



PURDUE POLYTECHNIC INSTITUTE
Department of Computer and Information Technology

CNIT 27200: Lab #1 – Introduction to Data Modeling

(Part A: 10 points, Part B: 15 points)

Part B is due the night before your next lab by 11:59pm

Objectives

- Apply design methodology by creating an ERD (Entity Relationship Diagram)
- Apply the basic concepts of Entities, Attributes, Primary Keys, Foreign Keys, Relationships and Verb Phrases
- Learn and use course ERD standards

PART A: Entity Relationship Diagram In-Lab Assignment (10 points)

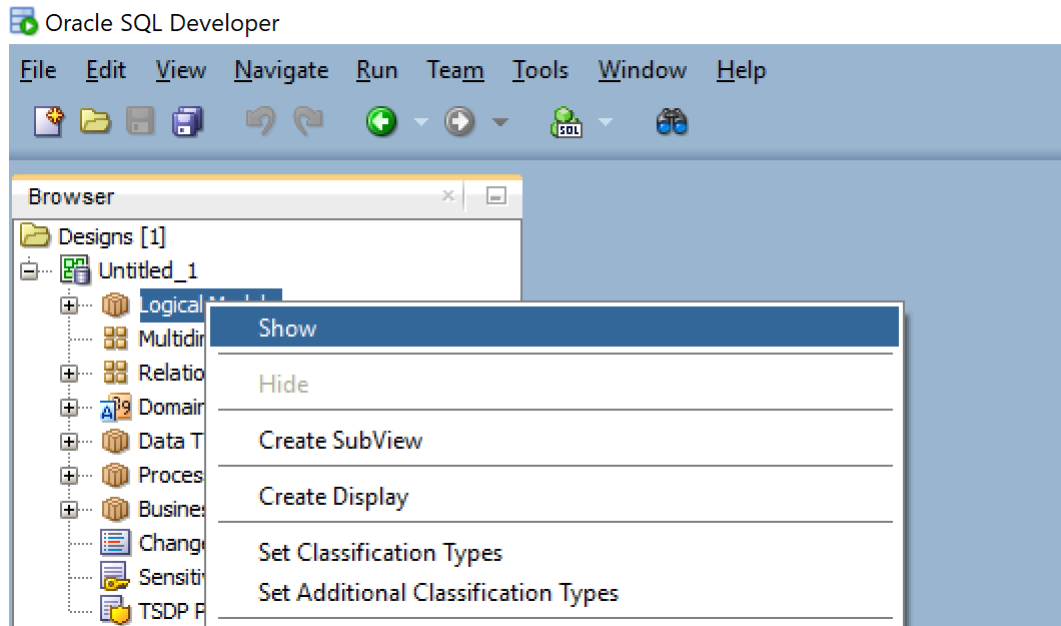
Getting started

1. Start SQL Developer
2. Move shapes and/or connections around to make your diagram more readable. Zoom in and out. Try clicking and double-clicking different places on the diagram to see what happens. The more you understand about the tool early on, the quicker you will be able to use it to make your data modeling more productive!

