

Free On-Line Oracle 12c 1z0-061 pdf

Oracle 12c 1z0-061 PDF Available Here:

<https://www.certification-questions.com/oracle12c-free-dumps/1z0-061-free-dumps.html>

Enrolling now you will get access to 75 unique Oracle 12c 1z0-061 questions.

Q1. 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 (*).

Option C is correct.

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.

Q2. Evaluate the following query:

<https://www.certification-questions.com/>

```
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.

Option C is incorrect.

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.

Q3. 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.

Option C is correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q4. 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

Option C is correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q5. 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  Ladwigh
Jason   Mallin
Samuel  McCain
Allan   MCEwen
Irene   Mikkilineni
Julia   Nayer
```

Which query the required outut?

- 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

Option B is correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q6. View the Exhibit and examine the structure of the promotions table.

```
+-----+
|                                     |
|          Table PROMOTIONS          |
+-----+-----+-----+-----+
|Name      | Null? | Type      |
+-----+-----+-----+-----+
|PROMO_ID  | NOT NULL | NUMBER(6) |
|
```

<https://www.certification-questions.com/>

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 '
      ENO 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.

Option A is correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q7. Examine the structure of the transactions table:

Name	Null?	Type
TRANS_ID	NOT NOLL	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_data,'dd-mon-yyyy hh24:mi:ss'), TO_CHAR(trans_amt, '\$9999999999')
 FROM transaction
 WHERE TO_NUMBER(TO_DATE(trans_date, 'hh24')) < 12 AND COALESCE(trans_amt,NULL)<>NULL;
- B) SELECT TO_CHAR(trans_data,'dd-mon-yyyy hh24:mi:ss'), NVL(TO_CHAR(trans_amt,'\$9999999999'),0)
 FROM transaction
 WHERE TO_CHAR(trans_date, 'hh24') < 12 ;

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

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

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Option B is correct.

Explanation: For this free product demo we don't provide explanation. To know more contact: info@certification-questions.com

Q8. 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 | NOT NULL     | 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.

Option C is correct.

Explanation: Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q9. You need to create a table with the following column specifications

1. Employee ID (numeric data type) for each employee
2. Employee Name (character data type) that stores the employee name
3. Hire date, which stores the date of joining the organization for each employee
4. Status (character data type), that contains the value 'active' if no data is entered
5. Resume (character large object [CLOB] data type), which contains the resume submitted by the employee. Which is the correct syntax to create this table?

- A) CREATE TABLE EMP_1
(emp_id NUMBER(4),
emp_name VARCHAR2 (25),
start_date DATE,
e_status VARCHAR2 (10) DEFAULT 'ACTIVE ',
resume CLOB (200));
- B) CREATE TABLE L_EMP
(emp_id NUMBER(4),
emp_name VARCHAR2 (25),
start_date DATE,
emp_status VARCHAR2 (10) DEFAULT 'ACTIVE' ,
resume CLOB);
- C) CREATE TABLE EMP_1
{emp_id NUMBER(4),
emp_name VARCHAR2{25},
start_date DATE,
emp_status VARCHAR2{10} DEFAULT "ACTIVE",
resume CLOB);
- D) CREATE TABLE EMP_1
{emp_id NUMBER,
emp_name VARCHAR2(25),
start_date DATE,
emp_status VARCHAR2{10} DEFAULT 'ACTIVE' ,
resume CLOB) ;

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Option D is correct.

Explanation:

CLOB Character data (up to 4 GB)

NUMBER ((p, s)) Number having precision p and scale s (Precision is the total number of decimal digits and scale is the number of digits to the right of the decimal point; precision can range from 1 to 38, and scale can range from -84 to 127)

Q10. Examine the structure proposed for the transactions table:

Name	Null?	Type
TRANS_ID	NOT NULL	NUMBER(6)
CUST_NAME	NOT NULL	VARCHAR2 (20)
CUST_STATUS	NOT NULL	CHAR
TRANS_DATE	NOT NULL	DATE
TRANS_VALIDITY		VARCHAR2
CUST_CREDIT_LIMIT		NUMBER

Which two statements are true regarding the creation and storage of data in the above table structure?

- A) The CUST_STATUS column would give an error.
- B) The TRANS_VALIDITY column would give an error
- C) The CUST_STATUS column would store exactly one character.

- D) The CUST_CREDIT_LIMIT column would not be able to store decimalvalues.
 E) The TRANS_VALIDITY column would have a maximum size of one character.
 F) The TRANS_DATE column would be able to store day, month, century, year, hour, minutes, seconds, and fractions of seconds

Option B, C are correct.

Explanation:

VARCHAR2(size) Variable-length character data (A maximum size must be specified minimum size is 1; maximum size is 4,000)
 CHAR [(size)] Fixed-length character data of length size bytes (Default and minimum size is 1; maximum size is 2,000.)
 NUMBER [(p, s)] Number having precision p and scale s (Precision is the total number of decimal digits and scale is the number of digits to the right of the decimal point; precision can range from 1 to 38, and scale can range from -84 to 127)
 DATE Date and time values to the nearest second between January 1, 4712 BC, and December 31, 9999 AD.

Q11. View the Exhibit and examine the data in the PROMO_NAME and PROMO_END_DATE columns of the promotions table, and the required output format.

