Question:

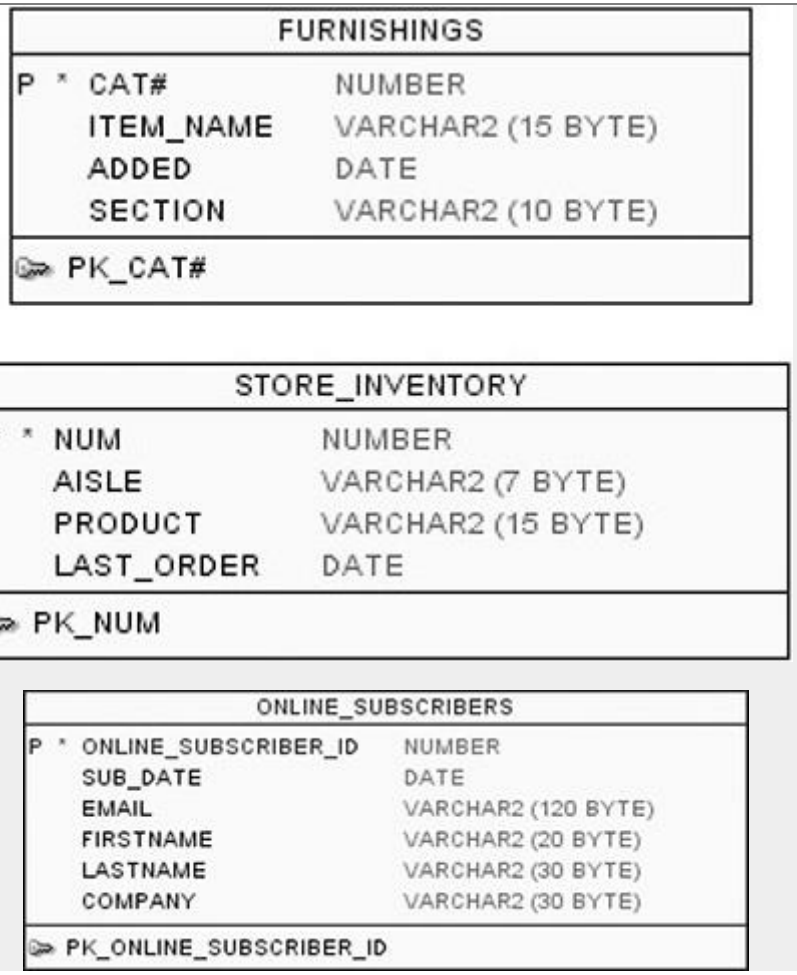
Review the first two illustrations as well as the ONLINE_SUBSCRIBERS table and then review this SQL code:

```
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 LAST_ORDER FROM STORE_INVENTORY
02            UNION
03            SELECT ADDED "Date Added" FROM FURNISHINGS)
04   FROM     ONLINE_SUBSCRIBERS
05   ORDER BY 1;
```

What will happen when this SQL statement is executed?

Response:

It will execute, but the UNION will not work as expected.

It will execute and display one column under the "LAST_ORDER" heading.

✓ It will fail with an execution error on line 1.

It will execute and display one column under the "Date Added" heading.

Score 1 of 1

Question:

Which statement correctly grants a system privilege?

Response:

```
GRANT CREATE SESSION
TO ALL;


GRANT ALTER TABLE
TO PUBLIC;


GRANT CREATE VIEW
ON table1 TO
user1;


GRANT CREATE TABLE
TO user1, user2;
```

Score 1 of 1

Question:

The first DROP operation is performed on PRODUCTS table using the following command:

DROP TABLE products PURGE;

Then you performed the FLASHBACK operation by using the following command:

FLASHBACK TABLE products TO BEFORE DROP;

Which statement describes the outcome of the FLASHBACK command?

Response:

It recovers only the table structure.

✓ It is not possible to recover the table structure, data, or the related indexes.

It recovers the table structure, data, and the indexes.

It recovers the table structure and data but not the related indexes.

Score 1 of 1

Question:

Which statement is true about Data Manipulation Language (DML)?

Response:

Each DML statement forms a transaction by default.

DML automatically disables foreign ley constraints when modifying primary key values in the parent table.

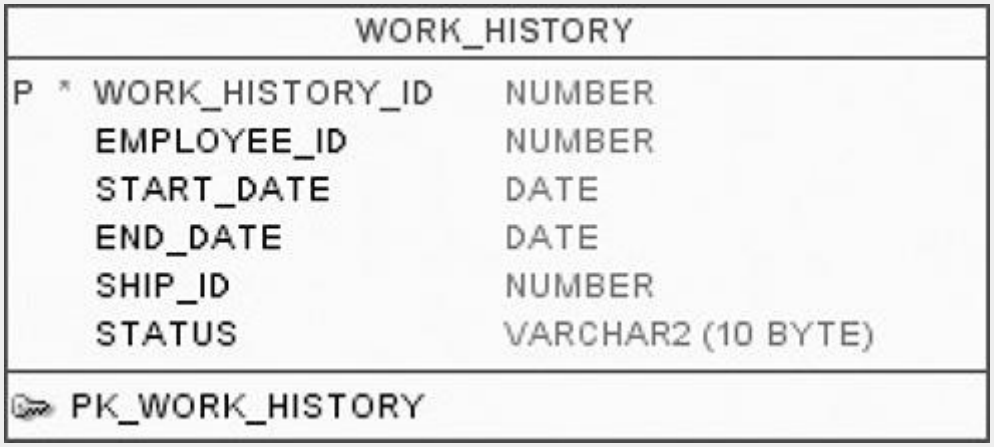
DML disables foreign key constraints when deleting primary key values in the parent table, only when the ON DELETE CASCADE option is set for the foreign key constraint.

