

# **JDBC\_Crud\_Operation\_Perfome**

# 1 create the data with data structure

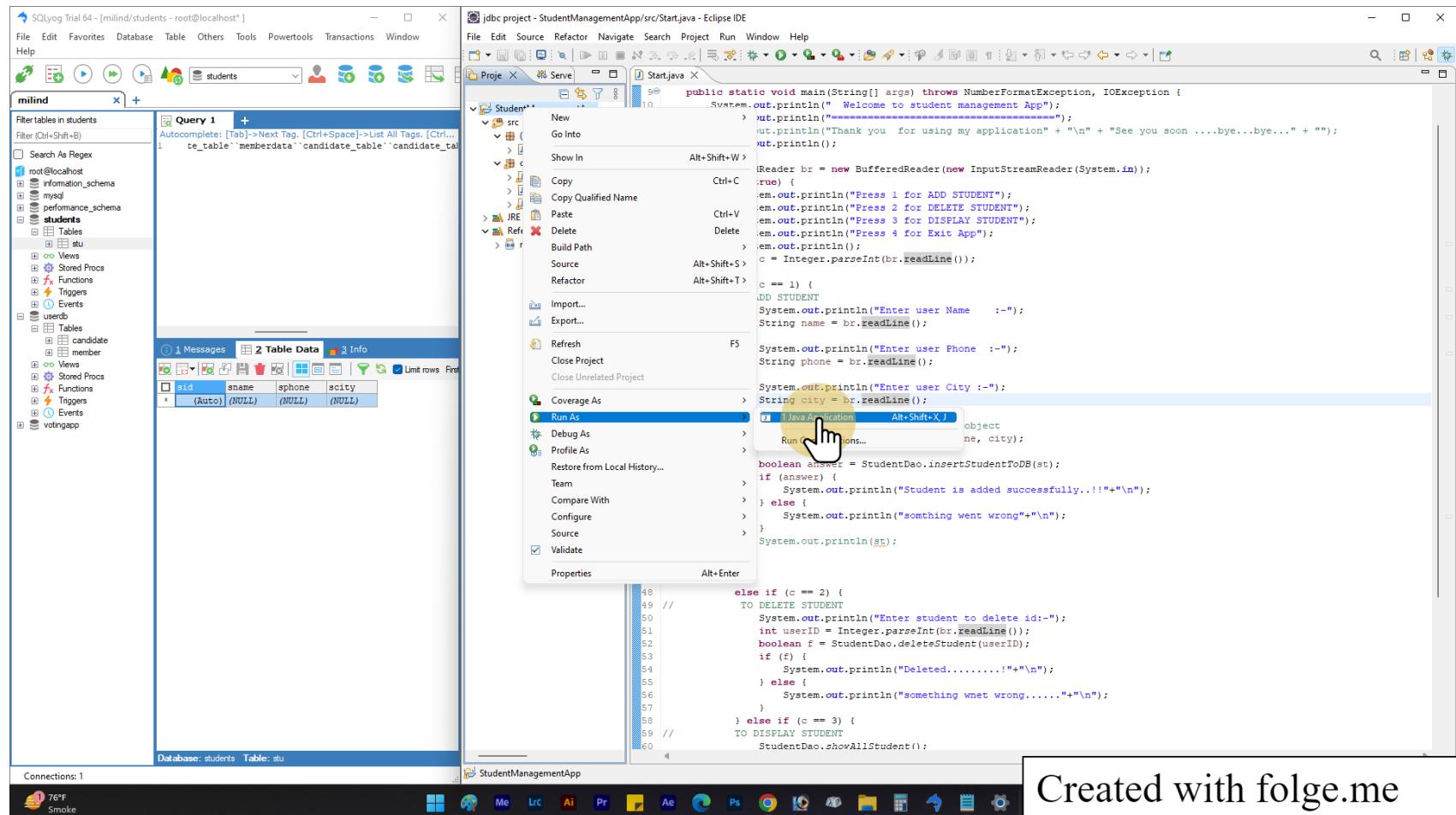
The screenshot shows the SQLyog interface with the following details:

- Left Panel:** Shows the database structure for the 'milind' schema, including tables like 'students', 'userdb', and 'votingapp'.
- Central Window:**
  - Query 1:** New Table
  - Table Name:** stu
  - Database:** students
  - Columns:**

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
sid	int	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>						
sname	varchar	44											
sphone	varchar	44											
scity	varchar	44											
  - Buttons:** 1 Columns, 2 Indexes, 3 Foreign Keys, 4 Check Constraint, 5 Advanced, 6 SQL Preview, Hide language options, Save, Revert.
- Bottom Taskbar:** Shows various application icons including Microsoft Office, Adobe Creative Suite, and system utilities.

