이클립스 기반 지능형 E-R 모델링 도구

eXERD

사 용 설 명 서

eXERD

Copyright 2010 (주)토마토시스템
이 사용 설명서의 내용과 eXERD 프로그램은 저작권법과 컴퓨터 프로그램 보호법으로 보호받고 있습니다.

본 소프트웨어 제품 및 소프트웨어의 복사본들에 대한 지적 재산권을 포함한 일체의 권리는 ㈜토마토시스템이 소유합니다.
이 권리는 대한민국의 저작권법과 국제 저작권 조양으로 보호받습니다.

설치 환경

항목	최소사양	권장사양
CPU	1.8GHz 이상	2.4GHz
RAM	512MB 이상	1GB 이상
HDD	500MB 이상	1GB 이상
OS	Microsoft Windows XP 이상, Apple Mac(이클립스 플러그인 버전)	

목차

eXERD 소개 **6**

1. eXERD 소개 7

2. eXERD의 주요기능 9

eXERD 설치 **10**

1. eXERD 설치하기 11

2. eXERD 제거하기 14

화면안내 **15**

1. 시작 화면 안내 16

2. 메인 화면 안내 17

3. 다이어그램 에디터 24

4. 테이블 속성 28

5. 컬럼 속성 29

6. 관계 속성 30

7. 데이터베이스 속성 31

8. 스키마 속성 32

9. 뷰 속성 33

10. 펑션 속성 34

11. 프로시저 속성 35

12. 트리거 속성 36

주요기능 **37**

1. 리버스 엔지니어링 38

2. 포워드 엔지니어링 43

3. 모델 비교 49

4. 용어 사전 59

5. 도메인 사전 70

6. 모델 검사 72

7. XScript 75

다이어그램 에디터 기능 92

1. 다이어그램 추가 93

2. 테이블 만들기 94

3. 컬럼 만들기 95

4. 관계 연결 96

5. 논리/물리 모드 변환 98

6. 편집 프리셋 99

7. 테이블 헤더 보이기 104

8. 용어사전 연결하기 105

9. 용어사전 일괄편집 106

10. 용어사전 역공학 108

11. 스타일 지정 109

12. 자동배치 111

13. 모델 검색 112

14. 테이블 검색 114

15. 테이블 개요 보기 115

부가기능 117

1. ERwin 파일 가져오기 118

2. 템플릿(보고서) 121

3. 모델 내보내기 126

4. XML 파일로 변환 133

환경설정 134

1. DBMS 연결 설정 135

2. 다이어그램 에디터 136

3. 명명 규칙 138

4. 모델 검사 139

5. 사용자에게 묻기 140

6. 용어 사전 141

기타 142

1. 업데이트 143

2. 제품등록 146

3. 도움말 147

4. eXERD에서 지원하는 파일 형식 148

5. eXERD 단축키 149

6. FAQ 153

7. XScript API 164

eXERD 소개

- eXERD 소개
- eXERD 의 주요 기능

eXERD 소개

1. eXERD 소개

무엇이 다른가요?

eXERD는 많은 점에서, 독특한 믿음을 바탕으로 새로운 가치를 추구 하고 있습니다.

- 사용자를 즐겁게 하여, 일의 능률을 높입니다.
- 어렵고 복잡한 수 많은 메시지 대신, 간단한 애니메이션으로 상황을 전달합니다.
- 프로그램의 기능 명세서보다, 실제 사용자의 작업패턴을 중시합니다.
- 오랜 시간이 걸리는, 관행적인 작업을 모두 자동화 하여, 사용자가 중요한 일에 집중 할 수 있도록 합니다.
- 사용자가 내린 명령의 의도를 정확하게 파악합니다.

■ 놀라운 인공지능

eXERD™에는 흥미롭고 놀라운 인공지능 기능들이, 사용자를 돕기위해 탑재되어 있습니다. 스타크래프트와 같은 게임에서나 사용되던, 최단 코스 찾기 알고리즘이, 선 배치 작업을 자동화 해줍니다. 뿐만 아니라, 위상학을 바탕으로, 테이블 배치를 최적으로 유지합니다.

또한 eXERD™는 사용자의 문서가 변경되면, 변경점을 계산하고, 가장 효과적인 시각적인 피드백을 공급할 뿐만 아니라, 사용자의 다음작업에 적합한 포커스나 선택을 제공합니다. 사용자가 최근 관심을 가졌던 대상들을 우선 나열하며 작업의 속도를 높입니다.

■ 사용자 중심의 인터페이스

사용자는 원할 때 언제라도, 논리나 물리로 전환 할 수 있으며, 테이블이나 컬럼의 배치 순서와 여부, 표시 방법 등 모든 것을 결정하고, 재사용할 수 있습니다.

각 협업자들은 관심사에 따라 자신만의 작업환경을 갖출 수 있습니다. eXERD 는 매우 직관적인 외형을 가진 위지윅 에디터 이면서, Eclipse 의 전통 및 UI 가이드라인을 준수하기 때문에, 도구에 익숙해지기 위하여 학습할 것이 거의 없습니다.

■ 아름답고 재미있는 시각효과

이는 단지 아름다울 뿐만이 아닙니다. 복잡한 메시지 없이도, 예쁘고 재미있는 애니메이션이 함축적인 정보를 즉각적으로 전달합니다. 예를 들어 Undo/Redo 시에 컬럼이 파괴되는 애니메이션을 통하여, 컬럼이 삭제되었다는 것을 바로 파악할 수 있습니다.

■ 복잡한 환경으로 부터 사용자를 보호합니다

ER 모델링은 매우 복잡한 작업 환경입니다.

예를 들어 어떤 PK 컬럼하나를 지움으로써, 관계한 모든 FK 들이 삭제될 수도 있고, 혹은 그저 관계만 무효 처리 되어야 할 수도 있습니다. 또, 그 FK 가 삭제됨으로써 연쇄적으로 또 다른 여파를 야기할 수도 있습니다.

전통적으로 ER 모델링은 DBA 한 명이 이 모든 상황을 수동으로 통제하고 관리하며 유지해야 했으며 높은 비용과 시간을 필요로 했습니다. eXERD 는 이러한 상황을 인식하고, 그 대처방안을 시각적으로 사용자에게 안내하여, 의도를 정확히 수행하며, 사용자가 예기치 못한 실수를 미연에 방지할 수 있게 합니다.

2. eXERD의 주요기능

사용자 편의성	**논리/물리 통합 모델링** 논리와 물리 모델링 동시 편집 가능, 다이어그램 컬럼에 대한 정보 조회 및 편집 가능
	관계선 최적경로 자동배치 테이블간 관계선 최적 경로 자동 배치, 관계선 배치 시간 절감 효과
	인터랙티브한 애니메이션 효과 수많은 정보를 애니메이션으로 전달하여 업무에 대한 집중도 및 작업 효율성 향상
	XScript XScript를 이용하면 대량의 모델을 빠르게 탐색 후 일괄 조작, 관련 산출물을 작성하는 프로그램을 직접 작성 강력한 편집 지원기능들과 함께 제공, 훌륭한 내장 스크립트들과 도움말을 포함
표준화 지원	**도메인 타입 지원** 컬럼의 용도와 역할에 따라 도메인을 정의하고 활용할 수 있음.
	표준용어 사전 지원 E-R다이어그램 작성시 사용할 표준용어와 물리명 정의, 표준용어사전
	작업 모델 검토 및 수정 논리적/물리적 오류 및 표준화 준수여부 검토, 문제 해결 자동 처리
호환성/통합성	**리버스 엔지니어링 지원** 기존DBMS 테이블 리버스 엔지니어링 지원, 역공학된 모델 재편집 가능, 관심영역별 다이어그램 생성 및 추가 가능
	포워드 엔지니어링 지원 설계한 E-R모델에 다양한 옵션 적용으로 DBMS테이블 생성 가능
	이클립스 기반 개발 툴 제공 완벽한 호환 및 연동 Eclipse와 완벽한 융합된 도구, 소스 버전관리 등 Eclipse 플러그인과 완벽하게 연동, 별도 프로그램 설치 없이 그대로 사용가능
	모델 비교 DB와 파일간의 차이나 시간대면 모델의 변경 내역 등을 시각적 파악 병합 작업 수행가능 변경 점에 대한 보고서를 출력하여 업무에 활용 가능
지원 DBMS	**DBMS** - Oracle 9, 10, 11 - DB2 9.X - MSSQ: 2005, 2008, 2012 - MySQL 5.1, 5.5

eXERD 설치

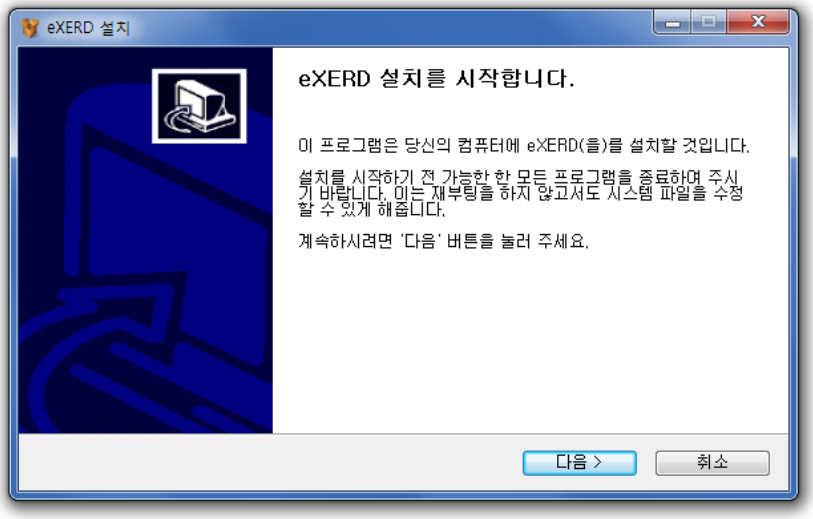
- eXERD 설치하기
- eXERD 제거하기

eXERD 설치

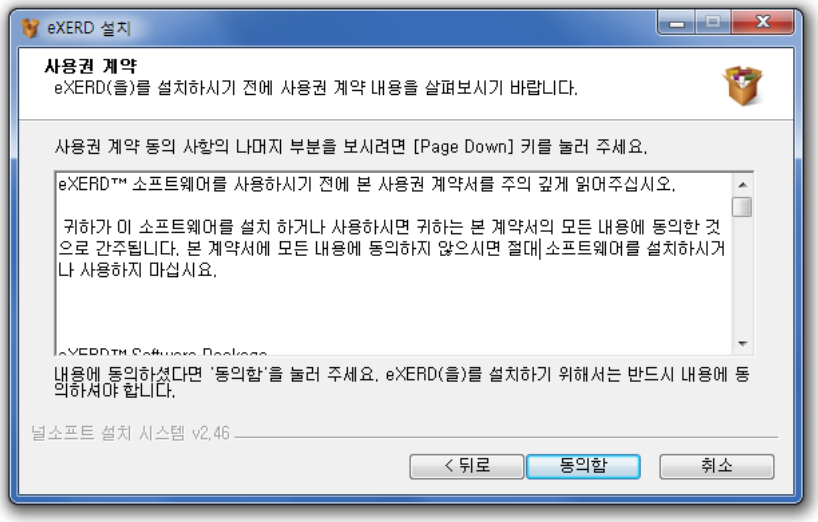
1) eXERD 설치하기

http://www.exerd.com 에서 다운 받아 설치 프로그램을 실행합니다.

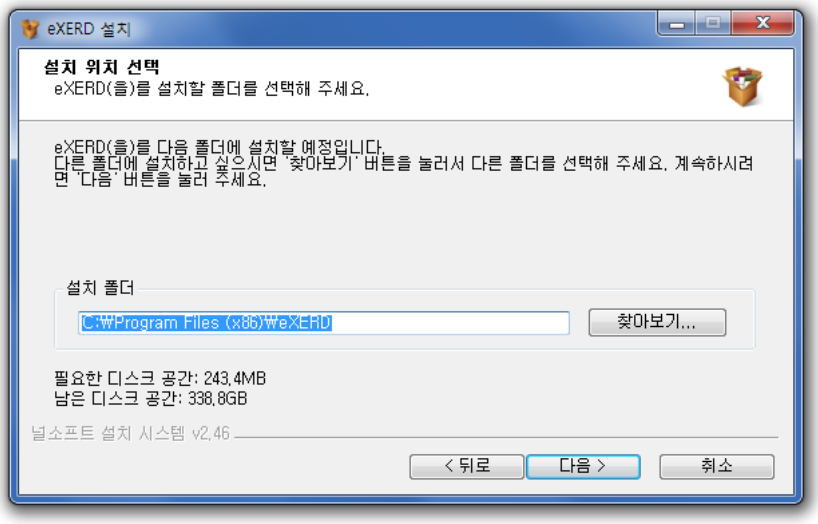
1) eXERD 설치 프로그램이 실행되면 시작 메시지가 나타납니다.



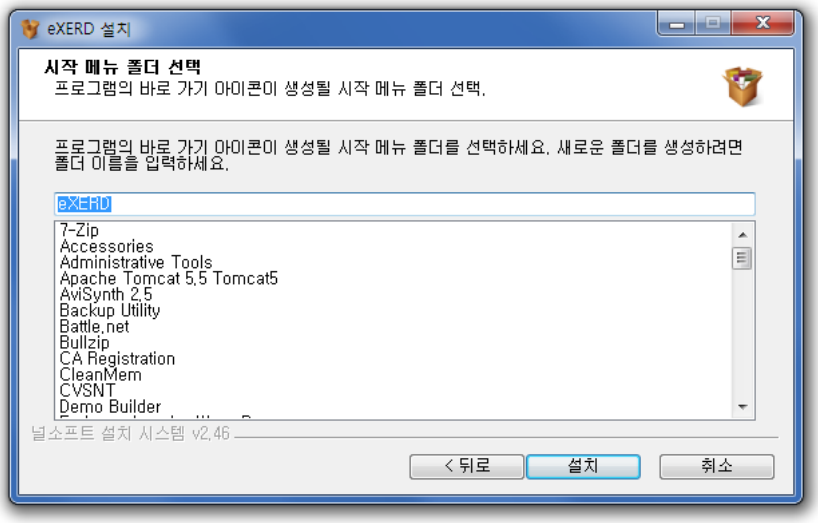
2) 소프트웨어 최종 사용자 사용권 계약서입니다.



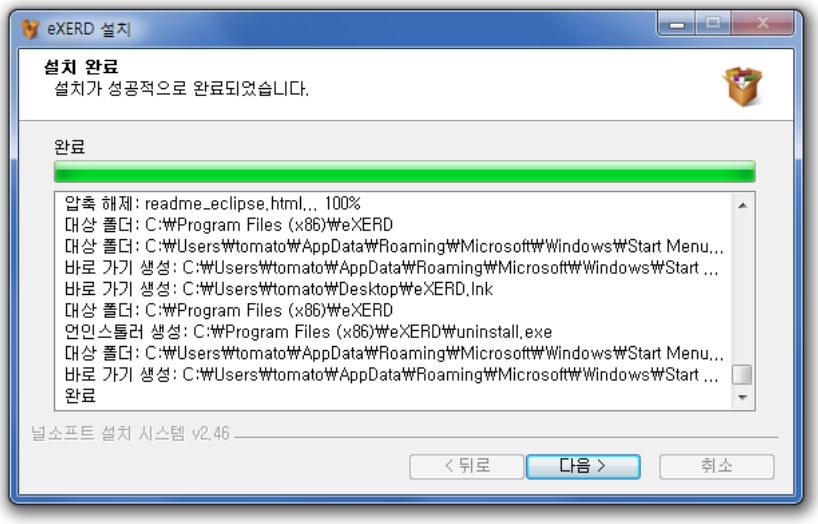
3) eXERD을 설치할 경로를 선택한 후 [다음] 버튼을 클릭합니다.



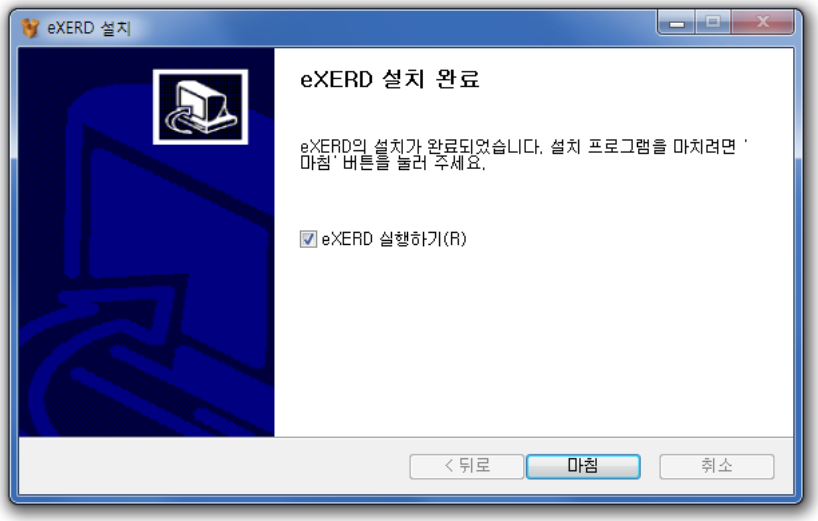
4) 시작메뉴 폴더를 선택합니다. 시작메뉴 폴더란 eXERD프로그램의 바로가기 아이콘이 만들어질 폴더를 의미합니다. 새로운 폴더를 만들 경우 직접 폴더 이름을 입력하십시오. 프로그램 그룹을 선택한 후 [다음] 버튼을 클릭합니다.



5) 설치가 완료되고 [다음] 버튼을 클릭합니다.



6) eXERD 설치 작업이 완료되었습니다. [마침] 버튼을 클릭합니다.

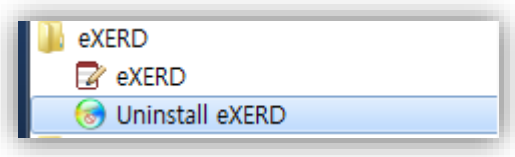


2) eXERD 제거하기

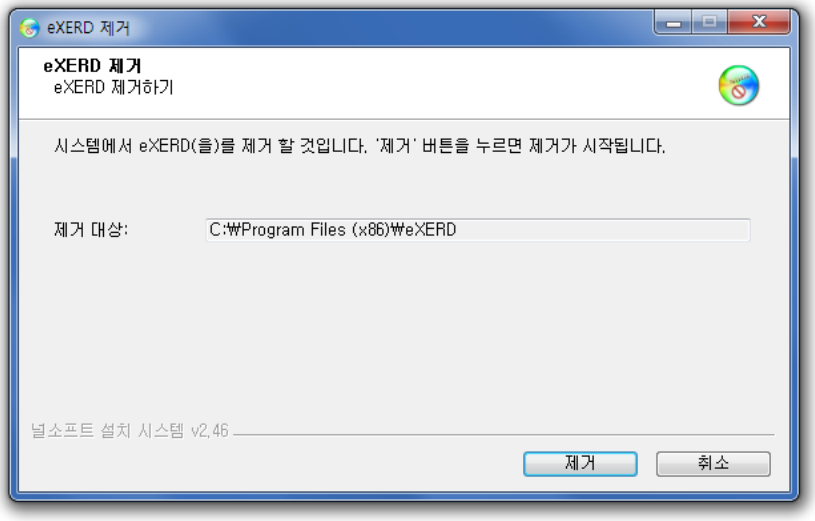
eXERD 프로그램 제거 방법입니다.

1) "시작-모든 프로그램" 메뉴를 펼칩니다.

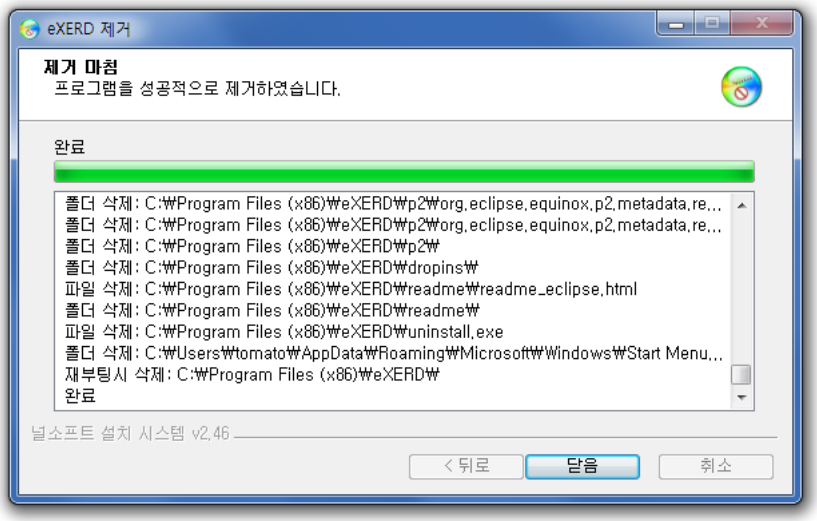
2) "eXERD-Uninstall eXERD"를 클릭합니다.



3) 제거 여부를 확인하는 창이 뜨면 [제거] 버튼을 클릭합니다.



4) 삭제가 완료되고 [닫기] 버튼을 클릭합니다.



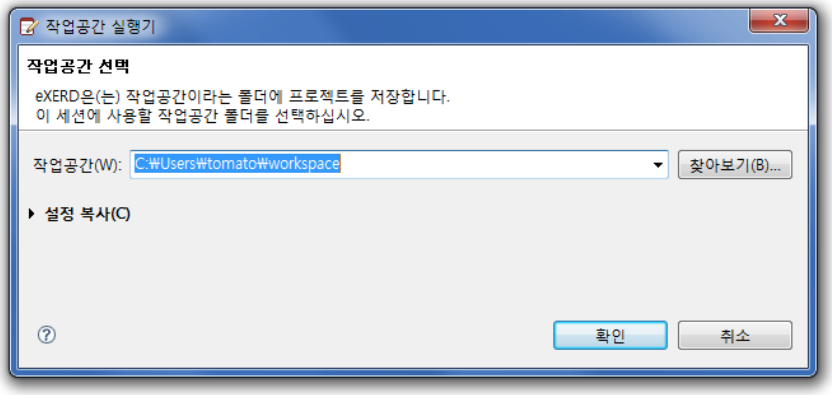
화면안내

- 시작 화면 안내
- 메인 화면 안내
- 다이어그램 에디터 화면 안내
- 테이블 속성 화면 안내
- 관계 속성 화면 안내
- 컬럼 속성 화면 안내
- 데이터베이스 속성 화면 안내
- 스키마 속성 화면 안내
- 뷰 속성 화면 안내
- 펑션 속성 화면 안내
- 프로시저 속성 화면 안내
- 트리거 속성 화면 안내

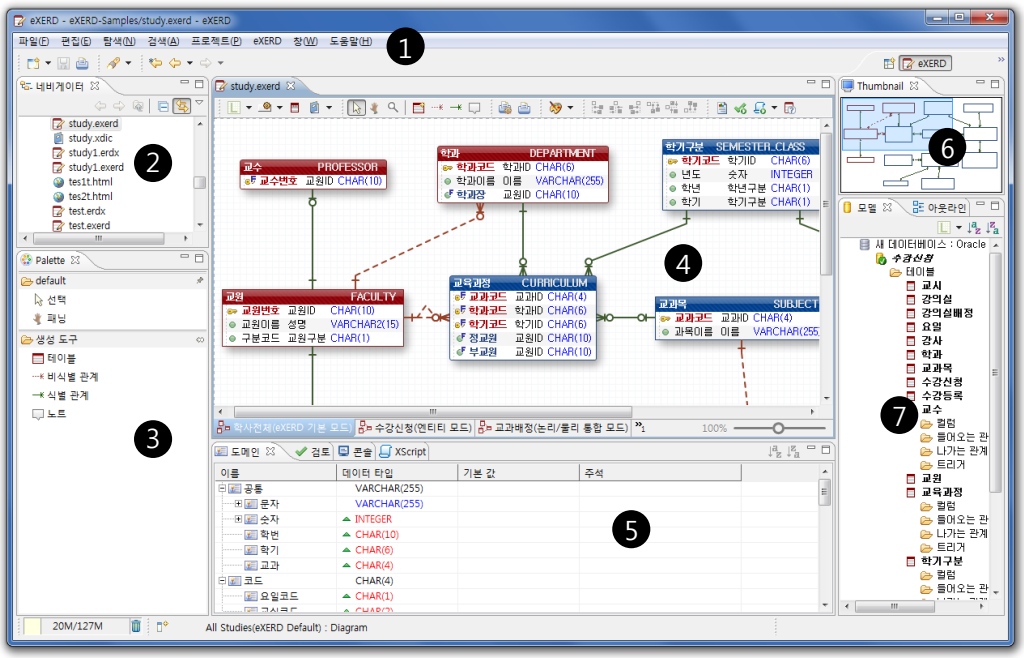
화면안내

1. 시작 화면 안내

eXERD를 실행하면 처음 작업 공간을 지정하는 화면이 나옵니다. 데이터를 저장할 공간을 지정하시고 [확인]을 클릭하십시오.



2. 메인 화면 안내



① 메뉴 영역

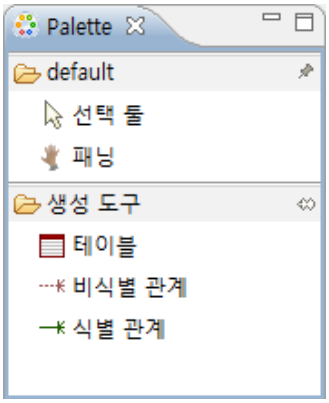
eXERD에서 지원하는 모든 기능을 실행 할 수 있는 메뉴 영역입니다.

② 네이게이션 뷰

파일을 폴더 구조로 볼수 있고, 관리 할 수 있는 영역입니다.

③ 팔레트 뷰

eXERD 다이어그램 에디터에서 사용 할 수 있는 툴들을 제공합니다.



다이어그램의 요소를 선택합니다.

다이어그램 표시 영역을 이동할 수 있습니다.

새 테이블을 생성합니다.

두 테이블 사이에 비식별 관계를 지정합니다.

두 테이블 사이에 식별 관계를 지정합니다.

팔레트 뷰의 툴들은 eXERD 에디터의 툴바에서도 동일하게 사용할 수 있습니다.

④ 다이어그램 에디터

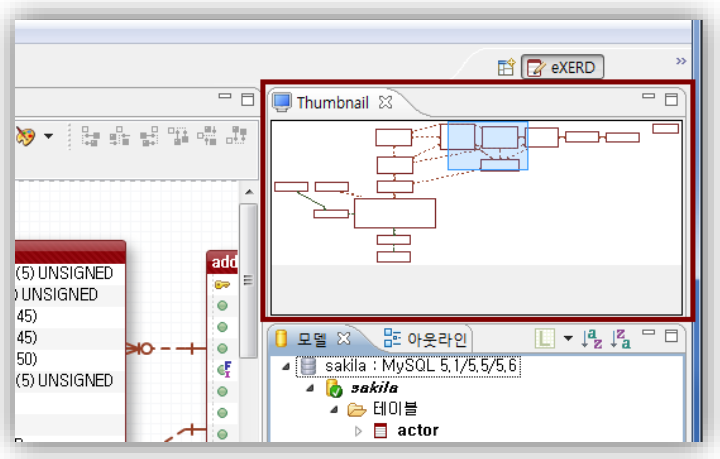
다이어그램을 작성 하거나 편집하는 화면입니다.

⑤ 도메인 뷰

도메인 타입을 구성하고 활용할 수 있습니다. 도메인을 사용하면 의미상 동일한 컬럼들에 대해 일관된 규칙을 적용할 수 있습니다.

⑥ 썸네일 뷰

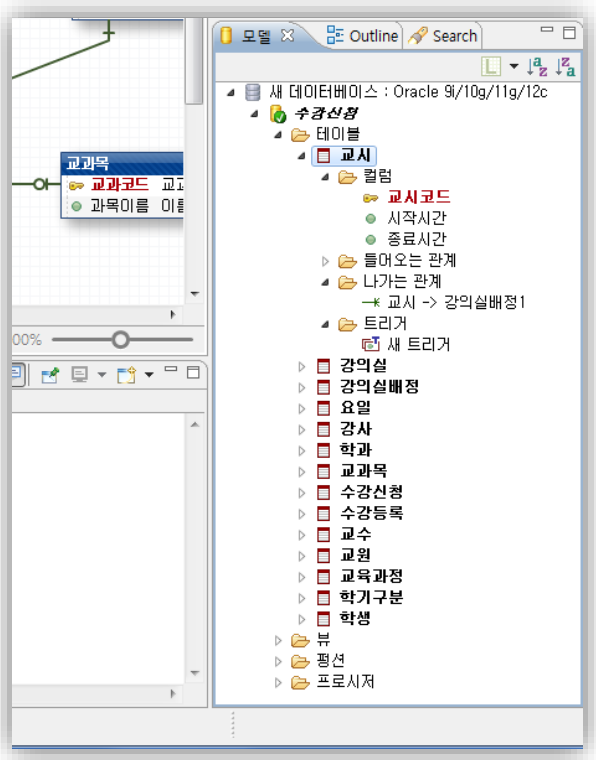
다이어그램 전체화면에 대한 축소화면을 보여줍니다. 썸네일 뷰에서 에디터에 표시할 영역을 빠르게 선택할 수 있습니다.



썸네일 뷰의 파란색 사각 영역을 드레그 하면 에디터의 표시 영역이 스크롤됩니다.

⑦ 모델 뷰

문서상의 전체 모델이 트리 구조로 표시 되며 스키마(소유자)에 대한 설정들을 할 수 있습니다.

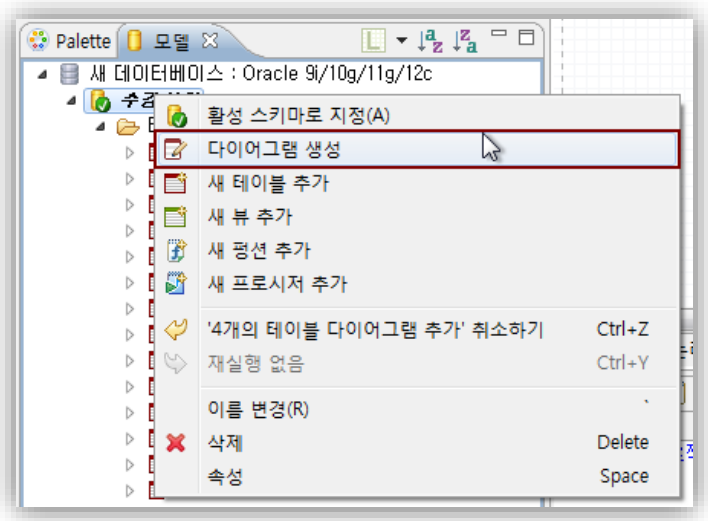


- 모델뷰 툴바의 정렬 버튼으로 오름차순/내림차순으로 정렬할 수 있습니다.
- 현재 활성화된 다이어그램에 배치된 모델(테이블)은 **굵은 글씨체(BOLD)**로 표시되며 **더블클릭** 하면 다이어그램에서 선택됩니다.
- 테이블의 관계가 **들어오는 관계**와 **나가는 관계**로 구분하여 표시됩니다.

■ 다이어그램 생성 마법사

다이어그램 생성 마법사를 이용하여 특정 테이블들을 배치하는 다이어그램을 손쉽게 작성할 수 있습니다.

1) 모델 뷰에서 팝업메뉴의 **다이어그램 생성** 항목을 실행합니다.



Tip! *마법사 실행 시 모델뷰에서 선택된 대상을 기준으로 시작됩니다.*

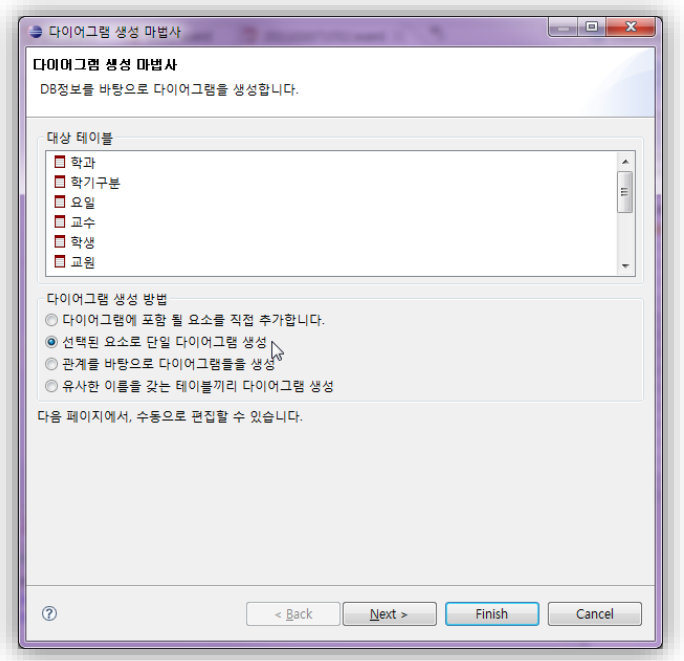
2) ***Document 항목 선택 시*** *: 전체 스키마의 모든 테이블들이 대상이 됩니다.*

3) ***스키마 선택 시*** *: 해당 스키마의 전체 테이블들이 대상이 됩니다.*

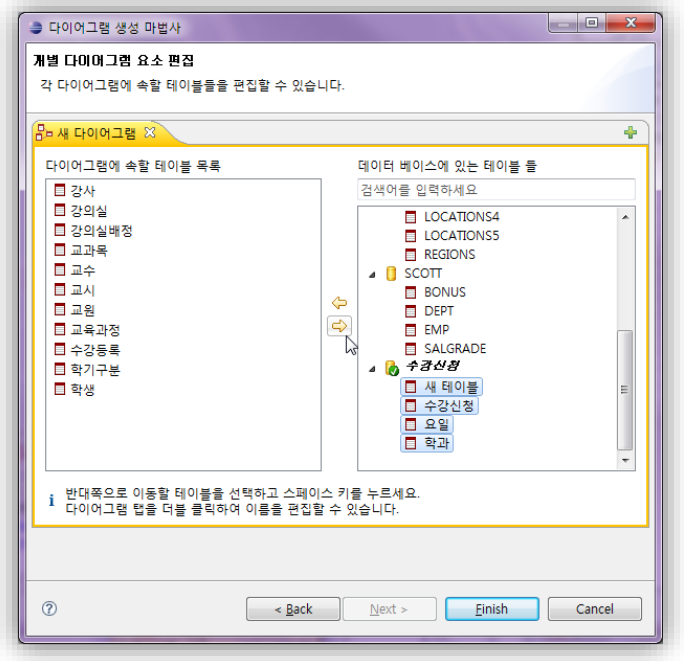
4) ***특정 테이블들을 선택 시*** *: 선택한 테이블들이 기본 대상이 됩니다.*

※ 초기 선택과 별도로 위저드 단계에서 직접 대상을 선택할 수도 있습니다.

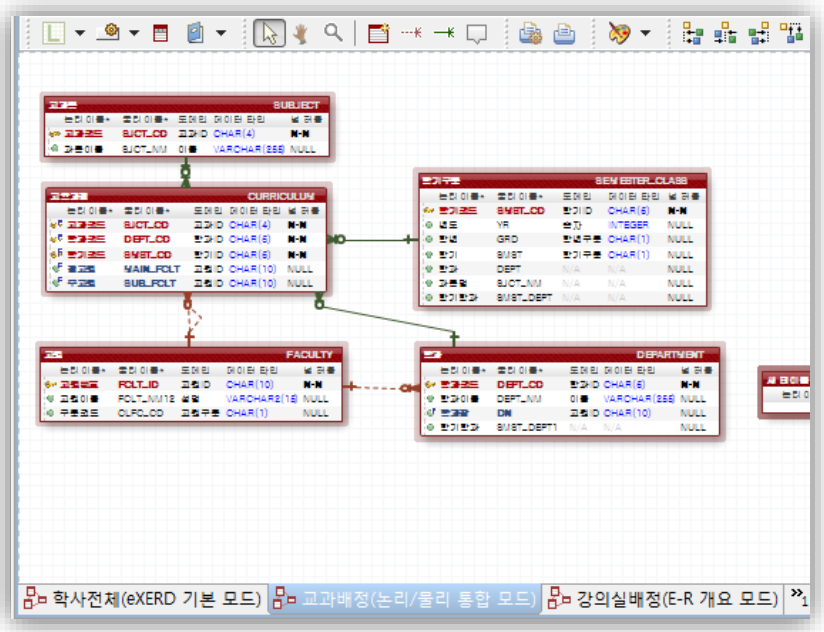
2) 자동 및 수동으로 다이어그램을 구성할 방식을 선택합니다.



3) 다이어그램의 이름 및 추가하거나 배제할 테이블을 선택합니다.



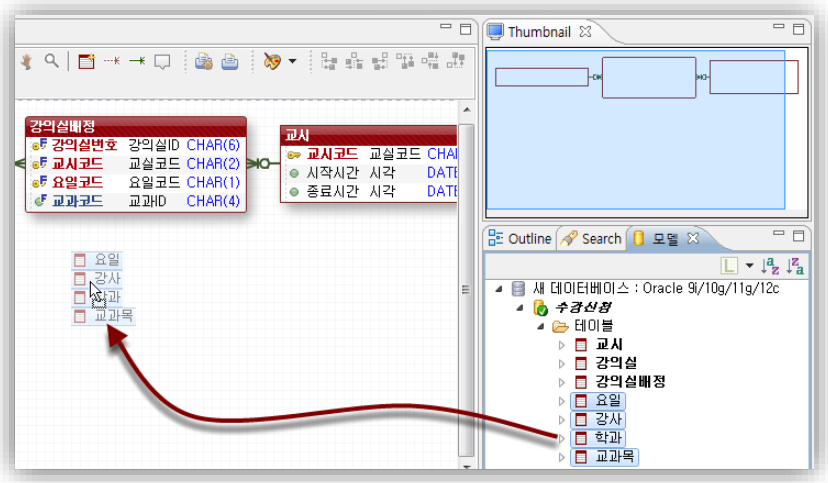
4) **[Finish]**를 누르면 선택한 테이블들로 구성된 새 다이어그램 탭이 추가됩니다.



■ 다이어그램에 모델을 직접 배치하기

모델 뷰에서 테이블을 선택해서 다이어그램으로 드레그할 수 있습니다.

한 다이어그램에 같은 모델(테이블)을 중복하여 배치할 수는 없지만 여러개의 다이어그램에 관심영역별로 테이블을 배치할 수 있습니다.

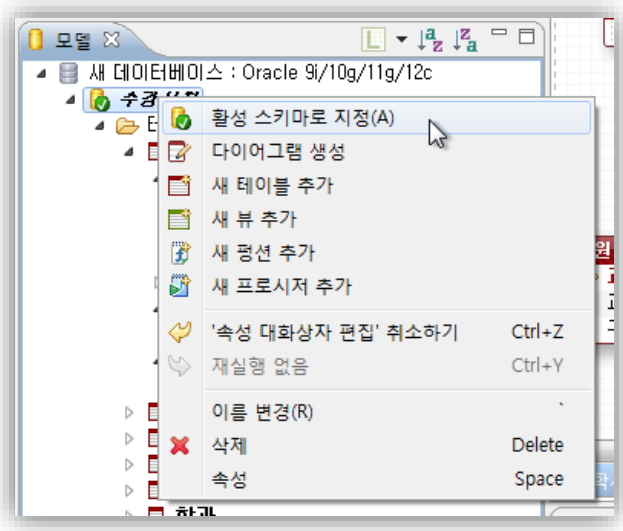


Tip! *자동 배치 기능을 이용하면 일렬 배치되어 추가된 테이블들을 보기좋게 나열할 수 있습니다.*

■ 활성 스키마 지정하기

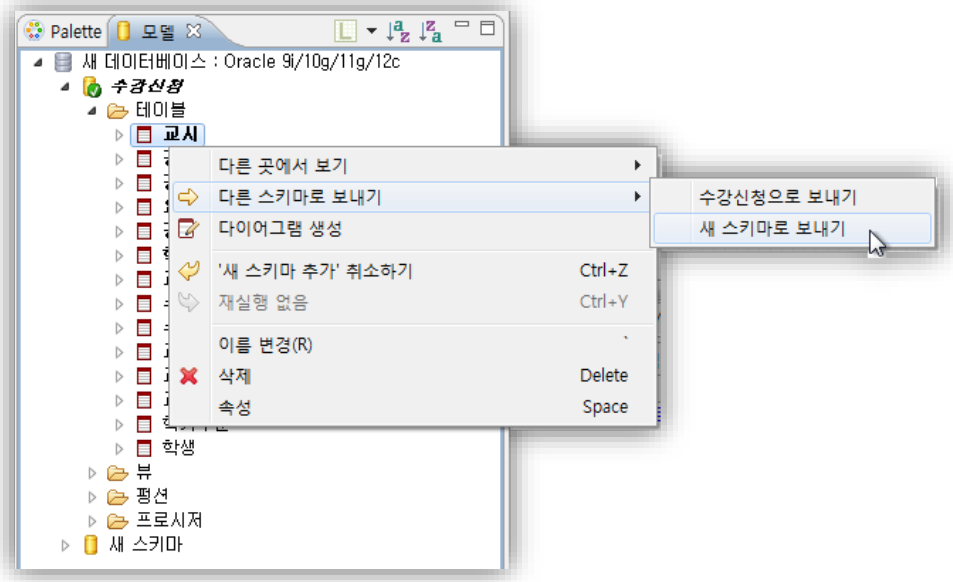
스키마를 선택하고 팝업메뉴의 **활성 스키마로 지정**을 실행합니다.

다이어그램에서 새로 생성한 테이블은 활성 스키마에 추가됩니다.



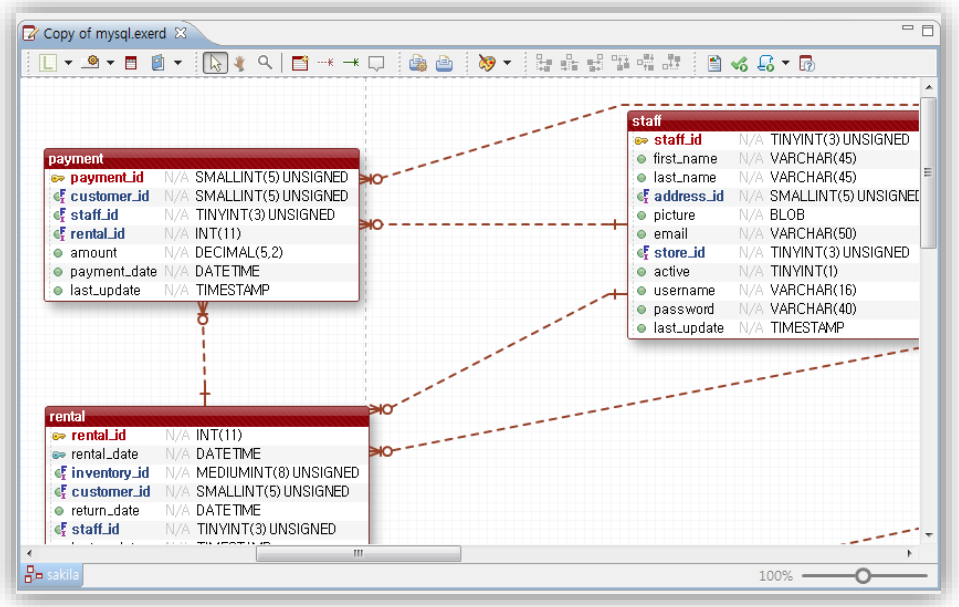
■ 스키마 변경하기

테이블 선택하고 팝업메뉴의 **다른 스키마로 보내기**를 실행합니다.



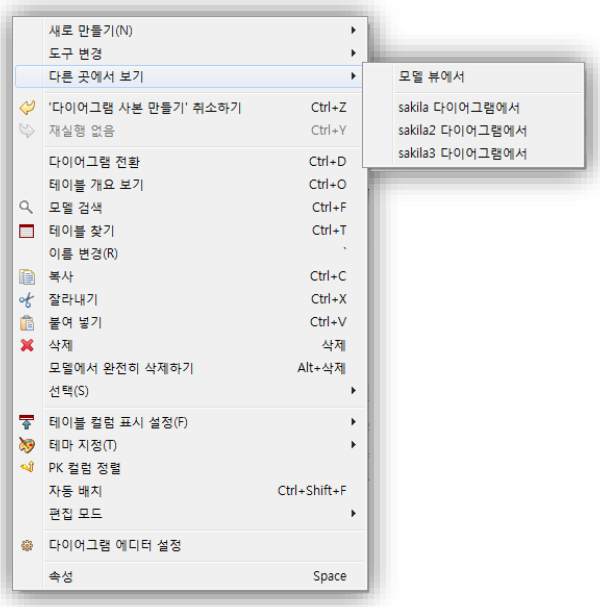
3. 다이어그램 에디터

.exerd 파일을 편집하는 기본 에디터입니다. E-R 다이어그램을 작성 하는데 최적화 되어 있으며 직관적이고 편리한 인터페이스를 제공합니다.



에디터 상단의 툴바는 다이어그램을 작성하는데 필요한 대부분의 기능을 제공합니다. 각 버튼에 마우스를 위치하면 간단한 설명과 단축키를 표시합니다.

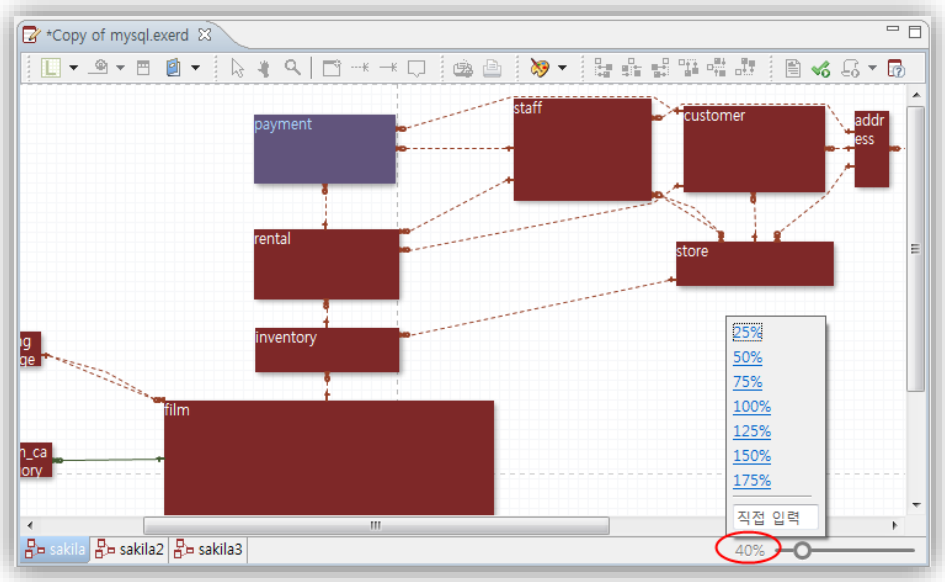
테이블이나 컬럼을 선택하고 팝업메뉴(컨텍스트메뉴)를 열면 아래 이미지와 같이 다양한 편집 명령들을 실행할 수 있습니다.



