



Oracle Database SQL (1Z0-071) - Full

You got **59** of **71** possible points.

Your score: **83 %**

Question Results

Question:

Score 1 of 1

Which statement is true about transactions?

Response:

A combination of DDL and DML statements executed in a sequence ending with a COMMIT forms a single transaction.



Each Data Definition Language (DDL) statement executed forms a single transaction.

A set of DDL statements executed in a sequence ending with a COMMIT forms a single transaction.

A set of Data Manipulation Language (DML) statements executed in a sequence ending with a SAVEPOINT forms a single transaction.

Question:

Score 1 of 1

The database object that stores lookup information to speed up querying in tables is:

Response:

VIEW

ROWID

LOOKUP



INDEX

Question:

Score 0 of 1

Examine the structure of the employees table.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8, 2)
COMMISSION_PCT		NUMBER(2, 2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

There is a parent/child relationship between EMPLOYEE_ID and MANAGER_ID.**You want to display the last names and manager IDs of employees who work for the same manager as the employee whose EMPLOYEE_ID is 123. Which query provides the correct output?**

A)

Exhibit

```
SELECT e.last_name, m.manager_id
FROM employees e RIGHT OUTER JOIN employees m
ON (e.manager_id = m.employee_id)
AND e.employee_id = 123;
```

B)

Exhibit

```
SELECT e.last_name, m.manager_id
FROM employees e LEFT OUTER JOIN employees m
ON (e.employee_id = m.manager_id)
WHERE e.employee_id = 123;
```

C)

Exhibit

```
SELECT e.last_name, e.manager_id
FROM employees e RIGHT OUTER JOIN employees m
ON (e.employee_id = m.employee_id)
WHERE e.employee_id = 123;
```

D)

Exhibit

```
SELECT m.last_name, e.manager_id
FROM employees e LEFT OUTER JOIN employees m
ON (e.manager_id = m.manager_id)
WHERE e.employee_id = 123;
```

Response:



Option C



Option D

Option B

Option A

Question:

Score 0 of 1

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:**

Performance would improve in query 2.



Performance would degrade in query 2.



There would be no change in performance.

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

Question:

Score 1 of 1

Which of the following can be said of the CASE statement?**Response:**

It converts text to uppercase.



It uses the keyword THEN.

Its END keyword is optional.

It uses the keyword IF.

Question:

Score 0 of 1

The best exam guide you could possibly get for preparing to take and pass the 1Z0-071 certification exam, SQL Associate, is which of the following?

(Choose all that apply.)

Response:



The book you are holding right now.



Don't make me tell you again.



This book.



This here book.

Question:

Score 1 of 1

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
GROUP BY PURPOSE, (DAYS > 3);
```

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

```
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;

Question:

Score 1 of 1

Review the following data listing for a table called SHIP_CABINS:

ROOM_NUMBER	STYLE	WINDOW
102	Suite	Ocean
103		Ocean
104		

The blank values are NULL. Now review the following SQL statement (line numbers are added for readability):

```
01  SELECT ROOM_NUMBER
02  FROM SHIP_CABINS
03  WHERE (STYLE = NULL) OR (WINDOW = NULL);
```

How many rows will the SQL statement retrieve?**Response:**

None because you cannot use parentheses in line 3 to surround the expressions

1

2

 0**Question:**

Score 1 of 1

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 an index called IX_SU_02

A table called SUPPLIES_02 and nothing else

A table called SUPPLIES_02 and two indexes called IX_SU_01 and IX_SU_02

None of the above

Question:**Score 1 of 1**

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.



A pending transaction in the session is committed.

All indexes on the table remain but they are invalidated.



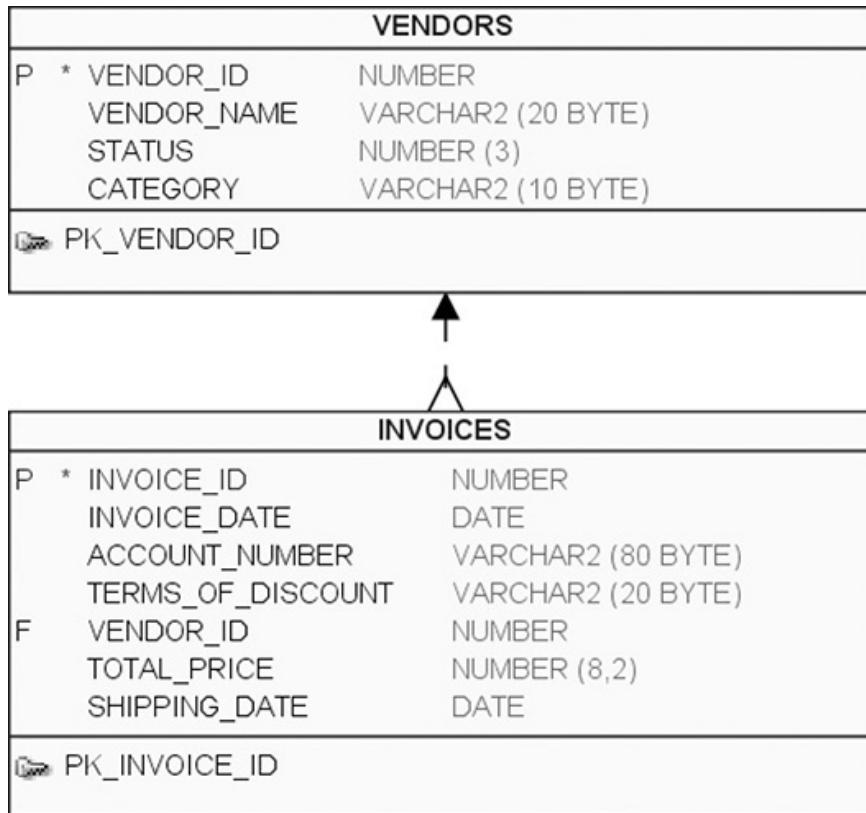
All views and synonyms on the table remain but they are invalidated.



All data along with the table structure is deleted.

Question:

Score 1 of 1

Review the INVOICES and VENDORS tables.**Next review the following SQL statement:**

```

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

```

Which of the following statements is true for the SQL statement?**Response:**

It will fail with a syntax error on line 3 because of the parentheses around VENDOR_ID.



It will execute successfully.

It will fail with a syntax error because there is no ON clause.

