

Workshop 4 - Creating E-R Model

In this workshop you will learn:

To draw an E-R diagram consisting of entities, attributes and relationships using Oracle SQL Developer Data Modeler

What is an E-R Model?

An E-R model or entity-relationship model is a logical representation of the data for a firm or business area. The E-R model consists of entities, relationships between or among entities, and the attributes of the entities. In order to create a detailed and fairly comprehensive E-R model, you will have to be clear about some of its relevant components, such as the entities, attributes, relationships, and the degree as well as the cardinality of a relationship. In this workshop, we will introduce a case study of simplified Student Registration System to learning the ERD technique.

Relevant Terms in E-R Modeling

An **entity** is a person, place, object or concept about which the organization wishes to maintain some data (e.g., STUDENT).

Attributes are certain properties of an entity that is of relevance to the firm (e.g. Student_ID).

Relationships are associations between the instances of one or more of the entities. e.g.

Student TAKES Courses where Student and Courses are two entities and "TAKES" is the relationship between them.

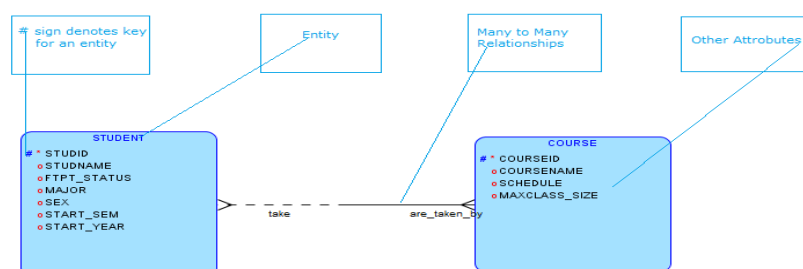
Degree of a Relationship is the number of entity types that participate in a relationship. In the example given above, the degree is two (or binary), since there are two entity types that participate in the relationship.

Cardinality of a relationship (from entity A to entity B) is the number of instances of entity B that can be associated with each instance of entity A. (See the followign for a graphical representation of the E-R model to be created).

You will perform the following major steps:

1. Develop the Logical Model.
2. Develop the Relational Model.
3. Generate the DDL

Final output:



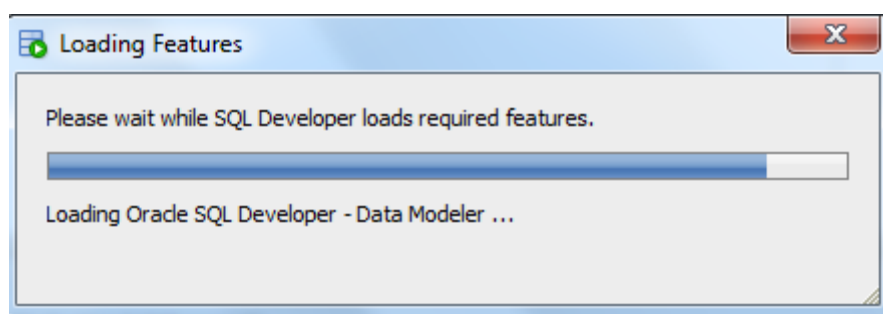
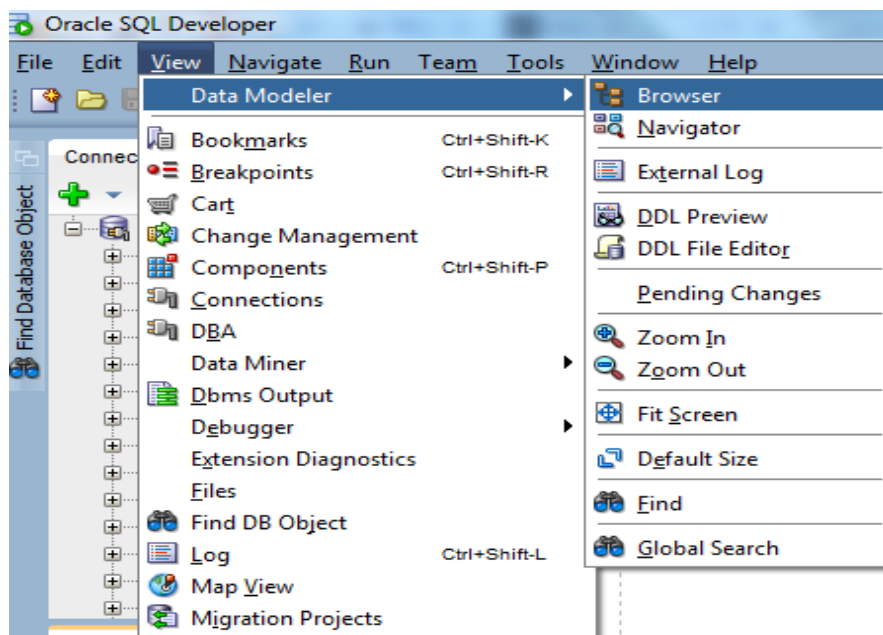
Creating the E-R Diagram Using the Oracle SQL Developer Data Modeler

