SQL Practise Exercises   Daniel Alldritt

Exercise 1 – Northwind Queries

1.1

Question: Write a query that lists all Customers in either Paris or London. Include Customer ID, Company Name and all address fields.

Answer:

```
SELECT  c.CustomerID
      , c.CompanyName
      , c.Address
      , c.City
      , c.PostalCode
FROM    Customers c
WHERE   c.City IN ('Paris', 'London')
```

1.2

Question: List all products stored in bottles.

Answer:

```
SELECT  p.ProductID
      , p.ProductName
      , p.QuantityPerUnit
FROM    Products p
WHERE   p.QuantityPerUnit LIKE '%bottles%'
```

1.3

Question: Repeat question above but add in the Supplier Name and Country.

Answer:

```
SELECT      p.ProductID
          , p.ProductName
          , p.QuantityPerUnit
          , s.SupplierID
          , s.CompanyName AS "Supplier Name"
          , s.Country
FROM        Products p
INNER JOIN  Suppliers s
ON          p.SupplierID = s.SupplierID
WHERE       p.QuantityPerUnit LIKE '%bottles%'
```

1.4

Question: Write an SQL Statement that shows how many products there are in each category. Include Category Name in result set and list the higher number first.

Answer:

```
SELECT      p.CategoryID
          , c.CategoryName
          , COUNT(p.ProductID) AS "Number of Products"
FROM        Products p
INNER JOIN  Categories c
ON          c.CategoryID = p.CategoryID
GROUP BY    p.CategoryID, c.CategoryName
ORDER BY    "Number of Products" DESC
```

1.5

Question: List all employees using concatenation to join their title of courtesy, first name and last name together. Also include their city of residence.

Answer:

```
SELECT  CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "Full Name"
      , e.City
FROM    Employees e
WHERE   e.Country IN ('UK')
```

1.6

Question: List Sales Total for all Sales Regions (via the Territories table using 4 joins) with a Sales Total greater than 1,000,000. Use rounding or FORMAT to present the numbers.

Answer:

```
SELECT      r.RegionID
          , r.RegionDescription
          , ROUND(SUM(od.UnitPrice * od.Quantity * (1.0-
od.Discount)), 2) AS "Sales Total"
FROM        [Order Details] od
INNER JOIN  Orders o
ON          od.OrderID = o.OrderID
INNER JOIN  EmployeeTerritories et
ON          o.EmployeeID = et.EmployeeID
INNER JOIN  Territories t
ON          et.TerritoryID = t.TerritoryID
INNER JOIN  Region r
ON          t.RegionID = r.RegionID
GROUP BY    r.RegionID, r.RegionDescription
HAVING      SUM(od.UnitPrice * od.Quantity * (1.0-od.Discount)) > 1000000
ORDER BY    1
```

1.7

Question: Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.

Answer:

```
2  SELECT      COUNT(*) AS "Number of Freights over 100.00 going to UK or USA"
3  FROM        Orders o
4  WHERE       o.Freight > 100.00
5  AND         o.ShipCountry IN ('USA', 'UK')
```

5.1

Question: Write an SQL Statement to identify the Order Number of the Order with the highest amount(value) of discount applied to that order.

Answer:

```
SELECT          o.OrderID
                , o.UnitPrice * o.Quantity * o.Discount AS "Discount Total"
FROM            [Order Details] o
WHERE           o.UnitPrice * o.Quantity * o.Discount =
                (SELECT
                        MAX(od.UnitPrice * od.Quantity * od.Discount)
                FROM    [Order Details] od)
ORDER BY        "Discount Total" DESC
```

Exercise 2 – Create Spartans Table

2.1

Question: Write the correct SQL statement to create the following table:

Spartans Table – include details about all the Spartans on this course. Separate Title, First Name and Last Name into separate columns, and include University attended, course taken and mark achieved. Add any other columns you feel would be appropriate.

Answer:

```
CREATE TABLE exercise_2_daniel
(
    id int IDENTITY(1,1) PRIMARY KEY,
    title VARCHAR(10),
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    university_attended VARCHAR(30),
    course_taken VARCHAR(20),
    mark_achieved VARCHAR(20),
    main_interest VARCHAR(20),
    favourite_integer int
)
```

2.2

Question: Write SQL statements to add the details of the Spartans in your course to the table you have created.

