## -- Notes for Trainers

## /\*

Be wary of copying SQL from Microsoft Word to SSMS, as the quote character is not copied correctly.

A few of these answers are flexible, they do not have to match EXACTLY, for example where the question has not specified which columns to include. The best approach is to ask them to imagine they are on site and the only info they have from the end user is this text (the question) and the user is on holiday for a week. How would they ensure they went the extra mile and deliver the best possible solution for the user? This may also include using column aliases for every column in the SELECT to ensure it looks as nice as possible, capitalisation and spaces etc.

Also note that FORMAT for numbers are not currently covered in the slides.

\*/

## --Question 1.1

SELECT CustomerID AS “Customer ID”, CompanyName AS “Customer Name”, Address + ‘, ‘ + City + ‘, ‘ + PostalCode + ‘, ‘ + Country AS “Address”

FROM Customers

WHERE City IN ('Paris', 'London'); /\* Best Answer \*/

--OR

WHERE City = 'Paris' OR City = 'London';  /\* Not as good but still correct \*/

--Should result in 8 Rows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerID | CompanyName | Address | City | PostalCode | Country |
| AROUT | Around the Horn | 120 Hanover Sq. | London | WA1 1DP | UK |
| BSBEV | B's Beverages | Fauntleroy Circus | London | EC2 5NT | UK |
| CONSH | Consolidated Holdings | Berkeley Gardens 12 Brewery | London | WX1 6LT | UK |
| EASTC | Eastern Connection | 35 King George | London | WX3 6FW | UK |
| NORTS | North/South | South House 300 Queensbridge | London | SW7 1RZ | UK |
| PARIS | Paris spécialités | 265, boulevard Charonne | Paris | 75012 | France |
| SEVES | Seven Seas Imports | 90 Wadhurst Rd. | London | OX15 4NB | UK |
| SPECD | Spécialités du monde | 25, rue Lauriston | Paris | 75016 | France |

## --Question 1.2

**--Question 1.2**

**SELECT ProductName, QuantityPerUnit**

**FROM Products**

**WHERE QuantityPerUnit LIKE '%Bottle%';**

--Should result in 12 Rows:

|  |  |
| --- | --- |
| ProductName | QuantityPerUnit |
| Chang | 24 - 12 oz bottles |
| Aniseed Syrup | 12 - 550 ml bottles |
| Genen Shouyu | 24 - 250 ml bottles |
| Sasquatch Ale | 24 - 12 oz bottles |
| Steeleye Stout | 24 - 12 oz bottles |
| Côte de Blaye | 12 - 75 cl bottles |
| Chartreuse verte | 750 cc per bottle |
| Sirop d'érable | 24 - 500 ml bottles |
| Louisiana Fiery Hot Pepper Sauce | 32 - 8 oz bottles |
| Laughing Lumberjack Lager | 24 - 12 oz bottles |
| Outback Lager | 24 - 355 ml bottles |
| Rhönbräu Klosterbier | 24 - 0.5 l bottles |

## --Question 1.3

SELECT ProductName, QuantityPerUnit, CompanyName, Country

FROM Products p

INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID

WHERE QuantityPerUnit LIKE '%Bottle%';

--Should result in 12 Rows:

|  |  |  |  |
| --- | --- | --- | --- |
| ProductName | QuantityPerUnit | CompanyName | Country |
| Chang | 24 - 12 oz bottles | Exotic Liquids | UK |
| Aniseed Syrup | 12 - 550 ml bottles | Exotic Liquids | UK |
| Genen Shouyu | 24 - 250 ml bottles | Mayumi's | Japan |
| Sasquatch Ale | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| Steeleye Stout | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| Côte de Blaye | 12 - 75 cl bottles | Aux joyeux ecclésiastiques | France |
| Chartreuse verte | 750 cc per bottle | Aux joyeux ecclésiastiques | France |
| Sirop d'érable | 24 - 500 ml bottles | Forêts d'érables | Canada |
| Louisiana Fiery Hot Pepper Sauce | 32 - 8 oz bottles | New Orleans Cajun Delights | USA |
| Laughing Lumberjack Lager | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| Outback Lager | 24 - 355 ml bottles | Pavlova, Ltd. | Australia |
| Rhönbräu Klosterbier | 24 - 0.5 l bottles | Plutzer Lebensmittelgroßmärkte AG | Germany |

