

1. 이상(Anomaly)

1-1. 이상이란

- 중복된 데이터 때문에 데이터에 의도하지 않은 현상이 발생되면 이를 이상현상이라고 한다.
- 이상 현상에는 삽입 이상, 갱신 이상, 삭제 이상이 존재한다.

1-2. 삽입 이상(Addition Anomaly)

주문 번호	상품번호	상품명	단가	주문수량
1001	P01	사과	1,500	2
1002	P02	바나나	2,000	20

- 릴레이션에서 새로운 인스턴스를 삽입할 때 발생하는 데이터 이상 현상이다.
- 불필요한 정보를 함께 저장하지 않고는 어떤 정보를 저장하는 것이 불가능하다.

주문 번호	상품번호	상품명	단가	주문수량
1001	P01	사과	1,500	2
1002	P02	바나나	2,000	20
1003	P01	사과	1,500	10
1004	P01	사과	1,500	15
NULL	P03	키위	3,000	NULL

- 인스턴스 삽입 시 기존의 상품이라도 상품번호, 상품명, 단가를 모두 추가해 주어야 한다.
- 주문이 아닌 상품을 추가하기 위한 인스턴스 추가일 경우 주문번호나 주문 수량이라는 불필요한 속성의 값도 추가하여야 한다.

1-3. 갱신 이상(Update Anomaly)

- 릴레이션에서 속성의 값을 업데이트할 때 발생하는 데이터 이상 현상이다.
- 반복된 데이터 중에 일부만 수정하면 데이터의 불일치가 발생한다.

주문 번호	상품번호	상품명	단가	주문수량
1001	P01	사과	2,500	2
1002	P02	바나나	2,000	20
1003	P01	사과	2,500	10
1004	P01	사과	2,500	15
NULL	P03	키위	3,000	NULL

- 상품명이나 단가와 같은 속성의 값들이 변경될 경우 같은 상품은 빠짐없이 모두 수정해 주어야 한다.

1-4. 삭제 이상(Deletion Anomaly)

- 릴레이션에서 인스턴스를 삭제할 때 발생하는 데이터 이상 현상이다.
- 유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능하다.

주문 번호	<u>상품번호</u>	상품명	단가	<u>주문수량</u>
1001	P01	사과	2,500	2
1002	P02	바나나	2,000	20
1003	P01	사과	2,500	10
1004	P01	사과	2,500	15
NULL	P03	키위	3,000	NULL



주문 번호	<u>상품번호</u>	상품명	단가	<u>주문수량</u>
1001	P01	사과	2,500	2
1002	P02	바나나	2,000	20
1003	P01	사과	2,500	10
1004	P01	사과	2,500	15

- 주문 내역을 삭제하려고 할때 관련 상품도 같이 소멸되어 상품 자체가 사라져 버리게 된다.

2. 논리 모델(Logical Model)

2-1. 논리 모델이란

- 개념 모델을 상세화하는 작업으로 전체 속성을 도출하고 도출되지 않은 대부분의 엔티티들과 관계들을 도출하는 단계이다.
- 정규화(Normalization)를 진행하는 단계이다.

2-2. 논리 모델의 목적

- 업무에 대해 충분히 의견을 교환하고 반영하여 진행하는데 도움을 준다.
- 중복값을 제거하여 이상(Anomaly)현상을 제거하기 위해 속성간에 종속관계를 확인하고 엔티티를 분할한다.

2-3. 논리 모델의 주의사항

- 업무 요건을 빠짐없이 정확하게 반영해야 한다.
- 더 이상 삭제할 엔티티나 속성은 없어야 한다.
- 주 식별자에 있어 효율성에 따라 인조 식별자를 채택할지에 대해 결정한다.
- 지나치게 성능 문제를 해결하려고 하지 않는다.

3. 정규화(Normalization)

3-1. 정규화란

3-1-1. 정규화의 목적

- 중복 데이터를 제거하여 안정성과 확장성을 도모한다.
 - 안정성: 함수 종속을 기반으로 데이터의 성격에 맞는 엔티티가 도출되어 모델 구조를 정의할 수 있다.
 - 확장성: 데이터의 정체성이 그대로 반영되어 업무가 수정되거나 추가 되더라도 엔티티에 반영하기가 수월해 진다.

