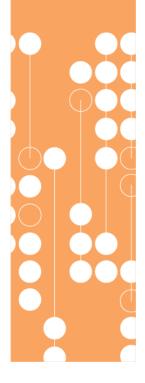
# **SQL Revision**

**Exercise Handout** 







# **Contents**

_	verview	2
	Objectives	2
	Setup: Launch SQL Server Management Studio (if necessary)	2
	Step 1	2
	Step 2	2
	Step 3	2
	Step 4	2
	Step 5	2
	Step 6	3
	Step 7	3
	Step 8	3
	Step 9	3
	Step 10	3
	Step 11	3
	Step 12	3
	Step 13	3
	Step 14	3
	Step 15	4
	Step 16	4
	Step 17	4
	Step 18	4



# **Overview**

In this set of exercises, you will work through tasks to revise what you have learned during the module. All exercises use the Northwind database.

## **Objectives**

At the end of this lab, you will have strengthened your knowledge obtained in the module and will have clarified any uncertain points.

## Setup: Launch SQL Server Management Studio (if necessary)

- 1. Launch the virtual machine.
- 2. Launch SQL Server Management Studio.
- 3. Connect to the server.
- 4. Select the Northwind database.

## Step 1

List the OrderID, CustomerID and OrderDate of all Orders that were placed on or before 31 December 1996 sorted in ascending order of OrderDate within CustomerID.

## Step 2

List the OrderID, CustomerID and OrderDate of all Orders that were placed between 16/06/1997 and 20/06/1997 sorted in ascending order OrderDate.

## Step 3

List all the data for all Products where the ReorderLevel is greater than 25 sorted in descending order of UnitsInStock.

## Step 4

List the CompanyName, ContactName, Country, Region and Phone of Customers whose Region is specified sorted in ascending order of Region within Country.

## Step 5

List the distinct Company Names of Customers who place their orders with employees whose last names are either "Davolio" or "Fuller".



#### Step 6

List the ProductName and UnitPrice of all Products with a CategoryID of 2 or 3 together with a NewUnitPrice (rounded to 2 decimal places) calculated by adding 3% to the UnitPrice, sorted in ascending order of ProductName.

#### Step 7

Show the total value of all the products currently held in stock with no decimal places.

#### Step 8

Show the total number of Customers.

## Step 9

List the CustomerID, CompanyName and the number of orders for each customer that has more than 20 orders grouping by CustomerID and CompanyName..

## Step 10

List the CustomerID, CompanyName and the number of orders for the Customer that has the highest number of orders using appropriate grouping.

## Step 11

List the CustomerID and CompanyName for all customers who have had no dealings with employees whose last names are either "Davolio" or "Fuller".

## Step 12

Add the following new customer:

ALAVO, Al's Avocados, Alan Tracy, Owner, 34 Porridge Pot Alley, Chelmsford, Essex, CM3 2BZ, UK, 01926 401697, 01926 401698.

# Step 13

Update all the unit prices for Products with an increase of 5%.

## Step 14

List all the data for Products sorted in ascending order of ProductID.



#### Step 15

- a) Add the following Region:
  - 5, Europe
- b) Add the following Territories:
  - 09000, UK, 5 09001, France, 5
- c) List all the territories in ascending order of TerritoryID.
- d) List all the regions in ascending order of RegionID.
- e) Imagine that some time has passed and the region with an ID of 5 and all its associated territories needs to be removed from the database. Delete the appropriate records.
- f) List all the territories in ascending order of TerritoryID.
- g) List all the regions in ascending order of RegionID.:

### Step 16

Produce a list of Products showing Percentage Raises, ProductID and old and new UnitPrices. Products with a CategoryID of 1, 3 or 5 are given a 5% rise, Products with a CategoryID of 2, 4 or 6 are given a 10% rise and other Products should not be given a rise. Display the results in ascending PercentageRaise sequence and display the new UnitPrices with 2 decimal places.

## Step 17

Create a new view for all employees who are managers only using all the fields from the employee table. Include the EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, HomePhone, Extension and ReportsTo columns. Apply a CHECK constraint.

## Step 18

Using the view for managers show all the fields for the manager(s) who don't report to anyone sorted in ascending order of EmployeeID.

