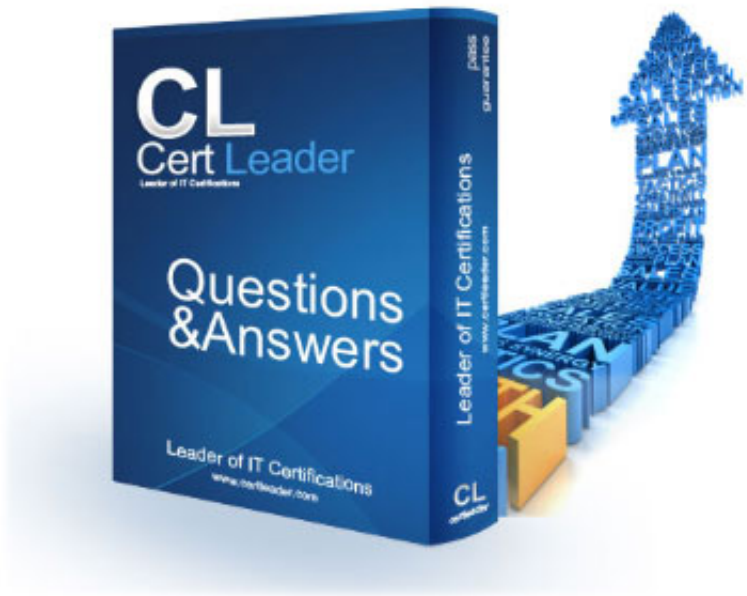
1Z0-061 - Oracle Database 12c SQL Fundamentals

http://www.certleader.com/1Z0-061-dumps.html



1. View the Exhibit and examine the structure of the SALES table.

The following query is written to retrieve all those product IDs from the SALES table that have more than 55000 sold and have been ordered more than 10 times.

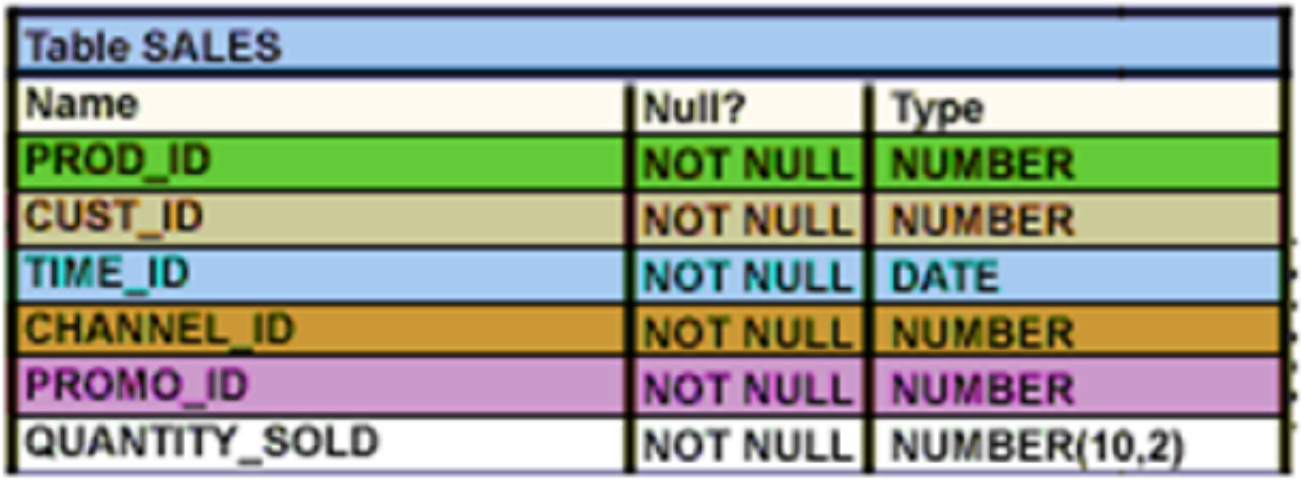


Table SALES		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER
CUST_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
CHANNEL_ID	NOT NULL	NUMBER
PROMO_ID	NOT NULL	NUMBER
QUANTITY_SOLD	NOT NULL	NUMBER(10,2)

Which statement is true regarding this SQL statement?

A. It executes successfully and generates the required result.

B. It produces an error because count(*) should be specified in the SELECT clause also.

C. It produces an error because count{*) should be only in the HAVING clause and not in the WHERE clause.

D. It executes successfully but produces no result because COUNT (prod_id) should be used instead of COUNT (*).

Answer: C

Explanation:

Restricting Group Results with the HAVING Clause

You use the HAVING clause to specify the groups that are to be displayed, thus further

restricting the groups on the basis of aggregate information.

