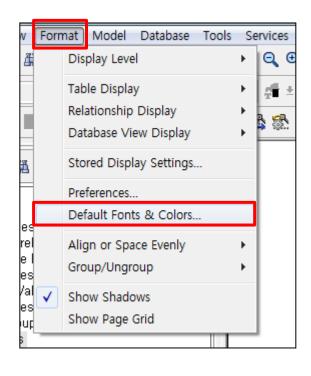
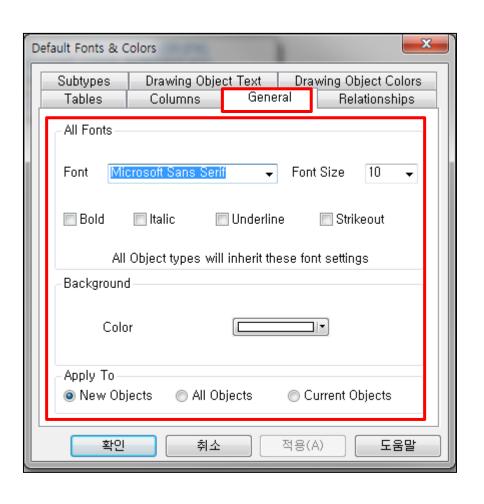


ERWIN 사용 실습

ERWIN 기본 설정 변경

• 폰트

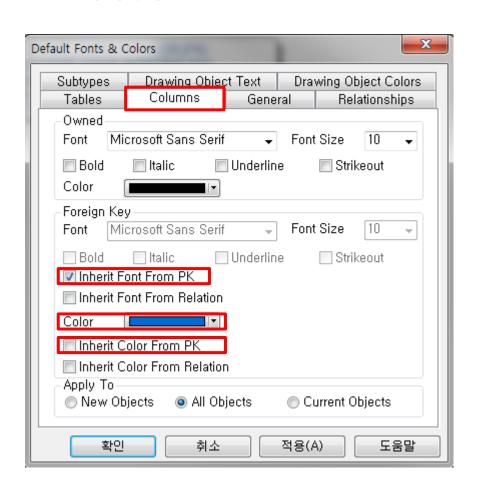






ERWIN 기본 설정 변경

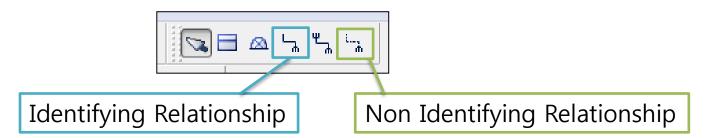
Relation



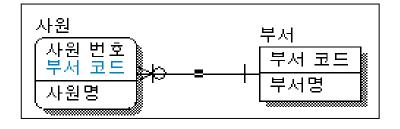




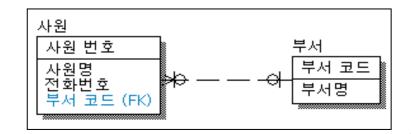
Relation 설정



- Identifying (식별)
 - 부모 테이블의 키가 자식 테이블의 키로 전이
 - FK가 PK의 일부
 - NULL 허용 안됨



- Non Identifying (비식별)
 - 부모 테이블의 키가 자식테이블의 일반 속성으로 전이





Modeling

- Logical
 - 논리적
 - Entity-Attribute-Tuple
 - 논리적인 데이터 구조
 - Business Logic 등에 의존
 - 일반적으로 한글로 하는 것이 이해하기 좋음
 - 추후에 변환 기능 사용

- Physical
 - 물리적
 - Table-Column-Record
 - 물리적 레벨 포함
 - 데이터 타입, 저장 구조 등
 - Target DB에 의존적
 - Target DB를 위해 주로 영문화 함



논리적 설계1. 엔티티 타입 도출

- 사용자의 요구사항 명세, 업무 프로세스, 관련된 양식 등을 분석하여 도출
- 관련된 방법
 - 요구사항, 업무 프로세스 등에서 명사 구분
 - 불분명한 것, 광범위 한 것들, 부적절한 것 제외
- 특징/체크리스트
 - 업무에서 관리할 필요가 있는 정보인가?
 - 식별자가 존재할 수 있는가?
 - 다수의 인스턴스가 존재 하는가?
 - 속성을 가지는가?
 - 다른 엔티티타입과 관계를 갖는가?
- 결과물: 엔티티 정의서



논리적 설계 2. 속성 식별

- 체크 리스트
 - 업무에 필요로 한 정보인가?
 - 의미상 더 이상 분리되지 않는가? (atomic)
 - 엔티티 인스턴스의 구성요소가 되는가?
- 도메인
 - 속성이 가질 수 있는 값의 범위
 - 데이터 타입, 크기, 제약사항...
 - 표준화가 필요
- 엔티티/속성의 이름
 - 표준 용어 사전을 통하여 표준화가 필요
 - 특히 영어, 약어 사용에 대한 표준 필요
 - 참고: 행정 표준 용어 사전 (2007)



논리적 설계 3. 관계 식별

- 관계 식별 방법
 - 업무에 필요한 관계 인가?
 - Cardinality 식별: 1:1, 1:N, M:N 가능
 - Optionality 식별: 필수 참여 여부 식별
 - 식별 관계 인가? 비 식별 관계 인가?

Q&A

- 셋 이상의 엔티티 사이의 관계는 가능한가?
- 하나의 엔티티 내부에서 관계가 가능한가?



