

1Z0-051: Oracle Database 11g SQL Fundamentals I

1

Examine the structure of the PROGRAMS table:

Name	Null?	Type
PROG_ID	NOT NULL	NUMBER(3)
PROG_COST		NUMBER(8,2)
START_DATE	NOT NULL	DATE
END_DATE		DATE

Which two SQL statements would execute successfully? (Choose two.)

- A. `SELECT NVL(ADD_MONTHS(END_DATE,1),SYSDATE)
FROM programs;`
- B. `SELECT TO_DATE(NVL(SYSDATE-END_DATE,SYSDATE))
FROM programs;`
- C. `SELECT NVL(MONTHS_BETWEEN(start_date,end_date), 'Ongoing')
FROM programs;`
- D. `SELECT NVL(TO_CHAR(MONTHS_BETWEEN(start_date,end_date)), 'Ongoing')
FROM programs;`

2

Where can subqueries be used? (Choose all that apply.)

- A. field names in the `SELECT` statement
- B. the `FROM` clause in the `SELECT` statement
- C. the `HAVING` clause in the `SELECT` statement
- D. the `GROUP BY` clause in the `SELECT` statement
- E. the `WHERE` clause in only the `SELECT` statement
- F. the `WHERE` clause in `SELECT` as well as all DML statements

3

The SQL statements executed in a user session are as follows:

```
SQL> CREATE TABLE product  
      (pcode NUMBER(2),  
       pname VARCHAR2(10));  
SQL> INSERT INTO product VALUES (1, 'pen');  
SQL> INSERT INTO product VALUES (2, 'pencil');  
SQL> SAVEPOINT a;  
SQL> UPDATE product SET pcode = 10 WHERE pcode = 1;  
SQL> SAVEPOINT b;  
SQL> DELETE FROM product WHERE pcode = 2;  
SQL> COMMIT;  
SQL> DELETE FROM product WHERE pcode=10;
```

Which two statements describe the consequences of issuing the `ROLLBACK TO SAVE POINT a` command in the session? (Choose two.)

- A. The rollback generates an error.
- B. No SQL statements are rolled back.
- C. Only the `DELETE` statements are rolled back.
- D. Only the second `DELETE` statement is rolled back.
- E. Both the `DELETE` statements and the `UPDATE` statement are rolled back.

4

The `ORDERS` table belongs to the user `OE`. `OE` has granted the `SELECT` privilege on the `ORDERS` table to the user `HR`.

Which statement would create a synonym `ORD` so that `HR` can execute the following query successfully?

```
SELECT * FROM ord;
```

- A. `CREATE SYNONYM ord FOR orders;` This command is issued by `OE`.
- B. `CREATE PUBLIC SYNONYM ord FOR orders;` This command is issued by `OE`.
- C. `CREATE SYNONYM ord FOR oe.orders;` This command is issued by the database administrator.
- D. `CREATE PUBLIC SYNONYM ord FOR oe.orders;` This command is issued by the database administrator.

5

ORDERS		
Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER(4)
ORDER_DATE	NOT NULL	DATE
ORDER_MODE		VARCHAR2(8)
CUSTOMER_ID	NOT NULL	NUMBER(6)
ORDER_TOTAL		NUMBER(8,2)

CUSTOMERS		
Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(6)
CUST_FIRST_NAME	NOT NULL	VARCHAR2(20)
CUST_LAST_NAME	NOT NULL	VARCHAR2(20)
CREDIT_LIMIT		NUMBER(9,2)
CUST_ADDRESS		VARCHAR2(40)

View the Exhibit and examine the structure of `ORDERS` and `CUSTOMERS` tables.

There is only one customer with the `cust_last_name` column having value `Roberts`. Which `INSERT` statement should be used to add a row into the `ORDERS` table for the customer whose `CUST_LAST_NAME` is `Roberts` and `CREDIT_LIMIT` is `600`?