It will fail with a syntax error on line 1 because VENDOR_ID is ambiguous.

Question:

Score 1 of 1

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_%';
```

Question:

Score 1 of 1

Which of the following can a subquery be used in?

(Choose all that apply.)

Response:



```
A WHERE clause in a SELECT statement
```

```
A GRANT statement
```



```
An inline view
```



```
An INSERT statement's SELECT
```

Question:

Score 1 of 1

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) <= 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);

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);

Question:

Score 1 of 1

You attempt to execute the following SQL statement:

```
CREATE TABLE VENDORS
  (VENDOR_ID      NUMBER,
   VENDOR_NAME    VARCHAR2,
   CATEGORY       CHAR);
```

Which one of the following is true?**Response:**

The execution succeeds, and the table is created.



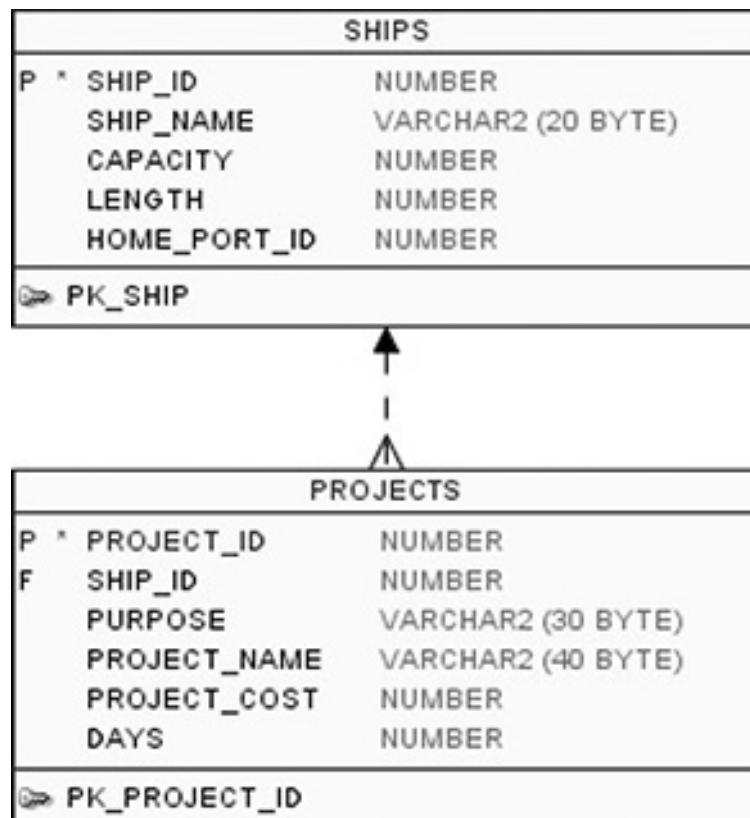
The execution fails because there is no precision indicated for VARCHAR2.

The execution fails because there is no precision indicated for NUMBER.

The execution fails because there is no precision indicated for CHAR.

Question:

Score 0 of 1

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 is a join.

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.

Question:

Score 0 of 1

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:**

Performance would degrade in query 2.

Performance would improve query 2 only if there are null values in the CUST CREDIT LIMIT column.

Performance would improve in query 2.



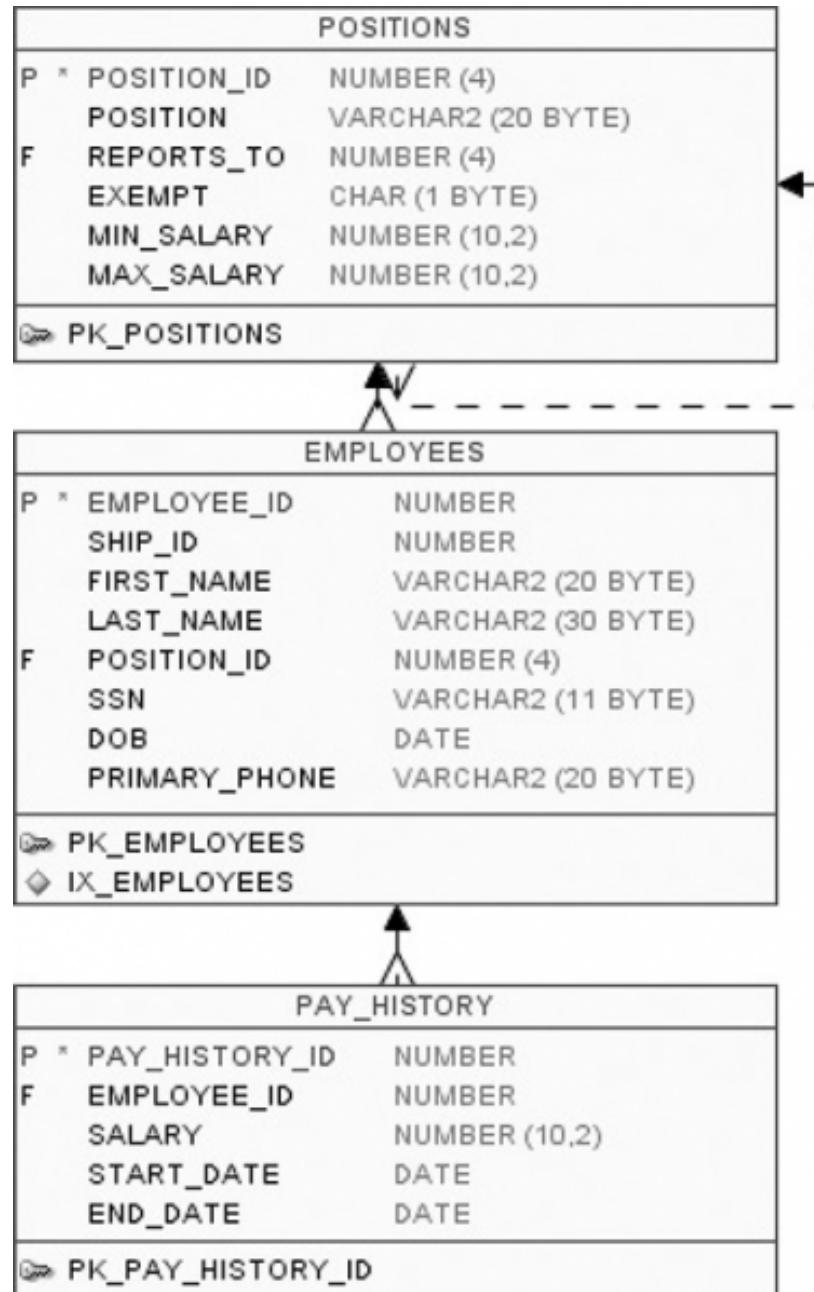
There would be no change in performance.

Question:

Score 1 of 1

Review the illustration. Which of the following is a valid self-join statement?

(Choose all that apply.)

**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 INNER 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;
```

Question:

Score 1 of 1

Which of the following is true about ROLES?**Response:**

Roles are in the same namespace as TABLES.