논리적 설계 4. 식별자

- Primary Key: Unique, Minimal, Not NULL, 변경 불가
- 하나이상의 식별자가 있을 때 PK를 선택하는 법
 - 해당 업무에서 인스턴스의 검색 등을 위해 자주 사용되는 것
 - 너무 길거나 많은 복합 속성을 이용하지 않도록
- Surrogate Key (대리키/대체키)
 - 엔티티에 Key가 될 수 있는 속성(들)이 있음에도 특정한 필요에
 의해 가상으로 생성된 키를 사용
 - 해당 키는 실제 업무에서는 존재하지 않음
- 식별 vs. 비식별 관계
 - 자식 엔티티가 독립적으로 존재 가능한가?
 - SQL처리가 너무 복잡해지는가?
 - 과도하게 인조 식별자(Surrogate Key)를 사용하는가?



물리적 설계 1. ERD를 관계형 테이블로 전환

- Entity / Relationship 을 테이블 하나로 변환 가능
- 1:1이나 1:N Relationship의 경우 한쪽 entity에 합칠 수 있음
- Weak Entityset의 경우 부모의 PK를 가져와야 함
- M:N관계는 항상 별도의 테이블로 전환할 필요
- Generalization(상속관계)는 일반적으로 각각 별도의 테이블
 - 경우에 따라 Super 타입에 모두 합치거나 Super 타입을 없애고 Subtype들만을 테이블로 만들 수도 있음



물리적 설계 2. 반정규화

- 필요에 따라 정규형을 무시하면서 특정한 값을 중복시킬수 있음
- 잘 못 사용하면 시스템의 성능이나 무결성을 해칠 수 있으므로 주의해서 사용해야 함
- 과도한 Join이나 복잡한 계산 로직, 처리 비용 등을 피하 기 위하여 사용

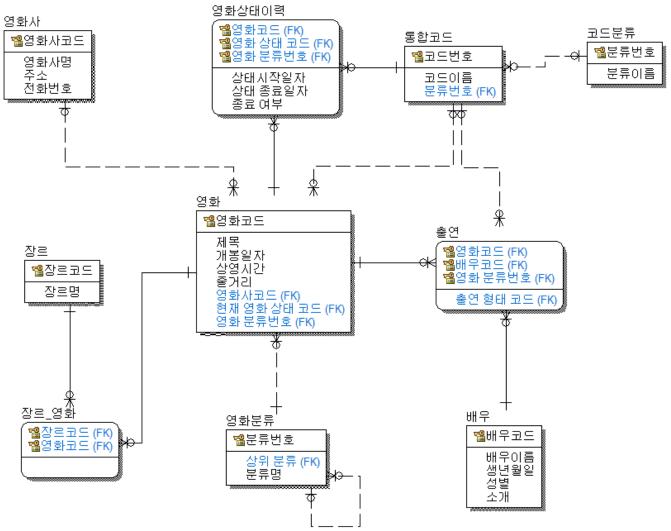


물리적 설계 3. 기타

- 인덱스: 질의에 자주 사용되는 컬럼에 인덱스 추가
 - 검색 성능 vs. 수정 성능의 trade off를 이해해야 함
- 뷰: 질의의 복잡성이나 성능상의 이유로 뷰 정의 가능
- 테이블 스페이스, 분산 데이터베이스 구성 등 결정



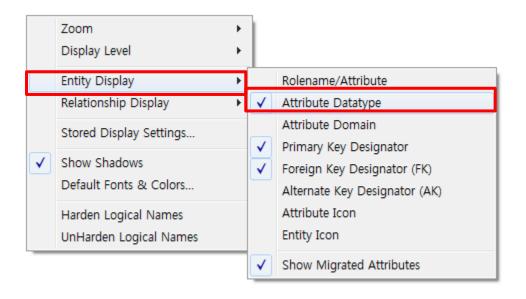
Logical Diagram





데이터 타입 정의

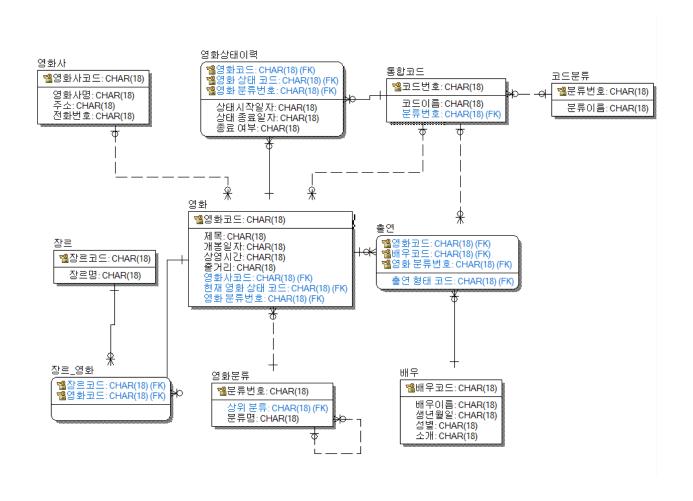
• 데이터 타입 표시





데이터 타입 정의

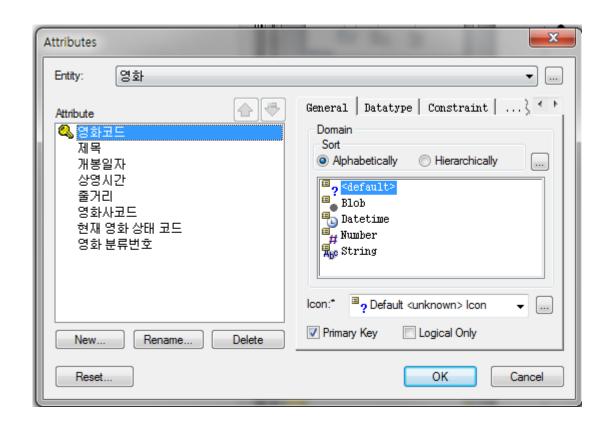
• 데이터 타입 설정 전 → 기본 데이터 타입 CHAR(18)





