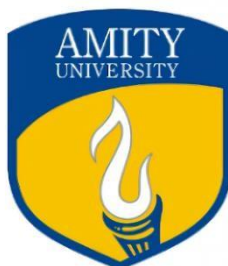


AMITY UNIVERSITY, PATNA
AMITY INSTITUTE OF INFORMATION TECHNOLOGY

Advanced Java Lab
LAB FILE

BCA



Name : Nishant Kumar
Program/Semester : BCA – 5 ‘A’
Enroll. Number : A45304821038
Submitted to : Dr. Naveen Kumar Singh

CRUD OPERATIONS

Problem description :

Develop a simple Java application that utilizes JDBC (Java Database Connectivity) to establish a connection with a relational database system and perform basic CRUD (Create, Read, Update, Delete) operations on a specified database table.

The application should:

1. Provide options to perform CRUD operations including inserting new records into the database table, retrieving existing records from the table based on specified criteria, updating records in the table and deleting records from the table.
2. Implement error handling to manage connection failures and database operation exceptions gracefully.

The application should focus on simplicity and functionality, serving as a basic template for JDBC usage in CRUD operations

DESIGN

The design of the problem statement for creating a simple Java application that establishes JDBC connection and performs CRUD operations involves several key components and considerations:

1. User Interface Design :

Upon running the application, users will be presented with a menu containing 5 options, with 4 of them representing crud operations and the last option for exiting the application gracefully. Based on the user's choice, the application will invoke the appropriate method from the Student class to perform the CRUD operation.

2. Database Connection Management:

The application needs to establish a JDBC connection with the relational database system using the correct connection details.

3. Error Handling:

Error handling should be implemented to manage exceptions during database operations.

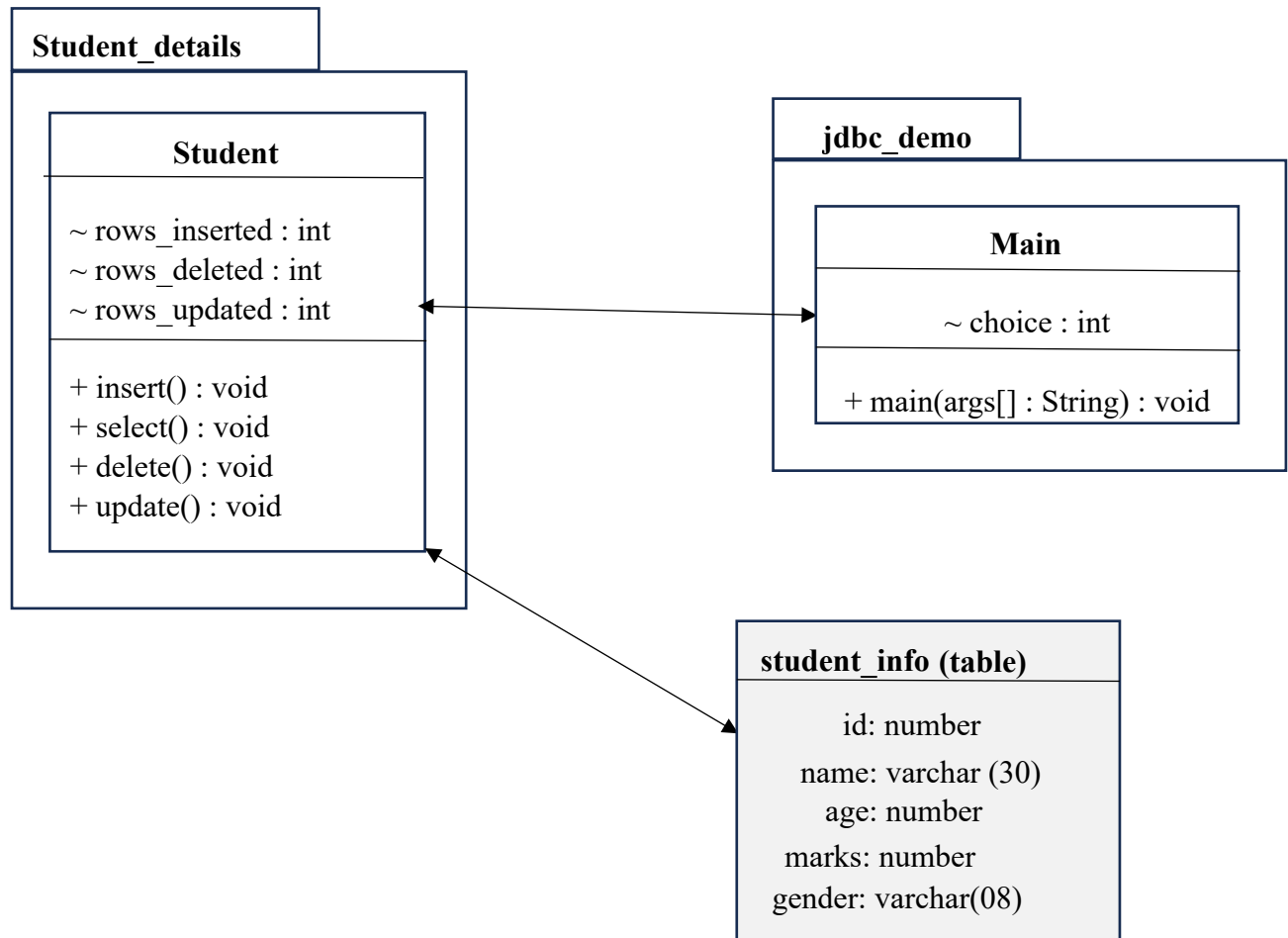
4. Code Modularity and Maintainability:

The application's code should be modular and well-organized, following best practices in software design and development. It should be easy to maintain and extend, allowing for future enhancements or modifications without significant refactoring.

5. Class Diagram:

A class diagram is crucial for design purposes as it visually illustrates the structure, relationships, and behavior of classes within a system. It aids in organizing and conceptualizing software components, facilitating communication among developers, guiding implementation, and ensuring consistency and scalability

throughout the design process. Here's a class diagram demonstrating our problem statement -



CODE

Student.java

```
package bca.model;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Student {

    public Student() {
        super();
        // TODO Auto-generated constructor stub
    }

    public void addStudent(Connection con, Scanner sc) throws SQLException
    {
        //create statement
        Statement st = con.createStatement();

        //read student details
        System.out.println("Enter Student Id: ");
        int id = sc.nextInt();

        System.out.println("Enter Student Name: ");
        String name = sc.next();

        System.out.println("Enter Student Age: ");
        int age = sc.nextInt();
```

```

System.out.println("Enter Student Marks: ");
double marks = sc.nextDouble();

System.out.println("Enter Student Gender: ");
String gender = sc.next();

//create sql squery string
String query = String.format("Insert Into student_info values(%d, '%s', %d, %f, '%s') ", id, name, age, marks, gender);

//execute sql query
int rows = st.executeUpdate(query);

System.out.println(rows + " record inserted!!!");

}

public void displayStudents(Connection con) throws SQLException {
    Statement st = con.createStatement();

    ResultSet rs = st.executeQuery("select * from student_info");

    while(rs.next()) {
        System.out.println(rs.getInt(1)+ "\t"+rs.getString(2)+ "\t"+
rs.getInt(3)+"\t"+rs.getDouble(4)+"\t"+rs.getString(5));
    }
}

public void updateStudentName(Connection con, Scanner sc) throws
SQLException {
    Statement st = con.createStatement();

    System.out.println("Enter Student ID: ");

```

```

int id = sc.nextInt();

System.out.println("Enter Student New Name: ");

String name = sc.next();

String query = String.format("update student_info set name='%s' where
id = %d", name, id);

int rowsAffected = st.executeUpdate(query);

System.out.println(rowsAffected+" recored updated!!!");

}

public void deleteStudent(Connection con, Scanner sc) throws
SQLException {

Statement st = con.createStatement();

System.out.println("Enter Student ID: ");

int id = sc.nextInt();

int rowAffected = st.executeUpdate("delete from student_info where id
= "+id);

System.out.println(rowAffected + " recored deleted!!!");

}

}

```

User Interface.java

```

package bca.drive;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Scanner;