Roles are schema objects but only when created from within a user account.



Roles are in the same namespace as USERS.

Roles are in the same namespace as CONSTRAINTS.

Question:

Score 1 of 1

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

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



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

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

Question:

Score 1 of 1

Evaluate the following SQL statements that are issued in the given order:

```
CREATE TABLE emp
(emp_no NUMBER(2) CONSTRAINT emp_emp_no_pk PRIMARY KEY,
ename VARCHAR2(15),
salary NUMBER (8,2),
mgr_no NUMBER(2) CONSTRAINT emp_mgr_fk REFERENCES emp(emp_no));
ALTER TABLE emp
DISABLE CONSTRAINT emp_emp_no_pk CASCADE;
ALTER TABLE emp
ENABLE CONSTRAINT emp_emp_no_pk;
```

What would be the status of the foreign key EMP_MGR_FK?

Response:

It would remain disabled and can be enabled only by dropping the foreign key constraint and recreating it.

It would be automatically enabled and deferred.

It would be automatically enabled and immediate.



It would remain disabled and has to be enabled manually using the ALTER TABLE command.

Question:

Score 0 of 1

Review this SQL statement: SELECT LASTNAME FROM CUSTOMERS WHERE LASTNAME = SOUNDEX('Franklin'); What is a possible result for the query?

Response:

Phrankline



Franklyn

Ellison



None of the above

Question:

Score 1 of 1

Now you have changed the purpose of the PIER column in the MARINA table and want to remove the comment you just created in the previous question. Which of the following statements will remove the comment?

Response:

COMMENT ON COLUMN MARINA.PIER IS NULL;



COMMENT ON COLUMN MARINA.PIER IS '';

COMMENT ON COLUMN MARINA.PIER DROP;

COMMENT ON COLUMN MARINA.PIER SET UNUSED;

Question:

Score 1 of 1

Which of the following statements is true about HAVING?

(Choose two.)

Response:

It must occur after the GROUP BY clause.



It must occur after the WHERE clause.

It cannot reference an expression unless that expression is first referenced in the GROUP BY clause.



It can be used only in the SELECT statement.

Question:

Score 1 of 1

The DECODE expression always ends with:**Response:**

Neither of the above

The keyword END

Both of the above



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

Question:

Score 0 of 1

The following are the steps for a correlated subquery, listed in random order:

- 1) The WHERE clause of the outer query is evaluated.
- 2) The candidate row is fetched from the table specified in the outer query.
- 3) The procedure is repeated for the subsequent rows of the table, till all the rows are processed.
- 4) Rows are returned by the inner query, after being evaluated with the value from the candidate row in the outer query.

Identify the option that contains the steps in the correct sequence in which the Oracle server evaluates a correlated subquery.**Response:**

4,1,2,3



2,4,1,3



4,2,1,3

2,1,4,3

Question:

Score 1 of 1

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

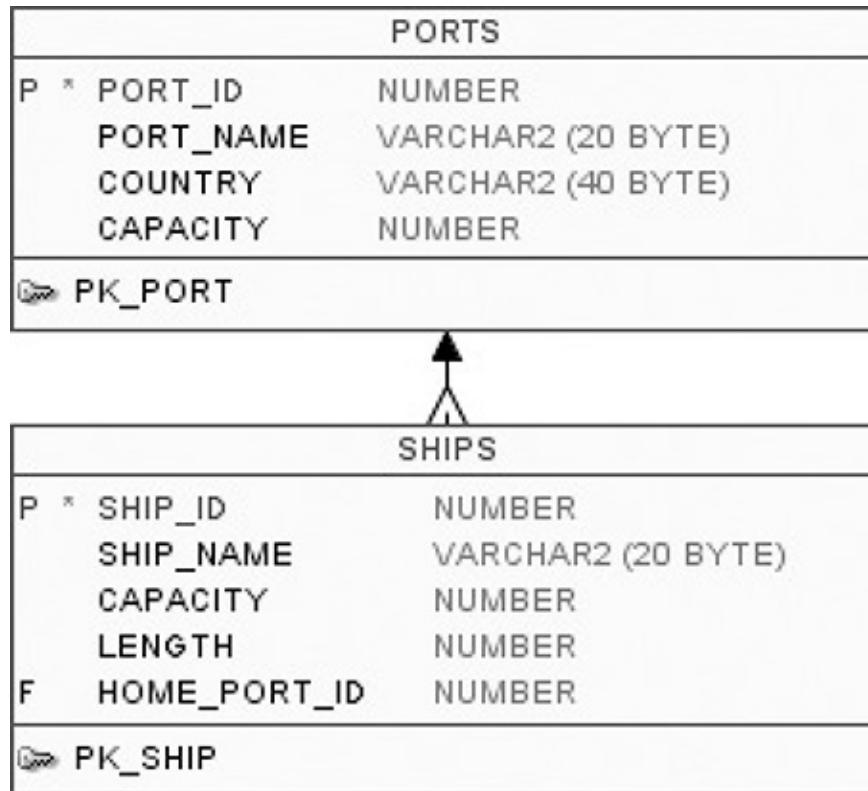


None of the above

Question:

Score 1 of 1

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:

None of the above.



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.

Question:

Score 1 of 1

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:

USER_PRIV_SEQUENCES



ALL_SEQUENCES

DBA_SEQUENCES

USER_SEQUENCES

Question:

Score 1 of 1

Which of the following problems can be solved with a subquery?

(Choose the two best answers.)

Response:

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 sequence.

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



You are tasked with creating a view.

Question:

Score 1 of 1

Your user account owns an updatable view, BACKLOG, which is based on the table PROJECTS. You are tasked to give SELECT and UPDATE capabilities to another user account named MARINO.

