

- Data Modeling -

- 요구사항분석
- 개념적모델링 : 객체관계모델(Entity Relational Model) - ERD
- 논리적모델링 : 관계형모델(Relational Data Model)
- 물리적모델링

개념적모델링 : 객체관계모델(Entity Relational Model) - ERD

Introduction to Database Systems

Data Modeling(**Conceptual**)

모든 서비스에 **DB**가 필요한가?

- 1. 입금, 출금, 이체 등 은행 거래
- 2. 호텔 객실의 예약
- 3. 신호등의 램프 제어
- 4. 온라인 쇼핑몰에서의 물품 구매
- 5. 전자식 개폐장치의 비밀번호 관리

DB 시스템의 특성

1. 최초 적재>Loading) -> 이벤트 발생에 따른 잦은 변경Interaction)
2. **대용량**의 데이터를 다룸
 - 사용자들이 원하는 순간 데이터를 접근하기 위해서는?
 - > **대용량**의 데이터가 체계적으로 조직화되어 있어야 함

Database System

- Database

- 데이터 및 데이터간 **관계의 집합**

- DBMS(Database Management Systems)

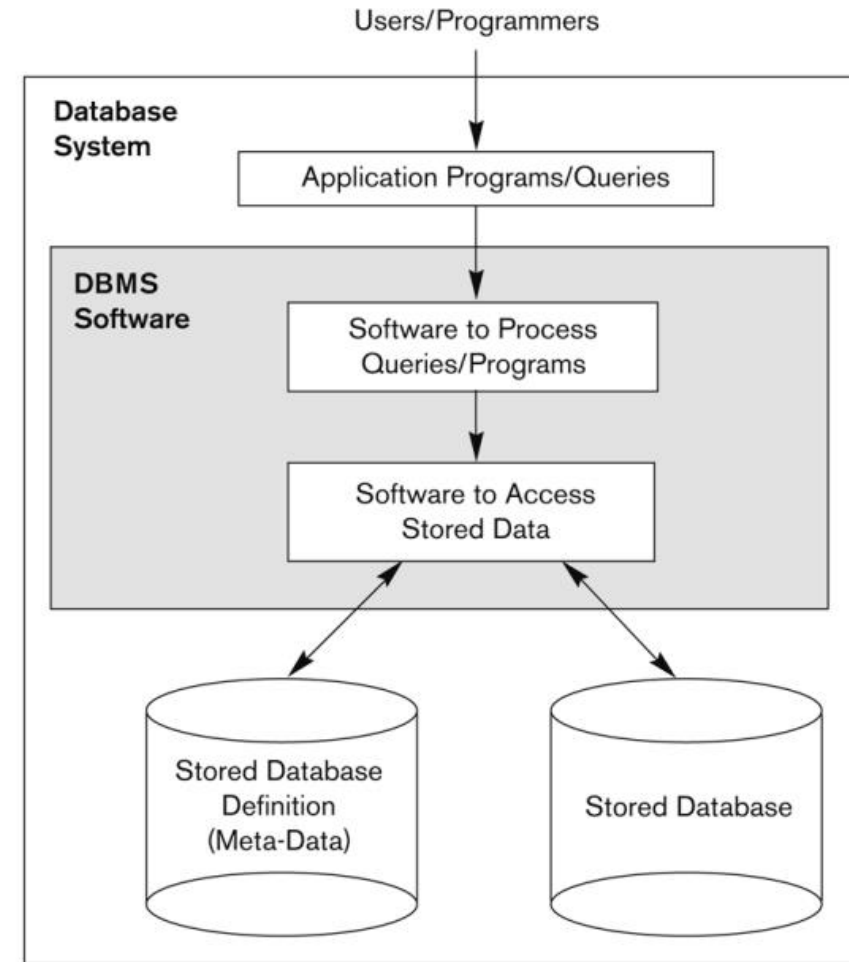
- 사용자가 Database에 접근할 수 있도록 **지원**해 주는

- * **프로그램의 집합**

- *Oracle

- *MS-SQL, MS-Access

- *MySQL



Example of a Database

course

course_number	course_name	credit_hours	department
1	수학	3	수학
2	영어 작문	4	영어
3	컴퓨터 과학	3	컴퓨터
4	역사	3	사회과학
5	물리학	4	물리학

section

section_idenfier	course_number	semester	year	instructor
1	1	가을	2022	스미스
2	2	봄	2023	존슨
3	3	가을	2022	데이비스
4	4	봄	2023	아담스
5	5	가을	2022	월슨

Example Query

- "물리학" Course의 Section을 하나라도 수강한 학생을 찾으시오.
- "Section "5", Student "5"의 Grade(성적)을 "B"로 수정하시오.

student

student_number	name
1	존 스미스
2	에밀리 존슨
3	마이클 데이비스
4	사라 아담스
5	데이비드 월슨

grade_report

student_number	section_idenfier	grade
1	1	A
1	2	B
2	3	B+
3	1	A-
3	4	C
4	2	A
5	3	B
5	5	B-

Schemas versus Instances

- 데이터베이스 스키마 (Schema)

- 데이터베이스 구조, 데이터 타입, 그리고 제약조건에 대한 명세
- 데이터베이스 설계 단계에서 명시되며, 자주 변경되지 않음

student_number	name
----------------	------

student(학생)

course_number	course_name	credit_hours	department
---------------	-------------	--------------	------------

course(과목)

section_identifier	course_number	semester	year	instructor
--------------------	---------------	----------	------	------------

section(분반)

student_number	section_identifier	grade
----------------	--------------------	-------

grade_report(성적리포트)

Schemas versus Instances

- 데이터베이스 인스턴스 (Instance)

- 특정 시점에 데이터베이스에 실제로 저장되어 있는 데이터의 값
= Database instance = Occurrence = Snapshot

section

section_identifier	course_number	semester	year	instructor
1	1	가을	2022	스미스
2	2	봄	2023	존슨
3	3	가을	2022	데이비스
4	4	봄	2023	아담스
5	5	가을	2022	월슨

SQL (Structured Query Language)

