

Allfusion ERwin Data Modeler User Manual

the best easy choice ...

Genesis Technologies Corp.

서울특별시 영등포구 양평동3가 46번지 E&C 드림타워 6F

Tel: 02)2628-0700

Fax: 02)2628-0701

이 책에 포함된 정보는 사전 통보 없이 변경될 수 있으며 제니스스기술(주)는 이에 대해 책임을 지지 않습니다. 이 책에 설명된 소프트웨어는 인가 협정하에서 제공되는 것으로서, 협정 내용에 의거할 경우에만 사용할 수 있습니다. 인가 협정에서 특별히 허가된 것을 제외한 매체에 복사하는 행위는 법에 저촉됩니다. 제니스스기술(주)의 서면 허가 없이는 복사를 비롯하여 어떠한 전자적, 기계적 형태나 방법으로도 이 매뉴얼을 복제 또는 전송할 수 없습니다.

판권소유) 제니스스기술(주)

CA는 Computer Associate International Inc. 등록상표이며, ERwin Data Modeler, Process Modeler, Model Manager는 Computer Associate International Inc.의 상표입니다.

1. Data Modeling 용어 정의	6
1.1 Model.....	6
1.2 Data Model	6
1.3 IDEF1X Model.....	6
1.4 IE Method	6
1.5 Logical Model.....	7
1.6 Physical Model	7
1.7 Dimensional Model.....	7
2. AllFusion ERwin Data Modeler r7 작업 환경	8
2.1 작업 창 설명	8
2.2 Menu 설명	9
2.3 Tool Box 설명	12
2.4 새로운 모델 생성	15
2.5 ERwin Diagram 저장.....	15
2.6 Diagram Print	16
3. Layer의 이해.....	18
3.1 Design Layer Architecture	18
3.3 Design Layer 구조의 예	19
3.4 Design Layer 관리.....	19
3.5 Design Layer의 추출 기준	19
3.6 Logical / Physical 모델(Classical Data Model Type).....	20
3.7 Logical / Physical모델의 분리작업	20
3.8 새로운 모델을 추출하기(Derive)	21
4. History 정보 관리.....	24
5. Logical Modeling	28
5.1 Entity, Attribute, Key 개념	28
5.2 Entity 생성	30
5.3 Entity의 이동	32
5.4 Attribute 생성	32
5.5 Relationship 개념.....	36
5.6 Referential Integrity(RI, 참조무결성).....	52
5.7 Subject Area (주제영역)	56
5.8 Stored Displays	58
5.9 Display Options	59

6. Physical Modeling	66
6.1 Physical / Logical Model의 표현	66
6.2 Physical Model 변환	66
6.3 Logical / Physical Object의 차별화	67
6.4 Target Server 지정	69
6.5 Table Property	70
6.6 Column Property	73
6.7 View Table	82
7. Transform	83
7.2 Transform 기본 원리	87
8. Forward / Reverse	110
8.1 Forward Engineer	110
8.2 Reverse Engineer	113
9. Complete Compare	116
9.1 Complete Compare	116
10. Add Model Source / Sync With Model Source	121
10.1 Add Model Source	121
10.2 Sync With Model Source	125
11. Reporting	129
11.1 Data Browser	129
11.2 Report Template Builder (RTB)	136
12. 모델의 표준화	145
12.1 Names	145
12.2 Datatypes	150
13. Dimensional Modeling	158
13.1 Data Warehouse	158
13.2 Dimensional Model 생성	162
13.3 Dimension Table 생성	164
13.4 Data Warehouse Rules 설정	167
13.5 Data Source 관리	169
14. SPY Utility	173
15. API를 이용한 산출물 가이드 (Add ins.)	174
15.1 VB를 활용한 산출물 만들기	174
15.2 Add Ins.. 등록	175

16. 변경된 내용에 대한 가이드 모음	179
16.1 Know Issues	179
16.2 Action Log	179
16.3 함수기반 인덱스(Function Based Index).....	180
16.4 Metadata Integration Wizard	182

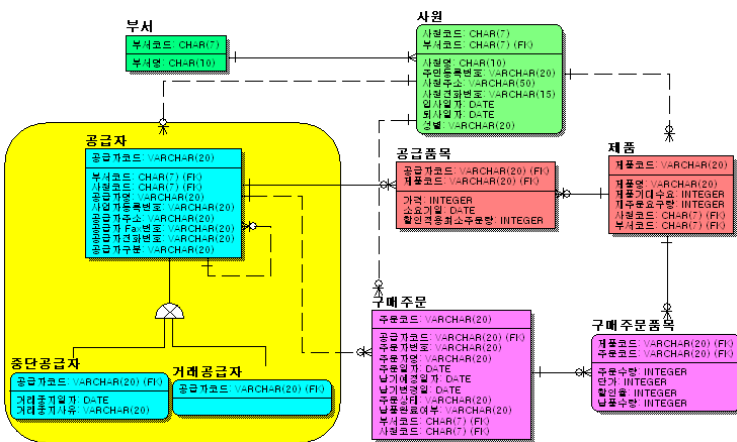
1. Data Modeling 용어 정의

1.1 Model

Model이란 문서화된 설명을 가지고 명확하게 정렬된(정의된) 그림으로 표현한 것을 말한다. 서로 다른 모델은 서로 다른 관점을 반영한다. (ex : 경영자의 관점, 설계자의 관점, 계약자의 관점)

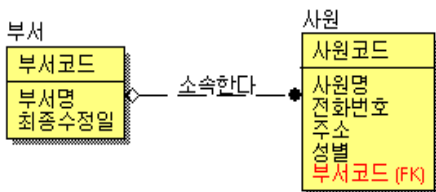
1.2 Data Model

Data model 이란 업무 요구사항들을 나타내는 데이터 구조와 업무규칙에 관한 명세서이다. 시각적으로 Data Model은 Entity-Relationship(ER) Diagram으로 표현된다.



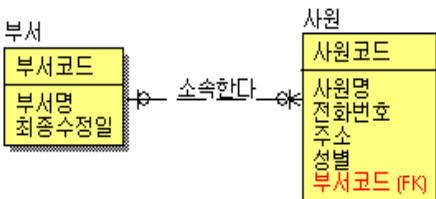
1.3 IDEF1X Model

1981년 미 공군에서 ICAM(Integrated Computer Aided Manufacturing)프로젝트의 생산 시스템 분석 및 설계 목적으로 개발된 IDEF 방법론 중에 시스템 내부의 데이터를 모델링 하는 기법이 IDEF1X 이다.



1.4 IE Method

J.Martin이 제창한 Information Engineering(IE)에서 사용된 기법으로 국내에서 가장 많이 사용하며 정보공학 방법론이라 부른다.



1.5 Logical Model

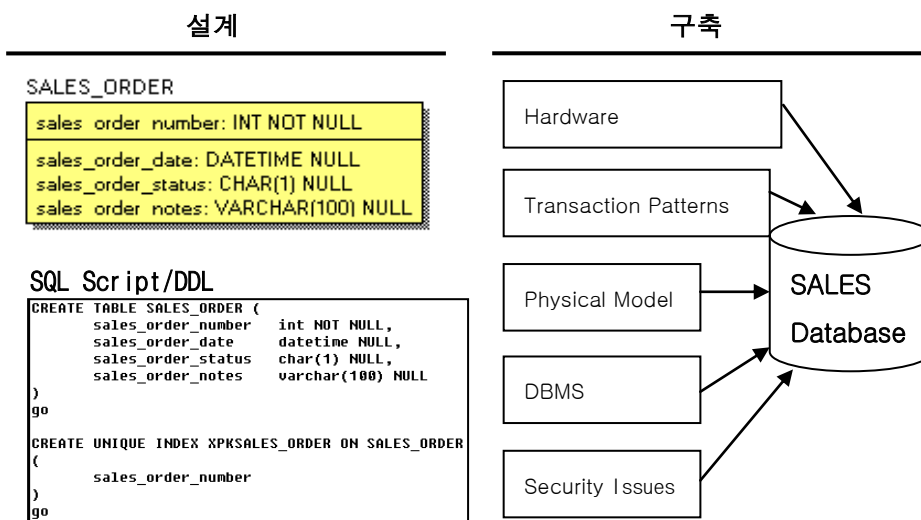
Logical Model은 업무를 지원하는데 필요한 정보와 업무규칙을 표현하는 것을 말하며 DBMS와는 독립적으로 표현할 수 있다.

1.6 Physical Model

Physical Model은 설계와 구축 사이에서 생성 할 수 있는 데이터 구조이다.

[특징] ■ DBMS에 의존적이다.

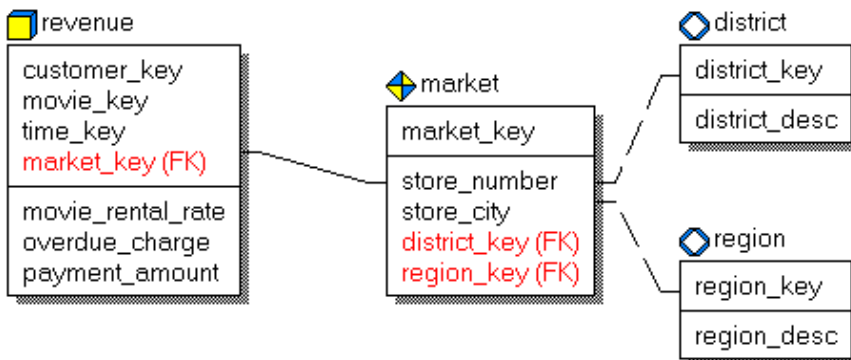
- SQL의 구성 요소인 DDL로 구현된다.
- DBMS 성능(Performance) 효과에 역점을 두어 모델링 해야 한다.



1.7 Dimensional Model

Dimensional Model은 Data Warehouse를 설계 할 때 사용되는 특수한 형태의 Data Model이다. 이 Dimensional Model의 중심에는 Fact Table이 존재하는데, 이것은 Data Warehouse의 분석 주제가 된다. Fact Table 주위에는 여러 개의 Dimension Table이 존재하며 이것은 Fact Table이 분석되는 카테고리를 가리킨다.

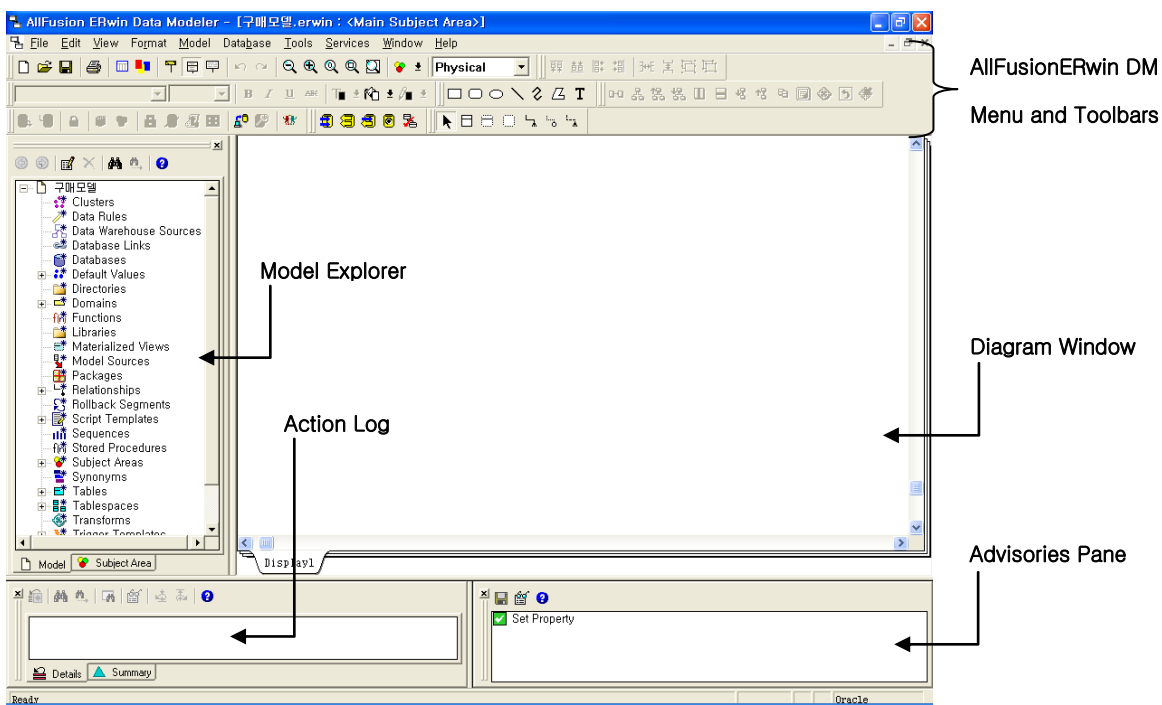
Fact Table의 Primary Key는 Dimension Table에서 이주된 Foreign Key로 구성되어 있다.



2. AllFusion ERwin Data Modeler r7 작업 환경

4.1 버전에서 R7으로 업그레이드 되면서 작업환경이 가장 두드러지게 변화된 것은 Undo와 Redo 기능이 추가된 것이다. 이 기능의 추가로 사용자는 자신이 원했던 작업시간으로 되돌아 갈 수 있게 되었으며, 추가된 Action Log / Advisories Pane 을 통해 과거 기록이나 현재 사용자가 어떠한 작업을 진행 하였는지를 알 수 가 있게 되었다.

2.1 작업 창 설명



항 목	기 능 설 명
■ Toolbars	모델링 작업을 좀더 빠르게 진행 할 수 있도록 자주사용 되는 기능이 포함 되어 있으며 사용자가 원하는 장소로 자유롭게 이동할 수 있다.
■ Advisories Pane	현재 사용자가 어떠한 작업을 진행 하였는가에 대한 정보를 보여준다.
■ Model Explorer	모델내의 탐색기 역할을 하며, 각 Object는 Hierarchy구조로 Drag and Drop 방식으로 추가할 수 있다. 4.1버전과 달라진 점은 Domain 탭이 사라지고 Model, Subject Area의 탭으로 구성 된다.
■ Action Log	실시간으로 사용자가 작업한 내용을 기록된다.
■ Diagram Windows	모델을 생성하면 기본적으로 창 이름이 Display1으로 설정되며 이곳에 사용자는 Graphical 하게 모델링 작업을 진행 할 수 있다.

2.2 Menu 설명

1) File

- New 신규 파일을 생성
- Open 저장되어 있는 ERwin Format 파일 열기
- Close ERwin 파일 닫기
- Save 작업(편집) 내용을 “.erwin” 파일로 저장
- Save as 다른 파일 Format으로 저장 (7.0.1, 4.1.X, Template, XML)
- Import PM, 다른 Application 파일을 Open 하여 “.erwin” 파일생성
- Export “.erwin” 파일을 PM, 다른 Application 파일로 변환
- Print 편집창의 내용(모델)을 인쇄
- Print Setup 용지 크기와 인쇄 옵션 변경

2) Edit

- Undo 작업 내역 되돌리기
- Redo Undo 작업 내역 되돌리기
- Cut 선택된 Object 잘라내기
- Copy 선택된 Object 복사하기
- Paste 복사한 Object 붙이기
- Select All 전체 Object 선택
- Goto 특정 Entity & Table로 이동
- Show Transform Source Object 변환 전 Source Object 보기
- Show Transform Target Object 변환 후 Target Object 보기
- Reverse All Transform 변환 작업 되돌리기
- Resolve All Transform 변환 작업 고정하기 (Target Object 고정)

3) View

- Redraw Diagram Diagram 다시 그리기 (Default)
- Toolbars Toolbars 보기 / 감추기
- Model Explorer Model Explorer 보기 / 감추기
- Action Log Pane Action Log Pane 보기 / 감추기
- Advisories Pane Advisories Pane 보기 / 감추기
- Stored Display Tabs Stored Display Tabs 보기 / 감추기
- Status Bar 상태 표시줄 보기 / 감추기
- Zoom 화면 확대, 축소, 전체화면 보기, 특정 영역 보기

7) Tools

■ Reverse Engineer	DB 스키마 / DDL Script의 정보를 읽어 ERD 자동 생성
■ Complete Compare	현재 모델과 다른 모델, DB Schema, DDL등과 비교하기
■ Add Model Source	모델 소스 추가하기
■ Link Model Source	모델 소스 연결하기
■ Sync Model Source	LM, PM으로 분리된 모델 동기화
■ Drive New Model	현재 모델로부터 새로운 모델 생성
■ Split L/P Model	복합모델을 LM과 PM으로 분리
■ Report Template Builder	모델 정보를 Text 와 ERD(그림)으로 출력
■ Data Browser	모델 정보를 Text으로 출력
■ Names	모델에 사용되는 용어 설정
■ Datatypes	모델에 사용되는 Datatype 설정
■ Add ins	사용자가 만든 프로그램 메뉴에 추가
■ Option	모델 정보의 log파일, XML, 진단 옵션 설정
■ Forward Engineer (Physical)	현재 모델을 이용 DDL 문장 생성 및 DB Schema 생성
■ Volumetrics (Physical)	Table Size를 설정

8) Window

■ Cascade	폭포수형 형태로 Diagram 창 보기
■ Tile Horizontal	수평분할 하여 Diagram 창 보기
■ Tile Vertical	수직분할 하여 Diagram 창 보기

9) Help

■ Help Topics	도움말 항목
■ What's New	추가된 내용
■ How to Use Help	도움말 메뉴얼 사용법
■ Update the ERwin License	라이선스 갱신하기
■ Generate Metamodel Document	Metamodel 정보를 HTML형식으로 생성
■ Generate Diagnostic Export	모델 진단 정보 내보내기
■ About AllFusion ERwin Data Modeler	프로그램의 버전정보 확인

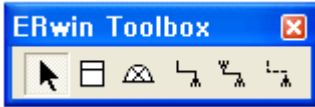
2.3 Tool Box 설명

1) ERwin Standard Toolbar : 모델링 작업에 필요한 공통적인 사항이 포함되어 있다.



아이콘	Toolbar Name	설명
	Create Model	새로운 Diagram 창을 생성
	Open Model	기존의 Diagram 창을 불러오기
	Save Model	모델을 저장
	Print	모델 일부 혹은 전부를 프린트
	Data Browser	Data Browser 수행
	Report Template Builder	Report Template Builder 수행
	Entity Level	엔티티 이름만 보기
	Attribute Level	엔티티 이름과 속성을 보기
	Definition Level	엔티티 이름과 정의 정의 보기
	Undo	작업 내용 되돌리기
	Redo	Undo 작업 되돌리기
	Zoom In	Diagram 축소하기
	Zoom Out	Diagram 확대하기
	No Magnification	확대 비율 0으로 보기 (기본 Size 보기)
	Fit Model	일정한 비율로 확대해 보기 (전체 모델 보기)
	Select Rectangle to fit	선택 영역만 확대해 보기
	Create Subject Area	주제영역 생성 / 편집
	Select Subject Area	주제영역 이동하기
	Logical	LDM / PDM으로 이동

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



아 이 콘	Toolbar Name	설 명
	Select	모델의 오브젝트 선택
	Entity / Table	엔티티 / 테이블 생성
	Exclusive sub-category	일반화(완전 배타) 관계를 표현
	View Table	뷰 테이블 생성
	Relationship	관계선을 그릴 때 사용

<Logical DM Toolbo



<Physical DM Toolbo

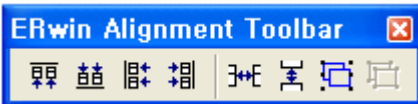
3) Font & Color Toolbar : 배경
 티터 등의 색이나 글자의 폰트
 한다.



화면, 엔
 를 설정

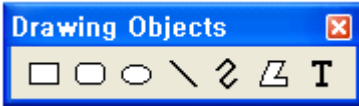
아 이 콘	Toolbar Name	설 명
	폰트 박스	문자의 폰트 설정
	크기 박스	문자의 크기 설정
	볼드 체	문자 굵게
	이탤릭 체	문자 기울이기
	언더라인	문자에 밑줄 긋기
	줄 긋기	문자에 줄 긋기
	텍스트 컬러	선택된 문자의 색 설정
	필(Fill)	엔티티 내부의 색 설정
	아웃라인(Outline)	엔티티 바깥 선의 색 설정

4) ERwin Alignment Toolbar : 엔티티나 객체를 다중 선택하여 정렬, 간격조절, 그룹을 설정한다.



아 이 콘	Toolbar Name	설 명
	Align Top / Bottom	상 · 하 정렬
	Align Left / Right	좌 · 우 정렬
	Space Horizontally / Vertically	수직 · 수평 간격 조정
	Group / Ungroup	그룹 설정 / 해제

5) Drawing Objects Toolbar : 직사각형, 선, 다각형, 타원과 Text를 입력 한다.



아이콘	Toolbar Name	설명
	Rectangle / Rounded Rectangle / Ellipse	직사각형, 타원, 원을 그린다.
	Line / Polyline	직선과 곡선을 그린다.
	Polygon	다각형을 그린다.
	Text	텍스트 상자를 그린다.

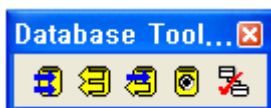
6) Services Toolbar : Model Manager사용시 사용하는 툴바이다.

7) ERwin Transform Toolbar : 모델내의 Object를 또 다른 구조 Object로 자동변환 시 사용



아이콘	Toolbar Name	설명
	Many-to-Many Transform	다대다 관계를 일대다 관계로
	Supertype-Subtype	속성을 합치거나 풀거나 일대다 관계로
	Vertical / Horizontal Partition	수직 / 수평 파티셔닝
	Roll-Up / Roll-Down Denormalization	부모나 자식으로 병합하기
	Column Denormalization	테이블의 컬럼을 복사 (컬럼 역정규화)
	Show Transform Source Objects	Transform 하기 전에 상태 보기
	Show Transform Target Objects	Transform 후에 상태 보기
	Reverse All Transforms	Transform 모두 되돌리기
	Resolve All Transforms	Transform 상태 고정하기

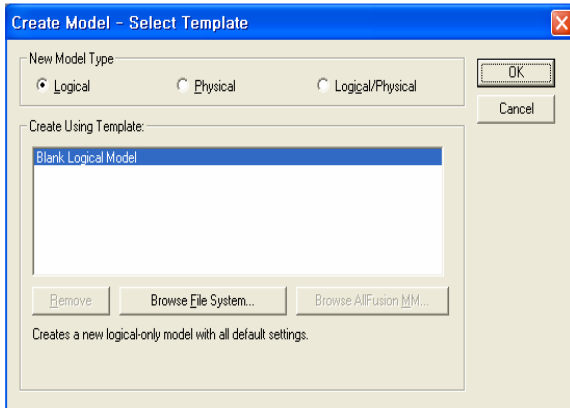
8) Database Toolbar : 물리모델에서 활성화되며 Forward와 Reverse등의 버튼이 있다.



아이콘	Toolbar Name	설명
	Forward Engineer	Forward Engineer 실행
	Reverse Engineer	Reverse Engineer
	Complete Compare	Complete Compare
	Select Target Server	Target Server 선택
	Validate SQL of current model	모델을 검증

2.4 새로운 모델 생성

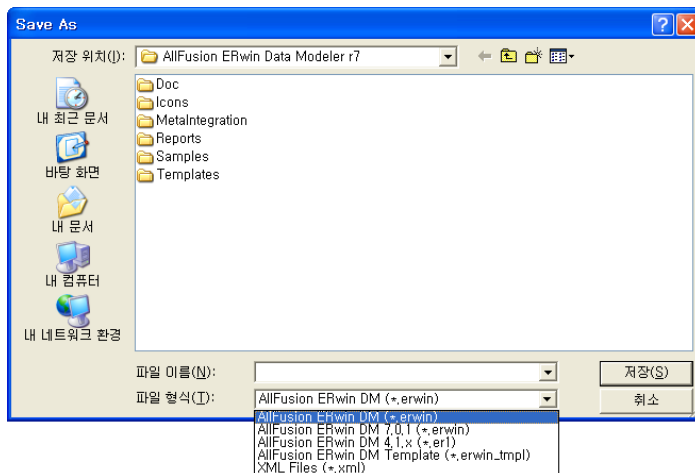
ERwin을 실행, File메뉴에서 New버튼, Toolbar에 이 버튼을 클릭하면 새로운 모델을 생성할 수 있으며 나타나는 화면은 다음과 같다.



항 목	설 명
Logical	논리 정보만으로 구성되는 모델 타입
Physical	물리 정보만으로 구성되는 모델 타입
Logical / Physical	ERwin 모델의 전형적인(Classical) 타입으로, 하나의 모델에 하나의 Logical side와 하나의 Physical side로 구성된다.
Browse File System	각 모델 타입 별로 사용자는 템플릿을 만들어 사용할 수 있으며

2.5 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.



1) ERwin 파일 형식

.erwin은 기본적인 파일 형식으로 정보를 그래픽(Graphical)하게 저장할 수 있도록 지원하며 ERwin 4. 버전의 .er1의 파일 형식이 r7으로 오면서 erwin으로 변경되었다.

erwin_tmpl(Template)는 새 모델 작성시 Template에 있는 ERwin Objects와 모든 Setting 정보를 모두 상속 받아서 저장하는 기능으로 ERwin 4.1 이하 버전의 ERT파일 R7에서 erwin_tmpl로 변경되었다. XML 형식도 지원한다.


2) ERwin 4.1버전의 파일 형식 지원하기

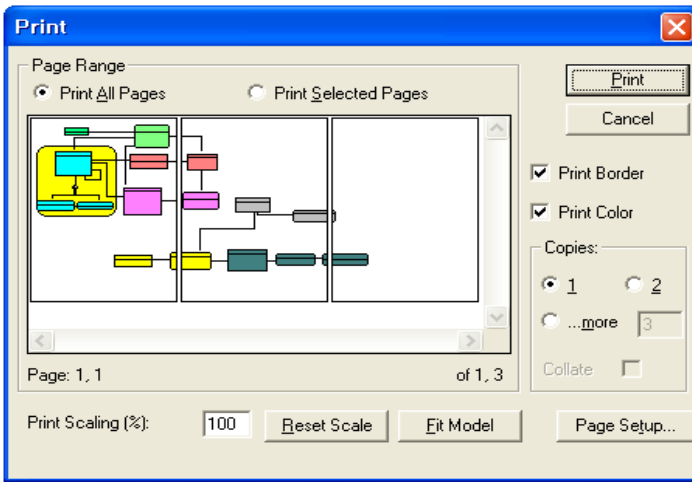
ERwin Data Modeler r7에서는 하위버전 ERwin 4.1버전의 파일로 저장과 오픈이 가능하다.

하지만 ERwin 3.5.2버전의 .er1파일을 직접 오픈 할 수 없으며 R7에서 오픈 하려면 ERwin 3.5.2에서 .ERX파일로 저장하거나 ERwin 4.1버전에서 저장 후 오픈 할 수 있다.

2.6 Diagram Print

2.6.1 인쇄 대화상자

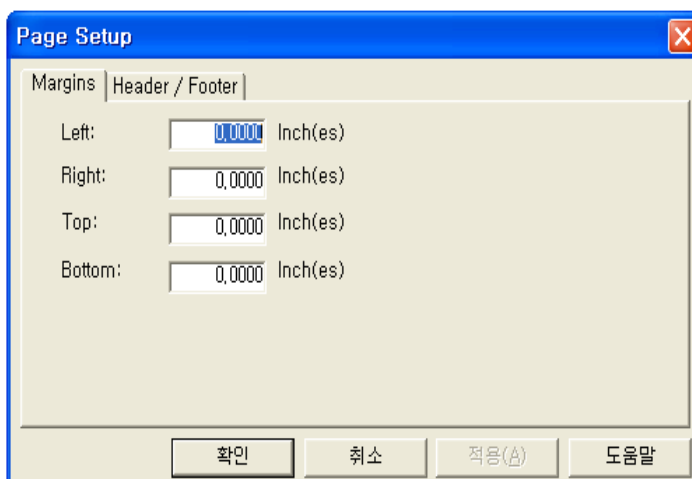
File 메뉴에서 Print...를 선택하거나 툴 바의  아이콘을 클릭하면 아래의 대화상자가 나타나고, “Print All Pages”를 선택하여 전체 페이지를 인쇄하거나 “Print Selected Pages”를 선택한 다음 인쇄할 페이지를 클릭하면 해당 페이지만 인쇄된다.



항 목	설 명
Print Border	Border를 인쇄한다.
Print Color	색을 넣어 인쇄한다.
Print Scaling	Print 스케일을 지정한다.

2.6.2 페이지 마진 설정

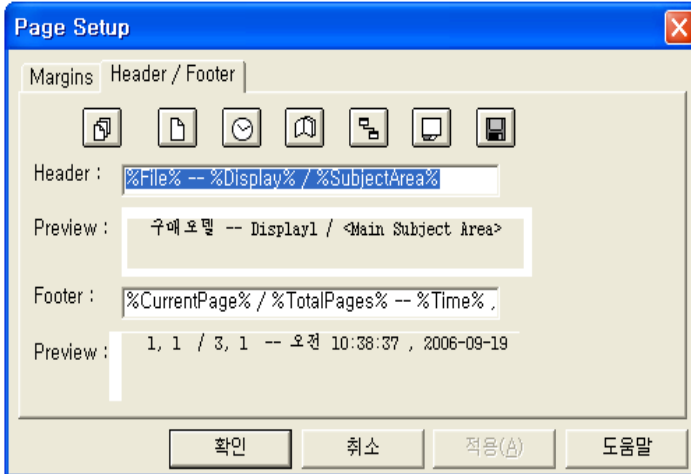
프린트 대화상자에서 Page Setup을 선택하면 페이지 여백과 Header / Footer를 설정 할 수 있다.



Page Setup의 Margins 탭에서는 인쇄 페이지의 상, 하, 좌, 우 여백을 정의한다.

2.6.3 Header / Footer 정의

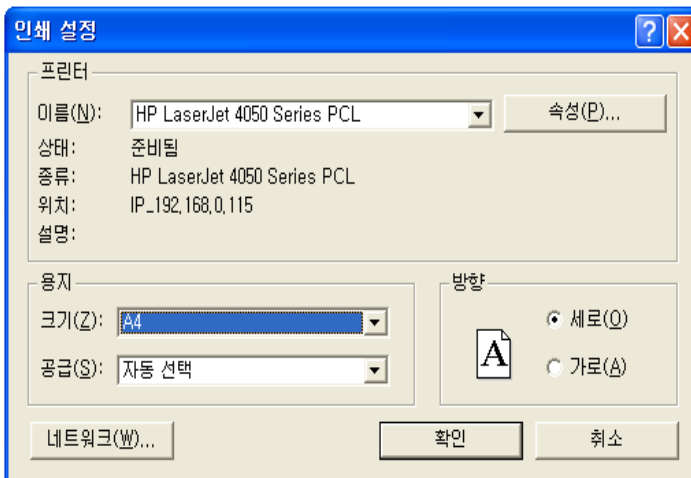
Header나 Footer는 Diagram을 인쇄할 때 페이지의 상단과 하단에 머리말, 꼬리말을 설정한다.



항 목	설 명
Header	머리말을 입력한다.
Footer	꼬리말을 입력한다.
Preview	Header / Footer를 미리 본다.

* Header / Footer는 ERwin에서 제공하는 매크로를 사용해 입력이 가능하며, 사용자가 임의의 문자나 정보를 입력 할 수 있다.

2.6.4 Print Setup Dialog



File > Print Setup을 클릭하면 인쇄 페이지에 대한 인쇄 방향, 프린트 설정, 용지의 크기 등을 설정할 수 있다.

3. Layer 의 이해

3.1 Design Layer Architecture

이것은 데이터 요구사항을 정확히 추출하고 그에 따른 적절한 모델타입을 지원하기 위한 개념이다. 사용자가 가장 첫 번째로 해야 할 일은 모델링의 대상이 무엇인지 알고 결정 해야 하는 것이다. 그 대상을 다음 세가지로 나누어 볼 수 있다.

- ▣ 전사시스템 구축
- ▣ 개별 어플리케이션
- ▣ 다양한 어플리케이션을 수용할 수 있는 전사 모델의 한 부분

다음으로는 사용자가 이와 같은 데이터 요구대상에 맞는 적당한 타입을 선택한다.

- ▣ 개념적인가
- ▣ 논리적인가
- ▣ 물리적인가

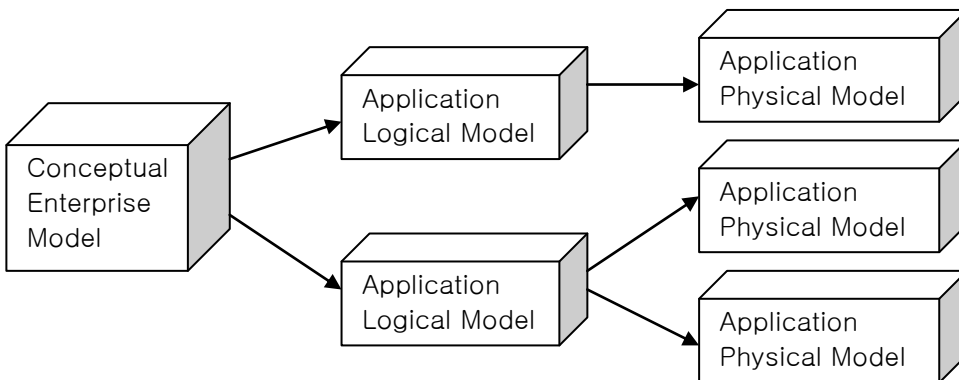
마지막으로는 사용자가 원하는 어플리케이션 타입을 결정한다.

- ▣ Transactional
- ▣ Replicated / Read-only
- ▣ Analytical / Warehouse

데이터 모델링은 프로젝트에서 요구하는 모든 데이터를 추출해 내기 위해 일반적으로 다양한 모델들이 적용된다. 이와 같은 모델들은 서로 다른 목적과 관점을 가지고 있을 수 있다. 즉 논리적 / 물리적 관점에서 프로젝트를 진행할 수 있다. r7에서는 논리와 물리 그리고 복합 모델링을 할 수 있도록 복합 모델을 지원한다.

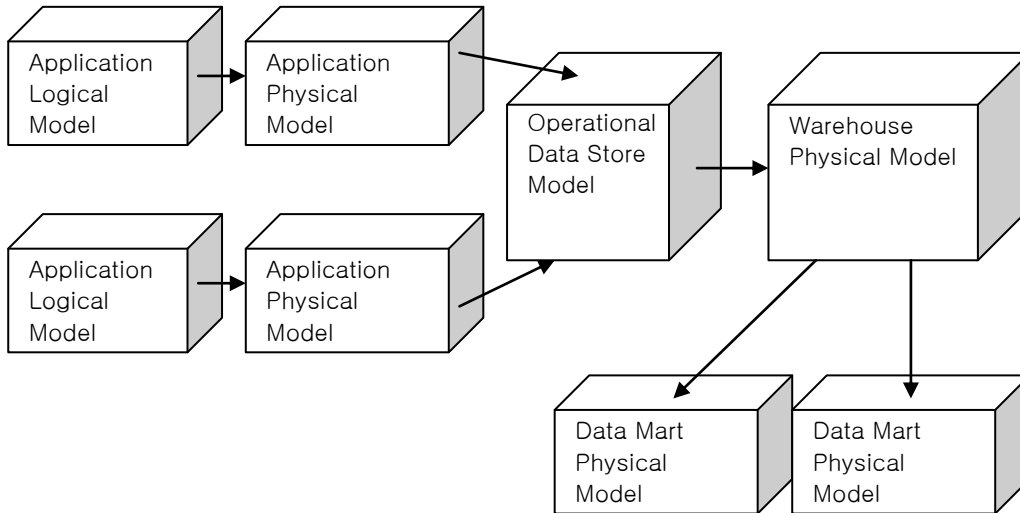
3.2 Design Layer 구조

사용자의 환경에서 적용될 수 있는 모델의 한 세트는 조직화 되어 구조적으로 표현할 있다. ERwin Data Modeler r7 은 각 Layer를 최적화 할 수 있도록 Logical Model과 Physical Model을 분리해 낼 수 있다. 연관되어 있는 Design Layer안에서 각각의 모델들은 링크 되며, 각 Layer의 변경사항은 다른 Layer에도 반영할 수 있도록 지원한다.



3.3 Design Layer 구조의 예

사용자는 많은 Logical Layer들과 많은 Physical Layer들을 관리할 수 있다.



3.4 Design Layer 관리

Design Layers를 관리하기 위해서 사용자는 다음 사항을 결정해야 한다.

- 1) 얼마나 많은 Layer가 필요 한가?
 - 모델들이 서로 다른 목적을 가질 때 여러 Layer로 나누어야 한다.
 - 서로 다른 사용자들을 위해 모델이 개발될 때 나누어야 한다.
- 2) 각 Layer에 대해 누가 담당할 것인가?
- 3) 연관된 Layer들 간의 관계를 관리하기 위한 업무 절차

3.5 Design Layer의 추출 기준

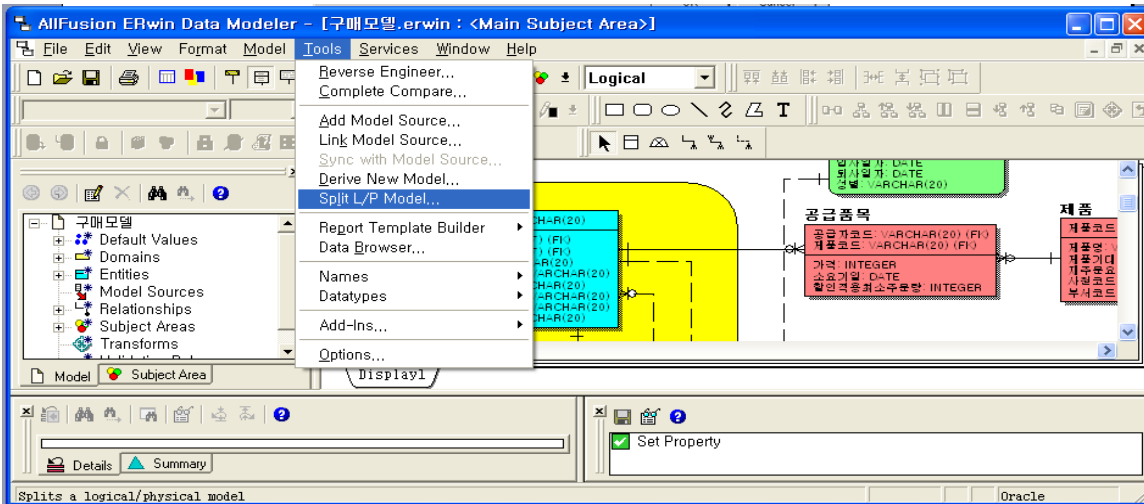
- 1) 하나의 모델은 다른 모델로부터 추출(Derive) 되어 질 수 있다.
 - Logical 모델은 또 다른 Logical 모델로부터 추출될 수 있다.
 - Logical 모델은 또 다른 Physical 모델로부터 추출될 수 있다.
 - Physical 모델은 또 다른 Logical 모델로부터 추출될 수 있다.
 - Physical 모델은 또 다른 Physical 모델로부터 추출될 수 있다.
- 2) 추출된 모델은 Child 모델이라 부른다.
- 3) Parent 모델은 Child 모델의 모델 소스(Source)라 부른다.
- 4) Child 모델은 Parent 모델과 연계된다.(각 Layer의 변경사항은 다른 Layer에도 반영 할 수 있다.)

3.6 Logical / Physical 모델(Classical Data Model Type)

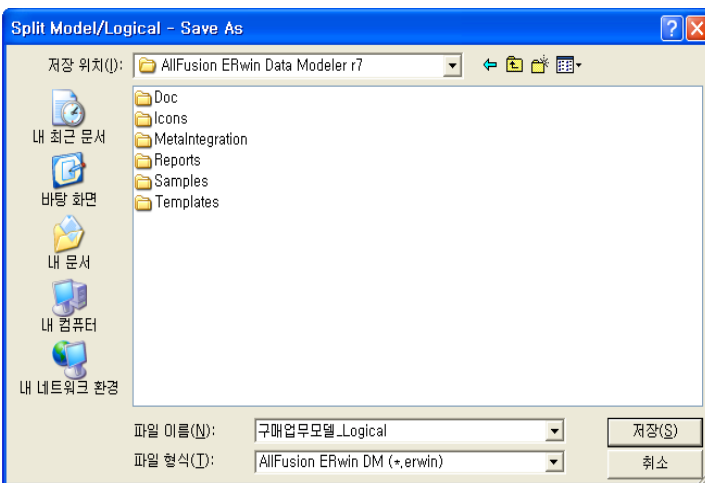
Logical과 Physical이 통합된 모델(ERwin Data Modeler 4.1 이전 버전에서 사용되던 모델)은 다음과 같은 상황에서 사용될 수 있다

- 1) 단 하나의 Target Server만 필요할 때
- 2) Logical모델과 Physical모델간의 차이점이 많지 않을 때
- 3) 모델링 작업이 병렬적이기 보다는 비교적 순차적으로 이루어 질 때
- 4) 사용자는 Logical/Physical모델을 분리할 수는 있으나, Logical모델과 Physical모델간의 링크만 이루어진다. 즉 Logical모델이 Physical모델의 소스가 되는 것이다.

3.7 Logical / Physical모델의 분리작업



① Logical/Physical모델을 열어 Tools메뉴의 Split L/P Model메뉴를 선택 한다.

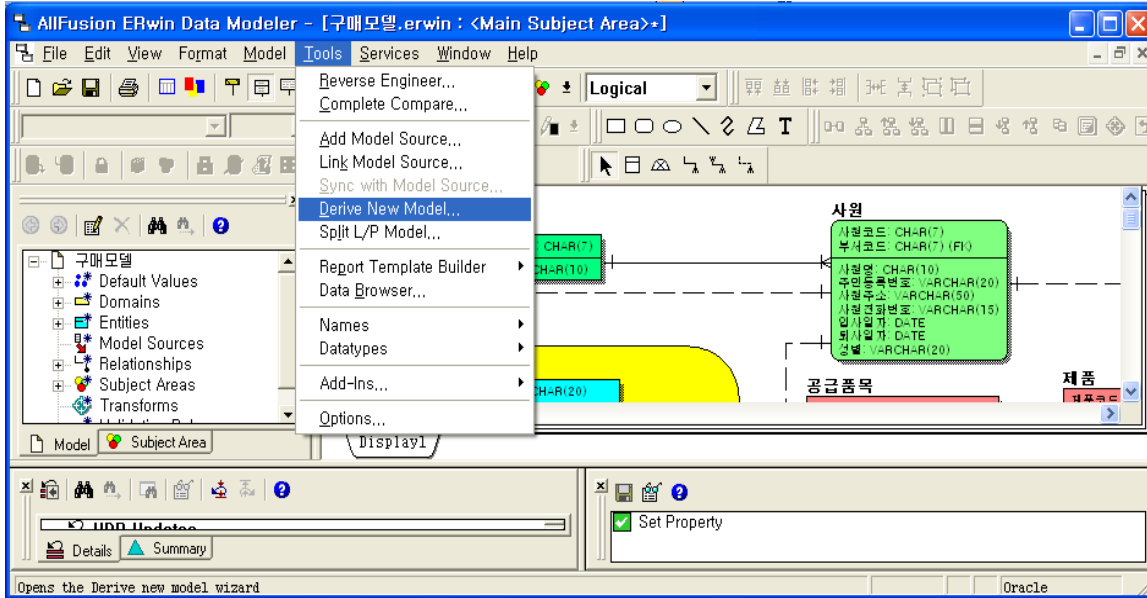


② Split Model / Logical-Save as 다이얼로그가 나타나면, Logical모델을 저장 한다.

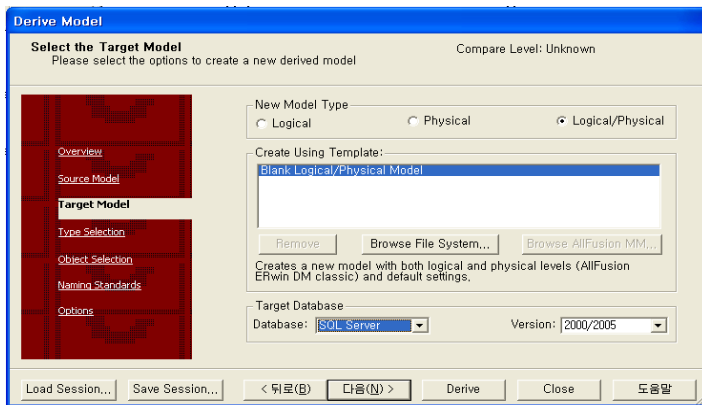
③ Split Model/Physical-Save as 다이얼로그가 나타나면, Physical 모델을 한다.

3.8 새로운 모델을 추출하기(Derive)

아래의 예제는 Oracle로 설정된 Physical모델을 SQL server Physical모델로 추출하는 작업이다.



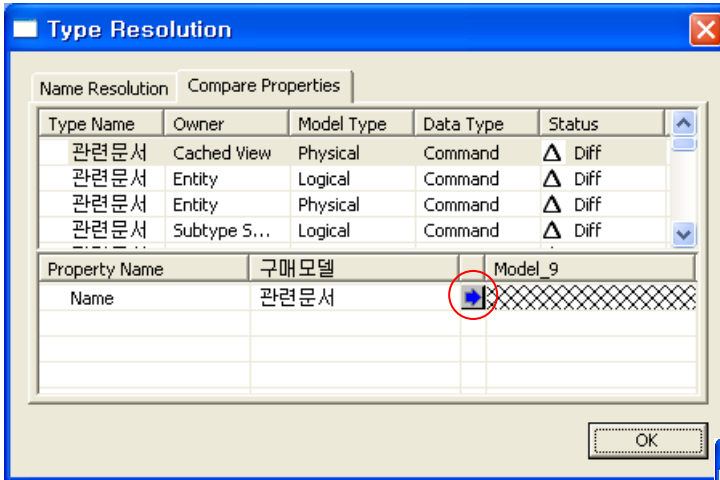
① Target DBMS가 Oracle로 설정된 모델을 Open하여 Tools >Derive New Model을 선택한다.



② 추출할 모델의 Model Type을 선택하고 Target Database를 선택하고 다음을 클릭한다.

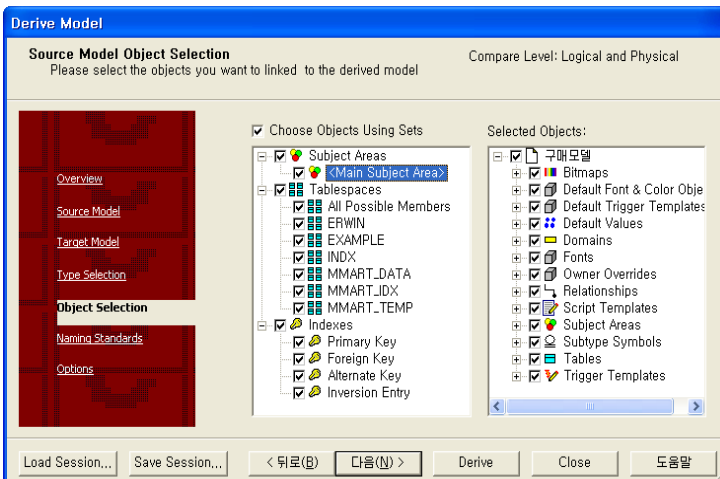
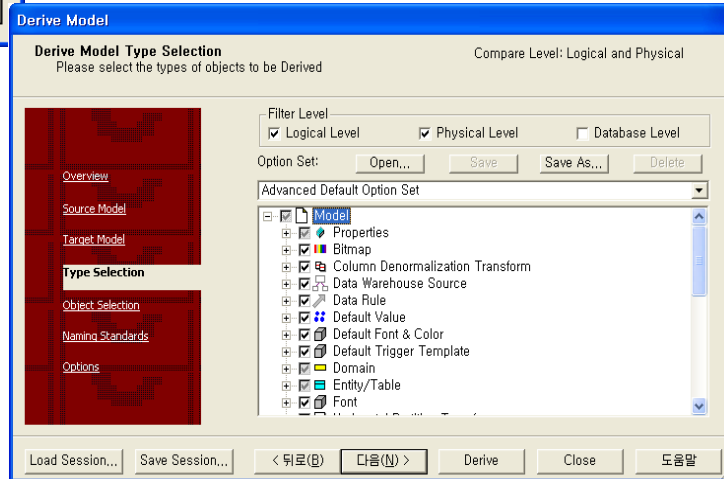
* Derive 기준

4.1.4				R_7			
Layer	Derive			Layer	Derive		
	L/P	L	P		L/P	L	P
L/P	X	O	O	L/P	O	O	O
L	O	O	O	L	O	O	O
P	X	O	O	P	O	O	O

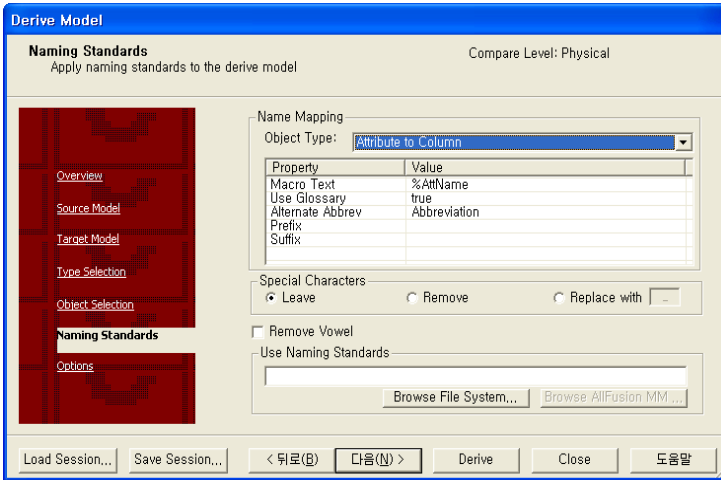


③ 원본 모델에 UDP 속성을 정의 했을 경우는 다음과 같은 화면이 나타난다. 상단 윈도우에서 항목을 선택하면 하단 윈도우에서 Detail한 정보를 확인 할 수 있다. 화살표를 클릭하면 원본 모델에 정의된 UDP 내용을 새로 만들 모델에 반영할 수 있다.

④ 원본 모델에서 추출할 항목을 선택한다. 엔티티, 도메인, 디폴트 등 다양한 모델의 정보를 추출 할 수 있다. 사용자는 추출 옵션을 저장 할 수 있으며 재사용 할 수 있다.

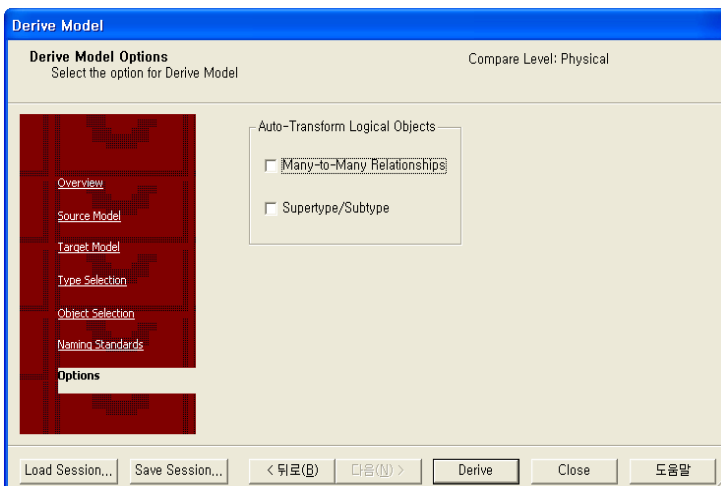


⑤ Source Model의 추출 항목을 필터링 (Filtering)할 수 있다.



⑥ 만약 Logical 모델을 추출하는 작업이었다면 ‘다음’ 버튼을 클릭했을 경우 ‘Naming standards does not apply to the current Source and Target model types!’ 라는 메시지가 표시되고 Physical 모델을 추출하려고 했다면 다음과 같은 화면이 나타난다.

Use Naming Standards는 Logical Name을 Physical Name으로 자동변환 시켜주기 위해 Naming Standards 파일을 매핑 시키거나 Name Rule을 정의 할 수 있다. 자세한 사항은 Naming Standard Chapter를 참고한다.



⑦ 마지막으로 Derive Model Options 화면이 나타난다. 여기서 다음과 같은 옵션을 최종적으로 선택한 후 Derive버튼을 클릭하여 새로운 모델을 추출할 수 있다.

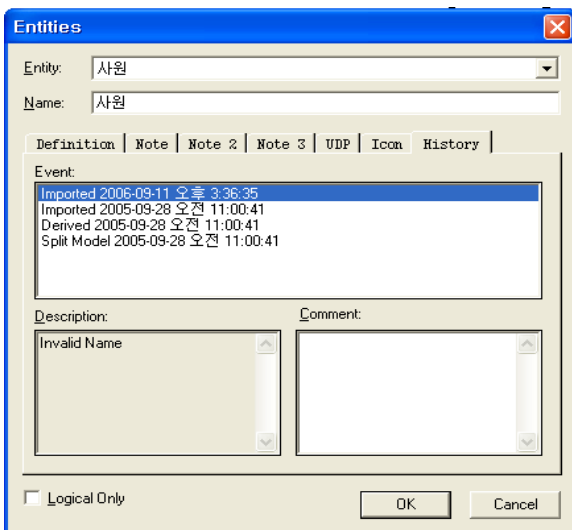
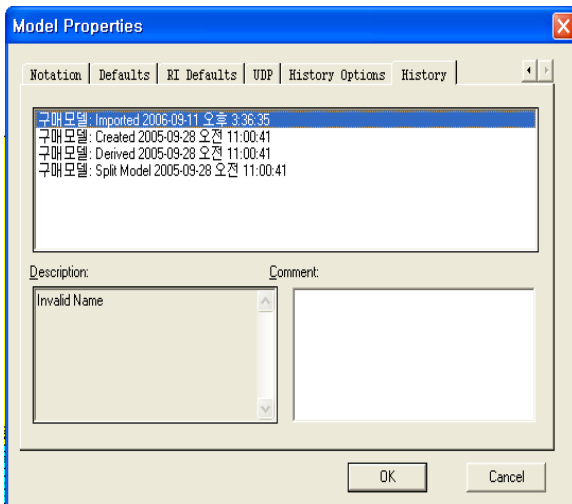
항 목	설 명
Many-to ...	M:M 관계를 연관테이블을 생성 하여 자동 변환한다.
Supertype ...	일반화 관계를 식별관계로 자동변환 한다.

4. History 정보 관리

다양한 Design Layer들은 모델 내에 존재하는 Object들의 출처가 어디에서부터 추출되었는지에 대한 관리가 요구된다. 추출된 모델마다 Model메뉴의 Model Source메뉴에 나타나 있으며, Table, Column Editor에 History 탭 페이지에서도 확인할 수 있다.

다음은 History 정보가 생성되는 작업 내용이다

- 1) 현재 모델에서 Object를 생성했을 때
- 2) 모델 소스에서 모델을 추출했을 때
- 3) 데이터베이스에서 모델을 불러왔을 때(Reverse Engineering)
- 4) Transform 기능으로 모델을 변환 했을 때



Logical Data Modeling



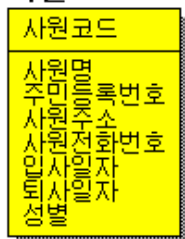
5. Logical Modeling

5.1 Entity, Attribute, Key 개념

5.1.1 Entity 와 Attribute

Entity 란
기업이 유지하고자 하는 데이터
로 사람, 사물, 사건, 장소, 개념

사원



Attribute 란
데이터가 유지 되어지는
엔티티의 뚜렷한 특성

각 Entity는 인스턴스라 불리는 개별 객체의 집합으로 표현한다. 각 인스턴스는 유일하며, 다른 인스턴스와 구별되는 속성을 가져야 한다. 한 인스턴스는 Entity의 견본이라 할 수 있다.

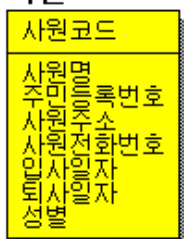
다음 질문들은 Entity 라고 정의할 수 있도록 도와줄 것이다.

1. 해당 Object는 특정 업무영역에서 중요성을 가지고 있는가?
2. 업무영역에서 사용할 수 있는 특징이 있는가?
3. 해당 Object는 유일하게 구분될 수 있는가?
4. 해당 Object는 다른 Object와 관계를 가지고 있는가?

5.1.2 Primary Key

Primary Key 란
Entity의 인스턴스를 유일하게 식별하도록 하는
하나의 Attribute 혹은 여러 Attribute들의 집합

사원



Primary Key는 항상
Entity의 구분선 위에
위치한다(PK 영역)

Entity 개념에서 살펴본것과, 각 인스턴스는 유일하게 식별되어야 한다. 식별자라고 하는 것은 하나 또는 그 이상의 특징, 혹은 Attribute의 값으로 이루어져 있는 것을 말하는데, 우리는 이것을 유일한 식별자(Unique Identifier) 혹은 Primary Key 라고 부른다.

Primary Key는 다음과 같은 특성을 가진다.

- 반드시 유일해야 한다. (Unique)
- null값을 가질 수 없다. (Not Null)
- 유용하고 쓸모 있는 최소한의 Attribute로 구성되어져야 한다. (Short)
- 정적인 값을 가져야 한다. (Static)

5.1.3 Alternate Key(대체키)

사원

사원코드
사원명 (IE1)
주민등록번호 (AK1)
전화번호
주소
성별
입사일자
퇴사일자

Alternate Key 란
 개개의 인스턴스를 유일하게 식별하게
 하지만, Primary Key로 선택되지
 않은 Attribute나 Attribute의 집합

Alternate Key는 Entity의 인스턴스를 구별하기 위한 Attribute의 집합으로 쓰여지나, Primary Key와 같지는 않다. 개개의 인스턴스를 유일하게 식별하게 하지만, Primary Key로 선택되지 않은 Attribute 나 Attribute의 집합을 말한다. Primary Key와 다른점은 Alternate Key의 Attribute는 Null값을 포함할 수 있다는 것이다.

각 Alternate Key Attribute는 AKx.y로 표시된다. 'x'와 'y'의 순서가 중요하다. 'x' 값의 의미는 엔티티내의 Alternate Key 순서이며, 'y'는 각 Alternate Key내의 Attribute 순서 정보이다. Alternate Key가 Index로 될 때에 두 번째 값인 'y'는 Index의 column 순서를 의미한다. 예를 들어 "book title"은 첫번째 Alternate Key의 세 번째 Attribute이며, "book lc catalog number"는 두 번째 Alternate Key의 첫 번째 Attribute가 된다.

5.1.4 Inversion Entry

사원

사원코드
사원명 (IE1)
주민등록번호 (AK1)
전화번호
주소
성별
입사일자
퇴사일자

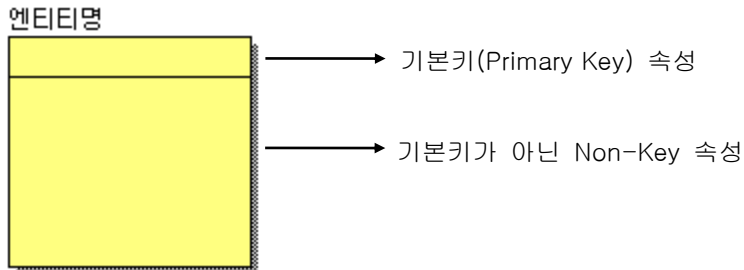
INVERSION ENTRY 란
 하나 또는 그 이상의 Entity 인스턴스를
 접근하는데 자주 사용되는 하나이상의 Attribute

Inversion Entry는 인스턴스를 찾는데 이용되는 Attribute의 그룹이다. 이것은 더 많은 요구사항을 받아들일 수 있으며 Physical 모델링 시 도움을 준다. (ex : 조회 시 많이 사용하는 '이름')

5.2 Entity 생성

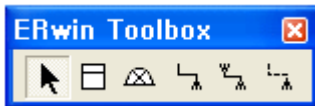
5.2.1 엔티티의 구성

엔티티는 Primary Key Attribute와 Non Key Attribute로 구성된다.



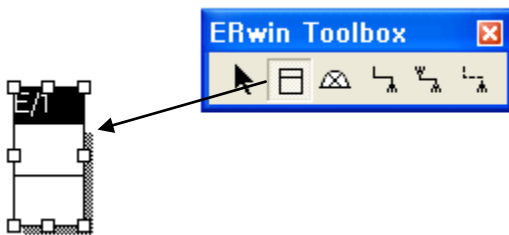
5.2.2 엔티티에 대한 툴박스의 지원

ERwin의 툴박스는 엔티티를 그릴 수 있는 툴을 제공한다.



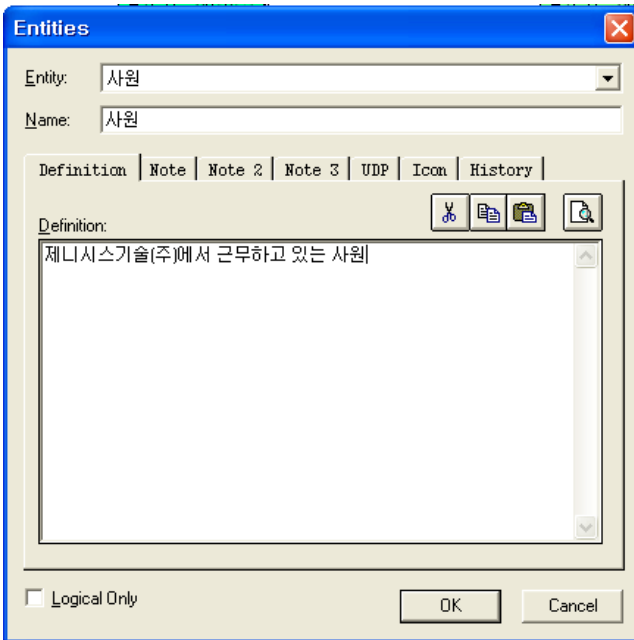
5.2.3 엔티티(Entity) 툴의 사용

툴 박스의 엔티티 툴을 마우스 오른쪽 버튼을 클릭한 후 원하는 위치에 클릭을 하면 엔티티가 Diagram 상에 나타난다.



- 1) Toolbox의 엔티티 툴을 선택한다.
- 2) 마우스의 커서가 Entity 모양으로 바뀐다.
- 3) Diagram의 원하는 위치에 마우스를 클릭하면 Entity가 생성된다.
- 4) Entity 이름을 입력한다.
- 5) Entity 툴을 다시 선택하여 Entity를 추가한다.
- 6) Shift + Entity 툴을 선택하면 다수의 엔티티를 생성 할 수 있다.

5.2.4 Entity Editor 이용하기



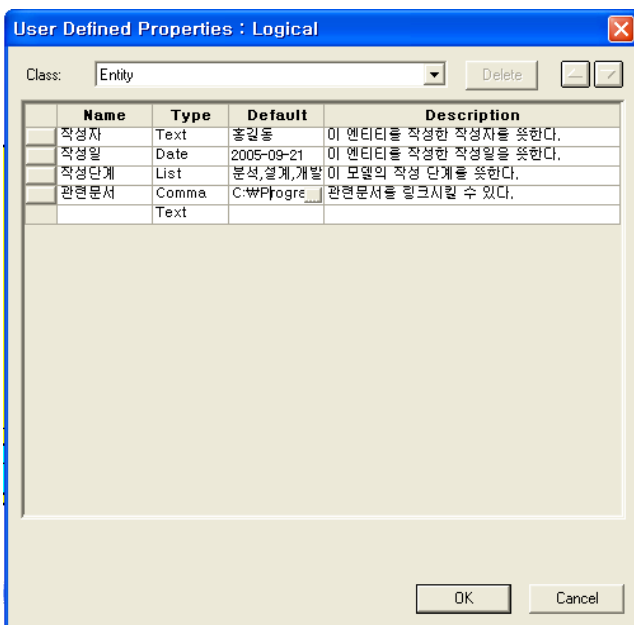
Entity를 선택하고 마우스 오른쪽을 클릭하여 Entity Properties를 선택한다.

Entity Editor에서는 아래와 같은 정보를 입력한다.

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

이 Editor의 용도는 Entity 이름 Rename, Entity 정의 또는 Note 정보 입력, Entity 별 UDP 값을 입력 시 사용한다. Definition, Note 탭 페이지에 해당 Entity의 정의 및 그 외 추가 사항들을 입력한다.

UDP는 사용자가 Definition, Note 이외에 Editor를 추가하고자 할 때 이용할 수 있으며, 단순히 Text를 입력하는 Editor 외에 여러 타입을 지원한다.



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.

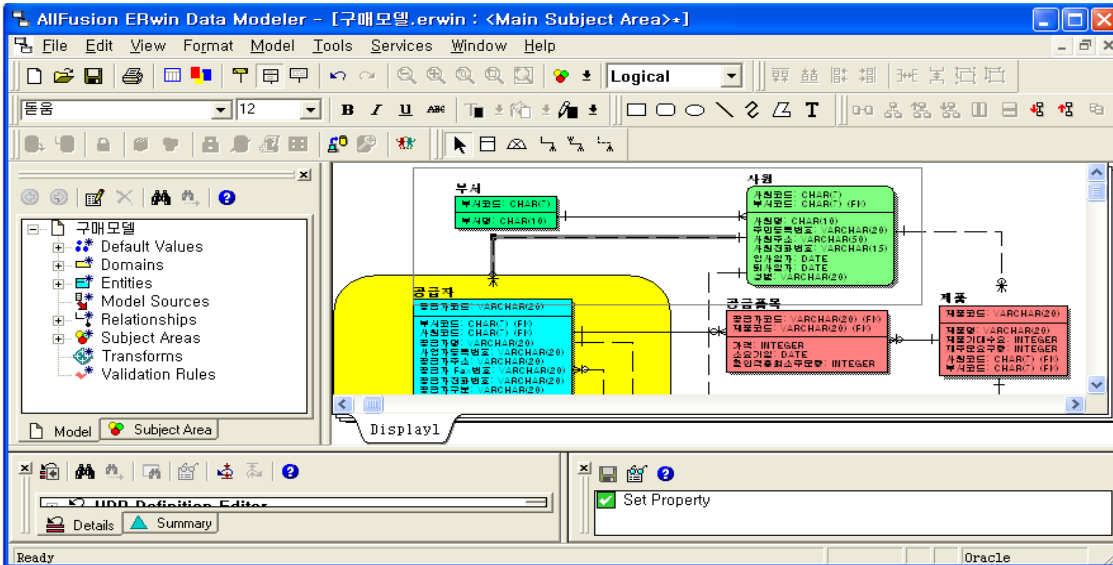
List Type 사용시 콤마(1년, 2년)로 구분하며 설정한 UDP 항목은 모든 엔티티에 동일하며, 값을 입력하면 된다.

5.3 Entity의 이동

1) Entity 의 이동 : 해당 엔티티를 클릭한 다음, 옮기고자 하는 위치로 드래그 한다.

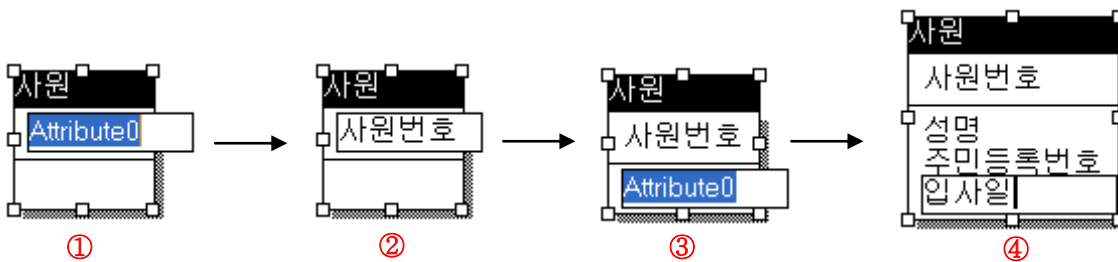
2) 라소(Lasso) 기법

툴 바의 화살표를 먼저 선택하고, 원하는 객체(엔티티나 텍스트 박스 등)를 드래그하여 선택 하면 아래의 그림과 같이 여러 개의 객체가 선택된다.



3) Shift-Click : Shift를 누른 상태에서 마우스를 클릭하면 다중 엔티티를 선택할 수 있다.

5.4 Attribute 생성



1) Entity를 선택한다.

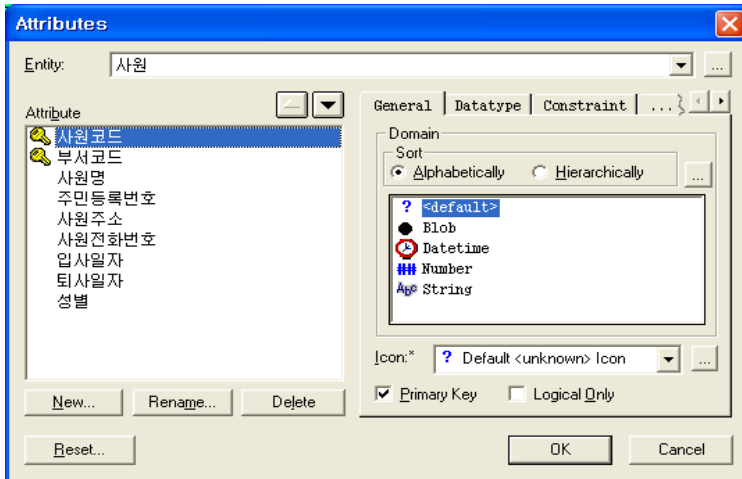
2) Tab키를 눌러 Attribute 이름을 입력한다. Primary Key Attribute가 여러 개인 경우는 Enter키를 눌러 Attribute를 추가한다.

