



Database Systems

Database 설계 실습 1



인하대학교

Database design

- **Database design**

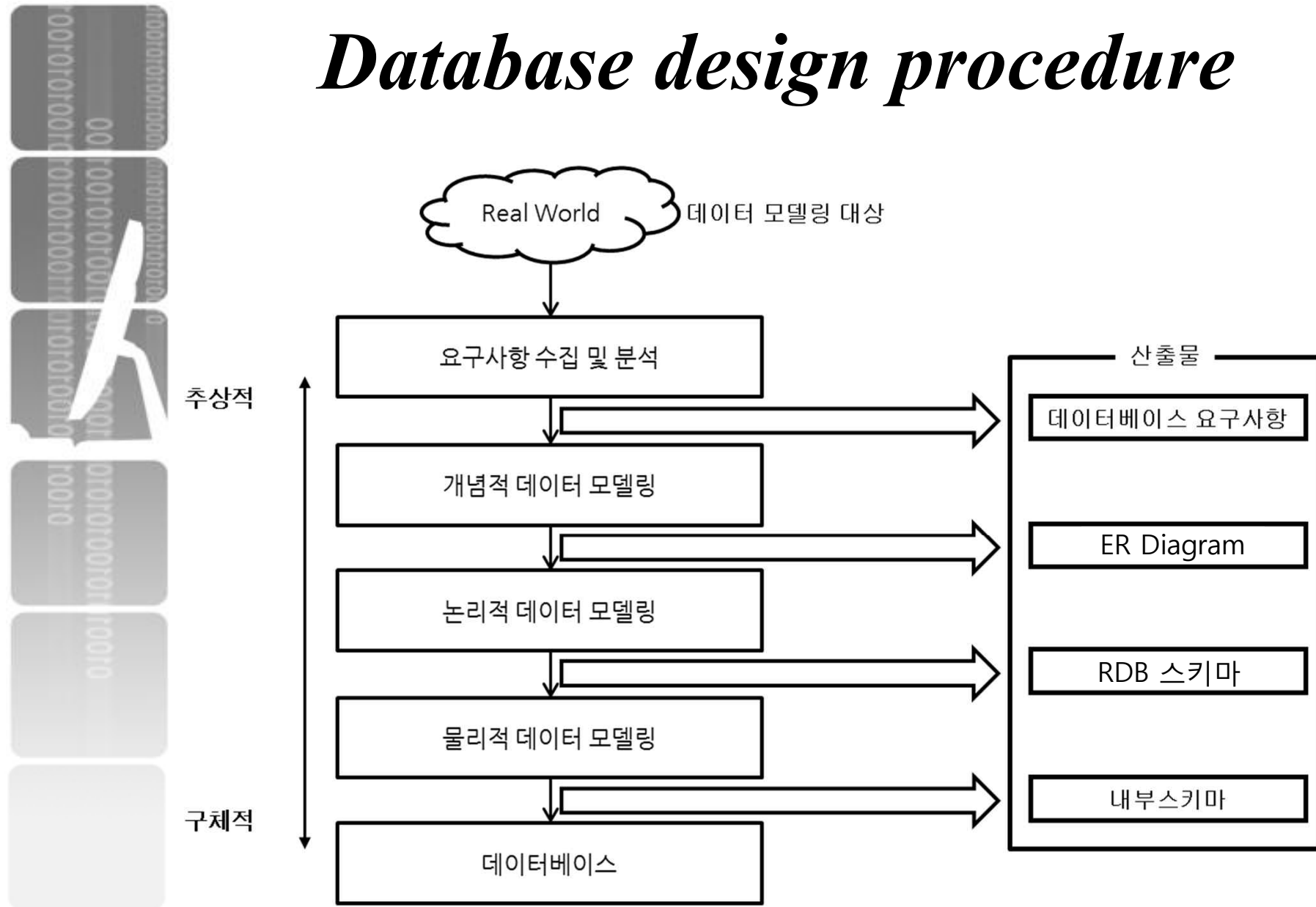
- 현실 세계의 모형을 약속된 표기법에 따라 표현하고 그 결과물을 목적 DataBase Management System (DBMS)의 타입에 맞는 database scheme으로 컴퓨터 세계에 표현하는 것

- 다양한 design methodology들이 있음

Database design goal

- 목적 : Database의 개념적, 논리적, 물리적 구조들을 디자인
 - 사용자가 필요로 하는 정보들을 모두 수용 할 수 있는 구조로 디자인 해야 함
- 목표 : 데이터를 정확하고, 효율적으로, 적합한 표현을 통해 관리 할 수 있도록 함

Database design procedure





Main steps for database design

- Requirement collection and analysis
- Conceptual database design
- Logical design (or data model mapping)
- Physical design
- System implementation and tuning



Phase 1: Requirements collection and analysis

- Activities

- 요구사항 수집 및 정리

- 목적 : 사용자 그룹과 application scope를 확인
 - 방법 : 현행업무조사, 관련 문서 수집, 사용자 면담 등
 - 정리 : 현행 업무, 프로그램, 데이터 등을 조사하고 그에 대한 내용을 문서화

- 데이터 모델링을 위한 요구사항 분석 및 상세화

- 목적 : 분석 대상 정의 및 상세 요구사항 확인
 - 다양한 요구사항 상세화 방법이 있음

Phase 1: Requirements example (COMPANY DB)

- 학술정보회사 Inha Corp.는 사내 직원들의 인사관리를 위해 필요한 간단한 ERP 프로그램을 개발하고자 함
- ERP 프로그램은 사원의 개인정보 및 가족관계, 부서 및 해당부서가 진행하는 프로젝트 그리고 개별 사원이 참여하고 있는 프로젝트 등의 정보를 관리할 수 있어야 함.
- 사원의 개인정보는 성명, 사번, 생년월일, 주소, 성별, 급여 등을 포함하며, 프로그램에서는 각 사원의 사수(Supervisor)와 부양가족의 정보를 확인할 수 있어야 함
- 부양가족의 정보는 성명, 성별, 생년월일, 가족관계 등을 포함함
- 부서에 관한 정보는 프로그램에서 부서명, 해당 부서의 장의 성명과 사번, 장의 임기 시작일 등을 확인할 수 있어야 하며 부서에 따라 필요한 경우 여러 사이트들을 가질 수 있으므로 부서의 위치정보들을 관리 필요
- 프로젝트는 동일 프로젝트 명으로 여러 프로젝트가 진행 될 수 있으며 각각에 관한 진행 부서, 및 위치정보를 따로 관리할 수 있어야 함



Phase 1: Requirements collection and analysis (cont'd.)

- 요구사항 상세화를 위한 **technique**들
 - Using Design Documents (basic and simple way)
 - Object-Oriented Analysis (OOA)
 - Data flow diagrams (DFDs)
 - Refinement of application goals
 - Computer-aided (CASE tools)

Phase 1: Requirements collection and analysis (cont'd.)

- 요구사항 분석 및 상세화 과정

- 요구사항을 정리한 문서로부터 Entity type 정의에 사용될 **명사** 구분
- 요구사항을 정리한 문서로부터 Entity type들 사이의 Relationship에 사용될 **동사** 구분
- 개념이 불분명하거나 광범위한 명사/동사 제거
- 다른 명사에 종속적이거나 **속성/특성**으로 보이는 것은 **Attribute**로 사용될 수 있으므로 별도 분리
- 포괄적 업무프로세스(기능)에 해당하는 명사 제거
- 중복 / 동의 / 유의 명사/동사 제거 및 별도 기록
- 지금까지 누락시킨 명사 / 동사 중 해당 단어가 없으면 요구사항의 수렴이 불가능한 단어 확인

Practice1: Requirement analysis example (COMPANY DB)

Entity	Explanation	Attribute	Relation with	(Note)
Employee	Storing information of employees	Name		
		Employee_Number		Key Attribute
		Birthday		
		Address		
		Sex		
		Salary		
		Participated_Project		Multi value Attribute
		Supervisor	Employee	
		Affiliated_Department	Department	

Entity	Explanation	Attribute	Relation with	(Note)
Project	Storing information of projects	Project_Name		
		Project_Number		Key Attribute
		Project_Location		
		Department	Department	

Entity	Explanation	Attribute	Relation with	(Note)
Dependent	Storing dependent information of employee	Employee	Employee	Key Attribute
		Name		Partial Key Attribute
		Sex		
		Birthday		
		Relation		

Entity	Explanation	Attribute	Relation with	(Note)
Department	Storing information of departments	Department_Name		Key Attribute
		Department_Number		Key Attribute
		Manager_EN	Employee	
		Managing_Start_Date		
		Location		Multi value Attribute

Table 1. COMPANY DB : Requirement analysis example



Phase 2: Conceptual database design

- **Conceptual database design**

- 데이터 모델링의 기초가 되는 High-level의 개념적 데이터 모델을 만들기 위한 중요한 과정

- Approaches

- **Centralized (or one shot) schema design approach**
 - **View integration approach**



Phase 2: Conceptual database design (cont'd.)

- Strategies of centralized (or one shot) schema design approach
 - **Top-down(↓)**
 - **Bottom-up(↑) : 속성 통합(attribute synthesis)**
- Schema (view) integration(통합) approach
 - 스키마들 사이의 이질성 존재
 - 이름, 타입, 도메인, 단위, 의미 등
 - 한쪽의 스키마를 다른 쪽에 맞추어 수정하거나 통합, 재구성 함



Phase 2: Conceptual database design (cont'd.)

- **Transaction Design**

- ‘Conceptual database design’과 동일한 의미
- Transaction을 conceptual design level에서 정의
- **Input/output** 과 **functional behavior** 등을 정의
- 특정 프로세스들을 위한 Notation을 정의

Practice 2: Conceptual database design

• 과정

1. Entity type 추출

- Entity : 실체가 존재하는 것(Thing)으로 해당 업무의 프로세스에서 필요하고 관리, 이용하고자 하는 정보
- Instance들의 집합(즉, 2개 이상의 instance가 있어야 함)

2. Attribute 선정 및 key attribute 설정

- Attribute : entity를 설명하기 위한 속성으로 instance의 구성요소
- 의미상으로 더 이상 분리되지 않으며(atomic) 정해진 범위의 값만을 가짐(domain)
- Key attribute : 각각의 instance를 서로 구분 가능하게 하는 attribute로 각각의 instance가 고유한 값을 가짐
- Attribute
 - Derived(유도) attribute : 다른 attribute를 통해 값을 유도
 - Multivalued attribute : 여러 값을 동시에 가질 수 있는 속성
 - Composite attribute : 몇 개의 단순 속성들로 분해 할 수 있는 속성

3. Relationship type 설정

Practice example (COMPANY DB)

Entity	Explanation	Attribute	Relation with	(Note)
Employee	Storing information of employees	Name		Composite attribute
		Employee_Number		Key Attribute
		Birthday		
		Address		
		Sex		
		Salary		
		Participated_Project		Multi value Attribute
		Supervisor	Employee	
		Affiliated_Department	Department	

Entity	Explanation	Attribute	Relation with	(Note)
Project	Storing information of projects	Project_Name		
		Project_Number		Key Attribute
		Project_Location		
		Department	Department	

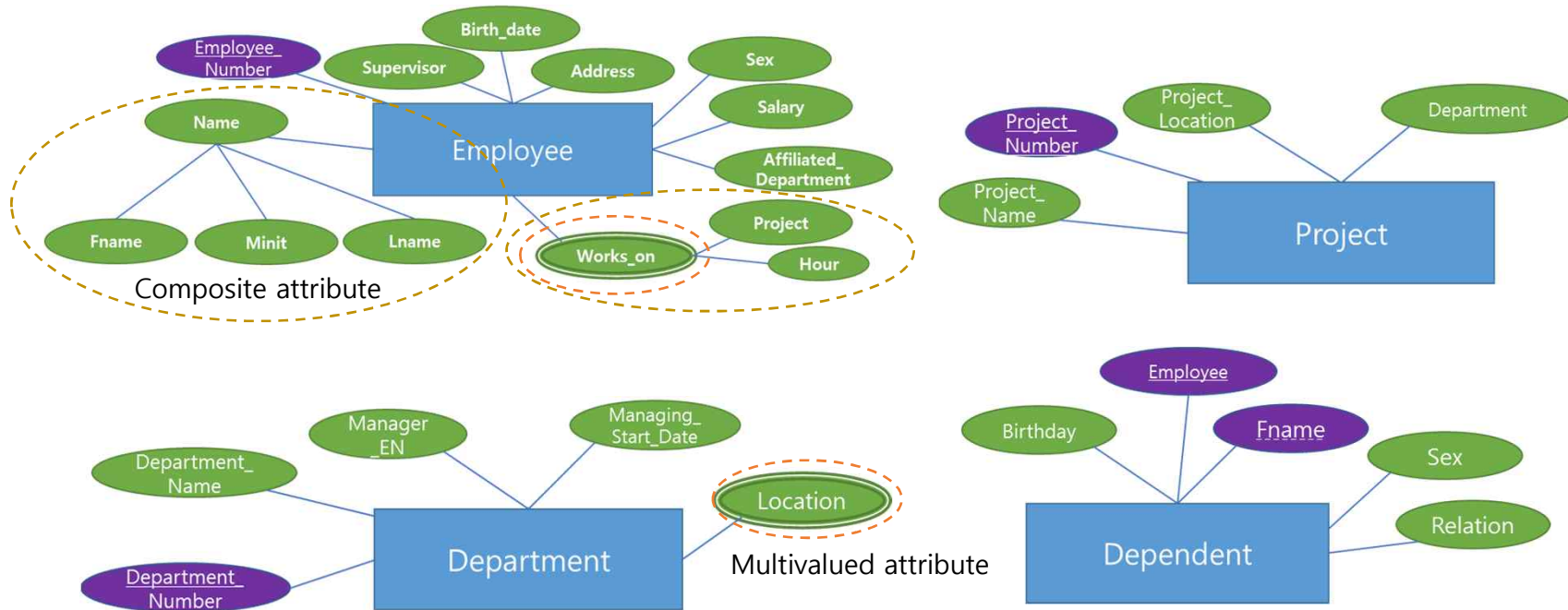
Entity	Explanation	Attribute	Relation with	(Note)
Dependent	Storing dependent information of employee	Employee	Employee	Key Attribute
		Name		Partial Key Attribute
		Sex		
		Birthday		
		Relation		

Entity	Explanation	Attribute	Relation with	(Note)
Department	Storing information of departments	Department_Name		Key Attribute
		Department_Number		Key Attribute
		Manager_EN	Employee	
		Managing_Start_Date		
		Location		Multi value Attribute

Table 1. COMPANY DB : Requirement analysis example

Practice example (COMPANY DB)

(cont'd.)



Practices about Phase 2 (cont'd.)

• 과정

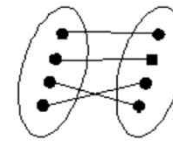
1. Entity type 추출
2. Attribute 선정 및 key attribute 설정
3. Relationship type 설정
 - Relationship : Entity들 사이의 논리적 연관성을 행위 또는 형태를 나타내는 동사로 기술한 것
 - Entity type의 속성 중 다른 entity를 참조하는 속성을 찾아 relationship type으로 변환
 - 관계 참여의 필수성(participation or optionality), 대응수(mapping cardinality), 식별성(identifying) 등에 따라 관계의 특성이 나뉘어짐

Practices about Phase 2 (cont'd.)

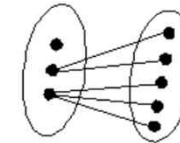
• Constraints of relationship types

– 관계 대응수(cardinality)

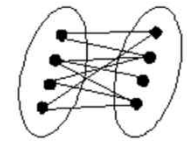
- 관계에 대응하는 최대 원소 수
 - 1:1 / 1:N / M:N 관계



1:1 관계



1:다 관계



다:다 관계

• 표기

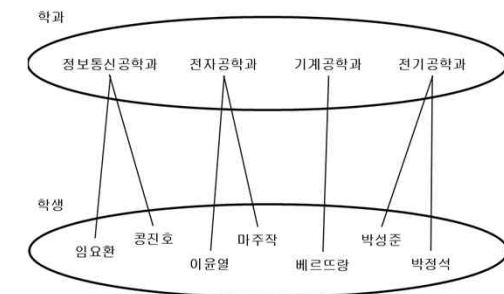
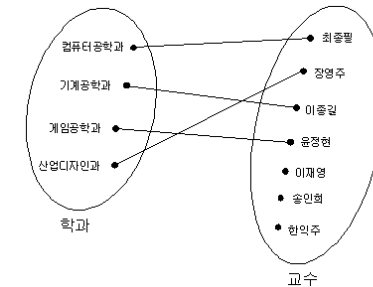
– 1:1



– 1:N



– M:N

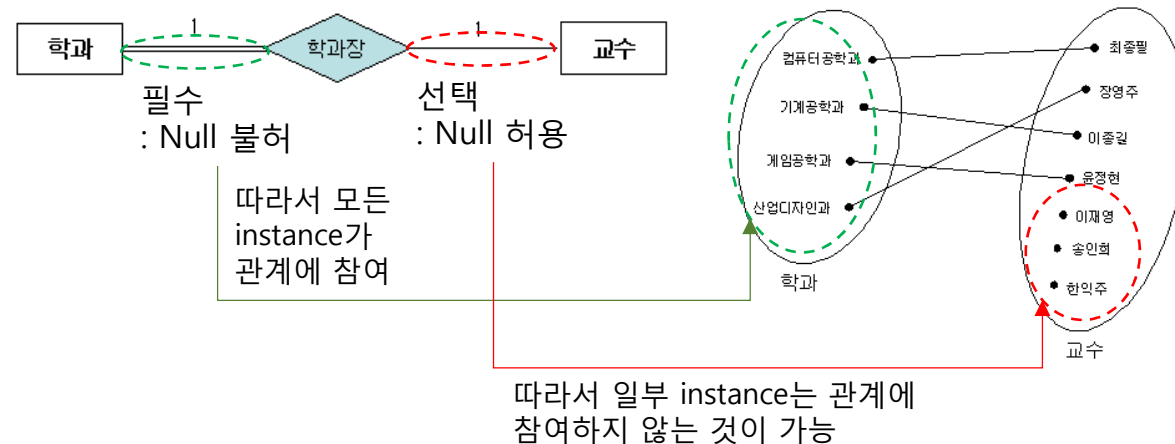


Practices about Phase 2 (cont'd.)

