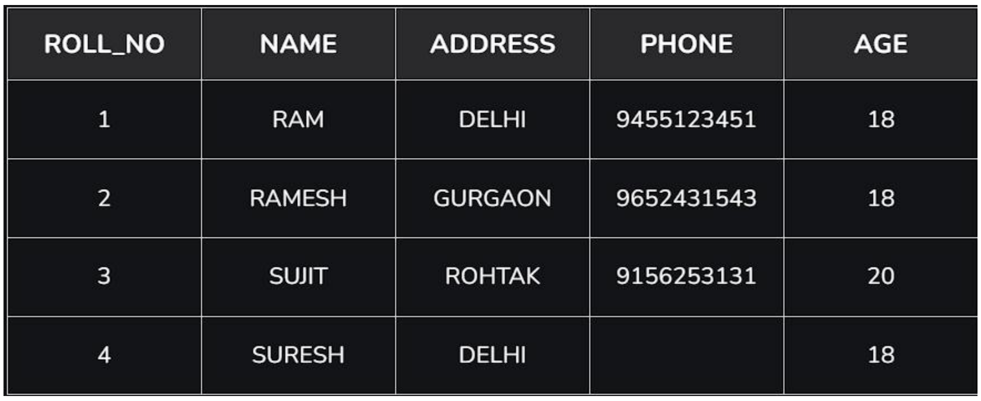
In modern enterprise computing applications, two prevalent database models are the relational database model and the Graph database model.

Relational Database Model(RDBMS):

A relational database stores related data in tables based on the relational model, where each row represents a record with a unique key. Columns hold attributes, facilitating relationships between data points. Tables, or relations, depict data and relationships. This model is record-based, widely adopted, and forms the foundation of many modern database systems.

The relational model offers several **advantages**, including simplicity, flexibility, security, data accuracy, integrity maintenance, and ease of applying operations. It provides a straightforward structure for organizing data into tables with relationships, supporting various operations like data definition, manipulation, and transaction management. However, it also has **limitations**, such as inefficiency with large databases, occasional difficulty in finding table relations, and slower query response times due to complex structures. Overall, the relational model represents data in rows and columns, ensuring distinct attribute representation and single-entity per row, forming the backbone of many database systems.

Example The relational model represents how data is stored in Relational Databases. A relational database consists of a collection of tables, each of which is assigned a unique name. Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE, and AGE shown in the table.

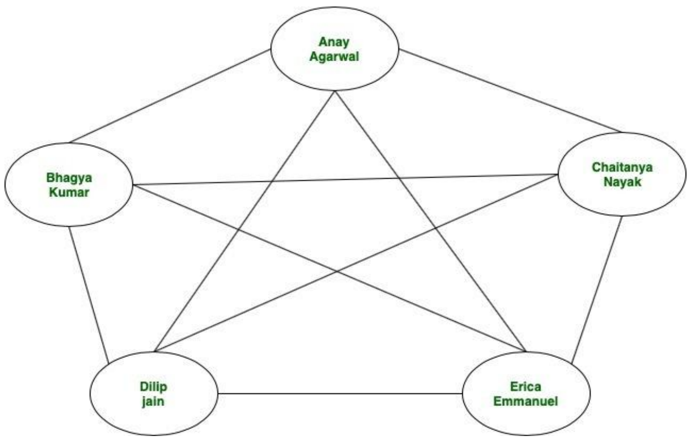


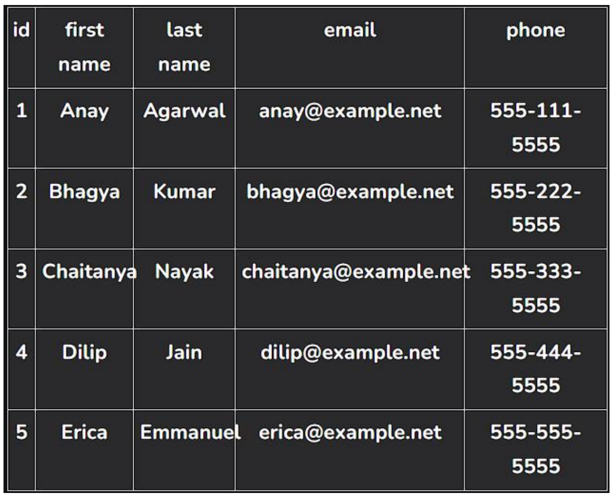
ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

Graph Database:

A graph database (GDB) is a database that uses graph structures for storing data. It uses nodes, edges, and properties instead of tables or documents to represent and store data. The edges represent relationships between the nodes. This helps in retrieving data more easily and, in many cases, with one operation. Graph databases are commonly referred to as a NoSQL.

Example We have a social network in which five friends are all connected. These friends are Anay, Bhagya, Chaitanya, Dilip, and Erica. A graph database that will store their personal information may look something like this:





id	first name	last name	email	phone
1	Anay	Agarwal	anay@example.net	555-111-5555
2	Bhagya	Kumar	bhagya@example.net	555-222-5555
3	Chaitanya	Nayak	chaitanya@example.net	555-333-5555
4	Dilip	Jain	dilip@example.net	555-444-5555
5	Erica	Emmanuel	erica@example.net	555-555-5555

Now, let's analyse the time taken in this Relational database approach. This will be approximately log(N) times where N represents the number of tuples in friendship table or number of relations. Here, the database maintains the rows in the order of id's. So, in general for 'M' no of queries, we have a time complexity of **M*log(N)** Only if we had used a graph database approach, the total time complexity would have been O(N).

Advantages: Frequent schema changes, managing volume of data, real-time query response time, and more intelligent data activation requirements are done by graph model.

Disadvantages: Note that graph databases aren't always the best solution for an application. We will need to assess the needs of application before deciding the architecture.

Limitations of Graph Databases: Graph Databases may not be offering better choice over the NoSQL variations.If application needs to scale horizontally this may introduces poor performance.

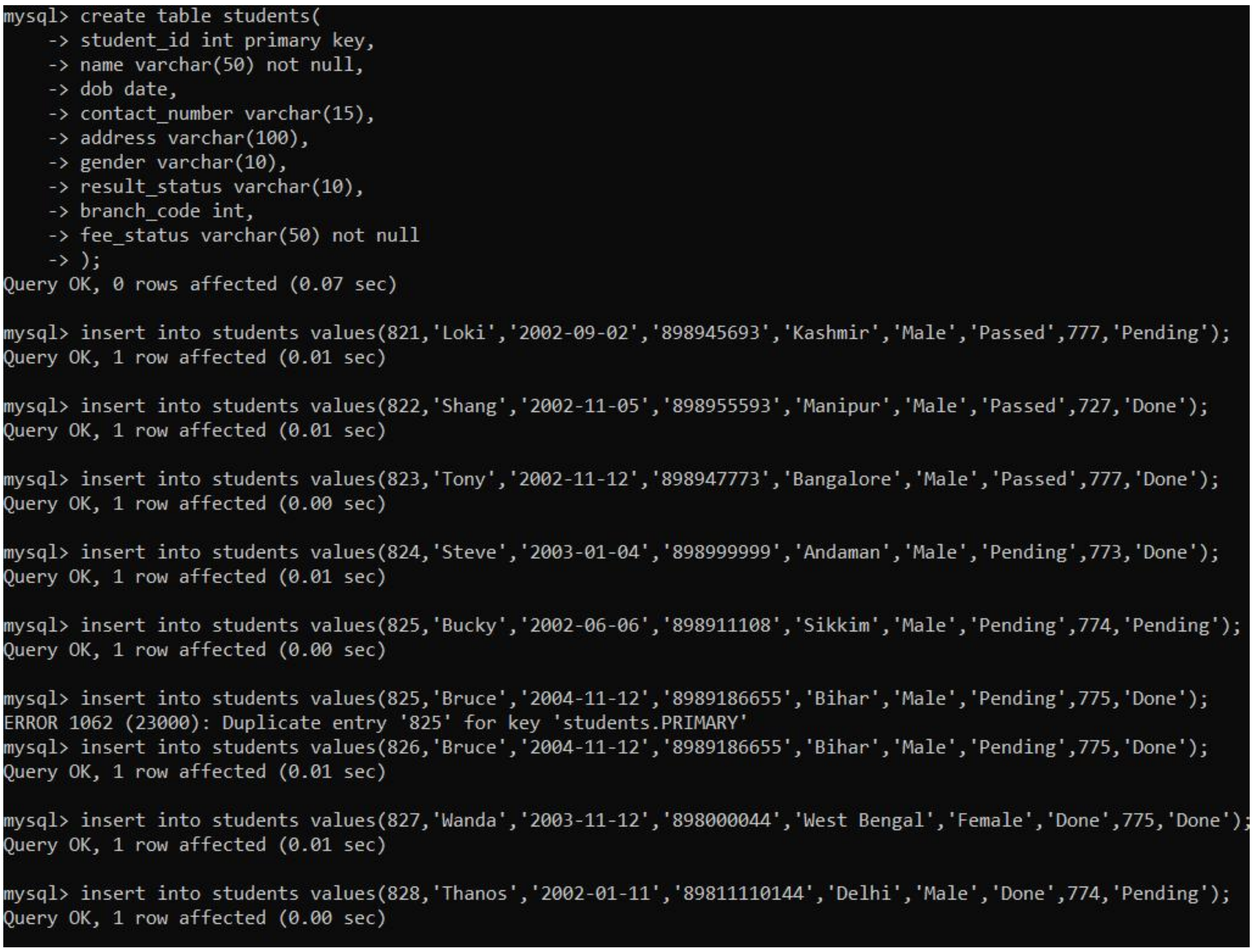
B)

List of Functional Requirements for RUAS Student Management System:

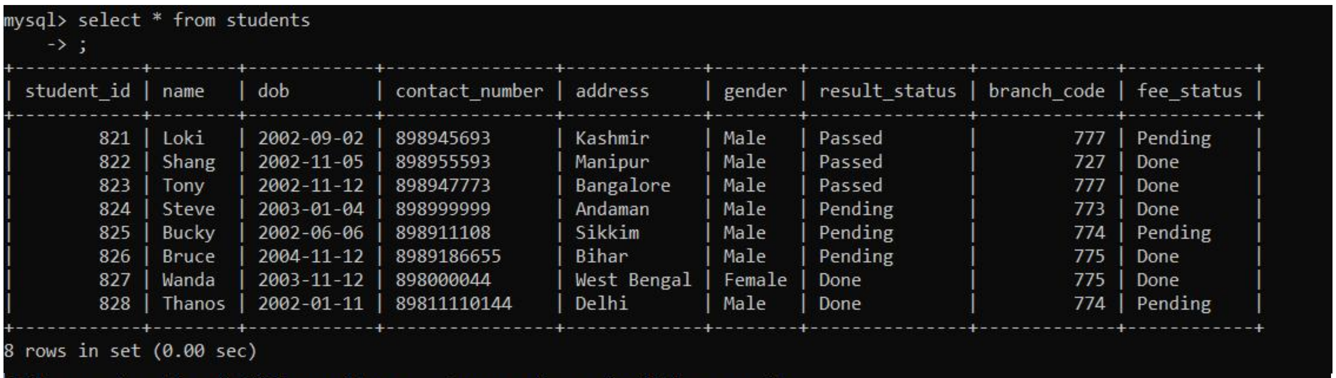
- the user should have the ability to display the list of all the students that are included in the student managemment system
- the user should have the ability to search for any specific student based on any of the parameters like name, student id, result status, fee status, etc.
- the user should have the ability to access the database.
- the user should have the ability to allow adding, updating, and deleting student records.
- the user should should have the ability to generate unique roll numbers for each student.

Implementation of Relational Database with MySQL commands:

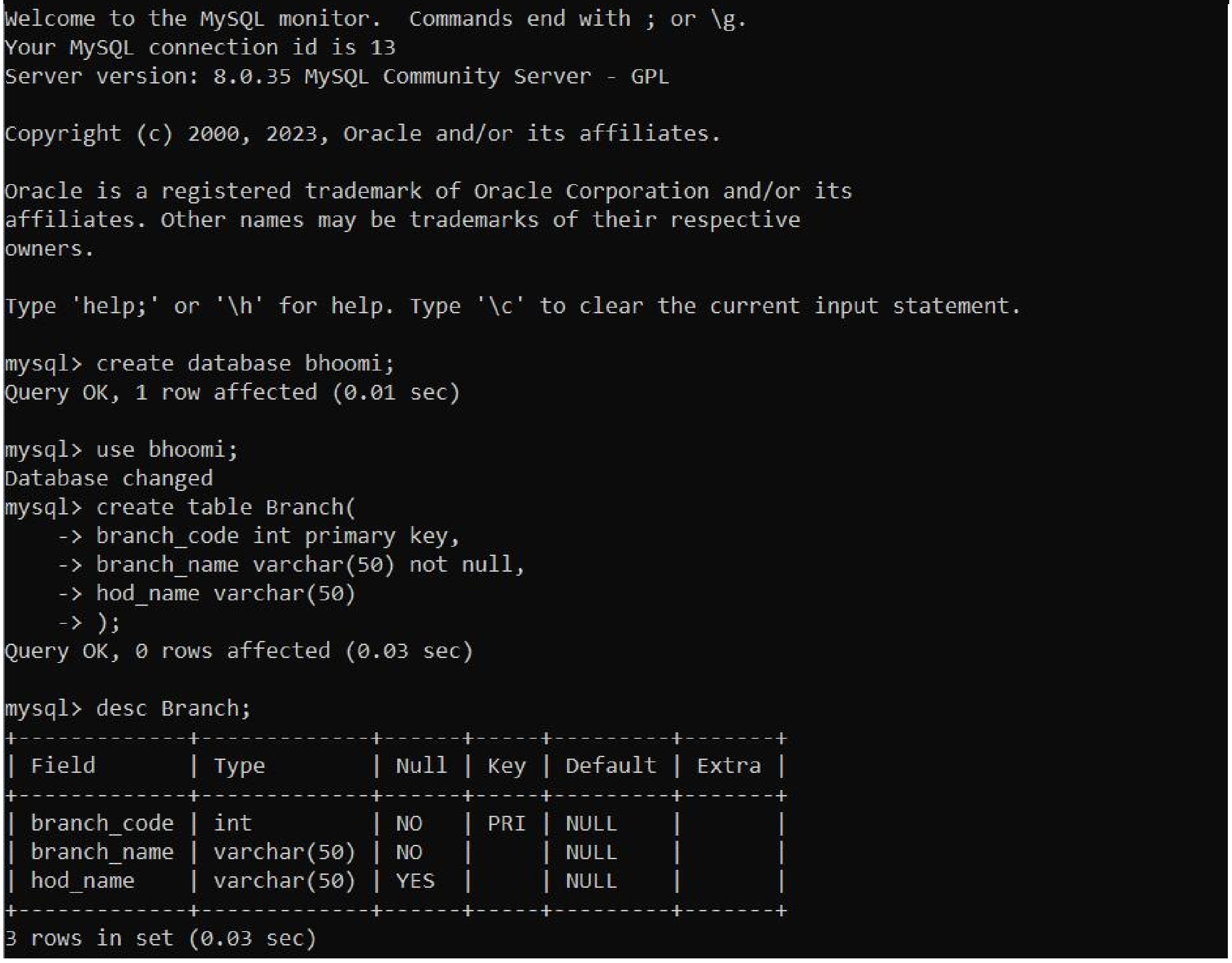
Creating tables:



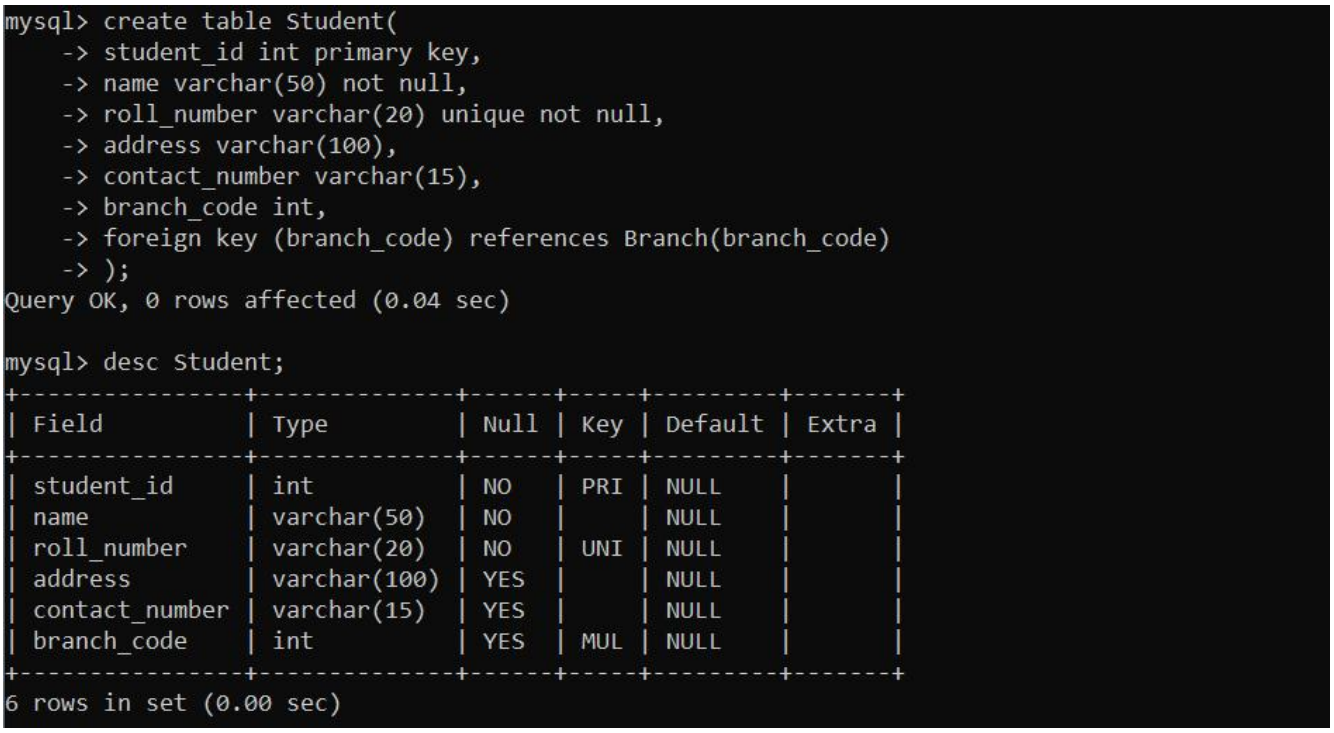
```
mysql> create table students(
    -> student_id int primary key,
    -> name varchar(50) not null,
    -> dob date,
    -> contact_number varchar(15),
    -> address varchar(100),
    -> gender varchar(10),
    -> result_status varchar(10),
    -> branch_code int,
    -> fee_status varchar(50) not null
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> insert into students values(821,'Loki','2002-09-02','898945693','Kashmir','Male','Passed',777,'Pending');
Query OK, 1 row affected (0.01 sec)

mysql> insert into students values(822,'Shang','2002-11-05','898955593','Manipur','Male','Passed',727,'Done');
Query OK, 1 row affected (0.01 sec)

mysql> insert into students values(823,'Tony','2002-11-12','898947773','Bangalore','Male','Passed',777,'Done');
Query OK, 1 row affected (0.00 sec)

mysql> insert into students values(824,'Steve','2003-01-04','898999999','Andaman','Male','Pending',773,'Done');
Query OK, 1 row affected (0.01 sec)

mysql> insert into students values(825,'Bucky','2002-06-06','898911108','Sikkim','Male','Pending',774,'Pending');
Query OK, 1 row affected (0.00 sec)

mysql> insert into students values(825,'Bruce','2004-11-12','8989186655','Bihar','Male','Pending',775,'Done');
ERROR 1062 (23000): Duplicate entry '825' for key 'students.PRIMARY'
mysql> insert into students values(826,'Bruce','2004-11-12','8989186655','Bihar','Male','Pending',775,'Done');
Query OK, 1 row affected (0.01 sec)

mysql> insert into students values(827,'Wanda','2003-11-12','898000044','West Bengal','Female','Done',775,'Done');
Query OK, 1 row affected (0.01 sec)

mysql> insert into students values(828,'Thanos','2002-01-11','89811110144','Delhi','Male','Done',774,'Pending');
Query OK, 1 row affected (0.00 sec)
```



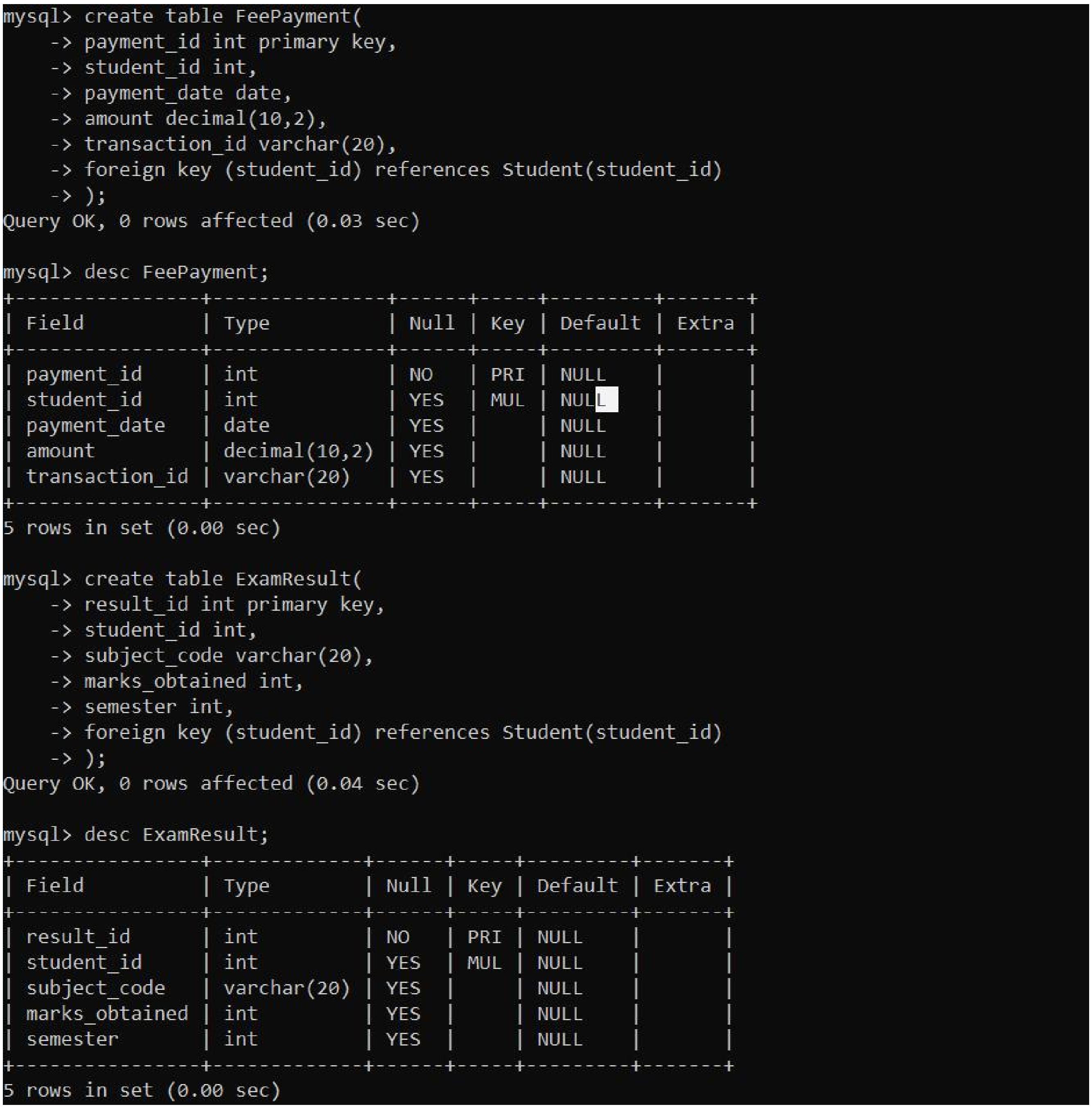
```
mysql> select * from students
    -> ;
+------------+--------+------------+----------------+-------------+--------+---------------+-------------+------------+
| student_id | name   | dob        | contact_number | address     | gender | result_status | branch_code | fee_status |
+------------+--------+------------+----------------+-------------+--------+---------------+-------------+------------+
|        821 | Loki   | 2002-09-02 | 898945693      | Kashmir     | Male   | Passed        |         777 | Pending    |
|        822 | Shang  | 2002-11-05 | 898955593      | Manipur     | Male   | Passed        |         727 | Done       |
|        823 | Tony   | 2002-11-12 | 898947773      | Bangalore   | Male   | Passed        |         777 | Done       |
|        824 | Steve  | 2003-01-04 | 898999999      | Andaman     | Male   | Pending       |         773 | Done       |
|        825 | Bucky  | 2002-06-06 | 898911108      | Sikkim      | Male   | Pending       |         774 | Pending    |
|        826 | Bruce  | 2004-11-12 | 8989186655     | Bihar       | Male   | Pending       |         775 | Done       |
|        827 | Wanda  | 2003-11-12 | 898000044      | West Bengal | Female | Done          |         775 | Done       |
|        828 | Thanos | 2002-01-11 | 89811110144    | Delhi       | Male   | Done          |         774 | Pending    |
+------------+--------+------------+----------------+-------------+--------+---------------+-------------+------------+
8 rows in set (0.00 sec)
```



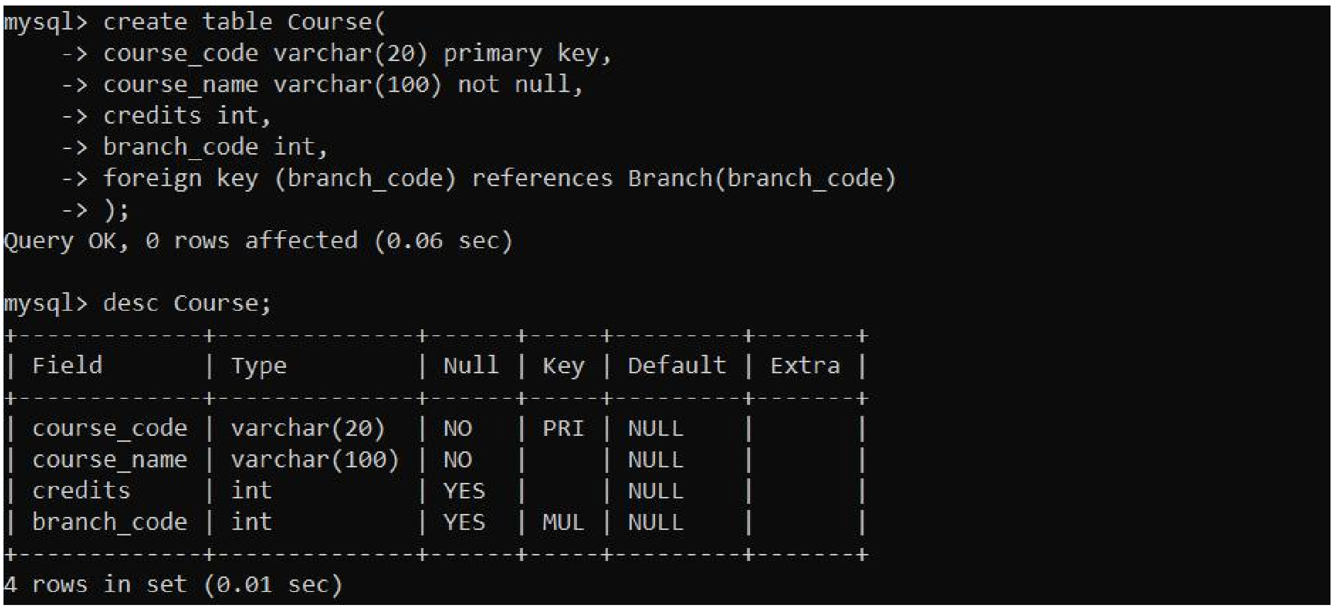
```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database bhoomi;
Query OK, 1 row affected (0.01 sec)

mysql> use bhoomi;
Database changed
mysql> create table Branch(
    -> branch_code int primary key,
    -> branch_name varchar(50) not null,
    -> hod_name varchar(50)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Branch;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| branch_code | int         | NO   | PRI | NULL    |       |
| branch_name | varchar(50) | NO   |     | NULL    |       |
| hod_name    | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```



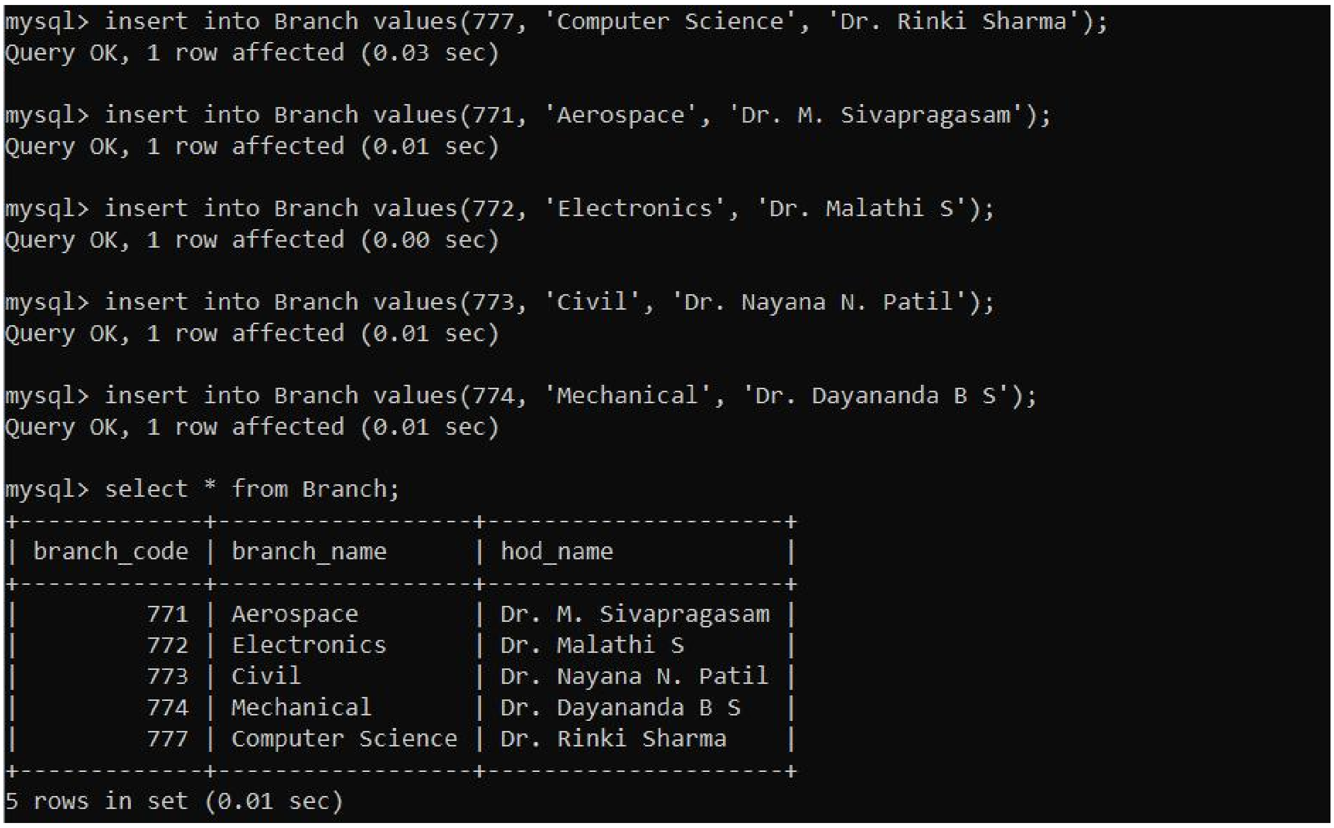
```
mysql> create table Student(
    -> student_id int primary key,
    -> name varchar(50) not null,
    -> roll_number varchar(20) unique not null,
    -> address varchar(100),
    -> contact_number varchar(15),
    -> branch_code int,
    -> foreign key (branch_code) references Branch(branch_code)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc Student;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| student_id     | int          | NO   | PRI | NULL    |       |
| name           | varchar(50)  | NO   |     | NULL    |       |
| roll_number    | varchar(20)  | NO   | UNI | NULL    |       |
| address        | varchar(100) | YES  |     | NULL    |       |
| contact_number | varchar(15)  | YES  |     | NULL    |       |
| branch_code    | int          | YES  | MUL | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```



```
mysql> create table FeePayment(
    -> payment_id int primary key,
    -> student_id int,
    -> payment_date date,
    -> amount decimal(10,2),
    -> transaction_id varchar(20),
    -> foreign key (student_id) references Student(student_id)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc FeePayment;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| payment_id     | int           | NO   | PRI | NULL    |       |
| student_id     | int           | YES  | MUL | NULL    |       |
| payment_date   | date          | YES  |     | NULL    |       |
| amount         | decimal(10,2) | YES  |     | NULL    |       |
| transaction_id | varchar(20)   | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> create table ExamResult(
    -> result_id int primary key,
    -> student_id int,
    -> subject_code varchar(20),
    -> marks_obtained int,
    -> semester int,
    -> foreign key (student_id) references Student(student_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc ExamResult;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| result_id      | int         | NO   | PRI | NULL    |       |
| student_id     | int         | YES  | MUL | NULL    |       |
| subject_code   | varchar(20) | YES  |     | NULL    |       |
| marks_obtained | int         | YES  |     | NULL    |       |
| semester       | int         | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

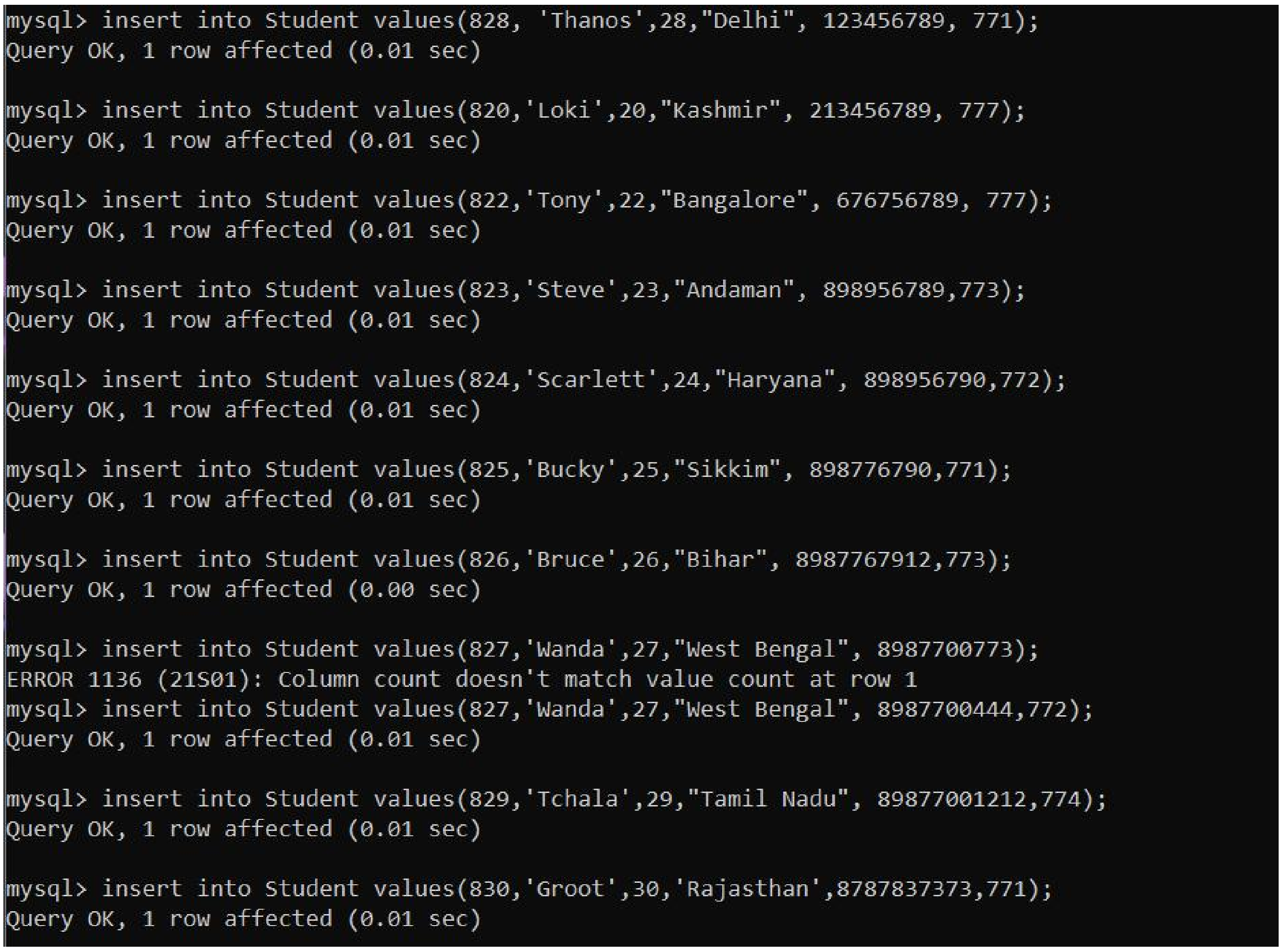


