Série n°6 Bis : Ateliers SQL Data Modeler (Oracle)

Getting started with data Modeler

Adding a Table to An Existing Database

Purpose

This tutorial shows you how to add a table to an existing database using Oracle SQL Developer Data Modeler.

Time to Complete

Approximately 15 minutes

Overview

Oracle SQL Developer Data Modeler offers a full spectrum of data and database modeling tools and utilities, including Entity Relationship modeling, Relational (Schema), Data Types or Object Type modeling, and Multidimensional modeling and DDL generation. It includes importing from and exporting to a variety of sources and targets, provides a variety of formatting options and validates the models through a predefined set of Design Rules.

Oracle SQL Developer Data Modeler will be released as an extension to SQL Developer and as a standalone product, for those developers who only want to work with visual data modeling.

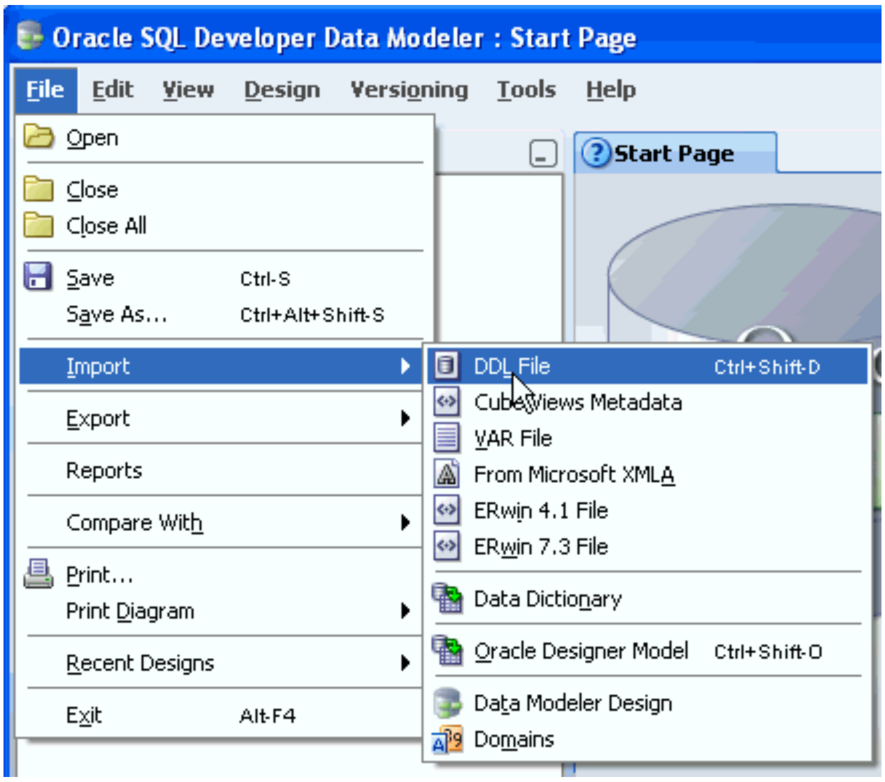
In this tutorial, you create an initial relational model by importing an existing script (DDL), add a new table, link the table to an existing table, create a sub view and generate the DDL.

Importing the DDL for the HR Schema

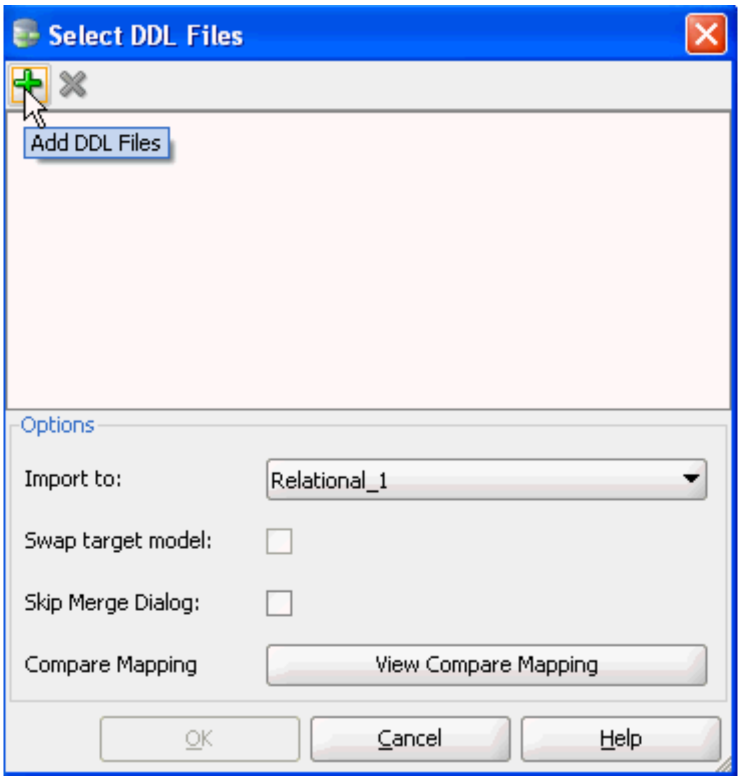
In this section, you import the DDL from the HR sample schema to create a relational model. Perform the following steps:

1 . Open **Oracle SQL Developer Data Modeler** from the icon on your desktop.

2 . Select **File** > **Import** > **DDL File**.

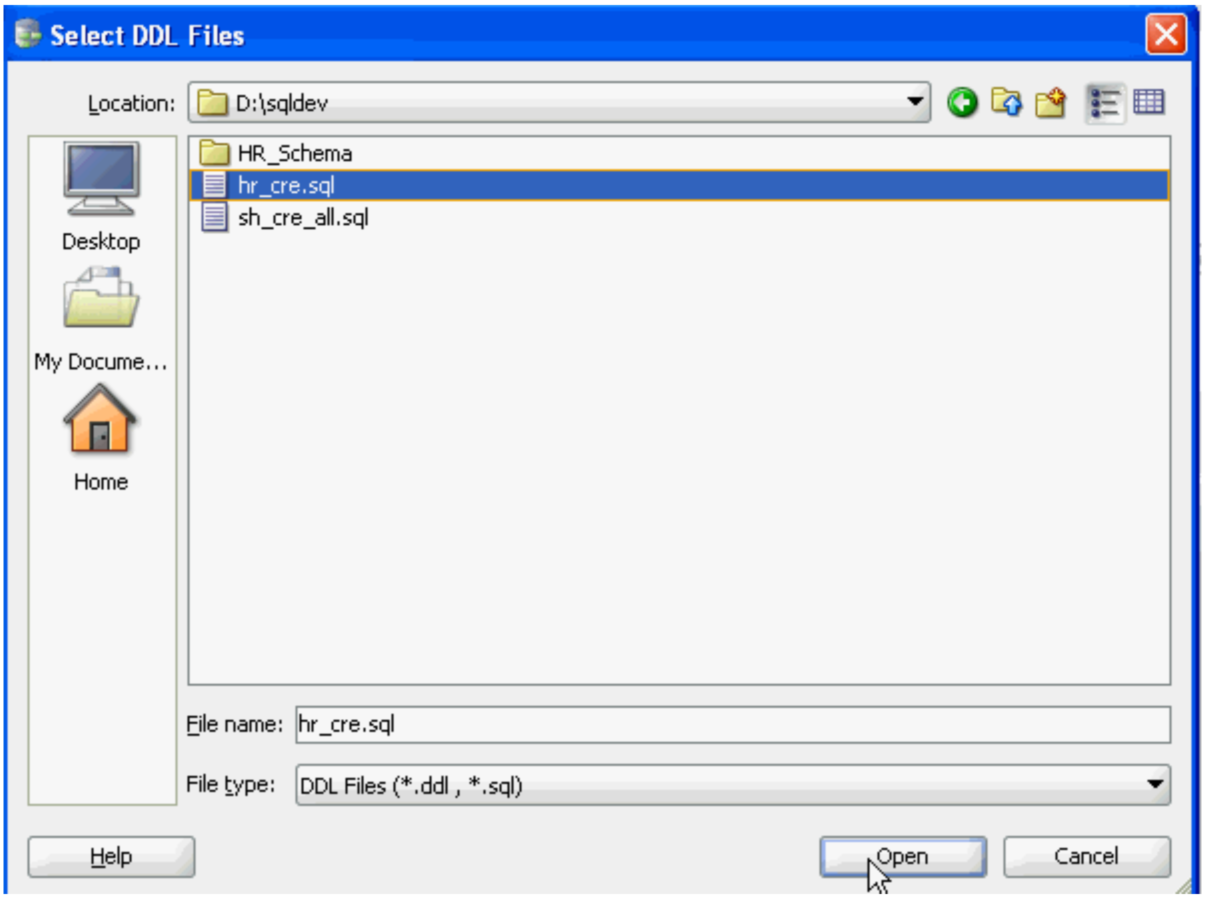


3 . You can add multiple DDL files to be imported at the same time. Click the '+' icon to add a DDL file.

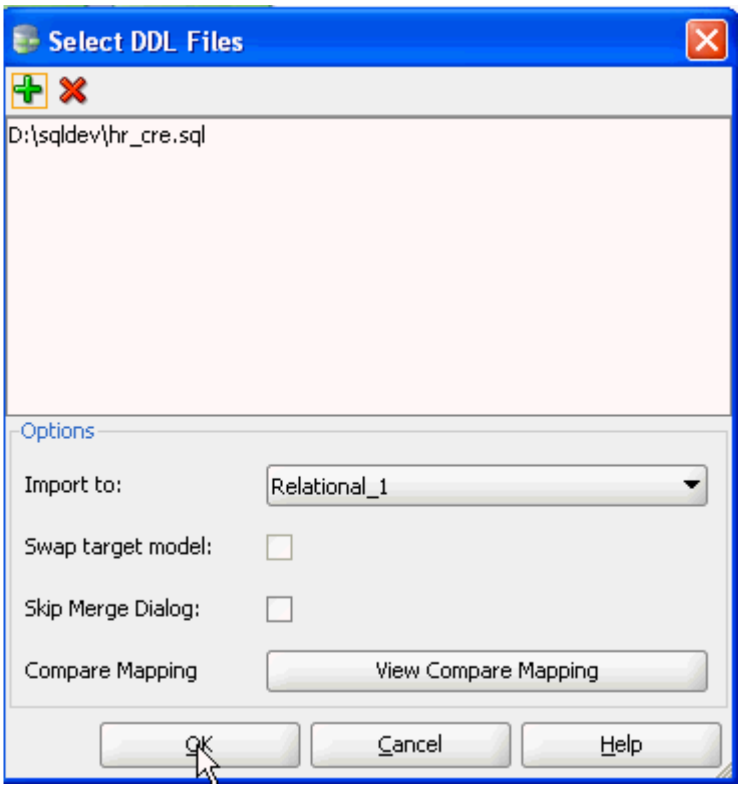


Import to: Relational_1
Swap target model: ☐
Skip Merge Dialog: ☐
Compare Mapping

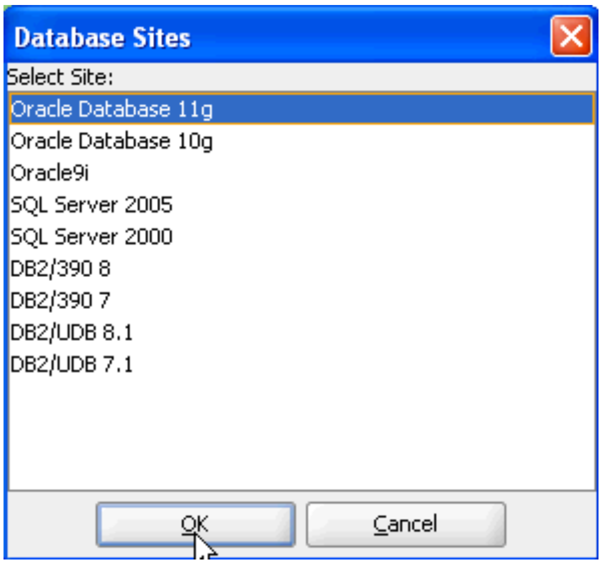
4 . Select **hr_cre.sql** from the **sqldev** directory and click **Open**.


Select DDL Files
Location: D:\sqldev
Desktop
My Docume...
Home
HR_Schema
hr_cre.sql
sh_cre_all.sql
File name: hr_cre.sql
File type: DDL Files (*.ddl , *.sql)
Help
Open
Cancel

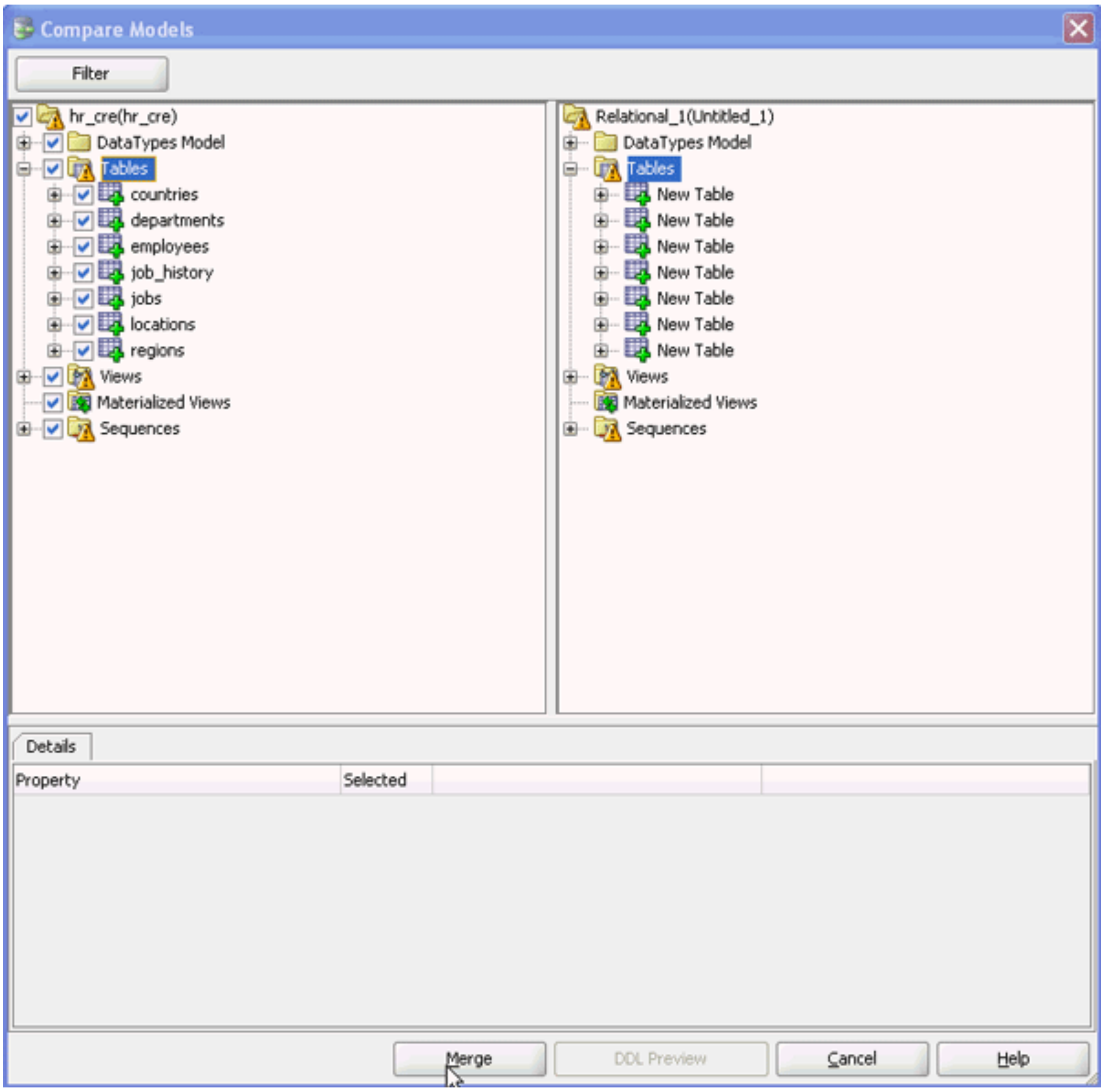
5 . Click **OK**.


