PLICKER CARD BATABASE FOR DBMS QUIZ

A
Report
Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY
By

B.SUCHITH <1602-21-737-058>
Under the guidance of Ms B. Leelavathy

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

2022-2023

ABSTRACT

The abstract for the Plicker card database for DBMS quiz would outline the basic features and functionality of the database. Plicker cards are used for conducting polls and quizzes in the classroom, and the database would be designed to store and manage the data generated by the use of these cards ,teachers,course,questions responses.The database would include tables to store information such as user accounts, quiz questions and answers, and the results of quizzes,teachers,courses. It would be designed to support the creation and management of multiple quizzes, with the ability to customize question types, answer choices, and scoring options.Overall, the Plicker card database for DBMS quiz would provide a robust and user-friendly solution for managing the data generated by the use of Plicker cards in the classroom.

Requirement Analysis

List of Tables:

1. Students
2. Teachers
3.enrollments
4.classes
5.courses
6.questions
7.responses

List of attibures:

1.students

- studentid
- sname
- emailid

2.teachers

- teacherid
- tname
- temailis

3.enrollments

- enrollmentid
- studentid
- classid

4.classes

- teacherid
- classid

5.courses

- coursed
- name

6.questions

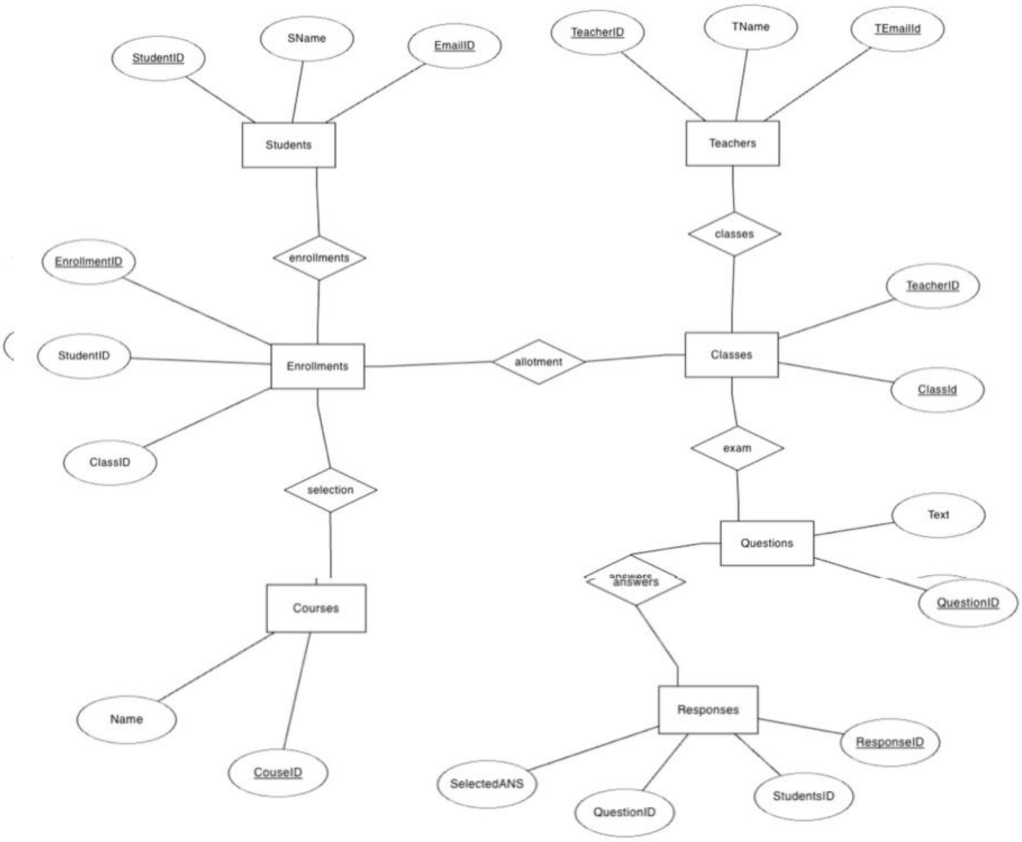
- questioned
- text

7.responses

- responseid
- studentid
- questioned
- selectedans

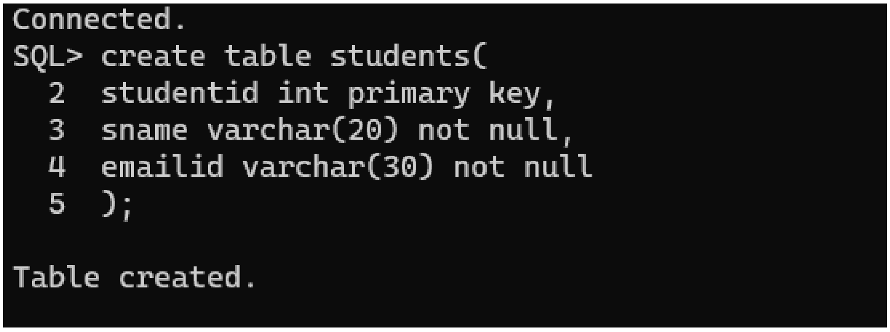
DESIGN

Entity relationship Diagram



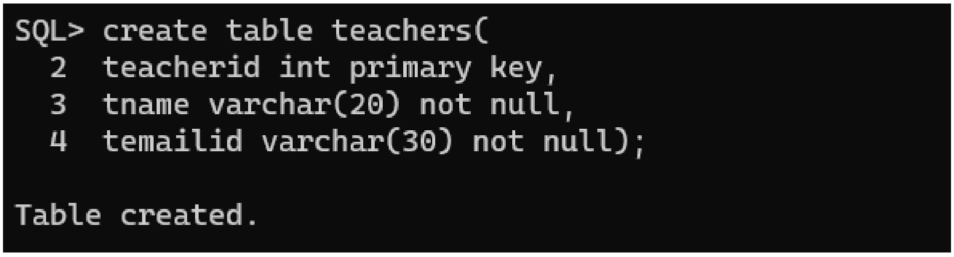
DDL Commands:

```
SQL> create table students(
  2  studentid int primary key,
  3  sname varchar(20) not null,
  4  emailid varchar(30) not null
  5  );
```



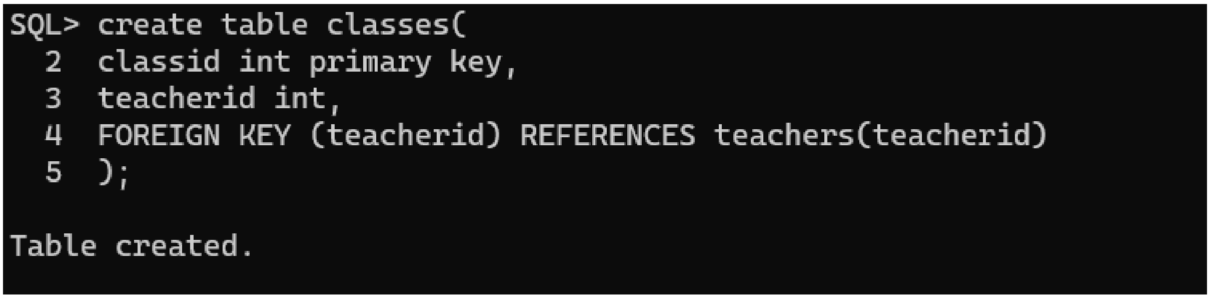
```
Connected.
SQL> create table students(
  2  studentid int primary key,
  3  sname varchar(20) not null,
  4  emailid varchar(30) not null
  5  );

Table created.
```

```
SQL> create table teachers(
  2  teacherid int primary key,
  3  tname varchar(20) not null,
  4  temailid varchar(30) not null);
```



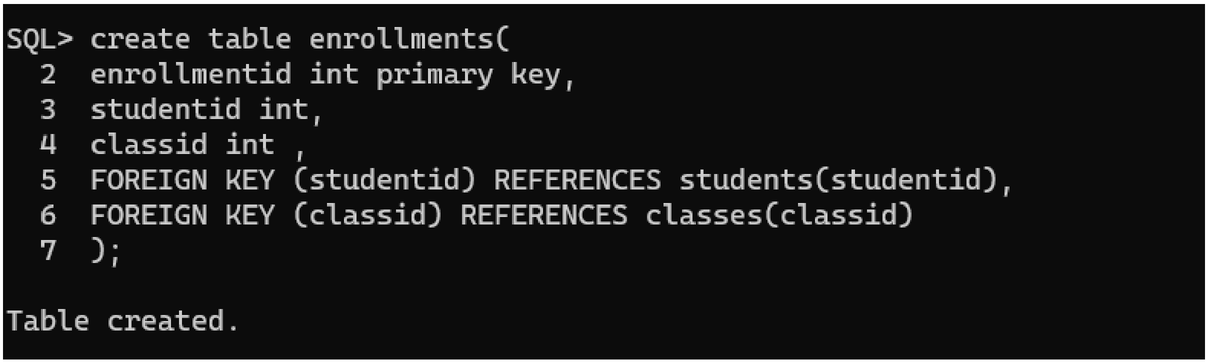
```
SQL> create table teachers(
  2  teacherid int primary key,
  3  tname varchar(20) not null,
  4  temailid varchar(30) not null);

Table created.
```

```
SQL> create table classes(
  2  classid int primary key,
  3  teacherid int,
  4  FOREIGN KEY (teacherid) REFERENCES teachers(teacherid)
5  );
```



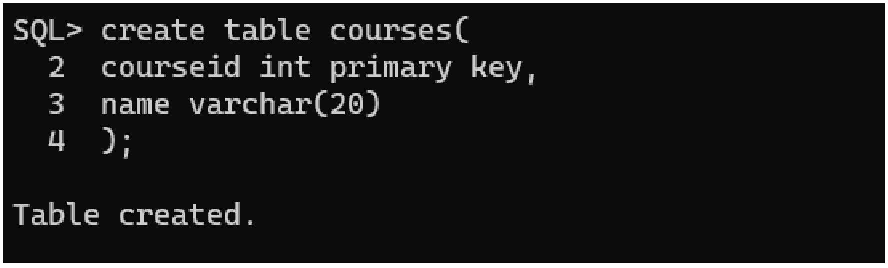
```
SQL> create table classes(
  2  classid int primary key,
  3  teacherid int,
  4  FOREIGN KEY (teacherid) REFERENCES teachers(teacherid)
  5  );

Table created.
```

```
SQL> create table enrollments(
  2  enrollmentid int primary key,
  3  studentid int,
  4  classid int ,
  5  FOREIGN KEY (studentid) REFERENCES students(studentid),
  6  FOREIGN KEY (classid) REFERENCES classes(classid)
  7  );
```



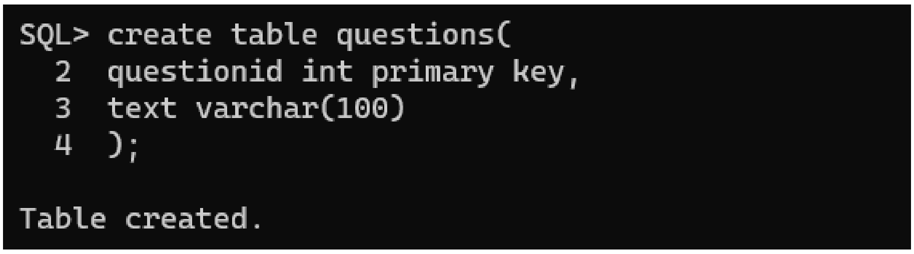
```
SQL> create table enrollments(
  2  enrollmentid int primary key,
  3  studentid int,
  4  classid int ,
  5  FOREIGN KEY (studentid) REFERENCES students(studentid),
  6  FOREIGN KEY (classid) REFERENCES classes(classid)
  7  );

Table created.
```

```
SQL> create table courses(
  2  courseid int primary key,
  3  name varchar(20)
  4  );
```



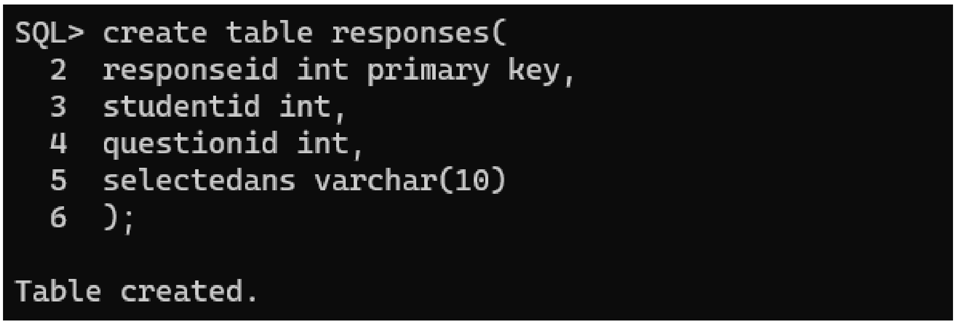
```
SQL> create table courses(
  2  courseid int primary key,
  3  name varchar(20)
  4  );

Table created.
```

```
SQL> create table questions(
  2  questionid int primary key,
  3  text varchar(100)
  4  );
```



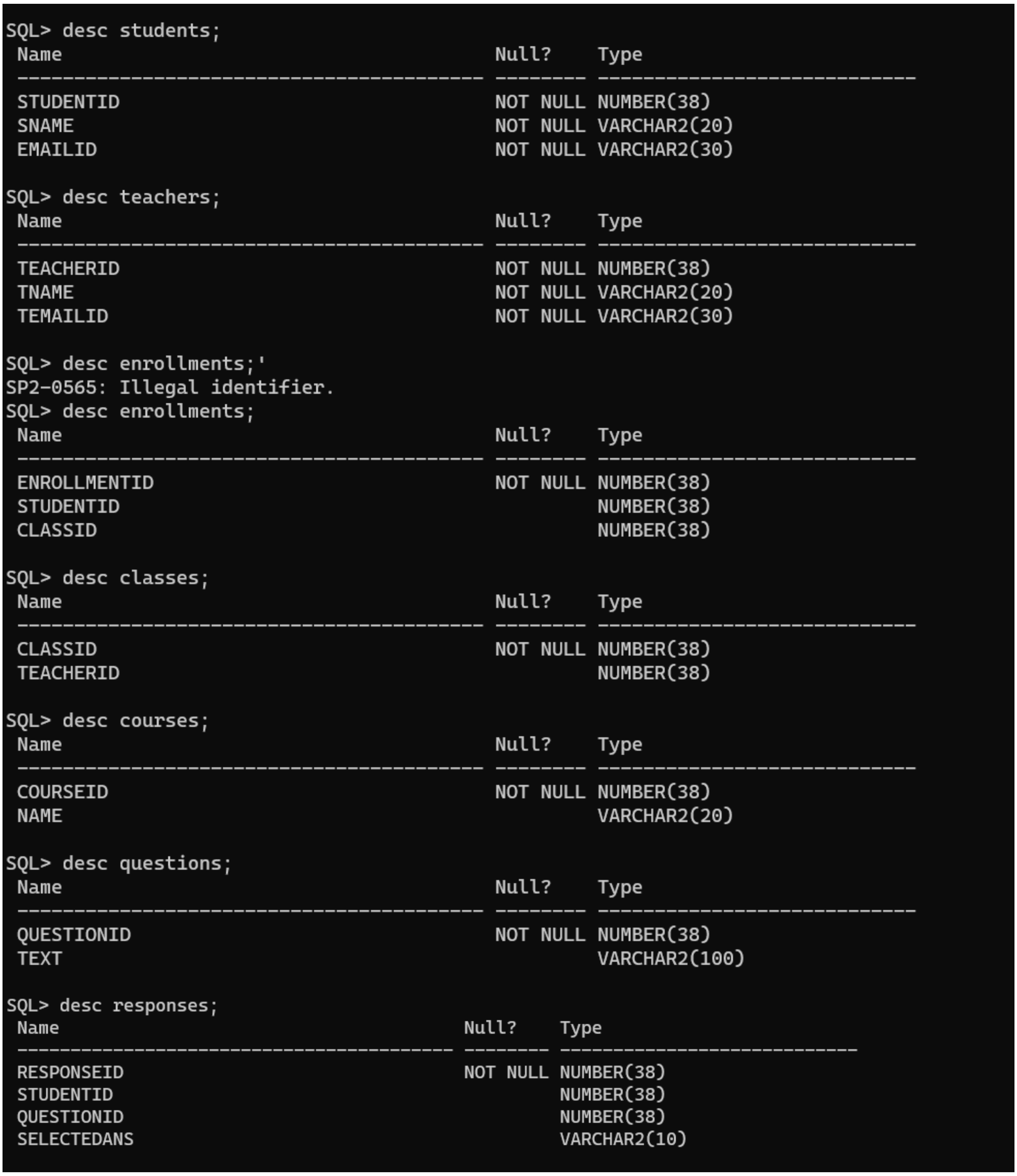
```
SQL> create table questions(
  2  questionid int primary key,
  3  text varchar(100)
  4  );

Table created.
```

```
SQL> create table responses(
  2  responseid int primary key,
  3  studentid int,
  4  questionid int,
  5  selectedans varchar(10)
  6  );
```



```
SQL> create table responses(
  2  responseid int primary key,
  3  studentid int,
  4  questionid int,
  5  selectedans varchar(10)
  6  );

Table created.
```