3) Tab키를 눌러 Non Key영역으로 이동 후 Attribute 이름을 입력한다. Enter키를 이용하여 Attribute를 계속 추가한다.

4) Tab키를 이용하여 Entity Name영역, Primary Key 영역, Non Key영역을 이동할 수 있다.

5.4.1 Attribute Editor 이용하기

Attribute Editor에서는 Attribute를 생성할 수 있으며, 도메인을 할당하거나, Definition, Note, UDP를 입력할 수 있다. 또한 Key Group을 등록할 수 있다.



① 엔티티를 선택 후 오른쪽 버튼을 클릭하고 Attributes를 선택한다.

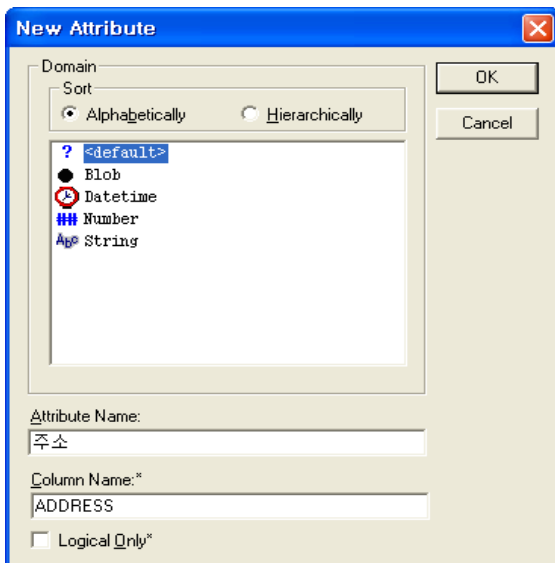
항 목	설 명
New	새로운 속성을 만든다.
Rename	속성명을 변경한다.
Delete	속성을 지운다.
Reset	속성을 초기화 한다.

② General 탭 페이지에서는 해당 Attribute마다 Logical Domain을 할당하고, Primary

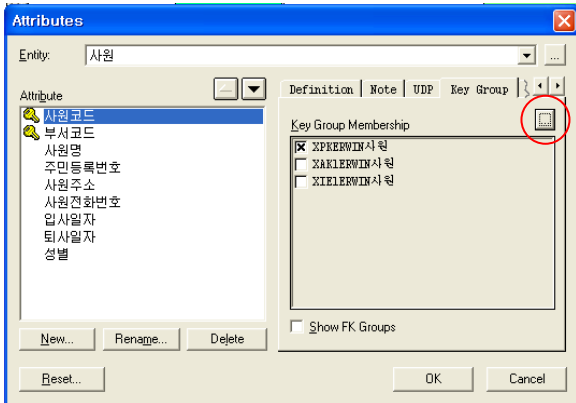
Key 체크박스를 이용하여 Attribute에 PK를 할당할 수 있으며, Logical Only 체크박스를 선택하면 Physical Model에서 해당 속성이 보이지 않도록 한다.

③ Datatype 탭 페이지에서는 속성의 데이터 타입을 지정해 줄 수 있다. 하지만 데이터 타입 작업은 주로 Physical Modeling 작업 시에 진행하며, 여기서 선택한 데이터 타입은 개념적인(논리적인) 데이터 타입입니다.

④ Constraint 탭 페이지에서는 Validation Rule, Default Value를 생성 할 수 있다.



⑤ Editor창에서 Attribute를 새로 생성하려면 New 버튼을 클릭한다 Attribute이름과 Column 이름을 함께 입력하여 Attribute를 생성시킨다.



버튼을 클릭하여 Key Group 타입을 선택하면 이름이 생성되며, Key Group 이름은 수정할 수



데이터베이스의 성능을 높이기 위하여 하나의 테이블에 여러 개의 인덱스(Index)를 설정할 수 있다. 기본키 (Primary Key)는 자동적으로 인덱스가 생기며 Alternate Keys나 Inversion Entry를 지정해줌으로써 인덱스로 활용할 수 있다.

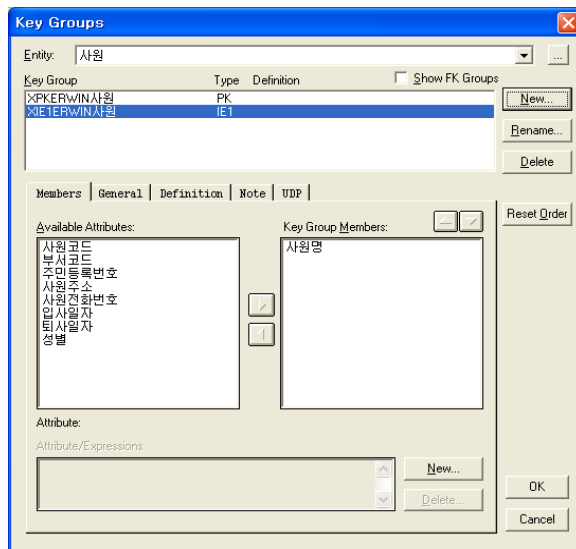
Alternate Keys는 어떤 인스턴스(Instance)를 유일하게 식별할 수 있는 하나 또는 그룹으로 묶여진 속성이다. 만약 어떤 Entity가 각각의 인스턴스를 유일하게 식별할 수 있는 하나 이상의 Attribute를 가지고 있다면 Alternate Keys로 지정할 수 있다. (ex : 주민등록번호)

▣ 역 엔트리(Inversion Entries)의 지정

Inversion Entries는 어떤 Entity의 인스턴스를 유일하게 식별할 수 없는 Attribute 이지만, Entity의 인스턴스에 접근하기 위하여 가끔 사용된다. (ex : 이름)

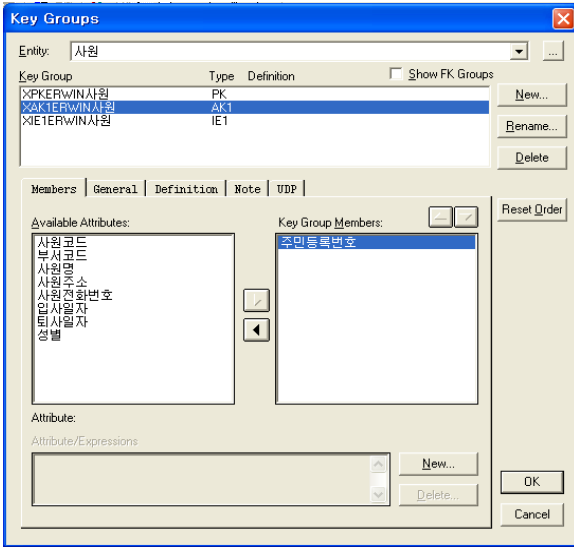
⑥ Key를 등록하려면 Key Group 탭 페이지를 선택하고 ... 버튼을 클릭하여 Key Group Editor로 이동한다.

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



⑦ New Default 있다.

▣ Alternate Keys(AK)의 설정

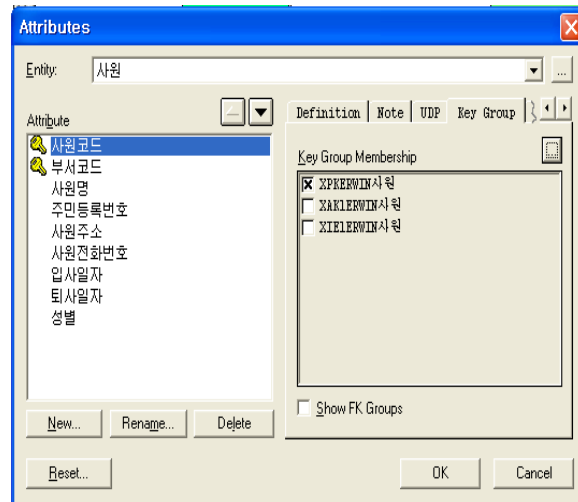


⑧ 생성된 Key Group에 포함시킬 Attribute를 선택하고 Key Group Members로 이동시킨다.

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

Group을 수정 하려면 엔티티 위에서 오른쪽 클릭 후 Attribute Editor 선택하고 그림과 같이 Group 탭을 클릭한다.

속성을 선택한 후 Key Group Membership에서 선택 / 해제를 통해 등록 및 변경, 삭제가 가능

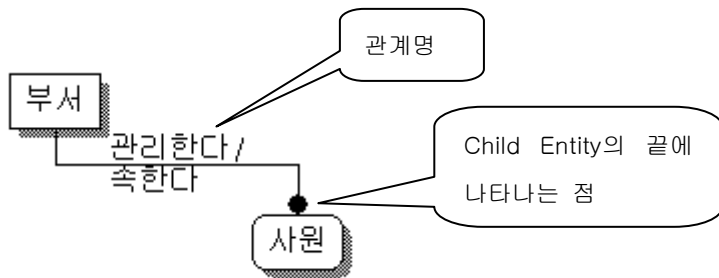


⑨ Key 마우스 Key

옵션을 한다.

5.5 Relationship 개념

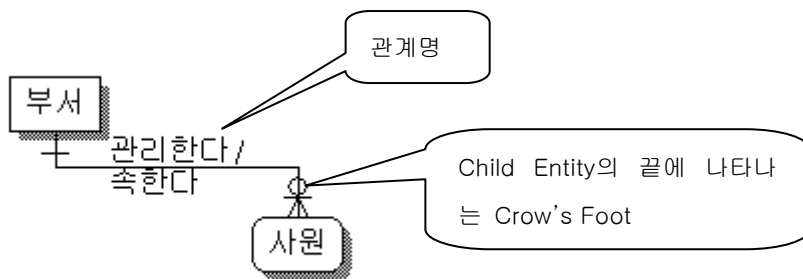
Relationship은 Business Rule이 정의된 것이다. 이 Rule은 결국 Database에 의해 참조 무결성 규칙에 적용된다.



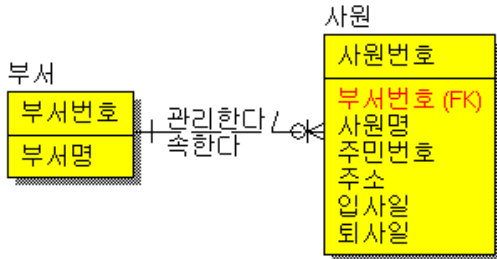
모든 Relationship은 Parent Entity와 Child Entity가 있다. Child는 항상 Relationship선 끝에 점이 존재한다. 위의 Entity간의 관계를 다음과 같이 읽을 수 있다.

각 부서는 여러 사원을 관리한다. / 각 사원은 한 부서에 속한다.

아래 그림은 IE(Information Engineering) 기법으로, 위의 그림과 같은 개념의 Relationship이다. 단지 Diagram상에서만 다르게 표현된다.



5.5.1 Foreign Key (FK, 외래키)



Entity가 Relationship에 의해 연결될 때

- ERwin은 Parent Entity의 Primary Key Attribute(s)를 Relationship을 통해 Child Entity의 Foreign Key Attribute(s)로 이주(migrate)시킨다
- ERwin은 이동작업을 자동으로 수행한다
- ERwin은 Attribute이름 뒤에 나타나는 (FK)표시로 Foreign Key를 식별한다.

5.5.2 Relationship Cardinality

Entity간의 관계를 “각 Parent Entity”, “0,1,그이상의 Child Entity”라고 말하는 것은 실제로 “각 Parent Entity의 인스턴스”, “0,1,그 이상의 Child Entity의 인스턴스”를 가리킨다.

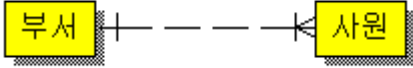
IDEF1x방법에서의 기수성(Cardinality)은 Parent당 Child 인스턴스의 개수를 의미하고 Child당 Parent 인스턴스의 개수는 선택성(Optionality)에 의해 결정 된다.

쉽게 설명하자면 한개 부서에 한명의 사원만이 지정되는 관계는 1:1이 되며, 한개 부서에 한명 이상의 사원이 지정되면 1:M(Many)이 되는 것이다. M:M의 경우는 한개의 부서에 한명 이상의 사원이 지정되며, 한명의 사원이 여러 부서에 동시에 소속될 수 있다는 의미가 된다.

다음 ERwin에서 카디널리티(Cardinality)를 어떻게 표현하는지 보겠다.

표기법을 정보공학기법(Information Engineering)으로 선택했을 경우 카디널리티가 1로 표현되는 Entity 쪽은 선으로 그리고, M으로 표현되는 Entity 쪽은 까마귀 발(Crow's Foot)과 같이 그려진다. 또한 관계로 표현된 점선의 양단에 ○ 기호를 확인 할 수 있는데, 이 기호는 선택성을 표시한 것이다. 즉, ○이 붙어 있는 경우를 선택(Optional), 그렇지 않은 경우를 필수(Mandatory)라고 표현하는데, 예를 들어 사원 Entity 쪽에 ○이 붙어 있는 경우는 각 부서에 사원이 한명도 지정되지 않을 수도 있다는 의미이고 반대로, 부서 Entity 쪽에 ○이 붙어 있는 경우는 사원 중에서 어떠한 부서에도 소속되지 않은 사원이 있을 수 있음을 의하는 것이다.

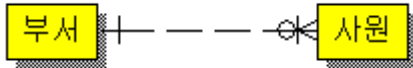
이 ERD는 부서와 사원 Entity 간의 관계가 1:M이며 선택도를 표현하는 ○이 양쪽 모두 누락된 경우이다.



좌 → 우 : 각 부서는 한명 이상의 사원이 반드시 소속된다.(Mandatory)

우 → 좌 : 각 사원은 한 개 부서에 반드시 배정된다.(Mandatory)

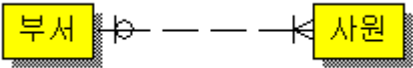
이 ERD는 사원 Entity에 ○이 추가된 경우이다.



좌 → 우 : 각 부서는 한명 이상의 사원이 소속될 수도 있다.(Optional)

우 → 좌 : 각 사원은 한 개 부서에 반드시 배정된다.(Mandatory)

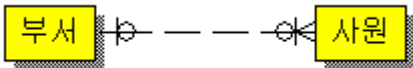
이 ERD는 부서 Entity에 ○이 추가된 경우이다.



좌 → 우 : 각 부서는 한명 이상의 사원이 반드시 소속된다.(Mandatory)

우 → 좌 : 각 사원은 한 개 부서에 배정될 수도 있다. (Optional)

이 ERD는 부서 Entity와 사원 Entity에 ○이 추가된 경우이다.



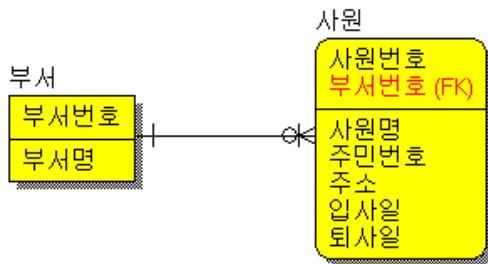
좌 → 우 : 각 부서는 한 명 이상의 사원이 소속될 수도 있다.(Optional)

우 → 좌 : 각 사원은 한 개 부서에 배정될 수도 있다.(Optional)

5.5.3 Relationship 종류

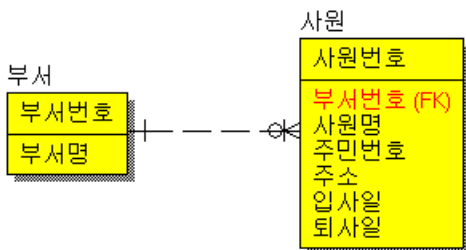
ERwin에서는 다음과 같은 방법으로 식별(Identifying), 비식별(Non-Identifying)을 표현한다.

1) Identifying Relationship(식별관계)



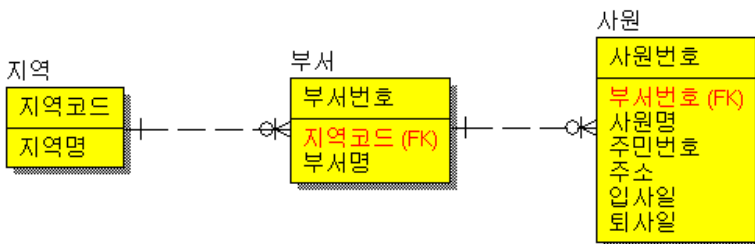
위와 같이 식별관계는 실선으로 표기하며 부모 Entity의 기본키 속성이 자식 Entity인 사원의 외래키(Foreign Key)로 구현되는 과정에서 사원의 기본키(Primary Key) 속성으로 추가되는 것이다. 다시말해 Child Entity는 의존 Entity이며, 각 인스턴스를 식별하기 위해 다른 Entity로부터 정보가 요구되어야 한다. 그리고 ERwin에서 Child Entity의 모양은 둥근 사각형으로 바뀌게 된다.

2) Non-Identifying Relationship(비식별관계)



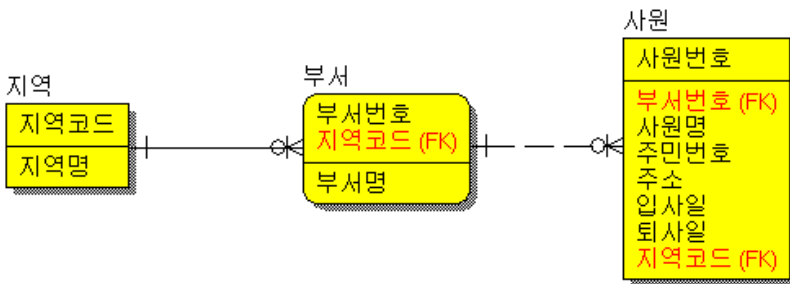
위와 같이 비식별관계는 점선으로 표기하며 부모 Entity의 기본키(Primary Key) 속성이 자식 Entity인 사원의 외래키(Foreign)로 구현되는 과정에서 일반 속성(Non Key)으로 추가된다는 의미이다.

3) Identifying Relationship(식별관계) 와 Non-Identifying Relationship(비식별관계)의 차이점



위 ERD는 지역 Entity, 부서 Entity, 사원 Entity로 구성되어 있으며 각각 1:M 관계로 설정되어 있다. 즉, 지역 Entity의 기본키인 지역코드가 부서 Entity의 외래키(일반속성)으로 추가되었고, 부서 Entity의 기본키인 부서코드가 사원 Entity의 외래키(일반속성)로 추가되었음을 확인할 수 있다. 이러한 ERD에서 사원의 이름을 검색조건으로 지역명을 검색하고자 한다면 Entity를 모두 조인(Join)하여야 한다.

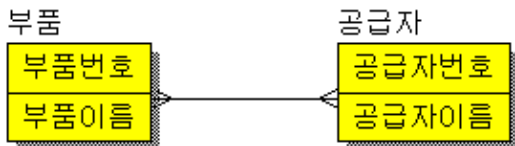
반면, 다음 ERD와 같이 지역 Entity와 부서 Entity간의 식별관계로 구성된 경우는 사원 Entity에 지역 Entity와 부서 Entity의 기본키가 각각 포함되므로 지역 Entity와 사원 Entity가 직접 조인(Join)이 가능해지기 때문에 위보다 수행성능이 향상될 것임을 예상할 수 있다.



4) Many-to-Many Relationship(다 대 다 관계)

아래의 업무규칙으로 다음과 같은 관계가 나올 수 있다.

- 각 부품은 0,1,그 이상의 공급자로부터 주문된다
- 각 공급자는 0,1,그 이상의 부품을 우리에게 공급한다



Many-to-Many Relationship에서는 두개의 관계명을 필요로 하며, 거기에는 상당한 의미의 내용이 담겨져 있다. 이것은 또한 실제로 Many-to-Many관계가 아니라는 것을 확인하기 위해 각 대상 Entity로 부터 1:M관계 처럼 읽어야 하는 것이 중요하다.


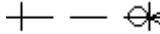
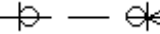

Logical Modeling단계에서는 M:M관계를 완전히 허용하고 있으며, DBMS로 생성되기 전에 해결되어져야만 한다. 만약 M:M관계를 두개의 간단한 1:M관계로 결론짓는다면, Logical Model에서 M:M관계를 그대로 두는 것이 가능하고 대신, Physical Model에서 해결한다.

M:M관계에서 Parent/Child Entity를 구분 할 수 없다 하더라도, 관계명의 순서가 그것을 말해 준다.

(Parent : Child / Child : Parent)

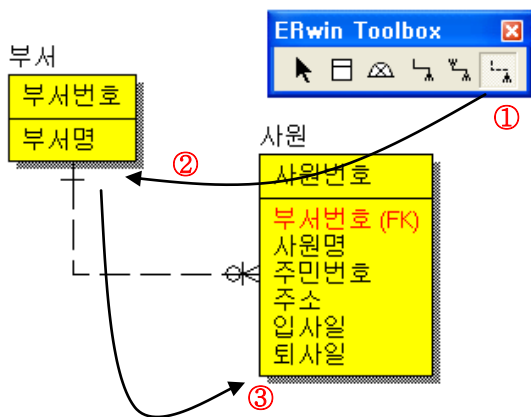
5.5.4 Relationship 타입 정리

1) Relationship의 종류

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

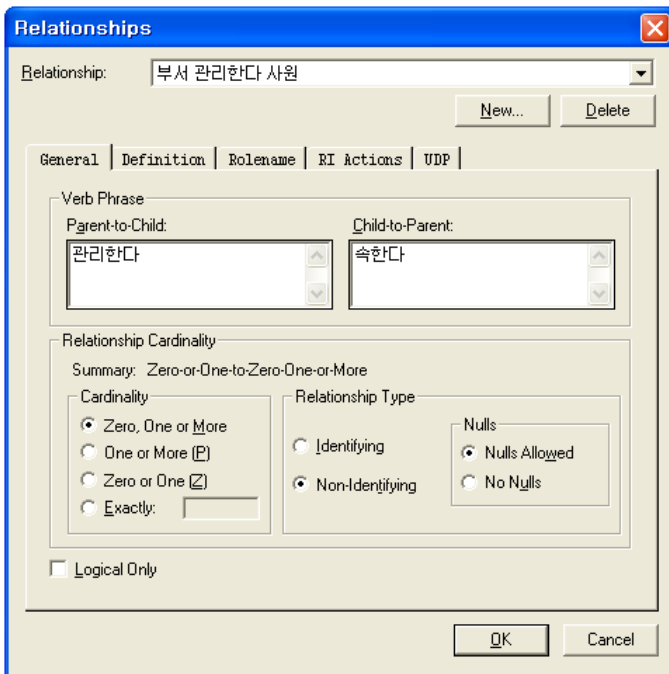
5.5.5 Relationship 생성하기

- 1) Toolbox에서 원하는 Relationship을 선택한다
- 2) Parent Entity를 클릭한다
- 3) Child Entity를 클릭한다
- 4) 관계선은 마우스로 선택, dragging하여 재배치 할 수 있다.

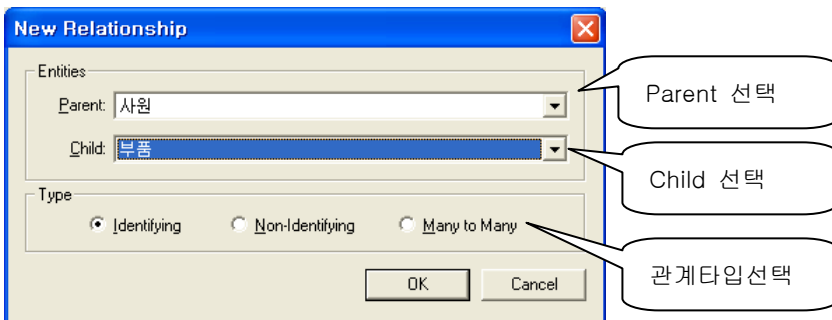


5.5.6 Relationship Editor이용하기

1. 관계선에서 마우스 오른쪽 클릭하여 Relationship Properties를 선택한다.
2. Relationship Editor에서 General 탭 페이지를 선택한다
 - 1) Verb Phrase를 입력한다.
 - 2) Cardinality를 변경한다.
 - 3) Relationship타입을 변경한다.
(Identifying/Non-Identifying, Nulls Allowed/No Nulls)
 - 4) Logical only를 체크 할 수 있다.



3. New 버튼을 클릭하면 아래의 그림이 나온다. 여기에서 새로운 관계를 설정해 줄 수가 있다.



5.5.7 Relationship의 Issues → One-to-Many Relationship

1. Unification(단일화)

한 Attribute가 Foreign Key로서 다른 Entity로 이동해 갈 때 이미 그것과 동일한 이름의 Attribute가 Entity내에 존재한다면, 두 개는 하나의 Attribute로 단일화 된다. 이 현상은 ‘한 object가 같은 이름을 가지고 있다면 그것은 동일하다’는 IDEF 규칙에 기인한다. IDEF 틀이 아닌 것들은 일반적으로 Unification을 언급하지 않는다.

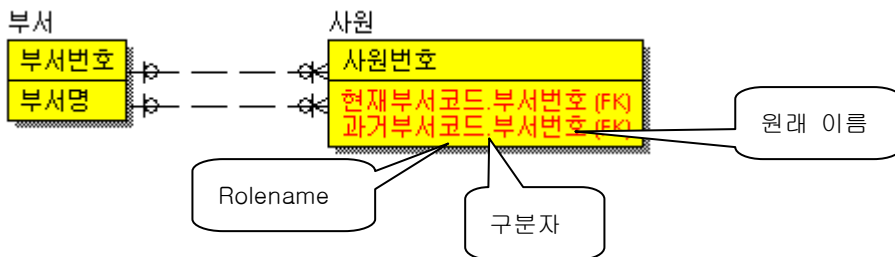
2. Rolename Foreign Key(역할이름)

Foreign Key Attribute의 역할이름(Rolename) 부여는 Attribute의 뜻을 명확히 하는데 도움을 준다.

또한 역할이름은 서로 다른 업무적인 개념을 표현하기 위해 두개의 관계선을 표현할 필요가 있을 때 원하지 않는 단일화 현상을 막아준다.

Logical 모델에서 생성한 역할이름은 Physical 모델의 Column명으로 전환된다.

역할이름은 원래 이름(Base Name)을 포함하고 있다. 예를 들어 과거부서코드.부서코드와 같이 역할 이름 내에 원래 이름을 보여준다면 FK이동으로 인한 혼란을 막을 수가 있다.

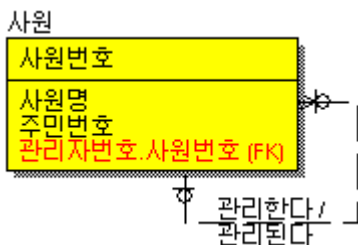


3. Recursive Relationship (재귀적 관계)

재귀관계는 한 Entity가 자기 자신과 관계를 맺는 Non-Identifying Optional Relationship 이다.

아래 그림에서 대부분의 사원은 자신의 관리자가 있다. 각 관리자는 또한 사원이다.

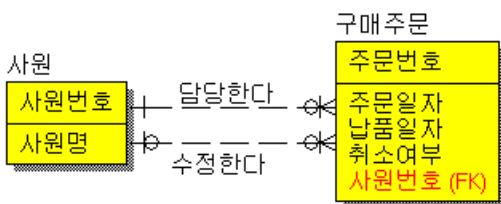
결론적으로, Parent Entity(관리자들)는 Child Entity(모든 사원들)의 부분 집합이다.



5.5.8 Foreign Key의 단일화 현상 예제

1. 사원 Entity와 구매주문 Entity를 생성한다
2. 업무적인 개념이 담겨있는 두개의 Non-Identifying Relationship을 생성한다.

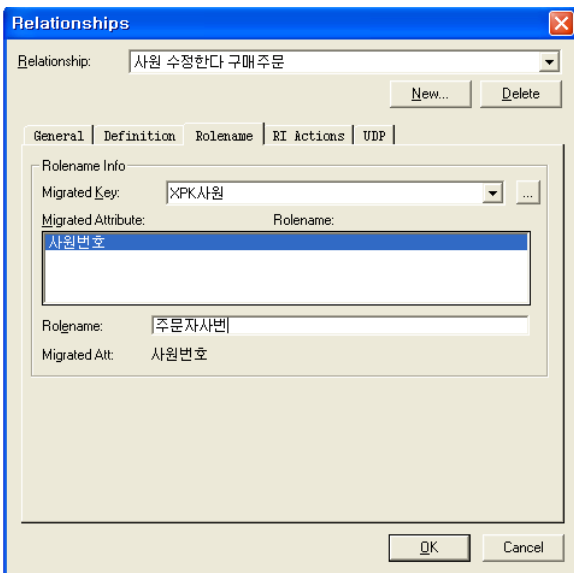
- ▣ 각 사원은 0,1,그 이상의 구매주문을 담당한다/각 구매주문(서)는 사원에 의해 기입된다
- ▣ 각 사원은 0,1,그 이상의 구매주문을 수정한다/각 구매주문(서)는 사원에 의해 수정될 수도 있고 아닐 수도 있다.



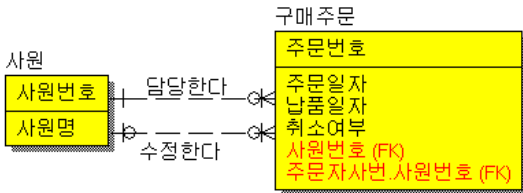
3. 구매주문 Entity로 이동된 사번이 사번(FK)으로 단일화된다

Foreign Key에 역할이름 부여하기

1. 관계선에서 마우스 오른쪽 버튼 클릭하여 Relationship Properties메뉴를 선택한다.
2. Rolename 탭페이지에서 '사번' 속성 Rolename을 '주문자 사번'이라고 입력한다
3. 또 다른 Relationship Editor에서 '사번'속성의 Rolename을 '수정자 사번'으로 입력한 뒤 OK버튼을 클릭하여 빠져 나온다.

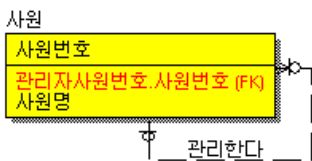


4. 바탕 화면에서 마우스 오른쪽버튼 클릭하면 Pop-up메뉴가 나타나며, Entity Display에서 Rolename/Attribute를 선택하면 아래와 같은 모델이 나타난다.



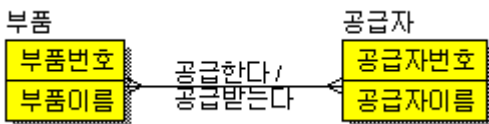
5.5.9 Recursive Relationship(재귀적 관계) 생성하기

1. ERwin Toolbox에서 Non-Identifying Relationship을 선택하여 사원 Entity를 클릭하고 잠시 후에 사원 Entity를 다시 클릭하면 재귀관계가 생성된다.
2. 관계선에서 마우스 오른쪽 버튼 클릭하여 Relationship Properties를 선택한다.
3. Rolename 탭 페이지에서 '관리자 사원번호' 라고 입력한다. OK버튼을 클릭하고 빠져 나오면 아래와 같은 모델이 나타난다.



5.5.10 Relationship의 Issues → Many-to-Many Relationship

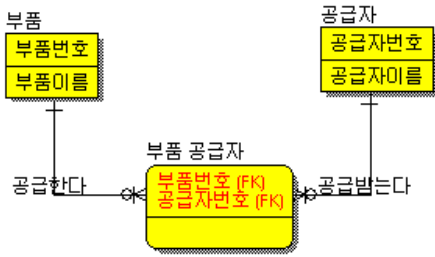
M : M 관계는 Logical모델에서만 허용된다. 예를 들어, 각 부품은 다양한 공급자로부터 제공될 수 있고, 각 공급자는 다양한 부품을 제공한다는 업무규칙을 아래 그림으로 표현한다.



M : M 관계는 Schema생성 전에 하나 이상의 테이블과 하나 이상의 Relationship을 추가함으로써 반드시 해결 되어져야 한다.

예를 들어, 업무규칙은 다음과 같다.

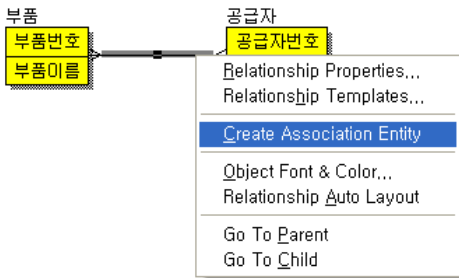
각 부품은 적어도 하나의 공급부품을 부품 테이블 내에 가지고 있고, 각 공급자는 그 공급부품을 공급한다. 각 공급자는 적어도 하나의 공급부품을 부품 테이블 내에 가지고 있고, 각 부품은 공급부품을 제공한다.



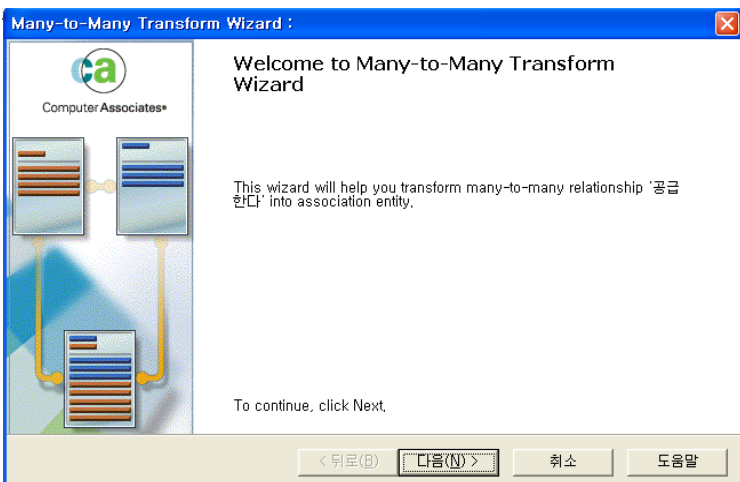
5.5.11 Many-to-Many Relationship 해결하기

Many-to-Many Transform Wizard기능으로 Physical모델에서 필요한 1:다 관계의 Table과 Relationship을 자동으로 생성시켜준다.

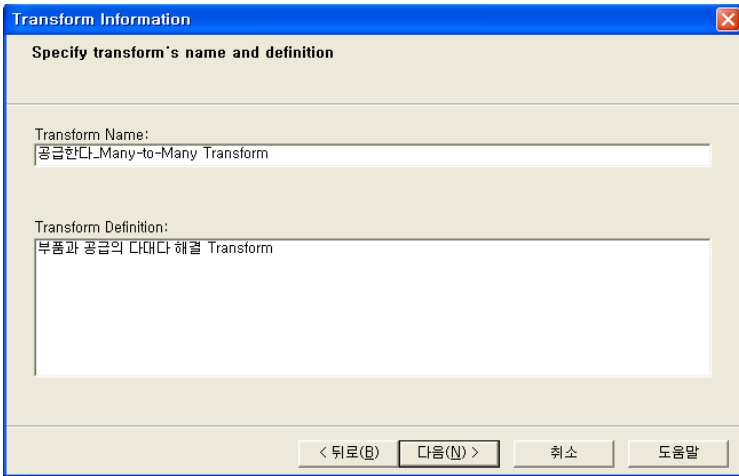
1. 제품Entity와 창고Entity사이의 M:M Relationship에서 마우스 오른쪽버튼 클릭하여 Create Association Entity 메뉴를 선택한다.



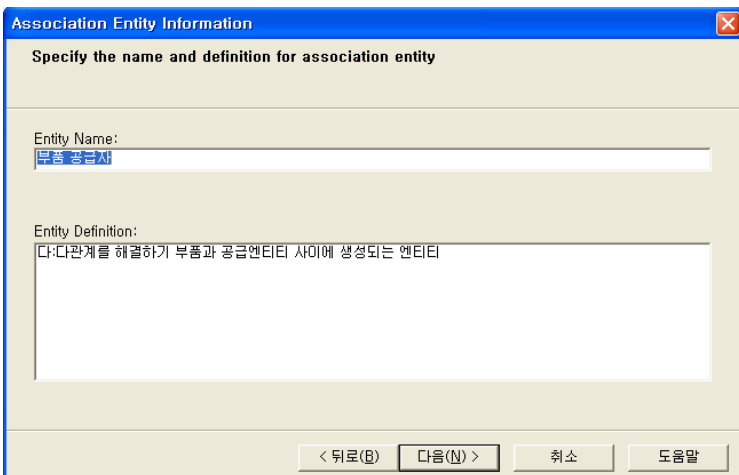
2. Wizard Guide화면이 나타나면 '다음'버튼을 클릭한다



3. Transform Name과 Definition을 입력한다.

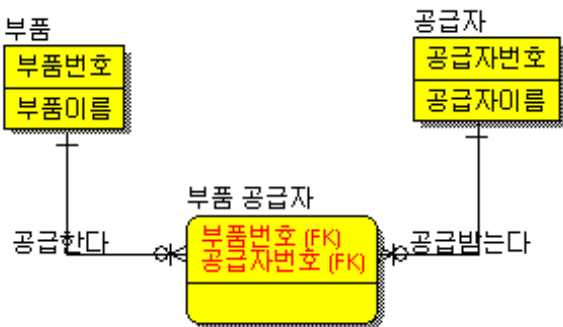


4. 새로 생성될 연관 Entity의 이름을 변경하고 Definition을 입력한다



계속 나타나는 화면에서 '다음'버튼으로 진행하다가 '마침' 버튼을 클릭한다.

5. 아래그림은 연관Table이 생성된 결과 화면이다

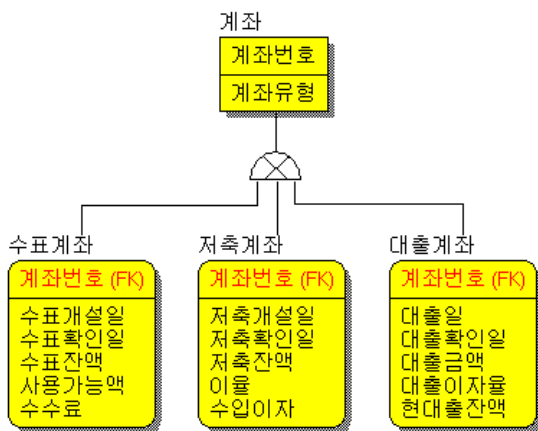


5.5.12 Relationship의 Issues → Generalization Hierarchy

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.

아래의 예제는 계좌에 대해 좀더 자세한 정보를 표현하고 있다. 계좌의 종류로는 수표계좌, 저축계좌, 대출계좌가 있다. 세 종류의 계좌는 별개의 Data를 포함하고 있다.



5.5.13 IE Generalization Hierarchy Symbols



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



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

계정에 관한 예제로 위의 두 가지 Symbol을 적용해 본다면 다음과 같다.

Exclusive(배타) : 하나의 계정은 반드시 저축 계정이거나 당좌 계정이어야 한다. 두 가지 이상 일 수는 없다.

Inclusive(포함) : 하나의 계정은 저축 계정일 수도 있고 당좌 계정일 수도 있다. 아니면 두 가지 모두일 수도 있다.

5.5.14 IDEF1X Generalization Hierarchy Symbols

IDEF1X에서 나타내는 Symbol은 IE Symbol의 의미와 약간 다르다.

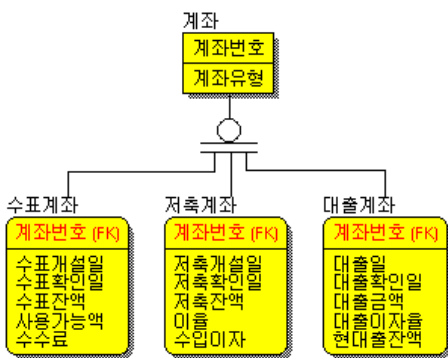


Complete(완전) : 모든 Subtype이 모델에 표현된 경우에는 이것을 이용하며, 각 Supertype 인스턴스는 Subtype중 단 하나의 인스턴스를 가져야만 한다.(must)

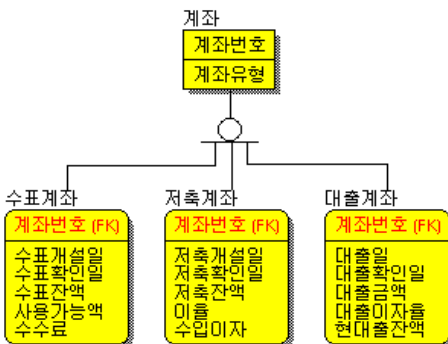


Incomplete(불완전) : 모든 Subtype이 별개의 Attribute를 필요로 하지 않기 때문에 모델에 표현하지 않았을 때 이것을 이용하며, 각 Supertype 인스턴스는 Subtype중 하나의 인스턴스를 가질 수 있다.(may)

다음 예제로 IDEF1X 방법을 설명할 수 있다.



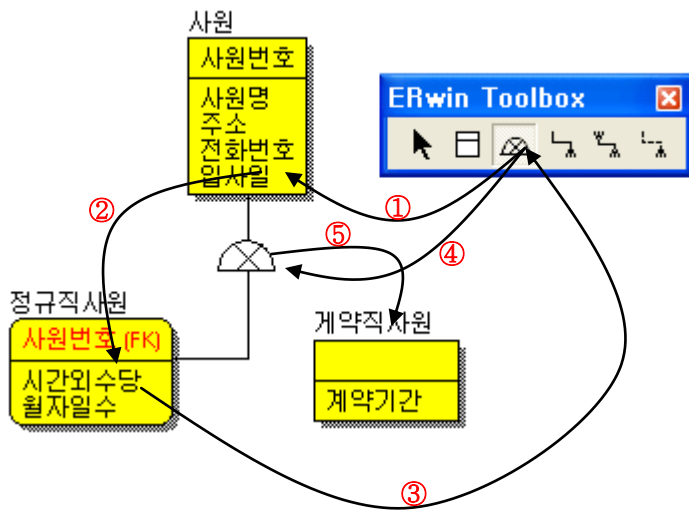
하나의 계좌는 반드시 수표계좌 이거나 저축계좌 혹은 대출계좌 이어야 한다. 다른 것일 수 없다. 계좌의 종류는 단 세가지 종류 만 존재한다



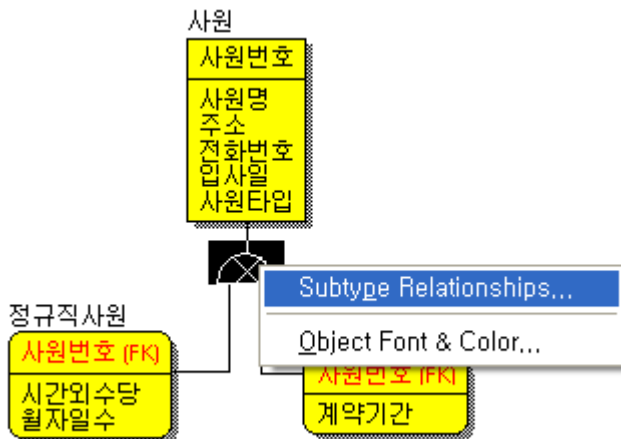
하나의 계좌는 수표계좌, 저축계좌, 혹은 대출계좌 일 수도 있다. 아니면 주식 연동계좌 일 수도 있다. 하지만 주식 연동계좌에 해당되는 Attribute는 이 업무에서 필요로 하지 않기 때문에 모델에 포함시키지 않은 것이다.

5.5.15 Generalization Hierarchy 생성하기

1. 사원 Entity의 Subtype Entity에 해당하는 '정규직 사원', '계약직 사원' Entity를 생성한다.
2. Toolbox에서 Category Relationship 툴을 선택한다.
3. Supertype Entity인 '사원'을 클릭하고 Subtype Entity중 '정규직 사원'을 클릭한다.
4. 다시 Category Relationship 툴을 선택하여 Symbol을 클릭하고, 다른 Subtype Entity인 '계약직 사원'을 클릭한다.

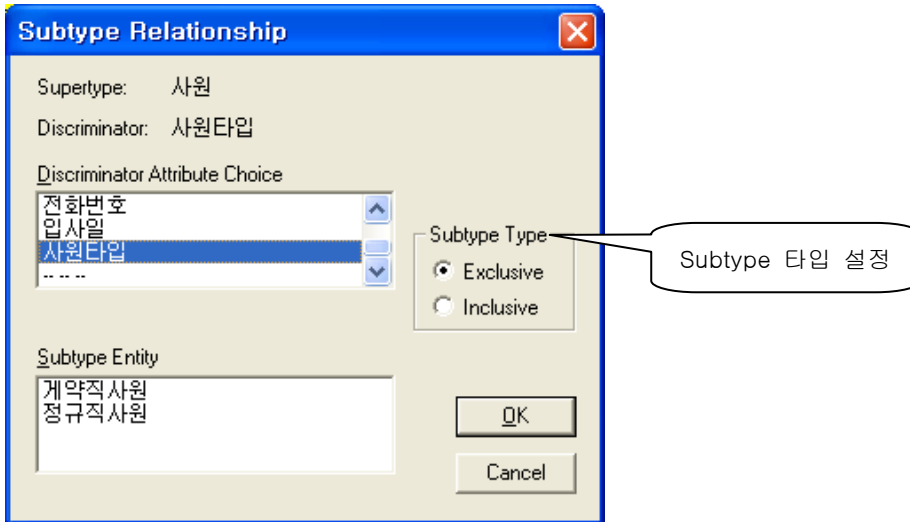


5. 사원 Entity에 '사원타입'이라는 Attribute를 하나 추가한다. 이것은 Subtype Entity를 구분하기 위한 구분자(Discriminator) 이다.
6. Symbol에서 마우스 오른쪽 버튼 클릭하여 Subtype Relationship메뉴를 선택한다



7. '사원타입' Attribute를 Discriminator Attribute Choice로 선택한다.

Subtype의 타입도 이곳에서 선택할 수 있다.



5.6 Referential Integrity(RI, 참조무결성)

참조 무결성 규칙은 Parent와 Child의 인스턴스가 입력, 수정, 삭제될 때 어떤 일이 일어나는지 규정하는 규칙을 가리킨다. 이것을 통해서 데이터베이스는 데이터 무결성을 유지한다.

결국 RI는 Parent가 없는 데이터의 입력을 막아주고, 업무규칙을 강제적으로 보증하는 수단이 된다.

관계형 데이터베이스에서는 Foreign Key로 관계를 유지한다. 이것은 부모 테이블의 Primary값과 동일한 값으로, 1 : M 관계를 통해 자식 테이블로 이주 된다.

아래 예제를 읽어보고 RI의 필요성에 대해 생각해 보자.

처방전을 기록하는 의사는 DEA(Drug Enforcement Agency) ID를 가지고 있다.

각 의사들에 대한 정보를 담고 있는 문서가 있으며 그 DEA id는 각 품의 오른쪽 상단에 나타나있다.

숫자로 표시되어 있는 그것은 의사의 Primary Key이다.

한 의사가 한 환자를 위해 작성하는 모든 처방전은 담당의사가 기록한다.

그러므로 DEA ID는 처방전 품에 기록되는데 그것이 바로 Foreign Key이다.

새 처방전을 기록할 때, 처방전 내의 Foreign Key에 담당의사 DEA ID가 사용자의 실수로 잘못 기록되어 진다면 유지되어지는 문서가 어떻게 될까?

ERwin과 RI Option

ERwin은 Relationship의 각 Action들에 대해 기본적인 RI 규칙을 가지고 있다.

이것은 Relationship 타입, Action, Entity의 역할(Parent or Child)에 기초한다.

기본적으로는, 주어진 Relationship에 대해 주어진 Action이 전체 모델에 반영된다.

예를 들어 '고객'과 '고객불만사항'이라는 Entity, 그리고 두 Entity사이의 관계가 있는 모델을 가지고 RI를 적용해 보면 '고객불만사항'의 모든 인스턴스는 '고객'을 참조하게 되어 있다.

고객의 특정 데이터를 수정하거나 삭제할 때, '고객불만사항'에서 Parent가 없는 데이터가 발생하는 것에 대해 주의 깊게 살펴보아야 한다.

Parent(고객)의 정보를 변경하기 원할 때, Child도 함께 변경되는 규칙을 'Cascade'라고 부른다.

고객의 인스턴스를 삭제할 때 Parent에 관련된 모든 '고객불만사항'의 인스턴스는 함께 삭제 된다.

5.6.1 기본으로 지원되는 RI 타입

ACTION	RELATIONSHIP TYPE			
	Identifying	Non-Identifying (Null Allowed)	Non-Identifying (No nulls)	Subtypes
Child Delete	None	None	None	None
Child Insert	Restrict	Set Null	Restrict	Restrict
Child Update	Restrict	Set Null	Restrict	Restrict
Parent Delete	Restrict	Set Null	Restrict	Cascade
Parent Insert	None	None	None	None
Parent Update	Restrict	Set Null	Restrict	Cascade

5.6.2 참조무결성의 강제

DBMS에서는 몇가지 RI를 실행할 수 있는 방법이 있다.

이들 방법을 몇가지 조합해서 사용하는 것이 최적이라고 볼 수 있다.

- ▣ 서버상에서 선언적 RI를 사용하는 방법
- ▣ 서버상에서 Trigger를 사용하는 방법
- ▣ Stored Procedure를 통해 갱신을 수행하는 방법
- ▣ Application 상에서 코딩으로 RI를 실행시키는 방법

Logical모델 개발 시, ERwin은 특정한 규칙에 근거하여 RI를 다루고 있다.

사용자가 ERwin에서 사용되는 기본적인 RI규칙을 변경할 수 있도록 지원한다.

5.6.3 Declarative RI(선언적 RI)

이것은 Non-Procedural Referential Integrity(비절차적 참조 무결성 규칙)라고도 부른다.

예: 각 고객은 0,1,그 이상의 불만을 기록한다는 관계에서 가질 수 있는 RI규칙이 Parent Delete Restrict라고 가정할 때 ERwin은 해당 DBMS에 따라 다르게 RI 코드를 생성할 수 있다.

- ▣ SQL Server

```
ALTER TABLE CUSTOMER
        ADD PRIMARY KEY (customer_number)
go
...
ALTER TABLE PROBLEM
        ADD FOREIGN KEY (customer_number)
                REFERENCES CUSTOMER
go
```

■ DB2

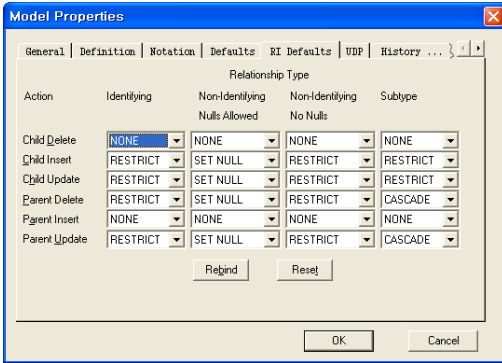
```
ALTER TABLE CUSTOMER
        ADD PRIMARY KEY (customer_number);
...
ALTER TABLE PROBLEM
        ADD FOREIGN KEY (customer_number)
                REFERENCES CUSTOMER
                ON DELETE RESTRICT;
```

5.6.4 Declarative RI의 이점

Declarative RI는 최적화된 코드이다. 왜냐하면 DBMS에서 직접 구현되는 코드이기 때문이다. 그리고, 단지 적당한 PK와 FK선언으로 코드가 생성되기 때문에 간단하다.

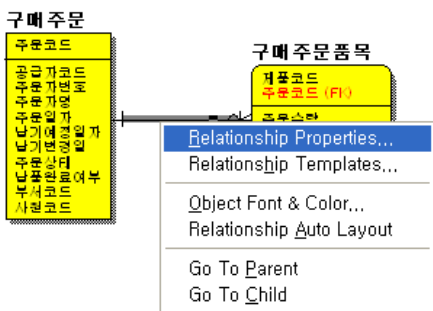
5.6.5 Global RI 옵션 변경하기

1. Model메뉴에서 Model Properties메뉴를 선택하여 RI Default 탭 페이지를 클릭한다.
2. Non-Identifying Nulls Allowed Action 컬럼의 모든 SET NULL을 RESTRICT로 바꾼다.
3. Rebind버튼을 클릭하여 Relationship들의 변경된 RI규칙을 본다. 주의할 것은, Model의 모든 Relationship에 Override되는 것이므로, Relationship마다 지정한 규칙은 모두 Global RI옵션으로 대체된다.



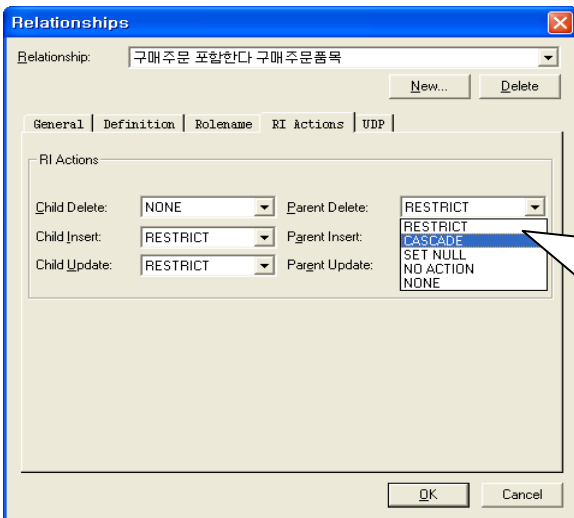
5.6.6 Relationship마다 RI 옵션 변경하기

1) '구매주문'과 '주문품목' Entity 간의 Relationship에서 마우스 오른쪽 버튼을 클릭하여 Relationship Properties 메뉴를 선택한다.



2) RI Action 탭 페이지로 이동한다.

3) Parent Delete 리스트박스에서 Restrict대신 Cascade를 선택한다.




구매주문품목 삭제 시 주문정보 또한 같이 삭제 된다.
이렇듯 사용자가 부모/자식 엔티티 타입의 입력/수정/삭제에 대한 참조 무결성 규칙을 정의 할 수 있다.

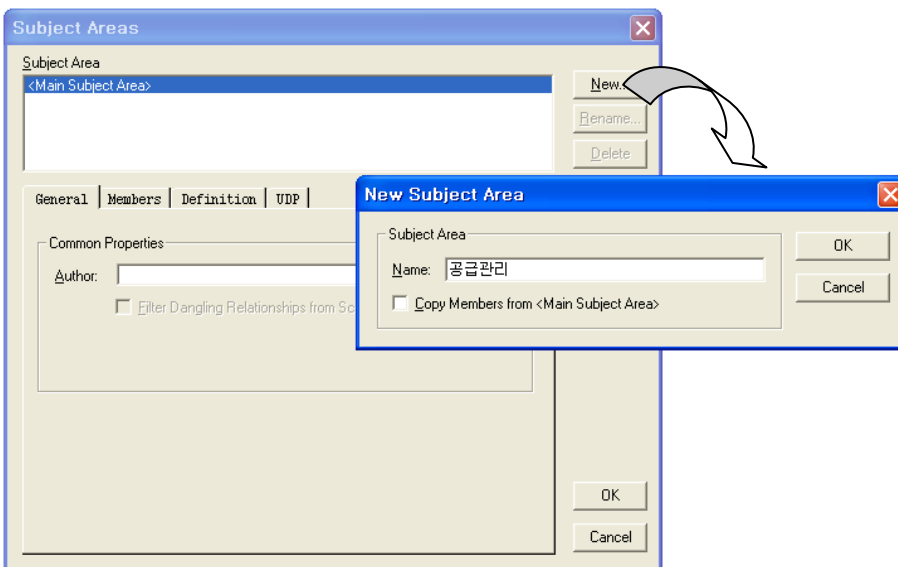
5.7 Subject Area (주제영역)

모델러가 전체 모델을 통합 관리하면서 Entity들의 부분집합을 만들어 작업할 수 있도록 지원하는 기능을 말하며 특정 비즈니스 단위나 프로세스에 의해 사용되는 Entity와 Relationship을 보여주기 위해 Subject Areas를 만들 수 있다. 디폴트로서 데이터 모델은 <Main Subject Areas>가 된다.


예를 들면 전체 회사를 모델링 하고자 할 때 하부 단위로 회계, 생산, 마케팅, 인사 등의 주제로 나누어 모델링 할 수 있는데 이를 Subject Area로 구분하여 작업할 수 있으며, 이 Subject Area별로 모델링 한 결과는 Main Subject Area에 반영되게 된다.

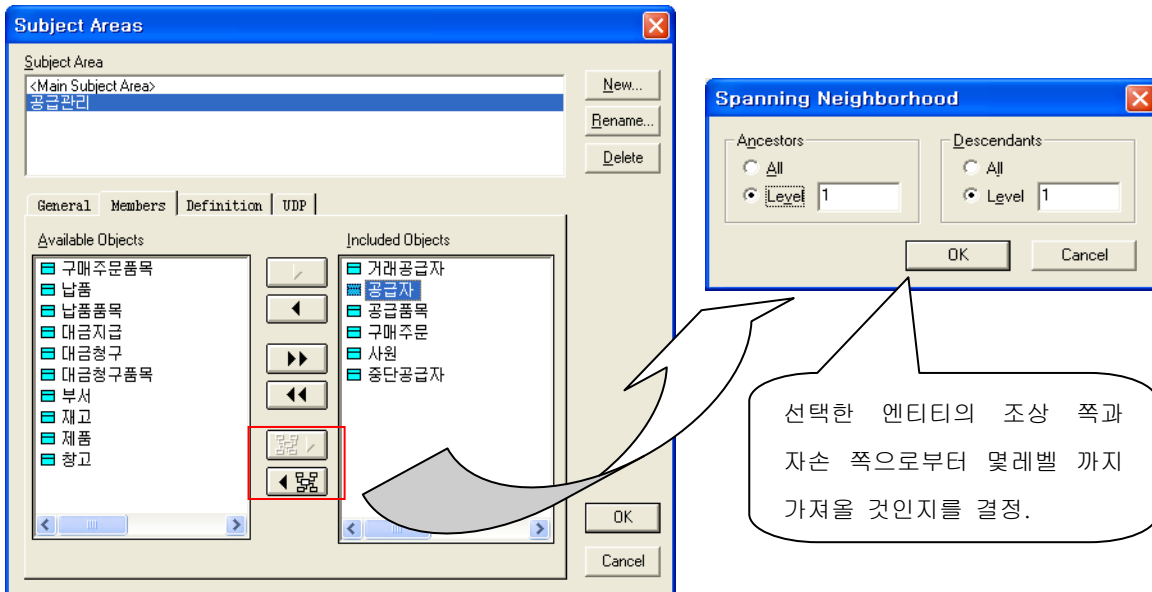
5.7.1 Subject Area 생성하기

- 1) Model메뉴에서 Subject Areas메뉴를 선택 하거나  아이콘을 선택한다.
- 2) New버튼을 클릭하여 새로운 주제영역명을 입력한다.



Members 탭 페이지에서 각 주제영역에 해당하는 Entity를 Included Objects쪽으로 이동시킨다. Entity를 옮기는 방법은 여러 가지가 있을 수 있다.

- ▣ Entity를 Double-Click하거나, Entity를 선택한 후 화살표 버튼으로 옮긴다
- ▣ 특정 Entity를 클릭  하고 버튼을 클릭하여 해당 Entity와 관계된 Entity들을 한꺼번에 옮긴다.

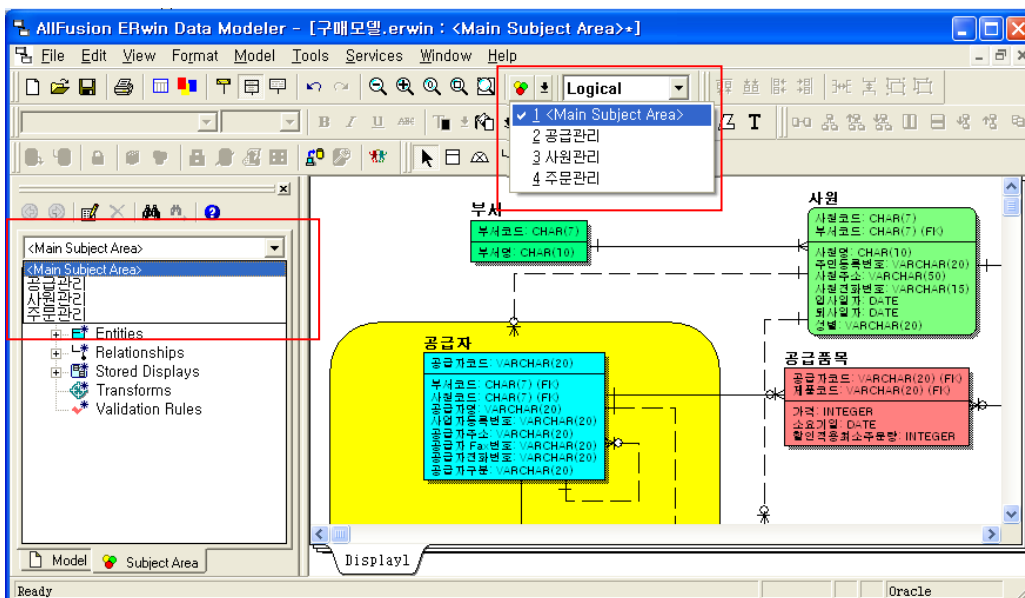


Subject Area를 삭제하고자 할 때는 메뉴 Edit에서 Subject Area를 선택한 다음 삭제하고자 하는 Subject Area를 선택하고 Delete버튼을 클릭하면 Subject Area가 삭제된다.

Subject Area를 저장하고자 할 때는 개별적으로 Subject Area가 저장되지 않고 파일로 저장되기 때문에 메뉴 File에서 Save나 Save As를 선택하여 전체 Diagram을 저장해야 한다.

5.7.2 다른 Subject Area로 이동하기

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.

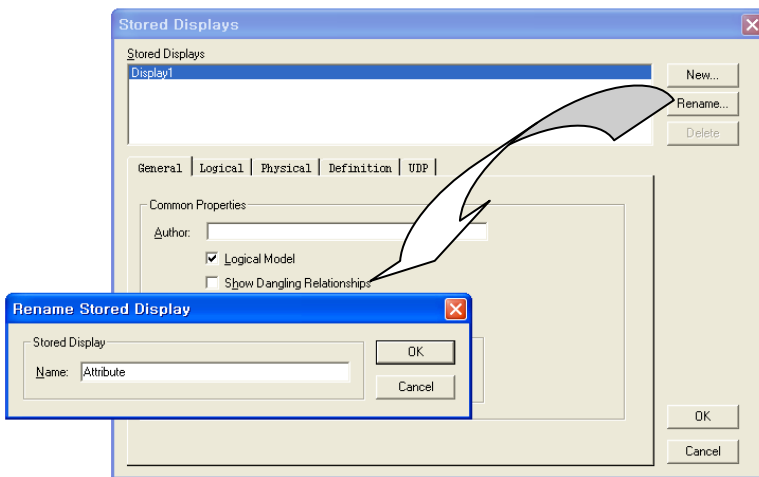


5.8 Stored Displays

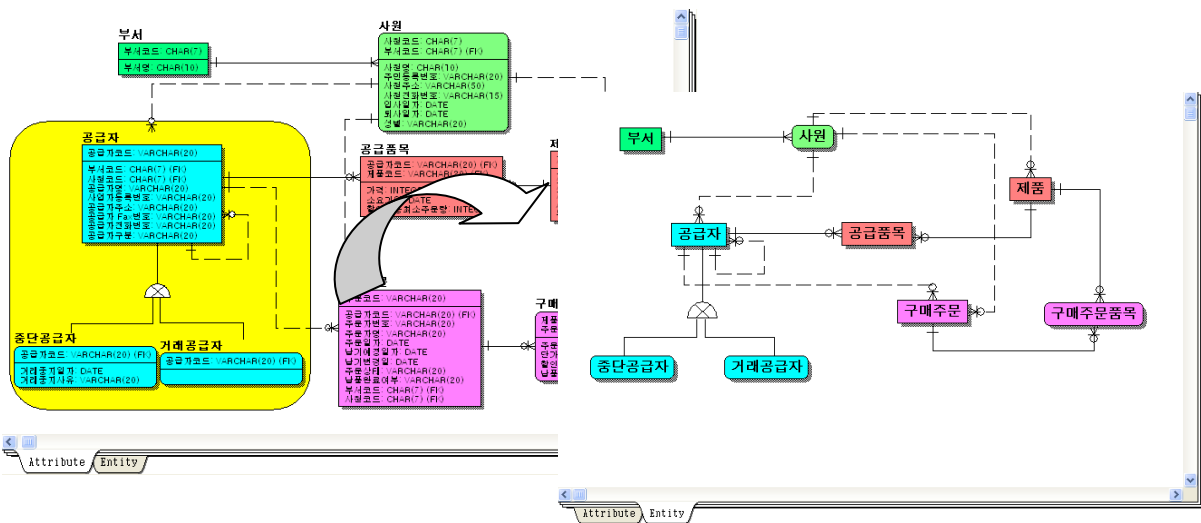
모델을 다양한 배열로 저장할 수 있도록 지원하는 기능이다. 이것은 서로 다른 배경지식이나 요구사항을 가지고 있는 사용자들에게 다양한 모델의 형태를 보여주는데 효과가 있다. Stored Display는 각 Subject Area마다 분리하여 작성할 수 있으며, 각 Subject Area는 자신만의 Stored Display를 가지고 있다. 이것은 화면 아래부분의 탭 페이지로 보여진다.

Stored Display 생성하기

- 1) Format메뉴의 Stored Display Settings...을 선택하여 Stored Displays Editor를 연다.
- 2) Rename버튼을 클릭하여 'Attribute'라고 입력한다.



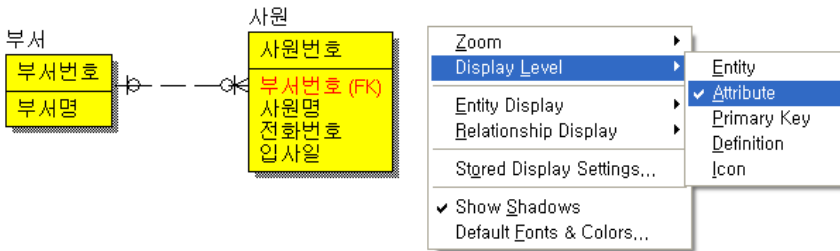
- 3) 다시 New버튼을 클릭하여 'Entity'라고 입력한다. OK버튼을 클릭하여 창을 나간다.
- 4) Entity Display 탭 페이지에서, 바탕화면에서 마우스 오른쪽버튼 클릭하여 Display Level의 Entity를 선택한다. Entity Display화면에서 Entity와 Relationship을 재배치 하여 모델을 정리한다.



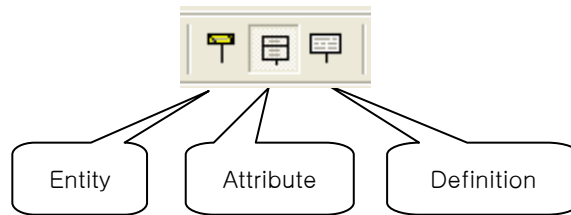
5.9.2 Display Option 설정하기

5.9.2.1 Display Level 설정

1. 바탕화면에서 마우스 오른쪽 버튼 클릭하여 Display Level메뉴의 5가지 Level 중 원하는 메뉴를 선택한다.



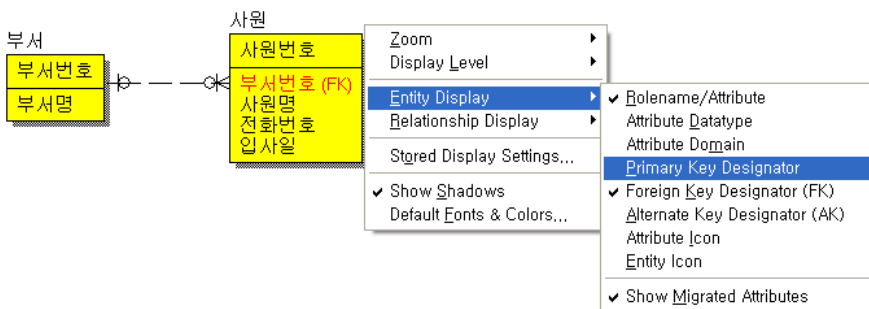
2. 또 다른 방법은 Standard Toolbar를 이용할 수 있다.



5.9.2.2 Hide/Show Option 설정

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



5.9.2.3 Zoom 설정

Standard Toolbar를 이용하여 모델의 크기를 설정해 본다



5.9.2.4 Graphics 설정하기

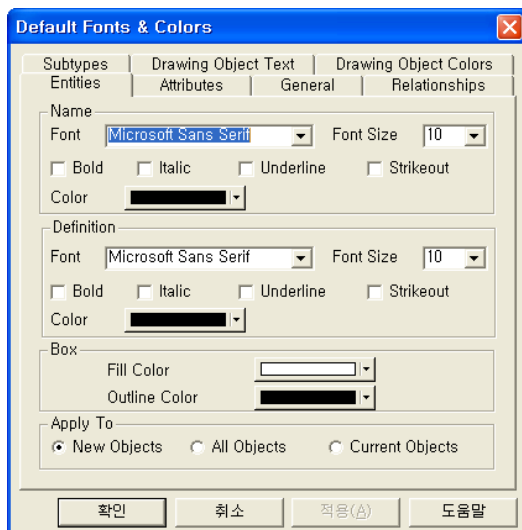
1) Entity 그림자 설정

바탕화면에서 마우스 오른쪽버튼 클릭하여 Show Shadows를 선택한다



2) Font와 Color 설정

바탕화면에서 마우스 오른쪽버튼 클릭하여 Default Fonts & Colors...를 선택하여 원하는 옵션을 설정한다.



