1 관계 데이터베이스 개요와 설계 절차

01 관계 데이터베이스 개요와 설계 절차

1 데이터 모델링

데이터 모델

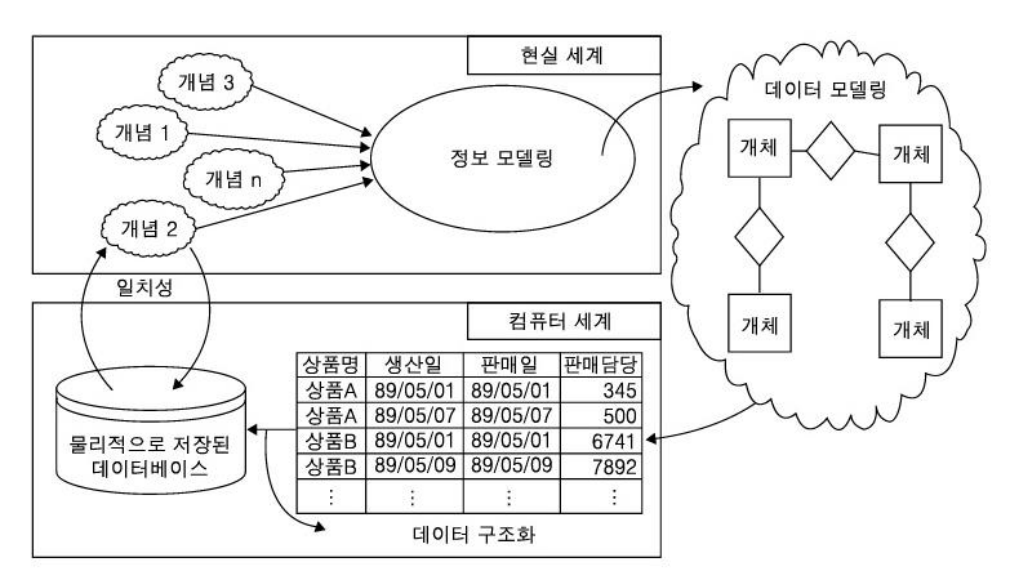
현실 세계의 정보들을 컴퓨터에 표현하기 위해 단순화, 추상화하여 체계적으로 표현한 모형

- 복잡한 실 세계를 단순화하여, 실 세계에 존재하는 개체들을 식별하고 개체와 개체 사이의 관계를 정의함으로써 컴퓨터 상의 데이터베이스를 추상화된 개념으로 이해하기 쉽게 할 뿐만 아니라 사용자들 사이의 의사소통을 원활히 할 수 있도록 도와주는 도구

01 관계 데이터베이스 개요와 설계 절차

1 데이터 모델링

데이터 모델의 예



상품명	생산일	판매일	판매담당
상품A	89/05/01	89/05/01	345
상품A	89/05/07	89/05/07	500
상품B	89/05/01	89/05/01	6741
상품B	89/05/09	89/05/09	7892
⋮	⋮	⋮	⋮

※ 출처 : 데이터베이스 개론과 실습, 정선호 저, 한빛미디어, 2004년

01 관계 데이터베이스 개요와 설계 절차

2 데이터 모델의 분류

물리적 데이터 모델 (저수준 데이터 모델)	▪ 어떻게 데이터가 컴퓨터에 저장되는지의 세부 사항을 명시하는 개념을 제공
개념적 데이터 모델 (고수준 데이터 모델)	▪ 사용자들이 데이터를 인식하는 방식에 대한 개념을 제공
표현적 데이터 모델 (구현 데이터 모델)	▪ 일반 사용자들이 이해할 수 있는 개념을 제공 ▪ 데이터 저장 구조의 세부 사항을 은폐하지만 컴퓨터 상에서 직접 구현 가능 ▪ 상용 DBMS에서 주로 사용함

