

--* BUSIT 103

DUE DATE: Consult course calendar

/*

Name: Christopher Singleton

*/

--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.,
GriggsDebiAssignment8.sql).

Submit your file to the instructor using through the course site. */

/* Ideas for consideration: Run the statement in stages: Write the SELECT and FROM clauses first
and run the statement. Add the ORDER BY clause. Then add the WHERE clause; if it is a compound
WHERE clause, add piece at a time. Remember that the order in which the joins are processed does make
a difference with OUTER JOINS.

You will not use Cross-Joins, Full Outer Joins, or Unions in the required exercises. All are to be
accomplished with outer joins or a combination of outer and inner joins using ANSI standard Join
syntax. */

-- Do not remove the USE statement

USE AdventureWorksDW2012;

--NOTE: When the task does not specify sort order, it is your responsibility to order the information
so that is easy to interpret.

--1. (4) List the name of ALL products and the name of the product subcategory to which the
product belongs. Sort on product subcategory name and product name.

```
SELECT p.[EnglishProductName], ps.[EnglishProductSubcategoryName]
FROM [dbo].[DimProduct] AS p
INNER JOIN [dbo].[DimProductSubcategory] AS ps ON p.[ProductSubcategoryKey] = ps.[ProductSubcategoryKey]
ORDER BY ps.[EnglishProductSubcategoryName], p.[EnglishProductName]
```

--2. (4) List the name of all Sales Reasons that have not been associated with a sale. Add a meaningful
sort. Explanation: AdventureWorks has a prepopulated list of reasons why customers purchase their
products. You are finding the reasons on the list that have not been selected by a customer buying
over the Internet. Hint: Use DimSalesReason and FactInternetSalesReason FactInternetSalesReason
and test for null in the matching field in the fact table.

/*--Checking SalesReasonKey (shows ten with 7 Listings, 3 are missing.)

```
SELECT distinct SalesReasonKey
FROM [dbo].[FactInternetSalesReason]
ORDER BY SalesReasonKey
```

*/

```
SELECT dsr.SalesReasonKey AS dsr_SalesReson_Key
      ,dsr.SalesReasonName
      ,fis.SalesReasonKey AS fis_SalesReason_Key
      ,fis.SalesOrderNumber
FROM [dbo].[DimSalesReason] dsr
LEFT JOIN [dbo].[FactInternetSalesReason] fis ON dsr.SalesReasonKey = fis.SalesReasonKey
WHERE fis.SalesOrderNumber IS NULL
ORDER BY dsr.SalesReasonKey ASC
```

/*

```
SELECT * FROM [dbo].[DimSalesReason]
SELECT TOP 10 * FROM [dbo].[FactInternetSalesReason]
```

*/

```

-----
/* CHECKING TABLES
SELECT *
FROM [dbo].[FactInternetSales]

SELECT *
FROM [dbo].[FactResellerSales]
*/

--3. (4) List all internet sales that do not have a sales reason associated. List SalesOrderNumber,
--      SalesOrderLineNumber and the order date. Add a meaningful sort.
--      Explanation: Now we are looking at sales reasons from another angle. Above we wanted to know
which
--      sales reasons had not been used, so we wanted the reason name. Now we are looking at which sales
do not
--      have a reason associated with the sale. Since we are looking at the sales, we don't need the
reason name
--      and the corresponding link to that table. Hint: Use FactInternetSales and
FactInternetSalesReason.
/*
SELECT DISTINCT * --fis.SalesOrderNumber, fis.SalesOrderLineNumber
FROM [dbo].FactInternetSales AS fis
SELECT DISTINCT *
FROM [dbo].FactInternetSalesReason AS fisr
SELECT DISTINCT *
FROM [dbo].[DimSalesReason]
*/

SELECT DISTINCT
    fis.SalesOrderNumber
    ,fis.SalesOrderLineNumber
    --,fisr.SalesOrderNumber AS fact_SaleReason_SalesOrder
    --,fisr.SalesOrderLineNumber AS fact_SaleReason_SalesLineNum
    ,fis.OrderDate
    --,fisr.SalesReasonKey
FROM [dbo].[FactInternetSales] AS fis
LEFT JOIN [dbo].[FactInternetSalesReason] AS fisr
ON (fisr.SalesOrderNumber = fis.SalesOrderNumber AND fisr.SalesOrderLineNumber =
fis.SalesOrderLineNumber) --(Two keys represent each record)
WHERE fisr.SalesReasonKey IS NULL
ORDER BY fis.OrderDate DESC, fis.SalesOrderNumber

--4.a. (4) List all promotions that have not been associated with a reseller sale. Show only
--      the English promotion name in alphabetical order.
--      Hint: Recall that details about sales to resellers are recorded in the FactResellerSales table.

SELECT dp.[EnglishPromotionName] --, dp.PromotionKey
FROM [dbo].[DimPromotion] AS dp
LEFT JOIN [dbo].[FactResellerSales] AS frs ON dp.PromotionKey = frs.PromotionKey
WHERE frs.PromotionKey IS NULL
ORDER BY dp.[EnglishPromotionName] ASC

/*
--Checking results of Promotion Keys Not Shown
SELECT *
FROM [dbo].[FactResellerSales]
WHERE PromotionKey IN(6, 10, 12, 15)
*/