Created with folge.me

## 2 run the project



Created with folge.me

### 3 press input as per requirement

The screenshot displays a development environment with three main windows:

- SQLyog Trial 64 - milind/students**: A database management tool showing the 'stu' table with the following data:

sid	sname	sphone	scity
(Auto)	(NULL)	(NULL)	(NULL)

- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The source code for the application. It contains a main method that prints a welcome message and a menu. It uses a BufferedReader to read user input and a StudentDao object to interact with the database.

```

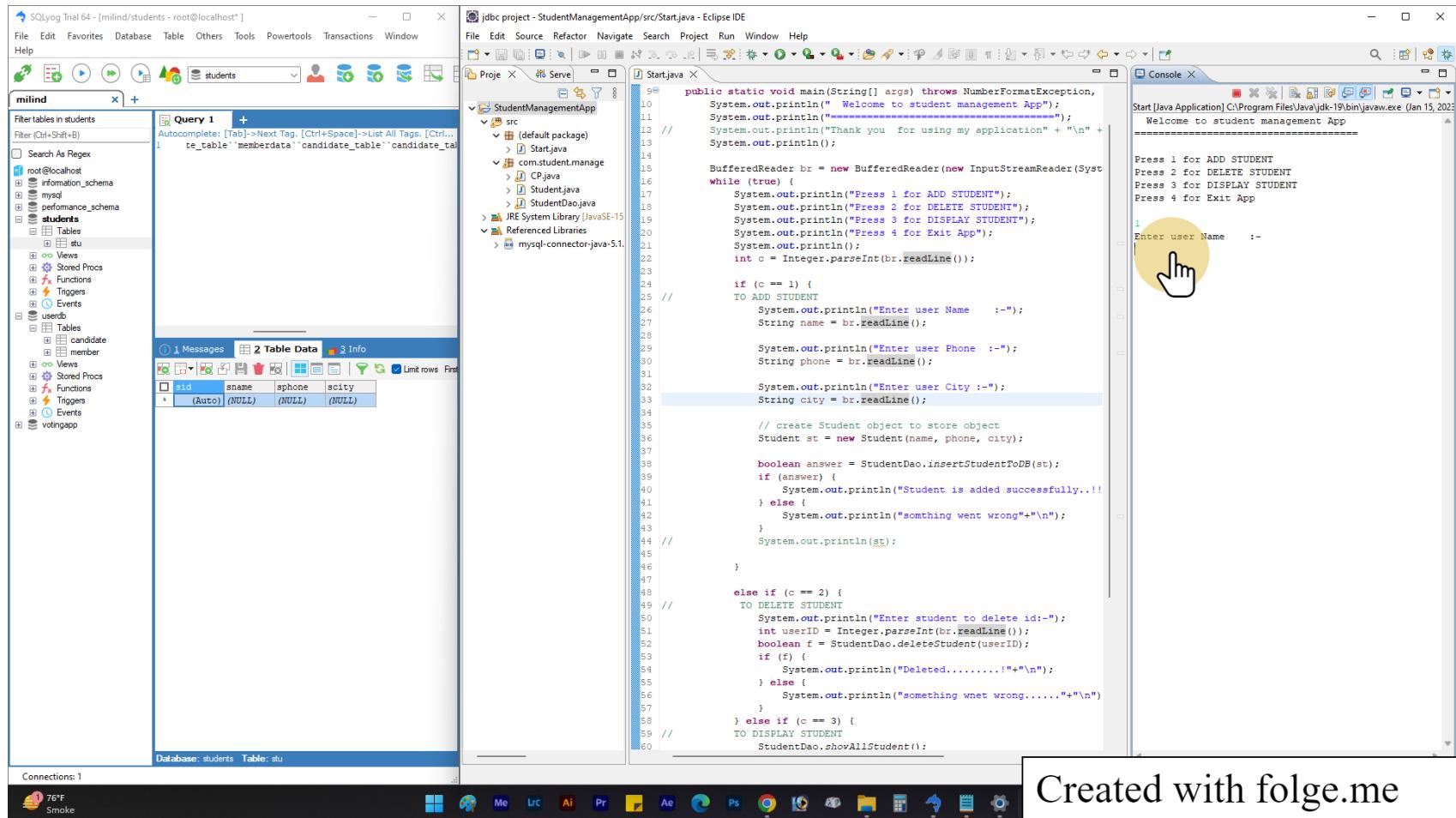
public static void main(String[] args) throws NumberFormatException,
System.out.println(" Welcome to student management App");
System.out.println("-----");
System.out.println("Thank you for using my application" + "\n" +
System.out.println();
BufferedReader br = new BufferedReader(new InputStreamReader(Syst
while (true) {
System.out.println("Press 1 for ADD STUDENT");
System.out.println("Press 2 for DELETE STUDENT");
System.out.println("Press 3 for DISPLAY STUDENT");
System.out.println("Press 4 for Exit App");
System.out.println();
int c = Integer.parseInt(br.readLine());
if (c == 1) {
TO ADD STUDENT
System.out.println("Enter user Name :-");
String name = br.readLine();
System.out.println("Enter user Phone :-");
String phone = br.readLine();
System.out.println("Enter user City :-");
String city = br.readLine();
// create Student object to store object
Student st = new Student(name, phone, city);
boolean answer = StudentDao.insertStudentToDB(st);
if (answer) {
System.out.println("Student is added successfully...!!"
} else {
System.out.println("somthing went wrong"+ "\n");
}
System.out.println(st);
}
else if (c == 2) {
TO DELETE STUDENT
System.out.println("Enter student to delete id:-");
int userID = Integer.parseInt(br.readLine());
boolean f = StudentDao.deleteStudent(userID);
if (f) {
System.out.println("Deleted.....!" + "\n");
} else {
System.out.println("something wnet wrong....." + "\n");
}
} else if (c == 3) {
TO DISPLAY STUDENT
StudentDao.showAllStudent();
}
}
}

