1) Based on a subset of rows from an underlying query, a **WINDOWS** function computes a **SCALAR** result.

2) Window functions are allowed only in the **SELECT** and **ORDER BY** clauses of an SQL query.

3) Pivoting data rotates it from **ROWS** to **COLUMNS**.

4) Unpivoting data takes each source row and turns it into potentially **MULTIPLE** result rows

5) Describe each of the set grouping subclauses (GROUPING SETS, CUBE, ROLLUP).

The GROUPING SETS subclause can be used to define multiple grouping sets in the same query. The CUBE subclause of the GROUP BY clause provides an abbreviated way to define multiple grouping sets. The ROLLUP subclause is an alternate way to define multiple grouping sets, but it doesn’t produce all the possible grouping sets.

6) Describe how you could use the GROUPING and GROUPING\_ID functions.

Both the GROUPING and GROUPING\_ID functions can only be used in conjunction with the SELECT <select> list, HAVING, or ORDER BY clauses when GROUP BY is specified. The GROUPING function accepts the name of a column and returns 0 if it is a member of the current grouping set and 1 otherwise. GROUPING\_ID further simplifies the process of associating result rows and grouping sets. You

provide the function with all elements that participate in any grouping set as inputs and the function

returns an integer bitmap in which each bit represents a different input element—the rightmost element represented by the rightmost bit.