

# Menu Driven JDBC Student Management System

```
public class Main_Menu {  
    public static void main(String[] arg){  
        case 1:  
            Query.insert();  
            break;  
        case 2:  
            Query.update();  
            break;  
        case 3:  
            Query.delete();  
            break;  
        case 4:  
            Query.select();  
    }  
}
```

Run Main\_Menu x Main\_Menu x

"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.6\lib\idea\_rt.jar=59952" -Dfile.encoding=UTF-8 -Dsun.java2d.cgs=off

.... MENU .....

Name : Mahesh Nikas, Branch : IT

Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit

1

Connection Succesful

Enter Student\_id

6

Enter the Name

Pawan\_Rathod

Enter Student Cource

Physics

Enter Student\_Age

22

1 Rows Inserted

.... MENU .....

```
public class Main_Menu {  
    public static void main(String[] arg){  
        case 1:  
            Query.insert();  
            break;  
        case 2:  
            Query.update();  
            break;  
        case 3:  
            Query.delete();  
            break;  
        case 4:  
            Query.select();  
    }  
}
```

Run Main\_Menu x Main\_Menu x

1 Rows Inserted

.... MENU .....

Name : Mahesh Nikas, Branch : IT

Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit

2

Connection Succesful

Enter Student\_id

1

Enter the Course for update

AI

1 rows update

.... MENU .....

Name : Mahesh Nikas, Branch : IT

Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit

|

Task1.javaMain.javaREADME.mdDBConnection.javaPropertyLoader.javaMain\_Menu.javaQuery.javadatabase.propertiesTask2.java

5public class Main\_Menu { @ Mahesh Nikas \*  
6public static void main(String[] arg){ @ Mahesh Nikas \*  
15case 1:  
16Query.insert();  
17break;  
18case 2:  
19Query.update();  
20break;  
21case 3:  
22Query.delete();  
23break;  
24case 4:  
25Query.select();

RunMain\_MenuMain\_Menu

1 rows deleted  
.... MENU .....  
Name : Mahesh Nikas, Branch : IT  
Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit  
4  
Connection Succesful  
2 Mayur\_Ughade Python 22  
3 Akash\_Warkhede .Net 22  
4 Anand\_Nagapure Java\_Script 22  
5 Saurav\_Ugalkar Kotlin 22  
6 Pawan\_Rathod Physics 22  
.... MENU .....  
Name : Mahesh Nikas, Branch : IT  
Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit  
|

Task > src > Menu\_Driven > Main\_Menu > main

Task1.javaMain.javaREADME.mdDBConnection.javaPropertyLoader.javaMain\_Menu.javaQuery.javadatabase.propertiesTask2.java

Commit Alt+Oclass Main\_Menu { @ Mahesh Nikas \*  
public static void main(String[] arg){ @ Mahesh Nikas \*  
15case 1:  
16Query.insert();  
17break;  
18case 2:  
19Query.update();  
20break;  
21case 3:  
22Query.delete();  
23break;  
24case 4:  
25Query.select();

RunMain\_MenuMain\_Menu

Enter the Course for update  
AI  
1 rows update  
.... MENU .....  
Name : Mahesh Nikas, Branch : IT  
Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit  
3  
Connection Succesful  
Enter Student\_id for delete  
1  
1 rows deleted  
.... MENU .....  
Name : Mahesh Nikas, Branch : IT  
Choose option 1: insert in students , 2: update in students , 3:Delete Students , 4: select students , 5: exit  
|

Task > src > Menu\_Driven > Main\_Menu > main

# Menu Driven JDBC Student Management System

## Project Overview

- Type: Console-based Java application
- Technology: Java, JDBC, SQL
- Purpose: Perform CRUD operations on students table

## Main Features

- Insert student record
- Update student details
- Delete student record
- Display all students

## Program Structure

- Main\_Menu.java – Displays menu and handles user choice
- Query.java – Contains insert, update, delete, select methods
- DBConnection.java – Establishes database connection
- database.properties – Stores database configuration

## Database Table Used

- Table Name: students
- Columns: student\_id, name, course, age
- Primary Key: student\_id

## JDBC Components Used

- Connection – To connect Java with database
- PreparedStatement – To execute SQL queries safely
- ResultSet – To retrieve data from database

## Execution Flow

- User selects an option from menu
- Selected operation method is called
- SQL query is executed using JDBC
- Result is displayed to the user
- Menu is shown again

## Outcome

- Data is inserted, updated, deleted, or displayed successfully
- Program runs until user chooses exit option