# JAVA SWING BASED – Database Reads - SQL CONNECTIVITY USING JDBC

*A*

*Report*

*Submitted in partial fulfillment of the*
*Requirements for the award of the Degree*
*of*

## BACHELOR OF ENGINEERING IN

## INFORMATION TECHNOLOGY

**By**

**Vaishnavi Sanga <1602-20-737-304>**

**Under the guidance of Ms B. Leelavathy**

**Department of Information Technology Vasavi**
**College of Engineering (Autonomous)**
**(Affiliated to Osmania University)**
**Ibrahimbagh, Hyderabad-31**

**2020-2021**

# BONAFIDE CERTIFICATE

This is to certify that this project report titled

***'Database Reads'***

is a project work of **Vaishnavi Sanga** bearing roll no. 1602-20-737-304 who carried out this project under my supervision in the IV semester for the academic year 2021- 2022

| | |
|---|---|
| Signature | Signature |
| External Examiner | Internal Examiner |

## ABSTRACT:

The database reads provides an information to read for the user. Here we are creating a database with all topics of DBMS and the resource material available which when clicked can be accessed with sample images. So when user accesses the database they can get the information they need.

**Requirement Analysis**

List of tables:

- Student
- DBMS Material
- Syllabus

**List of attributes with their domain types:**

Student:

| | |
|---|---|
| BRANCH | VARCHAR2(20) |
| STUDENT_NAME | VARCHAR2(20) |
| STUDENT_ID | NUMBER |

DBMS Material:

| | |
|---|---|
| AUTHOR | VARCHAR2(20) |
| BOOK_NAME | VARCHAR2(20) |
| IMAGE | BLOB |
| ID | NUMBER |
| CHAPTER_NUMBER | NUMBER |

Syllabus:

| | |
|---|---|
| CHAPTERS | VARCHAR2(15) |
| SUBTOPICS | VARCHAR2(10) |
| CHAPTER_NUMBER | NUMBER |

Accesses:

| | |
|---|---|
| STUDENT_ID | NUMBER |
| ID | NUMBER |

# AIM AND PRIORITY OF THEPROJECT

To create a **Java GUI-based** desktop application that connects students looking for internships with project managers looking for interns. It takes values like student name, student id, gender, symptoms, etc through forms which are then updated in the database using JDBC connectivity.

# ARCHITECTURE AND TECHNOLOGY

## Software used:

Java Eclipse, Oracle 11g Database, Java SE version 13, SQL*Plus.

## Java SWING:

**Java SWING** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) - an API for providing a graphical user interface (GUI) for Java programs.

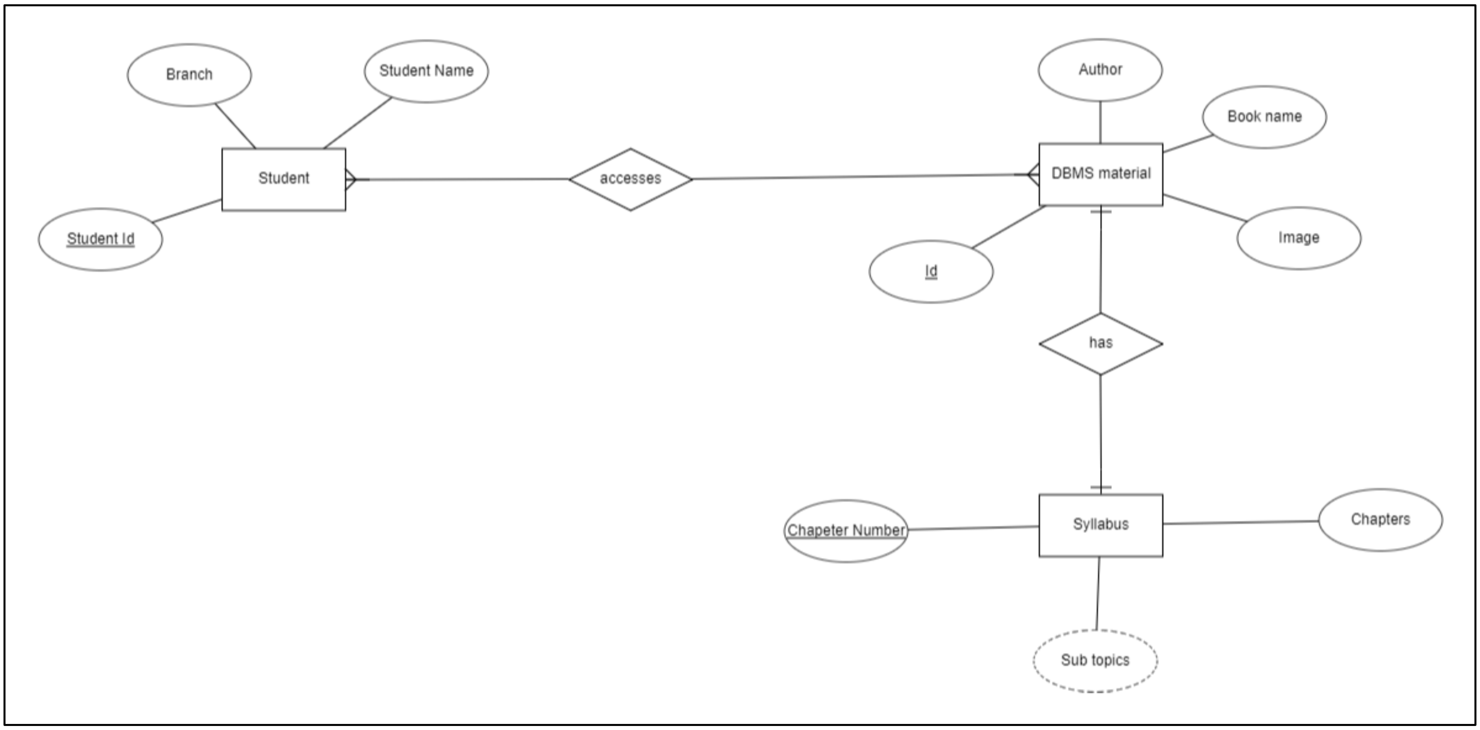
Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

## SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in **Relational** DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

# DESIGN

## Entity Relationship Diagram



# DATABASE DESIGN

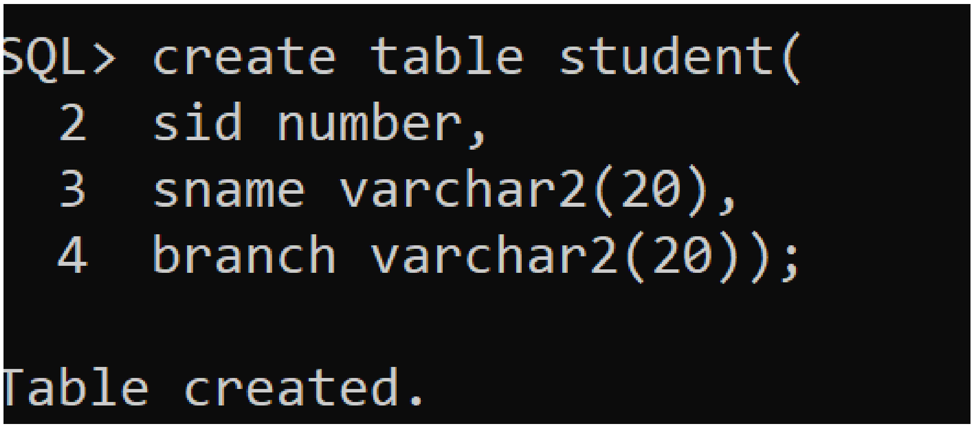
## Mapping Cardinality and Participation Constraints

The database reads provides an information to read for the user. Here we are creating a database with all topics of DBMS and the resource material available which when clicked can be accessed with sample images. So when user accesses the database they can get the information they need.

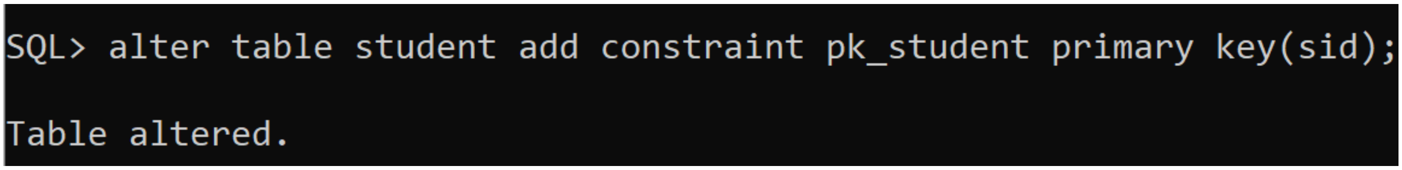
## DDL OPERATIONS

1)

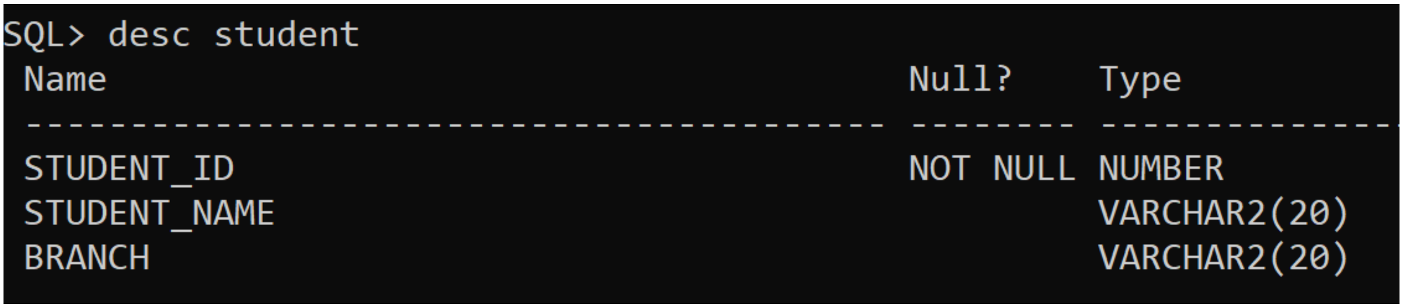
```
create table student(
   sid number,
   sname varchar2(20),
   branch varchar2(20));
alter table student add constraint pk_student primary key(sid);
```



```
SQL> create table student(
  2  sid number,
  3  sname varchar2(20),
  4  branch varchar2(20));

Table created.
```



```
SQL> alter table student add constraint pk_student primary key(sid);

Table altered.
```



```
SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 STUDENT_ID                                NOT NULL NUMBER
 STUDENT_NAME                                       VARCHAR2(20)
 BRANCH                                             VARCHAR2(20)
```

