

# 9장. 정규화

# 정규화의 개념과 이상 현상

---

## ❖ 이상(anomaly) 현상

- 불필요한 데이터 중복으로 인해 릴레이션에 대한 데이터 삽입·수정·삭제 연산을 수행할 때 발생할 수 있는 부작용

## ❖ 정규화

- 이상 현상을 제거하면서 데이터베이스를 올바르게 설계해 나가는 과정

# 정규화의 개념과 이상 현상

---

## ❖ 이상 현상의 종류

삽입 이상

새 데이터를 삽입하기 위해 불필요한 데이터도 함께 삽입해야 하는 문제

갱신 이상

중복 튜플 중 일부만 변경하여 데이터가 불일치하게 되는 모순의 문제

삭제 이상

튜플을 삭제하면 꼭 필요한 데이터까지 함께 삭제되는 데이터 손실의 문제

# 정규화의 개념과 이상 현상

---

## ❖ 이상 현상을 설명하기 위한 릴레이션의 예

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	고객이름	등급
apple	E001	Y	정소화	gold
apple	E005	N	정소화	gold
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver

# 정규화의 개념과 이상 현상

---

## ❖ 삽입 이상(insertion anomaly)

- 릴레이션에 새 데이터를 삽입하려면 불필요한 데이터도 함께 삽입해야 하는 문제

[삽입 이상이 발생하는 예]

- 아직 이벤트에 참여하지 않은 아이디가 “melon”, 이름이 “성원용”, 등급이 “gold”인 신규 고객의 데이터는 이벤트참여 릴레이션에 삽입할 수 없음
- 삽입하려면 실제로 참여하지 않은 임시 이벤트번호를 삽입해야 함

# 정규화의 개념과 이상 현상

## ❖ 이벤트참여 릴레이션의 삽입 이상

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	고객이름	등급
apple	E001	Y	정소화	gold
apple	E005	N	정소화	gold
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver
melon	NULL	NULL	성원용	gold

← 삽입 불가!

# 정규화의 개념과 이상 현상

---

## ❖ 갱신 이상(update anomaly)

- 릴레이션의 중복된 튜플들 중 일부만 수정하여 데이터가 불일치하게 되는 모순이 발생하는 문제

[갱신 이상이 발생하는 예]

- 아이디가 “apple”인 고객의 등급이 “gold”에서 “vip”로 변경되었는데, 일부 튜플에 대해서만 등급이 수정된다면 “apple” 고객이 서로 다른 등급을 가지는 모순이 발생

# 정규화의 개념과 이상 현상

## ❖ 이벤트참여 릴레이전의 갱신 이상

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	고객이름	등급
apple	E001	Y	정소화	vip
apple	E005	N	정소화	vip
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver

← 데이터 불일치 발생!



# 정규화의 개념과 이상 현상

---

## ❖ 삭제 이상(deletion anomaly)

- 릴레이션에서 튜플을 삭제하면 꼭 필요한 데이터까지 손실되는 연쇄 삭제 현상이 발생하는 문제

[삭제 이상이 발생하는 예]

- 아이디가 “orange”인 고객이 이벤트 참여를 취소해 관련 튜플을 삭제하게 되면 이벤트 참여와 관련이 없는 고객아이디, 고객이름, 등급 데이터까지 손실됨

# 정규화의 개념과 이상 현상

## ❖ 이벤트참여 릴레이전의 삭제 이상

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	고객이름	등급
apple	E001	Y	정소화	gold
apple	E005	N	정소화	gold
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
<del>orange</del>	<del>E004</del>	<del>N</del>	<del>김용욱</del>	<del>silver</del>

← 데이터 손실 발생!

# 정규화의 개념과 이상 현상

---

## ❖ 이상의 원인

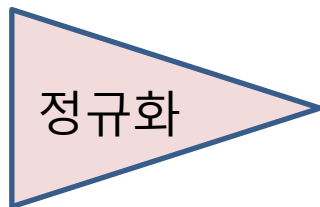
- 속성들 간에 존재하는 여러 종속관계를 하나의 릴레이션에 표현함

## ❖ 정규화

- 이상 현상이 발생하지 않도록, 릴레이션을 관련 있는 속성들로만 구성하기 위해 릴레이션을 분해(decomposition)하는 과정
- 함수적 종속성을 판단하여 정규화를 수행함

## ❖ 함수적 종속성(FD; Functional Dependency)

- 속성들 간의 관련성



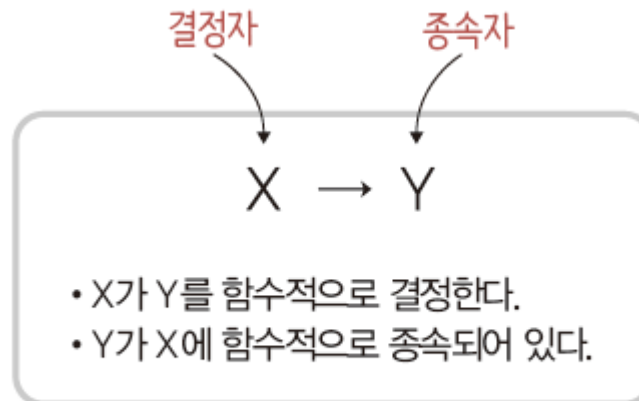
함수적 종속성을 이용하여, 릴레이션을 연관성이 있는 속성들로만 구성되도록 분해하여 이상 현상이 발생하지 않는 올바른 릴레이션으로 만들어 나가는 과정