```
mysql> create table Course(
    -> course_code varchar(20) primary key,
    -> course_name varchar(100) not null,
    -> credits int,
    -> branch_code int,
    -> foreign key (branch_code) references Branch(branch_code)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc Course;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| course_code | varchar(20)  | NO   | PRI | NULL    |       |
| course_name | varchar(100) | NO   |     | NULL    |       |
| credits     | int          | YES  |     | NULL    |       |
| branch_code | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

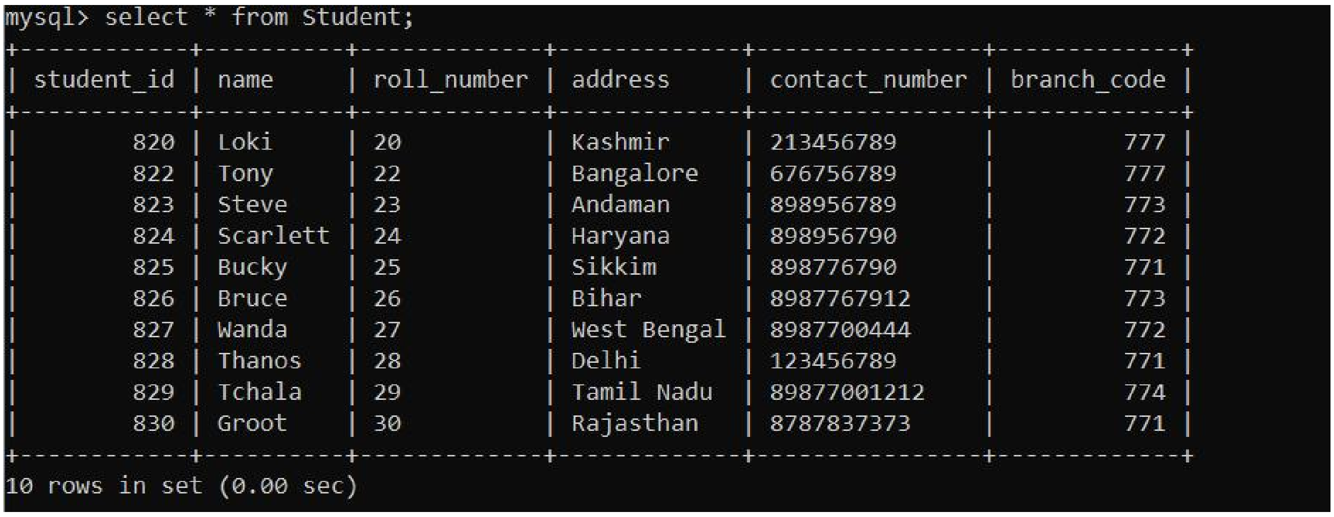


```
mysql> insert into Branch values(777, 'Computer Science', 'Dr. Rinki Sharma');
Query OK, 1 row affected (0.03 sec)

mysql> insert into Branch values(771, 'Aerospace', 'Dr. M. Sivapragasam');
Query OK, 1 row affected (0.01 sec)

mysql> insert into Branch values(772, 'Electronics', 'Dr. Malathi S');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Branch values(773, 'Civil', 'Dr. Nayana N. Patil');
Query OK, 1 row affected (0.01 sec)

mysql> insert into Branch values(774, 'Mechanical', 'Dr. Dayananda B S');
Query OK, 1 row affected (0.01 sec)

mysql> select * from Branch;
+-------------+------------------+---------------------+
| branch_code | branch_name      | hod_name            |
+-------------+------------------+---------------------+
|         771 | Aerospace        | Dr. M. Sivapragasam |
|         772 | Electronics      | Dr. Malathi S       |
|         773 | Civil            | Dr. Nayana N. Patil |
|         774 | Mechanical       | Dr. Dayananda B S   |
|         777 | Computer Science | Dr. Rinki Sharma    |
+-------------+------------------+---------------------+
5 rows in set (0.01 sec)
```

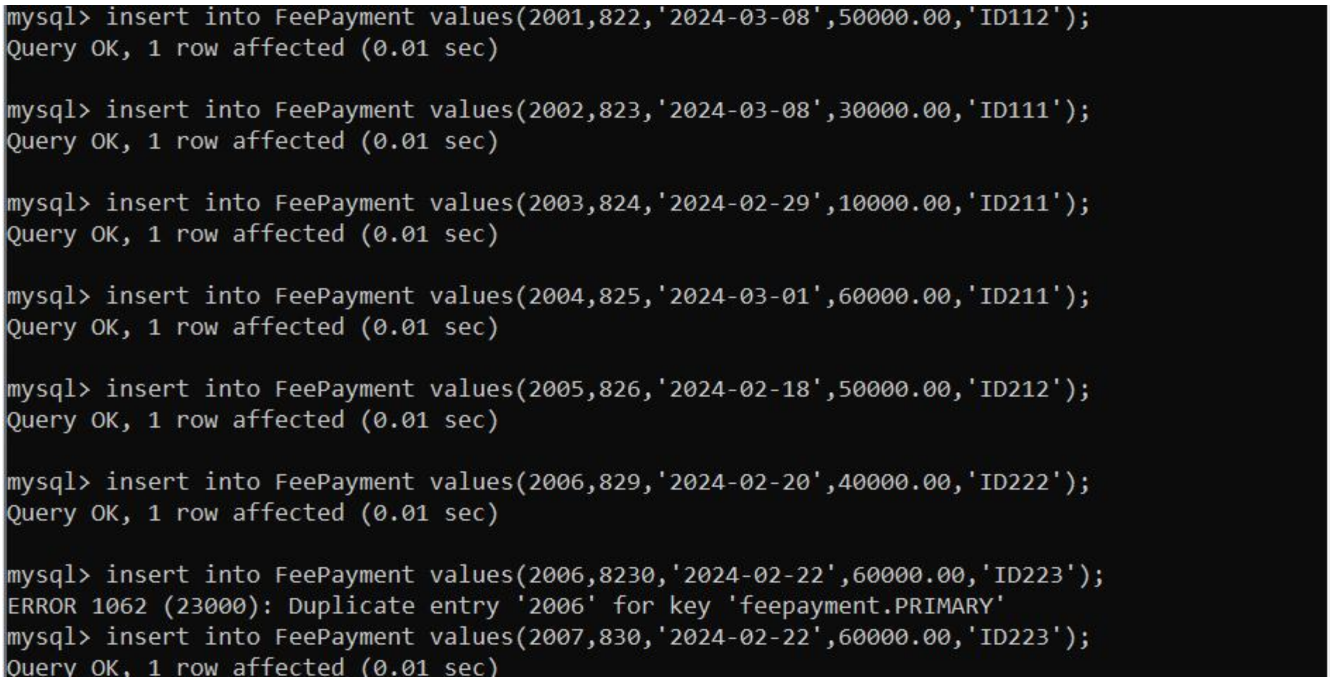
Inserting tuples:



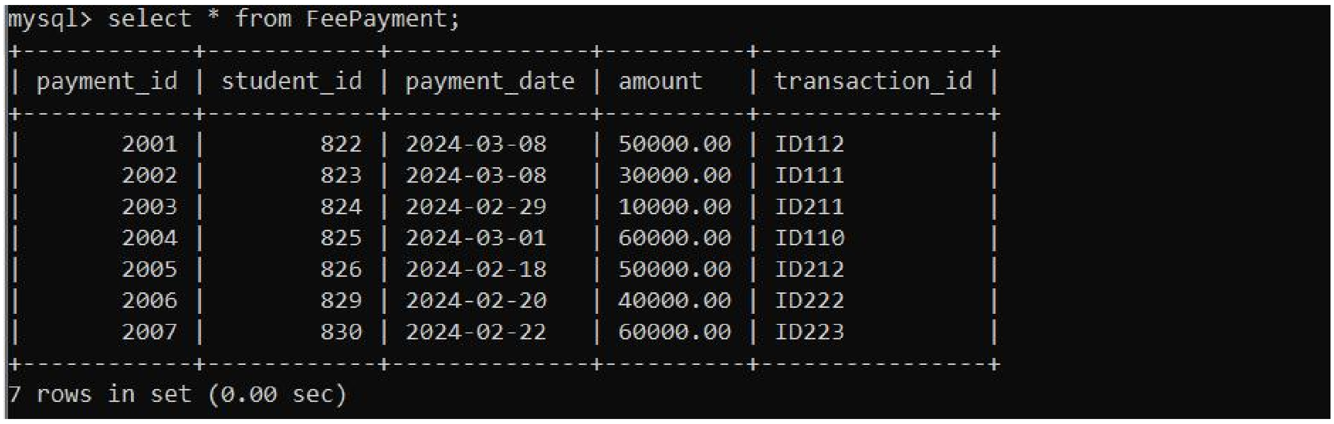
```
mysql> insert into Student values(828, 'Thanos',28,"Delhi", 123456789, 771);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(820,'Loki',20,"Kashmir", 213456789, 777);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(822,'Tony',22,"Bangalore", 676756789, 777);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(823,'Steve',23,"Andaman", 898956789,773);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(824,'Scarlett',24,"Haryana", 898956790,772);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(825,'Bucky',25,"Sikkim", 898776790,771);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(826,'Bruce',26,"Bihar", 8987767912,773);
Query OK, 1 row affected (0.00 sec)

mysql> insert into Student values(827,'Wanda',27,"West Bengal", 8987700773);
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> insert into Student values(827,'Wanda',27,"West Bengal", 8987700444,772);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(829,'Tchala',29,"Tamil Nadu", 89877001212,774);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(830,'Groot',30,'Rajasthan',8787837373,771);
Query OK, 1 row affected (0.01 sec)
```



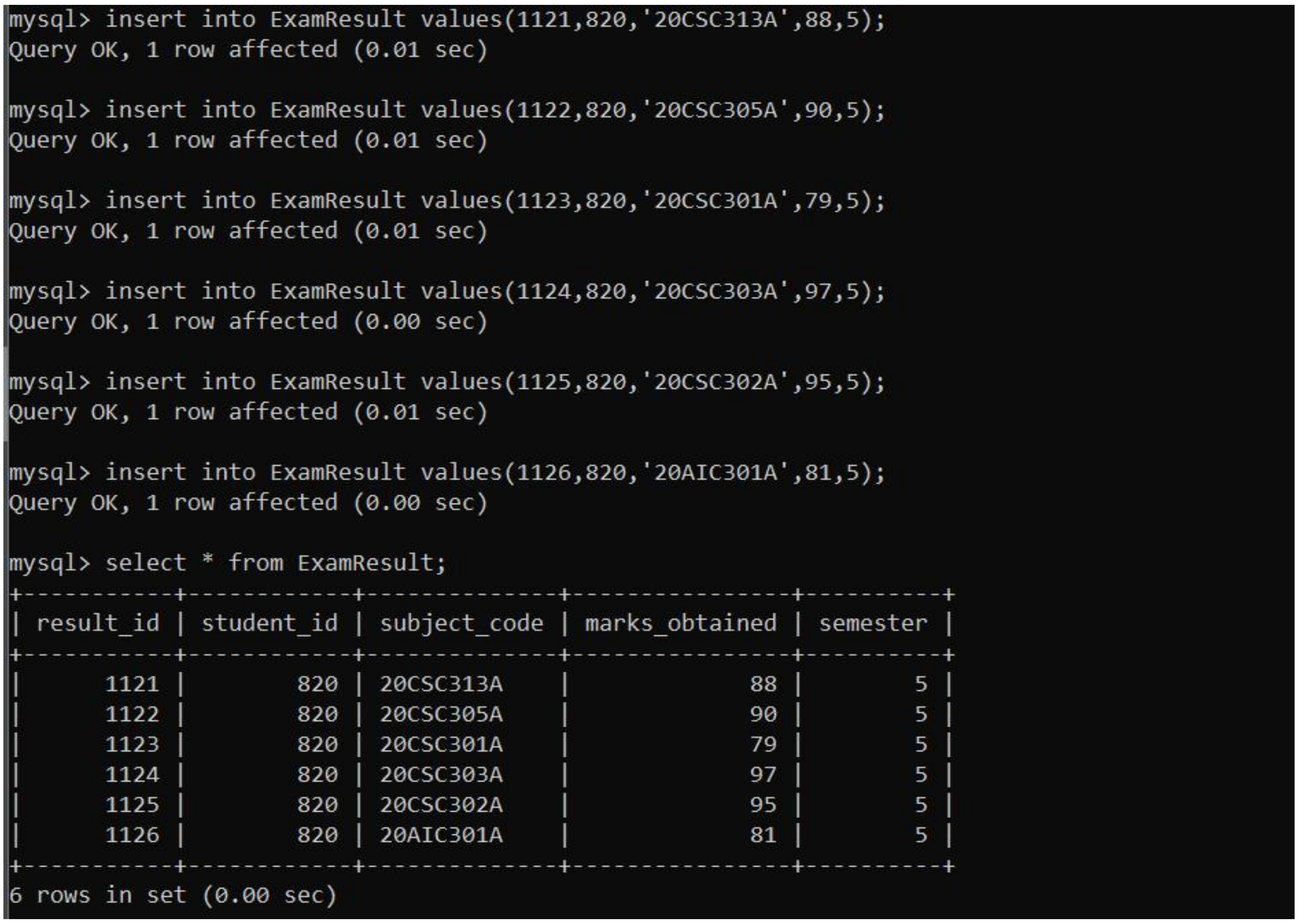
```
mysql> select * from Student;
+------------+----------+-------------+-------------+----------------+-------------+
| student_id | name     | roll_number | address     | contact_number | branch_code |
+------------+----------+-------------+-------------+----------------+-------------+
|        820 | Loki     | 20          | Kashmir     | 213456789      |         777 |
|        822 | Tony     | 22          | Bangalore   | 676756789      |         777 |
|        823 | Steve    | 23          | Andaman     | 898956789      |         773 |
|        824 | Scarlett | 24          | Haryana     | 898956790      |         772 |
|        825 | Bucky    | 25          | Sikkim      | 898776790      |         771 |
|        826 | Bruce    | 26          | Bihar       | 8987767912     |         773 |
|        827 | Wanda    | 27          | West Bengal | 8987700444     |         772 |
|        828 | Thanos   | 28          | Delhi       | 123456789      |         771 |
|        829 | Tchala   | 29          | Tamil Nadu  | 89877001212    |         774 |
|        830 | Groot    | 30          | Rajasthan   | 8787837373     |         771 |
+------------+----------+-------------+-------------+----------------+-------------+
10 rows in set (0.00 sec)
```