Default Fonts & Colors... 옵션을 이용하여 Foreign Key에 대하여 색을 다르게 줄 수 있다.

Attributes 탭 선택

원하는 색을 선택

체크를 풀어 준다

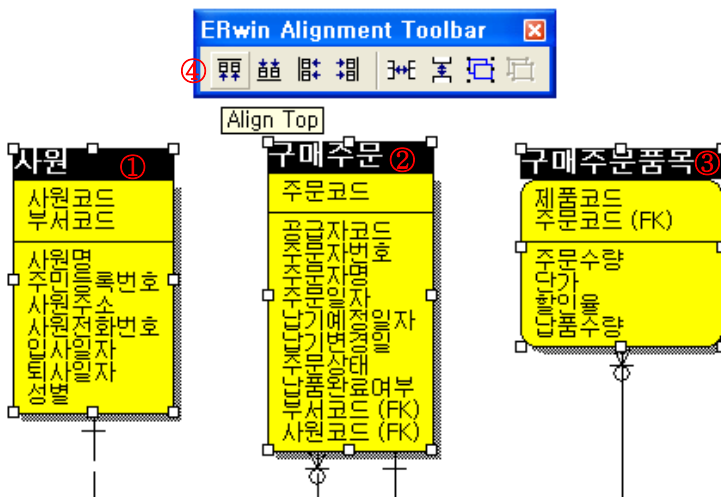
현재 오브젝트에 적용

새로 생성되는 오브젝트들에 적용

모든 오브젝트에 적용

3) Relationship과 Entity의 정렬

Entity와 Relationship은 클릭과 Dragging의 수작업으로 정렬 할 수 있고, ERwin의 Auto-Layout Tool을 이용하여 자동으로 정렬할 수 있다.



- 정렬할 Entity들을 Ctrl키를 이용하여 다중 선택하거나 마우스를 Dragging하는 Lasso기법으로 Entity들을 선택한다.
- ERwin Data Modeler의 Auto-Layout Tool중 하나를 클릭한다

5.9.2.5 ERwin Data Modeler 객체에 대한 편집

ERwin Data Modeler 모델 객체 복사하기.

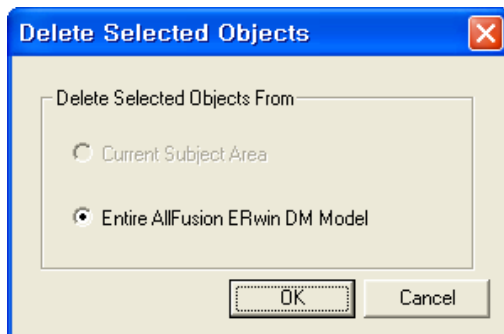
모든 모델에 대한 객체는 복사할 수 있는데, 복사 기능을 이용하여 모델의 일부분을 복사하여 다른 모델에 붙일 수 있고, 다른 Application에 붙일 수 있다.

방법으로는 복사하고자 하는 객체를 먼저 선택하고, 메뉴 Edit에서 Copy Entity를 이용하거나, Ctrl+C 단축키를 이용하면 된다.

복사된 오브젝트는 다른 ERwin Data Modeler Diagram에서 붙이기 기능(메뉴 Edit에서 Paste)을 이용하여 붙이기가 가능하고 워드프로세서에서도 붙이기를 이용하여 편집이 가능하다.

ERwin Data Modeler 모델 객체를 삭제하기.

삭제하기를 원하는 객체를 먼저 선택하고, 메뉴 Edit에서 Cut Entity(물리적 단계에서는 Table)을 선택하거나 Delete키를 누르면 아래와 같은 화면이 나타난다.



- 1) Current Subject Area : 현재 Subject Area에서만 Object가 삭제되며, Main Subject Area에서는 삭제되지 않는다.
- 2) Entire AllFusion ERwin DM Model : 전체 모델에서 해당 Object가 삭제된다.

선택된 객체를 삭제하고자 할 때는 OK버튼을 누르면 된다.

Physical Data Modeling



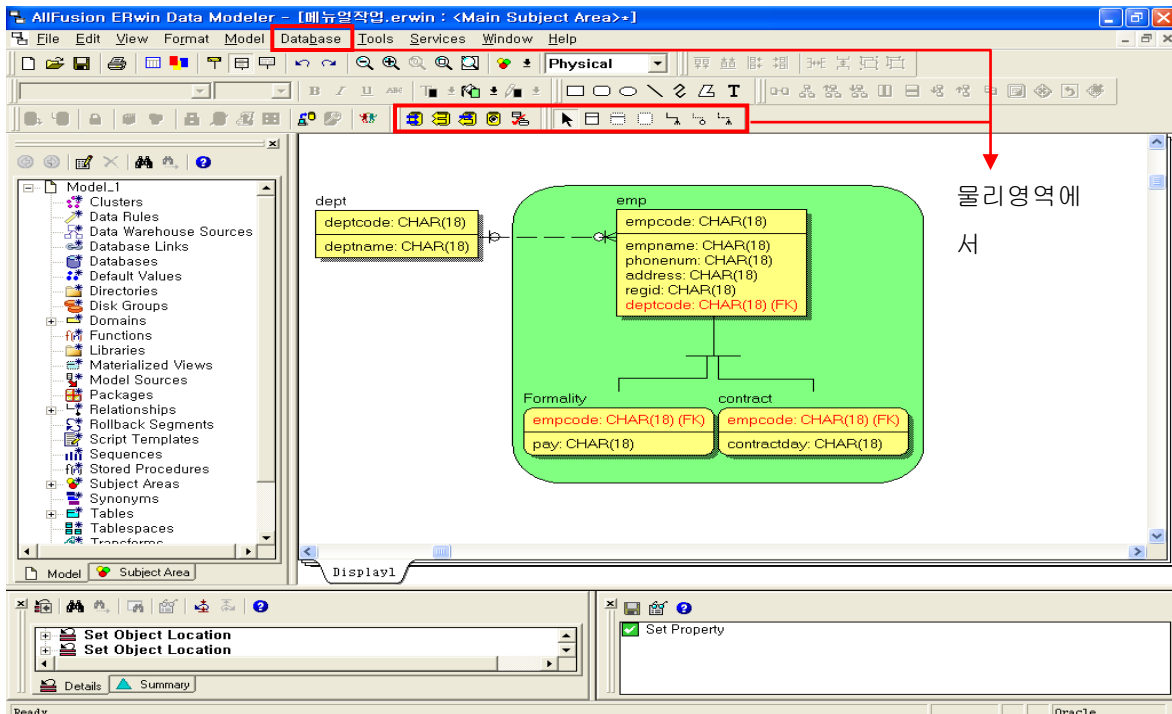
6. Physical Modeling

6.1 Physical / Logical Model의 표현

ERwin의 전통적인 모델 표현 방식인 Logical/Physical 복합 모델은 하나의 데이터 모델에서 Logical과 Physical 모델을 모두 표현할 수 있다. 그리고 “Logical Only”의 Entity, Attribute, Relationship, Key Group를 허용한다. 마찬가지로 “Physical Only”의 Table, Column, Relationship, Indexes도 허용한다. Target Server의 물리적 특성을 지원하며 물리적인 데이터 모델과 Target Server의 시스템 카탈로그를 동기화 한다. Entity나 Table을 추가하는 작업과 같이 한 모델의 구조를 변경하였을 때 “Logical Only”나 “Physical Only”로 정의하지 않는 한 동시에 Logical과 Physical Model 모두에서 변경이 일어난다. 선택적으로 Logical 모델의 M : M 관계와 일반화 관계를 Physical 모델에서 자동으로 변환할 수 있다. 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.

6.2 Physical Model 변환

완성한 논리 모델을 기반으로 물리 모델을 진행한다. 물리 모델로의 변환은 메뉴에서 Physical 모드를 선택함으로써 쉽게 변환이 가능하다. Physical 모드를 선택하면 물리 모델링에서 필요한 메뉴가 활성화되고 툴박스에는 뷰테이블 생성에 필요한 아이콘이 추가된다.



Tip

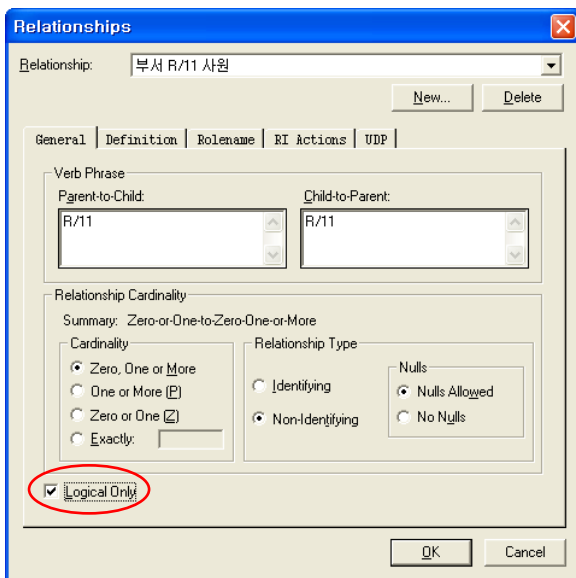
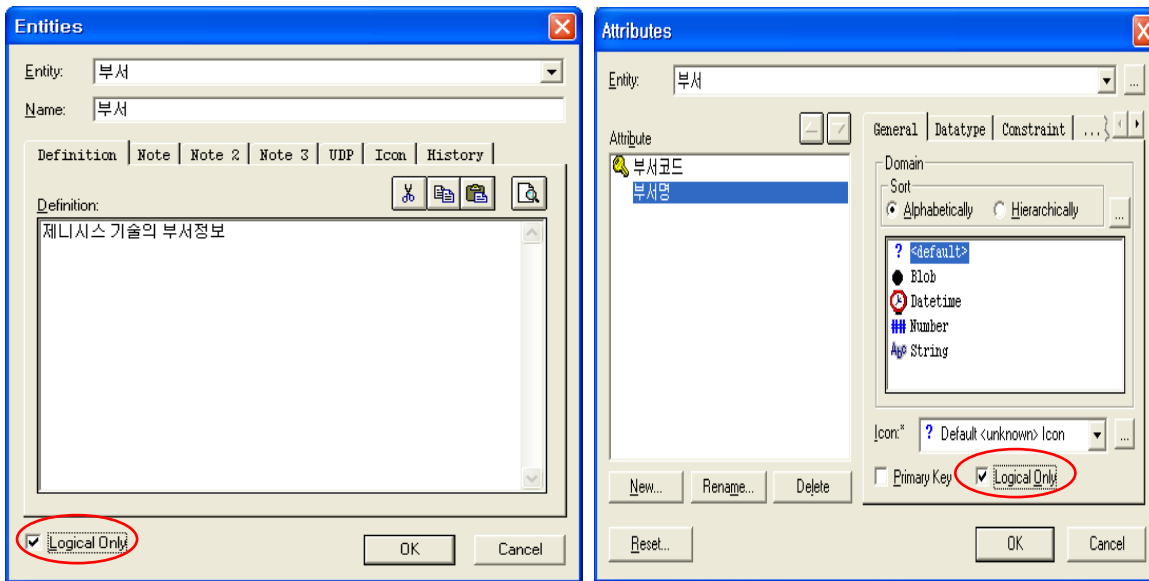
Layer 전환 단축키 : Ctrl + ↑, ↓

6.3 Logical / Physical Object의 차별화

Logical과 Physical 모델의 객체를 차별화 할 수 있다. Logical 모델에서 Entity, Attribute 그리고 Relationship 각각의 에디터에서 Logical Only 체크박스가 있으며, Physical 모델에서는 Table, Column, Relationship 에디터에 Physical Only 체크박스를 포함 한다.

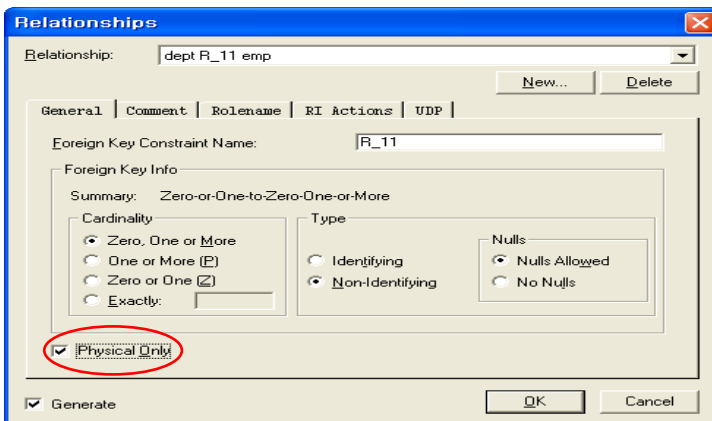
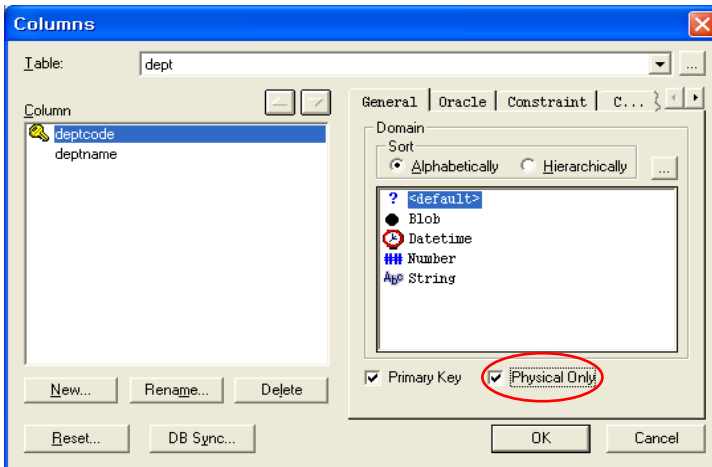
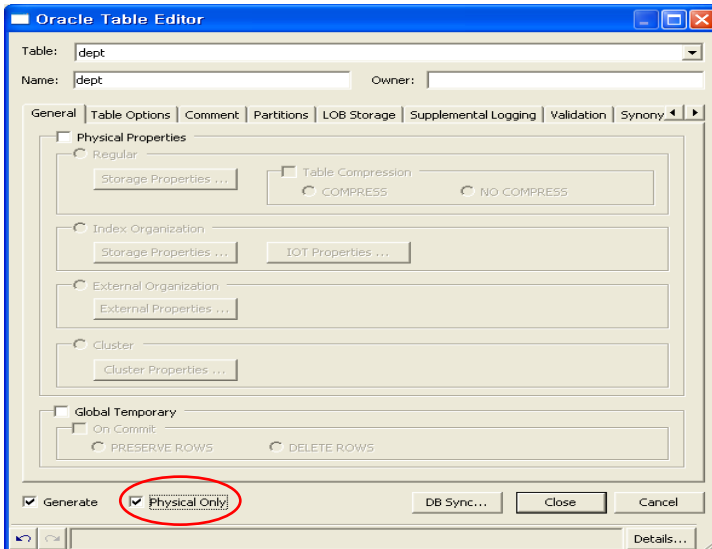
6.3.1 Logical Only

Logical Only 옵션을 체크한 Object들은 물리 모델에서는 보이지 않는다.



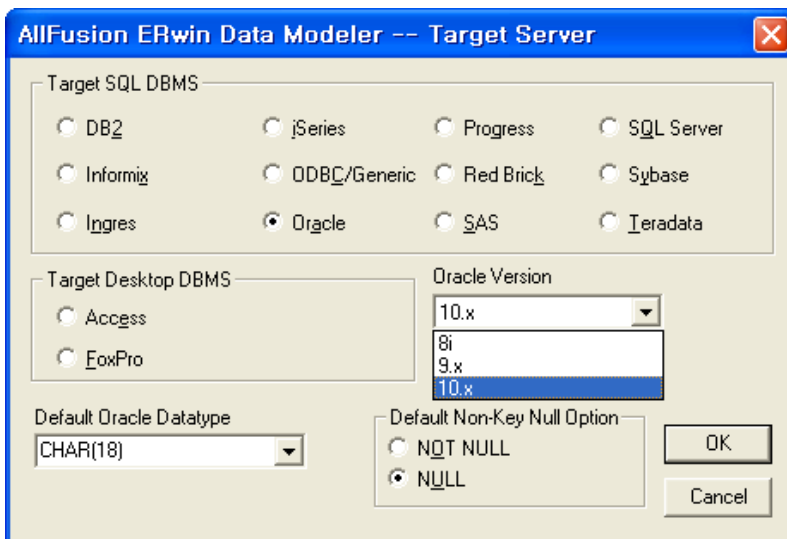
6.3.2 Physical Only

Physical Only 옵션을 체크한 Object들은 논리 모델에서는 보이지 않는다.



6.4 Target Server 지정

- Physical 모델은 DBMS에 의존적이다.
- 메뉴에서 Database > Choose Database를 선택하면 Target Server를 지정할 수 있다.
- Target DBMS를 선택하면 선택한 DBMS에 맞는 Default Datatype과 Null 옵션을 지정한다.
- Default 값 지정은 메뉴의 Model > Model Properties에서도 할 수 있다.



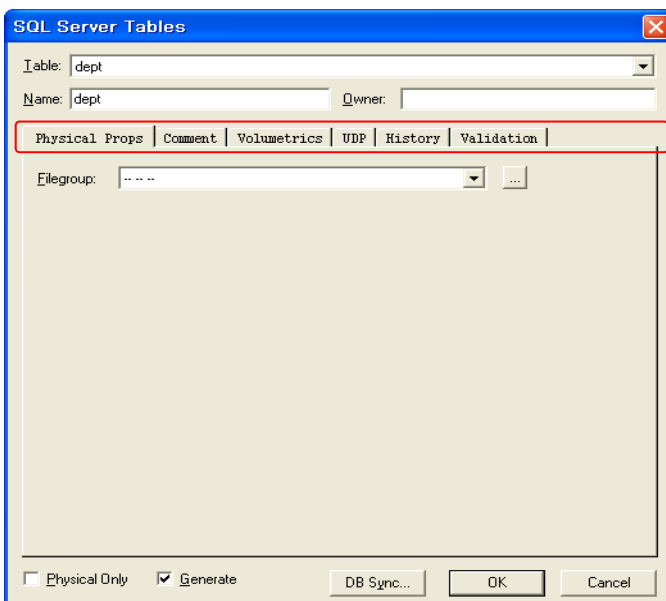
Physical 영역으로 이동하면 디폴트로 엔티티명(속성명)이 테이블명(컬럼명)으로 그대로 전환된다. 만약 엔티티명(속성명)과 테이블명(테이블명)을 다르게 하기 위해서는 Physical 영역에서 테이블명(컬럼명)을 재명명 하면 된다. 이렇게 테이블명(컬럼명)을 편집한 순간부터 엔티티명(속성명)과 테이블명(컬럼명)이 따로 관리된다.

6.5 Table Property

Physical 모델에서는 Logical 모델의 Entity가 테이블로 변환된다. 즉 기본적으로 Logical 모델의 정보를 물리 모델로 가져온다.

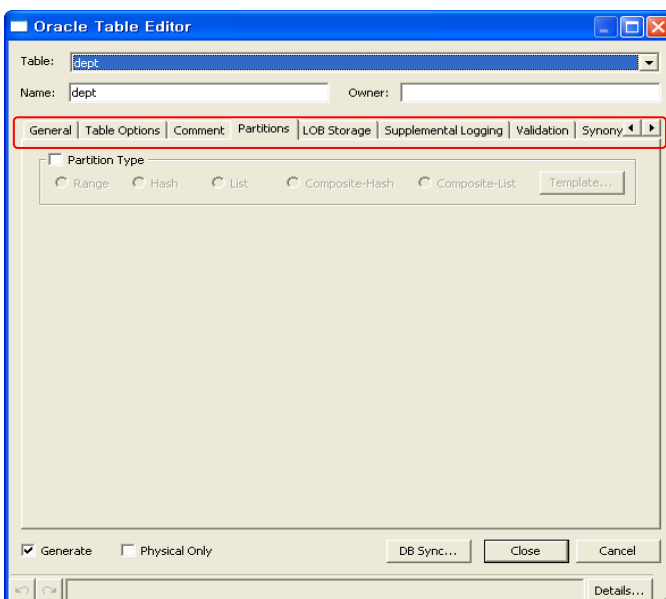
6.5.1 DBMS별 Table Property 항목 (MS SQL / Oracle)

1) MS SQL



Target DBMS 별로
다양한 속성을 지원한다.

2) Oracle



Tip

Table Editor 보기 : 테이블 선택 → 마우스 오른쪽 클릭 → Table Properties → General 선택

3) Table Properties 항목 설명

MS SQL	
<input type="checkbox"/> Owner (공통)	해당 테이블의 오너를 부여한다. SQL 문장 생성 시 옵션 사항으로 오너를 Table Name 앞에 정의 할 수 있다.
<input type="checkbox"/> Physical Props	Target DBMS의 물리적 속성을 지정한다. File Group 생성 및 Mapping 하고 DB Sync를 통해 DBMS나 Script와 매핑 할 수 있다.
<input type="checkbox"/> Comment (공통)	Entity Definition의 내용이 Comment로 들어온다.
<input type="checkbox"/> Generate (공통)	체크를 빼면 스키마 생성시 해당 테이블이 제외된다.
<input type="checkbox"/> Volumetrics (공통)	Table의 초기값(Initial Rows), 최대값(Max Rows), 그리고 월마다 증가 (Grow By)하는 Row의 수치를 입력해 대략의 디스크 용량을 가능해 볼 수 있다.
<input type="checkbox"/> UDP (공통)	사용자가 Table의 속성 항목을 정의해서 사용 할 수 있다.
<input type="checkbox"/> History (공통)	테이블의 History 정보를 확인 할 수 있다.
<input type="checkbox"/> Validation (공통)	테이블의 Validation Rule을 지정 할 수 있다.
Oracle	
<input type="checkbox"/> General	Target DBMS의 물리적 속성을 지정할 수 있으며, Tablespace를 생성 하거나 매핑 할 수 있고 다양한 Oracle 속성을 지정한다.
<input type="checkbox"/> Table Option	Volumetrics 기능이 포함되어 있으며, 테이블의 Parallel Type등 다양한 속성을 지정한다.
<input type="checkbox"/> Partition	Range, Hash, List, Composite(Hash, List) Partition을 지원한다.
<input type="checkbox"/> LOB storage	LOB Datatype인 Column을 등록하고 Parameter 값을 지정한다.
<input type="checkbox"/> Supplemental Logging	LogMiner 기능을 사용하기 위해 Supplemental Logging 기능을 활성화 한다. (Supplemental Log Group 등록)
<input type="checkbox"/> Synonym	Table의 Synonym을 생성한다.
<input type="checkbox"/> Materialized View log	Materialized View의 변경사항을 자동으로 갱신하기 위해서는 FAST REFRESH 옵션이 필요하다. 이 옵션을 사용하기 위한 Materialized View Log를 생성한다.

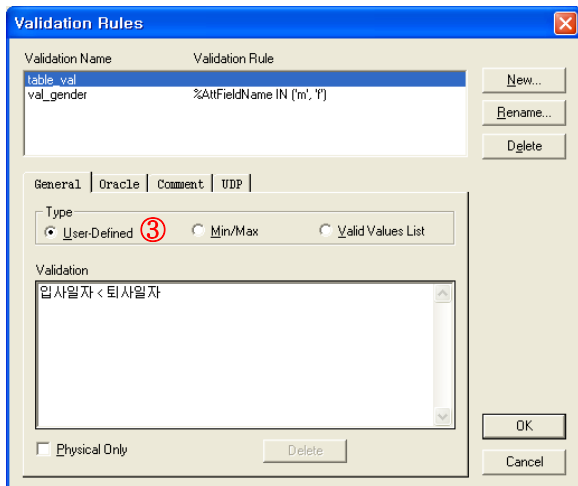
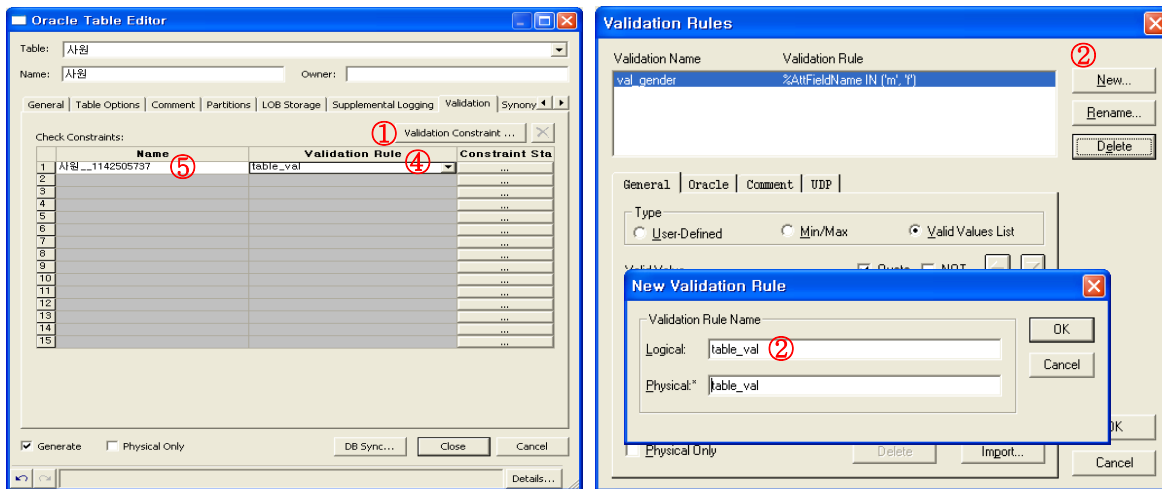
Tip

LogMiner : Log 파일을 추출하여, 그 안에 저장되어 있는 내용을 Database에 가해진 논리적인 작업을 표현하는 SQL 문장으로 변환한다. Transaction, 사용자, 테이블, 시간별 DB의 변경작업을 추적할 수 있다.

6.5.2 Table Validation

Table Validation은 각각의 Column에 대한 Rule과는 달리 Table 레벨에서 여러 Column 사이의 Validation Rule를 지정할 수 있다. 예를 들면 사원 테이블의 입사일자와 퇴사일자 사이에는 입사일자는 퇴사일자보다 클 수 없다. 라고 제한 한다면 다음과 같은 Rule을 만들 수 있다.

입사일자 < 퇴사일자



----- 작업절차 -----

- ① Validation Constraint 클릭
- ② New를 클릭하고 이름을 입력
- ③ Type을 선택하고 Rule을 입력
- ④ 생성한 Rule을 선택
- ⑤ Name을 입력

Tip

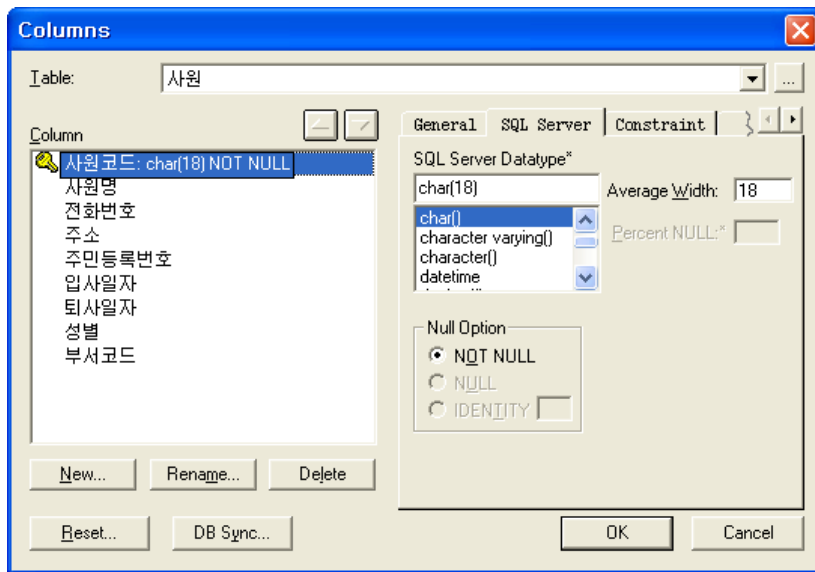
- ① User-Defined : 사용자가 직접 코드를 타이핑 한다.
- ② Min / Max : 값의 범위(최소값 / 최대값)를 정의한다.
- ③ Valid Values List : 허용되는 값의 목록을 정의한다.

6.6 Column Property

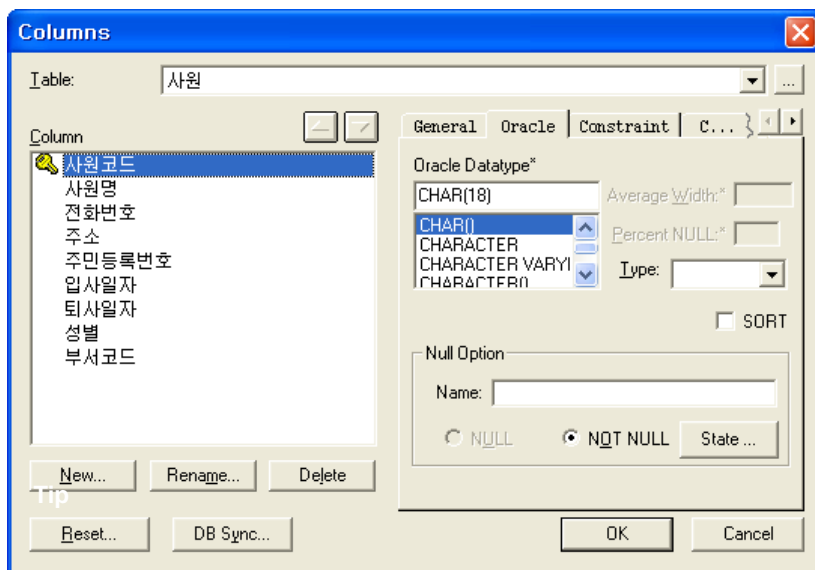
Target DBMS 별로 Column의 물리적인 속성을 지원한다. Logical 모델의 Attribute 정보가 Comment의 기본 정보가 된다. Attribute 이름이 Column 이름으로 변환 되고 스크린 상에서 바로 Column의 이름을 Rename 할 수 있으며 Column Editor에서 세부 속성을 정의 한다.

6.6.1 DBMS별 Column Property

1) MS SQL



2) Oracle



Tip

Column Editor 보기 : 테이블 선택 → 마우스 오른쪽 클릭 → Columns 선택

3) Column Properties 항목 설명

MS SQL / Oracle	
■ New (공통)	새로운 Column을 생성한다.
■ Rename (공통)	기존의 Column Name을 변경한다.
■ Delete (공통)	Column을 삭제한다.
■ Reset (공통)	선택한 Object의 Property 항목을 초기값으로 되돌린다.
■ DB Sync (공통)	DBMS에 연결하여 Schema 정보를 비교하여 필요한 항목을 Import 받는다.
■ General (공통)	Domain을 생성하고 Column에 Domain을 Mapping 한다.
■ SQL Server / Oracle	DBMS별 Data Type 과 Length, Null Option을 정의 한다.
■ Constraint (공통)	Column에 Validation 이나 Default Value Rule을 정의 한다.
■ Comment (공통)	Logical 모델의 Definition 값이 자동으로 입력된다. (수정가능)
■ UDP (공통)	UDP는 사용자가 정의하는 특성으로 Column에 대한 이슈 항목을 생성하고 데이터를 입력 한다.
■ Index (공통)	Index 정보를 확인 하고, 새로운 Index(AK, IE)를 만들 수 있다. 또한 Index Member를 추가, 삭제 할 수 있다.
■ History (공통)	Column의 History 정보를 확인 한다.

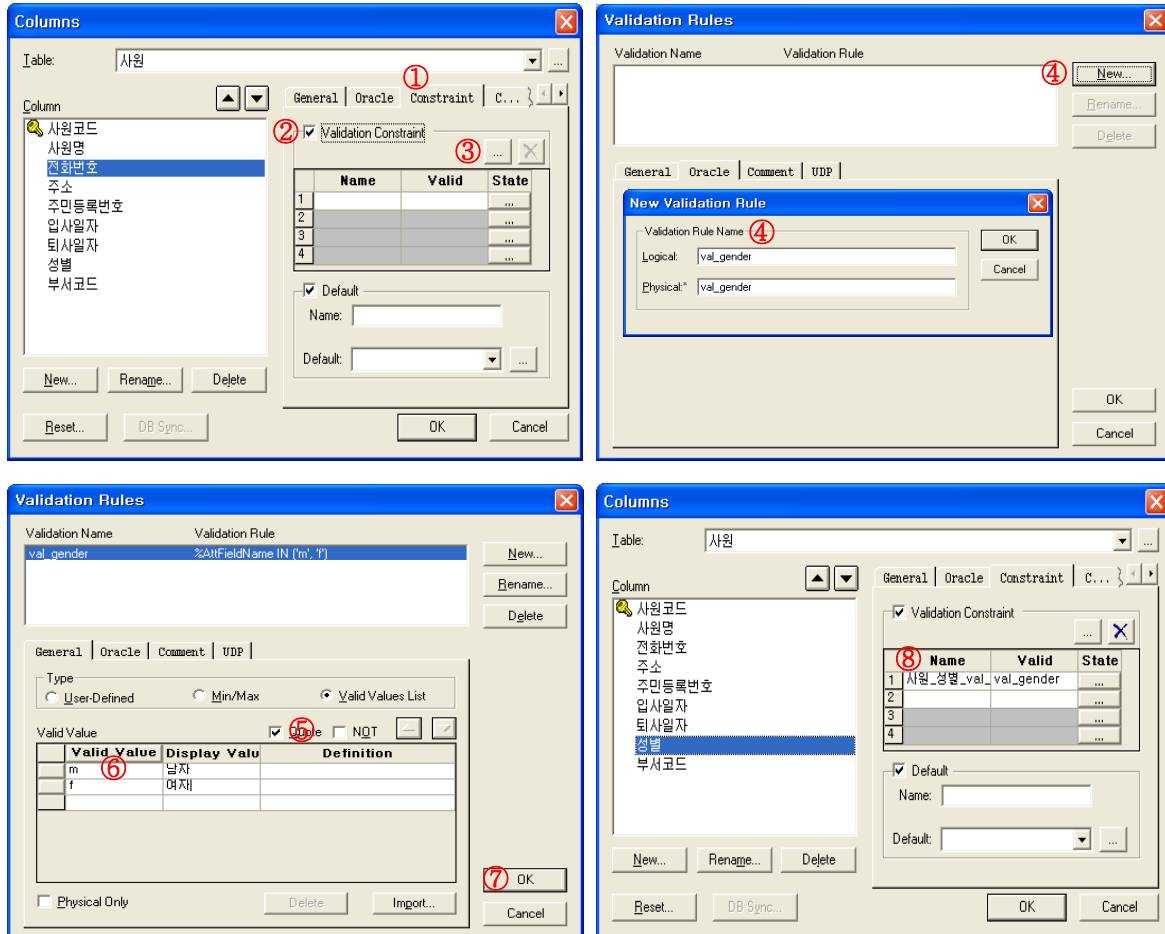
6.6.2 Column Constraint

ERwin 에서는 Column안에 저장할 수 있는 수용 가능한 값(Data)의 범위를 설정 할 수 있도록 지원한다. 즉 특정 Column에 제한된 데이터 값을 지정하기 위한 규칙(Rule)을 생성 하게 되는데 이를 Schema 생성 시 생성한 규칙에 대해서 자동으로 DDL 문을 생성할 수 있다.

6.6.2.1 Validation Rule 정의

Column Validation은 특정 Column에 대해 수용 가능한 값의 범위를 설정 할 수 있도록 지원한다. 예를 들어, 사원 테이블의 성별 컬럼은 업무규칙에 따라서 'm' 아니면 'f'의 값만 가지도록 제한 한다면 다음과 같은 Rule을 만들 수 있다.

성별 IN ('m', 'f')



- 작업절차 -----
- ① Column Editor → Constraint ② Validation Constraint 체크
 - ③ 확장 Button 클릭 ④ New → Name 입력
 - ⑤ Type 선택 → 규칙 입력 ⑦ OK 버튼 Click
 - ⑧ Name은 자동으로 입력 되나 사용자가 변경 할 수 있다.

Tip
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.

6.6.3 Domain

사용자가 정의하는 속성으로 공통 Attribute / Column 들이 가질 수 있는 데이터 타입이나 길이를 Domain으로 등록하여 관리한다. Domain은 Datatype, Length, Validation Rule, Default Value, Null Option 등의 정보가 포함 될 수 있다.

ERwin Data Modeler는 5가지 도메인을 기본으로 제공하며 이것을 상속 받는 Child 도메인의 계층적 구조를 이룬다.

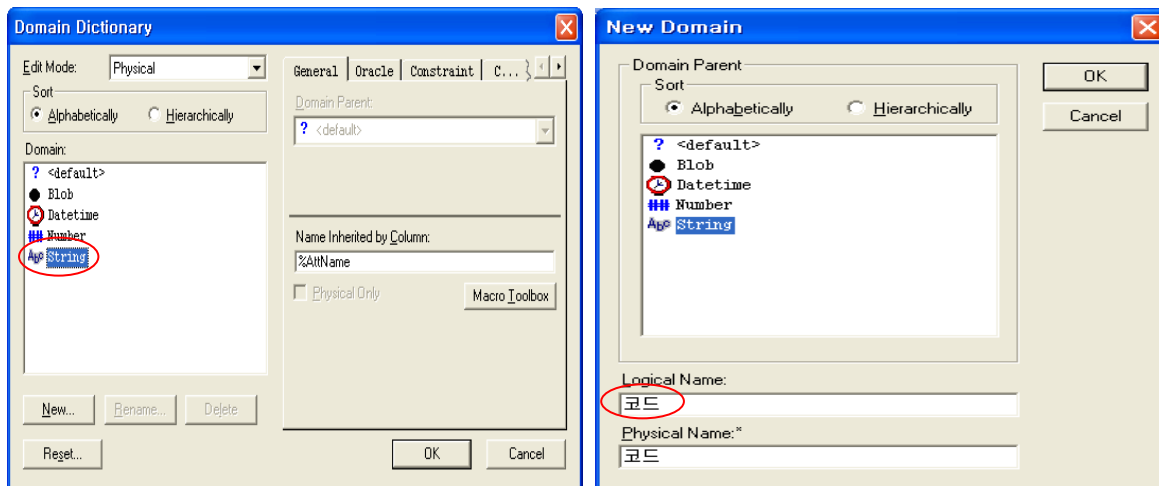
1) 5가지 기본 도메인

- Default : Target DBMS에서 선택한 디폴트 데이터 타입 설정
- Blob : 선택된 컬럼에 대용량 데이터 타입 설정
- String : 선택된 컬럼에 문자형 데이터 타입 설정
- Number : 선택된 컬럼에 숫자형 데이터 타입 설정
- Datetime : 선택된 컬럼에 날짜 데이터 타입 설정

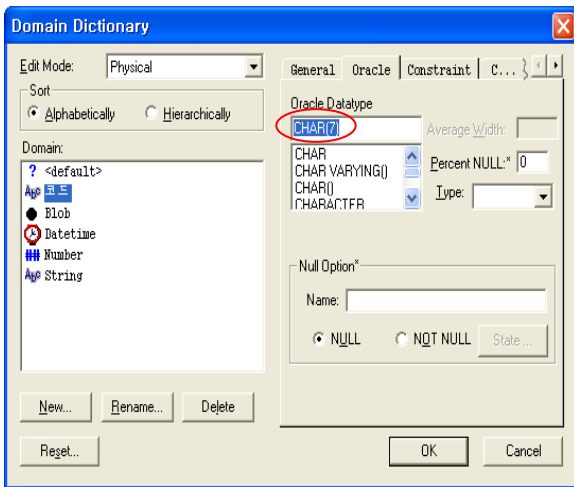
2) 도메인 사용시 장점

- 시간을 절약할 수 있다.
- 사용자의 실수를 줄여준다.
- 모델의 표준화를 지원한다.
- 생성한 도메인을 상속 받아 재사용 할 수 있다.

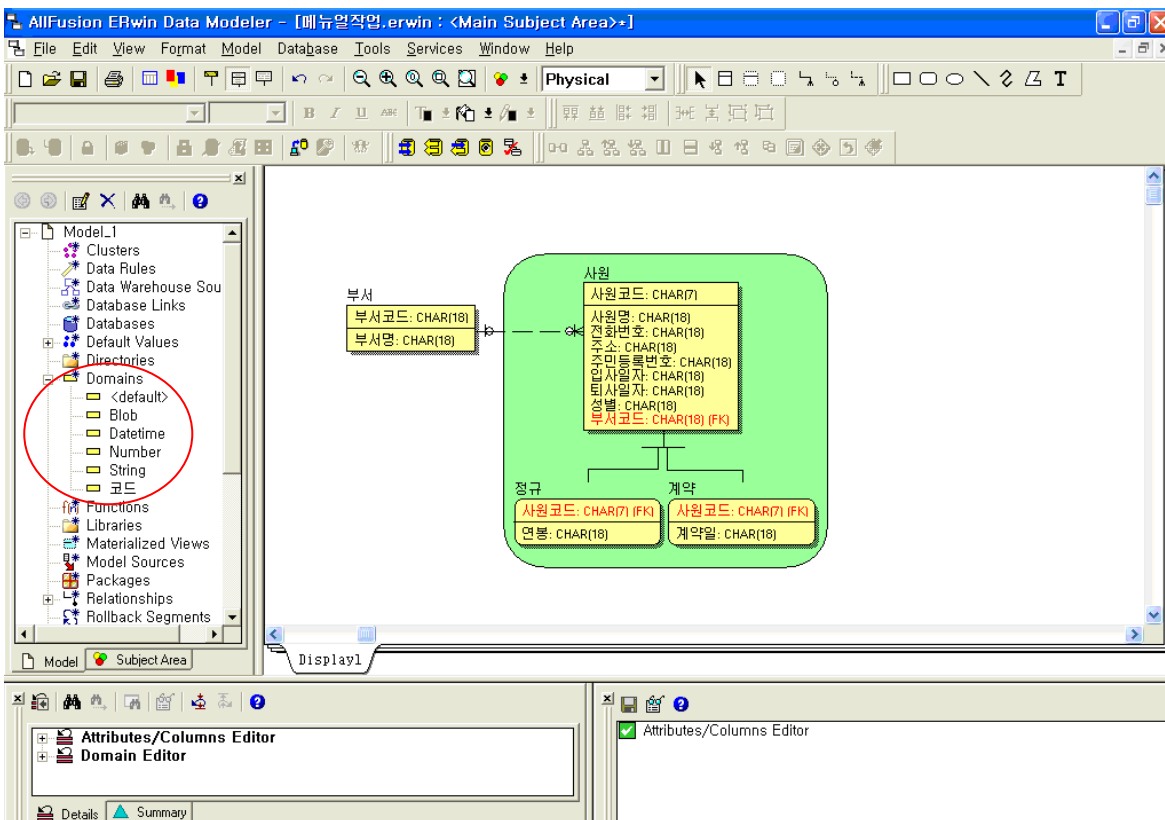
6.6.3.1 Domain 정의



- ① 기본 도메인을 선택하고 New를 클릭한다.
- ② 도메인 이름을 적고 OK 버튼 클릭한다.



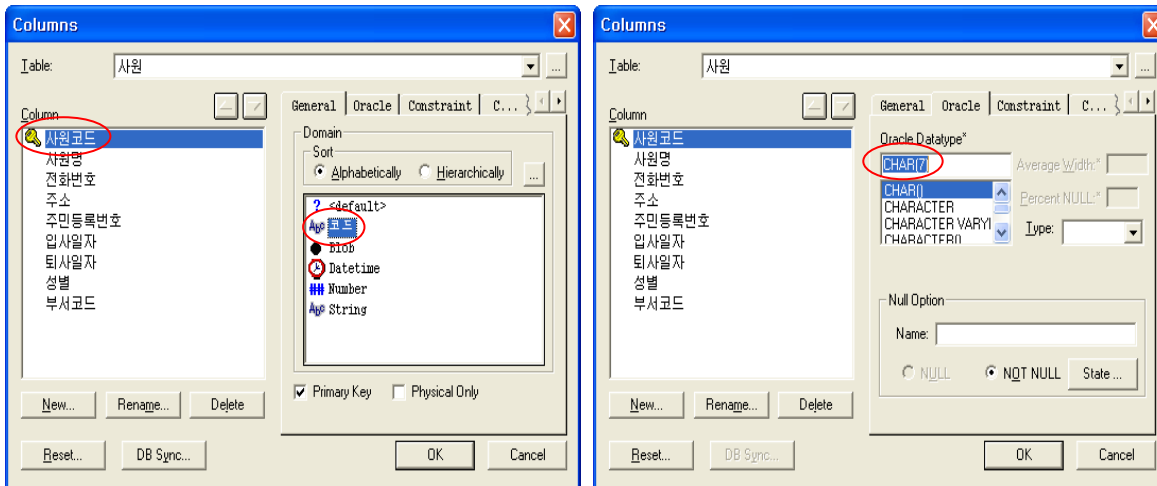
③ Oracle 탭 페이지 선택 후 타입을 정의하고 OK 버튼을 클릭한다.



④ Model Explorer > Domain 항목에 생성한 도메인이 추가된다.

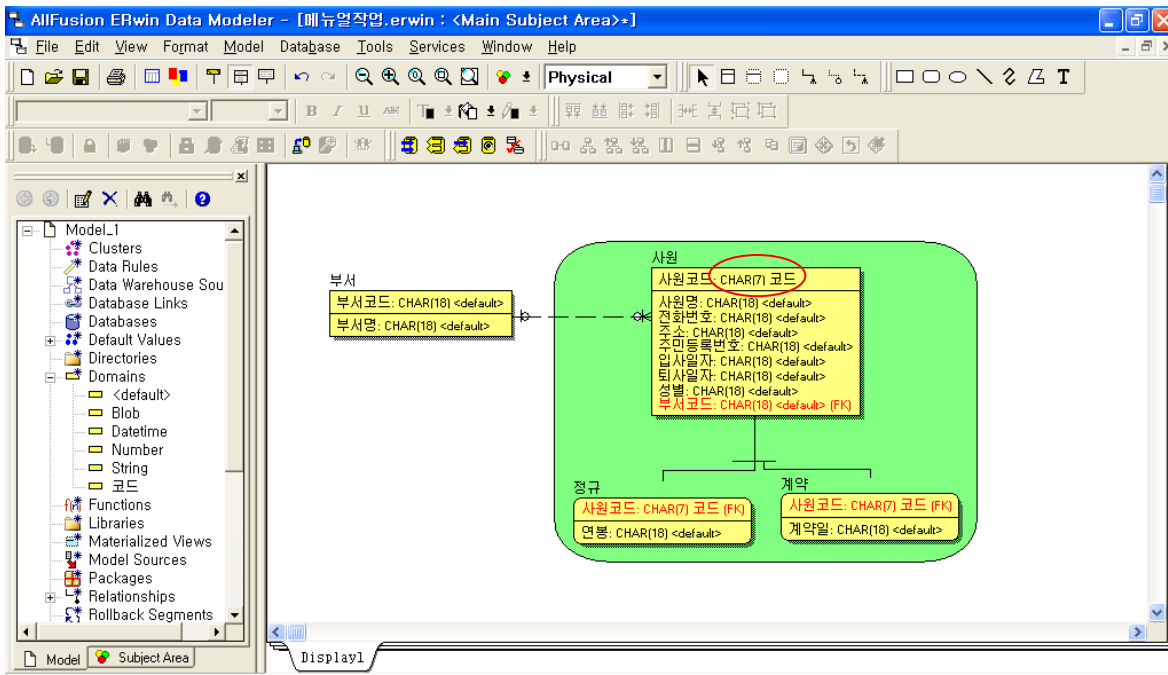
6.6.3.2 Domain Mapping

생성한 도메인과 컬럼을 매핑시킴으로서 컬럼은 도메인의 속성을(Datatype, Length) 상속 받는다.



① Column 선택 > Domain을 선택한다.

② Oracle 탭에서 상속 여부를 확인한다.



③ 모델을 보면 그림 과 같이 도메인의 속성을 상속 받는 것을 확인 할 수 있다.

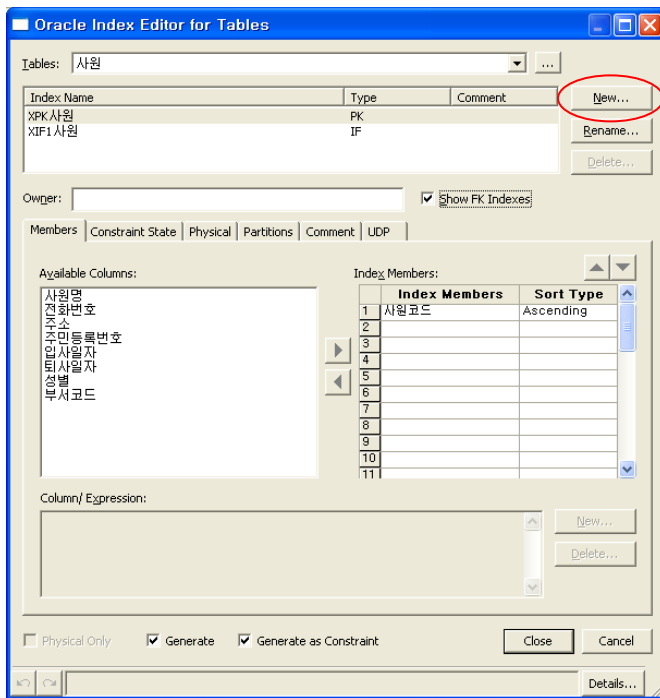
6.6.4 Index 정의

데이터의 검색 속도를 빠르게 하기 위해서 Table에 연관된 설계 타입을 정의하게 된다. 인덱스가 많다고 성능이 모두 향상 되는 것은 아니다. 인덱스는 질의에 있어 속도를 향상 시키지만 갱신과 같은 연산에는 오히려 속도가 저하 될 수 있다. 따라서 이 두 가지의 상관관계를 통해 가장 효과적인 인덱스 수를 고려해야 한다.

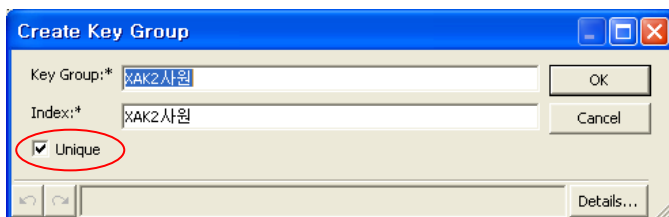
ERwin Data Modeler가 자동으로 생성하는 인덱스는 3가지가 있다.

- Primary Keys (PK)
- Alternate Keys (AK)
- Inversion Entries (IE)

이러한 속성들은 Logical 모델에서 작성된 Key Groups이 Physical 모델로 넘어오면서 인덱스로 매핑되는 것이다. 새로운 인덱스를 생성하는 방법은 다음과 같다.

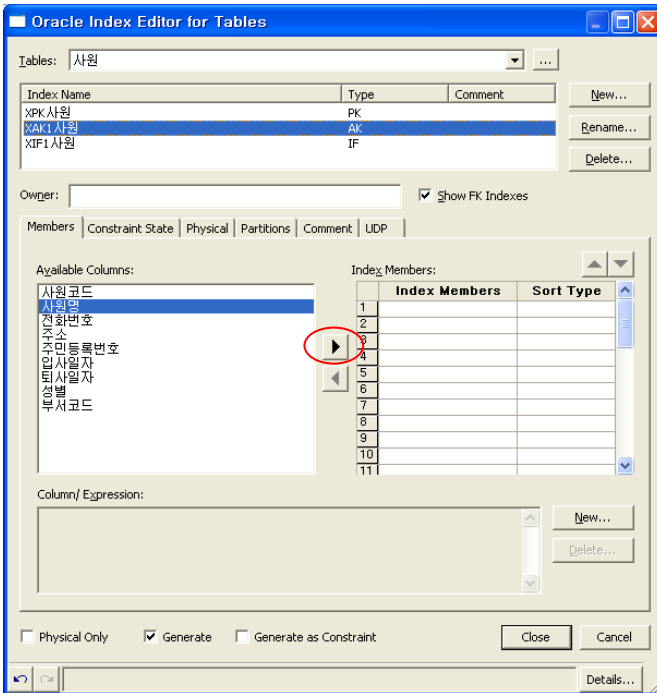


① Table 선택 > Index를 선택한다.

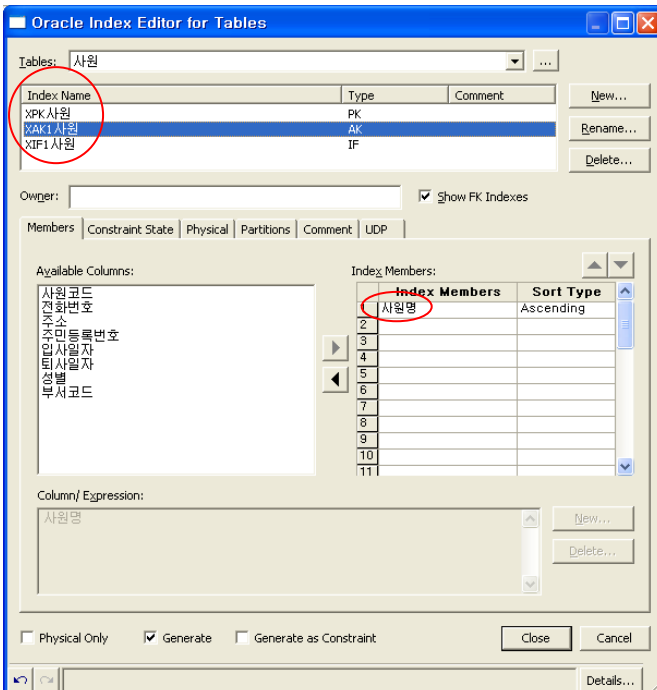


Tip
 Unique Check : AK 생성
 Unique Uncheck : IE 생성

② New 버튼을 클릭하고 Unique 여부를 선택하고 OK 버튼을 누른다.



- ③ 생성한 Index를 선택한다.
- ④ 인덱스 Member로 등록할 컬럼을 선택하고 ▶ 을 누른다.



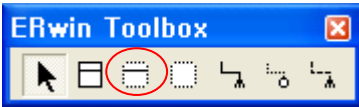
- ⑤ 인덱스 목록에 생성한 인덱스가 추가되며 인덱스를 선택하면 Member를 확인 할 수 있다.
- ⑥ Target DMBS에 따라 다양한 Physical Property를 정의 할 수 있다.

6.7 View Table

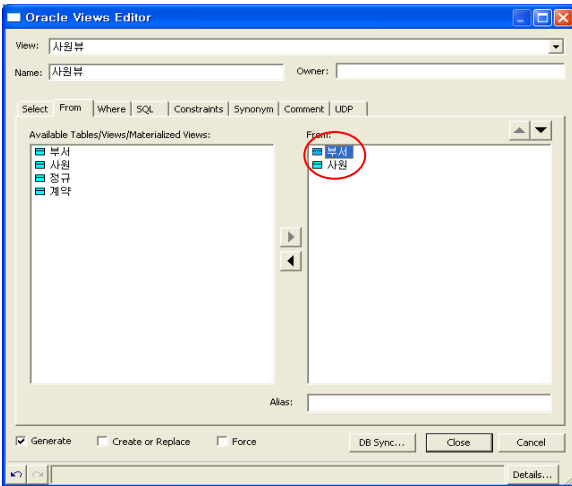
View Table은 하나 또는 그 이상의 테이블에 저장된 데이터를 사용자가 편집하는 형태로 ‘가상의 Table’ 또는 ‘저장된 질의’ 라고 한다. 이러한 View는 여러 Table로부터 일련의 Column을 포함 할 수 있으며 기초 Table 만이 아니라 또 다른 View를 통해서 데이터를 가져올 수 있다.

View를 사용하는 이점에는 Table에 향상된 보안을 제공하고 데이터의 복잡성을 숨긴다. 또한 사용자의 명령어를 간결화 하고 데이터의 또 다른 표현방법을 제공하게 된다.

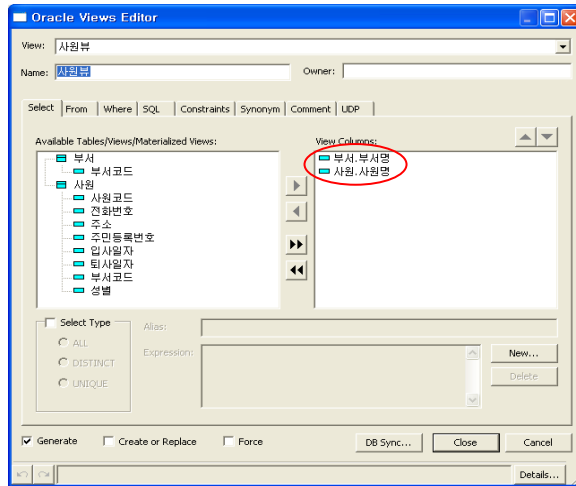
View Table을 만드는 방법은 다음과 같다.



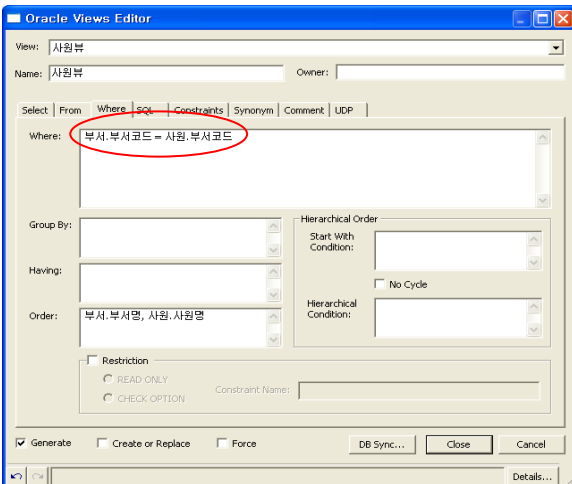
① View 아이콘을 선택하여 새로운 View를 생성한다.



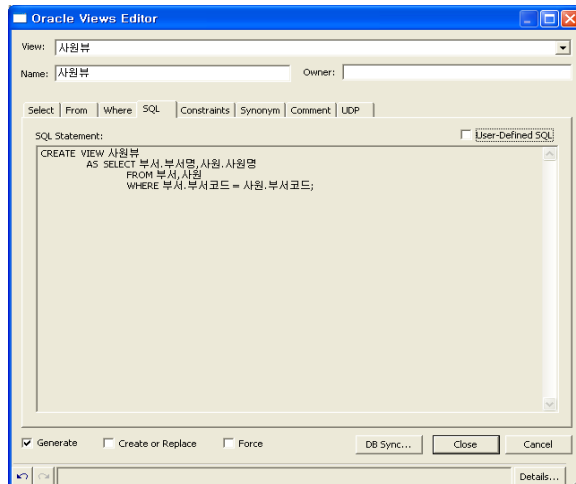
② FROM절에 가져올 Table을 Attach 한다.



③ SELECT절에 가져올 Column을 Attach 한다.



④ WHERE절에 조건을 입력한다.



⑤ SQL 탭에서 DDL문을 확인 할 수 있다.

7. Transform

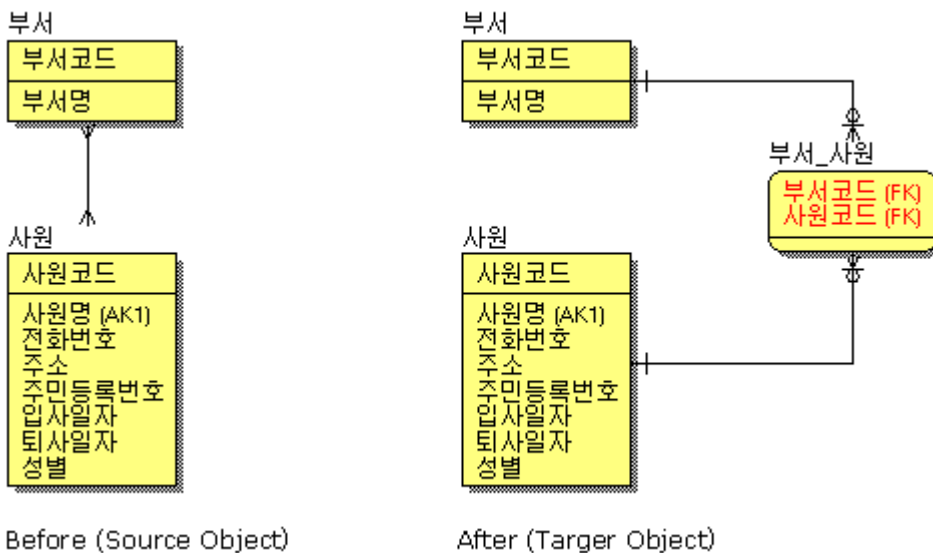
7.1 Transform 정의

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.

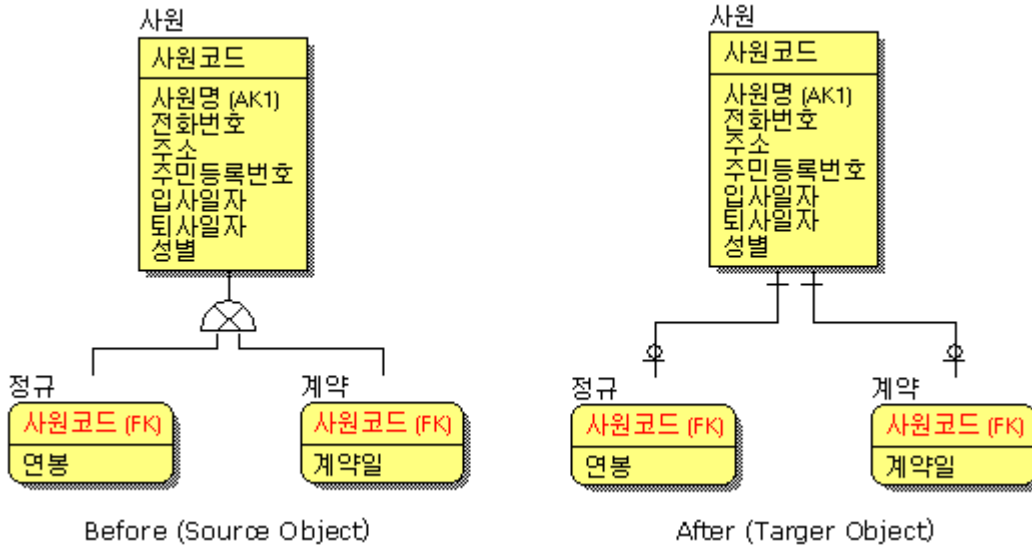
[지원되는 Transforms]

- 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

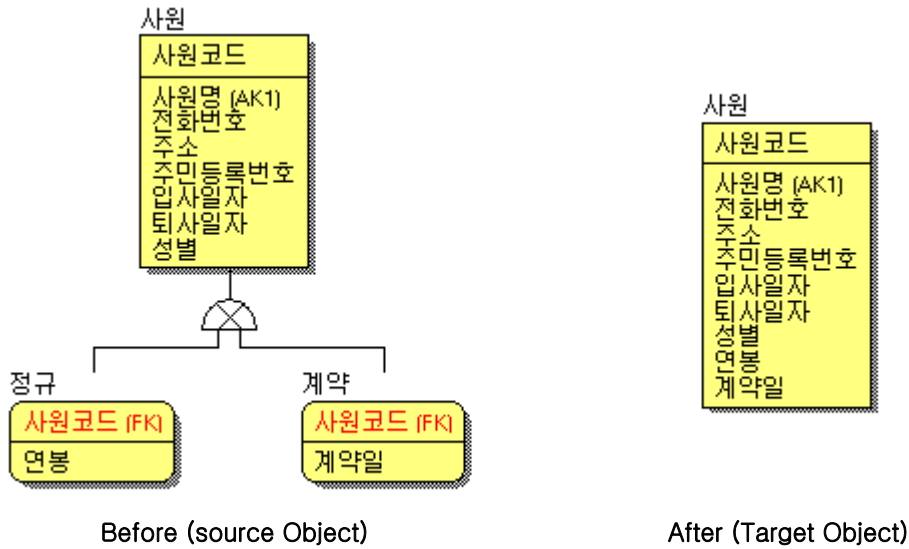
1) Many to Many Transform : M 대 M 관계를 자동으로 변환한다.



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



3) Supertype - Subtype Roll Up : 일반화 관계에 참여하는 자식 엔티티의 컬럼을 부모로 올린다.

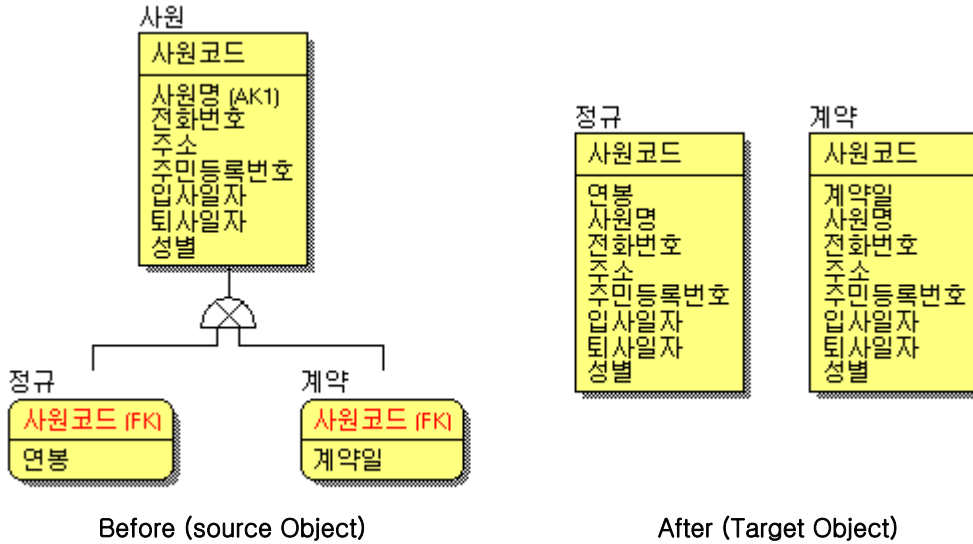


Tip

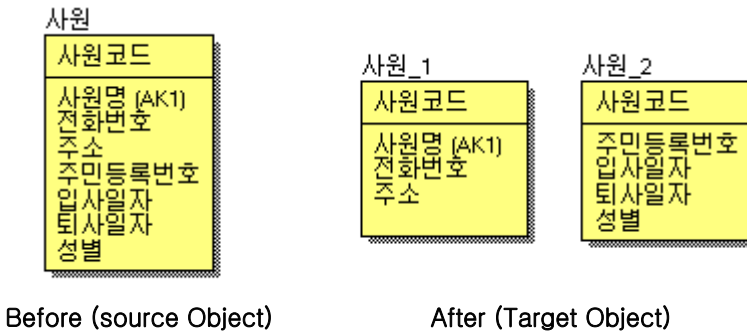
Source Object : Transform 기능으로 변환되기 전의 Object Set을 의미한다.

Target Object : Transform 기능으로 변환된 후의 Object Set을 의미한다.

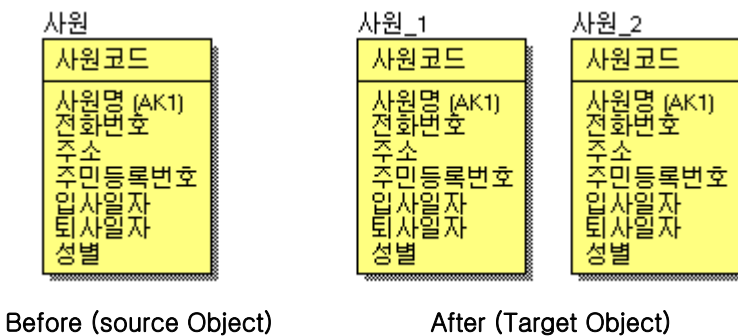
4) Supertype-Subtype Roll Down : 일반화 관계에 참여한 부모 엔티티의 컬럼을 자식으로 내린다.



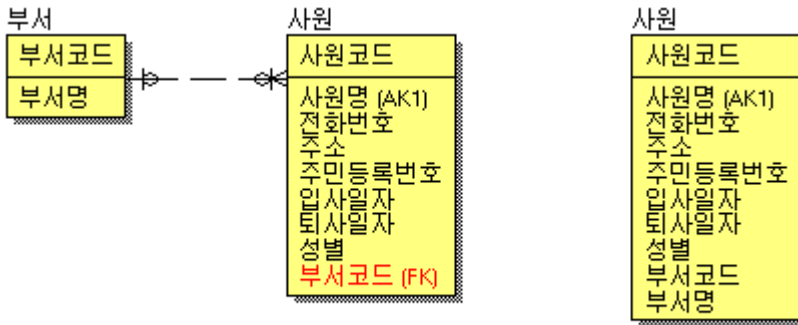
5) Vertical Partition : 해당 Table을 수직분할 한다.



6) Horizontal Partition : 해당 Table을 수평분할 한다.



