중복을 최소화하기위해서 정규화를 한다. 데이터베이스의 가장 큰 목적은 **중복을 제거**하는 것이다.

**왜?** 중복이 있으면 이상 현상이 발생할 수 있다.

**🎲  이상 현상 ( Anomaly )**

1. 삭제이상: 연쇄적 삭제 문제 발생
2. 삽입이상: NULL 값 문제 발생
3. 수정이상: 불일치 문제 발생

**🎲  함수 종속성( Functional Dependency )**

어떤 속성의 값을 알면 다른 속성 값이 **유일**하게 정해지는 의존 관계를 종속한다고 한다.

ex. 학생 번호를 알면 학생이름을 알 수 있다. 하지만 학생번호로 학생이 수강하는 강좌이름을 유일하게 알 수는 없다. (여러개를 수강할 수 도 있기때문에)

X 각각의 값이 Y의 값 한 개와 대응이 되면 ‘X는 Y를 함수적으로 결정한다’, X→Y로 표기한다. X가 결정자이다.

\* 함수 종속성 규칙이 많다. 그중 이행 규칙은 X → Y, Y → Z이면 X → Z가 성립한다.

**👉 이상현상은 한개의 릴레이션(테이블)에 두개 이상의 정보가 포함되어있으면 나타난다.**

**👉 기본키가 아니면서 결정자(ex. X, 결정권자)인 속성이 있으면 발생한다.**

**🎲  정규화 ( Normalization )**

이상현상이 발생하는 릴레이션(테이블)을 **분해**해 이상현상을 없애는 과정을 말한다.

BCNF까지 진행하면 대다수의 이상현상이 없어진다. 따라서 BCNF까지 정규화를 진행하면 현장에서 충분하다고 판단한다.

**🔉 제 1 정규형:** 모든 속성값이 원자값(하나의 값)을 가지는 것이다.

**🔉 제 2 정규형:** 제 1 정규형이며, 기본키가 아닌 속성이 기본키에 **완전 함수 종속 관계**일 때를 말한다.

\* 완전 함수 종속: 기본키가 여러 속성으로 구성되어 있을 때 기본키를 구성하는 속성중 하나가 다른 속성에게 종속되면 안된다. 기본키(A,B) → C를 종속할 때 B → D가 성립되면 안된다. 이런 경우 B,D을 가지고 다른 테이블을 구성해야한다. 기본키를 기준으로 다른 속성이 모두 기본키에 종속되어야한다.

**🔉 제 3 정규형:**제 2 정규형이며, 기본키가 아닌 속성이 기본키에 비이행적으로 종속하는 것이다.

\* 이행적 종속은 X → Y, Y → Z 가 성립하면 X → Z가 성립

아래 예시는 학생이 한 강좌만 신청할 수 있다고 가정한 테이블이다. 이 경우 강좌이름과 수강료를 다른 테이블로 분해해야한다.

|  |  |  |
| --- | --- | --- |
| **학생번호** | **강좌이름** | **수강료** |
| 101 | 알고리즘 | 10000 |
| 201 | 알고리즘 | 10000 |
| 202 | 데이터베이스 | 15000 |
| 102 | 자료구조 | 20000 |

**🔉 BCNF:**함수 종속성 X → Y 가 성립할 때 모든 결정자인 X가 후보키이면 BCNF 정규형이다. 값이 중복될 수 있는 속성은 결정자가 되면 안된다. 따라서 보통 ID값을 주로 사용한다. 즉, 결정자가 되려면 기본키와 같은 성향, 후보키이어야한다.

\* 후보키는 중복값이 없는 속성을 말한다.