■ 다이어그램 확대 / 축소

에디터 우측 하단의 Zoom 컨트롤을 이용하여 다이어그램을 빠르게 확대/축소 할 수 있습니다.

***Tip! Ctrl** 버튼을 누른채 마우스의 휠을 움직여도 확대/축소를 할 수 있습니다.*

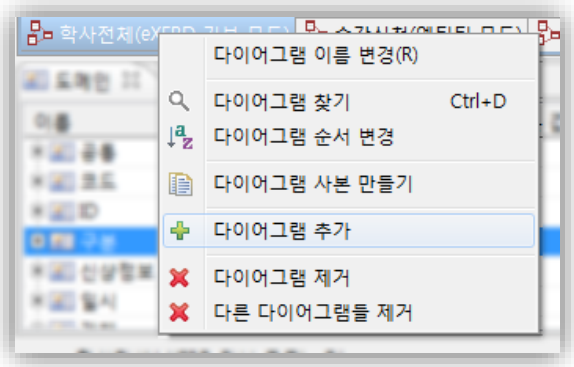


1) Zoom 컨트롤의 비율 표시부분을 클릭하면 100%로 적용됩니다.

- 마우스 오른쪽 버튼을 이용하면 특정 비율로 빠르게 변경할 수 있습니다.
- Zoom 컨트롤의 슬라이더 좌측/우측 영역을 클릭하면 비율이 10%씩 감소/증가 합니다.
- 50% 미만으로 축소하면 테이블명만 표시하여 식별이 용이합니다.

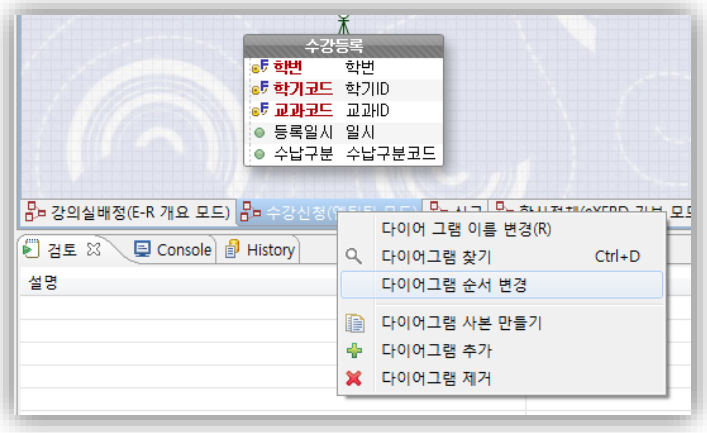
■ 다이어그램 추가하기

에디터 하단의 다이어그램 탭 위에서 팝업메뉴를 이용하여 추가 및 편집이 가능합니다.

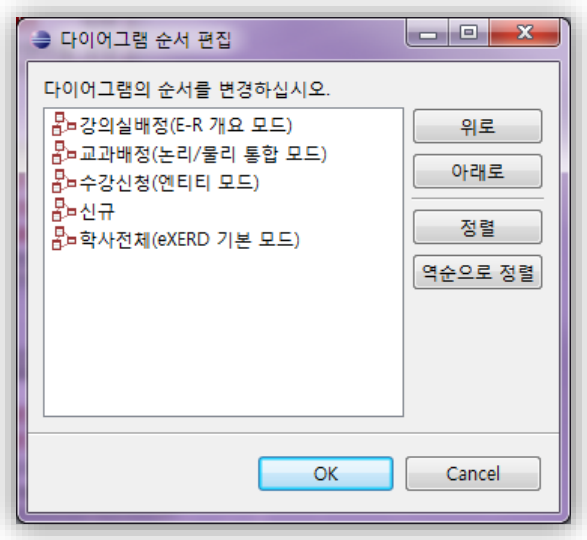


■ 다이어그램 탭 정렬하기

에디터 하단의 다이어그램 탭 위에서 팝업메뉴의 **다이어그램 순서 변경**을 이용하여 탭 순서를 변경할 수 있습니다.

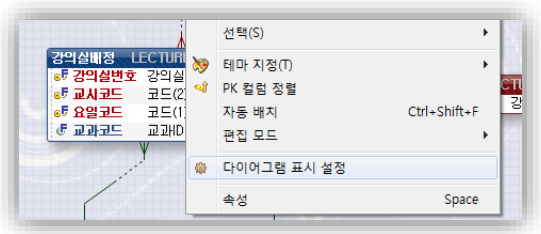


순서를 직접 조정하거나 오름차순/내림차순으로 정렬할 수 있습니다.

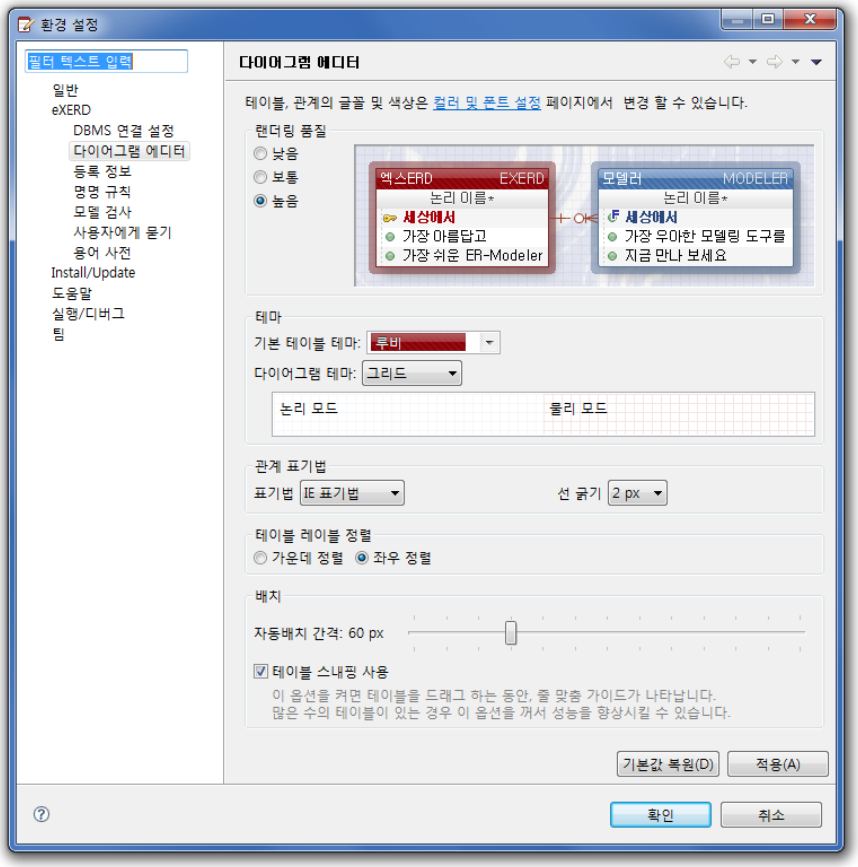


■ 다이어그램 표시 설정

팝업메뉴의 **다이어그램 표시 설정**에서 다이어그램의 랜더링 품질 및 관계선에 대한 표기법을 설정할 수 있습니다.



또는 메뉴 **Window > Preferences > eXERD > 다이어그램**으로 실행할 수 있습니다.

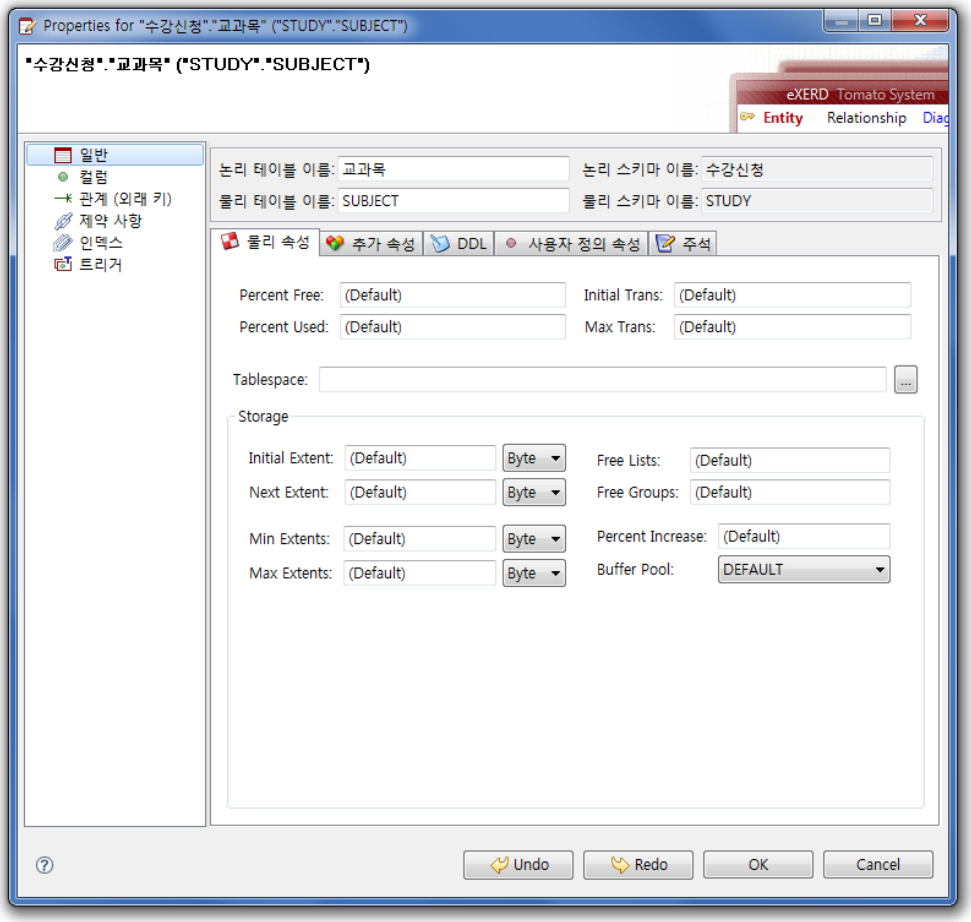


- 랜더링 품질을 설정할 수 있습니다. 저사양의 PC에서는 **낮음**으로 설정하시기를 권장합니다.
- 관계 표기법을 IE 또는 Barker 표기법으로 설정할 수 있습니다.
- 자동배치 시 적용될 테이블간의 간격을 설정할 수 있습니다.
- 테이블 스내핑 기능을 켜고 끌 수 있습니다.
- 새로 만들어질 테이블에 적용될 기본 테이블 테마를 지정할 수 있습니다.

4. 테이블 속성

테이블을 선택 후 **Space** 키 또는 **팝업메뉴 > 속성**을 실행합니다.

- 해당 테이블의 일반적인 속성들과 DBMS 의 물리적 속성들을 편집할 수 있습니다.
- 컬럼 및 제약사항, 관계, 인덱스 등 해당 테이블의 모든 내용을 편집할 수 있습니다.

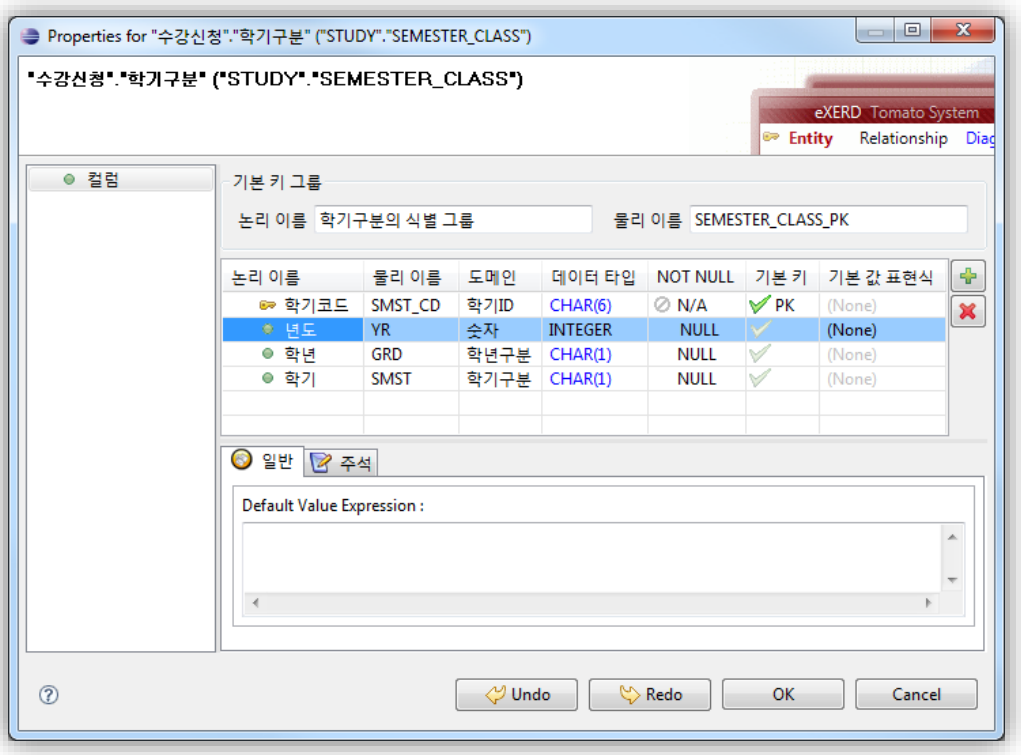


상기 스크린샷은 타겟 DBMS 가 'Oracle'로 설정된 경우 입니다.

5. 컬럼 속성

컬럼을 선택 후 **Space** 키 또는 **팝업메뉴 > 속성**을 실행합니다.

- 해당 테이블의 컬럼들이 표시되며 대상 컬럼이 선택됩니다.

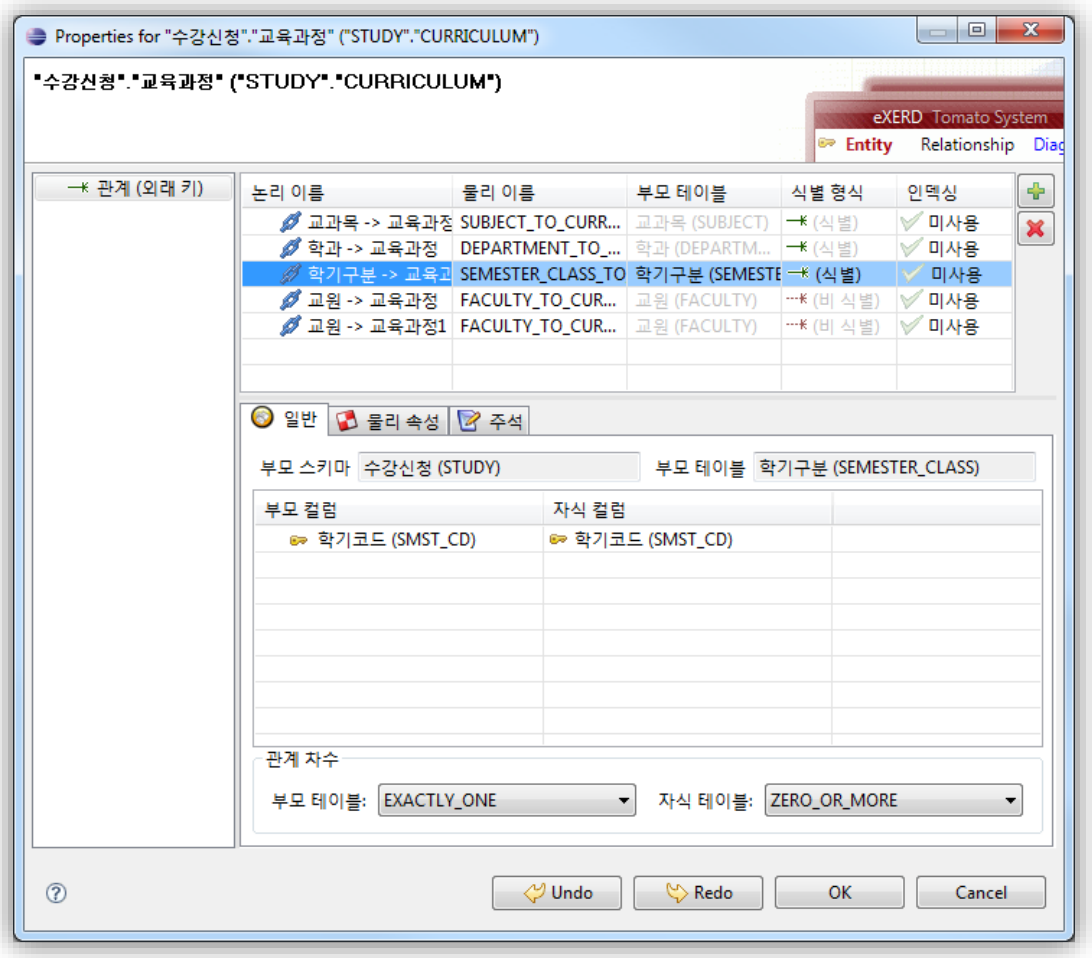


상기 스크린샷은 타겟 DBMS 가 'Oracle'로 설정된 경우 입니다.

6. 관계 속성

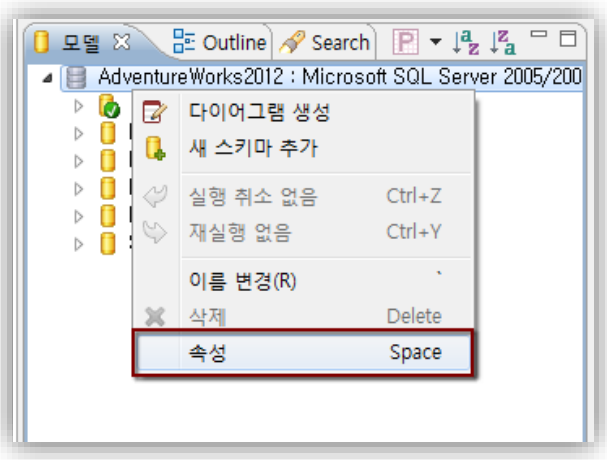
관계선을 선택 후 **Space** 키 또는 **팝업메뉴 > 속성**을 실행합니다.

- 참조 테이블의 관계들이 표시되며 대상 관계가 선택됩니다.
- 관계에 포함된 자식(FK) 컬럼들을 변경할 수 있습니다.



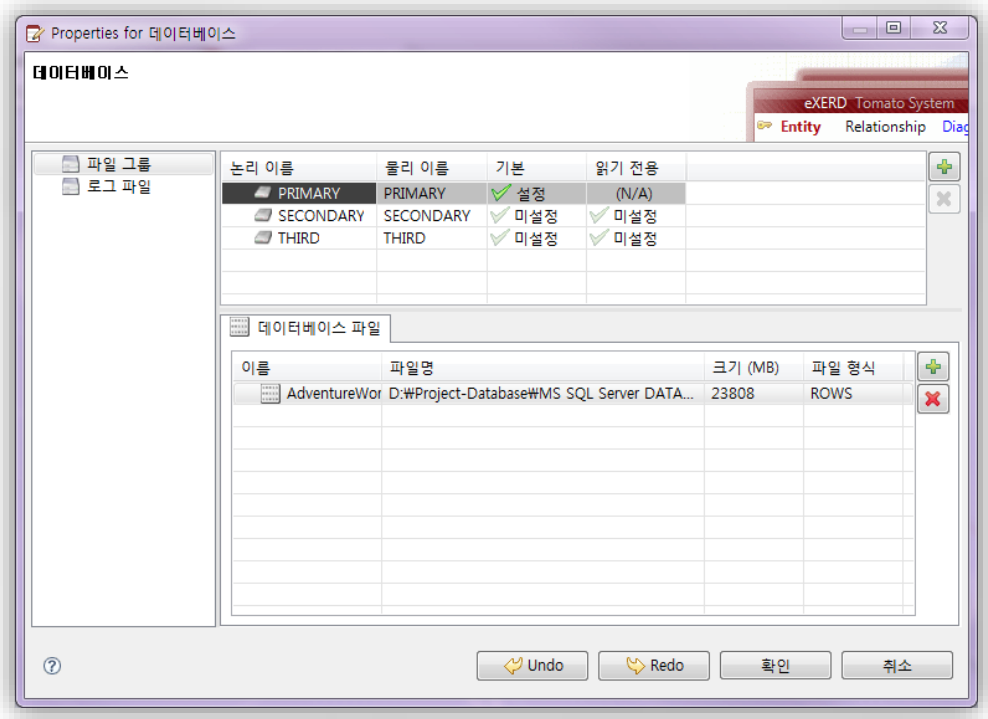
7. 데이터베이스 속성

모델 뷰에서 데이터베이스 항목을 선택하여 속성 대화상자를 열 수 있습니다.



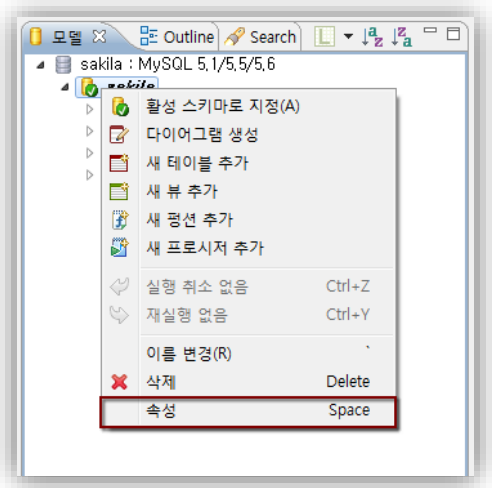
■ 파일 그룹

데이터베이스 속성 대화상자에서 파일그룹을 추가 / 제거, 속성 변경 등의 작업을 할 수 있습니다.



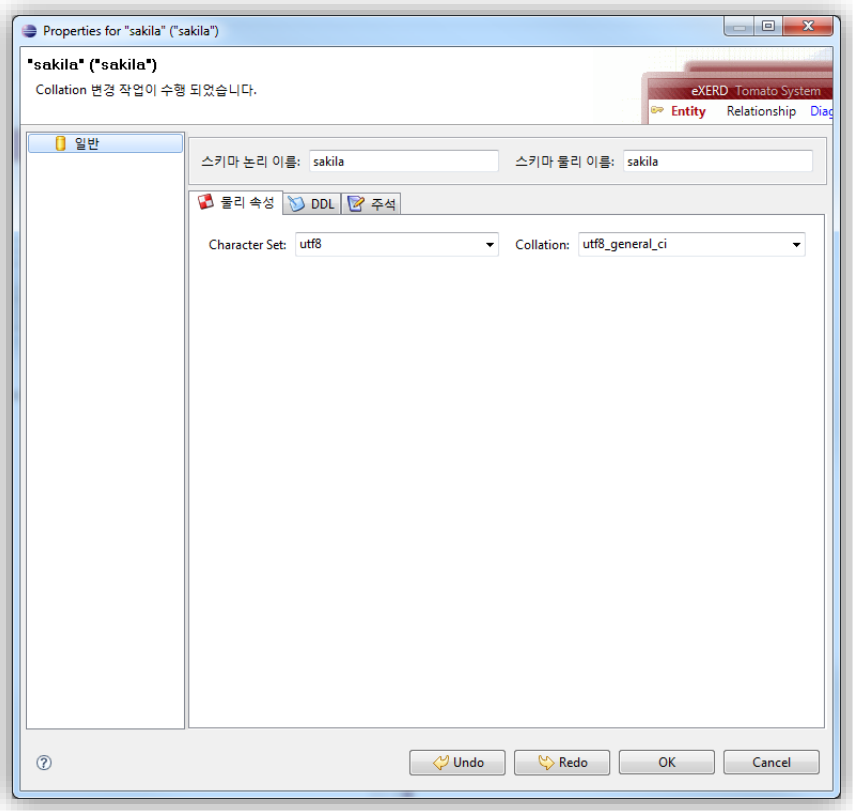
8. 스키마 속성

모델 뷰에서 스키마 항목을 선택하여 속성 대화상자를 열 수 있습니다.



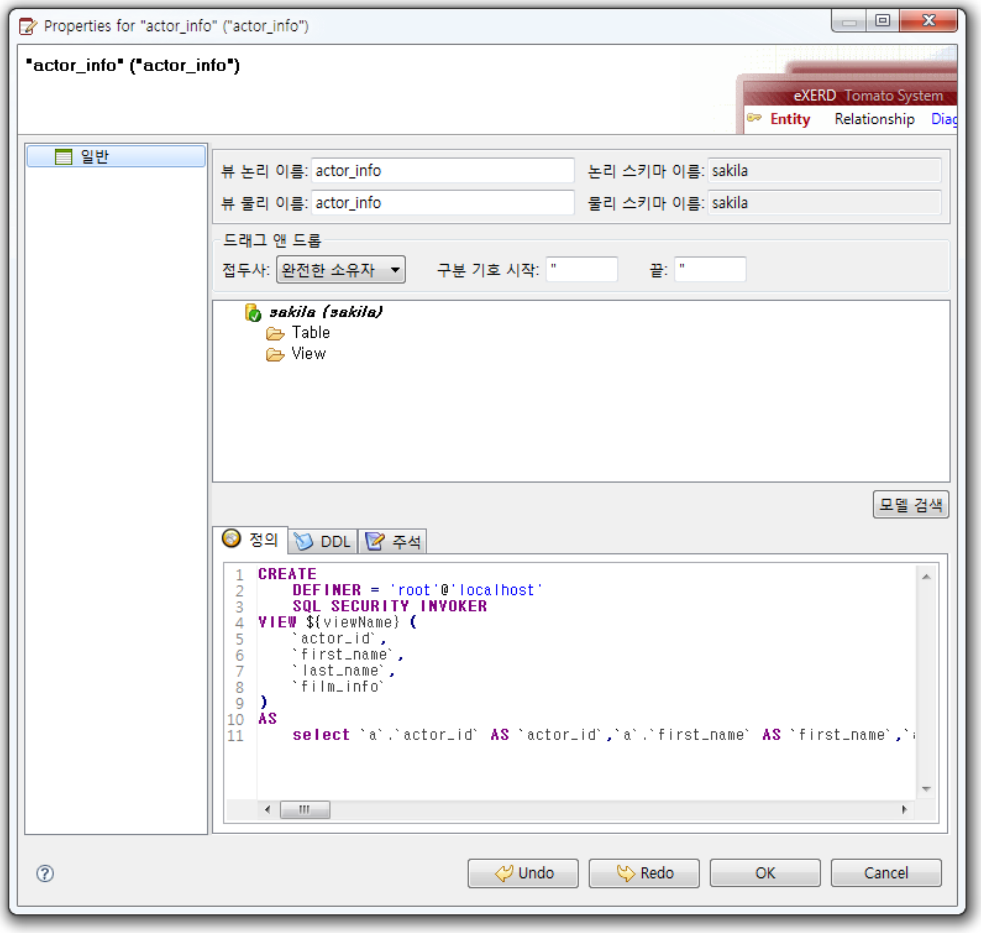
■ 스키마 속성 대화상자

스키마 속성 대화상자에서 스키마의 속성을 편집 할 수 있습니다.



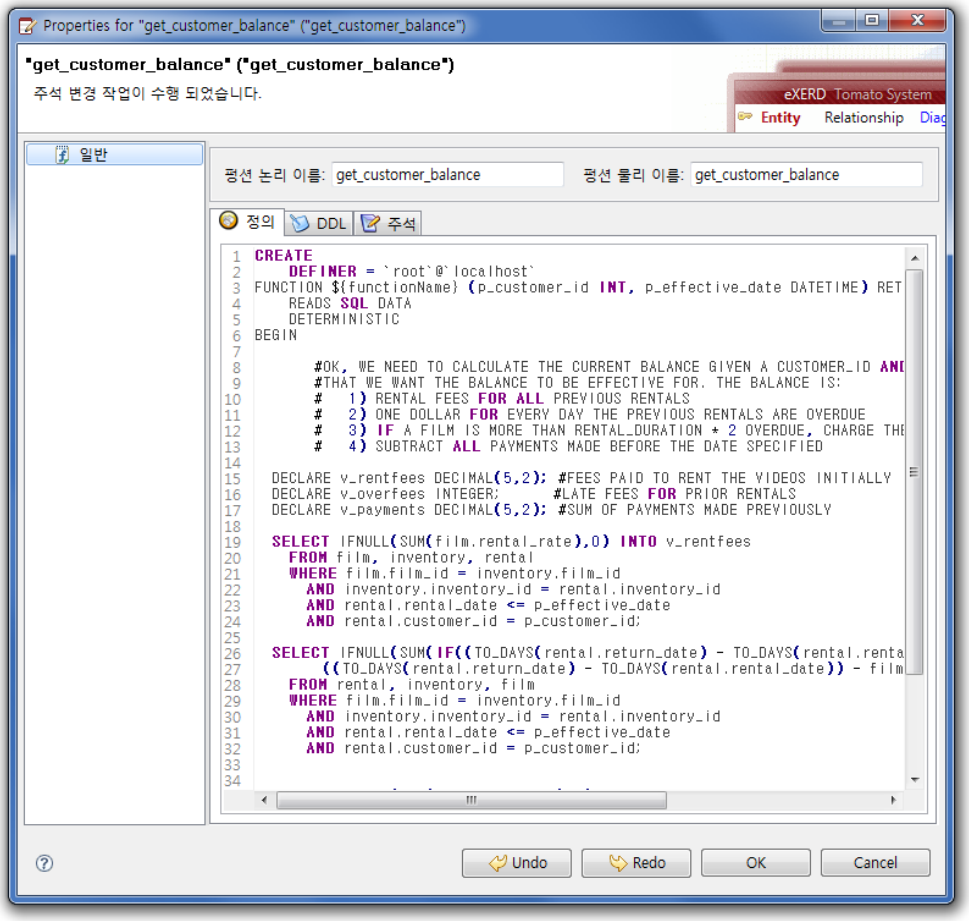
9. 뷰 속성

뷰 속성 대화상자에서 뷰의 속성을 편집 할 수 있습니다.



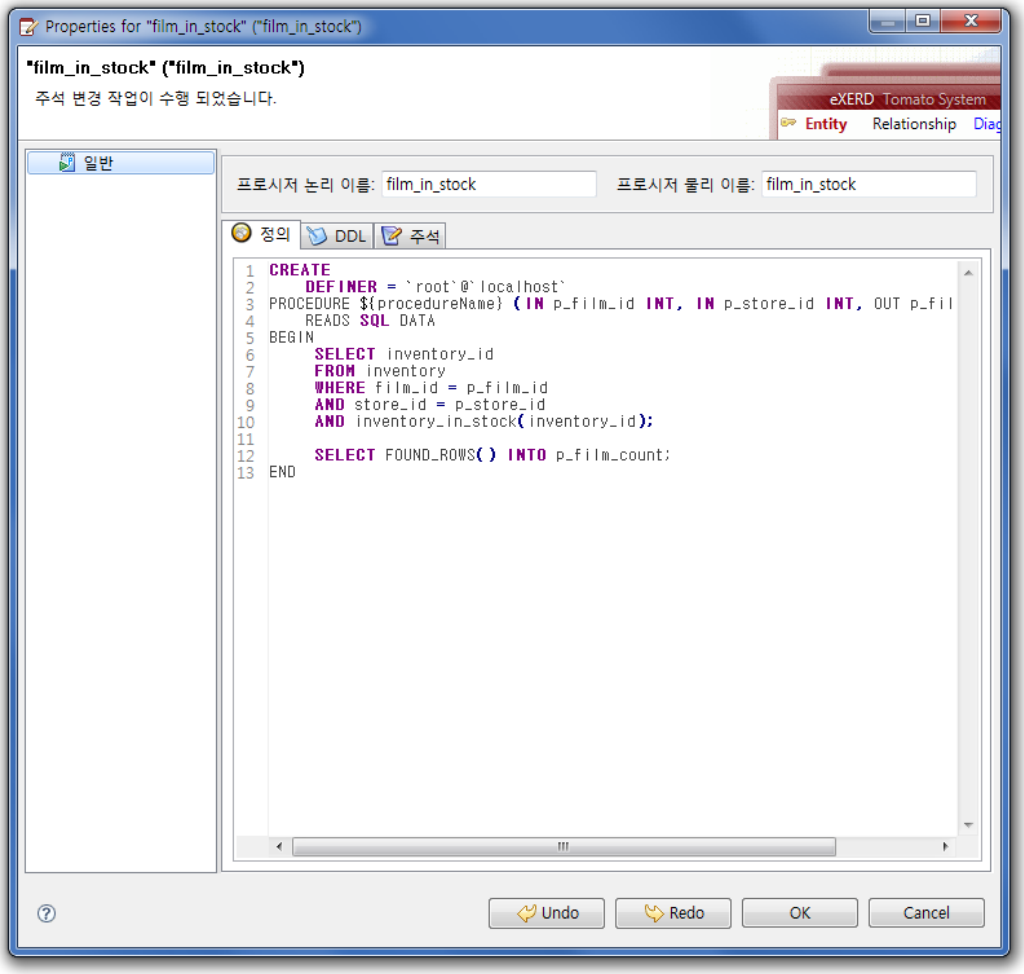
10. 펑션 속성

펑션 속성 대화상자에서 펑션의 속성을 편집 할 수 있습니다.



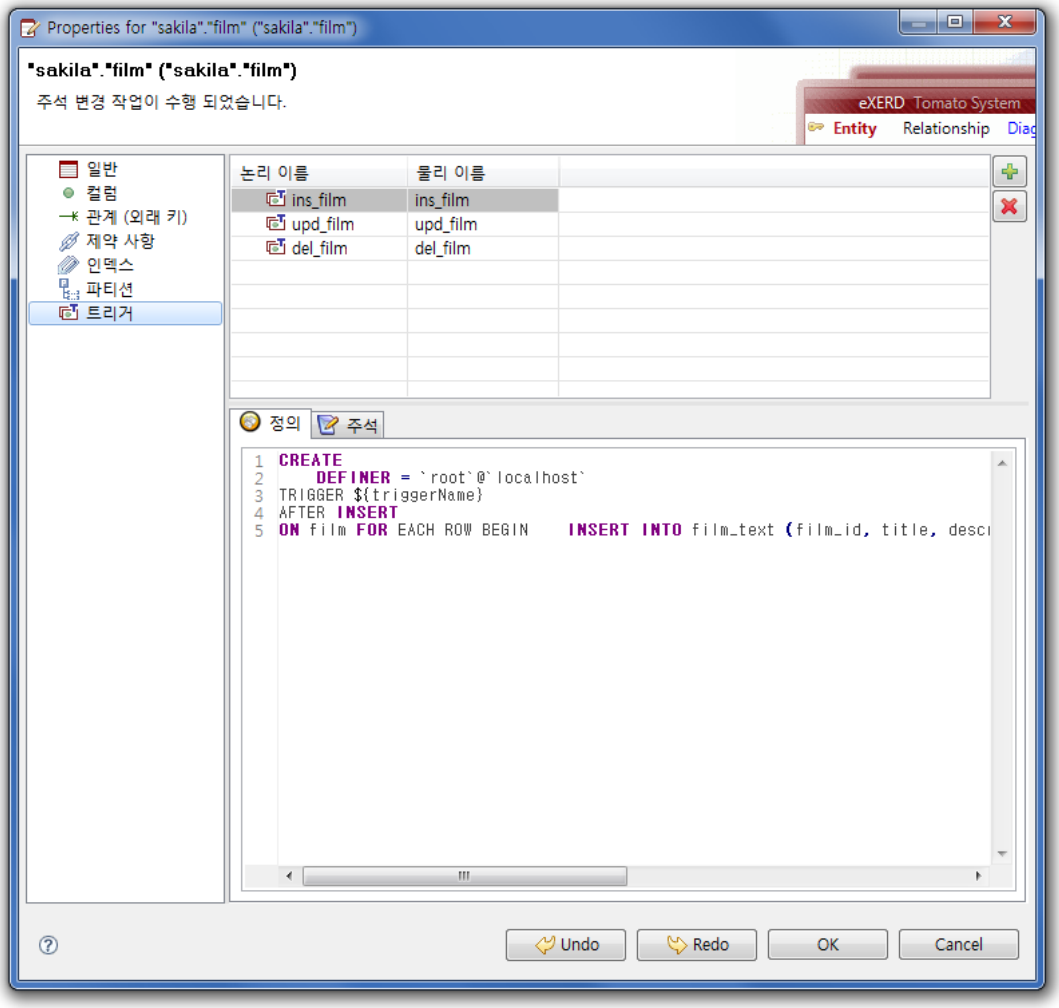
11. 프로시저 속성

프로시저 속성 대화상자에서 프로시저의 속성을 편집 할 수 있습니다.



12. 트리거 속성

트리거 속성 대화상자에서 트리거의 속성을 편집 할 수 있습니다.



주요기능

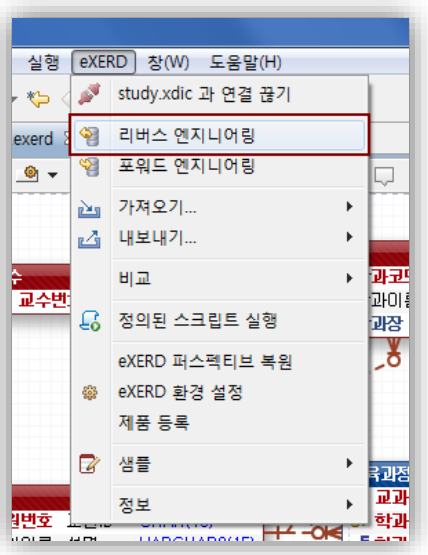
- 리버스 엔지니어링
- 포워드 엔지니어링
- 파일 비교
- 데이터베이스와 비교
- 용어 사전
- 도메인 사전
- 모델 검사
- XScript

주요기능

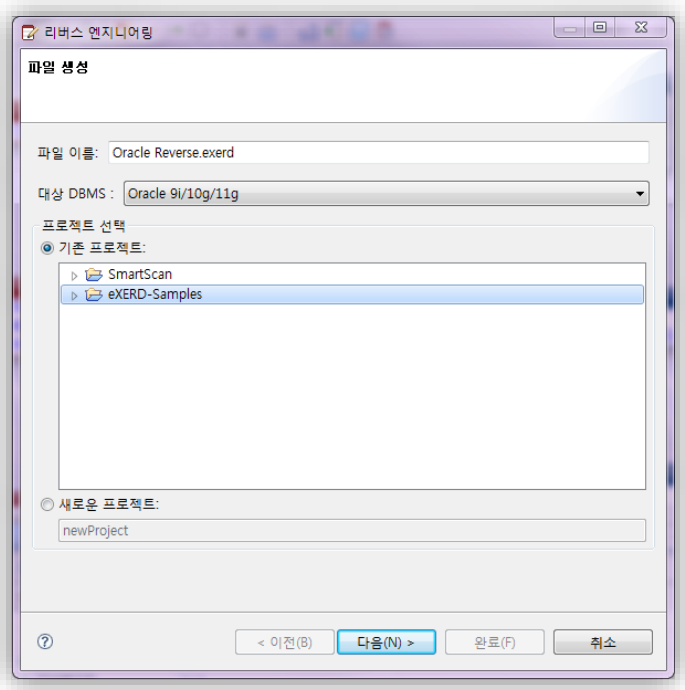
1. 리버스 엔지니어링

DBMS 의 객체들을 E-R 모델로 역공학하여 가져오는 기능입니다.

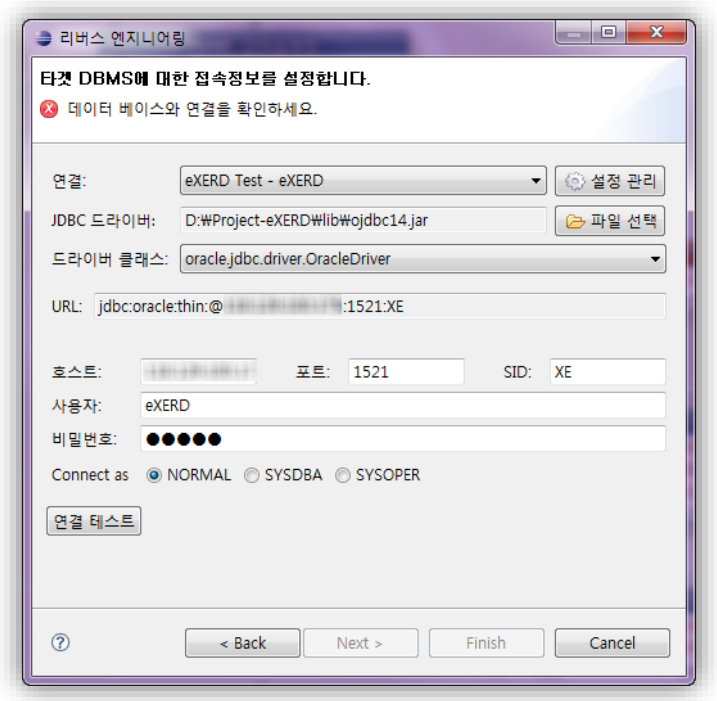
1) **eXERD > 리버스 엔지니어링** 메뉴를 선택합니다.



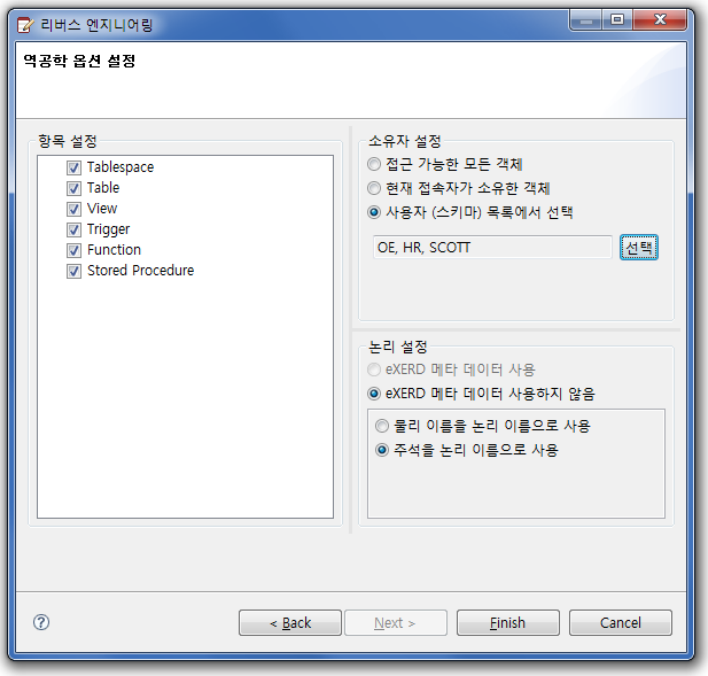
2) 리버스 엔지니어링 위저드가 실행됩니다. 파일 이름과 대상 DBMS 를 설정합니다.



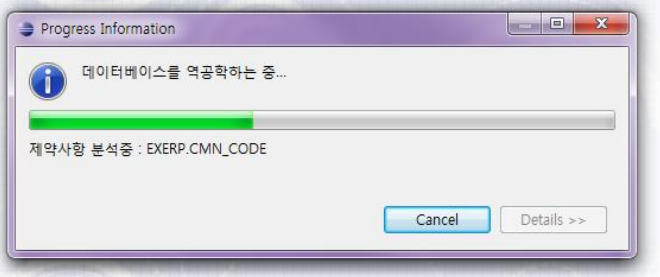
3) 데이터베이스 연결 정보를 입력하고 **[연결 테스트]** 버튼으로 접속 여부를 확인합니다.

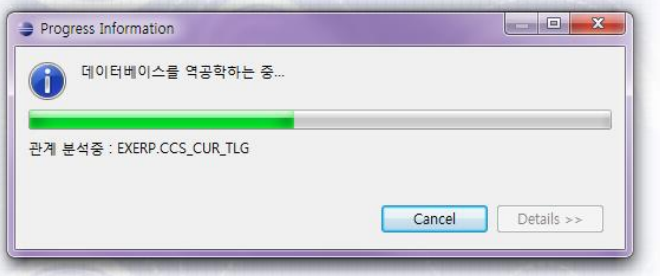


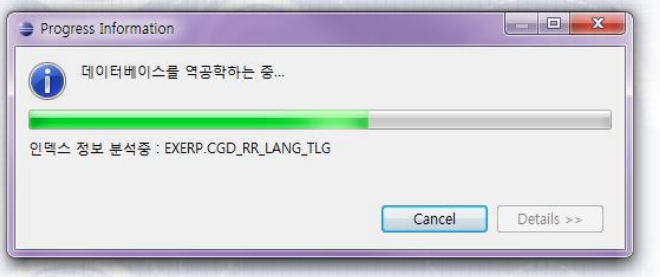
4) 리버스 엔지니어링으로 가져올 객체 종류를 선택하고 소유자 옵션을 설정합니다.



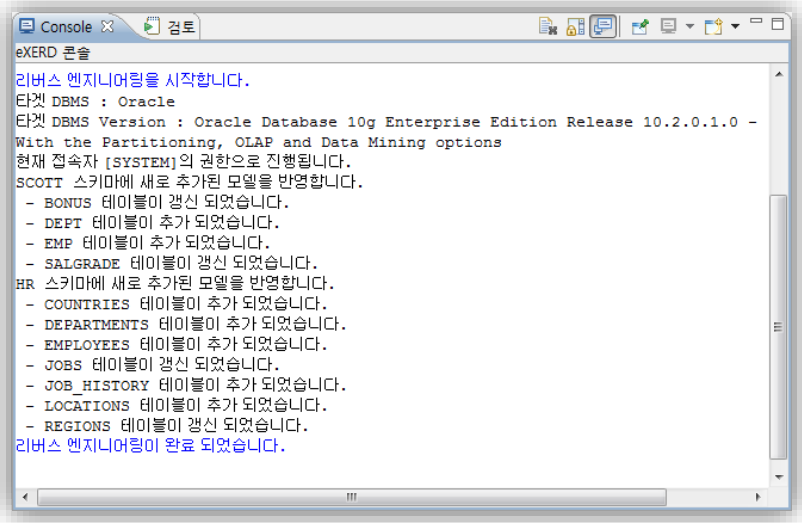
5) **[Finish]** 버튼을 누르면 리버스 엔지니어링을 시작합니다.