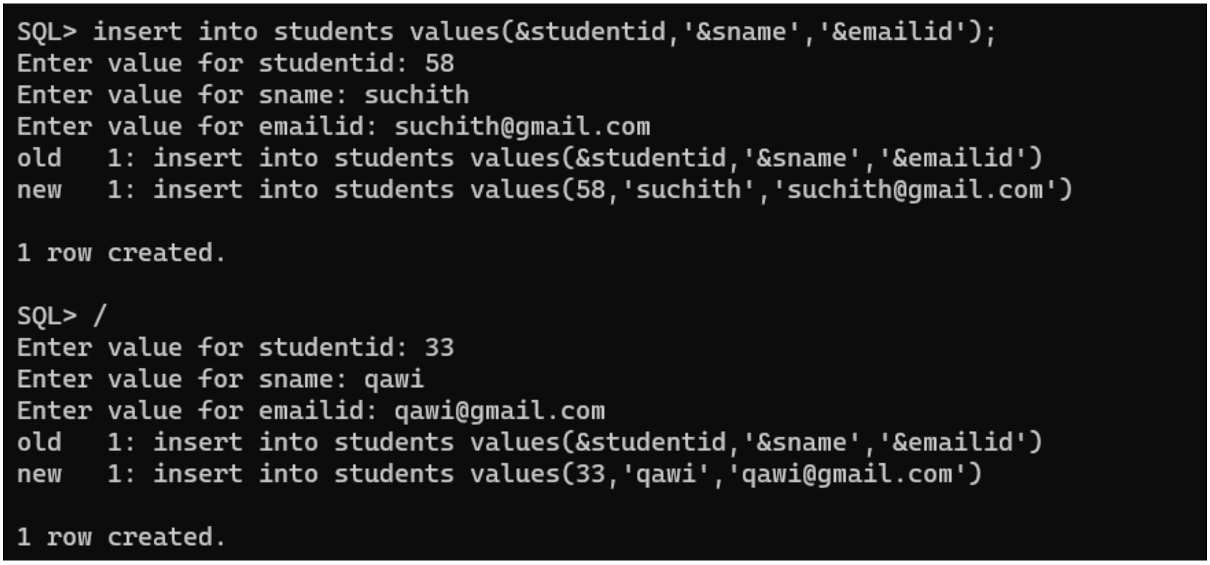
TABLES:



```
SQL> desc students;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENTID                                 NOT NULL NUMBER(38)
 SNAME                                     NOT NULL VARCHAR2(20)
 EMAILID                                   NOT NULL VARCHAR2(30)

SQL> desc teachers;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TEACHERID                                 NOT NULL NUMBER(38)
 TNAME                                     NOT NULL VARCHAR2(20)
 TEMAILID                                  NOT NULL VARCHAR2(30)

SQL> desc enrollments;'
SP2-0565: Illegal identifier.
SQL> desc enrollments;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ENROLLMENTID                              NOT NULL NUMBER(38)
 STUDENTID                                          NUMBER(38)
 CLASSID                                            NUMBER(38)

SQL> desc classes;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASSID                                   NOT NULL NUMBER(38)
 TEACHERID                                          NUMBER(38)

SQL> desc courses;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSEID                                  NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(20)

SQL> desc questions;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 QUESTIONID                                NOT NULL NUMBER(38)
 TEXT                                               VARCHAR2(100)

SQL> desc responses;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RESPONSEID                                NOT NULL NUMBER(38)
 STUDENTID                                          NUMBER(38)
 QUESTIONID                                         NUMBER(38)
 SELECTEDANS                                        VARCHAR2(10)
```

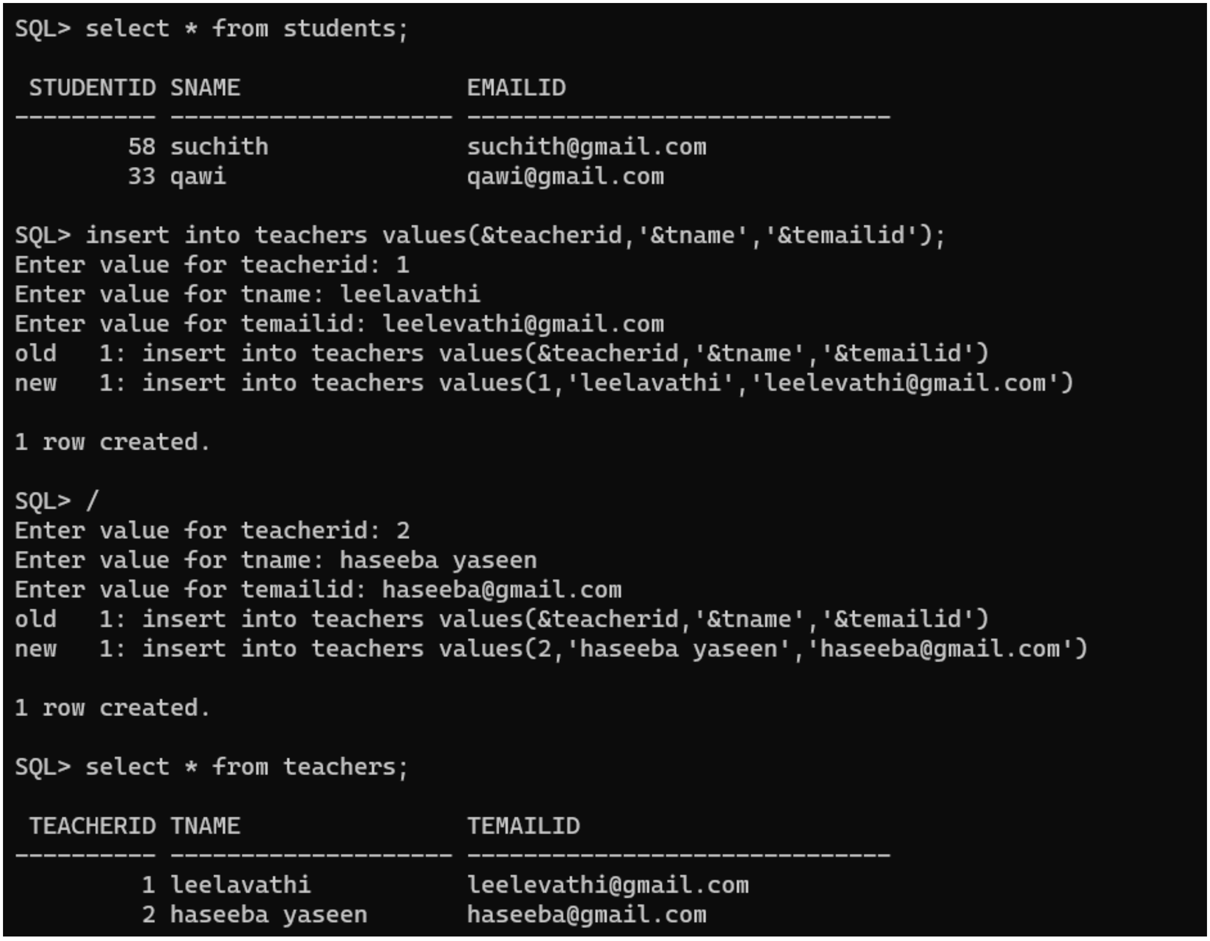
DML Commands:

SQL> insert into students values(&studentid,'&sname','&emailid');



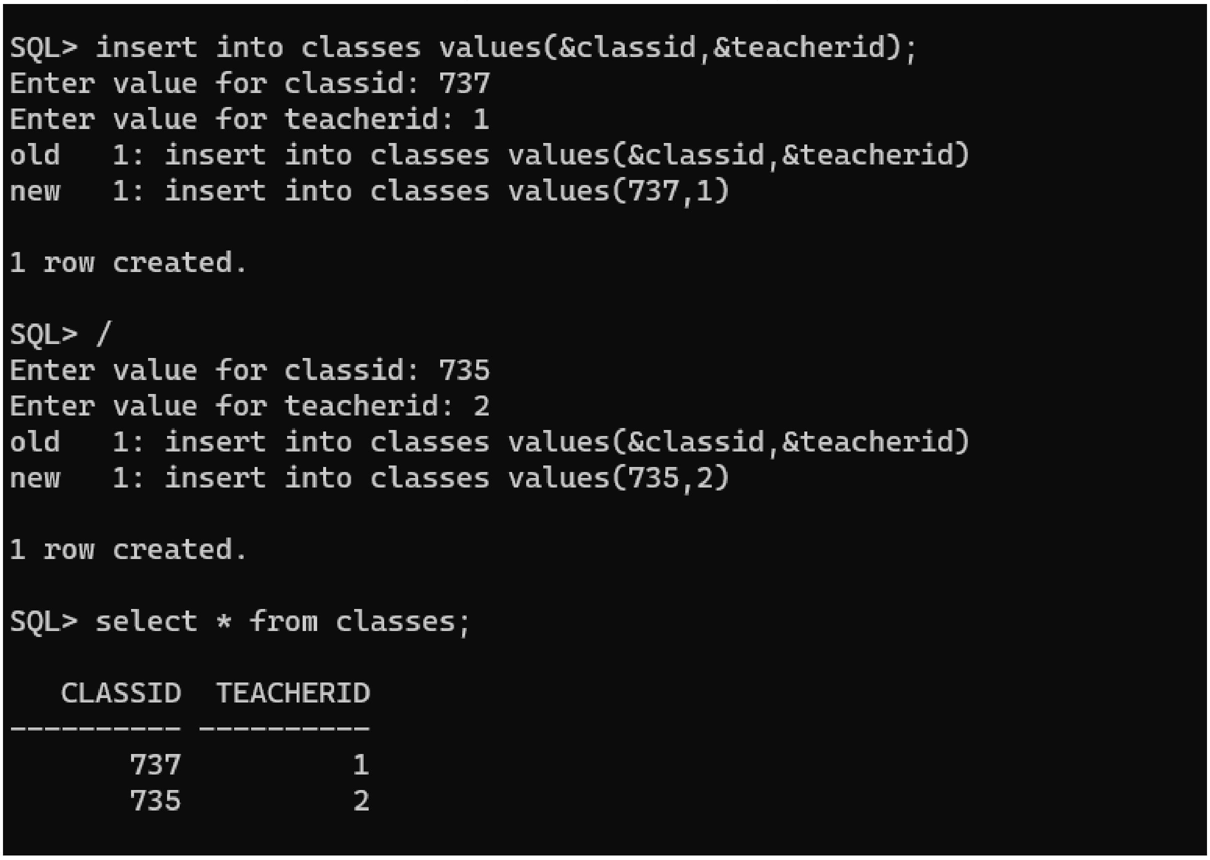
```
SQL> insert into students values(&studentid,'&sname','&emailid');
Enter value for studentid: 58
Enter value for sname: suchith
Enter value for emailid: suchith@gmail.com
old   1: insert into students values(&studentid,'&sname','&emailid')
new   1: insert into students values(58,'suchith','suchith@gmail.com')

1 row created.

SQL> /
Enter value for studentid: 33
Enter value for sname: qawi
Enter value for emailid: qawi@gmail.com
old   1: insert into students values(&studentid,'&sname','&emailid')
new   1: insert into students values(33,'qawi','qawi@gmail.com')

1 row created.
```

SQL> insert into teachers values(&teacherid,'&tname','&temailid');



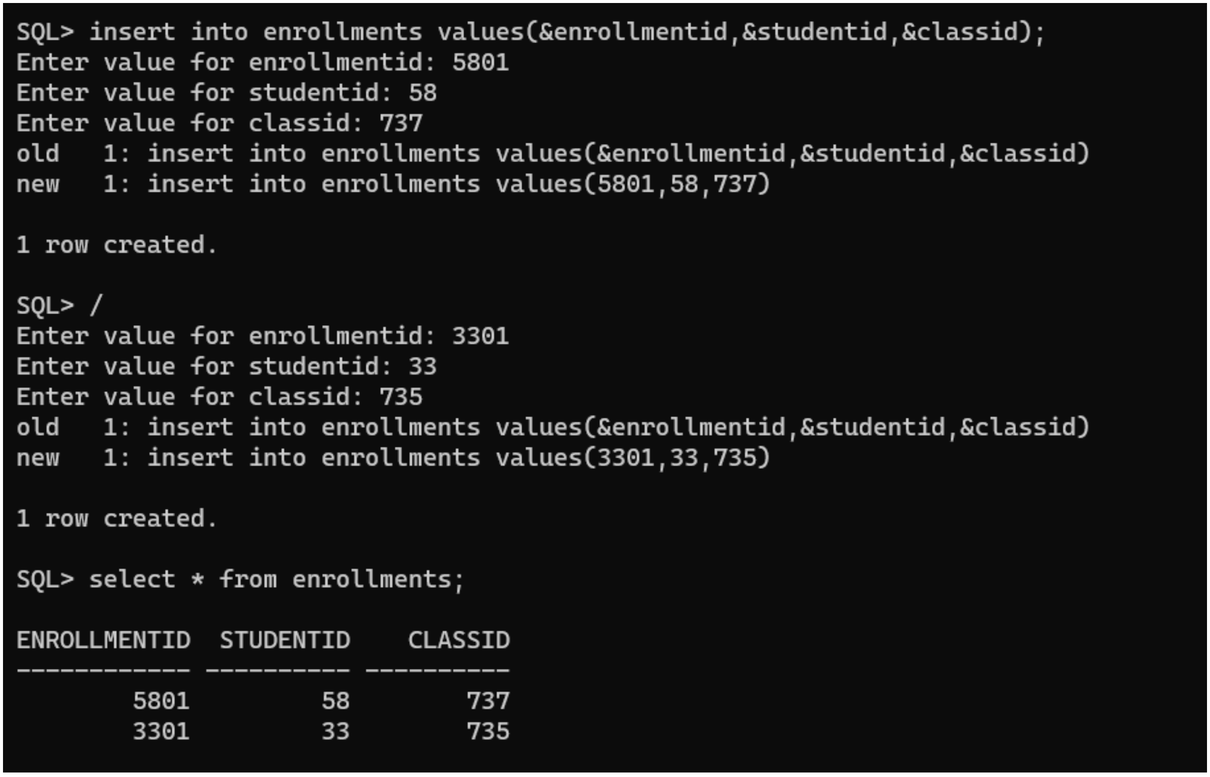
```
SQL> select * from students;

 STUDENTID SNAME                EMAILID
---------- -------------------- ------------------------------
        58 suchith              suchith@gmail.com
        33 qawi                 qawi@gmail.com

SQL> insert into teachers values(&teacherid,'&tname','&temailid');
Enter value for teacherid: 1
Enter value for tname: leelavathi
Enter value for temailid: leelevathi@gmail.com
old   1: insert into teachers values(&teacherid,'&tname','&temailid')
new   1: insert into teachers values(1,'leelavathi','leelevathi@gmail.com')

1 row created.

SQL> /
Enter value for teacherid: 2
Enter value for tname: haseeba yaseen
Enter value for temailid: haseeba@gmail.com
old   1: insert into teachers values(&teacherid,'&tname','&temailid')
new   1: insert into teachers values(2,'haseeba yaseen','haseeba@gmail.com')

1 row created.

SQL> select * from teachers;

 TEACHERID TNAME                TEMAILID
---------- -------------------- ------------------------------
         1 leelavathi           leelevathi@gmail.com
         2 haseeba yaseen       haseeba@gmail.com
```

