**Question: 71. (C)**

Which clause would you use in a SELECT statement to limit the display to those employees whose salary is greater then 5000?

1. ORDER BY SALARY > 5000.
2. GROUP BY SALARY > 5000
3. HAVING SALARY > 5000
4. WHERE SALARY > 5000

**Question: 72. (C)**

You need to calculate the total of all salaries in the accounting department. Which group function should you use?

1. MAX
2. MIN
3. SUM
4. COUNT
5. TOTAL
6. LARGEST

**Question: 73. (C)**

Which SELECT statement will the result ‘elloworld’ from the string ‘HelloWorld’?

1. SELECT SUBSTR( ‘HelloWorld’,1) FROM dual;
2. SELECT INITCAP(TRIM (‘HelloWorld’, 1,1)) FROM dual;
3. SELECT LOWER(SUBSTR(‘HellowWorld’, 1, 1) FROM dual;
4. SELECT LOWER(SUBSTR(‘HelloWorld’, 2, 1) FROM dual;
5. SELECT LOWER(TRIM (‘H’ FROM ‘HelloWorld’)) FROM dual;

**Question: 74. (C)**

Evaluate this SQL statement:

SELECT e.employee\_id, (.15\* e.salary) + (.5 \* e.commission\_pct)

+ (s.sales amount \* (.35 \* e.bonus)) AS CALC\_VALUE

FROM employees e, sales s

WHERE e.employee\_id = s.emp\_id;

What will happen if you remove all the parentheses from the calculation?

1. The value displayed in the CALC\_VALUE column will be lower.
2. The value displayed in the CALC\_VALUE column will be higher.
3. There will be no difference in the value displayed in the CALC\_VALUE column.
4. An error will be reported.

**Question: 75. (C)**

Which script displays '01-JAN-02' when the ENROLL\_DATE value is '01-JUL-01'?

1. SELECT ROUND(enroll\_date, 'DAY') FROM student;
2. SELECT ROUND(enroll\_date, 'YEAR')

FROM student;

1. SELECT ROUND(enroll\_date, 'MONTH') FROM student;
2. SELECT ROUND(TO\_CHAR(enroll\_date, 'YYYY')) FROM student;

**Explanation:**

ROUND function will round a value to the next higher value. In the above given scenario the ENROLL\_DATE will be round to the next year as the enroll\_date is higher than 30-jun-01 so it will be rounded to the next year by specifying the YEAR option with the ROUND function.

**Question: 76. (C)**

Which three functions can be used to manipulate character, number, or date column values?

(Choose three.)

1. CONCAT
2. ROUND
3. TRUNC
4. RPAD
5. INSTR

**Question: 77. (C)**

A new standard has been adopted in your department that all codes that contain only 3 characters must have a dash (-) and two character values appended to them. Which function can be used in your query to restrict the data displayed to only those codes containing 3 characters?

1. REPLACE
2. SUBSTR
3. LENGTH
4. RPAD

**Question: 78. (C)**

Which statement concerning SQL functions is true?

1. Character functions can return character or number values.
2. Conversion functions convert a column definition from one data type to another data type.
3. Single-row functions can only be used in SELECT and WHERE clauses.
4. All date functions return DATE data type values.

**Question: 79. (C)**

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

The registrar has asked for a report on the average grade point average (GPA) for students enrolled during semesters that end in the year 2000. Which statement accomplish this?

1. SELECT AVERAGE(gpa)

FROM student\_grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < 31- DEC-2000’;

1. SELECT COUNT(gpa)

FROM student grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < ’31-

DEC-2000’;

1. SELECT MID(gpa)

FROM student grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < ’31- DEC-2000’;.

1. SELECT AVG(gpa) FROM student\_grades

WHERE semester\_end BETWEEN ’01-JAN-2000’ and ’31.DEC.2000’;

1. SELECT SUM(gpa) FROM student grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < ’31- DEC-2000’;

1. SELECT MEDIAN(gpa)

FROM student\_grades

WHERE semester end > ’01-JAN-2000’ and semester end < ’31-

DEC-2000’;

**Question: 80. (C)**

Evaluate the SQL statement:

1. SELECT a.emp\_name, a.sal, a.dept\_id, b.maxsal

1. FROM employees a,

1. (SELECT dept\_id, MAX(sal) maxsal

4. FROM employees

1. GROUP BY dept\_id) b

1. WHERE a.dept\_id = b.dept\_id

1. AND a.sal < b.maxsal;

What is the result of the statement?

