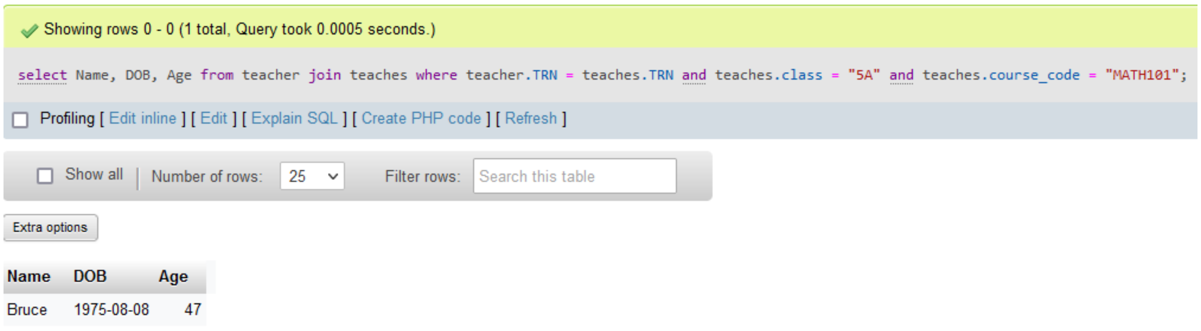
```
select Name, course_code from teacher join teaches where teacher.TRN = teaches.TRN and teaches.Class = "5B";
```

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

Name	course_code
John	PHY101
Beckett	CHEM101
Rogers	ECE101
Tony	DRAW101
Murphy	MATH101

2. Get all details of teacher teaching a particular subject for a class

   Get Name of teacher from the result of join of tables teacher and teaches where teaches.TRN = teacher.TRN teaches.class must be same as the class for which we are finding this result and teaches.course_code must be equal to course_code of the subject.


Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

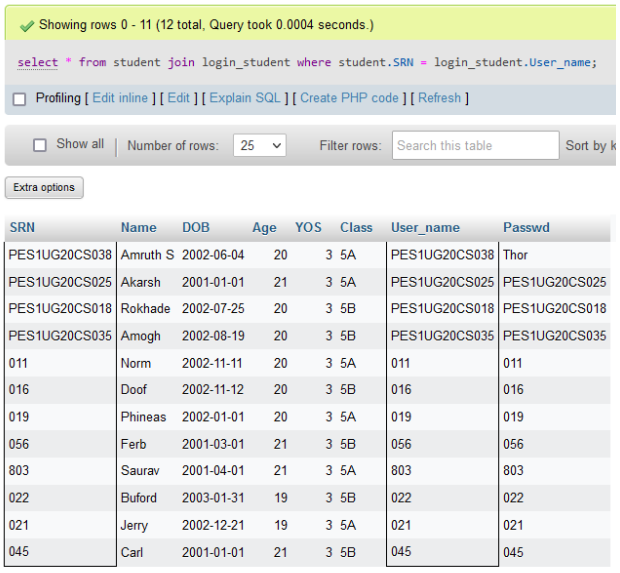
```
select Name, DOB, Age from teacher join teaches where teacher.TRN = teaches.TRN and teaches.class = "5A" and teaches.course_code = "MATH101";
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options

Name	DOB	Age
Bruce	1975-08-08	47

3. Get Login details and Details of all student

Get all columns from join of student and login_student on SRN. (SRN is the same as username)


Showing rows 0 - 11 (12 total, Query took 0.0004 seconds.)

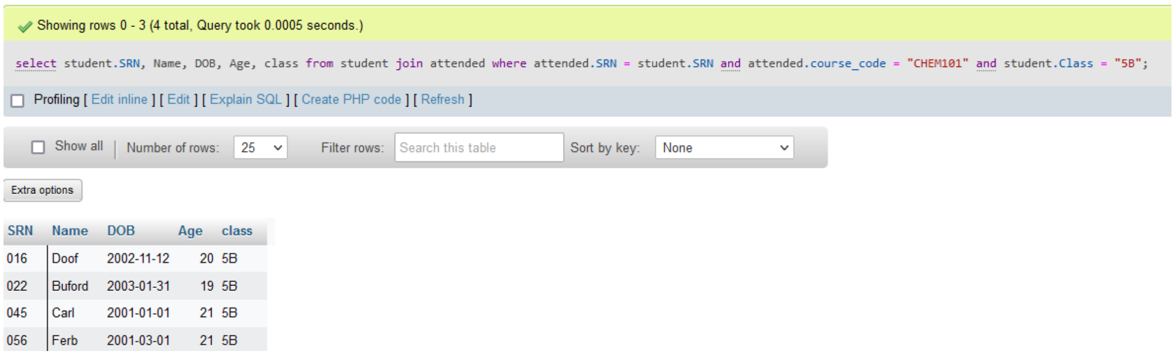
```
select * from student join login_student where student.SRN = login_student.User_name;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by k
Extra options

