



# 데이터베이스

## 관계의 개념





## 학습목표

- ➔ 핵심 관계의 정의를 설명할 수 있다.
- ➔ 개체들 간의 관계를 검토할 수 있다.



## 학습내용

- ➔ 핵심 관계의 정의
- ➔ 관계의 검토



## 핵심 관계의 정의



### 관계의 개념

01

#### 관계의 개념

##### 관계(Relationship)의 사전적 정의

- 상호 연관성이 있는 상태

##### 데이터 모델에 대입한 정의

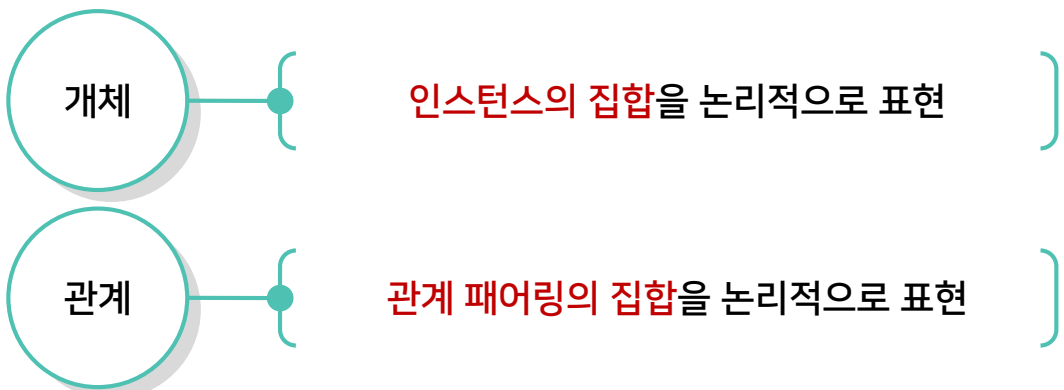
- 개체 사이의 논리적인 연관성으로서 존재의 형태나 행위로서 서로에게 연관성이 부여된 상태
- 관계는 개체 간 연관성을 표현하기 때문에 개체의 정의에 따라 영향을 받기도 하고, 속성 정의 및 관계 정의에 따라서도 다양하게 변할 수 있음

##### 유의사항

- 개체 안에 인스턴스가 개별적으로 관계를 가지는 것이고, 이것의 집합을 관계로 표현
- 개별 인스턴스가 각각 다른 종류의 관계를 가지고 있다면 두 개체 사이에 두 개 이상의 관계가 형성될 수 있음

##### 관계 패어링(Relationship Paring)

- 각각의 개체의 인스턴스들은 자신이 관련된 인스턴스들과 관계의 어커런스(Occurrence)로 참여하는 형태





## 핵심 관계의 정의



### 관계의 개념

01

#### 관계의 개념



- 관계의 표현
  - ▶ 이항 관계(Binary Relation)
  - ▶ 삼항 관계(Ternary Relation)
  - ▶ N항 관계(N-ary Relation)
    - ※ 실제로 있어서 삼항 관계 이상은 잘 나타나지 않음

데이터 모델에서 관계가 제대로 설정되지 않을 경우  
실세계의 업무 흐름이 정확하게 데이터 모델에 반영되지 않아 문제가 발생함



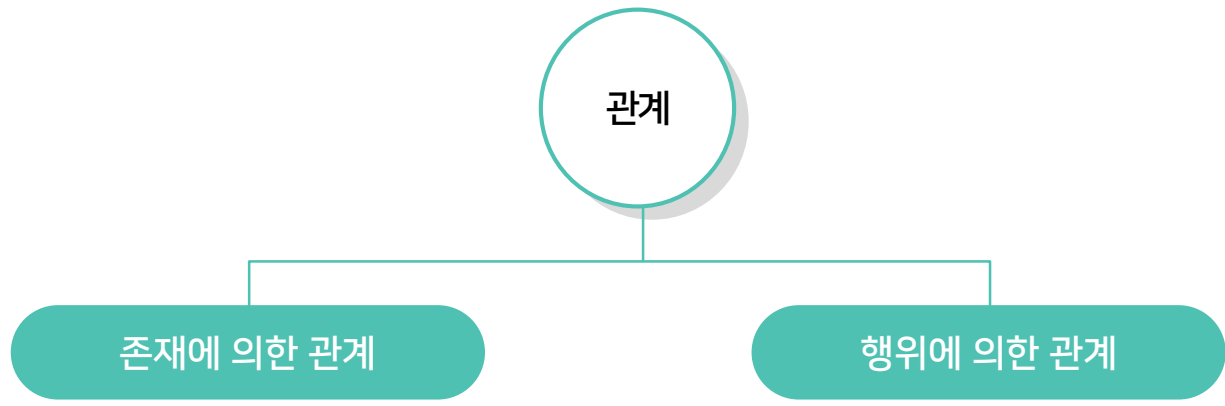
## 핵심 관계의 정의



### 관계의 개념



### 관계의 분류



관계를 연결함에 있어서 어떤 목적으로 연결되었느냐에 따라 분류



### 핵심 관계의 정의



#### 관계의 개념

#### 02 관계의 분류

##### 존재에 의한 관계

» 예 | 연구 팀에 김진형 사원이 소속되어 있다.

