

[2016-June-NEWFree Braindump2go 1Z0-051 VCE Dumps 303Q&As Download[NQ1-NQ10

2016 June Oracle Official: 1Z0-051: Oracle Database 11g: SQL Fundamentals I Exam Questions New Updated Today!
 Braindump2go.com [Offers 1Z0-051 PDF and VCE Dumps 303q for Free Downloading!](#)NEW QUESTION 11 - NEW QUESTION 10: QUESTION 11Examine the structure of the PROGRAMS table:Which two SQL statements would execute successfully?

(Choose two.)

Name	Null?	Type
PROG_ID	NOT NULL	NUMBER(3)
PROG_COST		NUMBER(6,2)
START_DATE	NOT NULL	DATE
END_DATE		DATE

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; Answer: ADExplanation:NVL FunctionConverts a null value to an actual value:Data types that can be used are date, character, and number.Data types must match: - NVL(commission_pct,0)- NVL(hire_date,'01-JAN-97')- NVL(job_id,'No Job Yet')MONTHS_BETWEEN(date1, date2): Finds the number of months between date1 and date2 The result can be positive or negative. If date1 is later than date2, the result is positive; if date1 is earlier than date2, the result is negative. The noninteger part of the result represents a portion of the month. MONTHS_BETWEEN returns a numeric value. - answer C NVL has different datatypes - numeric and strings, which is not possible!The data types of the original and if null parameters must always be compatible. They must either be of the same type, or it must be possible to implicitly convert if null to the type of the original parameter. The NVL function returns a value with the same data type as the original parameter. QUESTION 12The PRODUCTS table has the following structure:

Name	Null?	Type
PROD_ID	NOT NULL	NUM
PROD_NAME		VAR
PROD_EXPIRY_DATE		DA

Evaluate the following two SQL statements: `SQL>SELECT prod_id,NVL2(prod_expiry_date,prod_expiry_date + 15,SYSDATE) FROM products;` `SQL>SELECT prod_id,NVL2(prod_expiry_date,SYSDATE,prod_expiry_date + 15) FROM products;` Which statement is true regarding the outcome? A. Both the statements execute and give different results.B. Both the statements execute and give the same result. C. Only the first SQL statement executes successfully.D. Only the second SQL statement executes successfully. Answer: A Explanation:Using the NVL2 FunctionThe NVL2 function examines the first expression. If the first expression is not null, the NVL2 function returns the second expression. If the first expression is null, the third expression is returned.SyntaxNVL2(expr1, expr2, expr3)In the syntax:expr1 is the source value or expression that may contain a null expr2 is the value that is returned if expr1 is not null expr3 is the value that is returned if expr1 is null QUESTION 13Examine the structure of the INVOICE table:Which two SQL statements would execute successfully? (Choose two.)

Name	Null?	Type
INV_NO	NOT NULL	NUMBER
INV_DATE		DATE
INV_AMT		NUMBER(10,2)

A. SELECT inv_no,NVL2(inv_date,'Pending','Incomplete')FROM invoice;B. SELECT inv_no,NVL2(inv_amt,inv_date,'Not Available')FROM invoice;C. SELECT inv_no,NVL2(inv_date,sysdate-inv_date,sysdate)FROM invoice;D. SELECT inv_no,NVL2(inv_amt,inv_amt*.25,'Not Available')FROM invoice; Answer: ACEExplanation:The NVL2 FunctionThe NVL2 function provides an enhancement to NVL but serves a very similar purpose. It evaluates whether a column or expression of any data type is null or not.5-6 The NVL functionIf the first term is not null, the second parameter is returned, else the third parameter is returned. Recall that the NVL function is different since it returns the original term if it is not null. The NVL2 function takes three mandatory parameters. Its syntax is NVL2(original, ifnotnull, ifnull), where original represents the term being tested. Ifnotnull is returned if original is not null, and ifnull is returned if original is null. The data types of the ifnotnull and ifnull parameters must be compatible, and they cannot be of type LONG.They must either be of the same type, or it must be possible to convert ifnull to the type of the ifnotnull parameter. The data type returned by the NVL2 function is the same as that of the ifnotnull parameter.

