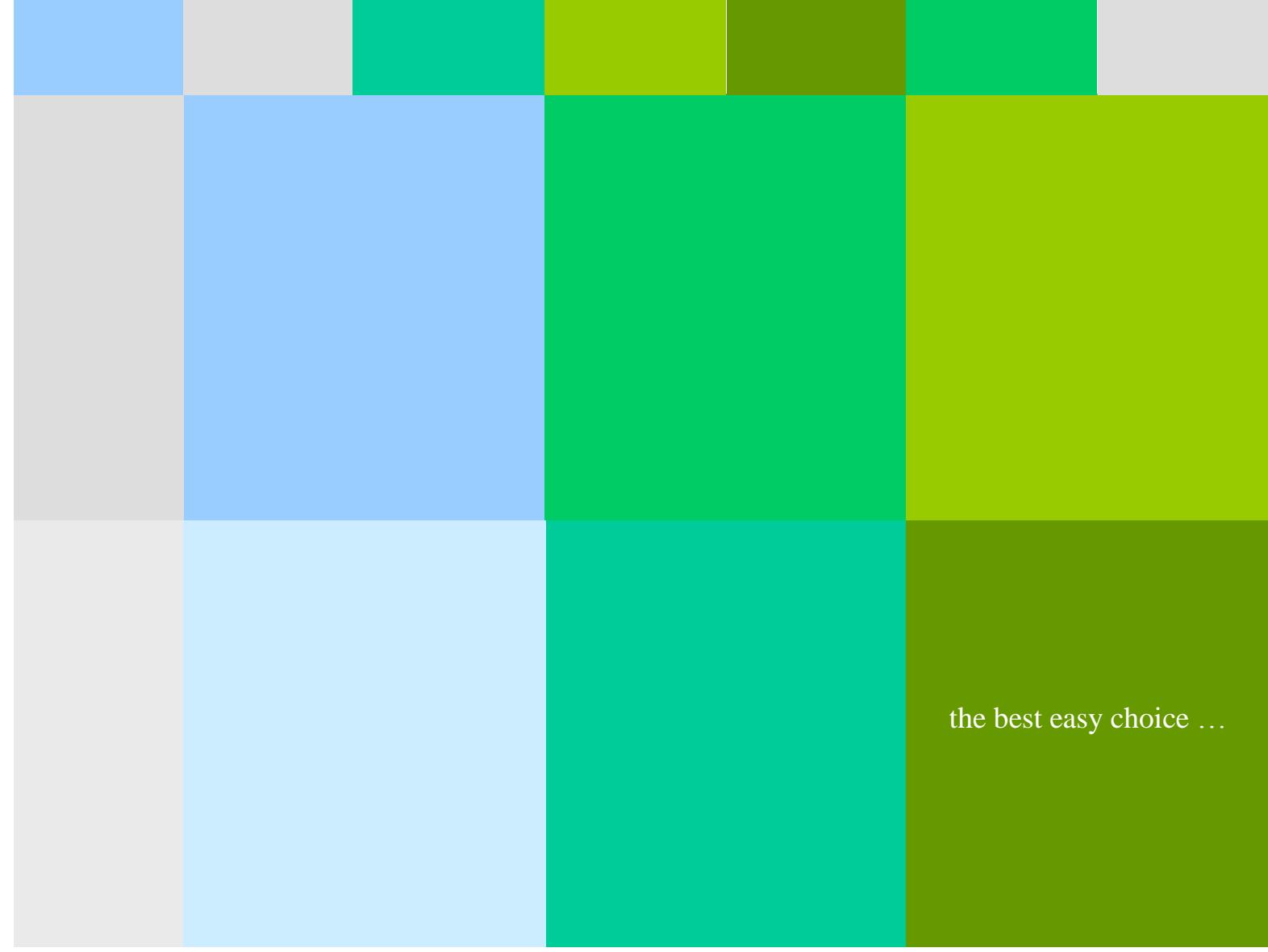


Allfusion ERwin Data Modeler r7

Tool Guide



the best easy choice ...

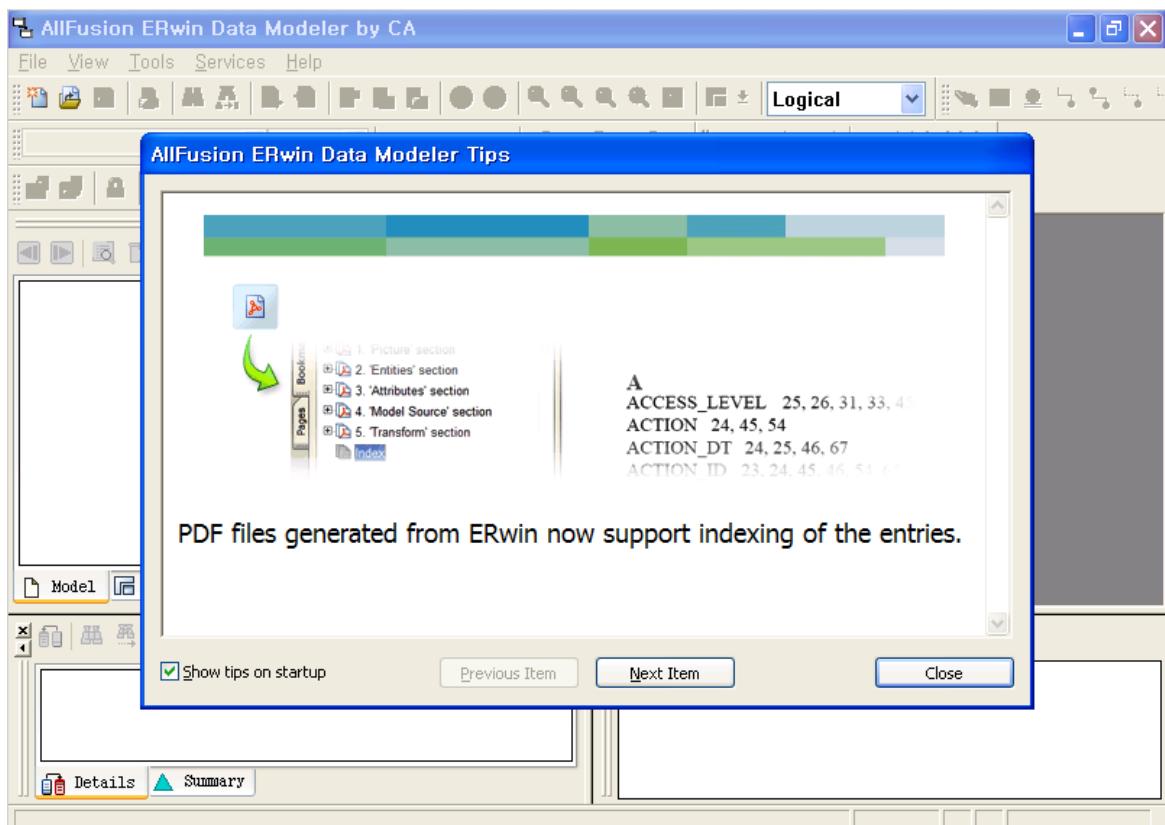
1. 문서 소개 및 작업 환경	3
1.1 AllFusion ERwin Data Modeler r7 실행	3
1.2 작업 창 설명	4
1.3 지원되는 표기 방식	5
1.4 Tool Box 설명	5
1.5 새로운 모델 생성	6
1.6 ERwin Diagram 저장	6
2. Logical Modeling	14
2.1 Entity 생성	14
2.2 Attribute 생성	16
2.3 Relationship 생성	20
2.4 Subject Area (주제영역).....	29
2.5 Stored Displays	31
3. Physical Modeling	34
3.1 Physical / Logical Model의 표현.....	34
3.2 Table 관리	36
3.3 Domain 관리.....	37
3.4 Column 관리.....	40
3.5 View Table 관리	45
3.6 모델의 표준화	47
4. Transform	52
4.1 Transform 정의	52
4.2 Transform 기본 원리	56
5. Forward Engineer / Reverse Engineer / Complete Compare	79
5.1 Forward Engineer	79
5.2 Reverse Engineer.....	82
5.3 Complete Compare.....	85

1. 문서 소개 및 작업 환경

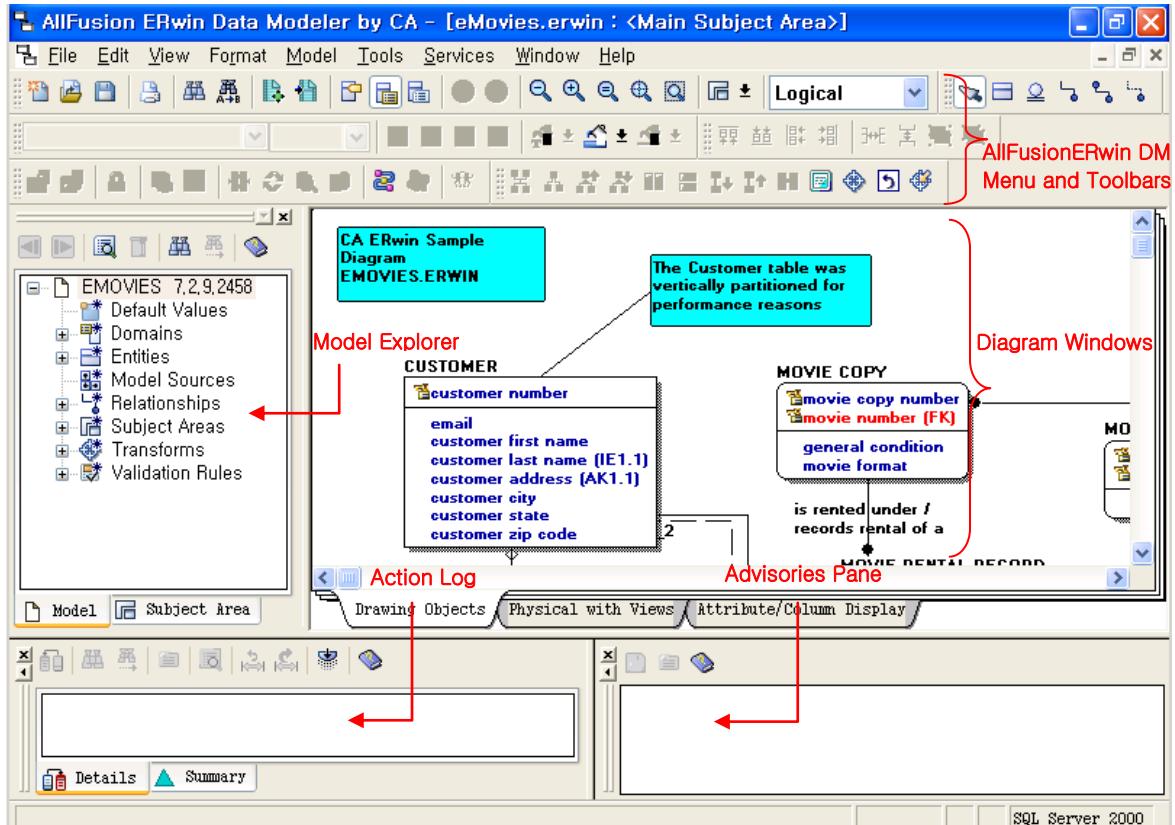
이 문서는 ERwin의 매뉴얼을 기반으로 하여 모델링에 과정에 대하여 다루고 있다. 표기방식은 정보 공학 표기 (IE; Information Engineering)로 설명되며 자세한 ERwin의 기능 설명은 물론, 모델링 작업을 하면서 꼭 필요한 기술 팀을 가이드 하도록 하겠다. 하지만 자세한 ERwin 사용 방법은 포함되어 있지 않으므로 매뉴얼을 찾고 하도록 한다.

1.1 AllFusion ERwin Data Modeler r7 실행

시작 > 프로그램 > CA > Allfusion > ERwin Data Modeler r7에서 ERwin Data Modeler r7 을 선택하면 ERwin 이 실행한다.



1.2 작업 창 설명



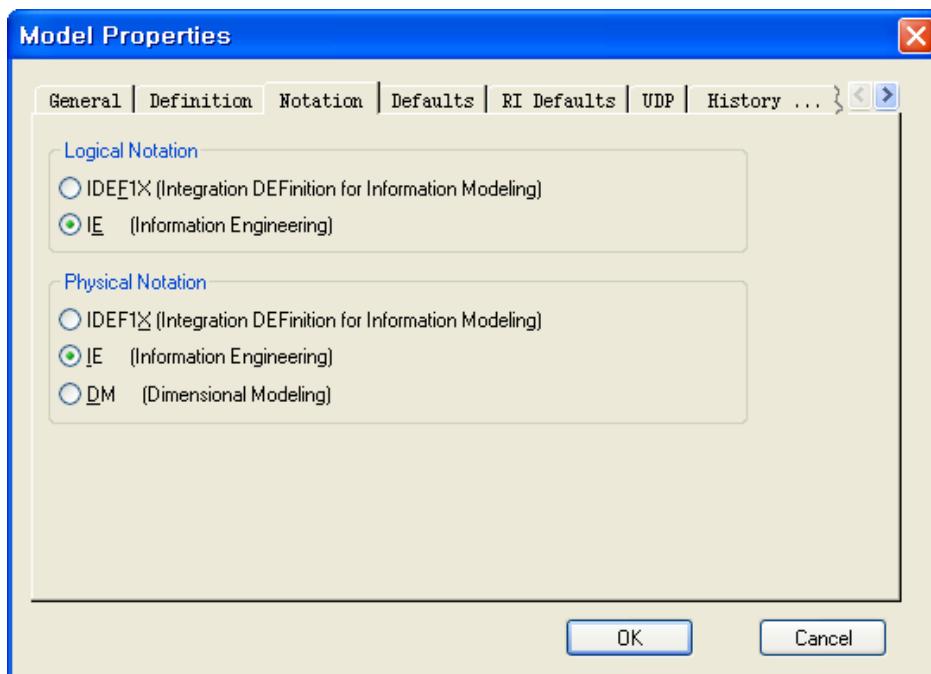
항 목	기 능 설 명
<input type="checkbox"/> Toolbars	모델링 작업을 빠르게 진행할 수 있도록 자주 사용하는 기능이 있으며 선택 여부 및 위치 이동으로 관리할 수 있다.
<input type="checkbox"/> Advisories Pane	사용자가 현재 진행한 작업 정보를 보여준다.
<input type="checkbox"/> Model Explorer	모델내의 탐색기 역할을 하며, 각 Object는 Hierarchy구조로 Drag and Drop 방식으로 추가할 수 있다.
<input type="checkbox"/> Action Log	실시간으로 사용자가 작업한 내용이 기록되며, 선택 지점에서 Undo/Redo 및 반전(reverse) 작업을 할 수 있다.
<input type="checkbox"/> Diagram Windows	이곳에 사용자는 Graphical 하게 모델링 작업을 진행 할 수 있다. 모델을 생성하면 기본적으로 창 이름이 Display1으로 설정되며 Stored Display를 통해 동일 모델을 다양한 관점의 View로 추가할 수 있다.

1.3 지원되는 표기 방식

ERwin에서 지원되는 표기방식은 두 가지이다. 첫 번째로 정보공학 표기 방식으로서 모델링 할 때 일반적으로 사용되는 IE(Information Engineering) 방식이고 두 번째로는 미합중국 공군에서 개발한 프로젝트 표준안인 IDEF1X(Integration DEFinition or Information Modeling) 방식이다.

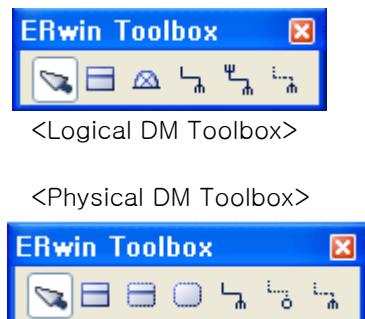
ERwin을 설치 한 뒤, 사용자가 설정을 하지 않는다면 기본적으로 IDEF1X 방식의 표기법으로 보여지게 된다.

우리는 IE 방식으로 모델링 작업을 할 것이므로 변경해보자. 변경하는 방법은 Model 메뉴에서 Model Properties를 클릭한 뒤, Notation 탭에서 IE를 선택하면 된다.



1.4 Tool Box 설명

ERwin Toolbox: 데이터모델을 만들고 편집할 수 있는 기능이 포함되어 있다.



아이콘	Toolbar Name	설명
	Select	모델의 오브젝트 선택
	Entity / Table	엔티티 / 테이블 생성
	Exclusive sub-category	일반화(배타/포함) 관계를 표현
	View Table	뷰, 물리뷰 테이블 생성
	Relationship	관계선을 그릴 때 사용

1.5 새로운 모델 생성

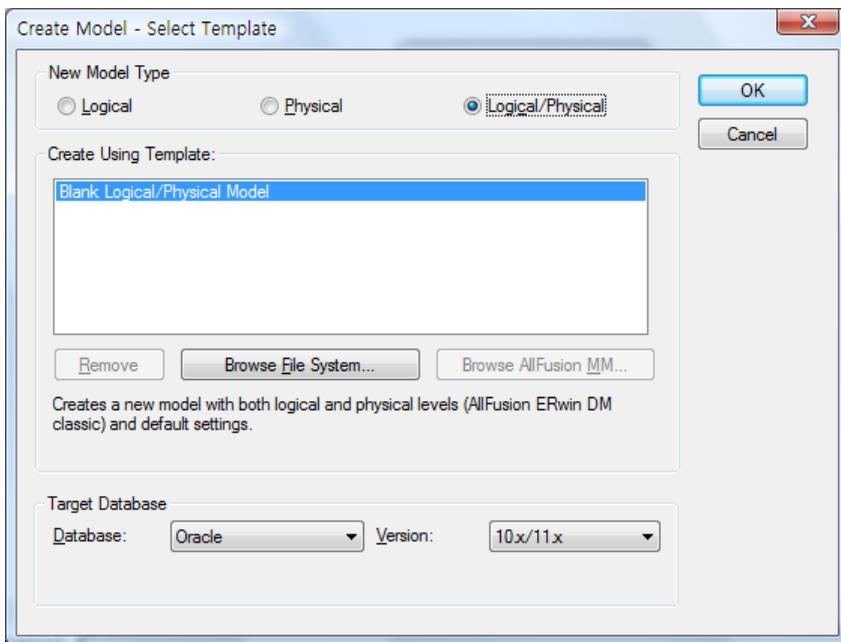
ERwin을 실행하고 File메뉴에서 New버튼 혹은 ‘create model’ 아이콘()을 선택하면 새로운 모델을 생성할 수 있으며 나타나는 화면은 다음과 같다.

Logical, Physical, Logical/Physical 세 가지가 있다.

논리 정보만으로 구성되는 모델 생성하려고 할 때에는 Logical 타입을 선택하고, 물리 정보만으로 구성되는 모델을 생성하려고 할 때에는 Physical 타입을 선택하면 된다. 우리는 ERwin 모델의 전형적인 타입으로, 하나의 모델에 하나의 Logical 타입과 Physical 타입으로 구성된 Logical/Physical 복합 타입을 선택한다.

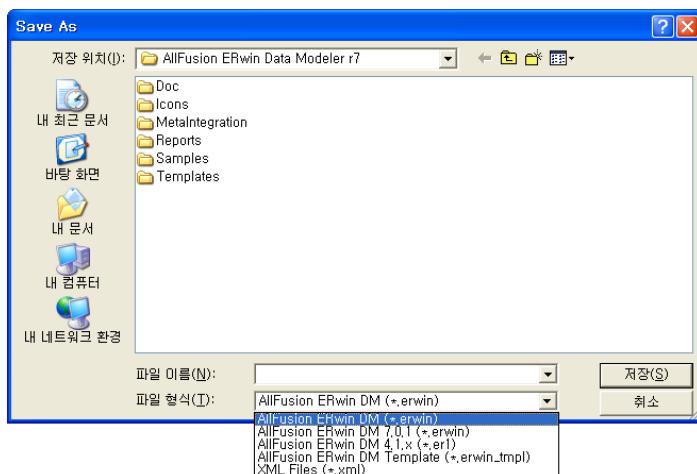
Local에 저장된 Template 모델을 이용하여 모델링 작업을 하려고 한다면 Browse File System 버튼을 클릭하여 새로운 모델을 생성할 수 있다. Browse AllFusion MM버튼을 선택하여 Model Manager에 저장되어 있는 Template 모델을 이용하여 새로운 모델을 생성할 수도 있다.

Target Database를 선택하여 OK 버튼을 클릭한다.



1.6 ERwin Diagram 저장

File > Save 나 Save As를 선택하면 아래의 대화 상자가 나타난다. 디폴트로 *.erwin(4.1이하 버전은 *.er1이다.) 파일 형식이 제공 되나 erwin_tmpl(4.1이하 버전은 ert이다.), xml 형식도 지원한다. ERwin Data Modeler r7에서는 하위버전 ERwin 4.1버전의 파일로 저장과 오픈이 가능하다. 또한 Import 기능을 이용하여 3.x 모델을 오픈할 수 있다.



[Tip] Template 파일 적용

ERwin Template파일을 사용함으로 다른 모델에 ERD를 생성, 설계하는 과정에서 재사용성을 높일 수 있고 일관성 있는 모델을 생성할 수 있다.

1. Template 파일의 효용성

1) ERD생성이 수행되기 전 Template파일을 이용하여 모델 표준 관리

- Model Properties, 도메인, Display Setting, Font & Color Setting, UDP등 ERD모델의 전반적인 사항을 설정하여 Template파일을 생성한 뒤, 해당 Template을 공유하여 일관성 있는 모델을 설계하는 최적의 방법을 제공한다.

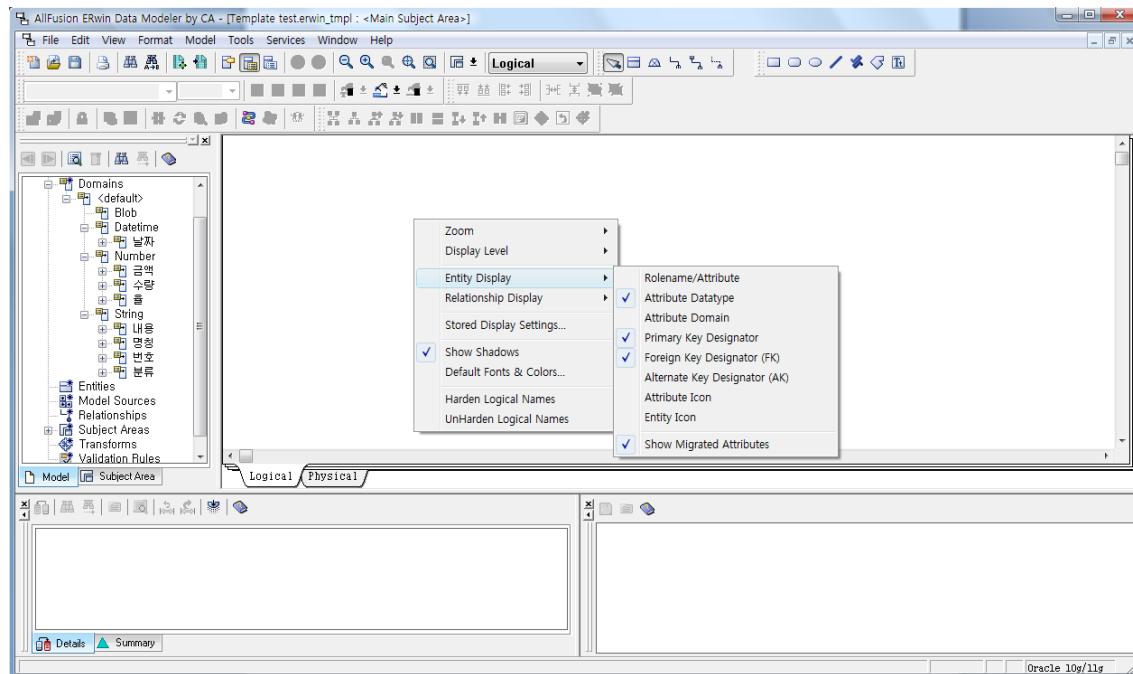
2) ERD생성이 수행된 후 모델 표준 관리

- Template파일을 이용하여, 작성되어 있는 모델을 Add Model Source / Link Model Source, Complete Compare등을 이용하여 Template파일에 작성된 ERD를 가져와야 한다.

3) AS-IS모델을 Reverse 작업을 할 경우에도 저장해둔 Template를 이용하여 표준형식에 맞게 모델을 가져올 수 있다.

2. 작성된 ERD에 Template적용하는 방법

① 적용할 Template파일을 생성하여 저장해 둔다.

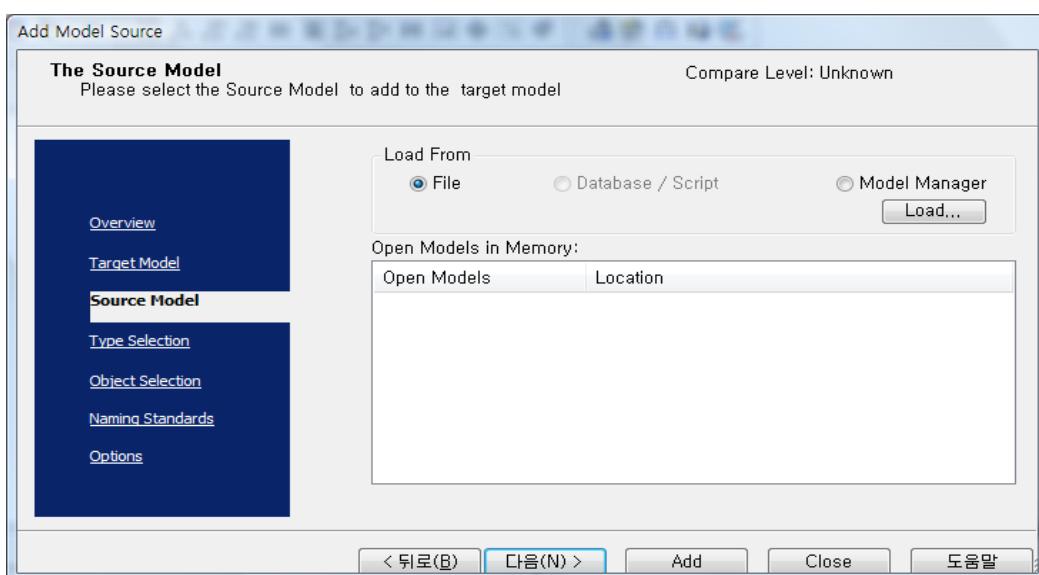


→ Model Properties, Domain정보, Stored Display, Display Option, Font 및 Color등 전반적인 ERD의 Template를 설정하고 저장시에 erwin_tmpl파일로 저장한다.

Domain 정보를 가져올 때에는 기준이 필요하다.

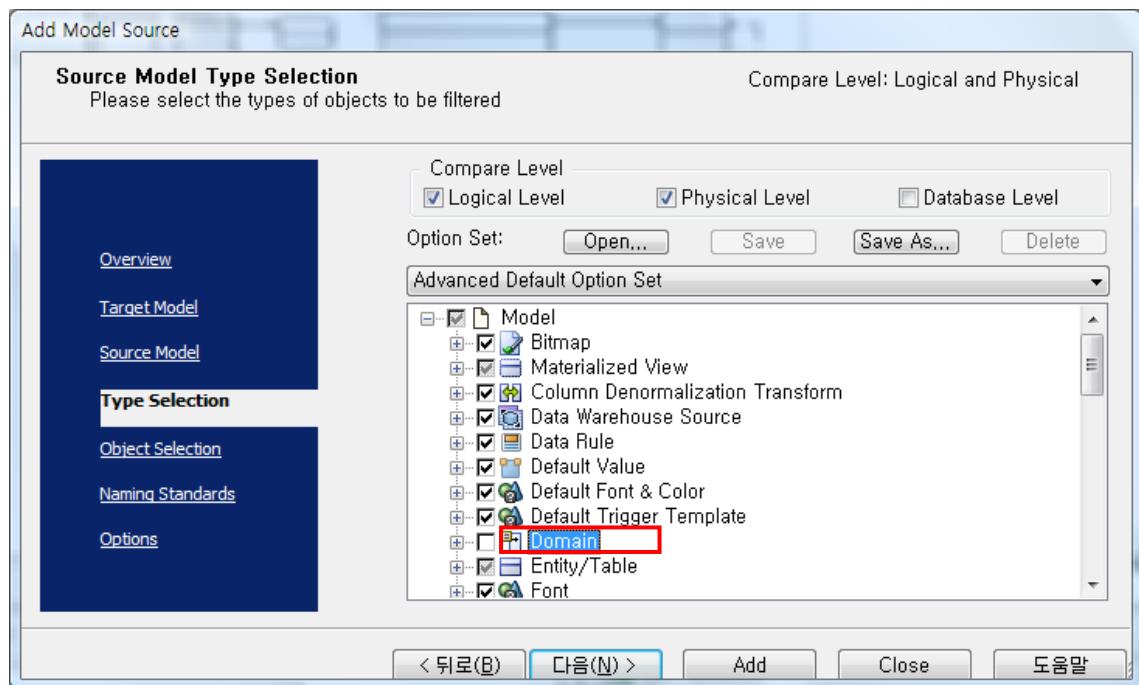
② 작성된 ERD의 도메인을 모델에 가져올 수 있다.

→ 표준에 생성되어 있는 Domain과 작성된 ERD에 있는 Domain의 중복여부를 확인해야 한다. 중복 명칭의 Domain이 있으면 Domain_000 등의 object ID가 붙어서 Domain이 생성된다. 작업 순서는 Tools메뉴의 Add Model Source를 실행시킨다.



Load를 Click하여 가져올 ERD를 선택하고 Add버튼을 클릭한다.

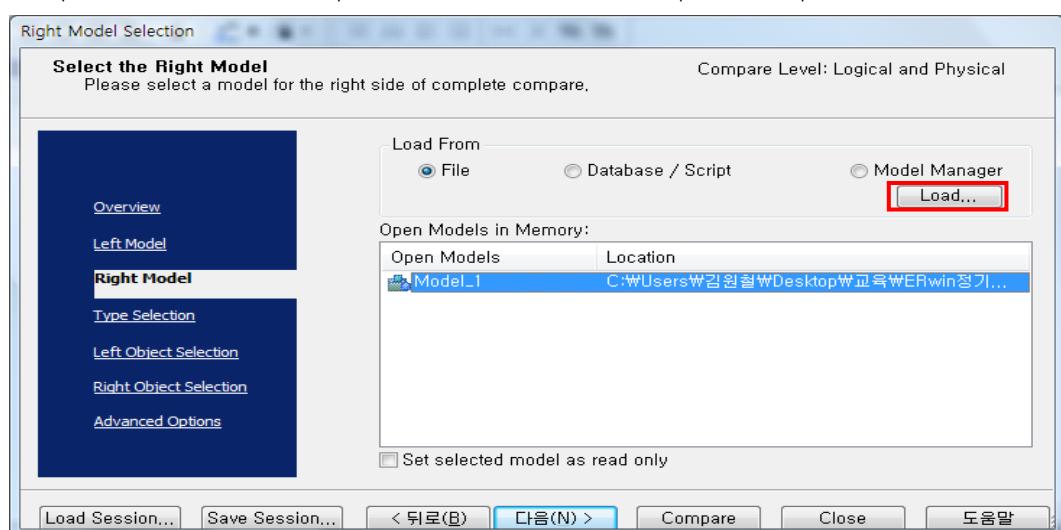
작성된 ERD의 도메인을 모델에 가져오지 않을 경우, 위 스크린샷과 동일하게 Source Model을 Load하여 Model 을 open하는 과정까지는 같다. 단지 다음 단계에서 아래와 같이 Domain부분을 제외시키면 된다.



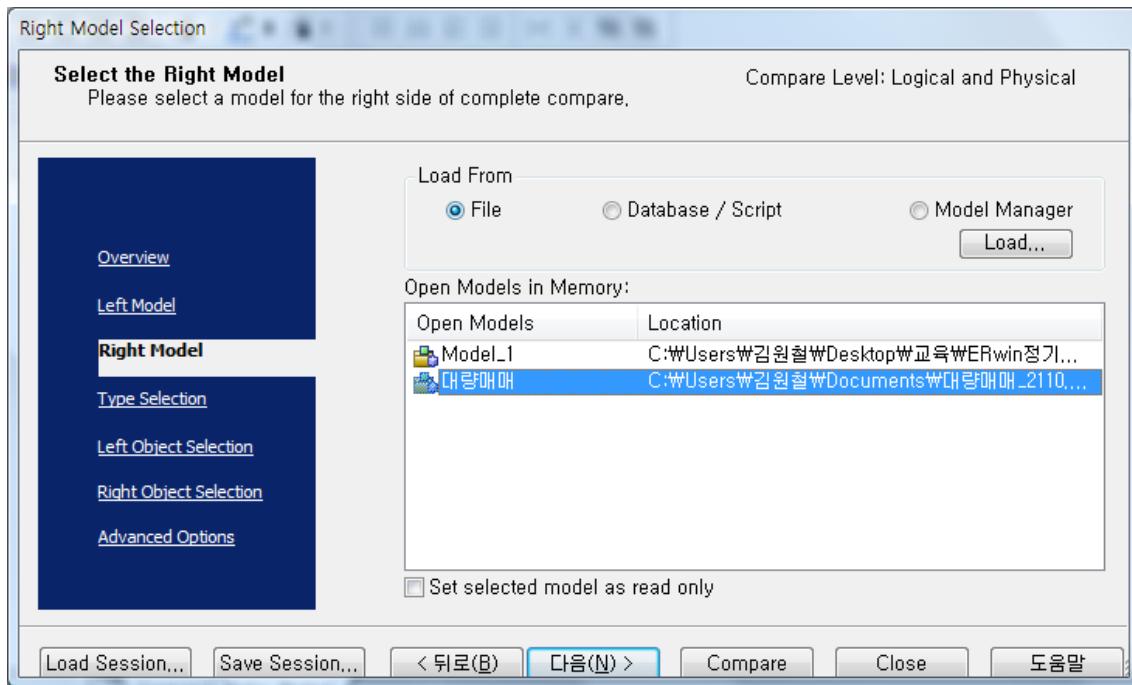
Domain을 제외하고 Add를 실행시키면 Processing후 작업이 완료된다.

③ Complete Compare를 통한 Template 표준 파일에 ERD모델을 통합한다.

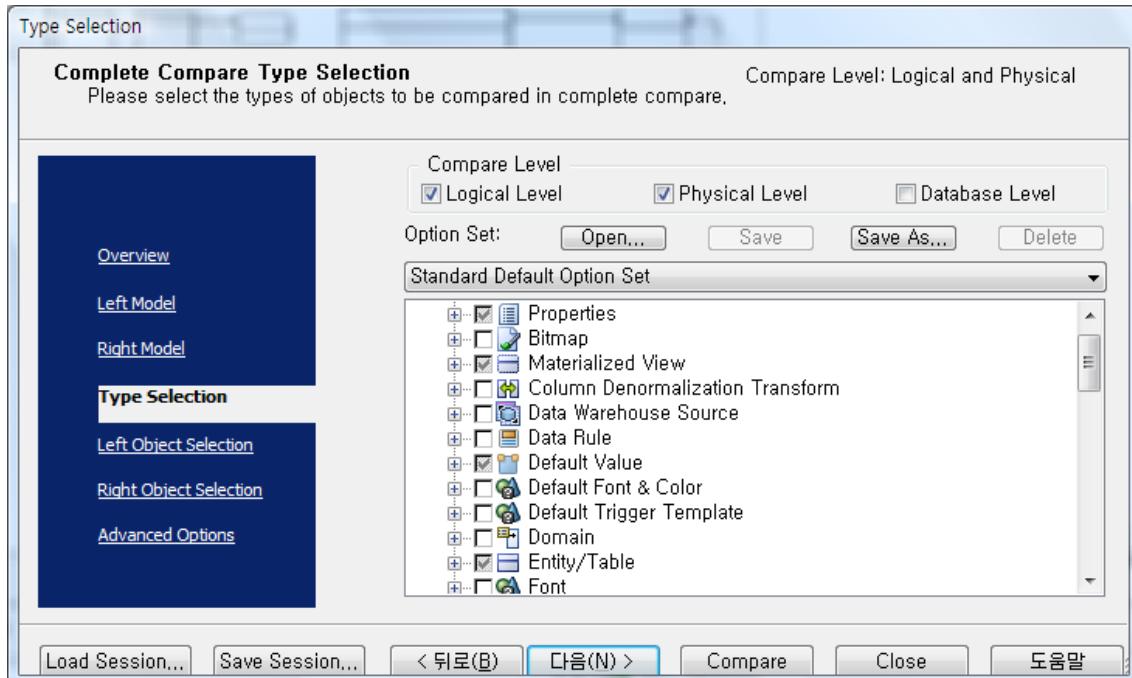
Template형식의 Model을 Open하고 Tools메뉴에서 Complete Compare 를 실행한다.



