**Question: 40.(C)**

Examine the description of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMP\_ID | NUMBER(4) | NOT NULL |
| LAST\_NAME | VARCHAR2(30) | NOT NULL |
| FIRST\_NAME | VARCHAR2(30) |  |
| DEPT\_ID | NUMBER(2) |  |
| JOB\_CAT | VARCHARD2(30) |  |
| SALARY | NUMBER(8,2) |  |

Which statement shows the maximum salary paid in each job category of each department?

1. SELECT dept\_id, job\_cat, MAX(salary)

FROM employees

WHERE salary > MAX(salary);

1. SELECT dept\_id, job\_cat, MAX(salary)

FROM employees

GROUP BY dept\_id, job\_cat;

1. SELECT dept\_id, job\_cat, MAX(salary)

FROM employees;

1. SELECT dept\_id, job\_cat, MAX(salary)

FROM employees

GROUP BY dept\_id;

1. SELECT dept\_id, job\_cat, MAX(salary)

FROM employees

GROUP BY dept\_id, job\_cat, salary;

**Question: 41. (C)**

Management has asked you to calculate the value 12\*salary\* comossion\_pct for all the employees in the EMP table. The EMP table contains these columns:

|  |  |  |
| --- | --- | --- |
| LAST NAME | VARCNAR2(35) | NOT NULL |
| SALARY | NUMBER(9,2) | NOT NULL |
| COMMISION\_PCT | NUMBER(4,2) |  |

Which statement ensures that a value is displayed in the calculated columns for all employees?

1. SELECT last\_name, 12\*salary\*commison\_pct

FROM emp;

1. SELECT last\_name, 12\*salary\* (commission\_pct,0)

FROM emp;

1. SELECT last\_name, 12\*salary\*(nvl(commission\_pct,0))

FROM emp;

1. SELECT last\_name, 12\*salary\*(decode(commission\_pct,0))

FROM emp;

**Question: 42. (C)**

Examine the description of the STUDENTS table:

|  |  |
| --- | --- |
| STD\_ID | NUMBER(4) |
| COURSE\_ID | VARCHARD2(10) |
| START\_DATE | DATE |
| END\_DATE | DATE. |

Which two aggregate functions are valid on the START\_DATE column? (Choose two)

1. SUM(start\_date)
2. AVG(start\_date)
3. COUNT(start\_date)
4. AVG(start\_date, end\_date)
5. MIN(start\_date)
6. MAXIMUM(start\_date)

**Question: 43. (C)**

The EMPLOYEE tables has these columns:

|  |  |
| --- | --- |
| LAST\_NAME | VARCNAR2(35) |
| SALARY | NUMBER(8,2) |
| COMMISSION\_PCT | NUMBER(5,2) |

You want to display the name and annual salary multiplied by the commission\_pct for all employees. For records that have a NULL commission\_pct, a zero must be displayed against the calculated column. Which SQL statement displays the desired results?

1. SELECT last\_name, (salary \* 12) \* commission\_pct

FROM EMPLOYEES;

1. SELECT last\_name, (salary \* 12) \* IFNULL(commission\_pct,

0)

FROM EMPLOYEES;

1. SELECT last\_name, (salary \* 12) \* NVL2(commission\_pct, 0)

FROM EMPLOYEES;

1. SELECT last\_name, (salary \* 12) \* NVL(commission\_pct, 0)

FROM EMPLOYEES;

**Question: 44. (C)**

You would like to display the system date in the format "Monday, 01 June, 2001". Which SELECT statement should you use?

1. SELECT TO\_DATE(SYSDATE, 'FMDAY, DD Month, YYYY')

FROM dual;

1. SELECT TO\_CHAR(SYSDATE, 'FMDD, DY Month, 'YYY')

FROM dual;

1. SELECT TO\_CHAR(SYSDATE, 'FMDay, DD Month, YYYY')

FROM dual;

1. SELECT TO\_CHAR(SYSDATE, 'FMDY, DDD Month, YYYY')

FROM dual;

1. SELECT TO\_DATE(SYSDATE, 'FMDY, DDD Month, YYYY')

FROM dual;

**Question: 45. (C)**

Evaluate the SQL statement:

SELECT ROUND(TRUNC(MOD(1600,10),-1),2)

FROM dual;

What will be displayed?

