2. 정규화 이야기

2.2. 정규화란?

// 정규화는 데이터를 완전히 이해하는 과정. 또한, 속성 간의 부정확한 종속성을 없애는 것을 의미.

2.3. 함수 종속이란?

- 데이터 종속성
  + 함수 종속 // 한 속성의 값을 알면 다른 속성의 값은 저절로 결정되는 두 속성 간의 일종의 제약
    즉, A 속성의 값이 B 속성의 값을 유일하게 식별할 수 있다면, B속성은 A에 함수적으로 종속됐다고 함.
  + 다가 종속 // 4정규형의 기준
  + 조인 종속 // 5정규형의 기준
  + 파생 종속

2.4. 결정자와 종속자

- 결정자 : 속성 간의 종속성을 분석할 때 기준이 되는 값.
- 종속자 : 결정자의 값에 의해 정해질 수 있는 값.
- '속성 Y가 속성 X에 의해 함수적으로 종속된다'
  + X를 결정자, Y를 종속자라고 함.
  + X는 Y를 함수적으로 결정한다고 함.
  + 기호 표기법
    # $X \rightarrow Y$
    # $y = f(x)$
    # X : 결정자, Y : 종속자
- 결정자는 식별자 속성이고 종속자는 비식별자 속성.

2.5. 함수 종속과 폐포

// 결정자는 릴레이션의 모든 속성을 함수적으로 결정하는 최소한의 속성. 이를 후보키라고 함.

- 폐포 : 어떤 속성(X)에 종속됐다고 추론할 수 있는 모든 속성의 집합을 의미.
  // 식별자가 정확한지를 검증하는 도구로 활용할 수 있음.
  + 기호 표기법
    # $X \rightarrow (Y,Z)$ 라면, 폐포는 $X+ = X,Y,Z$ 이와 같이 표기한다.
- 종속성을 추론할 수 있는 규칙
  + $Y \subseteq X$ 이면 $X \rightarrow Y$ 이 성립한다.
  + $X \rightarrow Y$ 이면 $XZ \rightarrow YZ$ 이 성립한다.
  + $X \rightarrow Y$ 이고 $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 이 성립한다.
  + $X \rightarrow YZ$ 이면 $X \rightarrow Y$ 이고 $X \rightarrow Z$ 이 성립한다.
  + $X \rightarrow Y$ 이고 $X \rightarrow Z$ 이면 $X \rightarrow YZ$ 이 성립한다.
  + $X \rightarrow Y$ 이고 $YZ \rightarrow W$ 이면 $XZ \rightarrow W$ 이 성립한다.

2.6. 함수 종속과 정규화

- 릴레이션의 정규화 두 가지 방법
  + 먼저 릴레이션의 키를 도출하는 것
  // 해당 방법 사용 시, 키를 도출하려면 폐포를 알아야하고 폐포를 구하려면 모든 함수 종속을 알아야 하기에 두 번째 방법과 같아짐.
  + 릴레이션에 존재하는 모든 함수 종속을 구하는 것

2.7. 그냥 릴레이션과 비정규형 릴레이션 (P.118)

// '비정규형을 할 지라도 반드시 정규화를 다 수행하고 나서 비정규형을 해야한다'는 내용

2.8. 등산과 정규화 (P.120)

// 앞의 2.7 단락의 내용을 강조

2.9. 정규화를 하면 좋아지는 게 무엇인가? // 크게 완전성과 확장성이 좋아진다.

- 완전성 // 중복 데이터를 제거함으로써 아노말리 발생하지 않게한다.
- 확장성 // 업무가 확장되더라도 변화에 맞게 유연하게 대처가능.
- 데이터 저장 공간의 사용 최소화
- 데이터 모델 단순화

2.10. 아노말리란**?** // 데이터의 이상 현상. 즉, 중복 데이터 때문에 발생하는 이상 현상.