2)

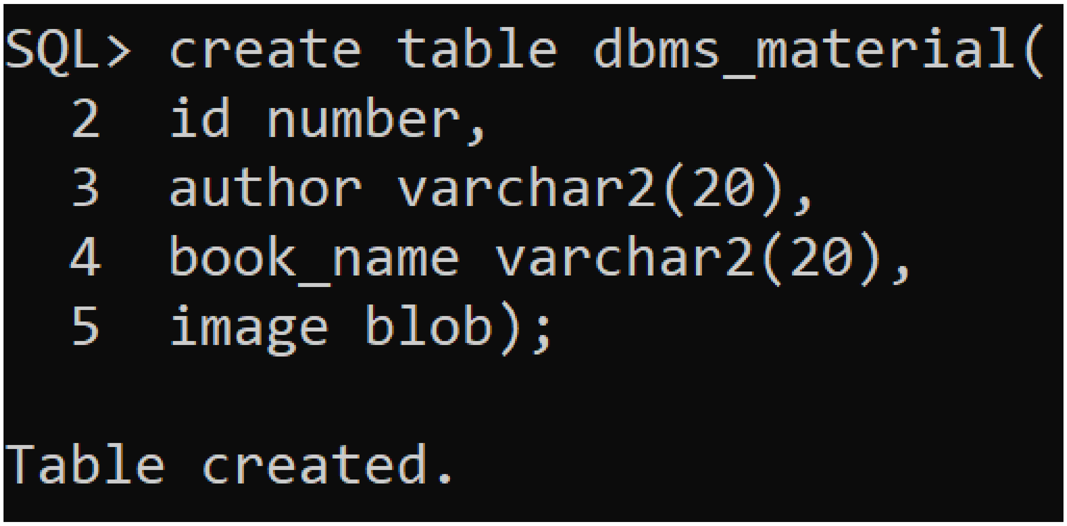
```
create table dbms_material(
   id number,
   author varchar2(20),
   book_name varchar2(20),
```

image blob

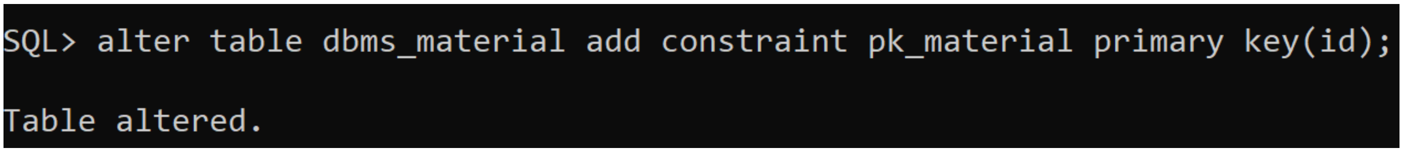
chapter_number number);

alter table dbms_material add constraint pk_material primary key(id);

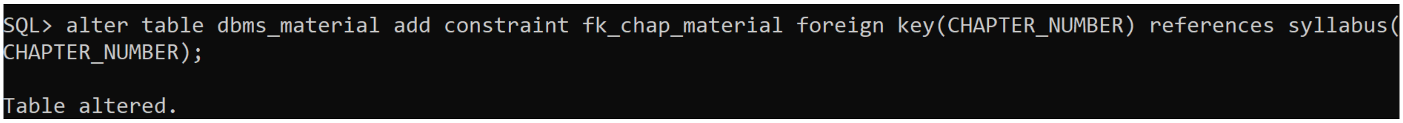
alter table dbms_material add constraint fk_chap_material foreign key(CHAPTER_NUMBER) references syllabus(CHAPTER_NUMBER);



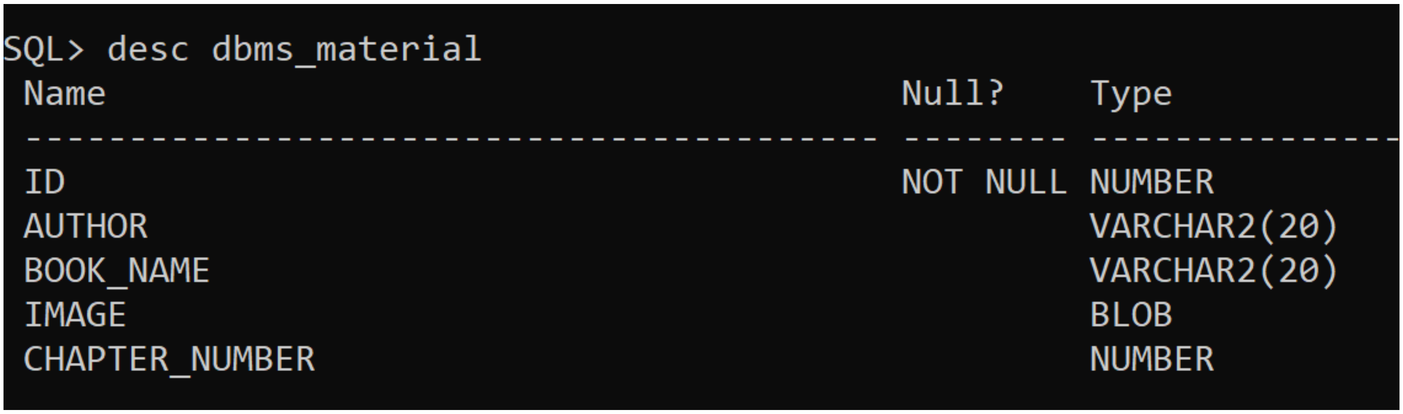
```
SQL> create table dbms_material(
  2  id number,
  3  author varchar2(20),
  4  book_name varchar2(20),
  5  image blob);

Table created.
```



```
SQL> alter table dbms_material add constraint pk_material primary key(id);

Table altered.
```



```
SQL> alter table dbms_material add constraint fk_chap_material foreign key(CHAPTER_NUMBER) references syllabus(
CHAPTER_NUMBER);

Table altered.
```



```
SQL> desc dbms_material
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 AUTHOR                                             VARCHAR2(20)
 BOOK_NAME                                          VARCHAR2(20)
 IMAGE                                              BLOB
 CHAPTER_NUMBER                                     NUMBER
```

3)

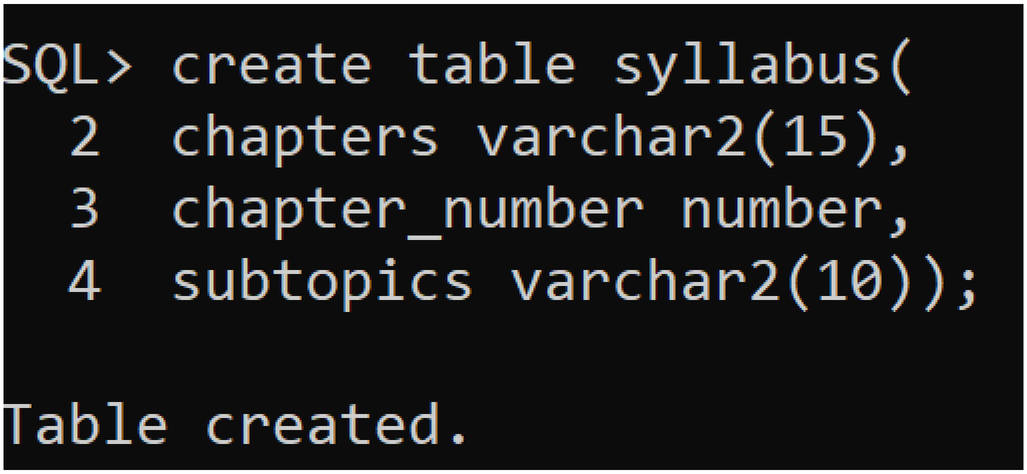
create table syllabus(

chapters varchar2(15),

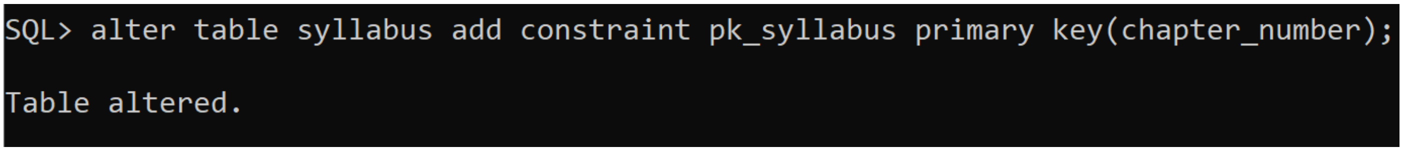
chapter_number number,

subtopics varchar2(10));

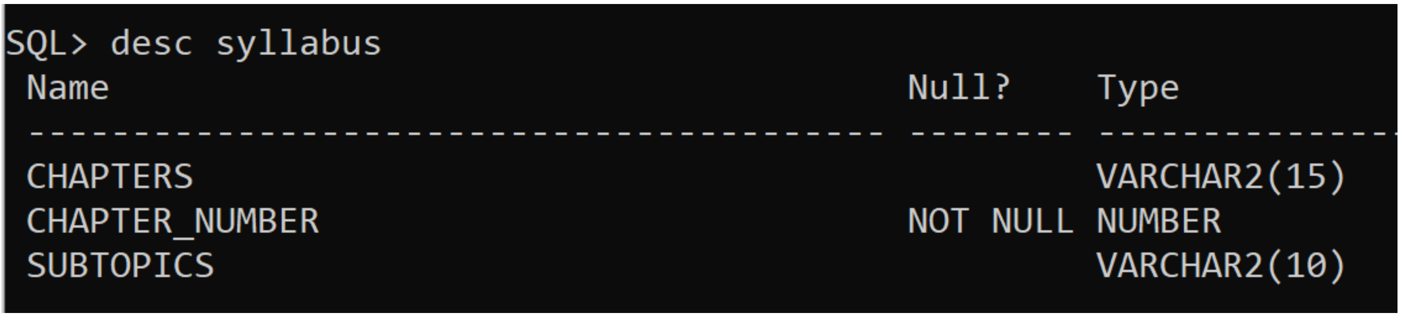
alter table syllabus add constraint pk_syllabus primary key(chapter_number);



```
SQL> create table syllabus(
  2  chapters varchar2(15),
  3  chapter_number number,
  4  subtopics varchar2(10));

Table created.
```



```
SQL> alter table syllabus add constraint pk_syllabus primary key(chapter_number);

Table altered.
```



```
SQL> desc syllabus
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 CHAPTERS                                           VARCHAR2(15)
 CHAPTER_NUMBER                            NOT NULL NUMBER
 SUBTOPICS                                          VARCHAR2(10)
```

