

## Assignment 4: Database Application Development

The Northwind database created by Microsoft contains the sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world. Here is the schema of the database:

```
Products (ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice,
UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued);
Suppliers (SupplierID, CompanyName, ContactName, ContactTitle, Address, City, Region,
PostalCode, Country, Phone, Fax, HomePage);
Categories (CategoryID, CategoryName, Description, Picture);
Orders (OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia,
Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry);
Order_details (ID, OrderID, ProductID, UnitPrice, Quantity, Discount);
Customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City,
Region, PostalCode, Country, Phone, Fax);
Employees (EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, BirthDate, HireDate,
Address, City, Region, PostalCode, Country, HomePhone, Extension, Photo, Notes, ReportsTo);
Shippers (ShipperID, CompanyName, Phone);
```

Run SQL script northwind.sql to create the above schema and load data in MySQL.

This assignment lets you exercise on database application development using a programming language of your choice. Write an interactive text-based program. It repeatedly gives the user a menu of options to choose from, then carry out the user's request, until he/she finally choose the exit option. Your program should take care of all SQL queries with proper error/exception handling. Your program should also have proper transaction support.

Your program should interact **using plain text interface only**. To make compiling and grading easier, do not include graphics or any other fancy interface.

Your top menu (and the tasks you are asked to implement) includes the following:

1. add a customer
2. add an order
3. remove an order
4. ship an order
5. print pending orders (not shipped yet) with customer information
6. restock products
7. exit

You'll receive 10 points for correct implementation of each of the above items 1-6. Keep in mind the following when you write your code:

- User will be promoted to enter the necessary info, one field at a time.
- All IDs except CustomerID are automatically generated. (e.g. the biggest existing number + 1).
- Many attribute values are required (NOT NULL).

- When add an order:
  - Add to both ORDERS and ORDER\_DETAILS.
  - Pay attention to the foreign key constraints on CustomerID, EmployeeID, ShipVia, ProductID, OrderID.
  - Update the Products's UnitsOnOrder.
  - The order should be rejected if a product in the order is discontinued.
- When remove an order:
  - Delete the entry in both the ORDER and the ORDER\_DETAILS tables.
  - Update the UnitsOnOrder attribute.
- When printing pending order list:
  - Print only pending orders (i.e. orders with NULL ShippedDate).
  - Print them in the order of order date.
- Your code is expected to provide support of database transactions in proper ways.
- Appropriate error-checking and error-handling is expected.
- A user might enter a new record whose key already exists in the table. Handle this appropriately.
- Always assume the way it works in real world, if the above rules are not sufficient or not clear.
- If you have implemented extra features, document the features clearly in the README file.

Refer to course slides for examples on how to connect to database. For MySQL database connectors, refer to the official MySQL online documentation.

Put your source code and a README file (describing how to compile your code, extra features you have implemented, and other stuff you want us to know) into a zip file and submit to e-learning.