SRN	Name	DOB	Age	YOS	Class	User_name	Passwd
PES1UG20CS038	Amruth S	2002-06-04	20	3	5A	PES1UG20CS038	Thor
PES1UG20CS025	Akarsh	2001-01-01	21	3	5A	PES1UG20CS025	PES1UG20CS025
PES1UG20CS018	Rokhade	2002-07-25	20	3	5B	PES1UG20CS018	PES1UG20CS018
PES1UG20CS035	Amogh	2002-08-19	20	3	5B	PES1UG20CS035	PES1UG20CS035
011	Norm	2002-11-11	20	3	5A	011	011
016	Doof	2002-11-12	20	3	5B	016	016
019	Phineas	2002-01-01	20	3	5A	019	019
056	Ferb	2001-03-01	21	3	5B	056	056
803	Saurav	2001-04-01	21	3	5A	803	803
022	Buford	2003-01-31	19	3	5B	022	022
021	Jerry	2002-12-21	19	3	5A	021	021
045	Carl	2001-01-01	21	3	5B	045	045

4. Get details of students who attended class “5B” with course_code “CHEM101” on 14th November 2022.

Get SRN, Name, DOB, Age, Class from the result of join of tables student and attended on SRN where course_code = “CHEM101” and Class = “5B”


Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

```
select student.SRN, Name, DOB, Age, class from student join attended where attended.SRN = student.SRN and attended.course_code = "CHEM101" and student.Class = "5B";
```

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

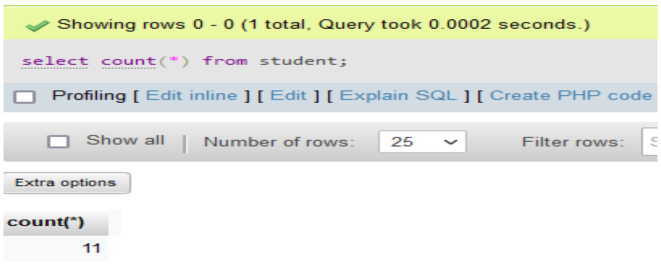
SRN	Name	DOB	Age	class
016	Doof	2002-11-12	20	5B
022	Buford	2003-01-31	19	5B
045	Carl	2001-01-01	21	5B
056	Ferb	2001-03-01	21	5B

Aggregate Functions

1. To get total number of students in the institution

   Get total number of rows in student table.

   SQL: select count(*) from student;

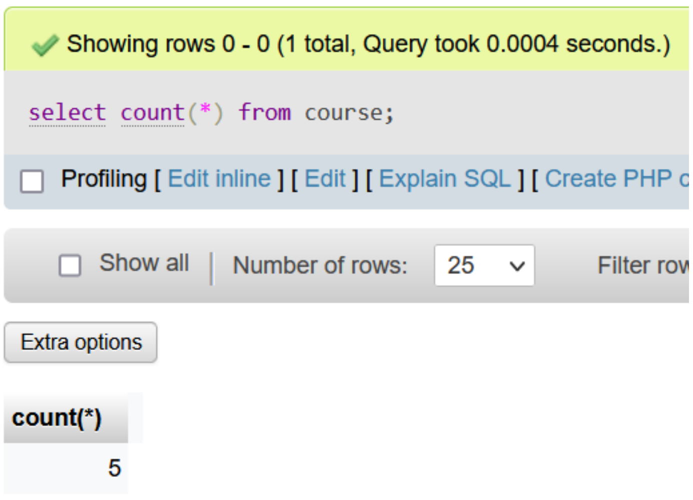


```
✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

select count(*) from student;

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

Show all | Number of rows: 25    Filter rows:

Extra options

count(*)
      11
```

2. To get total number of courses in the institution

   Get total number of rows in course table;

   SQL: select count(*) from course;

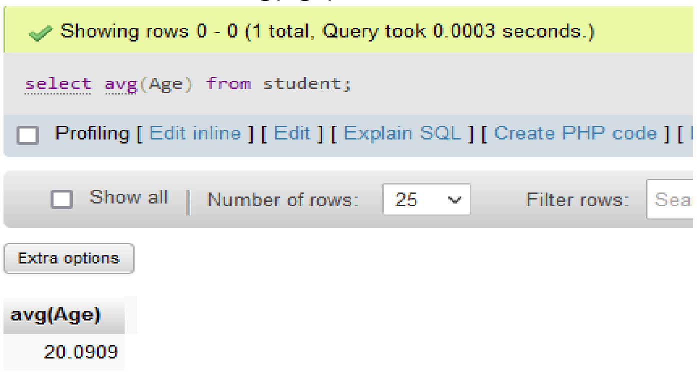


```
✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

select count(*) from course;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP c

Show all | Number of rows: 25    Filter row

Extra options

count(*)
       5
```

3. To get the average age of students in the institution.

   Get avg of Age column in student table.