## --Question 1.4

**SELECT c.CategoryName "Category Name", COUNT(\*) as "No of Products"**

**FROM Products p**

**INNER JOIN Categories c ON p.CategoryID=c.CategoryID**

**GROUP BY c.CategoryName**

**ORDER BY COUNT(\*) DESC;**

--Should result in 8 Rows:

|  |  |
| --- | --- |
| Category Name | **No of Products** |
| **Confections** | **13** |
| **Beverages** | **12** |
| **Condiments** | **12** |
| **Seafood** | **12** |
| **Dairy Products** | **10** |
| **Grains/Cereals** | **7** |
| **Meat/Poultry** | **6** |
| **Produce** | **5** |

## --Question 1.5

SELECT TitleOfCourtesy + ' ' + FirstName + ' ' + LastName As Employee, City

  FROM Employees

  WHERE Country = 'UK';

--Should result in 4 Rows:

|  |  |
| --- | --- |
| Employee | City |
| Mr. Steven Buchanan | London |
| Mr. Michael Suyama | London |
| Mr. Robert King | London |
| Ms. Anne Dodsworth | London |

## --Question 1.6

--IMPROVED ANSWER USING ONLY 5 TABLES

-- EmployeeID is on both Employees table and Orders table, so don’t need Employees.

SELECT r.RegionID, r.RegionDescription AS Region,

FORMAT(SUM((UnitPrice \* Quantity) \* (1-Discount)),'C')

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

|  |  |  |
| --- | --- | --- |
| RegionID | Region | Sales Total by Region |
| 1 | Eastern | $2,730,198.01 |
| 2 | Western | $1,615,248.00 |
| 3 | Northern | $1,048,605.58 |

--ORIGINAL ANSWER USING 6 TABLES (ALSO CORRECT)

SELECT t.RegionID AS "Region ID", RegionDescription As "Region",

FORMAT(Sum(UnitPrice \* Quantity \* (1-Discount)),'C') AS "Sales Total"

FROM Employees e

INNER JOIN EmployeeTerritories et ON e.EmployeeID=et.EmployeeID

INNER JOIN Territories t ON et.TerritoryID=t.TerritoryID

INNER JOIN Region r ON r.RegionID=t.RegionID

INNER JOIN Orders o ON o.EmployeeID=e.EmployeeID

INNER JOIN [Order Details] od ON od.OrderID=o.OrderID

GROUP BY t.RegionID, RegionDescription

HAVING Sum(UnitPrice \* Quantity \* (1-Discount)) > 1000000

ORDER BY Sum(UnitPrice \* Quantity \* (1-Discount)) DESC;

-- Note: Do not put FORMAT in ORDER BY clause.

|  |  |  |
| --- | --- | --- |
| Region ID | Region | Sales Total |
| 1 | Eastern | $2,730,198.01 |
| 2 | Western | $1,615,248.00 |
| 3 | Northern | $1,048,605.58 |

## --Question 1.7

SELECT COUNT(\*) AS 'No of Orders >100 from US or UK'

FROM Orders

WHERE Freight>100 AND ShipCountry IN ('USA','UK');

|  |
| --- |
| No of Orders >100 from US or UK |
| 49 |

## --Question 1.8

### --SIMPLIFIED ANSWER:

SELECT OrderID AS 'Order ID',

FORMAT((UnitPrice \* Quantity) \* Discount,'C') AS 'Discount Amount'

FROM [Order Details]

ORDER BY [Discount Amount] DESC;

-- This SQL results in 2155 rows returned. Due to the ORDER BY it is still possible to get the right answer (Discount Amount 2108).

