ASSIGNMENT – 2

NAME : HEMANTH KUMAR.B

REG NO : 192324238

COURSE : DATABASE MANAGEMENT SYSTEM

COURSE CODE : CSA0593

**SCENERIO:**

Create a database for managing courses, students, enrollment, and grades.

- Model tables for courses, students, enrollments, and grades.

- Write stored procedures for registering and deregistering students.

- Implement triggers to update course enrollments and grade entries.

- Write SQL queries to generate class performance reports.

**Here's the database schema with table definitions and explanations:**

**1. Courses Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| course\_id | INT | Primary key, unique identifier for each course |
| course\_name | VARCHAR(100) | Name of the course |
| course\_description | TEXT | Description of the course |

**2. Students Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| student\_id | INT | Primary key, unique identifier for each student |
| student\_name | VARCHAR(100) | Name of the student |
| student\_email | VARCHAR(100) | Email address of the student |
| student\_address | VARCHAR(255) | Address of the student |

**3. Enrollments Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| enrollment\_id | INT | Primary key, unique identifier for each enrollment |
| student\_id | INT | Foreign key referencing the student's ID |
| course\_id | INT | Foreign key referencing the course's ID |
| enrollment\_date | DATE | Date of enrollment |

**4. Grades Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| grade\_id | INT | Primary key, unique identifier for each grade |
| enrollment\_id | INT | Foreign key referencing the enrollment's ID |
| grade\_value | DECIMAL(5,2) | Grade value (e.g., 95.5) |
| grade\_date | DATE | Date the grade was assigned |

**Explanation of Relationships:**

* **One-to-many relationship between Instructors and Courses:** One instructor can teach multiple courses.
* **Many-to-many relationship between Students and Courses:** One student can enroll in multiple courses, and one course can have multiple students enrolled.
* **One-to-one relationship between Enrollments and Grades:** One enrollment can have one grade.

**Additional Considerations:**

* **Indexes:** Consider adding indexes to frequently queried columns like student\_id, course\_id, and enrollment\_date to improve query performance.
* **Constraints:** Implement appropriate constraints (e.g., primary keys, foreign keys, unique constraints, check constraints) to ensure data integrity.
* **Data Validation:** Validate input data to prevent invalid entries.
* **Security:** Implement security measures to protect sensitive student information.
* **Scalability:** Consider the potential growth of the database and design it to handle a large number of records.
* **Backup and Recovery:** Regularly back up the database and have a disaster recovery plan in place.

By following these guidelines and customizing the schema to your specific needs, you can create a robust and efficient database to manage courses, students, enrollments, and grades.