Select DDL Files
D:\sqldev\hr_cre.sql
Options
Import to: Relational_1
Swap target model:
Skip Merge Dialog:
Compare Mapping
View Compare Mapping
OK
Cancel
Help

6 . Select **Oracle Database 11g** and click **OK.**

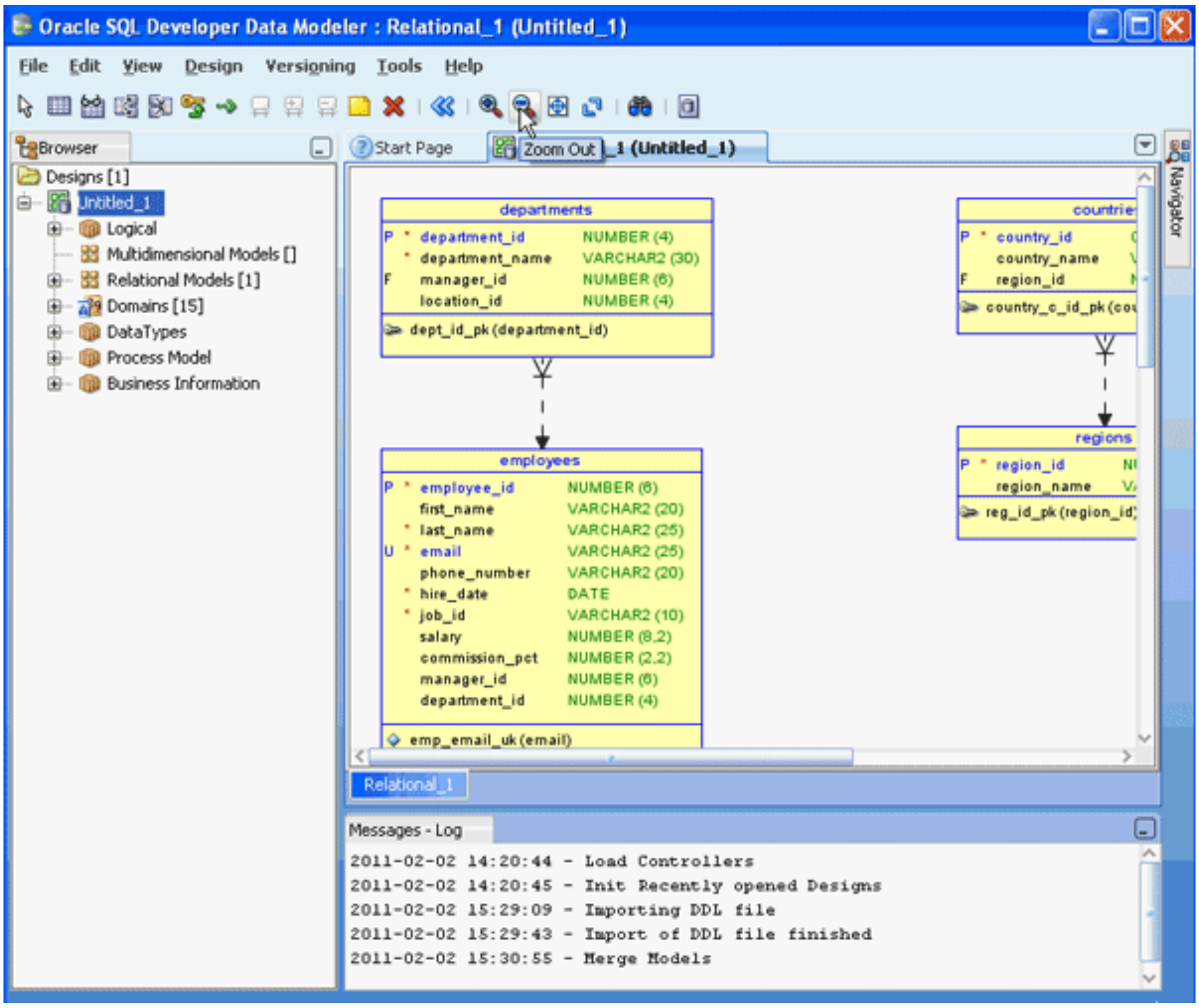


7 . The Compare Model window appears. You can view the changes that will occur when the DDL file is imported. Expand **T** the list of tables that will be created. Click **Merge**.



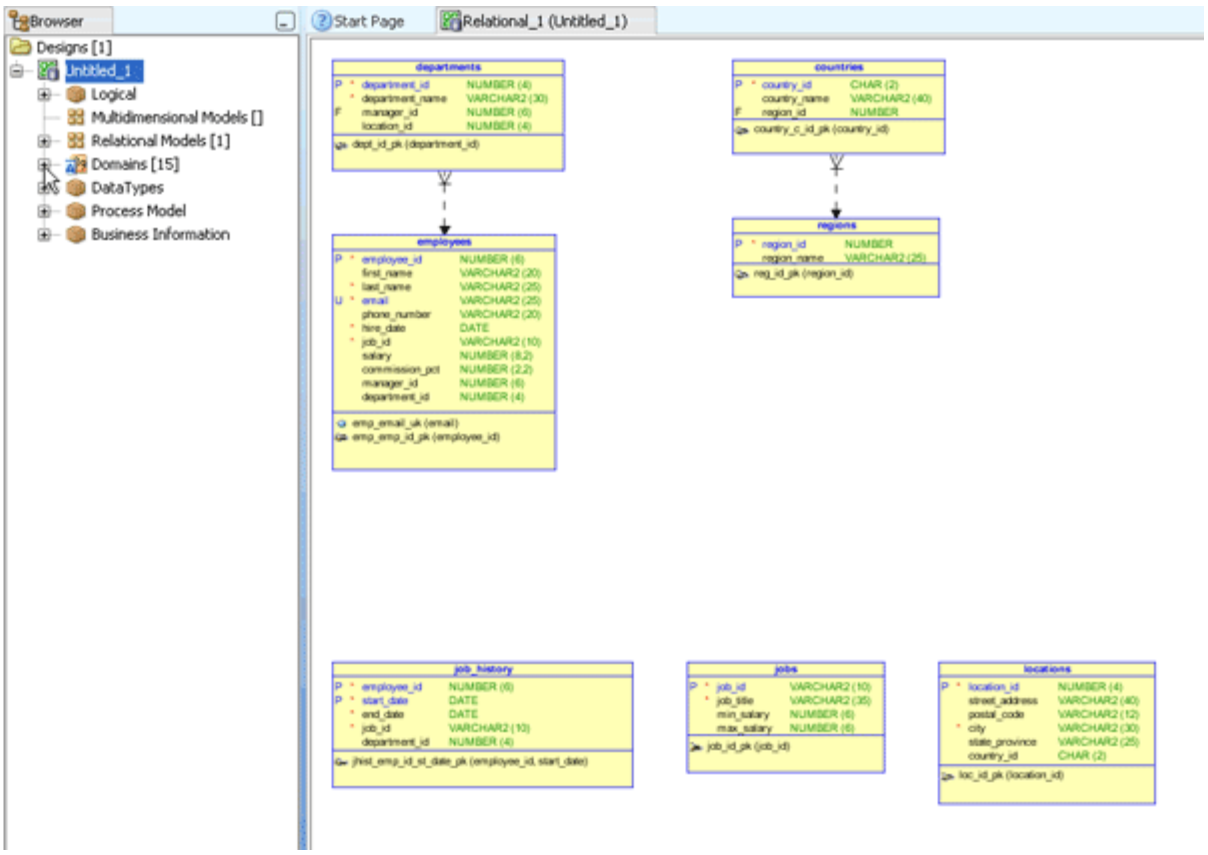
8 .

The relational diagram is displayed. You can zoom in and out. click the Zoom Out 🔍 icon.



```
2011-02-02 14:20:44 - Load Controllers
2011-02-02 14:20:45 - Init Recently opened Designs
2011-02-02 15:29:09 - Importing DDL file
2011-02-02 15:29:43 - Import of DDL file finished
2011-02-02 15:30:55 - Merge Models
```

9 . You can now see more of the diagram. To examine the domains that were created in the browser tree, expand**Domains**.



```
departments
P * department_id      NUMBER (4)
  * department_name    VARCHAR2 (30)
F   manager_id         NUMBER (6)
    location_id        NUMBER (4)
dept_id_pk (department_id)
```

```
employees
P * employee_id        NUMBER (6)
    first_name         VARCHAR2 (20)
  * last_name          VARCHAR2 (25)
U * email              VARCHAR2 (25)
    phone_number       VARCHAR2 (20)
  * hire_date          DATE
  * job_id             VARCHAR2 (10)
    salary             NUMBER (8,2)
    commission_pct     NUMBER (2,2)
    manager_id         NUMBER (6)
    department_id      NUMBER (4)
emp_email_uk (email)
emp_emp_id_pk (employee_id)
```

```
countries
P * country_id         CHAR (2)
    country_name       VARCHAR2 (40)
F   region_id          NUMBER
country_c_id_pk (country_id)
```

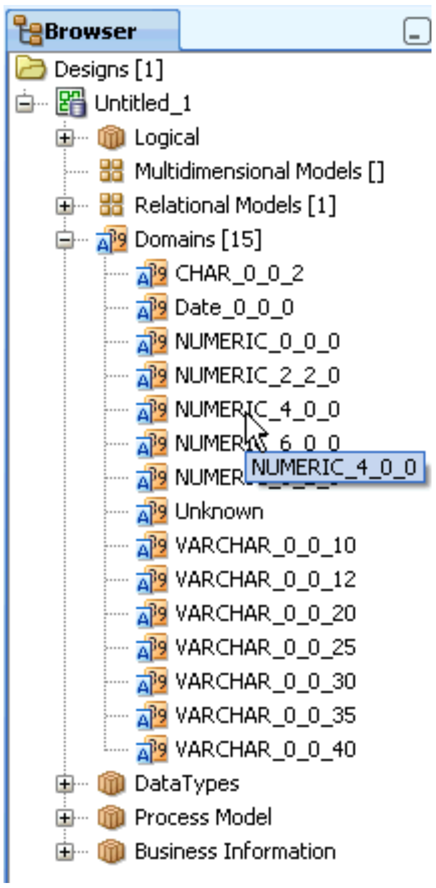
```
regions
P * region_id          NUMBER
    region_name        VARCHAR2 (25)
reg_id_pk (region_id)
```

```
job_history
P * employee_id        NUMBER (6)
P * start_date         DATE
  * end_date           DATE
  * job_id             VARCHAR2 (10)
    department_id      NUMBER (4)
jhist_emp_id_st_date_pk (employee_id, start_date)
```

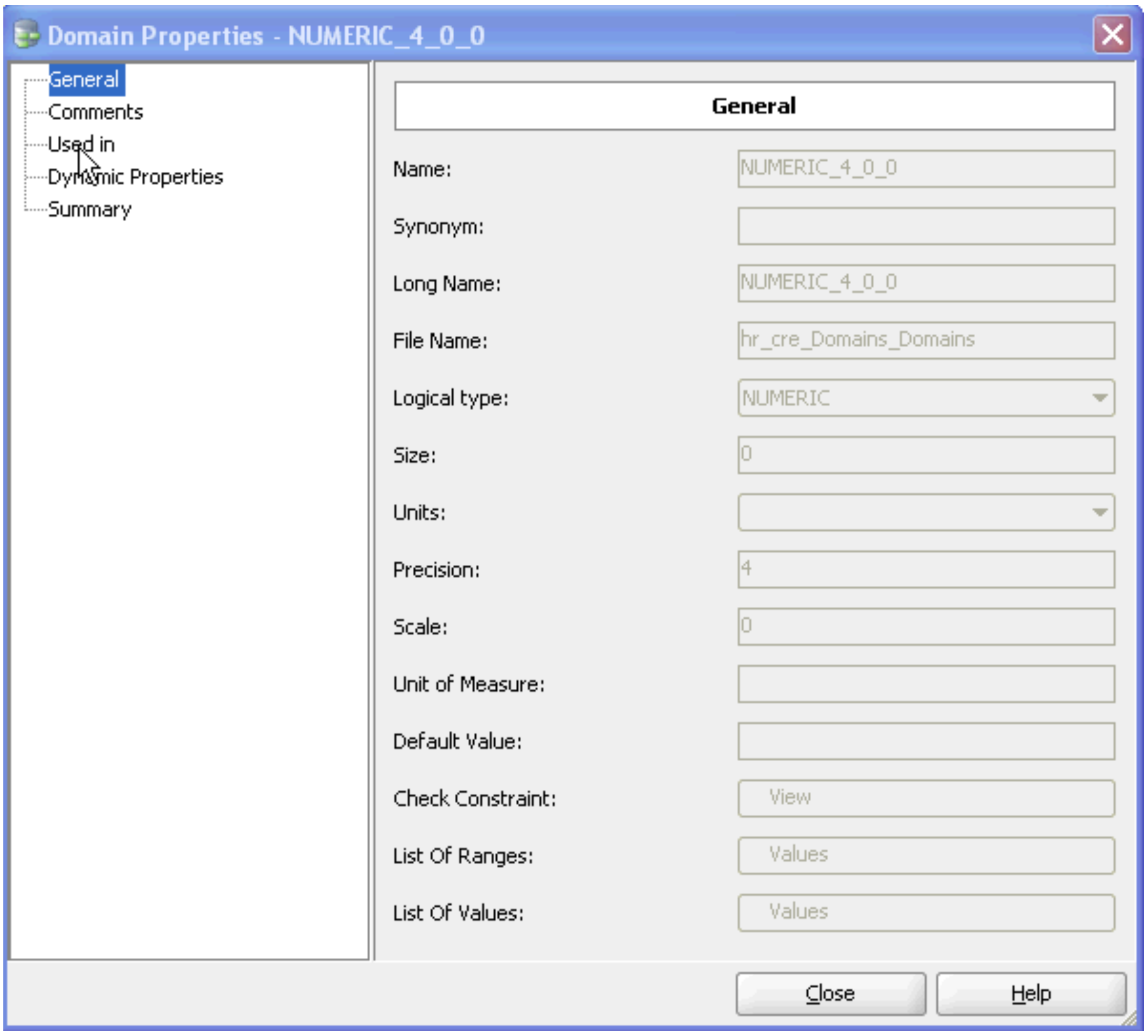
```
jobs
P * job_id             VARCHAR2 (10)
  * job_title          VARCHAR2 (35)
    min_salary         NUMBER (6)
    max_salary         NUMBER (6)
job_id_pk (job_id)
```

```
locations
P * location_id        NUMBER (4)
    street_address     VARCHAR2 (40)
    postal_code        VARCHAR2 (12)
  * city               VARCHAR2 (30)
    state_province     VARCHAR2 (25)
    country_id         CHAR (2)
loc_id_pk (location_id)
```

10 . Note that domains are used in data type definitions - domains like VARCHAR_0_0_20 or NUMERIC_4_0_0 are created c
data types (used in column definitions) are aggregated into domains. These names can be changed. Double-click **NUME**



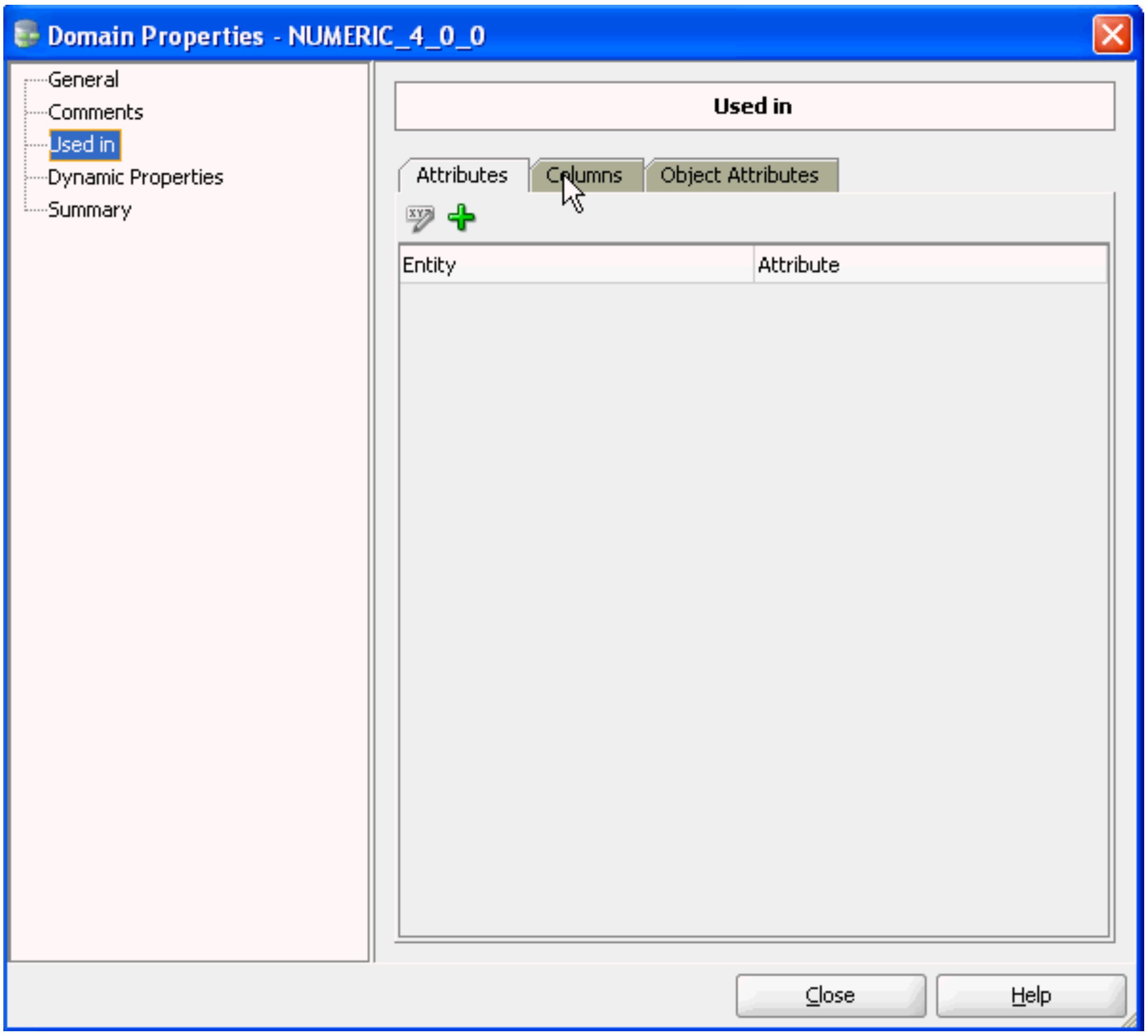
11 . For each domain you can view where it is used through the domain properties dialog. In the left navigator, select**Used in**.


