

**Oracle.1z0-051.Examcollection.Premium.Exam.292q**

Number: 1z0-051  
Passing Score: 800  
Time Limit: 120 min  
File Version: 14.12



**Exam Code: 1z0-051**

**Exam Name: Oracle Database: SQL Fundamentals I**

**Version 14.12**

1z0-051

### QUESTION 1

Evaluate the SQL statement:  
TRUNCATE TABLE DEPT;

Which three are true about the SQL statement? (Choose three.)

- A. It releases the storage space used by the table.
- B. It does not release the storage space used by the table.
- C. You can roll back the deletion of rows after the statement executes.
- D. You can NOT roll back the deletion of rows after the statement executes.
- E. An attempt to use DESCRIBE on the DEPT table after the TRUNCATE statement executes will display an error.
- F. You must be the owner of the table or have DELETE ANY TABLE system privileges to truncate the DEPT table

**Correct Answer:** ADF

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

A: The TRUNCATE TABLE Statement releases storage space used by the table,

D: Can not rollback the deletion of rows after the statement executes,

F: You must be the owner of the table or have DELETE ANY TABLE system privilege to truncate the DEPT table.

Incorrect answer:

C is not true

D is not true

E is not true

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-18

### QUESTION 2

You need to design a student registration database that contains several tables storing academic information. The STUDENTS table stores information about a student. The STUDENT\_GRADES table stores information about the student's grades. Both of the tables have a column named STUDENT\_ID. The STUDENT\_ID column in the STUDENTS table is a primary key.

You need to create a foreign key on the STUDENT\_ID column of the STUDENT\_GRADES table that points to the STUDENT\_ID column of the STUDENTS table. Which statement creates the foreign key?

- A. CREATE TABLE student\_grades (student\_id NUMBER(12),semester\_end DATE, gpa NUMBER(4,3), CONSTRAINT student\_id\_fk REFERENCES (student\_id) FOREIGN KEY students(student\_id));
- B. CREATE TABLE student\_grades(student\_id NUMBER(12),semester\_end DATE, gpa NUMBER(4,3), student\_id\_fk FOREIGN KEY (student\_id) REFERENCES students(student\_id));
- C. CREATE TABLE student\_grades(student\_id NUMBER(12),semester\_end DATE, gpa NUMBER(4,3), CONSTRAINT FOREIGN KEY (student\_id) REFERENCES students(student\_id));
- D. CREATE TABLE student\_grades(student\_id NUMBER(12),semester\_end DATE, gpa NUMBER(4,3), CONSTRAINT student\_id\_fk FOREIGN KEY (student\_id) REFERENCES students(student\_id));

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: CONSTRAINT name FOREIGN KEY (column\_name) REFERENCES table\_name (column\_name);

Incorrect answer:

A invalid syntax

B invalid syntax

C invalid syntax

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-14

**QUESTION 3**

Here is the structure and data of the CUST\_TRANS table:

Exhibit:

CUST_TRANS		
Name	Null?	Type
-----	-----	-----
CUSTNO	NOT NULL	CHAR(2)
TRANSDATE	DATE	
TRANSAMT		NUMBER(6,2)

CUSTNO	TRANSDATE	TRANSAMT
-----	-----	-----
11	01-JAN-07	1000
22	01-FEB-07	2000
33	01-MAR-07	3000

Dates are stored in the default date format dd-mm-rr in the CUST\_TRANS table. Which three SQL statements would execute successfully? (Choose three.)

- A. SELECT transdate + '10' FROM cust\_trans;
- B. SELECT \* FROM cust\_trans WHERE transdate = '01-01-07';
- C. SELECT transamt FROM cust\_trans WHERE custno > '11';
- D. SELECT \* FROM cust\_trans WHERE transdate='01-JANUARY-07';
- E. SELECT custno + 'A' FROM cust\_trans WHERE transamt > 2000;

**Correct Answer:** ACD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 4**

See the Exhibit and examine the structure and data in the INVOICE table:

Exhibit:

INVOICE			
Name	Null?	Type	
INV_NO	NOT NULL	NUMBER(3)	
INV_DATE		DATE	
CUST_ID		VARCHAR2(4)	
INV_AMT		NUMBER(8,2)	

INV_NO	INV_DATE	CUST_ID	INV_AMT
1	01-APR-07	AIQ	1000
2	01-OCT-07	BIK	2000
3	01-FEB-07		3000

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT MAX(inv\_date),MIN(cust\_id) FROM invoice;
- B. SELECT MAX(AVG(SYSDATE - inv\_date)) FROM invoice;
- C. SELECT (AVG(inv\_date)) FROM invoice;
- D. SELECT AVG(inv\_date - SYSDATE),AVG(inv\_amt) FROM invoice;

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 5

Which three statements are true regarding sub queries? (Choose three.)

- A. Multiple columns or expressions can be compared between the main query and sub query
- B. Main query and sub query can get data from different tables
- C. Sub queries can contain GROUP BY and ORDER BY clauses
- D. Main query and sub query must get data from the same tables
- E. Sub queries can contain ORDER BY but not the GROUP BY clause
- F. Only one column or expression can be compared between the main query and subquery

**Correct Answer:** ABC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 6

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_MARITAL_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 shown the average credit limit for customers in WASHINGTON and NEW YORK.

Which SQL statement would produce the required result?

- A. 

```
SELECT cust_city, AVG(cust_credit_limit)
FROM customers
WHERE cust_city IN ('WASHINGTON','NEW YORK')
GROUP BY cust_credit_limit, cust_city;
```
- B. 

```
SELECT cust_city, AVG(cust_credit_limit)
FROM customers
WHERE cust_city IN ('WASHINGTON','NEW YORK')
GROUP BY cust_city, cust_credit_limit;
```
- C. 

```
SELECT cust_city, AVG(cust_credit_limit)
FROM customers
WHERE cust_city IN ('WASHINGTON','NEW YORK')
GROUP BY cust_city;
```
- D. 

```
SELECT cust_city, AVG(NVL(cust_credit_limit,0))
FROM customers
WHERE cust_city IN ('WASHINGTON','NEW YORK');
```

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Creating Groups of Data: GROUP BY Clause Syntax

You can use the GROUP BY clause to divide the rows in a table into groups. You can then use the group functions to return summary information for each group.

In the syntax:

group\_by\_expression Specifies the columns whose values determine the basis for grouping rows Guidelines

- If you include a group function in a SELECT clause, you cannot select individual results as well, unless the individual column appears in the GROUP BY clause. You receive an error message if you fail to include the column list in the GROUP BY clause.
- Using a WHERE clause, you can exclude rows before dividing them into groups.

- You must include the columns in the GROUP BY clause.

- You cannot use a column alias in the GROUP BY clause.

## QUESTION 7

Evaluate these two SQL statements:

```
SELECT last_name, salary, hire_date FROM EMPLOYEES ORDER BY salary DESC;
SELECT last_name, salary, hire_date FROM EMPLOYEES ORDER BY 2 DESC;
```

What is true about them?

- A. The two statements produce identical results.
- B. The second statement returns a syntax error.
- C. There is no need to specify DESC because the results are sorted in descending order by default.
- D. The two statements can be made to produce identical results by adding a column alias for the salary column in the second SQL statement.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: the two statements produce identical results as ORDER BY 2 will take the second column as sorting column.

Incorrect answer:

B there is no syntax error

C results are sorted in ascending order by default

D ORDER BY 2 will take the second column as sorting column. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-22

### QUESTION 8

Where can sub queries be used? (Choose all that apply)

- A. field names in the SELECT statement
- B. the FROM clause in the SELECT statement
- C. the HAVING clause in the SELECT statement
- D. the GROUP BY clause in the SELECT statement
- E. the WHERE clause in only the SELECT statement
- F. the WHERE clause in SELECT as well as all DML statements

**Correct Answer:** ABCF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

SUBQUERIES can be used in the SELECT list and in the FROM, WHERE, and HAVING clauses of a query.

A subquery can have any of the usual clauses for selection and projection. The following are required clauses:

A SELECT list

A FROM clause

The following are optional clauses:

WHERE

GROUP BY

HAVING

The subquery (or subqueries) within a statement must be executed before the parent query that calls it, in order that the results of the subquery can be passed to the parent.

### QUESTION 9

Which three SQL statements would display the value 1890.55 as \$1,890.55? (Choose three.)

- A. SELECT TO\_CHAR(1890.55,'\$99G999D00')  
FROM DUAL;
- B. SELECT TO\_CHAR(1890.55,'\$9,999V99')

- FROM DUAL;
- C. SELECT TO\_CHAR(1890.55,'\$0G000D00')  
FROM DUAL;
- D. SELECT TO\_CHAR(1890.55,'\$99G999D99')  
FROM DUAL;
- E. SELECT TO\_CHAR(1890.55,'\$9,999D99')  
FROM DUAL;

**Correct Answer:** ACD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 10

Evaluate the following SQL statement:

```
SQL> SELECT promo_id, promo_category  
FROM promotions  
WHERE promo_category = 'Internet' ORDER BY 2 DESC  
UNION  
SELECT promo_id, promo_category  
FROM promotions  
WHERE promo_category = 'TV'  
UNION  
SELECT promo_id, promo_category  
FROM promotions  
WHERE promo_category = 'Radio';
```

Which statement is true regarding the outcome of the above query?

- A. It produces an error because the ORDER BY clause should appear only at the end of a compound query- that is, with the last SELECT statement
- B. It executes successfully and displays rows in the descending order of PROMO\_CATEGORY
- C. It executes successfully but ignores the ORDER BY clause because it is not located at the end of the compound statement
- D. It produces an error because positional notation cannot be used in the ORDER BY clause with SET operators

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the ORDER BY Clause in Set Operations

The ORDER BY clause can appear only once at the end of the compound query. Component queries cannot have individual ORDER BY clauses. The ORDER BY clause recognizes only the columns of the first SELECT query. By default, the first column of the first SELECT query is used to sort the output in an ascending order.

#### QUESTION 11

Which statement correctly describes SQL and /SQL\*Plus?

- A. Both SQL and /SQL\*plus allow manipulation of values in the database.

- B. /SQL\*Plus recognizes SQL statements and sends them to the server; SQL is the Oracle proprietary interface for executing SQL statements.
- C. /SQL\*Plus is a language for communicating with the Oracle server to access data; SQL recognizes SQL statements and sends them to the server.
- D. SQL manipulates data and table definitions in the database; /SQL\*Plus does not allow manipulation of values in the database.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 12

Which four are types of functions available in SQL? (Choose 4)

- A. string
- B. character
- C. integer
- D. calendar
- E. numeric
- F. translation
- G. date
- H. conversion

**Correct Answer:** BEGH

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: SQL have character, numeric, date, conversion function.

Incorrect answer:

A SQL have character, numeric, date, conversion function.

C SQL have character, numeric, date, conversion function.

D SQL have character, numeric, date, conversion function.

F SQL have character, numeric, date, conversion function.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 3-3

#### QUESTION 13

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

```
EMPLOYEES
EMPLOYEE_ID NUMBER Primary Key
FIRST_NAME VARCHAR2(25)
LAST_NAME VARCHAR2(25)
HIRE_DATE DATE
```

```
NEW_EMPLOYEES
EMPLOYEE_ID NUMBER Primary Key
NAME VARCHAR2(60)
```

Which MERGE statement is valid?



- A. MERGE INTO new\_employees c  
USING employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN MATCHED THEN  
UPDATE SET
- B. name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT  
value  
S(e.employee\_id, e.first\_name || '  
' || e.last\_name);
- C. MERGE new\_employees c  
USING employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN EXISTS THEN  
UPDATE SET
- D. name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN INSERT  
valueS(e.employee\_id, e.first\_name || '  
' || e.last\_name);
- E. MERGE INTO new\_employees cUSING employees e  
ON (c.employee\_id = e.employee\_id)  
WHEN EXISTS THEN  
UPDATE SET
- F. name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT  
value  
S(e.employee\_id, e.first\_name || '  
' || e.last\_name);
- G. MERGE new\_employees c  
FROM employees e ON (c.employee\_id = e.employee\_id)  
WHEN MATCHED THEN  
UPDATE SET
- H. name = e.first\_name || ',' || e.last\_name  
WHEN NOT MATCHED THEN  
INSERT INTO  
new\_employees valueS(e.employee\_id, e.first\_name || '  
' || e.last\_name);

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: this is the correct MERGE statement syntax

Incorrect answer:

B it should MERGE INTO table\_name

C it should be WHEN MATCHED THEN

D it should MERGE INTO table\_name

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-29

#### **QUESTION 14**

Which view should a user query to display the columns associated with the constraints on a table owned by the user?

- A. USER\_CONSTRAINTS
- B. USER\_OBJECTS
- C. ALL\_CONSTRAINTS
- D. USER\_CONS\_COLUMNS
- E. USER\_COLUMNS

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: view the columns associated with the constraint names in the USER\_CONS\_COLUMNS view.

Incorrect answer:

A table to view all constraints definition and names

B show all object name belong to user

C does not display column associated

E no such view

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-25

**QUESTION 15**

The COMMISSION column shows the monthly commission earned by the employee.

Exhibit

EMP_ID	DEPT_ID	COMMISSION
1	10	500
2	20	1000
3	10	
4	10	600
5	30	800
6	30	200
7	10	
8	20	200

Which two tasks would require sub queries or joins in order to be performed in a single step? (Choose two.)

- A. listing the employees who earn the same amount of commission as employee 3
- B. finding the total commission earned by the employees in department 10
- C. finding the number of employees who earn a commission that is higher than the average commission of the company
- D. listing the departments whose average commission is more that 600
- E. listing the employees who do not earn commission and who are working for department 20 in descending order of the employee ID
- F. listing the employees whose annual commission is more than 6000

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 16**

Examine the structure of the STUDENTS table:

STUDENT_ID	NUMBER	NOT NULL, Primary Key
STUDENT_NAME	VARCHAR2(30)	
COURSE_ID	VARCHAR2(10)	NOT NULL
MARKS	NUMBER	
START_DATE	DATE	
FINISH_DATE	DATE	

You need to create a report of the 10 students who achieved the highest ranking in the course INT SQL and who completed the course in the year 1999.

Which SQL statement accomplishes this task?

- A. SELECT student\_id, marks, ROWNUM "Rank"  
FROM students  
WHERE ROWNUM <= 10  
AND finish\_date BETWEEN '01-JAN-99' AND '31-DEC-99'  
AND course\_id = 'INT\_SQL'  
ORDER BY marks DESC;
- B. SELECT student\_id, marks, ROWID "Rank"  
FROM students  
WHERE ROWID <= 10  
AND finish\_date BETWEEN '01-JAN-99' AND '31-DEC-99'  
AND course\_id = 'INT\_SQL'  
ORDER BY marks;
- C. SELECT student\_id, marks, ROWNUM "Rank"  
FROM (SELECT student\_id, marks  
FROM students  
WHERE ROWNUM <= 10  
AND finish\_date BETWEEN '01-JAN-99' AND '31-DEC-99'  
AND course\_id = 'INT\_SQL'  
ORDER BY marks DESC);
- D. SELECT student\_id, marks, ROWNUM "Rank"  
FROM (SELECT student\_id, marks  
FROM students  
WHERE (finish\_date BETWEEN '01-JAN-99' AND '31-DEC-99'  
AND course\_id = 'INT\_SQL'  
ORDER BY marks DESC)  
WHERE ROWNUM <= 10 ;
- E. SELECT student id, marks, ROWNUM "Rank"  
FROM (SELECT student\_id, marks  
FROM students  
ORDER BY marks)  
WHERE ROWNUM <= 10  
AND finish date BETWEEN '01-JAN-99' AND '31-DEC-99'  
AND course\_id = 'INT\_SQL';

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 17**

Evaluate the following SQL statements:

Exhibit:

```
CREATE TABLE orders
(ord_no NUMBER(2) CONSTRAINT ord_pk PRIMARY KEY,
ord_date DATE,
cust_id NUMBER(4));
```

Exhibit:

```
CREATE TABLE ord_items
(ord_no NUMBER(2),
item_no NUMBER(3),
qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200),
expiry_date date CHECK (expiry_date > SYSDATE),
CONSTRAINT it_pk PRIMARY KEY (ord_no,item_no),
CONSTRAINT ord_fk FOREIGN KEY(ord_no) REFERENCES orders(ord_no));
```

The above command fails when executed. What could be the reason?

- A. The BETWEEN clause cannot be used for the CHECK constraint
- B. SYSDATE cannot be used with the CHECK constraint
- C. ORD\_NO and ITEM\_NO cannot be used as a composite primary key because ORD\_NO is also the FOREIGN KEY
- D. The CHECK constraint cannot be placed on columns having the DATE data type

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

CHECK Constraint

The CHECK constraint defines a condition that each row must satisfy. The condition can use the same constructs as the query conditions, with the following exceptions:

References to the CURRVAL, NEXTVAL, LEVEL, and ROWNUM pseudocolumns  
Calls to SYSDATE, UID, USER, and USERENV functions

Queries that refer to other values in other rows

A single column can have multiple CHECK constraints that refer to the column in its definition. There is no limit to the number of CHECK constraints that you can define on a column. CHECK constraints can be defined at the column level or table level.

CREATE TABLE employees

(...

salary NUMBER(8,2) CONSTRAINT emp\_salary\_min

CHECK (salary > 0),

**QUESTION 18**

Evaluate the following SQL statements:

DELETE FROM sales;

There are no other uncommitted transactions on the SALES table.  
Which statement is true about the DELETE statement?

