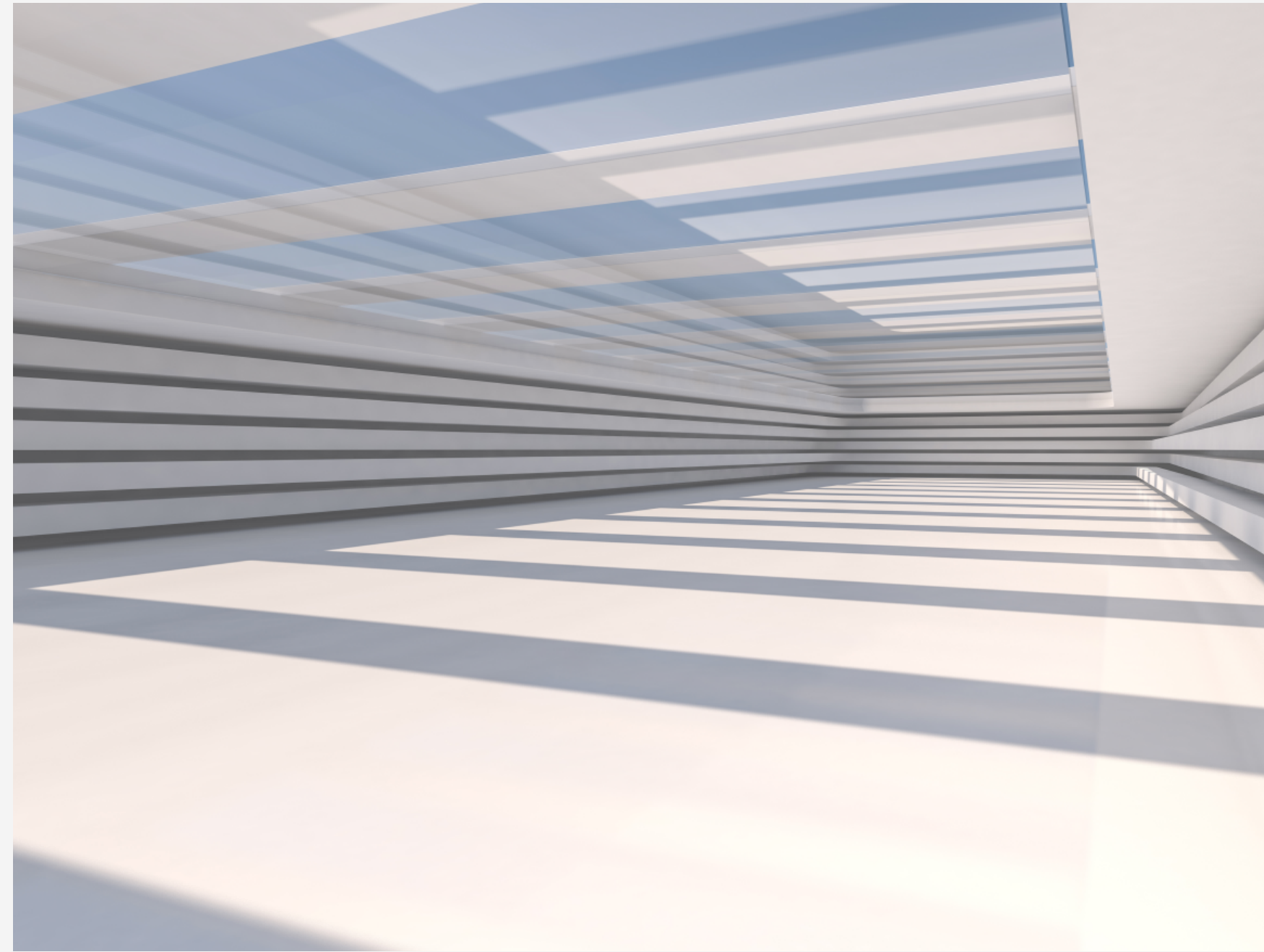


# 데이터 정규화

“

---

제 1 ~ 3 정규형 과정을 통해 데이터를 정규화해보자!



