

**Question: 1. (A)**

Which SELECT statement should you use if you want to display unique combinations of the POSITION and MANAGER values from the EMPLOYEE table?

- A. **SELECT DISTINCT position, manager  
FROM employee;**
- B. SELECT position, manager DISTINCT  
FROM employee;
- C. SELECT position, manager  
FROM employee;
- D. SELECT position, DISTINCT manager  
FROM employee;

**Question: 2. (A)**

Which two are attributes of /SQL\*Plus? (Choose two)

- A. /SQL\*Plus commands cannot be abbreviated.
- B. /SQL\*Plus commands are accesses from a browser.
- C. **/SQL\*Plus commands are used to manipulate data in tables.**
- D. **/SQL\*Plus commands manipulate table definitions in the database.**
- E. /SQL\*Plus is the Oracle proprietary interface for executing SQL statements.

**Question: 3. (A)**

You need to perform certain data manipulation operations through a view called EMP\_DEPT\_VU, which you previously created.

You want to look at the definition of the view (the SELECT statement on which the view was create.)

How do you obtain the definition of the view?

- A. Use the DESCRIBE command in the EMP\_DEPT VU view.
- B. Use the DEFINE VIEW command on the EMP\_DEPT VU view.
- C. Use the DESCRIBE VIEW command on the EMP\_DEPT VU view.
- D. **Query the USER\_VIEWS data dictionary view to search for the EMP\_DEPT\_VU view.**

**Question: 4. (A)**

Which is an /SQL\*Plus command?

- A. INSERT
- B. UPDATE
- C. SELECT
- D. **DESCRIBE**
- E. DELETE
- F. RENAME

**Question: 5. (A)**

You need to produce a report for mailing labels for all customers. The mailing label must have only the customer name and address. The CUSTOMERS table has these columns:

CUST_ID	NUMBER(4)	NOT NULL
CUST_NAME	VARCHAR2(100)	
CUST_ADDRESS	VARCHAR2(150)	
CUST_PHONE	VARCHAR2(20)	

Which SELECT statement accomplishes this task?

- A. SELECT\*  
FROM customers;

- B. SELECT name,  
address FROM  
customers;
- C. SELECT id, name, address,  
phone FROM customers;
- D. SELECT cust\_name,  
cust\_address FROM  
customers;
- E. SELECT cust\_id, cust\_name, cust\_address,  
cust\_phone FROM customers;.

**Question: 6. (A)**

Evaluate this SQL statement:

```
SELECT e.EMPLOYEE_ID,e.LAST_NAME,e.DEPARTMENT_ID, d.DEPARTMENT_NAME.
FROM EMP e, DEPARTMENT d
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID;
```

In the statement, which capabilities of a SELECT statement are performed?

- A. Selection, projection, join
- B. Difference, projection, join
- C. Selection, intersection, join
- D. Intersection, projection, join
- E. Difference, projection, product

**Question: 7. (A)**

Which two statements are true regarding the ORDER BY clause? (Choose two)

- A. The sort is in ascending by order by default.
- B. The sort is in descending order by default.
- C. The ORDER BY clause must precede the WHERE clause.
- D. The ORDER BY clause is executed on the client side.
- E. The ORDER BY clause comes last in the SELECT statement.
- F. The ORDER BY clause is executed first in the query execution.

**Question: 8. (A)**

From SQL\*Plus, you issue this SELECT statement:

```
SELECT*
```

From orders;

You use this statement to retrieve data from a data table for \_\_\_\_\_. (Choose all that apply)

- A. Updating
- B. Viewing
- C. Deleting
- D. Inserting
- E. Truncating

**Question: 9. (A)**

Evaluate this SQL\*Plus command:

```
START delaccount
```

Which task will this command accomplish?

- A. It executes the DELACCOUNT PL/SQL routine.
- B. It runs the DELACCOUNT.SQL script file.
- C. It creates the DELACCOUNT file using the default file extension.
- D. It invokes the editor to edit the contents of the DELACCOUNT file.

**Question: 10. (A)**

Which SQL SELECT statement performs a projection, a selection, and join when executed?

- A. **SELECT p.id\_number, m.manufacturer\_id, m.city FROM product p, manufacturer m WHERE p.manufacturer\_id = m.manufacturer\_id AND m.manufacturer\_id = 'NF10032';**
- B. SELECT id\_number, manufacturer\_id FROM product ORDER BY manufacturer\_id, id\_number;
- C. SELECT id\_number, manufacturer\_id FROM product WHERE manufacturer\_id = 'NF10032';
- D. SELECT manufacturer\_id, city FROM manufacturer AND manufacturer\_id = 'NF10032' ORDER BY city;

**Question: 11. (A)**

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)	

Which statement finds the rows in the CUSTOMERS table that do not have a postal code?

- A. SELECT customer\_id, customer\_name FROM customers WHERE postal\_code CONTAINS NULL;
- B. SELECT customer\_id, customer\_name FROM customers WHERE postal\_code = '\_\_\_\_\_';
- C. **SELECT customer\_id, customer\_name FROM customers WHERE postal\_code IS NULL;**
- D. SELECT customer\_id, customer\_name FROM customers WHERE postal code IS NVL;
- E. SELECT customer\_id, customer\_name FROM customers WHERE postal\_code = NULL;

**Question: 12. (A)**

Evaluate these two SQL statements:

```
SELECT last_name, salary , hire_date
FROM EMPLOYEES
ORDER BY salary DESC;
```

```
SELECT last_name, salary, hire_date
FROM EMPLOYEES
ORDER BY 2 DESC;
```

What is true about them?

- A. **The two statements produce identical results.**
- B. The second statement returns a syntax error.
- C. There is no need to specify DESC because the results are sorted in descending order by default.
- D. The two statements can be made to produce identical results by adding a column alias for the salary column in the second SQL statement.
- E.

**Question: 13. (A)**

Evaluate the set of SQL statements:

```
CREATE TABLE dept
(deptno NUMBER(2),
dname VARCHAR2(14),
loc VARCHAR2(13));
ROLLBACK;
DESCRIBE DEPT
```

What is true about the set?

- A. The DESCRIBE DEPT statement displays the structure of the DEPT table.
- B. The ROLLBACK statement frees the storage space occupies by the DEPT table.
- C. The DESCRIBE DEPT statement returns an error ORA-04043: object DEPT does not exist.
- D. The DESCRIBE DEPT statement displays the structure of the DEPT table only if the us a COMMIT statement introduced before the ROLLBACK statement..

**Question: 14. (A)**

Examine the data of the EMPLOYEES table.

EMPLOYEES (EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID)

EMPLOYEE_ID	EMP_NAME	DEPT_ID	MGR_ID	JOB_ID	SALARY
101	Smith	20	120	SA_REP	4000
102	Martin	10	105	CLERK	2500
103	Chris	20	120	IT_ADMIN	4200
104	John	30	108	HR_CLERK	2500
105	Diana	30	108	HR_MGR	5000
106	Bryan	40	110	AD_ASST	3000
108	Jennifer	30	110	HR_DIR	6500
110	Bob	40		EX_DIR	8000
120	Ravi	20	110	SA_DIR	6500

Evaluate this SQL statement:

```
SELECT e.employee_id "Emp_id", e.emp_name "Employee",
e.salary, m.employee_id "Mgr_id", m.emp_name "Manager"
FROM employees e, employees m
WHERE e.mgr_id = m.employee_id
AND e.salary > 4000;
```

What is its output?

- A.  
EMP\_id EMPLOYEE SALARY Mgr\_id Manager  
-----

```
110 Bob 8000 Bob
120 Ravi 6500 110 Ravi
108 Jennifer 6500 110 Jennifer
103 Chris 4200 120 Chris
105 Diana 5000 108 Diana
```

- B.  
EMP\_id EMPLOYEE SALARY Mgr\_id Manager  
-----

```
120 Ravi 6500 110 Bob
```

108 Jennifer 6500 110 Bob  
103 Chris 4200 120 Ravi  
105 Diana 500 108 Jennifer

C.

EMP\_id EMPLOYEE SALARY Mgr\_id Manager

-----  
110 Bob 800  
120 Ravi 6500 110 Bob  
108 Jennifer 6500 110 Bob  
103 Chris 4200 120 Ravi  
105 Diana 5000 108 Jennifer

D

EMP\_id EMPLOYEE SALARY Mgr\_id Manager

-----  
110 Bob 8000 110 Bob  
120 Ravi 6500 120 Ravi  
108 Jennifer 6500 108 Jennifer  
103 Chris 4200 103 Chris  
105 Diana 5000 105 Dina

E. The SQL statement produces an error.

**Question: 15. (A)**

Which /SQL\*Plus feature can be used to replace values in the WHERE clause?

- A. Substitution variables
- B. Replacement variables
- C. Prompt variables
- D. Instead-of variables
- E. This feature cannot be implemented through /SQL\*Plus.

**Question: 16. (A)**

You are formulating queries in a SQL\*Plus. Which of the following statement correctly describes how to specify a column alias?

- A. Place the alias at the beginning of the statement to describe the table.
- B. Place the alias after each column separated by a space to describe the column.
- C. Place the alias after each column separated by a comma to describe the column.
- D. Place the alias at the end of the statement to describe the table.

**Question: 17. (A)**

You want to use a function in you column clause of a SQL statement. The NVL function accomplishes which of the following tasks?

- A. Assists in the distribution of output across multiple columns.
- B. Enables you to specify alternate output for non-NULL column values.
- C. Enables you to specify alternated out for NULL column values.
- D. Nullifies the value of the column out put.

**Question: 18. (A)**

You want to use SQL\*Plus to connect to the oracle database. Which of the following choices does not indicate a component you must specify when logging into the oracle?

- A. The SQL\*Plus Keyword.
- B. The username
- C. The password.
- D. The database name.

**Question: 19. (A)**

The EMPLOYEE\_HISTORY table contains these columns:

```
EMPLOYEE_ID NUMBER
LAST_NAME VARCHAR2(25)
FIRST_NAME VARCHAR2(25)
DEPARTMENT_ID NUMBER
POSITION VARCHAR2(30)
SALARY NUMBER(6,2)
HIRE_DATE DATE
DEPART_DATE DATE
```

The EMPLOYEE\_HISTORY table contains only former employees.

You need to create a report to display all former employees that were hired on or after January 1, 1996. The data should display in this format:

Former Employee Term of Employment

```
-----
14837 - SMITH 10-MAY-92 / 01-JUN-01
```

Which SELECT statement could you use?

- A. SELECT employee\_id||' - '||last\_name AS Former Employee, hire\_date||' / '||depart\_date AS Term of Employment  
FROM employee\_history  
WHERE hire\_date > '31-DEC-95';
- B. SELECT employee\_id||' - '||last\_name "AS Former Employee", hire\_date||' / '||depart\_date "AS Term of Employment"  
FROM employee\_history  
WHERE hire\_date > '31-DEC-95';
- C. SELECT employee\_id||' - '||last\_name 'Former Employee', hire\_date||' / '||depart\_date 'Term of Employment'  
FROM employee\_history  
WHERE hire\_date > '31-DEC-95'  
AND depart\_date > NULL;
- D. SELECT employee\_id||' - '||last\_name "Former Employee", hire\_date||' / '||depart\_date "Term of Employment"  
FROM employee\_history  
WHERE hire\_date > '31-DEC-95'  
AND depart\_date <> NULL;
- E. SELECT employee\_id||' - '||last\_name "Former Employee", hire\_date||' / '||depart\_date "Term of Employment"  
FROM employee\_history  
WHERE hire\_date > '31-DEC-95'  
AND depart\_date IS NOT NULL;

**Question: 20. (A)**

The EMPLOYEE table contains these columns:

```
Empno Number(4)
Ename Varchar2(10)
job varchar2(10)
sal Varchar2(10)
```

You need to display the employees information by using this query.

How many columns are presented after executing this query:

```
SELECT Empno||','||Ename||','||Job "Employee Information" FROM employee;
```

- A. 1
- B. 2
- C. 3
- D. 0
- E. 4

**Question: 21.(B)**

Examine the data of the EMPLOYEES table.

EMPLOYEES (EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID)

EMPLOYEE_ID	EMP_NAME	DEPT_ID	MGR_ID	JOB_ID	SALARY
101	Smith	20	120	SA_REP	4000
102	Martin	10	105	CLERK	2500
103	Chris	20	120	IT_ADMIN	4200
104	John	30	108	HR_CLERK	2500
105	Diana	30	108	HR_MGR	5000
106	Bryan	40	110	AD_ASST	3000
108	Jennifer	30	110	HR_DIR	6500
110	Bob	40		EX_DIR	8000
120	Ravi	20	110	SA_DIR	6500

Which statement lists the ID, name, and salary of the employee, and the ID and name of the employee's manager, for all the employees who have a manager and earn more than 4000?

- A. SELECT employee\_id "Emp\_id", emp\_name "Employee", salary, employee\_id "Mgr\_id", emp\_name "Manager" FROM employees WHERE salary > 4000;
- B. SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary, m.employee\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m WHERE e.mgr\_id = m.mgr\_id AND e.salary > 4000;
- C. SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary, m.employee\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m WHERE e.mgr\_id = m.employee\_id AND e.salary > 4000;
- D. SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary, m.mgr\_id "Mgr\_id", m.emp\_name "manager" FROM employees e, employees m WHERE e.mgr\_id = m.employee\_id AND e.salary > 4000;
- E. SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary, m.mgr\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m WHERE e.employee\_id = m.employee\_id AND e.salary > 4000;.

**Question: 22. (B)**

You need to display the last names of those employees who have the letter "A" as the second character in their names.

Which SQL statement displays the required results?

- A. SELECT last\_name  
FROM EMP  
WHERE last\_name LIKE '\_A%';
- B. SELECT last\_name  
FROM EMP  
WHERE last name = '\*A%'
- C. SELECT last\_name  
FROM EMP  
WHERE last name = '\_A%';
- D. SELECT last\_name  
FROM EMP  
WHERE last name LIKE '\*A%'

**Question: 23. (B)**

In which scenario would TOP N analysis be the best solution?

- A. You want to identify the most senior employee in the company.
- B. You want to find the manager supervising the largest number of employees.
- C. You want to identify the person who makes the highest salary for all employees.
- D. You want to rank the top three sales representatives who have sold the maximum number of products.

**Question: 24. (B)**

Evaluate this SQL statement:

```
SELECT c.customer_id, o.order_id, o.order_date, p.product_name
FROM customer c, curr_order o, product p
WHERE customer.customer_id = curr_order.customer_id
AND o.product_id = p.product_id
ORDER BY o.order_amount;
```

This statement fails when executed. Which change will correct the problem?

- A. Include the ORDER\_AMOUNT column in the SELECT list.
- B. Use the table name in the ORDER BY clause.
- C. Remove the table aliases from the WHERE clause.
- D. Use the table aliases instead of the table names in the WHERE clause.
- E. Remove the table alias from the ORDER BY clause and use only the column name.

**Question: 25. (B)**

You want to display the titles of books that meet these criteria:

Purchased before January 21, 2001  
Price is less than \$500 or greater than \$900

You want to sort the results by their date of purchase, starting with the most recently bought book.

Which statement should you use?

- A. 

```
SELECT book_title
FROM books
WHERE price between 500 and 900
AND purchase_date < '21-JAN-2001'
ORDER BY purchase_date;
```
- B. 

```
SELECT book_title
FROM books
WHERE price IN (500,900)
AND purchase_date < '21-JAN-2001'
ORDER BY purchase date ASC;
```
- C. 

```
SELECT book_title
FROM books
WHERE price < 500 or > 900
AND purchase_date < '21-JAN-2001'
ORDER BY purchase date DESC;
```
- D. 

```
SELECT book_title
FROM books
WHERE (price < 500 OR price > 900)
AND purchase_date < '21-JAN-2001'
ORDER BY purchase date DESC;
```



**Question: 26. (B)**

For which task would you use the WHERE clause in a SELECT statement?

- A. to designate the ORDER table location
- B. to compare PRODUCT\_ID values to 7382
- C. to display only unique PRODUCT\_ID values
- D. to restrict the rows returned by a GROUP BY clause

**Question: 27. (B)**

The STUDENT\_GRADES table has these columns:

STUDENT_ID	NUMBER(12)
SEMESTER_END	DATE
GPA	NUMBER(4,3)

The registrar has requested a report listing the students' grade point averages (GPA), sorted from highest grade point average to lowest within each semester, starting from the earliest date. Which statement accomplishes this?

- A. SELECT student\_id, semester\_end,  
gpa FROM student\_grades  
ORDER BY semester\_end DESC, gpa DESC;
- B. SELECT student\_id, semester\_end,  
gpa FROM student\_grades  
ORDER BY semester\_end ASC, gpa ASC;
- C. SELECT student\_id, semester\_end,  
gpa FROM student\_grades  
ORDER BY semester\_end, gpa DESC;
- D. SELECT student\_id, semester\_end,  
gpa FROM student\_grades  
ORDER BY gpa DESC, semester\_end DESC;
- E. SELECT student\_id, semester\_end,  
gpa FROM student\_grades  
ORDER BY gpa DESC, semester\_end ASC;.

**Question: 28. (B)**

The ORDERS table has these columns:

ORDER_ID	NUMBER(4)	NOT NULL
CUSTOMER_ID	NUMBER(12)	NOT NULL
ORDER_TOTAL	NUMBER(10,2)	

The ORDERS table tracks the Order number, the order total, and the customer to whom the Order belongs. Which two statements retrieve orders with an inclusive total that ranges between 100.00 and 2000.00 dollars? (Choose two.)

- A. SELECT customer\_id, order\_id,  
order\_total FROM orders  
RANGE ON order\_total (100 AND 2000) INCLUSIVE;
- B. SELECT customer\_id, order\_id,  
order\_total FROM orders  
HAVING order\_total BETWEEN 100 and 2000;
- C. SELECT customer\_id, order\_id,  
order\_total FROM orders  
WHERE order\_total BETWEEN 100 and 2000;
- D. SELECT customer\_id, order\_id,  
order\_total FROM orders  
WHERE order\_total >= 100 and <= 2000;

- E.       SELECT customer\_id, order\_id,  
          order\_total FROM orders  
          WHERE order\_total >= 100 and order\_total <= 2000;

**Question: 29. (B)**

Examine the structure of the PRODUCT table.

**PRODUCT Table**

LIST_PRICE	NUMBER (5,2)	
COST	NUMBER (5,2)	

You want to display all product identification numbers of products for which there are 500 or more available for immediate sale. You want the product numbers displayed alphabetically by supplier, then by product number from lowest to highest. Which statement should you use to achieve the required results?

- A.   SELECT Product\_id  
      FROM product  
      WHERE qty\_per\_unit >= 500 ORDER BY supplier\_id, product\_id;
- B.   SELECT product\_id  
      FROM product  
      WHERE qty\_per\_unit >= 500 SORT BY supplier\_id, product\_id;
- C.   SELECT product\_id  
      FROM product  
      WHERE qty\_per\_unit >= 500  
      ORDER BY supplier\_id, product\_id DESC;
- D.   SELECT product\_id  
      FROM product  
      WHERE qty\_per\_unit > 500  
      SORT BY supplier\_id, product\_id;

**Question: 30. (B)**

Examine the data in TEACHER table.

ID	LAST_NAME	FIRST_NAME	SUBJECT_ID
88	Tsu	Ming	HST AMER
70	Smith	Ellen	HST INDIA
56	Jones	Karen	HST_REVOL
58	Hann	Jeff	HST CURR
63	Hopewell	Mary Elizabethn	HST_RELIG

Which query should you use to return the following values from the TEACHER table?

Name Subject

-----  
Jones, Karen HST\_REVOL  
Hopewell, Mary Elizabeth HST\_RELIG

- A.       SELECT last\_name||', '||first\_name "Name", subject\_id  
          "Subject" FROM teacher  
          WHERE subject\_id LIKE 'HST\\_%' ESCAPE '\';
- B.       SELECT last\_name||', '||first\_name "Name", subject\_id  
          "Subject" FROM teacher  
          WHERE subject\_id = 'HST\R%';
- C.       SELECT last\_name||', '||first\_name "Name", subject\_id  
          "Subject" FROM teacher  
          WHERE subject\_id LIKE '%HST\R%' ESC '\';
- D.       SELECT last\_name||', '||first\_name "Name", subject\_id  
          "Subject" FROM teacher  
          WHERE subject\_id LIKE 'HST\_%';

**Question: 31. (B)**

You query the database with this SQL statement:

```
SELECT bonus
FROM salary
WHERE bonus BETWEEN 1 AND 250
OR (bonus IN(190, 500, 600)
AND bonus BETWEEN 250 AND 500);
```

Which value could the statement return?

- A. 100
- B. 260
- C. 400
- D. 600

**Question: 32. (B)**

Examine the structure of the STUDENTS table:

STUDENT_ID	NUMBER	NOT NULL, Primary KEY
STUDENT_NAME	VARCHAR2(30)	
COURSE_ID	VARCHAR2(10)	NOT NULL
MARKS	NUMBER	
START_DATE	DATE	
FINISH_DATE	DATE	

You need to create a report of the 10 students who achieved the highest ranking in the course INT SQL and who completed the course in the year 1999.

Which SQL statement accomplishes this task?

- A. 

```
SELECT student_id, marks, ROWNUM "Rank"
FROM students
WHERE ROWNUM <= 10
AND finish_date BETWEEN '01-JAN-99' AND '31-DEC-99'
AND course_id = 'INT_SQL'
ORDER BY marks DESC;
```
- B. 

```
SELECT student_id, marks, ROWID "Rank"
FROM students
WHERE ROWID <= 10
AND finish_date BETWEEN '01-JAN-99' AND '31-DEC-99'
AND course_id = 'INT_SQL'
ORDER BY marks;
```
- C. 

```
SELECT student_id, marks, ROWNUM "Rank"
FROM (SELECT student_id, marks
FROM students
WHERE ROWNUM <= 10
AND finish_date BETWEEN '01-JAN-99' AND
'31-DEC-99'
AND course_id = 'INT_SQL'
ORDER BY marks DESC);
```
- D. 

```
SELECT student_id, marks, ROWNUM "Rank"
FROM (SELECT student_id, marks
FROM students
ORDER BY marks DESC)
WHERE ROWNUM <= 10
AND finish_date BETWEEN '01-JAN-99' AND '31-DEC-99'
AND course_id = 'INT_SQL';
```

**Question: 33. (B)**

Examine the structure of the LINE\_ITEM table.

LINE_ITEM_ID	NUMBER(9)	NOT NULL, Primary Key
ORDER_ID	NUMBER(9)	NOT NULL, Primary Key, Foreign Key to ORDER_ID column of the CURR_ORDER table
PRODUCT_ID	NUMBER(9)	NOT NULL, Foreign Key to PRODUCT_ID column of the PRODUCT table
QUANTITY	NUMBER(9)	

You must display the order number, line item number, product identification number, and quantity of each item where the quantity ranges from 10 through 100. The order numbers must be in the range of 1500 through 1575. The results must be sorted by order number from lowest to highest and then further sorted by quantity from highest to lowest.

Which statement should you use to display the desired result?

- A. SELECT order\_id, line\_item\_id, product\_id, quantity  
FROM line\_item  
WHERE quantity BETWEEN 9 AND 101  
AND order\_id BETWEEN 1500 AND 1575  
ORDER BY order\_id DESC, quantity DESC;
- B. SELECT order\_id, line\_item\_id, product\_id, quantity  
FROM line\_item  
WHERE (quantity > 10 AND quantity < 100)  
AND order\_id BETWEEN 1500 AND 1575  
ORDER BY order\_id ASC, quantity;
- C. SELECT order\_id, line\_item\_id, product\_id, quantity  
FROM line\_item  
WHERE (quantity > 9 OR quantity < 101)  
AND order\_id BETWEEN 1500 AND 1575  
ORDER BY order\_id, quantity;
- D. SELECT order\_id, line\_item\_id, product\_id, quantity  
FROM line\_item  
WHERE quantity BETWEEN 10 AND 100  
AND order\_id BETWEEN 1500 AND 1575  
ORDER BY order\_id, quantity DESC;

**Question: 34. (B)**

The ITEM table contains these columns:

```
ITEM_ID  NUMBER(9)
COST     NUMBER(7,2)
RETAIL   NUMBER(7,2)
```

You need to create a report that displays the cost, the retail price, and the profit for item number 783920. To calculate the profit, subtract the cost of the item from its retail price, and then deduct an administrative fee of 25 percent of this derived value.

Which SELECT statement produces the desired results?

- A. SELECT cost, retail, (retail - cost) - ((retail - cost) \* .25) "Profit"  
FROM item  
WHERE item\_id = 783920;
- B. SELECT cost, retail, (retail - cost) - retail - (cost \* .25) "Profit"  
FROM item  
WHERE item\_id = 783920;

- C. `SELECT cost, retail, (retail - cost - retail - cost) * .25 "Profit"`  
`FROM item`  
`WHERE item_id = 783920;`
- D. `SELECT cost, retail, retail - cost - retail - cost * .25 "Profit"`  
`FROM item`  
`WHERE item_id = 783920;`

**Question: 35. (B)**

The ITEM table contains these columns:

ITEM_ID	NUMBER(9)
COST	NUMBER(7,2)
RETAIL	NUMBER(7,2)

The RETAIL and COST columns contain values greater than zero.