1. The statement produces an error at line 1.
2. The statement produces an error at line 3.
3. The statement produces an error at line 6.
4. The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all departments that pay less salary then the maximum salary paid in the company.
5. The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all employees who earn less than the maximum salary in their department.

**Question: 81. (C)**

The EMP table has these columns:

ENAME VARCHAR2(35)

SALARY NUMBER(8,2)

HIRE\_DATE DATE

Management wants a list of names of employees who have been with the company for more than five years. Which SQL statement displays the required results?

1. SELECT ENAME

FROM EMP

WHERE SYSDATE-HIRE\_DATE > 5;

1. SELECT ENAME

FROM EMP

WHERE HIRE\_DATE-SYSDATE > 5;

1. SELECT ENAME

FROM EMP

WHERE (SYSDATE-HIRE\_DATE)/365 > 5;

1. SELECT ENAME

FROM EMP

WHERE (SYSDATE-HIRE\_DATE)\* 365 > 5;

**Question: 82. (C)**

Which SELECT statement will the result ‘ello world’ from the string ‘Hello World’?

1. SELECT SUBSTR( ‘Hello World’,1) FROM dual;
2. SELECT INITCAP(TRIM (‘Hello World’, 1,1)) FROM dual;
3. SELECT LOWER(SUBSTR(‘Hello World’, 1, 1) FROM dual;
4. SELECT LOWER(SUBSTR(‘Hello World’, 2, 1) FROM dual;
5. SELECT LOWER(TRIM (‘H’ FROM ‘Hello World’)) FROM dual;

**Incorrect Answers**

**Question: 83. (C)**

Evaluate the SQL statement:

SELECT ROUND(45.953, -1), TRUNC(45.936, 2)

FROM dual;

Which values are displayed?

1. 46 and 45
2. 46 and 45.93
3. 50 and 45.93
4. 50 and 45.9
5. 45 and 45.93
6. 45.95 and 45.93

**Question: 84. (C)**

The CUSTOMERS table has these columns:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CUSTOMER\_ID |  | NUMBER (4) | | NOT NULL |
| CUSTOMER\_NAME |  | VARCHAR2 (100) | | NOT NULL |
| STREET\_ADDRESS |  | VARCHAR2 (150) | |  |
| CITY\_ADDRESS | | VARHCAR2 (50) |
| STATE\_ADDRESS | | VARCHAR2 (50) |
| PROVINCE\_ADDRESS | | VARCHAR2 (50) |
| COUNTRY\_ADDRESS | | VARCHAR2 (50) |
| POSTAL\_CODE | | VARCHAR2 (12) |
| CUSTOMER\_PHONE | | VARCHAR2 (20) |

The CUSTOMER\_ID column is the primary key for the table. You need to determine how dispersed your customer base is. Which expression finds the number of different countries represented in the CUSTOMERS table?

1. COUNT(UPPER(country\_address))
2. COUNT(DIFF(UPPER(country\_address)))
3. COUNT(UNIQUE(UPPER(country\_address)))
4. COUNT DISTINTC UPPER(country\_address)
5. COUNT(DISTINTC (UPPER(country\_address)))

**Question: 86. (C)**

You are using single row function in a SELECT statement which function can best be catergorized as similar in function to an IF-THEN-ELSE statement?

1. SQRT
2. DECODE
3. NEW\_TIME D. ROWIDTOCHAR.

**Question: 87. (C)**

Which of the following functions are available in SQL? (Choose four)

1. INSTR.
2. NVL2.
3. TRUNCATE.
4. DECODE.
5. TRIM.
6. NVL.
7. NULLIF.

**Question: 88.(D)**

Examine the structure of the EMPLOYEES and DEPARTMENTS tables:

EMPLOYEES

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Remarks |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| EMP\_NAME | VARCHAR2 (30) |  |
| JOB\_ID | VARCHAR2 (20) |  |
| SALARY | NUMBER |  |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID COLUMN |
| DEPARTMENT ID | NUMBER | Foreign key to DEPARTMENT ID column of the DEPARTMENTS table |

DEPARTMENTS

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Remarks |
| DEPARTMENT\_ID | NUMBER | NOT NULL, Primary Key |
| DEPARTMENT\_NAME | VARCHAR2(30) |  |
| MGR\_ID | NUMBER | References MGR\_ID column of the EMPLOYEES table |

Evaluate this SQL statement:

SELECT employee\_id, e.department\_id, department\_name, salary

FROM employees e, departments d

WHERE e.department\_id = d.department\_id;

Which SQL statement is equivalent to the above SQL statement?

