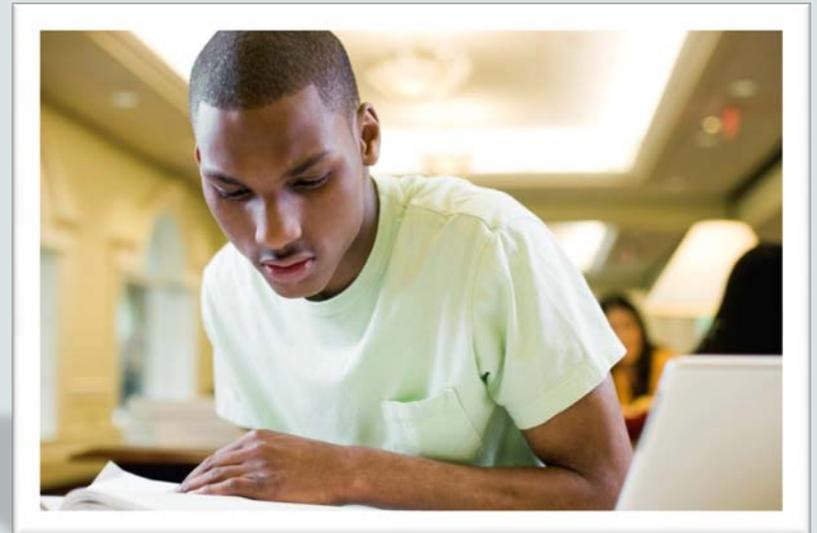




Database Foundations

5-2

Mapping Primary and Foreign Keys



Road Map

Mapping
Entities and
Attributes



You are here

Mapping
Primary and
Foreign Keys

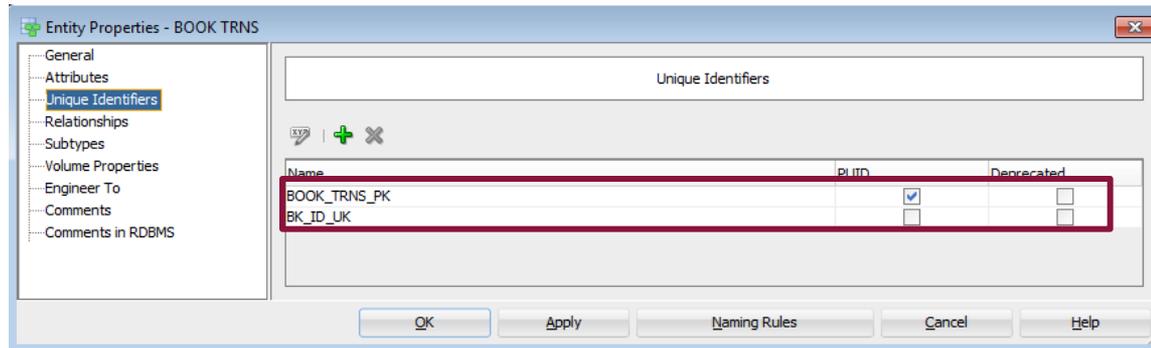
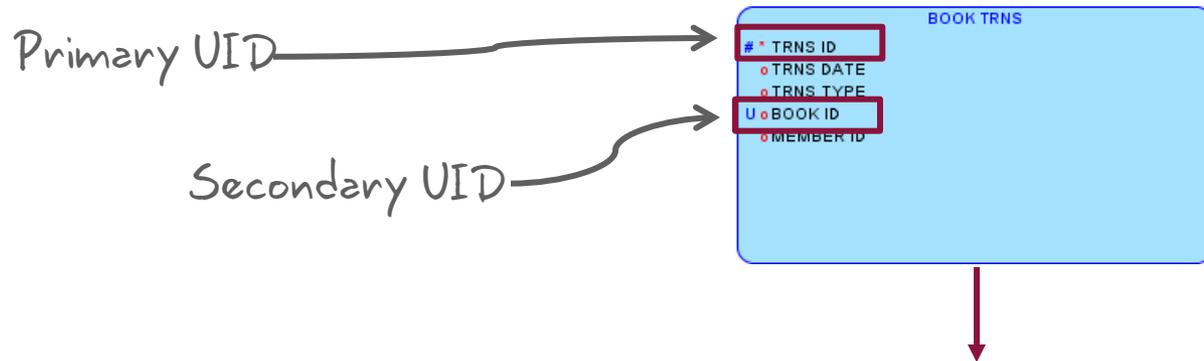
Objectives

This lesson covers the following objectives:

- Map UIDs to primary keys
- Engineer UIDs
- Map relationships to foreign keys
- Define naming templates
- Apply templates to the relational model
- Map exclusive relationships to foreign keys
- Map subtypes to tables
- Identify overlapping and folding keys