Domain Properties - NUMERIC_4_0_0

General

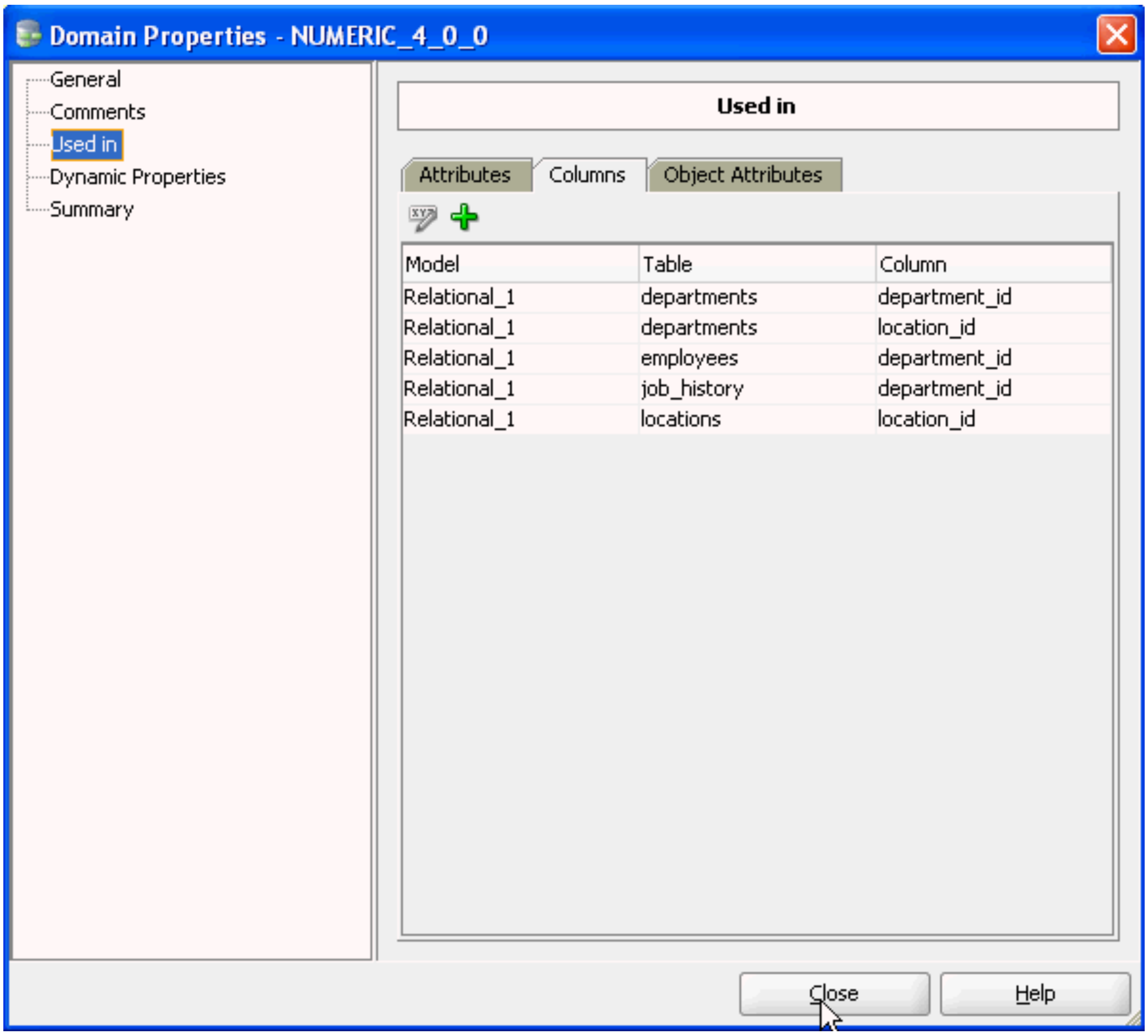
Name:	NUMERIC_4_0_0
Synonym:	
Long Name:	NUMERIC_4_0_0
File Name:	hr_cre_Domains_Domains
Logical type:	NUMERIC
Size:	0
Units:	
Precision:	4
Scale:	0
Unit of Measure:	
Default Value:	
Check Constraint:	View
List Of Ranges:	Values
List Of Values:	Values

12 . Click the **Columns** tab.



Entity	Attribute

13 . You see the list of columns that use this domain. Click **Close**. In the next section, you create a new table to store informa
dependents.

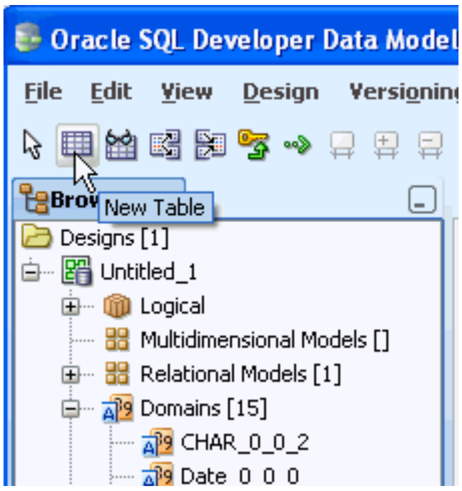


Model	Table	Column
Relational_1	departments	department_id
Relational_1	departments	location_id
Relational_1	employees	department_id
Relational_1	job_history	department_id
Relational_1	locations	location_id

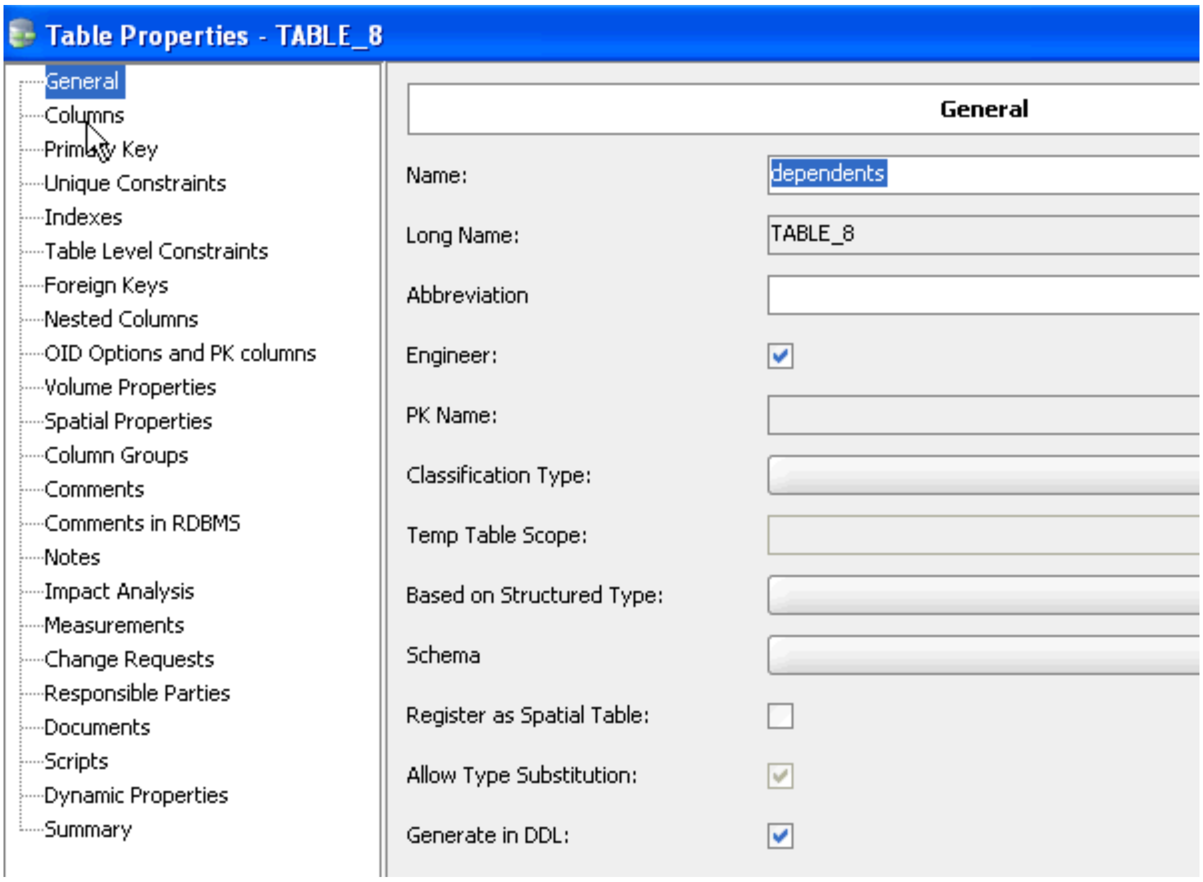
Creating a Table in the Relational Model

In this section, you create a new table called dependents and add a number of columns to the new table. Perform the following steps:

1. Select the New Table icon and click the white space of the diagram.



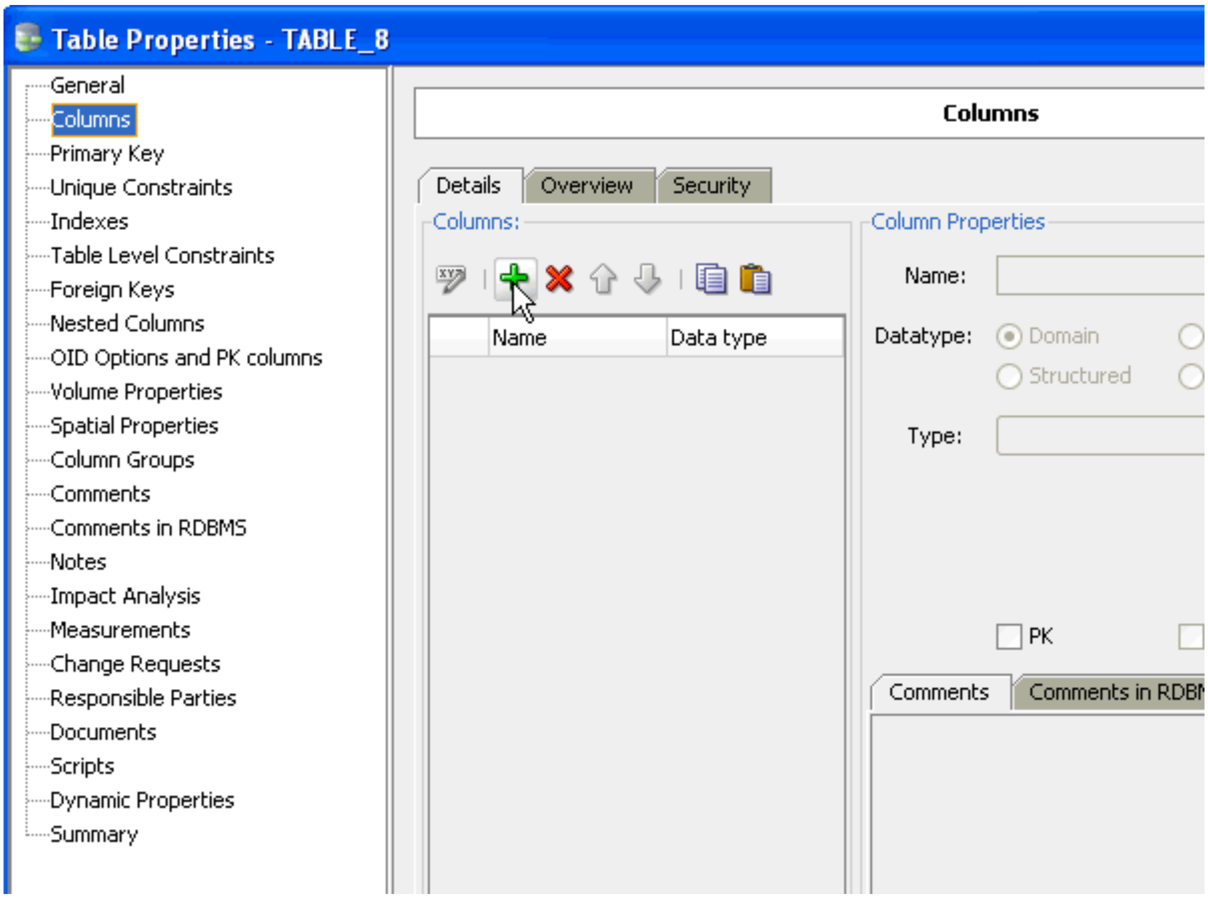
2 . Enter **dependents** for the Name and click **Columns** from the left navigator.



Name:	dependents
Long Name:	TABLE_8
Abbreviation	
Engineer:	✓
PK Name:	
Classification Type:	
Temp Table Scope:	
Based on Structured Type:	
Schema	
Register as Spatial Table:	☐
Allow Type Substitution:	✓
Generate in DDL:	✓

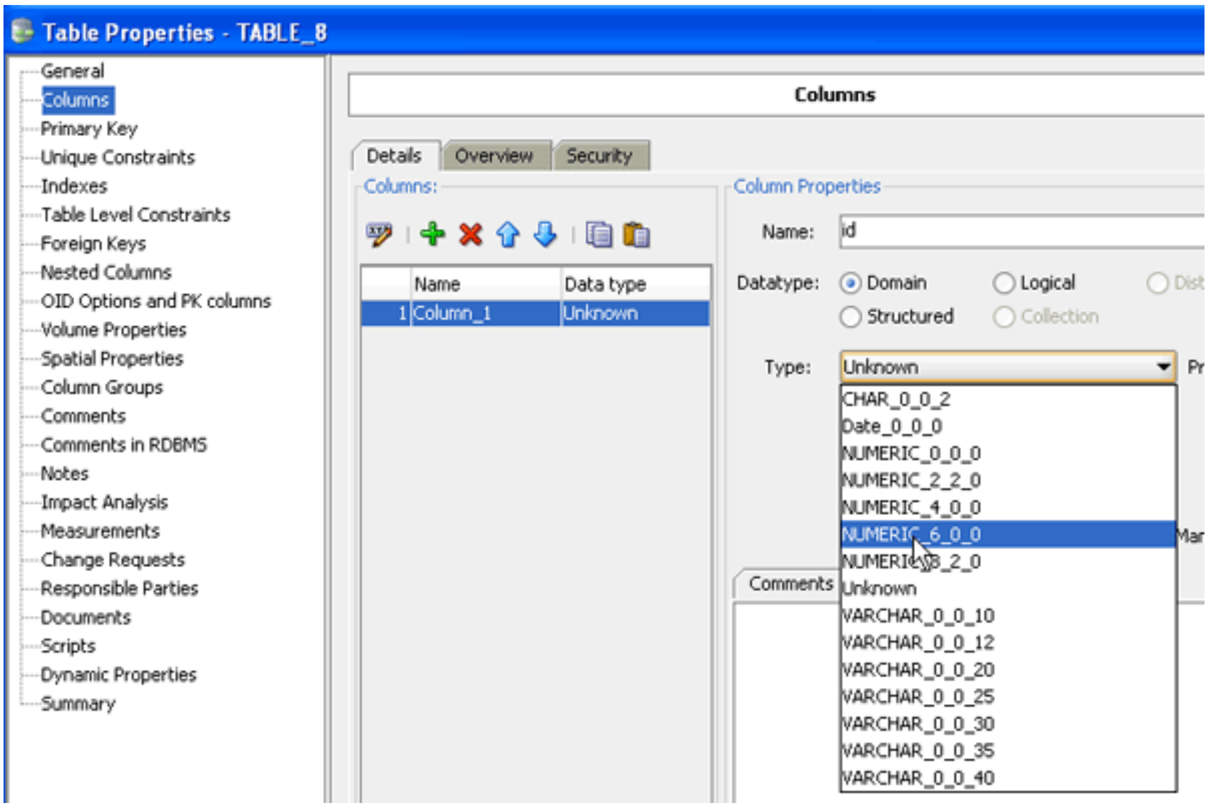
Note that you can click Apply on this page so that the header in the window changes from TABLE_8 to dependents.

3 . Select the create column ✚ icon.



4 . Change the name to **id.**

When you import a DDL file, a domain is created for each datatype in the DDL file. You can select one of these domains length. Select **NUMERIC_6_0_0** from the list.