Evaluate these two SQL statements:

```
SELECT item_id, (retail * 1.25) + 5.00 - (cost * 1.10) - (cost * .10) AS Calculated Profit
FROM item;
```

```
SELECT item_id, retail * 1.25 + 5.00 - cost * 1.10 - cost * .10 "Calculated Profit"
FROM item;
```

What will be the result?

- A. Statement 1 will display the 'Calculated Profit' column heading.
- B. Statement 1 and statement 2 will return the same value.
- C. Statement 1 will return a higher value than statement 2.
- D. One of the statements will NOT execute.

**Question: 36. (B)**

The EMP table contains these columns:

LAST NAME	VARCHAR2(25)
SALARY	NUMBER(6,2)
DEPARTMENT_ID	NUMBER(6)

You need to display the employees who have not been assigned to any department.

You write the SELECT statement:

```
SELECT LAST_NAME, SALARY, DEPARTMENT_ID
FROM EMP
WHERE DEPARTMENT_ID = NULL;
```

What is true about this SQL statement?

- A. The SQL statement displays the desired results.
- B. The column in the WHERE clause should be changed to display the desired results.
- C. The operator in the WHERE clause should be changed to display the desired results.
- D. The WHERE clause should be changed to use an outer join to display the desired results.

**Question: 37. (B)**

Which two statements are true about WHERE and HAVING clauses? (Choose two)

- A. A WHERE clause can be used to restrict both rows and groups.
- B. A WHERE clause can be used to restrict rows only.
- C. A HAVING clause can be used to restrict both rows and groups.
- D. A HAVING clause can be used to restrict groups only.

- E. A WHERE clause CANNOT be used in a query if the query uses a HAVING clause.  
F. A HAVING clause CANNOT be used in subqueries.

**Question: 38. (B)**

You are sorting data in a table in your SELECT statement in descending order. The column you are sorting on contains NULL records, where will the NULL record appear?

- A. At the beginning of the list.
- B. At the end of the list.
- C. In the middle of the list.
- D. At the same location they are listed in the unordered table.

**Question: 39. (B)**

The ACCOUNT table contains these columns:

ACCOUNT_ID	NUMBER(12)
PREVIOUS_BALANCE	NUMBER(7,2)
PAYMENTS	NUMBER(7,2)
NEW_PURCHASES	NUMBER(7,2)
CREDIT_LIMIT	NUMBER(7)

You need to display the account number, finance charge, and current balance for accounts 1500 through 2000 with a current balance greater than the account's credit limit.

The finance charge is .9 percent (.009) of the previous balance. Adding the previous balance value, new purchases value, and finance charge value, and then subtracting the payments value yields the current balance value.

Evaluate this statement:

```
SELECT account_id, NVL(previous_balance, 0) * .009 finance_charge,  
NVL(new_purchases, 0) + (NVL(previous_balance, 0) * 1.009) - NVL(payments, 0)  
current_balance  
FROM account  
WHERE (new_purchases + (previous_balance * 1.009)) - payments > credit_limit AND  
account_id BETWEEN 1500 AND 2000;
```

Which statement about this SELECT statement is true?

- A. The statement calculates the finance charge incorrectly.
- B. The statement calculates the current balance incorrectly.
- C. The statement returns only accounts that have NO previous balance.
- D. The statement returns only accounts that have new purchases, previous balance, and payments values.

**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	VARCHAR2(30)	
SALARY	NUMBER(8,2)	

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

- A. 

```
SELECT dept_id, job_cat, MAX(salary)  
FROM employees  
WHERE salary > MAX(salary);
```

- B. SELECT dept\_id, job\_cat, MAX(salary)  
FROM employees  
GROUP BY dept\_id, job\_cat;
- C. SELECT dept\_id, job\_cat, MAX(salary)  
FROM employees;
- D. SELECT dept\_id, job\_cat, MAX(salary)  
FROM employees  
GROUP BY dept\_id;
- E. 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  $12 * \text{salary} * \text{comossion\_pct}$  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?

- A. SELECT last\_name,  $12 * \text{salary} * \text{commison\_pct}$   
FROM emp;
- B. SELECT last\_name,  $12 * \text{salary} * (\text{commission\_pct}, 0)$   
FROM emp;
- C. SELECT last\_name,  $12 * \text{salary} * (\text{nvl}(\text{commission\_pct}, 0))$   
FROM emp;
- D. SELECT last\_name,  $12 * \text{salary} * (\text{decode}(\text{commission\_pct}, 0))$   
FROM emp;

**Question: 42. (C)**

Examine the description of the STUDENTS table:

STD_ID	NUMBER(4)
COURSE_ID	VARCHAR2(10)
START_DATE	DATE
END_DATE	DATE.

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

- A. SUM(start\_date)
- B. AVG(start\_date)
- C. COUNT(start\_date)
- D. AVG(start\_date, end\_date)
- E. MIN(start\_date)
- F. 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?

- A. SELECT last\_name, (salary \* 12) \* commission\_pct  
FROM EMPLOYEES;
- B. SELECT last\_name, (salary \* 12) \* IFNULL(commission\_pct, 0)  
FROM EMPLOYEES;
- C. SELECT last\_name, (salary \* 12) \* NVL2(commission\_pct, 0)  
FROM EMPLOYEES;
- D. 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?

- A. SELECT TO\_DATE(SYSDATE, 'FMDAY, DD Month, YYYY')  
FROM dual;
- B. SELECT TO\_CHAR(SYSDATE, 'FMDD, DY Month, 'YYY')  
FROM dual;
- C. SELECT TO\_CHAR(SYSDATE, 'FMDay, DD Month, YYYY')  
FROM dual;
- D. SELECT TO\_CHAR(SYSDATE, 'FMDY, DDD Month, YYYY')  
FROM dual;
- E. 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?

- A. 0
- B. 1
- C. 0.00
- D. 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.
- C. The statement returns an error at the WHERE clause.
- D. 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)

- A. SELECT TO\_CHAR (2000, ‘\$#,###.##’)  
FROM dual;
- B. SELECT TO\_CHAR (2000, ‘\$0,000.00’)  
FROM dual;
- C. SELECT TO\_CHAR (2000, ‘\$9,999.00’)  
FROM dual;
- D. SELECT TO\_CHAR (2000, ‘\$9,999.99’)  
FROM dual;
- E. SELECT TO\_CHAR (2000, ‘\$2,000.00’)  
FROM dual;
- F. 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?

- A. SELECT dept\_id, MIN(salary), MAX(salary)  
FROM employees  
WHERE MIN(salary) < 5000 AND MAX(salary) > 15000;
- B. SELECT dept\_id, MIN(salary), MAX(salary)  
FROM employees  
WHERE MIN(salary) < 5000 AND MAX(salary) > 15000  
GROUP BY dept\_id;
- C. SELECT dept\_id, MIN(salary), MAX(salary)  
FROM employees  
HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;
- D. SELECT dept\_id, MIN(salary), MAX(salary)  
FROM employees  
GROUP BY dept\_id  
HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;
- E. 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.)

- A. You can use aggregate functions in any clause of a SELECT statement.
- B. You can use aggregate functions only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.
- C. You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns.

- D. You can pass column names, expressions, constants, or functions as parameters to an aggregate function.
- E. You can use aggregate functions on a table, only by grouping the whole table as one single group.
- F. 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)

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

**Question: 51. (C)**

Examine the structures of the PATIENT, PHYSICIAN, and ADMISSION tables.

**PATIENT Table**

PATIENT_ID	NUMBER	NOT NULL, Primary Key
LAST_NAME	VARCHAR2 (30)	NOT NULL
FIRST_NAME	VARCHAR2 (25)	NOT NULL
DOB	DATE	
INS_CODE	NUMBER	

**PHYSICIAN Table**

PHYSICIAN_ID	NUMBER	NOT NULL, Primary Key
LAST_NAME	VARCHAR2 (30)	NOT NULL
FIRST_NAME	VARCHAR2 (25)	NOT NULL
LICENSE_NO	NUMBER (7)	NOT NULL
HIRE_DTAE	DATE	

**ADMISSION Table**

PATIENT_ID	NUMBER	NOT NULL, Primary Key, References PATIENT_ID column of the PATIENT table
PHYSICIAN_ID	NUMBER	NOT NULL, Primary Key, References PHYSICIAN_ID column of the PHYSICIA table
ADMIT_DATE	DATE	
DISCHG_DATE	DATE	
ROOM_ID	NUMBER	Foreign key to ROOM_ID of the ROOM table

Which SQL statement will produce a list of all patients who have more than one physician?

- A. 

```
SELECT p.patient_id
FROM patient p
WHERE p.patient_id IN (SELECT patient_id
FROM admission
GROUP BY patient_id
HAVING COUNT(*) >
1);
```
- B. 

```
SELECT DISTINCT a.patient_id FROM
admission a, admission a2 WHERE
```

- a.patient\_id = a2.patient\_id AND  
a.physician\_id <> a2.physician\_id;
- C. SELECT patient\_id  
FROM admission  
WHERE COUNT(physician\_id) > 1;
- D. SELECT patient\_id  
FROM patient FULL OUTER JOIN physician;

**Question: 52. (C)**

Which clause should you use to exclude group results?

- A. WHERE  
B. HAVING  
C. RESTRICT  
D. GROUP BY  
E. 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?

- A. Immediately after the SELECT clause  
B. Before the WHERE clause  
C. Before the FROM clause  
D. After the ORDER BY clause  
E. After the WHERE clause

**Question: 54. (C)**

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

- A. TRIM  
B. REPLACE  
C. TRUNC  
D. TO\_DATE  
E. MOD  
F. 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:

Display the salary multiplied by the commission\_pct.  
Exclude employees with a zero commission\_pct.  
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?

- A. All of the desired results

- B. Two of the desired results
- C. One of the desired results
- D. An error statement

**Question: 56. (C)**

Examine the structures of the EMPLOYEE and CURR\_ORDER tables as shown below:

**EMPLOYEE Table**

EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
LAST_NAME	VARCHAR2 (25)	
FIRST NAME	VARCHAR2 (25)	
DEPT_ID	NUMBER	Foreign key to DEPT_ID column of the DEPARTMENTS table
JOB_ID	NUMBER	Foreign key to JOB_ID column of the JOBS table
MGR_ID	NUMBER	References EMPLOYEE_ID column
SALARY	NUMBER (9,2)	
COMMISSION	NUMBER (7,2)	
HIRE_DATE	DATE	

**CURR\_ORDER Table**

ORDER_ID	NUMBER	NOT NULL, Primary Key
CUSTOMER_ID	NUMBER	Foreign key to CUSTOMER_ID column of the COSTOMERS table
EMPLOYEE_ID	NUMBER	Foreign key to EMP_ID column of the EMPLOYEES table
ORDER_DATE	DATE	
ORDER_AMT	NUMBER (7,2)	
SHIP_METHOD	NUMBER (5)	

You queried the database with this SQL statement:

```
SELECT a.last_name, a.first_name, a.job_id, NVL(a.commission, 0), b.avgcomm
FROM employee a, (SELECT job_id, AVG(commission) AVGCOMM
FROM employee
WHERE commission IS NOT NULL
GROUP BY job_id) b
WHERE a.job_id = b.job_id
AND a.commission < b.avgcomm;
```

Which is a result of this query?

- A. The AVG function's DISTINCT keyword must be used in the inner query or the statement will fail when executed.
- B. The employee information displayed will be for employees that have a commission that is less than the average commission of all employees who have the same job.
- C. The employee information displayed will be displayed in numeric order by the JOB\_ID and in alphabetical order by the LAST\_NAME where the JOB\_IDs are the same.
- D. A self join CANNOT be used in an outer query when the inner query is an inline view. The self join must be placed in the inner query for the statement to execute successfully.

**Question: 57. (C)**

Examine the structures of the EMPLOYEE and CURR\_ORDER tables.

**EMPLOYEE Table**

EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
LAST_NAME	VARCHAR2 (25)	
FIRST NAME	VARCHAR2 (25)	
DEPT_ID	NUMBER	Foreign key to DEPT_ID column of the

		DEPARTMENTS table
JOB_ID	NUMBER	Foreign key to JOB_ID column of the JOBS table
MGR_ID	NUMBER	References EMPLOYEE_ID column
SALARY	NUMBER (9,2)	
COMMISSION	NUMBER (7,2)	
HIRE_DATE	DATE	

**CURR\_ORDER Table**

ORDER_ID	NUMBER	NOT NULL, Primary Key
CUSTOMER_ID	NUMBER	Foreign key to CUSTOMER_ID column of the COSTOMERS table
EMPLOYEE_ID	NUMBER	Foreign key to EMP_ID column of the EMPLOYEES table
ORDER_DATE	DATE	
ORDER_AMT	NUMBER (7, 2)	
SHIP_METHOD	NUMBER (5)	

To keep your top sales representatives motivated, your company plans to increase the bonuses of employees. You need to create a SELECT statement that returns the name, bonus, and maximum order amount associated with each employee for all employees whose bonus is less than 8 percent of their maximum order amount.

Which SELECT statement should you use?

- A. SELECT e.last\_name, e.first\_name, e.commission, o.maxamt  
FROM employee e, (SELECT employee\_id, MAX(order\_amt) MAXAMT  
FROM curr\_order  
GROUP BY employee\_id) o  
WHERE e.employee\_id = o.employee\_id  
AND e.commission < .08 \* o.maxamt;
- B. SELECT e.last\_name, e.first\_name, e.commission, MAX(o.order\_amount) MAXAMT  
FROM employee e, curr\_order o  
WHERE e.employee\_id = o.employee\_id  
AND e.commission < .08 \* o.maxamt  
GROUP BY e.last\_name, e.first\_name, e.commission;
- C. SELECT e.last\_name, e.first\_name, e.commission, MAX(o.order\_amt)  
FROM employee e, curr\_order  
WHERE e.employee\_id = o.employee\_id  
AND e.commission < .08 \* (SELECT employee\_id, MAX(order\_amt) MAXAMT  
FROM curr\_order  
GROUP BY employee\_id);
- D. SELECT last\_name, first\_name, commission, max\_amt  
FROM employee, (SELECT employee\_id, MAX(order\_amt) MAXAMT  
FROM curr\_order  
GROUP BY employee\_id)  
WHERE employee.employee\_id = curr\_order.employee\_id  
AND commission < .08 \* maxamt;

**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?

- A. 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;
- B. 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;
- C. 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;
- D. 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"?

- A. SELECT TO\_CHAR(SYSDATE,'yyyy')  
FROM dual;
- B. SELECT TO\_DATE(SYSDATE,'yyyy')  
FROM dual;
- C. SELECT DECODE(SUBSTR(SYSDATE, 8), 'YYYY')  
FROM dual;
- D. SELECT DECODE(SUBSTR(SYSDATE, 8), 'year')  
FROM dual;
- E. 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.)

- A. SELECT TOTAL(\*)  
FROM customers;
- B. SELECT COUNT(\*)  
FROM customers;
- C. SELECT TOTAL(customer\_id)  
FROM customers;
- D. SELECT COUNT(customer\_id)  
FROM customers;
- E. SELECT COUNT(customers)  
FROM customers;
- F. 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
- C. 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
- D. 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?

- A. Using the SUM aggregate function with the AMOUNT column is allowed in any portion of a SELECT statement.
- B. Using the AVG aggregate function with any column in the table is allowed.
- C. Using the SUM aggregate function on the AMOUNT column will result in erroneous results because the column contains null values.
- D. Grouping on the REQUIRED\_DT and COMPL\_DT columns is NOT allowed.
- E. Using the AVG aggregate function on the AMOUNT column ignores null values.
- F. 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?

- A. SELECT category, SUM(price \* quantity) TOTAL, location  
FROM inventory  
WHERE price > 100.00  
GROUP BY category;
- B. SELECT category, location, SUM(price)  
FROM inventory  
WHERE price > 100.00 GROUP  
BY category, location;
- C. SELECT category, SUM(price \* quantity) TOTAL, location  
FROM inventory  
WHERE price > 100.00;
- D. 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?

- A. SELECT UNIQUE(location\_id), COUNT(event\_type)  
FROM event  
GROUP BY location\_id;
- B. SELECT COUNT(\*), DISTINCT(location\_id)  
FROM event;
- C. SELECT DISTINCT (event\_type)  
FROM event  
GROUP BY location\_id;
- D. SELECT location\_id, COUNT(DISTINCT event\_type)  
FROM event  
GROUP BY location\_id;
- E. 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.)

- A. The Oracle Server will evaluate a HAVING clause before a WHERE clause.
- B. The Oracle Server will evaluate a WHERE clause before a GROUP BY clause.
- C. The Oracle Server will evaluate a GROUP BY clause before a HAVING clause.
- D. The Oracle Server will evaluate an ORDER BY clause before a WHERE clause.
- E. The Oracle Server will evaluate an ORDER BY clause before a HAVING clause.

**Question: 66. (C)**

Examine the structures of the EMPLOYEE and DEPARTMENT tables:

EMPLOYEE

```

-----
EMP_ID   NUMBER      NOT NULL    PK
NAME     VARCHAR(30)  NOT NULL
FNAME    VARCHAR(25)  NOT NULL
DEPT_NO  NUMBER
TITLE    VARCHAR2(25)

```

DEPARTMENT

```

-----
DEPT_ID   NUMBER      NOT NULL    PK
DEPT_NAME VARCHAR2(25)

```

You need to produce a list of departments, including the department names, that have more than three administrative assistants. Which SELECT statement will produce the desired result?

- A. SELECT dept\_name  
FROM employee JOIN department  
ON employee.dept\_id = department.dept\_id  
WHERE UPPER(title) = 'ADMINISTRATIVE ASSISTANT'  
GROUP BY dept\_name  
HAVING emp\_id > 3;
- B. SELECT dept\_name  
FROM employee  
GROUP BY dept\_no  
HAVING LOWER(title) = 'administrative assistant' AND COUNT(\*) > 3;

- C. 

```
SELECT dept_name
FROM employee NATURAL JOIN department
WHERE LOWER(title) = 'administrative assistant'
GROUP BY dept_name
HAVING COUNT(emp_id) > 3;
```
- D. 

```
SELECT dept_name
FROM employee e JOIN department d
ON (e.dept_no = d.dept_id)
WHERE LOWER(title) = 'administrative assistant'
AND COUNT(*) > 3;
```
- E. 

```
SELECT d.dept_name
FROM employee e JOIN department d
ON (e.dept_no = d.dept_id)
WHERE LOWER(title) = 'administrative assistant'
GROUP BY dept_name
HAVING COUNT(emp_id) > 3;
```
- F. 

```
SELECT d.dept_name
FROM e.employee JOIN d.department
ON (e.dept_no = d.dept_id)
WHERE LOWER(title) = 'administrative assistant'
GROUP BY dept_name
HAVING COUNT(emp_id) > 3;
```

**Question: 67. (C)**

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

- A. Convert 10 to 'TEN'
- B. Convert '10' to 10
- C. Convert 10 to '10'
- D. Convert 'TEN' to 10
- E. Convert a date to a character expression
- F. Convert a character expression to a date

**Question: 68. (C)**

Examine the data in the LINE\_ITEM table.

LINE_ITEM_ID	ORDER_ID	PRODUCT_ID	QUANTITY
2	1494	A-2356	7
3	1533	A-7849	18
6	1589	C-589	33
1	1533	A-3209	100
2	1533	A-3210	1
4	1494	Z-78	1
10	1588	C-555	250
3	1494	Z-79	5

You query the database and return the value 23. Which script did you use?

- A. 

```
SELECT SUBSTR(product_id, 3)
FROM line_item
WHERE line_item_id = 2
AND order_id = 1494;
```
- B. 

```
SELECT SUBSTR(product_id, 3, -2)
FROM line_item
WHERE line_item_id = 2
AND order_id = 1494;
```
- C. 

```
SELECT SUBSTR(product_id, -3, 2)
FROM line_item
```

- ```
WHERE line_item_id = 2
AND order_id = 1494;
D. SELECT SUBSTR(product_id, 3, 2)
FROM line_item
WHERE line_item_id = 2
AND order_id = 1494;
```

**Question: 69. (C)**

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

- A. SELECT ename, salary\*12 'Annual Salary'  
FROM employees;
- B. SELECT ename, salary\*12 "Annual Salary"  
FROM employees;
- C. SELECT ename, salary\*12 AS Annual Salary  
FROM employees;
- D. SELECT ename, salary\*12 AS INITCAP("ANNUAL SALARY")  
FROM employees

**Question: 70. (C)**

Examine the structure of the EMPLOYEES table:

| EMPLOYEE_ID | NUMBER       | Primary Key |
|-------------|--------------|-------------|
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| DEPARTMENT_ | ID NUMBER    |             |
| SALARY      | NUMBER       |             |

What is the correct syntax for an inline view?

- A. SELECT a.last\_name, a.salary, a.department\_id, b.maxsal  
FROM employees a,  
(SELECT department\_id, max(salary)maxsal  
FROM employees  
GROUP BY department\_id) b  
WHERE a.department\_id = b.department\_id  
AND a.salary < b.maxsal;
- B. SELECT a.last name, a.salary, a.department\_id  
FROM employees a  
WHERE a.department\_id IN  
(SELECT department\_id  
FROM employees b  
GROUP BY department\_id having salary =  
(SELECT max(salary) from employees))
- C. SELECT a.last\_name, a.salary, a.department\_id  
FROM employees a  
WHERE a.salary =  
(SELECT max(salary)  
FROM employees b  
WHERE a.department\_id = b.department\_id);
- D. SELECT a.last\_name, a.salary, a.department\_id  
FROM employees a  
WHERE (a.department\_id, a.salary) IN  
(SELECT department\_id, a.salary) IN  
(SELECT department\_id max(salary)  
FROM employees b  
GROUP BY department\_id  
ORDER BY department\_id);

**Question: 71. (C)**

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

- A. ORDER BY SALARY > 5000.
- B. GROUP BY SALARY > 5000
- C. HAVING SALARY > 5000
- D. 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?

- A. MAX
- B. MIN
- C. SUM
- D. COUNT
- E. TOTAL
- F. LARGEST

**Question: 73. (C)**

Which SELECT statement will the result 'elloworld' from the string 'HelloWorld'?

- A. SELECT SUBSTR('HelloWorld',1) FROM dual;
- B. SELECT INITCAP(TRIM('HelloWorld', 1,1)) FROM dual;
- C. SELECT LOWER(SUBSTR('HelloWorld', 1, 1)) FROM dual;
- D. SELECT LOWER(SUBSTR('HelloWorld', 2, 1)) FROM dual;
- E. 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?

- A. The value displayed in the CALC\_VALUE column will be lower.
- B. The value displayed in the CALC\_VALUE column will be higher.
- C. There will be no difference in the value displayed in the CALC\_VALUE column.
- D. An error will be reported.

**Question: 75. (C)**

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

- A. SELECT ROUND(enroll\_date, 'DAY')  
FROM student;
- B. SELECT ROUND(enroll\_date, 'YEAR')  
FROM student;
- C. SELECT ROUND(enroll\_date, 'MONTH')  
FROM student;
- D. SELECT ROUND(TO\_CHAR(enroll\_date, 'YYYY'))  
FROM student;

**Question: 76. (C)**

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

- A. CONCAT
- B. ROUND
- C. TRUNC
- D. RPAD
- E. 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?

- A. REPLACE
- B. SUBSTR
- C. LENGTH
- D. RPAD

**Question: 78. (C)**

Which statement concerning SQL functions is true?

- A. Character functions can return character or number values.
- B. Conversion functions convert a column definition from one data type to another data type.
- C. Single-row functions can only be used in SELECT and WHERE clauses.
- D. 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?

- A. `SELECT AVERAGE(gpa)`  
`FROM student_grades`  
`WHERE semester_end > '01-JAN-2000' and semester end < 31-DEC-2000';`
- B. `SELECT COUNT(gpa)`  
`FROM student grades`  
`WHERE semester_end > '01-JAN-2000' and semester end < '31-DEC-2000';`
- C. `SELECT MID(gpa)`  
`FROM student grades`  
`WHERE semester_end > '01-JAN-2000' and semester end < '31-DEC-2000';.`
- D. `SELECT AVG(gpa)`  
`FROM student_grades`  
`WHERE semester_end BETWEEN '01-JAN-2000' and '31.DEC.2000';`
- E. `SELECT SUM(gpa)`  
`FROM student grades`  
`WHERE semester_end > '01-JAN-2000' and semester end < '31-DEC-2000';`
- F. `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`
2. `FROM employees a,`
3. `(SELECT dept_id, MAX(sal) maxsal`
4. `FROM employees`
5. `GROUP BY dept_id) b`
6. `WHERE a.dept_id = b.dept_id`
7. `AND a.sal < b.maxsal;`

What is the result of the statement?

- A. The statement produces an error at line 1.
- B. The statement produces an error at line 3.
- C. The statement produces an error at line 6.
- D. 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.
- E. 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?

- A. SELECT ENAME  
FROM EMP  
WHERE SYSDATE-HIRE\_DATE > 5;
- B. SELECT ENAME  
FROM EMP  
WHERE HIRE\_DATE -SYSDATE > 5;
- C. SELECT ENAME  
FROM EMP  
WHERE (SYSDATE -HIRE\_DATE)/365 > 5;
- D. 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'?