4)

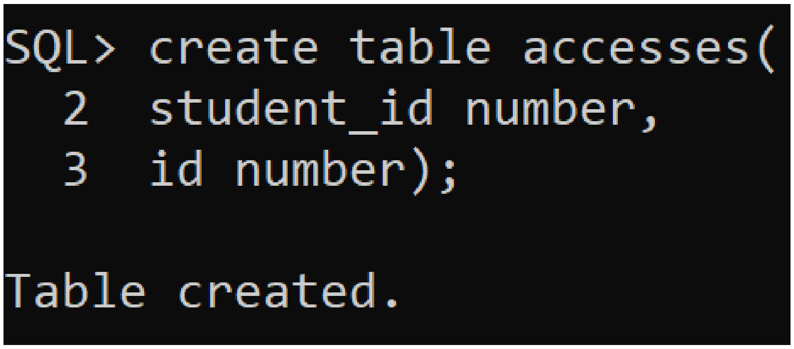
create table accesses(

student_id number,

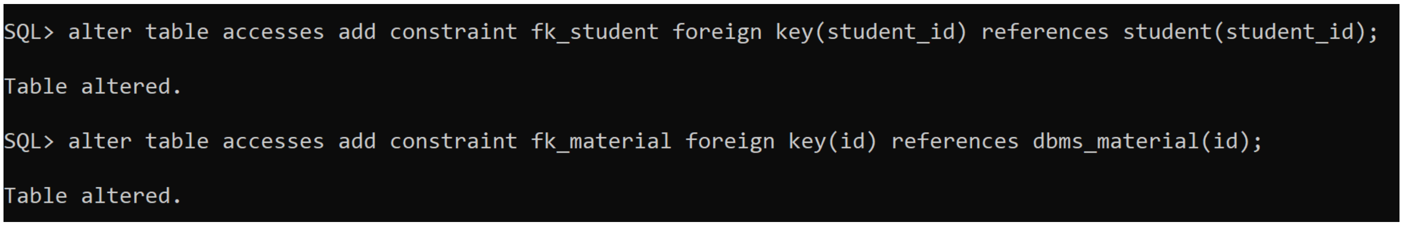
id number);

alter table accesses add constraint fk_student foreign key(student_id) references student(student_id);

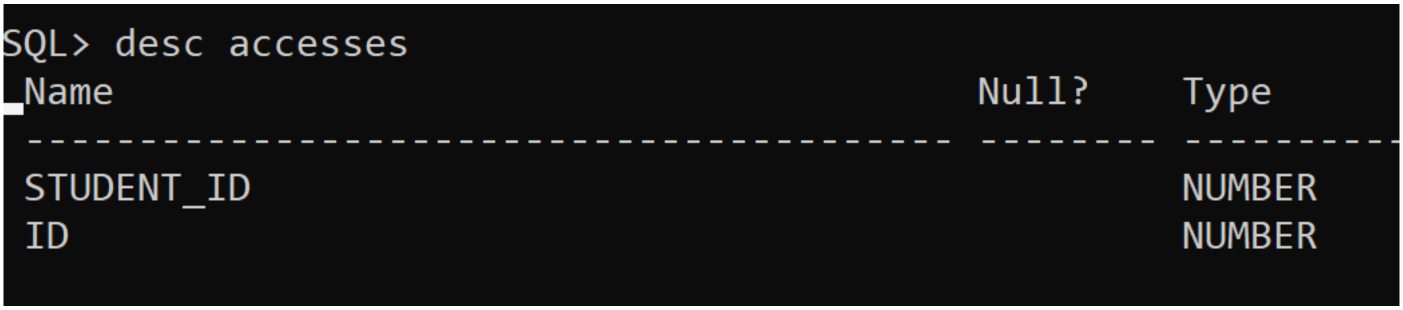
alter table accesses add constraint fk_material foreign key(id) references dbms_material(id);



```
SQL> create table accesses(
  2  student_id number,
  3  id number);

Table created.
```



```
SQL> alter table accesses add constraint fk_student foreign key(student_id) references student(student_id);

Table altered.

SQL> alter table accesses add constraint fk_material foreign key(id) references dbms_material(id);

Table altered.
```

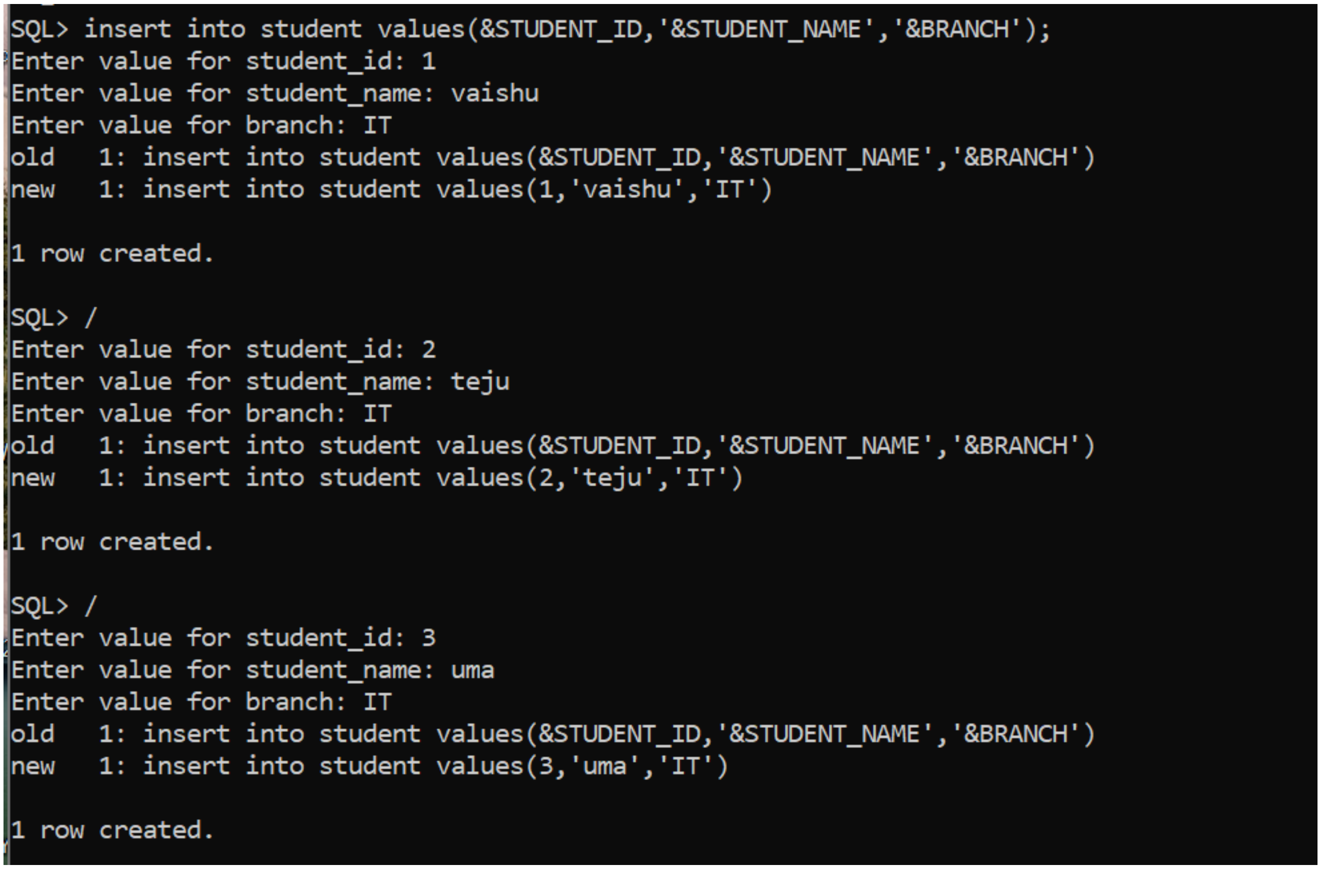


```
SQL> desc accesses
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------
 STUDENT_ID                                         NUMBER
 ID                                                 NUMBER
```

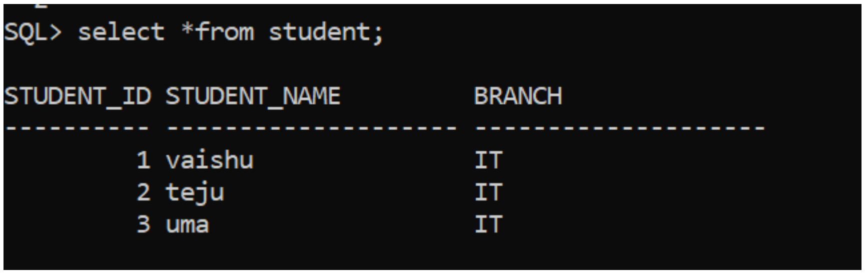
## DML OPERATIONS:

1) Inserting values into student table:

insert into student values(&STUDENT_ID, ‘&STUDENT_NAME’,’& BRANCH’);



