```
--*  BUSIT 103                          Assignment   #10                          DUE DATE :  Consult course
calendar
/*
Name: Christopher Singleton
Class: BUSIT103 - Online
Instructor: Art Lovestedt
Date: 11/29/2014
*/
--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.,
GriggsDebiAssignment10.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.
--  These statements will NOT use GROUP BY or HAVING. Those keywords are introduced in the next module. One
cell
--     results sets (one column and one row) do not need to have an ORDER BY clause.

--     Recall that sales to resellers are stored in the FactResellerSales table and sales to customers are
stored in
--     the FactInternetSales table.

--     Do not remove the USE statement
USE AdventureWorksDW2012;

-- 1.a.       (3) Find the count of customers who are married. Be sure give each derived field
--            an appropriate alias.

SELECT COUNT([MaritalStatus]) AS 'Count_CustMarried'
FROM [dbo].[DimCustomer]
WHERE [MaritalStatus] = 'M'

--Customers who are married (Using the COUNT Function): 10011

--1.b. (2) Check your result. Write queries to determine if the answer to 1.a. is correct.
--            You should be writing proofs for all of your statements.

SELECT Count([CustomerKey]) AS CustomerKey
FROM [dbo].[DimCustomer]
WHERE [MaritalStatus] = 'M'

--Returns 10011 rows.

--1.c. (3) Find the total children (sum) and the total cars owned (sum) for customers who are married.

SELECT SUM([TotalChildren]) AS 'SUM_CustTotalChildren',
       SUM([NumberCarsOwned]) AS 'SUM_CustTotalCarsOwned'
FROM [dbo].[DimCustomer]
WHERE [MaritalStatus] = 'M'

--1.d. (2) Find the total children, total cars owned, and average yearly income for customers who are married.

SELECT SUM([TotalChildren]) AS 'SUM_CustTotalChildren',
       SUM([NumberCarsOwned]) AS 'SUM_CustTotalCarsOwned',
          AVG([YearlyIncome]) AS 'AVG_CustYearlyIncome'
FROM [dbo].[DimCustomer]
WHERE [MaritalStatus] = 'M'

--2.a. (2) List the total dollar amount (SalesAmount) for sales to Resellers. Round to two decimal places.

SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2)) AS Total_ResellerSales
FROM [dbo].[FactResellerSales]
```

```sql
--2.b. (3) List the total dollar amount (SalesAmount) for 2008 sales to resellers in Germany.
--            Show only the total sales--one row, one column--rounded to two decimal places.

SELECT CAST(SUM([SalesAmount]) AS Decimal(8,2)) AS Total_ResellerSales
FROM [dbo].[FactResellerSales] AS frs
     INNER JOIN [dbo].[DimReseller] AS dr ON frs.ResellerKey = dr.ResellerKey
     INNER JOIN [dbo].[DimGeography] AS dg ON dr.[GeographyKey] = dg.[GeographyKey]
WHERE dg.EnglishCountryRegionName = 'Germany'
      AND frs.[OrderDateKey] BETWEEN 20080101 AND 20081231
        --(frs.[OrderDateKey] >= 20080101 AND frs.[OrderDateKey] <= 20081231)

--3.a. (2) List the total dollar amount (SalesAmount) for sales to Customers. Round to two decimal places.

SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2)) AS TotalCust_SalesAmount
FROM [dbo].[FactInternetSales]


--3.b.  (3) List the total dollar amount (SalesAmount) for 2005 sales to customers located in the
--            United Kingdom. Show only the total sales--one row, one column--rounded to two decimal places.

SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2)) AS TotalCustSalesAmount_2005
FROM [dbo].[FactInternetSales] AS fis
     INNER JOIN [dbo].[DimCustomer] AS c ON fis.CustomerKey = c.CustomerKey
        INNER JOIN [dbo].[DimGeography] AS g ON c.GeographyKey = g.GeographyKey
WHERE YEAR(fis.OrderDate) = 2005 AND g.EnglishCountryRegionName = 'United Kingdom'

--4.   (4) List the average unit price for a touring bike sold to customers. Round to
--            two decimal places.

