

1 관계 모델의 개념

01 관계 모델의 개념

1 관계형 모델(Relational Model)

- 🔍 관계(Relation)는 집합론에 기반한 수학적 개념
- 🔍 이 모델은 1970년에 IBM 연구소의 E.F. Codd 박사에 의해 처음 제안됨
- 🔍 비형식적으로(Informally), 릴레이션은 테이블로 간주될 수 있음

01 관계 모델의 개념

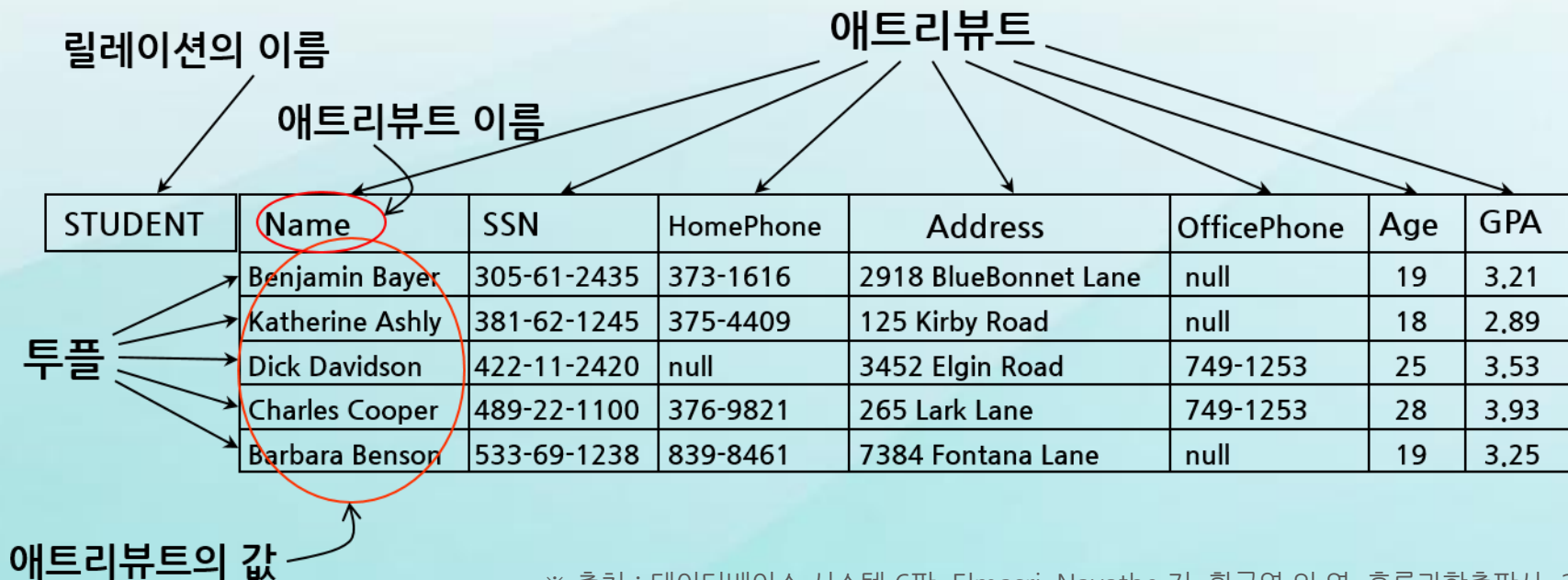
1 관계형 모델(Relational Model)

- 🔍 관계 모델에서 데이터베이스는 릴레이션(테이블)의 집합으로 표현됨
- 🔍 릴레이션은 튜플(행, 로우, 레코드, 인스턴스)들의 집합으로 표현됨
- 🔍 튜플은 애트리뷰트(열, 컬럼, 필드, 속성)들로 구성됨

01 관계 모델의 개념

1 관계형 모델(Relational Model)

[STUDENT 릴레이션]



