

Score 0 of 1  
(skipped)

**Question:**

**When combining two SELECT statements, which of the following set operators will produce a different result, depending on which SELECT statement precedes or follows the operator?**

**Response:**



MINUS

UNION ALL

INTERSECT

UNION

Score 0 of 1  
(skipped)

**Question:**

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

**Response:**



The HAVING clause conditions can have aggregate functions.



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



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.

The HAVING clause conditions can use aliases for the columns.

Score 0 of 1  
(skipped)

**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 'MEMBER ID', due\_date '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;

Score 0 of 1  
(skipped)

**Question:**

An inner join queries from two tables (looking at values in columns and optionally using expressions that reference columns) and compares the resulting values in one set of rows with the resulting values in another set of rows, looking for:

**Response:**

Values that match

Values that may or may not match

Values in the first set that are greater than values in the second set

Values in the first set that are less than values in the second set

Score 0 of 1  
(skipped)

**Question:**

**What can DDL be used for?**  
(Choose three.)

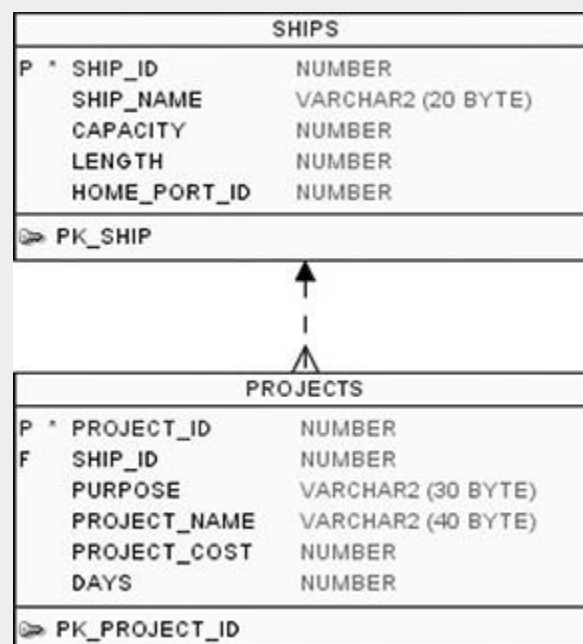
**Response:**

- ✓ Add comments to a database table
- ✓ Add columns to a database table
- Add data to a database table
- ✓ Add privileges for a user to a database table

Score 0 of 1  
(skipped)

**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.

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 uses a GROUP BY clause.

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

Score 0 of 1  
(skipped)

**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:**

 1

2

4

6

Score 0 of 1  
(skipped)

**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



4

6

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

Score 0 of 1  
(skipped)

Question:

Consider the following table listing from the table ALARM\_HISTORY:

TRACKING_DATE	INCIDENTS
-----	-----
17-OCT-2018	12
18-OCT-2018	3
19-OCT-2018	
20-OCT-2018	
21-OCT-2018	4

You are tasked to calculate the average number of alarm incidents per day in ALARM\_HISTORY. You know the following query is syntactically correct: SELECT AVG (INCIDENTS) FROM ALARM\_HISTORY;

However, you are aware that the value for INCIDENTS might be NULL, and you want the AVG returned to be calculated across every day in ALARM\_HISTORY, not just the non-NULL days. Which of the following queries will achieve this goal?

Response:



SELECT AVG (NVL (INCIDENTS)) FROM ALARM\_HISTORY;



SELECT AVG (NVL (INCIDENTS, 0)) FROM ALARM\_HISTORY;

SELECT NVL (AVG (INCIDENTS)) FROM ALARM\_HISTORY;

SELECT NVL (AVG (INCIDENTS, 0)) FROM ALARM\_HISTORY;

Score 0 of 1  
(skipped)

**Question:**

**Examine the structure of the members table:**

Name	Null?	Type
MEMBER_ID		
FIRST_NAME	NOT NULL	VARCHAR2(6)
LAST_NAME		VARCHAR2(50)
ADDRESS	NOT NULL	VARCHAR2(50)

You execute the SQL statement:

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

**What is the outcome?**

**Response:**

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

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 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.

Score 0 of 1  
(skipped)

**Question:**


**You can use a substitution variable to replace:**

**Response:**

A floating-point value in a WHERE clause

The name of a table in a SELECT statement

Neither


 Both


Score 0 of 1  
(skipped)

**Question:**

**Which of the following statements are true?**  
(Choose two.)

**Response:**

 You can use a data type conversion function to format numeric data to display dollar signs and commas.

 The presence of an explicit data type conversion documents your intent in the

Depending on the values, you can successfully use an explicit data type conversion to transform numeric values to text but not the other way around; you can't explicitly convert text to numeric.

An implicit data type conversion performs faster than an explicit data type conversion.

Score 0 of 1  
(skipped)

**Question:**

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

**Response:**

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

Roles are in the same namespace as CONSTRAINTS.

Roles are in the same namespace as TABLES.



