Question:

The BOOKS_TRANSACTIONS table exists in your schema in this database. You execute this SQL statement when connected to your schema in your database instance.

SQL> SELECT * FROM books transactions ORDER BY 3;

What is the result?

Response:

- ✓ All table rows are displayed sorted in ascending order of the values in the third column.
- The first three rows in the table are displayed in the order that they are stored.
- The execution fails unless the numeral 3 in the ORDER BY clause is replaced by a column name.
- Only the three rows with the lowest values in the key column are displayed in the order that they are stored.

Score 1 of 1

Question:

The unique identifier of a row in a database table is a(n):

Response:

ID

✓ Primary key

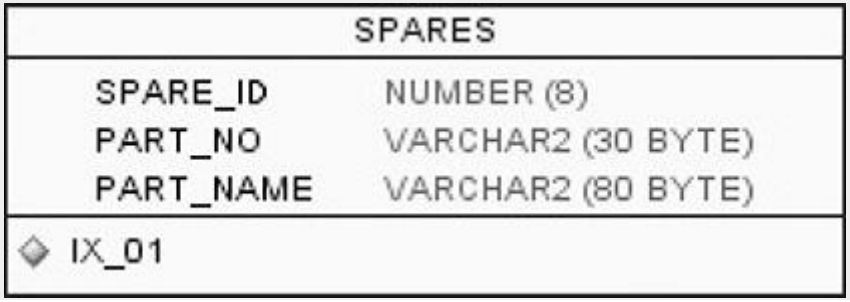
Primary column

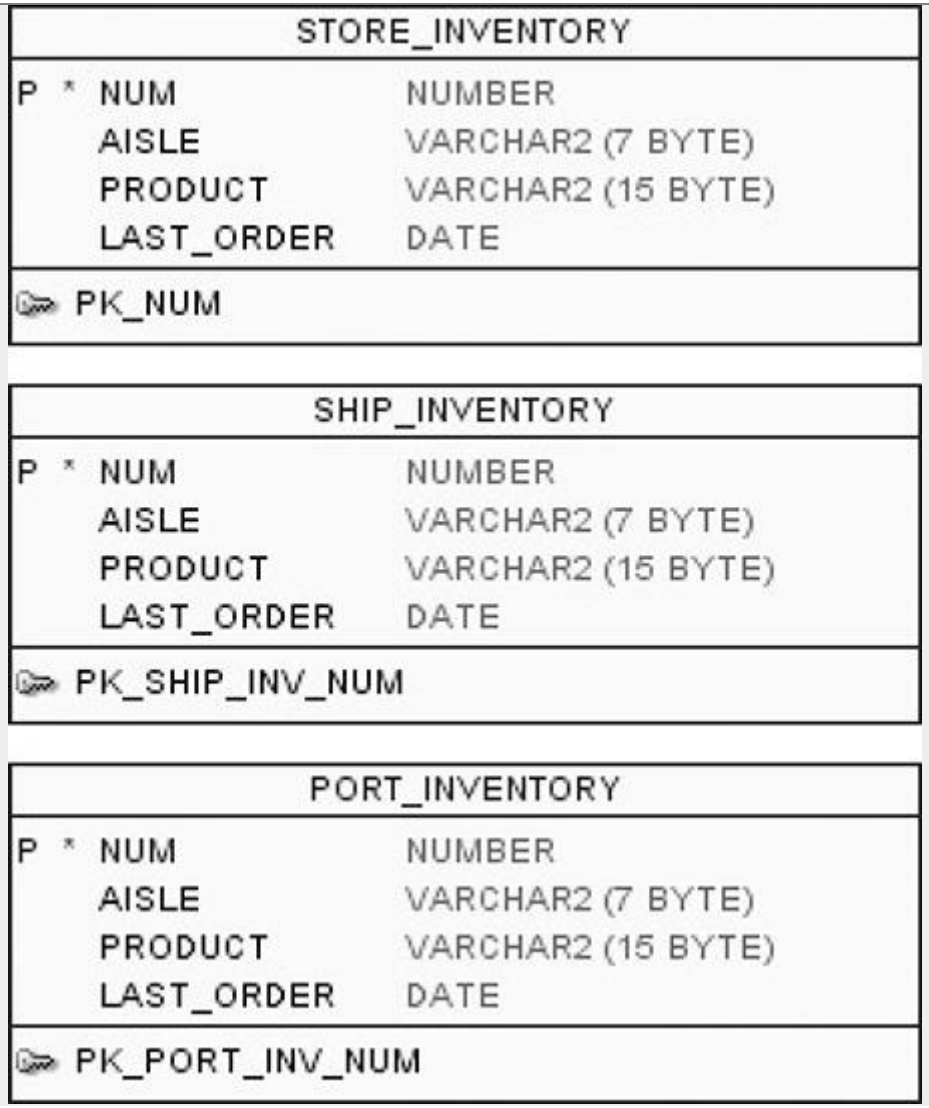
Column

Score 1 of 1

Question:

Review the diagrams and examine the following statement:




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

```
01   INSERT
02     WHEN (PART_NO < 500) THEN
03      INTO STORE_INVENTORY (NUM, PRODUCT)
04       VALUES (SPARE_ID, PART_NAME)
05      INTO PORT_INVENTORY (NUM, PRODUCT)
06       VALUES (SPARE_ID, PART_NAME)
07     WHEN (PART_NO >= 500) THEN
08       INTO SHIP_INVENTORY (NUM, PRODUCT)
09       VALUES (SPARE_ID, PART_NAME)
10     SELECT SPARE_ID, PART_NO, PART_NAME
11     FROM   SPARES;
```

Which of the following statements is true for this SQL statement?