How to open Oracle SQL Developer Data Modeler

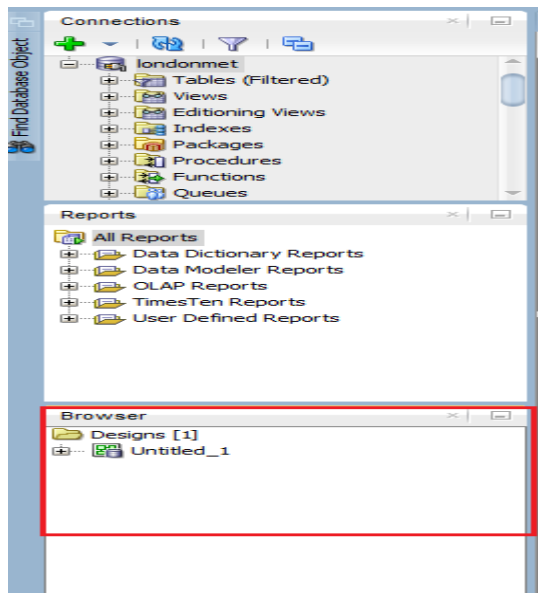
Oracle SQL Developer Data Modeler is a standalone solution that is also shipped in SQL Developer as an extension. The user interface of the Modeler is folded into SQL Developer.

Open or activate the Modeler inside of SQL Developer

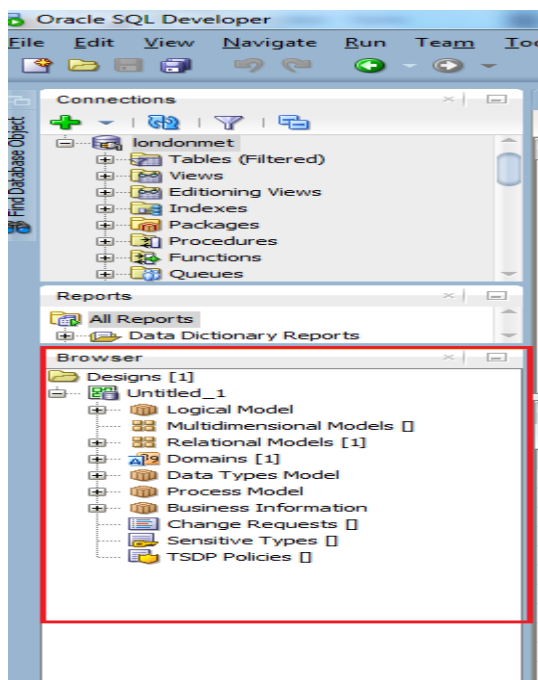
1. Go to the View menu
2. Expand the Data Modeler menu
3. Select the 'Browser'



Now , a Browser object appears on the bottom left corner ,expand Untitled_1



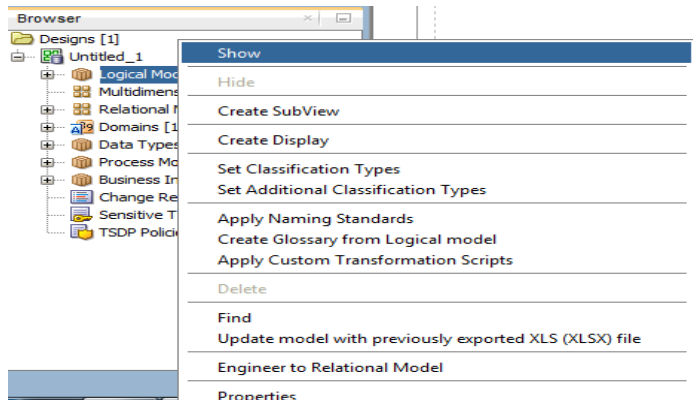
You will now see expanded screen as follow



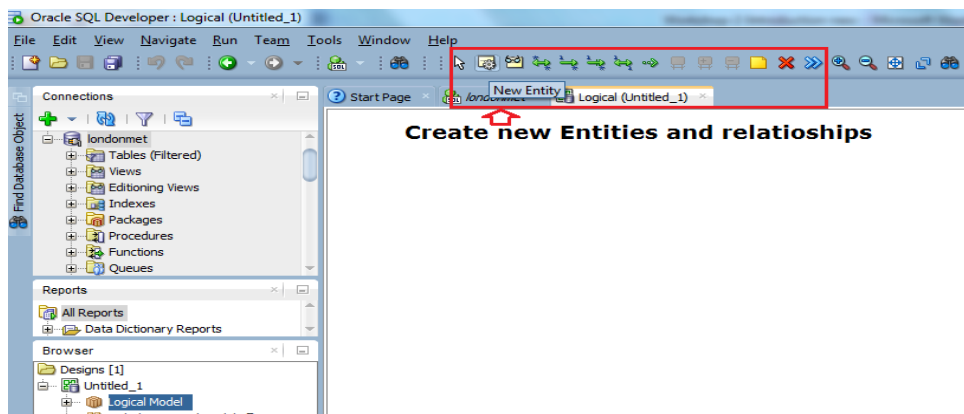
Next we will work on following menu options available at newly created design **Untitled_1**

- a) Logical Models
- b) Relational Model

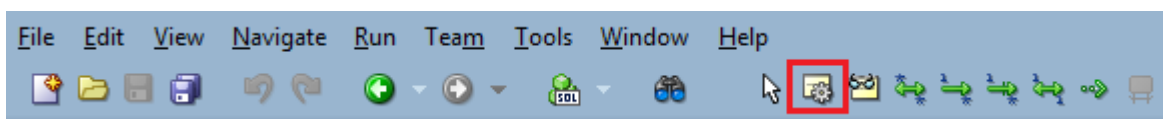
Right click on Logical Model and select show to enable the tool bar for creating entities