01 관계 데이터베이스 개요와 설계 절차

3 3단계-스키마 아키텍처

외부 단계 또는 뷰 단계

- 특정 사용자 그룹이 관심을 갖는 부분을 나타내고 나머지는 은폐함

개념 단계

- 전체 사용자를 위한 데이터베이스의 구조를 기술함
- 엔티티, 관계, 연산, 제약 조건들에 중점

내부 단계

- 저장구조의 세부 사항과 접근 경로를 기술

01 관계 데이터베이스 개요와 설계 절차

3 3단계-스키마 아키텍처

3단계-스키마 아키텍처 목적

논리적 데이터 독립성	▪ 외부 스키마나 응용 프로그램을 변경하지 않으면서 개념 스키마를 변경할 수 있는 성질
물리적 데이터 독립성	▪ 개념 스키마를 변경하지 않으면서 내부 스키마를 변경할 수 있는 성질

01 관계 데이터베이스 개요와 설계 절차

4 관계형 데이터베이스 설계 절차 개요

- 요구사항 수집 및 분석
- **개념적 설계(Conceptual design)**
  : 객체-관계 모델
- **논리적 설계(Logical design)**
  : 관계 모델
- 물리적 설계(Physical design)
  : 물리적 저장구조, 인덱스(해쉬, B+ 트리)

2 고수준의 개념적 데이터 모델

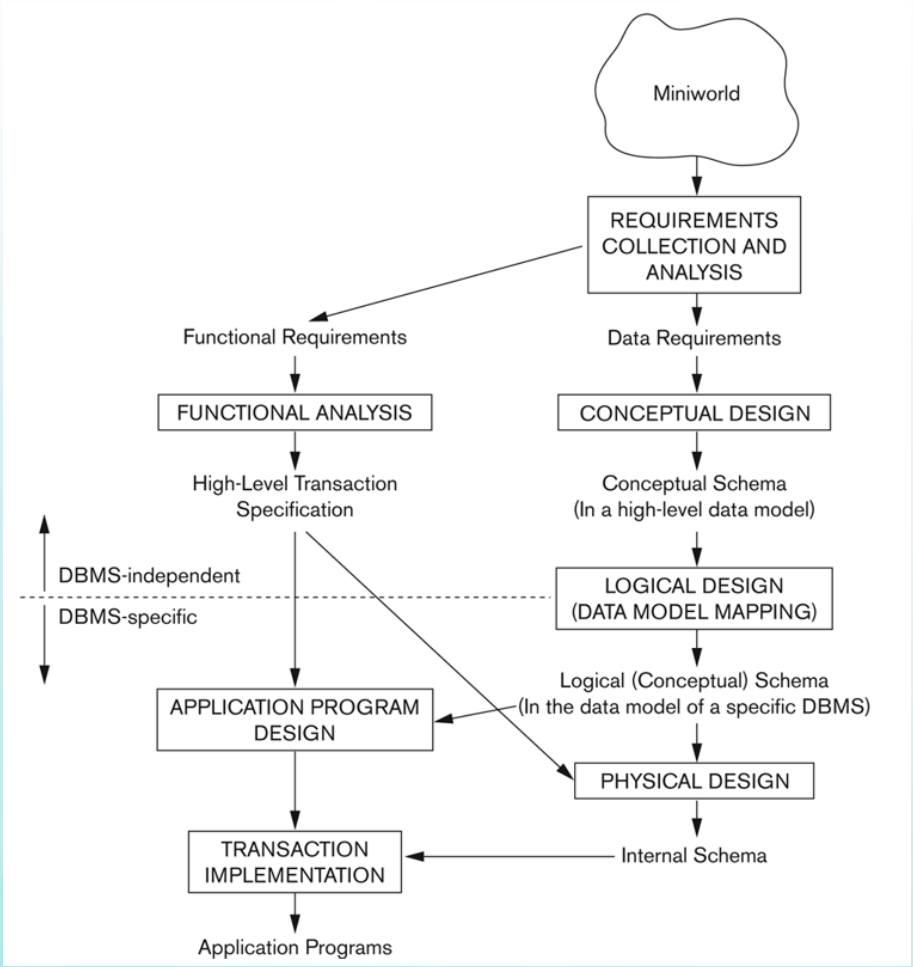
02 고수준의 개념적 데이터 모델

1 데이터베이스 설계의 주요 단계

- 요구사항 수집 및 분석(Requirement Analysis)
- 개념적 설계(Conceptual design)
  : 개념적 스키마, ER Model
- 논리적 설계(Logical design)
  : 논리적(개념적) 스키마, Relational Model
- 물리적 설계(Physical design)