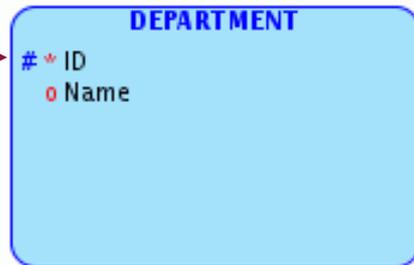
Basic Mapping: Unique Identifiers



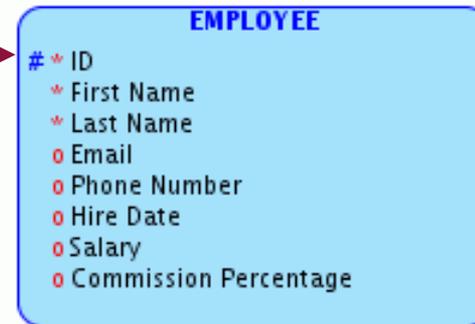
Mapping UIDs to Primary Keys

Entities

Unique identifier

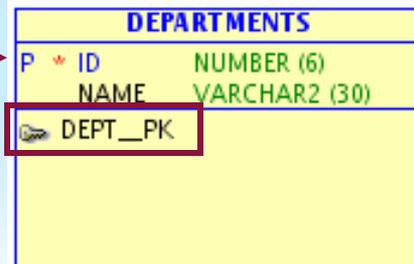


Unique identifier

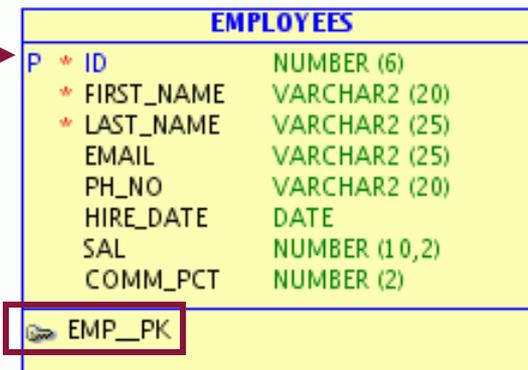


Tables

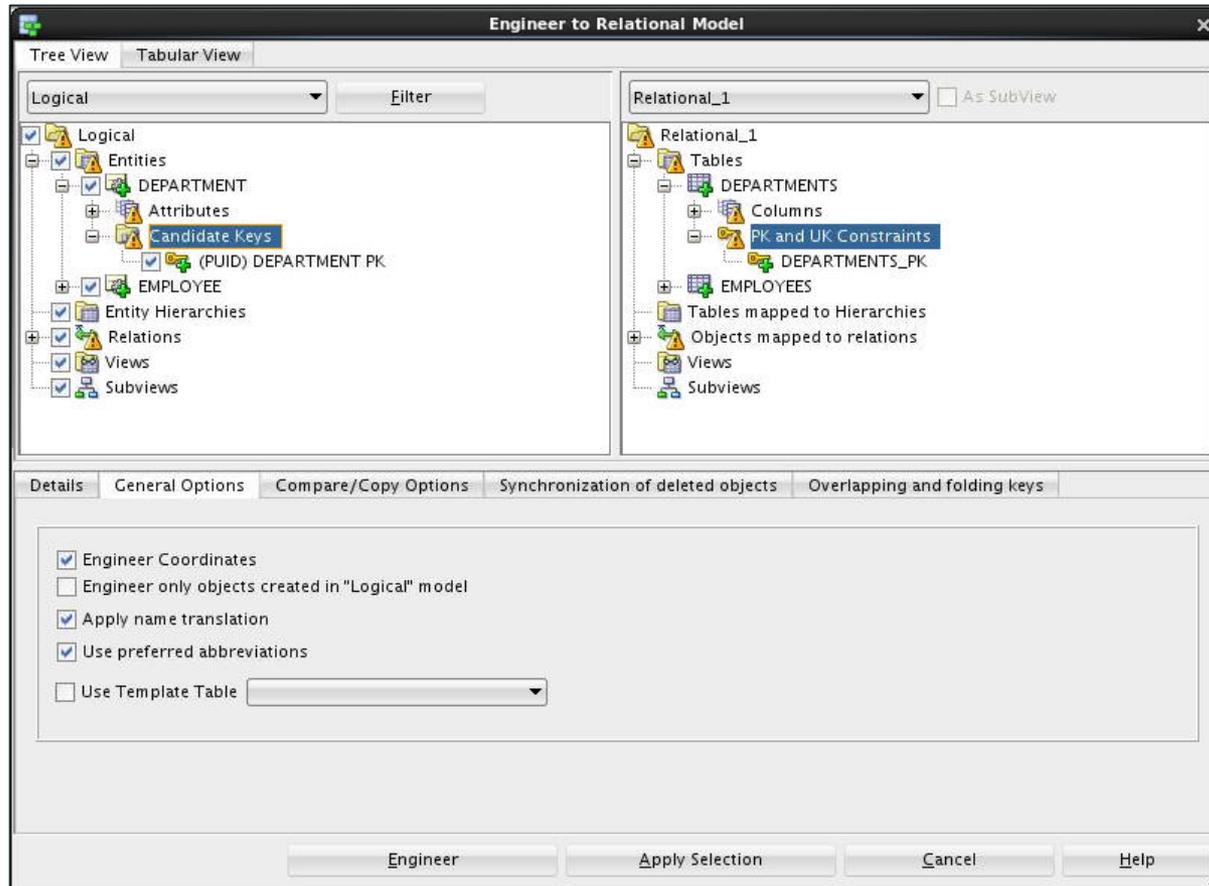
Primary key



Primary key

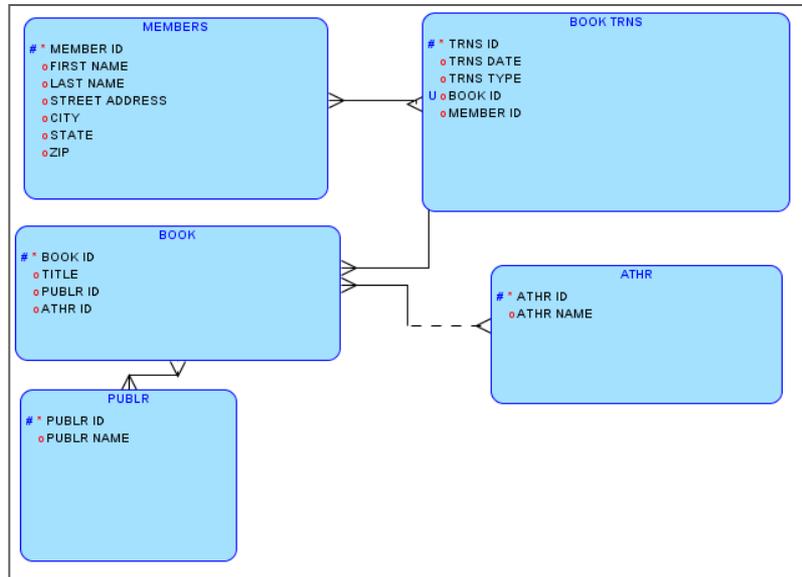


Engineering UIs

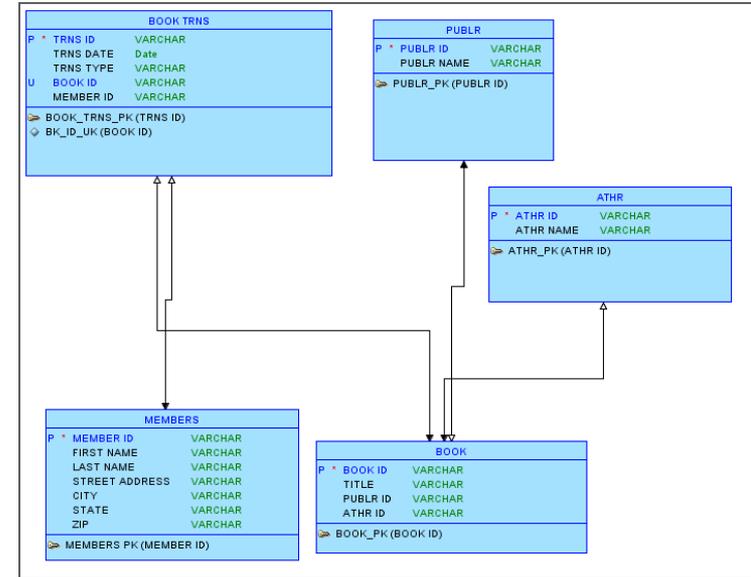


Relationship Mapping

Barker Notation

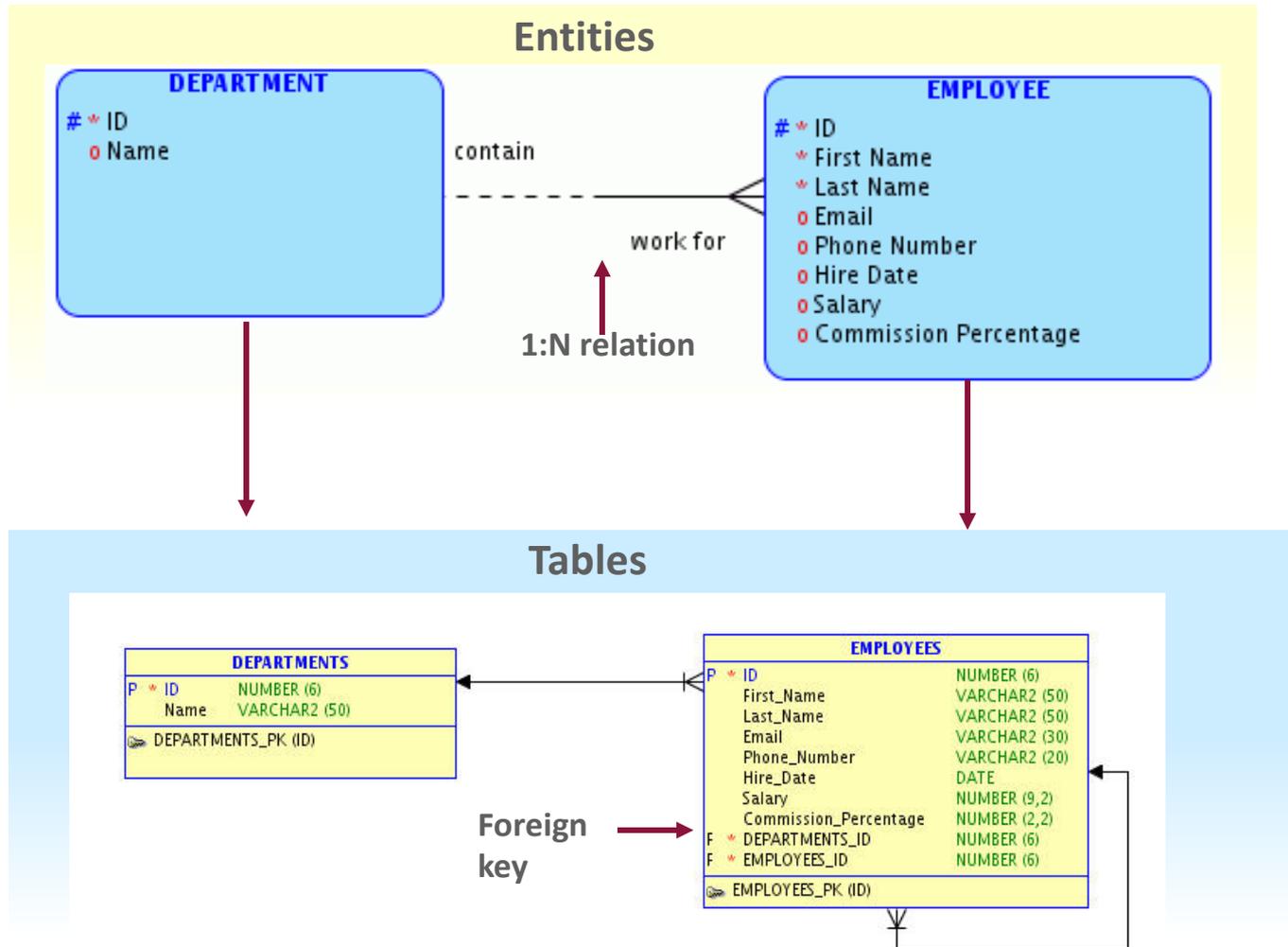


Bachman Notation

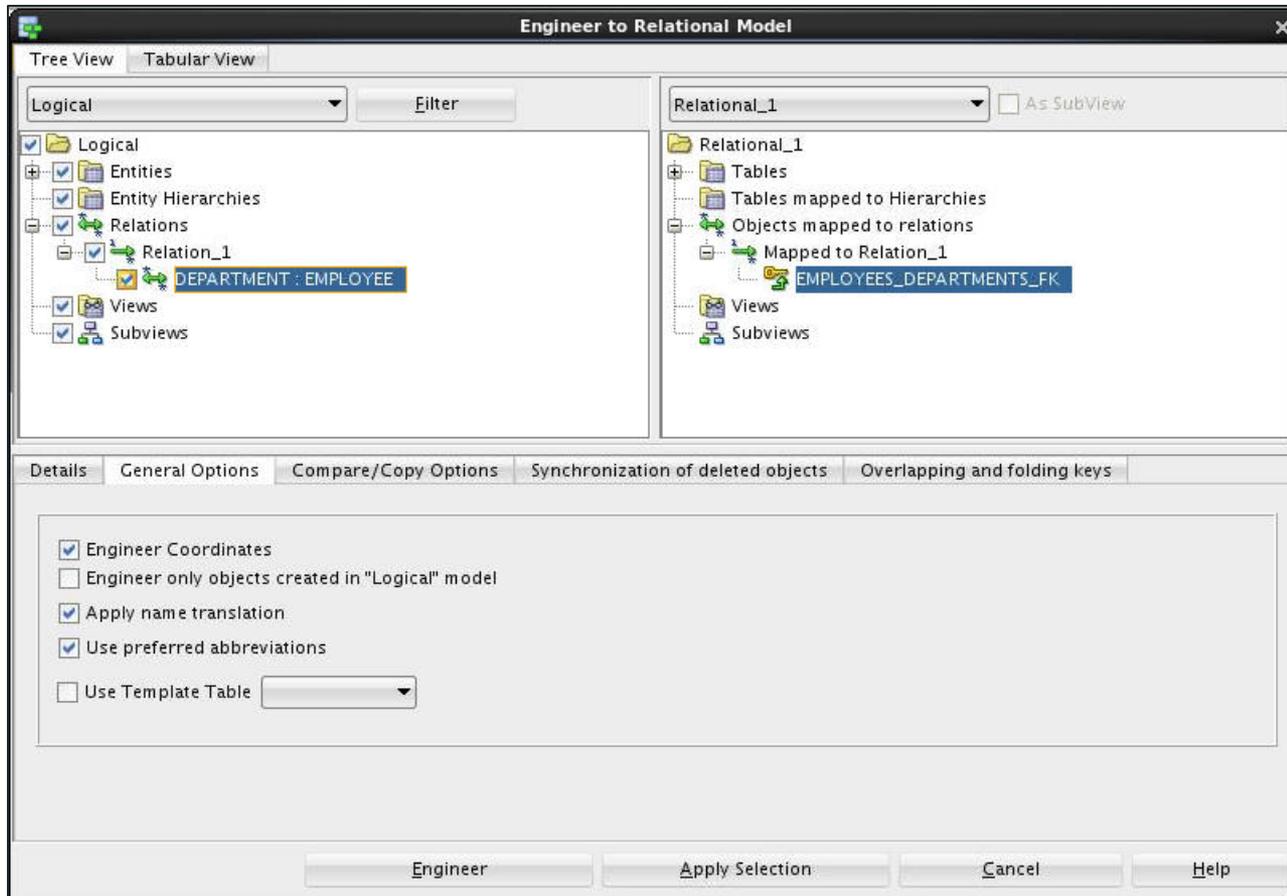


Barker Notation	Bachman Notation	Transformed To
#	P	Primary Key
U	U	Unique Key
Relationship	Relationship	Foreign Key

Mapping Relationships to Foreign Keys



Mapping Relationships to Foreign Keys



Defining Naming Templates

Right-click Design > Properties > Naming Standard > Templates.

The screenshot shows the Oracle Design Properties dialog for 'Untitled_1'. The left pane shows the 'Naming Standard' tree with 'Templates' selected. The right pane shows the 'Templates' configuration table.

Templates		
Table Constraints		
Primary Key	{table}_PK	Add Variable
Foreign Key	{child}_{parent}_FK	Add Variable
Check Constraint	{table}_CK	Add Variable
Unique Constraint	{table}_{column}_UN	Add Variable
Index	{table}_{column}_IDX	Add Variable