```

```
--4.b. (3) List all promotions that have not been associated with an internet sale. Show only
--         the English promotion name in alphabetical order.
--         Hint: Recall that details about sales to customers are recorded in the FactInternetSales table.
```

```
SELECT dp.[EnglishPromotionName] --,dp.PromotionKey, fis.PromotionKey
FROM [dbo].[DimPromotion] AS dp
     LEFT JOIN [dbo].[FactInternetSales] AS fis ON dp.PromotionKey = fis.PromotionKey
WHERE fis.PromotionKey IS NULL
ORDER BY dp.[EnglishPromotionName] ASC
```

```
--4.c. +3 Bonus. Write an INTERSECT to show the name of promotions that have not been associated
--         with either a reseller sale or an internet sale. HINT: This can be done with
--         copy/paste from 4a and b, dropping one order by clause, and the addition of one keyword.
```

```
SELECT dp.[EnglishPromotionName] --, dp.PromotionKey
FROM [dbo].[DimPromotion] AS dp
     LEFT JOIN [dbo].[FactResellerSales] AS frs ON dp.PromotionKey = frs.PromotionKey
WHERE frs.PromotionKey IS NULL
--ORDER BY dp.[EnglishPromotionName] ASC
```

```
INTERSECT
```

```
SELECT dp.[EnglishPromotionName] --,dp.PromotionKey, fis.PromotionKey
FROM [dbo].[DimPromotion] AS dp
     LEFT JOIN [dbo].[FactInternetSales] AS fis ON dp.PromotionKey = fis.PromotionKey
WHERE fis.PromotionKey IS NULL
ORDER BY dp.[EnglishPromotionName] ASC
```

```
--5.a. (4) Find any PostalCodes in which AdventureWorks has no internet customers.
--         List Postal Code and the English country/region name.
--         List each Postal Code only one time. Sort by country and postal code.
```

```
SELECT DISTINCT
    dg.[PostalCode]
    ,dg.[EnglishCountryRegionName]
    --,dg.[GeographyKey]
    --,dg.GeographyKey
    --,fis.CustomerKey
FROM [dbo].[DimGeography] AS dg
     LEFT JOIN [dbo].[DimCustomer] AS dc ON dg.[GeographyKey] = dc.[GeographyKey]
     LEFT JOIN [dbo].[FactInternetSales] AS fis ON dc.CustomerKey = fis.CustomerKey
WHERE fis.CustomerKey IS NULL --dc.GeographyKey IS NULL
ORDER BY dg.[EnglishCountryRegionName], dg.[PostalCode]
```

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

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

```
--5.b (3) Find any PostalCodes in which AdventureWorks has no resellers.
--         List Postal Code and the English country/region name.
--         List each Postal Code only one time. Sort by country and postal code.
```

```
SELECT DISTINCT
    dg.[PostalCode]
    ,dg.[EnglishCountryRegionName]
FROM [dbo].[DimGeography] AS dg
     LEFT JOIN [dbo].[DimReseller] AS dr ON dg.[GeographyKey] = dr.[GeographyKey]
     LEFT JOIN [dbo].[FactResellerSales] AS frs ON dr.ResellerKey = frs.ResellerKey
WHERE frs.ResellerKey IS NULL
ORDER BY dg.[EnglishCountryRegionName], dg.[PostalCode]
```

```

/*
SELECT *
FROM [dbo].[FactResellerSales]
SELECT *
FROM [dbo].[DimReseller]
*/
--5.c. +2 Bonus. Write an INTERSECT to show Postal Codes in which AdventureWorks has neither
--      Internet customers nor resellers.

