

Chap 09. 정규화

정규화의 목적

• 정규화(Normalization)

1. 관계형 데이터베이스를 설계할 때
2. 데이터의 중복을 최소화하여 저장 효율을 높이고 관계에서 **바람직하지 않은 삽입, 삭제, 갱신 이상이 발생하지 않도록 함**
3. 새로운 형태의 데이터가 삽입될 때 **관계를 재구성할 필요성을 줄일 수 있음**
4. 보다 **간단한 관계 연산에 기초하여 검색을 보다 효율적으로 할 수 있음**

정규화의 장점

- 데이터베이스의 일관성을 향상시킬 수 있다.
- 데이터베이스의 확장성을 보장할 수 있다.
- 데이터베이스의 논리적 구조를 견고하게 만들 수 있다

어노말리(이상현상)

- 정규화를 진행하지 않은 경우 발생
- 릴레이션을 조작할 때 이상 (Anomaly)현상이 발생하게 된다.
 1. 삽입이상 (insertion Anomaly)
 2. 삭제이상 (Deletion Anomaly)
 3. 갱신이상 (Update Anomaly)

삽입 이상 (Insertion Anomaly)

- 데이터를 삽입할 때 불필요한 데이터가 함께 삽입되는 현상으로, 제 1정규형에 문제가 있는 경우 발생하는 현상
- 다른 데이터가 존재하지 않아 원하는 데이터를 입력할 수 없는 것이 삽입 이상 현상이다.

이상현상에

- 삽입 이상

<u>STUDENT_ID</u>	STUDENT_NM	DEPARTMENT	<u>COURSE_ID</u>	GRADE
20800399	야봉	컴퓨터공학부	CSE011101	A+
20800399	야봉	컴퓨터공학부	CSE022202	A
20800399	야봉	컴퓨터공학부	CSE033303	B+
21300758	모찌	경영학부	MEC011101	F
21400001	팔빵	기계공학부	POD032939	C+

아직 수업을 하나도 수강하지 않은 학생이 있다고 가정하자. 현재 KEY 를 학번과 과목코드로 지정해 놓았고 기본키로 쓰이는 컬럼은 NULL 이 될 수 없으므로 그 학생은 이 테이블에 추가 될 수가 없다. 굳이 삽입하려면 '미수강' 같은 과목코드를 새로 만들어서 삽입해야 한다.

삭제 이상 (Deletion Anomaly)

- 릴레이션의 한 튜플을 삭제함으로써 연쇄 삭제로 인해 정보의 손실이 발생하는 현상

이상현상예

- 삭제 이상

<u>STUDENT_ID</u>	STUDENT_NM	DEPARTMENT	<u>COURSE_ID</u>	GRADE
20800399	야봉	컴퓨터공학부	CSE011101	A+
20800399	야봉	컴퓨터공학부	CSE022202	A
20800399	야봉	컴퓨터공학부	CSE033303	B+
21300758	모찌	경영학부	MEC011101	F
21400001	팔빵	기계공학부	POD032939	C+

위 테이블에서 모찌는 현재 1개의 과목(MEC011101) 만 수강하고 있다. 모찌가 수강정정기간에 MEC011101 라는 수업을 듣기 싫어져서 drop하는 경우, 위 테이블에 반영하기 위해서는 모찌에 대한 행을 모두 삭제하게 된다. 수강 취소를 반영하기 위해 학생등록정보를 모두 날리게 되는 것이다.

갱신 이상 (Update Anomaly)

- 튜플 중에서 일부 튜플의 값만을 갱신함으로써 정보의 모순성이 발생하는 현상
- 이유는 데이터의 중복

이상현상에

- 갱신 이상

<u>STUDENT_ID</u>	STUDENT_NM	DEPARTMENT	<u>COURSE_ID</u>	GRADE
20800399	야봉	컴퓨터공학부	CSE011101	A+
20800399	야봉	컴퓨터공학부	CSE022202	A
20800399	야봉	컴퓨터공학부	CSE033303	B+
21300758	모찌	경영학부	MEC011101	F
21400001	팔빵	기계공학부	POD032939	C+

야봉이 컴퓨터공학이 싫어서 음악학부로 옮기게 되는 경우 '컴퓨터공학부' 의 갯 수는 과목코드의 개수 만큼 있으므로 모두 '음악학부' 로 변경해주어야 한다. 이때 모두 변경하지 않고 두 개만 바꾸는 경우 야봉은 컴퓨터공학부인지 음악학부인지 알 수 없게 된다.