'소속된다'라는 의미는 행위에 따른 이벤트에 의해 발생하는 의미가 아니고 그냥 김진형 사원이 연구팀에 소속되어 있기 때문에 나타나는 것

→ 존재의 형태에 의해 관계가 형성

##### 행위에 의한 관계

» 예 | 고객이 주문하여 주문 개체(주문번호)가 생성된다.

주문 개체는 고객의 주문이라는 행위에 의하여 발생

→ 고객과 주문 개체 사이의 관계는 행위에 의한 관계



## 핵심 관계의 정의

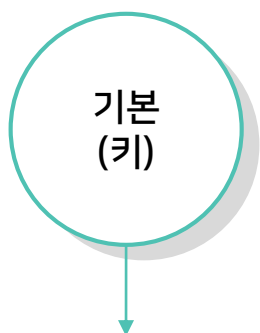


### 관계의 개념



### 02 관계의 분류

#### 개체 타입의 분류



주로 존재에 의한  
관계가 형성됨



이벤트가 발생함으로써 생성되는 개체 타입의 성격을  
지녔으므로 행위에 의한 관계가 많이 형성됨



## 핵심 관계의 정의



### 관계의 개념

03

### 관계의 명명

■ 각 관계에는 두 개의 멤버십이 있으며, 각 멤버십에 의해 **두 가지 관점으로 표현됨**

- 상호 연관성이 있는 상태

예 | 연구부서는 김진형 사원을 포함한다. 김진형 사원은 연구부서에 소속된다.

■ **관계 시작점**(The Beginning): 개체에서 관계가 시작되는 편

■ **관계 끝점**(The End): 관계를 받는 편

- 관계의 시작점과 끝점 모두 관계의 이름을 가져야 함
- 참여자의 관점에 따라 관계 이름은 능동적(Active)이거나 수동적(Passive)으로 명명함

■ 관계명은 명명 규칙에 따라 작성

01 애매한 동사를 피한다.

» 예 | '관계된다', '관련이 있다', '이다', '한다' 등은 구체적이지 않아 어떤 행위가 있는지 또는 두 참여자간 어떤 상태가 존재하는지 파악할 수 없음

02 현재형으로 표현한다.

» 예 | '수강을 신청했다', '강의를 할 것이다' 등의 과거 또는 미래형으로 표현해서는 안되고, '수강 신청한다', '강의를 한다' 등 현재형으로 표현해야 함





## 핵심 관계의 정의



### 다양한 관계

#### 01 관계의 차수(Cardinality)

- 두 개의 개체간의 관계에서 **참여자의 수를 표현**하는 것
- 가장 일반적인 관계 차수 표현방법

1:M 관계

1:1 관계

M:M 관계

- 가장 중요하게 고려해야 할 사항
  - 관계의 시작점과 끝점 모두 관계의 이름을 가져야 함

1:M 관계 ...>

관계에 참여하는 각각의 개체는 관계를 맺는 다른 개체에 대해 **하나 혹은 그 이상의 수와 관계를** 가지고 있는 것

1:1 관계 ...>

관계에 참여하는 각각의 개체는 관계를 맺는 다른 개체에 대해 **단지 하나의 관계만**을 가지고 있는 것

M:M 관계 ...>

관계에 참여하는 각각의 개체는 관계를 맺는 다른 개체에 대해 **하나 혹은 그 이상의 수와 관계를** 가지고 있는 것(반대 방향도 동일)



## 핵심 관계의 정의



### 다양한 관계

#### 01 관계의 차수(Cardinality)

##### 1:1 관계

- 개체 타입과 개체 타입이 독립적으로 분리되면서 **상호 간에 한가지 관계만 성립**하는 형태
- 특징
  - ▶ 현실에서 매우 드물게 나타나는 형태
  - ▶ 업무의 흐름에 따라 데이터가 설계된 형태에서 많이 나타남
  - ▶ 개체의 수직 분할 시에 많이 나타남
- 필수-선택 형태
  - ▶ 한쪽에 있는 개체는 대응되는 반대쪽의 개체가 반드시 존재해야 하지만 우측 개체와 대응되는 좌측의 개체는 없을 수도 있다는 것
- 필수-필수 형태
  - ▶ 이 관계를 수학적으로 풀어보면 두 개체는 같은 개체라는 것이 증명됨
  - ▶ 수학에서 사용하는 필요충분 조건은  $P \rightarrow Q$ 이고,  $Q \rightarrow P$ 이면,  $P = Q$  라는 법칙
    - 서로 필요충분 조건을 만족하면 두 개는 서로 동치(同値)라는 것을 의미
  - ▶ 대부분 데이터 양을 감안하여 수직분할을 한 경우가 대부분이며, 논리적 관점에서 보면 하나의 데이터 집합임
- 선택-선택 형태
  - ▶ 자주 발생되지 않는 형태