   SQL: select avg(Age) from student;

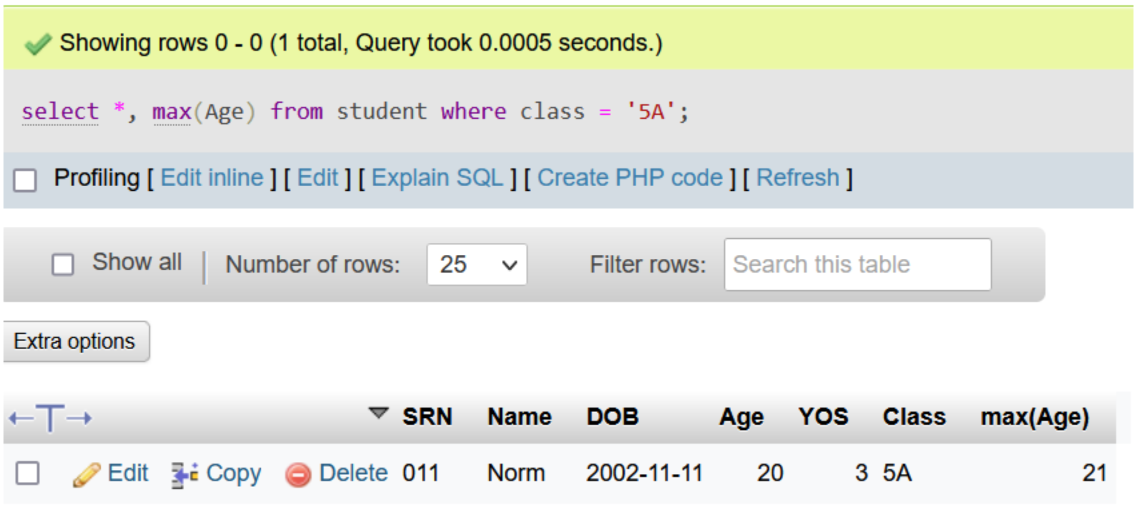


```
✓ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

select avg(Age) from student;

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

Show all | Number of rows: 25    Filter rows: Sea

Extra options

avg(Age)
 20.0909
```

4. Get Details of student who is the oldest in class ‘5A’.

   Get all details of student whose age is maximum in the age column.

   SQL: select *, max(Age) from student where class = ‘5A’;



Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
select *, max(Age) from student where class = '5A';
```

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

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

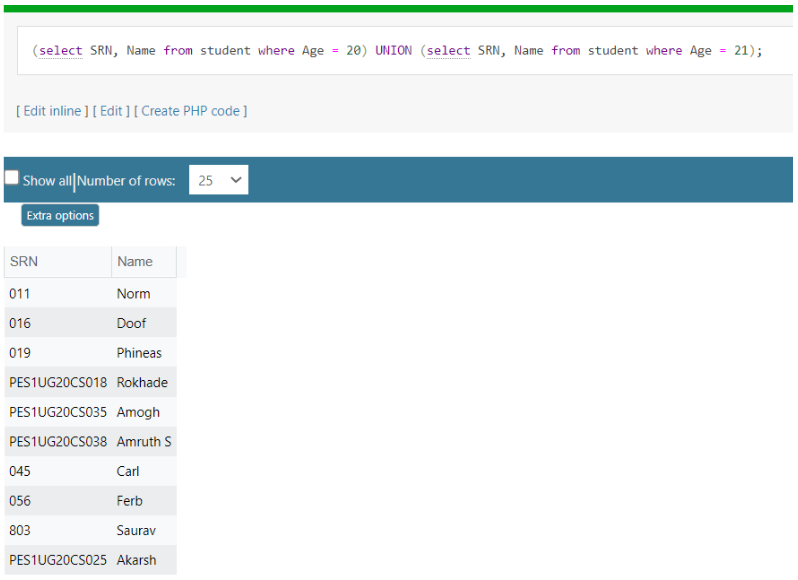
Extra options

	SRN	Name	DOB	Age	YOS	Class	max(Age)
Edit Copy Delete	011	Norm	2002-11-11	20	3	5A	21

Set Operations

1. Find the SRN, Name of students aged either 20 or 21.

   Get the SRN, Name of the student aged 20 union the SRN, Name of the students aged 21.



```
(select SRN, Name from student where Age = 20) UNION (select SRN, Name from student where Age = 21);
```

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

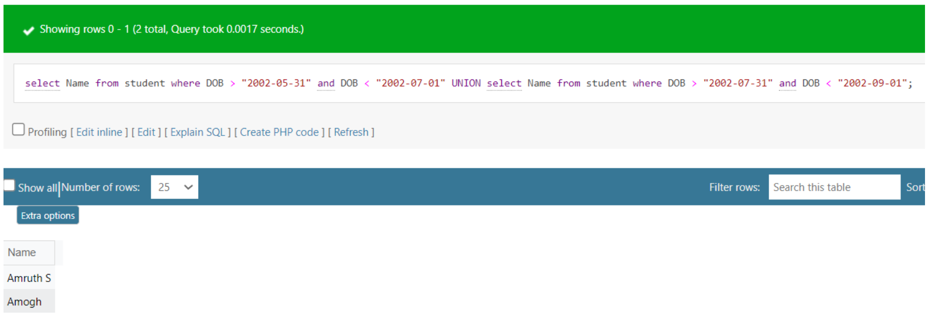
Show all | Number of rows: 25

Extra options

SRN	Name
011	Norm
016	Doof
019	Phineas
PES1UG20CS018	Rokhade
PES1UG20CS035	Amogh
PES1UG20CS038	Amruth S
045	Carl
056	Ferb
803	Saurav
PES1UG20CS025	Akarsh

2. Find students' Name born in June or August of 2002.

   Get the Name of students whose dob is greater than "31-05-2002" and less than "01-07-2002" UNION Name of students whose dob is greater than "31-07-2001" and less than less than "01-09-2001"



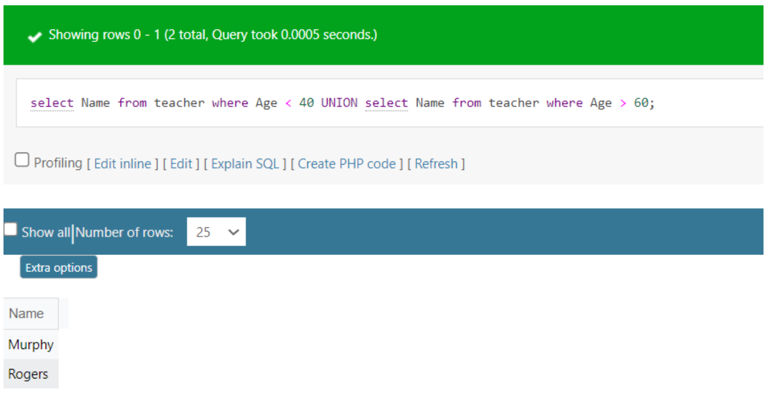
Showing rows 0 - 1 (2 total, Query took 0.0017 seconds.)