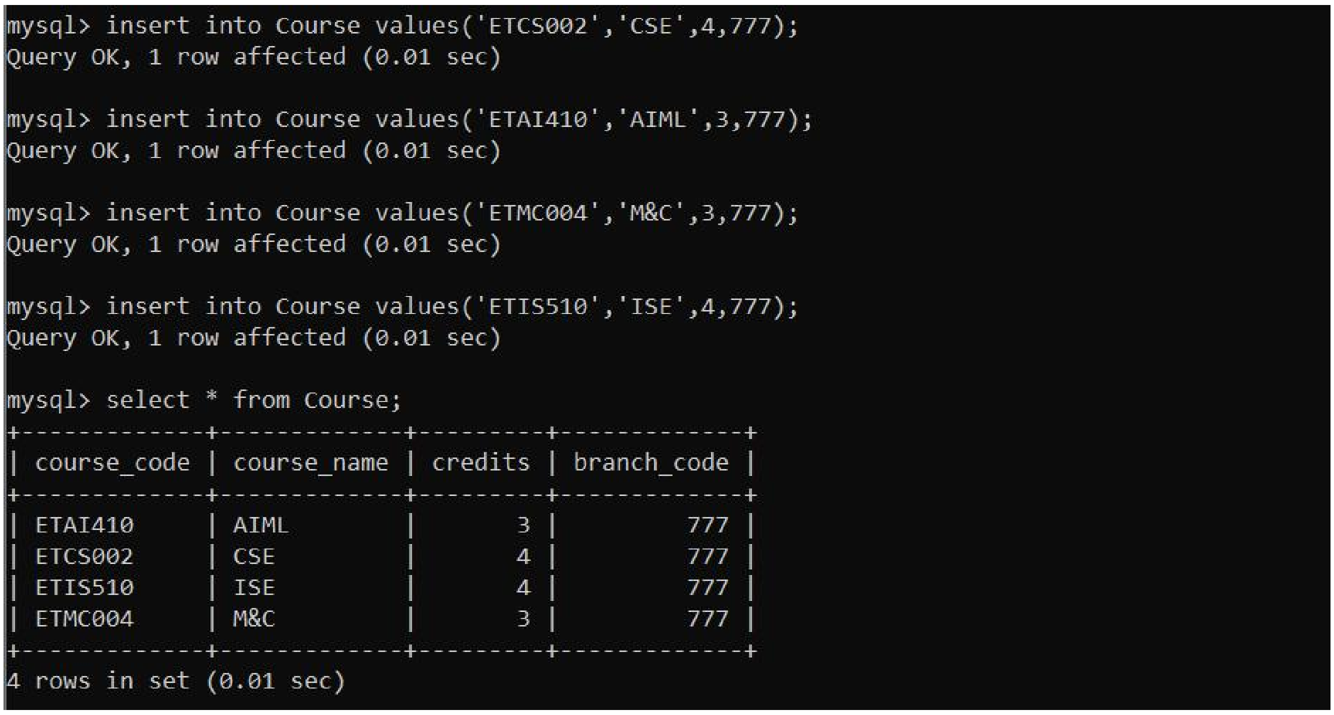
```
mysql> insert into FeePayment values(2001,822,'2024-03-08',50000.00,'ID112');
Query OK, 1 row affected (0.01 sec)

mysql> insert into FeePayment values(2002,823,'2024-03-08',30000.00,'ID111');
Query OK, 1 row affected (0.01 sec)

mysql> insert into FeePayment values(2003,824,'2024-02-29',10000.00,'ID211');
Query OK, 1 row affected (0.01 sec)

mysql> insert into FeePayment values(2004,825,'2024-03-01',60000.00,'ID211');
Query OK, 1 row affected (0.01 sec)

mysql> insert into FeePayment values(2005,826,'2024-02-18',50000.00,'ID212');
Query OK, 1 row affected (0.01 sec)

mysql> insert into FeePayment values(2006,829,'2024-02-20',40000.00,'ID222');
Query OK, 1 row affected (0.01 sec)

mysql> insert into FeePayment values(2006,8230,'2024-02-22',60000.00,'ID223');
ERROR 1062 (23000): Duplicate entry '2006' for key 'feepayment.PRIMARY'
mysql> insert into FeePayment values(2007,830,'2024-02-22',60000.00,'ID223');
Query OK, 1 row affected (0.01 sec)
```



```
mysql> select * from FeePayment;
+------------+------------+--------------+----------+----------------+
| payment_id | student_id | payment_date | amount   | transaction_id |
+------------+------------+--------------+----------+----------------+
|       2001 |        822 | 2024-03-08   | 50000.00 | ID112          |
|       2002 |        823 | 2024-03-08   | 30000.00 | ID111          |
|       2003 |        824 | 2024-02-29   | 10000.00 | ID211          |
|       2004 |        825 | 2024-03-01   | 60000.00 | ID110          |
|       2005 |        826 | 2024-02-18   | 50000.00 | ID212          |
|       2006 |        829 | 2024-02-20   | 40000.00 | ID222          |
|       2007 |        830 | 2024-02-22   | 60000.00 | ID223          |
+------------+------------+--------------+----------+----------------+
7 rows in set (0.00 sec)
```

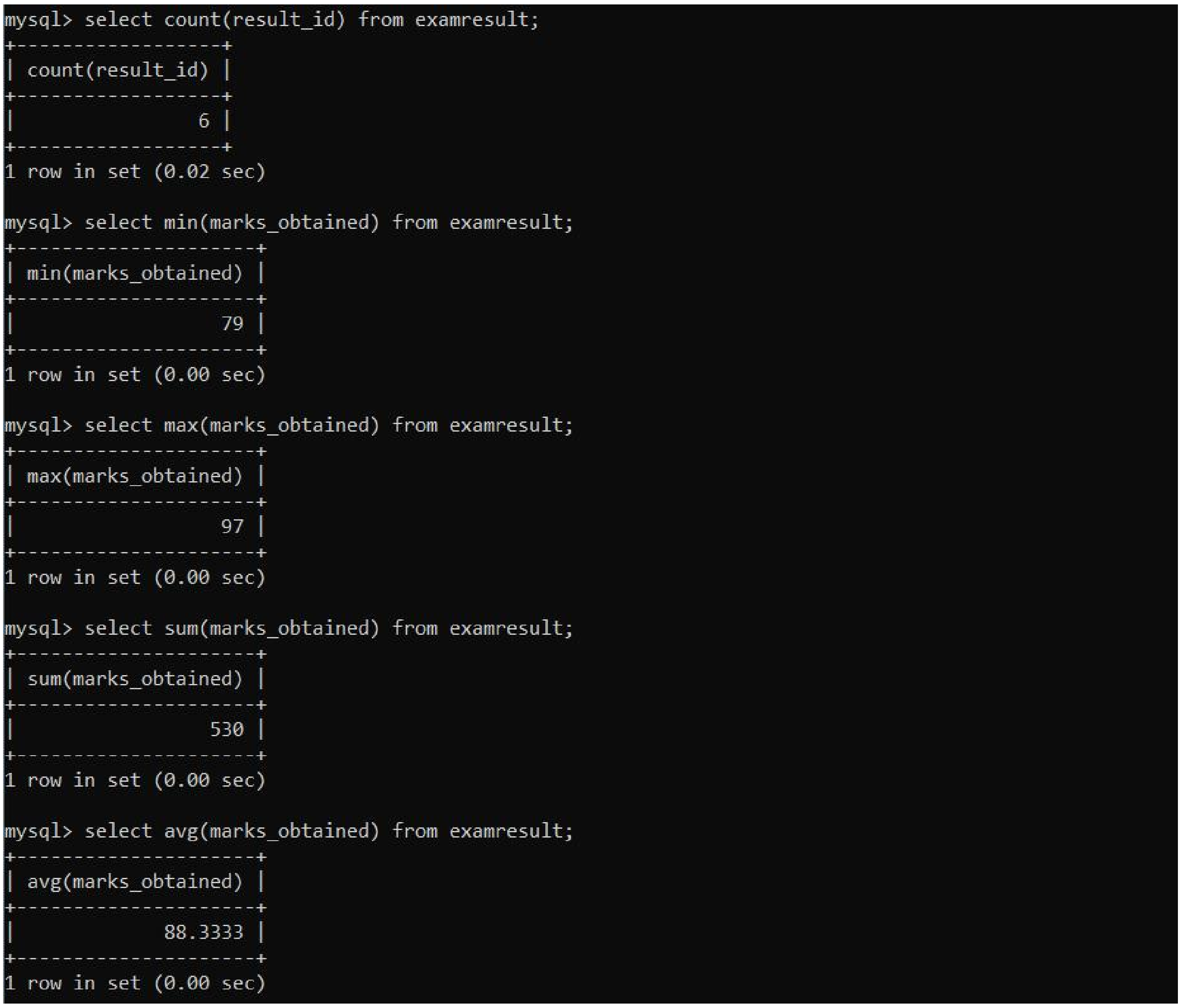


```
mysql> insert into ExamResult values(1121,820,'20CSC313A',88,5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into ExamResult values(1122,820,'20CSC305A',90,5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into ExamResult values(1123,820,'20CSC301A',79,5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into ExamResult values(1124,820,'20CSC303A',97,5);
Query OK, 1 row affected (0.00 sec)

mysql> insert into ExamResult values(1125,820,'20CSC302A',95,5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into ExamResult values(1126,820,'20AIC301A',81,5);
Query OK, 1 row affected (0.00 sec)

mysql> select * from ExamResult;
+-----------+------------+--------------+----------------+----------+
| result_id | student_id | subject_code | marks_obtained | semester |
+-----------+------------+--------------+----------------+----------+
|      1121 |        820 | 20CSC313A    |             88 |        5 |
|      1122 |        820 | 20CSC305A    |             90 |        5 |
|      1123 |        820 | 20CSC301A    |             79 |        5 |
|      1124 |        820 | 20CSC303A    |             97 |        5 |
|      1125 |        820 | 20CSC302A    |             95 |        5 |
|      1126 |        820 | 20AIC301A    |             81 |        5 |
+-----------+------------+--------------+----------------+----------+
6 rows in set (0.00 sec)
```

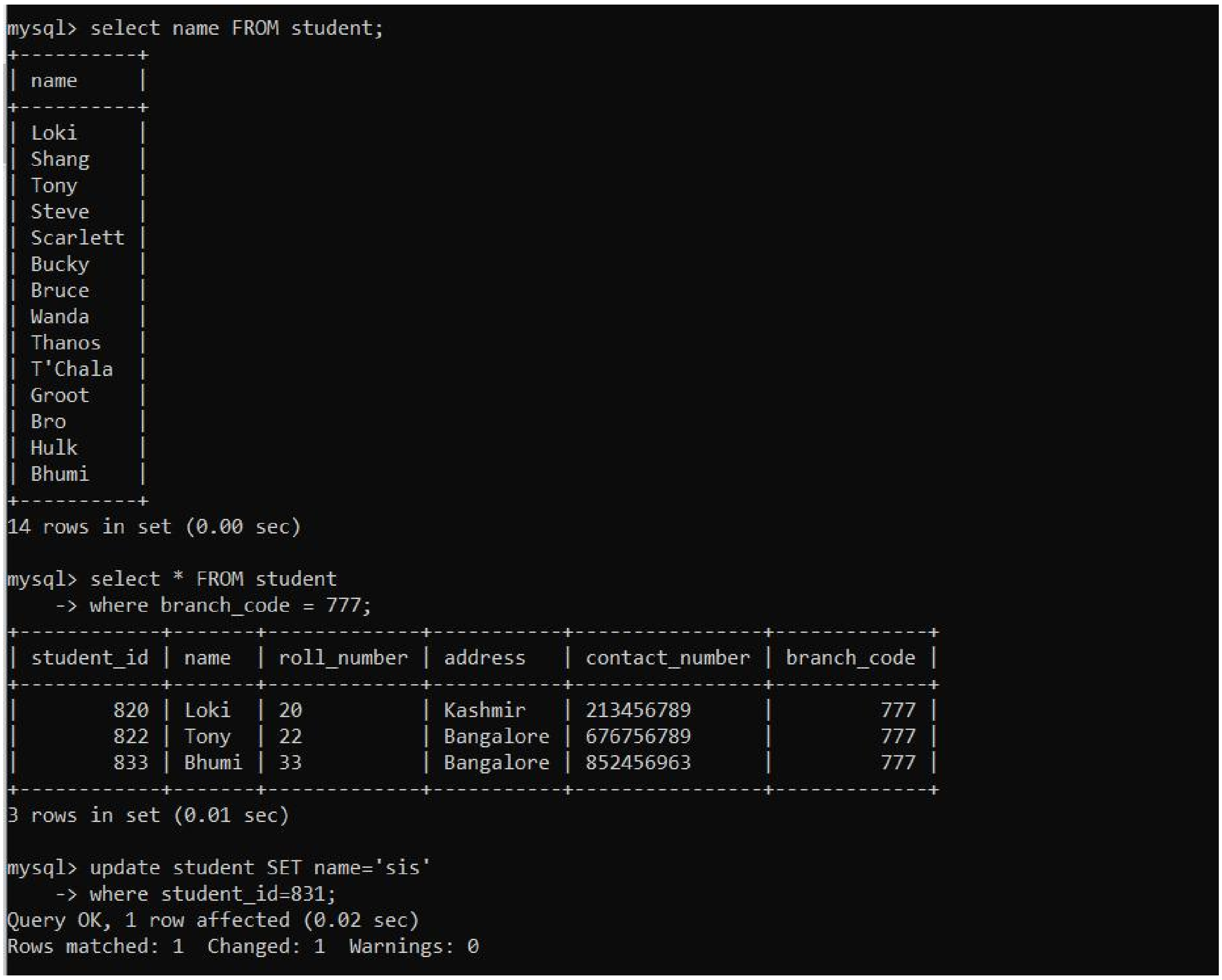


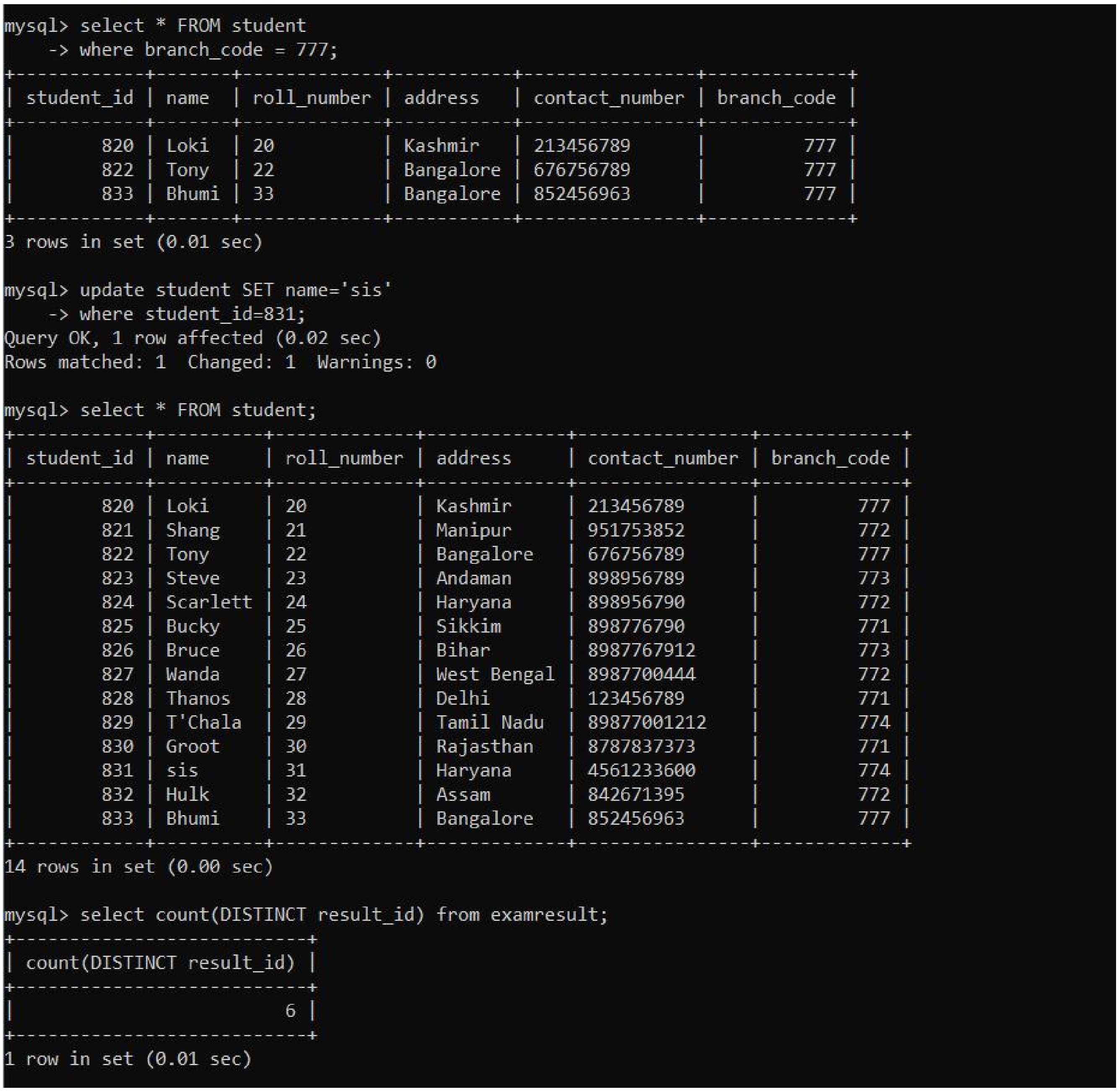
```
mysql> insert into Course values('ETCS002','CSE',4,777);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Course values('ETAI410','AIML',3,777);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Course values('ETMC004','M&C',3,777);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Course values('ETIS510','ISE',4,777);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Course;
+-------------+-------------+---------+-------------+
| course_code | course_name | credits | branch_code |
+-------------+-------------+---------+-------------+
| ETAI410     | AIML        |       3 |         777 |
| ETCS002     | CSE         |       4 |         777 |
| ETIS510     | ISE         |       4 |         777 |
| ETMC004     | M&C         |       3 |         777 |
+-------------+-------------+---------+-------------+
4 rows in set (0.01 sec)
```

