**AMITY UNIVERSITY, PATNA**

**AMITY INSTITUTE OF INFORMATION TECHNOLOGY**

**Advanced Java**



Name : Harshit Kumar

Semester : BCA – 6th‘A’

Enroll. Number : A45304821029

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 DESCRIPTION

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 :**

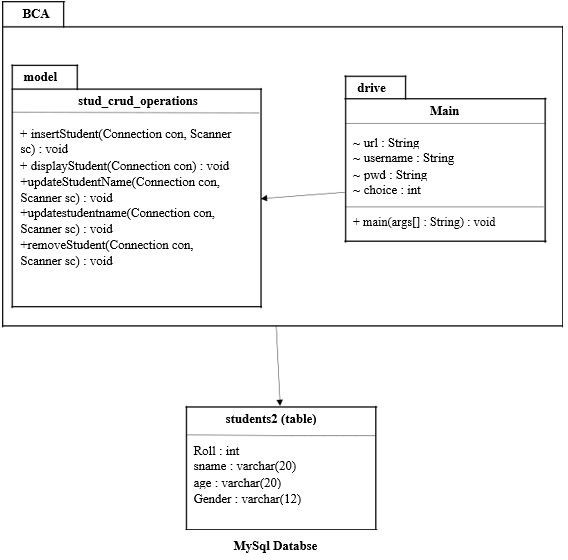
The Java code is a console-based application that manages student records, allowing users to view, insert, delete, or update student data. The program uses a switch-case structure to execute different operations based on user input, interacting with a `Student` class that likely represents a student database model. The code also includes exception handling for `ClassNotFoundException` and `SQLException`.

## Database Connection Management:

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

## 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.



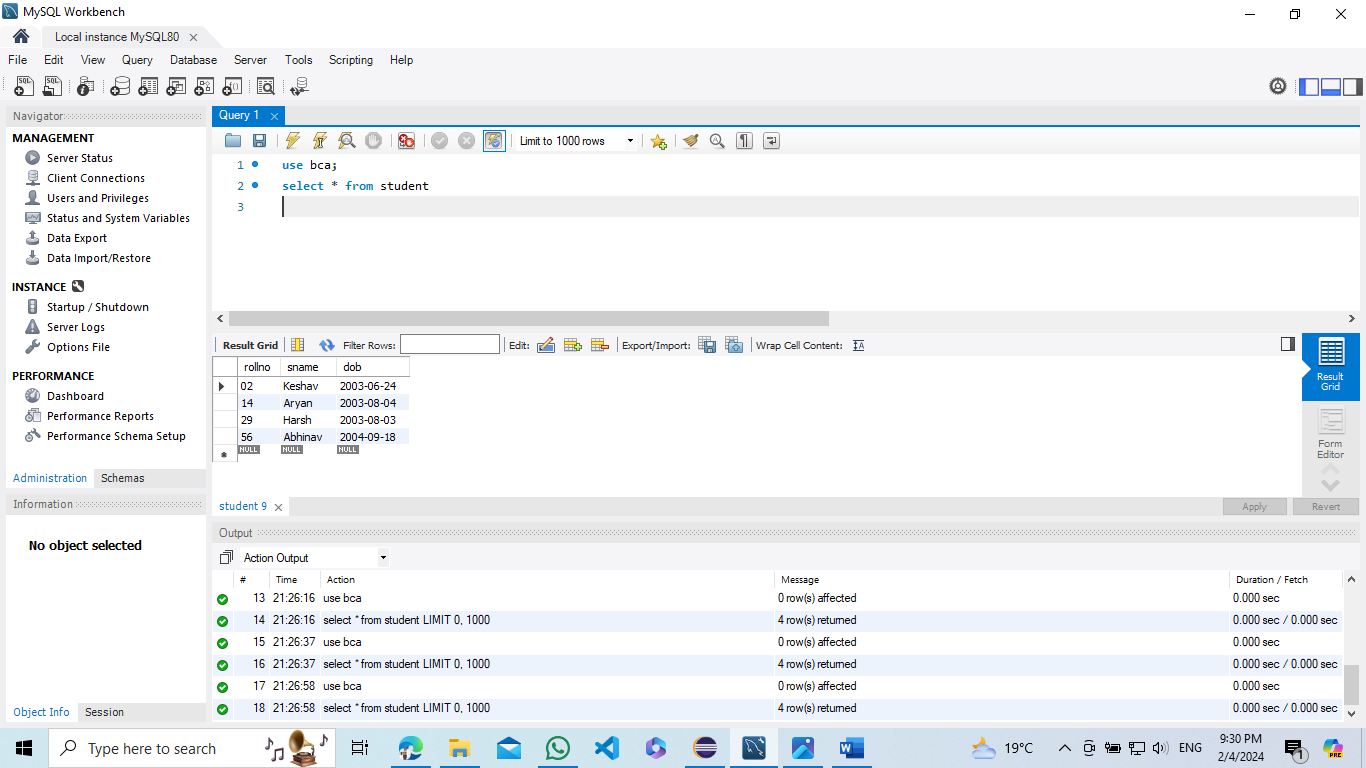
Student(table)