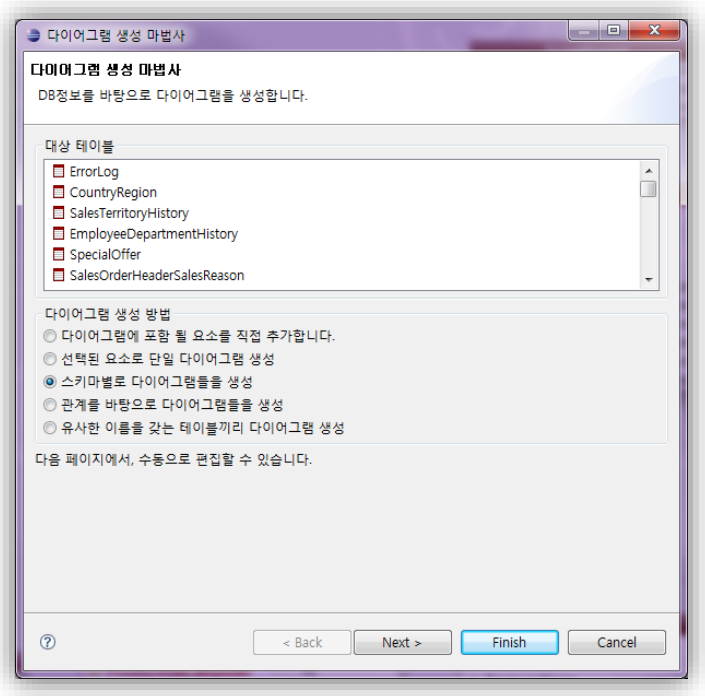




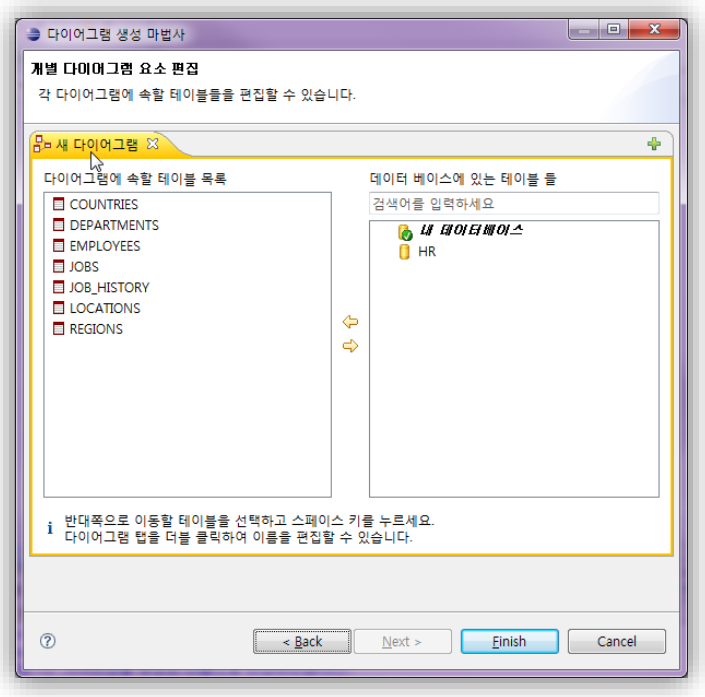
6) 리버스 엔지니어링 수행 정보가 **Console 뷰**에 기록됩니다.



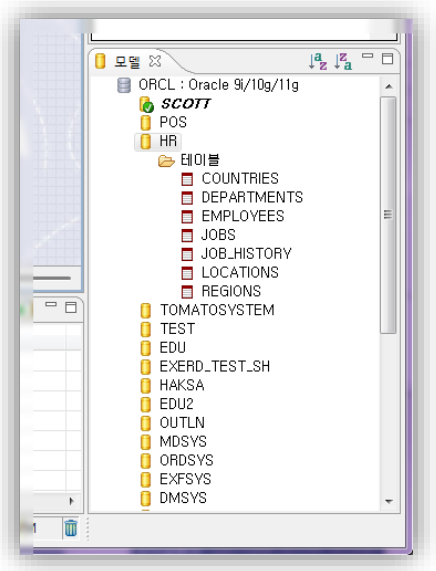
7) 새로 추가된 모델들을 대상으로 다이어그램 생성 마법사가 연동되어 진행됩니다.



다이어그램의 이름을 변경하거나 대상 테이블을 추가 및 제외할 수 있습니다.



8) 리버스 엔지니어링이 완료되면 **모델 뷰**에 새로운 스키마와 모델들이 추가됩니다.

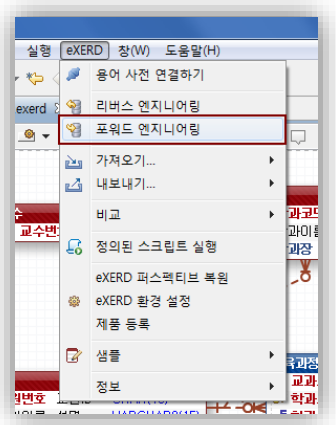


추가된 모델들은 eXERD 에디터의 다이어그램으로 드레그하여 새로운 다이어그램을 작성하거나 기존 다이어그램에 추가 할 수 있습니다.

2. 포워드 엔지니어링

준비된 E-R 모델을 DBMS 에 DDL 을 이용하여 생성하는 기능 입니다.

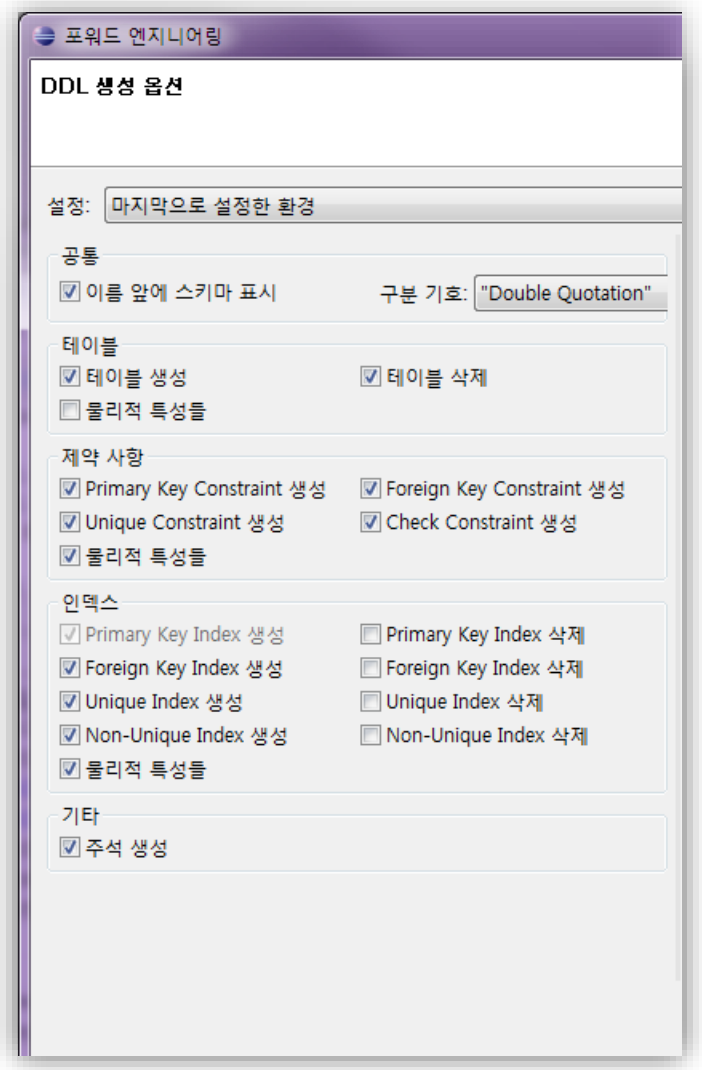
1) **eXERD > 포워드 엔지니어링** 메뉴를 선택합니다.



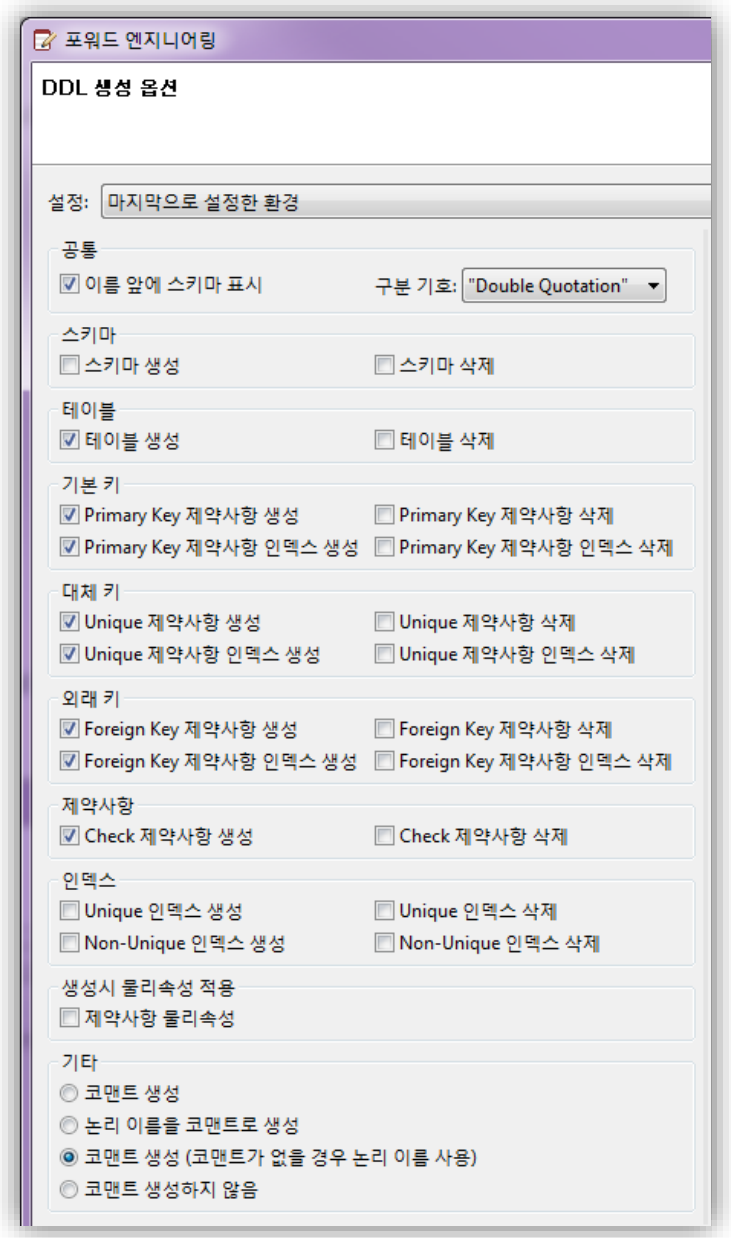
2) 포워드 엔지니어링 위저드가 실행됩니다. DDL 생성 옵션들을 설정합니다.

- 대상 DBMS 가 **Oralce** 인 경우 DDL 생성 옵션

- 대상 DBMS 가 SQL Server 인 경우 DDL 생성 옵션



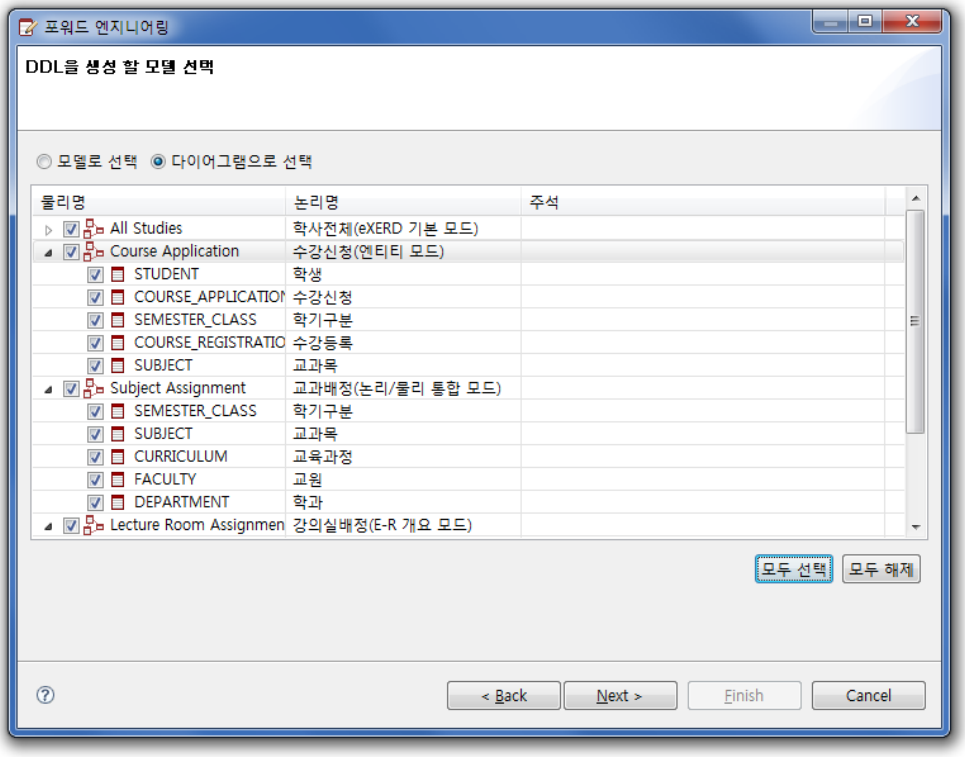
- 대상 DBMS 가 DB2 인 경우 DDL 생성 옵션



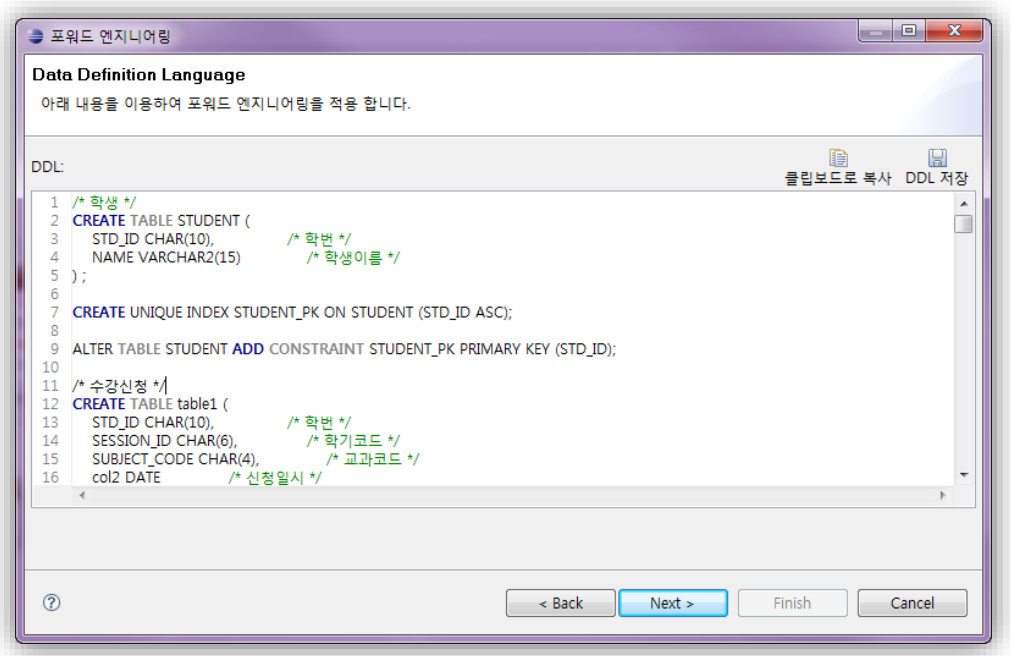
- 대상 DBMS 가 MySQL 인 경우 DDL 생성 옵션

Tip! *설정한 옵션 구성을 저장하면 재사용 할 수 있습니다.*

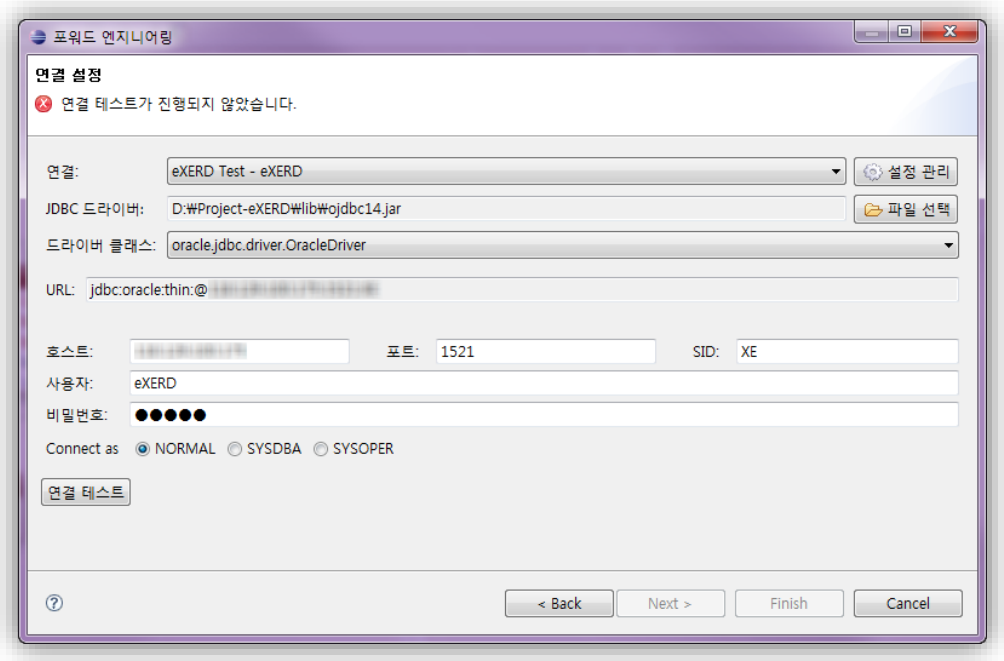
3) DDL 을 생성할 대상 모델을 선택합니다. 스키마(모델) 기준으로 테이블을 선택하거나 다이어그램을 기준으로 선택할 수 있습니다.



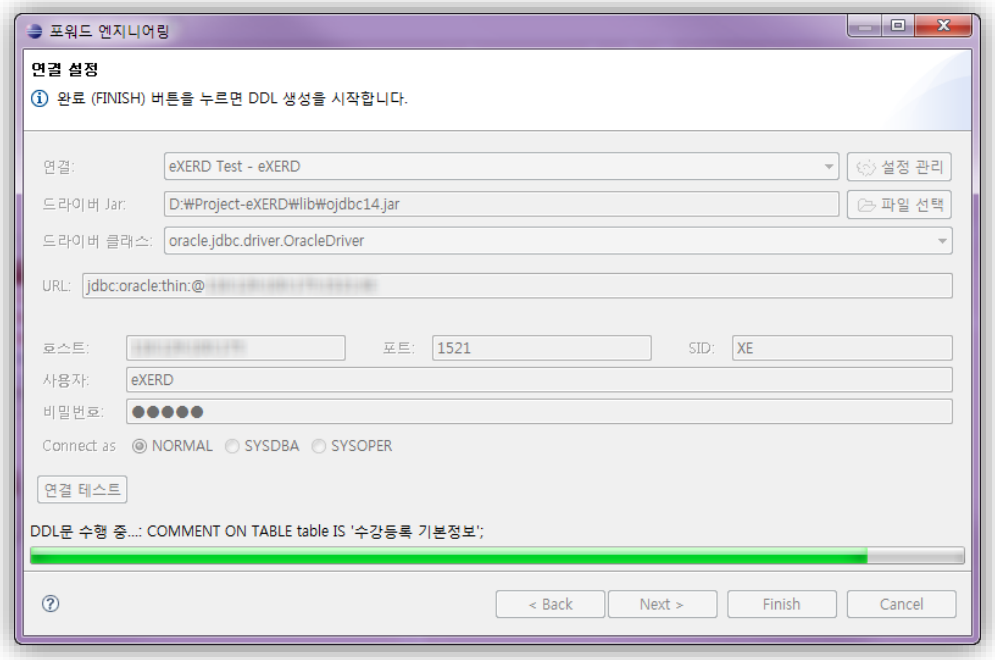
4) 선택한 옵션들을 기준으로 생성한 DDL 이 표시되어 복사하거나 파일로 저장할 수 있습니다.



5) 위저드에서 DDL 을 실행할 경우 진행하여 DBMS 접속 설정을 합니다.



6) DBMS 에 DDL 을 실행합니다.



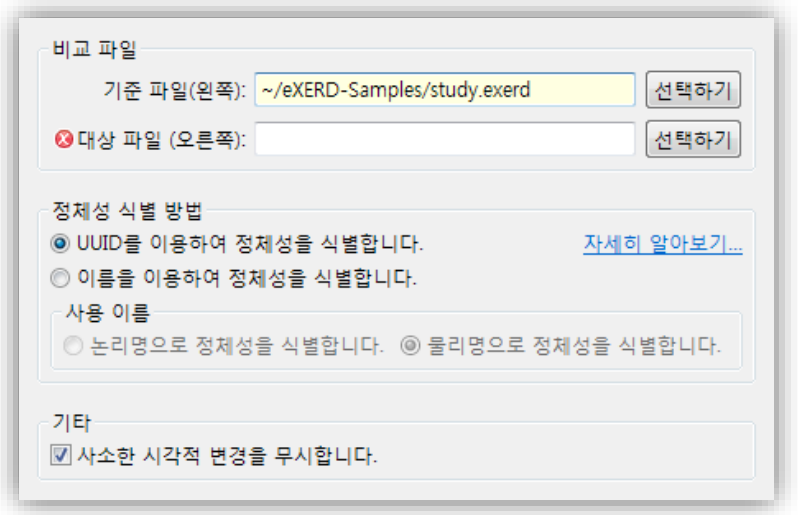
3. 모델 비교

모델 비교 전략

eXERD 에서 비교 전략은, 정체성 식별 방법 및 비교 내용에 대한 설정으로 구성됩니다.

■ 비교 전략 에디터

비교전략은 대부분 자동으로 수립되지만, 사용자의 확인이 필요하거나, 사용자가 강제로 비교 전략을 바꾸기 원하는 경우 아래와 같은 비교 전략 다이얼로그가 나타나게 됩니다:



■ 정체성 식별 방법

정체성 식별 방법이란, 비교 중인 두 문서에서 두 특정 요소가 동일한 요소인지 아닌지를 판단하는 방법입니다. eXERD 에서는 두 가지 방법이 제공 됩니다.

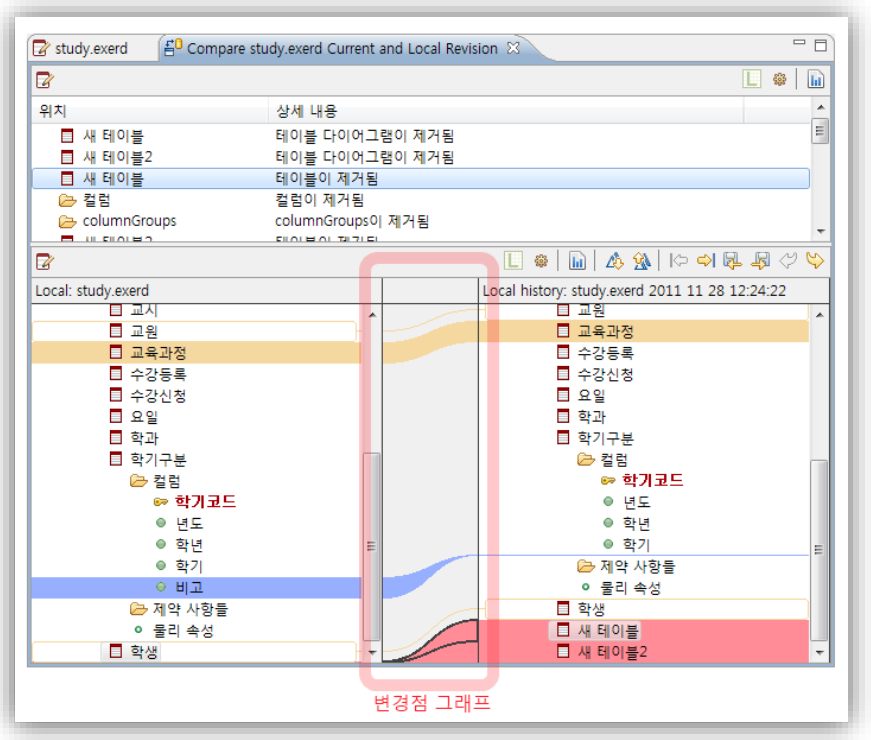
방식	적용 가능 대상	설명
UUID 기반	-같은 파일이 복제되어 수정된 경우 -같은 파일의 과거 버전과 비교하는 경우	eXERD 는 편집기 상에서 새로운 요소가 만들어지면 고유의 ID 를 발급합니다. 이를 이용하여 동일성 여부를 판단합니다. 논리명이나 물리명이 바뀌더라도 개체의 변경사항을 추적할 수 있습니다. 이 방식은 한 에디터 내에서 편집되었거나, 복제된 파일로 작업한 경우 사용할 수 있지만, 전혀 관계가 없거나, DB 와의 비교시에는 사용할 수 없습니다.
이름 기반	-별개로 만들어진 두 eXERD 파일 -DB와 파일간의 비교	이 방식은 물리명이나 논리명을 이용하여, 동일 정체성 여부를 판단합니다. 이름이 동일하면 동일한 대상으로 보게 됩니다. 만약 이름이 바뀐 경우, 기존 요소가 제거되고 새로운 요소가 추가된 것으로 인식하게 됩니다.

일반적으로 정체성 판단 기법은 자동으로 설정되지만, 불가피한 경우 사용자에게 질의하거나, 사용자가 강제로 판단 기법을 선택할 수 있습니다.

■ 기타

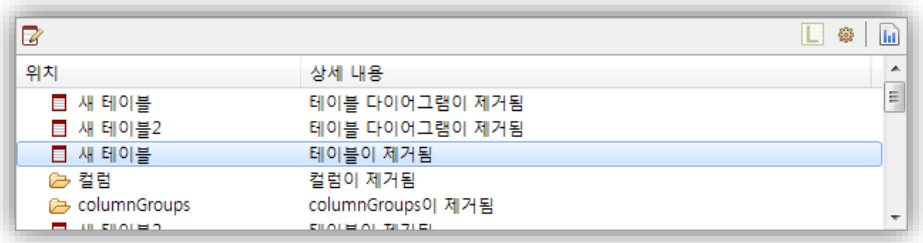
사소한 시각적 변경: '사소한 시각적 변경'이란, 테이블 다이어그램의 위치가 이동되거나, 사용된 테마가 변경되는 등, 실제 DB 에는 변화가 없는 시각적인 변경을 의미합니다.

■ 비교 에디터



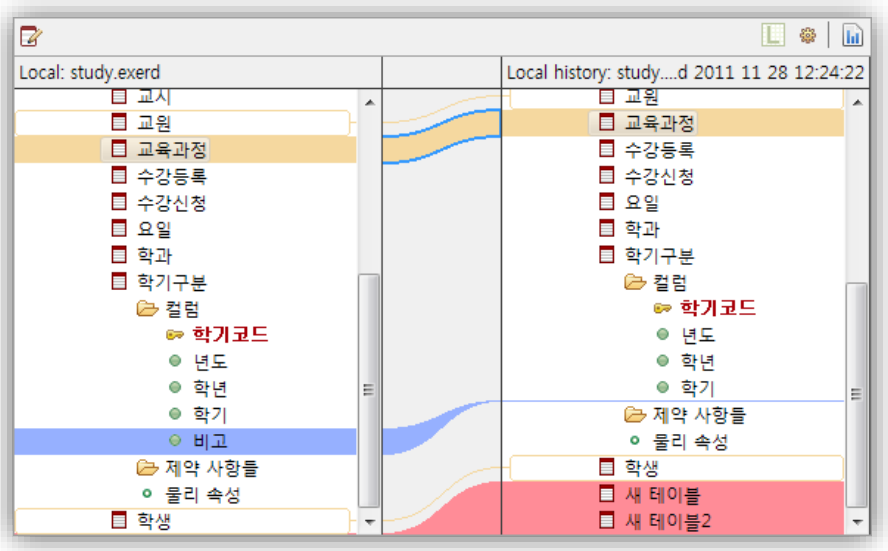
비교 뷰어의 상단에는 차이점 목록이, 아래쪽에는 트리 구조로 변경 내역에 대한 그래프가 나타납니다.

■ 차이점 목록 뷰어



차이점 뷰어의 상단에는 비교 요소들의 목록이 나열되어 있습니다. 각각의 차이점을 더블 클릭하면, 실제 변경 위치를 기준으로 변경점이 표현된 아랫쪽 뷰어에서 그 위치가 선택됩니다.

■ 구조적 차이점 뷰어



구조적 차이점 뷰어는 두개의 트리형태로 차이점을 그래프로 보여줍니다. 각 차이점은 다음과 같은 색상으로 나타내 집니다:

- 파랑: 새로 추가된 요소
- 빨강: 사라진 요소
- 노랑: 변경된 요소

● 밀치기

밀치기란 미는 쪽의 내용으로 밀쳐지는 쪽의 내용을 덮어써 차이점을 제거하는 동작입니다. 왼쪽에서 오른쪽으로 밀칠 경우, 오른쪽의 내용을 왼쪽의 내용으로 덮어 쓰게 됩니다.

차이점을 나타내는 그래프를 선택 하고, ⇦또는 ⇨버튼을 눌러, 차이점을 특정 방향으로 밀칠 수 있습니다.

⇨ 왼쪽의 내용으로 오른쪽을 덮어 씁니다. 예를 들어 오른쪽에 있던 내용이 왼쪽에서 사라진 경우, 오른쪽으로 밀면 오른쪽에서도 사라집니다.

⇦ 오른쪽의 내용으로 왼쪽을 덮어 씁니다. 예를 들어 왼쪽에는 없는 항목이 우측에 있는 경우, 왼쪽에도 해당 항목이 추가됩니다.

밀치기 작업은 ⇦또는 ⇨버튼을 이용하여, Undo / Redo를 수행할 수 있습니다.

● **밀치기 실패**

밀친 후, 모델의 정합성이 파괴되는 경우, 경우 밀치기는 실패하며, 밀치기 동작이 취소될 수 있습니다. 예를 들어 다음과 같은 경우 밀치기는 실패합니다:

① 밀치면 삭제될 테이블을 참조하는 자식테이블이 존재하는 경우.

② 밀쳐서 추가할 테이블이 참조하는 부모 테이블이 밀쳐질 문서에는 포함되어 있지 않은 경우.

이런 경우 여러 테이블 변경점 그래프를 동시에 선택하고 밀치면 해결 할 수 있습니다.

상기 경우를 제외하고도, 사용자의 명령에 따라 안전한 참조 복사가 불가능한 경우, 실패가 발생할 수 있으며, 이유가 표시됩니다.

● **저장**

비교 편집 중인 좌우 문서중 원하는 문서를 저장 할 수 있습니다.

비교 편집중인 왼쪽 문서를 저장합니다.

비교 편집중인 오른쪽 문서를 저장합니다.

● **기타**

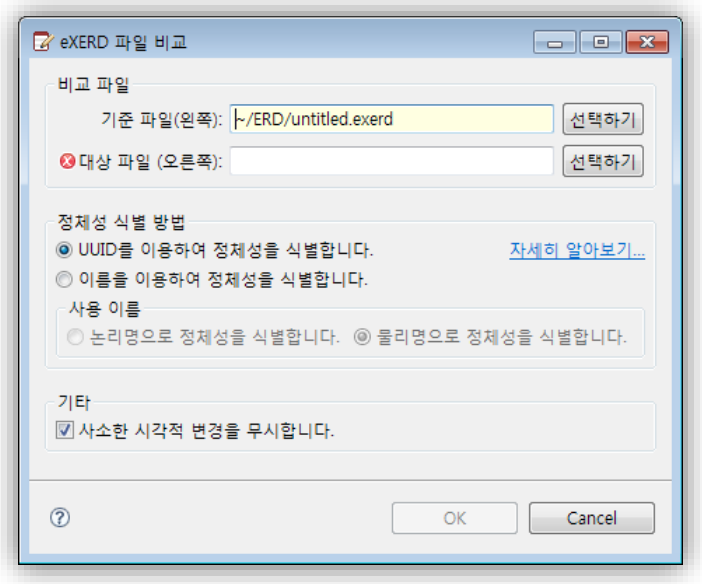
변경점 편집기가 표시할 이름을 논리/물리 모드로 전환합니다.

다른 비교 전략을 이용하여 다시 비교합니다.

HTML 비교 보고서 어플리케이션을 생성합니다.

■ 파일 비교

1) eXERD 파일과 파일을 비교하려면, 시스템 메뉴에서 eXERD > 파일 비교를 차례로 선택하십시오.



2) 비교할 두 파일을 선택합니다.

3) 정체성 식별 방법 및, 기타사항에 대해서는 비교 전략 문서를 참조 하십시오.

4) 비교 뷰어가 열립니다.

Tip! 두 eXERD파일을 선택하고 우클릭하면, "선택된 두파일을 비교하기" 메뉴를 선택할 수 있습니다.

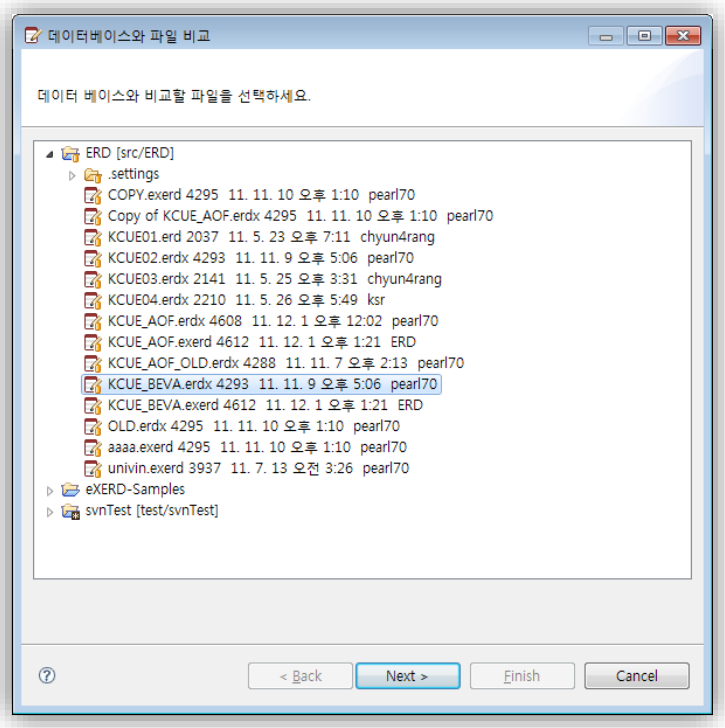
● 기타 정보

이클립스의 제한사항으로 인하여, exerd파일과 erdx(XML 형식)의 파일을 비교할 수 없습니다. 한 쪽을 열어 다른이름으로 저장하여 동일하게 지정하여 비교하여야 합니다.

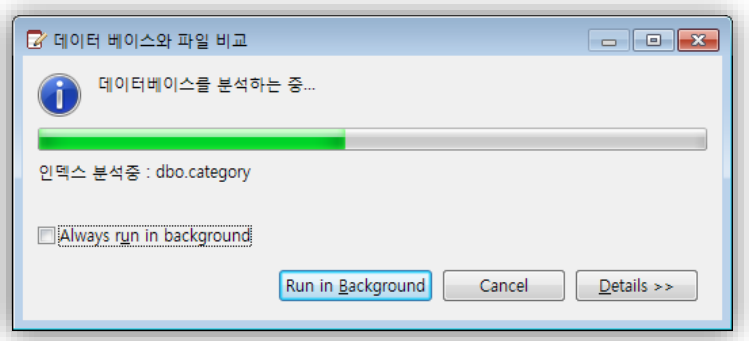
예를 들어 "A.exerd" 파일과 "B.erdx" 파일이 있다면, "B.erdx" 파일을 열어 '다른이름으로 저장'을 선택 하여 이름을 "B.exerd"로 바꾼 뒤 비교할 수 있습니다.

■ 데이터 베이스와 파일 비교

1) 시스템 메뉴 > eXERD > 데이터 베이스와 비교 를 차례로 선택합니다.
데이터 베이스와 비교할 파일을 선택하고 [다음]을 누릅니다.



2) 데이터 베이스에서 가져올 정보들을 설정합니다. "리버스 엔지니어링 도움말"을 참조할 수 있습니다.

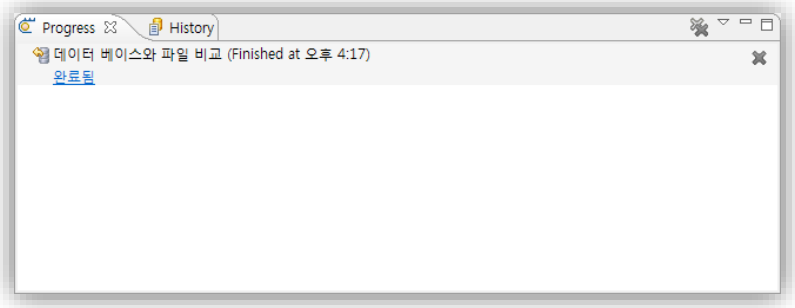


● **기다리는 동안 다른 작업 하기**

[Run in Background] 버튼을 눌러 데이터 베이스에서 정보를 가져오는 동안 다른 작업을 수행할 수 있습니다.

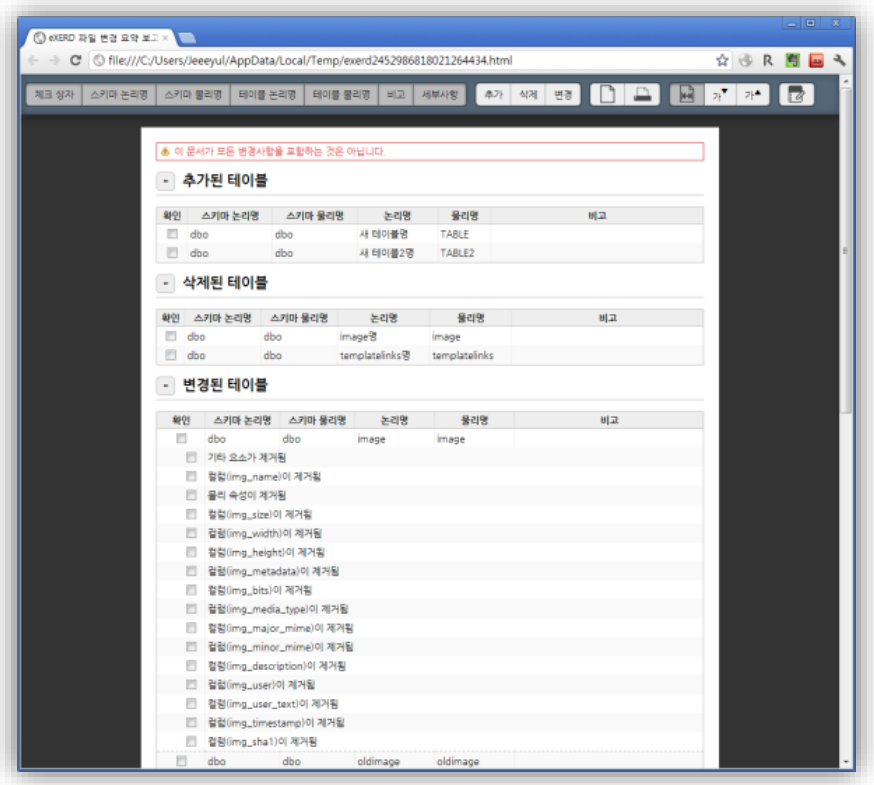
작업이 완료되면 우측 하단에, 작업 완료 아이콘 을 클릭하여, 비교작업을 계속 진행 할 수 있습니다.

또는 프로그레스 뷰에서 완료 링크를 클릭하여 작업을 계속 할 수 있습니다.

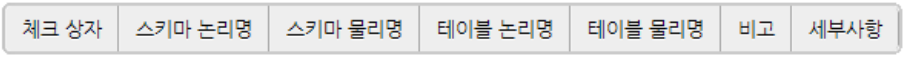


■ HTML 비교 보고서

비교 뷰어 에서 보고서 생성 기능을 이용하여, HTML5 보고서 애플리케이션을 생성할 수 있습니다



● **툴바: 컬럼 보이기 / 감추기**



각 버튼들을 클릭하여, 보고서에 표시할 컬럼을 켜거나 끌 수 있습니다.

● **툴바: 인쇄**

용지의 가로/세로 방향을 전환합니다.

현재 설정대로 인쇄합니다.

● **툴바: 기타 설정**

보고서의 테이블들을 용지 너비에 맞춥니다.

가▲ 글꼴을 더 크게 합니다.

가▼ 글꼴을 더 작게 합니다.

만약 글꼴이 원하는 크기만큼 작아지지 않는다면, 브라우저의 최소 글꼴 크기 제한이 설정되어 있는지 확인 하십시오.

- **기타**

① 각 섹션의 제목을 클릭하거나 펼치기 접기 버튼을 눌러, 섹션을 접을 수 있습니다. 접힌 섹션은 인쇄되지 않습니다.

② 보고서를 보관하려면 브라우저의 저장기능을 이용하여 다른 이름으로 저장하십시오. eXERD가 종료될 때, 따로 저장되지 않은 보고서는 삭제됩니다. 보고서는 한 개의 HTML 파일로만 이루어져 있기 때문에, 쉽게 공유할 수 있습니다.

4. 용어 사전

■ 표준 용어사전 활용

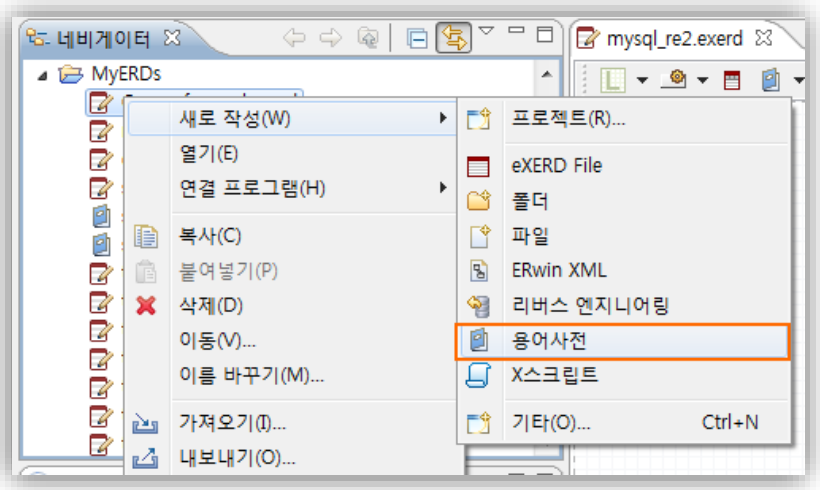
E-R 다이어그램을 작성할 때 사용할 표준 용어와 그 용어의 물리명을 정의하여 표준 용어 사전을 만들 수 있습니다.

표준 용어사전을 eXERD E-R 다이어그램 파일에 연결하면 컬럼의 논리명을 입력 시 물리명이 자동으로 입력되고 표준 용어 규칙의 준수 여부를 검토할 수 있습니다.

● 용어사전 만들기

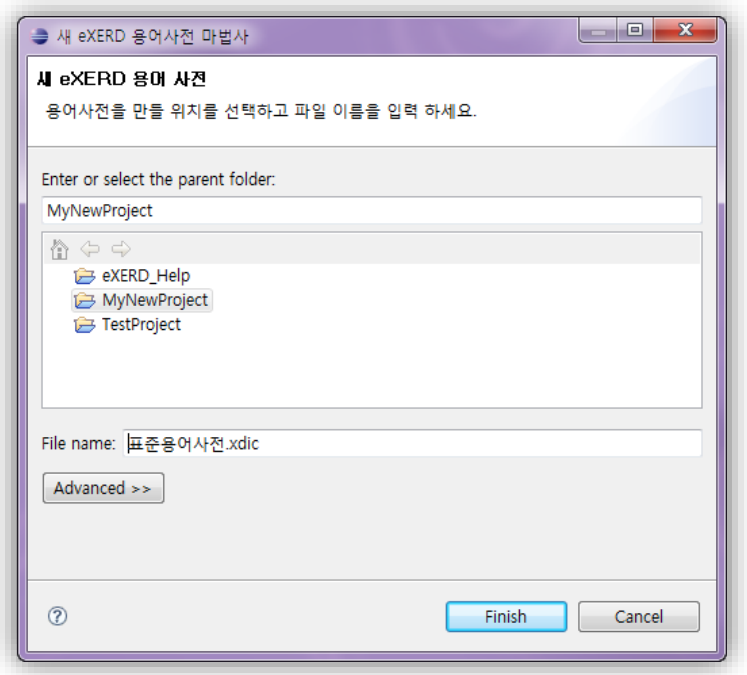
eXERD 용어사전은 다이어그램 파일과 별개의 **.xdic** 확장자를 가진 독립 파일로 만들어집니다.

① 메뉴에서 아래와 같이 **새 용어사전**을 실행합니다.



Note! *퍼스펙티브가* ***eXERD 퍼스펙티브****로 설정되어 있을 때 위와 같은 메뉴를 볼 수 있습니다.*

② 용어사전 이름을 입력합니다.

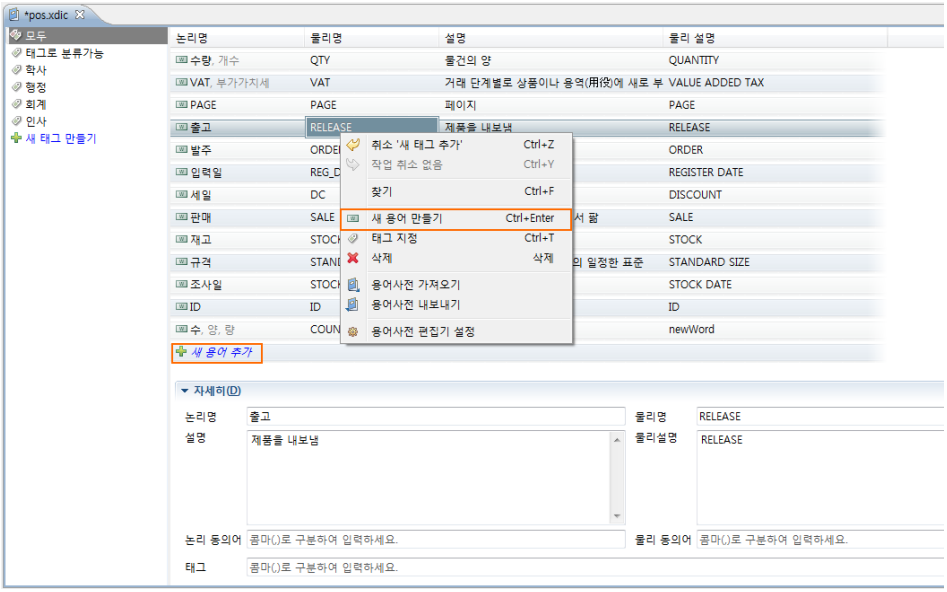


● 용어사전 에디터

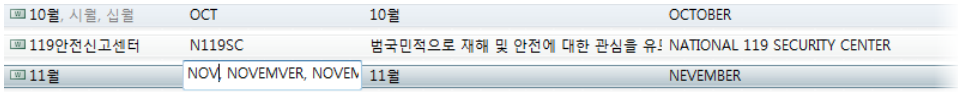
용어사전에 E-R 다이어그램에서 사용할 논리명과 물리명을 등록합니다.