Currently, MARINO has no privileges on either the table or the view. You want for MARINO to have the ability to grant SELECT on the view to other users as well.

Examine the following SQL code:

```
GRANT SELECT ON BACKLOG TO MARINO WITH GRANT OPTION;  
GRANT UPDATE ON BACKLOG TO MARINO;
```

Which of the following statements is true?**Response:**

The statements will fail, and MARINO will not be able to use the view.



The statements will execute successfully and perform as intended.

The statements will execute successfully, but MARINO will not be able to SELECT from the view because the PROJECTS table has not been granted to MARINO.

The statements will execute successfully, and MARINO will be able to SELECT from the view but not UPDATE the view.

Question:**Score 1 of 1****Which three statements are true regarding the WHERE and HAVING clauses in a SQL statement?**

(Choose three.)

Response:

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



The HAVING clause conditions can have aggregate functions.



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

The HAVING clause conditions can use aliases for the columns.

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

Question:**Score 0 of 1****Which two statements are true about Data Manipulation Language (DML) statements?****Response:**

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



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



A DELETE FROM..... statement can remove 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.



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

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

Question:

Score 1 of 1

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

Question:

Score 0 of 1

What is one of the purposes of DDL?

(Choose the best answer.)

Response:

None of the above



Remove existing data from a database table

Query data from a given table



Issue privileges to users

Question:

Score 1 of 1

Which two statements are true regarding the GROUP BY clause in a SQL statement?

(Choose two.)

Response:

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

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

You can use column alias in the GROUP BY clause.

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

 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 clause.

Question:

Score 1 of 1

Review the following diagrams of the SPARES table:

SPARES	
SPARE_ID	NUMBER (8)
PART_NO	VARCHAR2 (30 BYTE)
PART_NAME	VARCHAR2 (80 BYTE)
◆ IX_01	

Also examine the diagrams of the tables PORT_INVENTORY, STORE_INVENTORY, and SHIP_INVENTORY, shown here.

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

SHIP_INVENTORY	
P *	NUM NUMBER
	AISLE VARCHAR2 (7 BYTE)
	PRODUCT VARCHAR2 (15 BYTE)
	LAST_ORDER DATE
PK_	PK_SHIP_INV_NUM

PORT_INVENTORY	
P *	NUM NUMBER
	AISLE VARCHAR2 (7 BYTE)
	PRODUCT VARCHAR2 (15 BYTE)
	LAST_ORDER DATE
PK_	PK_PORT_INV_NUM

Now consider the following SQL statement:

```

01  INSERT ALL
02      WHEN (SUBSTR(PART_NAME,1,4) = 'MED-') THEN
03          INTO STORE_INVENTORY (NUM, AISLE, PRODUCT, LAST_ORDER)
04              VALUES (SPARE_ID, 'Back', PART_NAME, SYSDATE)
05          INTO SHIP_INVENTORY (NUM, AISLE, PRODUCT, LAST_ORDER)
06              VALUES (SPARE_ID, 'Back', PART_NAME, SYSDATE)
07      WHEN (SUBSTR(PART_NAME,1,4) = 'ARR-') THEN
08          INTO PORT_INVENTORY (NUM, AISLE, PRODUCT, LAST_ORDER)
09              VALUES (SPARE_ID, 'Back', PART_NAME, SYSDATE)
10      SELECT SPARE_ID, PART_NO, PART_NAME
11  FROM    SPARES;

```

Regarding this SQL statement, which of the following statements is true?

Response:

The statement will fail because it is missing a WHEN condition.

The statement will add every row returned from the SPARES table to the SHIP_INVENTORY table.

The statement will fail because there is no ELSE clause.



The statement will add a row returned from the SPARES table to the SHIP_INVENTORY table only if the WHEN condition on line 2 evaluates to true.

Question:

Score 0 of 1

Which three statements are true about the ALTER TABLE....DROP COLUMN....command?

Response:

A dropped column can be rolled back.



A parent key column in the table cannot be dropped.



The column in a composite PRIMARY KEY with the CASCADE option can be dropped.



A column can be dropped only if another column exists in the table.

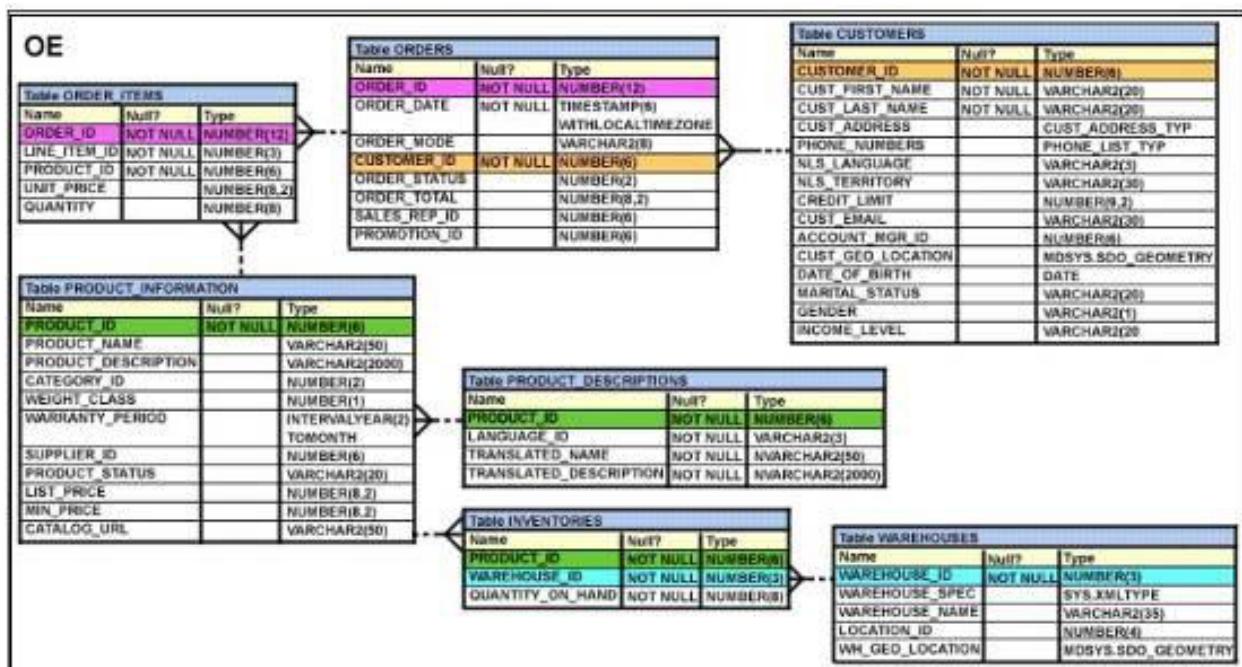
A column can be dropped only if it does not contain any data.