✓ A transaction can consist of one or more DML statements.

Score 1 of 1

Question:

Review this WORK_HISTORY table.



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

```
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);
```

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

2

✓ 1

6

Score 1 of 1

Question:

Which three statements are true about multiple-row subqueries?

Response:

✓ They can contain a subquery within a subquery.

✓ They can contain group functions and GROUP BY and HAVING clauses.

They can contain group functions and the GROUP BY clause, but not the HAVING clause.

They cannot contain a subquery within a subquery

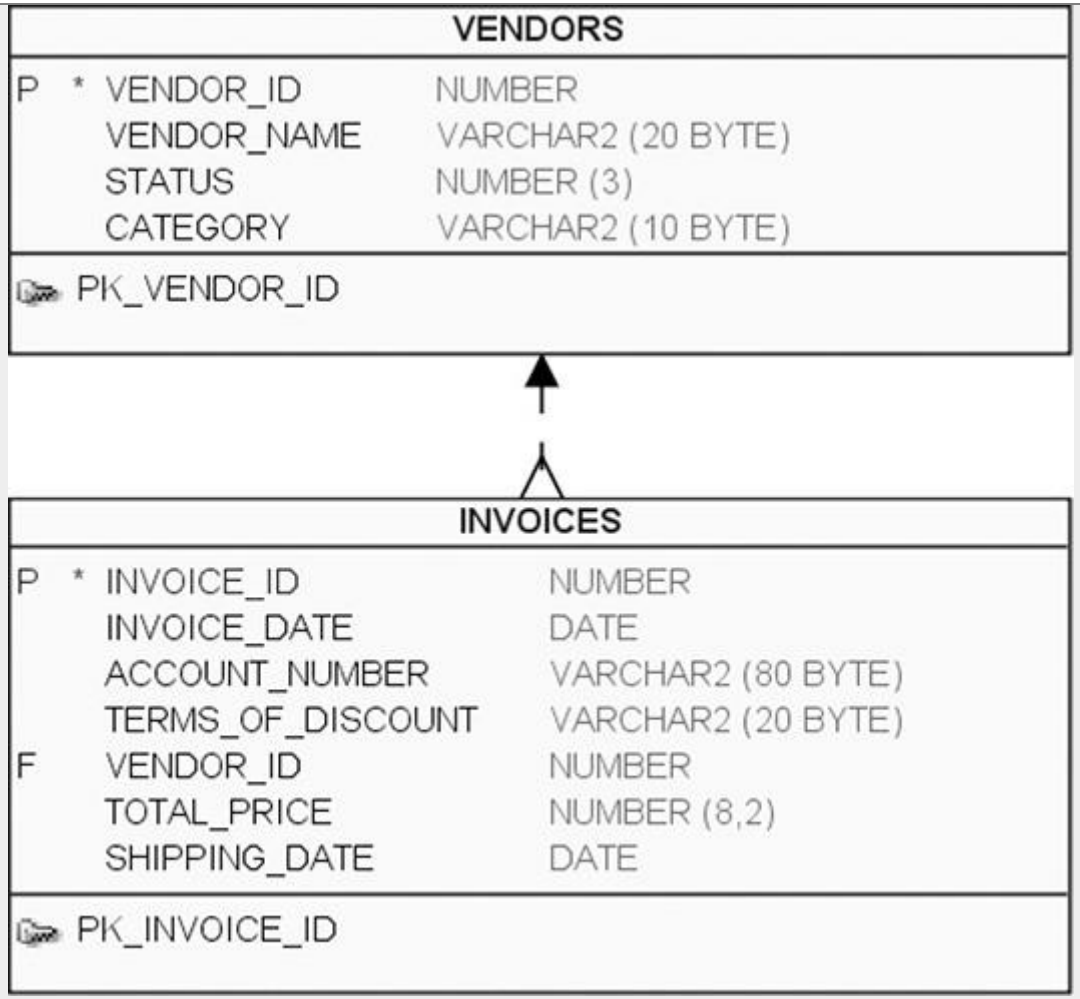
They can return only one column but multiple rows.

They can return multiple columns as well as rows.