QUESTION 14 View the Exhibit and evaluate the structure and data in the CUST_STATUS table. You issue the following SQL statement: Which statement is true regarding the execution of the above query?

```
SQL> SELECT custno, NVL2(NULLIF(amt_spent, credit_limit), 0, 1000)*'BONUS'  
FROM cust_status;
```

A. It produces an error because the AMT_SPENT column contains a null value.

B. It displays a bonus of 1000 for all customers whose AMT_SPENT is less than CREDIT_LIMIT.
C. It displays a bonus of 1000 for all customers whose AMT_SPENT equals CREDIT_LIMIT, or AMT_SPENT is null.
D. It produces an error because the TO_NUMBER function must be used to convert the result of the NULLIF function before it can be used by the NVL2 function.

Answer: C
Explanation: The NULLIF Function
The NULLIF function tests two terms for equality. If they are equal the function returns a null, else it returns the first of the two terms tested. The NULLIF function takes two mandatory parameters of any data type. The syntax is NULLIF(ifunequal, comparison_term), where the parameters ifunequal and comparison_term are compared. If they are identical, then NULL is returned. If they differ, the ifunequal parameter is returned.

QUESTION 15 Which statement is true regarding the COALESCE function?
A. It can have a maximum of five expressions in a list.
B. It returns the highest NOT NULL value in the list for all rows.
C. It requires that all expressions in the list must be of the same data type.
D. It requires that at least one of the expressions in the list must have a NOT NULL value.

Answer: C
Explanation: The COALESCE Function
The COALESCE function returns the first nonnull value from its parameter list. If all its parameters are null, then null is returned. The COALESCE function takes two mandatory parameters and any number of optional parameters. The syntax is COALESCE(expr1, expr2, ..., exprn), where expr1 is returned if it is not null, else expr2 if it is not null, and so on. COALESCE is a general form of the NVL function, as the following two equations illustrate:

COALESCE(expr1, expr2) = NVL(expr1, expr2)
COALESCE(expr1, expr2, expr3) = NVL(expr1, NVL(expr2, expr3))

The data type COALESCE returns if a not null value is found is the same as that of the first not null parameter. To avoid an "ORA-00932: inconsistent data types" error, all not null parameters must have data types compatible with the first not null parameter.

QUESTION 16 View the Exhibit and examine the structure of the PROMOTIONS table. Using the PROMOTIONS table, you need to find out the average cost for all promos in the ranges \$0-2000 and \$2000-5000 in category A. You issue the following SQL statement:

```
SQL>SELECT AVG(CASE
    WHEN promo_cost BETWEEN 0 AND 2000 AND promo_category='A'
    THEN promo_cost
    ELSE null END) *CAT_2000A'
    WHEN promo_cost BETWEEN 2001 AND 5000 AND promo_category='A'
    THEN promo_cost
    ELSE null END) *CAT_5000A'
FROM promotions;
```

What would be the outcome?

Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(8)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
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

A. It executes successfully and gives the required result. B. It generates an error because NULL cannot be specified as a return value. C. It generates an error because CASE cannot be used with group functions. D. It generates an error because multiple conditions cannot be specified for the WHEN clause. Answer: A
 Explanation: CASE Expression Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement: CASE expr WHEN comparison_expr1 THEN return_expr1 [WHEN comparison_expr2 THEN return_expr2 WHEN comparison_exprn THEN return_exprn ELSE else_expr] END QUESTION 17 View the Exhibit and examine the structure of the PROMOTIONS table. Which SQL statements are valid? (Choose all that apply.)

Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(8)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
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

A. SELECT promo_id, DECODE(NVL(promo_cost,0), promo_cost, promo_cost * 0.25, 100) "Discount" FROM promotions; B. SELECT promo_id, DECODE(promo_cost, 10000, DECODE(promo_category, 'G1', promo_cost *.25, NULL), NULL) "Catcost" FROM promotions; C. SELECT promo_id, DECODE(NULLIF(promo_cost, 10000), NULL, promo_cost*.25, 'N/A') "Catcost" FROM promotions; D. SELECT promo_id, DECODE(promo_cost, >10000, 'High', <10000, 'Low') "Range" FROM promotions; Answer: AB
 Explanation: The DECODE Function Although its name sounds mysterious, this function is straightforward. The DECODE function implements if-then-else conditional logic by testing its first two terms for equality and returns the third if they are equal and optionally returns another term if they are not. The DECODE function takes at least three mandatory parameters, but can take many more. The syntax of the function is DECODE(expr1, comp1, iftrue1, [comp2, iftrue2... [compN, iftrueN]], [iffalse]).

