

# Using Oracle Data Modeler v4.1.5

Oracle Data Modeler (ODM) is a database modeling tool that is used to create logical and/or physical models of your database using several notation styles.

This document assumes that you have access to the UMUC AWS DaaS lab platform and that you understand basic Microsoft Windows functionality.

Read the documentation carefully then follow the steps shown in this tutorial to create a basic data model which you will turn in to verify that you're able to access and utilize ODM correctly. Use the course "Discussion" area to pose any questions you have on using ODM.

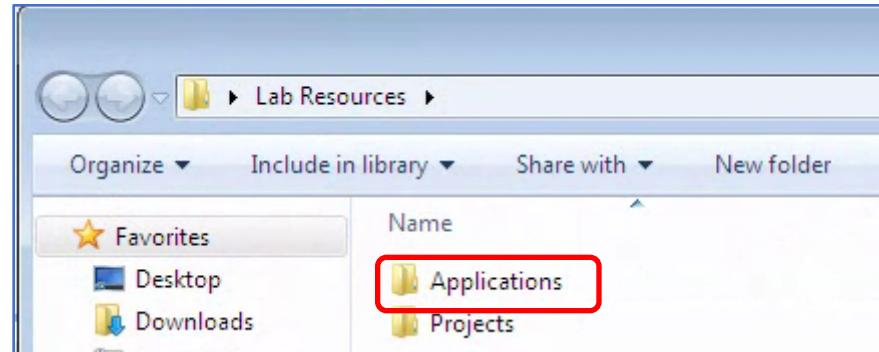
## Access Oracle Data Modeler

In the UMUC AWS DaaS environment, you can launch ODM from the **Lab Resources -> Applications** folders.

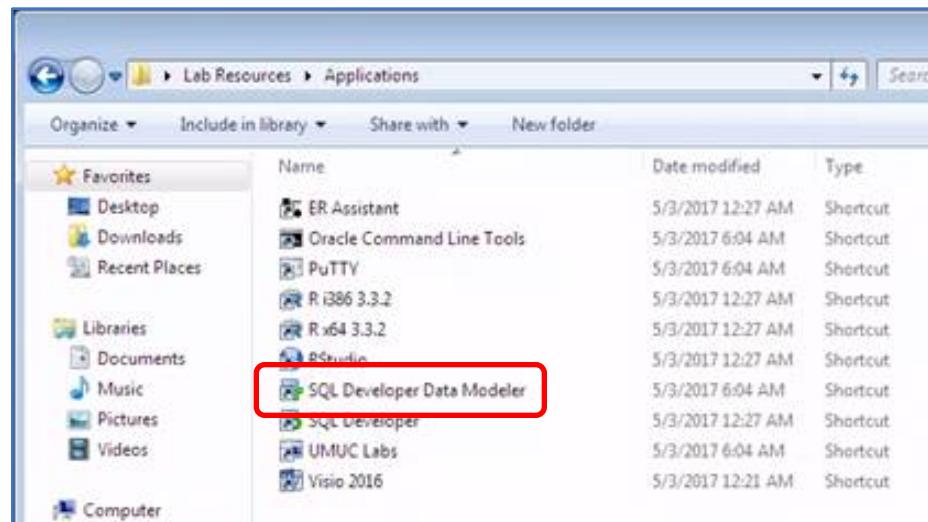
1. Double click on **Lab Resources**



2. Double click on **Applications**

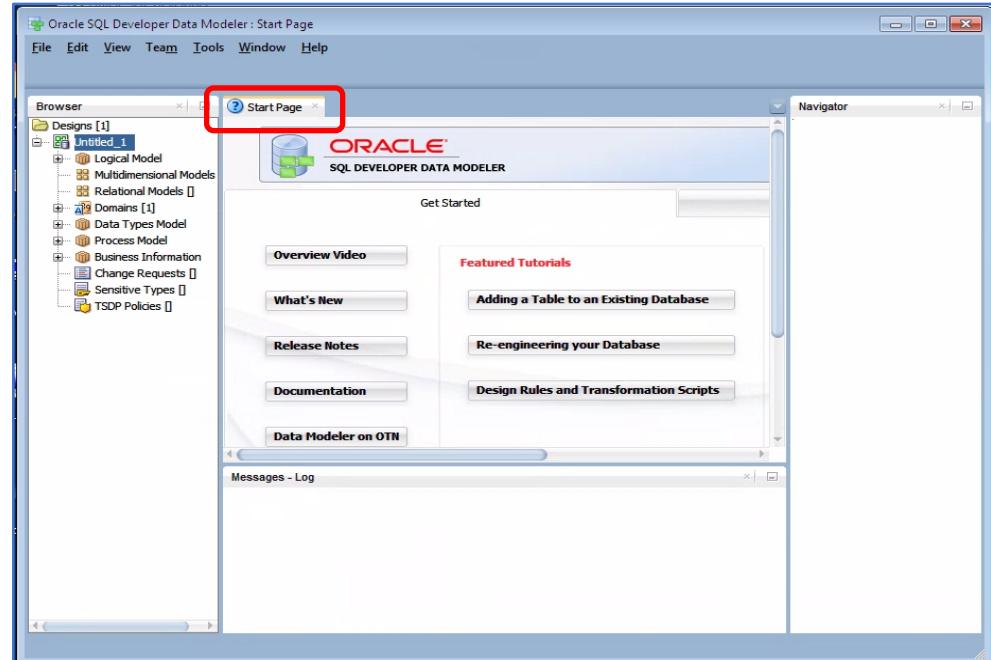


3. Double click on **SQL Developer Data Modeler**



## Access Oracle Data Modeler

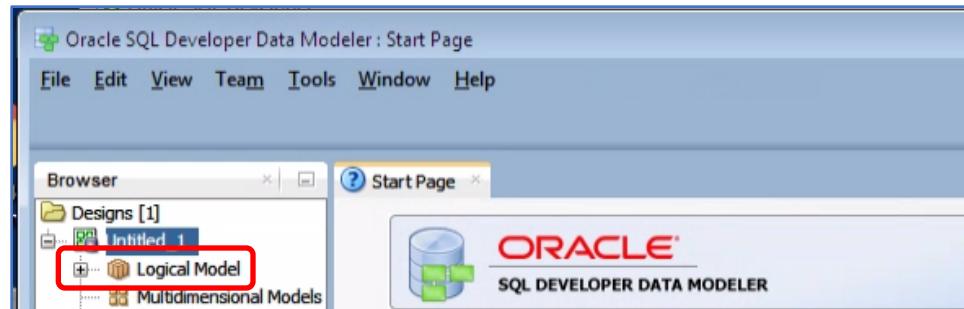
The Oracle SQL Developer Data Modeler – Start Page will be displayed. The **Start Page** contains reference links to documentation provided by Oracle.



