

```

--* BUSIT 103           Assignment   #11           DUE DATE : Consult course calendar
/*
Name: Christopher Singleton
Class: BUSIT103 - Online
Instructor: Art Lovestedt
Date: 12/06/2014
*/
--You are to develop SQL statements for each task listed.
--You should type your SQL statements under each task.

/*      Submit your .sql file named with your last name, first name and assignment # (e.g.,
GriggsAssignment11.sql).
      Submit your file to the instructor using through the course site.  */

-- It is your responsibility to provide a meaningful column name for the return value of the function
-- and use an appropriate sort order. All joins are to use the ANSI standard syntax.

USE AdventureWorksDW2012;

--1. AdventureWorks wants geographic information about its resellers.
--      Be sure to add a meaningful sort as appropriate and give each derived column an alias.

--1a. (2) First check to determine if there are resellers without geography info.

SELECT *
FROM [dbo].[DimReseller]
WHERE GeographyKey IS NULL

/*
SELECT *
FROM [dbo].[DimReseller]
SELECT *
FROM [dbo].[DimGeography]
*/

--1b. (4) Display a count of resellers in each Country.
--      Show country name and the count of resellers.

SELECT g.EnglishCountryRegionName
      ,COUNT(ResellerKey) AS ResellerKeyCount
FROM [dbo].[DimReseller] AS r
      INNER JOIN [dbo].[DimGeography] AS g ON r.[GeographyKey] = g.[GeographyKey]
GROUP BY EnglishCountryRegionName
ORDER BY g.EnglishCountryRegionName

/* Checking...
SELECT COUNT(ResellerKey) AS ResellerKeyCount
FROM [dbo].[DimReseller]
*/

```

--1c. (3) Display a count of resellers in each City.

-- Show count of resellers, City name, State name, and Country name.

```
SELECT COUNT(r.[ResellerKey]) AS ResellerNameCount
      ,g.[City]
      ,g.[StateProvinceName]
      ,g.[EnglishCountryRegionName]
FROM [dbo].[DimReseller] AS r
      INNER JOIN [dbo].[DimGeography] AS g ON r.[GeographyKey] = g.[GeographyKey]
GROUP BY g.[City]
      ,g.[StateProvinceName]
      ,g.[EnglishCountryRegionName]
ORDER BY ResellerNameCount DESC
```

--2. AdventureWorks wants banking and historical information about its resellers.

-- Be sure to add a meaningful sort as appropriate and give each derived column an alias.

--2a. (2) Check to see if there are any resellers without a value in the bank name field.

```
SELECT *
FROM [dbo].[DimReseller]
WHERE BankName = NULL
```

--2b. (2) List the name of each bank and the number of resellers using that bank.

```
SELECT DISTINCT BankName
      ,COUNT(ResellerKey) AS ResellerCount
FROM [dbo].[DimReseller]
GROUP BY BankName
```

/* Checking Table...

```
SELECT *
FROM [dbo].[DimReseller]
*/
```

--2c. (2) List the year opened and the number of resellers opening in that year.

```
SELECT [YearOpened] AS YearResellerOpened, COUNT([ResellerKey]) AS NumberofResellers
FROM [dbo].[DimReseller]
GROUP BY [YearOpened]
ORDER BY YearOpened DESC
```

--2d. (2) List the order frequency and the number of resellers with that order frequency.

```
SELECT [OrderFrequency], COUNT([ResellerKey]) AS NumberofResellers
FROM [dbo].[DimReseller]
GROUP BY [OrderFrequency]
ORDER BY NumberofResellers
```

/* Checking...

```
SELECT [OrderFrequency], [ResellerKey]
FROM [dbo].[DimReseller]
*/
```