```

```

SELECT DISTINCT
    dg.[PostalCode]
    ,dg.[EnglishCountryRegionName]
    --,dg.[GeographyKey]
    --,dg.GeographyKey
    --,fis.CustomerKey
FROM [dbo].[DimGeography] AS dg
    LEFT JOIN [dbo].[DimCustomer] AS dc ON dg.[GeographyKey] = dc.[GeographyKey]
    LEFT JOIN [dbo].[FactInternetSales] AS fis ON dc.CustomerKey = fis.CustomerKey
WHERE fis.CustomerKey IS NULL --dc.GeographyKey IS NULL
--ORDER BY dg.[EnglishCountryRegionName], dg.[PostalCode]

```

INTERSECT

```

SELECT DISTINCT
    dg.[PostalCode]
    ,dg.[EnglishCountryRegionName]
FROM [dbo].[DimGeography] AS dg
    LEFT JOIN [dbo].[DimReseller] AS dr ON dg.[GeographyKey] = dr.[GeographyKey]
    LEFT JOIN [dbo].[FactResellerSales] AS frs ON dr.ResellerKey = frs.ResellerKey
WHERE frs.ResellerKey IS NULL
ORDER BY dg.[EnglishCountryRegionName], dg.[PostalCode]

```

--6.a. (4) List the name of all currencies and the name of each organization that uses that currency.
 -- You will use an Outer Join to list the name of each currency in the Currency table regardless if
 -- it has a matching value in the Organization table. You will see NULL in many rows. Add a
 -- meaningful sort. Hint: Use DimCurrency and DimOrganization.

```

SELECT dc.[CurrencyName]
    , do.[OrganizationName]--, do.[CurrencyKey]
FROM [dbo].[DimCurrency] AS dc
    LEFT JOIN [dbo].[DimOrganization] AS do ON dc.CurrencyKey = do.CurrencyKey
ORDER BY dc.[CurrencyName] ASC

```

--6.b. (2) List the name of all currencies that are NOT used by any organization. In this situation
 -- we are using the statement from 6.a. and making a few modifications. We want to find the
 -- currencies that do not have a match in the common field in the Organization table.
 -- Sort ascending on currency name.

```

SELECT dc.[CurrencyName]
    , do.[OrganizationName]--, do.[CurrencyKey]
FROM [dbo].[DimCurrency] AS dc
    LEFT JOIN [dbo].[DimOrganization] AS do ON dc.CurrencyKey = do.CurrencyKey
WHERE do.[CurrencyKey] IS NULL
ORDER BY dc.[CurrencyName] ASC

```

--7.a. (3) List the unique name of all currencies and the CustomerKey of customers that use that currency. You will list the name of each currency in the Currency table regardless if it has a matching value in the Internet Sales table. You will see some currencies are repeated because more than one customer uses the currency. You may see the CustomerKey repeated because a customer may buy in more than one currency. You will see NULL in a few rows. Add a meaningful sort. Hint: This will be all customers, with some duplicated, and the unused currencies; 18,983 rows.

```
SELECT DISTINCT
    dc.CurrencyName
    ,fis.CustomerKey
FROM [dbo].[DimCurrency] AS dc
LEFT JOIN [dbo].[FactInternetSales] AS fis ON dc.CurrencyKey = fis.CurrencyKey
ORDER BY CurrencyName, fis.CustomerKey
```

--7.b. (2) Copy/paste 7.a. to 7.b. Modify 7.b. to list only the unique name of currencies that are not used by any internet customer. Add a meaningful sort. This will be a small number--just unused currencies.

```
SELECT DISTINCT
    dc.CurrencyName
FROM [dbo].[DimCurrency] AS dc
LEFT JOIN [dbo].[FactInternetSales] AS fis ON dc.CurrencyKey = fis.CurrencyKey
WHERE fis.CustomerKey IS NULL
ORDER BY CurrencyName
```