- 아노말리의 종류
  + 업데이트 아노말리 : 릴레이션에서 속성의 값을 업데이트할 때 발생하는 데이터 이상 현상
  + 삭제 아노말리 : 릴레이션에서 인스턴스를 삭제할 때 발생하는 데이터 이상 현상
    // '김길동'에 대한 데이터 삭제 시, 의도치 않은 '타이거즈','엘리펀트' 팀의 데이터가 같이 삭제되는 경우.
  + 삽입 아노말리 : 릴레이션에 새로운 인스턴스를 삽입할 때 발생하는 데이터 이상 현상
    // 선수는 존재하나, 팀이 정해지지 않아 데이터 입력이 안되는 경우.

2.11. 정규형의 종류

- **2.12. 1**정규화와 원자 값

  1정규형은 '모든 속성은 반드시 하나의 값을 가져야한다.'

  다가 속성 : 같은 종류의 값을 여러 개 가지는 속성. (세미콜론이나 콤마로 구분하는 경우)

  복합 속성 : 하나의 속성이 여러 개의 속성으로 분리되는 속성. ex) 날짜(년·월·일), 주소(시·구·동·번지)

  + 2.13. 1정규화의 대상

  # 다가 속성이 사용된 릴레이션
  # 복합 속성이 사용된 릴레이션
  # 유사한 속성이 반복된 릴레이션 ex) 상품번호1,상품번호2 …
  # 중첩 릴레이션 // 하나의 인스턴스 내부에 다시 인스턴스가 존재하는 형태
  # 동일 속성이 여러 릴레이션에 사용된 경우

  + 2.14. 1정규형과 비정규형

  # 비정규형의 특징
  > 업무 요건의 변경에 매우 취약. 즉, 모델의 확장성이 좋지 않음.
  > 인덱스 수가 증가하고, 속성을 종으로 보여주는 화면에 대한 쿼리가 복잡해짐.
  > 반복 속성이 추가될 가능성이 없을 때 사용할 수 있음.
  > 전체 속성 레벨로 관리되므로 해당 데이터의 자식 엔터티를 가질 수 없음.

  # 정규형의 특징
  > 업무 요건의 변경에 유연. 즉, 확장성이 좋은 모델임.
  > 인덱스 수 가 감소하고, 속성을 횡으로 보여주는 화면에 대한 쿼리가 비교적 복잡.
  > 반복 속성이 추가될 가능성이 존재할 때 사용함.
  > 인스턴스 레벨로 관리되므로 데이터의 자식 엔터티를 가질 수 있음.

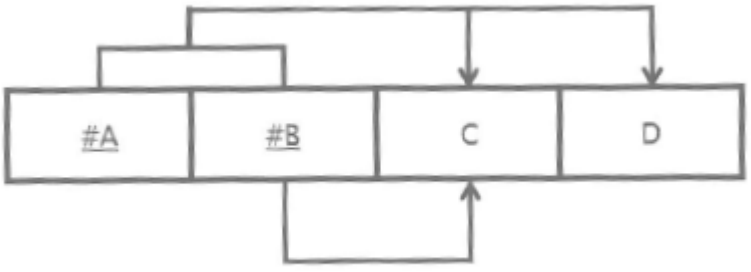
  + 2.15. 반복 속성으로 인한 **1**정규형 위반 사례 **(P.143)**

  // 반복 속성으로 인한 1정규형 위반 사례에 대한 내용

- **2.16. 2**정규형

  2정규형은 '모든 비식별자 속성은 후보 식별자 속성에 완전 함수 종속돼야 한다.'

  즉, 부분 함수 종속으로 말미암아 발생한 중복 데이터를 제거하는 것.



[그림 2.38] 부분 함수 종속이 존재하는 릴레이션

+ 2.17. 2정규형 위반인가**? (P.152)**

//리그대분류, 리그소분류 속성을 예시로 2정규형의 효율적인 활용에 대한 내용

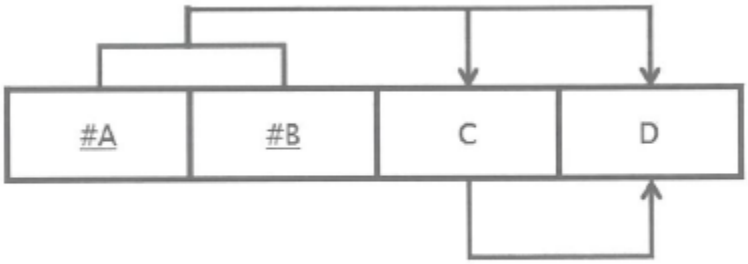
- **2.18. 3**정규형

  3정규형은 '식별자가 아닌 일반 속성 간에는 종속성이 존재하면 안 된다.'