- A. SELECT SUBSTR( 'Hello World',1) FROM dual;
- B. SELECT INITCAP(TRIM( 'Hello World', 1,1)) FROM dual;
- C. SELECT LOWER(SUBSTR('Hello World', 1, 1) FROM dual;
- D. SELECT LOWER(SUBSTR('Hello World', 2, 1) FROM dual;
- E. SELECT LOWER(TRIM( 'H' FROM 'Hello World')) FROM dual;

**Question: 83. (C)**

Evaluate the SQL statement:

```
SELECT ROUND(45.953, -1), TRUNC(45.936, 2)
FROM dual;
```

Which values are displayed?

- A. 46 and 45
- B. 46 and 45.93
- C. 50 and 45.93
- D. 50 and 45.9
- E. 45 and 45.93
- F. 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?

- A. COUNT(UPPER(country\_address))
- B. COUNT(DIFF(UPPER(country\_address)))
- C. COUNT(UNIQUE(UPPER(country\_address)))
- D. COUNT DISTINCTC UPPER(country\_address)
- E. COUNT(DISTINTC (UPPER(country\_address)))

**Question: 85. (C)**

You want to use single row function in your SQL statements which three of the following are number functions? (Choose three).

- A. SINH
- B. TO\_NUMBER.
- C. SQRT.
- D. ROUND.

**Question: 86. (C)**

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

- A. SQRT
- B. DECODE
- C. NEW\_TIME
- D. ROWIDTOCHAR.

**Question: 87. (C)**

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

- A. INSTR.
- B. NVL2.
- C. TRUNCATE.
- D. DECODE.
- E. TRIM.
- F. NVL.
- G. 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?

- A. SELECT employee\_id, department\_id, department\_name, salary.  
FROM employees  
WHERE department\_id IN (SELECT department\_id  
FROM departments);
- B. SELECT employee\_id, department\_id, department\_name, salary  
FROM employees  
NATURAL JOIN departments;
- C. SELECT employee\_id, d.department\_id, department\_name, salary  
FROM employees e  
JOIN departments d  
ON e.department\_id = d.department\_id;
- D. 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?

- A. SELECT last\_name, department\_name  
FROM employees , departments(+);
- B. SELECT last\_name, department\_name  
FROM employees JOIN departments (+);
- C. SELECT last\_name, department\_name FROM  
employees(+) e JOIN departments d ON  
(e.department\_id = d.department\_id);
- D. SELECT last\_name, department\_name  
FROM employees e

- RIGHT OUTER JOIN departments d ON (e.department\_id = d.department\_id);
- E. SELECT last\_name, department\_name  
FROM employees(+) , departments  
ON (e.department\_id = d.department\_id);
- F. 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?

- A. Both tables have NULL values.
- B. You want all unmatched data from one table.
- C. You want all matched data from both tables.
- D. You want all unmatched data from both tables.
- E. One of the tables has more data than the other.
- F. 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.)

- A. You cannot use IN operator in a condition that involves an outerjoin.
- B. You use (+) on both sides of the WHERE condition to perform an outerjoin.
- C. You use (\*) on both sides of the WHERE condition to perform an outerjoin.
- D. You use an outerjoin to see only the rows that do not meet the join condition.
- E. In the WHERE condition, you use (+) following the name of the column in the table without matching rows, to perform an outerjoin.
- F. 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?

- A. You can join a maximum of two tables through an equijoin.
- B. You can join a maximum of two columns through an equijoin.
- C. You specify an equijoin condition in the SELECT or FROM clauses of a SELECT statement.
- D. To join two tables through an equijoin, the columns in the join condition must be primary key and foreign key columns.
- E. 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: 93. (D)**

Examine the structure of the EMPLOYEES, DEPARTMENTS, 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 |

**DEPARTMENTS**

|                 |              |                                                 |
|-----------------|--------------|-------------------------------------------------|
| DEPARTMENT_ID   | NUMBER       | NOT NULL, Primary Key                           |
| DEPARTMENT_NAME | VARCHAR2  30 |                                                 |
| MGR_ID          | NUMBER       | References MGR_ID column of the EMPLOYEES table |

---

**TAX**

|             |        |
|-------------|--------|
| MIN_SALARY  | NUMBER |
| MAX_SALARY  | NUMBER |
| TAX_PERCENT | NUMBER |

For which situation would you use a nonequijoin query?

- A. To find the tax percentage for each of the employees.
- B. To list the name, job id, and manager name for all the employees.
- C. To find the name, salary, and department name of employees who are not working with Smith.
- D. To find the number of employees working for the Administrative department and earning less than 4000.
- E. To display name, salary, manager ID, and department name of all the employees, even if the employees do not have a department ID assigned.

**Question: 94. (D)**

Examine the structure of the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables.

**EMPLOYEES**

|               |               |                                                              |
|---------------|---------------|--------------------------------------------------------------|
| EMPLOYEE_ID   | NUMBER NOT    | 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**

|                 |               |                                                          |
|-----------------|---------------|----------------------------------------------------------|
| DEPARTMENT_ID   | NUMBER        | NOT NULL, Primary Key                                    |
| DEPARTMENT_NAME | VARCHAR2 (30) |                                                          |
| MGR_ID          | NUMBER        | References MGR_ID column of the EMPLOYEES table          |
| LOCATION_ID     | NUMBER        | Foreign key to LOCATION_ID column of the LOCATIONS table |

**LOCATIONS**

|             |               |                       |
|-------------|---------------|-----------------------|
| LOCATION_ID | NUMBER        | NOT NULL, Primary Key |
| CITY        | VARCHAR2 (30) |                       |
|             |               |                       |

Which two SQL statements produce the name, department name, and the city of all the employees who earn more than 10000? (Choose two)

- A. 

```
SELECT emp_name, department_name, city
FROM employees e
JOIN departments d
USING (department_id)
JOIN locations l USING
(location_id) WHERE
salary > 10000;
```
- B. 

```
SELECT emp_name, department_name, city FROM
employees e, departments d, locations l JOIN ON
(e.department_id = d.department_id) AND
(d.location_id = l.location_id)
AND salary > 10000;
```
- C. 

```
SELECT emp_name, department_name, city FROM
employees e, departments d, locations l
WHERE salary > 10000;
```
- D. 

```
SELECT emp_name, department_name, city FROM
employees e, departments d, locations l
```

WHERE e.department\_id = d.department\_id AND  
d.location\_id = 1.location\_id  
AND salary > 10000;

- E. SELECT emp\_name, department\_name, city  
FROM employees e  
NATURAL JOIN departments, locations  
WHERE salary > 10000;

**Question: 95. (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          | 3            |               |

**INSTRUCTOR Table**

| INSTRUCTOR_ID | LAST_NAME  | FIRST_NAME |
|---------------|------------|------------|
| 1             | Chao       | Ling       |
| 2             | Vanderbilt | Herbert    |
| 3             | Wigley     | Martha     |
| 4             | Page       | Albert     |

You have been asked to produce a report of all instructors, including the classes taught by each instructor. All instructors must be included on the report, even if they are not currently assigned to teach classes. Which two SELECT statements could you use? (Choose two.)

- SELECT i.last\_name, i.first\_name, c.class\_name  
FROM instructor i NATURAL JOIN class c  
ON (i.instructor\_id = c.instructor\_id);
- SELECT i.last\_name, i.first\_name, c.class\_name  
FROM instructor i, class c;
- SELECT i.last\_name, i.first\_name, c.class\_name  
FROM class c LEFT OUTER JOIN instructor i  
ON (i.instructor\_id = c.instructor\_id)  
ORDER BY i.instructor\_id;
- SELECT i.last\_name, i.first\_name, c.class\_name  
FROM instructor i, class c  
WHERE i.instructor\_id = c.instructor\_id (+)  
ORDER BY i.instructor\_id;
- SELECT i.last\_name, i.first\_name, c.class\_name  
FROM instructor i LEFT OUTER JOIN class c ON  
(i.instructor\_id = c.instructor\_id)  
ORDER BY i.instructor\_id;
- SELECT i.last\_name, i.first\_name, c.class\_name  
FROM instructor i, class c  
WHERE i.instructor\_id (+) = c.instructor\_id  
ORDER BY i.instructor\_id;

**Question: 96. (D)**

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

- =
- OR
- IN
- 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?

- A. SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName  
FROM instructor i, class c  
WHERE i.instructor\_id = c.instructor\_id (+);
- B. SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName  
FROM instructor I JOIN class c  
WHERE i.instructor\_id (+) = c.instructor\_id;
- C. 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);
- D. SELECT class\_name, last\_name || ', ' || first\_name InstructorName  
FROM instructor RIGHT OUTER JOIN class;
- E. 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?

- A. 8
- B. 2
- C. 3
- D. 4
- E. 5
- F. There is no such criteria

**Question: 99.(E)**

Examine the data in the EMPLOYEES table:

| LAST_NAME | DEPARTMENT_ID | SALARY |
|-----------|---------------|--------|
| Getz      | 10            | 3000   |
| Davis     | 20            | 1500   |
| King      | 20            | 2200   |
| Davis     | 30            | 5000   |
| ...       |               |        |

Which three subqueries work? (Choose three)

- A. SELECT \*  
FROM employees  
where salary > (SELECT MIN(salary)  
FROM employees  
GROUP BY department.id);
- B. SELECT \*  
FROM employees  
WHERE salary = (SELECT AVG(salary)  
FROM employees  
GROUP BY department\_id);
- C. SELECT distinct department\_id  
FROM employees  
Where salary > ANY (SELECT AVG(salary)  
FROM employees  
GROUP BY department\_id);
- D. SELECT department\_id  
FROM employees  
WHERE SALARY > ALL (SELECT AVG(salary)  
FROM employees.  
GROUP BY department\_id);
- E. SELECT last\_name  
FROM employees  
Where salary > ANY (SELECT MAX(salary)  
FROM employees  
GROUP BY department\_id);
- F. SELECT department\_id  
FROM employees  
WHERE salary > ALL (SELECT AVG(salary)  
FROM employees  
GROUP BY AVG(SALARY));

**Question: 100. (E)**

Examine the data from the ORDERS and CUSTOMERS table.

**ORDERS**

| ORD_ID | ORD_DATE     | CUST_ID | ORD_TOTAL |
|--------|--------------|---------|-----------|
| 100    | 12-JAN-2000  | 15      | 10000     |
| 101    | 09-MAR-2000  | 40      | 8000      |
| 102    | 09-MAR-2000  | 35      | 12500     |
| 103    | 15-MAR-2000  | 15      | 12000     |
| 104    | 25-JUN-2000  | 15      | 6000      |
| 105    | 18-JUL-2000  | 20      | 5000      |
| 106    | 18-JUL-2000  | 35      | 7000      |
| 107    | 21-JUL-2000  | 20      | 6500      |
| 108    | 04-AUG -2000 | 10      | 8000      |

**CUSTOMERS**

| CUST_ID | CUST_NAME | City          |
|---------|-----------|---------------|
| 10      | Smith     | LosAngeles    |
| 15      | Bob       | San Francisco |
| 20      | Martin    | Chicago       |
| 25      | Mary      | New York      |
| 30      | Rina      | Chicago       |
| 35      | Smith     | New York      |
| 40      | Linda     | New York      |

Which SQL statement retrieves the order ID, customer ID, and order total for the orders that are placed on the same day that Martin places his orders?

- A. 

```
SELECT ord_id, cust_id, ord_total
FROM orders, customers WHERE
cust_name='Mating'
AND ord_date IN ('18-JUL-2000','21-JUL-2000');
```
- B. 

```
SELECT ord_id, cust_id, ord_total
FROM orders
Where ord_date IN (SELECT ord_date
FROM orders
WHERE cust_id = (SELECT cust_id
FROM customers
WHERE cust_name =
'Martin'));
```
- C. 

```
SELECT ord_id, cust_id, ord_total
FROM orders
Where ord_date IN (SELECT ord_date
FROM orders, customers
Where cust_name = 'Martin');
```
- D. 

```
SELECT ord_id, cust_id, ord_total
FROM orders
WHERE cust_id IN (SELECT cust_id
FROM customers
WHERE cust name = 'Martin');
```

**Question: 101. (E)**

Examine the data from the ORDERS and CUSTOMERS tables.

**ORDERS**

| ORD_ID | ORD_DATE     | CUST_ID | ORD_TOTAL |
|--------|--------------|---------|-----------|
| 100    | 12-JAN-2000  | 15      | 10000     |
| 101    | 09-MAR-2000  | 40      | 8000      |
| 102    | 09-MAR-2000  | 35      | 12500     |
| 103    | 15-MAR-2000  | 15      | 12000.    |
| 104    | 25-JUN-2000  | 15      | 6000      |
| 105    | 18-JUL-2000  | 20      | 5000      |
| 106    | 18-JUL-2000  | 35      | 7000      |
| 107    | 21-JUL-2000  | 20      | 6500      |
| 109    | 04-AUG -2000 | 10      | 8000      |

**CUSTOMERS**

| CUST_ID | CUST_NAME | CITY          |
|---------|-----------|---------------|
| 10      | Smith     | Los Angeles   |
| 15      | Bob       | San Francisco |
| 20      | Martin    | Chicago       |
| 25      | Mary      | New York      |
| 30      | Rina      | Chicago       |
| 35      | Smith     | New York      |
| 40      | Lind      | New York      |

Evaluate the SQL statement:

```
SELECT *
FROM orders
WHERE cust_id = (SELECT cust_id
FROM customers
WHERE cust_name = 'Smith');
```

What is the result when the query is executed?



A.

| ORD_ID | ORD_DATE    | CUST_ID | ORD_TOTAL |
|--------|-------------|---------|-----------|
| 102    | 09-MAR-2000 | 35      | 12500     |
| 106    | 18-JUL-2000 | 35      | 7000      |
| 108    | 04-AUG-2000 | 10      | 8000      |

B.

| ORD_ID | ORD_DATE    | CUST_ID | ORD_TOTAL |
|--------|-------------|---------|-----------|
| 102    | 09-MAR-2000 | 35      | 12500     |
| 106    | 18-JUL-2000 | 35      | 7000      |

C.

| ORD_ID | ORD_DATE     | CUST_ID | ORD_TOTAL |
|--------|--------------|---------|-----------|
| 108    | 04-AUG -2000 | 10      | 8000      |

D. The query fails because the subquery returns more than one row.

E. The query fails because the outer query and the inner query are using different tables.

**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?

- A. 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.
- B. 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.
- C. 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.
- D. 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?

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

**Question: 104. (E)**

Evaluate this SELECT statement:

```
SELECT employee_id, name
FROM employee
WHERE employee_id NOT IN (SELECT employee_id
FROM employee
WHERE department_id = 30
AND job = 'CLERK');
```

What would happen if the inner query returned a NULL value?

- A. No rows would be selected from the EMPLOYEE table.
- B. All the EMPLOYEE\_ID and NAME values in the EMPLOYEE table would be displayed.
- C. Only the rows with EMPLOYEE\_ID values equal to NULL would be included in the results.
- D. A syntax error would be returned.

**Question: 105. (E)**

Examine the data from the DONATION table (PLEDGE\_ID is the primary key).

| PLEDGE_ID | DONOR_ID | PLEDGE_DT    | AMOUNT_PLEDGED | AMOUNT_PAID | PAYMENT_DT  |
|-----------|----------|--------------|----------------|-------------|-------------|
| 1         | 1        | 10-SEP-2001  | 1000           | 1000        | 02-OCT-2001 |
| 2         | 1        | 22-EFB -2002 | 1000           |             |             |
| 3         | 2        | 08-OCT-2001  | 10             | 10          | 28-OCT-2001 |
| 4         | 2        | 10-DEC-2001  | 50             |             |             |
| 5         | 3        | 02-NOV-2001  | 10000          | 9000        | 28-DEC-2001 |
| 6         | 3        | 05-JAN-2002  | 1000           | 1000        | 31-JAN-2002 |
| 7         | 4        | 09-NOV-2001  | 2100           | 2100        | 15-DEC-2001 |
| 8         | 5        | 09-DEC-2001  | 110            | 110         | 29-DEC-2001 |

This statement fails when executed:

```
SELECT amount_pledged, amount_paid
FROM donation
WHERE donor_id =
(SELECT donor_id
FROM donation
WHERE amount_pledged = 1000.00
OR pledge_dt = '05-JAN-2002');
```

Which two changes could correct the problem? (Choose two.)

- A. Remove the subquery WHERE clause.
- B. Change the outer query WHERE clause to 'WHERE donor\_id IN'.
- C. Change the outer query WHERE clause to 'WHERE donor\_id LIKE'.
- D. Remove the single quotes around the date value in the inner query WHERE clause.
- E. Change the subquery WHERE clause to 'WHERE amount\_pledged = 1000.00 AND pledge\_dt = '05-JAN-2002' '.
- F. Include the DONOR\_ID column in the select list of the outer query.

**Question: 106. (E)**

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

- A. Create groups of data.
- B. Sort data in a specific order
- C. Convert data to a different format
- D. Retrieve data based on an unknown condition

**Question: 107. (E)**

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

- A. A single row subquery can retrieve only one column and one row.
- B. A single row subquery can retrieve only one row but many columns.
- C. A multiple row subquery can retrieve multiple rows and multiple columns.
- D. A multiple row subquery can be compared by using the ">" operator.
- E. A single row subquery can use the IN operator.
- F. 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?

- A. The main query executes with the first value returned by the subquery.
- B. The main query executes with the last value returned by the subquery.
- C. The main query executes with all the values returned by the subquery.
- D. The main query fails because the multiple-row subquery cannot be used with the comparison operator
- E. 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?

- A. =
- B. LIKE
- C. BETWEEN
- D. NOT IN
- E. IS
- F. <>

**Question: 110. (E)**

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

- A. A single row subquery can retrieve data from only one table.
- B. 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.
- C. 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 cannot be used in a condition where the LIKE operator is used for comparison.
- E. A multiple-row subquery cannot be used in a condition where the LIKE operator is used for comparison.

**Question: 111. (E)**

Examine the data from the EMP table:

| EMP_ID | DEPT_ID | COMMISSION |
|--------|---------|------------|
| 1      | 10      | 500        |
| 2      | 20      | 1000       |
| 3      | 10      |            |
| 4      | 10      | 600        |
| 5      | 30      | 800        |
| 6      | 30      | 200        |
| 7      | 10      |            |
| 8      | 20      | 300        |

The COMMISSION column shows the monthly commission earned by the employee.

Which three tasks would require subqueries or joins in order to perform in a single step? (Choose three)

- A. Deleting the records of employees who do not earn commission.
- B. Increasing the commission of employee 3 by the average commission earned in department 20.

- C. Finding the number of employees who do NOT earn commission and are working for department 20.
- D. Inserting into the table a new employee 10 who works for department 20 and earns a commission that is equal to the commission earned by employee 3.
- E. Creating a table called COMMISSION that has the same structure and data as the columns EMP\_ID and COMMISSIONS of the EMP table.
- F. Decreasing the commission by 150 for the employees who are working in department 30 and earning a commission of more than 800.

**Question: 112. (E)**

Which substitution variable would you use if you want to reuse the variable without prompting the user each time?

- A. &
- B. ACCEPT
- C. PROMPT
- D. &&

**Question: 113. (E)**

Which SQL statement accepts user input for the columns to be displayed, the table name, and WHERE condition?

- A. SELECT &1, "&2"  
FROM &3  
WHERE last\_name = '&4';
- B. SELECT &1, '&2'  
FROM &3  
WHERE '&last\_name = '&4' ';
- C. SELECT &1, &2  
FROM &3  
WHERE last\_name = '&4';
- D. SELECT &1, '&2'  
FROM EMP  
WHERE last\_name = '&4';

**Question: 114. (E)**

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

EMPLOYEES:

|             |               |             |
|-------------|---------------|-------------|
| EMPLOYEE_ID | NUMBER        | Primary Key |
| FIRST_NAME  | VARCHAR2 (25) |             |
| LAST_NAME   | VARCHAR2 (25) |             |
| HIRE_DATE   | DATE          |             |

NEW\_EMPLOYEES:

|             |               |             |
|-------------|---------------|-------------|
| EMPLOYEE_ID | NUMBER        | Primary Key |
| NAME        | VARCHAR2 (60) |             |

Which DELETE statement is valid?

- A. DELETE FROM employees  
WHERE employee\_id = (SELECT employee\_id  
FROM employees);
- B. DELETE \* FROM employees  
WHERE employee\_id = (SELECT employee\_id  
FROM new\_employees);
- C. DELETE FROM employees  
WHERE employee\_id IN (SELECT employee\_id  
FROM new\_employees  
WHERE name = 'Carrey');
- D. DELETE \* FROM employees

```
WHERE employee_id IN (SELECT employee_id
FROM new_employees
WHERE last_name = 'Carrey');
```

**Question: 115. (E)**

Evaluate this SQL\*Plus command:

```
COLUMN teacher_name HEADING 'Teacher' FORMAT A25
```

Which two tasks will this command accomplish? (Choose two.)

- A. It will set the TEACHER\_NAME column heading to 'Teacher'.
- B. It will center the column heading of the TEACHER\_NAME column.
- C. It will limit the TEACHER\_NAME column heading to 25 characters.
- D. It will display the current settings for the TEACHER\_NAME column.
- E. It will set the display width of the TEACHER\_NAME column to 25.

**Question: 116. (E)**

The STUDENT table contains these columns:

|             |              |    |
|-------------|--------------|----|
| ID          | NUMBER(9)    | PK |
| LAST_NAME   | VARCHAR2(25) |    |
| FIRST_NAME  | VARCHAR2(25) |    |
| STU_TYPE    | VARCHAR2(1)  |    |
| ENROLL_DATE | DATE         |    |

Evaluate this iSQL\*Plus script:

```
SELECT id, last_name, first_name, enroll_date
FROM student
WHERE stu_type = '&stype'
AND enroll_date between &&bdate and &&edate
/
```

What happens when this script is run a second time in the same session?

- A. The user is prompted for all three substitution variables.
- B. An error occurs because the STYPE substitution variable is NOT correctly specified.
- C. The values of BDATE and EDATE from the previous execution are used.
- D. An error occurs because the BDATE and EDATE substitution variables are NOT correctly specified.

**Question: 117. (E)**

Evaluate this SELECT statement:

```
SELECT order_num, &order_date
FROM &&ordertbl
WHERE order_date = '&order_date';
```

Which statement regarding the execution of this statement is true?

- A. The user will be prompted for all values in the select list each time the statement is executed in a session.
- B. The user will be prompted for the table name each time the statement is executed in a session.
- C. An error will occur when executing this statement because substitution variables are NOT allowed in a WHERE clause.
- D. The user will be prompted for the table name only the first time the statement is executed in a session.
- E. An error will occur when executing this statement because substitution variables must be unique within a SELECT statement.

**Question: 118. (E)**

The INVENTORY table contains these columns:

|             |              |    |
|-------------|--------------|----|
| ID_NUMBER   | NUMBER       | PK |
| DESCRIPTION | VARCHAR2(30) |    |

SUPPLIER\_ID          NUMBER

You want to create a query that for each session allows the user to input a value for DESCRIPTION each time the query runs. While the DESCRIPTION column is stored in upper case, you want the query to retrieve matching values regardless of the case used when inputting the substitution variable value.

Which SELECT statement should you use?

- A. SELECT id\_number, supplier\_id  
FROM inventory  
WHERE description = UPPER(&description);
- B. SELECT id\_number, supplier\_id  
FROM inventory  
WHERE LOWER(description) = LOWER('&description');
- C. SELECT id\_number, supplier\_id  
FROM inventory  
WHERE LOWER(description) = '&description';
- D. SELECT id\_number, supplier\_id  
FROM inventory  
WHERE description = UPPER('&&description');

**Question: 119. (E)**

In which clauses of a SELECT statement can substitution variables be used?

- A. the SELECT, WHERE, GROUP BY, and ORDER BY clauses, but NOT the FROM clause
- B. the SELECT, FROM, WHERE, and GROUP BY clauses, but NOT the ORDER BY clause
- C. the SELECT and FROM clauses, but NOT the WHERE clause
- D. the SELECT, FROM, and WHERE clauses only
- E. the SELECT, FROM, WHERE, GROUP BY, ORDER BY, and HAVING clauses

**Question: 120. (E)**

What is the default character for specifying substitution variable in select statement?

- A. Ampersand.
- B. Ellipses.
- C. Quotations marks.
- D. Asterik

**Question: 121.(F)**

Which four are correct guidelines for naming database tables? (Choose four)

- A. Must begin with either a number or a letter.
- B. Must be 1-30 characters long.
- C. Should not be an Oracle Server reserved word.
- D. Must contain only A-Z, a-z, 0-+, \_, \*, and #.
- E. Must contain only A-Z, a-z, 0-9, \_, \$, and #.
- F. Must begin with a letter.

**Question: 122. (F)**

Examine the structure of the EMPLOYEES table:

| Column name   | Data type    | Remarks                       |
|---------------|--------------|-------------------------------|
| EMPLOYEE_ID   | NUMBER       | NOT NULL, Primary Key         |
| LAST_NAME     | VARCNAR2(30) |                               |
| FIRST_NAME    | VARCNAR2(30) |                               |
| JOB_ID        | NUMBER       |                               |
| SAL           | NUMBER       |                               |
| MGR_ID        | NUMBER       | References EMPLOYEE_ID column |
| DEPARTMENT_ID | NUMBER       |                               |

You need to create an index called NAME\_IDX on the first name and last name fields of the EMPLOYEES table. Which SQL statement would you use to perform this task?