```

- Console**: A terminal window showing the application's output. It prints a welcome message and a menu. A yellow circle with a hand cursor points to the 'Press 1 for ADD STUDENT' line.

Created with folge.me

## 4 Enter user name



The screenshot shows a development environment with three main components:

- Eclipse IDE (Top Right):** Displays the `Start.java` file containing the main method of a Java application. The code handles user input for adding, deleting, displaying, or exiting the application.
- SQLyog Trial 64 (Left):** A database management tool connected to the 'students' database. It shows the structure of the 'stu' table, which has columns: sid, sname, sphone, scity. A sample row is shown with values: (Auto), (NULL), (NULL), (NULL).
- Console (Bottom Right):** Shows the output of running the application. It prints a welcome message and a menu with four options: ADD STUDENT, DELETE STUDENT, DISPLAY STUDENT, and EXIT APP. A yellow circle highlights the first option, "Enter user Name :-", followed by a hand cursor icon pointing at it.

```

public static void main(String[] args) throws NumberFormatException,
    System.out.println(" Welcome to student management App");
    System.out.println("-----");
    System.out.println("Thank you for using my application" + "\n" +
    System.out.println();
    BufferedReader br = new BufferedReader(new InputStreamReader(Syst
    while (true) {
        System.out.println("Press 1 for ADD STUDENT");
        System.out.println("Press 2 for DELETE STUDENT");
        System.out.println("Press 3 for DISPLAY STUDENT");
        System.out.println("Press 4 for Exit App");
        System.out.println();
        int c = Integer.parseInt(br.readLine());
        if (c == 1) {
            TO ADD STUDENT
            System.out.println("Enter user Name :-");
            String name = br.readLine();
            System.out.println("Enter user Phone :-");
            String phone = br.readLine();
            System.out.println("Enter user City :-");
            String city = br.readLine();
            // create Student object to store object
            Student st = new Student(name, phone, city);
            boolean answer = StudentDao.insertStudentToDB(st);
            if (answer) {
                System.out.println("Student is added successfully...!!");
            } else {
                System.out.println("something went wrong"+ "\n");
            }
            System.out.println(st);
        }
        else if (c == 2) {
            TO DELETE STUDENT
            System.out.println("Enter student to delete id:-");
            int userID = Integer.parseInt(br.readLine());
            boolean f = StudentDao.deleteStudent(userID);
            if (f) {
                System.out.println("Deleted.....!"+ "\n");
            } else {
                System.out.println("something went wrong....."+ "\n");
            }
        } else if (c == 3) {
            TO DISPLAY STUDENT
            StudentDao.showAllStudent();
        }
    }
}

```

Created with folge.me

5

The screenshot displays a development environment with three main windows:

- SQLyog Trial 64 - milind/students - root@localhost\***: A database browser showing the structure of the 'students' database, including tables like 'stu', 'candidate', 'member', and 'votingapp'. A query editor window titled 'Query 1' shows a simple SELECT statement.
- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The code editor window contains the 'Start.java' file which implements a command-line interface for managing students. It uses BufferedReader to read user input and System.out.println to display messages and options.
- Console**: The terminal window shows the execution of the application. It prints a welcome message and a menu with four options: ADD STUDENT, DELETE STUDENT, DISPLAY STUDENT, and EXIT APP. The user enters '1' to add a student, followed by 'Milind' as the name.