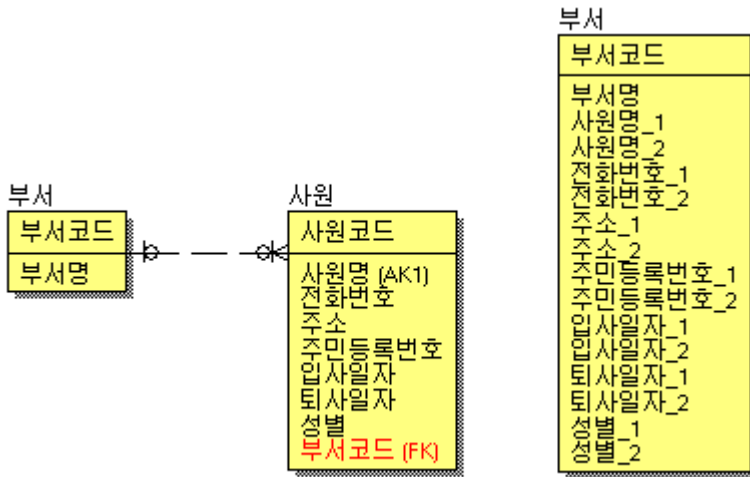
7) Roll - Down Denormalization : 부모 테이블의 Column이 자식 Table로 이동한다. (역정규화)



Before (source Object)

After (Target Object)

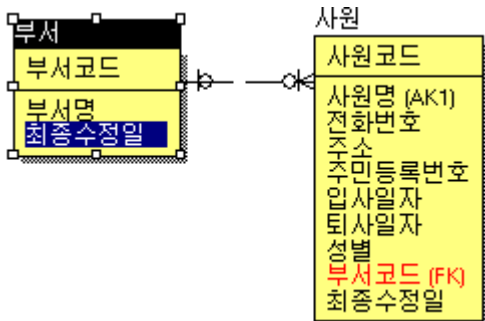
8) Roll - Up Denormalization : 자식 테이블의 Column이 부모 Table로 이동한다. (역정규화)



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.

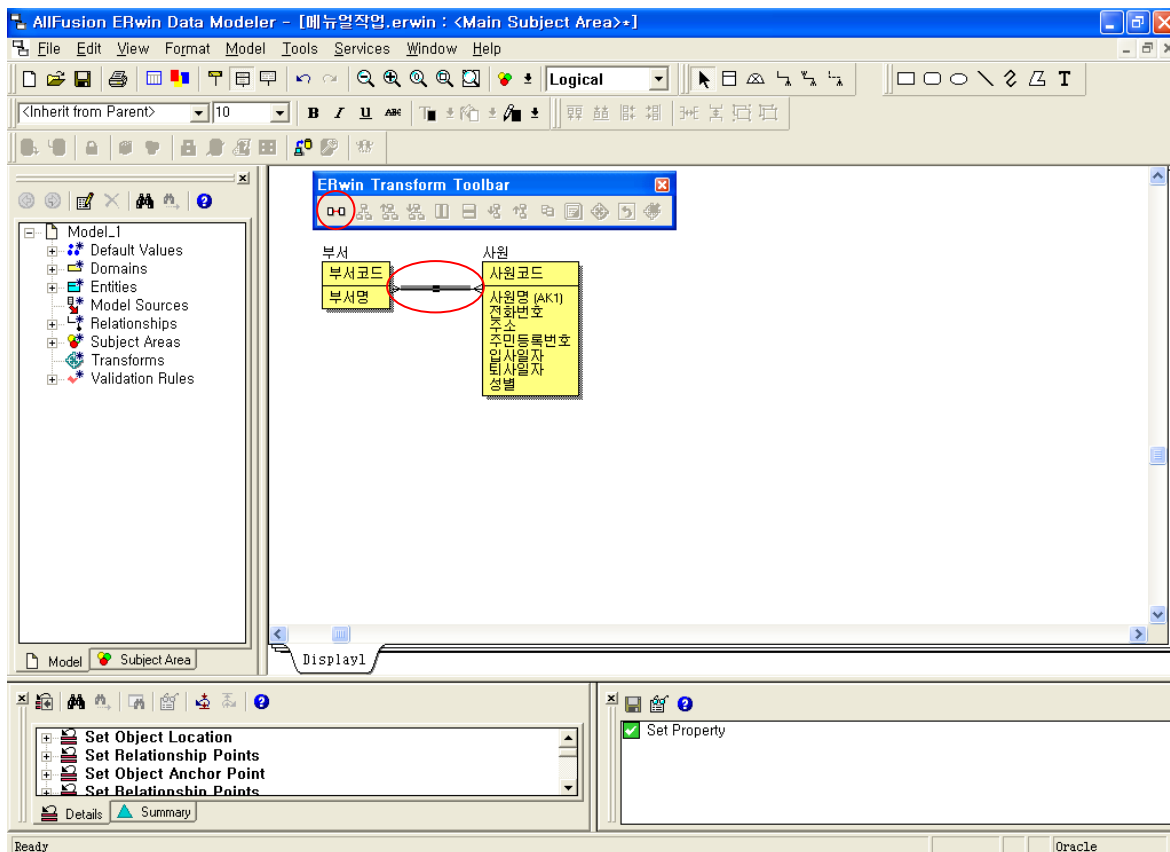
7.2 Transform 기본 원리

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 모델상에서는 확인 할 수 없다. 그러나 Model Explorer를 통해서 Transform 작업 리스트를 확인 할 수 있으며, 모델러는 작업을 되돌리거나 고정시킬 수 있다.

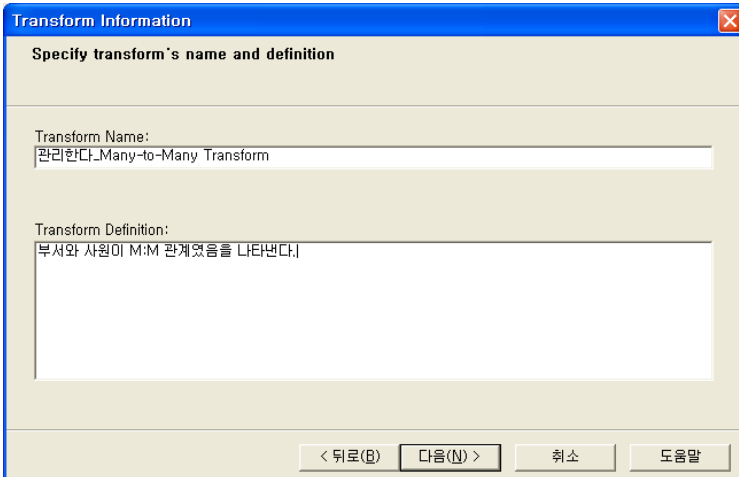
1) Many to Many Transform을 이용한 M : M 관계의 자동변환



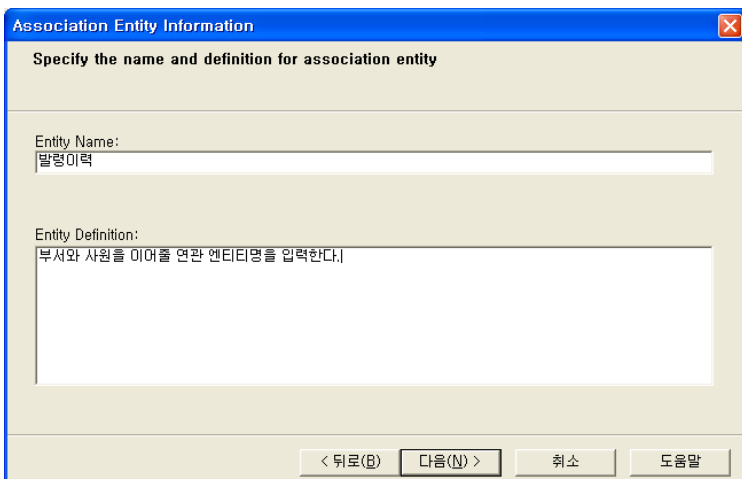
- ① M : M 관계선을 선택하고 Many to Many Transform 아이콘을 선택한다.



② Many to Many Transform Wizard가 실행된다.



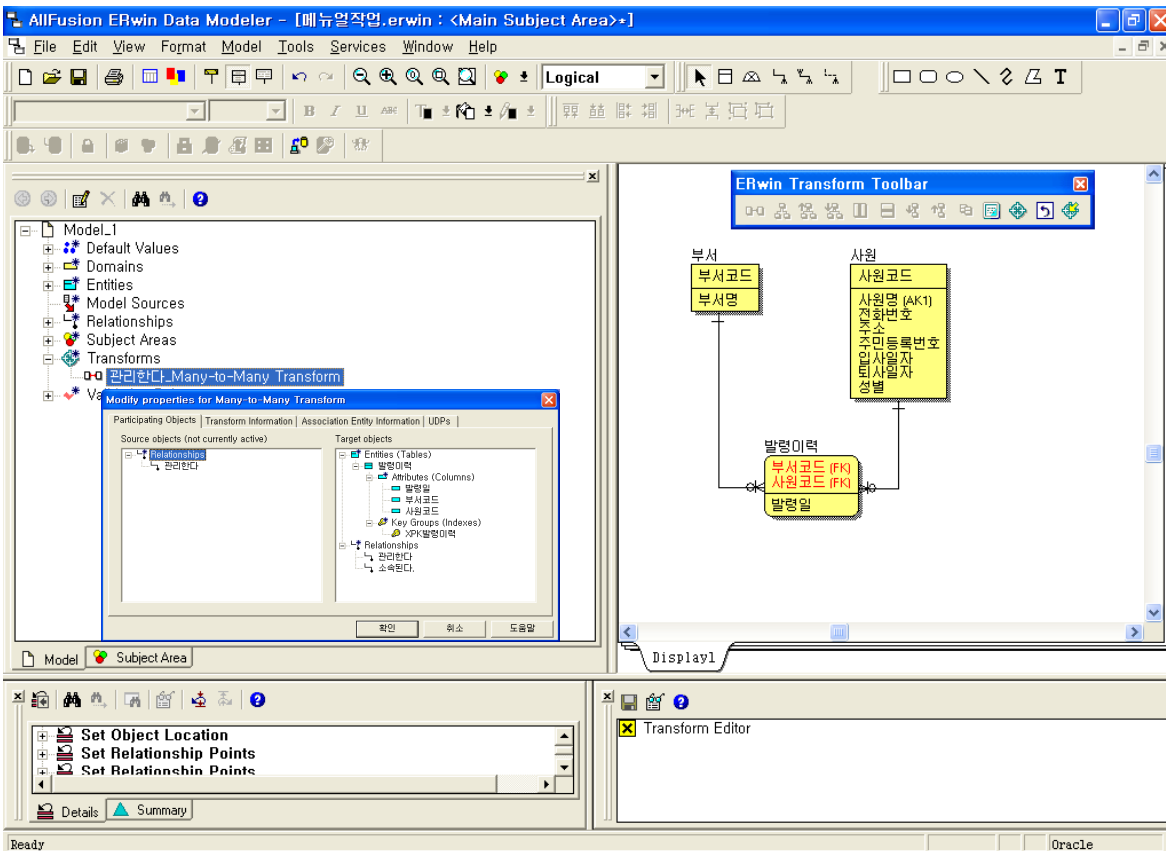
③ Transform의 이름과 Definition을 입력한다.



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



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

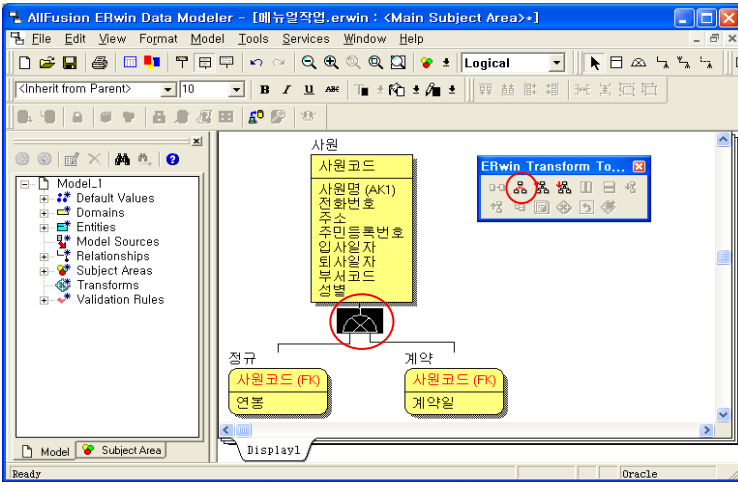


⑥ Model Explorer에서 리스트를 확인 할 수 있으며 Properties에서 세부 정보를 수정할 수 있다.

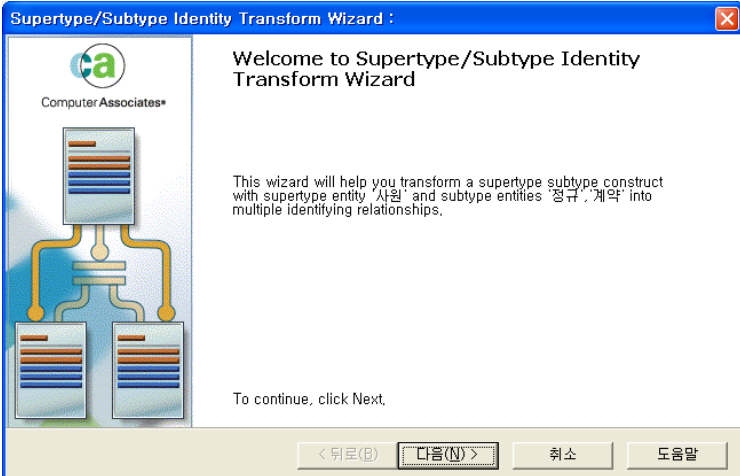
Tip

- : 변환 전의 Source Object 보기
- : 변환 후 Target Object 보기
- : 변환 작업을 취소한다.
- : 변환 내용으로 모델을 고정시킨다. (Source Object를 확인 할 수 없다.)

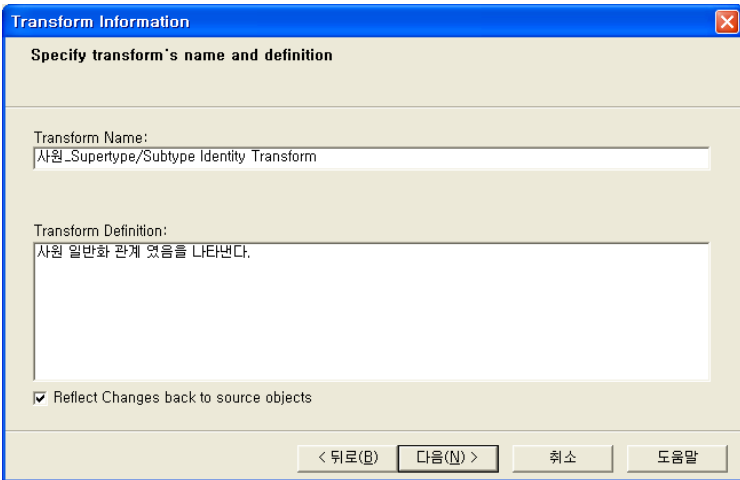
2) Supertype - Subtype Identity를 이용한 일반화 관계의 자동변환



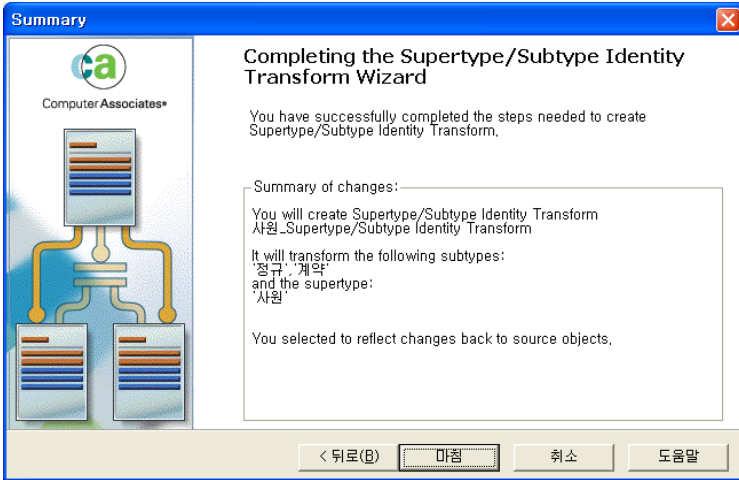
① 일반화 관계 Symbol을 선택하고 Supertype - Subtype Identity 아이콘을 선택한다.



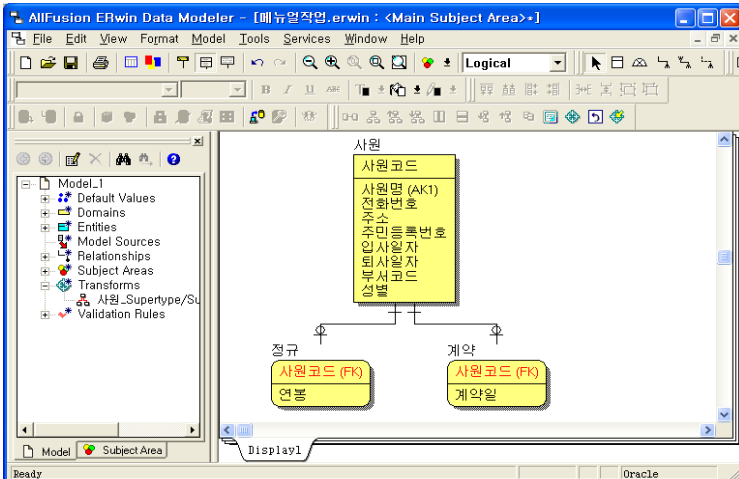
② Supertype / Subtype Identity Transform Wizard가 실행된다.



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

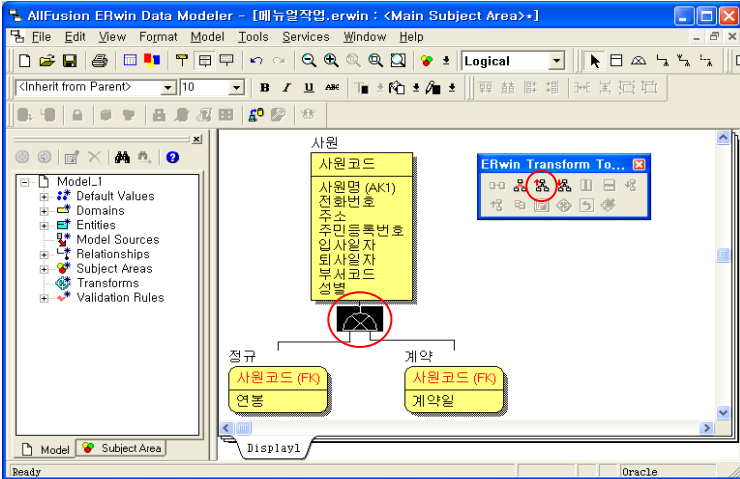


④ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.

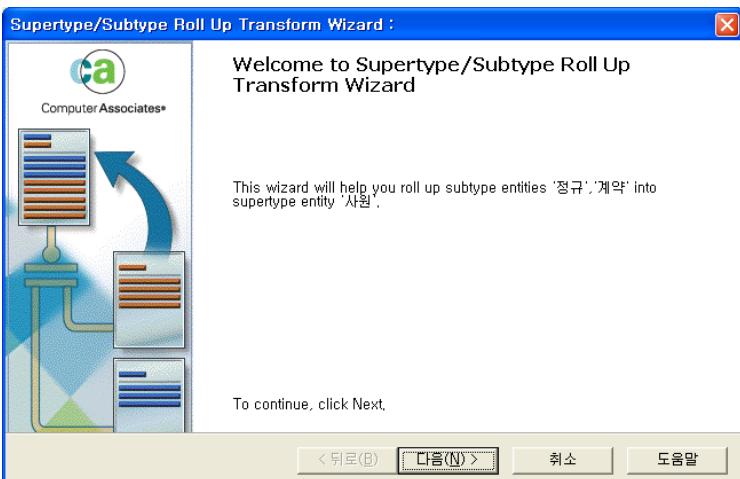


⑤ 일반화 관계가 부모 - 자식 관계로 변환된다.

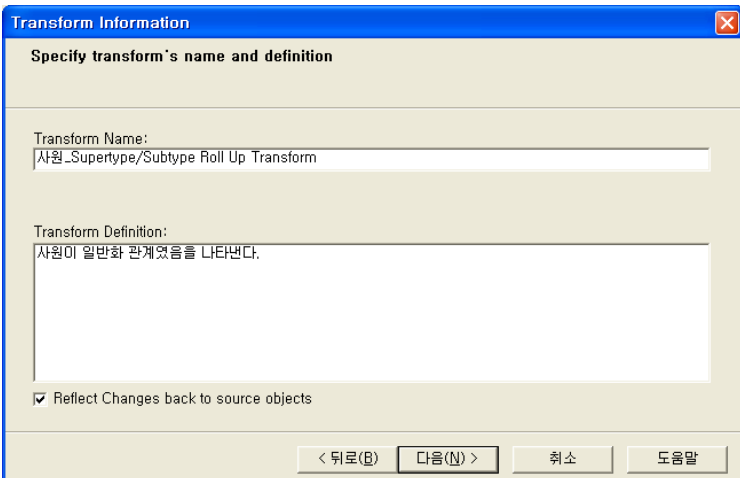
3) Supertype – Subtype Roll Up을 이용한 일반화 관계의 자동변환



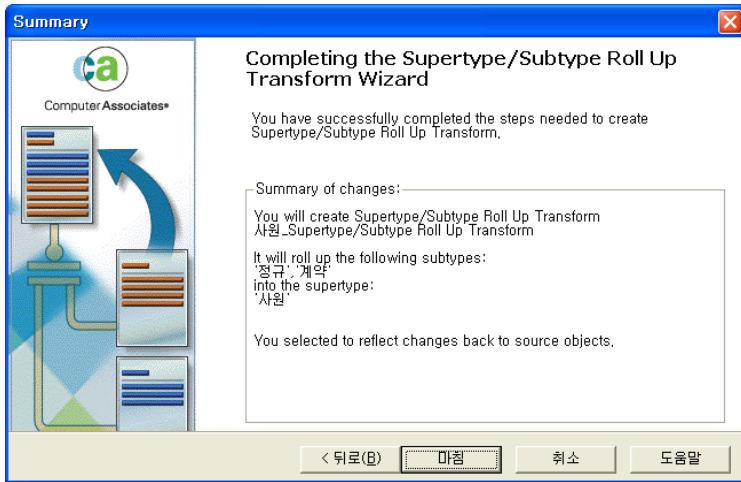
① 일반화 관계 Symbol을 선택하고 Supertype – Subtype Roll UP 아이콘을 선택한다.



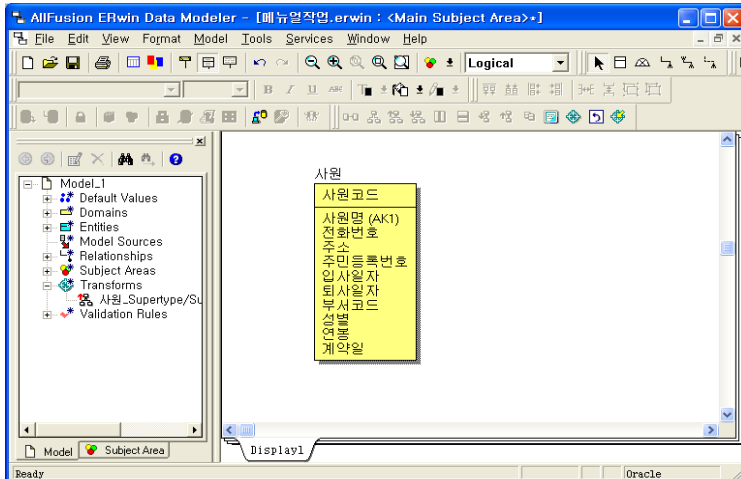
② Supertype / Subtype Roll UP Transform Wizard가 실행된다.



③ Transform의 이름과 Definition을 입력한다.

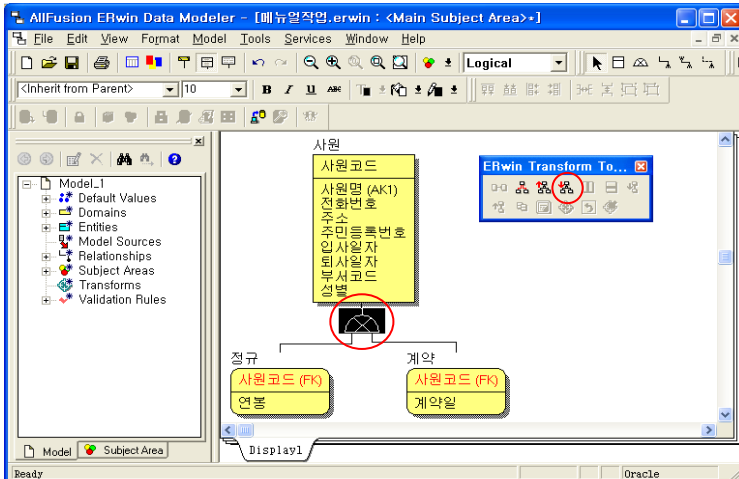


④ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.

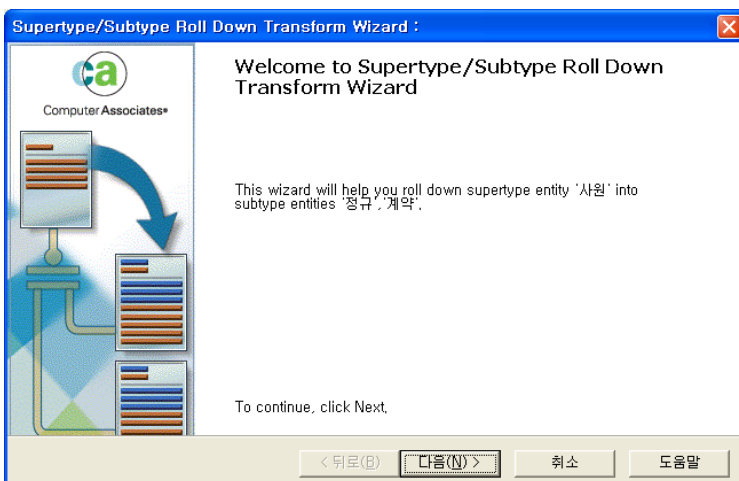


⑤ 일반한 관계의 하위(자식) 엔티티의 속성들이 상위(부모) 엔티티의 속성으로 포함된다.

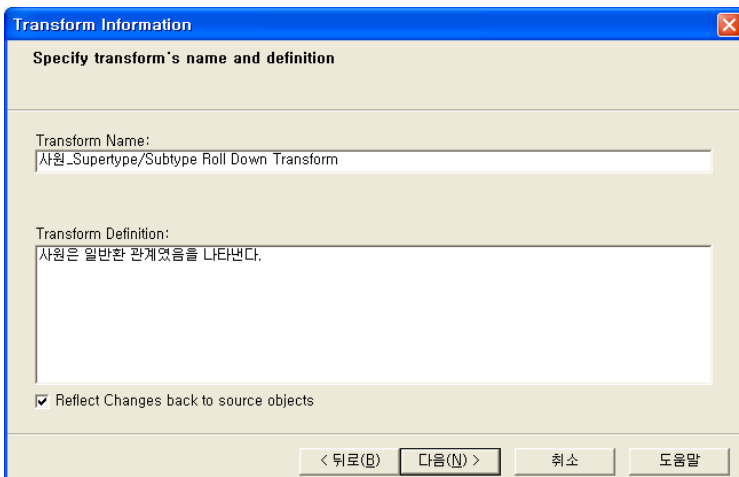
4) Supertype – Subtype Roll Down을 이용한 일반화 관계의 자동변환



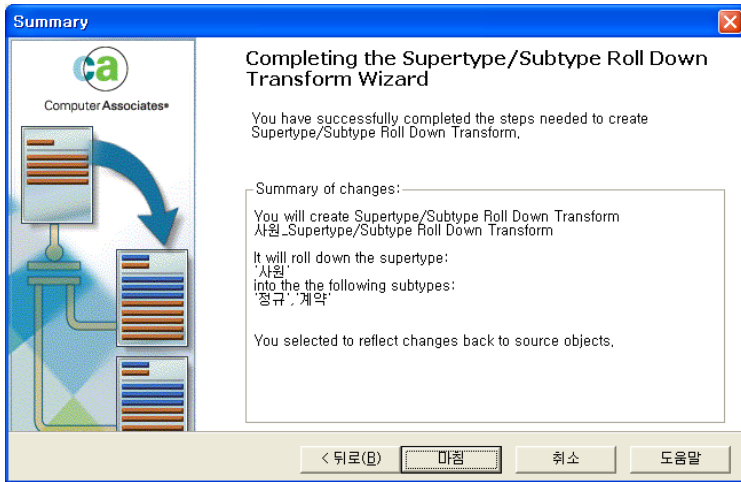
① 일반화 관계 Symbol을 선택하고 Supertype – Subtype Roll Down 아이콘을 선택한다.



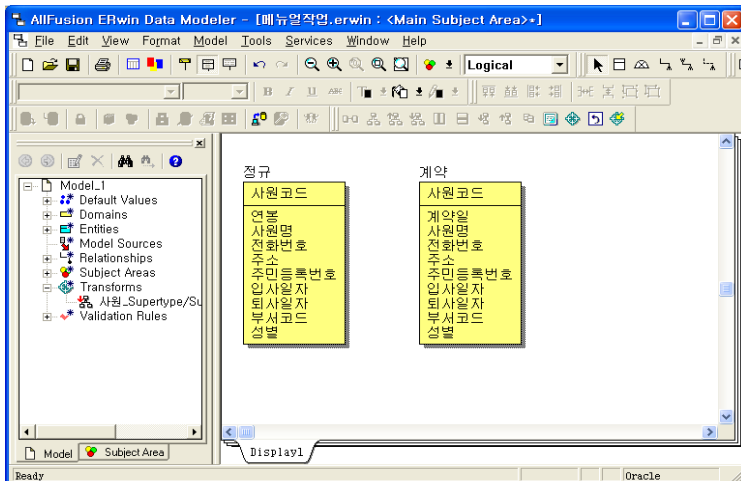
② Supertype / Subtype Roll Down Transform Wizard가 실행된다.



③ Transform의 이름과 Definition을 입력한다.



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



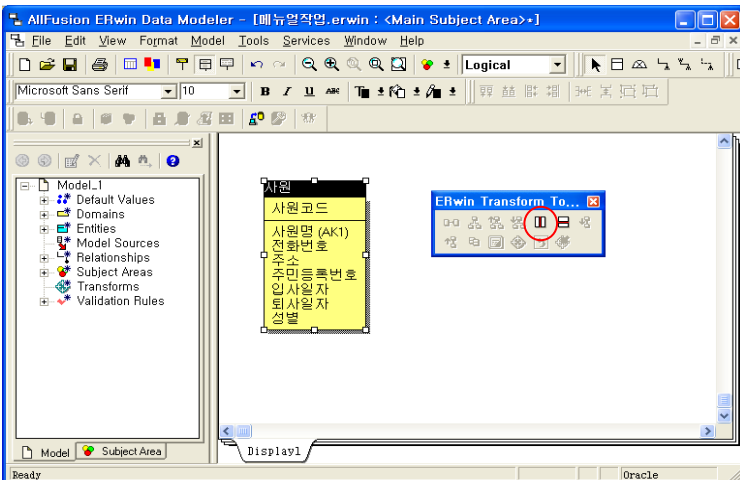
⑤ 일반한 관계의 상위(부모) 엔티티의 속성들이 하위(자식) 엔티티의 속성으로 포함된다.

5) Vertical Partition을 이용한 엔티티의 수직적 분할

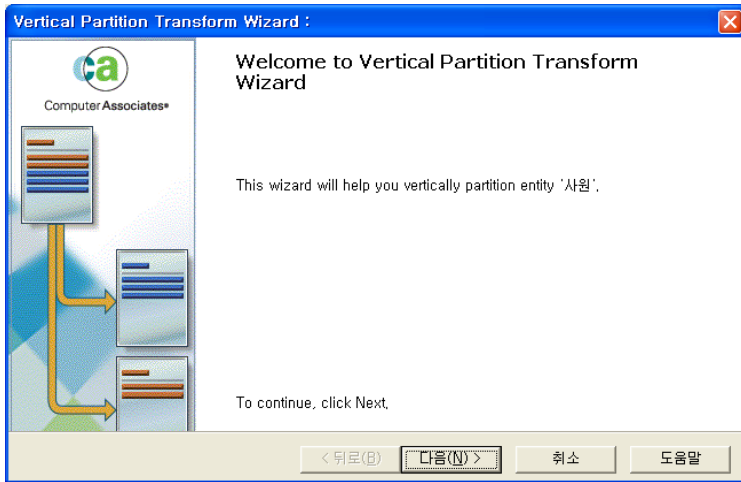
테이블의 한 세트로 테이블의 개수는 원하는 대로 생성할 수 있다. 테이블 이름은 디폴트로 설정 했을 때 Source table 명 + 일련번호가 붙는 형태이다. Target Table의 컬럼 형태는 Source Table에서 분리된 컬럼의 Set 이다.

CUSTOMER

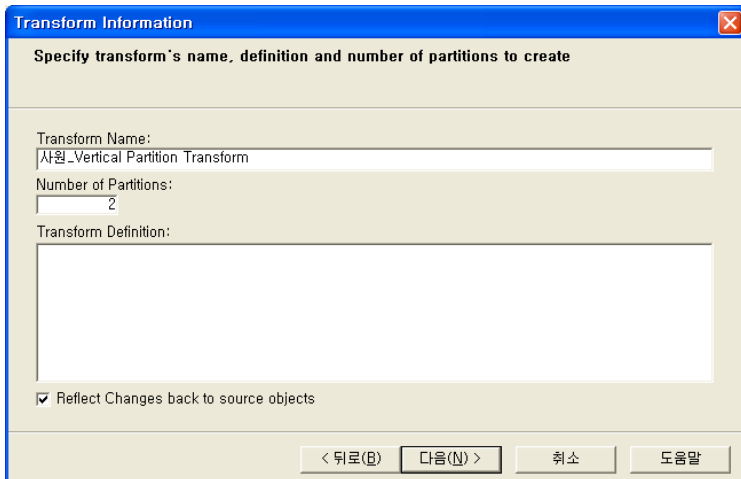
key	last_name	first_name	street_address	city	state	credit_card_number	expiration_date
48402	Adams	Joyce	340 North Street	Plainsboro	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48403	Cleary	Emily	1503 Sussex Street	Princeton	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48404	Evans	Joseph	10 North Maple Avenue	Princeton	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48405	Martin	David	3505 South Street	Plainsboro	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48406	Meyers	Chris	100 Main Street	Cranbury	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48407	Nevil	Adam	890 Scotts Corner Road	Plainsboro	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48408	Patterson	Michael	1900 Dey Road	Plainsboro	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48409	Tuttle	Richard	95 New Hyde Street	Cranbury	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99
48410	Young	Stella	1040 Cranbury Road	Cranbury	NJ	xxxx-xxxx-xxxx-xxxx	31-Dec-99



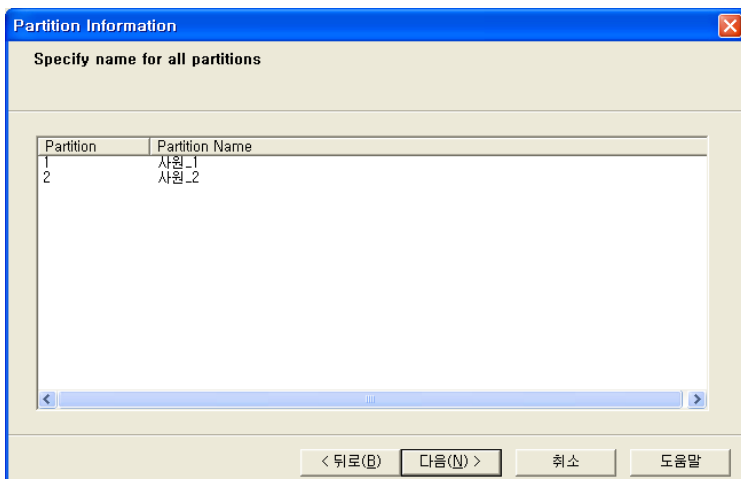
① 엔티티를 선택하고 Vertical Partition 아이콘을 선택한다.



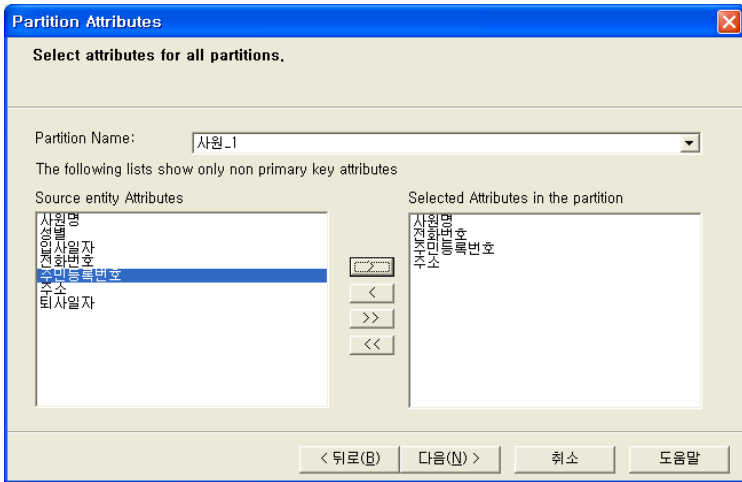
② Vertical Partition Transform Wizard가 실행된다



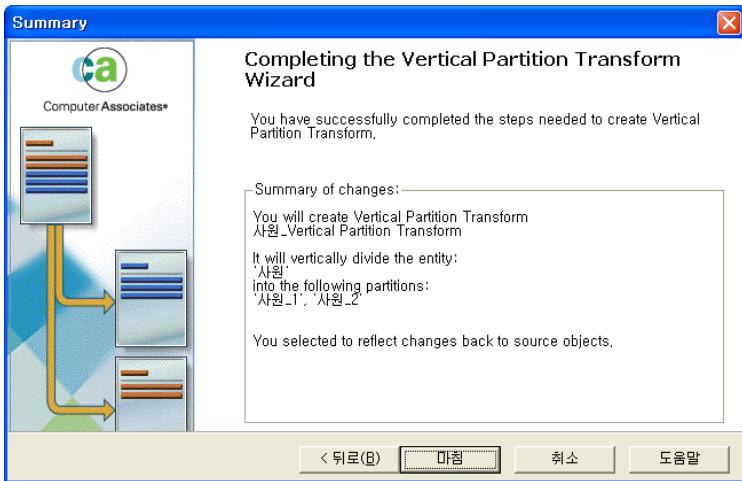
③ Transform의 이름과 분할할 테이블 개수, Definition을 입력한다.



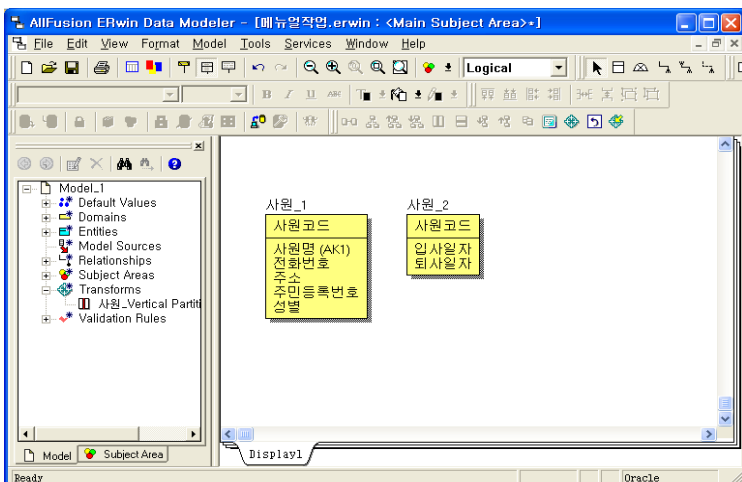
④ Partition Name이 기본적으로 생성되며 더블 클릭을 통해 수정이 가능하다.



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



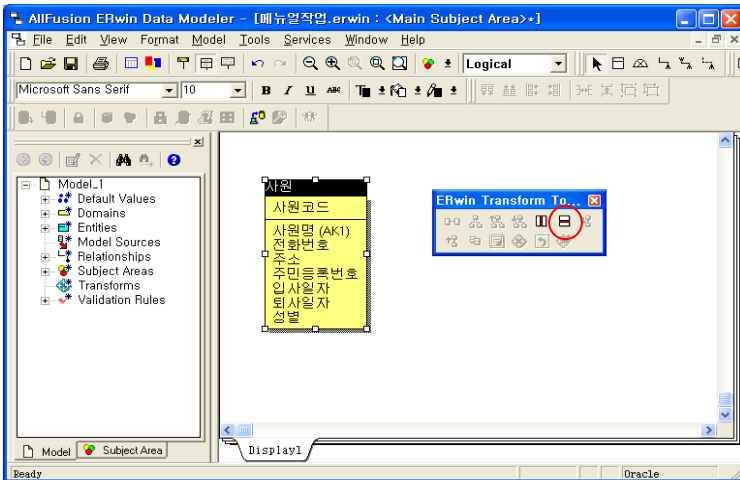
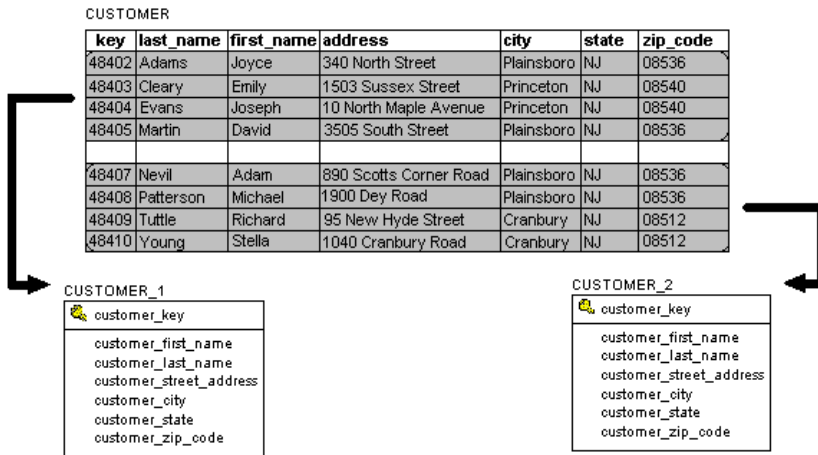
⑥ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



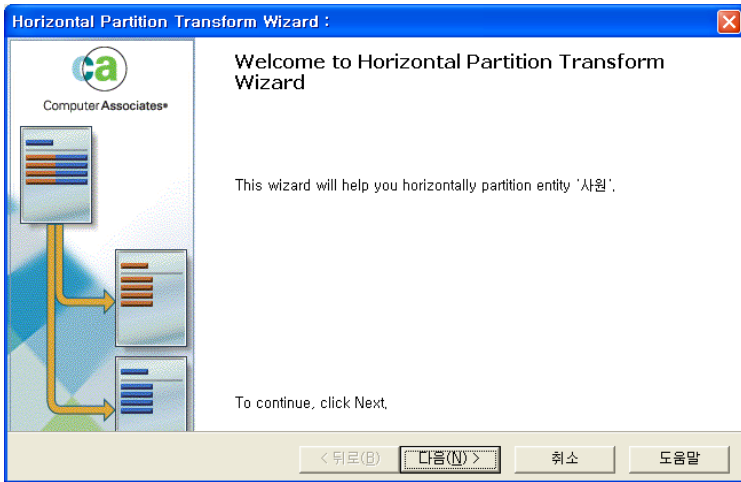
⑦ 테이블이 분할된 것을 확인 할 수 있다.

6) Horizontal Partition을 이용한 엔티티의 수평적 분할

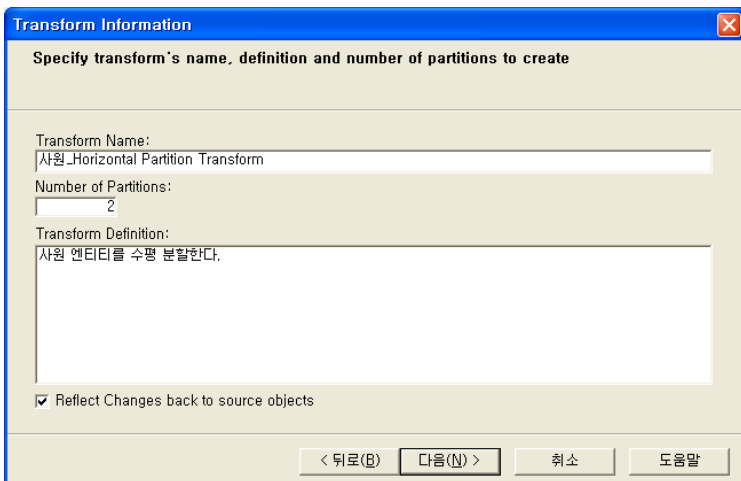
테이블의 한 세트로 테이블의 개수는 원하는 대로 생성할 수 있다. 테이블 이름은 디폴트로 설정 했을 때 Source Table 명 + 일련번호가 붙는 형태이다. Target Table의 컬럼 형태는 Source Table과 동일한 컬럼 Set 이며, 나뉘어진 테이블에 Row의 분리규칙을 입력할 수 있다.



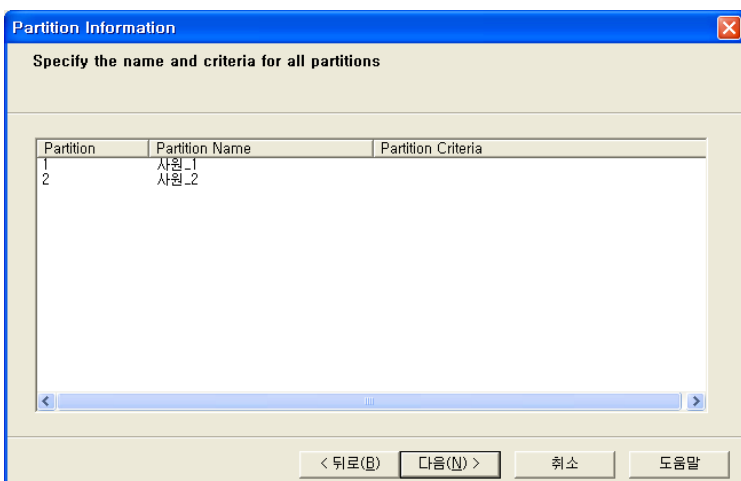
① 엔티티를 선택하고 Horizontal Partition 아이콘을 선택한다.



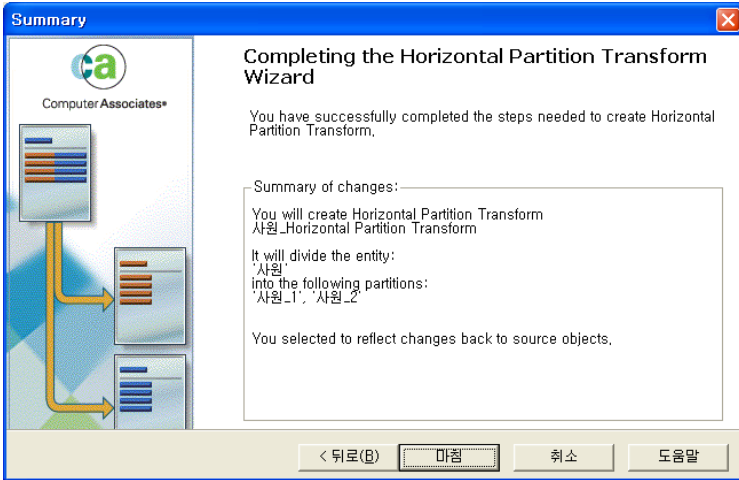
② Horizontal Partition Transform Wizard가 실행된다



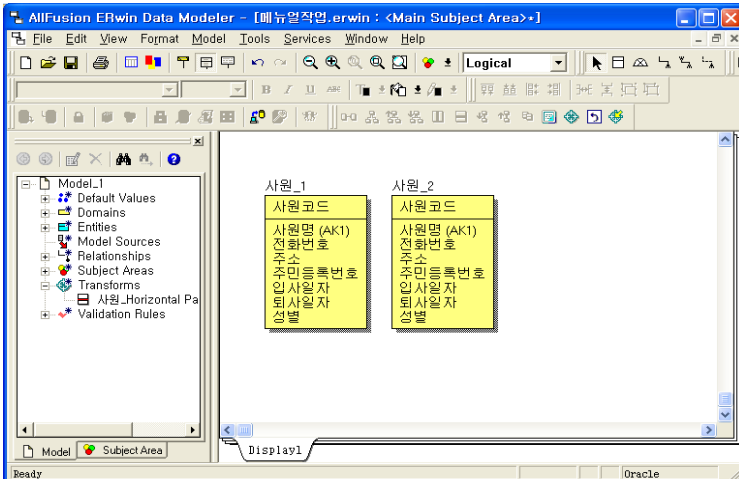
③ Transform의 이름과 분할할 테이블 개수, Definition을 입력한다.



④ Partition Name이 기본적으로 생성되며 더블 클릭을 통해 수정이 가능하다.



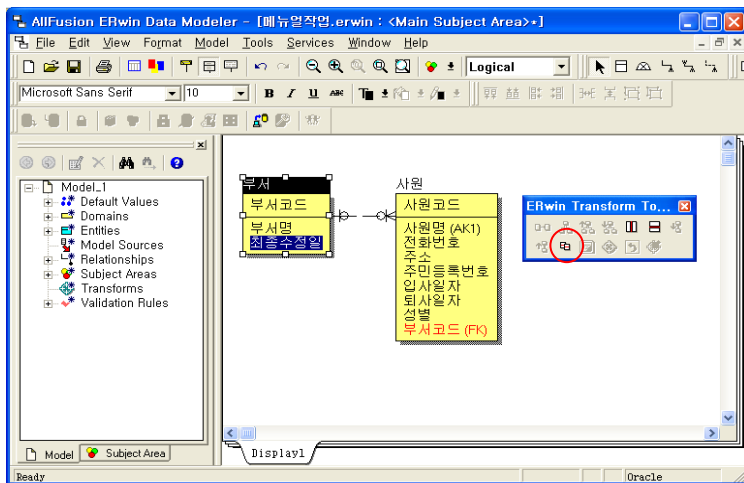
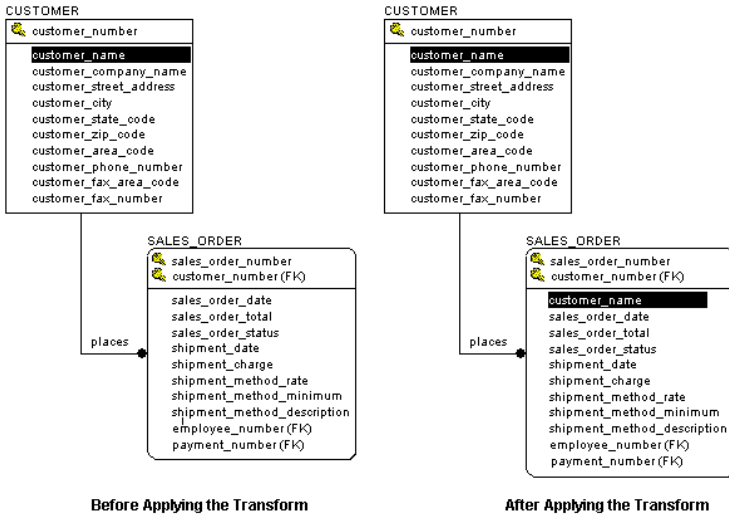
⑤ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



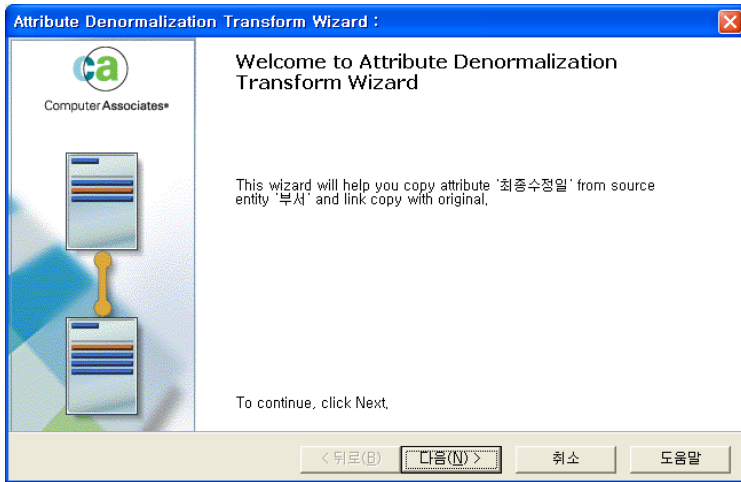
⑥ 테이블이 분할된 것을 확인 할 수 있다.

7) Column Denormalization을 이용한 Column 복사

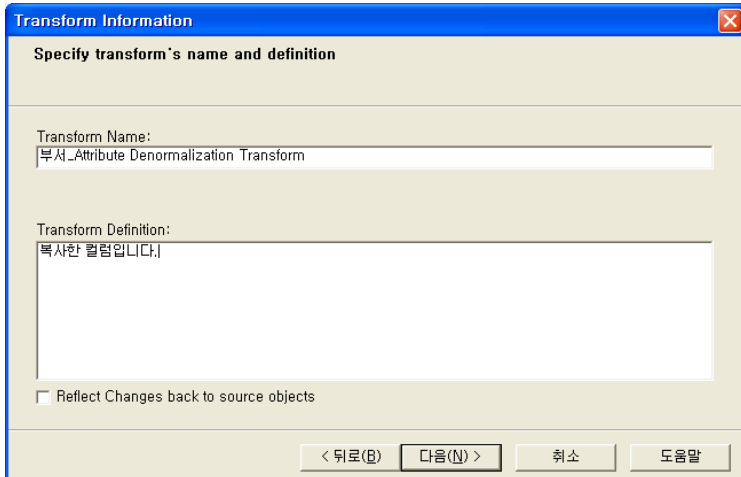
하나의 컬럼으로 Source Column에서 복사해 온 것이다. 이름과 속성은 Source Column과 동일하다.



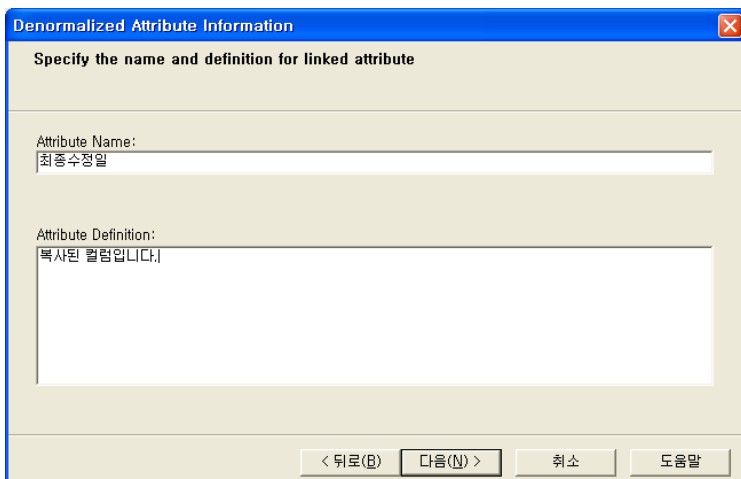
① 복사할 컬럼을 선택하고 Column Denormalization 아이콘을 선택한다.



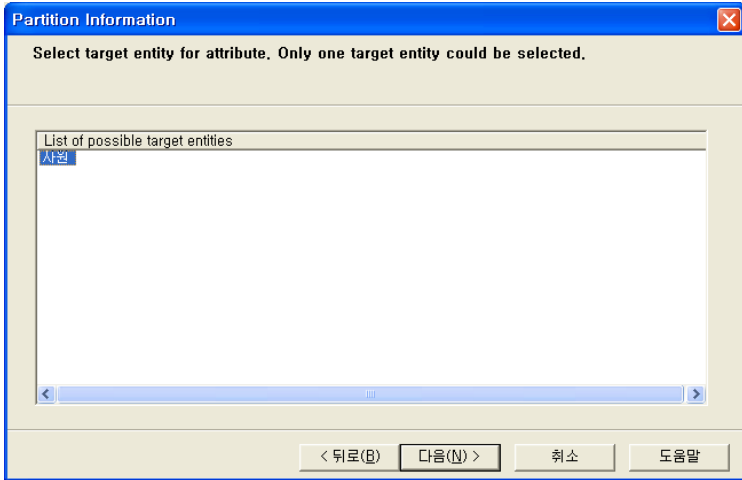
② Column Denormalization Transform Wizard가 실행된다.



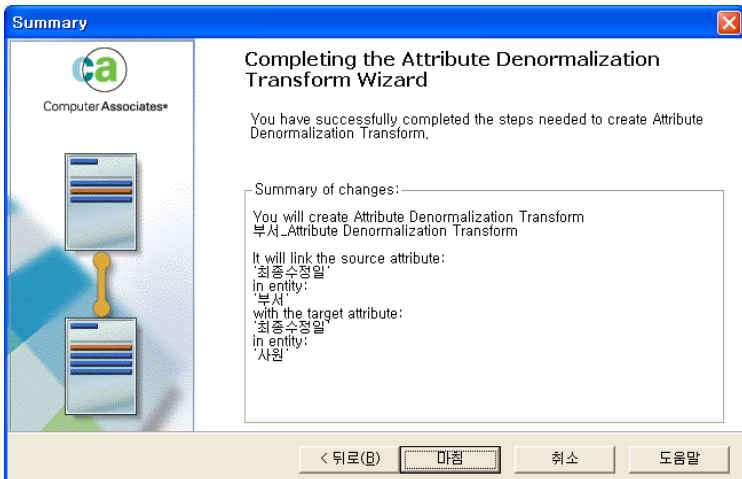
③ Transform의 이름과 Definition을 입력한다.



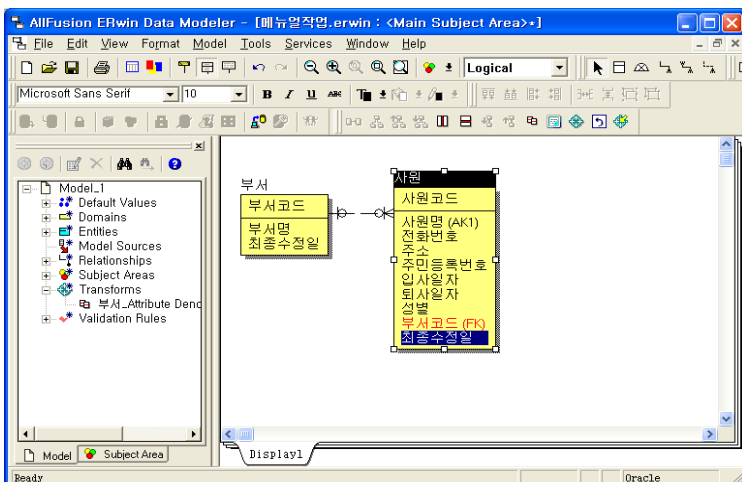
④ Attribute 이름은 자동 입력된다. Definition을 입력한다.



⑤ Target Entity를 선택한다.



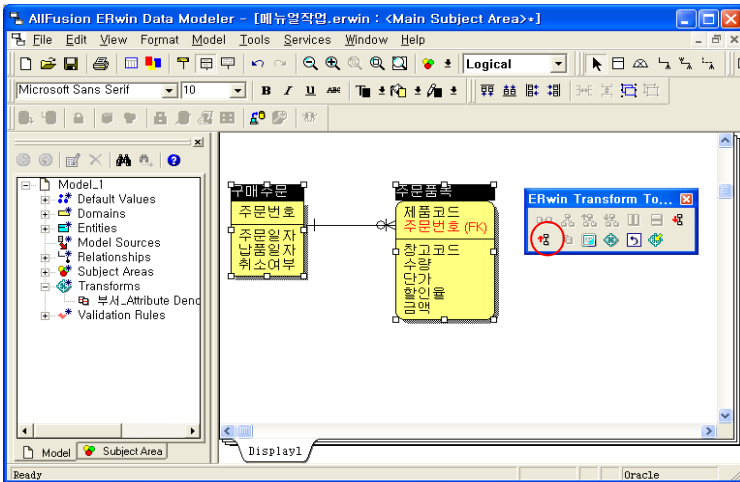
⑥ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



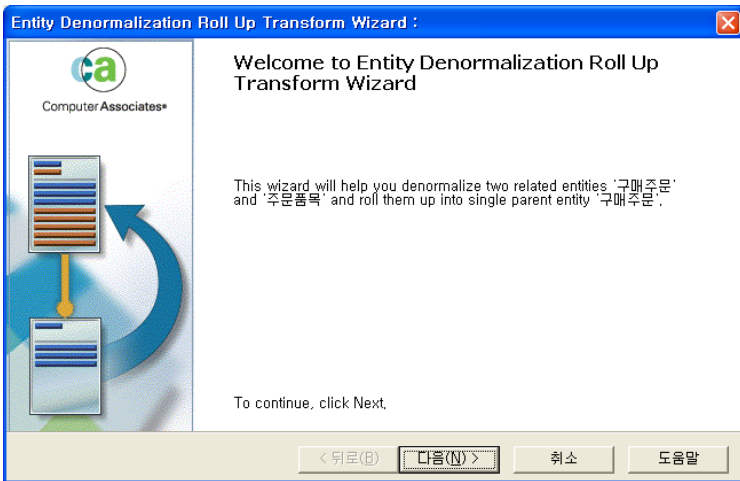
⑦ Attribute가 복사된 것을 확인 할 수 있다.

8) Roll – Up Denormalization을 이용한 Table 역정규화

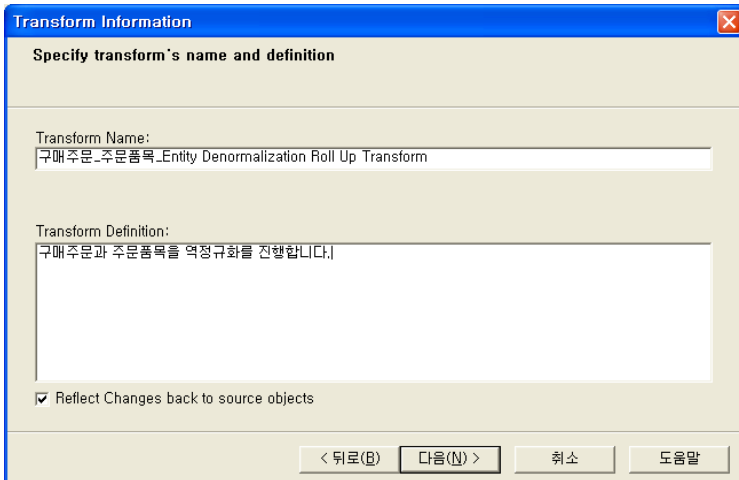
관계를 맺고 있는 두 테이블이 정규화 되었을 때 데이터 검색 속도가 느린 현상이 발생하는 경우가 발생한다. 이는 테이블간에 서로 조인을 걸어야 하기 때문에 발생하는 것으로 이를 방지하기 위해 테이블을 역정규화 할 수 있다. Roll – Up Denormalization은 자식 엔티티의 속성들을 부모 엔티티로 포함시켜 역정규화를 진행한다.



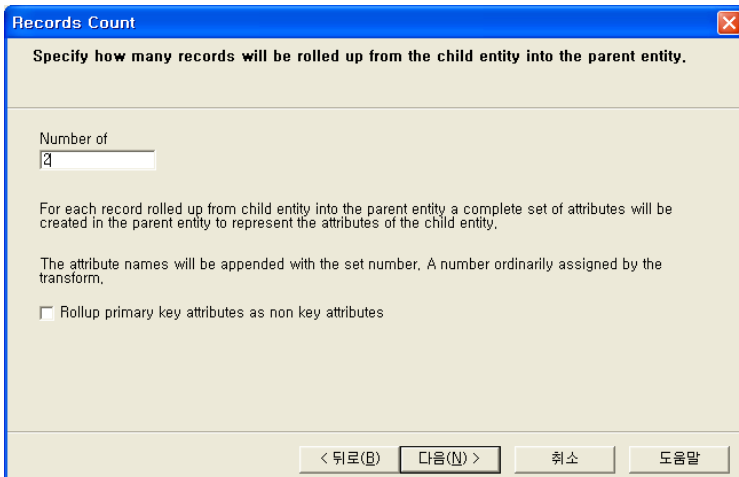
① 관계가 있는 두 테이블을 선택하고 Roll – Up Denormalization 아이콘을 선택한다.



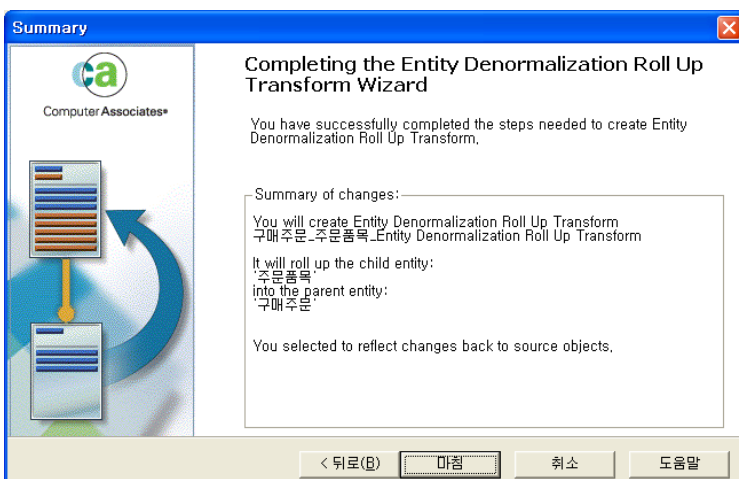
② Roll – Up Denormalization Transform Wizard가 실행된다.



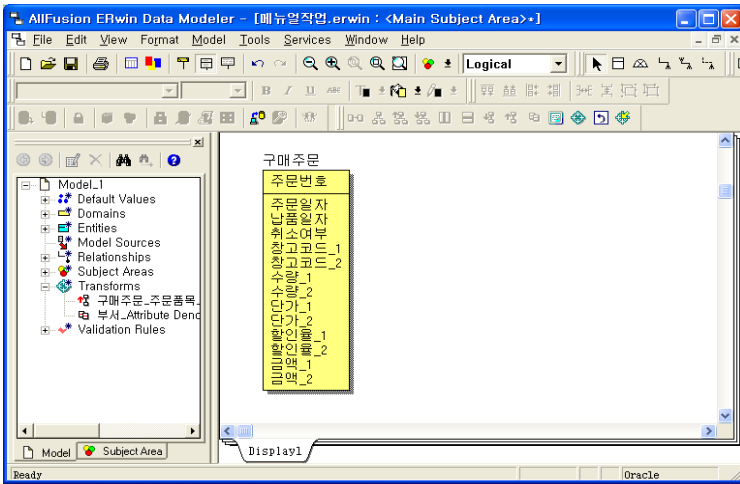
③ Transform의 이름과 Definition을 입력한다.



④ Target 테이블에 반복되는 레코드의 수를 입력한다.



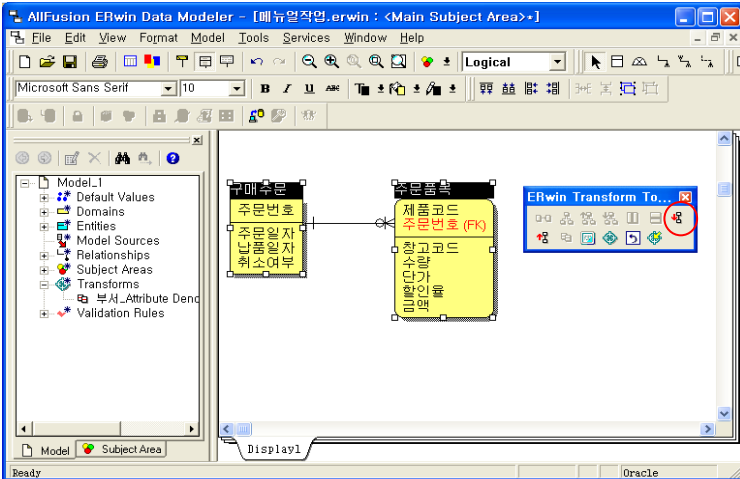
⑤ 변환 정보를 확인 할 수 있으며, 마침을 클릭해 작업을 종료한다.



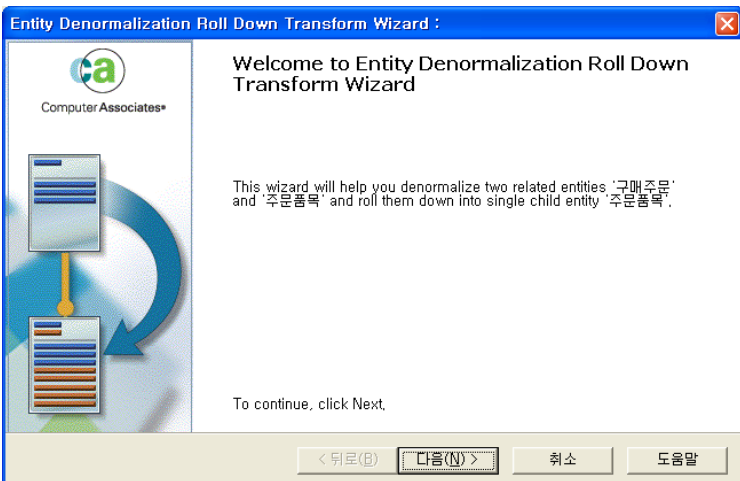
⑥ 주문품목의 컬럼들이 구매주문으로 포함된 것을 확인 할 수 있다. (부모 테이블로 이동)

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.