어노말리(Anomaly)를 해결하는 방법 => 정규화

정규화를 위해선 속성들 간의 관련성을 파악해야 하는데,
이 속성들 간의 관련성을 함수적 종속성이라고 한다.
일반적으로 하나의 릴레이션에는 하나의 함수적 종속만이 존재하도록
정규화 하게 된다.

예제1

<u>STUDENT_ID</u>	STUDENT_NM	DEPARTMENT
20800399	야옹	컴퓨터공학부
21300758	모찌	경영학부
21400001	팔빵	기계공학부

학번에 의해서 학생이름과 학부는 고유하게 구분되므로 학생이름, 학부 속성은 학번에 함수적으로 종속되어 있다고 할 수 있다. 여기서 학번은 결정자, 학생이름과 학부는 종속자가 되며 함수적 종속성은 아래와 같은 기호로 표현할 수 있다.

학번 -> (학생이름, 학부)

주의할 점은 현시점의 속성 값만으로 판단하면 안된다. 속성 값은 계속 변할 수 있는 것이기 때문에 속성 자체가 가지는 특성과 의미를 기반으로 판단해야 한다.

예제2

<u>STUDENT_ID</u>	<u>COURSE_ID</u>	GRADE	STUDENT_NM
20800399	CSE011101	A+	야봉
20800399	CSE022202	A	야봉
20800399	CSE033303	B+	야봉
21300758	MEC011101	F	모찌
21400001	POD032939	C+	팔빵

위 테이블 속성들 간 함수적 종속성은 아래와 같이 정의할 수 있다.