Score 1 of 1

Question:

Review the illustration and then review the following SQL statement:




```
01  SELECT VENDOR_ID, INVOICE_DATE, TOTAL_PRICE
02  FROM   VENDORS JOIN INVOICES
03  USING (VENDOR_ID);
```

What kind of join is this?
(Choose two.)

Response:

- NATURAL
- OUTER
- ✓ INNER
- ✓ Equijoin

Score 0 of 1

Question:

Which statements are true?

(Choose all that apply.)

Response:

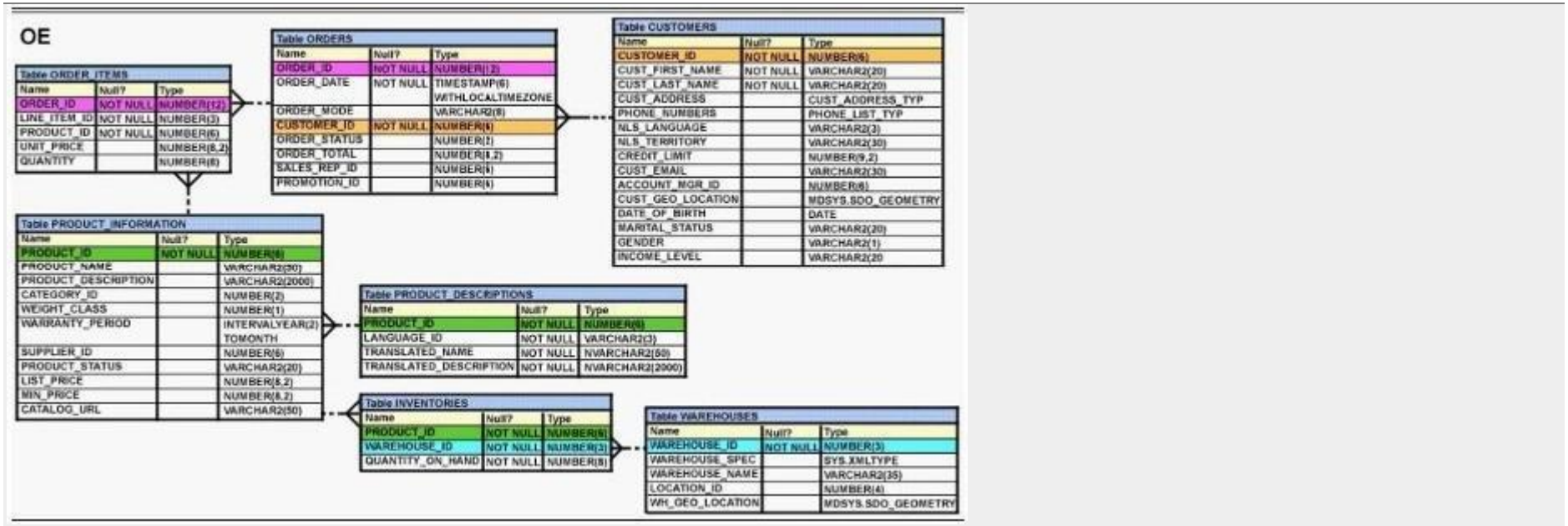
- ✗ The data dictionary views can consist of joins of dictionary base tables and user-defined tables.
- ✓ Views with the same name but different prefixes, such as DBA, ALL and USER, use the same base tables from the data dictionary

- The usernames of all the users including the database administrators are stored in the data dictionary.
- The USER_CONS_COLUMNS view should be queried to find the names of the columns to which a constraint applies.
- Both USER_OBJECTS and CAT views provide the same information about all the objects that are owned by the user.
- The data dictionary is created and maintained by the database administrator.

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?



Response:

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


SELECT COUNT (DISTINCT list_price)
FROM product_information
WHERE list_price is 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 1 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_YMINTERVAL('01-03') FROM DUAL;
```