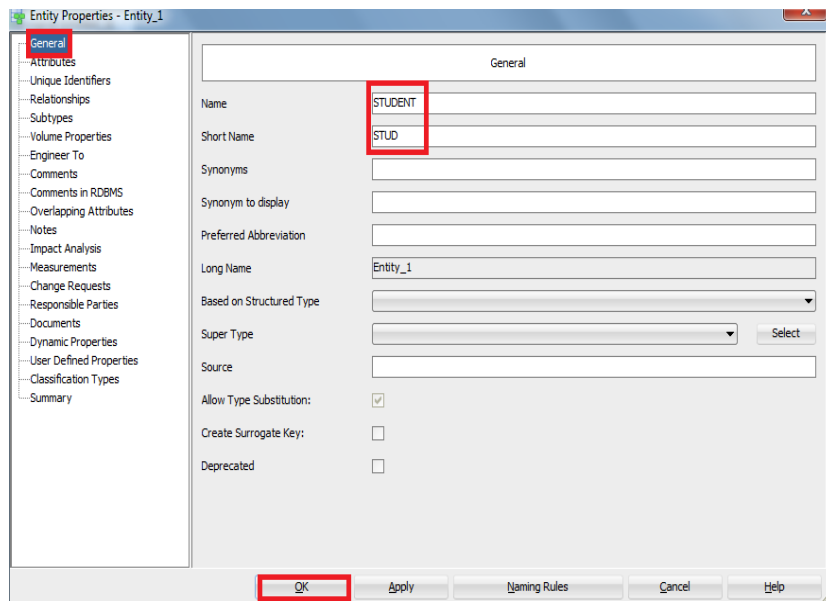
The image below illustrates a logical model selection area where can create entities ,attributes and their relationships



Click the New Entity icon

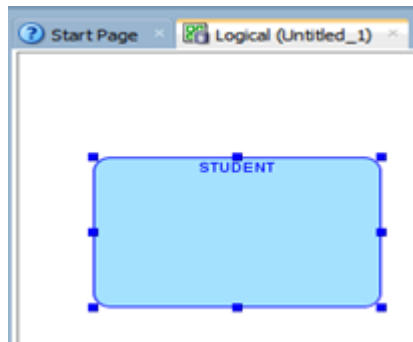


1. Click in the logical model pane in the main area; and in the Logical pane press, diagonally drag, and release the mouse button to draw an entity box. The Entity Properties dialog box is displayed
2. Click General on the left, and specify as follows: (Name: STUDENT, Short Name: STUD) and click OK

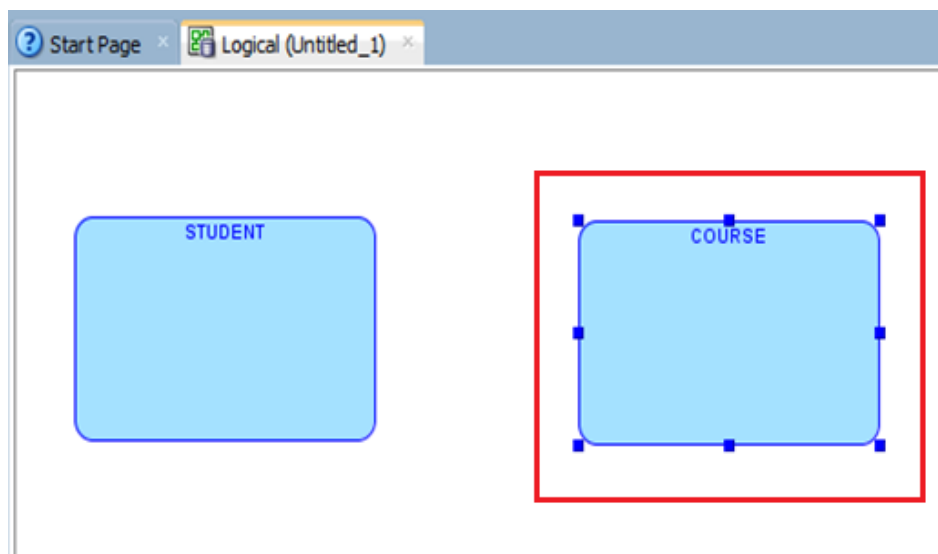


The image shows the 'Entity Properties - Entity_1' dialog box. The 'General' tab is selected in the left-hand tree. The 'Name' field is set to 'STUDENT' and the 'Short Name' field is set to 'STUD'. The 'Long Name' field is set to 'Entity_1'. The 'Based on Structured Type' dropdown is set to 'Entity_1'. The 'Super Type' dropdown is set to 'Entity_1'. The 'Source' field is empty. The 'Allow Type Substitution' checkbox is checked. The 'Create Surrogate Key' checkbox is unchecked. The 'Deprecated' checkbox is unchecked. The 'OK' button is highlighted with a red box.

You will see following output



In the similar way, create another entity **Course**. In the **Create Entity property** window we name it as **COURSE** (with short name **CRSE**, In this way you can create as many entities as you want. Now, your E-R diagram should now look like:



Relations show the relationships between entities: one-to-many, many-to-one, or many-to-many. For specifying the cardinality click **the New M:N Relation icon**. In this case we have chosen the M:N (O to O) relationship.



