

## Create University Table:

The screenshot shows a database management interface. On the left, a tree view displays the database structure, including a 'student' table under a 'databasename' schema. The main area is titled 'Run SQL query/queries on table databasename.student:'. It contains an SQL editor with the following code:

```
1 CREATE TABLE university (  
2   StudentID int,  
3   StudentName varchar(255),  
4   City varchar(255),  
5   PostalCode varchar(255),  
6   Number int,  
7   INCOME varchar(255)  
8 );
```

To the right of the editor, a list of columns is displayed: studentid, studentname, city, postalcode, and income. At the bottom, there are buttons for 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', 'Format', and a '<<' button.

## Insert Values in University:

The screenshot shows a database management interface. On the left, a tree view displays the database structure, including a database named 'sqa' which contains a table named 'university'. The 'university' table is selected. The main area displays an SQL query to insert data into the 'university' table. The query is as follows:

```
1 INSERT INTO university (StudentID, StudentName, City, PostalCode, Number, INCOME)
2 VALUES (101, 'torin', 'tangail', '1000', 01628236296, '3500'),
3 (102, 'nibir', 'rangpur', '1100', 01728236296, '3250'),
4 (103, 'masud', 'dhaka', '1200', 01727236296, '3750'),
5 (104, 'hridoy', 'dhaka', '1300', 01625236296, '3800'),
6 (105, 'tushar', 'tangail', '1400', 01623236296, '3800'),
7 (106, 'ferdus', 'comilla', '1500', 01720236296, '4000'),
8 (107, 'jiku', 'tangail', '1600', 01729236296, '4000'),
9 (108, 'taslima', 'dhaka', '1700', 01630236296, '4000'),
10 (109, 'akib', 'tangail', '1800', 01732236296, '3500');
```

On the right side of the interface, the columns of the 'university' table are listed: StudentID, StudentName, City, PostalCode, Number, and INCOME. At the bottom, there are buttons for 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', 'Format', and a '<<' button.

# University Table:

New

bugresistance

databasename

information\_schema

mysql

performance\_schema

phpmyadmin

sq

New

university

test

Profiling [ Edit inline ] [ Edit ]

Show all

Number of rows: 25

Filter rows: Search this table

+ Options

| StudentID | StudentName | City    | PostalCode | Number     | INCOME |
|-----------|-------------|---------|------------|------------|--------|
| 101       | torin       | tangail | 1000       | 1628236296 | 3500   |
| 102       | nibir       | rangpur | 1100       | 1728236296 | 3250   |
| 103       | masud       | dhaka   | 1200       | 1727236296 | 3750   |
| 104       | hridoy      | dhaka   | 1300       | 1625236296 | 3800   |
| 105       | tushar      | tangail | 1400       | 1623236296 | 3800   |
| 106       | ferdus      | comilla | 1500       | 1720236296 | 4000   |
| 107       | jiku        | tangail | 1600       | 1729236296 | 4000   |
| 108       | taslima     | dhaka   | 1700       | 1630236296 | 4000   |
| 109       | akib        | tangail | 1800       | 1732236296 | 3500   |

Show all

Number of rows: 25

Filter rows: Search this table

Query results operations

## Find Data by City:

The screenshot shows a database management tool interface. On the left is a sidebar with a tree view of databases and tables. The 'university' table is selected. The main area displays the SQL query: `select * from university where city='tangail';`. Below the query, there are controls for 'Show all', 'Number of rows' (set to 25), and a 'Filter rows' search box. The query results are shown in a table with 6 columns: StudentID, StudentName, City, PostalCode, Number, and INCOME. There are 4 rows of data. Below the results table, there is another set of controls identical to the one above. At the bottom, there is a 'Query results operations' button.

Recent Favorites

New

bugresistance

databasename

New

student

information\_schema

mysql

performance\_schema

phpmyadmin

sqla

New

university

test

Showing rows 0 - 3 (4 total, Query took 0.0015 seconds.)

```
select * from university where city='tangail';
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP](#) ]

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

+ Options

| StudentID | StudentName | City    | PostalCode | Number     | INCOME |
|-----------|-------------|---------|------------|------------|--------|
| 101       | torin       | tangail | 1000       | 1628236296 | 3500   |
| 105       | tushar      | tangail | 1400       | 1623236296 | 3800   |
| 107       | jiku        | tangail | 1600       | 1729236296 | 4000   |
| 109       | akib        | tangail | 1800       | 1732236296 | 3500   |

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

Query results operations

## Between Operator select values:

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The 'university' database is selected. The main area displays a SQL query: `SELECT * FROM university WHERE Income BETWEEN 3250 AND 3800;`. Below the query bar are controls for 'Show all', 'Number of rows' (set to 25), and a 'Filter rows' search box. A '+ Options' link is visible. The query results are shown in a table with 6 columns: StudentID, StudentName, City, PostalCode, Number, and INCOME. The results list 6 rows of data where the income is between 3250 and 3800.

**SQL Query:**