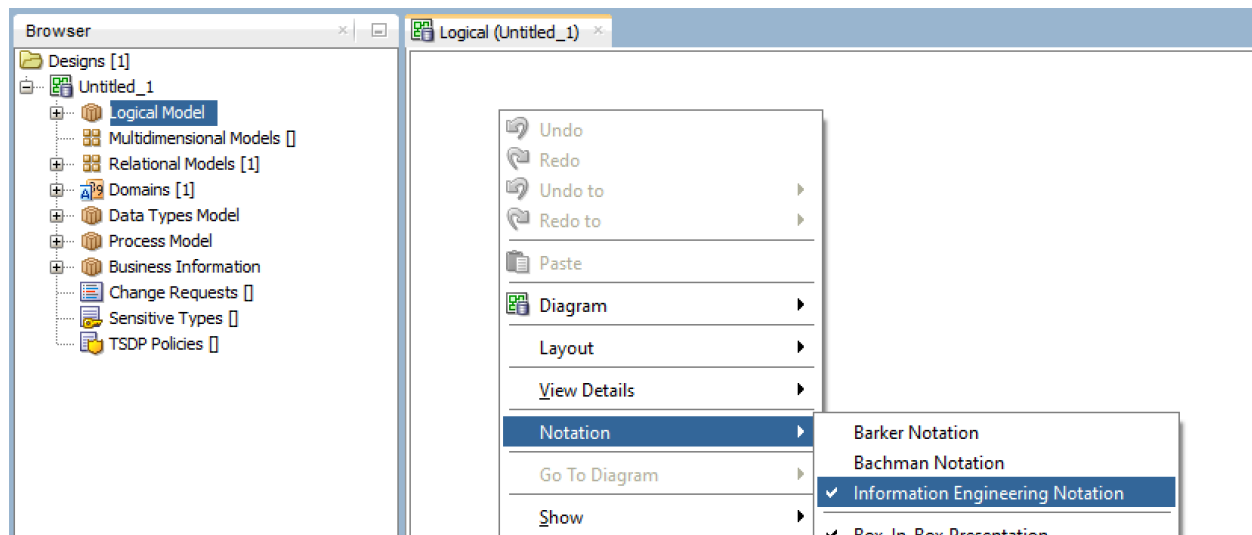
Diagram Setup (Review from Lab 0)

1. Before you start each new data model in this course, you should set the options and preferences correctly for that model. Set up a blank model and save it for future use.
 - a. Open SQL Developer (will take some time to load the first time).
 - b. To activate Data Modeler view, click on 'View' in the toolbar, go to 'Data Modeler' and click 'Browser'.

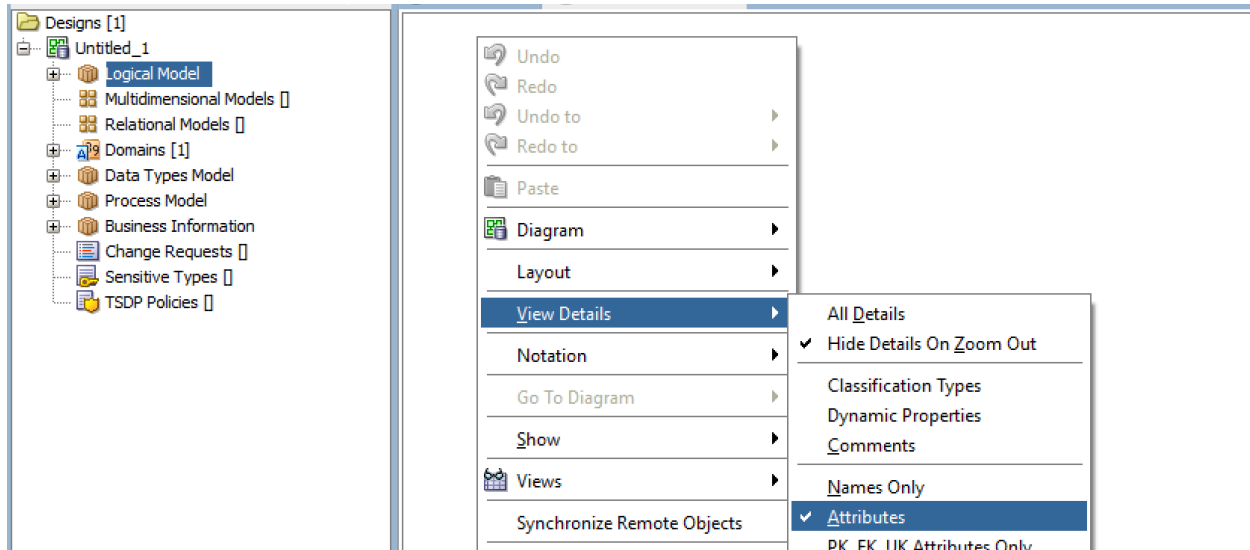
- c. To start a new design file, you should see project under Designs called 'Untitled_1'. Press the + sign to the left of Untitled_1 and right click on Logical Model. Select 'Show'.