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

01 관계 모델의 개념

2 릴레이션 관련 용어

릴레이션의 키(Key)

- 🔍 각 릴레이션은 포함하고 있는 튜플들을 유일하게 구별할 수 있도록 하는 속성을 가지고 있어야 함, 이 속성을 키 라고 함
- 🔍 예) STUDENT 테이블의 SSN 속성
- 🔍 때때로 릴레이션의 의미와는 아무 관계없는 행번호나 순차번호를 삽입하여 키로 사용하기도 함, 이런 키들을 인공키(Artificial key) 또는 대행키(Surrogate key) 라고 함

01 관계 모델의 개념

2 릴레이션 관련 용어

릴레이션 스키마(Relation schema)

- 릴레이션 이름 R 과 속성 A_i 들의 집합으로 표기
 $R(A_1, A_2, \dots, A_n)$
- 키 속성에는 밑줄을 그음
- 예) STUDENT(SSN, Name, BirthDate, Addr)

01 관계 모델의 개념

2 릴레이션 관련 용어

도메인(Domain)

- 🔍 속성이 가질 수 있는 값의 집합
- 🔍 Name 속성의 도메인 : 개인 이름들의 집합, 문자열
- 🔍 Age속성의 도메인 : 사원들 나이의 집합, 정수
- 🔍 실제 데이터 타입(int, float, char, varchar(10) 등)으로 명시

01 관계 모델의 개념

2 릴레이션 관련 용어

릴레이션 $R(A_1, A_2, \dots, A_n)$ 의 튜플

- 🔍 속성의 갯수에 맞춰 'n-튜플' 이라고 함
- 🔍 순서화된 값들의 집합 $t = \langle v_1, v_2, \dots, v_n \rangle$
- 🔍 값 v_i 는 $\text{dom}(A_i)$ 의 원소임

01 관계 모델의 개념

2 릴레이션 관련 용어

릴레이션 R의 인스턴스 $r(R)$

- 🔍 튜플들의 집합 $r(R) = \{t_1, t_2, \dots, t_m\}$
- 🔍 $r(R) \subseteq \text{dom}(A_1) \times \dots \times \text{dom}(A_n)$
- 🔍 $r(R)$ 은 실 세계의 특정 상태를 반영

01 관계 모델의 개념

2 릴레이션 관련 용어

릴레이션에서 튜플의 순서

- 🔍 릴레이션에서 튜플의 순서는 의미가 없음
- 🔍 집합에서 원소의 순서가 무의미한 것과 마찬가지로 임

01 관계 모델의 개념

2 릴레이션 관련 용어

튜플 내에서 값들의 순서

- 🔍 n-튜플은 n 개 값의 리스트이며,
튜플 내에서 값들의 순서는 지켜져야 함
- 🔍 그러나 각 속성과 값이 서로 대응될 수 있다면
값들의 순서는 중요하지 않을 수 있음
- 🔍 예를 들어, 하나의 튜플을 (속성, 값) 쌍들의
집합으로 표현하면 속성과 값은 서로 대응될 수
있으며 이때는 값의 순서가 중요하지 않음

01 관계 모델의 개념

2 릴레이션 관련 용어

튜플 내의 속성 값

- 🔍 더 이상 나눌 수 없는 원자 값들(Atomic Value)임
- 🔍 값을 알 수 없거나(Unknown) 해당되는 값이 없을 때 (Not applicable)에는 null 이라는 특수 값을 사용함
- 🔍 ER 모델에서의 다치 속성(Multi-valued Attribute)과 복합 속성(Composite Attribute)을 관계 모델에서는 허용되지 않음

01 관계 모델의 개념

3 관계 모델의 표기

- 차수가 n 인 릴레이션 스키마 R 은 $R(A_1, A_2, \dots, A_n)$ 으로 표기함
- 릴레이션 $r(R)$ 의 n -튜플 t 는 $t = \langle v_1, v_2, \dots, v_n \rangle$ 으로 표기함, 여기서 v_i 는 속성 A_i 의 값임
- $t[A_i]$ 또는 $t.A_i$ 는 t 에서 속성 A_i 의 값 v_i 를 가리킴
 - $t[A_i] = v_i$