학번 -> 이름

{학번, 과목코드} -> 성적

{학번, 과목코드} -> 이름

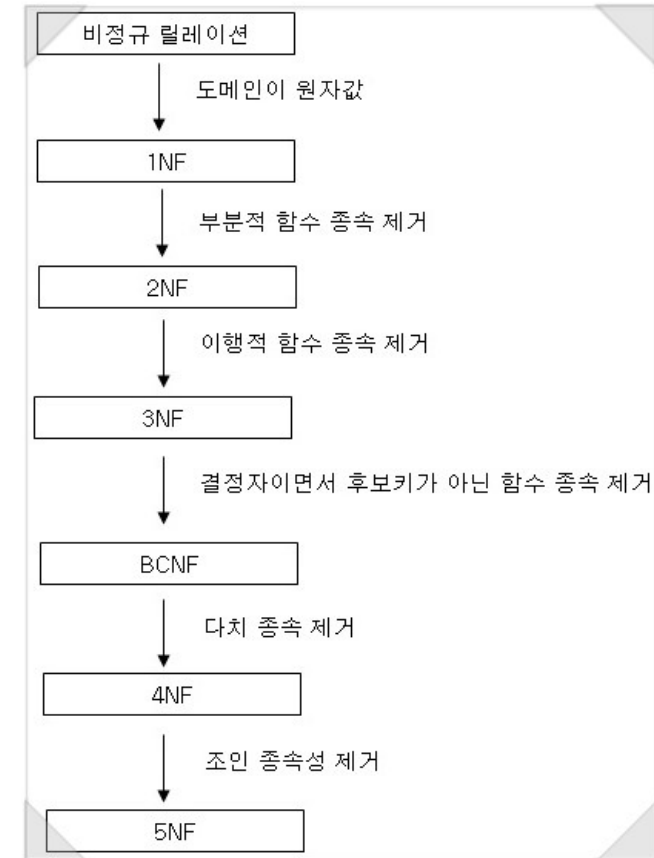
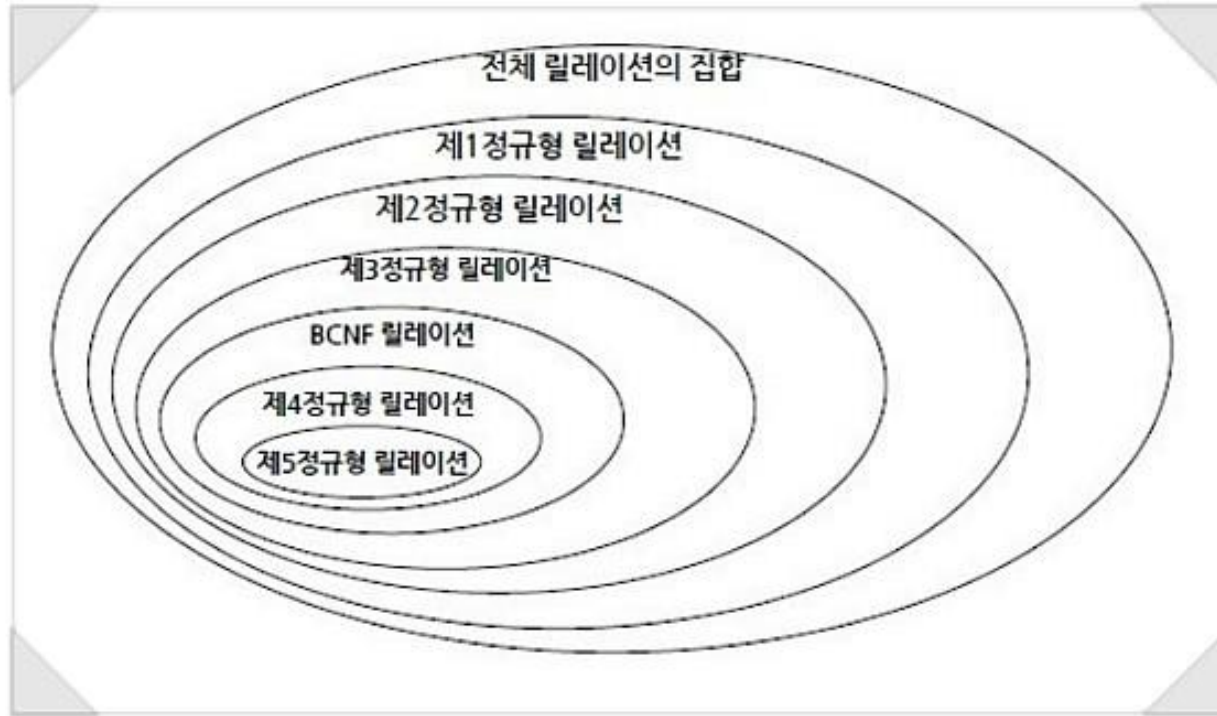
이름을 결정짓는 요소는 학번이다.

성적을 결정짓는 요소는 학번과 과목코드이다.

이름은 학번과 과목코드에 의해서도 고유하게 구분될 수 있다.

이름의 경우 해당하는 함수적 종속성이 두 개다.

정규형의 구조와 단계



제1 정규형

- 제1 정규형은 릴레이션의 모든 속성 값이 원자값을 갖는 경우입니다.
- 예를 들어 고객 취미들(이름, 취미들)이라는 릴레이션에 (추신호, (영화, 음악))이라는 열이 있다고 가정하면 이 속성들이 각각 다른 열로 분해된 릴레이션을 제1 정규형이라 합니다.

고객취미들(이름, 취미들)

이름	취미들
김연아	인터넷
추신수	영화, 음악
박세리	음악, 쇼핑
장미란	음악
박지성	게임



고객취미(이름, 취미)

이름	취미
김연아	인터넷
추신수	영화
추신수	음악
박세리	음악
박세리	쇼핑
장미란	음악
박지성	게임

그림 7-17 속성 값이 원자값을 갖도록 분해

제1정규형

<u>STUDENT_ID</u>	<u>COURSE_ID</u>	GRADE	STUDENT_NM
20800399	CSE011101, CSE022202, CSE033303	A+, A, B+	야봉



<u>STUDENT_ID</u>	<u>COURSE_ID</u>	GRADE	STUDENT_NM
20800399	CSE011101	A+	야봉
20800399	CSE022202	A	야봉
20800399	CSE033303	B+	야봉

제2 정규형 (부분 함수 종속성)

- 제2 정규형은 릴레이션이 제1 정규형을 만족하고, 기본키가 아닌 속성이 기본키에 완전 함수 종속일 때를 의미합니다.
- 여기서 완전 함수 종속이라는 말은 기본키로 묶인 복합키가 존재할 때 복합키(A,B,C)가 모여서 하나의 다른 값(X)를 결정하고 복합키의 부분집합이 결정자가 되면 안된다는 뜻입니다.