4. Click on **Help -> About** to see the version of ODM you're working with. You should see "Version 4.1.5"

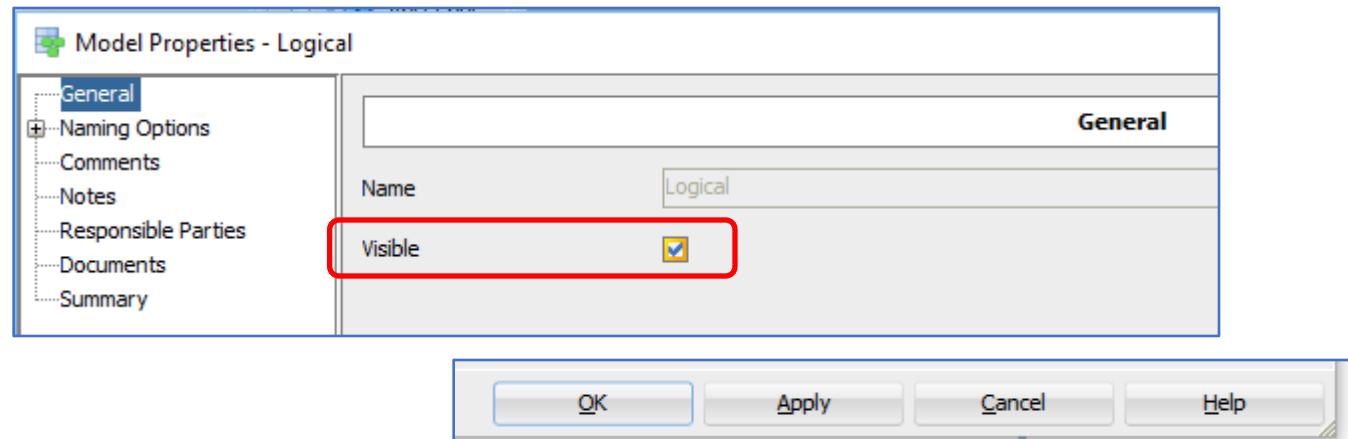
## Create the “Employee” Logical Database Model

5. On Start Page, Browser pane:
- Designs folder
  - Untitled\_1 section
  - Double click on **Logical Model**



6. Ensure **General** is highlighted:

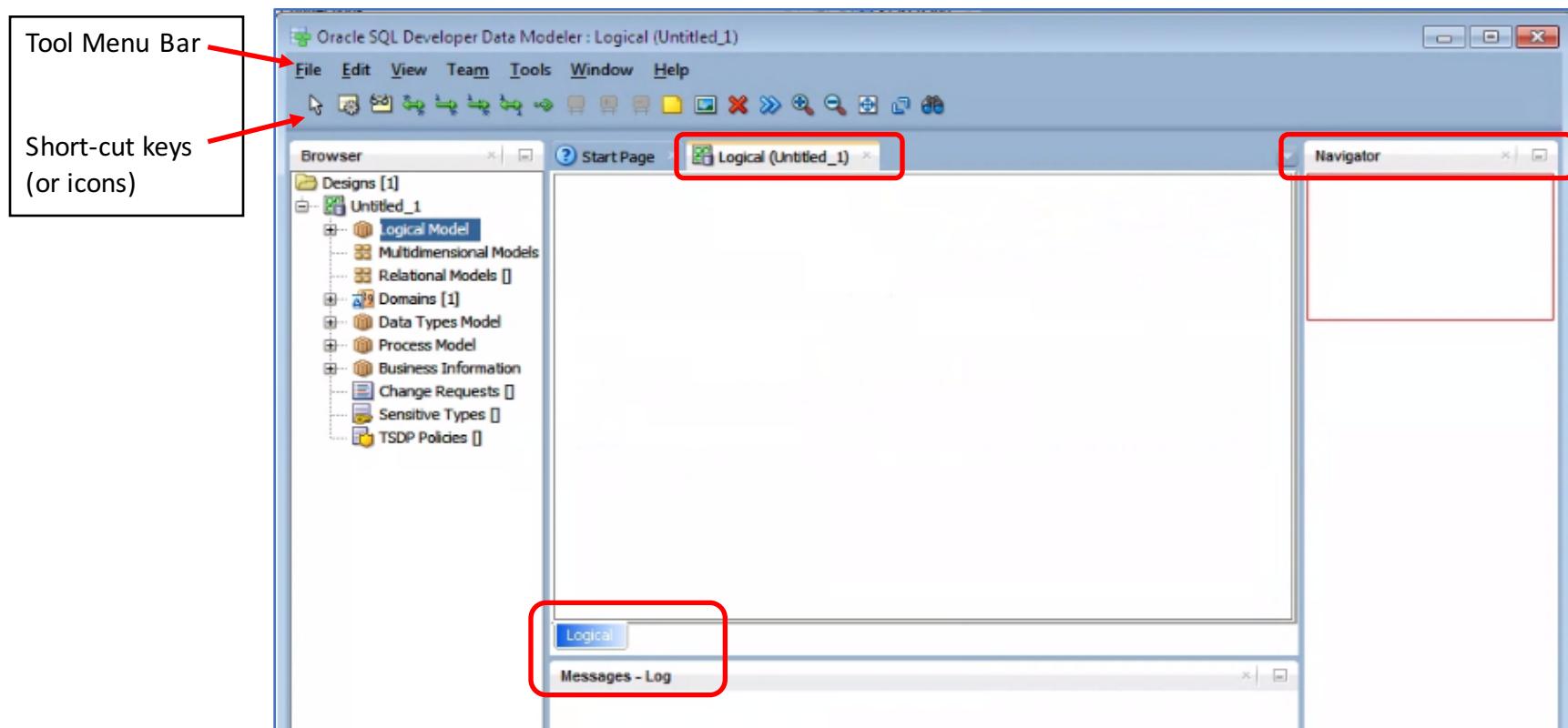
- Check **Visible**
- Click **OK** at the bottom right of the **Model Properties – Logical** window



## Create the “Employee” Logical Database Model

6. You should now see a tab for **Logical (Untitled\_1)**

- “Navigator” pane is to the far right
- “Messages – Log” pane is at the bottom center

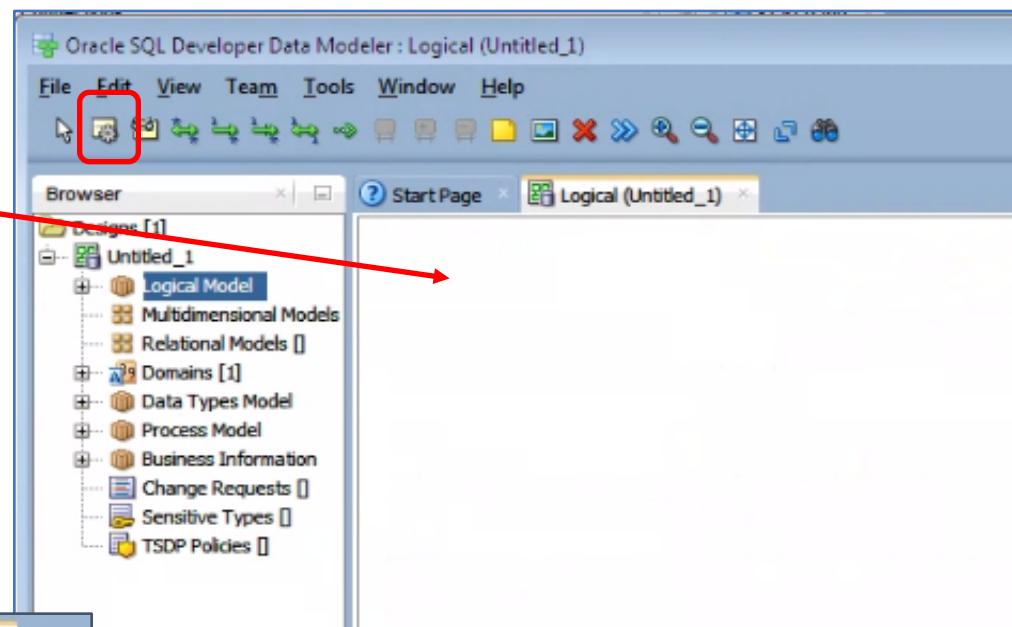


## Create the “Employee” Entity

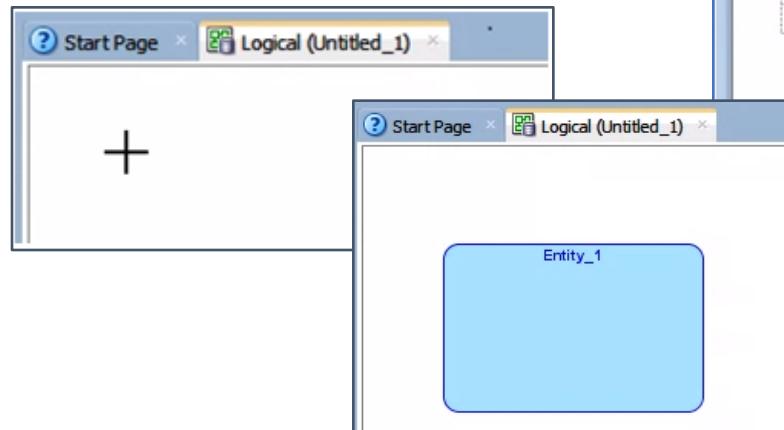
“Entities” are the tables in your database. The logical diagram can contain one or more entities belonging to your database. We’ll create 3 entities for this tutorial starting with the “Employee” entity.