Created with folge.me

## 6 Enter User phone

The screenshot illustrates the development environment for a Java application named 'StudentManagementApp'. The application uses JDBC to interact with a MySQL database named 'students'. The 'stu' table is used to store student information. The Java code in 'Start.java' handles user input through a BufferedReader and performs four main operations based on user choice: ADD STUDENT, DELETE STUDENT, DISPLAY STUDENT, and EXIT APP. The 'Console' window in Eclipse shows the application's output and the user's input 'Milind'.

```

public static void main(String[] args) throws NumberFormatException,
System.out.println(" Welcome to student management App");
System.out.println("-----");
System.out.println("Thank you for using my application" + "\n" +
System.out.println();
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
while (true) {
    System.out.println("Press 1 for ADD STUDENT");
    System.out.println("Press 2 for DELETE STUDENT");
    System.out.println("Press 3 for DISPLAY STUDENT");
    System.out.println("Press 4 for Exit App");
    System.out.println();
    int c = Integer.parseInt(br.readLine());
    if (c == 1) {
        TO ADD STUDENT
        System.out.println("Enter user Name :-");
        String name = br.readLine();
        System.out.println("Enter user Phone :-");
        String phone = br.readLine();
        System.out.println("Enter user City :-");
        String city = br.readLine();
        // create Student object to store object
        Student st = new Student(name, phone, city);
        boolean answer = StudentDao.insertStudentToDB(st);
        if (answer) {
            System.out.println("Student is added successfully...!!!");
        } else {
            System.out.println("something went wrong"+"\n");
        }
        System.out.println(st);
    }
    else if (c == 2) {
        TO DELETE STUDENT
        System.out.println("Enter student to delete id:-");
        int userID = Integer.parseInt(br.readLine());
        boolean f = StudentDao.deleteStudent(userID);
        if (f) {
            System.out.println("Deleted.....!"+"\n");
        } else {
            System.out.println("something went wrong.....!"+"\n");
        }
    } else if (c == 3) {
        TO DISPLAY STUDENT
        StudentDao.showAllStudent();
    }
}
}

```

Created with folge.me

The screenshot displays a development environment with three main windows:

- SQLyog Trial 64 - milind/students - root@localhost\***: A database browser showing the `students` schema. It lists tables like `stu`, `candidate`, and `member`. A query editor window titled "Query 1" shows the creation of a table named `candidate`.
- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The code editor window contains the `Start.java` file which implements a simple command-line interface for managing students.
- Console X**: The terminal window shows the execution of the application. It prints a welcome message and a menu with options 1 through 4. When option 1 is selected, it prompts for user name ("Milind") and user phone ("1234567890").

```

public static void main(String[] args) throws NumberFormatException,
System.out.println(" Welcome to student management App");
System.out.println("-----");
System.out.println("Thank you for using my application" + "\n" +
System.out.println();
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
while (true) {
    System.out.println("Press 1 for ADD STUDENT");
    System.out.println("Press 2 for DELETE STUDENT");
    System.out.println("Press 3 for DISPLAY STUDENT");
    System.out.println("Press 4 for Exit App");
    System.out.println();
    int c = Integer.parseInt(br.readLine());
    if (c == 1) {
        TO ADD STUDENT
        System.out.println("Enter user Name :-");
        String name = br.readLine();
        System.out.println("Enter user Phone :-");
        String phone = br.readLine();
        System.out.println("Enter user City :-");
        String city = br.readLine();
        // create Student object to store object
        Student st = new Student(name, phone, city);
        boolean answer = StudentDao.insertStudentToDB(st);
        if (answer) {
            System.out.println("Student is added successfully...!!!");
        } else {
            System.out.println("something went wrong"+"\n");
        }
        System.out.println(st);
    }
    else if (c == 2) {
        TO DELETE STUDENT
        System.out.println("Enter student to delete id:-");
        int userID = Integer.parseInt(br.readLine());
        boolean f = StudentDao.deleteStudent(userID);
        if (f) {
            System.out.println("Deleted.....!"+"\n");
        } else {
            System.out.println("something went wrong.....!"+"\n");
        }
    } else if (c == 3) {
        TO DISPLAY STUDENT
        StudentDao.showAllStudent();
    }
}
}

```

Created with folge.me

## 8 Enter user city

The screenshot displays three windows side-by-side:

- Eclipse IDE (Top Right):** Shows the code for `Start.java` which handles user input for adding, deleting, displaying, or exiting the application.
- MySQL Workbench (Left):** Shows the database schema for the `students` database, including tables `stu` and `candidate`, and their columns like `sname`, `sphone`, and `scity`.
- Java Console (Bottom Right):** Shows the application's output. It prints welcome messages and a menu. A user enters "Milind" for the name, "800441905" for the phone number, and "Pune" for the city. A yellow circle highlights the "Enter user City" prompt, and a hand cursor icon points to the "Pune" entry.

```

public static void main(String[] args) throws NumberFormatException,
    System.out.println(" Welcome to student management App");
    System.out.println("-----");
    System.out.println("Thank you for using my application" + "\n" +
    System.out.println();
    BufferedReader br = new BufferedReader(new InputStreamReader(Syst
    while (true) {
        System.out.println("Press 1 for ADD STUDENT");
        System.out.println("Press 2 for DELETE STUDENT");
        System.out.println("Press 3 for DISPLAY STUDENT");
        System.out.println("Press 4 for Exit App");
        System.out.println();
        int c = Integer.parseInt(br.readLine());
        if (c == 1) {
            TO ADD STUDENT
            System.out.println("Enter user Name :-");
            String name = br.readLine();
            System.out.println("Enter user Phone :-");
            String phone = br.readLine();
            System.out.println("Enter user City :-");
            String city = br.readLine();
            // create Student object to store object
            Student st = new Student(name, phone, city);
            boolean answer = StudentDao.insertStudentToDB(st);
            if (answer) {
                System.out.println("Student is added successfully...!!")
            } else {
                System.out.println("somthing went wrong"+ "\n");
            }
            System.out.println(st);
        }
        else if (c == 2) {
            TO DELETE STUDENT
            System.out.println("Enter student to delete id:-");
            int userID = Integer.parseInt(br.readLine());
            boolean f = StudentDao.deleteStudent(userID);
            if (f) {
                System.out.println("Deleted.....!" + "\n");
            } else {
                System.out.println("something wnet wrong....." + "\n");
            }
        } else if (c == 3) {
            TO DISPLAY STUDENT
            StudentDao.showAllStudent();
        }
    }
}

