JDBC Skill – 1 190031920, A. Nikhil Reddy

1. **Given question**

Develop a JDBC application where KL University students will be able to choose their specialization. But there will be limited slots for each specialization which must be stored and queried from database. After successfully selecting a specialization, Specialization along with student details must be stored in the database.

Complete code in java:

```
import java.sql.*;

import java.util.*;


class chooseSpecalizationSkill1 {

   public static void main(String[] args) throws Exception {

      Class.forName("oracle.jdbc.driver.OracleDriver");


      String dbUserName = "system";


      String dbUserPassword = "nikhil";


      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
XE", dbUserName, dbUserPassword);


      Statement stmt = con.createStatement();


      Scanner scanner = new Scanner(System.in);


      create_specalization_table(stmt, con);


      create_student_choice_table(stmt);
```

```
boolean exit = false;

while (!exit) {
    System.out.println("1. Select a specalization");
    System.out.println("2. View students choices");
    System.out.println("3. Exit the program");

    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            chooseSpecalization(stmt, con, scanner);
            break;

        case 2:
            print_student_table_data_to_console(stmt);
            break;

        case 3:
            exit = true;
            break;

        default:
            System.out.println("Please Try Again");
            break;
    }

}
```

```
        con.close();

        scanner.close();
    }

    public static void chooseSpecalization(Statement stmt, Connection con, Scanner sc) {
        System.out.println("Please enter your id number ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.println("Please select a specalization from the given list");
        System.out.println("Available specalizations");
        System.out.println("Specalization name      |     Slots available");
        print_available_specalizations(stmt);
        System.out.println("Please enter your preffered specalization name");
        String specalization_choice = sc.nextLine();
        if (is_selection_valid(con, specalization_choice)) {
            reduce_available_slots(con, specalization_choice);
            insert_into_student_table(con, id, specalization_choice);
            System.out.println("Specalization selected successfully");
        } else {
            System.out.println("please select a valid specalization from the given list");
        }

    }

    public static void reduce_available_slots(Connection con, String specalization_choice)
{
        try {
```

```
        String reduce_available_slots_query = "UPDATE specalization SET slots_available
= slots_available-1 WHERE specalization_name=?";


        PreparedStatement pstmt = con.prepareStatement(reduce_available_slots_query
);


        pstmt.setString(1, specalization_choice);
        pstmt.executeUpdate();
        System.out.println("decreased available slots succesfully");
      } catch (Exception e) {
        System.out.println("Error" + e);
      }
    }


    public static void insert_into_student_table(Connection con, int id, String specalizatio
n_choice) {
      try {
        String insert_into_student_table_query = "INSERT INTO student_choice(student_i
d, specalization_name) VALUES(?, ?)";
        PreparedStatement pstmt = con.prepareStatement(insert_into_student_table_qu
ery);


        pstmt.setInt(1, id);
        pstmt.setString(2, specalization_choice);
        pstmt.executeUpdate();


        System.out.println("inserted into Student database succesfully");
      } catch (Exception e) {
        System.out.println("Error: " + e);
      }
    }
```

```
    public static boolean is_selection_valid(Connection con, String specalization_choice) {
        try {

            boolean isValid = false;
            String is_choice_valid_query = "SELECT * FROM specalization WHERE specializatio
n_name=?";

            PreparedStatement pstmt = con.prepareStatement(is_choice_valid_query);

            pstmt.setString(1, specalization_choice);

            ResultSet rs = pstmt.executeQuery();
            if (!rs.isBeforeFirst()) {
                return isValid;
            } else {
                rs.next();
                if (rs.getInt(2) > 1) {
                    isValid = true;
                    return isValid;
                }
            }
            return isValid;
        } catch (Exception e) {
            System.out.println("Error: " + e);
            return false;
        }
    }

    public static void print_available_specalizations(Statement stmt) {
```

```
        try {
            ResultSet rs = stmt.executeQuery("select * from specalization where slots_availab
le>0");
            if (!rs.isBeforeFirst()) {
                System.out.println("All specalizations are full");
            } else {
                while (rs.next()) {
                    System.out.format("%24s|%10d", rs.getString(1), rs.getInt(2));
                    System.out.println();
                    // System.out.println(rs.getString(1) + " " + rs.getInt(2));
                }
            }
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }

    public static void create_student_choice_table(Statement stmt) {
        try {
            String create_student_specalization_choice_table_query = "CREATE TABLE studen
t_choice( student_id INTEGER, specalization_name VARCHAR(22))";
            stmt.executeUpdate(create_student_specalization_choice_table_query);
            System.out.println("student specalization choice table created succesfully");
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }

    public static void create_specalization_table(Statement stmt, Connection con) {
        try {
```

```
        String create_specalization_table_query = "CREATE TABLE specalization(specializat
ion_name VARCHAR(22), slots_available INTEGER)";
        stmt.executeUpdate(create_specalization_table_query);
        System.out.println("specalization table created succesfully");

        String[] specalization_name = { "cloud", "AI", "big data", "gaming", "ui/ux" };
        int[] slots_available = { 22, 44, 26, 28, 24 };

        String specalization_insert_query = "INSERT INTO specalization (specalization_na
me, slots_available) VALUES (?,?)";

        PreparedStatement pstmt = con.prepareStatement(specalization_insert_query);

        for (int i = 0; i < specalization_name.length; i++) {
          pstmt.setString(1, specalization_name[i]);
          pstmt.setInt(2, slots_available[i]);
          pstmt.executeUpdate();
          System.out.println("Inserted " + specalization_name[i] + " data into table");
        }
        System.out.println("Inserted specalization data succesfully");
        print_specialization_table_data_to_console(stmt);
      } catch (Exception e) {
        System.out.println("Error: " + e);
      }
    }

    public static void print_student_table_data_to_console(Statement stmt) {
      try {
        // prints data to console
        ResultSet rs = stmt.executeQuery("select * from student_choice");
```

```
            if (!rs.isBeforeFirst()) {
                System.out.println("No data");
            } else {
                while (rs.next())
                    System.out.println(rs.getInt(1) + "  " + rs.getString(2));
            }

        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }

    public static void print_specialization_table_data_to_console(Statement stmt) {
        try {
            // prints data to console
            ResultSet rs = stmt.executeQuery("select * from specalization");
            if (!rs.isBeforeFirst()) {
                System.out.println("No data");
            } else {
                while (rs.next())
                    System.out.println(rs.getString(1) + "  " + rs.getInt(2));
            }
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }
}
```

