```sql
--*  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.
       You should test each SQL statement using the database shown in the USE statement. The SQL
statement should
       execute against that database without errors. */

/*     Submit your .sql file named with your last name, first name and assignment # (e.g.,
GriggsDebiAssignment4.sql).
       Submit your file to the instructor using through the course site.  Do not zip the file. */

--Do not remove the USE statement
USE AdventureWorksLT2012;

--1.  (4) Use the SalesLT.Address table to list addresses in the United States. Select the address1,
--    city, state/province, country/region and postal code. Sort by state/province and city.

SELECT [AddressLine1]
     , [City], [StateProvince]
        , [CountryRegion]
        , [PostalCode]
FROM [SalesLT].[Address]
ORDER BY [StateProvince], [City]


--2.  (5) Use the SalesLT.Address table to list addresses in the US states of Idaho or Illinois.
--    Select the address1, city, state/province, country/region and postal code. Sort by state/province
and city.

SELECT [AddressLine1]
     , [City], [StateProvince]
        , [CountryRegion]
        , [PostalCode]
FROM [SalesLT].[Address]
WHERE [StateProvince] = 'Idaho' OR [StateProvince] = 'Illinois'
ORDER BY [StateProvince], [City]

--3.   (4) Use the SalesLT.Address table to list addresses in the cities of Victoria or Vancouver.
--            Select the address1, city, state/province, country/region and postal code.
--            Order the list by city.

SELECT [AddressLine1]
     , [City], [StateProvince]
        , [CountryRegion]
        , [PostalCode]
FROM [SalesLT].[Address]
WHERE [City] = 'Victoria' OR [City] = 'Vancouver'
ORDER BY [City]
```

```sql
--4.   (5) Use the SalesLT.Address table to list addresses in the cities of Victoria or Vancouver in
the
--            Canadian province of British Columbia. Select the address1, city, state/province,
country/region
--            and postal code. Order the list by city.

SELECT [AddressLine1]
     , [City], [StateProvince]
        , [CountryRegion]
        , [PostalCode]
FROM [SalesLT].[Address]
WHERE ([City] = 'Victoria' OR [City] = 'Vancouver') AND [StateProvince] = 'British Columbia'
ORDER BY [City]

--Above is correct, use parentheses when using AND and OR together.

--5.   (4) List the company name and phone for those customers whose phone number contains the
following sequence: 34.
--            Order the list by phone number in ascending order. "Contains" means that the sequence
exists within the phone number.

SELECT [CompanyName], [Phone]
FROM [SalesLT].[Customer]
WHERE [Phone] LIKE '%34%'
ORDER BY [Phone] ASC

--6.   (4) List the name, product number, size, standard cost, and list price in alphabetical order by
name
--            for Products whose standard cost is $1500 or more. Show all money values at exactly two
decimal places.
--            Be sure to give each derived column an alias.

SELECT [Name]
      ,[ProductNumber]
         ,[Size]
         ,[StandardCost]
         ,[ListPrice]
      ,CAST([StandardCost] AS DECIMAL(6,2)) AS [StandardCost_Rounded]
--    ,CAST([StandardCost] AS MONEY) AS [StandardCost_Rounded]
--       ,'$'+ CAST(CAST([StandardCost] AS DECIMAL(6,2)) as VARCHAR(7)) AS [StandardCost_Rounded]
FROM [SalesLT].[Product]
WHERE [StandardCost] >= 1500
ORDER BY [Name] ASC

--Notes:
--SELECT MAX(LEN([StandardCost]))
--FROM [SalesLT].[Product]
--Note: You can do the CAST function ether way, but remember if you use the '$' string, you must use
character format.

--7.   (4) List the name, product number, size, standard cost, and list price in alphabetical order by
name
--            for Products whose list price is $100 or less and standard cost is $40 or more.

SELECT [Name]
      ,[ProductNumber]
         ,[Size]
         ,[StandardCost]
         ,[ListPrice]
FROM [SalesLT].[Product]
WHERE [ListPrice] <= 100 AND [StandardCost] >= 40
ORDER BY [Name] ASC
```

```sql
--8.   (4) List the name, standard cost, list price, and size for products whose size is one of the
--            following:  XS, S, M, L, XL. Show all money values at exactly two decimal places. Be sure
to
--            give each derived column an alias. Order the list by name in alphabetical order.