데이터 타입 설정

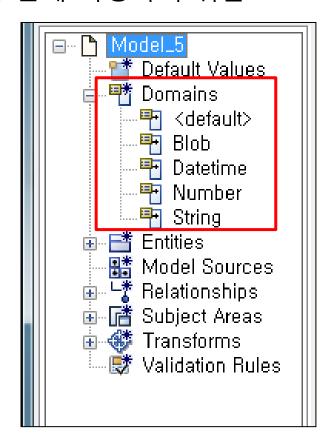
Entity 더블 클릭 하거나 마우스 우클릭 Attributes 선택
 → Attribute 설정, 이름 변경, 순서 변경 등





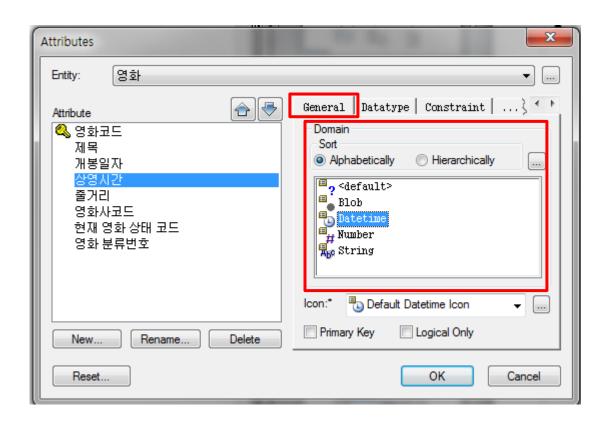
ERWIN Domain

- ERWIND Domain
 - 컬럼의 여러 속성을 쉽게 지정하기 위한 Predefined Set



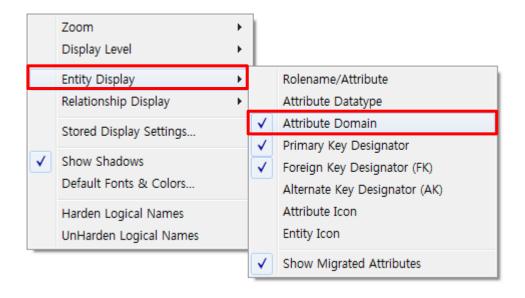


Domain 설정



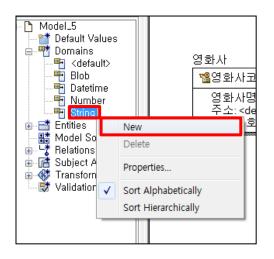


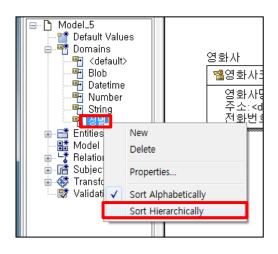
Domain 보기





Domain 생성

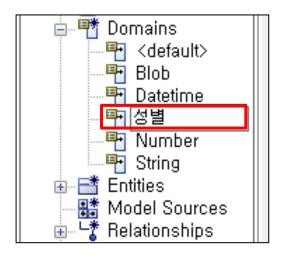


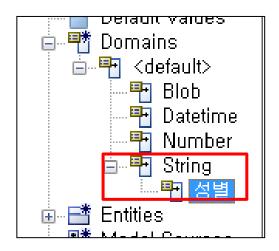






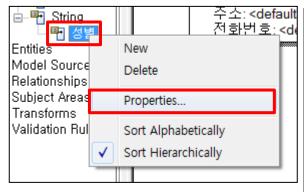


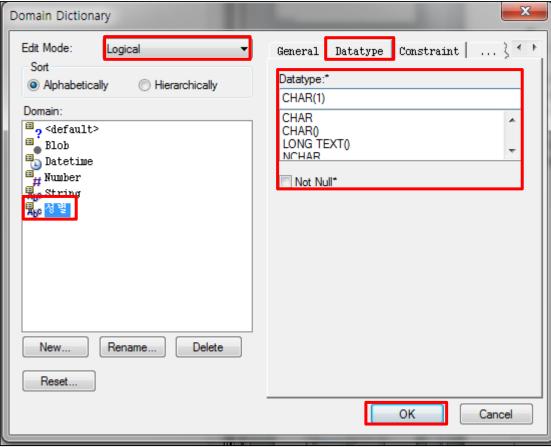




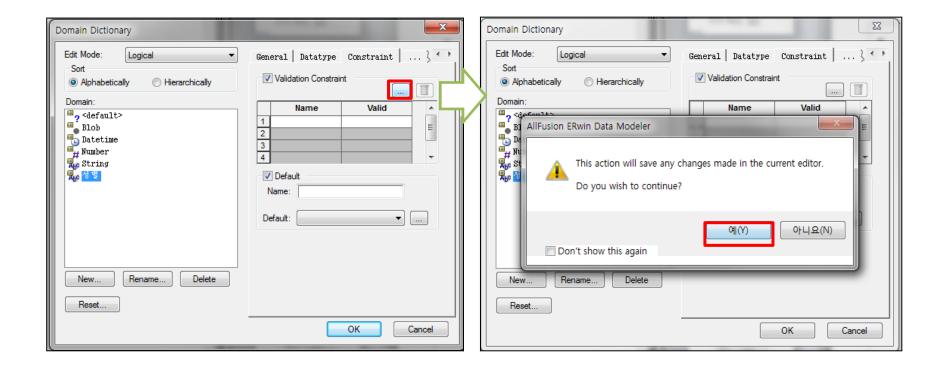