- A. It removes all the rows as well as the structure of the table
- B. It removes all the rows in the table and deleted rows cannot be rolled back
- C. It removes all the rows in the table and deleted rows can be rolled back
- D. It would not remove the rows if the table has a primary key

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 19

Examine the structure of the EMPLOYEES table:

EMPLOYEE_ID	NUMBER	NOT NULL
EMP_NAME	VARCHAR2(30)	
JOB_ID	VARCHAR2(20)	
SAL	NUMBER	
MGR_ID	NUMBER	
DEPARTMENT_ID	NUMBER	

You want to create a SQL script file that contains an INSERT statement. When the script is run, the INSERT statement should insert a row with the specified values into the EMPLOYEES table. The INSERT statement should pass values to the table columns as specified below:

EMPLOYEE_ID:	Next value from the sequence
EMP_ID_SEQEMP_NAME and JOB_ID:	As specified by the user during run time, through substitution variables
SAL:	2000
MGR_ID:	No value
DEPARTMENT_ID:	Supplied by the user during run time through substitution variable. The INSERT statement should fail if the user supplies a value other than 20 or 50.

Which INSERT statement meets the above requirements?

- A. INSERT INTO employees  
VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- B. INSERT INTO employees  
VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid',  
2000, NULL, &did IN (20,50));
- C. INSERT INTO (SELECT \*  
FROM employees  
WHERE department\_id IN (20,50))

VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);

- D. INSERT INTO (SELECT \*  
FROM employees  
WHERE department\_id IN (20,50)  
WITH CHECK OPTION)  
VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- E. INSERT INTO (SELECT \*  
FROM employees  
WHERE (department\_id = 20 AND  
department\_id = 50)  
WITH CHECK OPTION )  
VALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 20

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

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

**Correct Answer:** AE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 21

Which two statements are true about sequences created in a single instance database? (Choose two.)

- A. CURRVAL is used to refer to the last sequence number that has been generated
- B. DELETE <sequencename> would remove a sequence from the database
- C. The numbers generated by a sequence can be used only for one table
- D. When the MAXVALUE limit for a sequence is reached, you can increase the MAXVALUE limit by using the ALTER SEQUENCE statement
- E. When a database instance shuts down abnormally, the sequence numbers that have been cached but not used would be available once again when the database instance is restarted

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Gaps in the Sequence

Although sequence generators issue sequential numbers without gaps, this action occurs independent of a commit or rollback. Therefore, if you roll back a statement containing a sequence, the number is lost.

Another event that can cause gaps in the sequence is a system crash. If the sequence caches values in memory, those values are lost if the system crashes. Because sequences are not tied directly to tables, the same sequence can be used for multiple tables.

However, if you do so, each table can contain gaps in the sequential numbers.

#### Modifying a Sequence

If you reach the MAXVALUE limit for your sequence, no additional values from the sequence are allocated and you will receive an error indicating that the sequence exceeds the MAXVALUE. To continue to use the sequence, you can modify it by using the ALTER SEQUENCE statement

To remove a sequence, use the DROP statement:

```
DROP SEQUENCE dept_deptid_seq;
```

### QUESTION 22

The ORDERS TABLE belongs to the user OE. OE has granted the SELECT privilege on the ORDERS table to the user HR.

Which statement would create a synonym ORD so that HR can execute the following query successfully?

```
SELECT * FROM ord;
```

- A. CREATE SYNONYM ord FOR orders; This command is issued by OE.
- B. CREATE PUBLIC SYNONYM ord FOR orders; This command is issued by OE.
- C. CREATE SYNONYM ord FOR oe.orders; This command is issued by the database administrator.
- D. CREATE PUBLIC SYNONYM ord FOR oe.orders; This command is issued by the database administrator.

**Correct Answer:** D

**Section:** (none)

#### Explanation

#### Explanation/Reference:

Explanation:

Creating a Synonym for an Object

To refer to a table that is owned by another user, you need to prefix the table name with the name of the user who created it, followed by a period. Creating a synonym eliminates the need to qualify the object name with the schema and provides you with an alternative name for a table, view, sequence, procedure, or other objects. This method can be especially useful with lengthy object names, such as views.

In the syntax:

PUBLIC Creates a synonym that is accessible to all users synonym Is the name of the synonym to be created object Identifies the object for which the synonym is created Guidelines

The object cannot be contained in a package.

A private synonym name must be distinct from all other objects that are owned by the same user. If you try to execute the following command (alternative B, issued by OE):

### QUESTION 23

Evaluate this SQL statement:

```
SELECT e.emp_name, d.dept_name  
FROM employees e  
JOIN departments d  
USING (department_id)  
WHERE d.department_id NOT IN (10,40)  
ORDER BY dept_name;
```

The statement fails when executed. Which change fixes the error?

- A. remove the ORDER BY clause
- B. remove the table alias prefix from the WHERE clause
- C. remove the table alias from the SELECT clause
- D. prefix the column in the USING clause with the table alias

- E. prefix the column in the ORDER BY clause with the table alias
- F. replace the condition  
"d.department\_id NOT IN (10,40)"  
in the WHERE clause with  
"d.department\_id <> 10 AND d.department\_id <> 40"

**Correct Answer:** CE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Prefix the column in the ORDER BY Clause would cause the statement to succeed, assuming that the statement failed because the dept\_name existed in employee & department tables.

Not C: Removing the alias from the columns in the SELECT clause would cause the Statement to fail if the columns existing in both tables.

#### **QUESTION 24**

Examine the statement:

Create synonym emp for hr.employees;

What happens when you issue the statement?

- A. An error is generated.
- B. You will have two identical tables in the HR schema with different names.
- C. You create a table called employees in the HR schema based on you EMP table.
- D. You create an alternative name for the employees table in the HR schema in your own schema.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 25**

Evaluate the following SQL query;

```
SQL> SELECT TRUNC(ROUND(156.00,-1),-1)
      FROM DUAL;
```

What would be the outcome?

- A. 200
- B. 16
- C. 160
- D. 150
- E. 100

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:



#### Function Purpose

ROUND(column|expression, n) Rounds the column, expression, or value to n decimal places or, if n is omitted, no decimal places (If n is negative, numbers to the left of decimal point are rounded.)

TRUNC(column|expression, n) Truncates the column, expression, or value to n decimal places or, if n is omitted, n defaults to zero

#### QUESTION 26

Which two statements are true regarding single row functions? (Choose two.)

- A. They can be nested only to two levels
- B. They always return a single result row for every row of a queried table
- C. Arguments can only be column values or constant
- D. They can return a data type value different from the one that is referenced
- E. They accept only a single argument

**Correct Answer:** BD

**Section:** (none)

#### Explanation

#### Explanation/Reference:

Explanation:

A function is a program written to optionally accept input parameters, perform an operation, or return a single value. A function returns only one value per execution. Three important components form the basis of defining a function. The first is the input parameter list. It specifies zero or more arguments that may be passed to a function as input for processing. These arguments or parameters may be of differing data types, and some are mandatory while others may be optional. The second component is the data type of its resultant value. Upon execution, only one value is returned by the function. The third encapsulates the details of the processing performed by the function and contains the program code that optionally manipulates the input parameters, performs calculations and operations, and generates a return value.

#### QUESTION 27

Which statement is true regarding the UNION operator?

- A. The number of columns selected in all SELECT statements need to be the same
- B. Names of all columns must be identical across all SELECT statements
- C. By default, the output is not sorted
- D. NULL values are not ignored during duplicate checking

**Correct Answer:** A

**Section:** (none)

#### Explanation

#### Explanation/Reference:

Explanation:

The SQL UNION query allows you to combine the result sets of two or more SQL SELECT statements. It removes duplicate rows between the various SELECT statements. Each SQL SELECT statement within the UNION query must have the same number of fields in the result sets with similar data types.

#### QUESTION 28

Which two statements are true regarding working with dates? (Choose two.)

- A. The default internal storage of dates is in the numeric format
- B. The RR date format automatically calculates the century from the SYSDATE function but allows the user to enter the century if required
- C. The default internal storage of dates is in the character format
- D. The RR date format automatically calculates the century from the SYSDATE function and does not allow the

user to enter the century

**Correct Answer:** AB

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Working with Dates

The Oracle Database stores dates in an internal numeric format, representing the century, year, month, day, hours, minutes, and seconds.

The default display and input format for any date is DD-MON-RR.

RR Date Format

The RR date format is similar to the YY element, but you can use it to specify different centuries. Use the RR date format element instead of YY so that the century of the return value varies according to the specified two digit year and the last two digits of the current year. The table in the slide summarizes the behavior of the RR element.

Current Year	Given Date	Interpreted (RR)	Interpreted (YY)
1994	27-OCT-95	1995	1995
1994	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2048	27-OCT-52	1952	2052
2051	27-OCT-47	2147	2047

Note the values shown in the last two rows of the above table. As we approach the middle of the century, then the RR behavior is probably not what you want.

This data is stored internally as follows:

CENTURY YEAR MONTH DAY HOUR MINUTE SECOND 19 87 06 17 17 10 43

## QUESTION 29

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_MARITAL_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)

NEW\_CUSTOMERS is a new table with the columns CUST\_ID, CUST\_NAME and CUST\_CITY that have the same data types and size as the corresponding columns in the CUSTOMERS table.

Evaluate the following INSERT statement:

```
INSERT INTO new_customers (cust_id, cust_name, cust_city)
VALUES(SELECT cust_id,cust_first_name||' '||cust_last_name,cust_city
      FROM customers
      WHERE cust_id > 23004);
```

The INSERT statement fails when executed. What could be the reason?

- A. The VALUES clause cannot be used in an INSERT with a subquery
- B. The total number of columns in the NEW\_CUSTOMERS table does not match the total number of columns in the CUSTOMERS table
- C. The WHERE clause cannot be used in a sub query embedded in an INSERT statement
- D. Column names in the NEW\_CUSTOMERS and CUSTOMERS tables do not match

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Copying Rows from Another Table

Write your INSERT statement with a subquery:

Do not use the VALUES clause.

Match the number of columns in the INSERT clause to those in the subquery. Inserts all the rows returned by the subquery in the table, sales\_reps.

### QUESTION 30

View the Exhibit and examine the description for 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_MARITAL_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)

You want to update the CUST\_CREDIT\_LIMIT column to NULL for all the customers, where CUST\_INCOME\_LEVEL has NULL in the CUSTOMERS table. Which SQL statement will accomplish the task?

- A. UPDATE customers  
SET cust\_credit\_limit = NULL  
WHERE CUST\_INCOME\_LEVEL = NULL;
- B. UPDATE customers  
SET cust\_credit\_limit = NULL  
WHERE cust\_income\_level IS NULL;
- C. UPDATE customers  
SET cust\_credit\_limit = TO\_NUMBER(NULL)  
WHERE cust\_income\_level = TO\_NUMBER(NULL);
- D. UPDATE customers  
SET cust\_credit\_limit = TO\_NUMBER(' ',9999)  
WHERE cust\_income\_level IS NULL;

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 31

Which two statements about sub queries are true? (Choose two.)

- A. A sub query should retrieve only one row.
- B. A sub query can retrieve zero or more rows.
- C. A sub query can be used only in SQL query statements.
- D. Sub queries CANNOT be nested by more than two levels.
- E. A sub query CANNOT be used in an SQL query statement that uses group functions.
- F. When a sub query is used with an inequality comparison operator in the outer SQL statement, the column list in the SELECT clause of the sub query should contain only one column.

**Correct Answer:** BF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: sub query can retrieve zero or more rows, sub query is used with an inequality comparison operator in the outer SQL statement, and the column list in the SELECT clause of the sub query should contain only one column.

Incorrect answer:

A sub query can retrieve zero or more rows

C sub query is not SQL query statement

D sub query can be nested

E group function can be use with sub query

### QUESTION 32

View the Exhibit and examine the structure of the PROMOTIONS table.

Table PROMOTIONS		
Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(6)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY_ID	NOT NULL	NUMBER
PROMO_CATEGORY	NOT NULL	VARCHAR2(30)
PROMO_CATEGORY_ID	NOT NULL	NUMBER
PROMO_COST	NOT NULL	NUMBER(10,2)
PROMO_BEGIN_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

You have to generate a report that displays the promo name and start date for all promos that started after the last promo in the 'INTERNET' category.

Which query would give you the required output?

- A. `SELECT promo_name, promo_begin_date FROM promotions  
WHERE promo_begin_date > ALL (SELECT MAX(promo_begin_date) FROM promotions ) AND  
promo_category = 'INTERNET';`

- B. SELECT promo\_name, promo\_begin\_date FROM promotions  
WHERE promo\_begin\_date IN (SELECT promo\_begin\_date  
FROM promotions  
WHERE promo\_category='INTERNET');
- C. SELECT promo\_name, promo\_begin\_date FROM promotions  
WHERE promo\_begin\_date > ALL (SELECT promo\_begin\_date  
FROM promotions  
WHERE promo\_category = 'INTERNET');
- D. SELECT promo\_name, promo\_begin\_date FROM promotions  
WHERE promo\_begin\_date > ANY (SELECT promo\_begin\_date  
FROM promotions  
WHERE promo\_category = 'INTERNET');

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 33

Which are /SQL\*Plus commands? (Choose all that apply.)

- A. INSERT
- B. UPDATE
- C. SELECT
- D. DESCRIBE
- E. DELETE
- F. RENAME

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Describe is a valid iSQL\*Plus/ SQL\*Plus command.

INSERT, UPDATE & DELETE are SQL DML Statements. A SELECT is an ANSI Standard SQL Statement not an iSQL\*Plus Statement.

RENAME is a DDL Statement.

### QUESTION 34

Which two statements are true regarding the COUNT function? (Choose two.)

- A. COUNT(\*) returns the number of rows including duplicate rows and rows containing NULL value in any of the columns
- B. COUNT(cust\_id) returns the number of rows including rows with duplicate customer IDs and NULL value in the CUST\_ID column
- C. COUNT(DISTINCT inv\_amt) returns the number of rows excluding rows containing duplicates and NULL values in the INV\_AMT column
- D. A SELECT statement using COUNT function with a DISTINCT keyword cannot have a WHERE clause
- E. The COUNT function can be used only for CHAR, VARCHAR2 and NUMBER data types

**Correct Answer: AC**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Using the COUNT Function

The COUNT function has three formats:

COUNT(\*)

COUNT(expr)

COUNT(DISTINCT expr)

COUNT(\*) returns the number of rows in a table that satisfy the criteria of the SELECT statement, including duplicate rows and rows containing null values in any of the columns. If a WHERE clause is included in the SELECT statement, COUNT(\*) returns the number of rows that satisfy the condition in the WHERE clause.

In contrast,

COUNT(expr) returns the number of non-null values that are in the column identified by expr. COUNT(DISTINCT expr) returns the number of unique, non-null values that are in the column identified by expr.

**QUESTION 35**

Examine the description of the EMP\_DETAILS table given below:

Exhibit:

NAME	NULL	TYPE
EMP_ID	NOT NULL	NUMBER
EMP_NAME	NOT NULL	VARCHAR2 (40)
EMP_IMAGE		LONG

Which two statements are true regarding SQL statements that can be executed on the EMP\_DETAIL table? (Choose two.)

- A. An EMP\_IMAGE column can be included in the GROUP BY clause
- B. You cannot add a new column to the table with LONG as the data type
- C. An EMP\_IMAGE column cannot be included in the ORDER BY clause
- D. You can alter the table to include the NOT NULL constraint on the EMP\_IMAGE column

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

LONG Character data in the database character set, up to 2GB. All the functionality of LONG (and more) is provided by CLOB; LONGs should not be used in a modern database, and if your database has any columns of this type they should be converted to CLOB. There can only be one LONG column in a table.

Guidelines

A LONG column is not copied when a table is created using a subquery. A LONG column cannot be included in a GROUP BY or an ORDER BY clause.

Only one LONG column can be used per table.

No constraints can be defined on a LONG column.

You might want to use a CLOB column rather than a LONG column.

**QUESTION 36**

Which CREATE TABLE statement is valid?

- A. CREATE TABLE ord\_details  
(ord\_no NUMBER(2) PRIMARY KEY,  
item\_no NUMBER(3) PRIMARY KEY,  
ord\_date DATE NOT NULL);
- B. CREATE TABLE ord\_details

- ```
(ord_no NUMBER(2) UNIQUE, NOT NULL,
item_no NUMBER(3),
ord_date DATE DEFAULT SYSDATE NOT NULL);
```
- C. CREATE TABLE ord\_details  
(ord\_no NUMBER(2) ,  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT NOT NULL,  
CONSTRAINT ord\_uq UNIQUE (ord\_no),  
CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));
- D. CREATE TABLE ord\_details  
(ord\_no NUMBER(2),  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT SYSDATE NOT NULL,  
CONSTRAINT ord\_pk PRIMARY KEY (ord\_no, item\_no));

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

PRIMARY KEY Constraint

A PRIMARY KEY constraint creates a primary key for the table. Only one primary key can be created for each table. The PRIMARY KEY constraint is a column or a set of columns that uniquely identifies each row in a table. This constraint enforces the uniqueness of the column or column combination and ensures that no column that is part of the primary key can contain a null value.

Note: Because uniqueness is part of the primary key constraint definition, the Oracle server enforces the uniqueness by implicitly creating a unique index on the primary key column or columns.

**QUESTION 37**

See the exhibit and examine the structure of the CUSTOMERS and GRADES tables:

| CUSTOMERS         |          |              |
|-------------------|----------|--------------|
| Name              | Null?    | Type         |
| CUSTNO            | NOT NULL | NUMBER(2)    |
| CUSTNAME          |          | VARCHAR2(10) |
| CUSTADDRESS       |          | VARCHAR2(20) |
| CUST_CREDIT_LIMIT |          | NUMBER(5)    |

| GRADES   |          |             |
|----------|----------|-------------|
| Name     | Null?    | Type        |
| GRADE    | NOT NULL | VARCHAR2(1) |
| STARTVAL |          | NUMBER(5)   |
| ENDVAL   |          | NUMBER(5)   |

You need to display names and grades of customers who have the highest credit limit. Which two SQL statements would accomplish the task? (Choose two.)