Roles are in the same namespace as USERS.

Score 0 of 1  
(skipped)

**Question:**

**The user SCOTT who is the owner of ORDERS and ORDER\_ITEMS tables issues the following GRANT command:**

GRANT ALL ON orders, order\_items TO PUBLIC;

**What correction needs to be done to the above statement?**

**Response:**

PUBLIC should be replaced with specific usernames.

ALL should be replaced with a list of specific privileges.

WITH GRANT OPTION should be added to the statement.



Separate GRANT statements are required for ORDERS and ORDER\_ITEMS tables.

Score 0 of 1  
(skipped)

**Question:**

**Which three tasks can be performed using SQL functions built into Oracle Database?**

(Choose three.)

**Response:**

Combining more than two columns or expressions into a single column in the c

- ✓ Displaying a date in a nondefault format
- ✓ Substituting a character string in a text expression with a specified string
- ✓ Finding the number of characters in an expression

Score 0 of 1  
(skipped)

**Question:**

Review the following data listing for a table **VENDORS**:

VENDOR_ID	CATEGORY
1	Supplier
2	Teaming Partner

Now review the following SQL statement:

```
SELECT VENDOR_ID  
FROM VENDORS  
WHERE CATEGORY IN ('Supplier','Subcontractor','%Partner');
```

How many rows will the SELECT statement return?

**Response:**

2

✓ 1

0

None because it will fail due to a syntax error

Score 0 of 1  
(skipped)

**Question:**

Review the illustration and then look at the SQL code 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  COUNT (COUNT (PROJECT_COST) )
02  FROM    PROJECTS
03  GROUP BY PURPOSE;

```

What will happen if you try to execute this query on the **PROJECTS** table?

**Response:**

It will fail with a syntax error because line 1 is not correct.

It will fail with an execution error because you cannot use a VARCHAR2 column in the GROUP BY clause.

It will succeed and display one row for each different value in the PURPOSE column.

✓ It will succeed and display one row.

Score 0 of 1  
(skipped)

**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 fail with an execution error on line 1.

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

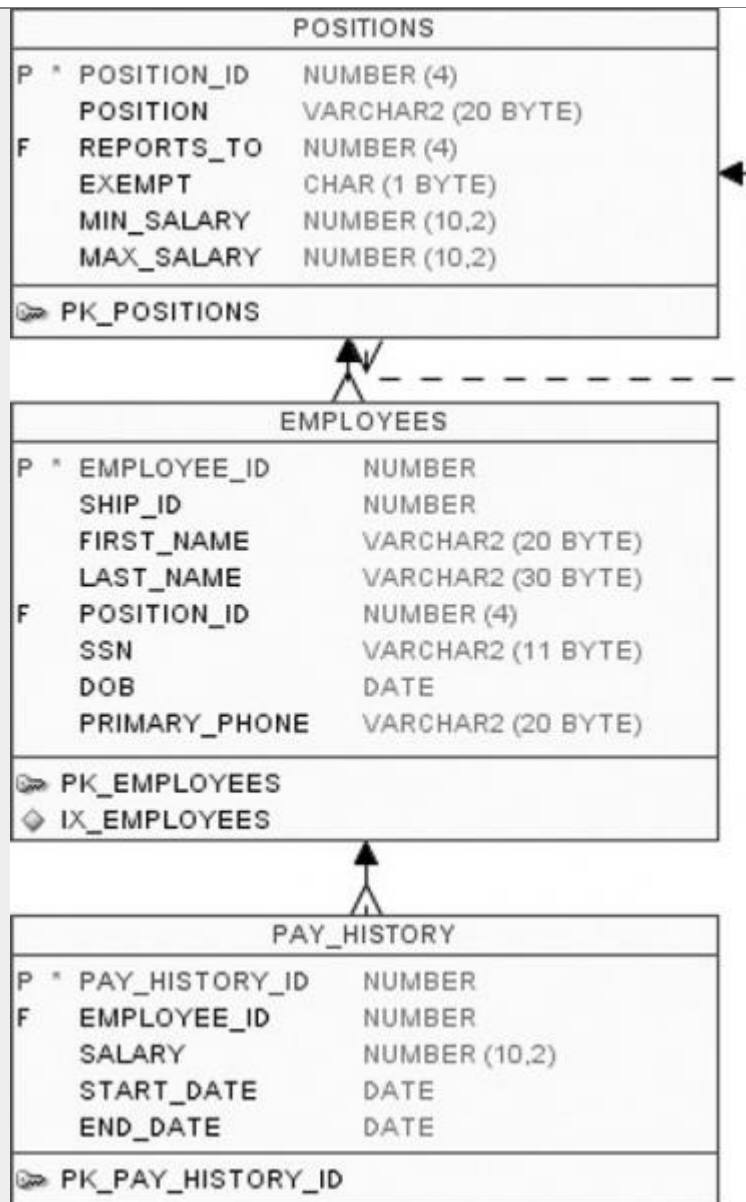
It will execute and display one column under the "Date Added" heading.

It will execute and display one column under the "LAST\_ORDER" heading.

Score 0 of 1  
(skipped)

**Question:**

**Review the illustration and then review the following SQL statement:**