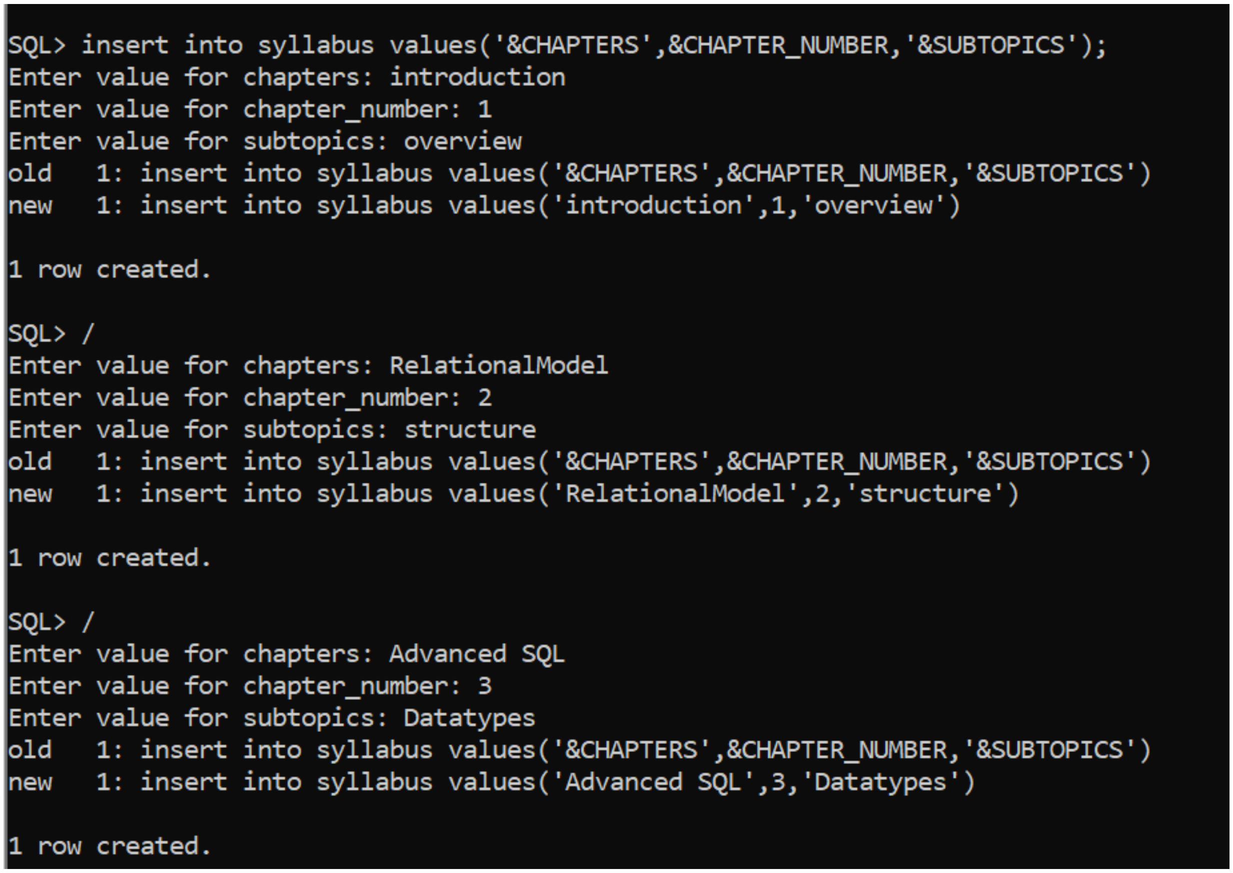
```
SQL> insert into student values(&STUDENT_ID,'&STUDENT_NAME','&BRANCH');
Enter value for student_id: 1
Enter value for student_name: vaishu
Enter value for branch: IT
old   1: insert into student values(&STUDENT_ID,'&STUDENT_NAME','&BRANCH')
new   1: insert into student values(1,'vaishu','IT')

1 row created.

SQL> /
Enter value for student_id: 2
Enter value for student_name: teju
Enter value for branch: IT
old   1: insert into student values(&STUDENT_ID,'&STUDENT_NAME','&BRANCH')
new   1: insert into student values(2,'teju','IT')

1 row created.

SQL> /
Enter value for student_id: 3
Enter value for student_name: uma
Enter value for branch: IT
old   1: insert into student values(&STUDENT_ID,'&STUDENT_NAME','&BRANCH')
new   1: insert into student values(3,'uma','IT')

1 row created.
```



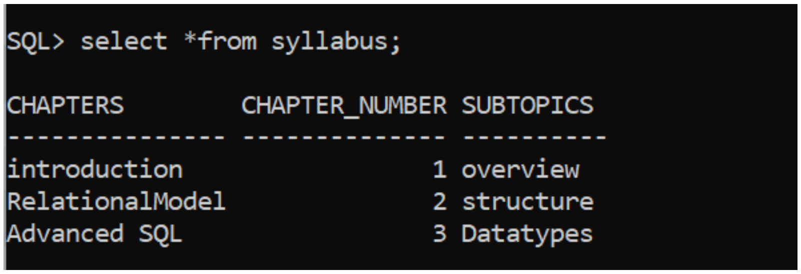
```
SQL> select *from student;

STUDENT_ID STUDENT_NAME         BRANCH
---------- -------------------- --------------------
         1 vaishu               IT
         2 teju                 IT
         3 uma                  IT
```

2) Inserting values into syllabus table:

insert into syllabus values('&CHAPTERS', & CHAPTER_NUMBER,'& SUBTOPICS');



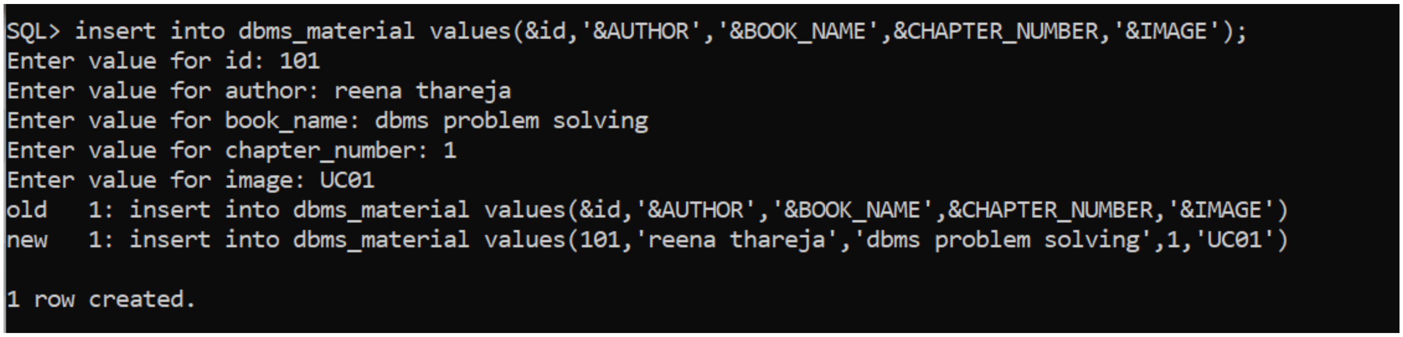
```
SQL> insert into syllabus values('&CHAPTERS',&CHAPTER_NUMBER,'&SUBTOPICS');
Enter value for chapters: introduction
Enter value for chapter_number: 1
Enter value for subtopics: overview
old   1: insert into syllabus values('&CHAPTERS',&CHAPTER_NUMBER,'&SUBTOPICS')
new   1: insert into syllabus values('introduction',1,'overview')

1 row created.

SQL> /
Enter value for chapters: RelationalModel
Enter value for chapter_number: 2
Enter value for subtopics: structure
old   1: insert into syllabus values('&CHAPTERS',&CHAPTER_NUMBER,'&SUBTOPICS')
new   1: insert into syllabus values('RelationalModel',2,'structure')

1 row created.

SQL> /
Enter value for chapters: Advanced SQL
Enter value for chapter_number: 3
Enter value for subtopics: Datatypes
old   1: insert into syllabus values('&CHAPTERS',&CHAPTER_NUMBER,'&SUBTOPICS')
new   1: insert into syllabus values('Advanced SQL',3,'Datatypes')

1 row created.
```



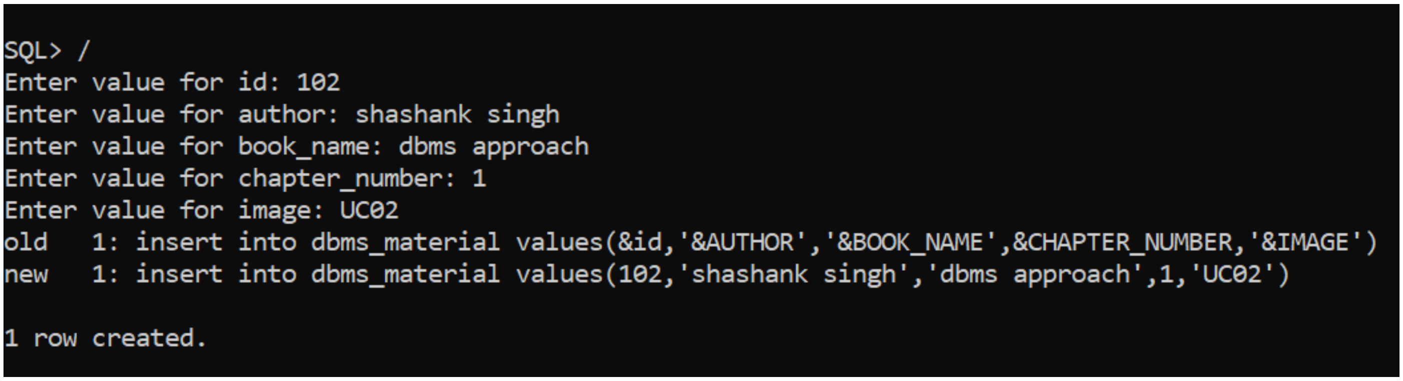
```
SQL> select *from syllabus;

CHAPTERS        CHAPTER_NUMBER SUBTOPICS
--------------- -------------- ----------
introduction                 1 overview
RelationalModel              2 structure
Advanced SQL                 3 Datatypes
```

3) Inserting values into dbms_material table:

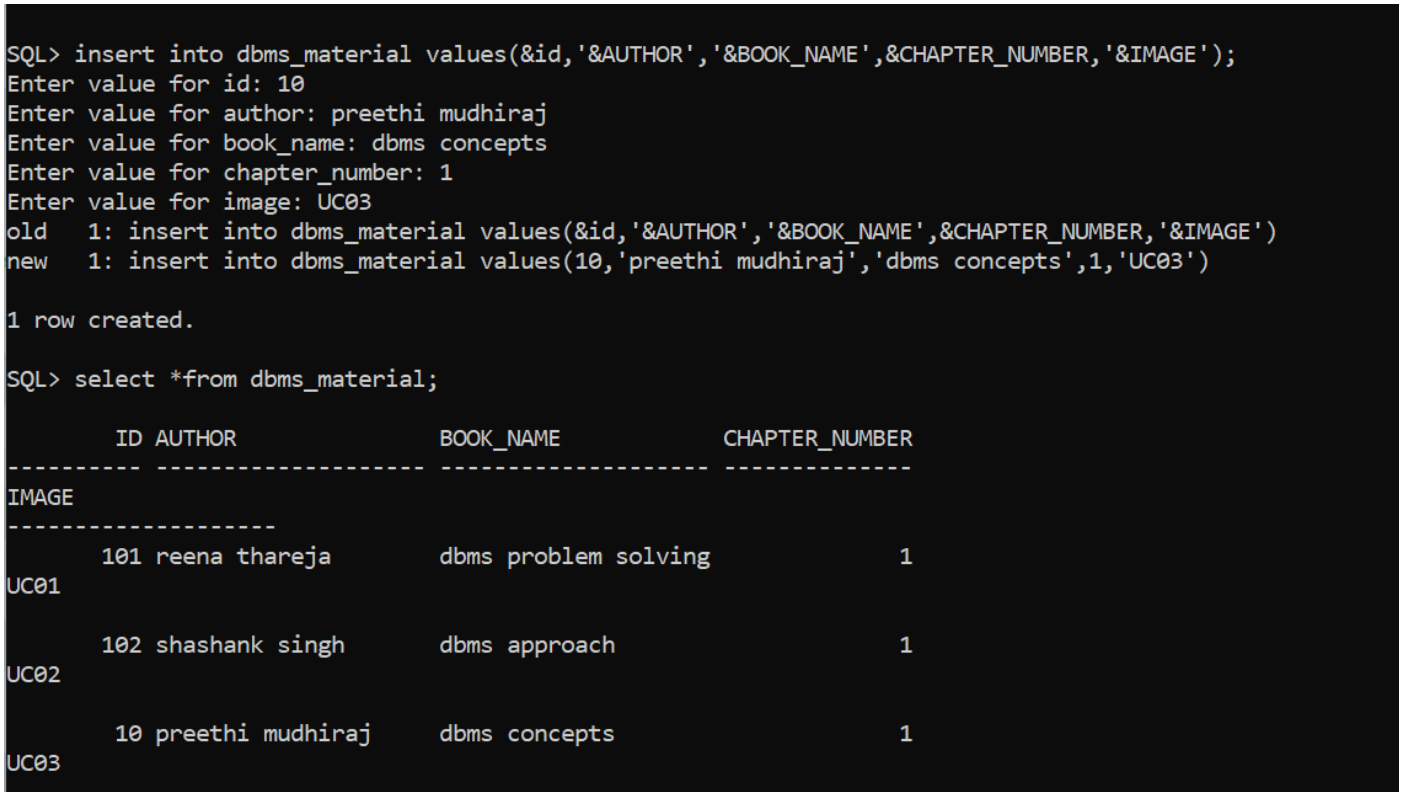
insert into dbms_material values(&id,'&AUTHOR','&BOOK_NAME',&image,&CHAPTER_NUMBER);



```
SQL> insert into dbms_material values(&id,'&AUTHOR','&BOOK_NAME',&CHAPTER_NUMBER,'&IMAGE');
Enter value for id: 101
Enter value for author: reena thareja
Enter value for book_name: dbms problem solving
Enter value for chapter_number: 1
Enter value for image: UC01
old   1: insert into dbms_material values(&id,'&AUTHOR','&BOOK_NAME',&CHAPTER_NUMBER,'&IMAGE')
new   1: insert into dbms_material values(101,'reena thareja','dbms problem solving',1,'UC01')

1 row created.
```



```
SQL> /
Enter value for id: 102
Enter value for author: shashank singh
Enter value for book_name: dbms approach
Enter value for chapter_number: 1
Enter value for image: UC02
old   1: insert into dbms_material values(&id,'&AUTHOR','&BOOK_NAME',&CHAPTER_NUMBER,'&IMAGE')
new   1: insert into dbms_material values(102,'shashank singh','dbms approach',1,'UC02')

1 row created.
```



```
SQL> insert into dbms_material values(&id,'&AUTHOR','&BOOK_NAME',&CHAPTER_NUMBER,'&IMAGE');
Enter value for id: 10
Enter value for author: preethi mudhiraj
Enter value for book_name: dbms concepts
Enter value for chapter_number: 1
Enter value for image: UC03
old   1: insert into dbms_material values(&id,'&AUTHOR','&BOOK_NAME',&CHAPTER_NUMBER,'&IMAGE')
new   1: insert into dbms_material values(10,'preethi mudhiraj','dbms concepts',1,'UC03')

1 row created.

SQL> select *from dbms_material;

        ID AUTHOR               BOOK_NAME            CHAPTER_NUMBER
---------- -------------------- -------------------- --------------
IMAGE
--------------------
       101 reena thareja        dbms problem solving              1
UC01

       102 shashank singh       dbms approach                     1
UC02

        10 preethi mudhiraj     dbms concepts                     1
UC03
```

# IMPLEMENTATION

## SyllabusUI

```
import javax.swing.*;

class SyllabusUI extends JFrame

{

    JTextField t1,t2,t3;

    JLabel l1,l2,l3;

    JPanel p;


    public SyllabusUI()

    {

        //setSize(450,450);

        //setLayout(null);

        //setVisible(true);

        createComponents();

        addComponents();

        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }


    void createComponents()

    {

        t1 = new JTextField();

        t1.setBounds(200,30,150,30);


        t2 = new JTextField();

        t2.setBounds(200,100,150,30);


        t3 = new JTextField();
```

```
		t3.setBounds(200,180,150,30);


		l1 = new JLabel("Chapters :  ");
		l1.setBounds(50,30,150,30);

		l2 = new JLabel("Chapter Number :  ");
		l2.setBounds(50,100,150,30);

		l3 = new JLabel("Sub Topic :  ");
		l3.setBounds(50,180,150,30);

		p = new JPanel(null);
		p.setBounds(0,0,400,250);
	}

	void addComponents()
	{
		p.add(l1);
		p.add(t1);
		p.add(l2);
		p.add(t2);
		p.add(l3);
		p.add(t3);
		add(p);
	}

	/*public static void main(String a[])
	{
```

```
                new SyllabusUI();
        }*/
}
```

## StudentUI

```
import javax.swing.*;
class StudentUI extends JFrame
{
        JTextField t1,t2,t3;
        JLabel l1,l2,l3;
        JPanel p;

        public StudentUI()
        {
                //setSize(450,450);
                //setLayout(null);
                //setVisible(true);
                createComponents();
                addComponents();
                //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }

        void createComponents()
        {
                t1 = new JTextField();
                t1.setBounds(200,30,150,30);

                t2 = new JTextField();
                t2.setBounds(200,100,150,30);
```

```
		t3 = new JTextField();
		t3.setBounds(200,180,150,30);


		l1 = new JLabel("Student ID :  ");
		l1.setBounds(50,30,150,30);

		l2 = new JLabel("Student Name :  ");
		l2.setBounds(50,100,150,30);

		l3 = new JLabel("Branch :  ");
		l3.setBounds(50,180,150,30);

		p = new JPanel(null);
		p.setBounds(0,0,400,250);
	}

	void addComponents()
	{
		p.add(l1);
		p.add(t1);
		p.add(l2);
		p.add(t2);
		p.add(l3);
		p.add(t3);
		add(p);
	}
```

```
    /*public static void main(String a[])
    {
        new StudentUI();
    }*/
}
```

## MaterialUI

```
import javax.swing.*;
class MaterialUI extends JFrame
{
    JTextField t1,t2,t3,t4;
    JLabel l1,l2,l3,l4;
    JPanel p;

    public MaterialUI()
    {
    //  setSize(450,450);
    //  setLayout(null);
        //setVisible(true);
        createComponents();
        addComponents();
        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    void createComponents()
    {
        t1 = new JTextField();
        t1.setBounds(200,30,150,30);
    t2 = new JTextField();
    t2.setBounds(200,80,150,30);

    t3 = new JTextField();
    t3.setBounds(200,140,150,30);

    t4 = new JTextField();
    t4.setBounds(200,200,150,30);

    l1 = new JLabel("Book ID :  ");
    l1.setBounds(50,30,150,30);

    l2 = new JLabel("Author ID :  ");
    l2.setBounds(50,80,150,30);

    l3 = new JLabel("Book Name :  ");
    l3.setBounds(50,140,150,30);

    l4 = new JLabel("Image :  ");
    l4.setBounds(50,200,150,30);

    p = new JPanel(null);
    p.setBounds(0,0,400,250);
}

void addComponents()
{
    p.add(l1);
    p.add(t1);
```

```
        p.add(l2);

        p.add(t2);

        p.add(l3);

        p.add(t3);

        p.add(l4);

        p.add(t4);

        add(p);

    }


    /*public static void main(String a[])

    {

        new MaterialUI();

    }*/

}
```

## AccessesUI

```
import javax.swing.*;

class AccessesUI extends JFrame

{

    JTextField t1,t2;

    JLabel l1,l2;

    JPanel p;


    public AccessesUI()

    {

        //setSize(450,450);

        //setLayout(null);

        //setVisible(true);

        createComponents();
        addComponents();
        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

void createComponents()
{
        t1 = new JTextField();
        t1.setBounds(200,30,150,30);

        t2 = new JTextField();
        t2.setBounds(200,100,150,30);



        l1 = new JLabel("Student ID :  ");
        l1.setBounds(50,30,150,30);

        l2 = new JLabel("Book ID :  ");
        l2.setBounds(50,100,150,30);



        p = new JPanel(null);
        p.setBounds(0,0,400,250);
}

void addComponents()
{
        p.add(l1);
        p.add(t1);
        p.add(l2);
```

```
                p.add(t2);

                add(p);
        }

        /*public static void main(String a[])
        {
                new AccessesUI();
        }*/
}
```

## MainUI

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class MainUI extends JFrame implements ActionListener
{
        StudentUI ob1;
        MaterialUI ob2;
        SyllabusUI ob3;
        AccessesUI ob4;