Domain 속성 변경



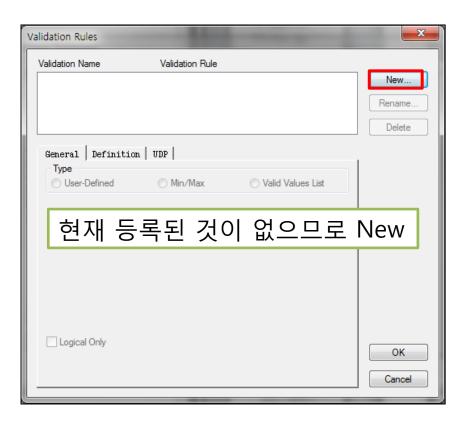


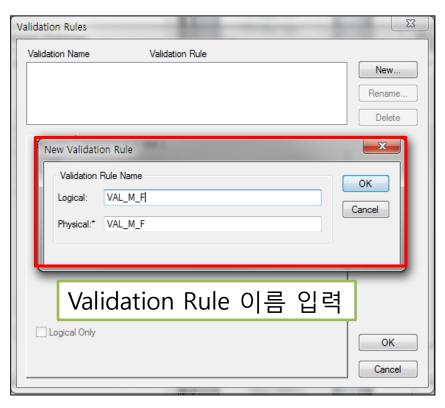






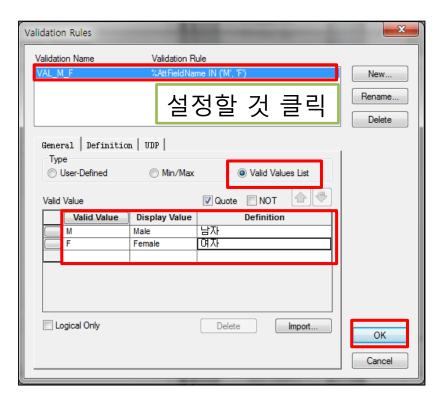
• Validation Rules 추가

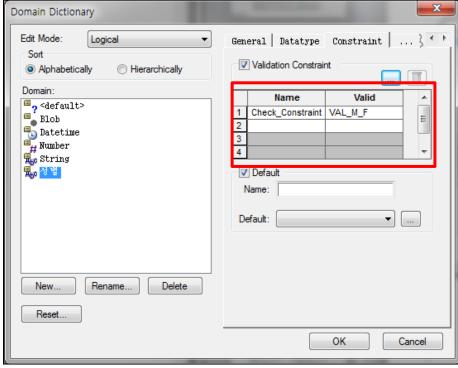






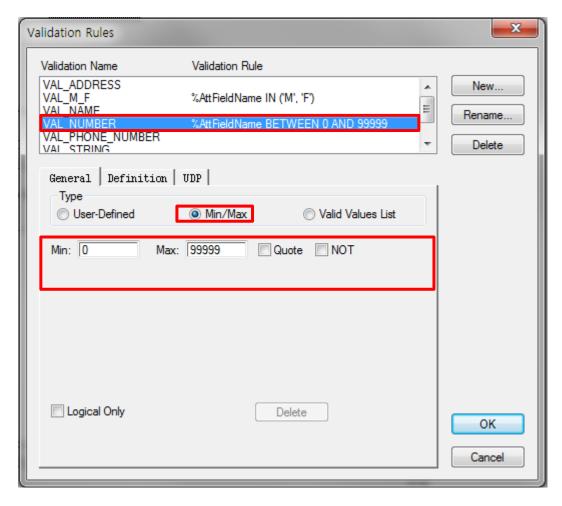
Valid Value 설정







Validation Rules Min/Max 설정



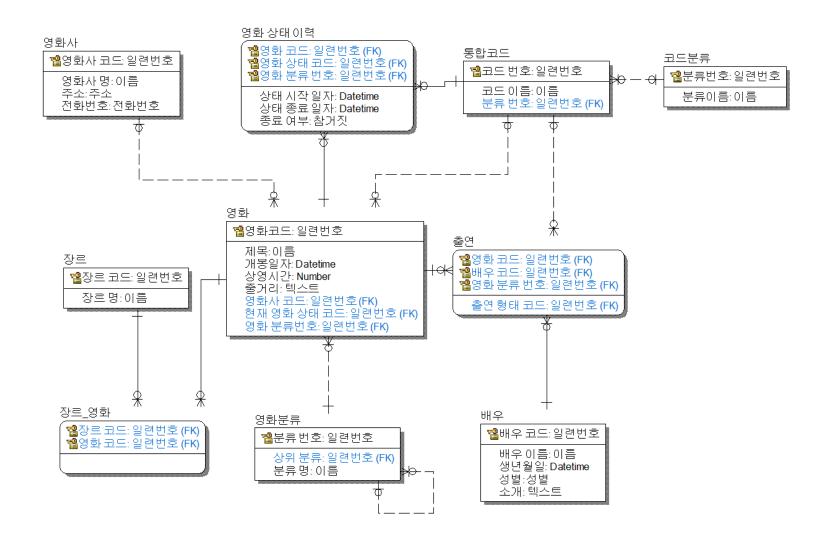


도메인 추가

	DomainParent	Datatype	Validation
일련번호	Number	NUMBER(10)	Min: 0 Max : 99999
이름	String	VARCHAR(20)	
전화번호	String	VARCHAR(30)	
참거짓	String	CHAR(1)	T: True F: False
텍스트	String	TEXT	
성별	String	CHAR(1)	F : Female M : Male
주소	String	VARCHAR(20)	

