CP363 Student-Record Database Management

University of Laurier

November 1st, 2023

Thomas Clipston, Thu Mai, Zorah Jawadi, Amirhossein Ahmadinoori

Table of Contents

1. [Introduction](#p3psvwc2tv2y)
2. [Project Objective](#u4ax5ph60jd0)
3. [Scope](#oxtjs67wrlnd)
4. [Project System Specifications](#xdoevmkk7q0t)
5. [Relational Model](#1aeryh6707e9)
6. [Entity Relationship Diagram](#1vs7a5z0nsvw)
7. [Relational Schema](#wt60ru1pq1v5)
8. [Normalization Checklist](#5nm9jbulz9k9)
9. [Database Overview](#sk8dw06lp6b7)
10. [Student Table](#hqq2kppwn344)
11. [Program Table](#sdavkl9jf1xu)
12. [Course Table](#fjo6ejouy6zd)
13. [Department Table](#forhl62zf119)
14. [Program-Enrolment Table](#vwyopo2ofo31)
15. [Course-Enrolment Table](#mcivo87jk76)
16. [Grade\_Content Table](#bdbcefndw96h)
17. [Conclusion](#te1jmulggzdg)
18. [References](#vix7bec0tvbp)
19. [Code](#pn0mi7q0stfd)

**Introduction:**

The Student Record-Keeping System aims to establish an efficient platform for storing comprehensive information about students and their academic pursuits. This system will be implemented using modern MySQL, adhering to coding and structural standards. The database will be designed to be fully inclusive, accommodating various types of students and capable of storing extensive information about different facets of their studies. Databases, like the one developed in this project, hold significant importance for educational institutions. They offer a virtual repository for information, eliminating the need for large physical filing systems. This shift towards digital storage not only ensures data accuracy and accessibility but also frees up manpower that would otherwise be dedicated to maintaining a labor-intensive physical database. By adopting a more modern database management system, educational institutes can redirect resources towards more strategic tasks, contributing to overall operational efficiency and effectiveness.

**Project Objective:**

* To create a database using MySQL for student information to store data relating to the student’s courses, grades, personal information, and faculty.
* Separate data based upon specific categories into different tables using decomposition.
* Understand the relationships between the different contrived tables.
* Reduce redundancy within the database.

**Scope:**

The project will focus mainly on students, their courses, and grades and enrollment information . The focus is on considering student information relevant to academics, and their enrollment in the University.

**Project System Specifications:**

This database runs off MySQL, running on the Windows 10/11 operating system. Data is accessible through the command prompt using the MySQL interface. It is important to have an account for the database, either being the root user or having an account created from the root user.

**Relational Model:**

Student:

| Student\_ID | First\_Name | Last\_Name | Birthdate | Phone | Address | Domestic\_Student | Email |
| --- | --- | --- | --- | --- | --- | --- | --- |

Program:

| Program\_ID | Department\_ID | Name | Description |
| --- | --- | --- | --- |

Department:

| Department\_ID | Name | Description |
| --- | --- | --- |

Course:

| Course\_ID | Name | Description | Domestic\_Tuition | International\_Tuition |
| --- | --- | --- | --- | --- |

Course-Enrolment:

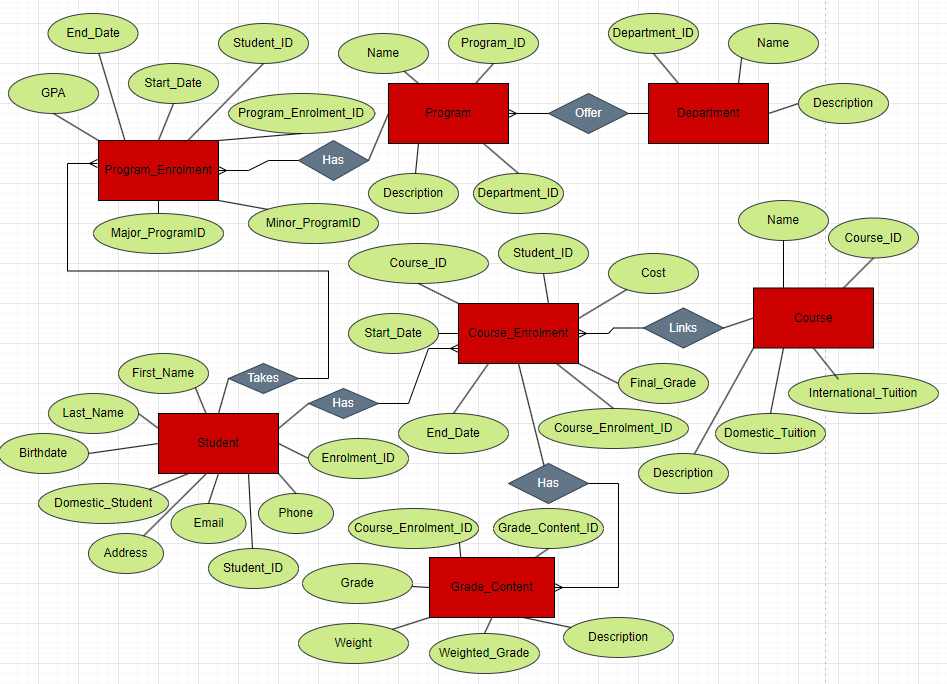
| Course\_Enrolment\_ID | Student\_ID | Course\_ID | Start\_Date | End\_Date | Final\_Grade | Cost |
| --- | --- | --- | --- | --- | --- | --- |

Program-Enrolment:

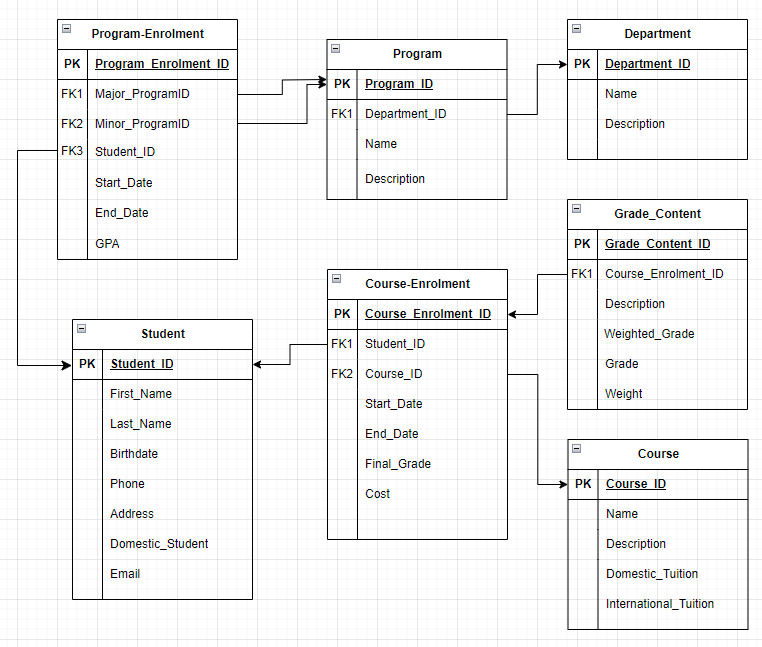
| Program\_Enrolment\_ID | Student\_ID | Major\_ProgramID | Minor\_ProgramID | Start\_Date | End\_Date | GPA |
| --- | --- | --- | --- | --- | --- | --- |

Grade\_Content:

| Grade\_Content\_ID | Course\_Enrolment\_ID | Description | Weighted\_Grade | Grade | Weight |
| --- | --- | --- | --- | --- | --- |

**Entity Relationship Diagram:**  


**Relational Schema:**

****

**Entity Relationship Diagram Explanation:**

Student Table:

* Relationships:
  + One-to-Many with Program-Enrolment table.
  + One-to-Many with Course-Enrolment table.
* Explanation:
  + A Student has many program enrolments. E.g., Lisa can have a double degree Computer Science and Business.
  + A Student has many course enrolments. E.g., John can have CP104 and CP202, and he can withdraw and retake them if he fails these classes.

Program Table:

* Relationships:
  + One-to-Many with Program-Enrolment table.
  + Many-to-One with the Department table.
* Explanation:
  + A Program has many program enrolments.
  + There are many Programs in a Department.