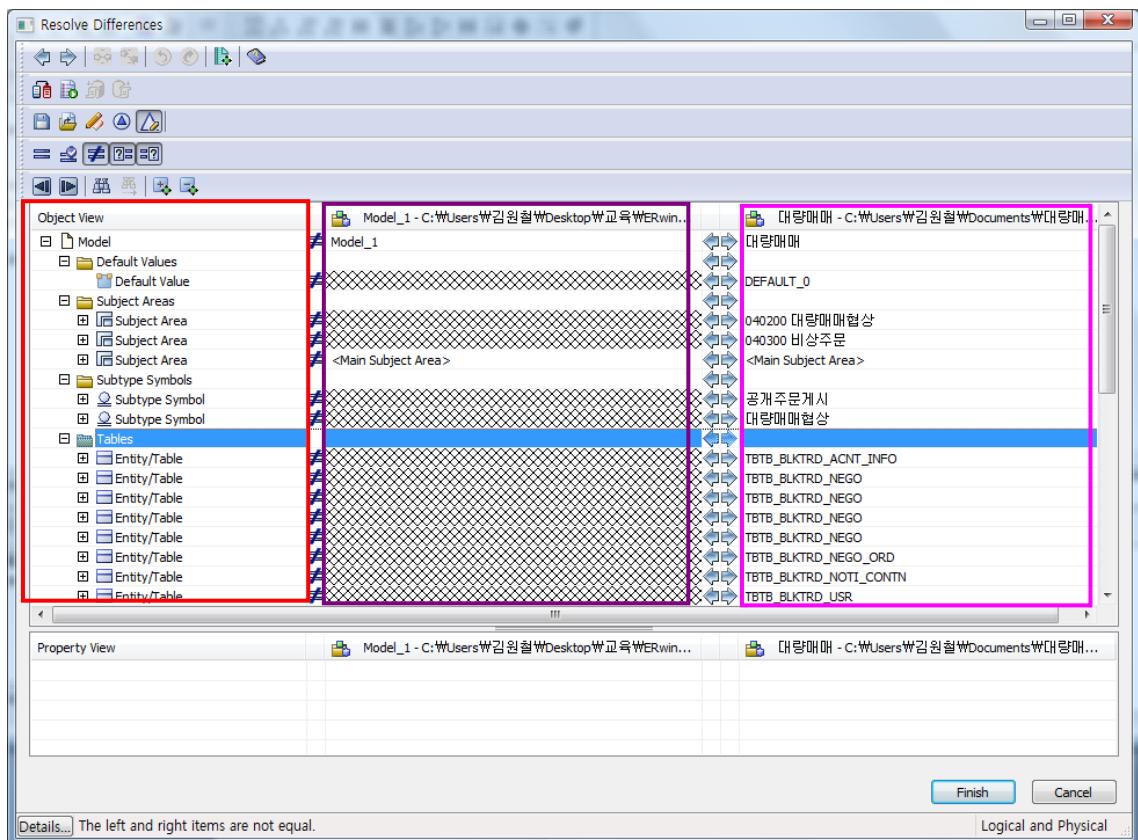
위 스크린샷의 Load를 통해 가져올 ERD를 선택한다.



다음으로 넘어가면 아래와 같이 Type을 선택한다.



Option Set을 Setting 해줄 때, 우선 Domain을 Check를 해제한다. 아래로 스크롤을 이동하면 Subject Area가 있는데, 이는 Check해 주어야 ERD의 Subject Area를 가져온다.
나머지는 Next진행하여 Compare를 실행한다.



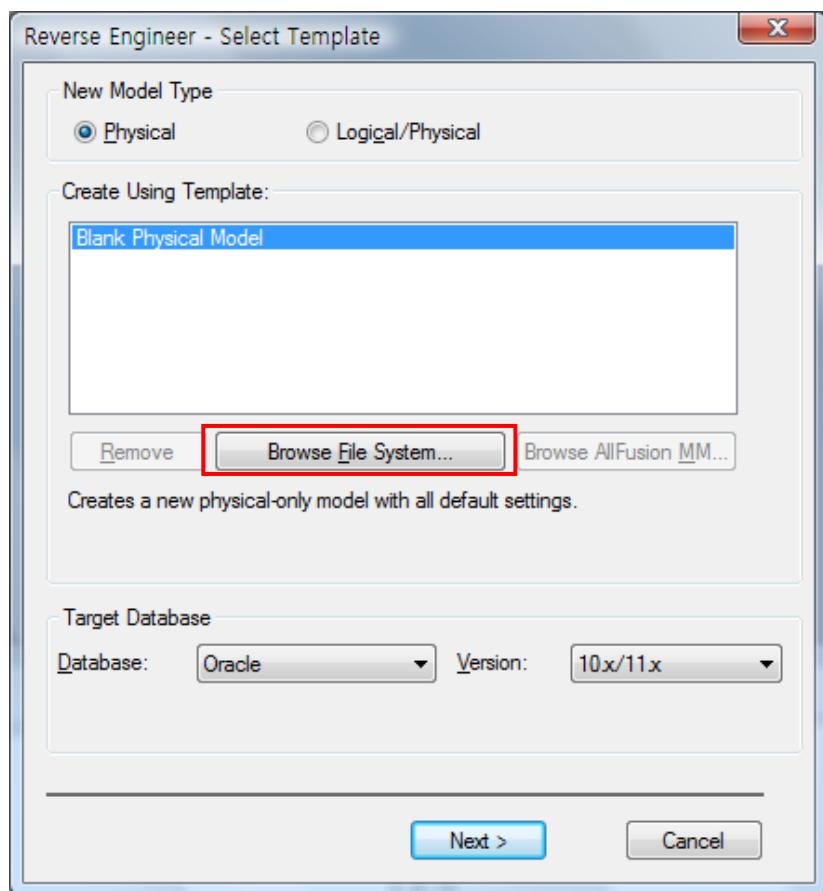
④ 화면구성은 총 3개의 Pane으로 되어 있는데 가장 왼쪽은 비교 항목이다. 가운데는 Template파일을 Left Model Source에 두었기 때문에 가운데 정보이고, 마지막 오른쪽은 Right모델로 생성된 ERD를 가져온 정보다.

최상위의 Model Level에서 화살표를 왼쪽으로 이동시키면 작성되어 있던 ERD의 내용들이 Subject를 포함하여 모두 Template파일로 이동하게 되며, 해당 파일을 다시 저장하면 ERwin파일로 저장하여 관리할 수 있다.

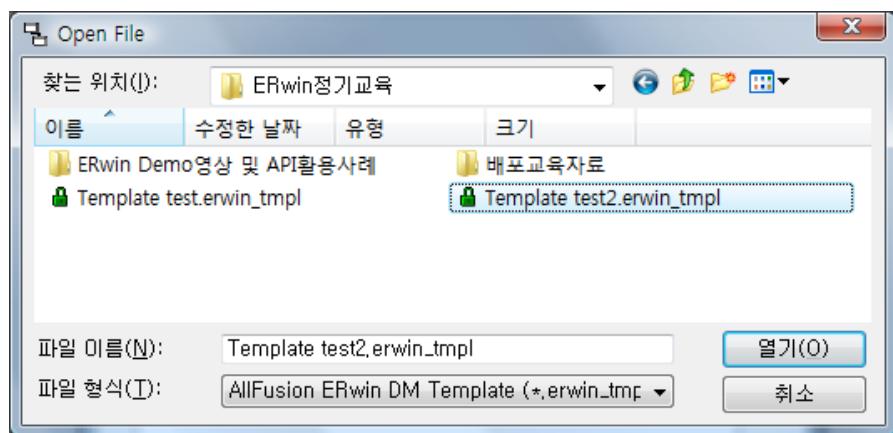
3. Template형식을 적용하여 Reverse 작업

전제조건으로 Template파일이 생성되어 있는 상태여야 한다.

- ① Tools메뉴에서 Reverse Engineer를 실행시킨다.

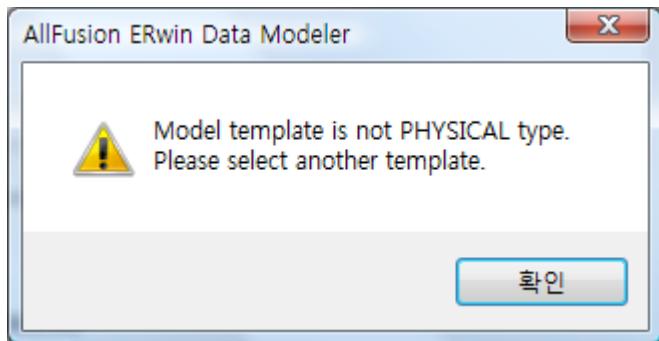


New Model Type을 선택하고, Reverse 밸을 Target Database를 선택한 뒤 위에 표시된 Browse File System을 클릭하여 Template파일을 가져온다.



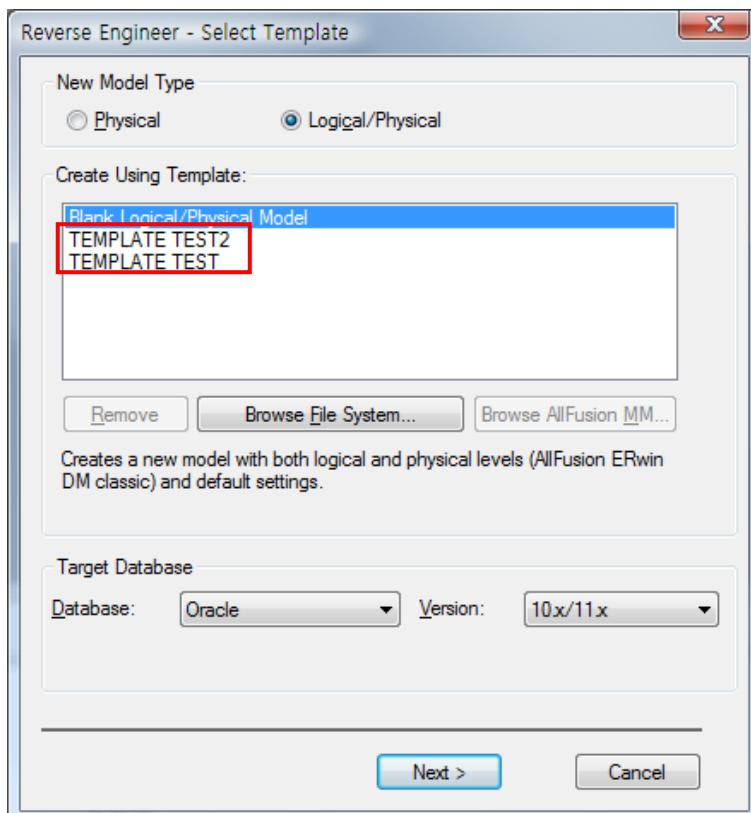
- ② 열기를 수행하는 순간 바로 Reverse Engineer의 Option Set을 설정하는 다음 화면으로 이동한다. Reverse Engineer를 진행하는 방식과 동일하다.

이때 주의해야 할 사항은 Template파일이 Logical / Physical 복합모델 상태의 Template라면 New Model Type에서 Physical을 선택하고 Browse File System을 선택하면 아래와 같은 메시지가 나타나고 다음 단계로 진행되지 않는다.



즉, Reverse받아 생성할 Model Type과 Template파일의 Model Type이 일치해야 한다.

- ③ 마지막으로 사용했던 Template파일은 Reverse Engineer실행시킬 때 보여진다.

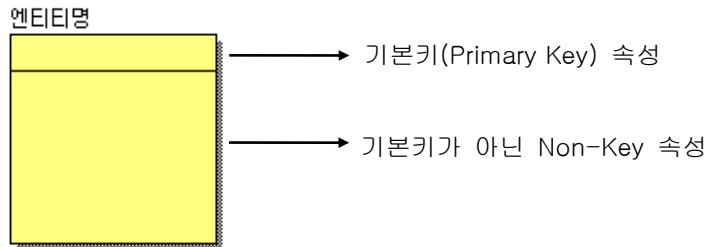


이를 이용하여 다른 DB의 AS-IS모델을 분석할 수 있다. Template파일은 아래쪽에 Remove를 이용하여 제거할 수 있다.

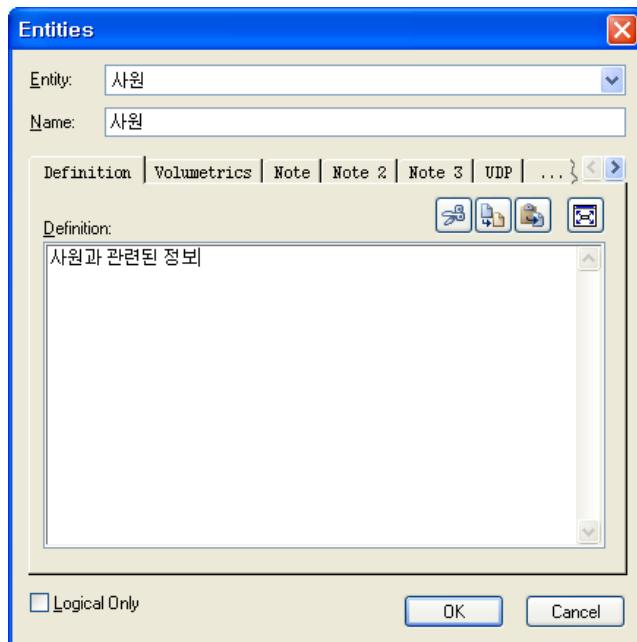
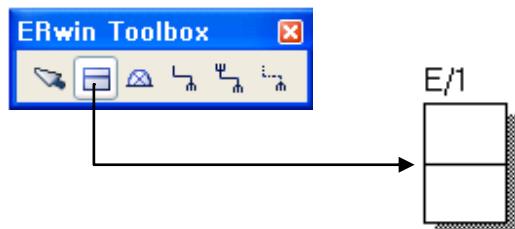
2. Logical Modeling

2.1 Entity 생성

엔티티는 아래와 같이 엔티티명, 기본키(Primary Key) 속성, 기본키가 아닌 Non-Key 속성 영역으로 나뉘게 된다. 영역간의 이동은 Tab 키를 통해 움직일 수 있다.



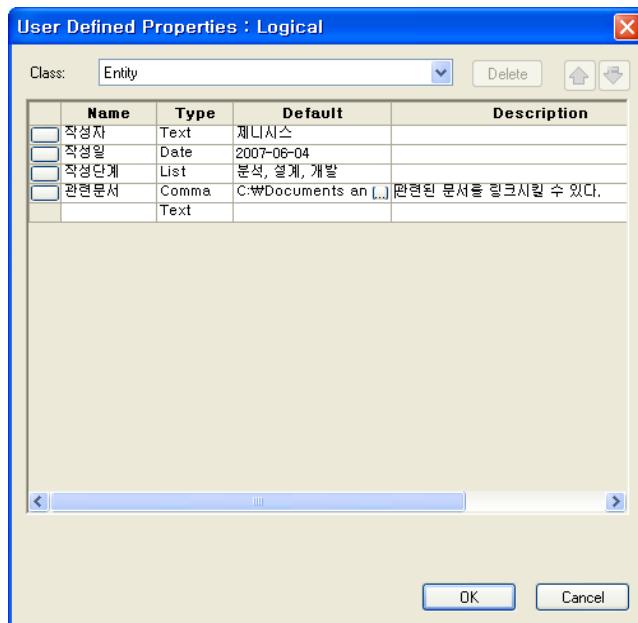
툴 박스의 엔티티 툴을 마우스 왼쪽 버튼을 클릭한 후 Window Diagram의 원하는 위치에 클릭을 하면 엔티티가 나타난다. 한번의 클릭으로 다수의 엔티티를 생성하기 위해서는 Ctrl + Entity 를 선택하면 된다.



Entity를 선택하고 마우스 오른쪽을 클릭하여 Entity Properties를 선택하여 Entity의 Editor를 관리할 수 있다.

항 목	설 명
Definition	엔티티의 정의사항을 기술한다.
Note	엔티티 세부 정보를 기술한다.
UDP	사용자가 정의한 속성을 기술한다.
Icon	아이콘을 설정한다.
History	엔티티의 History를 확인한다.

UDP는 Definition, Note 이외에 항목을 추가하고자 할 때 이용할 수 있으며, 단순히 Text를 입력하는 Editor 외에 여러 타입을 지원한다. 한 번 아래와 같이 UDP를 생성해보자.



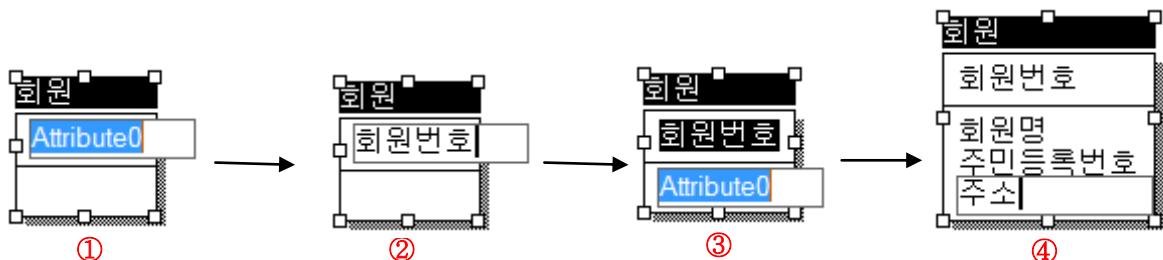
UDP 탭 페이지에서 [...] 버튼을 클릭하면 UDP 항목을 만들 수 있는 창이 나타난다.

항 목	설 명
Name	UDP 항목명을 입력한다.
Type	UDP Type을 선택한다. - Data : 날짜 입력 - Int : 정수입력 - Real : 실수를 입력 - Text : ASCII 텍스트 입력 - List : 리스트 박스를 사용 - Command : 첨부문서 연결
Default	기본 값을 입력한다.
Descript..	Description을 입력한다.

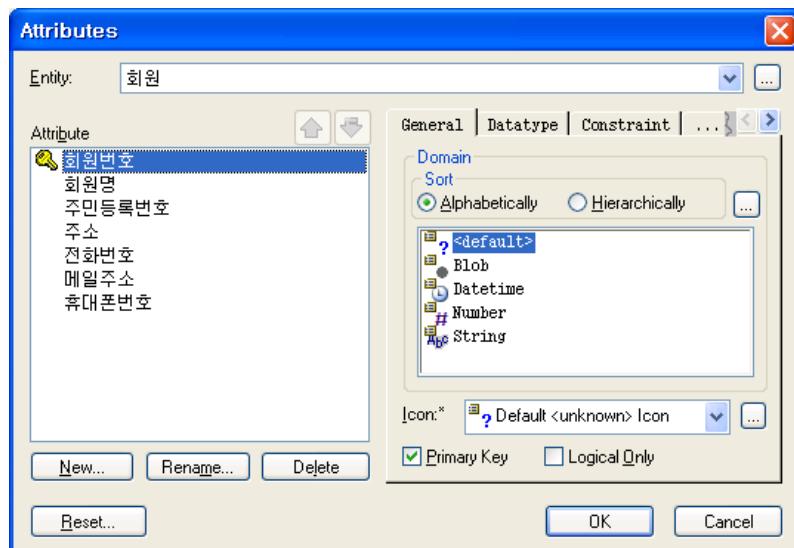
List Type 사용 시 콤마(,)로 구분하여 설정한 UDP 항목은 모든 엔티티에 동일하고, 각각의 값을 입력하면 된다.

2.2 Attribute 생성

Entity 를 선택하여 Tab 키를 눌러 Attribute 이름을 입력한다. PK 가 여러 개인 경우 Enter 키를 눌러 추가할 수 있다. 일반 속성을 입력하기 위해 Tab 키를 눌러 Non Key 영역으로 이동한 후 Attribute 이름을 작성한다. 추가 Attribute 를 작성해야 할 경우, PK 추가와 동일하게 Enter 키를 이용하여 추가할 수 있다.



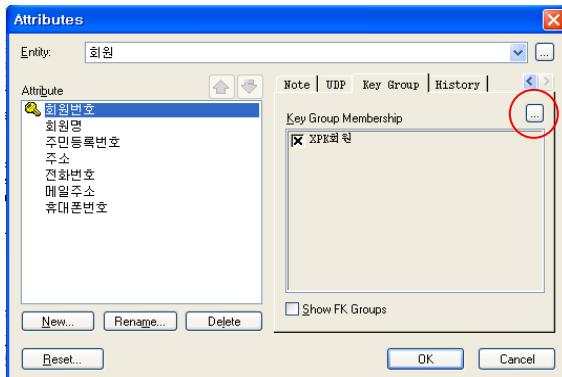
Attribute Editor에서는 Attribute를 생성, 도메인 할당, Definition, Note, UDP를 입력할 수 있다. 또한 Key Group을 등록할 수 있다.



엔티티를 선택 후 오른쪽 버튼을 클릭하고 Attributes를 선택하면 Attribute Editor가 나타난다.

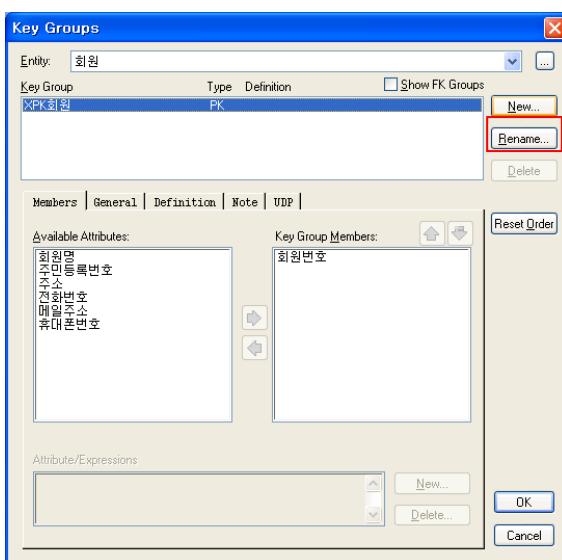
속성 추가, 속성명 변경, 속성 삭제를 할 수 있으며, Reset 버튼을 통해 속성에 대한 초기화 작업을 할 수 있다.

- ① General 탭 페이지에서는 각 Attribute마다 Logical Domain을 할당하고, Primary Key 체크박스를 이용하여 Attribute에 PK를 할당할 수 있으며, Logical Only 체크박스를 선택하면 Physical Model에서 해당 속성이 보이지 않도록 한다.
- ② Datatype 탭 페이지에서는 속성의 데이터 타입을 지정해 줄 수 있다.
- ③ Constraint 탭 페이지에서는 Validation Rule, Default Value를 생성 할 수 있다.
(Validation Rule, Default Value에 대한 자세한 설명은 Physical 쪽에서 할 것이다.)

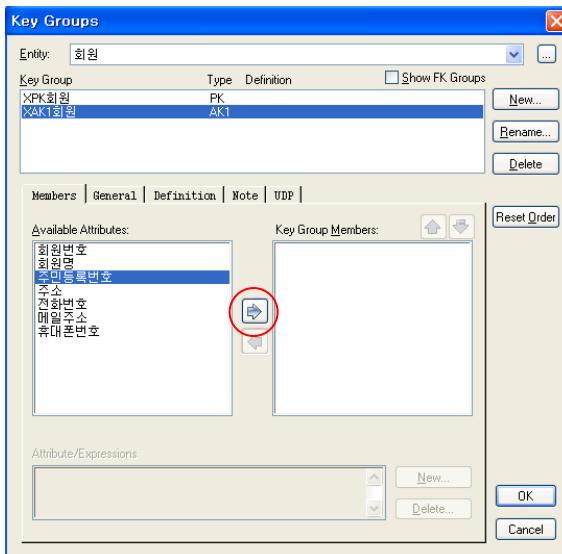


④ Key를 등록하려면 Key Group 탭 페이지를 선택하고 [...] 버튼을 클릭하여 Key Group Editor로 이동한다.

(Key Group은 PK, AK, IE와 같은 Key 들의 집합을 의미한다.)



⑤ New 버튼을 클릭하여 Key Group 타입을 선택하면 Default 이름이 생성되며, Key Group 이름은 수정할 수 있다.



⑥ 생성된 Key Group에 포함시킬 Attribute를 선택하고 Key Group Members로 이동시킨다.

생성한 Key Group에 대해서 이름을 변경하거나 키를 지울 수 있다. 또한 Definition, Note, UDP를 활용하여 Key에 대한 자세한 정보 입력 할 수 있도록 지원한다.

Key Group을 수정하려면 Key Group Membership에서 옵션을 선택 / 해제를 통해 등록 및 변경, 삭제가 가능하다.

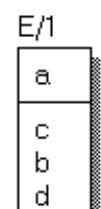
[Tip] 속성 / 컬럼 순서 맞추기

ERwin r7은 속성 / 컬럼 순서가 따로 나뉘어 관리된다. 실질적으로 ERwin r7에서 관리하는 순서는 속성 / 컬럼 / Physical 순서이다.

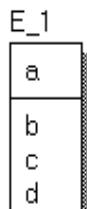
(Physical Order 는 ERwin에서 속성이 실제로 추가된 순서를 말한다.)

속성 순서와 컬럼 순서를 동일하게 관리하기 위해서는 아래와 같은 방법으로 작업해 보도록 하자.

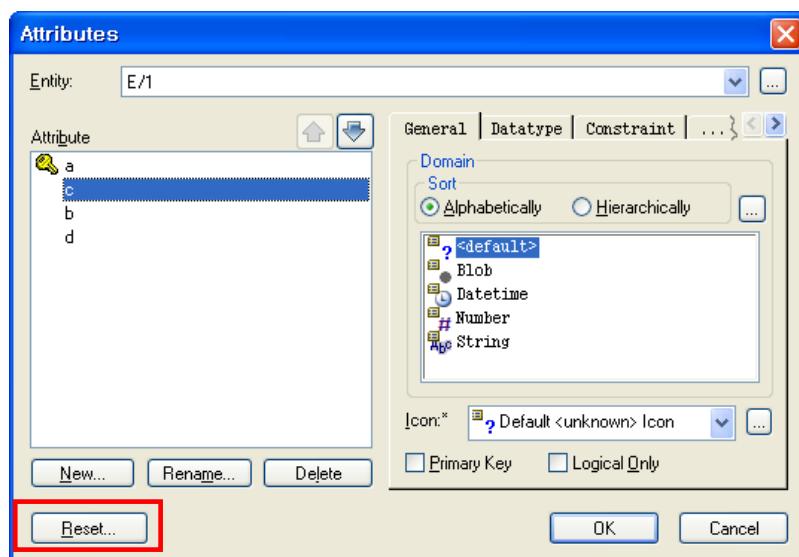
- 1) 컬럼 순서(Physical) 기준으로 속성 순서(Logical)를 맞추기 위해 Logical 영역의 Attribute Editor 를 선택하여 Reset 버튼을 클릭한다.



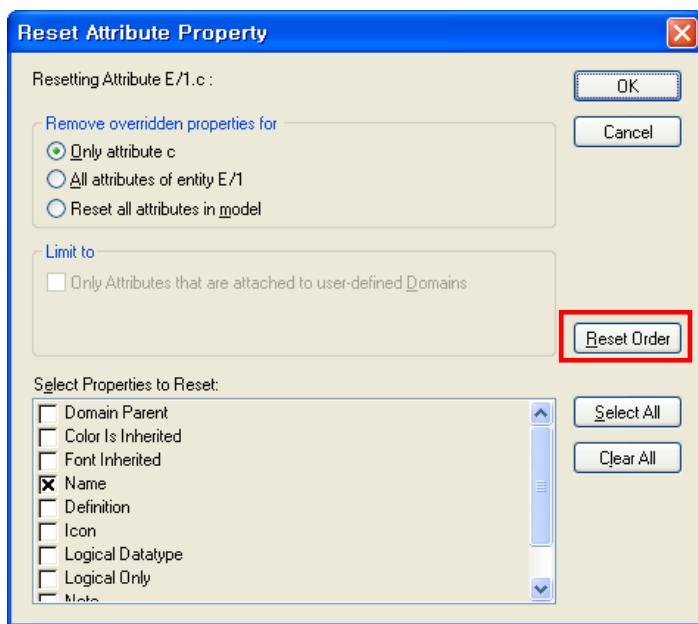
<Logical>



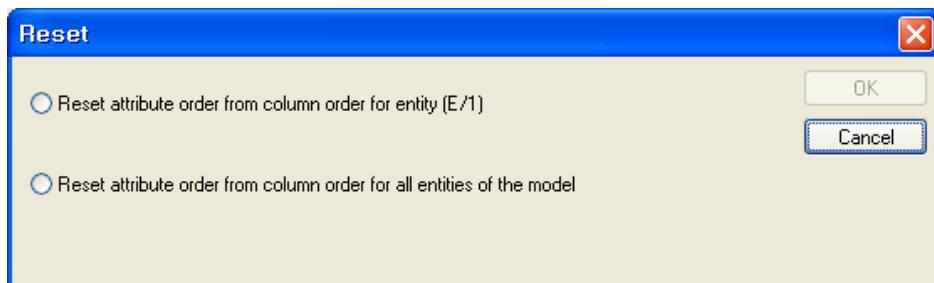
<Physical>



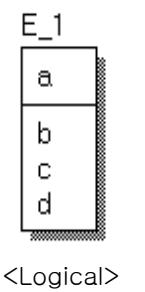
- 2) Reset Attribute Property 창에 들어 와서 Reset Order 버튼을 클릭 한다.



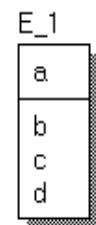
3) Reset Option 창이 뜬다.



첫 번째 옵션은 현재 선택한 엔티티만 컬럼 기준으로 속성 순서를 업데이트 하는 것이고, 두 번째 옵션은 전체 모델의 엔티티를 컬럼 기준으로 속성 순서를 업데이트 하는 것으로 원하는 옵션을 선택한다.



<Logical>



<Physical>

* 속성을 기준으로 컬럼 순서를 맞추려면 Physical 영역에서 위와 동일한 방법으로 진행 된다.

2.3 Relationship 생성

1) 식별관계, 비식별 관계

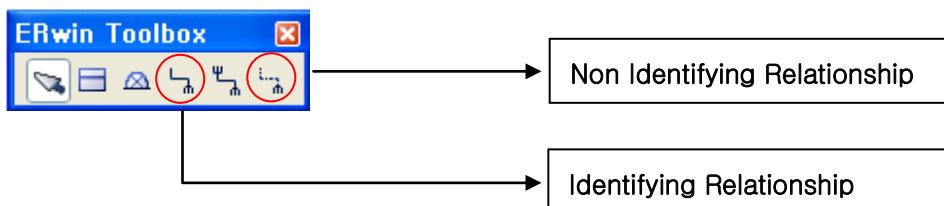
관계란 두 Entity 사이의 업무 규칙, 제약 조건 등을 표현하는 것이라고 정의 내릴 수 있으며 아래와 같은 유형으로 분류할 수 있다.

Relationship 타입	Parent Entity에 대한 Child Entity의 의존성	
	Identity	Existence
Identifying	+	Yes
Non-Identifying Mandatory	+	No
Non-Identifying Optional	o	No
Many-to-Many	→ ←	N/A

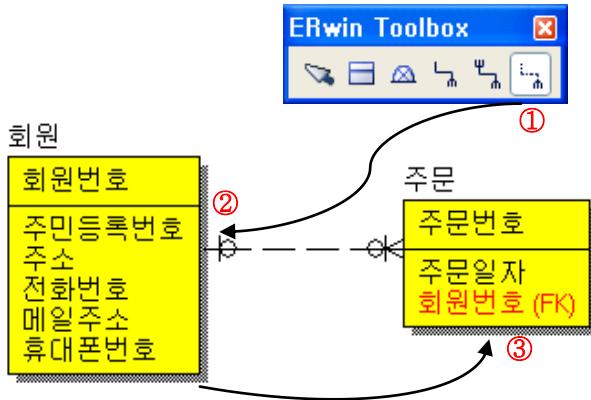
식별관계란 부모 엔티티의 PK들이 자식 엔티티의 식별자 일부로 형상되어 있기 때문에 자식 엔티티의 식별이 부모 엔티티에게 종속된다. 일반적으로 반드시 부모 엔티티가 존재해야 자식 엔티티가 존재할 경우. 자식 엔티티로부터 상속받은 다른 엔티티에 주식별자가 상속되어야 할 경우 혹은 업무적으로 엔티티 간에 강한 종속 관계로 연결되어야 할 때 식별 관계를 맺게 된다.

반면 비식별 관계는 부모 엔티티의 PK가 자식 엔티티의 기본 키의 일부가 아니기 때문에 부모 엔티티에 의해 식별되지 않는 비종속 관계이다. 부모 엔티티의 주식별자가 여러 개고 주식별자 속성 모두가 자식 엔티티에 주식별자 그대로 상속이 필요하지 않고 부분적으로만 참여되길 원할 때 비식별 관계를 맺으면 된다.

ERwin ToolBox에서 네 번째 버튼인 실선 아이콘이 식별 관계선이고, 여섯 번째 버튼인 점선 아이콘이 비식별 관계선이다.

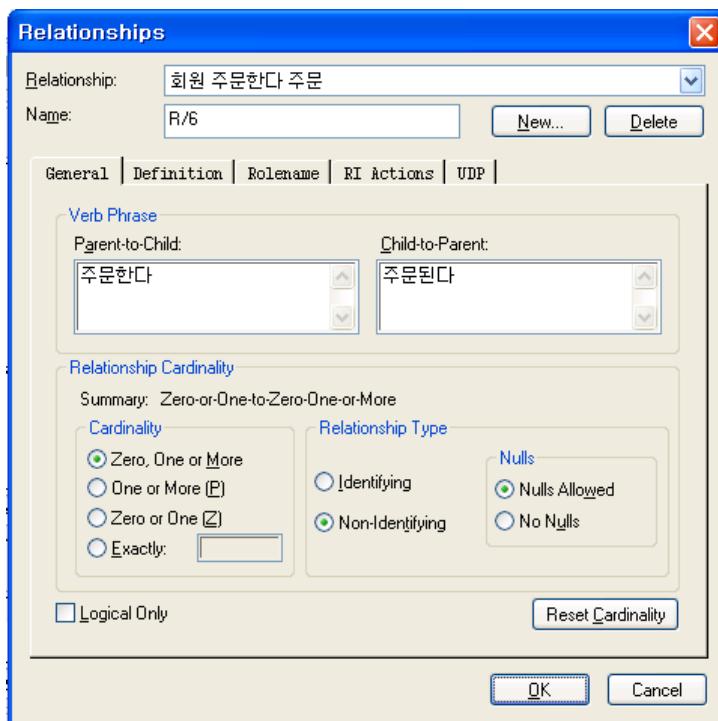


관계선을 생성하기 위해서는 Toolbox에서 원하는 Relationship을 선택한 뒤, Parent Entity와 Child Entity를 차례대로 클릭한다. 마우스로 관계선을 dragging하여 재배치 할 수 있다.