# 함수 종속

## ❖ 함수 종속

### ■ “X가 Y를 함수적으로 결정한다”

- 릴레이션 내의 모든 튜플에서 하나의 X 값에 대한 Y 값이 항상 하나임
- X와 Y는 하나의 릴레이션을 구성하는 속성들의 부분 집합
- “Y가 X에 함수적으로 종속되어 있다”와 같은 의미
- $X \rightarrow Y$ 로 표현(X는 결정자, Y는 종속자)



# 함수 종속

## ❖ 함수 종속 관계 판단 예 (1)

고객아이디	고객이름	등급
apple	정소화	gold
banana	김선우	vip
carrot	고명석	gold
orange	김용욱	silver

각 고객아이디 속성 값에 대응되는  
고객이름 속성과 등급 속성 값이  
단 하나임

함수 종속 관계 설명을 위한 릴레이션의 예 : 고객 릴레이션

고객아이디 → 고객이름  
고객아이디 → 등급

또는

고객아이디 → (고객이름, 등급)

고객 릴레이션에 존재하는 함수 종속 관계

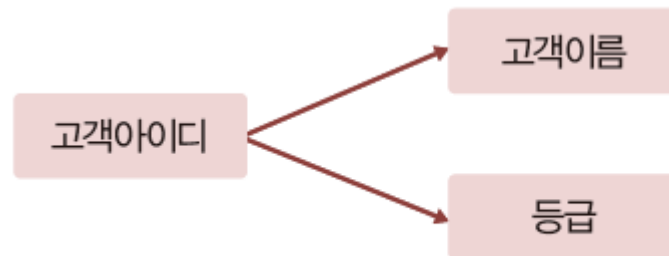
# 함수 종속

## ❖ 함수 종속 관계 판단 예 (1)

- 함수 종속 다이어그램 : 함수 종속 관계를 도식화하여 표현한 것



고객 릴레이션에 존재하는 함수 종속 관계



고객 릴레이션의 함수 종속 다이어그램

## ❖ 함수 종속 관계 판단 시 유의 사항

- 속성 자체의 특성과 의미를 기반으로 함수 종속성을 판단해야 함
  - 속성 값은 계속 변할 수 있으므로 현재 릴레이션에 포함된 속성 값만으로 판단하면 안됨
- 후보키(기본키와 대체키)
  - 릴레이션의 다른 모든 속성들을 함수적으로 결정함
- 후보키가 아닌 속성 값
  - 다른 속성 값을 유일하게 결정하면 함수 종속 관계에서 결정자가 될 수 있음

# 함수 종속

## ❖ 함수 종속 관계 판단 예 (2)

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	고객이름
apple	E001	Y	정소화
apple	E005	N	정소화
apple	E010	Y	정소화
banana	E002	N	김선우
banana	E005	Y	김선우
carrot	E003	Y	고명석
carrot	E007	Y	고명석
orange	E004	N	김용욱

함수 종속 관계를 설명하기 위한 릴레이션의 예 : 이벤트참여 릴레이션



# 함수 종속

## ❖ 함수 종속 관계 판단 예 (2)

고객아이디 → 고객이름

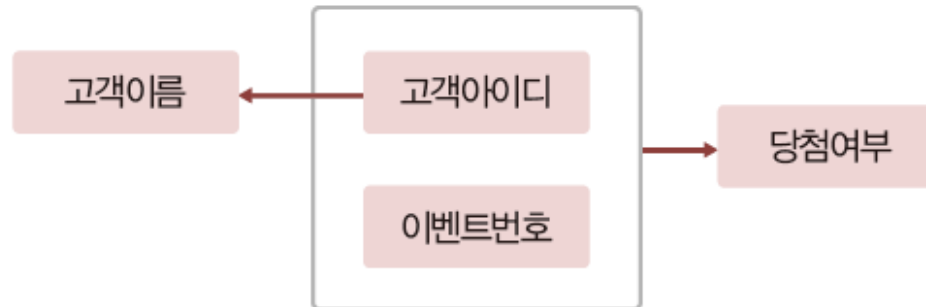
{고객아이디, 이벤트번호} → 당첨여부

{고객아이디, 이벤트번호} → 고객이름

고객이름은 {고객아이디, 이벤트번호}의  
일부분인 고객아이디에 종속되어 있음

→ **고객이름은 {고객아이디, 이벤트번호}에  
부분 함수 종속됨**

이벤트참여 릴레이션에 존재하는 함수 종속 관계



이벤트참여 릴레이션의 함수 종속 다이어그램

# 함수 종속

---

## ❖ 완전 함수 종속(FFD; Full Functional Dependency)

- 릴레이션에서 속성 집합 Y가 속성 집합 X에 함수적으로 종속되어 있지만, 속성 집합 X의 전체가 아닌 일부분에는 종속되지 않음을 의미
  - 일반적으로 함수 종속은 완전 함수 종속을 의미함
- 예) 당첨여부는 {고객아이디, 이벤트번호}에 완전 함수 종속됨

## ❖ 부분 함수 종속(PFD; Partial Functional Dependency)

- 릴레이션에서 속성 집합 Y가 속성 집합 X의 전체가 아닌 일부분에도 함수적으로 종속됨을 의미
- 예) 고객이름은 {고객아이디, 이벤트번호}에 부분 함수 종속됨