```
SELECT * FROM university WHERE Income BETWEEN 3250 AND 3800;
```

**Table Options:**

- ☐ Show all
- Number of rows: 25
- Filter rows: Search this table

**+ Options**

| StudentID | StudentName | City    | PostalCode | Number     | INCOME |
|-----------|-------------|---------|------------|------------|--------|
| 101       | torin       | tangail | 1000       | 1628236296 | 3500   |
| 102       | nibir       | rangpur | 1100       | 1728236296 | 3250   |
| 103       | masud       | dhaka   | 1200       | 1727236296 | 3750   |
| 104       | hridoy      | dhaka   | 1300       | 1625236296 | 3800   |
| 105       | tushar      | tangail | 1400       | 1623236296 | 3800   |
| 109       | akib        | tangail | 1800       | 1732236296 | 3500   |

**Table Options (Bottom):**

- ☐ Show all
- Number of rows: 25
- Filter rows: Search this table

## MAX Function:

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of databases and tables. The 'university' table is selected. The main area displays a SQL query: `SELECT * FROM university WHERE INCOME = ( select max(INCOME) from university);`. Below the query bar are controls for 'Show all', 'Number of rows' (set to 25), and a 'Filter rows' search box. A '+ Options' link is visible. The query result is shown as a table with 6 columns: StudentID, StudentName, City, PostalCode, Number, and INCOME. Three rows are displayed, all with an INCOME of 4000. At the bottom, there is another set of controls identical to the one above.

**SQL Query:**

```
SELECT * FROM university WHERE INCOME = ( select max(INCOME) from university);
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ]

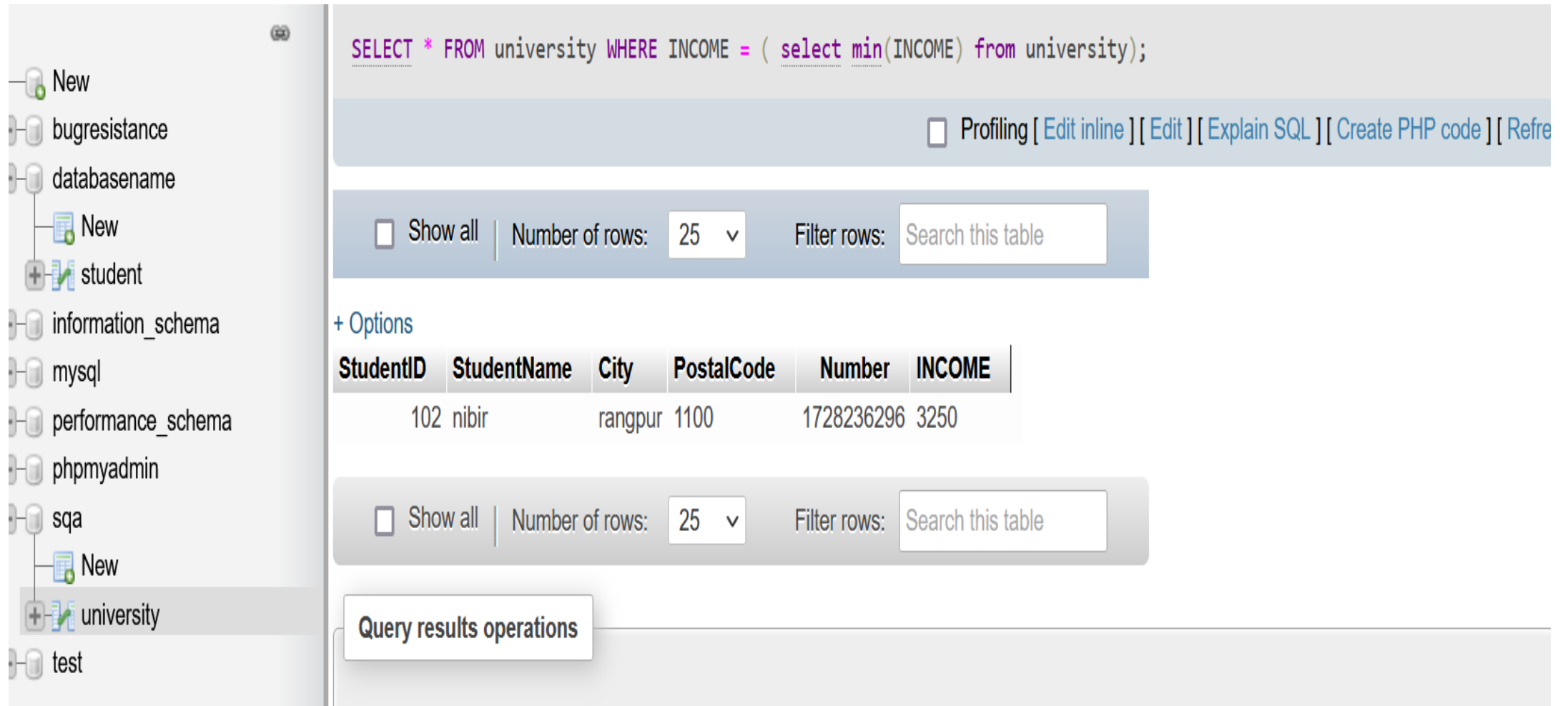
☐ Show all | Number of rows: 25 ▼ | Filter rows:

[+ Options](#)

| StudentID | StudentName | City    | PostalCode | Number     | INCOME |
|-----------|-------------|---------|------------|------------|--------|
| 106       | ferdus      | comilla | 1500       | 1720236296 | 4000   |
| 107       | jiku        | tangail | 1600       | 1729236296 | 4000   |
| 108       | taslima     | dhaka   | 1700       | 1630236296 | 4000   |

☐ Show all | Number of rows: 25 ▼ | Filter rows:

## MIN Function:



The screenshot shows a database management interface. On the left is a sidebar with a tree view of databases and tables. The 'university' table is selected. The main area displays a SQL query: `SELECT * FROM university WHERE INCOME = ( select min(INCOME) from university);`. Below the query are links for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. A control bar shows 'Show all', 'Number of rows: 25', and a 'Filter rows' search box. Below this is a '+ Options' link. The query results are shown in a table with columns: StudentID, StudentName, City, PostalCode, Number, and INCOME. One row is visible: 102, nibir, rangpur, 1100, 1728236296, 3250. Another control bar is below the table. At the bottom is a 'Query results operations' button.

**SQL Query:**

```
SELECT * FROM university WHERE INCOME = ( select min(INCOME) from university);
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▾ | Filter rows:

[+ Options](#)

| StudentID | StudentName | City    | PostalCode | Number     | INCOME |