	Name	Data type
1	Column_1	Unknown

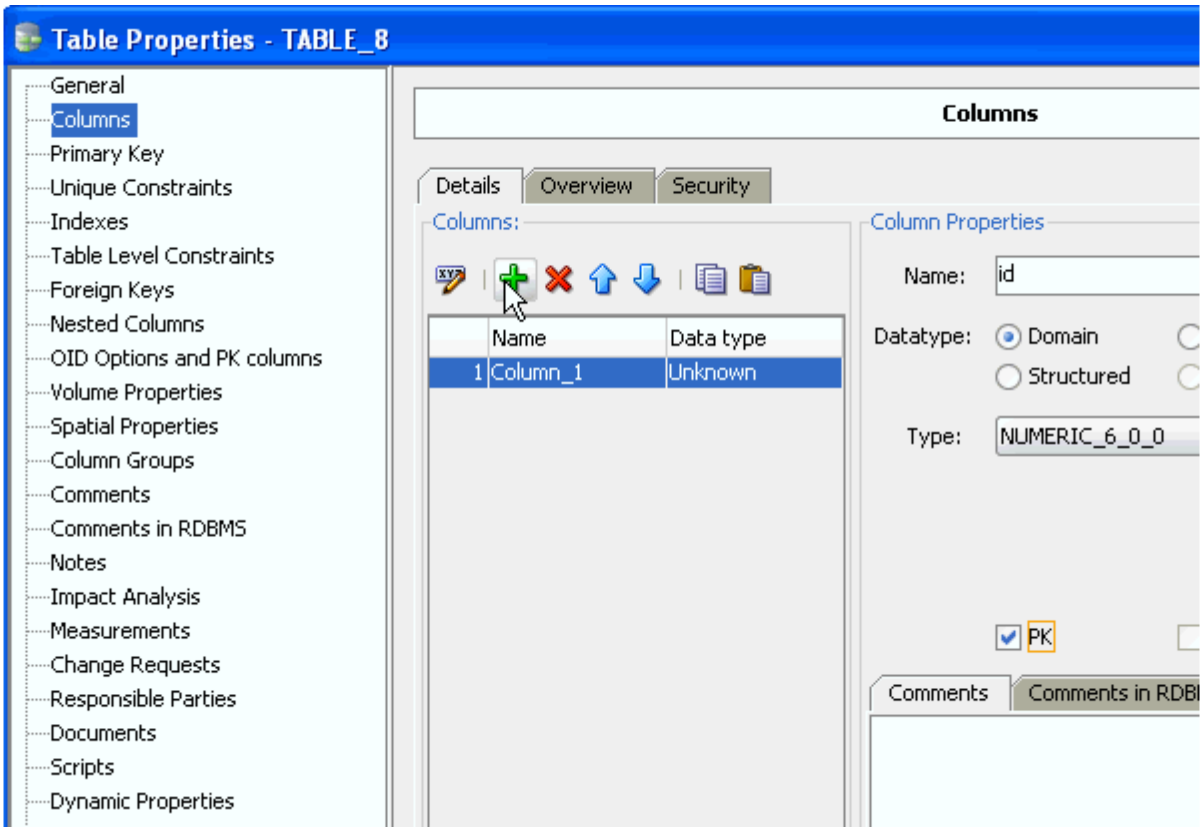
Name: id

Datatype: [✓ Domain] Logical Structured Collection

Type: Unknown

CHAR_0_0_2
Date_0_0_0
NUMERIC_0_0_0
NUMERIC_2_2_0
NUMERIC_4_0_0
NUMERIC_6_0_0
NUMERIC_8_2_0
Unknown
VARCHAR_0_0_10
VARCHAR_0_0_12
VARCHAR_0_0_20
VARCHAR_0_0_25
VARCHAR_0_0_30
VARCHAR_0_0_35
VARCHAR_0_0_40

5 . You want the id column to be the Primary Key. Click the **PK** check box. Then click the Create Column icon to create



	Name	Data type
1	Column_1	Unknown

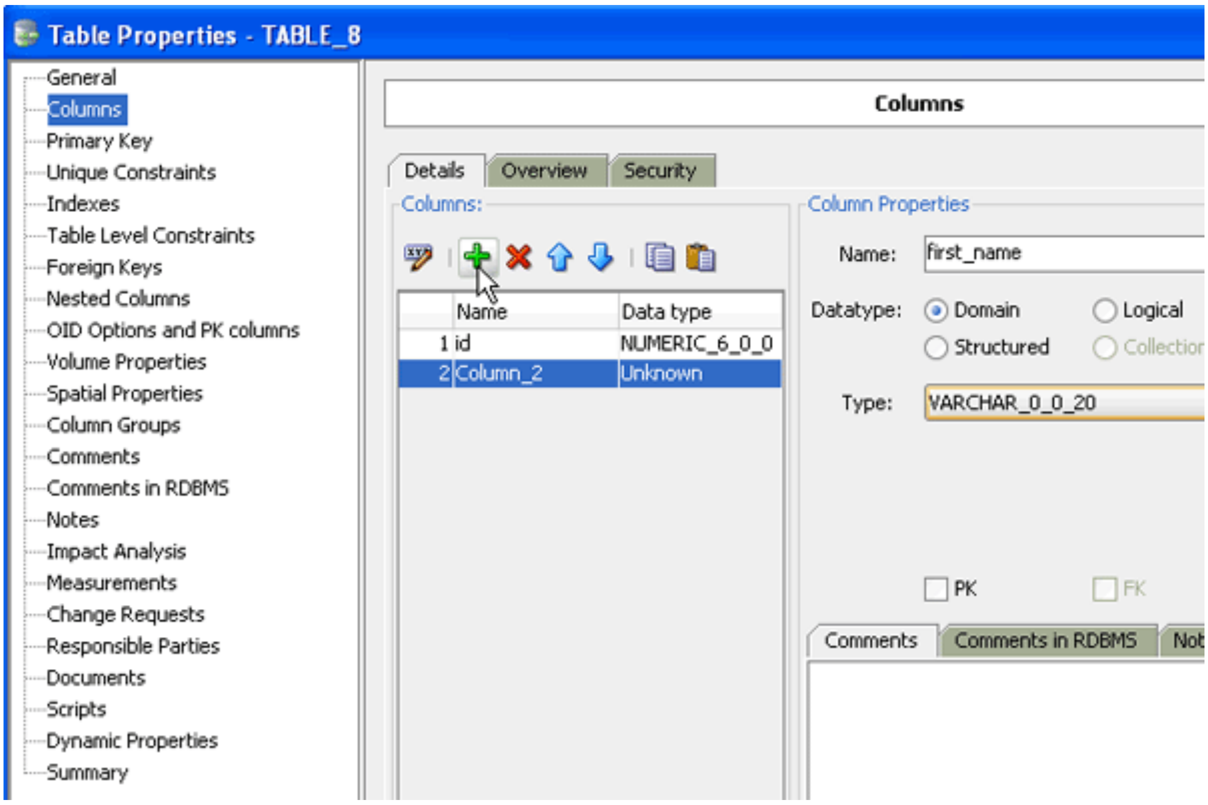
Name: id

Datatype: [✓ Domain] Structured

Type: NUMERIC_6_0_0

[✓ PK]

Enter **first_name** for the Name, select **Domain** for Datatype and select **VARCHAR_0_0_20**. Then click the Create Colur



	Name	Data type
1	id	NUMERIC_6_0_0
2	Column_2	Unknown

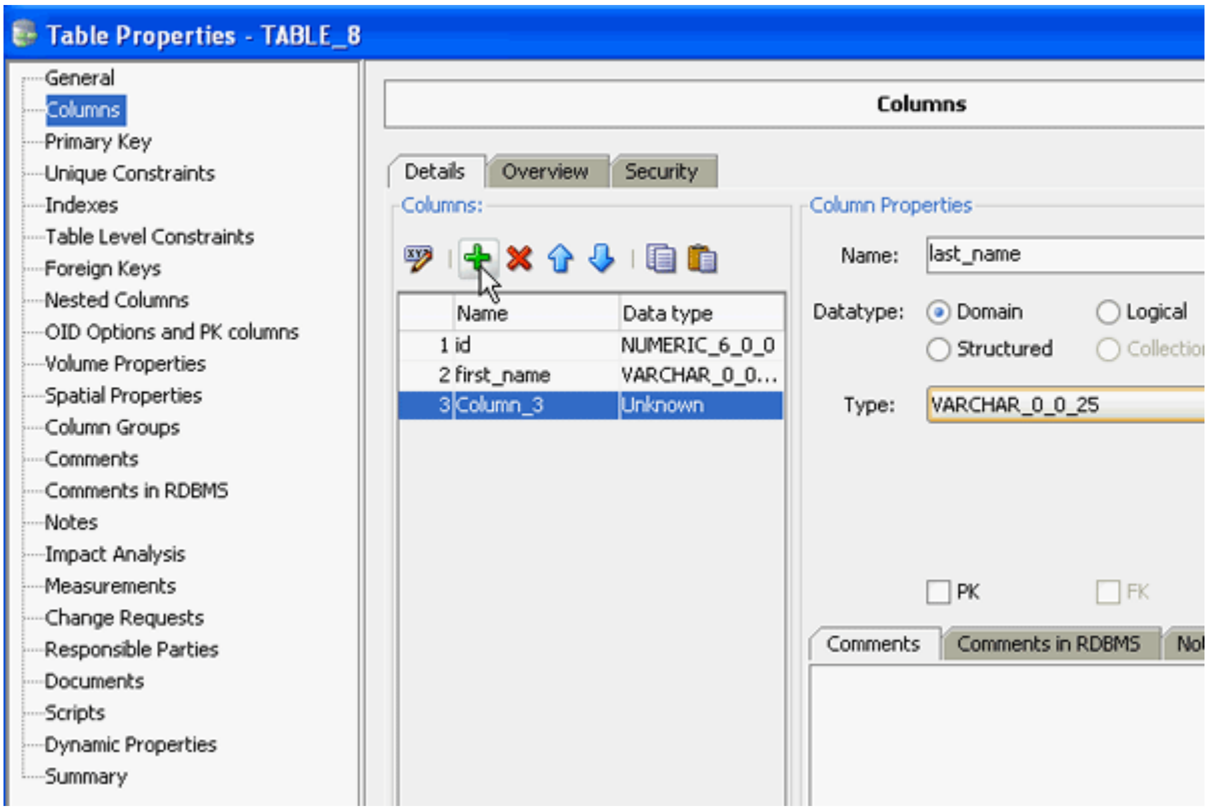
Name: first_name

Datatype: [✓ Domain] Logical Structured Collectior

Type: VARCHAR_0_0_20

PK FK

6 . Enter **last_name** for the Name, select **VARCHAR_0_0_25** for Type and click the Create Column icon again.



	Name	Data type
1	id	NUMERIC_6_0_0
2	first_name	VARCHAR_0_0...
3	Column_3	Unknown

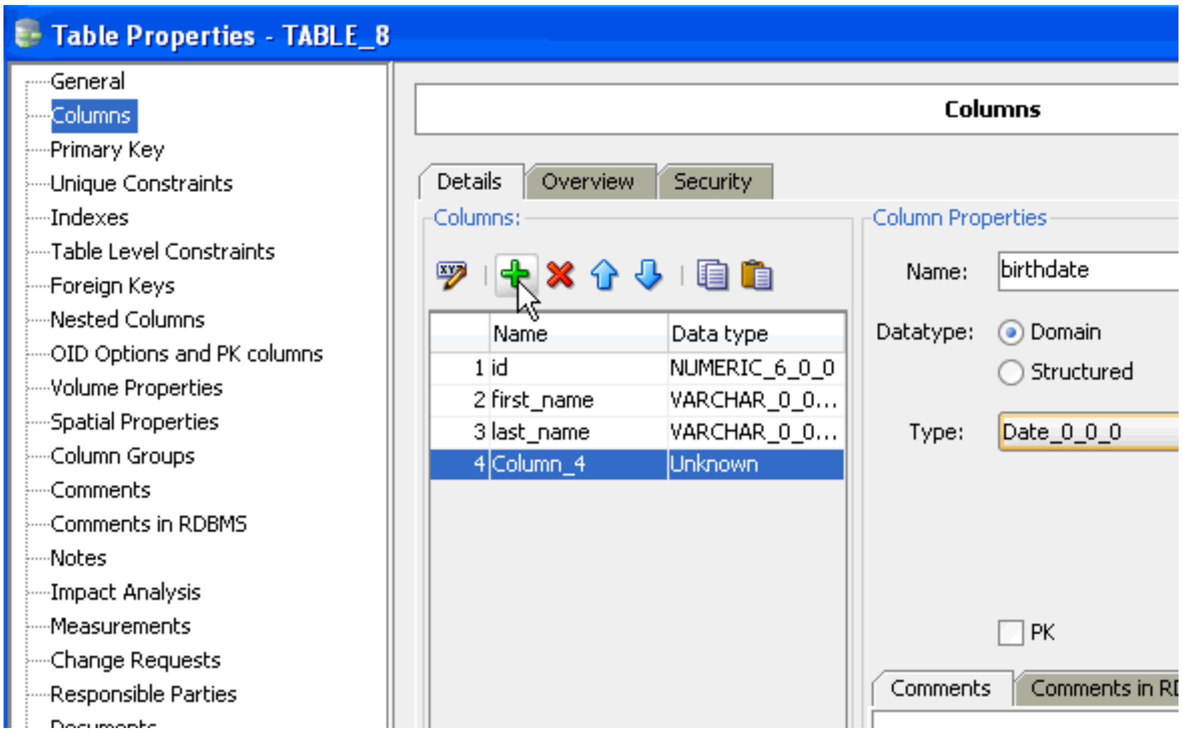
Name: last_name

Datatype: [✓ Domain] Logical Structured Collection

Type: VARCHAR_0_0_25

PK FK

7 . Enter **birthdate** for the Name and select **Date_0_0_0** for Type and click the Create Column icon again.



	Name	Data type
1	id	NUMERIC_6_0_0
2	first_name	VARCHAR_0_0...
3	last_name	VARCHAR_0_0...
4	Column_4	Unknown

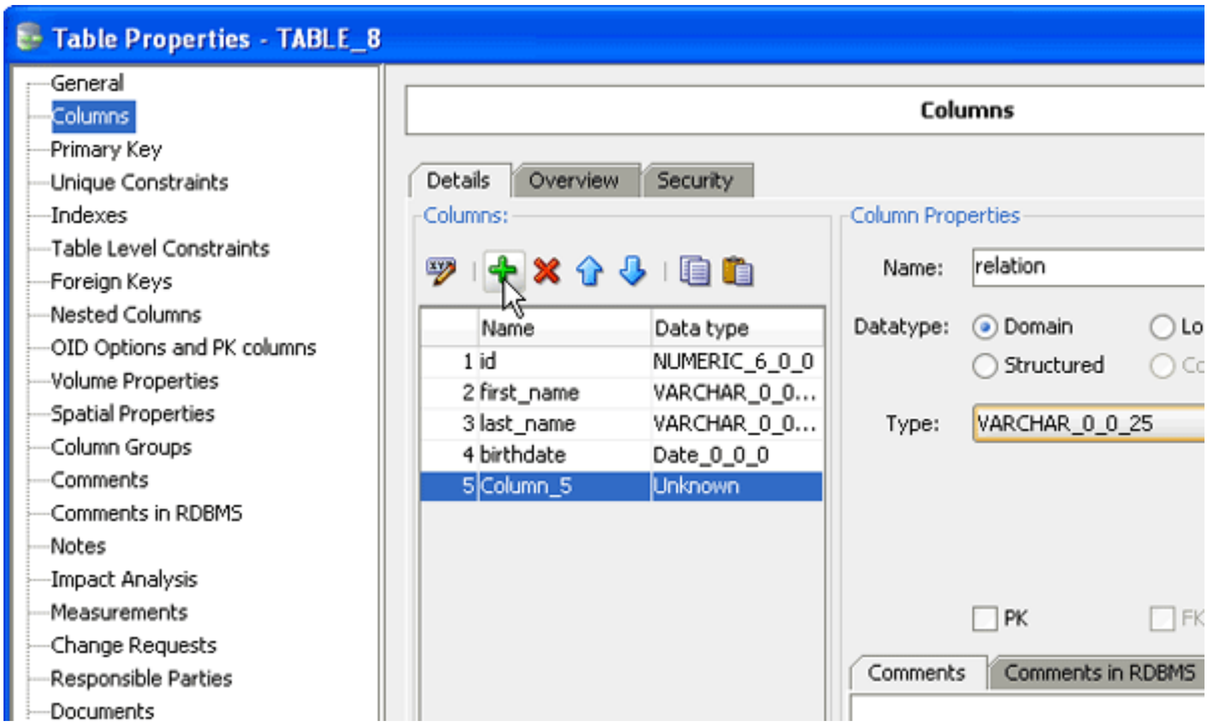
Name: birthdate

Datatype: [✓ Domain] Structured

Type: Date_0_0_0

PK

8 . Enter **relation** for the Name and select **VARCHAR_0_0_25** for Type and click the Create Column icon again.