- A. CREATE INDEX NAME\_IDX (first\_name, last\_name);
- B. CREATE INDEX NAME\_IDX (first\_name AND last\_name);
- C. CREATE INDEX NAME\_IDX  
ON (first\_name, last\_name);
- D. CREATE INDEX NAME\_IDX  
ON employees (first\_name AND last\_name);
- E. CREATE INDEX NAME\_IDX  
ON employees(first\_name, last\_name);
- F. CREATE INDEX NAME\_IDX  
FOR employees(first\_name, last\_name);

**Question: 123. (F)**

Which three are DATETIME data types that can be used when specifying column definitions?  
(Choose three.)

- A. TIMESTAMP
- B. INTERVAL MONTH TO DAY
- C. INTERVAL DAY TO SECOND
- D. INTERVAL YEAR TO MONTH
- E. TIMESTAMP WITH DATABASE TIMEZONE

**Question: 124. (F)**

What does the TRUNCATE statement do?

- A. Removes the table
- B. Removes all rows from a table
- C. Shortens the table to 10 rows
- D. Removes all columns from a table
- E. Removes foreign keys from a table

**Question: 125. (F)**

Which statement about a table is true?

- A. A table can have up to 10,000 columns.
- B. The size of a table does NOT need to be specified.
- C. A table CANNOT be created while users are using the database.
- D. The structure of a table CANNOT be modified while the table is online.

**Question: 126. (F)**

The ACCOUNT table contains these columns:

|                |             |
|----------------|-------------|
| ACCOUNT_ID     | NUMBER(12)  |
| FINANCE_CHARGE | NUMBER(7,2) |
| PREV_BALANCE   | NUMBER(7,2) |
| PAYMENTS       | NUMBER(7,2) |
| NEW_PURCHASES  | NUMBER(7,2) |

You created the ACCOUNT\_ID\_SEQ sequence to generate sequential values for the ACCOUNT\_ID column.

You issue this statement:

```
ALTER TABLE account  
MODIFY (finance_charge NUMBER(8,2));
```

Which statement about the ACCOUNT\_ID\_SEQ sequence is true?

- A. The sequence is dropped.
- B. The precision of the sequence is changed.
- C. The sequence is reverted to its minimum value.

D. The sequence is unchanged.

**Question: 127. (F)**

Examine the structure of the EMPLOYEE table.

**EMPLOYEE Table**

|             |               |                                                       |
|-------------|---------------|-------------------------------------------------------|
| EMPLOYEE_ID | NUMBER        | NOT NULL, Primary Key                                 |
| EMP_LNAME   | VARCHAR2 (25) |                                                       |
| EMP_FNAME   | VARCHAR2 (25) |                                                       |
| DEPT_ID     | NUMBER        | Foreign key to DEPT_ID column of the DEPARTMENT table |
| JOB_ID      | NUMBER        | Foreign key to JOB_ID column of the JOB table         |
| MGR_ID      | NUMBER        | References EMPLOYEE_ID column                         |
| SALARY      | NUMBER (9,2)  |                                                       |
| HIRE_DATE   | DATE          |                                                       |
| DPB         | DATE          |                                                       |

Which statement should you use to increase the EMP\_LNAME column length to 25 if the column currently contains 3000 records?

- A. You CANNOT increase the width of the EMP\_LNAME column.
- B. ALTER TABLE employee  
MODIFY emp\_lname VARCHAR2(25);
- C. ALTER TABLE employee  
RENAME emp\_lname VARCHAR2(25);
- D. ALTER employee TABLE  
MODIFY COLUMN emp\_lname VARCHAR2(25);
- E. ALTER employee TABLE  
MODIFY COLUMN (emp\_lname VARCHAR2(25));

**Question: 128. (F)**

Examine the structure of the PO\_DETAIL table.

**PO\_DETAIL Table**

| PO-NUM | PO_LINE_IN | PRODUCT_ID | QUANTITY | UNIT_PRICE |
|--------|------------|------------|----------|------------|
| 10052  | 1          | 1          | 100      | 10.30      |
| 10052  | 2          | 2          | 100      | 10.00      |
| 10054  | 1          | 1          | 50       | 72.00      |
| 10054  | 2          | 2          | 10       | 10.00      |
| 10054  | 3          | 3          | 10       | 10.00      |

Which statement will permanently remove all the data in, the indexes on, and the structure of the PO\_DETAIL table?

- A. DELETE TABLE po\_detail;
- B. TRUNCATE TABLE po\_detail;
- C. ALTER TABLE po\_detail  
SET UNUSED (po\_num, po\_line\_id, product\_id, quantity, unit\_price);
- D. DROP TABLE po\_detail;

**Question: 129.**

Examine the structure of the EMPLOYEE table.

**EMPLOYEE Table**

|             |               |                                                       |
|-------------|---------------|-------------------------------------------------------|
| EMPLOYEE_ID | NUMBER        | NOT NULL, Primary Key                                 |
| EMP_LNAME   | VARCHAR2 (25) |                                                       |
| EMP_FNAME   | VARCHAR2 (25) |                                                       |
| DEPT_ID     | NUMBER        | Foreign key to DEPT_ID column of the DEPARTMENT table |
| JOB_ID      | NUMBER        | Foreign key to JOB_ID column of the JOB table         |
| MGR_ID      | NUMBER        | References EMPLOYEE_ID column                         |



|           |              |  |
|-----------|--------------|--|
| SALARY    | NUMBER (9,2) |  |
| HIRE_DATE | DATE         |  |
| DPB       | DATE         |  |

The MGR\_ID column currently contains employee identification numbers, and you need to allow users to include text characters in the identification values. Which statement should you use to implement this?

- A. ALTER employee  
MODIFY (mgr\_id VARCHAR2(15));
- B. ALTER TABLE employee  
MODIFY (mgr\_id VARCHAR2(15));
- C. ALTER employee TABLE  
MODIFY COLUMN (mgr\_id VARCHAR2(15));
- D. ALTER TABLE employee  
REPLACE (mgr\_id VARCHAR2(15));
- E. You CANNOT modify the data type of the MGR\_ID column.

**Question: 130. (F)**

Which CREATE TABLE statements will fail? (Choose all that apply.)

- A. CREATE TABLE time1 (time1 NUMBER(9));
- B. CREATE TABLE date (time\_id NUMBER(9));
- C. CREATE TABLE time (time\_id NUMBER(9));
- D. CREATE TABLE time\* (time\_id NUMBER(9));
- E. CREATE TABLE \$time (time\_id NUMBER(9));
- F. CREATE TABLE datetime (time\_id NUMBER(9));

**Question: 131. (F)**

Evaluate this statement:

```
SELECT *
FROM greg.customer;
```

If as user Chan you execute this statement successfully, which statement must be true?

- A. The CUSTOMER table exists in user Greg's schema.
- B. The CUSTOMER table was created in the DBA schema.
- C. The CUSTOMER table was created in the public schema.
- D. The CUSTOMER table is accessible to all users.

**Question: 132. (F)**

Logged on as user Ann, you issue these SQL statements:

```
CREATE TABLE customer (
customer_id      NUMBER,
company_id      VARCHAR2(30),
contact_name     VARCHAR2(30),
contact_title    VARCHAR2(20),
address         VARCHAR2(30),
city            VARCHAR2(25),
region          VARCHAR2(10),
postal_code     VARCHAR2(20),
country_id      NUMBER,
phone           VARCHAR2(20),
fax             VARCHAR2(20),
credit_limit     NUMBER (7,2));
```

```
CREATE PUBLIC SYNONYM cust
FOR customer;
```

```
GRANT SELECT ON customer TO chan;
```

Which SELECT statements could user Chan issue to access the CUSTOMER table? (Choose all that apply.)

- A. SELECT \*  
FROM ann.customer;
- B. SELECT customer\_id  
FROM customer;
- C. SELECT customer\_name, phone  
FROM cust;
- D. SELECT \*  
FROM ann.cust;
- E. SELECT COUNT(city)  
FROM chan.cust;

**Question: 133. (F)**

Evaluate this CREATE TABLE statement:

```
CREATE TABLE curr_order (
```

```
id                NUMBER,  
customer_id       NUMBER,  
emp_id            NUMBER,  
order_dt          TIMESTAMP WITH LOCAL TIME ZONE,  
order_amt         NUMBER(7,2),  
ship_method       VARCHAR2(5));
```

Which statement about the ORDER\_DT column is true?

- A. Data will be normalized to the database time zone.
- B. Data will include a time zone displacement in its value.
- C. Data stored in the column will be returned in the server's local time zone.
- D. Data will be stored using a fractional seconds precision of 3.

**Question: 134. (F)**

Examine the structure of the PRODUCT table.

|               |               |                                                  |
|---------------|---------------|--------------------------------------------------|
| PRODUCT_ID    | NUMBER        | NOT NULL, Primary Key                            |
| PRODUCT_NAME  | VARCHAR2 (25) |                                                  |
| SUPPLIER_ID   | NUMBER        | Foreign key to SUPPLIER_ID of the SUPPLIER table |
| LIST_PRICE    | NUMBER (7,2)  |                                                  |
| COST          | NUMBER (7,2)  |                                                  |
| QTY_IN_STOCK  | NUMBER        |                                                  |
| QTY_ON_ORDER  | NUMBER        |                                                  |
| REORDER_LEVEL | NUMBER        |                                                  |
| REORDER_QTY   | NUMBER        |                                                  |

You need to reduce the LIST\_PRICE column precision to 6 with a scale of 2 and ensure that when inserting a row into the PRODUCT table without a value for the LIST\_PRICE column, a price of \$5.00 will automatically be inserted. The PRODUCT table currently contains no records.

Which statement should you use?

- A. ALTER TABLE product  
ADD OR REPLACE (list\_price NUMBER(8,2) DEFAULT 5);
- B. ALTER TABLE product  
MODIFY (list\_price NUMBER(6,2) DEFAULT 5);
- C. ALTER TABLE product  
MODIFY COLUMN (list\_price NUMBER(6,2) DEFAULT '\$5.00');
- D. ALTER TABLE product

MODIFY (list\_price NUMBER(8,2) DEFAULT 5)  
REPLACE COLUMN (list\_price NUMBER(6,2);

E. You CANNOT reduce the size of the LIST\_PRICE column.

**Question: 135. (F)**

Which statements about data types are true? (Choose all that apply.)

- A. The TIMESTAMP data type is an extension of the VARCHAR2 data type.
- B. The BLOB data type stores character data up to four gigabytes.
- C. The VARCHAR2 data type requires that a minimum size be specified when defining a column of this type.
- D. The CHAR datatype should be used for fixed-length character data.
- E. The INTERVAL YEAR TO MONTH data type allows time to be stored as an interval of years and months.

**Question: 136. (F)**

Which ALTER TABLE statement should you use to add a PRIMARY KEY constraint on the MANUFACTURER\_ID column of the INVENTORY table?

- A. ALTER TABLE inventory  
ADD CONSTRAINT manufacturer\_id PRIMARY KEY;
- B. ALTER TABLE inventory  
ADD PRIMARY KEY (manufacturer\_id);
- C. ALTER TABLE inventory  
MODIFY manufacturer\_id CONSTRAINT PRIMARY KEY;
- D. ALTER TABLE inventory  
MODIFY CONSTRAINT PRIMARY KEY manufacturer\_id;

**Question: 137. (F)**

The EMPLOYEES table has these columns:

|           |              |
|-----------|--------------|
| LAST NAME | VARCHAR2(35) |
| SALARY    | NUMBER(8,2)  |
| HIRE_DATE | DATE         |

Management wants to add a default value to the SALARY column. You plan to alter the table by using this SQL statement:

```
ALTER TABLE EMPLOYEES  
MODIFY (SALARY DEFAULT 5000);
```

What is true about your ALTER statement?

- A. Column definitions cannot be altered to add DEFAULT values.
- B. A change to the DEFAULT value affects only subsequent insertions to the table.
- C. Column definitions cannot be altered to add DEFAULT values for columns with a NUMBER data type.
- D. All the rows that have a NULL value for the SALARY column will be updated with the value 5000.

**Question: 138. (F)**

You need to change the definition of an existing table. The COMMERCIALS table needs its DESCRIPTION column changed to hold varying length characters up to 2000 bytes. The column can currently hold 1000 bytes per value. The table contains 20000 rows.

Which statement is valid?

- A. ALTER TABLE commercials MODIFY  
(description CHAR2(2000));

- B. ALTER TABLE commercials CHANGE (description CHAR2(2000));
- C. ALTER TABLE commercials CHANGE (description VARCHAR2(2000));
- D. ALTER TABLE commercials MODIFY (description VARCHAR2(2000));
- E. You cannot increase the size of a column if the table has rows.

**Question: 139. (F)**

Evaluate the SQL statement

DROP TABLE DEPT;

Which four statements are true of the SQL statement? (Choose four)

- A. You cannot roll back this statement.
- B. All pending transactions are committed.
- C. All views based on the DEPT table are deleted.
- D. All indexes based on the DEPT table are dropped.
- E. All data in the table is deleted, and the table structure is also deleted.
- F. All data in the table is deleted, but the structure of the table is retained.
- G. All synonyms based on the DEPT table are deleted.

**Question: 140. (F)**

Which statement describes the ROWID data type?

- A. Binary data up to 4 gigabytes.
- B. Character data up to 4 gigabytes.
- C. Raw binary data of variable length up to 2 gigabytes.
- D. Binary data stored in an external file, up to 4 gigabytes.
- E. A hexadecimal string representing the unique address of a row in its table.

**Question: 141. (F)**

You just issued the following statement.

Alter table marketing

Drop column profit;

Which of the following choices identified when the column will actually be removed from database.

- A. Immediately following statement execution.
- B. After the Alter table drop unused columns command is issued.
- C. After the Alter table set unused column command is issued.
- D. After the Alter table modify command is issued.

**Question: 142. (F)**

Which of the following can be a valid column name?

- A. Number
- B. 1966\_Invoices
- C. Catch\_#22
- D. #Invoices
- E. None of the above

**Question: 143.(G)**

You need to modify the STUDENTS table to add a primary key on the STUDENT\_ID column. The table is currently empty.

Which statement accomplishes this task?

- A. ALTER TABLE students  
ADD PRIMARY KEY student\_id;

- B. ALTER TABLE students  
ADD CONSTRAINT PRIMARY KEY (student\_id);
- C. ALTER TABLE students  
ADD CONSTRAINT stud\_id\_pk PRIMARY KEY student\_id;
- D. ALTER TABLE students  
ADD CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);
- E. ALTER TABLE students  
MODIFY CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

**Question: 144. (G)**

Which SQL statement defines the FOREIGN KEY constraint on the DEPTNO column of the EMP table?

- A. CREATE TABLE EMP (empno  
NUMBER(4), ename  
VARCNAR2(35), deptno  
NUMBER(7,2) NOT NULL  
CONSTRAINT emp\_deptno\_fk FOREIGN KEY deptno  
REFERENCES dept deptno);
- B. CREATE TABLE EMP  
(empno NUMBER(4),  
ename VARCHNAR2(35),  
deptno NUMBER(7,2)  
CONSTRAINT emp\_deptno\_fk REFERENCES dept (deptno));
- C. CREATE TABLE EMP  
(empno NUMBER(4)  
ename VARCHAR2(35),  
deptno NUMBER(7,2) NOT NULL,  
CONSTRAINT emp\_deptno\_fk REFERENCES dept (deptno)  
FOREIGN KEY (deptno));
- D. CREATE TABLE EMP (empno NUMBER(4),  
ename VARCHNAR2(35),  
deptno NUMBER(7,2) FOREIGN KEY  
CONSTRAINT emp\_deptno\_fk REFERENCES dept (deptno));

**Question: 145. (G)**

Which three statements correctly describe the functions and use of constraints? (Choose three.)

- A. Constraints provide data independence.
- B. Constraints make complex queries easy.
- C. Constraints enforce rules at the view level.
- D. Constraints enforce rules at the table level.
- E. Constraints prevent the deletion of a table if there are dependencies.
- F. Constraints prevent the deletion of an index if there are dependencies.

**Question: 146. (G)**

Which two statements about NOT NULL constraints are true? (Choose two.)

- A. NOT NULL constraints can only be defined at the column level.
- B. You CANNOT define a NOT NULL column if the column does NOT have a non-null value for every row.
- C. You CANNOT add a NOT NULL constraint to an existing column using the ALTER TABLE statement.
- D. You can modify the structure of a NOT NULL constraint using the ALTER TABLE statement.
- E. A NOT NULL constraint is stored in the data dictionary as a UNIQUE constraint.

**Question: 147. (G)**

The PO\_DETAIL table contains these columns:

PO\_NUM                      NUMBER NOT NULL, Primary Key

|            |                                                              |
|------------|--------------------------------------------------------------|
| PO_LINE_ID | NUMBER NOT NULL, Primary Key                                 |
| PRODUCT_ID | NUMBER Foreign Key to PRODUCT_ID column of the PRODUCT table |
| QUANTITY   | NUMBER                                                       |
| UNIT_PRICE | NUMBER(5,2)                                                  |

Evaluate this statement:

```
ALTER TABLE po_detail
ENABLE CONSTRAINT po_num_pk;
```

For which task would you issue this statement?

- A. to drop and recreate the PRIMARY KEY constraint on the PO\_NUM column
- B. to activate the previously disabled constraint on the PO\_NUM column while creating a PRIMARY KEY index
- C. to create a new PRIMARY KEY constraint on the PO\_NUM column
- D. to enable any previously disabled FOREIGN KEY constraints that are dependent on the PO\_NUM column

**Question: 148. (G)**

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

- A. Constraints only enforce rules at the table level.
- B. Constraints prevent a table with dependencies from being deleted.
- C. Constraints can be created at the same time as the table or after the table is created.
- D. You must provide a name for each constraint at the time of its creation.
- E. Constraint names are NOT required to follow the standard object-naming rules.

**Question: 149. (G)**

Examine the structure of the EMPLOYEE table.

|             |                  |                                                       |
|-------------|------------------|-------------------------------------------------------|
| EMPLOYEE_ID | NUMBER           | NOT NULL, Primary Key                                 |
| EMP_LNAME   | VARCHAR2<br>(25) |                                                       |
| EMP_FNAME   | VARCHAR2<br>(25) |                                                       |
| DEPT_ID     | NUMBER           | Foreign key to DEPT_ID column of the DEPARTMENT table |
| JOB_ID      | NUMBER           | Foreign key to JOB_ID column of the JOB table         |
| MGR_ID      | NUMBER           | References EMPLOYEE_ID column                         |
| SALARY      | NUMBER (9,2)     |                                                       |
| HIRE_DATE   | DATE             |                                                       |

Which CREATE TABLE statement should you use to create the EMPLOYEE table?

- A. 

```
CREATE TABLE employee
( employee_id    NUMBER,
  emp_lname VARCHAR2(25),
  emp_fname VARCHAR2(25),
  dept_id NUMBER,
  job_id NUMBER,
  mgr_id NUMBER,
  salary NUMBER(9,2),
  hire_date DATE,
  CONSTRAINT employee_id_pk PRIMARY KEY(employee_id),
  CONSTRAINT dept_id_fk FOREIGN KEY(dept_id) REFERENCES department(dept_id),
  CONSTRAINT job_id_fk FOREIGN KEY(job_id) REFERENCES job(job_id),
  CONSTRAINT mgr_id_fk FOREIGN KEY(mgr_id) REFERENCES employee(emp_id));
```

- B. CREATE TABLE employee  
 ( employee\_id NUMBER,  
 emp\_lname VARCHAR2(25),  
 emp\_fname VARCHAR2(25),  
 dept\_id NUMBER,  
 job\_id NUMBER,  
 mgr\_id NUMBER,  
 salary NUMBER(9,2),  
 hire\_date DATE,  
 CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id));
- C. CREATE TABLE employee  
 ( employee\_id NUMBER,  
 emp\_lname VARCHAR2(25),  
 emp\_fname VARCHAR2(25),  
 dept\_id NUMBER,  
 job\_id NUMBER,  
 mgr\_id NUMBER,  
 salary NUMBER(9,2),  
 hire\_date DATE,  
 CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),  
 CONSTRAINT dept\_id\_fk FOREIGN KEY(dept\_id) REFERENCES department(dept\_id),  
 CONSTRAINT job\_id\_fk FOREIGN KEY(job\_id) REFERENCES job(job\_id));
- D. CREATE TABLE employee  
 ( employee\_id NUMBER,  
 emp\_lname VARCHAR2(25),  
 emp\_fname VARCHAR2(25),  
 dept\_id NUMBER,  
 job\_id NUMBER,  
 mgr\_id NUMBER,  
 salary NUMBER(9,2),  
 hire\_date DATE,  
 CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),  
 CONSTRAINT mgr\_id\_fk FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id));

**Question: 150. (G)**

Which syntax turns an existing constraint on?

- A. ALTER TABLE table\_name  
 ENABLE constraint\_name;
- B. ALTER TABLE table\_name  
 STATUS = ENABLE CONSTRAINT constraint\_name;
- C. c) ALTER TABLE table\_name  
 ENABLE CONSTRAINT constraint\_name;
- D. d) ALTER TABLE table\_name  
 STATUS ENABLE CONSTRAINT constraint\_name;
- E. e) ALTER TABLE table\_name  
 TURN ON CONSTRAINT constraint\_name;
- F. f) ALTER TABLE table\_name  
 TURN ON CONSTRAINT constraint\_name;

**Question: 151. (G)**

Which two statements about creating constraints are true? (Choose two)

- A. Constraint names must start with SYS\_C.
- B. All constraints must be defined at the column level.
- C. Constraints can be created after the table is created.
- D. Constraints can be created at the same time the table is created.
- E. Information about constraints is found in the VIEW\_CONSTRAINTS dictionary view.

**Question: 152. (G)**

Which constraint can be defines only at the column level?

- A. UNIQUE
- B. NOT NULL
- C. CHECK
- D. PRIMARY KEY
- E. FOREIGN KEY

**Question: 153. (G)**

Which statement explicitly names a constraint?

- A. ALTER TABLE student\_grades  
ADD  
FOREIGN KEY (student\_id) REFERENCES students(student\_id);
- B. ALTER TABLE student\_grades  
ADD CONSTRAINT NAME = student\_id\_fk  
FOREIGN KEY (student\_id) REFERENCES students(student\_id);
- C. ALTER TABLE student\_grades  
ADD CONSTRAINT  
student\_id\_fk  
FOREIGN KEY (student\_id) REFERENCES students(student\_id);
- D. ALTER TABLE student grades  
ADD NAMED CONSTRAINT student\_id\_fk  
FOREIGN KEY (student\_id) REFERENCES students(student\_id);
- E. ALTER TABLE student grades ADD NAME student\_id\_fk  
FOREIGN KEY (student\_id) REFERENCES students(student\_id);

**Question: 154. (G)**

Examine the SQL statements that creates ORDERS table:

```
CREATE TABLE orders
(SER_NO          NUMBER          UNIQUE,
ORDER_ID         NUMBER,
ORDER_DATE       DATE            NOT NULL
STATUS           VARCHAR2(10)
CHECK (status IN ('CREDIT','CASH')),
PROD_ID_NUMBER   REFERENCES PRODUCTS(PRODUCT_ID),
ORD_TOTAL        NUMBER,
PRIMARY KEY (order id, order date));
```

For which columns would an index be automatically created when you execute the above SQL statement? (Choose two)

- a) SER\_NO
- b) ORDER\_ID
- c) STATUS
- d) PROD\_ID.
- e) ORD\_TOTAL
- f) Composite index on ORDER\_ID and ORDER\_DATE

**Question: 155. (G)**

For which two constraints does the Oracle Server implicitly create a unique index?  
(Choose two.)

- A. NOT NULL.
- B. PRIMARY KEY
- C. FOREIGN KEY
- D. CHECK
- E. UNIQUE



**Question: 156. (G)**

Your attempt to disable a constraints result in the following error:

Ora:02297: cannot disable constraint – dependencies exist.

Which of the following types of the constraints is likely causing interference with your disablement of this one?

- a) Check constraints.
- b) Not NULL constraints.
- c) Foreign key Constraints.
- d) Unique Constraints.

**Question: 157.(H)**

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

**EMPLOYEES**

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| HIRE_DATE   | DATE         |             |

**NEW\_EMPLOYEES**

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| NAME        | VARCHAR2(60) |             |

Which UPDATE statement is valid?

- A. UPDATE new\_employees SET name = (Select last\_name||  
first\_name  
FROM employees  
Where employee\_id =180)  
WHERE employee\_id =180;
- B. UPDATE new\_employees SET name = (SELECT  
last\_name||first\_name  
FROM employees) WHERE  
employee\_id =180;
- C. UPDATE new\_employees SET name = (SELECT last\_name||  
first\_name  
FROM employees  
WHERE employee\_id =180)  
WHERE employee\_id =(SELECT employee\_id  
FROM new employees);
- D. UPDATE new\_employees SET name = (SELECT  
last name|| first\_name  
FROM employees WHERE  
employee\_id= (SELECT  
employee\_id FROM  
new\_employees)) WHERE  
employee\_id =180;

**Question: 158. (H)**

Examine the structure if the EMPLOYEES and NEW\_EMPLOYEES tables:

**EMPLOYEES**

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| HIRE_DATE   | DATE         |             |

**NEW EMPLOYEES**

|             |                    |
|-------------|--------------------|
| EMPLOYEE_ID | NUMBER Primary Key |
| NAME        | VARCHAR2(60)       |

Which MERGE statement is valid?