```

```
import bca.model.Student;

public class Main {

    public static void main(String[] args) throws ClassNotFoundException,
        SQLException {
        // TODO Auto-generated method stub

        //1. load and register
        Class.forName("com.mysql.cj.jdbc.Driver");

        //2
        String url = "jdbc:mysql://localhost:3306/student";
        String username = "root";
        String pwd = "Mysql@2024";
        Connection con = DriverManager.getConnection(url, username, pwd);
        Scanner sc = new Scanner(System.in);
        Student s = new Student();

        //insert
        //s.addStudent(con, sc);

        while(true) {
            menu();

            int choice = sc.nextInt();

            switch(choice) {
                case 1: s.addStudent(con, sc);
                    break;

                case 2: s.displayStudents(con);
                    break;

                case 3: s.updateStudentName(con, sc);
                    break;
```



```
        case 4: s.deleteStudent(con, sc);
                break;

        case 5:
                System.out.println("Bye Bye ...");
                System.exit(0);
        default:
                System.out.println("Wrong Choice...");
        }
}

}

public static void menu() {
    System.out.println("-----Menu-----");
    System.out.println("1. Add New Student");
    System.out.println("2. Display All Students");
    System.out.println("3. Update Name of Student");
    System.out.println("4. Delete a Student");
    System.out.println("5. Exit");
    System.out.println("Your Choice...");
}
}
```

INPUT/OUTPUT

Describing the table

Query 1 x student_info student_info

Limit to 1000 rows

```
1
2 • desc student_info;
3
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	Field	Type	Null	Key	Default	Extra
▶	id	int	YES		NULL	
	name	varchar(50)	YES		NULL	
	age	int	YES		NULL	
	marks	double	YES		NULL	
	gender	varchar(10)	YES		NULL	

Selecting the table

Query 1 x student_info student_info

Limit to 1000 rows

```
1 • select * from student_info;
2
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	id	name	age	marks	gender
▶	38	Nishant	20	98	Male
	56	Priya	20	78	Female
	34	Aryan	22	80	Male

Display operation

```
Console ×
Main (1) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Feb 4, 2024, 5:10:18 PM) [pid: 2716]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Priya    20      80.0    Female
38      Nishant  20      90.0    Male
47      Aryan   22      99.0    Male
```

Insert operation

Before insertion :

```
Console ×
Main (1) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Feb 4, 2024, 5:08:38 PM) [pid: 13364]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Aditi    20      80.0    Female
```

After insertion :

```
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
1
Enter Student Id:
38
Enter Student Name:
Nishant
Enter Student Age:
20
Enter Student Marks:
90
Enter Student Gender:
Male
1 record inserted!!!

Console ×
Main (1) [Java Application] [pid: 19240]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Aditi    20      80.0    Female
38      Nishant  20      90.0    Male
47      Aryan   22      99.0    Male
```

Delete operation

Before Deletion :

```
Console ×
Main (1) [Java Application] [pid: 19240]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Aditi    20      80.0    Female
38      Nishant  20      90.0    Male
47      Aryan   22      99.0    Male
```

After Deletion :

```
Console ×
Main (1) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Feb 4, 2024, 5:10:18 PM) [pid: 2716]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Priya    20      80.0    Female
38      Nishant  20      90.0    Male
47      Aryan   22      99.0    Male
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
4
Enter Student ID:
47
1 record deleted!!!
```

```
Console X
Main (1) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Feb 4, 2024, 5:18:10 PM) [pid: 19676]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Priya    20      80.0    Female
38      Nishant  20      90.0    Male
```

Update operation

Before updation :

```
Console X
Main (1) [Java Application] [pid: 19240]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Aditi    20      80.0    Female
38      Nishant  20      90.0    Male
47      Aryan    22      99.0    Male
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
3
Enter Student ID:
56
Enter Student New Name:
Priya
1 record updated!!!
```

After updation :

```
Console ×
Main (1) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Feb 4, 2024, 5:10:18 PM) [pid: 2716]
-----Menu-----
1. Add New Student
2. Display All Students
3. Update Name of Student
4. Delete a Student
5. Exit
Your Choice...
2
56      Priya    20      80.0    Female
38      Nishant  20      90.0    Male
47      Aryan    22      99.0    Male
```