# 함수 종속

---

## ❖ 고려할 필요가 없는 함수 종속 관계

- 결정자와 종속자가 같거나, 결정자가 종속자를 포함하는 것처럼 당연한 함수 종속 관계는 고려하지 않음

고객아이디 → 고객아이디

{고객아이디, 이벤트번호} → 이벤트번호

고려할 필요가 없는 함수 종속 관계의 예

# 기본 정규형과 정규화 과정

---

## ❖ 정규화(normalization)

### ■ 개념

- 함수 종속성을 이용해 릴레이션을 연관성이 있는 속성들로만 구성되도록 분해해서, 이상 현상이 발생하지 않는 올바른 릴레이션으로 만들어 가는 과정

### ■ 원칙

#### 1) 무손실 분해(nonloss decomposition)

- 릴레이션이 의미상 동등한 릴레이션들로 분해되어야 하고, 분해로 인한 정보 손실이 발생하지 않아야 함
- 분해된 릴레이션들을 자연 조인하면 분해 전의 릴레이션으로 복원 가능해야 함

#### 2) 데이터의 중복성을 감소

#### 3) 분리의 원칙

- 독립적인 관계는 별개의 릴레이션으로 표현해야 함

# 기본 정규형과 정규화 과정

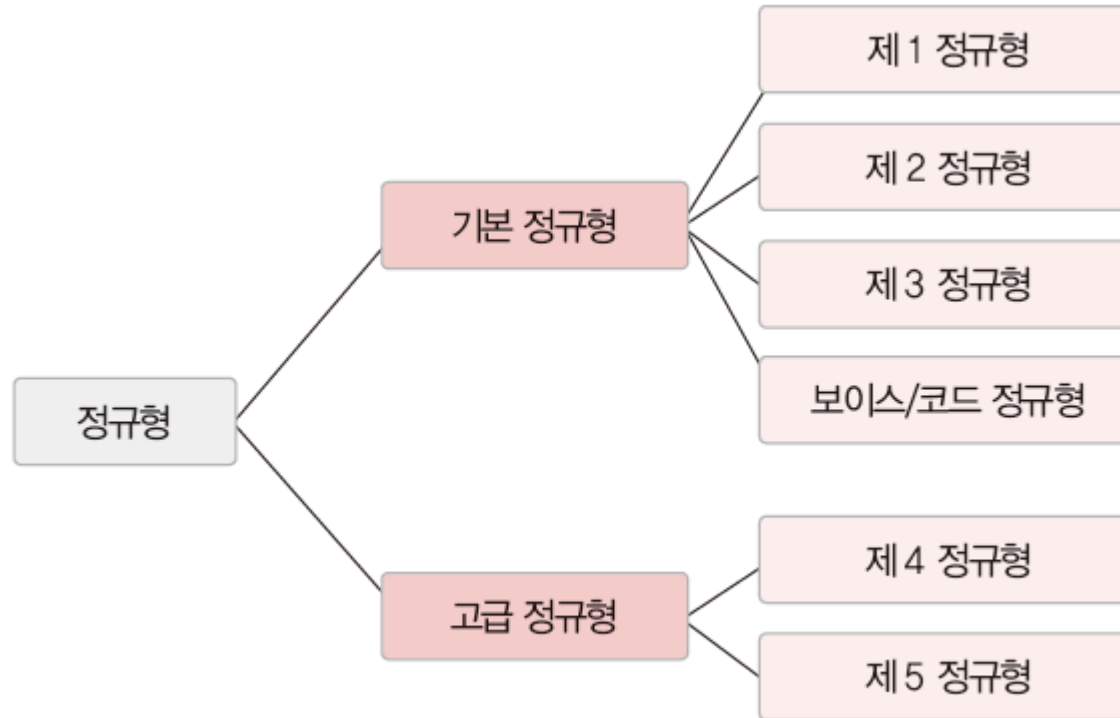
---

## ❖ 정규형(NF; Normal Form)

- 릴레이션이 정규화된 정도
- 각 정규형마다 제약조건이 존재
  - 정규형의 차수가 높아질수록 요구되는 제약조건이 많아지고 엄격해짐
- 릴레이션의 특성을 고려해서 적합한 정규형을 선택

# 기본 정규형과 정규화 과정

---



# 기본 정규형과 정규화 과정

---



정규형들의 관계

# 기본 정규형과 정규화 과정

---

## ❖ 제 1 정규형(1NF; First Normal Form)

- 릴레이션의 모든 속성이 더는 분해되지 않는 **원자 값(atomic value)**만 가짐
- 제 1 정규형을 만족해야 관계 데이터베이스의 릴레이션이 될 자격이 있음

### 제1 정규형(1NF)

릴레이션에 속한 모든 속성의 도메인이 원자 값 atomic value으로만 구성되어 있으면 제 1 정규형에 속한다.



# 기본 정규형과 정규화 과정

## ❖ 제 1 정규형(1NF; First Normal Form)

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	등급	할인율
apple	E001, E005, E010	Y, N, Y	gold	10%
banana	E002, E005	N, Y	vip	20%
carrot	E003, E007	Y, Y	gold	10%
orange	E004	N	silver	5%

다중 값 속성을 포함하는 이벤트참여 릴레이션

제 1정규형을 만족하지 않는 릴레이션

# 기본 정규형과 정규화 과정

## ❖ 제 1 정규형(1NF; First Normal Form)

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	등급	할인율
apple	E001	Y	gold	10%
apple	E005	N	gold	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%

제1정규형에 속하는 이벤트참여 릴레이션

# 기본 정규형과 정규화 과정

❖ 제 1 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	등급	할인율
apple	E001	Y	gold	10%
apple	E005	N	gold	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%