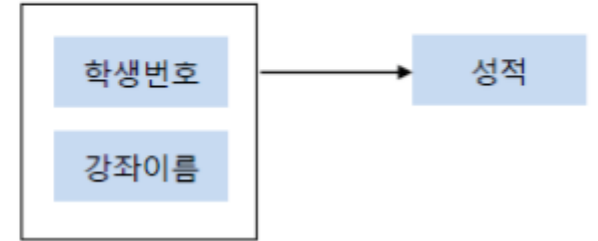
제2 정규형 (부분 함수 종속성)

수강강좌

학생번호	강좌이름	강의실	성적
501	데이터베이스	공학관 110	3.5
401	데이터베이스	공학관 110	4.0
402	스포츠경영학	체육관 103	3.5
502	자료구조	공학관 111	4.0
501	자료구조	공학관 111	3.5

수강

학생번호	강좌이름	성적
501	데이터베이스	3.5
401	데이터베이스	4.0
402	스포츠경영학	3.5
502	자료구조	4.0
501	자료구조	3.5



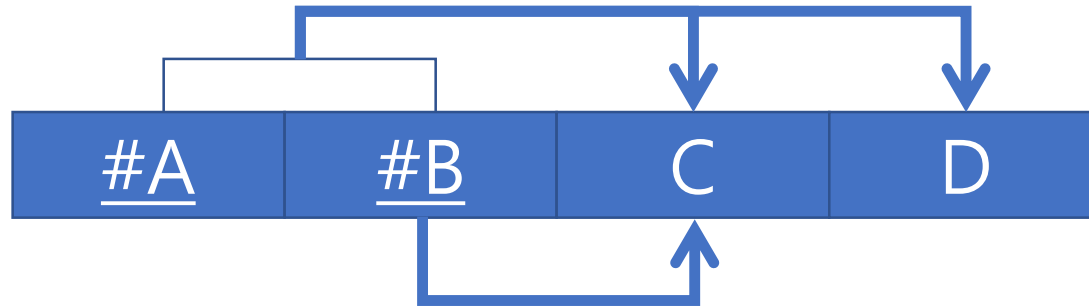
강의실

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111

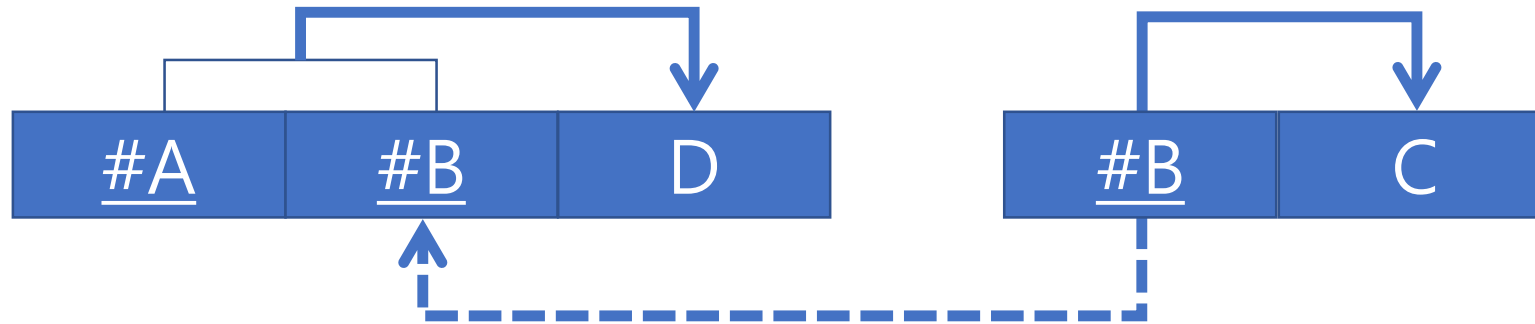


- 예를 들어 그림과 같이 수강강좌 릴레이션이 있고, (학생번호, 강좌이름)의 복합키를 가지고 있다고 가정합시다. 여기서 (501, 데이터베이스)가 모여서 성적이라는 하나의 값을 결정하지만, 강의실의 경우에는 (501, 데이터베이스) 중에서 학생 번호가 없어도 강의실을 결정 할 수 있습니다. 그러므로 이러한 관계를 부분 함수 종속이라고 하며 제2 정규형은 완전 함수 종속을 만족시켜야 하므로 강좌 이름과 강의실을 분리하면 제2 정규형이 만들어 집니다.