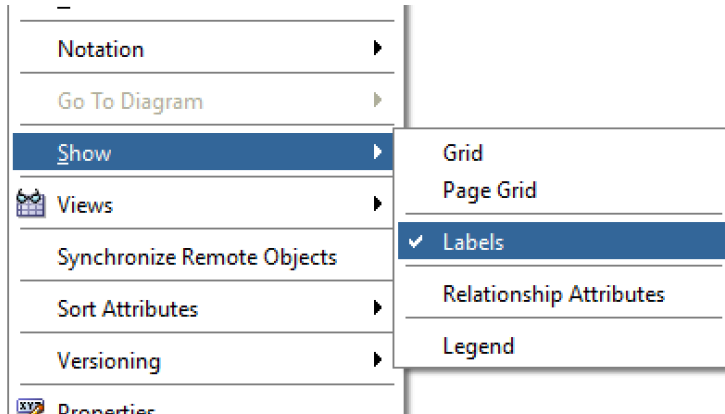
- d. Set logical model diagramming notation to IE. To do this, right-click on the logical design window to the right and select 'Information Engineering Notation' from the menu.



- e. In the logical model, we are not yet concerned with the datatypes. Change the display of the logical model to attributes only. Right click in the white space and select View Details, then click on Attributes. This will just show the attributes as well as the notations for PK and FK attributes.



- f. To have the “verb phrase” visible on the model, right click to get the Show option and turn on “Labels”.



2. *SQL Developer Data Modeler's* toolbox uses a “click and drop” metaphor – you click-and-release the shape or connection you want to use, and then you click-and-release on the drawing location or object to which the selected shape is to apply. You'll quickly discover that the selected object remains the “current” object until you click on a different toolbox pointer (an ‘arrow’) or a different object.
3. It is easiest to type the entity class name right after you place the entity on the drawing page, but you can use the properties tab to modify it later.

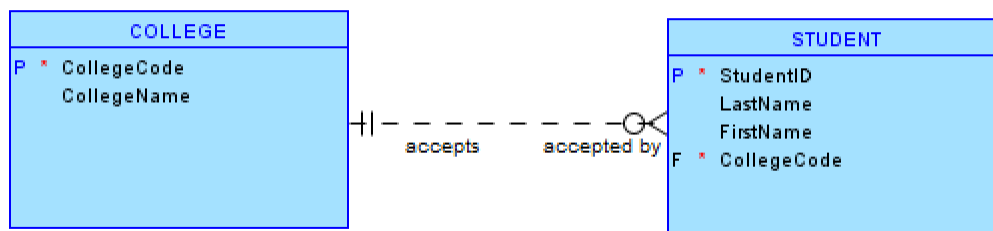
(**REVIEW** the course standards for NAMING standards which can be found on Blackboard)

4. To add relationships between entities, click-and-release on the identifying or non-identifying relationship tool in the toolbox (green arrows at the top); click the parent entity first, then click the child entity. The sequence is important. To move entities, use drag-and-drop. You can also move relationship 'lines' to make your diagram more readable, but it is not as intuitive. First, use the pointer tool to select the relationship.
 - 1) To change the bend-points or line locations, right-click on the relationship line and select "Add Elbow".
 - 2) To move the end-points, drag-and-drop the end point to a new location on the entity.
 - 3) As much as practically possible, draw the ERD with relationship flows from top to bottom and left to right.
5. To customize or modify an entity class or its attributes (including adding attributes or resetting keys), left-click to select an entity, then right click on "Properties".