• Constraints of relationship types

– 관계 참여(Participation or optionality)

- Entity type의 모든 instance가 관계에 참여하는지 여부
 - 필수 / 선택 : 연관 속성의 Null 허용 여부
- 표기



Practices about Phase 2 (cont'd.)

• Constraints of relationship types

– 식별(Identifying) 관계

- Entity type간의 존재가 의존적인 관계를 파악
- 어떤 의존적인 개체 타입의 소유자 역할의 개체 타입을 식별하기 위해 식별 관계 설정 (식별 관계의 소유 개체 타입의 key 속성을 의존 개체 타입의 외부속성으로 포함)
- 다른 개체 타입으로부터 연관 속성으로 가져온 외래 속성이 현재의 개체타입의 식별자로 쓰여야 하는지 여부
 - Identifying : 외부속성이 식별자 속성이 되는 관계



- Non-Identifying : 외부속성이 일반속성이 되는 관계



Practice example (COMPANY DB)

Entity	Explanation	Attribute	Relation with	(Note)
Employee	Storing information of employees	Name		Composite attribute
		Employee_Number		Key Attribute
		Birthday		
		Address		
		Sex		
		Salary		
		Participated_Project		Multi value Attribute
		Supervisor	Employee	
		Affiliated_Department	Department	

Entity	Explanation	Attribute	Relation with	(Note)
Project	Storing information of projects	Project_Name		
		Project_Number		Key Attribute
		Project_Location		
		Department	Department	

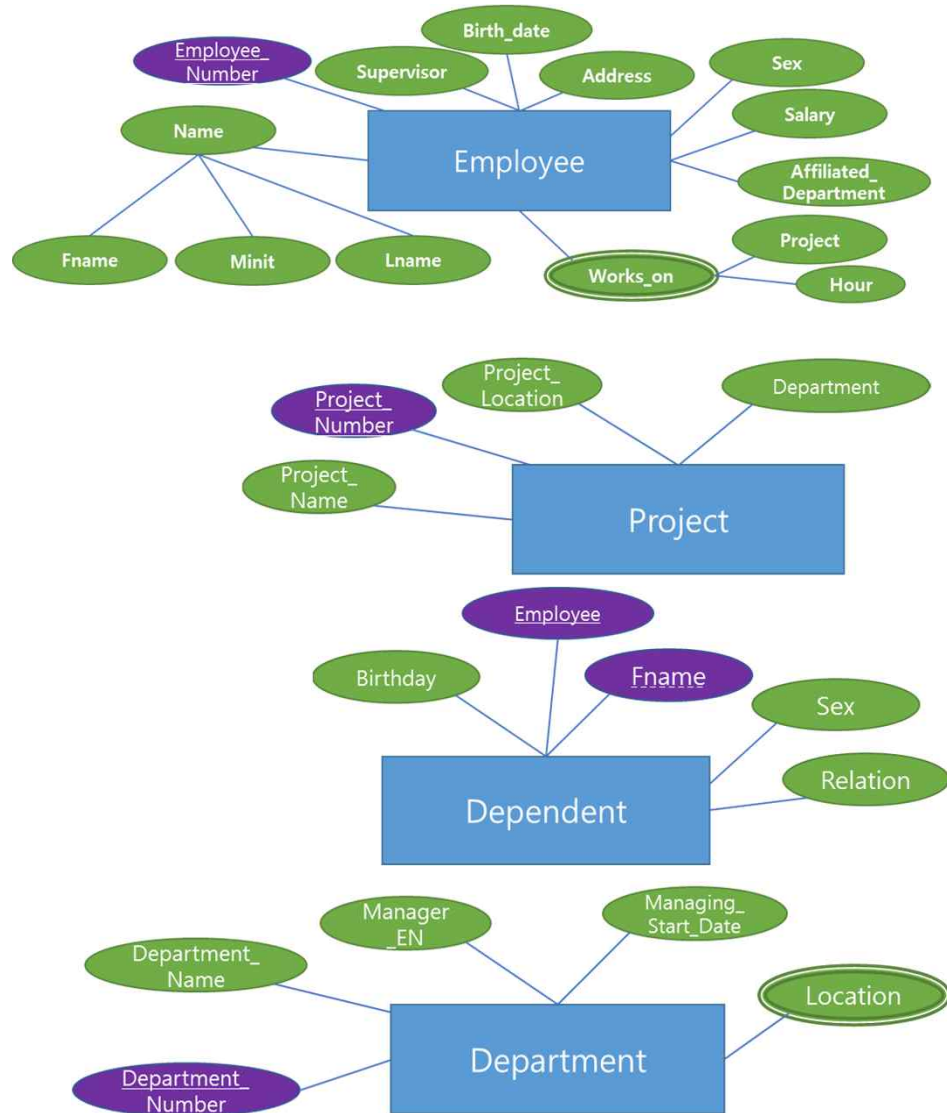
Entity	Explanation	Attribute	Relation with	(Note)
Dependent	Storing dependent information of employee	Employee	Employee	Key Attribute
		Name		Partial Key Attribute
		Sex		
		Birthday		
		Relation		

Entity	Explanation	Attribute	Relation with	(Note)
Department	Storing information of departments	Department_Name		
		Department_Number		Key Attribute
		Manager_EN	Employee	
		Managing_Start_Date		
		Location		Multi value Attribute

Table 1. COMPANY DB : Requirement analysis example

Practice example (COMPANY DB)

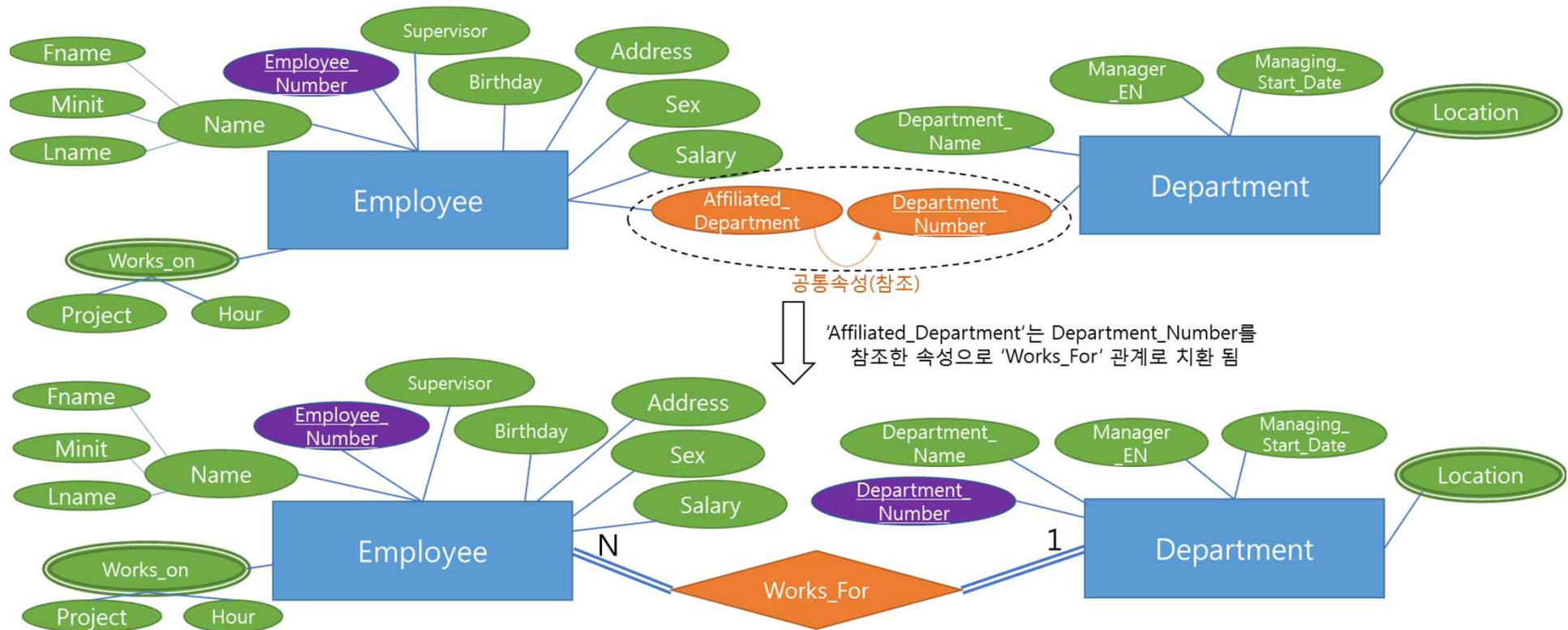
(cont'd.)



하나의 다이어그램으로 만듦

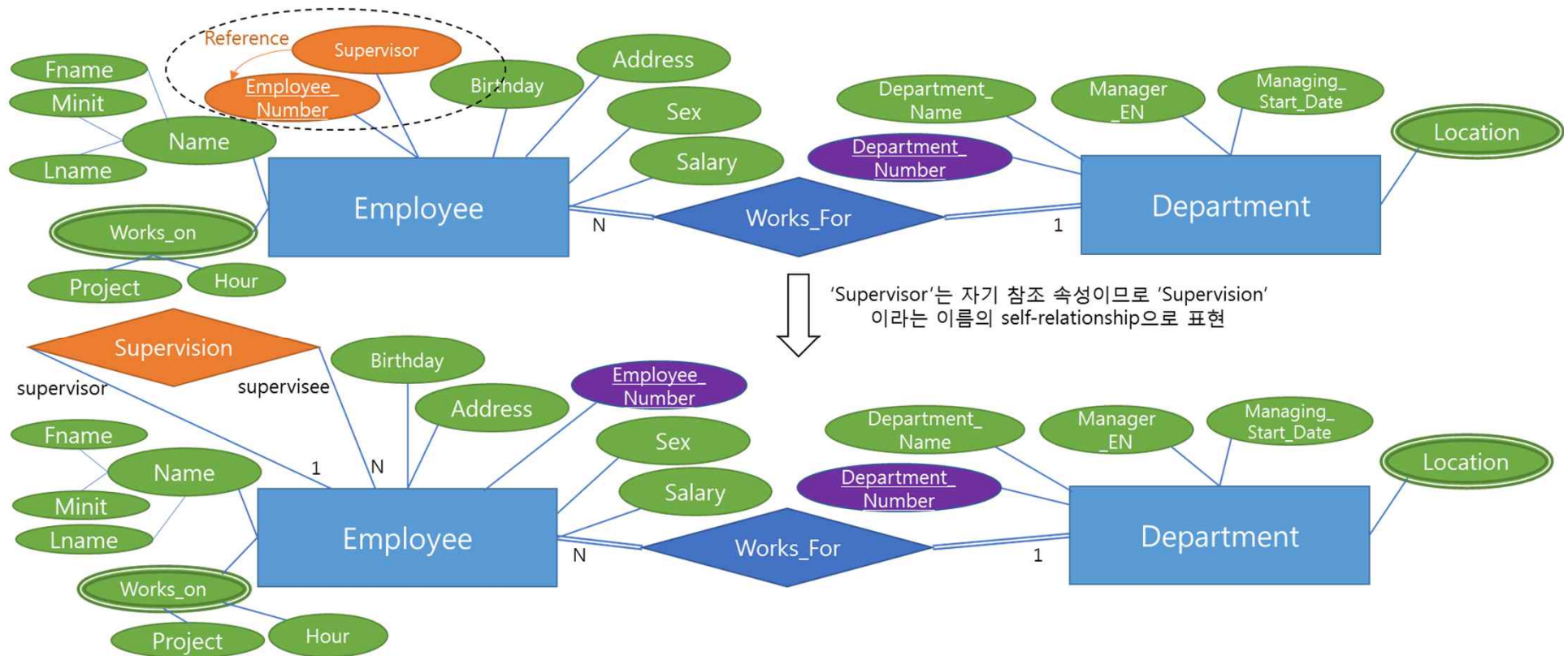
Practice example (COMPANY DB)

(cont'd.)

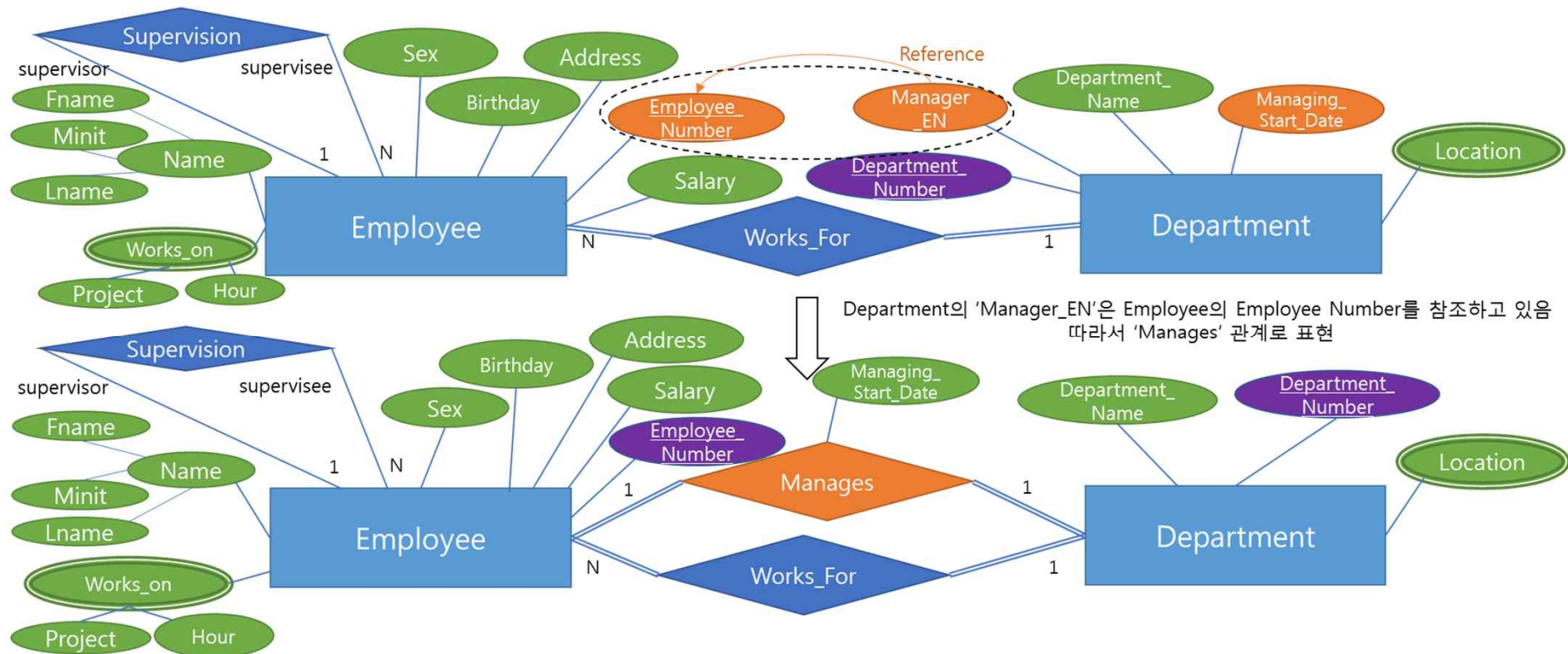


Practice example (COMPANY DB)

(cont'd.)

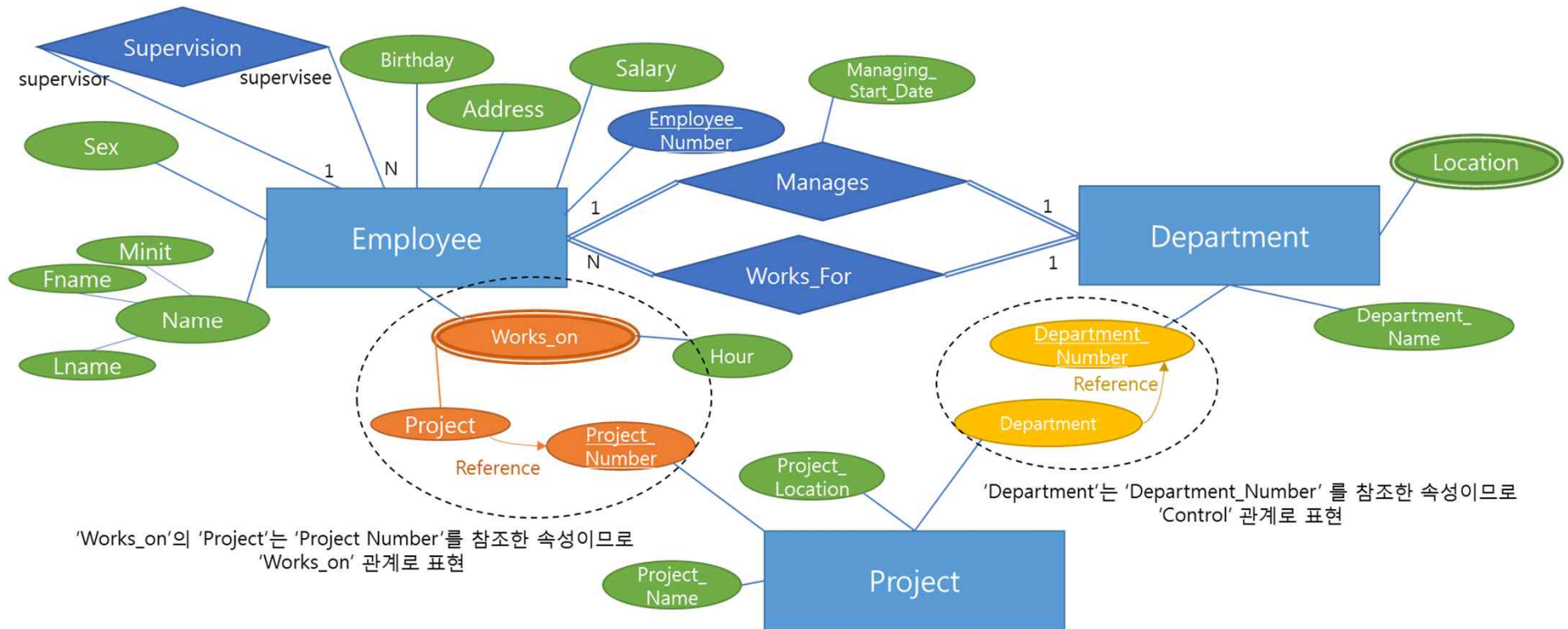


Practice example (COMPANY DB) (cont'd.)



