SQL PROJECT

- **Build a database with the following tables: Students, Courses, Professors, Grades**

```
CREATE TABLE Students (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    age INT,
    major VARCHAR(100)
);

CREATE TABLE Courses (
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    course_name VARCHAR(100),
    credits INT
);

CREATE TABLE Professors (
    professor_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    department VARCHAR(100)
);

CREATE TABLE Grades (
    grade_id INT PRIMARY KEY AUTO_INCREMENT,
    student_id INT,
    course_id INT,
    professor_id INT,
    grade DECIMAL(3, 2),
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
```

```
    FOREIGN KEY (course_id) REFERENCES Courses(course_id),
    FOREIGN KEY (professor_id) REFERENCES Professors(professor_id)
);
```

- **Create a script that populates all of the database tables with sample data**

```
INSERT INTO Students (name, age, major) VALUES
('Alice Johnson', 20, 'Computer Science'),
('Bob Smith', 22, 'Mathematics'),
('Charlie Brown', 21, 'Physics');

INSERT INTO Courses (course_name, credits) VALUES
('Database Systems', 4),
('Linear Algebra', 3),
('Quantum Mechanics', 4);

INSERT INTO Professors (name, department) VALUES
('Dr. Sarah Connor', 'Computer Science'),
('Dr. John Nash', 'Mathematics'),
('Dr. Richard Feynman', 'Physics');

INSERT INTO Grades (student_id, course_id, professor_id, grade) VALUES
(1, 1, 1, 3.5), -- Alice Johnson, Database Systems, Dr. Connor
(2, 2, 2, 4.0), -- Bob Smith, Linear Algebra, Dr. Nash
(3, 3, 3, 3.8), -- Charlie Brown, Quantum Mechanics, Dr. Feynman
(1, 2, 2, 3.9), -- Alice Johnson, Linear Algebra, Dr. Nash
(2, 1, 1, 3.6); -- Bob Smith, Database Systems, Dr. Connor
```

- **The average grade that is given by each professor**

```
SELECT
   P.name AS professor_name,
   AVG(G.grade) AS average_grade
FROM
   Grades G
JOIN
   Professors P ON G.professor_id = P.professor_id
GROUP BY
   P.professor_id;
```

- **The top grades for each student**

```
SELECT
   S.name AS student_name,
   MAX(G.grade) AS top_grade
FROM
   Grades G
JOIN
   Students S ON G.student_id = S.student_id
GROUP BY
   S.student_id;
```

- **Sort students by the courses that they are enrolled in**

```
SELECT
   S.name AS student_name,
   C.course_name
FROM
   Grades G
JOIN
   Students S ON G.student_id = S.student_id
JOIN
   Courses C ON G.course_id = C.course_id
ORDER BY
   S.name, C.course_name;
```

- **Create a summary report of courses and their average grades, sorted by the most challenging course (course with the lowest average grade) to the easiest course**

```
SELECT
   C.course_name,
   AVG(G.grade) AS average_grade
FROM
   Grades G
JOIN
   Courses C ON G.course_id = C.course_id
GROUP BY
   C.course_id
ORDER BY
   average_grade ASC;
```

- **Finding which student and professor have the most courses in common**

```
SELECT
   S.name AS student_name,
   P.name AS professor_name,
   COUNT(*) AS courses_in_common
FROM
   Grades G
JOIN
   Students S ON G.student_id = S.student_id
JOIN
   Professors P ON G.professor_id = P.professor_id
GROUP BY
   S.student_id, P.professor_id
ORDER BY
   courses_in_common DESC
LIMIT 1;
```