```

01 SELECT A.EMPLOYEE_ID, B.POSITION
02 FROM   PAY_HISTORY A JOIN POSITIONS B
03       ON   A.SALARY < B.MAX_SALARY AND A.SALARY > B.MIN_SALARY;
  
```

Which of the following statements accurately describe the SQL statement?  
(Choose two.)

**Response:**

It contains a syntax error on line 3.





It is an inner join.



It is a non-equijoin.

It contains a syntax error on line 2 and should have an additional keyword with JOIN keyword.

Score 0 of 1  
(skipped)

**Question:**

You need to display the date 11-oct-2017 in words as 'Eleventh of October, Two Thousand Seventeen'.

Which SQL statement would give the required result?

**Response:**

SELECT TO\_CHAR (TO\_DATE ('11-oct-2017'), 'fmDdthsp of Month, Year') FROM DUAL;

SELECT TO\_CHAR ('11-oct-2017', 'fmDdspth or Month, Year') FROM DUAL;



SELECT TO\_CHAR (TO\_DATE ('11-oct-2017'), 'fmDdspth "of" Month, Year') FROM DUAL;

SELECT TO\_DATE (TO\_CHAR ('11-oct-2017'), 'fmDdspth 'of Month, Year')) FROM DUAL;

Score 0 of 1  
(skipped)

**Question:**

View the Exhibit and examine the details of the PRODUCT\_INFORMATION table.

PRODUCT_NAME	CATEGORY_ID	SUPPLIER_ID
Inkjet C/8/HQ	12	
Inkjet C/4	12	
LaserPro 600/6/BW	12	
LaserPro 1200/6/BW	12	
Inkjet B/6	12	
Industrial 700/HD	12	
Industrial 600/DQ	12	
Compact 400/LQ	12	
Compact 400/DQ	12	
HD 12GB /R	13	
HD 10GB /I	13	
HD 12GB @7200 /SE	13	
HD 18.2GB @10000 /E	13	
HD 18.2GB@10000 /I	13	
HD 18GB /SE	13	
HD 6GB /I	13	
HD 8.2GB @5400	13	

You have the requirement to display **PRODUCT\_NAME** and **LIST\_PRICE** from the table where the **CATEGORYJD** column has values 12 or 13, and the **SUPPLIER\_ID** column has the value 102088.

You executed the following SQL statement:

```
SELECT product_name, list_price FROM product_information
WHERE (category_id = 12 AND category_id = 13) AND supplier_id = 102088;
```

Which statement is true regarding the execution of the query?

**Response:**



It would execute but the output would return no rows.

It would execute and the output would display the desired result.

It would not execute because the entire WHERE clause condition is not enclosed within the parentheses.

It would not execute because the same column has been used in both sides of AND logical operator to form the condition.

Score 0 of 1  
(skipped)

**Question:**

**Evaluate the following SQL statement:**

```
SELECT product_name || 'it's not available for order' FROM product_information  
WHERE product_status = 'obsolete';
```

**You received the following error while executing the above query:**

ERROR: ORA-01756: quoted string not properly terminated

**What would you do to execute the query successfully?**

**Response:**

Enclose the character literal string in the SELECT clause within the double quotation marks.

Do not enclose the character literal string in the SELECT clause within the single quotation marks.



Use Quote (q) operator and delimiter to allow the use of single quotation mark inside the character literal string.

Use escape character to negate the single quotation mark inside the character literal string in the SELECT clause.

Score 0 of 1  
(skipped)

**Question:**

**An aggregate function can be called from within:**  
(Choose two.)

**Response:**

The HAVING clause of an INSERT statement



The ORDER BY clause of a SELECT statement

The expression list of a DELETE statement



The select list of a SELECT statement

Score 0 of 1  
(skipped)

**Question:**

**Which three statements are true regarding the data types?**

**Response:**



Only one LONG column can be used per table.

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.



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



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

Score 0 of 1  
(skipped)

**Question:**

**Review this code:**

```
DROP SEQUENCE PROJ_ID_SEQ#;  
CREATE SEQUENCE PROJ_ID_SEQ# START WITH 1 INCREMENT BY 2;  
SELECT PROJ_ID_SEQ#.CURRVAL FROM DUAL;
```

**What will result from these SQL statements?**

**Response:**

The SELECT statement will fail because the sequence can be referenced only in an INSERT statement.



The SELECT statement will fail because you cannot reference the CURRVAL pseudocolumn of a sequence until after you have referenced NEXTVAL for the sequence in a session.

The SELECT statement will display a value of 1.

The SELECT statement will display a value of 3.

Score 0 of 1  
(skipped)

**Question:**


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:**

SUM

 COUNT

MEDIAN


HAVING

Score 0 of 1  
(skipped)

**Question:**

When transforming an ERD into a relational database, you often use an entity to build a database's:

**Response:**

 Table

Column

Attribute

Relationship

Score 0 of 1  
(skipped)

**Question:**

You want to display 5 percent of the rows from the sales table for products with the lowest `AMOUNT_SOLD` and also want to include the rows that have the same `AMOUNT_SOLD` even if this causes the output to exceed 5 percent of the rows.

Which query will provide the required result?

**Response:**