- A.

```
INSERT INTO orders
VALUES (1,'10-mar-2007', 'direct',
      (SELECT customer_id
       FROM customers
       WHERE cust_last_name='Roberts' AND
        credit_limit=600), 1000);
```
- B.

```
INSERT INTO orders (order_id,order_date,order_mode,
      (SELECT customer_id
       FROM customers
       WHERE cust_last_name='Roberts' AND
        credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &customer_id, 1000);
```
- C.

```
INSERT INTO(SELECT o.order_id, o.order_date,o.order_mode,c.customer_id, o.order_total
      FROM orders o, customers c
      WHERE o.customer_id = c.customer_id
      AND c.cust_last_name='Roberts' ANDc.credit_limit=600 )
VALUES (1,'10-mar-2007', 'direct', (SELECT customer_id
      FROM customers
      WHERE cust_last_name='Roberts' AND
        credit_limit=600), 1000);
```
- D.

```
INSERT INTO orders (order_id,order_date,order_mode,
      (SELECT customer_id
       FROM customers
       WHERE cust_last_name='Roberts' AND
        credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &customer_id, 1000);
```

6

Evaluate the following query:

```
SELECT INTERVAL '300' MONTH,  
INTERVAL '54-2' YEAR TO MONTH,  
INTERVAL '11:12:10.1234567' HOUR TO SECOND  
FROM dual;
```

What is the correct output of the above query?

- A. +25-00 , +54-02, +00 11:12:10.123457
- B. +00-300, +54-02, +00 11:12:10.123457
- C. +25-00 , +00-650, +00 11:12:10.123457
- D. +00-300 , +00-650, +00 11:12:10.123457

7

PROMOTIONS		
PROMO_ID	PROMO_CATEGORY	PROMO_SUBCATEGORY
506	magazine	discount
507	TV	general advt
508	newspaper	discount
509	post	general advt
510	post	discount
511	radio	general advt
512	newspaper	general advt
513	newspaper	discount
514	magazine	general advt
515	newspaper	discount
516	newspaper	general advt

View the Exhibit and examine the data in the PROMOTIONS table.

You need to display all promo categories that do not have 'discount' in their subcategory.

Which two SQL statements give the required result? (Choose two.)

- A.

```
SELECT promo_category  
FROM promotions  
MINUS  
SELECT promo_category  
FROM promotions  
WHERE promo_subcategory = 'discount';
```
- B.

```
SELECT promo_category  
FROM promotions  
INTERSECT  
SELECT promo_category  
FROM promotions  
WHERE promo_subcategory = 'discount';
```
- C.

```
SELECT promo_category  
FROM promotions  
MINUS  
SELECT promo_category  
FROM promotions  
WHERE promo_subcategory <> 'discount';
```
- D.

```
SELECT promo_category  
FROM promotions  
INTERSECT  
SELECT promo_category  
FROM promotions  
WHERE promo_subcategory <> 'discount';
```

CUSTOMERS		
Name	Null?	Type
CUSTNO	NOT NULL	NUMBER(2)
CUSTNAME		VARCHAR2(10)
CUSTADDRESS		VARCHAR2(20)
CUST_CREDIT_LIMIT		NUMBER(5)

GRADES		
Name	Null?	Type
GRADE	NOT NULL	VARCHAR2(1)
STARTVAL		NUMBER(5)
ENDVAL		NUMBER(5)

View the Exhibit and examine the structure of CUSTOMERS and GRADES tables.

You need to display names and grades of customers who have the highest credit limit.

Which two SQL statements would accomplish the task? (Choose two.)