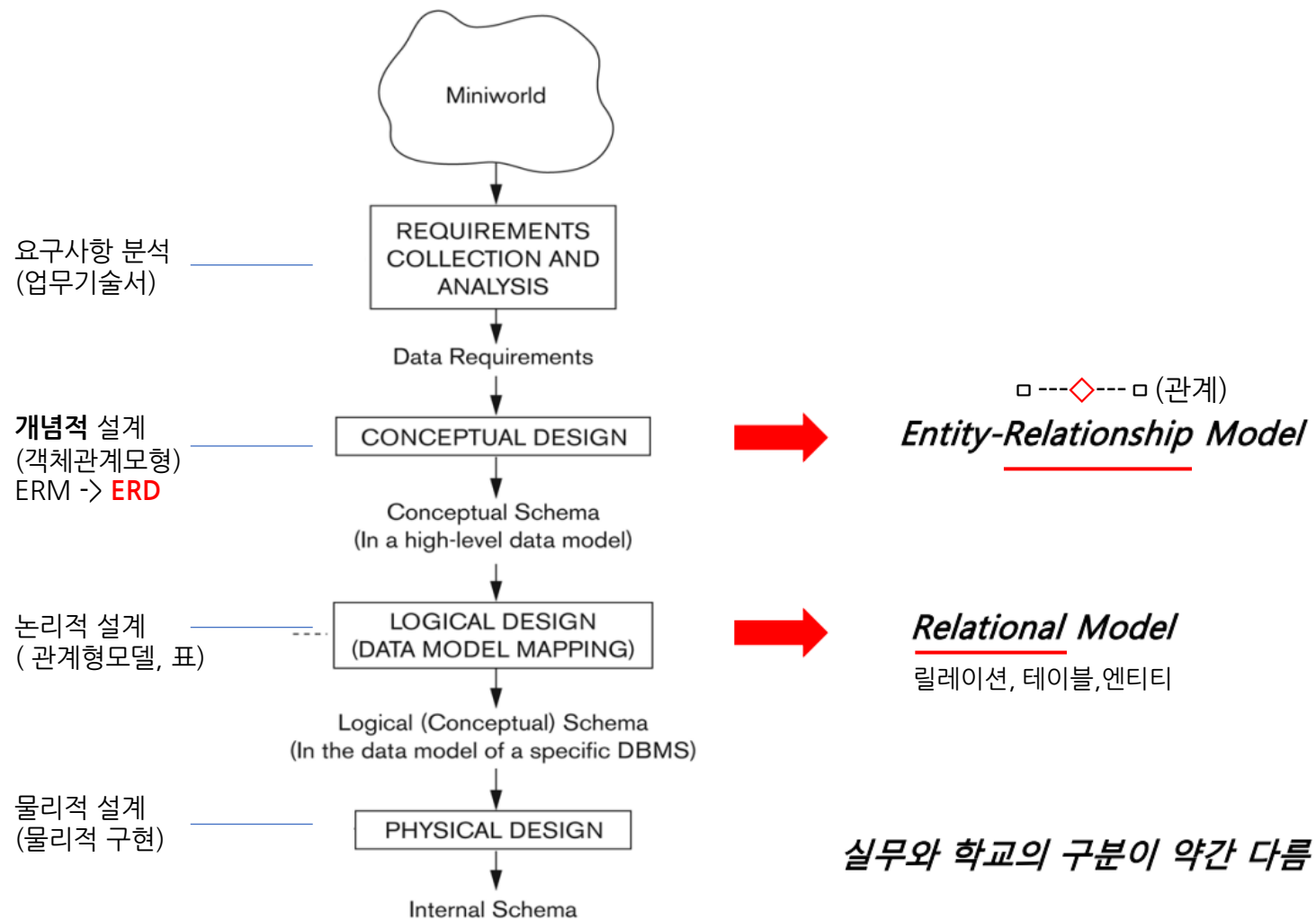
- **DML / DDL**

- 데이터 정의어 (**DDL**: Data Definition Language)
 - 스키마를 기술하기 위해 사용되며, 주로 DB 설계자가 사용함
- 데이터 조작어 (**DML**: Data Manipulation Language)
 - 데이터의 **조회(Retrieval)**, **삽입(Insertion)**, **삭제(Deletion)**, **갱신(Update)**에 사용됨
- cf) **DCL**(Data Control Language), **TCL**(Transaction Control Language)

- **독립 실행형 / 내장형**

- 독립 실행형
 - SQL 인터페이스를 이용하여 SQL 쿼리를 직접 DBMS에 입력
- 내장형
 - C, C++, Java 등의 프로그래밍 언어에 내장됨
 - Host language + Data sublanguage 로 구성됨

Overview of Database Design Process



ER Model Concept

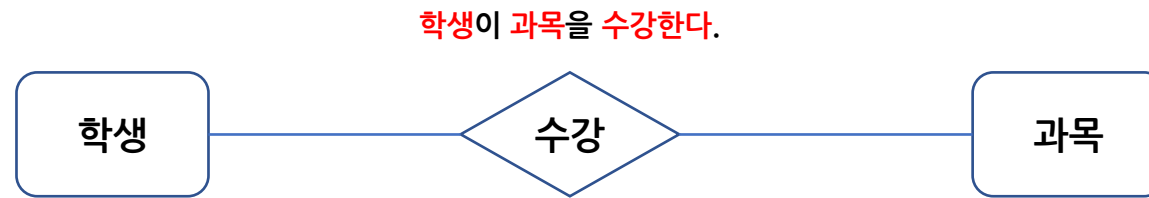
- 개체 (Entity)

- 실세계에 존재하는 의미 있는 하나의 정보 단위
- 물리적 객체 뿐 아니라 개념적 객체도 포함

(학생, 자동차, 강의실, ...) / (프로젝트, 직업, 교과목, ...)

- 관계 (Relationship)

- 개체들 사이의 **연관성**

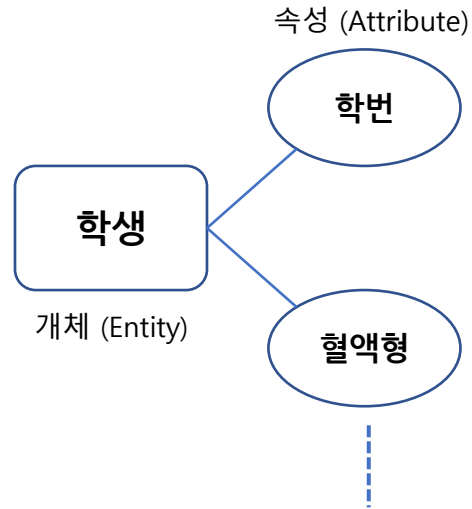


[학생]과 [과목] 사이의 [수강] 관계

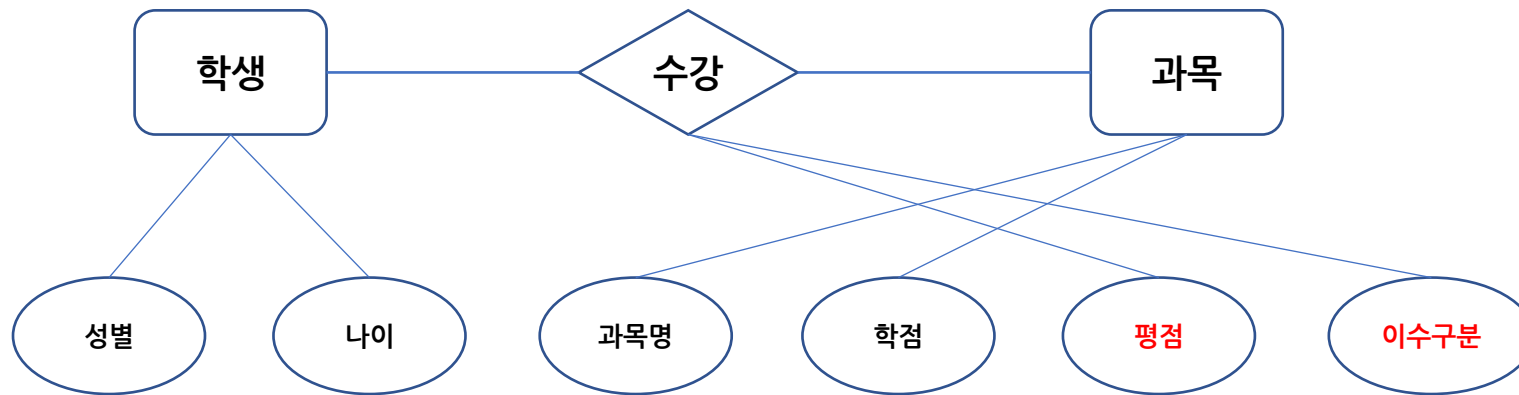
ER Model Concepts

- 속성 (Attribute)

- 개체 또는 관계의 본질적 성질



- 다음 개체 및 관계에서 주어진 속성의 주인 (Owner)는?



