Chapter 6 Lab – Coding Subqueries

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Exercise Instructions

1. Type your name and the date into the space provided.
2. Use the SQL Server Management Studio to complete this lab.
3. Write T-SQL statements to query the tables contained in the IST272EagleCorp database and complete each of the exercises in this lab per the directions provided below.
4. Upload and submit before the due date.

1. Write a SELECT that returns CustomerID, CompanyName, CustLastName and Phone of each customer with no orders.

Paste below the **code** you wrote **and the run results** you obtained for exercise 1:

------------------------------------------------

2. Write a SELECT that returns the EmployeeID, LastName, FirstName, and JobTitle of any employee that has packed orders.

Paste below the **code** you wrote **and the run results** you obtained for exercise 2:

------------------------------------------------

3. Write a SELECT statement that answers this question: Which inventory parts weigh less than the average inventory part weight? List the PartDescription for each part that weighs less than the average.

Paste below the **code** you wrote **and the run results** you obtained for exercise 3:

------------------------------------------------

4. Write a SELECT that returns the EmployeeID, LastName, and JobTitle of any employee that has not packed orders.

Paste below the **code** you wrote **and the run results** you obtained for exercise 4:

------------------------------------------------

5. Write a SELECT statement that answers this question: Which inventory parts are heavier than two times the average inventory part weight? List the PartDescription for each part that weighs more than two times the average.

Paste below the **code** you wrote **and the run results** you obtained for exercise 5:

------------------------------------------------

6. Write a SELECT statement that returns the LastName and JobTitle of each employee that has a

SalaryWage greater than that of the highest paid employee with a jobTitle of “Programmer Analyst”.

Paste below the **code** you wrote **and the run results** you obtained for exercise 6:

------------------------------------------------

7. Write a SELECT statement that returns the LastName and JobTitle of each employee with a jobtitle other than ‘Engineer’ that has a SalaryWage greater than 65% of what the lowest paid employee with a jobTitle of engineer is paid.

Paste below the **code** you wrote **and the run results** you obtained for exercise 7:

------------------------------------------------

8. Write a SELECT statement that returns the OrderID and OrderDate along with the CustomerID, contact (customer last name, first) and Phone of the customer who placed the order if the order has not yet been at least partially shipped. An order is at least partially shipped if the status of any part ordered on the order is SHIPPED.

Paste below the **code** you wrote **and the run results** you obtained for exercise 8:

------------------------------------------------

9. Write a SELECT to list the first name, last name and BirthDate of all employees older than the employee with a job title of “Chief Sales Officer”.

Paste below the **code** you wrote **and the run results** you obtained for exercise 9:

------------------------------------------------

10. Write a query to list Customer order information that consists of the CompanyName, City, State, and OrderWeight.

Limit the rows selected as follows:

Only select rows with state values of: MO, OH, PA, VA, and WY

AND

Only select rows with an OrderWeight more than 243

AND

Only select rows where the CompanyName is not NULL

Sort the rows selected by CompanyName in ASC sequence

Hints:

* The OrderWeight is a calculated value.
* I suggest using a CTE to solve this problem.
* The first step I would take is to write a query that returns the OrderID, and the OrderWeight for each OrderID (refer back to chapter 5 summary queries – you want to use the sum function and a group by). When you get this step working you should see results like the following:

OrderID OrderWeight

---------- ---------------------------------------

1999000256 38.070

1999000393 3.378

1999000485 67.000

.

.

.

1999000334 26.752

1999000443 32.750

2000000605 10.128

Note: that data in the database can change. The above numbers reflect the running of the step 1 query on the day this assignment was created. It is possible that the data present when you work this assignment will result in different numbers. These query run results are shown to convey the nature of what you are doing. Not necessarily the specific numbers you will get.

* Once you have written a query that lists the OrderID and the OrderWeight turn it into a CTE (see chapter 6 – page 206 – 209)
* At this point write a select that uses the CTE, the Customer and the CustOrder table (see chapter 4 joins – if you do not recall how to join tables)
  + Hints: If you are rusty with joins open another query tab and get a join of the CustOrder table and Customer table working before work this step.
  + This step is like a three table join:
    - Customer JOIN CustOrder ON Customer.CustomerID = CustOrder.CustomerID
    - Join CTEName ON CTEName.OrderID = CustOrder.OrderID (note the CTEName is whatever you named your CTE, refer back to what you did for the bullet that calls for turning your step 1 query into a CTE)
* When you get this query completed you should see results like the following:

CompanyName City State OrderWeight

---------------------------------------- -------------------- ----- ---------------------

Apartment Referrals Jewett OH 403.029

Apartment Referrals Jewett OH 962.640

BMA Advertising Design Scranton PA 343.560

BMA Advertising Design Scranton PA 441.750

Cheesman Corporation Roanoke VA 333.260

Cottingham Plastics Lima OH 712.740

Needle Center New York City NY 644.100

Realty Specialties Syracuse NY 442.970

Realty Specialties Syracuse NY 706.400

Security Installation York PA 289.630

Security Installation York PA 539.040

Trailor Rentals Rome NY 359.840

(12 row(s) affected)

Note: that data in the database can change. The above numbers reflect the running of this query on the day this assignment was created. It is possible that the data present when you work this assignment will result in different results. These query run results are shown to convey the nature of results. Not necessarily the specific rows you will get.

Paste below the **code** you wrote **and the run results** you obtained for exercise 10:

------------------------------------------------