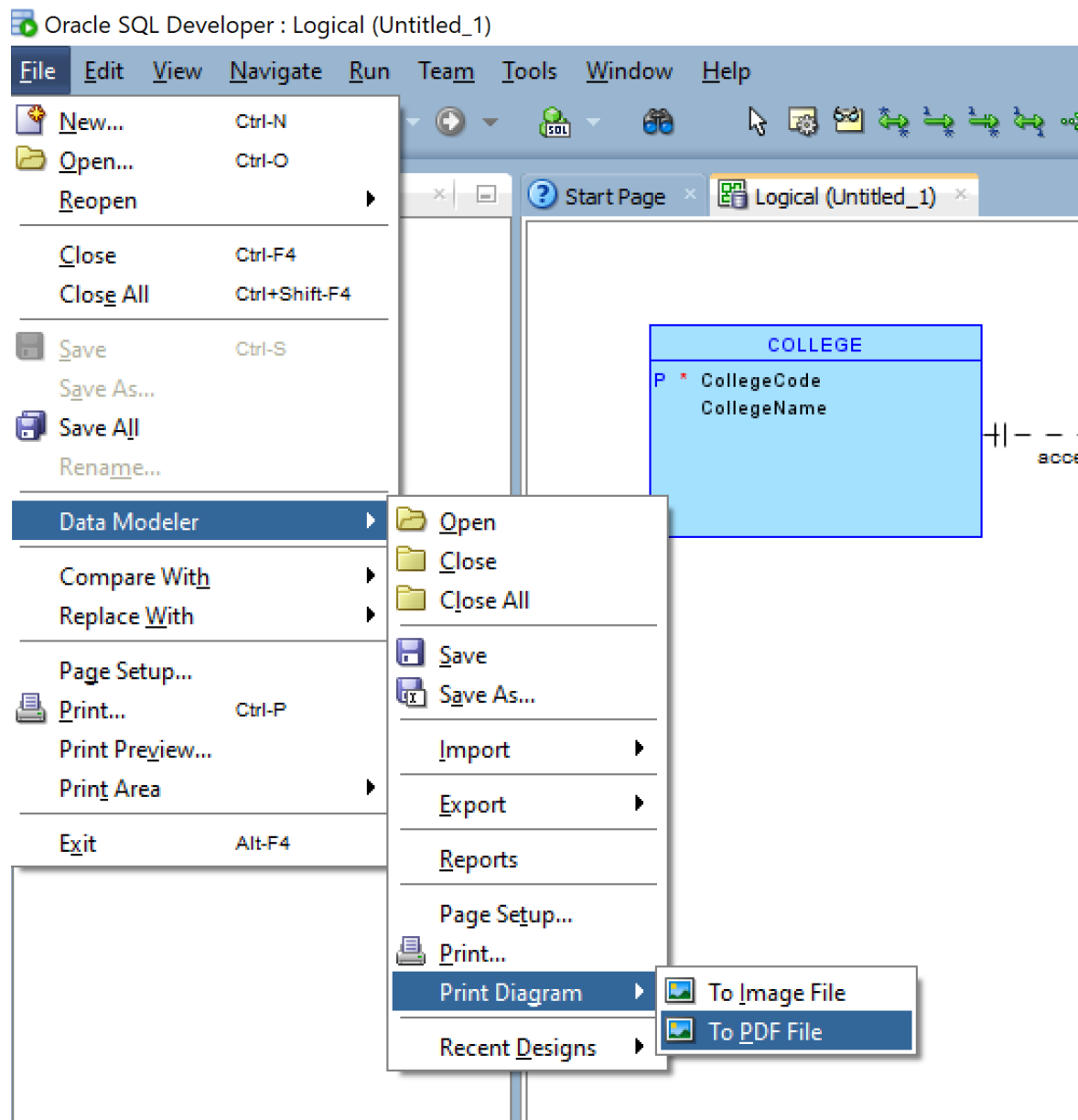
Important: Do NOT enter foreign keys into the related entity. That will be automatically added when you draw relationship classes ("identifying" places the foreign key in the child key area (PF notation); "non-identifying" places the foreign key in the child non-key area (F notation)).