8. Forward / Reverse

8.1 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.

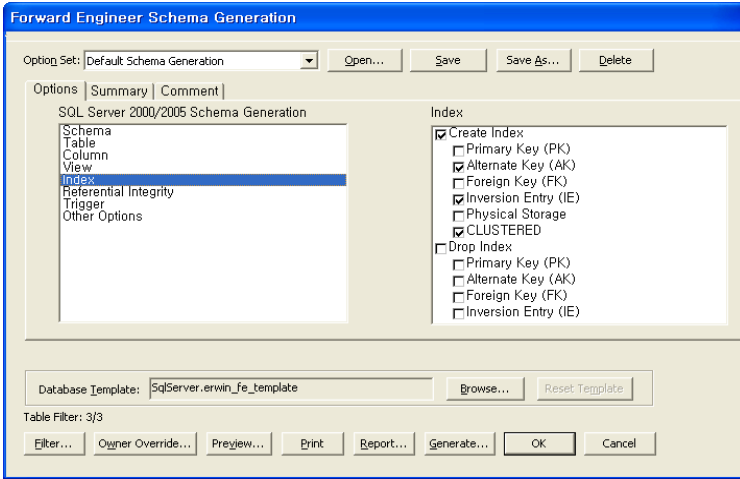
1) Schema Generation

Schema Generation 기능은 모델의 Object들을 사용자가 정의한 옵션에 따라 DDL 문장을 생성한다. DDL 문장은 모델내 입력된 정보와 사용자가 정의한 Generation Option에 의존하여 생성된다. 따라서 ERwin 에서 스키마를 생성할때는 모든 항목이 Option 이라는 사실을 주의해야 한다.

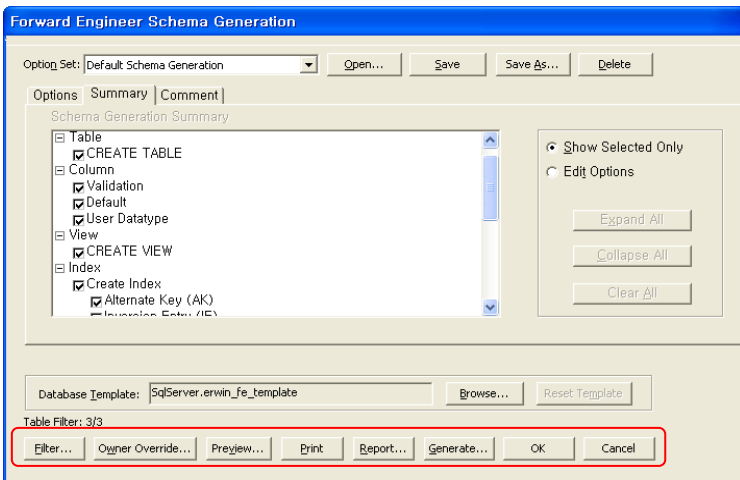
2) Generation Option 설명 (MS SQL)

항 목	기 능 설 명
■ Schema	모델 전체 즉 스키마 레벨의 옵션으로 사용자가 정의한 Rule, Default Value, 스크립트와 프로시저 생성에 관한 옵션
■ View	View table 생성에 관한 옵션
■ Table	Table 레벨의 Validation, DBMS가 지원하는 Physical Storage와 Table 에 대한 스크립트와 프로시저 생성 옵션이다
■ 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.
■ 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.
■ 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.
■ Trigger	관계선에 의한 트리거와 사용자가 정의한 트리거를 생성한다. 단 RI에서 부여한 FK 제한 조건이 트리거 보다 우선한다.
■ 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.

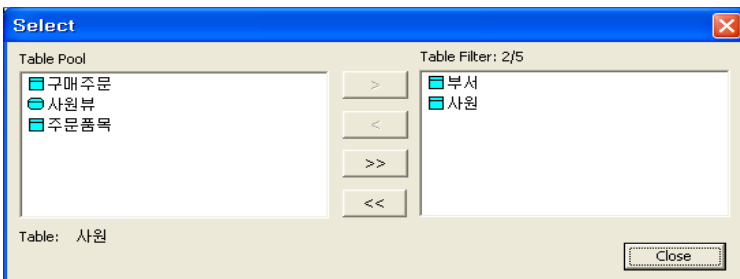


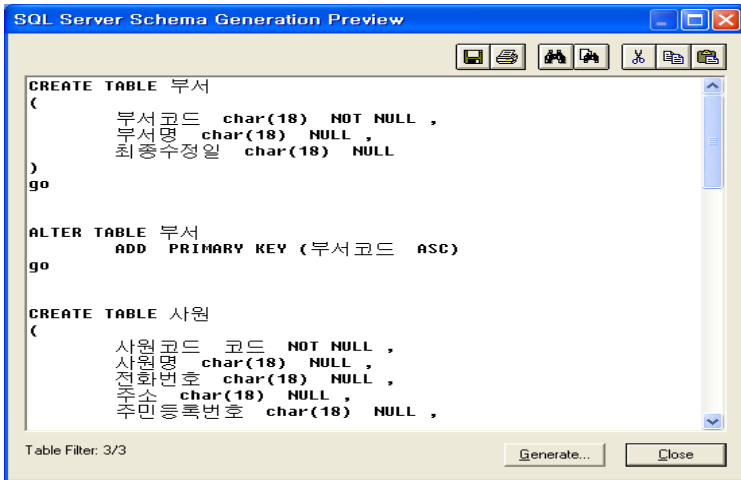
* 버튼설명 *

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

② Generation 옵션을 선택하는데, Target DBMS별로 다양한 옵션을 제공하게 된다.

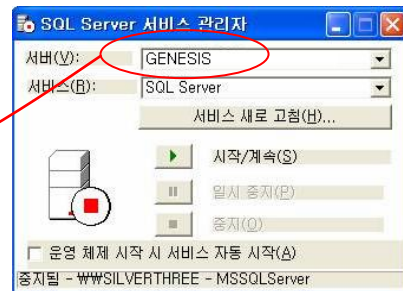
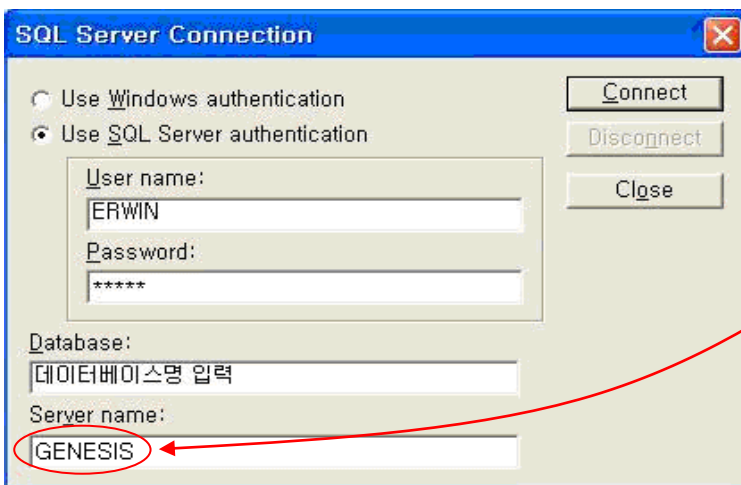
Summary 탭에서 옵션 설정 내용을 확인 할 수 있다. 이렇게 설정한 옵션은 Save를 통해서 Option Set으로 저장 할 수 있고, 이 Option Set 다른 모델에서 재사용이 가능하다. 또한 특정 테이블 대해서 DDL 문장을 생성하고자 할 때는 Filter 기능을 사용한다.





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 Failure가 디폴트로 설정되어 있다. 이때 나타나는 에러 메시지는 ERwin 에러가 아니고 해당 DBMS가 보여주는 에러이다.

8.2 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.

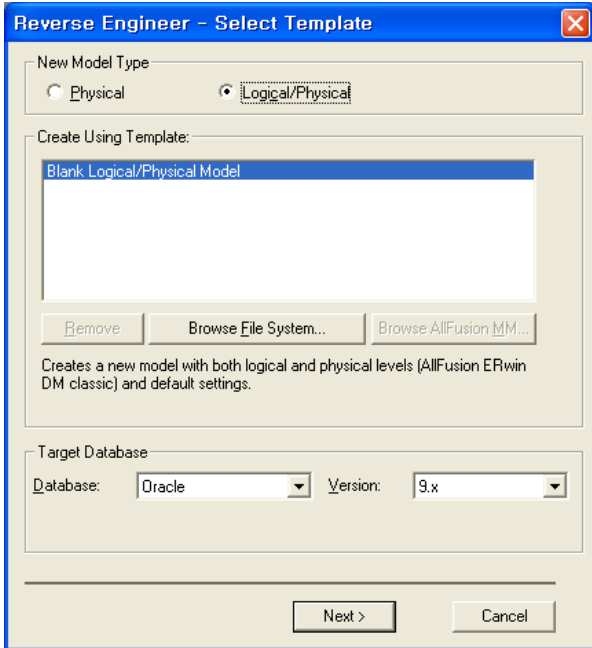
Reverse 대상은 Target DB 혹은 SQL DDL 스크립트 파일이다. 이러한 Database 정보는 Table, Column, Relationship, Triggers, Index, Stored Procedure 등을 포함한다.

1) Reverse Engineer Option 설명

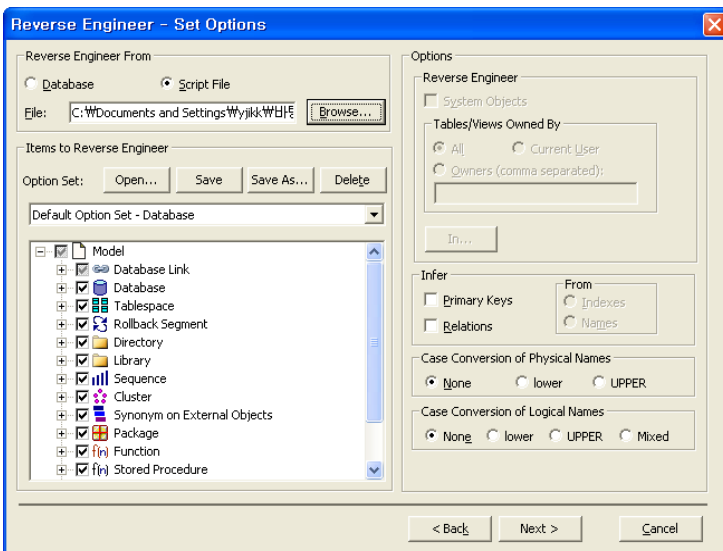
항 목	기 능 설 명
<input type="checkbox"/> Database	DBMS로부터 Reverse 한다.
<input type="checkbox"/> Script File	Script File로부터 Reverse 한다.
<input type="checkbox"/> System Object	Reverse 대상을 전체 시스템 객체로 설정한다. 이 옵션을 체크하면 사용자가 오너인 Table 뿐만 아니라 전체 시스템 Object를 가져온다.
<input type="checkbox"/> Table/Views Owned By	[All] 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.



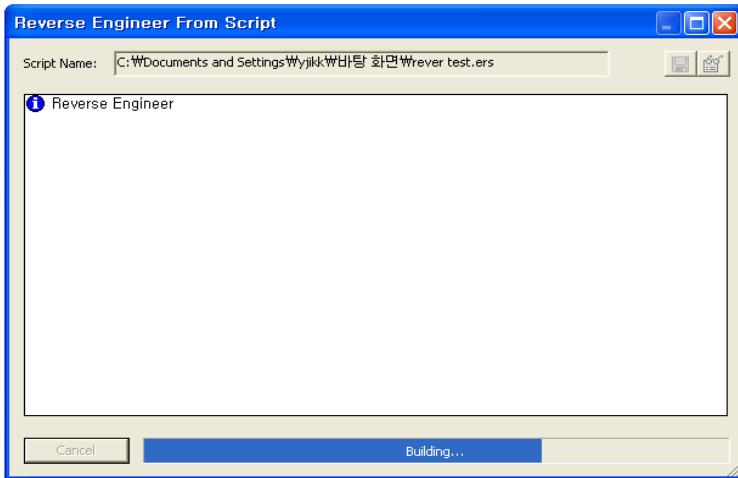
- ① Tools > Reverse Engineer를 선택한다.
- ② 생성할 모델을 Physical Only, Logical/Physical 복합 모델로 할지 선택한다.
- ③ Database 정보를 가져올 Target Database를 선택한다.



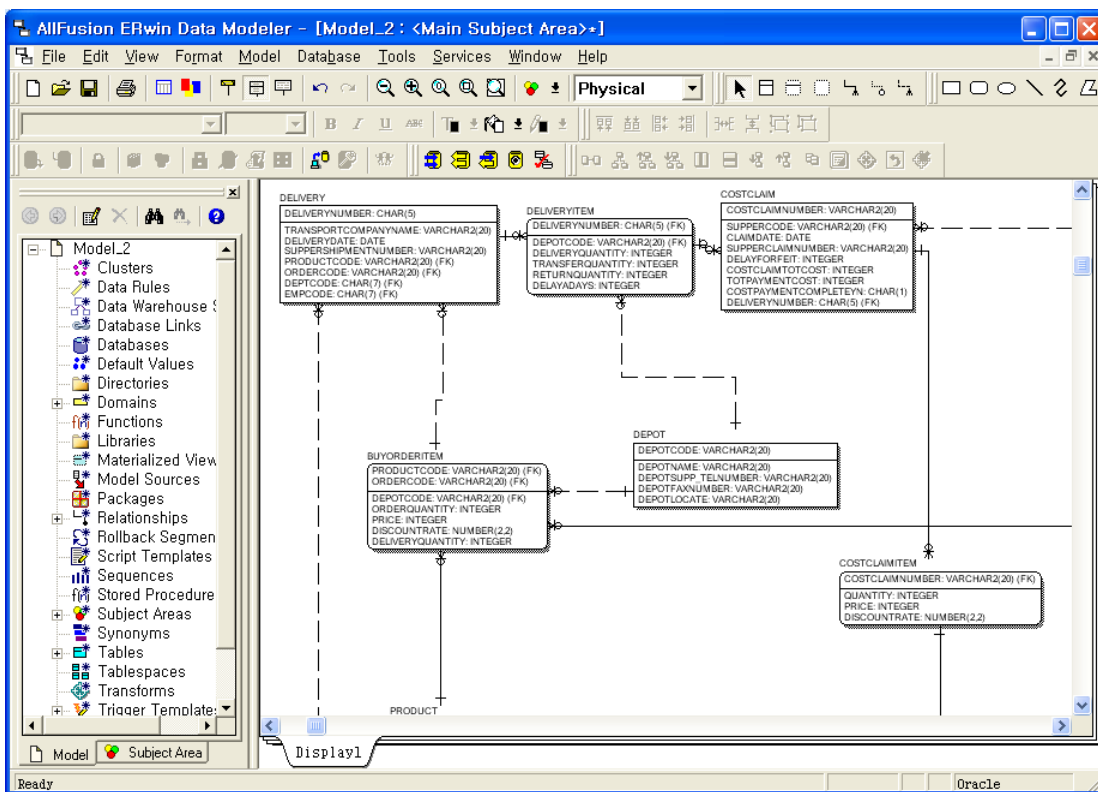
* 옵션 설명 *

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 Object를 리버스 한다.
Infer	Index와 Name을 기준으로 기본키와 관계를 추론한다.
lower	소문자로 리버스 한다.
Upper	대문자로 리버스 한다.
Mixed	대 / 소문자 혼합

- ④ Reverse 대상을 Script File을 선택하고 Browser를 통해 경로를 입력한다.
- 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.

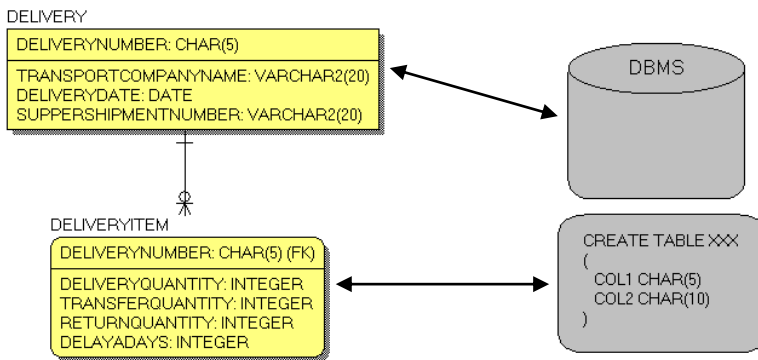
9. Complete Compare

9.1 Complete Compare

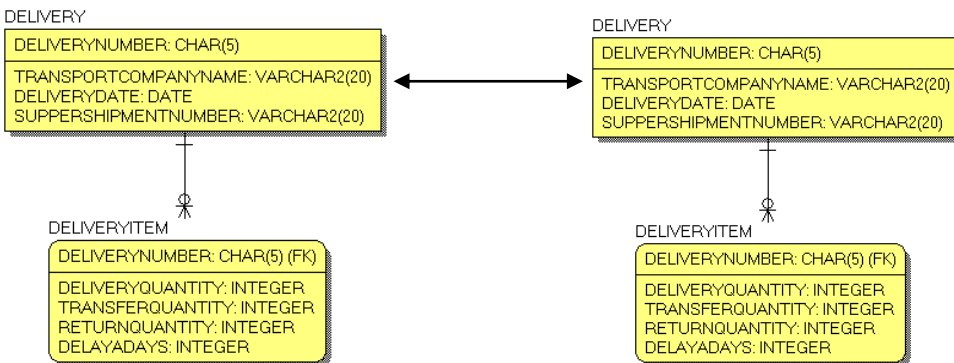
Complete Compare는 모델간의 비교 및 DBMS, DDL Script와의 비교를 통해 현재의 모델과 Database를 관리 하는데 매우 강력한 기능을 제공한다. R7에서는 4.1 제품의 Complete Compare 기능에서 한 단계 업그레이드 되었다. 모델의 세밀한 정보 까지도 비교가 가능하다. 예로 예전 제품에서는 인덱스의 Sort Type (Ascending, Descending)의 비교가 되지 않았다. 모델과 실 스카마의 Sort Type이 다름에도 이러한 부분을 찾아주지 못 하였다. 하지만 R7 에서는 이러한 부분까지 세밀하게 비교가 가능하다. 사용자가 정의한 비교 옵션에 따라 각 항목을 비교하여 차이점을 그리픽하게 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



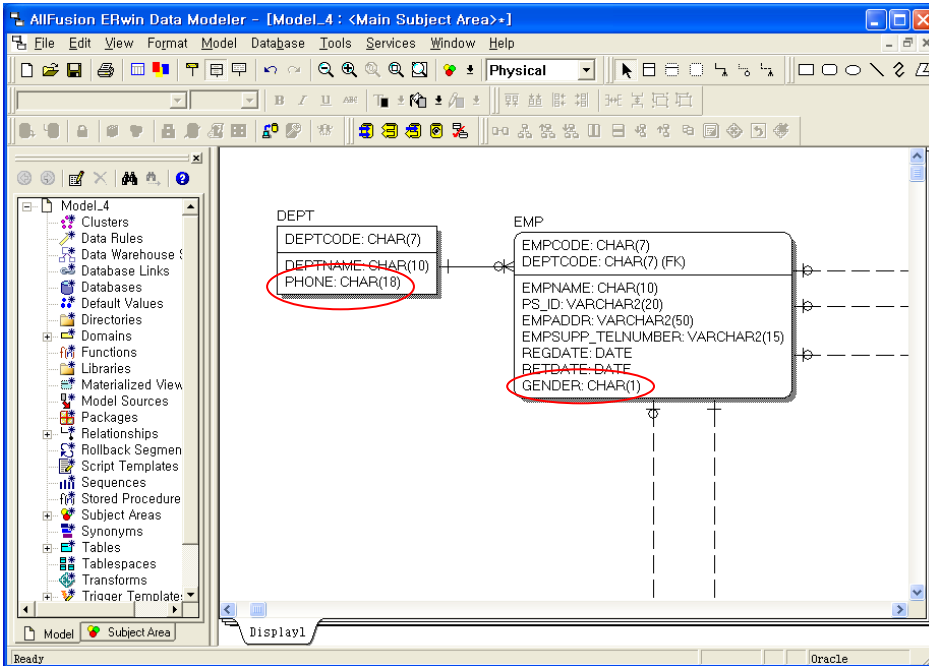
2) Logical / Physical Model ↔ Logical / Physical Model



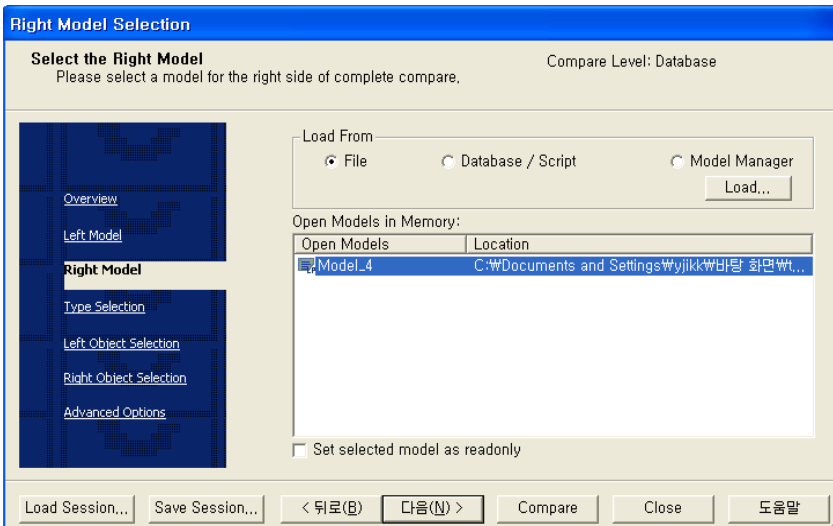
Complete Compare 기능은 Database에 이미 데이터가 있는 경우도 Table 구조 변경이 가능하다. 이 기능은 내부적으로 Table을 갱신할 때 Temp Table을 생성하고 데이터를 백업한 다음 원래의 Table 갱신한 후에 데이터를 로딩하는 과정으로 진행된다.

3) Complete Compare의 실행

Physical 모델과 DDL Script File 간의 비교를 해보자.

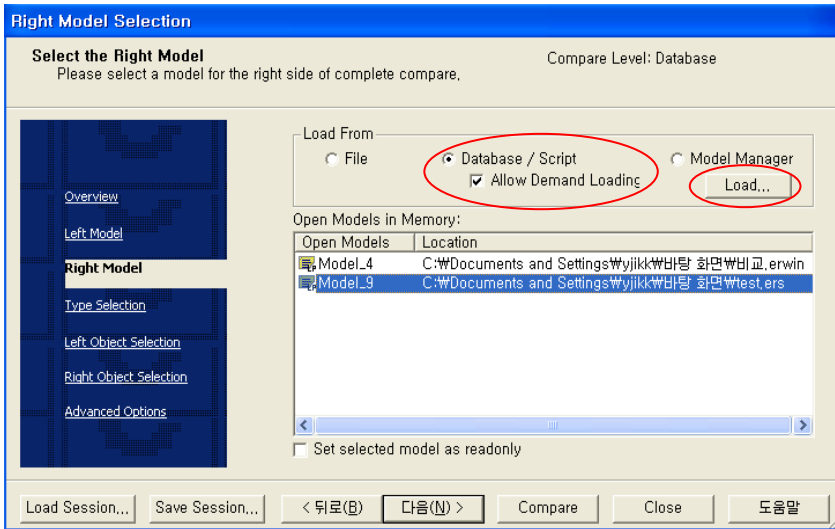


① DEPT와 EMP 테이블을 수정하였다. (PHONE 컬럼 추가, GENDER의 타입 수정)

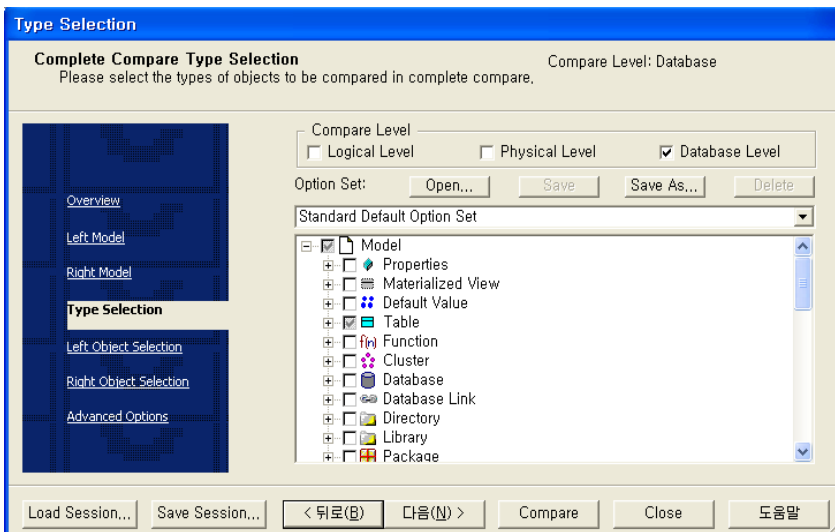


② Tools > Complete Compare를 선택한다.

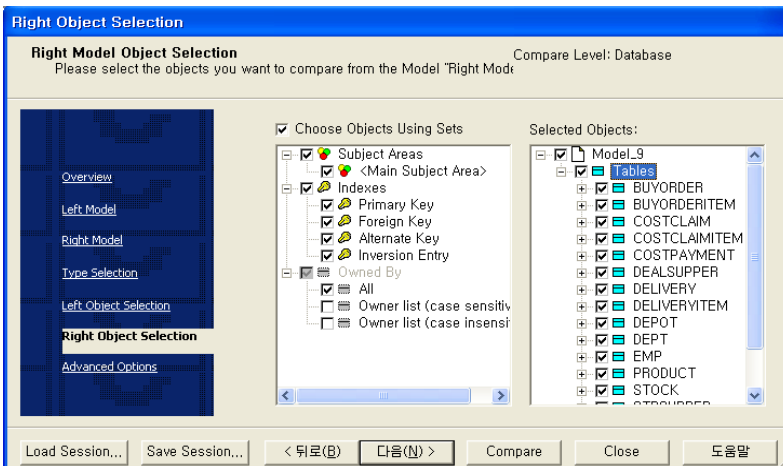
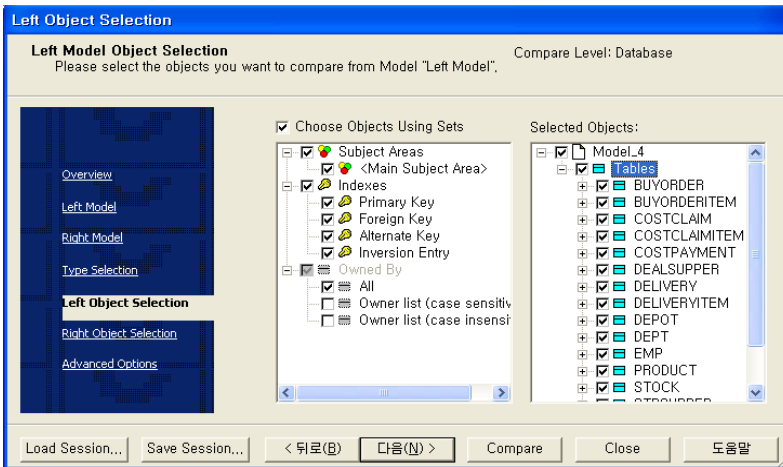
③ Right Model 선택 단계부터 실행되고, Left는 기본적으로 현재 열려 있는 모델정보를 가져온다.



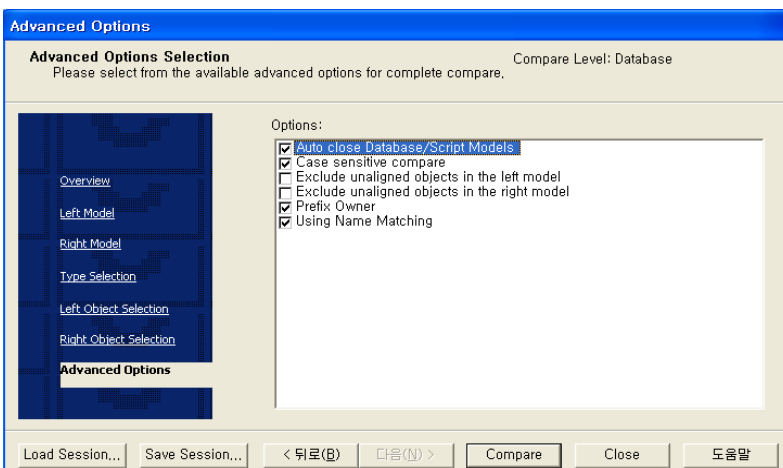
- ④ Right Model 선택 단계에서 비교 대상을 Database / Script로 선택한다.
- ⑤ Load를 통해 DDL Script File을 읽어 들인다. 다음을 선택한다.



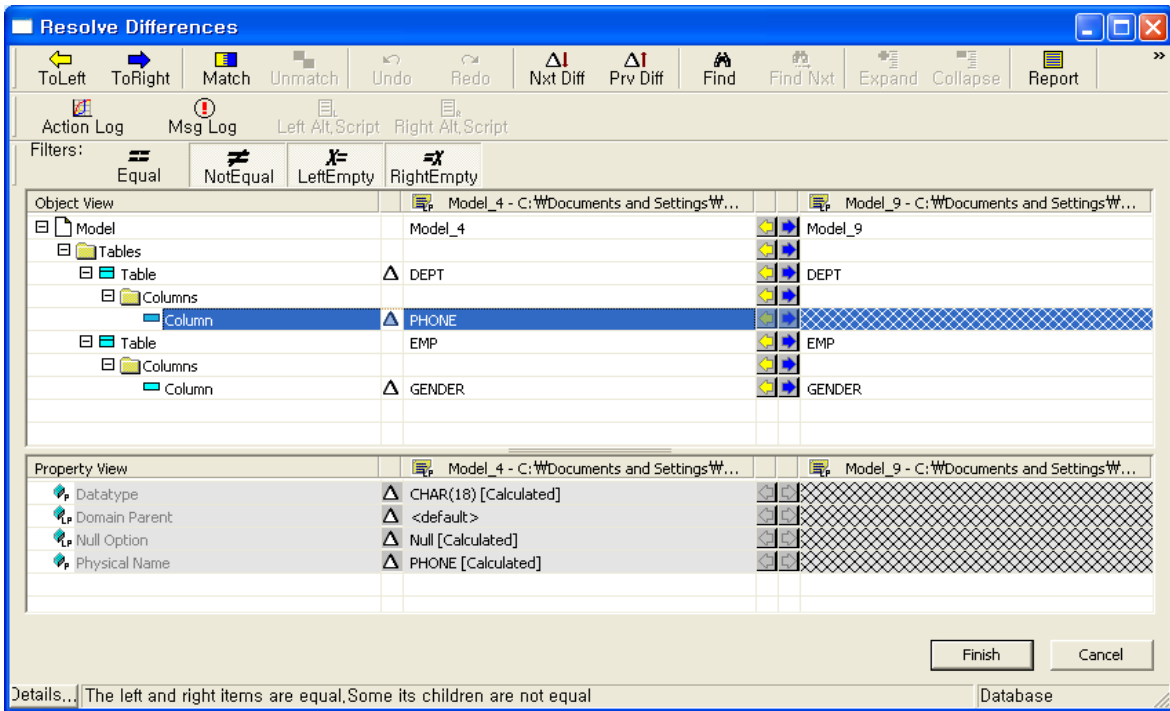
- ⑥ 비교 옵션을 Table로 선택한다.
- 옵션 셋을 저장할 수 있으며 재사용이 가능 하며, 비교 항목에 따라 사용자가 다양한 옵션을 선택 할 수 있다. 현재 모델은 Table만 선택해도 변경 사항에 대한 비교가 가능하다.



⑦ Left / Right Model을 Filter 할 수 있으나, Default로 하고 다음을 선택한다.




⑨ Compare을 선택한다.



그림과 같이 모델과 DDL Script 파일을 비교해 차이점을 찾아준다. 상위 그리드에서 상이한 Object List를 보여주며, 해당 Object를 선택하면 하위 그리드에서 자세한 차이점을 확인 할 수 있다. 또한 사용자는 다양한 비교 옵션을 통해 차이점을 분석하고, 차이점을 모델과 DBMS에 반영할 수 있다.

* 옵션 설명

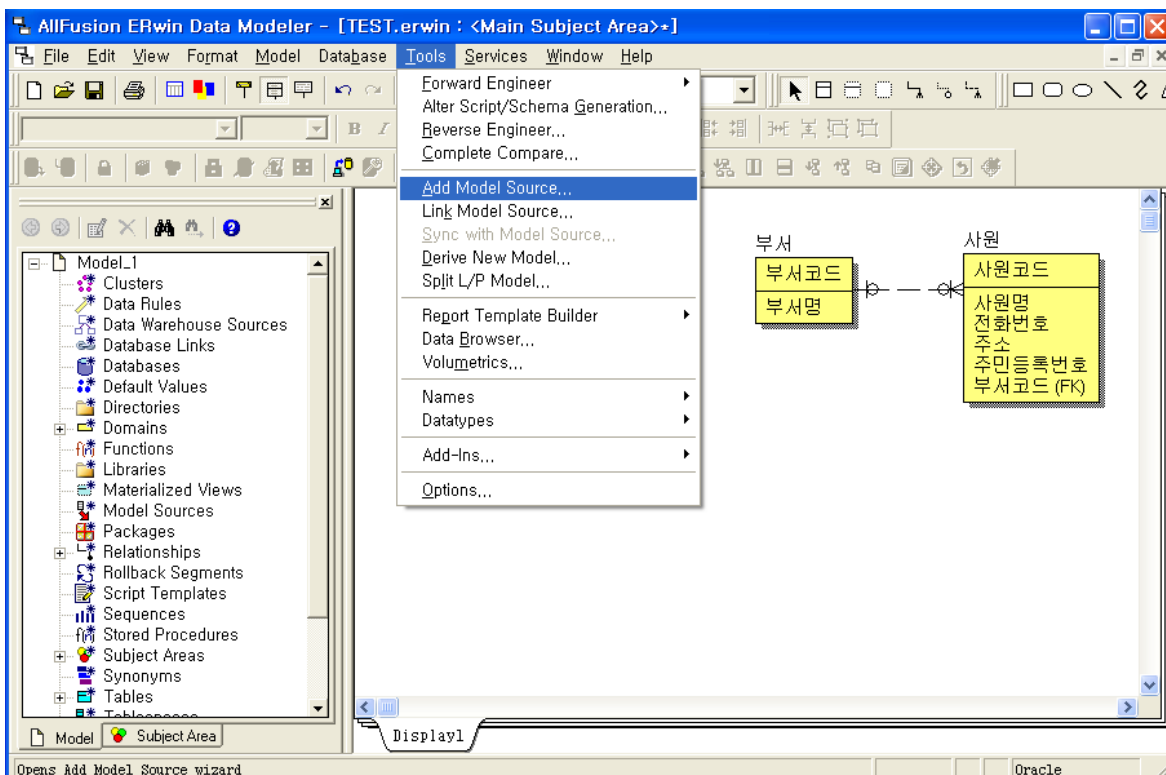
항 목	기 능 설 명
 Equal	모델간 같은 항목을 보여준다.
 NotEqual	비교 모델간 서로 같지 않은 항목을 보여준다.
 LeftEmpty	Left 모델에 없는 항목을 보여준다.
 RightEmpty	Right 모델에 없는 항목을 보여준다.
 ToLeft	Right 모델의 내용을 Left 모델로 반영한다.
 ToRight	Left 모델의 내용을 Right 모델로 반영한다.
 Match	모델간의 Object가 같으나 간혹 매치가 되지 않은 경우가 발생한다. 이런 경우 Match를 이용해 일치 시켜준다.

10. Add Model Source / Sync With Model Source

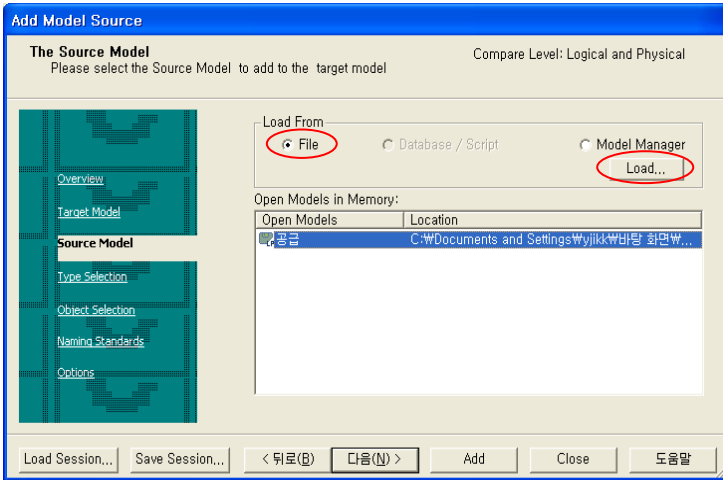
10.1 Add Model Source

Add Model Source는 현 모델에 다른 모델의 정보를 Add 할 수 있도록 지원하는 기능이다. 예를 들어 Source 모델의 Entity들을 현 모델에 Add 할 수 있다. 또한 Entity 뿐만 아니라 Subject Area, Domain등 모델의 다양한 정보를 Add 할 수 있도록 지원한다.

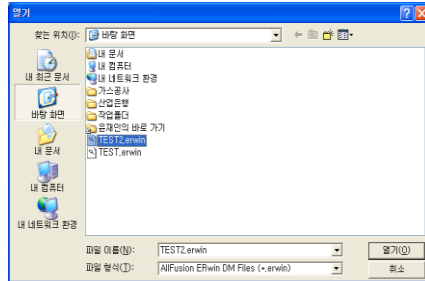
1) Add Model Source의 실행



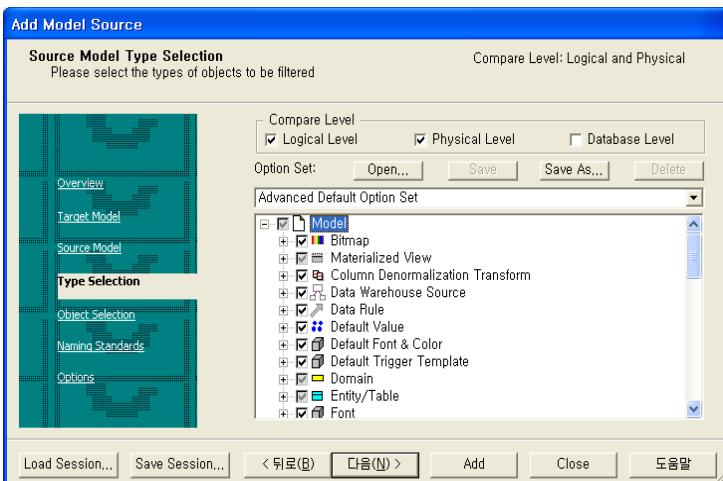
- ① Tools > Add Model Source..를 선택한다.



② Load From에서 File 선택 후 Load 버튼을 클릭한다.

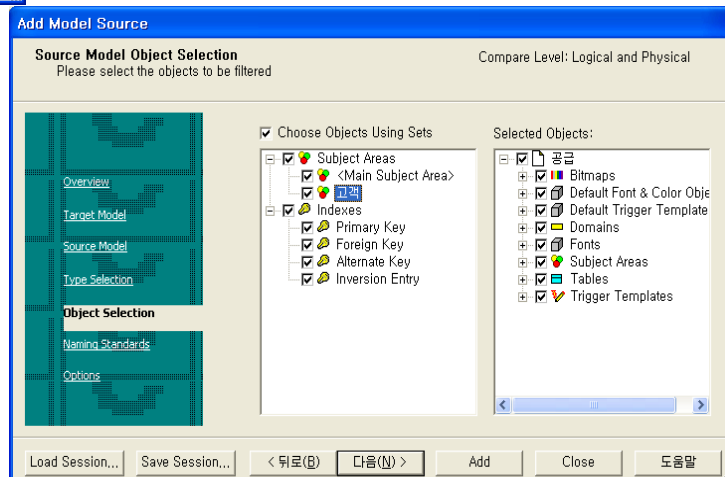


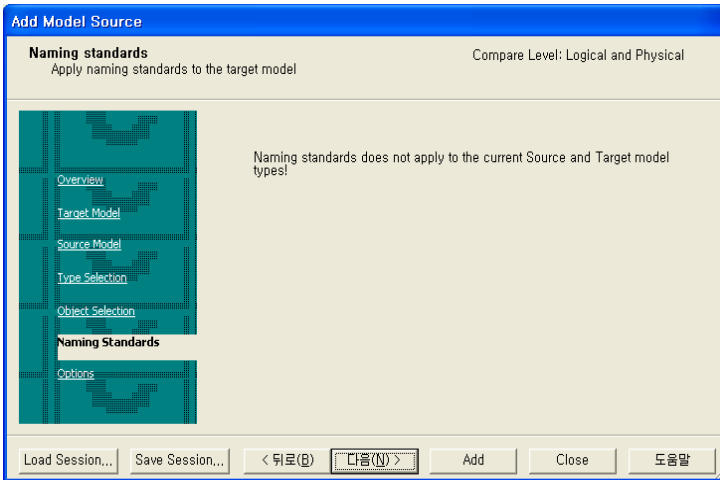
③ Source Model을 선택하고 다음 버튼을 클릭한다.



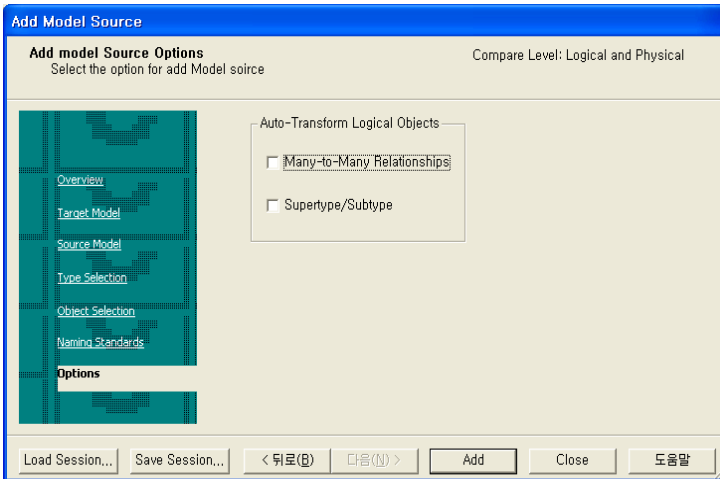
④ Source Model에서 추출 할 항목을 선택한다. Subject Area, Entity, Domain, Default 등 모델의 다양한 정보를 추출 할 수 있다. 사용자는 추출 옵션을 저장 할 수 있으며 재 사용 할 수 있다.

⑤ Source Model의 추출 항목을 필터링(Filtering) 할 수 있다.



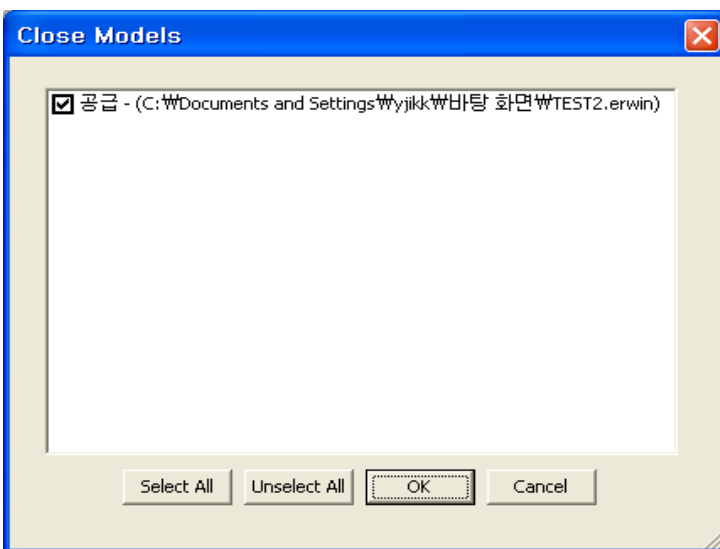
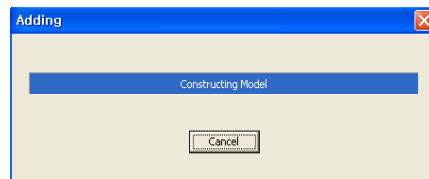


⑥ 다음 버튼을 클릭한다.

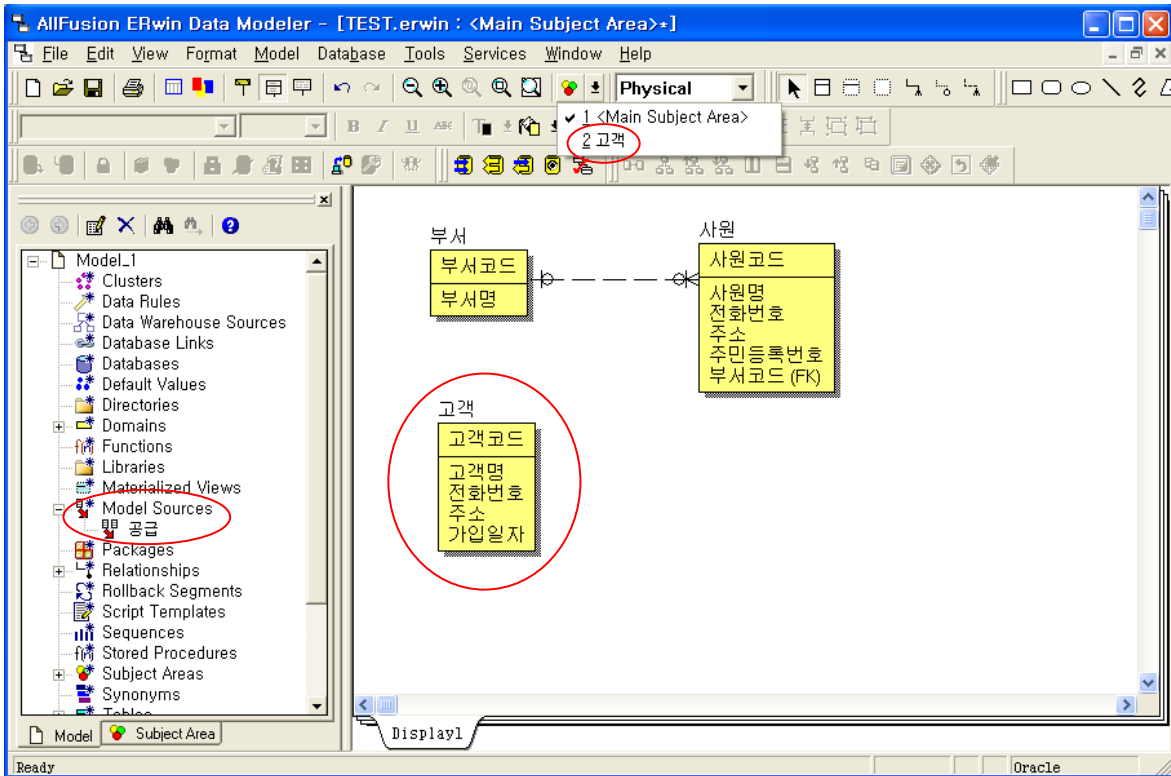


⑦ M : M 관계와 일반화 관계에 대해서 자동으로 변환 하도록 하는 옵션을 설정하는 화면이다.

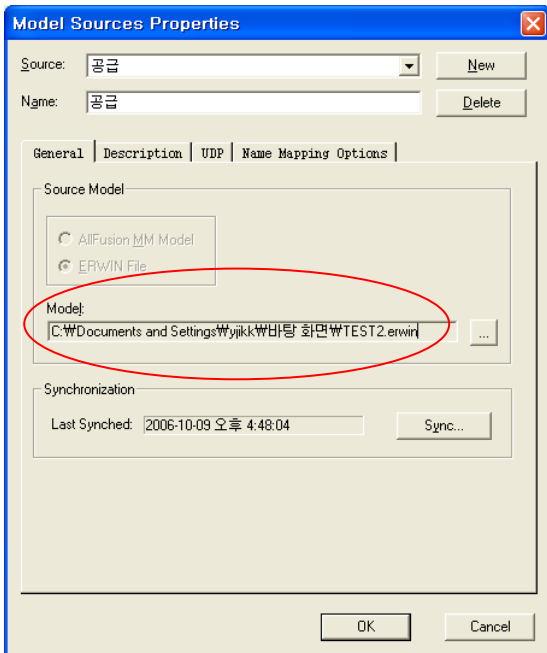
⑧ Add 버튼을 클릭한다.



⑨ OK 버튼을 클릭한다.



⑩ 현 Model로 Source Model의 정보들이(Subject Area, Entity) Add 된 것을 확인 할 수 있으며 Model Explorer > Model Sources에서 Source Model의 정보를 확인 할 수 있다.

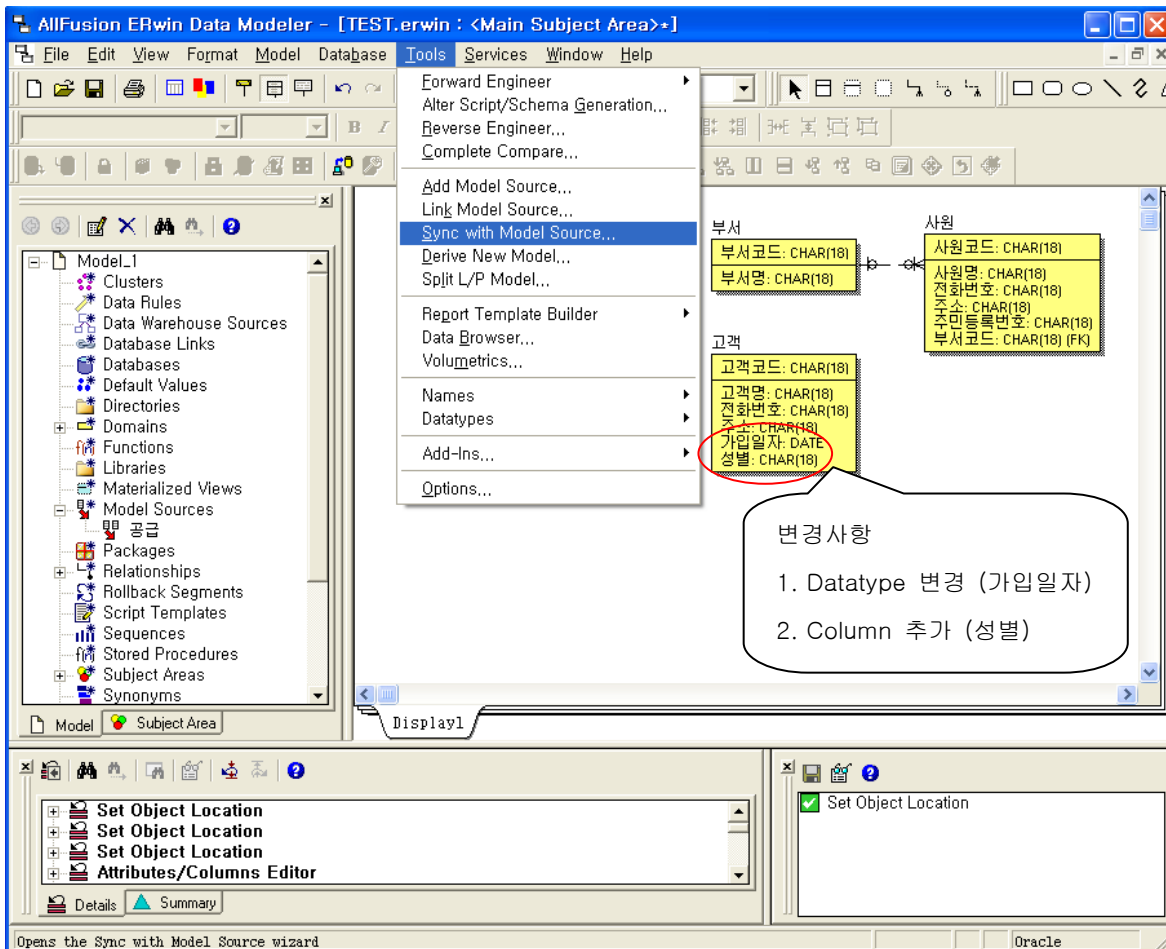


10.2 Sync With Model Source

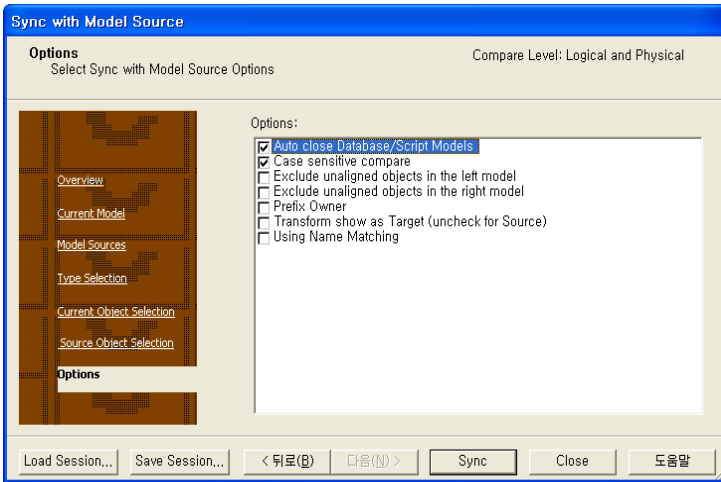
Sync With Model Source는 현 모델과 Add Model Source, Link Model Source 기능으로 연결된 Source Model과의 비교를 통해 차이점을 찾아주고 양방향으로 동기화 할 수 있도록 지원한다.

1) Sync With Model Source 실행

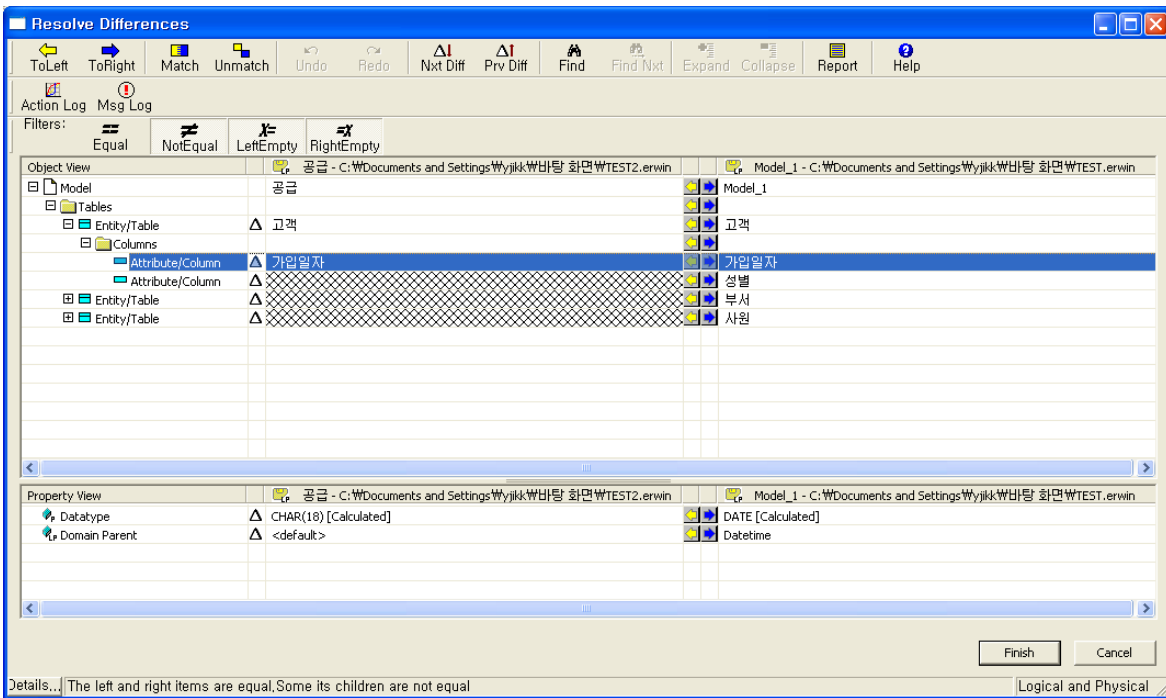
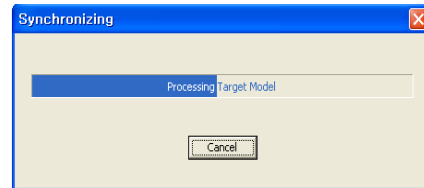
현 모델을 수정하고 Source Model과의 비교를 통해 동기화 작업을 해보자.



① Tools > Sync With Model Source를 선택한다.

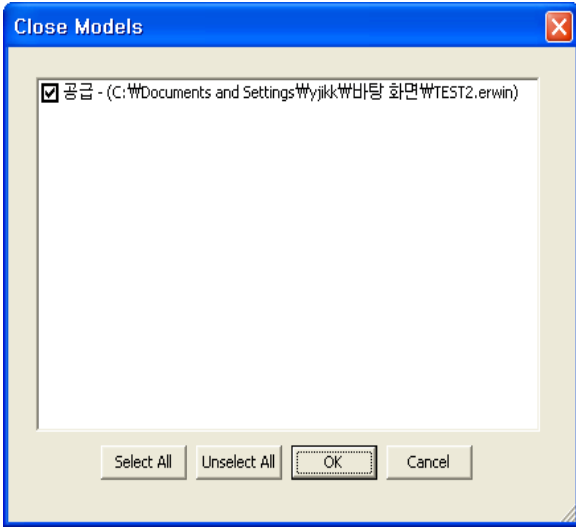


⑤ Sync 버튼을 클릭한다.



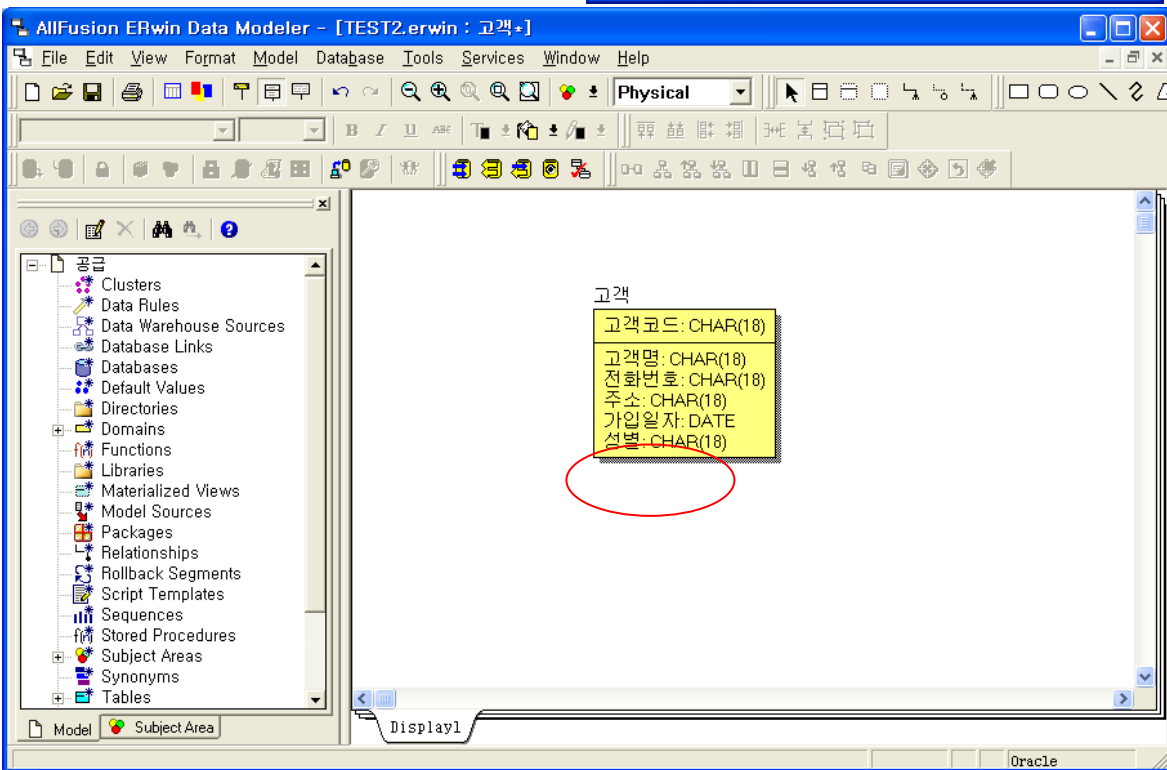
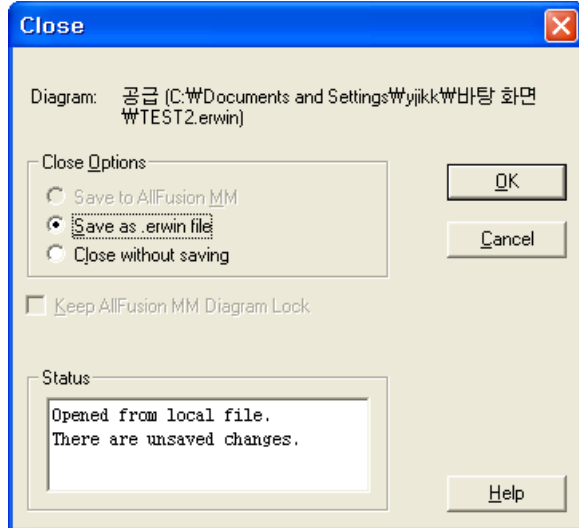
⑥ Finish 버튼을 클릭한다.

(그림과 같이 Current Model과 Source Model을 비교해 차이점을 찾아준다. 상위 그리드에서 상이한 Object List를 보여주며, 해당 Object를 선택하면 하위 그리드에서 자세한 차이점을 확인 할 수 있다. 또한 사용자는 다양한 비교 옵션을 통해 차이점을 분석하고, 차이점을 각각의 모델에 반영 할 수 있다.)



⑦ OK 버튼을 클릭한다.

⑧ Save as .erwin file 선택 후 OK를 클릭한다.



⑨ Source Model을 열어보면 Current Model의 변경사항이 반영된 것을 확인 할 수 있다.

11. Reporting

ERwin Data Modeler는 모델의 Object들의 정보를 다양한 형태로 출력 할 수 있도록 지원한다. ERwin에서는 Data Browser, Report Template Builder, 그리고 Add Ins.. 기능과 같이 3가지 형태의 Reporting을 지원한다.



11.1 Data Browser

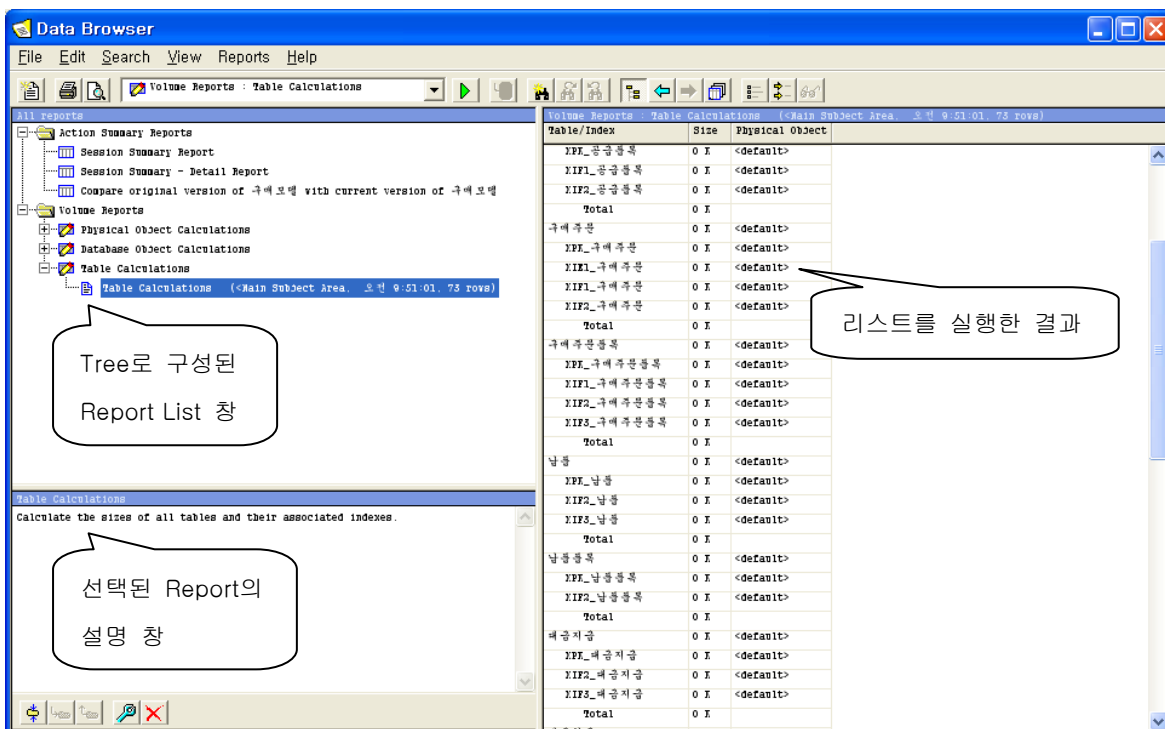
Data Browser는 사용자가 모델 내의 Entity와 Relationship의 항목 분석, 물리설계와 구현에 대한 모델의 정확성을 검증하는 데 이용할 수 있으며, 모델의 Object들을 직접 수정할 수 있도록 하는 기능도 지원한다.

이 기능은 모델이 가지고 있는 다양한 정보를 대상으로 모든 레포트를 생성 할 수 있으며, 다른 사용자와의 공유가 가능하다. 또한 ERwin Data Modeler외의 다른 어플리케이션이나 파일 포맷으로 Export할 수 있다. 정의된 레포트 양식은 .erp 파일로 저장할 수 있으며 재사용이 가능하다.

지원하는 파일 포맷은 DDE Application, CSV Files, HTML Pages, RPTwin 등이 있다.

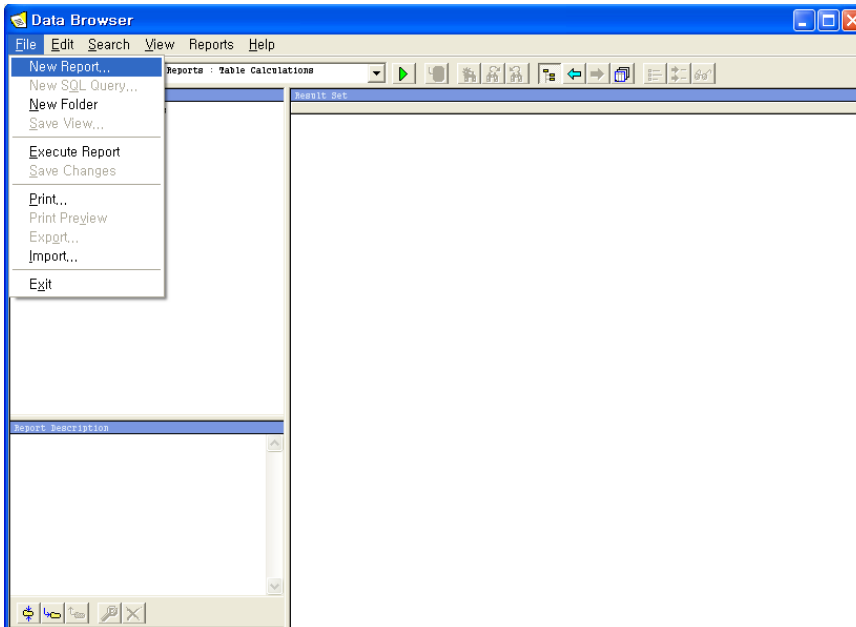
1) Data Browser의 사용

Data Browser를 사용 하려면 Tools > Data Browser 메뉴를 선택하거나  아이콘을 선택하면 3개의 창으로 나뉘어진 화면이 나타난다. 기본적으로 ERwin에서 제공하는 Report List를 통해 보고서를 출력할 수 있다. 실행방법은 리스트를 더블클릭 하거나  을 선택하면 된다.