02 고수준의 개념적 데이터 모델

1 데이터베이스 설계의 주요 단계



※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

02 고수준의 개념적 데이터 모델

2 예제 : Company 데이터베이스

요구사항

- 회사는 여러 부서(DEPARTMENT)들로 구성
- 각 부서는 부서명(name), 번호(number), 부서장을 가짐
- 부서장의 부임 날짜(start date)도 유지함

- 한 부서는 여러 개의 프로젝트(PROJECT)들을 관리
- 각 프로젝트는 이름(name)과 번호(number)를 가지며 한 곳(location)에 위치

02 고수준의 개념적 데이터 모델

2 예제 : Company 데이터베이스

요구사항

- 각 사원(EMPLOYEE)의 주민등록번호(social security number), 주소(address), 월급(salary), 성별(sex), 생년월일(birthdate)을 저장

- 각 사원은 한 부서에서 근무하며(works for) 여러 프로젝트에 관여함(work on)
- 각 사원이 각 프로젝트를 위해 주당 근무 시간을 저장
- 각 직원의 직속 상사(direct supervisor)도 유지

02 고수준의 개념적 데이터 모델

2 예제 : Company 데이터베이스

요구사항

- 각 사원은 여러 명의 부양가족(DEPENDENT)들을 가짐
- 각 부양가족에 대해 이름(name), 성별(sex), 생년월일(birthdate), 직원과의 관계(relationship)를 저장

3 엔티티, 엔티티타입, 애트리뷰트, 키

03 엔티티, 엔티티타입, 애트리뷰트, 키

1 엔티티와 애트리뷰트

엔티티는 데이터베이스 내에 표현된 작은 세계에 존재하는 객체 또는 실체

예시)
: EMPLOYEE 홍길동, DEPARTMENT 연구부, PROJECT X-프로젝트 등이 있음

애트리뷰트는 엔티티를 기술하기 위한 속성

예시)
: EMPLOYEE 엔티티는 속성으로 Name, SSN, Address, Sex, BirthDate를 가짐

03 엔티티, 엔티티타입, 애트리뷰트, 키

1 엔티티와 애트리뷰트

- **엔티티**는 자신의 각 애트리뷰트에 대해, 값을 가짐

  예시)
  : 한 특정한 사원 엔티티는 Name= ' 홍길동',
  SSN='1234567', Address ='서울 마포구',
  Sex='M', BirthDate='2009-01-15'를 값으로 가짐

- 각 **애트리뷰트**는 정수, 실수, 문자열과 같이 자신에게 연계된 값 집합(데이터 타입, 도메인)을 가짐

03 엔티티, 엔티티타입, 애트리뷰트, 키

2 애트리뷰트 타입

🔎 단순 애트리뷰트(Simple Attribute)

- 하나의 요소만으로 구성되는 애트리뷰트

예시) SSN, Salary

03 엔티티, 엔티티타입, 애트리뷰트, 키

2 애트리뷰트 타입

- 복합 애트리뷰트(Composite Attribute)
  - 복수 개의 요소들로 구성되는 애트리뷰트

예시) Address (Apt#, House#, Street, City, State, ZipCode, Country)
또는
Name (FirstName, MiddleName, LastName)

03 엔티티, 엔티티타입, 애트리뷰트, 키

2 애트리뷰트 타입

🔎 다치 애트리뷰트(Multi-valued Attribute)

- 여러 값을 가질 수 있는 애트리뷰트

예시) {Color} 로 표기되는 CAR 의 색 또는
{PreviousDegrees} 로 표기되는
STUDENT의 이전 학위 내역

03 엔티티, 엔티티타입, 애트리뷰트, 키

2 애트리뷰트 타입

- 일반적으로 복합 및 다치 애트리뷰트는 몇 단계로 내포될 수 있지만 그런 경우는 흔하지 않음

예시) STUDENT의 PreviousDegrees는 {PreviousDegrees (College, Year, Degree, Field)}로 표기되는 복합 다치 애트리뷰트일 수 있음