이 예에서 회원에 등록되지 않은 비회원이 주문하는 것이 가능한 경우, 회원이라는 부모 엔티티가 존재하지 않아도 자식 엔티티가 존재할 수 있다.

관계선에서 마우스 오른쪽 버튼을 클릭하여 Relationship을 선택하면 아래와 같은 화면이 나타난다. General 탭에서는 Verb Phrase, Cardinality, Relationship 타입을 변경(Identifying/Non-Identifying, Nulls Allowed/No Nulls) 등을 할 수 있다.



이 대화상자에서는 대부분 Cardinality와 Relationship Type을 설정한다. Cardinality는 두 테이블에서 레코드들의 맵핑 정보를 보여주게 되는데 옵션을 바꾸어가면서 실제 다이어그램에서의 바뀌는 모습을 확인해 보도록 한다.

위의 예에서 회원이 주문을 한 번, 혹은 여러 번을 할 수도 있고, 하지 않을 수도 있기 때문에 첫 번째 Zero, One or More 옵션을 선택하면 된다.

그리고 Verb Phrase에서 부모와 자식간의 관계, 자식과 부모와의 관계에 대한 설명을 작성함으로써, 관계에 대한 가독성을 높일 수 있다.

Relationship Type 에서는 회원테이블과 주문 테이블은 한 명 혹은 여러 명의 회원이 주문을 할 수도 있고, 비회원이 주문할 수도 있으므로 부모 엔티티가 존재하지 않아도 자식 엔티티가 존재 할 수 있는 비식별 관계에 Null을 허용할 수 있는 관계를 형성을 할 수 있다.

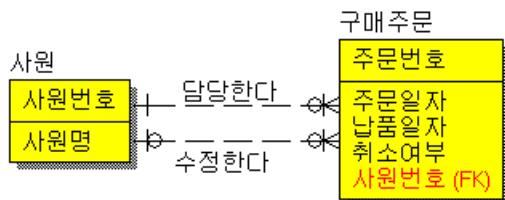
식별관계(Identifying Relationship)에서는 외부 키가 키 영역(PK)으로 이동한다. 부모 엔티티의 PK 들이 자식 엔티티 식별자의 일부를 형성하기 때문에 이 관계를 식별 관계라고 한다. 즉, 엔티티의 식별은 부모 엔티티에 종속된다.

비식별관계(Non-Identifying Relationship)는 점선으로 표시되며 역시 부모 엔티티를 자식 엔티티와 연결하게 되면 외부키가 자식 엔티티 Non-Key 영역에 놓이게 된다.

2) RoleName (역할 이름)

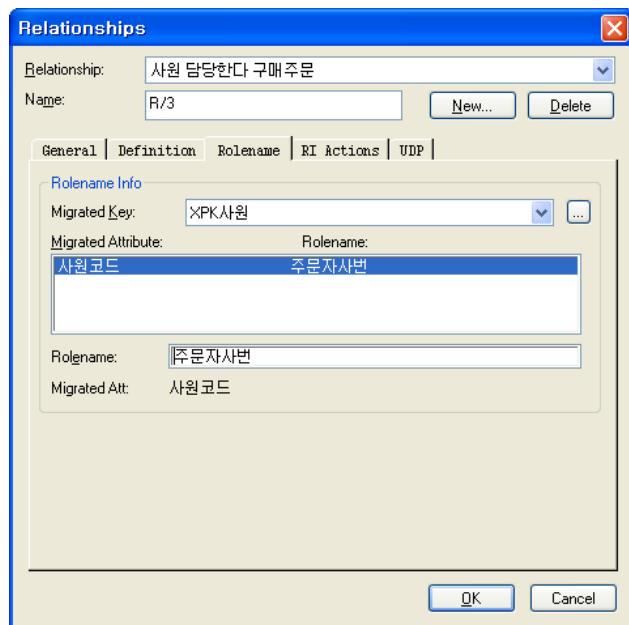
RoleName(역할이름)은 엔티티 내에서 속상(Attribute)dl 하는 역할을 정의하는 외부 키 속성의 새로운 이름이다. RoleName은 새 속성을 선언하는데, 이 속성의 이름은 외부 키를 부여하는 관계에 의해 구현된 업무에 관한 서술문을 설명하기 위해 붙여진 이름이다.

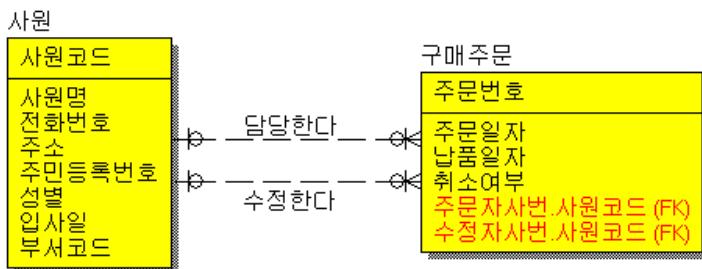
1) 관계선에서 마우스 오른쪽 버튼 클릭하여 Relationship Properties메뉴를 선택한다.



2) Rolename 탭 페이지에서 ‘사원번호’ 속성 Rolename을 ‘주문자 사번’이라고 입력한다

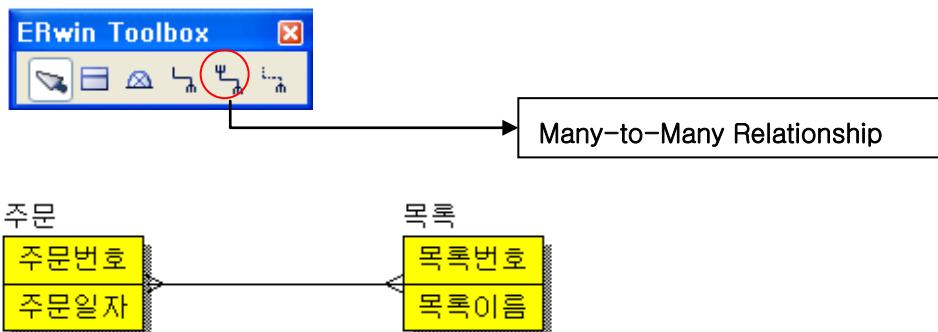
3) 또 다른 Relationship Editor에서 ‘사번’속성의 Rolename을 ‘수정자 사번’으로 입력한 뒤 OK버튼을 클릭한다.





3) M:M 관계

아래의 예처럼 M:M 의 관계선을 이용하여 주문 테이블과 목록 테이블을 만들어 보자.



이 관계는 업무적으로 발생할 수 있으므로 Logical Modeling에서는 M: M 관계를 완전히 허용하고 있으나, 이를 관계형 데이터베이스에 구현할 수 없기 때문에 1: M의 관계를 풀어서 Associate Entity를 생성해야 한다. 왜냐하면 대부분의 업무 프로세스와 상세 정보가 이 Associate Entity에서 정의 되기 때문이다.

M: M 관계를 해소하는 방법은 다음과 같다.



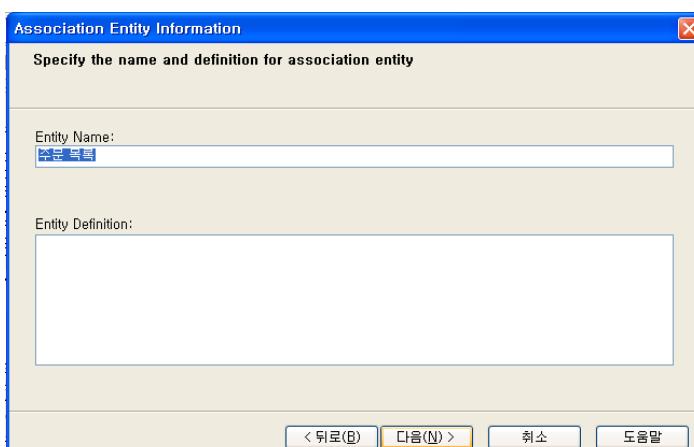
M: M 관계선을 선택하여 마우스 우측 버튼 클릭 후, 'Show Association Entity'를 선택한다.



Wizard Guide화면이 나타나면 ‘다음’ 버튼을 클릭한다.



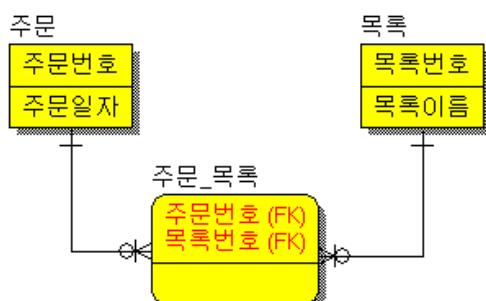
Transform Name과 Definition을 입력 한다.



새롭게 생성될 Association Entity의 이름을 입력한다.

기본적으로 두 엔티티 명이 합쳐진 이름이 보여진다.

계속 나타나는 화면에서 ‘다음’ 버튼을 클릭하고 ‘마침’ 버튼을 클릭한다.

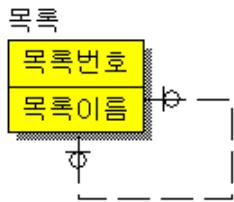


옆의 화면처럼 주문 엔티티와 목록 엔티티의 M: M 관계가 각각 1: M 의 관계로 나뉘어 지는 것을 볼 수 있다.

4) 재귀 관계

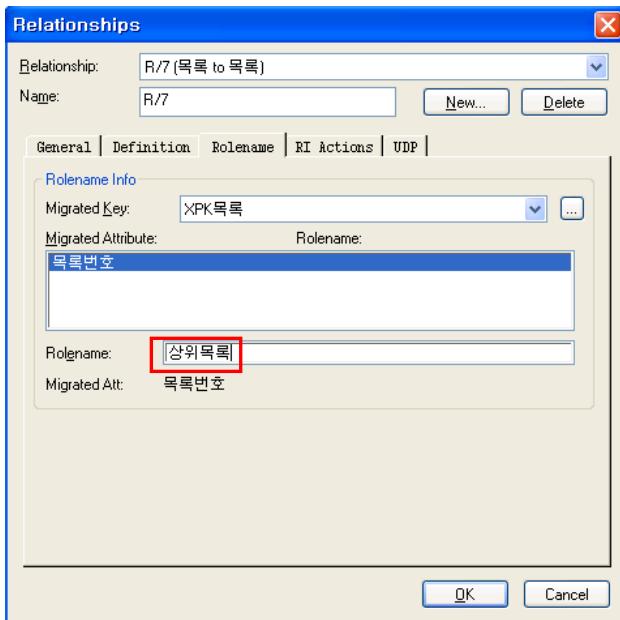
재귀 관계란 같은 엔티티가 부모 엔티티와 자식 엔티티인 경우이다. 재귀 관계는 모두 비식별 관계여야 한다.

아래의 예를 보면 목록이라는 테이블에 재귀 관계를 형성한 것이다.

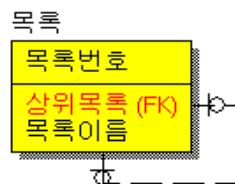


이 때, 목록번호라는 속성에게 단일화 현상이 발생한 것을 볼 수 있다. (단일화 현상이란 한 Attribute 가 Foreign Key 로서 다른 Entity 로 이동할 때 이미 동일한 이름의 Attribute 가 Entity내에 존재한다면, 두 개는 하나의 Attribute 로 합쳐진다.) 이렇게 단일화 된 속성의 뜻을 명확하게 하기 위해 RoleName(역할이름)을 부여할 수 있다.

RoleName(역할이름)을 부여하는 방법은 Relationship Properties 를 클릭하면 RoleName 이라는 탭이 있다.



RoleName 에 주식별자를 참조하는 속성인 ‘상위목록’을 기술한다.

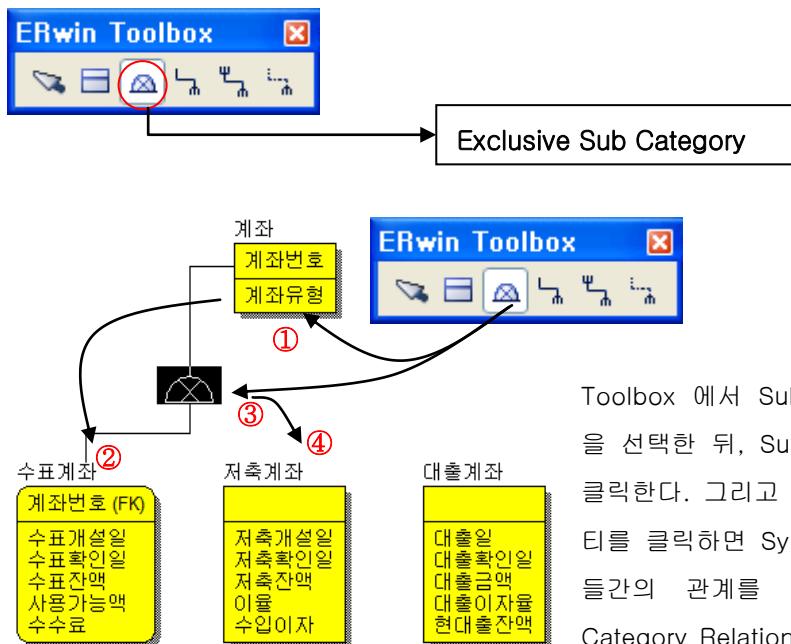


5) 일반화 관계 (Supertype/Subtype 관계)

일반화 구조(Generalization Hierarchy)는 공통 Attribute를 공유하는 Entity의 계층 그룹이다. Generic Parent(Supertype Entity)는 공통 Attribute와 Primary Key를 포함한다. Category(Subtype Entity)는 개별적인 Attribute와 Foreign Key를 포함한다.

일반화 구조는 Entity의 ‘종류’ 개념이 존재할 때 생성할 수 있다.

일반화 관계의 특징을 보면 Supertype Entity 와Subtype Entity 는 1:1 관계고, Subtype Entity 를 구분할 수 있는 구분 속성은 반드시 존재해야 한다. 또한 Subtype Entity에 대한 Subtype 을 지정할 수도 있지만 복잡성이 증가하므로 한 레벨로 제한하는 것이 좋다.



Toolbox에서 Sub Category Relationship 아이콘을 선택한 뒤, Supertype Entity 인 계좌 엔티티를 클릭한다. 그리고 Subtype Entity 인 수표계좌 엔티티를 클릭하면 Symbol 이 생긴다. 또 다른 엔티티들간의 관계를 맺기 위해, Toolbox 의 Sub Category Relationship 아이콘을 클릭하고, Symbol

을 클릭한 뒤, Subtype Entity 인 저축계좌 엔티티를 클릭한다.

이와 같은 방법으로 다른 Subtype Entity 와도 관계를 맺어보자.

일반화 관계에는 Exclusive 와 Inclusive 두 개의 종류로 나뉜다.



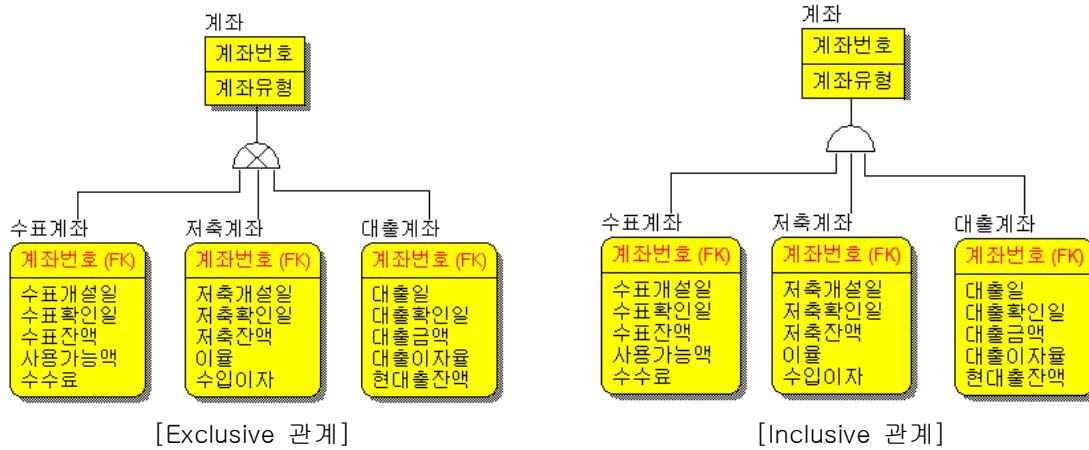
Exclusive: Supertype 인스턴스가 단 하나의 Subtype 엔트리와 관계를 가지고 있을 때 이용한다.



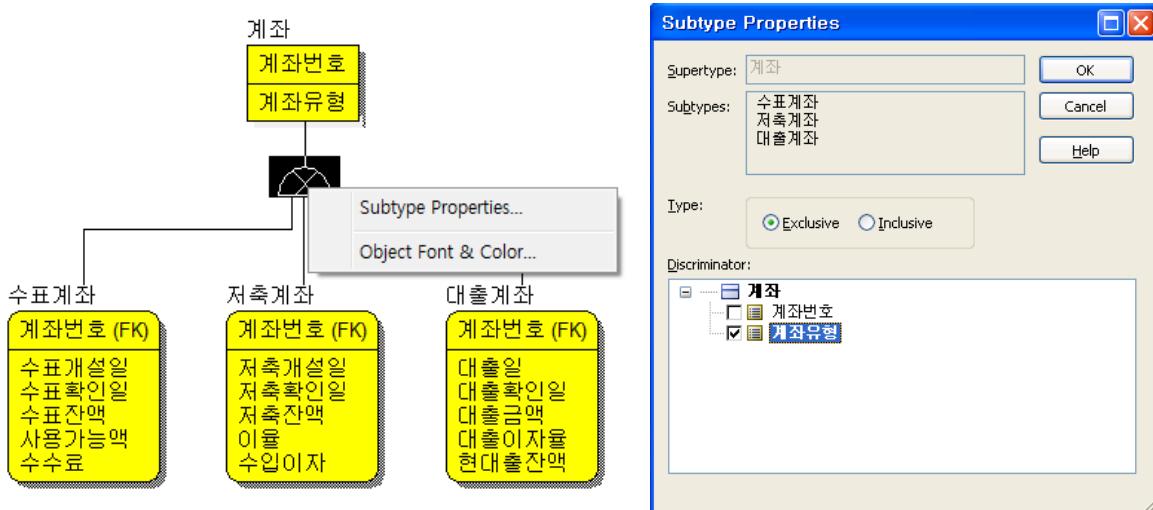
Inclusive(Non-exclusive): Supertype 인스턴스가 하나 또는 그 이상의 Subtype 엔트리와 관계를 가지고 있을 때 이용한다.

하나의 계정은 반드시 저축계정이거나 당좌 계정이어야 한다. 두 가지 이상일 수 없다면 Exclusive 관계로 맺으면 된다.

하나의 계정은 저축 계정일 수도 있고 당좌 계정일 수도 있다. 아니면 두 가지 모두일 수도 있다면 Inclusive 관계로 맺으면 된다.



일반화 관계에서 구분자 표시 및 Exclusive, Inclusive 종류의 선택은 Symbol 을 클릭한 후 마우스 오른쪽 버튼을 클릭하여 Subtype Relationships 를 선택하면 아래와 같은 Subtype Properties 가 나타난다. 이를 통해 Exclusive / Inclusive 타입을 선택할 수 있고, 구분자 속성을 지정할 수 있다.



2.4 Subject Area (주제영역)

Subject Area란, 하나의 모델에는 엔티티가 수십 개, 수백 개가 존재하는데 이를 비즈니스 단위 혹은 프로세스로 분류하는 것이다. Subject Area를 통해 업무적으로 관련되어있는 Entity, Relationship 등의 정보들만 확인하고 작업할 수 있기 때문에 편리하다.

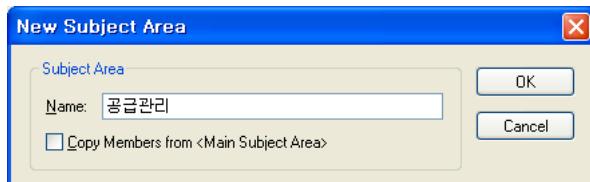
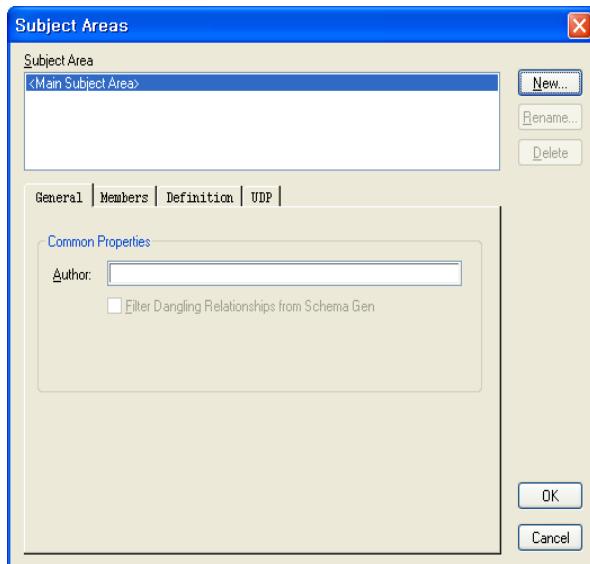
각 Subject Area에서 Object의 추가, 수정, 삭제 등의 작업한 내용이 전체 ERD에 반영되기 때문에 편리하고 데이터베이스 스키마를 생성할 때 Subject Area 별로 스키마를 데이터베이스에 생성할 수 있으므로 유용하다.

이 때, object의 추가, 수정, 삭제는 실시간으로 Main Subject Area와 Subject Area에 반영되지만 위치 정보의 변경은 반영이 되지 않는다. 일반적으로 Main Subject Area에는 수백 개, 수천 개의 object 가 있어서 관리하는 것이 쉽지 않으므로 Subject Area 별로 관리가 이루어 진다. 그러므로 Subject Area에서만 위치 정보를 관리하면 되므로 편리하다.

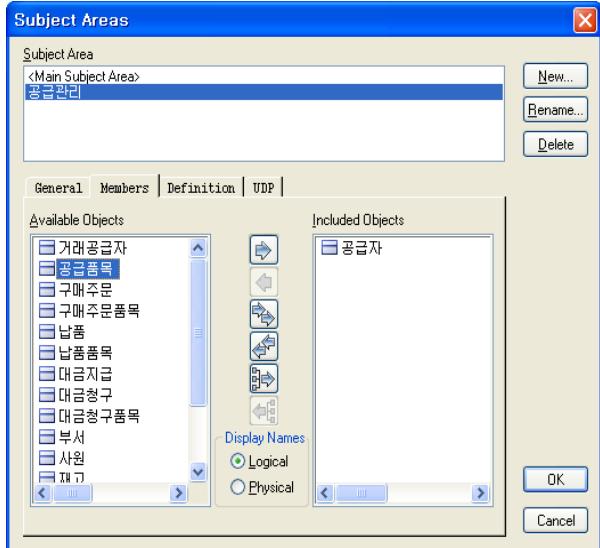
디폴트로서 데이터 모델은 <Main Subject Areas>가 된다.

Subject area를 생성해 보자.

1) Model메뉴에서 Subject Areas메뉴를 선택하거나  아이콘을 선택한 뒤, new 버튼을 클릭한다.



새롭게 생성할 주제영역 이름 '공급관리'를 작성한다.

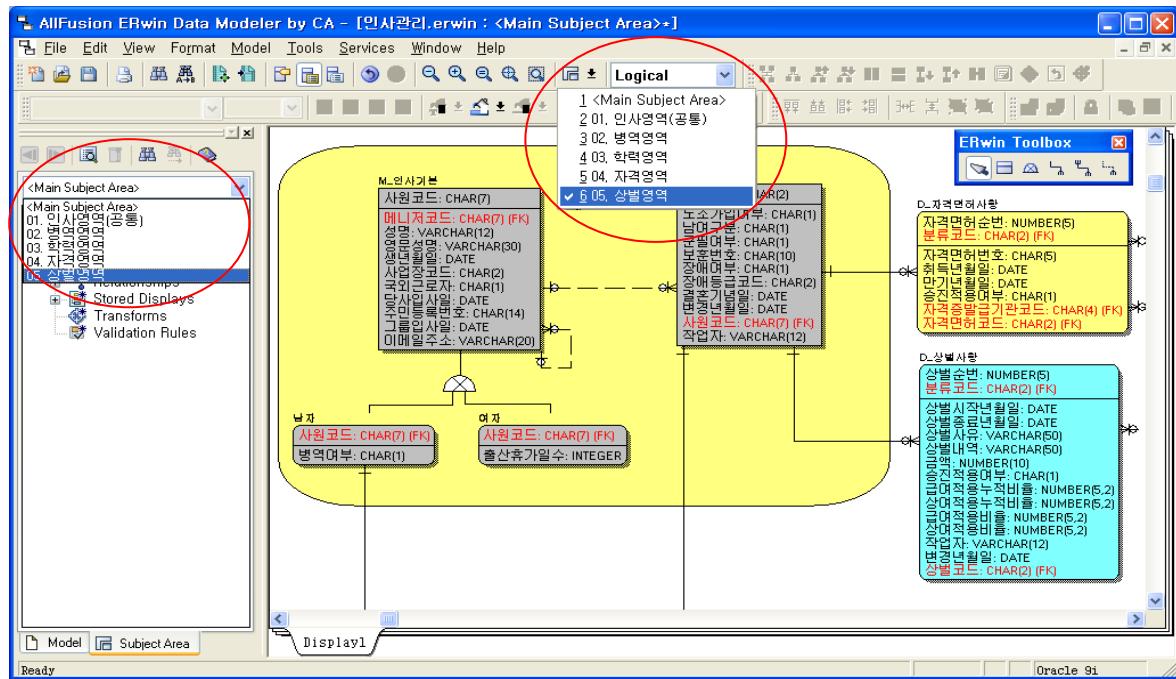


Members 탭 페이지에서 각 주제영역에 해당하는 Entity를 Included Objects쪽으로 이동시킨다.

특정 Entity를 클릭하고 버튼을 클릭하면 해당 Entity와 관계된 Entity들을 한꺼번에 옮길 수 있다. 이때 관계 맺고 있는 레벨을 선택할 수 있기 때문에 유용하다.

Subject Area를 삭제할 때는 메뉴 Edit에서 Subject Area를 선택하고 삭제하려는 Subject Area를 선택하고 Delete 버튼을 클릭한다.

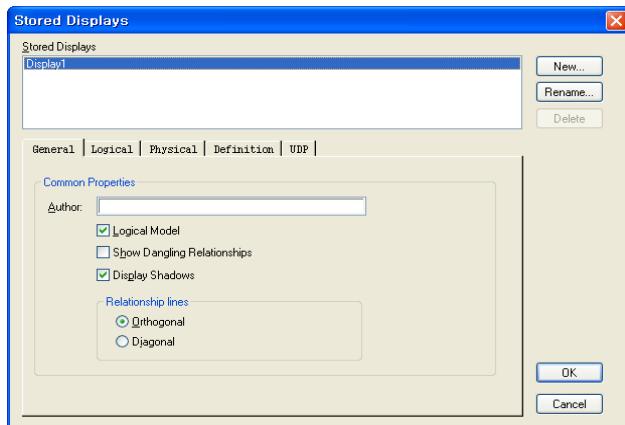
하나의 Diagram에 여러 개의 Subject Area를 만들었다면 Subject Area 리스트 박스에서 이동하고자 하는 Subject Area를 클릭하거나 Model Explorer창의 Subject Area 탭을 통해서 원하는 Subject Area로 이동할 수 있다.



2.5 Stored Displays

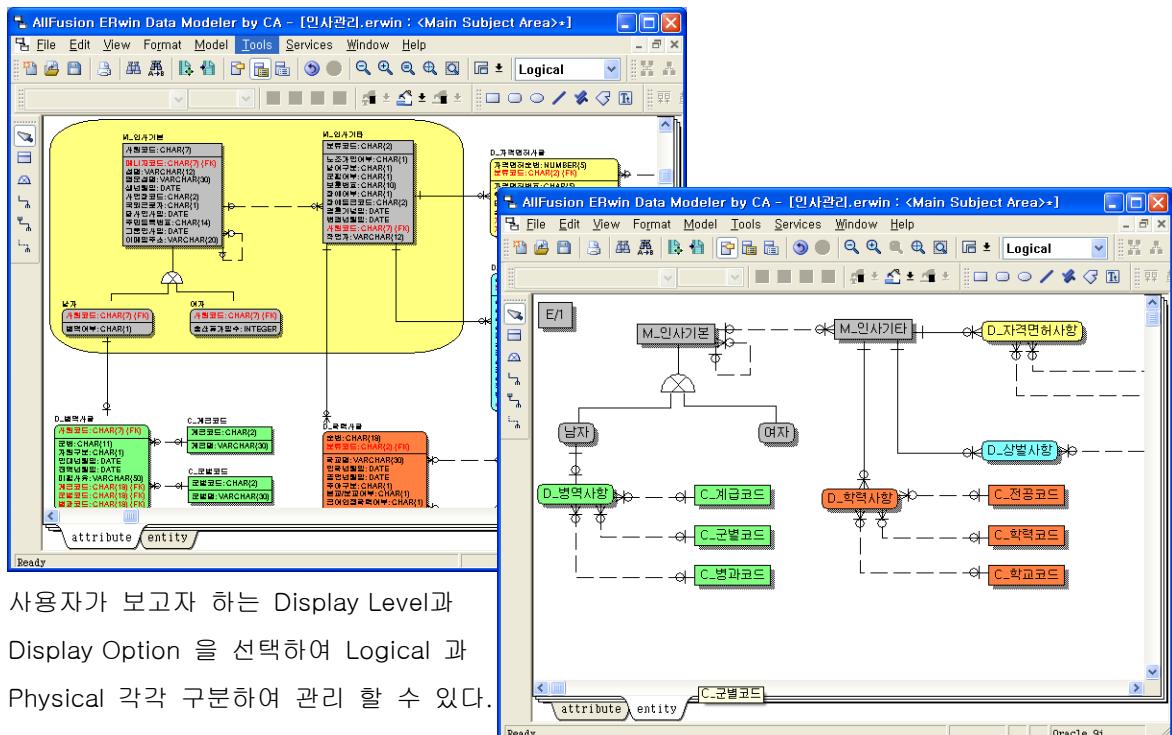
Stored Display를 통해 모델을 다양한 배열로 저장할 수 있다. 이것은 서로 다른 배경지식이나 요구 사항을 가지고 있는 사용자들에게 다양한 모델의 형태를 보여주는데 효과가 있다. Stored Display는 각 Subject Area마다 분리하여 작성할 수 있으며, 각 Subject Area는 자신만의 Stored Display 를 가질 수 있다. 이것은 화면 아래 부분의 탭 페이지로 보여진다.

Stored Display 생성해보자.



Format 메뉴의 Stored Display Settings…을 선택하면 Default로 ‘Display1’이라고 적혀 있을 것이다. Rename 버튼을 클릭 후 ‘Attribute’라고 입력하고 New 버튼 클릭 후 ‘Entity’라고 입력한다.

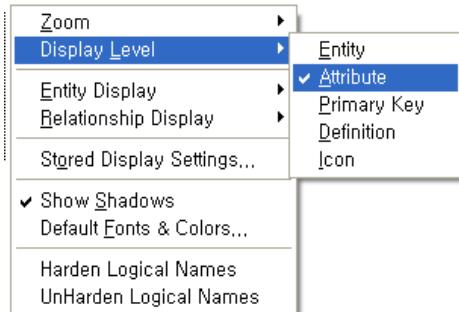
Entity Display 탭 선택하고 바탕화면에서 마우스 오른쪽 버튼을 클릭하여 Display Level 을 Entity로 한 후, 화면에서 Entity와 Relationship을 재배치하여 모델을 정리한다.



사용자가 보고자 하는 Display Level과 Display Option 을 선택하여 Logical 과 Physical 각각 구분하여 관리 할 수 있다.

그렇다면 Display Level 과 Display Option에 대해 알아보도록 하자.

바탕화면에서 마우스 오른쪽 버튼 클릭하여 Display Level 메뉴의 5가지 Level 중 원하는 메뉴를 선택하여 원하는 형태로 화면을 구성 할 수 있다.



Display Level을 통해 보여줄 수 있는 정보의 Level을 정할 수 있다.

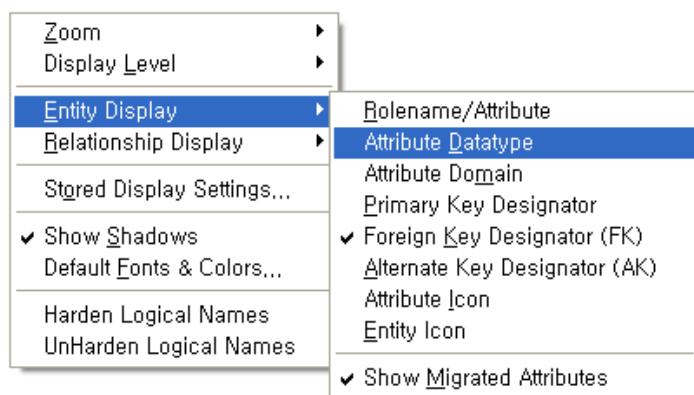
Entity Level은 전체 모델의 개괄적인 형태를 제공하여 프로젝트의 범위를 설정할 수 있도록 한다.

Attribute Level은 Logical 모델을 분석하고 검사하는데 기본적인 작업 환경을 제공한다. Primary Level은 모델러가 업무간의 관계를 살펴보는데 도움을 준다.

Entity마다 충분한 Definition을 정의한 경우에는, Definition Level로 보는 것이 의미가 있다.

Icon Level은 프리젠테이션 용으로 유용하게 활용된다.

바탕화면에서 마우스 오른쪽 버튼 클릭하여 Entity Display를 선택한다. 각 항목의 선택여부에 따라 Entity에 보여지는 내용이 달라진다



Entity: Rolename, Logical Datatype/Domain, FK Designator, AK Designator, Show Migrated Attribute 등을 선택

Relationship: Verb Phrase, Cardinality, Referential Integrity 등을 선택

[Tip] Default Fonts & Colors… 옵션을 이용하여 Foreign Key에 대하여 색을 다르게 줄 수 있다.



- ① Attribute 탭을 선택한다.
- ② Inherit Color From PK 체크를 풀어준다.
- ③ 원하는 색을 선택한다.
- ④ 적용 대상을 선택한다.
- ⑤ 확인 버튼을 클릭한다.