```
select Name from student where DOB > "2002-05-31" and DOB < "2002-07-01" UNION select Name from student where DOB > "2002-07-31" and DOB < "2002-09-01";
```

Name
Amruth S
Amogh

3. Find the Name and Age of teachers aged less than 40 and greater than 60.

   Get names' of teachers from teacher table whose age is less than 40 UNION the names' of teacher whose age is greater than 60.



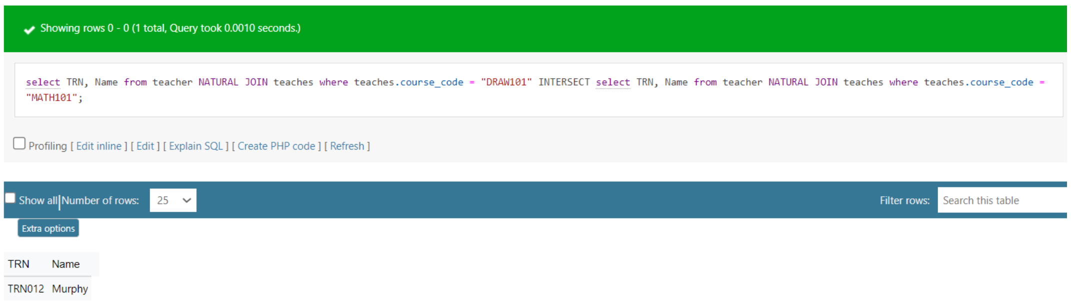
Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
select Name from teacher where Age < 40 UNION select Name from teacher where Age > 60;
```

Name
Murphy
Rogers

4. Find TRN, Name of teachers who teach both courses "MATH101" and "CHEM101"

   Get the TRN, Name from the result of the natural join of tables teacher and teaches where course_code is equal to "MATH101" intersected with the result of natural join of tables teacher and teaches where course_code is equal to "CHEM101".



Showing rows 0 - 0 (1 total, Query took 0.0010 seconds.)

```
select TRN, Name from teacher NATURAL JOIN teaches where teaches.course_code = "DRAW101" INTERSECT select TRN, Name from teacher NATURAL JOIN teaches where teaches.course_code =
"MATH101";
```

TRN	Name
TRN012	Murphy

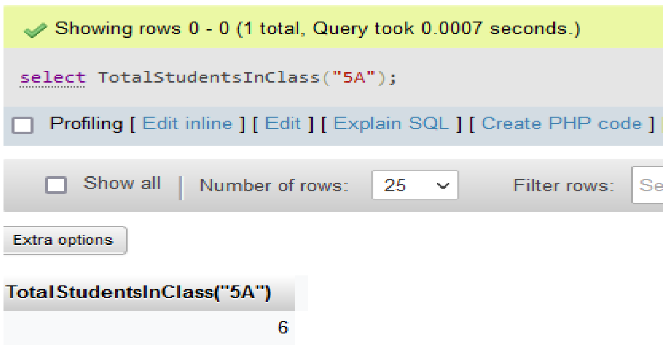
Functions and Procedures

Function.