1. 0
2. 1
3. 0.00
4. An error statement

**Question: 46. (C)**

Examine the description of the MARKS table:

|  |  |
| --- | --- |
| STD\_ID | NUMBER(4) |
| STUDENT\_NAME | VARCHAR2(30) |
| SUBJ1 | NUMBER(3) |
| SUBJ2 | NUMBER(3) |

SUBJ1 and SUBJ2 indicate the marks obtained by a student in two subjects. Examine this SELECT statement based on the MARKS table:

SELECT subj1+subj2 total\_marks, std\_id

FROM marks

WHERE subj1 > AVG(subj1) AND subj2 > AVG(subj2)

ORDER BY total marks;

What is the result of the SELECT statement?

A. The statement executes successfully and returns the student ID and sum of all marks for each student who obtained more than the average mark in each subject. B. The statement returns an error at the SELECT clause.

1. The statement returns an error at the WHERE clause.
2. The statement returns an error at the ORDER BY clause.

**Question: 47. (C)**

Which three SELECT statements displays 2000 in the format “$2,000.00”? (Choose three)

1. SELECT TO\_CHAR (2000, ‘$#,###.##’)

FROM dual;

1. SELECT TO\_CHAR (2000, ‘$0,000.00’)

FROM dual;

1. SELECT TO\_CHAR (2000, ‘$9,999.00’)

FROM dual;

1. SELECT TO\_CHAR (2000, ‘$9,999.99’)

FROM dual;

1. SELECT TO\_CHAR (2000, ‘$2,000.00’)

FROM dual;

1. SELECT TO\_CHAR (2000, ‘$N,NNN.NN’)

FROM dual;

**Question: 48. (C)**

Examine the description of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMP\_ID | NUMBER(4) | NOT NULL |
| LAST\_NAME | VARCHAR2(30) | NOT NULL |
| FIRST\_NAME | VARCHAR2(30). |  |
| DEPT\_ID | NUMBER(2) |  |
| JOB\_CAT | VARCHAR2(30) |  |
| SALARY | NUMBER(8,2) |  |
|  |  |  |

Which statement shows the department ID, minimum salary, and maximum salary paid in that department, only of the minimum salary is less then 5000 and the maximum salary is more than

15000?

1. SELECT dept\_id, MIN(salary(, MAX(salary)

FROM employees

WHERE MIN(salary) < 5000 AND MAX(salary) > 15000;

1. SELECT dept\_id, MIN(salary), MAX(salary)

FROM employees

WHERE MIN(salary) < 5000 AND MAX(salary) > 15000

GROUP BY dept\_id;

1. SELECT dept\_id, MIN(salary), MAX(salary)

FROM employees

HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;

1. SELECT dept\_id, MIN(salary), MAX(salary)

FROM employees

GROUP BY dept\_id

HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;

1. SELECT dept\_id, MIN(salary), MAX(salary)

FROM employees GROUP BY dept\_id, salary

HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;

**Question: 49. (C)**

Which two are true about aggregate functions? (Choose two.)

1. You can use aggregate functions in any clause of a SELECT statement.
2. You can use aggregate functions only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.
3. You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns.
4. You can pass column names, expressions, constants, or functions as parameters to an aggregate function.
5. You can use aggregate functions on a table, only by grouping the whole table as one single group.
6. You cannot group the rows of a table by more than one column while using aggregate functions.

**Question: 50. (C)**

Which four statements correctly describe functions that are available in SQL? (Choose four)

1. INSTR returns the numeric position of a named character.
2. NVL2 returns the first non-null expression in the expression list.
3. TRUNCATE rounds the column, expression, or value to n decimal places.
4. DECODE translates an expression after comparing it to each search value.
5. TRIM trims the heading of trailing characters (or both) from a character string.
6. NVL compares two expressions and returns null if they are equal, or the first expression of they are not equal.
7. NULLIF compares two expressions and returns null if they are equal, or the first expression if they are not equal.

**Question: 52. (C)**

Which clause should you use to exclude group results?

1. WHERE
2. HAVING
3. RESTRICT
4. GROUP BY
5. ORDER BY

**Question: 53. (C)**

In a SELECT statement that includes a WHERE clause, where is the GROUP BY clause placed in the SELECT statement?

1. Immediately after the SELECT clause
2. Before the WHERE clause
3. Before the FROM clause
4. After the ORDER BY clause
5. After the WHERE clause

**Question: 54. (C)**

Which two are character manipulation functions? (Choose two.)

1. TRIM
2. REPLACE
3. TRUNC
4. TO\_DATE
5. MOD
6. CASE

**Question: 55. (C)**

The EMPLOYEES table contains these columns:

|  |  |
| --- | --- |
| LAST\_NAME | VARCHAR2 (25) |
| SALARY | NUMBER (6,2) |
| COMMISSION\_PCT | NUMBER (6) |

You need to write a query that will produce these results:

1. Display the salary multiplied by the commission\_pct.
2. Exclude employees with a zero commission\_pct.
3. Display a zero for employees with a null commission value.

