You're assigned to develop a Student Data Management System for a college using JDBC and MySQL. The application should support CRUD (Create, Read, Update, Delete) operations on student data stored in a database. The system should be able to manage student details such as student ID, name, department, year of study, and grade.

Table Name: `student`

Table Structure:

- `student\_id`: INT (Primary Key)

- `student\_name`: VARCHAR(50) (NOT NULL)

- `department`: VARCHAR(50)

- `year\_of\_study`: INT

- `grade`: VARCHAR(2)

- `dateofbirth` : date

Operations:

1. Insert Student Details: Add a new student record with all the details including student\_id, student\_name,department, year\_of\_study, and grade. Do not allow duplicate student id. If duplicate student id, display “Student details with [Student\_id] already exist. Do not enter the data if year of study not greater then 1900.

2. Update Student Details: Update the department, year\_of\_study, and grade for a student based on their student\_id. If student\_Id does not exist display “Student with [Student\_ID] not found”.

3. Update grade : Update grade for all students based on given department and year of study except year of study 2000. If not record found display “No students found”.

4. Delete Student Details: Remove a student record based on their student\_id. If student\_id not found, display “Student with [Student\_ID] not found.

5. Delete student details based on year\_of\_study and grade

4. Show All Student Details: Display all student details, sorted by department in ascending order.

NOTE:

- U􀆟lize `try/catch` and `finally` blocks to handle `SQLExcep􀆟on`.

Compiled by Rohit Ahuja (9893075987)

- The table name is case-sensi􀆟ve and must match the one specified above.

- The table is already created and contains some records.

- `student\_id` should be between 101 and 105.

Input Format:

The input consists of a number represen􀆟ng the CRUD opera􀆟on:

1 - Insert (followed by all a􀆩ributes for the new student)

2 - Update (update department, year\_of\_study, and grade based on `student\_id`)

3 - Delete (remove a student based on `student\_id`)

4 - Show all details (display all exis􀆟ng student records)

Output Format:

Based on the opera􀆟on selected, display appropriate messages such as "Student added successfully," "Student updated

successfully," "Student deleted successfully," followed by the updated list of student records.

Sample Test Cases:

- Test Case 1:

Input:

1

106

Alice Johnson

Computer Science

2

A

Output:

Student added successfully.

Student ID: 101, Student Name: John Doe, Department: Computer Science, Year of Study: 3, Grade: B

Student ID: 102, Student Name: Jane Smith, Department: Mechanical, Year of Study: 2, Grade: A

Student ID: 103, Student Name: Mike Davis, Department: Electronics, Year of Study: 1, Grade: B

Student ID: 106, Student Name: Alice Johnson, Department: Computer Science, Year of Study: 2, Grade: A

- Test Case 2:

Input:

2

101

Electronics

2004

A

Output:

Student updated successfully.

Student ID: 101, Student Name: John Doe, Department: Electronics, Year of Study: 4, Grade: A

- Test Case 3:

Input:

3

101

Output:

Student deleted successfully.

- Test Case 4:

Input:

4

Output:

Student ID: 102, Student Name: Jane Smith, Department: Mechanical, Year of Study: 2, Grade: A

CREATE TABLE student(

studentid int primary key,

studentname varchar(50) not null,

department varchar(50),

yearofstudy int,

grade varchar(2),

dateofbirth date

);

INSERT INTO student VALUES(101, 'John Doe', 'Computer Science', 2005, 'B','1980-11-03');

INSERT INTO student VALUES(102, 'Jane Smith', 'Mechanical', 1999, 'A','2000-01-03');

INSERT INTO student VALUES(103, 'Mike Davis', 'Electronics', 2006, 'B','1980-05-03');