PURDUE POLYTECHNIC INSTITUTE
Department of Computer and Information Technology

CNIT 27200: Lab #2

25 pts

Due Date:

Part A is due within the Lab session   10 pts.

Part B is due **the day before your next lab by 11:59 p.m.**   15 pts. It must be submitted via Blackboard.

Objectives:

- Create the **logical and relational** models of the ERD (Entity Relationship Diagram) for a simple case.
- Learn to assign data types and sizes for the physical model.

Introduction:

SQLDeveloper Data Modeler allows you to view a model at various “display levels” in logical display mode and relational display mode.

- In this course, we commonly work at the “attribute” level in the logical model and the “column” level in the relational model. The logical model is independent of any DBMS. Because the relational model includes data types (with sizes, etc.), the relational model is **dependent on a specific DBMS**. This is because the data types vary from one DBMS to another.
- You will first build the logical model of the ERD, and then engineer it into the relational model.
  - To engineer the logical model to the relational model, click on the button above with the two blue arrows. If you hover over the button, it will display “Engineer to Relational Model”
  - A box will display with the logical information to the left and the relational information to the right. Click on the Engineer button at the bottom of the box.
- To have the models appear in Information Engineering (“crow’s feet” format), you need to be sure that **notation is set to IE.**
- **To see the datatypes in the relational mode, right click on the white space when in the Relational mode, select View details-> All Details**

- For Oracle data types to be used in the relational model, be sure Oracle is selected when a new Relational model is created.
- When selecting datatypes, double click on the entity you want to work in first. Once the Table Properties opens, select Columns on the left. For each attribute, you can change the Datatype.
    - To change the datatype, select the Logical bullet (these are generic datatypes)
    - Then pull down the Type menu to select the datatype. Once you have applied each datatype for each column (attribute), you can enter a size if the instructions state to do so.
    - After all have been updated, select the OK button.
    - When you go back to the Relational ERD, you will notice that the datatypes match the Oracle specific datatypes. Using a RDBMS other than Oracle will result in incorrect datatypes for this course.

<u>Before beginning the assignment read (or re-read):</u>

- Standards (in the Course standards folder in Blackboard Learn).
- Oracle Data Types (choose your source of choice – document or Internet)

<u>Tips</u>:

- For both Part A and Part B, you are making best-guess decisions about the verb phrases, cardinality and the data type and sizes.
- In SQL Data Modeler, the changes between the Logical tab and the Relational tab are not dynamic. Do not create the relational model until you are satisfied with the design of the logical model.
    - **Any changes between the two tabs will not be automatically updated.**
    - **If you need to make a new relational model, delete the first relational model. Then forward engineer it again.**
- There is not always one right answer, but there are reasonable answers and there are definitely wrong answers.
    - For example, choosing the wrong data type for “date” data is wrong.
    - You don’t have all the specifications, but you should make reasonable decisions.
    - Do not use “has” as a verb phrase.

Assignment Part A:	**10 points**

- In class only (nothing to be turned in, but it must be checked off by the instructor).
- Include entities, attributes, primary and foreign keys (with F displayed), and relationships with verb phrases.
- Use the **course naming standards**.
- Hint: You may want to save your work as it will be useful for the project and also for studying for the exams in the lecture portion of the course.

Using SQL Developer Data Modeler, develop a **Logical** ERD for the following case. Show entities, attributes, primary keys, foreign keys, relationships with cardinality and parent-to-child verb phrases. Do not create the relational model until you are satisfied with the design of the logical model. Any changes between the two tabs will not be automatically updated. Complete the **Relational Model** with appropriate (Oracle) data types and sizes.

Case: Consultant Database

You are working for a consulting company and want to create a very simple database. This database will be used to track issues clients are having and the consultants who are working on the issues. The consulting company has contracted with many companies to offer this service. The database should track information about the contracted companies, and the company that each client works for currently. Each client can have many support tickets open at once and many clients could be facing the same problems. Each time a client reports a problem, the database should track information about the client, what their problem is, the consultant who is working on it and the date it was logged into the system, its status and the date the problem was solved (note this is blank for open tickets).