제2 정규형 - 부분적 함수 종속



[부분 함수 종속이 존재하는 릴레이션]



[정규화한 릴레이션]

제2 정규형

학생 릴레이션

성적 릴레이션

학번	학부	등록금	학번	과목코드	성적
20800399	컴퓨터공학부	350	20800399	CSE011101	A+
21300758	경영학부	300	20800399	CSE022202	A
21400001	기계공학부	400	20800399	CSE033303	B+
21500399	컴퓨터공학부	350	21300758	MEC011101	F
제 2 정규형을 만족하면 이상현상이 없어질까?			21400001	POD032939	C+
			21500399	CSE011101	D

제3 정규형 - 이행적 함수 종속

- 제 2 정규형에 속하면서,
- 기본키가 아닌 모든 속성이 기본키에 이행적 함수 종속이 되지 않으면 제 3 정규형이다.

이행(移行)적 함수 종속 (Transitive Functional Dependency)

이행은 '움아가는' 이라는 의미로도 쓰인다. 옮길 이, 다닐 행 자를 써서 옮기고 다닌다는 뜻이다.

그래서 이행적 함수 종속이라는 건 삼단논법 같은 관계를 가진 함수 종속이다. X, Y, Z 에 대해 $X \rightarrow Y$ 이고 $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 가 성립한다. 이를 Z 가 X 에 이행적으로 함수 종속되었다고 한다.

제3 정규형 – 이행적 함수 종속

학생 릴레이션

학번	학부	등록금
20800399	컴퓨터공학부	350
21300758	경영학부	300
21400001	기계공학부	400
21500399	컴퓨터공학부	350

학번 -> 학부

학부 -> 등록금

학번 -> 등록금

- 논리적으로 말은 되는데 의미상 뭔가 이상하다. 학부에 따라 등록금이 결정되는 것이지 학번에 따라 결정되는 것은 아니다. 그냥 이걸 둘로 분리 해주면 된다.

제3 정규형 - 이행적 함수 종속

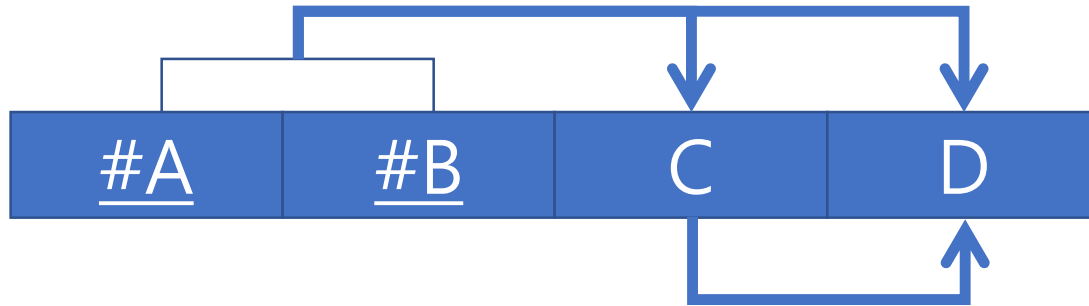
$X \rightarrow Y, Y \rightarrow Z$ 함수적 종속관계로 인해 $X \rightarrow Z$ 의 이행적 함수 종속 관계가 나타나면 $[X, Y], [Y, Z]$ 두 릴레이션으로 분해한다.

이행적 종속 관계를 제거하면 학생, 학부 2개의 릴레이션으로 나눌 수 있다.