- A. SELECT custname, grade  
FROM customers, grades  
WHERE (SELECT MAX(cust\_credit\_limit)  
FROM customers) BETWEEN startval and endval;

- B. SELECT custname, grade  
FROM customers, grades  
WHERE (SELECT MAX(cust\_credit\_limit)  
FROM customers) BETWEEN startval and endval  
AND cust\_credit\_limit BETWEEN startval AND endval;
- C. SELECT custname, grade  
FROM customers, grades  
WHERE cust\_credit\_limit = (SELECT MAX(cust\_credit\_limit)  
FROM customers)  
AND cust\_credit\_limit BETWEEN startval AND endval;
- D. SELECT custname, grade  
FROM customers , grades  
WHERE cust\_credit\_limit IN (SELECT MAX(cust\_credit\_limit)  
FROM customers)  
AND MAX(cust\_credit\_limit) BETWEEN startval AND endval;

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 38

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_MARITAL_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?

- A. SELECT NVL(cust\_credit\_limit,'Not Available')\*.15 "NEW CREDIT" FROM customers;
- B. SELECT NVL(cust\_credit\_limit\*.15,'Not Available') "NEW CREDIT" FROM customers;
- C. SELECT TO\_CHAR(NVL(cust\_credit\_limit\*.15,'Not Available')) "NEW CREDIT" FROM customers;
- D. SELECT NVL(TO\_CHAR(cust\_credit\_limit\*.15),'Not Available') "NEW CREDIT" FROM customers;

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

NVL Function



Converts a null value to an actual value:

Data types that can be used are date, character, and number.

Data types must match:

NVL(commission\_pct,0)

NVL(hire\_date,'01-JAN-97')

NVL(job\_id,'No Job Yet')

### QUESTION 39

You need to calculate the number of days from 1st Jan 2007 till date:

Dates are stored in the default format of dd-mm-rr.

Which two SQL statements would give the required output? (Choose two.)

- A. SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;
- B. SELECT TO\_DATE(SYSDATE,'DD/MONTH/YYYY')-'01/JANUARY/2007' FROM DUAL;
- C. SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL
- D. SELECT SYSDATE - '01-JAN-2007' FROM DUAL
- E. SELECT TO\_CHAR(SYSDATE,'DD-MON-YYYY')-'01-JAN-2007' FROM DUAL;

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 40

Which two are true about aggregate functions? (Choose two.)

- A. You can use aggregate functions in any clause of a SELECT statement.
- B. You can use aggregate functions only in the column list of the select clause and in the WHERE clause of a SELECT statement.
- C. You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns.
- D. You can pass column names, expressions, constants, or functions as parameter to an aggregate function.
- E. You can use aggregate functions on a table, only by grouping the whole table as one single group.
- F. You cannot group the rows of a table by more than one column while using aggregate functions.

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 41

See the structure of the PROGRAMS table:

| Name       | Null?    | Type        |
|------------|----------|-------------|
| -----      | -----    | -----       |
| PROG_ID    | NOT NULL | NUMBER(3)   |
| PROG_COST  |          | NUMBER(8,2) |
| START_DATE | NOT NULL | DATE        |
| END_DATE   |          | DATE        |

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT NVL(ADD\_MONTHS(END\_DATE,1),SYSDATE)  
FROM programs;
- B. SELECT TO\_DATE(NVL(SYSDATE-END\_DATE,SYSDATE))  
FROM programs;
- C. SELECT NVL(MONTHS\_BETWEEN(start\_date,end\_date),'Ongoing')  
FROM programs;
- D. SELECT NVL(TO\_CHAR(MONTHS\_BETWEEN(start\_date,end\_date)),'Ongoing') FROM programs;

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

NVL Function

Converts a null value to an actual value:

Data types that can be used are date, character, and number.

Data types must match:

NVL(commission\_pct,0)

NVL(hire\_date,'01-JAN-97')

NVL(job\_id,'No Job Yet')

MONTHS\_BETWEEN(date1, date2): Finds the number of months between date1 and date2 . The result can be positive or negative. If date1 is later than date2, the result is positive; if date1 is earlier than date2, the result is negative. The noninteger part of the result represents a portion of the month.

MONTHS\_BETWEEN returns a numeric value. - answer C NVL has different datatypes - numeric and strings, which is not possible!

The data types of the original and if null parameters must always be compatible. They must either be of the same type, or it must be possible to implicitly convert if null to the type of the original parameter. The NVL function returns a value with the same data type as the original parameter.

#### QUESTION 42

You issue the following command to drop the PRODUCTS table:

SQL>DROP TABLE products;

What is the implication of this command? (Choose all that apply.)

- A. All data in the table are deleted but the table structure will remain
- B. All data along with the table structure is deleted
- C. All views and synonyms will remain but they are invalidated
- D. The pending transaction in the session is committed
- E. All indexes on the table will remain but they are invalidated

**Correct Answer:** BCD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 43

Exhibit contains the structure of PRODUCTS table:

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

Evaluate the following query:

```
SQL> SELECT prod_name
FROM products
WHERE prod_id IN (SELECT prod_id FROM products
                  WHERE prod_list_price =
                    (SELECT MAX(prod_list_price) FROM products
                     WHERE prod_list_price <
                      (SELECT MAX(prod_list_price) FROM products))));
```

What would be the outcome of executing the above SQL statement?

- A. It produces an error
- B. It shows the names of products whose list price is the second highest in the table.
- C. It shown the names of all products whose list price is less than the maximum list price
- D. It shows the names of all products in the table

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 44

See the Exhibits and examine the structures of PRODUCTS, SALES and CUSTOMERS table:

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

| Table SALES   |          |              |
|---------------|----------|--------------|
| Name          | Null?    | Type         |
| PROD_ID       | NOT NULL | NUMBER       |
| CUST_ID       | NOT NULL | NUMBER       |
| TIME_ID       | NOT NULL | DATE         |
| CHANNEL_ID    | NOT NULL | NUMBER       |
| PROMO_ID      | NOT NULL | NUMBER       |
| QUANTITY_SOLD | NOT NULL | NUMBER(10,2) |

| 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_MARITAL_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) |

You issue the following query:

```
SQL>SELECT p.prod_id,prod_name,prod_list_price,
        quantity_sold,cust_last_name
FROM products p NATURAL JOIN sales s NATURAL JOIN customers c
WHERE prod_id =148;
```

Which statement is true regarding the outcome of this query?

- A. It produces an error because the NATURAL join can be used only with two tables
- B. It produces an error because a column used in the NATURAL join cannot have a qualifier
- C. It produces an error because all columns used in the NATURAL join should have a qualifier
- D. It executes successfully

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Creating Joins with the USING Clause

Natural joins use all columns with matching names and data types to join the tables. The USING clause can be used to specify only those columns that should be used for an equijoin.

The Natural JOIN USING Clause

The format of the syntax for the natural JOIN USING clause is as follows:

```
SELECT table1.column, table2.column
```

```
FROM table1
```

```
JOIN table2 USING (join_column1, join_column2...);
```

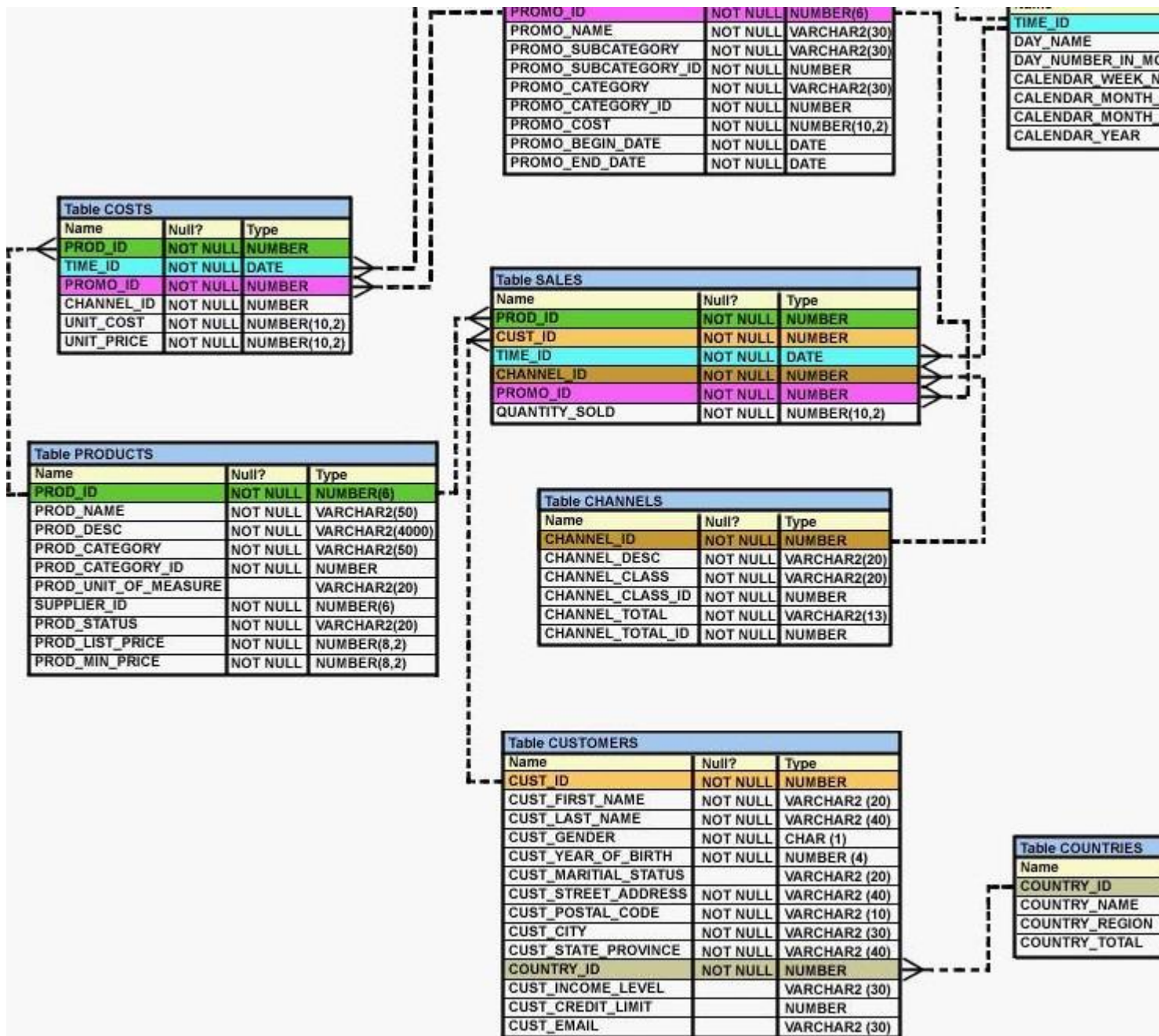
While the pure natural join contains the NATURAL keyword in its syntax, the JOIN...USING syntax does not.

An error is raised if the keywords NATURAL and USING occur in the same join clause. The JOIN...USING clause allows one or more equijoin columns to be explicitly specified in brackets after the USING keyword. This avoids the shortcomings associated with the pure natural join. Many situations demand that tables be joined only on certain columns, and this format caters to this requirement.

#### QUESTION 45

You work as a database administrator at ABC.com. You study the exhibit carefully.

Exhibit:



You want to create a SALE\_PROD view by executing the following SQL statements:

```
CREATE VIEW sale_prod
AS SELECT p.prod_id, cust_id, SUM(quantity_sold) "Quantity", SUM(prod_list_price) "Price"
FROM products p, sales s
WHERE p.prod_id=s.prod_id
GROUP BY p.prod_id, cust_id;
```

Which statement is true regarding the execution of the above statement?

- A. The view will be created and you can perform DLM operations on the view
- B. The view will not be created because the join statements are not allowed for creating a view

- C. The view will not be created because the GROUP BY clause is not allowed for creating a view
- D. The view will be created but no DML operations will be allowed on the view

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Rules for Performing DML Operations on a View

You cannot add data through a view if the view includes:

Group functions

A GROUP BY clause

The DISTINCT keyword

The pseudocolumn ROWNUM keyword

Columns defined by expressions

NOT NULL columns in the base tables that are not selected by the view

#### **QUESTION 46**

Which three statements are true regarding the data types in Oracle Database 10g/11g? (Choose three.)

- A. The BLOB data type column is used to store binary data in an operating system file
- B. The minimum column width that can be specified for a VARCHAR2 data type column is one
- C. A TIMESTAMP data type column stores only time values with fractional seconds
- D. The value for a CHAR data type column is blank-padded to the maximum defined column width
- E. Only One LONG column can be used per table

**Correct Answer:** BDE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

LONG Character data in the database character set, up to 2GB. All the functionality of LONG (and more) is provided by CLOB; LONGs should not be used in a modern database, and if your database has any columns of this type they should be converted to CLOB. There can only be one LONG column in a table.

DVARCHAR2 Variable-length character data, from 1 byte to 4KB. The data is stored in the database character set. The VARCHAR2 data type must be qualified with a number indicating the maximum length of the column.

If a value is inserted into the column that is less than this, it is not a problem: the value will only take up as much space as it needs. If the value is longer than this maximum, the INSERT will fail with an error.

VARCHAR2(size)

Variable-length character data (A maximum size must be specified: minimum size is 1; maximum size is 4,000.)

BLOB Like CLOB, but binary data that will not undergo character set conversion by Oracle Net. BFILE A locator pointing to a file stored on the operating system of the database server. The size of the files is limited to 4GB.

TIMESTAMP This is length zero if the column is empty, or up to 11 bytes, depending on the precision specified. Similar to DATE, but with precision of up to 9 decimal places for the seconds, 6 places by default.

#### **QUESTION 47**

Which three are true? (Choose three.)

- A. A MERGE statement is used to merge the data of one table with data from another.
- B. A MERGE statement replaces the data of one table with that of another.
- C. A MERGE statement can be used to insert new rows into a table.
- D. A MERGE statement can be used to update existing rows in a table.

**Correct Answer:** ACD



**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation: The MERGE Statement allows you to conditionally insert or update data in a table. If the rows are present in the target table which match the join condition, they are updated if the rows are not present they are inserted into the target table

**QUESTION 48**

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

- A. A sub query that defines a view cannot include the GROUP BY clause
- B. A view is created with the sub query having the DISTINCT keyword can be updated
- C. A Data Manipulation Language (DML) operation can be performed on a view that is created with the sub query having all the NOT NULL columns of a table
- D. A view that is created with the sub query having the pseudo column ROWNUM keyword cannot be updated

**Correct Answer: CD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Rules for Performing DML Operations on a View

You cannot add data through a view if the view includes:

Group functions

A GROUP BY clause

The DISTINCT keyword

The pseudocolumn ROWNUM keyword

Columns defined by expressions

NOT NULL columns in the base tables that are not selected by the view

**QUESTION 49**

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

EMPLOYEES

EMPLOYEE\_ID NUMBER Primary Key

FIRST\_NAME VARCHAR2(25)

LAST\_NAME VARCHAR2(25)

HIRE\_DATE DATE

NEW\_EMPLOYEES

EMPLOYEE\_ID NUMBER Primary Key

NAME VARCHAR2 (60)

Which DELETE statement is valid?

- A. DELETE FROM employees  
WHERE employee\_id = (SELECT employee\_id FROM employees);
- B. DELETE \* FROM employees  
WHERE employee\_id=(SELECT employee\_id FROM new\_employees);

- C. DELETE FROM employees  
WHERE employee\_id IN (SELECT employee\_id  
FROM new\_employees  
WHERE name = 'Carrey');
- D. DELETE \* FROM employees  
WHERE employee\_id IN (SELECT employee\_id  
FROM new\_employees  
WHERE name = 'Carrey');

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 50

View the Exhibits and examine the structures of the PROMOTIONS and SALES tables.

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

| Table SALES   |          |              |
|---------------|----------|--------------|
| Name          | Null?    | Type         |
| PROD_ID       | NOT NULL | NUMBER       |
| CUST_ID       | NOT NULL | NUMBER       |
| TIME_ID       | NOT NULL | DATE         |
| CHANNEL_ID    | NOT NULL | NUMBER       |
| PROMO_ID      | NOT NULL | NUMBER       |
| QUANTITY SOLD | NOT NULL | NUMBER(10,2) |

Evaluate the following SQL statements:

```
SQL>SELECT p.promo_id, p.promo_name, s.prod_id
FROM sales s RIGHT OUTER JOIN promotions p
ON (s.promo_id = p.promo_id);
```

Which statement is true regarding the output of the above query?

- A. It gives details of product IDs that have been sold irrespective of whether they had a promo or not
- B. It gives the details of promos for which there have been no sales
- C. It gives the details of promos for which there have been sales
- D. It gives details of all promos irrespective of whether they have resulted in a sale or not

**Correct Answer: D**

**Section: (none)**

**Explanation**



**Explanation/Reference:**

**QUESTION 51**

Examine the structure of the EMPLOYEES table:

|             |               |             |
|-------------|---------------|-------------|
| EMPLOYEE_ID | NUMBER        | Primary Key |
| FIRST_NAME  | VARCHAR2 (25) |             |
| LAST_NAME   | VARCHAR2 (25) |             |
| HIRE_DATE   | DATE          |             |

Which UPDATE statement is valid?

- A. UPDATE employees  
SET first\_name = 'John'  
SET last\_name = 'Smith'  
WHERE employee\_id = 180;
- B. UPDATE employees  
SET first\_name = 'John',  
SET last\_name = 'Smoth'  
WHERE employee\_id = 180;
- C. UPDATE employee  
SET first\_name = 'John'  
AND last\_name = 'Smith'  
WHERE employee\_id = 180;
- D. UPDATE employee  
SET first\_name = 'John', last\_name = 'Smith'  
WHERE employee\_id = 180;

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 52**

View the Exhibit and evaluate structures of the SALES, PRODUCTS, and COSTS tables.

| Table SALES   |          |              |
|---------------|----------|--------------|
| Name          | Null?    | Type         |
| PROD_ID       | NOT NULL | NUMBER       |
| CUST_ID       | NOT NULL | NUMBER       |
| TIME_ID       | NOT NULL | DATE         |
| CHANNEL_ID    | NOT NULL | NUMBER       |
| PROMO_ID      | NOT NULL | NUMBER       |
| QUANTITY_SOLD | NOT NULL | NUMBER(10,2) |

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

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

Evaluate the following SQL statements:

```
SQL>SELECT prod_id FROM products
INTERSECT
SELECT prod_id FROM sales
MINUS
SELECT prod_id FROM costs;
```

Which statement is true regarding the above compound query?