7. To create an entity on a logical database diagram:

- Click on the **Entity** icon, bottom row of icons, 2<sup>nd</sup> from the left
- Click anywhere in the **Logical (Untitled\_1)** tab area



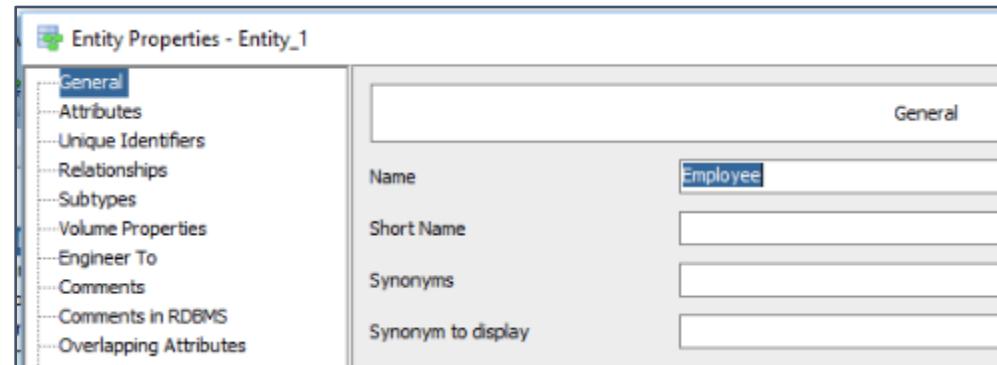
NOTE that you may see your mouse cursor as a large “plus/+”. You may also notice a blue “Entity\_1” box in the **Logical (Untitled\_1)** tab background.



## Create the “Employee” Entity - Continued

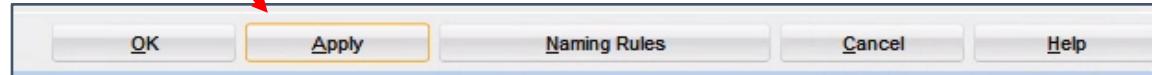
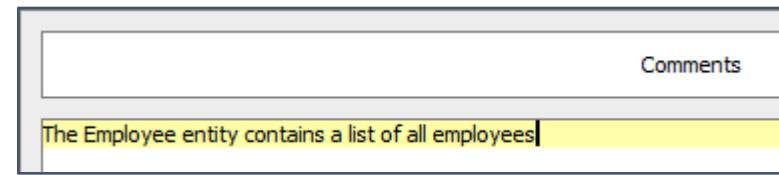
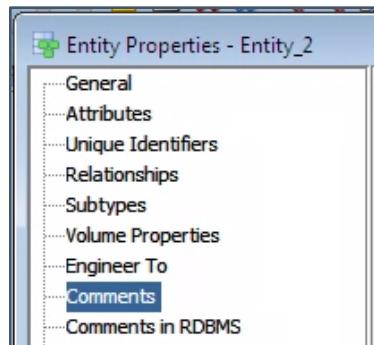
The Entity Properties window will be displayed allowing you to configure the “Employee” entity.

8. On the **Entity Properties** window, under **General -> Name**, enter: Employee



9. Add a comment to describe the “Employee” entity

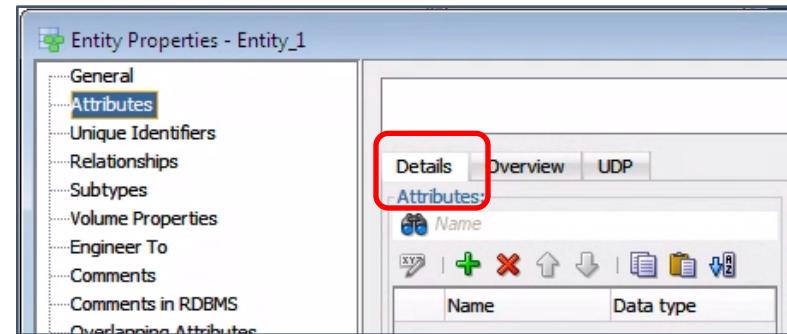
- Click **Comments**
- Enter the comment shown on the far right screenshot
- Click **Apply** at the bottom of the **Entity Properties** screen



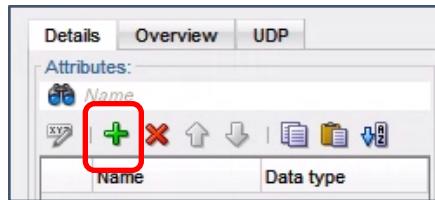
## Add “Attributes” to the Employee Entity

“Attributes” are the fields in each entity that hold data for your database. Each attribute has a name, data type and other characteristics that define the attribute.

10. On the **Entity Properties** screen, click **Attributes**. Ensure that the **Details** tab is active



11. Click on "green plus" icon to add an attribute



12. Enter Attribute Properties

- **Name:** Employee\_ID
- **Data Type:** Logical
- **Source Type:** NUMERIC
- **Precision** (total number of digits): 2
- **Scale** (how many digits before/after decimal): leave blank
- **Primary UID:** checked
- **Comments tab:** Enter the comment shown
- Click **Apply** when completed

The screenshot shows the Attribute Properties dialog for the Employee\_ID attribute. The 'Name' field is set to 'Employee\_ID'. Under 'Data Type', 'Logical' is selected. Under 'Source Type', 'NUMERIC' is selected. In the 'Precision' field, '2' is entered. The 'Scale' field is empty. At the bottom, the 'Primary UID' checkbox is checked. The 'Comments' tab is active, containing the note: 'Employee\_ID is the primary key for the Employee entity'. Other tabs like 'Comments in RDBMS' and 'Notes' are also visible.

## Add “Attributes” to the Employee Entity - Continued

You should see your new attribute under **Details -> Attributes**

13. Repeat the steps on the previous slide to add the following attributes: Fname, Lname, Title, Salary and HireDate

Click **Apply** after you complete each attribute

Attributes:		
Name	Data type	
Employee_ID	NUMERIC (2)	

Attribute Properties

Name	Fname
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	VARCHAR
Size	15
Units:	
<input type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments	Comments in RDBMS Notes
This is the employee's first name	

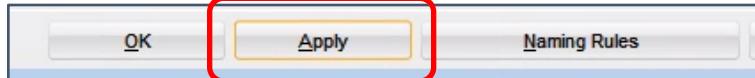
Attribute Properties

Name	Lname
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	VARCHAR
Size	25
Units:	
<input type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments	Comments in RDBMS Notes
This is the employee's last name	

Attribute Properties

Name	Title
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	VARCHAR
Size	35
Units:	
<input type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments	Comments in RDBMS Notes
This is the employee's job title	

14. Click **Apply** after you complete each attribute



## Add “Attributes” to the Employee Entity - Continued

15. Click **Apply** after you complete the “HireDate” attribute

Attribute Properties

Name	HireDate
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	Date <input type="checkbox"/> Preferred
<input type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments    Comments in RDBMS    Notes	
This is the employee's hire date	

Attribute Properties

Name	Salary
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	NUMERIC <input type="checkbox"/> Preferred
Precision	6
Scale	2
<input type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments    Comments in RDBMS    Notes	
This is the employee's salary with a total of 6 digits; 2 digits are after the decimal.	