# CONTENTS

## 목 차

정규화 과정 - 데이터를 과정에 맞게 정규화 해보자 -	p.01
--------------------------------	------

테이블 생성과 제 1 정규형	p.02
-----------------	------

제 2 정규형 - 학생 정보	p.03
-----------------	------

제 2 정규형 - 학과 정보, 강좌 정보	p.04
------------------------	------

제 3 정규형	p.05
---------	------

결과 - ERD	p.06
----------	------

결과 - Join	p.07
-----------	------

# 정규화 과정

데이터를 과정에 맞게 정규화 해보자

## 제 1 정규형

릴레이션에 속한 모든 속성의 도메인이 원자 값으로만 이루어져 있으면 제 1 정규형을 만족한다.

## 제 2 정규형

## 제 2 정규형

릴레이션이 제 1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 완전함수 종속되면 제 2 정규형에 속한다.

## 제 3 정규형

테이블이 제 2 정규형을 만족하고 테이블 내의 모든 속성이 기본 키에만 의존하며 다른 후보 키에 의존하지 않으면 제 3 정규형에 속한다.

## 결과

정규형 과정을 거치므로서 중복된 데이터를 허용하지 않고 무결성(Integrity)을 유지할 수 있으며, DB의 저장 용량 역시 줄일 수 있다

# 테 이 블 생 성 & 제 1 정 규 형

테이블 생성 sql문

```
CREATE TABLE `student` (
  `s_idx` int(11) NOT NULL COMMENT '학생 번호',
  `s_name` varchar(100) NOT NULL COMMENT '학생 이름',
  `s_addr` varchar(100) NOT NULL COMMENT '학생 주소',
  `s_subject` varchar(20) NOT NULL COMMENT '학과',
  `s_room` varchar(100) NOT NULL COMMENT '학과 사무실',
  `s_lecture` varchar(100) NOT NULL COMMENT '강좌 이름',
  `s_place` varchar(100) NOT NULL COMMENT '강의실',
  `s_score` float NOT NULL COMMENT '성적'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='학생수강성적'
```

현재 테이블 안의 값들이 모두 원자값이므로 제 1 정규형을 만족한다.

	123 s_idx 🔍	ABC s_name 🔍	ABC s_addr 🔍	ABC s_subject 🔍	ABC s_room 🔍	ABC s_lecture 🔍	ABC s_place 🔍	123 s_score 🔍
1	501	박지성	영국 맨체스터	컴퓨터과	공학관101	데이터베이스	공학관110	3.5
2	401	김연아	대한민국 서울	체육학과	체육관101	데이터베이스	공학관110	4
3	402	장미란	대한민국 강원도	체육학과	체육관101	스포츠경영학	체육관103	3.5
4	502	추신수	미국 클리블랜드	컴퓨터과	공학관101	자료구조	공학관111	4
5	501	박지성	영국 맨체스터	컴퓨터과	공학관101	자료구조	공학관111	3.5

## 제 2 정규형 - 학생 정보

- \* 완전 함수 종속: 기본키가 아닌 다른 속성에 의해 결정되는 것
- \* 부분 함수 종속: 기본키의 일부만 있어도 기본키 외의 다른 속성이 결정되는 것

### 1. 학생 정보

--> s\_idx(학생 번호), s\_name(학생 이름), s\_addr(학생 주소), s\_subject(학과), s\_lecture(강좌 이름), s\_score(성적)

\*\* 학생 정보에는 학생 번호, 이름, 주소, 소속 학과, 수강하는 강좌 이름과 성적이 들어간다

```
CREATE TABLE `student` (
  `s_idx` int(11) NOT NULL COMMENT '학생 번호',
  `s_name` varchar(100) NOT NULL COMMENT '학생 이름',
  `s_addr` varchar(100) NOT NULL COMMENT '학생 주소',
  `s_subject` varchar(20) NOT NULL COMMENT '학과',
  `s_lecture` varchar(100) NOT NULL COMMENT '강좌 이름',
  `s_score` float NOT NULL COMMENT '성적'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='학생 정보'
```

123 s_idx	ABC s_name	ABC s_addr	ABC s_subject	ABC s_lecture	123 s_score
501	박지성	영국 맨체스터	컴퓨터과	데이터베이스	3.5
401	김연아	대한민국 서울	체육학과	데이터베이스	4
402	장미란	대한민국 강원도	체육학과	스포츠경영학	3.5
502	추신수	미국 클리블랜드	컴퓨터과	자료구조	4
501	박지성	영국 맨체스터	컴퓨터과	자료구조	3.5

