To display the names of all promos done after January

1, 2001 starting with the latest promo.

Which query would give the required result? (Choose all that apply.)

Select one or more:

[✓ a.] SELECT promo_name,promo_begin_date "START DATE" ✓
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY "START DATE" DESC;

b. SELECT promo_name,promo_begin_date
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY promo_name DESC;

c. SELECT promo_name,promo_begin_date
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY 1 DESC;

[✓ d.] SELECT promo_name,promo_begin_date ✓
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY 2 DESC;

The correct answers are: SELECT promo_name,promo_begin_date "START DATE"
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY "START DATE" DESC;, SELECT promo_name,promo_begin_date
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY 2 DESC;

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?

Select one:

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;

The correct answer is: SELECT 'Dear Customer ' || customer_name || ',' FROM customers;

Which statement is true regarding the default behavior of the ORDER BY clause?

Select one:

- a. NULL values are not considered at all by the sort operation
- b. Only those columns that are specified in the SELECT list can be used in the ORDER BY clause
- c. Numeric values are displayed from the maximum to the minimum value if they have decimal positions
- [✓ d. In a character sort, the values are case-sensitive] ✓

The correct answer is: In a character sort, the values are case-sensitive

To update the CUST_CREDIT_LIMIT column to NULL for all the customers, where CUST_INCOME_LEVEL has NULL in the CUSTOMERS table. Which SQL statement will accomplish the task?

Select one:

[✓ a.] ✓
```
UPDATE customers
SET cust_credit_limit = NULL
WHERE cust_income_level IS NULL;
```

b.
```
UPDATE customers
SET cust_credit_limit = TO_NUMBER(' ',9999)
WHERE cust_income_level IS NULL;
```

c.
```
UPDATE customers
SET cust_credit_limit = TO_NUMBER(NULL)
WHERE cust_income_level = TO_NUMBER(NULL);
```

d.
```
UPDATE customers
SET cust_credit_limit = NULL
WHERE cust_income_level = NULL;
```

The correct answer is: UPDATE customers
SET cust_credit_limit = NULL
WHERE cust_income_level IS NULL;

Generate a list of all customer last names with their credit limits from the CUSTOMERS table. Customers who do not have a credit limit should appear last in the list. kindly note that customers who do not have credit card will have NULL against credit limit.

Which query would achieve the required result?

Select one:

- a. SELECT cust_last_name,cust_credit_limit
  FROM customers
  ORDER BY cust_last_name,cust_credit_limit NULLS LAST;
- b. SELECT cust_last_name,cust_credit_limit
  FROM customers;
- c. SELECT cust_last_name,cust_credit_limit
  FROM customers
  ORDER BY cust_credit_limit DESC;
- [✓ d.] SELECT cust_last_name,cust_credit_limit ✓
  FROM customers
  ORDER BY cust_credit_limit;

The correct answer is: SELECT cust_last_name,cust_credit_limit
FROM customers
ORDER BY cust_credit_limit;

To generate a report that shows an increase in the credit limit by 15% for all customers. Customers whose credit limit has not been entered should have the message "Not Available" displayed.

Which SQL statement would produce the required result?

Select one:

- a. SELECT NVL(cust_credit_limit,'Not Available')*.15 "NEW CREDIT"
  FROM customers;
- b. SELECT NVL(cust_credit_limit*.15,'Not Available')"NEW CREDIT"
  FROM customers;
- [✓ c.] SELECT NVL(TO_CHAR(cust_credit_limit*.15),'Not Available') "NEW CREDIT" ✓
  FROM customers;
- d. SELECT TO_CHAR(NVL(cust_credit_limit*.15,'Not Available')) "NEW CREDIT"
  FROM customers;

The correct answer is: SELECT NVL(TO_CHAR(cust_credit_limit*.15),'Not Available') "NEW CREDIT"
FROM customers;

To display the names of employees that are not assigned to a department.

Evaluate this SQL statement:

SELECT last_name, first_name

FROM employee

WHERE dept_id = NULL;

Which change should you make to achieve the desired result?

Select one:

- a. Create an outer join.
- [✓ b. Change the operator in the WHERE condition.] ✓
- c. Change the column in the WHERE condition.
- d. Add a second condition to the WHERE condition.

The correct answer is: Change the operator in the WHERE condition.

Select the suitable option for fetching the output of the following query.

select substr("Oracle World",1,6) from dual;

Select one:

- [✓ a. Oracle] ✓
- b. racle W
- c. racle
- d. racle Wo

The correct answer is: Oracle

ABC company wants to give each employee a $100 salary increment. You need to evaluate the

results from the EMP table prior to the actual modification. If you do not want to store the results in the database, which statement is valid?

Select one:

- a. You need to give the arithmetic expression that involves the salary increment in the UPDATE clause of the SELECT statement.
- b. You need to give the arithmetic expression that involves the salary increment in the SET clause of the UPDATE statement.
- [✓ c. You need to give the arithmetic expression that involves the salary increment in the DISPLAY clause of the SELECT statement.] ✓
- d. You need to add a column to the EMP table.

The correct answer is: You need to give the arithmetic expression that involves the salary increment in the DISPLAY clause of the SELECT statement.

To calculate the number of days from 1st Jan 2007 till date:

Dates are stored in the default format of dd-mm-rr.

Which SQL statements would give the required output?

Select one or more:

- a. SELECT SYSDATE - '01-JAN-2007' FROM DUAL ;
- [✓ b. SELECT SYSDATE - TO_DATE('01/JANUARY/2007') FROM DUAL;] ✓
- c. SELECT TO_DATE(SYSDATE,'DD/MONTH/YYYY')-'01/JANUARY/2007' FROM DUAL;
- [✓ d. SELECT SYSDATE - TO_DATE('01-JANUARY-2007) FROM DUAL;] ✓
- e. SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY')-'01-JAN-2007' FROM DUAL;

The correct answers are: SELECT SYSDATE - TO_DATE('01/JANUARY/2007') FROM DUAL;, SELECT SYSDATE - TO_DATE('01-JANUARY-2007) FROM DUAL;