```

Created with folge.me

The screenshot displays a development environment with three main windows:

- SQLyog Trial 64 - milind/students - root@localhost\***: A database browser showing the `students` schema with tables like `stu`, `candidate`, and `member`. A query editor window titled "Query 1" shows the creation of a table named `te_table`.
- jdb project - StudentManagementApp/src/Start.java - Eclipse IDE**: The code editor window contains the `Start.java` file which implements a simple command-line interface for managing students.
- Console**: The terminal window shows the execution of the application. It prints a welcome message and a menu with options 1 through 4. When option 1 is selected, it prompts for user name, phone, and city, then creates a new student object and inserts it into the database. The output shows the successful insertion of a student named Milind with phone 800741905 and city Ulhasnagar.

Created with folge.me

10

The screenshot displays a development environment with three main windows:

- SQLyog Trial 64 - milind/students - root@localhost\***: A database browser showing the `students` schema. It lists tables like `stu`, `candidate`, and `member`. A query editor window titled "Query 1" shows the creation of a table named `candidate`.
- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The code editor window contains the `Start.java` file which implements a simple command-line interface for managing students.
- Console**: The terminal window shows the application's execution. It prints a welcome message and a menu. When "Press 1 for ADD STUDENT" is selected, it prompts for user name, phone, and city, then adds a new student record to the database. The output concludes with a thank you message and a goodbye message.

```

public static void main(String[] args) throws NumberFormatException,
                                         IOException {
    System.out.println(" Welcome to student management App");
    System.out.println("-----");
    System.out.println("Thank you for using my application" + "\n" +
                      "-----");
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    while (true) {
        System.out.println("Press 1 for ADD STUDENT");
        System.out.println("Press 2 for DELETE STUDENT");
        System.out.println("Press 3 for DISPLAY STUDENT");
        System.out.println("Press 4 for Exit App");
        System.out.println();
        int c = Integer.parseInt(br.readLine());
        if (c == 1) {
            TO ADD STUDENT
            System.out.println("Enter user Name :-");
            String name = br.readLine();
            System.out.println("Enter user Phone :-");
            String phone = br.readLine();
            System.out.println("Enter user City :-");
            String city = br.readLine();
            // create Student object to store object
            Student st = new Student(name, phone, city);
            boolean answer = StudentDao.insertStudentToDB(st);
            if (answer) {
                System.out.println("Student is added successfully...!!!");
            } else {
                System.out.println("something went wrong"+"\n");
            }
            System.out.println(st);
        }
        else if (c == 2) {
            TO DELETE STUDENT
            System.out.println("Enter student to delete id:-");
            int userID = Integer.parseInt(br.readLine());
            boolean f = StudentDao.deleteStudent(userID);
            if (f) {
                System.out.println("Deleted.....!"+"\n");
            } else {
                System.out.println("something went wrong.....!"+"\n");
            }
        } else if (c == 3) {
            TO DISPLAY STUDENT
            StudentDao.showAllStudent();
        }
    }
}