Answer:

```
INSERT INTO exercise_2_daniel
VALUES
('Mr', 'Daniel', 'Alldritt', 'University of Unknown', 'Amazing Course 1', 'First', 'Coding', 11),
('Mr', 'Ahmed', 'Rahman', 'University of Unknown', 'Amazing Course 2', 'First', 'Coding', 12),
('Mr', 'Alex', 'Ng', 'University of Unknown', 'Amazing Course 3', 'First', 'Coding', 13),
```

```
('Mr', 'Andrei', 'Pavel', 'University of Unknown', 'Amazing Course 4', 'First', 'Coding', 1
4),
('Mr', 'Asakar', 'Hussain', 'University of Unknown', 'Amazing Course 5', 'First', 'Coding',
 15),
('Mr', 'Ben', 'Middlehurst', 'University of Unknown', 'Amazing Course 6', 'First', 'Coding'
, 16),
('Mr', 'Benjamin', 'Balls', 'University of Unknown', 'Amazing Course 7', 'First', 'Coding',
 17),
('Mr', 'Gregory', 'Spratt', 'University of Unknown', 'Amazing Course 8', 'First', 'Coding',
 18),
('Mr', 'Ismail', 'Kadir', 'University of Unknown', 'Amazing Course 9', 'First', 'Coding', 1
9),
('Mr', 'James', 'Fletcher', 'University of Unknown', 'Amazing Course 10', 'First', 'Coding'
, 20),
('Mr', 'Jamie', 'Hammond', 'University of Unknown', 'Amazing Course 11', 'First', 'Coding',
 21),
('Mr', 'Josh', 'Weeden', 'University of Unknown', 'Amazing Course 12', 'First', 'Coding', 2
2),
('Mr', 'Nathan', 'Johnston', 'University of Unknown', 'Amazing Course 13', 'First', 'Coding
', 23),
('Mr', 'Rashawn', 'Henry', 'University of Unknown', 'Amazing Course 14', 'First', 'Coding',
 24),
('Mr', 'Sidhant', 'Khosla', 'University of Unknown', 'Amazing Course 15', 'First', 'Coding'
, 25),
('Mr', 'Timin', 'Rickaby', 'University of Unknown', 'Amazing Course 16', 'First', 'Coding',
 26),
('Mr', 'Yusuf', 'Uddin', 'University of Unknown', 'Amazing Course 17', 'First', 'Coding', 2
7)
```

Exercise 3 – Northwind Data Analysis linked to Excel

3.1

Question: List all Employees from the Employees table and who they report to. No Excel required.

Answer:

```
SELECT          e.EmployeeID
              , CONCAT(e.FirstName, ' ', e.LastName) AS "Employee Name"
              , e.ReportsTo
              , CONCAT(e2.FirstName, ' ', e2.LastName) AS "Line Manager"

FROM            Employees e
LEFT JOIN       Employees e2
ON              e.ReportsTo = e2.EmployeeID
```