|-----------|-------------|---------|------------|------------|--------|
| 102       | nibir       | rangpur | 1100       | 1728236296 | 3250   |

☐ Show all | Number of rows: 25 ▾ | Filter rows:

[Query results operations](#)

## Create Student Table:

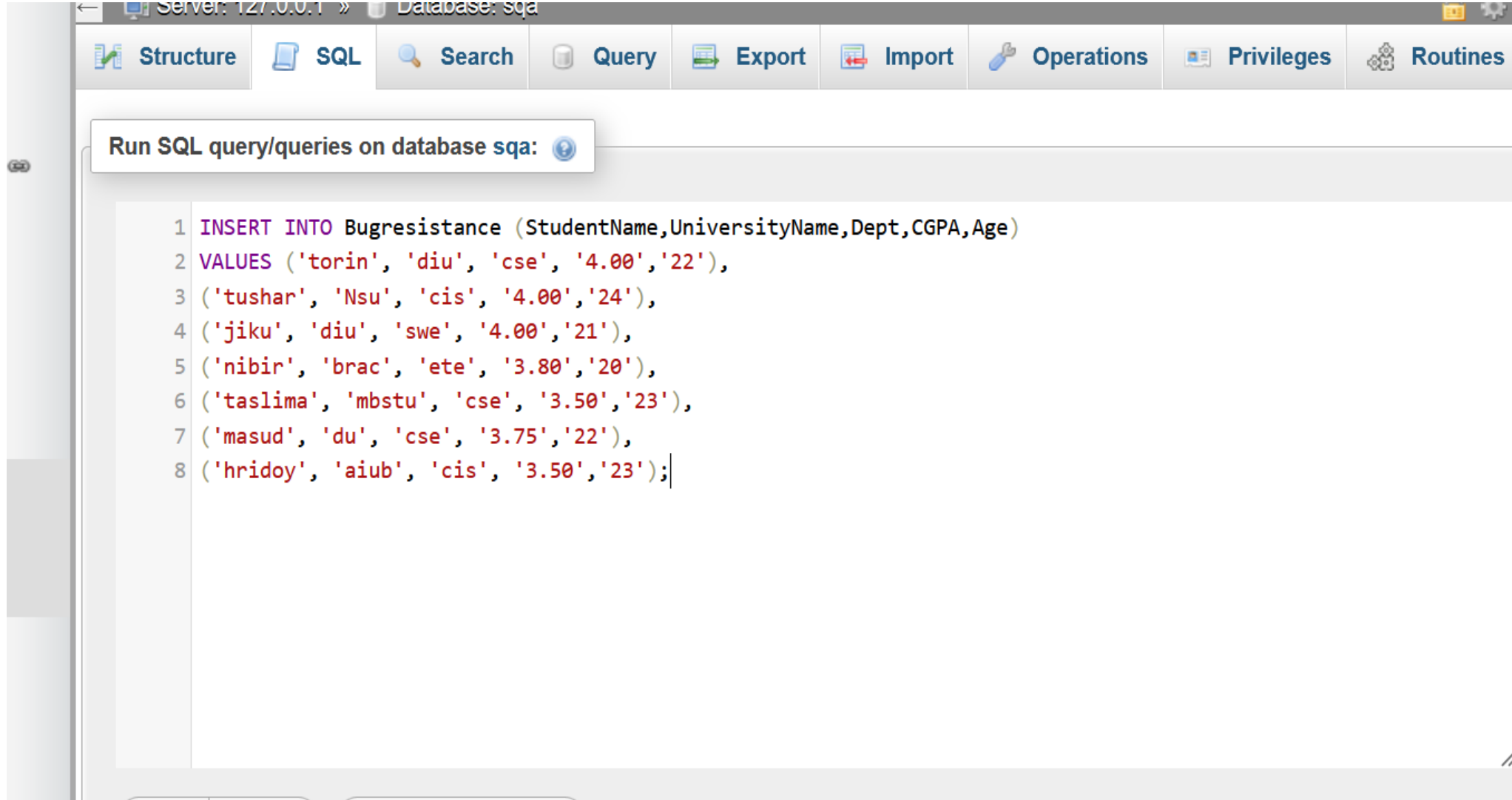
The screenshot shows a database management tool interface. At the top, there is a horizontal toolbar with icons and labels for 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', and 'Operations'. Below the toolbar, a tab is active with the title 'Run SQL query/queries on table sqa.student:'. The main area contains a text editor with the following SQL code:

```
1 CREATE TABLE student (  
2   StudentName varchar(255),  
3   UniversityName varchar(255),  
4   Dept varchar(255),  
5   CGPA varchar(255),  
6   Age varchar(255)  
7 );  
8 |
```

To the right of the text editor, there is a list of column names: Studentname, UniversityName, Dept, CGPA, and Age.



## Insert values in Student:



The screenshot shows a database management tool interface. At the top, there is a toolbar with icons and labels for 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', 'Privileges', and 'Routines'. Below the toolbar, a status bar indicates 'Server: 127.0.0.1' and 'Database: sqa'. A text box above the main editor area says 'Run SQL query/queries on database sqa:'. The main editor area contains an SQL query to insert data into a table named 'Bugresistance'.