Automatic Index	{table}_{column}_IDX	Add Variable
Column Check Constraint	CK_{table}_{column}	Add Variable
Column Foreign Key	{ref table}_{ref column}	Add Variable
Surrogate Key	{table abbr}_PK	Add Variable
Surrogate Key Column	{table abbr}_ID	Add Variable
Discriminator Column	{table abbr}_TYPE	Add Variable
Entity identifier		
Primary Identifier	{entity} PK	Add Variable
Attribute Relation	{ref entity}_{ref attribute}	Add Variable
Example		
Example	<input type="text"/>	Primary Key

Example: Naming Templates

- Table name: ADMIN
- Model name: ORACLEDEMO

Template	Result
{table}_PK	ADMIN_PK
SUBSTR(7,4,FRONT,{model})	DEMO
SUBSTR(1,3,FRONT,{table})	ADM
SUBSTR(1,3,FRONT,TABLE)	TAB (where "TABLE" is a constant rather than a variable)
IX_SUBSTR(7,4,FRONT,{model})_SUBSTR(1,3,FRONT,{table})_{seq nr}	IX_DEMO_ADM_1

Applying Templates to One Table

Table Properties - EMPLOYEES

General

Name: EMPLOYEES

Long Name: EMPLOYEES

Abbreviation: EMP

Engineer:

PK Name:

Based on Structure:

Schema:

Register as Spatial:

Object Identifier:

Allow Type Substitution:

Generate in DDL:

Engineer as Relation:

Allow Columns Reorder During Engineering:

Deprecated:

Apply Naming Rules

Apply to:

- Primary Keys
- Foreign Keys
- Check Constraints
- Unique Constraints
- Indexes
- Column Check Constraints
- Column Foreign Keys