The database needs to support these possible queries:

- Given a Client ID, list the client's first and last name, address information, and email address.
- Given a Company ID, list the company's name, the contract date for service, the terms of the contract, the company contact name, and the company contact email.
- Given a Consultant ID, list the consultant's first and last name, hiredate, phone and office.
- There is a knowledge base of issues to choose from when creating a support ticket. Each issue is stored in the database and assigned a unique code, as well as details about the problem, the level of the issue, and the initial date that the issue was added to the database.

BUSINESS RULES:

- Consultants also work on many issues at any given time.
- The client can have more that one issue open in the database.
- The client can also experience the same issue more than once.
- The issues are not unique to the client. Many clients can experience the same issue at the same time.
- A company has more than one client in employment, but a client is only employed by one company.

Assignment Part B (Due via Blackboard)	**15 points**

Part B is due **the day before your next lab by 11:59 p.m.** 15 pts. It must be submitted via Blackboard.

REMINDER: 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. 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 **all** students involved. This is individual work, and not collaborative or cooperative.

In SQL Data Modeler, the changes between the Logical tab and the Relational tab are not dynamic. **Do not create the relational model until you are satisfied with the design of the logical model. Any changes between the two tabs will not be automatically updated. If you need to redo the relational model after updating the logical model, delete the existing relational model, and then create a new relational model.**

Case: Property Tracking Database

A property tracking company has called you, a database consultant, to create a database to track data about its properties. Based upon interviews with the managers and office staff of the company, you find out that the database needs to support these queries and business rules:

- Given an employee ID, produce a list with the employee's first and last names, hire date, cell number, and realtor license number.
- Given a potential client, assign each client with a unique Client ID. The database should also store the first name and last name of a client, as well as a contact cell and email address.
- Each property has an assigned property number. Given a property number, list the street address, square footage, year the property was built, and property type.
- Each property can be viewed by a potential client. The database should track each date that a property is viewed, which client is viewing the property, the employee that provided the viewing, as well as any feedback that the client would like to provide from the activity.
- The property tracking company is also responsible for the inspection of their properties. There are various inspectors on contract with the property company. Given an Inspector ID, list the company name, hire date, the estimated hourly fee, the inspector's first name, and inspector's last name.
- The company would like to keep track of any inspections performed on their properties. Each instance should include the inspection date, property number, the inspector, the reason for the inspection, the number of hours it took to complete the task, and the actual inspection fee.

BUSINESS RULES:

- Not all employees are responsible for viewings. So there may be employees in the database that never host a property viewing.
- Clients may visit many properties in one day. A property can also be viewed by many different clients in the same date.
- There may be clients in the database that are added to the database before viewings are scheduled.
- Properties can be in the database without any inspections.
- A property may have a number of different inspectors on the same day with different objectives (one for plumbing, one for electrical, one for lead testing, etc). The contract inspector may also inspect multiple properties in the same day.

Deliverables: Submitted file with the SQLDeveloper Data Modeler **logical and physical (relational)** models built, including these characteristics:	
□	Appropriate Oracle data types; IE formatting.
□	Include parent-to-child and child-to-parent Verb Phrases to "describe" the relationships
□	F designation should be visible, to identify Foreign Keys.
□	For the relational model, select appropriate data types (NUMBER, VARCHAR2, CHAR, DATE, etc) and sizes for the attributes based on the case description. (Note: you will need to guess at sizes; use what you believe to be a good size considering the business.)
□	In a text box on the design (use the text tool), include a database title, your name, date/time of completion, and your lab session time.
□	Print the logical model diagram to a PDF, and also print the relational model diagram to a PDF. Save the model in the default SQLDeveloper Data Modeler format. See Lab 0 for directions.
□	Submit 3 items the Blackboard Lab 2 Assignment Box: 1. A PDF of the Logical Model 2. A PDF of the Relational Model 3. A zipped file of the SQL Developer Data Modeler folder and .dmd file Use the instructions from Lab 0 if you need to review how to print to .pdf in SQL Developer.