Assignment 16 Name Oliver Conover

Use the **SalesOrderModify** database for all questions. Notice it’s SalesOrder**Modify** not SalesOrder**Example**. (50 points)

Remember to begin your query with:

Use SalesOrdersModify;  
SET IDENTITY\_INSERT Customers ON  
GO

1. We will add 10 customers to the Customers table. (15 points)

a. Run a query to see how many rows are in the Customer table? Paste your query here.

select count(\*)

from Customers

b. Write the number of rows here.

28

c. Run a query to find the largest customerid currently in the Customer table. Paste your query here.

select max(CustomerID) HighestIDNUM

from Customers

d. You are going to add 10 customers to the Customer Table. Use the file CH16\_Adding\_10\_Customer.sql which can be downloaded from the Chapter 16 Assignment module. I already typed all the data, so you are just going to execute the query to insert 10 rows.

In SQL2014, select the File tab then Open and select File and then browse to find the File. It is ready to be executed. Execute the query. For each row inserted in the table, the query displays the message “(1 row(s) affected)“ Did it run okay?

e. Run the query from step 1 a, to ensure that the 10 records were added to the table. Do you have 10 more rows then you answer to question 1.b.?

yes

f. Add a record by revising the VALUE section of the query you just executed.

1. Change the first record in the file by add your own first and last name. You don’t need to change all the other fields.

2. Running a query from 1.c. to find the current maximum customerid and add 1 to that number. Type that number in the first field of your insert query.

3. Delete all the extra rows of customer data, so you only insert one record. Paste your query here.

Use SalesOrdersModify;

SET IDENTITY\_INSERT Customers ON

GO

INSERT INTO Customers

(CustomerID, CustFirstName, CustLastName, CustStreetAddress,

CustCity, CustState, CustZipCode, CustAreaCode,

CustPhoneNumber)

VALUES (1039, 'Oliver', 'Conover', '15127 NE 24th, #383',

'Redmond', 'WA', '98052', 425, '555-2686');

g. Now change the last query you ran to add another record but let the system assign the next customerid. You only need to change the VALUES section of the query for the customerid, unless you want to change other fields. Page 548 and your Chapter 15 & 16 Key Points document shows you how to insert a query to find max(customerid) and add one. Paste your query here.

Use SalesOrdersModify;

SET IDENTITY\_INSERT Customers ON

GO

INSERT INTO Customers

(CustomerID, CustFirstName, CustLastName, CustStreetAddress,

CustCity, CustState, CustZipCode, CustAreaCode,

CustPhoneNumber)

VALUES ((select max(CustomerID) from Customers) + 1, 'Connor', 'MacDougall', '15127 NE 24th, #383',

'Redmond', 'WA', '98052', 425, '555-2686');

h. Run a query that shows all rows and all columns in your Customer Table. Output your result set to an excel file. File name is “Adding” followed by your last name. Include this file with your assignment uploads. (5 pts)

2. The Employee table has fields similar to the Customer table. The field definition (nvarchar, smallint) and field length is exactly the same, the only difference is the field name. Using your query from 1 g. revise the field names in your insert query so you can insert a new row with your name. (6 points)

Because you had the insert function on for Customers, you need to turn it OFF before you can turn it on for Employees. Paste this at the top of your insert query.

Use SalesOrdersModify;

SET IDENTITY\_INSERT Customers Off

SET IDENTITY\_INSERT Employees ON

GO

a. Before you run your insert query, how many rows are in the Employee table?

9

b. Execute your insert query and Paste it here.

Use SalesOrdersModify;

SET IDENTITY\_INSERT Customers OFF

SET IDENTITY\_INSERT Employees ON

GO

INSERT INTO Employees

(EmployeeID, EmpFirstName, EmpLastName, EmpStreetAddress,

EmpCity, EmpState, EmpZipCode, EmpAreaCode, EmpPhoneNumber)

VALUES ((select max(CustomerID) from Customers) + 1, 'Oliver', 'Conover', '15127 NE 24th, #383',

'New Port Richey', 'FL', '34655', 727, '555-2686');

c. Now that you inserted the row how many rows are in the Employee table?