### --TOP ANSWER:

SELECT Orderid AS "Order ID",

FORMAT((UnitPrice \* Quantity) \* Discount,'C') AS "Total Discount"

FROM [Order Details]

WHERE ((UnitPrice \* Quantity) \* Discount) =

(

SELECT MAX(

(UnitPrice \* Quantity) \* Discount) -- AS 'Discount Amount'

FROM [Order Details]

)

;

|  |  |
| --- | --- |
| Order ID | Total Discount |
| 10353 | $2,108.00 |
| 10372 | $2,108.00 |

-- NOTE: When comparing the Mini Project with the Advanced SQL Project, the first 8 questions are identical – up to this point. This new Mini Project focusses solely on the SELECT statement and running queries apart from one simple CREATE TABLE). If you have a mix of abilities in the class, the stronger students can be invited to take on the main SQL Project. Or even both projects.

## Question 2.1

DROP TABLE Spartans;

**CREATE TABLE Spartans**

**(**

**EmployeeID int IDENTITY(1,1) NOT NULL PRIMARY KEY,**

**Title varchar(20) NOT NULL,**

**FirstName varchar(50) NOT NULL,**

**LastName varchar(50) NOT NULL,**

**SpartaCourse varchar(50) NOT NULL,**

**University varchar(50),**

**UniCourse varchar(50),**

**UniMark varchar(30),**

**StartDate date**

**);**

-- The above columns can change and flexibility can be allowed for this when marking.

## Question 2.2

**INSERT INTO Spartans**

**(Title, FirstName, LastName, SpartaCourse,University,UniCourse,UniMark,StartDate)**

**VALUES**

**(‘Miss’, ‘Giverny’, ‘Wilson-Martin’,’BA-Test’,’Staffordshire’,’Games Design’,’2:2’,’2014-02-10’);**

**INSERT INTO Spartans**

**(Title, FirstName, LastName, SpartaCourse,University,UniCourse,UniMark,StartDate)**

**VALUES**

**(‘Mr’, ‘Rob’, ‘Whitehouse’,’SDET’,’Sheffield’,’Physics’,’2:1’,’2015-01-01’);**

-- The number of and content of the inserts can change depending on the students so some flexibility can be allowed for this when marking but attention must be paid to the data type and order i.e. whether the right values have been inserted into the right columns.

## Question 3.1

-- Uses a SELF JOIN

SELECT e.FirstName + ' ' + e.LastName AS "Employee Name",

b.FirstName + ' ' + b.LastName AS "Reports To"

FROM Employees e

LEFT JOIN Employees b ON e.ReportsTo=b.EmployeeID

ORDER BY "Reports To","Employee Name";

|  |  |
| --- | --- |
| Employee Name | Reports To |
| Janet Leverling | Andrew Fuller |
| Laura Callahan | Andrew Fuller |
| Margaret Peacock | Andrew Fuller |
| Nancy Davolio | Andrew Fuller |
| Steven Buchanan | Andrew Fuller |
| Anne Dodsworth | Steven Buchanan |
| Michael Suyama | Steven Buchanan |
| Robert King | Steven Buchanan |

## Question 3.2

SELECT s.CompanyName,SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)) As "Supplier Total"

FROM [Order Details] od

INNER JOIN Products p ON od.ProductID=p.ProductID

INNER JOIN Suppliers s ON p.SupplierID=s.SupplierID

GROUP BY s.CompanyName

HAVING SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount))>10000

ORDER BY SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)) DESC;

|  |  |
| --- | --- |
| **CompanyName** | **Supplier Total** |
| **Aux joyeux ecclésiastiques** | **153691.275178909** |
| **Plutzer Lebensmittelgroßmärkte AG** | **145372.399160385** |
| **Gai pâturage** | **117981.180160522** |
| **Pavlova, Ltd.** | **106459.775501251** |
| **G'day, Mate** | **65626.7701091766** |
| **Forêts d'érables** | **61587.57006073** |
| **Pasta Buttini s.r.l.** | **50254.6100997925** |
| **Formaggi Fortini s.r.l.** | **48225.1649436951** |
| **Specialty Biscuits, Ltd.** | **46243.9798936844** |
| **Norske Meierier** | **43141.5100288391** |
| **Leka Trading** | **42017.6451034546** |
| **Grandma Kelly's Homestead** | **41953.299987793** |
| **Heli Süßwaren GmbH & Co. KG** | **38653.4194946289** |
| **Exotic Liquids** | **32188.0601043701** |
| **New Orleans Cajun Delights** | **31167.9898986816** |
| **Tokyo Traders** | **30526.3400268555** |
| **Karkki Oy** | **28442.7275676727** |
| **New England Seafood Cannery** | **26590.9748821259** |
| **Cooperativa de Quesos 'Las Cabras'** | **25159.4300842285** |
| **Bigfoot Breweries** | **22391.2000427246** |
| **Ma Maison** | **22154.6372413635** |
| **Svensk Sjöföda AB** | **20144.0599822998** |
| **Mayumi's** | **14736.7550230026** |
| **Nord-Ost-Fisch Handelsgesellschaft mbH** | **13424.1974983215** |
| **PB Knäckebröd AB** | **11724.060005188** |
| **Lyngbysild** | **10221.1749639511** |