Output Screenshots:

```
C:\study\3-1\jfsd_sdp\skill\skill-1> c: && cd c:\study\3-1\jfsd_sdp\skill\skill-1 && cmd /C "
ripts\launcher.bat "C:\Program Files\Java\jdk-15.0.1\bin\java.exe" -XX:+ShowCodeDetailsInExce
p\cp_9tf3pn8ctlcucy0nayivbbaxs.argfile chooseSpecalizationSkill1 "
specalization table created succesfully
Inserted cloud data into table
Inserted AI data into table
Inserted big data data into table
Inserted gaming data into table
Inserted ui/ux data into table
Inserted specalization data succesfully
cloud  22
AI  44
big data  26
gaming  28
ui/ux  24
student specalization choice table created succesfully
1. Select a specalization
2. View students choices
3. Exit the program
```

SQL Plus

```
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Wed Jul 28 2021 17:53:00 +05:30

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> select * from specalization;

SPECALIZATION_NAME     SLOTS_AVAILABLE
---------------------- ---------------
cloud                               21
AI                                  44
big data                            26
gaming                              27
ui/ux                               24

SQL> select * from student_choice;

STUDENT_ID SPECALIZATION_NAME
---------- ----------------------
 190031920 cloud
 190031927 gaming

SQL>
```

```
2
190031920  cloud
1. Select a specalization
2. View students choices
3. Exit the program
1
Please enter your id number
190031927
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        21
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
```

```
2. View students choices
3. Exit the program
2
No data
1. Select a specalization
2. View students choices
3. Exit the program
1
Please enter your id number
190031920
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        22
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
cloud
decreased available slots succesfully
inserted into Student database succesfully
Specalization selected successfully
1. Select a specalization
2. View students choices
3. Exit the program
2
190031920  cloud
1. Select a specalization
2. View students choices
3. Exit the program
```

```
1
Please enter your id number
190031927
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        21
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
gaming
decreased available slots succesfully
inserted into Student database succesfully
Specalization selected successfully
1. Select a specalization
2. View students choices
3. Exit the program
2
190031920  cloud
190031927  gaming
1. Select a specalization
2. View students choices
3. Exit the program
```

```
1. Select a specalization
2. View students choices
3. Exit the program
1
Please enter your id number
190031927
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        21
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
a
please select a valid specalization from the given list
1. Select a specalization
2. View students choices
3. Exit the program
```