Types of Attributes

- **Single-valued** vs. Multivalued

- 나이 vs. 취미

- **Simple** vs. Composite

- **Simple** Attribute - 더 이상 쪼개지지 않는 원자 값을 갖는 속성

- 나이, 학번, ...

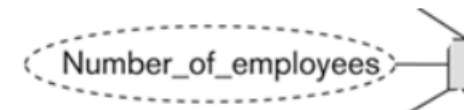
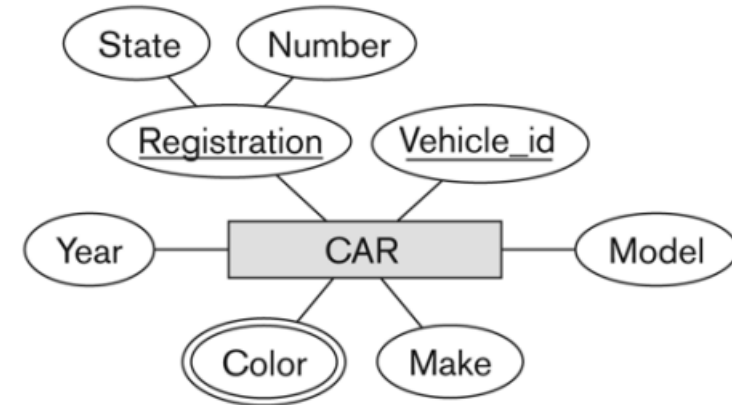
- **Composite** Attribute - 몇 개의 요소로 분해될 수 있는 속성

- 주소 -> 시, 군, 구, 번지, ...

- **Stored** vs. Derived

- **Derived** Attribute - 저장된 다른 데이터로부터 유도 가능한 속성

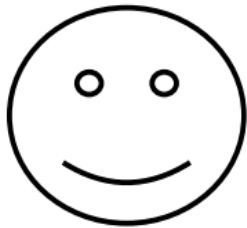
- 각 과목의 성적 -> **총점**, 주민등록번호 -> **나이**, ...



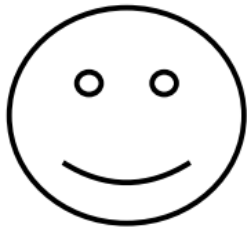
Entity Types and Key Attributes

- 키 속성 (Key Attributes)

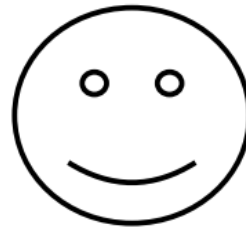
- 어떤 개체에 대해서 **항상 유일한 값**을 갖는 속성(또는 속성들의 집합)
 - 학생의 학번, 책의 ISBN, 자동차의 차량번호, ...
- 특정 Snapshot이 아닌, **해당 개체의 모든** 가능한 Snapshot의 집합을 고려하여 파악되어야 함
 - 다음의 **SSN**, 이름, 혈액형 중 키 속성을 찾으시오.



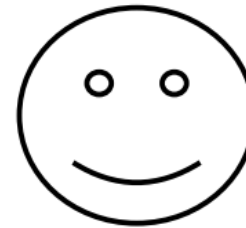
SSN: 740723-1111111
이름: 홍길동
혈액형 : A



SSN: 820424-2222222
이름: 유관순
혈액형 : A



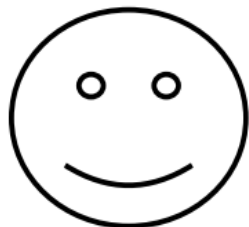
SSN: 100201-3333333
이름: 강감찬
혈액형 : A



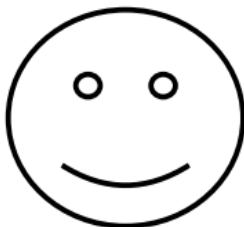
Entity Types and Key Attributes

- 키 속성 (Key Attributes)

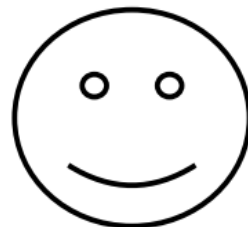
- 다음에서 키 속성을 찾으시오.



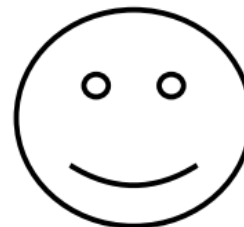
팀명: ManU
선수명: Park
등번호: 13
포지션: FW



팀명: Chelsea
선수명: Cech
등번호: 1
포지션: GK



팀명: Arsenal
선수명: Park
등번호: 9
포지션: SS

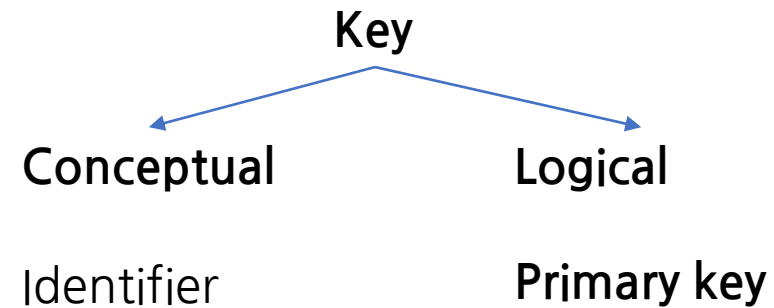


팀명: ManU
선수명:
Berbatov
등번호: 9
포지션: FW

Entity Types and Key Attributes

● 키 속성 (Key Attributes)

- 복합 키 (Composite Key)
 - **Composite Attribute**가 키 속성이 되는 경우
 - 복합 키는 **최소성**을 가져야 함
 - (팀명, 등번호) vs. (팀명, 등번호, ~~선수명~~)
- 각 개체는 **하나 이상의 키**를 가질 수 있음
- 어떤 개체는 키를 갖지 않을 수도 있음
 - 약성 개체 (Weak Entity)