Evaluate the SQL statement:

SELECT LAST\_NAME, SALARY\*COMMISSION\_PCT

FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NOT NULL;

What does the statement provide?

1. All of the desired results
2. Two of the desired results
3. One of the desired results
4. An error statement

**Question: 58. (C)**

The PRODUCT table contains these columns:

PRODUCT\_ID NUMBER(9) PRODUCT\_NAME VARCHAR2(25)

COST NUMBER(5,2)

LIST\_PRICE NUMBER(5,2)

SUPPLIER\_ID NUMBER(9)

You need to display product names, costs, supplier ids, and average list prices for all the products that cost more than the average cost of products provided by the same supplier. Which SELECT statement will achieve these results?

1. SELECT product\_name, cost, supplier\_id, AVG(list\_price)

FROM product p, product a

WHERE p.supplier\_id = a.supplier\_id GROUP BY product\_name, cost, supplier\_id;

1. SELECT product\_name, cost, p.supplier\_id, AVG(list\_price)

FROM product p, (SELECT supplier\_id, AVG(cost) avg\_cost

FROM product

GROUP BY supplier\_id) a

WHERE p.cost > a.avg\_cost

GROUP BY product\_name, cost, p.supplier\_id;

1. SELECT product\_name, cost, supplier\_id, AVG(list\_price)

FROM product

WHERE supplier\_id IN (SELECT supplier\_id, AVG(cost) avg\_cost

FROM product

GROUP BY supplier\_id)

GROUP BY product\_name, cost, supplier\_id;

1. SELECT product\_name, cost, p.supplier\_id, AVG(list\_price)

FROM product p, (SELECT supplier\_id, AVG(cost) avg\_cost

FROM product GROUP BY supplier\_id) a

WHERE p.supplier\_id = a.supplier\_id

AND p.cost > a.avg\_cost

GROUP BY product\_name, cost, p.supplier\_id;

**Question: 59. (C)**

Which SELECT statement should you use to extract the year from the system date and display it in the format "1998"?

1. SELECT TO\_CHAR(SYSDATE,'yyyy')

FROM dual;

1. SELECT TO\_DATE(SYSDATE,'yyyy') FROM dual;.
2. SELECT DECODE(SUBSTR(SYSDATE, 8), 'YYYY')

FROM dual;

1. SELECT DECODE(SUBSTR(SYSDATE, 8), 'year')

FROM dual;

1. SELECT TO\_CHAR(SUBSTR(SYSDATE, 8,2),'yyyy')

FROM dual;

**Question: 60. (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 | VARCHAR2(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. Which two statements find the number of customers? (Choose two.)

1. SELECT TOTAL(\*)

FROM customers;

1. SELECT COUNT(\*)

FROM customers;

1. SELECT TOTAL(customer\_id)

FROM customers;

1. SELECT COUNT(customer\_id)

FROM customers;

1. SELECT COUNT(customers)

FROM customers;

1. SELECT TOTAL(customer\_name)

FROM customers;

**Question: 61. (C)**

Examine the structures of the EMPLOYEES and TAX tables.

**EMPLOYEES**

|  |  |  |
| --- | --- | --- |
| 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 |

**TAX**

|  |  |  |
| --- | --- | --- |
| MIN\_SALARY | NUMBER |  |
| MAX\_SALARY | NUMBER |  |
| TAX\_PERCENT | NUMBER | Percentage tax for given salary range |

You need to find the percentage tax applicable for each employee. Which SQL statement would you use?

A. SELECT employee\_id, salary, tax\_percent

FROM employees e, tax t

WHERE e.salary BETWEEN t.min\_salary AND t.max\_salary; B. SELECT employee\_id, salary, tax\_percent FROM employees e, tax t.