Response:

If the first WHEN condition in line 2 is true, the INTO clause in line 3 and line 4 will be executed, after which processing will skip to the next row returned by the subquery.

✓ Regardless of whether the first WHEN condition is true, the second WHEN condition will be evaluated.

If the first WHEN condition in line 2 is true, the WHEN condition in line 7 will not be evaluated.

No matter which WHEN condition is true, the INTO clause in line 5 will be executed regardless.

Score 1 of 1

Question:

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

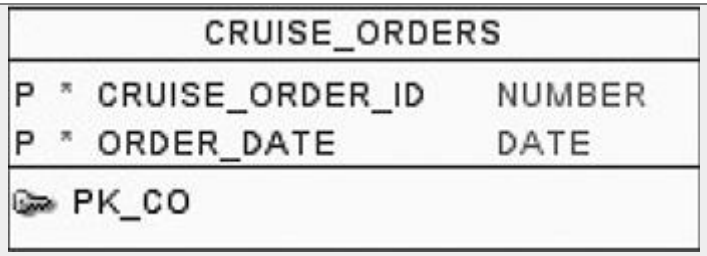
Response:

- ✓ The SET clause of an UPDATE statement
- The FROM clause of a DELETE statement
- ✓ The WHERE clause of a DELETE statement
- ✓ The WHERE clause of an UPDATE statement

Score 1 of 1

Question:

Review the illustration and then look at the SQL code that follows:




```
01  SELECT   TO_CHAR(ORDER_DATE,'Q') "Quarter", COUNT(*)
02  FROM     CRUISE_ORDERS
03  WHERE    TO_CHAR(ORDER_DATE,'YYYY') = '2009'
04  GROUP BY TO_CHAR(ORDER_DATE,'Q');
```

Recall that the 'Q' format model is for quarter, so TO_CHAR using a DATE data type with the 'Q' format mask is translating the date into the quarter in which it falls—1, 2, 3, or 4.

Given that, which of the following statements is true of the SQL statement?

Response:

None of the above.

It will fail because of a syntax error in line 1 since you cannot use the TO_CHAR function with the COUNT aggregate function.

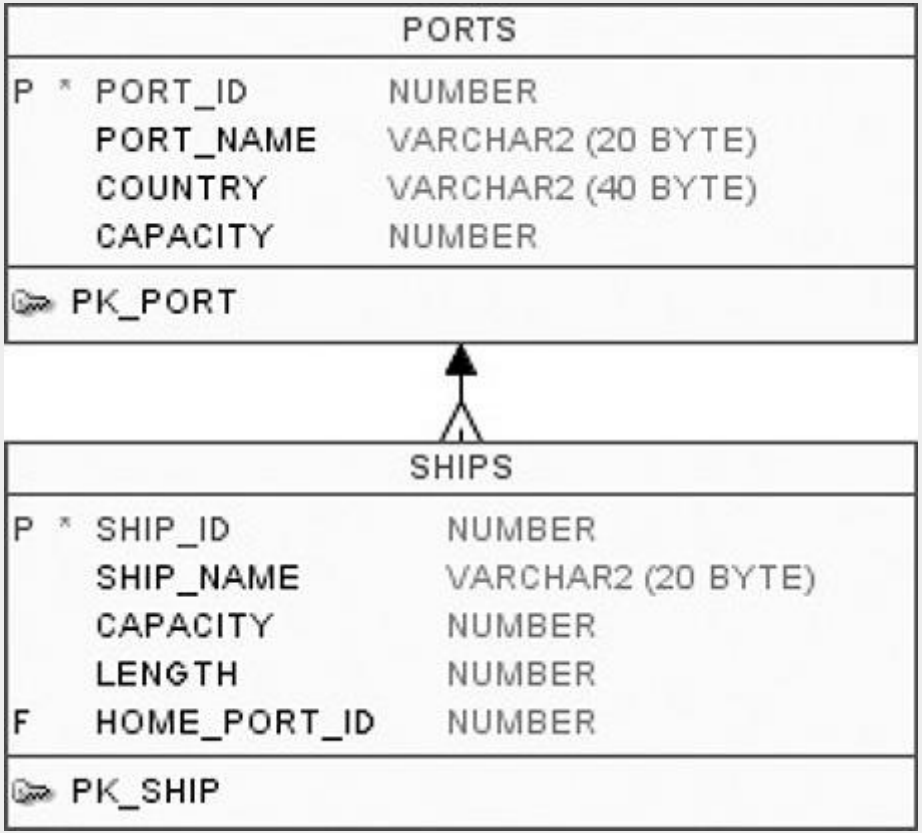
It will fail because of a syntax error in line 4 since you cannot use the TO_CHAR function in the GROUP BY clause.

It will execute and show the number of orders in the CRUISE_ORDERS table for each quarter in the year 2009.