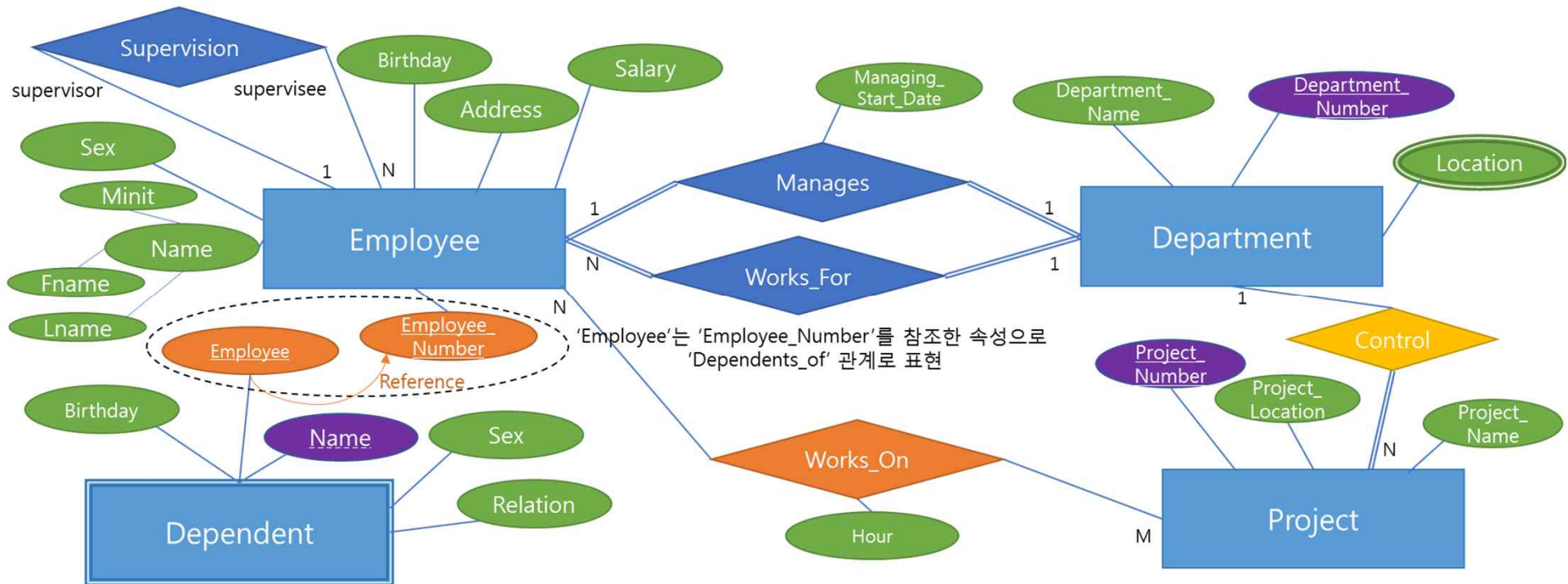
Practice example (COMPANY DB)

(cont'd.)



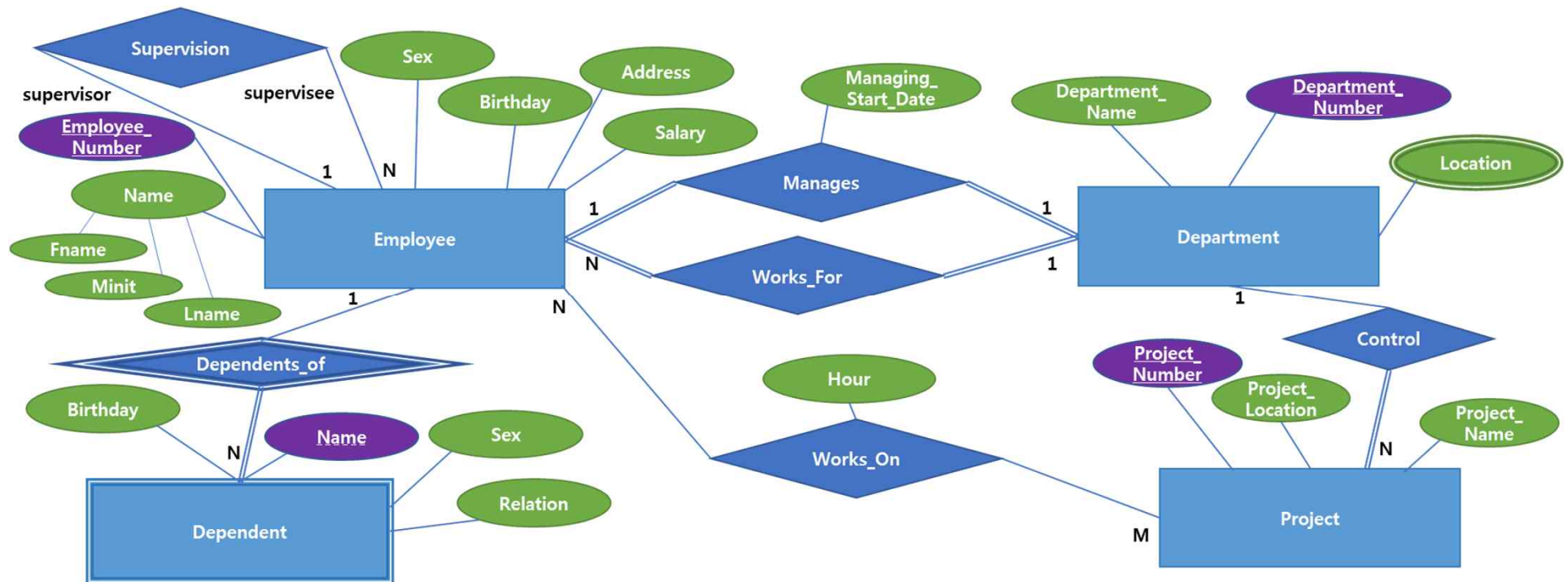
Practice example (COMPANY DB)

(cont'd.)



Practice example (COMPANY DB)

(cont'd.)



Phase 3: Choice of DBMS

- DBMS를 정할 때 여러 요인들을 고려해야 함
 - 기술적 issues
 - DBMS type (relational, OR, OO ..)
 - Storage structure and access methods
 - User & programmer interfaces
 - High-level query languages
 - Development tools
 - 비용적 issues
 - Software acquisition cost and maintenance cost
 - Hardware acquisition cost
 - Database creation and conversion cost
 - Personnel cost and training cost
 - 주요 DBMS
 - MSSQL
 - ORACLE
 - MYSQL
 - MariaDB 등

Phase 4: Logical design (data model mapping)

- 개념 다이어그램을 선택된 DBMS에 맞는 개념 스키마 및 외부 스키마를 설계하는 단계
- Stages
 - System-independent mapping
 - Tailoring schemas to a specific DBMS

Practices about Phase 4

- 대상 Tool : ER-Win

- Data modeling tool

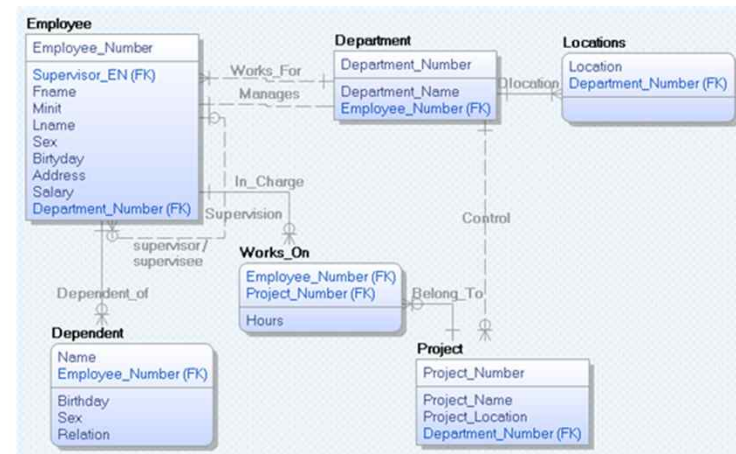
- 논리 데이터 모델을 만들기 위한 CASE tool
- 개념 설계 결과를 논리 스키마로 바꾸기 위해 사용
- ER-Diagram을 만드는데 주 목적이 있음
 - 스키마를 만들기 위한 DDL을 작성 지원해주는 기능도 있음 (forward-engineering)

Figure 9.1
The ER conceptual schema diagram for the COMPANY database.



Example of conceptual diagram

convert
→



Example of logical diagram

Target version-Erwin Data Modeler r9.7

• 프로그램 실행

- [start button] → [program] → [Erwin] → [AllFusion] → [Erwin Data Modeler r9.7] → [Erwin Data Modeler r9.7] 실행

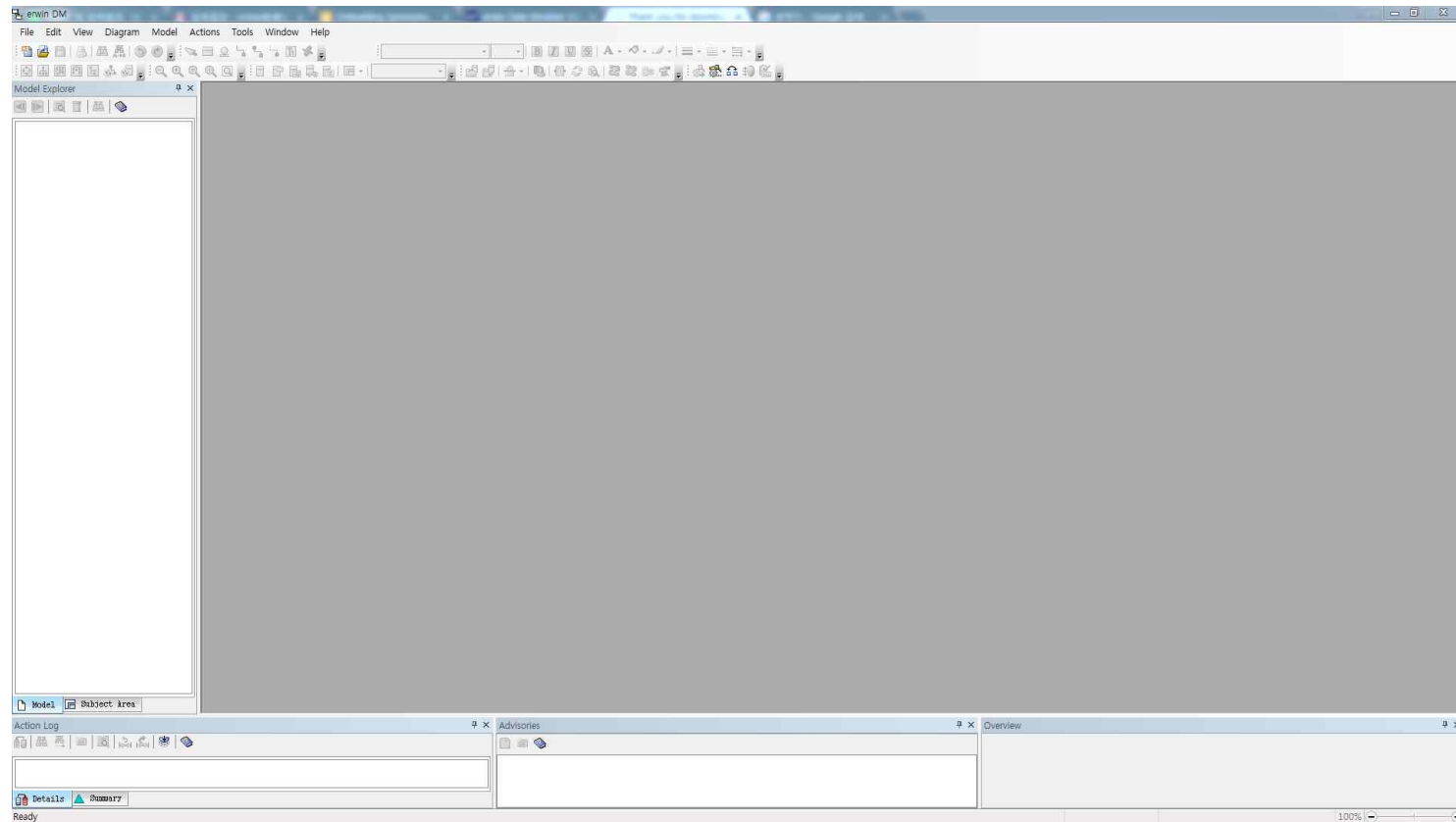
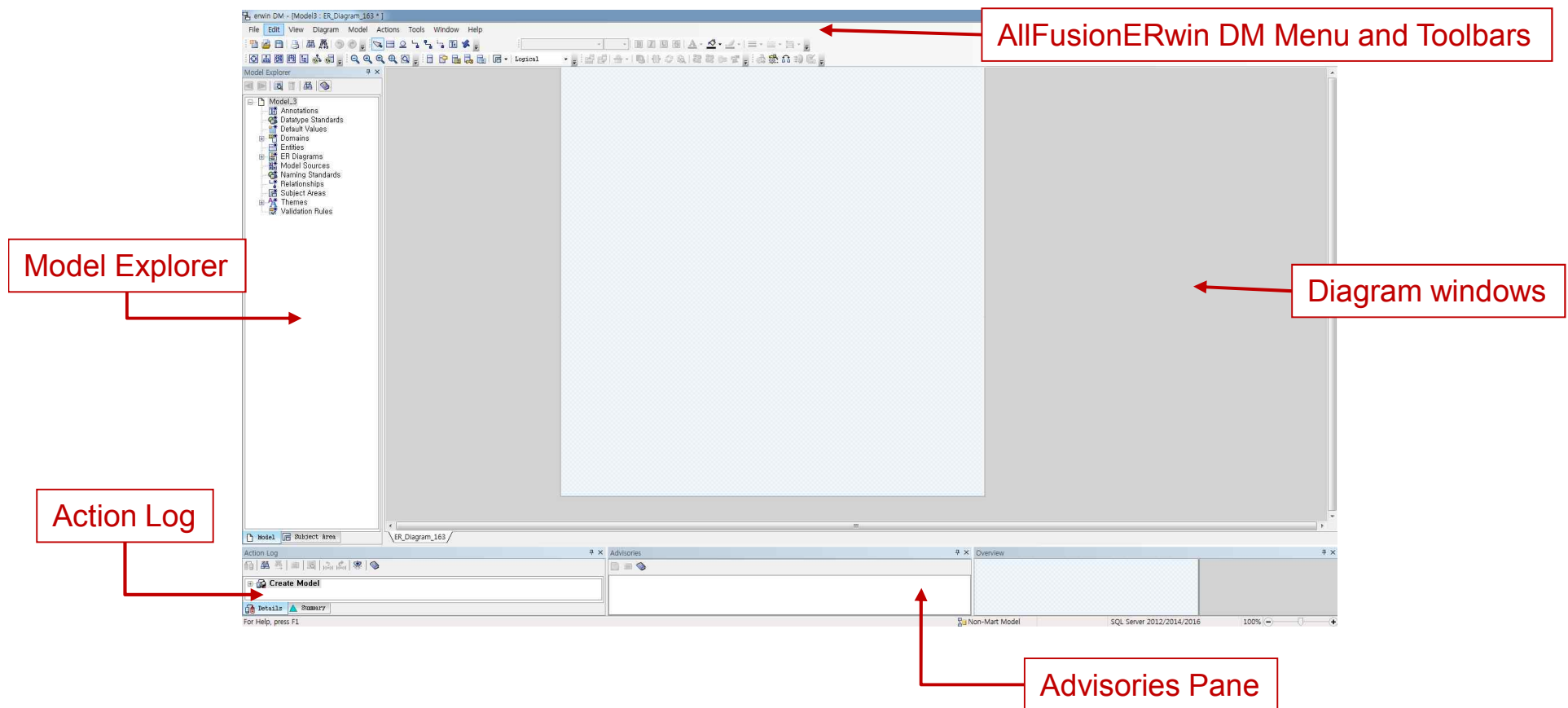


Figure 1. Initial screen

Basic screen configuration

- 초기 화면은 1개의 메뉴 바와 4개의 윈도우로 이루어져 있음.

