SELF TEST ANSWERS

Limit the Rows Retrieved by a Query

1. ☑ **C.** The SELECT clause facilitates projection by specifying the list of columns to be projected from a table, whilst the WHERE clause facilitates selection by limiting the rows retrieved based on its conditions.
   ☒ **A**, **B**, and **D** are incorrect because the FROM clause specifies the source of the rows being projected and the ORDER BY clause is used for sorting the selected rows.

2. ☑ **B.** The IN operator efficiently tests whether the JOB_ID for a particular row is either SA_REP or MK_MAN, whilst the BETWEEN operator efficiently measures whether an employee's SALARY value falls within the required range.
   ☒ **A** and **C** exclude employees who earn a salary of $1000 or $4000, since these SALARY values are excluded by the inequality operators. C also selects JOB_ID values like SA_REP% and MK_MAN%, potentially selecting incorrect JOB_ID values. **D** is half right. The first half returns the rows with JOB_ID equal to SA_REP having SALARY values between $1000 and $4000. However, the second part (the OR clause), correctly tests for JOB_ID equal to MK_MAN but ignores the SALARY condition.

3. ☑ **C.** The character literals being compared to the JOB_ID column by the IN operator must be enclosed by single quotation marks.
   ☒ **A**, **B**, and **D** are syntactically correct. Notice that B does not require quotes around the numeric literals. Having them, however, does not cause an error.

4. ☑ **B.** The LIKE operator tests the DEPARTMENT_NAME column of each row for values that contain the characters "er". The percentage symbols before and after the character literal indicate that any characters enclosing the "er" literal are permissible.
   ☒ **A** and **C** are syntactically correct. **A** uses the IN operator, which is used to test set membership. **C** tests whether the alphabetic value of the DEPARTMENT_NAME column is between the letter "e" and the letter "r." Finally, **D** uses the word "contains," which cannot be used in this context.

5. ☑ **A** and **D.** The IS NULL operator correctly evaluates the COMMISSION_PCT column for NULL values. **D** uses the NOT operator to negate the already negative version of the IS NULL operator, IS NOT NULL. Two negatives return a positive, and therefore **A** and **D** are equivalent.
   ☒ **B** and **C** are incorrect since NULL values cannot be tested by the equality operator or the IN operator.

6. ☑ **A, C,** and **D.** Each of these conditions tests for SALARY values in the range of $2000 to $5000.
   ☒ **B** and **E** are incorrect. **B** excludes values like $2500 from its set, and **E** is illegal since it is missing the SALARY column name reference after the AND operator.

Sort the Rows Retrieved by a Query

7. ☑ **C.** The terms specified in an ORDER BY clause can include column names, positional sorting, numeric values, and expressions.
   ☒ **A, B,** and **D** are true.
8. ☑ **C.** Positional sorting is performed, and the third term in the SELECT list, COMMISSION_PCT, is sorted first in descending order, and any NULL COMMISSION_PCT values are listed last. The second term in the SELECT list, SALARY, is sorted next in ascending order.
   ☒ **A, B,** and **D** are incorrect. **A** does not specify what to do with NULL COMMISSION_PCT values, and the default behavior during a descending sort is to list NULLS FIRST. **B** applies the NULLS LAST modifier to the SALARY column instead of the COMMISSION_PCT column, and **D** ignores NULLS completely.

Ampersand Substitution

9. ☑ **B.** A session-persistent substitution variable may be referenced using an ampersand symbol from within any SQL statement executed in that session.
   ☒ **A, C,** and **D** are incorrect. **A** and **D** attempt to reference the substitution variable using a colon prefix to its name and the variable name on its own. These are invalid references to substitution variables in SQL. **C** references a variable called TAX and not the variable TAX_RATE.
10. ☑ **D.** The first time this statement is executed, two single ampersand substitution variables are encountered before the third double ampersand substitution variable. If the first reference on line one of the query contained a double ampersand substitution, you would only be prompted to input a value once.
    ☒ **A, B,** and **C** are incorrect since you are prompted thrice to input a value for the JOB substitution variable. In subsequent executions of this statement in the same session you will not be prompted to input a value for this variable.