Project Report

CSE 3110: Database Systems Laboratory

Project name: Procrastination Slayer (Course Management System)

<u>Submitted By</u>

Md. Faysal Mahmud

Roll: 2007068

Year: 3rd, Term: 1st

Introduction

The Procrastination Slayer Project is meant to streamline and improve an academic institution's administrative and educational procedures. This extensive database project focuses on managing course enrolments, examinations, and complete course information, including the teachers in charge of each course. By centralising data on students, instructors, courses, and assessments, this project hopes to increase information availability, educational record accuracy, and academic programme administration efficiency.

Objectives

1. Centralized Student Information Management

To store comprehensive data on students, including personal details, academic majors, and enrolment years, thus providing a holistic view of the student body.

2. Instructor and Course Management

To maintain detailed records of instructors, including their departments and joining dates, alongside a catalogue of courses offered, complete with descriptions and credit details. This will ensure that all course offerings are up-to-date and relevant information is accessible.

3. Course Enrolment System

To implement a robust system for managing course enrolments, allowing students to register for courses each term. The system will track the courses students are enrolled in, facilitating course planning and resource allocation.

4. Assessment and Performance Tracking

To record and manage student assessments, scores, and GPAs for each course they are enrolled in. This feature will support instructors in monitoring student progress and providing feedback, while also helping students understand their academic standing.

5. Reporting and Analytics

To enable the generation of reports and analytics on student performance, course popularity, instructor workload, and other critical metrics. These insights will assist administrators in decision-making and strategic planning.

6. User-Friendly Interface

To develop an intuitive interface that allows students, instructors, and administrators to easily navigate and manage their information and tasks within the system.

Database Design & Planning

The database has been intelligently built to help academic institutions manage academic records, course enrolments, and assessment data more efficiently. This relational database structure promotes data integrity and retrieval convenience by defining clear associations between tables representing students, teachers, courses, enrolments, and assessments.

Table Descriptions:

1. Students

Holds detailed records of each student, including their name, email, date of birth, major, and year of enrolment. This table serves as the central repository for student demographics and academic specializations.

2. Instructors

Contains information on each instructor, such as their name, department, email, and the date they joined the institution.

3. Courses

Lists all courses offered, detailing course codes, names, the instructor responsible, course content descriptions, and credit values. This table is key to managing the academic offerings and ensuring that course descriptions are linked to the correct instructors.