Note! *일반적으로 논리명은 단어 단위의 명사를 입력하고 물리명은 영문 약어를 입력합니다.*

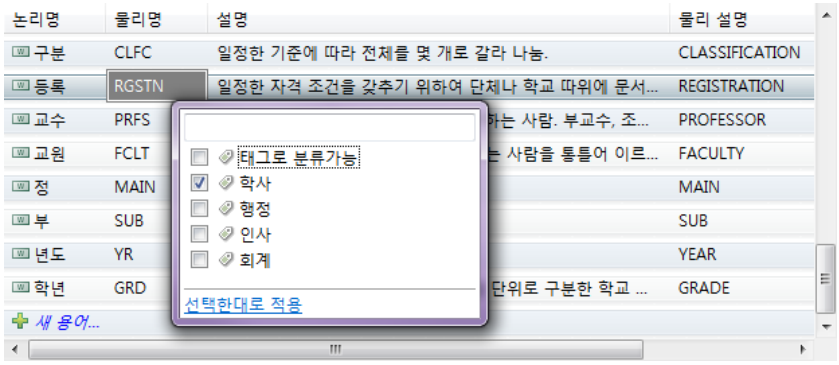
① **새 용어 추가**입력란에 용어를 입력하거나 **Ctrl + Enter**로 새 용어를 만듭니다.



Tip! *용어사전의 논리명과 물리명 컬럼에서 ';' 구분자로 동의어를 추가 할수 있습니다.*



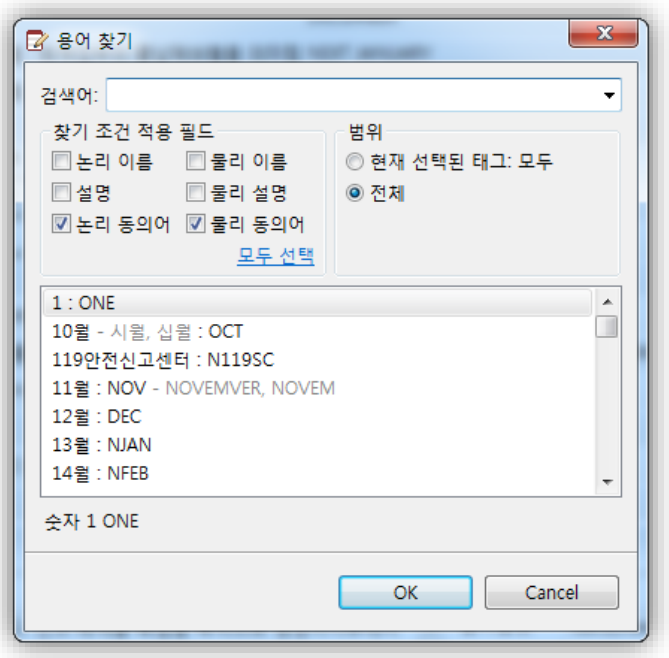
Tip! *태그 지정 기능을(**Ctrl + T**) 이용하면 용도에 따라 용어들을 분류하여 관리할 수 있습니다.*



● 용어사전 검색

용어사전의 필드를 선택하여 원하는 이름을 검색합니다.

② 용어사전 에디터 상에서 Ctrl + F(맥의 경우 Command + F)를 입력합니다.

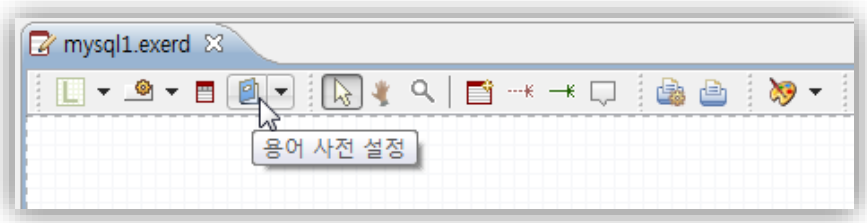


how to : *찾는 필드의 범위를 선택하고 이름을 검색한 다음 OK를 누르면 커서가 필드로 이동합니다.*

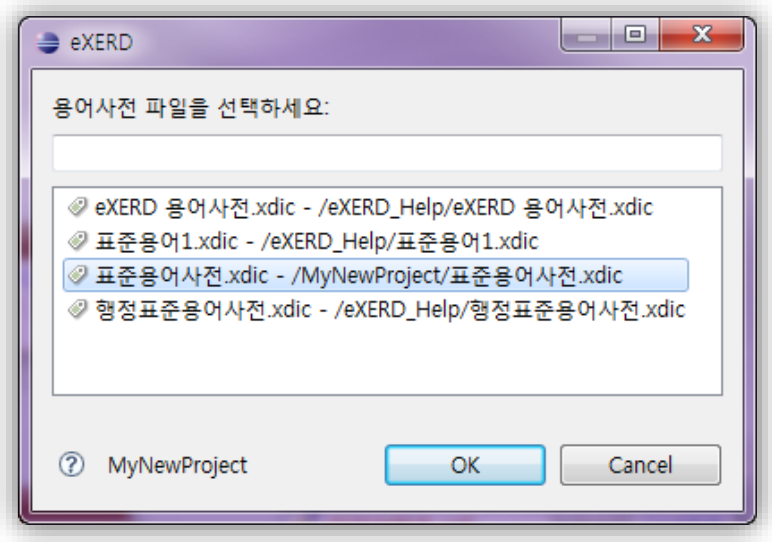
● 다이어그램에서 연결

준비된 용어사전을 eXERD 다이어그램 에디터에서 연결하여 E-R 다이어그램에 표준 용어 정책을 적용합니다.

① 툴바에서 **용어 사전 설정** 버튼을 실행합니다.



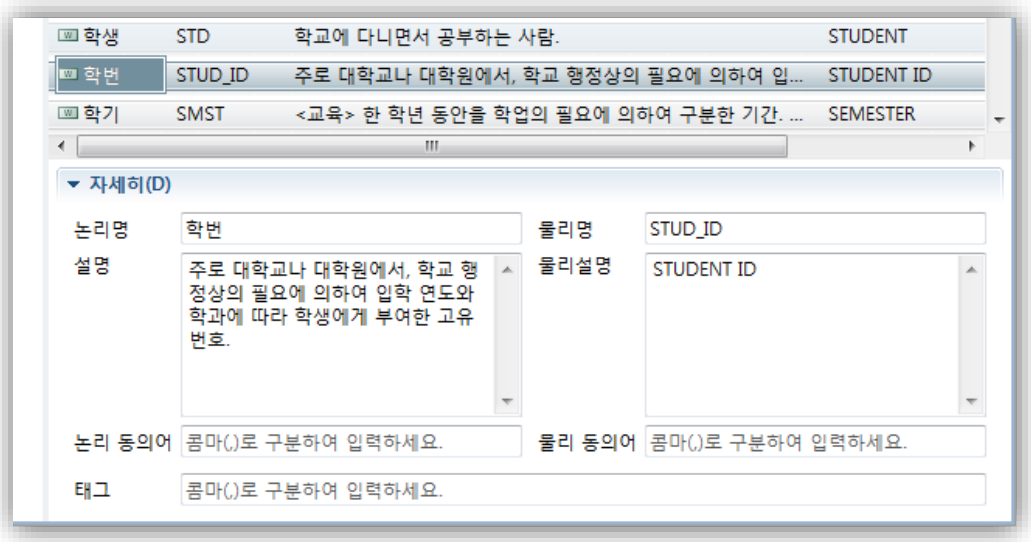
② 현재 E-R 다이어그램에 사용할 용어사전을 선택합니다.



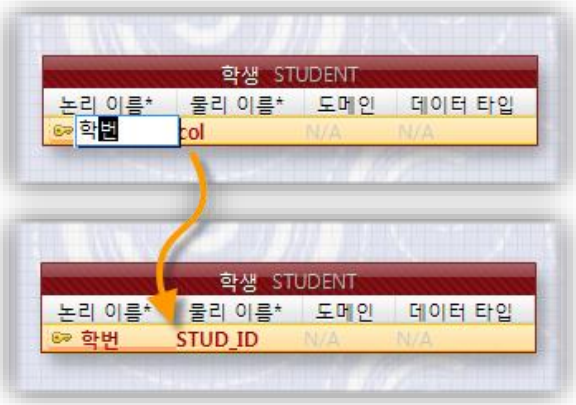
● 자동으로 물리명 제안

컬럼의 논리명이 입력되면 용어사전에서 논리명에 사용된 단어들을 조사하여 물리명을 구성합니다. 이때 생성된 물리명 또는 물리 합성어가 한개인 경우 자동으로 컬럼의 물리명으로 입력됩니다.

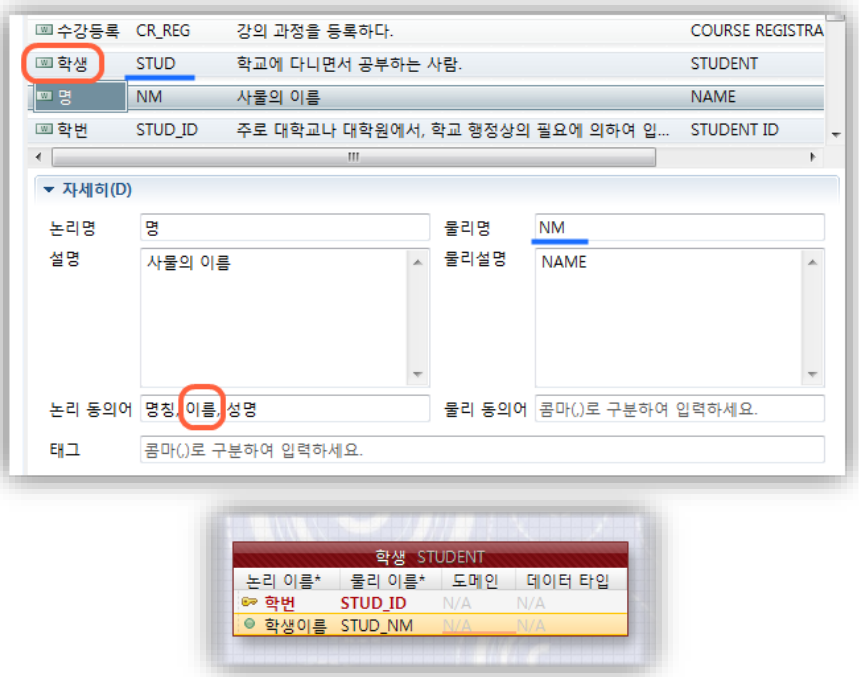
① 아래와 같이 **"학번"**용어가 등록되어 있는 용어사전을 연결합니다.



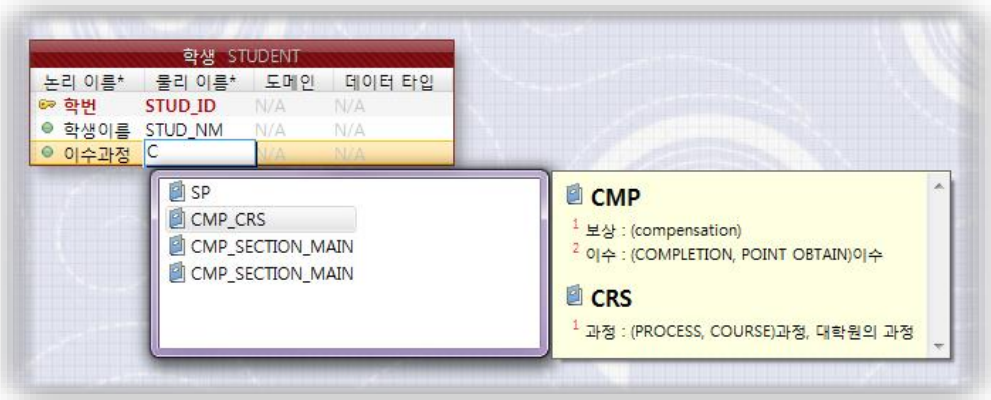
② 새 컬럼의 논리 이름에 **"학번"**을 입력하면 아래와 같이 물리 이름에 **"STUD_ID"** 가 자동으로 입력됩니다.



③ **"학생이름"**과 같은 합성어 역시 용어사전에서 **"학생"**의 물리명 **"STUD"**와 **"이름"**의 물리명 **"NM"**으로 **"STUD_NM"**을 구성하여 입력됩니다.



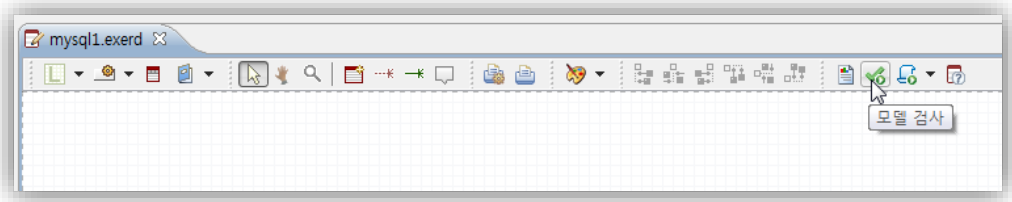
④ 용어사전에 등록된 용어가 동의어를 포함하여 여러개인 경우 구성 가능한 물리명도 여러개가 됩니다. 이러한 경우에는 아래와 같이 물리 이름을 편집할 때 선택 가능한 목록을 표시합니다.



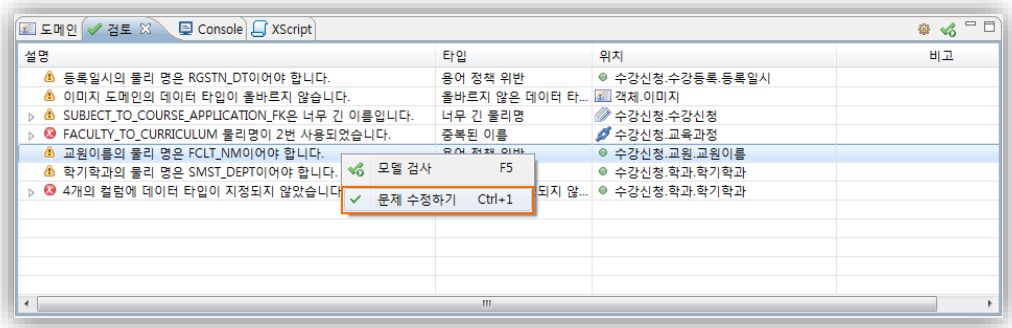
● 용어 정책으로 모델 검사

다이어그램의 전체 모델을 표준 용어사전을 이용한 용어 정책에 부합하는지 여부를 검사 할 수 있습니다.

① 툴바에서 **모델 검사** 버튼을 실행합니다.

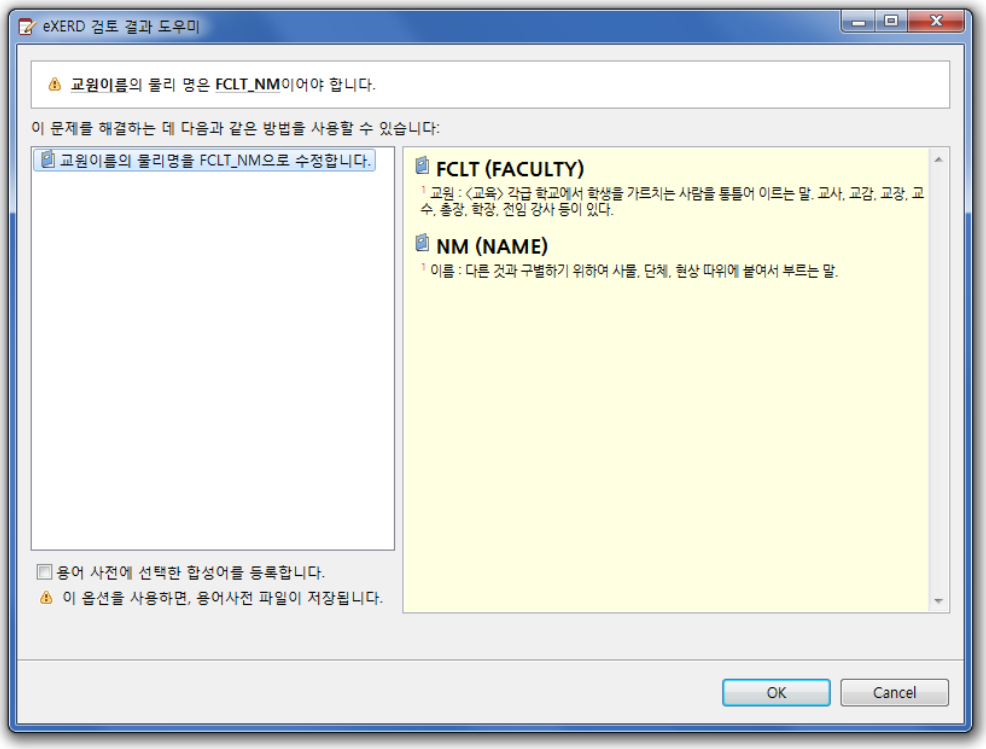


② 전체 모델을 검사한 결과가 **검토**뷰에 표시됩니다.



Tip! *항목을 더블클릭하면 다이어그램에서 해당 항목이 선택됩니다.*

③ **문제 수정하기 (Ctrl + 1)**를 실행하여 용어 정책에 맞는 물리명으로 적용할 수 있습니다.



● 용어 정책 일괄 적용

용어 정책 일괄 적용 기능을 이용하면, 용어사전을 기반으로, 물리명을 일괄 적용할 수 있습니다. 이 작업은 크게 네 단계로 나누어 집니다:

① 용어 정책 적용 계획 수립

② 자동 적용 가능한 물리명 계획 검토

③ 적용 가능한 물리명이 복수인 경우에 대한 계획 검토 및 수정

④ 계획 실행

1) 계획 작성

용어 사전 일괄 적용을 수행하기 위한 첫 단계는 계획을 세우는 것입니다. 적용 계획은 사용자가 정의한 옵션을 바탕으로 자동으로 수립되며, 적용 계획이 작성된 후, 다음 단계로 넘어갈 수 있습니다. 컬럼 및 용어사전의 규모에 따라 작업 시간이 길어 질 수 있으며, 다중 CPU가 탑재된 PC의 경우 더 빠른 속도로 계획이 수립됩니다.

적용 계획 생성 옵션:

- **용어사전과 일치하는 물리명을 제외하고 용어사전 적용 계획을 만듭니다.**

  이 옵션을 선택하면, 용어 사전에 일치하는 물리명들을 제외하고 적용 계획을 생성합니다. 용어사전 정책에 따라 여러 물리명을 가질 수 있는 경우, 물리명 중 하나와 일치하는 경우, 정책에 합치하는 것으로 보고, 계획 생성에서 무시됩니다.

- **대체 가능한 물리명이 존재하는 경우를 포함하여 용어사전 적용 계획을 만듭니다.**

이 옵션은 적용 가능한 물리명이 하나 밖에 없고, 이에 일치하는 물리명을 가진 논리명들을 제외하고 용어사전 적용 계획을 만듭니다. 용어사전과 일치하는 물리명을 소유한 물리명이라고 할 지라도, 대체 가능한 물리명이 존재하는 경우, 적용 계획에 포함됩니다.

- **모든 물리명에 대해 용어사전 적용 계획을 만듭니다.**

  이 옵션은 모든 논리명에 대해, 이미 정책에 부합하는 물리명을 가졌는지 여부와 상관없이, 정책 적용 계획을 작성합니다.

  필요한 옵션을 선택 한 뒤, [계획 작성] 버튼을 누르면, 계획이 생성되고, 생성이 완료되면 다음 단계로 넘어갑니다.

2) **물리명 자동 적용 계획 검토**

이 단계에서는 적용 가능한 물리명이 유일하여, 자동 적용이 가능한 논리명들에 대해 검토를 수행 합니다. 동일한 논리명은 그루핑 되어 나타나며, 각 항목의 체크 박스를 이용하여, 용어 사전의 물리명을 적용할 것인지 여부를 결정 할 수 있습니다.

작업하기 편리한 크기로 창의 크기를 조절 할 수 있으며, 이는 기억 됩니다.

3) **애매 모호성 검토**

이 단계에서는 적용 가능한 물리명이 복수 인 경우, 가능한 물리명 중 하나를 선택할 수 있도록 도와 줍니다.

노란색 영역은 직접 편집이 가능하며, 클릭하면 대체 가능한 다른 물리명들이 나열 됩니다. 다른 물리명을 선택하면, 용어사전의 설명을 함께 볼 수 있습니다. 이 값이 편집되면 자동으로 체크 박스가 채워져, 적용 계획에 포함됩니다.

초기 값으로 적용가능한 첫 번째 물리명이 씌여져 있지만, 체크 박스가 비워져 있음을 유의해 주십시오.

● 용어사전 역공학

용어사전 역공학 기능을 이용하면, 용어사전에 등록되지 않은 논리명을 사용한 후 용어사전에 등록하고 싶은 경우 일괄적으로 등록하게 도와줍니다.

이 기능은 다이어그램과 용어사전이 연결된 상태에서만 활성화됩니다.

① 다이어그램의 툴바에서 **용어사전 역공학** 버튼을 실행합니다.

② 작업 창이 활성화 되면 용어사전에 없는 컬럼들을 나열합니다.

③ **편집 영역**에 필요한 작업을 마치고 Finish를 실행합니다.

Note! *하나의 논리명에 여러개의 물리명이 있으면 대표 물리명을 선택 편집이 가능합니다. 나머지는 물리 동의어로 등록됩니다.*

● 용어사전 가져오기/내보내기

Excel로 작성한 용어들을 사전으로 가져오거나 용어사전에서 Excel로 내보내기를 할수 있습니다.

■ Excel 작성하기

용어 사전에 가져오려는 내용을 아래의 양식으로 Excel에 작성합니다.

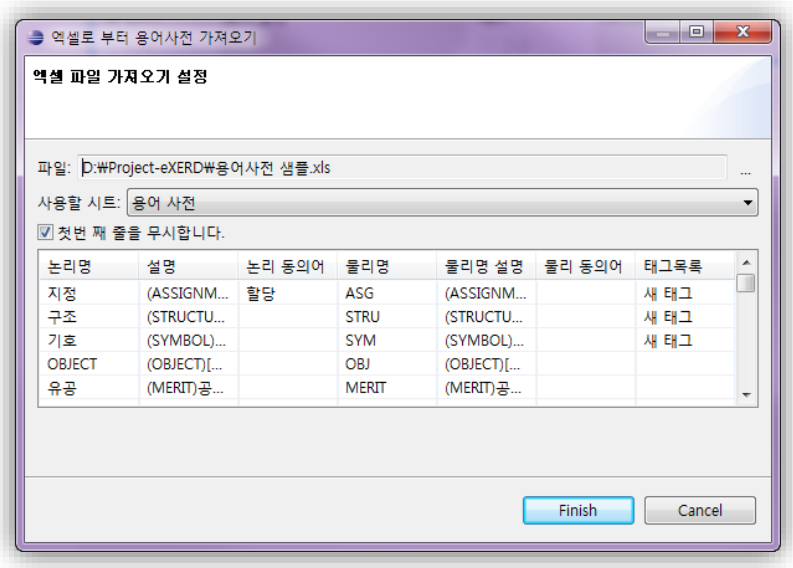
	A	B	C	D	E	F	G
1	논리명	논리설명	논리동의어	물리명	물리설명	물리동의어	태그
2	...	...	...	...	...	...	...

위와 같이 Excel에서 각 Cell에 값을 넣어 주시면 됩니다.

Excel은 97-2003(xls)을 지원 합니다. 저장 하실 때 확인해 주시기 바랍니다.

■ **Excel로 가져오기**

팝업 메뉴에서 **용어사전 가져오기** 기능을 이용하면 엑셀 파일로도 작성할 수 있습니다.



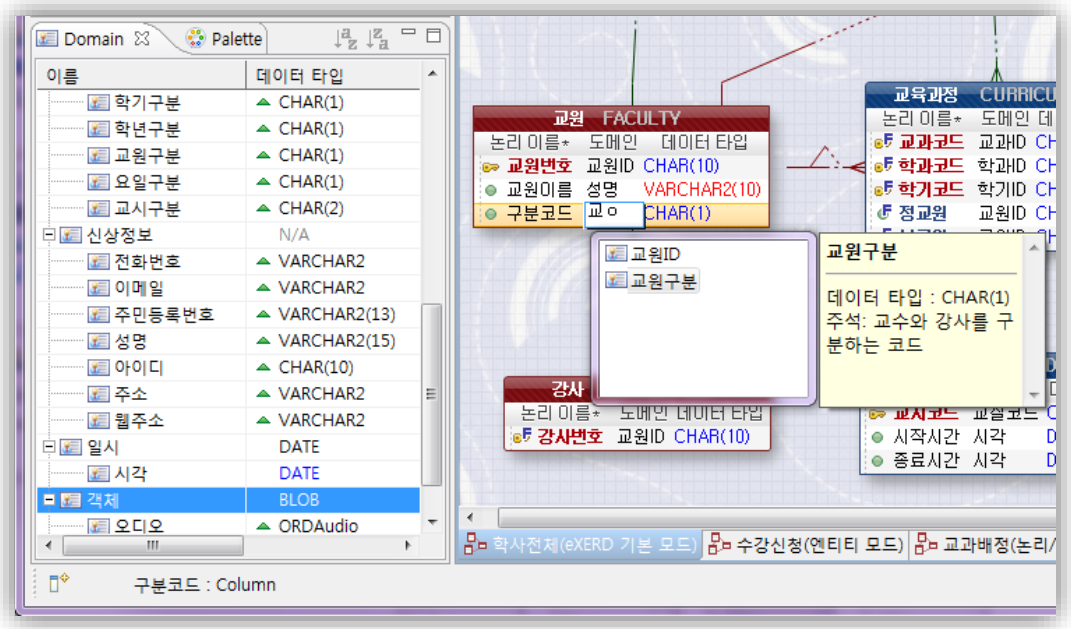
■ Excel로 내보내기

용어사전의 내용을 Excel(xls)로 내보냅니다.

5. 도메인 사전

도메인 타입을 구성하고 활용할 수 있습니다. 도메인을 사용하면 의미상 동일한 컬럼들에 대해 일관된 규칙을 적용할 수 있습니다.

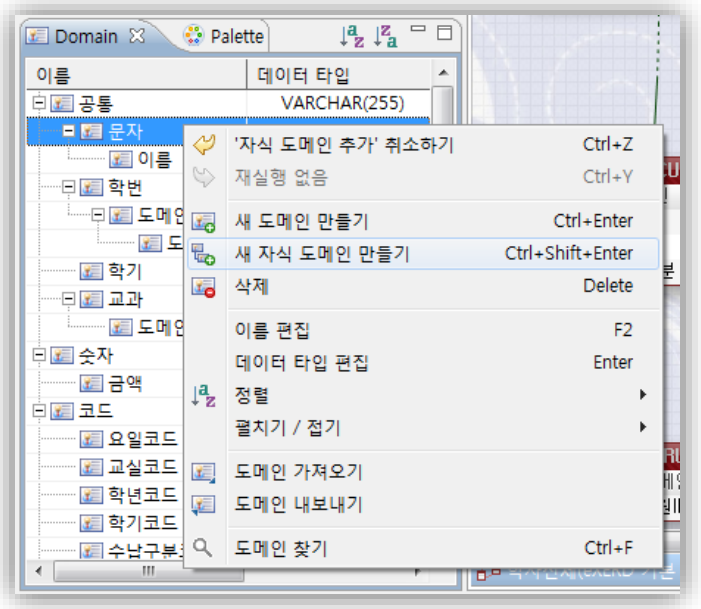
Note! *도메인 뷰에서 하위 도메인의 상속된 데이터타입은 파란색으로 표시되며 타입을 재정의 한 경우 ▲표시가 나타납니다.*



- 도메인 뷰에서 미리 등록하거나 컬럼 편집중에 새 도메인 항목을 지정하면 바로 등록됩니다.
- 컬럼에 도메인을 설정한 경우 데이터타입이 파란색으로 표시됩니다.
- 도메인이 설정된 컬럼의 데이터타입을 직접 지정한(타입을 재정의한) 경우에는 빨간색으로 표시됩니다.

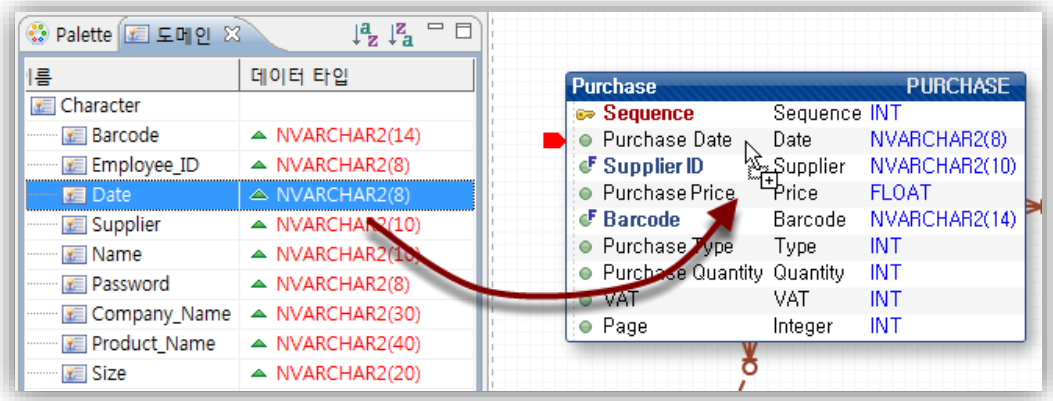
■ 도메인 구성하기

도메인 뷰에서 단축키 및 팝업메뉴를 이용하여 도메인을 구성할 수 있습니다.



- 가져오기 / 내보내기를 이용하여 도메인을 공유하거나 관리할 수 있습니다.
  - 도메인을 XML 파일(**.xdmn**)로 내보내거나 가져올 수 있습니다.
  - Microsoft Excel 파일(**.xls**)로 내보내거나 가져올 수 있습니다.
- 도메인 뷰에서 하위 도메인을 만들어서 관리할 수 있습니다.
- 기존 도메인을 마우스로 Drag 하여 하위 도메인으로 넣거나 뺄 수 있습니다.
- 도메인에 기본값을 정의할 수 있습니다. 정의된 기본값은 도메인이 지정된 컬럼에 반영됩니다.

Tip! *도메인 뷰에서 테이블의 컬럼으로 도메인을 드레그 & 드랍하여 지정하는 것도 가능합니다.*



6. 모델 검사

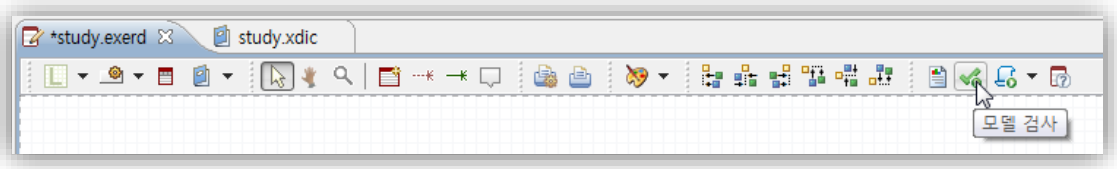
편집중인 전체 E-R 모델의 논리적/물리적 다양한 오류 및 유효성을 일괄 검사하고 수정 할 수 있습니다.

Note! *포워드 엔지니어링을 실행하기 전에 검사 기능을 이용하면 대부분의 오류를 미리 해결할 수 있습니다.*

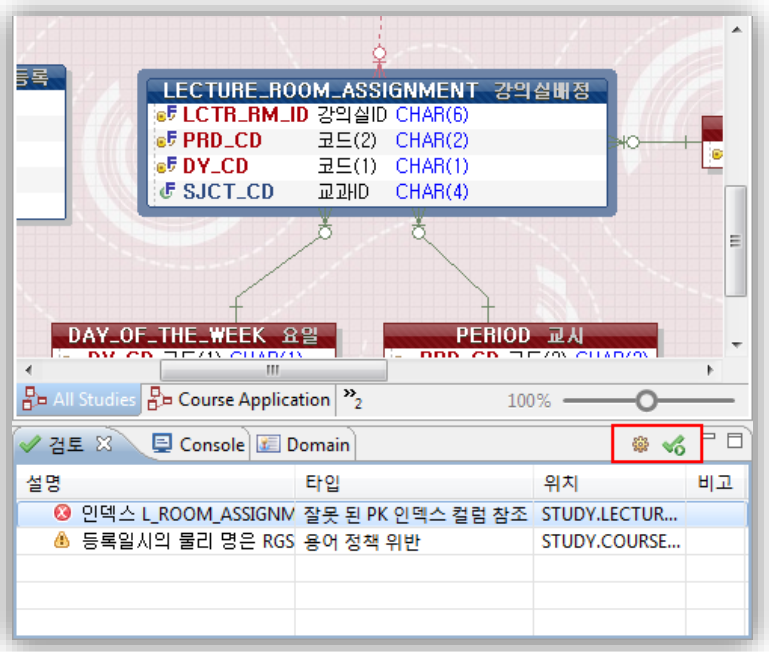
■ 모델 검사하기

모델 검사를 통하여 올바르지 않은 모델의 상태를 확인하고 손쉽게 수정 할 수 있습니다.

- 툴바에서 **모델 검사** 버튼을 실행합니다.



- 전체 모델에 대한 검사 결과가 **검토 뷰**에 표시됩니다.



Tip! *모델 보고서 항목을 더블 클릭하면 해당 모델이 다이어그램에서 선택됩니다.*

■ 검사하는 내용들

대부분 사용자의 실수 및 아직 설정(입력)되지 않은 필수 요소들에 대해서 모델 검사를 수행합니다.

▪ 공통

- ○ 물리이름이 용어사전에 정의된 물리명과 일치하는지 검사합니다.
- ○ 이름이 지정되지 않은 요소를 검사합니다.
- ○ 컬럼의 데이터 타입이 누락 되었는지 검사합니다.
- ○ 물리이름에 예약어가 사용 되었는지 검사합니다.
- ○ 어떤 다이어그램에도 사용되지 않은 테이블을 찾습니다.
- ○ 물리명이 DBMS의 이름길이 제약을 위반하는지 검사합니다.

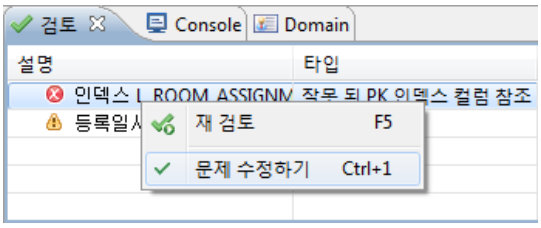
▪ Oracle (9i / 10g / 11g)

- ○ 테이블 스페이스, 제약사항, 인덱스의 이름을 검사합니다.
- ○ 컬럼의 데이터 타입이 올바르게 지정되었는지 검사합니다.
- ○ 인덱스의 컬럼 설정이 유효한지 검사합니다.
- ○ 도메인의 데이터 타입이 올바르게 지정되었는지 검사합니다.

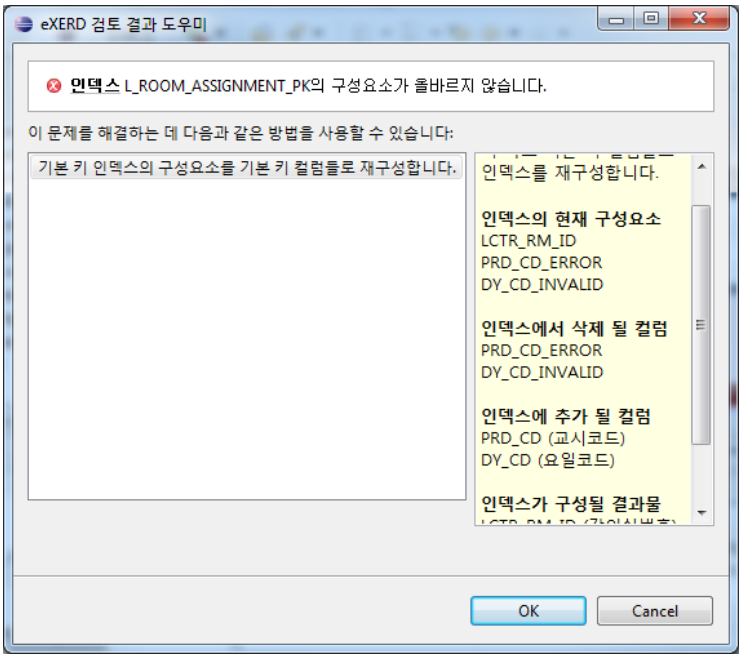
■ 문제 수정하기

문제 수정 기능을 통하여 문제를 자세하게 파악하고 해결하기 위한 도움을 받을 수 있습니다.

- 문제 수정하기(Ctrl + 1) 메뉴를 사용하여 **검토 결과 도우미**를 열 수 있습니다.



- 제시된 해결 방법을 사용하여 문제를 빠르고 쉽게 해결 할 수 있습니다. (해결 방법이 제시되지 않는 문제의 경우에는 직접 모델을 수정해야 합니다.)



■ 추가적인 기능

- **모델검사 설정** 기능을 이용하여 원하는 모델검사만 사용하여 모델을 검토 할 수 있습니다.

7. XScript

■ XScript 란?

XScript는 eXERD 파일에 대해 사용자가 직접 다이어그램, 테이블, 컬럼 등을 찾거나 수정할 수 있도록 도와주는 언어입니다.

● 기본 전략

XScript는 대부분의 동작이 다음과 같은 두 과정을 갖는다고 가정하고 설계 되었습니다:

① 주어 선택: 조작할 모델들을 선택

② 동사 기술: 선택된 모델들에 대해 동작을 기술

● 주어의 선택

주어를 선택할 때는 select() 문을 이용하며, 인자로 클로져가 전달됩니다. 클로져는 하나의 인자를 갖고 true 또는 false 를 리턴하는 function 형태이어야 합니다. 이 클로져의 구조는 다음과 같습니다:

```
function(it){
        return true 또는 false;
}
```

eXERD 문서를 구성하는 모든 모델은 각각 클로져의 인자인, it 으로 전달 되어, 클로져가 true 를 리턴하면 그 모델이 선택됩니다. 예를 들어 모든 컬럼을 선택하는 방법은 다음과 같습니다:

```
select(function(it){
        return it.get("type") == "column";
})
```

위의 코드는 type 속성이 "column" 인 모든 모델을 선택하게 됩니다. 이렇게 모델 선택함수 select(조건함수())를 이용해 선택된 모델들을 **모델 컨텍스트** 라고 합니다.

클로져의 인자로 it 이 사용되는 것은 권장사항일 뿐 강제사항은 아닙니다. 원하는 다른 변수명을 사용하셔도 무방합니다.

● **동사의 기술**

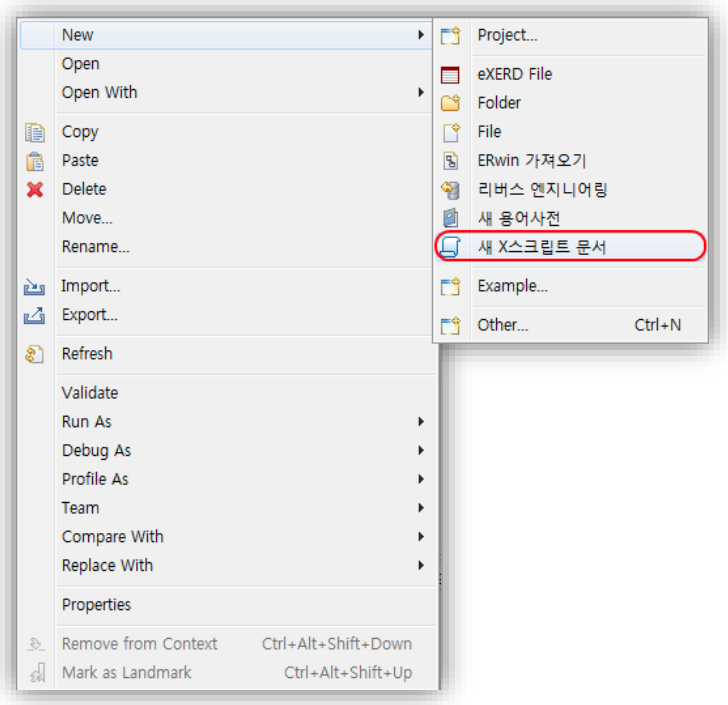
작업을 수행할 모델들을 선택했다면, 이제 동작을 기술해 봅시다. 모든 컬럼의 이름을 화면에 출력하는 간단한 예제로 시작하겠습니다:

```
select(function(it){
        return it.get("type") == "column";
}).each(function(it){
        log(it.get("logical-name"));
});
```

each(클로져) 함수는 모델 컨텍스트에 속한 모든 요소들이 주어진 클로져 동작을 차례로 수행하게 합니다. (비지터 패턴을 연상하면 이해가 쉽습니다.)

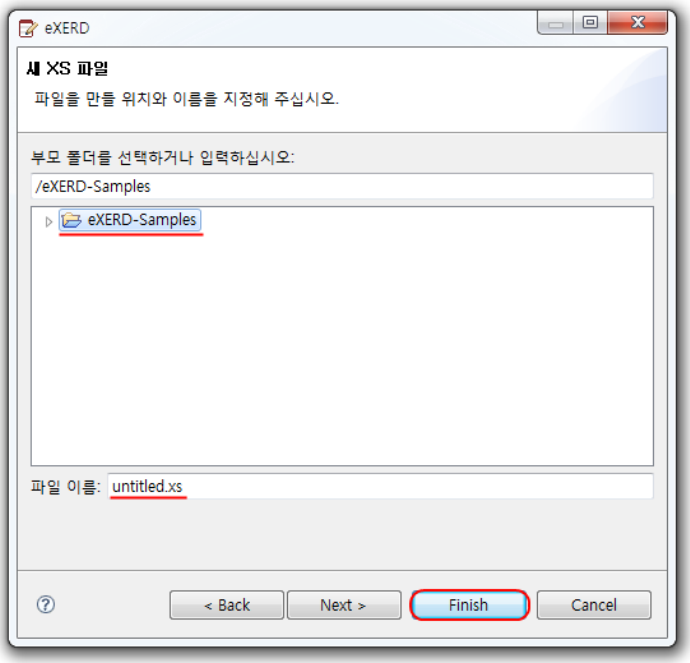
■ 빈 X스크립트 파일 만들기

1) 파일을 생성할 장소를 선택 후 마우스 오른쪽을 눌러 **[New->X스크립트]** 를 선택합니다.



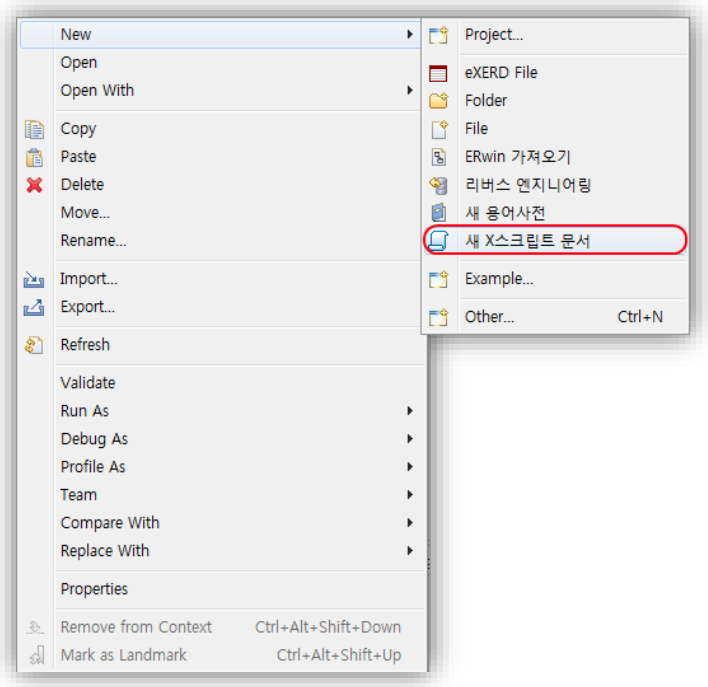
위 메뉴에서 안보일 경우 [Window->Reset Perspective...] 눌러 주세요.

2) 파일을 작성할 위치와 이름을 작성 후 **Finish** 를 눌러주세요.



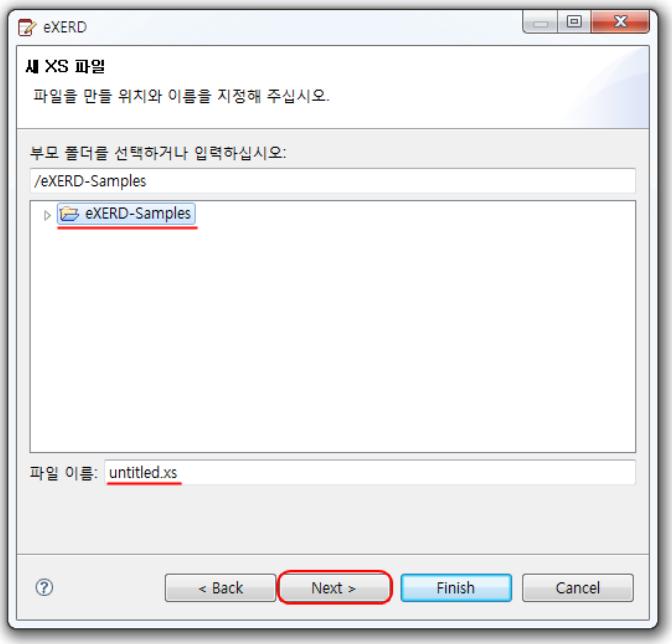
■ 템플릿 X스크립트 파일 만들기

1) 파일을 생성할 장소를 선택 후 마우스 오른쪽을 눌러 **[New->X스크립트]** 를 선택합니다.