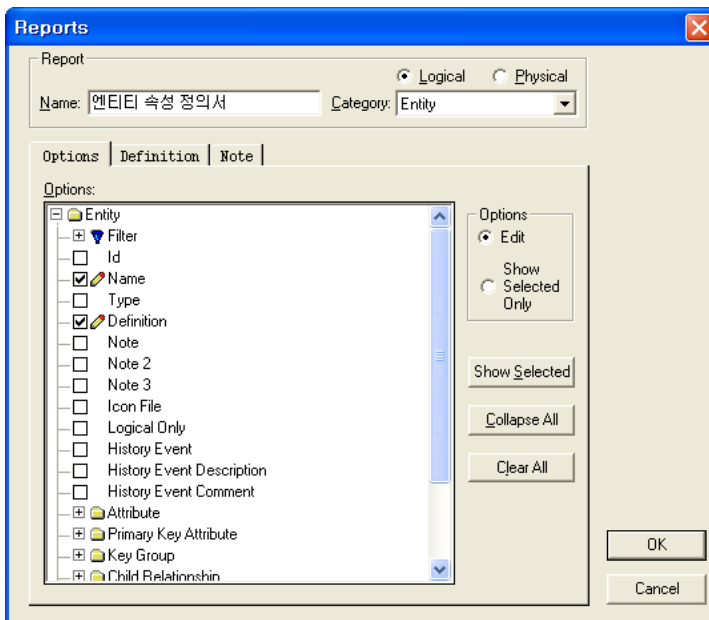


2) 새로운 레포트의 생성

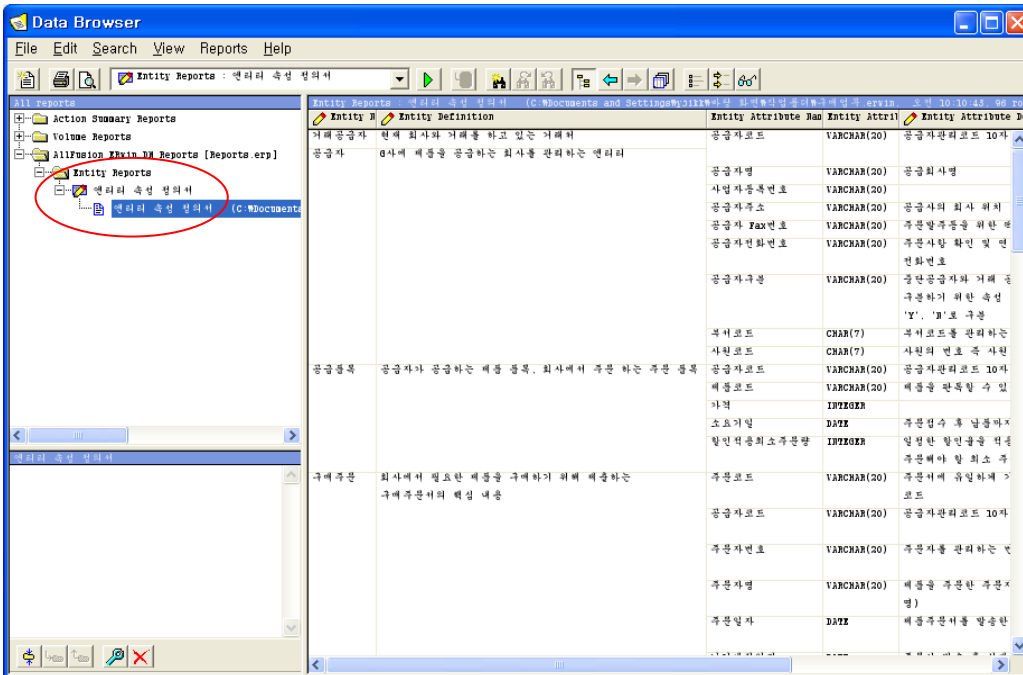
Data Browser를 사용 하여 새로운 레포트를 생성해보자. 다음은 Entity의 이름과 정의, 그리고 각 Entity가 가지고 있는 속성의 이름과 정의를 포함한 레포트를 생성하는 예제이다.



① Data Browser를 실행하고 File > New Report를 선택한다.

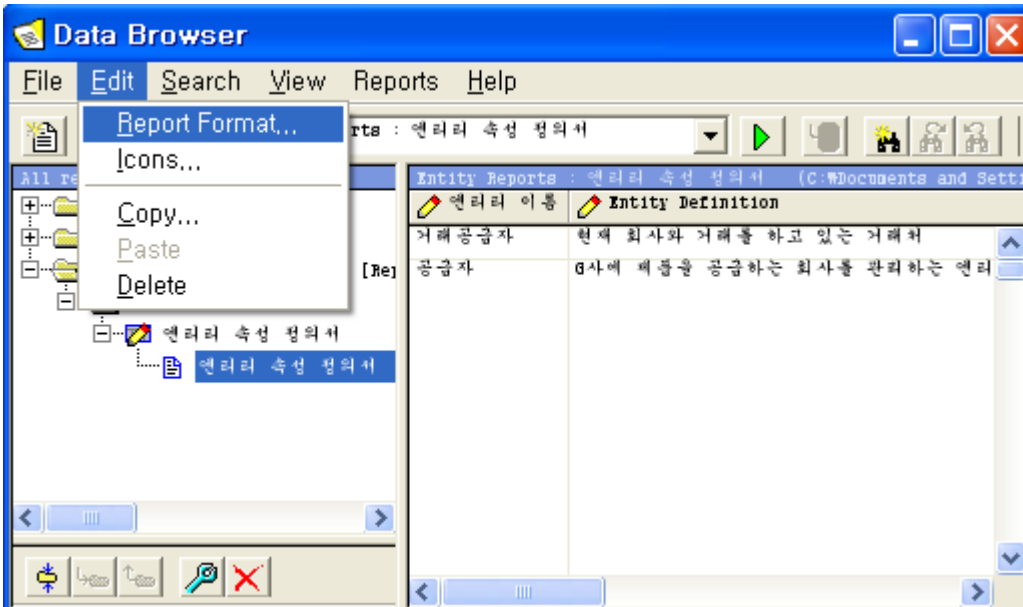


- ② Report Name을 입력한다.
- ③ Category를 선택한다.
- ④ Option 탭에서 옵션을 선택
- ⑤ OK 버튼을 누른다.

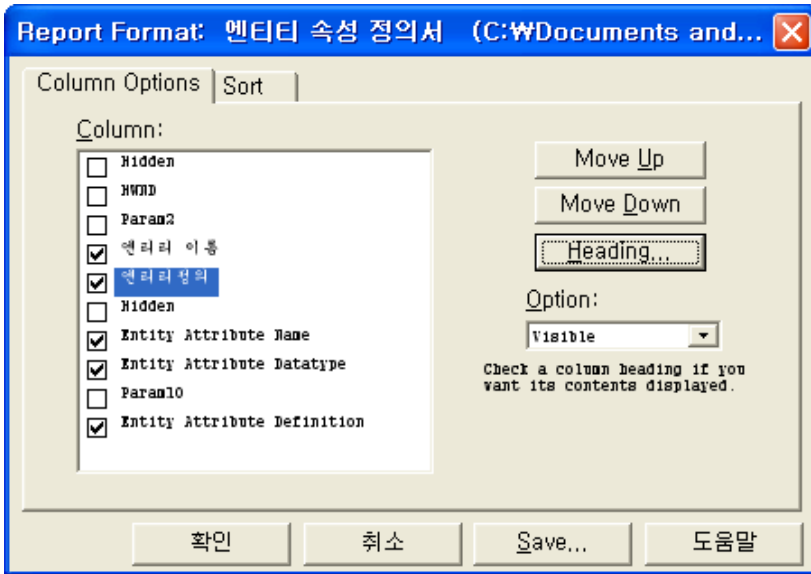


- ⑥ 좌측 트리에 생성한 레포트 항목이 리스트에 추가된다.
- ⑦ 리스트를 더블 클릭하면 위 그림과 같이 레포트가 생성된다.

3) 레포트 포맷의 수정



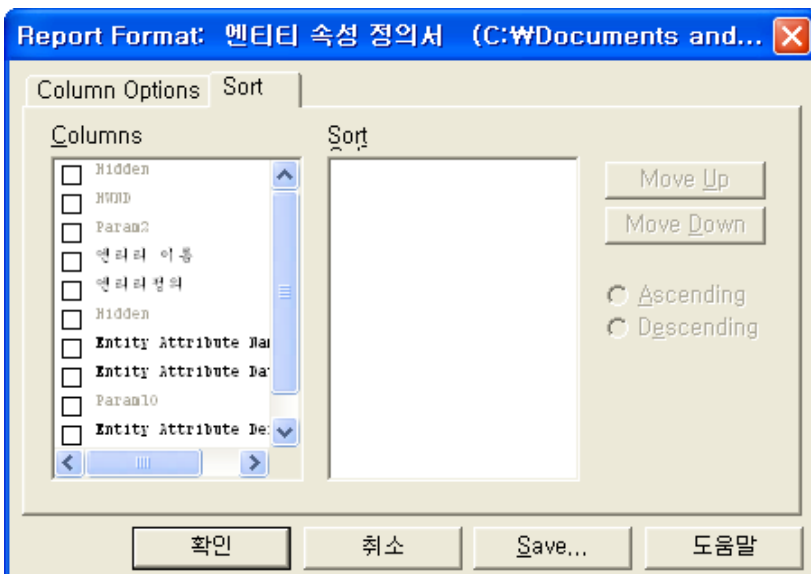
- ① Edit > Report Format을 선택한다.



② Column Options 탭에서 각 항목의 순서 변경시 Move Up / Down 버튼을 사용한다.



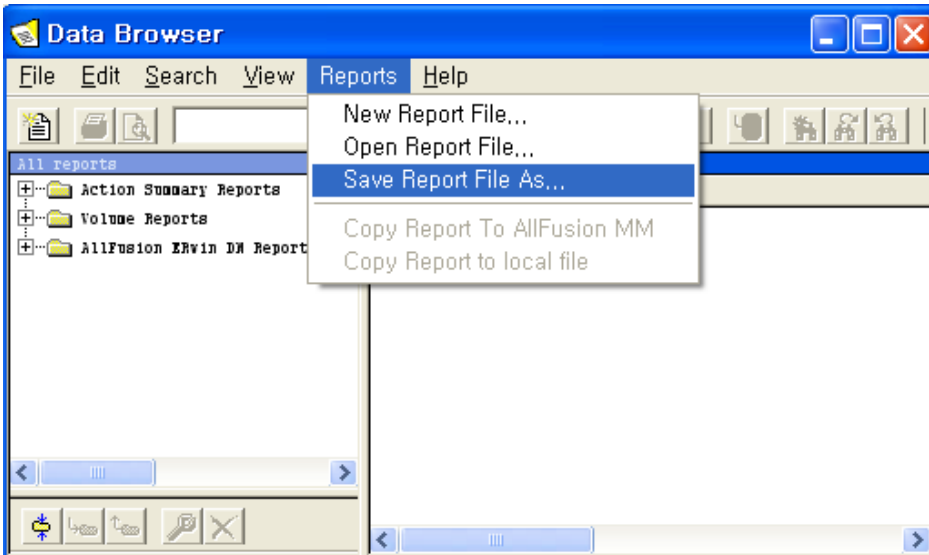
③ 각 항목의 명칭을 변경하고자 할 경우 Heading 버튼을 선택하고 바뀔 항목명을 입력한다.



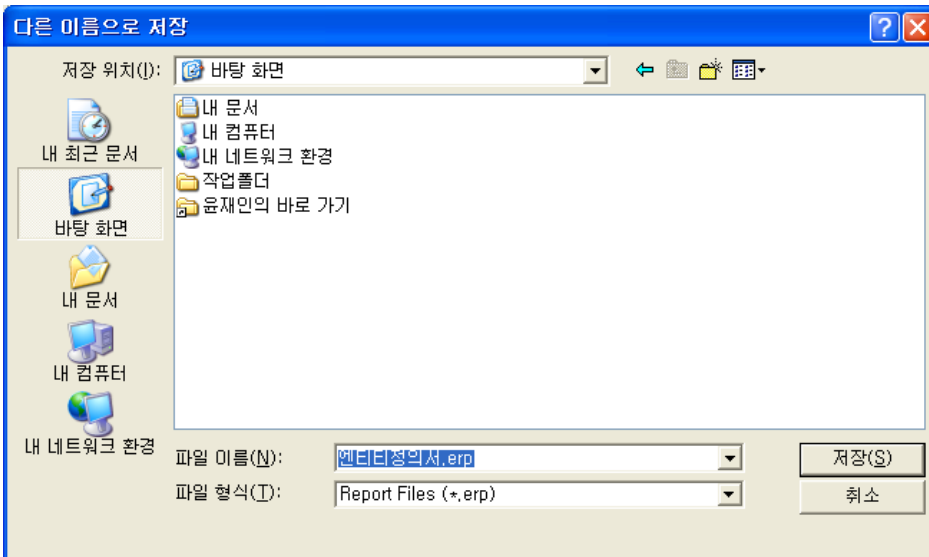
④ Sort 탭에서는 각 항목의 데이터를 정렬하고자 할 경우 사용한다.

3) 정의한 레포트 양식의 저장

사용자가 정의한 레포트 양식은 저장할 수 있으며(.erp) 여러 사용자와 공유가 가능하다.



① Reports > Save Report File as.. 선택한다.



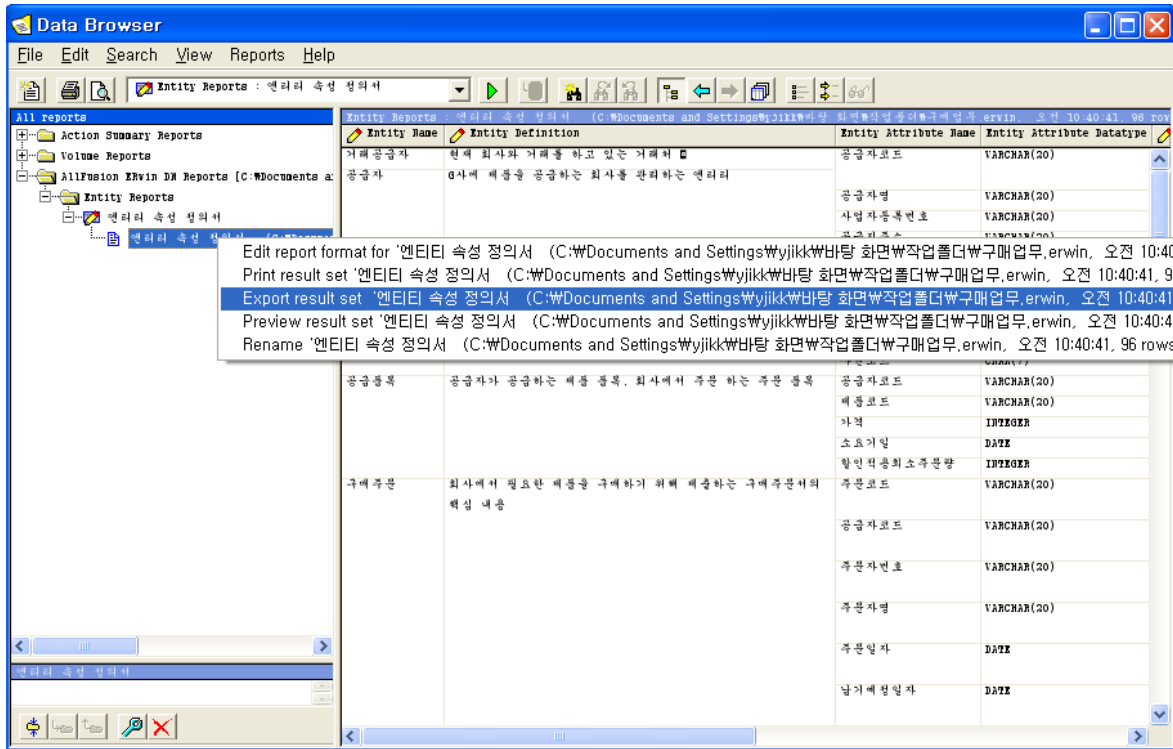
② 엔티티정의서.erp 파일을 저장한다.

Tip

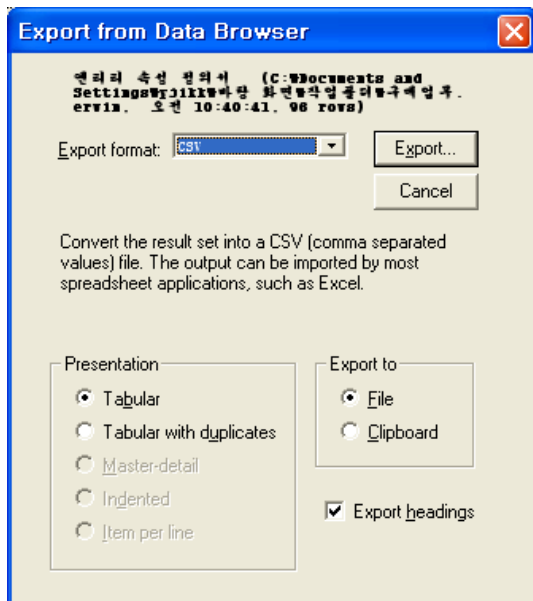
- 1) New Report File : 새로운 레포트를 생성한다.
- 2) Open Report File : 정의된 레포트 파일(.erp)을 연다.
- 2) Save Report File As : 정의한 레포트를 저장한다.

4) 레포트를 다른 포맷으로 내보내기

ERwin Data Modeler에서는 작성한 모델을 Data Browser를 통해서 포맷을 작성 할 뿐만 아니라, 또 다른 형태, 혹은 다른 어플리케이션으로 레포트를 변환 할 수 있도록 지원한다.



① 레포트를 선택 후 마우스 우측 버튼을 클릭해 Export result set을 선택한다.



② 원하는 Export Format을 선택한다.

③ Presentation은 동일한 항목을 어떻게 출력 할 것인가에 대한 옵션이다.

항 목	설 명
Tabular	중복되는 항목은 출력하지 않고 공백으로 출력한다.
Tabular with Duplicates	중복되는 항목 전부 출력한다.

④ Export to에서 File을 선택한다.

⑤ Export 버튼을 선택한다.

[CSV로 Export한 화면]

Entity Name	Entity Definition	Entity Attribute Name	Entity Attribute Datatype	Entity Attribute Definition
거래공급자	현재 회사와 거래를 하고 있는 거래공급자	공급자코드	VARCHAR(20)	공급자관리코드 10자리로 관리
공급자	G사에 제품을 공급하는 회사를 관리하는 엔티티	공급자명		공급회사명
		사업자등록번호		
		공급자주소		공급사의 회사 위치
		공급자 Fax번호		주문발주등을 위한 팩스 번호
		공급자전화번호		주문사항 확인 및 연락을 위한 전화번호
		공급자구분		중단공급자와 거래 공급자를 구분하기 위한 속성 'Y', 'N'로 구분
		부서코드	CHAR(7)	부서코드를 관리하는
		사원코드		사원의 번호 즉 사원
공급품목	공급자가 공급하는 제품 품목, 회	공급자코드	VARCHAR(20)	공급자관리코드 10
		제품코드		제품을 판독할 수 있
		가격	INTEGER	
		소요기일	DATE	주문접수 후 납품까지
		할인적용최소주문량	INTEGER	일정한 할인율을 적용
구매주문	회사에서 필요한 제품을 구매하기	주문코드	VARCHAR(20)	주문서에 유일하게 사용되는
		공급자코드		공급자관리코드 10
		주문자번호		주문자를 관리하는

[HTML로 Export한 화면]

엔티티 속성 정의서

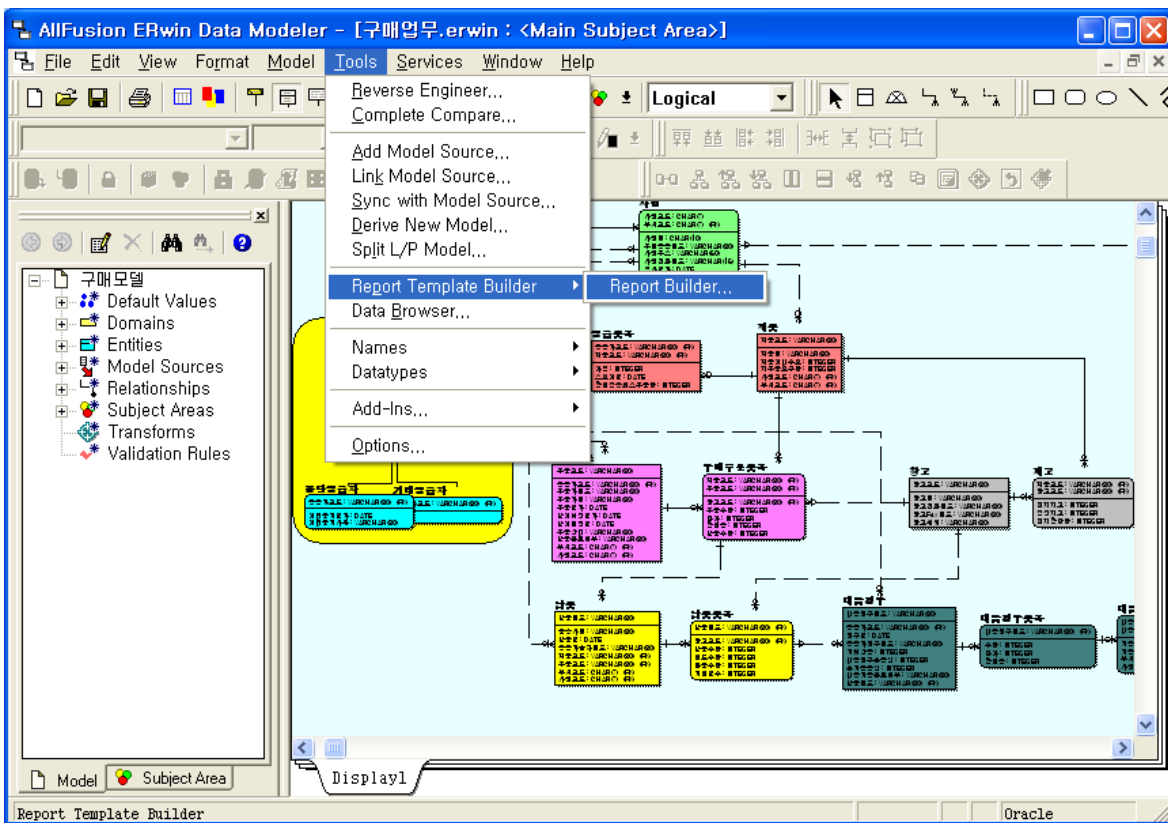
Entity Name	Entity Definition	Entity Attribute Name	Entity Attribute Datatype	Entity Attribute Definition
거래공급자	현재 회사와 거래를 하고 있는 거래공급자	공급자코드	VARCHAR(20)	공급자관리코드 10자리로 관리
공급자	G사에 제품을 공급하는 회사를 관리하는 엔티티	공급자명		공급회사명
		사업자등록번호		
		공급자주소		공급사의 회사 위치
		공급자 Fax번호		주문발주등을 위한 팩스 번호
		공급자전화번호		주문사항 확인 및 연락을 위한 전화번호
		공급자구분		중단공급자와 거래 공급자를 구분하기 위한 속성 'Y', 'N'로 구분

11.2 Report Template Builder (RTB)

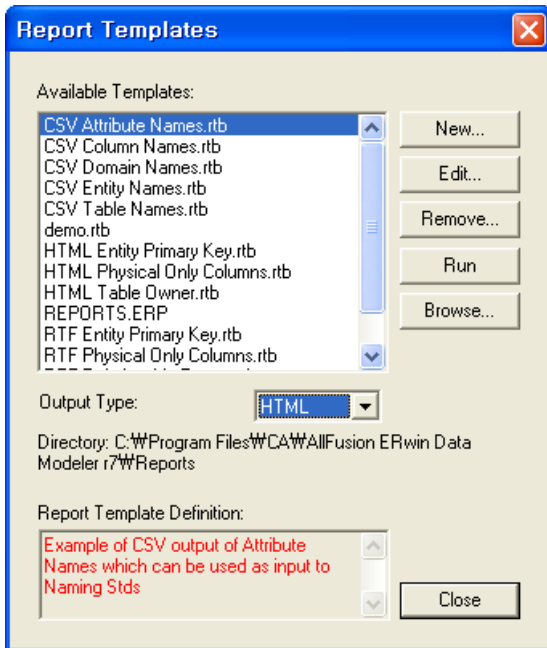
RTB는 ERwin 모델로부터 레포트를 생성 할 때 다양한 커스터마이징을 제공한다. 이것은 사용자가 작성한 레포트 템플릿을 재사용하고 재작성 하는데 도움을 주며, 레포트 타입별로 그룹을 만들어 낼 수 있다. 이 기능은 Data Browser와 동일하게 모델이 가지고 있는 다양한 정보를 대상으로 모든 레포트를 생성 할 수 있으며, 다른 사용자에게 배포 및 공유가 가능하다.

또한 RTB는 모델의 ERD 즉, 그림 정보까지 출력이 가능하도록 지원하고 있다. 레포트 양식은 .rtb 파일로 저장할 수 있으며 재사용이 가능하다. 지원하는 파일 포맷은 HTML, RTF, TEXT, PDF 등이 있다. R7 이전 버전에서는 PDF를 지원하지 않았으나 본 버전부터 PDF로도 출력이 가능하다.

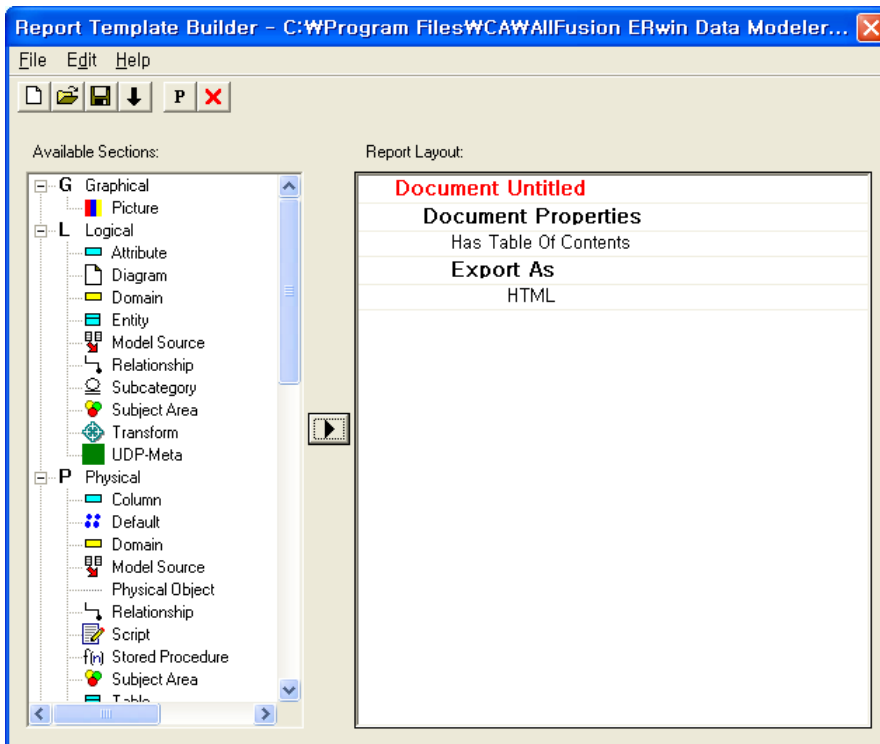
1) 새로운 레포트 템플릿의 생성



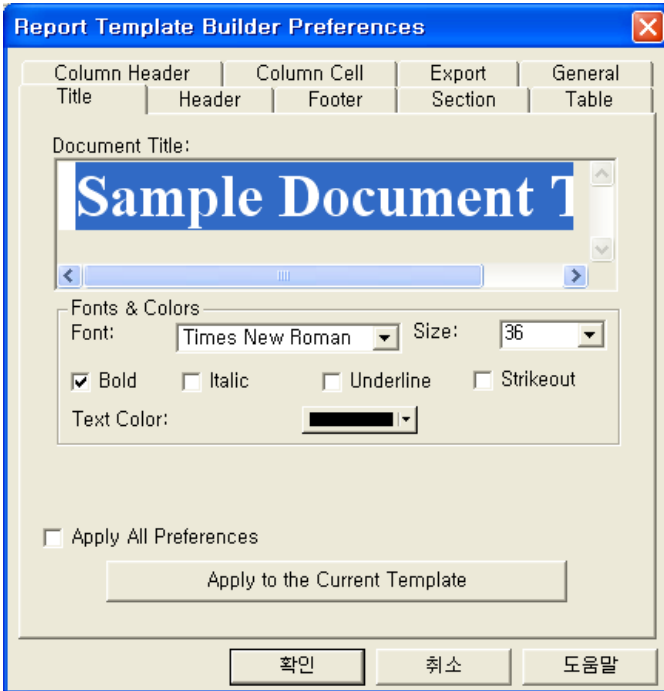
① Tools > Report Template Builder > Report Builder.. 을 선택한다.



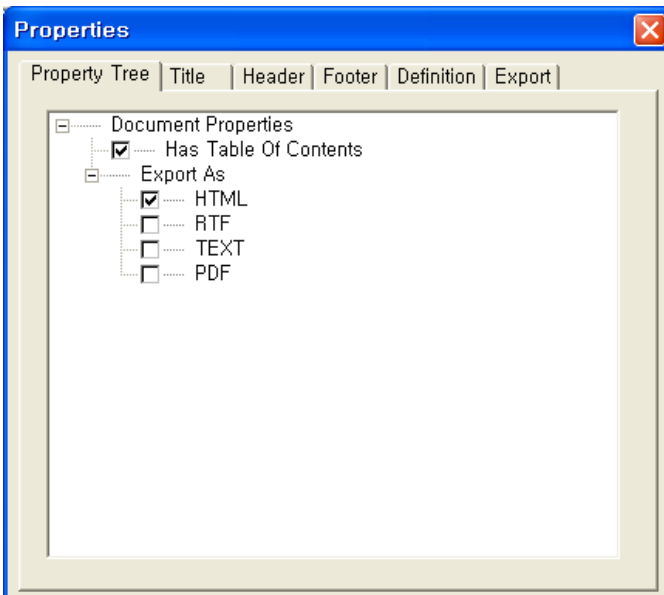
- ② Output Type을 HTML로 선택한다.
(HTML, RTF, TEXT, PDF 포맷 지원)
- ③ New를 클릭해 새로운 템플릿을 생성한다.



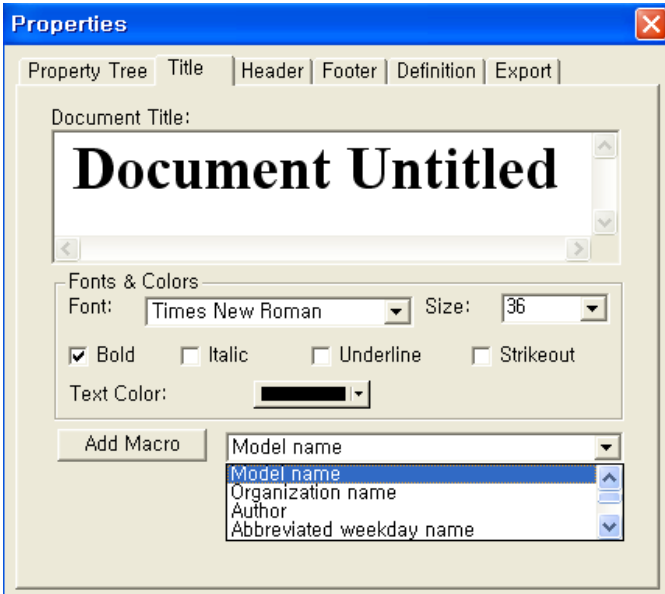
- ④ Report Template Builder Edit 화면이다. 좌측 트리에서 모델의 정보를 확인 할 수 있으며 트리에서 항목을 더블클릭 하면 우측의 Report Layout 화면에 항목이 추가가 된다.



- ⑤ Edit > Preference에서 기본적인 템플릿 항목에 대해서 글꼴, 크기, 색상 등 기본적인 세팅이 가능하다. (Title, Header, Footer, Export 등)



- ⑤ 레포트의 기본정보는 Edit 메뉴의 Properties에서 입력한다.
 - Property Tree : 레포트 포맷 지정
 - Title : 제목 입력
 - Header / Footer / Definition 입력
 - Export : 저장 환경 설정
- ②번 항목에서 HTML을 선택하였기 때문에 HTML로 체크가 되어있다 .하지만 여기서 변경이 가능하다.

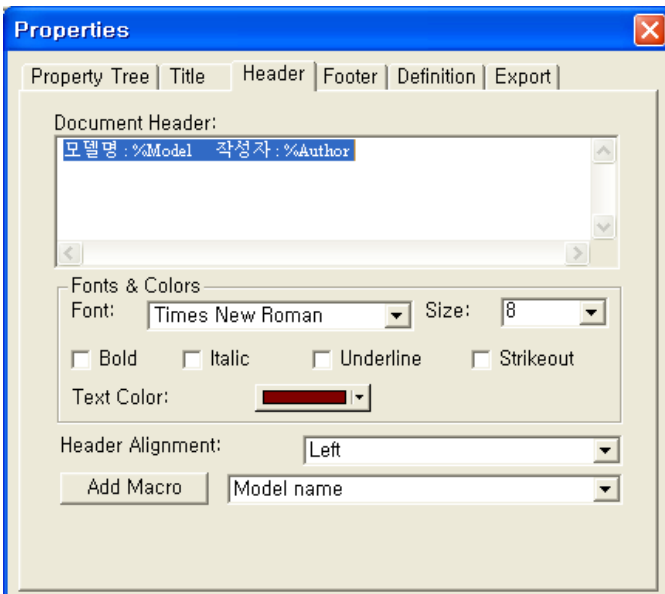


⑥ Title 탭에서는 레포트의 제목을 입력 할 수 있으며, 사용자는 Macro를 사용해서 제목을 자동으로 입력 할 수 있다.

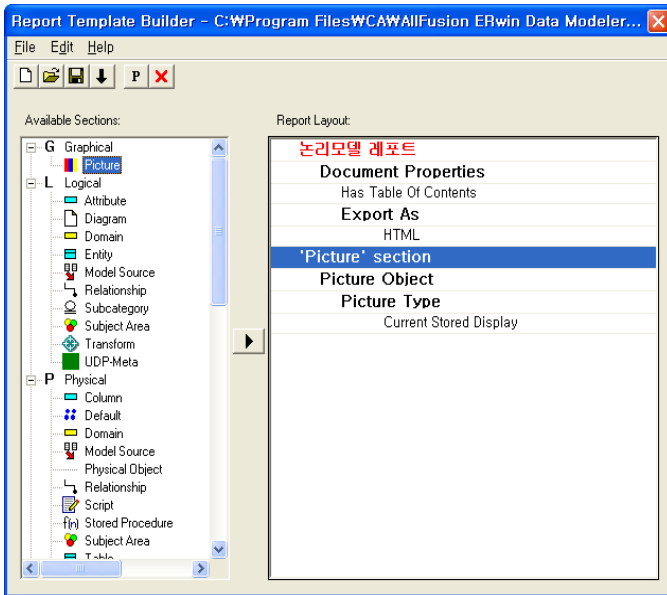
Macro는 모델의 정보를 자동으로 입력 할 수 있도록 지원한다.

항 목	설 명
Model Name	모델명 입력
Organization ..	조직명 입력
Author	사용자 명 입력
Abbreviated ..	날짜 정보 입력
기타	총 15개 항목 지원

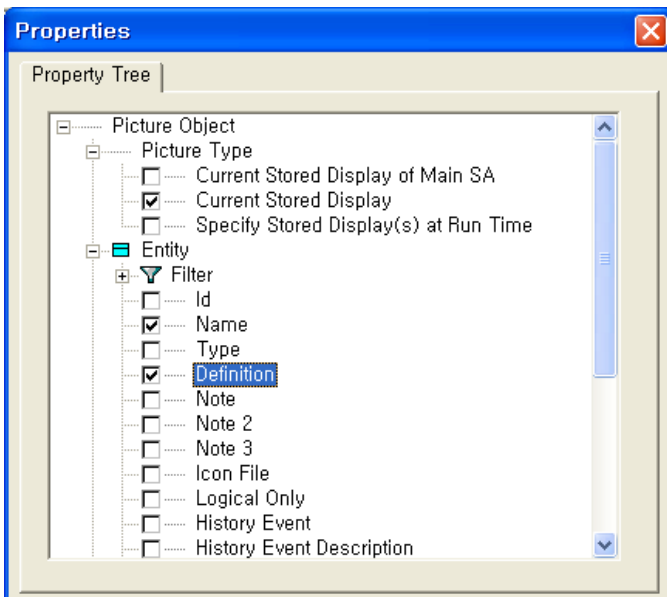
또한 Title의 스타일을 세팅 할 수 있다.



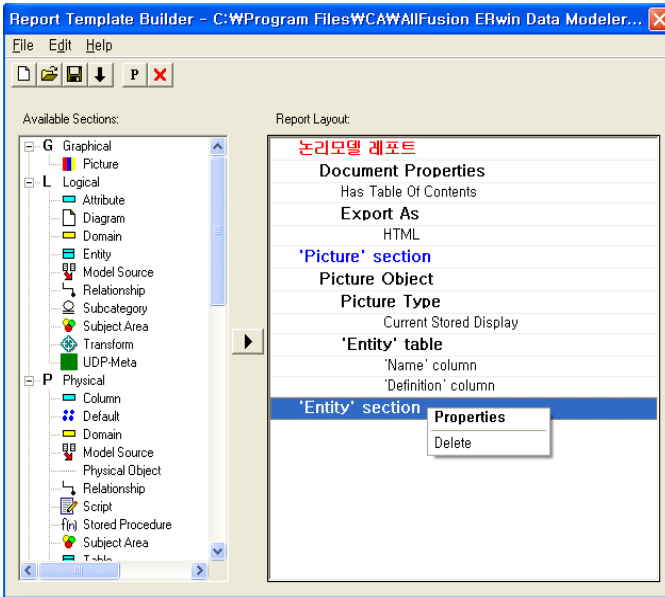
⑦ Header와 Footer 탭페이지에서는 각 레포트 섹션마다 페이지 윗부분, 아래 부분에 동일하게 입력되는 제목을 입력 할 수 있다. 레포트 정보에 대한 매크로를 함께 지원한다. (Add Macro) 예를 들어 모델명이나 작성자 명, 조직명 등에 대해서 일일이 데이터를 입력하지 않아도 한번 매크로를 통해 입력이 가능하다



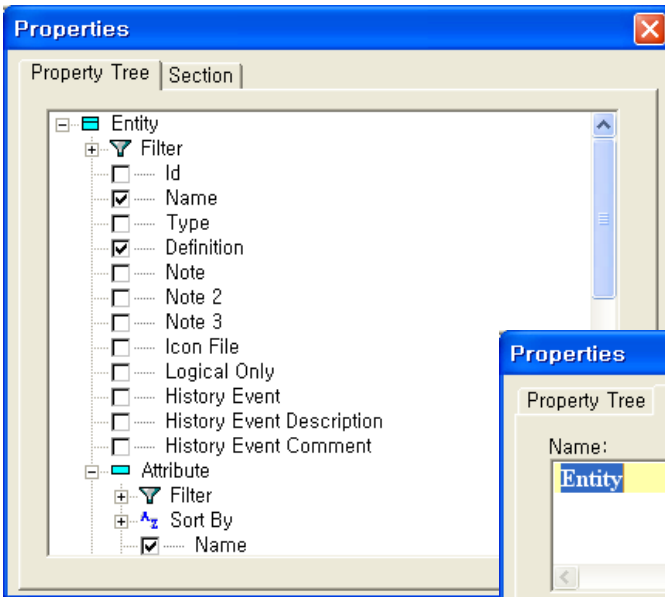
- ⑧ Report Template에 섹션을 추가하려면 왼쪽 트리에서 항목을 선택하고 더블클릭을 하거나 ▶ 아이콘을 선택한다. (Picture 섹션을 추가한다.)



- ⑨ Report Layout의 'Picture' section 항목을 더블클릭 하거나 마우스 오른쪽을 클릭하여 Properties를 선택하면 위와 같은 화면이 실행된다. Picture Type에서 Current Stored Display는 현재 디스플레이 되는 모델을 출력 하겠다는 것이며, Specify Stored Display at Run Time은 레포트가 실행 될 때 Display 항목(Subject Area)을 선택할 수 있는 옵션이다. 그리고 Entity의 선택 옵션은 Picture를 HTML로 보내고 엔티티 그림을 클릭 했을 때 엔티티에 대한 정보를 Text로 나타내는 옵션이다.



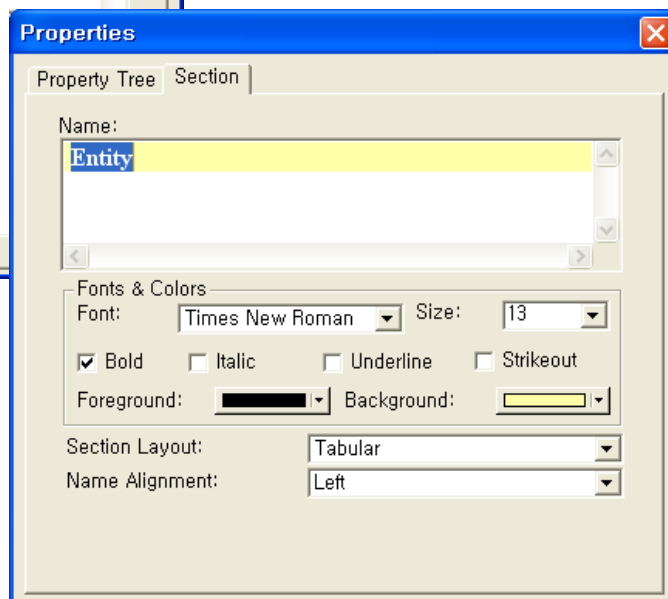
⑩ 또 다른 세션을 추가하고 Properties를 선택한다. (Entity 세션 추가)

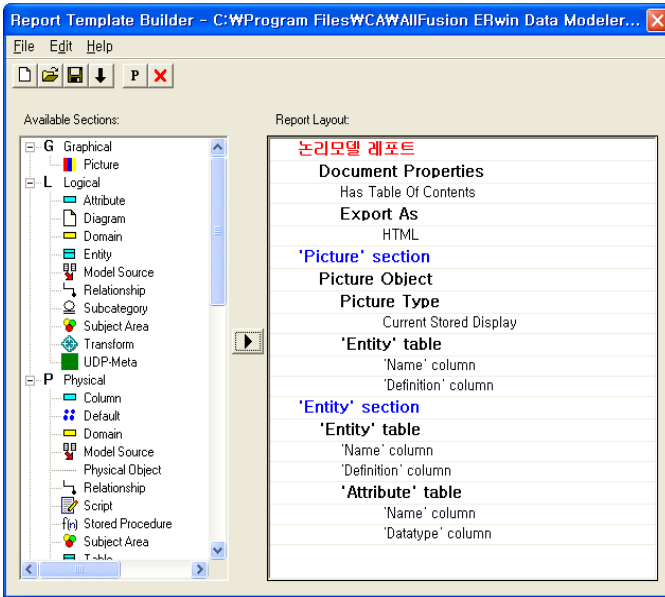


⑫ Section 탭에서는 항목의 스타일을 지정한다.

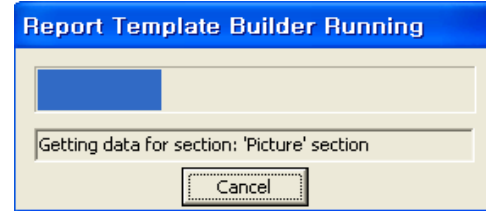
⑪ Property Tree에서 Entity 세션에서 출력할 정보를 선택한다.

- Entity Name
- Entity Definition
- Attribute Name
- Attribute Definition
- Attribute Datatype

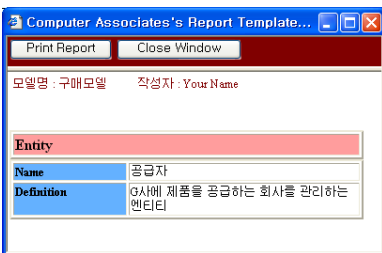
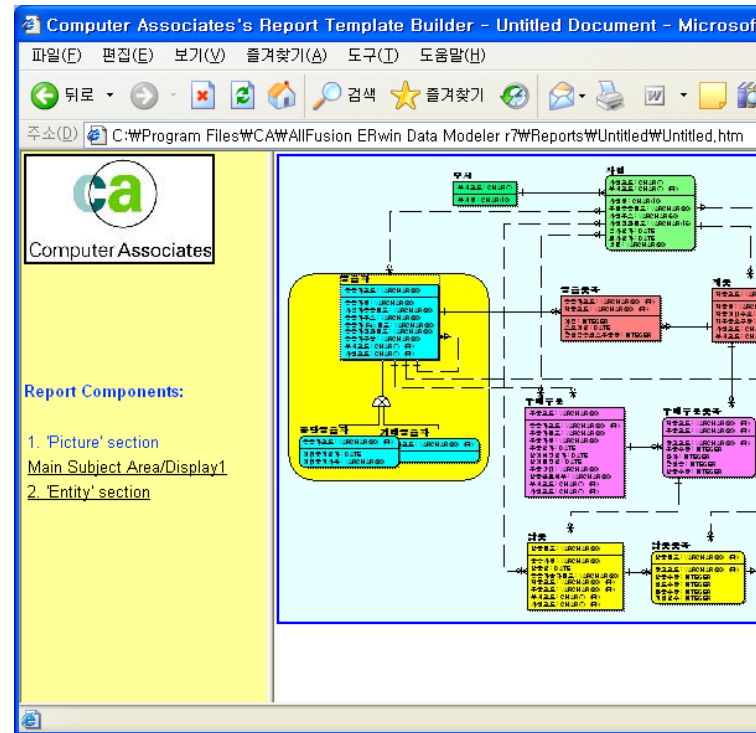




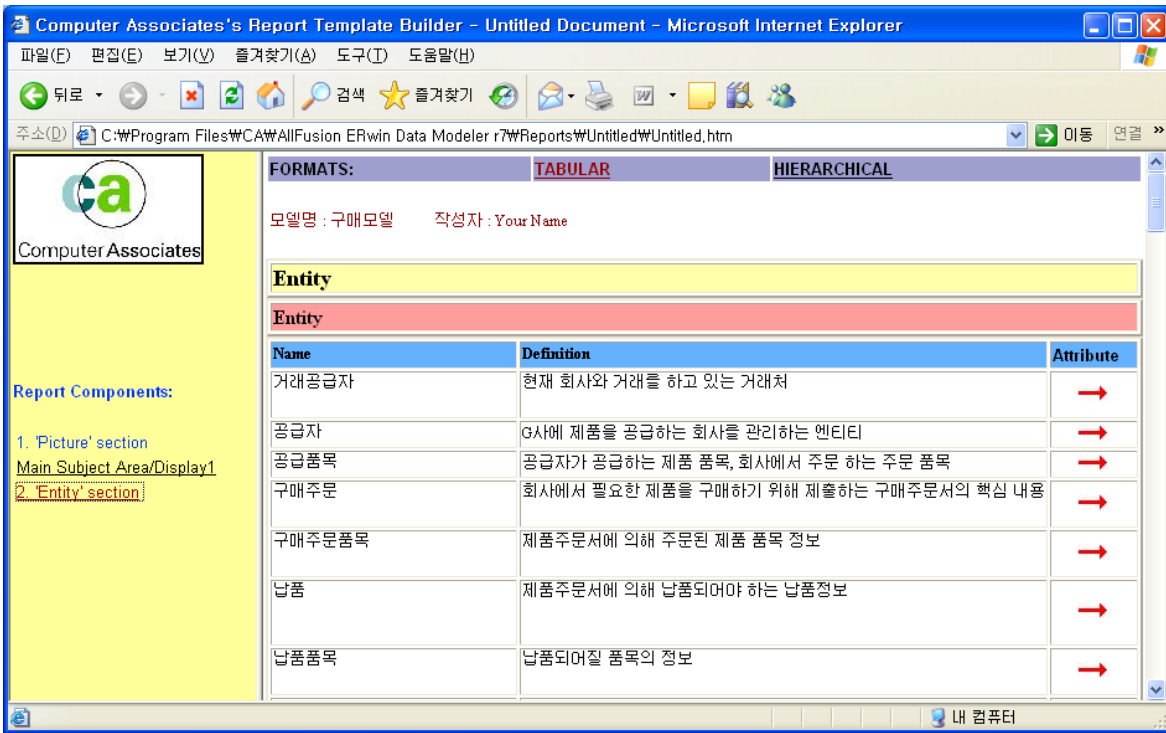
⑬ 레포트 항목과 스타일 세팅이 완료되면 File > Run을 실행하거나 ↓ 아이콘을 선택한다.



아래 그림과 같이 HTML 레포트가 자동 생성된다. 왼쪽 프레임에서 Picture 섹션을 클릭하면 ERD를 볼 수 있다.




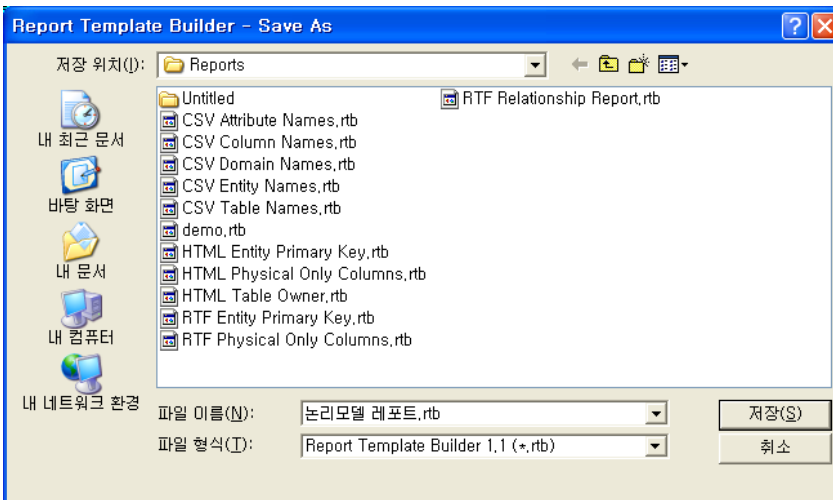
⑭ ERD에서 엔티티를 클릭 했을 경우 해당 엔티티의 Text 정보를 옆의 그림과 같이 Pop Up 창을 통하여 보여준다.



⑮ 'Entity' section을 선택하면 Text 정보를 확인 할 수 있으며, 화살표를 클릭하면 그와 관련된 하위 섹션으로 이동한다.

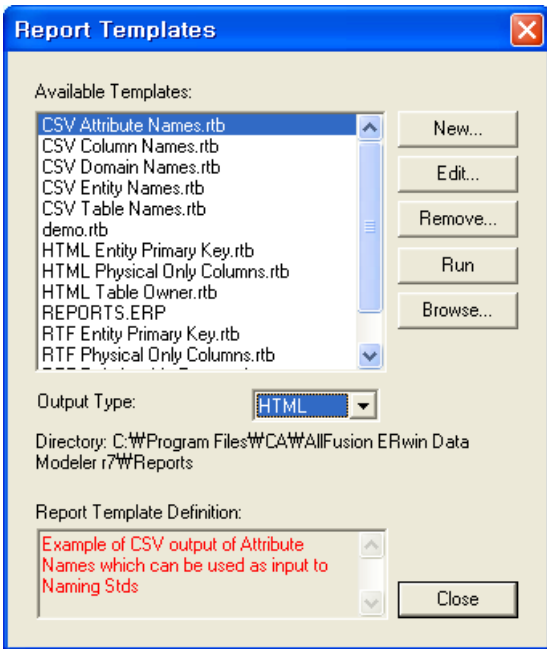
2) RTB 파일의 저장

생성한 RTB를 저장하려면 File > Save 메뉴나,  아이콘을 선택해 저장한다.

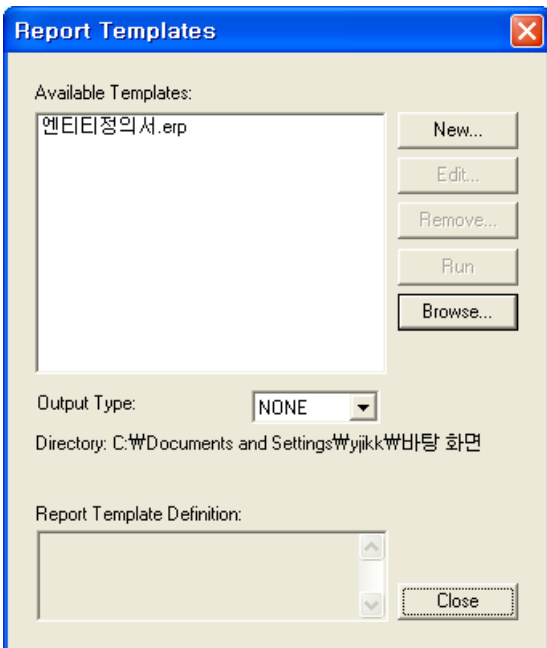


3) ERP → RTB 파일로의 변환

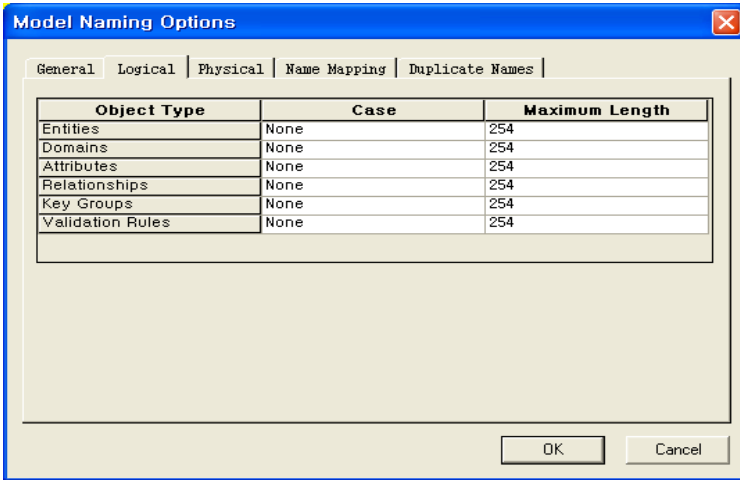
Data Browser 에서 작성된 ERP파일을 읽어 RTB에서 출력이 가능하다. ERP 파일을 RTB로 자동 변환 할 수 있으며 이렇게 변환된 RTB 파일을 통해 레포트를 출력 할 수 있도록 지원 한다.



- ① RTB를 실행한다.
- ② Browse를 클릭해 .erp 파일의 경로를 지정한다.



- ③ 그림과 같이 RTB에서 .erp 파일이 등록된다.
- ④ Output Type을 선택하고 Run을 실행한다.

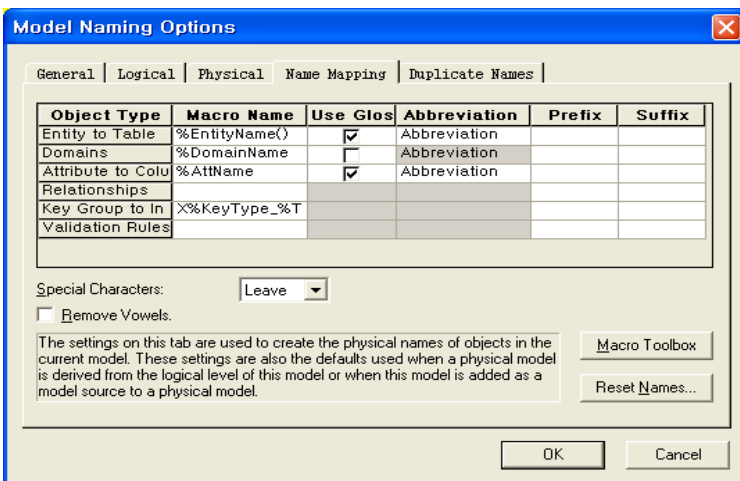


Logical / Physical 탭에서는 Logical 모델에 사용된 Object 명의 대소문자 표현 방법과 최대 길이를 지정 할 수 있다.

[Case]

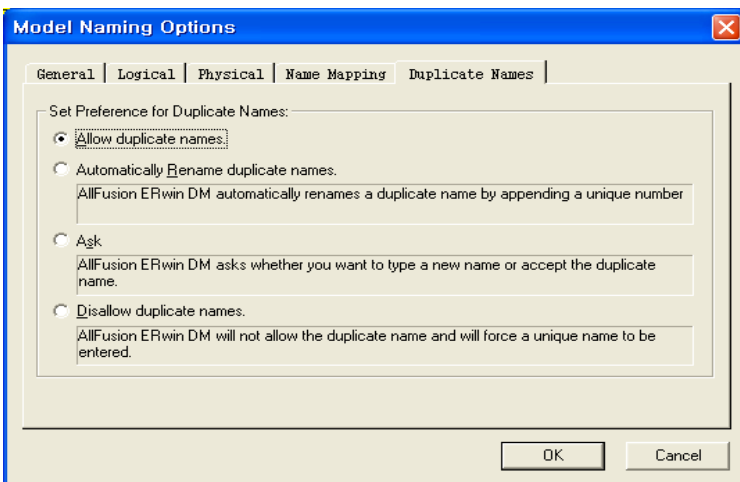
항 목	설 명
UPPER	대문자로 설정
lower	소문자로 설정
Initial	첫 글자만 대문자로 설정

Logical의 Maximum Length는 254byte 까지 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 Name	해당 Object 들에 대한 Macro를 지정한다.
Use G..	NSM을 매핑한 경우 Object 들에 대해서 적용 여부를 체크한다.
Abb..	Abbreviation(영문 풀 네임) 인지 Alter Abb..(영문 약어) 로 적용할지 선택
Prefix	접두어 설정
Suffix	접미어 설정

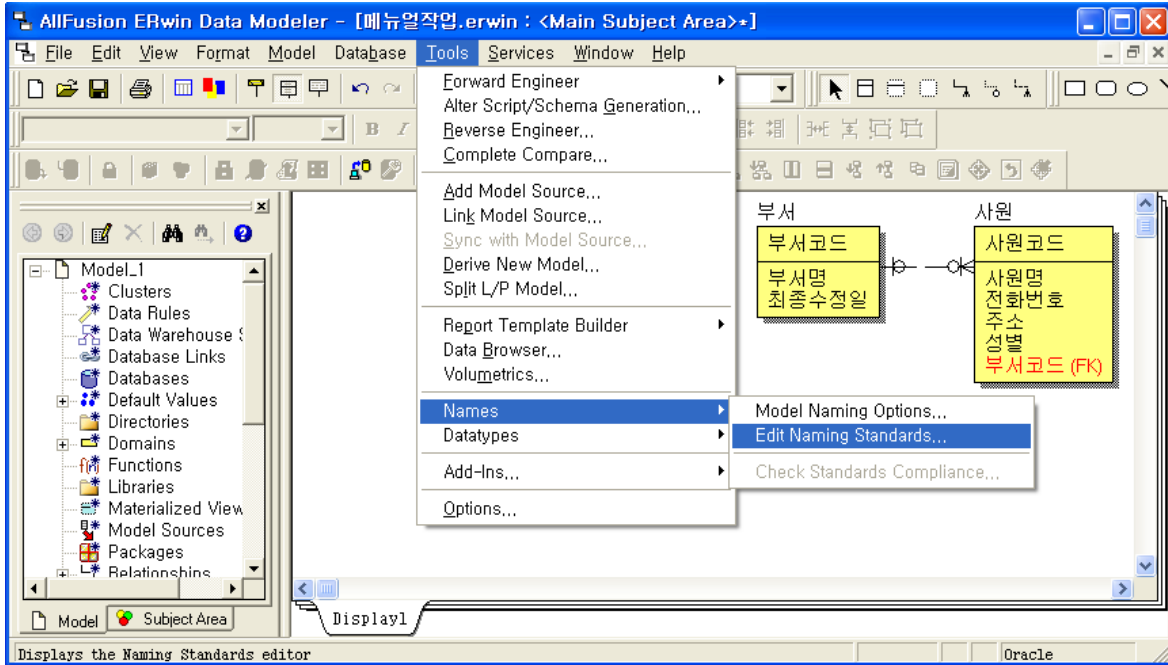


모델에 사용되는 이름이 중복될 경우 처리 옵션을 선택한다.

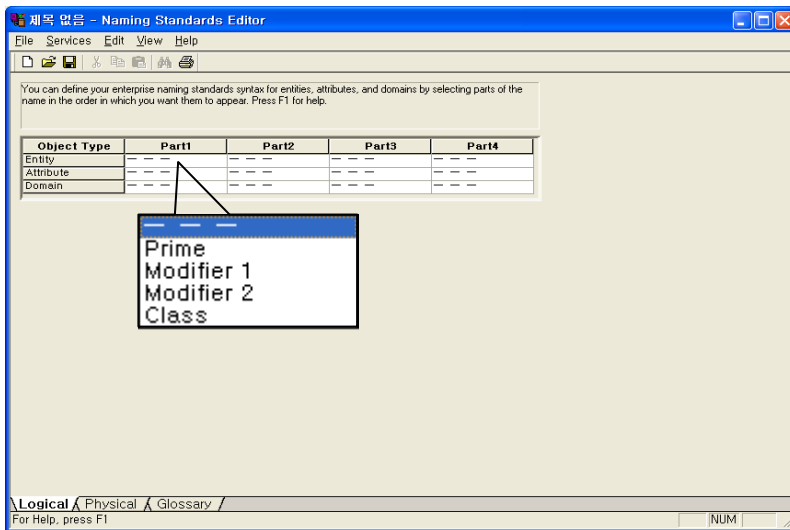
항 목	설 명
Allow	중복을 허용한다.
Auto...	중복되면 일련번호를 붙여 자동으로 Rename 한다.
Ask	사용자가 Rename 한다.
Disallow	중복을 허용하지 않으며 중복되면 E_2와 같이 Default로 이름이 부여된다.

2) Edit Naming Standards (.NSM 파일의 생성)

NSM 기능을 통해 용어사전(.nsm)을 생성 하고, 모델과 매핑하여 물리명을 영문으로 변환한다.

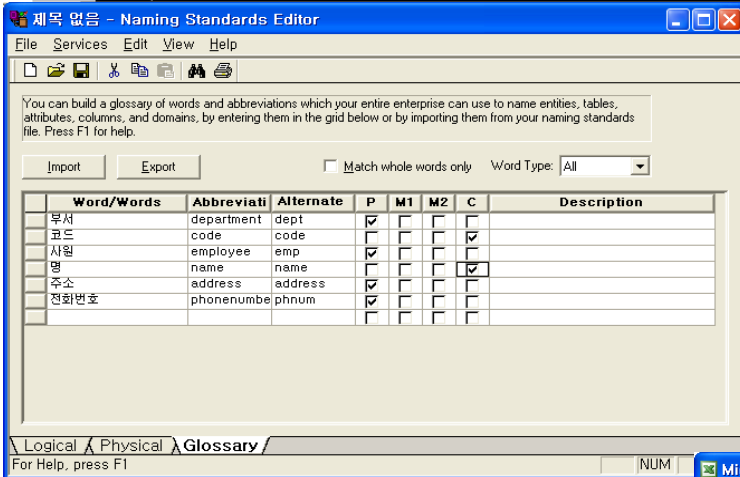


① Tools > Names > Edit naming Standards..를 선택한다.




② Logical / Physical 탭 에서는 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, 부서 혹은 Prime과 Calss를 구분할 수 있는 특정 단어
Class	데이터 타입이나 Editor의 주요 종류를 묶어 지칭하는 단어 (date, code, name)

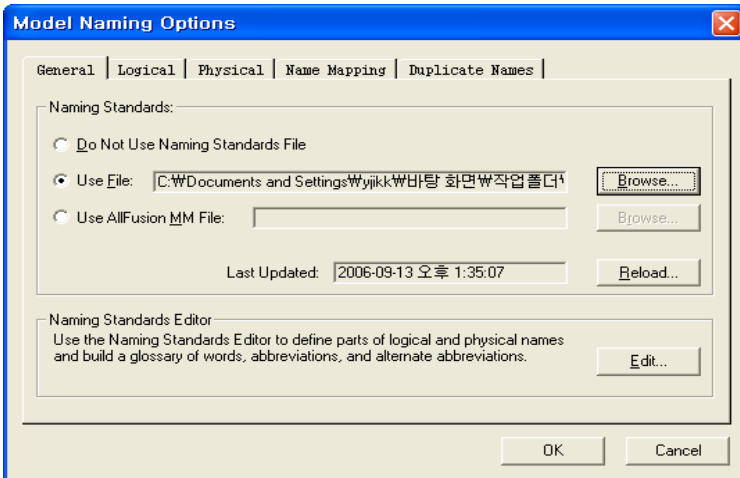
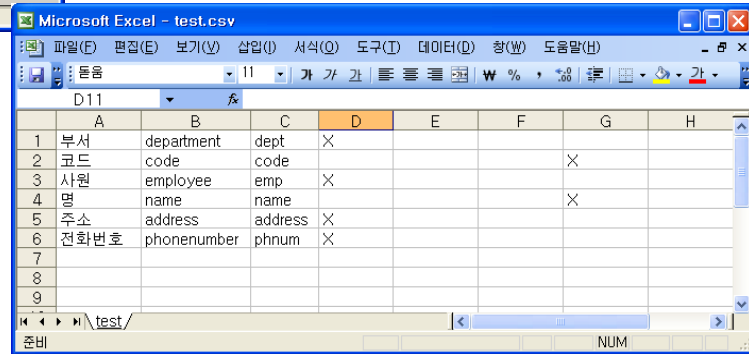


③ Glossary 탭에서 용어사전을 등록 한다.

항목	설명
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 가능하다.

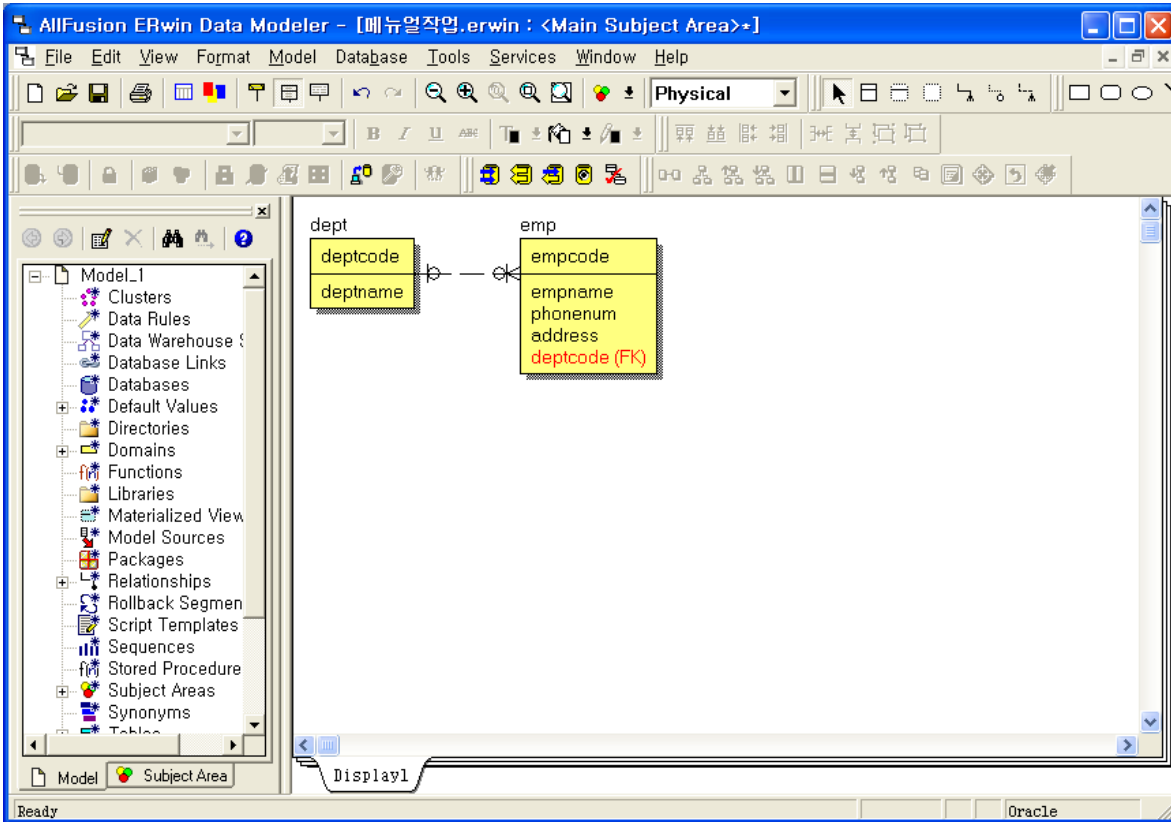


⑥ Use File 에서 적용할 Nsm 파일을 선택하고 OK를 버튼을 선택한다.

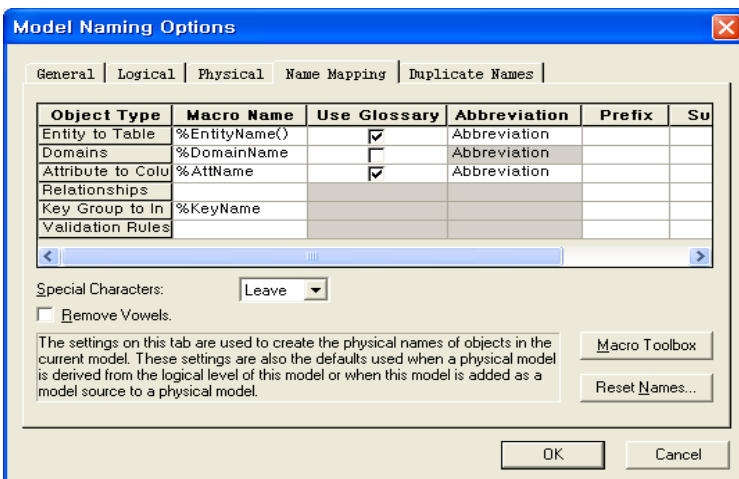
Tip

Match whole words Only

R7 에서는 NSM 변환 룰이 4.1.3_4033과 다르다. R7의 변환된 룰이 국내 실정에 맞지 않아 R7의 룰과 4033의 룰을 선택 할 수 있도록 개선 되었다. (Check : R7 룰, Uncheck : 4033 룰 적용)



⑦ 용어사전에 정의한 대로 물리명이 영문으로 변환된다.