QUESTION 18 Examine the data in the PROMO_BEGIN_DATE column of the PROMOTIONS table:
 PROMO_BEGIN_DATE
 -----04-jan-0010-jan-0015-dec-9918-oct-9822-aug-99
 You want to display the number of promotions started in 1999 and 2000. Which query gives the correct output?
 A. SELECT SUM(DECODE(SUBSTR(promo_begin_date,8),'00',1,0)) "2000", SUM(DECODE(SUBSTR(promo_begin_date,8),'99',1,0)) "1999" FROM promotions; B. SELECT SUM(CASE TO_CHAR(promo_begin_date,'yyy') WHEN '99' THEN 1 ELSE 0 END) "1999", SUM(CASE TO_CHAR(promo_begin_date,'yyy') WHEN '00' THEN 1 ELSE 0 END) "2000" FROM promotions; C. SELECT COUNT(CASE TO_CHAR(promo_begin_date,'yyy') WHEN '99' THEN 1 ELSE 0 END) "1999", COUNT(CASE TO_CHAR(promo_begin_date,'yyy') WHEN '00' THEN 1 ELSE 0 END) "2000" FROM promotions; D. SELECT COUNT(DECODE(SUBSTR(TO_CHAR(promo_begin_date,'yyy'), 8), '1999', 1, 0)) "1999", COUNT(DECODE(SUBSTR(TO_CHAR(promo_begin_date,'yyy'), 8), '2000', 1, 0)) "2000" FROM promotions; Answer: A

QUESTION 19 Examine the structure of the TRANSACTIONS table:
 name Null Type

 TRANS_ID NOT NULL NUMBER(3) CUST_NAME VARCHAR2(30) TRANS_DATE
 TIMESTAMPTRANS_AMT NUMBER(10,2)
 You want to display the date, time, and transaction amount of transactions that were 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 transactionsWHERE 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 transactionsWHERE 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 transactionsWHERE TO_DATE(trans_date,'hh24') < 12;

Answer: B QUESTION 20Examine the structure of the TRANSACTIONS table:

name	Null	Type
-----TRANS_ID	NOT NULL	NUMBER(3)
CUST_NAME		VARCHAR2(30)
TRANS_DATE		DATE
TRANS_AMT		NUMBER(10,2)

You want to display the transaction date and specify whether it is a
weekday or weekend.Evaluate the following two queries:Which statement is true regarding the above queries?

```
SQL>SELECT TRANS_DATE,CASE  
WHEN TRIM(TO_CHAR(trans_date,'DAY')) IN ('SATUR  
ELSE 'weekday'  
END "Day Type"  
FROM transactions  
SQL>SELECT TRANS_DATE,CASE  
WHEN TO_CHAR(trans_date,'DAY') BETWEEN 'MOND  
ELSE 'weekend'  
END "Day Type"FROM transactions;
```

A. Both give wrong results.B. Both give the correct result.C. Only the first query gives the correct result.D. Only the second query gives the correct result. Answer: CExplanation:Range Conditions Using the BETWEEN OperatorUse the BETWEEN operator to display rows based on a range of values:SELECT last_name, salaryFROM employeesWHERE salary BETWEEN 2500 AND 3500;Range Conditions Using the BETWEEN OperatorYou can display rows based on a range of values using the BETWEEN operator. The range that you specify contains a lower limit and an upper limit. The SELECT statement in the slide returns rows from the EMPLOYEES table for any employee whose salary is between \$2,500 and \$3,500. Values that are specified with the BETWEEN operator are inclusive. However, you must specify the lower limit first.You can also use the BETWEEN operator on character values:SELECT last_nameFROM employeesWHERE last_name BETWEEN 'King' AND 'Smith'; 2016 Valid Oracle 1Z0-051 Exam Study Materials: 1.| Latest 1Z0-051 PDF and VCE Dumps 303Q&As from Braindump2go: <http://www.braindump2go.com/1z0-051.html> [100% Exam Pass Guaranteed!] 2.| NEW 1Z0-051 Exam Questions and Answers: <https://drive.google.com/folderview?id=0B75b5xYLjSSNVGxLT202clFMbjA&usp=sharing> MORE Practice is the Most Important IF You want to PASS 1Z0-051 Exam 100%! ----- Braindump2go.com----- Pass All IT Exams at the first Try!