```
SELECT prod_id, cust_id, amount_sold FROM sales  
ORDER BY amount_sold  
FETCH FIRST 5 PERCENT ROWS WITH TIES;
```

```
SELECT prod_id, cust_id, amount_sold FROM sales  
ORDER BY amount_sold  
FETCH FIRST 5 PERCENT ROWS ONLY WITH TIES;
```

```
SELECT prod_id, cust_id, amount_sold FROM sales  
ORDER BY amount_sold  
FETCH FIRST 5 PERCENT ROWS WITH TIES ONLY;
```

```
SELECT prod_id, cust_id, amount_sold FROM sales  
ORDER BY amount_sold  
FETCH FIRST 5 PERCENT ROWS ONLY;
```

Score 0 of 1  
(skipped)

**Question:**

**Built-in SQL functions:**  
(Choose three.)

**Response:**



Can be invoked from a `DELETE` statement's `WHERE` clause.

Are written by SQL developers and also known as "user-defined" functions.



Are available for use from the UPDATE statement.



Are available for use within a SELECT statement's WHERE clause, as well as the SELECT statement's expression list.

Score 0 of 1  
(skipped)

**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:**

Performance would improve query 2 only if there are null values in the CUST\_CREDIT\_LIMIT column.



There would be no change in performance.

Performance would degrade in query 2.

Performance would improve in query 2.

Score 0 of 1  
(skipped)

**Question:**

**Consider the following query, its output, and a subsequent query:**

```
SQL> SELECT * FROM LINE_ITEMS;
```

```
LINE_ITEM  PRICE
```

```
-----
```

```
100      4.12
```

```
210
```

```
184      7.07
```

```
SQL> SELECT NVL(PRICE,10) FROM LINE_ITEMS;
```

What is true of the final query shown previously?

Response:

It will return "no rows found" because there is no PRICE of 10.

It will return only the row where LINE\_ITEM is 210.

It will return no rows because there is no PRICE of 10.



It will return three rows, but it will not change the price for line items 100 and

Score 0 of 1  
(skipped)

Question:

Which of the following is the system privilege that empowers the grantee to create an index in his or her own user account but not in the accounts of others?

Response:



CREATE TABLE

CREATE ANY TABLE

CREATE INDEX

CREATE ANY INDEX

Score 0 of 1  
(skipped)



**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\_YMINTERVAL('01-03') FROM DUAL;

SELECT TO\_INTERVALYM('01:03') FROM DUAL;

SELECT TO\_INTERVALYM('01-03') FROM DUAL;

Score 0 of 1  
(skipped)


**Question:**

Review the diagrams and examine the following statement:

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

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.

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.



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

Score 0 of 1  
(skipped)

**Question:**

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

**Response:**

adding a column constraint when inserting a row into a table

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

Score 0 of 1  
(skipped)

**Question:**

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

EMPLOYEES  
Name Null? Type

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

Name Null? Type