⑧ 만약 NSM 파일을 매핑 하였는데도 영문 명이 변환 되지 않는 경우에는 Model Naming Option 에서 Use Glossary를 다음과 같이 설정한다.

선택한 Object에 대해 용어사전을(Use Glossary)을 사용하겠다는 의미이다.

12.2 Datatypes..

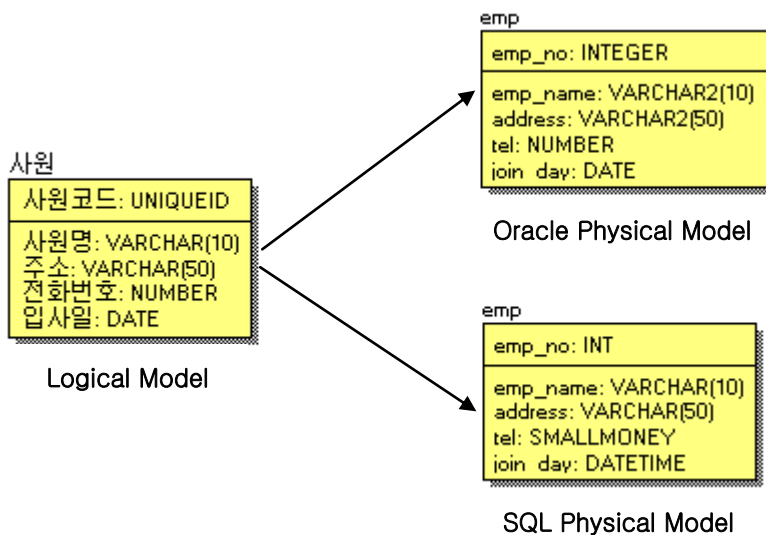
Datatypes(DSM) 기능은 NSM 기능과 매우 흡사하다. 이 기능은 모델의 Datatype을 시스템에 따라서 일괄 변경할 수 있도록 하는 기능을 제공한다. DSM은 사용자의 Design Layer 구조 생성시 매우 중요한 역할을 하며, 특히 Logical 모델에서 다양한 Physical 모델로의 변환 시 유용하게 사용된다. 즉 하나의 논리모델에서 다양한 물리 모델로 변환하고자 할 경우 ERwin의 Logical Datatype과 Target DBMS 별 Datatype 매핑 정보를 이용해 손쉽게 변경 할 수 가 있다.

1) ERwin에서 지원하는 Logical Datatypes

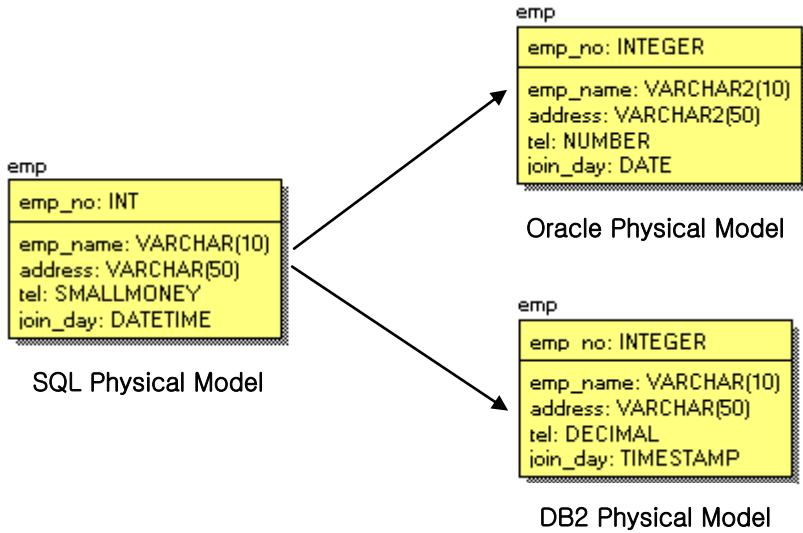
Logical Datatype은 Attribute 내에 저장되어 있는 가장 기초적인 데이터의 특성을 나타낸다. ERwin Data Modeler는 다음과 같은 Logical Datatype을 기본으로 지원한다.

Logical Datatypes			
AUDIO	MONEY	INTERVAL	TIMESTAMP
SHORT FLOAT	DECIMAL	BINARY	NCHAR
LARGE BINARY	UNIQUEID	SMALLINT	FLOAT
BOOLEAN	NUMBER	LONG	VARCHAR
TEXT	HUGE	CHAR	NVARCHAR
LONG FLOAT	VIDEO	TIME	INTEGER
DATE			

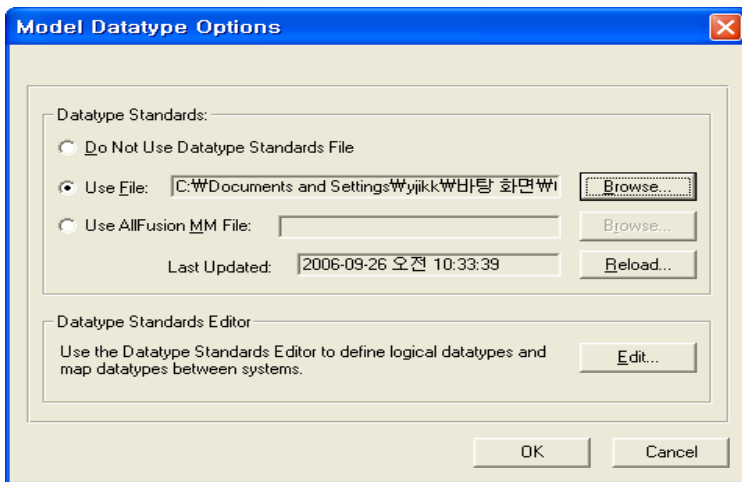
2) Logical → Physical 매핑



3) Physical → Physical 매핑



4) Model Datatype Options

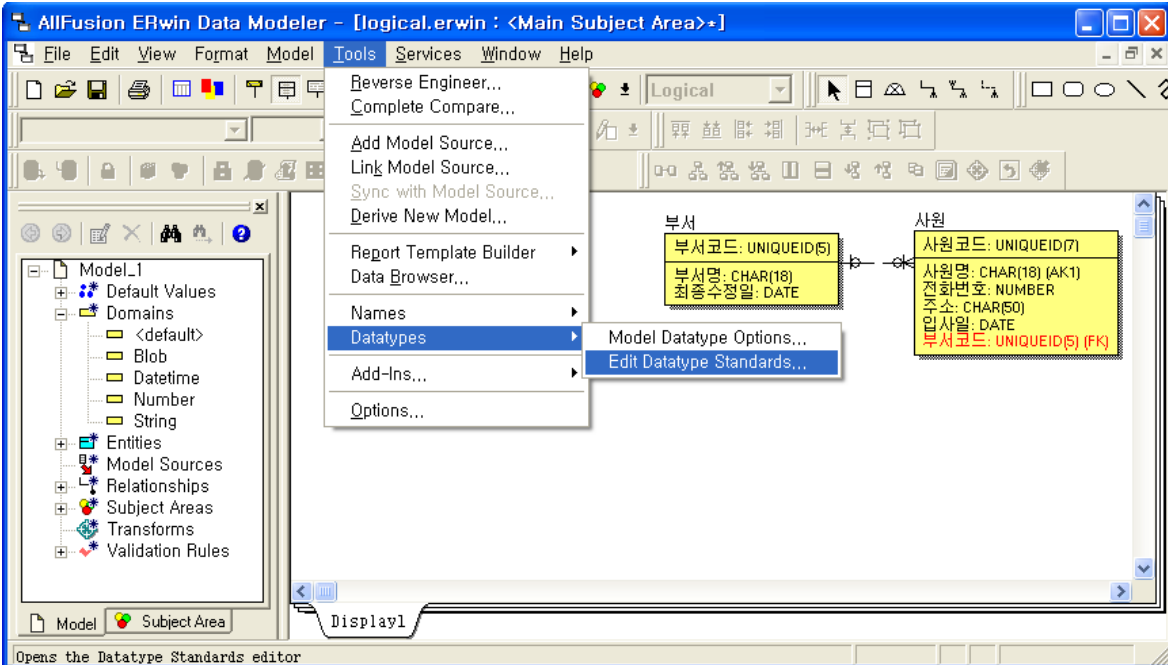


Model Datatype Options에서는 사용자가 작성한 .dsm 파일을 매핑 할 수 있다.

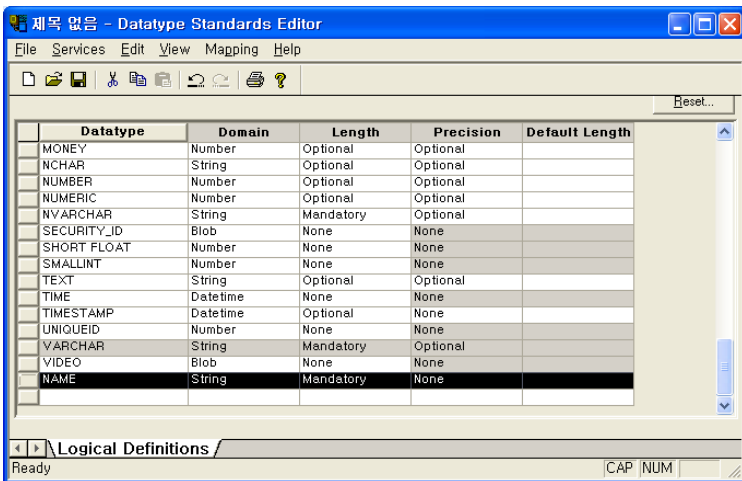
* 옵션 설명 *

항 목	기 능 설 명
Do not Use Datatype Standards File	dsm 파일을 사용하지 않는다.
Use File	dsm 파일을 모델과 매핑한다. (Browser을 통해 dsm 파일 경로 지정)
Use Allfusion MM File	Model Manager에 저장되어 있는 dsm 파일과 매핑한다.
Edit	Datatype Standards Editor를 실행한다.

5) Edit Datatype Standards (.DSM 파일의 생성)



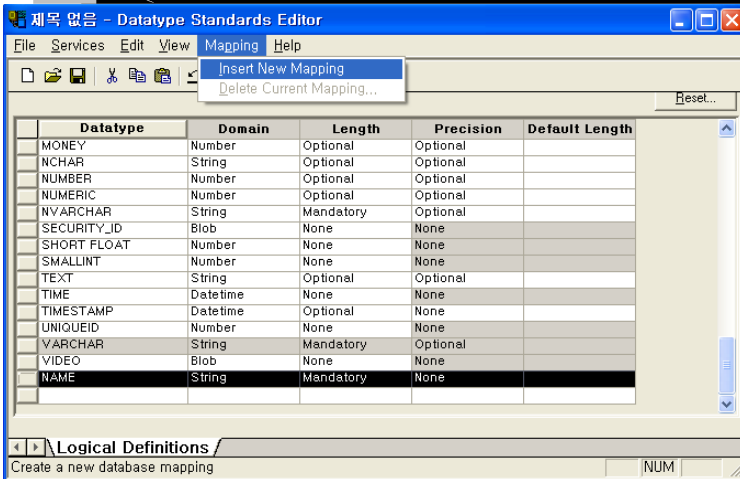
① Logica 모델을 열고 Tools > Datatypes > Edit Datatype Standards.. 을 선택한다.



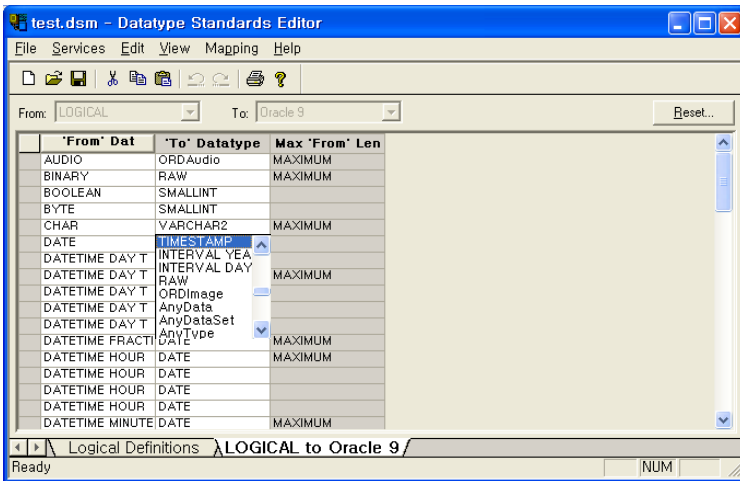
② Logical Definitions 탭에서는 ERwin에서 제공하는 Logical Data type을 확인 할 수 있으며 사용자는 새로운 데이터 타입을 생성 할 수 있다.

* 새로운 데이터 타입 추가 *

- 리스트의 맨 아래 줄로 이동하여 Datatype에 새로운 항목을 입력한다.
- 새 Datatype을 ERwin Domain에 등록한다.
- 데이터 길이(Length)를 지정한다. 예를 들어 NUMBER를 지정할 때 Optional로 선택할 경우 NUMBER, NUMBER()로 2개의 타입이 나오며, Mandatory로 지정하면 NUMBER()로 나타난다.
- 소수점(Precision) 옵션을 지정한다. 예를 들어 NUMBER를 지정할 때 Optional을 선택할 경우 NUMBER(,)가 나타난다.



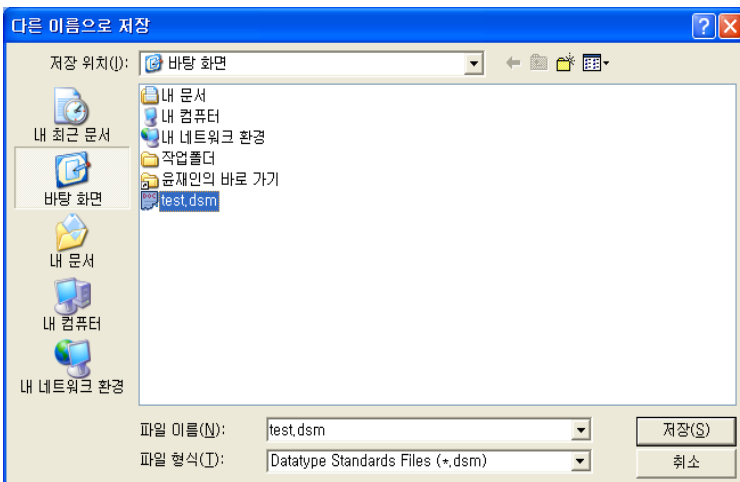
③ 새로운 매핑 옵션을 생성하려면 Mapping > Insert New Mapping을 선택한다.



④ Insert New Mapping을 하면 새로운 탭이 생성된다. 그리고 From 과 To의 항목을 선택한다.

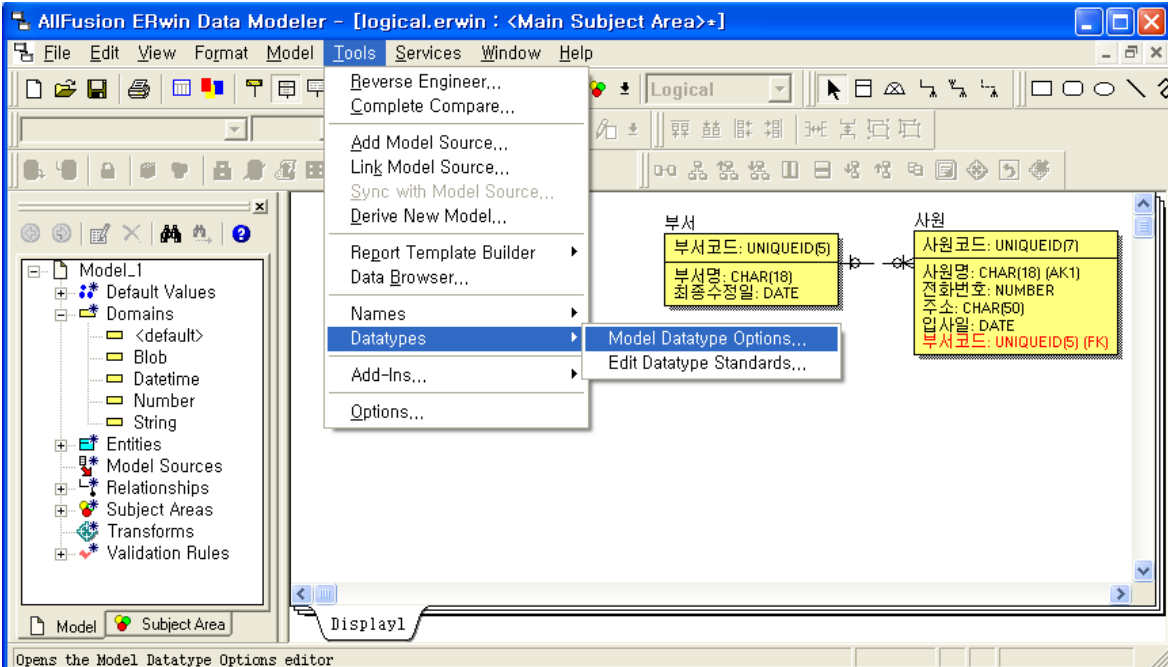
(From : Logical, To :Oracle 9)

⑤ 'To' Datatype을 클릭하여 개별적으로 매핑 작업을 한다.

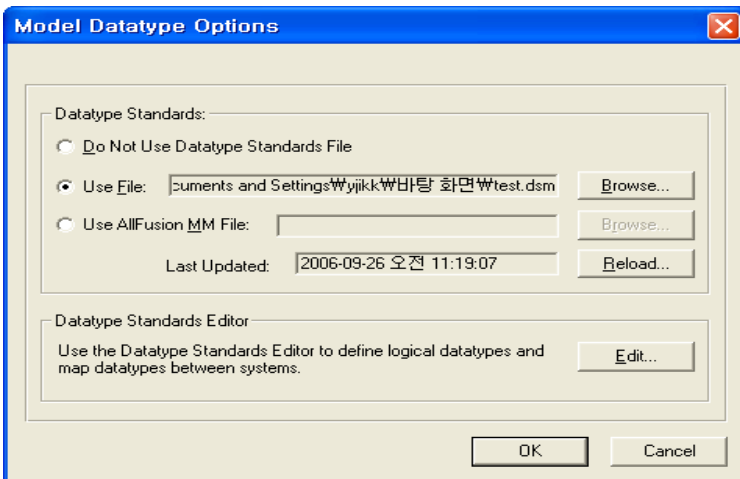


⑥ dsm 파일을 저장한다.

6) Datatype Standards(.dsm)과 모델의 매핑

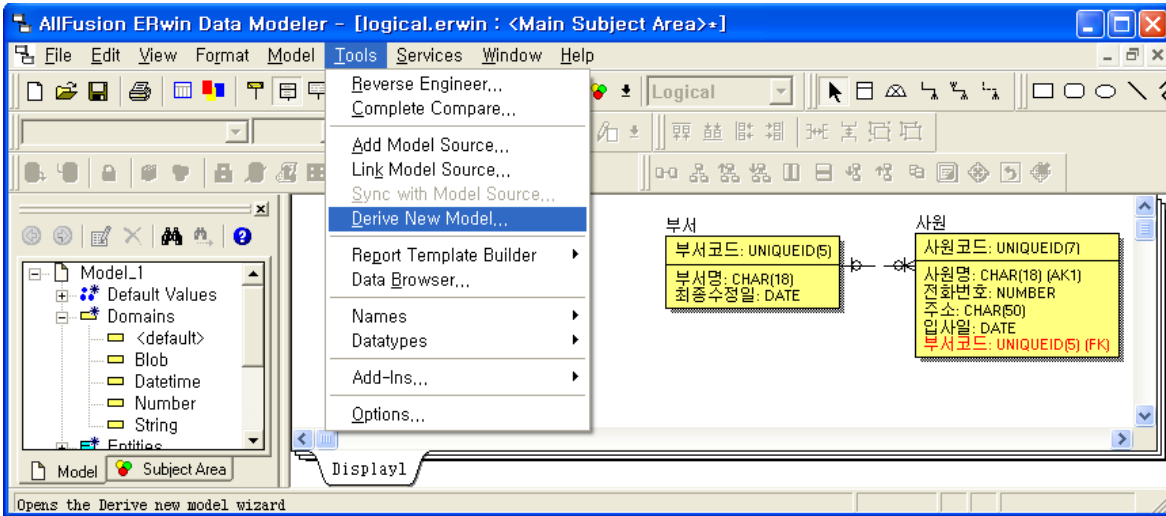


① Tools > Datatypes > Model Datatype Options.. 을 선택한다.

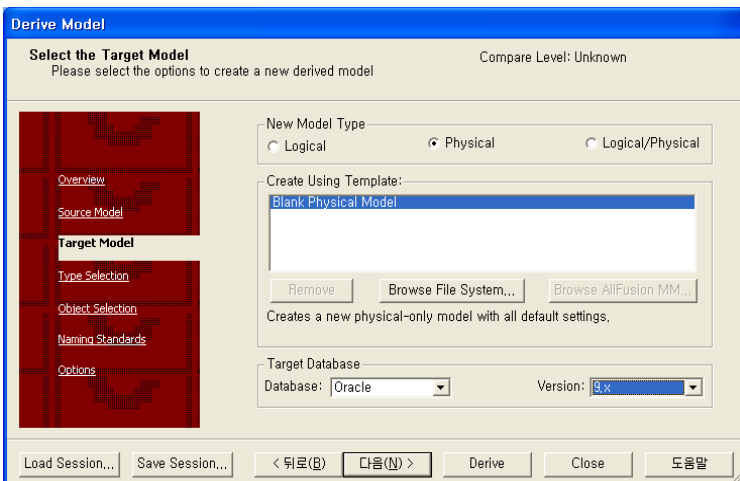


② Use File 에서 적용할 dsm 파일을 선택하고 OK를 버튼을 선택한다.

7) Design Layer 모델에 Datatype Standard(.dsm) 적용하기

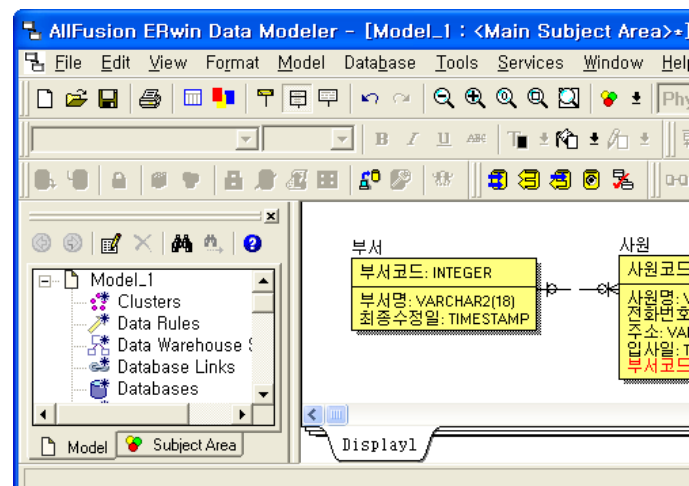


① Logical 모델에서 새로운 Physical 모델로 변환하기 위해 Tools > Derive New Model을 선택한다.



② Physical Type을 선택하고

Target Database를 Oracle 9.x를 선택 한다. 다음을 클릭하고 마지막 단계에서 Derive 버튼을 클릭해 작업을 종료한다.



③ 그림과 같이 dsm 에서 정의한 매핑 정보에 따라 데이터 타입이 바뀌게 된다.

Dimensional Modeling



13. Dimensional Modeling

ERwin Data Modeler r7에서는 데이터웨어하우스 또는 의사결정지원 시스템(DSS)을 설계하기 위해 사용되는 다차원모델링(Dimensional Modeling)을 지원한다.

다차원 모델이란, 비즈니스 사용자의 직관에 따라 손쉬운 분석을 할 수 있도록 설계된 분석 주제 중심의 데이터 모델을 말한다. 그렇기 때문에 실제 세상을 그대로 표현하고자 하는 정규화 모델(Normalized Model)과는 달리 빠른 분석 수행을 목적으로 테이블이 매우 비정규화 되어 있다.

다차원 모델은 분석하고자 하는 차원인 Dimension Table과, 분석하고자 하는 대상이 되는 Fact Table로 이루어져 있으며, Fact Table을 중심으로 Dimension Table이 별처럼 Join되어 있어 'Star Schema' 라고도 불린다. 예를 들어, '일자별 / 요일별 매출액 분석'의 경우 일자, 요일이 Dimension Table의 값이 되고, 분석의 대상인 매출액이 Fact Table의 값이 된다.

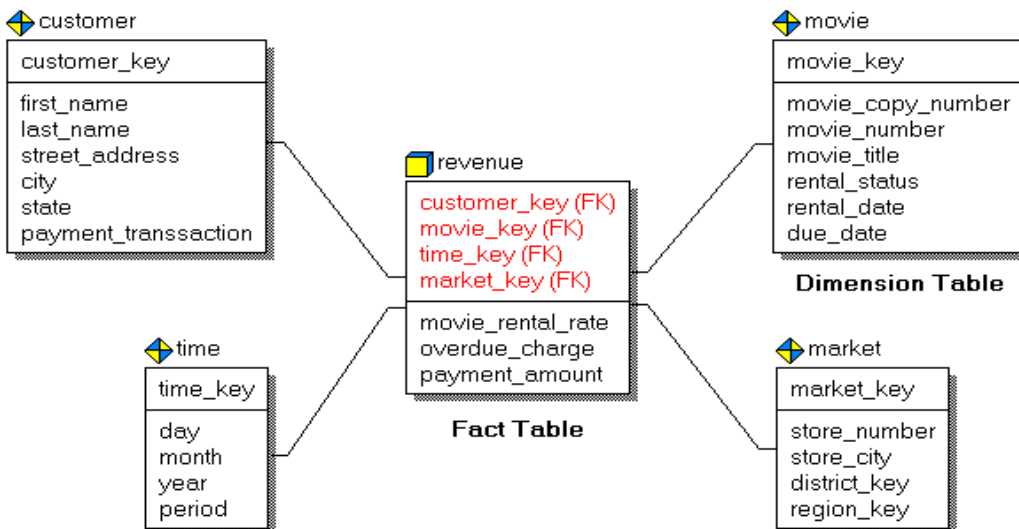
13.1 Data Warehouse

Data Warehouse는 사용자의 의사 결정에 도움을 주기 위해 다양한 운영 시스템에서 추출, 변환, 통합되고 요약된 데이터베이스를 말한다. Data Warehouse 설계할 경우 다차원 모델이 적합한데 이를 논리적(Conceptual)으로 설계할 때 자주 사용되는 스키마는 스타 스키마(Star Schema)와 스노우플레이크 스키마(Snowflake Schema)이다.

1) 스타 스키마(Star Schema)

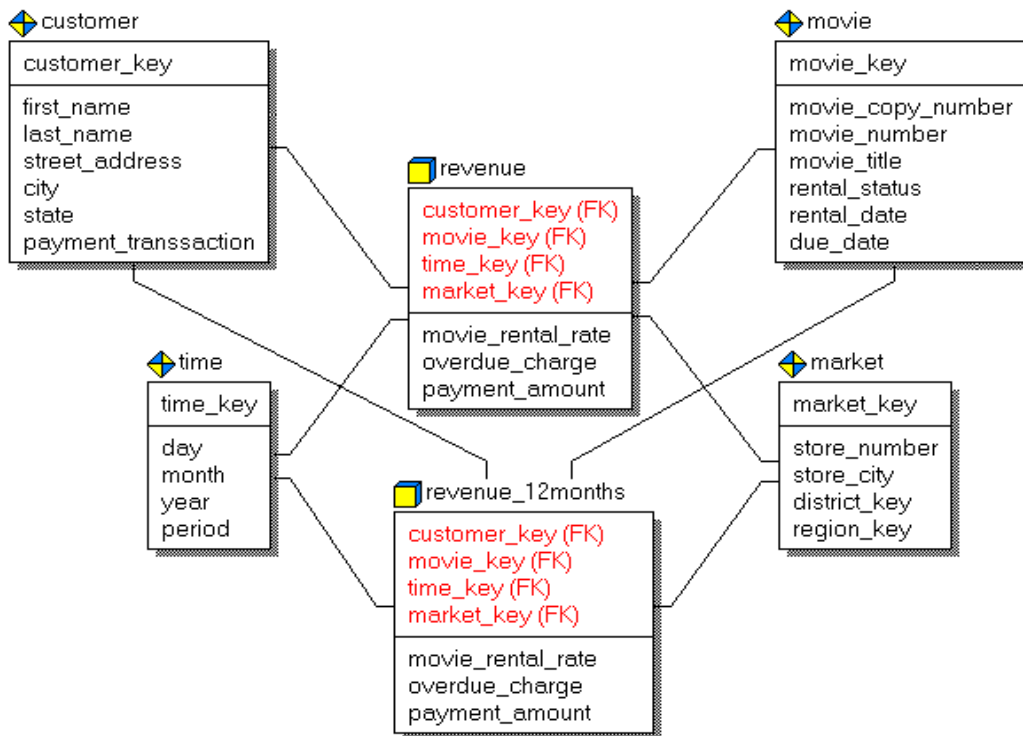
1.1) Fact / Dimension Table

스타 스키마는 필요한 사실과 기타 속성들로 이루어진 Fact Table과, 관심 있는 차원과 그에 따른 추가적인 정보들을 각각 하나의 차원 테이블로 설계해, Fact Table과 Dimension Table을 Foreign Key로 연결할 수 있도록 한다. 그림 상으로 보면 마치 별 모양이 되도록 구성한 스키마이다.

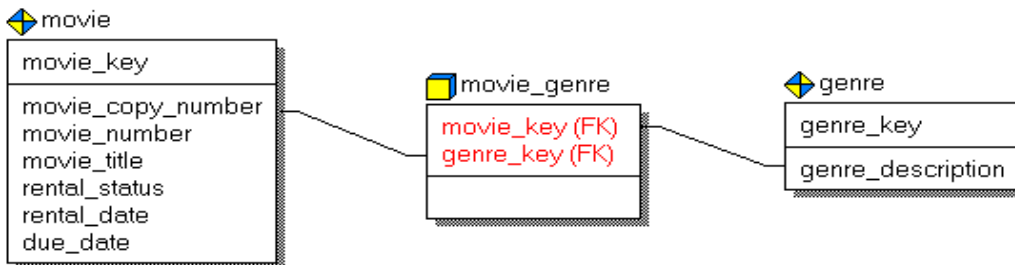


Fact Table은 Dimension Table과 식별관계를 맺고 있다. 즉 Fact Table의 PK는 Dimension Table의 PK들이 이주되어 생성된 FK로 구성된다.

Dimensional Model은 하나 이상의 Fact Table을 구성할 수 있다. 만약 그 Diagram이 여러 가지 업무 프로세스를 모델링 하게 된다면, Fact Table이 관련 없는 사실을 포함하고 있다면, 사실에 관한 데이터가 다른 시간에 수정되었다면 여러 Fact Table을 하나의 Diagram에서 포함할 수 있다. 아래 그림과 같이 REVENUE 테이블은 주간별로 업데이트 되지만 PREVIOUS_12MONTHS 테이블은 월별로 갱신된다.

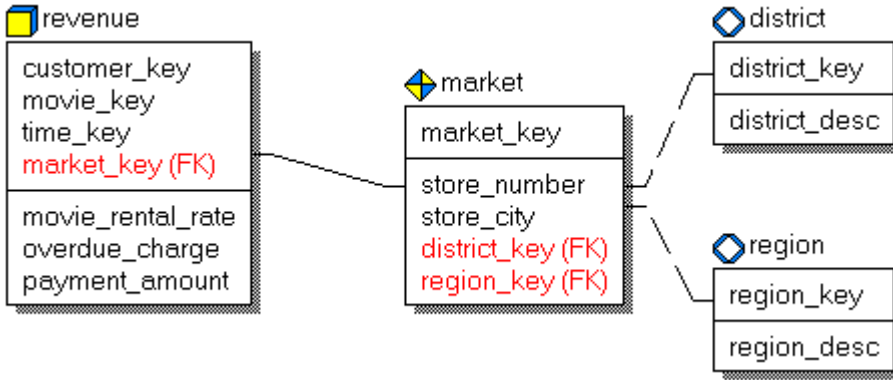


2개의 Dimension Table 사이에 M : M 관계를 해결하기 위해서 Multiple Fact Table을 사용할 수 있다.



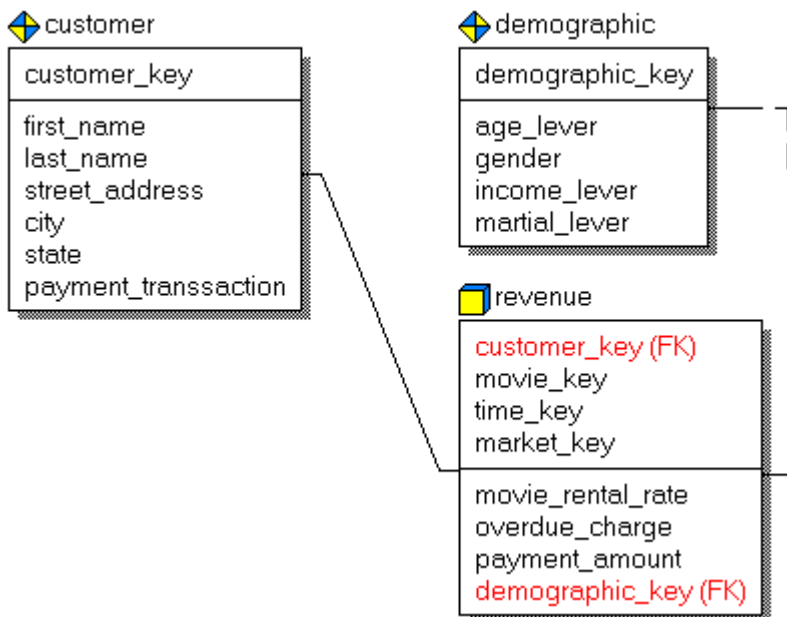
1.2) Outtrigger Tables

Outtrigger Table은 Dimension Table과 관계를 맺는다. 아래 예를 보면 MARKET은 DISTRICT와 REGION Outtrigger Table을 참조하는 Dimension Table이다



1.3) Breaking Up Large Dimensions

비정규화 된 Dimension Table이 많은 로우를 가지고 있고, 쿼리되는 컬럼들이 컬럼의 나머지 부분보다 많은 경우에 Dimension Table을 비식별 관계를 이용하여 2개의 테이블로 분리해야 할 필요성이 있다. 예를 들면, 만약 CUSTOMER Dimension Table이 고개주소와 통계정보를 모두 가지고 있다면 사용자는 나이별, 성별, 수입별로 통계필드를 질의할 수도 있다. 이 경우에 고객테이블 컬럼으로부터 통계 컬럼을 분리하는 것은 쿼리 응답시간을 늦추게 된다. 통계컬럼과 고객컬럼 사이의 관계를 유지하기 위해서 통계키 (Demographic key)는 고객 Dimension Table의 FK가 된다.



2) 스노우플레이크 스키마(Snowflake Schema)

스타 스키마의 변형인 스노우플레이크 스키마(Snowflake Schema)는 차원 테이블을 다시 정규화 (Normalize) 함으로서 이들 테이블들이 마치 눈꽃(Snowflake) 모양이 되도록 만든 스키마이다.

3) Multi Star Schema

Multi Star Schema는 Dimension Table들로부터 복합 FK가 유일한 식별자를 제공하지 않을 때 발생한다. Multi Star Schema에서 Fact Table은 Dimension Table을 참조하는 FK 셋과 각 로우를 유일하게 식별하도록 하는 또 다른 PK를 갖는다

4) Dimensional Modeling With ERwin

Dimensional Modeling은 E-R 모델링과 다른 방향으로 업무데이터를 접근하는 방법이다. E-R 모델은 데이터 무결성과 데이터의 조각을 입력함으로써 데이터 항목의 효율성에 초점을 맞추는 반면에 Dimension Modeling은 업무 프로세스(Business Process)와 업무 질의(Business Question)에 초점을 맞춘다.

데이터웨어하우스는 전형적으로 업무 질의를 답변하기 위해서 설계되었다. 예를 들면, “특정 시장에서 특정 제품에 의해서 생성되는 총 이익은 무엇인가?” 만약 이와 같은 질의에 답변하려면 데이터웨어하우스는 복잡한 쿼리에 빠른 답변을 제공할 필요가 있다. 이러한 필요성을 해결하기 위해서는 Dimension Model을 사용하여 데이터웨어하우스를 설계하는 것이 가장 효과적이다. ERwin은 Dimension Model과 전형적인 ER Model을 지원한다.

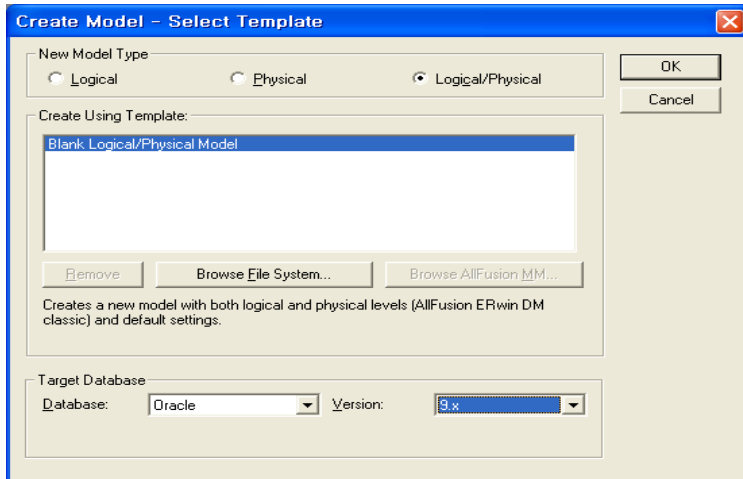
[모델을 작성하기 전에 업무를 분석하라.]

Dimension Model을 생성하기 전에 사용자가 필요로 하는 답변을 위한 질의를 분석한다. 중심 업무 질의를 식별하라. 그리고 프로세스 모델링을 시작한다.

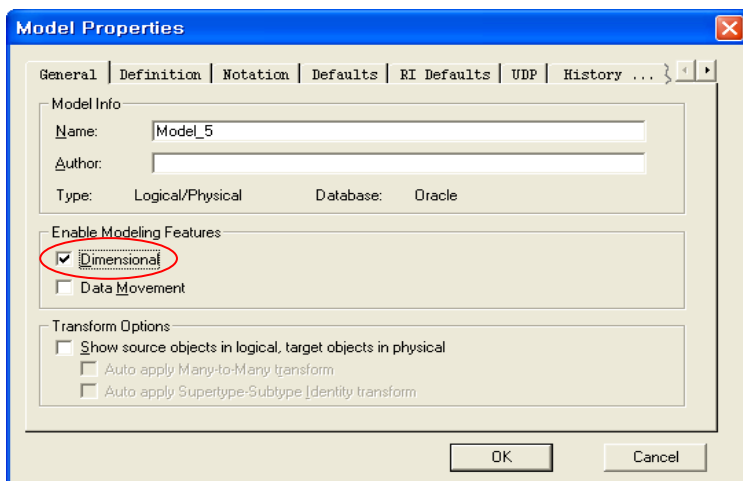
모델의 Fact Table에 넣을 데이터를 식별하기 위해서 업무 질의를 사용하라. 예를 들면, 특정 기간 동안 특정 시장에서 특정상품에 의해서 생산된 총 이익을 보기를 원한다면 “주어진 날에 어떤 시장에서 특정 제품에 의해서 생성된 총 이익을 표현하는 Fact Table에 있는 각 레코드”를 위해서 모델을 설계할 필요가 있다. 이 예제에서 보면 Fact Table은 금전적이고 증가하는 데이터를 포함할 것이다. 그리고 Dimension Table들은 제품, 시장, 시간에 대한 데이터를 포함하게 된다.

만약 기존 ERwin 엔티티 릴레이션 모델로부터 테이블이 컬럼을 재사용하기를 원한다면 그 .er1파일을 이용할 수 있다. 또는 새로운 Diagram을 오픈할 수 있으며 기존 .er1 파일로부터 새로운 Diagram으로 테이블과 컬럼 정보를 복사할 수 있다. Dimensional Diagram으로 각 데이터웨어하우스 컬럼에 대한 데이터 소스를 문서화할 수 있다. 그 소스는 .er1파일, 모델마트 Diagram, 데이터베이스, DDL Script, 또는 플랫폼파일(Flat File)이 될 수 있다. 특정 데이터웨어하우스 소스들은 데이터웨어하우스 데이터와 OLTP 시스템에 있는 데이터 사이에 관계를 문서화 하는 것을 도와준다.

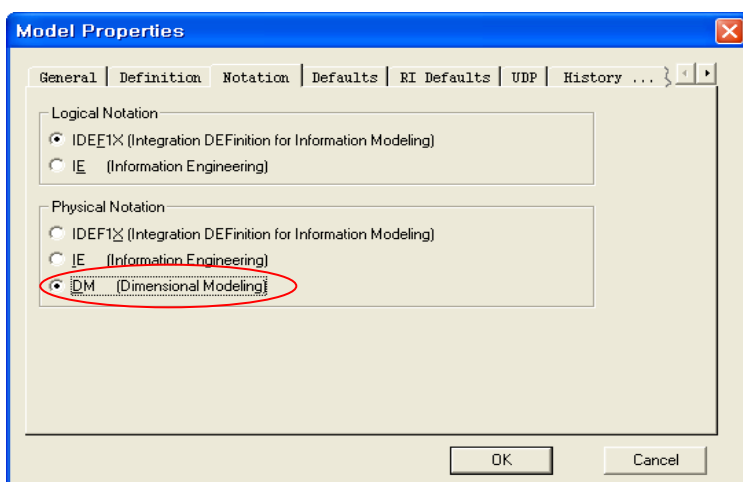
13.2 Dimensional Model 생성



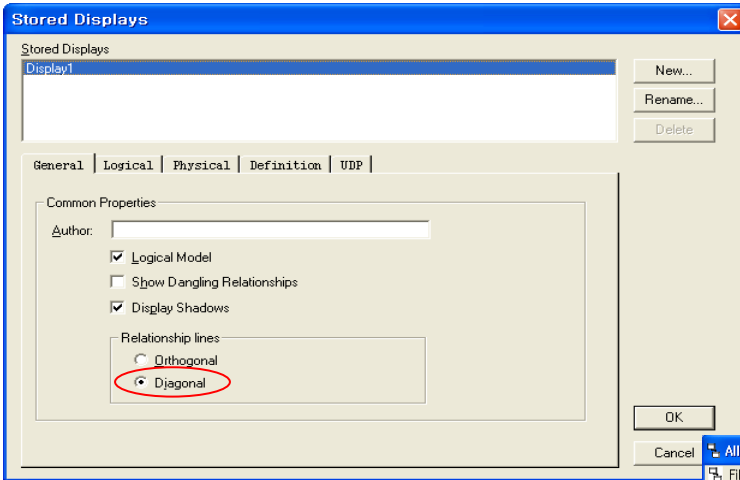
- ① File > New를 선택한다.
- ② Physical 혹은 Logical / Physical을 선택한다.



- ③ Model > Model Properties를 선택한다. General 탭에서 Dimensional 옵션을 체크한다.

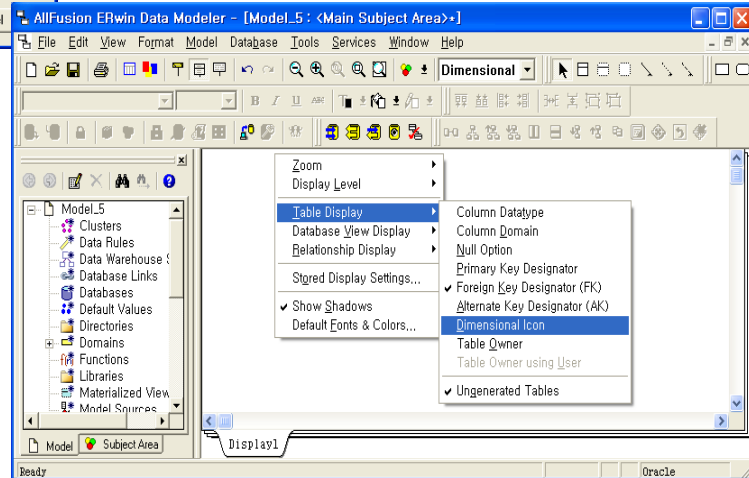





- ④ Notation 탭에서 DM 옵션을 체크하고 OK를 선택한다.



⑤ 백그라운드에서 마우스 오른쪽 버튼을 클릭하여 Stored Display Setting을 선택한다. General 탭에서 Diagonal 옵션을 체크한다. DW 릴레이션쉽(대각선 표현)으로 작성하기 위해서는 이 옵션을 선택해야 한다.

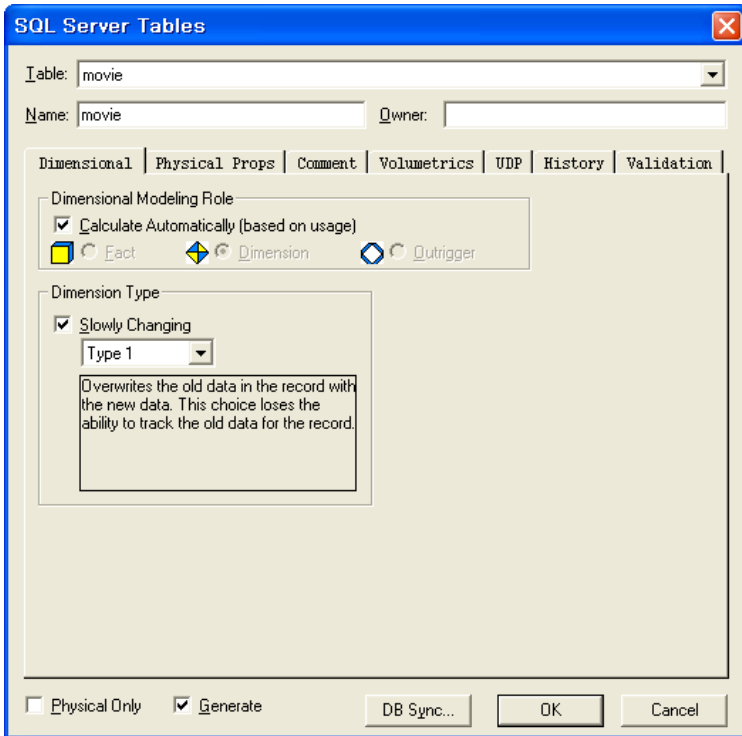
⑥ 백그라운드에서 마우스 오른쪽 버튼을 클릭하여 Table Display > Dimensional Icon을 선택하여 각 테이블 타입을 쉽게 구분 할 수 있도록 한다.



항목	설명
	Fact Table 아이콘
	Dimension Table 아이콘
	Outtrigger Table 아이콘

13.3 Dimension Table 생성

ERwin 툴박스나 테이블 에디터를 이용하여 Dimension Table과 Fact Table을 생성할 수 있다. ERD 작성시 On Screen 상에서 엔티티를 추가하는 방법으로, 그리고 테이블 에디터에서 컬럼을 추가 하는 방법으로 Dimension Table과 Fact Table을 만들 수 있다.



① 테이블을 선택하고 마우스 오른쪽 버튼을 클릭한다.

Dimensional 탭에서 테이블의 Modeling Role과 Dimension Type을 설정 할 수 있다. Dimension Model을 생성하고 테이블을 만들게 되면 기본적으로 Dimension Table로 생성된다. 사용자는 테이블 유형을 수정 할 수 있다.

* 옵션 설명 *

항 목	설 명
Dimensional Modeling	테이블이 유형을 결정한다. (Fact, Dimension, Outtrigger)
Dimension Type	각 Dimension type에 대해서 그것이 고정된 차원(Fixed Dimension) 인지 또는 변화하는 차원(Slow Changing Dimension)인지를 기록할 수 있다.

1) 테이블 설정

ERwin은 Dimensional Modeling Role을 기반으로 하여 한 테이블이 Fact Table, Dimension Table, Outrigger Table인지를 결정한다.



Calculate Automatically에 체크 되어있는 것을 제거하고 원하는 Dimensional Modeling Role을 선택한다. 만약 기존의 ERD를 가지고 Dimensional Modeling으로 바꾸고자 할 때 테이블간의 Relationship이 있다면 ERwin은 페어런트 테이블이 Dimensional Table이 되고 차일드 테이블이 Fact Table이 되도록 한다. Dimensional Modeling Table에 Relationship을 만들어 줌으로써 Outrigger Table을 생성할 수 있다.

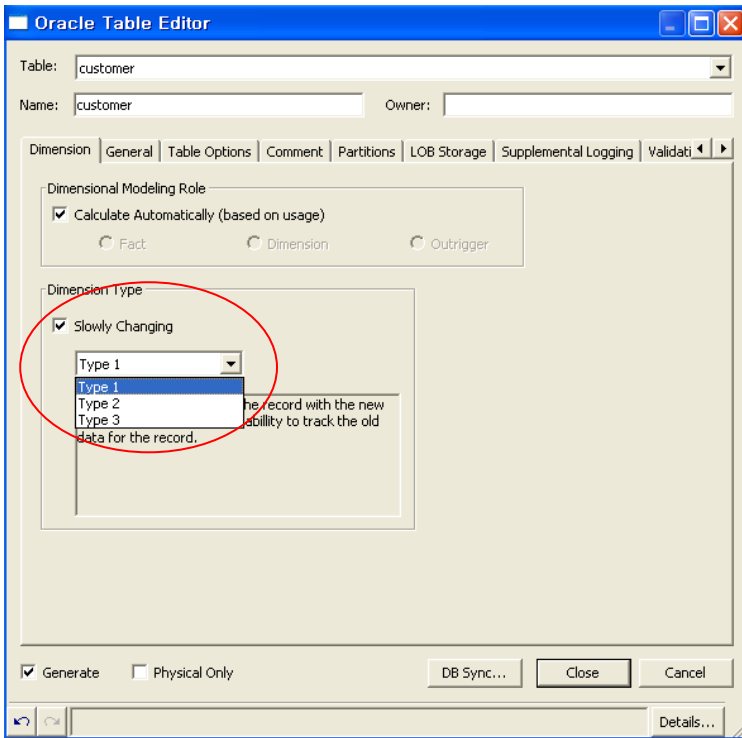
ERwin은 다음과 같은 규칙에 의해 Dimensional Modeling Role을 설정한다.

항 목	각 테이블이 가지고 있는 RelationShip
Fact Role	페어런트 릴레이션쉽이 없는 경우
Dimension Role	어떤 테이블과의 릴레이션쉽이 없는 경우
	Fact Table과 페어런트 릴레이션쉽을 맺고 있는 경우 페어런트와 차일드 릴레이션쉽을 모두 소유 하고 있는 경우
Outrigger Role	Dimensional Table과의 페어런트 릴레이션쉽을 맺고 있는 경우

2) Accounting For Slowly changing Dimensions

Dimensional Modeling 이론에 따르면 Dimensional Table은 정적인 데이터를 포함한다. 그러나 현실세계에서는 시간에 상관없이 고객명, 주소가 변화하고 회사 제품들이 출시되고 있다. 시간과 관련하여 변하는 차원을 Slowly Changing Dimension이라고 부른다.

데이터웨어하우스 환경에서 히스토리컬하게 저장된 데이터의 정확성은 현재 데이터의 정확성 만큼 중요하다. 그래서 레코드에 대한 새로운 정보를 어떻게 기존 데이터에 포함할 것인지를 레포트화 해야 한다. 업무 질의에 대한 차원의 내용을 기반으로 한 정보를 어떻게 수정할 것인지를 결정할 필요가 있다. Dimensional Modeler들은 Dimension 정보를 수정하는 방법을 3가지로 제안한다. 각각의 선택은 정보의 다양한 단계를 제공한다. 테이블 에디터의 Dimensional Tab에서 Slowly Changing Types1,2,3를 선택할 수 있도록 한다.

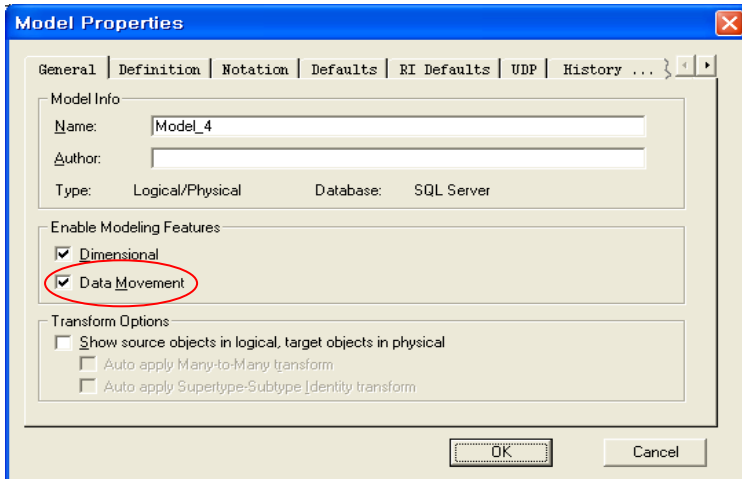


* Slowly Changing Type

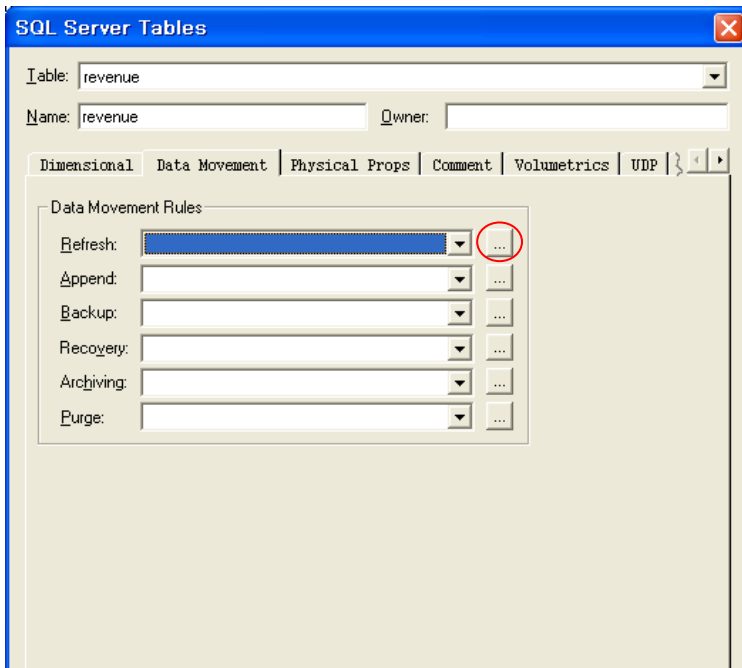
항 목	설 명
Type 1	새로운 데이터를 기존데이터로 덮어 씌운다. 레코드의 기존 데이터를 트래킹 하는 능력을 잃게 된다.
Type 2	새로운 데이터로 변화 시 마다 추가적인 디멘전 레코드를 생성한다. Type1과는 다르게 기존의 데이터와 새로운 데이터를 기반으로 하는 히스토리를 정확하게 트래킹 할 수 있다. 그러나 이것은 기존 레코드의 모든 것을 참조하기 위한 Generalizes key를 생성해 주어야 한다.
Type 3	새로운 데이터로 변화 시 마다 레코드에서 새로운 필드를 생성한다. 기존의 데이터만을 트래킹할 수 있으며 중간 단계의 값은 잃게 된다.

13.4 Data Warehouse Rules 설정

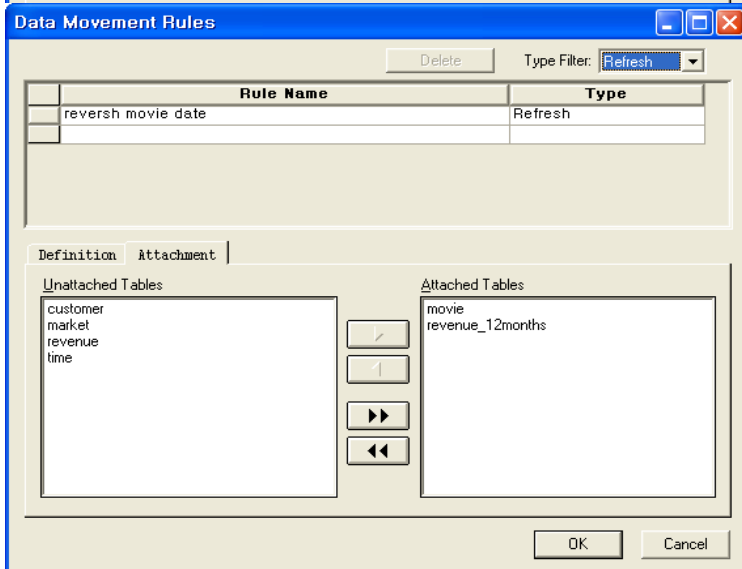
데이터웨어하우스를 지원하기 위해서 ERwin은 모델의 테이블과 연관된 6가지 데이터웨어하우스 룰(Rule)을 지원 한다. (Backup, Recovery, Archiving, Purge, Refresh, Append)



① Data Warehouse Rule을 설정하려면 Model > Model Properties를 선택한다. 그리고 General 탭에서 Data Movement를 체크해야 한다. 그러면 Table Editor에 Data Movement 항목이 추가된다.

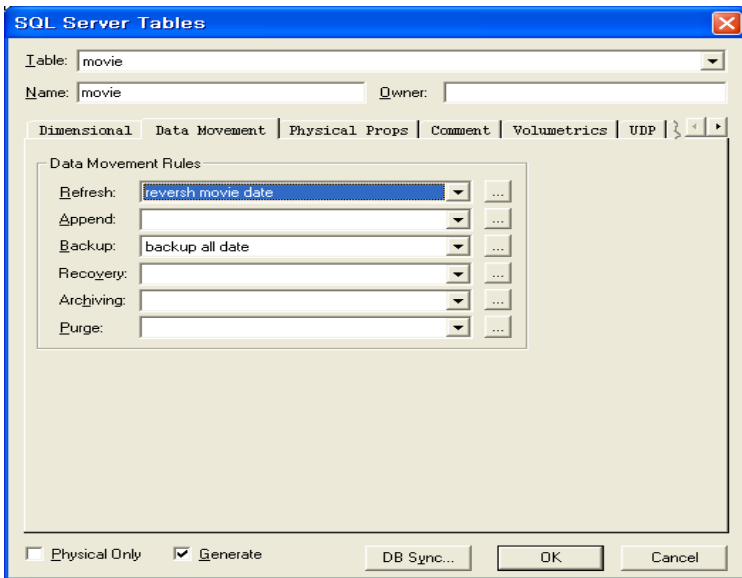


② Data Movement 탭을 선택하고 Rule을 선택하기 위해 확장 버튼을 선택한다.



③ Rule Name을 입력하고 적용하고자 하는 Rule Type을 선택한다. 그리고 Attachment

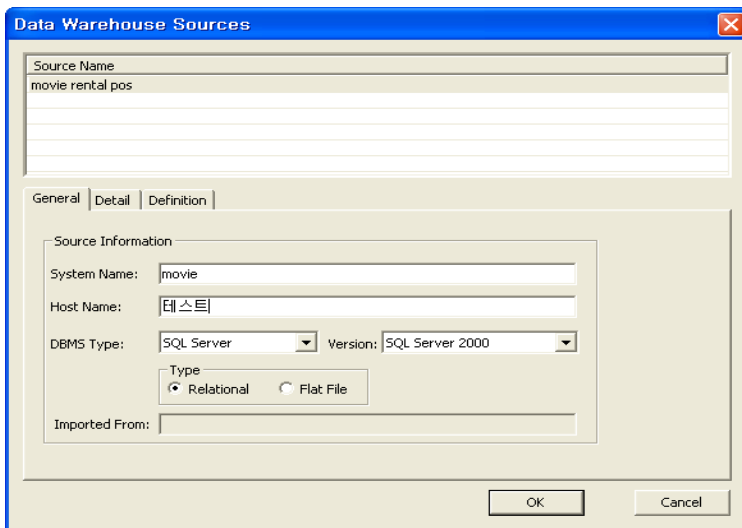
에서 해당 테이블을 첨부하고 OK 버튼 클릭한다.



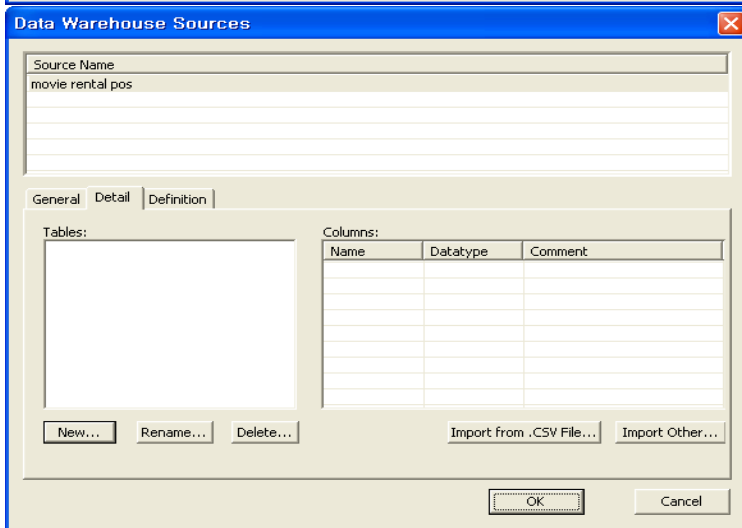
④ 각 테이블을 선택하고 마우스 오른쪽 버튼을 클릭하여 Data Movement로 가면 앞서 설정한 룰이 적용된 것을 확인 할 수 있다.

13.5 Data Source 관리

데이터웨어하우스 환경에서 데이터의 소스를 트래킹 하는 것은 중요한 일이다. 데이터웨어하우스는 아카이브 데이터(Archive Data), 다양한 OLTP시스템, DSS(의사결정지원시스템)으로부터 정보를 가지게 될 것이다. 주로 데이터는 관계형 소스로부터 온 데이터일 것이다. 규칙적인 수정과 데이터량을 체크하기 위해서 데이터웨어하우스에서는 각 컬럼에 대한 소스를 알 필요가 있다. ERwin Data Source Editor에서 이 작업을 할 수 있다.

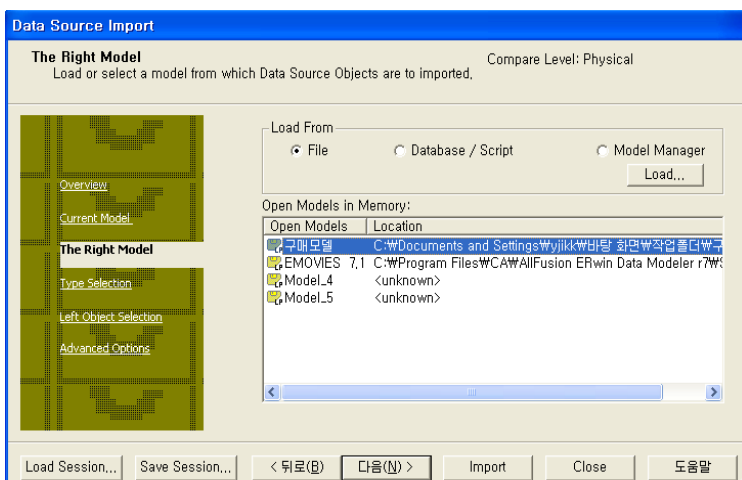


① Model Menu > Data Source를 선택한 후 Source Name을 입력한 후 General Tab에서 Source에 대한 정보를 확인 및 기술 한다.



② Detail 탭을 선택한다. 여기서 Data Source를 생성, 삭제 할 수 있으며, Surce 정보를 Import 할 수 있도록 지원한다.

③ Import Other를 클릭해 Data Source를 입력해보자.

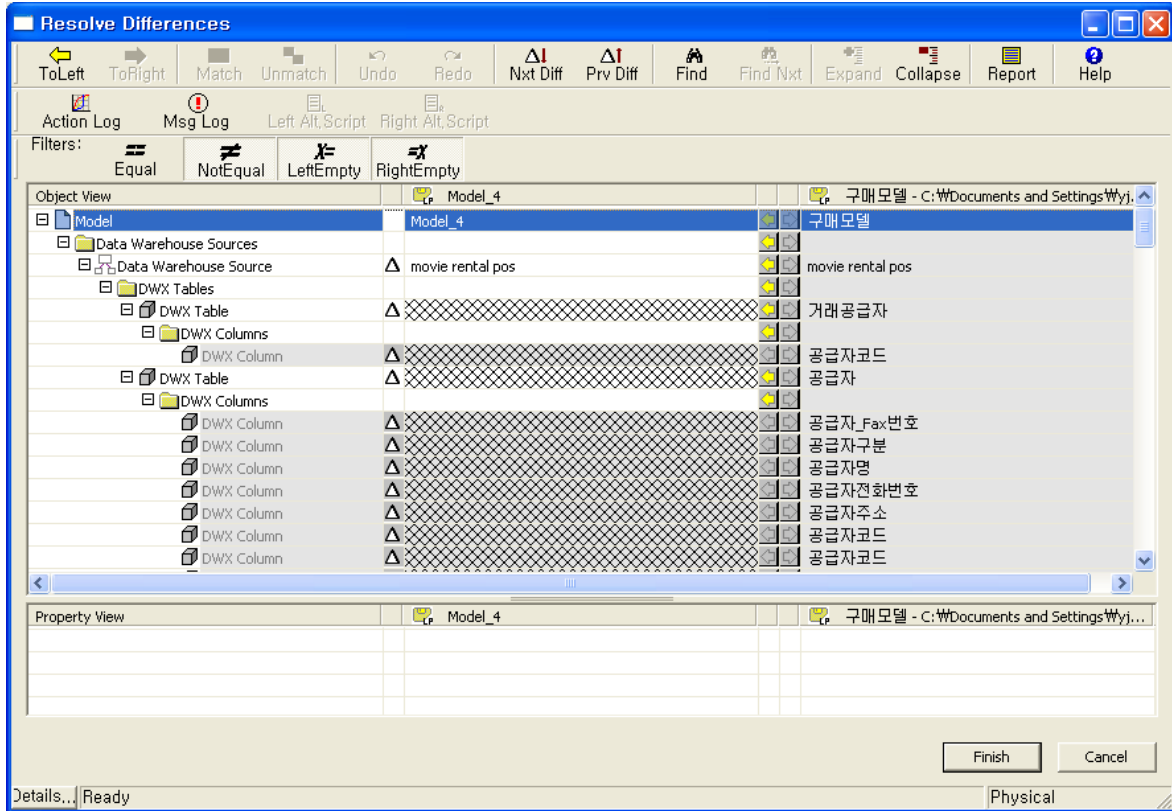



④ The Right Model에서 Import할 대상을 선택한다. Import 대상은 ER1 파일, Database

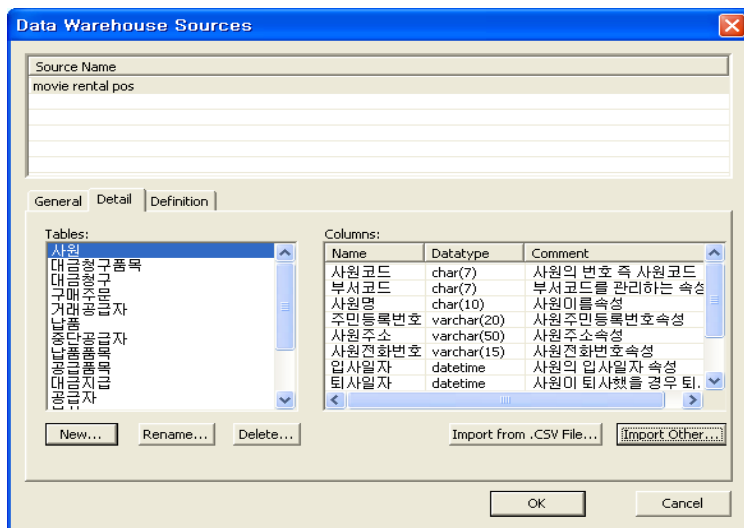
/ Script, Model Manager로부터 Import 할 수 있다.

File을 선택하고 Load를 클릭하여 Import 받을 ER1 파일을 등록한다.

다음 → 다음 → Import 순서로 진행하여 작업을 마친다.



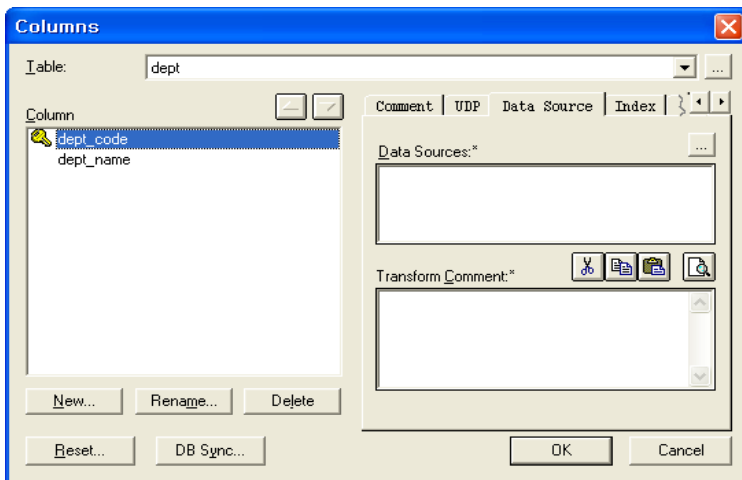
⑤ 우측이 Data Source로 가져올 항목들이다. 여기서 모델을(구매모델) 선택 하고  을 클릭하면 구매모델의 테이블과 컬럼들이 Data Source로 입력된다. 또한 테이블 / 컬럼을 선택해서 Import 할 수 있다. Import를 하고 나서 Finish를 클릭한다.



