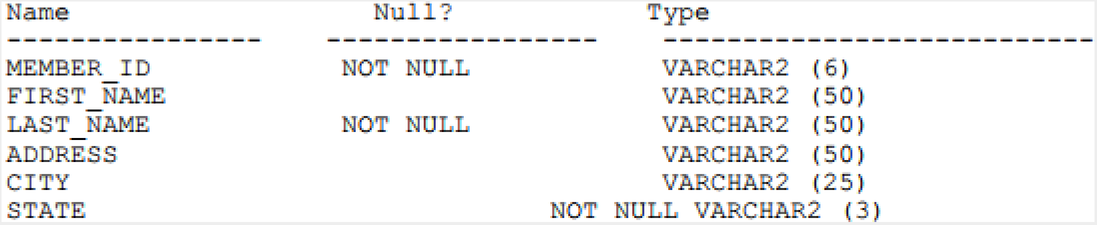
Question Results

Score 1 of 1

Question:

Examine the structure of the MEMBERS table.



```
Name                    Null?              Type
----------------    -----------------    ---------------------------
MEMBER_ID             NOT NULL             VARCHAR2 (6)
FIRST_NAME                                 VARCHAR2 (50)
LAST_NAME             NOT NULL             VARCHAR2 (50)
ADDRESS                                    VARCHAR2 (50)
CITY                                       VARCHAR2 (25)
STATE                                NOT NULL VARCHAR2 (3)
```

Which query can be used to display the last names and city names only for members from the states MO and MI?

Response:

SELECT last_name, city FROM members WHERE state LIKE 'M%';

✓ SELECT last_name, city FROM members WHERE state IN ('MO', 'MI');

SELECT last_name, city FROM members WHERE state ='MO' AND state ='MI';

SELECT DISTINCT last_name, city FROM members WHERE state ='MO' OR state ='MI';

Score 1 of 1

Question:

View the exhibit and examine the structures of the EMPLOYEES and DEPARTMENTS tables.

```
EMPLOYEES
Name Null? Type
----------------- ----- -------------
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME VARCHAR2(20)
LAST_NAME NOT NULL VARCHAR2(25)
HIRE_DATE NOT NULL DATE
JOB_ID NOT NULL VARCHAR2(10)
SALARY NUMBER(10,2)
COMMISSION NUMBER(6,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)
DEPARTMENTS

Name Null? Type
----------------- ----- -------------
DEPARTMENT_ID NOT NULL NUMBER(4)
DEPARTMENT_NAME NOT NULL VARCHAR2(30)
MANAGER_ID NUMBER(6)
LOCATION_ID NUMBER(4)
```

You want to update EMPLOYEES table as follows:

- Update only those employees who work in Boston or Seattle (locations 2900 and 2700).
- Set department_id for these employees to the department_id corresponding to London (location_id 2100).
- Set the employees' salary in location_id 2100 to 1.1 times the average salary of their department.
- Set the employees' commission in location_id 2100 to 1.5 times the average commission of their department.

You issue the following command:

```
SQL> UPDATE employees
SET department_id =
(SELECT department_id
FROM departments
WHERE location_id = 2100),
(salary, commission) =
(SELECT 1.1*AVG(salary), 1.5*AVG(commission)
FROM employees, departments
WHERE departments.location_id IN(2900, 2700, 2100))
WHERE department_id IN
(SELECT department_id
FROM departments
WHERE location_id = 2900
OR location_id = 2700;
```

What is outcome?

Response:

It generates an error because a subquery cannot have a join condition in a UPDATE statement.

✓ It executes successfully but does not give the correct result.

It generates an error because multiple columns (SALARY, COMMISSION) cannot be specified together in an UPDATE statement.

It executes successfully and gives the correct result.

Score 1 of 1

Question:

You are tasked with querying the data dictionary view that lists only those sequences to which you currently have privileges but don’t necessarily own. To do this, you log in to your own user account and query the data dictionary view called:

Response:

✓ ALL_SEQUENCES

USER_SEQUENCES

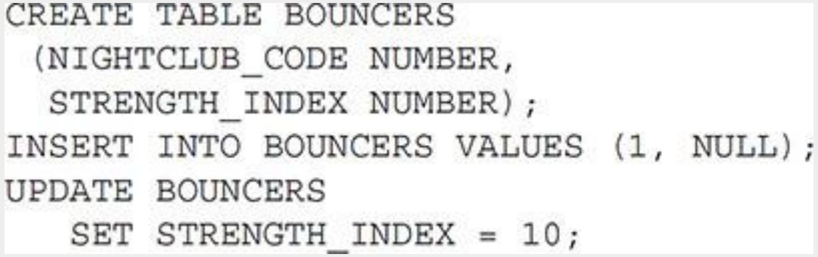
DBA_SEQUENCES

USER_PRIV_SEQUENCES

Score 1 of 1

Question:

Review the following SQL statements:



```
CREATE TABLE BOUNCERS
 (NIGHTCLUB_CODE NUMBER,
  STRENGTH_INDEX NUMBER);
INSERT INTO BOUNCERS VALUES (1, NULL);
UPDATE BOUNCERS
   SET STRENGTH_INDEX = 10;
```

What is the end result of the SQL statements listed here?

Response:

None of the above.

✓ The BOUNCERS table will contain one row.

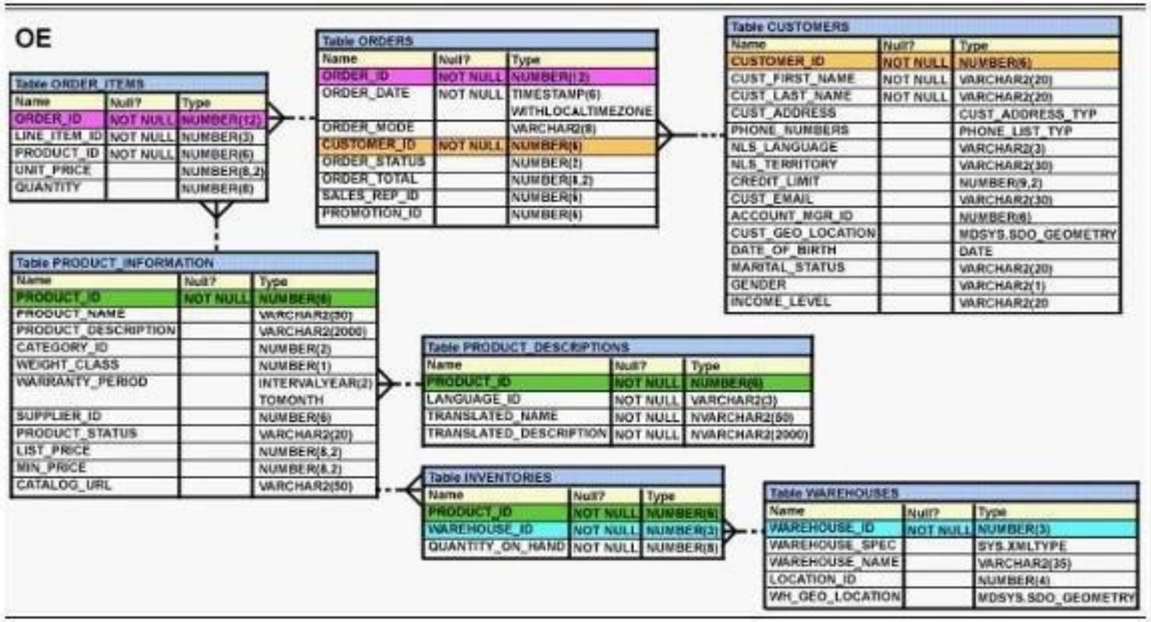
The UPDATE will fail because there is no WHERE clause.

The BOUNCERS table will contain two rows.

Score 1 of 1

Question:

View the Exhibit and examine the description of the PRODUCT_INFORMATION table. Which SQL statement would retrieve from the table the number of products having LIST_PRICE as NULL?