Performing different MySql Operations on tables:

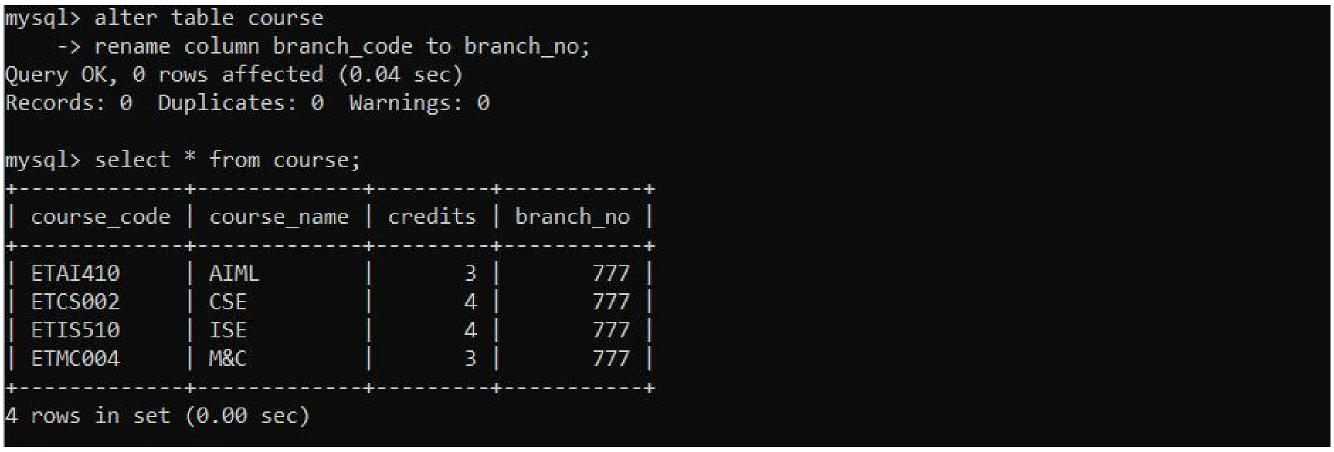


```
mysql> select count(result_id) from examresult;
+------------------+
| count(result_id) |
+------------------+
|                6 |
+------------------+
1 row in set (0.02 sec)

mysql> select min(marks_obtained) from examresult;
+---------------------+
| min(marks_obtained) |
+---------------------+
|                  79 |
+---------------------+
1 row in set (0.00 sec)

mysql> select max(marks_obtained) from examresult;
+---------------------+
| max(marks_obtained) |
+---------------------+
|                  97 |
+---------------------+
1 row in set (0.00 sec)

mysql> select sum(marks_obtained) from examresult;
+---------------------+
| sum(marks_obtained) |
+---------------------+
|                 530 |
+---------------------+
1 row in set (0.00 sec)

mysql> select avg(marks_obtained) from examresult;
+---------------------+
| avg(marks_obtained) |
+---------------------+
|             88.3333 |
+---------------------+
1 row in set (0.00 sec)
```

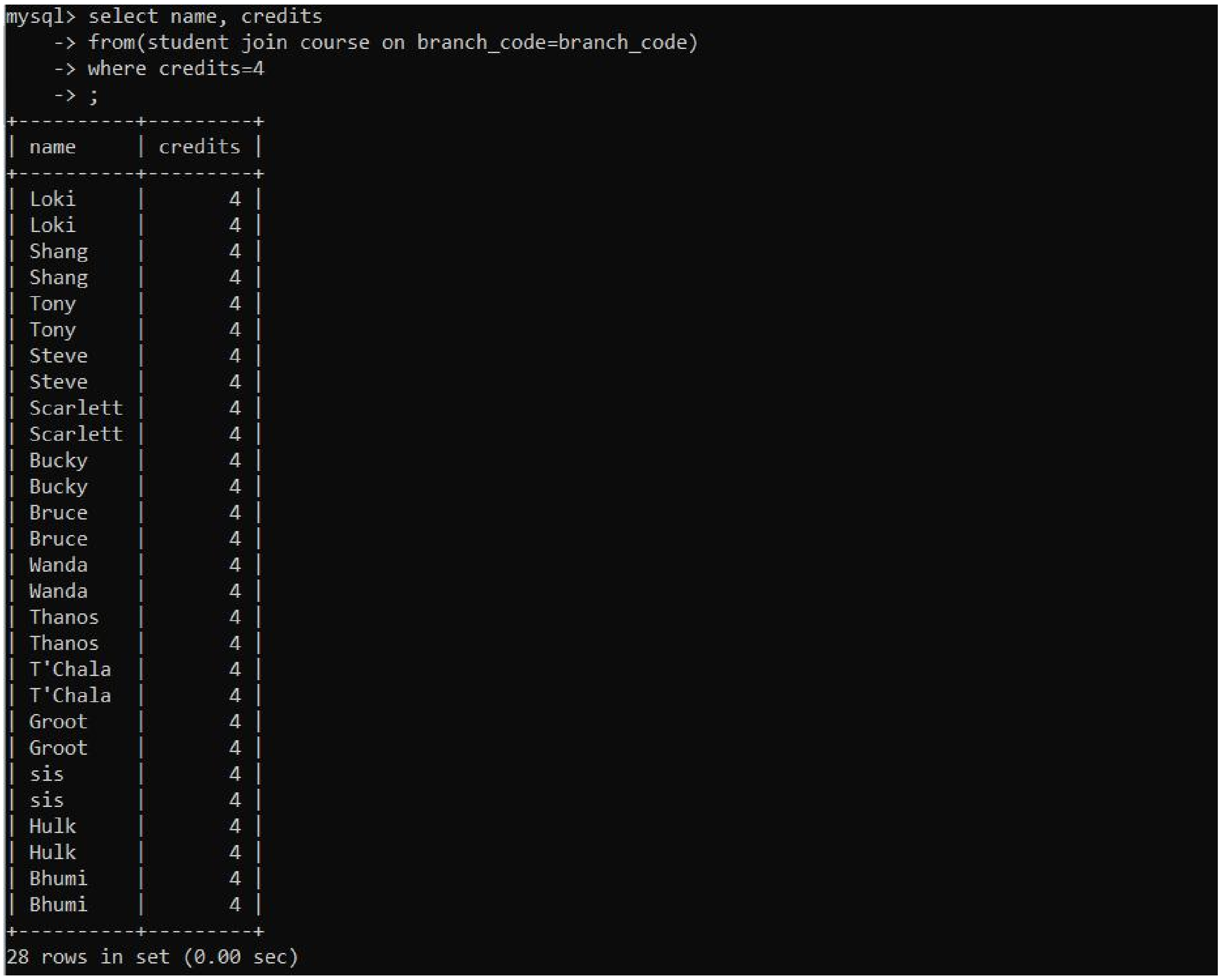

```
mysql> select name FROM student;
+----------+
| name     |
+----------+
| Loki     |
| Shang    |
| Tony     |
| Steve    |
| Scarlett |
| Bucky    |
| Bruce    |
| Wanda    |
| Thanos   |
| T'Chala  |
| Groot    |
| Bro      |
| Hulk     |
| Bhumi    |
+----------+
14 rows in set (0.00 sec)

mysql> select * FROM student
    -> where branch_code = 777;
+------------+-------+-------------+-----------+----------------+-------------+
| student_id | name  | roll_number | address   | contact_number | branch_code |
+------------+-------+-------------+-----------+----------------+-------------+
|        820 | Loki  | 20          | Kashmir   | 213456789      |         777 |
|        822 | Tony  | 22          | Bangalore | 676756789      |         777 |
|        833 | Bhumi | 33          | Bangalore | 852456963      |         777 |
+------------+-------+-------------+-----------+----------------+-------------+
3 rows in set (0.01 sec)

mysql> update student SET name='sis'
    -> where student_id=831;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```



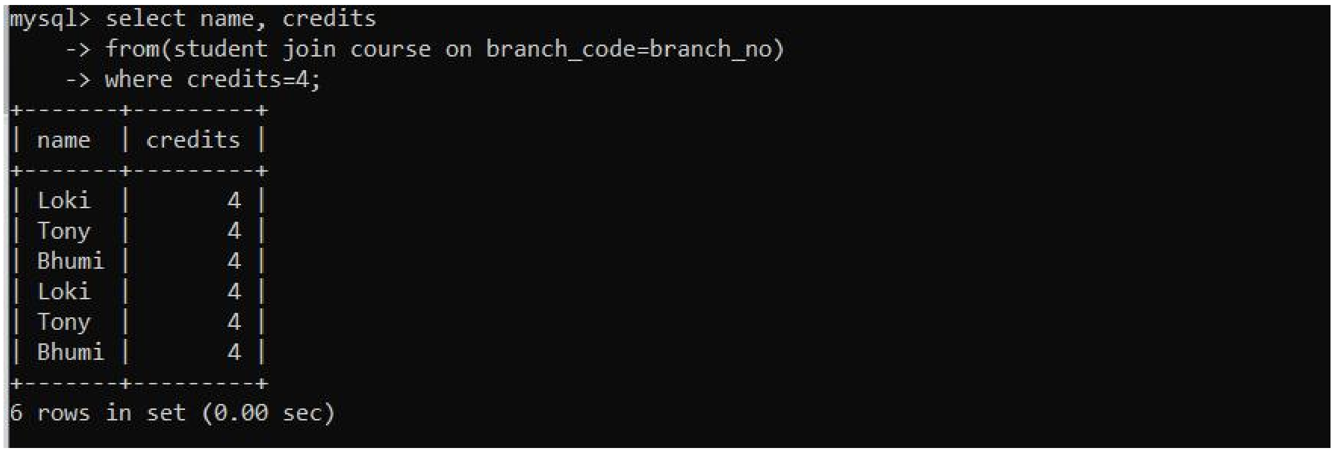
```
mysql> select * FROM student
    -> where branch_code = 777;
+------------+-------+-------------+-----------+----------------+-------------+
| student_id | name  | roll_number | address   | contact_number | branch_code |
+------------+-------+-------------+-----------+----------------+-------------+
|        820 | Loki  | 20          | Kashmir   | 213456789      |         777 |
|        822 | Tony  | 22          | Bangalore | 676756789      |         777 |
|        833 | Bhumi | 33          | Bangalore | 852456963      |         777 |
+------------+-------+-------------+-----------+----------------+-------------+
3 rows in set (0.01 sec)

mysql> update student SET name='sis'
    -> where student_id=831;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * FROM student;
+------------+----------+-------------+-------------+----------------+-------------+
| student_id | name     | roll_number | address     | contact_number | branch_code |
+------------+----------+-------------+-------------+----------------+-------------+
|        820 | Loki     | 20          | Kashmir     | 213456789      |         777 |
|        821 | Shang    | 21          | Manipur     | 951753852      |         772 |
|        822 | Tony     | 22          | Bangalore   | 676756789      |         777 |
|        823 | Steve    | 23          | Andaman     | 898956789      |         773 |
|        824 | Scarlett | 24          | Haryana     | 898956790      |         772 |
|        825 | Bucky    | 25          | Sikkim      | 898776790      |         771 |
|        826 | Bruce    | 26          | Bihar       | 8987767912     |         773 |
|        827 | Wanda    | 27          | West Bengal | 8987700444     |         772 |
|        828 | Thanos   | 28          | Delhi       | 123456789      |         771 |
|        829 | T'Chala  | 29          | Tamil Nadu  | 89877001212    |         774 |
|        830 | Groot    | 30          | Rajasthan   | 8787837373     |         771 |
|        831 | sis      | 31          | Haryana     | 4561233600     |         774 |
|        832 | Hulk     | 32          | Assam       | 842671395      |         772 |
|        833 | Bhumi    | 33          | Bangalore   | 852456963      |         777 |
+------------+----------+-------------+-------------+----------------+-------------+
14 rows in set (0.00 sec)

mysql> select count(DISTINCT result_id) from examresult;
+---------------------------+
| count(DISTINCT result_id) |
+---------------------------+
|                         6 |
+---------------------------+
1 row in set (0.01 sec)
```



```
mysql> alter table course
    -> rename column branch_code to branch_no;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from course;
+-------------+-------------+---------+-----------+
| course_code | course_name | credits | branch_no |
+-------------+-------------+---------+-----------+
| ETAI410     | AIML        |       3 |       777 |
| ETCS002     | CSE         |       4 |       777 |
| ETIS510     | ISE         |       4 |       777 |
| ETMC004     | M&C         |       3 |       777 |
+-------------+-------------+---------+-----------+
4 rows in set (0.00 sec)
```

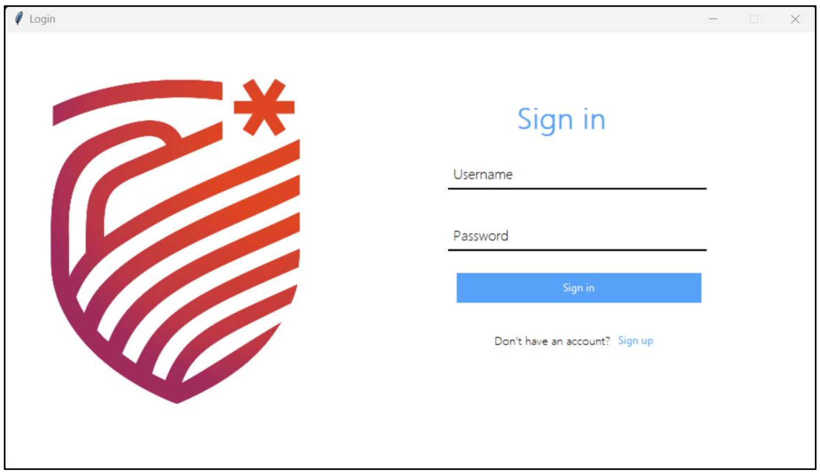


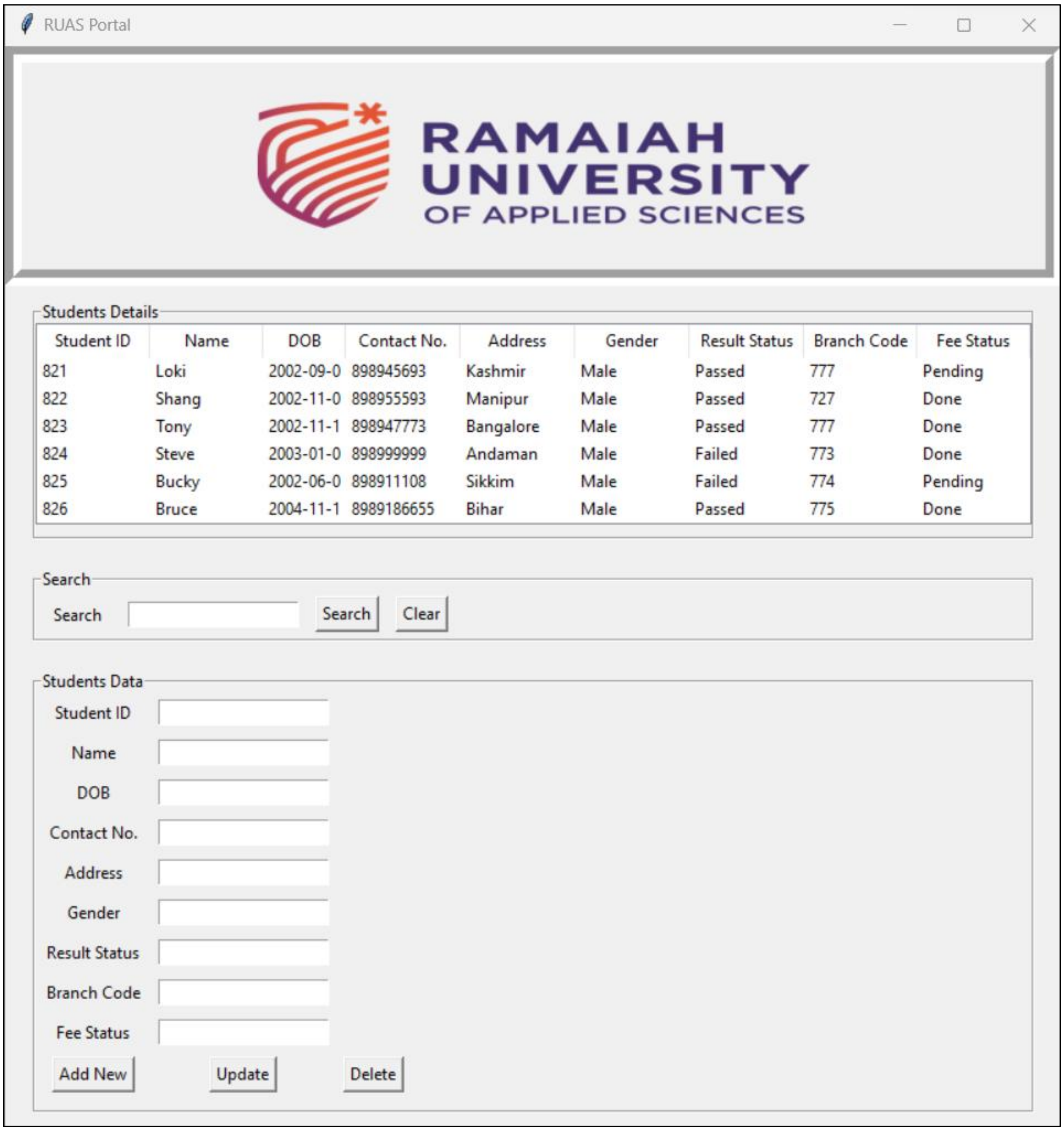
```
mysql> select name, credits
    -> from(student join course on branch_code=branch_code)
    -> where credits=4
    -> ;
+----------+---------+
| name     | credits |
+----------+---------+
| Loki     |       4 |
| Loki     |       4 |
| Shang    |       4 |
| Shang    |       4 |
| Tony     |       4 |
| Tony     |       4 |
| Steve    |       4 |
| Steve    |       4 |
| Scarlett |       4 |
| Scarlett |       4 |
| Bucky    |       4 |
| Bucky    |       4 |
| Bruce    |       4 |
| Bruce    |       4 |
| Wanda    |       4 |
| Wanda    |       4 |
| Thanos   |       4 |
| Thanos   |       4 |
| T'Chala  |       4 |
| T'Chala  |       4 |
| Groot    |       4 |
| Groot    |       4 |
| sis      |       4 |
| sis      |       4 |
| Hulk     |       4 |
| Hulk     |       4 |
| Bhumi    |       4 |
| Bhumi    |       4 |
+----------+---------+
28 rows in set (0.00 sec)
```