  즉, 비식별자 속성 간에 발생하는 이행적 종속성과 관련 있음.

  X → Y이고 Y → Z이면 X → Z가 성립한다. 이 때 Y는 릴레이션 의 후보 식별자나 후보 식별자의 일부가 아닌 일반

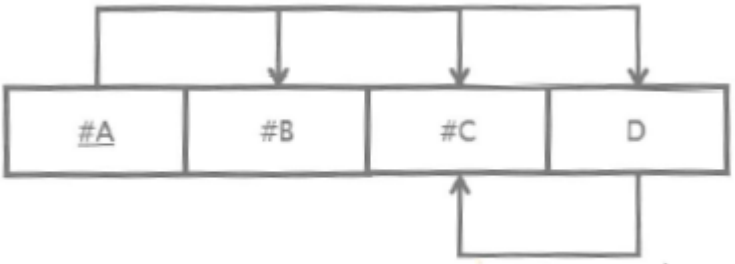
속성이다.



[그림 2.52] 3정규화 대상 릴레이션

- **2.19. BC**정규형

BC정규형은 '모든 결정자는 주 식별자여야 한다.' 3정규형과 다른 점은 '릴레이션에 존재하는 종속자는 후보 식별자가 아니어야 한다'는 점.



[그림 2.59] 후보 식별자가 종속돼 BC정규화 대상인 릴레이션

- **2.20. 4**정규형

4정규형은 *다가 종속 개념이 기반이 되는 정규형.

*다가 종속 : 두 개 이상의 다가 속성끼리 다대다 관계가 발생하여 하나의 다가 속성 값이 다른 다가 속성의 모든 값마다 중복되는 것.

- **2.21. 5**정규형

5정규형은 *조인 종속 개념이 기반이 되는 정규형. PJ정규형(Project-Join Normal Form) 이라고도 불림.

*조인 종속 : 어떤 릴레이션을 분해한 다음 조인해서 다시 원래의 릴레이션으로 복원할 수 있을 때를 정의

※ 지나치게 이론적이고 이상적이여서 실무에서 잘 활용하지 않음.

그럼에도 사용하는 이유는 중복이 발생하지 않도록 분해해서 데이터 무결성을 높이는 것이 목적이기 때문.

- **2.22.** 정규화 요약

구분	제거 대상	특징
1정규화	다가-복합 속성 제거, 반복 속성 제거, 중첩 릴레이션 제거	속성이 추가되거나 일대다(1:M) 관계의 엔터티가 추가되며 관계를 상속시킴
2정규화	부분 종속 제거	일대다(1:M) 관계의 엔터티가 추가되며 관계를 상속받음
3정규화	이행 종속 제거	일대다(1:M) 관계의 엔터티가 추가되며 관계를 상속받음
BC정규화	종속자가 키에 포함된 함수 종속 제거	모든 결정자는 키이어야 한다는 관점에서 3정규형과 같음
4정규화	다가 종속 제거	다가 속성의 개수만큼 일대다(1:M) 관계의 엔터티가 추가되며 관계를 상속시킴
5정규화	조인 종속 제거	조인 종속이 존재하는 엔터티가 오히려 사용하기 편함. 지나치게 이상적임

[그림 2.77] 정규화 요약

- **2.23. 3**정규화까지만 수행하면 된다**? (P.174)**

  // 책을 저술한 저자는 4정규화까지 해야한다고 함. 그에 따른 내용

  Question. 1정규형을 사용하지 않을 때가 있다고 하는데 해당 예시 필요.

- **2.24.** 정규형과 성능

  // 정규형을 한 모델이 비정규형을 한 모델보다 빠를 수 있는 이유에 대한 내용.

  → 비정규형 모델은 중복 데이터와 인스턴스의 크기가 커 메모리의 사용을 높일 수 있기 때문.