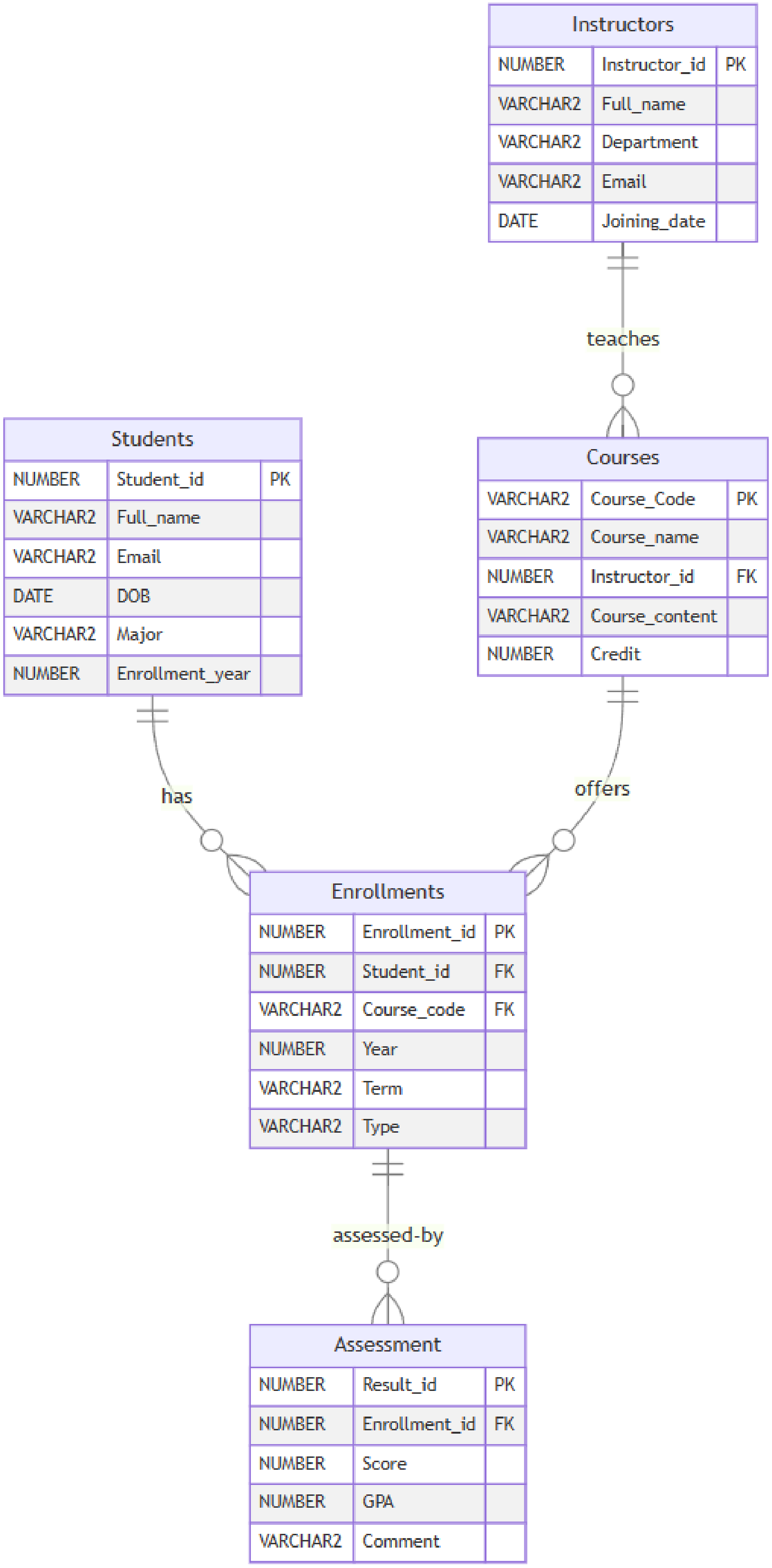
4. Enrolments

Manages the registration of students for specific courses each term, including data on the academic year, term, and type of enrolment. It connects students to the courses they are taking, facilitating course management and attendance tracking.

5. Assessment

Tracks the assessment outcomes for each enrolled course, storing scores, GPAs, and instructors' comments on student performance. This table is essential for academic performance evaluation and feedback.

ENTITY RELATIONSHIP DIAGRAM (ERD)



Instructors		
NUMBER	Instructor_id	PK
VARCHAR2	Full_name	
VARCHAR2	Department	
VARCHAR2	Email	
DATE	Joining_date	

Students		
NUMBER	Student_id	PK
VARCHAR2	Full_name	
VARCHAR2	Email	
DATE	DOB	
VARCHAR2	Major	
NUMBER	Enrollment_year	

Courses		
VARCHAR2	Course_Code	PK
VARCHAR2	Course_name	
NUMBER	Instructor_id	FK
VARCHAR2	Course_content	
NUMBER	Credit	

Enrollments		
NUMBER	Enrollment_id	PK
NUMBER	Student_id	FK
VARCHAR2	Course_code	FK
NUMBER	Year	
VARCHAR2	Term	
VARCHAR2	Type	

Assessment		
NUMBER	Result_id	PK
NUMBER	Enrollment_id	FK
NUMBER	Score	
NUMBER	GPA	
VARCHAR2	Comment	