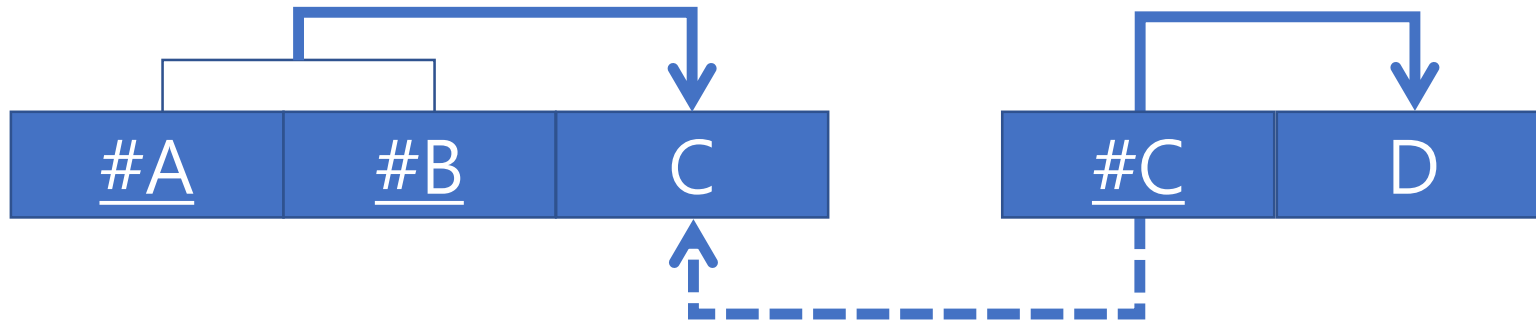
학번	학부
20800399	컴퓨터공학부
21300758	경영학부
21400001	기계공학부
21500399	컴퓨터공학부

학부	등록금
컴퓨터공학부	350
경영학부	300
기계공학부	400

제3 정규형 - 이행적 함수 종속



[3정규화 대상 릴레이션]



[정규화한 릴레이션]

제3 정규형 - 이행적 함수 종속

- 여기까지는 3NF 를 만족하는 릴레이션들은 모두 후보 키가 1개 밖에 없었기 때문에 3NF 를 만족시키는 정규화를 했지만 BCNF 도 만족한다.
- 하지만 후보키가 여러개인 경우에는 3NF를 만족시키지만 이상현상이 발생하는 경우가 있는데, 이를 해결하기 위한 정규형이 보이스-코드 정규형 (BCNF; Boyce-Codd Normal Form)이다. 제3정규형보다 조금 더 엄격한 제약조건을 가지기 때문에 Strong 3NF 라고도 한다.

Quiz.

1. 다음 테이블을 조건에 맞게 정규화를 한다.
 - 한 명의 학생은 복수개의 과목을 수강할 수 있다
 - 한 명의 학생은 한 명의 지도 교수가 있다
 - (학번과 과목코드)는 기본 키이다.

학번	지도교수	학과	과목번호	성적
100	김도식	컴퓨터	C123	A
100	김도식	컴퓨터	C231	B
200	이진수	전자	E111	A
300	최명수	영문학	K123	A
300	최명수	영문학	K322	B
300	최명수	영문학	K421	C
400	박진현	체육학과	Y921	A
400	박진현	체육학과	Y341	B
500	김도식	컴퓨터	C123	C
500	김도식	컴퓨터	C231	A

2. 다음 테이블을 조건에 맞게 정규화를 한다.
 - 한 명의 학생은 복수개의 과목을 수강할 수 있다
 - 한 명의 학생은 한 명의 지도 교수가 있다
 - (학번과 과목코드)는 기본 키이다.

학번	지도교수	학과	과목번호	성적	학생명
100	김도식	컴퓨터	C123	A	김백번
100	김도식	컴퓨터	C231	B	김백번
200	이진수	전자	E111	A	김이백
300	최명수	영문학	K123	A	최삼백
300	최명수	영문학	K322	B	최삼백
300	최명수	영문학	K421	C	최삼백
400	박진현	체육학과	Y921	A	우사백
400	박진현	체육학과	Y341	B	우사백
400	박진현	체육학과	Y125	C	우사백
400	박진현	체육학과	Y423	A	우사백

1번

학생			교수	
학번(외래키)	학과(외래키)		지도교수	학과(기본키)
성적				
학번(기본키)	과목번호	성적		

2번

학생				교수	
학번(외래키)	학생명	학과(외래키)		지도교수	학과(기본키)
	학번(기본키)	과목번호	성적		