--2e. (2) List the average number of employees in each of the three business types.

```
SELECT AVG([NumberEmployees]) AS NumberOfEmployees, [BusinessType]
FROM [dbo].[DimReseller]
GROUP BY [BusinessType]
ORDER BY NumberOfEmployees DESC
```

```
--2f. (4) List business type, the count of resellers in that type, and average of Annual Revenue
--      in that business type.
--      +2 Bonus Look up the Format function for Transact SQL. Find the function that will show the
--      money values in this format: $64,395.07.  FORMAT(AVG(YearlyIncome), 'c', 'en-us')
```

```
SELECT [BusinessType]
, COUNT([ResellerKey]) AS NumberOfResellers
, FORMAT(CAST(AVG([AnnualRevenue]) AS Decimal(10,2)), 'c', 'en-us') AS AVG_AnnualRevenue
FROM [dbo].[DimReseller]
GROUP BY [BusinessType]
ORDER BY AVG_AnnualRevenue DESC
```

```
SELECT BusinessType, AnnualSales, AnnualRevenue
FROM DimReseller
WHERE
```

```
--3. AdventureWorks wants information about sales to its resellers. Be sure to add a
--     meaningful sort and give each derived column an alias. Remember that Annual Revenue
--     is a measure of the size of the business and is NOT the total of the AdventureWorks
--     products sold to the reseller. Be sure to use SalesAmount when total sales are
--     requested.
```

```
--3a. (3) List the name of any reseller to which AdventureWorks has not sold a product.
--      Hint: Use a join.
```

```
SELECT [ResellerName]
FROM [dbo].[DimReseller] AS r
LEFT JOIN [dbo].[FactResellerSales] AS frs ON r.ResellerKey = frs.ResellerKey
WHERE frs.ProductKey IS NULL
ORDER BY ResellerName
```

```
--3b. (4) List ALL resellers and total of sales amount to each reseller. Show Reseller
--      name, business type, and total sales with the sales showing two decimal places.
--      Be sure to include resellers for which there were no sales. NULL will appear.
```

```
SELECT r.[ResellerName]
, r.[BusinessType]
, FORMAT(CAST(SUM(frs.SalesAmount) AS Decimal(10,2)), 'c', 'en-us') AS TotalSalesAmount
FROM [dbo].[DimReseller] AS r
LEFT JOIN [dbo].[FactResellerSales] AS frs ON r.ResellerKey = frs.ResellerKey
GROUP BY r.[ResellerName], r.[BusinessType]
ORDER BY TotalSalesAmount DESC
```

```
--3c. BONUS +2 AdventureWorks management wants to see 0.00 instead of NULL for resellers without a
--     value in the total of SalesAmount field. A technique to do this was demonstrated in the demo of
--     selected answers posted to the Module 03 discussion board. NOTE: Be wary of performing this
conversion.
--     You know that NULL and 0 are not the same thing and replacing NULL with 0 will impact
calculations.
--     This exercise is to show you how the conversion is done and is not an endorsement of doing it.
--     Don't perform this conversion elsewhere in this assignment.
```

```
SELECT r.[ResellerName]
, r.[BusinessType]
, --, CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) AS TotalSalesAmount
, CAST(SUM(ISNULL(frs.SalesAmount, 0.00)) AS Decimal(10,2)) AS TotalSalesAmount
FROM [dbo].[DimReseller] AS r
LEFT JOIN [dbo].[FactResellerSales] AS frs ON r.ResellerKey = frs.ResellerKey
GROUP BY r.[ResellerName], r.[BusinessType]
ORDER BY TotalSalesAmount DESC
```

```
--Research ideas... Not sure if this is right, but I tried. Could not find the info in module 3.
--SELECT COALESCE(CAST(NULL AS INT), NULL)
--COALESCE(frs.SalesAmount, 0.00))
--SELECT COALESCE( (SELECT SUM(qty) FROM Sales.OrderDetails), 0 )
--SELECT ISNULL( (SELECT SUM(qty) FROM Sales.OrderDetails), 0 )
--SELECT ProductName, UnitPrice*(UnitsInStock+COALESCE(UnitsOnOrder,0)) FROM Products
```

```
--3d. (3) List resellers and total sales to each. Show reseller name, business type, and total sales
--      with the sales showing two decimal places. Limit the results to resellers to which
--      total sales are less than $500 and greater than $500,000.
```

```
SELECT r.[ResellerName]
      ,r.[BusinessType]
      ,CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) AS TotalSalesAmount
FROM [dbo].[DimReseller] AS r
     INNER JOIN [dbo].[FactResellerSales] AS frs ON r.ResellerKey = frs.ResellerKey
GROUP BY r.[ResellerName], r.[BusinessType]
HAVING CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) < 500
      OR CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) > 500000
ORDER BY TotalSalesAmount DESC
```

```
--3e. (4) List resellers and total sales to each for 2008.
--      Show Reseller name, business type, and total sales with the sales showing two decimal places.
--      Limit the results to resellers to which total sales are between $5,000 and $7,500 and between
--      $50,000 and $75,000
```

```
SELECT r.[ResellerName]
      ,r.[BusinessType]
      ,CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) AS TotalSalesAmount
FROM [dbo].[DimReseller] AS r
     INNER JOIN [dbo].[FactResellerSales] AS frs ON r.ResellerKey = frs.ResellerKey
GROUP BY r.[ResellerName], r.[BusinessType]
HAVING (CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) >= 5000 AND CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) <=
7500)
      OR
      (CAST(SUM(frs.SalesAmount) AS Decimal(10,2)) >= 50000 AND CAST(SUM(frs.SalesAmount) AS
Decimal(10,2)) <= 75000)
ORDER BY TotalSalesAmount DESC
```

```
--4. AdventureWorks wants information about the demographics of its customers.
--      Be sure to add a meaningful sort as appropriate and give each derived column an alias.
```

```
--4a. (2) List customer education level (use EnglishEducation) and the number of customers reporting
--      each level of education.
```

```
SELECT [EnglishEducation], COUNT(CustomerKey) AS NumberOfCustomers
FROM [dbo].[DimCustomer]
GROUP BY [EnglishEducation]
ORDER BY NumberOfCustomers DESC
```

```
--4b. (3) List customer education level (use EnglishEducation), the number of customers reporting
--      each level of education, and the average yearly income for each level of education.
--      Show the average income rounded to two (2) decimal places.
```

```
SELECT [EnglishEducation]
      ,COUNT(CustomerKey) AS NumberOfCustomers
      ,CAST(AVG(YearlyIncome) AS Decimal(10,2)) AS AvgYearlyIncome
FROM [dbo].[DimCustomer]
GROUP BY [EnglishEducation]
ORDER BY AvgYearlyIncome DESC
```