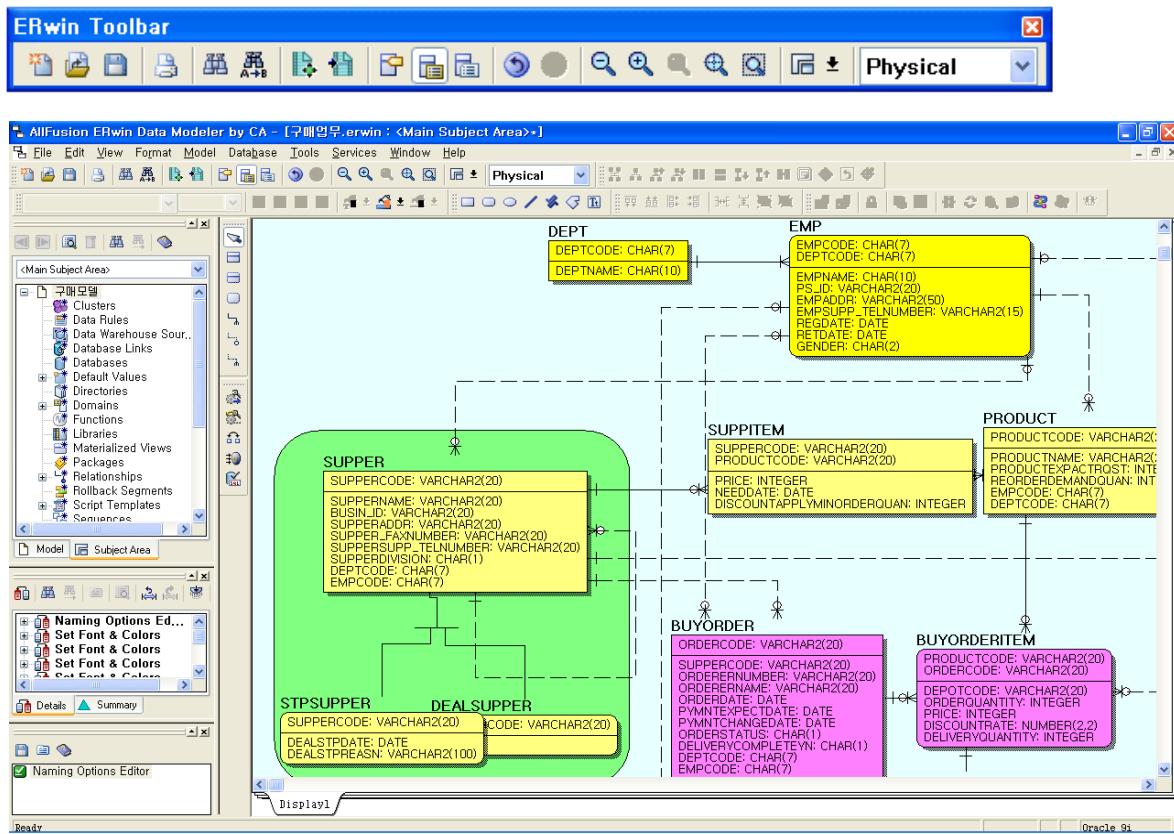
3. Physical Modeling

3.1 Physical / Logical Model의 표현

Physical 모델에서는 Target Server의 물리적 특성을 지원하여 물리적인 데이터 모델과 Target Server의 시스템 카탈로그를 동기화 한다.

ERwin Data Modeler는 선택된 타겟 서버 DBMS에 기초한 물리 모델을 관리한다. 즉 선택한 DBMS에 유용한 Column 옵션과 Physical storage properties를 제공하는 것이다. 또한 인덱스 옵션도 타겟 서버에 맞게 정의할 수 있다.

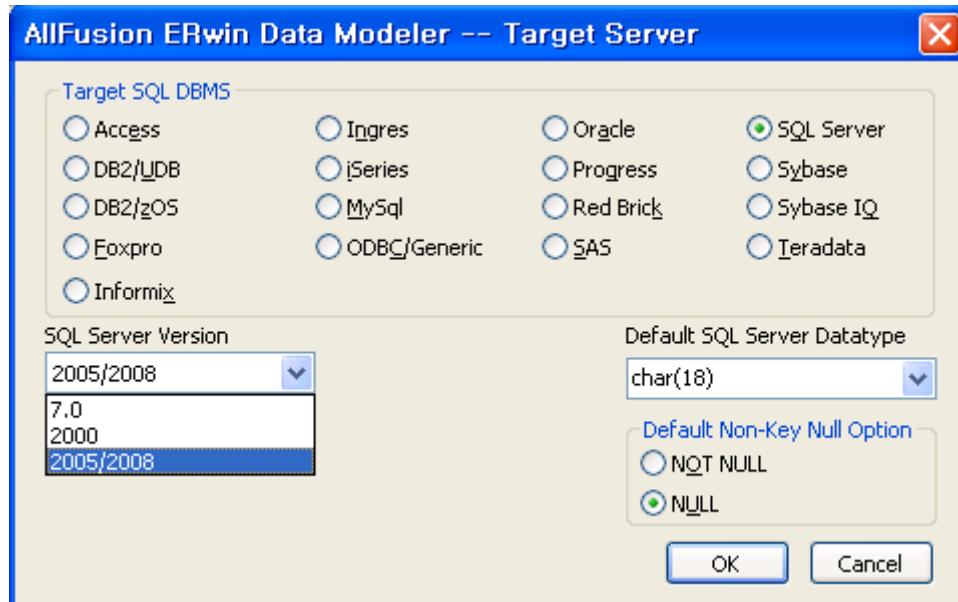
물리 모델로의 변환은 메뉴에서 Physical 모드를 선택함으로써 쉽게 변환이 가능하다.



Tip

Layer 전환 단축키 : Ctrl + ↑, ↓

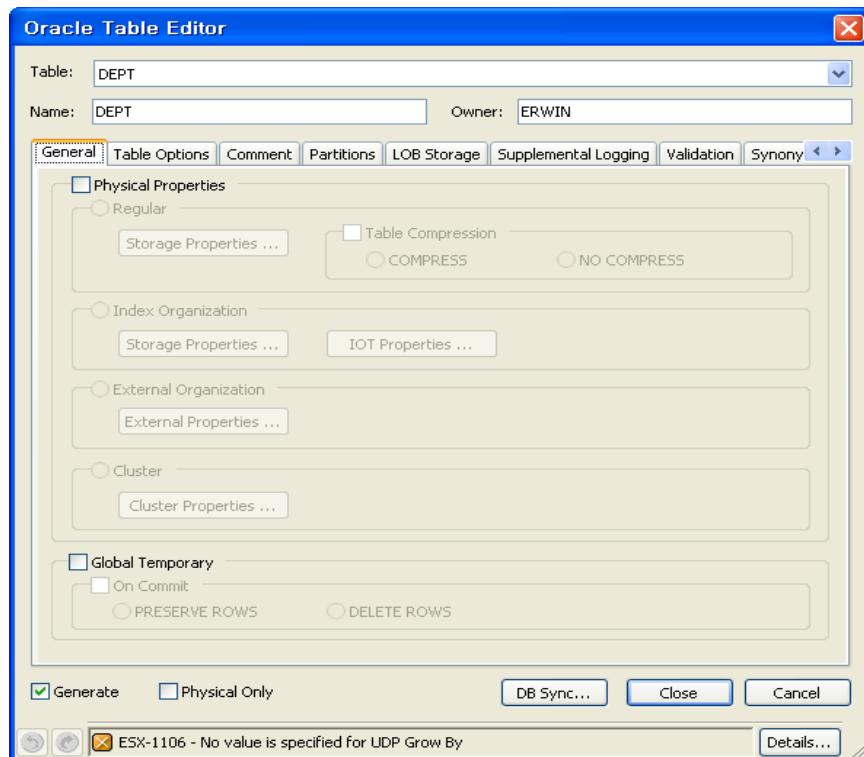
만약 Target Database를 변경하고자 하거나, 확인하려고 한다면 Database 메뉴에서 choose database 메뉴를 선택하도록 한다..



3.2 Table 관리

논리모델에서 만든 테이블의 Editor 를 확인해보자. 여기서 테이블 생성 / 삭제 /수정이 가능하며, DB와의 Sync 작업 및 순서도 변경할 수 있다.

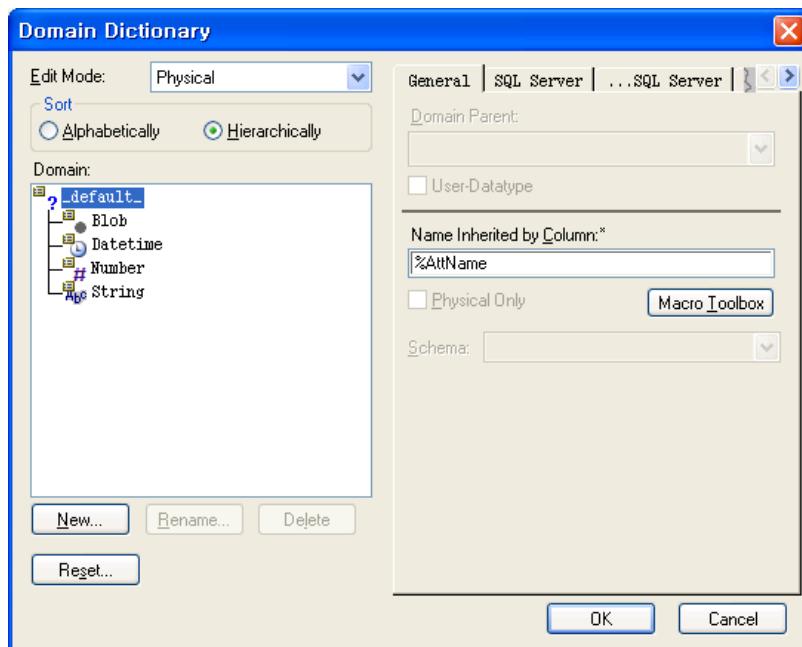
각 테이블에 대한 schema 선택 및 Physical Only 를 지정해줄 수 있다. 논리모델에서 관리하지 않을 테이블에 대해서는 Physical Only 에 체크해주면 된다. 일반적으로 Forward Engineer 작업 시, 모든 테이블들이 생성되도록 Generate 부분에 체크가 되어 있으나 생성하지 않을 테이블에는 체크를 해지해주도록 한다.



3.3 Domain 관리

모델링 작업을 할 때, 속성에 대한 데이터타입과 크기, 제약사항을 지정하는 도메인 작업은 굉장히 중요하다. 도메인을 정의하여 사용하면 속성이 아무리 많아도 일관된 규칙에 따라 데이터 타입과 크기가 부여되며 프로젝트 도중 도메인이 변경되었을 때 일관성 있게 관리할 수 있으므로 모델의 관리가 용이하다. 유사한 속성이 많이 쓰이고 엔티티의 개수가 많거나 다수의 모델러들이 작업하는 경우에는 반드시 도메인이 사용되어야 한다.

ERwin에서는 이러한 도메인을 지원하기 위해 Blob, Datetime, Number, String 의 네 가지 기본 도메인을 정의하고 있다.



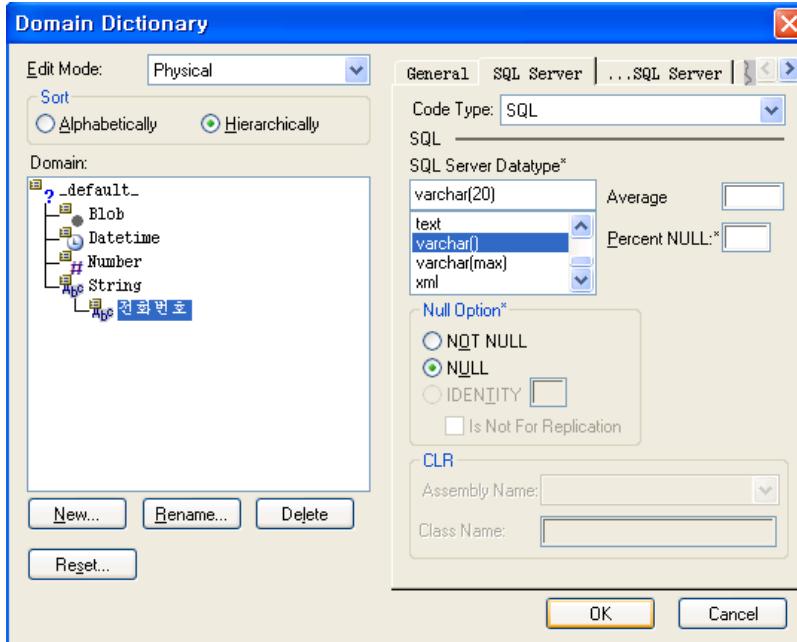
일반적으로 코드, 이름, 날짜, 전화번호 등과 같은 도메인은 공통 속성이 존재한다. ERwin을 통해 한번 만들어 보도록 하자.

- 1) New 버튼을 클릭하여 도메인 명 ‘전화번호’를 입력한다.
- 2) 우측 General 탭에서 생성하려는 Domain Parent 를 String 으로 지정해준다.



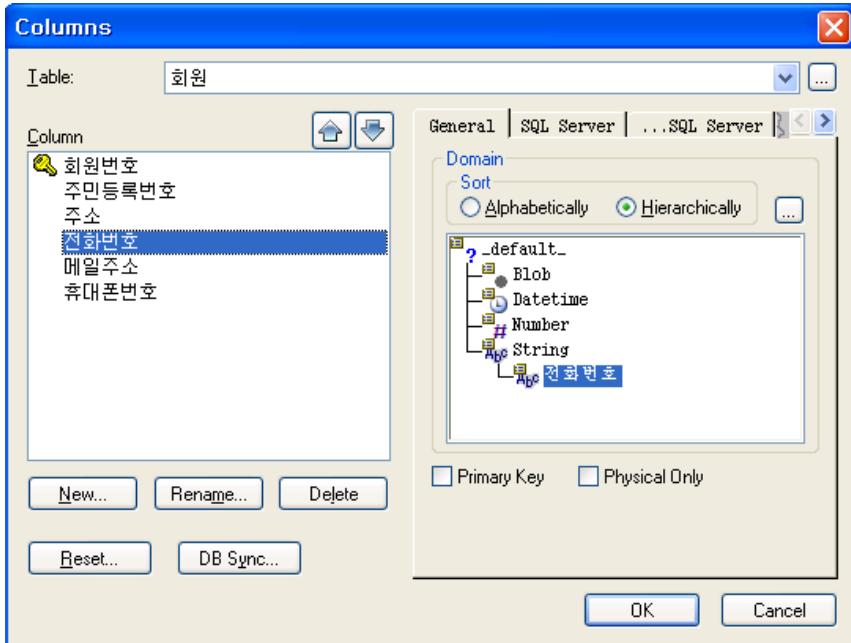
[User-Datatype]에 체크를 하면 해당 도메인의 데이터 타입을 통해서만 데이터 타입과 크기를 변경할 수 있다. 체크를 하지 않는다면, 각 속성들의 데이터 타입과 크기를 Column Editor에서 변경할 수 있다.

3) SQL Server 탭에서 도메인의 데이터타입과 길이를 지정한다.



3) 회원이라는 엔티티의 전화번호 속성을 전화번호 도메인과 매핑하는 작업을 해보자.

회원 테이블을 선택하고 마우스 우측 버튼을 클릭하여 Column..을 선택하면 Column Editor가 오픈된다. 전화번호 속성을 선택한 뒤, General 탭에서 전화번호 도메인을 선택하면 전화번호 속성은 전화번호 도메인의 데이터 타입과 길이를 상속 받는다.



도메인을 수정하면 해당 도메인을 매핑하고 있는 속성의 데이터타입도 같이 변경된다.

<주의사항>

하지만 도메인을 수정하지 않고 직접 속성 에디터 창에서 데이터타입에 대한 수정을 가하면 도메인과 속성의 연관관계가 끊어진다.

아래는 회원이라는 테이블에 각 도메인을 매핑한 것이다.

회원	
회원번호:	char(?) 코드
주소:	varchar(20) 주소
전화번호:	varchar(20) 전화번호
메일주소:	varchar(20) 이메일주소
휴대폰번호:	varchar(20) 전화번호
주민등록번호:	varchar(20) 주민등록번호

만일 도메인을 지정하지 않으면 '<Default>' 도메인이 지정된다. <Default> 도메인은 특별히 도메인을 지정하지 않는 속성들을 뜻하며, 도메인이 설정되지 않았으므로 속성에 맞는 데이터타입을 지정하면 된다. 도메인이 매핑되지 않은 속성들의 데이터 타입 지정과 관련된 설명은 아래 '3.4 Column 관리'를 참고하도록 한다.

3.4 Column 관리

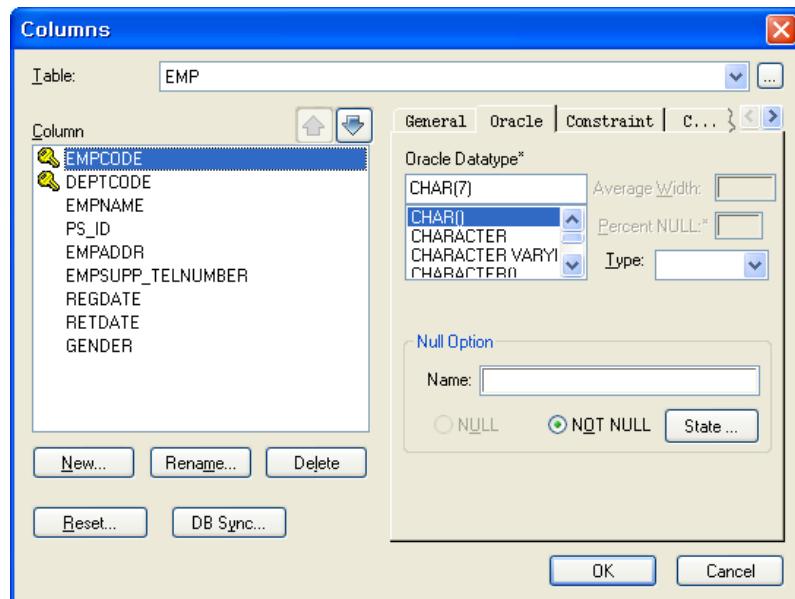
1) Column 의 Datatype 관리

도메인이 매핑되지 않은 속성들의 데이터타입은 기본적으로 Char(18)로 매핑되어 있고 이는 각 속성의 성격에 맞게 데이터타입과 크기를 정의할 수 있다.

정의하는 방법은 다음과 같다.

① 해당 테이블을 선택하고 마우스 우측 버튼을 클릭하여 Column Editor 를 오픈한다.

② 해당 속성에 맞는 데이터타입과 사이즈를 정의해준다.



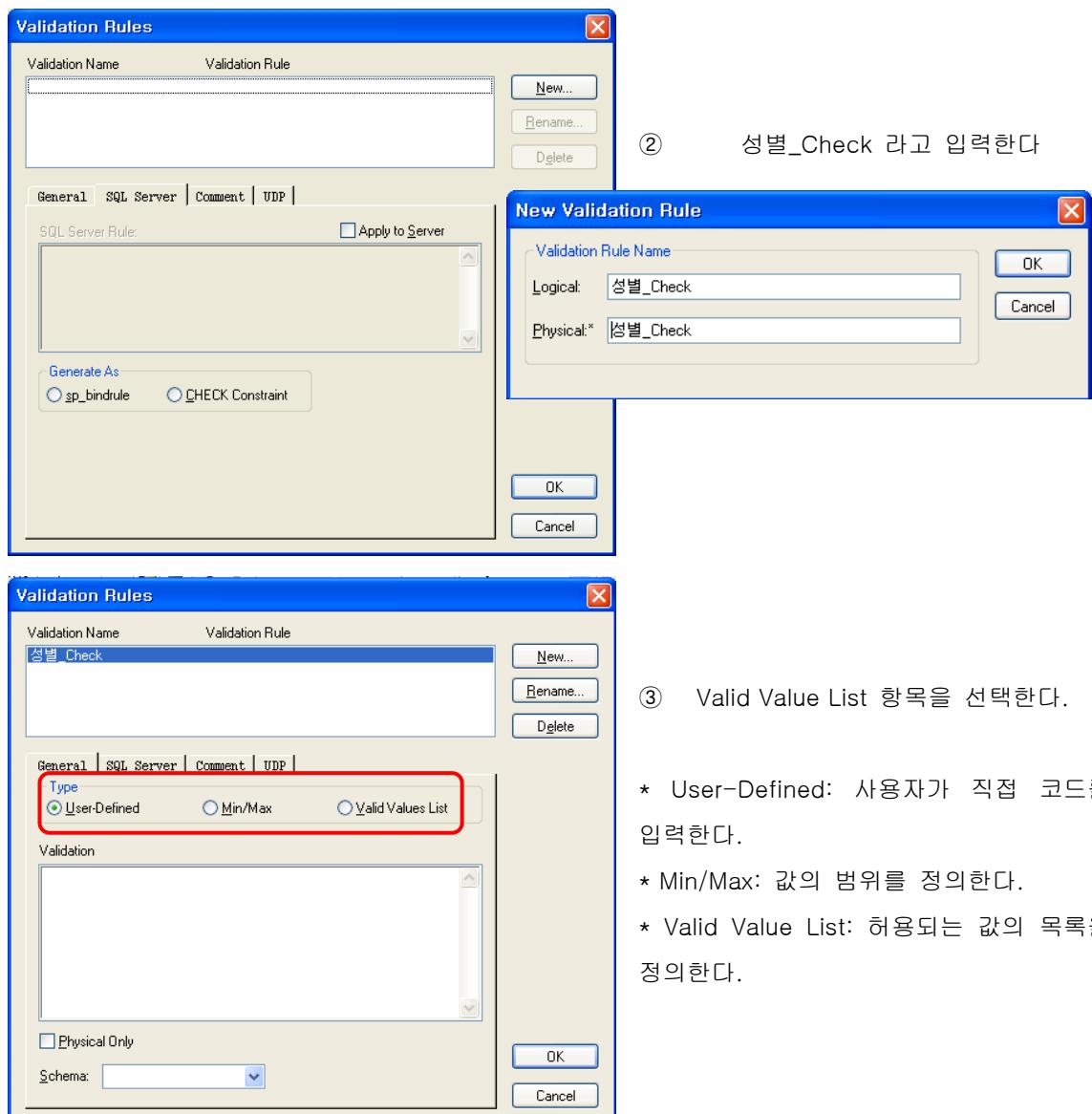
3) Column 의 Validation Constraint / Default

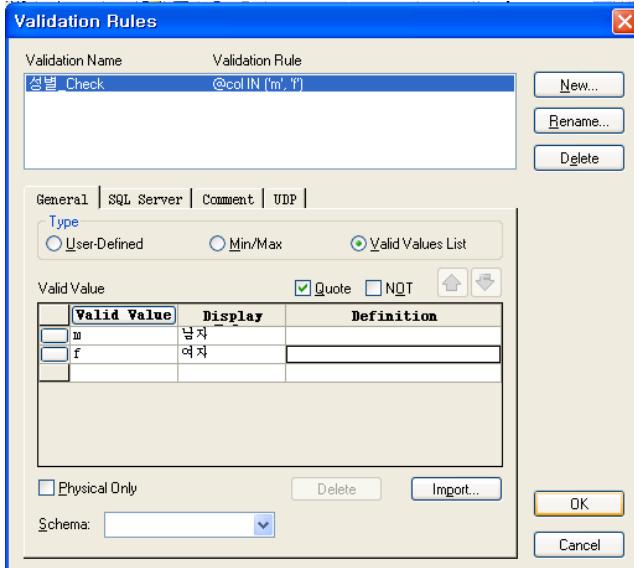
Validation Constraint란 값이 올바른지, 혹은 올바른 범위의 값들이 입력되었는지의 체크 제약 조건이다. 테이블 혹은 컬럼에 대해 갖게 되는 값의 범위를 지정해 줄 수 있다.

ERwin에서도 Table과 Column에 Validation Constraint 를 설정할 수 있다. 이를 통해 특정 Table / Column에 제한된 데이터 값을 지정하기 위한 규칙(Rule)을 생성 하여 이를 Schema 생성시 생성한 규칙에 대해서 자동으로 DDL 문을 생성할 수 있다.

예를 들어 회원 테이블의 성별이란 컬럼에 업무규칙에 따라 ‘m’ 혹은 ‘f’의 값만 갖도록 제한해보자.

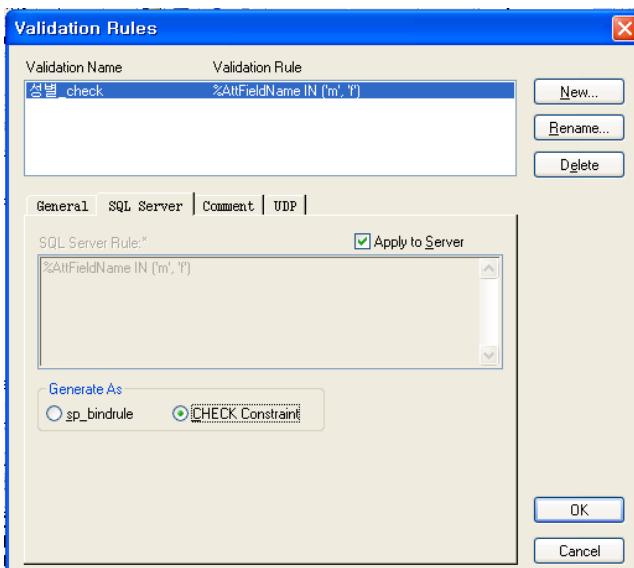
- ① Model < Validation Rules 를 클릭한다





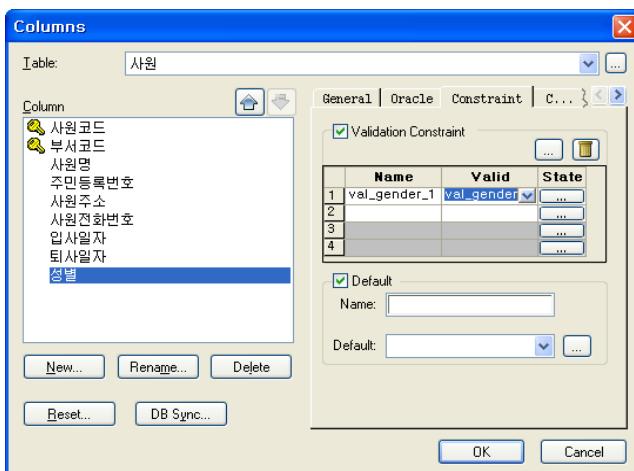
④ Valid Value List에 옆의 그림과 같이 입력한다.

* Valid Values List 정의시 Valid Value에 직접 유효값을 입력한다. 만약 Value 값이 문자형 값이라면 Quote를 체크 하고, 지정한 Value 값을 포함하지 않도록 하려면 스크립트에 NOT을 체크한다.



⑤ SQL Server 탭으로 가서 Generate As 영역에서 CHECK Constraint 을 선택한다

위의 예는 성별 컬럼에 적용되는 제약 조건을 정의하는 것이므로 CHECK Constraint 옵션을 선택해야 한다.
Sp_bindrule 옵션은 Rule을 만들 때 적용하는 옵션이다.

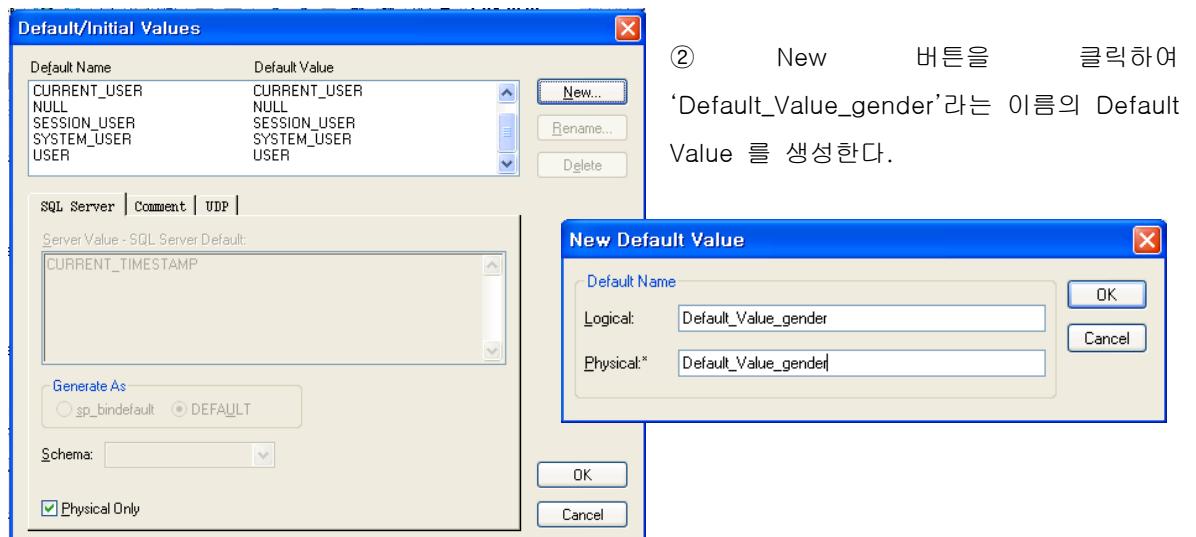


⑥ Columns Editor 를 오픈한 뒤, 성별 컬럼을 선택한 뒤, Valid 콤보 상자에서 '성별_Check'를 선택하여 Check제약 조건을 매칭시킨다.

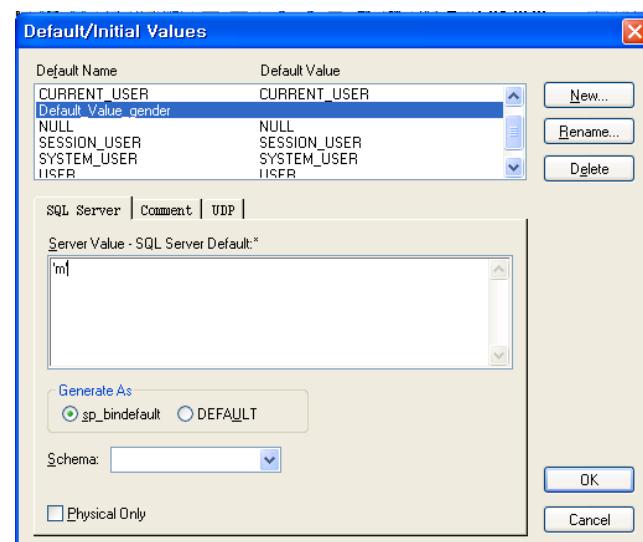
Default값에 대해 정의해 보도록 하자.

위에서 성별이라는 컬럼에 ‘남자’라는 기본 값을 생성해둔다면, 우리가 인서트 절에서 성별 컬럼을 특별히 지정하지 않거나, Default 를 명시했을 때 남자라는 기본값이 사용될 것이다.

- ① Model < Default Value 를 클릭한다. Default / initial Values 대화상자는 기본적으로 Default 값을 정의할 수 있도록 TIMESTAMP나 USER등의 Default Value가 미리 정의되어 있다. 이들을 사용 하려면 원하는 컬럼에 바로 Default 값을 지정하면 될 것이다.



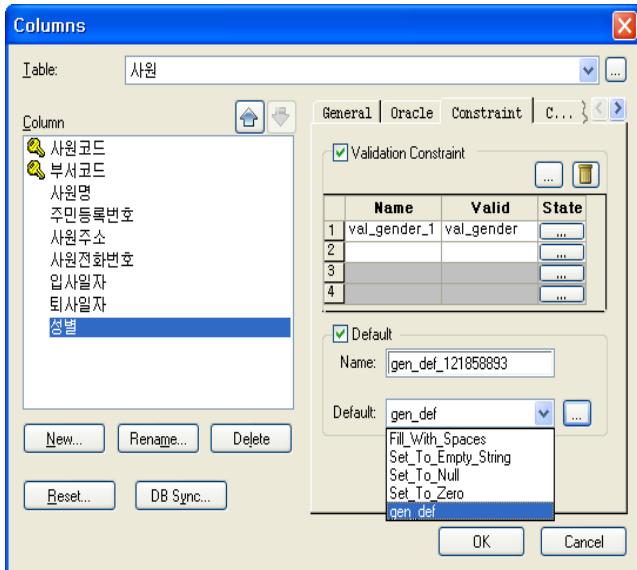
- ② New 버튼을 클릭하여 ‘Default_Value_gender’라는 이름의 Default Value 를 생성한다.



- ③ SQL Server 탭에 ‘m’ 이라고 작성한다. Generate as 의 옵션에서 DEFAULT 옵션을 선택한다.

Generate As에 sp_Bindefault 옵션과 Default 옵션 두 가지가 있다. Sp_Bindefault 방식은 Default 를 데이터베이스 안에 Object로써 생성한 뒤, 여러 개의 컬럼에 바인딩 시키는 방법이다. 이는 여러 개의 컬럼을 대상으로 동일한 Default 속성을 정의하고 할 때 sp_Bindefault 옵션을 선택

하면 된다. Default 옵션은 일반적으로 사용되는 컬럼에 Default 속성을 정의하고 수정하는 방법이다.



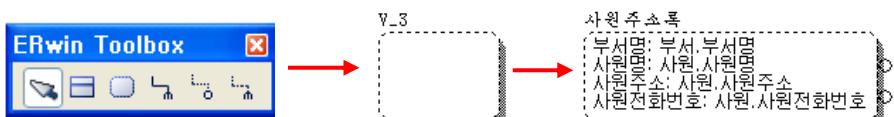
- ④ Columns Editor 를 오픈하여 컬럼을 선택한 뒤, Default 콤보 상자에서 'Default_Value_gender'를 선택한다

3.5 View Table 관리

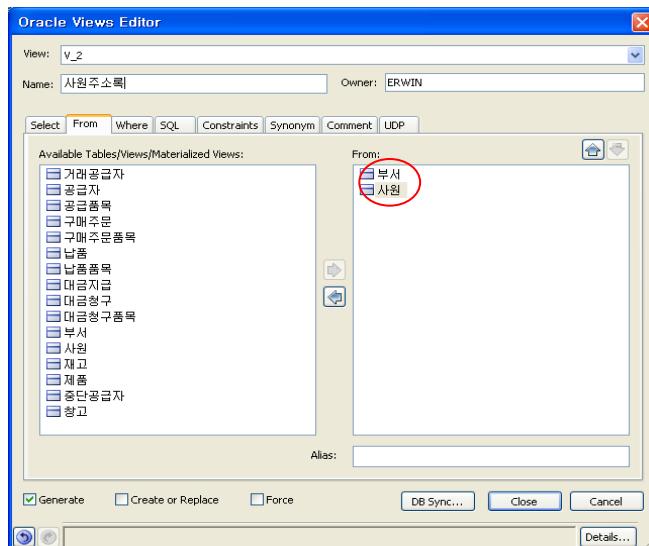
View 란 데이터베이스 사용자들이 접근하도록 테이블을 이용하여 가상 테이블을 구성하는 데이터베이스 오브젝트라는 것을 알고 있을 것이다.

View Table 을 생성하는 방법은 Table 생성하는 방법과 동일하다.

부서와 사원이라는 테이블로부터 ‘사원주소록’이라는 View Table 을 만들어 보도록 하자.



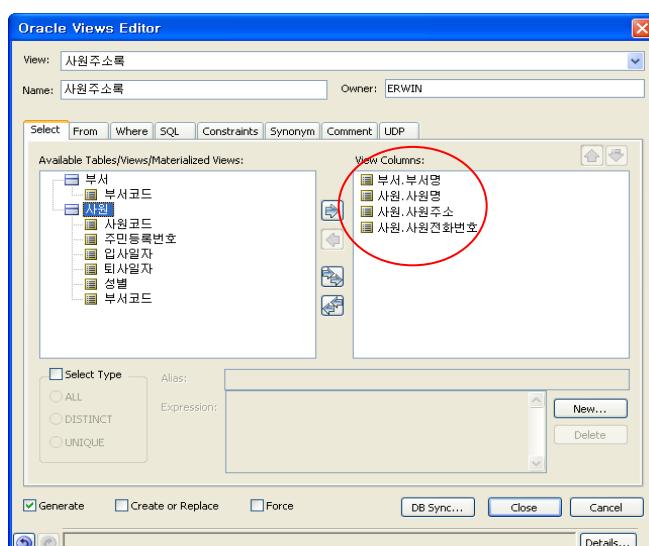
① View 아이콘을 선택하여 새로운 View를 생성 한 후 View Table을 더블 클릭한다.



② View Table의 이름을 변경한다.