```
mysql> select name, credits
    -> from(student join course on branch_code=branch_no)
    -> where credits=4;
+-------+---------+
| name  | credits |
+-------+---------+
| Loki  |       4 |
| Tony  |       4 |
| Bhumi |       4 |
| Loki  |       4 |
| Tony  |       4 |
| Bhumi |       4 |
+-------+---------+
6 rows in set (0.00 sec)
```

Design and Implementation of GUI:





Students Details

Student ID	Name	DOB	Contact No.	Address	Gender	Result Status	Branch Code	Fee Status
821	Loki	2002-09-0	898945693	Kashmir	Male	Passed	777	Pending
822	Shang	2002-11-0	898955593	Manipur	Male	Passed	727	Done
823	Tony	2002-11-1	898947773	Bangalore	Male	Passed	777	Done
824	Steve	2003-01-0	898999999	Andaman	Male	Failed	773	Done
825	Bucky	2002-06-0	898911108	Sikkim	Male	Failed	774	Pending
826	Bruce	2004-11-1	8989186655	Bihar	Male	Passed	775	Done

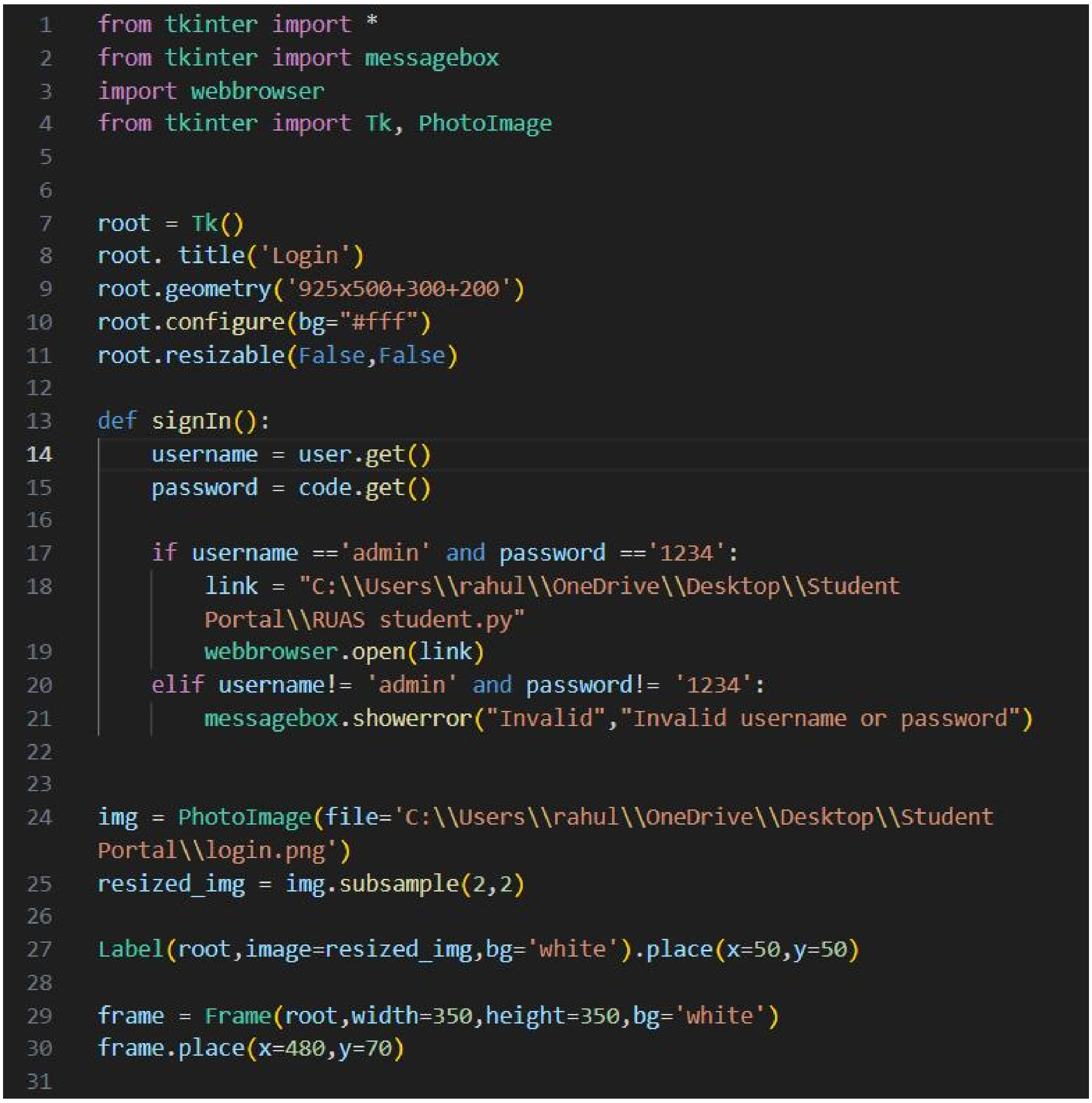
Search

Search [ ] Search Clear

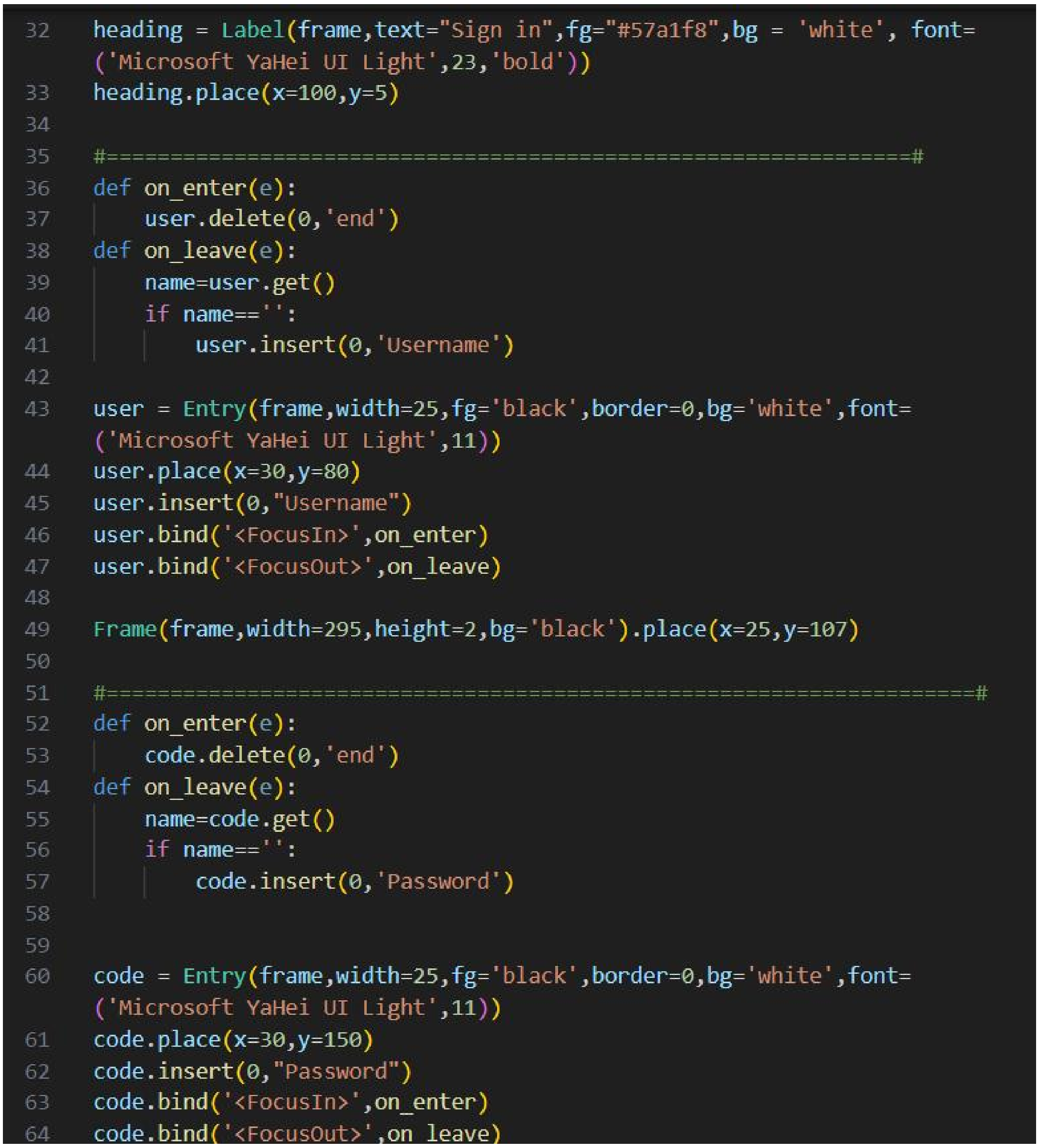
Students Data

Student ID	
Name	
DOB	
Contact No.	
Address	
Gender	
Result Status	
Branch Code	
Fee Status	

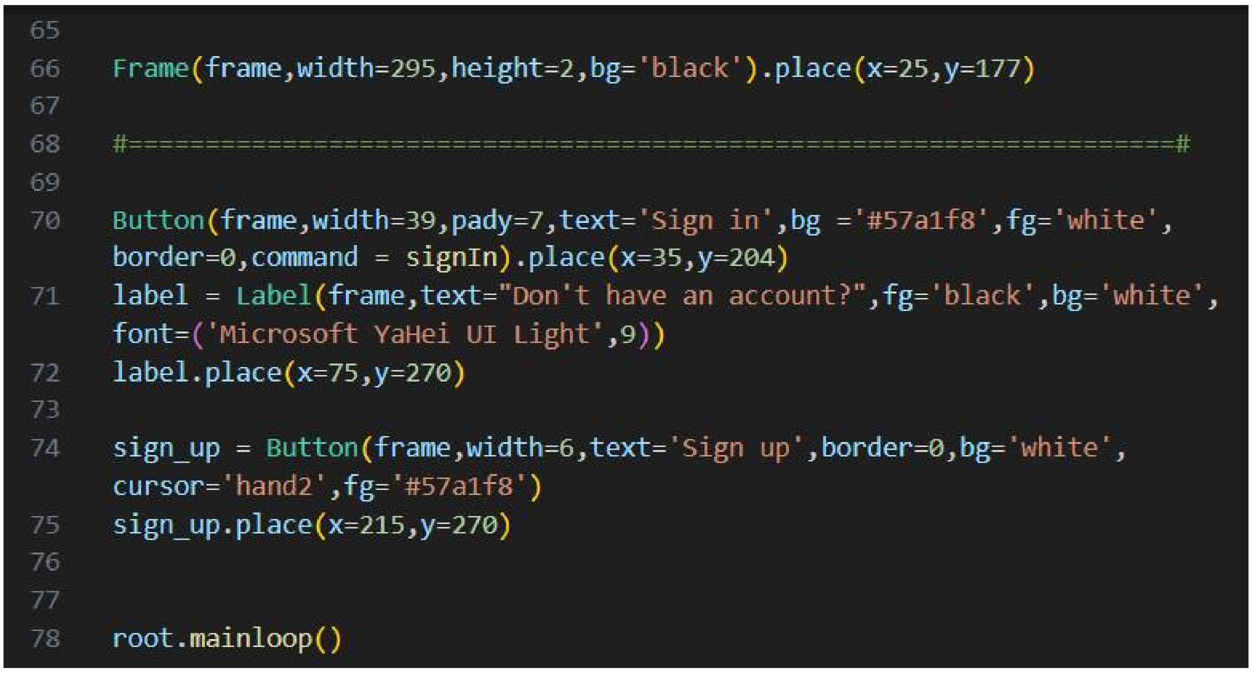
Code for Login Page:



```python
from tkinter import *
from tkinter import messagebox
import webbrowser
from tkinter import Tk, PhotoImage


root = Tk()
root. title('Login')
root.geometry('925x500+300+200')
root.configure(bg="#fff")
root.resizable(False,False)

def signIn():
    username = user.get()
    password = code.get()

    if username =='admin' and password =='1234':
        link = "C:\\Users\\rahul\\OneDrive\\Desktop\\Student
        Portal\\RUAS student.py"
        webbrowser.open(link)
    elif username!= 'admin' and password!= '1234':
        messagebox.showerror("Invalid","Invalid username or password")


img = PhotoImage(file='C:\\Users\\rahul\\OneDrive\\Desktop\\Student
Portal\\login.png')
resized_img = img.subsample(2,2)

Label(root,image=resized_img,bg='white').place(x=50,y=50)

frame = Frame(root,width=350,height=350,bg='white')
frame.place(x=480,y=70)

```

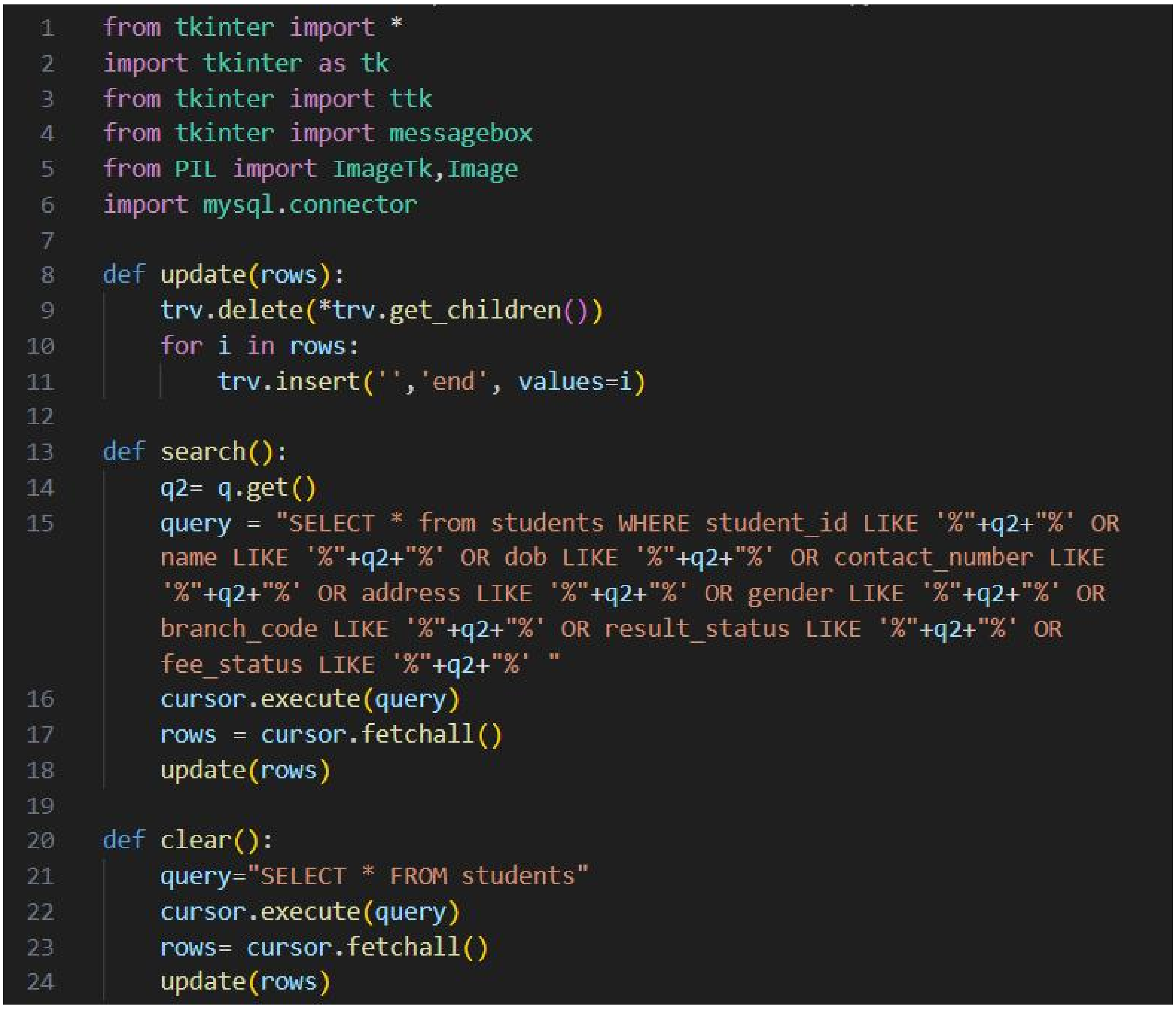


