



Dr. Öğr. Üyesi ROA'A ALI
ABDULLAH MOHAMMEDQASEM

20.05.2025

Database Systems

UNITRACK

Esra PALA

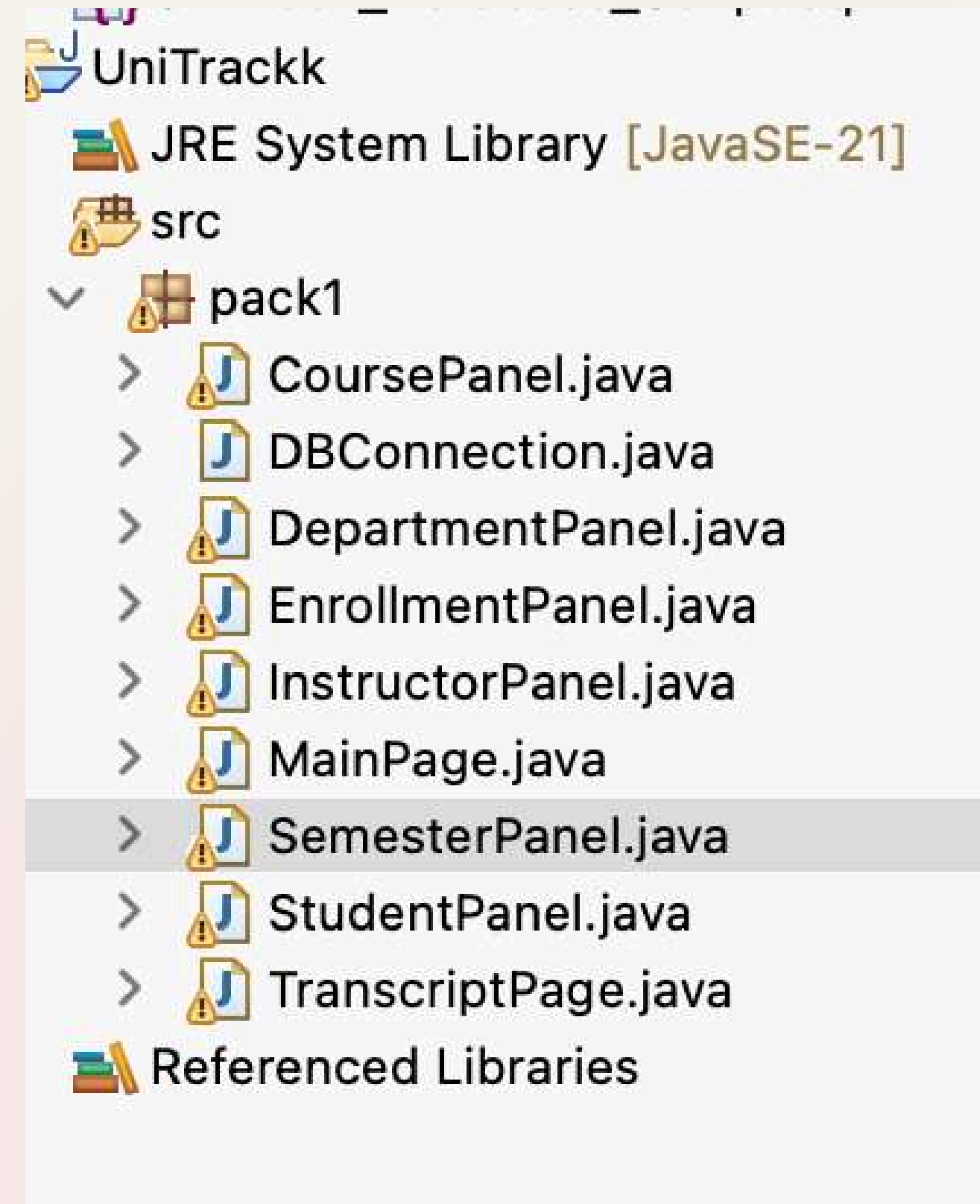
B2205.010134

Abdallah Dwikat

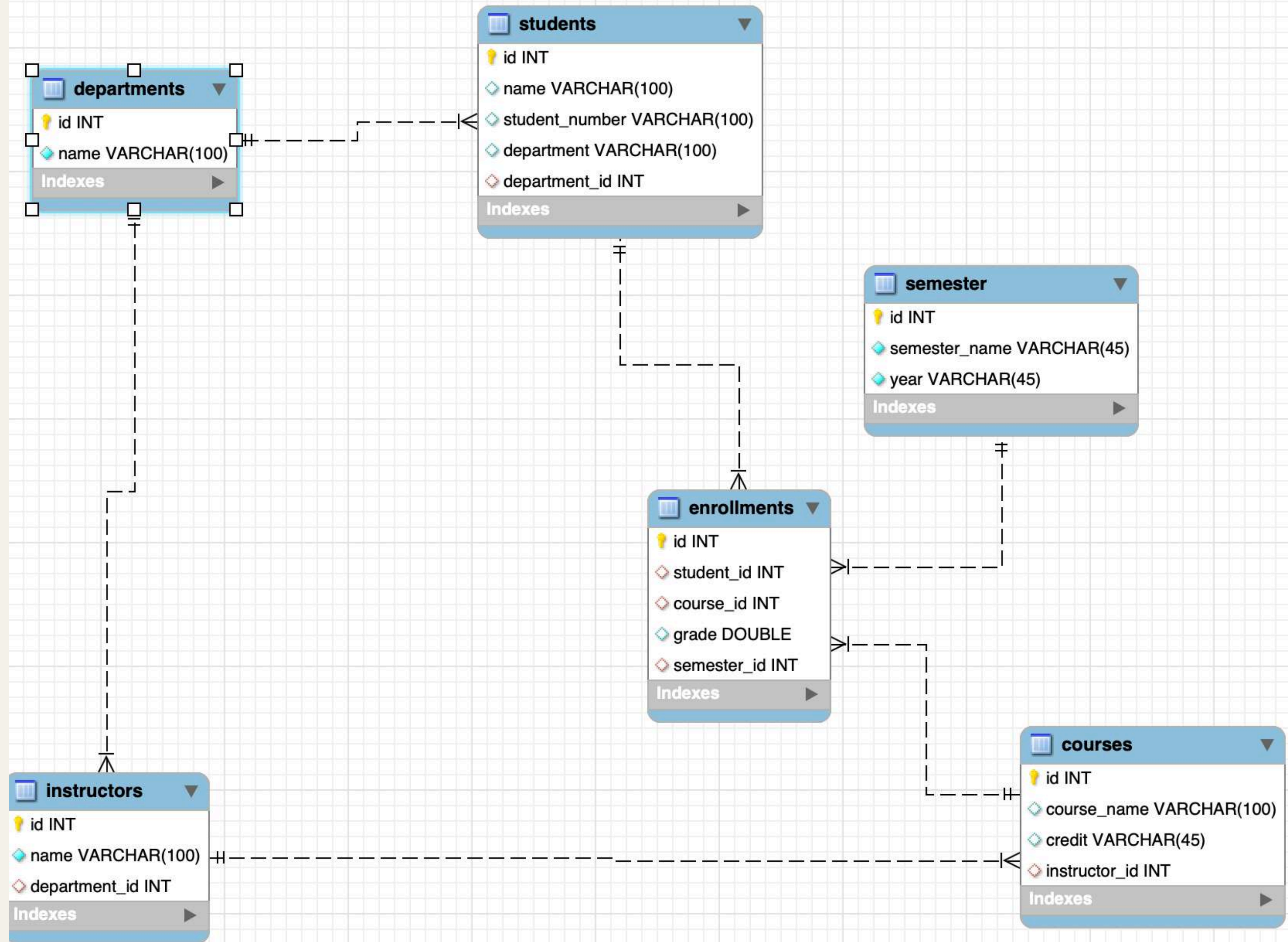
B2205.010045

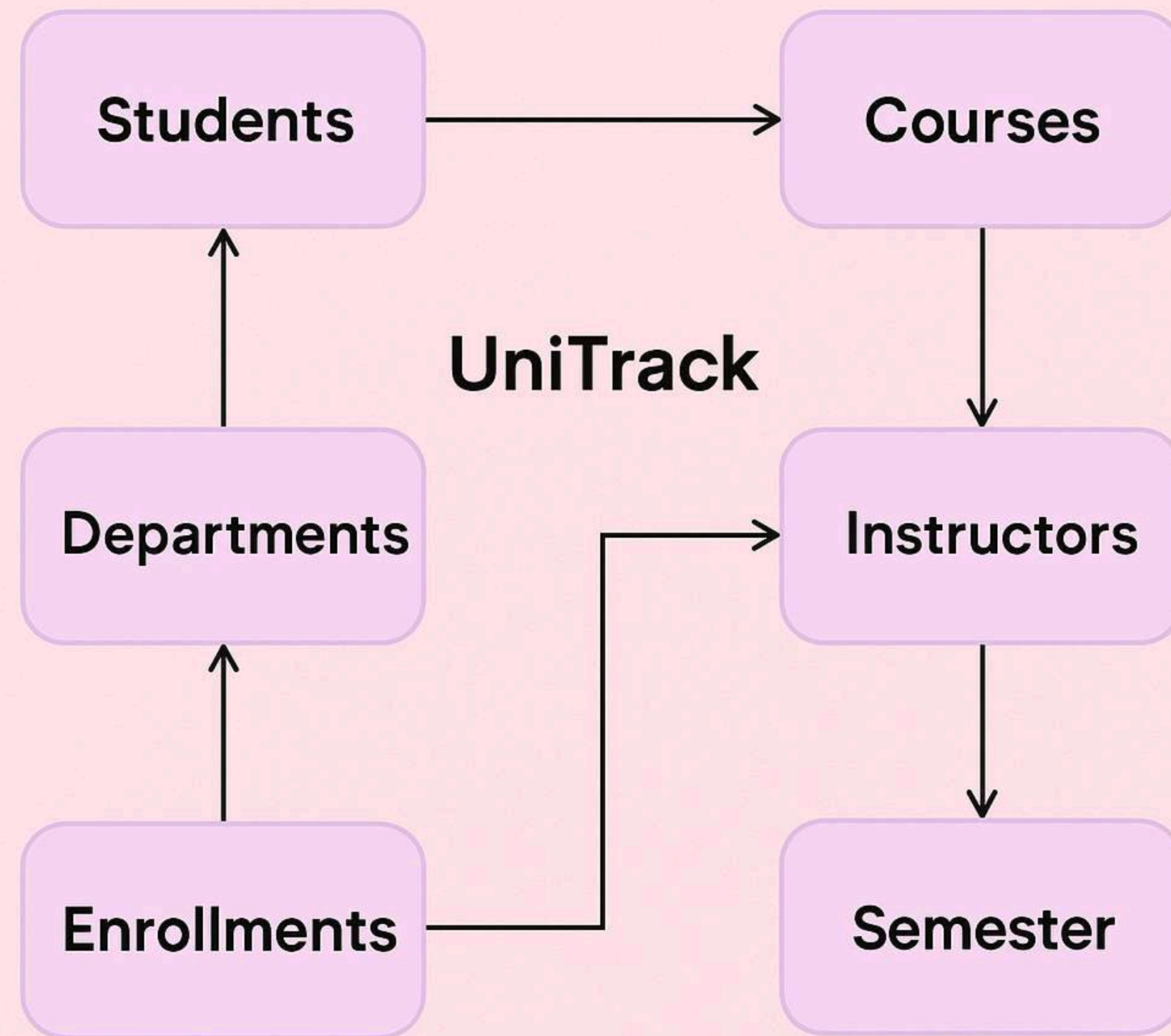
PROJECT OVERVIEW

- UniTrack is a student information system built using Java Swing, MySQL, and JDBC. It allows managing students, courses, instructors, semesters, and enrollments in a user-friendly GUI.



ER Diagram





The UniTrack system is built upon a relational database structure that includes six core tables: students, courses, instructors, departments, semester, and enrollments. These tables are connected through foreign key constraints to ensure data integrity and prevent inconsistency.

- students – Stores each student's name, student number, and the department they belong to.
- courses – Contains course name, credit, and the instructor responsible for that course.
- instructors – Lists instructors and links them to their respective departments.
- departments – Represents all academic departments in the university.
- semester – Contains the semester name (e.g., Fall 2025) and academic year.
- enrollments – A junction table linking students, courses, and semesters, along with the student's grade.

Foreign Key Relationships:

- Each key table is connected via foreign keys:
- students.department_id → departments.id
- courses.instructor_id → instructors.id
- enrollments.student_id → students.id
- enrollments.course_id → courses.id
- enrollments.semester_id → semester.id

These ensure:

- Students can only be assigned to valid departments
- Courses are always linked to existing instructors
- Enrollments can only happen for valid students, courses, and semesters

Why Foreign Keys Matter:

- They enforce referential integrity (no “orphan” records)
- They prevent accidental deletion or insertion of unrelated data
- They allow for efficient JOIN queries, such as generating transcripts or full academic reports

DBCONNECTION

This class is responsible for establishing a connection between the Java application and the MySQL database using JDBC.

