Project 1 – SalesOrderExample Database from your book (100 points)

For this project you will use only the **SalesOrderExample** database from our book. The data diagram is on page 724. I also have scanned the page in and included it in the Reference Material Module. Each question is worth 7 points and the excel output file is worth 2 points.

Paste your query after every question. You will be using INNER JOINS and can use any of the syntax shown in the ‘Four ways to write an INNER JOIN’ document in the Reference Material Module

1. Show the customerid, first name, last name, order number and orderdate for customerid 1001. Should have 44 rows in your result set.

select distinct cus.CustomerID, cus.CustFirstName, cus.CustLastName, od.OrderNumber, od.OrderDate

from dbo.Customers cus, dbo.Orders as od

where cus.CustomerID = od.CustomerID

and cus.CustomerID = 1001

2. Add the INNER JOIN for the order\_details table. Should have 130 rows in your result set.

select distinct cus.CustomerID, cus.CustFirstName, cus.CustLastName, o.OrderNumber, o.OrderDate

from dbo.Customers as cus

join dbo.Orders as o

on cus.CustomerID = o.CustomerID

join dbo.Order\_Details as od

on o.OrderNumber = od.OrderNumber

where cus.CustomerID = 1001

3. Add the INNER JOIN for the Products table and add the productname to your Select statement. There is one product name for every product number, so should still have 130 rows in your result set.

select distinct cus.CustomerID, cus.CustFirstName, cus.CustLastName, o.OrderNumber, o.OrderDate, pro.ProductName

from dbo.Customers as cus

join dbo.Orders as o

on cus.CustomerID = o.CustomerID

join dbo.Order\_Details as od

on o.OrderNumber = od.OrderNumber

join dbo.Products as pro

on od.ProductNumber = pro.ProductNumber

where cus.CustomerID = 1001

4. Add the INNER JOIN for the Categories table and add the categorydescription to your select statement. Each category has one description, so still 130 rows. You have joined 5 tables, Congratulations!

select distinct cus.CustomerID, cus.CustFirstName, cus.CustLastName, o.OrderNumber, o.OrderDate, pro.ProductName

from dbo.Customers as cus

join dbo.Orders as o

on cus.CustomerID = o.CustomerID

join dbo.Order\_Details as od

on o.OrderNumber = od.OrderNumber

join dbo.Products as pro

on od.ProductNumber = pro.ProductNumber

join dbo.Categories as cat

on cat.CategoryID = pro.CategoryID

where cus.CustomerID = 1001

5. Starting a new query, JOIN the employees and orders tables. List the order numbers and order date, for any orders for the employee with the last name of ‘Berg’. Should be 105 rows.

select OrderNumber, OrderDate

from dbo.Employees as emp

join dbo.Orders as o

on emp.EmployeeID = o.EmployeeID

where emp.EmpLastName = 'Berg'

6. Revise the query from the previous step, how many different (unique) customers does Berg place orders for. You do not need to add any tables to answer this question, just change your select statement. 24 rows in your result set.

select distinct CustomerID

from dbo.Employees as emp

join dbo.Orders as o

on emp.EmployeeID = o.EmployeeID

where emp.EmpLastName = 'Berg'

7. Using just the Categories table, what are the unique category descriptions. 6 rows, notice one description is Bikes which we will use in the next query. Paste your query here.

select distinct CategoryID, CategoryDescription

from dbo.Categories

8. JOIN the products and the categories tables. Select the product names that are in the category description of ‘Bikes’. Should be 4 rows in the result set.

select cat.CategoryID, CategoryDescription

from dbo.Categories as cat

join dbo.Products as pro

on cat.CategoryID = pro.CategoryID

where cat.CategoryID = 2

9. Using just the vendor table, find out what are the unique vendor names. You should have 10 rows in your result set. Paste your query here.

select distinct VendName, VendorID

from dbo.Vendors

10. JOIN the Products, ProductVendors and Vendors tables using INNER JOIN. Show me the vendor number and name and the product number and name. For only product numbers 5 to 25. Sort by vendor id. Should have 42 rows. Paste your query here.

select ven.VendorID, ven.VendName, pro.ProductNumber, pro.ProductName

from dbo.Product\_Vendors as pven

join dbo.Vendors as ven

on pven.VendorID = ven.VendorID

join dbo.Products as pro

on pro.ProductNumber = pven.ProductNumber

where pro.ProductNumber between 5 and 25

order by ven.VendorID

11. Using the same tables as the previous question, write a query to answer “How many products does “Sun Sports Suppliers” supply to us? 5 rows in the result set.

select distinct pro.ProductName, pro.ProductNumber

from dbo.Product\_Vendors as pven

join dbo.Vendors as ven

on pven.VendorID = ven.VendorID

join dbo.Products as pro

on pro.ProductNumber = pven.ProductNumber

where ven.VendorID = 8

12. Using only the Products table, list every product number and name, with the calculation of inventory value on hand (retail price times quantity on hand) and called this field InventoryValue in your result set. Should have 40 products in your result set.

select ProductNumber, ProductName, QuantityOnHand, cast(QuantityOnHand \* RetailPrice as int) as InventoryValue

from dbo.Products

13. Using only the Products table, list every product number and name that contains the word Pedal. Should be 3 rows in your result set.

select ProductNumber, ProductName

from dbo.Products

where ProductName like '%pedal%'

14. From the Customers table, concatenate the First Name and Last name with a space in between. Call this new field Fullname. Include the city, state and zip code. Sort by customerid. Example on page 135 but remember SQLServer2012 does not use the | symbol as explained on page 122.

select CustFirstName + ' ' + CustLastName as FullName, CustCity, CustState, CustZipCode

from dbo.Customers

Order By CustomerID

15. Once you run the query successfully, then select **Query** from the top menu bar. Then scroll down to select **Results to** - then **Results to File**. Then run the query again and it will ask you where to save your file which has an extension of rpt.

Then **open up Excel** and find the file that you saved. Since it ends in rpt., you will need excel to look for **all file types**. Once you open the file, you will want to select **Fixed width** so the data is separated into different cells. When it's open you need to "clean it up" which means delete the second row which contains dashes and at the bottom of the query it will say how many rows are in the result set. Delete that information and then save the file as **an excel file**, so the person opening it can just start working with the data.

Upload that file with your completed Project # 1 word document to MyPHSC.

Now try to run your query to display the results on the screen. Select Query from the menu bar, from the drop-down menu select Results To, then select Results to Grid.