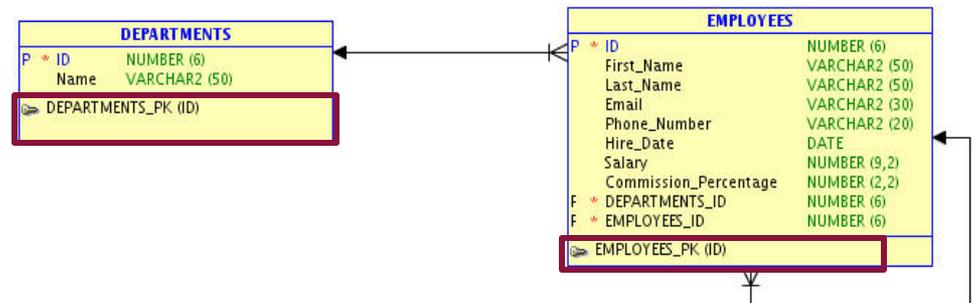
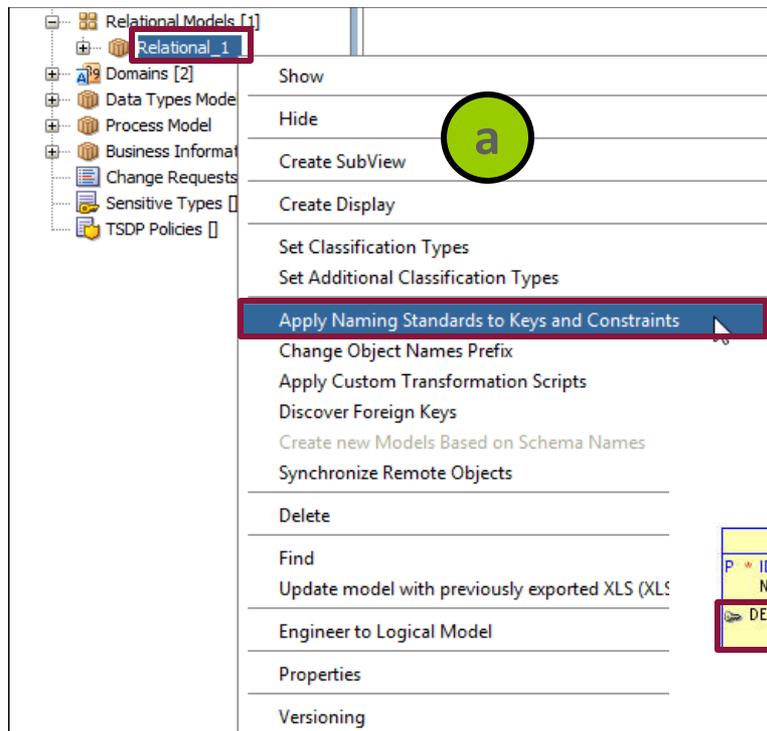
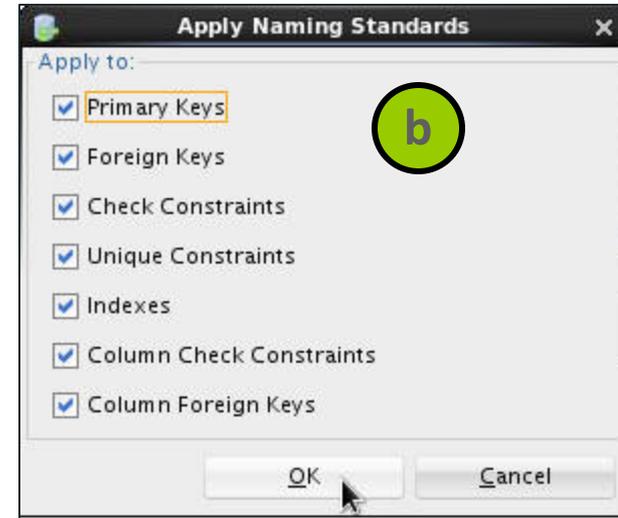
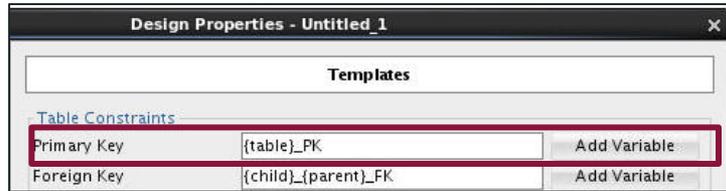
OK Cancel

OK Apply Naming Rules

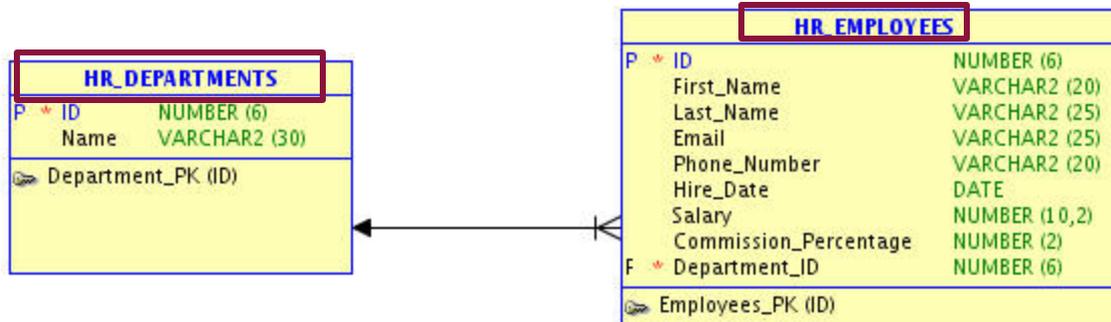
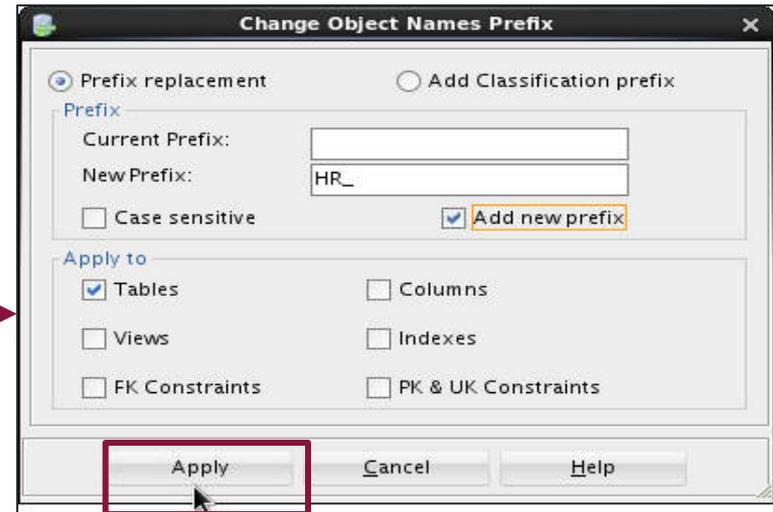
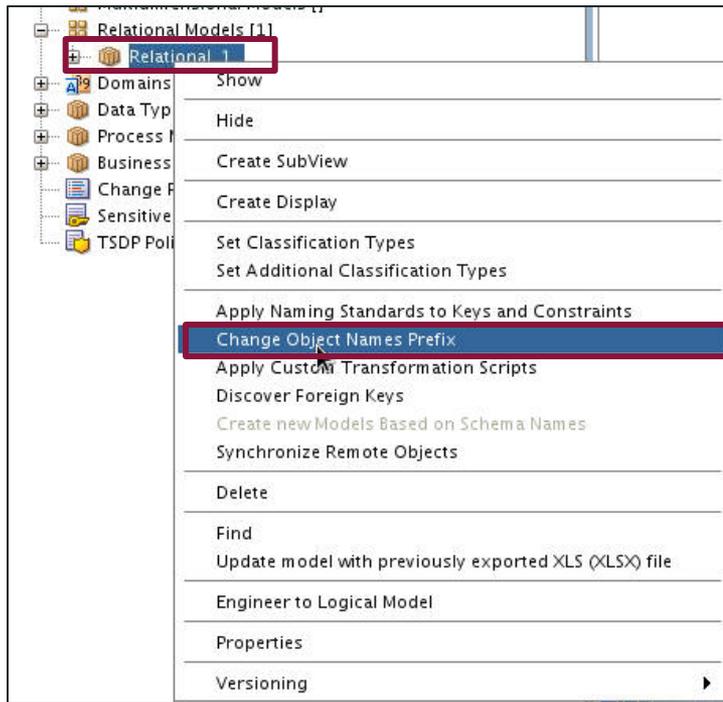
EMPLOYEES

P	ID	NUMBER (6)
*	FIRST_NAME	VARCHAR2 (20)
*	LAST_NAME	VARCHAR2 (25)
	EMAIL	VARCHAR2 (25)
	PH_NO	VARCHAR2 (20)
	HIRE_DATE	DATE
	SAL	NUMBER (10,2)
	COMM_PCT	NUMBER (2)
F	DEPARTMENTS_ID	NUMBER (6)
	EMPLOYEES_PK	