03 엔티티, 엔티티타입, 애트리뷰트, 키

3 엔티티타입과 키 애트리뷰트

엔티티타입

- 동일한 애트리뷰트들을 갖는 엔티티들의 집합으로 정의함

키 애트리뷰트

- 한 엔티티 타입에서 각 엔티티가 유일한 값을 가지는 애트리뷰트를 그 엔티티타입의 키 애트리뷰트라 함 (보통은 단일 키(Single Key) 형태)

예시)
: 사원의 사번, 학생의 학번, 선원의 선원번호

03 엔티티, 엔티티타입, 애트리뷰트, 키

3 엔티티타입과 키 애트리뷰트

- 복합 키(Composite Key) 형태

예시)
: 학생의 (이름, 학과), 예약의 (선원번호, 선박번호)

- **엔티티타입**은 한 개 이상의 키를 가질 수 있음

예시)
: CAR 엔티티타입은 다음과 같이 2개의 키를 가짐
- 차량 고유 번호(VehicleIdentificationNumber)
- 차량 넘버(VehicleTagNumber)

03 엔티티, 엔티티타입, 애트리뷰트, 키

4 요구사항으로부터 엔티티 식별

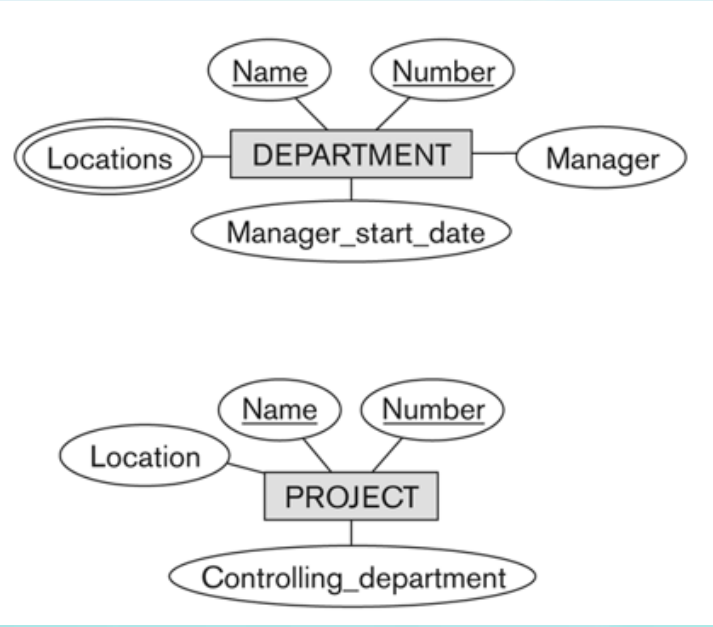
Company 데이터베이스의 요구사항을 토대로
초기 4개의 엔티티타입을 찾을 수 있음

- Department(부서)
- Project(프로젝트)
- Employee(사원)
- Dependent(피부양자)