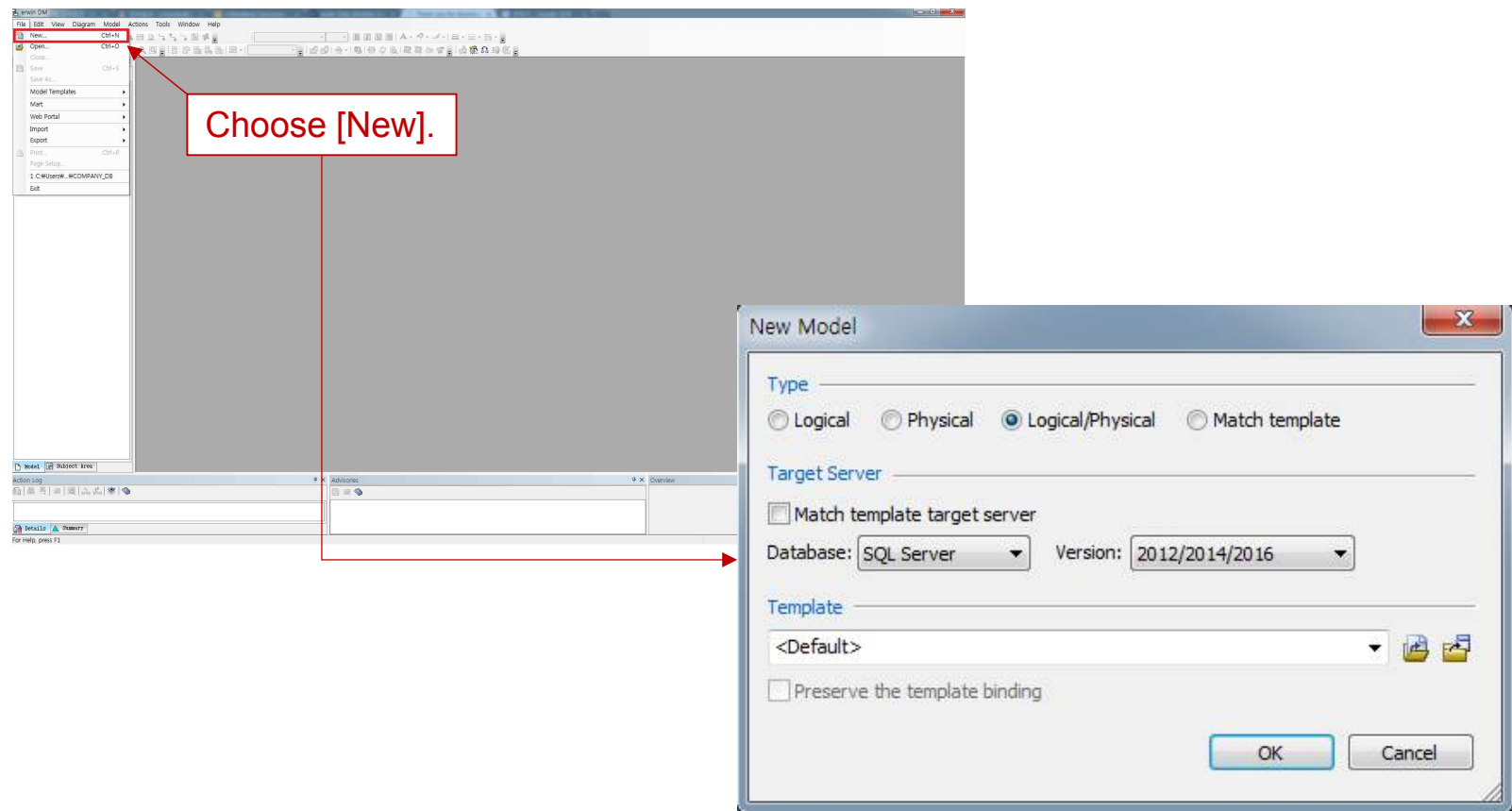


Basic screen configuration (cont'd.)

Division	Function
Toolbars	<ul style="list-style-type: none">➤ 모델링 작업에 필요한 버튼들을 모아둠.➤ 사용자 정의에 따라 구성을 변경 할 수 있음.
Advisories Pane	<ul style="list-style-type: none">➤ 현재 유저에 의해 진행되었던 프로세스에 대한 일정량의 정보를 보여줌.
Model Explorer	<ul style="list-style-type: none">➤ 모델의 explorer.➤ 여기에 보이는 각각의 object는 계층적 구조로 구성됨.
Action Log	<ul style="list-style-type: none">➤ 현재 유저에 의해 진행되는 작업을 실시간으로 기록.
Diagram Windows	<ul style="list-style-type: none">➤ 유저는 이 영역에서 모델링 작업을 수행하며, 새 모델을 생성하면 이 영역은 기본적으로 'Display1'이라는 이름을 가짐.

Choose model type and target DBMS

- 메뉴 바에서 **[File]** 선택 → '**New Model**' dialog box를 열기 위해 **[New]** 탭 선택.



Choose model type and target DBMS

- **Choose model type**

- **‘Logical’ model**

- Attribute, primary key 등과 같은 entity type과 relationship type의 상세한 정보들을 논리적 모델 디자인에 포함시키기 위해 사용됨.
 - Target database나 version, 트리거, 저장프로시저 등 database 자체에 관한 특성에 관한 설계사항은 정의할 수 없음.

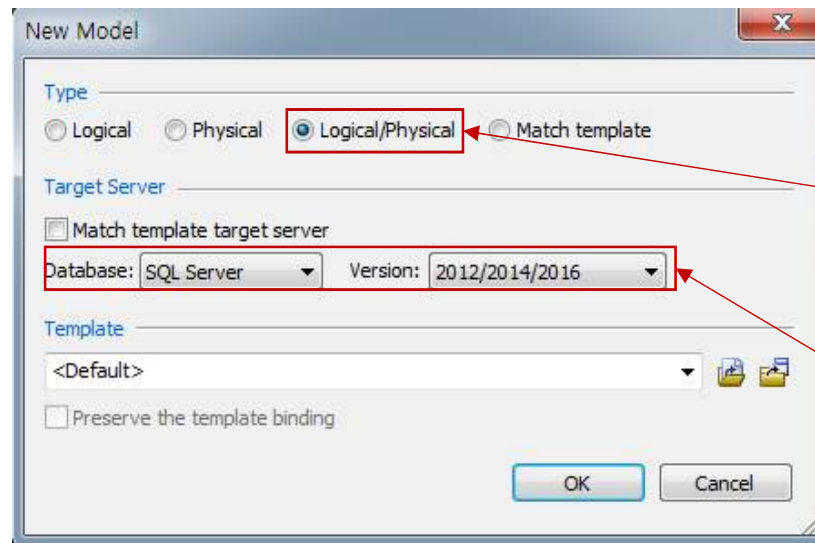
- **‘Physical’ model**

- Database 자체에 관련된 특징들을 물리적 모델 디자인에 포함시키기 위해 사용됨.
 - 모델의 논리적 특성과 관련된 사항들은 정의할 수 없음.

➤ 따라서, 정확, 상세한 데이터 모델을 디자인하기 위해 **‘logical/physical’ model type** 을 선택해야 함.

Choose model type and target DBMS

- **Choose model type and target DBMS**
 - 본 실습에서 사용할 target database server는 **“SQL Server 201X”**.



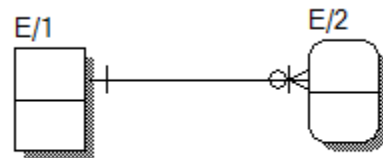
Click [Logical/Physical] type.

Choose [SQL Server] database.
Choose [2012/2014/2016] version.

Choose data model diagram notation

- **Information Engineering(IE)**

- 선들과 원의 조합을 사용하는 notation.
- 그 모양이 까마귀 발과 닮아 crow's foot이라고도 함.
- 가장 많이 사용되는 notation



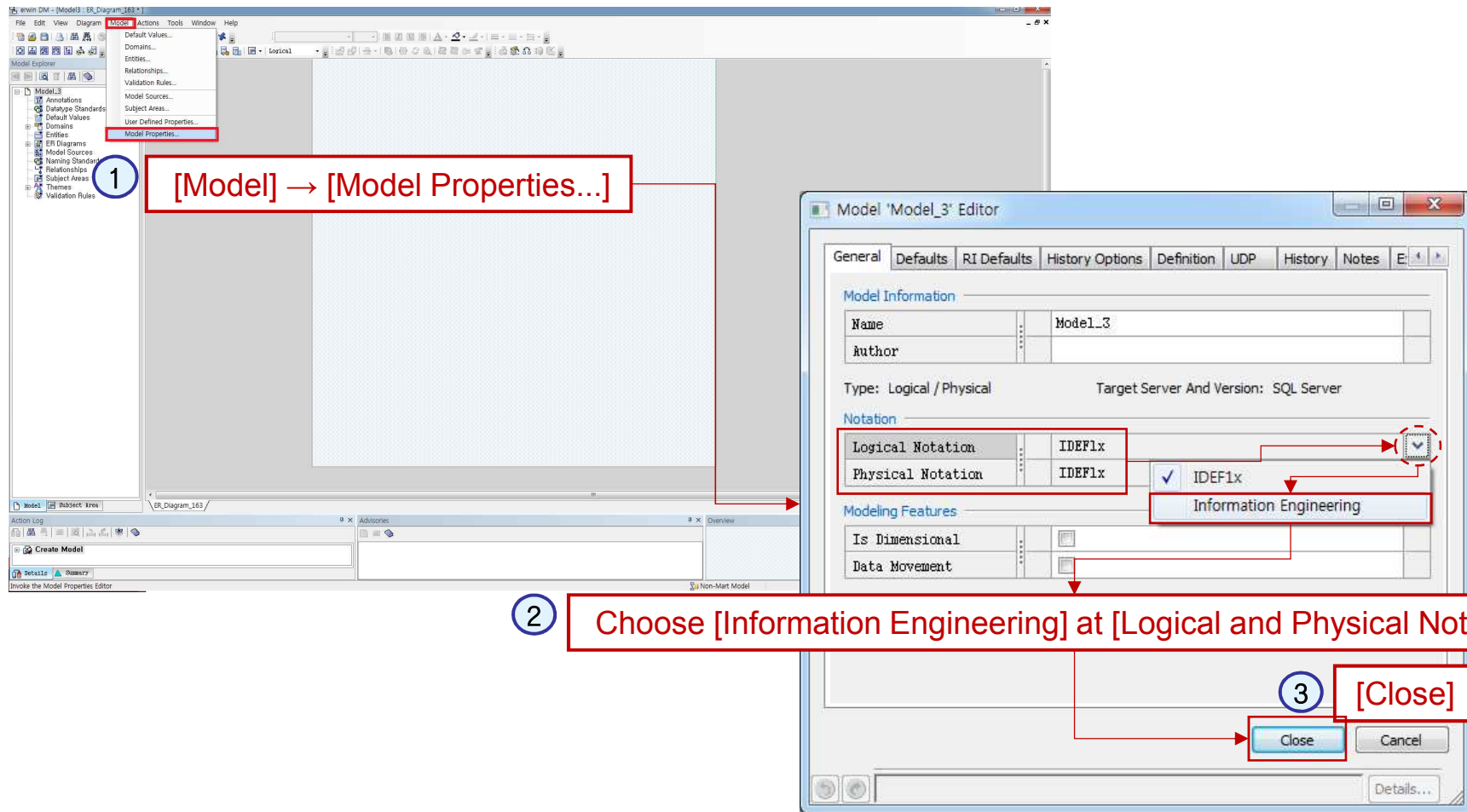
- **IDEF1X**

- 원과 다이아몬드 모양을 사용하는 notation.
- ER-Win의 기본 notation



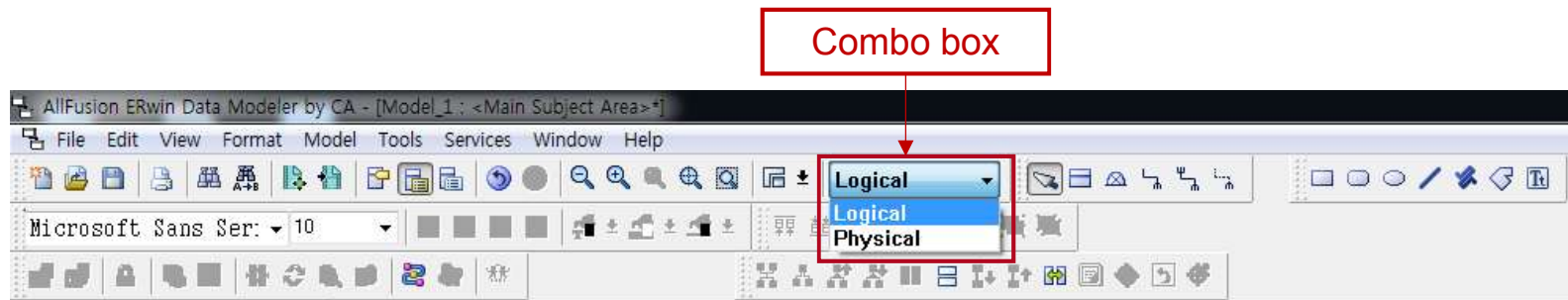
Choose data model diagram notation (cont'd.)

- 'IDEF1X' → 'Information Engineering'



Choose data modeling phase

- 데이터 모델링 방법('Logical'/'physical') 선택법.
 - 메뉴 바의 combo box에서 'Logical/Physical' type model 모델 중 자신이 작업할 model type을 선택
 - 각 방법의 상세사항은 slide 35 참고.



Logical data modeling with ER-Win

Target model (Company DB)

- **logical data modeling with ER-Win**
 - Step 1: Entity 그리기
 - Step 2: 1에서 그린 Entity들을 적절히 배치
 - Step 3: Relationship 설정
 - Step 4: Relationship에 이름 할당
 - Step 5: Relationship의 차수 등 상세사항 설정

Logical data modeling with ER-Win

Target model (Company DB) (cont'd.)

- 아래 예시의 company database의 개념다이어그램을 바탕으로 logical data modeling을 진행.

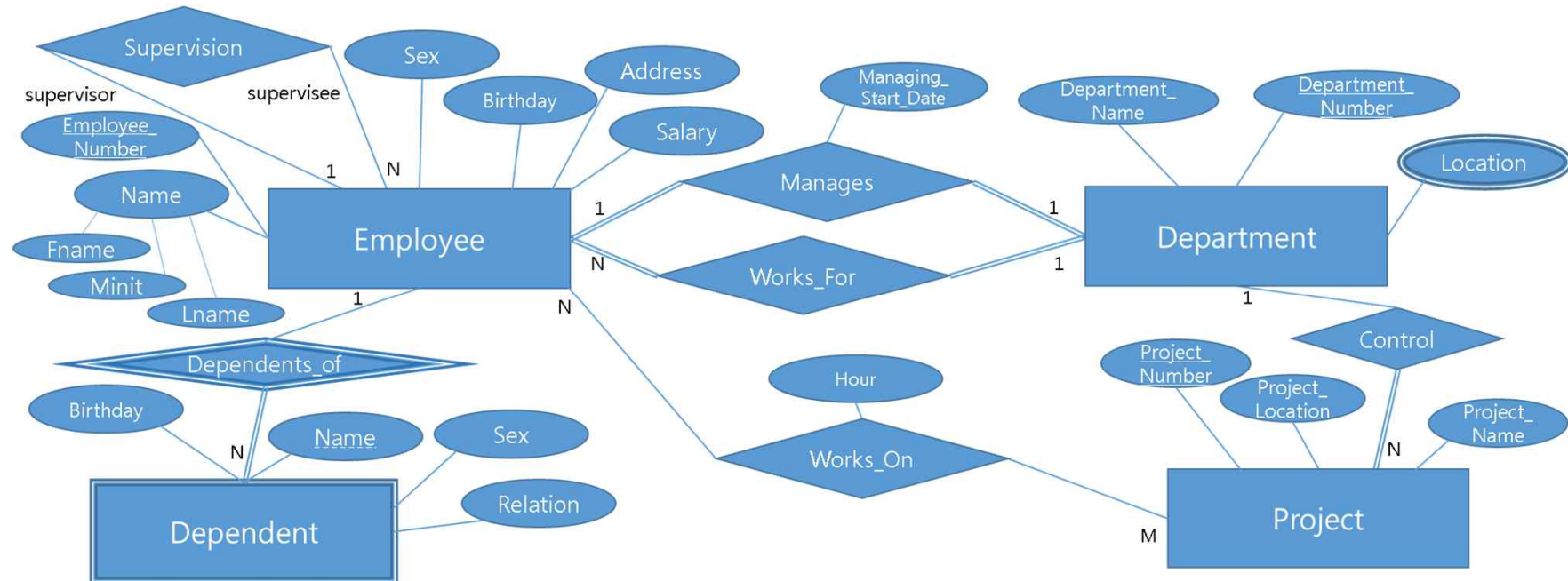


Figure 2. Conceptual diagram of company DB

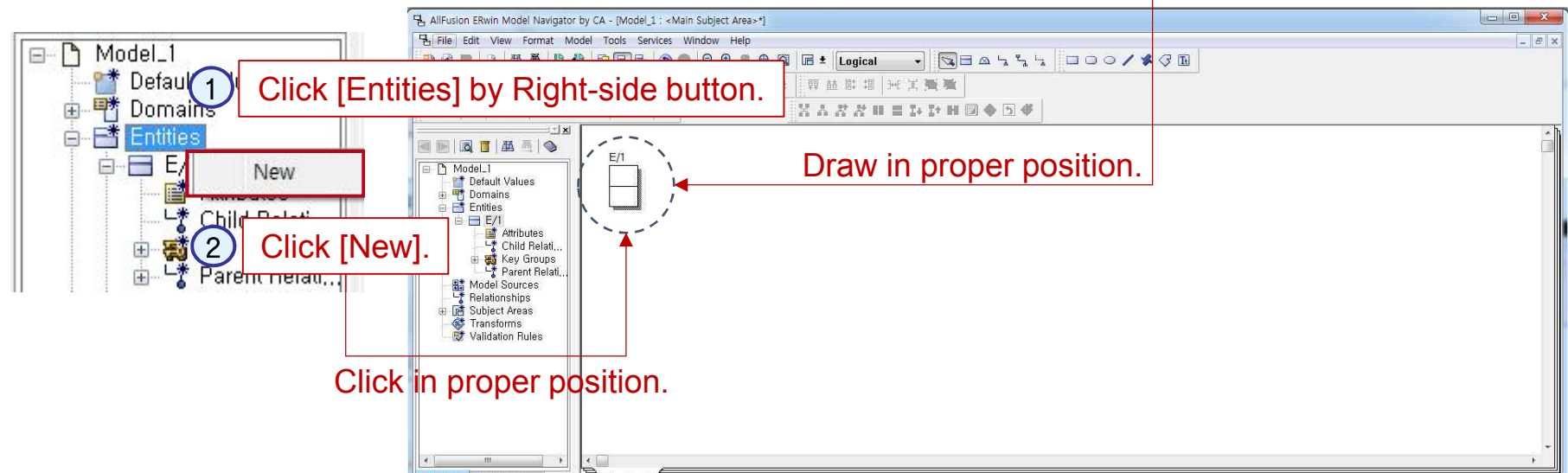
Step 1: Entity 그리기

- Entity 는 아래 두 가지 방법으로 만들 수 있음.