10

3. The Categories table currently has 7 rows. You are going to write a query to add a categoryid of 8 (you can either type the value or use the max query to get the categoryid) and the CategoryDescription of Moped. (3 points)

Start your query with

Use SalesOrdersModify;

SET IDENTITY\_INSERT Employees Off

SET IDENTITY\_INSERT Categories ON

GO

Paste your query here.

se SalesOrdersModify;

SET IDENTITY\_INSERT Employees Off

SET IDENTITY\_INSERT Categories ON

GO

INSERT INTO Categories

(CategoryID, CategoryDescription)

VALUES((select max(CategoryID) from Categories) + 1, 'Moped')

4. For the Product table, we are only going to insert the required fields, which are the Not Null and the Foreign key. Insert a row that adds 1 to the max productnumber and add the foreign key categoryID set to 8 (the value we just added to the category table). (3 points)

Start your query with

Use SalesOrdersModify;

SET IDENTITY\_INSERT categories Off

SET IDENTITY\_INSERT Products ON

GO

Paste your query here.

Use SalesOrdersModify;

SET IDENTITY\_INSERT categories Off

SET IDENTITY\_INSERT Products ON

GO

INSERT INTO Products

(ProductNumber, CategoryID)

VALUES((select max(ProductNumber) from Products) + 1, 8)

5. Revise your query from questions 4 to insert a record that populates all the fields in the Product table. You make up the values for the fields. (3 points)

a. Paste your query here.

Use SalesOrdersModify;

SET IDENTITY\_INSERT categories Off

SET IDENTITY\_INSERT Products ON

GO

INSERT INTO Products

(ProductNumber, ProductName, ProductDescription, RetailPrice, QuantityOnHand, CategoryID)

VALUES((select max(ProductNumber) from Products) + 1, 'Gecko Food', 'Thats a lot of worms!', 3.99, 12, 8)

b. Insert another record, you can change the values if you want but it’s not required. Paste your query here.

Use SalesOrdersModify;

SET IDENTITY\_INSERT categories Off

SET IDENTITY\_INSERT Products ON

GO

INSERT INTO Products

(ProductNumber, ProductName, ProductDescription, RetailPrice, QuantityOnHand, CategoryID)

VALUES((select max(ProductNumber) from Products) + 1, 'Turtle Pellets', 'So many pellets... So many turtles...', 10.99, 85, 8)

b. Run a query that shows all rows and all columns in your Product Table. Output your result set to an excel file. File name is “Products” followed by your last name. Include this file with your assignment uploads. (5 pts)

6. We are going to archive data from the Orders and Order\_Details tables. (10 points)

a. Write a query to show the number of rows in the **Orders** table for the year 2012. Your result set should return a single row with your count. The count should be 598. Paste your query here.

select count(\*)

from Orders

where OrderDate < '2013-01-01'

b. We are archiving the Orders table for Orders for the year 2012 to the empty **Orders\_Archive** table which has already been created. Since we are archiving all fields it’s similar to the example on the top of page 553. Make sure you archive 598 rows. Paste your query here.

INSERT INTO Orders\_Archive

Select Orders.\*

From Orders

where Orders.OrderDate < '2013-01-01'

c. Write a query to show the number of rows in the **Order\_Details** table for the year 2012. You will link the Orders and Order\_Details to utilize the Orderdate on the Orders table. For this query, your result set should return a single row with your count. The count should be 2520. Paste your query here.

Select count(\*)

from Order\_Details od

join Orders o

on o.OrderNumber = od.OrderNumber

where o.OrderDate < '2013-01-01'

d. The empty **Order\_Details\_Archive** table has already been created. Since we are archiving all fields for Orderdate in 2012. Write a query similar to your query for 6 b. Make sure you archive 2520 rows. Paste your query here.

INSERT INTO Order\_Details\_Archive

(OrderNumber, ProductNumber, QuotedPrice, QuantityOrdered)

SELECT od.OrderNumber, od.ProductNumber, od.QuotedPrice, od.QuantityOrdered

from Order\_Details od

join Orders o

on o.OrderNumber = od.OrderNumber

where o.OrderDate < '2013-01-01'

e. Now run your query for 6 a. What is your row count?

598

It should still be 598 because the insert copies the rows to a new table but doesn’t remove the rows from the original table. We will do that in Chapter 17.