```
SELECT TO_INTERVALYM('01:03') FROM DUAL;
```

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?

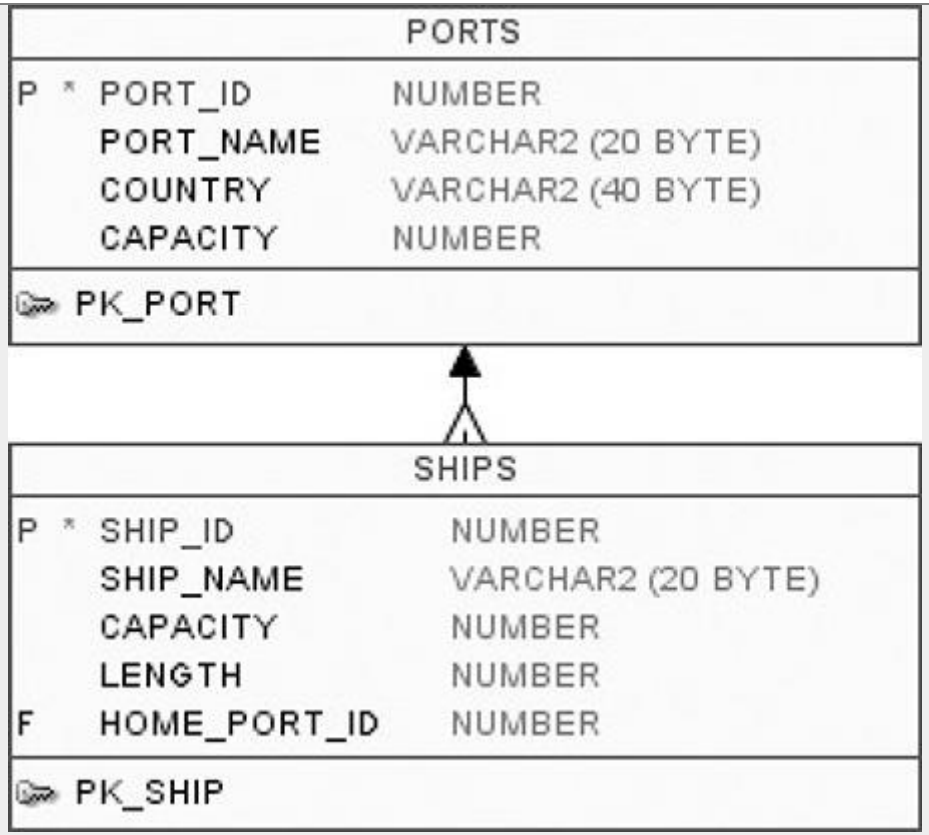
Response:

- It executes successfully and gives the correct result.
- ✔ It executes successfully but does not give the correct result.
- It generates an error because a subquery cannot have a join condition in a UPDATE statement.
- It generates an error because multiple columns (SALARY, COMMISSION) cannot be specified together in an UPDATE statement.

Score 0 of 1

Question:

Review the PORTS and SHIPS tables:



Next, review the following SQL code:

```
01   SELECT P.COUNTRY, P.CAPACITY
02   FROM   PORTS P
03   WHERE  P.PORT_ID > (SELECT S.HOME_PORT_ID
04                       FROM SHIPS S WHERE S.LENGTH > 900);
```

You know that there are five rows in the SHIPS table with a length greater than 900. What will result from an attempt to execute this SQL statement?

Response:

An execution error will result because the subquery will return more than one row and the parent query is expecting only one row from the subquery.

None of the above.

A syntax error will result because PORT_ID and HOME_PORT_ID in line 3 have different column names.

The statement will execute and produce output as intended.

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 2

Both 1 and 2

Only 1

Neither 1 nor 2

Score 0 of 1

Question:

Examine the SQL syntax in question 8. Which of the following two alternatives for line 3 are syntactically correct?

```
01   MERGE INTO SHIP_INVENTORY A
02   USING PORT_INVENTORY B
03   ON (A.NUM = B.NUM)
04   WHEN NOT MATCHED THEN INSERT
05      (A.NUM, A.AISLE, A.PRODUCT, A.LAST_ORDER)
06      VALUES
07      (B.NUM, B.AISLE, B.PRODUCT, B.LAST_ORDER)
08   WHERE TO_CHAR(A.LAST_ORDER,'RRRR') = '2019';
```

```
OPTION 1:  ON (A.NUM = B.NUM AND A.AISLE = B.AISLE)
OPTION 2:  ON (A.LAST_ORDER < B.LAST_ORDER)
```

Response:

Neither option 1 nor option 2

✗ Only option 1

Only option 2

Both option 1 and option 2

Score 1 of 1

Question:

You are designing the structure of a table in which two columns have the specifications:

COMPONENT_ID - must be able to contain a maximum of 12 alphanumeric characters and uniquely identify the row
EXECUTION_DATETIME - contains Century, Year, Month, Day, Hour, Minute, Second to the maximum precision and is used for calculations and comparisons between components.

Which two options define the data types that satisfy these requirements most efficiently?

Response:

The COMPONENT_ID must be of VARCHAR2 data type.

The EXECUTION_DATETIME must be of DATE data type.

The COMPONENT_ID column must be of CHAR data type.

The EXECUTION_DATETIME must be of TIMESTAMP data type.

The EXECUTION_DATETIME must be of INTERVAL DAY TO SECOND data type.

The COMPONENT_ID must be of ROWID data type.

Score 1 of 1

Question:

Assume a schema with only two tables: one named PRODUCTS and one named ENGINEERING. Review the following SQL statements:

```
01  SELECT PRODUCT_ID FROM PRODUCTS;
02  DROP TABLE SHIP_STAFF;
03  INSERT INTO ENGINEERING (PROJECT_ID, MGR) VALUES (27,21);
04  COMMIT;
05  INSERT INTO ENGINEERING (PROJECT_ID, MGR) VALUES (400,17);
06  ROLLBACK;
```

In this series of SQL statements, which line represents the first commit event?

Response:

Line 1

✓ Line 2

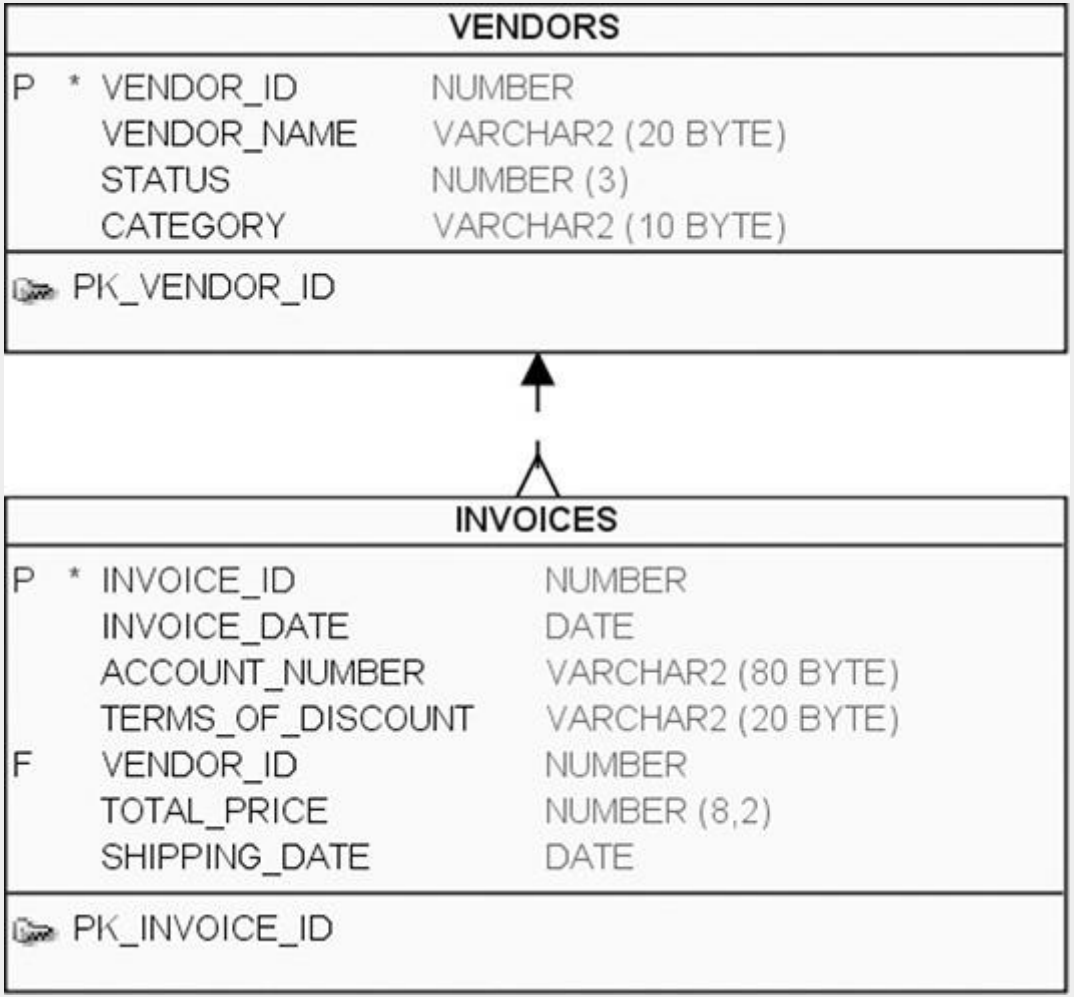
Line 6

Line 4

Score 1 of 1

Question:

Review the illustration. Which of the following is a syntactically correct outer join query?
(Choose two.)



Response:

```
SELECT VENDOR_NAME, INVOICE_DATE
FROM   VENDORS LEFT JOIN  INVOICES
ON     VENDORS.VENDOR_ID = INVOICES.VENDOR_ID;
```

```
SELECT VENDOR_NAME, INVOICE_DATE
FROM   VENDORS RIGHT OUTER JOIN  INVOICES
ON     VENDORS.VENDOR_ID = INVOICES.VENDOR_ID;
```

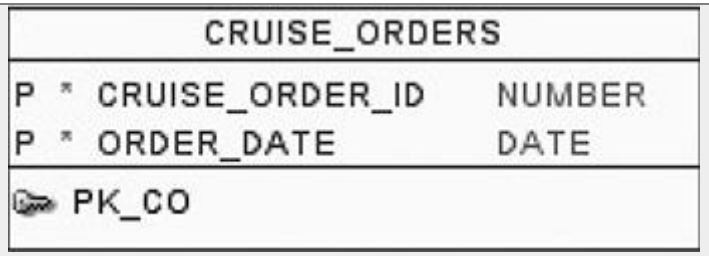
```
SELECT VENDOR_NAME, INVOICE_DATE
FROM   VENDORS OUTER JOIN  INVOICES
ON     VENDORS.VENDOR_ID = INVOICES.VENDOR_ID;
```

```
SELECT VENDOR_NAME, INVOICE_DATE
FROM   VENDORS FULL OUTER  INVOICES
ON     VENDORS.VENDOR_ID = INVOICES.VENDOR_ID;
```

Score 1 of 1

Question:

Review the illustration and then review the following SQL statement:




```
SELECT AVG(CRUISE_ORDER_ID), MIN(ORDER_DATE)
FROM   CRUISE_ORDERS;
```

What will result from an attempt to execute this SQL statement on the CRUISE_ORDERS table?

Response:

It will fail with an execution error if the table contains only one row.

It will fail with an execution error because you cannot use the AVG function on a PRIMARY KEY column.

✓ It will execute and perform as intended.

It will fail with an execution error because you cannot use the MIN function on a DATE data type.

Score 1 of 1

Question:

To list all the currently defined variables, use:

Response:

- ✓ DEFINE
- DEFINE ALL
- SHOW ALL
- SHOW DEFINE

Score 1 of 1

Question:

View 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)

Which two tasks would require subqueries or joins to be executed in a single statement?
(Choose two.)

Response:

- ✓ listing of those customers, whose credit limit is the same as the credit limit of customers residing in the city 'Tokyo'.
- finding the number of customers, in each city, who’s marital status is 'married'.
- ✓ finding the number of customers, in each city, whose credit limit is more than the average credit limit of all the customers
- finding the average credit limit of male customers residing in 'Tokyo' or 'Sydney'
- listing of customers who do not have a credit limit and were born before 1980

Score 1 of 1

Question:

Which three statements are true reading subquenes?

Response:

- ✓ The subquery and main query can retrieve data from different tables.
- Only one column or expression can be compared between the subquery and main query.

The subquery and main query must retrieve date from the same table.

✓ Multiple columns or expressions can be compared between the subquery and main query.

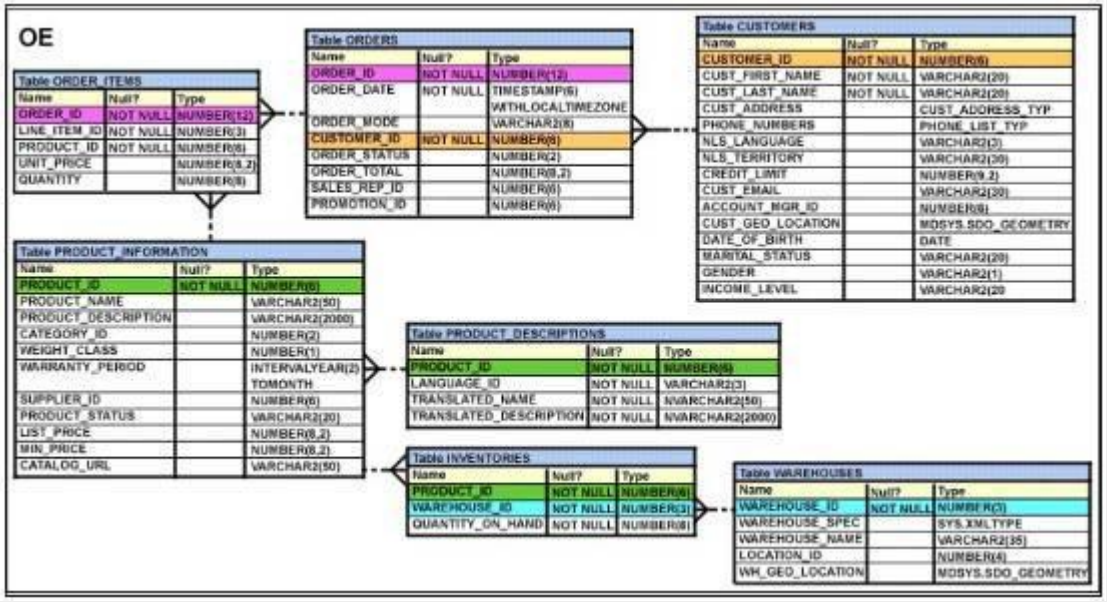
✓ A Main query can have many subqueries.

A subquery can have more than one main query

Score 1 of 1

Question:

View the Exhibit and examine the data in ORDERS and ORDER_ITEMS tables. You need to create a view that displays the ORDER ID, ORDER_DATE, and the total number of items in each order.



Which CREATE VIEW statement would create the view successfully?

Response:

CREATE OR REPLACE VIEW ord_vu
AS SELECT o.order_id, o.order_date, COUNT(i.line_item_id) FROM orders o
JOIN order_items i ON (o.order_id = i.order_id) GROUP BY
o.order_id,o.order_date;

✓ CREATE OR REPLACE VIEW ord_vu
AS SELECT o.order_id, o.order_date, COUNT(i.line_item_id) "NO OF ITEMS"
FROM orders o JOIN order_items i ON (o.order_id = i.order_id)
GROUP BY o.order_id,o.order_date;

CREATE OR REPLACE VIEW ord_vu (order_id,order_date) AS SELECT

```
o.order_id, o.order_date, COUNT(i.line_item_id) "NO OF ITEMS"
FROM orders o JOIN order_items i ON (o.order_id = i.order_id)
GROUP BY o.order_id,o.order_date;

