

데이터베이스 정규화 과정

기본 릴레이션

학생수강성적

학생번호	학생이름	주소	학과	학과사무실	강좌이름	강의실	성적
501	박지성	영국 맨체스터	컴퓨터과	공학관101	데이터베이스	공학관 110	3.5
401	김연아	대한민국 서울	체육학과	체육관101	데이터베이스	공학관 110	4.0
402	장미란	대한민국 강원도	체육학과	체육관101	스포츠경영학	체육관 103	3.5
502	추신수	미국 클리블랜드	컴퓨터과	공학관101	자료구조	공학관 111	4.0
501	박지성	영국 맨체스터	컴퓨터과	공학관101	자료구조	공학관 111	3.5

이미 원자 값으로만 구성되어 있기 때문에 해당 기본 릴레이션은 제1 정규형임.
 하지만 기본 릴레이션는 이상현상이 발생하는 릴레이션으로 릴레이션을 분리시켜주어야함.

1. 학생 개인정보 관련 릴레이션로 분리(학생번호, 학생이름, 주소, 학과)
2. 학과 관련 릴레이션으로 분리(학과, 학과사무실)
3. 강좌 관련 릴레이션으로 분리(강좌이름, 강의실)
4. 성적 관련 릴레이션으로 분리(학생번호, 강좌이름, 성적)

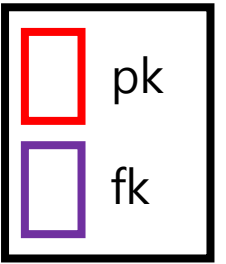
릴레이션 분리

1. 학생 개인정보 릴레이션

학생번호	학생이름	주소	학과
501	박지성	영국 맨체스터	컴퓨터과
401	김연아	대한민국 서울	체육학과
402	장미란	대한민국 강원도	체육학과
502	추신수	미국 클리블랜드	컴퓨터과

2. 학과 관련 릴레이션

학과	학과사무실
컴퓨터과	공학관101
체육학과	체육관101



개체 무결성 : 기본키(pk)를 구성하는 속성은 unique해야하며, null일 수 없다.

참조 무결성 : 외래키(fk)는 참조 릴레이션의 pk 값과 동일해야하며 참조할 수 없는 외래키 값을 가질 수 없다.

3. 강좌 관련 릴레이션

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111

4. 성적 관련 릴레이션

순번	학생번호	강좌이름	성적
1	501	데이터베이스	3.5
2	401	데이터베이스	4.0
3	402	스포츠경영학	3.5
4	502	자료구조	4.0
5	501	자료구조	3.5

학생 개인정보 릴레이션

학생번호	학생이름	주소	학과
501	박지성	영국 맨체스타	컴퓨터과
401	김연아	대한민국 서울	체육학과
402	장미란	대한민국 강원도	체육학과
502	추신수	미국 클리블랜드	컴퓨터과

- 1. 모든 속성값은 원자 값을 가지고 있기 때문에 제1 정규형을 만족
- 2. 학생번호(pk)값에 따라 학생이름, 주소, 학과가 결정
 - > 학생이름, 주소, 학과가 학생번호(pk) 기본키에 완전종속적
 - > 제2 정규형을 만족
- 3. 각각의 컬럼이 이행종속의 관계가 아니므로 제 3정규형을 만족

학과 릴레이션

학과	학과사무실
컴퓨터과	공학관101
체육학과	체육관101
체육학과	체육관101
컴퓨터과	공학관101
컴퓨터과	공학관101

- 1. 모든 속성값은 원자 값을 가지고 있기 때문에 제1 정규형을 만족
- 2. 학과(pk)값에 따라 학과사무실 속성 값이 결정
 - > 학과사무실 속성값이 학과(pk)값에 완전종속적 관계
 - > 제2 정규형을 만족
- 3. 컬럼이 이행종속의 관계를 가질 수 없음.

강좌 릴레이션

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111

1. 모든 속성값은 원자 값을 가지고 있기 때문에 제1 정규형을 만족
2. 강좌이름(pk)값에 따라 강의실 속성값이 결정
-> 강의실 속성값이 강좌이름(pk)값에 완전종속적 관계
-> 제2 정규형을 만족
3. 컬럼이 이행종속의 관계를 가질 수 없음.