PROMO_NAME	PROMO_END_DATE
post promotion #20-343	19-JUN-99
post promotion #20-274	16-JUL-99
TV promotion #12-530	13-APR-99
post promotion #17-157	29-JUN-99
TV promotion #12-481	05-JAN-00
newspaper promotion #19-4	16-AUG-98
everyday low price	01-JAN-99

OUTPUT

PROMO_NAME	LAST_DAY
post promotion #20-343	Saturday, June 19, 1999
post promotion #20-274	Friday, July 16, 1999
TV promotion #12-530	Tuesday, April 13, 1999
post promotion #17-157	Tuesday, June 29, 1999
TV promotion #12-481	Wednesday, January 05, 2000
newspaper promotion #19-4	Sunday, August 16, 1998
everyday low price	Friday, January 01, 1999

Which two queries give the correct result?

A) SELECT promo_name, TO_CHAR(promo_end_date, 'Day') AS last_day, TO_CHAR(promo_end_date, 'Month') AS month, TO_CHAR(promo_end_date, 'DD, YYYY') AS last_day FROM promotions;

B) SELECT promo_name, TO_CHAR(promo_end_date, 'fxDay') AS last_day, TO_CHAR(promo_end_date, 'fxMonth') AS month, TO_CHAR(promo_end_date, 'fxDD, YYYY') AS last_day FROM promotions;

C) SELECT promo_name, TRIM(TO_CHAR(promo_end_date, 'Day ')) AS last_day, TRIM(TO_CHAR(promo_end_date, 'Month ')) AS month, TRIM(TO_CHAR(promo_end_date, 'DD, YYYY ')) AS last_day FROM promotions;

D) SELECT promo_name, TO_CHAR(promo_end_date, 'fmDay') AS last_day, TO_CHAR(promo_end_date, 'fmMonth') AS month, TO_CHAR(promo_end_date, 'fmDD, YYYY') AS last_day FROM promotions;

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Option C, D are correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q12.Evaluate the following SQL statement:

```
SQL> SELECT promo_id, promo_category
FROM promotions
WHERE promo_category = 'Internet' ORDER BY 2 DESC UNION
SELECT promo_id, promo_category FROM promotions
WHERE promo_category = 'TV '
UNION
SELECT promo_id, promo_category FROM promotions
WHERE promo_cate9ory = 'Radio';
```

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

- A)It executes successfully and displays rows inthe descending order of PROMO_CATEGORY
- B)It produces an error because positionalnotation cannot be used inthe order by clause with set operators.
- C)It executes successfully but ignores the order by clause because it is not located at the end of the compound statement
- D)It produces an error because the order by clause should appear only at the end of a compound query- that is,with the last select statement.

Option D is correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q13.Youwant to display the name,joining date, and manager for all the employees Newly hired employees are yet to be assigned a department or a manager For them,'No Manager1 should be displayed inthe manager column.

Which SQL query gets the required output?

- A) SELECT e .last_name, e.hire_date, NVL{m .last name, 'No Manager ') Manager FROM employees e JOIN employees m ON (e.manager_id = m .employee_id);
- B) SELECT e .last_name, e.hire_date, NVL {m .last_name, 'No Manager ') Manager FROM employees e LEFT OUTER JOIN employees m ON {e.manager_id = m .employee_id);
- C)SELECT e.last_name, e.hire_date, NVL (m.last_name, 'No Manager ') Manager FROM employees e RIGHT OUTER JOIN employees m ON {e.manager_id = m .employee_id);
- D)SELECT e.last_name, e.hire_date, NVL(m .last_name, 'No Manager ') Manager FROM employees e NATURAL JOIN employees m ON {e.manager_id = m .employee_id);

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Option D is correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com

Reference: <http://ivrainbow65.blogspot.com/>

<https://www.certification-questions.com/>

Q14. You want to display 5 percent of the rows from the sales table for products with the lowest AMOUNT_SOLD and also want to include the rows that have the same AMOUNT_SOLD even if this causes the output to exceed 5 percent of the rows.

Which query will provide the required result?

A) SELECT prod_id, cust_id, amount_sold FROM sales
ORDER BY amount_sold
FETCH FIRST 5 PERCENT ROWS ONLY;

B) SELECT prod_id, cust_id, amount_sold FROM sales
ORDER BY amount_sold
FETCH FIRST 5 PERCENT ROWS WITH TIES ONLY;

C) SELECT prod_id, cust_id, amount_sold FROM sales
ORDER BY amount_sold
FETCH FIRST 5 PERCENT ROWS ONLY WITH TIES;

D) SELECT prod_id, cust_id, amount_sold FROM sales
ORDER BY amount_sold
FETCH FIRST 5 PERCENT ROWS WITH TIES;

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Option D is correct.

Explanation: Explanation not provided for the free version. To know more contact: www.certification-questions.com

Q15. Examine the structure of the employees table.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
EMAIL	NOT NULL	VARCHAR2 (25)
PHONE_NUMBER		VARCHAR2 (20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2 (10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

You want to display the maximum and minimum salaries of employees hired 1 year ago. Which two statements would get the correct output?

A) SELECT MIN(salary), MAX(salary) FROM (SELECT salary
FROM employees
WHERE hire date < SYSDATE-365) ,

B) SELECT minsal, maxsal
FROM (SELECT MIN(salary) minsal, MAX(salary) maxsal FROM employees
WHERE hire date < SYSDATE-365
GROUP BY MIN(salary), MAX (salary)) ;

C) SELECT minsal, maxsal
FROM (SELECT MIN (salary) minsal, MAX(salary) maxsal FROM employees
WHERE hire date < SYSDATE-365)
GROUP BY maxsal, minsal ;

```
D) SELECT MIN(salary) minsal, MAX(salary) maxsal FROM employees
WHERE hire date < SYSDATE-365
GROUP BY MIN(salary), MAX(salary);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Option C,D are correct.

Explanation:Explanation not provided for the free version. To know more contact: www.certification-questions.com