3-1-2. 함수 종속

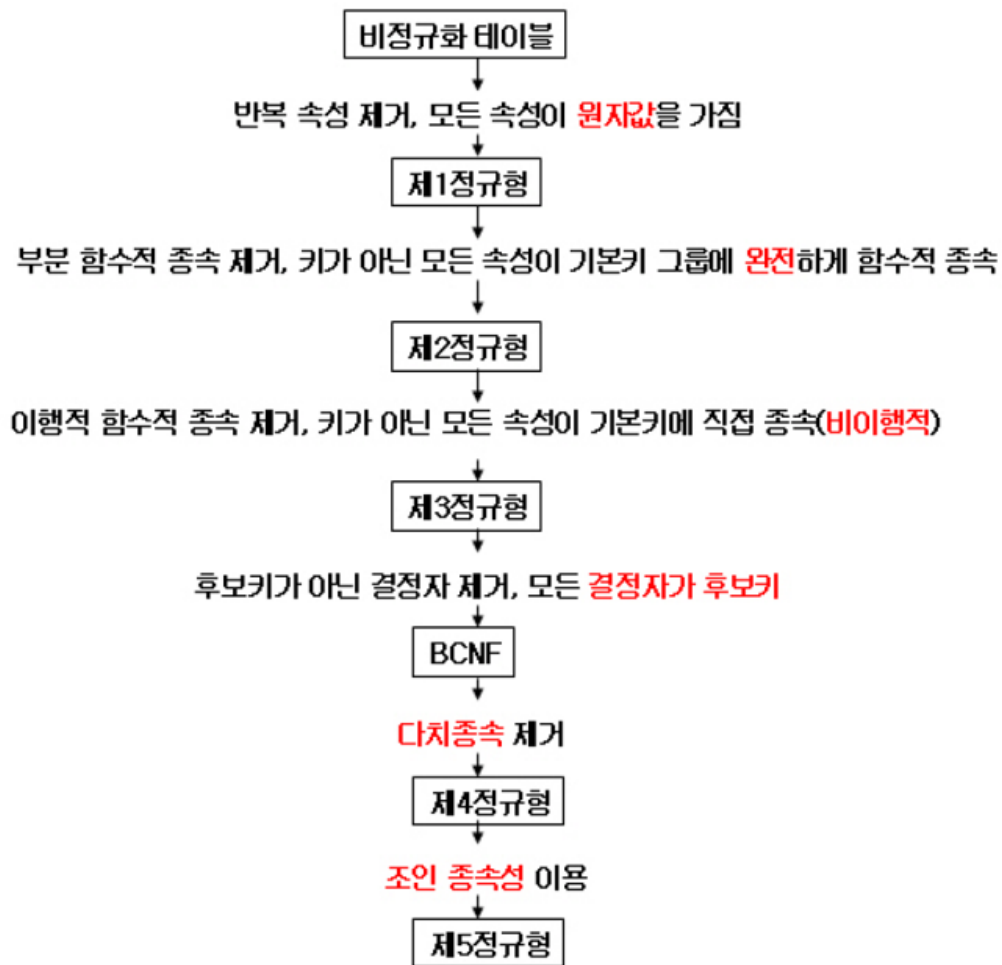
- 릴레이션 내에 존재하는 속성 간의 종속성을 의미한다.
- 대표 속성(식별자)이 나머지 속성을 유일하게 식별할 수 있다면 대표 속성과 나머지 속성 사이에는 연관관계가 성립하고 이것을 함수 종속이라고 한다.
- 속성 간의 종속성을 나눌 때 기준이 되는 결정자(Determinant)와 결정자의 값에 의해 유일하게 식별될 수 있는 값을 종속자(Dependent)라고 한다.
-

$$X \rightarrow Y, y = f(x)$$

속성 Y가 속성 X에 의해 함수적으로 종속됨을 표현한 것이다.