16. Click OK after you finish the “Salary” attribute  
You should now see all 5 attributes under the **Details** tab



Details    Overview    UDP

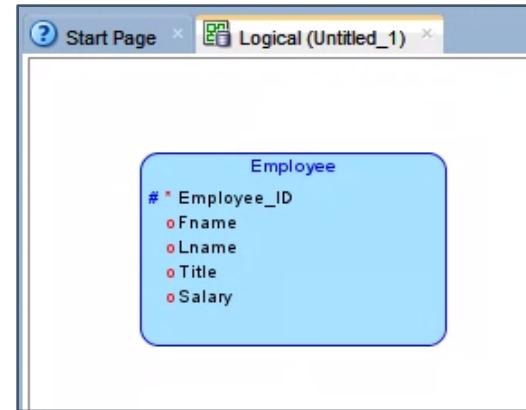
Attributes:

Name	Data type
1 Employee_ID	NUMERIC (2)
2 Fname	VARCHAR (15)
3 Lname	VARCHAR (25)
4 Title	VARCHAR (35)
5 Salary	NUMERIC (6, 2)
6 HireDate	Date

## Update the Diagram Notation

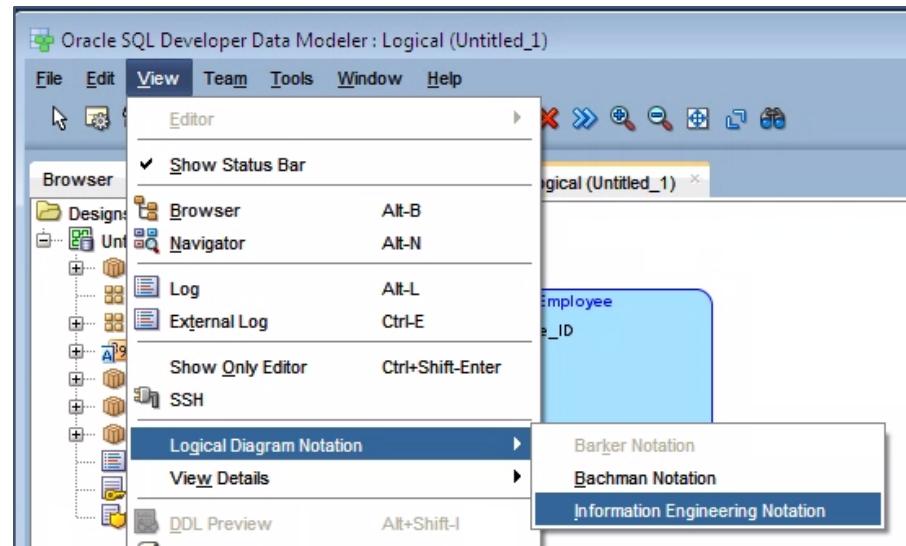
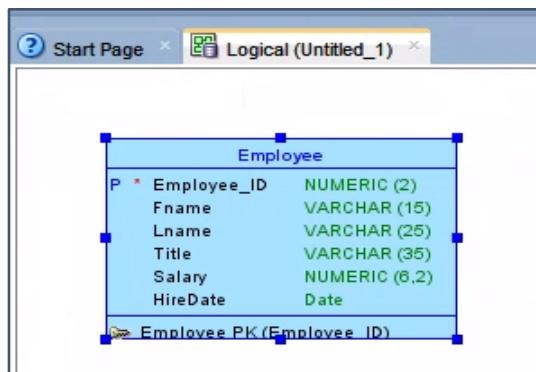
17. Click **Cancel**

You should now see the “Employee” entity and the 5 attributes you added



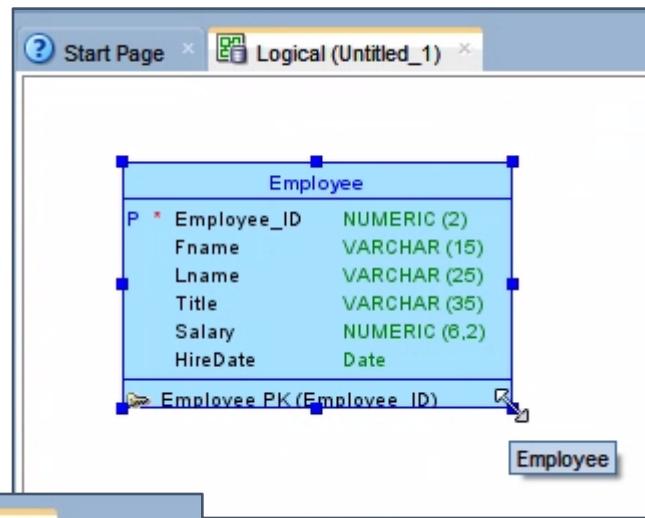
18. Click **View – Logical Diagram Notation -> Information Engineering Notation**

You should now see the “Employee” entity as shown below

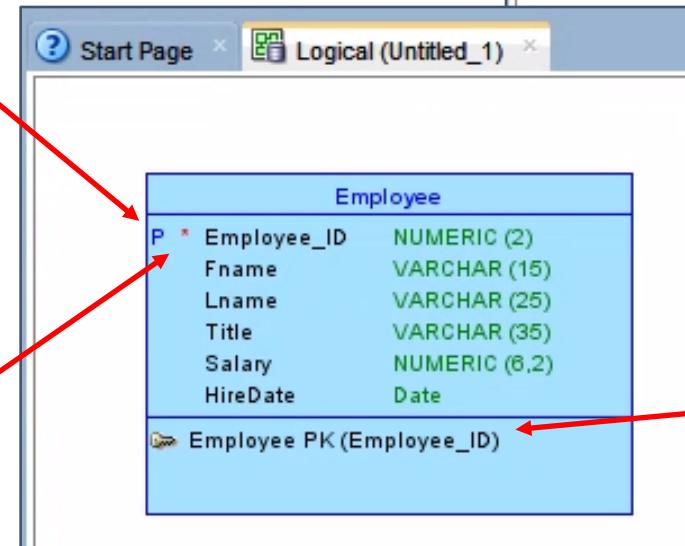


## About the “Employee” Entity

19. You can hover your mouse cursor over one of the “handles” (or squares) and drag to resize the entity window



20. "P" indicate that the "Employee\_ID" attribute is a "primary key"



21. Primary key attributes are "required" by default, which is what the red "\*" indicates

22. The "primary key constraint" definition is in the bottom section of the "Employee" entity information

## Create the “Department” Entity

### 23. Add the “Department” entity

Entity Properties - Entity\_9

General

Name	Department
Short Name	

Entity Properties - Entity\_9

General

Attributes

Unique Identifiers

Relationships

Subtypes

Volume Properties

Engineer To

Comments

Comments in RDBMS

The “Department” entity contains a list of departments an employee can work in.