OE

Table ORDER_ITEMS

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER(12)
LINE_ITEM_ID	NOT NULL	NUMBER(3)
PRODUCT_ID	NOT NULL	NUMBER(6)
UNIT_PRICE		NUMBER(8,2)
QUANTITY		NUMBER(8)

Table ORDERS

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER(12)
ORDER_DATE	NOT NULL	TIMESTAMP(6) WITHLOCALTIMEZONE
ORDER_MODE		VARCHAR2(8)
CUSTOMER_ID	NOT NULL	NUMBER(6)
ORDER_STATUS		NUMBER(2)
ORDER_TOTAL		NUMBER(8,2)
SALES_REP_ID		NUMBER(6)
PROMOTION_ID		NUMBER(6)

Table CUSTOMERS

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(6)
CUST_FIRST_NAME	NOT NULL	VARCHAR2(20)
CUST_LAST_NAME	NOT NULL	VARCHAR2(20)
CUST_ADDRESS		CUST_ADDRESS_TYP
PHONE_NUMBERS		PHONE_LIST_TYP
NLS_LANGUAGE		VARCHAR2(3)
NLS_TERRITORY		VARCHAR2(30)
CREDIT_LIMIT		NUMBER(9,2)
CUST_EMAIL		VARCHAR2(30)
ACCOUNT_MGR_ID		NUMBER(6)
CUST_GEO_LOCATION		MDSYS.SDO_GEOMETRY
DATE_OF_BIRTH		DATE
MARITAL_STATUS		VARCHAR2(20)
GENDER		VARCHAR2(1)
INCOME_LEVEL		VARCHAR2(20

Table PRODUCT_INFORMATION

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(6)
PRODUCT_NAME		VARCHAR2(50)
PRODUCT_DESCRIPTION		VARCHAR2(2000)
CATEGORY_ID		NUMBER(2)
WEIGHT_CLASS		NUMBER(1)
WARRANTY_PERIOD		INTERVALYEAR(2) TOMONTH
SUPPLIER_ID		NUMBER(6)
PRODUCT_STATUS		VARCHAR2(20)
LIST_PRICE		NUMBER(8,2)
MIN_PRICE		NUMBER(8,2)
CATALOG_URL		VARCHAR2(50)

Table PRODUCT_DESCRIPTIONS

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(6)
LANGUAGE_ID	NOT NULL	VARCHAR2(3)
TRANSLATED_NAME	NOT NULL	NVARCHAR2(50)
TRANSLATED_DESCRIPTION	NOT NULL	NVARCHAR2(2000)

Table INVENTORIES

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(6)
WAREHOUSE_ID	NOT NULL	NUMBER(3)
QUANTITY_ON_HAND	NOT NULL	NUMBER(8)

Table WAREHOUSES

Name	Null?	Type
WAREHOUSE_ID	NOT NULL	NUMBER(3)
WAREHOUSE_SPEC		SYS.XMLTYPE
WAREHOUSE_NAME		VARCHAR2(35)
LOCATION_ID		NUMBER(4)
WH_GEO_LOCATION		MDSYS.SDO_GEOMETRY

Response:

```
SELECT COUNT (DISTINCT list_price)
FROM product_information
WHERE list_price is NULL
```

```
SELECT COUNT (list_price)
FROM product_information
WHERE list_price i= NULL
```

```
SELECT COUNT (list_price)
FROM product_information
WHERE list_price is NULL
```

✓

```
SELECT COUNT (NVL(list_price, 0))
FROM product_information
WHERE list_price is NULL
```

Score 0 of 1

Question:

Which of the following can be used to remove data from a table?
(Choose two.)

Response:

✓ DELETE

✗ ALTER

✓ UPDATE

MODIFY

Score 1 of 1

Question:

Examine the structure of the INVOICE table.

```
Name                 Null?            Type
------------------ ------------------ -------------
INV_NO             NOT NULL     NUMBER(3)
INV_DATE                              DATE
INV_AMT                               NUMBER(10,2)
```

Which two SQL statements would execute successfully?

Response:

```
SELECT inv_no, NVL2(inv_amt, inv_amt*.25, 'Not Available')
FROM invoice;
```

✓
```
SELECT inv_no, NVL2(inv_date, 'Pending', 'Incomplete')
FROM invoice;
```

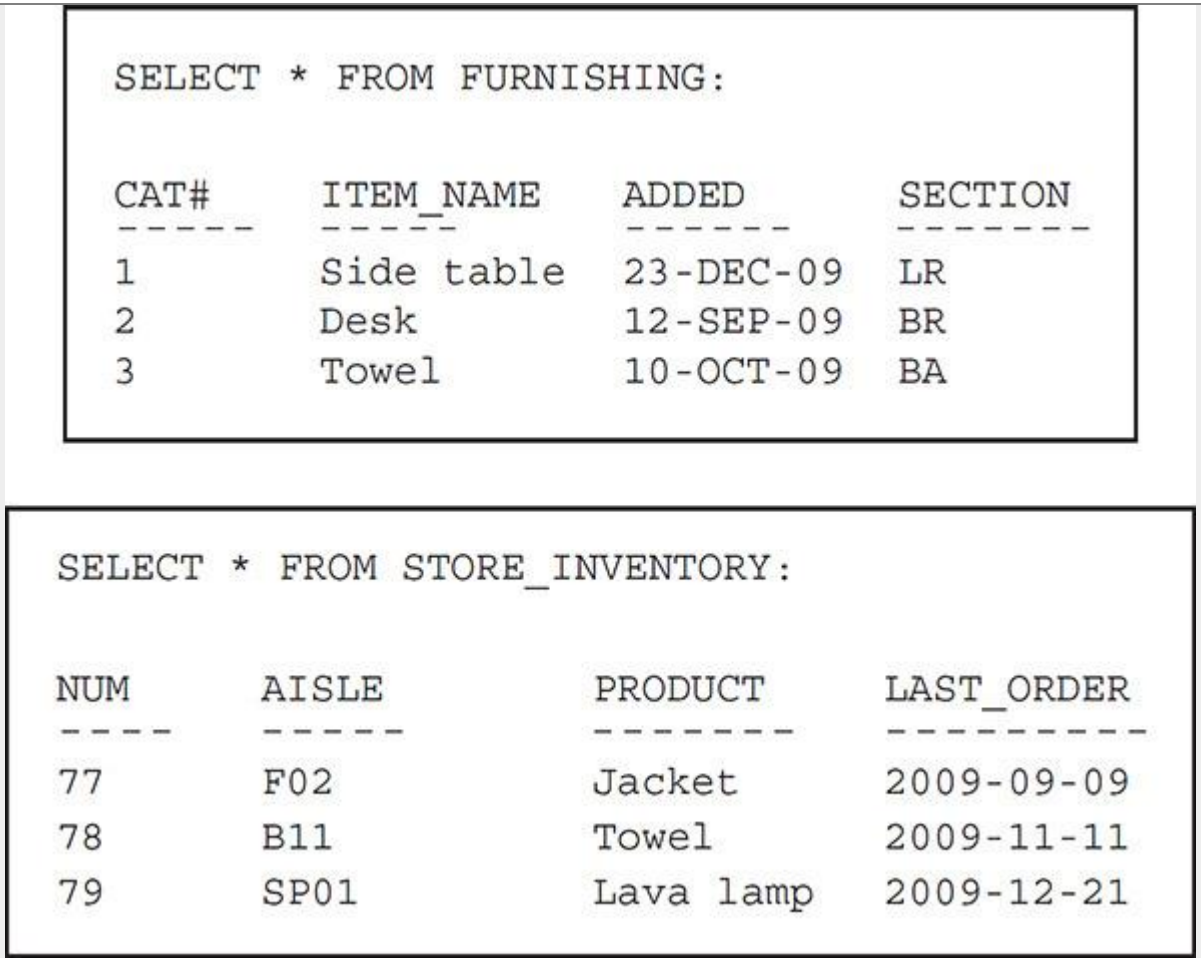
✓
```
SELECT inv_no, NVL2(inv_date, sysdate-inv_date, sysdate)
FROM invoice;
```

```
SELECT inv_no, NVL2(inv_amt, inv_date, 'Not Available')
FROM invoice;
```

Score 1 of 1

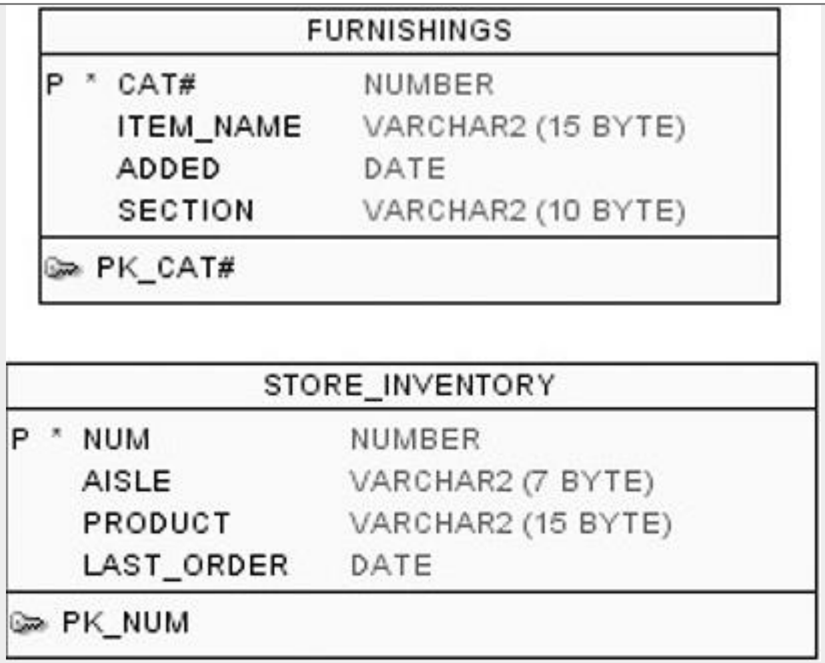
Question:

Review the first two illustrations and then review this SQL code:



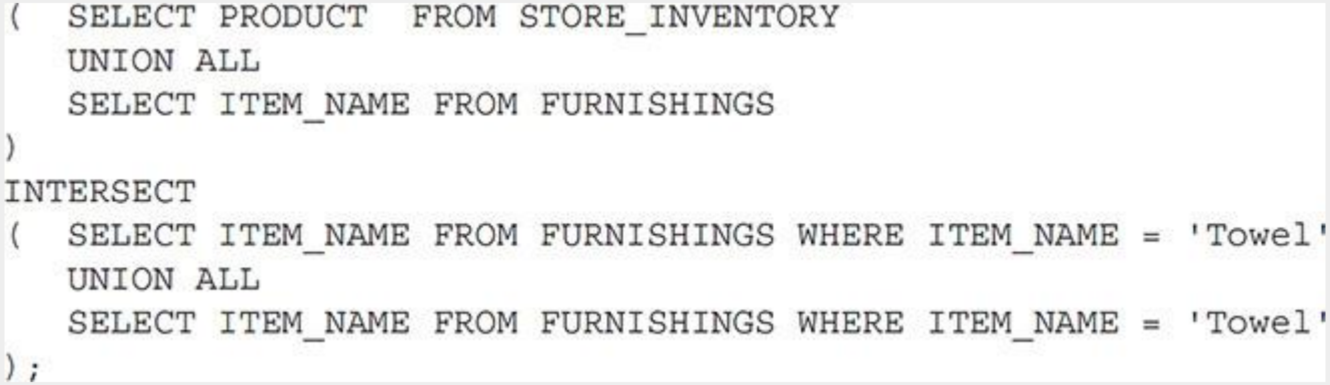
```
SELECT * FROM FURNISHING:

CAT#     ITEM_NAME    ADDED       SECTION
-----    -----        -------     --------
1        Side table   23-DEC-09   LR
2        Desk         12-SEP-09   BR
3        Towel        10-OCT-09   BA
```

```
SELECT * FROM STORE_INVENTORY:

NUM      AISLE          PRODUCT      LAST_ORDER
----     -----          -------      ----------
77       F02            Jacket       2009-09-09
78       B11            Towel        2009-11-11
79       SP01           Lava lamp    2009-12-21
```



FURNISHINGS			
P	*	CAT#	NUMBER
		ITEM_NAME	VARCHAR2 (15 BYTE)
		ADDED	DATE
		SECTION	VARCHAR2 (10 BYTE)
		PK_CAT#	

STORE_INVENTORY			
P	*	NUM	NUMBER
		AISLE	VARCHAR2 (7 BYTE)
		PRODUCT	VARCHAR2 (15 BYTE)
		LAST_ORDER	DATE
		PK_NUM	



```
(   SELECT PRODUCT  FROM STORE_INVENTORY
    UNION ALL
    SELECT ITEM_NAME FROM FURNISHINGS
)
INTERSECT
(   SELECT ITEM_NAME FROM FURNISHINGS WHERE ITEM_NAME = 'Towel'
    UNION ALL
    SELECT ITEM_NAME FROM FURNISHINGS WHERE ITEM_NAME = 'Towel'
);
```

How many rows will result from this code?

Response:

4

6

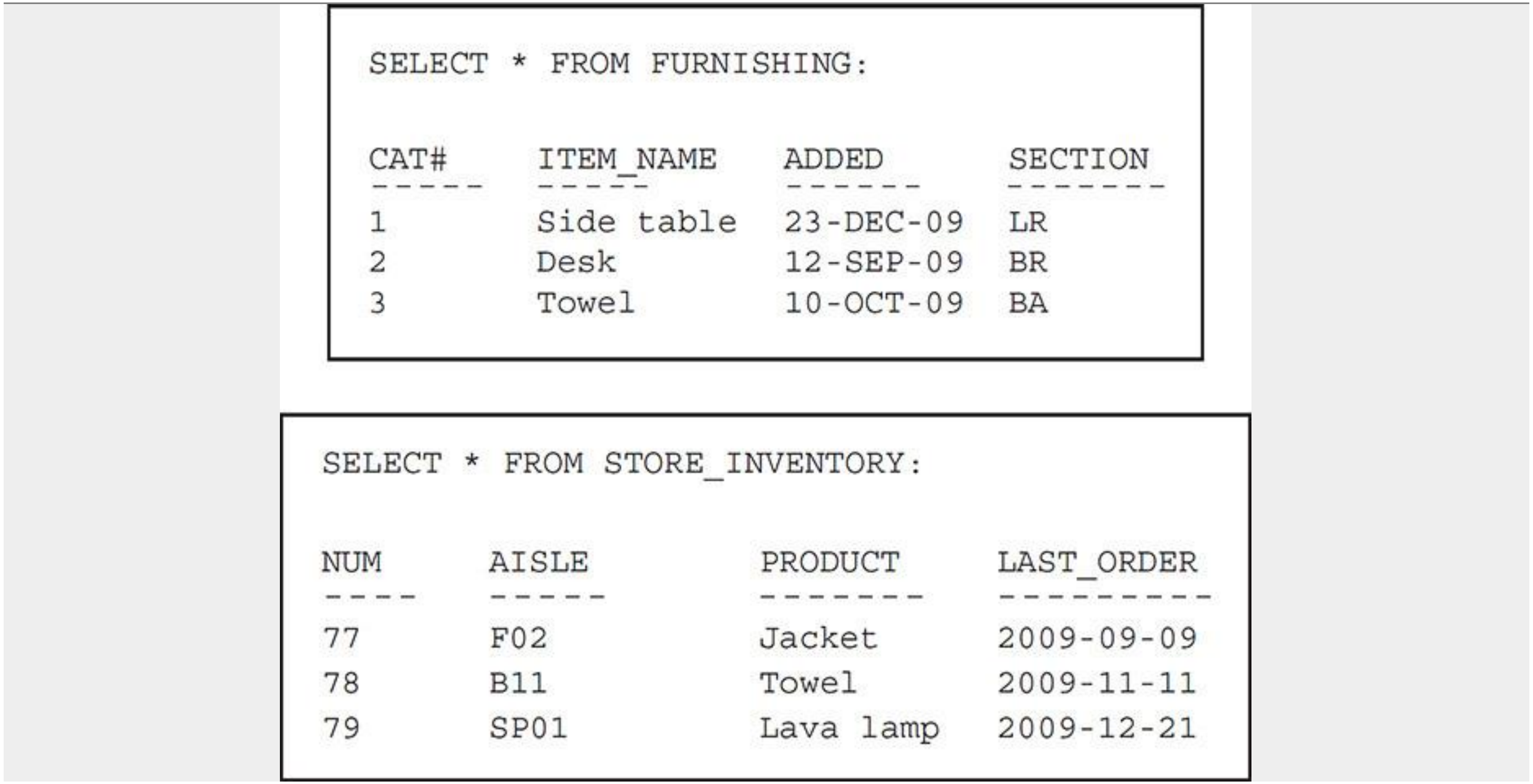
2

✓ 1

Score 1 of 1

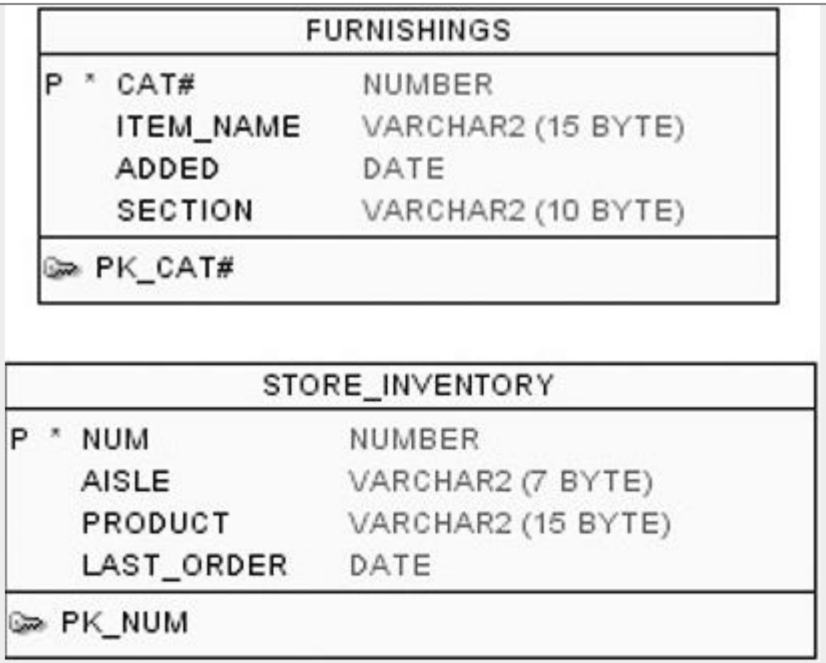
Question:

Review the first two illustrations and then review this SQL code



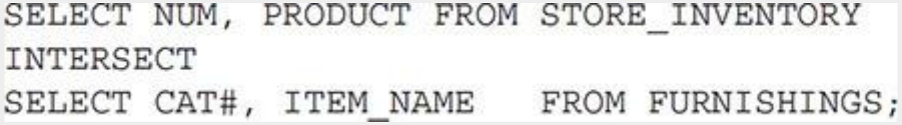
```
SELECT * FROM FURNISHING:

CAT#     ITEM_NAME    ADDED       SECTION
-----    -----        -------     --------
1        Side table   23-DEC-09   LR
2        Desk         12-SEP-09   BR
3        Towel        10-OCT-09   BA
```

```
SELECT * FROM STORE_INVENTORY:

NUM      AISLE         PRODUCT      LAST_ORDER
----     -----         -------      ----------
77       F02           Jacket       2009-09-09
78       B11           Towel        2009-11-11
79       SP01          Lava lamp    2009-12-21
```



```
FURNISHINGS
P * CAT#          NUMBER
    ITEM_NAME     VARCHAR2 (15 BYTE)
    ADDED         DATE
    SECTION       VARCHAR2 (10 BYTE)
PK_CAT#
```

```
STORE_INVENTORY
P * NUM           NUMBER
    AISLE         VARCHAR2 (7 BYTE)
    PRODUCT       VARCHAR2 (15 BYTE)
    LAST_ORDER    DATE
PK_NUM
```



```
SELECT NUM, PRODUCT FROM STORE_INVENTORY
INTERSECT
SELECT CAT#, ITEM_NAME   FROM FURNISHINGS;
```

How many rows will result from this query?

Response:

6

1

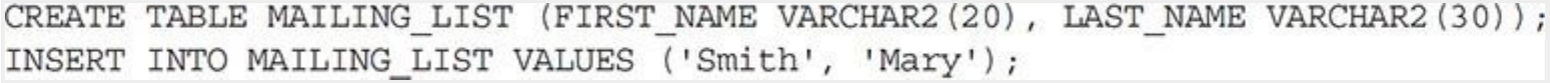
✓ 0

3

Score 1 of 1

Question:

Consider the following set of SQL statements:



```
CREATE TABLE MAILING_LIST (FIRST_NAME VARCHAR2(20), LAST_NAME VARCHAR2(30));
INSERT INTO MAILING_LIST VALUES ('Smith', 'Mary');
```

What will be the result of the INSERT statement?

Response:

It will fail because the last name and first name values are reversed.

✓ It will execute and create a new row in the table.

It will fail because there is no column list in the INSERT statement.

It will fail because there is no PRIMARY KEY in the table.

Score 1 of 1

Question:

Evaluate the following two queries:

```
SQL> SELECT cust_last_name, cust_city
FROM customers
WHERE cust_credit_limit IN (1000, 2000, 3000);

SQL> SELECT cust_last_name, cust_city
FROM customers
WHERE cust_credit_limit = 1000 or cust_credit_limit = 2000 or
cust_credit_limit = 3000
```

Which statement is true regarding the above two queries?

Response:

✓ There would be no change in performance.

Performance would improve in query 2.

Performance would degrade in query 2.

Performance would improve in query 2 only if there are null values in the CUST_CREDIT_LIMIT column.

Score 1 of 1

Question:

The WITH clause can be used to name a subquery. Which of the following is also true?
(Choose two.)

Response:

✓ The name of the subquery can be used in the SELECT statement following the WITH clause.

The name of the subquery is stored in the database by the WITH statement and can be referenced by other SQL statements in later sessions.

✓ The name of the subquery can be joined to other tables in the SELECT statement following the WITH clause.

The name of the subquery can be invoked from within the subquery that is named.

Score 1 of 1

Question:

Which of the following can a correlated subquery be used in?
(Choose three.)

Response:

✓ The WHERE clause of a DELETE statement

✓ The SET clause of an UPDATE statement

The FROM clause of a DELETE statement

✓ The WHERE clause of an UPDATE statement

Score 1 of 1

Question:

The DECODE expression always ends with:

Response:

Both of the above

The keyword END

Neither of the above

✓ A default expression to return if no other value matched the source expression

Score 1 of 1

Question:

Which two statements are true about Data Manipulation Language (DML) statements?

Response:

A DELETE FROM..... statement can remove multiple rows based on multiple conditions on a table.

An UPDATE...SET.... statement can modify multiple rows based on only a single condition on a table.

✓ AH INSERT INTO. . .VALUES. . statement can add multiple rows per execution to a table.

An UPDATE...SET... statement can modify multiple rows based on multiple conditions on a table.

An INSERT INTO...VALUES..... statement can add a single row based on multiple conditions on a table.

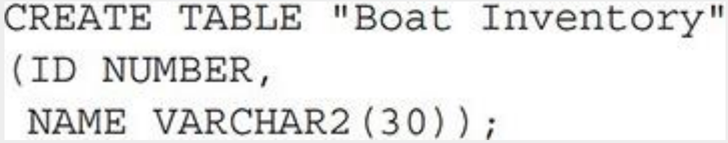
✓ A DELETE FROM ..... statement can remove rows based on only a single condition on a table.

Score 1 of 1

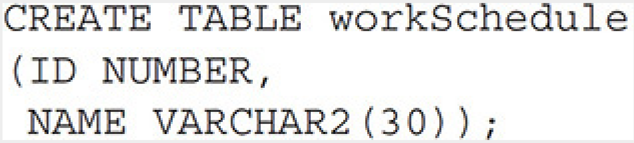
Question:

Which of the following are valid CREATE TABLE statements?
(Choose three.)

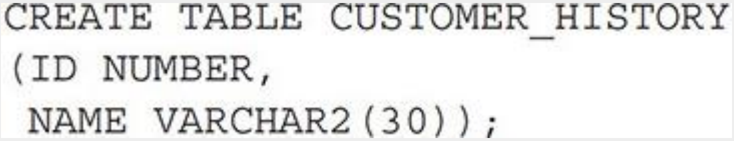
Response:

✓ 

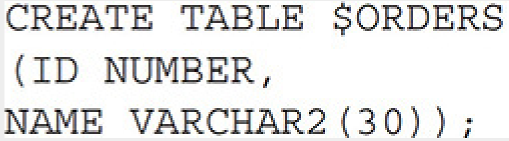
```
CREATE TABLE "Boat Inventory"
(ID NUMBER,
 NAME VARCHAR2(30));
```

✓ 

```
CREATE TABLE workSchedule
(ID NUMBER,
 NAME VARCHAR2(30));
```

✓ 

```
CREATE TABLE CUSTOMER_HISTORY
(ID NUMBER,
 NAME VARCHAR2(30));
```



```
CREATE TABLE $ORDERS
(ID NUMBER,
NAME VARCHAR2(30));
```

Score 1 of 1

Question:

Which of the following statements is false?

Response:

✓ It is possible to merge into two or more tables.

The USING clause can reference two or more tables.

It is possible to merge into a view.

You cannot perform an update to a column that is referenced in the ON clause.

Score 1 of 1

Question:

The 1Z0-071 exam (which is the subject of this book) has been officially validated by Oracle Corporation against which of the following versions of the Oracle database?
(Choose all that apply.)

Response:

✓ 12c

9i

✓ 11g

Every version

Score 1 of 1

Question:

Consider the following: SELECT MOD(5,3), REMAINDER(5,3) FROM DUAL; Which of the following will be the result?

Response:

✓ 2, −1

1, 2

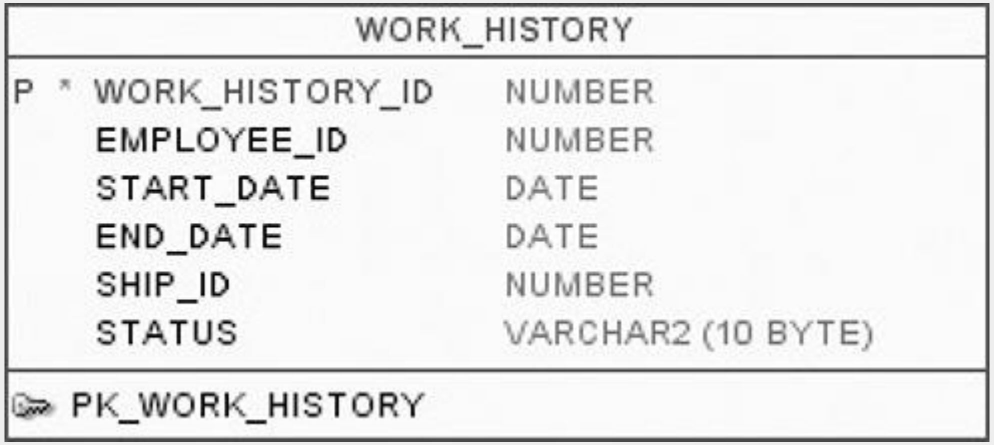
−1, 2

2, 1

Score 1 of 1

Question:

Review this WORK_HISTORY table.