SQL> insert into classes values(&classid,&teacherid);



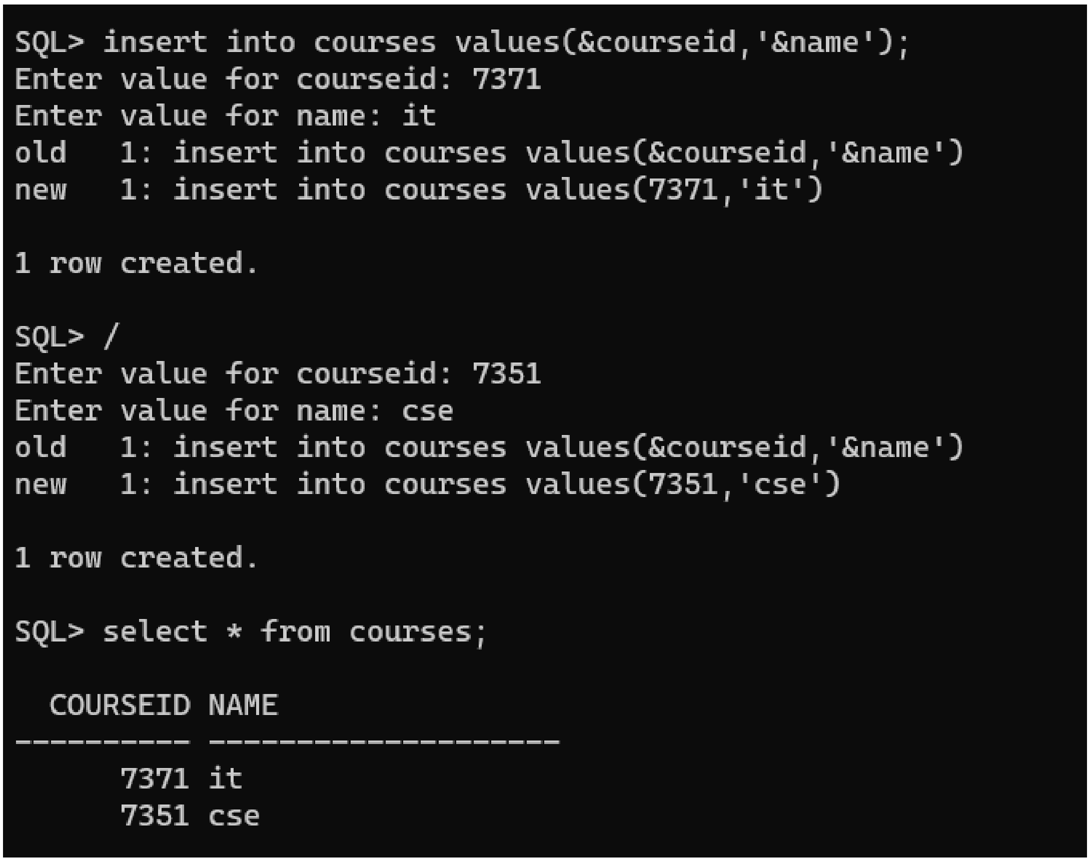
```
SQL> insert into classes values(&classid,&teacherid);
Enter value for classid: 737
Enter value for teacherid: 1
old   1: insert into classes values(&classid,&teacherid)
new   1: insert into classes values(737,1)

1 row created.

SQL> /
Enter value for classid: 735
Enter value for teacherid: 2
old   1: insert into classes values(&classid,&teacherid)
new   1: insert into classes values(735,2)

1 row created.

SQL> select * from classes;

   CLASSID  TEACHERID
---------- ----------
       737          1
       735          2
```

SQL> insert into enrollments values(&enrollmentid,&studentid,&classid);



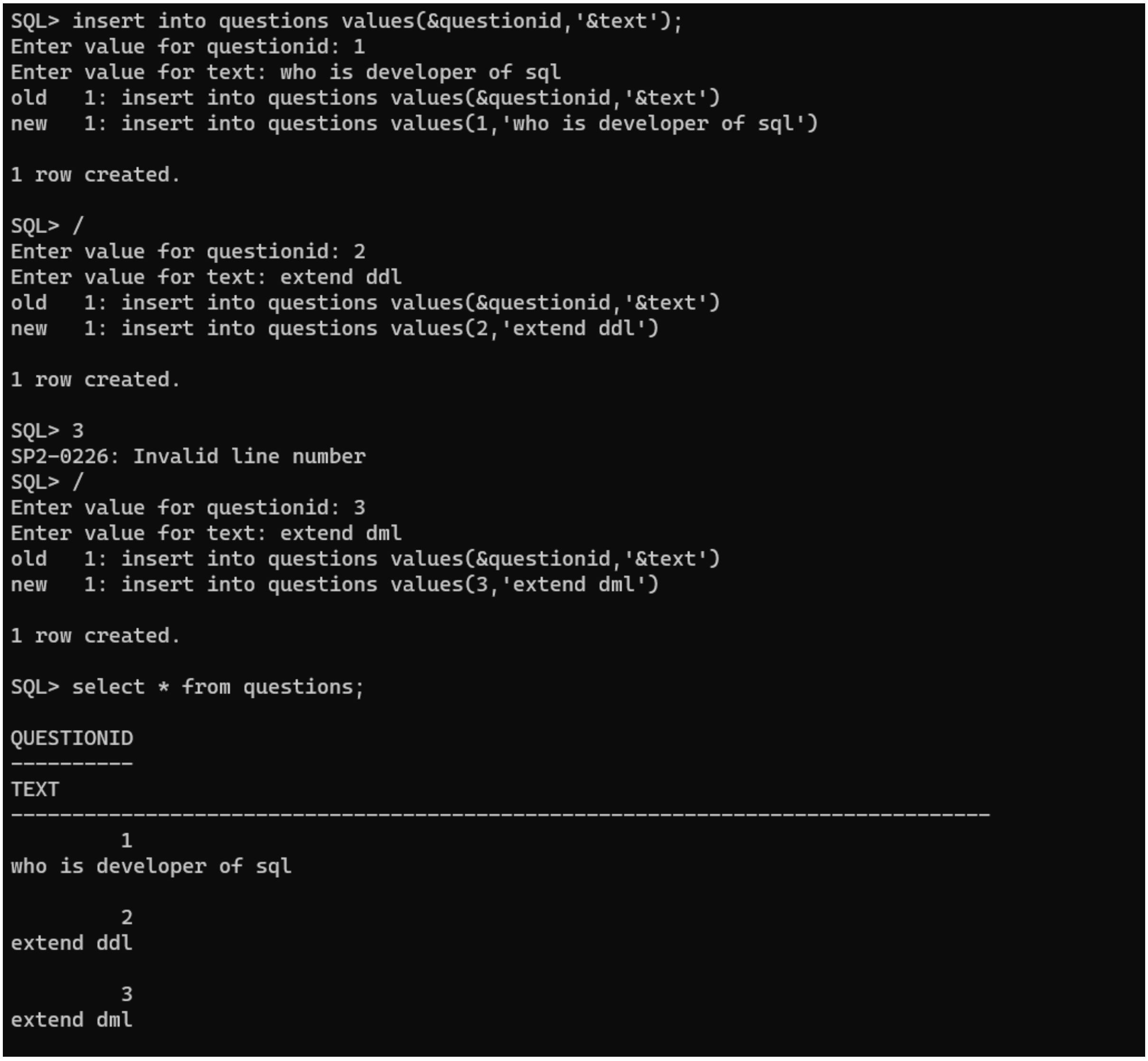
```
SQL> insert into enrollments values(&enrollmentid,&studentid,&classid);
Enter value for enrollmentid: 5801
Enter value for studentid: 58
Enter value for classid: 737
old   1: insert into enrollments values(&enrollmentid,&studentid,&classid)
new   1: insert into enrollments values(5801,58,737)

1 row created.

SQL> /
Enter value for enrollmentid: 3301
Enter value for studentid: 33
Enter value for classid: 735
old   1: insert into enrollments values(&enrollmentid,&studentid,&classid)
new   1: insert into enrollments values(3301,33,735)

1 row created.

SQL> select * from enrollments;

ENROLLMENTID  STUDENTID    CLASSID
------------ ---------- ----------
        5801         58        737
        3301         33        735
```

SQL> insert into courses values(&courseid,'&name');



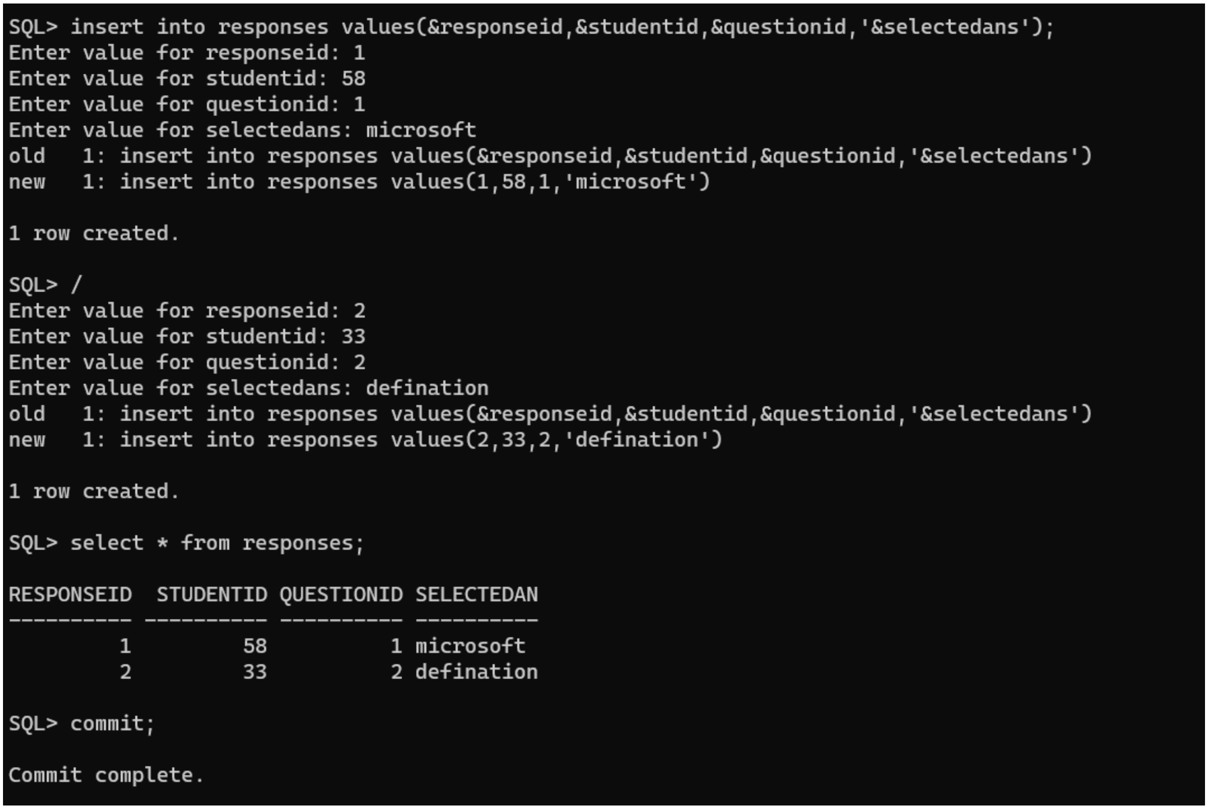
```
SQL> insert into courses values(&courseid,'&name');
Enter value for courseid: 7371
Enter value for name: it
old   1: insert into courses values(&courseid,'&name')
new   1: insert into courses values(7371,'it')

1 row created.

SQL> /
Enter value for courseid: 7351
Enter value for name: cse
old   1: insert into courses values(&courseid,'&name')
new   1: insert into courses values(7351,'cse')

1 row created.

SQL> select * from courses;

  COURSEID NAME
---------- --------------------
      7371 it
      7351 cse
```

SQL> insert into questions values(&questionid,'&text');



```
SQL> insert into questions values(&questionid,'&text');
Enter value for questionid: 1
Enter value for text: who is developer of sql
old   1: insert into questions values(&questionid,'&text')
new   1: insert into questions values(1,'who is developer of sql')

1 row created.

SQL> /
Enter value for questionid: 2
Enter value for text: extend ddl
old   1: insert into questions values(&questionid,'&text')
new   1: insert into questions values(2,'extend ddl')

1 row created.

SQL> 3
SP2-0226: Invalid line number
SQL> /
Enter value for questionid: 3
Enter value for text: extend dml
old   1: insert into questions values(&questionid,'&text')
new   1: insert into questions values(3,'extend dml')

1 row created.

SQL> select * from questions;

QUESTIONID
----------
TEXT
--------------------------------------------------------------------------------
         1
who is developer of sql

         2
extend ddl

         3
extend dml
```

SQL> insert into responses values(&responseid,&studentid,&questionid,'&selectedans');



```
SQL> insert into responses values(&responseid,&studentid,&questionid,'&selectedans');
Enter value for responseid: 1
Enter value for studentid: 58
Enter value for questionid: 1
Enter value for selectedans: microsoft
old   1: insert into responses values(&responseid,&studentid,&questionid,'&selectedans')
new   1: insert into responses values(1,58,1,'microsoft')

1 row created.

SQL> /
Enter value for responseid: 2
Enter value for studentid: 33
Enter value for questionid: 2
Enter value for selectedans: defination
old   1: insert into responses values(&responseid,&studentid,&questionid,'&selectedans')
new   1: insert into responses values(2,33,2,'defination')

1 row created.

SQL> select * from responses;

RESPONSEID  STUDENTID QUESTIONID SELECTEDAN
---------- ---------- ---------- ----------
         1         58          1 microsoft
         2         33          2 defination

SQL> commit;

Commit complete.
```

IMPLEMENTATION

JAVA-SQL Connectivity using JDBC:

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases. The connection to the database can be performed using Java programming (JDBC API) as:

```
{ DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver()); // Connect to Oracle
Database Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE" ,"shruthi","kovvurss");
Statementstatement = con.createStatement() String query = "UPDATE SKILLS SET SS1=" +"'"+
jTextField3.getText() +"',SS2=" +"'"+ jTextField5.getText() +"', AOI ="+" '"+
jTextField2.getText() +"' WHERE SID =+" + jTextField4.getText(); ResultSet rs =
statement.executeQuery(query); JOptionPane.showMessageDialog(new JFrame(),
"Upadated Successfully", "INFORMATION", JOptionPane.INFORMATION_MESSAGE);
rs.close(); statement.close(); con.close(); }
```

Front-end Programs (User Interfaces) Home Page:

1.Home Page:

```
import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class QuizApplication1 {
   private JFrame frame;
   private JLabel questionLabel;
   private JLabel questionid;
   private JRadioButton option1RadioButton;
   private JRadioButton option2RadioButton;
   private JRadioButton option3RadioButton;
   private JRadioButton option4RadioButton;

   private JButton submitButton;
   public int id;
   private int currentQuestionIndex;
   private int score;

   private Question[] questions = {
         new Question("Question 1: Which type of data can be stored in database? ",
               new String[]{"image oriented data", "text files", "data in form of audio", "all th
above ","QuestionID :vceoo1"}, 3),
         new Question("Question 2: Which of following is not a database?",
               new String[]{"heiarchial", "network", "decentralized", "distributed","QuestionID
:vceoo2"}, 2),
         new Question("Question 3: Which is not a dbms?",
               new String[]{"My sql", "microsoft acess", "google", "ibm db2","QuestionID
:vceoo3"},2 ),
         new Question("Question 4: Who painted the Mona Lisa?",
               new String[]{"Leonardo da Vinci", "Pablo Picasso", "Vincent van Gogh",
"Michelangelo","QuestionID :vceoo4"}, 0),
         new Question("Question 5: What is the chemical symbol for gold?",
               new String[]{"Au", "Ag", "Fe", "Cu","QuestionID :vceoo5"}, 0)
   };
```

```
public QuizApplication1(int x) {
        this.id=x;
    currentQuestionIndex = 0;
    score = 0;
    initialize();
}


private void initialize() {
    frame = new JFrame("Quiz Application");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 300);

    JPanel panel = new JPanel();
    panel.setLayout(new BorderLayout());

    questionLabel = new JLabel();
    panel.add(questionLabel, BorderLayout.NORTH);

    JPanel optionsPanel = new JPanel();
    optionsPanel.setLayout(new GridLayout(5, 1));

    option1RadioButton = new JRadioButton();
    optionsPanel.add(option1RadioButton);

    option2RadioButton = new JRadioButton();
    optionsPanel.add(option2RadioButton);

    option3RadioButton = new JRadioButton();
    optionsPanel.add(option3RadioButton);

    option4RadioButton = new JRadioButton();
    optionsPanel.add(option4RadioButton);

    questionid=new JLabel();
    optionsPanel.add(questionid);


    ButtonGroup buttonGroup = new ButtonGroup();
    buttonGroup.add(option1RadioButton);
    buttonGroup.add(option2RadioButton);
    buttonGroup.add(option3RadioButton);
    buttonGroup.add(option4RadioButton);
```

```
        panel.add(optionsPanel, BorderLayout.CENTER);


        submitButton = new JButton("Submit");
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                submitAnswer();
            }
        });
        panel.add(submitButton, BorderLayout.SOUTH);

        frame.getContentPane().add(panel);
        displayQuestion();
        frame.setVisible(true);
    }

    private void displayQuestion() {
        if (currentQuestionIndex >= 0 && currentQuestionIndex < questions.length) {
            Question currentQuestion = questions[currentQuestionIndex];
            questionLabel.setText(currentQuestion.getQuestion());
            String[] options = currentQuestion.getOptions();
            option1RadioButton.setText(options[0]);
            option2RadioButton.setText(options[1]);
            option3RadioButton.setText(options[2]);
            option4RadioButton.setText(options[3]);
            questionid.setText(options[4]);

        } else {
            showResult();
        }
    }

    private void submitAnswer() {
        Question currentQuestion = questions[currentQuestionIndex];
        int selectedOption = getSelectedOption();

        if (selectedOption == currentQuestion.getCorrectOption()) {
            score++;
        }

        currentQuestionIndex++;
        displayQuestion();
    }
    private int getSelectedOption() {
        if (option1RadioButton.isSelected()) {
            return 0;
        } else if (option2RadioButton.isSelected()) {
            return 1;
        } else if (option3RadioButton.isSelected()) {
            return 2;
        } else if (option4RadioButton.isSelected()) {
            return 3;
        } else {
            return -1;
        }
    }

    private void showResult() {
             String driverClassName = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        String username = "suchith1";
        String pass = "2004";

try {
    // Load the JDBC driver
    Class.forName(driverClassName);

    // Establish a connection to the database
    Connection con = DriverManager.getConnection(url, username, pass);

    // Perform database operations using the connection
Statement stmt=con.createStatement();
int a=stmt.executeUpdate("insert into scores values("+id+","+score+")");
if(a>0)
JOptionPane.showMessageDialog(null, "Inserted successfully", "Error",
JOptionPane.INFORMATION_MESSAGE);
else
JOptionPane.showMessageDialog(null, "Fail", "Error", JOptionPane.ERROR_MESSAGE);
    // Close the connection
    con.close();

    System.out.println("Connection closed successfully.");
} catch (ClassNotFoundException s) {
    System.err.println("Failed to load JDBC driver: " + s.getMessage());
} catch (SQLException s) {
    System.err.println("Failed to connect to the database: " + s.getMessage());
}

        JOptionPane.showMessageDialog(frame, "Quiz completed!\nScore: " + score + " out of "
+ questions.length);
```

```
        System.exit(0);
    }

  /* public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new QuizApplication1();
            }
        });
    }
*/
    private class Question {
        private String question;
        private String[] options;
        private int correctOption;

        public Question(String question, String[] options, int correctOption) {
            this.question = question;
            this.options = options;
            this.correctOption = correctOption;
        }

        public String getQuestion() {
            return question;
        }

        public String[] getOptions() {
            return options;
        }

        public int getCorrectOption() {
            return correctOption;
        }
    }
}
```

2.quiz:

```
import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class QuizApplication1 {
    private JFrame frame;
```

```
   private JLabel questionLabel;
   private JLabel questionid;
   private JRadioButton option1RadioButton;
   private JRadioButton option2RadioButton;
   private JRadioButton option3RadioButton;
   private JRadioButton option4RadioButton;

   private JButton submitButton;
   public int id;
   private int currentQuestionIndex;
   private int score;

   private Question[] questions = {
         new Question("Question 1: Which type of data can be stored in database? ",
               new String[]{"image oriented data", "text files", "data in form of audio", "all th
above ","QuestionID :vceoo1"}, 3),
         new Question("Question 2: Which of following is not a database?",
               new String[]{"heiarchial", "network", "decentralized", "distributed","QuestionID
:vceoo2"}, 2),
         new Question("Question 3: Which is not a dbms?",
               new String[]{"My sql", "microsoft acess", "google", "ibm db2","QuestionID
:vceoo3"},2 ),
         new Question("Question 4: Who painted the Mona Lisa?",
               new String[]{"Leonardo da Vinci", "Pablo Picasso", "Vincent van Gogh",
"Michelangelo","QuestionID :vceoo4"}, 0),
         new Question("Question 5: What is the chemical symbol for gold?",
               new String[]{"Au", "Ag", "Fe", "Cu","QuestionID :vceoo5"}, 0)
   };


   public QuizApplication1(int x) {
          this.id=x;
      currentQuestionIndex = 0;
      score = 0;
      initialize();
   }


   private void initialize() {
      frame = new JFrame("Quiz Application");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(400, 300);

      JPanel panel = new JPanel();
      panel.setLayout(new BorderLayout());

      questionLabel = new JLabel();
```

```
        panel.add(questionLabel, BorderLayout.NORTH);

        JPanel optionsPanel = new JPanel();
        optionsPanel.setLayout(new GridLayout(5, 1));

        option1RadioButton = new JRadioButton();
        optionsPanel.add(option1RadioButton);

        option2RadioButton = new JRadioButton();
        optionsPanel.add(option2RadioButton);

        option3RadioButton = new JRadioButton();
        optionsPanel.add(option3RadioButton);

        option4RadioButton = new JRadioButton();
        optionsPanel.add(option4RadioButton);

        questionid=new JLabel();
        optionsPanel.add(questionid);


        ButtonGroup buttonGroup = new ButtonGroup();
        buttonGroup.add(option1RadioButton);
        buttonGroup.add(option2RadioButton);
        buttonGroup.add(option3RadioButton);
        buttonGroup.add(option4RadioButton);





        panel.add(optionsPanel, BorderLayout.CENTER);


        submitButton = new JButton("Submit");
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                submitAnswer();
            }
        });
        panel.add(submitButton, BorderLayout.SOUTH);

        frame.getContentPane().add(panel);
        displayQuestion();
        frame.setVisible(true);
    }
```

```
private void displayQuestion() {
    if (currentQuestionIndex >= 0 && currentQuestionIndex < questions.length) {
        Question currentQuestion = questions[currentQuestionIndex];
        questionLabel.setText(currentQuestion.getQuestion());
        String[] options = currentQuestion.getOptions();
        option1RadioButton.setText(options[0]);
        option2RadioButton.setText(options[1]);
        option3RadioButton.setText(options[2]);
        option4RadioButton.setText(options[3]);
        questionid.setText(options[4]);

    } else {
        showResult();
    }
}

private void submitAnswer() {
    Question currentQuestion = questions[currentQuestionIndex];
    int selectedOption = getSelectedOption();

    if (selectedOption == currentQuestion.getCorrectOption()) {
        score++;
    }

    currentQuestionIndex++;
    displayQuestion();
}

private int getSelectedOption() {
    if (option1RadioButton.isSelected()) {
        return 0;
    } else if (option2RadioButton.isSelected()) {
        return 1;
    } else if (option3RadioButton.isSelected()) {
        return 2;
    } else if (option4RadioButton.isSelected()) {
        return 3;
    } else {
        return -1;
    }
}

private void showResult() {
         String driverClassName = "oracle.jdbc.driver.OracleDriver";
    String url = "jdbc:oracle:thin:@localhost:1521:xe";
    String username = "suchith1";
    String pass = "2004";
```

```
try {
   // Load the JDBC driver
   Class.forName(driverClassName);

   // Establish a connection to the database
   Connection con = DriverManager.getConnection(url, username, pass);

   // Perform database operations using the connection
Statement stmt=con.createStatement();
int a=stmt.executeUpdate("insert into scores values("+id+","+score+")");
if(a>0)
JOptionPane.showMessageDialog(null, "Inserted successfully", "Error",
JOptionPane.INFORMATION_MESSAGE);
else
JOptionPane.showMessageDialog(null, "Fail", "Error", JOptionPane.ERROR_MESSAGE);
   // Close the connection
   con.close();

   System.out.println("Connection closed successfully.");
} catch (ClassNotFoundException s) {
   System.err.println("Failed to load JDBC driver: " + s.getMessage());
} catch (SQLException s) {
   System.err.println("Failed to connect to the database: " + s.getMessage());
}

      JOptionPane.showMessageDialog(frame, "Quiz completed!\nScore: " + score + " out of "
+ questions.length);
      System.exit(0);
   }

  /* public static void main(String[] args) {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            new QuizApplication1();
         }
      });
   }
*/
   private class Question {
      private String question;
      private String[] options;
      private int correctOption;

      public Question(String question, String[] options, int correctOption) {
         this.question = question;
         this.options = options; import javax.swing.*;
```

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class QuizApplication1 {
    private JFrame frame;
    private JLabel questionLabel;
    private JLabel questionid;
    private JRadioButton option1RadioButton;
    private JRadioButton option2RadioButton;
    private JRadioButton option3RadioButton;
    private JRadioButton option4RadioButton;

    private JButton submitButton;
    public int id;
    private int currentQuestionIndex;
    private int score;

    private Question[] questions = {
            new Question("Question 1: Which type of data can be stored in database? ",
                    new String[]{"image oriented data", "text files", "data in form of audio", "all th
above ","QuestionID :vceoo1"}, 3),
            new Question("Question 2: Which of following is not a database?",
                    new String[]{"heiarchial", "network", "decentralized", "distributed","QuestionID
:vceoo2"}, 2),
            new Question("Question 3: Which is not a dbms?",
                    new String[]{"My sql", "microsoft acess", "google", "ibm db2","QuestionID
:vceoo3"},2 ),
            new Question("Question 4: Who painted the Mona Lisa?",
                    new String[]{"Leonardo da Vinci", "Pablo Picasso", "Vincent van Gogh",
"Michelangelo","QuestionID :vceoo4"}, 0),
            new Question("Question 5: What is the chemical symbol for gold?",
                    new String[]{"Au", "Ag", "Fe", "Cu","QuestionID :vceoo5"}, 0)
    };


    public QuizApplication1(int x) {
             this.id=x;
        currentQuestionIndex = 0;
        score = 0;
        initialize();
    }


    private void initialize() {
```

```
frame = new JFrame("Quiz Application");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(400, 300);

JPanel panel = new JPanel();
panel.setLayout(new BorderLayout());

questionLabel = new JLabel();
panel.add(questionLabel, BorderLayout.NORTH);

JPanel optionsPanel = new JPanel();
optionsPanel.setLayout(new GridLayout(5, 1));

option1RadioButton = new JRadioButton();
optionsPanel.add(option1RadioButton);

option2RadioButton = new JRadioButton();
optionsPanel.add(option2RadioButton);

option3RadioButton = new JRadioButton();
optionsPanel.add(option3RadioButton);

option4RadioButton = new JRadioButton();
optionsPanel.add(option4RadioButton);

questionid=new JLabel();
optionsPanel.add(questionid);


ButtonGroup buttonGroup = new ButtonGroup();
buttonGroup.add(option1RadioButton);
buttonGroup.add(option2RadioButton);
buttonGroup.add(option3RadioButton);
buttonGroup.add(option4RadioButton);





panel.add(optionsPanel, BorderLayout.CENTER);


submitButton = new JButton("Submit");
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        submitAnswer();
    }
```

```
   });
   panel.add(submitButton, BorderLayout.SOUTH);

   frame.getContentPane().add(panel);
   displayQuestion();
   frame.setVisible(true);
}

private void displayQuestion() {
   if (currentQuestionIndex >= 0 && currentQuestionIndex < questions.length) {
      Question currentQuestion = questions[currentQuestionIndex];
      questionLabel.setText(currentQuestion.getQuestion());
      String[] options = currentQuestion.getOptions();
      option1RadioButton.setText(options[0]);
      option2RadioButton.setText(options[1]);
      option3RadioButton.setText(options[2]);
      option4RadioButton.setText(options[3]);
      questionid.setText(options[4]);

   } else {
      showResult();
   }
}

private void submitAnswer() {
   Question currentQuestion = questions[currentQuestionIndex];
   int selectedOption = getSelectedOption();

   if (selectedOption == currentQuestion.getCorrectOption()) {
      score++;
   }

   currentQuestionIndex++;
   displayQuestion();
}

private int getSelectedOption() {
   if (option1RadioButton.isSelected()) {
      return 0;
   } else if (option2RadioButton.isSelected()) {
      return 1;
   } else if (option3RadioButton.isSelected()) {
      return 2;
   } else if (option4RadioButton.isSelected()) {
      return 3;
   } else {
      return -1;
```