```python
heading = Label(frame,text="Sign in",fg="#57a1f8",bg = 'white', font=('Microsoft YaHei UI Light',23,'bold'))
heading.place(x=100,y=5)

#================================================================#
def on_enter(e):
    user.delete(0,'end')
def on_leave(e):
    name=user.get()
    if name=='':
        user.insert(0,'Username')

user = Entry(frame,width=25,fg='black',border=0,bg='white',font=('Microsoft YaHei UI Light',11))
user.place(x=30,y=80)
user.insert(0,"Username")
user.bind('<FocusIn>',on_enter)
user.bind('<FocusOut>',on_leave)

Frame(frame,width=295,height=2,bg='black').place(x=25,y=107)

#=====================================================================#
def on_enter(e):
    code.delete(0,'end')
def on_leave(e):
    name=code.get()
    if name=='':
        code.insert(0,'Password')


code = Entry(frame,width=25,fg='black',border=0,bg='white',font=('Microsoft YaHei UI Light',11))
code.place(x=30,y=150)
code.insert(0,"Password")
code.bind('<FocusIn>',on_enter)
code.bind('<FocusOut>',on_leave)
```

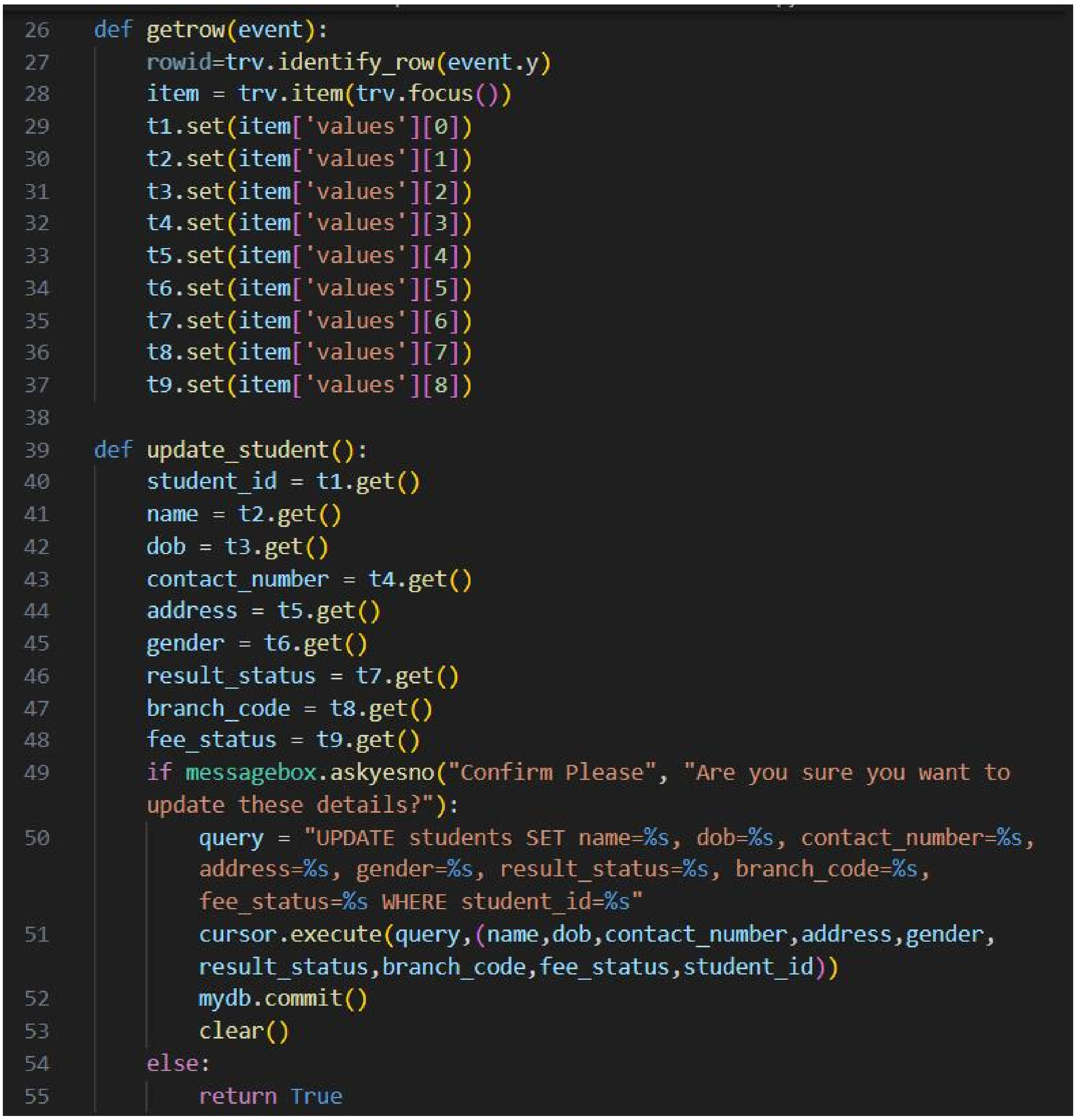


```python

Frame(frame,width=295,height=2,bg='black').place(x=25,y=177)

#===================================================================#

Button(frame,width=39,pady=7,text='Sign in',bg ='#57a1f8',fg='white',border=0,command = signIn).place(x=35,y=204)
label = Label(frame,text="Don't have an account?",fg='black',bg='white',font=('Microsoft YaHei UI Light',9))
label.place(x=75,y=270)

sign_up = Button(frame,width=6,text='Sign up',border=0,bg='white',cursor='hand2',fg='#57a1f8')
sign_up.place(x=215,y=270)


root.mainloop()
```

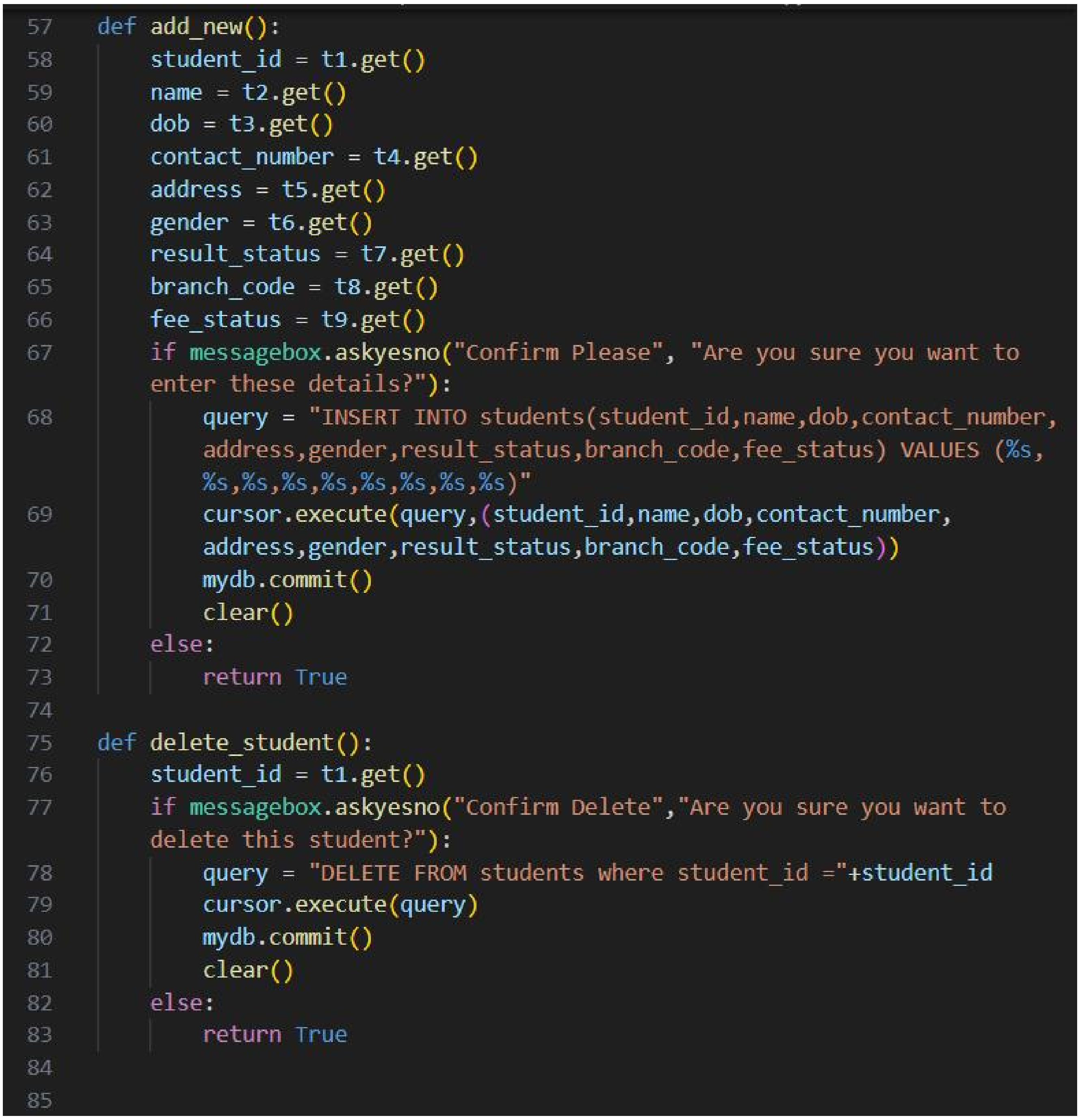
Connection of Front end with Database:



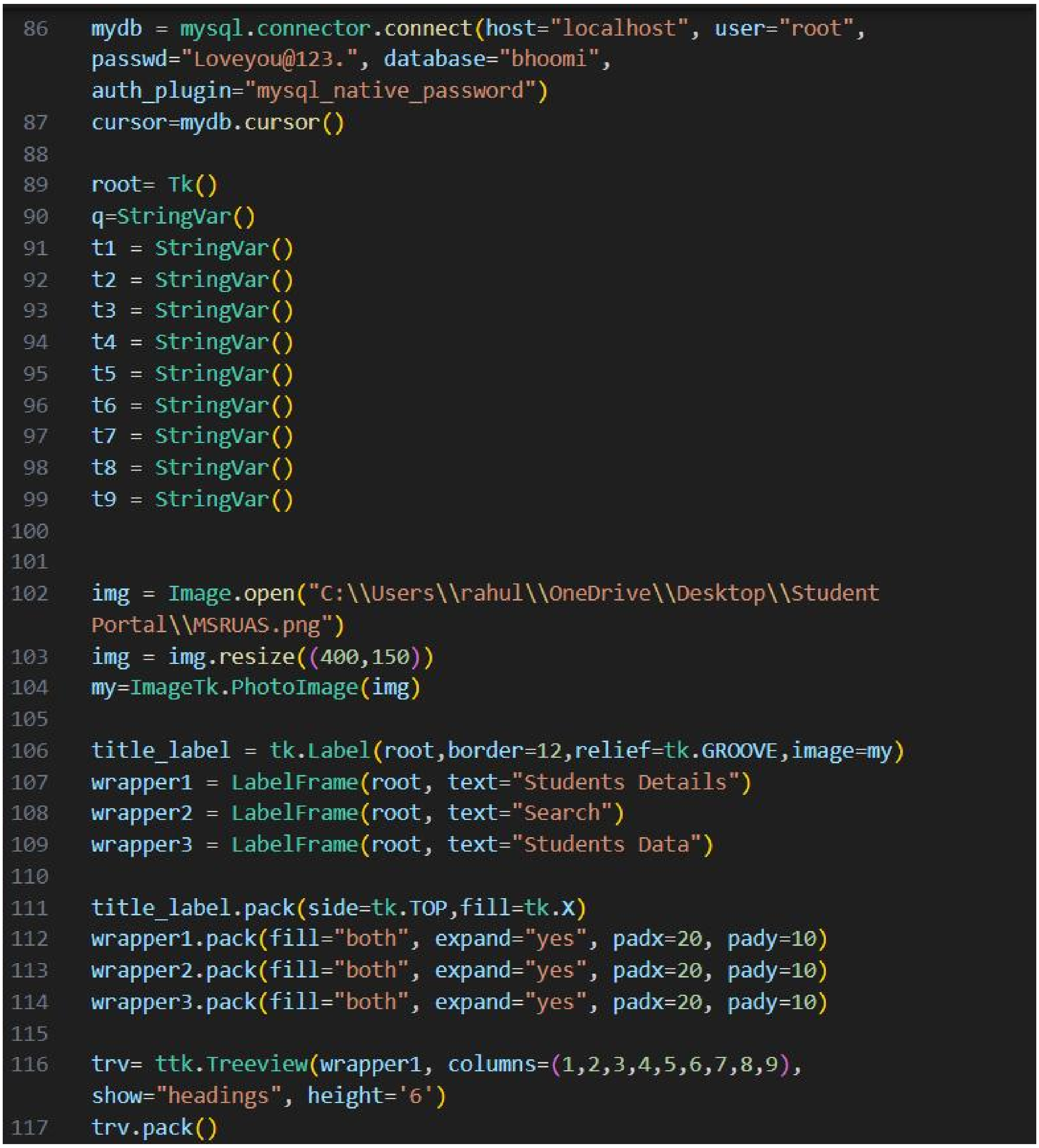
```python
from tkinter import *
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk,Image
import mysql.connector

def update(rows):
    trv.delete(*trv.get_children())
    for i in rows:
        trv.insert('','end', values=i)

def search():
    q2= q.get()
    query = "SELECT * from students WHERE student_id LIKE '%"+q2+"%' OR name LIKE '%"+q2+"%' OR dob LIKE '%"+q2+"%' OR contact_number LIKE '%"+q2+"%' OR address LIKE '%"+q2+"%' OR gender LIKE '%"+q2+"%' OR branch_code LIKE '%"+q2+"%' OR result_status LIKE '%"+q2+"%' OR fee_status LIKE '%"+q2+"%' "
    cursor.execute(query)
    rows = cursor.fetchall()
    update(rows)

def clear():
    query="SELECT * FROM students"
    cursor.execute(query)
    rows= cursor.fetchall()
    update(rows)
```



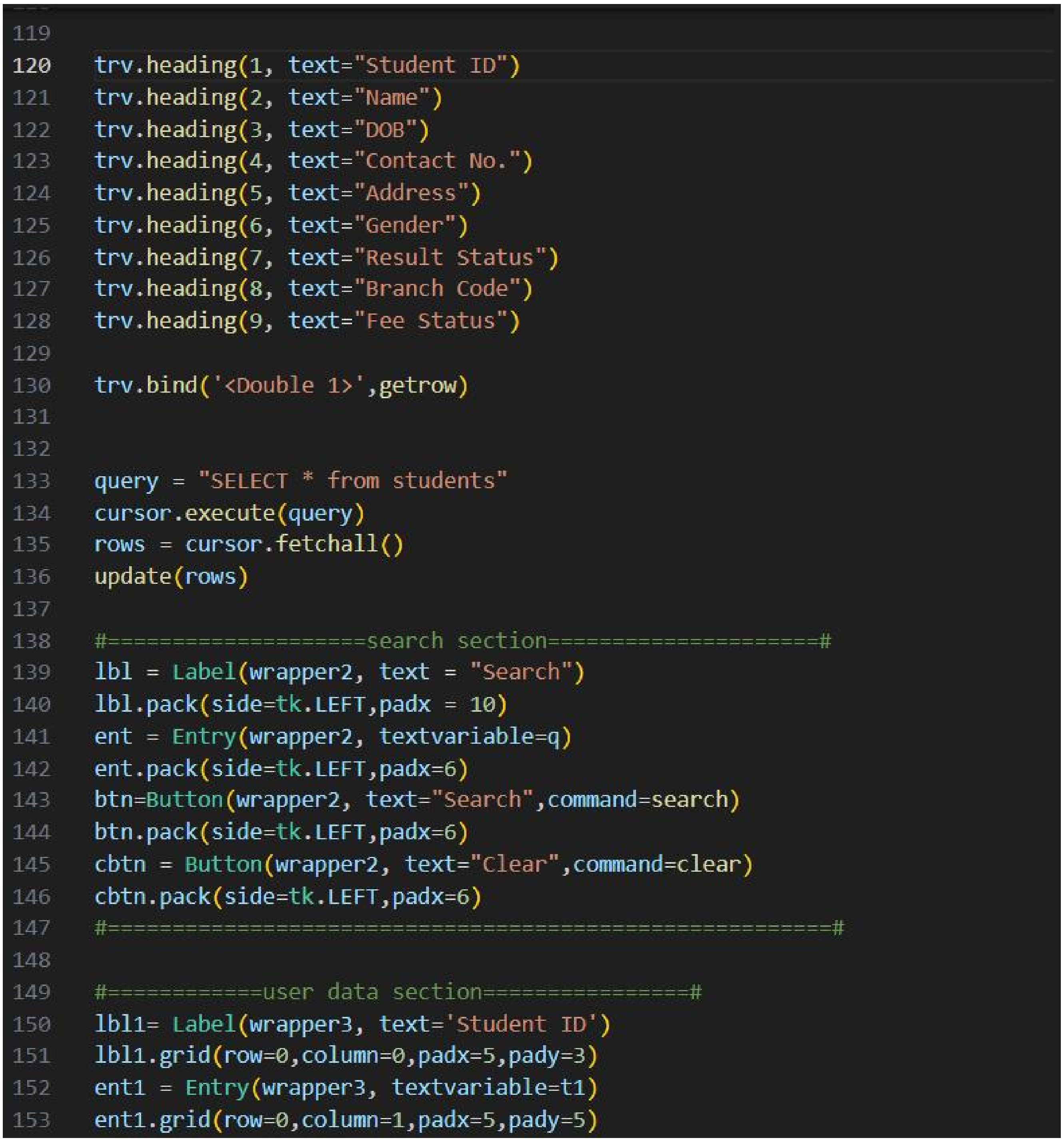
```python
def getrow(event):
    rowid=trv.identify_row(event.y)
    item = trv.item(trv.focus())
    t1.set(item['values'][0])
    t2.set(item['values'][1])
    t3.set(item['values'][2])
    t4.set(item['values'][3])
    t5.set(item['values'][4])
    t6.set(item['values'][5])
    t7.set(item['values'][6])
    t8.set(item['values'][7])
    t9.set(item['values'][8])