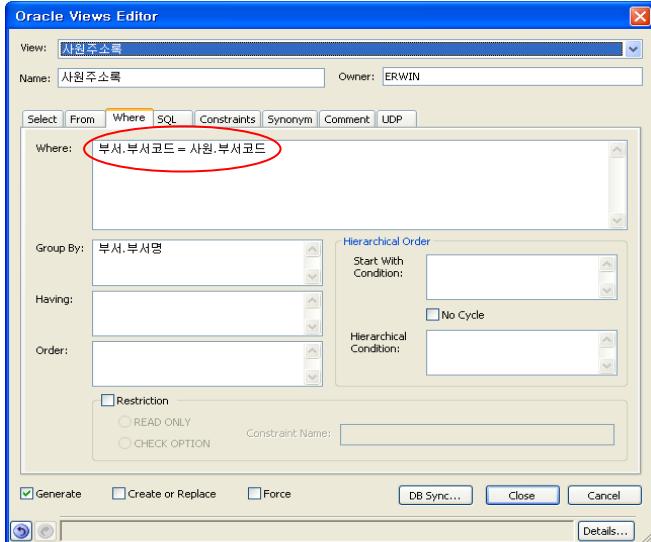
FROM 절에 가져올 TABLE 을 Attach 한다.

FROM 절에 가져올 Table을 Attach 한다.
(FROM 탭에서 뷰에 사용된 테이블들이
보여지게 된다.)



③ SELECT 절에 가져올 Column을 Attach 한다.

(SELECT 탭에서 뷰에 사용된 테이블의 컬럼들이 보여진다. 여기서 뷰로 구성될 컬럼을 선택한다.)

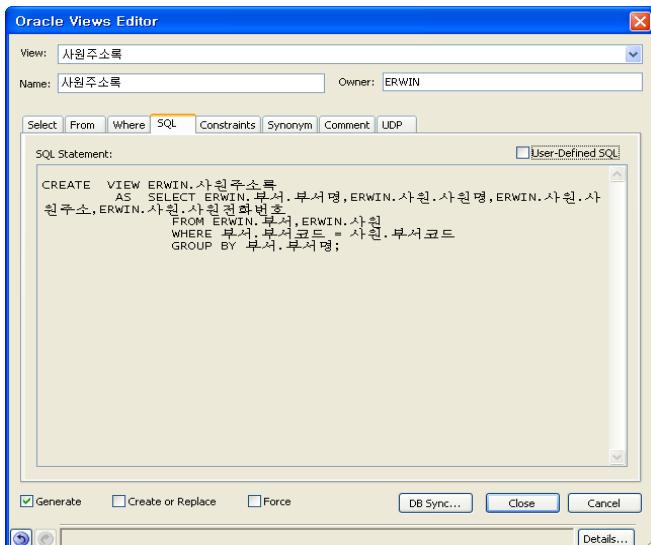


④ WHERE 절에 조건을 입력한다.

WHERE 탭에서 조회의 조건을 지정할 수 있다.

아래와 같은 구문을 작성해보자.

부서.부서코드 = 사원.부서코드



⑤ SQL 탭에서 DDL문을 확인/수정 할 수 있다

SQL 탭에서 완성된 뷰의 문장을 볼 수 있다. User-Defined SQL 옵션에 체크를 하여 사용자가 직접 SQL 문장을 수정할 수 있도록 지원된다.

3.6 모델의 표준화

Names(NSM) 기능은 Data Modeling의 명명 규칙에 따라 사용자가 작성한 용어사전을 이용하여 모델에 써여진 단어를 검사하고, 사전에 정의된 단어나 약어 등을 모델에 적용할 수 있도록 제공되는 기능이다. 따라서 모델에 사용되는 Object Name의 정확성과 일관성을 제공한다.

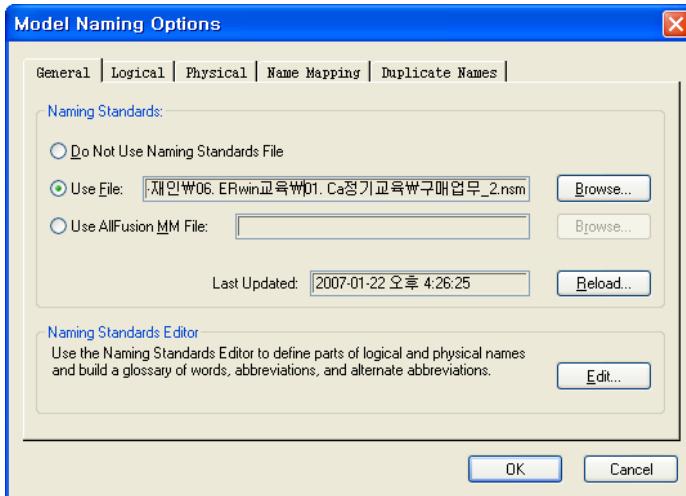
Data Modeler의 용어사전은 .nsm 파일로 저장되며, .nsm 파일을 모델에 매핑하여 사용하게 된다.

Naming Standards 기능은 Logical 모델에서 Physical 모델로 변환 시 물리명을 영문명으로 자동 변환 할 경우 사용 할 수 있다.

1) Names – 명칭 관리

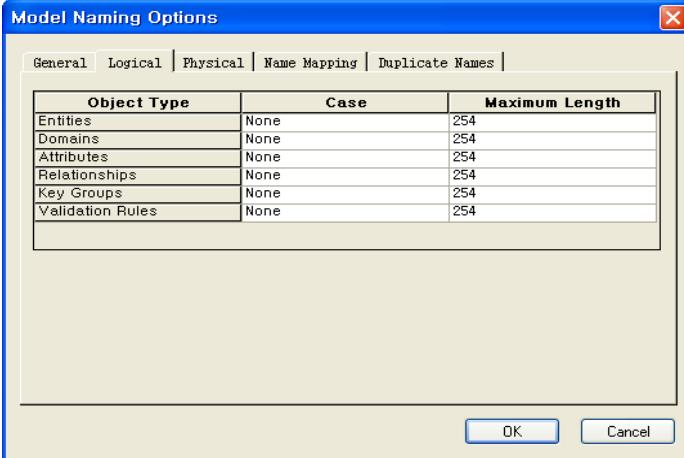
1) Model Naming Option

General 탭에서 사용자가 작성한 용어사전(.nsm) 파일을 매핑 할 수 있다.



* General 옵션 설명 *

항 목	기 능 설 명
Do not User Naming Standards File	nsm 파일을 사용하지 않는다.
Use File	nsm 파일을 모델과 매핑한다. (Browser을 통해 nsm 파일 경로 지정)
Use Allfusion MM File	Model Manager에 저장되어 있는 nsm 파일과 매핑한다.
Reload	nsm 파일을 다시 읽어 들인다.
Edit	Naming Standards Editor를 실행한다. (용어사전 편집)

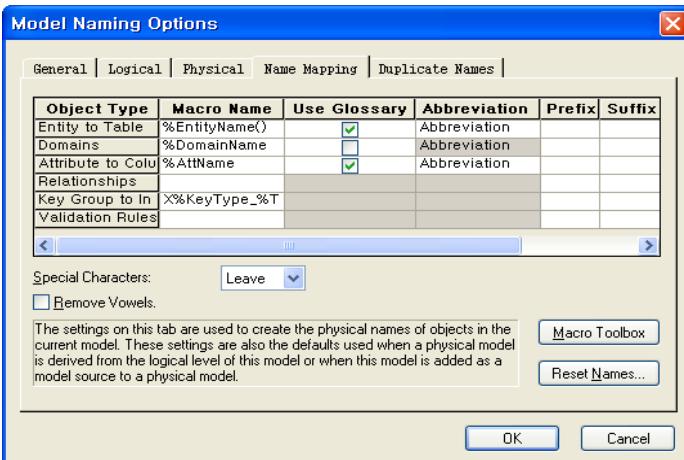


Logical / Physical 탭에서는 Logical 모델에 사용된 Object 명의 대소문자 표현 방법과 최대 길이를 지정 할 수 있다.

[Case Option]

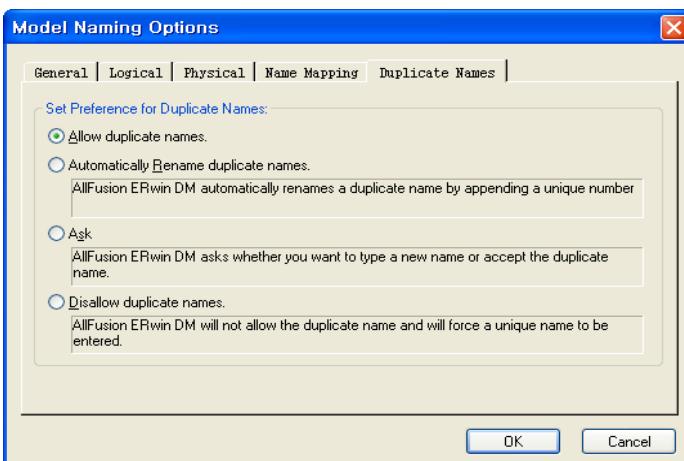
항 목	설 명
UPPER	대문자로 설정
lower	소문자로 설정
Initial	첫 글자만 대문자로 설정

Logical의 Maximum Length는 254byte까지 가능하며, Physical의 경우는 Target DBMS가 받아 들일 수 있는 최대 길이와 동일하다. (Oracle: 30byte, MS SQL: 128byte)



Name Mapping 탭에서는 Object 들의 Name Rule을 지정한다.

항 목	설 명
Macro	해당 Object 들에 대한 Macro를 지정한다.
Use G..	NSM을 매핑한 경우 Object들에 대해서 적용 여부를 체크한다.
Abb..	Abbreviation(영문 풀 네임)인지 Alter Abb..(영문 약어)로 적용할지 선택
Prefix	접두어 설정
Suffix	접미어 설정

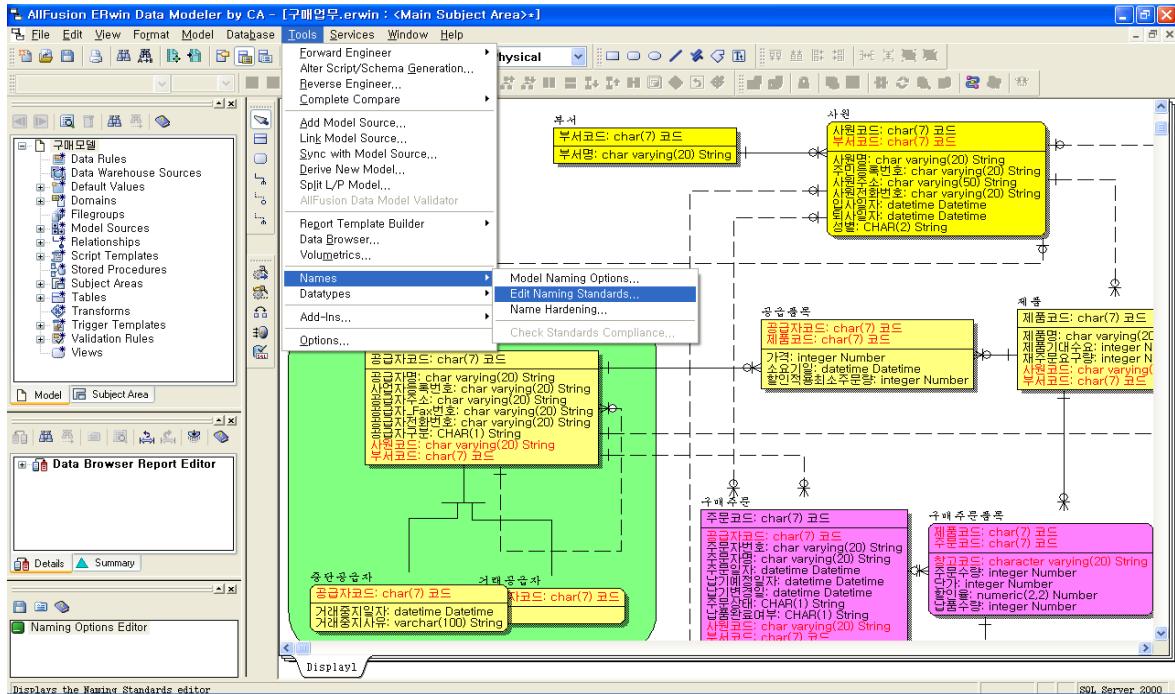


모델에 사용되는 이름이 중복될 경우 처리 옵션을 선택한다.

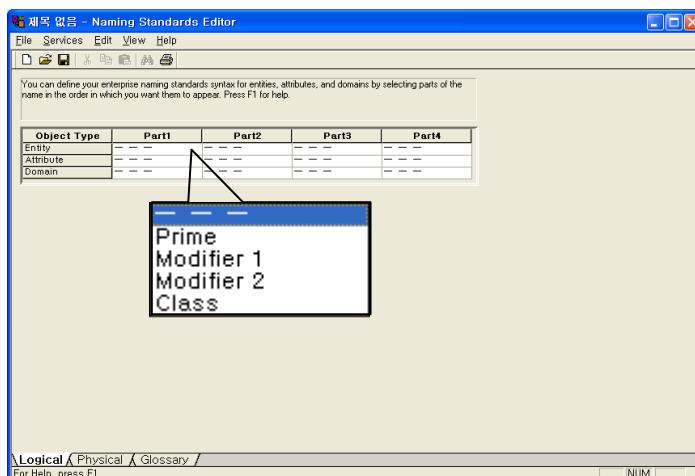
항 목	설 명
Allow	중복을 허용한다.
Auto...	중복되면 일련번호를 붙여 자동으로 Rename 한다.
Ask	사용자가 Rename 한다.
Disallow	중복을 허용하지 않으며 중복되면 E_2와 같이 Default로 이름이 부여된다.

2) Edit Naming Standards (.NSM 파일의 생성)

NSM 기능을 통해 용어사전(.nsm)을 생성하고, 모델과 매핑하여 물리명을 영문으로 변환 해보자.

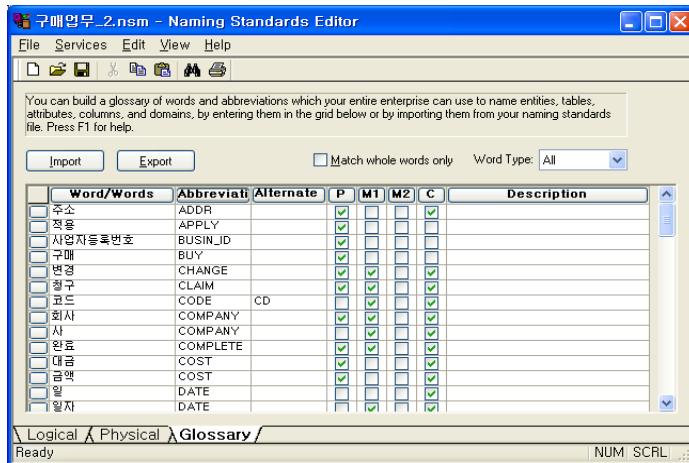


- ① Tools > Names > Edit naming Standards..를 선택한다.



- ② Logical / Physical 탭에서는 Object Name의 명명 규칙을 설정한다. 그러나 특별한 경우를 제외하고는 사용하지 않는다.

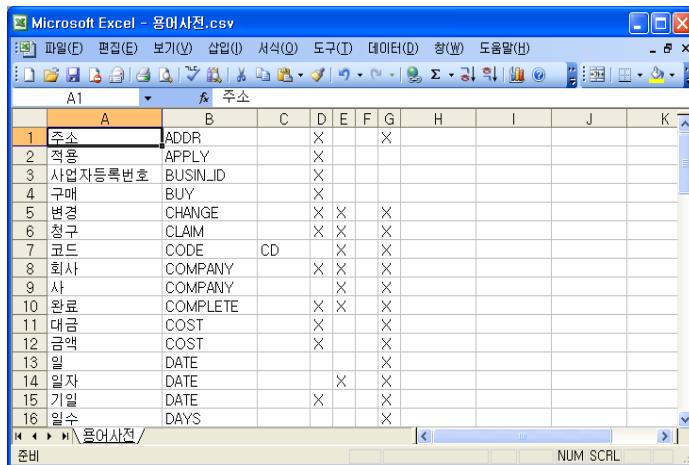
항 목	설 명
Part 1, 2, 3, 4	단어의 나열 순서를 나타낸다.
Prime	중심어, 주제어 (예: account, employee)
Modifier 1, 2	형용사, 부사 혹은 Prime과 Class를 구분할 수 있는 특정 단어
Class	데이터 타입이나 Editor의 주요 종류를 둑어 지칭하는 단어 (date, code, name)



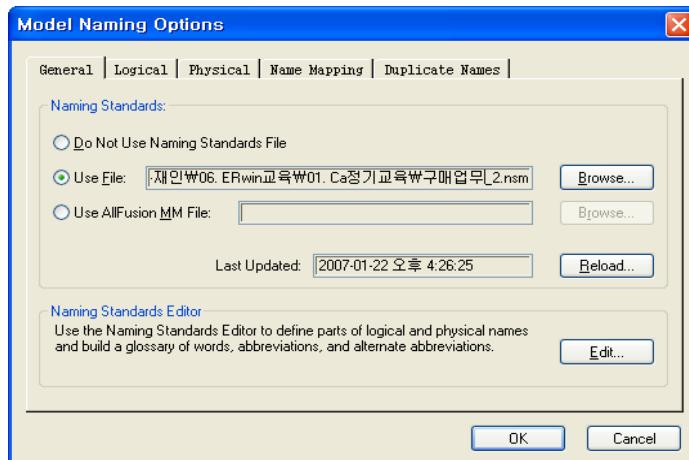
- ③ Glossary 탭에서 용어사전을 등록 한다.

항 목	설 명
Word..	한글명 입력
Abb..	영문 Full Name을 입력
Alterna..	영문 약어를 입력

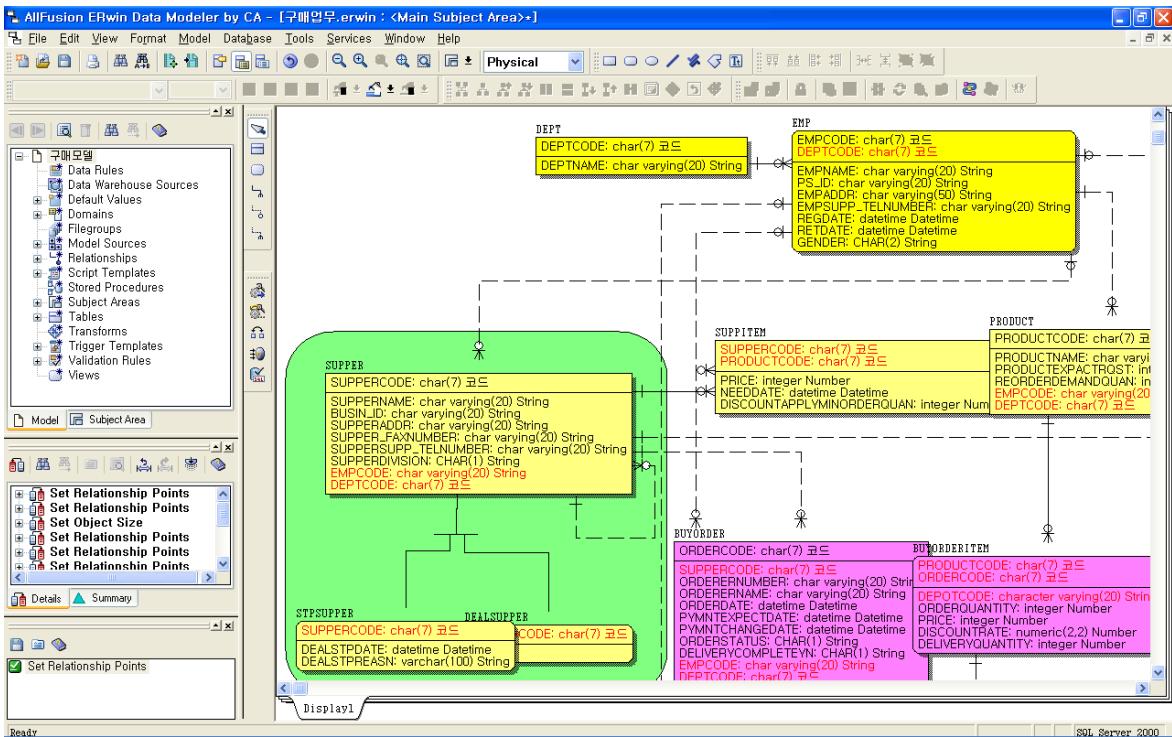
- ④ File > Save나 아이콘을 통해 용어사전을 저장한다.



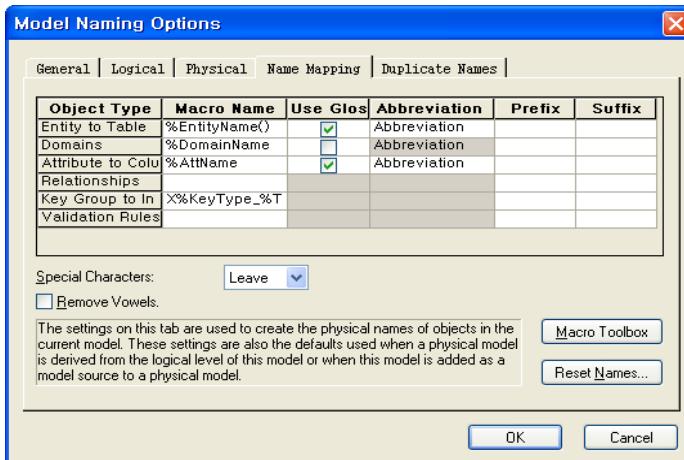
- ⑤ Export를 통해 NSM 정보를 CSV 파일로 생성 할 수 있으며, Import 도 지원한다.



- ⑥ Use File 에서 적용할 Nsm 파일을 선택하고 OK를 버튼을 클릭한다.



⑦ 용어사전에 정의한 대로 물리명이 영문으로 자동변환 된다.



⑧ 만약 NSM 파일을 맵핑 하였는데도
영문명이 변환 되지 않는 경우에는
Model Naming Option에서 Use
Glossary를 다음과 같이 설정한다.
선택한 Object에 대해 용어사전(Use
Glossary)을 사용하겠다는 의미이다.

4. Transform

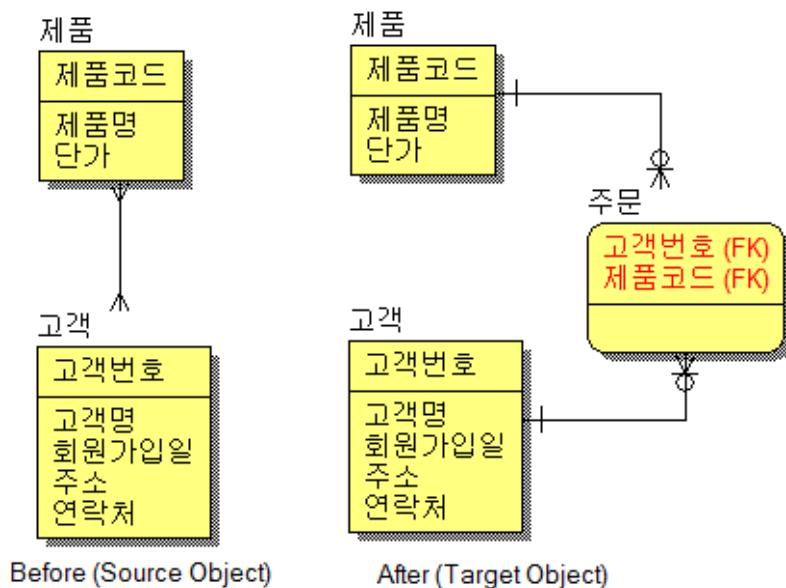
4.1 Transform 정의

모델내의 Object들은 또 다른 구조의 Object Set으로 변환시킬 수 있다. 그러한 변환은 모델 내에서 구조화되고 저장된다. 일반적으로 Layer를 설계할 때 이러한 결정을 내릴 수가 있는데 예를 들어 비 정규화 시 사용된다. ERwin에서는 DBMS의 분산 형태를 고해해서 Entity, Table에 대해 자동으로 변환 할 수 있는 다양한 형태(9가지)의 변환 기능을 제공한다.

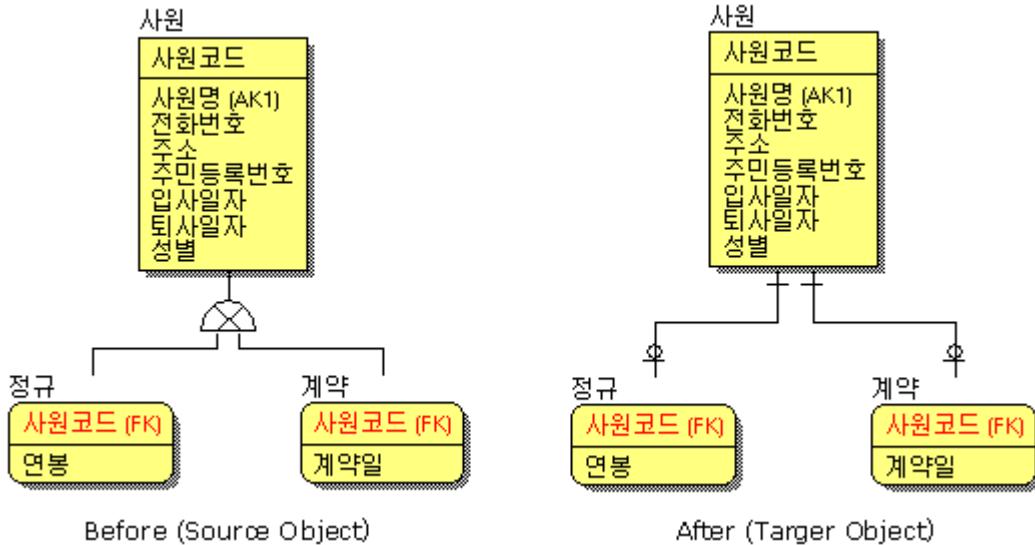
[지원되는 Transforms]

- Many to Many Transform
- Supertype – Subtype Identity
- Supertype – Subtype Roll Up
- Supertype – Subtype Roll Down
- Vertical Partition
- Horizontal Partition
- Roll – Down Denormalization
- Roll – Up Denormalization
- Column Denormalization

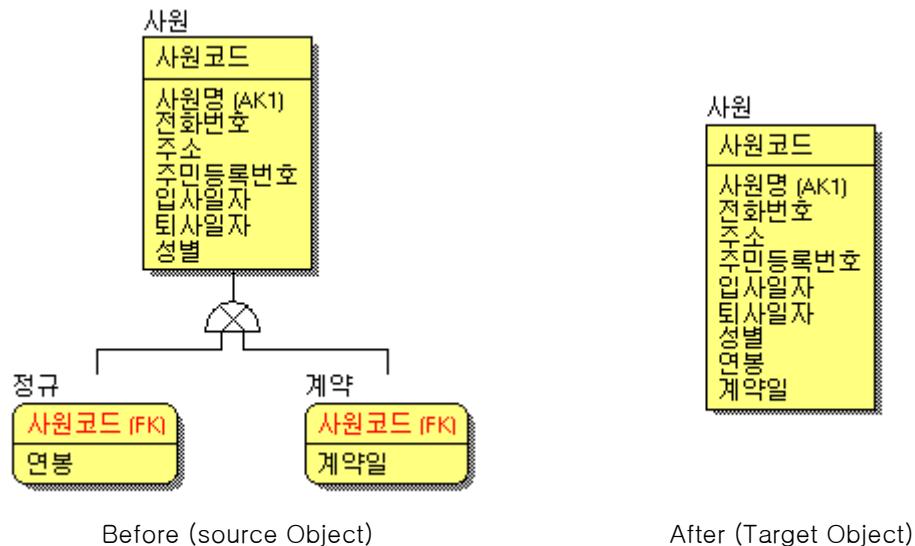
1) Many to Many Transform: M 대 M 관계를 자동으로 변환한다.



2) Supertype – Subtype Identity: 일반화 관계를 부모 – 자식 관계로 자동 변환한다.



3) Supertype – Subtype Roll Up: 일반화 관계에 참여하는 자식 엔티티의 컬럼을 부모로 옮긴다.

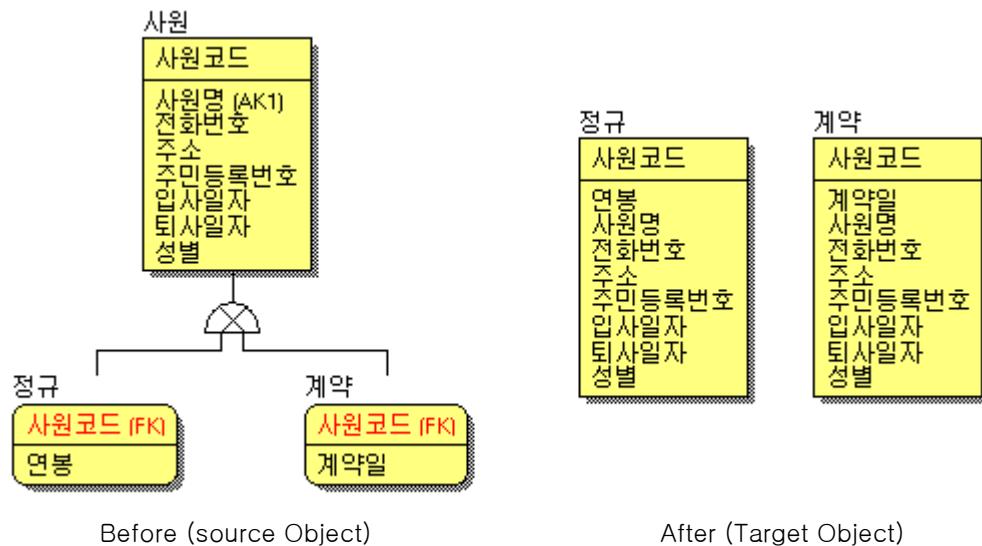


Tip

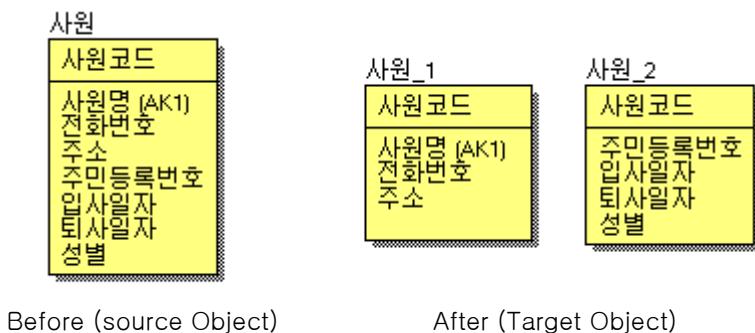
Source Object: : Transform 기능으로 변환되기 전의 Object Set을 의미한다.

Target Object: : Transform 기능으로 변환된 후의 Object Set을 의미한다.

4) Supertype–Subtype Roll Down: 일반화 관계에 참여한 부모 엔티티의 컬럼을 자식으로 내린다.



5) Vertical Partition: 해당 Table을 수직분할 한다.



6) Horizontal Partition: 해당 Table을 수평분할 한다.



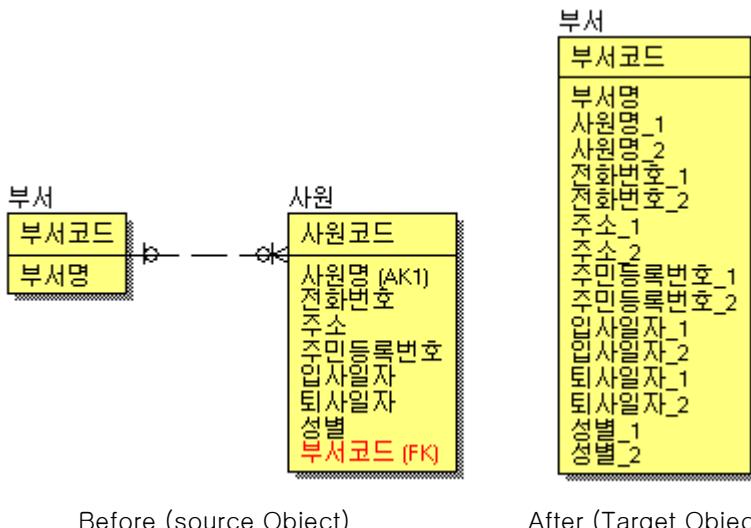
7) Roll – Down Denormalization: 부모 테이블의 Column이 자식 Table로 이동한다. (역정규화)



Before (source Object)

After (Target Object)

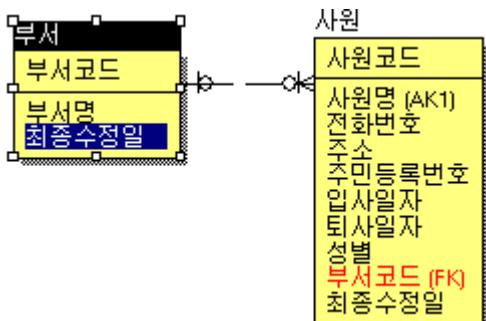
8) Roll – Up Denormalization: 자식 테이블의 Column이 부모 Table로 이동한다. (역정규화)



Before (source Object)

After (Target Object)

9) Column Denormalization : 한 테이블이 소유한 특정 컬럼이 다른 테이블로 복사된다.

**Tip**

Column Denormalization에 의해 복사된 컬럼은 원본 컬럼과 연결되어 있다. 원본 컬럼을 수정하면 복사된 컬럼에도 수정 사항이 반영 된다. 하지만 복사된 컬럼을 수정하게 되면 원본과의 연결이 끊어지게 된다. 이렇게 되면 원본을 수정하여도 수정사항이 반영되지 않는다는 것을 주의하자.

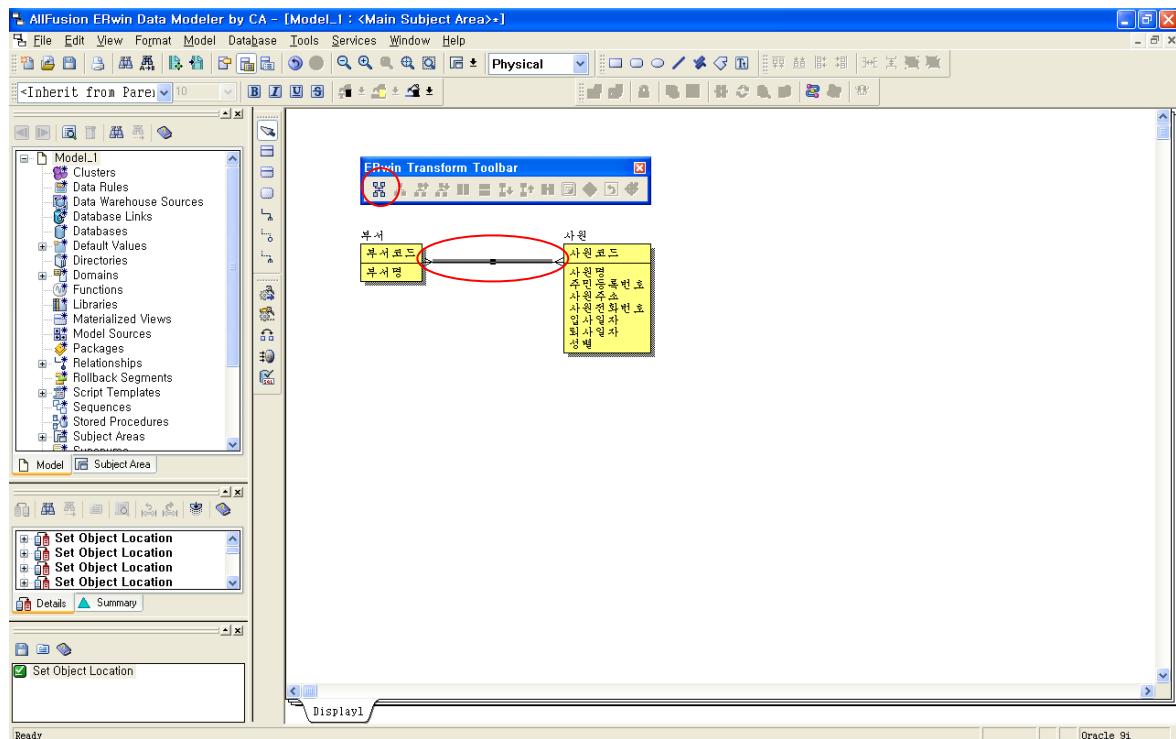
4.2 Transform 기본 원리

Transform 작업을 하려면 반드시 변환시킬 Source Object를 선택해야 한다. 그리고 나서 원하는 Transform Tool을 선택해야 한다. 그러면 Wizard가 실행되며 절차에 따라 진행하면 된다.

