ASSIGNMENT 5

CSA0593

-19232489

RAKINA S

QUESTION:

University Management System Database with Student Information Management

- Design tables for students, courses, registrations, grades, faculty, and departments.

- Implement stored procedures for student registration, course registration, and grade updates.

- Develop SQL queries for reporting student transcripts, course enrollment, and faculty workload.

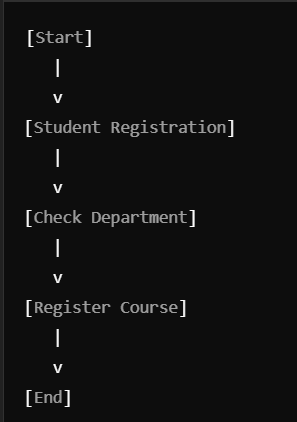
- Implement ACID properties to ensure reliable data management.

ANSWER:

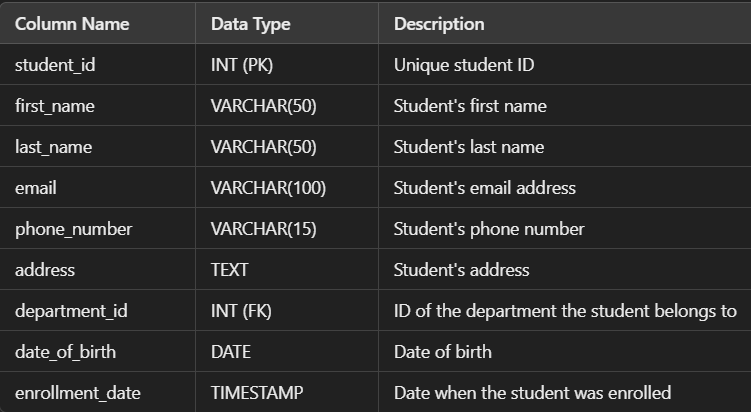
A University Management System Database with Student Information Management is designed to efficiently manage student data, academic records, and administrative processes. The database consists of multiple tables, including Students, Courses, Departments, Faculty, Grades, and Payments. The Students table stores demographic information, contact details, and academic history. The Courses table stores information about course offerings, credits, and prerequisites.

The Departments table stores information about academic departments, including department heads and faculty members. The Faculty table stores information about instructors, including their qualifications, courses taught, and office hours. The Grades table stores student grades for each course, including GPA calculations. The Payments table tracks tuition payments, fees, and financial aid. Relationships between tables are established using foreign keys, enabling efficient data retrieval and updating.

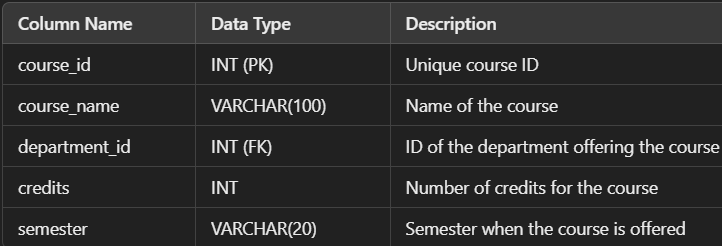
To ensure data security and integrity, the database can be designed with role-based access control, data encryption, and regular backups. By implementing a University Management System Database with Student Information Management, universities can improve student services, enhance academic decision-making, and increase operational efficiency. The database can also be customized to meet the specific needs of different types of universities, such as research universities, liberal arts colleges, and community colleges.



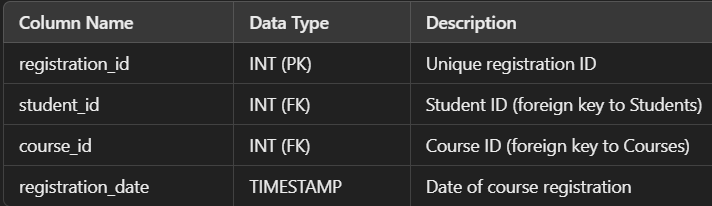
Database Tables Design:



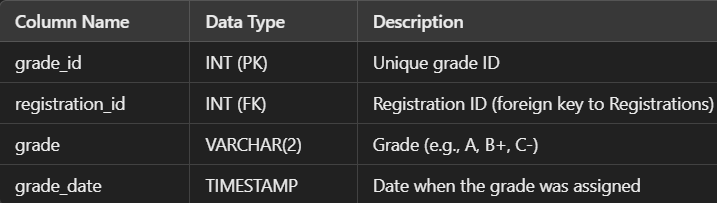
Courses Table:



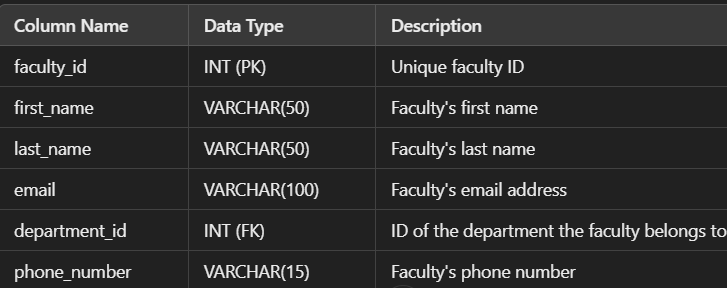
Registrations Table:



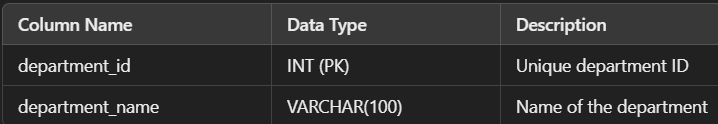
Grades Table:



Faculty Table:

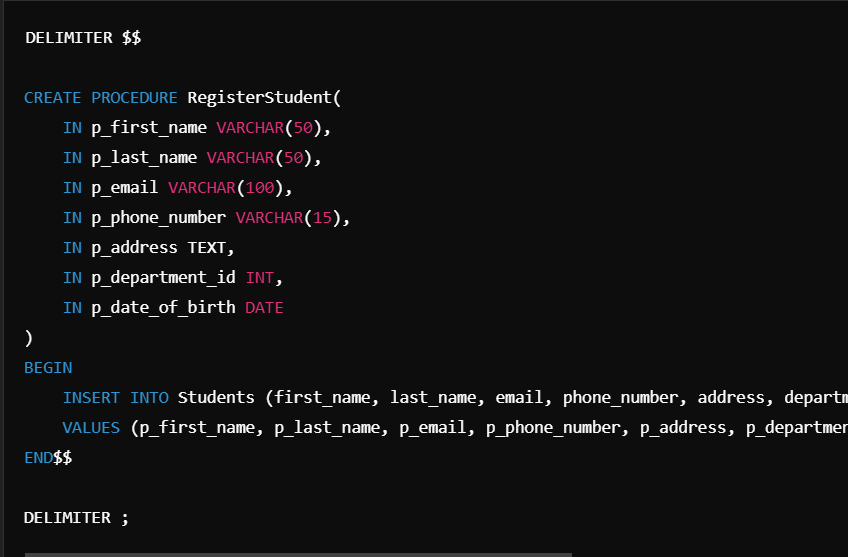


Departments Table:

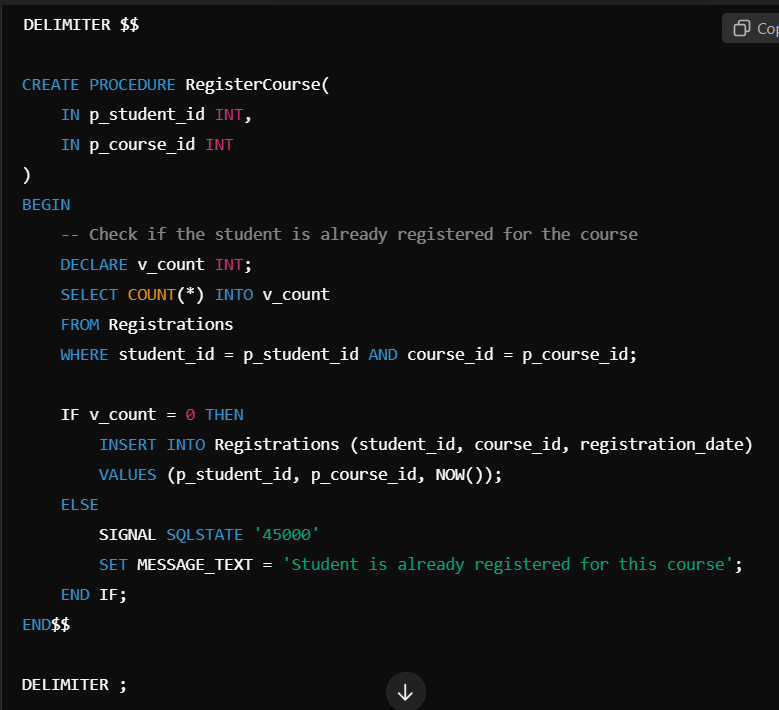


**Stored Procedures**

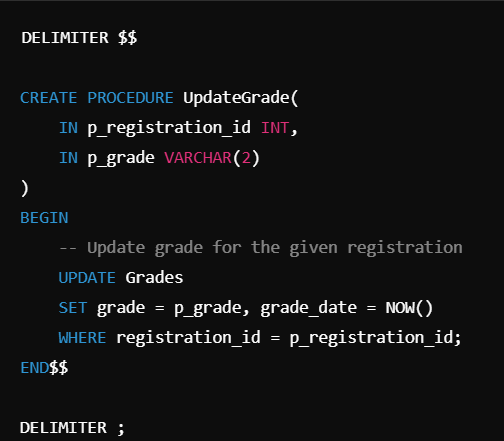
**Student Registration Procedure**

****

**Course Registration Procedure**

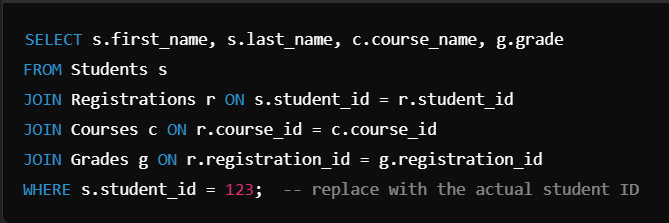
****

**Grade Update Procedure**

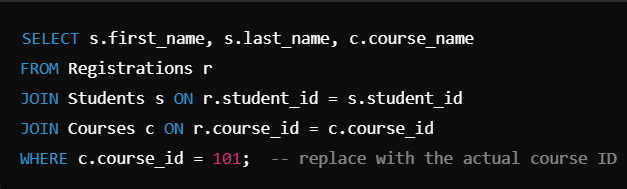
****

**SQL Queries for Reporting**

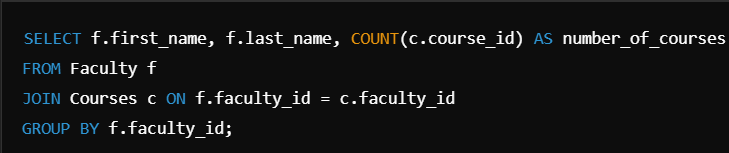
**Student Transcript Report**

****

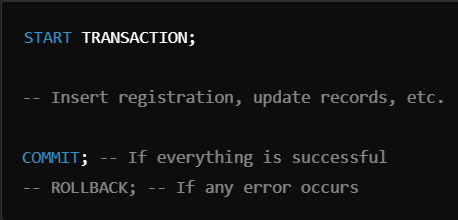
**Course Enrollment Report**

****

**Faculty Workload Report**

****

**Ensuring ACID Properties**

* **Atomicity: The stored procedures (e.g., RegisterStudent, RegisterCourse, and UpdateGrade) ensure that all operations in the transaction are completed successfully, or none are applied (e.g., if a student is already registered for a course, the transaction is aborted).**
* **Consistency: All foreign key constraints ensure data integrity. For example, a student cannot be registered without a valid department, or a course cannot be registered without the student and course existing.**
* **Isolation: Transactions should be used to ensure that no two operations (e.g., student registration or course enrollment) interfere with each other. The SQL queries can be wrapped in a transaction to ensure isolation.**
* ****