성적 릴레이션

순번	학생번호	강좌이름	성적
1	501	데이터베이스	3.5
2	401	데이터베이스	4.0
3	402	스포츠경영학	3.5
4	502	자료구조	4.0
5	501	자료구조	3.5

1. 모든 속성값은 원자 값을 가지고 있기 때문에 제1 정규형을 만족
2. 순번(pk)값에 따라 학생번호, 강좌이름, 성적이 결정
-> 학생번호, 강좌이름, 성적이 순번(pk)값에 완전종속적
-> 제2 정규형을 만족
3. 각각의 컬럼이 이행종속의 관계가 아니므로 제 3정규형을 만족

각 릴레이션 정보

테이블명: student

Engine: InnoDB

Auto Increment: 0

Charset: utf8

Collation: utf8_general_ci

Description:

컬럼명	#	Data Type	Not Null	Auto Increment	Key
student_no	1	varchar(100)	[X]	[]	PRI
student_name	2	varchar(100)	[X]	[]	
address	3	varchar(100)	[X]	[]	
major	4	varchar(100)	[X]	[]	MUL

테이블명: subject

Engine: InnoDB

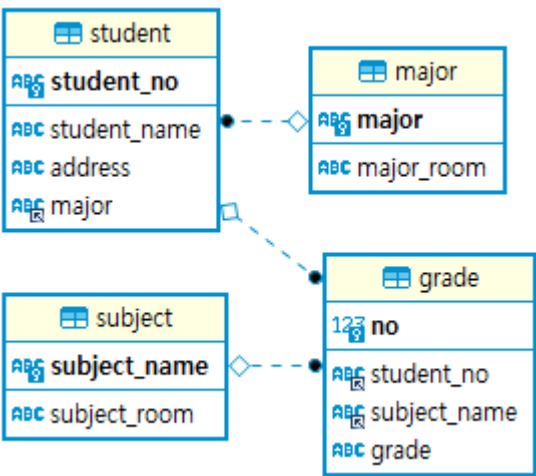
Auto Increment: 0

Charset: utf8

Collation: utf8_general_ci

Description:

컬럼명	#	Data Type	Not Null	Auto Increment	Key
subject_name	1	varchar(100)	[X]	[]	PRI
subject_room	2	varchar(100)	[X]	[]	



테이블명: major

Engine: InnoDB

Auto Increment: 0

Charset: utf8

Collation: utf8_general_ci

Description:

컬럼명	#	Data Type	Not Null	Auto Increment	Key
major	1	varchar(100)	[X]	[]	PRI
major_room	2	varchar(100)	[X]	[]	

테이블명: grade

Engine: InnoDB

Auto Increment: 1

Charset: utf8

Collation: utf8_general_ci

Description:

컬럼명	#	Data Type	Not Null	Auto Increment	Key
no	1	int(11)	[X]	[X]	PRI
student_no	2	varchar(100)	[X]	[]	MUL
subject_name	3	varchar(100)	[X]	[]	MUL
grade	4	varchar(100)	[X]	[]	

SQL문 활용(select)

```
select * from student;
```

```
SELECT * FROM major;
```

```
SELECT * FROM subject;
```

```
SELECT * FROM grade;
```

```
select s.student_no, s.student_name, s.address, m.major, m.major_room  
from student s left join major m on s.major = m.major;
```

```
create view v10 as  
select s.student_no, s.student_name, s.address, m.major, m.major_room  
from student s left join major m on s.major = m.major;
```

```
select g.student_no, g.subject_name, sub.subject_room, g.grade from grade g  
join subject sub on g.subject_name = sub.subject_name;
```

```
create view v11 as  
select g.student_no, g.subject_name, sub.subject_room, g.grade from grade g  
join subject sub on g.subject_name = sub.subject_name;
```

```
select v10.student_no, v10.student_name, v10.address, v10.major, v10.major_room,  
v11.subject_name, v11.subject_room, v11.grade from v10  
join v11 on v10.student_no = v11.student_no;
```

student 1 major 2 subject 3 grade 4 student(+) 5				
select * from student Enter a SQL expression to filter results (use Ctrl+Space)				
	123 student_no	ABC student_name	ABC address	ABC major
1	401	김연아	대한민국 서울	체육학과
2	402	장미란	대한민국 강원도	체육학과
3	501	추신수	미국 클리블랜드	컴퓨터과
4	502	박지성	영국 맨체스터	컴퓨터과