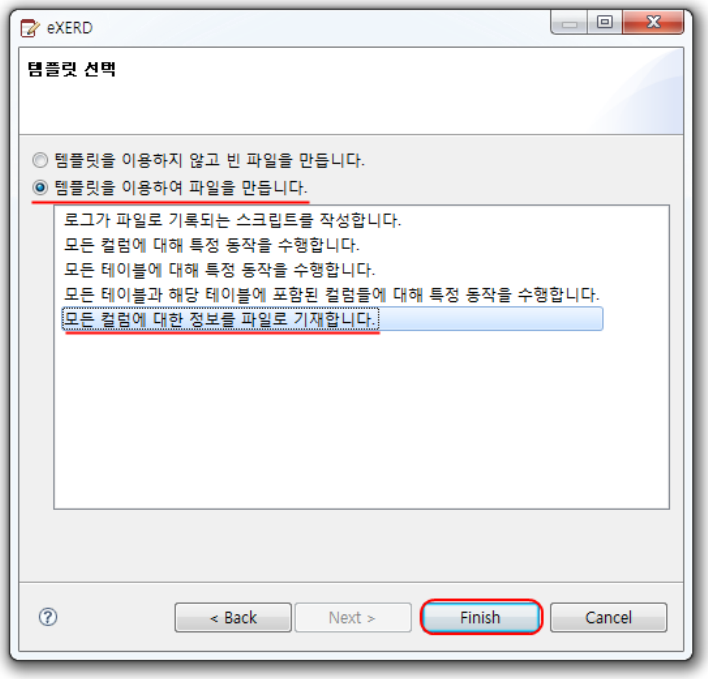


위 메뉴에서 안보일 경우 [Window->Reset Perspective...] 눌러 주세요.

2) 파일을 작성할 위치와 이름을 작성 후 **Next** 를 눌러주세요.



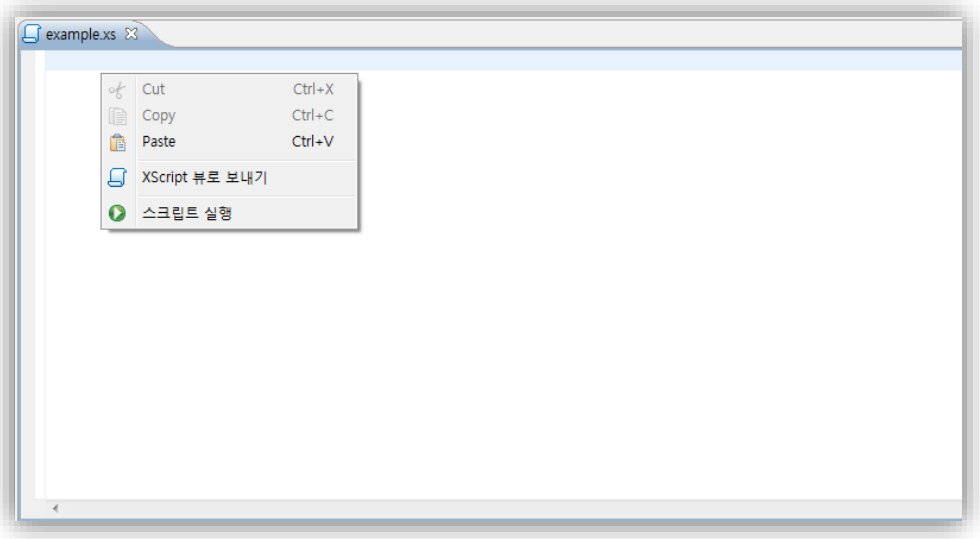
3) 원하는 템플릿을 선택하고 **Finish**를 눌러 완료 합니다.



선택된 템플릿은 스크립트에 작성됩니다.

■ XScript 편집기

.xs 파일를 가진 스크립트 파일의 편집을 도와주는 편집기입니다.



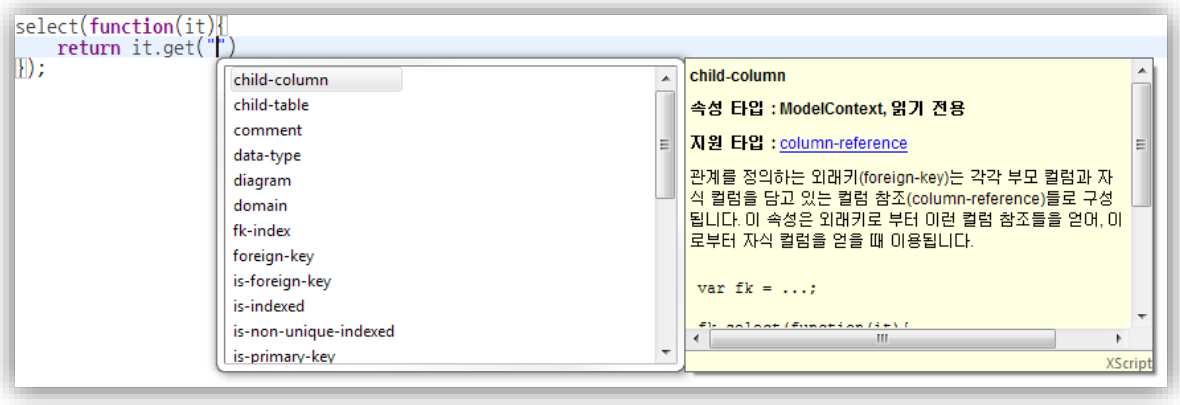
- **XScript 뷰로 보내기**는 현재 편집기에 있는 내용을 뷰로 복사 해주는 기능입니다.
- **스크립트 실행**은 현재 편집기에 있는 내용을 **.exerd 파일**이나 현재 열려있는 **다이어그램** 또는 **최근 다이어그램**에 적용하여 실행하여 줍니다.

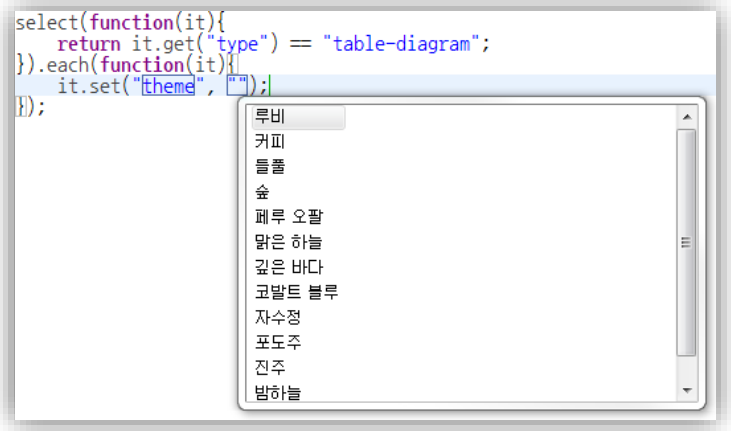
스크립트 실행 *단축키는* ***Ctrl+Enter*** *입니다.*

■ 강력한 타입 추론 엔진

XScript 에디터는 강력한 타입 추론 엔진과, 이에 기반한 의도 지원 컨텐트 어시스트 기능을 갖고 있습니다.

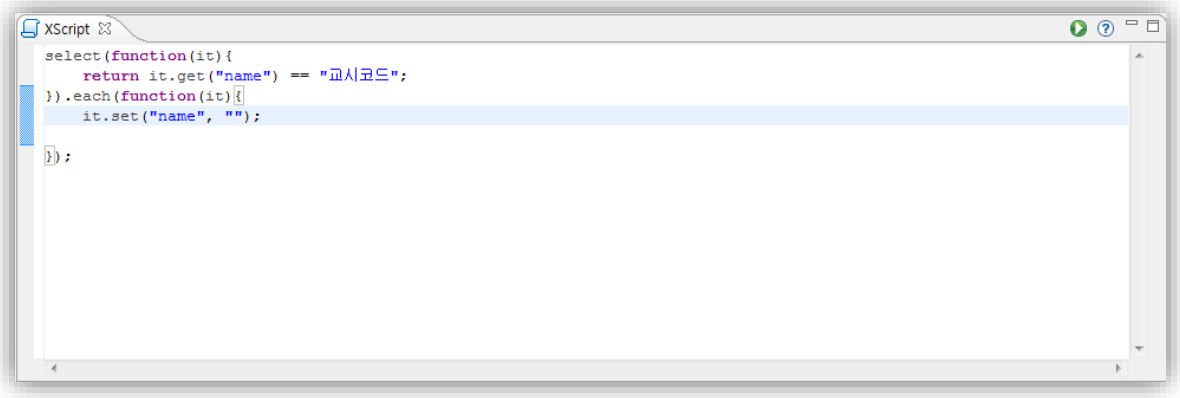
편집 도중 언제나**CTRL + SPACE**를 눌러 강력한 컨텐트 지원을 받으십시오.





■ XScript 뷰

다이어그램을 직접 보면서 스크립트 작업을 할수 있게 도와주는 뷰입니다.



▪ **스크립트 실행**은 현재 편집기에 있는 내용을 현재 열려있는 다이어그램에 적용하여 실행합니다.

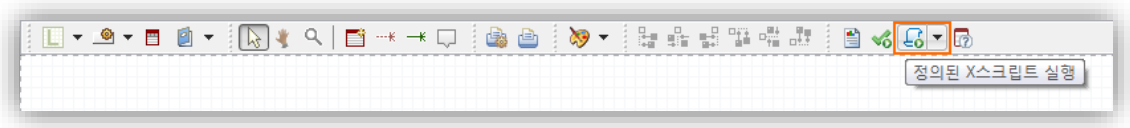
X스크립트 실행 *단축키는* ***Ctrl+Enter*** *입니다.*

■ 정의된 스크립트 실행

미리 작성된 스크립트를 현재 다이어그램에서 사용할 수 있는 기능입니다.

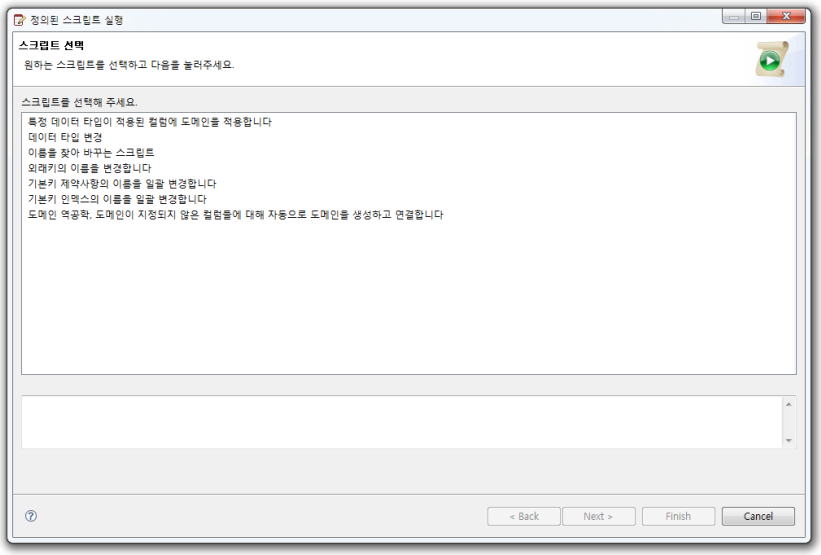
1) **실행하기**

다이어그램의 툴바에서 아이콘을 선택합니다.



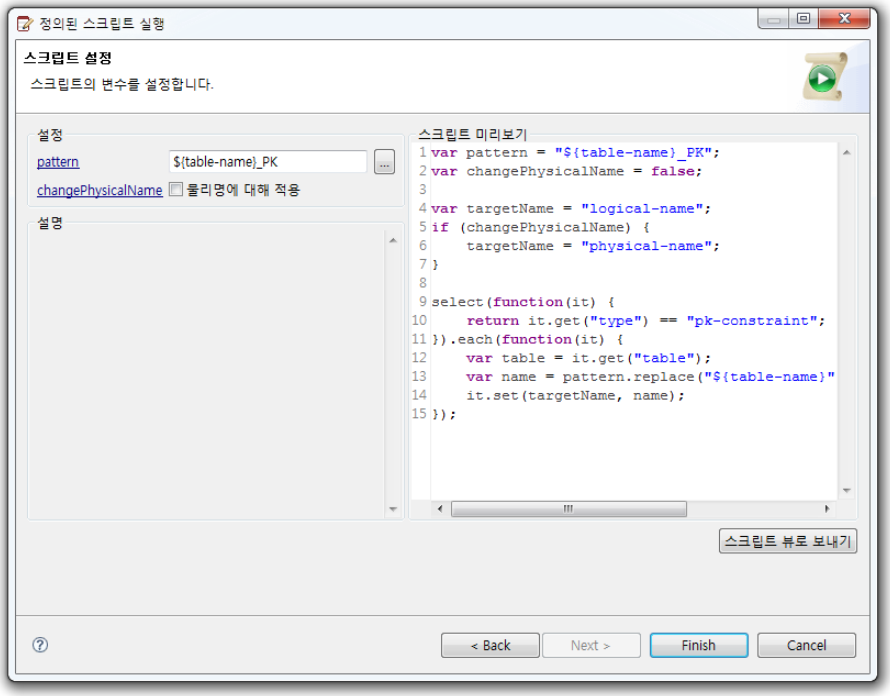
2) **스크립트 선택**

필요한 스크립트를 선택합니다.



3) **스크립트 설정**

① 원하는 값을 입력 후 **Finish** 버튼을 누르면 스크립트가 실행 됩니다.



② **스크립트 뷰로 보내기** 버튼은 미리보기의 코드를 **스크립트 뷰**로 보냅니다. 뷰가 열려 있지 않을경우 뷰를 열고 보냅니다.

note! *필드의 이름이나 입력상자를 선택하면 설명이 나옵니다.*

■ 주요 패턴

● 모델 컨텍스트 얻기

하나의 eXERD 문서는 수 많은 eXERD 문서를 구성하는 모델들로 이루어져 있습니다. 스크립트로 원하는 작업을 수행하기 위해서는 먼저 원하는 모델을 정확히 선택하여 모델 컨텍스트를 얻는 방법을 배워야 합니다. 모델 컨텍스트는 원하는 모델들을 편집하거나 정보를 열람할 수 있는 API들을 제공합니다.

XScript는 select() 함수를 제공하며, 이를 이용하여 모델을 선택합니다. select 함수는 인자로 true 또는 false를 리턴하는 평가 함수를 받습니다. 이 평가 함수가 주어진 모델 컨텍스트를 검사하고 true를 리턴하면 해당 모델을 선택하게 됩니다.

이 함수는 선택된 모델들을 다룰 수 있는 모델 컨텍스트 객체를 반환 합니다.

● 모델 타입으로 모델 선택

가장 보편적인 모델 선택 방법 중 하나는, 모델의 유형(타입)을 기준으로 선택하는 것입니다.

다음 예제에서는 모델의 타입이 table인 경우 선택하도록 합니다:

```
select(function(it){
        return it.get("type") == "table";
});
```

평가함수를 따로 정의해 두고 사용해도 됩니다:

```
var selectTable = function(it){
        return it.get("type") == "table";
};

// 또는

function selectTable(it){
        return it.get("type") == "table";
}
select(selectTable);
```

■ 스텝 인 검색

주어 없이 전역 select() 함수를 이용할 경우 자동으로 루트 컨텍스트의 select() 함수가 호출 되며, 문서에 포함된 모든 모델을 대상으로 선별 과정이 진행됩니다.

이와는 달리 이미 선택된 모델을 바탕으로 추가적인 선택 작업을 수행 할 수 도 있습니다. 다음은 특정 이름을 가진 테이블을 선택한 뒤, 그 테이블 내에서 컬럼들을 선택하는 것을 보여줍니다.

```
select(function(it){
        return it.get("type") == "table" && it.get("logical-name") == "테스트 테이블";
}).select(function(it){
        return it.get("type") == "column";
})
```

다음과 같이, 얻은 모델 컨텍스트를 변수로 지정해 두고 사용해도 됩니다:

```
var theTable = select(function(it){
        return it.get("type") == "table" && it.get("logical-name") == "테스트 테이블";
});

theTable.select(function(it){
        return it.get("type") == "column";
})
```

■ 속성 접근

모델 컨텍스트로부터, 컨텍스트에 담긴 모델들의 속성을 얻거나 지정할 수 있습니다.

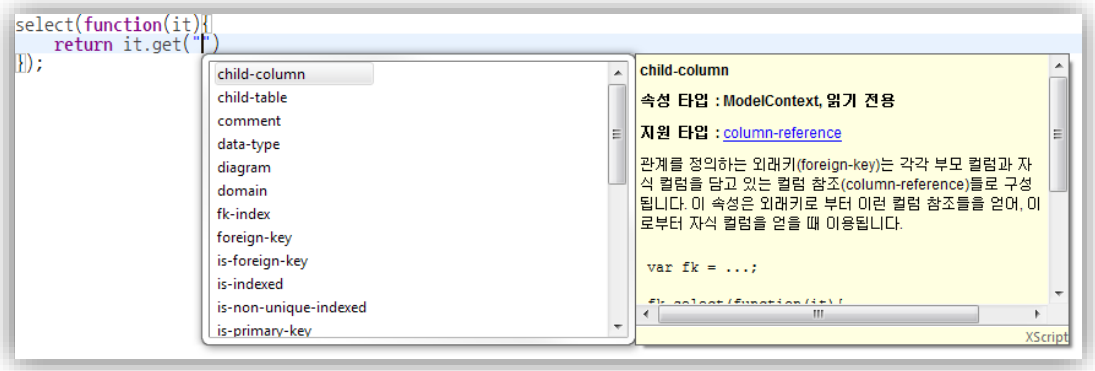
■ 속성 얻기

속성 값을 얻는 기본 적인 형태는 다음과 같습니다:

```
var property = context.get("속성이름");
```

전체 속성 목록 문서에서, 속성들을 확인할 수 있습니다.

하지만, 가장 진보적인 eXERD XScript 에디터는 단순한 타입 추론에 의한 메서드 뿐만 아니라, 속성명이나 속성 비교 값들에 대해서도, 모두 자동으로 컨텐트 어시스트를 지원하므로, 이러한 속성들을 암기하실 필요가 없습니다.

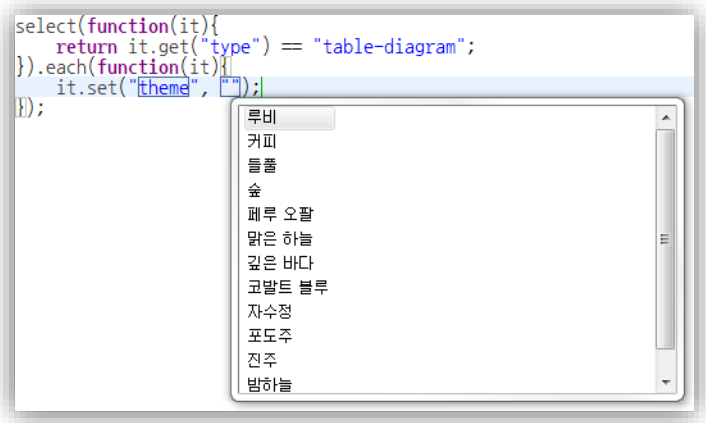


■ 속성 지정

속성 값을 지정하는 기본 적인 형태는 다음과 같습니다:

var property = context.set("속성이름", 속성값);

물론 속성 값을 지정할 때에도, 놀라운 컨텐트 어시스트를 제공 받으실 수 있습니다.



■ 진행 상태 관리

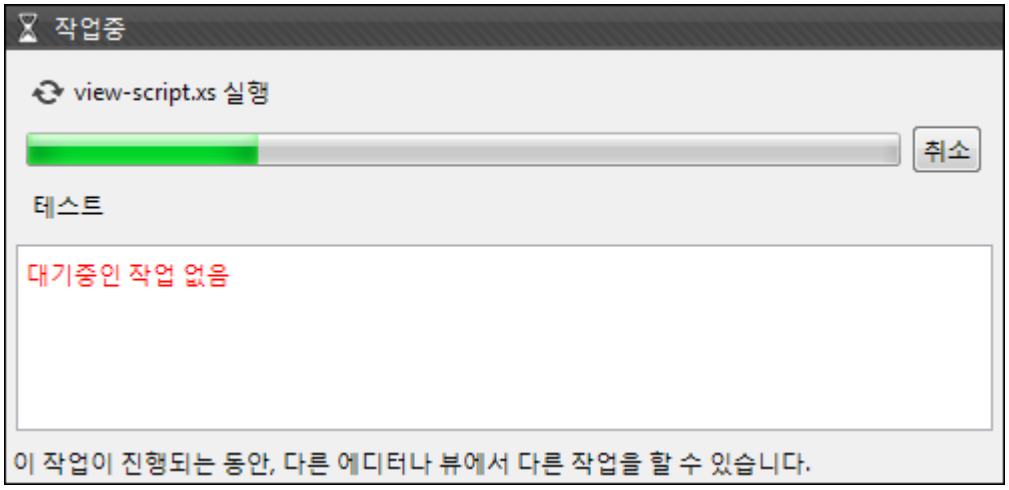
만약 어떤 스크립트의 수행시간이 다소 긴 편이라면, 진행 정도가 UI로 표시되고 사용자가 취소할 수 있는 것이 바람직 할 것입니다. 이런 경우를 위해 XScript는 monitor 전역 변수가 공급됩니다.

● 기본 진행 상태 표시

```
// 화면에 표시될 작업 이름과 작업의 양.
monitor.beginTask("테스트", 100);
for(var i = 0; i < 100; i++){
        // 의도적으로 시간을 지연시킴.
        sleep(100);
        // 작업이 1만큼 진행 되었음을 알림.
        monitor.worked(1);
}
monitor.done();
```

위 스크립트가 수행되면, 진행상태를 표시하는 다이얼로그가 나타나며, "테스트"라는 작업의 진행상황을 시각적으로 표시하게 됩니다.

● 작업 취소의 지원



스크립트 실행 중 취소 버튼을 누르면, 스크립트가 강제로 중단되고, 그 때까지 스크립트가 일으킨 모든 변경은 자동으로 취소됩니다.

잘 못 작성된 스크립트가 무한 루프에 빠진 경우에도, 안전하게 종료시킬 수 있습니다.

■ 파일 작성

● 파일의 취득

파일을 작성하려면 먼저 파일 객체를 얻어야 합니다. 파일 객체를 얻는 방법은 다음과 같습니다.

① UI API를 이용하여 파일 다이얼로그를 열어 선택하게 함

② 직접 파일 객체를 만듦

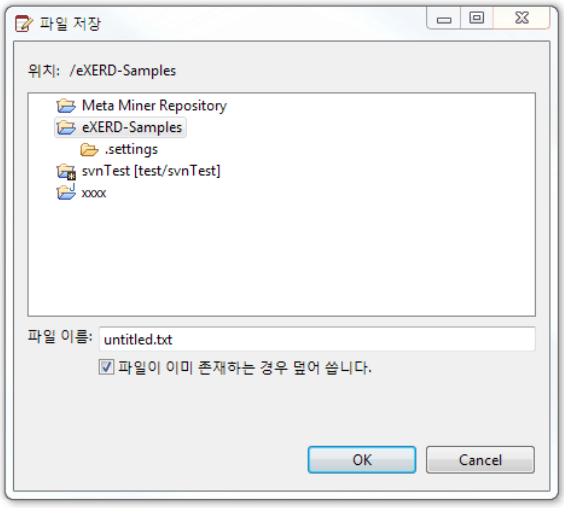
● 직접 파일 객체 생성

```
/*
 * 상대 경로: 스크립트가 실행될 대상 eXERD 파일이 담긴 폴더에 파일을 만듬
 */
var file = newFile("test.txt");
```

```
/*
 * 절대 경로: helloWorld 프로젝트 안에 test.txt 파일을 만듬.
 */
var file2 = newFile("/helloWorld/test.txt");
```

● 사용자가 파일을 선택하도록 하기

```
var file = promptSaveFile();
```



● **파일 내용 쓰기**

파일의 내용을 쓰려면, 파일 객체로 부터, 쓰기 스트림 객체를 얻어야 합니다. 쓰기 스트림 객체를 얻는 방법은 다음과 같습니다.

```
var file = ...;
var outputStream = file.getOutputStream(); // 쓰기 스트림 얻기
```

일단 스트림을 얻으면 다음과 같이 내용을 쓸 수 있습니다:

```
var outputStream = ...;
outputStream.printf("%s(%df세)₩n", "홍길동", 32);
```

파일의 내용:

홍길동(32세)

예를 들어 모든 컬럼의 이름과 데이터 타입을 출력하는 예제는 다음과 같습니다:

```
var outputStream = ...;

select(function(it){
        return it.get("type") == "column";
}).each(function(it){
        outputStream.printf("%s : %s₩n", it.get("logical-name"), it.get("data-type"));
});

outputStream.close();
```

● 기본 로그

로그 작성을 위해 log() 전역 함수가 제공됩니다.

```
log("hello world");
```

아래와 같이 문자열 연산을 포함한 연산 결과를 인자로 전달 할 수도 있습니다.

```
log(1 + 4);
log("hello " +"World!");
```

● 양식에 맞추어 출력하기

logf() 전역함수를 사용하면, 양식을 지정하여 데이터를 출력할 수 있습니다. 이는 C의 전통적인 printf() 함수와 유사합니다. printf와 달리 한글등의 문자도 정렬 기능을 이용할 수 있습니다.

```
logf("%2.2f₩n", 31.249);
logf("%-10s%-10s", "Hello", "World");
logf("#%02X%02X%02X₩n", 255, 128, 128);
```

● 출력 결과

```
31.25
Hello      World
#FF8080
```

더 자세한 양식들을 확인 하려면 콘솔 API를 참고 하십시오.

● 로그 내용을 저장하기

log() 및 logf()는 console.log()와 console.logf()의 약식 표현 입니다. console의 setLogFile() 메서드를 이용하여, 콘솔의 출력내용을 파일에 연결할 수 있습니다.

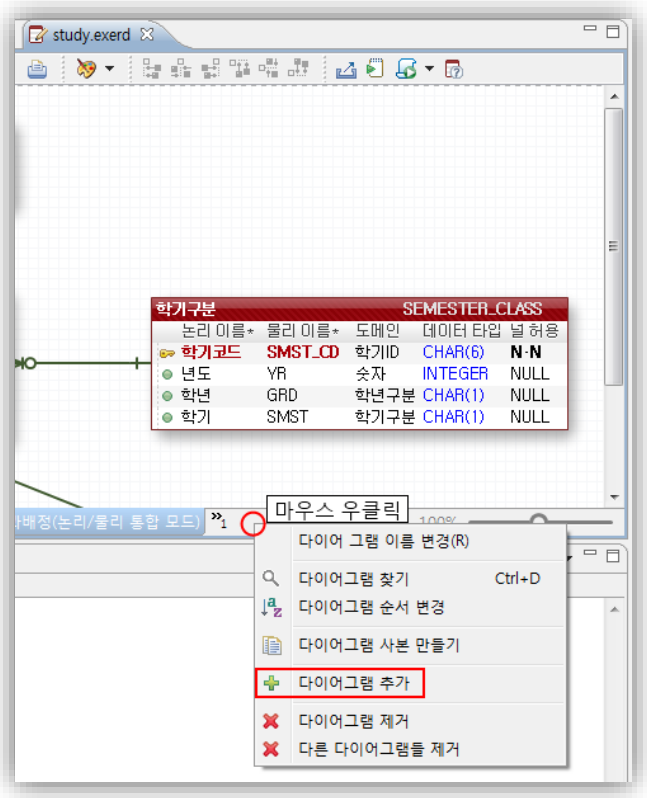
다이어그램 에디터 기능

- 테이블 만들기
- 컬럼 만들기
- 식별 관계 만들기
- 비식별 관계 만들기
- 논리/물리 모드 변환
- 편집 프리셋
- 테이블 헤더 보이기
- 용어사전 연결하기
- 용어사전 일괄편집
- 용어사전 역공학
- 스타일 지정
- 자동배치
- 모델 검색
- 테이블 검색
- 테이블 개요 보기

다이어그램 에디터 기능

1. 다이어그램 추가

다이어그램 편집기 아래 탭에 마우스 우클릭하여 메뉴에서 추가를 누릅니다.



Note! *단축키, Ctrl+3 에서 "다이어그램 추가"를 검색해서 작성이 됩니다.*

2. 테이블 만들기

1) 툴바에서 버튼을 선택하거나 단축키 "**3**"을 누릅니다.

2) 마우스 포인터 모양이 변경되어 표시됩니다.

3) 다이어그램의 원하는 위치에서 클릭합니다.

Note! *새로 생성된 테이블은 자동으로 활성 스키마가 소유하게 됩니다.*

3. 컬럼 만들기

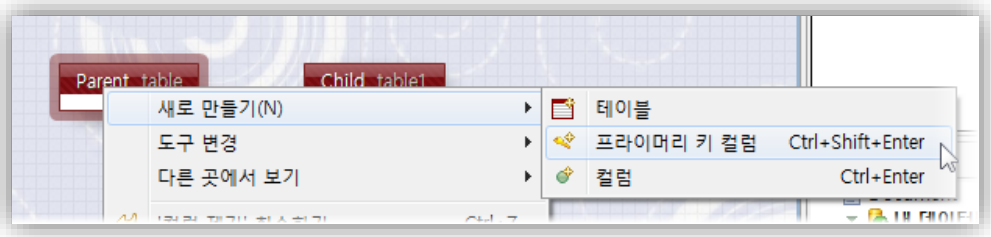
테이블이 선택된 상태에서 단축키로 컬럼을 빠르게 추가할 수 있습니다.

Tip! *이름 편집중에도 해당 단축키를 입력하면 새 컬럼이 추가됩니다.*

1) 기본키 추가 (**Ctrl + Shift + Enter**)

2) 일반 컬럼 추가 (**Ctrl + Enter**)

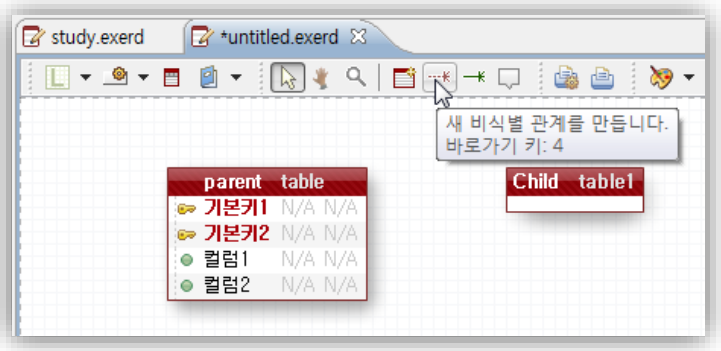
3) 팝업메뉴로 추가



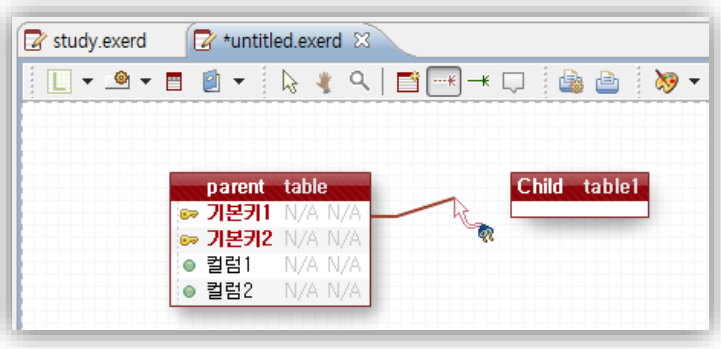
4. 관계 연결

간단한 방법으로 테이블간의 관계를 연결할 수 있습니다. 식별 관계와 비식별 관계를 연결하는 방법은 동일하므로 비식별 관계를 예를 들어 설명합니다.

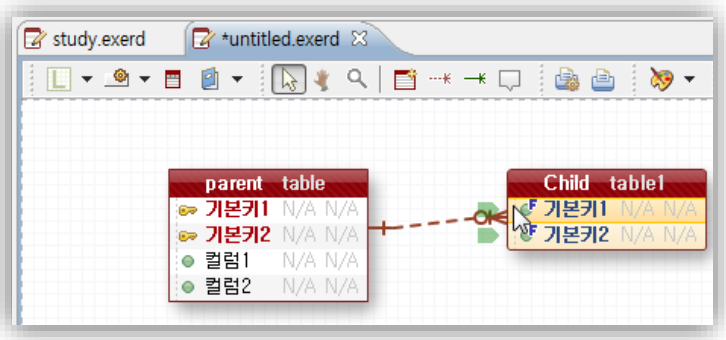
1) 툴바에서 **비식별 관계** 버튼을 선택하거나 단축키 "**4**"를 누릅니다.



2) 마우스로 부모 테이블을 선택합니다. 마우스를 이동하면 관계선이 표시됩니다.

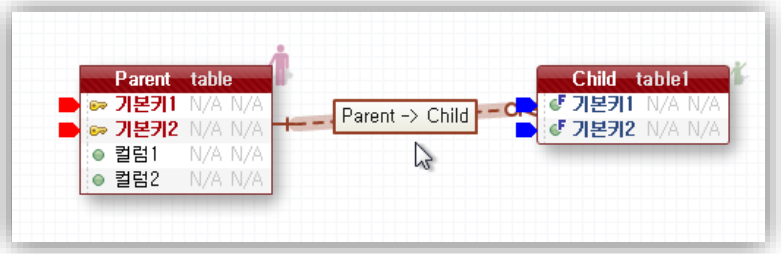


3) 마우스로 자식 테이블을 선택합니다. 자동으로 자식테이블에 참조 컬럼이 생성됩니다.



Tip! *자기 자신을 선택하면 셀프참조 관계가 만들어집니다.*

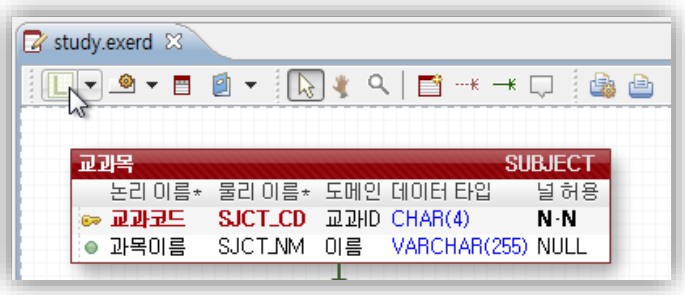
4) 관계선을 마우스로 선택하면 아이콘으로 부모 테이블과 자식 테이블이 표시되며 해당 관계로 연결된 컬럼이 강조 됩니다.



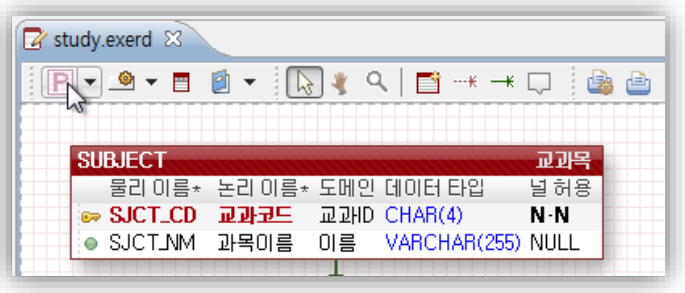
5. 논리/물리 모드 변환

논리 모드와 물리 모드의 전환은 툴바에서 버튼을 선택하거나 단축키 "**F3**"으로 할 수 있습니다.

1) 논리모드에서는 푸른색 배경이 나타납니다.



2) 물리모드는 붉은색 배경이 나타납니다.

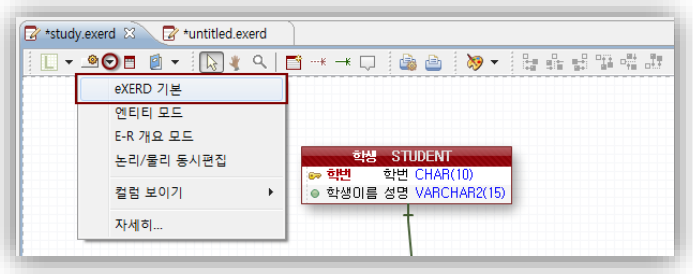


6. 편집 프리셋

미리 설정된 보기 구성을 사용할 수 있습니다.

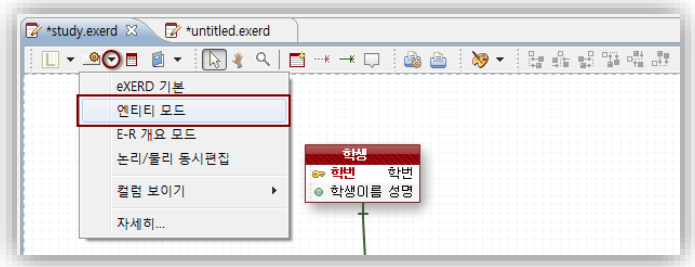
- **[eXERD 기본]** 툴바에서 표기부분의 ▼ 버튼을 눌러서 **eXERD 기본**을 선택합니다.

eXERD 기본 프리셋은 .erd 파일을 생성했을 때 적용되는 기본 구성입니다. 컬럼이름, 도메인, 데이터타입이 표시되도록 설정되어 있습니다.



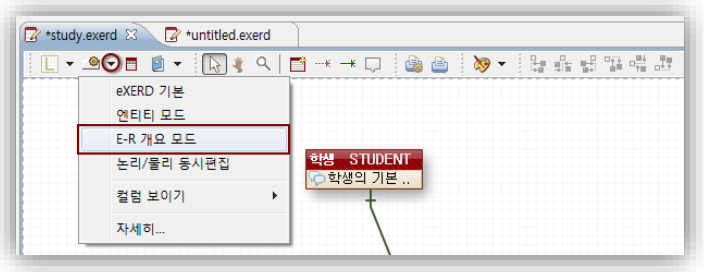
- **[엔티티 모드]** 툴바에서 표기부분의 ▼ 버튼을 눌러서 **엔티티 모드**를 선택합니다.

컬럼이름과 도메인만 표시되어 초기 논리모델링 작업에 유용한 프리셋입니다.



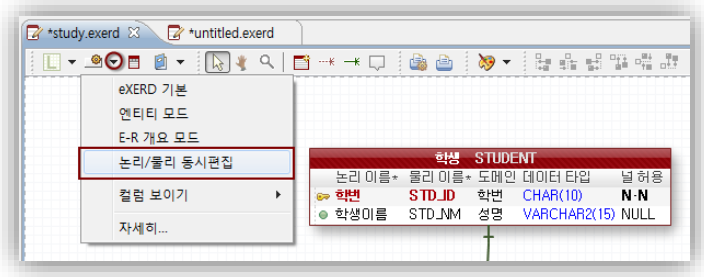
• **[E-R 개요 모드]** 툴바에서 표기부분의 ▼ 버튼을 눌러서 **E-R 개요 모드**를 선택합니다.

테이블명과 주석만 표시되어 전체적인 E-R 을 한눈에 파악하기 용이 합니다.



• **[논리/물리 동시편집]** 툴바에서 표기부분의 ▼ 버튼을 눌러서 **논리/물리 동시편집**를 선택합니다.

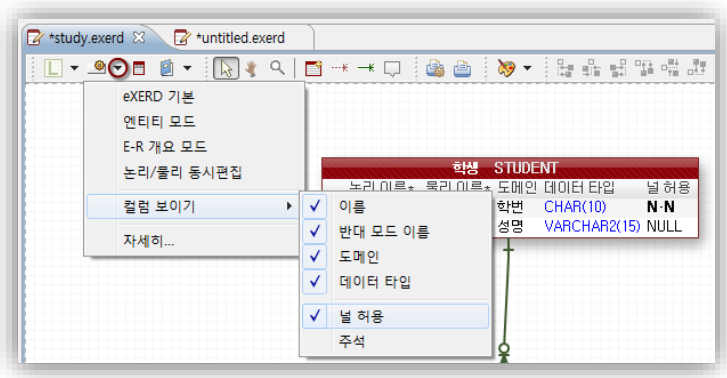
논리명과 물리명이 동시에 편집할 수 있는 구성입니다.



■ 메뉴에서 지정

컬럼의 다양한 정보들에 대한 표시여부를 메뉴에서 간단히 설정할 수 있습니다.

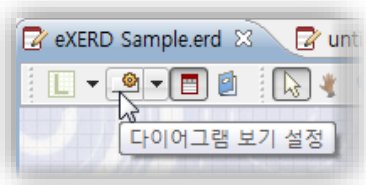
▪ 툴바에서 표기부분의 ▼ 버튼을 눌러서 컬럼 보이기 하위 메뉴 항목을 선택합니다.



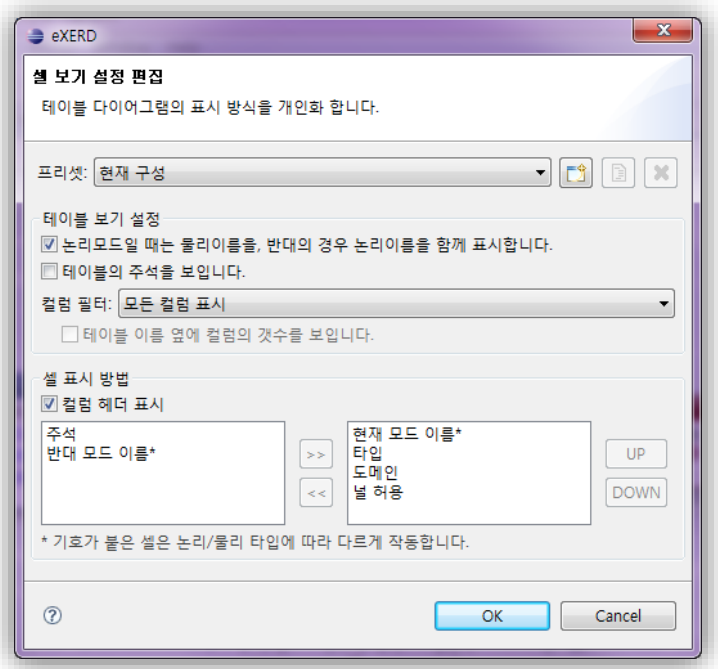
■ 상세 설정 및 프리셋 만들기

추가적인 상세한 항목들을 설정하거나 설정한 보기 구성을 프리셋으로 저장하여 다른 다이어그램에도 적용할 수 있습니다.

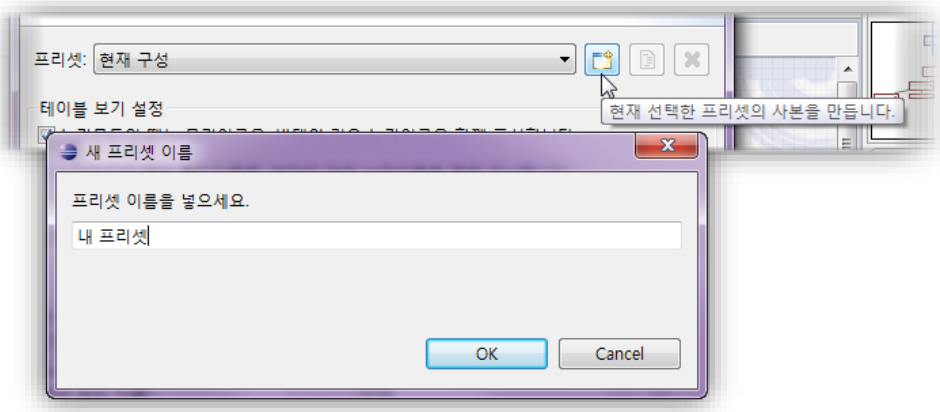
▪ 툴바에서 **다이어그램 편집기 설정** 버튼을 선택하면 설정 대화상자가 나타납니다.



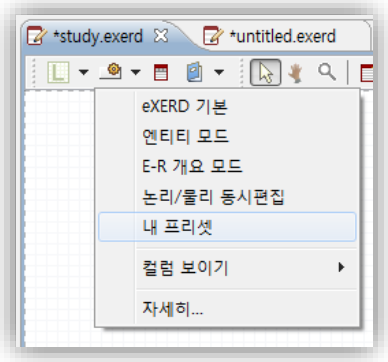
▪ 테이블 이름과 주석 및 셀 표시 방법등을 설정할 수 있습니다.



▪ 설정한 보기 구성을 새로운 이름으로 저장할 수 있습니다.



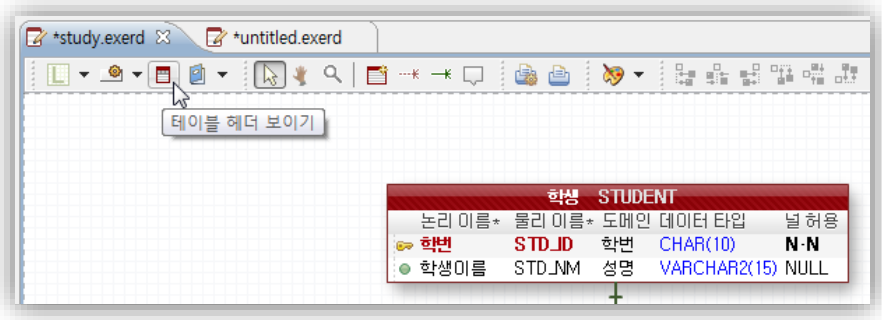
▪ 새로 만들어진 프리셋은 메뉴에 표시되어 다른 다이어그램에도 간단히 적용할 수 있습니다.



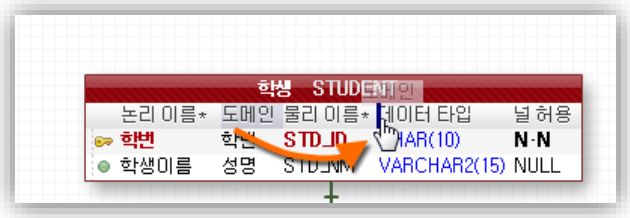
7. 테이블 헤더 보이기

컬럼에 표시된 셀들의 헤더에 대한 표시여부를 설정할 수 있습니다.

1) 툴바에서 **테이블 헤더 보이기** 버튼을 선택합니다. 다시 누르면 설정이 토글됩니다.



