Order Entry Database – Relational Data Model

The problems use the **Customer**, **OrderTbl**, **OrderDetail**, **Employee** and **Product** tables of the simplified **Order Entry** database. The structure and data for these tables can be found in `AccDB2_OrdEntry.accdb` and in `AccDB2_OrdEntry_Data.xlsx` files, under `Assign` subfolder of `02_Relate_DM` folder.

Start by using **Workbench** to create a new database named `order_entry`. Then switch to **Notepad++**, open the blank SQL script `MySQL2_OrdEntry.sql` that will contain all the code necessary to create all the tables in this assignment, including all the required relationships, keys and constraints. Then copy/paste each of the **CREATE TABLE** statements from Notepad++ into Workbench one by one, execute those scripts carefully, and verify appropriate tables are created. As always, for any errors you encounter, you are expected to try to resolve them yourself first. If, after repeated attempts, the script does not work, dial into online office hours dedicated to providing help with the assignment, and if we run out of time, please email me directly with any questions.

Each of the questions is worth **0.25** points, except **0.75** points for import question #10.

1. Write a CREATE TABLE statement for the `Customer` table. Choose data types appropriate for MySQL DBMS. Note that the `CustBal` column contains numeric data, and make sure you have enough digits to accommodate larger balances. The currency symbols are not stored in the database. The `CustFirstName` and `CustLastName` columns are required (not null).
2. Write a CREATE TABLE statement for the `Employee` table. Choose data types appropriate for MySQL DBMS. The `EmpFirstName`, `EmpLastName`, and `EmpEMail` columns are required (not null).
3. Write a CREATE TABLE statement for the `OrderTbl` table. Choose data types appropriate for MySQL DBMS. The `OrdDate` column is required (not null).
4. Extend your CREATE TABLE statement from problem #3 with referential integrity constraints that will establish the required relationships. Make updates and deletes on related rows restricted. Hint: Use RESTRICT keyword instead of CASCADE.
5. From examination of the sample data and your common understanding of order entry businesses, are null values allowed for the foreign keys in the **OrderTbl** table? Extend the CREATE TABLE statement in problem #4 to enforce the null value restrictions if any.
6. Extend your CREATE TABLE statement for the **Employee** table in problem #2 with a unique constraint for `EmpEMail`, giving it an appropriate name.
7. Write CREATE TABLE statement for `Product` table, with the appropriate primary and foreign keys.
8. Write CREATE TABLE statement for `OrderLine` table, with the appropriate primary and foreign keys. Make sure to CASCADE updates and deletes appropriately.
9. Extend the relationships by adding a self-referencing foreign key to the Employee table. The foreign key SupEmpNo is the employee number of the supervising employee. Thus, the SupEmpNo references Employee table, i.e., it is a self-referencing relationship. Remember to add this constraint to the SQL script after importing all the data.
10. The final part of the assignment involves using `AccDB2_OrdEntry.accdb` Access database or `Acc2DB_OrdEntry.xlsx` Excel file as the provided data sources for `order_entry` MySQL database. You should use both the Workbench's **Table Data Import Wizard** and appropriate import SQL scripts when loading the data into tables.

Submission: You must submit `MySQL2_OrdEntry.sql` SQL script on Canvas by the designated due date.