student 1 major 2 subject 3		
SELECT * FROM major Enter a SQL expression to filter results (use Ctrl+Space)		
	ABC major	ABC major_room
1	체육학과	체육관101
2	컴퓨터과	공학관101

student 1 major 2 subject 3 grade 4		
SELECT * FROM subject Enter a SQL expression to filter results (use Ctrl+Space)		
	ABC subject_name	ABC subject_room
1	데이터베이스	공학관 110
2	스포츠경영학	체육관 103
3	자료구조	공학관 111

student 1 major 2 subject 3 grade 4 student(+) 5				
SELECT * FROM grade Enter a SQL expression to filter results (use Ctrl+Space)				
	123 no	123 student_no	ABC subject_name	123 grade
1	1	501	데이터베이스	3.5
2	2	401	데이터베이스	4
3	3	402	스포츠경영학	3.5
4	4	502	자료구조	4
5	5	501	자료구조	3.5

SQL문 활용(join, view)

▼ university

▼ Tables

- > basic
- > grade
- > major
- > student
- > subject

▼ Views

- > v10
- > v11

```
select s.student_no, s.student_name, s.address, m.major, m.major_room  
from student s left join major m on s.major = m.major;
```

```
create view v10 as  
select s.student_no, s.student_name, s.address, m.major, m.major_room  
from student s left join major m on s.major = m.major;
```

```
select g.student_no, g.subject_name, sub.subject_room, g.grade from grade g  
join subject sub on g.subject name = sub.subject name;
```

```
create view v11 as  
select g.student_no, g.subject_name, sub.subject_room, g.grade from grade g  
join subject sub on g.subject_name = sub.subject_name;
```

	student 1	major 2	subject 3	grade 4	student(+) 5	grade(+) 6
SQL	select s.student_no, s.student_name, s.address, m.major, m.major_room					
1	123	401	김연아	대한민국 서울	체육학과	체육관101
2		402	장미란	대한민국 강원도	체육학과	체육관101
3		501	추신수	미국 클리브랜드	컴퓨터과	공학관101
4		502	박지성	영국 맨체스터	컴퓨터과	공학관101

	student 1	major 2	subject 3	grade 4	student(+) 5	grade(+) 6
SQL	select g.student_no, g.subject_name, sub.subject_room, g.grade					
1	123	501	데이터베이스	공학관 110		3.5
2		401	데이터베이스	공학관 110		4
3		402	스포츠경영학	체육관 103		3.5
4		502	자료구조	공학관 111		4
5		501	자료구조	공학관 111		3.5

Join 기능을 활용하여 2개의 테이블을 합칠 수 있다. Join을 통해 만들어진 테이블을 활용하기 위해 view를 만들어 두면 코드를 간략하게 할 수 있는 장점이 있다.

SQL문 활용(join, view)

```
select v10.student_no, v10.student_name, v10.address, v10.major, v10.major_room,
v11.subject_name, v11.subject_room, v11.grade from v10
join v11 on v10.student_no = v11.student_no;
```

student 1 major 2 subject 3 grade 4 student(+) 5 grade(+) 6 v10(+) 7								
select v10.student_no, v10.student_name, v10.address, v10.major, v10.major_room, v11.subject_name, v11.subject_room, v11.grade								
Enter a SQL expression to filter results (use Ctrl+Space)								
	student_no	student_name	address	major	major_room	subject_name	subject_room	grade
1	501	추신수	미국 클리브랜드	컴퓨터과	공학관101	데이터베이스	공학관 110	3.5
2	401	김연아	대한민국 서울	체육학과	체육관101	데이터베이스	공학관 110	4
3	402	장미란	대한민국 강원도	체육학과	체육관101	스포츠경영학	체육관 103	3.5
4	502	박지성	영국 맨체스터	컴퓨터과	공학관101	자료구조	공학관 111	4
5	501	추신수	미국 클리브랜드	컴퓨터과	공학관101	자료구조	공학관 111	3.5

View로 설정된 v10과 v11을 활용하여 조인을 통해 원하는 data를 얻을 수 있다.