### 24. Add the “DepartmentID”, “DepartmentName” and “Description” attributes

Attribute Properties

Name: DepartmentName

Data Type: Logical

Source Type: VARCHAR

Size: 25

Units:

Comments: "DepartmentName" is the name of the department

Attribute Properties

Name: DepartmentID

Data Type: Logical

Source Type: NUMERIC

Precision: 2

Scale:

Primary UID:

Comments: The "DepartmentID" is the primary key attribute for the "Department" entity

Attribute Properties

Name: Description

Data Type: Logical

Source Type: VARCHAR

Size: 25

Units:

Comments: This attribute holds the department description

## Complete the “Department” Entity

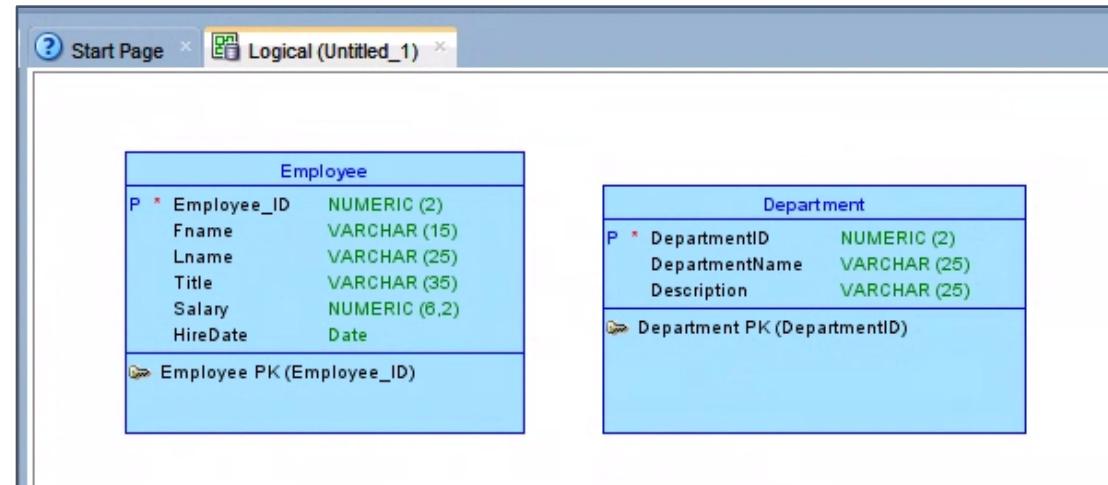
25. Click **Cancel** after you complete the “Description” attribute

The 3 attributes you added should be listed under the **Details** tab



Attributes:	
Name	Data type
1 DepartmentID	NUMERIC (2)
2 DepartmentName	VARCHAR (25)
3 Description	VARCHAR (25)

26. You should now see the “Department” entity and the 3 attributes you added



## Create a Relationship Between “Employee” and “Department” Entities

Entities are related based on a set of business rules. For this tutorial, the relationship business rules are:

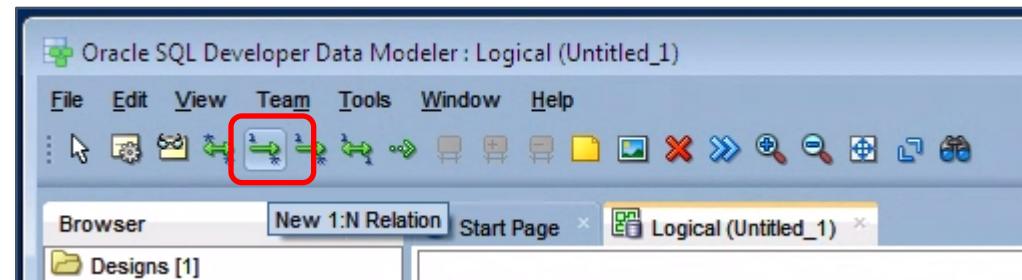
- Each employee must be listed in the "Employee" entity
- Each department in the company must be listed in the "Department" entity
- An employee must work in only one department
- A department may have 0 or more employees

Note that "must" indicates a "required" situation and "may" or "can" are "optional" indicators. The diagram notation must match the verbiage used in the business rules.

We're going to create a "1:N", or "one-to-many", relationship between "Employee" and "Department" that will satisfy the business rules given above.

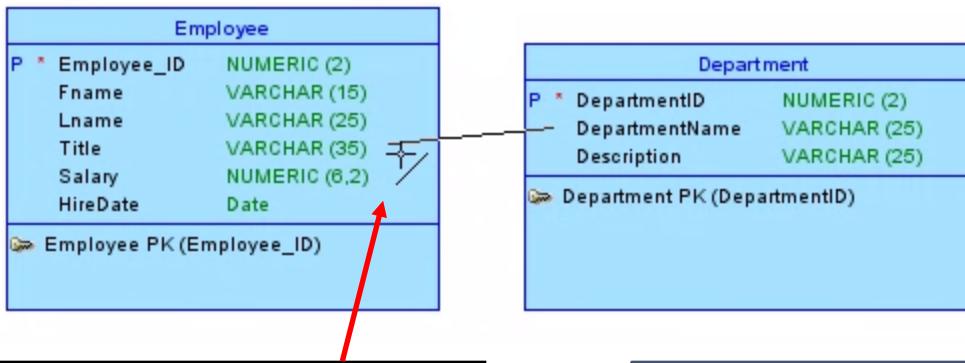
27. The "green arrow" icons on the 2<sup>nd</sup> row are for creating relationships.

Click on the 2<sup>nd</sup> one from the left to create a "1:N", or "one-to-many" relationship



## Create a Relationship Between “Employee” and “Department” Entities - Continued

28. Click on Department then click on Employee

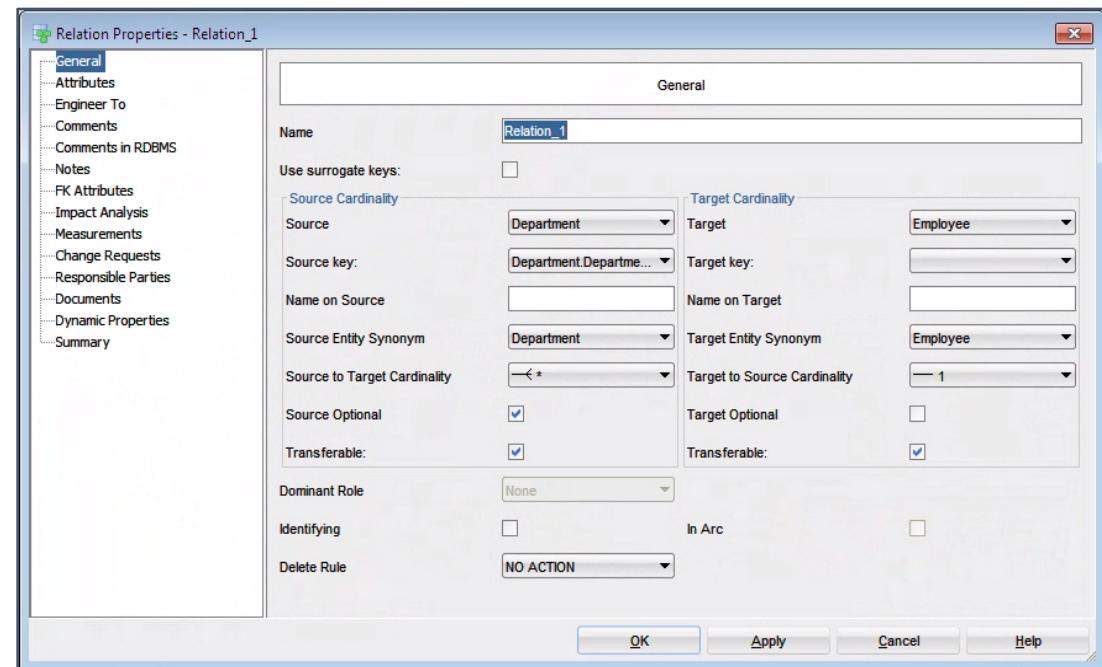


29. Note that your mouse cursor will show a "+", "/" indicator as you hover over each entity

31. The **Relation Properties** screen will be displayed showing the two tables you picked.

If you clicked the entities in the wrong order, update the **Source** and **Target** entities to what you see in the **Relation Properties** window to the right.