```
-----  
DEPARTMENT_ID NOT NULL NUMBER(4)  
DEPARTMENT_NAME NOT NULL VARCHAR2(30)  
MANAGER_ID NUMBER(6)  
LOCATION_ID NUMBER(4)
```

**You want to update EMPLOYEES table as follows:**

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

**You issue the following command:**

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

**What is outcome?**

**Response:**

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

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

It executes successfully and gives the correct result.



It executes successfully but does not give the correct result.

Score 0 of 1  
(skipped)

**Question:**

**Which of the following data dictionary views contains information about grants on tables that have been made by other users to your user account, as well as grants on tables that have been made by your user account to other user accounts?**

**Response:**

USER\_TAB\_COLUMNS



USER\_TAB\_PRIVS

USER\_TABLES

ALL\_TAB\_PRIVS\_RECD

Score 0 of 1  
(skipped)

**Question:**

**Which of the following SQL statements will display the current time, in hours, minutes, and seconds, as determined by the operating system on which the database server resides?**

**Response:**

SELECT TO\_CHAR(SYSDATE) FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'HR:MI:SE') FROM DUAL;



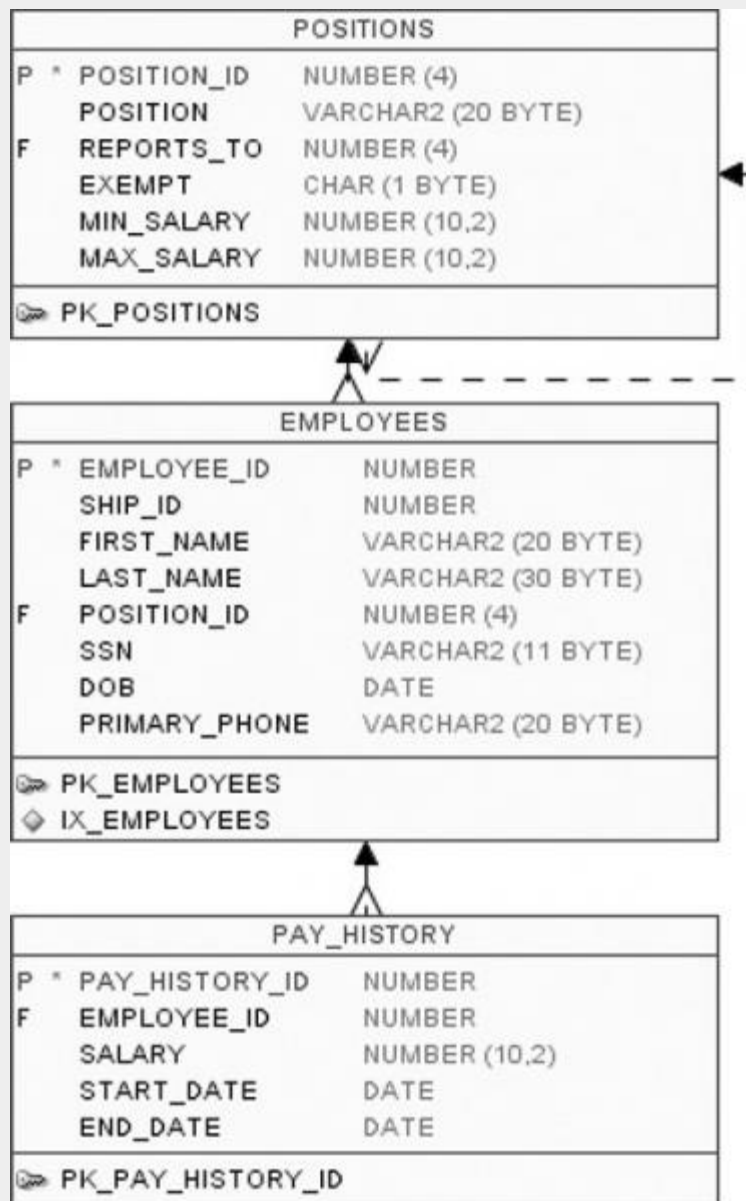
SELECT TO\_CHAR(SYSDATE, 'HH:MI:SS') FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'HH:MM:SS') FROM DUAL;

Score 0 of 1  
(skipped)

**Question:**

**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 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 INNER 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;
```

Score 0 of 1  
(skipped)

**Question:**

Using the CUSTOMERS table, you need to generate a report that shows 50% of each credit amount in each income level. The report should NOT show any repeated credit amounts in each income level.

Which query would give the required result?

**Response:**

```
SELECT cust_income_level || ' ' || cust_credit_limit * 0.50 AS '50% Credit Lim
FROM customers;
```

✓  

```
SELECT DISTINCT cust_income_level || ' ' || cust_credit_limit * 0.50 AS "50%
Limit" FROM customers;
```

```
SELECT DISTINCT cust_income_level, DISTINCT cust_credit_limit * 0.50 AS '5
Credit Limit° FROM customers, IT;
```

```
SELECT cust_income_level, DISTINCT cust_credit_limit * 0.50 AS '50% Credit
```

FROM customers;

Score 0 of 1  
(skipped)

**Question:**

**Which among the following is considered an acceptable query with V\$DATAFILE?**

**Response:**

A join with two other objects in the data dictionary

A complex GROUP BY with multiple levels of aggregation

 A query that displays rows from the table with no joins

All of the above

Score 0 of 1  
(skipped)

**Question:**

**Examine the following query:**


```
SQL> SELECT prod_id, amount_sold  
      FROM sales  
      ORDER BY amount_sold  
      FETCH FIRST 5 PERCENT ROWS ONLY;
```

**What is the output of this query?**

**Response:**

It displays 5 percent of the products with the highest amount sold.

It displays the first 5 percent of the rows from the SALES table.

 It displays 5 percent of the products with the lowest amount sold.

It results in an error because the ORDER BY clause should be the last clause.



Score 0 of 1  
(skipped)

**Question:**

**Which of the following topics are not included in the SQL Fundamentals I exam but are addressed on the SQL Associate exam?**

(Choose all that apply.)