Applying Templates to the Relational Model



Managing Prefixes



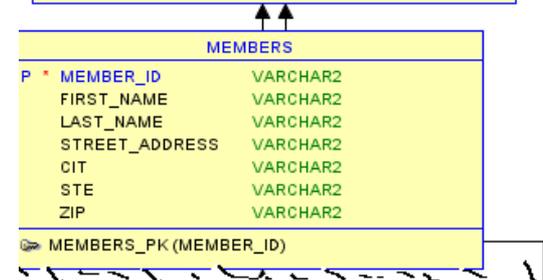
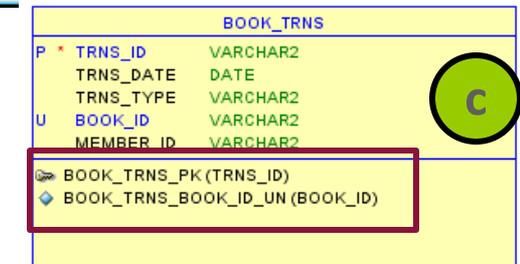
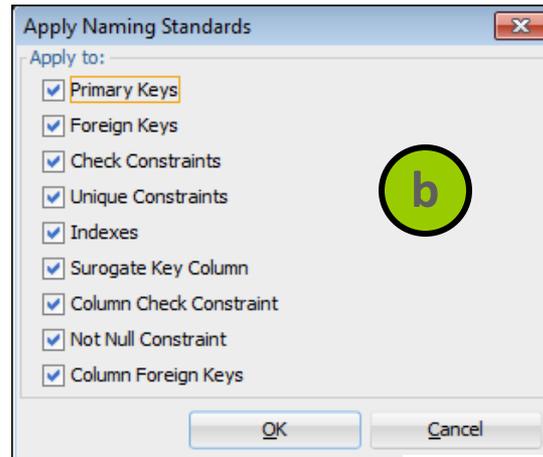
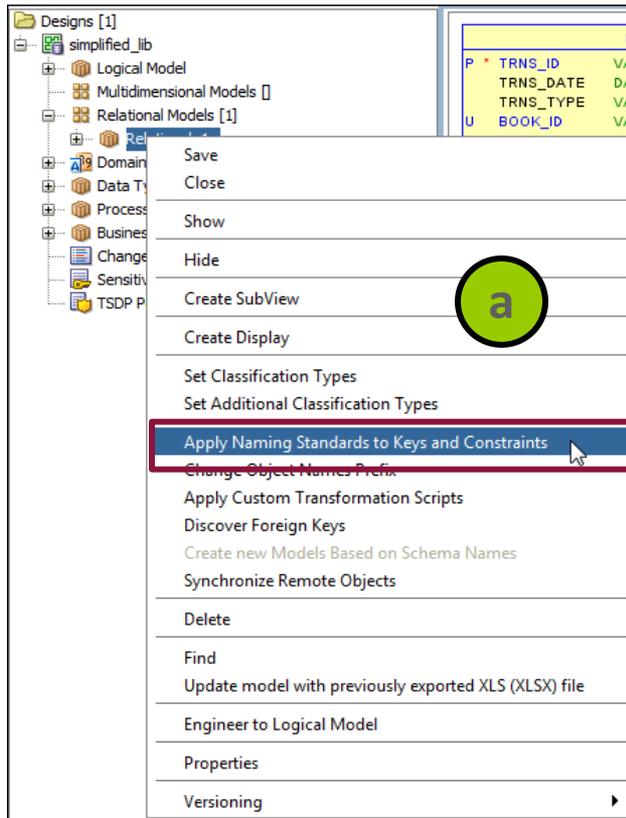
Case Scenario: Template and Prefix

How about applying the naming templates to the Simplified Library Management Relational Model?

Maybe it might be a good idea to prefix the objects with "SLM"?



Case Scenario: Applying Naming Templates



Case Scenario: Applying the Prefix

Designs [1]
simplified_lib
Logical Model
Multidimensional Models []
Relational Models [1]
Relational 1
Domains [2]
Data Types M
Process Mod
Business Inf
Change Req
Sensitive Typ
TSDP Policies

- Save
- Close
- Show
- Hide
- Create SubView
- Create Display
- Set Classification Types
- Set Additional Classification Types
- Apply Naming Standards to Keys and Constraints
- Change Object Names Prefix**
- Apply Custom Transformation Scripts
- Discover Foreign Keys

Change Object Names Prefix

Prefix replacement Add Classification prefix

Prefix

Current Prefix:

New Prefix:

Case sensitive Add new prefix

Apply to

Tables Columns

Views Indexes

FK Constraints PK & UK Constraints

Apply Cancel Help

SLM_BOOK_TRNS

P	TRNS_ID	VARCHAR2
	TRNS_DATE	DATE
	TRNS_TYPE	VARCHAR2
U	BOOK_ID	VARCHAR2
	MEMBER_ID	VARCHAR2

BOOK_TRNS_PK (TRNS_ID)
BOOK_TRNS_BOOK_ID_UN (BOOK_ID)

SLM_MEMBERS

P	MEMBER_ID	VARCHAR2
	FIRST_NAME	VARCHAR2
	LAST_NAME	VARCHAR2
	STREET_ADDRESS	VARCHAR2
	CIT	VARCHAR2
	STE	VARCHAR2
	ZIP	VARCHAR2

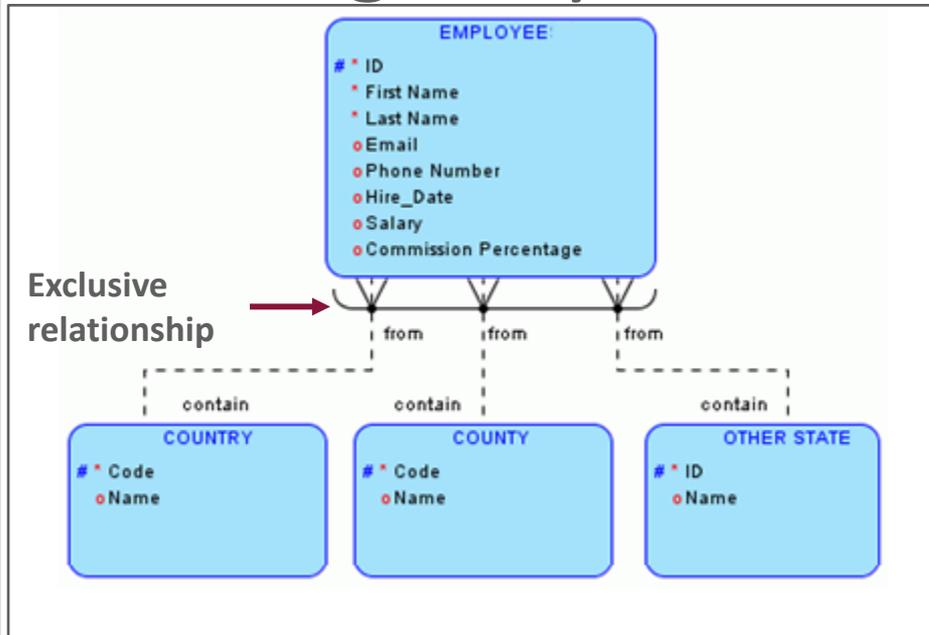
MEMBERS_PK (MEMBER_ID)