1. Click first in the Student box, then in the Course box .Relationship property page will appear.

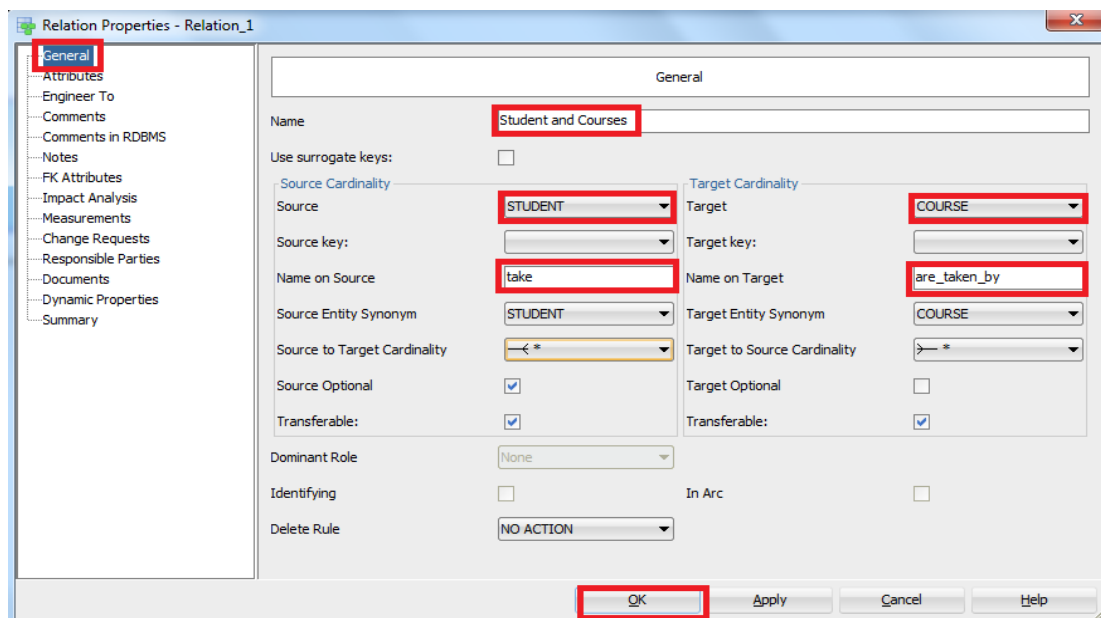
Define values as follows and click ok

Name: Student and Courses

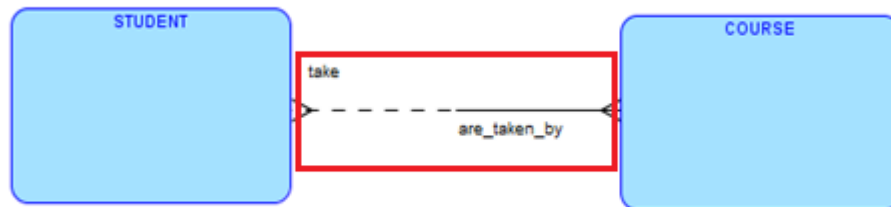
Name on source : take

Name on target: 'are_taken_by'

Here we will type '**take**' in the Name on source and '**are_taken_by**' in the Name on target. (Note that you could have used names without underscore or written in upper case letters. In this case we have used underscore for clarity purposes). In other words, "**take**" and "**are_takn_by**".

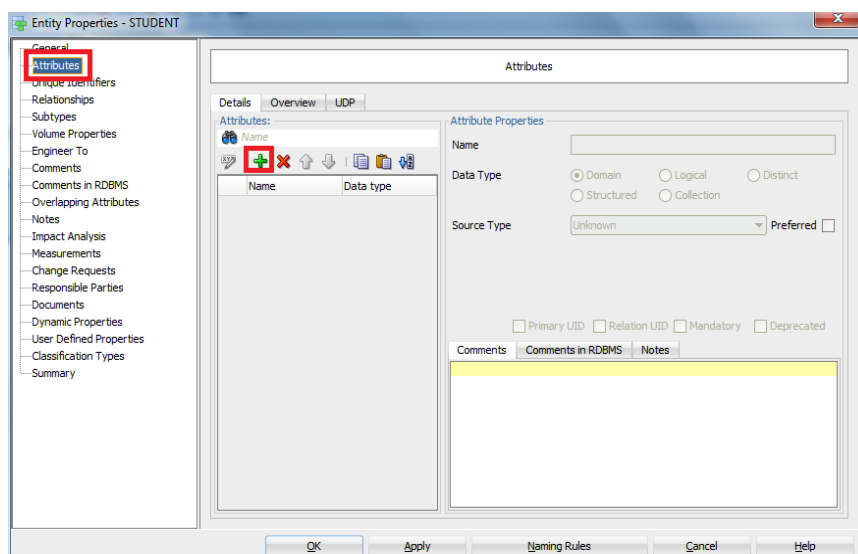


A line with an arrowhead is drawn from Student to Course and Now, your E-R diagram should now look like:



The next step will be to create the attributes for the various entities. In order to do that: Right-click on the **Student** entity and click on **Properties**. You will immediately see the Edit Entity window.

