**Project Milestone #2 – Physical Database Design and Implementation**

**Overview:**

**Task 1: Update Your Logical Database Design from Project 1**

First, make sure your ERwin data model from Project 1 is a logical/physical data model that uses MS SQL Server as the database. If you have created a logical only data model or pointed to another type of database in Project 1, you will need to create a new logical/physical model that uses SQL Server as the database, copy and paste everything from your previous model.

Next, update your ERwin data model based on instructor’s review and feedback on Project 1. Make sure you have the correct primary keys, foreign keys, data types, relationships, and cardinalities for these tables.

If you don’t specify the data type for each attribute appropriately, you may encounter errors during the forward engineering process because SQL Server will not accept inappropriate data types, such as precision and scales for decimal type of data. For example, use Decimal (10,2) for prices instead of Decimal (). Similarly, if you don’t specify the field width for each attribute/column as described in the table, data may be truncated during the data populating process.

**Task 2. Create Database in SQL Server Using ERwin Forward Engineering**

Use ERwin forward engineering function to automatically create the database schema in Microsoft SQL Server. Save the .ere file generated during the process. Watch the YouTube video (link posted on Blackboard -> Course Documents -> YouTube videos) to learn how to forward engineer ERwin model to SQL Server.

**Task 3. Normalize and populate the data**

**3.1 Normalize the data**

Next you need to normalize the data provided in “Project 2 Starting Data Summer 2017” to the 3rd level, and populate them to your MS SQL database tables created from above.

In addition to that the business rules in Project #1, more hints are listed in the following:

Products with the exact same name are considered as the same product. In your normalized data, you will need to create a product ID starting from 1, incremental by 1, for each product that has a unique name. Sort the products by name alphabetically and then assign each a product ID, for example, “Bandages (Box of 1000)” will have product ID 1.

All products that PSC sells are purchased from suppliers. A product can be purchased from multiple suppliers at various costs.

Some products may not have been ordered yet, such as Silicon Spatula.

The Quantity in the "Customer Orders Detail" is the Order Quantity associated with each item ordered by each customer, and should go into ORDER\_LINE\_T.

The Quantity in the "Product Supply Detail " is the Supply Quantity that should go into PRODUCT\_SUPPLIER\_T. Because the same item/product can be purchased from multiple suppliers, so the sum of the quantity you can find for this item/product is the Stock Quantity that should go into PRODUCT\_T.

The Item Price in the "Customer Orders Detail" is the Unit Price the product is sold for, and should go into PRODUCT\_T.

The Cost in the "Supplier Purchasing Details" is the Supply Unit Cost and should go into PRODUCT\_SUPPLIER\_T.

PSC decided that one product can be in one and only one category, while one category contains one or more products.

Lastly, do not forget that in “Customer Orders Detail”, “Customer 3876 wants to add 1 snow mobile for $5,400 to their order 1530. Please populate the database with this new information as well.”

**3.2 Populate the data in MS SQL Server using SQL**

When you populate the data, be aware that existing integrity constraints will force you to enter data in certain orders. For example, customer ID is required in Order\_T so you cannot populate the orders before you populate the customers. So a good approach is to write down the order you populate the tables with data following integrity constraints. For example,

CUSTOMER\_T -> ORDER\_T, PRODUCT\_T -> ORDERLINE\_T…

One way to populate the data is to use INSERT SQL statements (SQL Server Management Studio -> New Query). When you use SQL insert statements to populate the data, be aware of the columns that do not have any data, enter a pair of empty quotes (,‘’,) for empty string type of columns, and null (,,) for empty number type of columns, otherwise the SQL Server will not execute your insert statement. Note that date is text-based too so when you insert a date value don’t forget to use ‘’, otherwise incorrect value will be inserted. Alternatively, you can insert values into selected columns instead of all columns.

Sometimes you will realize your database structure is built incorrectly once you started to populate data into it, and because you created the database structure using ERwin Forward Engineering function, it may have limitation of what you can change afterwards – such as changing the data type for a non-key column in the table design in Management Studio. You may first need to turn off the default “Prevent saving changes that require table re-creation", by going to top menu, select Tools -> Options -> Designer, uncheck the "Prevent saving changes that require table re-creation" option. In most cases you probably will find it easier to detach and delete the database, correct your ERwin model and re-create the SQL Server database from ERwin.

Another way to populate the data to import the data from an Excel spreadsheet using the Query Wizard, however the same integrity constraints apply, so you will still need to populate the tables in order.

Lastly you are able to populate the data manually in Design View in Management Studio – this is not preferred because it is highly labor-intensive and subject to human error. The method may work with this class project but not realistic in real-world scenario. Also when you do this do not turn on auto-generate identifier option.

**What to include in your project report:**

* Screenshot of your revised ERwin diagram (1 full page, use landscape layout if needed)
* Make the screenshot big enough to be readable. Use “Landscape” instead of “Portrait” setting in Word if you can.
* Each page of your report should show the screenshot of one table structure (Right click on the table name and select “Design” or use Design View), AND the screenshot of the data in that table (right click on the table name and select “select top 1000 rows” or use Data View). And make sure your table data screenshot clearly shows the number of rows returned in the right bottom corner. Print table name on top of page.
* Make each screenshot is big enough to be readable but it should take about half page (in Word) not entire page like ERwin screenshot.
* In all screenshots, show your SQL server name and all table names in left explorer pane in Management Studio. You can take multiple screenshots if you have to in order to capture the entire result set (i.e., all records), but make sure to crop each screenshot and put them together so they are readable as a whole, and without repeating data.

**Submit your project to Blackboard:**

* **Submit the project report in Word:**
  + Include a cover page: Project Milestone #, course number and name, year and term, first and last name, and project due date.
  + Organize content in the exact order specified (i.e. Task 1, 2)
* **Also submit** Logical data model in ERwin format (.erwin) and .ere report at forward engineering.
* DO NOT compress the files; instead just submit them as separate files at the same time.
* Failure to adhere to any of the above instructions will result in deduction from the project’s total score.