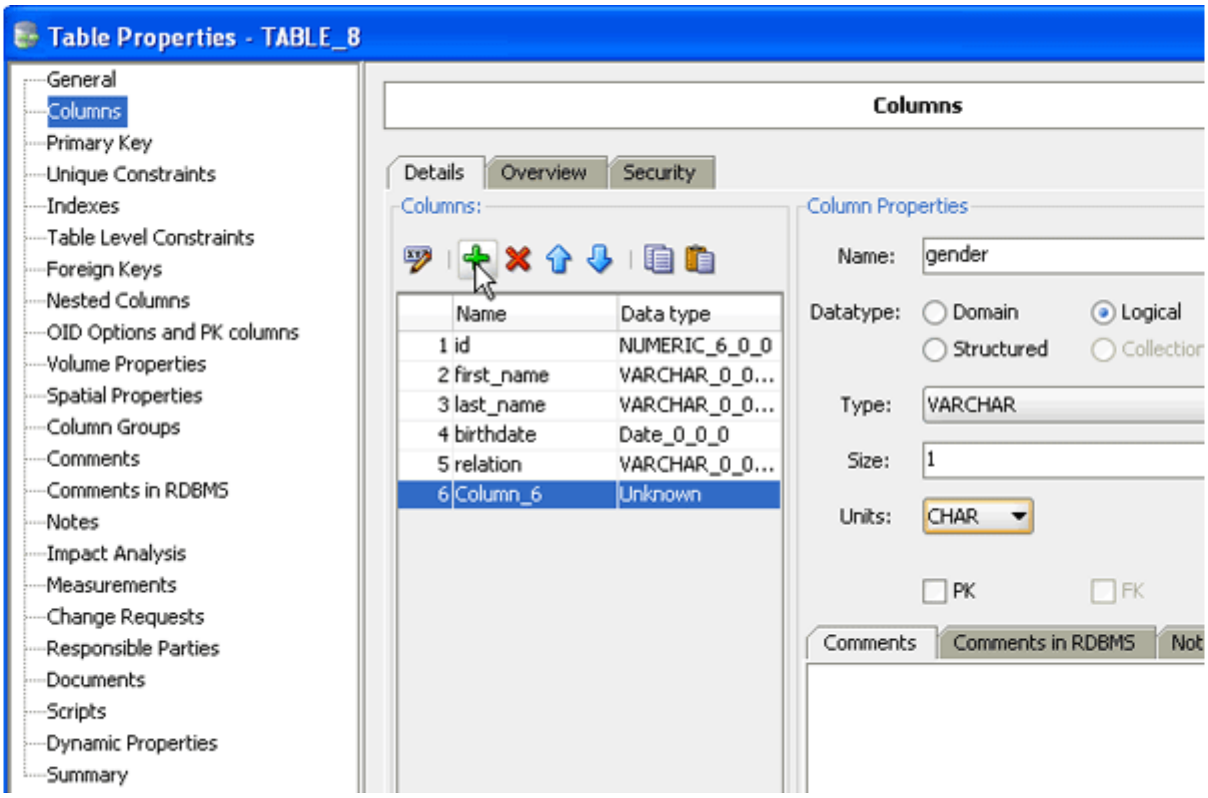


	Name	Data type
1	id	NUMERIC_6_0_0
2	first_name	VARCHAR_0_0...
3	last_name	VARCHAR_0_0...
4	birthdate	Date_0_0_0
5	Column_5	Unknown

Name: relation
Datatype: [✓ Domain] Lo Structured Co
Type: VARCHAR_0_0_25
PK FK

9 . Enter **gender** for the Name. There is no character domain with the length of 1 so you can use logical type. Select**Logica**

select **VARCHAR** from the Type drop list, enter **1** for Size and select **CHAR** for Units. Then click the Create Column



	Name	Data type
1	id	NUMERIC_6_0_0
2	first_name	VARCHAR_0_0...
3	last_name	VARCHAR_0_0...
4	birthdate	Date_0_0_0
5	relation	VARCHAR_0_0...
6	Column_6	Unknown

Name: gender
Datatype: Domain [✓ Logical] Structured Collection
Type: VARCHAR
Size: 1
Units: CHAR
PK FK

10 . Enter **relative_id** for the Name, select **Domain** for Datatype and select **NUMERIC_6_0_0** from the drop list. Then click **C**

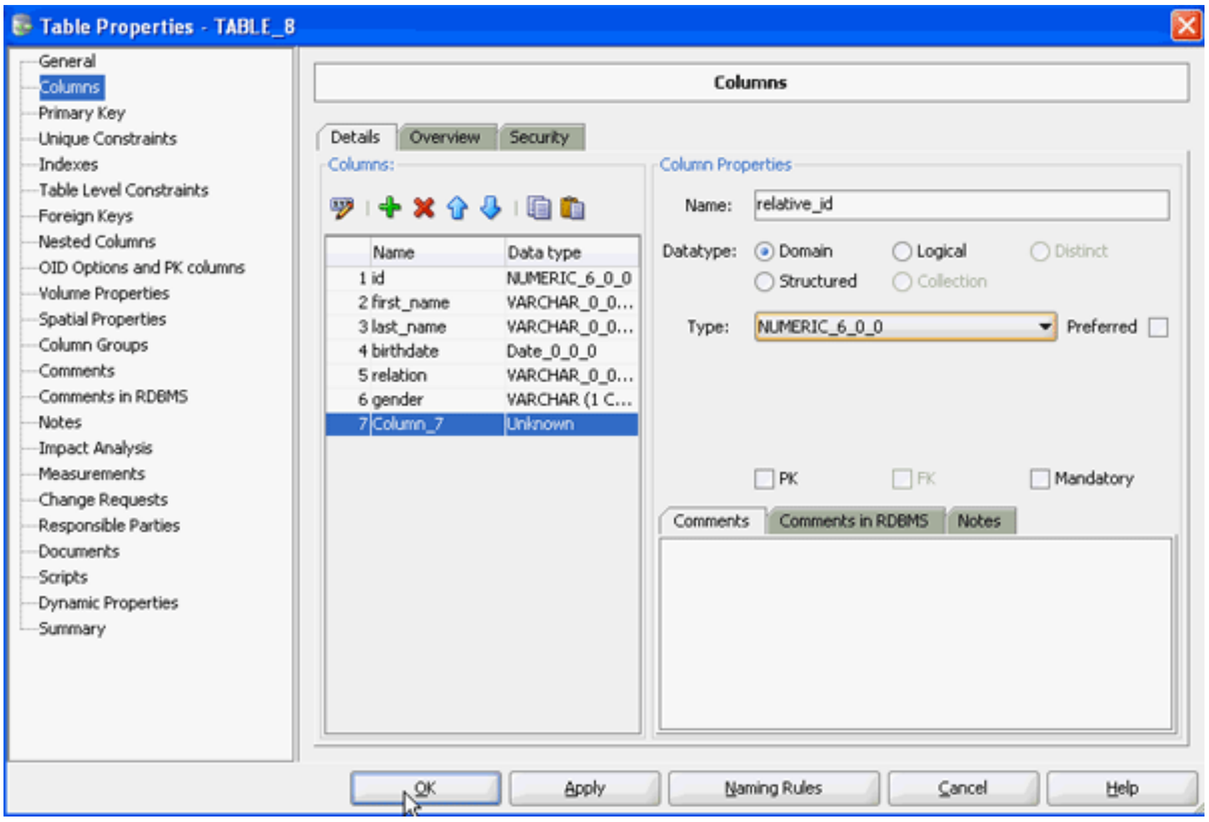


Table Properties - TABLE_8

- General
- Columns
- Primary Key
- Unique Constraints
- Indexes
- Table Level Constraints
- Foreign Keys
- Nested Columns
- OID Options and PK columns
- Volume Properties
- Spatial Properties
- Column Groups
- Comments
- Comments in RDBMS
- Notes
- Impact Analysis
- Measurements
- Change Requests
- Responsible Parties
- Documents
- Scripts
- Dynamic Properties
- Summary

Columns

Details | Overview | Security

Columns:

	Name	Data type
1	id	NUMERIC_6_0_0
2	first_name	VARCHAR_0_0...
3	last_name	VARCHAR_0_0...
4	birthdate	Date_0_0_0
5	relation	VARCHAR_0_0...
6	gender	VARCHAR (1 C...
7	Column_7	Unknown

Column Properties

Name: relative_id

Datatype: [✓ Domain] Logical Distinct Structured Collection

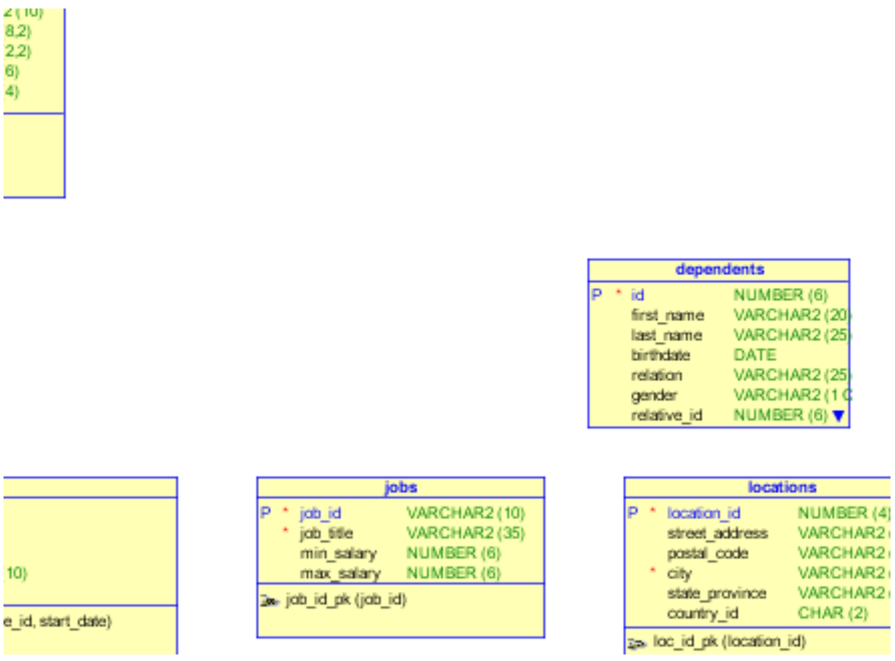
Type: NUMERIC_6_0_0 Preferred

PK FK Mandatory

Comments | Comments in RDBMS | Notes

OK | Apply | Naming Rules | Cancel | Help

11 . The dependents table was created successfully. In the next section, you add a foreign key between employees anddepe


dependents

P	*	id	NUMBER (6)
		first_name	VARCHAR2 (20)
		last_name	VARCHAR2 (25)
		birthdate	DATE
		relation	VARCHAR2 (25)
		gender	VARCHAR2 (1 C
		relative_id	NUMBER (6)

jobs

P	*	job_id	VARCHAR2 (10)
	*	job_title	VARCHAR2 (35)
		min_salary	NUMBER (6)
		max_salary	NUMBER (6)

job_id_pk (job_id)

locations

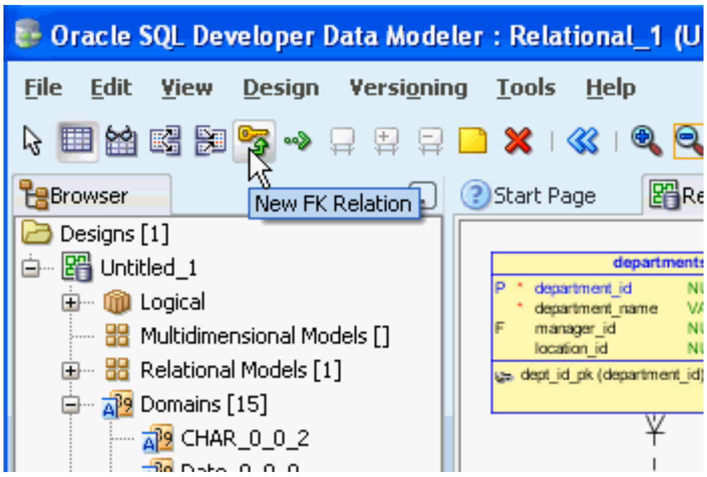
P	*	location_id	NUMBER (4)
		street_address	VARCHAR2
		postal_code	VARCHAR2
	*	city	VARCHAR2
		state_province	VARCHAR2
		country_id	CHAR (2)

loc_id_pk (location_id)

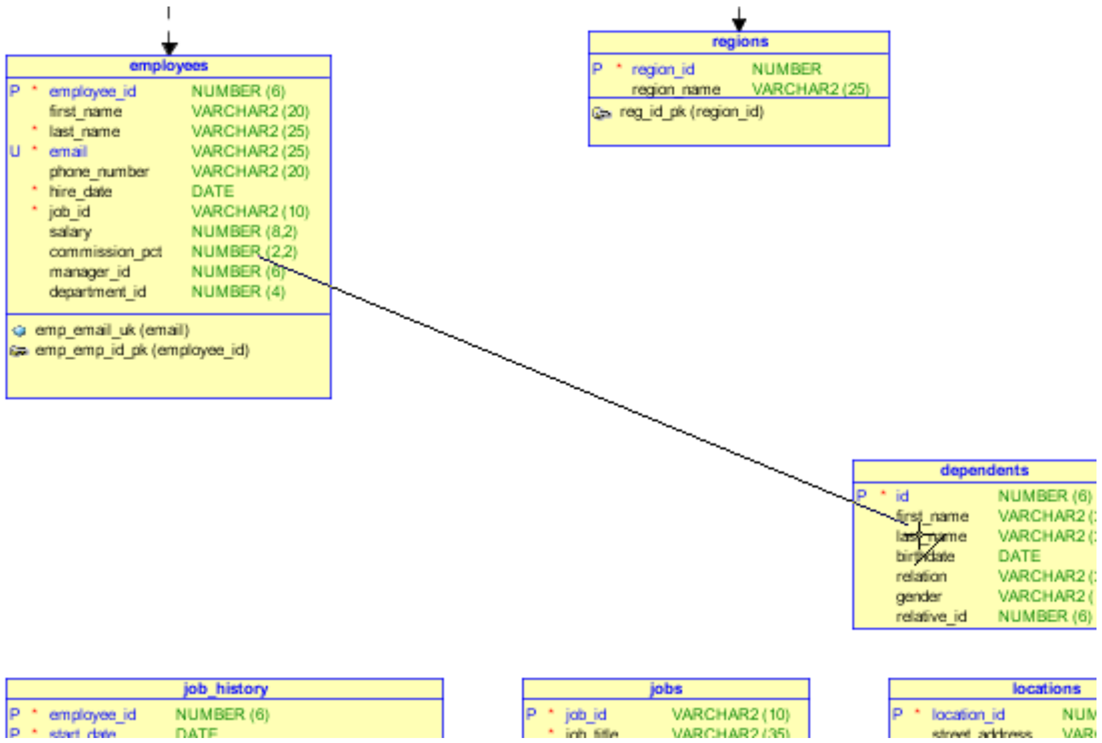
Adding a Foreign Key Between Tables

In this section, you add a foreign key between the employees and dependents tables. Perform the following steps:

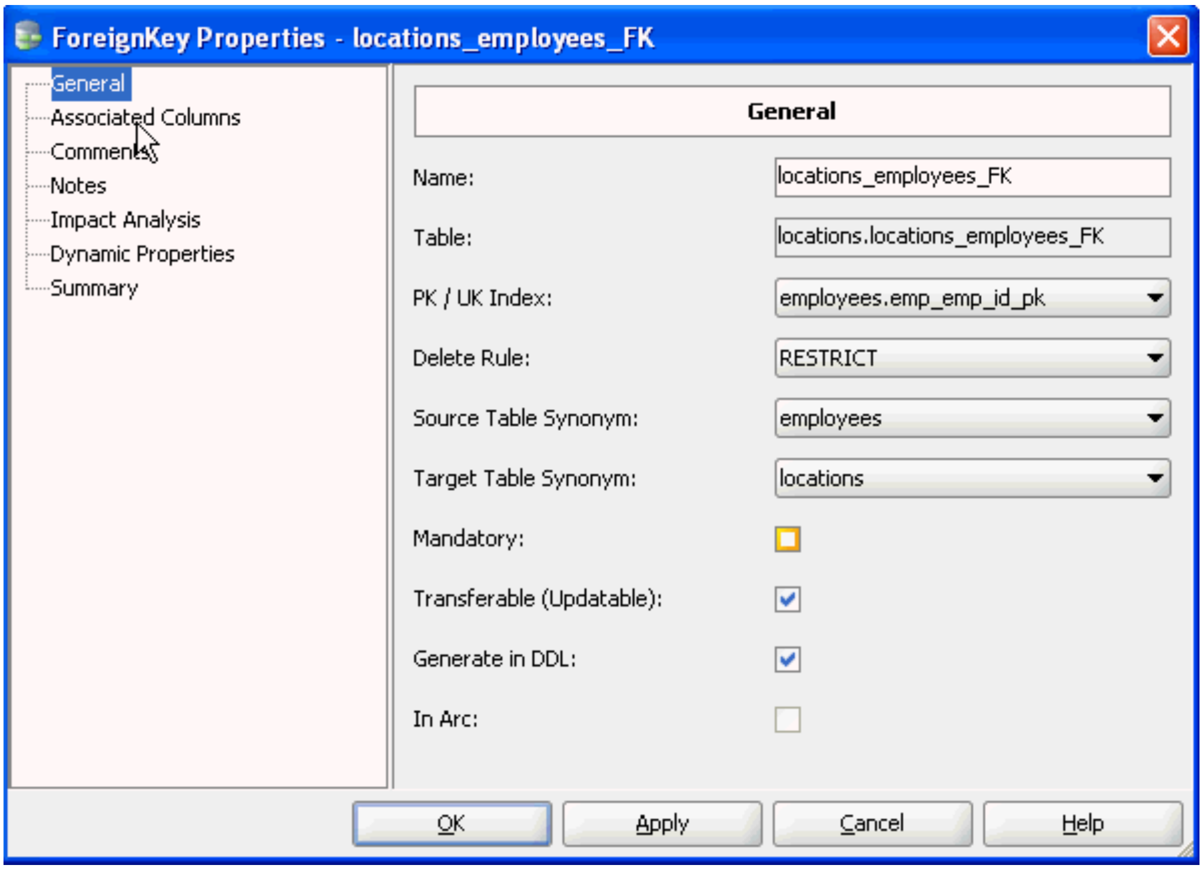
1 . Select the New FK Relation icon.



2 . Select the **employees** table then the **dependents** table to create the new FK relation.