```
1 INSERT INTO Bugresistance (StudentName,UniversityName,Dept,CGPA,Age)
2 VALUES ('torin', 'diu', 'cse', '4.00','22'),
3 ('tushar', 'Nsu', 'cis', '4.00','24'),
4 ('jiku', 'diu', 'swe', '4.00','21'),
5 ('nibir', 'brac', 'ete', '3.80','20'),
6 ('taslima', 'mbstu', 'cse', '3.50','23'),
7 ('masud', 'du', 'cse', '3.75','22'),
8 ('hridoy', 'aiub', 'cis', '3.50','23');
```

## Student Table:

New

bugresistance

databasename

information\_schema

mysql

performance\_schema

phpmyadmin

sqla

New

+

student

+

university

test

☐ Show all

Number of rows: 25 ▾

Filter rows:

+ Options

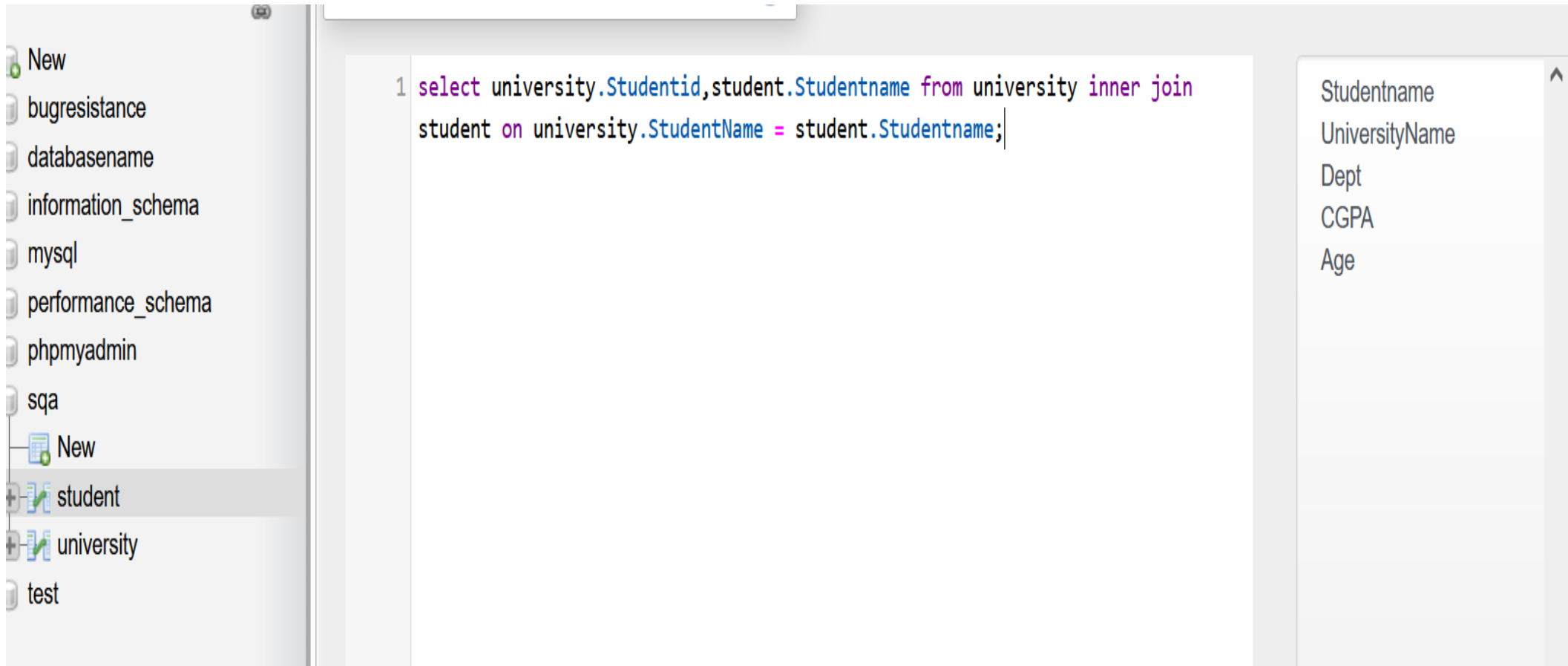
| Studentname | UniversityName | Dept | CGPA | Age |
|-------------|----------------|------|------|-----|
| torin       | diu            | cse  | 4.00 | 22  |
| tushar      | nsu            | cis  | 4.00 | 24  |
| jiku        | diu            | swe  | 4.00 | 21  |
| nibir       | brac           | ete  | 3.80 | 20  |
| taslima     | mbstu          | cse  | 3.50 | 23  |
| masud       | du             | cse  | 3.75 | 22  |
| hridoy      | aiub           | cis  | 3.50 | 23  |

☐ Show all

Number of rows: 25 ▾

Filter rows:

## Insert Values in Inner Join:



The screenshot displays a database management interface. On the left, a sidebar shows a tree view of database objects, including 'New', 'bugresistance', 'databasename', 'information\_schema', 'mysql', 'performance\_schema', 'phpmyadmin', 'sqa', 'New', 'student', 'university', and 'test'. The 'student' table is selected. The main area shows an SQL query in a text editor:

```
1 select university.Studentid,student.Studentname from university inner join  
student on university.StudentName = student.Studentname;
```

On the right, a results pane displays the output of the query, showing the following columns:

| Studentname    |
|----------------|
| UniversityName |
| Dept           |
| CGPA           |
| Age            |

## Inner Join Table:

New

bugresistance

databasename

information\_schema

mysql

performance\_schema

phpmyadmin

sqa

New

student

university

test

```
select university.Studentid,student.Studentname from university inner join student on university.StudentName = student.Studentname;
```

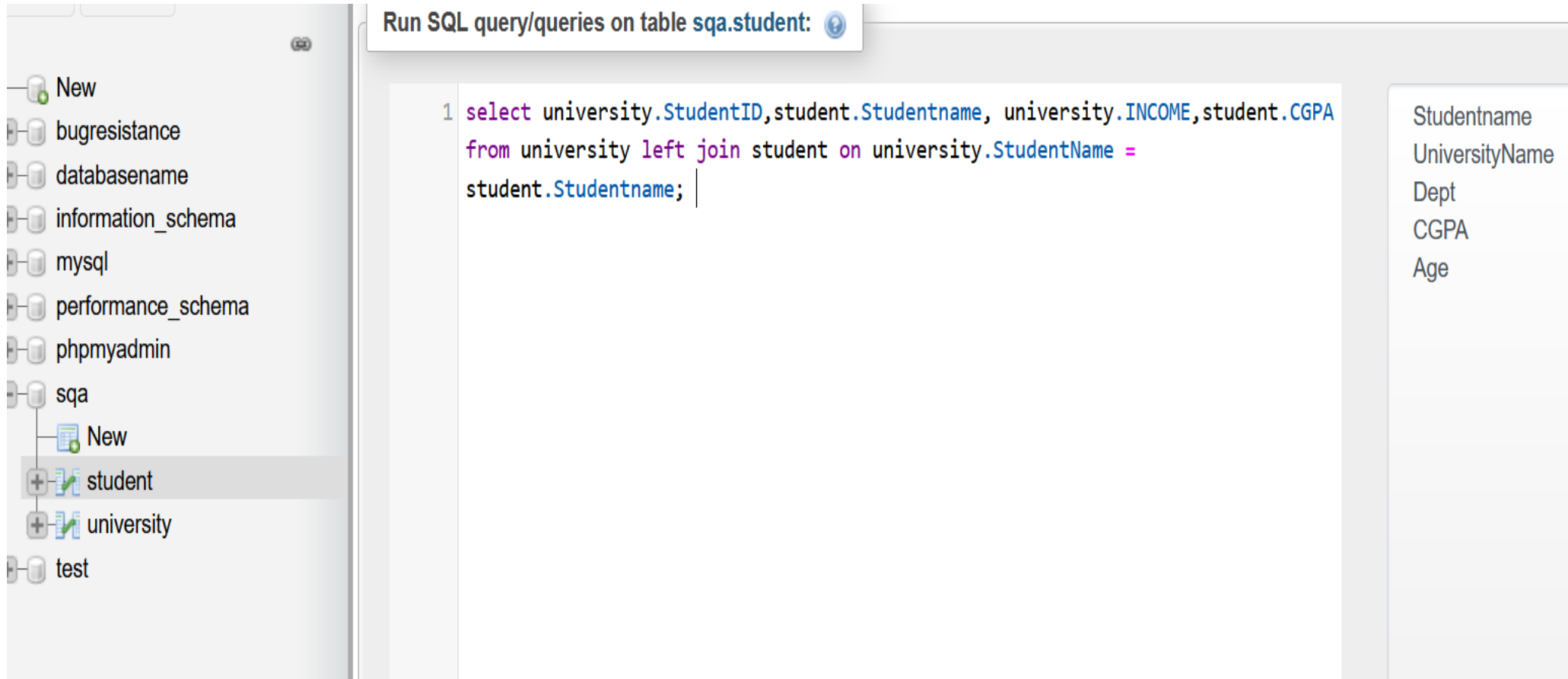
☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Re \]](#)

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

+ Options

| Studentid | Studentname |
|-----------|-------------|
| 101       | torin       |
| 102       | nibir       |
| 103       | masud       |
| 104       | hridoy      |
| 105       | tushar      |
| 107       | jiku        |
| 108       | taslima     |

## Insert Values in Left Join:



The screenshot displays a database management interface. On the left, a tree view shows the database structure, with the 'student' table under the 'sqa' schema selected. The main area shows a SQL query being executed on the 'sqa.student' table. The query is a left join between the 'university' and 'student' tables, selecting 'StudentID', 'Studentname', 'INCOME', and 'CGPA' from 'university' and 'Studentname' from 'student'. The results pane on the right shows the columns 'Studentname', 'UniversityName', 'Dept', 'CGPA', and 'Age'.

Run SQL query/queries on table sqa.student: ?

