Data Base Management System

Student Record Management

Why DBMS:

In Student Database, we have many records and contain large data which cannot be efficiently managed with File Processing System

DBMS serves as a critical tool for efficiently managing and manipulating large volumes of data in various applications and industries.

Introduction

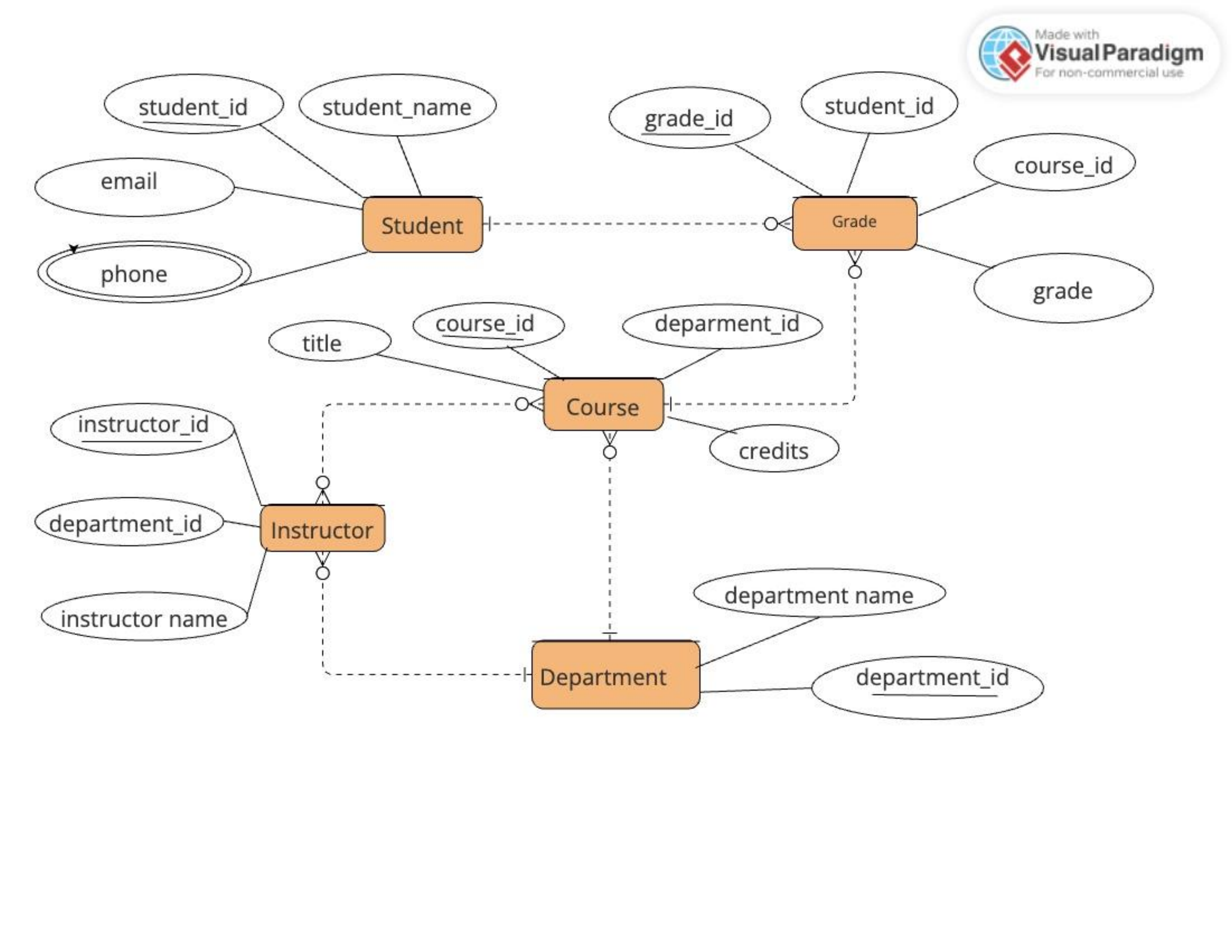
This is the project that is based on mySQL language and with DBMS tools.

This project enhances the efficiency, transparency, and effectiveness of student record management within educational institutions.

Educational Institutions have different entities like student, grade, course, instructor, department.