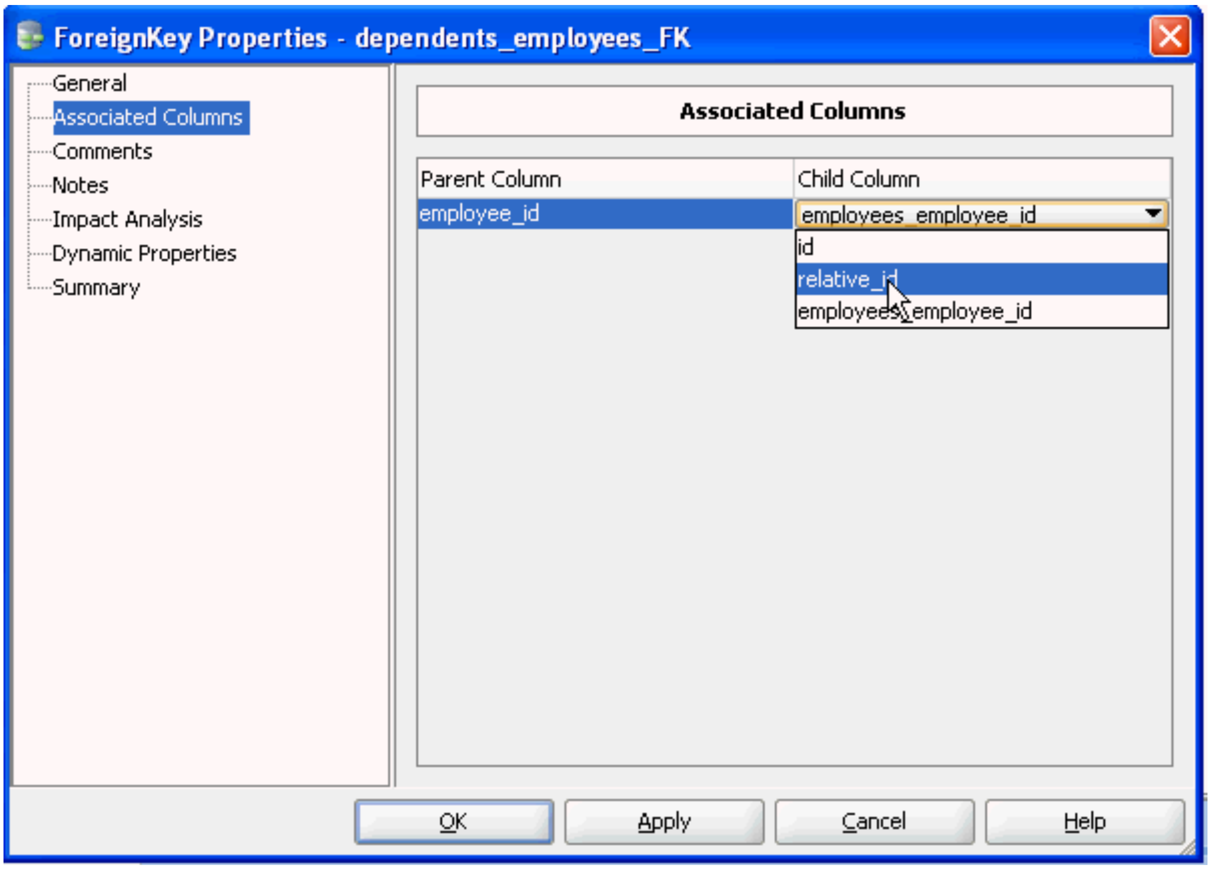
3 . Not all dependents have an assigned employee, so you want to change the relation to an optional relation. Deselect the
select **Associated Columns** in the left navigator.



General

Name:	locations_employees_FK
Table:	locations.locations_employees_FK
PK / UK Index:	employees.emp_emp_id_pk
Delete Rule:	RESTRICT
Source Table Synonym:	employees
Target Table Synonym:	locations
Mandatory:	☐
Transferable (Updatable):	✓
Generate in DDL:	✓
In Arc:	☐

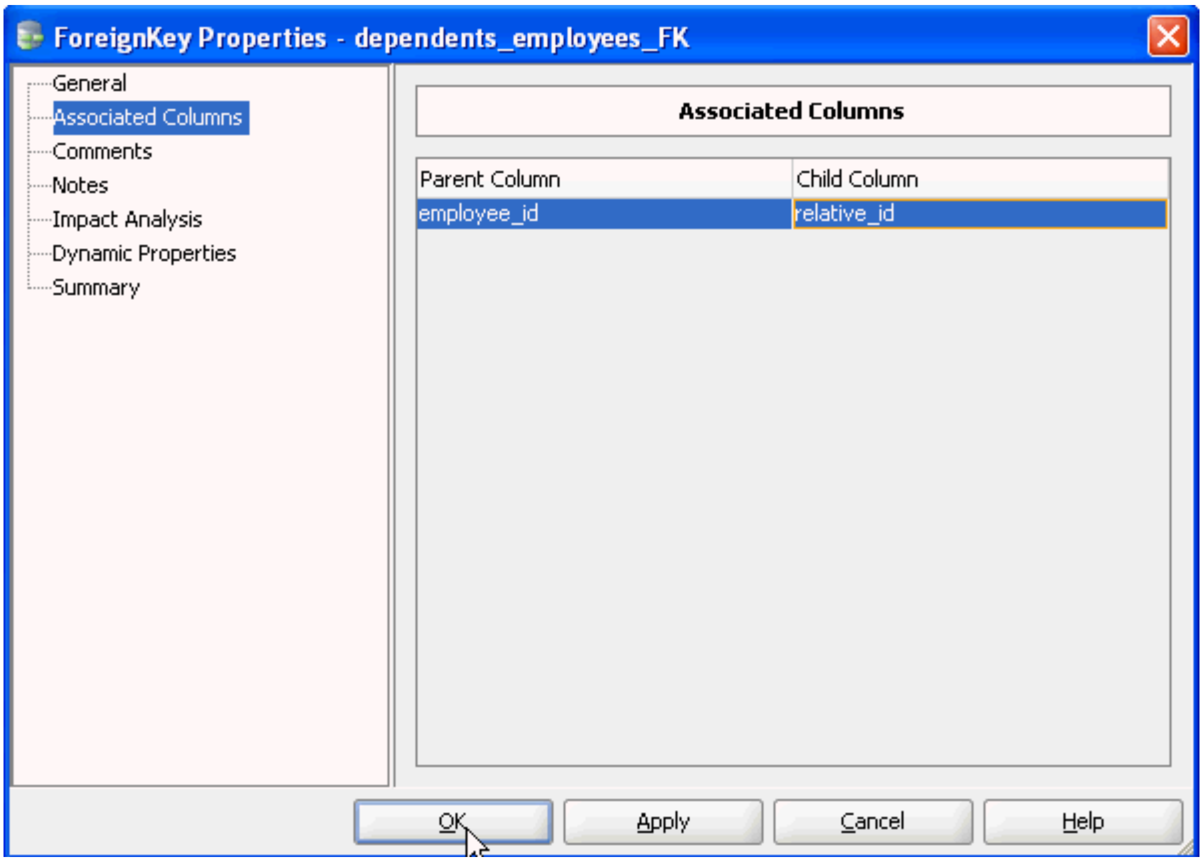
4 . Select **relative_id** for the Child Column.


ForeignKey Properties - dependents_employees_FK

Associated Columns

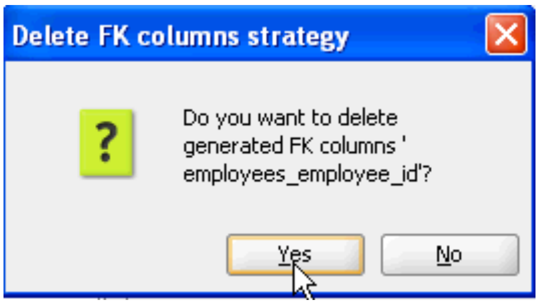
Parent Column	Child Column
employee_id	employees_employee_id

5 . Click **OK**.



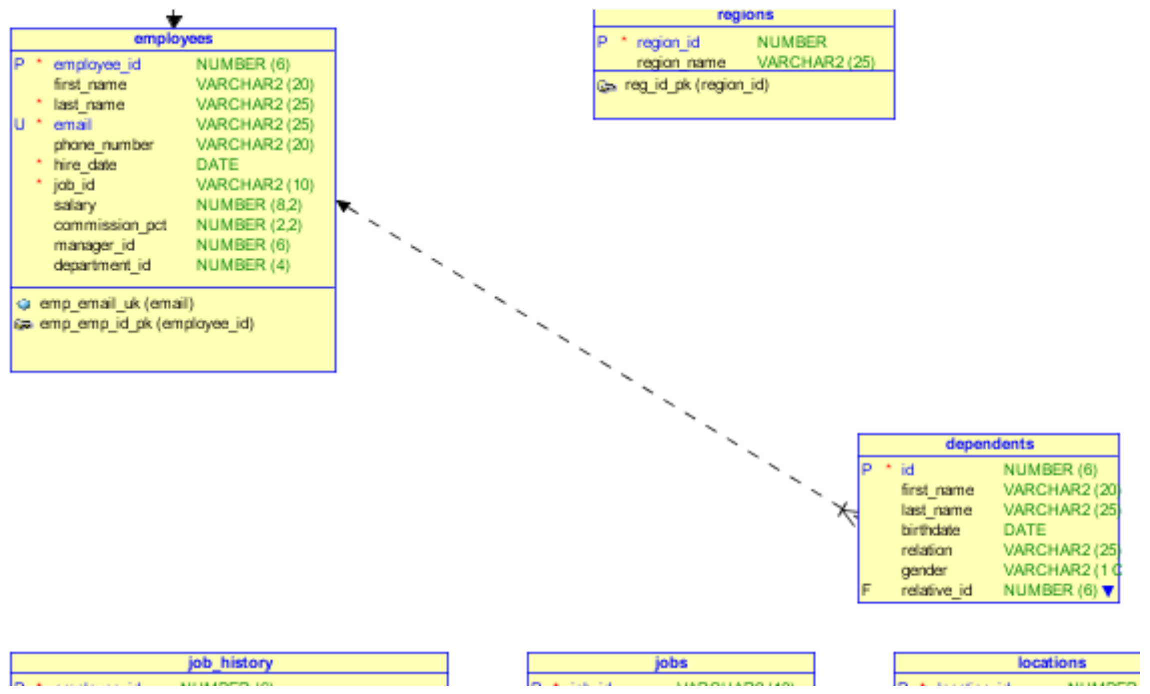
Parent Column	Child Column
employee_id	relative_id

6 . Since you want to use an existing column for the FK instead of the generated column, Click **Yes** to delete the generated



Do you want to delete generated FK columns ' employees_employee_id'?

7 . Notice that the relative_id column now has an F next to it indicating that it is the foreign key column and the foreign key r
the dotted line).

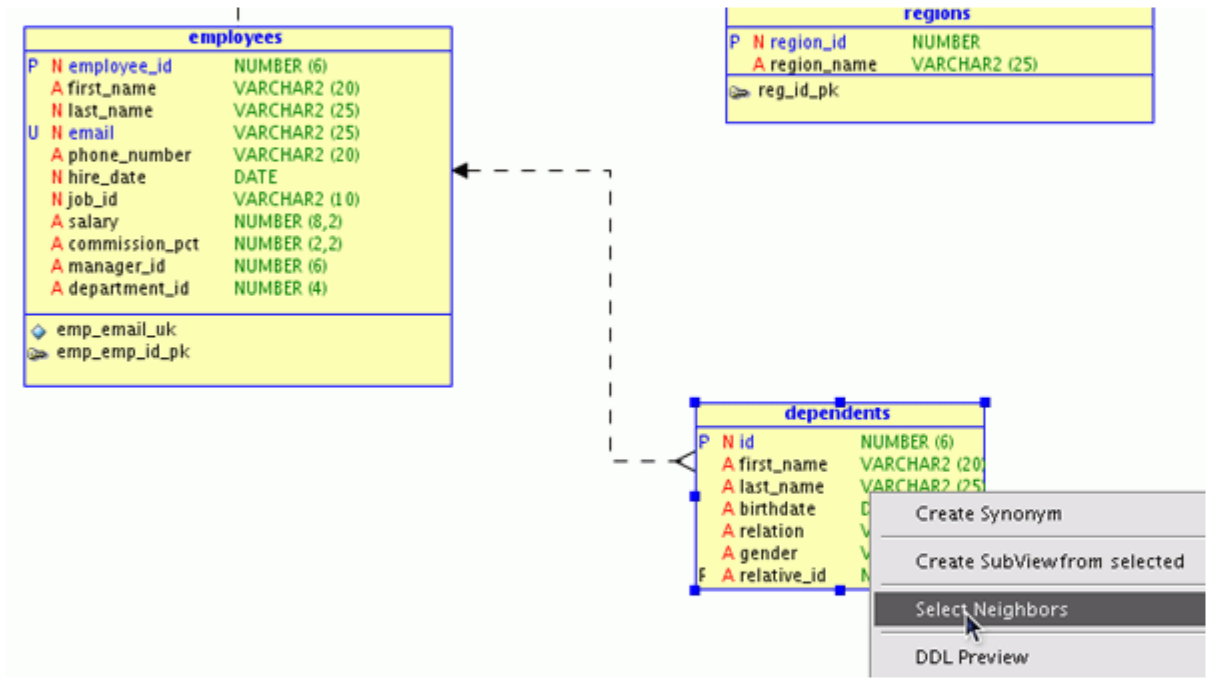


In the next section, you create a subview.

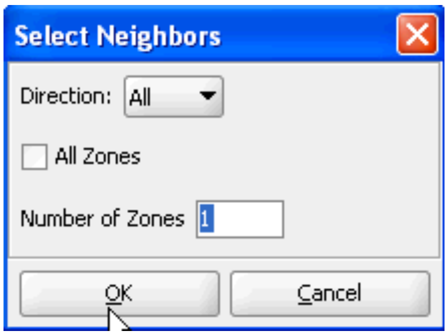
Creating a SubView

In this section, you create a subview with the employees and dependents tables. Perform the following steps:

1 . There are several possible ways to create subset of the tables you are interested in. One way is to create a subview of ta another. Right-click the **dependents** table and select **Select Neighbors**.



2 . Accept the default of **1** zone and click **OK**.

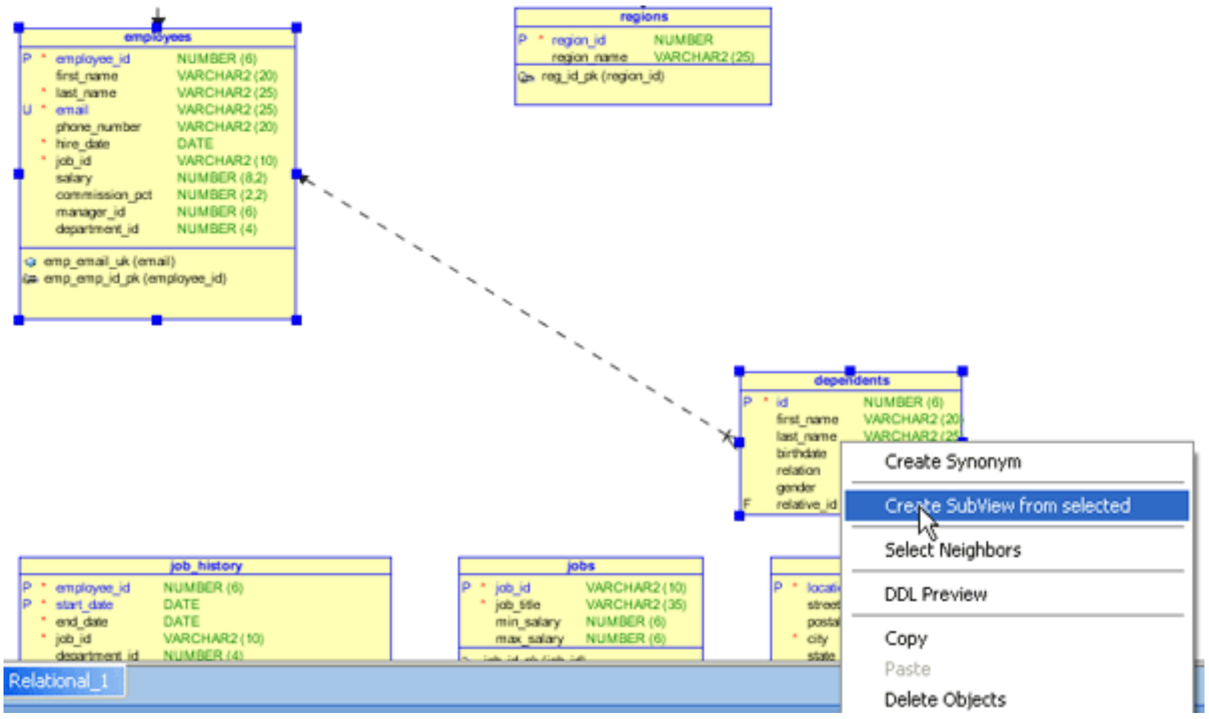

Select Neighbors

Direction: All

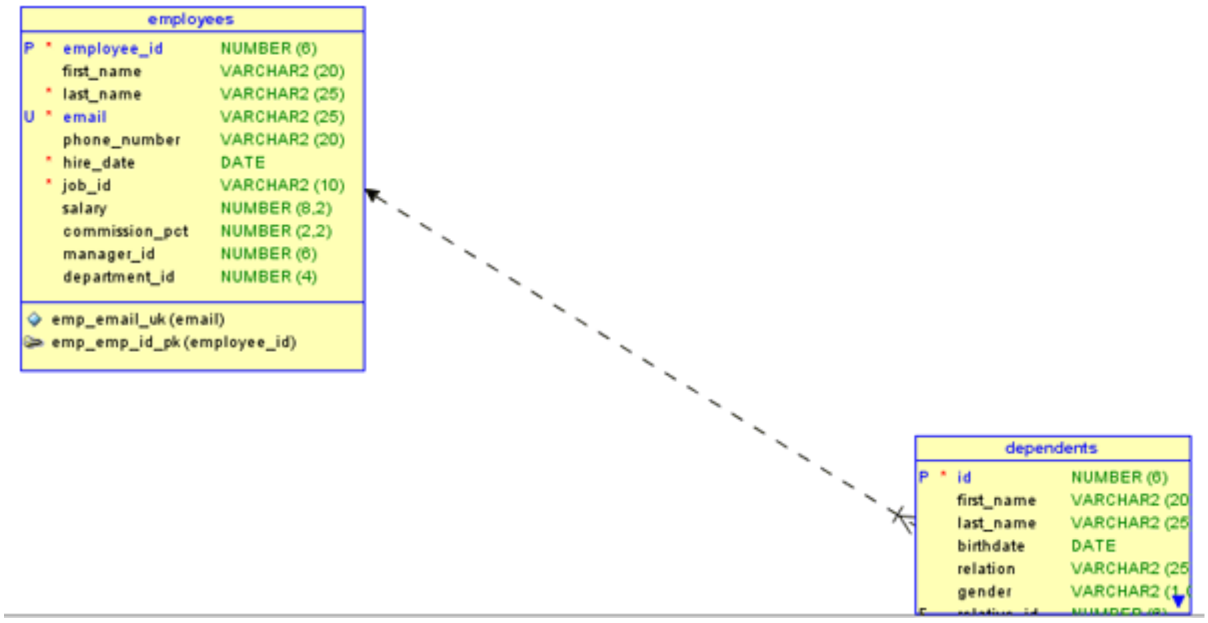
All Zones

Number of Zones 1

3 . Notice that the FK relation and employees table are now selected because they are the neighbors of the dependents tab
the **dependents** table again and select **Create SubView from selected**.