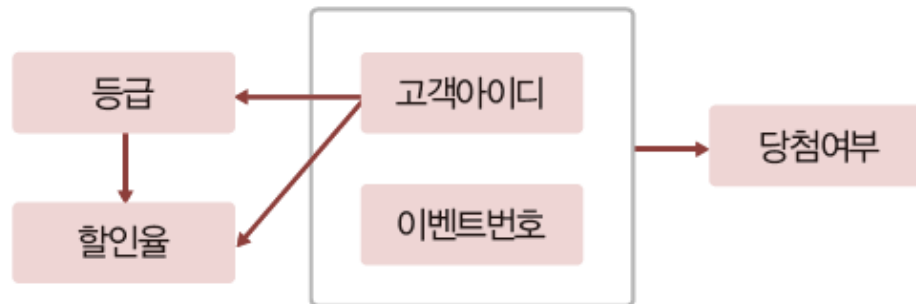
제1정규형에 속하는 이벤트참여 릴레이션

# 기본 정규형과 정규화 과정

## ❖ 제 1 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

고객아이디 → 등급  
고객아이디 → 할인율  
등급 → 할인율  
{고객아이디, 이벤트번호} → 당첨여부

이벤트참여 릴레이션에 존재하는 함수 종속 관계



이벤트참여 릴레이션의 함수 종속 다이어그램

# 기본 정규형과 정규화 과정

## ❖ 제 1 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

고객아이디	이벤트번호	당첨여부	등급	할인율
apple	E001	Y	vip	10%
apple	E005	N	vip	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
<del>orange</del>	<del>E004</del>	<del>N</del>	<del>silver</del>	<del>5%</del>
grape	NULL	NULL	silver	5%

← 데이터 불일치로 인한 갱신 이상

← 데이터 손실로 인한 삭제 이상

← 삽입 불가로 인한 삽입 이상

- 이상 현상의 발생 이유는?

기본키에 완전 함수 종속되지 못한 등급과 할인율 때문

# 기본 정규형과 정규화 과정

---

## ❖ 제 1 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

### ■ 이상 현상의 발생 이유

- 기본키인 {고객아이디, 이벤트번호}에 완전 함수 종속되지 못하고 일부분인 고객아이디에 종속되는 등급과 할인을 속성이 존재하기 때문

### ■ 문제 해결 방법

- 부분 함수 종속이 제거되도록 이벤트참여 릴레이션을 분해  
→ 분해된 릴레이션들은 제 2 정규형에 속하게 됨

# 기본 정규형과 정규화 과정

---

## ❖ 제 2 정규형(2NF; Second Normal Form)

- 릴레이션이 제 1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 완전 함수 종속됨
- 제 1 정규형에 속하는 릴레이션이 제 2 정규형을 만족하게 하려면?
  - 부분 함수 종속을 제거하고 모든 속성이 기본키에 완전 함수 종속되도록 분해

### 제2 정규형(2NF)

릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 완전 함수 종속되면 제2 정규형에 속한다.

# 기본 정규형과 정규화 과정

## ❖ 제 2 정규형(2NF; Second Normal Form)

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부	등급	할인율
apple	E001	Y	gold	10%
apple	E005	N	gold	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%

제1정규형에 속하는 이벤트참여 릴레이션

- 이상 현상의 발생 이유는?

기본키에 완전 함수 종속되지 않은 등급과 할인율 속성 때문



# 기본 정규형과 정규화 과정

## ❖ 제 2 정규형(2NF; Second Normal Form)

분해 전의 이벤트참여 릴레이션

고객아이디	이벤트번호	당첨여부	등급	할인율
apple	E001	Y	gold	10%
apple	E005	N	gold	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%

부분 함수 종속을 제거하려고 분해

고객 릴레이션

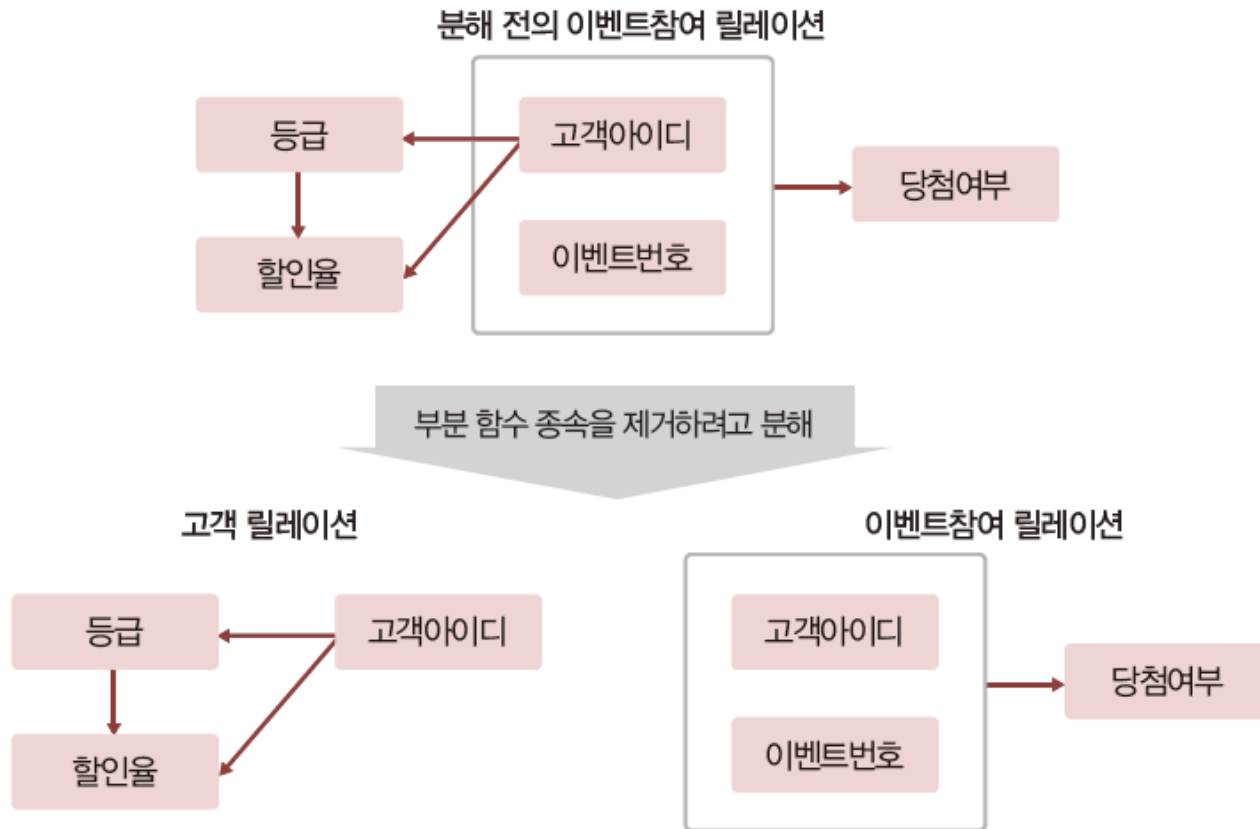
고객아이디	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