WHERE e.salary > t.min\_salary, tax\_percent

1. SELECT employee\_id, salary, tax\_percent

FROM employees e, tax t

WHERE MIN(e.salary) = t.min\_salary

AND MAX(e.salary) = t.max\_salary

1. You cannot find the information because there is no common column between the two tables.

**Question: 62. (C)**

Examine the data in the WORKORDER table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WO\_ID | CUST\_ID | REQUIRED\_DT | COMPLE\_DT | AMOUNT |
| 1 | 1 | 04-DEC-2001 | 02-DEC-01 | 520.32 |
| 2 | 1 | 02-JAN-2002 |  |  |
| 3 | 2 | 17-JAN-2002 |  |  |
| 4 | 2 | 20-JAN-2002 | 05-JAN-2002 | 274.11 |
| 6 | 3 | 14-JAN-2001 | 13-JAN-2002 | 400.00 |
| 7 | 3 | 04-FEB-2002 |  |  |
| 8 | 4 | 01-FEB-2002 |  |  |
| 9 | 5 |  | 14-JAN-2002 |  |

The WORKORDER table contains these columns:

WO\_ID NUMBER PK

CUST\_ID NUMBER

REQUIRED\_DT DATE

COMPL\_DT DATE

AMOUNT NUMBER(7,2)

Which statement regarding the use of aggregate functions on the WORKORDER table is true?

1. Using the SUM aggregate function with the AMOUNT column is allowed in any portion of a SELECT statement.
2. Using the AVG aggregate function with any column in the table is allowed.
3. Using the SUM aggregate function on the AMOUNT column will result in erroneous results because the column contains null values.
4. Grouping on the REQUIRED\_DT and COMPL\_DT columns is NOT allowed.
5. Using the AVG aggregate function on the AMOUNT column ignores null values.
6. Using the MIN aggregate function on the COMPL\_DT column will return a null value.

**Question: 63.**

The INVENTORY table contains these columns:

ID\_NUMBER NUMBER PK

CATEGORY VARCHAR2(10)

LOCATION NUMBER

DESCRIPTION VARCHAR2(30)

PRICE NUMBER(7,2)

QUANTITY NUMBER

You want to return the total of the extended amounts for each item category and location, including only those inventory items that have a price greater than $100.00. The extended amount of each item equals the quantity multiplied by the price. Which SQL statement will return the desired result?

1. SELECT category, SUM(price \* quantity) TOTAL, location

FROM inventory

WHERE price > 100.00

GROUP BY category;

1. SELECT category, location, SUM(price)

FROM inventory

WHERE price > 100.00 GROUP BY category, location;

1. SELECT category, SUM(price \* quantity) TOTAL, location

FROM inventory

WHERE price > 100.00;

1. SELECT category, SUM(price \* quantity) TOTAL, location

FROM inventory WHERE price > 100.00

GROUP BY category, location;

**Question: 64. (C)**

The EVENT table contains these columns:

EVENT\_ID NUMBER

EVENT\_NAME VARCHAR2(30)

EVENT\_DESC VARCHAR2(100)

EVENT\_TYPE NUMBER

LOCATION\_ID NUMBER

You have been asked to provide a report of the number of different event types at each location. Which SELECT statement will produce the desired result?

1. SELECT UNIQUE(location\_id), COUNT(event\_type)

FROM event

GROUP BY location\_id;

1. SELECT COUNT(\*), DISTINCT(location\_id) FROM event;
2. SELECT DISTINCT (event\_type)

FROM event

GROUP BY location\_id;

1. SELECT location\_id, COUNT(DISTINCT event\_type)

FROM event

GROUP BY location\_id;

1. SELECT location\_id, MAX(DISTINCT event\_type)

FROM event

GROUP BY location\_id;

**Question: 65. (C)**

Which two statements about the evaluation of clauses in a SELECT statement are true? (Choose two.)

1. The Oracle Server will evaluate a HAVING clause before a WHERE clause.
2. The Oracle Server will evaluate a WHERE clause before a GROUP BY clause.
3. The Oracle Server will evaluate a GROUP BY clause before a HAVING clause.
4. The Oracle Server will evaluate an ORDER BY clause before a WHERE clause.
5. The Oracle Server will evaluate an ORDER BY clause before a HAVING clause.

**Question: 67. (C)**

Which two tasks can you perform by using the TO\_CHAR function? (Choose two)

1. Convert 10 to ‘TEN’
2. Convert ‘10’ to 10
3. Convert ‘10’ to ‘10’
4. Convert ‘TEN’ to 10
5. Convert a date to a character expression
6. Convert a character expression to a date

**Question: 69. (C)**

Which SQL statement generates the alias Annual Salary for the calculated column SALARY\*12?

1. SELECT ename, salary\*12 ‘Annual Salary’

FROM employees;

1. SELECT ename, salary\*12 “Annual Salary”

FROM employees;

1. SELECT ename, salary\*12 AS Annual Salary

FROM employees;

1. SELECT ename, salary\*12 AS INITCAP(“ANNUAL SALARY”)

FROM employees