1. Click Attributes on the left, and use the **Add (+)** icon to add the following attributes, one at a time. (For data types, select from the Logical)



The attributes that you will create for **STUDENT** are:

(Note: We are showing you one example, how to add attributes, Remaining you need to add by yourself)

- **STUDID** (primary)
- **STUDNAME**
- **FTPT_STATUS**
- **MAJOR**
- **SEX**
- **START_SEM**
- **START_YEAR**

Fill in the Format, Size and Primary fields as shown in the following screen shot, and click Apply or OK

Note: We are showing you one example, how to add attributes, Remaining you need to add by yourself.

Adding FTPT_STATUS attribute:

Name = FTPT_STATUS

Data Type = Logical

Source Type = VARCHAR

Size = 2

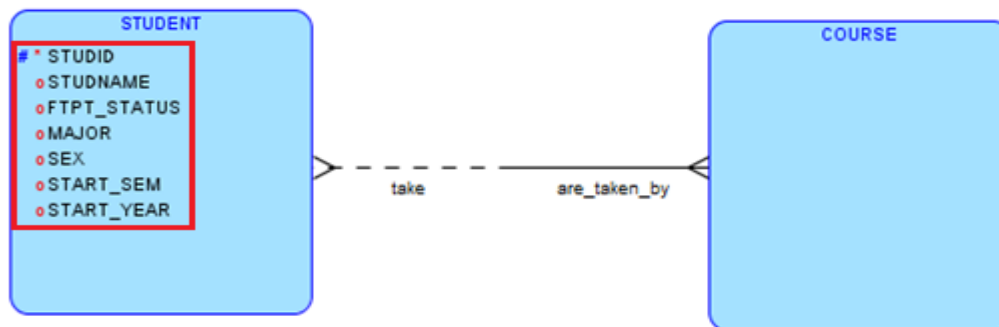
The screenshot shows the 'Attributes' dialog box with the 'Details' tab selected. The 'Attribute Properties' section is visible, showing the following configuration for the 'FTPT_STATUS' attribute:

- Name:** FTPT_STATUS
- Data Type:** Logical (selected)
- Source Type:** VARCHAR
- Size:** 2

The 'Apply' button at the bottom is highlighted with a red box.

Screen should look like this after adding all attributes:

You will see following screen after click OK

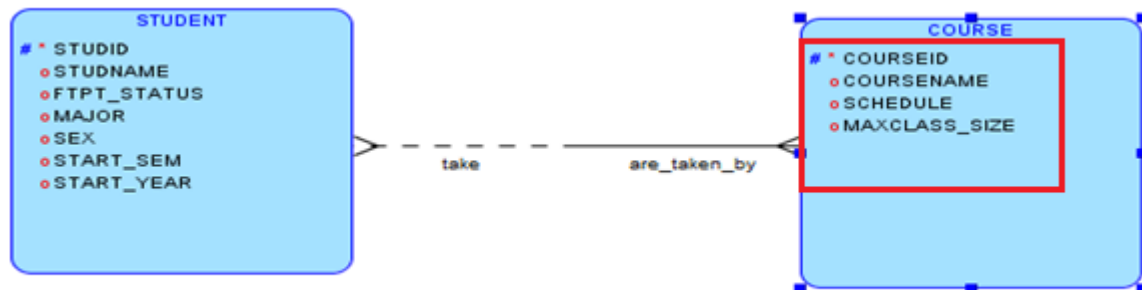


In the similar way create the attributes for **COURSE** entity

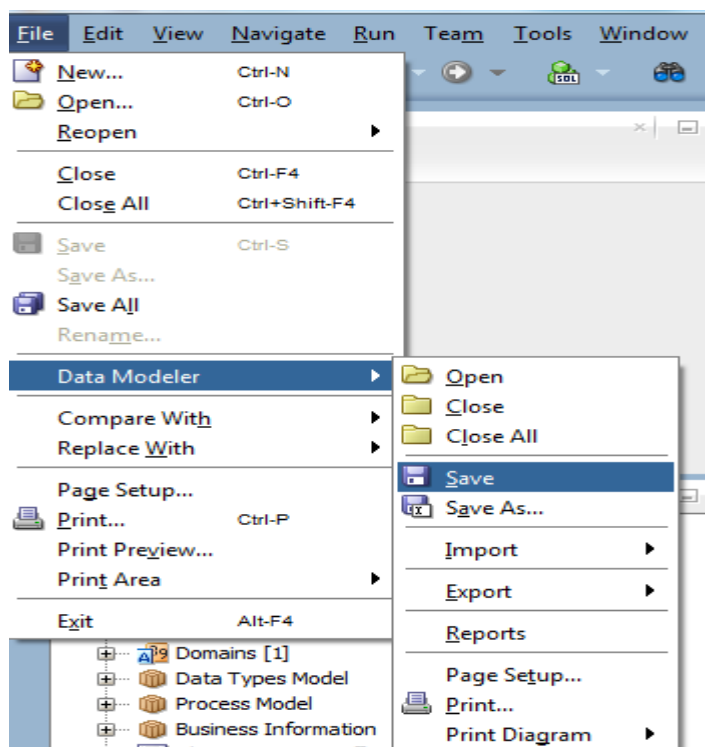
- **COURSEID(primary)**
- **COURSENAME**
- **SCHEDULE**
- **MAXCLASS_SIZE**

and Fill in the Format, Size and Primary fields as shown in the following, then click Apply or OK.

The attributes will be shown on the entity. (The # sign to the left of an attribute indicates that the attribute is the identifier/key for the entity).