- A.       MERGE INTO  
new\_employees c USING  
employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN MATCHED THEN  
UPDATE SET  
c.name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT VALUES(e.employee\_id, e.first\_name || ', ' ||  
e.last\_name);
- B.       MERGE  
new\_employees c USING  
employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN EXIST THEN  
UPDATE SET  
c.name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT VALUES(e.employee\_id, e.first\_name || ', ' ||  
e.last\_name);
- C.       MERGE INTO new  
employees c USING employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN EXISTS THEN  
UPDATE SET  
c.name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT VALUES(e.employee\_id, e.first\_name || ', ' ||  
e.last\_name);
- D.       MERGE  
new\_employees c FROM  
employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN MATCHED THEN  
UPDATE SET  
c.name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT INTO new\_employees VALUES(e.employee\_id, e.first\_name || ', ' ||  
e.last\_name);

**Question: 159. (H)**

Examine the data in the EMPLOYEES and EMP\_HIST tables:

**EMPLOYEES**

| EMPLOYEE_ID | NAME     | DEPT_ID | MGR_ID | JOB_ID   | SALARY |
|-------------|----------|---------|--------|----------|--------|
| 101         | Smith    | 20      | 120    | SA_REP   | 4000   |
| 102         | Martin   | 10      | 105    | CLERK    | 2500   |
| 103         | Chris    | 20      | 120    | IT_ADMIN | 4200   |
| 104         | John     | 30      | 108    | HR_CLERK | 2500   |
| 105         | Diana    | 30      | 108    | IT_ADMIN | 5000   |
| 106         | Smith    | 40      | 110    | AD_ASST  | 3000   |
| 108         | Jennifer | 30      | 110    | HR_DIR   | 6500   |
| 110         | Bob      | 40      |        | EX_DIR   | 8000   |

|     |      |    |     |        |      |
|-----|------|----|-----|--------|------|
| 120 | Ravi | 20 | 110 | SA_DIR | 6500 |
|-----|------|----|-----|--------|------|

### EMP\_HIST

| EMPLOYEE_ID | NAME     | JOB_ID   | SALARY |
|-------------|----------|----------|--------|
| 101         | Smith    | SA_CLERK | 2000   |
| 103         | Chris    | IT_CLERK | 2200   |
| 104         | John     | HR_CLERK | 2000   |
| 106         | Smith    | AD_ASST  | 3000   |
| 108         | Jennifer | HR_MGR   | 4500   |

The EMP\_HIST table is updated at the end of every year. The employee ID, name, job ID, and salary of each existing employee are modified with the latest data. New employee details are added to the table.

Which statement accomplishes this task?

- A.       UPDA TE emp\_hist  
SET employee\_id, name, job\_id, salary =  
(SELECT employee\_id, name, job\_id, salary  
FROM employees)  
WHERE employee\_id IN  
(SELECT employee\_id  
FROM employees);
- B.       MERGE INTO  
emp\_hist eh USING  
employees e  
ON (eh.employee\_id = e.employee\_id)  
WHEN MATCHED THEN  
UPDATE SET eh.name = e.name,  
eh.job\_id = e.job\_id,  
eh.salary = e.salary  
WHEN NOT MATCHED THEN  
INSERT VALUES (e.employee\_id, e.name,  
e.job\_id, e.salary);
- C.       MERGE INTO  
emp\_hist eh USING  
employees e  
ON (eh.employee\_id = e.employee\_id)  
WHEN MATCHED THEN  
UPDATE emp\_hist SET eh.name = e.name,  
eh.job\_id = e.job\_id,  
eh.salary = e.salary  
WHEN NOT MATCHED THEN  
INSERT INTO emp\_hist  
VALUES (e.employee\_id, e.name, e.job\_id, e.salary);
- D.       MERGE INTO  
emp\_hist eh USING  
employees e.  
WHEN MATCHED THEN  
UPDATE emp\_hist SET eh.name = e.name,  
eh.job\_id = e.job\_id,  
eh.salary = e.salary  
WHEN NOT MATCHED THEN  
INSERT INTO emp\_hist  
VALUES (e.employee\_id, e.name, e.job\_id, e.salary);

### Question: 160. (H)

Which are DML statements? (Choose all that apply)

- a.       COMMIT...
- b.       MERGE...

- c. UPDATE...
- d. DELETE...
- e. CREATE...
- f.DROP...

**Question: 161. (H)**

Examine the data in the EMPLOYEES and DEPARTMENTS tables:

**EMPLOYEES**

| EMPLOYEE_ID | EMP_NAME | DEPT_ID | MGR_ID | JOB_ID   | SALARY |
|-------------|----------|---------|--------|----------|--------|
| 101         | Smith    | 20      | 120    | SA_REP   | 4000   |
| 102         | Martin   | 10      | 105    | CLERK    | 2500   |
| 103         | Chris    | 20      | 120    | IT_ADMIN | 4200   |
| 104         | John     | 30      | 108    | HR_CLERK | 2500   |
| 105         | Diana    | 30      | 108    | IT_ADMIN | 5000   |
| 106         | Smith    | 40      | 110    | AD_ASST  | 3000   |
| 108         | Jennifer | 30      | 110    | HR_DIR   | 6500   |
| 110         | Bob      | 40      | 110    | EX_DIR   | 8000   |
| 120         | Ravi     | 20      | 110    | SA*DIR   | 6500   |

**DEPARTMENTS**

| DEPARTMENT_ID | DEPARTMENT_NAME |
|---------------|-----------------|
| 10            | Admin           |
| 20            | Education       |
| 30            | IT              |
| 40            | Human Resources |

Also examine the SQL statements that create the EMPLOYEES and DEPARTMENTS tables:

```
CREATE TABLE departments
(department_id      NUMBER      PRIMARY KEY,
 department_name    VARCHAR2(30));

CREATE TABLE employees
(EMPLOYEE_ID        NUMBER PRIMARY KEY,
 EMP_NAME           VARCHAR2(20),
 DEPT_ID            NUMBER REFERENCES
 departments(department_id),
 MGR_ID             NUMBER REFERENCES
 employees(employee id),
 MGR_ID             NUMBER REFERENCES
 employees(employee id),
 JOB_ID             VARCHAR2(15).
 SALARY             NUMBER);
ON the EMPLOYEES,
```

On the EMPLOYEES table, EMPLOYEE\_ID is the primary key.  
MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID.  
DEPT\_ID is foreign key to DEPARTMENT\_ID column of the DEPARTMENTS table.  
On the DEPARTMENTS table, DEPARTMENT\_ID is the primary key.  
Examine this DELETE statement:

```
DELETE
FROM departments
WHERE department id = 40;
```

What happens when you execute the DELETE statement?

- A. Only the row with department ID 40 is deleted in the DEPARTMENTS table.
- B. The statement fails because there are child records in the EMPLOYEES table with department ID 40.
- C. The row with department ID 40 is deleted in the DEPARTMENTS table. Also the rows with employee IDs 110 and 106 are deleted from the EMPLOYEES table.
- D. The row with department ID 40 is deleted in the DEPARTMENTS table. Also the rows with employee IDs 106 and 110 and the employees working under employee 110 are deleted from the EMPLOYEES table.
- E. The row with department ID 40 is deleted in the DEPARTMENTS table. Also all the rows in the EMPLOYEES table are deleted.
- F. The statement fails because there are no columns specifies in the DELETE clause of the DELETE statement.

**Question: 162. (H)**

You maintain two tables, CUSTOMER and PROSPECT, that have identical structures but different data. You want to synchronize these two tables by inserting records from the PROSPECT table into the CUSTOMER table, if they do not exist. If the customer already exists in the CUSTOMER table, you want to update customer data.

Which DML statement should you use to perform this task?

- A. INSERT
- B. UPDATE
- C. SYNC
- D. MERGE
- E. You CANNOT perform this task with one DML operation.

**Question: 163. (H)**

Examine the structure of the EMPLOYEES table:

|               |              |                  |
|---------------|--------------|------------------|
| EMPLOYEE_ID   | NUMBER       | NOT NULL         |
| EMP_NAME      | VARCHAR2(30) |                  |
| JOB_ID        | VARCHAR2(20) | DEFAULT 'SA_REP' |
| SAL           | NUMBER       |                  |
| COMM_PCT      | NUMBER       |                  |
| MGR_ID        | NUMBER.      |                  |
| DEPARTMENT_ID | NUMBER       |                  |

You need to update the records of employees 103 and 115. The UPDATE statement you specify should update the rows with the values specified below:

JOB\_ID: Default value specified for this column definition.  
 SAL: Maximum salary earned for the job ID SA\_REP.  
 COMM\_PCT: Default value specified for this commission percentage column, if any.  
 If no default value is specified for the column, the value should be NULL.  
 DEPARTMENT\_ID: Supplied by the user during run time through substitution variable.

Which UPDATE statement meets the requirements?

- A. UPDATE  
employees SET job\_id =  
DEFAULT  
AND Sal = (SELECT MAX(sal)  
FROM employees  
WHERE job\_id = 'SA\_REP' AND  
comm\_pct = DEFAULT AND  
department\_id = &did WHERE  
employee\_id IN (103,115);
- B. UPDATE  
employees SET job\_id =  
DEFAULT AND Sal =  
MAX(sal)

```

AND comm_pct = DEFAULT OR NULL
AND department_id = &did
WHERE employee_id IN (103,115)
AND job_id = 'SA_REP';
C.      UPDATE employees
SET job_id = DEFAULT, Sal
= (SELECT MAX(sal) FROM
employees WHERE job_id =
'SA_REP'), comm_pct =
DEFAULT, department_id =
&did
WHERE employee_id IN (103,115);
D.      UPDATE
employees SET job_id =
DEFAULT, Sal =
MAX(sal), comm_pct =
DEFAULT,
department_id = &did
WHERE employee_id IN (103,115)
AND job_id = 'SA_REP';
E.      UPDATE employees.
SET job_id = DEFAULT, Sal
= (SELECT MAX(sal) FROM
employees WHERE job_id =
'SA_REP')
comm_pct = DEFAULT OR NULL,
department_id = &did
WHERE employee_id IN (103,115);

```

**Question: 164. (H)**

You added a PHONE-NUMBER column of NUMBER data type to an existing EMPLOYEES table. The EMPLOYEES table already contains records of 100 employees. Now, you want to enter the phone numbers of each of the 100 employees into the table. Some of the employees may not have a phone number available.

Which data manipulation operation do you perform?

- a. MERGE
- b. INSERT
- c. UPDATE
- d. ADD
- e. ENTER
- f. You cannot enter the phone numbers for the existing employee records.

**Question: 165. (H)**

Which two statements complete a transaction? (Choose two)

- a. DELETE employees;
- b. DESCRIBE employees;
- c. ROLLBACK TO SAVEPOINT C;
- d. GRANT SELECT ON employees TO SCOTT;
- e. ALTER TABLE  
employees SET UNUSED  
COLUMN sal;
- f. Select MAX(sal)  
FROM employees WHERE  
department\_id = 20;

**Question: 166. (H)**

Examine the data in the EMPLOYEES table.

## EMPLOYEES

| EMPLOYEE_ID | EMP_NAME | DEPT_ID | MGR_ID | JOB_ID   | SALARY |
|-------------|----------|---------|--------|----------|--------|
| 101         | Smith    | 20      | 120    | SA_REP   | 4000   |
| 102         | Martin   | 10      | 105    | CLERK    | 2500   |
| 103         | Chris    | 20      | 120    | IT_ADMIN | 4200   |
| 104         | John     | 30      | 108    | HR_CLERK | 2500   |
| 105         | Diana    | 30      | 108    | IT_ADMIN | 5000   |
| 106         | Smith    | 40      | 110    | AD_ASST  | 3000   |
| 108         | Jennifer | 30      | 110    | HR_DIR   | 6500   |
| 110         | Bob      | 40      |        | EK_DIR   | 8000   |
| 120         | Revi     | 20      | 110    | SA_DIR   | 6500   |

On the EMPLOYEES table, EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID. The JOB\_ID column is a NOT NULL column.

Evaluate this DELETE statement:

```
DELETE employee_id, salary, job_id
FROM employees
WHERE dept_id = 90;
```

Why does the DELETE statement fail when you execute it?

- A. There is no row with dept\_id 90 in the EMPLOYEES table.
- B. You cannot delete the JOB\_ID column because it is a NOT NULL column.
- C. You cannot specify column names in the DELETE clause of the DELETE statement.
- D. You cannot delete the EMPLOYEE\_ID column because it is the primary key of the table.

### Question: 167. (H)

Examine the structure of the EMPLOYEES table:

| EMPLOYEE_ID | NUMBER       | Primary Key |
|-------------|--------------|-------------|
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |

Which three statements insert a row into the table? (Choose three)

- a. INSERT INTO employees  
VALUES ( NULL,  
'John','Smith');
- b. INSERT INTO employees( first\_name,  
last\_name) VALUES('John','Smith');
- c. INSERT INTO  
employees VALUES  
( '1000','John',NULL);
- d. INSERT INTO employees(first\_name,last\_name,  
employee\_id) VALUES ( 1000, 'John','Smith');
- e. INSERT INTO employees  
(employee\_id) VALUES (1000);
- f. INSERT INTO employees (employee\_id, first\_name,  
last\_name) VALUES ( 1000, 'John','');

### Question: 168. (H)

Examine the structure of the EMPLOYEES table:

| EMPLOYEE_ID | NUMBER       | Primary Key |
|-------------|--------------|-------------|
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |

|           |      |  |
|-----------|------|--|
| HIRE_DATE | DATE |  |
|-----------|------|--|

You issue these statements:

```
CREATE table new_emp ( employee_id NUMBER, name VARCHAR2(30));
INSERT INTO new_emp SELECT employee_id , last_name from
employees;
Savepoint s1;
UPDATE new_emp set name = UPPER(name);
Savepoint s2;
Delete from new_emp;
Rollback to s2;
Delete from new_emp where employee_id =180;
UPDATE new_emp set name = 'James';
Rollback to s2;
UPDATE new_emp set name = 'James' WHERE employee_id =180;
Rollback;
```

At the end of this transaction, what is true?

- A. You have no rows in the table.
- B. You have an employee with the name of James.
- C. You cannot roll back to the same savepoint more than once.
- D. Your last update fails to update any rows because employee ID 180 was already deleted.

**Question: 169. (H)**

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          | 3            |               |

**INSTRUCTOR Table**

| INSTRUCTOR_ID | LAST_NAME  | FIRST_NAME |
|---------------|------------|------------|
| 1             | Chao       | Ling       |
| 2             | Vanderbilt | Herbert    |
| 3             | Wigley     | Martha     |
| 4             | Page       | Albert     |

You want to delete the classes that do NOT have an instructor assigned. Which DELETE statement will accomplish the desired result?

- a. DELETE class\_id, class\_name, hours\_credit,  
instructor\_id FROM class  
WHERE instructor\_id IS NULL;
- b. DELETE FROM class  
WHERE instructor\_id NOT IN  
(SELECT instructor\_id
- c. DELETE FROM  
instructor NATURAL JOIN class
- d. DELETE FROM class  
WHERE instructor\_id IS NULL;

**Question: 170. (H)**



Which action will cause an automatic rollback?

- a. GRANT statement
- b. CREATE statement
- c. System crash
- d. exiting from iSQL\*Plus without first committing the changes
- e. subsequent DML statement
- f. SAVEPOINT statement

**Question: 171. (H)**

The PRODUCT table contains these columns:

PRODUCT\_ID NUMBER NOT NULL

PRODUCT\_NAME VARCHAR2(25)

SUPPLIER\_ID NUMBER

LIST\_PRICE NUMBER(7,2)

COST NUMBER(7,2)

You need to increase the list price and cost of all products supplied by Global Imports, Inc. by 5.5 percent.

The SUPPLIER\_ID for Global Imports, Inc. is 105. Which statement should you use?

- a. UPDATE product  
SET list\_price = list\_price \* 1.055  
SET cost = cost \* 1.055 WHERE  
supplier\_id = 105;
- b. UPDATE product  
SET list\_price = list\_price \* .055 AND  
cost = cost \* .055  
WHERE supplier\_id = 105;
- c. UPDATE product  
SET list\_price = list\_price \* 1.055, cost = cost \* 1.055  
WHERE supplier\_id = 105;
- d. UPDATE product  
SET list\_price = list\_price + (list\_price \* .055), cost = cost + (cost \* .055)  
WHERE supplier\_id LIKE 'Global Imports, Inc.'
- e. UPDATE product  
SET list\_price = list\_price + (list\_price \* .055), cost = cost + (cost \* .055)  
WHERE supplier\_id LIKE 'Global Imports, Inc.';

**Question: 172. (H)**

Which two statements would cause an implicit COMMIT to occur? (Choose two.)

- a. GRANT
- b. UPDATE
- c. COMMIT
- d. SELECT
- e. ROLLBACK
- f. RENAME

**Question: 173. (H)**

Examine the structures of the EVENT and NEW\_EVENT tables.

**EVENT Table**

|                   |                |
|-------------------|----------------|
| EVENT_ID          | NUMBER         |
| EVENT_NAME        | VARCHAR2 (30)  |
| EVENT_DESCRIPTION | VARCHAR2 (100) |
| EVENT_TYPE_ID     | NUMBER         |

**NEW\_EVENT Table**

|            |               |
|------------|---------------|
| EVENT_ID   | NUMBER        |
| EVENT_NAME | VARCHAR2 (30) |

|                   |                |
|-------------------|----------------|
| EVENT_DESCRIPTION | VARCHAR2 (100) |
| EVENT_TYPE_ID     | NUMBER         |
| START_DT          | DATE           |

You execute this MERGE statement:

```

MERGE INTO EVENT e
USING (SELECT *
FROM new_event
WHERE event_type_id = 4) n
ON (e.event_id = n.event_id)
WHEN MATCHED THEN
UPDATE SET
e.event_type_id = n.event_type_id,
e.start_dt = n.start_dt
WHEN NOT MATCHED THEN
INSERT (event_id, event_name, event_type_id)
VALUES(n.event_id, n.event_name, n.event_type_id);

```

This MERGE statement generates an error. Which statement describes the cause of the error?

- A subquery CANNOT be used in the USING clause of a MERGE statement.
- Table aliases CANNOT be used in a MERGE statement.
- The ON clause of the statement is invalid.
- The UPDATE portion of the statement is invalid.
- The INSERT portion of the statement is invalid.
- A MERGE statement CANNOT be used with tables that do NOT have an identical structure.

**Question: 174. (H)**

Which statement regarding DML statement functionality is true?

- DELETE can be used to delete rows or columns from a table.
- MERGE will delete rows that do NOT exist in either table.
- UPDATE will add rows to a table if an INTO clause is specified.
- UPDATE can update multiple columns in one table.
- INSERT must contain a VALUES clause.

**Question: 175. (H)**

The STUDENT table contains these columns:

|             |              |          |
|-------------|--------------|----------|
| STU_ID      | NUMBER(9)    | NOT NULL |
| LAST_NAME   | VARCHAR2(30) | NOT NULL |
| FIRST_NAME  | VARCHAR2(25) | NOT NULL |
| DOB         | DATE         |          |
| STU_TYPE_ID | VARCHAR2(1)  | NOT NULL |
| ENROLL_DATE | DATE         |          |

You create another table, named PT\_STUDENT, with an identical structure. You want to insert all part-time students, who have a STU\_TYPE\_ID value of "P", into the new table. You execute this INSERT statement:

```

INSERT INTO pt_student
(SELECT stu_id, last_name, first_name, dob, sysdate
FROM student
WHERE UPPER(stu_type_id) = 'P');

```

What is the result of executing this INSERT statement?

- All part-time students are inserted into the PT\_STUDENT table.
- An error occurs because the PT\_STUDENT table already exists.
- An error occurs because you CANNOT use a subquery in an INSERT statement.

- d. An error occurs because the STU\_TYPE\_ID column is NOT included in the subquery select list.
- e. An error occurs because both the STU\_TYPE\_ID and ENROLL\_DATE columns are NOT included in the subquery select list.
- f. An error occurs because the INSERT statement does NOT contain a VALUES clause.

**Question: 176. (H)**

Examine the structures of the AR\_TRX and AR\_TRX\_HY tables.

**AR\_TRX Table**

|            |              |                       |
|------------|--------------|-----------------------|
| TRX_ID     | NUMBER       | NOT NULL, Primary Key |
| TRX_TYPE   | VARCHAR2 (5) |                       |
| QUANTITY   | NUMBER       |                       |
| UNIT_PRICE | NUMBER (7,2) |                       |
| EXT_AMT    | NUMBER (9,2) |                       |
| TAX_AMT    | NUMBER (7,2) |                       |

**AR\_TRX\_HY Table**

|             |               |                       |
|-------------|---------------|-----------------------|
| TRX_ID      | NUMBER        | NOT NULL, Primary Key |
| TRX_TYPE    | VARCHAR2 (5)  |                       |
| QUANTITY    | NUMBER        |                       |
| UNIT_PRICE  | NUMBER (7,2)  |                       |
| EXT_AMT     | NUMBER (9,2)  |                       |
| TAX_AMT     | NUMBER (7,2)  |                       |
| GRAND_TOTAL | NUMBER (10,2) |                       |
| LOAD_DATE   | DATE          |                       |

You are loading historical accounts receivable data from the AR\_TRX table into the AR\_TRX\_HY table. During the load, you want to transform the data so that the GRAND\_TOTAL column of the AR\_TRX\_HY table is equal to the sum of the EXT\_AMT and TAX\_AMT columns in the AR\_TRX table. You want to set LOAD\_DATE to the current date. If the record already exists in the target table, all values, except TRX\_ID and TRX\_TYPE, should be refreshed with the most recent data.

Which MERGE statement should you execute?

- a. 

```
MERGE INTO
ar_trx_hy h USING ar_trx a
ON (h.trx_id = a.trx_id)
WHEN MATCHED THEN
UPDATE SET a.quantity =
h.quantity, a.unit_price =
h.unit_price, a.ext_amt =
h.ext_amt, a.tax_amt =
h.tax_amt,
a.grand_total = h.ext_amt + h.tax_amt,
a.load_date = sysdate
WHEN NOT MATCHED THEN
INSERT VALUES(a.trx_id, a.trx_type, a.quantity, a.unit_price,
a.ext_amt, a.tax_amt, a.ext_amt + a.tax_amt, sysdate);
```
- b. 

```
MERGE INTO
ar_trx_hy USING ar_trx
USING (trx_id)
WHEN MATCHED THEN
UPDATE SET quantity =
quantity, unit_price =
unit_price, ext_amt =
ext_amt, tax_amt =
tax_amt,
```

```

grand_total = ext_amt + tax_amt,
load_date = sysdate
WHEN NOT MATCHED THEN
INSERT VALUES(trx_id, trx_type, quantity, unit_price,
ext_amt, tax_amt, ext_amt + tax_amt, sysdate);
c.      MERGE INTO
ar_trx_hy h USING ar_trx a
ON (h.trx_id = a.trx_id)
WHEN MATCHED THEN UPDATE
ar_trx_hy SET h.quantity = a.quantity,
h.unit_price = a.unit_price, h.ext_amt =
a.ext_amt, h.tax_amt = a.tax_amt,
h.grand_total = a.ext_amt + a.tax_amt,
h.load_date = sysdate
WHEN NOT MATCHED THEN
INSERT VALUES(a.trx_id, a.trx_type, a.quantity, a.unit_price,
a.ext_amt, a.tax_amt, a.ext_amt + a.tax_amt, sysdate);
d.      MERGE INTO
ar_trx_hy h USING ar_trx a
ON (h.trx_id = a.trx_id)
WHEN MATCHED THEN
UPDATE SET h.quantity =
a.quantity, h.unit_price =
a.unit_price, h.ext_amt =
a.ext_amt, h.tax_amt =
a.tax_amt,
h.grand_total = a.ext_amt + a.tax_amt,
h.load_date = sysdate
WHEN NOT MATCHED THEN
INSERT VALUES(a.trx_id, a.trx_type, a.quantity,
a.unit_price, a.ext_amt, a.tax_amt, a.ext_amt + a.tax_amt,
sysdate);
e.      MERGE
INTO ar_trx_hy h
USING ar_trx a
ON (h.trx_id = a.trx_id)
WHEN MATCHED
UPDATE SET
h.quantity = a.quantity,
h.unit_price =
a.unit_price, h.ext_amt
= a.ext_amt, h.tax_amt
= a.tax_amt,
h.grand_total = a.ext_amt +
a.tax_amt, h.load_date = sysdate
WHEN NOT MATCHED
INSERT VALUES(a.trx_id, a.trx_type, a.quantity,
a.unit_price, a.ext_amt, a.tax_amt, a.ext_amt + a.tax_amt,
sysdate);

```

**Question: 177. (H)**

EMPLOYEES and DEPARTMENTS data:

**EMPLOYEES**

| EMPLOYEE_ID | EMP_NAME | DEPT_ID | MGR_ID | JOB_ID   | SALARY |
|-------------|----------|---------|--------|----------|--------|
| 101         | Smith    | 20      | 120    | SA_REP   | 4000   |
| 102         | Martin   | 10      | 105    | CLERK    | 2500   |
| 103         | Chris    | 20      | 120    | IT_ADMIN | 4200   |
| 104         | John     | 30      | 108    | HR_CLERK | 2500   |

|     |          |    |     |          |      |
|-----|----------|----|-----|----------|------|
| 105 | Diana    | 30 | 108 | IT_ADMIN | 5000 |
| 106 | Smith    | 40 | 110 | AD_ASST  | 3000 |
| 108 | Jennifer | 30 | 110 | HR_DIR   | 6500 |
| 110 | Bob      | 40 |     | EX_DIR   | 8000 |
| 120 | Ravi     | 20 | 110 | SA_DIR   | 6500 |

#### DEPARTMENTS

| DEPARTMENT_ID | DEPARTMENT_NAME |
|---------------|-----------------|
| 10            | Admin           |
| 20            | Education       |
| 30            | IT              |
| 40            | Human Resources |

On the EMPLOYEES table, EMPLOYEE\_ID is the primary key. MGR\_ID is the ID managers and refers to the EMPLOYEE\_ID.