## 제 2 정규형 - 학과 정보, 강좌 정보

\* 완전 함수 종속: 기본키가 아닌 다른 속성에 의해 결정되는 것

\* 부분 함수 종속: 기본키의 일부만 있어도 기본키 외의 다른 속성이 결정되는 것

2. 학과 정보 --> s\_subject(학과), s\_room(학과 사무실)

\*\* 학과 정보에는 학과와 학과 사무실이 들어간다

```
CREATE TABLE `subject` (
  `s_subject` varchar(20) NOT NULL COMMENT '학과',
  `s_room` varchar(100) NOT NULL COMMENT '학과 사무실'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='학과 정보'
```

ABC s_subject ↑↓	ABC s_room ↑↓
체육학과	체육관101
컴퓨터과	공학관101

3. 강좌 정보 --> s\_lecture(강좌 이름), s\_place(강의실)

\*\* 강좌 정보에는 강좌 이름과 강의실이 들어간다

```
CREATE TABLE `lecture` (
  `s_lecture` varchar(100) NOT NULL COMMENT '강좌 이름',
  `s_place` varchar(100) NOT NULL COMMENT '강의실'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='강좌 정보'
```

ABC s_lecture ↑↓	ABC s_place ↑↓
데이터베이스	공학관110
스포츠경영학	체육관103
자료구조	공학관111

## 제 3 정규형

- \* 이행적 함수 종속이 있다면 제 3 정규형이라고 할 수 없다.
- \* 이행적 함수 종속: 갑-을의 관계가 아닌 갑-을-병-정의 관계에 놓인 상태를 말한다
- \* 현재 학생 정보가 이행적 함수 종속 상태이다. (학생 이름 - 강좌 이름 - 성적)

따라서 학생 정보를 학생 정보와 성적 정보로 나눈다

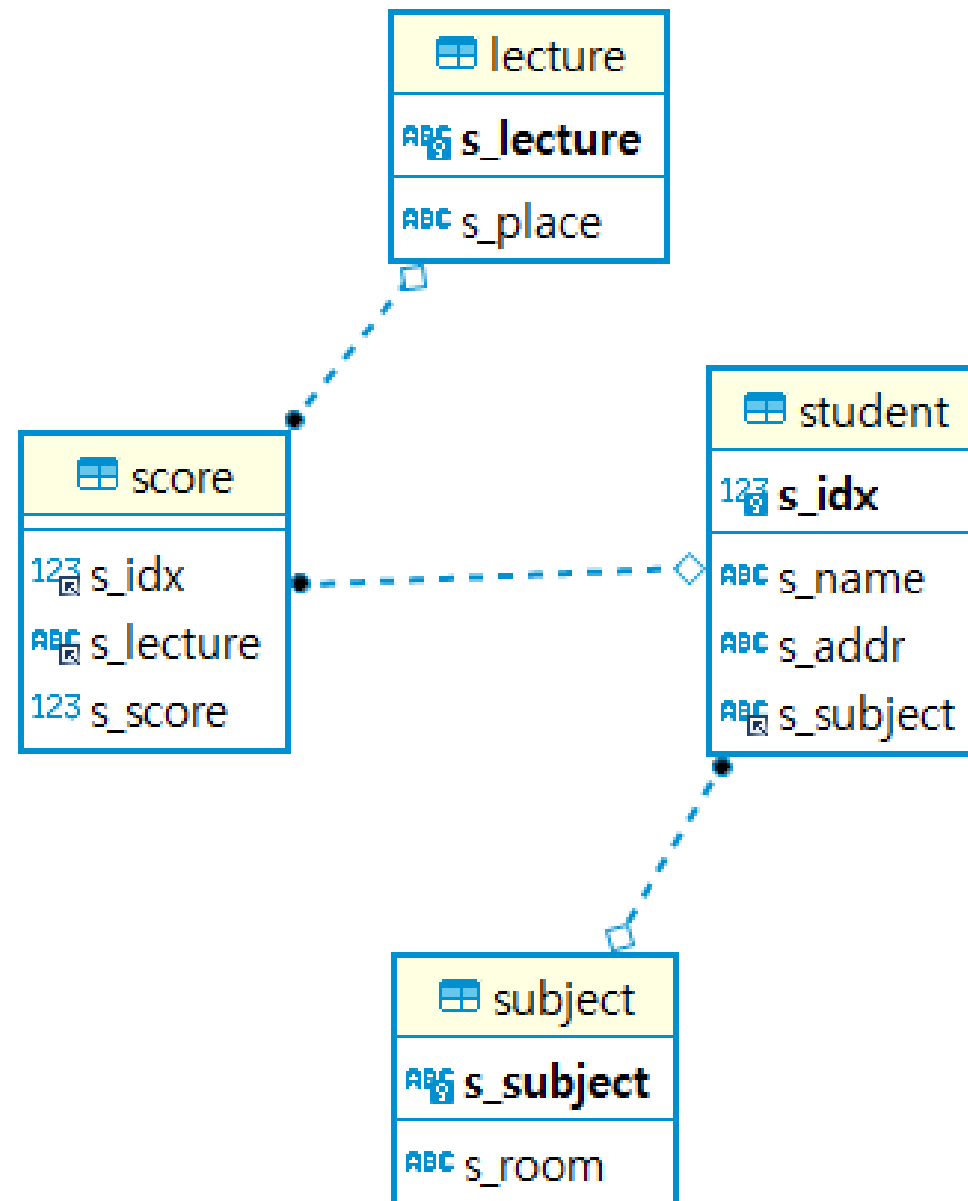
```
CREATE TABLE `student` (
  `s_idx` int(11) NOT NULL COMMENT '학생 번호',
  `s_name` varchar(100) NOT NULL COMMENT '학생 이름',
  `s_addr` varchar(100) NOT NULL COMMENT '학생 주소',
  `s_subject` varchar(20) NOT NULL COMMENT '학과'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='학생 정보'
```