Course Table:

* Relationships:
  + One-to-Many with Course-Enrolment table.
* Explanation:
  + A Course has many course enrolments. E.g., Lisa and John both registered course CP213 in this term.

Department Table:

* Relationships:
  + One-to-Many with Program table.
* Explanation:
  + A Department offers many programs. E.g. The Science Department has Data Science, Computer Science, Psychology, etc.

Program-Enrolment Table:

* Relationships:
  + Many-to-One with Program table.
  + Many-to-One with Student table.
* Explanation:
  + A program has many program enrolments.
  + A student has many program enrolments.

Course-Enrolment Table:

* Relationships:
  + Many-to-One with Student table.
  + Many-to-One with Course table.
  + One-to-Many with Grade\_Content table.
* Explanation:
  + A Student can take many course enrolments.
  + A course can have many course enrolments.
  + Each course enrolment has a grade content. E.g., CP213 has assignments, labs, test 1, test 2, and final exam.

**Normalization Checklist**

**1NF:**

* ~~Each column must have a unique name.~~
* ~~The order of the rows and the order of columns does not matter.~~
* ~~Each column must have a single data type.~~
* ~~No two rows can contain identical values.~~
* ~~Each column must contain a single value.~~
* ~~Columns cannot contain repeating groups.~~

**2NF:**

* ~~The table is in 1NF.~~
* ~~All non-key attributes are fully functionally dependent on the primary key.~~

**3NF:**

* ~~The table is in 2NF.~~
* ~~The table contains no transitive.~~

**Database Overview:**

This database constructs a series of tables around students and their academic life. The database is capable of accessing and storing data relating to personal, academic and department information.

**Student Table**

The "Student" table holds comprehensive information about each student. It includes a unique student ID as the primary key, along with personal details such as first name, last name, birthdate, phone number, address, student type (domestic student or international student) and email.

| Student\_ID | First\_Name | Last\_Name | Birthdate | Phone | Address | Domestic\_Student | Email |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 169547812 | Samantha | Hayes | 2000-05-15 | 519-456-7890 | 123B Oriole St, Waterloo | True | samantha.hayes@mylaurier.ca |
| 169212345 | Tien | Nguyen | 1999-08-20 | 519-654-3210 | 6 Rodney St, Waterloo | False | tien.nguyen@mylaurier.ca |
| 169876543 | Aarav | Sharma | 2001-03-03 | 519-555-5555 | 78 Coghill Place, Waterloo | False | aarav.sharma@mylaurier.ca |

* Student\_ID:
  + A unique identifier for each student serves as the primary key for the table.
  + Data type: Integer (Auto-increment)
  + Constraints: Primary Key
  + Nullability: NOT NULL
* First\_Name:
  + The first name of the student.
  + Data type: String
  + Nullability: NOT NULL
* Last\_Name:
  + The last name of the student.
  + Data type: String
  + Nullability: NOT NULL
* Birthdate:
  + The date of birth of the student.
  + Data type: DATE
  + Nullability: Can be NULL
* Phone:
  + The contact phone number of the student.
  + Data type: String
  + Nullability: Can be NULL
* Address:
  + The current address of the student, typically including street, city, state, and ZIP code.
  + Data type: String
  + Nullability: Can be NULL (if address information is not always available)
* Domestic\_Student:
  + Flag indicating whether the student is domestic or international.
  + Data type: Boolean (True: Domestic, False: International)
* Email:
  + The email address that students use for communication and notifications.
  + Data type: String
  + Nullability: Can be NULL

**Course-Enrolment Table**

The "Course-Enrolment" table records details about student enrollments in specific program courses. It includes a course enrollment ID as a primary key, a reference to the student ID, course ID, start and end dates of the enrollment, final grade achieved, and associated cost.

| Course\_Enrolment\_ID | Student\_ID | Course\_ID | Start\_Date | End\_Date | Final\_Grade | Cost |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 169547812 | 104 | ‘2023-01-15’ | ‘2023-03-30’ | 87.50 | 5000.00 |
| 2 | 169212345 | 465 | ‘2023-04-8’ | ‘2023-08-15’ | 92.30 | 1000.00 |
| 3 | 169876543 | 340 | ‘2023-01-15’ | ‘2023-03-30’ | 78.90 | 1500.00 |