Score 1 of 1

Question:

Review the illustration and the following SQL code:



```
01  UPDATE PORTS P
02  SET    CAPACITY = CAPACITY + 1
03  WHERE  EXISTS (SELECT *
04                 FROM SHIPS
05                 WHERE HOME_PORT_ID = P.PORT_ID);
```

The PORTS table has 15 rows. The SHIPS table has 20 rows. Each row in PORTS has a unique value for PORT_ID. Each PORT_ID value is represented in the HOME_PORT_ID column of at least one row of the SHIPS table.

What can be said of this UPDATE statement?

Response:

The value for CAPACITY will not increase.

✔ The value for CAPACITY will increase once for each of the 15 rows in the PORTS table.

The statement will fail to execute because of an error in the syntax.

The value for CAPACITY will increase by 20 for each of the 15 rows in the PORTS table.

Score 1 of 1

Question:

Which two statements are true regarding multiple-row subqueries?

(Choose two.)

Response:

They can be used to retrieve multiple rows from a single table only.

They always contain a subquery within a subquery.

✓ They can contain group functions.

✓ They should not be used with the NOT IN operator in the main query if NULL is likely to be a part of the result of the subquery.

They use the < ALL operator to imply less than the maximum.

Score 1 of 1

Question:

Review this SQL statement: SELECT MONTHS_BETWEEN(LAST_DAY('15-JAN-12')+1,'01-APR-12')FROM DUAL; What will result from this query?

Response:

< −2 (some number less than negative 2)

✓ −2

2

> 2 (some number greater than 2)

Score 0 of 1

Question:

Equijoins look for:

Response:

Ranges of data matches

None of the above

✓ Exact data matches

✗ Comparisons using any comparison operator provided that the resulting correlations occur in both tables

Score 1 of 1

Question:

Which of the following forms of subquery never returns more than one row?

Response:

✓ Scalar

Correlated

Multiple-column

None of the above

Score 1 of 1

Question:

What can an INSERT statement do?
(Choose two.)

Response:

✓ Add data into more than one column in a table

✓ Add rows into more than one table

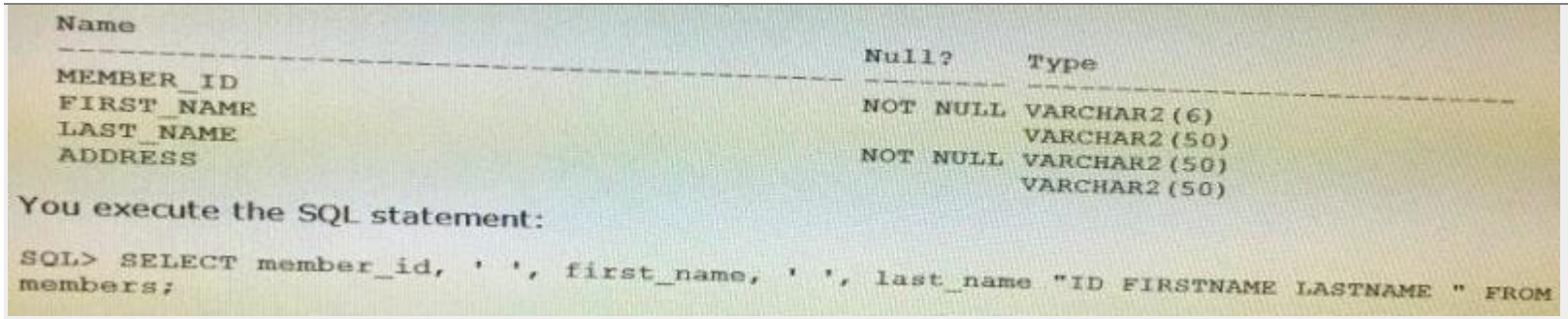
Delete rows by overwriting them

Join tables together

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)
```

You execute the SQL statement:

```
SQL> SELECT member_id, ' ', first_name, ' ', last_name "ID FIRSTNAME LASTNAME " FROM
members;
```

What is the outcome?

Response:

It fails because the space specified in single quotation marks after the first two column names is invalid.

It executes successfully and displays the column details in a single column with only the alias column heading.

✔ It executes successfully and displays the column details in three separate columns and replaces only the last column heading with the alias.

It fails because the alias name specified after the column names is invalid.

Score 0 of 1

Question:

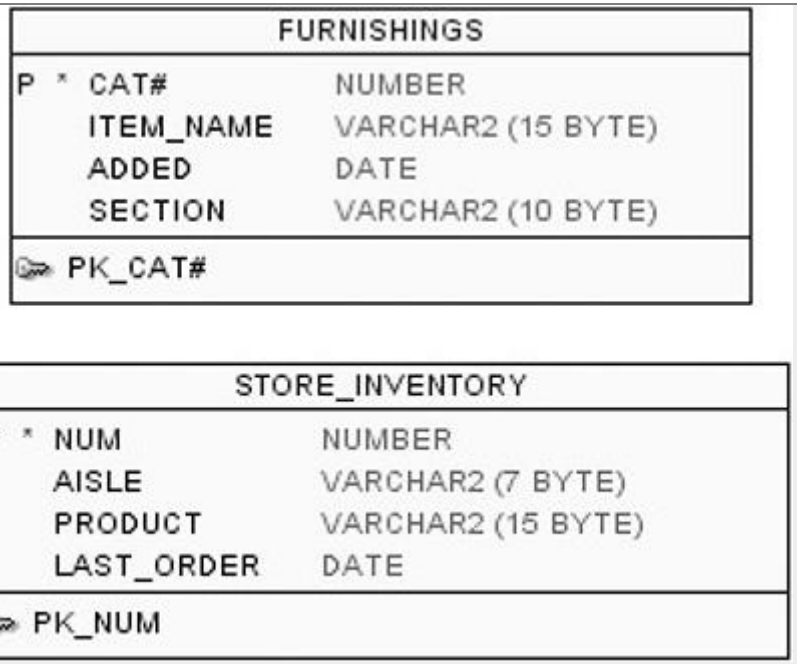
Review the first two illustrations and then review this SQL code:

```
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
```




```
01   SELECT '--', SECTION
02   FROM   FURNISHINGS
03   WHERE  CAT# NOT IN (1,2)
04   UNION ALL
05   SELECT TO_CHAR(LAST_ORDER,'Month'), AISLE
06   FROM   STORE_INVENTORY;
```

How many rows will result from this query?

Response:

✗ 0

It will not execute because it will fail with a syntax error.

6

✓ 4

Score 1 of 1

Question:

View the Exhibit and examine the ORDERS table. The ORDERS table contains data and all orders have been assigned a customer ID. Which statement would add a NOT NULL constraint to the CUSTOMER_ID column?

ORDERS

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER(4)
ORDATE	DATE	DATE
CUSTOMER_ID		NUMBER(3)
ORDER_TOTAL		NUMBER(7,2)

Response:

✓ ALTER TABLE orders
MODIFY customer_id CONSTRAINT orders_cust_nn NOT NULL (customer_id);

ALTER TABLE orders
ADD CONSTRAINT orders_cust_id_nn NOT NULL (customer_id);

```
ALTER TABLE orders
MODIFY CONSTRAINT orders_cust_id_nn NOT NULL (customer_id);


ALTER TABLE orders
ADD customer_id NUMBER(6)CONSTRAINT orders_cust_id_nn NOT NULL;
```

Score 1 of 1

Question:

See the Exhibit and Examine the structure of the CUSTOMERS table:

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)

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?

Response:

SELECT NVL(cust_credit_limit,'Not Available')*1.15 "NEW CREDIT" FROM customers;

SELECT NVL(cust_credit_limit*1.15,'Not Available') "NEW CREDIT" FROM customers;

SELECT TO_CHAR(NVL(cust_credit_limit*1.15,'Not Available')) "NEW CREDIT" FROM customers;

✓ SELECT NVL(TO_CHAR(cust_credit_limit*1.15),'Not Available') "NEW CREDIT" FROM customers;

Score 1 of 1

Question:

If you are using an ORDER BY to sort values in descending order, in which order will they appear?

Response:

If the data type is character, the value '130' will appear first before '75'.

If the data type is date, the value for June 25, 2010, will appear before the value for August 29, 2010.

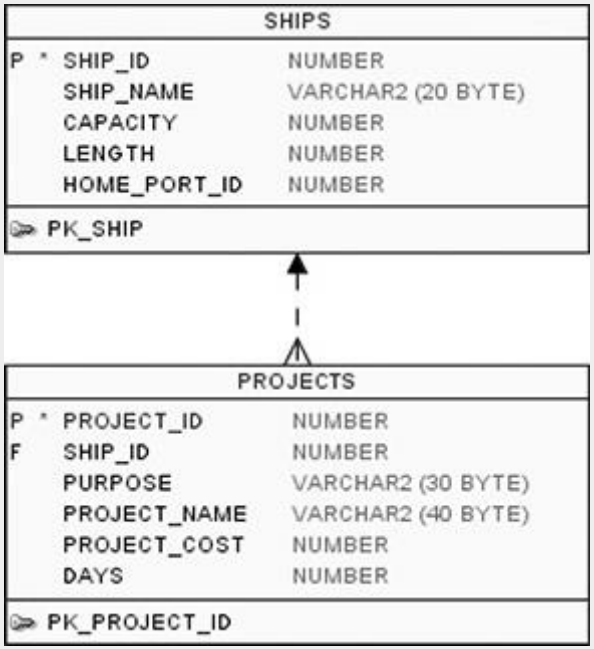
✓ If the data type is character, the value 'Michael' will appear first before the value 'Jackson'.

If the data type is numeric, the value 400 will appear first before the value 800.

Score 1 of 1

Question:

Review the illustration and the following SQL code:




```
CREATE OR REPLACE VIEW PROJECTS_ROLLUP AS
       SELECT SHIP_NAME, CAPACITY,
              COUNT(PROJECT_ID) NUM_PROJECTS, ROUND(SUM(DAYS)) TOTAL_DAYS
       FROM   SHIPS A JOIN PROJECTS B
       ON     A.SHIP_ID = B.SHIP_ID
       GROUP BY SHIP_NAME, CAPACITY;
```

What can be said of this code?

Response:

- ✓ After the view is created, a valid SELECT statement will work on the PROJECTS_ROLLUP view, but an INSERT will not.

- The attempt to create the view will fail because you cannot create a VIEW with a SELECT statement that uses a GROUP BY clause.

- After the view is created, a valid SELECT and valid INSERT statement will work on the PROJECTS_ROLLUP view.

- The attempt to create the view will fail because you cannot create a VIEW with a SELECT statement that is a join.

Score 1 of 1

Question:

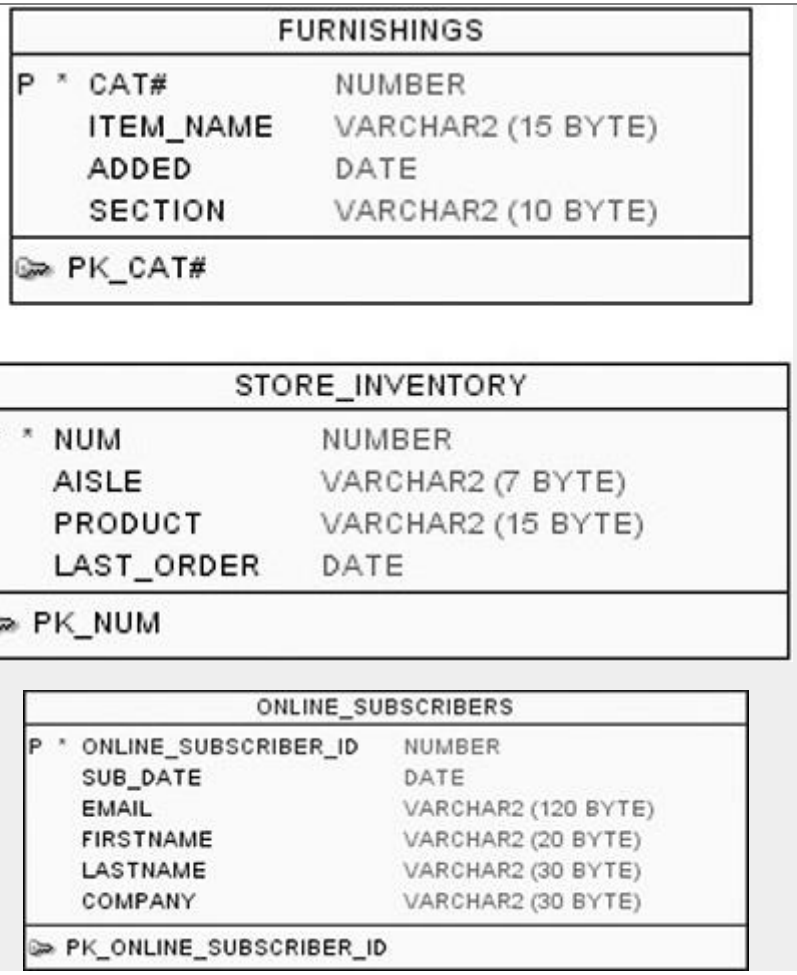
Review the first two illustrations as well as the ONLINE_SUBSCRIBERS table and then review this SQL code:

```
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
ONLINE_SUBSCRIBERS
P * ONLINE_SUBSCRIBER_ID NUMBER
SUB_DATE DATE
EMAIL VARCHAR2 (120 BYTE)
FIRSTNAME VARCHAR2 (20 BYTE)
LASTNAME VARCHAR2 (30 BYTE)
COMPANY VARCHAR2 (30 BYTE)
PK_ONLINE_SUBSCRIBER_ID

```
SELECT ONLINE_SUBSCRIBER_ID, EMAIL
FROM   ONLINE_SUBSCRIBERS;

ONLINE_SUBSCRIBER_ID   EMAIL
---------------------- ------------------------------
1                      pendicott77@kasteelinc.com
2                      watcher@foursigma.org
3                      hardingpal@ckofca.com
```

```
01   SELECT  (SELECT PRODUCT FROM STORE_INVENTORY
02            INTERSECT
03            SELECT ITEM_NAME FROM FURNISHINGS)
04   FROM     ONLINE_SUBSCRIBERS;
```

What will happen when this SQL statement is executed?

Response:

- ✓ It will execute and repeat the value 'Towel' for each row of the ONLINE_SUBSCRIBERS table.
- It will fail with a general syntax error.
- It will execute, but the INTERSECT will not work correctly.
- It will fail with an execution error.

Score 1 of 1

Question:

Which two statements are true regarding constraints?

(Choose two.)

Response:

- ✓ A column with the UNIQUE constraint can contain NULL.
- All the constraints can be defined at the column level as well as the table level
- A foreign key cannot contain NULL values.
- A constraint is enforced only for the INSERT operation on a table.
- ✓ A constraint can be disabled even if the constraint column contains data.

Score 1 of 1

Question:

Conversion functions:

Response:

- Are not required because SQL performs automatic data type conversion where necessary.

Are similar to ALTER TABLE … MODIFY statements.

✓ Change a value's data type in an equation to tell SQL to treat the value as that specified data type.

Change a column's data type so that future data stored in the table will be preserved in the converted data type.

Score 0 of 1

Question:

User HARDING owns a table TEAPOT. User HARDING then executes the following SQL statements to give access to the table to user ALBERT:

```
CREATE PUBLIC SYNONYM TEAPOT FOR HARDING.TEAPOT;
CREATE ROLE DOME;
GRANT DOME TO ALBERT;
GRANT SELECT ON TEAPOT TO DOME;
```

Which of the following statements can user ALBERT now execute on the TEAPOT table?