이벤트참여 릴레이션

고객아이디	이벤트번호	당첨여부
apple	E001	Y
apple	E005	N
apple	E010	Y
banana	E002	N
banana	E005	Y
carrot	E003	Y
carrot	E007	Y
orange	E004	N

# 기본 정규형과 정규화 과정

## ❖ 제 2 정규형(2NF; Second Normal Form)



# 기본 정규형과 정규화 과정

## ❖ 제 2 정규형(2NF; Second Normal Form)

분해 전의 이벤트참여 릴레이션

부분 함수 종속을 제거하려고 분해

고객 릴레이션

<u>고객아이디</u>	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

함수 종속성을 여러 개 포함하여  
이상 현상이 발생할 수 있음

이벤트참여 릴레이션

<u>고객아이디</u>	<u>이벤트번호</u>	당첨여부
apple	E001	Y
apple	E005	N
apple	E010	Y
banana	E002	N
banana	E005	Y
carrot	E003	Y
carrot	E007	Y
orange	E004	N

함수 종속성을 하나만 포함하여 이상 현상 발생하지 않음

부분 함수 종속 해소

# 기본 정규형과 정규화 과정

---

❖ 제 2 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

고객 릴레이션

고객아이디	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

# 기본 정규형과 정규화 과정

## ❖ 제 2 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

고객아이디	등급	할인율
apple	gold	15%
<del>banana</del>	<del>vip</del>	<del>20%</del>
carrot	gold	10%
orange	silver	5%
NULL	bronze	1%

← 데이터 불일치로 인한 갱신 이상

← 데이터 손실로 인한 삭제 이상

← 삽입 불가로 인한 삽입 이상

- 이상 현상의 발생 이유는?

**이행적 함수 종속이 존재하기 때문**

# 기본 정규형과 정규화 과정

---

## ❖ 제 2 정규형은 만족하지만 이상 현상이 발생하는 릴레이션 예

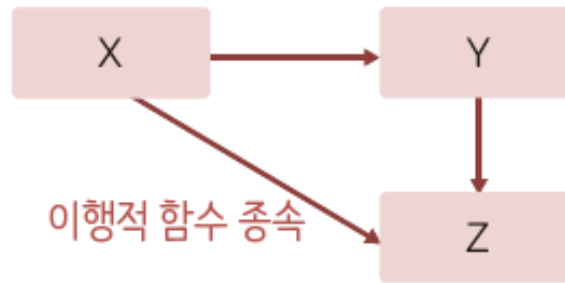
- 이상 현상의 발생 이유
  - 이행적 함수 종속이 존재하기 때문
- 문제 해결 방법
  - 이행적 함수 종속이 제거되도록 고객 릴레이션을 분해
    - 분해된 릴레이션들은 제 3 정규형에 속하게 됨

# 기본 정규형과 정규화 과정

---

## ❖ 이행적 함수 종속(transitive FD)

- 릴레이션을 구성하는 3개의 속성 집합  $X, Y, Z$ 에 대해 함수 종속 관계  $X \rightarrow Y$ 와  $Y \rightarrow Z$ 가 존재하면 논리적으로  $X \rightarrow Z$ 가 성립되는데, 이때  $Z$ 가  $X$ 에 이행적으로 함수 종속되었다고 함



# 기본 정규형과 정규화 과정

---

## ❖ 제 3 정규형(3NF; Third Normal Form)

- 릴레이션이 제 2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 이행적 함수 종속이 되지 않음
- 제 2 정규형에 속하는 릴레이션이 제 3 정규형을 만족하게 하려면?
  - 모든 속성이 기본키에 이행적 함수 종속이 되지 않도록 분해

### 제3 정규형(3NF)

릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 이행적 함수 종속이 되지 않으면 제3 정규형에 속한다.



# 기본 정규형과 정규화 과정

## ❖ 제 3 정규형(3NF; Third Normal Form)

<u>고객아이디</u>	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

- 이상 현상의 발생 이유는?