On the DEPARTMENTS table DEPARTMENT\_ID is the primary key.

Evaluate this UPDATE statement.

```
UPDATE employees
SET mgr_id =
    (SELECT mgr_id
     FROM employees
     WHERE dept_id=
        (SELECT department_id
         FROM departments
         WHERE department_name = 'Administration')),
    Salary = (SELECT salary
              FROM employees
              WHERE emp_name = 'Smith')
WHERE job_id = 'IT_ADMIN';
```

What happens when the statement is executed?

- A. The statement executes successfully, leaves the manager ID as the existing value, and changes the salary to 4000 for the employees with ID 103 and 105.
- B. The statement executes successfully, changes the manager ID to NULL, and changes the salary to 4000 for the employees with ID 103 and 105.
- C. The statement executes successfully, changes the manager ID to NULL, and changes the salary to 3000 for the employees with ID 103 and 105.
- D. The statement fails because there is more than one row matching the employee name Smith. E. The statement fails because there is more than one row matching the IT\_ADMIN job ID in the EMPLOYEES table.
- F. The statement fails because there is no 'Administration' department in the DEPARTMENTS table.

#### Question: 178. (H)

The transaction control which prevent more than one user from updating data in a table is which of the following?

- A. Lock.
- B. Commit.
- C. Rollback.
- D. Savepoint.

#### Question: 179.(I)

What is necessary for your query on an existing view to execute successfully?

- A. The underlying tables must have data.
- B. You need SELECT privileges on the view.
- C. The underlying tables must be in the same schema.

D. You need SELECT privileges only on the underlying tables.

**Question: 180. (I)**

Examine the structure of the EMPLOYEES table:

| Column name   | Data Type    | Remarks                                                      |
|---------------|--------------|--------------------------------------------------------------|
| EMPLOYEE_ID   | NUMBER       | NOT NULL, Primary Key                                        |
| EMP_NAME      | VARCHAR2(30) |                                                              |
| JOB_ID        | VARCHAR2(20) | NOT NULL                                                     |
| SAL           | NUMBER       |                                                              |
| MGR_ID        | NUMBER       | References EMPLOYEE_ID column                                |
| DEPARTMENT_ID | NUMBER       | Foreign key to DEPARTMENT_ID column of the DEPARTMENTS table |

You need to create a view called EMP\_VU that allows the user to insert rows through the view. Which SQL statement, when used to create the EMP\_VU view, allows the user to insert rows?

- A. CREATE VIEW emp\_Vu  
AS SELECT employee\_id,  
emp\_name, department\_id  
FROM employees  
WHERE mgr\_id IN (102, 120);
- B. CREATE VIEW emp\_Vu AS  
SELECT employee\_id, emp\_name, job\_id  
department\_id  
FROM employees  
WHERE mgr\_id IN (102, 120);
- C. CREATE VIEW emp\_Vu AS  
SELECT department\_id, SUM(sal) TOTALSAL  
FROM employees  
WHERE mgr\_id IN (102, 120)  
GROUP BY department\_id;
- D. CREATE VIEW emp\_Vu AS  
SELECT employee\_id, emp\_name, job\_id,  
DISTINCT department\_id  
FROM employees;

**Question: 181. (I)**

You created a view called EMP\_DEPT\_VU that contains three columns from the EMPLOYEES and DEPARTMENTS tables:

EMPLOYEE\_ID, EMPLOYEE\_NAME AND DEPARTMENT\_NAME.

The DEPARTMENT\_ID column of the EMPLOYEES table is the foreign key to the primary key DEPARTMENT\_ID column of the DEPARTMENTS table.

You want to modify the view by adding a fourth column, MANAGER\_ID of NUMBER data type from the EMPLOYEES tables.

How can you accomplish this task?

- A. ALTER VIEW emp\_dept\_vu (ADD manager\_id NUMBER);
- B. MODIFY VIEW emp\_dept\_vu (ADD manager\_id NUMBER);
- C. ALTER VIEW emp\_dept\_vu AS  
SELECT employee\_id, employee\_name,  
department\_name, manager\_id FROM employee  
e, departments d WHERE e.department\_id =  
d.department\_id;
- D. MODIFY VIEW emp\_dept\_vu AS  
SELECT employee\_id, employee\_name,  
department\_name, manager\_id

```

FROM employees e, departments d
WHERE e.department_id = d.department_id;
E.      CREATE OR REPLACE VIEW
emp_dept_vu AS SELECT employee_id,
employee_name, department_name, manager_id
FROM employees e, departments d
WHERE e.department_id = d.department_id;
F.      You must remove the existing view first, and then run the CREATE VIEW command
with a new column list to modify a view.

```

**Question: 182. (I)**

Which statements concerning the creation of a view are true? (Choose all that apply.)

- A. A constraint name must be provided when using the WITH CHECK OPTION clause or the statement will fail.
- B. View columns that are the result of derived values must be given a column alias.
- C. When the view already exists, using the OR REPLACE option requires the re-granting of the object privileges previously granted on the view.
- D. A view may have column names that are different than the actual base table(s) column names by using column aliases.

**Question: 183. (I)**

You are granted the CREATE VIEW privilege. What does this allow you to do?

- A. Create a table view.
- B. Create a view in any schema.
- C. Create a view in your schema.
- D. Create a sequence view in any schema.
- E. Create a view that is accessible by everyone.
- F. Create a view only of it is based on tables that you created.

**Question: 184. (I)**

Examine the structure of the CURR\_ORDER table as shown below:

**CURR\_ORDER Table**

|             |              |                                                                    |
|-------------|--------------|--------------------------------------------------------------------|
| ORDER_ID    | NUMBER       | NOT NULL, Primary Key                                              |
| CUSTOMER_ID | NUMBER       | NOT NULL, Foreign Key to CUSTOMER_ID column of the CUSTOMERS table |
| EMP_ID      | NUMBER       | NOT NULL, Foreign Key to EMP_ID column of the EMPLOYEES table      |
| ORDER_DT    | DATE         | NOT NULL                                                           |
| ORDER_AMT   | NUMBER (7,2) |                                                                    |
| SHIP_METHOD | VARCHAR2 (5) |                                                                    |

You created the ORDER\_V view selecting all rows and columns from the ORDER table where the amount of the order was over \$250 and the date of the order was after January 1, 2000. The CREATE VIEW statement included the WITH CHECK OPTION clause.

Which statement will execute successfully?

- A. INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_dt, order\_amt) VALUES (840, 292, 104, '10-OCT-2001', 318);
- B. INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_amt, ship\_method) VALUES (936, 292, 104, 256.3, 'UPXS');
- C. INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_dt, order\_amt, ship\_method) VALUES (164, 292, 104, '10-MAY-2001', 3.56, 'UPXS');
- D. INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_dt, order\_amt, ship\_method) VALUES (203, 292, 104, '10-OCT-1999', 298.4, 'UPXS');

**Question: 185. (I)**

An inline view is a SELECT statement that is given an alias and is embedded in the \_\_\_\_\_ clause of another SELECT statement.

- A. FROM
- B. WHERE
- C. SELECT
- D. CASE

**Question: 186. (I)**

You need to create a view EMP\_VU. The view should allow the users to manipulate the records of only the employees that are working for departments 10 or 20.

Which SQL statement would you use to create the view EMP\_VU?

- A. CREATE VIEW  
emp\_vu AS SELECT \*  
FROM employees  
WHERE department\_id IN (10,20);
- B. CREATE VIEW  
emp\_vu AS SELECT \*  
FROM employees  
WHERE department\_id IN (10,20)  
WITH READ ONLY;
- C. CREATE VIEW  
emp\_vu AS SELECT \*  
FROM employees  
WHERE department\_id IN (10,20)  
WITH CHECK OPTION;
- D. CREATE FORCE VIEW  
emp\_vu AS SELECT \*  
FROM employees  
WHERE department\_id IN (10,20);
- E. CREATE FORCE VIEW  
emp\_vu AS SELECT \*  
FROM employees  
WHERE department\_id IN (10,20)  
NO UPDATE;

**Question: 187. (I)**

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

- A. A view can be created as read only.
- B. A view can be created as a join on two or more tables.
- C. A view cannot have an ORDER BY clause in the SELECT statement.
- D. A view cannot be created with a GROUP BY clause in the SELECT statement.
- E. A view must have aliases defined for the column names in the SELECT statement.

**Question: 188.(J)**

The database administrator of your company created a public synonym called HR for the HUMAN\_RESOURCES table of the GENERAL schema, because many users frequently use this table.

As a user of the database, you created a table called HR in your schema. What happens when you execute this query?

```
SELECT *  
FROM HR;
```

- A. You obtain the results retrieved from the public synonym HR created by the database administrator.
- B. You obtain the results retrieved from the HR table that belongs to your schema.
- C. You get an error message because you cannot retrieve from a table that has the same name as a public synonym.



- D. You obtain the results retrieved from both the public synonym HR and the HR table that belongs to your schema, as a Cartesian product.
- E. You obtain the results retrieved from both the public synonym HR and the HR table that belongs to your schema, as a FULL JOIN.

**Question: 189. (J)**

Examine the structure of the EMPLOYEES table:

| Column name   | Data type    | Remarks                       |
|---------------|--------------|-------------------------------|
| EMPLOYEE_ID   | NUMBER       | NOT NULL, Primary Key         |
| LAST_NAME     | VARCNAR2(30) |                               |
| FIRST_NAME    | VARCNAR2(30) |                               |
| JOB_ID        | NUMBER       |                               |
| SAL           | NUMBER       |                               |
| MGR_ID        | NUMBER       | References EMPLOYEE_ID column |
| DEPARTMENT_ID | NUMBER       |                               |

You need to create an index called NAME\_IDX on the first name and last name fields of the EMPLOYEES table. Which SQL statement would you use to perform this task?

- A. CREATE INDEX NAME\_IDX (first\_name, last\_name);
- B. CREATE INDEX NAME\_IDX (first\_name AND last\_name);
- C. CREATE INDEX  
NAME\_IDX ON (first\_name,  
last\_name);
- D. CREATE INDEX NAME\_IDX  
ON employees (first\_name AND last\_name);
- E. CREATE INDEX NAME\_IDX  
ON employees(first\_name, last\_name);
- F. CREATE INDEX NAME\_IDX  
FOR employees(first\_name, last\_name);

**Question: 190. (J)**

In which scenario would index be most useful?

- A. The indexed column is declared as NOT NULL.
- B. The indexed columns are used in the FROM clause.
- C. The indexed columns are part of an expression.
- D. The indexed column contains a wide range of values.

**Question: 191. (J)**

Mary has a view called EMP\_DEPT\_LOC\_VU that was created based on the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables. She granted SELECT privilege to Scott on this view. Which option enables Scott to eliminate the need to qualify the view with the name MARY.EMP\_DEPT\_LOC\_VU each time the view is referenced?

- A. Scott can create a synonym for the EMP\_DEPT\_LOC\_VU bus using the command:  
CREATE PRIVATE SYNONYM EDL\_VU  
FOR mary.EMP DEPT\_LOC\_VU;  
then he can prefix the columns with this synonym.
- B. Scott can create a synonym for the EMP\_DEPT\_LOC\_VU by using the command:  
CREATE SYNONYM EDL\_VU  
FOR mary.EMP\_DEPT\_LOC\_VU;  
then he can prefix the columns with this synonym.
- C. Scott can create a synonym for the EMP\_DEPT\_LOC\_VU by using the command:  
CREATE LOCAL SYNONYM EDL\_VU  
FOR mary.EMP DEPT\_LOC\_VU;  
then he can prefix the columns with this synonym.
- D. Scott can create a synonym for the EMP\_DEPT\_LOC\_VU by using the command:

CREATE SYNONYM EDL\_VU.

ON mary(EMP\_DEPT\_LOC\_VU);

then he can prefix the columns with this synonym.

E. Scott cannot create a synonym because synonyms can be created only for tables.

F. Scott cannot create any synonym for Mary's view. Mary should create a private synonym for the view and grant SELECT privilege on that synonym to Scott.

**Question: 192. (J)**

Which two statements about sequences are true? (Choose two)

- A. You use a NEXTVAL pseudo column to look at the next possible value that would be generated from a sequence, without actually retrieving the value.
- B. You use a CURRVAL pseudo column to look at the current value just generated from a sequence, without affecting the further values to be generated from the sequence.
- C. You use a NEXTVAL pseudo column to obtain the next possible value from a sequence by actually retrieving the value from the sequence.
- D. You use a CURRVAL pseudo column to generate a value from a sequence that would be used for a specified database column.
- E. If a sequence starting from a value 100 and incremented by 1 is used by more than one application, then all of these applications could have a value of 105 assigned to their column whose value is being generated by the sequence.
- F. You use REUSE clause when creating a sequence to restart the sequence once it generates the maximum value defined for the sequence.

**Question: 193. (J)**

Which SELECT statement will display the next value of the PARTS\_ID\_SEQ sequence by actually retrieving the value from the sequence?

- A. SELECT NEXTVAL(parts\_id\_seq)  
FROM SYS.DUAL;
- B. SELECT  
parts\_id\_seq.NEXTVAL FROM  
inventory;
- C. SELECT parts\_id\_seq.NEXTVAL  
FROM SYS.DUAL;
- D. SELECT  
NEXTVAL(parts\_id\_seq) FROM  
inventory;
- E. SELECT parts\_id\_seq  
NEXTVAL FROM inventory;

**Question: 194. (J)**

You issue this statement:

```
CREATE PUBLIC SYNONYM part  
FOR linda.product;
```

Which task was accomplished by this statement?

- A. A new segment object was created.
- B. A new object privilege was assigned.
- C. A new system privilege was assigned.
- D. The need to qualify an object name with its schema was eliminated.

**Question: 195. (J)**

User Mark wants to eliminate the need to type the full table name when querying the TRANSACTION\_HISTORY table existing in her schema. All other database users should use the schema and full table name when referencing this table.

Which statement should user Marilyn execute?

- A. CREATE PUBLIC SYNONYM  
trans\_hist FOR mark;
- B. CREATE SYNONYM  
trans\_hist FOR  
transaction\_history;
- C. CREATE PRIVATE SYNONYM  
trans\_hist FOR mark.transaction\_history;
- D. CREATE PUBLIC trans\_hist  
SYNONYM FOR mark.transaction\_history;

**Question: 196. (J)**

The TEACHER table in your schema contains these columns:

|            |              |                       |
|------------|--------------|-----------------------|
| ID         | NUMBER(9)    | NOT NULL, Primary Key |
| LAST_NAME  | VARCHAR2(25) |                       |
| FIRST_NAME | VARCHAR2(25) |                       |
| SUBJECT_ID | NUMBER(9)    |                       |

You execute this statement:

```
CREATE INDEX teacher_name_idx
ON teacher(first_name, last_name);
```

Which statement is true?

- A. The statement creates a composite non-unique index.
- B. The statement creates a composite unique index.
- C. You must have the CREATE ANY INDEX privilege for the statement to succeed.
- D. The statement will fail because it contains a syntax error.

**Question: 197. (J)**

The LINE\_ITEM table contains these columns:

|             |             |
|-------------|-------------|
| LINE_ITEMID | NUMBER(9)   |
| ORDER_ID    | NUMBER(9)   |
| PRODUCT_ID  | VARCHAR2(9) |
| QUANTITY    | NUMBER(5)   |

You created a sequence called LINE\_ITEMID\_SEQ to generate sequential values for the LINE\_ITEMID column.

Evaluate this SELECT statement:

```
SELECT line_itemid_seq.CURRVAL
FROM dual;
```

Which task will this statement accomplish?

- A. displays the next value of the LINE\_ITEMID\_SEQ sequence
- B. displays the current value of the LINE\_ITEMID\_SEQ sequence
- C. populates the LINE\_ITEMID\_SEQ sequence with the next value
- D. increments the LINE\_ITEMID column

**Question: 198. (J)**

Evaluate this statement:

```
CREATE SEQUENCE line_item_id_seq
START WITH 10001
MAXVALUE 999999999
NOCYCLE;
```

Which statement about this CREATE SEQUENCE statement is true?

- A. The sequence will reuse numbers and will start with 10001.
- B. The sequence will never reuse any numbers and will increment by 1.
- C. The sequence will continue to generate values after it reaches its maximum value.
- D. The CREATE SEQUENCE statement will cause a syntax error because an INCREMENT BY value is not included.

**Question: 199. (J)**

Which statement should you use to eliminate the need for all users to qualify Marilyn's INVENTORY table with her schema when querying the table?

- A. CREATE SYNONYM  
inventory FOR inventory;
- B. CREATE PUBLIC SYNONYM  
inventory FOR marilyn;
- C. CREATE PUBLIC SYNONYM  
inventory FOR marilyn.inventory;
- D. CREATE PUBLIC inventory  
SYNONYM FOR marilyn.inventory;

**Question: 200. (J)**

Which statement will user Barbara use to create a private synonym when referencing the EMPLOYEE table existing in user Chan's schema?

- A. CREATE  
SYNONYM emp FOR  
chan.employee;
- B. CREATE PUBLIC  
SYNONYM emp FOR chan.barbara;
- C. CREATE PRIVATE SYNONYM emp  
FOR chan.employee;
- D. CREATE PUBLIC emp  
SYNONYM FOR chan.employee;

**Question: 201. (J)**

Why would you NOT create an index on a column in the CLASS\_SCHEDULE table?

- A. to reduce disk I/O
- B. to speed up row retrieval
- C. to speed up queries if the table has less than 50 rows
- D. to speed up queries that return less than 3 percent of the rows
- E. to speed up queries that include a foreign key reference to the STUDENT table

**Question: 202.(K)**

Which data dictionary table should you query to view the object privileges granted to the user on specific columns?

- A. USER\_TAB\_PRIVS\_MADE
- B. USER\_TAB\_PRIVS
- C. USER\_COL\_PRIVS\_MADE
- D. USER\_COL\_PRIVS

**Question: 203. (K)**

Which two statements accurately describe a role? (Choose two.)

- A. A role can be given to a maximum of 1000 users.
- B. A user can have access to a maximum of 10 roles.
- C. A role can have a maximum of 100 privileges contained in it.
- D. Privileges are given to a role by using the CREATE ROLE statement.

- E. A role is a named group of related privileges that can be granted to the user.
- F. A user can have access to several roles, and several users can be assigned the same role.

**Question: 204. (K)**

Scott issues the SQL statements:

```
CREATE TABLE dept
(deptno NUMBER(2),
dname VARCHAR2(14),
loc VARCHAR2(13));
GRANT SELECT
ON DEPT
TO SUE;
```

If Sue needs to select from Scott's DEPT table, which command should she use?

A.

```
SELECT *
```

```
FROM DEPT;
```

B. SELECT \*

```
FROM SCOTT.DEPT;
```

C. SELECT \*

```
FROM DBA.SCOTT.DEPT;
```

D. SELECT \*

```
FROM ALL_USERS
```

```
WHERE USER_NAME = 'SCOTT'
```

```
AND TABLE NAME = 'DEPT';
```

**Question: 205. (K)**

Which statement creates a new user?

A. CREATE USER susan;

B. CREATE OR REPLACE USER susan;

C. CREATE NEW USER susan  
DEFAULT;

D. CREATE

```
USER susan
```

```
IDENTIFIED BY blue;
```

E. CREATE NEW

```
USER susan IDENTIFIED by  
blue;
```

F. CREATE OR REPLACE USER  
susan IDENTIFIED BY blue;

**Question: 206. (K)**

What is true about the WITH GRANT OPTION clause?

- A. It allows a grantee DBA privileges.
- B. It is required syntax for object privileges.
- C. It allows privileges on specified columns of tables.
- D. It is used to grant an object privilege on a foreign key column.
- E. It allows the grantee to grant object privileges to other users and roles.

**Question: 207. (K)**

You need to give the MANAGER role the ability to select from, insert into, and modify existing rows in the STUDENT\_GRADES table. Anyone given this MANAGER role should be able to pass those privileges on to others.

Which statement accomplishes this?

- A. GRANT select, insert,  
update ON student\_grades  
TO manager;
- B. GRANT select, insert,  
update ON student\_grades  
TO ROLE manager;
- C. GRANT select, insert,  
modify ON student\_grades  
TO manager  
WITH GRANT OPTION;
- D. GRANT select, insert,  
update ON student\_grades.  
TO manager  
WITH GRANT OPTION;
- E. GRANT select, insert,  
update ON student\_grades  
TO ROLE manager  
WITH GRANT OPTION;
- F. GRANT select, insert,  
modify ON student\_grades  
TO ROLE manager  
WITH GRANT OPTION;

**Question: 208. (K)**

When should you create a role? (Choose two)

- A. To simplify the process of creating new users using the CREATE USER xxx IDENTIFIED by yyy statement.
- B. To grant a group of relate privileges to a user.
- C. When the number of people using the database is very high.
- D. To simplify the process of granting and revoking privileges.
- E. To simplify profile maintenance for a user who is constantly traveling.

**Question: 209. (K)**

The DBA issues this SQL command:

```
CREATE USER scott
IDENTIFIES by tiger;
```

What privileges does the user Scott have at this point?

- A. No privileges.
- B. Only the SELECT privilege.
- C. Only the CONNECT privilege.
- D. All the privileges of a default user.

**Question: 210. (K)**

You granted user Joe the INDEX and REFERENCES privileges on the INVENTORY table. Which statement did you use?

- A.  
  
GRANT ALL  
ON inventory  
TO joe;
- B. GRANT ANY  
PRIVILEGE ON inventory  
TO joe;
- C. GRANT INDEX AND  
REFERENCES ON inventory  
TO joe;

D. GRANT ALL WITH GRANT  
OPTION ON inventory  
TO joe;

**Question: 211. (K)**

Which condition would allow a user to grant SELECT privileges on the CUSTOMER table to everyone using the PUBLIC keyword?

- A. The user has SELECT privileges on the CUSTOMER table.
- B. The user owns the CUSTOMER table.
- C. The user has been granted the PUBLIC privilege.
- D. The user has been granted the SELECT privilege with the PUBLIC OPTION.

**Question: 212. (K)**

You issued this statement:

```
REVOKE REFERENCES  
ON inventory  
FROM joe  
CASCADE CONSTRAINTS;
```

Which two tasks were accomplished by executing this statement? (Choose two.)

- A. All the PRIMARY KEY constraints created by user Joe were removed.
- B. All the FOREIGN KEY constraints on the INVENTORY table created by user Joe were removed.
- C. The ability to create a PRIMARY KEY constraint was revoked from user Joe.
- D. The ability to create a FOREIGN KEY constraint on the INVENTORY table was revoked from user Joe.
- E. All the constraints created by user Joe were removed.
- F. The ability to create any constraints was revoked from user Joe.

**Question: 213. (K)**

Which privilege is an object privilege?

- A. INDEX
- B. DROP USER
- C. CREATE SESSION
- D. BACKUP ANY TABLE

**Question: 214. (K)**

You issued this statement:

```
GRANT UPDATE  
ON inventory  
TO joe  
WITH GRANT OPTION;
```

Which task was accomplished?

- A. Only a system privilege was granted to user Joe.
- B. Only an object privilege was granted to user Joe.
- C. User Joe was granted all privileges on the INVENTORY object.
- D. Both an object privilege and a system privilege were granted to user Joe.

**Question: 215. (K)**

Which statement is true concerning a new user that has only been granted the CREATE SESSION privilege?

- A. The user can create a table.
- B. The user can alter their password.

- C. The user can select from tables they have created.
- D. The user can grant privileges on objects that they own.

**Question: 216. (K)**

Evaluate this SQL script:

```
CREATE USER hr IDENTIFIED BY hr01;
CREATE ROLE hr_manager;
CREATE ROLE hr_clerk;
GRANT hr_clerk TO hr_manager;
CREATE ROLE hr_director;
GRANT hr_manager TO hr_director;
GRANT hr_director TO hr
/
```

How many roles will the HR user have access to?

- A. 0
- B. 1
- C. 2
- D. 3

**Question: 217 (A)**

Examine the structure of the EMP\_DEPT\_VU view:

| Column Name   | Type         | Remarks                    |
|---------------|--------------|----------------------------|
| EMPLOYEE_ID   | NUMBER       | From the EMPLOYEES table   |
| EMP_NAME      | VARCHAR2(30) | From the EMPLOYEES table   |
| JOB_ID        | VARCHAR2(20) | From the EMPLOYEES table   |
| SALARY        | NUMBER       | From the EMPLOYEES table   |
| DEPARTMENT_ID | NUMBER       | From the DEPARTMENTS table |
| DEPT_NAME     | VARCHAR2(30) | From the DEPARTMENTS table |

Which SQL statement produces an error?