Function to get the total number of students in a given class.

```
DELIMITER $$

CREATE FUNCTION TotalStudentsInClass(Class varchar(5))

RETURNS int

NOT DETERMINISTIC

BEGIN

DECLARE count_of_students int;

select count(*) into count_of_students from student where student.class = Class GROUP BY class;

RETURN count_of_students;

END $$

DELIMITER ;
```

Screenshot of output:



```
Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

select TotalStudentsInClass("5A");
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]
Show all | Number of rows: 25 Filter rows: Se
Extra options

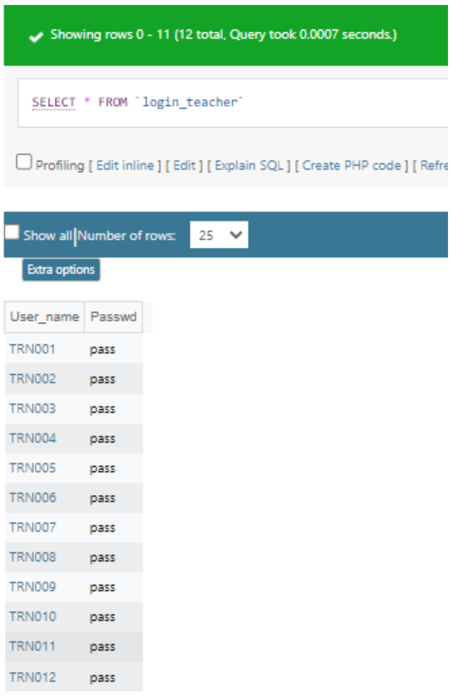
```
TotalStudentsInClass("5A")
                         6
```

Procedure

To update the password of a teacher or a student.

```
DELIMITER $$

CREATE PROCEDURE UpdatePassword(IN Username varchar(20) , IN newPassword varchar(20), IN state varchar(20))

BEGIN

if state = "Student" THEN

update login_student SET Passwd = newPassword where User_name = Username;

ELSEIF state = "Teacher" THEN

update login_teacher SET Passwd = newPassword where User_name = Username;

END IF;

END $$

DELIMITER ;
```

Before calling procedure



Showing rows 0 - 11 (12 total, Query took 0.0007 seconds.)

```
SELECT * FROM `login_teacher`
```

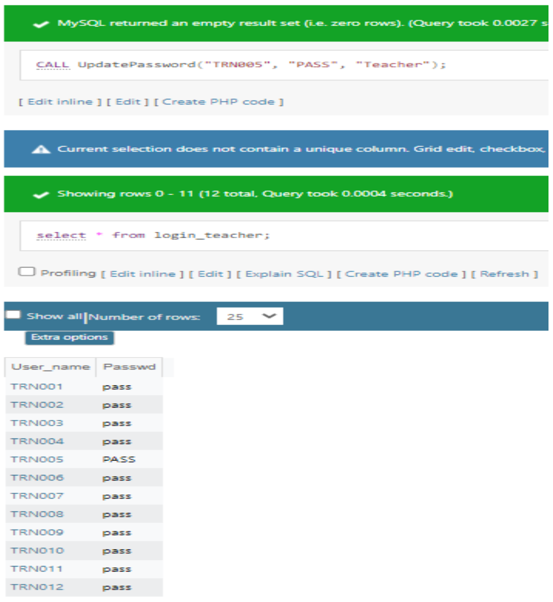
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refre

Show all | Number of rows: 25

Extra options

User_name	Passwd
TRN001	pass
TRN002	pass
TRN003	pass
TRN004	pass
TRN005	pass
TRN006	pass
TRN007	pass
TRN008	pass
TRN009	pass
TRN010	pass
TRN011	pass
TRN012	pass

After calling procedure



MySQL returned an empty result set (i.e. zero rows). (Query took 0.0027 s

```
CALL UpdatePassword("TRN005", "PASS", "Teacher");
```

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

Current selection does not contain a unique column. Grid edit, checkbox,

Showing rows 0 - 11 (12 total, Query took 0.0004 seconds.)

```
select * from login_teacher;
```

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

Show all | Number of rows: 25

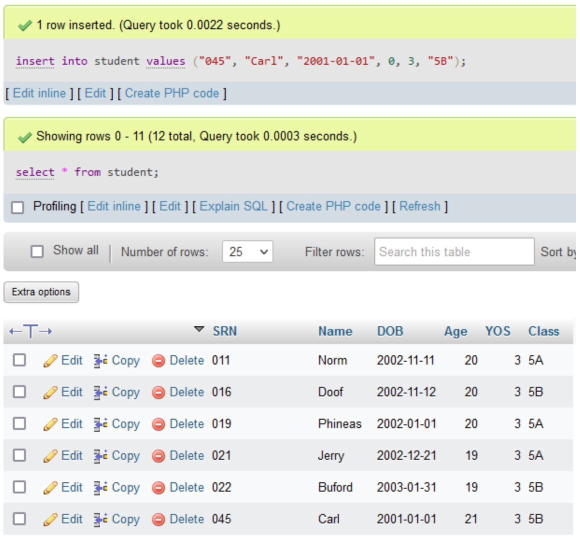
Extra options

User_name	Passwd
TRN001	pass
TRN002	pass
TRN003	pass
TRN004	pass
TRN005	PASS
TRN006	pass
TRN007	pass
TRN008	pass
TRN009	pass
TRN010	pass
TRN011	pass
TRN012	pass

Triggers and Cursors

1. Trigger to calculate age of student or teacher when a new row inserted.

```
DELIMITER $$
create trigger CalcAge before insert on student for each row
BEGIN
        SET new.age = DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(),new.DOB)), '%Y');
END $$
DELIMITER ;

DELIMITER $$
create trigger CalcAgeforteacher before insert on teacher for each row
BEGIN
        SET new.age = DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(),new.DOB)), '%Y');
END $$
DELIMITER ;
```



✓ 1 row inserted. (Query took 0.0022 seconds.)