* Course\_Enrolment\_ID:
  + A unique identifier for each student-course relationship.
  + Data type: Integer (Auto-increment)
  + Constraints: Primary Key
  + Nullability: NOT NULL
* Student\_ID:
  + A unique identifier for each student.
  + Data type: Integer
  + Constraints: Foreign Key REFERENCING Student table
  + Nullability: NOT NULL
* Course\_ID:
  + A unique identifier for each course.
  + Data type: Integer
  + Constraints: Foreign Key REFERENCING course table
  + Nullability: NOT NULL
* Start\_Date:
  + The start date of your enrollment
  + Data type: Date
  + Nullability: NOT NULL
* End\_Date:
  + The end date of your enrollment
  + Data type: Date
  + Nullability: NOT NULL
* Final\_Grade:
  + The final grade achieved in the course.
  + Data type: DECIMAL
  + Nullability: Can be NULL
* Cost:
  + The cost associated with the enrollment.
  + Data type: DECIMAL
  + Nullability: Can be NULL (if the cost information is not always available)

**Program-Enrolment Table:**

The "Program-Enrolment" table captures information about students' participation in programs. It includes a unique program enrolment ID as the primary key, along with the student ID, major program ID and minor program ID, indicating the start date and end date of the program and GPA for each program enrolment

| Program\_Enrolment\_ID | Student\_ID | Major\_ProgramID | Minor\_ProgramID | Start\_Date | End\_Date | GPA |
| --- | --- | --- | --- | --- | --- | --- |
| 101 | 169547812 | 11 | 51 | ‘2023-01-15’ | ‘2027-03-30’ | 67.00 |
| 102 | 169212345 | 12 | 52 | ‘2023-04-8’ | ‘2028-08-15’ | 89.00 |
| 103 | 169876543 | 13 | 53 | ‘2023-01-15’ | ‘2029-03-30’ | 56.00 |

* Program\_Enrolment\_ID:
  + A unique identifier for each student-program relationship.
  + Data type: Integer
  + Constraints: Primary Key
  + Nullability: NOT NULL
* Student\_ID:
  + A unique identifier for each student.
  + Data type: Integer
  + Constraints: Foreign Key REFERENCING Student table
  + Nullability: NOT NULL
* Major\_ProgramID:
  + A unique identifier for major program.
  + Data type: Integer
  + Constraints: Foreign Key REFERENCING Program table
  + Nullability: NOT NULL
* Minor\_ProgramID:
  + A unique identifier for minor program.
  + Data type: Integer
  + Constraints: Foreign Key REFERENCING Program table
  + Nullability: Can be NULL
* Start Date:
  + The date when the student joined the program.
  + Data type: DATE
  + Nullability: NOT NULL
* End Date:
  + The date when the student finished the program.
  + Data type: DATE
  + Nullability: NOT NULL
* GPA:
  + The final grade achieved in the program.
  + Data type: DECIMAL
  + Nullability: Can be NULL

**Course Table:**

The "Course" table contains details about different courses offered. It includes a course ID as the primary key, along with the course name, description, domestic tuition cost, and international tuition cost. The table provides essential information about the courses available.

| Course\_ID | Name | Description | Domestic Cost | International cost |
| --- | --- | --- | --- | --- |
| 104 | Introduction to Programming | Learn the basics of programming | 1000.00 | 3000.00 |
| 465 | Database II | Explore the fundamentals of database management | 1200.00 | 3600.00 |
| 340 | E-commerce | Learn the fundamentals of online business, from setting up an online store to managing transactions. | 1100.00 | 3300.00 |

* Course\_ID:
  + A unique identifier for each course
  + Data Type: Integer (Auto-increment)
  + Constraints: Primary Key
  + Nullability: NOT NULL

Name:

* + The name or title of the course
  + Data Type: String
  + Constraints: NOT NULL

Description:

* + The description or details about the course
  + Data Type: String
  + Constraints: Can be NULL
* Domestic\_Cost:
  + Cost of the course for domestic students.
  + Data Type: Decimal
  + Constraints: Can be NULL
* International\_Cost:
  + Cost of the course for international students.
  + Data Type: Decimal
  + Constraints: Can be NULL

**Program Table:**

The "Program" table provides information about academic programs. It includes a program ID as the primary key, along with the program name, description, and the department ID. The department ID is a foreign key linking to the "Department" table.

| Program\_ID | Name | Description | Department\_ID |
| --- | --- | --- | --- |
| 301 | Computer Science | The Computer Science program explores algorithms, software development, and cutting-edge technologies, shaping graduates for success in the rapidly evolving tech landscape. | 401 |
| 302 | Biochemistry | The Biochemistry program delves into molecular processes, bridging biology and chemistry, preparing students for impactful careers in scientific research and discovery. | 402 |
| 303 | Business Administration | The Business Administration program develops strategic thinkers, equipping students with essential skills for leadership roles in diverse business environments and industries. | 403 |

Program\_ID:

* A unique identifier for each program.
* Data Type: Integer (Auto-increment)
* Constraints: Primary Key
* Nullability: NOT NULL

Department\_ID:

* A unique identifier for each Department.
* Data Type: Integer
* Constraints: Foreign Key REFERENCING Department table
* Nullability: NOT NULL

Name:

* The name or title of the Programt.
* Data Type: String
* Constraints: NOT NULL

Description:

* Description or details about the program.
* Data Type: String
* Constraints: Can be NULL

**Department Table:**

The "Department" table contains information about academic departments. It includes a department ID as the primary key, along with the department name and description. This table helps organize and categorize academic programs.

| Department\_ID | Name | Description |
| --- | --- | --- |
| 401 | Science | The Science Department offers diverse programs, fostering curiosity and critical thinking, preparing students for scientific exploration, innovation, and research. |
| 402 | Biology | The Biology Department specializes in life sciences, fostering a deep understanding of organisms, ecosystems, and genetics for scientific inquiry and discovery. |
| 403 | Business | The Business Department cultivates business acumen, leadership skills, and strategic thinking, preparing students for success in a dynamic and competitive corporate landscape. |

Department\_ID:

* A unique identifier for each department.
* Data Type: Integer (Auto-increment)
* Constraints: Primary Key
* Nullability: NOT NULL

Name:

* The name or title of the department.
* Data Type: String
* Constraints: NOT NULL

Description:

* Description or details about the department.
* Data Type: String
* Constraints: Can be NULL

**Grade\_Content Table**

The "Grade\_Content" table stores information related to the grades obtained by students. It includes a grade content ID as the primary key, the course enrollment ID (foreign key referencing "course-enrolment" table), the description, the grade achieved, the weight of each description, the weighted grade (= multiply each grade by its weight).

| Grade\_Content\_ID | Course\_Enrolment\_ID | Description | Weight | Grade | Weighted\_Grade |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | Midterm Exam | 0.35 | 65.00 | 23.00 |
| 2 | 2 | Quizzes | 0.10 | 90.00 | 9.00 |
| 3 | 3 | Project Report | 0.15 | 70.00 | 11.00 |

Grade\_Content\_ID:

* A unique identifier for each grade content
* Data Type: Integer (Auto-increment)
* Constraints: Primary Key
* Nullability: NOT NULL

Course\_Enrolment\_ID:

* A unique identifier for the type of content for each course enrolment.
* Data Type: Integer
* Constraints: Foreign Key REFERENCING Course-Enrolment table
* Nullability: NOT NULL

Description:

* Describes the nature or type of the grade content (e.g., Midterm, Exam, Quizzes, etc.)
* Data Type: String
* Nullability: Can be NULL

Weight:

* The weight assigned to the grade content, indicating its relative importance in the overall assessment
* Data Type: DECIMAL
* Nullability: Can be NULL

Grade:

* The calculated grade considering the specific grade content
* Data Type: DECIMAL
* Nullability: Can be NULL

Weighted\_Grade:

* The calculated grade considering the weight assigned to it
* Data Type: DECIMAL
* Nullability: Can be NULL

**Conclusion:**

Therefore, the execution of this project has deepened our comprehension of developing efficient and structurally sound databases. Through our commitment to adhering to accurate coding standards and best practices, we have successfully crafted a database capable of efficiently managing a diverse set of student information. The database presented here marks a significant step towards enhancing documentation practices within educational institutes, offering a virtual storage option for information. This system brings about improved efficiency within these institutions, affording them the capability to seamlessly execute queries on the stored data. The potential value of such queries is substantial, as they empower users to compare different sets of student data, providing valuable insights for informed decision-making and the administrative processes. As a result, the project not only contributes to the technical advancement of database management but also holds practical implications for improving overall operational efficiency in educational settings.