고객아이디가 등급을 통해 할인율을 결정하는  
이행적 함수 종속 관계가 존재하기 때문

# 기본 정규형과 정규화 과정

## ❖ 제 3 정규형(3NF; Third Normal Form)

분해 전의 고객 릴레이션

고객아이디	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

이행적 함수 종속을 제거하려고 분해

고객 릴레이션

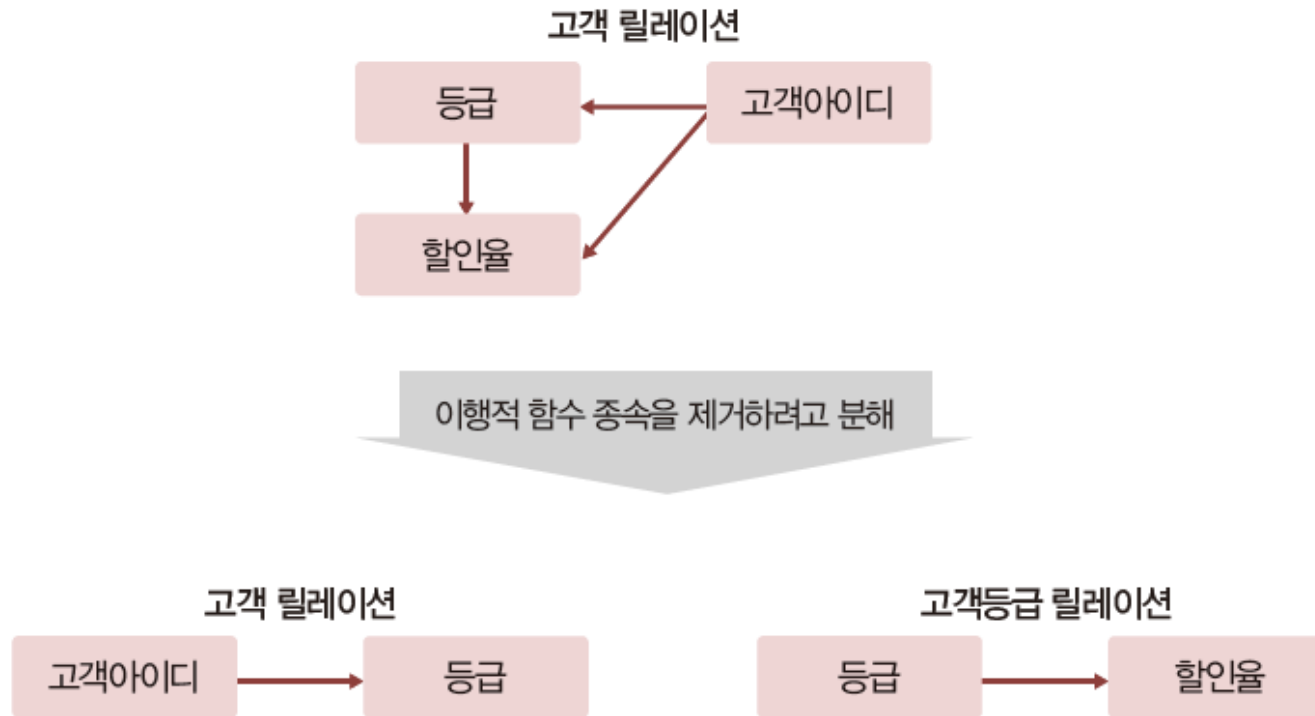
고객아이디	등급
apple	gold
banana	vip
carrot	gold
orange	silver

고객등급 릴레이션

등급	할인율
gold	10%
vip	20%
silver	5%

# 기본 정규형과 정규화 과정

## ❖ 제 3 정규형(3NF; Third Normal Form)



# 기본 정규형과 정규화 과정

## ❖ 보이스/코드 정규형(BCNF; Boyce/Codd Normal Form)

### ■ 필요성

- 하나의 릴레이션에 여러 개의 후보키가 존재하는 경우, 제 3 정규형까지 모두 만족해도 이상 현상이 발생할 수 있음

### ■ 의미

- 강한 제 3 정규형(strong 3NF)
  - 후보키를 여러 개 가지고 있는 릴레이션에 발생할 수 있는 이상 현상을 해결하기 위해 제 3 정규형 보다 좀 더 엄격한 제약조건을 제시
  - 보이스/코드 정규형에 속하는 모든 릴레이션은 제 3 정규형에 속하지만, 제 3 정규형에 속하는 모든 릴레이션이 보이스/코드 정규형에 속하는 것은 아님

#### 보이스/코드 정규형(BCNF)

릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이면 보이스/코드 정규형에 속한다.

# 기본 정규형과 정규화 과정

## ❖ 보이스/코드 정규형을 만족하지 않는 릴레이션 예

고객아이디	인터넷강좌	담당강사번호
apple	영어회화	P001
banana	기초토익	P002
carrot	영어회화	P001
carrot	기초토익	P004
orange	영어회화	P003
orange	기초토익	P004

[강좌신청 릴레이션의 후보키]