Response:

SELECT * FROM HARDING.DOME.TEAPOT;

✗ SELECT * FROM DOME.HARDING.TEAPOT;

None of the above

SELECT * FROM HARDING.TEAPOT;

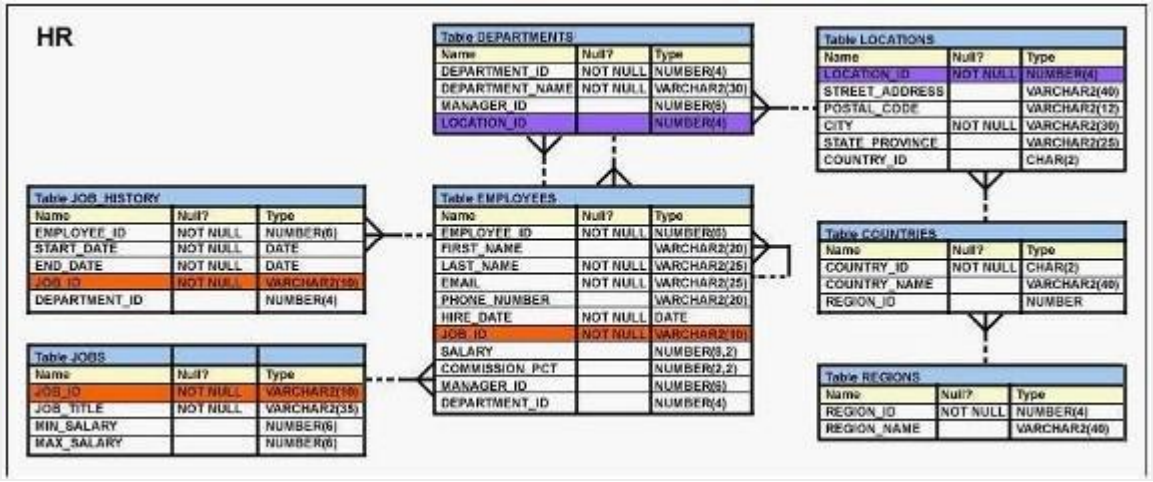
Score 1 of 1

Question:

View the Exhibit and examine the structure of the EMPLOYEES table. You want to display all employees and their managers having 100 as the MANAGER_ID.

You want the output in two columns: the first column would have the LAST_NAME of the managers and the second column would have LAST_NAME of the employees.

Which SQL statement would you execute?



Response:

```
SELECT m.last_name "Manager", e.last_name "Employee"
FROM employees m JOIN employees e
WHERE m.employee_id = e.manager_id and AND e.manager_id = 100
```

```
SELECT m.last_name "Manager", e.last_name "Employee"
FROM employees m JOIN employees e
ON m.employee_id = e.manager_id
WHERE m.manager_id = 100;
```

```
SELECT m.last_name "Manager", e.last_name "Employee"
FROM employees m JOIN employees e
ON e.employee_id = m.manager_id
WHERE m.manager_id = 100;
```

```
SELECT m.last_name "Manager", e.last_name "Employee"
FROM employees m JOIN employees e
ON m.employee_id = e.manager_id
WHERE e.manager_id = 100;
```

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

Score 1 of 1

Question:

To list all the currently defined variables, use:

Response:

- SHOW DEFINE
- ✓ DEFINE

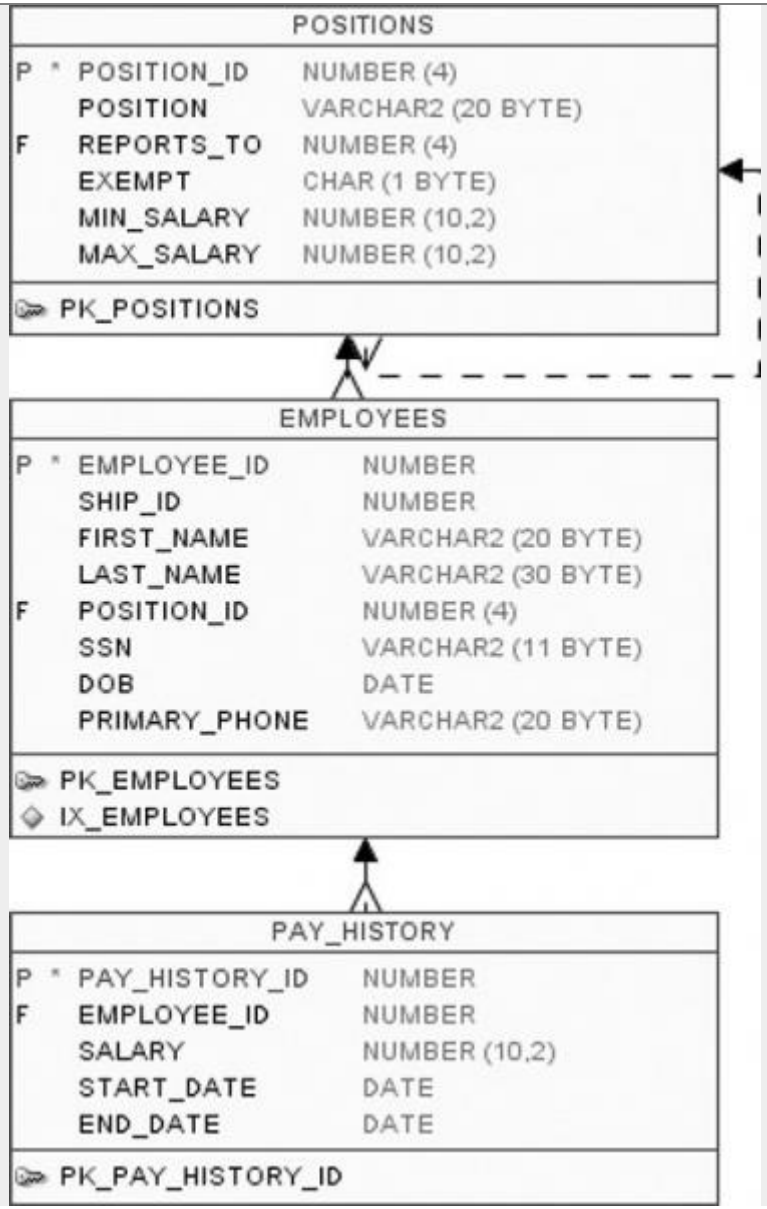
DEFINE ALL

SHOW ALL

Score 0 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 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 INNER JOIN POSITIONS P2
ON     P1.REPORTS_TO = P2.POSITION_ID;
```