01 관계 모델의 개념

3 관계 모델의 표기

- 대문자 Q, R, S 등은 릴레이션 이름을 나타냄
- 소문자 q, r, s 등은 릴레이션 상태를 나타냄
- 소문자 t, u, v 등은 튜플을 나타냄
- ‘STUDENT’ 처럼 릴레이션 스키마의 이름은 릴레이션의 현재 튜플들의 집합, 즉 현재의 릴레이션 상태를 가리킴
- 반면에 ‘STUDENT(Name, SSN, ...)’ 는 릴레이션 스키마를 가리킴

01 관계 모델의 개념


3 관계 모델의 표기

 용어 요약

| informal 용어 | formal 용어 |
|---|------------------------|
| 테이블 table | 릴레이션 relation |
| 컬럼 헤더 column header | 애트리뷰트, 속성 attribute |
| 모든 가능한 컬럼 값 all possible column values | 도메인 domain |

01 관계 모델의 개념

3 관계 모델의 표기

 용어 요약

| informal 용어 | formal 용어 |
|---------------------------------|---|
| 행 row | 튜플 tuple |
| 테이블 정의 table definition | 릴레이션 스키마 relation schema |
| 데이터가 입력된 테이블 populated table | 릴레이션 상태(/인스턴스) relation state(/instance) |

2 관계 모델의 제약조건

02 관계 모델의 제약조건

1 무결성 제약조건(IC ; Integrity Constraints)

- 🔍 모든 릴레이션 인스턴스들이 항상 만족해야 하는 조건
- 🔍 간단히 '제약조건' 이라고도 함
- 🔍 주요 제약조건
 - 도메인 제약조건(Domain constraints)
 - 키 제약조건(Key constraints)
 - 참조 제약조건(Referential constraints)

02 관계 모델의 제약조건

2 도메인 제약조건

🔍 각 튜플에서 속성 A 에 해당 되는 값은 반드시 A 의 도메인 $\text{dom}(A)$ 에 속하는 원자값이어야 함

🔍 도메인과 관련된 데이터 타입

- 정수, 실수와 같은 표준 숫자형
- 문자, 고정길이 문자열, 가변길이 문자열
- 날짜, 시간 등


02 관계 모델의 제약조건

3 키 제약조건

- 🔍 엔티티 무결성 제약 조건이라고도 함
- 🔍 키 속성에 해당되는 값은 유일해야 하며, null 값을 가질 수 없다는 제약 조건
 - 기본키가 각 튜플들을 식별하는 데에 이용되기 때문
- 🔍 참고로 R의 기본키에 속하지 않는 속성들도 null 값을 가질 수 없도록 제한할 수 있음
 - 릴레이션의 속성을 정의할 때 not null 로 명시

02 관계 모델의 제약조건

4 키의 종류

 유일성(Uniqueness)과 최소성(Minimality)을 갖는 키


- 후보키(CK; Candidate Key) 모든 가능한 키 집합
- 주키(PK; Primary Key)
 - : 설계자가 릴레이션 상의 엔티티들을 유일하게 구별하기 위해 후보 키 중에서 선택한 키, 기본적으로 인덱스를 구성할 때 사용됨
- 대체키/이차키(AK; Alternate Key)
 - : 주 키로 선택 받지 못한 키, 인덱스 구성에 사용될 수도 있음

02 관계 모델의 제약조건

4 키의 종류

 유일성만 가지는 키


- 수퍼키(Super Key)
: Key를 포함하는 수퍼셋(Super Set)

 유일성을 갖지 않으며 단지 참조용으로만 쓰이는 키

- 외래키(Foreign Key)
: 참조하는 릴레이션으로부터 빌려온 키

02 관계 모델의 제약조건

4 키의 종류


 예제)