Question:

Score 1 of 1

View the Exhibit and examine the structure of ORDERS and ORDER_ITEMS tables. ORDER ID is the primary key in the ORDERS table.

It is also the foreign key in the ORDER_ITEMS table wherein it is created with the ON DELETE CASCADE option.



Which DELETE statement would execute successfully?

Response:

```
DELETE orders o, order_items i WHERE o.order_id = i.order_id;
```

```
DELETE order_id FROM orders WHERE order_total < 1000;
```

```
DELETE FROM orders WHERE (SELECT order_id FROM order_items);
```



```
DELETE orders WHERE order_total < 1000;
```

Question:

Score 1 of 1

Which of the following SQL statements creates a table that will reject attempts to INSERT a row with NULL values entered into the POSITION_ID column?

Response:

```
CREATE TABLE POSITIONS
(POSITION_ID NUMBER(3),
 CONSTRAINT POSITION_CON REQUIRED (POSITION_ID));
```

```
CREATE TABLE POSITIONS
(POSITION_ID NUMBER(3),
 CONSTRAINT POSITION_CON UNIQUE (POSITION_ID));
```

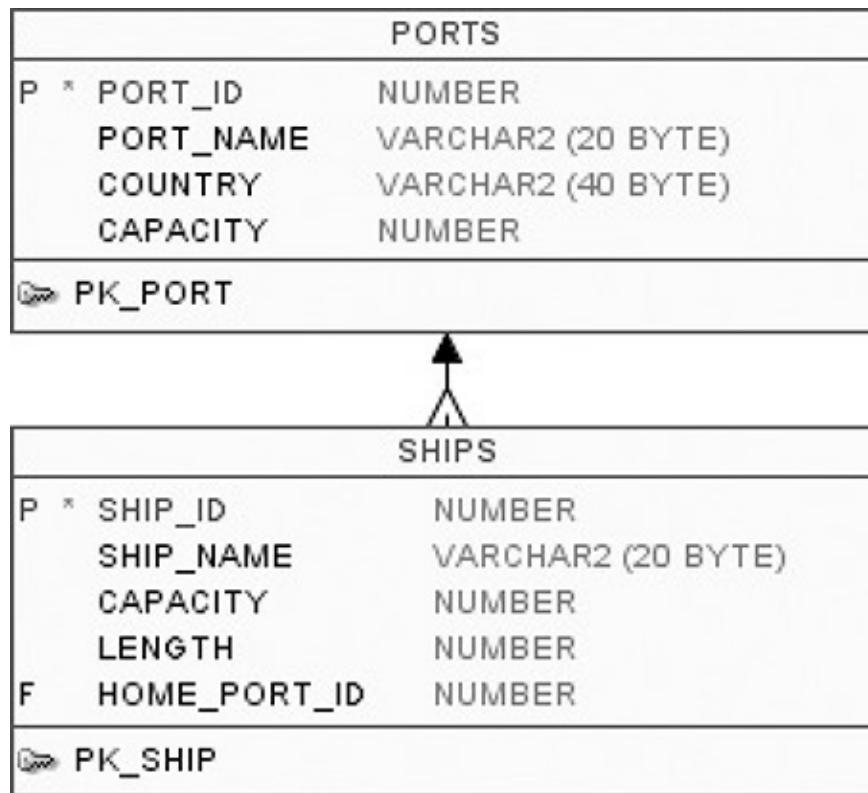
None of the above

CREATE TABLE POSITIONS
(POSITION_ID NUMBER(3),
CONSTRAINT POSITION_CON PRIMARY KEY (POSITION_ID));

Question:

Score 1 of 1

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 increase once for each of the 15 rows in the PORTS table.

The value for CAPACITY will increase by 20 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 not increase.

Question:

Score 1 of 1

Examine the following data listing of a table called PERMITS:

PERMIT_ID	FILED_DATE	VENDOR_ID
1	05-DEC-09	101
2	12-DEC-09	310903
3	14-DEC-09	101

Which one of the following aggregate functions could be used to determine how many permits have been filed by VENDOR_ID 101?

Response:



COUNT

SUM

MEDIAN

HAVING

Question:

Score 1 of 1

If an ALTER TABLE . . . DROP COLUMN statement is executed against an underlying table upon which a view is based, the status of that view in the data dictionary changes to:

Response:



INVALID

ALTERED

FLAG

COMPILE

Question:

Score 1 of 1

Analytic functions are processed:

Response:

As the first set of operations prior to the SELECT column list processing

As the first set of operations before processing the WHERE clause

As the last set of operations before processing the WHERE clause



As the last set of operations before processing the ORDER BY clause

Question:

Score 1 of 1

Which two statements are true regarding the COUNT function?

(Choose two.)

Response:



COUNT(DISTINCT inv_amt) returns the number of rows excluding rows containing duplicates and NULL values in the INV_AMT column

COUNT(cust_id) returns the number of rows including rows with duplicate customer IDs and NULL value in the CUST_ID column

A SELECT statement using COUNT function with a DISTINCT keyword cannot have a WHERE clause

The COUNT function can be used only for CHAR, VARCHAR2 and NUMBER data types



COUNT(*) returns the number of rows including duplicate rows and rows containing NULL value in any of the columns

Question:

Score 0 of 1

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 D



Option C

Option A

Option B

Question:

Score 1 of 1

Review the following illustration:

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

Which of the following SQL statements will execute correctly?**Response:**

```
SELECT RANK(7500000) GROUP BY (ORDER BY PROJECT_COST) FROM PROJECTS;
```

```
SELECT RANK(100,000) WITHIN GROUP (ORDER BY PROJECT_COST) FROM PROJECTS;
```