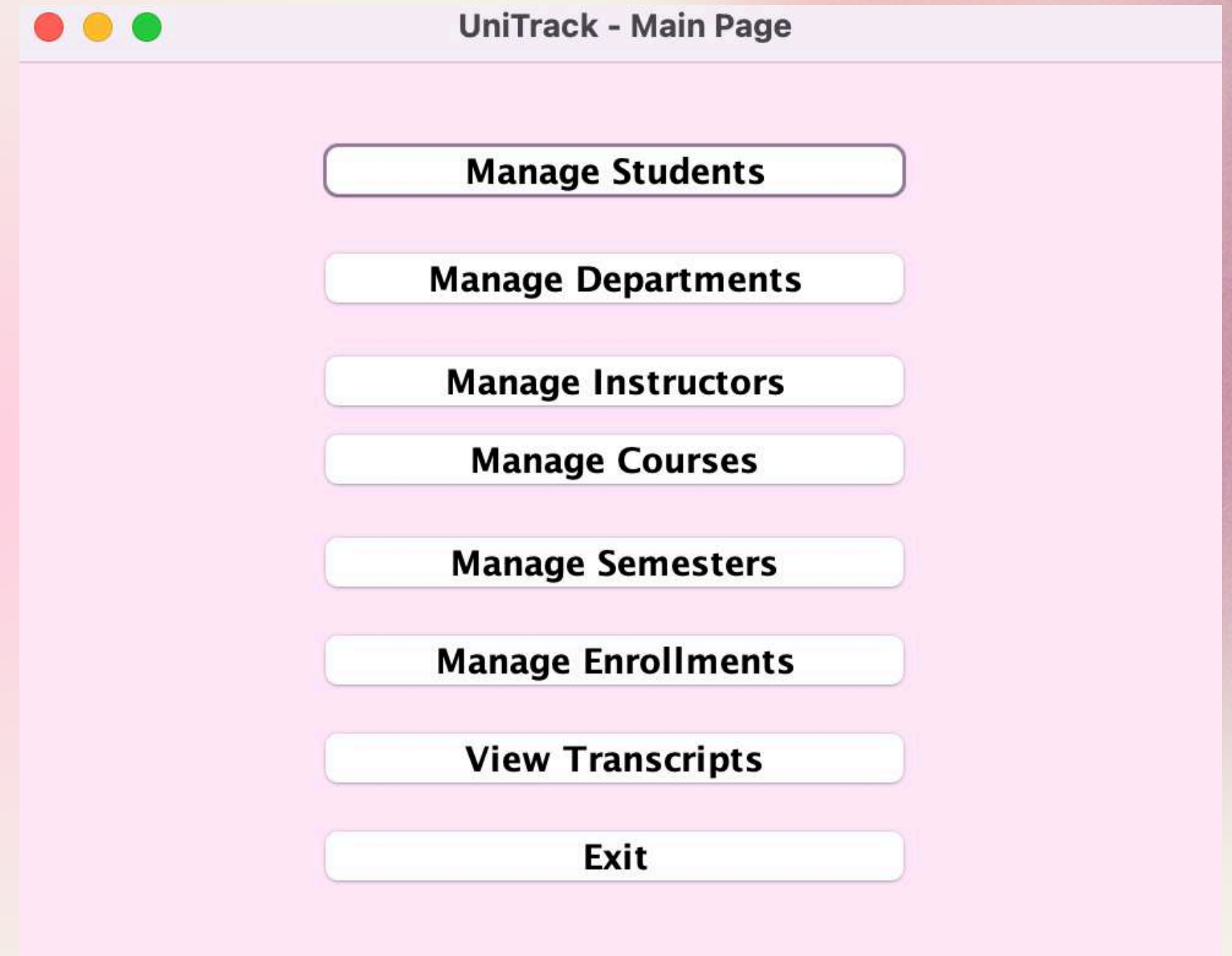
```
package pack1;

import java.sql.Connection;
import java.sql.DriverManager;

public class DBConnection {
    public static Connection connect() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            return DriverManager.getConnection("jdbc:mysql://localhost:3306/unitrack", "root", "watchmeshine");
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

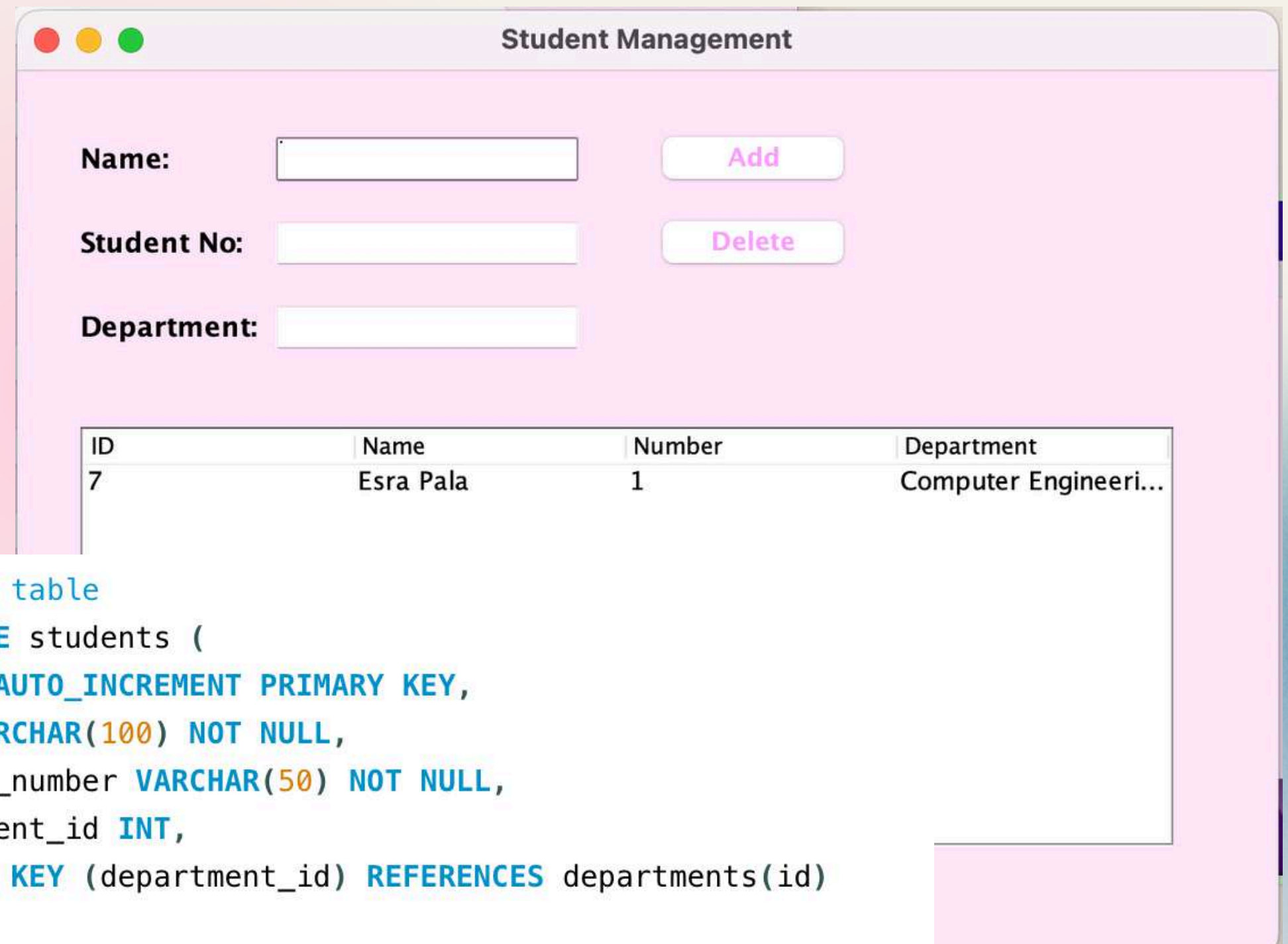

MainPage

Acts as the navigation hub of the application. Provides buttons to access each management panel with a soft pink theme.



StudentPanel

Allows adding and deleting students. Includes a JTable to display student data and uses PreparedStatement to insert/delete records.



The screenshot shows a Java Swing window titled "Student Management". It contains three input fields: "Name:", "Student No:", and "Department:". To the right of the "Name:" field is a button labeled "Add". To the right of the "Student No:" field is a button labeled "Delete". Below the input fields is a JTable with the following data:

| ID | Name | Number | Department |
|----|-----------|--------|-----------------------|
| 7 | Esra Pala | 1 | Computer Engineeri... |

Below the JTable is a text area containing SQL code for creating the students table:

```
-- Students table
CREATE TABLE students (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  student_number VARCHAR(50) NOT NULL,
  department_id INT,
  FOREIGN KEY (department_id) REFERENCES departments(id)
);
```


Instructor & Department Panels

Each panel allows managing independent data. Instructors are linked to departments. Deletion checks for foreign key constraints.

The image shows two overlapping web panels. The top panel, titled 'Department Management', has a form for 'Department Name' with an 'Add' button and a 'Delete' button. Below the form is a table with two columns: 'ID' and 'Name'. The table contains two rows: ID 6, Computer Engineering and ID 7, Software Engineering. The bottom panel, titled 'Instructor Management', has a form for 'Instructor Name' and 'Department' (a dropdown menu showing 'Computer En...'). It also has 'Add' and 'Delete' buttons. Below the form is a table with three columns: 'ID', 'Name', and 'Department'. The table contains two rows: ID 4, Roa, Computer Engineering and ID 5, Selçuk Şener, Software Engineering.

| ID | Name |
|----|----------------------|
| 6 | Computer Engineering |
| 7 | Software Engineering |

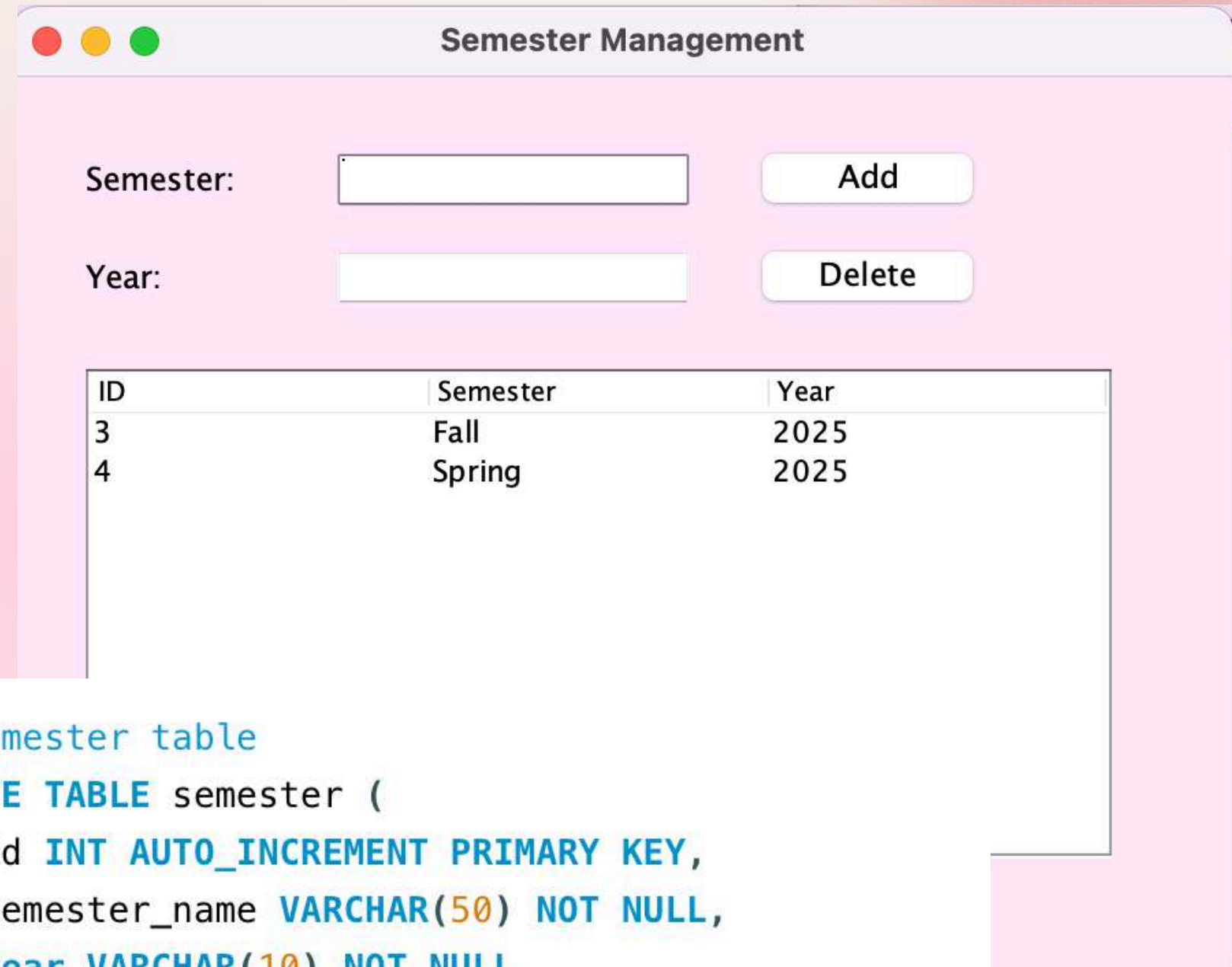
| ID | Name | Department |
|----|--------------|----------------------|
| 4 | Roa | Computer Engineering |
| 5 | Selçuk Şener | Software Engineering |