- 모 자동차 회사에서 자사의 자동차에 대한 자사 직원들의 선호도를 조사하기 위해 관계 모델로 설계한 결과이다. 각 릴레이션의 후보키, 주키, 대체키, 외래키를 밝히시오.

| 직원 | | | 자동차 | | | | |
|-----|----|------|-----|------|----|-----|----|
| 직원명 | 부서 | 근무연수 | 번호판 | 모델 | 색상 | 직원명 | 부서 |
| 갑돌 | 총무 | 5 | 12 | 소라타 | 흰색 | 갑돌 | 자재 |
| 을순 | 총무 | 8 | 34 | 소라타 | 검정 | 을순 | 영업 |
| 갑돌 | 자재 | 11 | 56 | 익센 | 흰색 | 갑돌 | 자재 |
| 을순 | 영업 | 8 | 78 | 크레타스 | 검정 | 을순 | 총무 |

02 관계 모델의 제약조건

4 키의 종류

 예제) → 답

- 사원 Table
CK={사원명, 부서}, {부서, 근무연수}
PK={사원명, 부서}
FK 는 없다
- 자동차 Table
주의점 : 외래키 속성은 키 를 식별할 때 제외해야 함
CK={번호판}, {모델, 색상}
PK={번호판}
FK={사원명, 부서}




02 관계 모델의 제약조건

5 참조 제약조건

- 하나의 릴레이션 R에서 외래키 FK 의 값으로 다른 릴레이션 S의 기본키 PK 값을 참조하는 경우에 R과 S는 참조 무결성 제약 조건을 가진다고 함
- $t1[FK] = t2[PK]$ 이면 R의 튜플 t1이 S의 튜플 t2를 참조한다고 하며, FK를 외래키(Foreign key)라고 부름
- R을 참조한(Referencing) 릴레이션, S를 참조된(Referenced) 릴레이션이라고 부름