**References:**

- Andrew, C. Ronald, B. Patrick, G. “Expert PHP and MySQL”(2010), *Wiley Publishing.* [*https://books.google.ca/books?id=43ZKWL\_Rmx4C&printsec=frontcover#v=onepage&q&f=false*](https://books.google.ca/books?id=43ZKWL_Rmx4C&printsec=frontcover#v=onepage&q&f=false)

- Crayon Corporation “Top 10 best practices in MySQL”(2018) *Crayon,* [*https://crayondata.ai/top-10-best-practices-in-mysql*](https://crayondata.ai/top-10-best-practices-in-mysql/)

*-* Gao, S. “Cp363 CourseWare Slides” *Wilfrid Laurier University.* Access through the University Courseware for Cp363

- Halpin, T. Morgan, T. “Information Modeling and Relational Databases” (2008) *Elsevier Inc.* <https://books.google.ca/books?hl=en&lr=&id=puO_VlbR_x4C&oi=fnd&pg=PP1&dq=relational+databases&ots=LeUrpYnYbr&sig=fYUU8lCni7MGDMJYg5kbDD9OJ58#v=onepage&q=relational%20databases&f=false>

”

**CODE:**

Create Database Student\_Record\_Keeping;

Use Student\_Record\_Keeping;

CREATE TABLE Student (

student\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

birthdate DATE NOT NULL,

phone VARCHAR(20),

address VARCHAR(255),

domestic\_student BOOLEAN,

email VARCHAR(255)

);

INSERT INTO Student (first\_name, last\_name, birthdate, phone, address, domestic\_student, email)

VALUES

('Samantha', 'Hayes', '2000-01-15', '519-1234', '123 Abbotswood St', TRUE, 'samantha.hayes@mylaurier.ca'),

('Jane', 'Smith', '1999-08-22', '519-5678', '46 Albert St', FALSE, 'jane.smith@mylaurier.ca'),

('Bob', 'Johnson', '2001-03-10', '519-9876', '89 Allison St', TRUE, 'bob.johnson@mylaurier.ca'),

('Alice', 'Williams', '2000-05-29', '519-4321', '321 Box Grove Place', FALSE, 'alice.williams@mylaurier.ca'),

('Michael', 'Davis', '1999-11-18', '519-8765', '6 Bricker Avenue', TRUE, 'michael.davis@mylaurier.ca'),

('Aarav', 'Sharma', '2001-02-04', '519-2345', '87 Dorset St', FALSE, 'aarav.sharma@mylaurier.ca'),

('Li', 'Wei', '2000-09-12', '519-6789', '16 Maple St', TRUE, 'li.wei@mylaurier.ca'),

('Ella', 'Jones', '1999-07-03', '519-5432', '4 Walnut St', FALSE, 'ella.jones@mylaurier.ca'),

('David', 'Wilson', '2001-01-25', '519-8765', '543 Karen Walk', TRUE, 'david.wilson@mylaurier.ca'),

('Tien', 'Nguyen', '2000-06-14', '519-3210', '75B Pacific Avenue', FALSE, 'tien.nguyen@mylaurier.ca');

CREATE TABLE Department (

department\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

description TEXT

);

INSERT INTO Department (name, description)

VALUES

('Science', 'The Science Department fosters exploration, discovery, and innovation through diverse disciplines, cultivating a dynamic learning environment for scientific inquiry and advancement.'),

('Business', 'The Business Department drives strategic leadership and entrepreneurial thinking, equipping students with skills for global markets and sustainable business practices.'),