- 속성 X의 값을 이용해 속성 Y의 값을 유일하게 식별할 수 있다는 의미로 X의 값에 대해 Y의 값이 오직 하나만 연관되어 있음을 의미한다.

3-2. 정규화의 종류



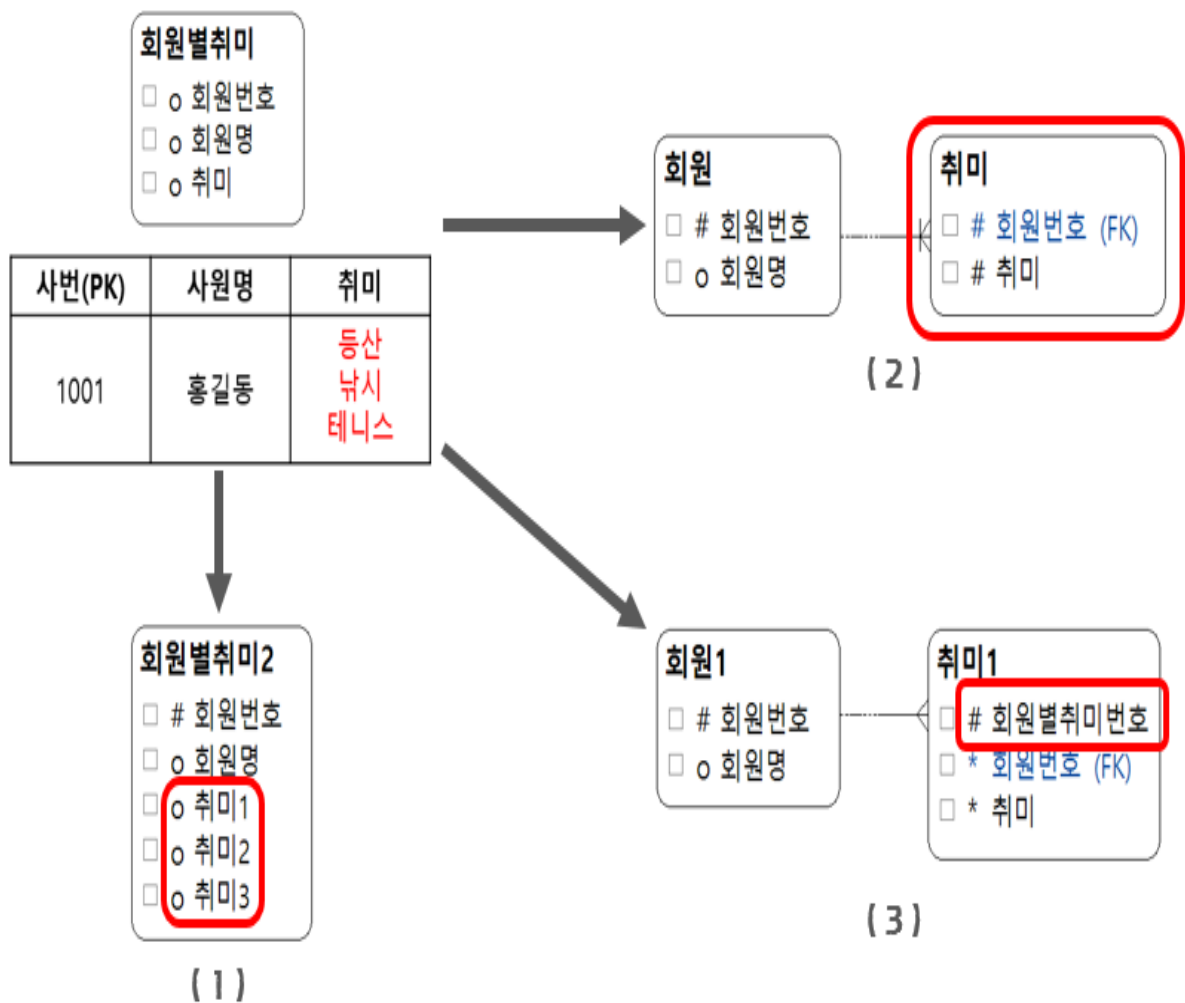
3-2-1. 1 정규형(1NF)