Message

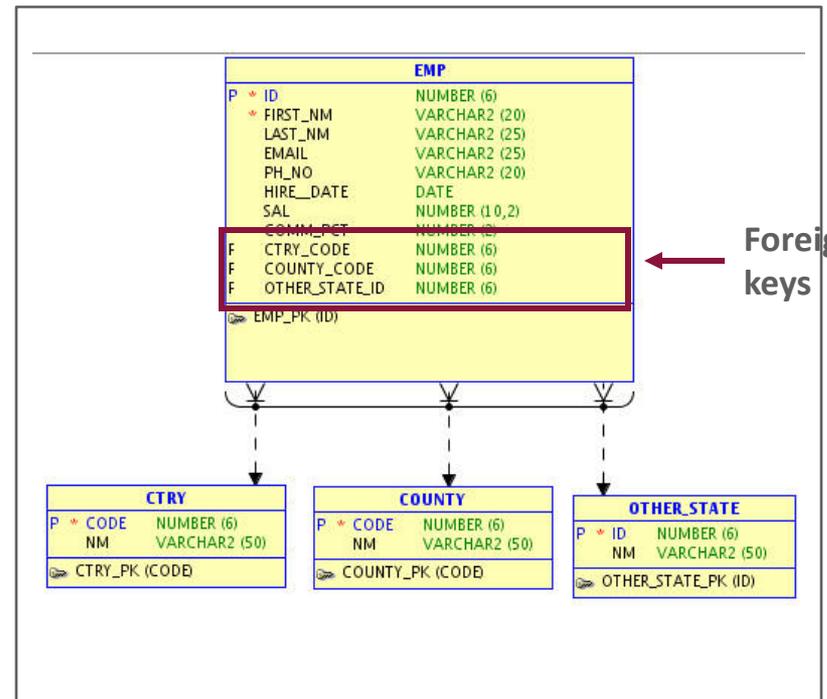
i 9 names have been changed

OK

Mapping Exclusive Relationships to Foreign Keys



← Entities



← Foreign keys

Tables →

Engineering Exclusive Relationships



Engineer to Relational Model

Tree View Tabular View

Logical Filter

Logical

- Logical
 - Entities
 - COUNTRY
 - COUNTY
 - EMPLOYEE
 - OTHER STATE
 - Entity Hierarchies
 - Relations
 - Relation_1
 - COUNTRY : EMPLOYEE
 - Relation_2
 - COUNTY : EMPLOYEE
 - Relation_3
 - OTHER STATE : EMPLOYEE
 - Views
 - Subviews

Relational_1 As SubView

Relational_1

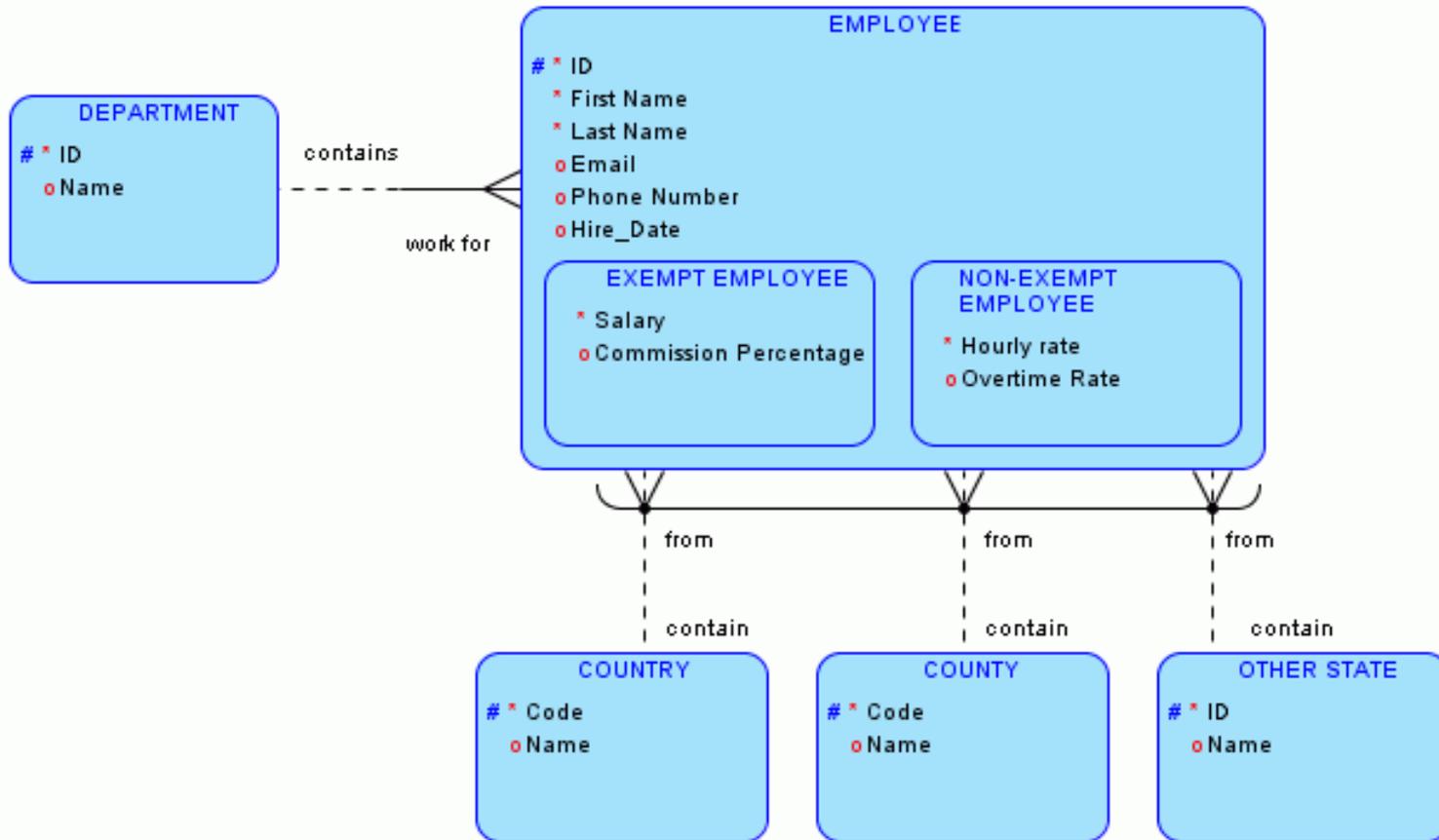
- Tables
 - COUNTRY
 - COUNTY
 - EMPLOYEE
 - OTHER_STATE
- Tables mapped to Hierarchies
- Objects mapped to relations
 - Mapped to Relation_1
 - EMPLOYEE_COUNTRY_FK
 - Mapped to Relation_2
 - EMPLOYEE_COUNTY_FK
 - Mapped to Relation_3
 - EMPLOYEE_OTHER_STATE_FK
- Views
- Subviews

Details General Options Compare/Copy Options Synchronization of deleted objects Overlapping and folding keys

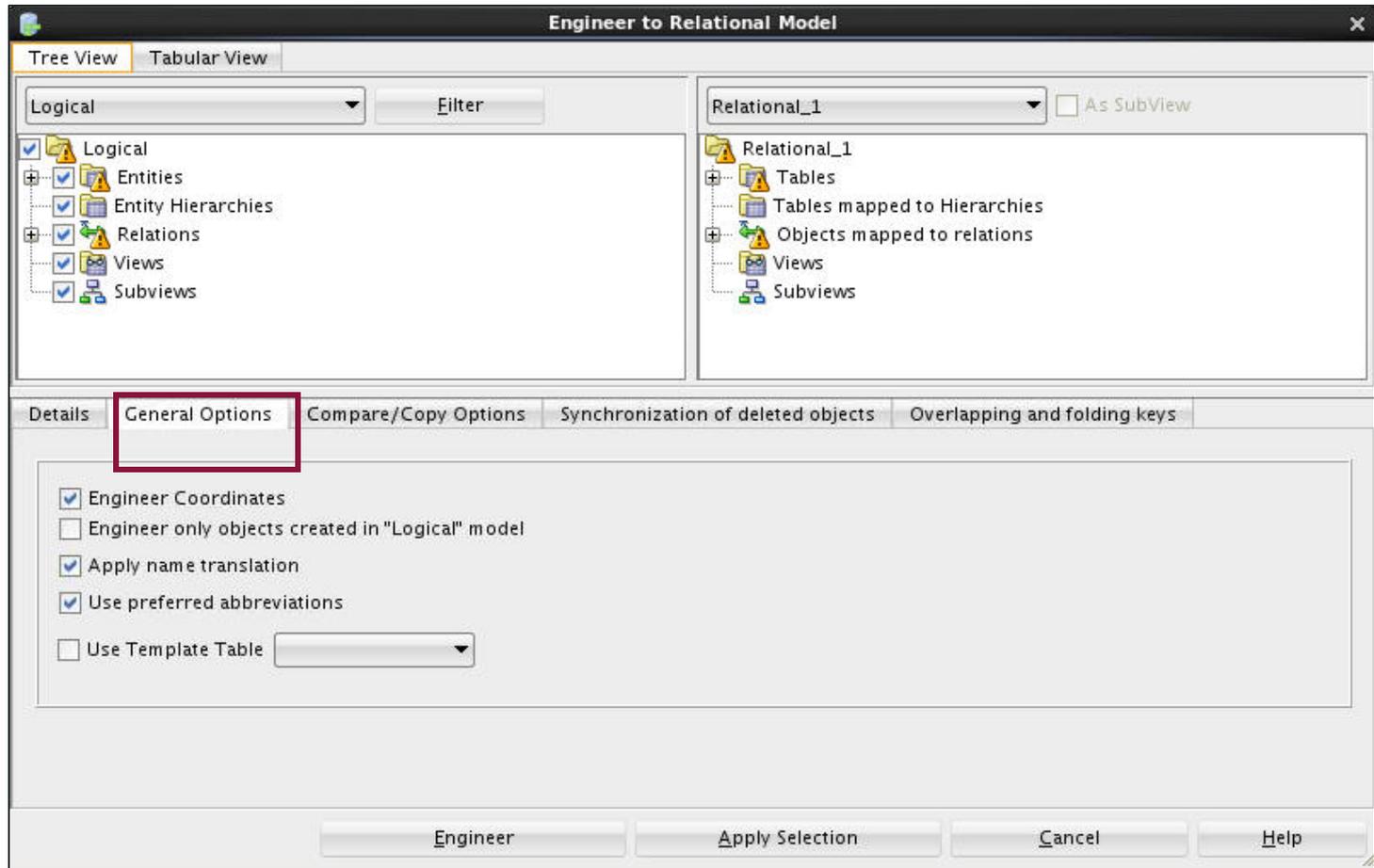
Engineer Coordinates
 Engineer only objects created in "Logical" model
 Apply name translation
 Use preferred abbreviations
 Use Template Table