WORK_HISTORY			
P	*	WORK_HISTORY_ID	NUMBER
		EMPLOYEE_ID	NUMBER
		START_DATE	DATE
		END_DATE	DATE
		SHIP_ID	NUMBER
		STATUS	VARCHAR2 (10 BYTE)

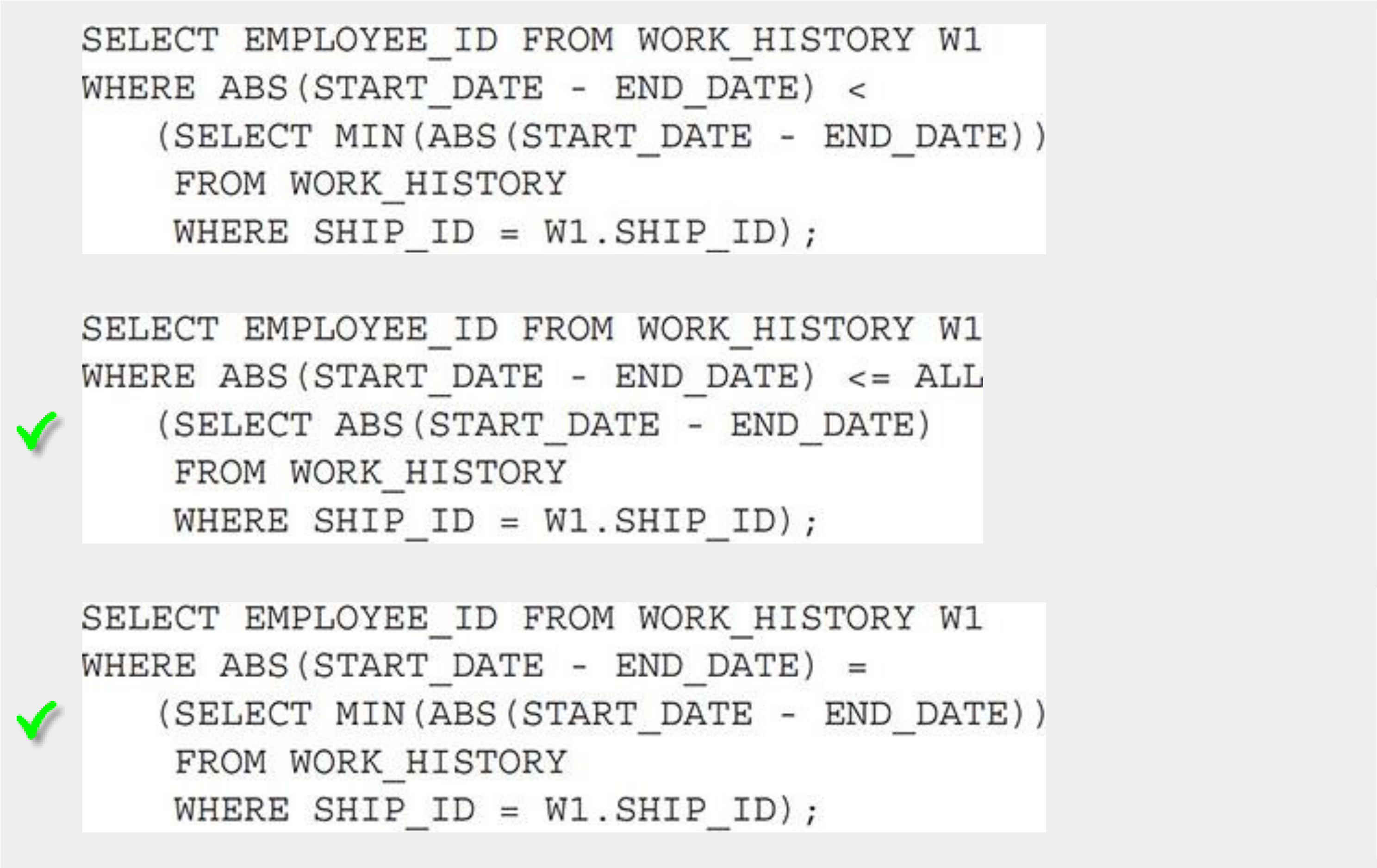
PK_WORK_HISTORY

Your task is to create a query that will list—for each ship—all of the EMPLOYEE_ID values for all the employees who have the shortest work history for their ship.

In other words, if there are two ships, you want to list all the employees assigned to the first ship who have the shortest work history, all the employees assigned to the second ship who have the shortest work history, and so on.

Which of the following queries will accomplish this task?
(Choose two.)

Response:



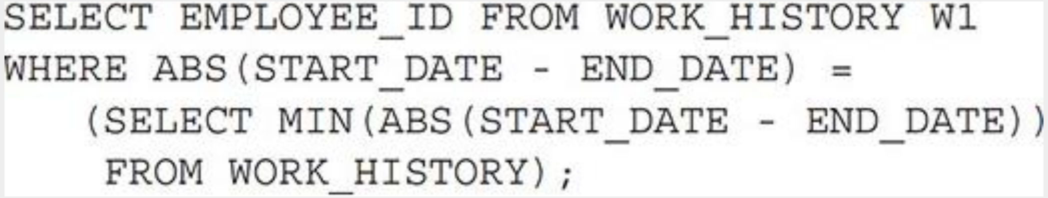
```
SELECT EMPLOYEE_ID FROM WORK_HISTORY W1
WHERE ABS(START_DATE - END_DATE) <
   (SELECT MIN(ABS(START_DATE - END_DATE))
    FROM WORK_HISTORY
    WHERE SHIP_ID = W1.SHIP_ID);
```

✓

```
SELECT EMPLOYEE_ID FROM WORK_HISTORY W1
WHERE ABS(START_DATE - END_DATE) <= ALL
   (SELECT ABS(START_DATE - END_DATE)
    FROM WORK_HISTORY
    WHERE SHIP_ID = W1.SHIP_ID);
```

✓

```
SELECT EMPLOYEE_ID FROM WORK_HISTORY W1
WHERE ABS(START_DATE - END_DATE) =
   (SELECT MIN(ABS(START_DATE - END_DATE))
    FROM WORK_HISTORY
    WHERE SHIP_ID = W1.SHIP_ID);
```



```
SELECT EMPLOYEE_ID FROM WORK_HISTORY W1
WHERE ABS(START_DATE - END_DATE) =
   (SELECT MIN(ABS(START_DATE - END_DATE))
    FROM WORK_HISTORY);
```

Score 1 of 1

Question:

View the Exhibits and examine the structures of the costs and promotions tables?

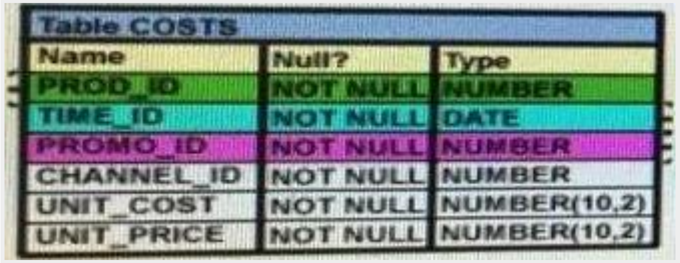


Table COSTS

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
PROMO_ID	NOT NULL	NUMBER
CHANNEL_ID	NOT NULL	NUMBER
UNIT_COST	NOT NULL	NUMBER(10,2)
UNIT_PRICE	NOT NULL	NUMBER(10,2)