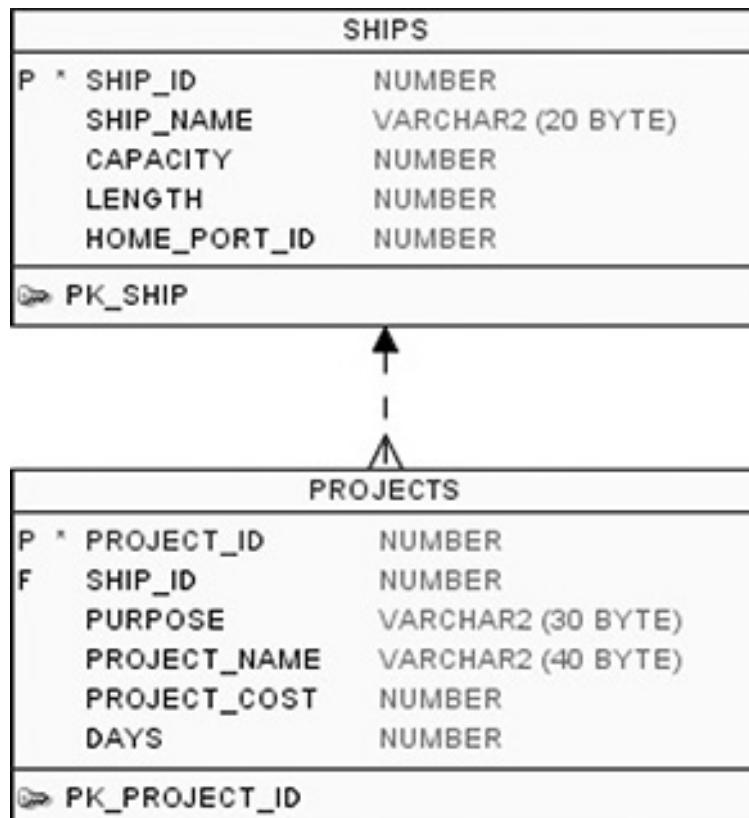
 SELECT RANK(100000) WITHIN GROUP (ORDER BY PROJECT_COST) FROM PROJECTS;

SELECT RANK('Upgrade') WITHIN GROUP (ORDER BY PROJECT_COST) FROM PROJECTS;

Question:

Score 1 of 1

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 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.



The CREATE and INSERT statements will successfully execute.

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

Question:

Score 1 of 1

Review the illustration. Your task is to define a SELECT statement that groups rows according to their value for PURPOSE and, for each purpose, adds up the values stored in DAYS.

Which one of the following queries 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 SUM(DAYS), PURPOSE
FROM PROJECTS
GROUP BY PURPOSE, SUM(DAYS);
```



```
SELECT SUM(DAYS), PURPOSE
FROM PROJECTS
GROUP BY PURPOSE;
```

```
SELECT PURPOSE, COUNT(DAYS)
FROM PROJECTS
GROUP BY PURPOSE;
```

```
SELECT PURPOSE, RANK(DAYS) ON (ORDER BY)
FROM PROJECTS
GROUP BY PURPOSE;
```

Question:

Score 1 of 1

Assume a table LAMPS that has no constraints. Which of the following is true about the UPDATE statement and the LAMPS table?

(Choose all that apply.)

Response:

✓ For existing rows in LAMPS, UPDATE can add values to any column with a NULL value.

✓ For existing rows in LAMPS, UPDATE can remove values from any column by changing its value to NULL.

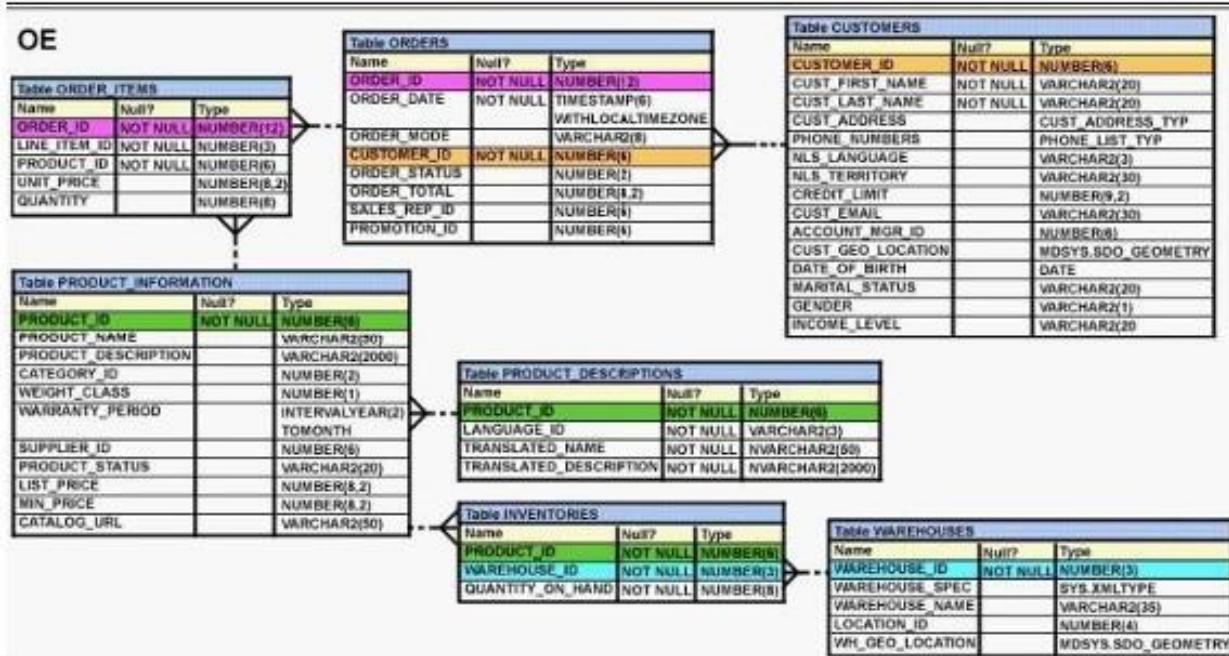
UPDATE can be used to add rows to LAMPS by setting values to all the columns.

UPDATE can be used to remove a row from LAMPS by setting all of the row's columns to a value of NULL.

Question:

Score 1 of 1

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

✓ SELECT COUNT (NVL(list_price, 0))
FROM product_information

WHERE list_price is NULL

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

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

Question:

Score 1 of 1

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:

3

6

 0

1

Question:

Score 1 of 1

Which of the following symbols is most likely to be used in a SELECT statement using a non-equijoin?

Response:

None of the above

 <=

!=

<>

Question:**Score 1 of 1****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 fail because there is no PRIMARY KEY in the table.

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



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

Question:**Score 1 of 1****To permanently delete a substitution variable named THE_NAME so that it can no longer be used, use:****Response:**

UNDEFINE THE_NAME

REMOVE THE_NAME

SET DEFINE OFF

You cannot delete a substitution variable.

Question:**Score 1 of 1****Which of the following is true of character functions?****Response:**

They always accept characters as parameters and nothing else.

They always return a character value.

They generally have the letters CHAR somewhere in the function name.

-  They are generally used to process text data.

Question:

Score 1 of 1

Review the following data listing for the SHIPS table:

SHIP_ID	SHIP_NAME	CAPACITY	LENGTH	LIFEBOATS
1	Codd Crystal	2052	855	80
2	Codd Elegance	2974	952	95

Now review the following SQL statement (line numbers are added for readability):