        JButton submit,modify,delete,m1,m2,m3,m4;
        JPanel p1,p2,p3,p4,pb;
        JMenuBar mb;

        public MainUI()
        {
```

```
        setSize(450,450);
        setLayout(null);
        setVisible(true);
        setTitle("Personal Counselling Management System");

        ob1 = new StudentUI();
        ob2 = new MaterialUI();
        ob3 = new SyllabusUI();
        ob4 = new AccessesUI();

        createPanels();
        createMenu();
        createButtons();
        addComponents();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

void createPanels()
{
        p1 = ob1.p;
        p2 = ob2.p;
        p3 = ob3.p;
        p4      = ob4.p;

        pb = new JPanel(new FlowLayout(FlowLayout.CENTER,50,0));
        pb.setBounds(0,300,400,200);
}

void createMenu()
```

```
{
        mb = new JMenuBar();

        m1 = new JButton("Student");
        m1.setFocusable(false);

        m2 = new JButton("DBMS Material");
        m2.setFocusable(false);



        m3 = new JButton("Syllabus");
        m3.setFocusable(false);

        m4 = new JButton("Accesses");
        m4.setFocusable(false);

        m1.addActionListener(this);
        m2.addActionListener(this);
        m3.addActionListener(this);
        m4.addActionListener(this);

        mb.add(m1);
        mb.add(m2);
        mb.add(m3);
        mb.add(m4);

}

public void actionPerformed(ActionEvent e)
```

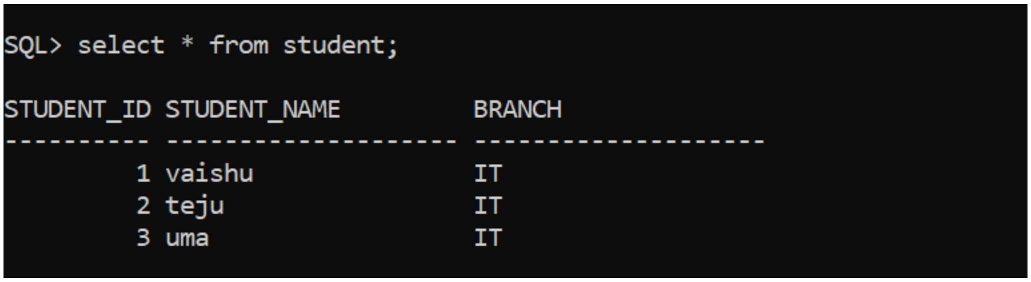
```
    {
        remove(p1);
        remove(p2);
        remove(p3);
        remove(p4);

        if(e.getSource()==m1)
            add(p1);

        else if(e.getSource()==m2)
            add(p2);

        else if(e.getSource()==m3)
            add(p3);

        else
            add(p4);
    }

    void createButtons()
    {
        submit = new JButton("Submit");
        submit.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                JOptionPane.showMessageDialog(new JFrame(),"Successfully Inserted!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
            }
        });



        modify = new JButton("Modify");

        modify.addActionListener(new ActionListener()

        {

            public void actionPerformed(ActionEvent e)

            {

                JOptionPane.showMessageDialog(new JFrame(),"Successfully
Modified!","NOTICE",JOptionPane.INFORMATION_MESSAGE);

            }

        });



        delete = new JButton("Delete");

        delete.addActionListener(new ActionListener()

        {

            public void actionPerformed(ActionEvent e)

            {

                JOptionPane.showMessageDialog(new JFrame(),"Successfully
Deleted!","NOTICE",JOptionPane.INFORMATION_MESSAGE);

            }

        });


        pb.add(submit);

        pb.add(modify);

        pb.add(delete);

    }


    void addComponents()
```

```
        {
                add(p1);
                add(pb);
                setJMenuBar(mb);
        }
        public static void main(String a[])
        {
                new MainUI();
        }
}
```

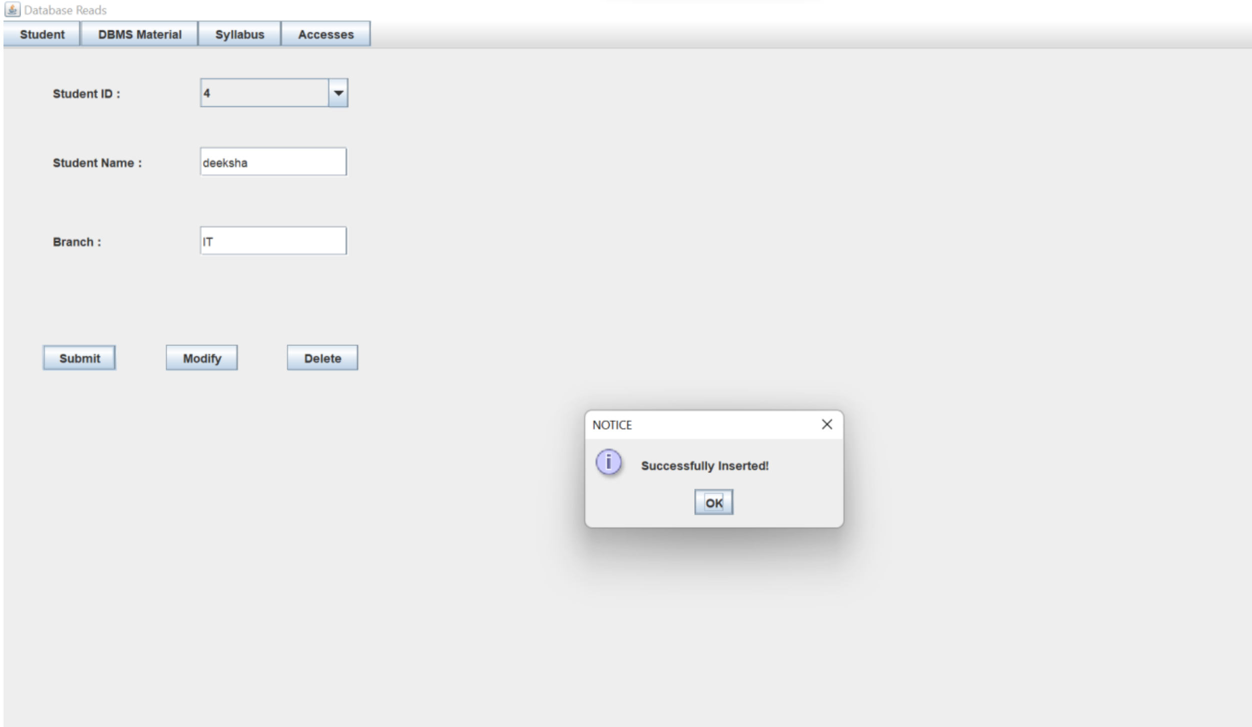
## DBConnection

```
import java.sql.*;
public class TrialConnect{
        public static void main(String[] args){
                try{
                        Class.forName("oracle.jdbc.OracleDriver");
                        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","leena","vasavi");
                        Statement stmt=con.createStatement();
                        ResultSet rs=stmt.executeQuery("select * from Customer");
                        while(rs.next())
                                System.out.println(rs.getInt(1)+" "+rs.getString(2));
                        con.close();
                }
                catch(Exception e){
                        System.out.println(e);
                }
        }
}
```

# OUTPUT:



```
SQL> select * from student;

STUDENT_ID STUDENT_NAME         BRANCH
---------- -------------------- --------------------
         1 vaishu               IT
         2 teju                 IT
         3 uma                  IT
```



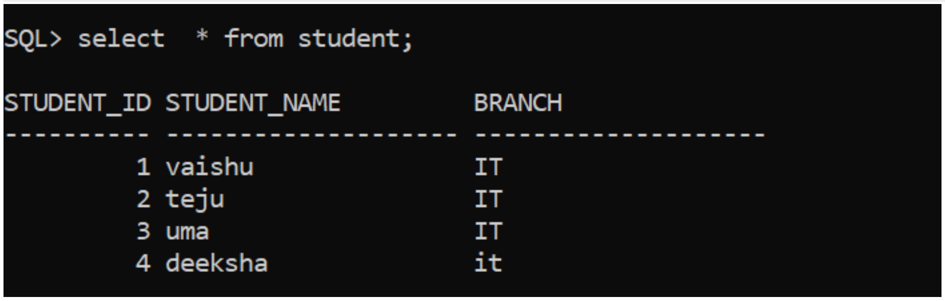
Student ID : 4

Student Name : deeksha

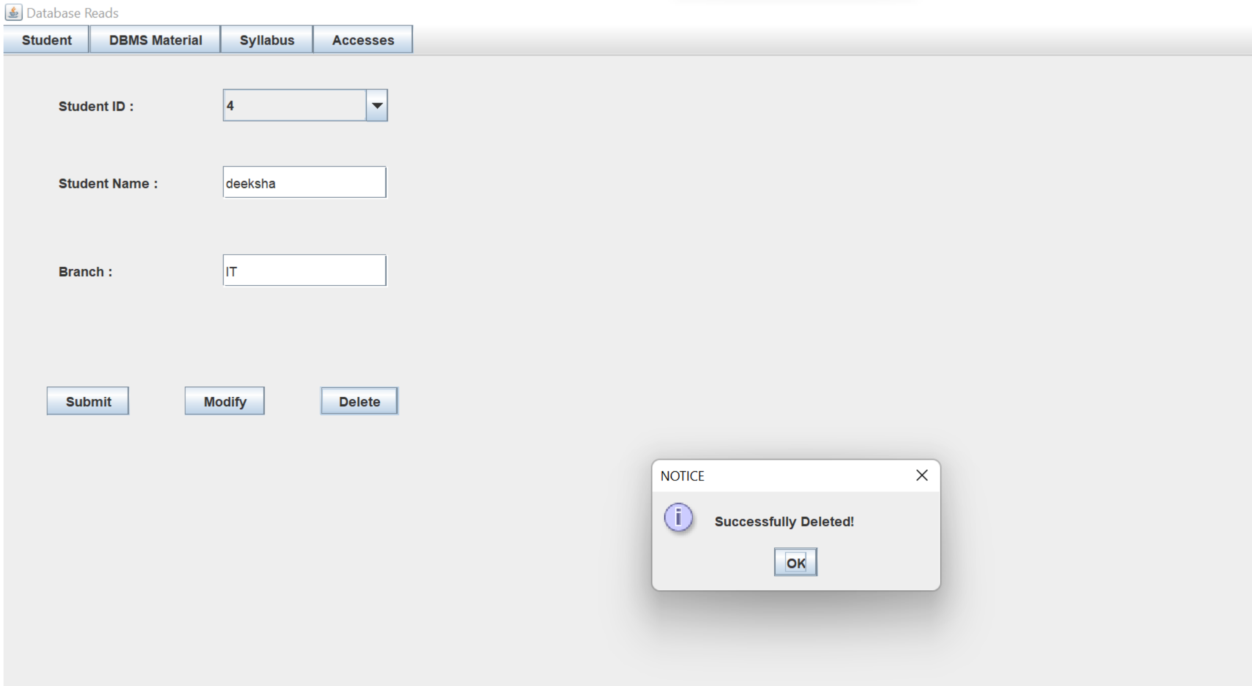
Branch : IT

NOTICE

Successfully Inserted!