회원 별취미

- ☐ ○ 회원번호
- ☐ ○ 회원명
- ☐ ○ 취미

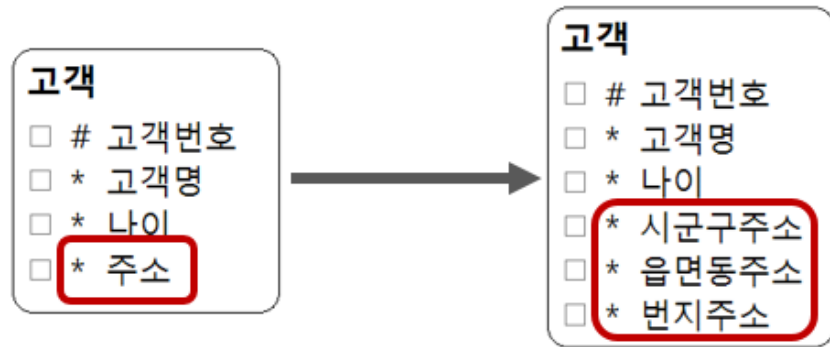
사번(PK)	사원명	취미
1001	홍길동	등산 낚시 테니스

- 모든 속성은 반드시 하나의 값을 가져야 한다.(속성은 원자값을 지녀야 한다.)
- 다가 속성(Multivalued Attributes)과 복합 속성(Composite Attributes)에 대해 판단하여 처리한다.
 - 다가 속성: 하나의 인스턴스에 하나의 속성이 여러 값을 가지게 되는 속성
 - 복합 속성: 여러 속성이 하나의 속성으로 묶여있는 속성
- 다가 속성과 관련된 1 정규형 처리 방법