123 s_idx ↕	ABC s_name ↕	ABC s_addr ↕	ABC s_subject ↕
401	김연아	대한민국 서울	체육학과
402	장미란	대한민국 강원	체육학과
501	박지성	영국 맨체스터	컴퓨터과
502	추신수	미국 클리블랜드	컴퓨터과

```
CREATE TABLE `score` (
  `s_idx` int(11) NOT NULL COMMENT '학생 번호',
  `s_lecture` varchar(100) NOT NULL COMMENT '강좌 이름',
  `s_score` float NOT NULL COMMENT '성적'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='성적 정보'
```

123 s_idx ↕	ABC s_lecture ↕	123 s_score ↕
501	데이터베이스	3.5
401	데이터베이스	4
402	스포츠경영학	3.5
502	자료구조	4
501	자료구조	3.5

## 결과 - ERD



ERD (An Entity Relationship Diagram)

시스템의 엔티티들이 무엇이 있는지 어떤 관계가 있는지를 나타내는 다이어그램

PK (Primary Key : 기본 키)

--> lecture.s\_lecture, student.s\_idx, subject.s\_subject

FK (Foreign Key : 외래 키)

--> score.s\_lecture, score.s\_idx, student.s\_subject



# 결과 - Join

1) inner join

```
select s_idx from score inner join lecture
where score.s_lecture = lecture.s_lecture and lecture.s_place = '공학관110';
```

123 s_idx
501
401

--> score의 s\_lecture와 lecture의 s\_lecture가 일치하는 행 중 s\_place가 '공학관110'인 행의 s\_idx를 select

2) left join

```
select * from student
left join score on student.s_idx = score.s_idx
left join subject on student.s_subject = subject.s_subject
left join lecture on score.s_lecture = lecture.s_lecture;
```

- 1. student와 score의 s\_idx가 일치하는 행을 left join
- 2. student와 subject의 s\_subject가 일치하는 행을 left join
- 3. score와 lecture의 s\_lecture가 일치하는 행을 left join

123 s_idx	ABC s_name	ABC s_addr	ABC s_subject	123 s_idx	ABC s_lecture	123 s_score	ABC s_subject	ABC s_room	ABC s_lecture	ABC s_place
401	김연아	대한민국 서울	체육학과	401	데이터베이스	4	체육학과	체육관101	데이터베이스	공학관110
402	장미란	대한민국 강원도	체육학과	402	스포츠경영학	3.5	체육학과	체육관101	스포츠경영학	체육관103
501	박지성	영국 맨체스터	컴퓨터과	501	데이터베이스	3.5	컴퓨터과	공학관101	데이터베이스	공학관110
501	박지성	영국 맨체스터	컴퓨터과	501	자료구조	3.5	컴퓨터과	공학관101	자료구조	공학관111
502	추신수	미국 클리블랜드	컴퓨터과	502	자료구조	4	컴퓨터과	공학관101	자료구조	공학관111