Engineer Apply Selection Cancel Help

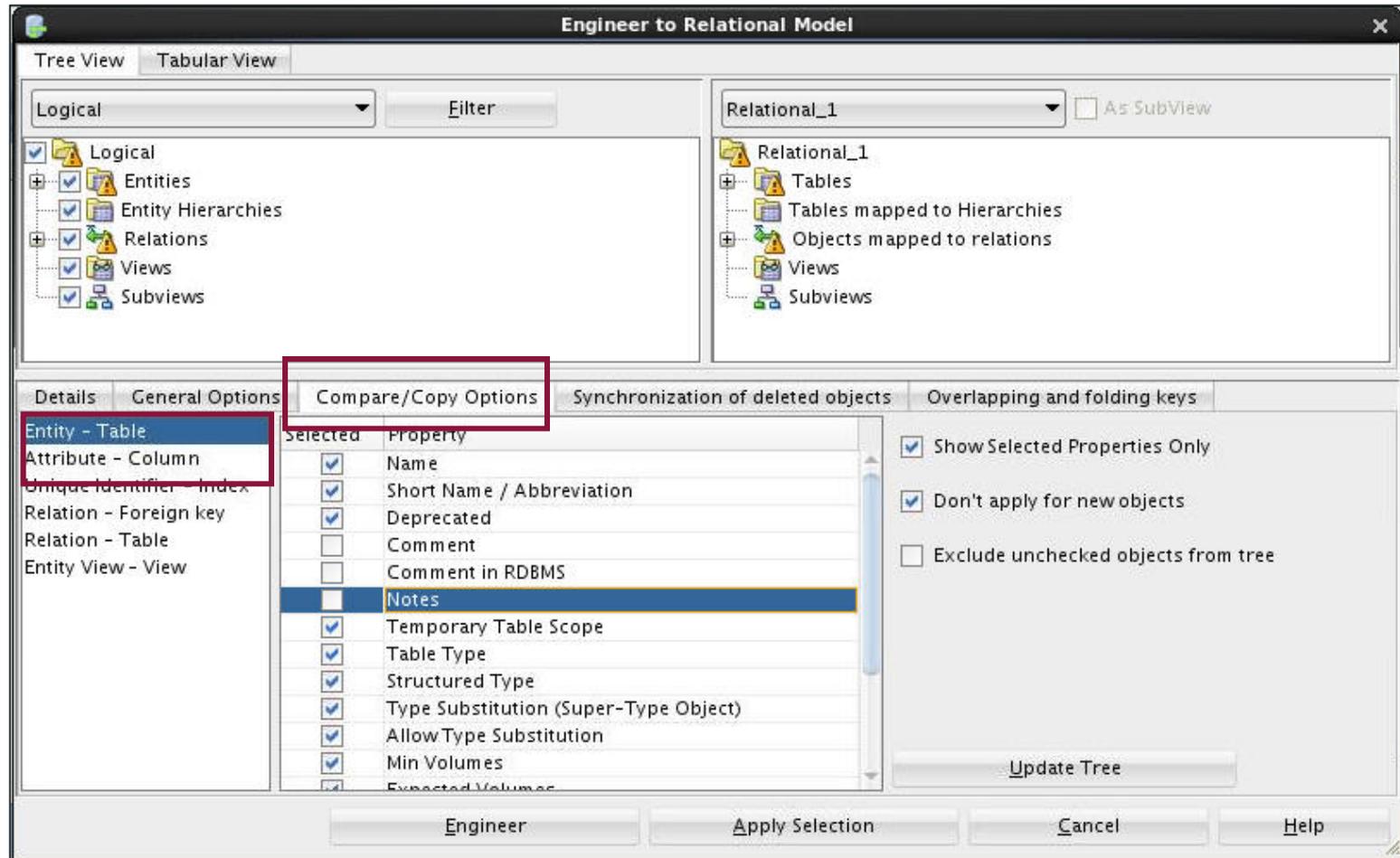
Mapping Subtypes to Tables



Applying General Options



Setting Compare/Copy Options



Viewing the Mapping Comparison

The screenshot shows the 'Engineer to Relational Model' dialog box. The 'Logical' tree view on the left has 'DEPARTMENT' selected. The 'Relational_1' tree view on the right has 'HR_DEPARTMENTS' selected. The 'Details' tab is active, showing a comparison table between the logical entity and the relational table.

Property	Selected	entity: DEPARTMENT	table: HR_DEPARTMENTS
Name	<input type="checkbox"/>	DEPARTMENT	HR_DEPARTMENTS
Short Name / Abbreviation	<input type="checkbox"/>		
Deprecated	<input type="checkbox"/>	No	No
Temporary Table Scope	<input type="checkbox"/>		
Table Type	<input type="checkbox"/>		
Structured Type	<input type="checkbox"/>		
Type Substitution (Super-Type Object)	<input type="checkbox"/>		
Allow Type Substitution	<input type="checkbox"/>	true	true
Min Volumes	<input type="checkbox"/>	0	0
Expected Volumes	<input type="checkbox"/>	0	0
Max Volumes	<input type="checkbox"/>	9999999	9999999
Growth Percent	<input type="checkbox"/>	0	0
Growth Type	<input type="checkbox"/>	Year	Year

Buttons at the bottom: Engineer, Apply Selection, Cancel, Help.