### 다양한 관계

#### 01 관계의 차수(Cardinality)

##### 1:M 관계

- 가장 흔하게 나타나는 매우 **일반적인 형태**  
(부모와 자식과의 관계처럼 계층적인 구조로 이해)
- 특징
  - 가장 흔하게 나타나는 관계의 형태
  - 내가 참조할(부모 쪽) 정보는 나에 대해서 반드시 하나만 존재해야만 참조할 때 내 집합에 변화가 생기지 않으므로 내가 참조하는 개체는 반드시 1쪽이 될 수 밖에 없음
- 필수-선택 형태
  - 현실 세계에서 그리 많은 경우는 아니지만 가끔 발생하는 형태
  - 주문과 주문 아이템의 관계에서 하나의 주문이 여러 개의 아이템을 포함하고 있는 경우, 주문 아이템이 없는 주문은 업무적으로 의미가 없고 마찬가지로 주문이 없는 주문 아이템도 의미가 없음
- 선택-필수 형태
  - 현실 세계에서 가장 흔한 형태
  - 부서와 직원 간의 관계에서 모든 직원은 반드시 하나의 부서에 소속되어야 하지만 직원이 하나도 없는(신생) 부서도 존재할 수 있음



## 핵심 관계의 정의



### 다양한 관계

#### 01 관계의 차수(Cardinality)

##### 1:M 관계

###### 필수-선택 형태

- ▶ one to many 관계에서 현실적으로 드문 형태

» 예 | 여러 번의 납품 건에 대해서 한번에 모아 대금 청구를 하는 경우  
선 납품 후 대금 청구하는 계약 조건에서 아직 대금이 청구되지  
않는 납품 건은 청구번호가 정해지지 않은 경우

###### 선택-선택 형태

- ▶ 현실 세계에서 흔한 형태 중의 하나
- ▶ 관계의 선택성이 증가할수록 모델의 모호성도 증가하므로 될 수 있으면  
선택성을 해소해야 함

» 예 | 고객-예약 관계: 고객 없이도 여행사가 미리 예약을 잡아  
놓을 수 있음

##### M:M 관계

###### 관계를 가진 양쪽 당사자 모두에서 1:M 관계가 존재할 때 나타나는 형태

###### 특징

- ▶ 의외로 아주 빈번하게 발생하는 관계이지만 데이터 모델링이 완료된  
후에는 M:M은 존재하지 않음  
→ M:M 관계가 아직 덜 풀려진 형태라는 것을 의미

- ▶ 관계가 해소되면서 새로운 개체(Relation Entity/Intersection Entity)를 생성

» 예 | 상품과 주문 사이의 M:M 관계



## 핵심 관계의 정의



### 다양한 관계

#### 02 관계의 선택성(Optionality)

관계에 참여하는 개체집합의 개체 인스턴스가  
항상 관계에 참여해야 하는가?



필수참여 관계

VS



선택참여 관계





## 핵심 관계의 정의



### 다양한 관계

02

### 관계의 선택성(Optionality)

- 만약, 지하철의 문이 닫히지 않았는데 지하철이 떠난다면 무슨 일이 발생할까?
  - “반드시 지하철의 문이 닫혀야만 지하철은 출발한다.”



지하철 출발과 지하철 문 닫힘은  
필수(Mandatory)적으로 연결 관계가 있는 것

- ▶ 지하철 안내방송 시스템에서 지하철의 출발을 알리는 안내방송은 지하철의 출발과 상관없이 방송해도 큰 문제가 발생하지 않음
  - 안내방송 시스템이 고장이 나도 지하철 운행에는 큰 영향을 주지 않음
  - 즉, 지하철의 출발과 안내방송은 관련은 있지만 서로가 선택적인 관계(Optional)임



## 핵심 관계의 정의



### 다양한 관계

#### 02 관계의 선택성(Optionality)

- 필수참여는 참여하는 모든 참여자가 반드시 관계를 가지는, 타개체의 참여자와 연결이 되어야 하는 관계

» 예 | 주문서와 주문 목록, 상품과 주문 목록의 관계

[주문서와 주문 목록]

- ▶ 주문서는 반드시 주문 목록을 가져야 함
- ▶ 주문 목록이 없는 주문서는 의미가 없음



필수참여 관계



[상품과 주문 목록]