CREATE OR REPLACE VIEW ord_vu
AS SELECT o.order_id, o.order_date, COUNT(i.line_item_id)ll' NO OF ITEMS'
FROM orders o JOIN order_items i
ON (o.order_id = i.order_id) GROUP BY o.order_id,o.order_date WITH CHECK
OPTION;
```

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 not fire any DML triggers on the table.

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

✓ The statement will fail.

The statement will result in an implicit commit.

Score 1 of 1

Question:

Review the SQL statement in the preceding question. If one of the INTO clauses executed on a table and resulted in a constraint violation on that table, what would result?

Response:

None of the above.

✓ The row would not be inserted, the INSERT statement would stop, and all rows affected by the INSERT statement would be rolled back, as if the INSERT statement had never been executed.

The row would not be inserted, and the INSERT statement would skip to the next row returned by the subquery and perform another pass through the WHEN conditions.

The row would not be inserted, and the INSERT statement would stop. No additional rows would be returned by the subquery or processed, but rows that have already been processed are unaffected.

Score 1 of 1

Question:

Which three statements are true regarding the WHERE and HAVING clauses in a SQL statement?
(Choose three.)

Response:

✓ The HAVING clause is used to exclude one or more aggregated results after grouping data.

✓ The HAVING clause conditions can have aggregate functions.

The HAVING clause conditions can use aliases for the columns.

✓ The WHERE clause is used to exclude rows before the grouping of data.

WHERE and HAVING clauses cannot be used together in a SQL statement.

Score 1 of 1

Question:

Which of the following problems can be solved with a subquery?
(Choose the two best answers.)

Response:

You are tasked with determining the minimum sales for every division in a multinational corporation.

✓ You are tasked with determining which divisions in a corporation earned sales last year that were less than the average sales for all divisions in the prior year.

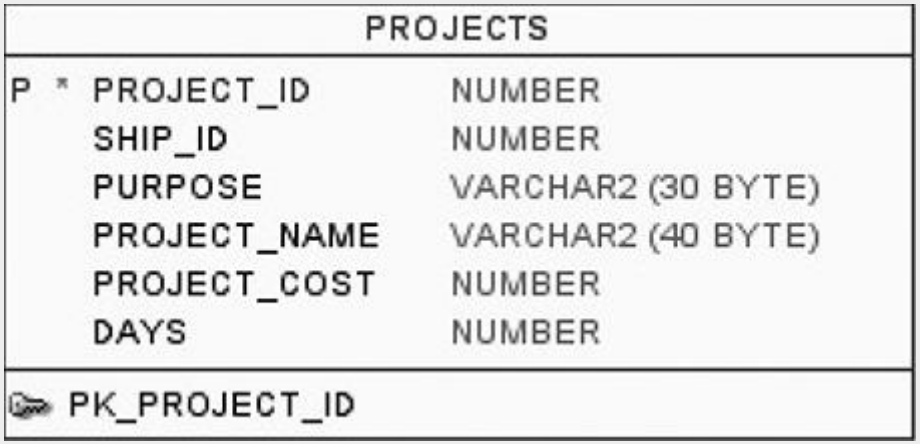
✓ You are tasked with creating a view.

You are tasked with creating a sequence.

Score 1 of 1

Question:

Review the illustration and review the SQL statement that follows:



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		

