

Assignment 4: Database Application Development (60 points)

The Northwind database contains the sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world. It was prepared by Microsoft for technical training. The most recent version of the database can be found in Microsoft Access by entering “Northwind” in the Search for Online Templates box. The database contains the following list of tables (some tables are further indented to make clear their relationships):

- *Employees*
 - *EmployeePrivileges*
 - *Privileges*
- *Customers*
- *Shippers*
- *Products*
- *Orders*
 - *OrderStatus*
 - *OrderTaxStatus*
 - *OrderDetails*
 - *OrderDetailsStatus*
 - *Invoices*
- *Suppliers*
- *PurchaseOrders*
 - *PurchaseOrderStatus*
 - *PurchaseOrderDetails*
- *InventoryTransactions*
 - *InventoryTransactionTypes*
- *Messages*

These tables and their attributes are further described in the ER diagram, which is given in the PDF file NorthWindER.pdf.

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

This assignment lets you exercise on database application development using a programming language of your choice. You are asked to write an interactive text-based program to support basic order fulfillment operations. It repeatedly gives the user a menu of options to choose from, then carries out the user’s request, until the user finally chooses 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. more options
7. exit

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 are automatically generated. (e.g. the biggest existing number + 1). MySQL has a function `LAST_INSERT_ID()` to obtain the auto-increment value immediately after an insert.
- To add an order:
 - Populate proper information (list price, order date, ship address) to `ORDERS` and `ORDER_DETAILS`.
 - Pay attention to foreign key constraints on Customer, Employee, Shipper, Product, OrderID, etc.
 - Multiple products can be placed in an order.
 - The order should be rejected if a product in the order is discontinued.
- To remove an order:
 - Delete the entries in `ORDERS` and `ORDER_DETAILS` tables.
- To ship an order:
 - Check whether there are enough units in stock of every product in the order. To find units in stock of a product, use the `InventoryTransactions` table, find total quantity purchased and subtract quantities sold and on hold. If there are not enough units of any product in the order, the order cannot be shipped.
 - Fill in `ShippedDate`, Shipper ID and Shipping Fee.
 - For each product in the order, insert inventory transaction (`TransactionType="Sold"`) into the `InventoryTransactions` table.
- To print 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 are expected.
- A user might enter a 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.

The main menu has an item: 6. more options. This is where you are encouraged to implement extra features in a separate menu. These extra features may include purchase orders (activate a purchase order when stock is low, requisition, approval), inventory update, invoice, etc. For further explanation of the NorthWind database, please read the paper <http://jise.org/Volume26/n2/JISEv26n2p85.html>.

If you have implemented extra features, document the features clearly in the README file. You can receive bonus points up to the full mark of this assignment.

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

Put the following into a zip file and submit to e-learning:

- README.txt file (describing how to compile your code, extra features you have implemented, and other stuff you want us to know),
- your source code,
- a typescript that logs step-by-step how you compile and run your program.

Grading Rubric:

README.txt and typescript (5 pts)

Compile error-free (5 pts)

Connect to database (10 pts)

Support transactions (10 pts)

1. add a customer (6 pts)
2. add an order (6 pts)
3. remove an order (6 pts)
4. ship an order (6 pts)
5. print pending orders with customer information (6 pts)