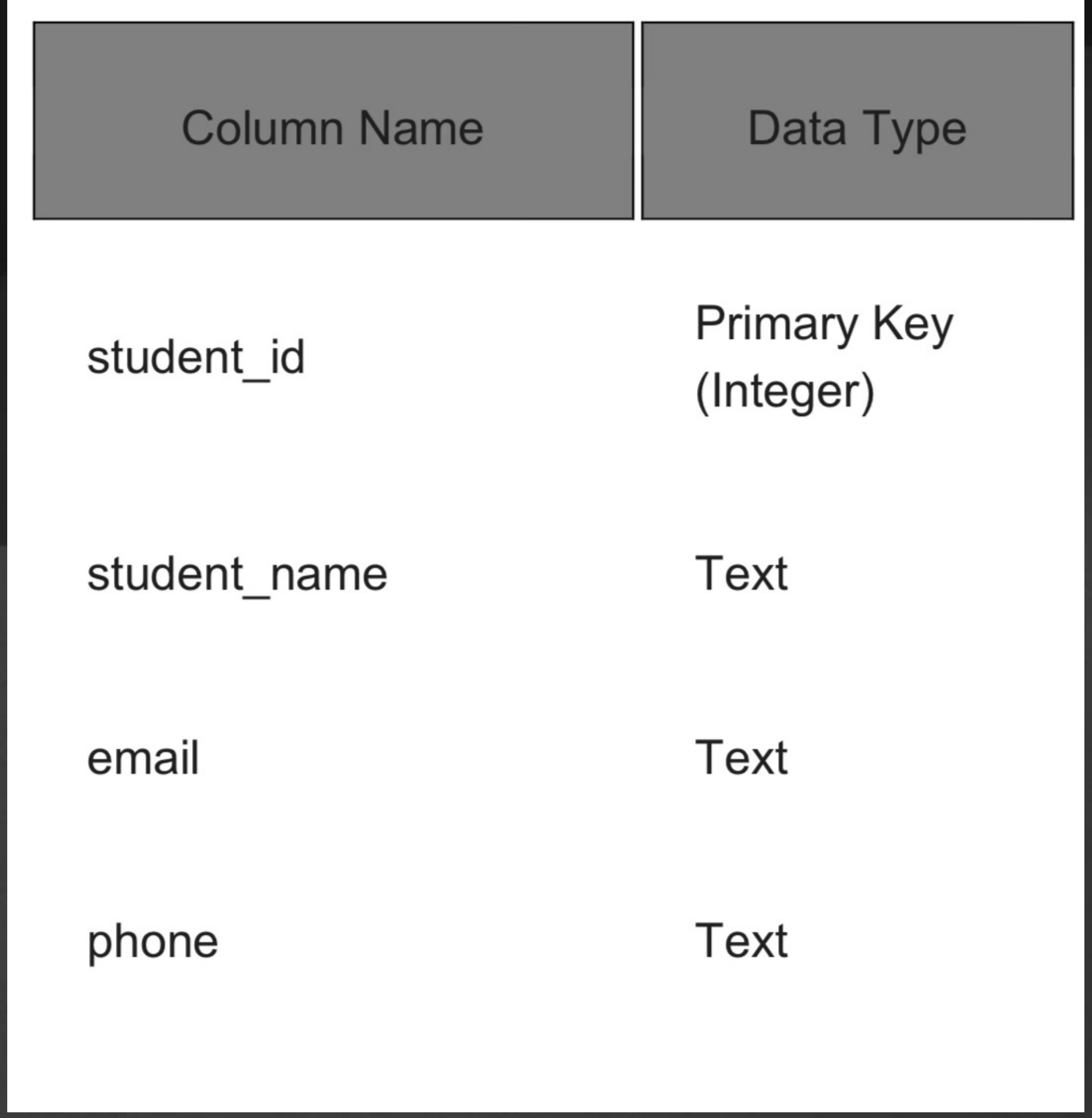
We use some steps to make database to work efficiently.

Entity Relation Diagram



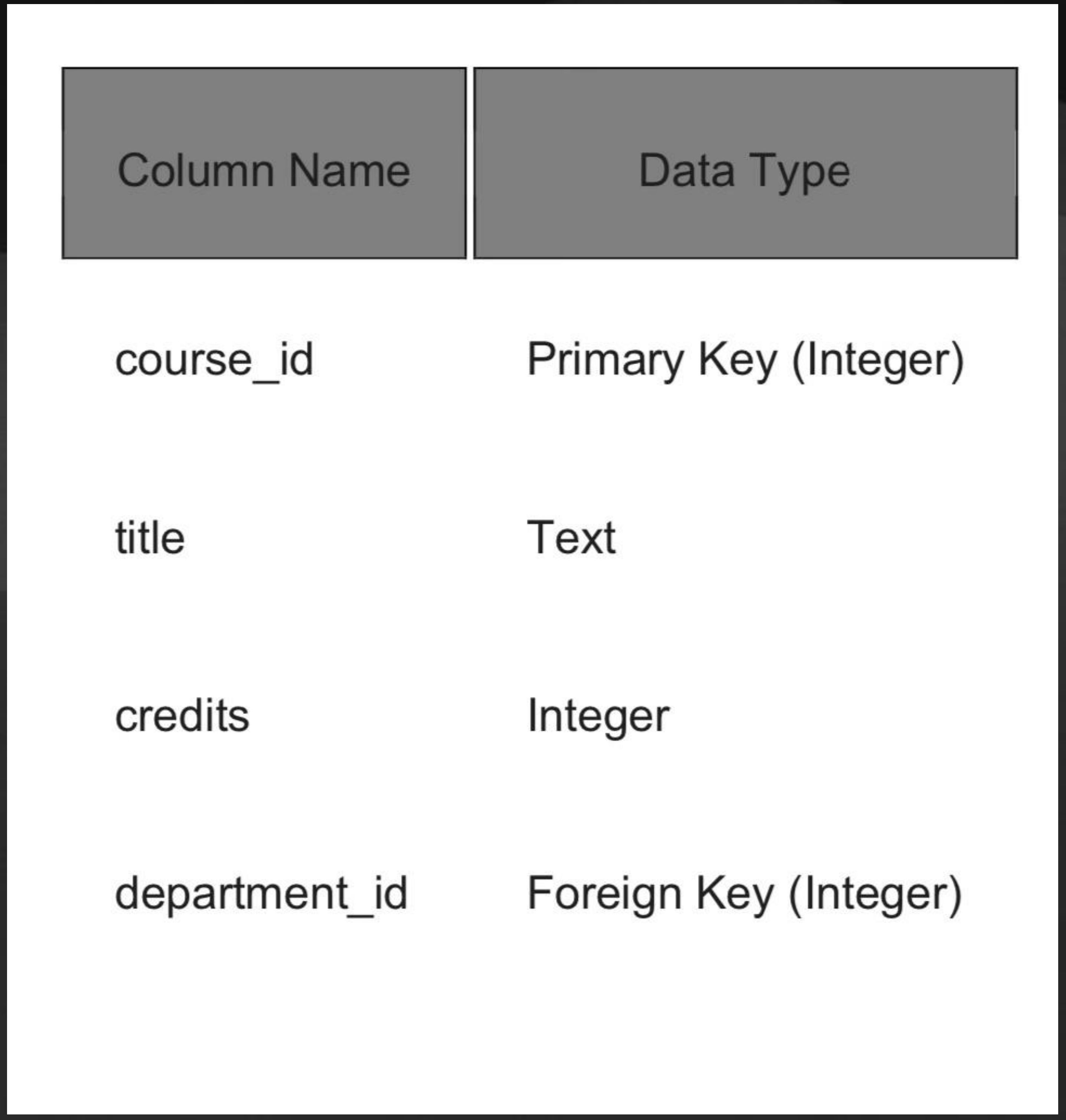
Tables

Student Table:



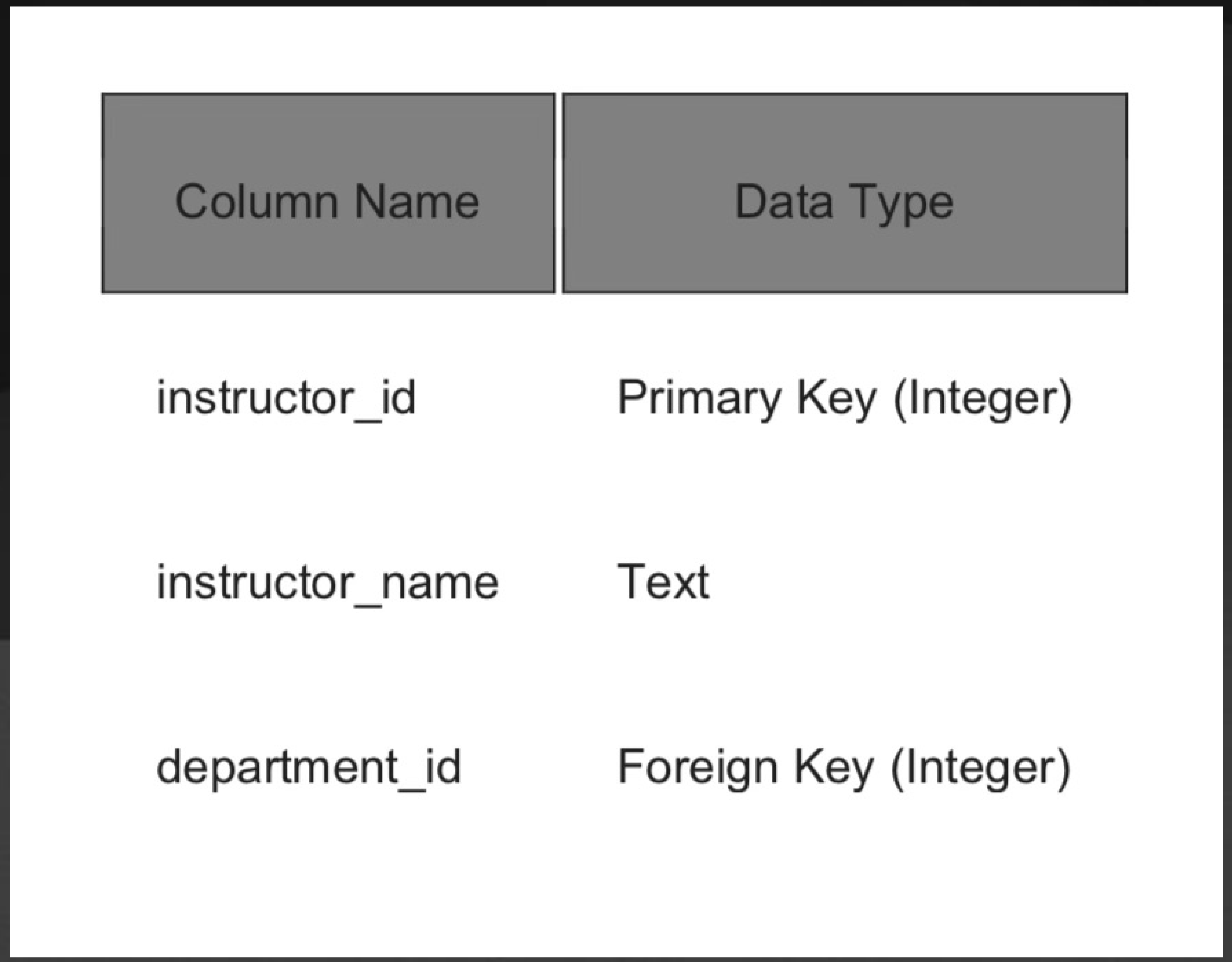
Column Name	Data Type
student_id	Primary Key (Integer)
student_name	Text
email	Text
phone	Text

Course Table:



Column Name	Data Type
course_id	Primary Key (Integer)
title	Text
credits	Integer
department_id	Foreign Key (Integer)

Instructor Table:



Column Name	Data Type
instructor_id	Primary Key (Integer)
instructor_name	Text
department_id	Foreign Key (Integer)

Department Table:

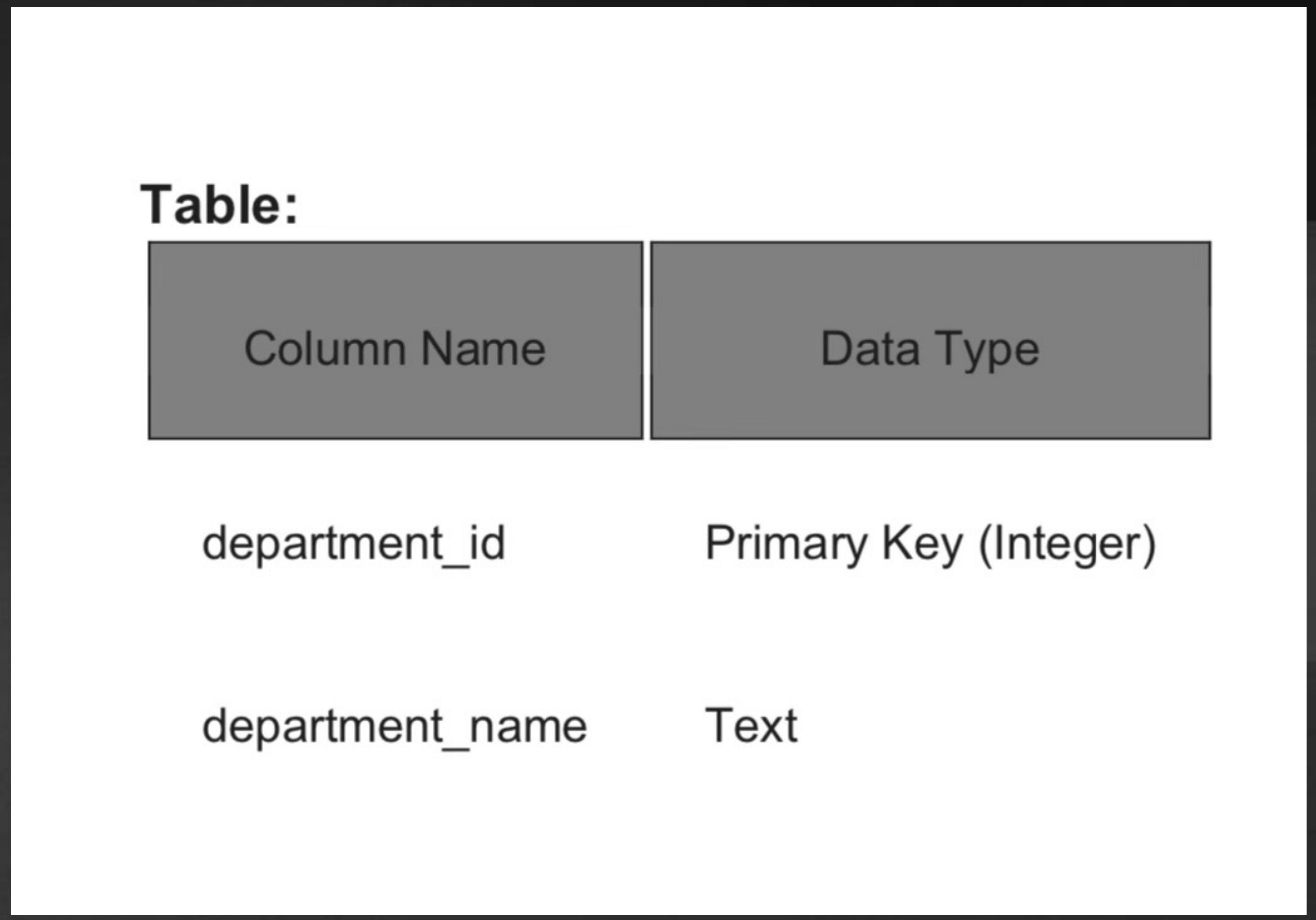
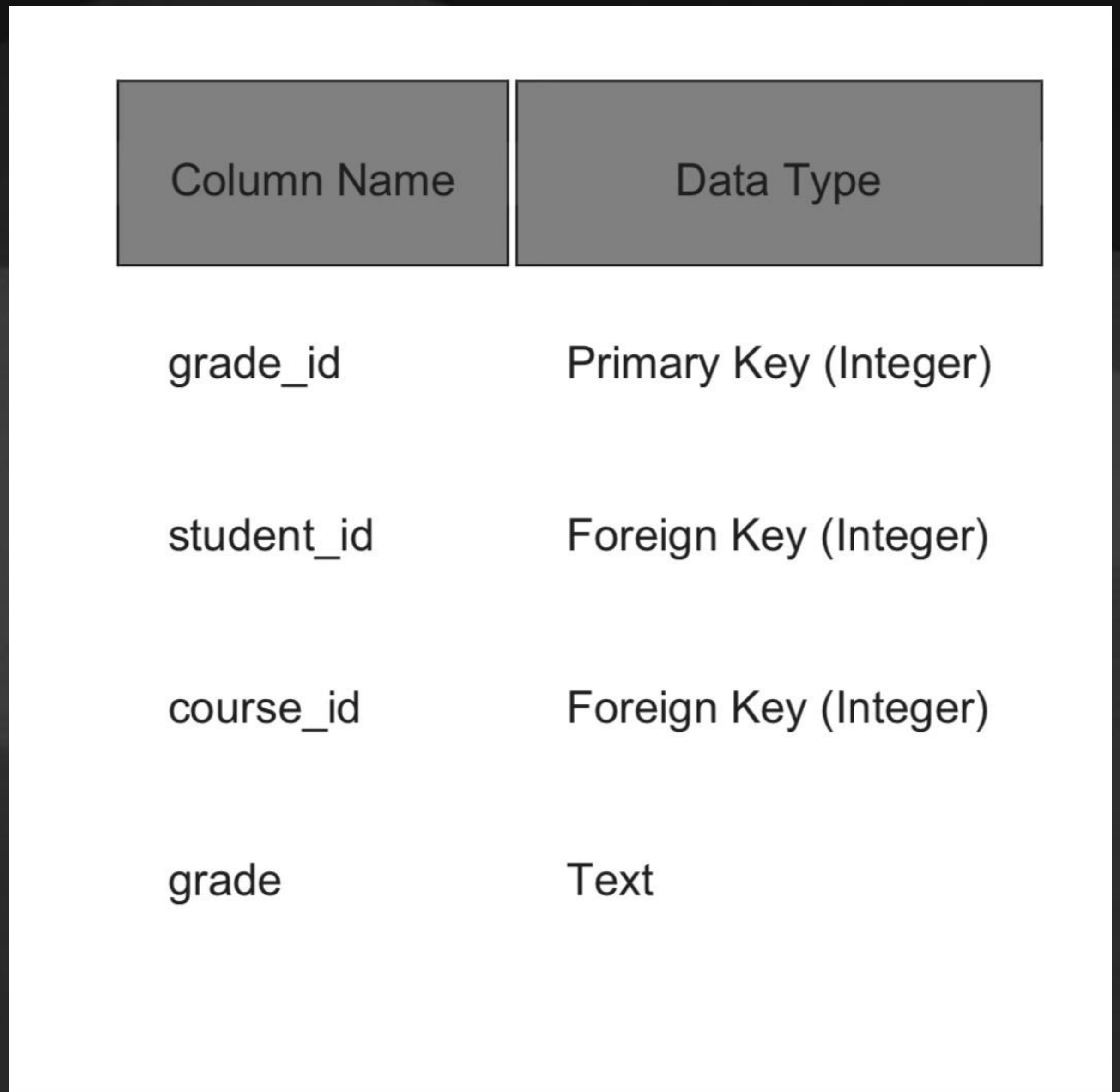


Table:

Column Name	Data Type
department_id	Primary Key (Integer)
department_name	Text

Grade Table:



Column Name	Data Type
grade_id	Primary Key (Integer)
student_id	Foreign Key (Integer)
course_id	Foreign Key (Integer)
grade	Text

Normalizing Tables

The main objective of database normalization is to eliminate redundant data, minimize data modification errors, and simplify the query process.