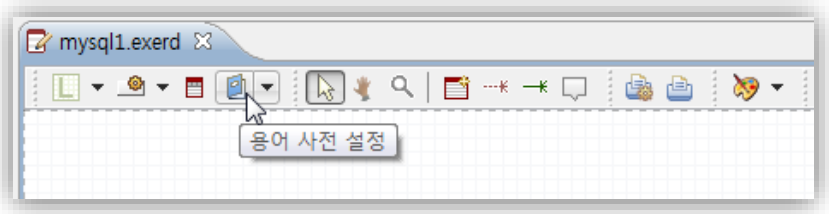
2) 헤더에서 항목을 선택하고 좌우로 드레그하면 위치를 이동할 수 있습니다.



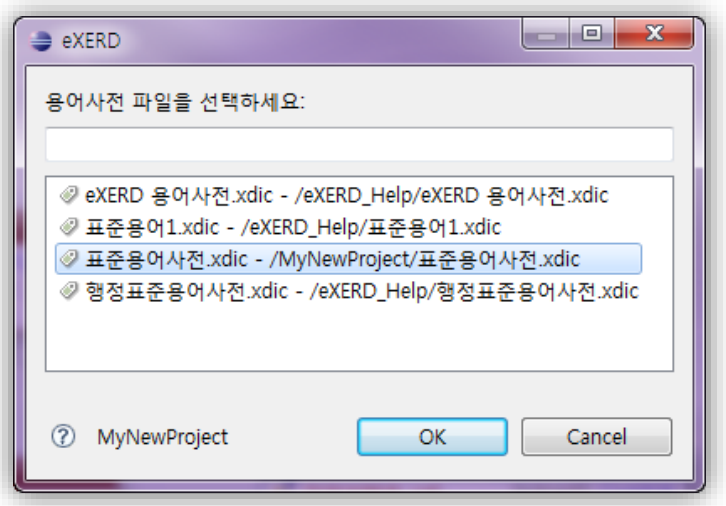
8. 용어사전 연결하기

준비된 용어사전을 eXERD 다이어그램 에디터에서 연결하여 E-R 다이어그램에 표준 용어 정책을 적용합니다.

1) 툴바에서 **용어 사전 설정** 버튼을 실행합니다.



2) 현재 E-R 다이어그램에 사용할 용어사전을 선택합니다.



9. 용어사전 일괄편집

■ 용어 정책 일괄 적용

용어 정책 일괄 적용 기능을 이용하면, 용어사전을 기반으로, 물리명을 일괄 적용할 수 있습니다. 이 작업은 크게 네 단계로 나누어 집니다:

①. 용어 정책 적용 계획 수립

②. 자동 적용 가능한 물리명 계획 검토

③. 적용 가능한 물리명이 복수인 경우에 대한 계획 검토 및 수정

④. 계획 실행

1) 계획 작성

용어 사전 일괄 적용을 수행하기 위한 첫 단계는 계획을 세우는 것입니다. 적용 계획은 사용자가 정의한 옵션을 바탕으로 자동으로 수립되며, 적용 계획이 작성된 후, 다음 단계로 넘어갈 수 있습니다. 컬럼 및 용어사전의 규모에 따라 작업 시간이 길어 질 수 있으며, 다중 CPU가 탑재된 PC의 경우 더 빠른 속도로 계획이 수립됩니다.

적용 계획 생성 옵션:

- **용어사전과 일치하는 물리명을 제외하고 용어사전 적용 계획을 만듭니다.**

  이 옵션을 선택하면, 용어 사전에 일치하는 물리명들을 제외하고 적용 계획을 생성합니다. 용어사전 정책에 따라 여러 물리명을 가질 수 있는 경우, 물리명 중 하나와 일치하는 경우, 정책에 합치하는 것으로 보고, 계획 생성에서 무시됩니다.

- **대체 가능한 물리명이 존재하는 경우를 포함하여 용어사전 적용 계획을 만듭니다.**

  이 옵션은 적용 가능한 물리명이 하나 밖에 없고, 이에 일치하는 물리명을 가진 논리명들을 제외하고 용어사전 적용 계획을 만듭니다. 용어사전과 일치하는 물리명을 소유한 물리명이라고 할 지라도, 대체 가능한 물리명이 존재하는 경우, 적용 계획에 포함됩니다.

- **모든 물리명에 대해 용어사전 적용 계획을 만듭니다.**

  이 옵션은 모든 논리명에 대해, 이미 정책에 부합하는 물리명을 가졌는지 여부와 상관 없이, 정책 적용 계획을 작성합니다.

  필요한 옵션을 선택 한 뒤, [계획 작성] 버튼을 누르면, 계획이 생성되고, 생성이 완료되면 다음 단계로 넘어갑니다.

2) 물리명 자동 적용 계획 검토

이 단계에서는 적용 가능한 물리명이 유일하여, 자동 적용이 가능한 논리명들에 대해 검토를 수행 합니다. 동일한 논리명은 그루핑 되어 나타나며, 각 항목의 체크 박스를 이용하여, 용어사전의 물리명을 적용할 것인지 여부를 결정 할 수 있습니다.

3) **애매 모호성 검토**

이 단계에서는 적용 가능한 물리명이 복수 인 경우, 가능한 물리명 중 하나를 선택할 수 있도록 도와 줍니다.

노란색 영역은 직접 편집이 가능하며, 클릭하면 대체 가능한 다른 물리명들이 나열 됩니다. 다른 물리명을 선택하면, 용어사전의 설명을 함께 볼 수 있습니다. 이 값이 편집되면 자동으로 체크 박스가 채워져, 적용 계획에 포함됩니다.

초기 값으로 적용가능한 첫 번째 물리명이 씌여져 있지만, 체크 박스가 비워져 있음을 유의해 주십시오.

10. 용어사전 역공학

용어사전 역공학 기능을 이용하면, 용어사전에 등록되지 않은 논리명을 사용한 후 용어사전에 등록하고 싶은 경우 일괄적으로 등록하게 도와줍니다.

이 기능은 다이어그램과 용어사전이 연결된 상태에서만 활성화됩니다.

1) 다이어그램의 툴바에서 **용어사전 역공학** 버튼을 실행합니다.
2) 작업 창이 활성화 되면 용어사전에 없는 컬럼들을 나열합니다.
3) **편집 영역**에 필요한 작업을 마치고 Finish 를 실행합니다.

Note! *하나의 논리명에 여러개의 물리명이 있으면 대표 물리명을 선택 편집이 가능합니다. 나머지는 물리 동의어로 등록됩니다.*

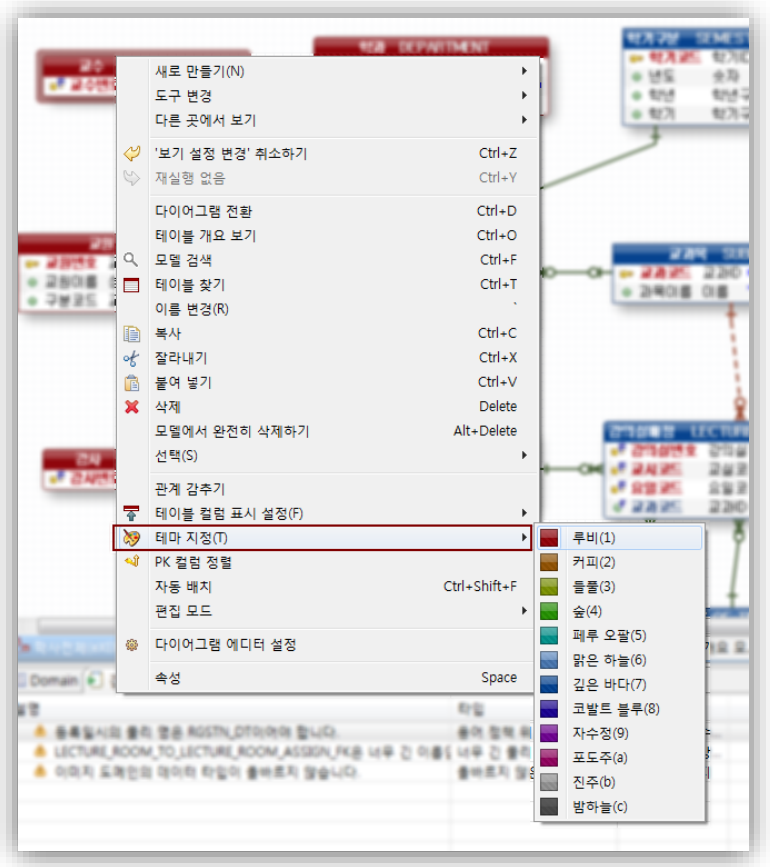
11. 스타일 지정

E-R 다이어그램에서 특정 테이블 및 컬럼들을 강조하는 것이 필요하기도 합니다. 이런 경우 eXERD에서 제공하는 테마와 색상 테그를 이용하면 자연스럽게 강조 효과를 낼 수 있습니다.

■ 테마 지정

특정 테이블에 다른 테마를 지정하여 강조 효과를 낼 수 있습니다. 테마를 지정하면 테이블 이름 영역과 외곽선 색상 및 선택 색상이 변경됩니다.

▪ 테이블을 선택하고 팝업메뉴에서 **테마 지정**의 하위메뉴로 테마를 지정할 수 있습니다.



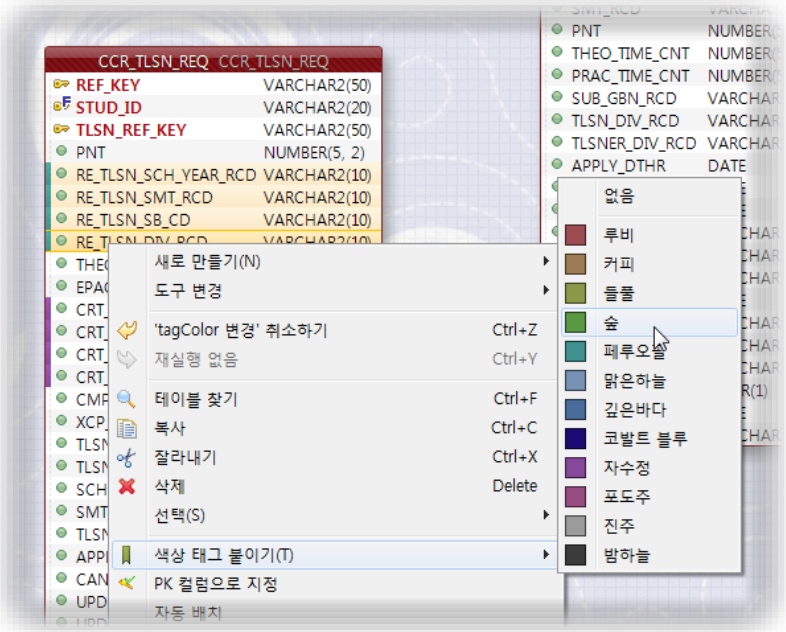
Note! *설정한 테마는 현재 다이어그램의 선택한 테이블들에 반영되며 이후 생성되는 테이블들에도 적용됩니다.*

■ 컬럼에 색상 태그 붙이기

테이블에서 특정 컬럼들이 서로 밀접한 관계가 있는 경우에 보통 비슷한 이름으로 연속되도록 작성합니다. eXERD에서는 이와 더불어 컬럼에 **색상 태그**를 설정하여 더 확실히 구분되도록 할 수 있습니다.

색상 태그는 컬럼의 앞부분에만 간략하게 표시하므로 다이어그램의 전체 맥락을 어지럽게 만들지 않으면서 센스있게 강조합니다.

▪ 컬럼들을 선택하고 팝업메뉴에서 **색상 태그 붙이기**의 하위메뉴에서 색상을 선택합니다.



Note! *컬럼에 설정한 색상 태그는 해당 테이블을 표시하는 모든 다이어그램에서도 동일하게 표시됩니다.*

■ 추가적인 기능

▪ **글꼴 바꾸기** 기능을 이용하여 의미있는 문장의 색깔과 글자체를 변경할 수 있습니다.

12. 자동배치

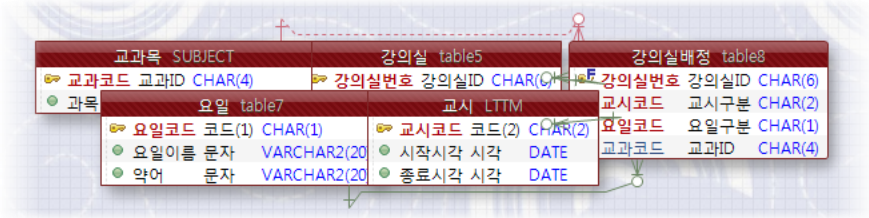
■ 자동 배치와 격자 맞춤

모델뷰에서 다이어그램으로 테이블들을 추가하거나 직접 배치한 테이블들을 자동으로 보기 좋게 배치하는 기능입니다.

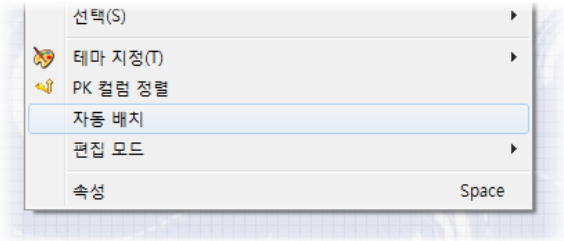
● 자동 배치

다이어그램에 배치된 테이블들의 관계를 기준으로 적절한 위치에 겹치지 않게 배치하는 기능입니다.

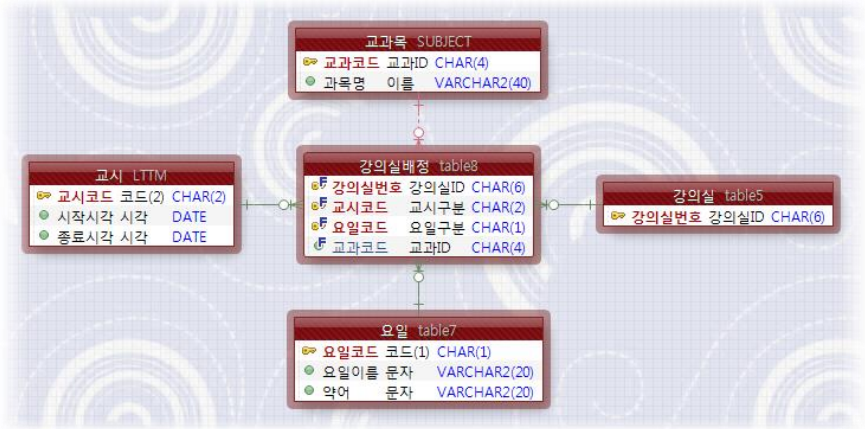
1) 다이어그램에 테이블들을 배치합니다.



2) **Ctrl + Shif + F** 를 누르거나 팝업메뉴에서 **자동 배치**를 선택합니다.



3) 관계선을 중심으로 테이블들이 자동으로 배치됩니다.



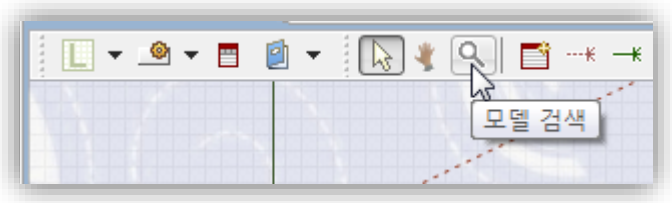
Tip! *자동배치(Ctrl + Shif + F)를 재실행 하면 매번 다른 알고리즘이 적용되어 관계에 따라 5~8가지의 다른 모습으로 배치됩니다.*

13. 모델 검색

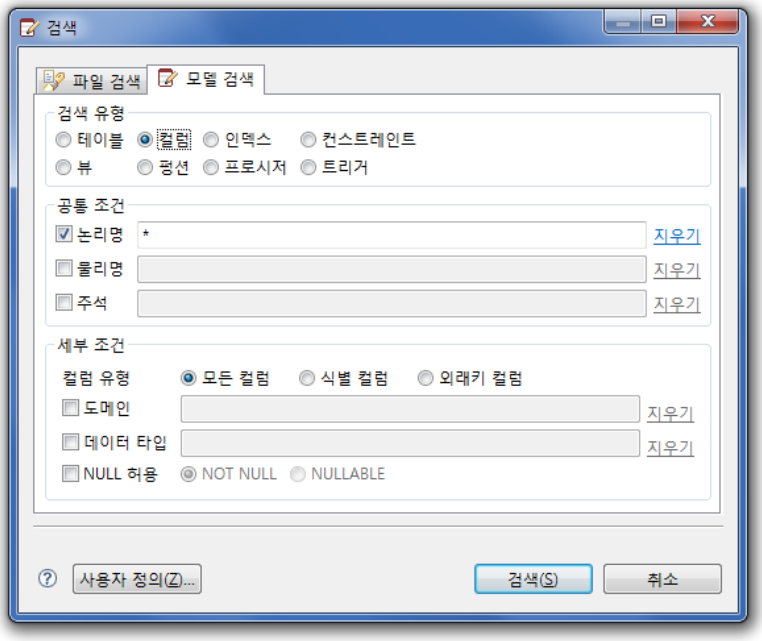
전체 모델에서 조건에 맞는 대상들을 검색하는 기능을 제공합니다.

■ 모델 검색

1) 단축키 **Ctrl + F**를 입력하거나 툴바 및 팝업메뉴에서 **모델 검색**을 실행합니다.



2) 검색어을 입력하고 찾을 대상과 조건을 설정합니다.

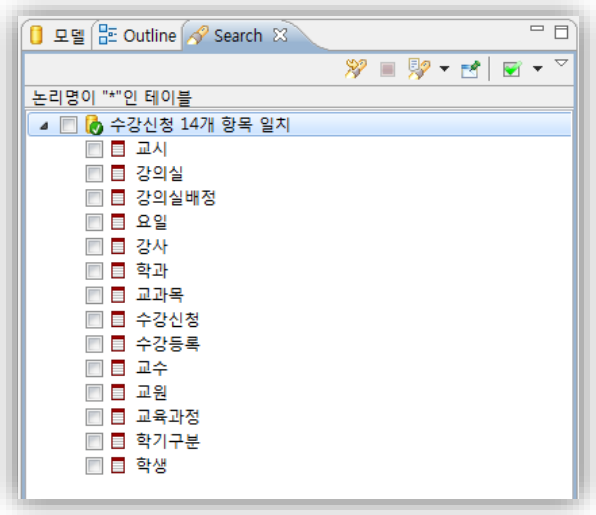


Tip! *위 화면은 컬럼 논리명이 "교과"로 시작하고 도메인이 "교과"로 지정된 컬럼을 검색하게 됩니다.*

■ 검색 결과 뷰

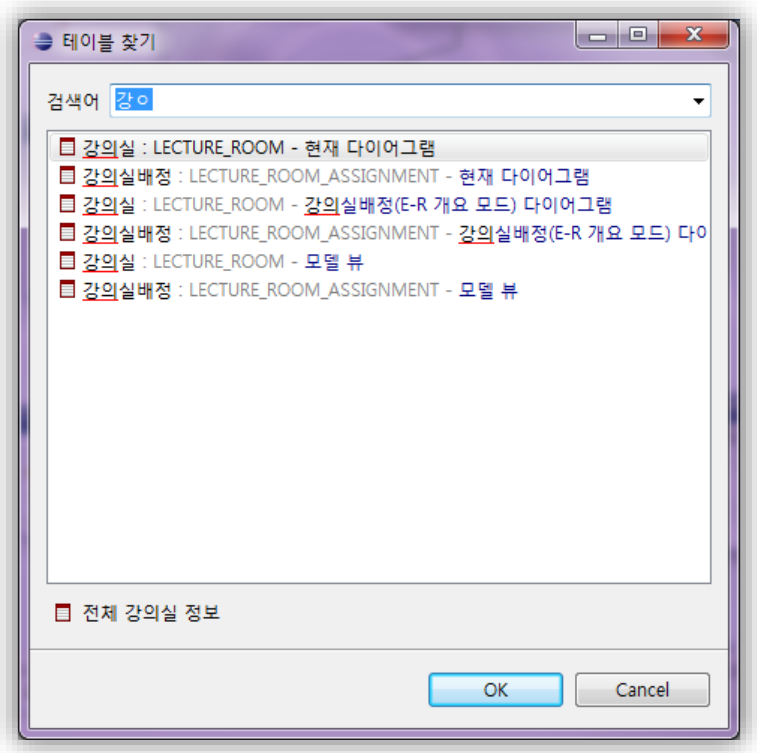
검색을 실행하면 검색된 결과가 "검색 뷰"에 표시됩니다.

1)검색된 항목을 더블클릭하면 다이어그램에서 해당 모델이 선택됩니다.



14. 테이블 검색

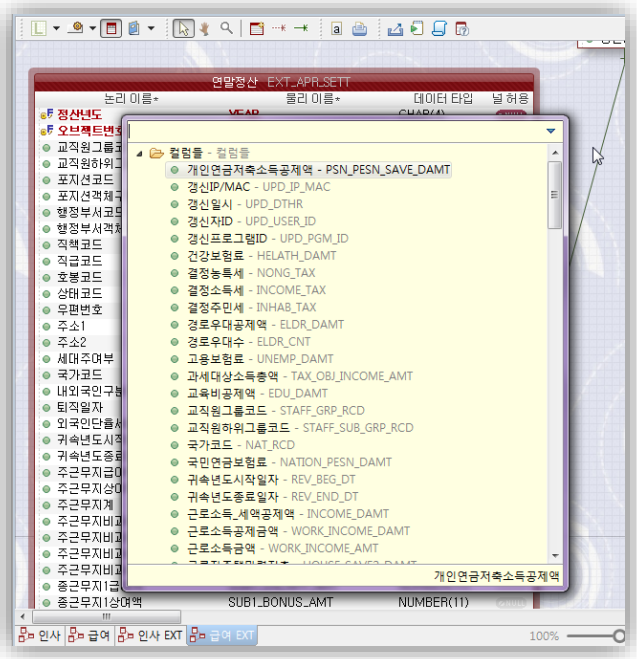
1) 단축키 **Ctrl + T** 를 입력하거나 팝업메뉴에서 **테이블 찾기**를 실행합니다.
2) 이름을 입력하면 매치되는 항목들이 나열됩니다.



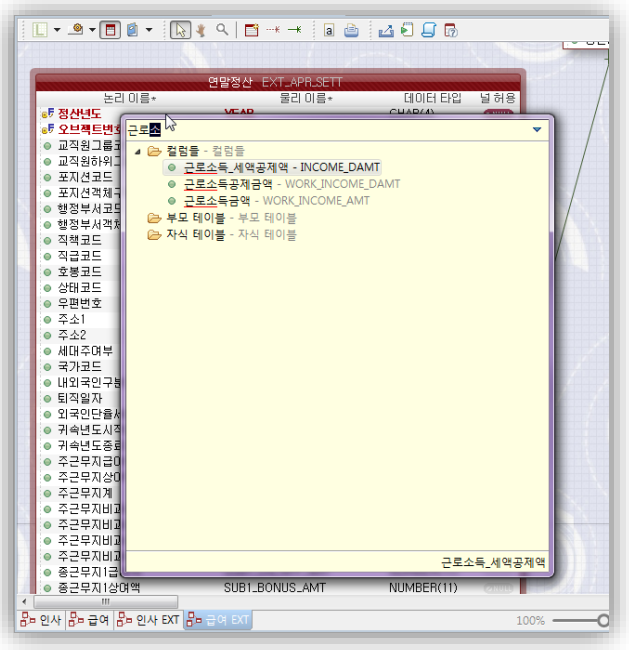
15. 테이블 개요 보기

테이블 내에서 컬럼이 많은 경우에도 특정 컬럼을 빠르게 찾을 수 있습니다.

1) 테이블을 선택하고 단축키 **Ctrl + O** 를 입력하거나 팝업메뉴에서 **테이블 개요 보기**를 선택합니다.

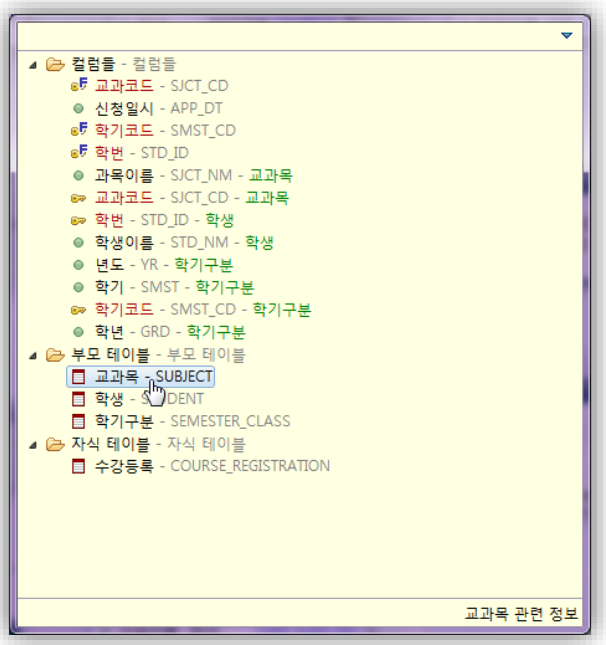


2) 논리 및 물리이름을 입력하면 매치되는 항목들만 나열됩니다.



***Tip! Ctrl + O** 를 한번 더 누르면 부모테이블의 컬럼까지 검색이 확장됩니다.*

3) 컬럼에 대한 검색 뿐만 아니라 해당 테이블의 부모테이블과 자식테이블에 대한 빠른 접근도 가능합니다.



부가기능

- 템플릿 내보내기
- 모델 내보내기
- XML 파일로 변환

부가기능

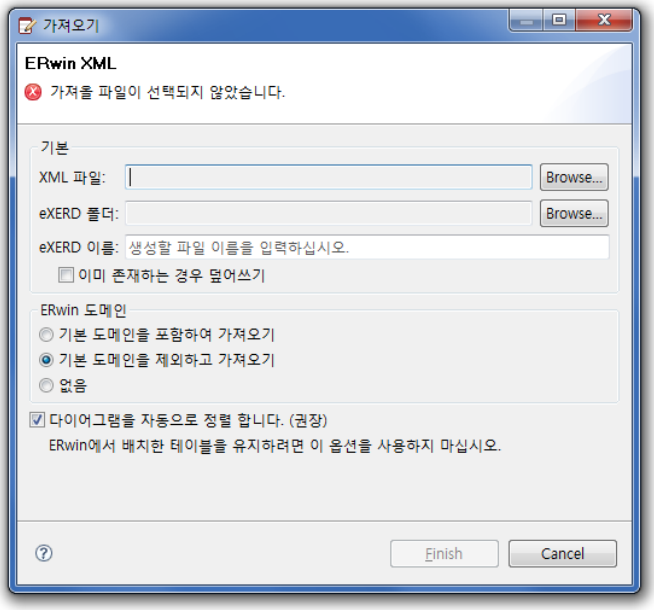
1. ERwin 파일 가져오기

■ ERwin 파일 가져오기

eXERD의 ERwin XML 파일 가져오기 기능을 이용하면, ERwin에서 작업한 내용을 eXERD로 가져올 수 있습니다.

■ ERwin 가져오기

1) eXERD로 변환된 파일을 위치시킬 폴더를 우클릭합니다.
2) 들여오기(Import)를 선택합니다.
3) eXERD 항목에서 ERwin을 선택합니다.
4) 가져올 ERwin XML 파일을 선택하고,
5) 도메인 처리 방법을 지정한 후, 완료 또는 Finish 버튼을 누릅니다.
6) 변환된 eXERD파일이 생성되고 에디터가 저절로 열립니다.



■ 도메인 처리 방법

방법	설명
기본 도메인을 포함하여 가져옴	ERwin에서 컬럼의 기본 속성으로 취급하는 도메인을 포함하여 가져옵니다.
기본 도메인을 제외하고 가져옴	사용자가 명시적으로 작성한 도메인만 가져옵니다.
ERwin의 도메인을 가져오지 않음	ERwin에서 작성한 도메인을 가져오지 않습니다. 도메인의 정보는 컬럼으로 모두 복제됩니다.

"다이어그램을 자동 정렬합니다." 항목을 체크한 경우, ERwin에서 작업한 테이블 배치를 사용하지 않고, eXERD에서 최적으로 보이도록 재 배치합니다.

■ 지원 ERwin 파일

지원 ERwin XML 파일 버전:

- 7.2, 7.3

지원 데이터베이스

- 오라클: 9, 10, 11
- MS-SQL: 2005 이상
- DB2: UDB 8, 9
- mysql: 5.X

지원 ERwin 속성

- 도메인
- 디스플레이
- 논리명/물리명/데이터 타입/널 여부/주석
- 기본키/유니크 인덱스/논 유니크 인덱스/포린 키
- 사각형 메모
- 관계

제한 사항

가져올 ERwin 파일은 반드시 논리/물리 또는 물리 모드로 작업된 파일이어야 합니다.

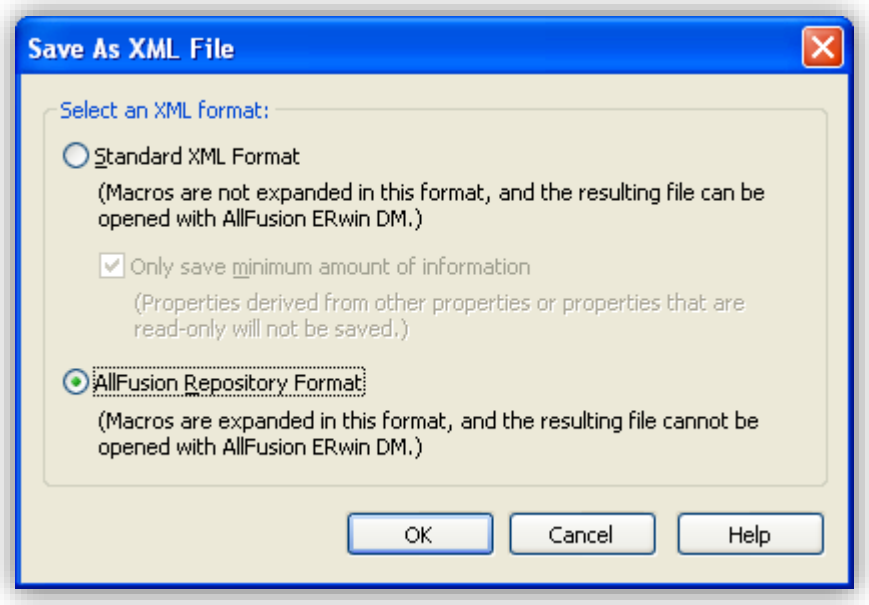
가져올 ERwin 파일은 반드시 전체 정보를 포함하여 XML로 내보내진 상태여야 합니다.

■ ERwin에서 내보내기 방법

1) File / Save As 를 차례로 선택합니다.

2) Save as type 에서 XML을 선택합니다.

3) XML 저장 옵션에서 AllFusion Repository Format을 선택합니다.



2. 템플릿(보고서)

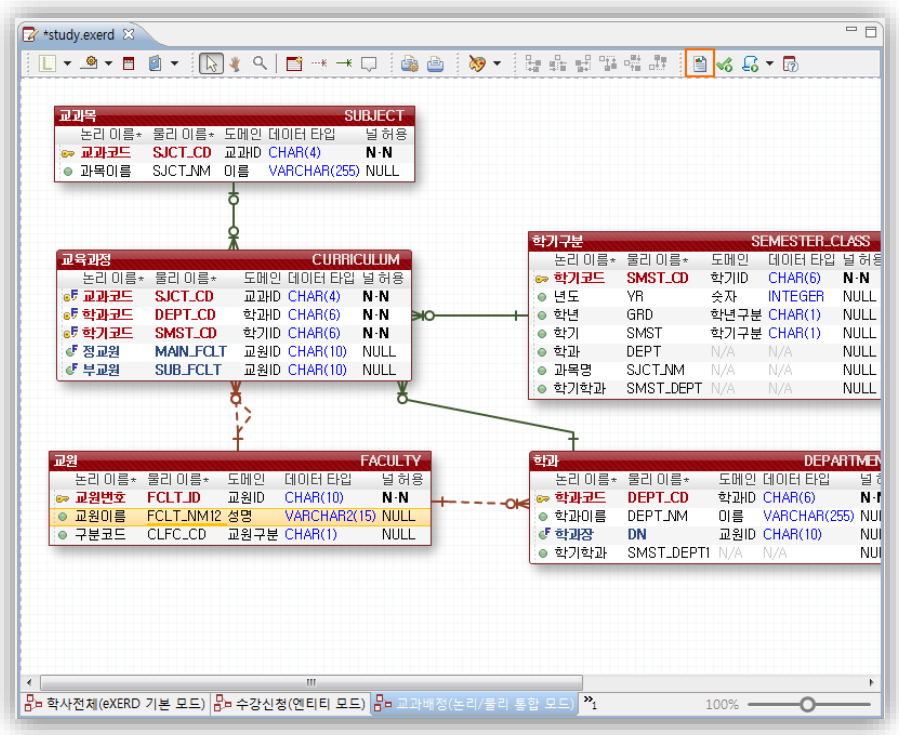
eXERD의 테이블 등의 모델 내용을 문서화(보고서) 할 때 이용하는 기능입니다.
이 기능을 이용하면 테이블 명세서와 같은 문서를 만들 수 있습니다.

■ 템플릿

모델을 HTML이나 XML등으로 내보 낼 수 있고 내보내는 양식을 편집 할 수도 있습니다.

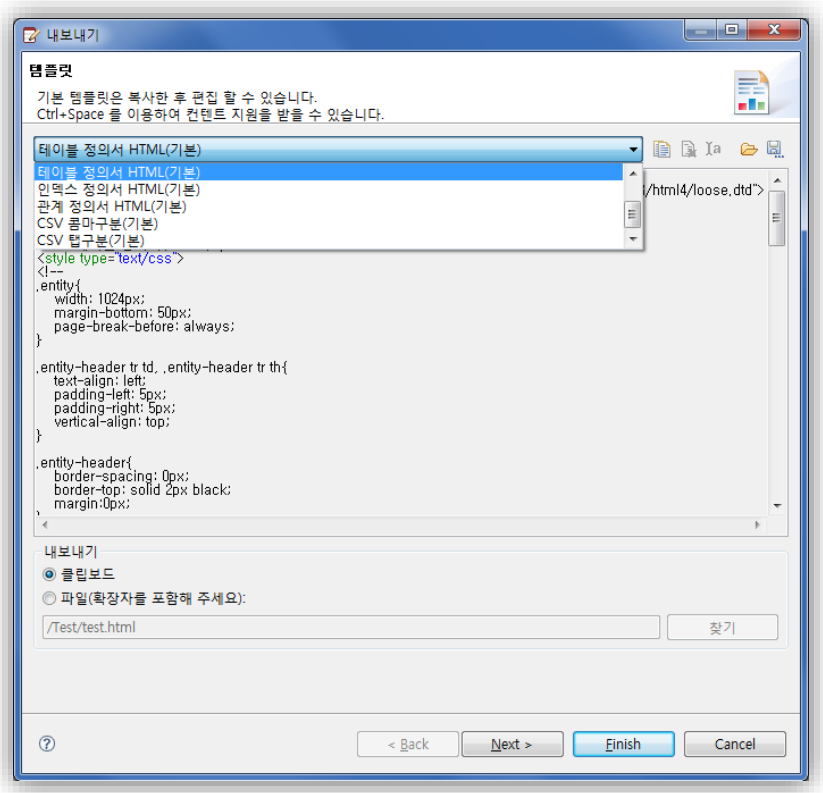
Tip! *CSV 형식으로 내보낸 파일은 엑셀(Excel)에서 바로 불러들일 수 있습니다.*

- 다이어그램 툴바에서 **템플릿**버튼을 실행합니다.



■ 양식 선택하기

- 모델을 여러가지 템플릿을 이용하여 내보 낼 수 있습니다.
- 아래 양식들이 기본으로 제공됩니다.
  - HTML 표
  - XML
  - 스타일 적용된 HTML
  - 테이블 정의서 HTML
  - 인덱스 정의서 HTML
  - 관계 정의서 HTML
  - CSV 콤마구분
  - CSV 탭구분

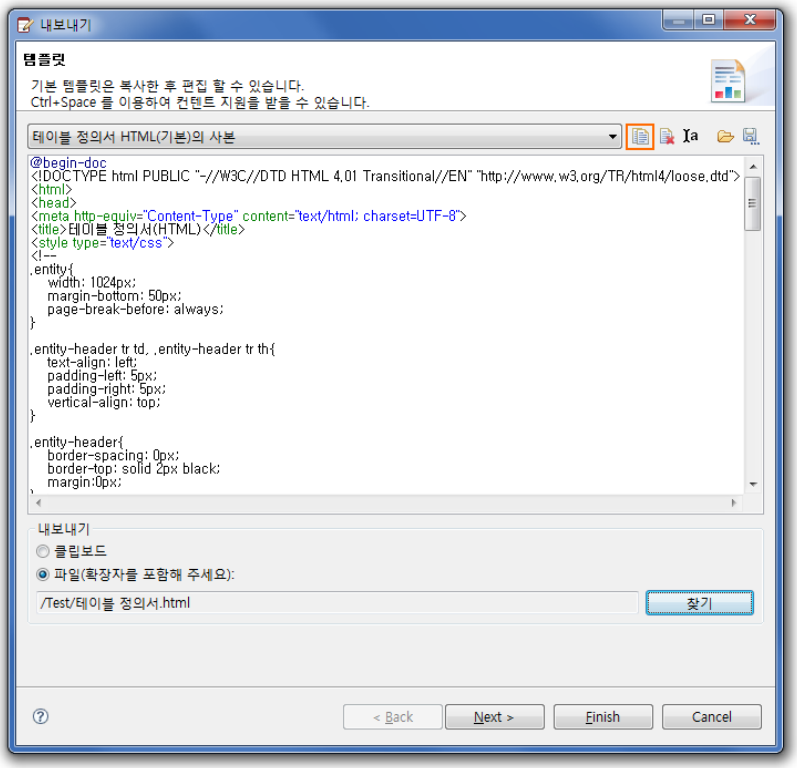


Note! *기본적으로 내장 되어 있는 양식은 편집 할 수 없습니다.*

■ 템플릿 편집하기

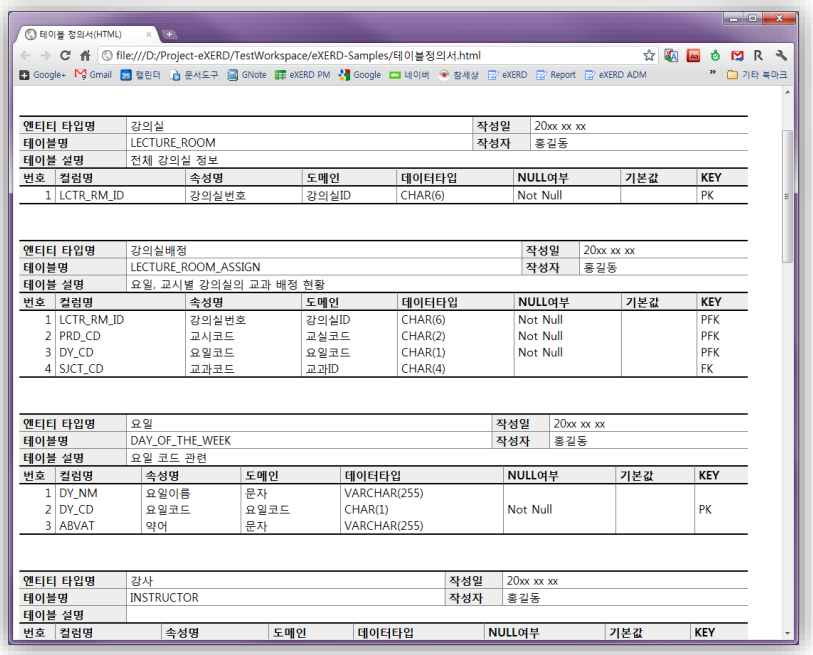
템플릿 선택 콤보박스 우측의 **템플릿 복사** 버튼을 이용하여 복사본에 대해 편집 할 수 있습니다.

▪ 양식을 편집 할 때 **Ctrl+Space**를 누르면 변수에 대한 컨텐트 어시스트가 지원됩니다.



▪ 양식에 의해 생성된 내용을 클립보드로 보내거나 파일로 내보낼 수 있습니다.

▪ 파일로 내보내는 경우 **[완료]**버튼을 누르면 아래와 같이 생성된 HTML이 표시됩니다.



테이블 정의서(HTML)

file:///D:/Project-eXERD/TestWorkspace/eXERD-Samples/테이블정의서.html

엔티티 타입명	강의실	작성일	20xx xx xx
테이블명	LECTURE_ROOM	작성자	홍길동
테이블 설명	전체 강의실 정보		

번호	컬럼명	속성명	도메인	데이터타입	NULL여부	기본값	KEY
1	LCTR_RM_ID	강의실번호	강의실ID	CHAR(6)	Not Null		PK

엔티티 타입명	강의실배정	작성일	20xx xx xx
테이블명	LECTURE_ROOM_ASSIGN	작성자	홍길동
테이블 설명	요일, 교시별 강의실의 교과 배정 현황		

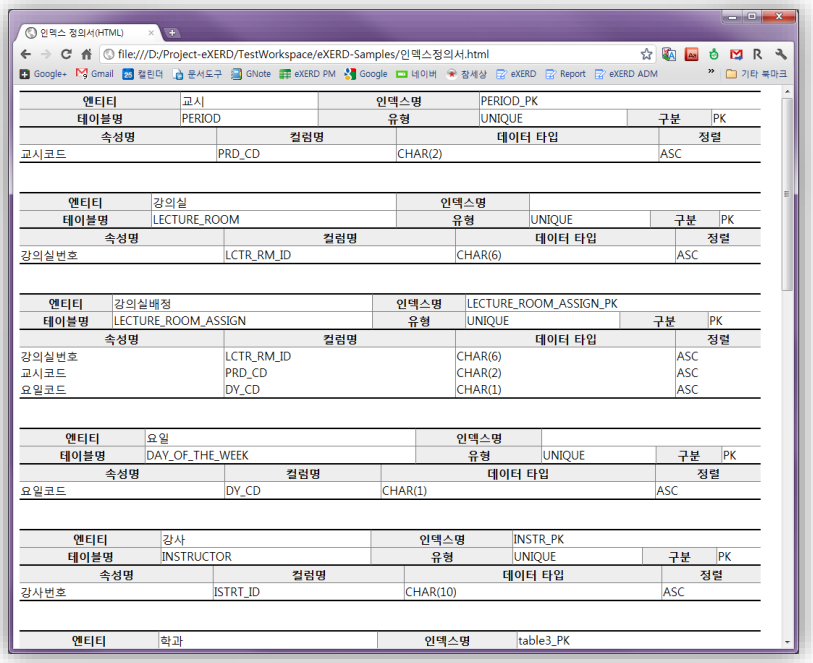
번호	컬럼명	속성명	도메인	데이터타입	NULL여부	기본값	KEY
1	LCTR_RM_ID	강의실번호	강의실ID	CHAR(6)	Not Null		PFK
2	PRD_CD	교시코드	교실코드	CHAR(2)	Not Null		PFK
3	DY_CD	요일코드	요일코드	CHAR(1)	Not Null		PFK
4	SJCT_CD	교과코드	교과ID	CHAR(4)			FK

엔티티 타입명	요일	작성일	20xx xx xx
테이블명	DAY_OF_THE_WEEK	작성자	홍길동
테이블 설명	요일 코드 관련		

번호	컬럼명	속성명	도메인	데이터타입	NULL여부	기본값	KEY
1	DY_NM	요일이름	문자	VARCHAR(255)			
2	DY_CD	요일코드	요일코드	CHAR(1)	Not Null		PK
3	ABVAT	약어	문자	VARCHAR(255)			

엔티티 타입명	강사	작성일	20xx xx xx
테이블명	INSTRUCTOR	작성자	홍길동
테이블 설명			

번호	컬럼명	속성명	도메인	데이터타입	NULL여부	기본값	KEY



인덱스 정의서(HTML)

file:///D:/Project-eXERD/TestWorkspace/eXERD-Samples/인덱스정의서.html

엔티티	교시	인덱스명	PERIOD_PK		
테이블명	PERIOD	유형	UNIQUE	구분	PK

속성명	컬럼명	데이터 타입	정렬
교시코드	PRD_CD	CHAR(2)	ASC

엔티티	강의실	인덱스명			
테이블명	LECTURE_ROOM	유형	UNIQUE	구분	PK

속성명	컬럼명	데이터 타입	정렬
강의실번호	LCTR_RM_ID	CHAR(6)	ASC

엔티티	강의실배정	인덱스명	LECTURE_ROOM_ASSIGN_PK		
테이블명	LECTURE_ROOM_ASSIGN	유형	UNIQUE	구분	PK

속성명	컬럼명	데이터 타입	정렬
강의실번호	LCTR_RM_ID	CHAR(6)	ASC
교시코드	PRD_CD	CHAR(2)	ASC
요일코드	DY_CD	CHAR(1)	ASC

엔티티	요일	인덱스명			
테이블명	DAY_OF_THE_WEEK	유형	UNIQUE	구분	PK

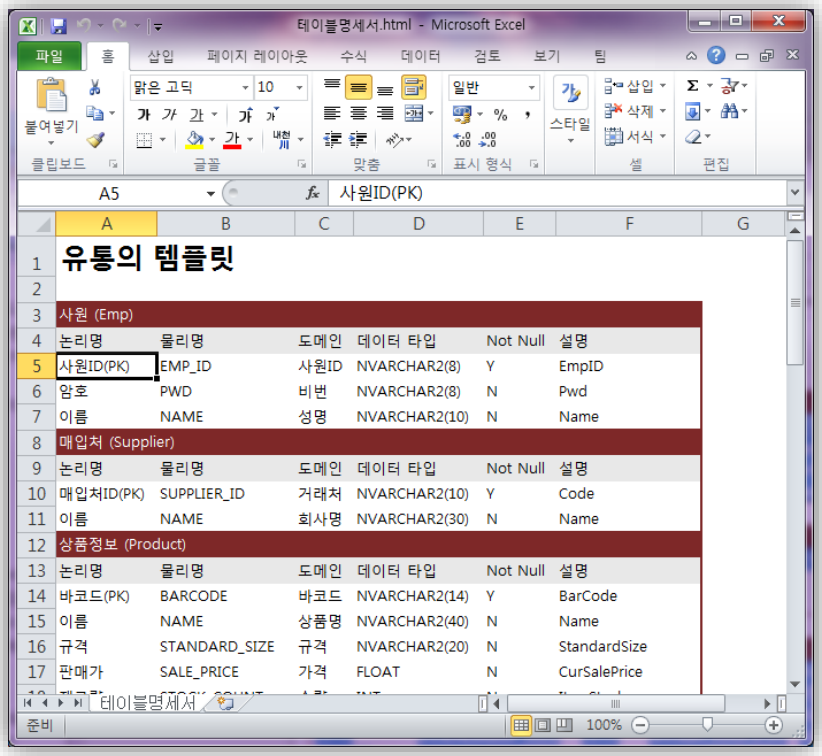
속성명	컬럼명	데이터 타입	정렬
요일코드	DY_CD	CHAR(1)	ASC

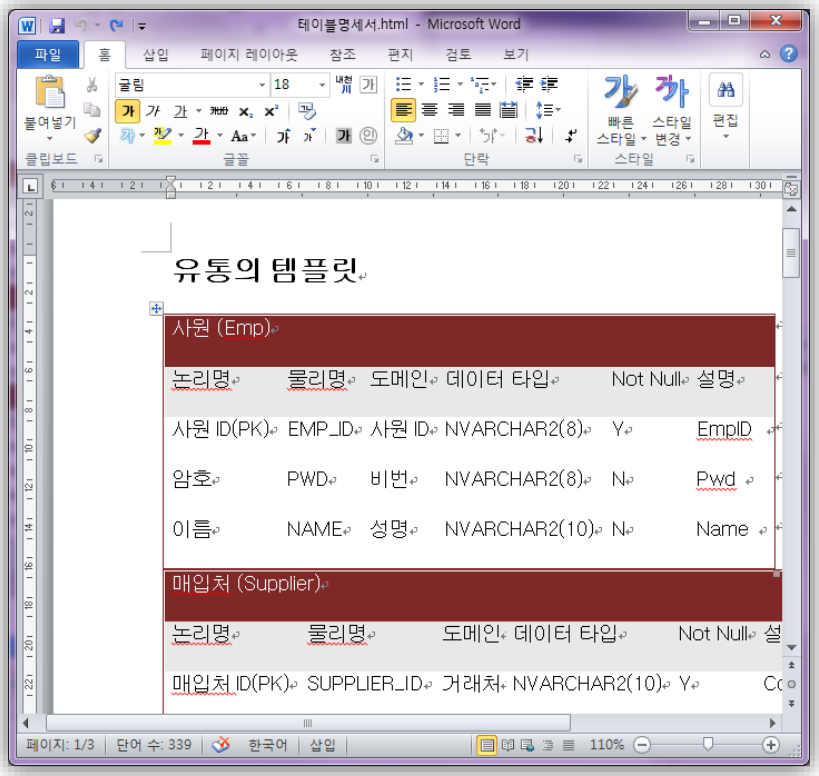
엔티티	강사	인덱스명	INSTR_PK		
테이블명	INSTRUCTOR	유형	UNIQUE	구분	PK

속성명	컬럼명	데이터 타입	정렬
강사번호	ISTRT_ID	CHAR(10)	ASC

엔티티	학과	인덱스명	table3_PK

▪ 생성한 HTML 파일을 **Excel**이나 **Word**에서 열어서 산출물 작업을 마무리합니다.