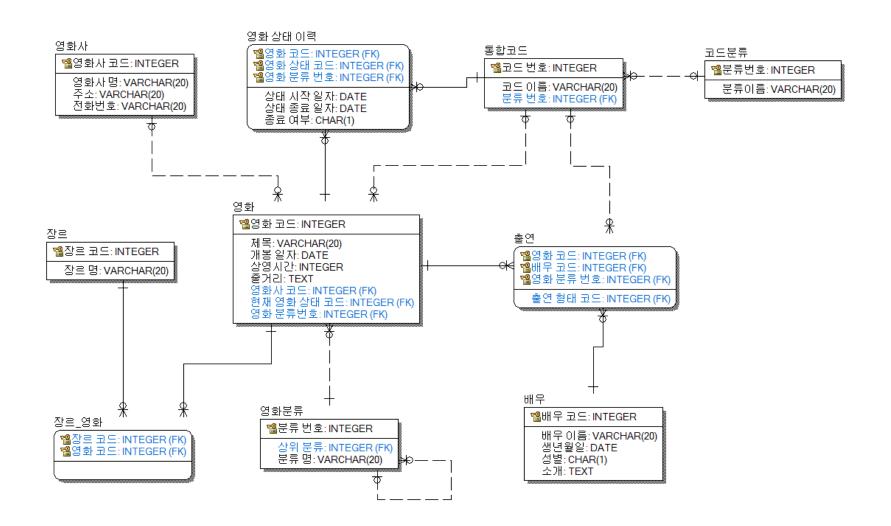


도메인 부여



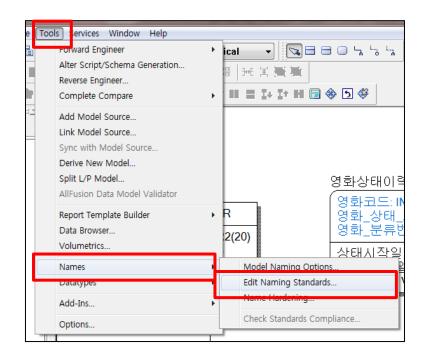


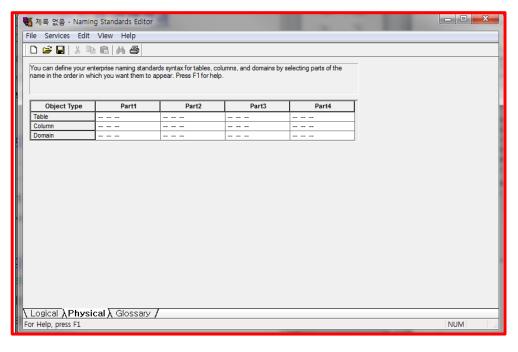
Logical → Physical





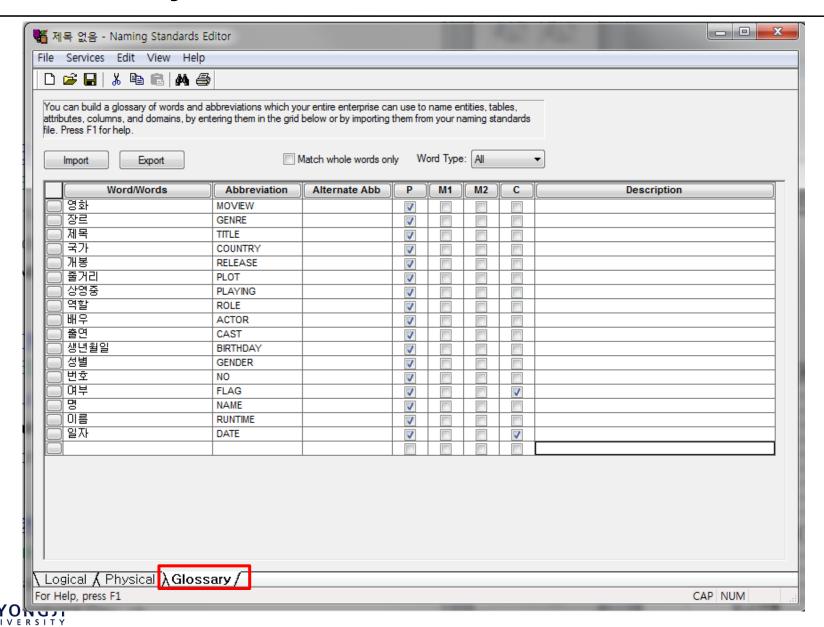
Dictionary 사용







Dictionary 사용

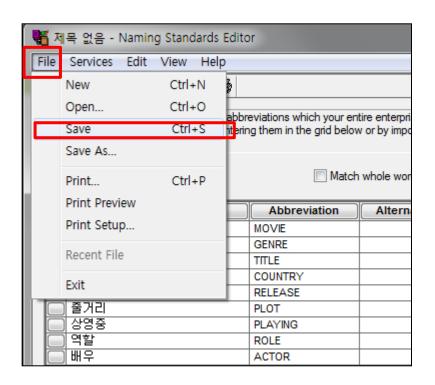


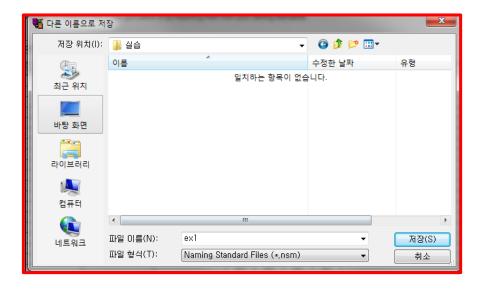
Dictionary 사용

• Dictionary 내용

	Word/Words	Abbreviation	Alternate Abbreviation	P	M1	M2	С
영화		MOVIE		V			
장르		GENRE		V			
제목		TITLE		V			
국가		COUNTRY		V			
개봉		RELEASE		V			
줄거리		PLOT		V			
상영중		PLAYING		V			
역할		ROLE		V			
배우		ACTOR		V			
출연		CAST		V			
생년월일		BIRTHDAY		V			
성별		GENDER		V			
변호		NO		V			
여부		FLAG		V			
명		NAME		V			7
이름		NAME		V			
상영시간		RUNTIME		V			
일자		DATE		V			7
코드		CODE		V			
분류		CATEGORY		V			
소개		INTRODUCTION		V			
주소		ADDRESS		V			
전화		PHONE		V			
상태		STATUS		V			
시작		START		V			
종료		END		V			
영화사		STUDIO		V			
이력		HISTORY		V			
· 통합		TOTAL		V			
현재		CURRENT		V			
<u>형태</u>		TYPE		V			
상위		HIGH_RANK		V			
WI I ON GOT							