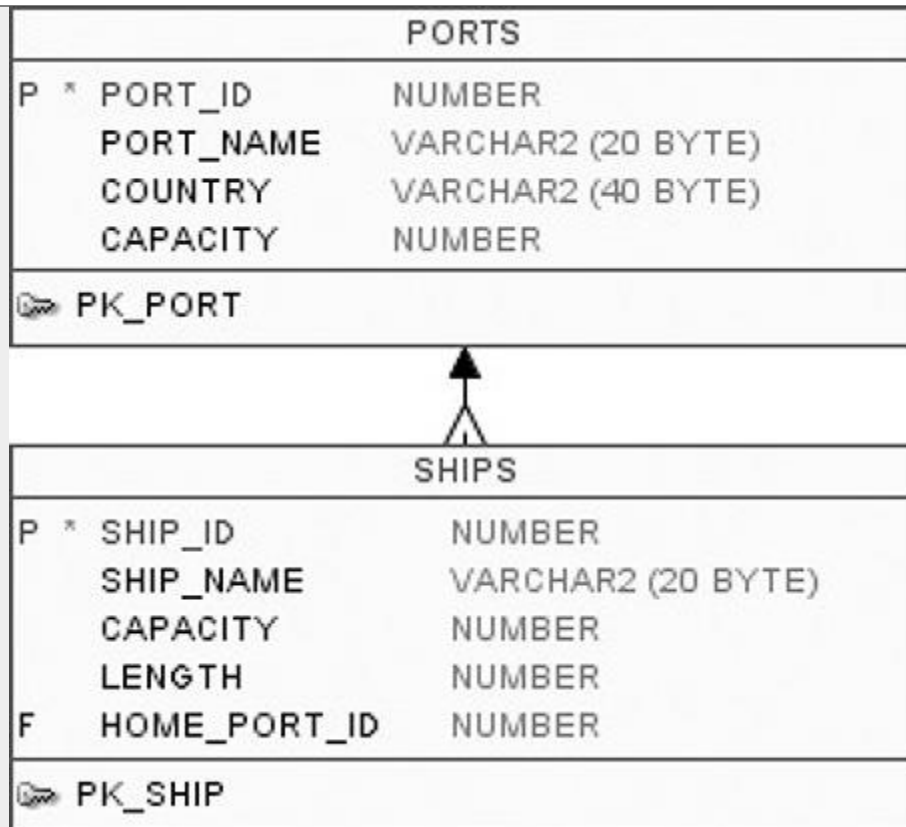
**Response:**

- ☒ MERGE
- ☒ Conversion functions
- ☒ FLASHBACK
- ☒ External tables

Score 0 of 1  
(skipped)

**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.

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

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

None of the above.

Score 0 of 1  
(skipped)

**Question:**

**Which three arithmetic operations can be performed on a column by using a SQL function that is built into Oracle database?**

(Choose three.)

**Response:**

- ☒ Finding the lowest value
- ☐ Finding the quotient
- ☒ Raising to a power
- ☐ Subtraction
- ☒ Addition

Score 0 of 1  
(skipped)

**Question:**

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

**Response:**

- ☐ Franklyn
- ☐ Phrankline
- ☐ Ellison
- ☒ None of the above

Score 0 of 1  
(skipped)

**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 DOME.HARDING.TEAPOT;

SELECT \* FROM HARDING.DOME.TEAPOT;

 SELECT \* FROM HARDING.TEAPOT;

None of the above

Score 0 of 1  
(skipped)


**Question:**

Which of the following is a true statement?

**Response:**

If a **SELECT** includes a **GROUP BY** clause, then any subquery used within the **SELECT** must also have a **GROUP BY** clause.

If a query returns multiple rows, it may not be used as a subquery for a **SELECT** statement that uses a **GROUP BY** clause.

 A **SELECT** statement with a **GROUP BY** may use a subquery to return a value to the outermost **WHERE** clause.

The only form of subquery permitted with a **GROUP BY** clause is a correlated subquery.

Score 0 of 1  
(skipped)

**Question:**


**The LEAD function returns data from:**

**Response:**

A row prior to the current row as specified by the LEAD function's ORDER BY clause

A row following the current row as specified by the SELECT statement's ORDER BY clause

The LAG function's window's specified column

 The row specified by the LEAD function's offset

Score 0 of 1  
(skipped)


**Question:**

**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 fail because you cannot create two primary key constraints on the same table.

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 successfully create the table and two primary keys.

Score 0 of 1  
(skipped)

**Question:**

**The difference between dropping a column from a table with DROP and setting a column to be UNUSED is:**

**Response:**

An UNUSED column can be recovered.



The UNUSED column and its data are retained within the table's storage allocation and counts against the total limit on the number of columns the table is allowed to have.

A column that is dropped with DROP no longer appears within the table's description as shown with the DESC or DESCRIBE statement, whereas a column that is set to UNUSED still appears in the table's structure as shown in the output of the DESCRIBE statement.

Nothing.