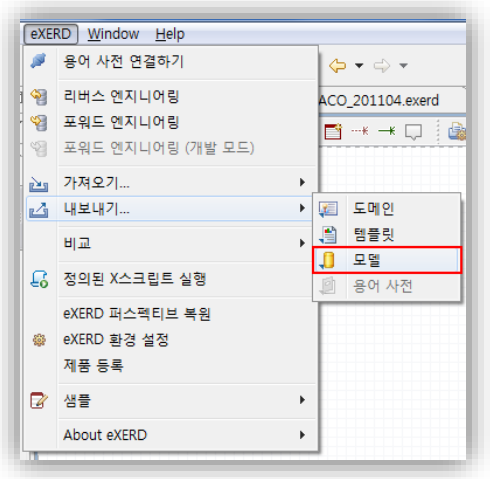




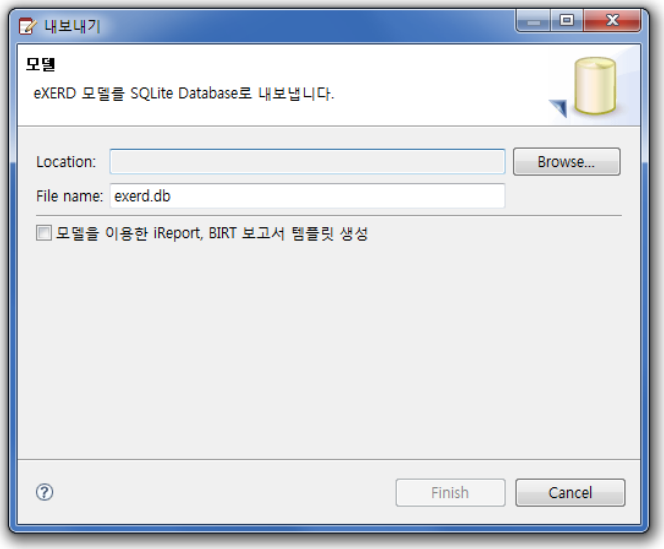
3. 모델 내보내기

eXERD 의 모델을 SQLite Database 로 내보냅니다.

1) eXERD 메뉴에서 내보내기 - 모델을 선택합니다.



2) 내보낼 위치와 파일 이름을 지정합니다.



3) 완료를 누릅니다.

4) 지정된 위치에 sqlite 파일이 생성됩니다.

5) 옵션 체크 했을 경우 하위 폴더로 birt, ireport 가 생성됩니다.

■ 테이블 정보

schemas

컬럼	설명
logical_name	스키마 논리명
physical_name	스키마 물리명
comment	스키마 주석

tables

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
logical_name	테이블 논리명
physical_name	테이블 물리명
comment	테이블 주석

columns

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
table_logical	테이블 논리명
table_physical	테이블 물리명
logical_name	컬럼 논리명
physical_name	컬럼 물리명
domain	컬럼의 도메인 이름
datatype	컬럼의 데이터타입
column_order	컬럼 순서
column_type	컬럼 유형(PK,FK 등)
nullable	NULL 여부(Not Null, 공백)
comment	컬럼 주석
default_value	기본값

indexes

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
table_logical	테이블 논리명
table_physical	테이블 물리명
logical_name	인덱스 논리명
physical_name	인덱스 물리명
kind	종류(PK,FK,INDEX)
uniqueness	유니크 속성(UNIQUE, NON UNIQUE)

index_columns

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
table_logical	테이블 논리명
table_physical	테이블 물리명
index_logical	인덱스 논리명
index_physical	인덱스 물리명
logical_name	컬럼 논리명
physical_name	컬럼 물리명
comment	컬럼 주석
datatype	컬럼의 데이터타입
sort	컬럼의 인덱스 정렬
nullable	컬럼의 NULL 여부(Not Null,공백)
domain	컬럼에 지정된 도메인 이름

relations

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
table_logical	테이블 논리명(자식 테이블)
table_physical	테이블 물리명(자식 테이블)
parent_table_logical	부모 테이블의 논리명
parent_table_physical	부모 테이블의 물리명
logical_name	관계 논리명
physical_name	관계 물리명
comment	관계 주석
identity_kind	식별 종류(비식별,식별)
child_cardinality	관계 차수 (EXACTLY_ONE, ZERO_OR_MORE,ONE_OR_MORE,ZERO_OR_ONE)
parent_cardinality	관계 차수 (EXACTLY_ONE, ZERO_OR_MORE,ONE_OR_MORE,ZERO_OR_ONE)

relation_columns

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
table_logical	테이블 논리명
table_physical	테이블 물리명
relation_logical	관계 논리명
relation_physical	관계 물리명
logical_name	컬럼 논리명
physical_name	컬럼 물리명
comment	컬럼 주석
column_type	컬럼의 타입(PK,FK)
datatype	컬럼의 데이터타입
nullable	컬럼의 NULL 여부(Not Null,공백)
domain	컬럼에 지정된 도메인 이름
parent_column_logical	부모 컬럼의 논리명
parent_column_physical	부모 컬럼의 물리명
parent_column_datatype	부모 컬럼의 데이터타입
parent_table_logical	부모 테이블의 논리명
parent_table_physical	부모 테이블의 물리명

routines

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
type	타입(PROCEDURE,FUNCTION)
logical_name	논리명
physical_name	물리명
comment	관계 주석
definition	정의

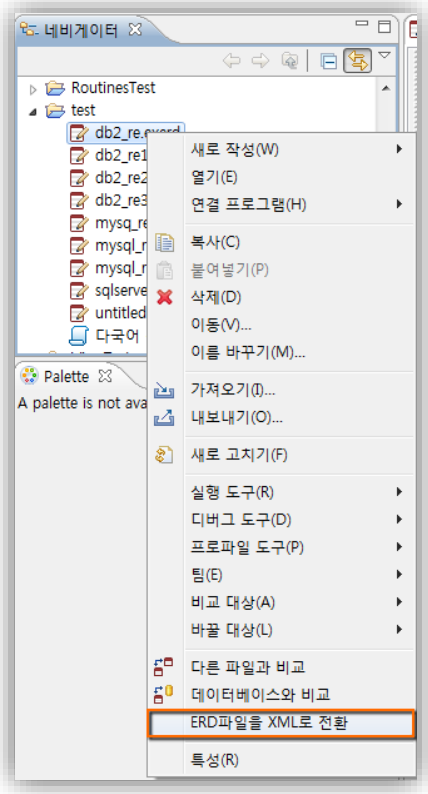
views

컬럼	설명
schema_logical	스키마 논리명
schema_physical	스키마 물리명
logical_name	논리명
physical_name	물리명
comment	관계 주석
definition	정의

4. XML 파일로 변환

ERD 파일을 XML형식의 erdx파일로 변환합니다.

Exerd의 파일을 선택하여 "ERD파일을 XML로 전환" 선택하여 변환합니다.



환경설정

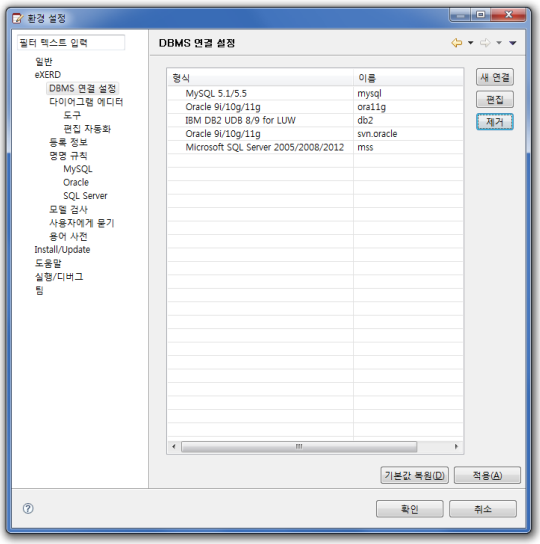
- DBMS 연결 설정
- 다이어그램 에디터
- 명명 규칙
- 모델 검사
- 사용자에게 묻기
- 용어 사전

환경설정

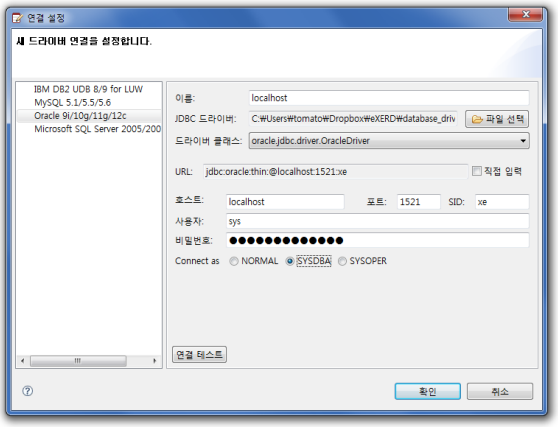
1. DBMS 연결 설정

eXERD > DBMS 연결 설정에서 데이터베이스 연결 설정을 관리 할 수 있습니다.

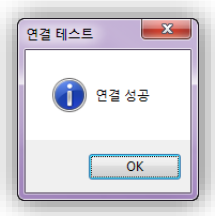
1) 데이터베이스 연결을 생성하고 편집하고 제거 할 수 있습니다.



2) **새 연결** 버튼을 눌러 **연결 설정** 다이얼로그를 열면 아래와 같이 연결 설정을 만들 수 있습니다.



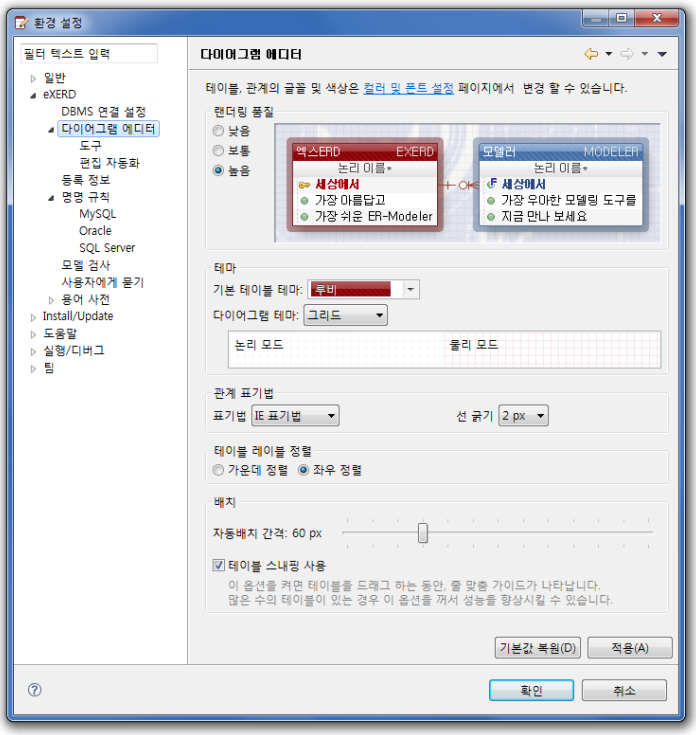
3) 설정을 마친 후 **연결 테스트** 버튼을 눌러 연결이 올바르게 되었는지 확인합니다.



Note! *연결이 올바르게 되었을 경우 위와 같은 확인 창이 나타납니다.*

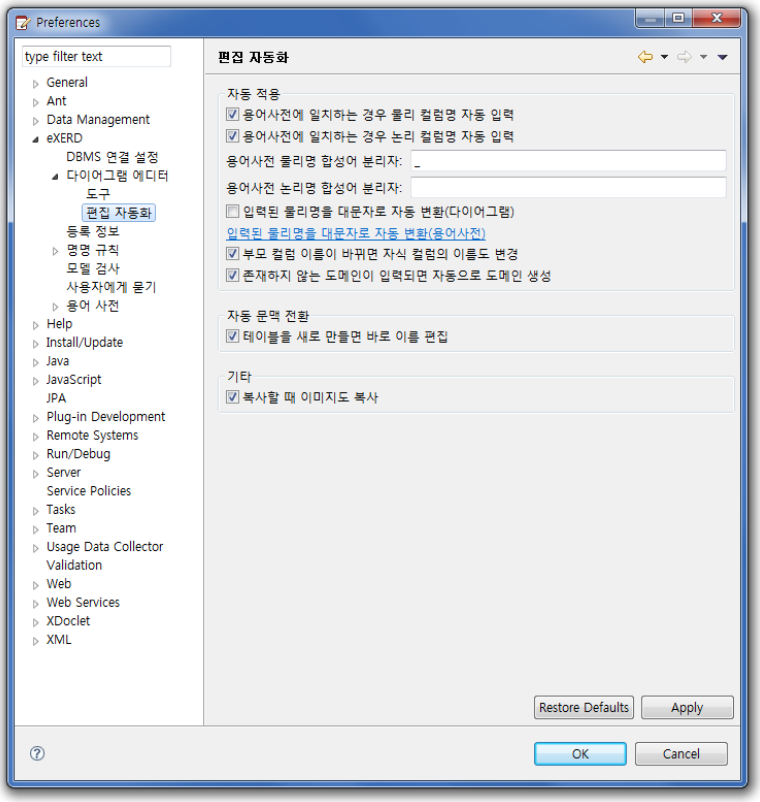
2. 다이어그램 에디터

eXERD > 다이어그램 에디터에서 다이어그램의 표시 방식등의 설정을 할 수 있습니다.
랜더링 품질을 변경하여 속도를 보장하거나 효과를 보장하도록 설정 할 수 있습니다.
렌더링 품질은 시스템 성능에 따라 화면 반응속도에 많은 영향을 받으므로 적절하게 조정하시기 바랍니다.



- 관계선의 표기 방식을 IE 표기법과 Barker 표기법 중에서 선택할 수 있습니다.
- 다이어그램 설정에서 자동배치 시 테이블간의 간격을 설정할 수 있습니다.
- 테이블을 Drag 할 때 나타나는 안내선(스내핑) 기능을 켜고 끌 수 있습니다.
- 새로 만들어질 테이블에 적용될 기본 테이블 테마를 지정할 수 있습니다.
- 테이블의 이름을 좌우 정렬 또는 가운데 정렬로 설정할 수 있습니다.
- 다이어그램 테마를 설정할 수 있습니다.

편집 자동화를 통하여 물리명 대문자 자동 변환 등 자동 변환에 관련된 설정을 할수 있습니다.

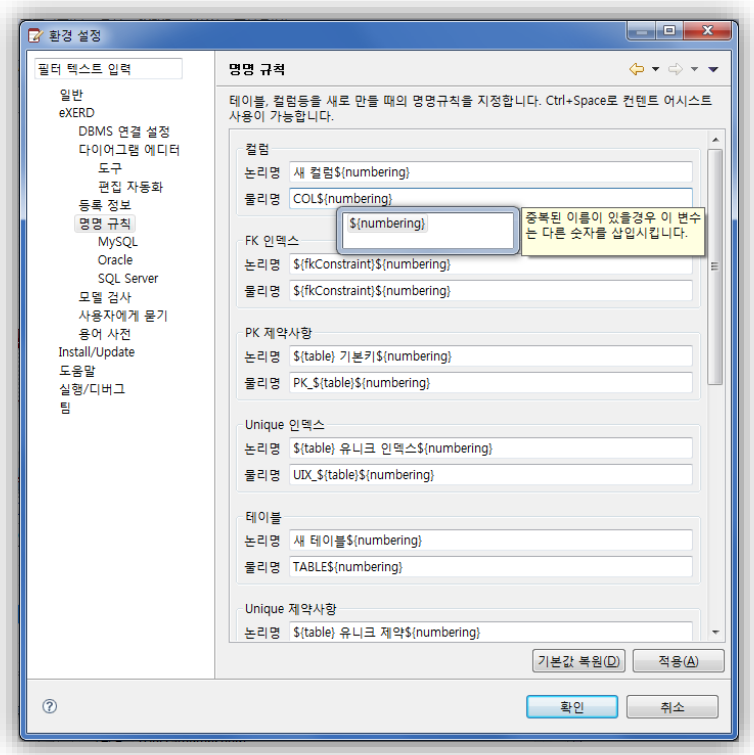


- 용어사전에 일치하는 경우 논리 컬럼명 자동 입력 [**기본 값 : ON**]
- 용어사전에 일치하는 경우 물리 컬럼명 자동 입력 [**기본 값 : ON**]
- 입력된 물리명을 대문자로 자동 변환 [**기본 값 : OFF**]
- 부모 컬럼 이름이 바뀌면 자식 컬럼의 이름도 변경 [**기본 값 : ON**]
- 존재하지 않는 도메인이 입력되면 자동으로 도메인 생성 [**기본 값 : ON**]
- 테이블을 새로 만들면 바로 이름 편집 [**기본 값 : ON**]
- 복사할 때 이미지도 복사 [**기본 값 : ON**]

3. 명명 규칙

eXERD > 명명 규칙

- 테이블 또는 컬럼 등을 새로 만들 때의 명명 규칙을 지정 할 수 있습니다.



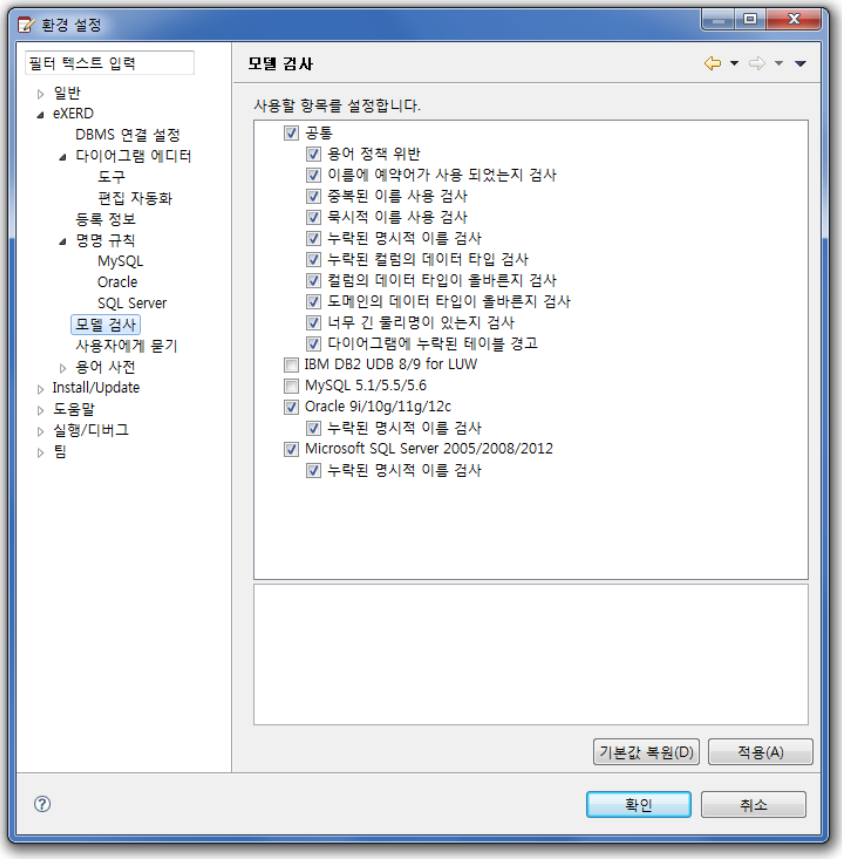
***Note!** 명명 규칙을 지정 할 때에도 **컨텐트 어시스트(CTRL + SPACE)**가 지원됩니다.*

- 프로젝트의 속성 대화 상자에서, eXERD 명명 규칙을 지정하면, 프로젝트 전용 명명 규칙을 지정할 수 있습니다.

4. 모델 검사

eXERD > 모델 검사에서 검토 뷰에서 사용 할 벨리데이터를 설정 할 수 있습니다.

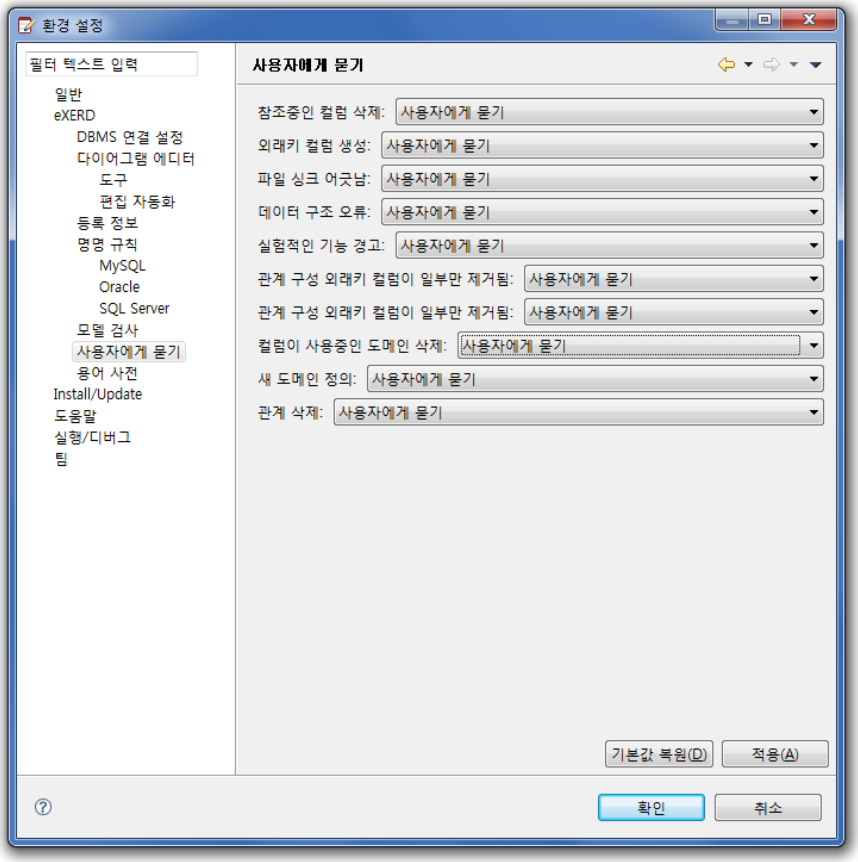
- 검토 뷰에서 모델 검사를 할 때 환경설정에서 선택한 옵션만 사용하여 모델 검사를 할 수 있습니다.



5. 사용자에게 묻기

eXERD > 사용자에게 묻기에서 사용자가 ERD를 디자인 하고 편집 할 때 발생 할 수 있는 행위에 관해 사용자에게 질문하는 것에 대한 설정을 할 수 있습니다.

- 특정 행위에 관해 그 해답을 사용자에게 묻거나 미리 지정 해 둔 행위를 수행하고 사용자에게 다시 묻지 않을 수도 있습니다.

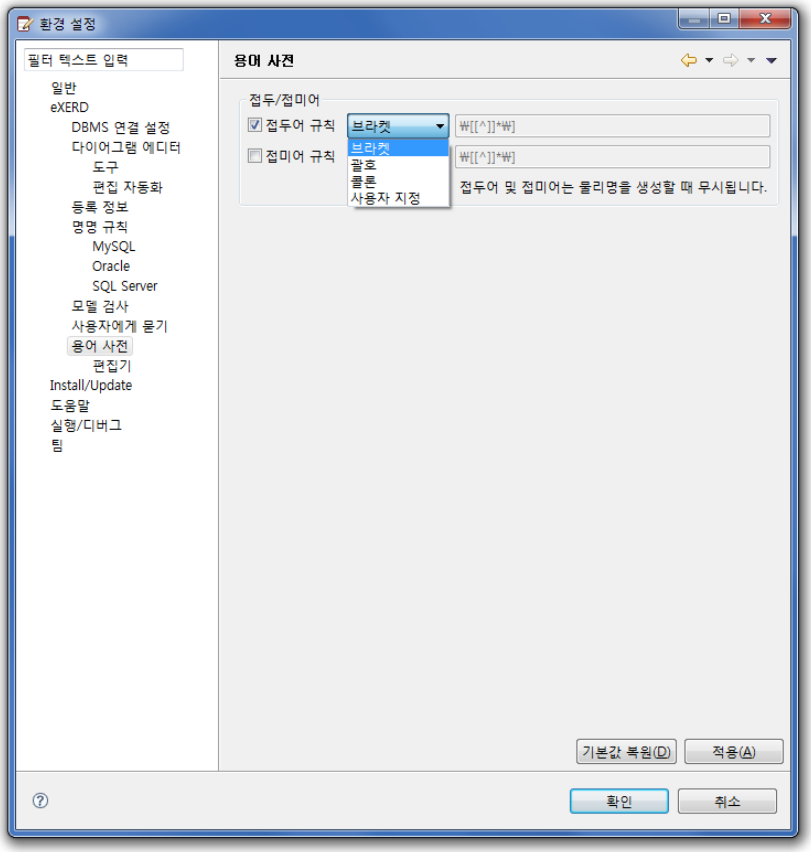


6. 용어 사전

eXERD > 용어 사전

- 컬럼의 물리명을 제안할 때 사용하는 논리명에 대한 접두어/접미어 패턴을 설정할 수 있습니다.

  ex) "학과코드[CD]"와 같은 논리명에서 "[CD]" 부분을 접미어 패턴으로 제외하여 물리명을 제안합니다.



편집기 *탭에서는 물리명 편집시 소문자를 대문자로 자동 변환 등을 설정할수 있습니다.*

- ***테이블 편집기 테마 적용*** *: 맥 또는 일분 환경에서 성능이나 표시 문제가 발생하는 경우, 이 옵션을 끄십시오. [**기본값 : ON**]*
- ***물리명 대문자 자동변환*** *: 용어사전 편집기 작성 또는 수정에서 대문자로 변환을 원하지 않는 경우, 이 옵션을 끄십시오. [**기본값 : ON**]*

기타

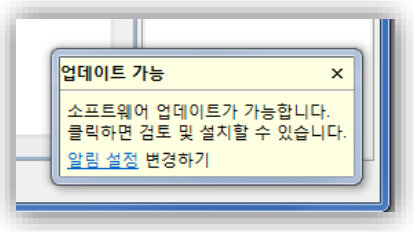
- 업데이트
- 도움말
- eXERD에서 지원하는 파일 형식
- eXERD 단축키
- eXERD FAQ

기타

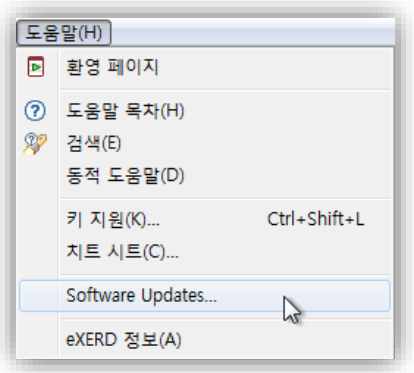
1. 업데이트

eXERD를 최신 버전으로 업데이트 합니다.

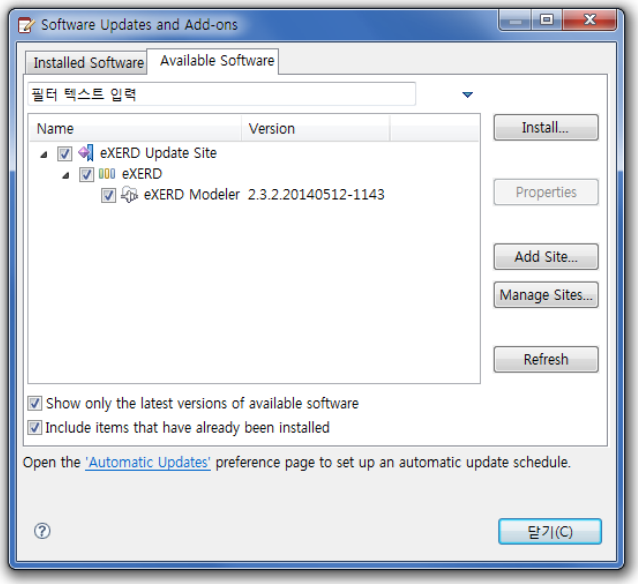
1) 인터넷이 연결 되어 있는 경우 eXERD 실행시 업데이트가 가능할 때 오른쪽 하단에 "업데이트 가능" 알림이 나타납니다.(자동 업데이트 알림)



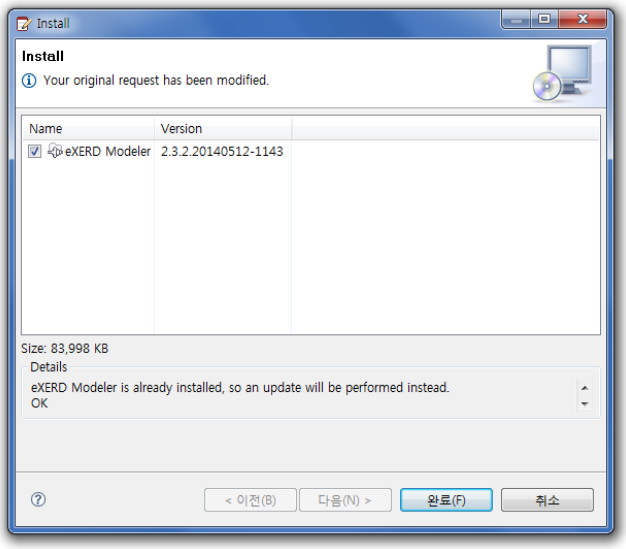
2) 메뉴 도움말-Software Updates... 를 선택합니다.(수동 업데이트)



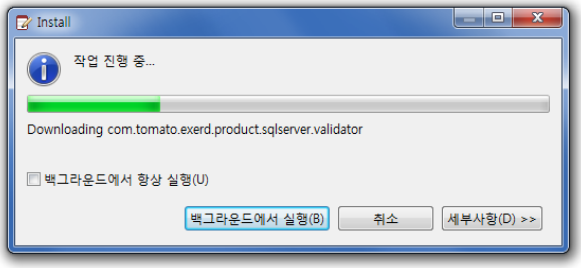
3) eXERD Update Site 를 확인을 누르시고, [Install...]를 클릭합니다.



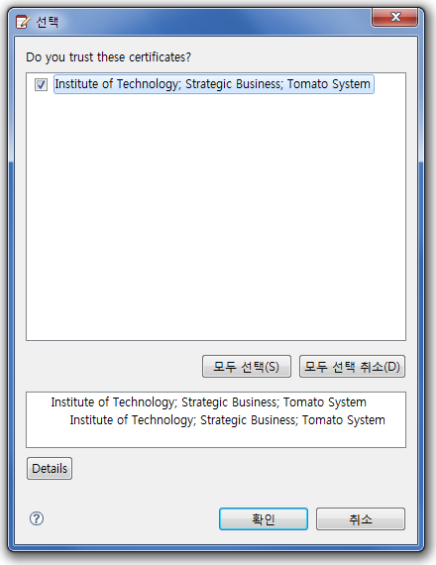
4) 진행에 문제가 없는 경우 [완료] 버튼을 클릭합니다.



5) 설치를 진행 합니다.

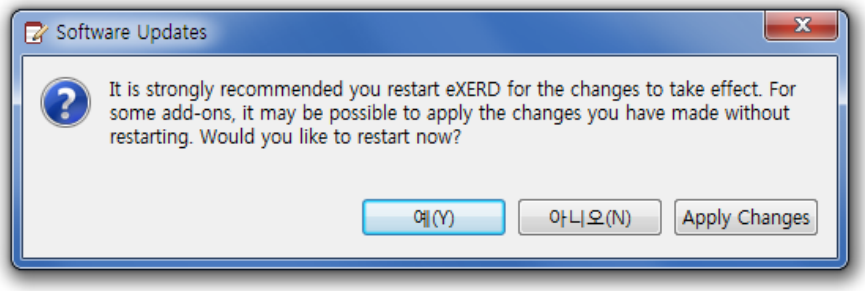


6) 설치 진행중 인증서 확인창에서 내용에 Institute of... 확인을 하고, [확인]을 클릭합니다.



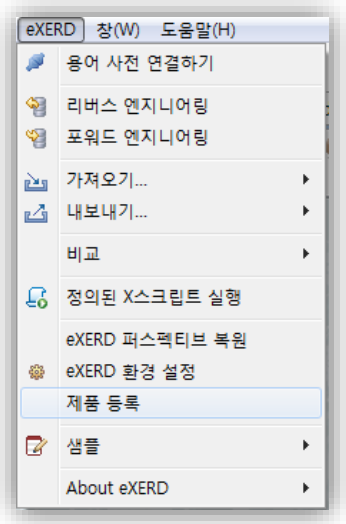
7) 설치가 완료되고 업데이트를 적용하기 위해서 프로그램 재시작을 묻습니다.

예: 재시작(권장) / **아니오**: 재시작 안함 / **Apply Changes**: 재시작 하지 않고 적용



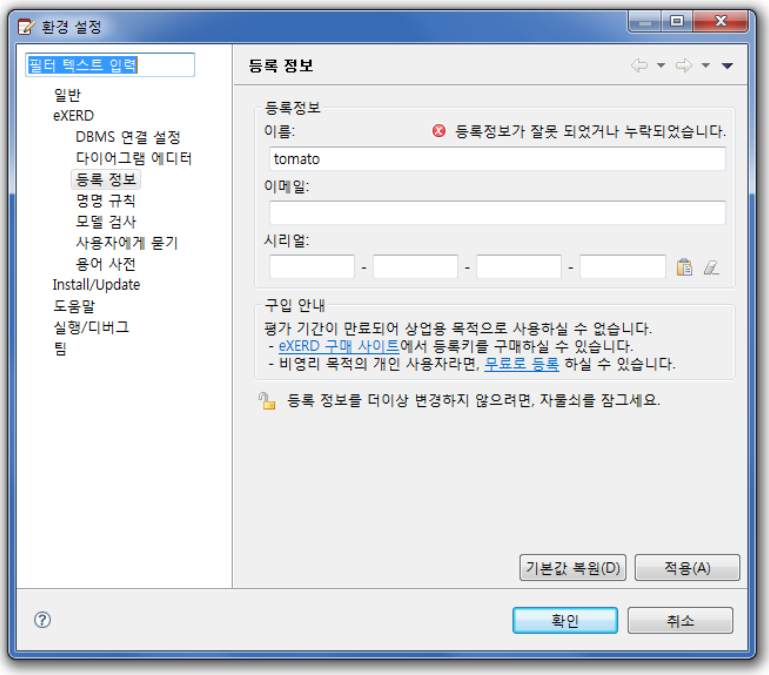
2. 제품등록

1) 라이선스 키를 등록합니다.



2) 라이선스 정보를 입력하고, [확인]을 클릭합니다.

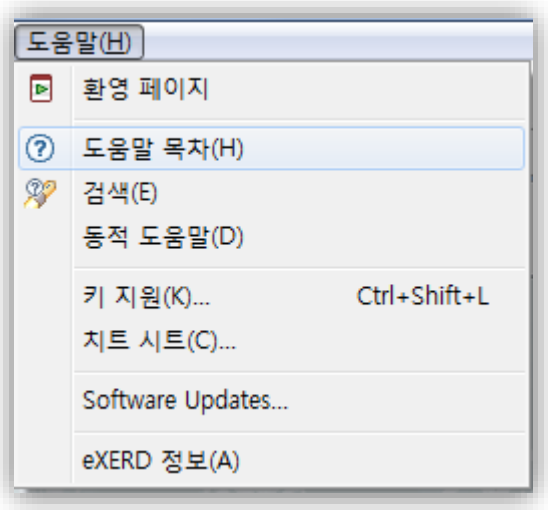
(적용 누르고 취소를 누르면 등록이 안될수도 있습니다.)



3. 도움말

사용자 설명서와 같은 정보를 가지고 있는 eXERD 도움말입니다. 도움말은 eXERD프로그램 및 홈페이지에서 확인하실 수 있습니다.

1) eXERD 메뉴에서 도움말-도움말 목차를 선택합니다.



2) 도움말 창이 열립니다.

4. eXERD에서 지원하는 파일 형식

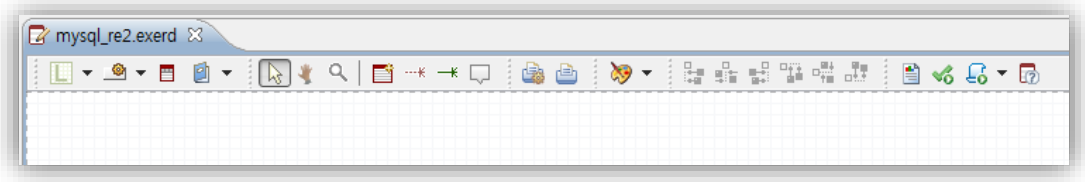
확장자	설명
exrd	erd 기본 확장자 입니다.
erdx	erd 파일을 XML 형식의 구조로 저장합니다.
xdmn	도메인을 내보내는 확장자입니다.
xdic	용어 사전 파일을 저장하는 확장자입니다.
xs	XScript를 저장하는 확장자입니다.

5. eXERD 단축키

단축키를 이용하여 작업 속도를 높일 수 있습니다.

■ 다이어그램 단축키

다이어그램에서 사용하는 단축키 목록



단축키	설명
F3	편집 모드를 반대 모드로 변경합니다. 단축키를 사용 할 때마다 논리 / 물리간의 편집 모드가 전환됩니다.
F4	논리 모드로 변경합니다.
F5	물리 모드로 변경합니다.
1	가장 기본적으로 사용 되는 선택 툴. (선택 / 이동 등)
2	다이어그램을 잡고 이동시킵니다.
3	테이블을 만듭니다.
4	비식별 관계를 만듭니다.
5	식별 관계를 만듭니다.
Ctrl + A	모든 구성요소를 선택합니다.
Ctrl + D	다이어그램을 전환합니다.
Ctrl + F	모델을 검색합니다.
Ctrl + SHIFT + F	다이어그램 내의 테이블을 자동으로 배치합니다.
Ctrl + P	다이어그램을 출력합니다.
Ctrl + Wheel Up, Ctrl + 더하기)	다이어그램을 한단계 확대합니다.
Ctrl + Wheel Down, CTRL + 빼기	다이어그램을 한단계 축소합니다.
Ctrl + 0	다이어그램을 원래 크기로 표시합니다.

■ 테이블 단축키

다이어그램에서 테이블이 선택 되었을 때 사용하는 단축키 목록

단축키	설명
Ctrl + Shift + Enter	선택 된 테이블에 새 기본 키 컬럼을 추가합니다.
Ctrl + Enter	선택 된 테이블에 새 컬럼을 추가합니다.
Alt + Delete	선택 된 테이블을 모델에서 완전히 삭제합니다.
`	선택된 테이블이나 컬럼, 다이어그램의 논리, 물리 이름 및 주석을 동시에 변경합니다.
Ctrl + UP	선택된 테이블의 부모 테이블들을 선택합니다.
Ctrl + DOWN	선택된 테이블의 자식 테이블들을 선택합니다.
Ctrl + R	선택된 테이블에 연결 된 모든 관계를 선택합니다.
Ctrl + C	선택된 개체들을 복사합니다.
Ctrl + O	테이블의 개요를 봅니다.
Ctrl + X	선택된 개체들을 잘라냅니다.
Delete	선택된 개체들을 삭제합니다. 테이블을 다이어그램에서 Delete로 삭제할 경우 실제모델은 삭제되지 않습니다.

■ 팔레트 뷰 단축키

팔레트 뷰에서 사용하는 단축키 목록

단축키	설명
1	가장 기본적으로 사용 되는 선택 툴. (선택 / 이동 등)
2	다이어그램을 잡고 이동시킵니다.
3	테이블을 만듭니다.
4	비식별 관계를 만듭니다.
5	식별 관계를 만듭니다.

■ 모델 뷰 단축키

모델 뷰에서 사용하는 단축키 목록

단축키	설명
F3	편집 모드를 반대 모드로 변경합니다. 단축키를 사용 할 때 마다 논리 / 물리간의 편집 모드가 전환됩니다.
F4	논리 모드로 변경합니다.
F5	물리 모드로 변경합니다.
`	선택된 테이블이나 컬럼의 논리, 물리 이름 및 주석을 동시에 변경합니다.

■ 도메인 뷰 단축키

도메인 뷰에서 사용하는 단축키 목록

단축키	설명
F2	선택 된 도메인의 이름을 변경합니다.
Enter	선택 된 도메인의 데이터 타입을 변경합니다.
Ctrl + Enter	도메인을 추가합니다.
Ctrl + Shift + Enter	선택 된 도메인에 자식 도메인을 추가합니다.
Delete	선택 된 도메인을 삭제합니다.
Ctrl + F	도메인을 검색합니다.

■ 검토 뷰 단축키

검토 뷰에서 사용하는 단축키 목록

단축키	설명
F5	모델 검사를 수행합니다.
Ctrl + 1	선택 된 문제를 수정합니다.

■ 용어사전 편집기 단축키

용어사전 편집기에서 사용하는 단축키 목록

단축키	설명
Ctrl + Enter	새 용어를 추가합니다.
Ctrl + T	선택 된 용어에 태그를 지정합니다.
Delete	선택 된 용어를 삭제합니다.
Ctrl + F	용어를 검색합니다.

6. FAQ

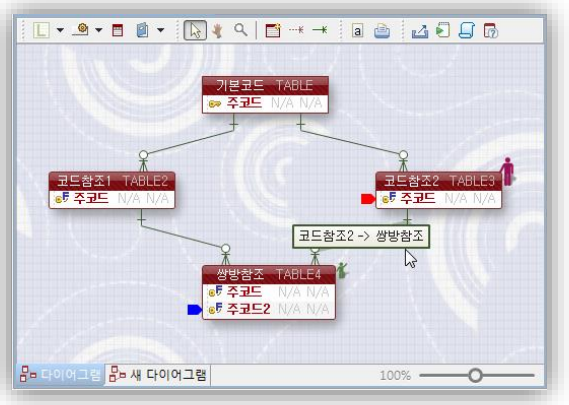
Q1. 한 컬럼이 서로 다른 여러 테이블을 참조하게 하려면 어떻게 하나요?

A1. 테이블의 속성 대화상자에서 지정할 수 있습니다.

Note! *eXERD에서는 관계를 연결할 때 자동으로 추가되는 참조컬럼과 같은 이름의 컬럼이 이미 존재하는 경우 해당 컬럼을 사용할 것인지 물어보지만 기존 컬럼이 이미 다른 테이블을 참조하고 있는 경우에는 실수를 방지하기 위해서 재사용하지 않고 새로운 참조컬럼으로 추가됩니다.*

간단한 예제를 이용하여 설명하도록 하겠습니다.

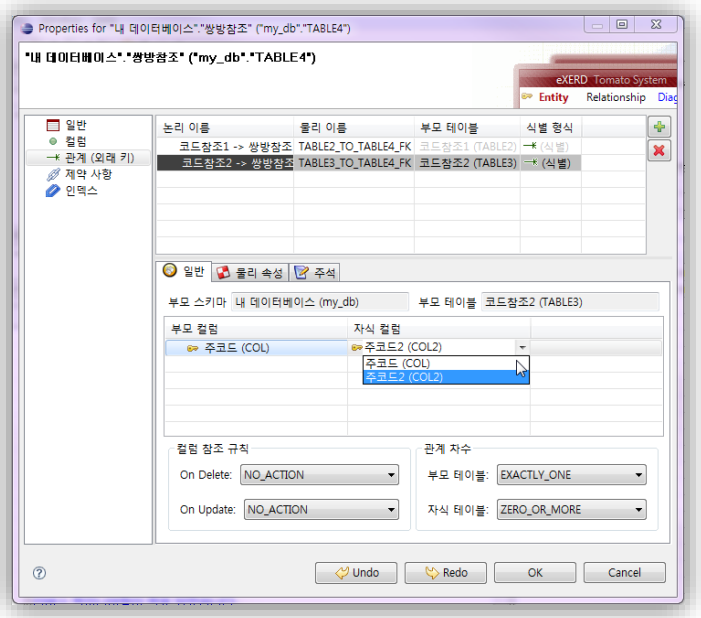
1) 아래와 같은 관계에서 **쌍방참조**테이블의 **주코드 2** 컬럼 대신에 **주코드**컬럼이 양쪽 부모테이블을 참조하게 변경할 수 있습니다.