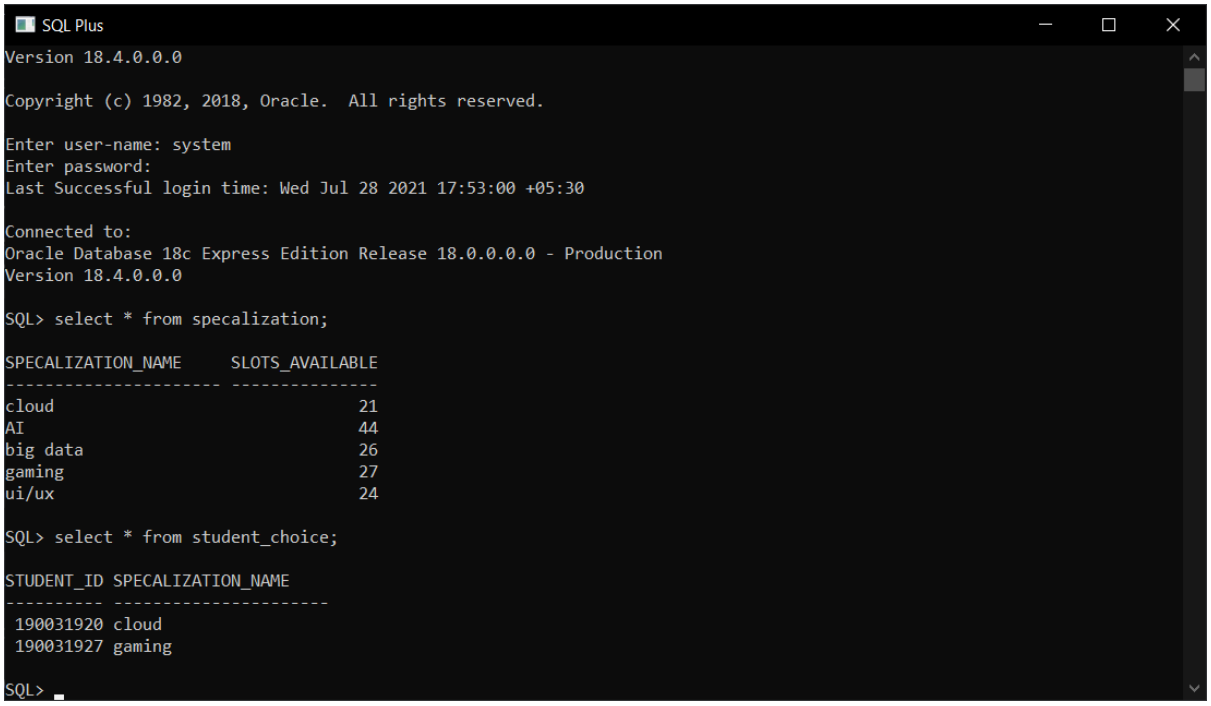
```
2
190031920  cloud
1. Select a specalization
2. View students choices
3. Exit the program
1
Please enter your id number
190031927
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        21
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
```

```
2. View students choices
3. Exit the program
2
No data
1. Select a specalization
2. View students choices
3. Exit the program
1
Please enter your id number
190031920
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        22
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
cloud
decreased available slots succesfully
inserted into Student database succesfully
Specalization selected successfully
1. Select a specalization
2. View students choices
3. Exit the program
2
190031920  cloud
1. Select a specalization
2. View students choices
3. Exit the program
```

```
1
Please enter your id number
190031927
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        21
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
gaming
decreased available slots succesfully
inserted into Student database succesfully
Specalization selected successfully
1. Select a specalization
2. View students choices
3. Exit the program
2
190031920  cloud
190031927  gaming
1. Select a specalization
2. View students choices
3. Exit the program
```

```
1. Select a specalization
2. View students choices
3. Exit the program
1
Please enter your id number
190031927
Please select a specalization from the given list
Available specalizations
Specalization name      |     Slots available
                  cloud|        21
                     AI|        44
               big data|        26
                 gaming|        28
                  ui/ux|        24
Please enter your preffered specalization name
a
please select a valid specalization from the given list
1. Select a specalization
2. View students choices
3. Exit the program
```

```
C:\study\3-1\jfsd_sdp\skill\skill-1> c: && cd c:\study\3-1\jfsd_sdp\skill\skill-1 && cmd /C "
ripts\launcher.bat "C:\Program Files\Java\jdk-15.0.1\bin\java.exe" -XX:+ShowCodeDetailsInExce
p\cp_9tf3pn8ctlcucy0nayivbbaxs.argfile chooseSpecalizationSkill1 "
specalization table created succesfully
Inserted cloud data into table
Inserted AI data into table
Inserted big data data into table
Inserted gaming data into table
Inserted ui/ux data into table
Inserted specalization data succesfully
cloud  22
AI  44
big data  26
gaming  28
ui/ux  24
student specalization choice table created succesfully
1. Select a specalization
2. View students choices
3. Exit the program
```