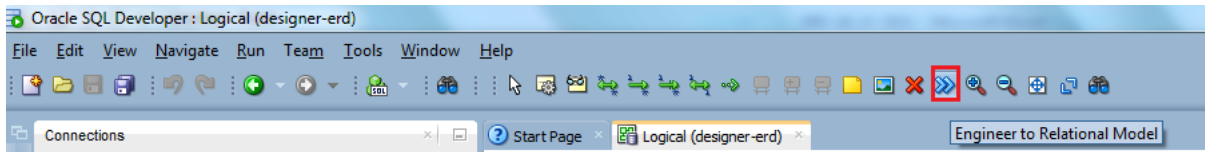


You have now successfully created your first simple E-R diagram. Save your work by File ---->Data Modeler-----> Save As **ERDSRS**

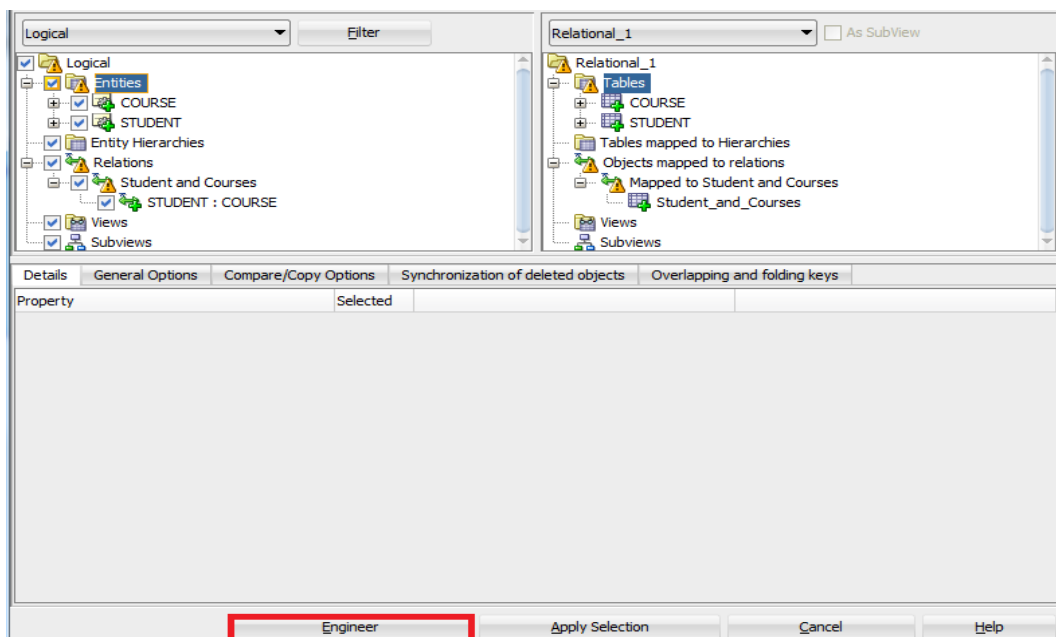


Develop the relational model as follows:

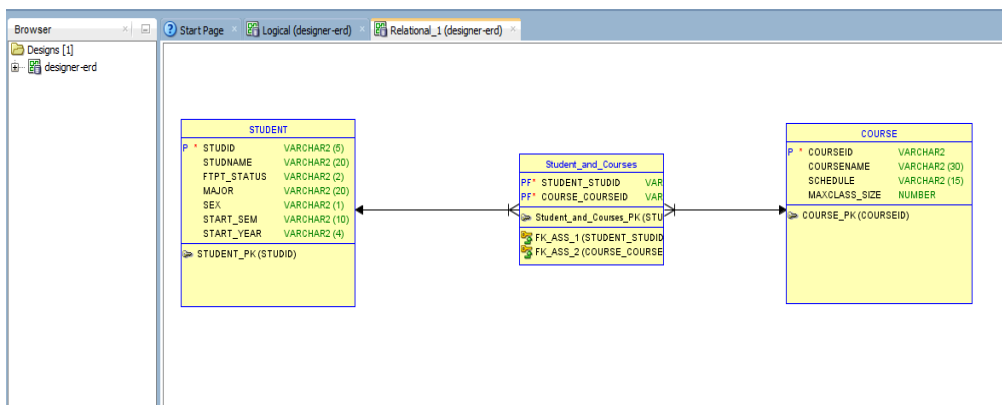
With the logical model selected, click Engineer to Relational Model.



The Engineering dialog box is displayed. Accept all defaults (do not filter), and click Engineer.