```

01   SELECT SHIP_ID FROM   SHIPS
02   WHERE  SHIP_NAME IN ('Codd Elegance','Codd Victorious')
03       OR  (LIFEBOATS >= 80
04       OR  LIFEBOATS <= 100)
05   AND  CAPACITY / LIFEBOATS > 25;

```

Which of the following statements is true about this SELECT statement?

Response:

-  The syntax is correct.

Lines 3 and 4 have correct syntax but could be replaced with OR LIFEBOATS BETWEEN 80 AND 100.

The syntax on lines 3 and 4 is incorrect.

Line 5 is missing parentheses.

Question:

Score 1 of 1

Which of the following aggregate functions ignores NULL values in its calculations?

(Choose all that apply.)

Response:

-  SUM

-  AVG

MAX

MEDIAN

Question:

Score 1 of 1

Which of the following SQL statements will authorize the user account JESSE to create tables in each and every user account in the database?

Response:

GRANT CREATE PUBLIC TABLE TO JESSE;

GRANT CREATE TABLE TO JESSE WITH PUBLIC OPTION;

GRANT CREATE ALL TABLE TO JESSE;



GRANT CREATE ANY TABLE TO JESSE;

Question:

Score 1 of 1

You are logged in to user account FRED and have been tasked with granting privileges to the user account ETHEL. You execute the following SQL statements:

GRANT CREATE ANY TABLE TO ETHEL WITH ADMIN OPTION;
REVOKE CREATE ANY TABLE FROM ETHEL;

Assuming both statements execute successfully, what is the result?

Response:

ETHEL does not have the system privilege CREATE ANY TABLE or the right to grant the CREATE ANY TABLE system privilege to any other user.

ETHEL has the system privilege CREATE ANY TABLE because the WITH ADMIN OPTION clause wasn't included in the REVOKE statement.

ETHEL no longer has the system privilege CREATE ANY TABLE but still has the right to grant the CREATE ANY TABLE system privilege to any other user since the WITH ADMIN OPTION clause was omitted. Furthermore, ETHEL may grant the CREATE ANY TABLE privilege to herself because of the WITH ADMIN OPTION clause.

ETHEL no longer has the system privilege CREATE ANY TABLE but still has the right to grant the CREATE ANY TABLE system privilege to any other user, since the WITH

ADMIN OPTION clause was omitted from the REVOKE statement. However, ETHEL may not grant the CREATE ANY TABLE privilege to herself.

Question:

Score 1 of 1

Review the illustration and then review the following SQL statement:

CRUISE_ORDERS			
P	*	CRUISE_ORDER_ID	NUMBER
P	*	ORDER_DATE	DATE
 PK_CO			

```
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 because you cannot use the MIN function on a DATE data type.



It will execute and perform as intended.

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

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

Question:

Score 1 of 1

Which three statements are true regarding subqueries?

Response:

Main query and subquery can get data from different tables.

Subqueries can contain ORDER BY but not the GROUP BY clause.



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

Only one column or expression can be compared between the main query and

subquery.



Subqueries can contain GROUP BY and ORDER BY clauses

Main query and subquery must get data from the same tables

Question:

Score 1 of 1

Review the following SQL statement:

```
CREATE TABLE personnel
( personnel_ID      NUMBER(6),
  division_ID       NUMBER(6),
  CONSTRAINT personnel_ID_PK PRIMARY KEY (personnel_ID),
  CONSTRAINT division_ID_PK PRIMARY KEY (division_ID));
```

Assume there is no table already called PERSONNEL in the database. What will be the result of an attempt to execute the preceding SQL statement?

Response:

The statement will successfully create the table and the first primary key but not the second.

The statement will successfully create a single table and one composite primary key consisting of two columns.



The statement will fail because you cannot create two primary key constraints on the table.

The statement will successfully create the table and two primary keys.

Question:

Score 1 of 1

Which of the following is true of functions?

Response:

They often return a value.

They never return a value.

There is no consistent answer to whether they return a value or not.



They always return a value.

Question:

Score 0 of 1

Which statement is true about SQL query processing in an Oracle database instance?

Response:

During row source generation, rows that satisfy the query are retrieved from the database and stored in memory.

 During optimization, execution plans are formulated based on the statistics gathered by the database instance, and the lowest cost plan is selected for execution.

 During executing, the oracle server may read data from storage if the required data is not already in memory.

During parsing, a SQL statement containing literals in the WHERE clause that has been executed by any session and which is cached in memory, is always reused for the current execution.

Question:

Score 1 of 1

One place to get a master list of all the views that form the data dictionary is:

Response:

 DICTIONARY

CATALOG

USER_CATALOG

DATA_DICTIONARY

Question:

Score 1 of 1

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
P *	ITEM_NAME
	VARCHAR2 (15 BYTE)
	ADDED
	DATE
	SECTION
	VARCHAR2 (10 BYTE)
PK	PK_CAT#

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

2

4

6

1**Question:****Score 1 of 1****Which three statements are true regarding the data types?****Response:**

ATIMESTAMP data type column stores only time values with fractional seconds.

The BLOB data type column is used to store binary data in an operating system file.



The value for a CHAR data type column is blank-padded to the maximum defined column width.



Only one LONG column can be used per table.