In the syntax, group_condition restricts the groups of rows returned to those groups for

which the specified condition is true.

The Oracle server performs the following steps when you use the HAVING clause:

1. Rows are grouped.

2. The group function is applied to the group.

3. The groups that match the criteria in the HAVING clause are displayed.

The HAVING clause can precede the GROUP BY clause, but it is recommended that you

place the GROUP BY clause first because it is more logical. Groups are formed and group

functions are calculated before the HAVING clause is applied to the groups in the SELECT

list.

Note: The WHERE clause restricts rows, whereas the HAVING clause restricts groups.

2. Evaluate the following query:

```
SQL> SELECT promo_name || q'{'s start date was \}' || promo_begin_date
                          AS "Promotion Launches"
FROM promotions;
```

What would be the outcome of the above query?

A. It produces an error because flower braces have been used.

B. It produces an error because the data types are not matching.

C. It executes successfully and introduces an 's at the end of each PROMO_NAME in the output.

D. It executes successfully and displays the literal "{'s start date was \} * for each row in the output.

Answer: C Explanation:

So, how are words that contain single quotation marks dealt with? There are essentially two mechanisms available. The most popular of these is to add an additional single quotation mark next to each naturally occurring single quotation mark in the character string

Oracle offers a neat way to deal with this type of character literal in the form of the alternative quote (q) operator. Notice that the problem is that Oracle chose the single quote characters as the special pair of symbols that enclose or wrap any other character literal.

These character-enclosing symbols could have been anything other than single quotation marks.

Bearing this in mind, consider the alternative quote (q) operator. The q operator enables you to choose from a set of possible pairs of wrapping symbols for character literals as alternatives to the single quote symbols. The options are any single-byte or multibyte character or the four brackets: (round brackets), {curly braces}, [squarebrackets], or <angle brackets>. Using the q operator, the character delimiter can effectively be changed from a single quotation mark to any other character

The syntax of the alternative quote operator is as follows:

q'delimiter'character literal which may include the single quotes delimiter' where delimiter can be any character or bracket.

Alternative Quote (q) Operator

Specify your own quotation mark delimiter.

Select any delimiter.

Increase readability and usability.

SELECT department_name || q'[Department's Manager Id:]'

|| manager_id

AS "Department and Manager"

FROM departments;

Alternative Quote (q) Operator

Many SQL statements use character literals in expressions or conditions. If the literal itself contains a single quotation mark, you can use the quote (q) operator and select your own quotation mark delimiter.

You can choose any convenient delimiter, single-byte or multi byte, or any of the following character pairs: [ ], { }, ( ), or < >.

In the example shown, the string contains a single quotation mark, which is normally interpreted as a delimiter of a character string. By using the q operator, however, brackets

[] are used as the quotation mark delimiters. The string between the brackets delimiters is interpreted as a literal character string.

3. Evaluate the following SQL statement:

```
SQL> SELECT cust_id, cust_last_name
FROM customers
WHERE cust_credit_limit IN
                        (select cust_credit_limit
                         FROM customers
                         WHERE cust_city ='Singapore');
```

Which statement is true regarding the above query if one of the values generated by the subquery is null?

A. It produces an error.

B. It executes but returns no rows.

C. It generates output for null as well as the other values produced by the subquery.

D. It ignores the null value and generates output for the other values produced by the subquery.

Answer: C

4. Examine the types and examples of relationships that follow:

1. One-to-one a) Teacher to students

2. One-to-many b) Employees to Manager

3. Many-to-one c) Person to SSN

4. Many-to-many d) Customers to products

Which option indicates the correctly matched relationships?

A. 1-a, 2-b, 3-c, and 4-d

B. 1-c, 2-d, 3-a, and 4-b

C. 1-c, 2-a, 3-b, and 4-d

D. 1-d, 2-b, 3-a, and 4-c

Answer: C

5. Examine the data in the CUST_NAME column of the customers table.

You need to display customers' second names where the second name starts with "Mc" or "MC."

```
CUST_NAME
------------------
Renske Ladwig
Jason Mallin
Samuel McCain
Allan MCEwen
Irene Mikkilineni
Julia Nayer
```

Which query gives the required output?