- A. SELECT \* FROM emp\_dept\_vu;
- B. SELECT department\_id, SUM(salary) FROM emp\_dept\_vu GROUP BY department\_id;
- C. SELECT department\_id, job\_id, AVG(salary) FROM  
emp\_dept\_vu GROUP BY department\_id, job\_id;
- D. SELECT job\_id, SUM(salary) FROM emp\_dept\_vu WHERE  
department\_id IN (10,20) GROUP BY job\_id HAVING SUM(salary) >  
20000;
- E. None of the statements produce an error; all are valid.

**Question: 218 (H)**

You own a table called EMPLOYEES with this table structure:

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| HIRE_DATE   | DATE         |             |

What happens when you execute this DELETE statement?

DELETE employees;

- A. You get an error because of a primary key violation.
- B. The data and structure of the EMPLOYEES table are deleted.
- C. The data in the EMPLOYEES table is deleted but not the structure.
- D. You get an error because the statement is not syntactically correct.

**Question: 219 (H)**



Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

| EMPLOYEES   |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| HIRE_DATE   | DATE         |             |

| NEW_EMPLOYEES |              |             |
|---------------|--------------|-------------|
| EMPLOYEE_ID   | NUMBER       | Primary Key |
| NAME          | VARCHAR2(60) |             |

Which MERGE statement is valid?

- A. 

```
MERGE INTO new_employees c USING employees e ON
(c.employee_id = e.employee_id) WHEN MATCHED THEN UPDATE
SET c.name = e.first_name ||','|| e.last_name WHEN NOT MATCHED
THEN INSERT VALUES (e.employee_id, e.first_name ||','||
e.last_name);
```
- B. 

```
MERGE new_employees c USING employees e ON (c.employee_id =
e.employee_id) WHEN EXISTS THEN UPDATE SET c.name = e.first_name ||','||
e.last_name WHEN NOT MATCHED THEN INSERT VALUES (e.employee_id,
e.first_name ||','|| e.last_name);
```
- C. 

```
MERGE INTO new_employees c USING employees e ON
(c.employee_id = e.employee_id) WHEN EXISTS THEN UPDATE SET
c.name = e.first_name ||','|| e.last_name WHEN NOT MATCHED THEN
INSERT VALUES(e.employee_id, e.first_name ||','||
e.last_name);
```
- D. 

```
MERGE new_employees c FROM employees e ON (c.employee_id
= e.employee_id) WHEN MATCHED THEN UPDATE SET c.name = e.first_name ||','||
e.last_name WHEN NOT MATCHED THEN INSERT INTO new_employees VALUES
(e.employee_id, e.first_name ||','|| e.last_name);
```

**Question: 220 (C)**

The EMPLOYEES table contains these columns:

|             |               |
|-------------|---------------|
| EMPLOYEE_ID | NUMBER(4)     |
| ENAME       | VARCHAR2 (25) |
| JOB_ID      | VARCHAR2(10)  |

Which SQL statement will return the ENAME, length of the ENAME, and the numeric position of the letter "a" in the ENAME column, for those employees whose ENAME ends with a the letter "n"?

- A. 

```
SELECT ENAME, LENGTH(ENAME), INSTR(ENAME, 'a') FROM EMPLOYEES
WHERE SUBSTR(ENAME, -1, 1) = 'n';
```
- B. 

```
SELECT ENAME, LENGTH(ENAME), INSTR(ENAME, , -1, 1) FROM EMPLOYEES
WHERE SUBSTR(ENAME, -1, 1) = 'n';
```
- C. 

```
SELECT ENAME, LENGTH(ENAME), SUBSTR(ENAME, -1, 1) FROM EMPLOYEES
WHERE INSTR(ENAME, 1, 1) = 'n';
```
- D. 

```
SELECT ENAME, LENGTH(ENAME), SUBSTR(ENAME, -1, 1) FROM EMPLOYEES
WHERE INSTR(ENAME, -1, 1) = 'n';
```

**Question: 221 (G)**

Which four are valid Oracle constraint types? (Choose four.)

- A. CASCADE
- B. UNIQUE
- C. NONUNIQUE
- D. CHECK
- E. PRIMARY KEY
- F. CONSTANT
- G. NOT NULL

**Question: 222 (I)**

Which SQL statement would you use to remove a view called EMP\_DEPT\_VU from your schema?

- A. DROP emp\_dept\_vu;
- B. DELETE emp\_dept\_vu;
- C. REMOVE emp\_dept\_vu;
- D. DROP VIEW emp\_dept\_vu;
- E. DELETE VIEW emp\_dept\_vu;
- F. REMOVE VIEW emp\_dept\_vu;

**Question: 223 (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)

Which statement produces the number of different departments that have employees with last name Smith?

- A. SELECT COUNT(\*) FROM employees WHERE last\_name='Smith';
- B. SELECT COUNT (dept\_id) FROM employees WHERE last\_name='Smith';
- C. SELECT DISTINCT(COUNT(dept\_id)) FROM employees WHERE last\_name='Smith';
- D. SELECT COUNT(DISTINCT dept\_id) FROM employees WHERE last\_name='Smith';
- E. SELECT UNIQUE(dept\_id) FROM employees WHERE last\_name='Smith';

**Question: 224 (I)**

Top N analysis requires \_\_\_\_ and \_\_\_\_\_. (Choose two.)

- A. the use of rowed
- B. a GROUP BY clause
- C. an ORDER BY clause
- D. only an inline view
- E. an inline view and an outer query

**Question: 225 (G)**

Which statement adds a constraint that ensures the CUSTOMER\_NAME column of the CUSTOMERS table holds a value?

- A. ALTER TABLE customers ADD CONSTRAINT cust\_name\_nn CHECK customer\_name IS NOT NULL;
- B. ALTER TABLE customers MODIFY CONSTRAINT cust\_name\_nn CHECK customer\_name IS NOT NULL;
- C. ALTER TABLE customers MODIFY customer\_name CONSTRAINT cust\_name\_nn NOT NULL;
- D. ALTER TABLE customers MODIFY customer\_name CONSTRAINT cust\_name\_nn IS NOT NULL;
- E. ALTER TABLE customers MODIFY name CONSTRAINT cust\_name\_nn NOT NULL;
- F. ALTER TABLE customers ADD CONSTRAINT cust\_name\_nn CHECK customer\_name NOT NULL;

**Question: 226 (A)**

Evaluate this SQL statement:

```
SELECT ename, sal, 12*sal+100 FROM emp;
```

The SAL column stores the monthly salary of the employee. Which change must be made to the above syntax to calculate the annual compensation as "monthly salary plus a monthly bonus of \$100, multiplied by 12"?

- A. No change is required to achieve the desired results.
- B. SELECT ename, sal, 12\*(sal+100) FROM emp;

- C. SELECT ename, sal, (12\*sal)+100 FROM emp;
- D. SELECT ename, sal+100,\*12 FROM emp;

**Question: 227 (K)**

You are the DBA for an academic database. You need to create a role that allows a group of users to modify existing rows in the STUDENT\_GRADES table.

Which set of statements accomplishes this?

- A. CREATE ROLE registrar; GRANT MODIFY ON student\_grades TO registrar; GRANT registrar to user1, user2, user3
- B. CREATE NEW ROLE registrar; GRANT ALL ON student\_grades TO registrar; GRANT registrar to user1, user2, user3
- C. CREATE ROLE registrar; GRANT UPDATE ON student\_grades TO registrar; GRANT ROLE registrar to user1, user2, user3
- D. CREATE ROLE registrar; GRANT UPDATE ON student\_grades TO registrar; GRANT registrar to user1, user2, user3;
- E. CREATE registrar; GRANT CHANGE ON student\_grades TO registrar; GRANT registrar;

**Question: 228 (G)**

You need to modify the STUDENTS table to add a primary key on the STUDENT\_ID column. The table is currently empty.

Which statement accomplishes this task?

- A. ALTER TABLE students ADD PRIMARY KEY student\_id;
- B. ALTER TABLE students ADD CONSTRAINT PRIMARY KEY (student\_id);
- C. ALTER TABLE students ADD CONSTRAINT stud\_id\_pk PRIMARY KEY student\_id;
- D. ALTER TABLE students ADD CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);
- E. ALTER TABLE students MODIFY CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

**Question: 229 (B)**

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)  
SEMESTER\_END DATE  
GPA NUMBER(4,3)

The registrar requested a report listing the students' grade point averages (GPA) sorted from highest grade point average to lowest.

Which statement produces a report that displays the student ID and GPA in the sorted order requested by the registrar?

- A. SELECT student\_id, gpa FROM student\_grades ORDER BY gpa ASC;
- B. SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa ASC;
- C. SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa;
- D. SELECT student\_id, gpa FROM student\_grades ORDER BY gpa;
- E. SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa DESC;
- F. SELECT student\_id, gpa FROM student\_grades ORDER BY gpa DESC;

**Question: 230 (F)**

Which describes the default behavior when you create a table?

- A. The table is accessible to all users.
- B. Tables are created in the public schema.
- C. Tables are created in your schema.
- D. Tables are created in the DBA schema.
- E. You must specify the schema when the table is created.

**Question: 231 (C)**

Which four are attributes of single row functions? (Choose four.)

- A. cannot be nested
- B. manipulate data items
- C. act on each row returned
- D. return one result per row
- E. accept only one argument and return only one value
- F. accept arguments which can be a column or an expression

**Question: 232 (F)**

You need to create a table named ORDERS that contain four columns:

- an ORDER\_ID column of number data type
- a CUSTOMER\_ID column of number data type
- an ORDER\_STATUS column that contains a character data type
- a DATE\_ORDERED column to contain the date the order was placed.

When a row is inserted into the table, if no value is provided when the order was placed, today's date should be used instead.

Which statement accomplishes this?

- A. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE = SYSDATE);
- B. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE DEFAULT SYSDATE);
- C. CREATE OR REPLACE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE DEFAULT SYSDATE);
- D. CREATE OR REPLACE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE = SYSDATE);
- E. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status NUMBER (10),  
date\_ordered DATE = SYSDATE);
- F. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status NUMBER (10),  
date\_ordered DATE DEFAULT SYSDATE);

**Question: 233 (H)**

Examine the structure of the EMPLOYEES table:

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |

Which three statements insert a row into the table? (Choose three.)

- A. INSERT INTO employees VALUES ( NULL, 'John', 'Smith');
- B. INSERT INTO employees( first\_name, last\_name) VALUES( 'John', 'Smith');
- C. INSERT INTO employees VALUES ( '1000', 'John', NULL);
- D. INSERT INTO employees (first\_name, last\_name, employee\_id)  
VALUES ( 1000, 'John', 'Smith');
- E. INSERT INTO employees (employee\_id) VALUES (1000);
- F. INSERT INTO employees (employee\_id, first\_name, last\_name) VALUES ( 1000,  
'John', '');

**Question: 234 (K)**

The user Sue issues this SQL statement:

GRANT SELECT ON sue.EMP TO alicE WITH GRANT OPTION;

The user Alice issues this SQL statement:

GRANT SELECT ON sue.EMP TO reena WITH GRANT OPTION;

The user Reena issues this SQL statement:

GRANT SELECT ON sue.EMP TO timber;

The user Sue issues this SQL statement:

REVOKE select on sue.EMP FROM alicE;

For which users does the revoke command revoke SELECT privileges on the SUE.EMP table?

- A. Alice only
- B. Alice and Reena
- C. Alice, Reena, and Timber
- D. Sue, Alice, Reena, and Timber

**Question: 235 (B)**

The EMPLOYEES table contains these columns:

|             |               |
|-------------|---------------|
| EMPLOYEE_ID | NUMBER(4)     |
| LAST_NAME   | VARCHAR2 (25) |
| JOB_ID      | VARCHAR2(10)  |

You want to search for strings that contain 'SA\_' in the JOB\_ID column. Which SQL statement do you use?

- A. SELECT employee\_id, last\_name, job\_id FROM employees WHERE  
job\_id LIKE '%SA\\_%' ESCAPE '\';
- B. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE  
'%SA\_';
- C. SELECT employee\_id, last\_name, job\_id FROM employees  
WHERE job\_id LIKE '%SA\_' ESCAPE '"';
- D. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id = '%SA\_';

**Question: 236 (A)**

The CUSTOMERS table has these columns:

|                  |               |          |
|------------------|---------------|----------|
| CUSTOMER_ID      | NUMBER(4)     | NOT NULL |
| CUSTOMER_NAME    | VARCHAR2(100) | NOT NULL |
| CUSTOMER_ADDRESS | VARCHAR2(150) |          |
| CUSTOMER_PHONE   | VARCHAR2(20)  |          |

You need to produce output that states "Dear Customer customer\_name, ". The customer\_name data values come from the CUSTOMER\_NAME column in the CUSTOMERS table.

Which statement produces this output?

- A. SELECT dear customer, customer\_name, FROM customers;
- B. SELECT "Dear Customer", customer\_name || ',' FROM customers;
- C. SELECT 'Dear Customer ' || customer\_name ',' FROM customers;
- D. SELECT 'Dear Customer ' || customer\_name || ',' FROM customers;
- E. SELECT "Dear Customer " || customer\_name || "," FROM customers;

F. `SELECT 'Dear Customer ' || customer_name || ', ' || FROM customers;`

**Question: 237 (J)**

What is true about sequences?

- A. Once created, a sequence belongs to a specific schema.
- B. Once created, a sequence is linked to a specific table.
- C. Once created, a sequence is automatically available to all users.
- D. Only the DBA can control which sequence is used by a certain table.
- E. Once created, a sequence is automatically used in all INSERT and UPDATE statements.

**Question: 238 (K)**

Which object privileges can be granted on a view?

- A. none
- B. DELETE, INSERT, SELECT
- C. ALTER, DELETE, INSERT, SELECT
- D. DELETE, INSERT, SELECT, UPDATE

**Question: 239 (F)**

Examine the SQL statement that creates ORDERS table:

```
CREATE TABLE orders (SER_NO NUMBER UNIQUE, ORDER_ID NUMBER,  
ORDER_DATE DATE NOT NULL, STATUS VARCHAR2(10) CHECK (status IN  
( 'CREDIT', 'CASH')), PROD_ID NUMBER REFERENCES PRODUCTS(PRODUCT_ID),  
ORD_TOTAL NUMBER, PRIMARY KEY (order_id, order_date));
```

For which columns would an index be automatically created when you execute the above SQL statement?  
(Choose two.)

- A. SER\_NO
- B. ORDER\_ID
- C. STATUS
- D. PROD\_ID
- E. ORD\_TOTAL
- F. composite index on ORDER\_ID and ORDER\_DATE

**Question: 240 (C)**

What is true of using group functions on columns that contain NULL values?

- A. Group functions on columns ignore NULL values.
- B. Group functions on columns returning dates include NULL values.
- C. Group functions on columns returning numbers include NULL values.
- D. Group functions on columns cannot be accurately used on columns that contain NULL values.
- E. Group functions on columns include NULL values in calculations if you use the keyword INC\_NULLS.

**Question: 241 (C)**

The STUDENT\_GRADES table has these columns:

|              |             |
|--------------|-------------|
| STUDENT_ID   | NUMBER(12)  |
| SEMESTER_END | DATE        |
| GPA          | NUMBER(4,3) |

Which statement finds the highest grade point average (GPA) per semester?

- A. `SELECT MAX(gpa) FROM student_grades WHERE gpa IS NOT NULL;`
- B. `SELECT (gpa) FROM student_grades GROUP BY semester_end WHERE gpa IS NOT NULL;`

- C.                   SELECT MAX(gpa) FROM student\_grades WHERE gpa IS NOT NULL GROUP BY semester\_end;
- D.   SELECT MAX(gpa) GROUP BY semester\_end WHERE gpa IS NOT NULL FROM student\_grades;
- E.                   SELECT MAX(gpa) FROM student\_grades GROUP BY semester\_end WHERE gpa IS NOT NULL;

**Question: 242 (E)**

In which four clauses can a subquery be used? (Choose four.)

- A. in the INTO clause of an INSERT statement
- B. in the FROM clause of a SELECT statement
- C. in the GROUP BY clause of a SELECT statement
- D. in the WHERE clause of a SELECT statement
- E. in the SET clause of an UPDATE statement
- F. in the VALUES clause of an INSERT statement

**Question: 243 (H)**

Examine this statement:

```
SELECT student_id, gpa FROM student_grades WHERE gpa > &&value;
```

You run the statement once, and when prompted you enter a value of 2.0. A report is produced. What happens when you run the statement a second time?

- A. An error is returned.
- B. You are prompted to enter a new value.
- C. A report is produced that matches the first report produced.
- D. You are asked whether you want a new value or if you want to run the report based on the previous value.

**Question: 244 (C)**

Which SQL statement returns a numeric value?

- A.                   SELECT ADD\_MONTHS(MAX(hire\_Date), 6) FROM EMP;
- B.                   SELECT ROUND(hire\_date)FROM EMP;
- C.                   SELECT sysdate-hire\_date FROM EMP;
- D.                   SELECT TO\_NUMBER(hire\_date + 7)FROM EMP;

**Question: 245 (J)**

What are two reasons to create synonyms? (Choose two.)

- A. You have too many tables.
- B. Your tables are too long.
- C. Your tables have difficult names.
- D. You want to work on your own tables.
- E. You want to use another schema's tables.
- F. You have too many columns in your tables.

**Question: 246 (I)**

What is true about updates through a view?

- A. You cannot update a view with group functions.
- B. When you update a view group functions are automatically computed.
- C. When you update a view only the constraints on the underlying table will be in effect.
- D. When you update a view the constraints on the views always override the constraints on the underlying tables.

**Question: 247 (E)**

You need to write a SQL statement that returns 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.

Which statement accomplishes this task?

- A. SELECT a.emp\_name, a.sal, b.dept\_id, MAX(sal) FROM employees a, departments b WHERE a.dept\_id = b.dept\_id AND a.sal < MAX(sal) GROUP BY b.dept\_id;
- B. SELECT a.emp\_name, a.sal, a.dept\_id, b.maxsal FROM employees a, (SELECT dept\_id, MAX(sal) maxsal FROM employees GROUP BY dept\_id) b WHERE a.dept\_id = b.dept\_id AND a.sal < b.maxsal;
- C. SELECT a.emp\_name, a.sal, a.dept\_id, b.maxsal FROM employees a WHERE a.sal < (SELECT MAX(sal) maxsal FROM employees b GROUP BY dept\_id);
- D. SELECT emp\_name, sal, dept\_id, maxsal FROM employees, (SELECT dept\_id, MAX(sal) maxsal FROM employees GROUP BY dept\_id) WHERE a.sal < maxsal;

**Question: 248 (B)**

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)  |          |

A promotional sale is being advertised to the customers in France. Which WHERE clause identifies customers that are located in France?

- A. WHERE lower(country\_address) = "france"
- B. WHERE lower(country\_address) = 'france'
- C. WHERE lower(country\_address) IS 'france'
- D. WHERE lower(country\_address) = '%france%'
- E. WHERE lower(country\_address) LIKE %france%

**Question: 249 (H)**

Which are iSQL\*Plus commands? (Choose all that apply.)

- A. INSERT
- B. UPDATE
- C. SELECT
- D. DESCRIBE
- E. DELETE
- F. RENAME

**Question: 250 (C)**

Examine the description of the CUSTOMERS table:

|                  |               |          |
|------------------|---------------|----------|
| 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 statement returns the city address and the number of customers in the cities Los Angeles or San Francisco?



- A.       SELECT city\_address,  
COUNT(\*) FROM customers  
WHERE city\_address IN ( 'Los Angeles', 'San Fransisco');
- B.       SELECT city\_address,  
COUNT (\*) FROM customers  
WHERE city address IN ( 'Los Angeles', 'San Fransisco')
- C.       SELECT city\_address,  
COUNT(customer\_id) FROM customers  
WHERE city\_address IN ( 'Los Angeles', 'San Fransisco')
- D.       SELECT city\_address, COUNT  
(customer\_id) FROM customers  
GROUP BY city\_address IN ( 'Los Angeles', 'San Fransisco');

**Question: 251 (I)**

What does the FORCE option for creating a view do?

- A. creates a view with constraints
- B. creates a view even if the underlying parent table has constraints
- C. creates a view in another schema even if you don't have privileges
- D. creates a view regardless of whether or not the base tables exist

**Question: 252 (H)**

A data manipulation language statement \_\_\_\_.

- A. completes a transaction on a table
- B. modifies the structure and data in a table
- C. modifies the data but not the structure of a table
- D. modifies the structure but not the data of a table

**Question: 253 (C)**

You need to perform these tasks:

Create and assign a MANAGER role to Blake and Clark  
Grant CREATE TABLE and CREATE VIEW privileges to Blake and Clark

Which set of SQL statements achieves the desired results?

- A. CREATE ROLE manager;  
GRANT create table, create view  
TO manager;  
GRANT manager TO BLAKE,CLARK;
- B. CREATE ROLE manager;  
GRANT create table, create voew  
TO manager;  
GRANT manager ROLE TO BLAKE,CLARK;
- C. GRANT manager ROLE TO BLAKE,CLARK;  
GRANT create table, create voew  
TO BLAKE CLARK;

**Question: 254 (K)**

The DBA issues this SQL command:

CREATE USER scott IDENTIFIED by tiger;

What privileges does the user Scott have at this point?

- A. no privileges
- B. only the SELECT privilege
- C. only the CONNECT privilege

D. all the privileges of a default user

**Question: 255 (E)**

Exhibit:

| LAST_NAME | DEPARTMENT_ID | SALARY |
|-----------|---------------|--------|
| Getz      | 10            | 3000   |
| Davis     | 20            | 1500   |
| King      | 20            | 2200   |
| Davis     | 30            | 6000   |
| ...       |               |        |

Examine the data in the EMPLOYEES table.

Examine the subquery:

```
SELECT last_name
FROM employees
WHERE salary IN (SELECT MAX(salary)
FROM employees
GROUP BY department_id);
```

Which statement is true?

- A. The SELECT statement is syntactically accurate.
- B. The SELECT statement does not work because there is no HAVING clause.
- C. The SELECT statement does not work because the column specified in the GROUP BY clause is not in the SELECT list.
- D. The SELECT statement does not work because the GROUP BY clause should be in the main query and not in the subquery.

**Question: 256 (K)**

Examine the statement:

```
GRANT select, insert, update
ON student_grades
TO manager
WITH GRANT OPTION;
```

Which two are true? (Choose two.)

- A. MANAGER must be a role.
- B. It allows the MANAGER to pass the specified privileges on to other users.
- C. It allows the MANAGER to create tables that refer to the STUDENT\_GRADES table.
- D. It allows the MANAGER to apply all DML statements on the STUDENT\_GRADES table.
- E. It allows the MANAGER the ability to select from, insert into, and update the STUDENT\_GRADES table.
- F. It allows the MANAGER the ability to select from, delete from, and update the STUDENT\_GRADES table.

**Question: 257 (I)**

Which best describes an inline view?

- A. a schema object
- B. a subquery that can contain an ORDER BY clause
- C. another name for a view that contains group functions
- D. a subquery that is part of the FROM clause of another query

**Question: 258 (D)**

Examine the structure of the EMPLOYEES and DEPARTMENTS tables:

**EMPLOYEES**

|               |        |
|---------------|--------|
| EMPLOYEE_ID   | NUMBER |
| DEPARTMENT_ID | NUMBER |

|            |              |
|------------|--------------|
| MANAGER_ID | NUMBER       |
| LAST_NAME  | VARCHAR2(25) |

### **DEPARTMENTS**

|                 |              |
|-----------------|--------------|
| DEPARTMENT_ID   | NUMBER       |
| MANAGER_ID      | NUMBER       |
| DEPARTMENT_NAME | VARCHAR2(35) |
| LOCATION_ID     | NUMBER       |

You want to create a report displaying employee last names, department names, and locations. Which query should you use to create an equi-join?

- A. `SELECT last_name, department_name, location_id FROM employees , departments ;`
- B. `SELECT employees.last_name, departments.department_name, departments.location_id FROM employees e, departments D WHERE e.department_id =d.department_id;`
- C. `SELECT e.last_name, d.department_name, d.location_id FROM employees e, departments D WHERE manager_id =manager_id;`
- D. `SELECT e.last_name, d.department_name, d.location_id FROM employees e, departments D WHERE e.department_id =d.department_id;`

### **Question: 259 (B)**

The PRODUCTS table has these columns:

|              |              |
|--------------|--------------|
| PRODUCT_ID   | NUMBER(4)    |
| PRODUCT_NAME | VARCHAR2(45) |
| PRICE        | NUMBER(8,2)  |

Evaluate this SQL statement:

```
SELECT * FROM PRODUCTS ORDER BY price, product_name;
```

What is true about the SQL statement?

- A. The results are not sorted.
- B. The results are sorted numerically.
- C. The results are sorted alphabetically.
- D. The results are sorted numerically and then alphabetically.

### **Question: 260 (D)**

In which two cases would you use an outer join? (Choose two.)

- A. The tables being joined have NOT NULL columns.
- B. The tables being joined have only matched data.
- C. The columns being joined have NULL values.
- D. The tables being joined have only unmatched data.
- E. The tables being joined have both matched and unmatched data.
- F. Only when the tables have a primary key/foreign key relationship.

### **Question: 261 (G)**

Which constraint can be defined only at the column level?