('Psychology', 'The Psychology Department explores the complexities of the human mind, behavior, and emotions, fostering understanding and critical thinking in psychological science and applications.'),

('Biology', 'The Biology Department delves into the wonders of life, studying organisms and ecosystems to advance knowledge and inspire a profound appreciation for biological diversity.'),

('History', 'The History Department illuminates the past, analyzing events, cultures, and societies, fostering a deep understanding of human experiences and shaping informed perspectives on the present.'),

('Mathematics', 'The Mathematics Department explores abstract structures and patterns, cultivating logical reasoning and problem-solving skills essential for various disciplines and real-world applications.'),

('Education', 'The Education Department shapes future educators, emphasizing pedagogy, inclusivity, and transformative teaching approaches to empower students and positively impact educational practices.'),

('Economics', 'The Economics Department delves into economic systems, policies, and behaviors, cultivating analytical thinkers and preparing students to address complex economic challenges and opportunities.'),

('Sociology', 'The Sociology Department examines societal structures, relationships, and dynamics, fostering a critical understanding of human societies and empowering students to address social issues with insight and empathy.'),

('Chemistry', 'The Chemistry Department investigates the properties and transformations of matter, promoting scientific inquiry and innovation, and preparing students for impactful contributions to the chemical sciences.');

CREATE TABLE Course (

course\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

description TEXT,

domestic\_cost DECIMAL(10, 2),

international\_cost DECIMAL(10, 2)

);

INSERT INTO Course (name, description, domestic\_cost, international\_cost)

VALUES

('Computer Science 101', 'Introduction to Computer Science', 1000.00, 3000.00),

('Business Administration 201', 'Principles of Business Management', 1200.00, 3600.00),

('Psychology 301', 'Cognitive Psychology', 1100.00, 3300.00),

('Biology 101', 'Introduction to Biology', 1050.00, 3150.00),

('History 201', 'World History: 18th to 20th Century', 1000.00, 3000.00),

('Mathematics 301', 'Advanced Calculus', 1150.00, 3450.00),

('Education 101', 'Foundations of Education', 950.00, 2850.00),

('Economics 201', 'Microeconomics', 1120.00, 3360.00),

('Sociology 301', 'Social Inequality', 1080.00, 3240.00),

('Chemistry 101', 'General Chemistry', 1100.00, 3300.00);

CREATE TABLE Program (

program\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

description TEXT,

department\_id int,

FOREIGN KEY (department\_id) REFERENCES department(department\_id)

);

INSERT INTO Program (name, description, department\_id)

VALUES

('Computer Science', 'Explore the world of computing and software development.', 1),

('Business Administration', 'Develop business leadership and management skills.', 2),

('Counseling Psychology', 'Study the human mind and behavior.', 3),

('Ecology and Environmental Biology', 'Examine the diversity of living organisms and ecosystems.', 4),

('Medieval and Renaissance History', 'Explore historical events and societal changes.', 5),

('Applied Mathematics', 'Develop strong analytical and problem-solving skills.', 6),

('Early Childhood Education', 'Prepare for a career in teaching and educational leadership.', 7),

('Microeconomics and Macroeconomics', 'Analyze economic systems and policies.', 8),

('Social Work', 'Study social structures and relationships.', 9),

('Biochemistry', 'Investigate the properties and transformations of matter.', 10);

CREATE TABLE Program\_Enrolment (

Program\_Enrolment\_ID INT PRIMARY KEY,

Student\_ID INT NOT NULL,

Major\_ProgramID INT NOT NULL,

Minor\_ProgramID INT,

Start\_Date DATE NOT NULL,

End\_Date DATE NOT NULL,

GPA DECIMAL,

CONSTRAINT fk\_Student FOREIGN KEY (Student\_ID) REFERENCES Student(Student\_ID),

CONSTRAINT fk\_Major\_Program FOREIGN KEY (Major\_ProgramID) REFERENCES Program(Program\_ID),

CONSTRAINT fk\_Minor\_Program FOREIGN KEY (Minor\_ProgramID) REFERENCES Program(Program\_ID)

);

INSERT INTO Program\_Enrolment (Program\_Enrolment\_ID, Student\_ID, Major\_ProgramID, Minor\_ProgramID, Start\_Date, End\_Date, GPA)