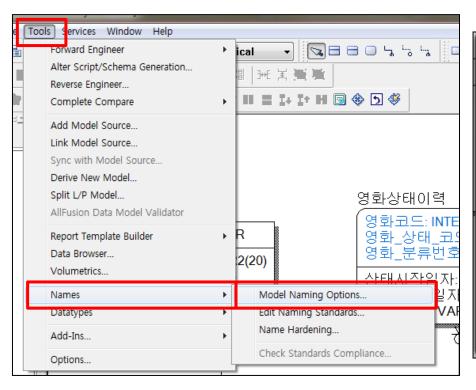
Dictionary 저장

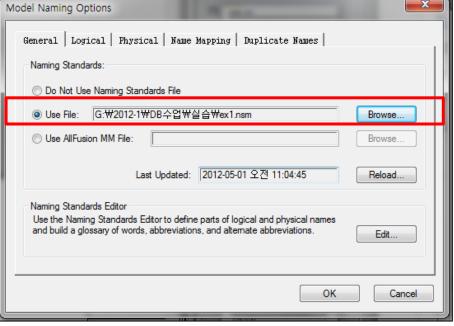






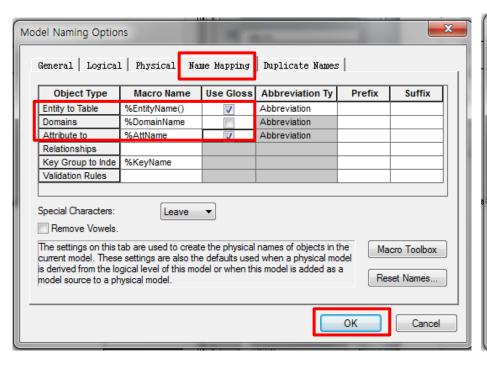
Dictionary 연결

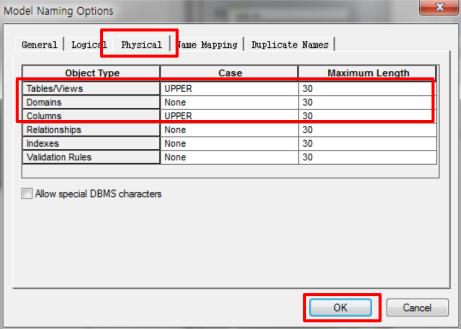






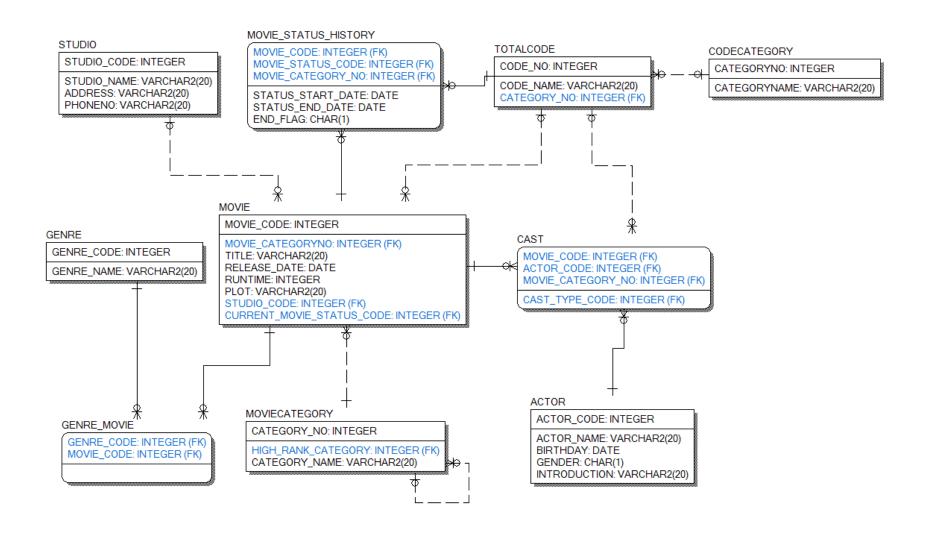
Dictionary 연결 (Cont.)





