**Case Scenario: Designing a School Management System**

You are tasked with creating a database system for a school to streamline administrative tasks and improve efficiency.

**Objectives:**

1. Develop a database to manage student records, teacher information, class schedules, and exam results.
2. Ensure data integrity and minimize redundancy.
3. Provide an intuitive interface for easy data management.

**Requirements:**

1. **Students:**
   * Record student details including name, class, date of birth, and contact information.
   * Assign students to classes.
2. **Teachers:**
   * Maintain teacher information such as name, subject, and contact details.
   * Assign teachers to classes.
3. **Classes:**
   * Manage class schedules, room assignments, and teacher allocations.
   * Each class should have a unique identifier, teacher, room number, and schedule.
4. **Exams:**
   * Schedule exams for each class, recording the subject and date.
   * Each exam should be associated with a specific class.
5. **Results:**
   * Record exam results for each student, including marks obtained and grades.
   * Ensure results are linked to both students and exams.

**Implementation Steps:**

1. **Analysis:**
   * Identify key entities (Students, Teachers, Classes, Exams, Results) and their attributes.
   * Understand the relationships between entities.
2. **Normalization:**
   * Normalize attributes to ensure data integrity and minimize redundancy.
   * Remove any data duplication.
3. **Logical Data Model (ER Diagram):**
   * Create an ER diagram depicting entities, attributes, and relationships.
   * Ensure clarity and accuracy in representing the database structure.
4. **SQL Schema Creation:**
   * Implement the logical data model using SQL.
   * Define tables, primary keys, foreign keys, and relationships.
5. **Data Insertion and Basic Queries:**
   * Insert sample data into the database tables.
   * Practice basic SQL queries to retrieve and manipulate data.
6. **Advanced SQL and Optimization:**
   * Implement advanced SQL features such as transactions, indexing, and stored procedures.
   * Optimize database performance where necessary.
7. **Testing and Validation:**
   * Test the database system to ensure it meets all requirements.
   * Validate data integrity and accuracy.
8. **Documentation and Presentation:**
   * Document the database schema, queries, and any custom functions.
   * Prepare a presentation to showcase the school management system and its features.

**Conclusion:**

Through this case scenario, you will gain practical experience in designing and implementing a database system for a school management system. The hands-on approach will enhance your understanding of database management concepts and their real-world applications.

### MySQL Queries for Database Schema Creation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Students | Teachers | Classes | Exams | Results |
| CREATE TABLE Students (  StudentID INT PRIMARY KEY AUTO\_INCREMENT,  Name VARCHAR(100),  Class VARCHAR(50),  DOB DATE,  ContactInfo VARCHAR(100)  ); | CREATE TABLE Teachers (  TeacherID INT PRIMARY KEY AUTO\_INCREMENT,  Name VARCHAR(100),  Subject VARCHAR(50),  ContactInfo VARCHAR(100)  ); | CREATE TABLE Classes (  ClassID INT PRIMARY KEY AUTO\_INCREMENT,  ClassName VARCHAR(50),  TeacherID INT,  RoomNumber VARCHAR(20),  Schedule VARCHAR(100),  FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID)  ); | CREATE TABLE Exams (  ExamID INT PRIMARY KEY AUTO\_INCREMENT,  Subject VARCHAR(50),  Date DATE,  ClassID INT,  FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)  ); | CREATE TABLE Results (  StudentID INT,  ExamID INT,  MarksObtained INT,  Grade CHAR(1),  PRIMARY KEY (StudentID, ExamID),  FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  FOREIGN KEY (ExamID) REFERENCES Exams(ExamID)  ); |

### Additional Queries:

#### Insert Sample Data into Students Table:

|  |
| --- |
| INSERT INTO Students (Name, Class, DOB, ContactInfo) VALUES  ('John Doe', '10th Grade', '2005-05-10', 'john.doe@example.com'),  ('Jane Smith', '11th Grade', '2004-08-15', 'jane.smith@example.com'); |

#### Insert Sample Data into Teachers Table:

|  |
| --- |
| INSERT INTO Teachers (Name, Subject, ContactInfo) VALUES  ('Mr. Johnson', 'Mathematics', 'mr.johnson@example.com'),  ('Ms. Williams', 'Science', 'ms.williams@example.com'); |

#### Insert Sample Data into Classes Table:

|  |
| --- |
| INSERT INTO Classes (ClassName, TeacherID, RoomNumber, Schedule) VALUES  ('Math', 1, 'A101', 'Monday, Wednesday, Friday 10:00 AM - 12:00 PM'),  ('Science', 2, 'B201', 'Tuesday, Thursday 9:00 AM - 11:00 AM'); |

#### Insert Sample Data into Exams Table:

|  |
| --- |
| INSERT INTO Exams (Subject, Date, ClassID) VALUES  ('Mathematics Test', '2024-06-10', 1),  ('Science Exam', '2024-06-15', 2); |

|  |
| --- |
| INSERT INTO Results (StudentID, ExamID, MarksObtained, Grade) VALUES  (1, 1, 85, 'A'),  (2, 2, 78, 'B'); |

#### Insert Sample Data into Results Table:

### ****Database Creation and Table Setup****

### ****WHERE Clause****

1. Write a query to find all students born after January 1, 2010.
2. Write a query to list all teachers who teach 'Mathematics'.
3. Write a query to find all exam results where students scored more than 90 marks.

### ****SELECT Clause****

1. Write a query to retrieve only the names and contact information of all students.
2. Write a query to find all unique subjects taught by teachers.
3. Write a query to retrieve the names of teachers along with their subjects, using aliases for the columns.

### ****Aggregate Functions****

1. Write a query to count the number of students in the **students** table
2. Write a query to calculate the average marks obtained in exams.
3. Write a query to find the total marks obtained by student with ID 1.

### ****GROUP BY and HAVING Clause****

1. Write a query to find the average marks for each class.
2. Write a query to count the number of students in each class.
3. Write a query to list classes that have an average mark greater than 80.

### ****Operators****

1. Write a query to find all students in classes '10A' and '10B'.
2. Write a query to find all exams scheduled between '2024-06-01' and '2024-06-30'.
3. Write a query to find all teachers whose names start with 'J'.

### ****Joins****

1. Write a query to retrieve all students along with their class schedules.
2. Write a query to list all teachers and the classes they teach, including teachers who do not teach any class.
3. Write a query to find the names of students, their class names, and the names of their teachers.

**Note:**

**These questions cover different SQL functionalities and will help students practice and understand the various aspects of querying a database.**