

간단한 **학사관리 시스템**을 구축하려고 합니다.

고객과의 상담을 통해서 관리할 필요가 있는 다음 정보들을 파악 하였습니다.

학생(student),학과(department),교수(professor),개설과목(course),수강(student_course)

학생은 학번, 이름, 키, 학과코드로 이루어져 있습니다.

학과는 학과코드, 학과명으로 이루어져 있습니다.

교수는 교수코드, 교수 명, 학과코드로 이루어져 있습니다.

개설과목은 과목코드, 과목명, 교수코드, 시작일, 종료일로 이루어져 있습니다.

수강은 학번, 과목코드로 이루어져 있습니다.

학과와 학생은 1:N 관계

학과는 많은 학생을 가질 수 있다.

학생은 한 학과의 소속된다.

학과와 교수는 1:N 관계

학과는 많은 교수를 가질 수 있다.

교수는 한 학과의 소속 된다.

교수와 개설과목은 1:N 관계

교수는 많은 과목을 가르칠 수 있다.

과목은 강의하는 교수 한 명이 지정된다.

과목과 학생은 M:N 관계

과목은 수강하는 많은 학생을 가진다.

학생은 많은 과목을 수강할 수 있다.

관계형 데이터베이스는 M:N(다:다) 관계는 허용하지 않는다. 과목과 학생의 M:N 관계를 해소하기 위해서는 관계테이블이라는 역할을 수행할 수강이라는 테이블을 추가한다.

- 과목과 수강은 1:N 관계

- 학생과 수강은 1:M 관계

```
-- Drop the existing tables (if they exist)
DROP TABLE IF EXISTS Student_Course;
DROP TABLE IF EXISTS Course;
DROP TABLE IF EXISTS Professor;
DROP TABLE IF EXISTS Student;
DROP TABLE IF EXISTS Department;

-- Create Department table
CREATE TABLE Department (
    department_code INT PRIMARY KEY,
    department_name VARCHAR(50)
);

-- Create Student table
CREATE TABLE Student (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    height DECIMAL(5,2),
    department_code INT
);

-- Create Professor table
CREATE TABLE Professor (
    professor_code INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    department_code INT
);

-- Create Course table
CREATE TABLE Course (
    course_code INT PRIMARY KEY AUTO_INCREMENT,
    course_name VARCHAR(50),
    professor_code INT,
    start_date DATE,
    end_date DATE
);
```

```

-- Create Student_Course table
CREATE TABLE Student_Course (
  id INT PRIMARY KEY AUTO_INCREMENT,
  student_id INT,
  course_code INT
);
-- Add foreign key constraints using ALTER TABLE
ALTER TABLE Student
ADD CONSTRAINT FK_Student_Department
FOREIGN KEY (department_code) REFERENCES Department(department_code);

ALTER TABLE Professor
ADD CONSTRAINT FK_Professor_Department
FOREIGN KEY (department_code) REFERENCES Department(department_code);

ALTER TABLE Course
ADD CONSTRAINT FK_Course_Professor
FOREIGN KEY (professor_code) REFERENCES Professor(professor_code);

ALTER TABLE Student_Course
ADD CONSTRAINT FK_Student_Course_Student
FOREIGN KEY (student_id) REFERENCES Student(student_id);

ALTER TABLE Student_Course
ADD CONSTRAINT FK_Student_Course_Course
FOREIGN KEY (course_code) REFERENCES Course(course_code);

```

SQL

테스트 데이터 생성

```

-- 학과
-- Insert data into Department table
INSERT INTO Department (department_code, department_name) VALUES
(1, '수학과'),
(2, '국어문학과'),
(3, '정보통신학과'),
(4, '모바일공학과');

```

-- 학생

-- Insert data into Student table

```
INSERT INTO Student (name, height, department_code) VALUES
('가길동', 170.5, 1),
('나길동', 165.2, 1),
('다길동', 180.0, 2),
('라길동', 175.8, 3),
('마길동', 160.7, 4),
('바길동', 168.3, 1),
('사길동', 172.1, 2),
('아길동', 175.0, 1);
```

-- 교수

-- Insert data into Professor table

```
INSERT INTO Professor (name, department_code) VALUES
('교수 교수', 1),
('나 교수', 1),
('다 교수', 2),
('빌 게이즈', 3),
('스티브 잡스', 4);
```

-- 개설과목

-- Insert data into Course table

```
INSERT INTO Course (course_name, professor_code, start_date, end_date) VALUES
('교양 영어', 1, '2023-07-01', '2023-08-15'),
('데이터베이스 입문', 2, '2023-07-01', '2023-08-31'),
('회로이론', 3, '2023-07-15', '2023-09-15'),
('공학수학', 4, '2023-07-15', '2023-09-30'),
('객체지향 프로그래밍', 5, '2023-07-01', '2023-08-31');
```

-- 수강

-- Insert data into Student_Course table

```
INSERT INTO Student_Course (student_id, course_code) VALUES
(1, 1),
(2, 1),
(3, 2),
(4, 3),
(5, 4),
(6, 5),
(7, 5);
```

	department_code	department_name
▶	1	수학과
	2	국어문학과
	3	정보통신학과
	4	모바일공학과

	student_id	name	height	department_code
▶	1	가길동	170.50	1
	2	나길동	165.20	1
	3	다길동	180.00	2
	4	라길동	175.80	3
	5	마길동	160.70	4
	6	바길동	168.30	1
	7	사길동	172.10	2
	8	아길동	175.00	1

	professor_code	name	department_code
▶	1	교수 교수	1
	2	나 교수	1
	3	다 교수	2
	4	빌 게이츠	3
	5	스티브 잡스	4

	course_code	course_name	professor_code	start_date	end_date
▶	1	교양 영어	1	2023-07-01	2023-08-15
	2	데이터베이스 입문	2	2023-07-01	2023-08-31
	3	회로이론	3	2023-07-15	2023-09-15
	4	공학수학	4	2023-07-15	2023-09-30
	5	객체지향 프로그래밍	5	2023-07-01	2023-08-31

	id	student_id	course_code
▶	1	1	1
	2	2	1
	3	3	2
	4	4	3
	5	5	4
	6	6	5
	7	7	5

1. 엔티티(테이블, 릴레이션, 테이블, 객체)들을 등록한다.
2. 해당 엔티티의 고유 프로퍼티(필드, 속성)를 등록한다.
3. 엔티티 프로퍼티의 자료형을 설정한다.
4. 엔티티의 PK를 설정한다.
5. 엔티티들의 관계를 맺어 외부참조 키(FK)를 추가한다.

Physical Model(물리적인 모델)