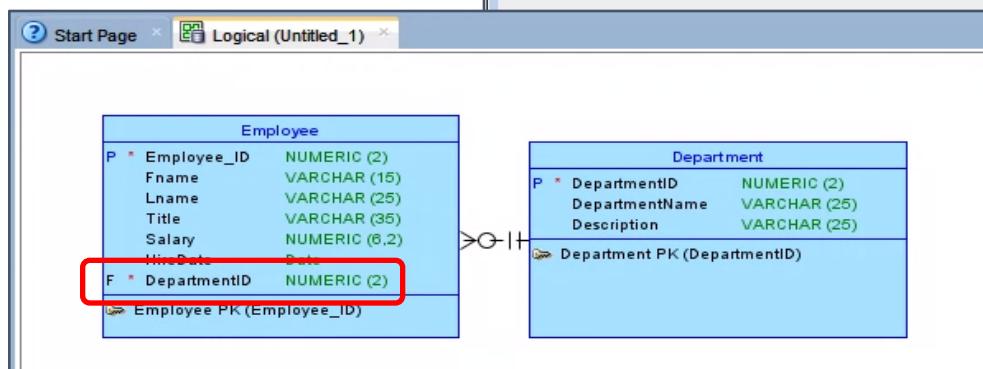
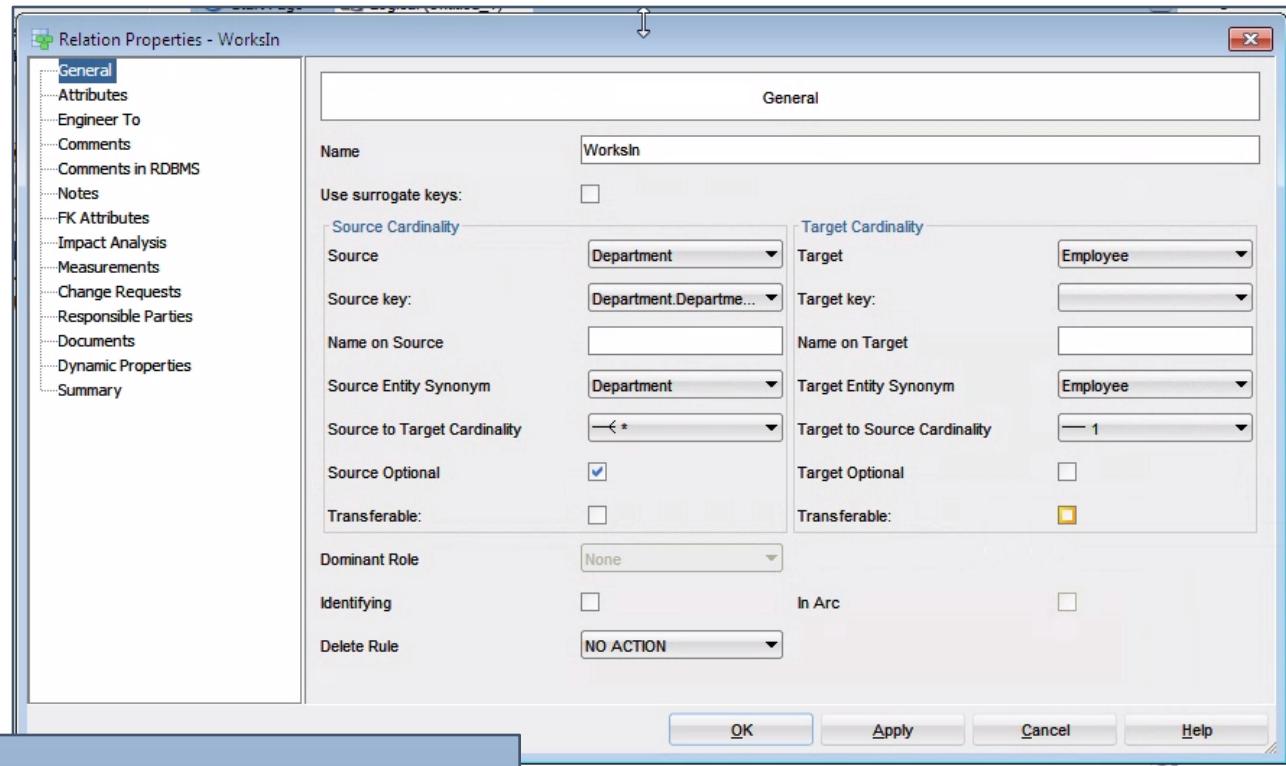
30. Also note that the "Source" entity is clicked on first, then the "Target" entity so that the relationship settings are created for you.



## Create a Relationship Between “Employee” and “Department” Entities - Continued

32. Update your Relation Properties window as shown

- **Name:** WorksIn
- Both **Transferable** boxes: Unchecked



33. You will now see a "foreign key" attribute in **Employee**, noted with an "F" and that it is required, as noted with a red asterisk.

## Relationship Diagram Details

34. The "Employee" side of the relationship has "cardinality" of "many" and "participation" of "optional". The foreign key field is "required".

Rules:

- Each Employee must have a department
- A department can be used for 0, 1 or more employees (a department can be in the "Department" entity and not be assigned to an "Employee")

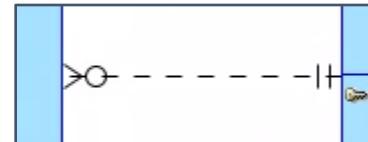
Employee	
P *	Employee_ID NUMERIC (2)
Fname	VARCHAR (15)
Lname	VARCHAR (25)
Title	VARCHAR (35)
Salary	NUMERIC (6,2)
HireDate	Date
F *	DepartmentID NUMERIC (2)
Employee PK (Employee_ID)	

Department	
P *	DepartmentID NUMERIC (2)
DepartmentName	VARCHAR (25)
Description	VARCHAR (25)
Department PK (DepartmentID)	

35. The "Department" side of the relationship has "cardinality" of "one" and "participation" of "mandatory".

Rules:

- Each department is only listed once
- Each department must be listed in "Department" before it can be assigned to an Employee



36. Since "Identifying" wasn't checked in the General settings window for this relationship, a dotted line indicates a "strong" relationship with each entity having its own primary key.

## Create the “Region” Entity

37. Add a “Region” entity and “RegionID” and “RegionName” attributes using the following details:

### Entity Properties

**General -> Name:** Region

**Comments ->** The “Region” entity holds a list of region names

**NOTE:** Make “RegionName” a “Mandatory” field by checking the box

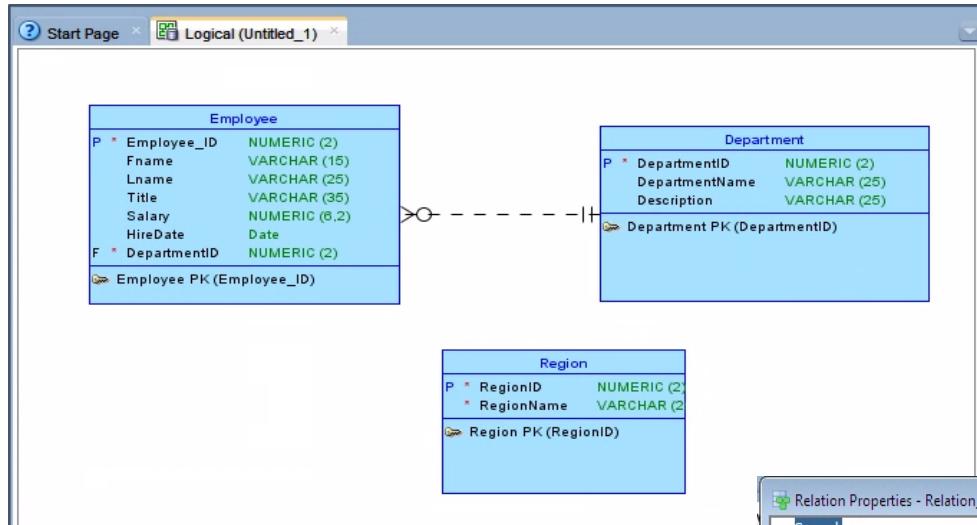
Attribute Properties

Name	RegionID
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	NUMERIC
Precision	2
Scale	0
<input checked="" type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input checked="" type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments    Comments in RDBMS    Notes	
"RegionID" is primary key attribute for the "Region" entity	