- ▶ 상품은 주문이 될 수도 있고 주문이 되지 않은 상품이 있을 수도 있음



선택참여 관계

- 선택참여 된 항목은 물리 속성에서 외래키(Foreign Key)로 연결될 경우 널(Null)을 허용할 수 있는 항목이 됨
  - 만약 선택참여로 지정해야 할 관계를 필수참여로 잘못 지정하면 애플리케이션에서 데이터가 발생할 때 반드시 한 개의 트랜잭션으로 제어해야 하는 제약사항이 발생

설계단계에서 필수참여와 선택참여는 개발시점에서 업무 로직과 직접적으로 관련된 부분이므로 반드시 고려되어야 함



### 핵심 관계의 정의



#### 다양한 관계

#### 02 관계의 선택성(Optionality)

- 관계의 선택 사양은 관계를 통한 **상대방과의 업무적인 제약 조건을 표현**하는 것으로서 간단하면서 아주 중요한 표기법임
- 어떻게 설정했는지에 따라 참조 무결성 제약조건의 규칙이 바뀌게 되므로 주의 깊게 모델링을 해야 함

» 예 | 관계가 표시된 양쪽 개체에 모두 “선택참여”가 표시되는 경우

**0:0(Zero to Zero)의 관계**가 될 경우 그 관계는 잘못될 확률이 많음  
→ **관계 설정이 잘못되었는지 반드시 검토 필요**





## 관계의 개념



### 핵심 관계의 정의



#### 핵심 관계의 파악

##### 01 관계의 정의 방법

- 01 관계를 정의할 수 있는 두 개의 핵심 개체 간의 관계를 도출하기
- 02 핵심 개체 간의 관계성을 정의하기
- 03 식별 관계를 설정하기
- 04 관계의 성격을 구분하여 관계를 확정하기



## 핵심 관계의 정의



### 핵심 관계의 파악

#### 01 관계의 정의 방법

##### 01 관계를 정의할 수 있는 두 개의 핵심 개체 간의 관계를 도출하기

- 항상 두 개체 간에 존재하며, 업무적 연관성을 가지고 **양방향의 관점**을 가지고 있는 **핵심 개체를 선발**
  - ▶ 업무 기술서, 장표, 인터뷰 정리 문서에서 동사 구분
  - ▶ 도출된 핵심 개체와 관계를 이용하여 관계 정의를 작성
  - ▶ 관계를 더 세분화하고 정확하게 도출하는 작업을 수행

##### 02 핵심 개체 간의 관계성을 정의하기

- 데이터 양방향의 업무 규칙을 표현하는 것으로 관계를 통하여 정보의 활용 가치를 상승할 수 있도록 **정합성을 유지**
- **기수성(Degree/Cardinality)**의 관계 형태를 검토하여 정의

하나 이상 (Many)	<ul style="list-style-type: none"> <li>• <b>하나 이상인 것이 적어도 한 가지는 존재</b>하고 있다는 것을 의미 → 1:M의 관계에는 1:1의 관계가 포함되어 있다는 것을 의미</li> <li>• 향후 관계 형태를 검토할 때도 하나 이상인 경우가 한 가지라도 있는지에 대해 집중적으로 검토함으로써 이 관계를 분석해야 함</li> </ul>
단 하나 (Only One)	<ul style="list-style-type: none"> <li>• 단 하나뿐임을 규명하는 것</li> <li>• 하나 이상인 경우가 하나라도 존재하는지를 검토함으로써 관계를 분석할 수 있음</li> </ul>



## 핵심 관계의 정의



### 핵심 관계의 파악

#### 01 관계의 정의 방법

#### 02 핵심 개체 간의 관계성을 정의하기

- **선택성(Optionality)**의 관계 형태를 검토하여 정의

일반적인 형태	<ul style="list-style-type: none"> <li>• 주로 참조하는 개체는(M쪽) 참조되는 개체(1쪽)가 반드시 존재해야 하는 경우가 많이 나타남</li> <li>• 반대로 1쪽 개체는 M쪽 개체와 반드시 관계를 맺지 않아도 되는 경우가 가장 일반적인 형태라고 할 수 있음</li> </ul>
바람직한 형태	<ul style="list-style-type: none"> <li>• M쪽 관계의 선택 사양을 필수 관계로 만들려고 노력해야 함</li> <li>• 부모 개체에 개체가 존재하지 않는 자식 개체에 개체가 많이 발생한다면 정보의 정합성에 많은 문제점이 생기게 됨</li> </ul>