2. Given Question

We are entering into the 3rd year of our B. Tech; we all are aware that CRT (Campus Recruitment Training) starts from the sixth semester. Students are instructed to clear all the backlogs and try to increase their CGPA (Cumulative Grade Point Average) to get selected in CRT Programming. For that our Management wants to maintain the data in a database. Our Management wants you to create a table containing the details of students (Student ID Number, Student name, Gender, Year of Study, Department, K L Mail, CGPA, No of Backlogs present). Store at least 5 records in the database.

Complete code in java:

```
import java.sql.*;

import java.util.*;


class Skill1Q2 {

   public static void main(String[] args) throws Exception {

      Connection con = getConnection();


      Scanner scanner = new Scanner(System.in);
```

```
boolean exit = false;

create_students_table(con);

while (!exit) {
   System.out.println("1. Update students data");
   System.out.println("2. View qualified students");
   System.out.println("3. Delete non qualified students");
   System.out.println("4. View all data in students table");
   System.out.println("5. Exit the program");

   int choice = scanner.nextInt();

   switch (choice) {
      case 1:
         update_student_data(con, scanner);
         break;

      case 2:
         print_qualified_students_data(con);
         break;

      case 3:
         delete_unqualified_students_data(con);
         break;

      case 4:
         print_all_students_data(con);
         break;
```

```
            case 5:
                exit = true;
                break;

            default:
                System.out.println("Please Try Again");
                break;
          }

      }
      scanner.close();
      con.close();
   }

   public static Connection getConnection() throws ClassNotFoundException, SQLExcepti
on {

      Class.forName("oracle.jdbc.driver.OracleDriver");

      String dbUserName = "system";

      String dbUserPassword = "nikhil";

      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
XE", dbUserName, dbUserPassword);

      return con;

   }
```

```
public static void create_students_table(Connection con) {
    try {
        String create_students_table_query = "CREATE TABLE students (id INTEGER PRIM
ARY KEY, name VARCHAR(255) NOT NULL, gender VARCHAR(6), year_of_study INTEGER,
department VARCHAR(3), kl_mail VARCHAR(255),cgpa DOUBLE PRECISION, num_of_bac
klogs INTEGER)";

        Statement stmt = con.createStatement();

        stmt.executeUpdate(create_students_table_query);
        System.out.println("students table created succesfully");

        int[] id = { 1920, 1925, 1927, 1943, 1988 };
        String[] names = { "nikhil", "goutham", "kishan", "vyshnav", "charan" };
        String[] gender = { "male", "male", "male", "male", "male" };
        int[] year = { 3, 3, 3, 3, 3 };
        String[] department = { "cse", "cse", "cse", "cse", "cse" };
        String[] kl_mail = { "1920@kluniversity.in", "1925@kluniversity.in", "1927@kluniv
ersity.in",
            "1943@kluniversity.in", "1988@kluniversity.in" };
        double[] cgpa = { 8.00, 7.26, 7.80, 7.50, 6.20 };
        int[] num_of_backlogs = { 0, 0, 1, 0, 1 };

        String students_insert_query = "insert into students(id, name, gender, year_of_st
udy, department, kl_mail, cgpa, num_of_backlogs) values(?, ?, ?, ?, ?, ?, ?, ?)";

        PreparedStatement prepStmt = con.prepareStatement(students_insert_query);

        for (int i = 0; i < id.length; i++) {
            prepStmt.setInt(1, id[i]);
```