```
1 select university.StudentID, student.Studentname, university.INCOME, student.CGPA
   from university left join student on university.StudentName =
   student.Studentname;
```

Studentname  
UniversityName  
Dept  
CGPA  
Age

## Left Join Table:

|                    |  |
|--------------------|--|
| New                |  |
| bugresistance      |  |
| databasename       |  |
| information_schema |  |
| mysql              |  |
| performance_schema |  |
| phpmyadmin         |  |
| sqa                |  |
| New                |  |
| + student          |  |
| + university       |  |
| test               |  |

+ Options

| StudentID | Studentname | INCOME | CGPA |
|-----------|-------------|--------|------|
| 101       | torin       | 3500   | 4.00 |
| 105       | tushar      | 3800   | 4.00 |
| 107       | jiku        | 4000   | 4.00 |
| 102       | nibir       | 3250   | 3.80 |
| 108       | taslima     | 4000   | 3.50 |
| 103       | masud       | 3750   | 3.75 |
| 104       | hridoy      | 3800   | 3.50 |
| 106       | NULL        | 4000   | NULL |
| 109       | NULL        | 3500   | NULL |

## Insert Values in Right Join:

The screenshot shows a database management interface. On the left is a sidebar with a tree view of database objects. The main area contains a SQL query editor with a single query. On the right is a panel showing the column names of the tables involved in the query. At the bottom is a toolbar with buttons for SQL operations.

**Database Structure (Left Sidebar):**

- New
- bugresistance
- databasename
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- sqla
  - New
  - student
  - university
- test

**SQL Query (Center Editor):**