--5. (6) List all customers and the most recent date on which they placed an order (2 fields). Show the customer's first name, middle name, and last name in one column with a space between each part of the name. No name should show NULL. Show the date of the most recent order as mm/dd/yyyy. It is your responsibility to make sure you do not miss any customers. If you need to add one more field to the SELECT or the GROUP BY clause, do it. Just don't add more than one.

```
SELECT FirstName + ' ' + ISNULL(MiddleName, '') + ' ' + ISNULL(LastName, '') AS FullName
      , CONVERT(VARCHAR(10), MAX(fis.[OrderDate]), 101) AS MostRecentOrderDate
FROM [dbo].[DimCustomer] AS c
     INNER JOIN [dbo].[FactInternetSales] AS fis ON c.CustomerKey = fis.CustomerKey
GROUP BY FirstName + ' ' + ISNULL(MiddleName, '') + ' ' + ISNULL(LastName, '')
```

--6. (2) In your own words, write a business question that you can answer by querying the data warehouse and using an aggregate function with the having clause. Then write the complete SQL query that will provide the information that you are seeking.
/* My question: What are the number of total sales per type of bicycle throughout all the years with the price of 10000 dollars or more. List Year, Name of Bicycle and Sales Amount. Sort by Year decending.

```
SELECT * FROM [dbo].[FactInternetSales] --[SalesAmount], [OrderDate]
SELECT * FROM [dbo].[DimProduct] --[EnglishProductName]
SELECT * FROM [dbo].[DimProductSubcategory]
SELECT * FROM [dbo].[DimProductCategory] --ProductCategoryKey = 1 */

SELECT YEAR(fis.[OrderDate]) AS YearOfOrder
      , p.[EnglishProductName]
      , CAST(SUM([SalesAmount]) AS Decimal(10,2)) AS TotalSalesAmount
FROM [dbo].[DimProduct] AS p
     INNER JOIN [dbo].[DimProductSubcategory] AS psc ON p.ProductSubcategoryKey = psc.ProductSubcategoryKey
     INNER JOIN [dbo].[DimProductCategory] AS pc ON psc.ProductCategoryKey = pc.ProductCategoryKey
     INNER JOIN [dbo].[FactInternetSales] AS fis ON p.ProductKey = fis.ProductKey
WHERE pc.ProductCategoryKey = 1
GROUP BY YEAR(fis.[OrderDate]), p.[EnglishProductName]
HAVING CAST(SUM([SalesAmount]) AS Decimal(10,2)) >= 10000
ORDER BY TotalSalesAmount

/*
Excellent job on this Chris - great way to end the quarter :-) Quick notes: - 2f; +2 - 3c; +2 -3e is very close!
Need to also specify the year 2008

*/
```