def update_student():
    student_id = t1.get()
    name = t2.get()
    dob = t3.get()
    contact_number = t4.get()
    address = t5.get()
    gender = t6.get()
    result_status = t7.get()
    branch_code = t8.get()
    fee_status = t9.get()
    if messagebox.askyesno("Confirm Please", "Are you sure you want to update these details?"):
        query = "UPDATE students SET name=%s, dob=%s, contact_number=%s, address=%s, gender=%s, result_status=%s, branch_code=%s, fee_status=%s WHERE student_id=%s"
        cursor.execute(query,(name,dob,contact_number,address,gender,result_status,branch_code,fee_status,student_id))
        mydb.commit()
        clear()
    else:
        return True
```



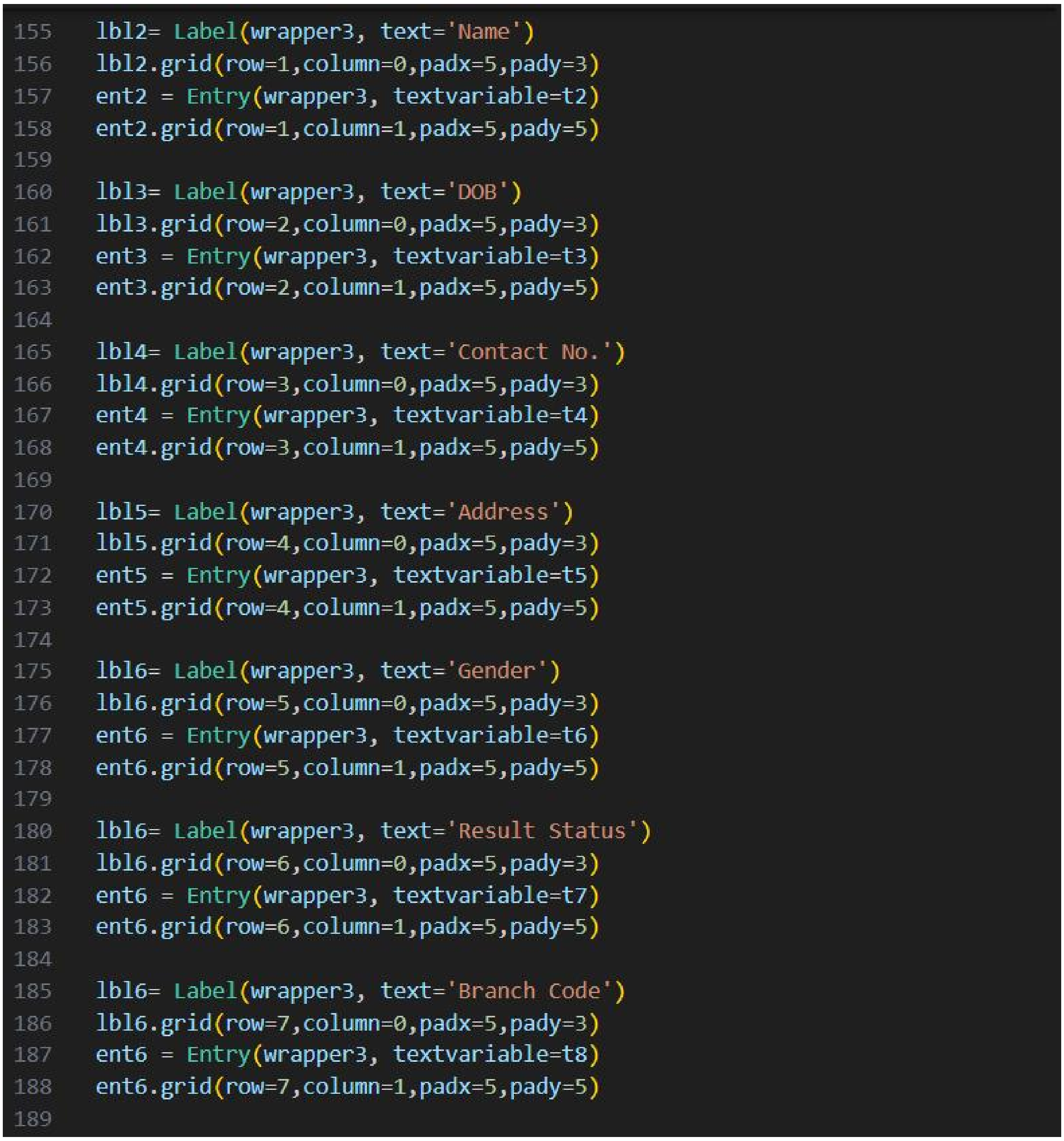
```python
def add_new():
    student_id = t1.get()
    name = t2.get()
    dob = t3.get()
    contact_number = t4.get()
    address = t5.get()
    gender = t6.get()
    result_status = t7.get()
    branch_code = t8.get()
    fee_status = t9.get()
    if messagebox.askyesno("Confirm Please", "Are you sure you want to enter these details?"):
        query = "INSERT INTO students(student_id,name,dob,contact_number,address,gender,result_status,branch_code,fee_status) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)"
        cursor.execute(query,(student_id,name,dob,contact_number,address,gender,result_status,branch_code,fee_status))
        mydb.commit()
        clear()
    else:
        return True

def delete_student():
    student_id = t1.get()
    if messagebox.askyesno("Confirm Delete","Are you sure you want to delete this student?"):
        query = "DELETE FROM students where student_id ="+student_id
        cursor.execute(query)
        mydb.commit()
        clear()
    else:
        return True
```



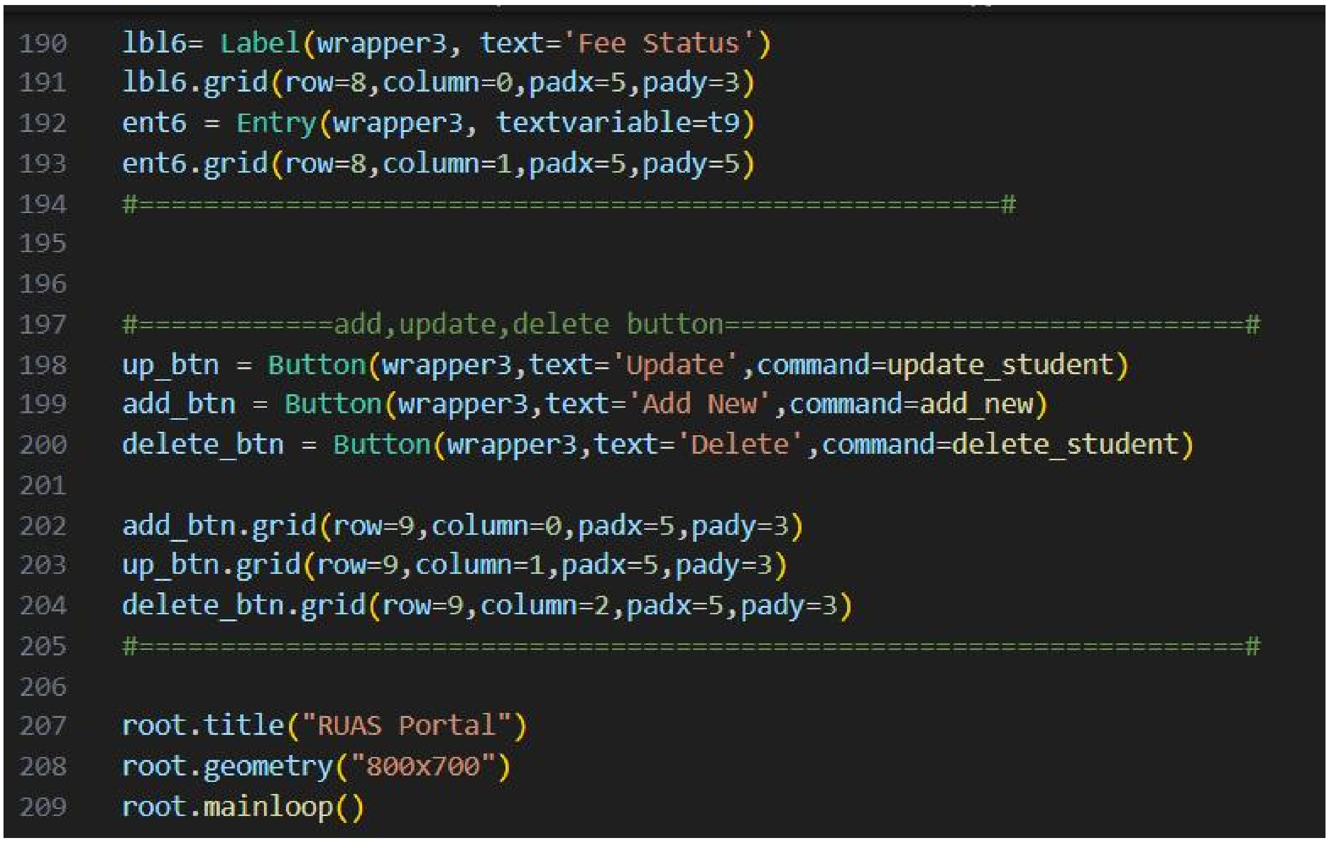
```python
mydb = mysql.connector.connect(host="localhost", user="root", passwd="Loveyou@123.", database="bhoomi", auth_plugin="mysql_native_password")
cursor=mydb.cursor()

root= Tk()
q=StringVar()
t1 = StringVar()
t2 = StringVar()
t3 = StringVar()
t4 = StringVar()
t5 = StringVar()
t6 = StringVar()
t7 = StringVar()
t8 = StringVar()
t9 = StringVar()


img = Image.open("C:\\Users\\rahul\\OneDrive\\Desktop\\Student Portal\\MSRUAS.png")
img = img.resize((400,150))
my=ImageTk.PhotoImage(img)

title_label = tk.Label(root,border=12,relief=tk.GROOVE,image=my)
wrapper1 = LabelFrame(root, text="Students Details")
wrapper2 = LabelFrame(root, text="Search")
wrapper3 = LabelFrame(root, text="Students Data")

title_label.pack(side=tk.TOP,fill=tk.X)
wrapper1.pack(fill="both", expand="yes", padx=20, pady=10)
wrapper2.pack(fill="both", expand="yes", padx=20, pady=10)
wrapper3.pack(fill="both", expand="yes", padx=20, pady=10)

trv= ttk.Treeview(wrapper1, columns=(1,2,3,4,5,6,7,8,9), show="headings", height='6')
trv.pack()
```



```python

trv.heading(1, text="Student ID")
trv.heading(2, text="Name")
trv.heading(3, text="DOB")
trv.heading(4, text="Contact No.")
trv.heading(5, text="Address")
trv.heading(6, text="Gender")
trv.heading(7, text="Result Status")
trv.heading(8, text="Branch Code")
trv.heading(9, text="Fee Status")

trv.bind('<Double 1>',getrow)


query = "SELECT * from students"
cursor.execute(query)
rows = cursor.fetchall()
update(rows)

#===================search section====================#
lbl = Label(wrapper2, text = "Search")
lbl.pack(side=tk.LEFT,padx = 10)
ent = Entry(wrapper2, textvariable=q)
ent.pack(side=tk.LEFT,padx=6)
btn=Button(wrapper2, text="Search",command=search)
btn.pack(side=tk.LEFT,padx=6)
cbtn = Button(wrapper2, text="Clear",command=clear)
cbtn.pack(side=tk.LEFT,padx=6)
#=========================================================#

#===========user data section===============#
lbl1= Label(wrapper3, text='Student ID')
lbl1.grid(row=0,column=0,padx=5,pady=3)
ent1 = Entry(wrapper3, textvariable=t1)
ent1.grid(row=0,column=1,padx=5,pady=5)
```



```
lbl2= Label(wrapper3, text='Name')
lbl2.grid(row=1,column=0,padx=5,pady=3)
ent2 = Entry(wrapper3, textvariable=t2)
ent2.grid(row=1,column=1,padx=5,pady=5)

lbl3= Label(wrapper3, text='DOB')
lbl3.grid(row=2,column=0,padx=5,pady=3)
ent3 = Entry(wrapper3, textvariable=t3)
ent3.grid(row=2,column=1,padx=5,pady=5)

lbl4= Label(wrapper3, text='Contact No.')
lbl4.grid(row=3,column=0,padx=5,pady=3)
ent4 = Entry(wrapper3, textvariable=t4)
ent4.grid(row=3,column=1,padx=5,pady=5)

lbl5= Label(wrapper3, text='Address')
lbl5.grid(row=4,column=0,padx=5,pady=3)
ent5 = Entry(wrapper3, textvariable=t5)
ent5.grid(row=4,column=1,padx=5,pady=5)

lbl6= Label(wrapper3, text='Gender')
lbl6.grid(row=5,column=0,padx=5,pady=3)
ent6 = Entry(wrapper3, textvariable=t6)
ent6.grid(row=5,column=1,padx=5,pady=5)

lbl6= Label(wrapper3, text='Result Status')
lbl6.grid(row=6,column=0,padx=5,pady=3)
ent6 = Entry(wrapper3, textvariable=t7)
ent6.grid(row=6,column=1,padx=5,pady=5)

lbl6= Label(wrapper3, text='Branch Code')
lbl6.grid(row=7,column=0,padx=5,pady=3)
ent6 = Entry(wrapper3, textvariable=t8)
ent6.grid(row=7,column=1,padx=5,pady=5)

```



```python
lbl6= Label(wrapper3, text='Fee Status')
lbl6.grid(row=8,column=0,padx=5,pady=3)
ent6 = Entry(wrapper3, textvariable=t9)
ent6.grid(row=8,column=1,padx=5,pady=5)
#=======================================================#


#============add,update,delete button================================#
up_btn = Button(wrapper3,text='Update',command=update_student)
add_btn = Button(wrapper3,text='Add New',command=add_new)
delete_btn = Button(wrapper3,text='Delete',command=delete_student)

add_btn.grid(row=9,column=0,padx=5,pady=3)
up_btn.grid(row=9,column=1,padx=5,pady=3)
delete_btn.grid(row=9,column=2,padx=5,pady=3)
#======================================================================#

root.title("RUAS Portal")
root.geometry("800x700")
root.mainloop()
```

Conclusion:

A well-designed student database management system (DBMS) crafted with Tkinter and MySQL presents an effective solution for organizing, managing, and retrieving student information efficiently. Through the development process, we have successfully integrated Tkinter's user-friendly interface with MySQL's robust database management capabilities, resulting in a comprehensive system tailored to meet the needs of educational institutions.
By leveraging Tkinter's GUI toolkit, we have created an intuitive interface that allows users to interact with the database seamlessly. Through intuitive forms and controls, users can easily input, update, and retrieve student data with minimal effort. Additionally, Tkinter's versatility enables us to design a visually appealing interface that enhances user experience and promotes ease of navigation.
Furthermore, by harnessing the power of MySQL, we have established a secure and reliable database backend to store and manage student records. MySQL's scalability and performance ensure that the system can accommodate large volumes of data without compromising speed or efficiency. With features such as data integrity constraints, indexing, and relational database management, MySQL enables us to maintain the integrity and consistency of student data while facilitating efficient data retrieval and analysis.
In conclusion, the integration of Tkinter and MySQL has enabled us to develop a robust student database management system that streamlines administrative tasks, enhances data organization, and improves overall productivity within educational institutions. Whether it's managing student enrollment, tracking academic progress, or generating reports, our system provides a comprehensive solution that meets the diverse needs of educators, administrators, and students alike. Through continued refinement and adaptation, we are committed to evolving our system to ensure it remains a valuable asset in the educational landscape.

References:

https://www.google.com/
https://geeksforgeeks.org/
https://www.wikipedia.org/
https://www.youtube.com/