SELECT CAST(AVG(fis.[UnitPrice]) AS Decimal(6,2)) AS TourBike_AVGUnitPrice
FROM [dbo].[FactInternetSales] AS fis
     INNER JOIN [dbo].[DimCustomer] AS c ON fis.CustomerKey = c.CustomerKey
        INNER JOIN [dbo].[DimProduct] AS p ON fis.ProductKey = p.ProductKey
WHERE p.[ProductSubcategoryKey] = 3

--5.   (5) List bikes that have a list price less than the average list price for all bikes.
--            Show product key, English product name, and list price.
--            Order descending by list price.

SELECT p.[ProductKey]
      ,p.[EnglishProductName]
         ,p.[ListPrice]
FROM [dbo].[DimProduct] AS p
WHERE p.ProductSubCategoryKey IN (1,2,3)
      AND p.[ListPrice] < (SELECT AVG([ListPrice])
                           FROM [dbo].[DimProduct]
                                        WHERE ProductSubCategoryKey IN (1,2,3))
ORDER BY p.[ListPrice] DESC
```

```sql
--6.   (4) List the lowest list price, the average list price, the highest list price, and product count for
road bikes.

SELECT MIN(ListPrice) AS MinListPrice
      ,AVG(DISTINCT ListPrice) AS AvgListPrice
         ,MAX(ListPrice) AS MaxListPrice
         ,COUNT(ProductSubcategoryKey) AS NumOfRoadBikes
FROM [dbo].[DimProduct]
WHERE ProductSubcategoryKey = 2

/*--Another way to COUNT:
SELECT MIN(ListPrice) AS MinListPrice
      ,AVG(DISTINCT ListPrice) AS AvgListPrice
         ,MAX(ListPrice) AS MaxListPrice
         ,COUNT([EnglishProductName]) AS NumOfRoadBikes
FROM [dbo].[DimProduct]
WHERE [EnglishProductName] LIKE 'Road%'AND ProductSubcategoryKey = 2

SELECT [EnglishProductName]
FROM [dbo].[DimProduct]
*/
/*
SELECT ProductSubcategoryKey
FROM [dbo].[DimProduct]
*/
--SELECT ListPrice, ProductSubcategoryKey FROM [dbo].[DimProduct] WHERE ProductSubcategoryKey = 2


-- 7.  (4) List the product alternate key, product name, and list price for the product(s)
--            with the lowest List Price. There can be multiple products with the lowest list price.

SELECT p1.ProductAlternateKey
      ,p1.EnglishProductName
         ,p1.ListPrice AS LowestListPrice
FROM [dbo].[DimProduct] AS p1
WHERE p1.ListPrice = (SELECT MIN(ListPrice)
                      FROM [dbo].[DimProduct] AS p2
                                    WHERE p1.ProductAlternateKey = p2.ProductAlternateKey)
ORDER BY p1.ProductAlternateKey, p1.EnglishProductName

/* --Checking:
ProductAlternateKey  EnglishProductName     ListPrice
BK-M68S-42            Mountain-200 Silver, 42      2071.4196 (MIN)
BK-M68S-42            Mountain-200 Silver, 42      2319.99
*/

-- 8.a.       (4) List the product alternate key, product name, list price, dealer price, and the
--            difference (calculated field) for all product(s). Show all money values to 2 decimal places.
--            Sort on difference from highest to lowest.

SELECT [ProductAlternateKey]
      ,[EnglishProductName]
         ,CAST([ListPrice] AS Decimal(8,2)) AS ListPrice
         ,CAST([DealerPrice] AS Decimal(8,2)) AS DealerPrice
         ,CAST(ListPrice - DealerPrice AS Decimal(8,2)) AS PriceDifference
FROM [dbo].[DimProduct]
ORDER BY PriceDifference DESC
```

```
-- 8.b.       (3) Use the statement from 8.a. and modify to find the product(s) with the largest difference
--            between the list price and the dealer price. Show all money values to 2 decimal places.