Example COMPANY Database

● 요구사항 분석

- 회사는 여러 개의 **부서**로 구성되어 있다.
- 각 **부서**는 부서명, 부서번호, 부서 내 직원 수, 그리고 그 **부서**를 관리하는 **직원**(관리자)의 정보를 갖고 있다.
- **직원**은 최대 하나의 부서를 관리할 수 있다. 한편 해당 직원이 부서 관리자로 근무를 시작한 정보도 저장해야 한다.
- 각 부서는 여러 지역에 위치하고 있을 수 있으며, 각 부서는 여러 **프로젝트**를 동시에 관리할 수 있다. 각 프로젝트는 고유의 프로젝트명, 프로젝트번호를 가지며, 하나의 프로젝트는 하나의 지역에서만 진행된다.
- 직원에 대해 직원명, 주민번호, 주소(시/도, 상세주소), 급여, 성별, 생년월일의 정보를 저장한다.
- 각 직원은 하나의 부서에 소속되어 있으며, 여러 프로젝트에 참여할 수 있다.
- 한편 각 직원이 참여한 프로젝트에 대해 해당 직원의 주당근무시간 정보도 관리해야 한다.
- 각 직원에 대해 해당 직원의 직속상사(감독자)에 대한 정보를 관리한다.
- 각 직원은 여러 명의 **부양가족**을 가질 수 있다. 부양가족에 각각에 대해 부양가족명, 성별, 생년월일, 그리고 해당 직원과의 관계에 대한 정보를 관리한다.

Initial Schema for COMPANY Database

● 요구사항 정리

- Entity 부서:

- 부서명, 부서번호, 관리자, 관리시작일, 직원수, 지역

- Entity 프로젝트:

- 프로젝트명, 프로젝트번호, 지역, 부서

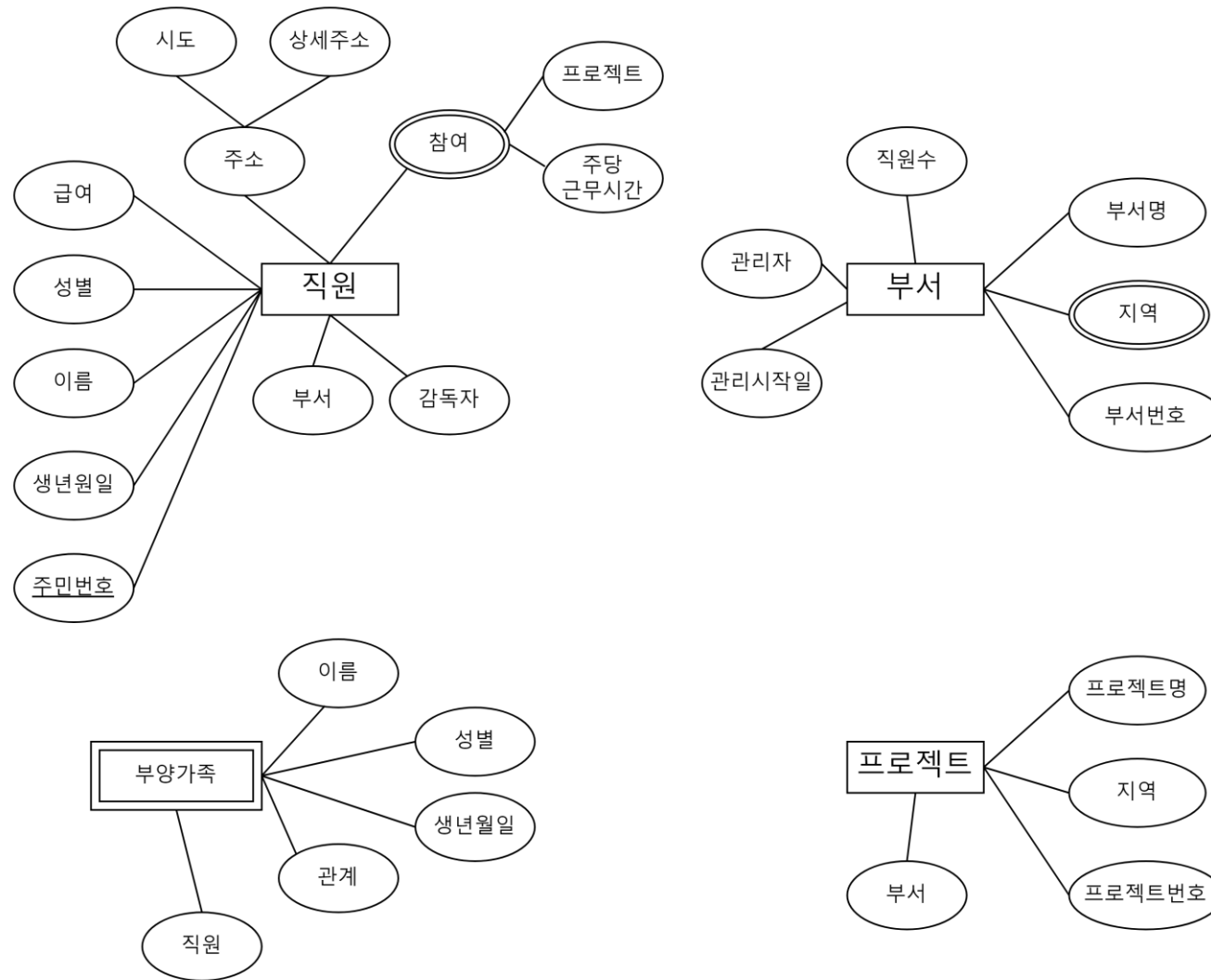
- Entity 직원:

- 이름, 주민번호, 성별, 주소, 급여, 생년월일, 부서, 감독자, (프로젝트, 주당근무시간)