- **관계명**의 정의

두 개의 관계 멤버십에 각각 부여	<ul style="list-style-type: none"> <li>• 각자 상대방 입장에서의 관계명을 기술</li> </ul>
현업에서 사용하는 간결한 동사형으로 표현	<ul style="list-style-type: none"> <li>• M두 개체 타입 간의 업무적 연관성을 나타내는 이름을 부여하고 현재 시제를 사용</li> </ul>
업무적 의미가 없거나 애매모호한 용어 배제	<ul style="list-style-type: none"> <li>• 예: ~(관계가) 있다. ~(관련이) 있다. ~한다. ~ 할 수 있다 등</li> </ul>



## 관계의 개념



## 핵심 관계의 정의



### 핵심 관계의 파악

#### 01 관계의 정의 방법

##### 03 식별 관계를 설정하기

- 식별(Identifying) 관계: 강한 개체와 약한 개체의 관계

##### » 예 | 존재 존속 관계

- 식별(Identifying)  
관계를 통하여 이주한 부모의 식별자가 자식의 주식별자의 일부가 되는 경우에 설정
- 비식별(Non Identifying)  
부모의 주식별자가 자식의 Non-Key 영역으로 이주하고 자식을 식별하는 데 관계하지 않음

##### 04 관계의 성격을 구분하여 관계를 확정하기

- 관계 도출 소스
  - ▶ 개체 타입 도출에 사용되었던 6가지 방법과 동일
  - ▶ 단, 관계를 업무 기술서나 장표 등에 정확하게 기술되는 경우는 거의 없으므로 대부분의 경우 내용을 보고 관계를 유추하는 작업이 필요함
  - ▶ 개체 간의 관계 설정은 고객과의 회의가 필수적으로 선행되어야 함



## 핵심 관계의 정의



### 핵심 관계의 파악

#### 01 관계의 정의 방법

##### 04 관계의 성격을 구분하여 관계를 확정하기

###### ① 업무 기술서, 장표, 인터뷰 정리 문서 등에서 동사 구분

- 개체 타입이 명사형으로 구분되듯이 **관계는 동사형으로 구분**할 수 있음

» 예 | 기술 대학원 학사 관리에 대한 시나리오

강사는 여러 개의 강좌를 **강의할 수 있다**.

기술 대학원에서는 여러 명의 강사를 **기록하고 관리한다**.

기술 대학원에서 과목당 **개설한** 강좌는 강사 1명이 강의를 **진행한다**.

→ 동사: 강의한다, 기록하고 관리한다, 개설한다

- 도출된 동사는 개체 타입과 연관하여 관계 정의서에 표현하거나 개체-관계 모델(ERD)로 표현한 후 정제 과정을 거침



## 핵심 관계의 정의



### 핵심 관계의 파악



#### 관계의 정의 방법

04 관계의 성격을 구분하여 관계를 확정하기

② 도출된 개체 타입과 관계를 이용하여 관계 정의서를 작성한다.

» 예 | 관계 정의서의 예

개체 타입	관계 형태	참여 방법	관련 개체 타입
사원	각 사원은 한 부서에 속한다. 각 부서에는 여러 사원이 존재할 수 있다.	필수 선택	부서
	각 사원은 여러 주문을 접수할 수 있다. 각 주문은 한 명의 사원에 의해서만 접수된다.	선택 필수	주문



## 핵심 관계의 정의



### 핵심 관계의 파악

#### 01 관계의 정의 방법

##### 04 관계의 성격을 구분하여 관계를 확정하기

③ 고객에게 질문하여 관계를 좀 더 세분화하고 정확하게 도출하는 작업을 수행한다.

- 관계 도출을 위해서는 고객에게 정확한 질문을 하는 것이 필요함
  - ▶ 기준 개체 하나 / 여러 관련 개체 선택 / 필수참여방법

각 사원은 한 부서에만 소속할 수 있는가?  
 각 부서에는 여러 사원이 항상 소속되어 있는가?

- 다음과 같이 3가지로 나누어 질문하면 좀 더 쉬움
  - ▶ 각 사원은 한 부서에 속합니까? ← 사원과 부서의 참여 형태 및 1:1
  - ▶ 각 사원은 부서에 속하지 않을 수도 있습니까? ← 관계의 선택/필수
  - ▶ 각 사원은 여러 부서에도 소속될 수 있습니까? ← 1:M?

④ 데이터 모델링 도구나 칠판 등을 이용하여 모델을 직접 그려 본다.

- 칠판에 모델을 그려가며 프로젝트 참여 인원들과 토의하는 방법
  - ▶ 의사소통에 도움
  - ▶ 개체 타입 배치 용이
  - ▶ 관계 설정도 쉽게 고칠 수 있음



## 핵심 관계의 정의



### 핵심 관계의 파악

01

#### 관계의 정의 방법

04

#### 관계의 성격을 구분하여 관계를 확정하기

⑤ 고객과 질문하고 협의하여 모델을 검토한다.