- A. `SELECT custname, grade
FROM customers, grades
WHERE (SELECT MAX(cust_credit_limit)
FROM customers) BETWEEN startval and endval;`
- B. `SELECT custname, grade
FROM customers, grades
WHERE (SELECT MAX(cust_credit_limit)
FROM customers) BETWEEN startval and endval
AND cust_credit_limit BETWEEN startval AND endval;`
- C. `SELECT custname, grade
FROM customers, grades
WHERE cust_credit_limit = (SELECT MAX(cust_credit_limit)
FROM customers)
AND cust_credit_limit BETWEEN startval AND endval;`
- D. `SELECT custname, grade
FROM customers, grades
WHERE cust_credit_limit IN (SELECT MAX(cust_credit_limit)
FROM customers)
AND MAX(cust_credit_limit) BETWEEN startval AND endval;`

Evaluate the following SQL statement:

```
SQL> SELECT cust_id, cust_last_name "Last Name"
FROM customers
WHERE country_id = 10
UNION
SELECT cust_id CUST_NO, cust_last_name
FROM customers
WHERE country_id = 30;
```

Which ORDER BY clauses are valid for the above query? (Choose all that apply.)

- A. `ORDER BY 2,1`
- B. `ORDER BY CUST_NO`
- C. `ORDER BY 2,cust_id`
- D. `ORDER BY "CUST_NO"`
- E. `ORDER BY "Last Name"`

10

You need to generate a list of all customer last names with their credit limits from the `CUSTOMERS` table. Those customers who do not have a credit limit should appear last in the list.

Which two queries would achieve the required result? (Choose two.)

- A.

```
SELECT cust_last_name, cust_credit_limit
FROM customers
ORDER BY cust_credit_limit DESC;
```
- B.

```
SELECT cust_last_name, cust_credit_limit
FROM customers
ORDER BY cust_credit_limit;
```
- C.

```
SELECT cust_last_name, cust_credit_limit
FROM customers
ORDER BY cust_credit_limit NULLS LAST;
```
- D.

```
SELECT cust_last_name, cust_credit_limit
FROM customers
ORDER BY cust_last_name, cust_credit_limit NULLS LAST;
```

11

You need to display the first names of all customers from the `CUSTOMERS` table that contain the character 'e' and have the character 'a' in the second last position.

Which query would give the required output?

- A.

```
SELECT cust_first_name
FROM customers
WHERE INSTR(cust_first_name, 'e') <> 0 AND
      SUBSTR(cust_first_name, -2, 1) = 'a';
```
- B.

```
SELECT cust_first_name
FROM customers
WHERE INSTR(cust_first_name, 'e') <> '' AND
      SUBSTR(cust_first_name, -2, 1) = 'a';
```
- C.

```
SELECT cust_first_name
FROM customers
WHERE INSTR(cust_first_name, 'e') IS NOT NULL AND
      SUBSTR(cust_first_name, 1, -2) = 'a';
```
- D.

```
SELECT cust_first_name
FROM customers
WHERE INSTR(cust_first_name, 'e') <> 0 AND
      SUBSTR(cust_first_name, LENGTH(cust_first_name), -2) = 'a';
```

12

Which `CREATE TABLE` statement is valid?

- A.

```
CREATE TABLE ord_details
(ord_no NUMBER(2) PRIMARY KEY,
item_no NUMBER(3) PRIMARY KEY,
ord_date DATE NOT NULL);
```
- B.

```
CREATE TABLE ord_details
(ord_no NUMBER(2) UNIQUE, NOT NULL,
item_no NUMBER(3),
ord_date DATE DEFAULT SYSDATE NOT NULL);
```
- C.

```
CREATE TABLE ord_details
(ord_no NUMBER(2) ,
item_no NUMBER(3),
ord_date DATE DEFAULT NOT NULL,
CONSTRAINT ord_uq UNIQUE (ord_no),
CONSTRAINT ord_pk PRIMARY KEY (ord_no));
```
- D.

```
CREATE TABLE ord_details
(ord_no NUMBER(2),
item_no NUMBER(3),
ord_date DATE DEFAULT SYSDATE NOT NULL,
CONSTRAINT ord_pk PRIMARY KEY (ord_no, item_no));
```