3NF: Each table has a unique primary key, Each non-key attribute is fully functionally dependent on the entire primary key, There are no transitive dependencies.

Student Table : Already in 3NF

Course Table : Already in 3NF

Department Table : Already in 3NF

Instructor Table : Already in 3NF

Grade Table : Not in 3NF

Normalizing Grade Table

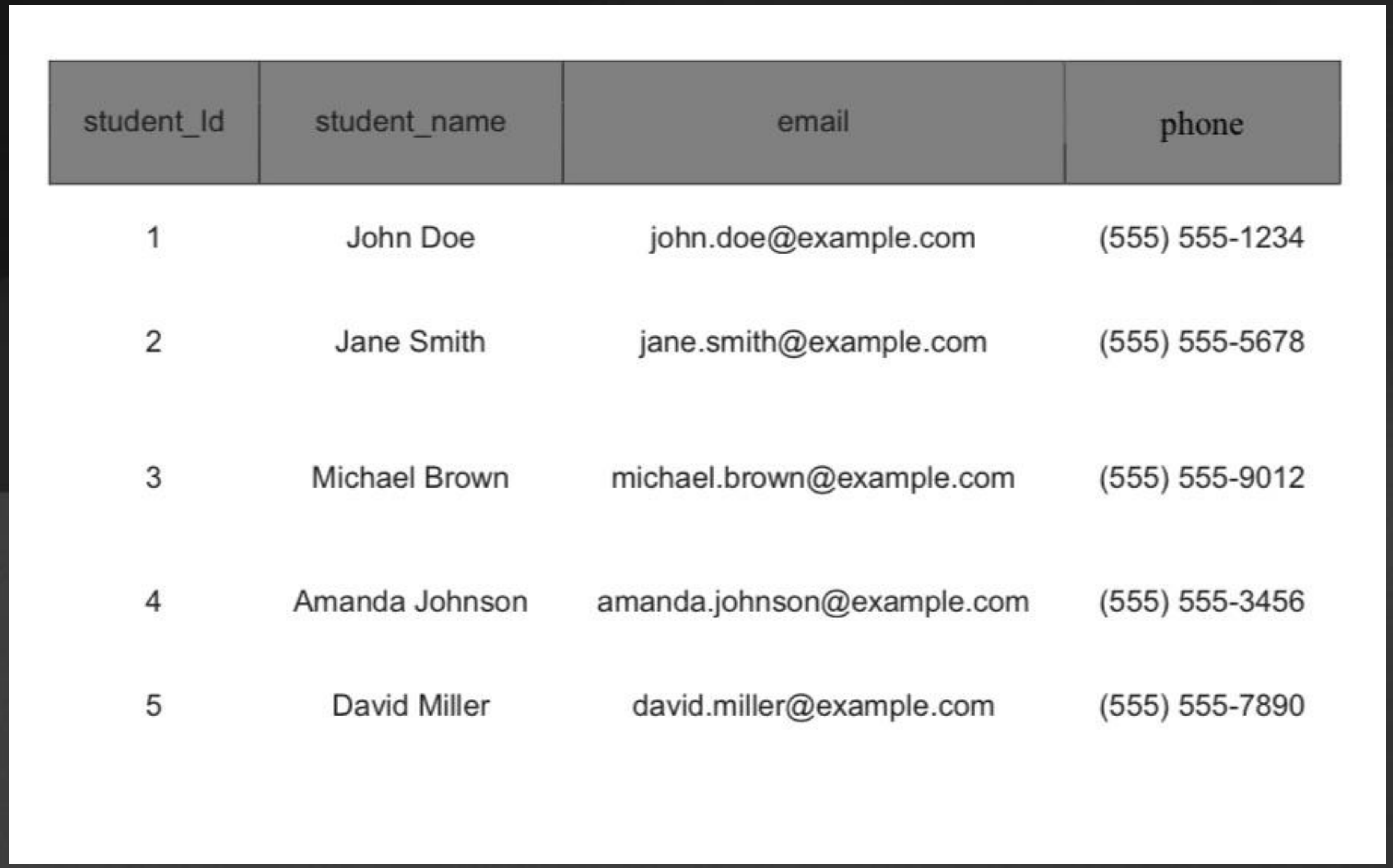
In this table we have transitive dependancy which can lead to data redundancy.

Here's the breakdown of Grade table normalization to 3NF, along with the corresponding table structures:

```
CREATE TABLE Enrollment (

enrollment_id INT PRIMARY KEY AUTO_INCREMENT,

student_id INT NOT NULL,

course_id INT NOT NULL,

-- Optional attributes (semester, term, etc.)

FOREIGN KEY (student_id) REFERENCES Student(student_id),

FOREIGN KEY (course_id) REFERENCES Course(course_id)

);
```