- A. It shows products that have a cost recorded irrespective of sales
- B. It shows products that were sold and have a cost recorded
- C. It shows products that were sold but have no cost recorded
- D. It reduces an error

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 53

The PRODUCTS table has the following structure:

| Name             | Null?    | Type         |
|------------------|----------|--------------|
| PROD_ID          | NOT NULL | NUMBER(4)    |
| PROD_NAME        |          | VARCHAR2(25) |
| PROD_EXPIRY_DATE |          | DATE         |

Evaluate the following two SQL statements:

```
SQL>SELECT prod_id, NVL2(prod_expiry_date, prod_expiry_date + 15, '') FROM products;
SQL>SELECT prod_id, NVL(prod_expiry_date, prod_expiry_date + 15) FROM products;
```

Which statement is true regarding the outcome?

- A. Both the statements execute and give the same result
- B. Both the statements execute and give different results
- C. Only the second SQL statement executes successfully
- D. Only the first SQL statement executes successfully

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Using the NVL2 Function

The NVL2 function examines the first expression. If the first expression is not null, the NVL2 function returns the second expression. If the first expression is null, the third expression is returned.

Syntax

NVL2(expr1, expr2, expr3)

In the syntax:

expr1 is the source value or expression that may contain a null  
expr2 is the value that is returned if expr1 is not null

expr3 is the value that is returned if expr1 is null

#### QUESTION 54

Which statements are correct regarding indexes? (Choose all that apply.)

- A. For each data manipulation language (DML) operation performed, the corresponding indexes are automatically updated.
- B. A nondeferred PRIMARY KEY or UNIQUE KEY constraint in a table automatically creates a unique index.
- C. A FOREIGN KEY constraint on a column in a table automatically creates a non unique key
- D. When a table is dropped, the corresponding indexes are automatically dropped

**Correct Answer: ABD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 55

You work as a database administrator at ABC.com. You study the exhibit carefully.

Exhibit:

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

Which two SQL statements would execute successfully? (Choose two.)

- A. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE TO\_CHAR(promo\_end\_date, 'yyyy') > '2000';
- B. SELECT promo\_begin\_date  
FROM promotions  
WHERE TO\_CHAR(promo\_begin\_date, 'mon dd yy')='jul 01 98';
- C. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8));
- D. SELECT TO\_CHAR(promo\_begin\_date, 'dd/month')  
FROM promotions  
WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'));

**Correct Answer:** AB

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 56

Examine these statements:

```
CREATE ROLE registrar;
GRANT UPDATE ON student_grades TO registrar;
GRANT registrar to user1, user2, user3;
```

What does this set of SQL statements do?

- A. The set of statements contains an error and does not work.
- B. It creates a role called REGISTRAR, adds the MODIFY privilege on the STUDENT\_GRADES object to the role, and gives the REGISTRAR role to three users.
- C. It creates a role called REGISTRAR, adds the UPDATE privilege on the STUDENT\_GRADES object to the role, and gives the REGISTRAR role to three users.
- D. It creates a role called REGISTRAR, adds the UPDATE privilege on the STUDENT\_GRADES object to the role, and creates three users with the role.
- E. It creates a role called REGISTRAR, adds the UPDATE privilege on three users, and gives the REGISTRAR role to the STUDENT\_GRADES object.
- F. It creates a role called STUDENT\_GRADES, adds the UPDATE privilege on three users, and gives the UPDATE role to the registrar.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation: the statement will create a role call REGISTRAR, grant UPDATE on student\_grades to registrar, grant the role to user1,user2 and user3.

Incorrect answer:

A the statement does not contain error

B there is no MODIFY privilege

D statement does not create 3 users with the role

E privilege is grant to role then grant to user

F privilege is grant to role then grant to user

**QUESTION 57**

Examine the structure of the MARKS table:

Exhibit:

| Name         | Null?    | Type         |
|--------------|----------|--------------|
| -----        | -----    | -----        |
| STUDENT_ID   | NOT NULL | VARCHAR2(4)  |
| STUDENT_NAME |          | VARCHAR2(25) |
| SUBJECT1     |          | NUMBER(3)    |
| SUBJECT2     |          | NUMBER(3)    |
| SUBJECT3     |          | NUMBER(3)    |

Which two statements would execute successfully? (Choose two.)

- A. SELECT student\_name,subject1  
FROM marks  
WHERE subject1 > AVG(subject1);
- B. SELECT student\_name,SUM(subject1)  
FROM marks  
WHERE student\_name LIKE 'R%';
- C. SELECT SUM(subject1+subject2+subject3)  
FROM marks  
WHERE student\_name IS NULL;
- D. SELECT SUM(DISTINCT NVL(subject1,0)), MAX(subject1)  
FROM marks  
WHERE subject1 > subject2;

**Correct Answer: CD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 58**

You are currently located in Singapore and have connected to a remote database in Chicago.  
You issue the following command:

Exhibit:

```
SQL> SELECT ROUND(SYSDATE-promo_begin_date,0)
      FROM promotions
      WHERE (SYSDATE-promo_begin_date)/365 > 2;
```

PROMOTIONS is the public synonym for the public database link for the PROMOTIONS table.

What is the outcome?

- A. Number of days since the promo started based on the current Singapore data and time.
- B. An error because the ROUND function specified is invalid
- C. An error because the WHERE condition specified is invalid
- D. Number of days since the promo started based on the current Chicago data and time

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 59

Evaluate the following two queries:

Exhibit:

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

Exhibit:

```
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?

- A. Performance would improve in query 2 only if there are null values in the CUST\_CREDIT\_LIMIT column
- B. Performance would degrade in query 2
- C. There would be no change in performance
- D. Performance would improve in query 2

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Note: The IN operator is internally evaluated by the Oracle server as a set of OR conditions, such as a=value1 or a=value2 or a=value3. Therefore, using the IN operator has no performance benefits and is used only for logical simplicity.

#### QUESTION 60

View the Exhibit and examine the structure of the CUSTOMERS and CUST\_HISTORY tables.

| CUSTOMERS    |          |              |
|--------------|----------|--------------|
| Name         | Null?    | Type         |
| -----        | -----    | -----        |
| CUST_ID      | NOT NULL | NUMBER(4)    |
| CUST_NAME    |          | VARCHAR2(20) |
| CUST_ADDRESS |          | VARCHAR2(30) |
| CUST_CITY    |          | VARCHAR2(20) |
|              |          |              |
| CUST_HISTORY |          |              |
| Name         | Null?    | Type         |
| -----        | -----    | -----        |
| CUST_ID      | NOT NULL | NUMBER(4)    |
| CUST_NAME    |          | VARCHAR2(20) |
| CUST_CITY    |          | VARCHAR2(20) |
| CHANGE_DATE  |          | DATE         |

The CUSTOMERS table contains the current location of all currently active customers. The CUST\_HISTORY table stores historical details relating to any changes in the location of all current as well as previous customers who are no longer active with the company.

You need to find those customers who have never changed their address. Which SET operator would you use to get the required output?

- A. INTERSECT
- B. UNION ALL
- C. MINUS
- D. UNION

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 61

You created an ORDERS table with the following description:

Exhibit:

| Name       | Null     | Type         |
|------------|----------|--------------|
| -----      | -----    | -----        |
| ORD_ID     | NOT NULL | NUMBER(2)    |
| CUST_ID    | NOT NULL | NUMBER(3)    |
| ORD_DATE   | NOT NULL | DATE         |
| ORD_AMOUNT | NOT NULL | NUMBER(10,2) |

You inserted some rows in the table. After some time, you want to alter the table by creating the PRIMARY KEY constraint on the ORD\_ID column.

Which statement is true in this scenario?

- A. You cannot add a primary key constraint if data exists in the column
- B. You can add the primary key constraint even if data exists, provided that there are no duplicate values
- C. The primary key constraint can be created only at the time of table creation
- D. You cannot have two constraints on one column

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 62**

Which object privileges can be granted on a view?

- A. none
- B. DELETE, INSERT, SELECT
- C. ALTER, DELETE, INSERT, SELECT
- D. DELETE, INSERT, SELECT, UPDATE

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE.

Incorrect answer:

A Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE  
B Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE  
C Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-12

#### **QUESTION 63**

When does a transaction complete? (Choose all that apply.)

- A. When a PL/SQL anonymous block is executed
- B. When a DELETE statement is executed
- C. When a data definition language statement is executed
- D. When a TRUNCATE statement is executed after the pending transaction
- E. When a ROLLBACK command is executed

**Correct Answer:** CDE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 64**

View the Exhibit and examine the structure of the CUSTOMERS table.

Exhibit:



| 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_MARITAL_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) |

you issue the following SQL statement on the CUSTOMERS table to display the customers who are in the same country as customers with the last name 'king' and whose credit limit is less than the maximum credit limit in countries that have customers with the last name 'king'.

```
SQL> SELECT cust_id,cust_last_name
FROM customers
WHERE country_id IN(SELECT country_id
                     FROM customers
                     WHERE cust_last_name ='King')
AND cust_credit_limit < (SELECT MAX(cust_credit_limit)
                        FROM customers
                        WHERE country_id IN(SELECT country_id
   FROM customers
   WHERE cust_last_name='King'));
```

Which statement is true regarding the outcome of the above query?

- A. It produces an error and the < operator should be replaced by < ANY to get the required output
- B. It produces an error and the IN operator should be replaced by = in the WHERE clause of the main query to get the required output
- C. It executes and shows the required result
- D. It produces an error and the < operator should be replaced by < ALL to get the required output

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 65

You need to create a table for a banking application. One of the columns in the table has the following requirements:

You want a column in the table to store the duration of the credit period The data in the column should be stored in a format such that it can be easily added and subtracted with DATE data type without using conversion

The maximum period of the credit provision in the application is 30 days the interest has to be calculated for the number of days an individual has taken a credit for

Which data type would you use for such a column in the table?

- A. INTERVAL YEAR TO MONTH
- B. NUMBER
- C. TIMESTAMP
- D. DATE
- E. INTERVAL DAY TO SECOND

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 66

The STUDENT\_GRADES table has these columns:

|              |             |
|--------------|-------------|
| STUDENT_ID   | NUMBER(12)  |
| SEMESTER_END | DATE        |
| GPA          | NUMBER(4,3) |

Which statement finds students who have a grade point average (GPA) greater than 3.0 for the calendar year 2001?

- A. SELECT student\_id, gpa  
FROM student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001' OR gpa > 3.;
- B. SELECT student\_id, gpa  
FROM student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001' AND gpa gt 3.0;
- C. SELECT student\_id, gpa  
FROM student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001' AND gpa > 3.0;
- D. SELECT student\_id, gpa  
FROM student\_grades  
WHERE semester\_end BETWEEN '01-JAN-2001' AND '31-DEC-2001' OR gpa > 3.0;
- E. SELECT student\_id, gpa  
FROM student\_grades  
WHERE semester\_end > '01-JAN-2001' OR semester\_end < '31-DEC-2001' AND gpa >= 3.0;

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 67

View the Exhibit and examine the data in the PROMOTIONS table.

## PROMOTIONS

| PROMO_ID | PROMO_CATEGORY | PROMO_SUBCATEGORY |
|----------|----------------|-------------------|
| 506      | magazine       | discount          |
| 507      | TV             | general advt      |
| 508      | newspaper      | discount          |
| 509      | post           | general advt      |
| 510      | post           | discount          |
| 511      | radio          | general advt      |
| 512      | newspaper      | general advt      |
| 513      | newspaper      | discount          |
| 514      | magazine       | general advt      |
| 515      | newspaper      | discount          |
| 516      | newspaper      | general advt      |

You need to display all promo categories that do not have 'discount' in their subcategory. Which two SQL statements give the required result? (Choose two.)

- A. 

```
SELECT promo_category
FROM promotions
MINUS
SELECT promo_category
FROM promotions
WHERE promo_subcategory = 'discount';
```
- B. 

```
SELECT promo_category
FROM promotions
INTERSECT
SELECT promo_category
FROM promotions
WHERE promo_subcategory = 'discount';
```
- C. 

```
SELECT promo_category
FROM promotions
MINUS
SELECT promo_category
FROM promotions
WHERE promo_subcategory <> 'discount';
```
- D. 

```
SELECT promo_category
FROM promotions
INTERSECT
SELECT promo_category
FROM promotions
WHERE promo_subcategory <> 'discount';
```

**Correct Answer:** AD

**Section:** (none)

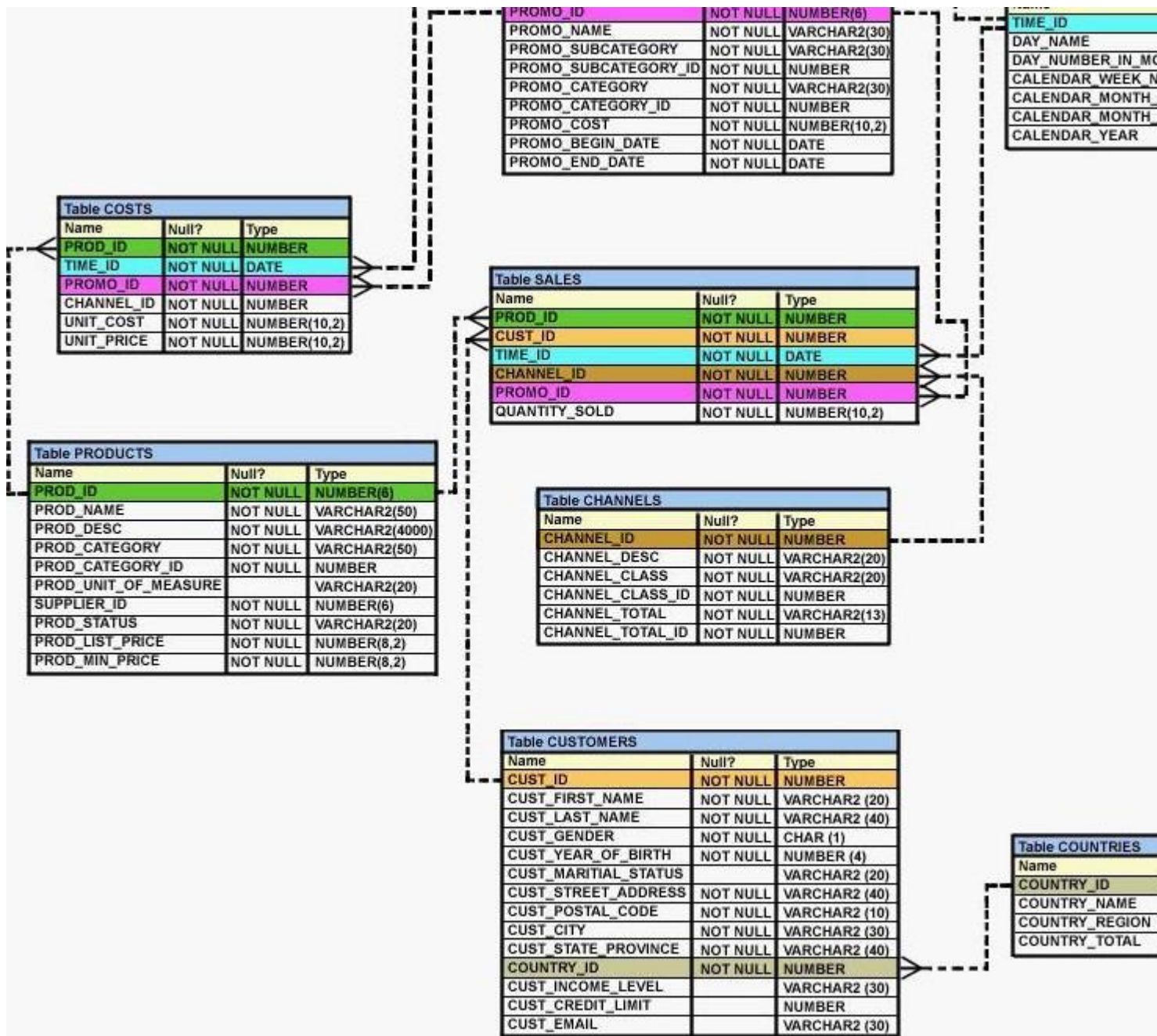
**Explanation**

**Explanation/Reference:**

### QUESTION 68

You work as a database administrator at ABC.com. You study the exhibit carefully.

Exhibit:



and examine the structure of CUSTOMERS AND SALES tables:

Evaluate the following SQL statement:

Exhibit:

```
UPDATE (SELECT prod_id, cust_id, quantity_sold, time_id
        FROM sales)
SET time_id = '22-MAR-2007'
WHERE cust_id = (SELECT cust_id
                  FROM customers
                  WHERE cust_last_name = 'Roberts' AND
                     credit_limit = 600);
```

Which statement is true regarding the execution of the above UPDATE statement?

- A. It would not execute because the SELECT statement cannot be used in place of the table name
- B. It would execute and restrict modifications to only the column specified in the SELECT statement
- C. It would not execute because a sub query cannot be used in the WHERE clause of an UPDATE statement
- D. It would not execute because two tables cannot be used in a single UPDATE statement

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 69

See the Exhibit and examine the structure of the PROMOTIONS table:

Exhibit:

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

Using the PROMOTIONS table, you need to find out the average cost for all promos in the range \$0-2000 and \$2000-5000 in category A.

You issue the following SQL statements:

Exhibit:

```
SQL>SELECT AVG(CASE
                WHEN promo_cost BETWEEN 0 AND 2000 AND promo_category='A'
                THEN promo_cost
                ELSE null END) "CAT_2000A",
        AVG(CASE
                WHEN promo_cost BETWEEN 2001 AND 5000 AND promo_category='A'
                THEN promo_cost
                ELSE null END) "CAT_5000A"
FROM promotions;
```

What would be the outcome?

- A. It generates an error because multiple conditions cannot be specified for the WHEN clause
- B. It executes successfully and gives the required result
- C. It generates an error because CASE cannot be used with group functions
- D. It generates an error because NULL cannot be specified as a return value

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

CASE Expression

Facilitates conditional inquiries by doing the work of an

IF-THEN-ELSE statement:

CASE expr WHEN comparison\_expr1 THEN return\_expr1

[WHEN comparison\_expr2 THEN return\_expr2

WHEN comparison\_exprn THEN return\_exprn

ELSE else\_expr]

END

## QUESTION 70

Which two statements are true regarding the USING and ON clauses in table joins? (Choose two.)

- A. The ON clause can be used to join tables on columns that have different names but compatible data types
- B. A maximum of one pair of columns can be joined between two tables using the ON clause
- C. Both USING and ON clause can be used for equijoins and nonequijoins
- D. The WHERE clause can be used to apply additional conditions in SELECT statement containing the ON or the USING clause

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Creating Joins with the USING Clause

If several columns have the same names but the data types do not match, use the USING clause to specify the columns for the equijoin.

Use the USING clause to match only one column when more than one column matches.

The NATURAL JOIN and USING clauses are mutually exclusive

Using Table Aliases with the USING clause

When joining with the USING clause, you cannot qualify a column that is used in the USING clause itself.

Furthermore, if that column is used anywhere in the SQL statement, you cannot alias it. For example, in the query mentioned in the slide, you should not alias the location\_id column in the WHERE clause because the

column is used in the USING clause. The columns that are referenced in the USING clause should not have a qualifier (table name or alias) anywhere in the SQL statement.

#### Creating Joins with the ON Clause

The join condition for the natural join is basically an equijoin of all columns with the same name. Use the ON clause to specify arbitrary conditions or specify columns to join. **ANSWER C** The join condition is separated from other search conditions. **ANSWER D** The ON clause makes code easy to understand.

#### QUESTION 71

Examine the structure of the EMPLOYEES table:

|             |              |             |
|-------------|--------------|-------------|
| EMPLOYEE_ID | NUMBER       | Primary Key |
| FIRST_NAME  | VARCHAR2(25) |             |
| LAST_NAME   | VARCHAR2(25) |             |
| HIRE_DATE   | DATE         |             |

Which INSERT statement is valid?

- A. INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date) VALUES ( 1000, 'John', 'Smith', '01/01/01');
- B. INSERT INTO employees(employee\_id, first\_name, last\_name, hire\_date) VALUES ( 1000, 'John', 'Smith', '01 January 01');
- C. INSERT INTO employees(employee\_id, first\_name, last\_name, Hire\_date) VALUES ( 1000, 'John', 'Smith', To\_date('01/01/01'));
- D. INSERT INTO employees(employee\_id, first\_name, last\_name, hire\_date) VALUES ( 1000, 'John', 'Smith', 01-Jan-01);

**Correct Answer: D**

**Section: (none)**

**Explanation**

#### Explanation/Reference:

Explanation: It is the only statement that has a valid date; all other will result in an error.

Answer A is incorrect, syntax error, invalid date format

#### QUESTION 72

The user Alice wants to grant all users query privileges on her DEPT table. Which SQL statement accomplishes this?

- A. GRANT select ON dept  
TO ALL\_USERS;
- B. GRANT select ON dept  
TO ALL;
- C. GRANT QUERY ON dept  
TO ALL\_USERS
- D. GRANT select ON dept  
TO PUBLIC;

**Correct Answer: D**

**Section: (none)**

**Explanation**

#### Explanation/Reference:

Explanation: view the columns associated with the constraint names in the USER\_CONS\_COLUMNS view.



Incorrect answer:

- A table to view all constraints definition and names
- B show all object name belong to user
- C does not display column associated
- E no such view

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-25

### QUESTION 73

Which is an iSQL\*Plus command?

- A. INSERT
- B. UPDATE
- C. SELECT
- D. DESCRIBE
- E. DELETE
- F. RENAME

**Correct Answer:** D

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

The only SQL\*Plus command in this list: DESCRIBE. It cannot be used as SQL command. This command returns a description of tablename, including all columns in that table, the datatype for each column and an indication of whether the column permits storage of NULL values.

Incorrect answer:

- A INSERT is not a SQL\*PLUS command
- B UPDATE is not a SQL\*PLUS command
- C SELECT is not a SQL\*PLUS command
- E DELETE is not a SQL\*PLUS command
- F RENAME is not a SQL\*PLUS command

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7

### QUESTION 74

You work as a database administrator at ABC.com. You study the exhibit carefully.

Exhibit

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

Using the PROMOTIONS table, you need to display the names of all promos done after January 1, 2001 starting with the latest promo.

Which query would give the required result? (Choose all that apply.)



- A. 

```
SELECT promo_name,promo_begin_date
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY 1 DESC;
```
- B. 

```
SELECT promo_name,promo_begin_date "START DATE"
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY "START DATE" DESC;
```
- C. 

```
SELECT promo_name,promo_begin_date
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY 2 DESC;
```
- D. 

```
SELECT promo_name,promo_begin_date
FROM promotions
WHERE promo_begin_date > '01-JAN-01'
ORDER BY promo_name DESC;
```

**Correct Answer:** BC

**Section:** (none)

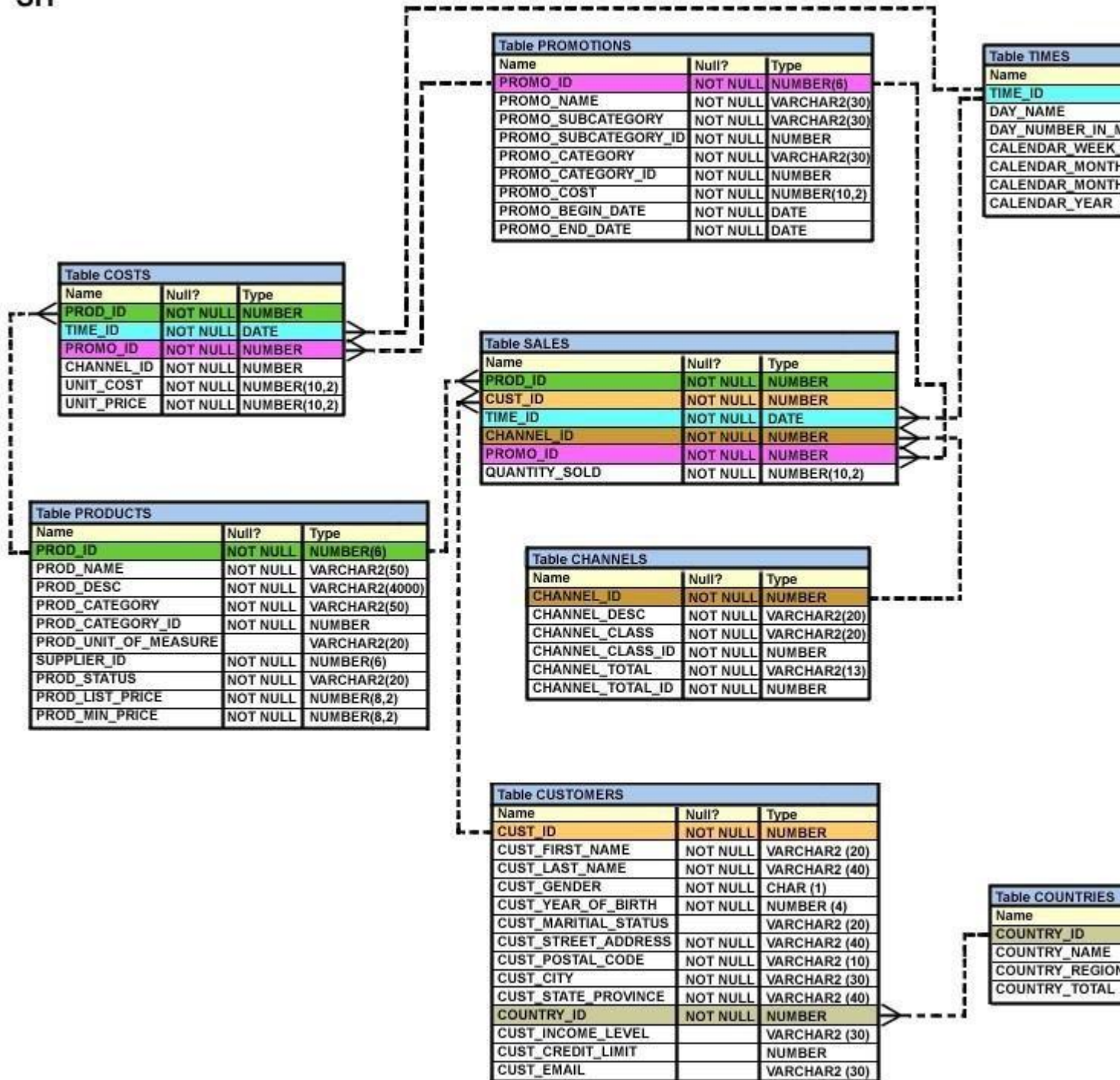
**Explanation**

**Explanation/Reference:**

#### **QUESTION 75**

See the Exhibit and examine the structure of the SALES, CUSTOMERS, PRODUCTS and ITEMS tables:

SH



The PROD\_ID column is the foreign key in the SALES table, which references the PRODUCTS table. Similarly, the CUST\_ID and TIME\_ID columns are also foreign keys in the SALES table referencing the CUSTOMERS and TIMES tables, respectively.

Evaluate the following the CREATE TABLE command:

Exhibit:

```
CREATE TABLE new_sales(prod_id, cust_id, order_date DEFAULT SYSDATE)
AS
SELECT prod_id, cust_id, time_id
FROM sales;
```

Which statement is true regarding the above command?

- A. The NEW\_SALES table would not get created because the column names in the CREATE TABLE command and the SELECT clause do not match
- B. The NEW\_SALES table would get created and all the NOT NULL constraints defined on the specified columns would be passed to the new table
- C. The NEW\_SALES table would not get created because the DEFAULT value cannot be specified in the column definition
- D. The NEW\_SALES table would get created and all the FOREIGN KEY constraints defined on the specified columns would be passed to the new table

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Creating a Table Using a Subquery

Create a table and insert rows by combining the CREATE TABLE statement and the AS subquery option.

CREATE TABLE table

[(column, column...)]

AS subquery;

Match the number of specified columns to the number of subquery columns.

Define columns with column names and default values.

Guidelines

The table is created with the specified column names, and the rows retrieved by the SELECT statement are inserted into the table.

The column definition can contain only the column name and default value. If column specifications are given, the number of columns must equal the number of columns in the subquery SELECT list.

If no column specifications are given, the column names of the table are the same as the column names in the subquery.

The column data type definitions and the NOT NULL constraint are passed to the new table. Note that only the explicit NOT NULL constraint will be inherited. The PRIMARY KEY column will not pass the NOT NULL feature to the new column. Any other constraint rules are not passed to the new table. However, you can add constraints in the column definition.

**QUESTION 76**

The CUSTOMERS table has the following structure:

Exhibit:

| Name              | Null?    | Type         |
|-------------------|----------|--------------|
| -----             | -----    | -----        |
| CUST_ID           | NOT NULL | NUMBER       |
| CUST_FIRST_NAME   | NOT NULL | VARCHAR2(20) |
| CUST_LAST_NAME    | NOT NULL | VARCHAR2(30) |
| CUST_INCOME_LEVEL |          | VARCHAR2(30) |
| CUST_CREDIT_LIMIT |          | NUMBER       |

You need to write a query that does the following task:

- \* Display the first name and tax amount of the customers. Tax is 5% of their credit limit
- \* Only those customers whose income level has a value should be considered
- \* Customers whose tax amount is null should not be considered

Which statement accomplishes all the required tasks?

- A. `SELECT cust_first_name, cust_credit_limit * .05 AS TAX_AMOUNT FROM customers WHERE cust_income_level IS NOT NULL AND tax_amount IS NOT NULL;`
- B. `SELECT cust_first_name, cust_credit_limit * .05 AS TAX_AMOUNT FROM customers WHERE cust_income_level IS NOT NULL AND cust_credit_limit IS NOT NULL;`
- C. `SELECT cust_first_name, cust_credit_limit * .05 AS TAX_AMOUNT FROM customers WHERE cust_income_level <> NULL AND tax_amount <> NULL;`
- D. `SELECT cust_first_name, cust_credit_limit * .05 AS TAX_AMOUNT FROM customers WHERE (cust_income_level,tax_amount) IS NOT NULL;`

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 77

You need to display the date 11-Oct-2007 in words as 'Eleventh of October, Two Thousand Seven'. Which SQL statement would give the required result?

- A. `SELECT TO_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year') FROM DUAL;`
- B. `SELECT TO_CHAR(TO_DATE('11-oct-2007'), 'fmDdspth of month, year') FROM DUAL;`
- C. `SELECT TO_CHAR(TO_DATE('11-oct-2007'), 'fmDthsp "of" Month, Year') FROM DUAL;`
- D. `SELECT TO_DATE(TO_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year')) FROM DUAL;`

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the TO\_CHAR Function with Dates

TO\_CHAR converts a datetime data type to a value of VARCHAR2 data type in the format specified by the format\_model. A format model is a character literal that describes the format of datetime stored in a character string. For example, the datetime format model for the string '11- Nov-1999' is 'DD-Mon-YYYY'. You can use the

TO\_CHAR function to convert a date from its default format to the one that you specify.

Guidelines

- The format model must be enclosed with single quotation marks and is case-sensitive. • The format model can include any valid date format element. But be sure to separate the date value from the format model with a comma.
- The names of days and months in the output are automatically padded with blanks. • To remove padded blanks or to suppress leading zeros, use the fill mode fm element.

Elements of the Date Format Model

----- DY Three-letter abbreviation of the day of the week

DAY Full name of the day of the week

DD Numeric day of the month

MM Two-digit value for the month  
MON Three-letter abbreviation of the month  
MONTH Full name of the month  
YYYY Full year in numbers  
YEAR Year spelled out (in English)

#### **QUESTION 78**

Which statement is true regarding the INTERSECT operator?

- A. It ignores NULL values
- B. The number of columns and data types must be identical for all SELECT statements in the query
- C. The names of columns in all SELECT statements must be identical
- D. Reversing the order of the intersected tables the result

**Correct Answer:** B

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

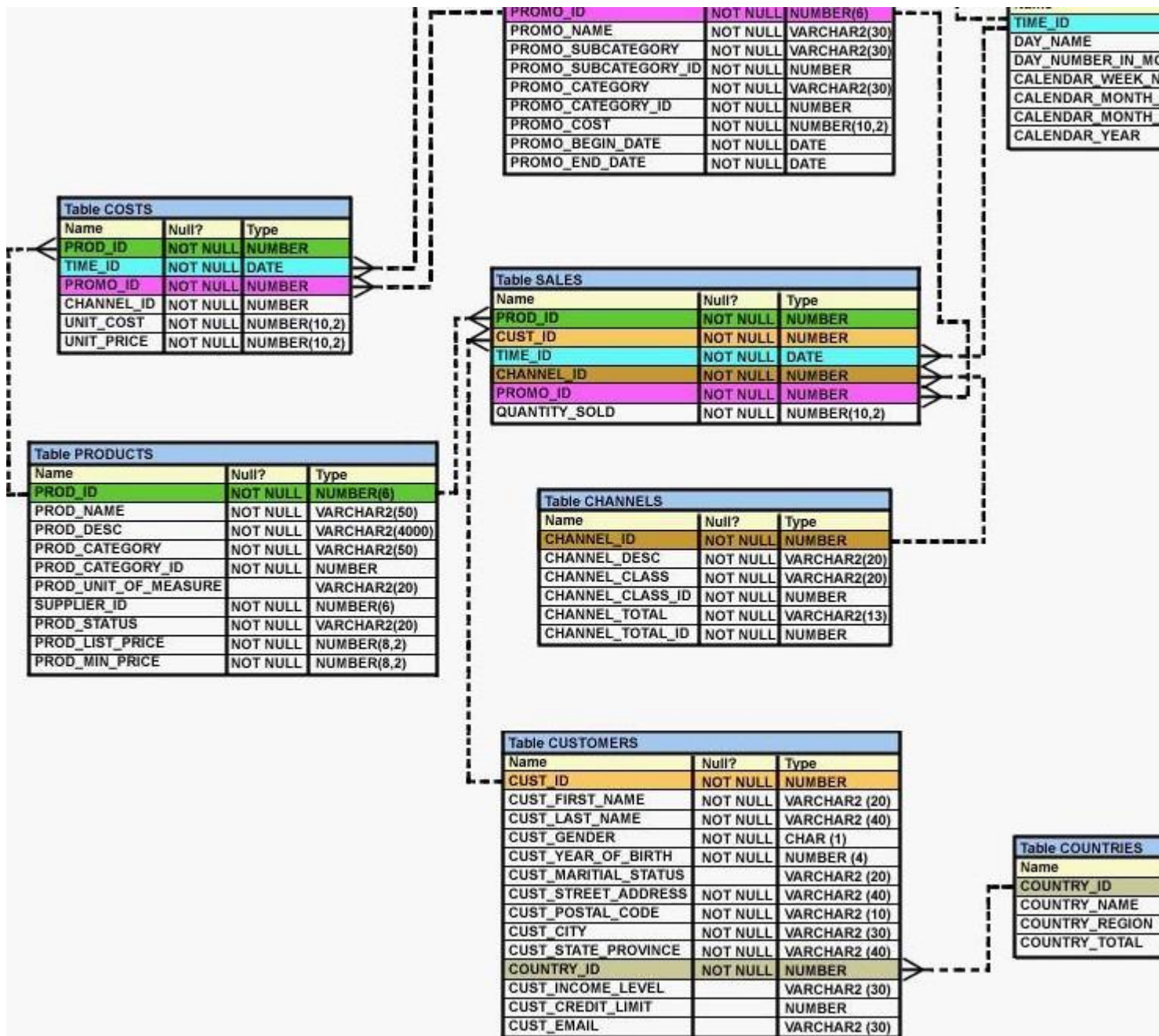
Explanation:

INTERSECT Returns only the rows that occur in both queries' result sets, sorting them and removing duplicates.

The columns in the queries that make up a compound query can have different names, but the output result set will use the names of the columns in the first query.

#### **QUESTION 79**

You work as a database administrator at ABC.com. You study the exhibit carefully and examine the structure of CUSTOMRS AND SALES tables.



Evaluate the following SQL statement:

Exhibit:

```
UPDATE (SELECT prod_id, cust_id, quantity_sold, time_id
        FROM sales)
SET time_id = '22-MAR-2007'
WHERE cust_id = (SELECT cust_id
                  FROM customers
                  WHERE cust_last_name = 'Roberts' AND
                     credit_limit = 600);
```

Which statement is true regarding the execution of the above UPDATE statement?

- A. It would execute and restrict modifications to only the column specified in the SELECT statement
- B. It would not execute because two tables cannot be used in a single UPDATE statement
- C. It would not execute because a sub query cannot be used in the WHERE clause of an UPDATE statement
- D. It would not execute because the SELECT statement cannot be used in place of the table name

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 80**

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)  
SEMESTER\_END DATE  
GPA NUMBER(4,3)

The registrar has asked for a report on the average grade point average (GPA), sorted from the highest grade point average to each semester, starting from the earliest date.

Which statement accomplish this?

- A. SELECT student\_id, semester\_end, gpa  
FROM student\_grades  
ORDER BY semester\_end DESC, gpa DESC;
- B. SELECT student\_id, semester\_end, gpa  
FROM student\_grades  
ORDER BY semester\_end, gpa ASC
- C. SELECT student\_id, semester\_end, gpa  
FROM student\_grades  
ORDER BY gpa DESC, semester\_end ASC;
- D. SELECT student\_id, semester\_end, gpa  
FROM student\_grades  
ORDER BY gpa DESC, semester\_end DESC;
- E. SELECT student\_id, semester\_end, gpa  
FROM student\_grades  
ORDER BY gpa DESC, semester\_end ASC;
- F. SELECT student\_id, semester\_end, gpa  
FROM student\_grades  
ORDER BY semester\_end, gpa DESC

**Correct Answer:** F

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 81**

You work as a database administrator at ABC.com. You study the exhibit carefully.

Exhibit:

**CUSTOMERS**

| CUST_NO | CUST_NAME | CUST_CITY | CUST_CREDIT_LIMIT |
|---------|-----------|-----------|-------------------|
| 101     | KING      | NEW YORK  | 100000            |
| 102     | GREEN     | BOSTON    | 150000            |
| 103     | SCOTT     | LONDON    |                   |
| 104     | SMITH     | BOSTON    |                   |

Evaluate the following query:  
Exhibit:

```
SQL> SELECT cust_name AS "NAME", cust_credit_limit/2 AS MIDPOINT,  
MIDPOINT+100 AS "MAX  
LOWER LIMIT"  
FROM customers;
```

The above query produces an error on execution. What is the reason for the error?

- A. An alias cannot be used in an expression
- B. The alias MIDPOINT should be enclosed within double quotation marks for the CUST\_CREDIT\_LIMIT/2 expression
- C. The MIDPOINT +100 expression gives an error because CUST\_CREDIT\_LIMIT contains NULL values
- D. The alias NAME should not be enclosed within double quotation marks

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 82**

Which statement is true regarding synonyms?

- A. Synonyms can be created only for a table
- B. Synonyms are used to reference only those tables that are owned by another user
- C. The DROP SYNONYM statement removes the synonym and the table on which the synonym has been created becomes invalid
- D. A public synonym and a private synonym can exist with the same name for the same table

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 83**

You work as a database administrator at ABC.com. You study the exhibit carefully.  
Exhibit:



| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

Examine the structure of PRODUCTS table.

Using the PRODUCTS table, you issue the following query to generate the names, current list price and discounted list price for all those products whose list price falls below \$10 after a discount of 25% is applied on it.

Exhibit:

```
SQL>SELECT prod_name, prod_list_price,
           prod_list_price - (prod_list_price * .25) "DISCOUNTED_PRICE"
FROM products
WHERE discounted_price < 10;
```

The query generates an error.

What is the reason of generating error?

- A. The column alias should be put in uppercase and enclosed within double quotation marks in the WHERE clause
- B. The parenthesis should be added to enclose the entire expression
- C. The column alias should be replaced with the expression in the WHERE clause
- D. The double quotation marks should be removed from the column alias

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Note: You cannot use column alias in the WHERE clause.

#### QUESTION 84

Which one is a system privilege?

- A. SELECT
- B. DELETE
- C. EXECUTE
- D. ALTER TABLE
- E. CREATE TABLE

**Correct Answer: E**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 85

What is true about sequences?

- A. The start value of the sequence is always 1.
- B. A sequence always increments by 1.
- C. The minimum value of an ascending sequence defaults to 1.
- D. The maximum value of descending sequence defaults to 1.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 86

Examine the structure of the INVOICE table:  
Exhibit:

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

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT inv\_no,NVL2(inv\_date,'Pending','Incomplete')  
FROM invoice;
- B. SELECT inv\_no,NVL2(inv\_amt,inv\_date,'Not Available')  
FROM invoice;
- C. SELECT inv\_no,NVL2(inv\_date,sysdate-inv\_date,sysdate)  
FROM invoice;
- D. SELECT inv\_no,NVL2(inv\_amt,inv\_amt\*.25,'Not Available')  
FROM invoice;

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The NVL2 Function

The NVL2 function provides an enhancement to NVL but serves a very similar purpose. It evaluates whether a column or expression of any data type is null or not.

5-6 The NVL function\

If the first term is not null, the second parameter is returned, else the third parameter is returned. Recall that the NVL function is different since it returns the original term if it is not null. The NVL2 function takes three mandatory parameters. Its syntax is NVL2(original, ifnotnull, ifnull), where original represents the term being tested. Ifnotnull is returned if original is not null, and ifnull is returned if original is null. The data types of the ifnotnull and ifnull parameters must be compatible, and they cannot be of type LONG.

They must either be of the same type, or it must be possible to convert ifnull to the type of the ifnotnull parameter. The data type returned by the NVL2 function is the same as that of the ifnotnull parameter.

#### QUESTION 87

View the Exhibit and examine the description for 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_MARITAL_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) |

You want to update the CUST\_INCOME\_LEVEL and CUST\_CREDIT\_LIMIT columns for the customer with the CUST\_ID 2360. You want the value for the CUST\_INCOME\_LEVEL to have the same value as that of the customer with the CUST\_ID 2560 and the CUST\_CREDIT\_LIMIT to have the same value as that of the customer with CUST\_ID 2566.

Which UPDATE statement will accomplish the task?

- A. UPDATE customers  
SET cust\_income\_level = (SELECT cust\_income\_level  
FROM customers  
WHERE cust\_id = 2560),  
cust\_credit\_limit = (SELECT cust\_credit\_limit  
FROM customers  
WHERE cust\_id = 2566)  
WHERE cust\_id=2360;
- B. UPDATE customers  
SET (cust\_income\_level,cust\_credit\_limit) = (SELECT  
cust\_income\_level, cust\_credit\_limit  
FROM customers  
WHERE cust\_id=2560 OR cust\_id=2566)  
WHERE cust\_id=2360;
- C. UPDATE customers  
SET (cust\_income\_level,cust\_credit\_limit) = (SELECT  
cust\_income\_level, cust\_credit\_limit  
FROM customers  
WHERE cust\_id IN(2560, 2566)  
WHERE cust\_id=2360;
- D. UPDATE customers  
SET (cust\_income\_level,cust\_credit\_limit) = (SELECT  
cust\_income\_level, cust\_credit\_limit  
FROM customers  
WHERE cust\_id=2560 AND cust\_id=2566)  
WHERE cust\_id=2360;

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Updating Two Columns with a Subquery

You can update multiple columns in the SET clause of an UPDATE statement by writing multiple subqueries.

The syntax is as follows:

UPDATE table

```
SET column =  
(SELECT column  
FROM table  
WHERE condition)  
[ ,  
column =  
(SELECT column  
FROM table  
WHERE condition)]  
[WHERE condition ] ;
```

### QUESTION 88

A SELECT statement can be used to perform these three functions:

1. Choose rows from a table.
2. Choose columns from a table
3. Bring together data that is stored in different tables by creating a link between them.

Which set of keywords describes these capabilities?

- A. difference, projection, join
- B. selection, projection, join
- C. selection, intersection, join
- D. intersection, projection, join
- E. difference, projection, product

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation: choose rows from a table is SELECTION,  
Choose column from a table is PROJECTION  
Bring together data in different table by creating a link between them is JOIN.

Incorrect answer:

A answer should have SELECTION, PROJECTION and JOIN.  
C answer should have SELECTION, PROJECTION and JOIN.  
D answer should have SELECTION, PROJECTION and JOIN.  
E answer should have SELECTION, PROJECTION and JOIN.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 1-6

### QUESTION 89

See the Exhibit and examine the structure and data in the INVOICE table:

Exhibit:

| INVOICE  |           |             |         |
|----------|-----------|-------------|---------|
| Name     | Null?     | Type        |         |
| INV_NO   | NOT NULL  | NUMBER(3)   |         |
| INV_DATE |           | DATE        |         |
| CUST_ID  |           | VARCHAR2(4) |         |
| INV_AMT  |           | NUMBER(8,2) |         |
| INV_NO   | INV_DATE  | CUST_ID     | INV_AMT |
| 1        | 01-APR-07 | A1Q         | 1000    |
| 2        | 01-OCT-07 | B1R         | 2000    |
| 3        | 01-FEB-07 |             | 3000    |

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT MAX(inv\_date),MIN(cust\_id)  
FROM invoice;
- B. SELECT AVG(inv\_date-SYSDATE),AVG(inv\_amt)  
FROM invoice;
- C. SELECT MAX(AVG(SYSDATE-inv\_date))  
FROM invoice;
- D. SELECT AVG(inv\_date)  
FROM invoice;

**Correct Answer:** AB

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 90

You are currently located in Singapore and have connected to a remote database in Chicago. You issue the following command:

Exhibit:

```
SQL> SELECT ROUND(SYSDATE-promo_begin_date,0)
      FROM promotions
      WHERE (SYSDATE-promo_begin_date)/365 > 2;
```

PROMOTIONS is the public synonym for the public database link for the PROMOTIONS table.

What is the outcome?

- A. Number of days since the promo started based on the current Chicago data and time
- B. Number of days since the promo started based on the current Singapore data and time.
- C. An error because the WHERE condition specified is invalid
- D. An error because the ROUND function specified is invalid

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 91**

Which is a valid CREATE TABLE statement?

- A. CREATE TABLE EMP9\$# AS (empid number(2));
- B. CREATE TABLE EMP\*123 AS (empid number(2));
- C. CREATE TABLE PACKAGE AS (packid number(2));
- D. CREATE TABLE 1EMP\_TEST AS (empid number(2));

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Table names and column names must begin with a letter and be 1-30 characters long. Characters A-Z,a-z, 0-9, \_, \$ and # (legal characters but their use is discouraged).

Incorrect answer:

B Non alphanumeric character such as "\*" is discourage in Oracle table name.

D Table name must begin with a letter.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 9-4

**QUESTION 92**

Evaluate the following SQL statements:

Exhibit:

```
CREATE TABLE employees
(employee_id   NUMBER(2) PRIMARY KEY,
 last_name    VARCHAR2(25) NOT NULL,
 department_id NUMBER(2) NOT NULL,
 job_id       VARCHAR2(8),
 salary       NUMBER(10,2));
```

You issue the following command to create a view that displays the IDs and last names of the sales staff in the organization.

Exhibit:

```
CREATE OR REPLACE VIEW sales_staff_vu AS
SELECT employee_id,
 last_name, job_id
FROM employees
WHERE job_id LIKE 'SA_%' WITH CHECK OPTION;
```

Which two statements are true regarding the above view? (Choose two.)

- A. It allows you to update job IDs of the existing sales staff to any other job ID in the EMPLOYEES table
- B. It allows you to delete details of the existing sales staff from the EMPLOYEES table
- C. It allows you to insert rows into the EMPLOYEES table
- D. It allows you to insert IDs, last names, and job IDs of the sales staff from the view if it is used in multitable INSERT statements

Correct Answer: BD

Section: (none)

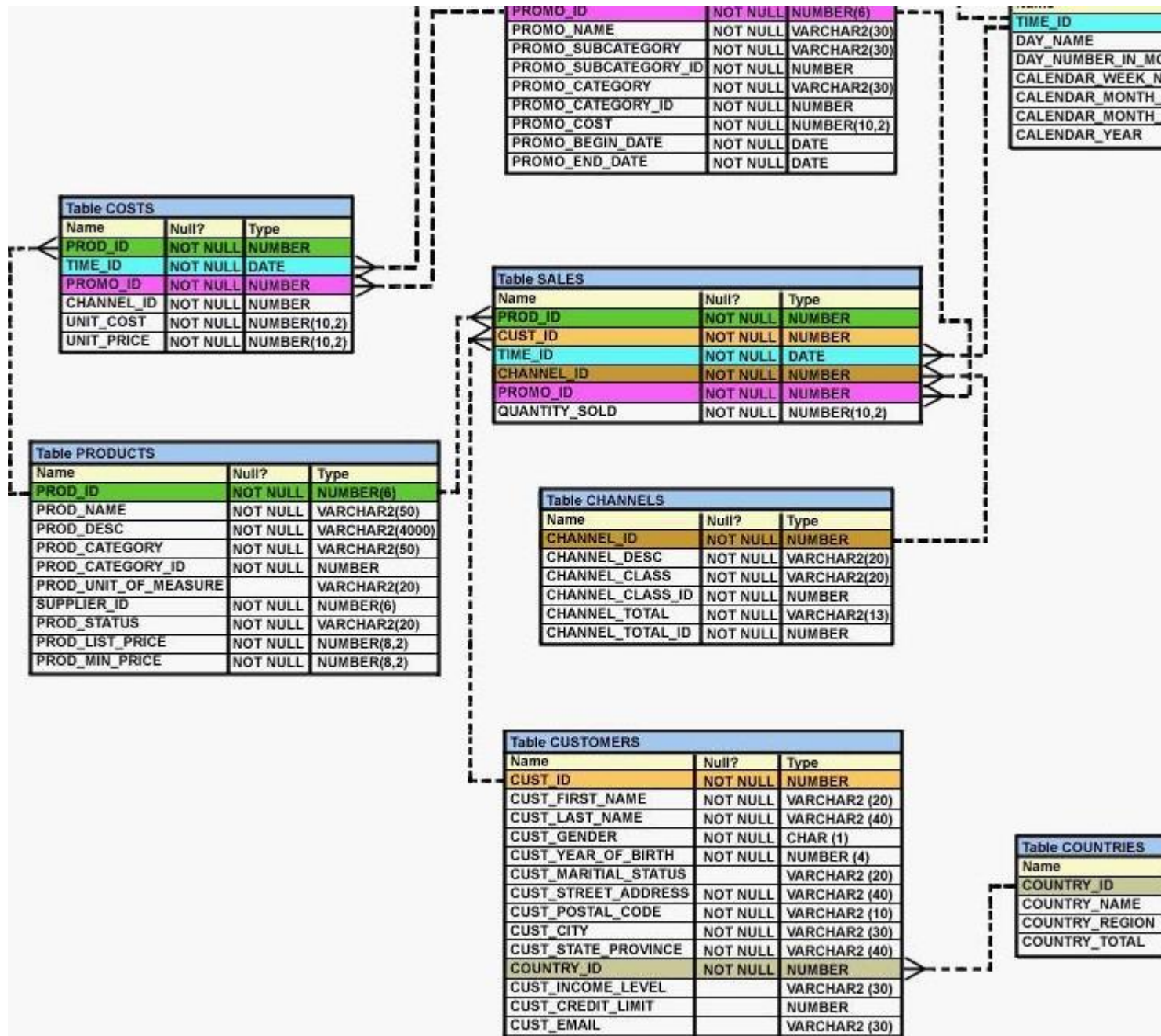
Explanation

Explanation/Reference:

### QUESTION 93

See the Exhibit and Examine the structure of SALES and PROMOTIONS tables:

Exhibit:



You want to delete rows from the SALES table, where the PROMO\_NAME column in the PROMOTIONS table

has either blowout sale or everyday low price as values. Which DELETE statements are valid? (Choose all that apply.)

- A. DELETE  
FROM sales  
WHERE promo\_id = (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'blowout sale')  
AND promo\_id = (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'everyday low price');
- B. DELETE  
FROM sales  
WHERE promo\_id = (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'blowout sale')  
OR promo\_id = (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'everyday low price');
- C. DELETE  
FROM sales  
WHERE promo\_id IN (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'blowout sale'  
OR promo\_name = 'everyday low price');
- D. D DELETE  
FROM sales  
WHERE promo\_id IN (SELECT promo\_id  
FROM promotions  
WHERE promo\_name IN ('blowout sale','everyday low price'));

**Correct Answer:** BCD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 94

Which three statements/commands would cause a transaction to end? (Choose three.)

- A. COMMIT
- B. SELECT
- C. CREATE
- D. ROLLBACK
- E. SAVEPOINT

**Correct Answer:** ACD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 95

You want to create an ORD\_DETAIL table to store details for an order placed having the following business requirement:



- 1) The order ID will be unique and cannot have null values.
- 2) The order date cannot have null values and the default should be the current date.
- 3) The order amount should not be less than 50.
- 4) The order status will have values either shipped or not shipped.
- 5) The order payment mode should be cheque, credit card, or cash on delivery (COD).

Which is the valid DDL statement for creating the ORD\_DETAIL table?

- A. 