{고객아이디, 인터넷강좌} : 기본키

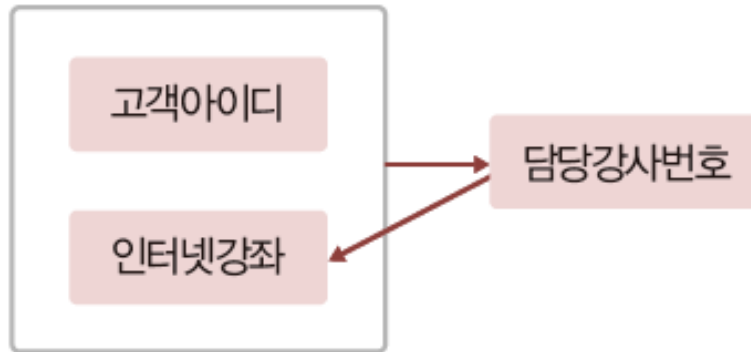
{고객아이디, 담당강사번호}

보이스/코드 정규형을 설명하기 위한 릴레이션의 예 : 강좌신청 릴레이션

# 기본 정규형과 정규화 과정

---

## ❖ 보이스/코드 정규형을 만족하지 않는 릴레이션 예



강좌신청 릴레이션의 함수 종속 다이어그램

# 기본 정규형과 정규화 과정

## ❖ 보이스/코드 정규형을 만족하지 않는 릴레이션 예

고객아이디	인터넷강좌	담당강사번호	
apple	영어회화	P001	
<del>banana</del>	<del>기초토익</del>	<del>P002</del>	← 데이터 손실로 인한 삭제 이상
carrot	영어회화	P001	
carrot	중급토익	P004	← 데이터 불일치로 인한 갱신 이상
orange	영어회화	P003	
orange	기초토익	P004	← 삽입 불가로 인한 삽입 이상
NULL	중급토익	P005	

- 이상 현상의 발생 이유는?

담당강사번호가 후보키가 아님에도 인터넷강좌 속성을 결정하기 때문

# 기본 정규형과 정규화 과정

## ❖ 보이스/코드 정규형(BCNF; Boyce/Codd Normal Form)

강좌신청 릴레이션

고객아이디	인터넷강좌	담당강사번호
apple	영어회화	P001
banana	기초토익	P002
carrot	영어회화	P001
carrot	기초토익	P004
orange	영어회화	P003
orange	기초토익	P004

후보키가 아닌 결정자를  
제거하려고 분해

고객담당강사 릴레이션

고객아이디	담당강사번호
apple	P001
banana	P002
carrot	P001
carrot	P004
orange	P003
orange	P004