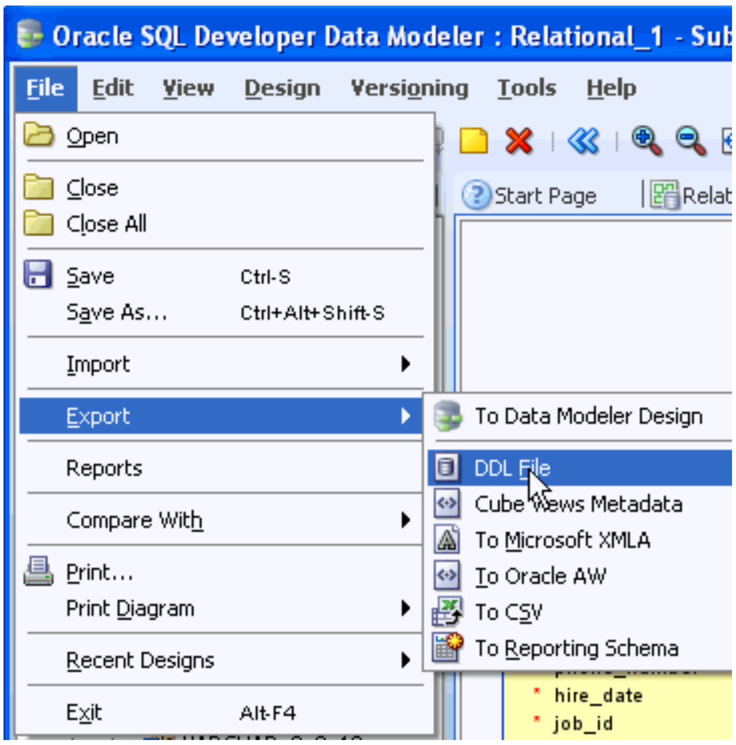
4 . The subview is created. You can view the list of objects in the subview from the browser window by expanding the object



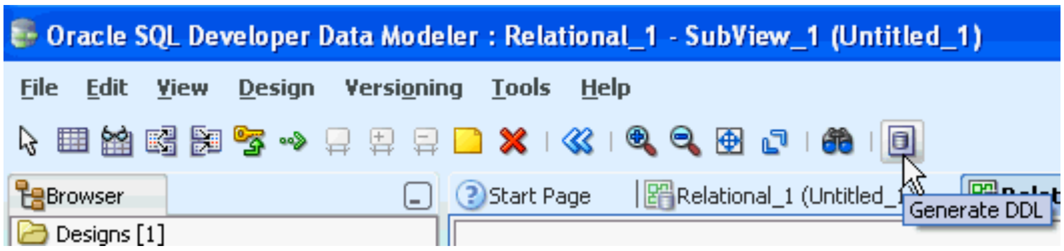
Generate the DDL for the New Table

In this section, you create and export the DDL for the dependents table. Perform the following steps:

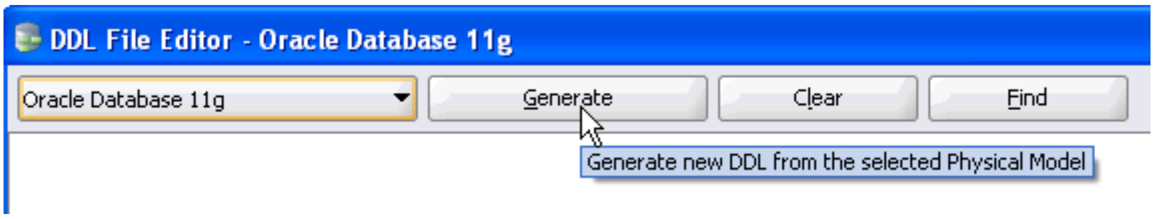
1 . Select **File** > **Export** > **DDL File**.



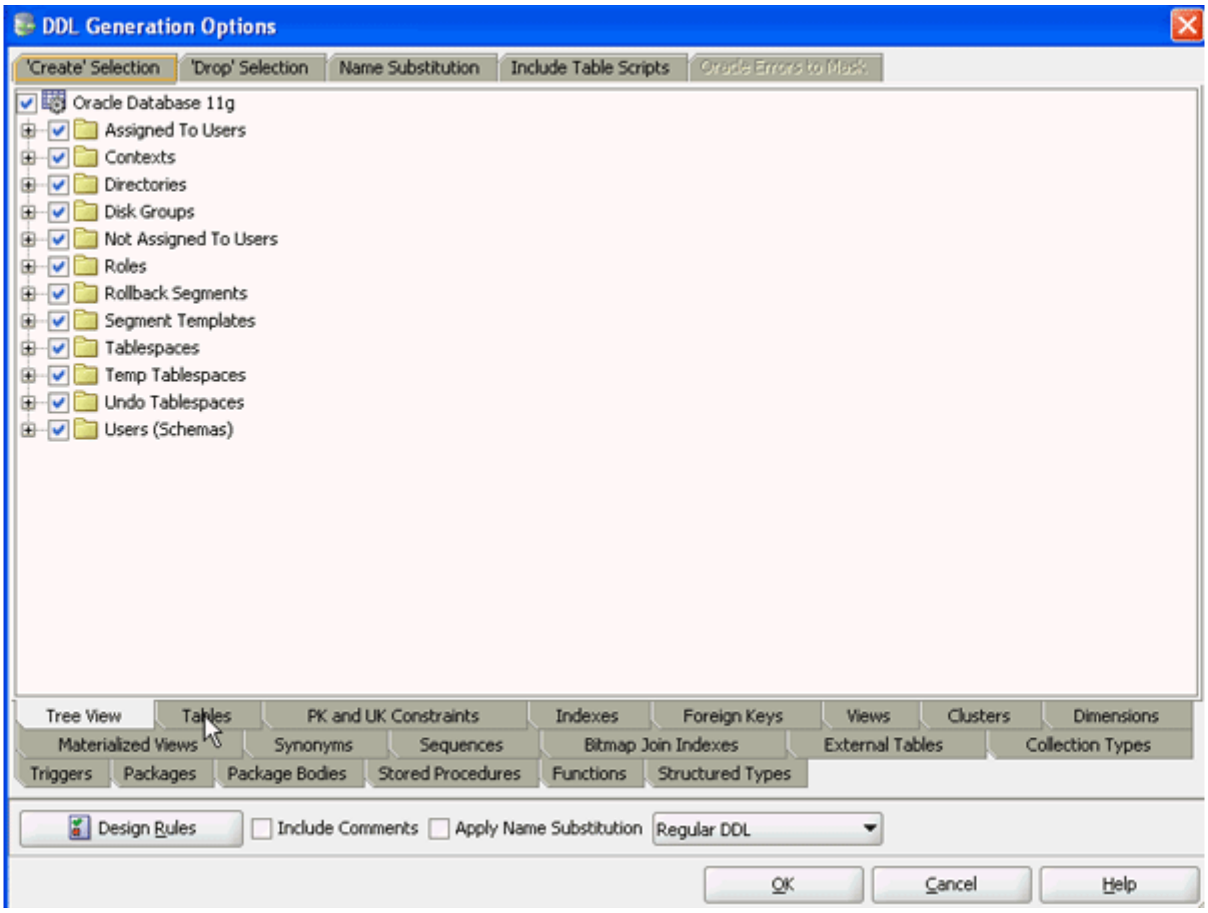
OR, you can select the Generate DDL icon.


Oracle SQL Developer Data Modeler : Relational_1 - SubView_1 (Untitled_1)
File Edit View Design Versioning Tools Help
Browser
Designs [1]
Start Page
Relational_1 (Untitled_1)
Generate DDL

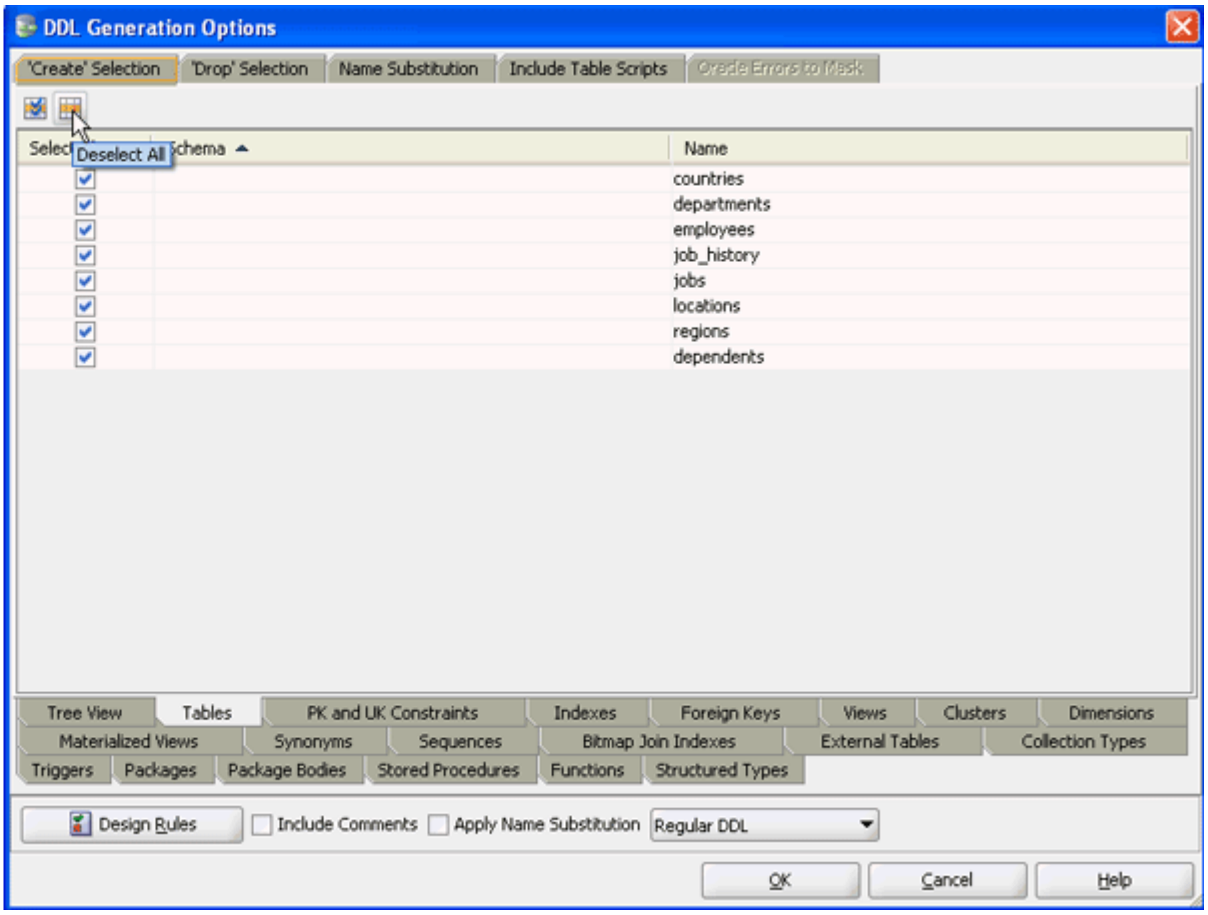
2 . Click **Generate**.


DDL File Editor - Oracle Database 11g
Oracle Database 11g
Generate
Clear
Find
Generate new DDL from the selected Physical Model

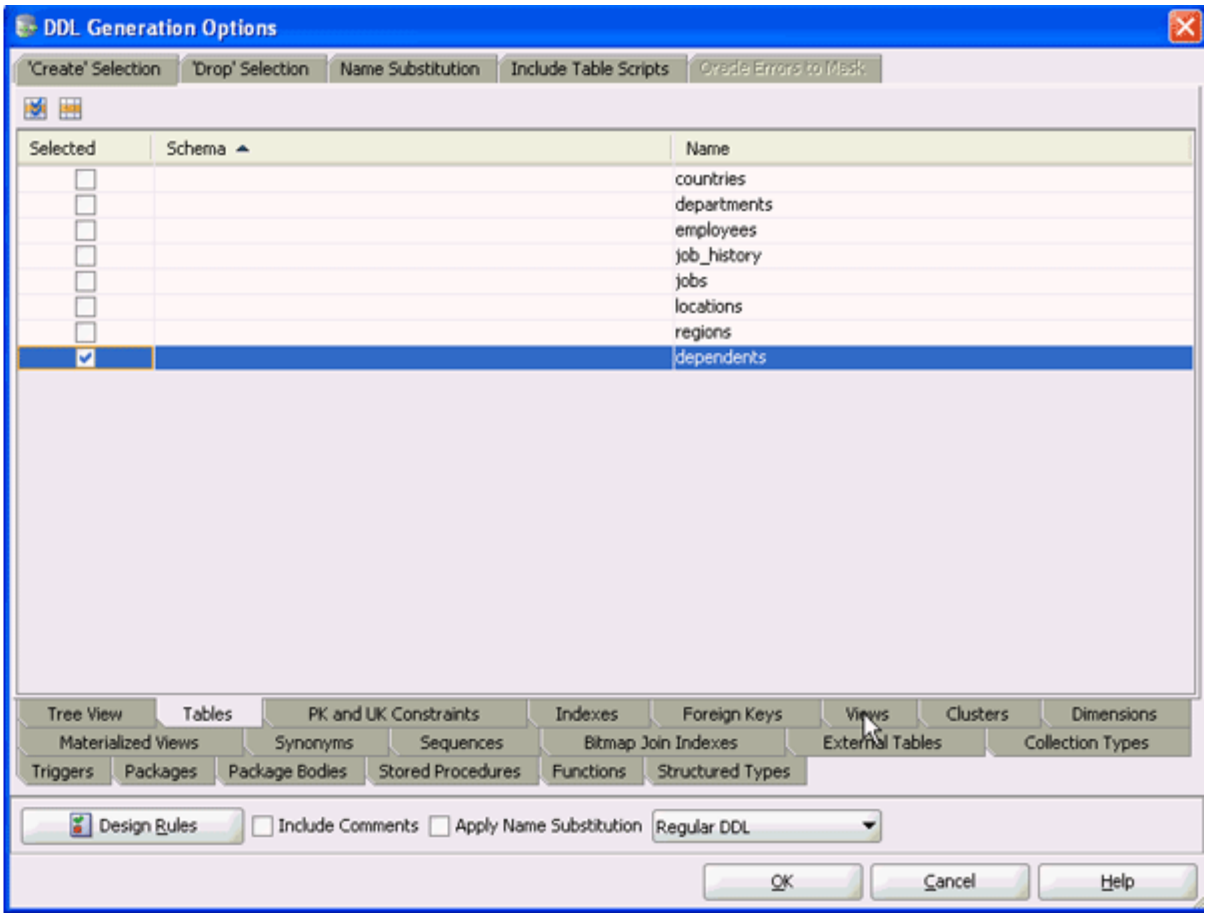
3 . Select the **Tables** tab.



4 . You only want to generate the DDL for dependents. Click the **Deselect All** icon.