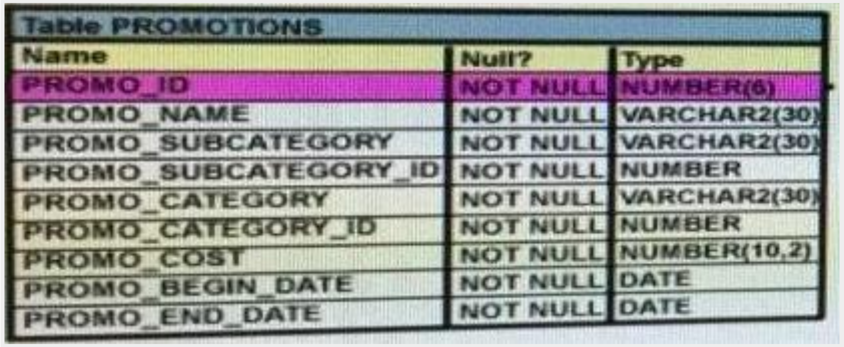


Table PROMOTIONS

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 prod_id FROM costs
WHERE promo_id IN (SELECT promo_id FROM promotions WHERE promo_cost < ALL (SELECT MAX(promo_cost) FROM promotions
GROUP BY (promo_end_datepromo_ begin_date)));
```

What would be the outcome of the above SQL statement?

Response:

It displays prod IDs in the promo with the lowest cost.

It displays prod IDs in the promos with the lowest cost in the same time interval.

It displays prod IDs in the promos with the highest cost in the same time interval.

✓ It displays prod IDs in the promos with cost less than the highest cost in the same time interval.

Score 1 of 1

Question:

Which task can be performed by using a single Data Manipulation Language (DML) statement?

Response:

removing all data only from one single column on which a primary key constraint is defined

adding a column constraint when inserting a row into a table

✓ removing all data only from one single column on which a unique constraint is defined

adding a column with a default value when inserting a row into a table

Score 1 of 1

Question:

Which two statements are true regarding subqueries?
(Choose two.)

Response:

✓ A subquery can retrieve zero or more rows.

A subquery can be used only in SQL query statements.

✓ A subquery can appear on either side of a comparison operator.

Only two subqueries can be placed at one level.

There is no limit on the number of subquery levels in the WHERE clause of a SELECT statement.

Score 1 of 1

Question:

What is one of the purposes of DDL?
(Choose the best answer.)

Response:

Issue privileges to users

Query data from a given table

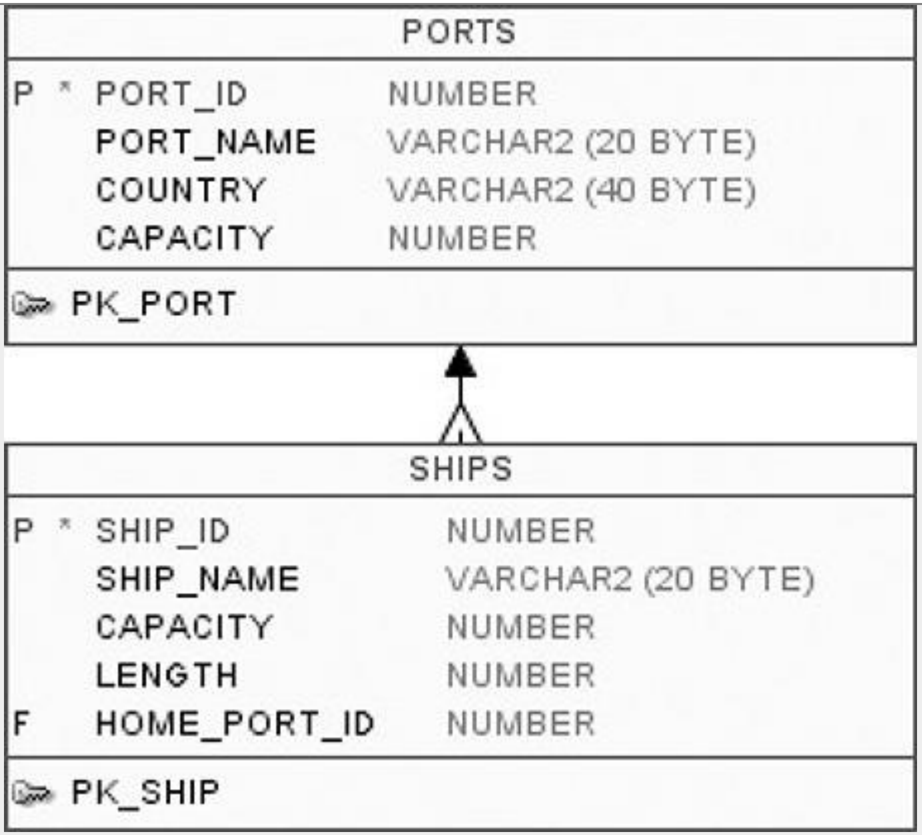
None of the above

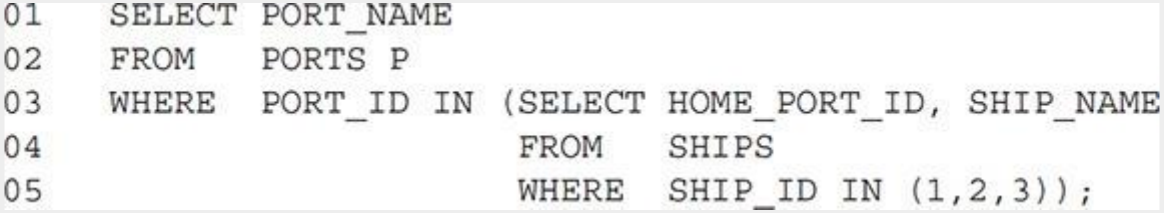
✓ Remove existing data from a database table

Score 1 of 1

Question:

Review the PORTS and SHIPS tables shown. Then review the following SQL code:





```
SELECT PORT_NAME
FROM   PORTS P
WHERE  PORT_ID IN (SELECT HOME_PORT_ID, SHIP_NAME
                   FROM   SHIPS
                   WHERE  SHIP_ID IN (1,2,3));
```

Which of the following is true of this statement?

Response:

The statement will fail with a syntax error because of line 5.

None of the above.

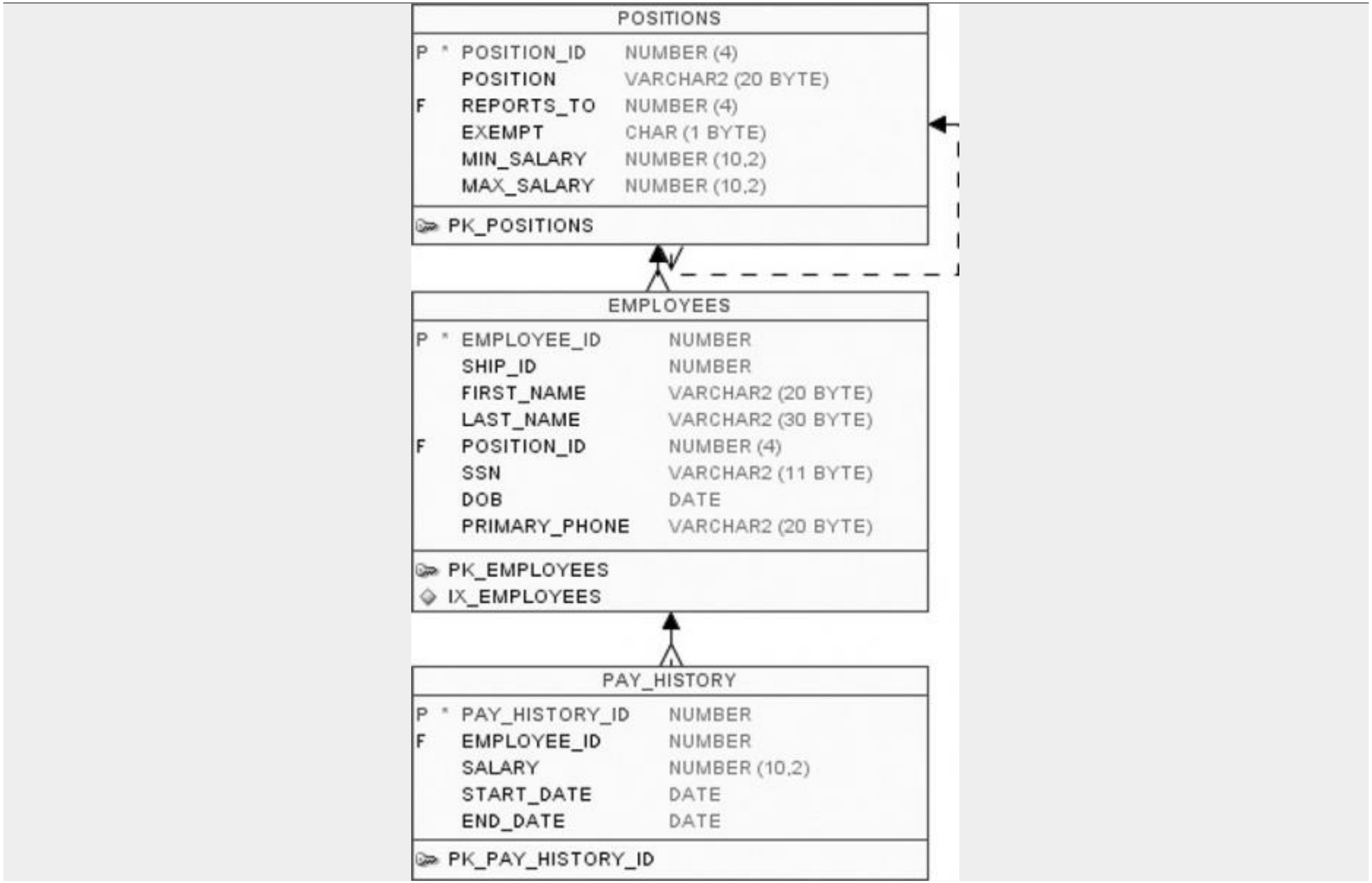
Whether the statement fails depends on how many rows are returned by the subquery in lines 3 through 5.

✓ The statement will fail with a syntax error because of line 3.

Score 1 of 1

Question:

Review the illustration. Which of the following is a valid self-join statement?
(Choose all that apply.)

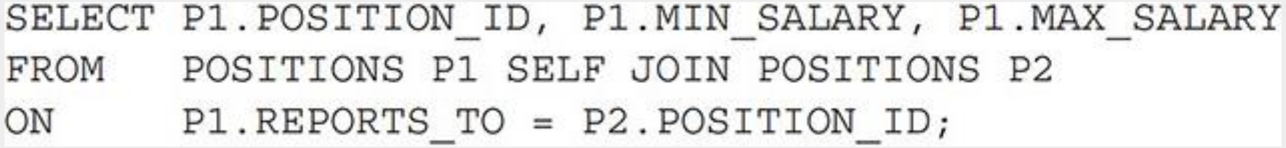