```
    }
  }

  private void showResult() {
       String driverClassName = "oracle.jdbc.driver.OracleDriver";
    String url = "jdbc:oracle:thin:@localhost:1521:xe";
    String username = "suchith1";
    String pass = "2004";

try {
  // Load the JDBC driver
  Class.forName(driverClassName);

  // Establish a connection to the database
  Connection con = DriverManager.getConnection(url, username, pass);

  // Perform database operations using the connection
Statement stmt=con.createStatement();
int a=stmt.executeUpdate("insert into scores values("+id+","+score+")");
if(a>0)
JOptionPane.showMessageDialog(null, "Inserted successfully", "Error",
JOptionPane.INFORMATION_MESSAGE);
else
JOptionPane.showMessageDialog(null, "Fail", "Error", JOptionPane.ERROR_MESSAGE);
  // Close the connection
  con.close();

  System.out.println("Connection closed successfully.");
} catch (ClassNotFoundException s) {
  System.err.println("Failed to load JDBC driver: " + s.getMessage());
} catch (SQLException s) {
  System.err.println("Failed to connect to the database: " + s.getMessage());
}

    JOptionPane.showMessageDialog(frame, "Quiz completed!\nScore: " + score + " out of "
+ questions.length);
    System.exit(0);
  }

  /* public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new QuizApplication1();
      }
    });
  }
*/
```

```
    private class Question {
        private String question;
        private String[] options;
        private int correctOption;

        public Question(String question, String[] options, int correctOption) {
            this.question = question;
            this.options = options;
            this.correctOption = correctOption;
        }

        public String getQuestion() {
            return question;
        }

        public String[] getOptions() {
            return options;
        }

        public int getCorrectOption() {
            return correctOption;
        }
    }
}
            this.correctOption = correctOption;
        }

        public String getQuestion() {
            return question;
        }

        public String[] getOptions() {
            return options;
        }

        public int getCorrectOption() {
            return correctOption;
        }
    }
}
```

3. digital imaginary:

```
import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class QuizApplication1 {
    private JFrame frame;
    private JLabel questionLabel;
    private JLabel questionid;
    private JRadioButton option1RadioButton;
    private JRadioButton option2RadioButton;
    private JRadioButton option3RadioButton;
    private JRadioButton option4RadioButton;

    private JButton submitButton;
    public int id;
    private int currentQuestionIndex;
    private int score;

    private Question[] questions = {
            new Question("Question 1: Which type of data can be stored in database? ",
                    new String[]{"image oriented data", "text files", "data in form of audio", "all th
above ","QuestionID :vceoo1"}, 3),
            new Question("Question 2: Which of following is not a database?",
                    new String[]{"heiarchial", "network", "decentralized", "distributed","QuestionID
:vceoo2"}, 2),
            new Question("Question 3: Which is not a dbms?",
                    new String[]{"My sql", "microsoft acess", "google", "ibm db2","QuestionID
:vceoo3"},2 ),
            new Question("Question 4: Who painted the Mona Lisa?",
                    new String[]{"Leonardo da Vinci", "Pablo Picasso", "Vincent van Gogh",
"Michelangelo","QuestionID :vceoo4"}, 0),
            new Question("Question 5: What is the chemical symbol for gold?",
                    new String[]{"Au", "Ag", "Fe", "Cu","QuestionID :vceoo5"}, 0)
    };


    public QuizApplication1(int x) {
             this.id=x;
        currentQuestionIndex = 0;
        score = 0;
        initialize();
    }


    private void initialize() {
        frame = new JFrame("Quiz Application");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout());

        questionLabel = new JLabel();
        panel.add(questionLabel, BorderLayout.NORTH);

        JPanel optionsPanel = new JPanel();
        optionsPanel.setLayout(new GridLayout(5, 1));

        option1RadioButton = new JRadioButton();
        optionsPanel.add(option1RadioButton);

        option2RadioButton = new JRadioButton();
        optionsPanel.add(option2RadioButton);

        option3RadioButton = new JRadioButton();
        optionsPanel.add(option3RadioButton);

        option4RadioButton = new JRadioButton();
        optionsPanel.add(option4RadioButton);

        questionid=new JLabel();
        optionsPanel.add(questionid);


        ButtonGroup buttonGroup = new ButtonGroup();
        buttonGroup.add(option1RadioButton);
        buttonGroup.add(option2RadioButton);
        buttonGroup.add(option3RadioButton);
        buttonGroup.add(option4RadioButton);





        panel.add(optionsPanel, BorderLayout.CENTER);


        submitButton = new JButton("Submit");
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                submitAnswer();
            }
        });
        panel.add(submitButton, BorderLayout.SOUTH);
```

```
        frame.getContentPane().add(panel);
        displayQuestion();
        frame.setVisible(true);
    }

    private void displayQuestion() {
        if (currentQuestionIndex >= 0 && currentQuestionIndex < questions.length) {
            Question currentQuestion = questions[currentQuestionIndex];
            questionLabel.setText(currentQuestion.getQuestion());
            String[] options = currentQuestion.getOptions();
            option1RadioButton.setText(options[0]);
            option2RadioButton.setText(options[1]);
            option3RadioButton.setText(options[2]);
            option4RadioButton.setText(options[3]);
            questionid.setText(options[4]);

        } else {
            showResult();
        }
    }

    private void submitAnswer() {
        Question currentQuestion = questions[currentQuestionIndex];
        int selectedOption = getSelectedOption();

        if (selectedOption == currentQuestion.getCorrectOption()) {
            score++;
        }

        currentQuestionIndex++;
        displayQuestion();
    }

    private int getSelectedOption() {
        if (option1RadioButton.isSelected()) {
            return 0;
        } else if (option2RadioButton.isSelected()) {
            return 1;
        } else if (option3RadioButton.isSelected()) {
            return 2;
        } else if (option4RadioButton.isSelected()) {
            return 3;
        } else {
            return -1;
        }
    }
```

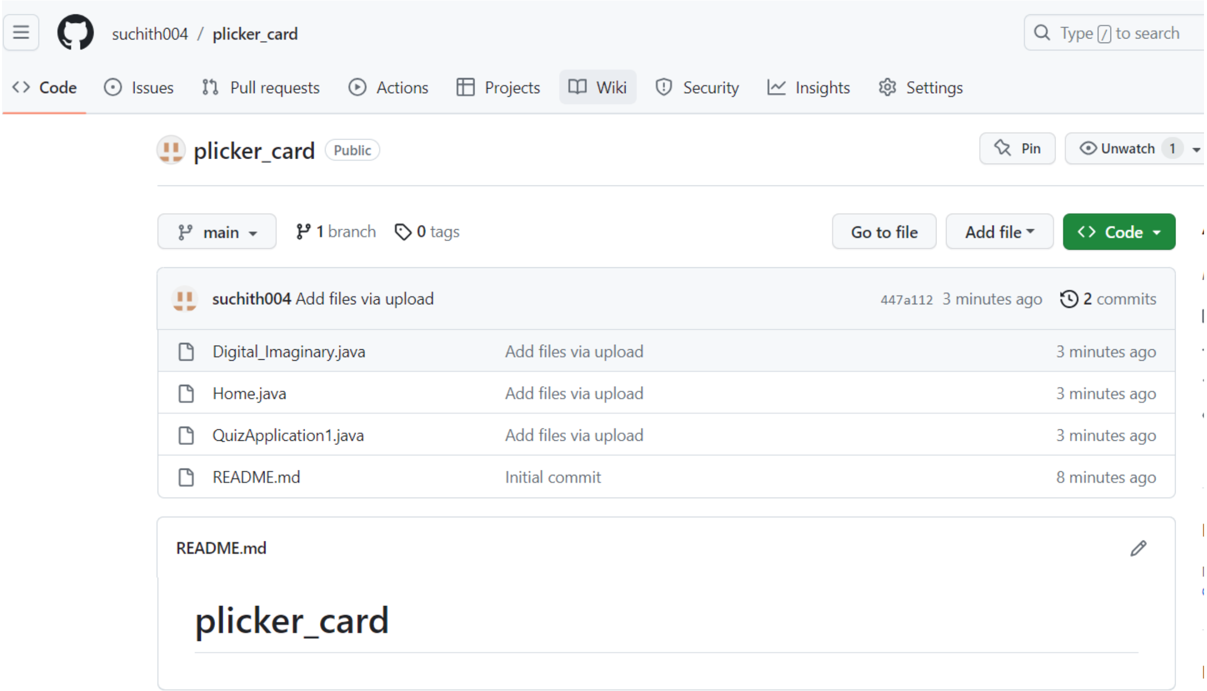
```
  private void showResult() {
        String driverClassName = "oracle.jdbc.driver.OracleDriver";
    String url = "jdbc:oracle:thin:@localhost:1521:xe";
    String username = "suchith1";
    String pass = "2004";

try {
  // Load the JDBC driver
  Class.forName(driverClassName);

  // Establish a connection to the database
  Connection con = DriverManager.getConnection(url, username, pass);

  // Perform database operations using the connection
Statement stmt=con.createStatement();
int a=stmt.executeUpdate("insert into scores values("+id+","+score+")");
if(a>0)
JOptionPane.showMessageDialog(null, "Inserted successfully", "Error",
JOptionPane.INFORMATION_MESSAGE);
else
JOptionPane.showMessageDialog(null, "Fail", "Error", JOptionPane.ERROR_MESSAGE);
  // Close the connection
  con.close();

  System.out.println("Connection closed successfully.");
} catch (ClassNotFoundException s) {
  System.err.println("Failed to load JDBC driver: " + s.getMessage());
} catch (SQLException s) {
  System.err.println("Failed to connect to the database: " + s.getMessage());
}

    JOptionPane.showMessageDialog(frame, "Quiz completed!\nScore: " + score + " out of "
+ questions.length);
    System.exit(0);
  }

 /* public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new QuizApplication1();
      }
    });
  }
*/
  private class Question {
    private String question;
```

```
        private String[] options;
        private int correctOption;

        public Question(String question, String[] options, int correctOption) {
            this.question = question;
            this.options = options;
            this.correctOption = correctOption;
        }