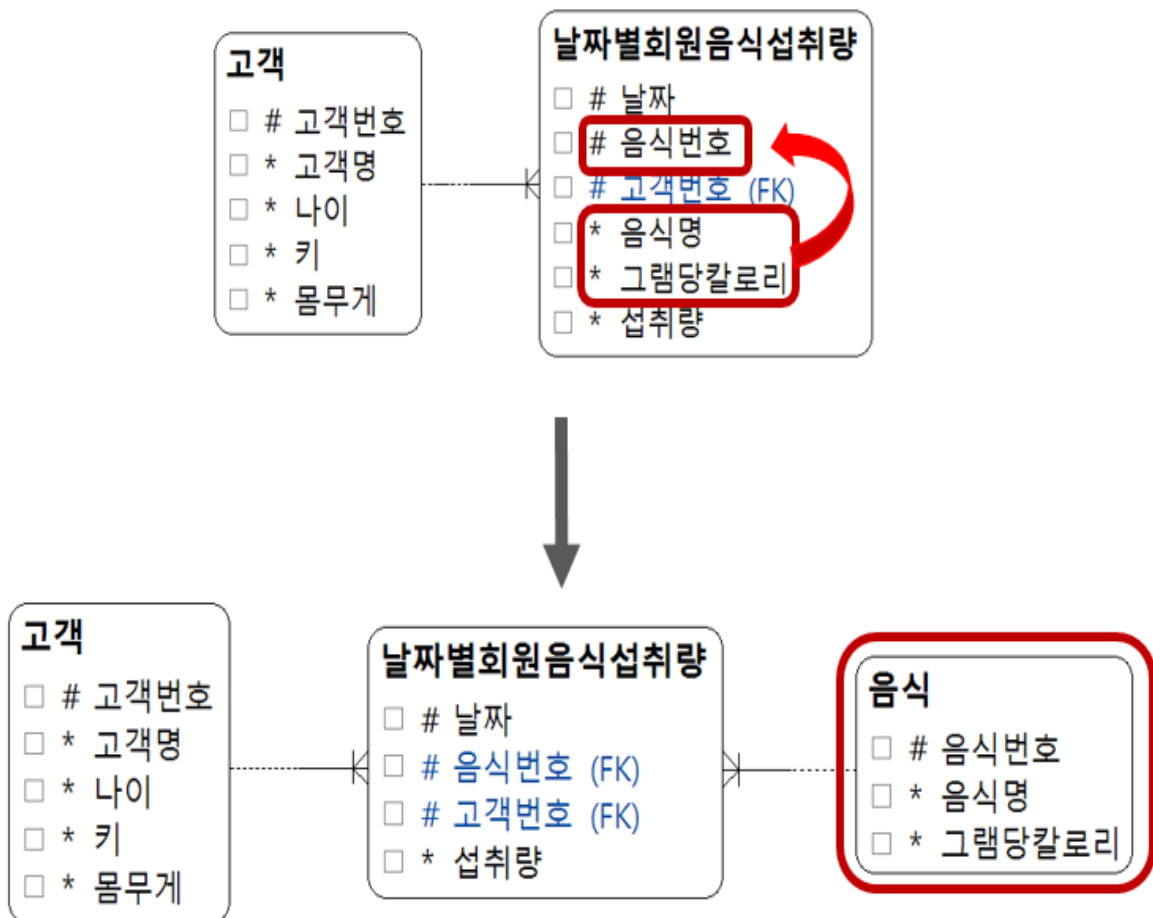
- (1)의 경우와 같이 속성을 추가해서 인스턴스별로 하나의 속성에 하나의 값이 들어갈 수 있게 할 수 있다. 다만 이렇게 처리할 경우 속성의 갯수가 더 적을 경우가 많거나 추가한 속성의 수보다 많은 값은 처리할 수 없다. 속성이 한정적이고 제한적으로 존재할 때 처리할 수 있는 방법이다.
- (2)의 경우와 같이 엔터티를 따로 추출해서 1대 다 관계로 처리할 수 있다. 새로 생성된 엔터티의 인스턴스들을 구분하기 위해 속성들을 모두 식별자(복합키)로 활용하는 경우이다. (인스턴스의 구분 기준을 고려하여 식별자를 선정)
- (3)의 경우는 (2)에서 추가적인 속성으로 인조식별자를 추가하여 인스턴스를 구분하고 나머지 속성은 비식별자로 처리하는 경우이다.

- 복합 속성과 관련된 1정규형 처리 방법



- 주소라는 속성은 여러 속성이 하나로 묶인 복합속성의 의미가 될 수 있다.
- 업무상 주소를 구분해서 사용하고자 하는 경우가 많을 때 속성을 구별하며 되려 하나의 주소로 다룰 때가 많을 때는 속성을 구분해서 사용하는 것이 좋지 않다.

3-2-2. 2 정규형(2NF)



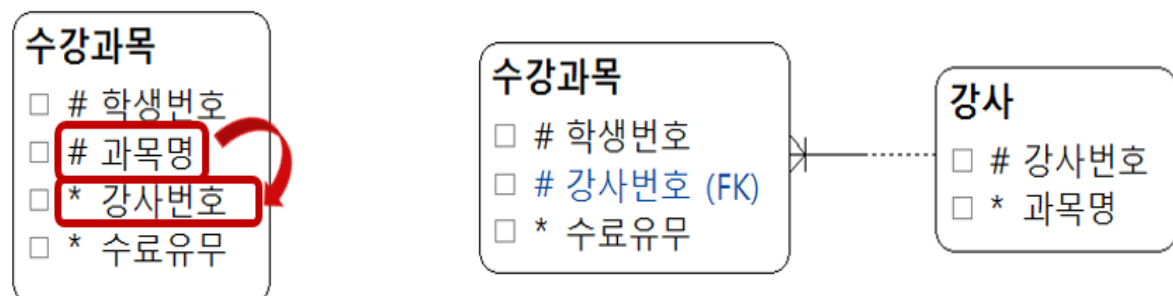
- 후보 식별자 속성과 일반 속성 간의 종속성을 판단하여 처리한다.