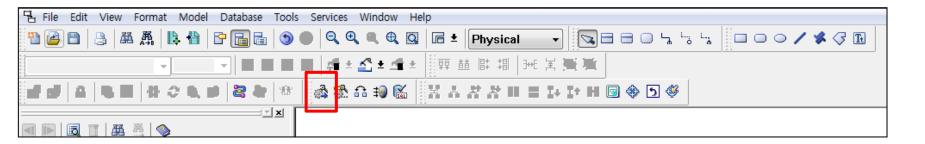


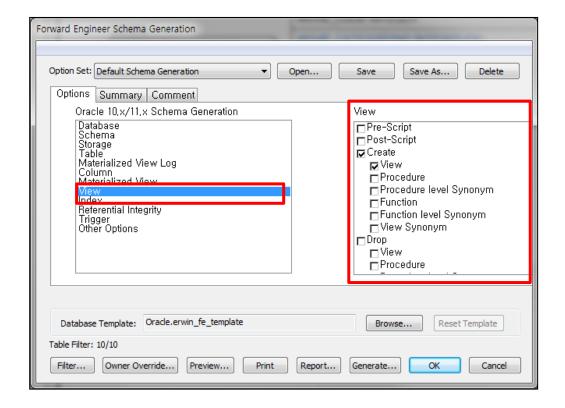
Physical Diagram



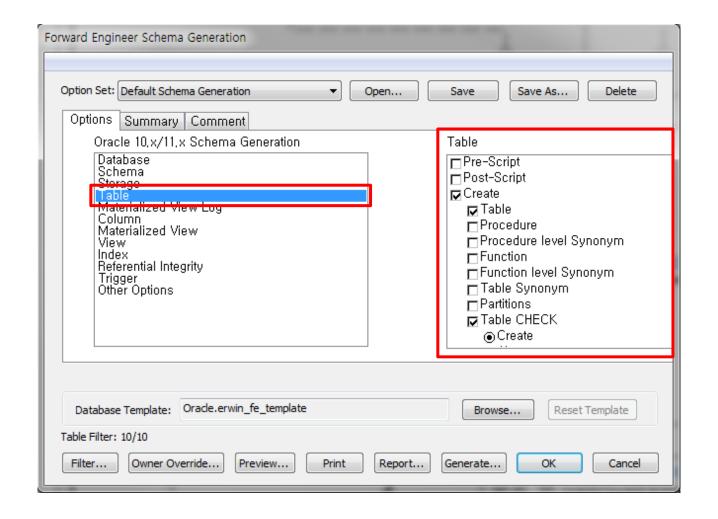


스키마 생성

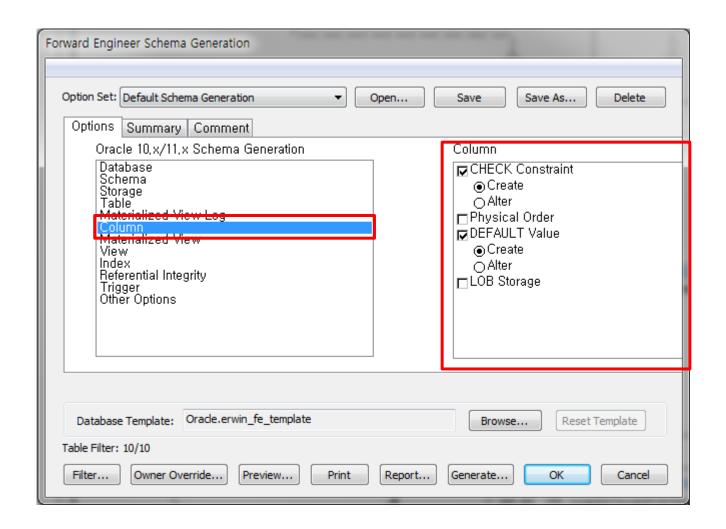




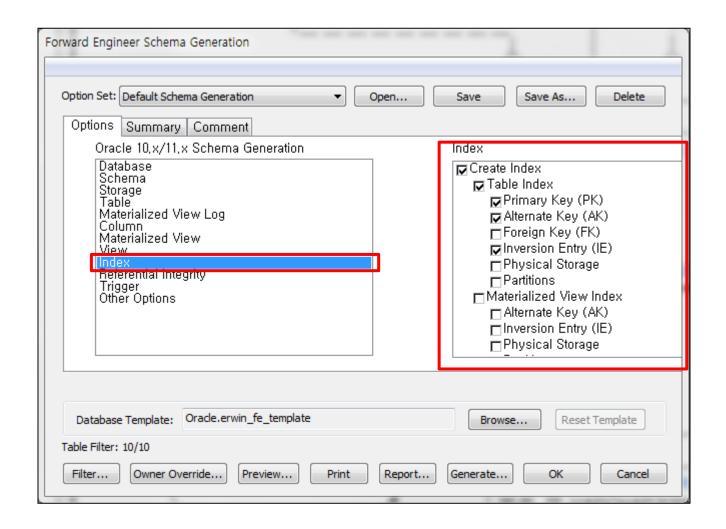




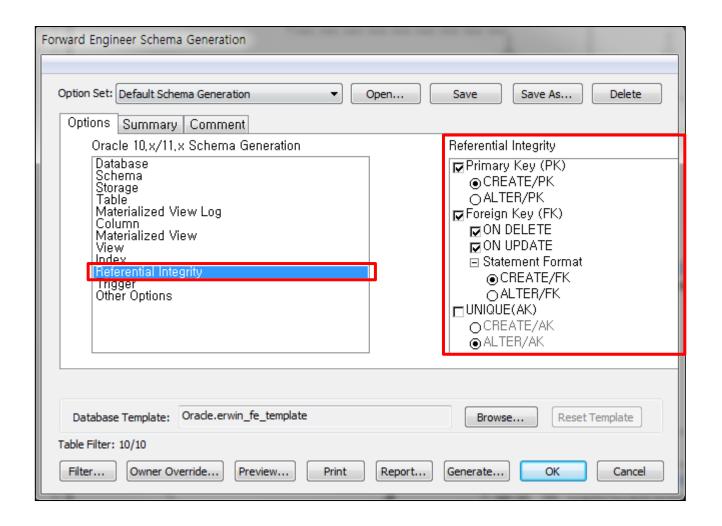




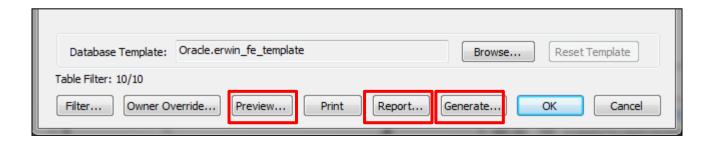












- Preview:
 - 미리보기
- Report (추천):
 - SQL 파일로 생성
- Generate:
 - 직접 접속 후 생성

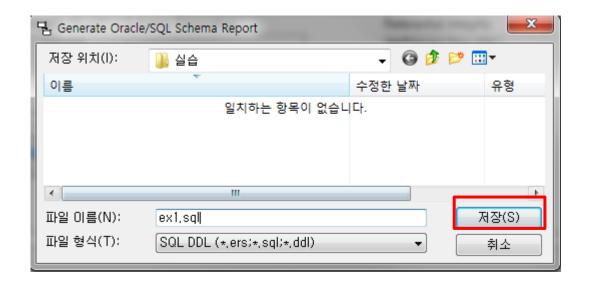


Preview

```
×
                                                      Oracle Schema Generation Preview
                                               羅 舞
                                                      A 🔁 🖺
CREATE TABLE GENRE
        GENRE CODE
                              INTEGER NOT NULL CONSTRAINT
UAL NUMBER 1344680426 CHECK (GENRE CODE BETWEEN 0 AND 99999),
        GENRE NAME
                              VARCHAR2(20) NULL,
CONSTRAINT XPK장르 PRIMARY KEY (GENRE CODE)
);
CREATE TABLE STUDIO
        STUDIO CODE
                              INTEGER NOT NULL CONSTRAINT
VAL NUMBER 969403095 CHECK (STUDIO_CODE BETWEEN 0 AND 99999),
                              VARCHAR2(20) NULL,
        STUDIO NAME
        ADDRESS
                              VARCHAR2(20) NULL,
        PHONENO
                              VARCHAR2(20) NULL,
CONSTRAINT XPK영화사 PRIMARY KEY (STUDIO CODE)