⑥ 옆의 그림과 같이 구매모델의 Table / Column들이 Data Source로 입력된다.

1) 컬럼에 데이터웨어하우스 소스 설정하기

Import 받은 Dimensional Table의 각 컬럼들의 소스가 어떻게 구성되어 있는지 지정해 줄 수 있다.

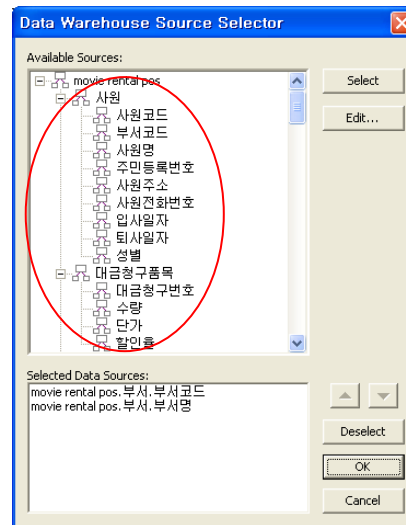


① 컬럼 에디터에서 Data Source 탭을 선택하고 확장 버튼을 클릭한다.

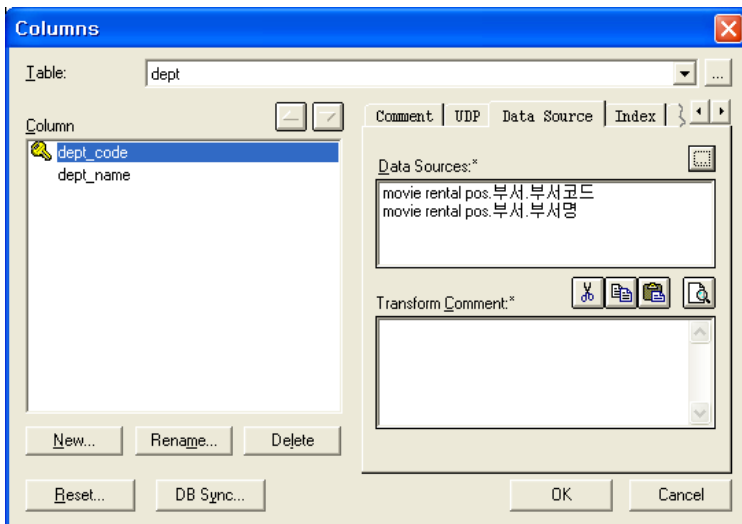
되며 소스로

정의할 컬럼을 선택하고 Select 버튼을 클릭한다.

OK 버튼을 클릭한다. (여러 개 지정 가능)



디스플레이

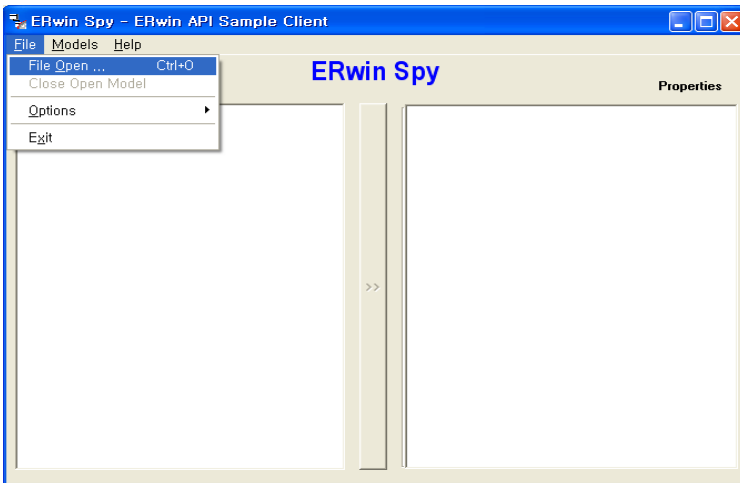


③ 앞에서 선택한 데이터 소스가 화면 우측의 Data Source 창에 입력된다. 데이터 소스는 여러 개 지정 할 수 도 있다.

Transform Comment에는 데이터 소스에 대한 정의를 적어 둔다.

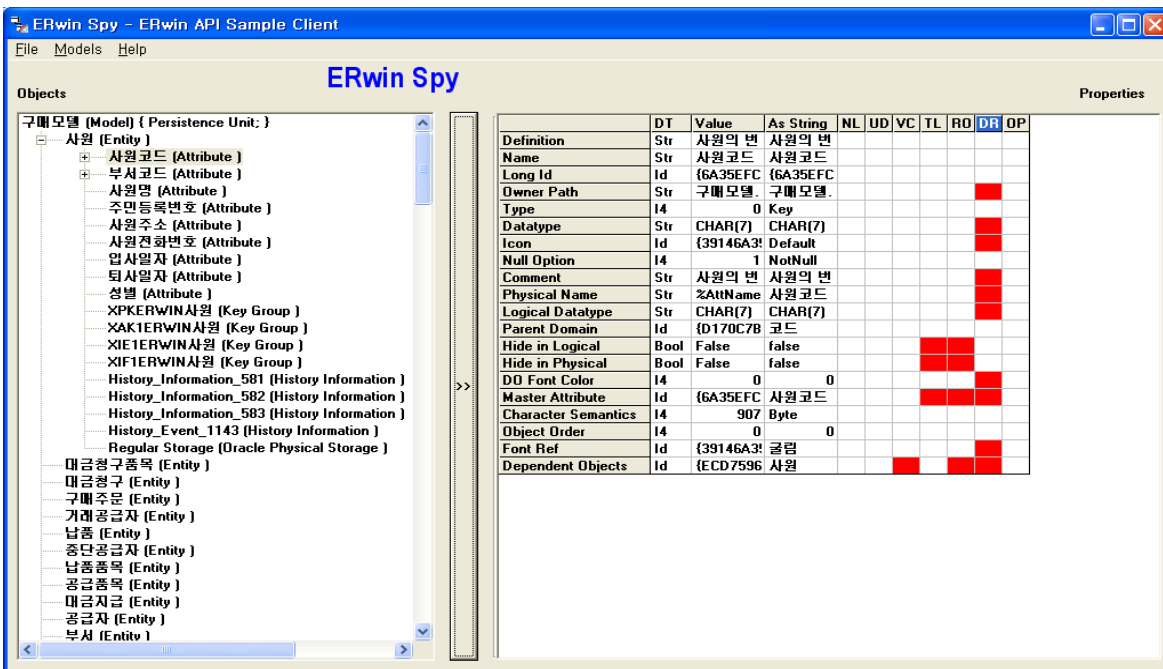
14. SPY Utility

ERwin Data Modeler는 .erwin 파일 즉, 모델의 정보를 분석 할 수 있도록 SPY Utility를 지원한다. ERwin Data Modeler는 API가 공개되어 있으며, SPY Utility는 API를 활용해 만든 것으로 모델의 모든 정보를 계층 구조로 보여준다. 이를 이용해 사용자는 타 Application(VB, C++ 등)을 통해 ERwin API에 접근 할 수 있으며 모델의 다양한 정보를 추출 할 수 있다.



① 시작 > 모든 프로그램 > Computer Associates > AllFusion > ERwin Data Modeler r7 > ERwin DM Spy Utility 를 선택 한다.

② File > File Open을 선택해 ER1 파일을 Open 한다.



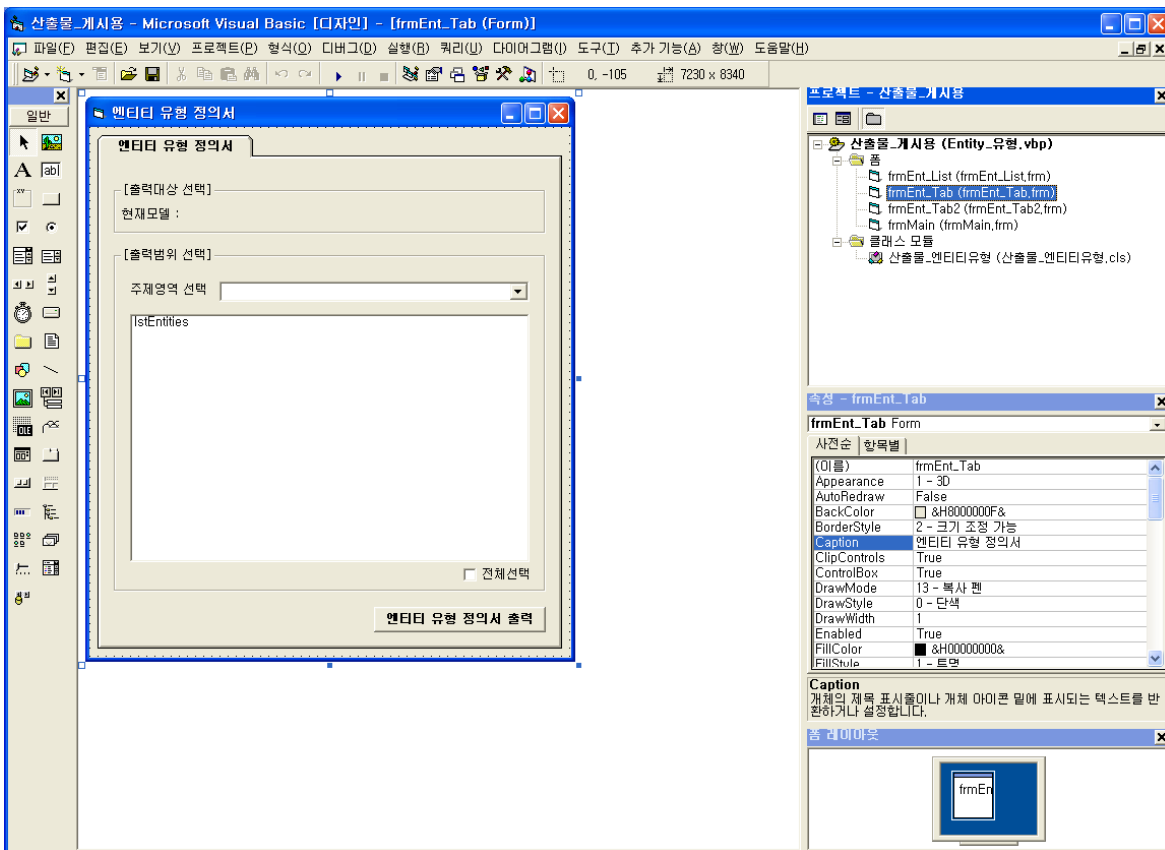
③ 좌측 창을 통해 모델의 정보를 계층 구조로 확인 할 수 있다. 또한 엔티티, 속성 등 여러 Object들의 Detailed 속성을 보고자 할 경우는 해당 Object를 선택하고 >> 버튼을 클릭한다. 그러면 우측 창에 해당 Object의 자세한 정보를 확인 할 수 있다.

15. API 를 이용한 산출물 가이드 (Add ins..)

ERwin은 공개된 API를 이용하여 고객의 요구에 맞는 다양한 산출물을 생성할 수 있다. ERwin에서 기본적으로 지원하는 산출물 도구(Data Browser, Report Builder) 외에 고객의 품에 100% 일치하는 자동 산출물을 API를 통해 모델을 열어놓은 상태에서 고객의 양식 파일에 바로 출력할 수 있는 프로그램을 지원하고 있으며, 자동 입력, 검증 등 필요한 작업에 맞게 얼마든지 수정할 수 있다. 즉, ERwin의 API 정보에서 사용자의 요구에 해당하는 정보를 받아와 사용자가 정의한 산출물 품에 뿌려주게 되는 것이다.

15.1 VB를 활용한 산출물 만들기

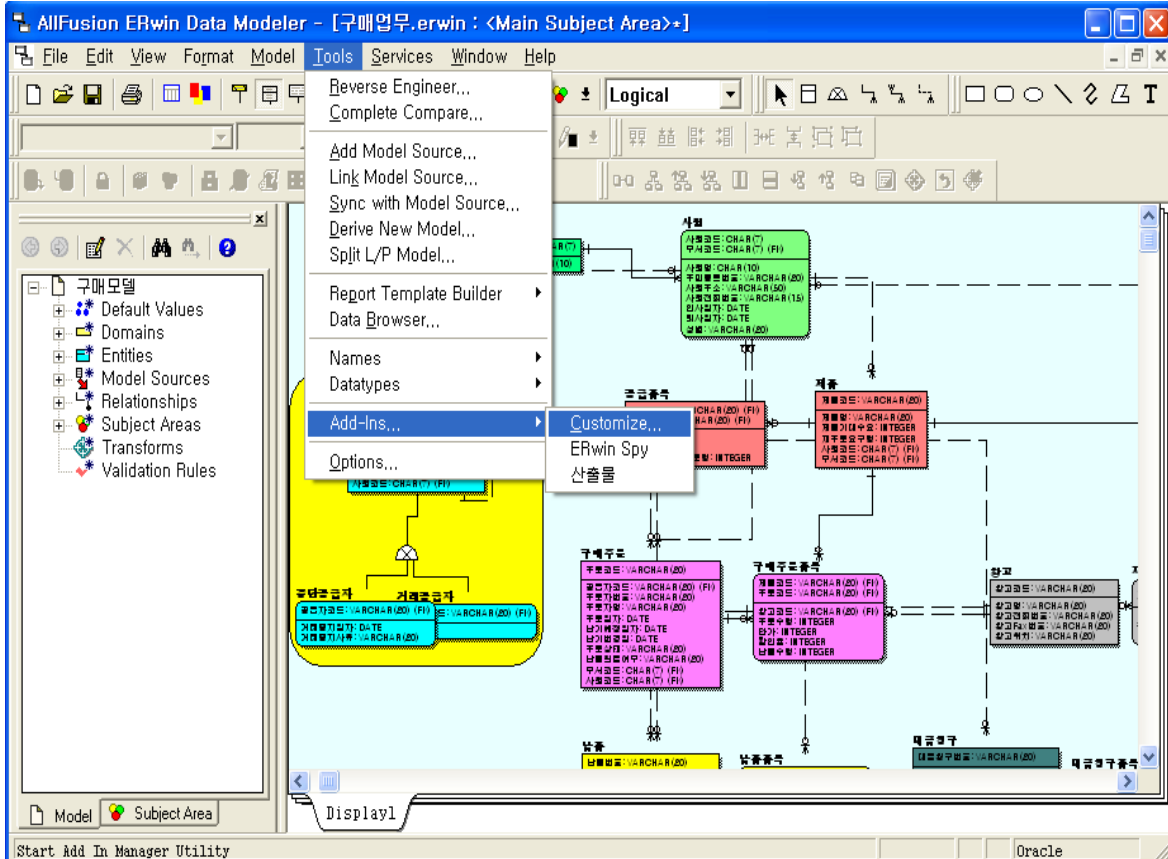
VB를 통해 ERwin API에 접근하여 데이터를 추출 할 수 있다. 이렇게 추출한 데이터를 이용해 Excel, MS-word 등으로 산출물을 자동 생성 할 수 있다. ERwin API를 활용한 산출물 프로그램 예제는 제니스 홈페이지(<http://www.genesis.co.kr>) 게시판에 공개되어 있다.



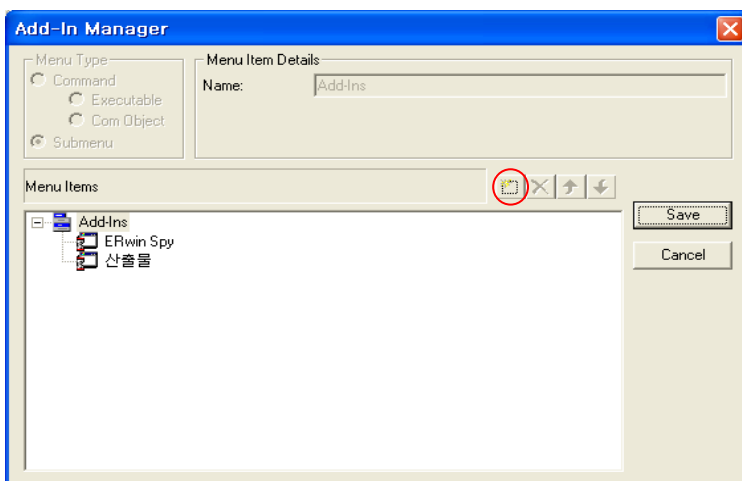
① VB를 통해 산출물 프로그램을 만들고 DLL을 등록한다.


15.2 Add Ins.. 등록

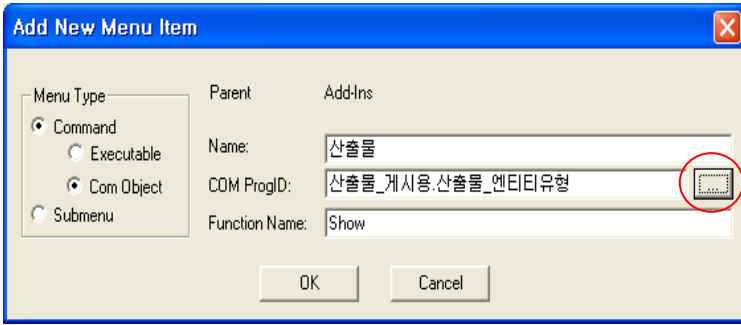
VB를 통해 작성한 프로그램을 Add-Ins.. 에 등록하여 사용하게 된다.



① 모델을 열고 Tools > Add-Ins.. > Customize를 선택한다.

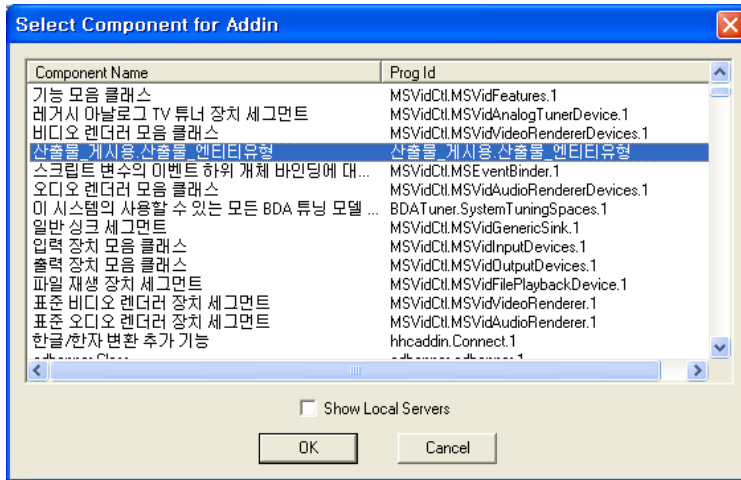


② Menu Items을 추가하기 위해  버튼을 클릭한다.

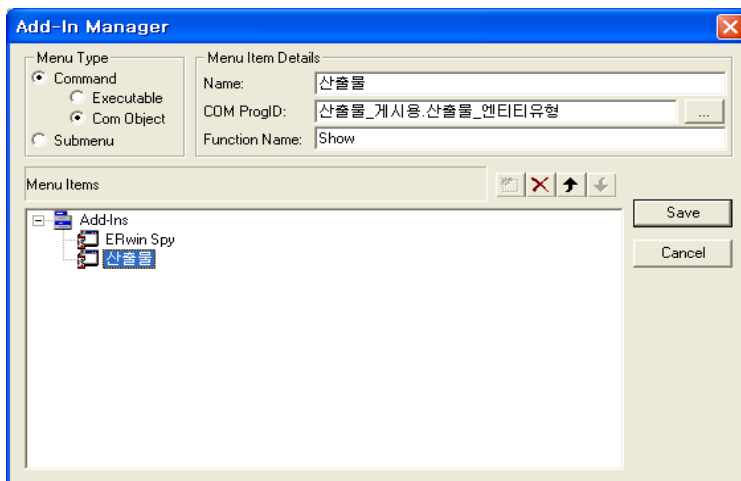


③ Name을 입력한다.

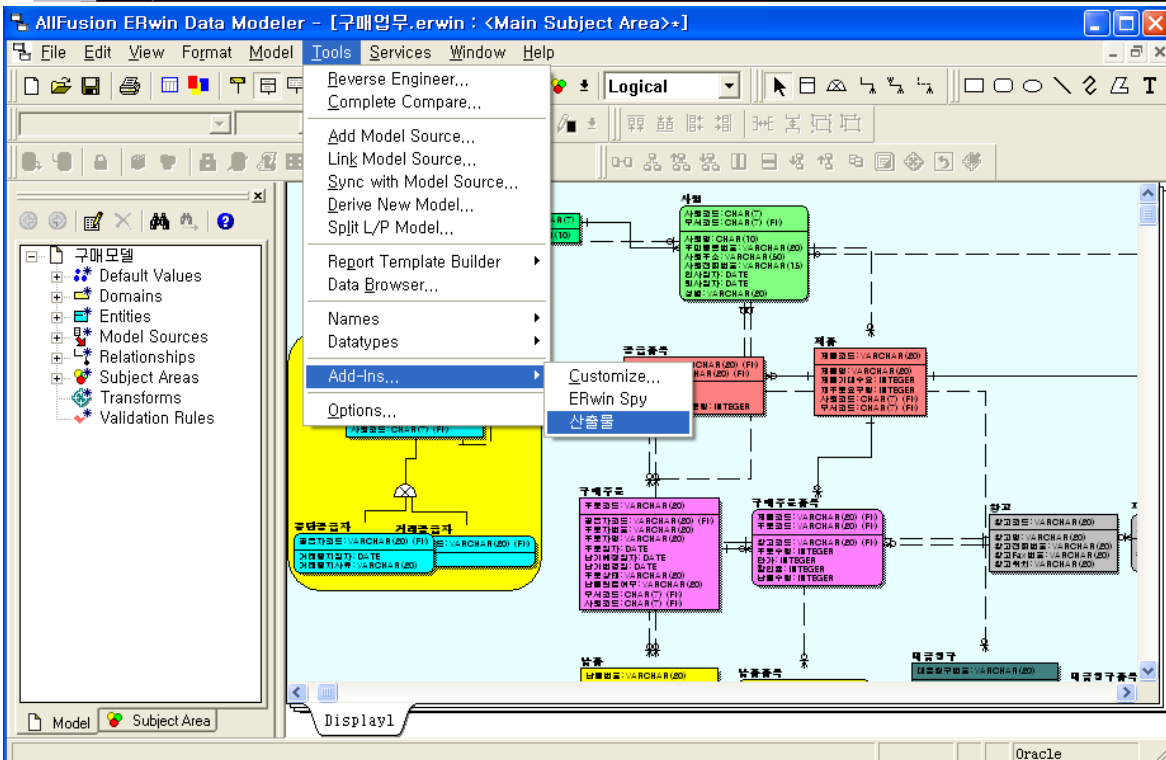
④ COM ProgID에 DLL을 등록 하기 위해 확장 버튼을 클릭한다.



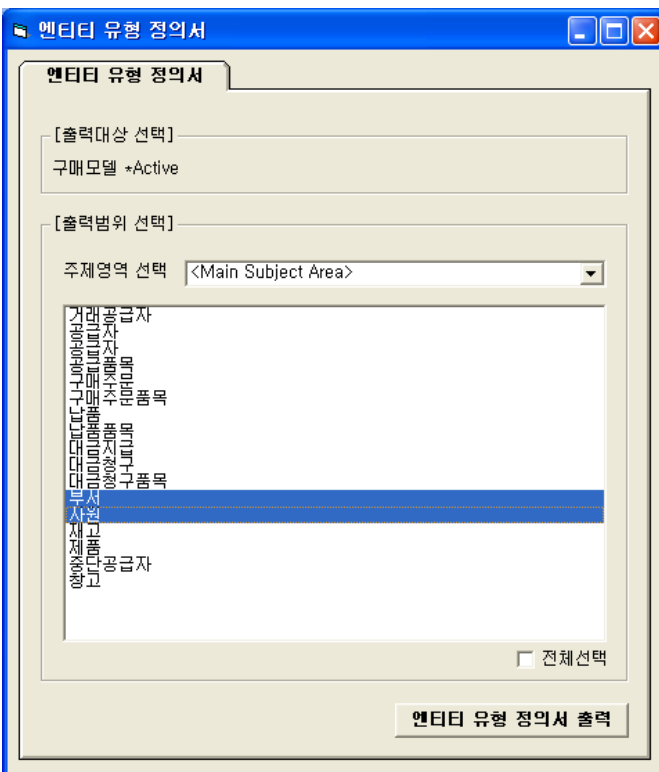
⑤ 해당 Component Name을 선택하고 OK 버튼을 클릭한다.



⑥ 그림과 같이 Add-Ins에 해당 프로그램이 등록된다. Save를 클릭한다.

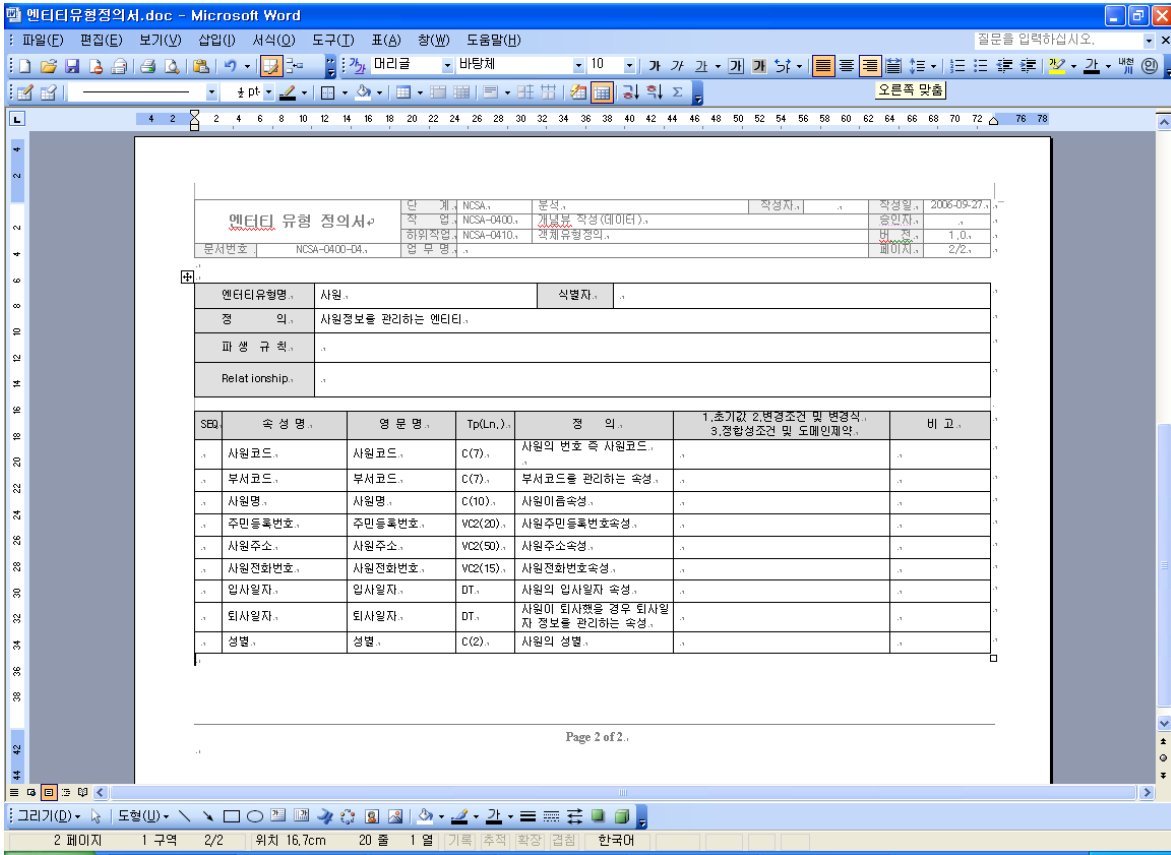


⑦ Tools > Add-Ins.. > 산출물을 선택한다.



⑧ 출력 하고자 하는 엔티티를 선택하고 출력 버튼을 클릭한다.

모델에 주제 영역이 있을 경우 주제 영역을 선택하면 엔티티 리스트를 확인할 수 있다.



⑨ 그림과 같이 산출물이 자동으로 출력된다.

16. 변경된 내용에 대한 가이드 모음

16.1 Know Issues

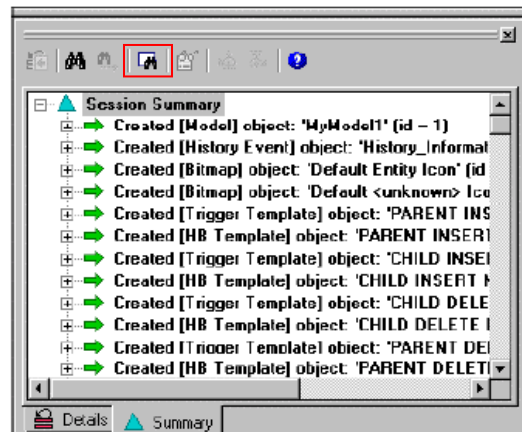
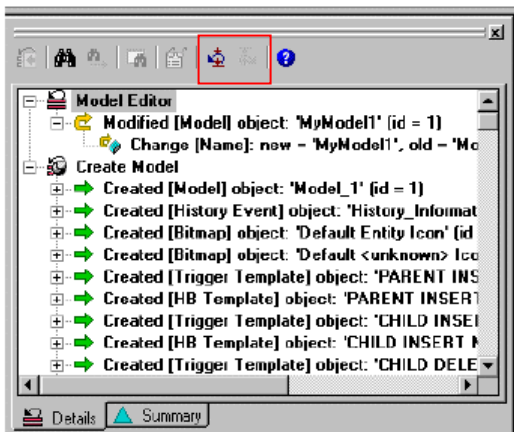
SQL Server 2000 Client 가 설치되어 있지 않은 컴퓨터에서 SQL Server 2005 연결을 시도한다면, AllFusion ERwin Data Modeler나 AllFusion Model Manager에서는 연결할 수 없다. AllFusion ERwin Data Modeler나 AllFusion Model Manager에서 SQL Server Databases로 연결을 위해서는 DB-Library가 요구된다. DB-Library는 SQL Server 2005의 연결을 위해 필요한데 SQL Server 2005는 더 이상의 DB-Library파일을 배포하지 않는다.

SQL Server 2005 에 연결을 위하여서는, NTWDBLIB.DLL 파일이 필요하다. SQL Server 2000 Installation CD 가 있는 경우에, SQL Server 2000 클라이언트는 CD에서 ₩ x86 ₩ 아래의 다른 폴더 안에 있는 sqlredis.exe를 수행해서 설치될 수 있다. SQL Server 2000 installation CD가 없는 경우에, 마이크로소프트 지원을 필요로 한다.

16.2 Action Log

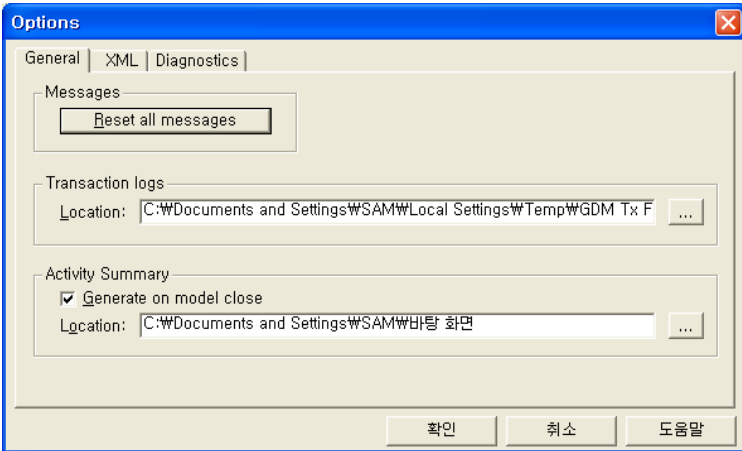
Action Log는 실시간으로 모델링작업에서 일어나는 일들을 기록하고 Details와 Summary 탭을 통해서 작업 내용을 확인 할 수 있다.

작업내용에 대한 기록은 모델을 저장하고 닫을 때 까지 기록을 하게 되며 Logical 쪽의 Action Log는 Logical에 대한 정보를 Physical 쪽의 Action Log는 Physical에 대한 정보만을 보여준다.



Details Pane에서는 각각의 상황을 선택하여 Undo와 Redo동작을 할 수 있는 반면에 Summary Pane에서는 Undo와 Redo동작을 할 수 없다. 하지만 Find/Filter동작을 실행해 작업 상황에 대한 상세내용을 filter하고 리포트를 할 수 있다.

또한 Action Log에 대한 내용을 엑셀파일로 저장해서 보관을 할 수 있다. 저장방법은 Tools에 Option 메뉴를 클릭하면 다음과 같은 화면이 나타나게 된다.



이 화면에서 Activity Summary영역에 Generate on model close에 체크를 한 후 원하는 경로를 설정해 주면 모델을 닫을 때 해당경로에 Action Log에 대한 정보 파일이 엑셀(csv)파일로 저장 된다.

16.3 함수기반 인덱스(Function Based Index)

SQL 문장의 WHERE 절에 SQL함수(SQL Function)가 사용되는 경우 인덱스 컬럼에 변형이 생기는 경우이므로 인덱스를 사용못하는 경우가 있다. 이러한 경우를 해결할 수 있는 것이 함수 기반 인덱스인데 ERwin R7 버전부터 이러한 인덱스를 구현 할 수 있도록 지원하고 있다.

사원

사원코드
사원명
사원주소
사원전화번호
사원이메일
사원성별

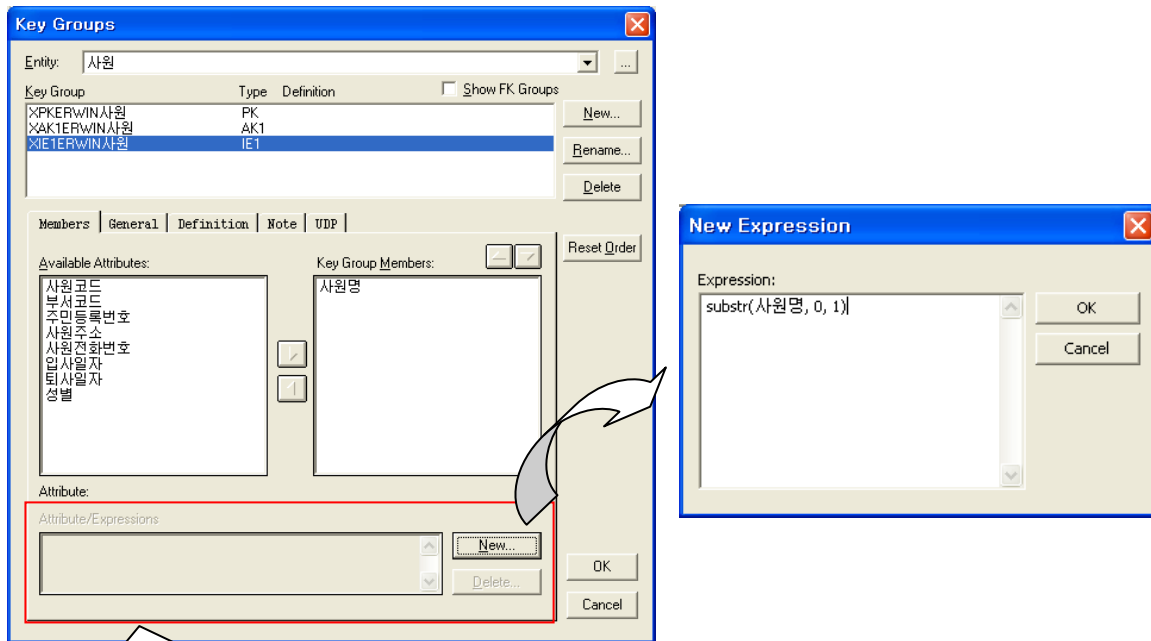
위와 같은 사원테이블에서 사원명에 인덱스가 걸려 있다고 가정하면

```
SQL> SELECT 사원명, 사원전화번호
FROM 사원
WHERE SUBSTR(사원명,0,1) = '김';
```

물론 위의 경우 WHERE ENAME LIKE '김%' 라고 하는 것이 바람직하겠지만 인덱스 컬럼에 대해 변형을 가하는 예문이라고 생각을 하고, 위 경우처럼 인덱스가 존재하는 컬럼에 대해 SQL함수를 사용하게 되면 인덱스 컬럼에 변형이 생겨 사원명에 대해 생성되어 있는 인덱스를 사용하지 못하게 된다. 이럴 때 함수 기반 인덱스(Function Based Index)를 사용하면 해결할 수 있다. 다음처럼 인덱스를 만들면 된다. 또한 인덱스가 생성될 컬럼에는 SQL함수나 산술식, 상수, 사용자 정의 함수를 적용할 수 있다.

SQL> create XIE1_사원 on 사원(substr(사원명, 0, 1));

ERwin에서 적용하는 방법은 각 엔티티/테이블을 클릭한 후 KeyGroup/Index Property에 들어가면 ERwin4.1과 바뀐 내용을 볼 수 가 있을 것이다.



새롭게 Index에 대한 식을 넣을 수 있게 추가된 부분

Expressions에 대한 속성은 Inversion Entry 나 Alternate Key를 선택했을 경우만 활성화 되며 New버튼을 클릭하면 표현식을 입력 할 수 있는 창이 활성화 된다. 이 창을 통해 조건에 맞는 식을 정의해 입력하면 된다.

Tip

이 기능은 Target DB를 어떠한 것을 선택 하느냐에 따라 달라진다. 즉 DB가 이러한 함수 기반 인덱스를 지원 해야지만 사용할 수 있다.

16.4 Metadata Integration Wizard

ERwin 4.1버전에서 R7으로 업데이트 되면서 추가된 기능 중 가장 강력한 기능 중에 하나이다.

타 제품의 여러 모델을 Metadata Integration Wizard 통해 R7에서 그대로 가져오거나 다른 제품의 모델로 내보낼 수 있다.

약 100여 개의 Metadata 자료를 가져올 수 있는데 File 메뉴에 Import에 From External Format wizard와 Export에 To External Format Wizard 사용해 수행 할 수 있다.

다음은 현재 AllFusion ERwin DM 에서 사용할 수 있는 Meta Integration Technology, Inc를 나타낸 것 이다.

Application	Mode
Adaptive Repository & Foundation (CWM)	Import/Export
AllFusion Component Modeler 4.x (XMI)	Import
AllFusion ERwin DM 4.x	Import/Export
AllFusion ERwin DM r7	Import/Export
AllFusion Gen 4.1a to 7.5	Import/Export
AllFusion Repository DS – ODBC	Import/Export
Altova XMLSpy 2004 to current	Export
ArgoUML (XMI)	Import/Export
Ascential DataStage	Import/Export
Borland Together UML (XMI)	Import/Export
BusinessObjects Crystal Reports 8.5 to 9.0	Import
BusinessObjects Data Integrator (CWM)	Import/Export
BusinessObjects Designer (File)	Import/Export
BusinessObjects Designer (Repository)	Import
BusinessObjects Reporter (File)	Import
BusinessObjects Reporter (Repository)	Import
BusinessObjects WebIntelligence 11.x	Import
Cobol Copybook Flat Files	Import
Cognos 8 Framework Manager (File)	Import/Export
Cognos Impromptu	Import
Cognos ReportNet Framework Manager (File)	Import/Export
Cognos ReportNet Model (Content Manager)	Import
Cognos ReportNet ReportStudio (Content Manager)	Import
Cognos ReportNet ReportStudio (File))	Import

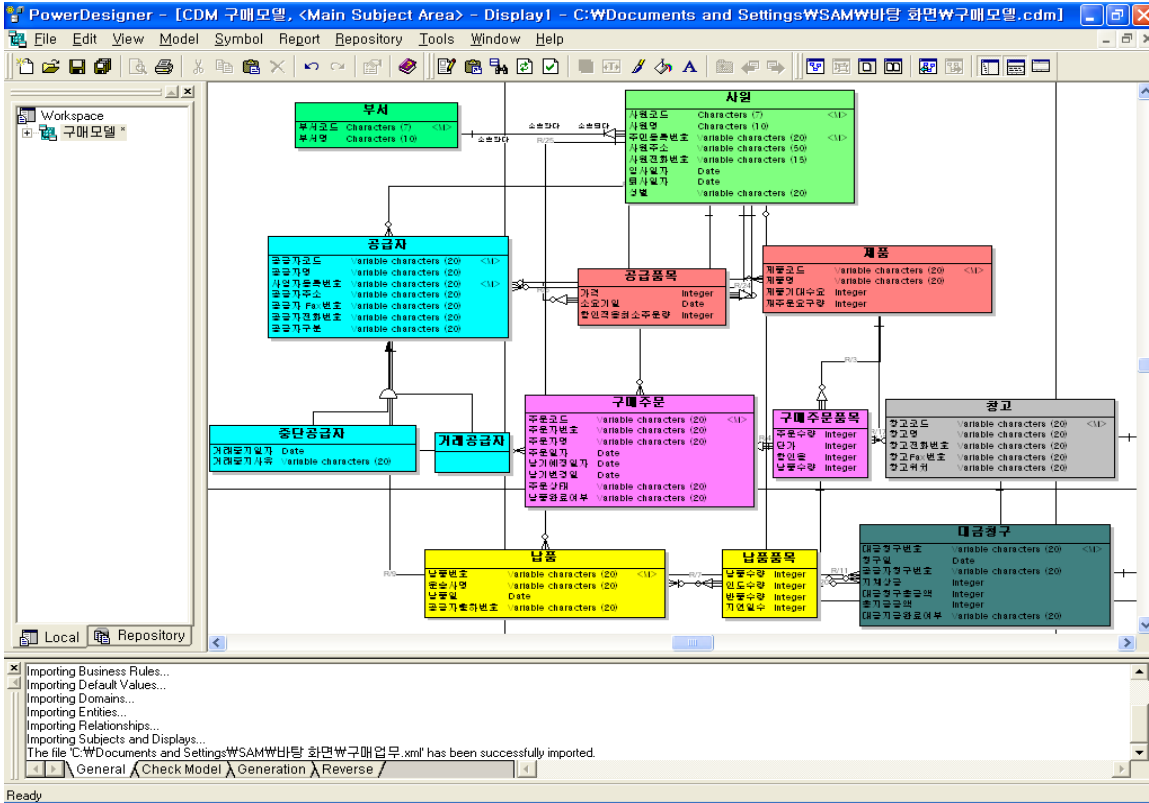
COOL: Biz 5.1	Import
COOL: BusinessTeam (GroundWorks) 2.2.1	Import
COOL: DBA (Terrain for DB2) 5.3.2	Import
COOL: Enterprise (ADW) 2.7	Import
COOL: Xtras Mapper (TerrainMap for DB2)	Import
Embarcadero ER/Studio 5.1 to 7.0	Import
Embarcadero ER/Studio (ERX)	Export
ERwin 3.x (ERX)	Import/Export
Gentleware Poseidon UML (XMI)	Import/Export
Hummingbird ETL/Genio 5.04	Import/Export
Hyperion Application Builder (CWM)	Import/Export
Hyperion Essbase Integration Services 7.0	Import/Export
IBM DB2 Cube Views	Import/Export
IBM DB2 OLAP Integration Server 8.1	Import/Export
IBM DB2 Warehouse Manager (CWM)	Import/Export
IBM Rational Rose 4.0 (MDL)	Import/Export
IBM Rational Rose 98(i) to 2000 (MDL)	Import/Export
IBM Rational Rose 2000e to 2003 (MDL)	Import/Export
IBM Rational Software Architect (MDL)	Export
IBM Rational Software Modeler (MDL)	Export
IBM Rational XDE 2002 (MDL)	Export
IBM Rose XMI Toolkit 1.0 & 1.05 & 1.15 (XMI)	Import
IBM VisualAge for Java 3.0 (XMI)	Import/Export
IBM WebSphere DataStage	Import/Export
IBM WebSphere Studio 3.0 (XMI)	Import/Export
IBM XMI Toolkit 1.0 & 1.05 & 1.15 (XMI)	Export
Informatica PowerAnalyzer	Export
Informatica PowerCenter	Export
Informatica PowerCenter (File)	Import
Informatica PowerCenter (Repository)	Import
Informatica SuperGlue	Export
Merant App Master Designer 4.0	Import
Meta Integration Metadata (MIM)	Import/Export
Meta Integration Metadata (XMI/XML)	Import/Export

Meta Integration Repository (MIR) on Access	Import/Export
Meta Integration Repository (MIR) using ODBC	Import/Export
MetaMatrix MetaBase Modeler (MIR XMI)	Export
Microsoft Excel	Import/Export
Microsoft Office Access	Export
Microsoft Repository 2.1b (XIF)	Import/Export
Microsoft Repository 3.x (MDC)	Import/Export
Microsoft SQLServer 2000 Analysis Services (ODBO)	Import
Microsoft SQLServer 2005 Data Source View	Import/Export
Microsoft SQLServer 7.0 to 2005 Analysis Services (DSO)	Import
Microsoft Visio Database (ERX)	Import/Export
Microsoft Visio UML (XMI)	Import
Microsoft Visual Studio/Modeler 2.0 (MDL)	Import/Export
MicroStrategy 7.0 to 8.0	Import/Export
MicroStrategy MDX Adapter (ODBO)	Import
NCR Teradata MDS 5.0 to 6.0	Import/Export
NoMagic MagicDraw UML (XMI)	Import/Export
OLAP Database using ODBO	Import
OMG CWM 1.0 and 1.1 XMI 1.1	Import/Export
OMG CWM Pre-1.0 XMI 1.1	Import/Export
OMG UML 1.1 & 1.3 XMI 1.0 & 1.1	Export
OMG UML 1.1 to 1.4 XMI 1.x	Import
Oracle Designer 1.3.2, 2.1.2, 6.0, 6i & 9i	Import
Oracle Designer 6i & 9i	Export
Oracle Warehouse Builder (CWM)	Import/Export
ParadigmPlus 3.52	Import/Export
Popkin System Architect 7.1.12 to current	Import/Export
ProActivity 3.x & 4.0	Import
Relational Database using JDBC	Import
SAP Business Warehouse (ODBO)	Import
SAS ETL Studio (CWM)	Import/Export
SAS OLAP Server (ODBO)	Import
Select SE 7.0	Import
Silverrun-RDM 2.4.4 to 2.7.2	Import

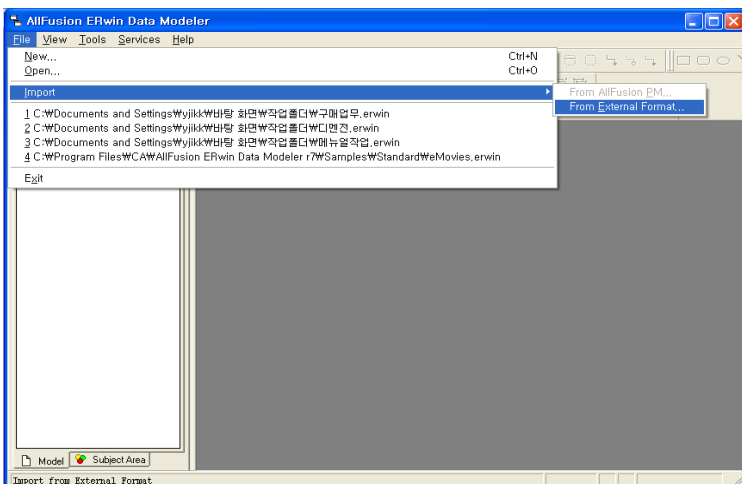
Sybase PowerDesigner CDM 6.1.x	Import
Sybase PowerDesigner CDM 7.0 (ERX)	Export
Sybase PowerDesigner CDM 7.2 to 12.0	Export
Sybase PowerDesigner CDM 7.5 to current	Import
Sybase PowerDesigner OOM 9.0 to current (XMI)	Import/Export
Sybase PowerDesigner PDM 6.1.x	Import
Sybase PowerDesigner PDM 7.5 to 12.0	Export
Sybase PowerDesigner PDM 7.5 to current	Import
Sybase PowerDesigner XSM 10.0 to current	Export
Telelogic System Architect 7.1.12 to current	Import/Export
Telelogic Tau UML (XMI)	Import/Export
Unisys Rose UML utility 1.1 (XMI)	Import/Export
Unisys Rose XMI Interchange (XMI)	Import/Export
Visible IE:Advantage 6.1	Import
XML DTD 1.0 (W3C)	Import
XML Schema (Microsoft XDR)	Import
XML Schema 1.0 (W3C XSD)	Import/Export

16.4.1 Import / Export 샘플

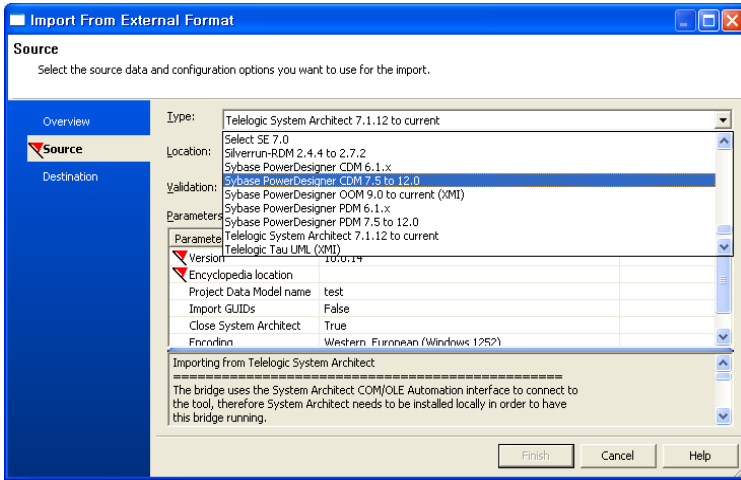
1) Power Designer 12.0



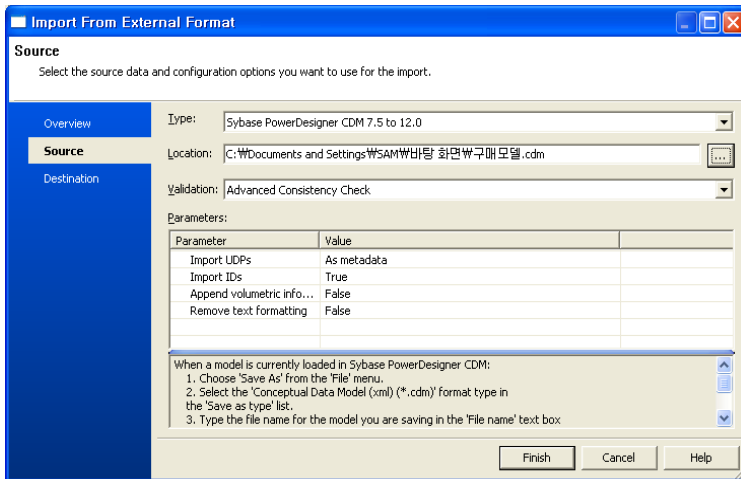
위와 같은 Power Designer 파일 CDM을 ERwin R7을 통해 Import 해보자.



① File -> Import -> From External Format 을 선택한다.

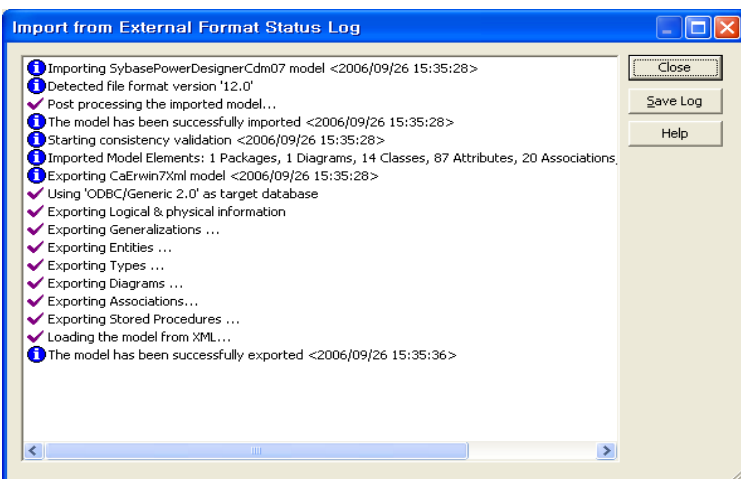


② Source 화면에서 Import Type을 Power Designer로 선택한다.



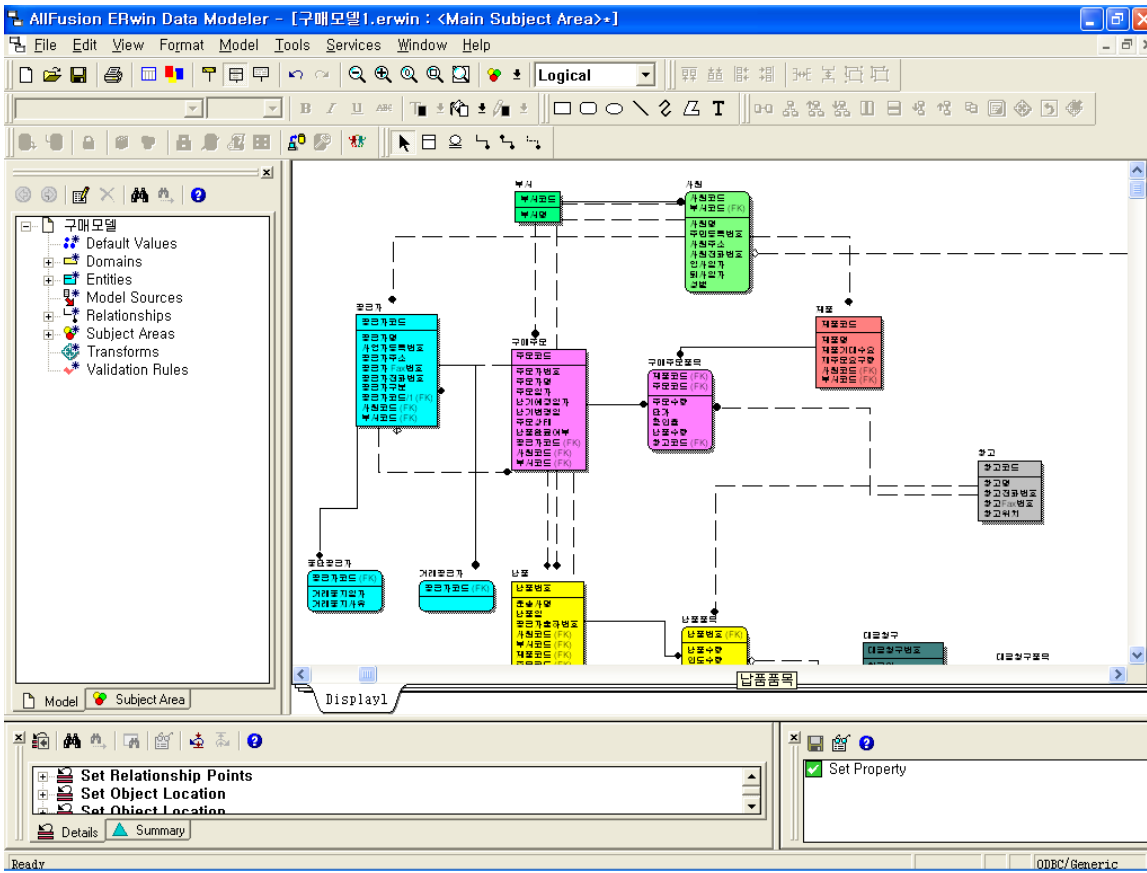
③ Location에서 파워 디자이너의 CDM 파일의 경로를 지정한다.

④ Finish 버튼을 선택한다.



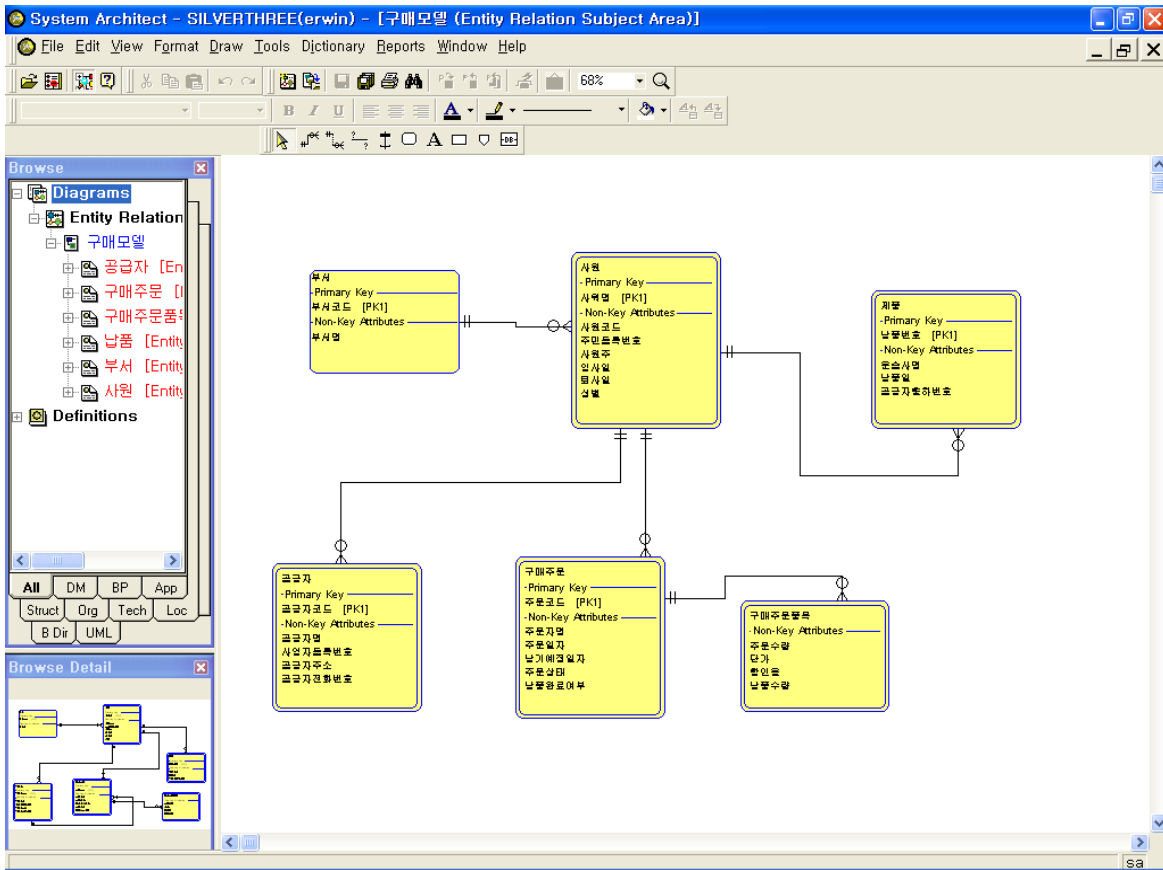
⑤ 로그 파일을 통해 변환 과정과 오류 발생 시 오류에 대한 내용을 확인 할 수 있다.

⑥ Close 버튼을 선택한다.

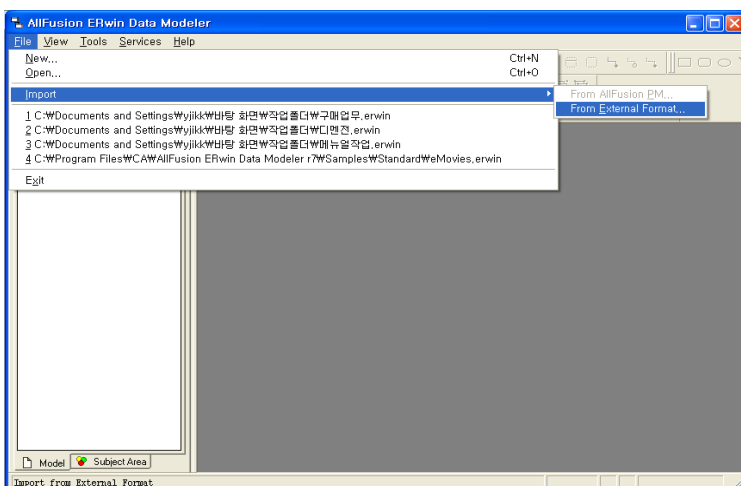


[CDM 모델을 ERwin으로 Conversion 한 화면]

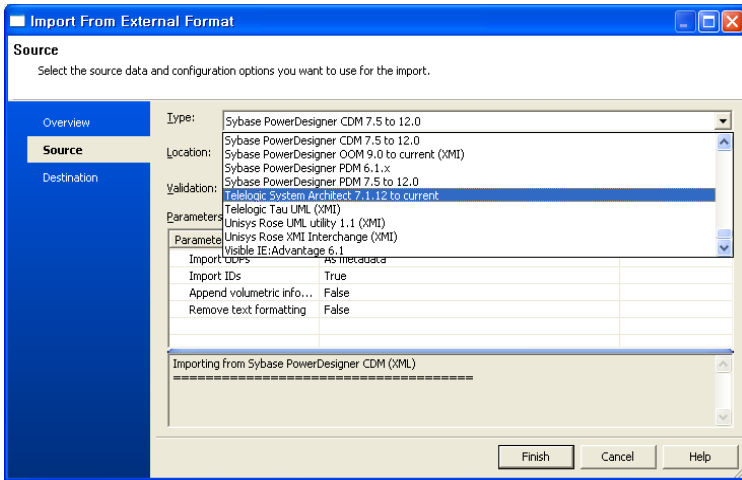
2) System Architect(SA)



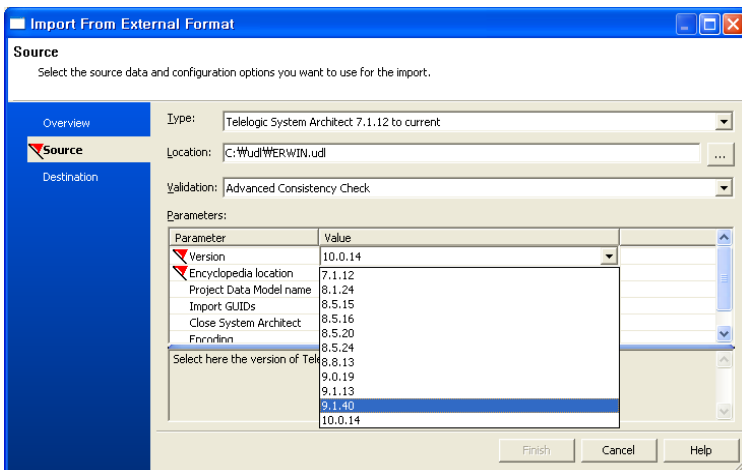
SA의 Encyclopedia에 저장되어 있는 Data Modeling 내용을 ERwin으로 Import 해보자.



① File > Import > From External Format을 선택한다.

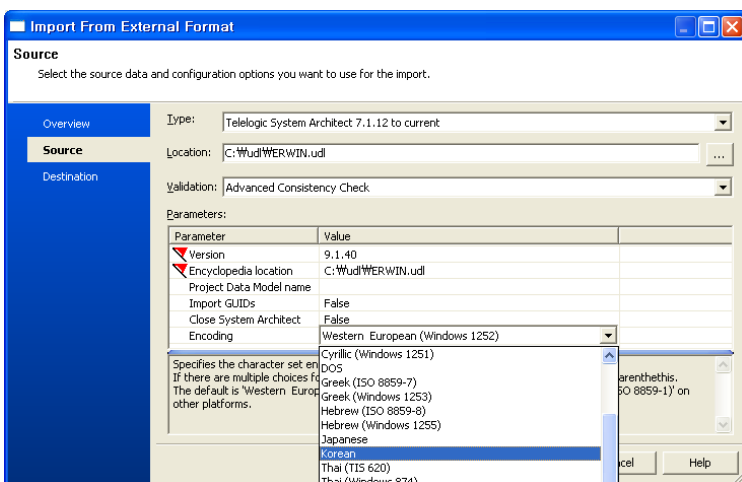


② Source 화면에서 Import Type을 SA로 선택한다.



③ Location에서 SA의 확장자인 UDL 파일의 경로를 지정한다.

④ Version을 선택한다.



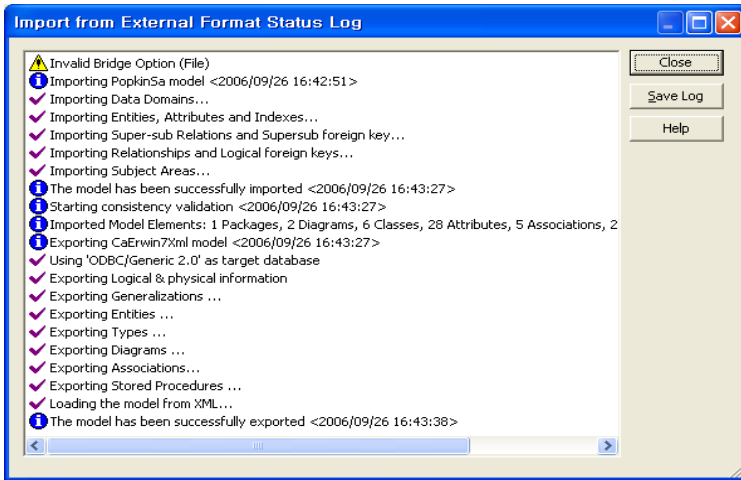
⑤ Encyclopedia Location에서 Encyclopedia의 경로를 지정한다.

⑥ Project Data Model Name을 입력한다. (한글로 입력 안됨)

⑦ Encoding을 Korean으로 선택 해야, 한글 모델을 정상적으로 불러 올 수 있다.

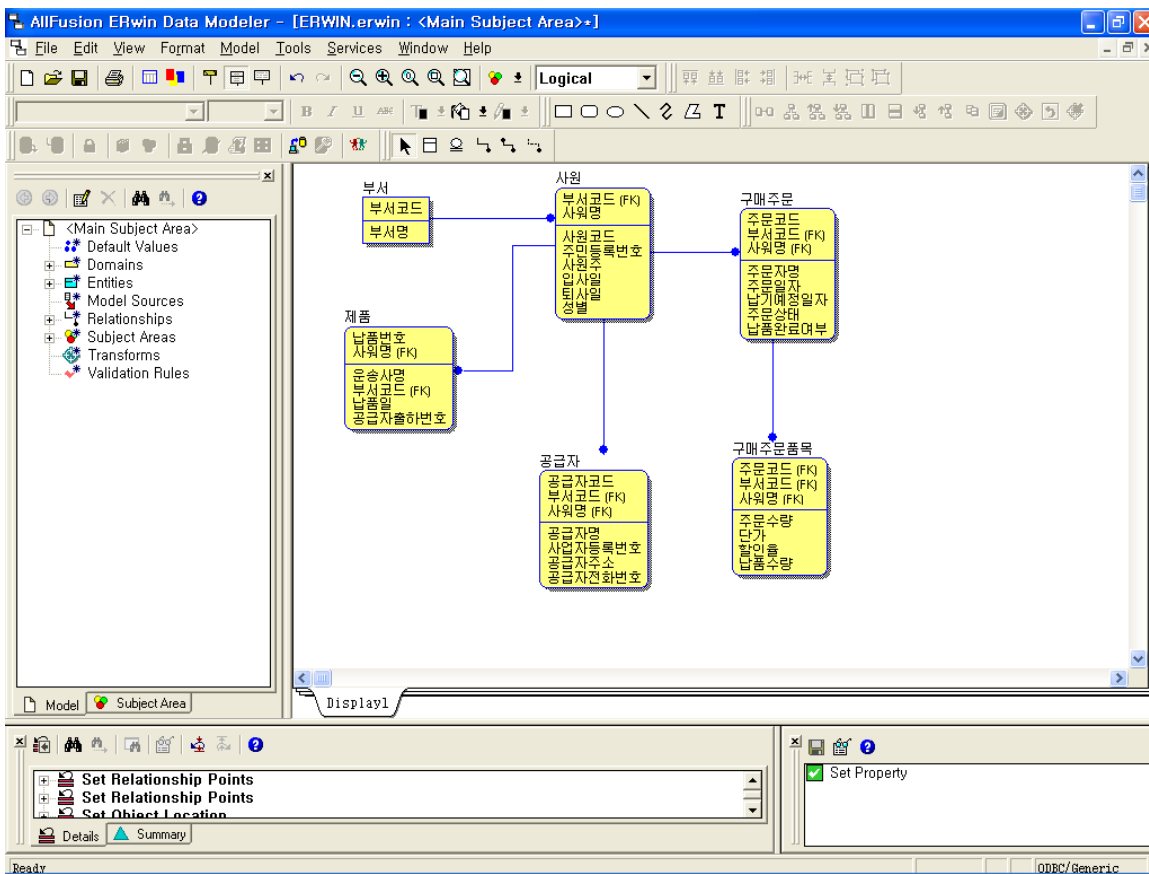
⑧ Finish 버튼을 클릭한다.

Finish 버튼은 위와 같은 절차에 따라 필요한 정보를 정상적으로 입력 해야만 버튼이 활성화 된다.



⑨ 로그 파일을 통해 변환 과정과 오류 발생 시 오류에 대한 내용을 확인 할 수 있다.

⑩ Close 버튼을 선택한다.



[SA 모델을 ERwin으로 Conversion 한 화면]