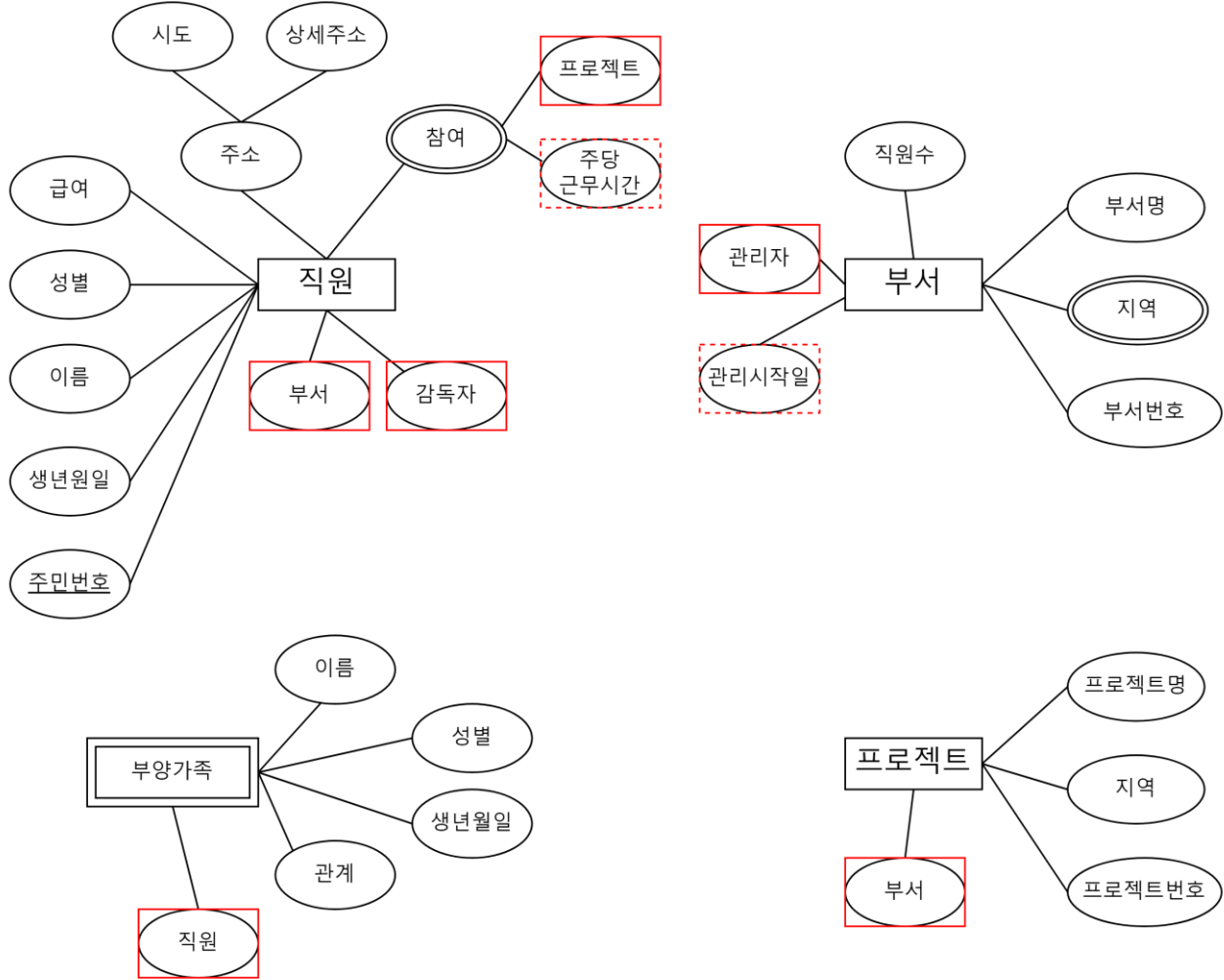
- Entity 부양가족:

- 직원, 이름, 성별, 생년월일, 관계

Initial Schema for COMPANY Database



Initial Schema for COMPANY Database

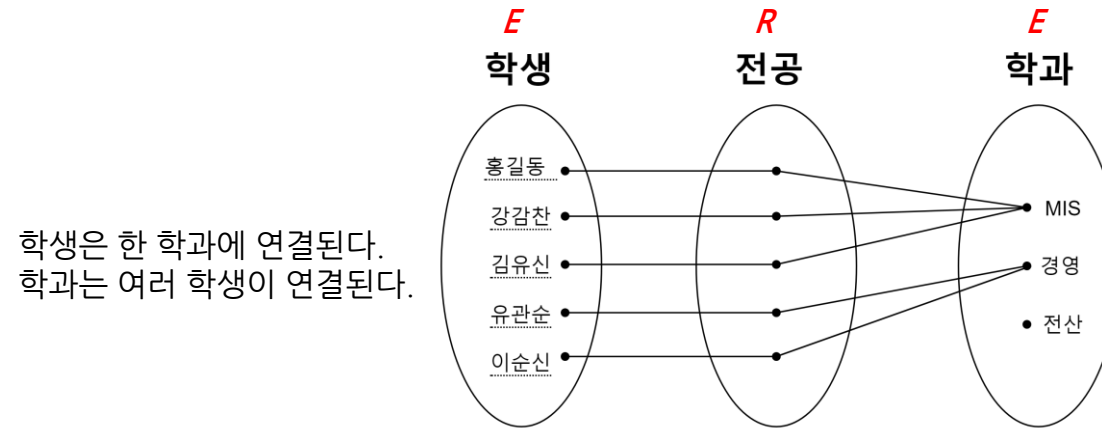


Relationships

- 관계(Relationship) 설정
 - 한 개체의 속성이 다른 개체를 참조할 때 관계가 형성됨
 - [직원]의 [프로젝트] 속성이 [프로젝트] 개체를 참조함
 - [부서]의 [관리자] 속성은 [직원] 개체를 참조함
 - ...
- 관계의 차수(Degree) *차수(Degree or Cardinality)
 - 관계에 참여하는 개체(entity)의 수
 - Binary(2), Ternary(3), Unary(1), ... N-ary
- 관계의 대응 수(Cardinality)
 - 해당 개체가 해당 관계에서 참여할 수 있는 관계 인스턴스의 최대 수
 - 1:1, 1:N (or N:1), M:N

Relationship

- 대응 수에 따른 관계의 분류



*하나의 학생은 몇 개의 학과에 연결되는가?

*하나의 학과는 몇 명의 학생과 연결되는가?

