ADVANCED JAVA

AMITY INSTITUTE OF INFORMATION TECHNOLOGY

LAB-1

Name : Priya Kumari
Course : Advanced java
Program/Semester : BCA – 6 'B'
Enrollment Number : A45304821056

Submitted to :-

Dr. Naveen Kumar Singh

Department:- Amity Institude Of Information Technology
Session:- 2021-2024

CRUD OPERATIONS

Problem description :

The problem description for JDBC CRUD operations typically involves creating, reading, updating, and deleting records in a relational database using Java Database Connectivity (JDBC).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.

3. SQL Syntax Error: Double check your SQL queries for syntax errors,Incorrect queries can lead to unexpected results or failures.

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(“Add New Employe”,”Display all Employe”,”Update Name of Employe,”Delete an Employe”)
and the last option for exiting the application gracefully. Based on the user's choice, the application will invoke the appropriate method from the Employe class to perform the CRUD operation.

2. Database Connection Management:

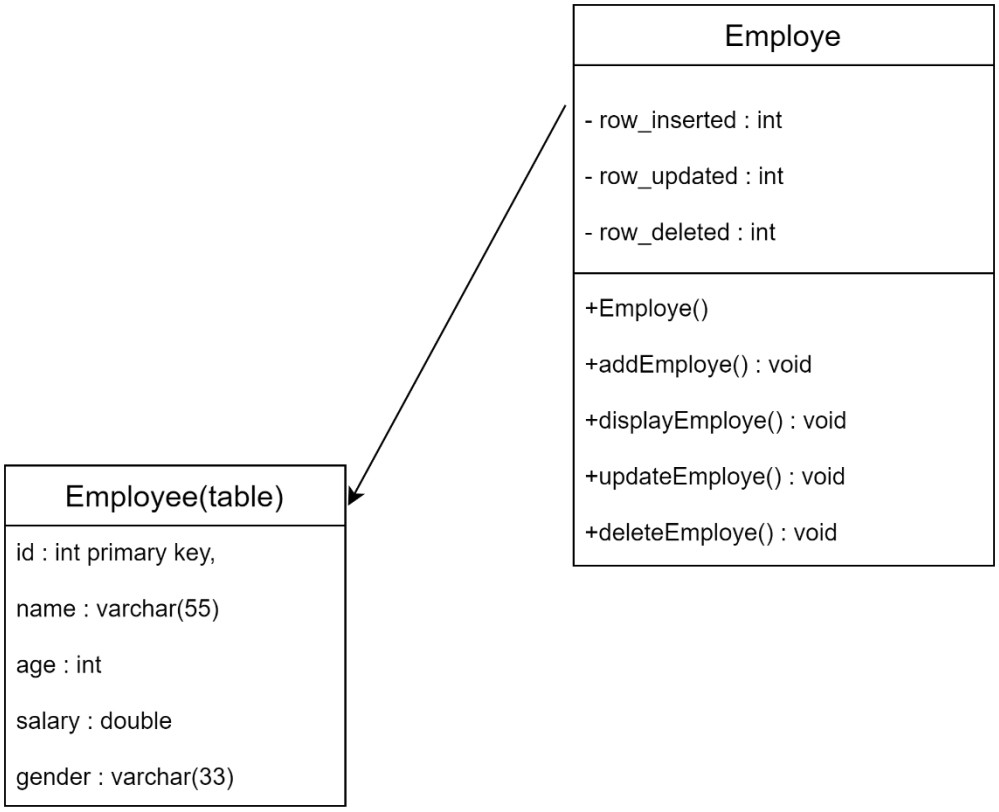
The application needs to establish a JDBC connection with the relational database system using the correct connection details.