        public String getQuestion() {
            return question;
        }

        public String[] getOptions() {
            return options;
        }

        public int getCorrectOption() {
            return correctOption;
        }
    }
}
```

GitHub Links and Folder Structure:

Link : https://github.com/suchith004/plicker_card

Folder Structure:



suchith004 Add files via upload 447a112 3 minutes ago 2 commits

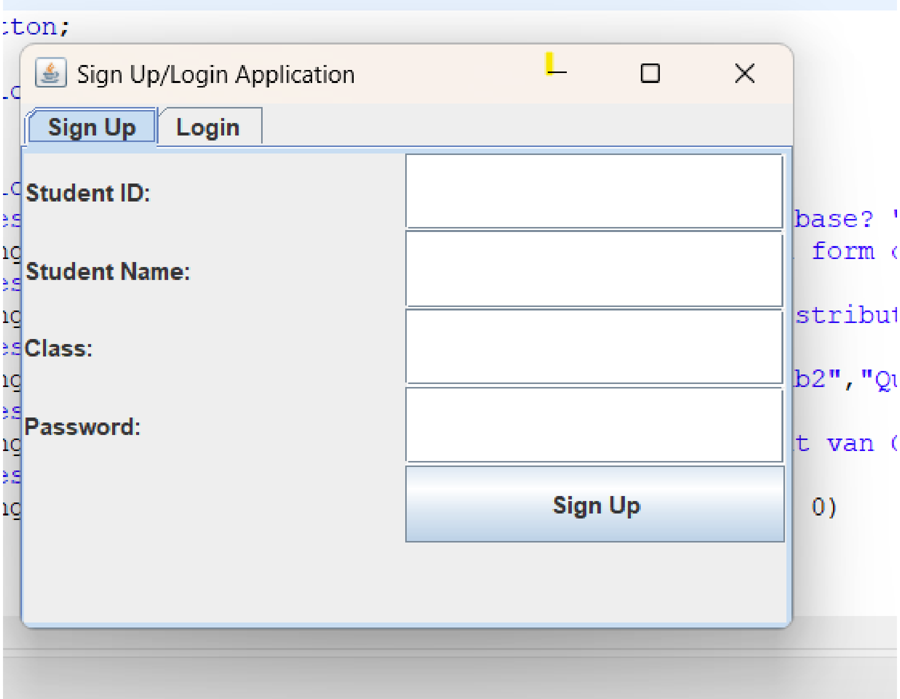
Digital_Imaginary.java	Add files via upload	3 minutes ago
Home.java	Add files via upload	3 minutes ago
QuizApplication1.java	Add files via upload	3 minutes ago
README.md	Initial commit	8 minutes ago

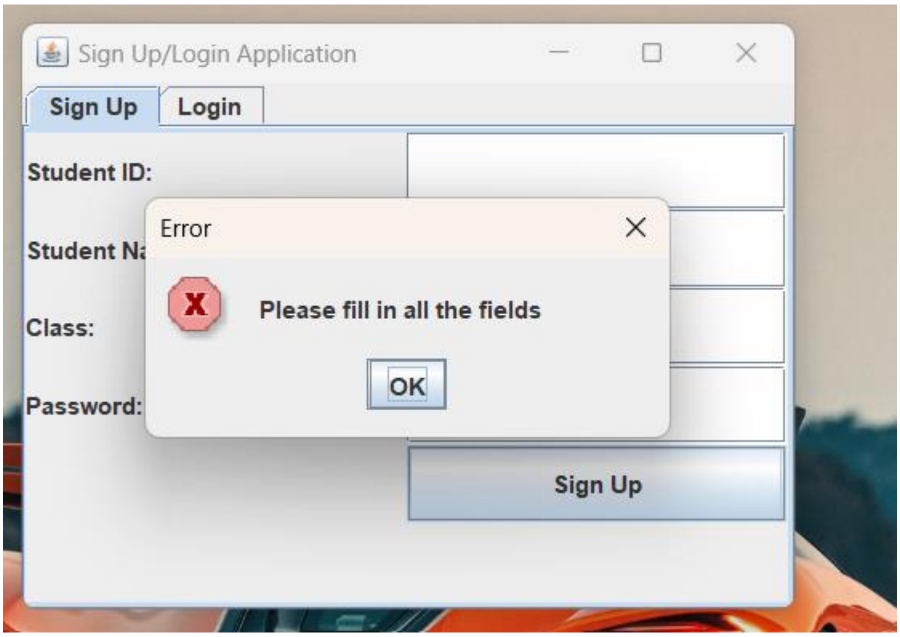
README.md

plicker_card

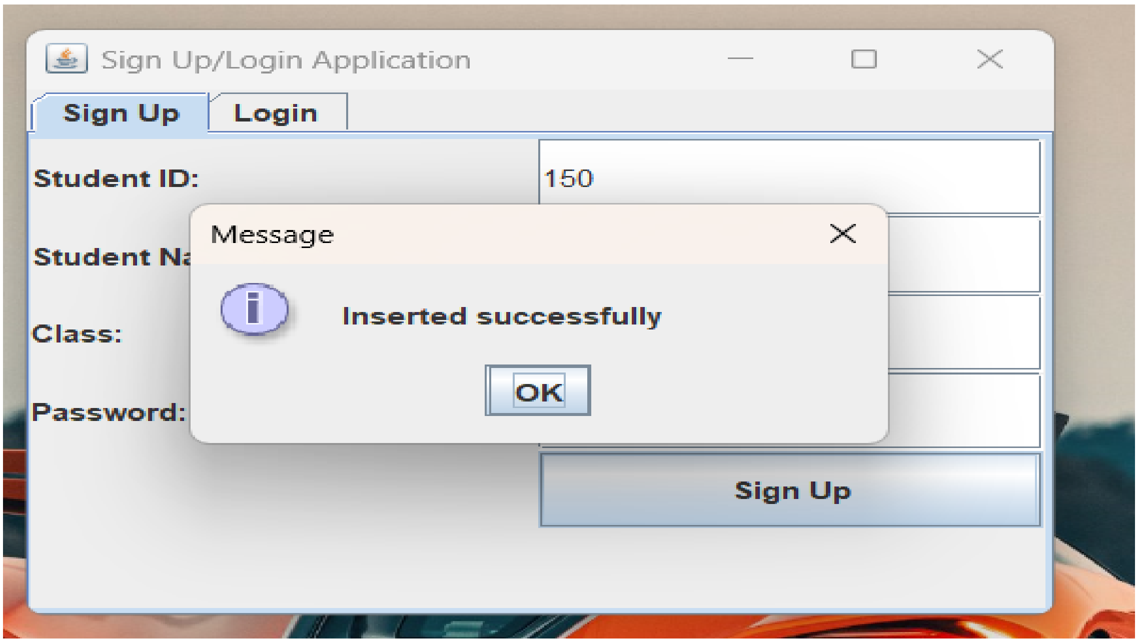
TESTING

1. login/signup

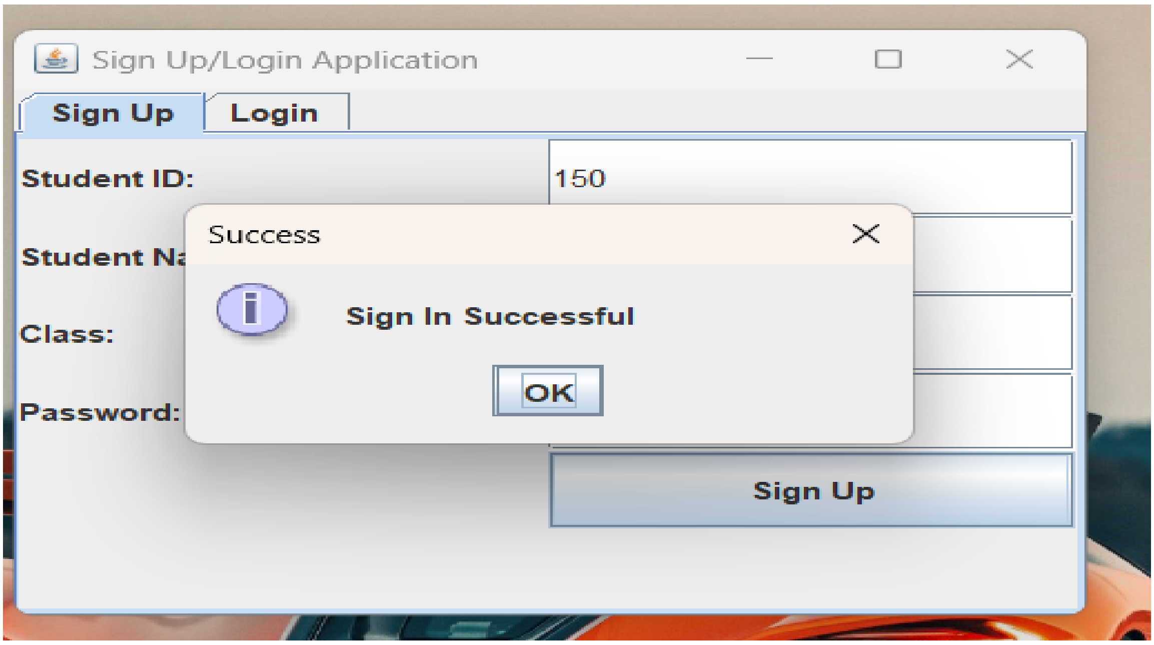




Please fill in all the fields

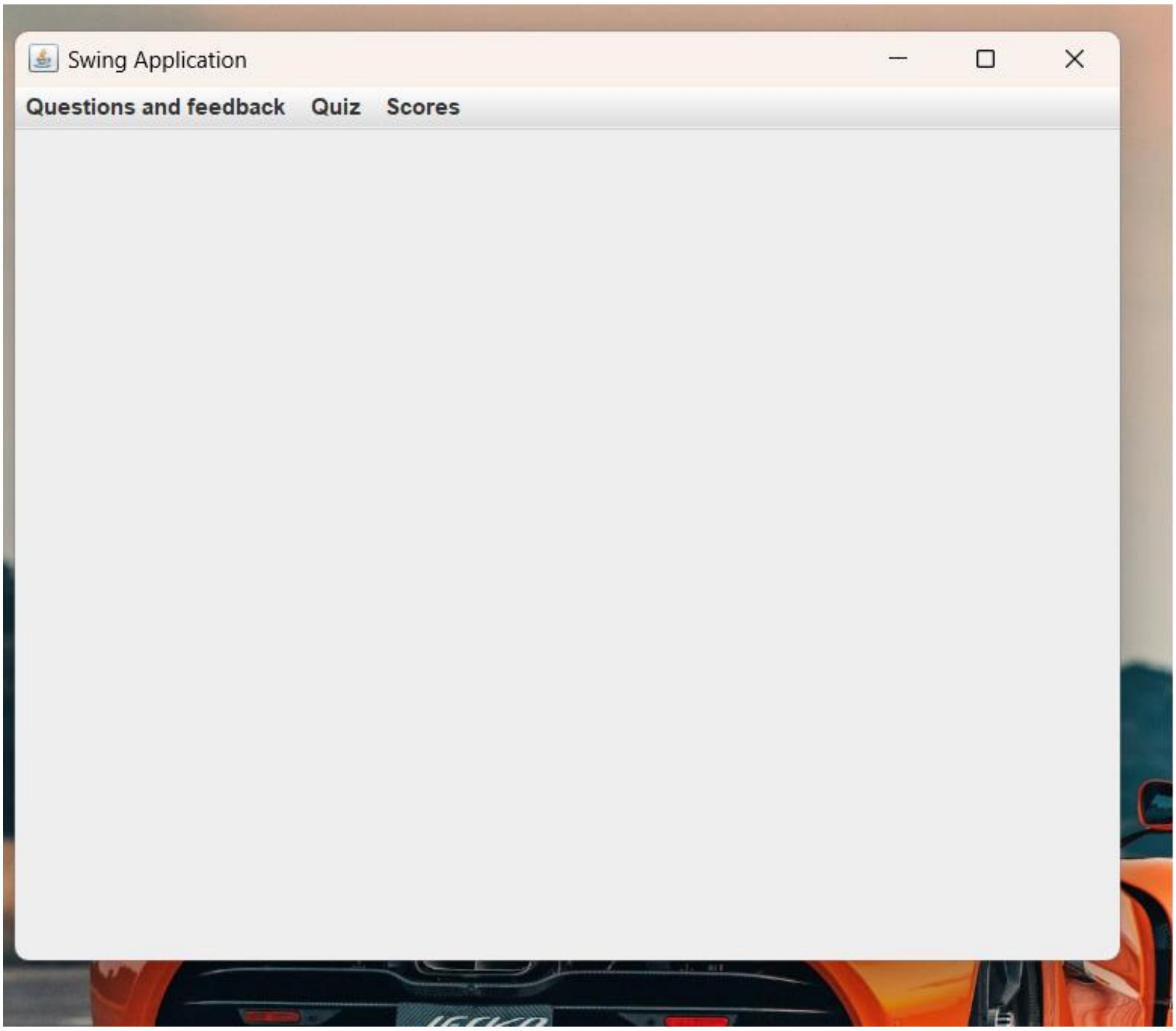


Inserted successfully

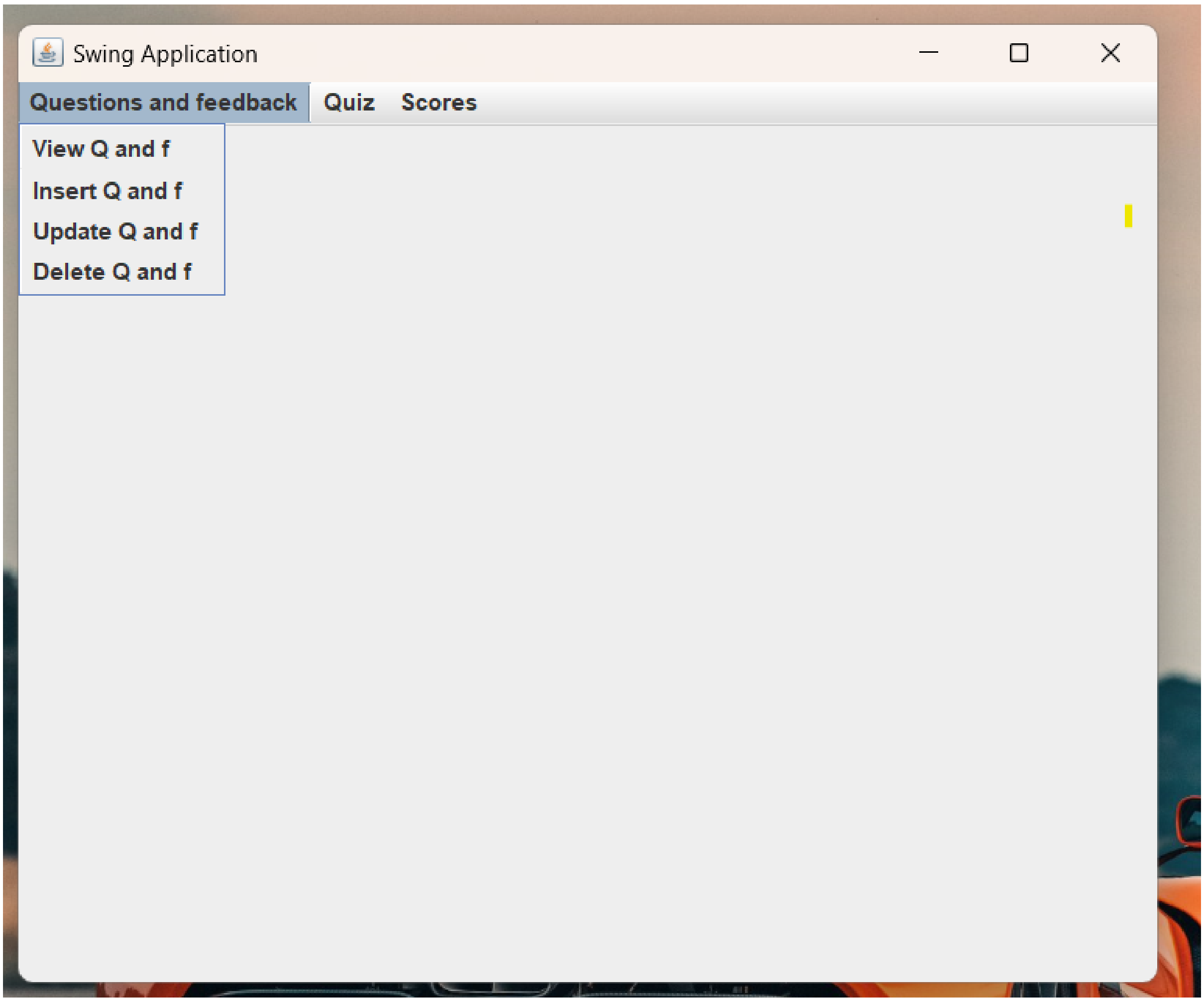


Sign In Successful

2. Home