3-2-3. 3 정규형(3NF)



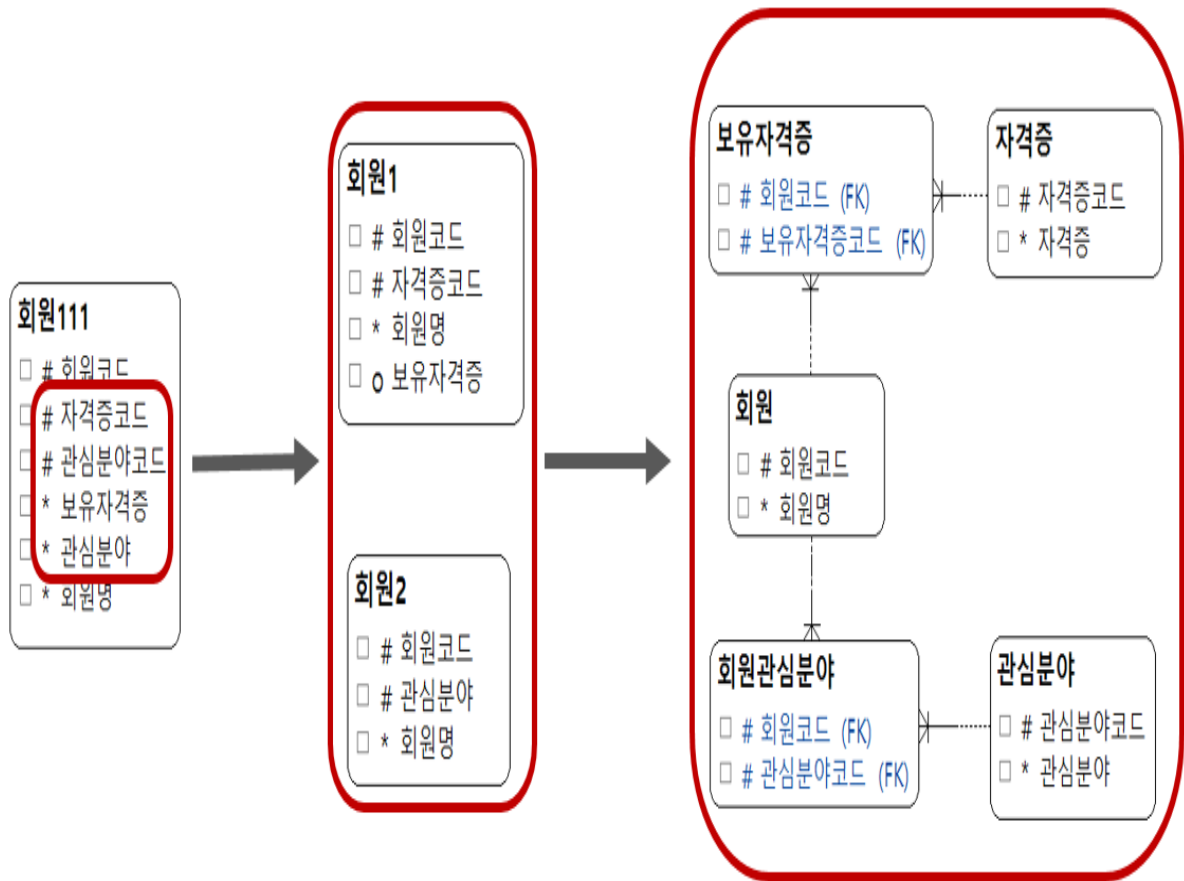
- 이행적 종속성(Transitive Dependency)을 제거하고 일반속성 간에 종속관계가 없도록 처리한다.
- 일반 속성에 결정자 속성과 종속자 속성이 있는 경우 다른 엔티티로 구별하여 분리한다.