Score 0 of 1  
(skipped)

**Question:**

**Conversion functions cannot be used to:**

**Response:**

Format date values



Convert columns to new data types

Transform data



Create user-defined data types

Score 0 of 1  
(skipped)

**Question:**

**Which of the following aggregate functions can be used on character data?**  
(Choose two.)

**Response:**



COUNT



MIN

AVG

MEDIAN

Score 0 of 1  
(skipped)

**Question:**

**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 DROP;

COMMENT ON COLUMN MARINA.PIER IS NULL;

COMMENT ON COLUMN MARINA.PIER SET UNUSED;



COMMENT ON COLUMN MARINA.PIER IS '';

Score 0 of 1  
(skipped)

**Question:**

**Which of the following statements are true?**  
(Choose two.)

**Response:**

A single-row subquery can also be a multiple-row subquery.

✓ A single-row subquery can also be a multiple-column subquery.

A scalar subquery can also be a multiple-column subquery.

✓ A correlated subquery can also be a single-row subquery.

Score 0 of 1  
(skipped)

**Question:**

**View the Exhibit for the structure of the STUDENT and FACULTY tables.**

STUDENT		
Name	Null?	Type
-----		
STUDENT_ID	NOT NULL	NUMBER(2)
STUDENT_NAME		VARCHAR2(20)
FACULTY_ID		VARCHAR2(2)
LOCATION_ID		NUMBER(2)
FACULTY		
Name	Null?	Type
-----		
FACULTY_ID	NOT NULL	NUMBER(2)
FACULTY_NAME		VARCHAR2(20)
LOCATION_ID		NUMBER(2)

**You need to display the faculty name followed by the number of students handled by the faculty at the base location. Examine the following two SQL statements:**



Statement 1

```
SQL>SELECT faculty_name,COUNT(student_id)
FROM student JOIN faculty
USING (faculty_id, location_id)
GROUP BY faculty_name;
```

Statement 2

```
SQL>SELECT faculty_name,COUNT(student_id)
FROM student NATURAL JOIN faculty
GROUP BY faculty_name;
```

Which statement is true regarding the outcome?

Response:

Only statement 2 executes successfully and gives the required result.

Only statement 1 executes successfully and gives the required result.

Both statements 1 and 2 execute successfully and give different results.



Both statements 1 and 2 execute successfully and give the same required result.

Score 0 of 1  
(skipped)

Question:

Which of the following is true about aggregate functions?

(Choose two.)

Response:



Return one value for each group of rows specified in a SELECT statement.



Are also called group functions.

Will cause a run-time error when used in SELECT statements that return zero or one row.

Can operate only with numeric data.




Score 0 of 1  
(skipped)

**Question:**

**Which three statements are true regarding group functions?**

(Choose three.)

**Response:**

-  They can be used on columns or expressions.
-  They can be passed as an argument to another group function.
- ☐ They can be used only with a SQL statement that has the GROUP BY clause.
- ☐ They can be used on only one column in the SELECT clause of a SQL statement.
-  They can be used along with the single-row function in the SELECT clause of a statement.



Score 0 of 1  
(skipped)

**Question:**

**Which two statements are true regarding constraints?**

(Choose two.)

**Response:**

- ☐ A foreign key cannot contain NULL values.
-  A column with the UNIQUE constraint can contain NULL.
- ☐ A constraint is enforced only for the INSERT operation on a table.
-  A constraint can be disabled even if the constraint column contains data.

All the constraints can be defined at the column level as well as the table level

Score 0 of 1  
(skipped)

**Question:**

**The difference between an INNER and an OUTER join is:**

**Response:**

The INNER join relates a table to itself; the OUTER join relates a table to other tables.



The INNER join displays rows that match in all joined tables; the OUTER join displays data that doesn't necessarily match.

The OUTER join relates a table to tables in other user accounts; the INNER join does not.

The INNER join runs on data inside the table; the OUTER join runs on data outside of the table.

Score 0 of 1  
(skipped)

**Question:**

**If you focus on trying to achieve the minimum passing grade requirement for the exam, you can study more efficiently.**

**Response:**

True



False

Score 0 of 1  
(skipped)

**Question:**

**Assume a database with three valid users: NEIL, BUZZ, and MICHAEL. Assume all users have the appropriate privileges they require to perform the tasks shown here. Assume NEIL owns a table called PROVISIONS.**


**Examine the following code (assume all password references are valid):**

```
01  CONNECT NEIL/neilPassword
02  GRANT SELECT ON PROVISIONS TO BUZZ, MICHAEL;
03
04  CONNECT BUZZ/buzzPassword
05  CREATE VIEW PROVISIONS AS SELECT * FROM NEIL.PROVISIONS;
06  GRANT SELECT ON PROVISIONS TO MICHAEL;
07  CREATE PUBLIC SYNONYM PROVISIONS FOR BUZZ.PROVISIONS;
08
09  CONNECT MICHAEL/michaelPassword
10  CREATE SYNONYM PROVISIONS FOR NEIL.PROVISIONS;
11  SELECT * FROM PROVISIONS;
```