3. Insert / update / delete



Swing Application

Questions and feedback

Quiz

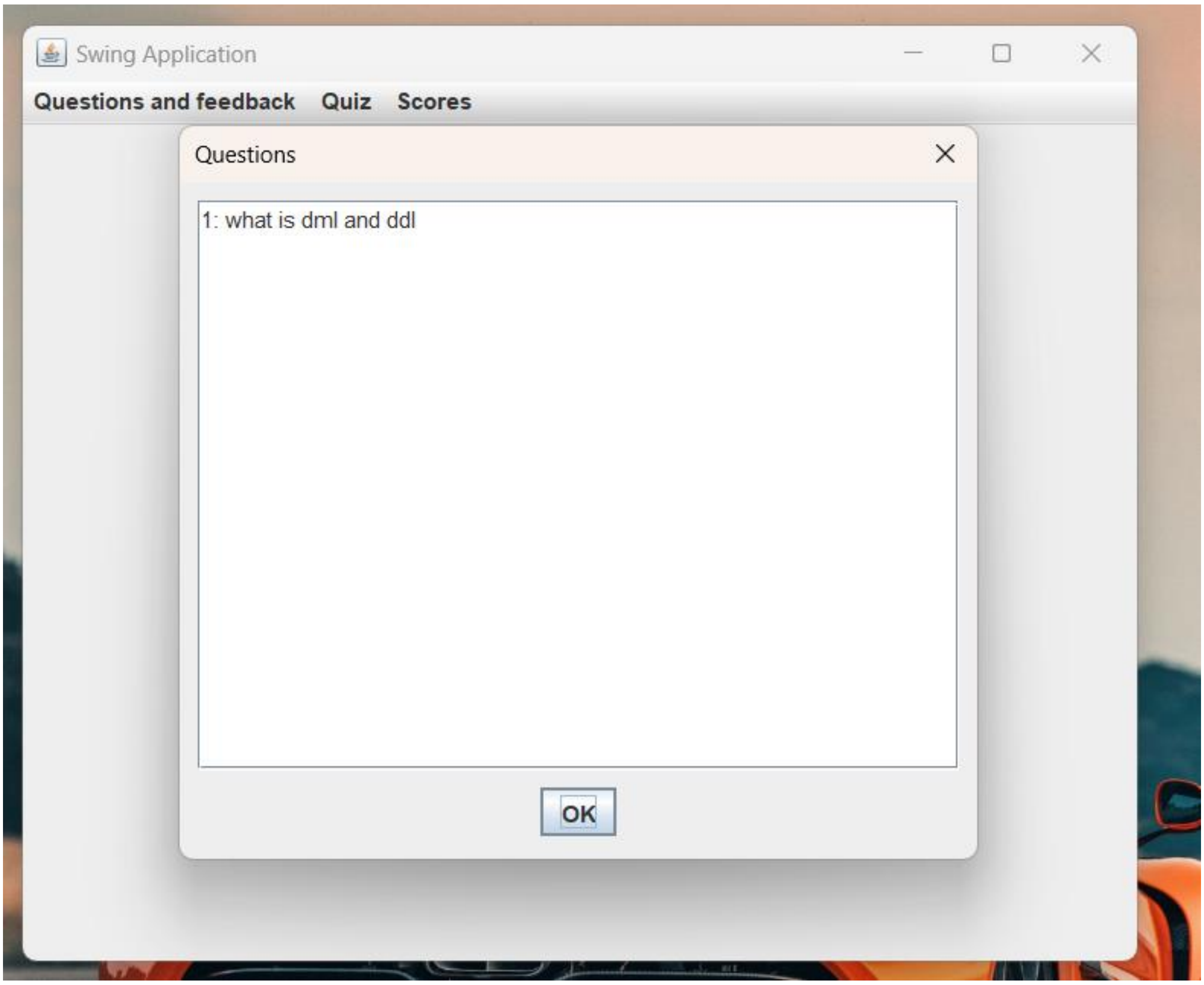
Scores

View Q and f

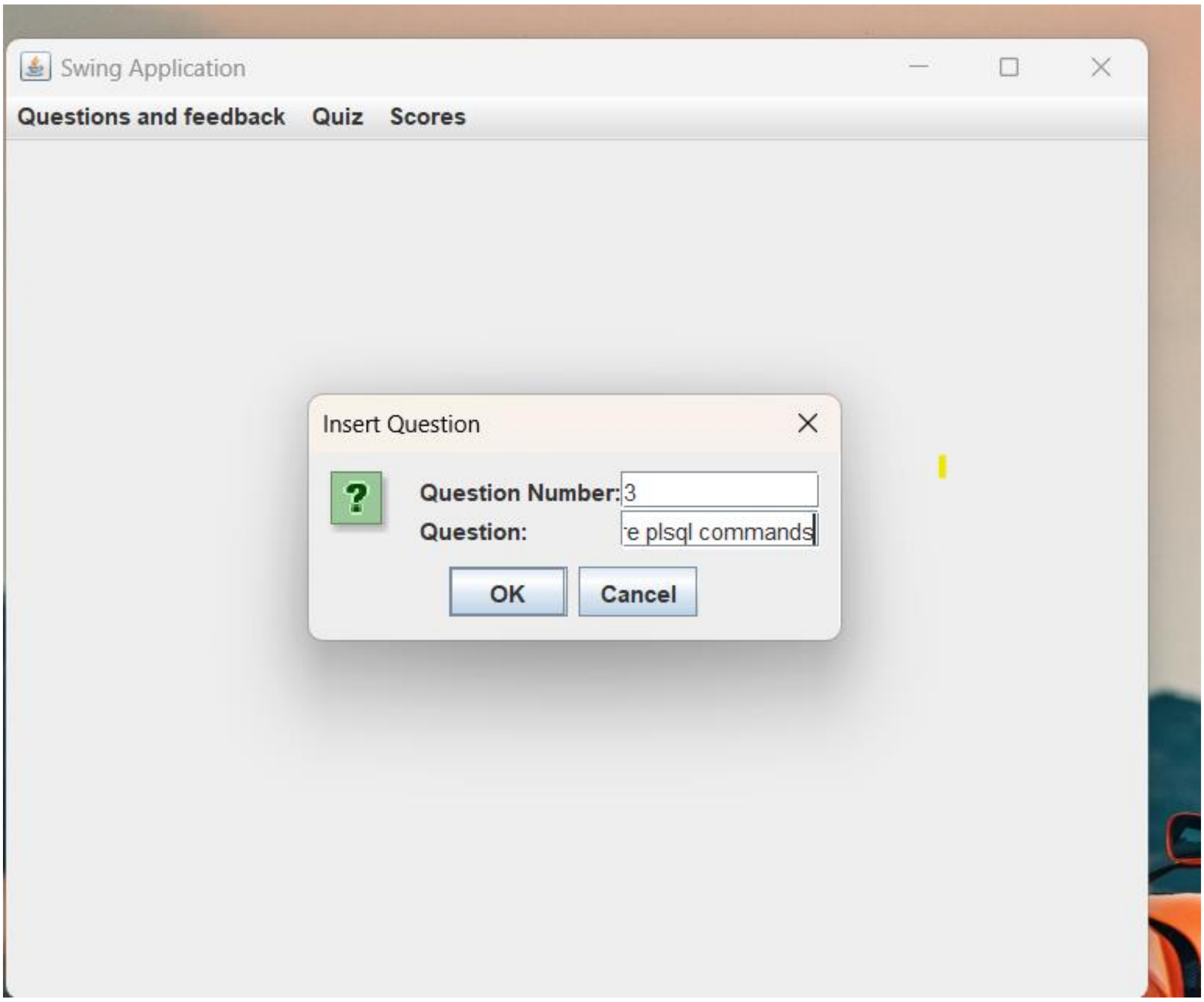
Insert Q and f

Update Q and f

Delete Q and f

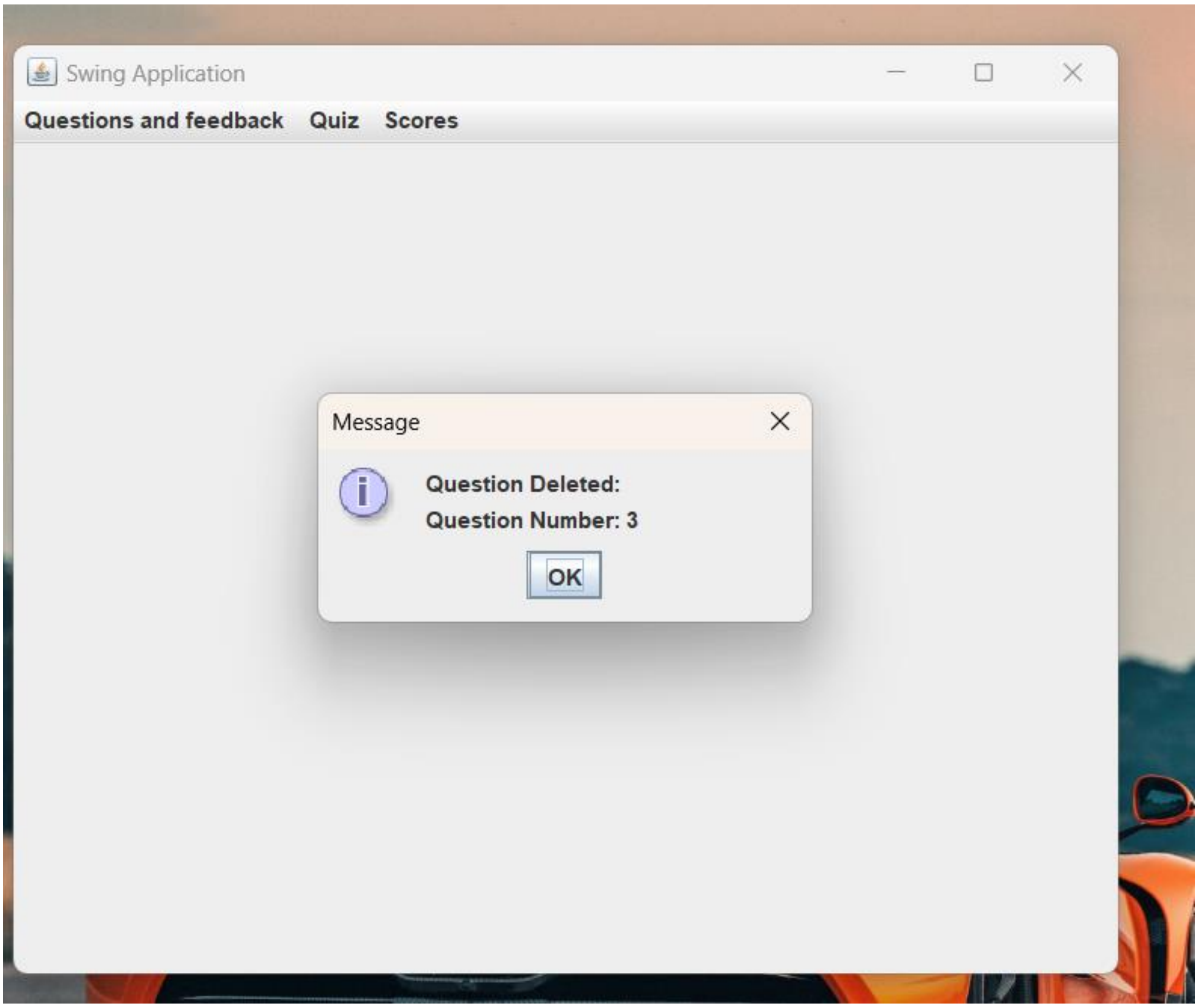


1: what is dml and ddl



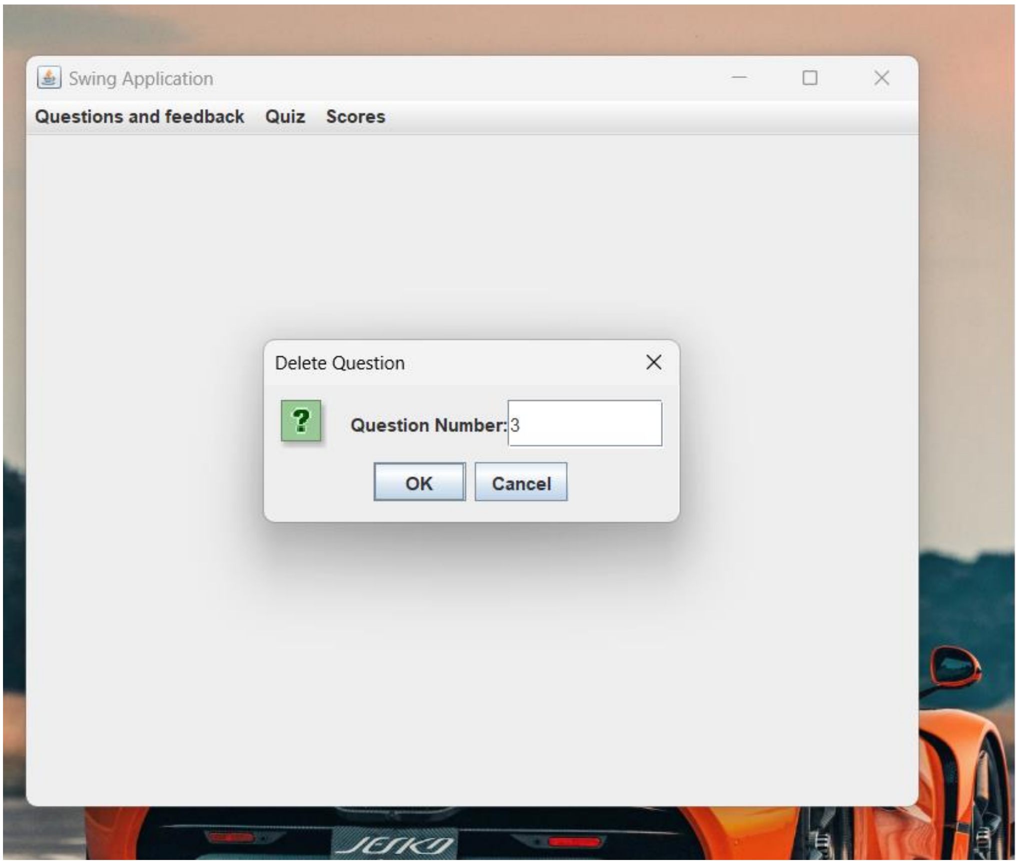
Question Number: 3

Question: e plsql commands



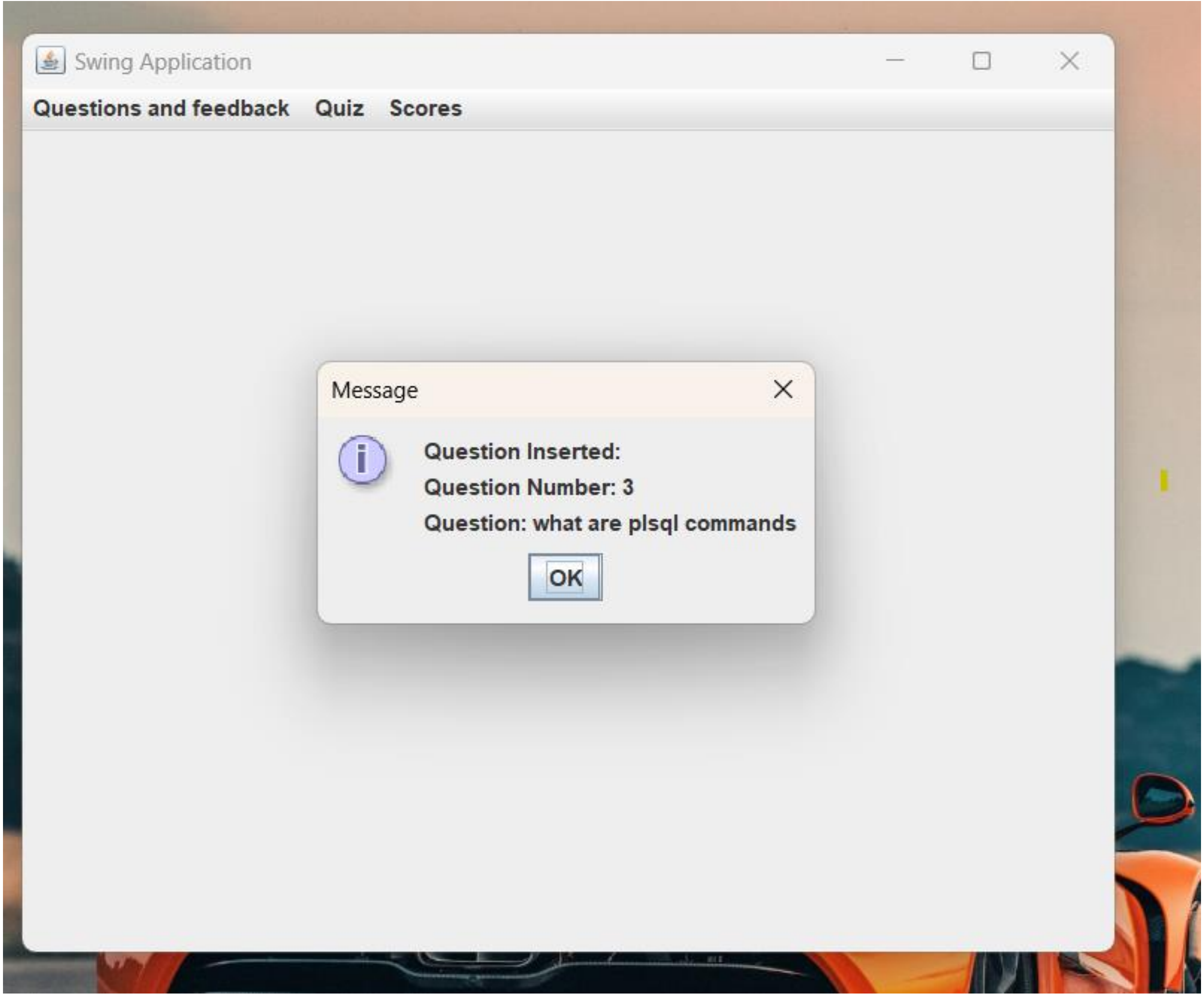
Question Deleted:

Question Number: 3



Delete Question

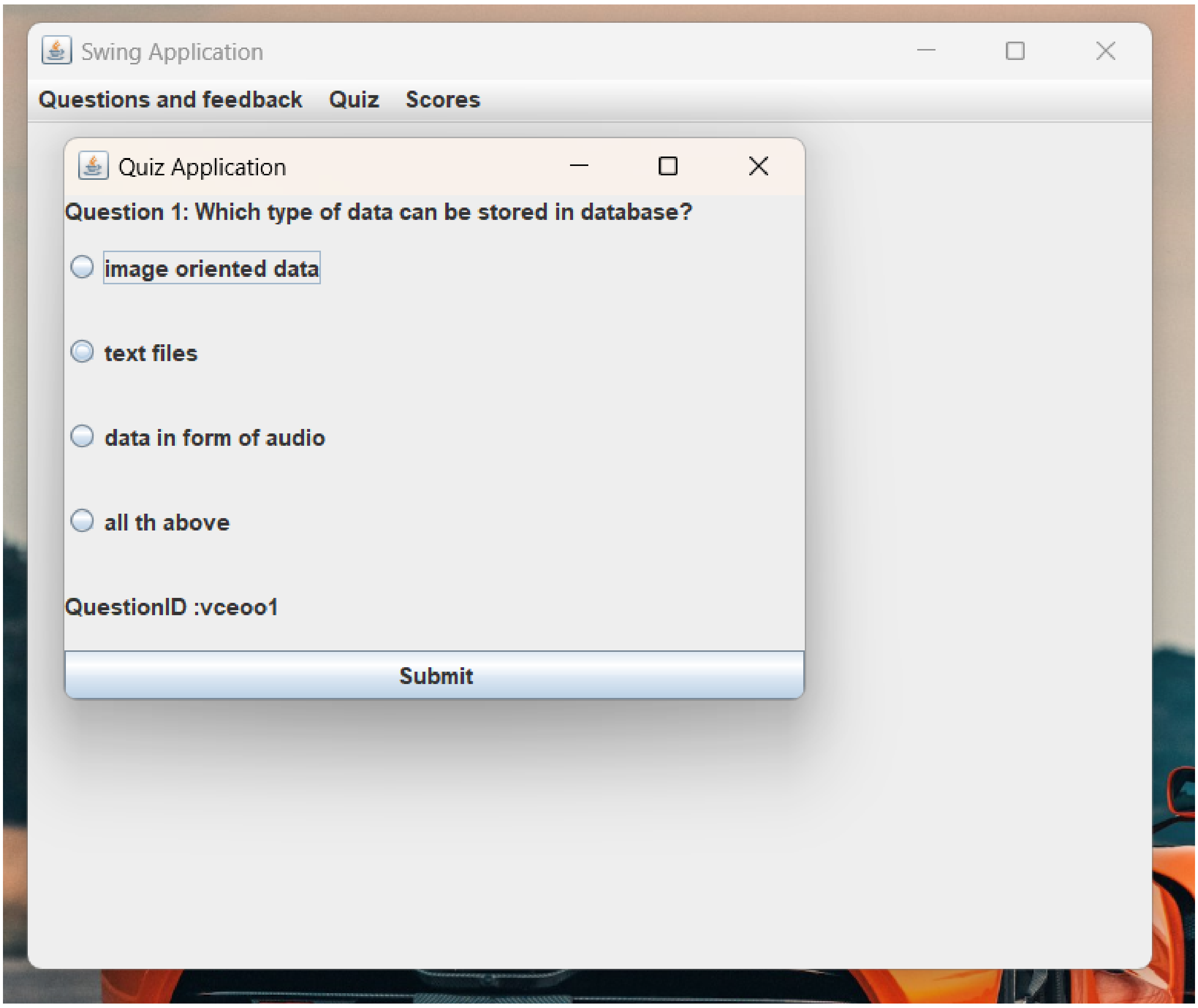
Question Number: 3



Message

Question Inserted:
Question Number: 3
Question: what are plsql commands

4.Quiz



Question 1: Which type of data can be stored in database?

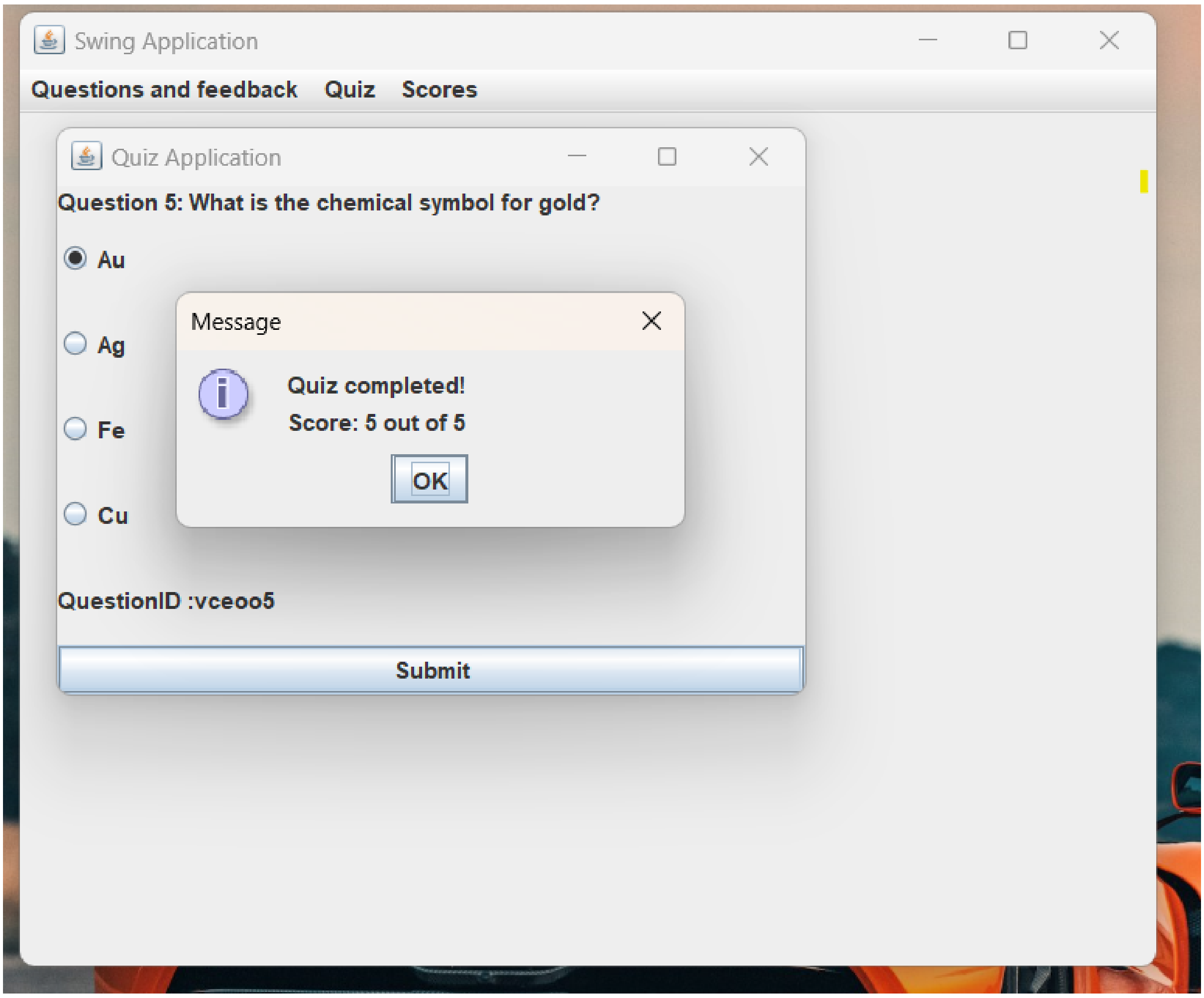
○ image oriented data

○ text files

○ data in form of audio

○ all th above

QuestionID :vceoo1



Question 5: What is the chemical symbol for gold?

● Au

○ Ag

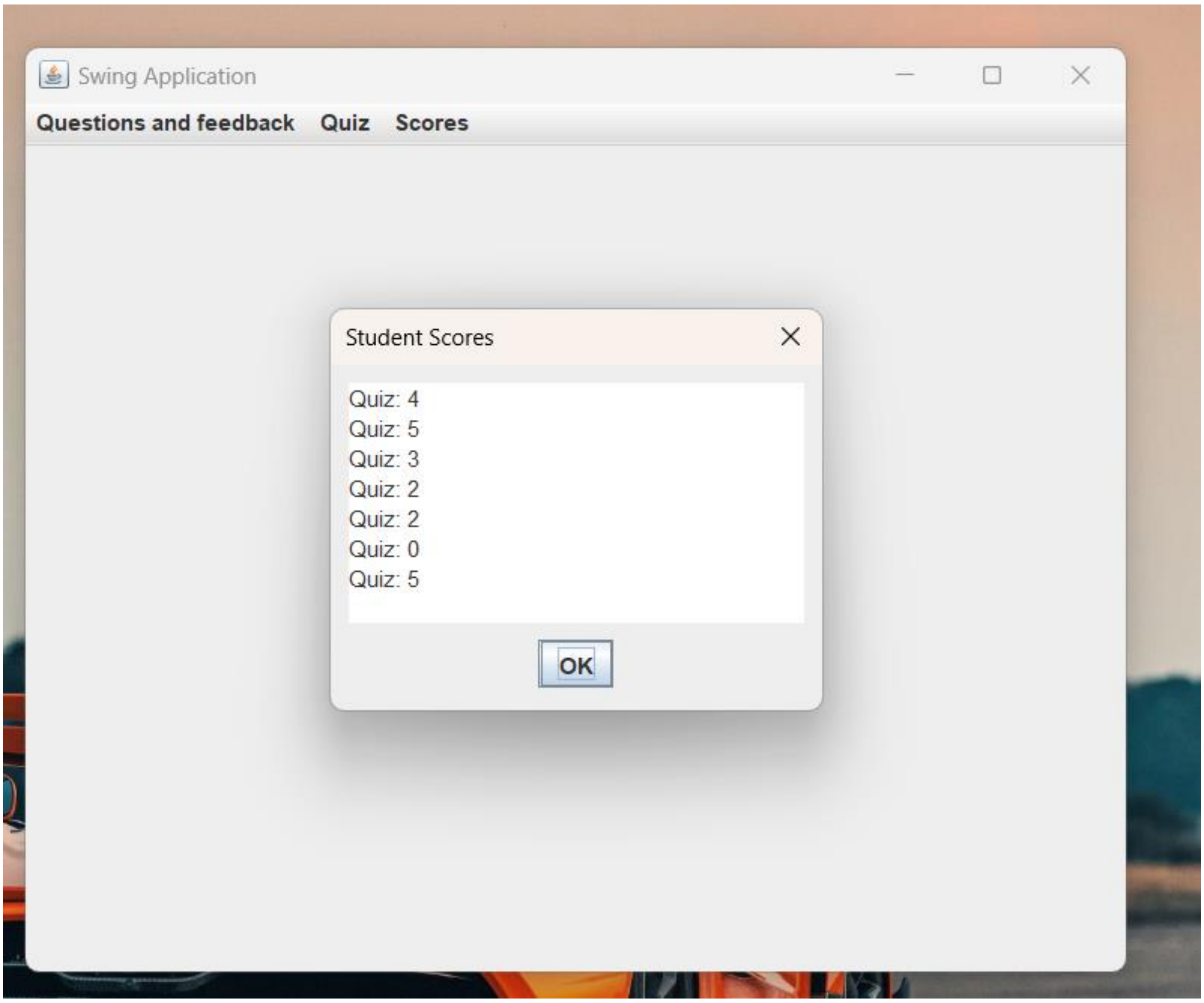
○ Fe

○ Cu

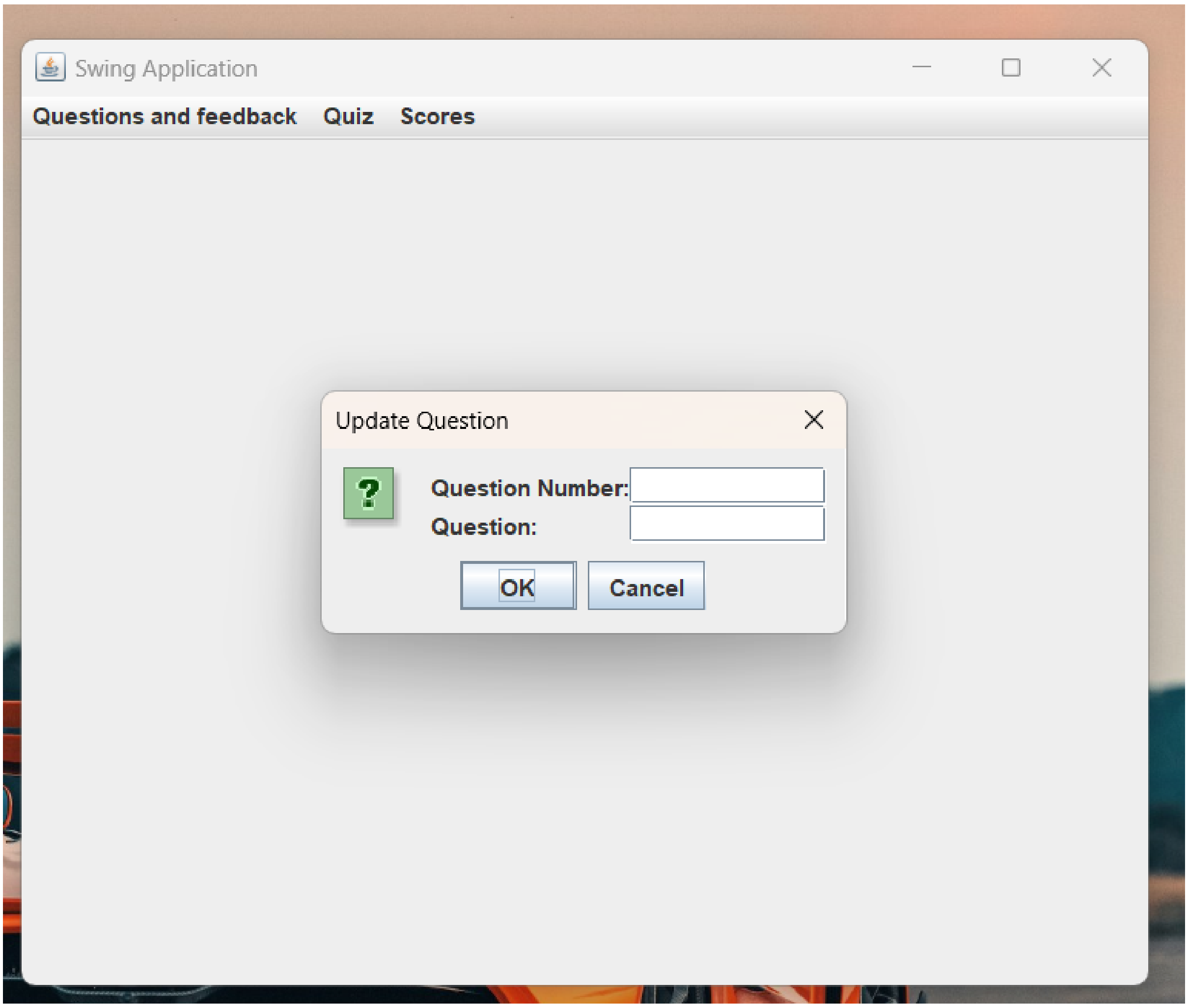
QuestionID :vceoo5

Quiz completed!
Score: 5 out of 5

5.scores



```
Quiz: 4
Quiz: 5
Quiz: 3
Quiz: 2
Quiz: 2
Quiz: 0
Quiz: 5
```



Swing Application
Questions and feedback
Quiz
Scores
Update Question
Question Number:
Question:
OK
Cancel

RESULTS

I have successfully completed the mini-project “Plicker Cards” .

DISCUSSION AND FUTURE WORK

This project contains the basic interaction of giving information by students for suggesting the correct career choice. It has a very basic user interface. Future scope would be to make the UI more appealing by using graphics. more feature would be to allow student-users to upload their resumes and official One documents required so that we can suggest more accurate career choices. We can also think of including a feedback system to allow the users to leave their valuable feedback after using this app. Making this feedback to be publicly viewable, would attract many more users to use this app.

REFERENCES

- https://docs.oracle.com/javase/7/docs/api/
- https://www.javatpoint.com/java-swing
- https://stackoverflow.com