- 개체-관계 모델(ERD)를 그리기 전
  - ▶ 관계를 정리하고 도출하기 위하여 고객과 질문하는 시간을 가짐
- 개체-관계 모델(ERD)를 그린 후
  - ▶ 관계가 정확히 설정되었는지 누락된 것은 없는지 전체 모습을 보면서 고객과 검토
- 고객의 의견을 모두 반영해야 하는 것은 아님
  - ▶ 현업의 업무 전문가는 현재 업무 흐름에 익숙해져 있음





## 핵심 관계의 정의



### 핵심 관계의 파악

#### 02 관계 정의의 예

여러 부서에 소속된 각 사원은 고객으로부터 필요한 제품을 주문 받아 처리

개체 타입	관계 형태	참여 방법	관련 개체 타입
고객	각 고객은 여러 번 주문할 수 있다. 각 주문은 반드시 한 고객에 의하여 주문된다.	선택 필수	주문
제품	각 제품은 여러 개 주문을 접수 받을 수 있다. 각 주문은 반드시 한 제품을 주문한다.	선택 필수	주문
사원	사원 한 명은 여러 개 주문을 접수 받을 수 있다. 각 주문은 한 사원에 의해서만 접수 받는다.	선택 필수	주문
부서	각 부서에는 여러 명의 사원을 포함한다. 각 사원은 한 부서에만 포함된다.	선택 필수	사원



### 잘못된 관계의 검토

#### 01 관계의 개념

- 개체 타입 간의 업무적 연관성을 정의한 것
- 사용자 인터뷰, 업무 흐름도, DFD 등의 데이터 흐름에서 개체 타입 간의 관계를 파악할 수 있음

#### 도출된 관계에 대하여 검증해야 할 내역

- ① 개체 타입 간에 M:M 관계가 없는가?
- ② 개체 타입 간의 관계는 업무 흐름과 규약에 일치하는가?
- ③ 업무 흐름에 비추어 도출되지 않은 관계는 없는가?
- ④ 관계에 대한 표현이 적절한 수준에서 이루어졌는가?



## 관계의 개념



## 관계의 검토



### 잘못된 관계의 검토

#### 02 검증 내역

Q

개체 타입 타입 간의 관계는 업무 흐름과 규약에 일치하는가?

- 개체 타입 간의 관계(선택성 관련)
  - ▶ 모델링 규칙 상 일반적인 관계  
: 한쪽은 필수 관계, 상대 쪽은 필수 관계나 선택 관계 둘 중 하나
    - \* 한 쪽을 필수 관계로 지정하는 이유  
참조하는 쪽에서 물리 테이블을 생성하면 FK로 참조되기 때문  
→ **FK 중 참조 무결성을 보장하기 위함**
  - ▶ 모델링 규칙상 정상적이지는 않지만 업무적으로 발생할 수 있는 관계로 양쪽 모두 선택 관계 되는 경우가 있음

즉, 주식별자에 의한 식별 관계가 아닌 속성으로만 나타나는 **비식별 관계**  
(두 개체 타입 간의 관계는 존재하지만 완전한 의존 관계는 아님)

#### » 예 | 급여와 전표 관계

- ▶ 각 급여 마다 전표가 발행되며 하나의 전표는 하나의 급여만을 표현함

전표는 매출, 매입 등과 같은 개체 타입과도 관계가 있음

전표(전표번호) : 급여(사번, 전표번호(FK)) → 1(필수) : 1(필수)

- ▶ 실제 업무 흐름  
→ 급여가 계산되면 급여 테이블에 등록되고 급여 확정 시 전표가 자동 발생됨
- ▶ 양쪽 다 필수 관계처럼 보이나 실제 업무 시 급여가 먼저 만들어짐
- ▶ 이는 급여 생성 시 전표가 생성되지 않음으로 급여가 필수 관계일 수 없음



### 잘못된 관계의 검토

#### 02 검증 내역

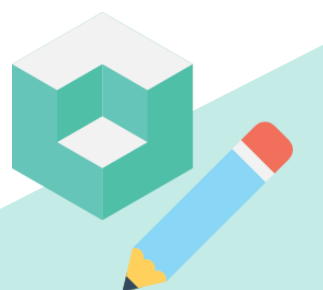
Q

개체 타입 타입 간의 관계는 업무 흐름과 규약에 일치하는가?

- 데이터 모델링에서 양쪽 모두 선택 관계가 되는 것은 좋은 모습은 아니지만 현업에서는 원론적 규칙을 뛰어넘는 사례가 많음



해당 업무 상황에 맞는 유연한 모델링 필요!





## 관계의 개념



## 관계의 검토



### 잘못된 관계의 검토

#### 02 검증 내역

Q

업무 흐름에 비추어 도출되지 않은 관계는 없는가?