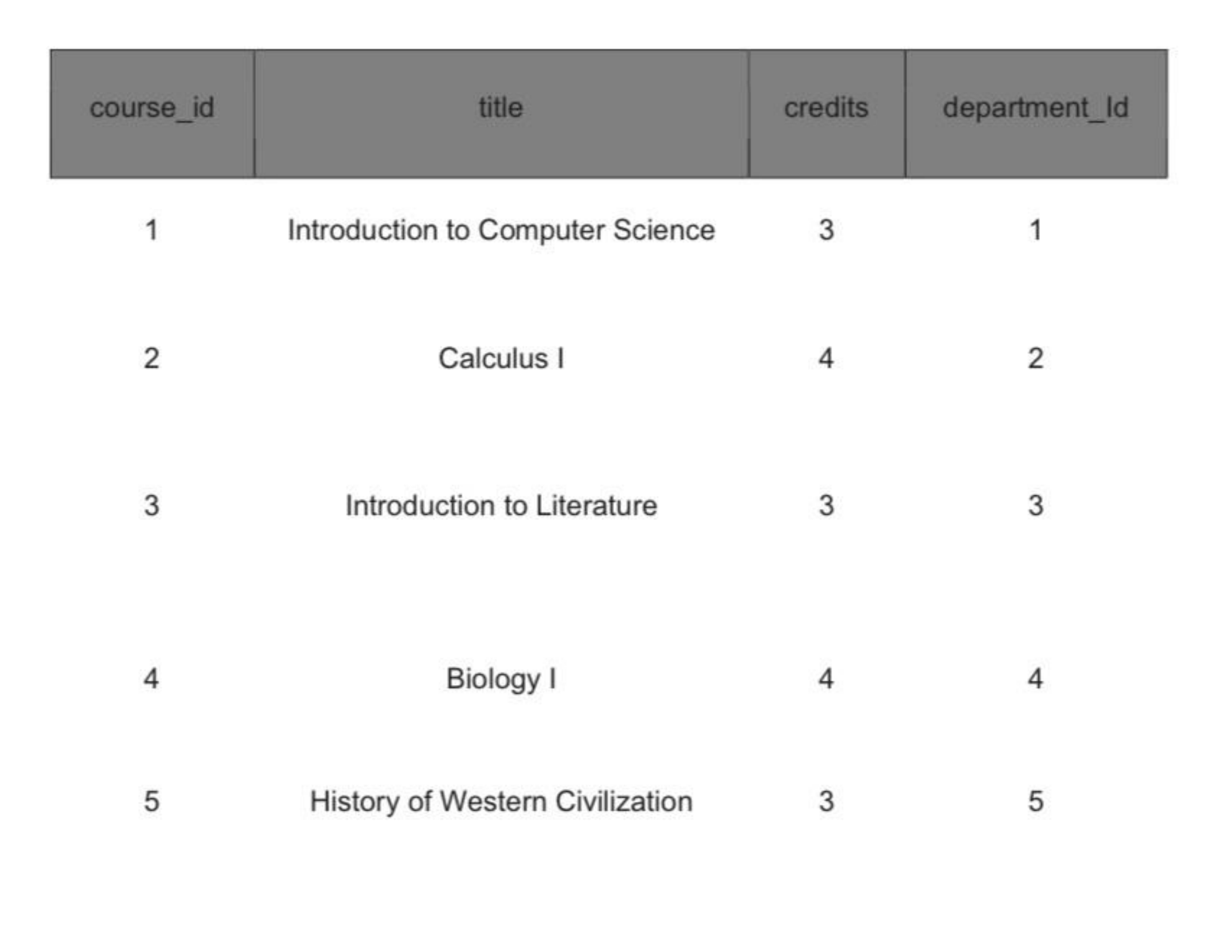
Tables with some Data

Student Table:



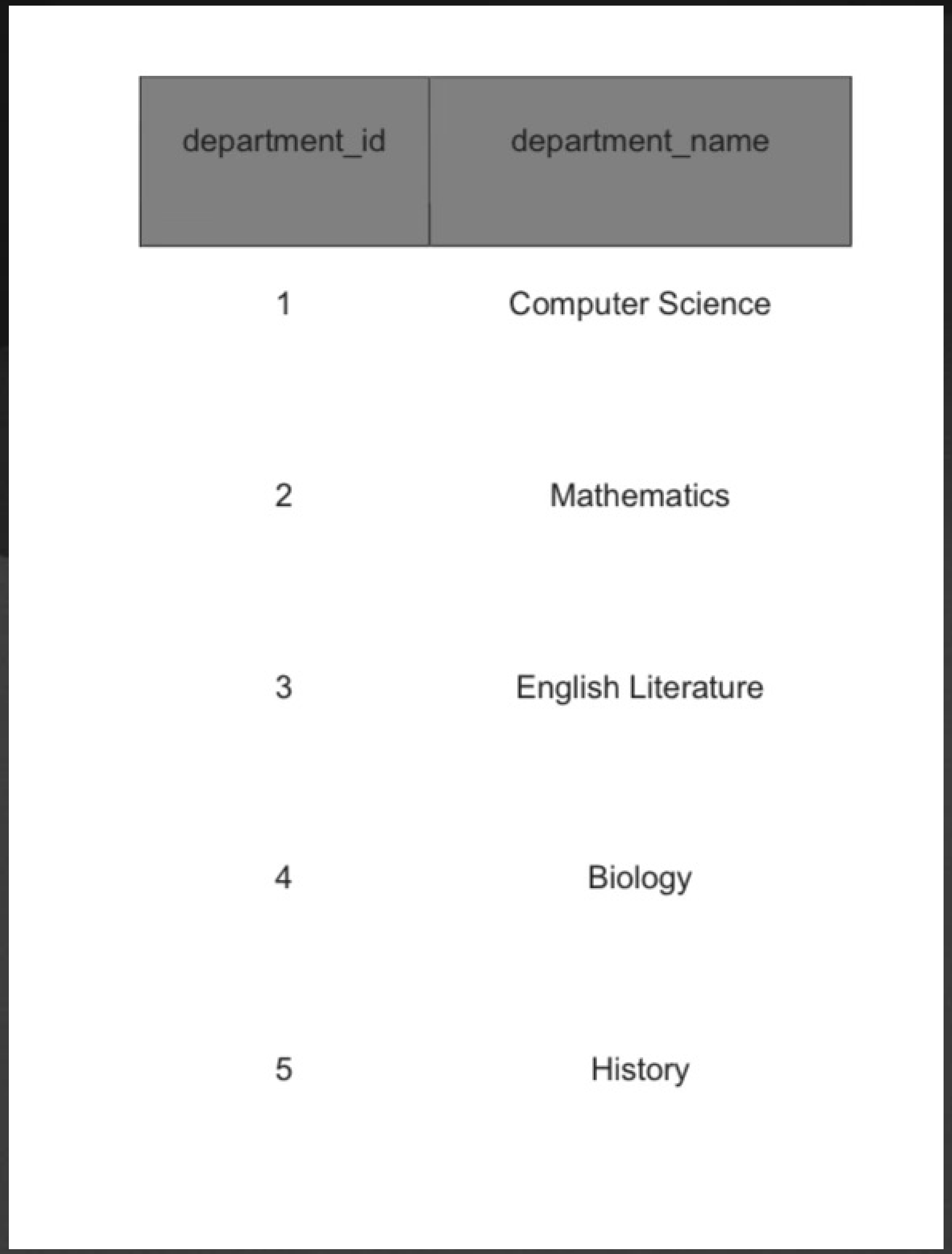
student_Id	student_name	email	phone
1	John Doe	john.doe@example.com	(555) 555-1234
2	Jane Smith	jane.smith@example.com	(555) 555-5678
3	Michael Brown	michael.brown@example.com	(555) 555-9012
4	Amanda Johnson	amanda.johnson@example.com	(555) 555-3456
5	David Miller	david.miller@example.com	(555) 555-7890