– 일대다 (1:N) 관계



– 일대일 (1:1) 관계

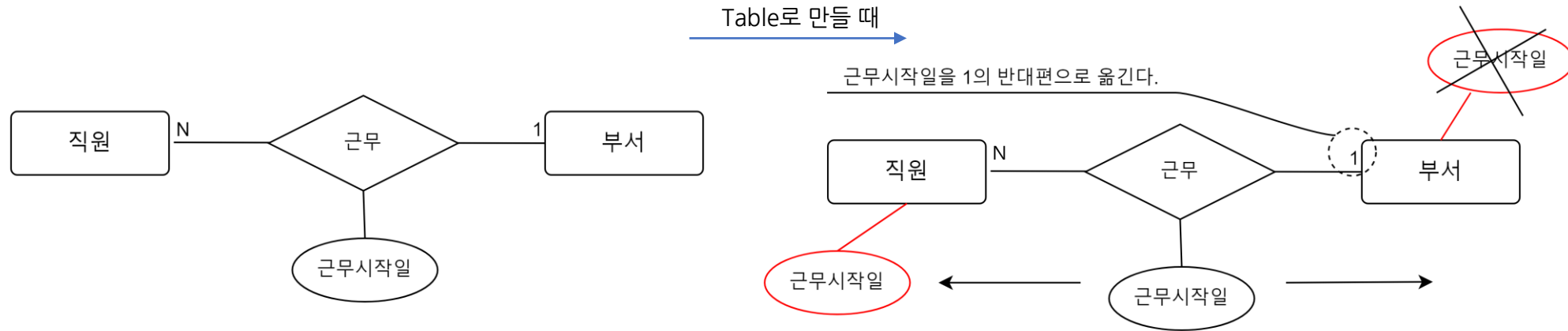


– 다대다 (M:N) 관계



Relationship

- 대응 수에 따른 관계의 분류

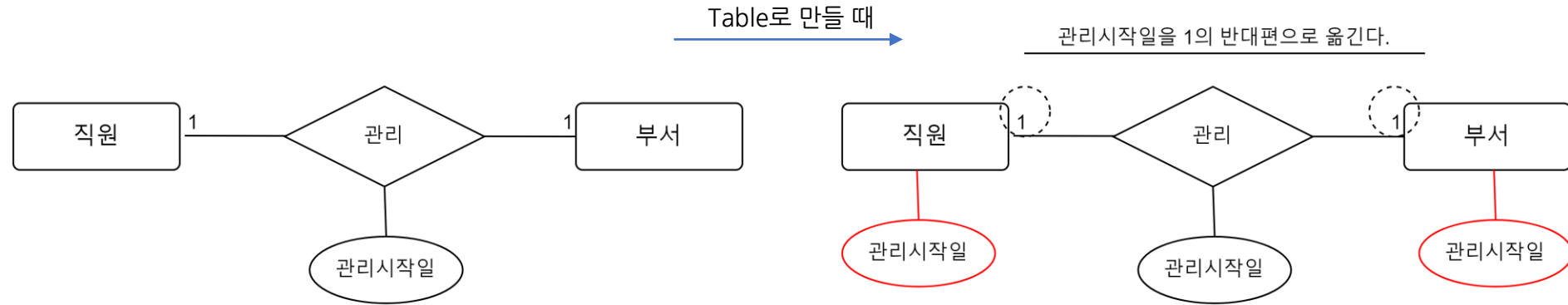


- 일대다 (1:N) 관계

- 위의 [근무] 관계의 [근무시작일] 속성이 이동한다면...
 - [근무시작일]은 어떤 개체에 속할 수 있는가?

Relationship

- 대응 수에 따른 관계의 분류

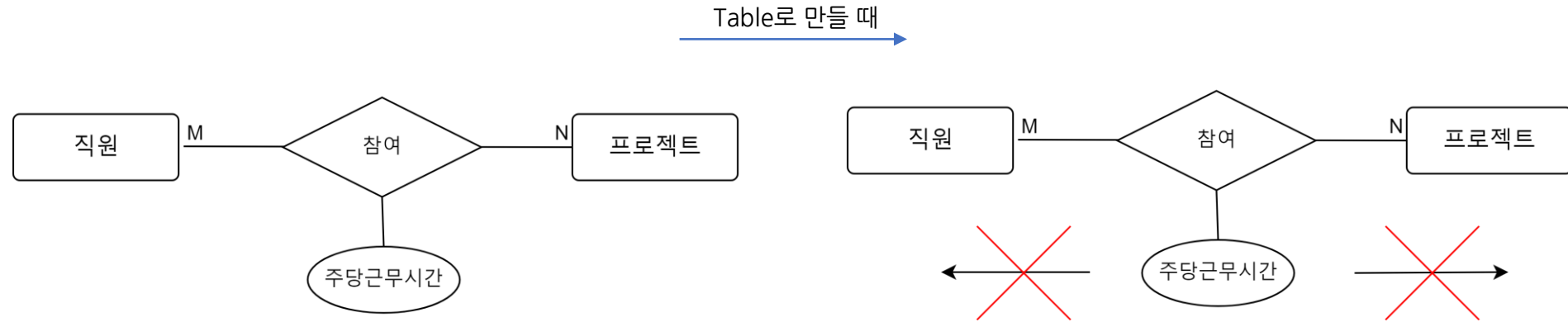


- 일대다 (1:1) 관계

- 위의 [관리] 관계의 [관리시작일] 속성이 이동한다면...
 - [관리시작일]은 어떤 개체에 속할 수 있는가?

Relationship

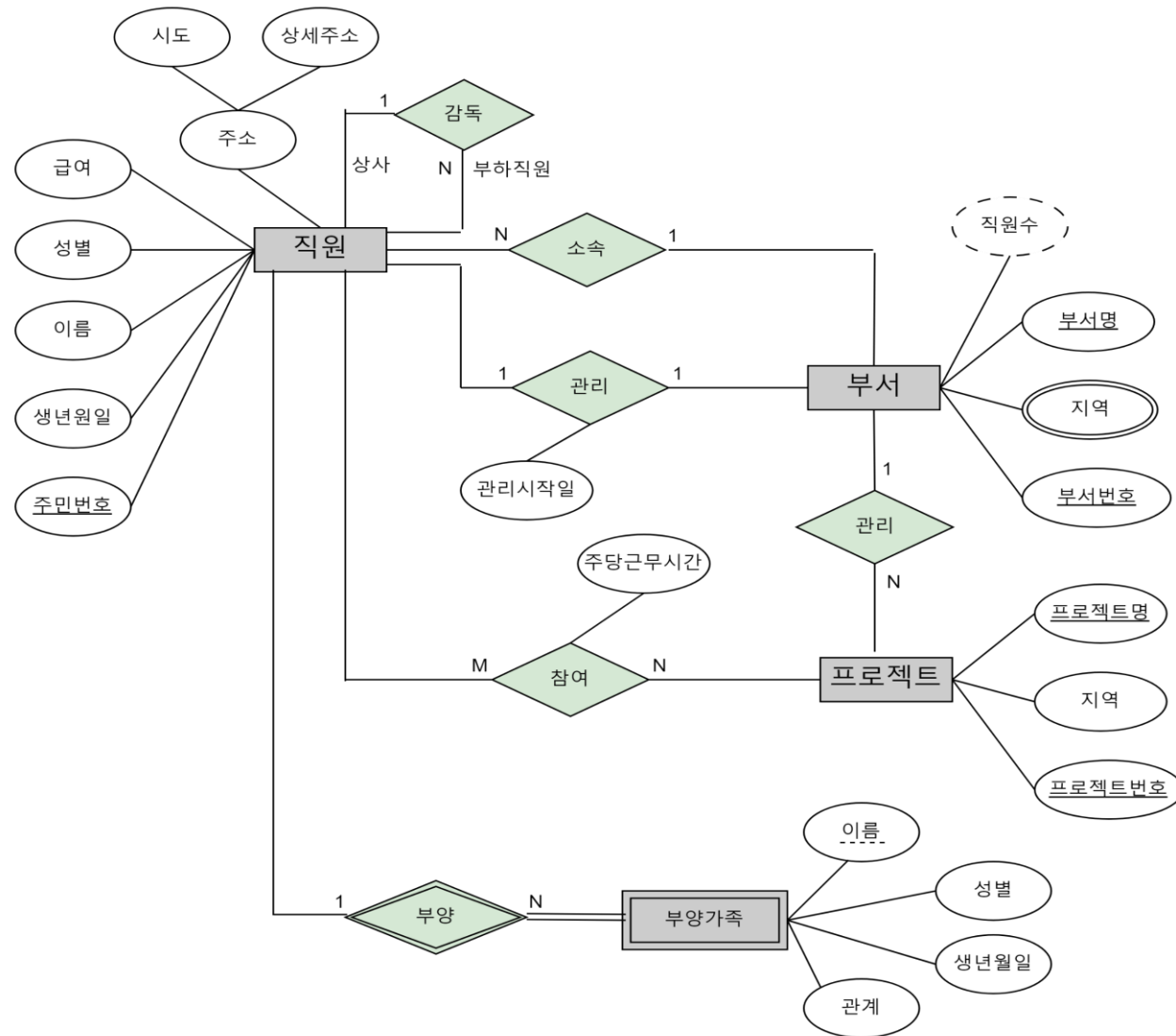
- 대응 수에 따른 관계의 분류



- 다대다 (M:N) 관계

- 위의 [참여] 관계의 [주당근무시간] 속성이 이동한다면...
 - [주당근무시간]은 어떤 개체에 속할 수 있는가?