- 업무 흐름에 비추어 관계가 도출되지 않는 상황
  - ▶ 업무적으로 명확히 정의되지 않아 개체 타입만 도출하고 추후 관계 정의를 하려는 경우
    - 모델러(modeler)가 인식하고 있는 경우 특별한 문제 없음
  - ▶ 단위 시스템 간의 업무 연계가 정의되지 않은 경우
    - 큰 문제가 될 소지가 있음
- 프로젝트 개발은 단위 팀 단위로 이루어지기 때문에 **단위 시스템 내의 연계가 대부분 도출됨**
- 단위 시스템 개발 조직은 타 시스템과의 연계에 일반적으로 관심을 두지 않음
  - 시스템 통합 조직에 의한 모델링 통합 작업 필요



## 관계의 개념



## 관계의 검토



### 잘못된 관계의 검토

#### 02 검증 내역

Q

관계에 대한 표현이 적절한 수준에서 이루어졌는가?

- 관계 표현 시 어려운 점은 **관계를 어디까지 표현**하느냐 하는 것임
  - 개체 타입 간의 PK만 보면 모든 개체 타입들과 관계를 가지고 있는 것으로 보이고 이를 다 표현하면 그물망처럼 됨
    - ▶ 코드 및 통계 개체 타입처럼 업무 흐름에 관여하는 모습을 직접적으로 나타내지 않아도 상식적으로 알 수 있는 개체 타입의 관계를 생략
    - ▶ 나머지 업무 흐름에 대해서는 관계를 정의하여 표현해야 함
  - 주요 사례 - **코드 및 통계 개체 타입과의 관계 연결**
    - ▶ 코드성 개체 타입과의 관계를 모두 나타내면 업무 규칙을 나타내기는 하나 매우 복잡한 데이터 모델이 됨
    - ▶ 코드성 개체 타입과 통계성 개체 타입의 관계는 과감히 생략함
  - 주요 사례 - **PK를 상속받은 개체 타입과 조상 개체 타입 간의 관계 연결**
    - ▶ 데이터 모델에서 부모는 관계를 주식별자로 상속하는 식별자 관계를 가지고 있고, 자식은 또 다른 자식 개체 타입과 비식별자 관계를 가지는 경우가 있음
    - ▶ 가장 마지막 손자는 부모로부터 주식별자를 받았음에도 원래 주식별자를 생성했던 자신의 할아버지 개체 타입과 관계를 가지도록 하는 경우
- » 예 | 사원(사원번호) , 급여이력(사원번호(FK)), 수당이력(사원번호(FK))  
→ 사원과 수당이력 사이에 관계를 표시한 경우



## 관계의 개념



## 관계의 검토



### M:M 관계 해소

#### 01 관계 타입

- 프로젝트를 진행하는 도중 많이 발생하는 개체 타입
- 관계 타입의 특징
  - 도출되지 않은 업무 규칙이 새롭게 발견되었거나 이력 관리와 같은 업무 규칙을 추가하는 작업에서 많이 도출됨
  - 관계 타입을 발생시키는 업무 규칙이 발견되었다면, 이유를 불문하고 **개체-관계 모델(ERD)**에 표현
  - 관계 타입이 반드시 별도로 존재해야 하는지 검토

#### 02 관계 타입 분리

- M:M 관계
  - 개체 타입의 관계 도출 시 상호 간의 관계가 **1:M, M:1로 나타나는 경우**

##### » 예 | 학생- 과목

- 학생이 여러 개의 과목을 수강 → 1:M
- 여러 학생이 한 과목을 수강 → M:1
- 이 두 가지 규칙을 합하면 M:M 관계로 나타남



정보 공학에서 M:M 관계는 반드시 정보를 추적할 수 있도록  
**1:M 형식으로 해소해주어야 함**



### M:M 관계 해소

#### 02 관계 타입 분리

##### » 예 | 업무 규칙

한번 납부할 때 여러 개의 요금 고지서를 납부할 수 있고, 또 다른 요금 고지서는 여러 번에 걸쳐 납부할 수 있음

요금과 납부라는 개체 타입이 존재, M:M 관계

- 요금납부라는 관계 타입 생성
- 요금에서 요금납부 → 1:M
- 요금납부에서 납부 → M:1

##### 관계 타입의 명명

- 업무에서 사용하는 적절한 이름이 있다면 이를 사용
- 데이터 모델링 도중 관계에 의해 생성되었다면 이름이 없는 경우가 많음
  - ▶ 개체 발생 순서에 따라서 먼저 발생하는 개체명을 먼저, 나중에 발생하는 개체명을 뒤에 위치하는 형태로 의미 있게 부여하는 것이 좋음
  - ▶ “요금납부”라고 관계 타입을 명명