);
Table Filter: 10/10
                                             Generate...
                                                          Close
```



• Report





Oracle에 적용

• 계정 만들기

- Id: db2012

Password : database2012

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: system

Enter password:

Connected to:

Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production

SQL> CREATE USER db2012 identified by database2012;

User created.

SQL> GRANT connect, resource TO db2012;

Grant succeeded.



Oracle에 적용

|SQL> select * from tab; no rows selected @ 명령어로 sql 파일 실행 SQL> @ex1.sql ex) @ 파일경로/파일명.sql Table created. Table created. Table created. Table created. Table created.



Oracle에 적용 (Cont).

TNAME	TABTYPE CLUSTERID
ACTOR	TABLE
CAST	TABLE
CODECATEGORY	TABLE
GENRE	TABLE
GENRE_MOVIE	TABLE
MOVIE	TABLE
MOVIECATEGORY	TABLE
MOVIE_STATUS_HISTORY	TABLE
STUDIO	TABLE
TOTALCODE	TABLE



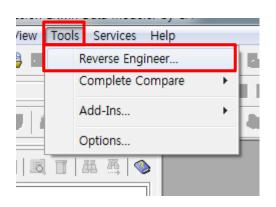
Oracle에 적용 (Cont).

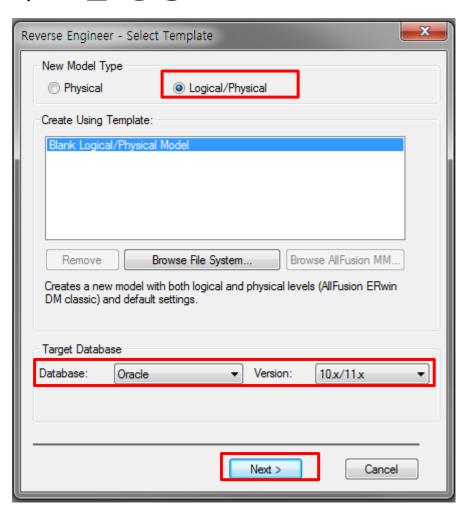
SQL> desc studio; Name	Null?	Туре
STUDIO_CODE STUDIO_NAME ADDRESS PHONENO	NOT NULL	NUMBER(38) VARCHAR2(20) VARCHAR2(20) VARCHAR2(20)



Reverse Engineering

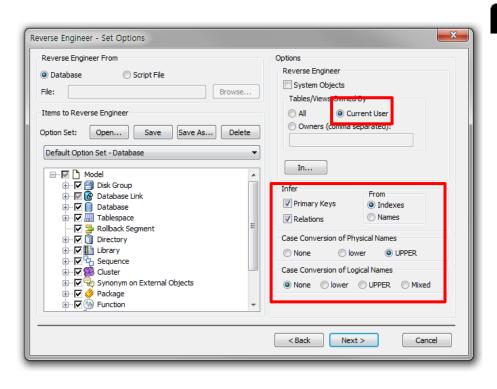
• 생성되어있는 테이블로부터 모델 생성

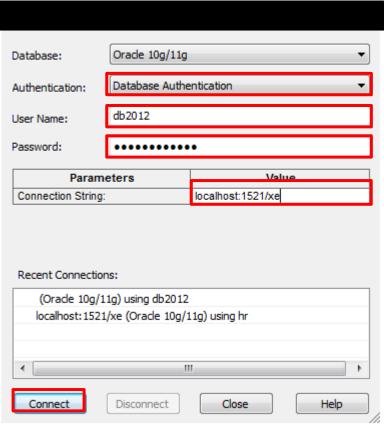






Reverse Engineering (Cont.)







Reverse Engineering (Cont.)