변환이 완료되면 Source Object는 모델 내에 숨겨지며 Target object로 대체된다. Target Object가 Source Object와 거의 똑같이 보이지만, 그것은 실제로 Transform 기능에 의해 새롭게 생성된 Object이다. (Column Copy Transform은 제외)

Object의 변환 정보는 ERD상에서는 확인 할 수 없다. 그러나 Model Explorer를 통해서 Transform 작업 리스트 및 상세 정보를 확인 할 수 있으며, 모델러는 작업을 되돌리거나 고정시킬 수 있다.

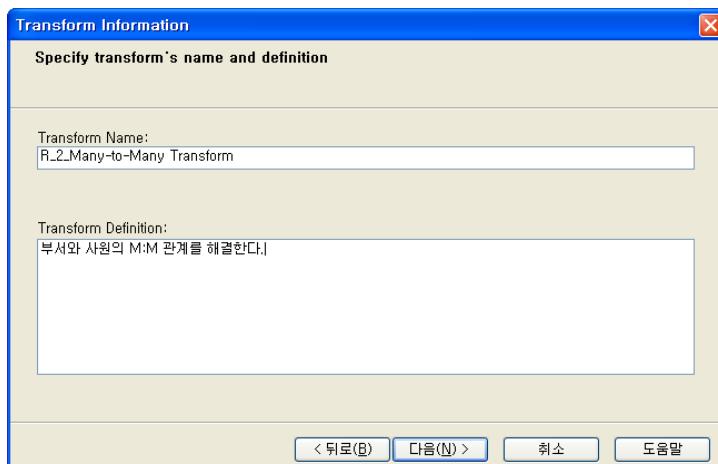
1) Many to Many Transform을 이용한 M : M 관계의 자동변환



- ① M : M 관계선을 선택하면 Many to Many Transform 아이콘을 활성화 된다. 이를 클릭하거나 관계선을 선택하고 마우스 오른쪽 버튼을 클릭하여 메뉴에서 Create Association Table을 선택한다.



② Many to Many Transform Wizard 가 실행된다. 여기서 다음 버튼을 클릭한다.



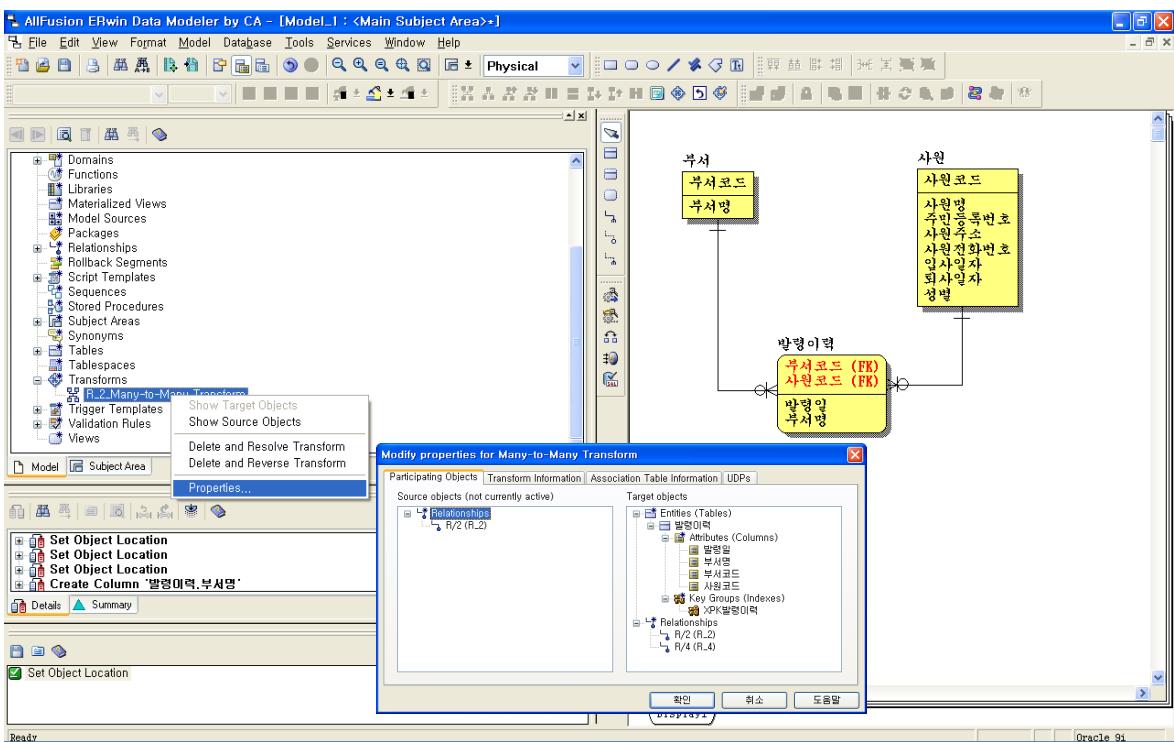
③ Transform의 이름과 Definition을 입력한다.(Model Explorer에서 확인 가능한 정보)



④ Many to Many를 해결할 때 자동으로 생성되는 연관 엔티티 이름 및 Definition을 입력한다.



⑤ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



⑥ Model Explorer에서 리스트를 확인 할 수 있으며 Properties에서 세부 정보를 수정할 수 있다.

Tip

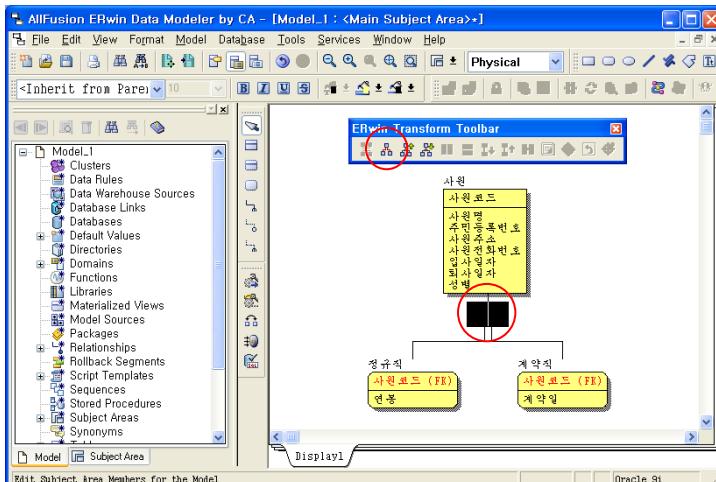
복사 : 변환 전의 Source Object 보기

복사 : 변환 후 Target Object 보기

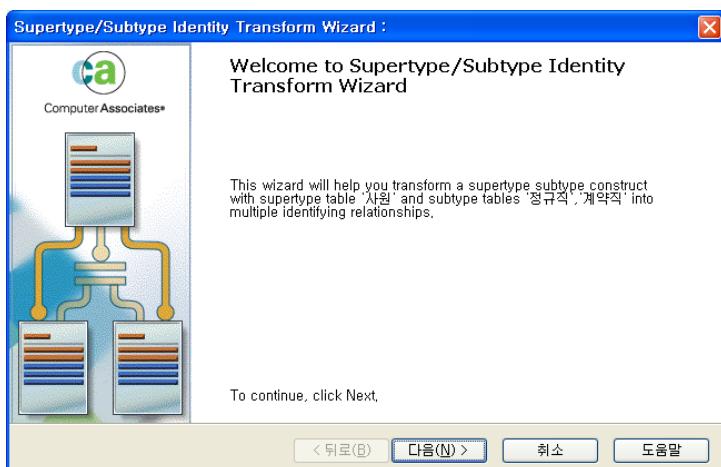
복사 : 변환 작업을 취소한다.

복사 : 변환 내용으로 모델을 고정시킨다. (Source Object를 확인 할 수 없다.)

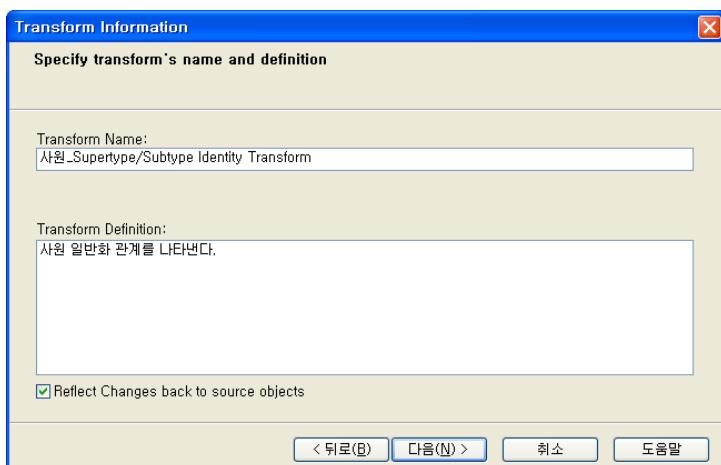
2) Supertype – Subtype Identity을 이용한 일반화 관계의 자동변환



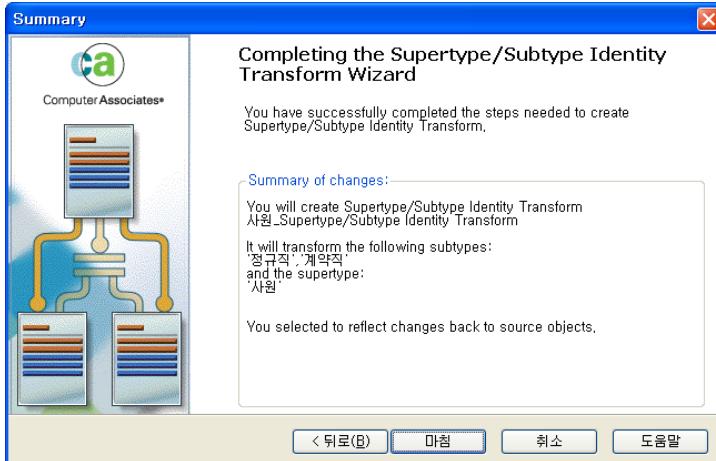
① 일반화 관계 Symbol을 선택하고 Supertype – Subtype Identity 아이콘을 선택한다.



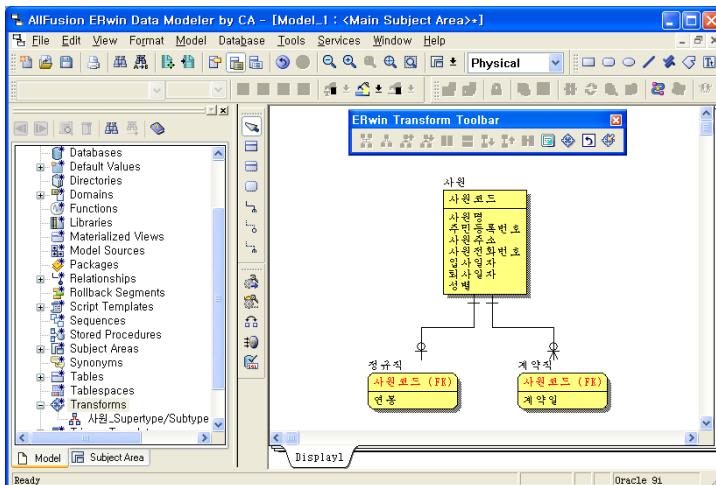
② Supertype / Subtype Identity Transform Wizard가 실행된다. 여기서 다음 버튼을 클릭한다.



③ Transform의 이름과 Definition을 입력한다.
Reflect Changes back to source objects Check옵션은 Target Entity / Table에서 변환되는 내용이 Source Object에 반영되게 하는 옵션이다.

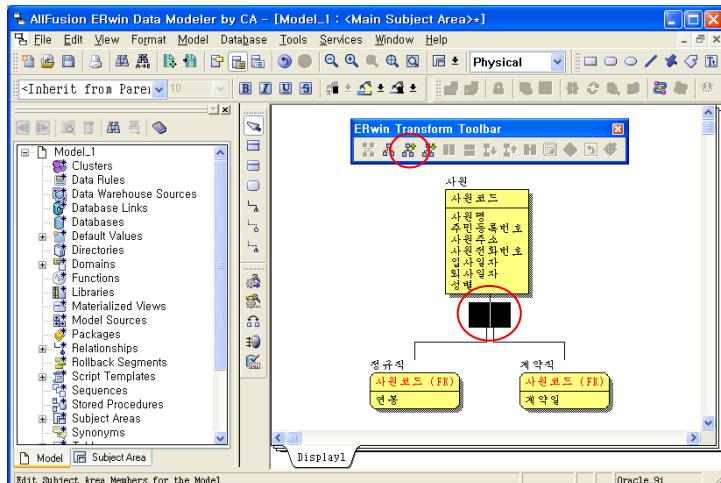


④ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



⑤ 일반화 관계가 부모 – 자식 관계로 변환된 것을 확인 할 수 있다.

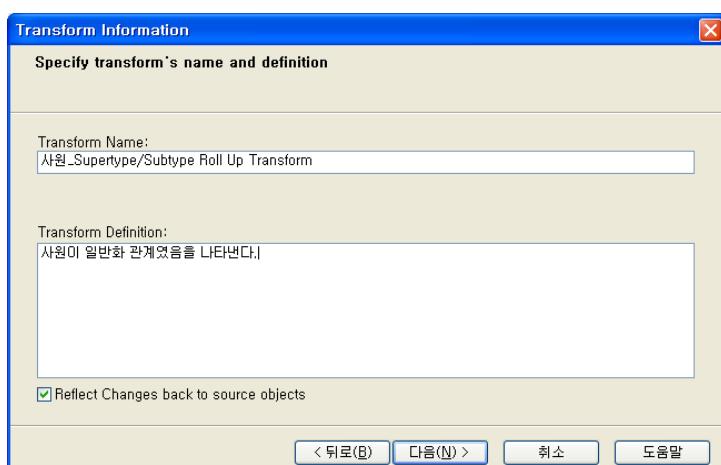
3) Supertype – Subtype Roll Up을 이용한 일반화 관계의 자동변환



① 일반화 관계 Symbol을 선택하고 Supertype – Subtype Roll UP 아이콘을 선택한다.



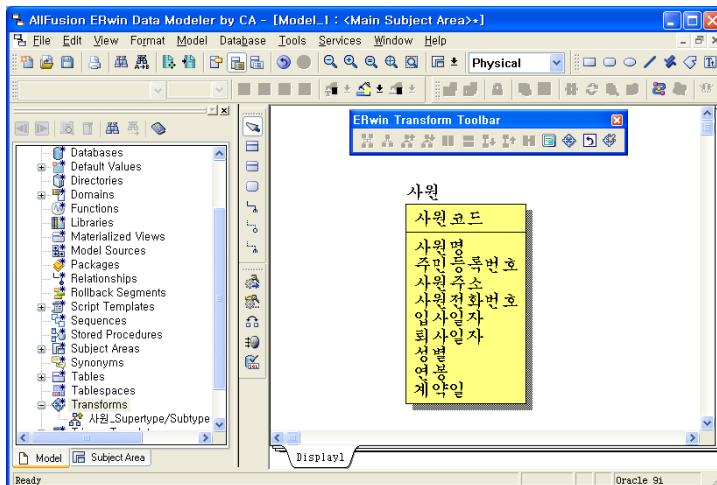
② Supertype / Subtype Roll UP Transform Wizard가 실행된다. 여기서 다음 버튼을 클릭한다.



③ Transform의 이름과 Definition을 입력한다.

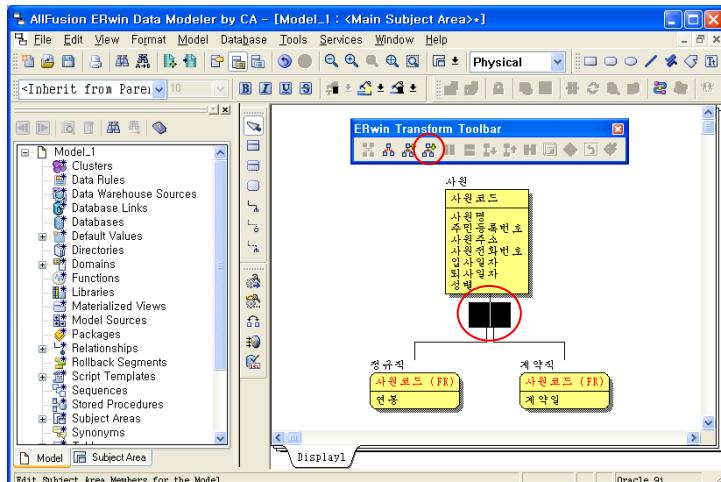


④ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.

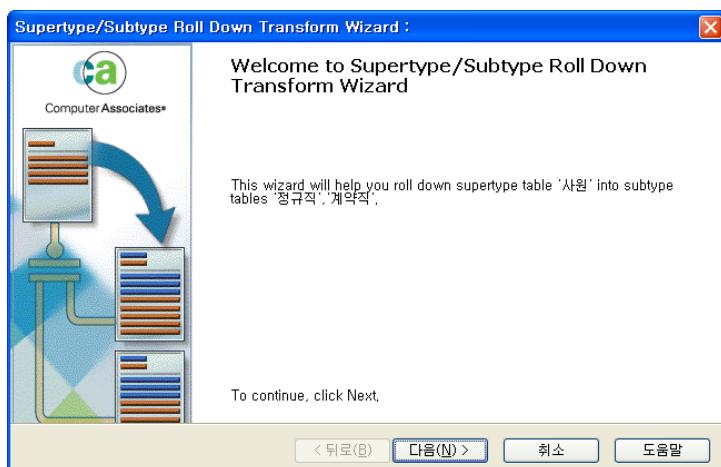


⑤ 일반화 관계의 하위(자식) 엔티티의 속성들이 상위(부모) 엔티티의 속성으로 포함된다.

4) Supertype – Subtype Roll Down을 이용한 일반화 관계의 자동변환

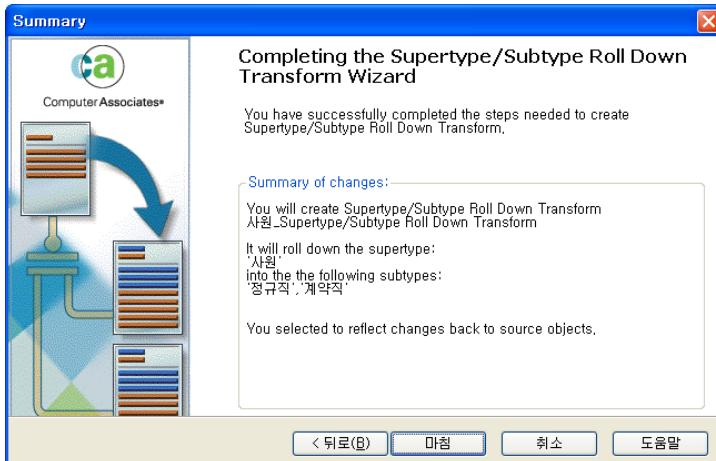


① 일반화 관계 Symbol을 선택하고 Supertype – Subtype Roll Down 아이콘을 선택한다.

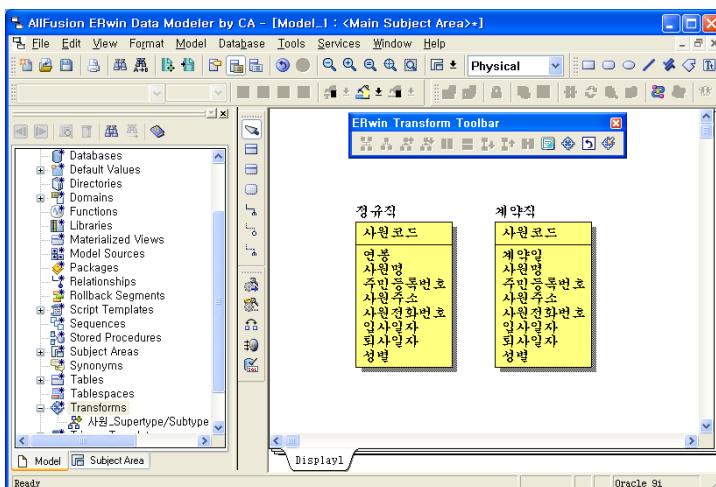


② Supertype / Subtype Roll Down Transform Wizard가 실행된다. 여기서 다음 버튼을 클릭한다.

③ Transform의 이름과 Definition을 입력한다.



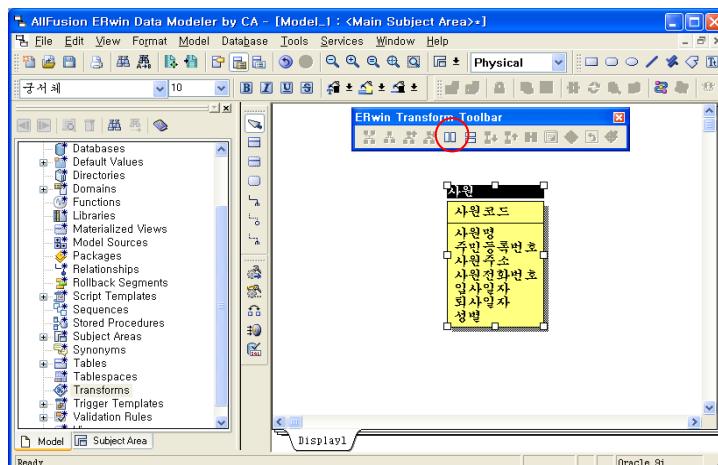
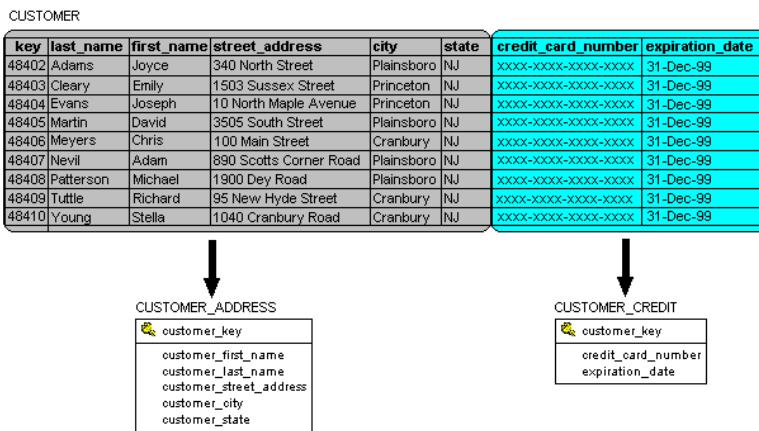
④ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



⑤ 일반화 관계의 상위(부모) 엔티티의 속성들이 하위(자식) 엔티티의 속성으로 포함된다.

5) Vertical Partition을 이용한 엔티티의 수직적 분할

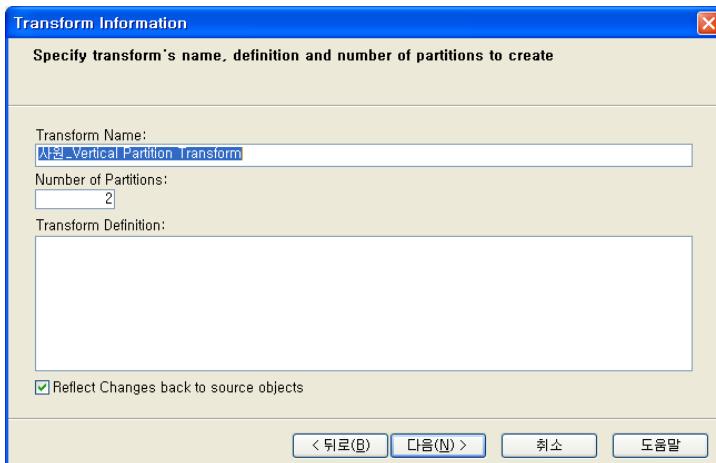
Target Table의 컬럼 형태는 Source Table에서 분리된 컬럼의 Set이다. 한 테이블의 속성이 많이 존재하고 이 중 자주 조회하는 컬럼들이 있을 경우, 자주 조회하는 컬럼들만 또 다른 테이블로 분리하려고 할 때 많이 사용된다. 테이블 이름은 디폴트로 설정 했을 때 Source table 명 + 일련번호가 붙는 형태이다.



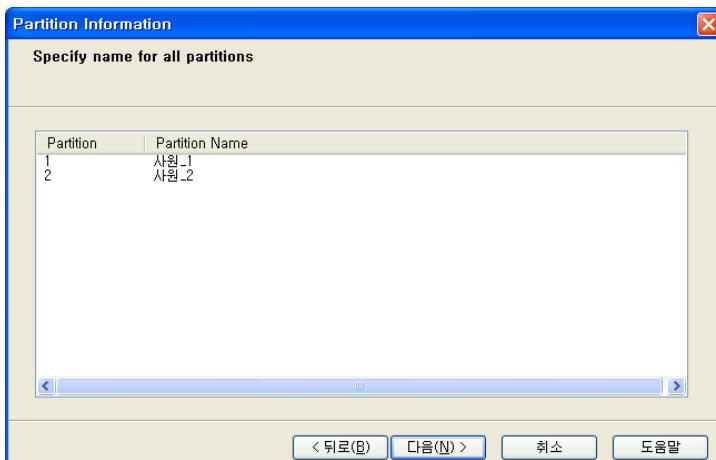
- ① 엔티티를 선택하고 Vertical Partition 아이콘을 선택한다.



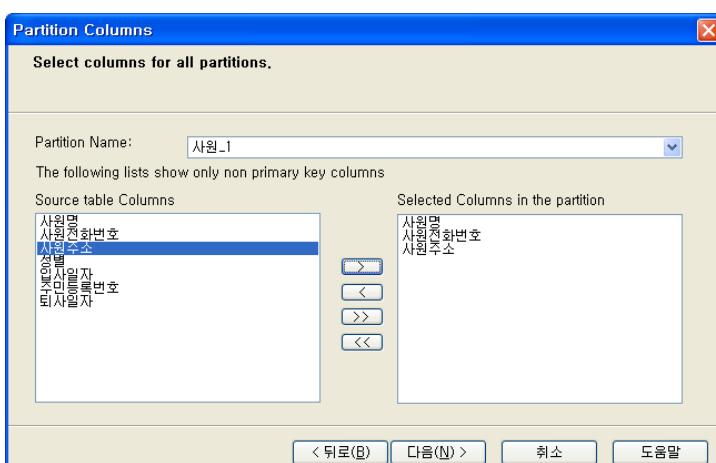
- ② Vertical Partition Transform Wizard가 실행된다
여기서 다음 버튼을 클릭한다.



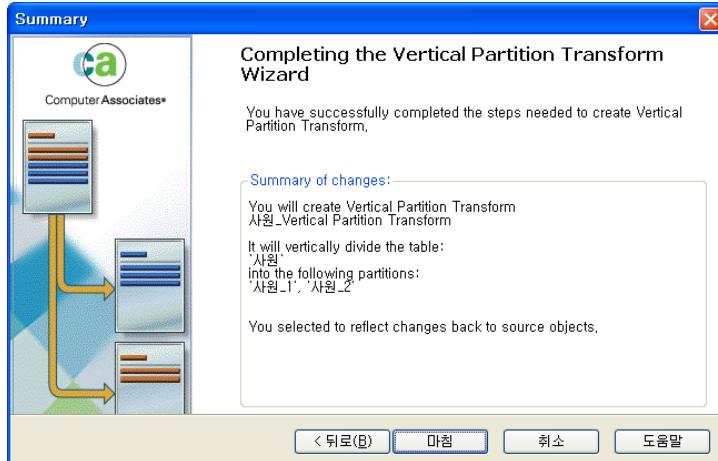
③ Transform의 이름과 분할할 데이터 개수, Definition을 입력한다.



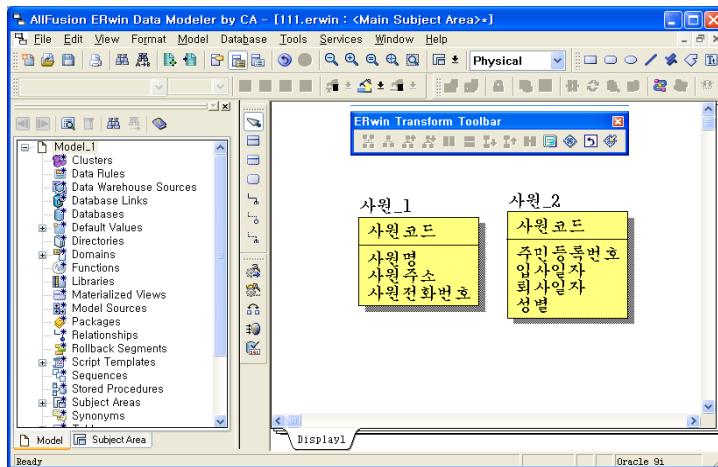
④ Partition Name이 기본적으로 생성되며 더블 클릭을 통해 수정이 가능하다.



⑤ Partition Table에 포함될 속성을 Attach 한다. (Partition Table 모두)



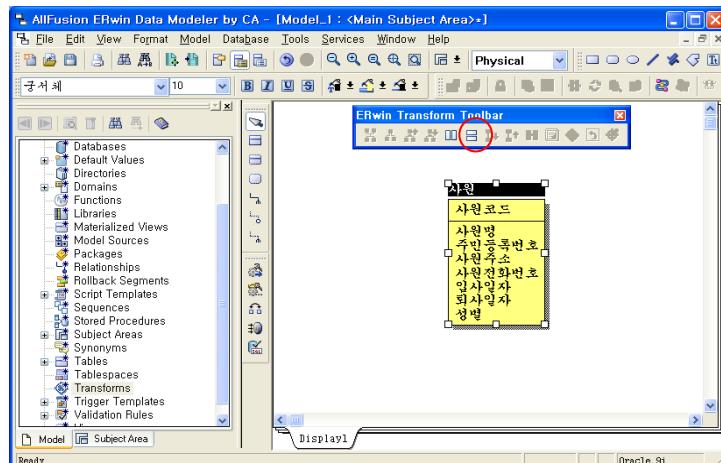
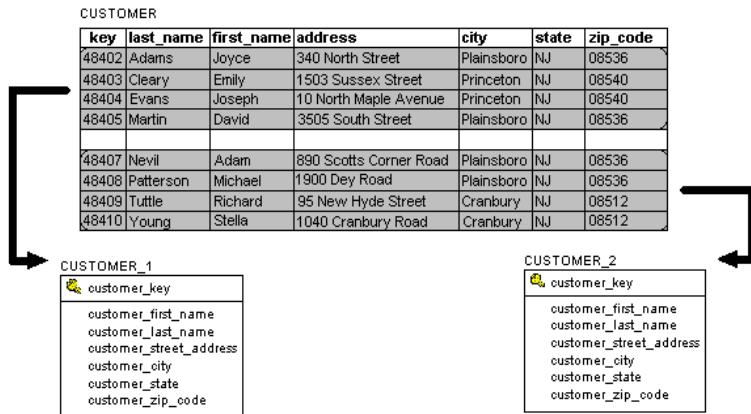
⑥ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



⑦ 테이블이 분할된 것을 확인 할 수 있다.

6) Horizontal Partition을 이용한 엔티티의 수평적 분할

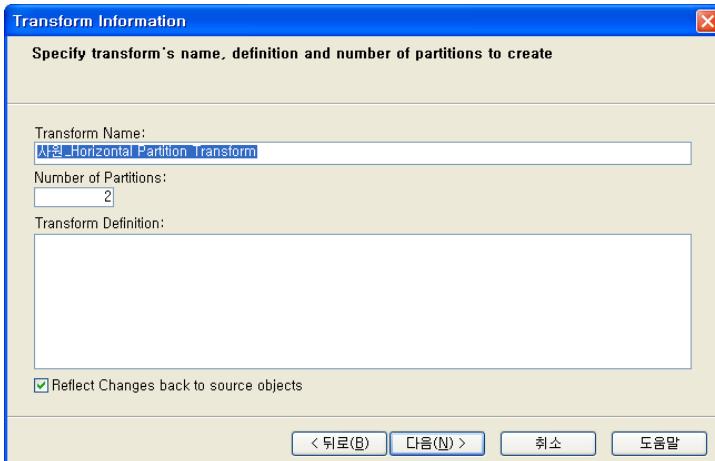
테이블의 한 세트로 테이블의 개수는 원하는 대로 생성할 수 있다. 테이블 이름은 디폴트로 설정 했을 때 Source Table 명 + 일련번호가 붙는 형태이다. Target Table의 컬럼 형태는 Source Table과 동일한 컬럼 Set이며, 나뉘어진 테이블에 Row의 분리규칙을 입력할 수 있다.



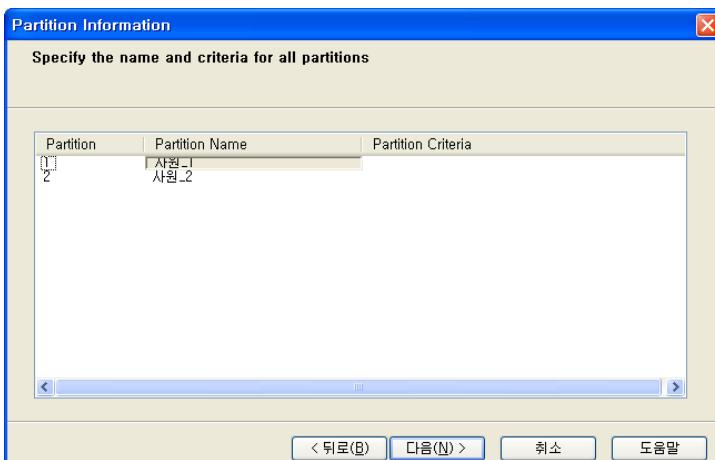
- ① 엔티티를 선택하고 Horizontal Partition 아이콘을 선택한다.



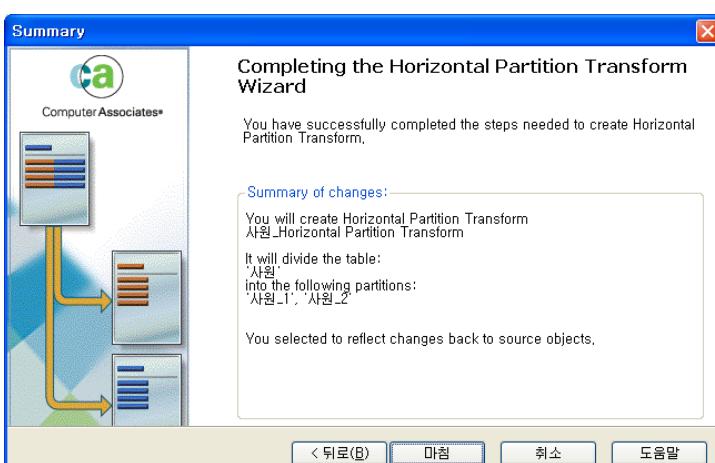
- ② Horizontal Partition Transform Wizard가 실행된다.
여기서 다음 버튼을 클릭한다.