- 방법 1: toolbar 이용하기



- 방법 2: model explorer 이용하기



Step 1: Entity 그리기 (cont'd.)

• Entity

Entity 명

primary key attribute를 위한 공간

normal attribute를 위한 공간

- 각 공간에 내용을 작성하는 법
 - ① Entity 선택
 - ② 'Tap' key를 눌러 작성 할 공간으로 이동
 - ③ 내용 작성

작성 후

Employee

Employee_Number

Fname

Minit

Lname

Sex

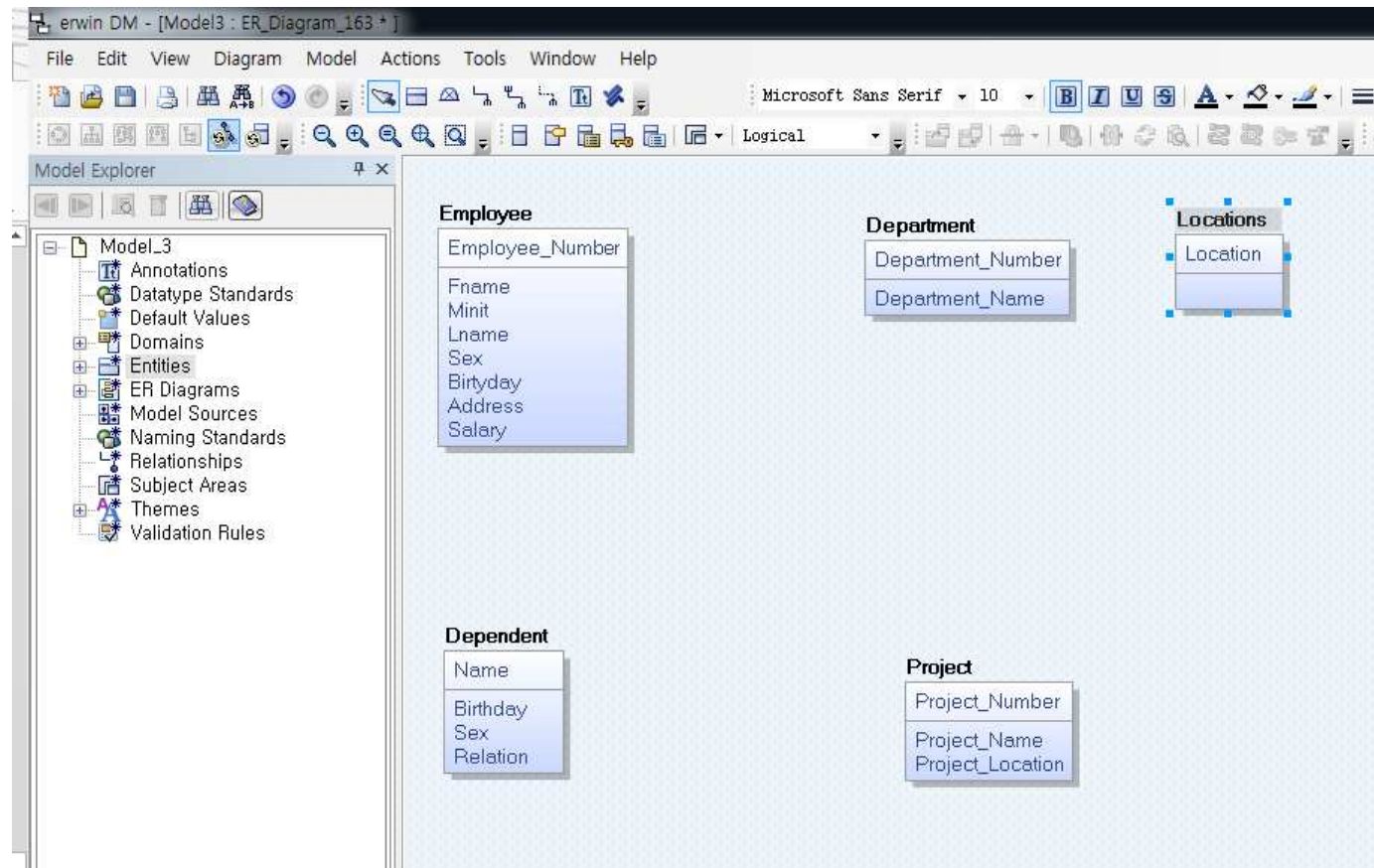
Birthday

Address

Salary

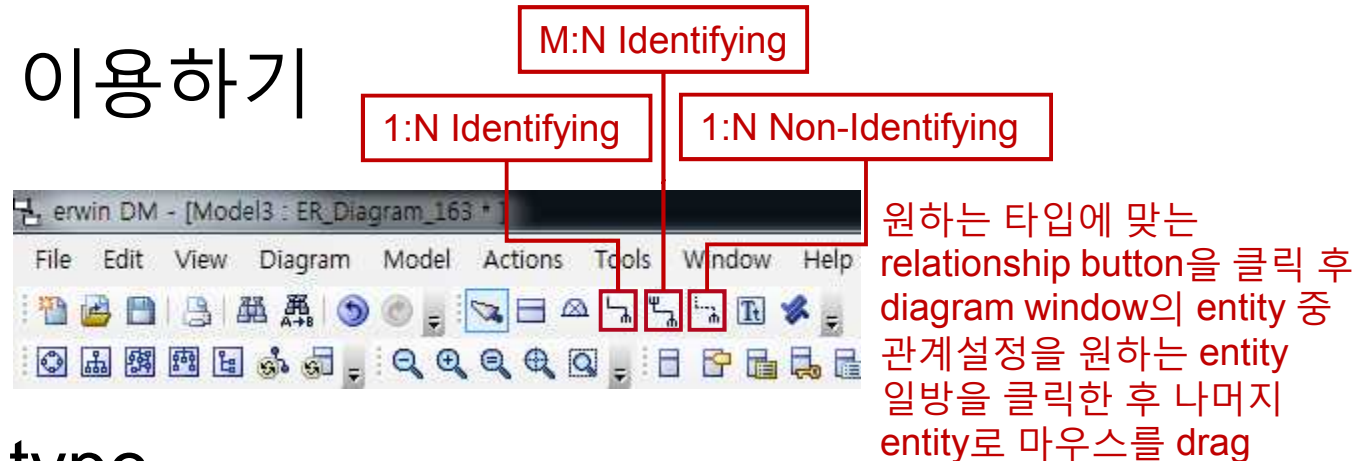
Step 2: Entity 배치

- Entity들을 윈도우 내 공간에 적절히 배치



Step 3: Relationship 설정

- 방법: toolbar 이용하기



- Relationship type
 - **Identifying** (Owner entity(parent) ↔ weak entity(child)):
 - Parent의 primary key가 child의 primary key로 사용됨.
 - Child는 entity의 테두리가 원형으로 변함.
 - **Non-Identifying** (between strong entity):
 - Parent의 primary key가 child의 normal attribute로 사용됨.



Figure 3. Example of identifying

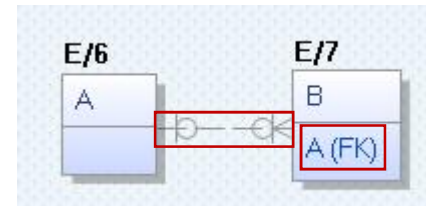
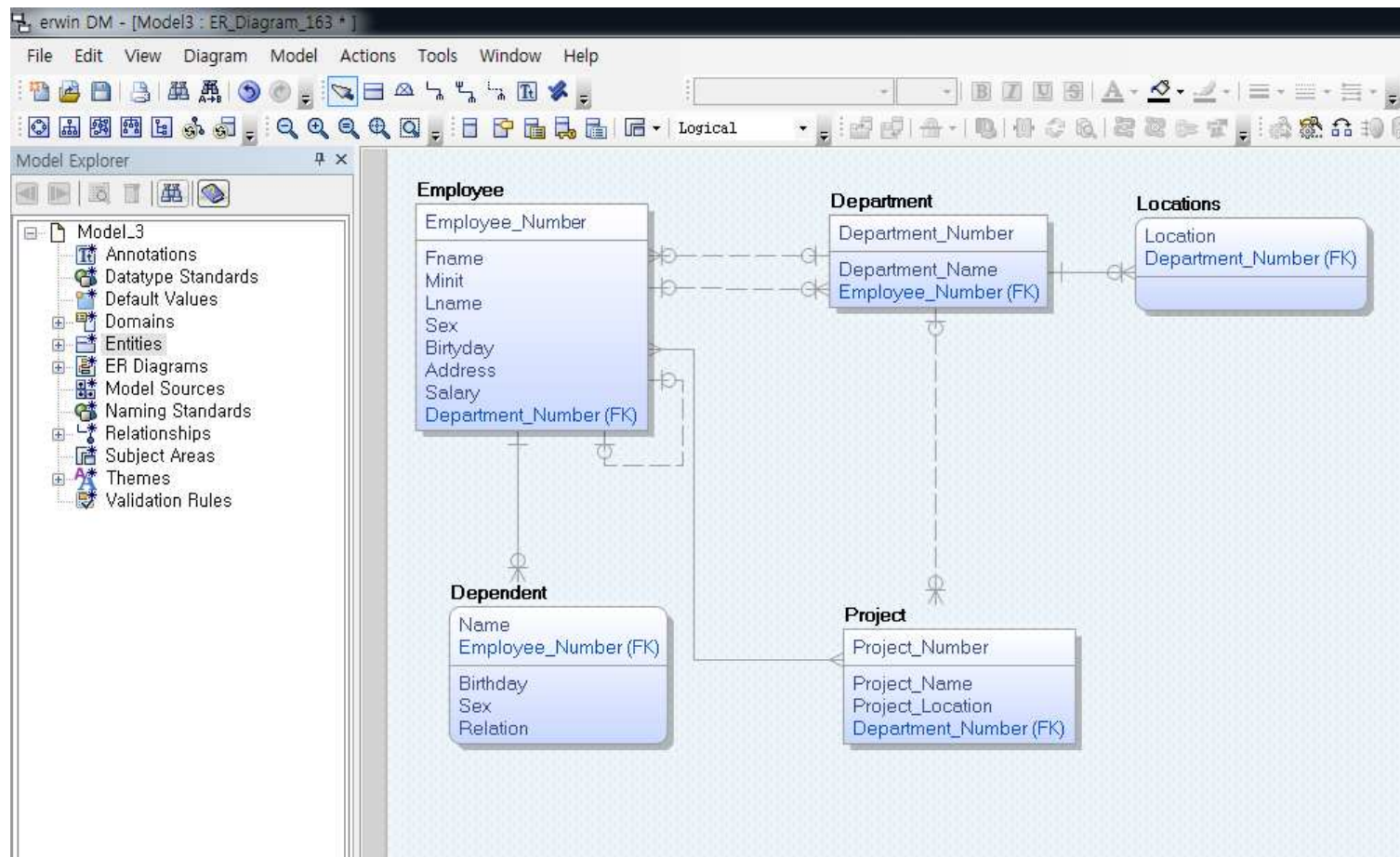


Figure 4. Example of non-identifying

Step 3: Relationship 설정(cont'd.)

- Relationship 설정 결과



Step 4: Relationship에 이름 할당

- 방법: relationship property dialog box 사용

The diagram illustrates the steps to name a relationship in a database model:

1. 오른쪽 버튼 click. (Right button click.)
2. Choose

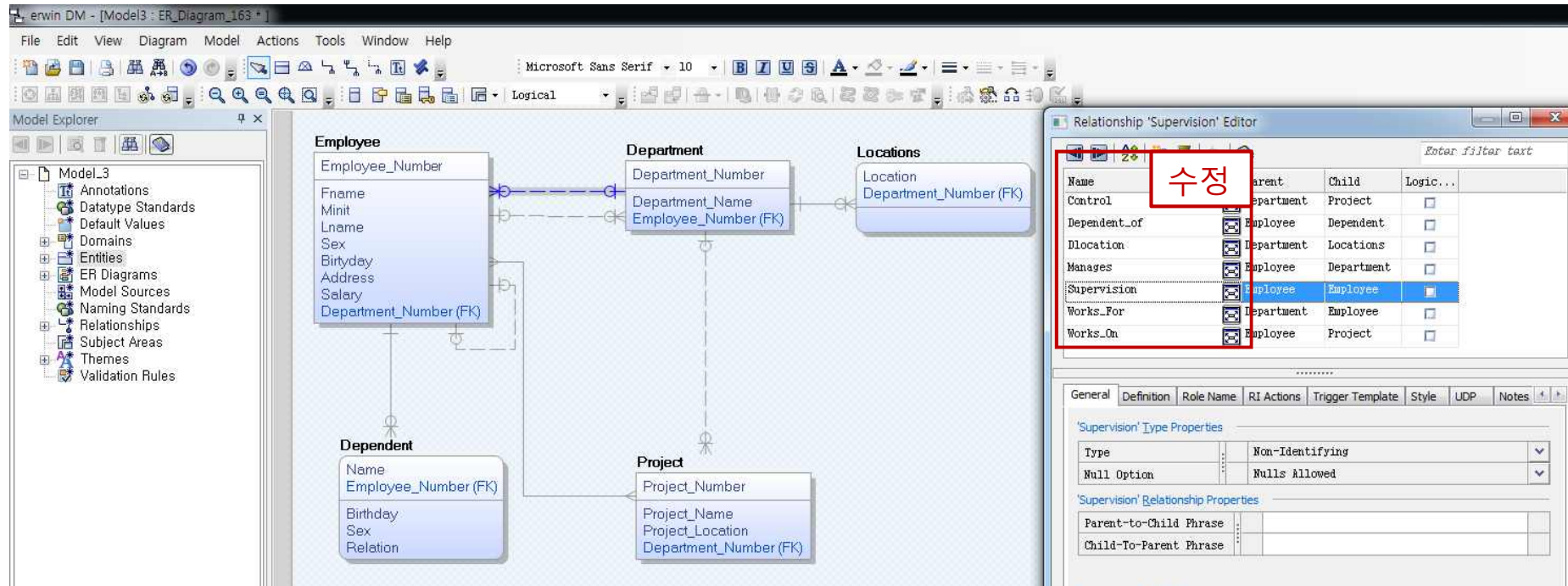
The 'Relationship 'R/12' Editor' dialog box is shown, with the 'General' tab selected. The 'Name' field is highlighted, and the text 'Relationship 이름' (Relationship name) is written next to it.

The dialog box contains the following sections:

- 'R/12' Type Properties**
 - Type: Identifying
- 'R/12' Relationship Properties**
 - Parent-to-Child Ph...
 - Child-To-Parent Ph...
- 'R/12' Cardinality Properties**
 - Cardinality: Zero, One or More
 - Cardinality Value

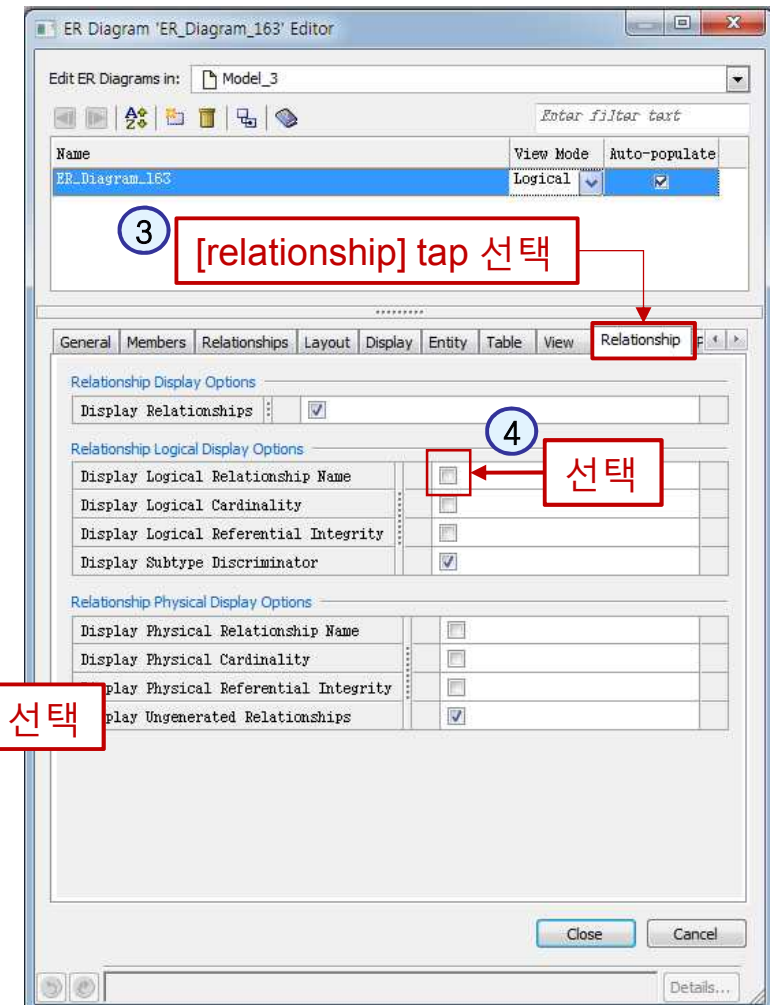
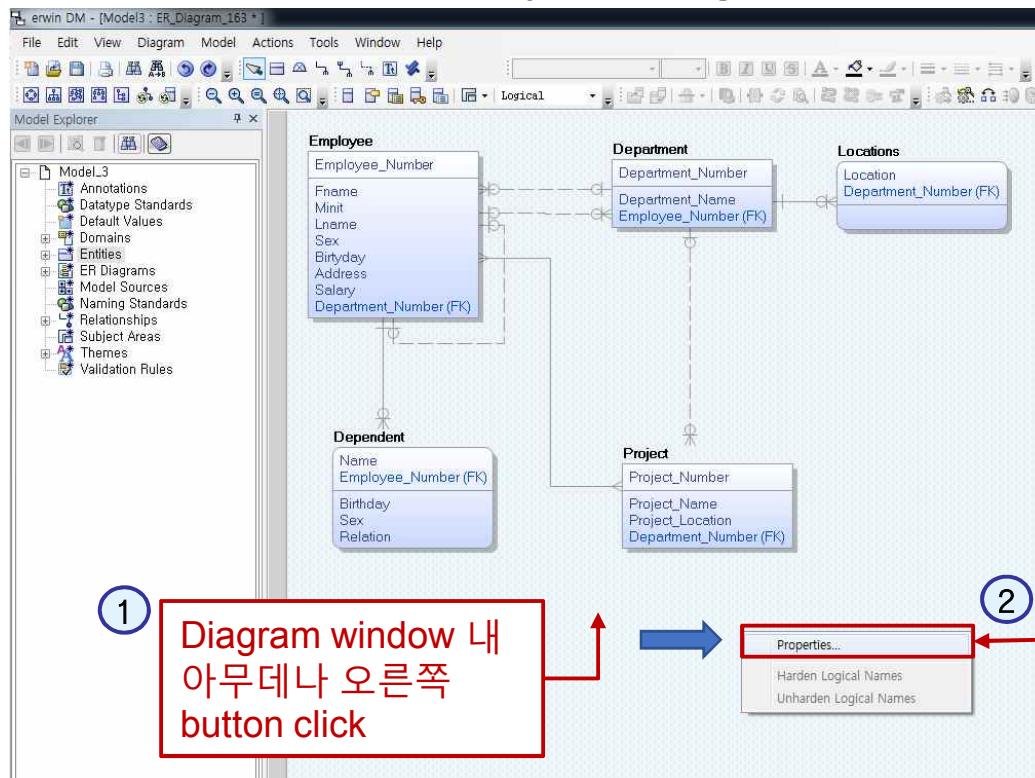
The dialog box also includes a 'Close' button and a 'Cancel' button.