03 엔티티, 엔티티타입, 애트리뷰트, 키

4 요구사항으로부터 엔티티 식별

그림에서 보여주는 속성 중 일부 속성은 관계로 재정의



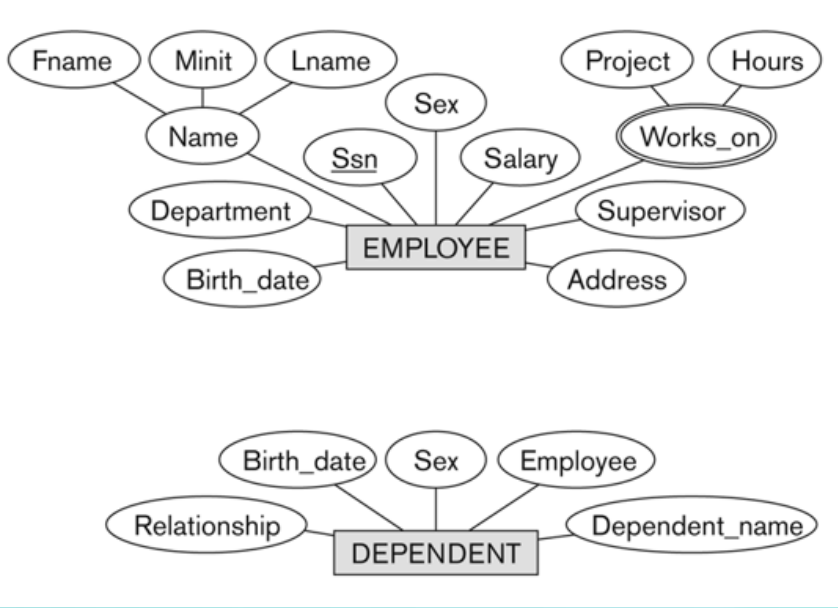
◀ Company DB 요구사항으로부터의 초기 엔티티타입과 속성들

※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

03 엔티티, 엔티티타입, 애트리뷰트, 키

4 요구사항으로부터 엔티티 식별

그림에서 보여주는 속성 중 일부 속성은 관계로 재정의



◀ Company DB 요구사항으로부터의 초기 엔티티타입과 속성들

※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

4 관계, 관계타입, 관계집합, 구조적 제약조건

04 관계, 관계타입, 관계집합, 구조적 제약조건

1 관계, 관계타입, 관계집합

관계(relationship)는 두 개 또는
그 이상의 엔티티들을 특정한 의미로 연관 짓는 것

예시) EMPLOYEE John Smith는 ProductX PROJECT에서 일하며, EMPLOYEE Franklin Wong은 Research DEPARTMENT를 관리한다

04 관계, 관계타입, 관계집합, 구조적 제약조건

1 관계, 관계타입, 관계집합

- 같은 형의 관계들은 **관계타입**으로 그룹화됨

예시)
- EMPLOYEE들과 PROJECT들이 참여하는 WORKS_ON 관계타입
- EMPLOYEE들과 DEPARTMENT들이 참여하는 WORKS_FOR 관계타입

- 데이터베이스에 표현되어 있은 관계 인스턴스의 집합 또는 관계타입의 현재 상태를 **관계집합**이라고 함

04 관계, 관계타입, 관계집합, 구조적 제약조건

1 관계, 관계타입, 관계집합

- **관계타입의 차수(Degree)**는 참여하는 엔티티타입의 개수임
  - WORKS_ON 과 WORKS_FOR 는 모두 이진 관계

- 두 엔티티타입들에 대해 한 개 이상의 관계타입이 있을 수 있음

예시) EMPLOYEE와 DEPARTMENT 간의 관계 WORKS_FOR 와 관계 MANAGES는 서로 다른 연관 의미를 가짐

04 관계, 관계타입, 관계집합, 구조적 제약조건

1 관계, 관계타입, 관계집합

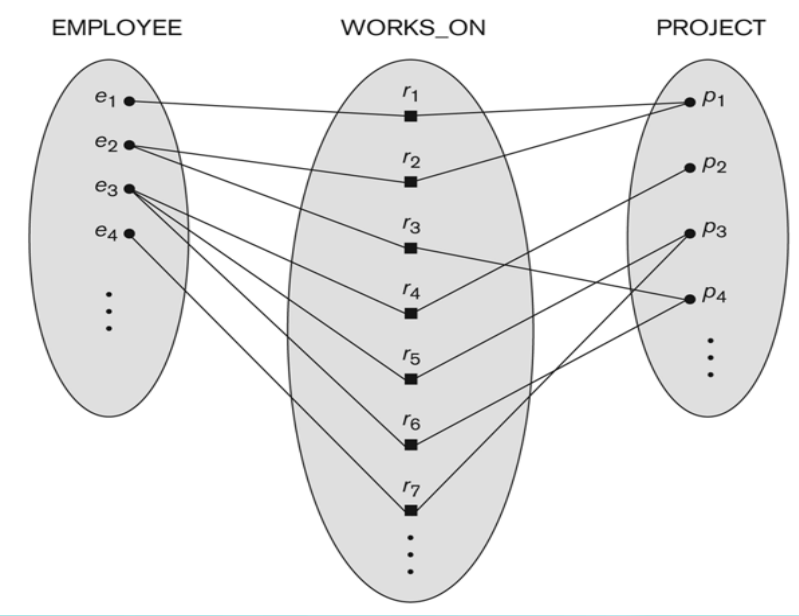
관계타입의 애트리뷰트

- 엔티티타입처럼 관계타입도 애트리뷰트들을 가질 수 있다
- 사원이 프로젝트에 근무하는 주당 시간을 기록하기 위한 WORKS_ON 관계타입에 Hours 애트리뷰트 추가
- 관리 사원이 부서를 관리하기 시작한 날짜를 기록하기 위한 MANAGES 관계타입에 Start_date 애트리뷰트 추가