```
-- Departments table
CREATE TABLE departments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  department_name VARCHAR(100) NOT NULL
);

-- Instructors table
CREATE TABLE instructors (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  department_id INT,
  FOREIGN KEY (department_id) REFERENCES departments(id)
);
```

SemesterPanel

Stores semester name and year.
Provides error handling for incorrect SQL column usage.
Table reloads dynamically.



The screenshot shows a window titled "Semester Management" with a light pink background. It features two input fields: "Semester:" and "Year:", each followed by a text box and a button ("Add" and "Delete" respectively). Below these is a table with three columns: "ID", "Semester", and "Year". The table contains two rows of data.

| ID | Semester | Year |
|----|----------|------|
| 3 | Fall | 2025 |
| 4 | Spring | 2025 |

```
-- Semester table
• CREATE TABLE semester (
  id INT AUTO_INCREMENT PRIMARY KEY,
  semester_name VARCHAR(50) NOT NULL,
  year VARCHAR(10) NOT NULL
);
```


Enrollmen?Panel

Allows selecting student, course, semester, and entering a grade.
Handles foreign key relations and prevents ID mismatch errors.

The screenshot shows a macOS-style window titled "Enrollments". It features four dropdown menus for "Student", "Course", and "Semester", and a text input for "Grade". The "Student" dropdown is set to "Esra Pala", "Course" to "Database Systems", and "Semester" to "Fall 2025". To the right of these fields are "Add" and "Delete" buttons. Below the form is a table with the following data:

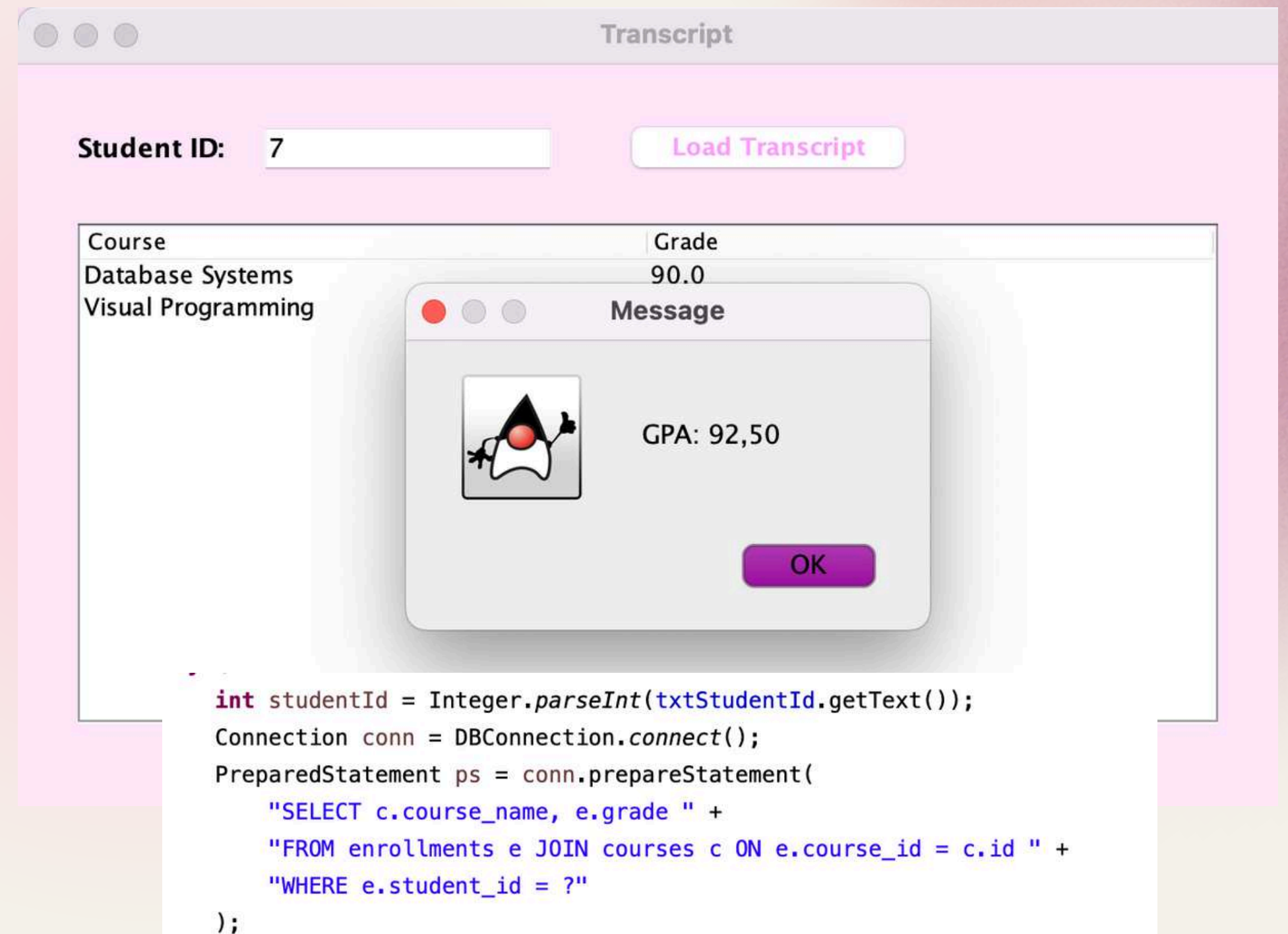
| ID | Student | Course | Semester | Grade |
|----|-----------|--------------------|-----------|-------|
| 3 | Esra Pala | Database Systems | Fall 2025 | 90.0 |
| 4 | Esra Pala | Visual Programming | Fall 2025 | 95.0 |

Below the table is a code editor showing the SQL schema for the "enrollments" table:

```
-- Enrollments table
CREATE TABLE enrollments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  student_id INT,
  course_id INT,
  semester_id INT,
  grade VARCHAR(5),
  FOREIGN KEY (student_id) REFERENCES students(id),
  FOREIGN KEY (course_id) REFERENCES courses(id),
  FOREIGN KEY (semester_id) REFERENCES semester(id)
);
```

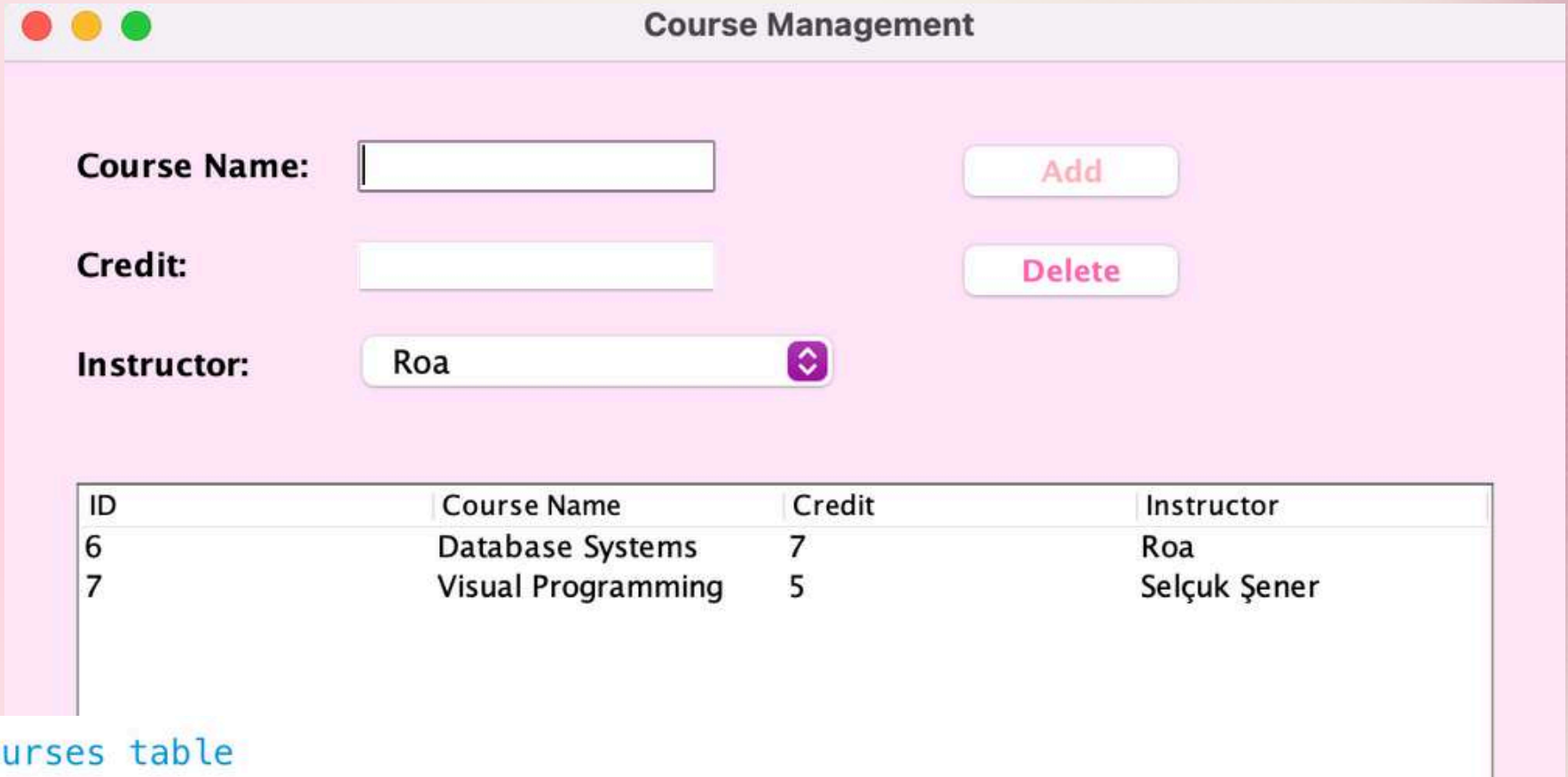

TranscripPanel

Displays a student's full academic history using JOIN queries across multiple tables. Read-only table view.



CoursePanel

This panel allows users to manage course records by adding new courses, assigning them to instructors, and deleting existing ones.



The interface is titled "Course Management" and features a form with three input fields: "Course Name:", "Credit:", and "Instructor:". The "Instructor:" field is a dropdown menu currently showing "Roa". To the right of the form are two buttons: "Add" and "Delete". Below the form is a table with the following data:

| ID | Course Name | Credit | Instructor |
|----|--------------------|--------|--------------|
| 6 | Database Systems | 7 | Roa |
| 7 | Visual Programming | 5 | Selçuk Şener |

```
-- Courses table
CREATE TABLE courses (
  id INT AUTO_INCREMENT PRIMARY KEY,
  course_name VARCHAR(100) NOT NULL,
  credit INT,
  instructor_id INT,
  FOREIGN KEY (instructor_id) REFERENCES instructors(id)
);
```


CONCLUSION

UniTrack demonstrates a modular, interactive and visually clean information system. It uses prepared statements, foreign key management, and is easily extendable.