Step 4: Relationship에 이름 할당 (cont'd.)



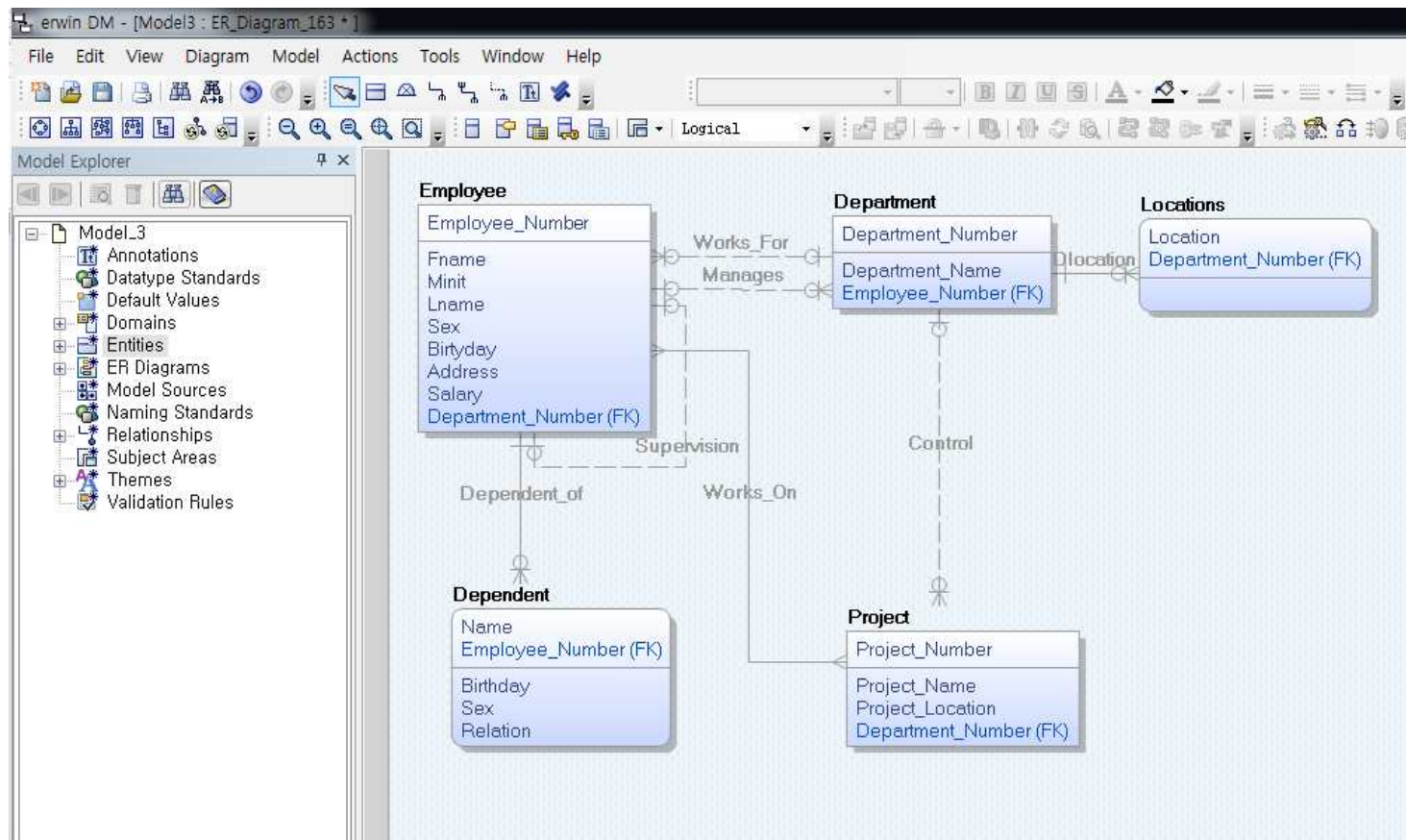
Step 4: Relationship 에 이름 할당 (cont'd.)

- Relationship 이름을 diagram에 표시하기
 - Model property dialog box 이용



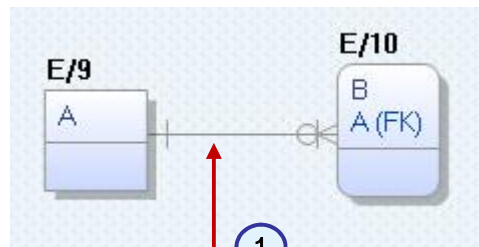
Step 4: Relationship에 이름 할당 (cont'd.)

- Relationship 이름 표시 결과

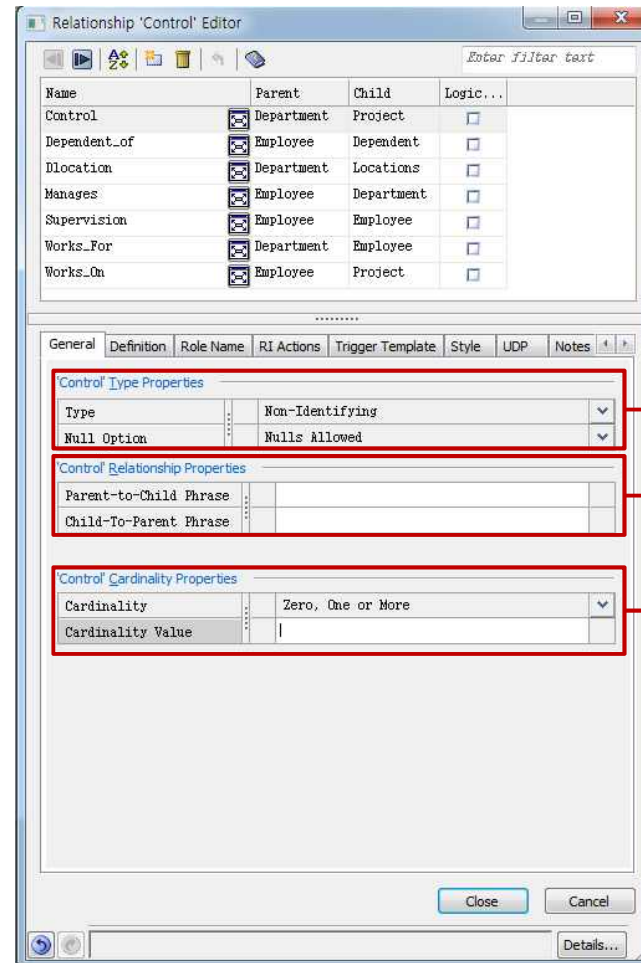
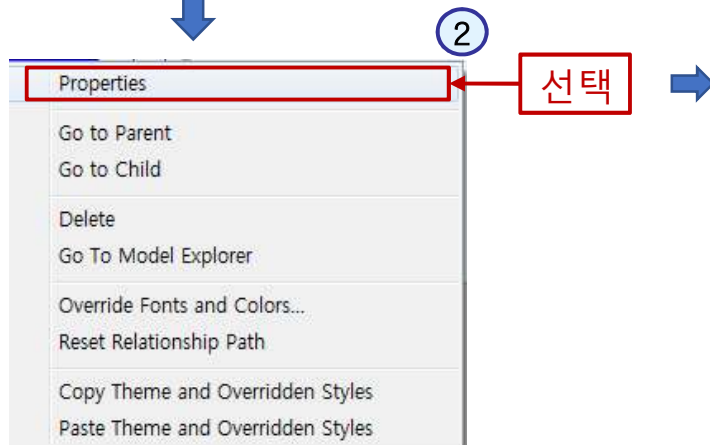


Step 5: Relationship 상세사항 설정

- 방법: Relationship property dialog box 이용



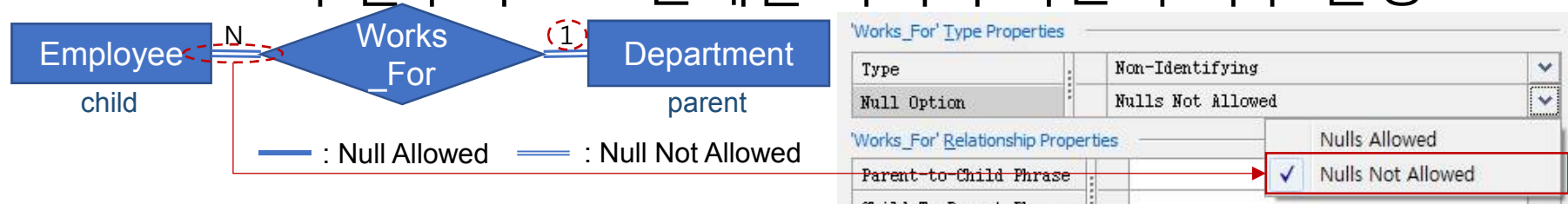
① 오른쪽 버튼 Click.



Step 5: Relationship 상세사항 설정 (cont'd.)

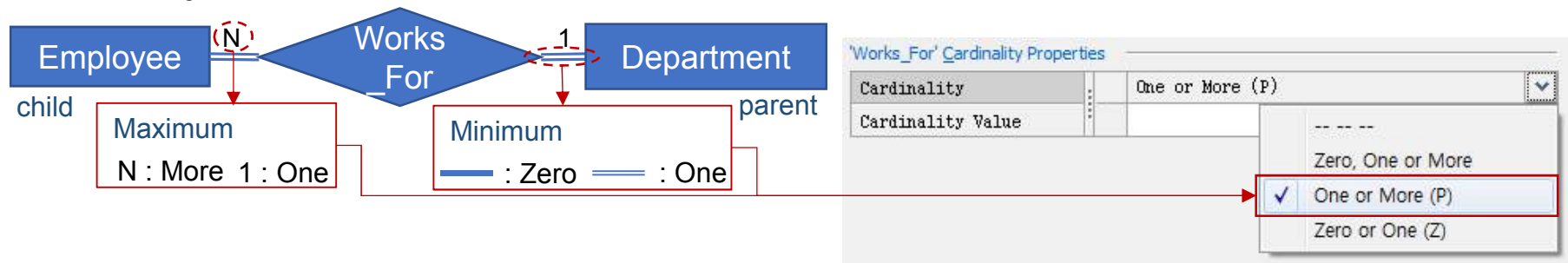
- **Optionality(Null option) : 관계참여 필수 여부**

- Child entity가 갖는 하나의 instance가 parent entity가 가진 instance와 필수적으로 관계를 가져야 하는지 여부 설정



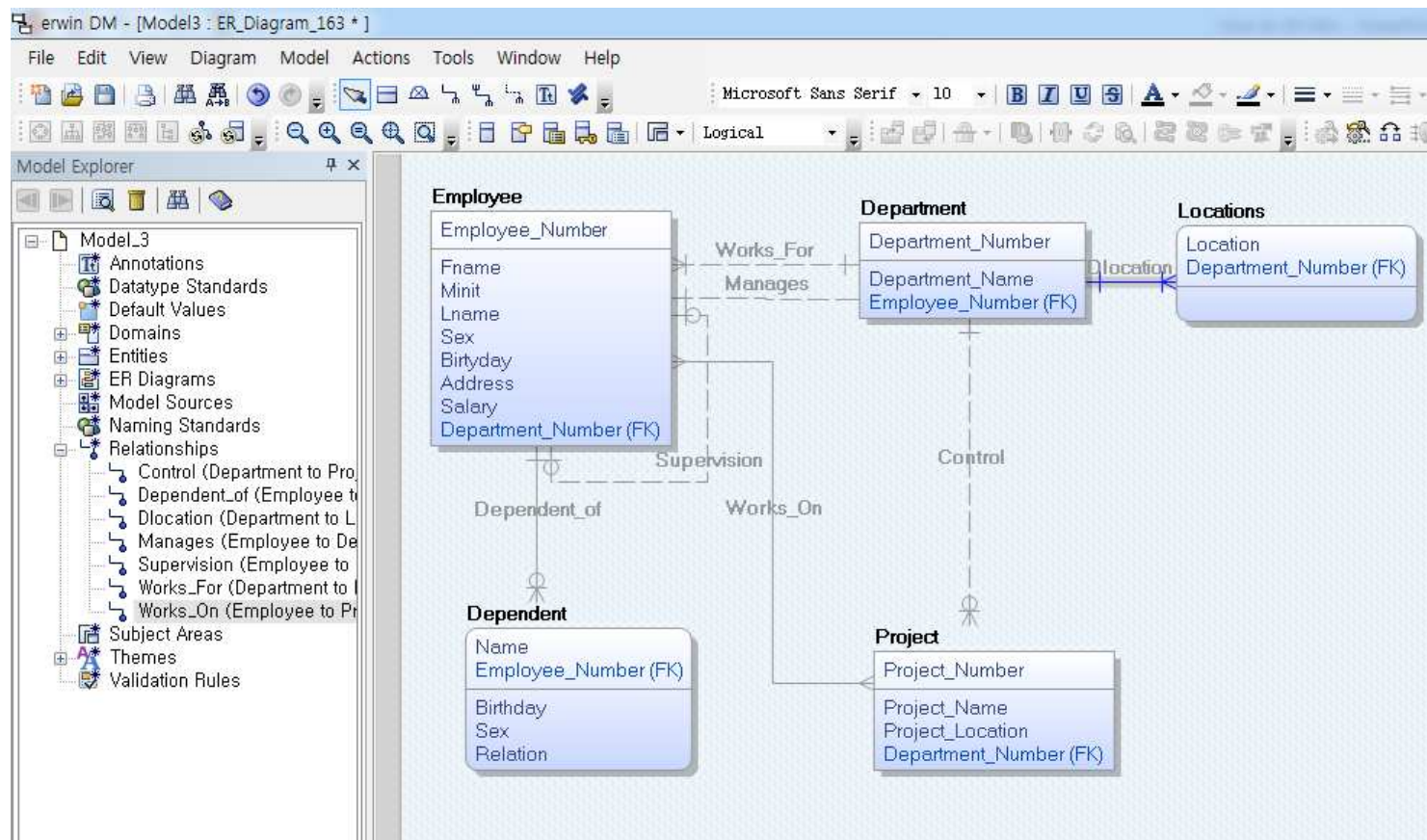
- **Mapping cardinality : 관계 차수**

- Parent entity가 갖는 하나의 instance가 최대 몇 개의 child entity가 갖는 instance들과 관계를 가질 수 있는지 설정



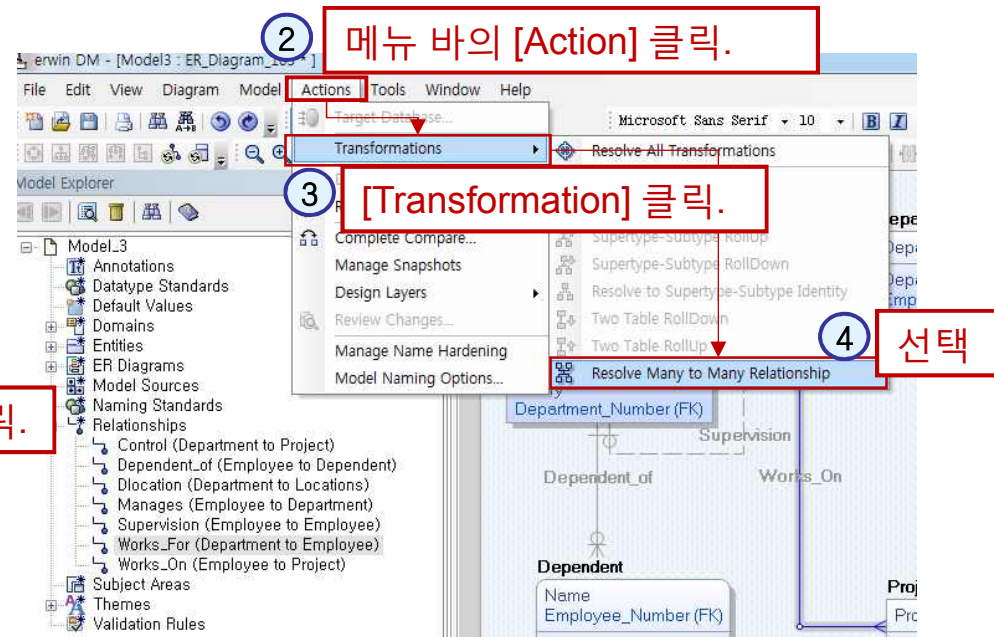
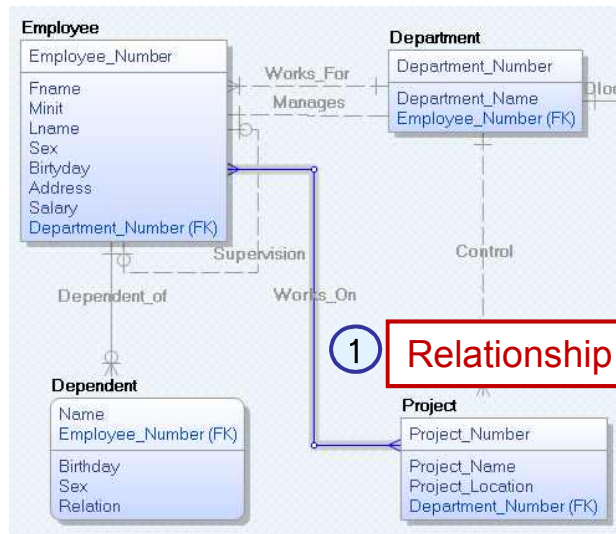
Step 5: Relationship 상세사항 설정 (cont'd.)