```

Created with folge.me

# 11 data is added successfully

The screenshot displays two windows side-by-side. On the left is the SQLyog Trial 64 database management tool, connected to the 'milind' database. It shows the 'stu' table with one row of data: sid=1, sname='Milind', sphone='8007441905', scity='Ulhasnagar'. A cursor is hovering over this row. On the right is the Eclipse IDE running a Java application named 'StudentManagementApp'. The 'Start.java' file is open, containing the main method which adds a student to the database. The 'Console' tab shows the application's output, including user input and system messages. The output includes:

```

Welcome to student management App
=====
Press 1 for ADD STUDENT
Press 2 for DELETE STUDENT
Press 3 for DISPLAY STUDENT
Press 4 for Exit App

1
Enter user Name :-
Milind
Enter user Phone :-
8007441905
Enter user City :-
Ulhasnagar
Sun Jan 15 21:02:55 IST 2023 WARN: Establishing SSL
Student is added successfully...!!

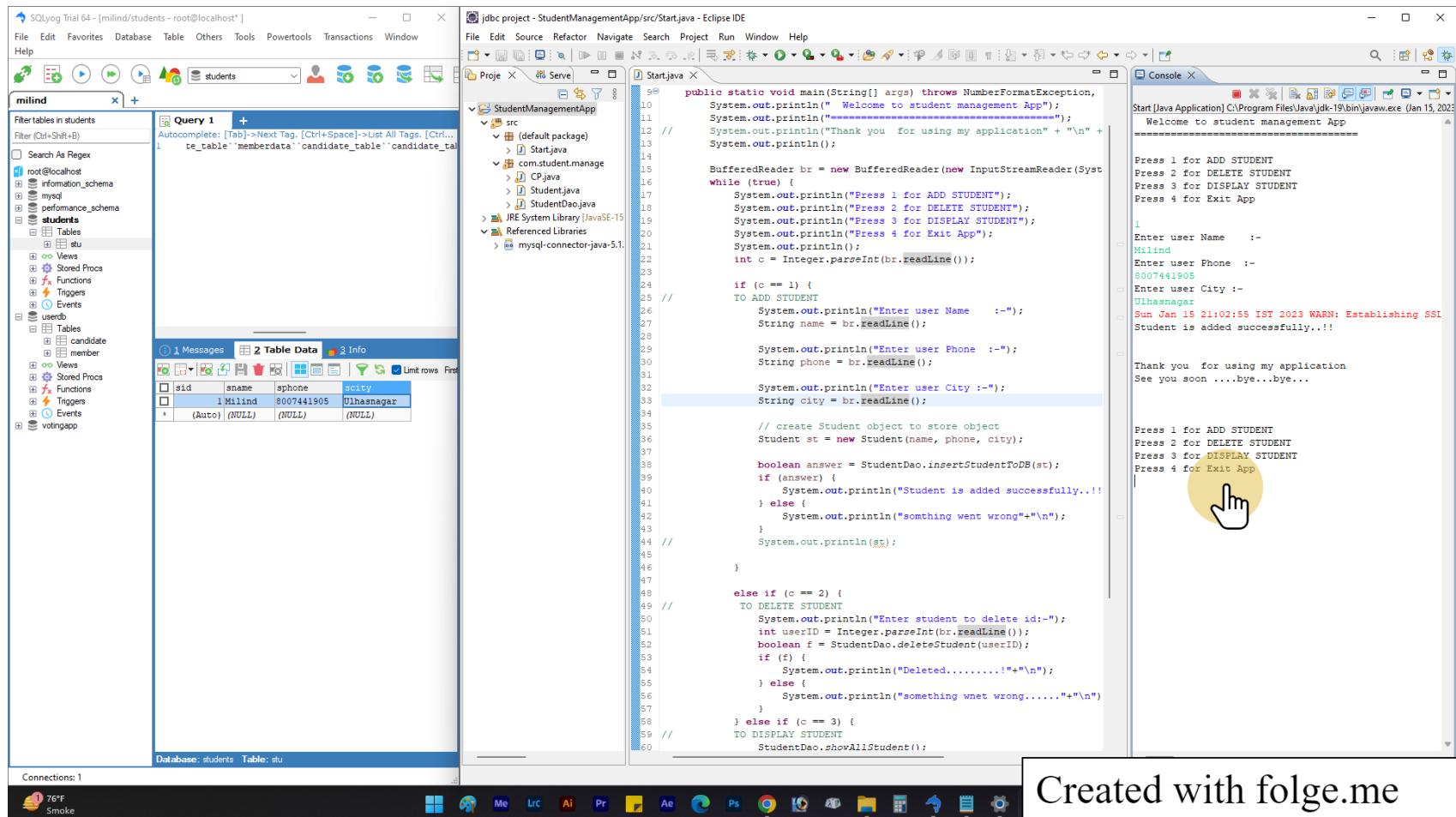
Thank you for using my application
See you soon....bye...bye...

Press 1 for ADD STUDENT
Press 2 for DELETE STUDENT
Press 3 for DISPLAY STUDENT
Press 4 for Exit App

```

Created with folge.me

12



The screenshot displays a dual-monitor setup. On the left monitor, the SQLyog Trial 64 database interface is open, showing the 'students' database with tables like 'stu', 'candidate', and 'member'. A query result set for the 'stu' table is shown, with one row: sid 1, sname Milind, sphone 8007441905, scity Ulhasnagar.