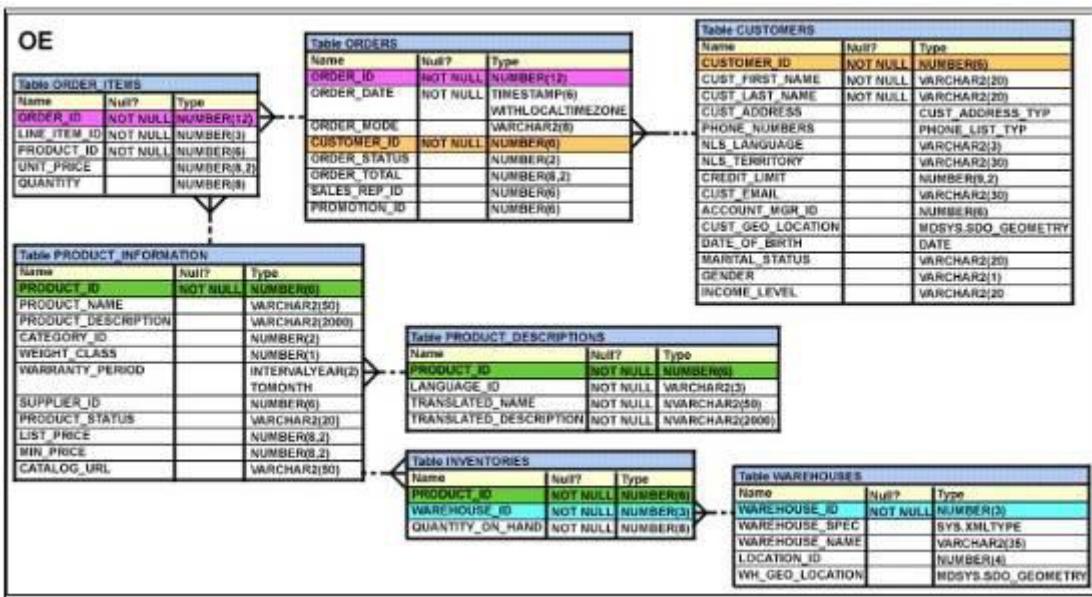


The minimum column width that can be specified for a varchar2 data type column is one.

Question:**Score 1 of 1****View the Exhibit and examine the structure of the ORDERS table. The ORDER_ID column is the PRIMARY KEY in the ORDERS table. Evaluate the following CREATE TABLE command:**

```
CREATE TABLE new_orders(ord_id, ord_date DEFAULT SYSDATE, cus_id) AS SELECT  
order_id.order_date,customer_id FROM orders;
```

Which statement is true regarding the above command?

**Response:**

The NEW_IDRTERS table would not get created because the DEFAULT value cannot be specified in the column definition.

The NEW_IDRTERS table would not get created because the column names in the CREATE TABLE command and the SELECT clause do not match.

The NEW_IDRTERS table would get created and all the constraints defined on the specified columns in the ORDERS table would be passed to the new table.



The NEW_IDRTERS table would get created and only the NOT NULL constraint defined on the specified columns would be passed to the new table.

Question:

Score 1 of 1

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:

Only the three rows with the lowest values in the key column are displayed in the order that they are stored.



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.

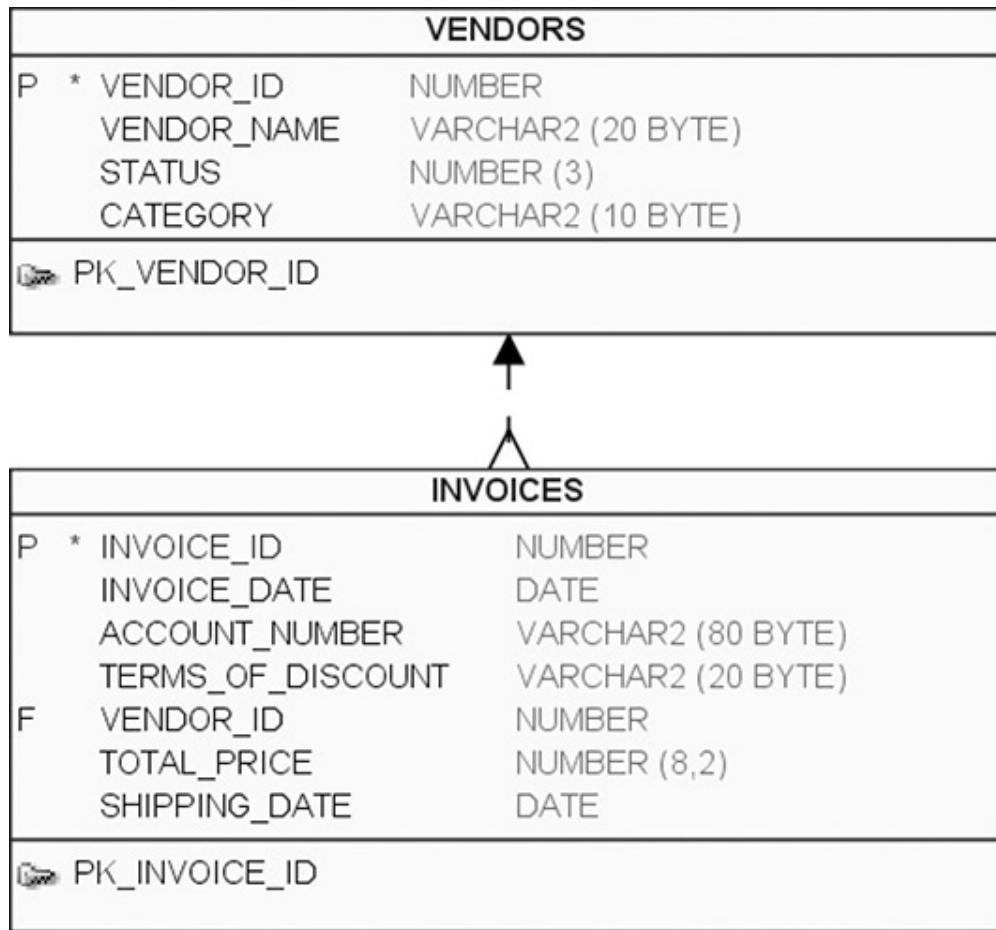
The execution fails unless the numeral 3 in the ORDER BY clause is replaced by a column name.

Question:

Score 1 of 1

Review the illustration. Which of the following is a syntactically correct outer join query?

(Choose two.)


Response:

✓ 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 FULL OUTER JOIN INVOICES
 ON VENDORS.VENDOR_ID = INVOICES.VENDOR_ID;

✓ SELECT VENDOR_NAME, INVOICE_DATE
 FROM VENDORS LEFT 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;
```
