

## SQL Practice - Visual DB Tool

Practice SQL DDL and DML using Microsoft's northwinds database with Oracle's MySQL Workbench Visual DB Design Tool

- Download MySQL Community Server:
    - Try: sudo apt-get install mysql
    - Or download from here: <https://dev.mysql.com/downloads/mysql/>
  - Download and Install MySQL Workbench:
    - Try: sudo apt-get install mysql-workbench
    - Or download from here: <https://dev.mysql.com/downloads/workbench/>
  - Open MySQL Workbench and choose local MySQL instance host
  - Type “create database northwinds;” in the Query editor and run it by clicking the lightning bolt icon or ctrl+enter while the cursor is on the query.
  - Type “use northwinds;” in the Query editor and run it.
  - Run each of the five scripts from the *tables.zip* folder to create the tables in the northwinds database, then refresh your schemas and the tables will appear.
    - If creating these tables for the first time, remove the “drop table” command.
  - You can now execute queries against your database. Write (and save) the SQL commands for the following problems in a SQL script:
1. List all the information in the Customer table. **(87 rows)**
  2. List the OrderID and OrderDate for all orders, ordered by most recent OrderDate. **(830 rows)**
  3. List the CustomerID of the customers who have more than 20 orders. **(3 rows)**
  4. How many Northwinds customers are NOT from France? **(1 row)**
  5. List the unique cities from the Employee table and the count of employees living in each city. **(5 rows)**

6. What is the average price (rounded to two decimal places) of all the products sold by Northwinds? **(1 row)**
7. List the Company Names of customers who purchased orders from the employee with *EmployeeID = 1* (use aliases for the different tables you query from). **(118 rows)**
8. List the ProductID, ProductName and InventoryValue for all products whose inventory value is greater than \$2000. (Inventory Value = UnitPrice\*UnitsInStock). **(13 rows)**
9. Create a “Top Customers” List. List the Customer Company Name, Customer Country, and the Value (rounded to two decimal places) of all their orders sorted highest to lowest value for all customers with a total order value greater than \$30,000. **(7 rows)**

(HINT: the value of an order is UnitPrice times Quantity less the discount.)  
(Another HINT: you can convert the output format of a number by using the CAST command. For example: CAST (unitprice\*quantity)AS DECIMAL(9,2)) will format the result of the multiplication into a decimal number with 9 total digits and 2 digits to the right of the decimal.)
10. List CustomerID, CompanyName, and OrderIDs for *all* customers. (Hint: use Left Join with nwOrders OR Right Join with nwCustomers). Order by CustomerID. **(787 rows)**
  - a. Which Customers have no orders? (Modify your query to only display the rows where OrderID is NULL.) **(2 Rows)**
11. List the ProductName and QuantityPerUnit of all products that come in boxes. (Hint: Use LIKE with wildcard). **(11 rows)**
12. Create a VIEW called “EmployeeOrders” listing the employees lastname, firstname, and the total count of the orders that employee has placed.
13. Run a query against the Employee Orders View listing the “top three” employees and their order counts in order from largest to smallest number of orders. **(3 rows)**
  - a. Who is the top selling employee?

14. Create a NEW table named “topItems” with the following columns: ItemID (integer), ItemCode (integer), ItemName (varchar(40)), InventoryDate (DATE), SupplierID (integer), ItemQuantity (integer)and ItemPrice (decimal (9,2)) . None of these columns can be NULL. Default the ItemQuantity to 0 and the ItemPrice to 0.00. Include a PRIMARY KEY constraint on ItemID.
15. Populate the new table “topItems” using the following columns from the nwProducts table (renamed) for products whose inventory value is greater than \$2500. (Use a subquery). **(9 rows inserted)**
- a. ProductID → ItemID
  - b. CategoryID → ItemCode
  - c. ProductName → ItemName
  - d. Today's date→ Inventory Date
  - e. UnitsInStock → ItemQuantity
  - f. UnitPrice → ItemPrice
  - g. SupplierID → SupplierID
16. Delete the rows in topItems for products that have been discontinued. (Hint: use a subquery). **(1 row deleted)**
- \*Note: if Workbench throws this error: “You are using safe update mode...”, Run this command: SET SQL\_SAFE\_UPDATES=0;
17. Add a new column to the topItems table called InventoryValue (decimal (9,2)) after the inventory date.
18. Update the topItems table, setting the InventoryValue column equal to ItemPrice times ItemQuantity.
19. List all columns in the topItems table. **(7 rows)**
20. Drop the topItems table.