**-- The absence of FORMAT for the Supplier Total will ensure Excel can process the numbers as pasted – if FORMAT is used then Excel does not treat them as numbers due (potentially) to incompatibilities with the currency symbol.**

## Question 3.3

SELECT TOP 10 c.CustomerID AS "Customer ID", c.CompanyName As "Company",

FORMAT(SUM(UnitPrice \* Quantity \* (1-Discount)),'C')

AS "YTD Sales"

FROM Customers c

INNER JOIN Orders o ON o.CustomerID=c.CustomerID

INNER JOIN [Order Details] od ON od.OrderID=o.OrderID

WHERE YEAR(OrderDate)=(SELECT MAX(YEAR(OrderDate)) From Orders)

--WHERE YEAR(OrderDate)=1998 --WHERE YEAR(OrderDate)='1998'

AND o.ShippedDate IS NOT NULL

GROUP BY c.CustomerID, c.CompanyName

ORDER BY SUM(UnitPrice \* Quantity \* (1-Discount)) DESC;

-- Note three choices of WHERE clause – subquery gets maximum marks

|  |  |  |
| --- | --- | --- |
| **Customer ID** | **Company** | **YTD Sales** |
| **QUICK** | **QUICK-Stop** | **$37,217.32** |
| **SAVEA** | **Save-a-lot Markets** | **$36,310.11** |
| **ERNSH** | **Ernst Handel** | **$31,311.75** |
| **HANAR** | **Hanari Carnes** | **$23,821.20** |
| **HUNGO** | **Hungry Owl All-Night Grocers** | **$20,402.12** |
| **RATTC** | **Rattlesnake Canyon Grocery** | **$19,982.55** |
| **KOENE** | **Königlich Essen** | **$19,582.77** |
| **WHITC** | **White Clover Markets** | **$15,278.90** |
| **FOLKO** | **Folk och fä HB** | **$13,644.07** |
| **SUPRD** | **Suprêmes délices** | **$11,644.60** |

## Question 3.4

SELECT MONTH(OrderDate) Month, YEAR(OrderDate) Year, AVG(CAST(DATEDIFF(d, OrderDate, ShippedDate) As DECIMAL(10,2))) As ShipTime

FROM orders

WHERE ShippedDate IS NOT NULL

GROUP BY YEAR(OrderDate),MONTH(OrderDate)

ORDER BY Year ASC, Month ASC

|  |  |  |
| --- | --- | --- |
| Month | Year | ShipTime |
| 7 | 1996 | 8.045454 |
| 8 | 1996 | 8.000000 |
| 9 | 1996 | 10.608695 |
| 10 | 1996 | 6.500000 |
| 11 | 1996 | 8.360000 |
| 12 | 1996 | 7.516129 |
| 1 | 1997 | 9.969696 |
| 2 | 1997 | 9.310344 |
| 3 | 1997 | 8.300000 |
| 4 | 1997 | 9.000000 |
| 5 | 1997 | 9.156250 |
| 6 | 1997 | 8.833333 |
| 7 | 1997 | 8.696969 |
| 8 | 1997 | 6.787878 |
| 9 | 1997 | 9.135135 |
| 10 | 1997 | 8.710526 |
| 11 | 1997 | 8.735294 |
| 12 | 1997 | 9.854166 |
| 1 | 1998 | 9.054545 |
| 2 | 1998 | 7.222222 |
| 3 | 1998 | 9.397260 |
| 4 | 1998 | 6.650793 |
| 5 | 1998 | 2.500000 |