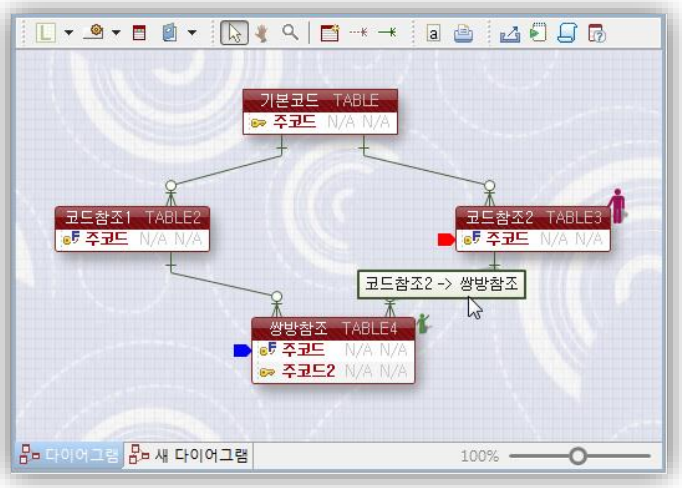
2) 쌍방참조 테이블을 선택한 후 **Space** 키를 눌러서 속성대화상자를 실행합니다.

3) 관계(외래키) 항목에서 **코드참조 2 -> 쌍방참조**관계를 선택합니다.

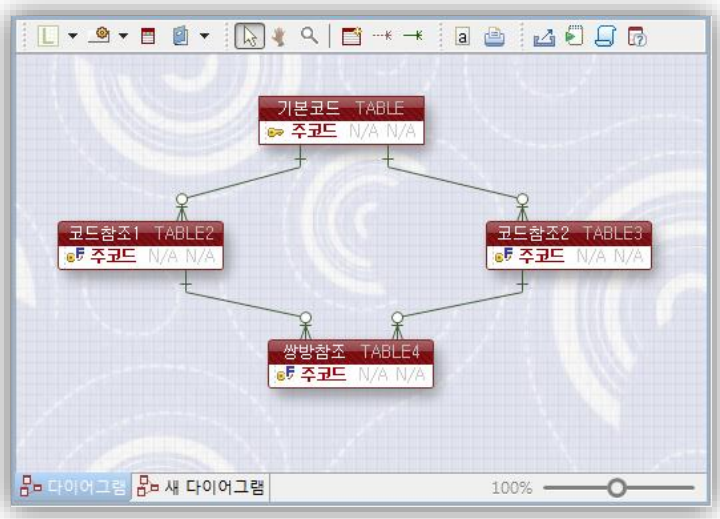
4) 자식컬럼을 기존 **주코드 2** 에서 **주코드**로 변경합니다.



5) 이제 **쌍방참조** 테이블의 **주코드** 컬럼이 **코드참조 2** 테이블도 참조하는 것을 확인 할 수 있습니다.



6) 이제 **쌍방참조** 테이블에서 필요없는 **주코드 2** 컬럼을 삭제하여 완료합니다.



Q2. ERWin 및 기존 모델링 툴 파일을 eXERD 파일로 변환할 수 있나요?

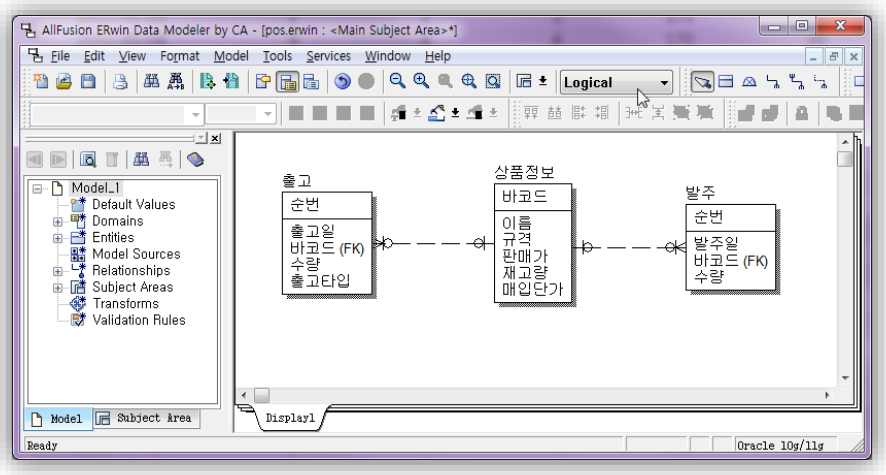
A2. ERwin의 경우 XML로 저장한 후 불러올 수 있습니다. 다른 모델링 도구인 경우, eXERD의 리버스 엔지니어링 기능을 이용하여 간접적으로 가져올 수 있습니다.

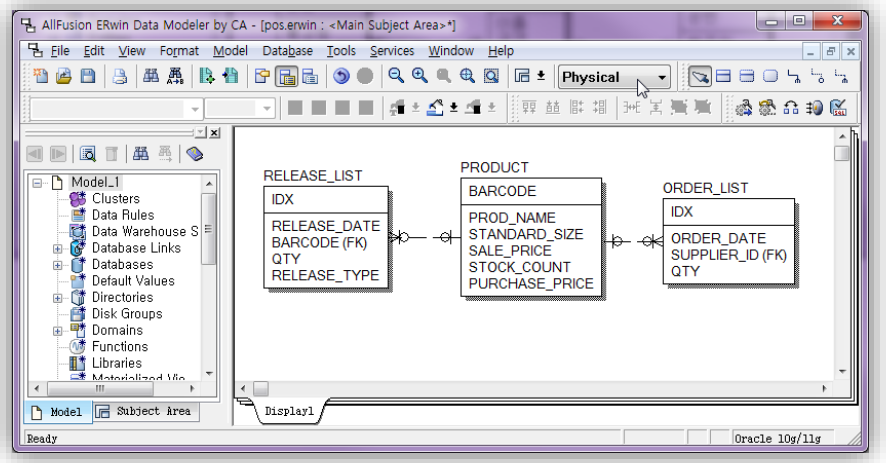
ERWin 파일을 Oracle에 포워드엔지니어링으로 생성 후 eXERD에서 리버스엔지니어링으로 가져오는 방법을 간단히 안내해드립니다.

이 가이드를 참고하면 **ERWin에서 작업한 모델의 논리명 포함하여 eXERD에서 동일하게** 가져올 수 있습니다.

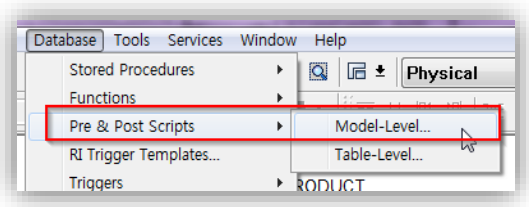
1) 아래와 같은 간단한 ERWin 파일을 예로 들어 설명합니다.

2) ***Note!*** *예제 파일은 논리명과 물리명이 각각 입력되어 있습니다.*

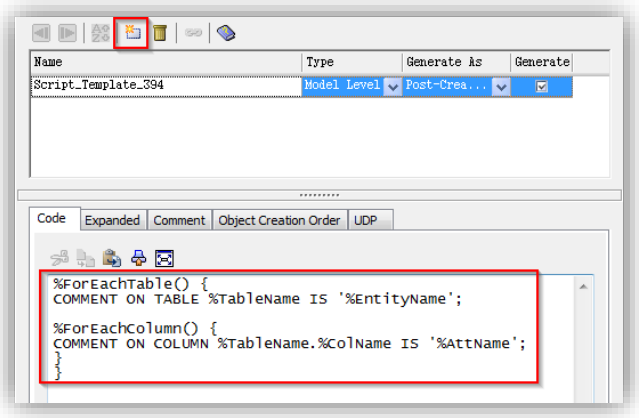




3) ERWin의 메뉴에서 **Database > Pre & Post Script > Model-Level**를 실행합니다.



4) 테이블의 논리명과 컬럼의 논리명을 Oracle의 COMMENT 구문으로 만들도록 새로운 Post Script를 작성합니다.

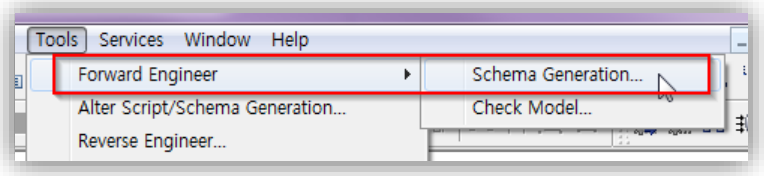


Note! *모든 테이블의 모든 컬럼에 대해 적용되도록 ERWin의 매크로를 이용하여 아래와 같이 작성합니다.*

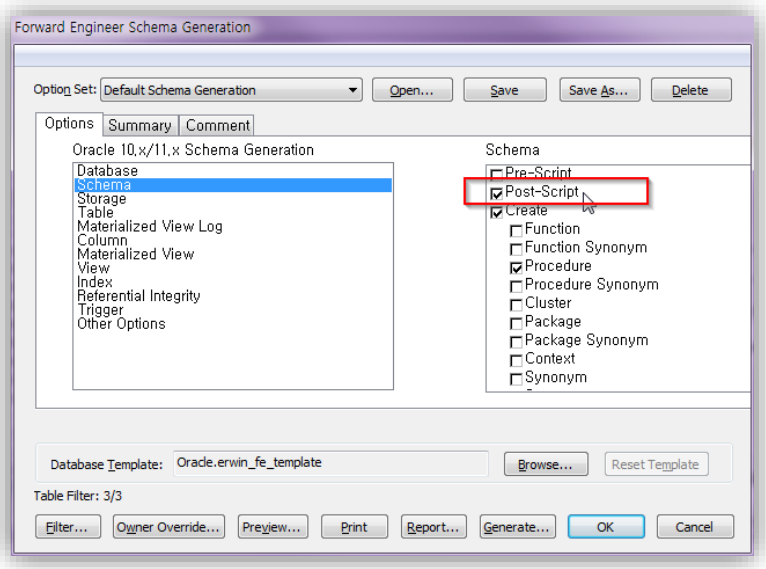
```
%ForEachTable() {
 COMMENT ON TABLE %TableName IS '%EntityName';

 %ForEachColumn() {
 COMMENT ON COLUMN %TableName.%ColName IS '%AttName';
 }
}
```

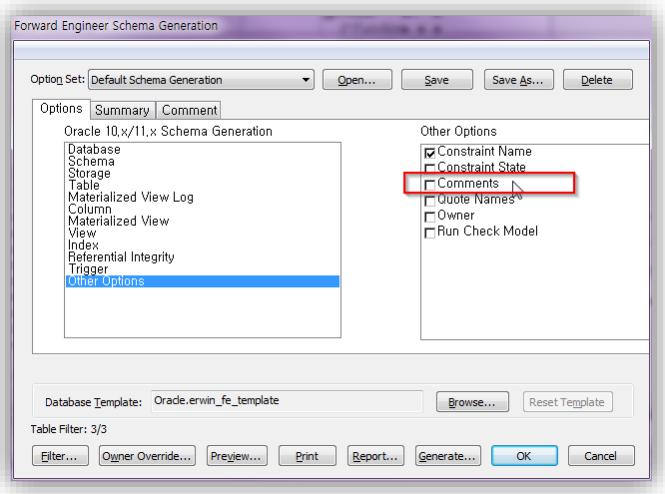
5) 이제 **Tools > Forward Engineer > Schema Generation** 을 실행합니다.



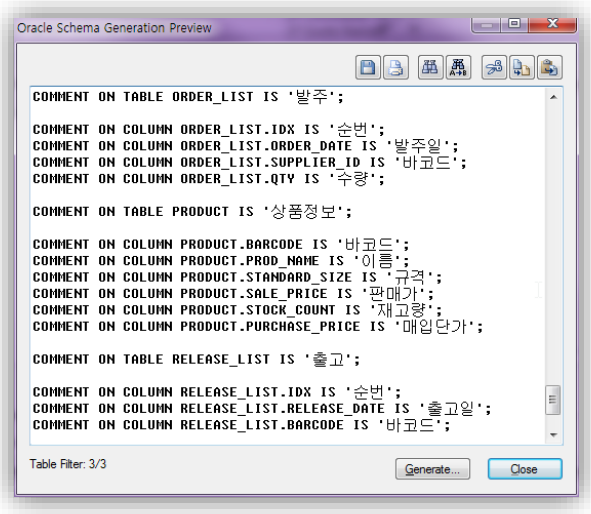
6) Options 탭에서 Schema 항목의 Post-Script를 체크합니다.



7) Other Options 항목에서 Comments를 제외시킵니다.

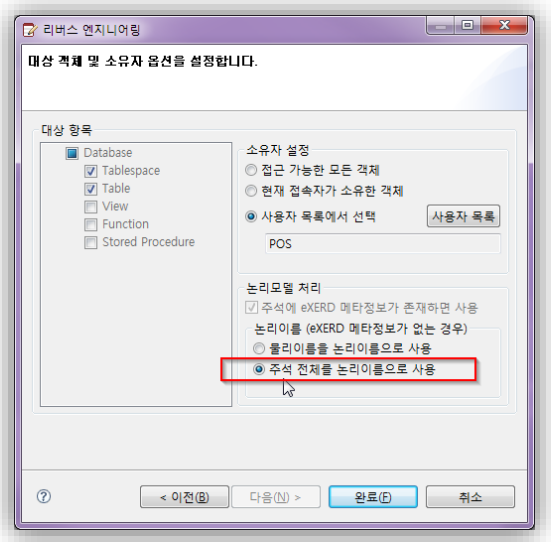


8) DDL에서 테이블의 논리명 및 컬럼의 논리명이 Oracle COMMENT 구문으로 생성됩니다. 이 DDL을 실제 Oracle에 실행하여 테이블을 생성합니다.

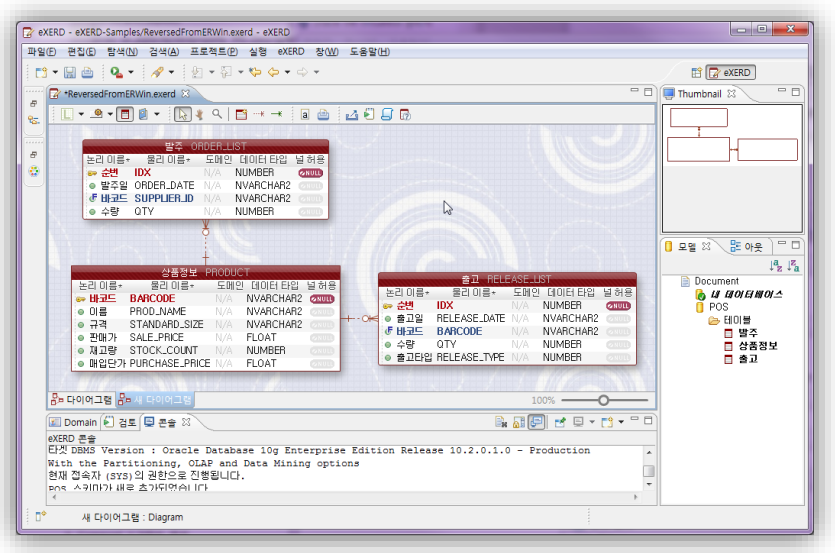


9) 이제 Oracle에 생성된 테이블들을 eXERD에서 리버스 엔지니어링으로 가져옵니다. 이때 **'주석 전체를 논리이름으로 사용'**옵션을 선택합니다.

10) ***Note!*** *eXERD의 메뉴 : eXERD > 리버스 엔지니어링*

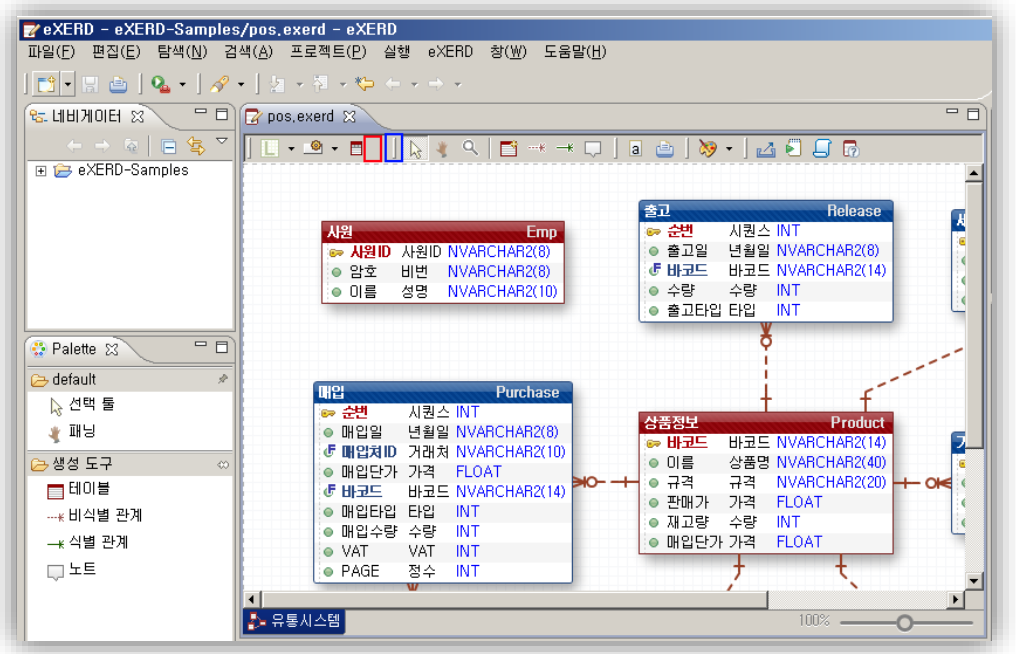


11) 리버스 엔지니어링이 완료되고 나면 아래와 같이 논리/물리명이 모두 정상적으로 표시되는 것을 확인 할 수 있습니다.



Q3. 다이어그램 툴바에서 용어사전 아이콘이 안 보입니다.

A3. 이 경우는 보통 Windows XP 고전 테마를 하신 분에게만 해당하는데 이것은 용어사전이 배치되어 있는 자리를 늘려 주시면 됩니다.



빨간색 영역에 용어사전 아이콘이 보여야 하는데 안 보이시는 분들은 파란색 영역의 바를 잡아서 늘려주시면 보입니다.

***Tip!**단축키 **Ctrl + 3** 으로 **용어사전**을 검색해 보세요.*

Q4 "ERwin 가져오기" 에서 "지원되지 않는 DB파일입니다." 메세지가 나옵니다.

A4. ERwin에서 XML 저장하기 전에 지정되어있는 DBMS또는 DBMS 버전이 맞지 않는 문제입니다.

ERwin에서 지정되어있는 DBMS와 DBMS의 버전을 확인해 주세요. 또는 XML을 열어 Target_Server_Version 을 확인하셔도 됩니다.

ERwin 가져오기에서 지원 되는 버전은 아래와 같습니다.

지원 ERwin XML 파일 버전:

- 7.2, 7.3

지원 데이터베이스

- 오라클: 9, 10, 11
- MS-SQL: 2005 이상
- DB2: 8, 9
- mysql: 5.X

Q5 라이선스 등록이 안됩니다.

A5. 라이선스 코드는 받은 후 바로 사용할수 있습니다.

라이선스 입력란의 내용을 다시 한번 정확히 입력해 주시기 바랍니다.

이름 또는 메일을 복사해서 붙여넣는 과정 중 공백이 있을수도 있습니다.

Q6 eXERD 실행시 "JVM terminated. Exit code=-1" 메세지가 나옵니다.

A6. eXERD가 설치된 폴더의 eXERD.ini 파일에 설정된 메모리가 가용메모리에 비해 너무 큰 값을 설정된 경우 발생 할 수 있는 문제입니다.

eXERD.ini 파일에 설정된 메모리의 값을 더 작은 값으로 설정 한 후 실행해 보시기 바랍니다.

eXERD.ini 파일의 기본 설정
-showsplash com.tomato.exerd.core --launcher.XXMaxPermSize 256M -framework plugins₩org.eclipse.osgi_3.4.3.R34x_v20081215-1030.jar -vmargs -Dosgi.requiredJavaVersion=1.6 -Xms128m -Xmx512m -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+UseParNewGC

-Xms128m 옵션

eXERD가 실행될 때 사용할 초기 힙(heap) 메모리를 128m(메가바이트)로 설정합니다.
-Xmx 에 설정한 값보다 작은 값을 설정하여야 하며 지정한 메모리가 많을 수록 초기 수행속도가 빨라집니다.

-Xmx256m 옵션

eXERD가 최대로 사용할 수 있는 힙(heap) 메모리를 256m(메가바이트)로 설정합니다.
최대값을 많이 줄수록 대규모 작업 시에도 무리없이 동작합니다.

※ 최대값은 사용자의 시스템에 설치된 메모리에 따라 설정할 수 있는 한도가 다릅니다.

Q7 SQLSERVER 윈도우인증으로 접속이 안됩니다.

A7. 통합 인증 설정이 안되어 있거나 관리자 권한으로 실행되어 있지 않습니다.

JDBC 드라이버는 integratedSecurity 연결 문자열 속성을 통해 Windows 운영 체제에서 형식 2 통합 인증을 사용할 수 있습니다. 통합 인증을 사용하려면 sqljdbc_auth.dll 파일을 JDBC 드라이버가 설치된 컴퓨터의 Windows 시스템 경로에 있는 디렉터리에 복사합니다.

sqljdbc_auth.dll 파일은 다음 위치에 설치되어 있습니다.

(installation directory)₩sqljdbc_(version)₩(language)₩auth₩

참고: 32비트 JVM(Java Virtual Machine)을 실행할 경우 운영 체제가 x64 버전이라도 x86 폴더에 있는 sqljdbc_auth.dll 파일을 사용하십시오. x64 프로세서에서 64비트 JVM을 실행할 경우 x64 폴더의 sqljdbc_auth.dll 파일을 사용하십시오. Itanium 프로세서에서 64비트 JVM을 실행할 경우 IA64 폴더의 sqljdbc_auth.dll 파일을 사용하십시오.

또는 java.libary.path 시스템 속성에 sqljdbc_auth.dll의 디렉터리를 지정할 수도 있습니다. 예를 들어 JDBC 드라이버가 기본 디렉터리에 설치된 경우 Java 응용 프로그램이 시작될 때 다음과 같은 가상 컴퓨터(VM) 인수를 사용하여 DLL의 위치를 지정할 수 있습니다.

-Djava.library.path=C:₩Microsoft SQL Server JDBC Driver₩sqljdbc_(version)₩enu₩auth₩x86

참고: JDBC 드라이버가 Windows 이외의 운영 체제에서 실행되는 경우에는 통합 인증을 사용할 수 없습니다. Windows 이외의 운영 체제에서 SQL Server로 연결하는 경우 이 드라이버는 사용자 이름 및 암호 등 Windows 인증 자격 증명을 제공하지 않습니다. 이러한 경우 응용 프로그램에서는 SQL Server 인증을 사용해야 합니다.

참조 : 마이크로 소프트 도움말 - 통합 인증으로 연결

7. XScript API

■ UI관련 API

chooseTable()

테이블 선택 다이얼로그를 엽니다.

반환값 : 모델 컨텍스트

사용자가 선택한 테이블에 대한 컨텍스트. 테이블 선택 다이얼로그를 엽니다.

chooseTable 예제

```
var table = chooseTable();
if (table.size() == 1) {
        log("선택한 테이블은 " + table.get("logical-name") + "입니다.");
}
```

promptOpenFile(title, extension)

파일을 선택하는 다이얼로그를 표시합니다.

title : String (생략 가능, 기본값: 파일 열기)

다이얼로그 타이틀.

extension : String (생략 가능, 기본값: txt)

선택할 파일의 확장자.

반환값 : 파일 객체

선택된 {@link XFile}.

promptQuestion(message)

true 또는 false를 선택할 수 있는 다이얼로그를 엽니다.

message : String (생략 가능, 기본값: 계속 진행하시겠습니까?)

표시할 메시지.

반환값 : boolean

true 또는 false를 선택할 수 있는 다이얼로그를 엽니다.

promptQuestion 예제

```
if (!promptQuestion("계속 진행 하시겠습니까?")) {
        exit();
}
```

promptSaveFile(title, extension)

저장할 목적으로 파일을 선택하는 다이얼로그를 엽니다.

title : String (생략 가능, 기본값: 파일 저장)

다이얼로그 타이틀.

extension : String (생략 가능, 기본값: txt)

파일의 확장자.

반환값 : 파일 객체

선택된 XFile 객체.

promptText(message, initial)

텍스트를 입력받는 UI를 표시합니다.

message : String : String (생략 가능, 기본값: 텍스트를 입력하세요.)

화면에 표시할 메시지.

initial : String (생략 가능, 기본값: "" (Empty String))

초기 입력 값.

반환값 : String

사용자가 입력한 텍스트 값.

promptText 예제

```
var name = promptText("성함을 입력하세요", "홍길동");
log("안녕하세요, " + name + "님");
```

sleep(millis)

주어진 시간 만큼 대기합니다.

millis : long (생략 가능, 기본값: 1000)

대기할 시간, 단위 밀리초.

주어진 시간 만큼 대기합니다.

sleep 예제

```
monitor.beginTask("테스트 작업", 100);
for (var i = 0; i < 100; i++) {
        sleep(100);
        monitor.worked(1);
}
monitor.done();
```

■ 콘솔

콘솔 객체는 화면에 텍스트를 출력하는 일들에 대한 전역 API를 공급합니다. 아래에 나열된 API 들은 정적인 함수 형태로 사용하실 수 있습니다.

log(msg)

한 문장을 콘솔에 인쇄합니다.

msg : Object

인쇄할 문장.

log 예제
log("hello world!");

logf(format, args)

양식에 맞추어 정보를 콘솔에 출력합니다.

format : String

출력할 데이터 포맷. 자세한 포멧 양식에 대해서는 자바 문서를 참조 하십시오.

args : java.lang.Object[]

출력할 데이터.

양식에 맞추어 정보를 콘솔에 출력합니다.

첫번째 인자인 format에서 다음과 같은 변환 인자들을 사용할 수 있습니다:

- b, B : 주어진 boolean 타입을 true 또는 false로 출력합니다.
- s, S : 주어진 String 타입을 출력합니다.
- d : 주어진 정수를 10진수 형태로 표기합니다.
- o : 주어진 정수를 8진수 형태로 표기합니다.
- x, X : 주어진 정수를 16진수 형태로 표기합니다.

각 변환 인자는 길이및 정렬 옵션 가질 수 있으며 다음의 형태로 사용합니다:
%[-][0][소수점 이상 길이.소수점 이하 길이]변환인자
길이 앞에 - 부호를 붙이면, 주어진 길이내에 왼쪽으로 정렬됩니다. 길이가 0으로 시작하면 남는 길이를 0으로 채우게 됩니다.

logf 예제

```
select(function(it){
        return it.get("type") == "table";
}).each(function(it){
        var columns = it.select(function(it){
                return it.get("type") == "column";
        });
        logf("%s 테이블에서는 %df개의 컬럼이 있습니다.", it.get("logical-name"), columns.size());
});
```

setLogFile(file)

로그 파일을 지정합니다.
file : 파일 객체
지정할 파일.
로그 파일이 지정되면 log() 또는 logf()등으로 콘솔에 출력되는 모든 메시지가 해당 파일로도 출력 됩니다. 이 파일에 대한 다른 스트림이 열려있지 않아야 하며, 파일이 이미 존재하는 경우, 내용이 뒤 부분에 추가됩니다.

setLogFile(file, replace)

로그 파일을 지정합니다.
log() 또는 logf()등으로 콘솔에 출력되는 메시지를 저장할 파일을 지정합니다.
file : 파일 객체
저장할 파일
replace : boolean
이 값이 true이면, 기존 파일을 덮어쓰고, 그렇지 않으면 뒤 부분에 내용을 추가합니다.
이 값이 true로 지정하면, 이전 파일을 덮어 쓰며 경고하지 않습니다.
로그 파일을 지정합니다.
log() 또는 logf()등으로 콘솔에 출력되는 메시지를 저장할 파일을 지정합니다.

■ Form

openDialog(json, title, message)

사용자 다이얼로그를 엽니다.

json : NativeObject

사용자 다이얼로그 정의 JSON.

title : String (생략 가능, 기본값: 사용자 폼) (Ignorable, Default: User Form)

다이얼로그 타이틀.

message : String (생략 가능, 기본값: 스크립트 수행에 필요한 정보를 입력하세요.)

다이얼로그 메시지.

반환값 : Object

JSON

형태의 다이얼로그에 입력된 값.

사용자 다이얼로그를 엽니다. 다이얼로그는 JSON 객체를 통해 정의 합니다. 기본 규칙은 다음과 같습니다.

openDialog 예제 1

```
{
        필드명 : "화면에 보일 레이블",
        필드명2 : "화면에 보일 레이블2"
        ...
}
```

위와 같은 경우, 모든 필드의 타입은 텍스트가 됩니다. 다른 타입을 이용하고 싶은 경우, 레이블을 입력하는 곳에 다음 문법을 사용합니다.

openDialog 예제 2

```
{
        label : "화면에 보일 레이블",
        type : "text|boolean|table"
}
```

아래의 예제를 참고 하십시오.

openDialog 예제 3

```
var result = openDialog({
        name : "이름",
        age : "나이"
});
logf("입력한 이름은 %s 입니다.₩n", result.name);
```

openDialog 예제 4

```
var config = openDialog({
        table : {
                label : "테이블",
                type : "table"
        },
        search : "찾을 이름",
        replace : "치환할 이름",
        suppressDuplication : {
                type : "boolean",
                label : "중복 무시"
        }
});
log("%s 테이블에서 %s 이름을 가진 컬럼을 %s로 변경합니다.₩n",
        config.table.get("name"),
        config.search,
        config.replace);
```

■ **Views**

showDomainView()

도메인 뷰를 엽니다.

showModelView()

모델 뷰를 엽니다.

흐름제어 API

exit()

스크립트를 종료합니다.

스크립트를 종료합니다. 이후 문맥이 남아 있어도 강제로 종료됩니다.

■ 유틸리티

보편적으로 사용되는 자료구조등과 같은 API들을 제공합니다.

compilePattern(regexp)

정규 표현식을 컴파일 하고, 자바 패턴 객체를 얻습니다.

regexp : String

정규 표현식.

반환값 : Pattern

패턴 객체.

compilePattern 예제

```
var input = "안녕하세요, 올해는 2012년입니다.";

var matcher = compilePattern("[0-9]+").matcher(input);
while (matcher.find()) {
        log(matcher.group());
}
```

compilePattern 결과

```
2012
```

format(format, objects)

양식에 맞추어 문자열을 만듭니다.

format : String

출력할 데이터 포맷. 자세한 포멧 양식에 대해서는 자바 문서를 참조 하십시오.

objects : java.lang.Object[]

출력할 데이터.

반환값 : String

양식화된 문자열.

format 예제

```
select(function(it){
        return it.get("type") == "table";
}).each(function(it){
        var columns = it.select(function(it){
                return it.get("type") == "column";
        });
        var data = format("%s 테이블에서는 %d개의 컬럼이 있습니다.", it.get("logical-name"),
columns.size());
        log(data);
});
```

format 결과

```
XXXX 테이블에는 10개의 컬럼이 있습니다.
```

newDate()

현재 시간을 담은 자바 Date 객체를 얻습니다.

반환값 : Date

현재 시간을 담은 새 자바 java.util.Date 객체.

newList()

새로 리스트를 하나 만듭니다.

반환값 : XList

java.util.List

newMap()

새로운 맵을 하나 만듭니다.

반환값 : Map

새로 만들어진 java.util.Map.

newSet()

새로운 집합을 하나 만듭니다.

반환값 : Set

새 자바 java.util.Set.

newStack()

새로운 스택을 하나 만듭니다.

반환값 : Stack

새 자바 java.util.Stack.

■ 모델 관련.

RootContext

루트 컨텍스트는 최초 원하는 eXERD 모델들을 선택할 수 있도록 해주며, 도메인을 새로 만드는 기능을 제공합니다.

newDomain(name)

새 도메인을 만듭니다.

name : String

도메인 이름.

반환값 : 모델 컨텍스트

새 도메인을 담은 모델 컨텍스트.

newDomain(parentDomainCtx, name)

새로운 도메인을 만듭니다.

parentDomainCtx : 모델 컨텍스트

부모 도메인을 담은 컨텍스트.

name : String

새 도메인의 이름.

반환값 : 모델 컨텍스트

select(rule)

주어진 클로져를 만족하는 모델 컨텍스트를 얻습니다.

rule : 클로져(Closure)

선택 표현식. true 또는 false를 리턴해야 합니다. 이 클로져는 일반적으로 모델컨텍스트의 API를 이용하여, 선택하고자 하는 모델의 속성을 검사합니다.

반환값 : 모델 컨텍스트

조건을 만족하는 모델을 담은 컨텍스트.

select 예제

```
select(function(it){
        return it.get("type") == "column";
});
```

■ 모델 컨텍스트

모델 컨텍스트에 대한 API 입니다. 테스트.

each(visitor)

클로져 수행.

visitor : 클로져(Closure)

동작을 담은 클로져.

반환값 : 모델 컨텍스트

현재 모델 컨텍스트.

클로져 수행. 현재 컨텍스트에 속한 모델들이 각각 주어진 특정 동작을 수행하게 합니다.

each 예제

```
selection.each(function(it){
        log(it.get("name"));
});
```

filter(rule)

현재 모델 컨텍스트 내에서 주어진 조건을 만족하는 모델만 남깁니다.

rule : 클로져(Closure)

남길 모델을 평가하는 클로져. 이 클로져는 모델을 인자로 하여, true 또는 false를 리턴해야 합니다.

반환값 : 모델 컨텍스트

평가식을 통과한 모델들만 담은 모델 컨텍스트.

get(attrName)

속성을 얻습니다. 만약 이 모델 컨텍스트가 여러 모델을 담고 있는 경우, 첫 번째 모델의 속성을 얻습니다.

attrName : String

얻을 속성의 이름.

반환값 : Object

속성 값.

get 예제

```
// 논리명 얻기
var logicalName = context.get("logical-name");
```

getUserProperty(key)

사용자 정의 속성을 얻습니다.

key : String

사용자 정의 속성 키.

반환값 : String

주어진 속성 키에 대한 값.

setUserProperty()를 통해 사용자 정의 속성을 지정할 수 있습니다.

has(attrName)

현재 컨텍스트의 첫번째 모델이 특정 속성을 가지고 있는지 여부를 반환합니다.

attrName : String

속성.

반환값 : boolean

속성의 존재 여부.

현재 컨텍스트가 비어있는 경우 항상 false가 리턴됩니다. 현재 컨텍스트가 여러 모델을 갖고 있는 경우 첫 번째 모델의 속성 존재 여부를 리턴합니다.

has 예제

```
// 물리명을 소유한 모든 모델에 대한 컨텍스트 얻기
select(function(it){
        return it.has("physical-name");
});
```

head()

현재 모델 컨텍스트에 포함된 첫 번째 모델만을 담은 컨텍스트를 얻습니다.

반환값 : 모델 컨텍스트

첫번째 모델이 담긴 모델 컨텍스트.

isEmpty()

이 모델 컨텍스트가 비어있는지 여부를 얻습니다.

반환값 : boolean

모델 컨텍스트가 비어 있는 경우 true, 비어있지 않은 경우 true.

join(separator, transformer)

현재 모델 컨텍스트에 포함된 모델들을 스트링 형태로 변경하고 연결합니다.

separator : String

각 모델 사이의 구분자.

transformer : 클로져(Closure)

각 모델을 문자열로 변환하는 변환 함수.

Conversion function that convert to string each model.

반환값 : String

각 모델이 주어진 변환 함수로 변환된 뒤 연결된 문자열.

현재 모델 컨텍스트에 포함된 모델들을 스트링 형태로 변경하고 연결합니다.

예를들어 Foo와 Bar라는 두개의 테이블이 담긴 모델컨텍스트가 있다고 할 때, "Foo, Bar"라는 문자열을 얻으려면 다음과 같이 할 수 있습니다.

join 예제

```
var ctx = ...;

var result = ctx.join(", ", function(it){
        return it.get("name");
});
log(result);
```

join(prefix, suffix, separator, transformer)

현재 모델 컨텍스트에 포함된 모델들을 스트링 형태로 변경하고 연결합니다.

prefix : String

결과물의 앞에 붙을 문자열.

suffix : String

결과물의 뒤에 붙을 문자열.

String to attach at the rear of the result

separator : String

각 모델 사이의 구분자.

transformer : 클로져(Closure)

각 모델이 주어진 변환 함수로 변환된 뒤 연결된 문자열.

반환값 : String

현재 모델 컨텍스트에 포함된 모델들을 스트링 형태로 변경하고 연결합니다.

예를들어 Foo와 Bar라는 두개의 테이블이 담긴 모델컨텍스트가 있다고 할 때, "[Foo, Bar]"라는 문자열을 얻으려면 다음과 같이 할 수 있습니다.

join 예제

```
var ctx = ...;
var result = ctx.join("[", "]", ", ", function(it){
        return it.get("name");
});
log(result);
```

last()

현재 모델 컨텍스에 담긴 마지막 모델만을 담은 컨텍스트를 얻습니다.

반환값 : 모델 컨텍스트

map(mapper)

현재 모델 컨텍스트에 담긴 각각의 모델을 다른 형태로 변환합니다.

mapper : 클로져(Closure)

변환 함수. 이 변환 함수는 모델을 인자로 하여 변형된 값을 리턴하는 형태이어야 합니다.

반환값 : Object

변환된 모델 컨텍스트 또는 자바 스크립트 배열.

map 예제

```
// 모델들을 논리명 배열로 변경하는 예제
var nameArray = selection.map(function(it){
        return it.get("logical-name");
});
for(i in nameArray){
        log(nameArray[i]);
}
```

mapAndPack(mapper)

map()과 유사하지만, 변형 후 중복되는 요소가 제거 됩니다.

mapper : 클로져(Closure)

변환 함수. 이 변환 함수는 모델을 인자로 하여 변형된 값을 리턴하는 형태이어야 합니다.

반환값 : Object

변환된 모델 컨텍스트 또는 자바 스크립트 배열.

pack()

컨텍스트 내에 중복되는 요소를 제거합니다.

반환값 : 모델 컨텍스트

중복된 요소가 제거된 모델 컨텍스트.

select(rule)

현재 컨텍스트에 선택된 모델 및, 하위 모델들을 대상으로 검색을 수행합니다.

rule : 클로져(Closure)

검색 조건을 담은 클로져.

반환값 : 모델 컨텍스트

주어진 검색 조건을 만족하는 모델이 담긴 모델 컨텍스트.

현재 컨텍스트에 선택된 모델 및, 하위 모델들을 대상으로 검색을 수행합니다.

select 예제

```
var tables = select(function(it){
        return it.get("type") == "table";
}).each(function(it){
        // 각 테이블 내의 모든 컬럼을 다시 검색
        var columns = it.select(function(it){
                return it.get("type") == "columns";
        });
});
```

selectInEditor()

현재 컨텍스트에 포함된 요소를 다이어그램 에디터에서 선택합니다.

현재 컨텍스트에 포함된 요소를 다이어그램 에디터에서 선택합니다. 만약 현재 선택된 모델이 활성 다이어그램 내에 존재 하지 않는 경우 선택되지 않습니다.

set(json)

JSON을 이용해 복수의 속성을 한번에 지정합니다.

json : NativeObject

속성 이름/값을 담은 JSON 객체.

반환값 : 모델 컨텍스트

현재 모델 컨텍스트.

set 예제

```
var template = {
        "datatype" : "varchar(10)",
        "color-tag" : "red"
};
select(function(it){
        return it.get("type") == "column" && it.get("logical-name").endsWith("이름");
}).each(function(it){
        it.set(template);
});
```

set(attrName, value)

속성을 지정합니다.

attrName : String

지정할 속성 이름.

value : Object

속성 값.

반환값 : 모델 컨텍스트

현재 모델 컨텍스트.

setUserProperty(key, value)

사용자 정의 속성을 지정합니다.

key : String

지정할 속성 키.

value : String

값. 값은 String 으로만 저장됩니다. 값이 null이면 속성을 지웁니다.

반환값 : 모델 컨텍스트

현재 모델 컨텍스트.

getUserProperty()를 통해 사용자 정의 속성을 지정할 수 있습니다.

size()

이 컨텍스트에 속한 모델의 갯수를 얻습니다.

반환값 : int

컨텍스트에 포함된 모델의 갯수.

■ 타입 목록

column (Column)

column-reference (Column Reference)

diagram (Diagram)

domain (Domain)

fk-index (Foreign Key Index)

foreign-key (Foreign Key)

function (Function)

non-unique-index (Non-Unique Index)

pk-constraint (Primary Key Constraint)

primary-index (Primary Key Index)

procedure (Procedure)

schema (Schema)

table (Table)

table-diagram (Table Diagram)

tag (Tag)

term (Term)

trigger (Trigger)

unique-index (Unique Index)

view (View)

■ 속성 목록

child-column
child-table
comment
data-type
default-value
diagram
domain
fk-index
foreign-key
function
is-foreign-key
is-indexed
is-non-unique-indexed
is-primary-key
is-unique-indexed
logical-description
logical-name
logical-synonym
name
not-null
parent-column
parent-table
physical-description
physical-name
physical-synonym
procedure
schema
table
table-diagram
tag
tag-color
theme
trigger
type
view

■ 파일 관련 API

파일 관련 API들을 제공합니다.

newFile(path)

XFile 객체를 생성합니다.

path : String

파일의 경로 및 이름. / 으로 시작하는 경우 절대 주소이며, 그렇지 않은경우 스크립트가 수행되는 eXERD 파일의 경로에 대해 상대 경로로 인식됩니다.

반환값 : 파일 객체

XFile객체.

■ 파일 객체

exist()

해당 파일이 존재하는지 여부를 반환합니다.

반환값 : boolean

파일의 존재 유무.

getAppendStream()

이 파일에 텍스트를 추가하여 쓸 수 있는 프린트 스트림을 얻습니다.

반환값 : 파일 쓰기 PrintStream

파일의 내용을 쓸 수 있는 프린트 스트림.

getName()

파일의 이름을 얻습니다.

반환값 : String

파일의 이름.

getOutputStream()

이 파일에 텍스트를 쓸 수 있는 프린트 스트림을 얻습니다.

반환값 : 파일 쓰기 PrintStream

파일의 내용을 쓸 수 있는 프린트 스트림.

getPath()

파일의 전체 경로를 얻습니다.

반환값 : String

파일의 전체 경로를 얻습니다. 경로 값은 워크스페이스를 기준으로 합니다.

openInEditor()

이 파일을 이클립스 에디터로 엽니다.

■ 파일 쓰기 PrintStream

close()

쓰기 스트림을 닫습니다.

쓰기 스트림을 닫습니다. 쓰기 스트림은 사용후 일반적으로 닫아 주어야 합니다. 스크립트가 종료될 때 까지 닫기지 않은 경우, 또는 이 스트림의 파일이 openInEditor() 등으로 파일을 열려고 할 경우 자동으로 닫기게 됩니다.

getFile()

이 스트림이 컨텐츠를 쓰고 있는 XFile 객체를 얻습니다.

반환값 : 파일 객체

XFile 객체.

isClosed()

반환값 : boolean

스트림이 닫혔는지 여부를 반환 합니다.

printf(format, args)

양식에 맞추어 정보를 출력합니다.

format : String

출력할 데이터 포맷. 자세한 포멧 양식에 대해서는 자바 문서를 참조 하십시오.

args : java.lang.Object[]

출력할 데이터.

양식에 맞추어 정보를 출력합니다.

첫번째 인자인 format에서 다음과 같은 변환 인자들을 사용할 수 있습니다:

- b, B : 주어진 boolean 타입을 true 또는 false로 출력합니다.
- s, S : 주어진 String 타입을 출력합니다.
- d : 주어진 정수를 10진수 형태로 표기합니다.
- o : 주어진 정수를 8진수 형태로 표기합니다.
- x, X : 주어진 정수를 16진수 형태로 표기합니다.

각 변환 인자는 길이및 정렬 옵션 가질 수 있으며 다음의 형태로 사용합니다:

%[-][0][소수점 이상 길이.소수점 이하 길이]변환인자

길이 앞에 - 부호를 붙이면, 주어진 길이내에 왼쪽으로 정렬됩니다. 길이가 0으로 시작하면 남는 길이를 0으로 채우게 됩니다.

printf 예제

```
var out = newFile("log.txt").getOutputStream();
select(function(it){
        return it.get("type") == "table";
}).each(function(it){
        var columns = it.select(function(it){
                return it.get("type") == "column";
        });
        out.printf("%s 테이블에서는 %d개의 컬럼이 있습니다.", it.get("logical-name"),
columns.size());
});
```

println(o)

스트림을 통해 한 문장을 출력합니다.

o : Object

출력할 문장.

반환값 : 파일 쓰기 PrintStream

이 스트림.

■ XPluginRegistry

plugins 객체를 이용하여 XScript의 기능을 추가할 수 있습니다. 현재 버전에서는 모델 컨텍스트에서 get으로 속성을 얻을때, 확장 속성을 추가할 수 있습니다.

addGetter(name, nativeFunction)

모델 컨텍스트의 get 메서드를 확장합니다.

name : String

확장할 속성.

nativeFunction : NativeFunction

확장된 속성을 얻는 함수.

반환값 : XPluginRegistry

플러그인 객체.

모델 컨텍스트의 get 메서드를 확장합니다.

addGetter 예제

```
// name-type 이라는 속성을 정의
plugins.addGetter("name-type", function(it){
        return format("%s : %s", it.get("logical-name"), it.get("data-type"));
});
select(function(it){
        return it.get("type") == "column");
}).each(function(it){
        // 사용자가 정의한 속성을 화면에 출력.
        log(it.get("name-type"));
});
```

addSetter(name, nativeFunction)

모델 컨텍스트의 set 메서드를 확장합니다.

name : String

확장할 속성.

nativeFunction : NativeFunction

확장된 속성을 지정하는 함수.

반환값 : XPluginRegistry

Plugin Object.

모델 컨텍스트의 set 메서드를 확장합니다.

addSetter 예제

```
// both-name 이라는 속성을 정의, 논리명 물리명을 동일하게 한 번에 지정.
plugins.addSetter("both-name", function(ctx, value){
        ctx.set("logical-name", value).set("physical-name", value);
});
```