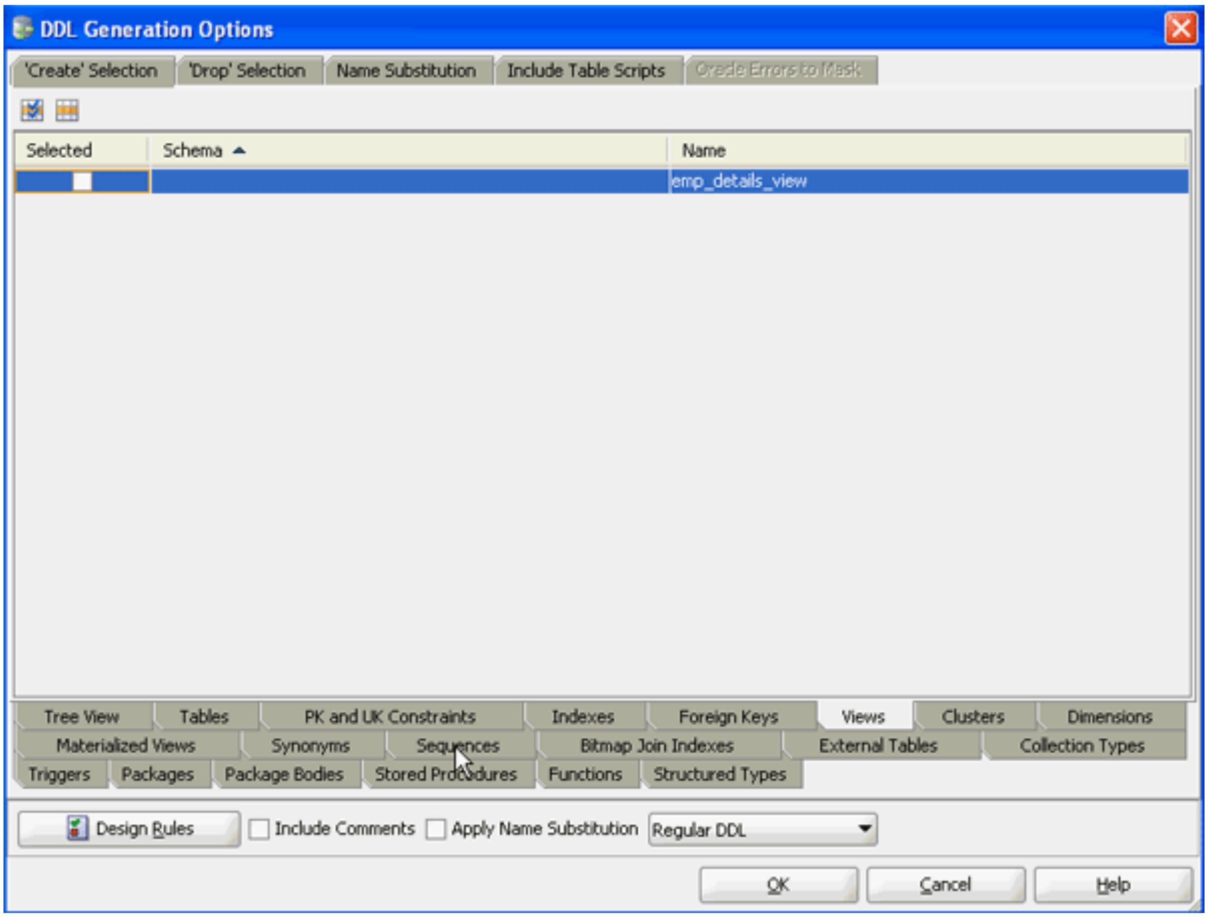
5 . Select **dependents**. Then click the **Views** tab.



Selected	Schema	Name
☐		countries
☐		departments
☐		employees
☐		job_history
☐		jobs
☐		locations
☐		regions
✓		dependents

☐ Include Comments ☐ Apply Name Substitution Regular DDL

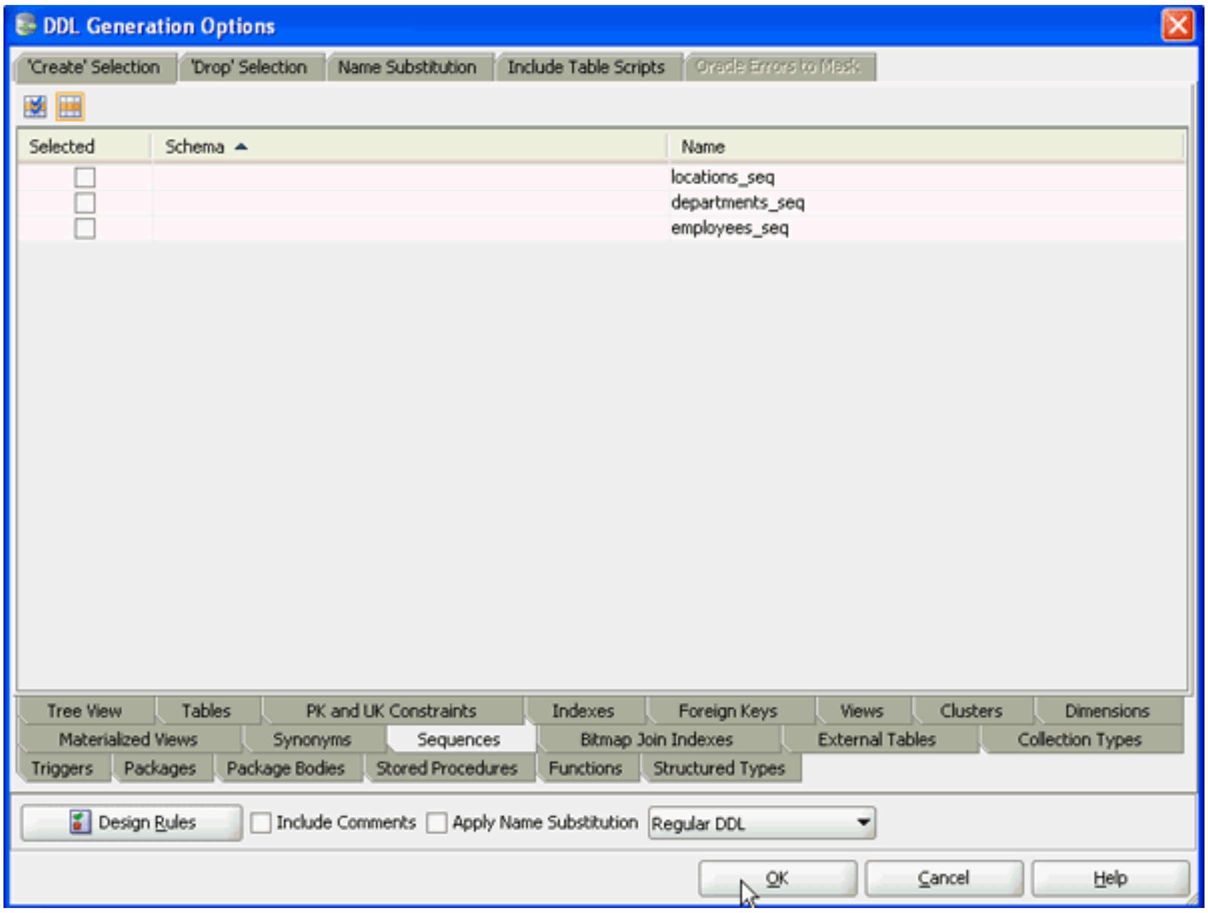
6 . Deselect **emp_details_view** and click the **Sequences** tab.



Selected	Schema	Name
☐		emp_details_view

☐ Include Comments ☐ Apply Name Substitution Regular DDL

7 . Click the Deselect All icon and click **OK** to generate the DDL for your selections.



Selected	Schema	Name
☐		locations_seq
☐		departments_seq
☐		employees_seq

8 . View the contents of the DDL file.

You can save the DDL and run it in SQL Developer. In this tutorial, click **Close** and perform the next tutorial.

Annexe : hr_cre.sql

```
Rem
Rem $Header: hr_cre.sql 29-aug-2002.11:44:03 hyeh Exp $
Rem
Rem hr_cre.sql
Rem
Rem Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.
Rem
Rem    NAME
Rem      hr_cre.sql - Create data objects for HR schema
Rem
Rem    DESCRIPTION
Rem      This script creates six tables, associated constraints
Rem      and indexes in the human resources (HR) schema.
Rem
Rem    NOTES
Rem
```

```
Rem    CREATED by Nancy Greenberg, Nagavalli Pataballa - 06/01/00
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    hyeh        08/29/02 - hyeh_mv_comschema_to_rdbms
Rem    ahunold     09/14/00 - Added emp_details_view
Rem    ahunold     02/20/01 - New header
Rem    vpatabal   03/02/01 - Added regions table, modified regions
Rem                          column in countries table to NUMBER.
Rem                          Added foreign key from countries table
Rem                          to regions table on region_id.
Rem                            Removed currency name, currency symbol
Rem                          columns from the countries table.
Rem                               Removed dn columns from employees and
Rem                          departments tables.
Rem                          Added sequences.
Rem                          Removed not null constraint from
Rem                          salary column of the employees table.

SET FEEDBACK 1
SET NUMWIDTH 10
SET LINESIZE 80
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100
SET ECHO OFF

REM ********************************************************************
REM Create the REGIONS table to hold region information for locations
REM HR.LOCATIONS table has a foreign key to this table.

Prompt ******  Creating REGIONS table ....

CREATE TABLE regions
    ( region_id      NUMBER
       CONSTRAINT  region_id_nn NOT NULL
    , region_name    VARCHAR2(25)
    );

CREATE UNIQUE INDEX reg_id_pk
ON regions (region_id);

ALTER TABLE regions
ADD ( CONSTRAINT reg_id_pk
                 PRIMARY KEY (region_id)
    ) ;

REM ********************************************************************
REM Create the COUNTRIES table to hold country information for customers
REM and company locations.
REM OE.CUSTOMERS table and HR.LOCATIONS have a foreign key to this table.

Prompt ******  Creating COUNTRIES table ....

CREATE TABLE countries
    ( country_id      CHAR(2)
       CONSTRAINT  country_id_nn NOT NULL
    , country_name    VARCHAR2(40)
    , region_id       NUMBER
    , CONSTRAINT     country_c_id_pk
                PRIMARY KEY (country_id)
    )
    ORGANIZATION INDEX;

ALTER TABLE countries
ADD ( CONSTRAINT countr_reg_fk
            FOREIGN KEY (region_id)
             REFERENCES regions(region_id)
    ) ;

REM ********************************************************************
REM Create the LOCATIONS table to hold address information for company depar
REM HR.DEPARTMENTS has a foreign key to this table.

Prompt ******  Creating LOCATIONS table ....

CREATE TABLE locations
    ( location_id    NUMBER(4)
    , street_address VARCHAR2(40)
    , postal_code    VARCHAR2(12)
    , city       VARCHAR2(30)
     CONSTRAINT     loc_city_nn  NOT NULL
    , state_province VARCHAR2(25)
    , country_id     CHAR(2)
    ) ;

CREATE UNIQUE INDEX loc_id_pk
ON locations (location_id) ;

ALTER TABLE locations
ADD ( CONSTRAINT loc_id_pk
                 PRIMARY KEY (location_id)
    , CONSTRAINT loc_c_id_fk
                 FOREIGN KEY (country_id)
             REFERENCES countries(country_id)
    ) ;

Rem   Useful for any subsequent addition of rows to locations table
Rem   Starts with 3300

CREATE SEQUENCE locations_seq
 START WITH     3300
 INCREMENT BY   100
 MAXVALUE       9900
 NOCACHE
 NOCYCLE;

REM ********************************************************************
REM Create the DEPARTMENTS table to hold company department information.
REM HR.EMPLOYEES and HR.JOB_HISTORY have a foreign key to this table.

Prompt ******  Creating DEPARTMENTS table ....

CREATE TABLE departments
    ( department_id    NUMBER(4)
    , department_name  VARCHAR2(30)
     CONSTRAINT  dept_name_nn  NOT NULL
    , manager_id       NUMBER(6)
    , location_id      NUMBER(4)
    ) ;

CREATE UNIQUE INDEX dept_id_pk
ON departments (department_id) ;

ALTER TABLE departments
ADD ( CONSTRAINT dept_id_pk
                 PRIMARY KEY (department_id)
    , CONSTRAINT dept_loc_fk
                 FOREIGN KEY (location_id)
             REFERENCES locations (location_id)
     ) ;

Rem   Useful for any subsequent addition of rows to departments table
Rem   Starts with 280

CREATE SEQUENCE departments_seq
 START WITH     280
 INCREMENT BY   10
 MAXVALUE       9990
 NOCACHE
 NOCYCLE;

REM ********************************************************************
REM Create the JOBS table to hold the different names of job roles within th
REM HR.EMPLOYEES has a foreign key to this table.

Prompt ******  Creating JOBS table ....

CREATE TABLE jobs
    ( job_id         VARCHAR2(10)
    , job_title      VARCHAR2(35)
      CONSTRAINT     job_title_nn  NOT NULL
    , min_salary     NUMBER(6)
    , max_salary     NUMBER(6)
    ) ;

CREATE UNIQUE INDEX job_id_pk
ON jobs (job_id) ;

ALTER TABLE jobs
ADD ( CONSTRAINT job_id_pk
                 PRIMARY KEY(job_id)
    ) ;

REM ********************************************************************
REM Create the EMPLOYEES table to hold the employee personnel
REM information for the company.
REM HR.EMPLOYEES has a self referencing foreign key to this table.

Prompt ******  Creating EMPLOYEES table ....

CREATE TABLE employees
    ( employee_id    NUMBER(6)
    , first_name     VARCHAR2(20)
    , last_name      VARCHAR2(25)
```

```
       CONSTRAINT     emp_last_name_nn  NOT NULL
    , email           VARCHAR2(25)
      CONSTRAINT     emp_email_nn  NOT NULL
    , phone_number   VARCHAR2(20)
    , hire_date       DATE
      CONSTRAINT     emp_hire_date_nn  NOT NULL
    , job_id          VARCHAR2(10)
      CONSTRAINT     emp_job_nn  NOT NULL
    , salary          NUMBER(8,2)
    , commission_pct NUMBER(2,2)
    , manager_id      NUMBER(6)
    , department_id  NUMBER(4)
    , CONSTRAINT      emp_salary_min
                      CHECK (salary > 0)
    , CONSTRAINT      emp_email_uk
                      UNIQUE (email)
    ) ;

CREATE UNIQUE INDEX emp_emp_id_pk
ON employees (employee_id) ;


ALTER TABLE employees
ADD ( CONSTRAINT      emp_emp_id_pk
                      PRIMARY KEY (employee_id)
    , CONSTRAINT      emp_dept_fk
                      FOREIGN KEY (department_id)
                       REFERENCES departments
    , CONSTRAINT      emp_job_fk
                      FOREIGN KEY (job_id)
                       REFERENCES jobs (job_id)
    , CONSTRAINT      emp_manager_fk
                      FOREIGN KEY (manager_id)
                       REFERENCES employees
    ) ;

ALTER TABLE departments
ADD ( CONSTRAINT dept_mgr_fk
                  FOREIGN KEY (manager_id)
                   REFERENCES employees (employee_id)
    ) ;


Rem   Useful for any subsequent addition of rows to employees table
Rem   Starts with 207


CREATE SEQUENCE employees_seq
 START WITH     207
 INCREMENT BY   1
 NOCACHE
 NOCYCLE;

REM ***********************************************************************
REM Create the JOB_HISTORY table to hold the history of jobs that
REM employees have held in the past.
REM HR.JOBS, HR_DEPARTMENTS, and HR.EMPLOYEES have a foreign key to this tab
```

```
Prompt ******  Creating JOB_HISTORY table ....

CREATE TABLE job_history
    ( employee_id   NUMBER(6)
       CONSTRAINT    jhist_employee_nn  NOT NULL
    , start_date    DATE
      CONSTRAINT    jhist_start_date_nn  NOT NULL
    , end_date      DATE
      CONSTRAINT    jhist_end_date_nn  NOT NULL
    , job_id        VARCHAR2(10)
      CONSTRAINT    jhist_job_nn  NOT NULL
    , department_id NUMBER(4)
    , CONSTRAINT    jhist_date_interval
                    CHECK (end_date > start_date)
    ) ;

CREATE UNIQUE INDEX jhist_emp_id_st_date_pk
ON job_history (employee_id, start_date) ;

ALTER TABLE job_history
ADD ( CONSTRAINT jhist_emp_id_st_date_pk
      PRIMARY KEY (employee_id, start_date)
    , CONSTRAINT     jhist_job_fk
                     FOREIGN KEY (job_id)
                     REFERENCES jobs
    , CONSTRAINT     jhist_emp_fk
                     FOREIGN KEY (employee_id)
                     REFERENCES employees
    , CONSTRAINT     jhist_dept_fk
                     FOREIGN KEY (department_id)
                     REFERENCES departments
    ) ;

REM *******************************************************************
REM Create the EMP_DETAILS_VIEW that joins the employees, jobs,
REM departments, jobs, countries, and locations table to provide details
REM about employees.

Prompt ******  Creating EMP_DETAILS_VIEW view ...

CREATE OR REPLACE VIEW emp_details_view
  (employee_id,
   job_id,
   manager_id,
   department_id,
   location_id,
   country_id,
   first_name,
   last_name,
   salary,
   commission_pct,
   department_name,
   job_title,
   city,
   state_province,
   country_name,
   region_name)
AS SELECT
```

```
  e.employee_id,
  e.job_id,
  e.manager_id,
  e.department_id,
  d.location_id,
  l.country_id,
  e.first_name,
  e.last_name,
  e.salary,
  e.commission_pct,
  d.department_name,
  j.job_title,
  l.city,
  l.state_province,
  c.country_name,
  r.region_name
FROM
  employees e,
  departments d,
  jobs j,
  locations l,
  countries c,
  regions r
WHERE e.department_id = d.department_id
  AND d.location_id = l.location_id
  AND l.country_id = c.country_id
  AND c.region_id = r.region_id
  AND j.job_id = e.job_id

WITH READ ONLY;
```