04 관계, 관계타입, 관계집합, 구조적 제약조건

1 관계, 관계타입, 관계집합

EMPLOYEE와 PROJECT 간의 WORKS_ON 관계

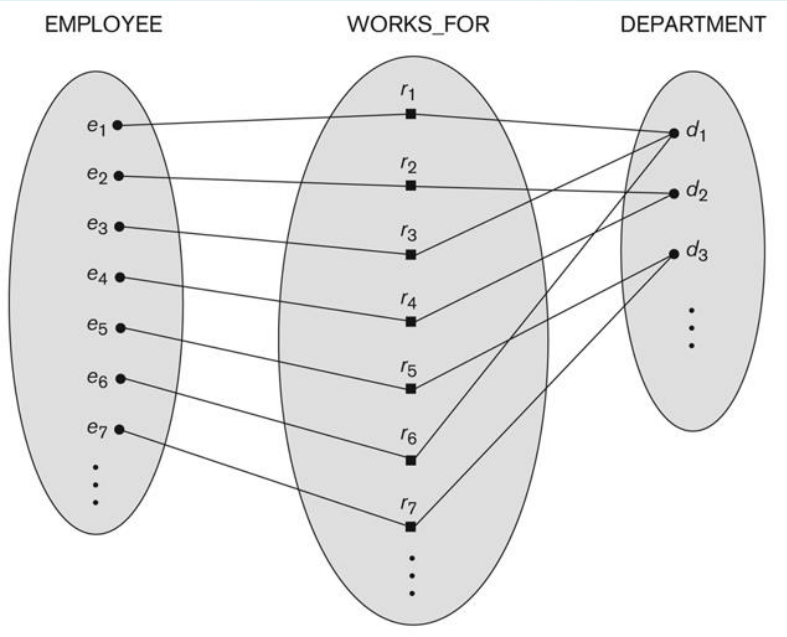


※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

04 관계, 관계타입, 관계집합, 구조적 제약조건

1 관계, 관계타입, 관계집합

EMPLOYEE와 PROJECT 간의 WORKS_FOR 관계



※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

04 관계, 관계타입, 관계집합, 구조적 제약조건

2 구조적 제약조건 (관계타입에서의 제약조건)

카디낼러티 제약조건(Cardinality Constraints)

- 엔티티가 참여할 수 있는 최대 관계 인스턴스들의 수를 명시함, WORKS_FOR에서 Department : Employee 는 1:N 인데 이는 한 부서가 다수의 사원들과 연관되고 한 사원은 하나의 부서와 연관됨을 의미
- 이진 관계에서 가능한 카디낼러티 비율은 1:1, 1:N, N:M
- MANAGES에서 Department : Employee 은 1:1
- WORKS_ON에서 Employee : Project 는 N:M

04 관계, 관계타입, 관계집합, 구조적 제약조건

2 구조적 제약조건 (관계타입에서의 제약조건)

참여 제약조건(Participation Constraints)

- 관계 타입에서 한 엔티티의 존재가 연관되어 있는 다른 엔티티에 의존하는지의 여부를 명시
- 각 엔티티가 참여할 수 있는 관계 인스턴스의 최소 수를 명시하므로 최소 카디낼러티 제약조건이라고도 함
- 전체 참여(Total Participation)와 부분 참여 (Partial Participation)가 있음

04 관계, 관계타입, 관계집합, 구조적 제약조건

2 구조적 제약조건 (관계타입에서의 제약조건)

참여 제약조건(Participation Constraints)

- 모든 사원이 반드시 특정한 부서에 소속되어 근무해야 한다면, 사원 엔티티는 그것이 적어도 하나의 WORKS_FOR 관계 인스턴스에 참여할 때만 존재할 수 있음
- 이때 EMPLOYEE가 WORKS_FOR에 참여하는 것은 전체참여
- 모든 사원이 반드시 한 부서를 관리하는 것이 아니므로 EMPLOYEE가 MANAGES 관계 타입에 참여하는 것은 부분참여