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

Employe.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 Employe {

	public Employe() {
		super();
		// TODO Auto-generated constructor stub
	}
	public void addEmploye(Connection con, Scanner sc) throws
SQLException {
		//create statement
		Statement st = con.createStatement();

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

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

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

		System.out.println("Enter Employe salary : ");
		double salary = sc.nextDouble();

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

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

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

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

	}

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

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

		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 updateEmployeName(Connection con, Scanner sc) throws
SQLException {
		Statement st = con.createStatement();
		System.out.println("Enter Employe ID: ");
		int id = sc.nextInt();
		System.out.println("Enter Employe New Name: ");
		String name = sc.next();

		String query = String.format("update Employee set name='%s'
where id = %d", name, id);
		int rowsAffected = st.executeUpdate(query);
		System.out.println(rowsAffected+" recored updated!!!");

	}

	public void deleteEmploye(Connection con, Scanner sc) throws
SQLException {
		Statement st = con.createStatement();
		System.out.println("Enter Employee ID: ");
		int id = sc.nextInt();

		int rowAffected = st.executeUpdate("delete from Employee where
id = "+id);
		System.out.println(rowAffected + " recored deleted!!!");
		System.out.println("Row deleted");
	}

}
```

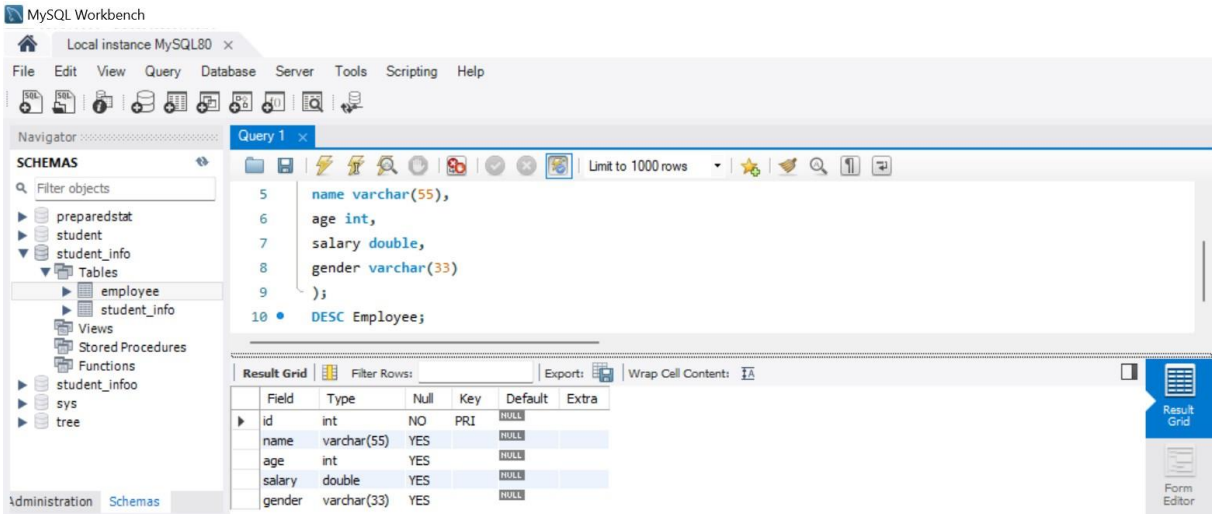
Main.java

```
package Bca.Drive;

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

import Bca.Model.Employe;

public class Mainn {

	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/empdetails";
		String username = "root";
		String pwd = "Root@123";
		Connection con = DriverManager.getConnection(url, username,
pwd);
		Scanner sc = new Scanner(System.in);
		Employe e = new Employe();
		//insert
		//s.addStudent(con, sc);
		while(true) {
```

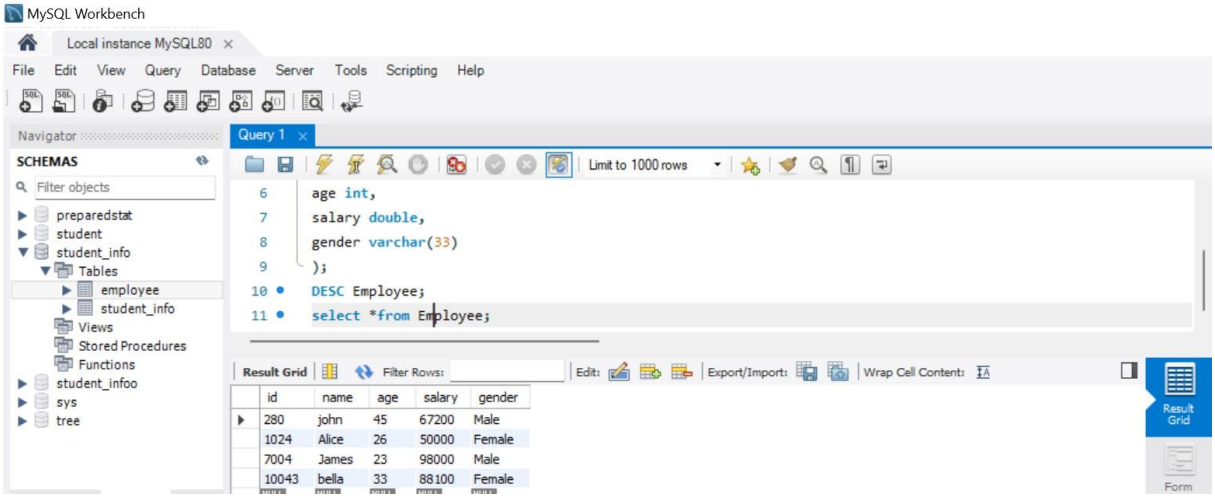
```
                menu();
                int choice = sc.nextInt();
                switch(choice) {
                case 1: e.addEmploye(con,sc);
                      break;