--7.c. (4) This question is a variation on 7.a. You will need to join to an additional table. List the unique name of all currencies, the last name, first name, and the CustomerKey of customers that use that currency. You will list the name of each currency in the Currency table regardless if it has a matching value in the Internet Sales table. Same number of rows as 7.a.

```
SELECT DISTINCT
    dc.CurrencyName
    ,c.[LastName]
    ,c.[FirstName]
    ,fis.CustomerKey
FROM [dbo].[DimCurrency] AS dc
LEFT JOIN [dbo].[FactInternetSales] AS fis ON dc.CurrencyKey = fis.CurrencyKey
LEFT JOIN [dbo].[DimCustomer] AS c ON fis.CustomerKey = c.CustomerKey
ORDER BY CurrencyName, fis.CustomerKey
```

```
/*
SELECT *
FROM [dbo].[DimCurrency]
SELECT *
FROM [dbo].[FactInternetSales]
SELECT *
FROM [dbo].[DimCustomer]
*/
```

-- READ 8.a. and 8.b. BEFORE beginning the syntax. Hint: Refer to the Outer Joins Demo and look at the example where a query is used in place of table for one possible method of answering these two questions. They can also be done with multiple joins. NULL will show in the ResellerName and OrderDate for a few records. We are showing ALL promotions.

```
--8.a. (4) Find all promotions and any related reseller sales. List unique instances of the
--         English promotion name, reseller name, and the order date. Show the OrderDate as mm/dd/yyyy.
--         Sort by the promotion name. Be sure to list all promotion names even if there is no related
sale.
```

```
SELECT DISTINCT
    dp.[EnglishPromotionName]
    ,dr.[ResellerName]
    ,CONVERT(VARCHAR(10), frs.[OrderDate], 101) AS OrderDate
FROM [dbo].[DimPromotion] AS dp
    LEFT JOIN [dbo].[FactResellerSales] AS frs ON dp.PromotionKey = frs.PromotionKey
    LEFT JOIN [dbo].[DimReseller] AS dr ON frs.ResellerKey = dr.ResellerKey
ORDER BY dp.[EnglishPromotionName]
```

```
--8.b. (3) Copy, paste, and modify 8.a. "No Discount" is not a promotion; eliminate those sales
--         without a promotion from your results set. Show the OrderDate as mm/dd/yyyy.
--         Look for ways to double-check your results.
```

```
SELECT DISTINCT
    dp.[EnglishPromotionName]
    ,dr.[ResellerName]
    ,CONVERT(VARCHAR(10), frs.[OrderDate], 101) AS OrderDate
FROM [dbo].[DimPromotion] AS dp
    LEFT JOIN [dbo].[FactResellerSales] AS frs ON dp.PromotionKey = frs.PromotionKey
    LEFT JOIN [dbo].[DimReseller] AS dr ON frs.ResellerKey = dr.ResellerKey
WHERE dp.[EnglishPromotionName] <> 'No Discount' -- <> meaning !=
ORDER BY dp.[EnglishPromotionName]
```

```
--9. (2) In your own words, write a business question that you can answer by querying the data warehouse
--         and using an outer join. Be sure that your business question appears as a comment (all green)
--         Then write the SQL query that will provide the information that you are seeking.
```

```
--Business Question in my own words:
--List all countries do not have internet sales.
--List Columns of CountryName, StateProvinceName, PostalCode.
```

```
SELECT DISTINCT
    dg.EnglishCountryRegionName
    ,dg.StateProvinceName
    ,dg.PostalCode
FROM [dbo].[DimGeography] AS dg
    LEFT JOIN [dbo].[DimCustomer] dc ON dg.GeographyKey = dc.GeographyKey
    LEFT JOIN [dbo].[FactInternetSales] AS fis ON dc.CustomerKey = fis.CustomerKey
WHERE fis.CustomerKey IS NULL
```

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

```
SELECT * FROM [dbo].[DimCustomer]
```

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

```
--NOTES:
/*Art, Please find the enclosed assignment 8. Thank you! Chris Singleton
Chris Singleton, Nov 15 at 10:47pm
Excellent work Chris! A few comments: - #1 should be a LEFT join instead of an INNER join. -
#2 should just show the SalesReasonName column - 4c; +3 - 5b should only join DimGeography and DimReseller;
your WHERE would then be where the [DimReseller].[ResellerKey] IS NULL - 5c; +1 (results differ due to 5b) -
#9; if your question is to just list the Countries, no point in listing states and postal codes :-)
```

```
/*
```