```
insert into student values ("045", "Carl", "2001-01-01", 0, 3, "5B");
```

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

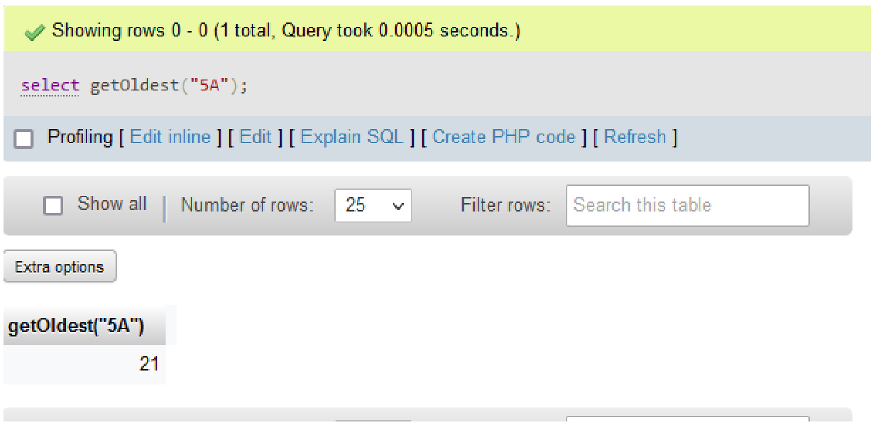
✓ Showing rows 0 - 11 (12 total, Query took 0.0003 seconds.)

```
select * from student;
```

SRN	Name	DOB	Age	YOS	Class
011	Norm	2002-11-11	20	3	5A
016	Doof	2002-11-12	20	3	5B
019	Phineas	2002-01-01	20	3	5A
021	Jerry	2002-12-21	19	3	5A
022	Buford	2003-01-31	19	3	5B
045	Carl	2001-01-01	21	3	5B

2. Cursor to find the person oldest in a class.

```
DELIMITER $$
create function getOldest(Class varchar(20))
returns int
NOT DETERMINISTIC
BEGIN
DECLARE age int default 0;
DECLARE s1 cursor for select Age from student where student.Class = Class order by Age desc;
open s1;
FETCH NEXT from s1 into age;
close s1;
return age;
END $$
DELIMITER ;
```



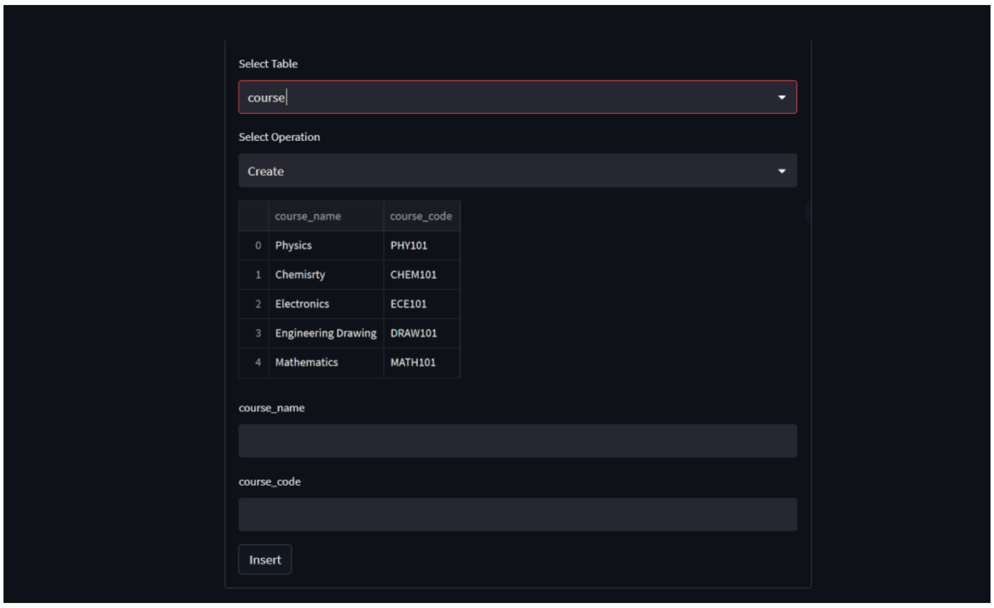
✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
select getOldest("5A");
```