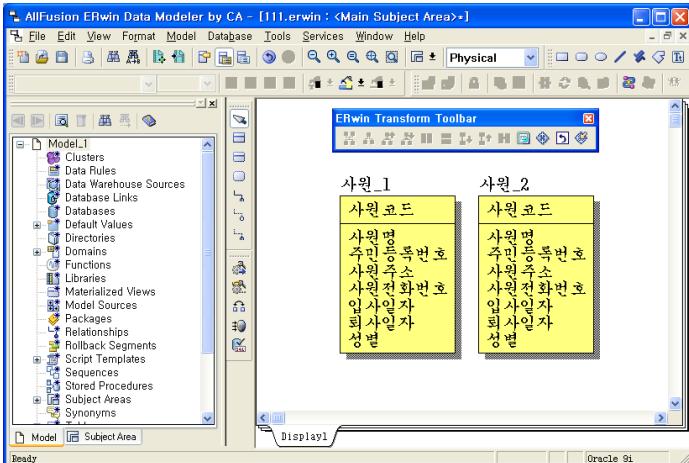
③ Transform의 이름과 분할할 테이블 개수, Definition을 입력한다.



④ Partition Name이 기본적으로 생성되며 더블 클릭을 통해 수정이 가능하다.



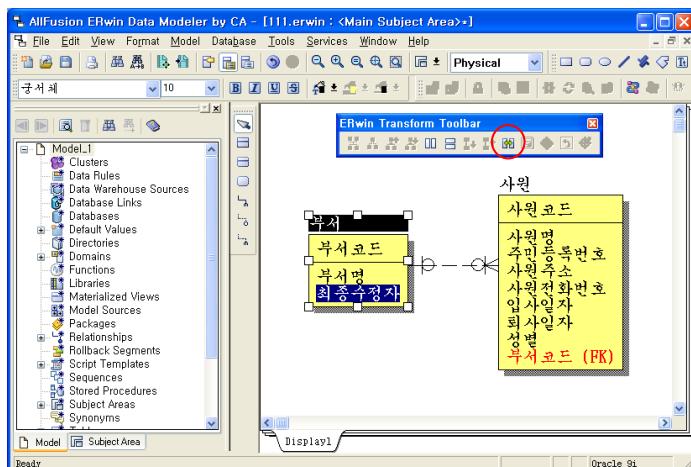
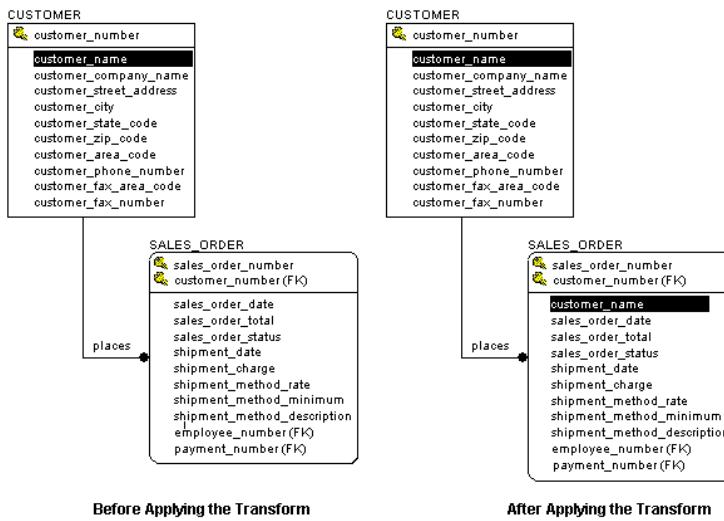
⑤ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



⑥ 테이블이 분할된 것을 확인 할 수 있다.

7) Column Denormalization을 이용한 Column 복사

하나의 컬럼으로 Source Column에서 복사해 온 것이다. 이름과 속성은 Source Column과 동일하다.



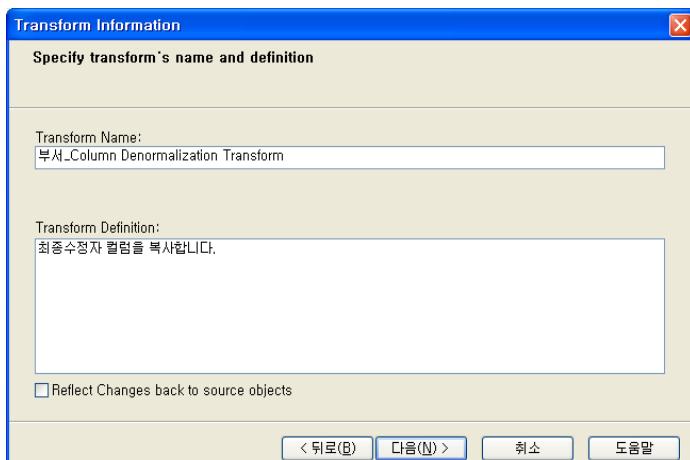
- ① 복사할 컬럼을 선택하고 Column Denormalization 아이콘을 선택한다.



② Column Denormalization Transform

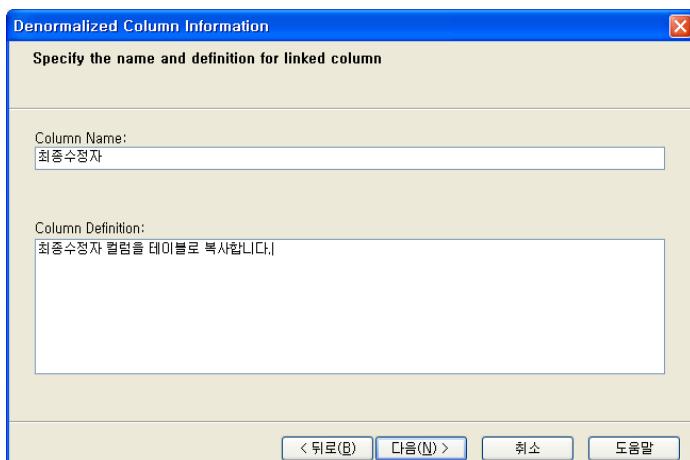
Wizard가 실행된다.

여기서 다음 버튼을 클릭한다.



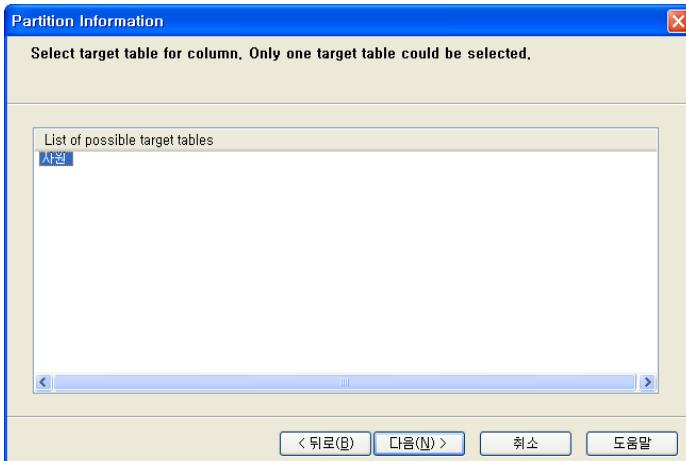
③ Transform의 이름과 Definition을 입력한다.

(Reflect Changes back to source objects Check 옵션을 check하면 Target Attribute가 수정될 때 Source Attribute가 같이 변경되며, 옵션이 Check되지 않으면 Source Attribute가 수정될 때에만 Target Attribute를 동기화 해준다.)

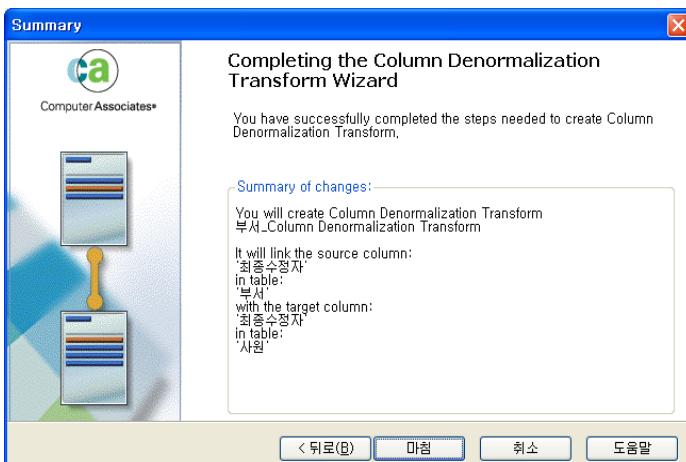


④ Attribute 이름은 자동 입력된다.

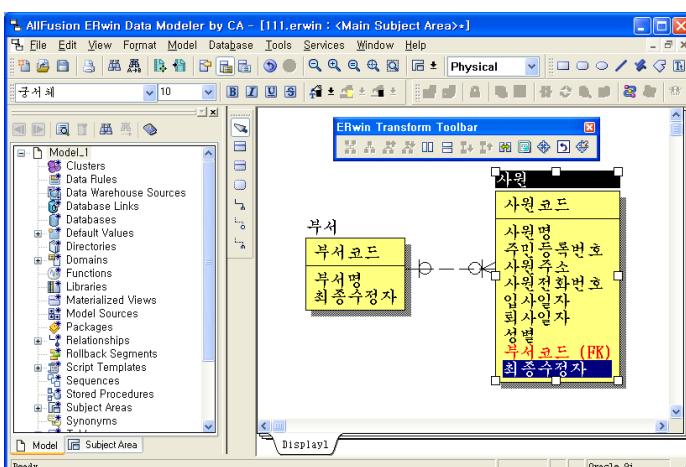
Definition을 입력한다.



⑤ Target Entity를 선택한다.



⑥ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.

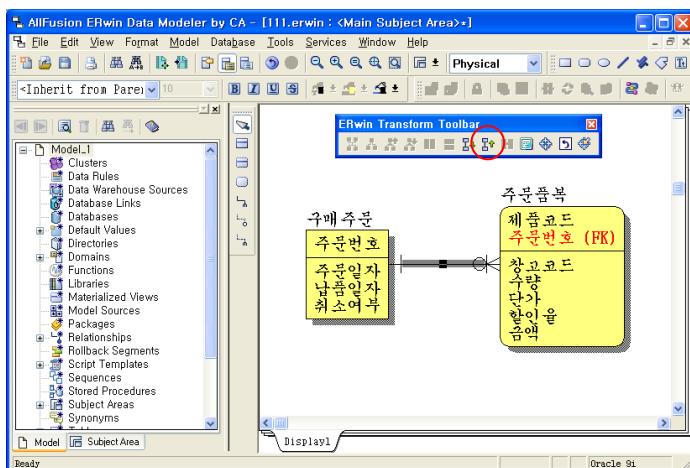


⑦ Attribute가 복사된 것을 확인 할 수 있다.

이렇게 컬럼을 복사 한 경우 원본 컬럼을 변경하면 변경내용이 복사된 컬럼에 동일하게 적용된다.

8) Roll – Up Denormalization을 이용한 Table 역정규화

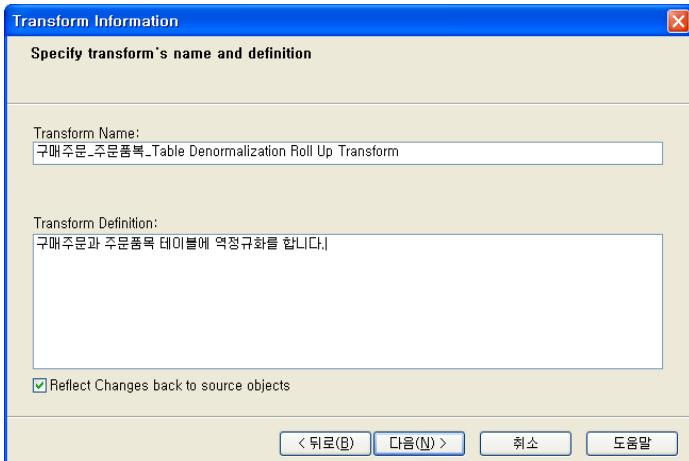
관계를 맺고 있는 두 테이블이 정규화 되었을 때 데이터 검색 속도가 느린 현상이 발생하는 경우가 발생한다. 이는 테이블간에 서로 조인을 걸어야 하기 때문에 발생하는 것으로 이를 방지하기 위해 테이블을 역정규화 할 수 있다. Roll – Up Denormalization은 자식 엔티티의 속성들을 부모 엔티티로 포함시켜 역정규화를 진행한다.



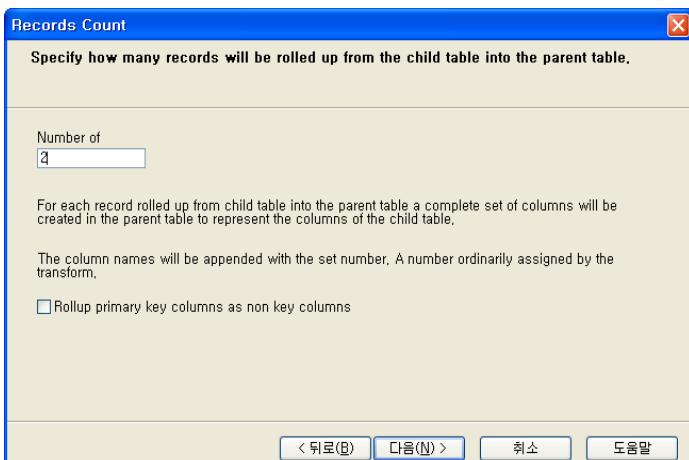
- ① 관계가 있는 두 테이블을 선택하고 Roll-Up Denormalization 아이콘을 선택한다.



- ② Roll-Up Denormalization Transform Wizard가 실행된다.
여기서 다음 버튼을 클릭한다.



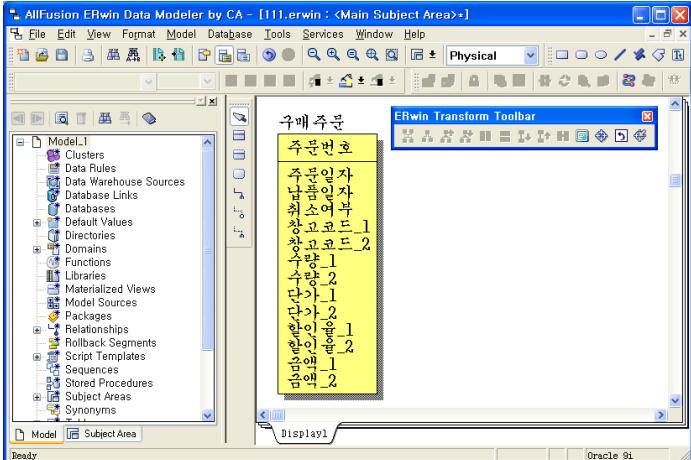
- ③ Transform의 이름과 Definition을 입력한다.



- ④ Target 테이블에 반복되는 레코드의 수를 입력한다.



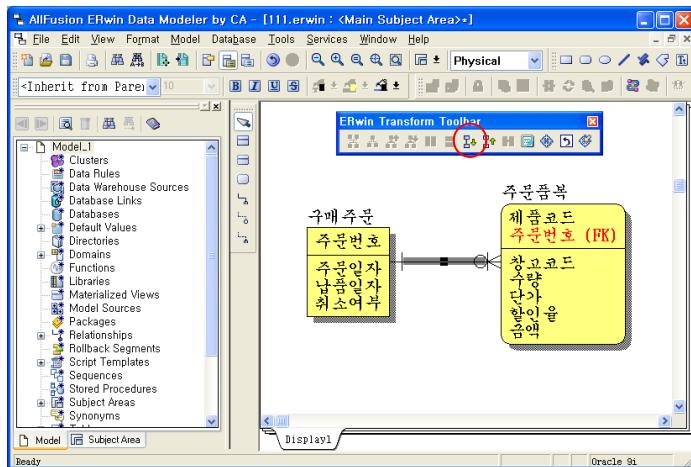
- ⑤ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



⑥ 주문품목의 컬럼들이 구매주문으로 포함된 것을 확인 할 수 있다. (부모 테이블로 이동)

9) Roll – Down Denormalization을 이용한 Table 역정규화

관계를 맺고 있는 두 테이블이 정규화 되었을 때 데이터 검색 속도가 느린 현상이 발생하는 경우가 발생한다. 이는 테이블간에 서로 조인을 걸어야 하기 때문에 발생하는 것으로 이를 방지하기 위해 테이블을 역정규화 할 수 있다. Roll–Down Denormalization은 부모 엔티티의 속성들을 자식 엔티티로 포함시켜 역정규화를 진행한다.



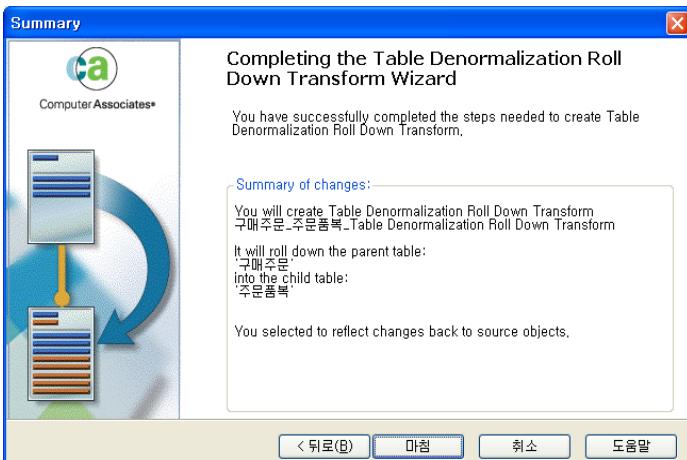
- ① 관계가 있는 두 테이블을 선택하고 Roll – Down Denormalization 아이콘을 선택한다.



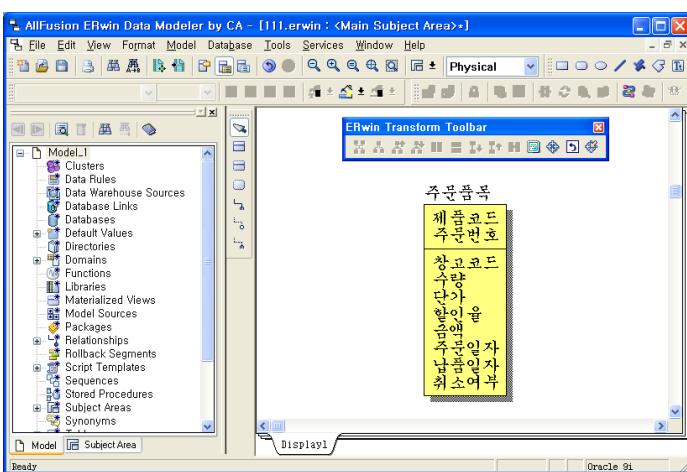
- ② Roll– Down Denormalization Transform Wizard가 실행된다. 여기서 다음 버튼을 클릭한다.



- ③ Transform의 이름과 Definition을 입력한다.



- ④ 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



- ⑤ 구매주문의 컬럼들이 주문품목으로 포함된 것을 확인 할 수 있다. (자식 테이블로 이동)

5. Forward Engineer / Reverse Engineer / Complete Compare

5.1 Forward Engineer

ERwin Data Modeler는 생성한 모델의 Object(Table, Column, Index, Validation Rule, Default 등)에 대해서 사용자가 정의한 옵션에 따라 DDL 문장을 생성 할 수 있으며, DBMS로 직접 스키마를 생성 할 수 있도록 지원한다. Forward Engineer 에서는 DDL을 생성하고 스키마를 생성 해주는 Schema Generation과 모델에 대해서 검사를 하는 Check Model 기능을 제공한다.

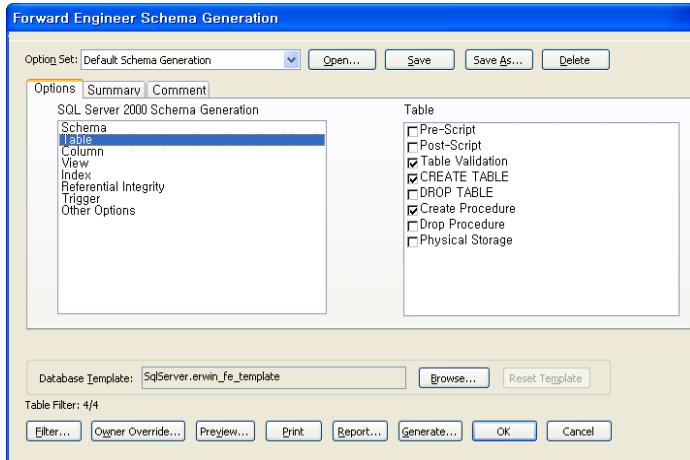
1) Schema Generation

Schema Generation 기능은 모델의 Object들을 사용자가 정의한 옵션에 따라 DDL 문장을 생성한다. DDL 문장은 모델 내 입력된 정보와 사용자가 정의한 Generation Option에 의존하여 생성된다. 따라서 ERwin에서 스키마를 생성할 때는 모든 항목이 Option 이라는 사실을 주의해야 한다.

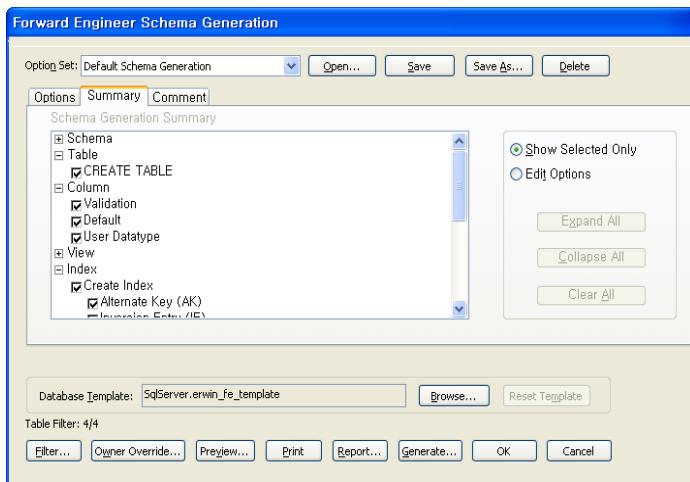
2) Generation Option 설명 (MS SQL)

항 목	기 능 설 명
<input checked="" type="checkbox"/> Schema	모델 전체 즉 스키마 레벨의 옵션으로 사용자가 정의한 Rule, Default Value, 스크립트와 프로시저 생성에 관한 옵션
<input checked="" type="checkbox"/> Table	Table 레벨의 Validation, DBMS가 지원하는 Physical Storage와 Table에 대한 스크립트와 프로시저 생성 옵션이다
<input checked="" type="checkbox"/> Column	Column 생성 옵션으로 Validation, Default Value, User Datatype 그리고 Physical order 가 있다. ERwin은 Column에 대한 오더(순서)를 두 가지로 관리하고 있다. Column Order는 Logical 모델에서의 속성순서를 의미하며, Physical Order는 PK구분 없이 물리적인 Column Order를 의미한다. 따라서 Logical 모델에서의 속성순서대로 테이블을 생성하려면 Default로 되어 있는 Physical order 체크옵션을 뺀다.
<input checked="" type="checkbox"/> View	View table 생성에 관한 옵션
<input checked="" type="checkbox"/> Index	인덱스 생성 옵션으로 PK, FK, AK, IE, Physical Storage, Clustered 등을 지원한다. 여기서 한가지 유의할 점은 PK는 자동 인덱스 생성으로 옵션에서 중복하여 체크 할 필요는 없다.
<input checked="" type="checkbox"/> Referential Integrity	Primary Key와 Foreign Key를 생성하는 옵션이다. 즉 관계선에 따른 제한조건을 부여하는 것이다. DDL문장은 CREATE 문과 ALTER문 중 선택할 수 있다. Foreign Key에 대한 옵션은 ALTER 문으로 하는 것을 권장하고 있다. ERwin의 ALTER문은 테이블이 모두 생성된 후 문장이 나오기 때문에 앞서 생성되지 않은 테이블을 참조하는 오류를 피할 수 있다.
<input checked="" type="checkbox"/> Trigger	관계선에 의한 트리거와 사용자가 정의한 트리거를 생성한다. 단 RI에서 부여한 FK 제한 조건이 트리거 보다 우선한다.
<input checked="" type="checkbox"/> Other Option	Constraint Name은 ERwin Data Modeler의 이름 규칙에 따를 경우 사용하며 Quote Name은 모든 이름에 큰 따옴표가 추가된다. Owner는 Table 각각에 부여한 Owner 명이 삽입된 DDL 문장을 만든다.

3) Schema Generation 실행



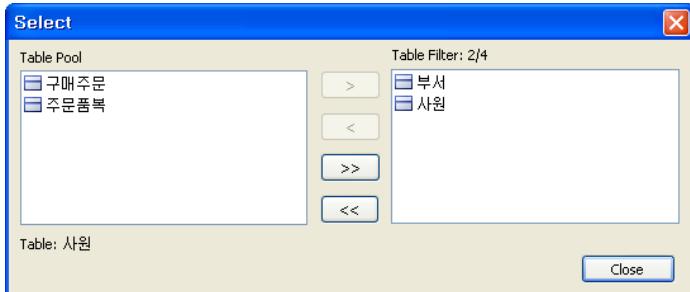
- ① Tools > Forward Engineer > Schema Generation을 선택한다.



- ② Option 탭에서 스키마 생성 옵션을 선택할 수 있다. Target DBMS별로 다양한 옵션을 제공한다.

항 목	설 명
Filter	테이블 선택 가능
Owner ...	Owner 일괄 생성
Preview	DDL문 미리 보기
Print	DDL Script 출력
Report	DDL Script 저장
Generate	스키마 생성

Summary 탭에서는 옵션 설정 내용을 확인 할 수 있다. 이렇게 설정한 옵션은 Save를 통해서 해당 모델의 Option Set으로 저장 할 수 있고, Save As 를 통해 .xml 형태로 옵션을 저장하여 다른 모델에서 Option Set을 재사용 할 수 있다. 또한 특정 테이블에 대해서 DDL 문장을 생성하고자 할 때는 Filter 기능을 사용한다.



SQL Server Schema Generation Preview

```

CREATE TABLE 사원
(
    사원코드 코드 NOT NULL ,
    사원명 char varying(20) NULL ,
    주민등록번호 char varying(20) NULL ,
    사원주소 char varying(50) NULL ,
    사원전화번호 char varying(20) NULL ,
    입사일자 datetime NULL ,
    퇴사일자 datetime NULL ,
    성별 CHAR(2) NULL ,
    휴대폰수정자 char(18) NULL ,
    부서코드 char(18) NULL
)
go

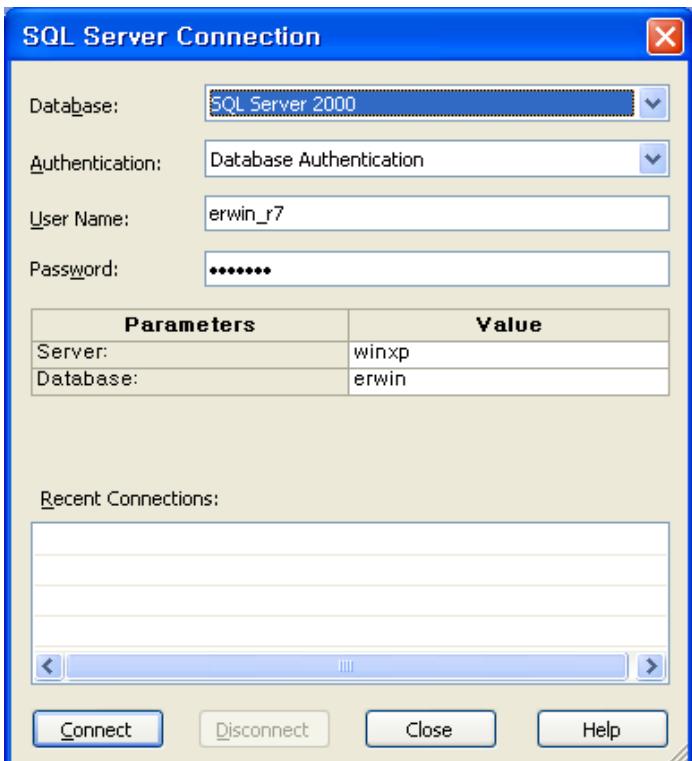
ALTER TABLE 사원
ADD CONSTRAINT XPK사원 PRIMARY KEY (사원코드 ASC)
go

```

Table Filter: 2/4 Generate... Close

③ Preview를 통해 DDL문을 확인할 수 있다. DDL 문장은 .SQL 파일로 저장 가능하며 프린트와 찾기 Option을 통해 문장을 검색 할 수 있다. 또한 특정 부분을 선택하여 복사, 붙여 넣기도 가능하다. 단 이 곳에서 수정된 사항은 모델에 반영되지 않는다.

Generate Button을 선택하면 Connect 대화 상자가 나타난다.



④ DBMS 접속 권한 및 Object 생성 권한이 있는 User의 이름과 암호를 입력하고 해당 Database Name과 Server Name을 입력한다. Database가 있는 서버가 원격에 위치하고 사용자는 Client 라면 DBMS의 Client 버전이 Client PC에 설치되어 있어야 한다. 그리고 해당 DBMS 서버는 물론 Client Network Utility(SQL 2000인 경우)에 등록이 되어야 ERwin Data Modeler를 통해 원격 DBMS에 접속할 수 있다. 이와 같은 연결 정보를 입력하고 Connect 하면 해당 DBMS에 자동으로 연결이 되고 스키마를 생성하게 되는데 Generation 중 에러가 발생하면 해당 스크립트에서 멈추게 하는 Stop If 메시지는 Erwin의 에러가 아니고 해당

Failure가 디폴트로 설정되어 있다. 이때 나타나는 에러 DBMS가 보여주는 에러이다.

5.2 Reverse Engineer

운영 시스템의 Database 정보를 분석하여 모델(ERD)을 생성한다. 이 기능은 운영 DBMS 또는 Platform에서 새로운 DBMS, Platform, Client/Server로 변환 작업을 할 때 이용될 수 있다.

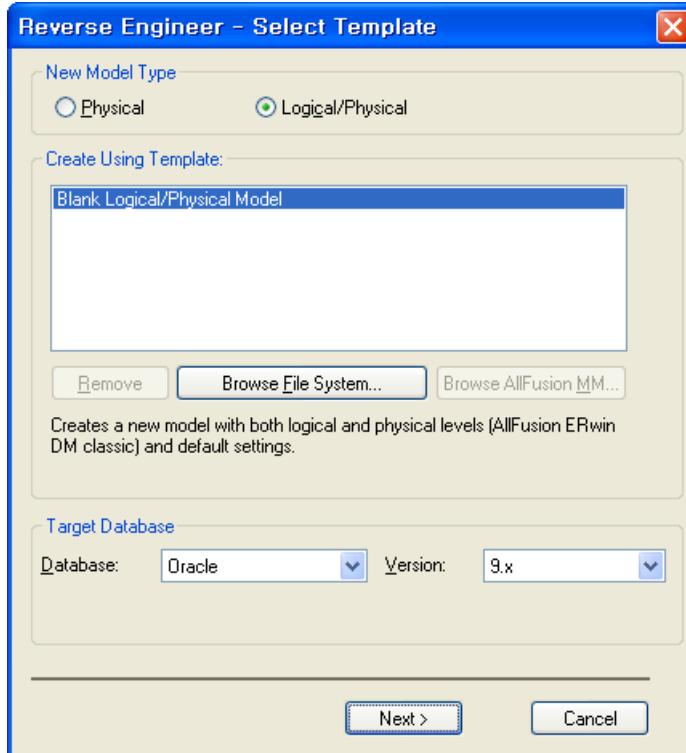
Reverse 대상은 Target DB 혹은 SQL DDL 스크립트 파일이다. 이러한 Database 정보는 Table, Column, Relationship, Triggers, Index, Stored Procedure 등을 포함한다.

1) Reverse Engineer Option 설명

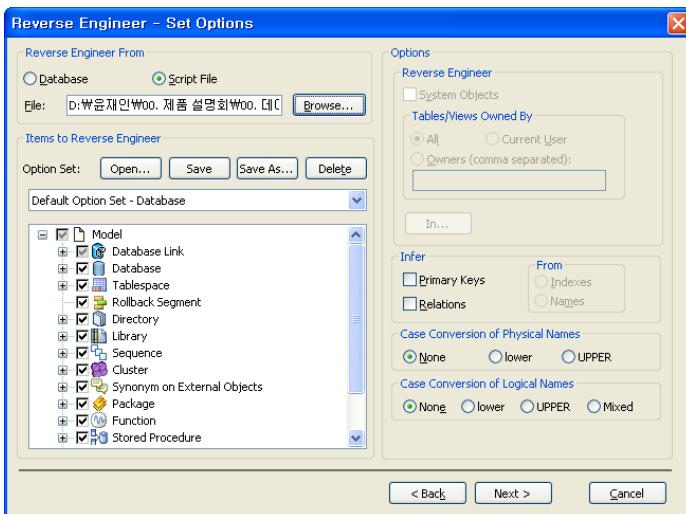
항 목	기 능 설 명
<input type="checkbox"/> Database	DBMS로부터 Reverse 한다.
<input type="checkbox"/> Script File	Script File로부터 Reverse 한다.
<input type="checkbox"/> System Object	Reverse 대상을 전체 시스템 객체로 설정한다. 이 옵션을 체크하면 사용자가 오너인 Table 뿐만 아니라 전체 시스템 Object를 가져온다.
<input type="checkbox"/> Table/Views Owned By	[All] 오너에 관계없이 Database 내의 모든 Table, View를 검색한다. [Current User] Database Connect Dialog에서 접속하는 사용자가 오너인 Table, View를 검색한다. [Owners (Comma Separated)] 정의한 사용자가 오너인 Table, View를 검색한다. 콤마로 구분하여 여러 사람을 정의할 수 있다.
<input type="checkbox"/> Infer	인덱스를 기초로 Table의 PK와 Relationship을 추론하는 옵션이다.
<input type="checkbox"/> Case Conversion of Physical name / Logical Name	Conversion 시 대 / 소문자 유형 선택에 관한 옵션이다. [lower] 소문자 [UPPER] 대문자 [Mixed] 혼합

2) Reverse Engineer 실행

DDL Script 파일로부터 리버스를 해보자.



- ① Tools > Reverse Engineer 메뉴를 선택한다.
- ② 생성할 모델의 타입을 선택한다.
- ③ Database Schema 정보를 가져올 Target Database를 선택한다.