You will see following output

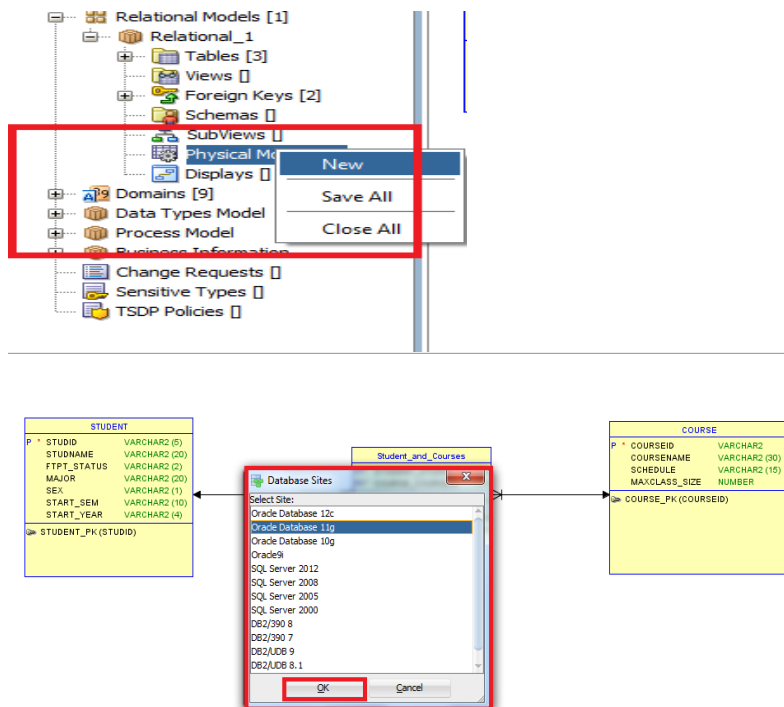


Generate DDL:

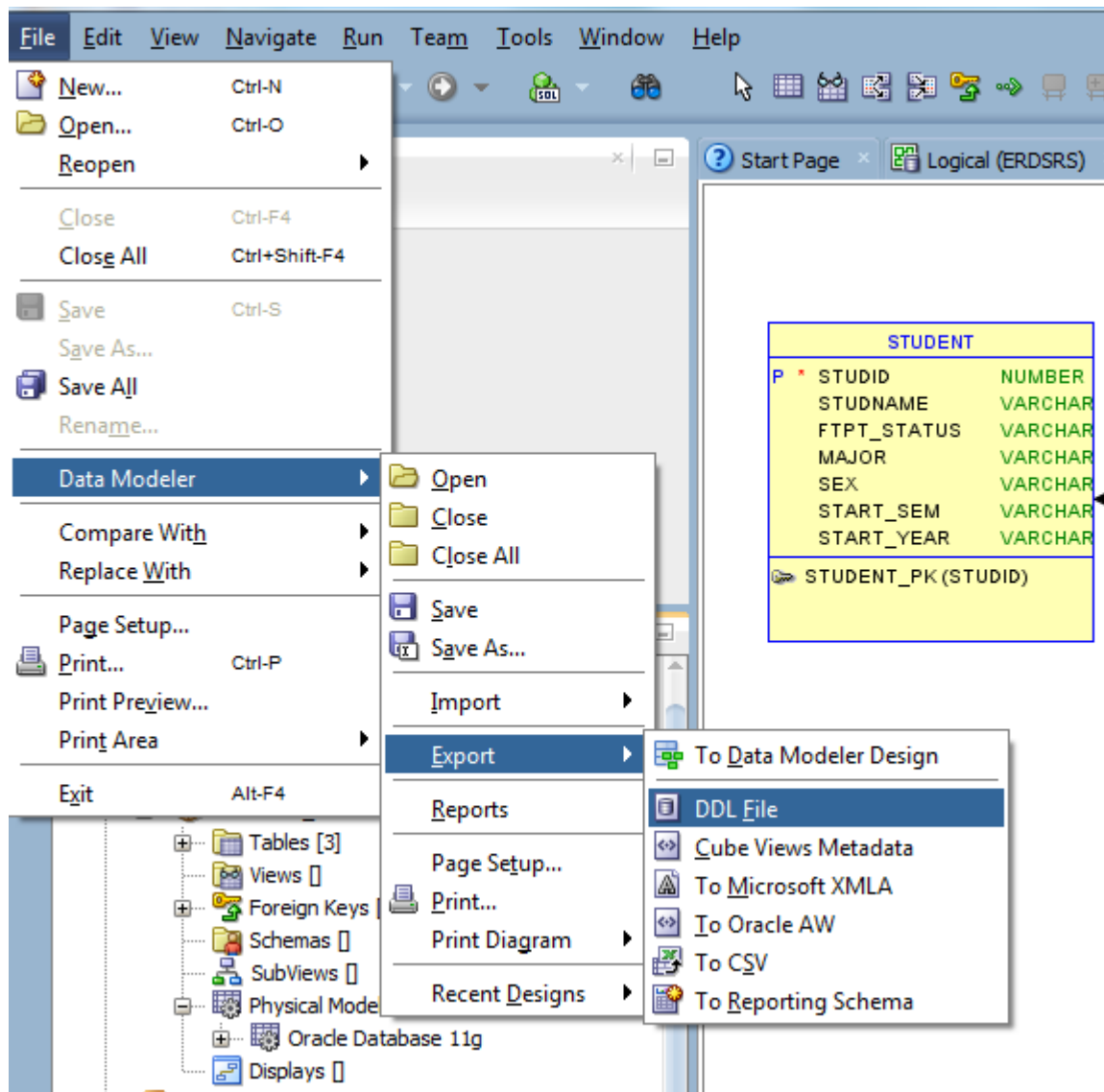
Generate Data Definition Language (DDL) statements that you can use to create database objects that reflect the models that you have designed. The DDL statements will implement the physical model (type of database, such as Oracle Database 11g) that you specify

Develop the physical model as follows:

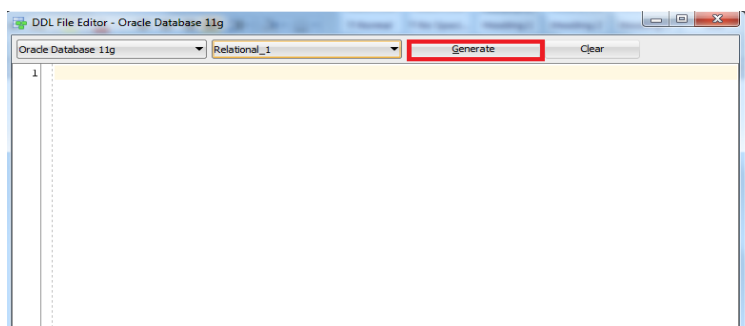
1. Optionally, view the physical model before you generate DDL statements:
2. With the **ERDSRS model** selected, Expand the **Relational Model[1]** then expand **Relational_1**.
3. Right click on **Physical Models[1]** and select **New**. A dialog box is displayed for selecting the type of database for which to create the physical model



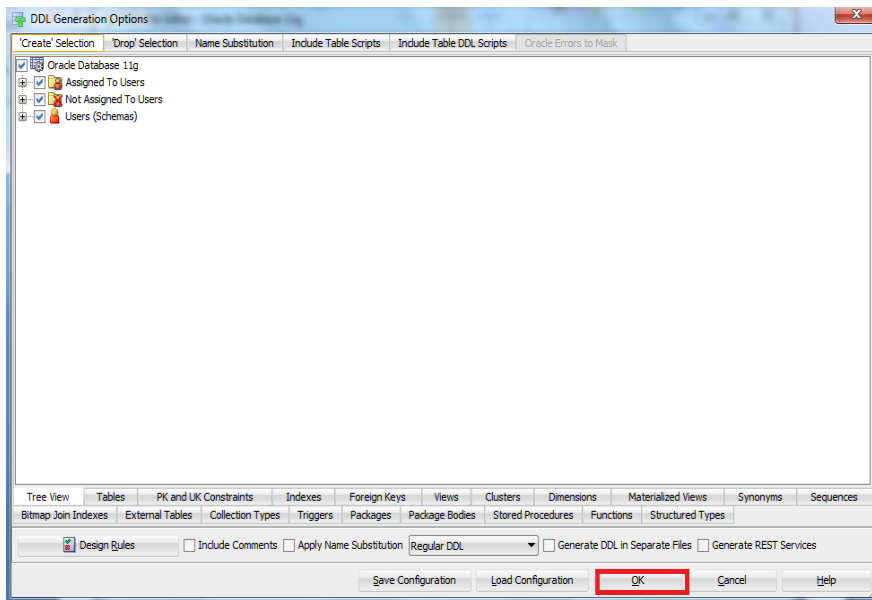
4. Specify the type of database ,**Oracle Database 11g**, and click OK. In the hierarchy display on the left side of the window, a Physical Models node is added under the **ERDSRS** relational model node, and a physical model reflecting the type of database is created under the Physical Models node.
5. Expand the Physical Models node under **ERDSRS**, and expand the newly created physical model and nodes under it that contain any entries (such as Tables and Columns), to view the objects created.
6. Click File----->Data Modeler----->Export-----> DDL File



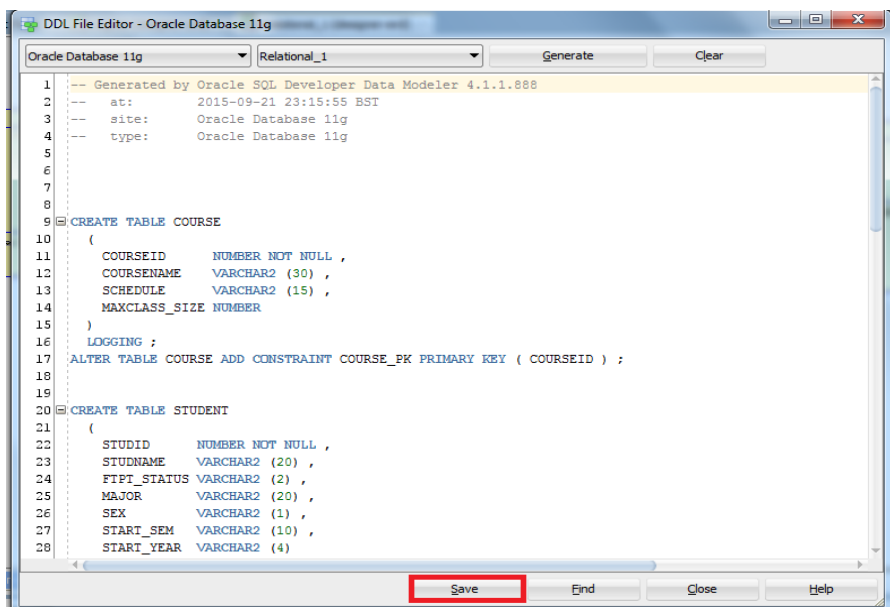
Select the database type (for example, Oracle Database 11g) and click Generate. The DDL Generation Options dialog box is displayed



Accept all defaults, and click OK. A DDL file editor is displayed, with SQL statements to create the tables and add constraints. (Although you can edit statements in this window, do not edit any statements for this tutorial exercise.)



1. Click Save to save the statements to **erdsrs.sql** script file on your local system



2. Run the script (for example, using a database connection and SQL Worksheet in SQL Developer) to create the objects in the desired database.
3. Click Close to close the DDL file editor