On the right monitor, the Eclipse IDE is running a Java project named 'StudentManagementApp'. The 'Start.java' file contains the main method for the application:

```

public static void main(String[] args) throws NumberFormatException,
System.out.println(" Welcome to student management App");
System.out.println("-----");
System.out.println("Thank you for using my application" + "\n" +
System.out.println();
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
while (true) {
    System.out.println("Press 1 for ADD STUDENT");
    System.out.println("Press 2 for DELETE STUDENT");
    System.out.println("Press 3 for DISPLAY STUDENT");
    System.out.println("Press 4 for Exit App");
    System.out.println();
    int c = Integer.parseInt(br.readLine());
    if (c == 1) {
        TO ADD STUDENT
        System.out.println("Enter user Name :-");
        String name = br.readLine();
        System.out.println("Enter user Phone :-");
        String phone = br.readLine();
        System.out.println("Enter user City :-");
        String city = br.readLine();
        // create Student object to store object
        Student st = new Student(name, phone, city);
        boolean answer = StudentDao.insertStudentToDB(st);
        if (answer) {
            System.out.println("Student is added successfully...!!!");
        } else {
            System.out.println("something went wrong"+"\n");
        }
        System.out.println(st);
    }
    else if (c == 2) {
        TO DELETE STUDENT
        System.out.println("Enter student to delete id:-");
        int userID = Integer.parseInt(br.readLine());
        boolean f = StudentDao.deleteStudent(userID);
        if (f) {
            System.out.println("Deleted.....!"+"\n");
        } else {
            System.out.println("something went wrong.....!"+"\n");
        }
    } else if (c == 3) {
        TO DISPLAY STUDENT
        StudentDao.showAllStudent();
    }
}

```

The application's output is displayed in the 'Console' window:

```

=====
Press 1 for ADD STUDENT
Press 2 for DELETE STUDENT
Press 3 for DISPLAY STUDENT
Press 4 for Exit App
1
Enter user Name :-
Milind
Enter user Phone :-
8007441905
Enter user City :-
Ulhasnagar
Sun Jan 15 21:02:55 IST 2023 WARN: Establishing SSL
Student is added successfully...!!

Thank you for using my application
See you soon....bye...bye...

Press 1 for ADD STUDENT
Press 2 for DELETE STUDENT
Press 3 for DISPLAY STUDENT
Press 4 for Exit App
|
```

Created with folge.me

# 13 display the enter data

The screenshot displays three windows illustrating the execution of a Java application for managing student data:

- SQLyog Trial 64 - milind/students - root@localhost\***: A MySQL management tool showing the 'stu' table. The table has columns: sid, sname, sphone, scity. One row is present: 1 Milind 8007441905 Ulhasnagar.
- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The Java code for the application. It includes imports for java.util, javax.swing, and StudentManagementApp. The main method handles user input for adding, deleting, displaying, or exiting the application.
- Console Window**: Shows the application's output. It prompts the user to choose an option (1-4). Upon selecting option 1, it asks for user name, phone number, and city, then adds the student to the database. The console also shows the current date and time (Sun Jan 15 21:02:55 IST 2023) and a warning about establishing SSL.

```

public static void main(String[] args) throws NumberFormatException,
                                         IOException {
    System.out.println(" Welcome to student management App");
    System.out.println("-----");
    System.out.println("Thank you for using my application" + "\n" +
                      "-----");
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    while (true) {
        System.out.println("Press 1 for ADD STUDENT");
        System.out.println("Press 2 for DELETE STUDENT");
        System.out.println("Press 3 for DISPLAY STUDENT");
        System.out.println("Press 4 for Exit App");
        System.out.println();
        int c = Integer.parseInt(br.readLine());
        if (c == 1) {
            TO ADD STUDENT
            System.out.println("Enter user Name :-");
            String name = br.readLine();
            System.out.println("Enter user Phone :-");
            String phone = br.readLine();
            System.out.println("Enter user City :-");
            String city = br.readLine();
            // create Student object to store object
            Student st = new Student(name, phone, city);
            boolean answer = StudentDao.insertStudentToDB(st);
            if (answer) {
                System.out.println("Student is added successfully...!!!");
            } else {
                System.out.println("Something went wrong" + "\n");
            }
            System.out.println(st);
        }
        else if (c == 2) {
            TO DELETE STUDENT
            System.out.println("Enter student to delete id:-");
            int userID = Integer.parseInt(br.readLine());
            boolean f = StudentDao.deleteStudent(userID);
            if (f) {
                System.out.println("Deleted.....!" + "\n");
            } else {
                System.out.println("Something went wrong.....!" + "\n");
            }
        } else if (c == 3) {
            TO DISPLAY STUDENT
            StudentDao.showAllStudent();
        }
    }
}

```

Created with folge.me

## 14 delete the data successfully

The screenshot shows three windows illustrating a Java application for managing student data:

- SQLyog Trial 64 - milind/students - root@localhost\***: A MySQL management tool showing the 'stu' table with one record: id=1, sname='Milind', sphone='8007441905', scity='Ulhasnagar'.
- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The Java code for the application. It includes methods for adding, deleting, displaying, and exiting the application.
- Console Window**: The application's output window showing the process of adding a student and then deleting it.

```

public static void main(String[] args) throws NumberFormatException,
                                         IOException {
    System.out.println(" Welcome to student management App");
    System.out.println("-----");
    System.out.println("Thank you for using my application" + "\n" +
    System.out.println();

    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    while (true) {
        System.out.println("Press 1 for ADD STUDENT");
        System.out.println("Press 2 for DELETE STUDENT");
        System.out.println("Press 3 for DISPLAY STUDENT");
        System.out.println("Press 4 for Exit App");
        System.out.println();
        int c = Integer.parseInt(br.readLine());

        if (c == 1) {
            TO ADD STUDENT
            System.out.println("Enter user Name :-");
            String name = br.readLine();
            System.out.println("Enter user Phone :-");
            String phone = br.readLine();
            System.out.println("Enter user City :-");
            String city = br.readLine();
            // create Student object to store object
            Student st = new Student(name, phone, city);

            boolean answer = StudentDao.insertStudentToDB(st);
            if (answer) {
                System.out.println("Student is added successfully...!!!");
            } else {
                System.out.println("Something went wrong" + "\n");
            }
            System.out.println(st);
        }

        else if (c == 2) {
            TO DELETE STUDENT
            System.out.println("Enter student to delete id:-");
            int userID = Integer.parseInt(br.readLine());
            boolean f = StudentDao.deleteStudent(userID);
            if (f) {
                System.out.println("Deleted.....!" + "\n");
            } else {
                System.out.println("Something went wrong.....!" + "\n");
            }
        } else if (c == 3) {
            TO DISPLAY STUDENT
            StudentDao.showAllStudent();
        }
    }
}

```

**Console Output:**

```

Press 1 for ADD STUDENT
Press 2 for DELETE STUDENT
Press 3 for DISPLAY STUDENT
Press 4 for Exit App

3
Sun Jan 15 21:03:31 IST 2023 WARN: Establishing SSL
ID :1
Name :Milind
Phone:8007441905
City :Ulhasnagar
-----
Thank you for using my application
See you soon ....bye...bye...

Press 1 for ADD STUDENT
Press 2 for DELETE STUDENT
Press 3 for DISPLAY STUDENT
Press 4 for Exit App

2
Enter student to delete id:
1
Sun Jan 15 21:04:04 IST 2023 WARN: Establishing SSL
Deleted.....
|
Thank you for using my application
See you soon ....bye...

```

Created with folge.me

15

The screenshot shows the SQLyog Trial 64 interface. On the left, the database schema is displayed under the 'milind' database. The 'students' schema contains a single table named 'stu'. The 'stu' table has four columns: 'sid', 'sname', 'sphone', and 'scity'. The table data view shows one row with all columns set to NULL.

sid	sname	sphone	scity
(Auto)	(NULL)	(NULL)	(NULL)

At the bottom of the interface, there is a toolbar with various icons for file operations, and a status bar indicating 'Database: students Table: stu' and '0 row(s)'. The bottom right corner of the window has a watermark that reads 'Created with folge.me'.

# 16 exit from the app press input 4

The screenshot displays a development environment with three main windows:

- SQLyog Trial 64 - milind/students - root@localhost\***: A database management tool showing the 'stu' table with one row: id 1, sname 'Milind', sphone '8007441905', scity 'Ulhasnagar'.
- Eclipse IDE - jdbc project - StudentManagementApp/src/Start.java**: The Java code for the application. It includes imports for java.util, javax.swing, and StudentManagementApp. The main method handles user input for adding, deleting, displaying, or exiting the application.
- Console Window**: Shows the application's output. It starts with a welcome message, adds a student successfully, and then enters a loop where it prints instructions and waits for input. It handles deletion, display, and exit requests correctly, but there is a bug in the addition logic that prints the same student details again after insertion.

```

public static void main(String[] args) throws NumberFormatException,
                                         IOException {
    System.out.println(" Welcome to student management App");
    System.out.println("-----");
    System.out.println("Thank you for using my application" + "\n" +
    System.out.println();

    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    while (true) {
        System.out.println("Press 1 for ADD STUDENT");
        System.out.println("Press 2 for DELETE STUDENT");
        System.out.println("Press 3 for DISPLAY STUDENT");
        System.out.println("Press 4 for Exit App");
        System.out.println();
        int c = Integer.parseInt(br.readLine());

        if (c == 1) {
            TO ADD STUDENT
            System.out.println("Enter user Name :-");
            String name = br.readLine();
            System.out.println("Enter user Phone :-");
            String phone = br.readLine();
            System.out.println("Enter user City :-");
            String city = br.readLine();
            // create Student object to store object
            Student st = new Student(name, phone, city);

            boolean answer = StudentDao.insertStudentToDB(st);
            if (answer) {
                System.out.println("Student is added successfully...!!!");
            } else {
                System.out.println("Something went wrong" + "\n");
            }
            System.out.println(st);
        }

        else if (c == 2) {
            TO DELETE STUDENT
            System.out.println("Enter student to delete id:-");
            int userID = Integer.parseInt(br.readLine());
            boolean f = StudentDao.deleteStudent(userID);
            if (f) {
                System.out.println("Deleted.....!" + "\n");
            } else {
                System.out.println("Something went wrong.....!" + "\n");
            }
        } else if (c == 3) {
            TO DISPLAY STUDENT
            StudentDao.showAllStudent();
        }
    }
}

```

Created with folge.me