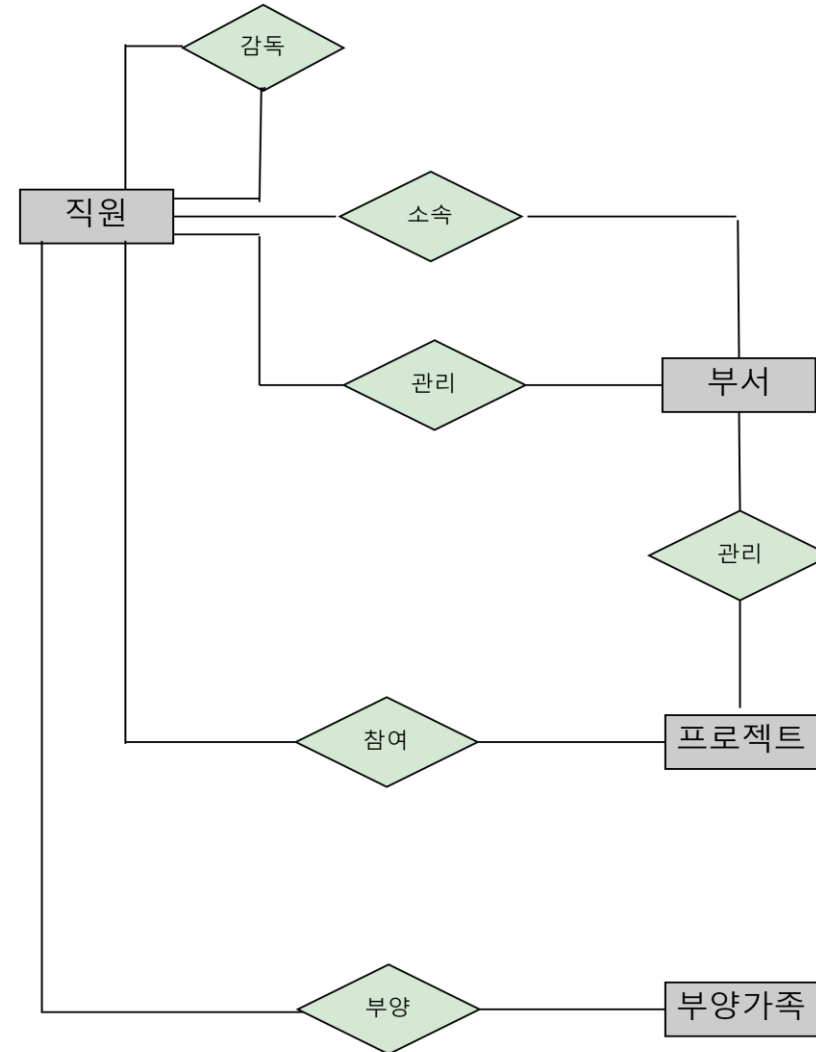
Refined Schema for COMPANY Database



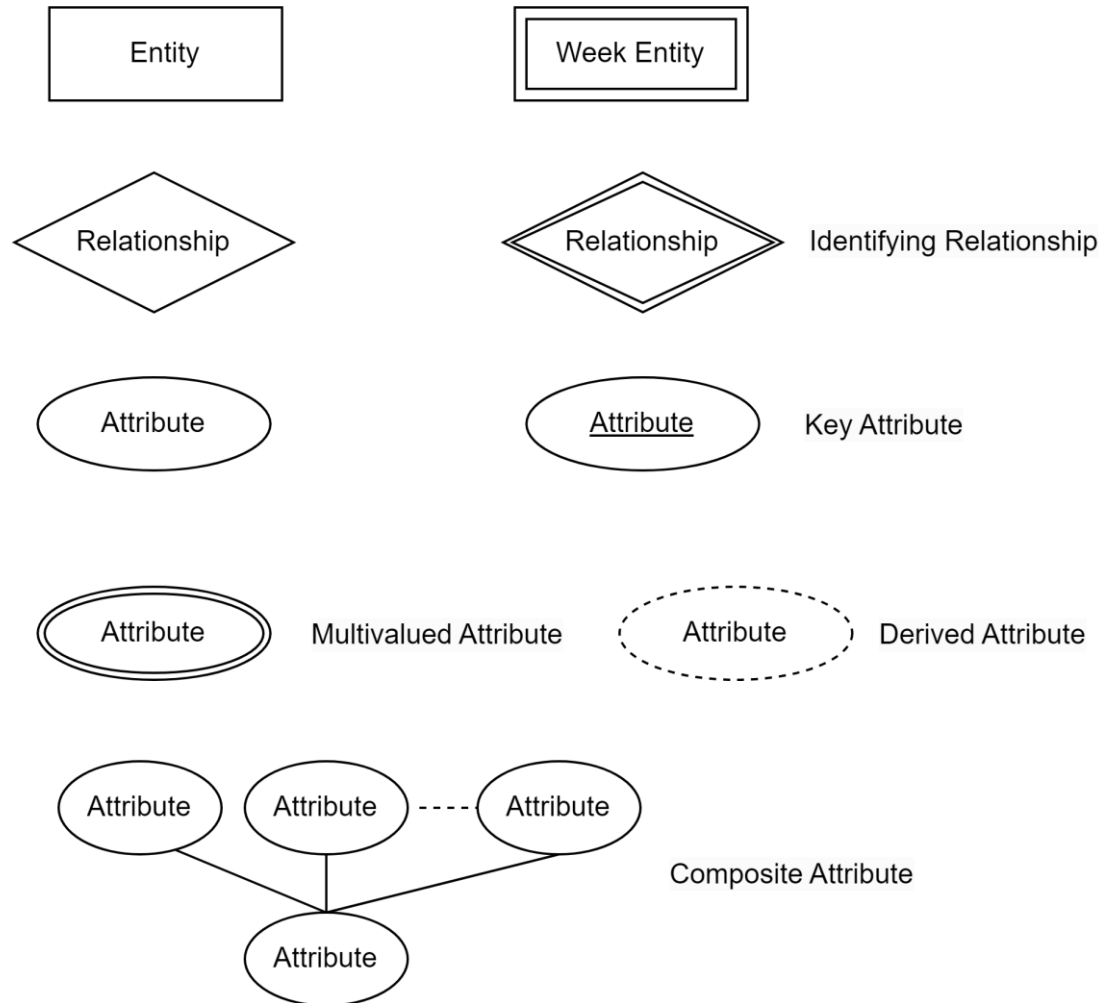
Practical Approach for ER Modeling

- 주요 **개체** 도출
 - 직원, 부서, 프로젝트, 부양가족
- 개체 간 **관계** 도출
 - 관계 도출
 - 대응 수 도출
- 개체 및 관계의 **속성** 도출
 - **키** 속성 도출
 - 약성 개체 확인
 - **일반** 속성 도출

