1. **In general, why would you even want to join two (or more) tables together? This is a good time to think about the nature of relational algebra.**

You would want to join two tables together when you want to view certain information from table one in relation to information from table 2. For example, if you have a table of orders and products and you want to find out which OrderID is associated with which ProductNames, you would need to join the tables.

1. **Describe in your own words the output from an inner join.**

An inner join applies two phases: Cartesian Product and filter. It essentially combines all rows from two tables and then filters based on the ON predicate.

1. **Describe in your own words the output from an outer join.**

An outer join applies three phases: Cartesian Product, filter, and Add Outer Rows. It returns the inner joined tables specified in the query as well as the table joined on that respective side identified in the join (left, right, full). If the outer row has no relation to the other table, a NULL is returned.

1. **Describe in your own words the output from a cross join.**

A cross join applies one phase: Cartesian Product. It takes one complete row from a table and matches it with a complete row from another table. The output is every possible combination of the two tables.

1. **A convenient mnemonic for remembering the various joins is “Ohio.” Why is this true?**

INSTRUCTED TO IGNORE THIS QUESTION

1. **Give an example of a composite join.**

FROM dbo.Table1 AS T1  
INNER JOIN dbo.Table2 AS T2  
ON T1.col1 = T2.col1  
AND T1.col2 = T2.col2

You have a foreign key defined on dbo.Table2, columns col1, col2, referencing dbo.Table1, columns col1, col2, and you need to write a query that joins the two based on this relationship.

**See Next Page for Questions 7-8**

1. **What is the difference between the following two queries? The business problem is “How many orders do we have from each customer?”**

================first query===============

SELECT C.custid, COUNT(\*) AS numorders

FROM Sales.Customers AS C

LEFT OUTER JOIN Sales.Orders AS O

ON C.custid = O.custid

GROUP BY C.custid;

================second query===============

SELECT C.custid, COUNT(O.orderid) AS numorders

FROM Sales.Customers AS C

LEFT OUTER JOIN Sales.Orders AS O

ON C.custid = O.custid

GROUP BY C.custid;

The COUNT\* function cannot detect if a row really represents an order whereas the COUNT(column) ignores outer rows with a NULL in the column. Query 1 = 832, Query 2 = 830

1. **What might be one reason the following query does not return the column custID in this query?**

SELECT C.custid, C.companyname, O.orderid, O.orderdate

FROM Sales.Customers AS C

LEFT OUTER JOIN Sales.Orders AS O

ON C.custid = O.custid

WHERE O.orderdate >= ’20160101’;

If there were no orders placed after 20160101 then there would be no data to return.