VALUES

(1, 169428753, 201, 301, '2022-01-15', '2023-05-20', 3.5),

(2, 169662194, 202, NULL, '2021-09-10', '2023-04-30', 3.8),

(3, 169234567, 201, 302, '2021-08-22', '2022-12-18', 3.2),

(4, 169875321, 203, 303, '2022-03-05', '2023-06-15', 3.9),

(5, 169456789, 204, NULL, '2021-11-12', '2022-10-05', 3.0),

(6, 169345789, 202, 304, '2022-02-28', '2023-05-25', 3.7),

(7, 167789012, 203, NULL, '2021-07-08', '2022-11-30', NULL),

(8, 145123789, 201, 305, '2021-12-10', '2023-03-15', 3.4),

(9, 189222567, 204, NULL, '2022-05-18', '2023-08-22', 3.6),

(10, 169230192, 202, 306, '2021-10-01', '2023-01-28', 3.1);

CREATE TABLE Course\_Enrolment (

Course\_Enrolment\_ID INT PRIMARY KEY AUTO\_INCREMENT = 0,

Student\_ID INT NOT NULL,

Course\_ID INT NOT NULL,

Start\_Date DATE NOT NULL,

End\_Date DATE NOT NULL,

Final\_Grade DECIMAL,

Cost DECIMAL,

CONSTRAINT fk\_Student\_Course\_Enrolment FOREIGN KEY (Student\_ID) REFERENCES Student(Student\_ID),

CONSTRAINT fk\_Course\_Course\_Enrolment FOREIGN KEY (Course\_ID) REFERENCES Course(Course\_ID)

);

INSERT INTO Course\_Enrolment (Student\_ID, Course\_ID, Start\_Date, End\_Date, Final\_Grade, Cost)

VALUES

(1, 1,'2021-09-10', '2022-04-30', 3.8, 600.00),

(2, 2, '2022-03-05', '2022-06-15', 3.9, 550.00),

(3, 2, '2021-11-12', '2022-10-05', 3.0, 700.00),

(4, 2, '2022-02-28', '2022-05-25', 3.7, 480.00),

(5, 3, '2021-07-08', '2022-11-30', NULL, 800.00),

(6, 3, '2021-12-10', '2022-03-15', 3.4, 600.00),

(7, 2, '2021-10-01', '2022-01-28', 3.1, 750.00);

(8, 5, '2021-11-01', '2022-01-28', 3.1, 750.00);

(9, 5, '2021-09-01', '2022-10-28', 3.1, 750.00);

(10, 4, '2021-10-20', '2023-01-28', 3.1, 750.00);

CREATE TABLE Grade\_Content (

Grade\_Content\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Course\_Enrolment\_ID INT NOT NULL,

Description VARCHAR(255),

Weight DECIMAL,

Grade DECIMAL,

Weighted\_Grade DECIMAL,

CONSTRAINT Course\_Enrolment FOREIGN KEY (Course\_Enrolment\_ID) REFERENCES Course\_Enrolment(Course\_Enrolment\_ID)

);

INSERT INTO Grade\_Content (Course\_Enrolment\_ID, Description, Weight, Grade, Weighted\_Grade)

VALUES

(30, 'Assignment 1', 0.1, 90, 9),

(31, 'Midterm Exam', 0.3, 85, 25.5),

(32, 'Lab Report', 0.2, 95, 19),

(33, 'Final Exam', 0.4, 88, 35.2),

(33, 'Quiz 1', 0.15, 75, 11.25),

(33, 'Project', 0.25, 92, 23),

(34, 'Homework', 0.2, 80, 16),

(34, 'Presentation', 0.15, 88, 13.2),

(35, 'Quiz 2', 0.1, 82, 8.2),

(35, 'Term Paper', 0.3, 93, 27.9);

-- Drop the tables in the reverse order of their creation to avoid fk constraints

DROP TABLE Grade\_Content;

DROP TABLE Course\_Enrolment;

DROP TABLE Program\_Enrolment;

DROP TABLE Program;

DROP TABLE Course;

DROP TABLE Department;

DROP TABLE Student;