```
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 JOIN POSITIONS P2
ON     P1.REPORTS_TO = P2.POSITION_ID;
```

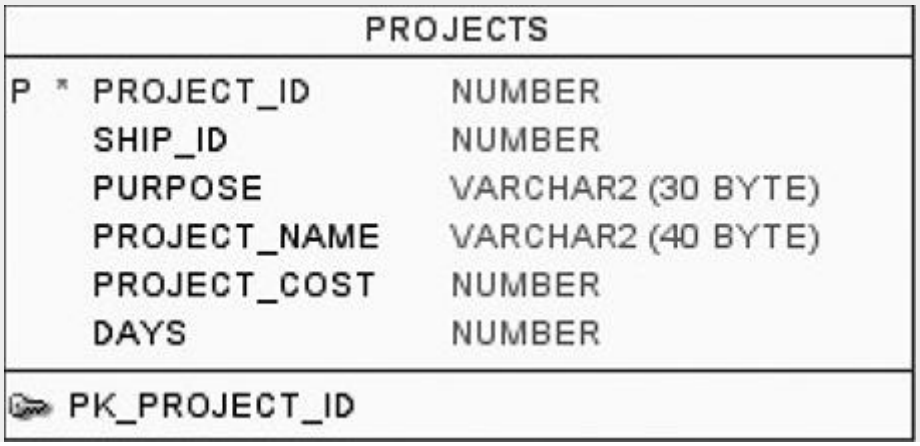
Score 1 of 1

Question:

Review the illustration. Your assignment: create a SELECT statement that queries the PROJECTS table to show the average project cost for each PURPOSE.

You know there are only two values for PURPOSE in the table: 'Upgrade' and 'Maintenance'. You want to restrict output to those rows where DAYS is greater than 3.

Which of the following SELECT statements will perform this task?



PROJECTS		
P *	PROJECT_ID	NUMBER
	SHIP_ID	NUMBER
	PURPOSE	VARCHAR2 (30 BYTE)
	PROJECT_NAME	VARCHAR2 (40 BYTE)
	PROJECT_COST	NUMBER
	DAYS	NUMBER
PK_PROJECT_ID		

Response:

```
SELECT PURPOSE, AVG(PROJECT_COST)
FROM PROJECTS
WHERE DAYS > 3
GROUP BY PURPOSE, DAYS
HAVING DAYS > 3;
```

✓

```
SELECT PURPOSE, AVG(PROJECT_COST)
FROM PROJECTS
WHERE DAYS > 3
GROUP BY PURPOSE;
```

```
SELECT PURPOSE, AVG(PROJECT_COST)
FROM PROJECTS
GROUP BY PURPOSE
HAVING DAYS > 3;

SELECT PURPOSE, AVG(PROJECT_COST)
FROM PROJECTS
GROUP BY PURPOSE, (DAYS > 3);
```

Score 1 of 1

Question:

Examine the structure of the BOOKS_TRANSACTIONS table:

```
Name                      Null?             Type
----------------     -----------------     ----------------------------
TRANSACTION_ID       NOT NULL            VARCHAR2 (6)
BORROWED_DATE                            VARCHAR2 (50)
DUE_DATE                                 DATE
BOOK_ID                                  DATE
MEMBER_ID                                VARCHAR2 (6)
```

You want to display the member IDs, due date, and late fee as $2 for all transactions. Which SQL statement must you execute?

Response:

✔ SELECT member_id AS "MEMBER ID", due_date AS "DUE DATE", '$2' AS "LATE FEE"
FROM BOOKS_TRANSACTIONS;

SELECT member_id AS "MEMBER ID", due_date AS "DUE DATE", $2 AS "LATE FEE" FROM BOOKS_TRANSACTIONS;

SELECT member_id AS MEMBER_ID, due_date AS DUE_DATE, $2 AS LATE_FEE FROM BOOKS_TRANSACTIONS;

SELECT member_id 'MEMBER ID', due_date 'DUE DATE', '$2 AS LATE FEE' FROM BOOKS_TRANSACTIONS;

Score 0 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 result in an implicit commit.

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

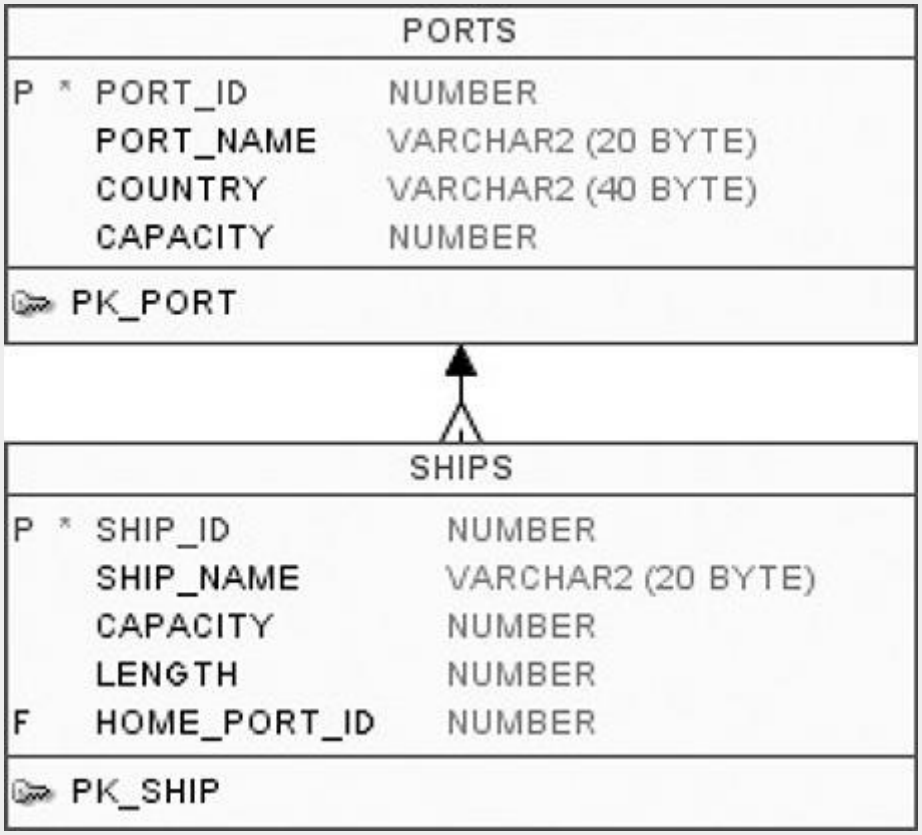

The statement will fail.


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

Score 1 of 1

Question:

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




```
01   SELECT PORT_NAME
02   FROM   PORTS P
03   WHERE  PORT_ID IN (SELECT HOME_PORT_ID, SHIP_NAME
04                      FROM   SHIPS
05                      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 3.
- 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 5.
- None of the above.

Score 0 of 1

Question:

You issue the following command to drop the PRODUCTS table:
SQL > DROP TABLE products;

Which three statements are true about the implication of this command?

Response:

- All data in the table is deleted but the table structure remains.
- ✗ All indexes on the table remain but they are invalidated.

- A pending transaction in the session is committed.
- All data along with the table structure is deleted.
- All views and synonyms on the table remain but they are invalidated.

Score 1 of 1

Question:

Which of the following is true about aggregate functions?
(Choose two.)

Response:

- Are also called group functions.
- Can operate only with numeric data.
- Return one value for each group of rows specified in a SELECT statement.
- Will cause a run-time error when used in SELECT statements that return zero rows or one row.

Score 1 of 1

Question:

Review the following series of SQL statements:

```
CREATE TABLE SUPPLIES_01
(  SUPPLY_ID NUMBER(7),
   SUPPLIER  VARCHAR2(30),
   ACCT_NO   VARCHAR2(50));
CREATE INDEX IX_SU_01 ON SUPPLIES_01(ACCT_NO);
DROP TABLE SUPPLIES_01;
CREATE TABLE SUPPLIES_02
(  SUPPLY_ID NUMBER(7),
   SUPPLIER  VARCHAR2(30),
   ACCT_NO   VARCHAR2(50));
CREATE INDEX IX_SU_02 ON SUPPLIES_02(ACCT_NO,SUPPLIER);
```

Assuming there are no objects already in existence named SUPPLIES_01 or SUPPLIES_02 prior to the execution of the preceding statements, what database objects will result from these statements?

Response:

A table called SUPPLIES_02 and two indexes called IX_SU_01 and IX_SU_02

✓ A table called SUPPLIES_02 and an index called IX_SU_02

None of the above

A table called SUPPLIES_02 and nothing else

Score 0 of 1

Question:

A correlated subquery:

Response:

- ✗ All of the above
- May be used in a SELECT but not an UPDATE
- ✓ Cannot be executed as a standalone query
- Must use a table alias when referencing a column in the outer query

Score 0 of 1

Question:

Which query returns an expression of the data type INTERVAL YEAR TO MONTHS representing an interval of 1 year and 3 months?

Response:

- SELECT TO_YMINTERVAL('01:03') FROM DUAL;
- ✗ SELECT TO_INTERVALYM('01-03') FROM DUAL;
- SELECT TO_INTERVALYM('01:03') FROM DUAL;

✓ SELECT TO_YMINTERVAL('01-03') FROM DUAL;

Score 1 of 1

Question:

Examine the following two claims:

[1] The DBA_TAB_PRIVS data dictionary view allows a user account to see object privileges it has granted to other user accounts.
[2] The DBA_TAB_PRIVS data dictionary view allows a user account to see object privileges granted by other user accounts to itself.

Which of these claims is true?

Response:

Only 1

✓ Both 1 and 2

Only 2

Neither 1 nor 2

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?

A)

Exhibit

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

B)

Exhibit

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

C)

Exhibit

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

D)

Exhibit

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

Response:

Option B

Option D

Option A

✔ Option C