```
SQL> select  * from student;

STUDENT_ID STUDENT_NAME         BRANCH
---------- -------------------- --------------------
         1 vaishu               IT
         2 teju                 IT
         3 uma                  IT
         4 deeksha              it
```

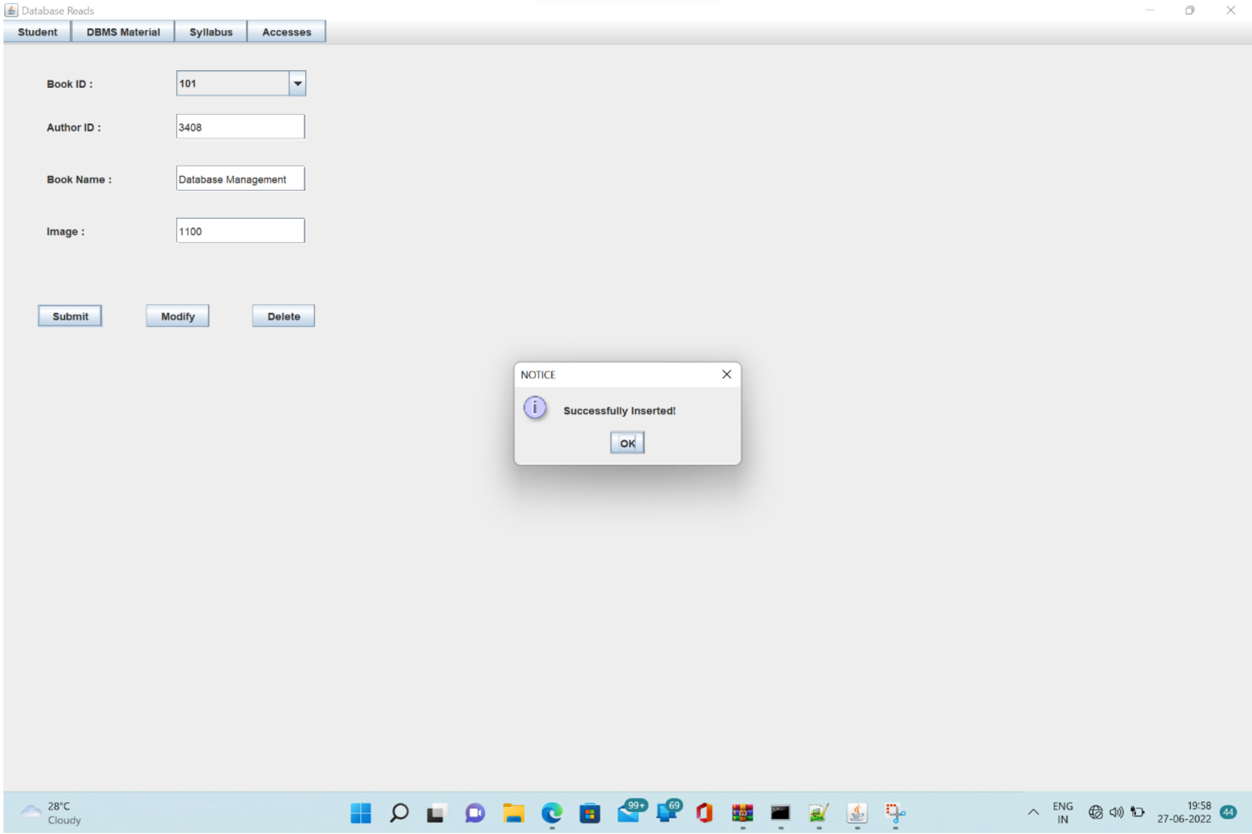


Student ID : 4

Student Name : deeksha

Branch : IT

Successfully Deleted!



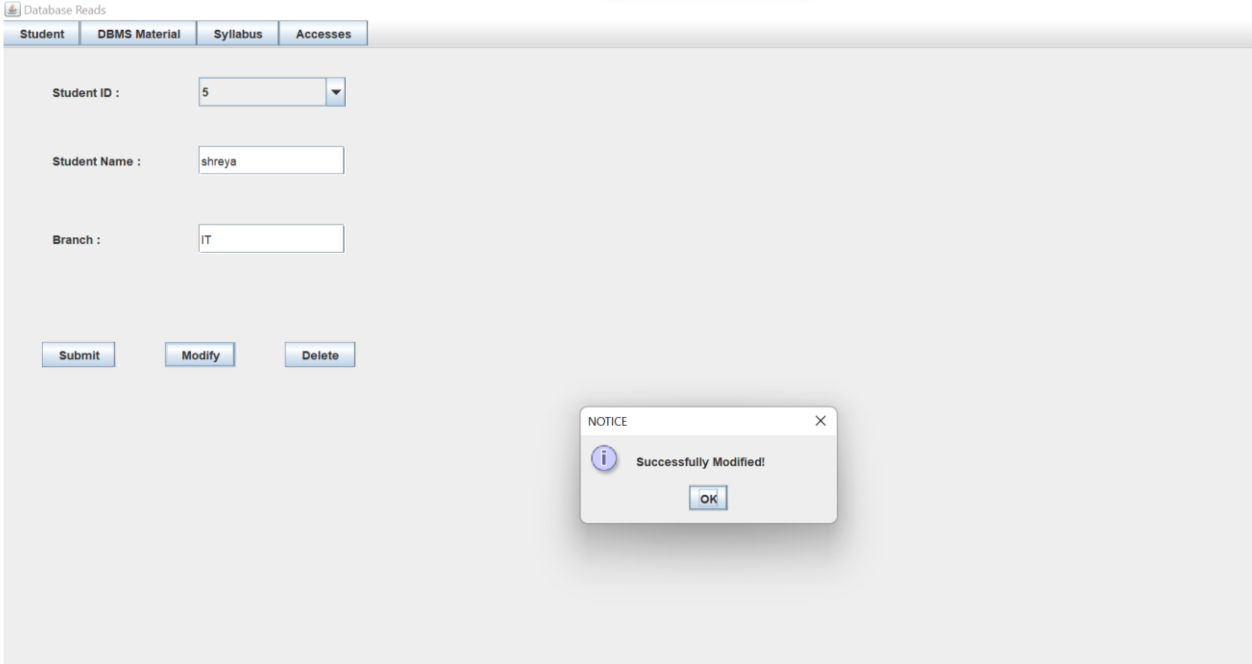
Book ID : 101

Author ID : 3408

Book Name : Database Management

Image : 1100

Successfully Inserted!



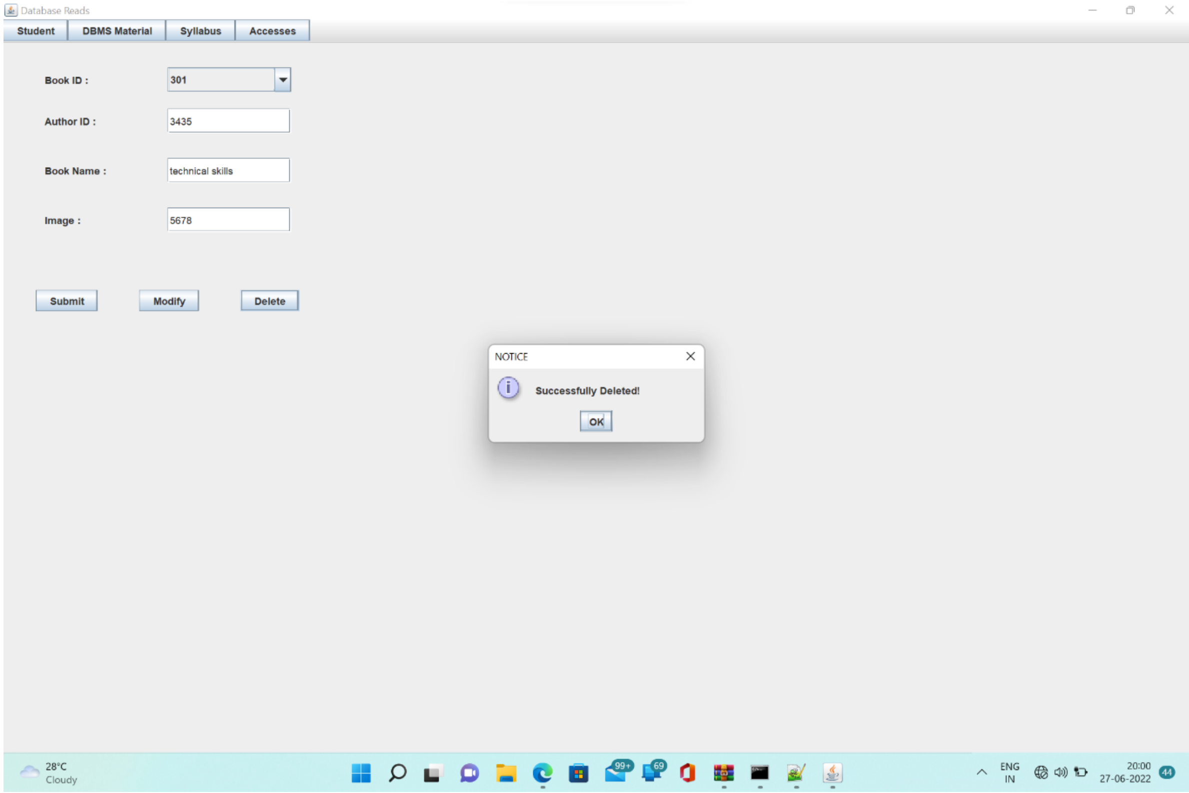
Student ID : 5

Student Name : shreya

Branch : IT

NOTICE

Successfully Modified!



Book ID : 301

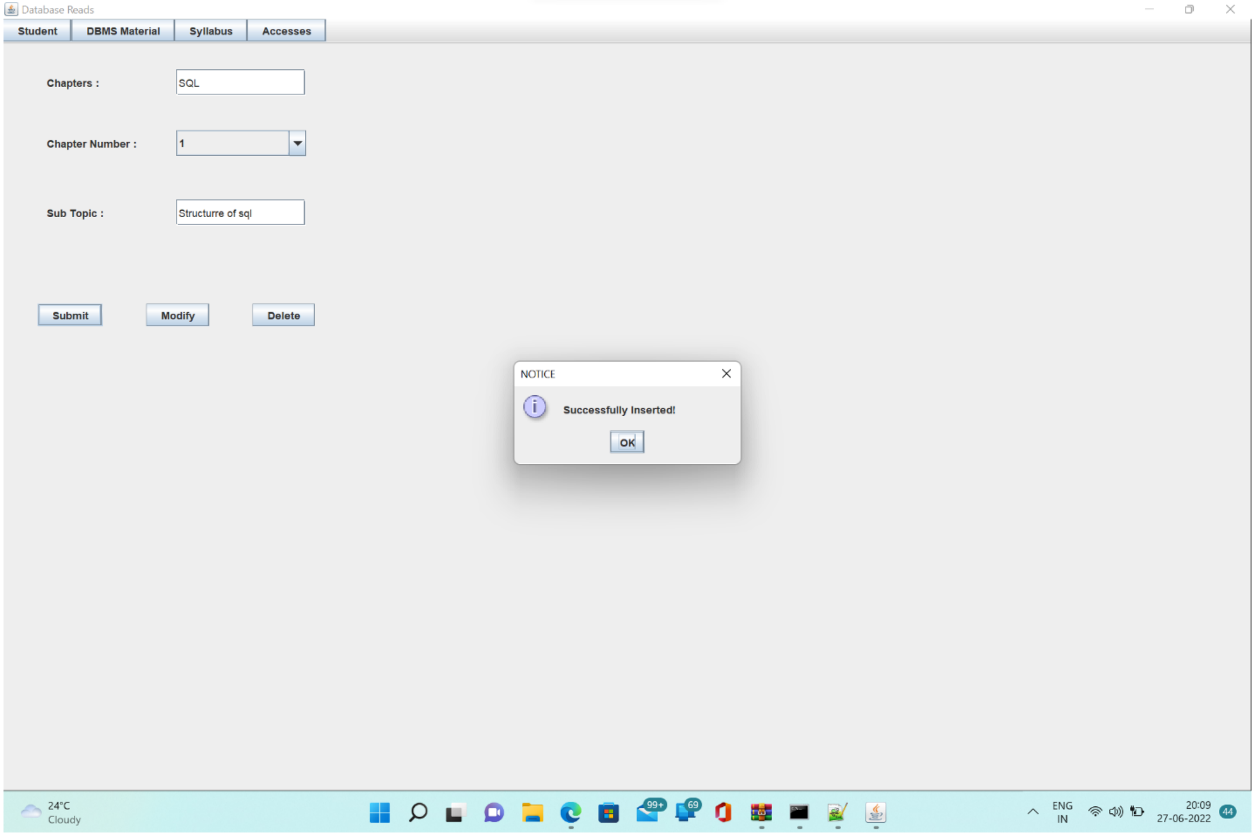
Author ID : 3435

Book Name : technical skills

Image : 5678

NOTICE

Successfully Deleted!

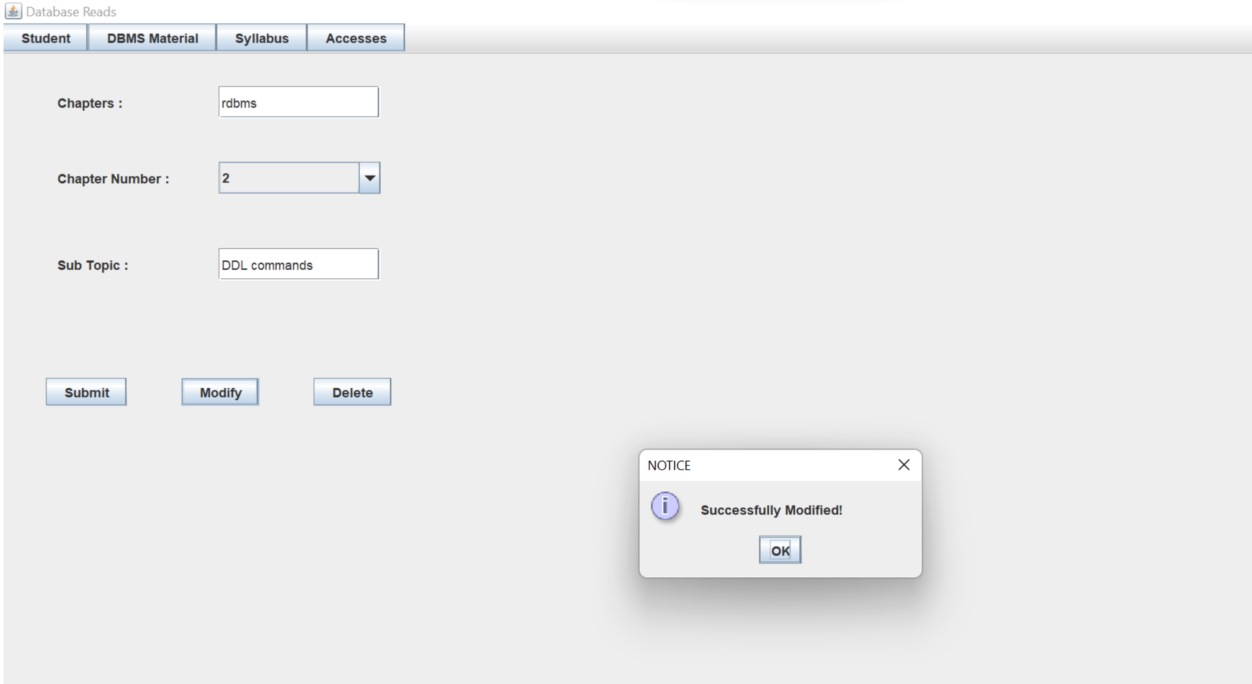