6. Double-click on a relationship to define it. Use the "relationship properties" to specify a verb-phrase or relationship definition in the 'General' section.
7. Remember to use the HELP system when you get in trouble or get confused. Use the HELP/ HELP TOPICS menu for indexed contents help, and the F1 key for context sensitive help (meaning help based on the current state or proposed action for your model.

Here is an example of what your ERD may look like with the proper settings:



8. To save a file in SQL Developer, there are two things that you should do when submitting homework. This seems to be confusing, so be sure to cover this with the TA if you are unsure.
 - a. The very first thing to do is create a new folder for saving your diagrams. Each diagram will create a folder to accompany the .dmd file. Create a folder called CNIT272ERD.
 - b. Always start by saving a PDF of your entity relationship diagram. It is added security in case you lose your work. No one wants a zero on an assignment for a blank file. Go to File -> Data Modeler -> Print Diagram -> To PDF File. Save it in your new CNIT272ERD folder with your name in the PDF file (ie, Smith_Lab0_Practice.PDF)



- c. Once you are done with the ERD and saved it as a PDF, click on the Browser tab to the left and under Designs, you will see that your current Design is called Untitled_1 (most likely Untitled_1 unless you opened multiple new designs) if you have never saved it. Right click on the Untitled_1 name and then select 'Save Design'. Select the CNIT272ERD folder that you created and save it there.
 - i. It will save as a .dmd file and also add a folder with the same name. Use the same name from the PDF file. Make sure you select the design that you have open from the list, otherwise you will just save a blank design. Do not move the .dmd file from the location where it saves.
 - ii. Alternatively, if the Save Design option does not work, you can also try File -> Data Modeler -> Export -> To Data Modeler Design. Then move the Logical model to the Selected Models box, and press the Export button at the bottom of the screen.
9. Verify that you can open the saved file after you have closed SQL Developer. To reopen a file:
 - a. Open SQL Developer first. Do not try to open the .dmd file by double-clicking on it.
 - b. Go to File -> Data Modeler -> Open. Go to your CNIT272ERD folder. Find the .dmd file and open it.
 - c. If you try reopening your file and it is blank, let the TA know so that it can be resolved for future labs.

Use the following scenario to develop an entity relationship diagram (ERD) in SQL Developer:

Part A ERD: Tree Removal (in-lab assignment)

Assignment Part A:	10 points
<ul style="list-style-type: none"> • In class only (nothing to be turned in, but it must be checked off by the lab instructor for points). • Include entities, attributes, primary (P) and foreign keys (with F or PF displayed), and relationships with verb phrases. • Use the course naming standards. • Hint: You may want to save your work as it will be useful as a reference for Part B assignments and also as study material for exams. 	

A small tree removal company wants to keep track of service requests, the properties it has serviced, and the equipment used to service the properties.

Queries that might be requested from this database:

- Given a Property ID, identify its property dimensions, initial contact date, property type, and the property owner's name (separate first and last name in the DB).
- Given an Equipment Code, identify the manufacturer, purchase date, depreciation amount, and warranty of the equipment item.
- Each service request is assigned an InvoiceID. For each service record, list the invoice ID, Property ID of the property requesting service, service request date, service labor fee, description of service, and billing date.
- A service request may include multiple items of equipment. So given an Equipment Code and Invoice ID, include the service fee for using the equipment, a description of how the equipment was utilized, and condition of the work elements for each item of equipment used.

BUSINESS RULES:

- A property ID uniquely identifies each property in the database.
- An equipment code uniquely identifies each item of equipment that the tree removal service owns.
- The service request may require multiple items of equipment.
- The request date is the date that the service is performed on the property.
- The invoice ID uniquely identifies each service request.

Deliverable for Lab 1 Part A:

1. Create an Entity Relationship Diagram (ERD) USING COURSE NAMING STANDARDS which includes these logical design elements:
 - a. Entities
 - b. attributes
 - c. relationships
 - d. primary key and foreign key designations
 - e. parent-to-child verb phrases
2. Think about what is needed to make each instance in an entity unique. What would be a proper PK based on the business rules.
3. Do not leave the lab without the TA checking off your final ERD for Part A.