3-2-4. 보이스코드 정규형(BCNF)



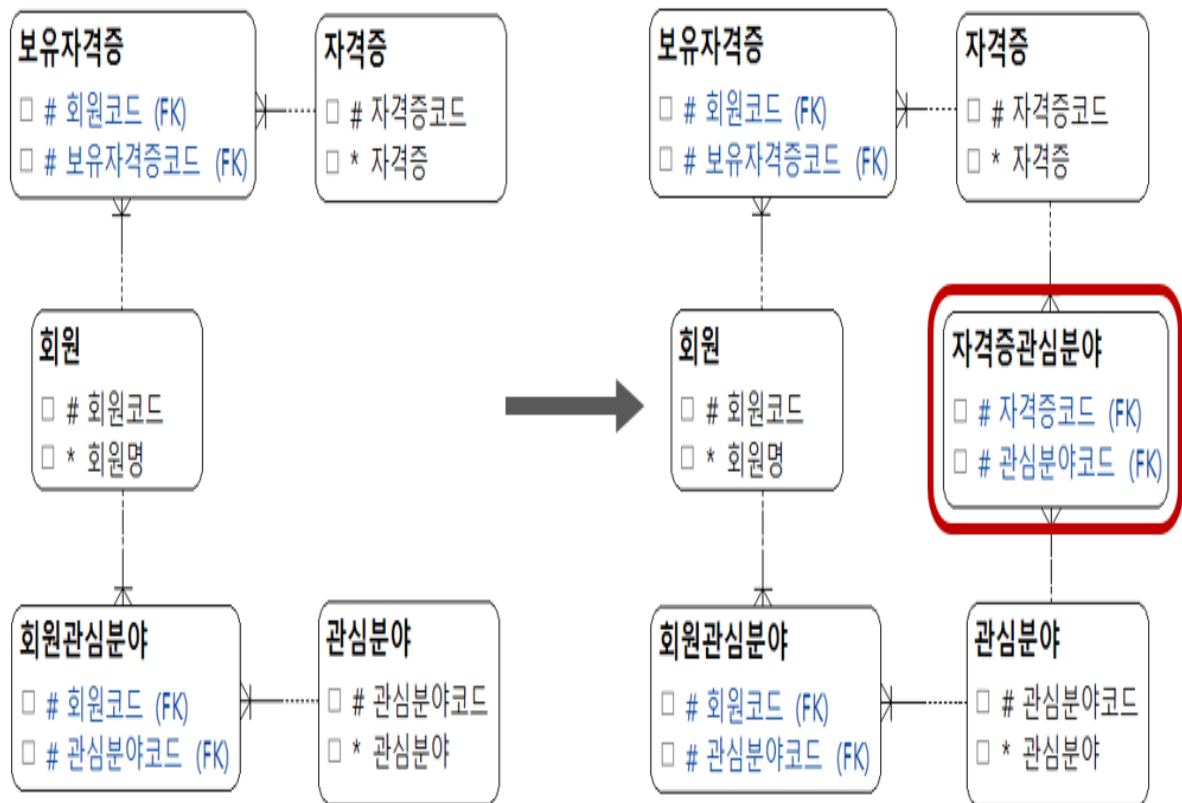
- 모든 결정자는 주식별자가 되도록 한다.
- 결정자 속성이 일반 속성이며 종속자 속성이 주식별자에 포함되어 있다.

3-2-5. 4 정규형(4NF)



- 다가 종속인 속성들간에 서로 직접적인 연관이 없으면 다가 종속인 속성들은 새로운 엔터티로 분해하여 다가종속 관계를 제거한다.
- 보유자격증과 관심분야는 각각 다가 속성이면서 서로 관련이 없으므로 엔터티를 분리한다.

3-2-6. 5 정규형(5NF)



- 조인해서 원래의 릴레이션으로 복원할 수 있도록 분해하여 조인 종속성을 제거한다.
 - 무손실 조인(Lossiess Join): 하나의 릴레이션을 여러 개의 릴레이션으로 분해하고 공통 속성으로 조인하여 데이터 손실 없이 다시 원래의 릴레이션으로 복원할 수 있음
 - 비부가적 조인: 하나의 릴레이션을 여러 개의 릴레이션으로 분해하고 원본 릴레이션에 없는 데이터가 존재하지 않음
 - 5정규형은 무손실 분해가 되고 필요 없는 데이터가 생기지 않는 비부가적 분해가 된 것이다.