A)
```
SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)
FROM customers
WHERE INITCAP(SUBSTR(cust_name, INSTR(cust_name,' ')+1))='Mc';
```

B)
```
SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)
FROM customers
WHERE INITCAP(SUBSTR(cust_name, INSTR(cust_name,' ')+1)) LIKE 'Mc%';
```

C)
```
SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)
FROM customers
WHERE SUBSTR(cust_name, INSTR(cust_name,' ')+1) LIKE INITCAP('MC%');
```

D)
```
SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)
FROM customers
WHERE INITCAP(SUBSTR(cust_name, INSTR(cust_name,' ')+1)) = INITCAP('MC%');
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

6. View the Exhibit and examine the structure of the promotions table.

Table PROMOTIONS

Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(6)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY_ID	NOT NULL	NUMBER
PROMO_CATEGORY	NOT NULL	VARCHAR2(30)
PROMO_CATEGORY_ID	NOT NULL	NUMBER
PROMO_COST	NOT NULL	NUMBER(10,2)
PROMO_BEGIN_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

Evaluate the following SQL statement:

```
SQL>SELECT promo_name,CASE
                      WHEN promo_cost >=(SELECT AVG(promo_cost)
                                         FROM promotions
                                         WHERE promo_category='TV')
                      THEN 'HIGH'
                      ELSE 'LOW'
                      END COST_REMARK
FROM promotions;
```

Which statement is true regarding the outcome of the above query?

A. It shows COST_REMARK for all the promos in the table.

B. It produces an error because the SUBQUERY gives an error.

C. It shows COST_REMARK for all the promos in the promo category 'TV'

D. It produces an error because SUBQUERIES cannot be used with the case expression.

Answer: A

7. Examine the structure of the transactions table:

Name	Null?	Type
TRANS_ID	NOT NULL	NUMBER(3)
CUST_NAME		VARCHAR2(30)
TRANS_DATE		TIMESTAMP
TRANS_AMT		NUMBER(10,2)

You want to display the date, time, and transaction amount of transactions that where done before 12 noon. The value zero should be displayed for transactions where the transaction amount has not been entered.

Which query gives the required result?

A)
```
SELECT TO_CHAR(trans_date,'dd-mon-yyyy hh24:mi:ss'), TO_CHAR(trans_amt,'$99999999D99')
FROM transactions
WHERE TO_NUMBER(TO_DATE(trans_date,'hh24')) < 12 AND COALESCE(trans_amt,NULL)<>NULL;
```

B)
```
SELECT TO_CHAR(trans_date,'dd-mon-yyyy hh24:mi:ss'), NVL(TO_CHAR(trans_amt,'$99999999D99'),0)
FROM transactions
WHERE TO_CHAR(trans_date,'hh24') < 12;
```

C)
```
SELECT TO_CHAR(trans_date,'dd-mon-yyyy hh24:mi:ss'), COALESCE(TO_NUMBER(trans_amt,'$99999999.99'),0)
FROM transactions
WHERE TO_DATE(trans_date,'hh24') < 12;
```

D)
```
SELECT TO_DATE (trans_date,'dd-mon-yyyy hh24:mi:ss'), NVL2(trans_amt,TO_NUMBER(trans_amt,'$99999999.99'), 0)
FROM transactions
WHERE TO_DATE(trans_date,'hh24') < 12;
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

8. View the Exhibit and examine the structure of the products table.

Table PRODUCTS		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(6)
PROD_NAME	NOT NULL	VARCHAR2(50)
PROD_DESC	NOT NULL	VARCHAR2(4000)
PROD_CATEGORY	NOT NULL	VARCHAR2(50)
PROD_CATEGORY_ID	NOT NULL	NUMBER
PROD_UNIT_OF_MEASURE		VARCHAR2(20)
SUPPLIER_ID	NOT NULL	NUMBER(6)
PROD_STATUS	NOT NULL	VARCHAR2(20)
PROD_LIST_PRICE	NOT NULL	NUMBER(8,2)
PROD_MIN_PRICE	NOT NULL	NUMBER(8,2)

Using the products table, you issue the following query to generate the names, current list price, and discounted list price for all those products whose list price falls below $10 after a discount of 25% is applied on it.

```
SQL>SELECT prod_name, prod_list_price,
          prod_list_price - (prod_list_price * .25) "DISCOUNTED_PRICE"
    FROM products
    WHERE discounted_price < 10;
```

The query generates an error. What is the reason for the error?

A. The parenthesis should be added to enclose the entire expression.

B. The double quotation marks should be removed from the column alias.

C. The column alias should be replaced with the expression in the where clause.

D. The column alias should be put in uppercase and enclosed within double quotation marks in the where clause.

Answer: C

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your 1Z0-061 Exam with Our Prep Materials Via below:

http://www.certleader.com/1Z0-061-dumps.html