Course Table:



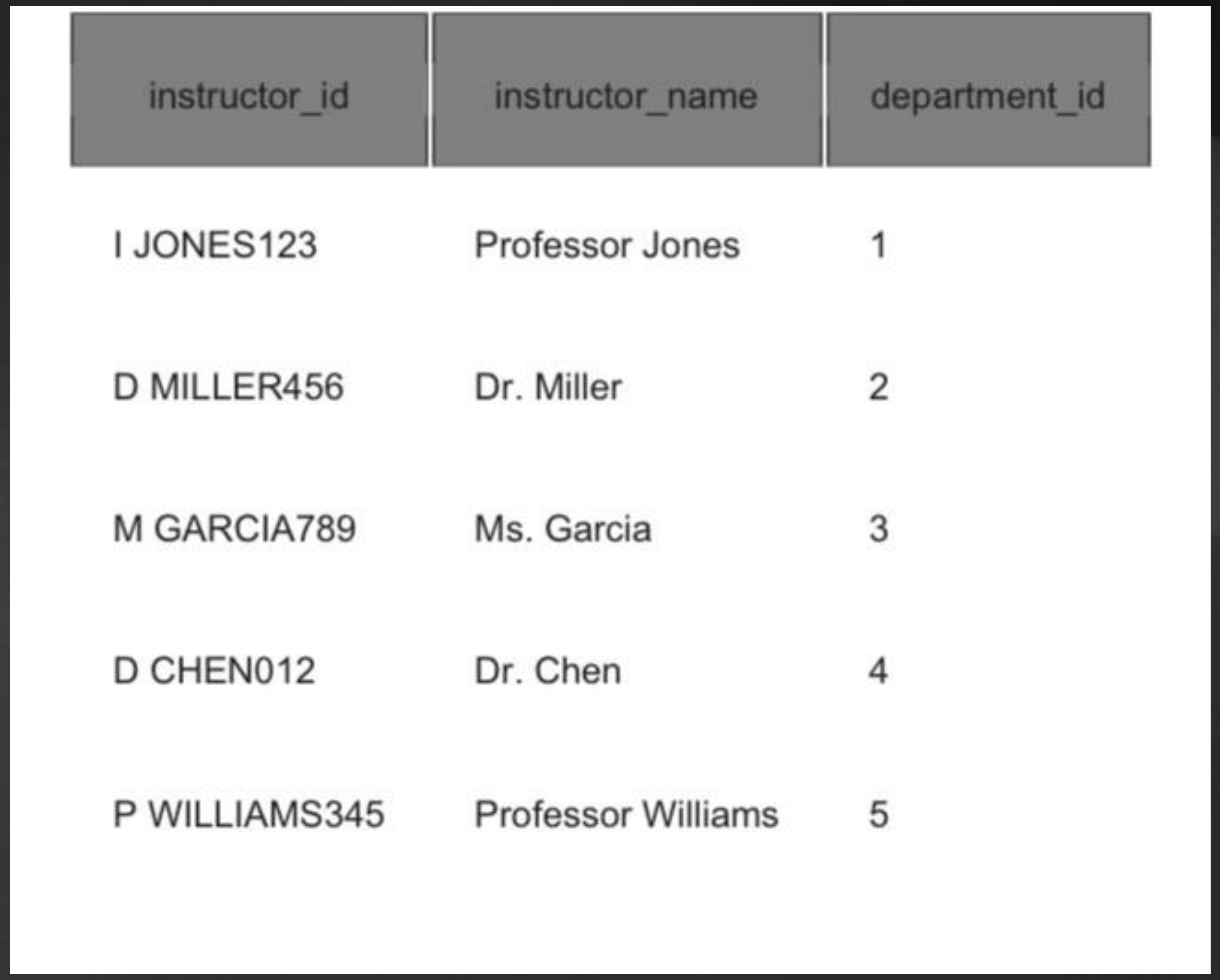
course_id	title	credits	department_Id
1	Introduction to Computer Science	3	1
2	Calculus I	4	2
3	Introduction to Literature	3	3
4	Biology I	4	4
5	History of Western Civilization	3	5

Department Table:



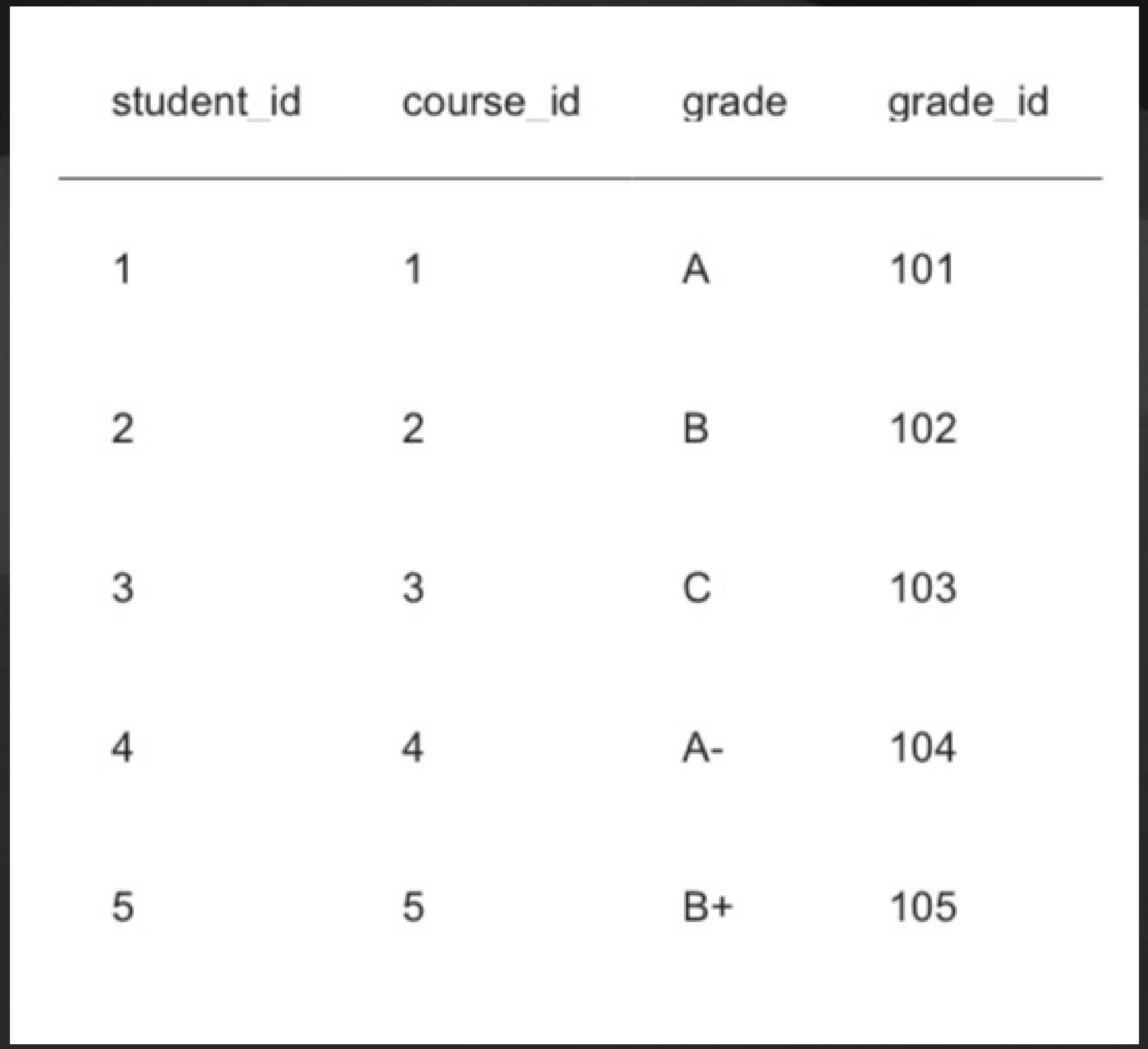
department_id	department_name
1	Computer Science
2	Mathematics
3	English Literature
4	Biology
5	History

Instructor Table:



instructor_id	instructor_name	department_id
I JONES123	Professor Jones	1
D MILLER456	Dr. Miller	2
M GARCIA789	Ms. Garcia	3
D CHEN012	Dr. Chen	4
P WILLIAMS345	Professor Williams	5

Grade Table:



student_id	course_id	grade	grade_id
1	1	A	101
2	2	B	102
3	3	C	103
4	4	A-	104
5	5	B+	105

Few SQL queries:

1. List all instructors and their departments:

```sql
SELECT i.instructor_name, d.department_name
FROM Instructor i
INNER JOIN Department d ON i.department_id = d.department_id;
```

2. Find all students enrolled in a specific course (course ID = 101):

```sql
SELECT s.student_name
FROM Student s
INNER JOIN Enrollment e ON s.student_id = e.student_id
WHERE e.course_id = 101;
```

3. Get the average grade for a particular course (course ID = 101):

```sql
SELECT AVG(g.grade) AS average_grade
FROM Grade g
INNER JOIN Enrollment e ON g.student_id = e.student_id
WHERE e.course_id = 101;
```

4. List all students with a grade of 'A' and their instructors:

```
SELECT s.student_name, i.instructor

_

name

FROM Student s

INNER JOIN Enrollment e ON s.student_id = e.student_id

INNER JOIN Grade g ON e.student_id = g.student_id AND e.course_id =

g.course_id

INNER JOIN Instructor i ON e.course_id = i.course_id -- Assuming

instructors teach courses they are enrolled in

WHERE g.grade = 'A';
```

Views:

1. View for Instructor Information with Department Details:

```sql
CREATE VIEW InstructorDetails AS

SELECT i.instructor_name, d.department_name

FROM Instructor i

INNER JOIN Department d ON i.department_id = d.department_id;
```

2. View for Enrolled Students with Course Details:

```sql
CREATE VIEW EnrolledStudents AS

SELECT s.student_name, c.title, c.credits

FROM Student s

INNER JOIN Enrollment e ON s.student_id = e.student_id

INNER JOIN Course c ON e.course_id = c.course_id;
```

3. View for StEnrollmentes with Course Information:

```sql
CREATE VIEW StudentGrades AS

SELECT s.student_name, c.title, g.grade

FROM Student s

INNER JOIN Enrollment e ON s.student_id = e.student_id

INNER JOIN Grade g ON e.student_id = g.student_id AND e.course_id =

g.course_id

INNER JOIN Course c ON e.course_id = c.course_id;
```

4. View for Average Grades per Course:

```sql
CREATE VIEW AverageGradesPerCourse AS

SELECT c.title, AVG(g.grade) AS average_grade

FROM Course c

INNER JOIN Enrollment e ON c.course_id = e.course_id

INNER JOIN Grade g ON e.student_id = g.student_id AND e.course_id =

g.course_id

GROUP BY c.course_id;
```

6. View for Students with Specific Grade in a Department:

```
CREATE VIEW StudentsByGradeInDept AS

SELECT s.student_name, i.department_name, g.grade

FROM Student s

INNER JOIN Enrollment e ON s.student_id = e.student_id

INNER JOIN Grade g ON e.student_id = g.student_id AND e.course_id =

g.course_id

INNER JOIN Instructor, i ON e.course_id = i.course_

INNER JOIN Department d ON i.department_id = d.department_id

WHERE g.grade = 'A';
```

Thank you