```
POSITIONS
P * POSITION_ID     NUMBER (4)
    POSITION        VARCHAR2 (20 BYTE)
F   REPORTS_TO      NUMBER (4)
    EXEMPT          CHAR (1 BYTE)
    MIN_SALARY      NUMBER (10,2)
    MAX_SALARY      NUMBER (10,2)

PK_POSITIONS
```

```
EMPLOYEES
P * EMPLOYEE_ID        NUMBER
    SHIP_ID            NUMBER
    FIRST_NAME         VARCHAR2 (20 BYTE)
    LAST_NAME          VARCHAR2 (30 BYTE)
F   POSITION_ID        NUMBER (4)
    SSN                VARCHAR2 (11 BYTE)
    DOB                DATE
    PRIMARY_PHONE      VARCHAR2 (20 BYTE)

PK_EMPLOYEES
◇ IX_EMPLOYEES
```

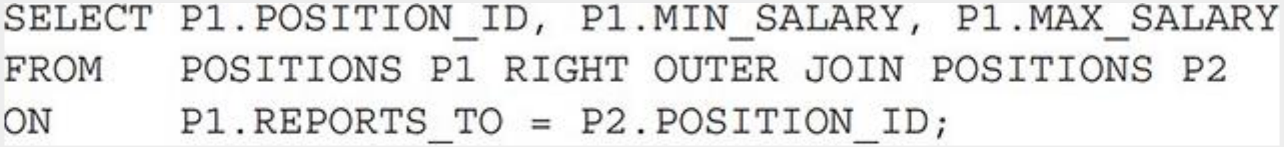
```
PAY_HISTORY
P * PAY_HISTORY_ID     NUMBER
F   EMPLOYEE_ID        NUMBER
    SALARY             NUMBER (10,2)
    START_DATE         DATE
    END_DATE           DATE

PK_PAY_HISTORY_ID
```

Response:



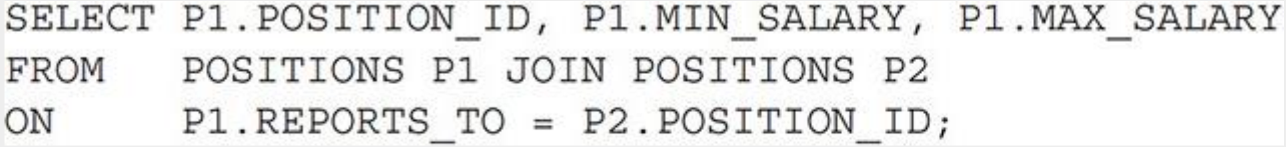
```
SELECT P1.POSITION_ID, P1.MIN_SALARY, P1.MAX_SALARY
FROM   POSITIONS P1 SELF JOIN POSITIONS P2
ON     P1.REPORTS_TO = P2.POSITION_ID;
```

✓



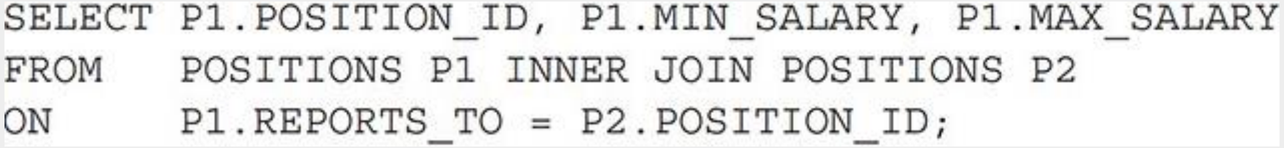
```
SELECT P1.POSITION_ID, P1.MIN_SALARY, P1.MAX_SALARY
FROM   POSITIONS P1 RIGHT OUTER JOIN POSITIONS P2
ON     P1.REPORTS_TO = P2.POSITION_ID;
```

✓



```
SELECT P1.POSITION_ID, P1.MIN_SALARY, P1.MAX_SALARY
FROM   POSITIONS P1 JOIN POSITIONS P2
ON     P1.REPORTS_TO = P2.POSITION_ID;
```

✓

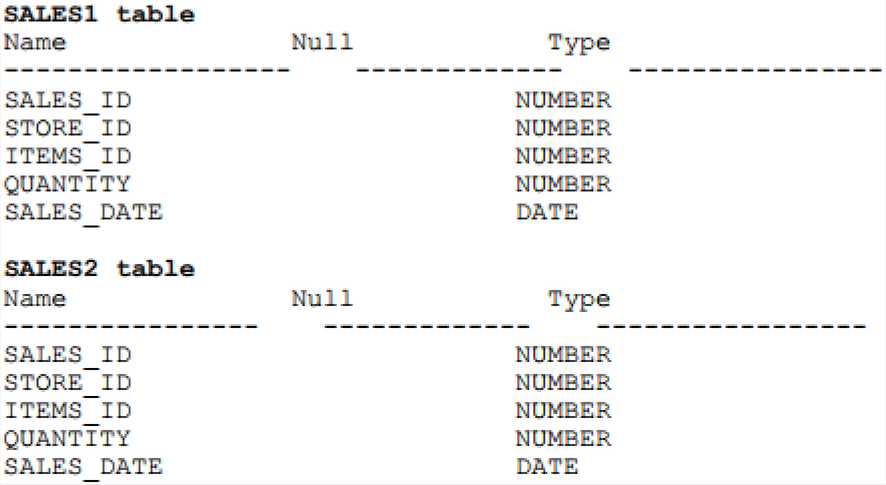


```
SELECT P1.POSITION_ID, P1.MIN_SALARY, P1.MAX_SALARY
FROM   POSITIONS P1 INNER JOIN POSITIONS P2
ON     P1.REPORTS_TO = P2.POSITION_ID;
```

Score 1 of 1

Question:

Sales data of a company is stored in two tables, SALES1 and SALES2, with some data being duplicated across the tables. You want to display the results from the SALES1 table, which are not present in the SALES2 table.



```
SALES1 table
Name                  Null               Type
------------------    -------------    ----------------
SALES_ID                                NUMBER
STORE_ID                                NUMBER
ITEMS_ID                                NUMBER
QUANTITY                                NUMBER
SALES_DATE                              DATE

SALES2 table
Name                  Null               Type
----------------    -------------    -----------------
SALES_ID                                NUMBER
STORE_ID                                NUMBER
ITEMS_ID                                NUMBER
QUANTITY                                NUMBER
SALES_DATE                              DATE
```

Which set operator generates the required output?

Response:

PLUS

✓ MINUS

UNION

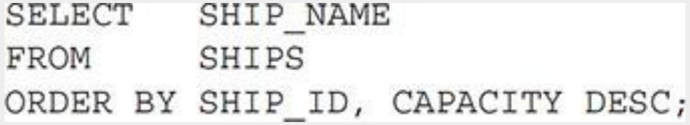
INTERSECT

SUBTRACT

Score 1 of 1

Question:

Review this SELECT statement:



```
SELECT   SHIP_NAME
FROM     SHIPS
ORDER BY SHIP_ID, CAPACITY DESC;
```

Assume that all table and column references exist within the database. What can be said of this SELECT statement?

Response:

The rows will sort in order by SHIP_ID and then by CAPACITY. All rows will sort in descending order.

✓ The rows will sort in order by SHIP_ID in ascending order and then by CAPACITY in descending order.

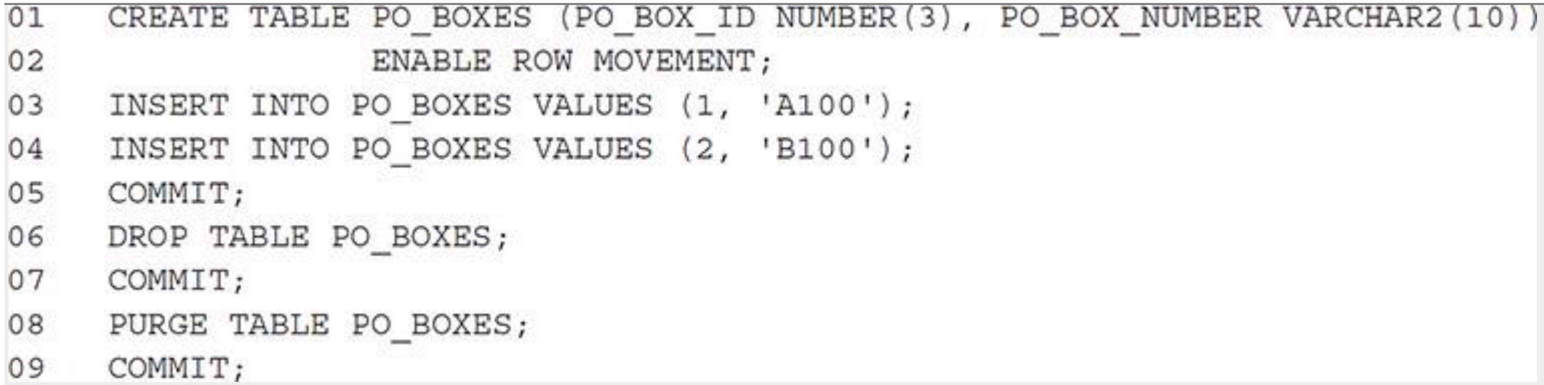
The statement will fail to execute because there is no WHERE clause.

The statement will fail to execute because the ORDER BY list includes a column that is not in the select list.

Score 1 of 1

Question:

Review the following SQL code:



```
CREATE TABLE PO_BOXES (PO_BOX_ID NUMBER(3), PO_BOX_NUMBER VARCHAR2(10))
             ENABLE ROW MOVEMENT;
INSERT INTO PO_BOXES VALUES (1, 'A100');
INSERT INTO PO_BOXES VALUES (2, 'B100');
COMMIT;
DROP TABLE PO_BOXES;
COMMIT;
PURGE TABLE PO_BOXES;
COMMIT;
```

What statement will recover the PO_BOXES table after these statements are executed?

Response:

FLASHBACK TABLE PO_BOXES TO TIMESTAMP SYSTIMESTAMP—INTERVAL ‘0 00:00:03’ DAY TO SECOND;

✓ None of the above—the table cannot be recovered.

FLASHBACK TABLE PO_BOXES TO BEFORE COMMIT;

FLASHBACK TABLE PO_BOXES TO BEFORE DROP;

Score 1 of 1

Question:

All database data is stored in:

Response:

None of the above

TABLES, VIEWS, and SEQUENCES

✓ TABLES

TABLES and VIEWS

Score 1 of 1

Question:

The purpose of NULLIF is to:

Response:

Return a NULL if a single column is NULL

Return a NULL if a single expression is NULL

Both of the above

✓ None of the above

Score 1 of 1

Question:

View the Exhibit and examine the structure of CUSTOMERS table.

Using the CUSTOMERS table, you need 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?

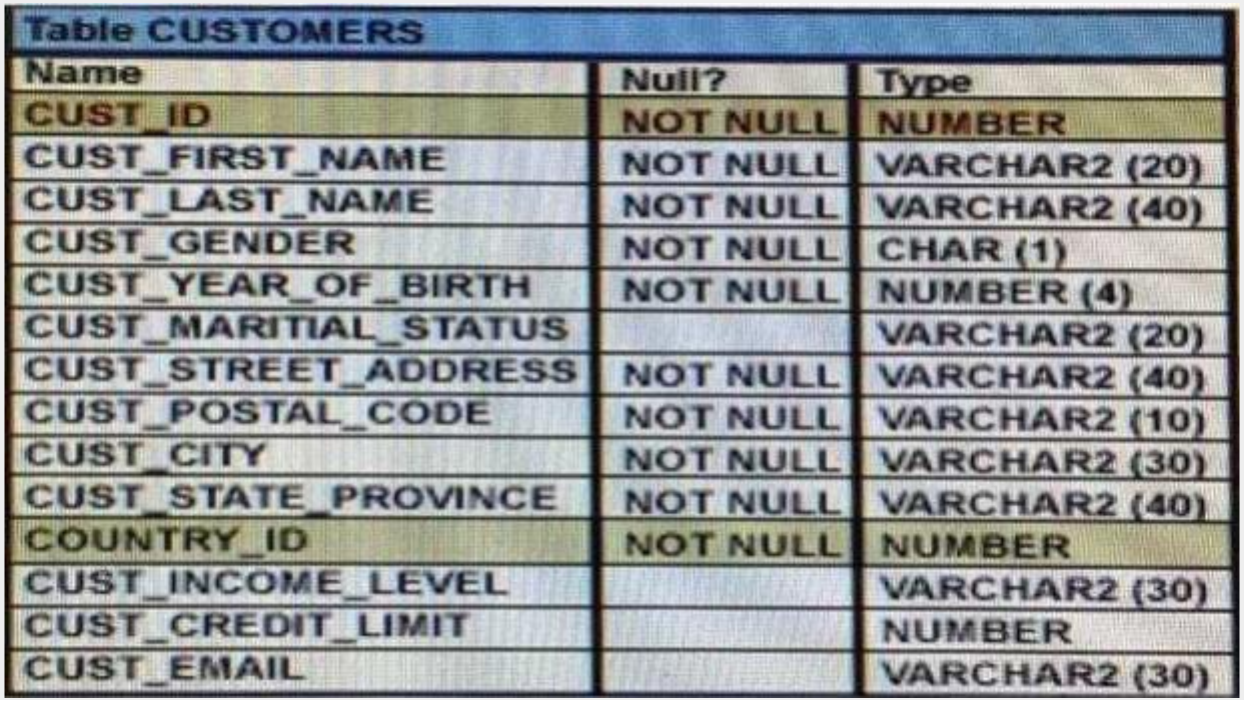


Table CUSTOMERS

Name	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_FIRST_NAME	NOT NULL	VARCHAR2 (20)
CUST_LAST_NAME	NOT NULL	VARCHAR2 (40)
CUST_GENDER	NOT NULL	CHAR (1)
CUST_YEAR_OF_BIRTH	NOT NULL	NUMBER (4)
CUST_MARITIAL_STATUS		VARCHAR2 (20)
CUST_STREET_ADDRESS	NOT NULL	VARCHAR2 (40)
CUST_POSTAL_CODE	NOT NULL	VARCHAR2 (10)
CUST_CITY	NOT NULL	VARCHAR2 (30)
CUST_STATE_PROVINCE	NOT NULL	VARCHAR2 (40)
COUNTRY_ID	NOT NULL	NUMBER
CUST_INCOME_LEVEL		VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2 (30)

Response:

✓ SELECT NVL (TO CHAR(cust_credit_limit *.15), 'Not Available') "NEW CREDIT" FROM customers;

SELECT TO_CHAR (NVL(cust_credit_limit *.15), 'Not Available') "NEW CREDIT" FROM customers;

SELECT NVL(cust_credit_limit *.15), 'Not Available') "NEW CREDIT" FROM customers;

SELECT NVL(cust_credit_limit), 'Not Available') "NEW CREDIT" FROM customers;

Score 1 of 1

Question:

Review the following SQL statement: TRUNCATE personnel; Which of the following is true of the previous statement?
(Choose all that apply.)

Response:

✓ The statement will fail.

The statement will result in an implicit commit.

The statement will not fire any DML triggers on the table.

The statement will remove all data from any INDEX objects associated with that table.

Score 0 of 1

Question:

The USER_CONSTRAINTS view in the data dictionary lists FOREIGN KEY constraints in the CONSTRAINT_TYPE column with which of the following single-letter abbreviations?

Response:

K

G

✓ R

✗ F

Score 1 of 1

Question:

Which two statements are true regarding the GROUP BY clause in a SQL statement?
(Choose two.)

Response:

Using the WHERE clause after the GROUP BY clause excludes the rows after creating groups.

The GROUP BY clause is mandatory if you are using an aggregate function in the SELECT clause.

✓ Using the WHERE clause before the GROUP BY clause excludes the rows before creating groups.

You can use column alias in the GROUP BY clause.

✓ If the SELECT clause has an aggregate function, then those individual columns without an aggregate function in the SELECT clause should be included in the GROUP BY cause.