getOldest("5A")
21

Developing a Frontend

1. CRUD operations.
    - Create



Select Table: course

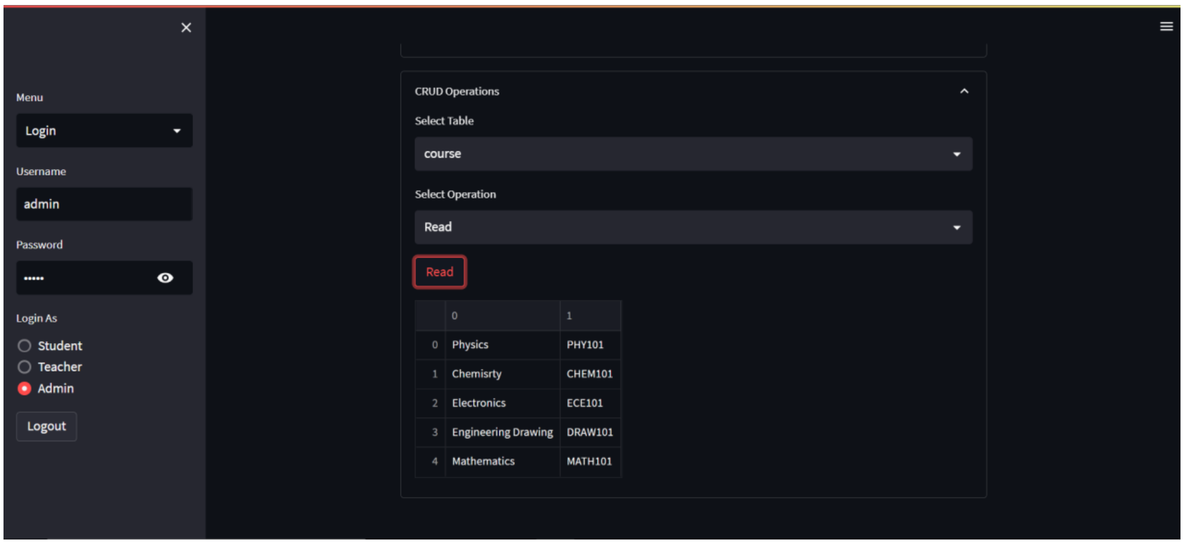
Select Operation: Create

	course_name	course_code
0	Physics	PHY101
1	Chemisrty	CHEM101
2	Electronics	ECE101
3	Engineering Drawing	DRAW101
4	Mathematics	MATH101

course_name

course_code

    - Read



Menu: Login

Username: admin

Password: •••••

Login As: ( ) Student ( ) Teacher [✓ Admin]

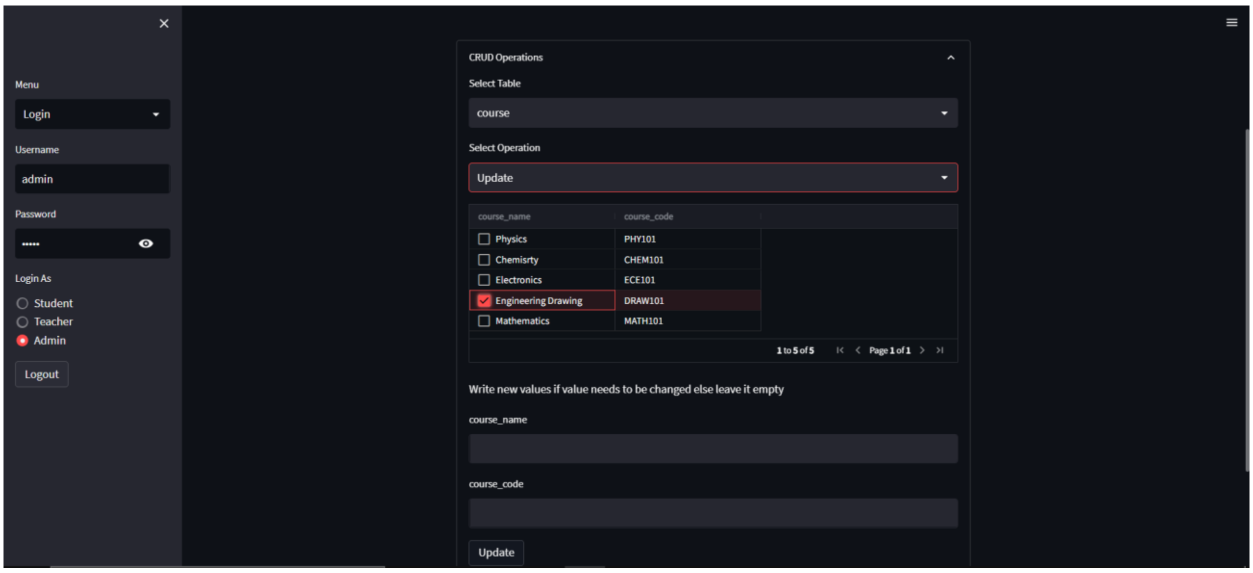
CRUD Operations

Select Table: course

Select Operation: Read

	0	1
0	Physics	PHY101
1	Chemisrty	CHEM101
2	Electronics	ECE101
3	Engineering Drawing	DRAW101
4	Mathematics	MATH101

- Update



Menu
Login
Username
admin
Password
•••••
Login As
Student
Teacher
[✓ Admin]
Logout