```
1 select university.StudentID,student.Studentname,university.number,student.age
from university right join student on university.StudentName =
student.Studentname;
```

**Table Columns (Right Panel):**

- Studentname
- UniversityName
- Dept
- CGPA
- Age

**Toolbar (Bottom):**

- SELECT \*
- SELECT
- INSERT
- UPDATE
- DELETE
- Clear
- Format
- <<

## Right Join Table:

New

bugresistance

databasename

information\_schema

mysql

performance\_schema

phpmyadmin

sq

New

student

university

test

☐ Show all | Number of rows: 25 ▾ | Filter rows:

+ Options

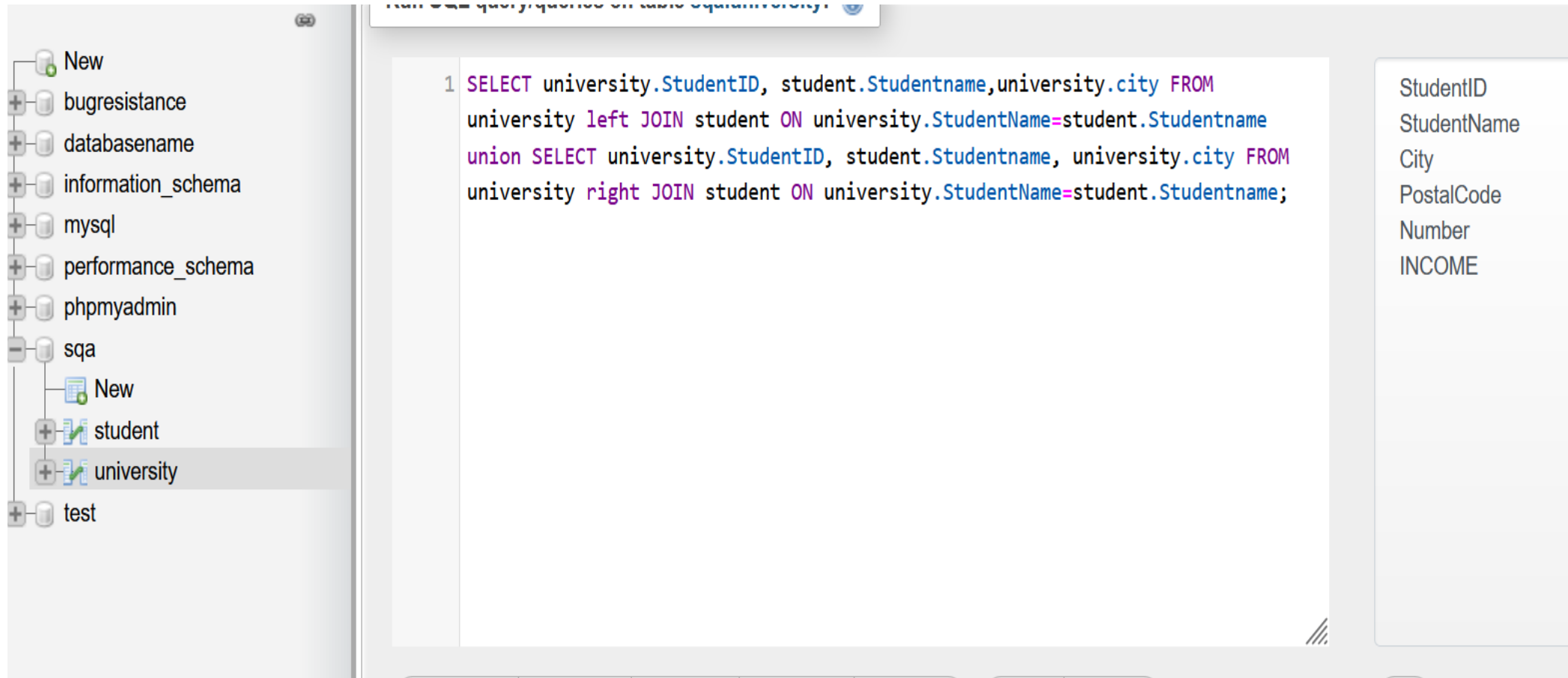
| StudentID | Studentname | number     | age |
|-----------|-------------|------------|-----|
| 101       | torin       | 1628236296 | 22  |
| 102       | nibir       | 1728236296 | 20  |
| 103       | masud       | 1727236296 | 22  |
| 104       | hridoy      | 1625236296 | 23  |
| 105       | tushar      | 1623236296 | 24  |
| 107       | jiku        | 1729236296 | 21  |
| 108       | taslima     | 1630236296 | 23  |

☐ Show all | Number of rows: 25 ▾ | Filter rows:

Query results operations



## Insert Values in Full Join:



The screenshot displays a database management interface. On the left, a tree view shows a database named 'sqa' containing two tables, 'student' and 'university'. The 'university' table is currently selected. The central pane shows a SQL query designed to perform a full join between the 'student' and 'university' tables. The query uses a combination of left and right joins followed by a union. The right-hand pane lists the columns that will be returned by the query: StudentID, StudentName, City, PostalCode, Number, and INCOME.

```
1 SELECT university.StudentID, student.Studentname,university.city FROM
university left JOIN student ON university.StudentName=student.Studentname
union SELECT university.StudentID, student.Studentname, university.city FROM
university right JOIN student ON university.StudentName=student.Studentname;
```

StudentID  
StudentName  
City  
PostalCode  
Number  
INCOME

## Full Join Table:

New

bugresistance

databasename

information\_schema

mysql

performance\_schema

phpmyadmin

sqa

- New
- + student
- + university

test

```
SELECT university.StudentID, student.Studentname, university.city FROM university left JOIN student ON university.StudentName=student.Studentname union SELECT university.StudentID, student.Studentname, university right JOIN student ON university.StudentName=student.Studentname;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP](#) ]

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

+ Options

| StudentID | Studentname | city    |
|-----------|-------------|---------|
| 101       | torin       | tangail |
| 105       | tushar      | tangail |
| 107       | jiku        | tangail |
| 102       | nibir       | rangpur |
| 108       | taslima     | dhaka   |
| 103       | masud       | dhaka   |
| 104       | hridoy      | dhaka   |
| 106       | NULL        | comilla |
| 109       | NULL        | tangail |