Rollno : (int)

Name : varchar(25)

DOB : date

# SCHEMA

****

**CODE**

#### Student.java

package bca.Model;

import java.sql.\*;

import java.util.Scanner;

public class Student

{

public void Select() throws ClassNotFoundException, SQLException

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BCA","root","Harsh@2003");

Statement st = con.createStatement();

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

System.out.println("rollno\t\tName\t\t\tDOB");

while (rs.next())

{

System.out.println(rs.getString(1)+"\t"+rs.getString(2)+"\t\t"+rs.getDate(3));

}

con.close();

}

public void Delete(String rollno) throws ClassNotFoundException, SQLException

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BCA","root","Harsh@2003");

String query = "Delete from Student where rollno = ?";

PreparedStatement pst = con.prepareStatement(query);

pst.setString(1, rollno);

int affected\_rows = pst.executeUpdate();

System.out.println("Affected Rows :" + affected\_rows);

con.close();

}

public void Insert(String rollno , String name , String date) throws ClassNotFoundException, SQLException

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BCA","root","Harsh@2003");

String query = "Insert into Student values (? , ? ,?)";

PreparedStatement pst = con.prepareStatement(query);

pst.setString(1, rollno);

pst.setString(2, name);

pst.setString(3, date);

int affected\_rows = pst.executeUpdate();

System.out.println("Affected Rows :" + affected\_rows);

con.close();

}

public void Update(String rollno , String name , String date) throws ClassNotFoundException, SQLException

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BCA","root","Harsh@2003");

String query = "Update Student set SName = ? , DOB = ? where rollno = ?";

PreparedStatement pst = con.prepareStatement(query);

pst.setString(1, name);

pst.setString(2, date);

pst.setString(3, rollno);

int affected\_rows = pst.executeUpdate();

System.out.println("Affected Rows :" + affected\_rows);

con.close();

}

public void Select\_Specific(String rollno) throws ClassNotFoundException, SQLException

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BCA","root","Harsh@2003");

String query = "select \* from Student where rollno = ?";

PreparedStatement pst = con.prepareStatement(query);

pst.setString(1, rollno);

ResultSet rs = pst.executeQuery();

System.out.println("rollno\t\tName\t\tDOB");

while (rs.next())

{

System.out.println(rs.getString(1)+"\t"+rs.getString(2)+"\t\t"+rs.getDate(3));

}

con.close();

}

}

#### Main.java

package bca.Drive;

import java.sql.\*;

import java.util.Scanner;

import bca.Model.Student;

public class App

{

public static void main(String[] args) throws ClassNotFoundException, SQLException

{

Scanner sc = new Scanner(System.in);

Student s = new Student();

System.out.println("Enter Your Choice (Any Other Number beside 1,2,3,4, means Exit\n1 -> View\n2 -> Insert\n3 -> Delete\n4 -> Update\n");

int ch = sc.nextInt();

switch (ch)

{

case 1:

{

s.Select();

break;

}

case 2:

{

System.out.print("Enter The Details of the Student\nRoll = ");

String roll = sc.next();

System.out.print("Name = ");

String name = sc.next();

System.out.print("DOB (YYYY-MM-DD) = ");

String date = sc.next();

s.Insert(roll, name, date);

break;

}

case 3:

{

System.out.println("Enter The Roll whose Details needs to be Deleted");

String roll = sc.next();

s.Delete(roll);

break;

}

case 4:

{

System.out.println("Enter The Roll of the Student Whose Data needs to be Updated");

String roll = sc.next();

System.out.println("Present Details");

s.Select\_Specific(roll);

System.out.print("\nEnter The Details To Be Updated (If Same Data Needs to be Kept for some field then Rewrite it as same\nName = ");

String name = sc.next();

System.out.print("DOB (YYYY-MM-DD) = ");

String dob = sc.next();

s.Update(roll, name, dob);

System.out.println("\n-----New Data-----");

s.Select\_Specific(roll);

break;

}

default:

throw new IllegalArgumentException("Unexpected value: " + ch);

}

}

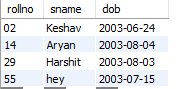
}

# INPUT/OUTPUT

## View operation :

## 

**Display operation :**

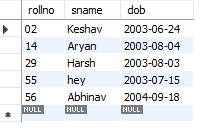
****

**Insert operation :**

**A white background with black text

Description automatically generated**

**Display operation :**

****

**Delete operation :**

**A screenshot of a computer screen

Description automatically generated**

**Display operation :**

**A screenshot of a computer

Description automatically generated**

**Update operation :**

**A screenshot of a computer

Description automatically generated**

**Display operation :**

A screenshot of a computer

Description automatically generated