02 관계 모델의 제약조건

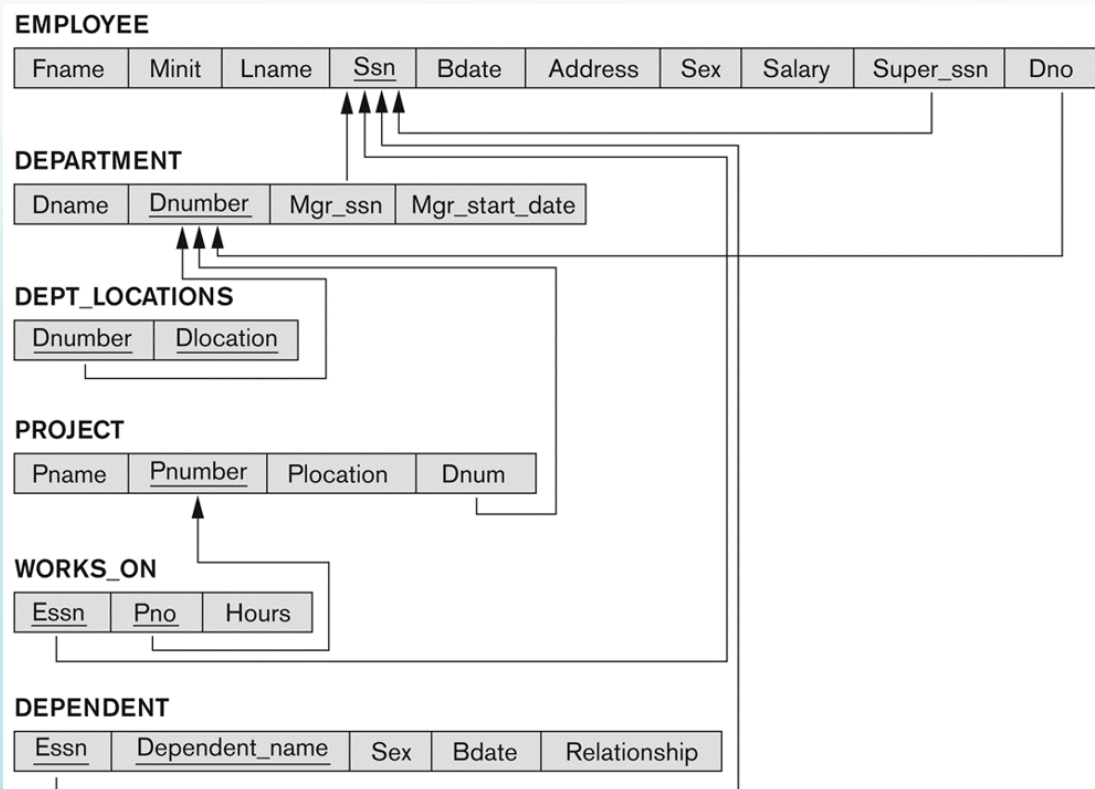
5 참조 제약조건

-  FK의 값은 참조하고 있는 릴레이션의 PK에 나타나는 값들만 가능함, 하지만 참조 관계가 없다면 FK가 null 값을 가질 수도 있음
-  앞서 소개된 키 제약조건과 도메인 제약조건은 하나의 릴레이션에 대한 제약조건이지만, 참조 무결성은 두 릴레이션에 대한 제약조건임을 유의해야 함
-  스키마 다이어그램에서 참조 무결성 제약조건은 R1.FK에서 R2.PK로의 화살표로 표시함

02 관계 모델의 제약조건

5 참조 제약조건

[COMPANY 데이터베이스 스키마에
표시된 참조 무결성 제약 조건들]



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

3 관계 데이터베이스 스키마

03 관계 데이터베이스 스키마

1 RDB 스키마와 RDB 인스턴스



릴레이션 스키마들의 집합 S 와
무결성 제약조건 IC 로 구성됨

- 릴레이션 스키마 집합 S 를 데이터베이스 이름이라 함
- $S = \{R_1, R_2, \dots, R_n\}$




RDB 인스턴스(/상태)

- 릴레이션 상태들의 집합

03 관계 데이터베이스 스키마

1 RDB 스키마와 RDB 인스턴스

 RDB 스키마의 예 Company = {Employee, ..., Dependent}

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

DEPARTMENT

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|-------|----------------|---------|----------------|
|-------|----------------|---------|----------------|

DEPT_LOCATIONS

| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|
|----------------|------------------|

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-------|----------------|-----------|------|
|-------|----------------|-----------|------|

WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

DEPENDENT

| <u>Essn</u> | <u>Dependent_name</u> | Sex | Bdate | Relationship |
|-------------|-----------------------|-----|-------|--------------|
|-------------|-----------------------|-----|-------|--------------|

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

03 관계 데이터베이스 스키마

1 RDB 스키마와 RDB 인스턴스

RDB 상태(/인스턴스)의 예

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

EMPLOYEE

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

DEPARTMENT

| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
|----------------|---------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

DEPT_LOCATIONS

| Dnumber | Dlocation |
|---------|-----------|
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

WORKS_ON

| Essn | Pno | Hours |
|-----------|-----|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

PROJECT

| Pname | Pnumber | Plocation | Dnum |
|-----------------|---------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

DEPENDENT

| Essn | Dependent_name | Sex | Bdate | Relationship |
|-----------|----------------|-----|------------|--------------|
| 333445555 | Alice | F | 1986-04-05 | Daughter |
| 333445555 | Theodore | M | 1983-10-25 | Son |
| 333445555 | Joy | F | 1958-05-03 | Spouse |
| 987654321 | Abner | M | 1942-02-28 | Spouse |
| 123456789 | Michael | M | 1988-01-04 | Son |
| 123456789 | Alice | F | 1988-12-30 | Daughter |
| 123456789 | Elizabeth | F | 1967-05-05 | Spouse |