- 관계 설정 결과 (p41의 개념적 다이어그램과 비교)



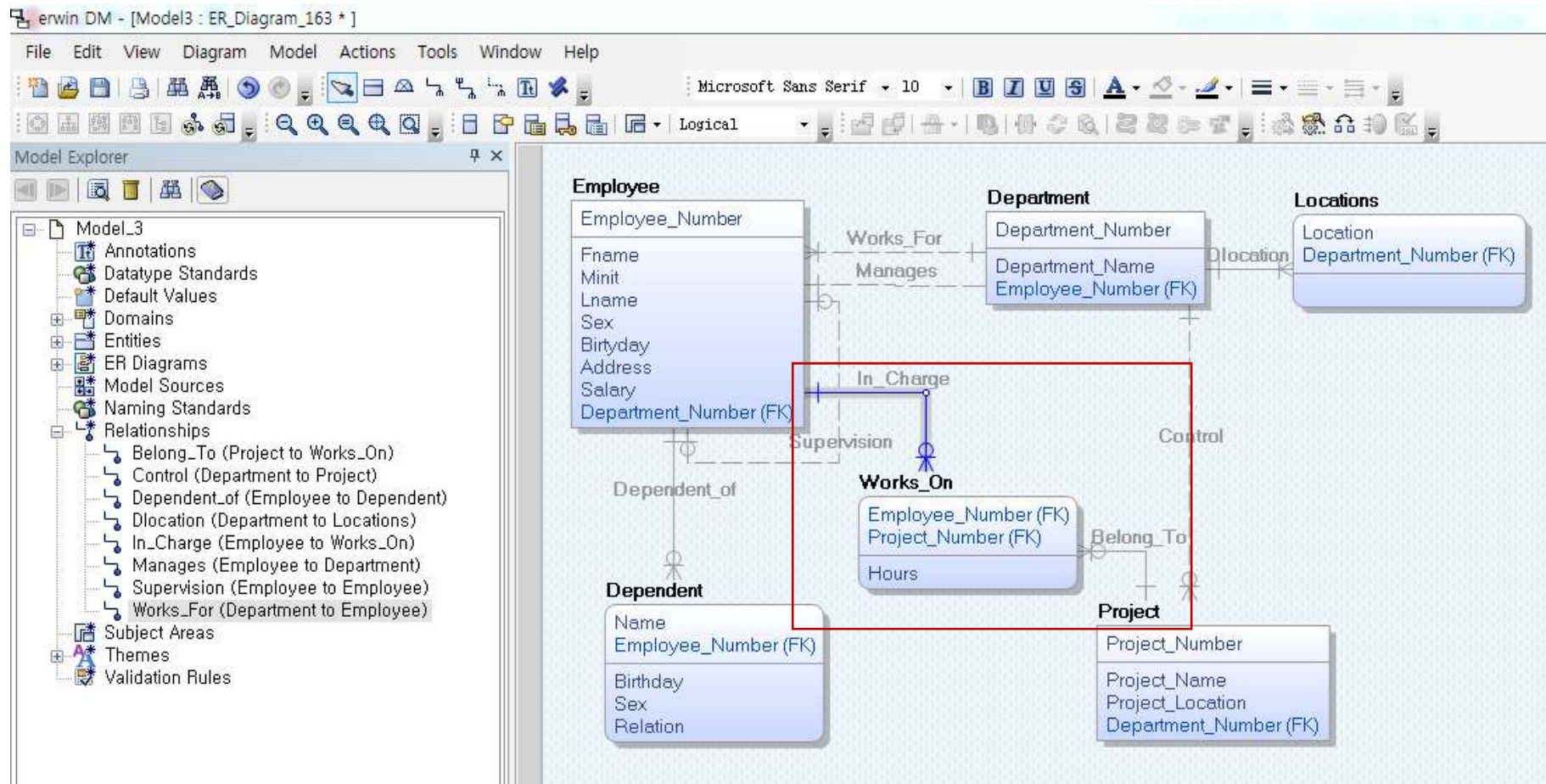
Step 5: Relationship 상세사항 설정 (cont'd.)

- M:N relationship 해소하기  M:N relationship
 - 방법: Transformation



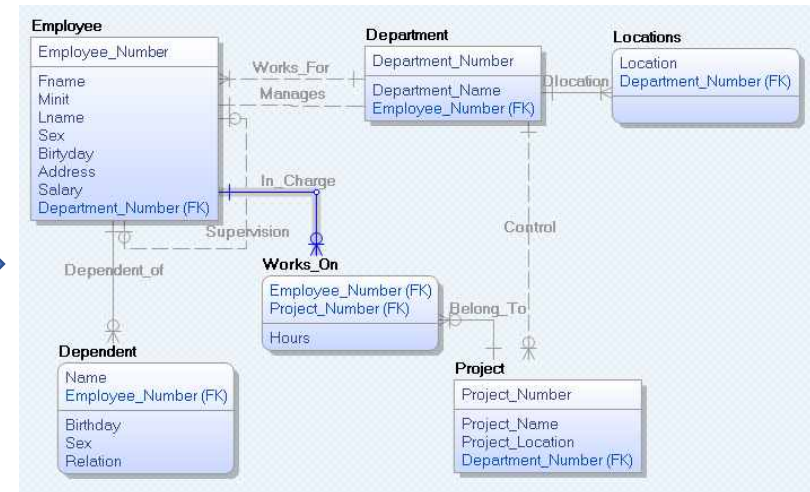
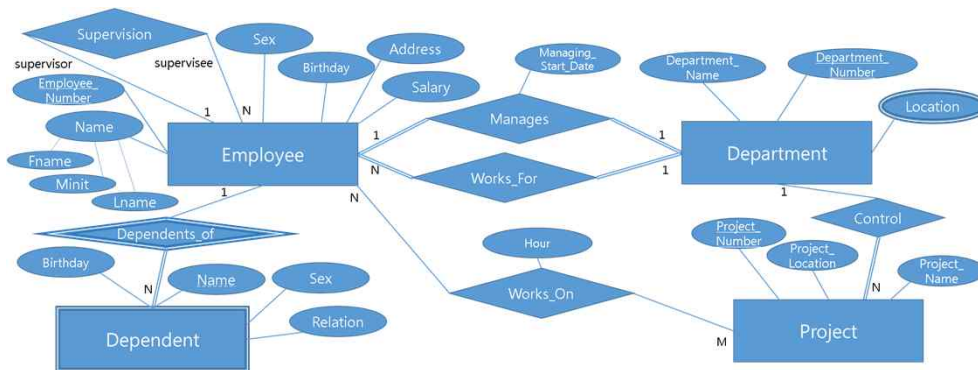
Step 5: Relationship 상세사항 설정 (cont'd.)

- M:N relationship의 transformation 및 이름 설정 결과



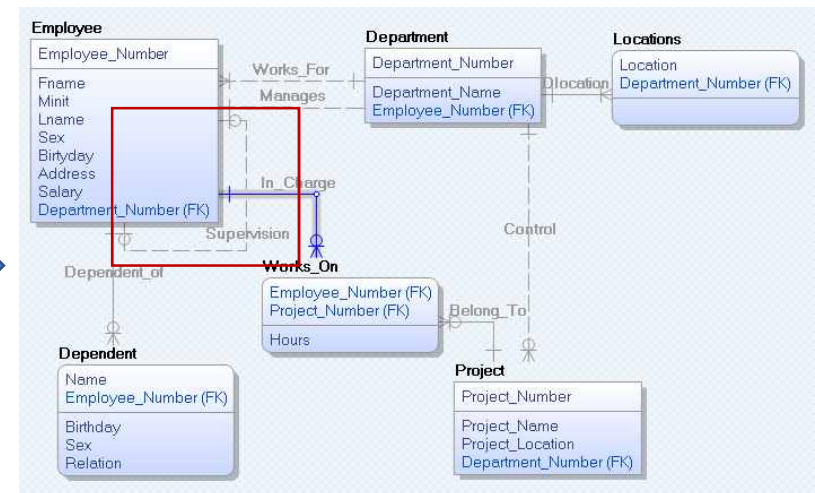
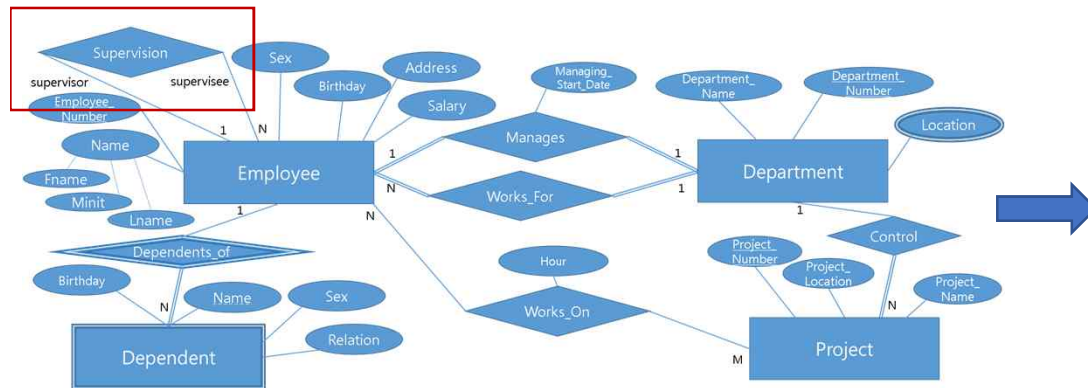
Step 5: Relationship 상세사항 설정 (cont'd.)

- 최종 결과



보충사항

- self-relationship 처리하기



보충사항 (cont'd.)

- **self-relationship**에 의한 attribute 나타내기
 - 방법: relationship property dialog box 이용

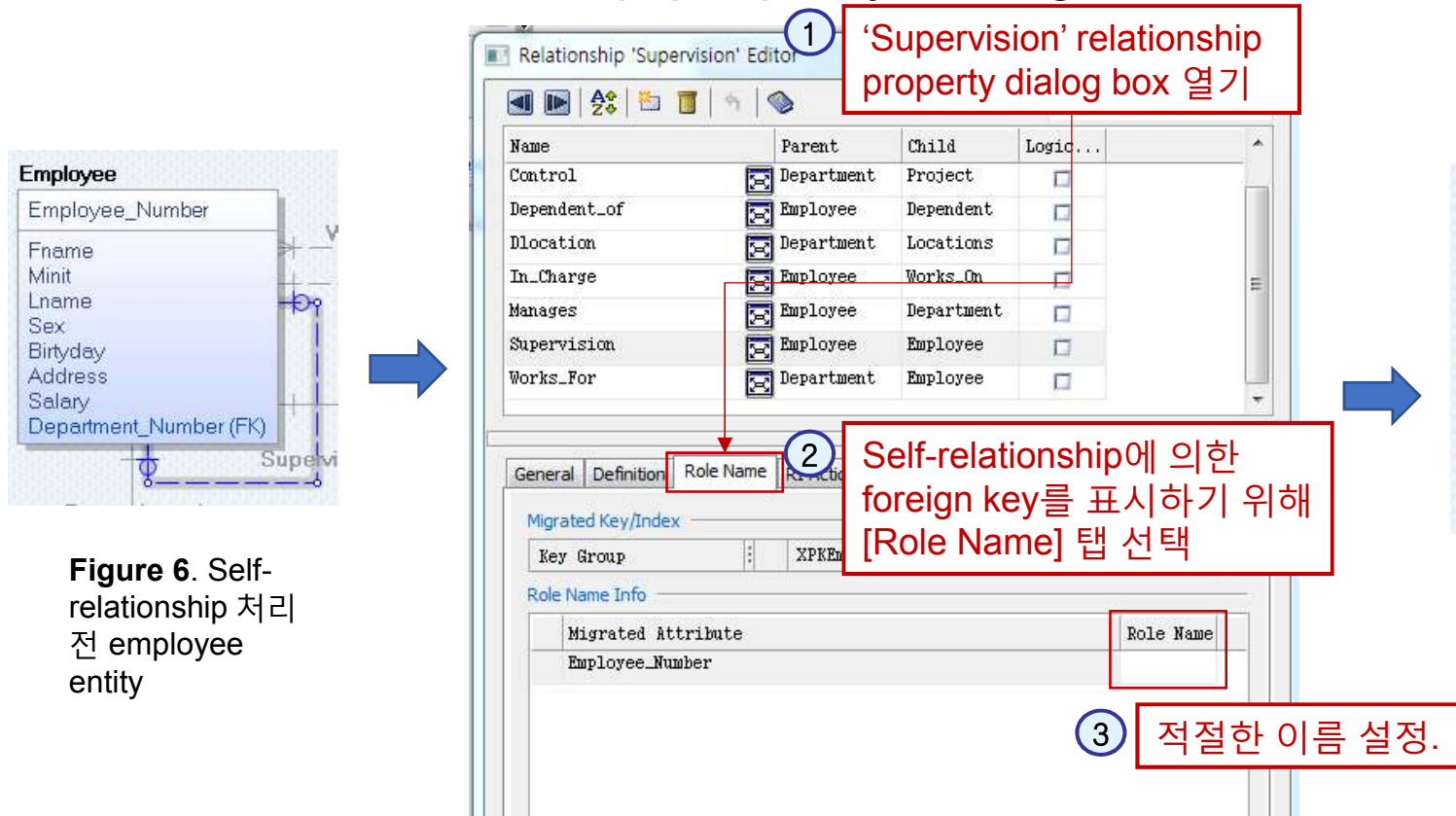


Figure 6. Self-relationship 처리 전 employee entity

Figure 7. [Role Name] 설정을 통한 Self-relationship 처리 후 employee entity

보충사항 (cont'd.)

- **self-relationship**에서 parent와 child의 이름 설정
 - 방법: relationship property dialog box 이용

Figure 6. Self-relationship 처리 전 employee entity

Figure 7. 'Supervision' relationship property dialog box 열기

Figure 8. 역할 이름 설정을 통한 Self-relationship 처리 후 employee entity

Figure 9. [General] tab 선택

Figure 10. 부모

Figure 11. 자식

Figure 12. 적절한 이름 설정.

Name	Parent	Child	Logic...
Control	Department	Project	<input type="checkbox"/>
Dependent_of	Employee	Dependent	<input type="checkbox"/>
Dlocation	Department	Locations	<input type="checkbox"/>
In_Charge	Employee	Works_On	<input type="checkbox"/>
Manages	Employee	Department	<input type="checkbox"/>
Supervision	Employee	Employee	<input type="checkbox"/>
Works_For	Department	Employee	<input type="checkbox"/>

Supervision Type Properties

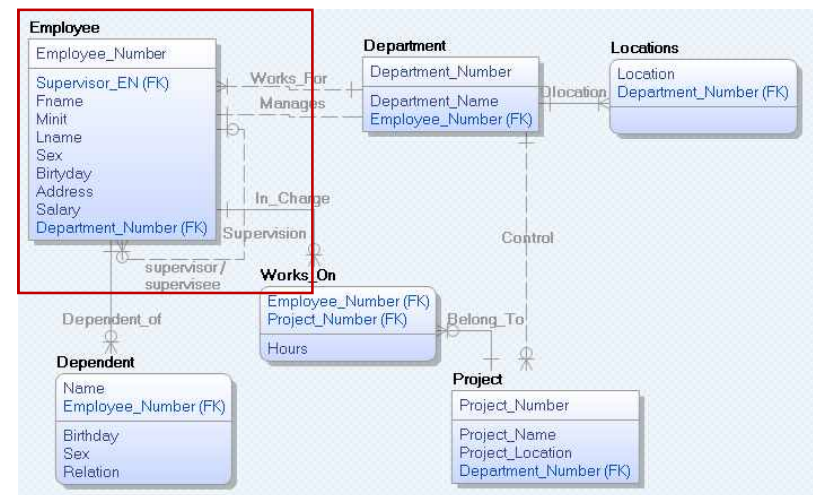
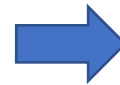
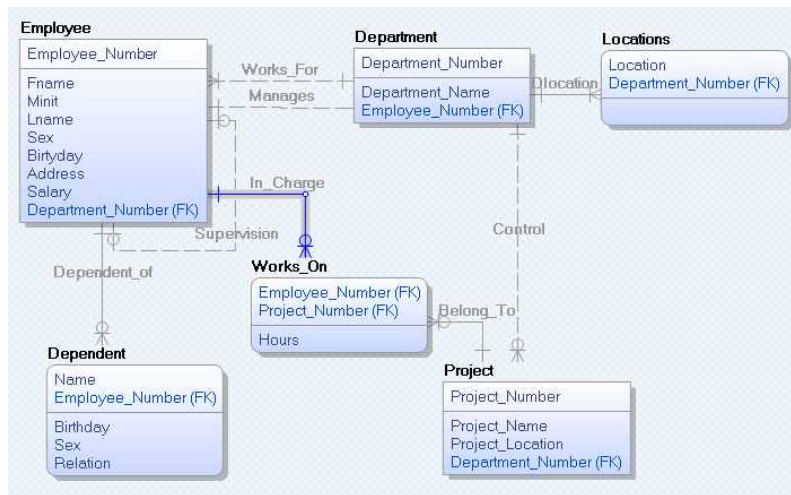
Type	Non-Identifying
Null Option	Nulls Allowed

Supervision Relationship Properties

Parent-to-Child Phrase	supervisor
Child-to-Parent Phrase	supervisee

보충사항 (cont'd.)