/*Checking:
ProductAlternateKey  EnglishProductName      ListPrice  DealerPrice   PriceDifference
BK-M68S-42           Mountain-200 Silver, 42     2071.42        1242.85          828.57
BK-M68S-42           Mountain-200 Silver, 42     2319.99            1391.99      928.00 (MAX)
*/
SELECT [ProductAlternateKey]
      ,[EnglishProductName]
         ,CAST(p1.[ListPrice] AS Decimal(8,2)) AS ListPrice
         ,CAST(p1.[DealerPrice] AS Decimal(8,2)) AS DealerPrice
         ,CAST(p1.ListPrice - p1.DealerPrice AS Decimal(8,2)) AS PriceDifference
FROM [dbo].[DimProduct] AS p1
WHERE p1.ListPrice - p1.DealerPrice =
                                    (SELECT MAX(ListPrice - DealerPrice)
                                     FROM [dbo].[DimProduct] AS p2
                                          WHERE p1.ProductAlternateKey = p2.ProductAlternateKey)
ORDER BY PriceDifference DESC


-- 9.  (4) List total Internet sales for product BK-M82S-44 using two methods: Total the sales amount field
--            and calculate the total amount using unit price and quantity. Place both calculations in
different
--            columns in the SAME Select statement. There will be one results set with two columns and one
row.
--            Show all money values to 2 decimal places. The values should be the same.

SELECT CAST(SUM(fis.[SalesAmount]) AS Decimal(8,2)) AS TotalSalesAmount
      ,CAST(SUM(fis.[UnitPrice] * fis.[OrderQuantity]) AS Decimal(8,2)) AS ComputedTotalSalesAmount
FROM [dbo].[FactInternetSales] AS fis
     INNER JOIN [dbo].[DimProduct] AS p ON fis.[ProductKey] = p.[ProductKey]
WHERE ProductAlternateKey = 'BK-M82S-44'

--10.  (2) In your own words, write a business question that you can answer by querying the data warehouse
--            and using an aggregate function.
--            Then write the complete SQL query that will provide the information that you are seeking.
/*My Question:
  List total internet sales in 2007 and 2008.
  Also, list total internet sales between 2007 and 2008.
  Round to two decimal places and give alias for three coloumns.
*/

SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2)) AS TotalSalesAmount2007  --2007 Outer query
      ,(SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2))                   --2008 Subquery
              FROM [dbo].[FactInternetSales]
              WHERE YEAR([OrderDate]) = 2008) AS TotalSalesAmount2008
         ,CAST(SUM([SalesAmount]) - (SELECT SUM([SalesAmount])                --2007-2008 = Difference
Calculated Subquery
                                     FROM [dbo].[FactInternetSales]
                                                WHERE YEAR([OrderDate]) = 2008) AS Decimal(8,2)) AS
YearlySalesDifference
FROM [dbo].[FactInternetSales]
WHERE YEAR([OrderDate]) = 2007
```

```
/*
Nice job on this Chris - just a few quick notes :-): - 1b isn't really a proof; it is just counting a
different column.
There are several ways you could "prove" your query from #1; one might be to just SELECT * without a count
and see how many rows are returned; another may be to count total customers, then count single customers,
and know that total - married should equal single, etc. -

#6 should not use DISTINCT on the AVG; If we use a traditional approach to this,
you would say SUM(ListPrice)/COUNT(ListPrice) - but if you call "DISTINCT", it won't take into account items
that have the same ListPrice. So by using DISTINCT, the average price is 1550.5168,
but when you take it out, it changes to 1430.6081; this is an important difference, and we want to include all
items prices
 (even if they are the same) when calculating the average price of a product or group of products :-) -

#7 should just return one row: SELECT ProductAlternateKey, EnglishProductName, ListPrice FROM dbo.DimProduct
 WHERE ListPrice = (SELECT MIN(ListPrice) FROM dbo.DimProduct) ORDER BY ListPrice ASC; -
 #8b can be simplified a bit to: SELECT ProductAlternateKey, EnglishProductName, ListPrice,
 ROUND(DealerPrice, 2) AS DealerPrice, ROUND((ListPrice - DealerPrice), 2) AS PriceDifference FROM
dbo.DimProduct
 WHERE (ListPrice - DealerPrice) = (SELECT MAX(ListPrice - DealerPrice) FROM dbo.DimProduct)
 ORDER BY PriceDifference DESC;

*/
```