3.2

Question: List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table and present as a bar chart.
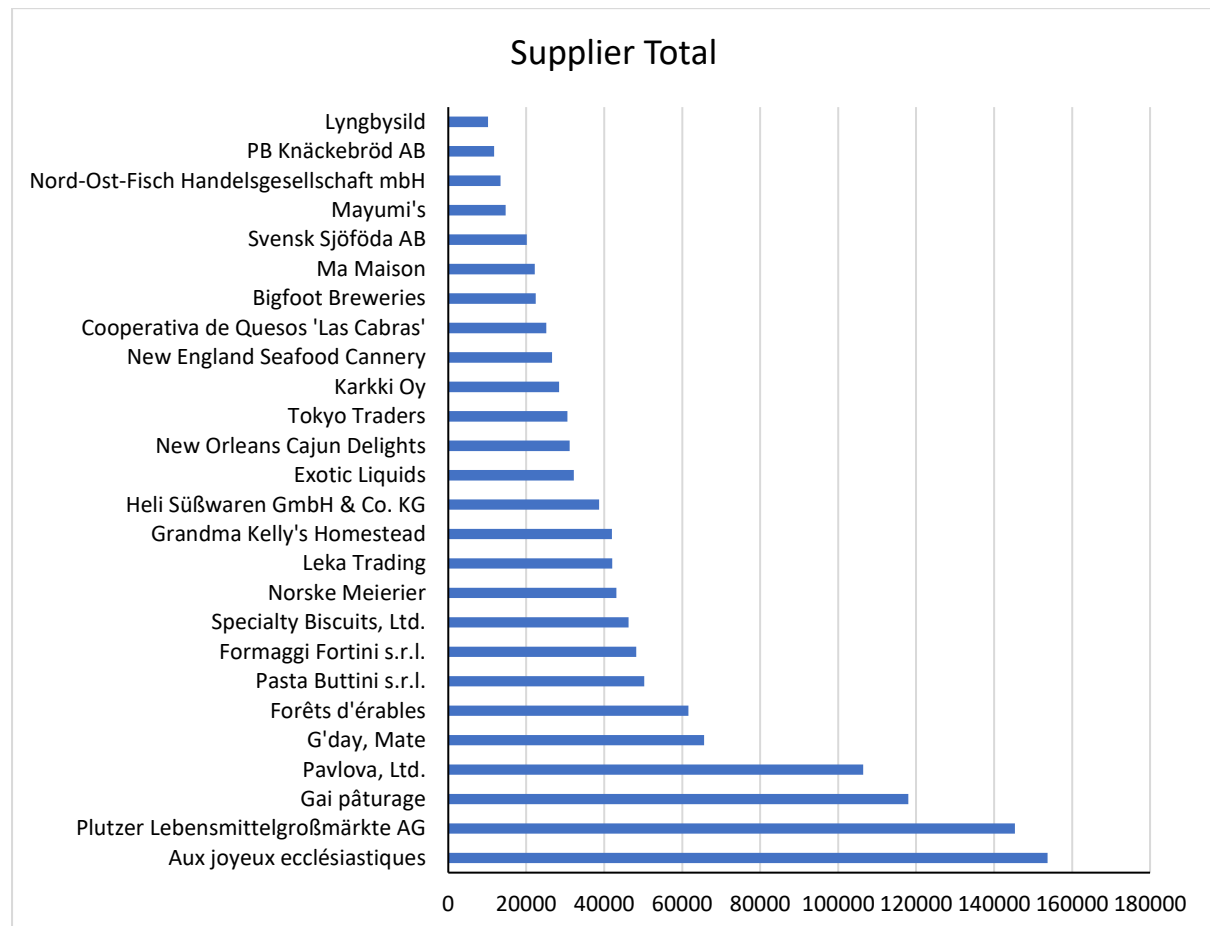
Answer:

```
SELECT      s.SupplierID
          , s.CompanyName
          , ROUND(SUM(od.UnitPrice * od.Quantity * (1.0-
od.Discount)), 2) AS "Total Sales"
```

```
FROM        Suppliers s
INNER JOIN  Products p
ON          s.SupplierID = p.SupplierID
INNER JOIN  [Order Details] od
ON          p.ProductID = od.ProductID
GROUP BY    s.SupplierID, s.CompanyName
HAVING      SUM(od.UnitPrice * od.Quantity * (1.0-od.Discount)) > 10000
ORDER BY    3 DESC
```


Supplier Total

3.3

Question: 3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped. No Excel required.

Answer:

```
SELECT TOP 10   c.CustomerID
              , c.CompanyName
              , ROUND(SUM(od.UnitPrice * od.Quantity * (1-
od.Discount)), 2) AS "Total Value with Discount Added"
FROM        Customers c
INNER JOIN  Orders o
ON          c.CustomerID = o.CustomerID
INNER JOIN  [Order Details] od
ON          o.OrderID = od.OrderID
WHERE       YEAR(o.OrderDate) = (SELECT TOP 1 YEAR(o2.OrderDate) AS "Latest Year"
            FROM Orders o2
```

```
                ORDER BY o2.OrderDate DESC)
GROUP BY        c.CustomerID, c.CompanyName
ORDER BY        "Total Value with Discount Added" DESC
```

3.4

Question: Plot the Average Ship Time by Month for all data in the Orders Table using a line chart as below.

Answer:

```
SELECT          YEAR(o.OrderDate) AS "Year"
                , MONTH(o.OrderDate) AS "Month"
                , FORMAT(o.OrderDate, 'MMM-yy') AS "Year-Month"
                , AVG(CAST(DATEDIFF(d, o.OrderDate, o.ShippedDate) AS Decimal(4,2))) AS "Av
erage Number of Ship Days"
FROM            Orders o
GROUP BY        YEAR(o.OrderDate), MONTH(o.OrderDate), FORMAT(o.OrderDate, 'MMM-yy')
ORDER BY        1, 2
```


Average Ship Time by Months