Chapters : SQL

Chapter Number : 1

Sub Topic : Structurre of sql

Successfully Inserted!



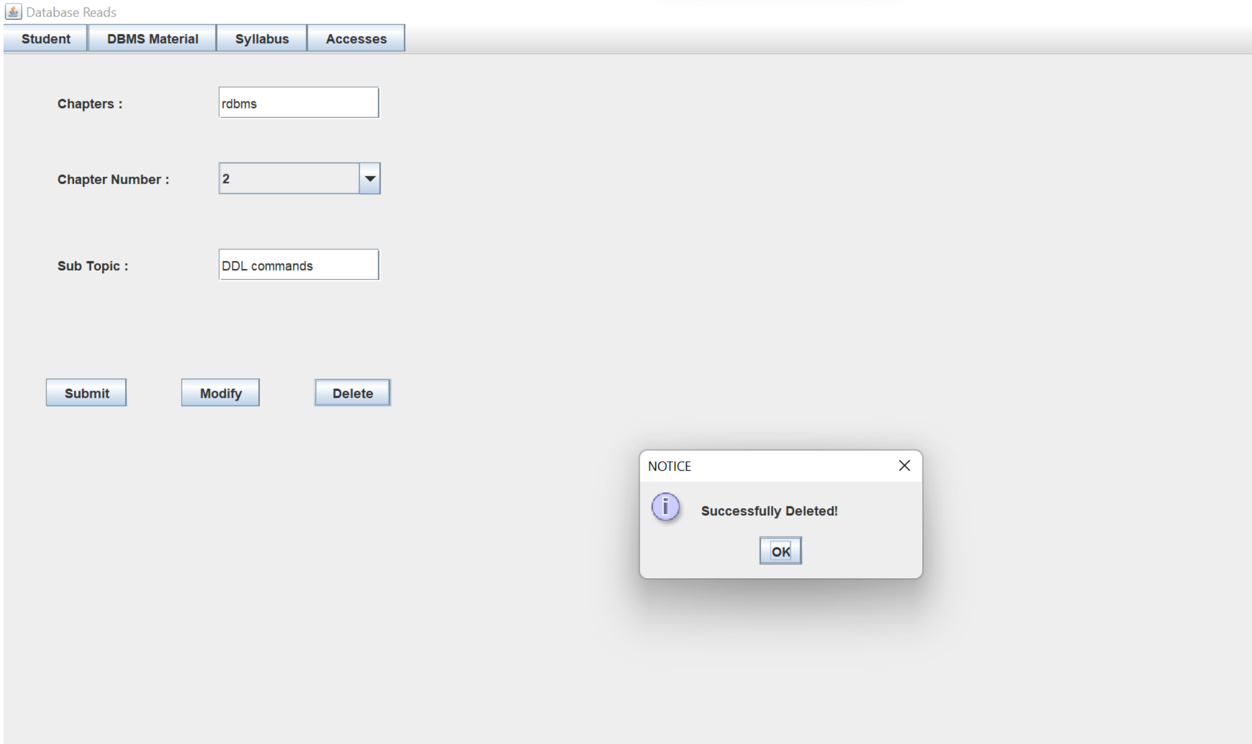
Chapters : rdbms

Chapter Number : 2

Sub Topic : DDL commands

NOTICE

Successfully Modified!



Chapters : rdbms

Chapter Number : 2

Sub Topic : DDL commands

NOTICE

Successfully Deleted!

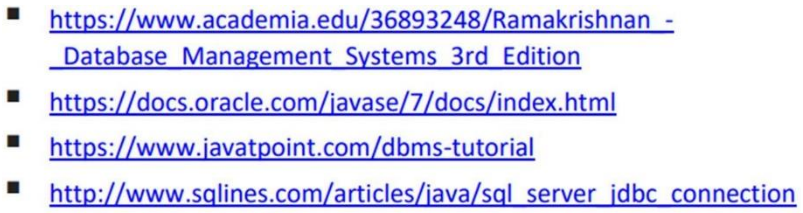
**RESULT:**

**I successfully completed the MINI PROJECT "DATABASE READS".**

FUTUREWORK:

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create a android app by which I can control my project on my hand and connect to it.

REFERENCES:



- https://www.academia.edu/36893248/Ramakrishnan_-_Database_Management_Systems_3rd_Edition
- https://docs.oracle.com/javase/7/docs/index.html
- https://www.javatpoint.com/dbms-tutorial
- http://www.sqlines.com/articles/java/sql_server_jdbc_connection