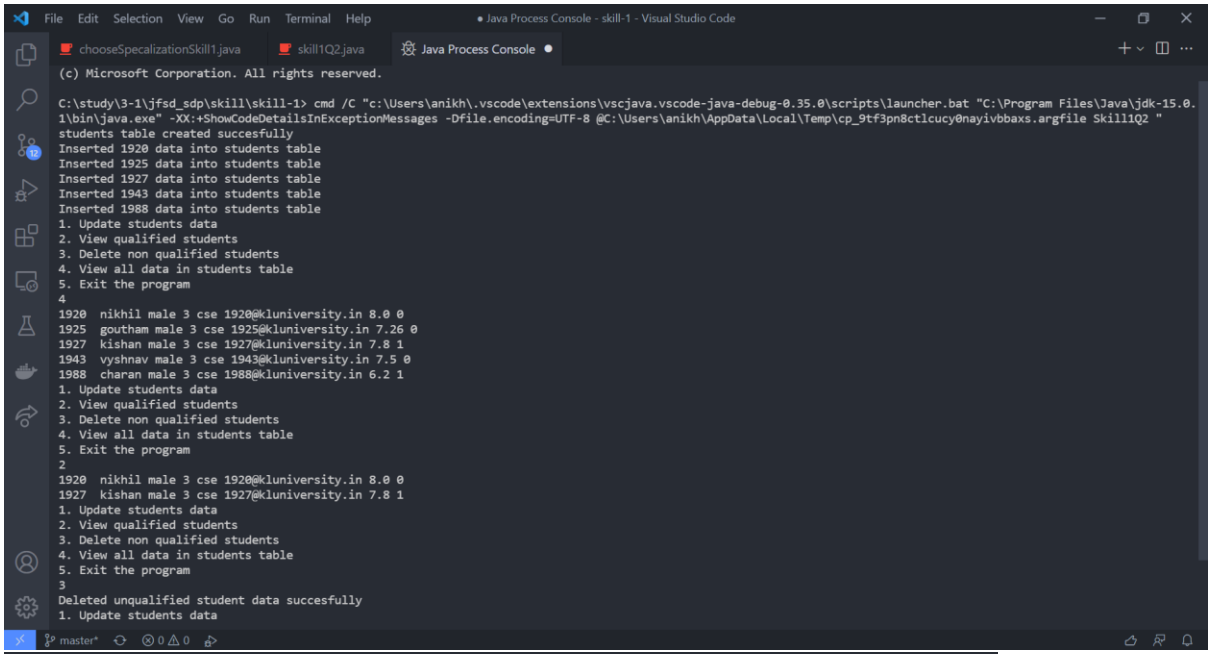
```
            prepStmt.setString(2, names[i]);
            prepStmt.setString(3, gender[i]);
            prepStmt.setInt(4, year[i]);
            prepStmt.setString(5, department[i]);
            prepStmt.setString(6, kl_mail[i]);
            prepStmt.setDouble(7, cgpa[i]);
            prepStmt.setInt(8, num_of_backlogs[i]);
            prepStmt.executeUpdate();
            System.out.println("Inserted " + id[i] + " data into students table");
         }

      } catch (Exception e) {
         System.out.println("Error: " + e);
      }
   }

   public static void print_qualified_students_data(Connection con) {
      try {
         Statement stmt = con.createStatement();
         ResultSet rs = stmt.executeQuery("SELECT * FROM students where cgpa > 7.5");
         if (!rs.isBeforeFirst()) {
            System.out.println("No data");
         } else {
            while (rs.next())
               System.out.println(rs.getInt(1) + "  " + rs.getString(2) + " " + rs.getString(3) + "
"
                     + rs.getInt(4) + " " + rs.getString(5) + " " + rs.getString(6) + " " + rs.getDou
ble(7) + " "
                     + rs.getInt(8));
         }
```