SQL

Table Creation – DDL

```
-- Creating the Students table
CREATE TABLE Students (
    Student_id NUMBER PRIMARY KEY,
    Full_name VARCHAR2(100),
    Email VARCHAR2(100),
    DOB DATE,
    Major VARCHAR2(100),
    Enrollment_year NUMBER
);

-- Creating the Instructors table
CREATE TABLE Instructors (
    Instructor_id NUMBER PRIMARY KEY,
    Full_name VARCHAR2(100),
    Department VARCHAR2(100),
    Email VARCHAR2(100),
    Joining_date DATE
);

-- Creating the Courses table
CREATE TABLE Courses (
    Course_Code VARCHAR2(10) PRIMARY KEY,
    Course_name VARCHAR2(100),
    Instructor_id NUMBER,
    Course_content VARCHAR2(400),
    Credit NUMBER,
    CONSTRAINT fk_instructor
        FOREIGN KEY (Instructor_id)
        REFERENCES Instructors(Instructor_id) ON DELETE CASCADE
);

-- Creating the Enrollments table
CREATE TABLE Enrollments (
    Enrollment_id NUMBER PRIMARY KEY,
    Student_id NUMBER,
    Course_code VARCHAR2(10),
    Year NUMBER,
    Term VARCHAR2(20),
    Type VARCHAR2(20),
    CONSTRAINT fk_student
```

```
        FOREIGN KEY (Student_id)
        REFERENCES Students(Student_id) ON DELETE CASCADE,
    CONSTRAINT fk_course
        FOREIGN KEY (Course_code)
        REFERENCES Courses(Course_Code) ON DELETE CASCADE
);

-- Creating the Assessment table
CREATE TABLE Assessment (
    Result_id NUMBER PRIMARY KEY,
    Enrollment_id NUMBER,
    Score NUMBER,
    GPA NUMBER,
    Comments VARCHAR2(200),
    CONSTRAINT fk_enrollment
        FOREIGN KEY (Enrollment_id)
        REFERENCES Enrollments(Enrollment_id) ON DELETE CASCADE
);

-- View the tables
SELECT * FROM Students;
SELECT * FROM Instructors;
SELECT * FROM Courses;
SELECT * FROM Enrollments;
SELECT * FROM Assessment;
```

Sample Queries – DML

```
-- 1. The Number of Students enrolled in each course
SELECT c.Course_name, COUNT(e.Student_id) AS Student_Count
FROM Courses c
JOIN Enrollments e ON c.Course_Code = e.Course_code
GROUP BY c.Course_name;

-- 2. The Latest Enrolled Student
SELECT s.Full_name, s.Major
FROM Students s
WHERE s.Enrollment_year = (SELECT MAX(Enrollment_year) FROM Students);

-- 3. Maximum and Minimum GPA
SELECT MAX(GPA) AS HighestGPA, MIN(GPA) AS LowestGPA
FROM Assessment;
```

Sample PL/SQL

```
-- Trigger to validate the Date of Birth
CREATE OR REPLACE TRIGGER trg_validate_dob
BEFORE INSERT ON Students
FOR EACH ROW
BEGIN
  IF :NEW.DOB > SYSDATE THEN
    RAISE_APPLICATION_ERROR(-20001, 'Date of birth cannot be in the future.');
  END IF;
END;
```

Discussion

The Procrastination Slayer database project addresses important demands in educational institution management by providing an organised and efficient approach to academic and administrative data processing. Throughout the project, the database's design and implementation were carefully considered to ensure that it not only met functional needs but also improved the user experience for students, instructors, and administrators.

The project's key features included building strong links between diverse entities such as students, courses, and teachers. These linkages allow for complicated queries that can reveal information about course enrolments, instructor workloads, and student performance. Furthermore, the database design prioritises scalability and security, guaranteeing that it can support future development while protecting sensitive data.

The primary challenges encountered during the project were guaranteeing data integrity. Addressing these difficulties involved implementing solutions such as enforcing foreign key limitations.

Conclusion

The database project has successfully delivered a comprehensive system capable of storing extensive information for students, courses, instructors, and assessments. As a result, it makes a substantial contribution to expediting educational processes, from enrollment to assessment. The solution not only helps educational institutions operate more efficiently, but it also serves as a platform for increased communication and data-driven decision-making.

Overall, the study demonstrates how well-designed database systems can alter school administration and improve the academic experience.