## 관계의 개념



## 관계의 검토



### M:M 관계 해소

03

#### 주식별자 통합

■ M:M 관계 해소를 관계타입으로 할 경우 개체-관계 모델(ERD)이 복잡해짐

- 이를 막기 위해 PK를 이용하여 하나의 개체 타입으로 통합 가능

- ▶ 요금의 주식별자: 요금번호
- ▶ 납부의 주식별자: 납부번호  
→ 납부의 주식별자 : **납부번호 + 요금납부 순차 번호**

» 예 | “한번 납부할 때 여러 개의 요금 고지서를 한번에 납부할 수 있고, 또 하나의 요금 고지서는 여러 번에 걸쳐 납부할 수 있다.”의 경우

하나의 요금 고지서를 여러 번에 걸쳐 납부할 수 있다는 것은  
요금에 대한 납부를 여러 개 올 수 있음으로 만족

- 한번 납부할 때 여러 개의 요금에 대하여 납부할 수 있다는 것은 어떻게 만족하나?
- ▶ 납부 개체의 PK에 납부번호와 요금납부 순차번호로 되어 있는 것이 답임 즉, 동일한 납부번호에 대하여 요금납부 순차번호가 요금고지서 만큼 증가하는 것을 대신함
- ▶ 한번의 납부로도 여러 개의 요금고지서의 납부가 가능한 형태가 됨



## 관계의 개념



## 관계의 검토



### M:M 관계 해소

03

#### 주식별자 통합

##### 주식별자 통합의 유의점

유의점 1



- 부모(요금)에 대하여 자식이 생사를 같이 하는 지 (존재 종속인지) 검증한 후 해야 함 (존재 종속이 아니면 쓰기 어려움)

유의점 2



- 통합되는 개체 타입의 속성이 많지 않고, 데이터 수정이 많지 않으며 읽는 작업이 많이 발생하는 개체 타입의 경우에 적합함

유의점 3



- 속성이 많을 경우 데이터의 불필요한 반복, 즉, 중복이 증가하여 좋지 않음

유의점 4



- 수정이 잦은 경우에 여러 개의 튜플에 대한 작업이 발생되어(동일 납부번호를 가지는 튜플들) 좋지 않음

##### 주식별자 통합의 장점

01

개체 타입이 통합되면 데이터 모델의 복잡도는 감소함

02

물리적 테이블에서 데이터를 가져올 때 여러 개의 테이블에 대한 조인을 하지 않음

03

대체적으로 업무를 간단히 하거나 트랜잭션 수가 적지만 반드시 업무에 필요한 경우 주식별자 통합을 적용하는 경우가 많음



### M:M 관계 해소

#### 04 부모 개체 타입에 대한 속성 추가

» 예 | 하나의 요금 고지서에 대한 분할 납부는 최대 2번까지 만 가능하다는 업무 규칙이 있을 경우 관계타입 분리 없이 표현 가능

요금(요금번호) → 요금(요금번호, 납부번호1 (FK), 납부번호2(FK))

#### ● 유의사항

- ▶ 해당 업무 규칙의 최대값이 지정되어 있고, 최대값이 변경될 가능성이 적은 것이어야 함
- ▶ 최대값이 5개나 된다면, 지속적으로 변경될 가능성이 있다면 요금 개체 타입의 속성이 변경되어야 하고 이에 따라 응용프로그램이 수정되는 경우가 발생할 수 있음
- ▶ 하나의 개체 타입 속성으로 통합하기 위해서는 반드시 **업무 규칙 발생 최대 수와 최대값의 변경 가능성을 조사**하여 확실한 경우에만 적용해야 함

## 1 핵심 관계의 정의

- ✓ 관계란 개체 사이의 논리적인 연관성으로서 존재의 형태로서나 행위로서 서로에게 연관성이 부여된 상태
- ✓ 관계의 차수(Cardinality)
  - 1:1 관계: 관계에 참여하는 각각의 개체는 관계를 맺는 다른 개체에 대해 단지 하나의 관계만을 가지고 있는 것
  - 1:M 관계: 관계에 참여하는 각각의 개체는 관계를 맺는 다른 개체에 대해 하나나 그 이상의 수와 관계를 가지고 있는 것
  - M:M 관계: 관계에 참여하는 각각의 개체는 관계를 맺는 다른 개체에 대해 하나나 그 이상의 수와 관계를 가지고 있는 것(반대의 방향도 동일)
- ✓ 관계의 선택성(Optionality)
  - 필수참여(전체 참여): 관계에 참여하는 개체 집합의 모든 개체는 타개체와 연관성을 지님
  - 선택참여(부분 참여): 관계에 참여하는 개체 집합의 일부 개체만 타개체와 연관성을 지님

## 2 관계의 검토

- ✓ M:M 관계 해소
  - 관계 타입 분리: 관계를 하나의 타입으로 분리
  - 주식별자를 통합
  - 부모 개체 타입에 속성을 추가