1. SELECT employee\_id, department\_id, department\_name, salary.

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM departments);

1. SELECT employee\_id, department\_id, department\_name, salary FROM employees

NATURAL JOIN departments;

1. SELECT employee\_id, d.department\_id, department\_name, salary FROM employees e JOIN departments d

ON e.department\_id = d.department\_id;

1. SELECT employee\_id, department\_id, department\_name,

Salary

FROM employees

JOIN departments

USING (e.department\_id, d.department\_id);

**Question: 89. (D)**

Examine the data in the EMPLOYEES and DEPARTMENTS tables.

**EMPLOYEES**

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | DEPARTMENT\_ID | SALARY |
| Getz | 10 | 3000 |
| Davis | 20 | 1500 |
| King | 20 | 2200 |
| Davis | 30 | 5000 |
| Kochhar |  | 5000 |

**DEPARTMENTS**

|  |  |
| --- | --- |
| DEPARTMENT\_ID | DEPARTMENT\_NAME |
| 10 | Sales |
| 20 | Marketing |
| 30 | Accounts |
| 40 | Administration |

You want to retrieve all employees, whether or not they have matching departments in the departments table. Which query would you use?

1. SELECT last\_name, department\_name

FROM employees , departments(+);

1. SELECT last\_name, department\_name

FROM employees JOIN departments (+);

1. SELECT last\_name, department\_name

FROM employees(+) e JOIN departments d

ON (e.department\_id = d.department\_id);

1. SELECT last\_name, department\_name

FROM employees e

RIGHT OUTER JOIN departments d ON (e.department\_id = d.department\_id);

1. SELECT last\_name, department\_name

FROM employees(+) , departments ON (e.department\_id = d.department\_id);

1. SELECT last\_name, department\_name

FROM employees e LEFT OUTER

JOIN departments d ON (e.department\_id = d.department\_id);

**Question: 90. (D)**

In which case would you use a FULL OUTER JOIN?

1. Both tables have NULL values.
2. You want all unmatched data from one table.
3. You want all matched data from both tables.
4. You want all unmatched data from both tables.
5. One of the tables has more data than the other.
6. You want all matched and unmatched data from only one table.

**Question: 91. (D)**

Which three are true regarding the use of outer joins? (Choose three.)

1. You cannot use IN operator in a condition that involves an outerjoin.
2. You use (+) on both sides of the WHERE condition to perform an outerjoin.
3. You use (\*) on both sides of the WHERE condition to perform an outerjoin.
4. You use an outerjoin to see only the rows that do not meet the join condition.
5. In the WHERE condition, you use (+) following the name of the column in the table without matching rows, to perform an outerjoin.
6. You cannot link a condition that is involved in an outerjoin to another condition by using the OR operator.

**Question: 92. (D)**

What is true about joining tables through an equijoin?

1. You can join a maximum of two tables through an equijoin.
2. You can join a maximum of two columns through an equijoin.
3. You specify an equijoin condition in the SELECT or FROM clauses of a SELECT statement.
4. To join two tables through an equijoin, the columns in the join condition must be primary key and foreign key columns.
5. You can join n tables (all having single column primary keys) in a SQL statement by specifying a minimum of n-1 join conditions.

**Question: 96. (D)**

Which two operators can be used in an outer join condition? (Choose two.)

1. **=**
2. OR
3. IN
4. AND

**Question: 97. (D)**

Examine the data from the CLASS and INSTRUCTOR tables.

**CLASS Table**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS\_ID | CLASS\_NAME | HOURS\_CREDIT | INSTRUCTOR\_ID |
| 1 | Introduction to Accounting | 3 | 4 |
| 2 | Computer Basics | 3 | 1 |
| 3 | Tax Accounting Principles | 3 | 4 |
| 4 | American History | 3 | 2 |
| 5 | Basic Engineering |  |  |

**INSTRUCTOR Table**

|  |  |  |
| --- | --- | --- |
| INSTRUCTOR\_ID | LAST\_NAME | FIRST\_NAME |
| 1 | Chao | Ling |
| 2 | Vanderbilt | Herbert |
| 3 | Wigley | Martha |
| 4 | Page | Albert |

You want to retrieve a list of all classes, including those with no instructor currently assigned. Which SELECT statement should you use?

1. SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName

FROM instructor i, class c

WHERE i.instructor\_id = c.instructor\_id (+);

1. SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName

FROM instructor I JOIN class c

WHERE i.instructor\_id (+) = c.instructor\_id;

1. SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName

FROM instructor i RIGHT OUTER JOIN class c ON (i.instructor\_id = c.instructor\_id);

1. SELECT class\_name, last\_name || ', ' || first\_name InstructorName FROM instructor RIGHT OUTER JOIN class;
2. SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName

FROM instructor i RIGHT OUTER JOIN class c

USING (instructor\_id);

**Question: 98. (D)**

To produce a meaningful result set without any cartesian products, what is the minimum number of conditions that should appear in the WHERE clause of a four-table join?

1. 8
2. 2 C) 3
3. 4
4. 5
5. There is no such criteria

**Question: 102. (E)**

Evaluate this SQL statement:

SELECT product\_id, product\_name, price

FROM product WHERE supplier\_id IN

(SELECT supplier\_id

FROM product WHERE price > 120

OR qty\_in\_stock > 100);

Which values will be displayed?

1. The PRODUCT\_ID, PRODUCT\_NAME, and PRICE of products that are priced greater than

$120.00 and have a QTY\_IN\_STOCK value greater than 100.

1. The PRODUCT\_ID , PRODUCT\_NAME, and PRICE of products that are priced greater than $120.00 or that have a QTY\_IN\_STOCK value greater than 100.
2. The PRODUCT\_ID, PRODUCT\_NAME, and PRICE of products that are priced greater than $120.00 or that have a QTY\_IN\_STOCK value greater than 100 and have a supplier.
3. The PRODUCT\_ID, PRODUCT\_NAME, and PRICE of products supplied by a supplier with products that are priced greater than $120.00 or with products that have a QTY\_IN\_STOCK value greater than 100.

**Question: 103. (E)**

Which statement regarding subqueries is true?

1. A subquery CANNOT reference a table that is not included in the outer query's FROM clause.
2. Subqueries can be nested up to 5 levels.
3. A subquery must be placed on the right side of the comparison operator.
4. Subqueries can return multiple columns.

**Question: 106. (E)**

A subquery can be used to \_\_\_\_\_\_\_\_\_.

1. Create groups of data.
2. Sort data in a specific order
3. Convert data to a different format
4. Retrieve data based on an unknown condition

**Question: 107. (E)**

Which three statements about subqueries are true? (Choose three)

1. A single row subquery can retrieve only one column and one row.
2. A single row subquery can retrieve only one row but many columns.
3. A multiple row subquery can retrieve multiple rows and multiple columns.
4. A multiple row subquery can be compared by using the “>” operator.
5. A single row subquery can use the IN operator.
6. A multiple row subquery can use the “=” operator.

**Question: 108. (E)**

You define a multiple-row subquery in the WHERE clause of an SQL query with a comparison operator "=".

What happens when the main query is executed?

1. The main query executes with the first value returned by the subquery.
2. The main query executes with the last value returned by the subquery.
3. The main query executes with all the values returned by the subquery.
4. The main query fails because the multiple-row subquery cannot be used with the comparison operator
5. You cannot define a multiple-row subquery in the WHERE clause of a SQL query.

**Question: 109. (E)**

Which operator can be used with a multiple-row subquery?

1. =
2. LIKE
3. BETWEEN
4. NOT IN
5. IS
6. <>

**Question: 110. (E)**

Which two statements about subqueries are true? (Choose two.)

1. A single row subquery can retrieve data from only one table.
2. A SQL query statement cannot display data from table B that is referred to in its subquery, unless table B is included in the main query's FROM clause.
3. A SQL query statement can display data from table B that is referred to in its subquery, without including table B in its own FROM clause.

D A single row subquery can retrieve data from more than one table.

1. A single row subquery cannot be used in a condition where the LIKE operator is used for comparison.
2. A multiple-row subquery cannot be used in a condition where the LIKE operator is used for comparison.