- self-relationship 처리
 - 결과



Phase 5:Physical design

- Database를 위한 특정 파일 저장 구조나 적절한 접근 경로를 선택하면 성능 향상 가능.
- 물리적 database design시 고려사항:
 - Response time
 - Space utilization
 - Transaction throughput

Physical design을 위한 보충사항

- ER-Win의 [Physical] modeling mode
 - 이 모드는 모델의 물리적 특성을 target database에 적합한 방법으로 디자인하도록 기초 설정 지원
 - 방법



Click & [Physical]로 mode 전환.

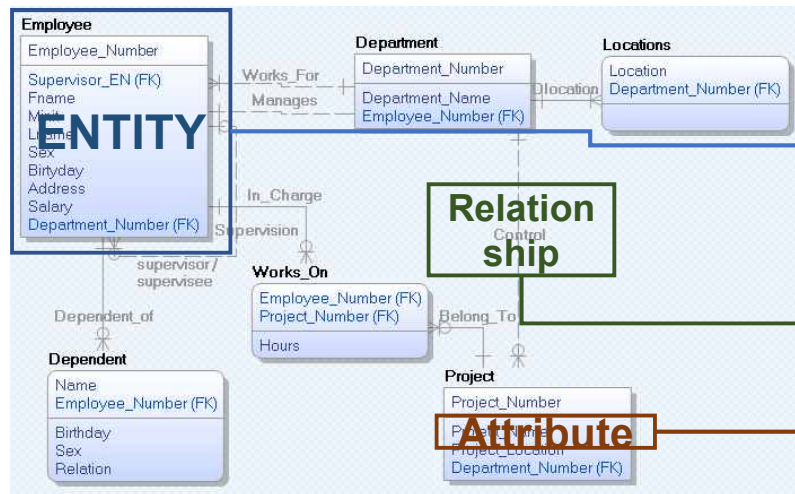


Figure . Logical diagram of company DB

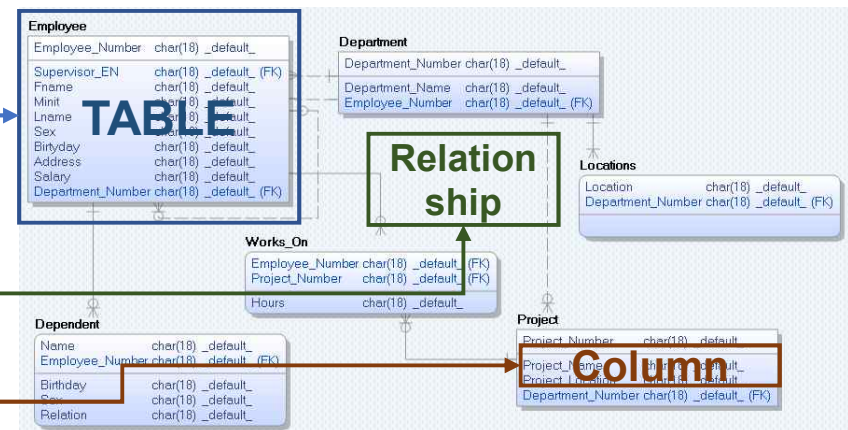


Figure . Physical diagram of company DB

보충사항 (cont'd.)

- [Physical] modeling mode
 - Column의 domain 이나 datatype 설정하기
 - 방법: column editor 사용

Employee

Column Name	Data Type
Employee_Number	char(18) _default_
Supervisor_EN	char(18) _default_ (FK)
Fname	char(18) _default_
Minit	char(18) _default_
Lname	char(18) _default_
Sex	char(18) _default_
Birthday	char(18) _default_
Address	char(18) _default_
Salary	char(18) _default_
Department_Number	char(18) _default_ (FK)

Department

Column Name	Data Type
Department_Number	char(18) _default_

Works_On

Column Name	Data Type
Employee_Number	char(18) _default_ (FK)
Project_Number	char(18) _default_ (FK)
Hours	char(18) _default_

Dependent

Column Name	Data Type
Name	char(18) _default_
Employee_Number	char(18) _default_ (FK)
Birthday	char(18) _default_
Sex	char(18) _default_
Relation	char(18) _default_

Project

Column Name	Data Type
Project_Number	char(18) _default_
Project_Name	char(18) _default_
Project_Location	char(18) _default_
Department_Number	char(18) _default_ (FK)

SQL Server Table 'Employee' Column 'Fname' Editor

Physical Name	Domain Parent	Physical Data Type	Primary Key	Physical Only
Employee_Number	default_	char(18)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Supervisor_EN	default_	char(18)	<input type="checkbox"/>	<input type="checkbox"/>
Fname	default_	char(18)	<input type="checkbox"/>	<input type="checkbox"/>
Minit	default_	char(18)	<input type="checkbox"/>	<input type="checkbox"/>

Domain Parent

- default_
- Blob
- Datetime
- Number
- String

Physical Name

Physical Data Type

Null Option

2-1 Domain 설정

2-2 Datatype 설정

보충사항 (cont'd.)

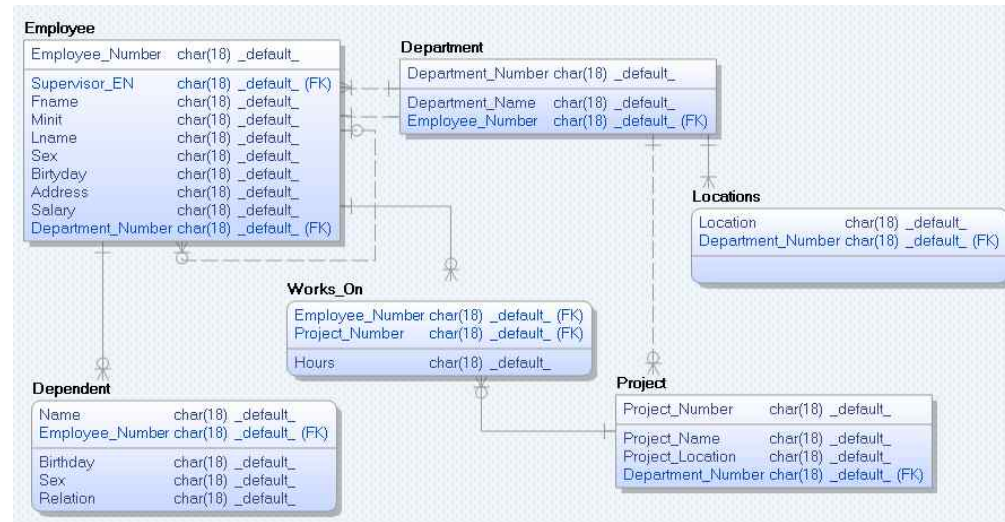


Figure .
Domain and datatype 설정 전 Physical diagram

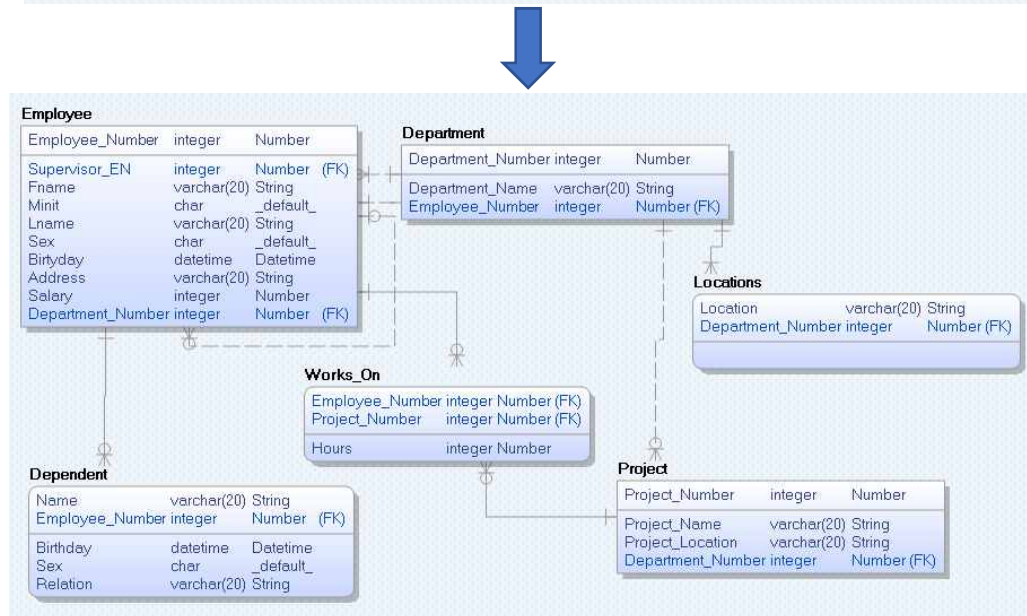



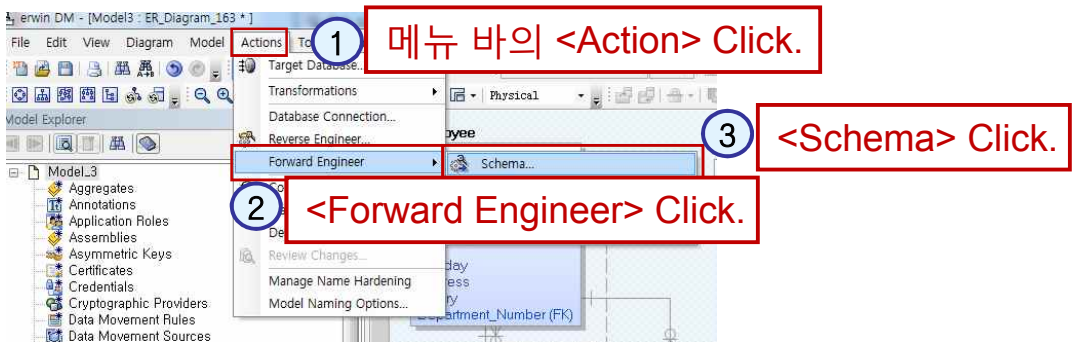
Figure .
Domain and datatype 설정 후 Physical diagram

보충사항 (cont'd.)

3. Forward engineering

- High-level model로 부터 low-level model의 특성, 상세사항 등을 만들어 내기 위한 과정
- ER-Win도 이 기능을 가지고 있으며 이를 통해 논리적 데이터 모델을 물리적 데이터 모델로 변환하는 것을 도와줌
- 방법

1.  [Physical] mode로 전환.

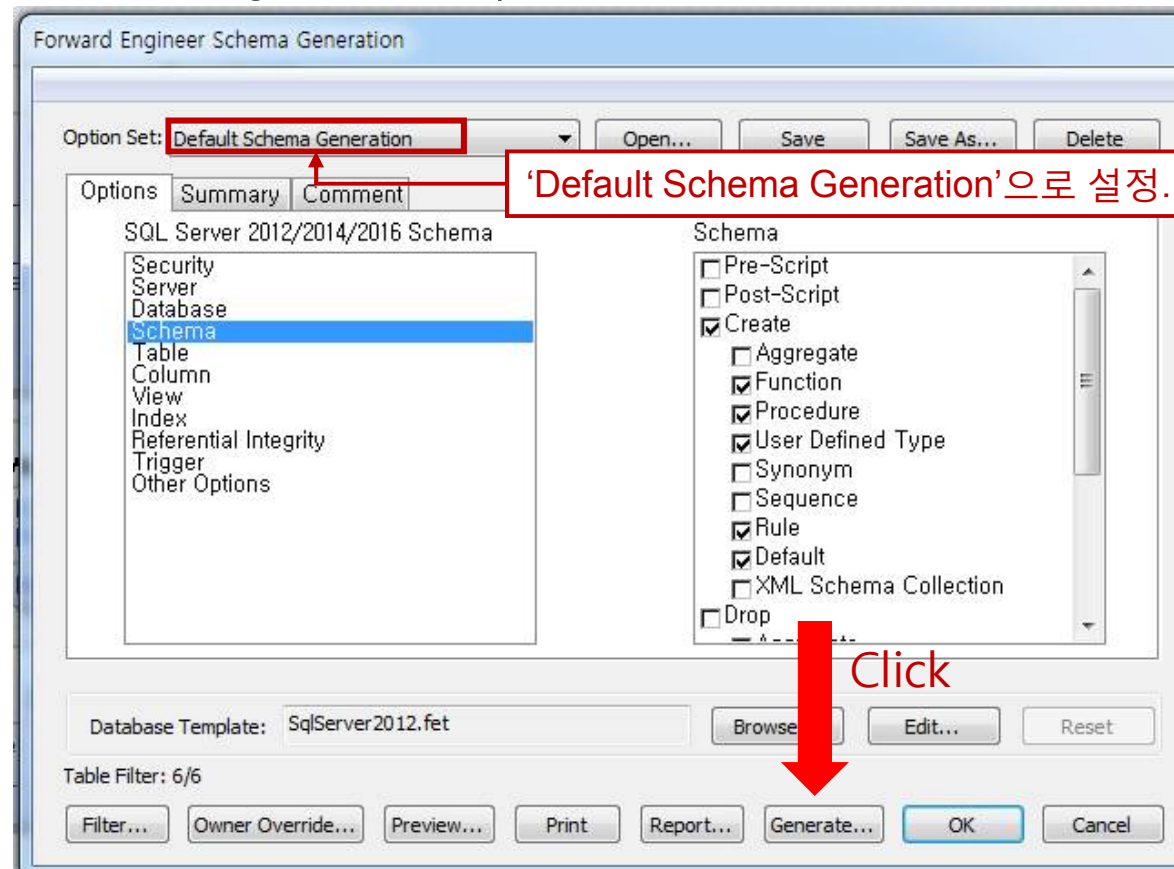
2.  ① 메뉴 바의 <Action> Click.
② <Forward Engineer> Click.
③ <Schema> Click.

보충사항 (cont'd.)

3. Forward engineering

– 방법

3. (DB schema generation)

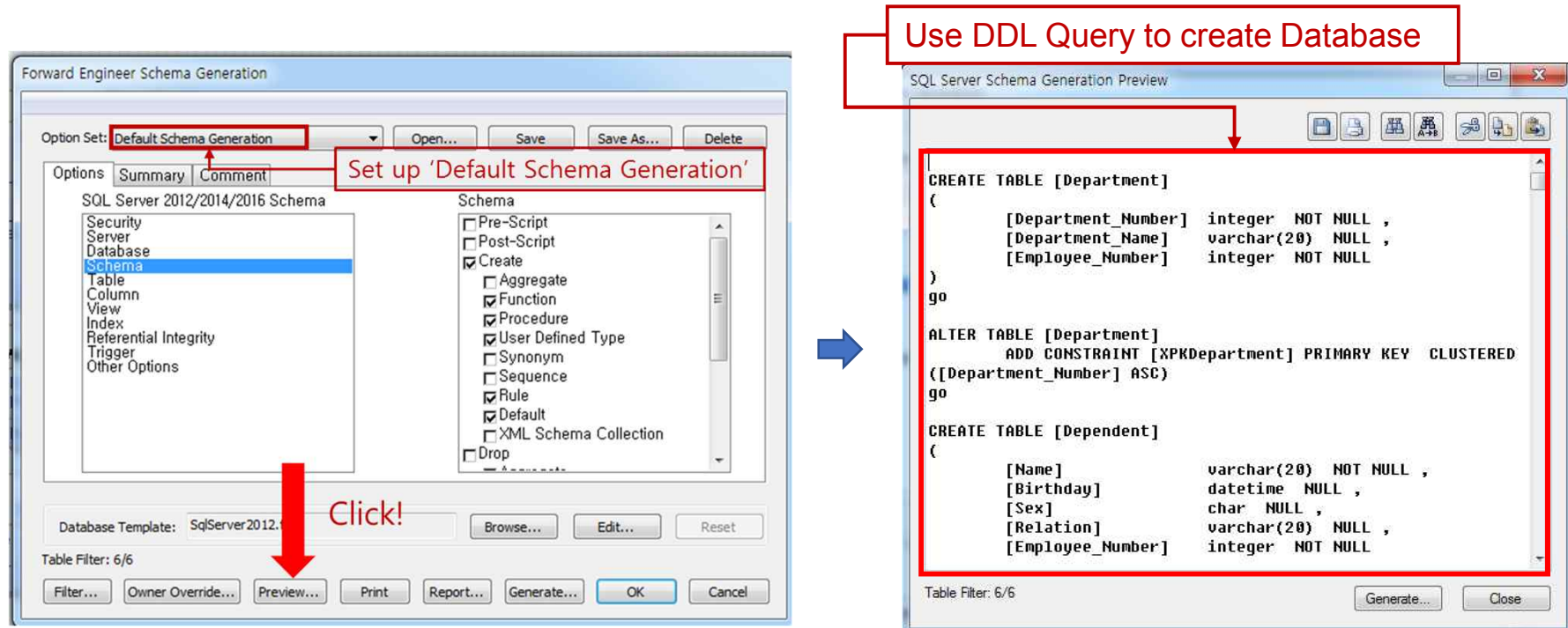


보충사항 (cont'd.)

3. Forward engineering

방방

3. (DDL generation for DB schema)



보충사항 (*cont'd.*)

3. Forward engineering

– 방법

4. 생성된 DDL 예시

```
CREATE TABLE DEPARTMENT  
  
(  
  
    Dnumber            int  NOT NULL ,  
  
    Dname              varchar(20)  NULL ,  
  
    Mgr_start_date     datetime  NULL ,  
  
    Mgr_ssn            varchar(10)  NOT NULL  
  
)  
  
go
```

➤ SQL에 대한 자세한 내용은 SQL 실습 시간에...



THANK YOU



인하대학교