```
01    SELECT   SHIP_ID, MAX(DAYS)
02    FROM     PROJECTS
03    GROUP BY SHIP_ID
04    HAVING   AVG(PROJECT_COST) < 500000;
```

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

Response:

- ✓ It will include only those groups of rows for a given SHIP_ID with an average value of PROJECT_COST less than 500000.

- It will fail to execute because of a syntax error on line 1.

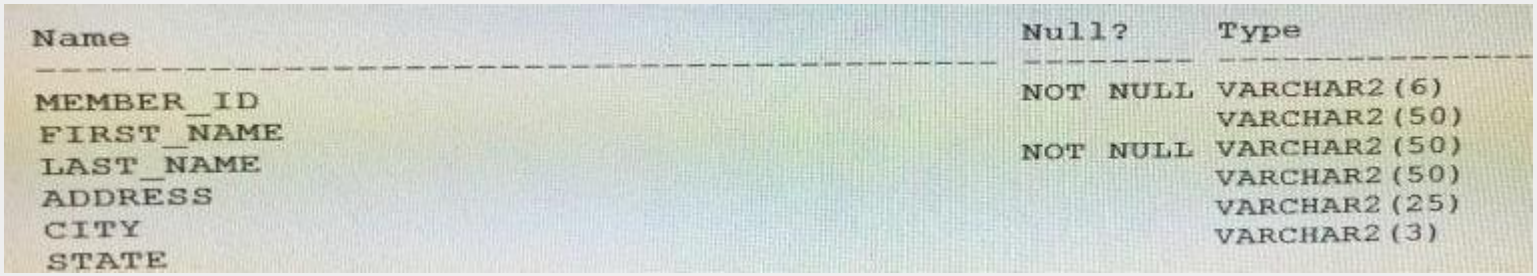
It will include only those rows with a PROJECT_COST value of less than 500000.

It will fail to execute because of a syntax error on line 4.

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		VARCHAR2(3)

You want to display details of all members who reside in states starting with the letter A followed by exactly one character. Which SQL statement must you execute?

Response:

SELECT * FROM MEMBERS WHERE state LIKE 'A%';

SELECT * FROM MEMBERS WHERE state LIKE '%A_' ;

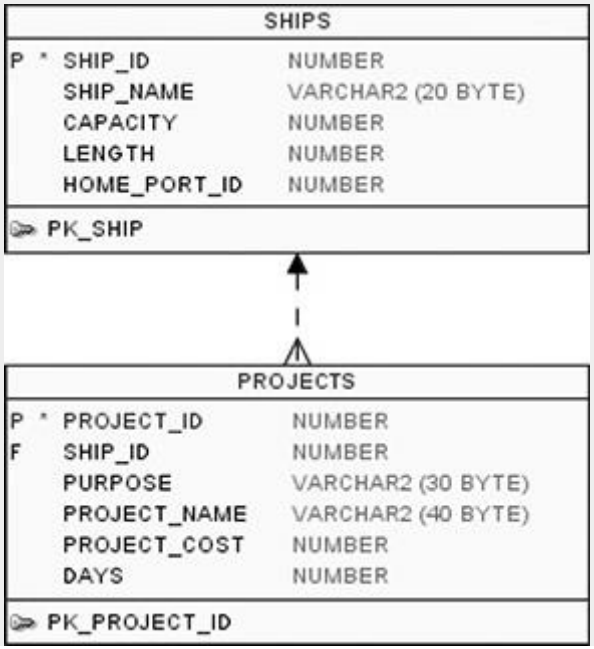
✔ SELECT * FROM MEMBERS WHERE state LIKE 'A_';

```
SELECT * FROM MEMBERS WHERE state LIKE 'A_%';
```

Score 1 of 1

Question:

Review the illustration and the following SQL code:



```
01  CREATE OR REPLACE VIEW MAJOR_PROJECTS AS
02    SELECT PROJECT_ID, SHIP_ID, PROJECT_NAME, PROJECT_COST
03    FROM   PROJECTS
04    WHERE  PROJECT_COST > 10000;
05
06  INSERT INTO MAJOR_PROJECTS
07   (PROJECT_ID, SHIP_ID, PROJECT_NAME, PROJECT_COST)
08   VALUES
09   ((SELECT MAX(PROJECT_ID)+1 FROM PROJECTS),
10    (SELECT MAX(SHIP_ID) FROM SHIPS),
11    'Small Project',
12    500);
```

What will result from an attempt to execute these two SQL statements?

Response:

The INSERT statement will fail because the PROJECT_COST value being inserted is not consistent with the WHERE clause on line 4.

✓ The CREATE and INSERT statements will successfully execute.

The CREATE statement will fail because it omits the PURPOSE column from the PROJECTS table.

The INSERT statement will fail because of an error on lines 9 and 10.

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

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

Score 0 of 1

Question:

What can you use to submit SQL statements for execution?
(Choose all that apply.)

Response:

- ✓ SQL Developer
- ✓ PHP
- ✓ SQL*Plus
- ✓ Java

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		

```
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.

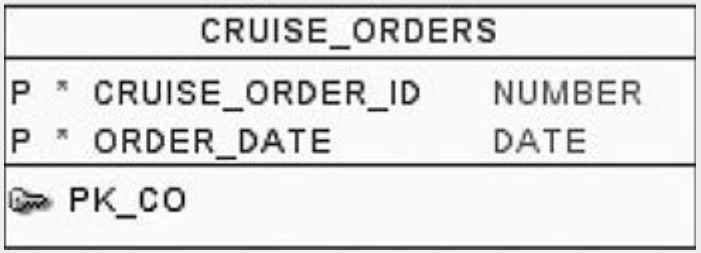
6

✔ 4

Score 1 of 1

Question:

Review the following illustration:



Now review this SQL statement:

```
SELECT CRUISE_ORDER_ID, COUNT(ORDER_DATE)
FROM   CRUISE_ORDERS;
```

What can be said of this statement?

Response:

It will execute successfully but not produce any meaningful output.

It will fail to execute because ORDER_DATE is a date data type, and no aggregate function can work with a date data type.

There is nothing wrong with the SQL statement.

✓ It will fail to execute because it mixes scalar and aggregate data in the select list.

Score 1 of 1

Question:

You execute the following commands:

```
SQL > DEFINE hiredate = '01-APR-2011'

SQL >SELECT employee_id, first_name, salary
FROM employees
WHERE hire_date > '&hiredate'
AND manager_id > &mgr_id;
```

For which substitution variables are you prompted for the input?

Response:

both the substitution variables ''hiredate' and 'mgr_id'.

✓ only 'mgr_id'

none, because no input required

```
only hiredate'
```