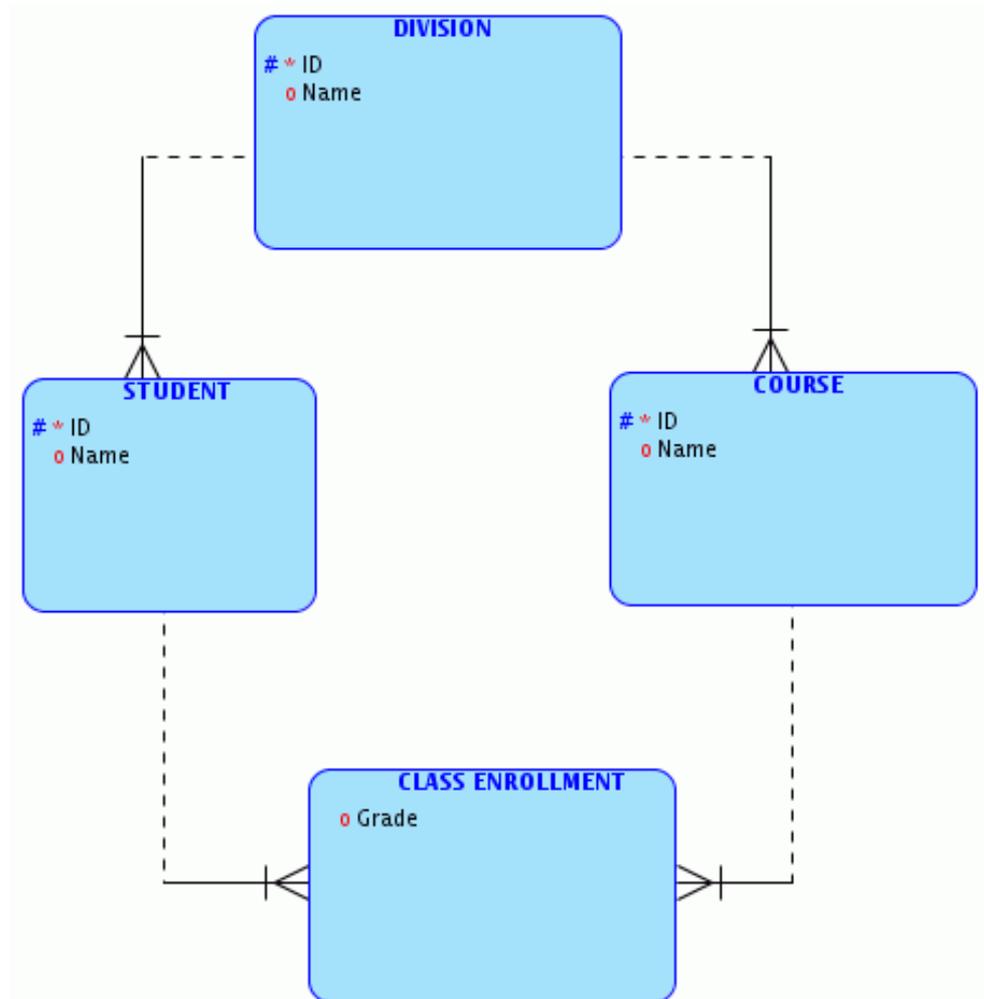
Synchronizing Deleted Objects

The screenshot displays the 'Engineer to Relational Model' dialog box. The 'Logical' pane on the left shows a tree view of the 'Rental Agreement' entity, including attributes like 'Number', 'Date of Rental', 'Duration', 'Date of Return', 'Deposit Paid', 'Individual_Driver's License State', 'Individual_Driver's License Number', 'Company_Number', 'Vehicle_ID', 'Office_Number', 'Office_Number2', 'Daily Rental Rate', and 'Rate Per Mile'. The 'Relational_1' pane on the right shows the corresponding 'RENTAL_AGREEMENTS' table with columns like 'NO', 'DATE_OF RENTAL', 'DURATION', 'DATE_OF RETURN', 'DEP_PAID', 'IND__DRIVER'S_LIC_ST', 'IND__DRIVER'S_LIC_NO', 'COMPANY__NUMBER', 'VEHICLE__ID', 'OFFICE__NUMBER', 'OFFICE__NUMBER2', 'DAILY RENTAL RATE', and 'RATE PER MILE'. The 'Synchronization of deleted objects' tab is active, showing a table with columns 'Select', 'Deleted', and 'To be deleted'. Two rows are listed: 'Daily Rental Rate' and 'Rate Per Mile'. A red box highlights the 'Synchronization of deleted objects' tab.

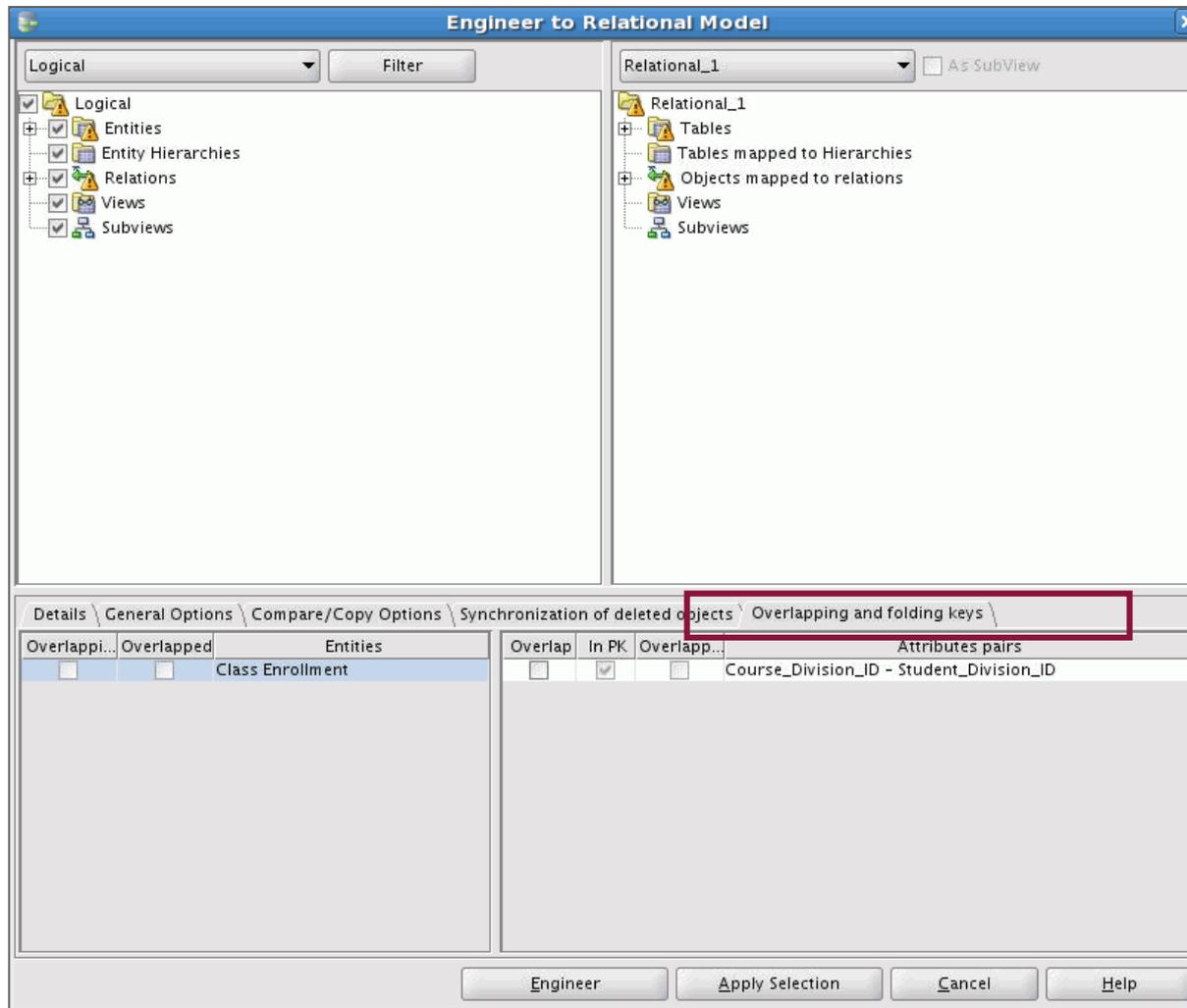
Select	Deleted	To be deleted
<input type="checkbox"/>	Daily Rental Rate	RENTAL_AGREEMENTS.DAILY RENTAL RATE
<input type="checkbox"/>	Rate Per Mile	RENTAL_AGREEMENTS.RATE PER MILE

Identifying Overlapping and Folding Keys

- Two attributes in the same entity relate to the same UID attribute.
- You can fold the keys into one column in the relational model during engineering.



Identifying Overlapping and Folding Keys



Summary

In this lesson, you should have learned how to:

- Map UIDs to primary keys
- Engineer UIDs
- Map relationships to foreign keys
- Define naming templates
- Apply templates to the relational model
- Map exclusive relationships to foreign keys
- Map subtypes to tables
- Identify overlapping and folding keys