03 관계 데이터베이스 스키마

2 Simple Company 데이터베이스 설계 예

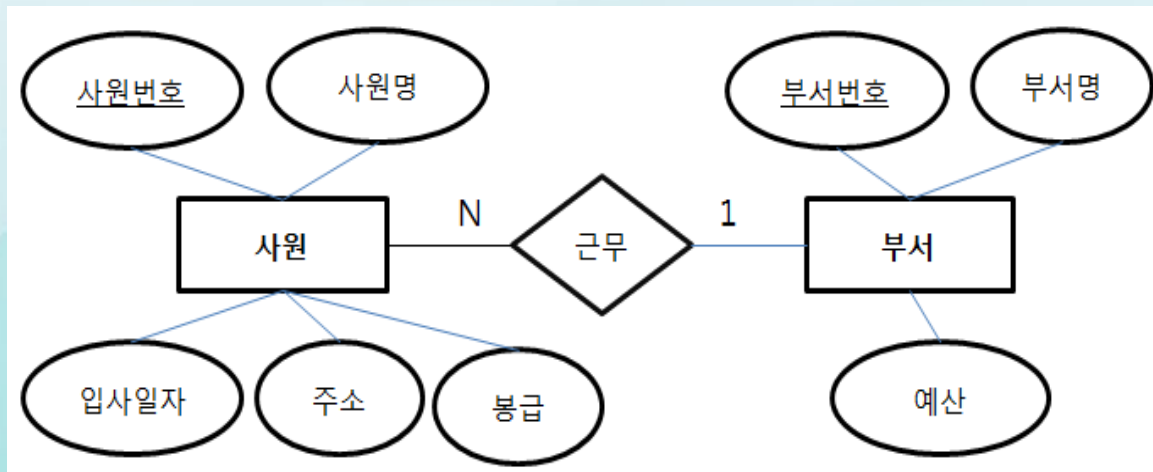
요구사항 명세서

- A라는 회사에서는 직원과 부서 정보를 관리하고자 한다. 직원 정보에는 직원번호, 직원명, 입사일자, 주소, 봉급이 포함되어야 하며 직원번호로 각 직원들을 유일하게 식별할 수 있다. 부서 정보에는 부서번호, 부서명, 예산이 포함되어야 하며 부서번호로 각 부서를 식별할 수 있다. 이 회사의 직원수는 약 3000 명이고, 모든 직원들은 특정한 하나의 부서에 속해 있어야 한다. 20개의 부서 중에는 '고객만족부서' 라는 직원이 존재하지 않는 부서가 있다. 이 부서는 고객만족을 실천하기 위한 회사의 의지를 소비자에게 광고하기 위해 만들어진 부서이다. 이 외의 나머지 모든 부서에는 여러 명의 직원들이 근무하고 있다.

03 관계 데이터베이스 스키마

2 Simple Company 데이터베이스 설계 예

🔍 개념적 설계(개체-관계 모델, ERD)



03 관계 데이터베이스 스키마

2 Simple Company 데이터베이스 설계 예

🔍 논리적 설계(관계 모델, Relations)

직원

| 사번 | 직원명 | 입사일자 | 주소 | 봉급 | 부서번호 |
|----|-----|------|----|-----|------|
| 1 | 갑 | 2010 | 서울 | 340 | 100 |
| 2 | 을 | 2013 | 광주 | 300 | 100 |
| 3 | 병 | 2018 | 부산 | 200 | null |

부서

| 부서번호 | 부서명 | 예산 |
|------|-----|------|
| 100 | 총무 | 300 |
| 200 | 자재 | 6000 |

- 직원(사번, 직원명, 입사일자, 주소, 봉급, 부서번호)
PK={사번}, FK={부서번호}
- 부서(부서번호, 부서명, 예산)
PK={부서번호}