Continue to Part B out of lab assignment...

PART B: ERD Practice in SQL Developer (15 points)

Part B is due the night before your next lab by 11:59pm

Assignment Part B (Due via Blackboard)		15 points
<p>The case for this exercise is an Access database sample that pertains to a vehicle purchase. (Note: We use this model type because most of you are familiar with Access. That allows you to concentrate on the SQL Developer Data Modeler Software.) In particular, notice the notation differences in Access vs. SQL Developer data modeling.</p> <p><u>You may discuss your lab assignments with classmates, but each student is required to create and synthesize their results individually in one's own document.</u> Copying an ERD, result set, and/or a report off of someone else is academic dishonesty and will be addressed as such resulting in a zero for <u>all</u> students involved. This is individual work, and not collaborative or cooperative.</p>		
<input type="checkbox"/>	Given the Access developed database model image at the end of this document, <u>re-create</u> it as a logical model using the <i>SQL Developer Data Modeler</i> tool (don't do anything with the Physical yet--only your "logical" work will be evaluated). Add two additional entities to the original model - Create two entities that would contribute additional data to the database. One of the relationships should be identifying and one should be non-identifying.	NOTE: A screenshot of the database model is at the end of this document.
<input type="checkbox"/>	Utilize Part A's SQL Developer Data Modeler Notes: Getting Started and Diagram Setup sections. It is important that you begin each new model with the correct settings.	

<u>Deliverables:</u>	
<input type="checkbox"/>	Step 1: the Access version and your SQL Developer Data Modeler version of the database <u>should start with same number of entities and attributes.</u> Rename attributes and entities according to course naming standards (for the attributes, the names must be different since the Access version does not use our naming standards).
<input type="checkbox"/>	Step 2: <u>Add two additional entities to the database</u> with the proper relationships to other entities in the existing database. Include at least 4 attributes in each of your new entities (at least one of the 4 attributes should be the PK). Questions to ask as you think about what entity to add: What additional data could contribute to this database? What could also be tracked? One relationship should be identifying and one should be non-identifying

<input type="checkbox"/>	Remember that foreign keys will automatically be set by SQL Developer when you click the parent and then the child entity. Use the correct relationship (identifying vs non-identifying) in your ER diagram.
<input type="checkbox"/>	Include parent-to-child and child-to-parent Verb Phrases to “describe” the relationships (not included in the Access model).
<input type="checkbox"/>	In a note on the design (use the ‘New Note’ tool), include a database title, your name, Lab 1 Spring, completion date/time, and your lab session time. Add an explanation of your 2 additional entities in a text box as well.
<input type="checkbox"/>	Save the model in the default SQL Developer Data Modeler format. Review how to properly save your model, and also how to export to PDF.
<input type="checkbox"/>	Submit both the SQL Developer Data Modeler file <u>and</u> a PDF of your ERD design via Blackboard.

Notes and Helpful Hints:

Change all names to conform to the course naming standards for all entities and attributes (Note: this means changing something like lname to “LastName” or “Last_Name”, or an entity called Customers to “CUSTOMER”).)

- Let *SQL Developer Data Modeler* create all of your foreign key attributes. A common error is to have a foreign key and non-keyed attribute with the same name: CustomerID and CustomerID1. This is an indicator that you manually entered in the attribute for the FK and also used the software to create the FK.
- Use an online thesaurus to think of appropriate verb phrases. You should be able to read the relationship as a sentence. “*A customer initiates many orders.*” “*Each order is initiated by a customer.*”
- Make your data model easy to read. Move the placement of shapes, connections, bends, labels, etc., as needed. Try to make the model “flow” from top to bottom and left to right.
- You can find the Access database model reproduced below as a screen shot.
- Pay attention to whether the relationship is identifying or non-identifying based on the foreign keys.
- Your additional entities and subsequent attributes should appropriately relate to the existing entities in the database.