SELECT [Name]
      ,[StandardCost]
         ,[ListPrice]
         ,[Size]
         ,CAST([StandardCost] AS DECIMAL(4,2)) AS [StandardCost_Rounded]
--       ,'$'+ CAST(CAST([StandardCost] AS DECIMAL(6,2)) AS VARCHAR(7)) AS [StandardCost_Rounded]
FROM [SalesLT].[Product]
WHERE [Size] LIKE 'XS'
   OR [Size] LIKE 'S'
   OR [Size] LIKE 'M'
   OR [Size] LIKE 'L'
   OR [Size] LIKE 'XL'
ORDER BY [Name] ASC

--Notes:
--SELECT MAX(LEN([StandardCost]))
--FROM [SalesLT].[Product]
--SELECT standardcost from [SalesLT].[Product]
--In order to have the $ symbol, you need to change the DataType to Varchar (character format).

--9.   (4) List the name, product number, and sell end date for all products in the Product table
--            that are not currently sold. Sort by the sell end date from most recent to oldest date.
Show
--            only the date (no time) in the sell end date field. Be sure to give each derived column
an alias.

SELECT [Name]
     , [ProductNumber]
        , [SellEndDate]
     ,CAST([SellEndDate] AS DATE) AS [SellEndDate_DATE]
FROM [SalesLT].[Product]
WHERE [SellEndDate] IS NOT NULL
ORDER BY [SellEndDate] DESC

--10.  (6) List the name, product number, standard cost, list price, and weight for products whose
standard cost
--    is less than $50, list price is greater than $100, and weight is greater than 1,000. Round money
values
--    to exactly 2 decimal places and give each derived column a meaningful alias. Sort by weight.

SELECT [Name]
     , [ProductNumber]
        , [StandardCost]
        , [ListPrice]
        , [Weight]
        ,CAST([StandardCost] AS DECIMAL(4,2)) AS [StandardCost_Rounded]
--    ,'$'+ CAST(CAST([StandardCost] AS DECIMAL(6,2)) AS VARCHAR(7)) AS [StandardCost_Rounded]
FROM [SalesLT].[Product]
WHERE ([StandardCost] < 50 AND [ListPrice] > 100) AND [Weight] > 1000
--WHERE ([Weight] > 1000 AND [ListPrice] > 100) AND [StandardCost] < 50
ORDER BY [Weight]

--Note: You can do the CAST function ether way, but remember if you use the '$' string, you must use
character format.
```

```sql
--11. In a and b below, explore the data to better understand how to locate products.

--a.  (3) List the name, product number, and product category ID for all products in the Product
--    table that include 'bike' in the name. Sort by the name.
--    Something to consider: How many of these products are actually bikes?

SELECT [Name]
     , [ProductNumber]
       , [ProductCategoryID]
FROM [SalesLT].[Product]
WHERE [NAME] like '%Bike%'
ORDER BY  [Name]

--Note: Bike cannot be all lower case letters.
--Only two in this result are actually Bikes. (ProductCategoryID 27)

--b.  (3) List the name and product category id, and parent id for all categories in the product
category
--    table that include 'bike' in the name. Sort by the parent product category id.
--    Something to consider: How many of these product categories are actually bikes? What is the
--       ProductCategoryID for Bikes?

SELECT [Name]
     , [ProductCategoryID]
       , [ParentProductCategoryID]
FROM [SalesLT].[ProductCategory]
WHERE  [NAME] LIKE '%Bike%'
ORDER BY [ParentProductCategoryID]

--Only 4 Product Categories have the Name 'Bikes' that are actually bikes.
--ProductCategoryID for Bikes are 1, 5, 6 and 7.
--Note: This is case sensative.
 /*Nice job on this Chris! A few quick notes: - #'s 6, 8 and 10 should also have ListPrice defined as 2
decimal places
  - #8 shouldn't use "LIKE" as your comparator, it should be = or use the "IN" method
Art Lovestedt, Oct 26 at 3:45p*/
```