**What object is identified in line 11 by the name PROVISIONS?**

**Response:**

The public synonym created in line 7

 The synonym created in line 10

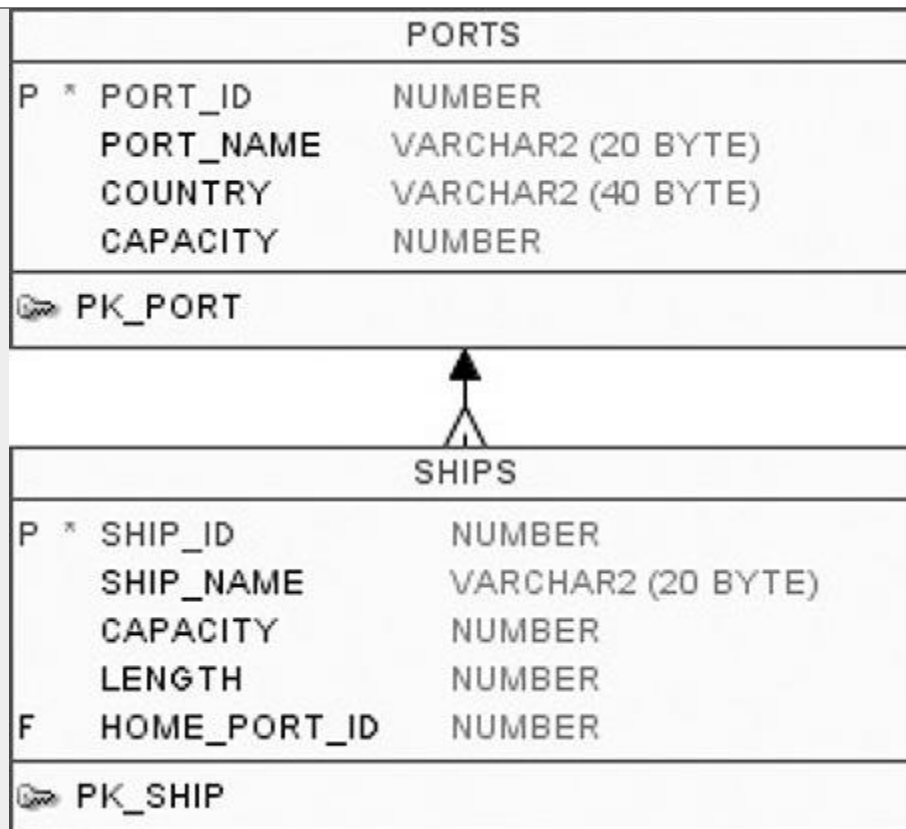
Nothing, because user NEIL did not include WITH GRANT OPTIONS in the GRANT SELECT ON PROVISIONS TO BUZZ statement

Something else not listed above

Score 0 of 1  
(skipped)

**Question:**

**Review the illustration and the following SQL code:**



```
01 DELETE FROM PORTS P
02 WHERE PORT_ID NOT EXISTS (SELECT PORT_ID
03                             FROM SHIPS
04                             WHERE HOME_PORT_ID = P.PORT_ID);
```

The code is attempting to delete any row in the **PORTS** table that is not a home port for any ship in the **SHIPS** table, as indicated by the **HOME\_PORT\_ID** column.

In other words, only keep the **PORTS** rows that are currently the **HOME\_PORT\_ID** value for a ship in the **SHIPS** table; get rid of all other **PORT** rows. That's the intent of the SQL statement.

What will result from an attempt to execute the preceding SQL statement?

Response:



It will fail because of a syntax error on line 2.

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

It will fail because of an execution error in the subquery.

It will execute successfully and perform as intended.


Score 0 of 1  
(skipped)

**Question:**


**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_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	

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 there is no ELSE clause.

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



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.

The statement will add every row returned from the SPARES table to the SHIP\_INVENTORY table.

Score 0 of 1  
(skipped)

**Question:**

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

**Response:**



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

Only two subqueries can be placed at one level.



A subquery can retrieve zero or more rows.

A subquery can be used only in SQL query statements.

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

Score 0 of 1  
(skipped)

**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:**



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 condition.

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.



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.

None of the above.

Score 0 of 1  
(skipped)

**Question:**

**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 UNIQUE (POSITION_ID));
```



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

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

None of the above

Score 0 of 1  
(skipped)

**Question:**

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(100000) WITHIN GROUP (ORDER BY PROJECT\_COST) FROM PROJECTS;

SELECT RANK(100,000) WITHIN GROUP (ORDER BY PROJECT\_COST) FROM PROJECTS;

SELECT RANK(7500000) GROUP BY (ORDER BY PROJECT\_COST) FROM PROJECTS;

SELECT RANK('Upgrade') WITHIN GROUP (ORDER BY PROJECT\_COST) FROM PROJECTS;

Score 0 of 1  
(skipped)

**Question:**

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

**Response:**



Add rows into more than one table



Add data into more than one column in a table

Delete rows by overwriting them

Join tables together