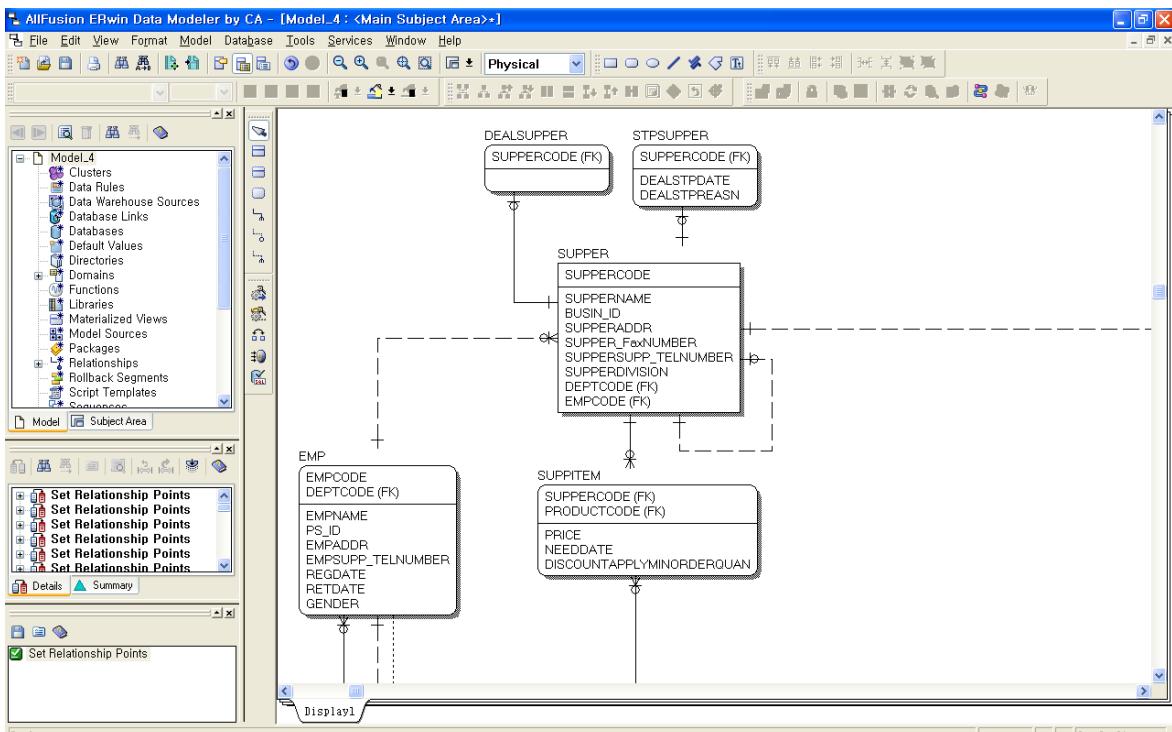
- ④ Reverse 대상을 Script File을 선택하고 Browser를 통해 File을 선택한다.

항 목	설 명
System Object	System Object까지 리버스 한다.
All	모든 User의 Object를 전부 리버스 한다.
Current User	현재 접속 User의 Object를 리버스 한다.
Owner	해당 Owner의 Object만 리버스 한다.
Infer	Index와 Name을 기준으로 기본키와 관계를 추론한다.
lower	소문자로 리버스 한다.
Upper	대문자로 리버스 한다.
Mixed	대 / 소문자 혼합

- ⑤ Option은 사용자가 정의 할 수 있으나 Default로 두고 Next를 선택한다.



⑥ Script File로부터 정보를 읽어 들인다.



⑦ Script File을 읽어 모델이 자동 생성 된 것을 확인 할 수 있다.

Tip

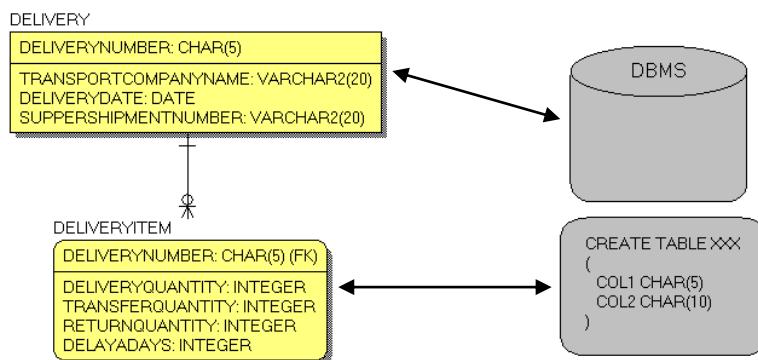
Reverse Engineering 중 다음과 같은 에러가 발생할 수 있다.

- 1) SQL Script Parse Error : SQL DDL 문장이 사용자가 선택한 DBMS와 맞지 않을 때 발생한다.
- 2) Parsing Error : DDL 문장상의 에러이다.

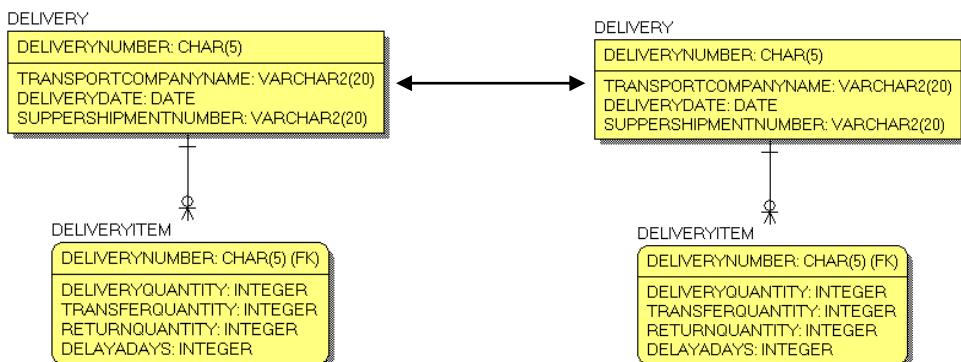
5.3 Complete Compare

Complete Compare는 모델간의 비교 및 DBMS, DDL Script와의 비교를 통해 현재의 모델과 Database를 관리하는데 매우 강력한 기능을 제공한다. R7에서는 4.1 제품의 Complete Compare 기능에서 한 단계 업그레이드 되었다. 모델의 세밀한 정보 까지도 비교가 가능하다. 예로 예전 제품에서는 인덱스의 Sort Type (Ascending, Descending)의 비교가 되지 않았다. 모델과 실제 스키마의 Sort Type이 다름에도 이러한 부분을 찾아주지 못하였다. 하지만 R7에서는 이러한 부분까지 세밀하게 비교가 가능하다. 사용자가 정의한 비교 옵션에 따라 각 항목을 비교하여 차이점을 그래픽하게 보여준다. 이러한 Object 간의 차이점은 선택적으로 Synchronization(동기화) 할 수 있다. ERwin에서는 Database Level 그리고 Model Level의 비교가 가능하다.

1) Physical Model ↔ Database / DDL Script



2) Logical / Physical Model ↔ Logical / Physical Model

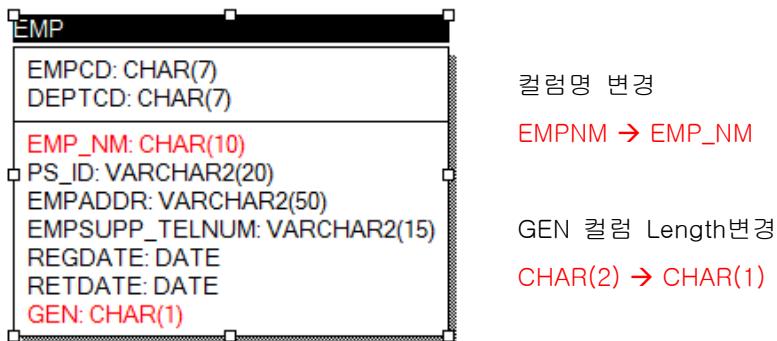


Complete Compare 기능은 Database에 이미 데이터가 있는 경우도 Table 구조 변경이 가능하다. 이 기능은 내부적으로 Table을 갱신할 때 Temp Table을 생성하고 데이터를 백업한 다음 원래의 Table 갱신한 후에 데이터를 로딩하는 과정으로 진행된다.

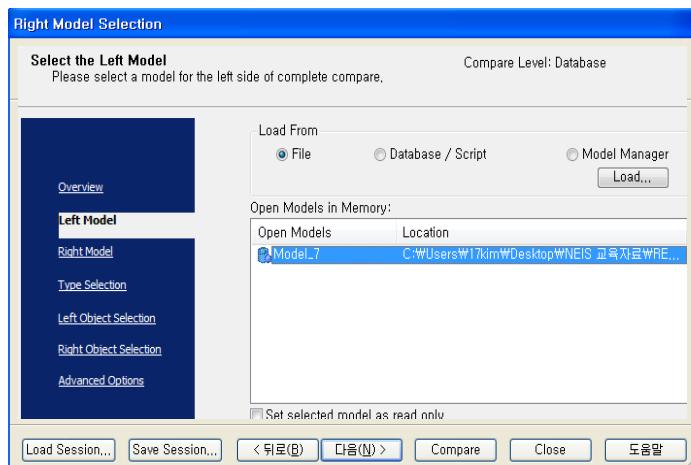
3) Complete Compare의 실행

Physical 모델과 DDL Script File 간의 비교를 해보자.

먼저 아래와 같이 ERD에 수정이 있다고 가정하도록 하겠다. (ERD에서 붉은 색으로 표시)

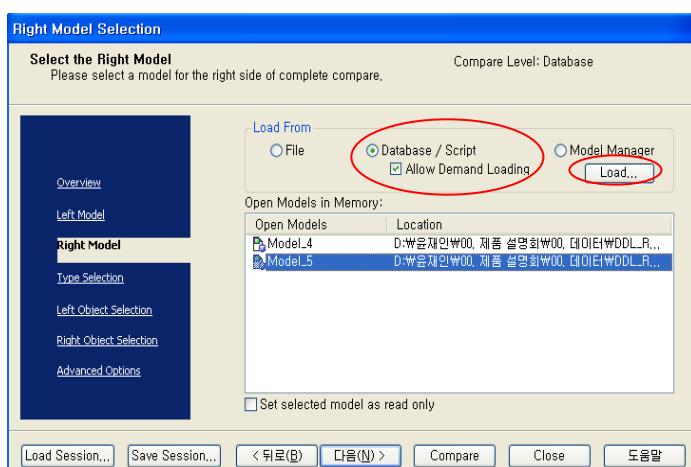


Compare화면의 각 단계별로 살펴보면,

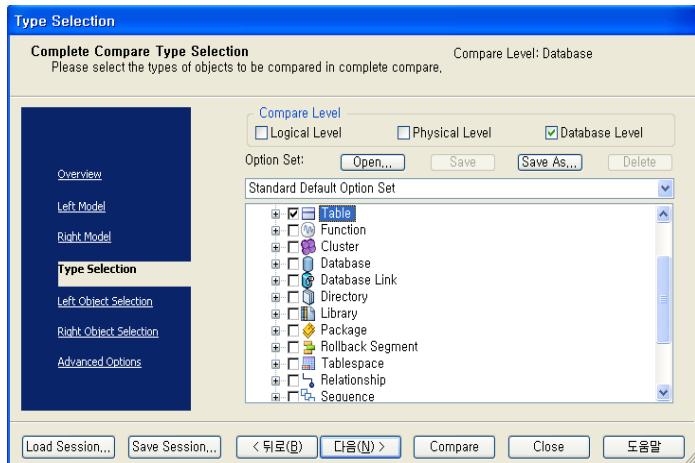


① Tools > Complete Compare > Compare 메뉴를 선택한다.

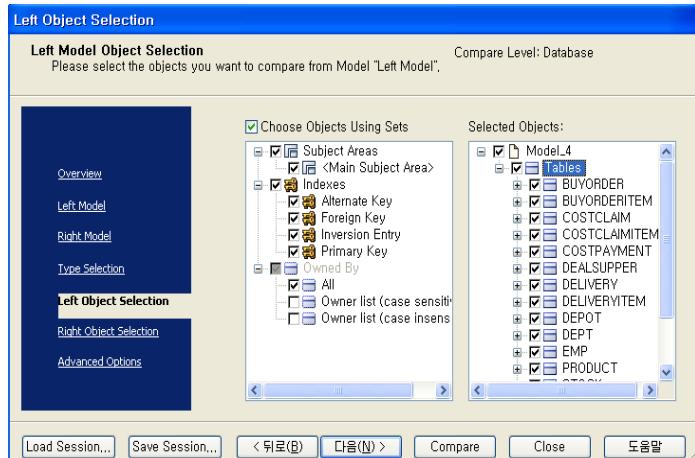
② Left Model은 Default로 현재 Open되어 있는 모델을 선택하게 된다.
(최종 Resolve Differences Dialog 화면 왼쪽에서 나타난다.)



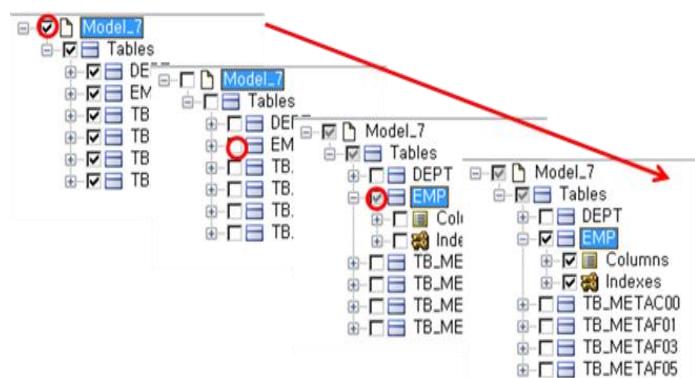
③ Right Model 선택 단계에서 비교 대상을 Database / Script로 선택한다.
Load를 통해 DDL Script File을 선택한 후 다음을 클릭한다.
(Reverse Engineer를 통하여 ERD를 생성하게 됨)



④ 비교 옵션을 Table로 선택한다.
옵션 셋을 저장할 수 있으며 재사용이 가능하다. 비교 항목에 따라 사용자가 다양한 옵션을 선택 할 수 있다.
현재 모델은 Table만 선택해도 변경 사항에 대한 비교가 가능하다.

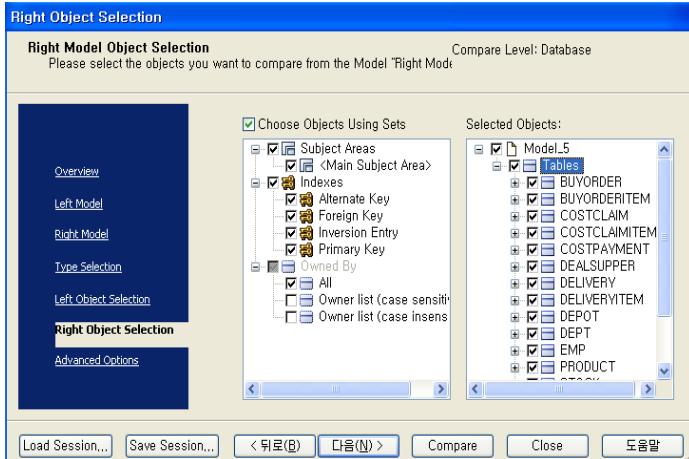


⑤ 전 단계에서 선택된 Option Set에 대하여 Left Model의 Object를 Filter할 수 있는 단계이다.
오른쪽 Tree를 확장하여 보면 Table 목록이 나오는데, 변경된 Table만 선택한다.
Table 목록에서 비교할 Table을 선택할 때에는 아래와 같은 순서로 Table을 Click해 준다.

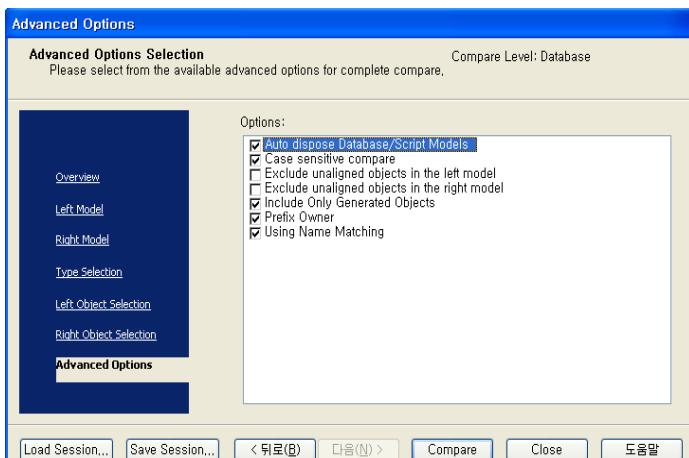


⑥ 최상위에서 모델을 Click하면 모든 Check가 사라진다

⑦ 작업 대상 Table을 Click한다



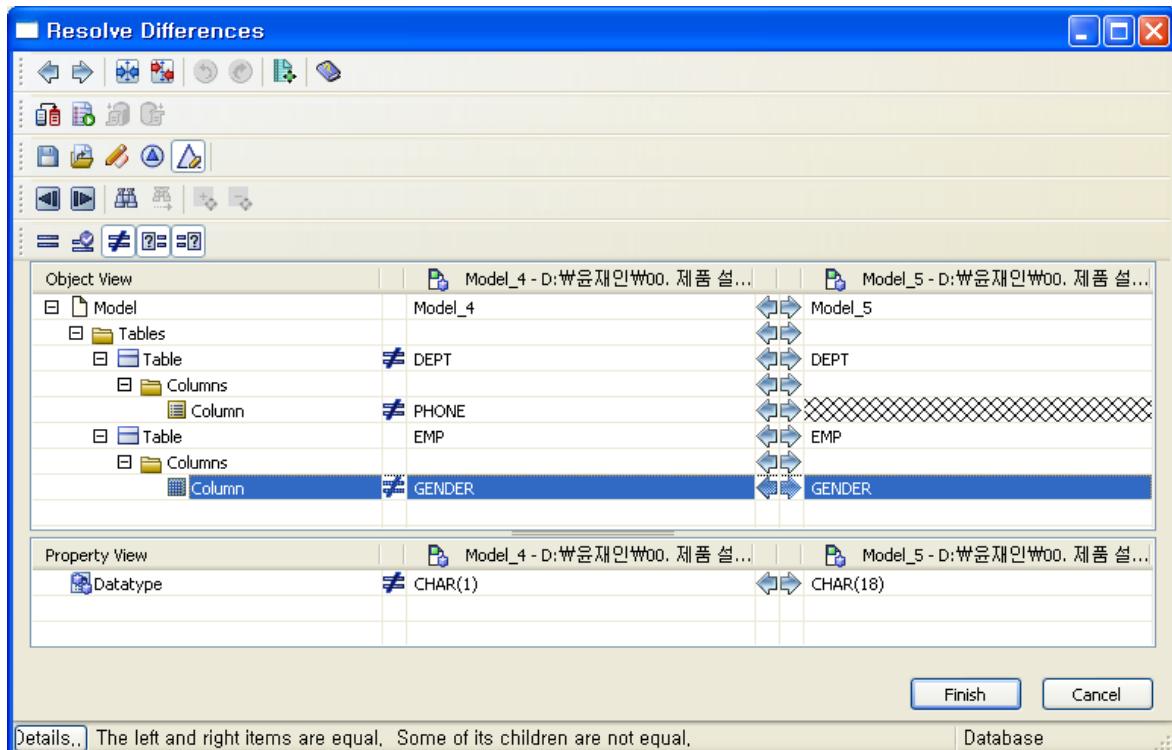
⑧ Right Model의 대상 Object를 Filter 할 수 있으며, 위에서 설명한 바와 같이 동일하게 선택한다.
선택이 종료되었으면 다음으로 이동한다.



⑨ 마지막 단계인 Advanced Option 단계이며, 비교 시 반영되는 옵션을 선택한다.
선택이 종료되었으면, Compare 버튼을 클릭한다.

* 옵션 설명

Options	옵션 설명
Auto dispose Database/Script Models	check되어 있으면 Compare 완료 후 Reverse하여 가져온 모델을 자동으로 저장하지 않고 모델을 Closed
Case-sensitive compare	check되어 있으면 대/소문자를 구분하여 Compare를 수행
Exclude unaligned objects in the left model	왼쪽 모델을 기준으로 선택한 Object와 비교 대상이 없으면 왼쪽 모델의 object는 보여주지 않음
Exclude unaligned objects in the right model	오른쪽 모델을 기준으로 선택한 Object와 비교 대상이 없으면 오른쪽 모델의 object는 보여주지 않음
Include Only Generated Objects	Check되어 있으면 Table Editor에서 Generate에 Check가 되어 있지 않은 항목은 비교 대상에서 제외하는 옵션
Prefix Owner	check되어 있으면 모델의 Owner 정보를 테이블명 앞에 붙여서 비교
Using Name Matching	check되어 있으면 명칭을 기준으로 비교



그림과 같이 모델과 DBMS간 차이점을 분석하여 찾아준다.

화면은 크게 3가지로 구분할 수 있다.

첫 번째, Complete Compare를 수행할 때 사용되는 Toolbar Icon으로 구성된다.

두 번째, 상위 그리드에서 상이한 Object List를 보여준다.

가운데 부분은 다시 3부분으로 나뉘는데, Object view부분은 Type Selection단계에서 선택한 Object를 구분하고, Object View옆으로 Left / Right Model의 실제 object를 비교하여 정렬한 화면이다.

세 번째, 상위 그리드에서 선택한 Object List의 자세한 차이점을 하위 그리드에서 확인 할 수 있다.

Tip

Property View에는 다음과 같은 정보를 포함하고 있다.

Calculate : Calculate는 그 값이 model의 다른 값으로부터 계산되어 나왔음을 의미한다. 예를 들자면, object view에서 Child Entity를 선택하셨다면, Foreign Key 같은 property는 Parent entity에서 상속받은 대로 계산된다. 다른 예는 domain에서 상속받은 property들인데, 이런 것들은 calculated로 보여진다.

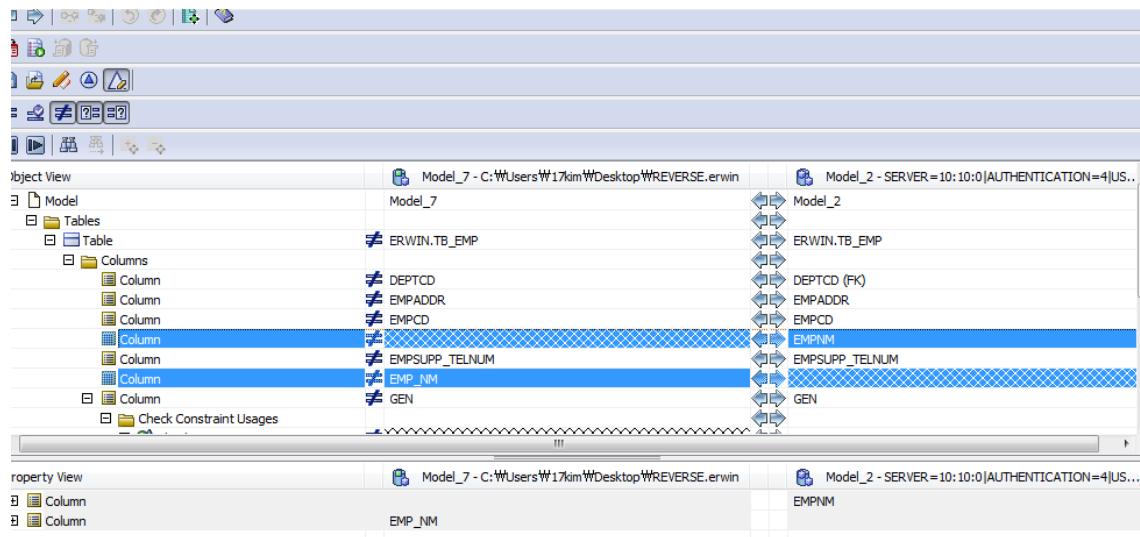
Filtered : 이것은 달라진 점이 존재함을 보여주며, 그 차이는 Complete Compare로부터 필터 되어 나온 특정한 object이기 때문에 보여지는 것이다. 바꿔 말하자면, Complete Compare에서 filtering options을 선택하였기 때문에 차이가 생긴 것이다.

또한 사용자는 다양한 비교 옵션을 통해 차이점을 분석하고, 차이점을 모델과 DBMS에 반영할 수 있다.

* 아이콘 설명

항 목	기 능 설 명
	비교 모델간 같은 항목을 보여준다.
	비교 모델간 서로 같지 않은 항목을 보여준다.
	Left 모델에 없는 항목을 보여준다.
	Right 모델에 없는 항목을 보여준다.
	Right 모델의 내용을 Left 모델로 반영한다.
	Left 모델의 내용을 Right 모델로 반영한다.
	모델간의 Object가 같으나 간혹 매치가 되지 않은 경우가 발생한다. 이런 경우 Match를 이용해 일치 시켜준다.
	사용자가 반영한 Action Log / Message Log를 확인 할 수 있다.
	Known Differences에 대한 비교 항목을 snapshot 으로 저장한다.
	저장한 Snapshot 을 불러온다.
	Resolve Difference 창에서 특정 row(속성항목)를 비교하지 않을 수 있다.
	아이콘을 클릭함으로써 Resolve Difference 창에서 특정 Row를 비교 대상에서 제외했을 때, 해당 아이콘을 클릭하여 비교 제외된 속성을 확인할 수 있다.
	비교 내용(Equal, Not Equal)을 Reporting 한다. (Internet Browser, *.doc, *.xls)
	Left(Right) Alter Script / Schema Generation – DBMS에 반영할 DDL문을 생성 / 실행한다.

DB의 정보를 변경하고자 하는 작업이기 때문에 이 경우 아래와 같이 작업한다.



우선 변경된 정보를 찾는다. (변경 예제는 page. 118 참고) 첫 번째 EMPNM과 EMP_NM은 명칭이 다르기 때문에 두 대상이 각기 다른 Object로 인식하여 결과를 나타낸다. 상단에 버튼을 이용하여 두 Object가 같은 항목임을 인식하게 한다. 순서에 상관없이 EMPNM을 선택하고, 를 누르고 나머지 EMP_NM을 선택한다.

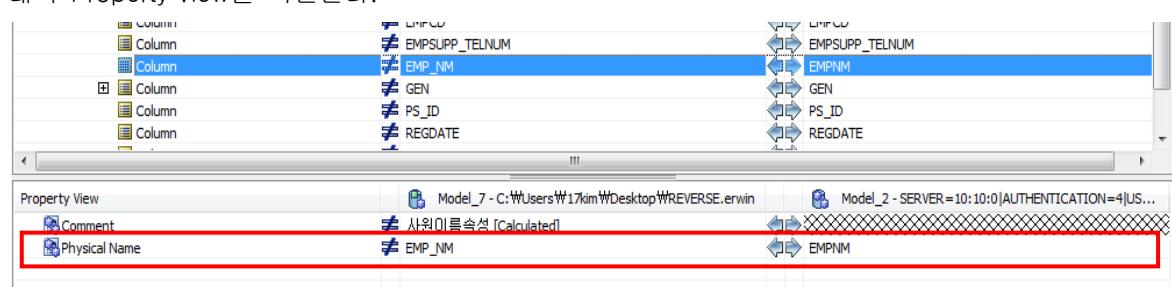


Do you want to match "EMP_NM" with "EMPNM"?

예(Y)

아니요(N)

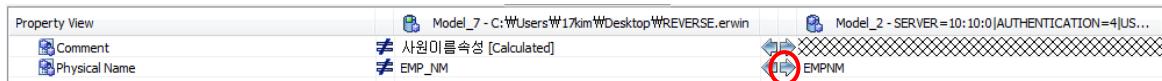
EMPNM은 모델에서 EMP_NM으로 변경된 상태로, 모델의 변경 사항을 DB에 반영하시기 위해서 아래쪽 Property view를 확인한다.



Property view에 나타난 정보는 두 건이다.

Comment정보는 모델에만 있는 정보로, DB에 반영할 사항이 아니므로, skip한다.

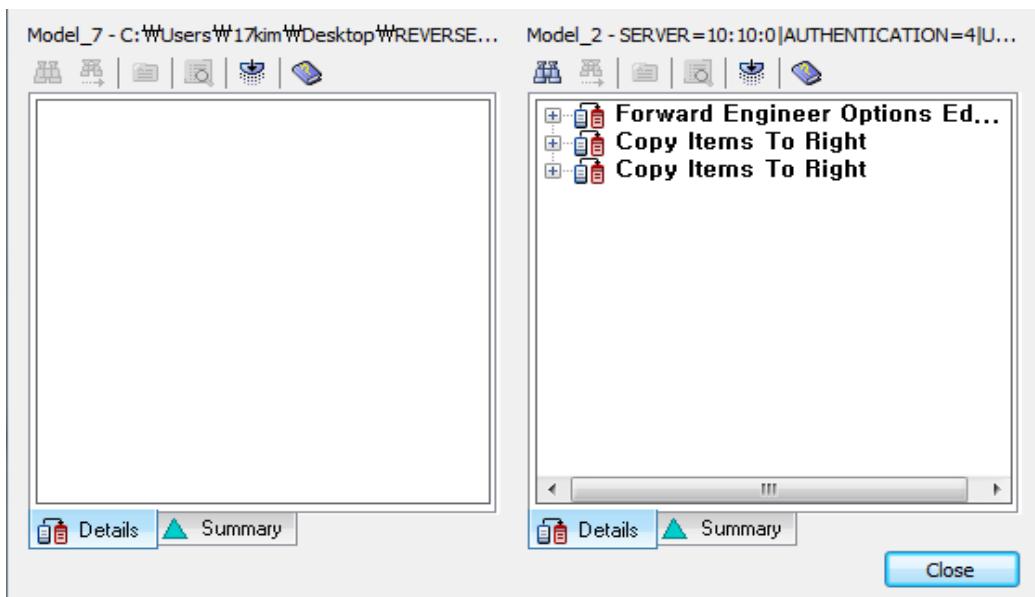
아래의 Physical Name부분은 모델의 내용을 DB에 반영하므로 화살표를 오른쪽으로 선택한다.



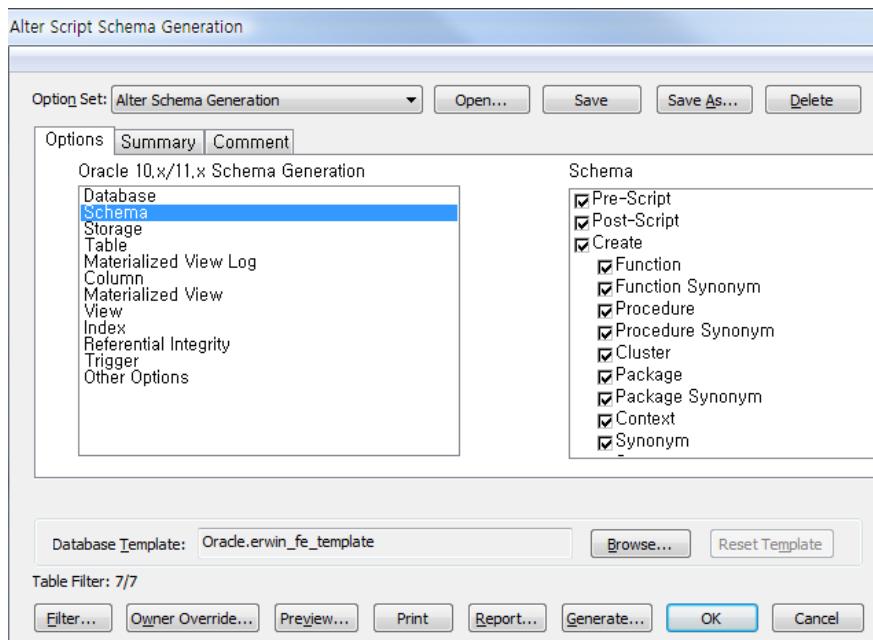
화살표를 선택하는 Physical Name은 DB에 반영될 항목으로 포함되었고, Resolve Differences에서 사라지게 된다.

DB에 변경된 내용을 반영하기 위해서는 Alter Script를 생성하여 작업해야 하기 때문에 현재 상태에서 Finish를 하시게 되면 DB에 변경은 수행되지 않는 것을 유의한다.

나머지 변경 작업까지 모두 끝났으면 상단의 를 이용하여 Complete Compare상에서 반영한 작업 내용을 확인할 수 있다.



다음으로  Right Alter Script/Schema Generation icon을 이용하여 Alter문장을 생성하여 DB변경 작업을 수행한다. 버튼을 클릭하면 Forward Engineer와 동일한 창이 나타난다.



해당 Dialog에서는 별도의 Option 선택 없이 바로 Preview를 통하여 결과를 확인할 수 있다. 그 이유는 Complete Compare를 통하여 DB방향으로 반영한 Object를 ERwin에서 기억하고 있기 때문에 별도의 Action없이 Alter문장을 생성해 주게 된다.

```

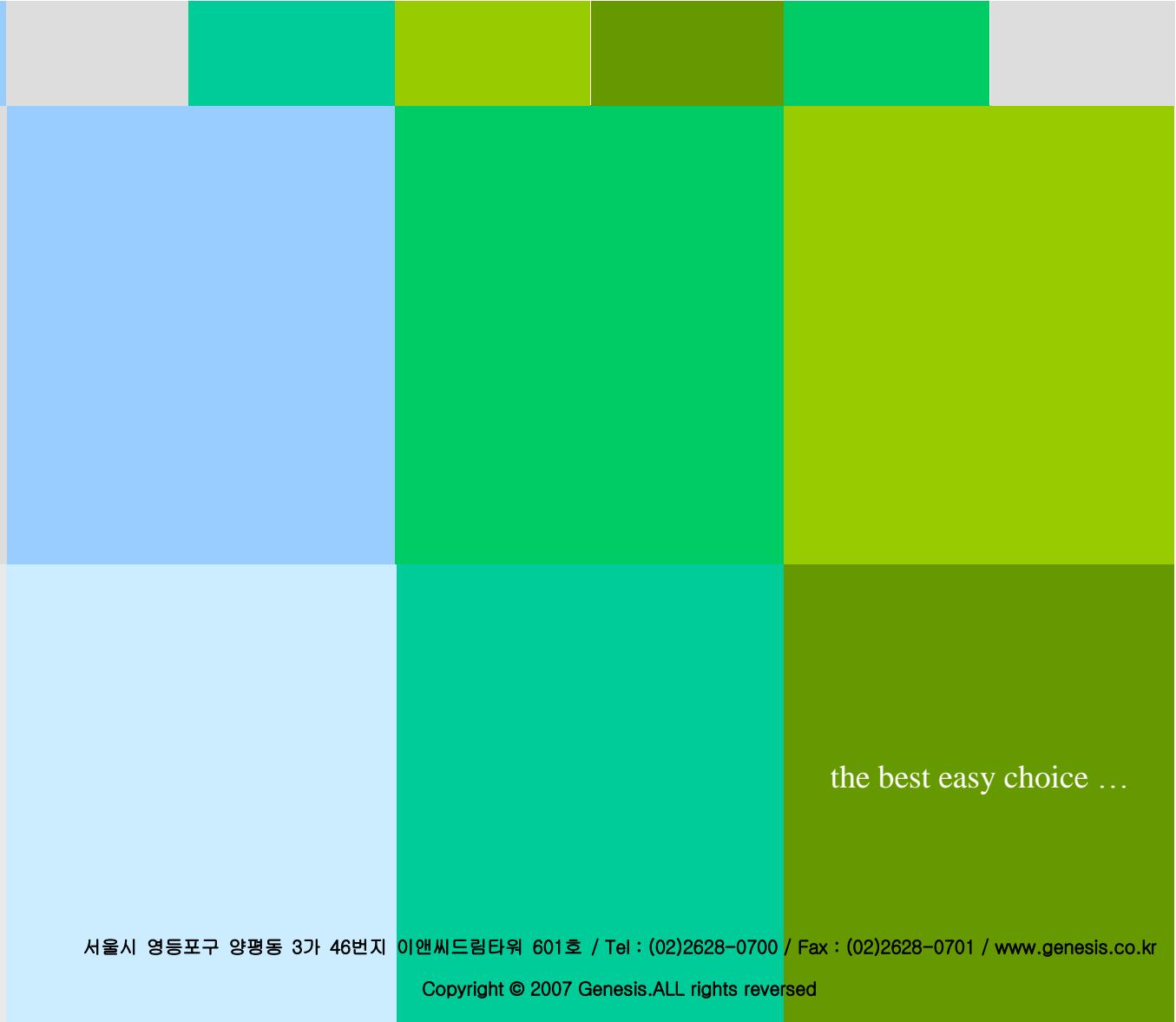
ALTER TABLE ERWIN.TB_EMP RENAME COLUMN EMPNM TO EMP_NM;
ALTER TABLE ERWIN.TB_EMP MODIFY(GEN CHAR(1 BYTE));

```

아래 Generate를 이용하여 바로 DB에 변경을 적용할 수 있고, 이를 복사하여 Script를 이용할 수 있다.

작업 결과는 Schema가 정상 변경 되었는지 확인하면 쉽게 알 수 있다.

Genesis Technologies Corp.



the best easy choice ...