90% 완성



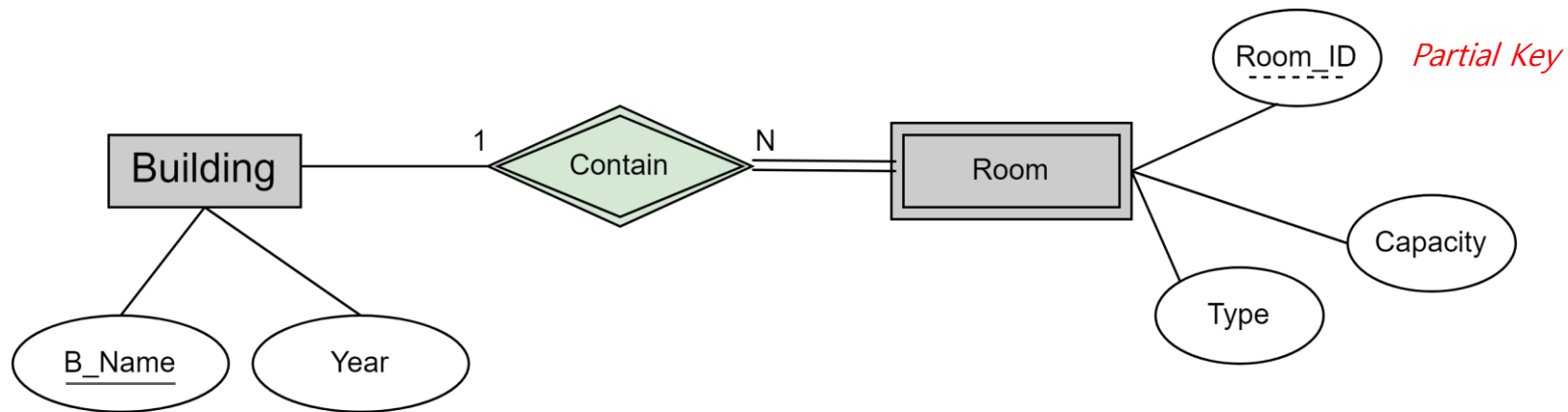
Notation for ER diagrams



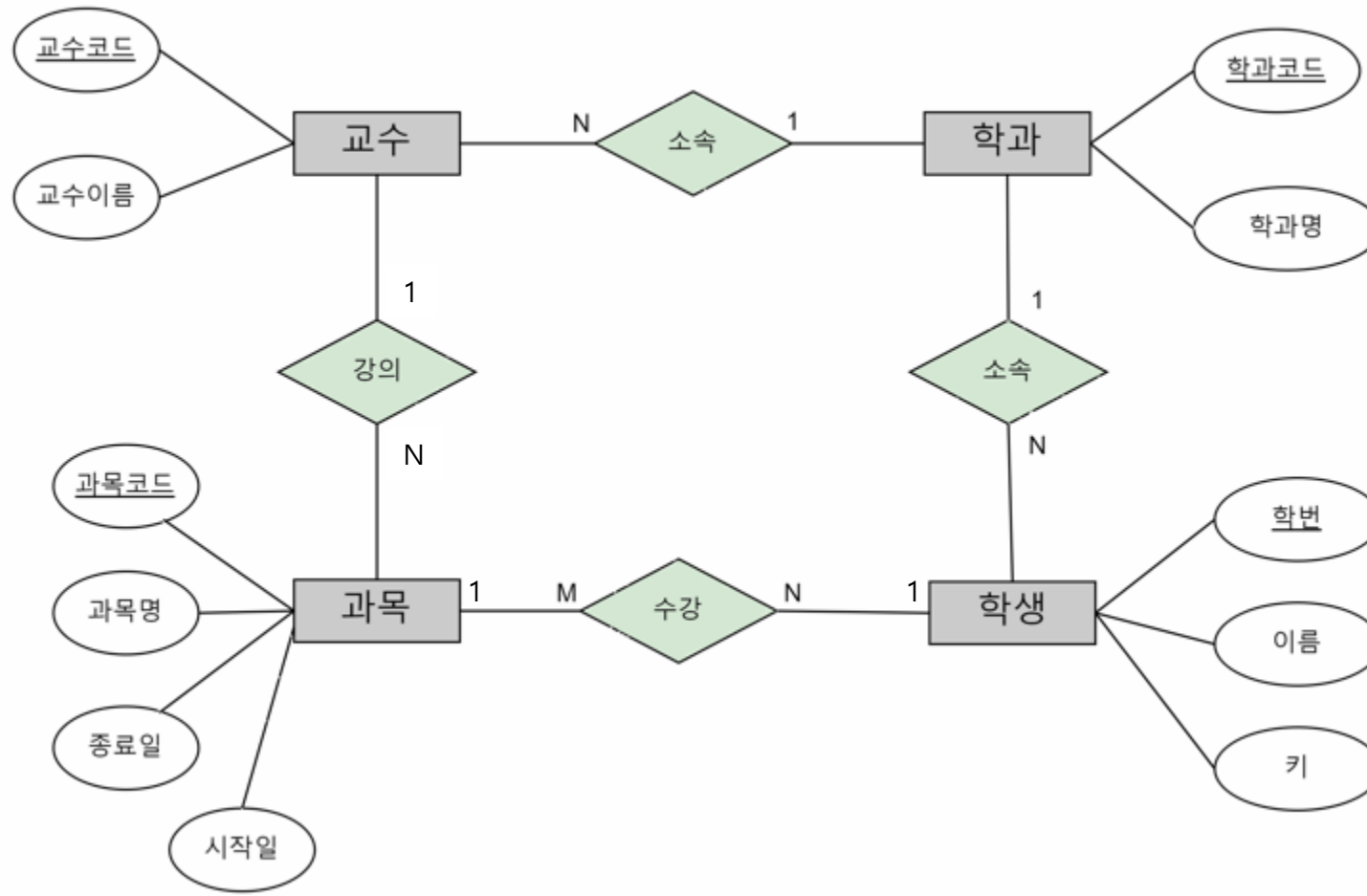
Weak Entity Types

- 약성 개체 (Weak Entity) (\leftrightarrow Strong Entity)

- 키 속성을 갖고 있지 않은 개체 \rightarrow 부분키 (Partial Key)만을 가짐
- 약성 개체는 개체를 식별할 수 있는 다른 개체와 식별 관계 (Identifying Relationship)으로 맺어져야 함
- 약성 개체의 식별자는 다음 속성의 조합으로 구성됨
 - 약성 개체의 부분키 속성+ 식별 개체(Identifying Entity)의 키 속성
 - ex) B_Name + Room_ID \rightarrow 경영관 413호



학사관리시스템 ERD



END