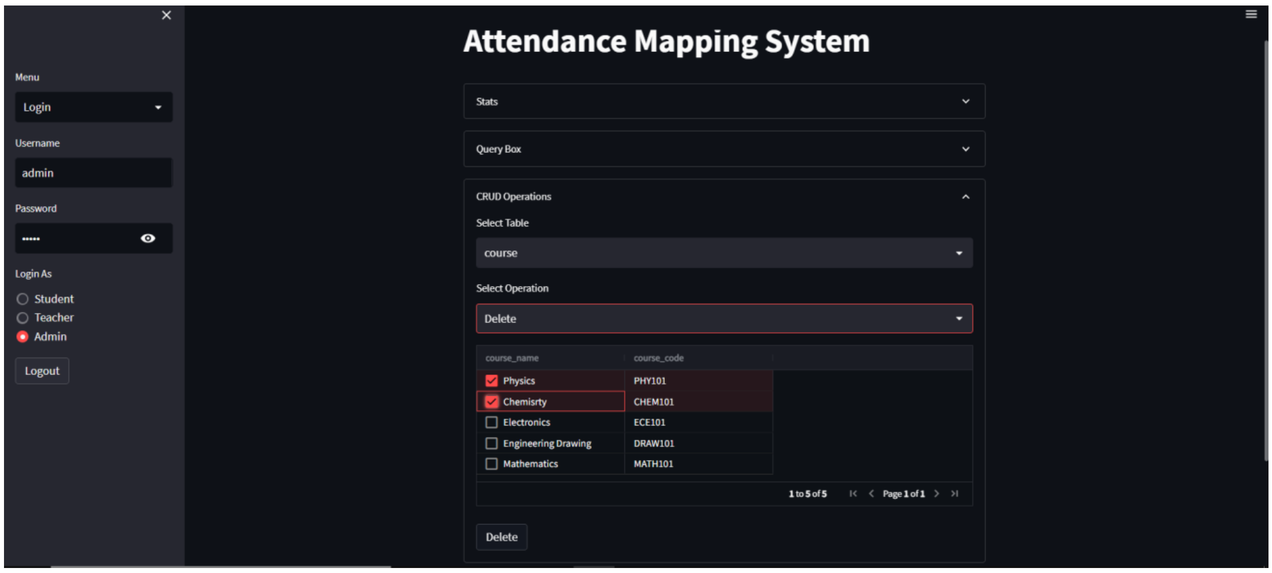
CRUD Operations
Select Table
course
Select Operation
Update

	course_name	course_code
☐	Physics	PHY101
☐	Chemisrty	CHEM101
☐	Electronics	ECE101
✓	Engineering Drawing	DRAW101
☐	Mathematics	MATH101

1 to 5 of 5 Page 1 of 1

Write new values if value needs to be changed else leave it empty
course_name
course_code
Update

- Delete



Attendance Mapping System

Menu
Login
Username
admin
Password
•••••
Login As
Student
Teacher
[✓ Admin]
Logout

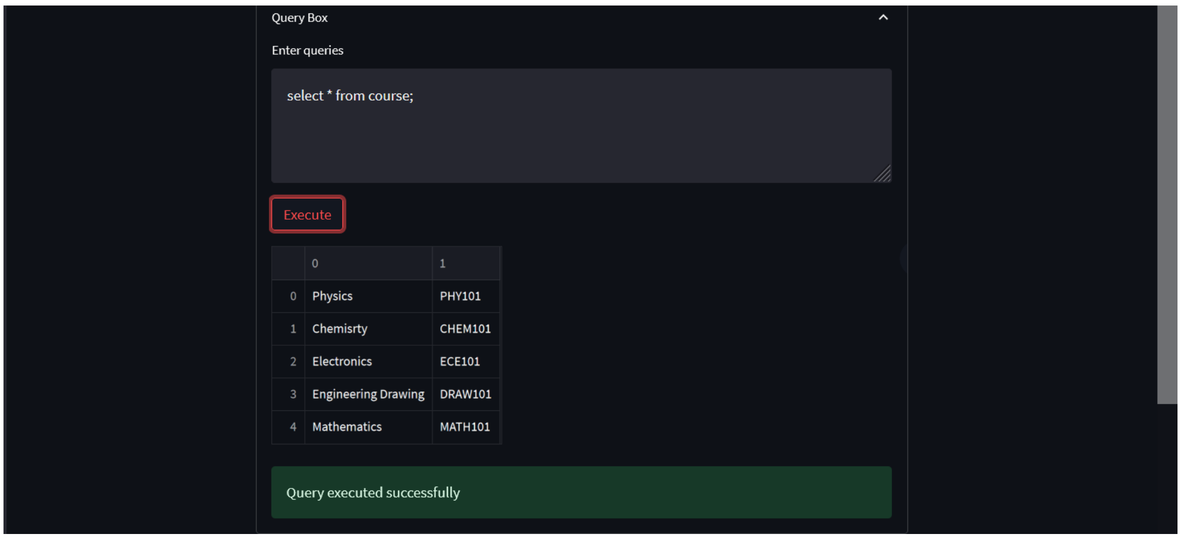
Stats
Query Box
CRUD Operations
Select Table
course
Select Operation
Delete

	course_name	course_code
✓	Physics	PHY101
✓	Chemisrty	CHEM101
☐	Electronics	ECE101
☐	Engineering Drawing	DRAW101
☐	Mathematics	MATH101

1 to 5 of 5 Page 1 of 1

Delete

2. Query Box to run queries.



Query Box
Enter queries

```
select * from course;
```

Execute

	0	1
0	Physics	PHY101
1	Chemisrty	CHEM101
2	Electronics	ECE101
3	Engineering Drawing	DRAW101
4	Mathematics	MATH101

Query executed successfully