- A. UNIQUE
- B. NOT NULL
- C. CHECK
- D. PRIMARY KEY
- E. FOREIGN KEY

**Question: 262 (H)**

Examine the structure of the EMPLOYEES table:

|               |              |         |
|---------------|--------------|---------|
| EMPLOYEE_ID   | NUMBER       | NOT NUL |
| EMP_NAME      | VARCHAR2(30) |         |
| JOB_ID        | VARCHAR2(20) |         |
| SAL           | NUMBER       |         |
| MGR_ID        | NUMBER       |         |
| DEPARTMENT_ID | NUMBER       |         |

You want to create a SQL script file that contains an INSERT statement. When the script is run, the INSERT statement should insert a row with the specified values into the EMPLOYEES table. The INSERT statement should pass values to the table columns as specified below:

|                                |                                                                                                                                                       |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| EMPLOYEE_ID:                   | Next value from the sequence                                                                                                                          |
| EMP_ID_SEQEMP_NAME and JOB_ID: | As specified by the user during run time, through substitution variables                                                                              |
| SAL:                           | 2000                                                                                                                                                  |
| MGR_ID:                        | No value                                                                                                                                              |
| DEPARTMENT_ID:                 | Supplied by the user during run time throughsubstitution variable. The INSERT statement should fail if the user supplies a value other than 20 or 50. |

Which INSERT statement meets the above requirements?

- A. INSERT INTO employeesVALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- B. INSERT INTO employeesVALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did IN (20,50));
- C. INSERT INTO (SELECT \* FROM employees WHERE department\_id IN (20,50)) VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- D. INSERT INTO (SELECT \* FROM employees WHERE department\_id IN (20,50) WITH CHECK OPTION)VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- E. INSERT INTO (SELECT \* FROM employees WHERE (department\_id = 20 AND department\_id = 50) WITH CHECK OPTION )VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);

**Question: 263 (K)**

The user Alice wants to grant all users query privileges on her DEPT table. Which SQL statement accomplishes this?

- A. GRANT select ON dept TO ALL\_USERS;
- B. GRANT select ON dept TO ALL;
- C. GRANT QUERY ON dept TO ALL\_USERS
- D. GRANT select ON dept TO PUBLIC;

**Question: 264 (G)**

Which view should a user query to display the columns associated with the constraints on a table owned by the user?

- A. USER\_CONSTRAINTS
- B. USER\_OBJECTS
- C. ALL\_CONSTRAINTS
- D. USER\_CONS\_COLUMNS
- E. USER\_COLUMNS

**Question: 265 (K)**

Examine these statements:

```
CREATE ROLE registrar;  
GRANT UPDATE ON student_grades TO registrar; GRANT  
registrar to user1, user2, user3;
```

What does this set of SQL statements do?

- A. The set of statements contains an error and does not work.
- B. It creates a role called REGISTRAR, adds the MODIFY privilege on the STUDENT\_GRADES object to the role, and gives the REGISTRAR role to three users.
- C. It creates a role called REGISTRAR, adds the UPDATE privilege on the STUDENT\_GRADES object to the role, and gives the REGISTRAR role to three users.
- D. It creates a role called REGISTRAR, adds the UPDATE privilege on the STUDENT\_GRADES object to the role, and creates three users with the role.
- E. It creates a role called REGISTRAR, adds the UPDATE privilege on three users, and gives the REGISTRAR role to the STUDENT\_GRADES object.
- F. It creates a role called STUDENT\_GRADES, adds the UPDATE privilege on three users, and gives the UPDATE role to the registrar.

**Question: 266 (E)**

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

- A. A subquery should retrieve only one row.
- B. A subquery can retrieve zero or more rows.
- C. A subquery can be used only in SQL query statements.
- D. Subqueries CANNOT be nested by more than two levels.
- E. A subquery CANNOT be used in an SQL query statement that uses group functions.
- F. When a subquery is used with an inequality comparison operator in the outer SQL statement, the column list in the SELECT clause of the subquery should contain only one column.

**Question: 267 (G)**

You need to design a student registration database that contains several tables storing academic information. The STUDENTS table stores information about a student. The STUDENT\_GRADES table stores information about the student's grades. Both of the tables have a column named STUDENT\_ID. The STUDENT\_ID column in the STUDENTS table is a primary key. You need to create a foreign key on the STUDENT\_ID column of the STUDENT\_GRADES table that points to the STUDENT\_ID column of the STUDENTS table. Which statement creates the foreign key?

- A. 

```
CREATE TABLE student_grades (student_id NUMBER(12), semester_end DATE, gpa  
NUMBER(4,3), CONSTRAINT student_id_fk REFERENCES (student_id) FOREIGN KEY  
students(student_id));
```
- B. 

```
CREATE TABLE student_grades(student_id  
NUMBER(12), semester_end DATE, gpa NUMBER(4,3), student_id_fk FOREIGN KEY  
(student_id) REFERENCES students(student_id));
```
- C. 

```
CREATE TABLE student_grades(student_id NUMBER(12), semester_end DATE,  
gpa NUMBER(4,3), CONSTRAINT FOREIGN KEY (student_id) REFERENCES  
students(student_id));
```
- D. 

```
CREATE TABLE student_grades(student_id NUMBER(12), semester_end DATE, gpa  
NUMBER(4,3), CONSTRAINT student_id_fk FOREIGN KEY (student_id) REFERENCES  
students(student_id));
```

**Question: 268 (H)**

Evaluate the SQL statement:  
TRUNCATE TABLE DEPT;

Which three are true about the SQL statement? (Choose three.)

- A. It releases the storage space used by the table.
- B. It does not release the storage space used by the table.
- C. You can roll back the deletion of rows after the statement executes.

- D. You can NOT roll back the deletion of rows after the statement executes.
- E. An attempt to use DESCRIBE on the DEPT table after the TRUNCATE statement executes will display an error.
- F. You must be the owner of the table or have DELETE ANY TABLE system privileges to truncate the DEPT table

**Question: 269 (G)**

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

- A. The UNIQUE constraint does not permit a null value for the column.
- B. A UNIQUE index gets created for columns with PRIMARY KEY and UNIQUE constraints.
- C. The PRIMARY KEY and FOREIGN KEY constraints create a UNIQUE index.
- D. The NOT NULL constraint ensures that null values are not permitted for the column.

**Question: 270 (H)**

Which three are true? (Choose three.)

- A. A MERGE statement is used to merge the data of one table with data from another.
- B. A MERGE statement replaces the data of one table with that of another.
- C. A MERGE statement can be used to insert new rows into a table.
- D. A MERGE statement can be used to update existing rows in a table.

**Question: 271 (F)**

Which is a valid CREATE TABLE statement?

- A. CREATE TABLE EMP9\$# AS (empid number(2));
- B. CREATE TABLE EMP\*123 AS (empid number(2));
- C. CREATE TABLE PACKAGE AS (packid number(2));
- D. CREATE TABLE 1EMP\_TEST AS (empid number(2));

**Question: 272 (A)**

A SELECT statement can be used to perform these three functions:

- Choose rows from a table.
- Choose columns from a table.
- Bring together data that is stored in different tables by creating a link between them.

Which set of keywords describes these capabilities?

- A. difference, projection, join
- B. selection, projection, join
- C. selection, intersection, join
- D. intersection, projection, join
- E. difference, projection, product

**Question: 273 (C)**

Which four are types of functions available in SQL? (Choose 4)

- A. string
- B. character
- C. integer
- D. calendar
- E. numeric
- F. translation
- G. date
- H. conversion

**Question: 274 (J)**

Examine the structure of the EMPLOYEES table:

|               |              |                                                              |
|---------------|--------------|--------------------------------------------------------------|
| EMPLOYEE_ID   | NUMBER       | NOT NULL, Primary Key                                        |
| EMP_NAME      | VARCHAR2(30) |                                                              |
| JOB_ID        | NUMBER       |                                                              |
| SAL           | NUMBER       |                                                              |
| MGR_ID        | NUMBER       | References EMPLOYEE_ID column                                |
| DEPARTMENT_ID | NUMBER       | Foreign key to DEPARTMENT_ID column of the DEPARTMENTS table |

You created a sequence called EMP\_ID\_SEQ in order to populate sequential values for the EMPLOYEE\_ID column of the EMPLOYEES table. Which two statements regarding the EMP\_ID\_SEQ sequence are true? (Choose two.)

- A. You cannot use the EMP\_ID\_SEQ sequence to populate the JOB\_ID column.
- B. The EMP\_ID\_SEQ sequence is invalidated when you modify the EMPLOYEE\_ID column.
- C. The EMP\_ID\_SEQ sequence is not affected by modifications to the EMPLOYEES table.
- D. Any other column of NUMBER data type in your schema can use the EMP\_ID\_SEQ sequence.
- E. The EMP\_ID\_SEQ sequence is dropped automatically when you drop the EMPLOYEES table.
- F. The EMP\_ID\_SEQ sequence is dropped automatically when you drop the EMPLOYEE\_ID column.

**Question: 275 (B)**

Evaluate these two SQL statements:

```
SELECT last_name, salary, hire_date FROM EMPLOYEES ORDER BY salary DESC;
SELECT last_name, salary, hire_date FROM EMPLOYEES ORDER BY 2 DESC;
```

What is true about them?

- A. The two statements produce identical results.
- B. The second statement returns a syntax error.
- C. There is no need to specify DESC because the results are sorted in descending order by default.
- D. The two statements can be made to produce identical results by adding a column alias for the salary column in the second SQL statement.

**Question: 276 (H)**

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

**EMPLOYEES**

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| HIRE_DATE   | DATE         |             |

**NEW\_EMPLOYEES**

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| NAME        | VARCHAR2(60) |             |

Which MERGE statement is valid?

- A. 

```
MERGE INTO
new_employees c USING
employees e
ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN
UPDATE SET
```

```

C.name = e.first_name ||','|| e.last_name
WHEN NOT MATCHED THEN
INSERT
value
S(e.employee_id, e.first_name ||','||
e.last_name);
B.          MERGE
new_employees c USING
employees e
ON (c.employee_id = e.employee_id)
WHEN EXISTS THEN
UPDATE SET
C.name = e.first_name ||','|| e.last_name
WHEN NOT MATCHED THEN INSERT
valueS(e.employee_id, e.first_name ||','||
e.last_name);
C.          MERGE INTO new_employees
cUSING employees e ON (c.employee_id =
e.employee_id)
WHEN EXISTS THEN
UPDATE SET
C.name = e.first_name ||','|| e.last_name
WHEN NOT MATCHED THEN
INSERT
value
S(e.employee_id, e.first_name ||','||
e.last_name);
D.          MERGE new_employees c
FROM employees e ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN
UPDATE SET
c.name = e.first_name ||','|| e.last_name
WHEN NOT MATCHED THEN
INSERT INTO
new_employees valueS(e.employee_id, e.first_name ||','||
e.last_name);

```

**Question: 277 (C)**

Which SQL statement displays the date March 19, 2001 in a format that appears as “Nineteenth of March 2001 12:00:00 AM”?

- A. SELECT  
TO\_CHAR(TO\_DATE(‘19-Mar-2001’, ‘DD-Mon-YYYY’), ‘fmDdspth  
“of” Month YYYY fmHH:MI:SS AM’) NEW\_DATE  
FROM dual;
- B. SELECT  
TO\_CHAR(TO\_DATE(‘19-Mar -2001’, ‘DD-Mon -YYYY’),  
‘Ddspth “of” Month YYYY fmHH:MI:SS AM’) NEW\_DATE  
FROM dual;
- C. SELECT  
TO\_CHAR(TO\_DATE(‘19-Mar -2001’, ‘DD-Mon -YYYY’), ‘fmDdspth “of”  
Month YYYY  
HH:MI:SS AM’) NEW\_DATE
- D. SELECT  
TO\_CHAR(TO\_DATE(‘19-Mar -2001’, ‘DD-Mon -YYYY’), ‘fmDdspth “of”  
Month YYYYfmtHH:HI:SS AM’) NEW\_DATE FROM dual;

**Question: 278 (K)**

Which one is a system privilege?



- A. SELECT
- B. DELETE
- C. EXECUTE
- D. ALTER TABLE
- E. CREATE TABLE

**Question: 279 (A)**

Which statement correctly describes SQL and /SQL\*Plus?

- A. Both SQL and /SQL\*plus allow manipulation of values in the database.
- B. /SQL\*Plus recognizes SQL statements and sends them to the server; SQL is the Oracle proprietary interface for executing SQL statements.
- C. /SQL\*Plus is a language for communicating with the Oracle server to access data; SQL recognizes SQL statements and sends them to the server.
- D. SQL manipulates data and table definitions in the database; /SQL\*Plus does not allow manipulation of values in the database.

**Question: 280 (D)**

Exhibit:

| EMPLOYEES   |           |               |            |
|-------------|-----------|---------------|------------|
| EMPLOYEE_ID | LAST_NAME | DEPARTMENT_ID | MANAGER_ID |
| 100         | Getz      | 10            | 103        |
| 101         | Davis     | 20            | 104        |
| 102         | King      | 20            | 104        |
| 103         | Davis     | 30            |            |
| 104         | John      |               | 103        |

  

| DEPARTMENTS   |                 |
|---------------|-----------------|
| DEPARTMENT_ID | DEPARTMENT_NAME |
| 10            | Sales           |
| 20            | Marketing       |
| 30            | Accounts        |
| 40            | Administration  |

Examine the data in the EMPLOYEES and DEPARTMENTS tables.

You want to retrieve all employees' last names, along with their manager's last names and their department names. Which query would you use?

- A. SELECT last\_name, manager\_id,  
department\_name FROM employees e  
FULL OUTER JOIN departments d ON (e.department\_id = d.department\_id);
- B. SELECT e.last\_name, m.last\_name,  
department\_name FROM employees e  
LEFT OUTER JOIN employees m on ( e.managaer\_id = m.employee\_id) LEFT  
OUTER JOIN departments d ON (e.department\_id = d.department\_id);
- C. SELECT e.last\_name, m.last\_name,  
department\_name FROM employees e  
RIGHT OUTER JOIN employees m on ( e.manager\_id = m.employee\_id) LEFT  
OUTER JOIN departments d ON (e.department\_id = d.department\_id);
- D. SELECT e.last\_name, m.last\_name,  
department\_name FROM employees e  
LEFT OUTER JOIN employees m on ( e.manager\_id = m.employee\_id) RIGHT  
OUTER JOIN departments d ON (e.department\_id = d.department\_id);
- E. SELECT e.last\_name, m.last\_name,  
department\_name FROM employees e  
RIGHT OUTER JOIN employees m on ( e.manager\_id = m.employee\_id) RIGHT  
OUTER JOIN departments d ON (e.department\_id = d.department\_id)

F.           SELECT last\_name, manager\_id,  
department\_name FROM employees e  
JOIN departments d ON (e.department\_id = d.department\_id) ;

**Question: 281 (H)**

Examine the structure of the EMPLOYEES table:

EMPLOYEE\_ID NUMBER Primary Key  
FIRST\_NAME VARCHAR2(25)  
LAST\_NAME VARCHAR2(25)  
HIRE\_DATE DATE

Which INSERT statement is valid?

- A.           INSERT INTO employees (employee\_id, first\_name,  
last\_name, hire\_date) VALUES ( 1000, 'John', 'Smith', '01/01/01');
- B.           INSERT INTO employees(employee\_id, first\_name,  
last\_name,                   hire\_date) VALUES ( 1000, 'John', 'Smith', '01  
January 01');
- C.           INSERT INTO employees(employee\_id, first\_name,  
last\_name, Hire\_date) VALUES ( 1000, 'John', 'Smith', To\_date('01/01/01'));
- D.           INSERT INTO employees(employee\_id, first\_name,  
last\_name, hire\_date) VALUES ( 1000, 'John', 'Smith', 01-Jan-01);

**Question: 282 (I)**

User Mary has a view called EMP\_DEPT\_LOC\_VU that was created based on the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables. She has the privilege to create a public synonym, and would like to create a synonym for this view that can be used by all users of the database.

Which SQL statement can Mary use to accomplish that task?

- A.           CREATE PUBLIC SYNONYM  
EDL\_VU ON  
emp\_dept\_loc\_vu;
- B.           CREATE PUBLIC  
SYNONYM  
EDL:VU FOR mary  
(emp\_dept\_loc\_vu);
- C.           CREATE PUBLIC SYNONYM  
EDL\_VU FOR  
emp\_dept\_loc\_vu;
- D.           CREATE SYNONYM EDL\_VU ON emp\_dept\_loc\_vu  
FOR EACH USER;
- E.           CREATE SYNONYM EDL\_VU  
FOR EACH USER  
ON emp\_dept\_loc\_vu;
- F.           CREATE PUBLIC  
SYNONYM EDL\_VU ON  
emp\_dept\_loc\_vu  
FOR ALL USERS;

**Question: 283 (E)**

**Exhibit**

| EMP_ID | DEPT_ID | COMMISSION |
|--------|---------|------------|
| 1      | 10      | 500        |
| 2      | 20      | 1000       |
| 3      | 10      |            |
| 4      | 10      | 600        |
| 5      | 30      | 800        |
| 6      | 30      | 200        |
| 7      | 10      |            |
| 8      | 20      | 300        |

The COMMISSION column shows the monthly commission earned by the employee. Which two tasks would require subqueries or joins in order to be performed in a single step? (Choose two.)

- A. listing the employees who earn the same amount of commission as employee 3
- B. finding the total commission earned by the employees in department 10
- C. finding the number of employees who earn a commission that is higher than the average commission of the company
- D. listing the departments whose average commission is more than 600
- E. listing the employees who do not earn commission and who are working for department 20 in descending order of the employee ID
- F. listing the employees whose annual commission is more than 6000

**Question: 284 (I)**

Examine the statement:

Create synonym emp for hr.employees;  
What happens when you issue the statement?

- A. An error is generated.
- B. You will have two identical tables in the HR schema with different names.
- C. You create a table called employees in the HR schema based on you EMP table.
- D. You create an alternative name for the employees table in the HR schema in your own schema.

**Question: 285 (H)**

Examine the structure of the EMPLOYEES table:

EMPLOYEE\_ID NUMBER Primary Key  
FIRST\_NAME VARCHAR2 (25)  
LAST\_NAME VARCHAR2 (25)  
HIRE\_DATE DATE

Which UPDATE statement is valid?

- A. UPDATE employees  
SET first\_name = 'John'  
SET last\_name = 'Smith'  
WHERE employee\_id = 180;
- B. UPDATE employees  
SET first\_name = 'John',  
SET last\_name = 'Smoth'  
WHERE employee\_id = 180;
- C. UPDATE employee  
SET first\_name = 'John'  
AND last\_name = 'Smith'  
WHERE employee\_id = 180;
- D. UPDATE employee  
SET first\_name = 'John', last\_name = 'Smith'  
WHERE employee\_id = 180;

**Question: 286 (J)**

What is true about sequences?

- A. The start value of the sequence is always 1.

- B. A sequence always increments by 1.
- C. The minimum value of an ascending sequence defaults to 1.
- D. The maximum value of descending sequence defaults to 1.

**Question: 287 (B)**

The STUDENT\_GRADES table has these columns

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

Which statement finds students who have a grade point average (GPA) greater than 3.0 for the calendar year 2001?

- A.       SELECT  
student\_id, gpa FROM  
student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001' OR  
gpa > 3.;
- B.       SELECT  
student\_id, gpa FROM  
student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001'  
AND gpa gt 3.0;
- C.       SELECT student\_id, gpa  
FROM student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001'  
AND gpa > 3.0;
- D.       SELECT  
student\_id, gpa FROM  
student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001'  
AND gpa > 3.0;
- E.       SELECT  
student\_id, gpa FROM  
student\_grades  
WHERE semester\_end > '01-JAN-2001' OR semester\_end < '31-DEC-2001' AND  
gpa >= 3.0;

**Question: 288 (D)**

**Exhibit:**

**ORDERS**

| ORD_ID | ORD_DATE    | CUST_ID | ORD_TOTAL |
|--------|-------------|---------|-----------|
| 100    | 12-JAB-2000 | 15      | 10000     |
| 101    | 09-MAR-2000 | 40      | 8000      |
| 102    | 09-MAR-2000 | 35      | 12500     |
| 103    | 15-MAR-2000 | 15      | 12500     |
| 104    | 25-JUN-2000 | 15      | 6000      |
| 105    | 18-JUL-2000 | 20      | 5000      |
| 106    | 18-JUL-2000 | 35      | 7000      |
| 108    | 04-AUG-2000 | 10      | 8000      |

**CUSTOMERS**

| CUST_ID | CUST_NAME | CITY |
|---------|-----------|------|
|---------|-----------|------|

|    |        |               |
|----|--------|---------------|
| 10 | Smith  | Los Angeles   |
| 15 | Bob    | San Francisco |
| 20 | Martin | Chicago       |
| 25 | Mary   | New York      |
| 30 | Rina   | Chicago       |
| 35 | Smith  | New York      |
| 40 | Linda  | New York      |

Evaluate this SQL statement:

```
SELECT cust_id, ord_total
FROM orders
WHERE ord_total > ANY (SELECT ord_total
FROM orders
WHERE cust_id IN (SELECT cust_id
FROM customers
WHERE city LIKE
'New York'));
```

What is the result when the above query is executed?

A.

| CUST_ID | ORD_TOTAL |
|---------|-----------|
| 15      | 10000     |
| 40      | 8000      |
| 35      | 12500     |
|         | 12500     |
|         | 8000      |

B.

| CUST_ID | ORD_TOTAL |
|---------|-----------|
| 15      | 10000     |
|         | 12500     |
|         | 12000     |

C.

| CUST_ID | ORD_TOTAL |
|---------|-----------|
| 15      | 10000     |
| 40      | 8000      |
| 15      | 12000     |
| 15      | 6000      |
| 20      | 5000      |
| 35      | 7000      |
| 20      | 6500      |
| 10      | 8000      |

D.

| CUST_ID | ORD_TOTAL |
|---------|-----------|
| 15      | 6000      |
| 20      | 5000      |
|         | 6500      |

E. The query returns no rows.

F. The query fails because ANY is not a valid operator with a subquery.

**Question: 289 (E)**

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

- A. A single row subquery can retrieve data from only one table.
- B. 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.
- C. A SQL query statement can display data from table B that is referred to in its subquery, without including B in its own FROM clause.
- D. A single row subquery can retrieve data from more than one table.
- E. A single row subquery cannot be used in a condition where the LIKE operator is used for comparison. F. A multiple-row subquery cannot be used in an INSERT statement to insert multiple rows at a time.

**Question: 290 (E)**

What is true regarding subqueries?

- A. The inner query always sorts the results of the outer query
- B. The outer query always sorts the results of the inner query
- C. The outer query must return a value to the outer query
- D. The inner query returns a value to the outer query
- E. The inner query must always return a value or the outer query will give an error

**Question: 291 (C)**

For which action can you use the TO\_DATE function?

- A. Convert any date literal to a date
- B. Convert any numeric literal to a date
- C. Convert any character literal to a date
- D. Convert any date to a character literal
- E. Format '10-JAN-99' to 'January 10 1999'

**Question: 292 (B)**

**EXHIBIT, Emp Table**

| EMP_ID | DEPT_ID | COMMISSION |
|--------|---------|------------|
| 1      | 10      | 500        |
| 2      | 20      | 1000       |
| 3      | 10      |            |
| 4      | 10      | 800        |
| 5      | 30      | 800        |
| 6      | 30      | 200        |
| 7      | 10      |            |
| 8      | 20      | 300        |

**Exhibit A**

| EMP_ID | DEPT_ID | COMMISSION |
|--------|---------|------------|
| 3      | 10      |            |
| 7      | 10      |            |

**Exhibit B**

| EMP_ID | DEPT_ID | COMMISSION |
|--------|---------|------------|
| 7      | 10      |            |

Examine the data from the EMP table.

Evaluate this SQL statement:

```
SELECT * FROM emp
WHERE commission = (SELECT commission
FROM emp
WHERE emp_id=3);
```

What is the result when the query is executed?

- A. Exhibit A
- B. Exhibit B
- C. The query returns no rows
- D. The query fails because the outer query is retrieving more than one column
- E. The query fails because both the inner and outer queries are retrieving data from the same table.

**Question: 293**

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

- A. A main query can have more than one subquery.
- B. A subquery can have more than one main query.
- C. The subquery and main query must retrieve data from the same table.
- D. The subquery and main query can retrieve data from different tables.
- E. Only one column or expression can be compared between the subquery and main query.
- F. Multiple columns or expression can be compared between the subquery and main query.

**Question: 294**

Evaluate this SQL statement:

```
SELECT e.emp_name, d.dept_name
FROM employees e
JOIN departments d
USING (department_id)
WHERE d.department_id NOT IN (10,40)
ORDER BY dept_name;
```

The statement fails when executed. Which change fixes the error?

- A. remove the ORDER BY clause
- B. remove the table alias prefix from the WHERE clause
- C. remove the table alias from the SELECT clause
- D. prefix the column in the USING clause with the table alias
- E. prefix the column in the ORDER BY clause with the table alias
- F. replace the condition  
"d.department\_id NOT IN (10,40)"  
in the WHERE clause with  
"d.department\_id <> 10 AND d.department\_id <> 40"

**Question: 295**

Evaluate the SQL statement:

```
SELECT LPAD (salary,10,*)
FROM EMP
WHERE EMP_ID = 1001;
```

If the employee with the EMP\_ID 1001 has a salary of 17000, what is displayed?

- A. 17000.00
- B. 17000\*\*\*\*\*
- C. \*\*\*170.00
- D. \*\*17000.00
- E. an error statement