Attribute Properties

Name	RegionName
Data Type	<input type="radio"/> Domain <input checked="" type="radio"/> Logical <input type="radio"/> Distinct <input type="radio"/> Structured <input type="radio"/> Collection
Source Type	VARCHAR
Size	25
Units:	
<input type="checkbox"/> Primary UID <input type="checkbox"/> Relation UID <input checked="" type="checkbox"/> Mandatory <input type="checkbox"/> Deprecated	
Comments    Comments in RDBMS    Notes	
RegionName is the name of the region	

## Create a Relationship Between “Department” and “Region” Entities



38. You should now have a diagram that looks like the screenshot to the left.

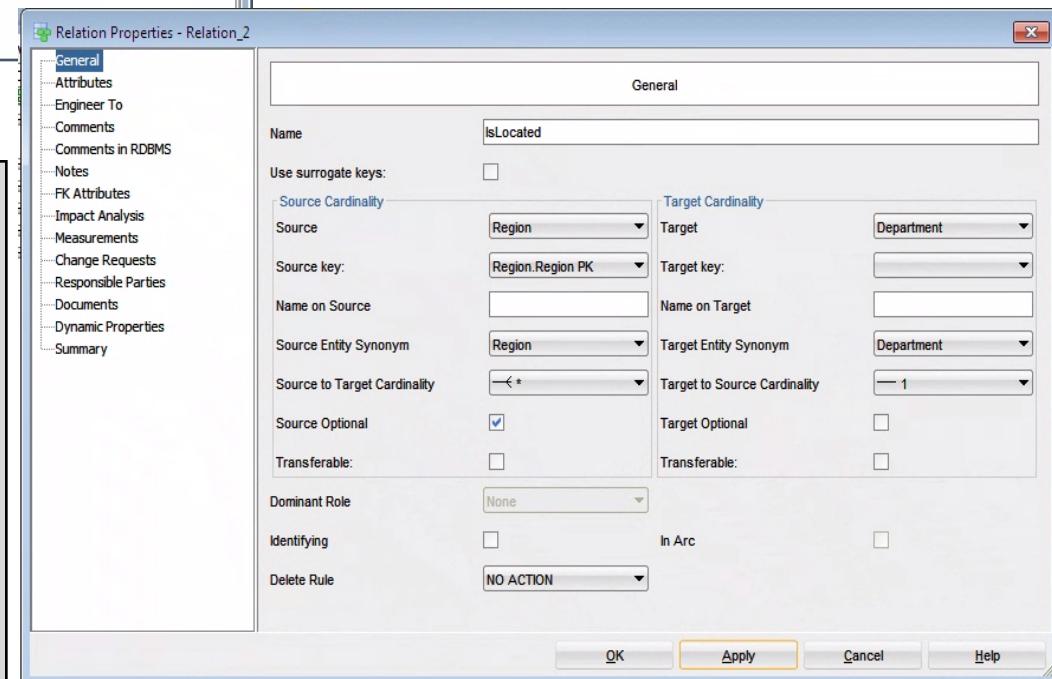
Note that there is a red asterisk (\*) next to "RegionName" as it is a "required" attribute.

39. Create a relationship called “isLocated” with the configuration shown to the right

Relationship business rules:

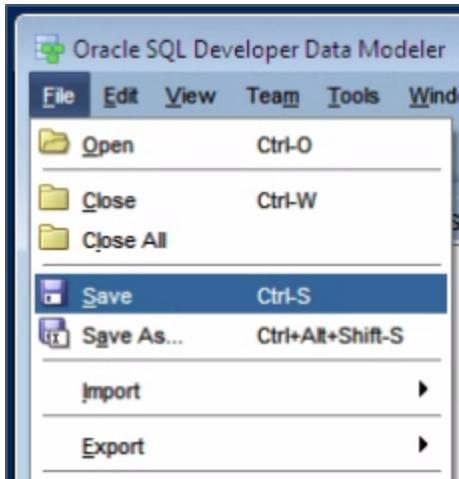
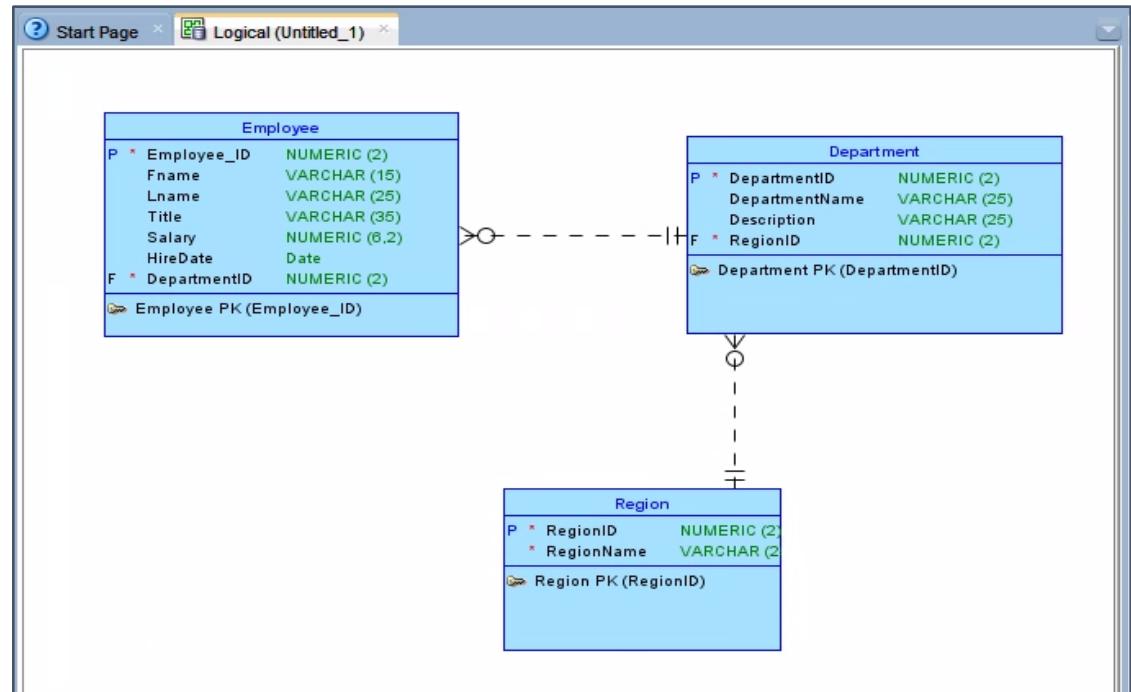
- A region is only listed once in the "Region" entity
- A department must be associated with one "Region"
- A region can be assigned to more than one department

Note that the last rule indicates an "optional" condition where a region is listed in the "Region" entity but may not be assigned any departments



## View and Save the Logical Model

40. Your logical model should now look similar to the screenshot shown here



41. Go to **File -> Save** to save your logical model

## View and Save the Logical Model - Continued

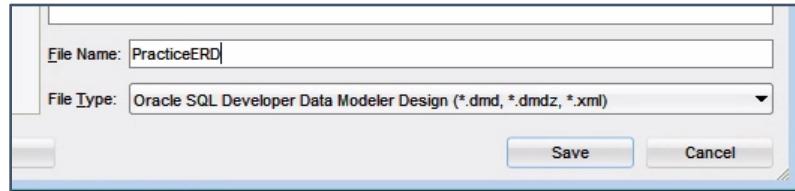
42. Click **Documents**, then double click **Models**