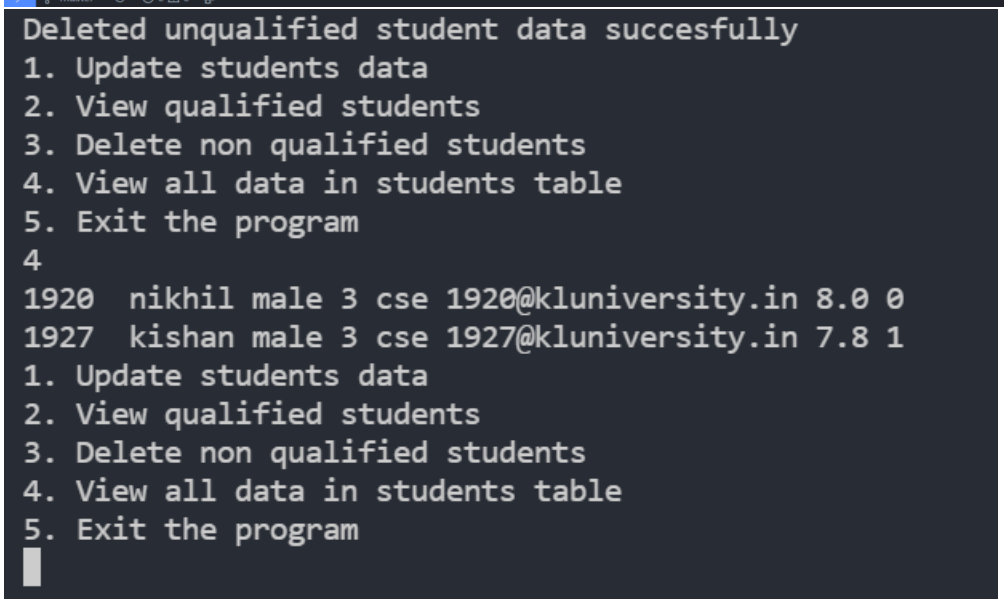
```
        } catch (Exception e) {


        }
    }

    public static void print_all_students_data(Connection con) {
        try {
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM students");
            if (!rs.isBeforeFirst()) {
                System.out.println("No data");
            } else {
                while (rs.next())
                    System.out.println(rs.getInt(1) + "  " + rs.getString(2) + " " + rs.getString(3) + "
"
                            + rs.getInt(4) + " " + rs.getString(5) + " " + rs.getString(6) + " " + rs.getDou
ble(7) + " "
                            + rs.getInt(8));
            }
        } catch (Exception e) {


        }
    }


    public static void update_student_data(Connection con, Scanner scanner) {
        try {
            System.out.println("Enter the id of the student");
            int id = scanner.nextInt();
            System.out.println("Enter new number of backlogs");
            int num_of_backlogs = scanner.nextInt();
```

```
            System.out.println("Enter new cgpa");
            double cgpa = scanner.nextDouble();

            String students_table_update_query = "UPDATE students SET cgpa = ?, num_of_backlogs = ? WHERE id = ?";
            PreparedStatement pstmt = con.prepareStatement(students_table_update_query);
            pstmt.setDouble(1, cgpa);
            pstmt.setInt(2, num_of_backlogs);
            pstmt.setInt(3, id);
            pstmt.executeUpdate();
            System.out.println("updated student data");
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }

    }

    public static void delete_unqualified_students_data(Connection con) {
        try {
            String delete_unqualified_students_query = "DELETE FROM students WHERE cgpa <= 7.5";

            Statement stmt = con.createStatement();

            stmt.executeUpdate(delete_unqualified_students_query);

            System.out.println("Deleted unqualified student data succesfully");
        } catch (Exception e) {
            System.out.println("Error: " + e);
```

```
        }
    }
}
```

Output Screenshots:





```
C:\study\3-1\jfsd_sdp\skill\skill-1> cmd /C "c:\Users\anikh\.vscode\extensions\vscjava.vscode-java-debug-0.35.0\scripts\launch
1\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 @C:\Users\anikh\AppData\Local\Temp\cp_9tf3pn8ctl
students table created succesfully
Inserted 1920 data into students table
Inserted 1925 data into students table
Inserted 1927 data into students table
Inserted 1943 data into students table
Inserted 1988 data into students table
1. Update students data
2. View qualified students
3. Delete non qualified students
4. View all data in students table
5. Exit the program
2
1920  nikhil male 3 cse 1920@kluniversity.in 8.0 0
1927  kishan male 3 cse 1927@kluniversity.in 7.8 1
1. Update students data
2. View qualified students
3. Delete non qualified students
4. View all data in students table
5. Exit the program
1
Enter the id of the student
1943
Enter new number of backlogs
0
Enter new cgpa
7.6
updated student data
1. Update students data
```

```
1. Update students data
2. View qualified students
3. Delete non qualified students
4. View all data in students table
5. Exit the program
2
1920  nikhil male 3 cse 1920@kluniversity.in 8.0 0
1927  kishan male 3 cse 1927@kluniversity.in 7.8 1
1943  vyshnav male 3 cse 1943@kluniversity.in 7.6 0
1. Update students data
2. View qualified students
3. Delete non qualified students
4. View all data in students table
5. Exit the program
3
Deleted unqualified student data succesfully
1. Update students data
2. View qualified students
3. Delete non qualified students
4. View all data in students table
5. Exit the program
4
1920  nikhil male 3 cse 1920@kluniversity.in 8.0 0
1927  kishan male 3 cse 1927@kluniversity.in 7.8 1
1943  vyshnav male 3 cse 1943@kluniversity.in 7.6 0
1. Update students data
2. View qualified students
3. Delete non qualified students
4. View all data in students table
5. Exit the program
```