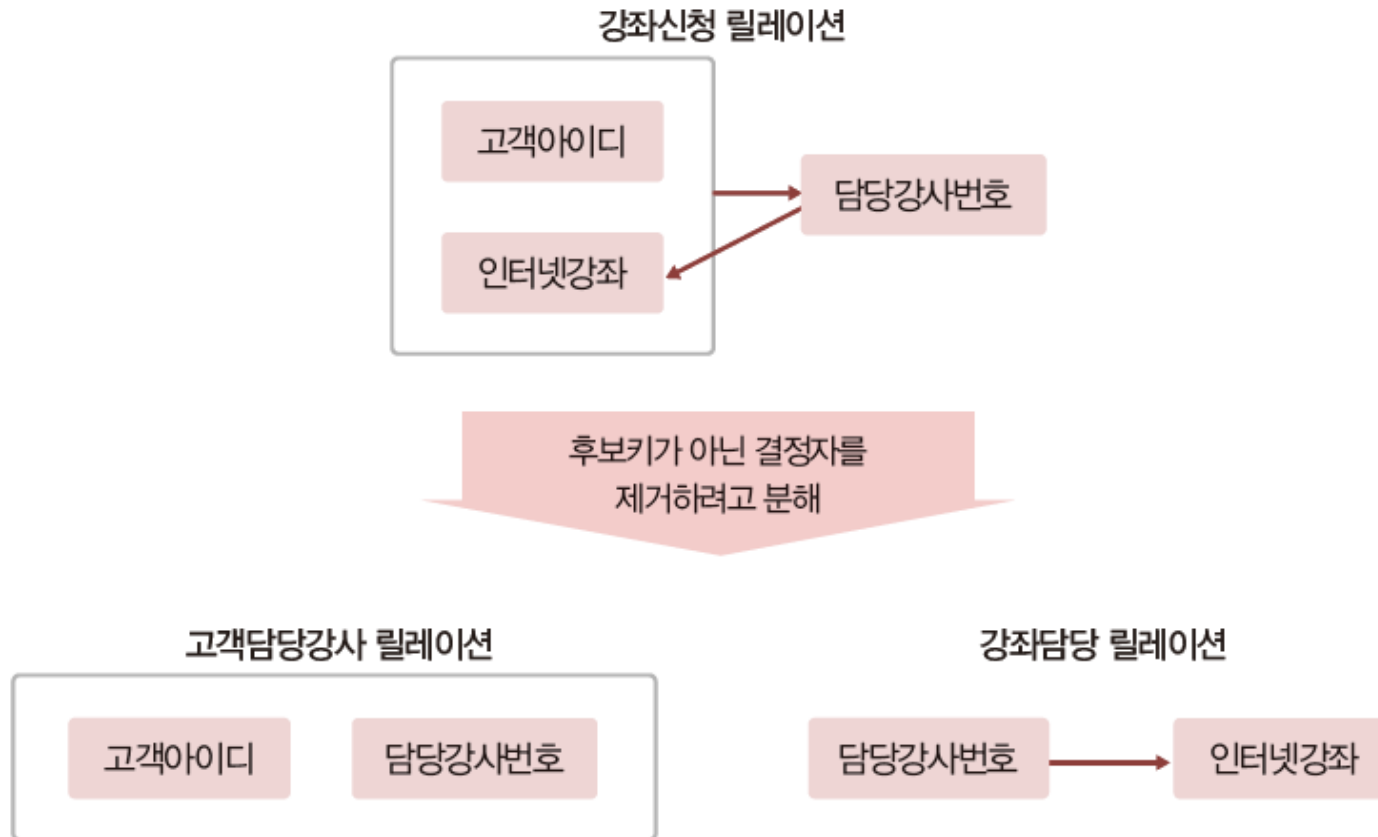
강좌담당 릴레이션

담당강사번호	인터넷강좌
P001	영어회화
P002	기초토익
P003	영어회화
P004	기초토익

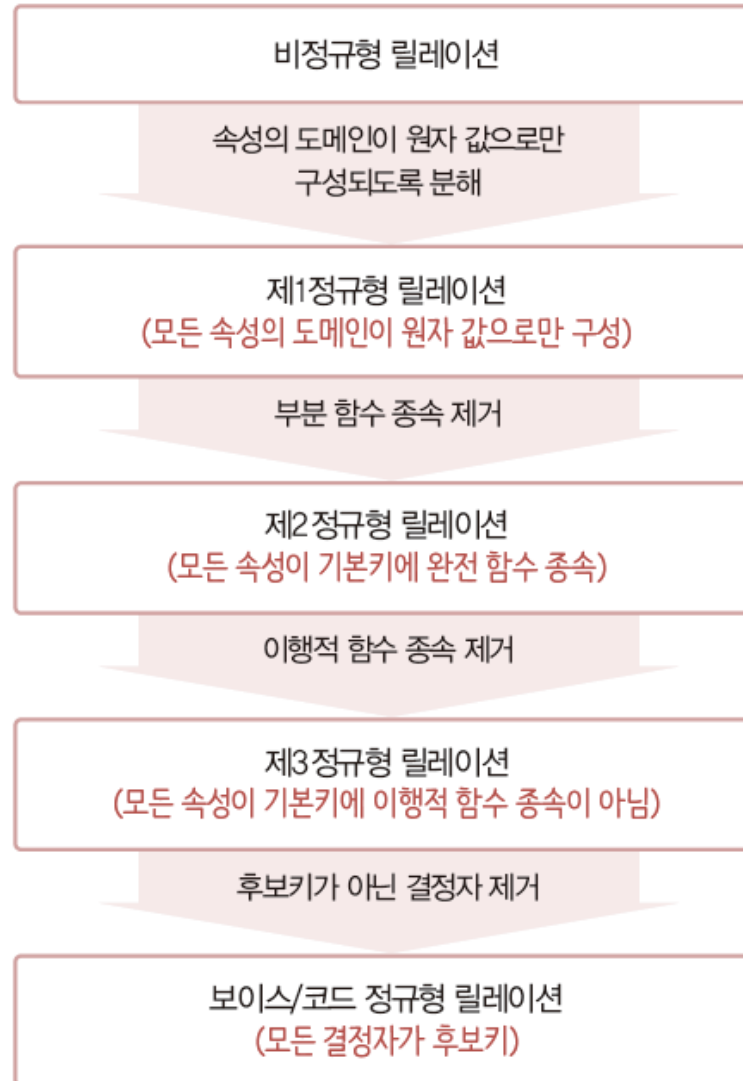


# 기본 정규형과 정규화 과정

## ❖ 보이스/코드 정규형(BCNF; Boyce/Codd Normal Form)



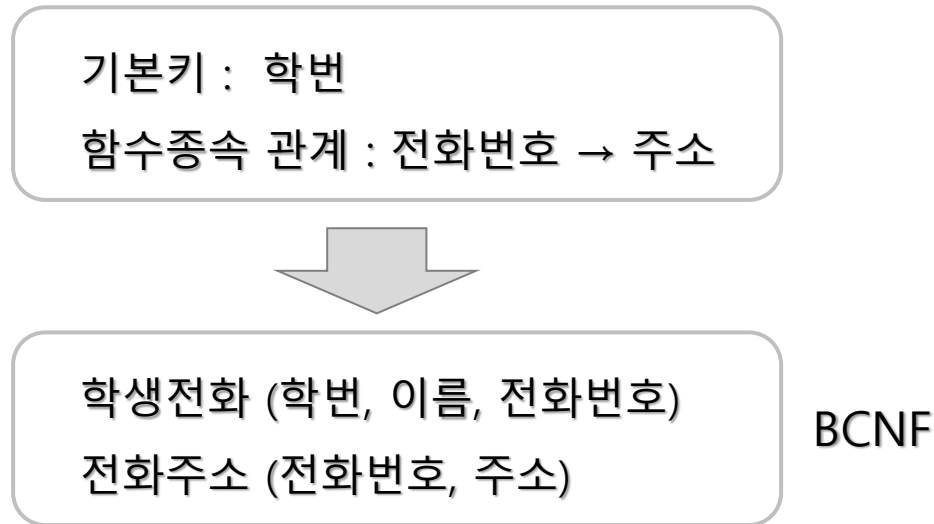
# 기본 정규형과 정규화 과정



# 기본 정규형과 정규화 과정

## ❖ 정규화 시 주의 사항

- 현실적으로 모든 릴레이션을 반드시 BCNF에 속하도록 분해할 필요는 없음
  - 학생주소 (학번, 이름, 주소, 전화번호)



이름, 전화번호, 주소는 분리하지 않고 사용하는 것이 편리하므로  
위와 같은 BCNF으로의 분해는 무의미함

# 기본 정규형과 정규화 과정

---

기본키 : 학번

함수종속 관계 : 전화번호  $\rightarrow$  주소

학생전화(학번,이름,전화번호)

전화주소(전화번호,주소)