Note that I made a "Models" folder to save my Data Modeler files

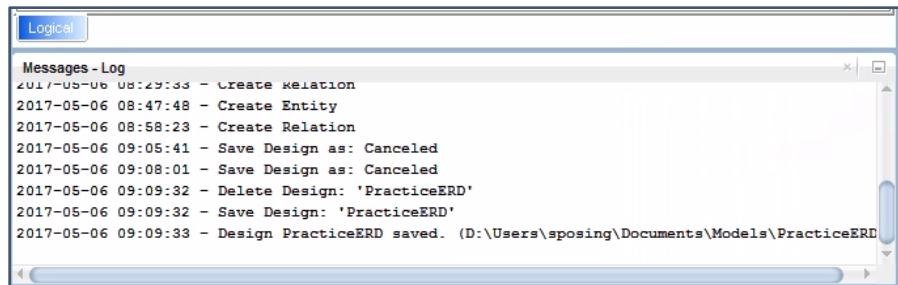


43. Type in file name: "PracticeERD"

Click **Save**



44. You should see the "Save" action in the "Messages – Log" pane at the bottom of the Data Modeler tool



45. Data Modeler saves your designs as a "DMD" file and a folder of the same name. The folder contains additional information about your design.

Note that these files CANNOT be sent to someone else to view your model. You must use the "Export" function to create a DMD file and folder to share with someone else. (See tutorial "Export ODM Models")

### Documents library

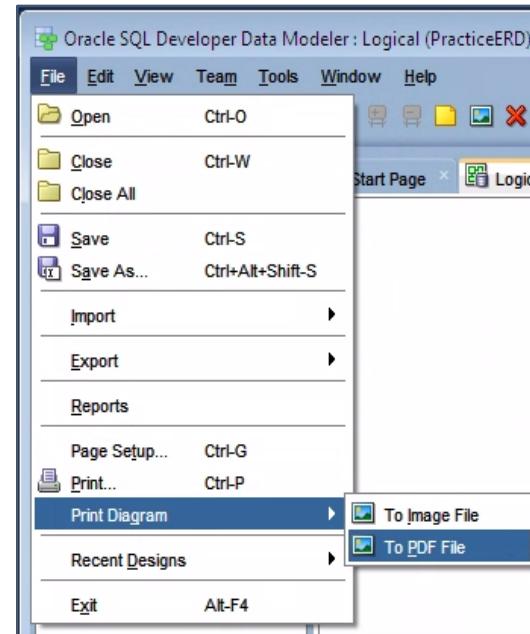
Models

Name	Date modified	Type
PracticeERD	5/6/2017 9:09 AM	File folder
PracticeERD.dmd	5/6/2017 9:09 AM	DMD File

## Print the Logical Model as PDF

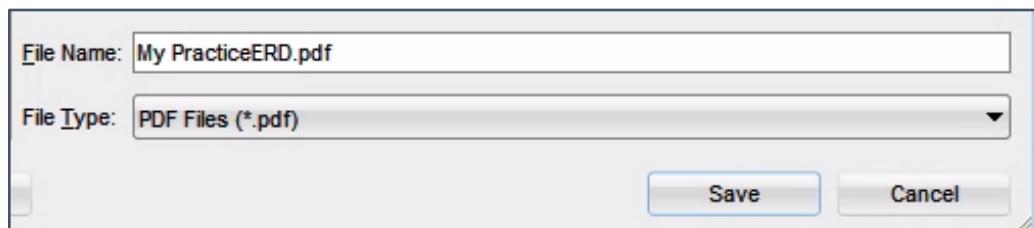
46. You can print the design as a "PDF" file in order to turn it in for this tutorial.

Click **File -> Print Diagram -> To PDF File**



47. Give your file a name

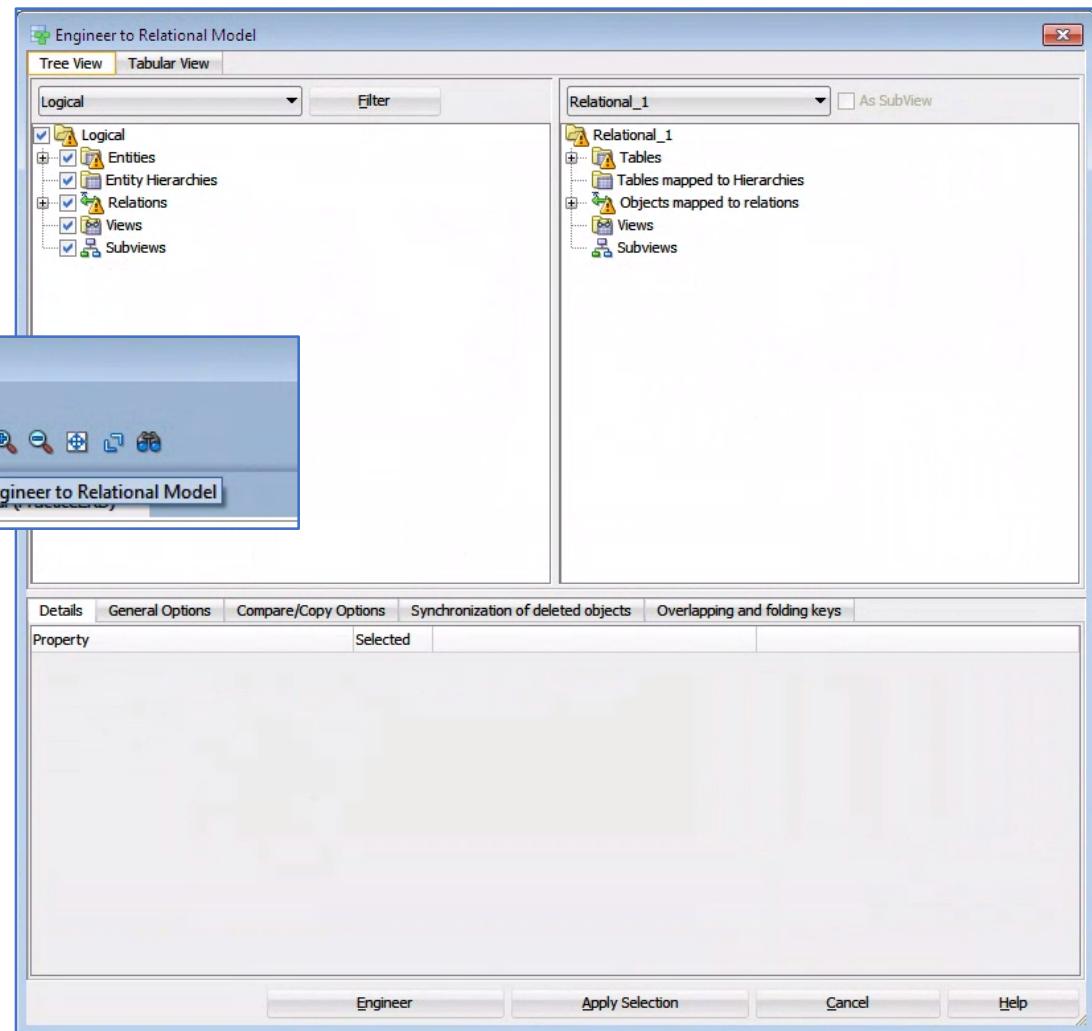
Click **Save**



## Convert Logical Model to Relational Model

48. You can engineer a relational model from your logical one.

With your logical model open, click double, blue, right arrow icon in the toolbar



49. Ensure that all of your logical model items are checked

Click **Engineer**

## Convert Logical Model to Relational Model - Continued

50. Your Relational model should now be displayed and look something like what's shown below. You'll notice that the foreign key references are listed on this model.