                case 2: e.displayEmploye(con);
                      break;

                case 3: e.updateEmployeName(con, sc);
                      break;

                case 4: e.deleteEmploye(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("------------Operation Perform-----------");
            System.out.println("1. Add New Employe");
            System.out.println("2. Display All Employe");
            System.out.println("3. Update Name of Employe");
            System.out.println("4. Delete a Employe");
            System.out.println("5. Exit");
            System.out.println("Enter any above number of Your Choice...");
        }

    }
```

INPUT/OUTPUT

Describe The Table



Select The Table



CRUD Operation perform

Inserting operation

```
eclipse-workspace - Bca-6B/src/Bca/Model/Employe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Target Window Help
database servlet
Console ×
<terminated> Mainn [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (10-Feb-2024, 11:39:54 am – 11:43:39 am) [pid: 8700]
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...

1
Enter Employe Id:
1024
Enter Employe Name:
Alice
Enter Employe Age:
26
Enter Employe salary :
50000
Enter Employe Gender:
Female
1 record inserted!!!
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
1
Enter Employe Id:
280
Enter Employe Name:
John
Enter Employe Age:
29
Enter Employe salary :
```

Display

```
eclipse-workspace - Bca-6B/src/Bca/Model/Employe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Target Window Help
database servlet
Console ×
<terminated> Mainn [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (10-Feb-2024, 11:39:54 am – 11:43:39 am) [pid: 8700]
Enter Employe Name:
Bella
Enter Employe Age:
33
Enter Employe salary :
88100
Enter Employe Gender:
Female
1 record inserted!!!
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
2
280     John    29      65000.0 Male
1024    Alice   26      50000.0 Female
5611    Adward  40      100000.0        Male
7004    James   23      98000.0 Male
10043   Bella   33      88100.0 Female
```

Update Operation

```
eclipse-workspace - Bca-6B/src/Bca/Model/Employe.java - Eclipse IDE
File  Edit  Source  Refactor  Navigate  Search  Project  Run  Target  Window  Help
database servlet
Console
Mainn [Java Application]  [pid: 18820]
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
2
280     John    45      67200.0 Male
1024    Alice   26      50000.0 Female
5611    Adward  40      100000.0        Male
7004    James   23      98000.0 Male
10043   bella   33      88100.0 Female
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
3
Enter Employe ID:
280
Enter Employe New Name:
john
1 recored updated!!!
```

Delete Operation

```
eclipse-workspace - Bca-6B/src/Bca/Model/Employe.java - Eclipse IDE
File  Edit  Source  Refactor  Navigate  Search  Project  Run  Target  Window  Help
database servlet
Console
Mainn [Java Application]  [pid: 18820]
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
2
280     john    45      67200.0 Male
1024    Alice   26      50000.0 Female
5611    Adward  40      100000.0        Male
7004    James   23      98000.0 Male
10043   bella   33      88100.0 Female
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
4
Enter Employee ID:
5611
1 recored deleted!!!
Row deleted
------------Operation Perform-----------
1. Add New Employe
2. Display All Employe
3. Update Name of Employe
4. Delete a Employe
5. Exit
Enter any above number of Your Choice...
2
280     john    45      67200.0 Male
1024    Alice   26      50000.0 Female
7004    James   23      98000.0 Male
10043   bella   33      88100.0 Female
```