Here’s a new schema with sample data and corresponding practice questions:

Schema

```sql
CREATE TABLE courses (
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    course_name VARCHAR(100) NOT NULL,
    department VARCHAR(50) NOT NULL,
    credits INT NOT NULL
);

CREATE TABLE students (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    enrollment_date DATE NOT NULL
);

CREATE TABLE enrollments (
    enrollment_id INT PRIMARY KEY AUTO_INCREMENT,
    student_id INT NOT NULL,
    course_id INT NOT NULL,
    grade CHAR(1),
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);
```

Sample Data

```
INSERT INTO courses (course_name, department, credits) VALUES
('Database Systems', 'Computer Science', 4),
('Data Structures', 'Computer Science', 3),
('Linear Algebra', 'Mathematics', 3),
('Calculus I', 'Mathematics', 4),
('Marketing 101', 'Business', 3),
('Financial Accounting', 'Business', 4);

INSERT INTO students (first_name, last_name, enrollment_date) VALUES
('John', 'Doe', '2022-09-01'),
('Jane', 'Smith', '2021-09-01'),
('Alice', 'Johnson', '2023-01-15'),
('Bob', 'Brown', '2020-09-01'),
('Charlie', 'Davis', '2022-01-10');

INSERT INTO enrollments (student_id, course_id, grade) VALUES
(1, 1, 'A'),
(1, 2, 'B'),
(2, 3, 'A'),
(2, 4, 'B'),
(3, 1, 'C'),
(3, 5, 'A'),
(4, 6, 'B'),
(5, 2, 'A'),
(5, 3, 'B');
```

Practice Questions

1. Basic Queries

1. Retrieve all courses offered by the Mathematics department.
2. List all students who enrolled after January 1, 2022.
3. Find all enrollments where the grade is 'A'.

2. Relational Operators

1. Retrieve courses with more than 3 credits.
2. Find students whose last name is not 'Smith'.

3. List enrollments where the grade is either 'A' or 'B'.

3. Logical Operators

1. Find students enrolled in courses from the Computer Science department with grades of 'A'.
2. Retrieve courses with credits greater than 3 or offered by the Business department.
3. List students who are not enrolled in any Mathematics courses.

4. Aggregate Functions

1. Count the total number of courses offered.
2. Calculate the average number of credits for courses in the Business department.
3. Find the highest and lowest grades assigned in the enrollments table.

5. Grouping and HAVING

1. Group enrollments by course and count the number of students in each course.
2. Find departments where the average course credits are greater than 3.
3. List courses with more than 2 students enrolled.

6. CASE Statements

1. Categorize courses based on credits:
    - 'High Credit' for courses with 4 credits.
    - 'Medium Credit' for courses with 3 credits.
2. Categorize students based on enrollment date:
    - 'Recent' for students enrolled after 2022.
    - 'Old' for students enrolled before 2022.

7. Common Functions

1. Combine the first and last names of students into a single column called `full_name`.
2. Convert all course names to uppercase.
3. Calculate the number of days each student has been enrolled since their enrollment date.

8. Keyword-Based Operators

1. Find courses offered by the Computer Science or Mathematics departments using the `IN` operator.
2. Exclude courses from the Business department using the `NOT IN` operator.

9. ORDER BY, LIMIT, and OFFSET

1. Retrieve the names of students, sorted by their enrollment date in ascending order.
2. Retrieve the top 3 courses with the highest number of credits.
3. Skip the first 2 students and retrieve the next 3.