```
CREATE TABLE ord_details
(ord_id NUMBER(2) CONSTRAINT ord_id_nn NOT NULL,
ord_date DATE DEFAULT SYSDATE NOT NULL,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount > 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```
- B. 

```
CREATE TABLE ord_details
(ord_id NUMBER(2) CONSTRAINT ord_id_uk UNIQUE NOT NULL,
ord_date DATE DEFAULT SYSDATE NOT NULL,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount > 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```
- C. 

```
CREATE TABLE ord_details
(ord_id NUMBER(2) CONSTRAINT ord_id_pk PRIMARY KEY,
ord_date DATE DEFAULT SYSDATE NOT NULL,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount >= 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```
- D. 

```
CREATE TABLE ord_details
(ord_id NUMBER(2),
ord_date DATE NOT NULL DEFAULT SYSDATE,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount >= 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

## QUESTION 96

Which three statements are true regarding sub queries? (Choose three.)

- A. Multiple columns or expressions can be compared between the main query and sub query
- B. Sub queries can contain GROUP BY and ORDER BY clauses
- C. Only one column or expression can be compared between the main query and subquery
- D. Main query and sub query can get data from different tables
- E. Main query and sub query must get data from the same tables
- F. Sub queries can contain ORDER BY but not the GROUP BY clause

**Correct Answer:** ABD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 97

See the Exhibit and examine the structure of the PROMOTIONS table:

Exhibit:

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

Which SQL statements are valid? (Choose all that apply.)

- A. 

```
SELECT promo_id, DECODE(NVL(promo_cost,0), promo_cost,
  promo_cost * 0.25, 100) "Discount"
FROM promotions;
```
- B. 

```
SELECT promo_id, DECODE(promo_cost, 10000,
  DECODE(promo_category, 'G1', promo_cost *.25, NULL),
  NULL) "Catcost"
FROM promotions;
```
- C. 

```
SELECT promo_id, DECODE(NULLIF(promo_cost, 10000),
  NULL, promo_cost*.25, 'N/A') "Catcost"
FROM promotions;
```
- D. 

```
SELECT promo_id, DECODE(promo_cost, >10000, 'High',
  <10000, 'Low') "Range"
FROM promotions;
```

**Correct Answer:** AB

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The DECODE Function

Although its name sounds mysterious, this function is straightforward. The DECODE function implements

ifthen-else conditional logic by testing its first two terms for equality and returns the third if they are equal and optionally returns another term if they are not. The DECODE function takes at least three mandatory parameters, but can take many more. The syntax of the function is DECODE(expr1,comp1, iftrue1, [comp2,iftrue2...[ compN,iftrueN]], [iffalse]).

#### QUESTION 98

Which SQL statement displays the date March 19, 2001 in a format that appears as "Nineteenth of March 2001 12:00:00 AM"?

- A. 

```
SELECT
TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'fmDdspth
"of" Month YYYY fmHH:MI:SS AM') NEW_DATE
FROM dual;
```
- B. 

```
SELECT
TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'Ddspth
"of" Month YYYY fmHH:MI:SS AM') NEW_DATE
FROM dual;
```
- C. 

```
SELECT
TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'fmDdspth "of" Month YYYY
HH:MI:SS AM') NEW_DATE FROM dual;
```
- D. 

```
SELECT
TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'fmDdspth "of" Month YYYYfmtHH:HI:SS AM')
NEW_DATE FROM dual;
```

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 99

Which three tasks can be performed using SQL functions built into Oracle Database? (Choose three.)

- A. Combining more than two columns or expressions into a single column in the output
- B. Displaying a date in a nondefault format
- C. Substituting a character string in a text expression with a specified string
- D. Finding the number of characters in an expression

**Correct Answer:** BCD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 100

For which action can you use the TO\_DATE function?

- A. Convert any date literal to a date
- B. Convert any numeric literal to a date
- C. Convert any character literal to a date
- D. Convert any date to a character literal
- E. Format '10-JAN-99' to 'January 10 1999'

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 101**

Which statement is true regarding the default behavior of the ORDER BY clause?

- A. In a character sort, the values are case-sensitive
- B. NULL values are not considered at all by the sort operation
- C. Only those columns that are specified in the SELECT list can be used in the ORDER BY clause
- D. Numeric values are displayed from the maximum to the minimum value if they have decimal positions

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Character Strings and Dates

Character strings and date values are enclosed with single quotation marks. Character values are case-sensitive and date values are format-sensitive.

The default date display format is DD-MON-RR.

**QUESTION 102**

Evaluate the following SQL statements:

Exhibit:

```
SELECT INTERVAL '300' MONTH,  
INTERVAL '54-2' YEAR TO MONTH,  
INTERVAL '11:12:10.1234567' HOUR TO SECOND  
FROM dual;
```

Which is the correct output of the above query?

- A. +00-300, +54-02, +00 11:12:10.123457
- B. +00-300, +00-650, +00 11:12:10.123457
- C. +25-00, +54-02, +00 11:12:10.123457
- D. +25-00, +00-650, +00 11:12:10.123457

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 103**

See the Exhibit and examine the structure of ORD table:

Exhibit:

**ORD**

| Name     | Null?    | Type      |
|----------|----------|-----------|
| ORD_NO   | NOT NULL | NUMBER(2) |
| ORD_DATE |          | DATE      |
| CUST_ID  |          | NUMBER(4) |

Evaluate the following SQL statements that are executed in a user session in the specified order:

```
CREATE SEQUENCE ord_seq;  
SELECT ord_seq.nextval  
FROM dual;  
INSERT INTO ord  
VALUES (ord_seq.CURRVAL, '25-jan-2007',101);  
UPDATE ord  
SET ord_no= ord_seq.NEXTVAL  
WHERE cust_id =101;
```

What would be the outcome of the above statements?

- A. All the statements would execute successfully and the ORD\_NO column would contain the value 2 for the CUST\_ID 101.
- B. The CREATE SEQUENCE command would not execute because the minimum value and maximum value for the sequence have not been specified.
- C. The CREATE SEQUENCE command would not execute because the starting value of the sequence and the increment value have not been specified.
- D. All the statements would execute successfully and the ORD\_NO column would have the value 20 for the CUST\_ID 101 because the default CACHE value is 20.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 104**

Which two statements are true about constraints? (Choose two.)

- A. The UNIQUE constraint does not permit a null value for the column.
- B. A UNIQUE index gets created for columns with PRIMARY KEY and UNIQUE constraints.
- C. The PRIMARY KEY and FOREIGN KEY constraints create a UNIQUE index.
- D. The NOT NULL constraint ensures that null values are not permitted for the column.

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

B: A unique constraint can contain null values because null values cannot be compared to anything.

D: The NOT NULL constraint ensure that null value are not permitted for the column

Incorrect answer:

A statement is not true

C statement is not true

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-9

### QUESTION 105

Examine the structure of the EMPLOYEES table:

EMPLOYEE\_ID NUMBER NOT NULL, Primary Key

EMP\_NAME VARCHAR2(30)

JOB\_ID NUMBER\

SAL NUMBER

MGR\_ID NUMBER References EMPLOYEE\_ID column

DEPARTMENT\_ID NUMBER Foreign key to DEPARTMENT\_ID column of the DEPARTMENTS table

You created a sequence called EMP\_ID\_SEQ in order to populate sequential values for the EMPLOYEE\_ID column of the EMPLOYEES table.

Which two statements regarding the EMP\_ID\_SEQ sequence are true? (Choose two.)

- A. You cannot use the EMP\_ID\_SEQ sequence to populate the JOB\_ID column.
- B. The EMP\_ID\_SEQ sequence is invalidated when you modify the EMPLOYEE\_ID column.
- C. The EMP\_ID\_SEQ sequence is not affected by modifications to the EMPLOYEES table.
- D. Any other column of NUMBER data type in your schema can use the EMP\_ID\_SEQ sequence.
- E. The EMP\_ID\_SEQ sequence is dropped automatically when you drop the EMPLOYEES table.
- F. The EMP\_ID\_SEQ sequence is dropped automatically when you drop the EMPLOYEE\_ID column.

**Correct Answer:** CD

**Section:** (none)

**Explanation**

#### Explanation/Reference:

Explanation: the EMP\_ID\_SEQ sequence is not affected by modification to the EMPLOYEES table. Any other column of NUMBER data type in your schema can use the EMP\_ID\_SEQ sequence.

Incorrect answer:

A EMP\_ID\_SEQ sequence can be use to populate JOB\_ID

B EMP\_ID\_SEQ sequence will not be invalidate when column in EMPLOYEE\_ID is modify. E EMP\_ID\_SEQ sequence will be dropped automatically when you drop the EMPLOYEES table.

F EMP\_ID\_SEQ sequence will be dropped automatically when you drop the EMPLOYEE\_ID column.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 12-4

### QUESTION 106

The SQL statements executed in a user session as follows:

Exhibit:

```
SQL> CREATE TABLE product
      (pcode NUMBER(2),
       pname VARCHAR2(10));
SQL> INSERT INTO product VALUES (1, 'pen');
SQL> INSERT INTO product VALUES (2, 'pencil');
SQL> SAVEPOINT a;
SQL> UPDATE product SET pcode = 10 WHERE pcode = 1;
SQL> SAVEPOINT b;
SQL> DELETE FROM product WHERE pcode = 2;
SQL> COMMIT; SQL> DELETE FROM product WHERE pcode=10;
```

Which two statements describe the consequence of issuing the ROLLBACK TO SAVE POINT a command in the session? (Choose two.)

- A. Both the DELETE statements and the UPDATE statement are rolled back
- B. The rollback generates an error
- C. Only the DELETE statements are rolled back
- D. Only the seconds DELETE statement is rolled back
- E. No SQL statements are rolled back

**Correct Answer:** BE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 107

You work as a database administrator at ABC.com. You study the exhibit carefully.

Exhibit:

| CUST_STATUS  |           |              |
|--------------|-----------|--------------|
| Name         | Null?     | Type         |
| -----        | -----     | -----        |
| CUSTNO       | NOT NULL  | NUMBER(2)    |
| AMT_SPENT    |           | NUMBER(10,2) |
| CREDIT_LIMIT |           | NUMBER(10,2) |
|              |           |              |
| CUSTNO       | AMT_SPENT | CREDIT_LIMIT |
| -----        | -----     | -----        |
| 1            | 1000      | 1000         |
| 2            | 2000      | 2500         |
| 3            |           | 3000         |
| 4            | 3000      | 2800         |

You issue the following SQL statement:

```
SQL> SELECT custno, NVL2(NULLIF(amt_spent, credit_limit), 0, 1000)"BONUS"  
      FROM cust_status;
```

Which statement is true regarding the execution of the above query?

- A. It produces an error because the AMT\_SPENT column contains a null value.
- B. It displays a bonus of 1000 for all customers whose AMT\_SPENT is less than CREDIT\_LIMIT.
- C. It displays a bonus of 1000 for all customers whose AMT\_SPENT equals CREDIT\_LIMIT, or AMT\_SPENT is null.
- D. It produces an error because the TO\_NUMBER function must be used to convert the result of the NULLIF function before it can be used by the NVL2 function.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

### The NULLIF Function

The NULLIF function tests two terms for equality. If they are equal the function returns a null, else it returns the first of the two terms tested.

The NULLIF function takes two mandatory parameters of any data type. The syntax is NULLIF(ifunequal, comparison\_term), where the parameters ifunequal and comparison\_term are compared. If they are identical, then NULL is returned. If they differ, the ifunequal parameter is returned.

### QUESTION 108

View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables.

| ORDERS      |          |               |
|-------------|----------|---------------|
| Name        | Null?    | Type          |
| ORDER_ID    | NOT NULL | NUMBER (4)    |
| ORDER_DATE  | NOT NULL | DATE          |
| ORDER_MODE  |          | VARCHAR2 (8)  |
| CUSTOMER_ID | NOT NULL | NUMBER (6)    |
| ORDER_TOTAL |          | NUMBER (8, 2) |

| CUSTOMERS       |          |               |
|-----------------|----------|---------------|
| Name            | Null?    | Type          |
| CUSTOMER_ID     | NOT NULL | NUMBER (6)    |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME  | NOT NULL | VARCHAR2 (20) |
| CREDIT_LIMIT    |          | NUMBER (9, 2) |
| CUST_ADDRESS    |          | VARCHAR2 (40) |

There is only one customer with the CUST\_LAST\_NAME column having value Roberts. Which INSERT statement should be used to add a row into the ORDERS table for the customer whose CUST\_LAST\_NAME is Roberts and CREDIT\_LIMIT is 600?

- A. INSERT INTO orders  
VALUES (1,'10-mar-2007', 'direct',  
(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600), 1000);
- B. INSERT INTO orders (order\_id,order\_date,order\_mode,  
(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600),order\_total)  
VALUES(1,'10-mar-2007', 'direct', &&customer\_id, 1000);
- C. INSERT INTO(SELECT o.order\_id, o.order\_date,o.order\_mode,c.customer\_id, o.order\_total FROM orders  
o, customers c  
WHERE o.customer\_id = c.customer\_id  
AND c.cust\_last\_name='Roberts' ANDc.credit\_limit=600 )  
VALUES (1,'10-mar-2007', 'direct',(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600), 1000);
- D. INSERT INTO orders (order\_id,order\_date,order\_mode,  
(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND



```
credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &customer_id, 1000);
```

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 109

User Mary has a view called EMP\_DEPT\_LOC\_VU that was created based on the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables. She has the privilege to create a public synonym, and would like to create a synonym for this view that can be used by all users of the database.

Which SQL statement can Mary use to accomplish that task?

- A. CREATE PUBLIC SYNONYM EDL\_VU  
ON emp\_dept\_loc\_vu;
- B. CREATE PUBLIC SYNONYM EDL:VU  
FOR mary (emp\_dept\_loc\_vu);
- C. CREATE PUBLIC SYNONYM EDL\_VU  
FOR emp\_dept\_loc\_vu;
- D. CREATE SYNONYM EDL\_VU  
ON emp\_dept\_loc\_vu  
FOR EACH USER;
- E. CREATE SYNONYM EDL\_VU  
FOR EACH USER  
ON emp\_dept\_loc\_vu;
- F. CREATE PUBLIC SYNONYM EDL\_VU  
ON emp\_dept\_loc\_vu  
FOR ALL USERS;

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The general syntax to create a synonym is:

```
CREATE [PUBLIC] SYNONYM synonym FOR object;
```

#### QUESTION 110

View the Exhibit and examine the structure of the CUSTOMERS table. You have been asked to produce a report on the CUSTOMERS table showing the customers details sorted in descending order of the city and in the descending order of their income level in each city. Which query would accomplish this task?

| Table CUSTOMERS     |                 |               |
|---------------------|-----------------|---------------|
| Name                | Null?           | Type          |
| <b>CUST_ID</b>      | <b>NOT NULL</b> | <b>NUMBER</b> |
| CUST_FIRST_NAME     | NOT NULL        | VARCHAR       |
| CUST_LAST_NAME      | NOT NULL        | VARCHAR       |
| CUST_GENDER         | NOT NULL        | CHAR (1)      |
| CUST_YEAR_OF_BIRTH  | NOT NULL        | NUMBER (      |
| CUST_MARITAL_STATUS |                 | VARCHAR       |
| CUST_STREET_ADDRESS | NOT NULL        | VARCHAR       |
| CUST_POSTAL_CODE    | NOT NULL        | VARCHAR       |
| CUST_CITY           | NOT NULL        | VARCHAR       |
| CUST_STATE_PROVINCE | NOT NULL        | VARCHAR       |
| <b>COUNTRY_ID</b>   | <b>NOT NULL</b> | <b>NUMBER</b> |
| CUST_INCOME_LEVEL   |                 | VARCHAR       |
| CUST_CREDIT_LIMIT   |                 | NUMBER        |
| CUST_EMAIL          |                 | VARCHAR       |

- A. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_city desc, cust\_income\_level DESC;
- B. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_income\_level desc, cust\_city DESC;
- C. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY (cust\_city, cust\_income\_level) DESC;
- D. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_city, cust\_income\_level DESC;

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 111**

View the Exhibit and examine the structure of the PROMOTIONS, SALES, and CUSTOMER tables.

# PROMOTIONS

| Name             | Null?    | Type         |
|------------------|----------|--------------|
| -----            | -----    | -----        |
| PROMO_ID         | NOT NULL | NUMBER(2)    |
| PROMO_NAME       |          | VARCHAR2(10) |
| PROMO_CAT        |          | VARCHAR2(10) |
| PROMO_COST       |          | NUMBER(8,2)  |
| PROMO_BEGIN_DATE |          | DATE         |
| PROMO_END_DATE   |          | DATE         |

# SALES

| Name     | Null?    | Type        |
|----------|----------|-------------|
| -----    | -----    | -----       |
| PROD_ID  | NOT NULL | NUMBER(3)   |
| PROMO_ID | NOT NULL | NUMBER(3)   |
| TIME_ID  |          | DATE        |
| QTY_SOLD |          | NUMBER(6,2) |
| CUST_ID  | NOT NULL | NUMBER(2)   |

# CUSTOMER

| Name         | Null?    | Type         |
|--------------|----------|--------------|
| -----        | -----    | -----        |
| CUST_ID      | NOT NULL | NUMBER(3)    |
| CUST_NAME    |          | VARCHAR2(20) |
| CUST_ADDRESS |          | VARCHAR2(30) |

You need to generate a report showing the promo name along with the customer name for all products that were sold during their promo campaign and before 30th October 2007.  
You issue the following query:

```
SQL> SELECT promo_name,cust_name  
FROM promotions p JOIN sales s  
ON(time_id BETWEEN promo_begin_date AND promo_end_date)  
JOIN customer c  
ON (s.cust_id = c.cust_id) AND time_id < '30-oct-2007';
```

Which statement is true regarding the above query?

- A. It executes successfully and gives the required result.
- B. It executes successfully but does not give the required result.
- C. It produces an error because the join order of the tables is incorrect.
- D. It produces an error because equijoin and nonequijoin conditions cannot be used in the same SELECT statement.

**Correct Answer: B**

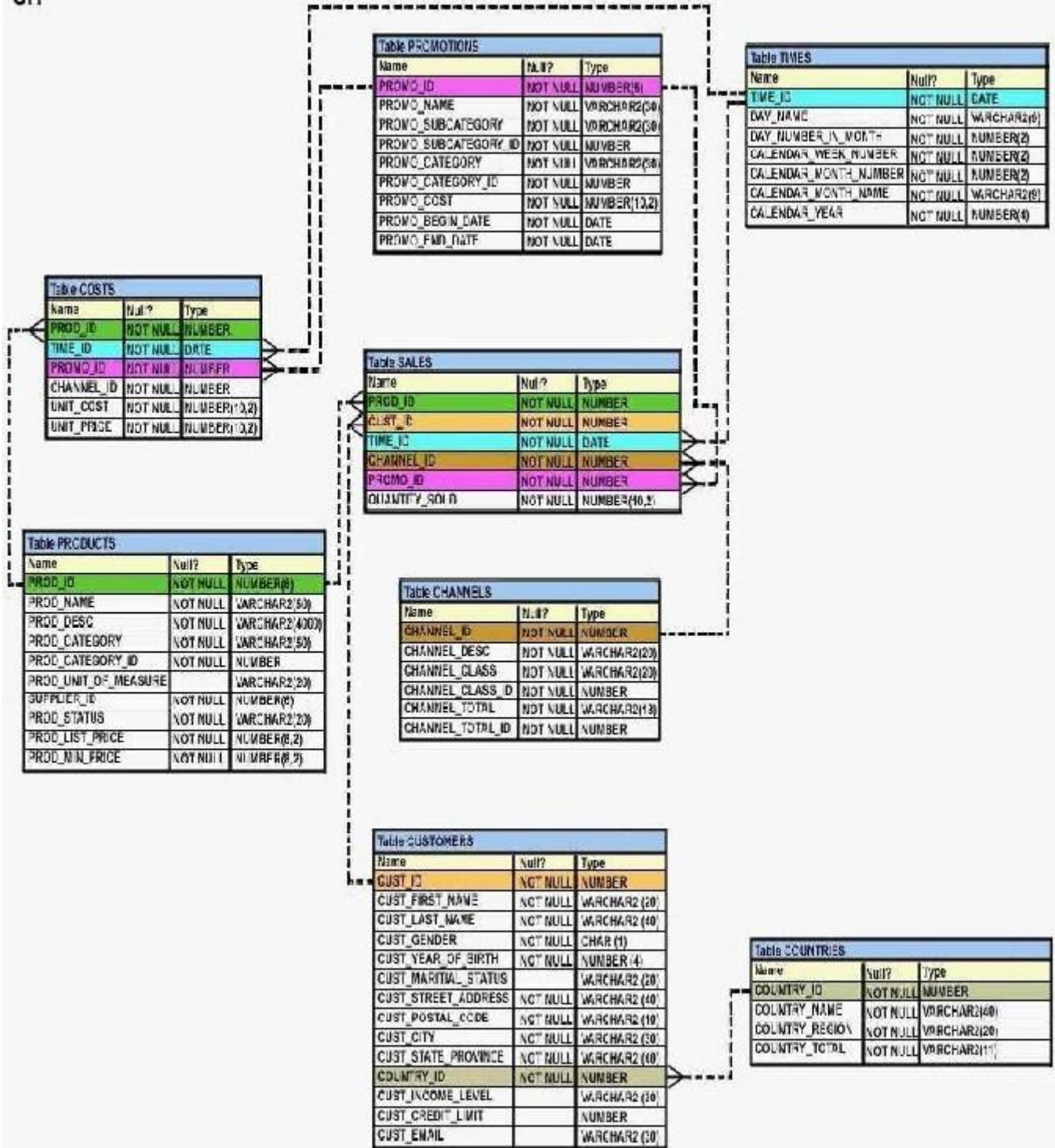
**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### **QUESTION 112**

View the Exhibit and examine the description for the PRODUCTS and SALES table.



PROD\_ID is a primary key in the PRODUCTS table and foreign key in the SALES table. You want to remove all the rows from the PRODUCTS table for which no sale was done for the last three years. Which is the valid DELETE statement?

- A. DELETE  
FROM products  
WHERE prod\_id = (SELECT prod\_id

- FROM sales  
WHERE time\_id - 3\*365 = SYSDATE );
- B. DELETE  
FROM products  
WHERE prod\_id = (SELECT prod\_id  
FROM sales  
WHERE SYSDATE >= time\_id - 3\*365 );
- C. DELETE  
FROM products  
WHERE prod\_id IN (SELECT prod\_id  
FROM sales  
WHERE SYSDATE - 3\*365 >= time\_id);
- D. DELETE  
FROM products  
WHERE prod\_id IN (SELECT prod\_id  
FROM sales  
WHERE time\_id >= SYSDATE - 3\*365 );

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 113

You need to display the first names of all customers from the CUSTOMERS table that contain the character 'e' and have the character 'a' in the second last position.

Which query would give the required output?

- A. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')<>0 AND  
SUBSTR(cust\_first\_name, -2, 1)='a';
- B. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')<>' AND  
SUBSTR(cust\_first\_name, -2, 1)='a';
- C. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')IS NOT NULL AND  
SUBSTR(cust\_first\_name, 1,-2)='a';
- D. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')<>0 AND  
SUBSTR(cust\_first\_name, LENGTH(cust\_first\_name),-2)='a';

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The SUBSTR(string, start position, number of characters) function accepts three parameters and returns a string consisting of the number of characters extracted from the source string, beginning at the specified start position:

substr('http://www.domain.com',12,6) = domain



The position at which the first character of the returned string begins.

When position is 0 (zero), then it is treated as 1.

When position is positive, then the function counts from the beginning of string to find the first character.

When position is negative, then the function counts backward from the end of string.

substring\_length

The length of the returned string. SUBSTR calculates lengths using characters as defined by the input character set. SUBSTRB uses bytes instead of characters. SUBSTRC uses Unicode complete characters. SUBSTR2 uses UCS2 code points. SUBSTR4 uses UCS4 code points. When you do not specify a value for this argument, then the function

The INSTR(source string, search item, [start position],[nth occurrence of search item]) function returns a number that represents the position in the source string, beginning from the given start position, where the nth occurrence of the search item begins:

instr('http://www.domain.com',',',1,2) = 18

#### QUESTION 114

View the Exhibit and examine the structure of the CUSTOMERS table. Evaluate the query statement:

```
SQL> SELECT cust_last_name, cust_city, cust_credit_limit
FROM customers
WHERE cust_last_name BETWEEN 'A' AND 'C' AND cust_credit_limit BETWEEN
1000 AND 3000;
```

What would be the outcome of the above statement?

| 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_MARITAL_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) |

- A. It executes successfully.
- B. It produces an error because the condition on CUST\_LAST\_NAME is invalid.
- C. It executes successfully only if the CUST\_CREDIT\_LIMIT column does not contain any null values.
- D. It produces an error because the AND operator cannot be used to combine multiple BETWEEN clauses.

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 115

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?

- A. SELECT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;
- B. SELECT DISTINCT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;
- C. SELECT DISTINCT cust\_income\_level ' ' cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;
- D. SELECT cust\_income\_level ' ' cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Duplicate Rows

Unless you indicate otherwise, SQL displays the results of a query without eliminating the duplicate rows.

To eliminate duplicate rows in the result, include the DISTINCT keyword in the SELECT clause immediately after the SELECT keyword.

You can specify multiple columns after the DISTINCT qualifier. The DISTINCT qualifier affects all the selected columns, and the result is every distinct combination of the columns.

#### QUESTION 116

View the Exhibit and examine the structure of the PROMOTIONS table. Which SQL statements are valid? (Choose all that apply.)

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(8)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

- A. SELECT promo\_id. DECODE(NVL(promo\_cost.O).promo\_cost \* 0.25. 100) "Discount" FROM promotions;
- B. SELECT promo id. DECODE(promo\_cost. 10000. DECODE(promo\_category. 'GI\ promo\_cost \* 25. NULL). NULL) "Catcost" FROM promotions;
- C. SELECT promo\_id. DECODE(NULLIF(promo\_cost. 10000). NULL. promo\_cost\*.25, \*N/A') "Catcost" FROM promotions;
- D. SELECT promo\_id. DECODE(promo\_cost. >10000. 'High'. <10000. 'Low') "Range"FROM promotions;

**Correct Answer: AB**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Note: there are some syntax issues in this question.

#### QUESTION 117

Evaluate the following SQL statement:

```
SQL> SELECT cust_id, cust_last_name FROM customers
WHERE cust_credit_limit IN
(select cust_credit_limit
FROM customers
WHERE cust_city='Singapore');
```

Which statement is true regarding the above query if one of the values generated by the sub query is NULL?

- A. It produces an error.
- B. It executes but returns no rows.
- C. It generates output for NULL as well as the other values produced by the sub query.
- D. It ignores the NULL value and generates output for the other values produced by the sub query.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 118

View the Exhibit and examine the structure of the PRODUCTS table. You need to generate a report in the following format:

CATEGORIES

5MP Digital Photo Camera's category is Photo

Y Box's category is Electronics

Envoy Ambassador's category is Hardware

Which two queries would give the required output? (Choose two.)

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

- A. SELECT prod\_name || q"'s category is ' || prod\_category CATEGORIES FROM products;
- B. SELECT prod\_name || q'[s ]'category is ' || prod\_category CATEGORIES FROM products;
- C. SELECT prod\_name || q'\s\ ' ' category is ' || prod\_category CATEGORIES FROM products;
- D. SELECT prod\_name || q'<'s >' || 'category is ' || prod\_category CATEGORIES FROM products;

**Correct Answer: CD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

So, how are words that contain single quotation marks dealt with? There are essentially two mechanisms

available. The most popular of these is to add an additional single quotation mark next to each naturally occurring single quotation mark in the character string Oracle offers a neat way to deal with this type of character literal in the form of the alternative quote (q) operator. Notice that the problem is that Oracle chose the single quote characters as the special pair of symbols that enclose or wrap any other character literal. These character- enclosing symbols could have been anything other than single quotation marks. Bearing this in mind, consider the alternative quote (q) operator. The q operator enables you to choose from a set of possible pairs of wrapping symbols for character literals as alternatives to the single quote symbols. The options are any single-byte or multibyte character or the four brackets: (round brackets), {curly braces}, [squarebrackets], or <angle brackets>. Using the q operator, the character delimiter can effectively be changed from a single quotation mark to any other character. The syntax of the alternative quote operator is as follows: q'delimiter'character literal which may include the single quotes delimiter' where delimiter can be any character or bracket.

#### Alternative Quote (q) Operator

Specify your own quotation mark delimiter.

Select any delimiter.

Increase readability and usability.

```
SELECT department_name || q'[ Department's Manager Id: ]'
```

```
|| manager_id
```

```
AS "Department and Manager"
```

```
FROM departments;
```

#### Alternative Quote (q) Operator

Many SQL statements use character literals in expressions or conditions. If the literal itself contains a single quotation mark, you can use the quote (q) operator and select your own quotation mark delimiter.

You can choose any convenient delimiter, single-byte or multibyte, or any of the following character pairs: [ ], { }, ( ), or < >.

In the example shown, the string contains a single quotation mark, which is normally interpreted as a delimiter of a character string. By using the q operator, however, brackets [] are used as the quotation mark delimiters.

The string between the brackets delimiters is interpreted as a literal character string.

#### QUESTION 119

The PART\_CODE column in the SPARES table contains the following list of values:

**PART\_CODE**

-----

A%\_WQ123

A%BWQ123

AB\_WQ123

Evaluate the following query:

```
SQL> SELECT part_code
```

```
FROM spares
```

```
WHERE part_code LIKE '%\%_WQ12%' ESCAPE '\';
```

Which statement is true regarding the outcome of the above query?

- A. It produces an error.
- B. It displays all values.
- C. It displays only the values A%\_WQ123 and AB\_WQ123 .
- D. It displays only the values A%\_WQ123 and A%BWQ123 .
- E. It displays only the values A%BWQ123 and AB\_WQ123.

**Correct Answer: D**

**Section: (none)**

## Explanation

### Explanation/Reference:

Explanation:

Combining Wildcard Characters

The % and \_ symbols can be used in any combination with literal characters. The example in the slide displays the names of all employees whose last names have the letter "o" as the second character.

ESCAPE Identifier

When you need to have an exact match for the actual % and \_ characters, use the ESCAPE identifier. This option specifies what the escape character is. If you want to search for strings that contain SA\_, you can use the following SQL statement:

```
SELECT employee_id, last_name, job_id
FROM employees WHERE job_id LIKE '%SA\_%' ESCAPE '\';
```

### QUESTION 120

View the Exhibit and examine the structure of the PROMOTIONS table. Evaluate the following SQL statement:

```
SQL>SELECT promo_category, AVG(promo_cost) Avg_Cost,
AVG(promo_cost)*.25 Avg_Overhead
FROM promotions
WHERE UPPER(promo_category) IN ('TV', 'INTERNET', 'POST')
GROUP BY Avg_Cost
ORDER BY Avg_Overhead;
```

The above query generates an error on execution.

Which clause in the above SQL statement causes the error?

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

- A. WHERE
- B. SELECT
- C. GROUP BY
- D. ORDER BY

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 121**

Which statement is true regarding sub queries?

- A. The LIKE operator cannot be used with single- row subqueries.
- B. The NOT IN operator is equivalent to IS NULL with single- row subqueries.
- C. =ANY and =ALL operators have the same functionality in multiple- row subqueries.
- D. The NOT operator can be used with IN, ANY, and ALL operators in multiple- row subqueries.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the ANY Operator in Multiple-Row Subqueries

The ANY operator (and its synonym, the SOME operator) compares a value to each value returned by a subquery.

<ANY means less than the maximum.

>ANY means more than the minimum.

=ANY is equivalent to IN

Using the ALL Operator in Multiple-Row Subqueries

The ALL operator compares a value to every value returned by a subquery.

>ALL means more than the maximum and

<ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators.

**QUESTION 122**

View the Exhibits and examine the structures of the PRODUCTS SALES and CUSTOMERS tables.

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(6,2)    |

| Table SALES   |          |              |
|---------------|----------|--------------|
| Name          | Null?    | Type         |
| PROD_ID       | NOT NULL | NUMBER       |
| CUST_ID       | NOT NULL | NUMBER       |
| TIME_ID       | NOT NULL | DATE         |
| CHANNEL_ID    | NOT NULL | NUMBER       |
| PROMO_ID      | NOT NULL | NUMBER       |
| QUANTITY_SOLD | NOT NULL | NUMBER(10,2) |

| 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_MARITAL_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) |

You need to generate a report that gives details of the customer's last name, name of the product, and the quantity sold for all customers in Tokyo'. Which two queries give the required result? (Choose two.)

- A. `SELECT c.cust_last_name,p.prod_name, s.quantity_sold`  
`FROM sales s JOIN products p`  
`USING(prod_id)`  
`JOIN customers c`  
`USING(cust_id)`  
`WHERE c.cust_city='Tokyo';`
- B. `SELECT c.cust_last_name, p.prod_name, s.quantity_sold`  
`FROM products p JOIN sales s JOIN customers c`  
`ON(p.prod_id=s.prod_id)`  
`ON(s.cust_id=c.cust_id)`  
`WHERE c.cust_city='Tokyo';`
- C. `SELECT c.cust_last_name, p.prod_name, s.quantity_sold`  
`FROM products p JOIN sales s`  
`ON(p.prod_id=s.prod_id)`  
`JOIN customers c`



```
ON(s.cust_id=c.cust_id)
AND c.cust_city='Tokyo';
```

- D. SELECT c.cust\_id,c.cust\_last\_name,p.prod\_id, p.prod\_name, s.quantity\_sold FROM products p JOIN sales s  
USING(prod\_id)  
JOIN customers c  
USING(cust\_id)  
WHERE c.cust\_city='Tokyo';

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 123

View the Exhibit and examine the structure of the CUSTOMERS table .Which statement would display the highest credit limit available in each income level in each city in 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_MARITAL_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) |

- A. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit ) FROM customers GROUP BY cust\_city, cust\_income\_level, cust\_credit\_limit;
- B. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_city, cust\_income\_level;
- C. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_credit\_limit, cust\_income\_level, cust\_city ;
- D. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_city, cust\_income\_level, MAX(cust\_credit\_limit);

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 124

Evaluate the following SQL statement:

```
SQL> SELECT cust_id, cust_last_name "Last Name"
FROM customers
WHERE country_id = 10
UNION
SELECT cust_id CUST_NO, cust_last_name
FROM customers
WHERE country_id = 30;
```

Which ORDER BY clause are valid for the above query? (Choose all that apply.)

- A. ORDER BY 2,1
- B. ORDER BY CUST\_NO
- C. ORDER BY 2,cust\_id
- D. ORDER BY "CUST\_NO"
- E. ORDER BY "Last Name"

**Correct Answer:** ACE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the ORDER BY Clause in Set Operations

- The ORDER BY clause can appear only once at the end of the compound query.
- Component queries cannot have individual ORDER BY clauses.
- The ORDER BY clause recognizes only the columns of the first SELECT query.
- By default, the first column of the first SELECT query is used to sort the output in an ascending order.

#### QUESTION 125

Which is the valid CREATE [TABLE] statement?

- A. CREATE TABLE emp9\$# (emp\_no NUMBER(4));
- B. CREATE TABLE 9emp\$# (emp\_no NUMBER(4));
- C. CREATE TABLE emp\*123 (emp\_no NUMBER(4));
- D. CREATE TABLE emp9\$# (emp\_no NUMBER(4). date DATE);

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Schema Object Naming Rules

Every database object has a name. In a SQL statement, you represent the name of an object with a quoted identifier or a nonquoted identifier.

A quoted identifier begins and ends with double quotation marks ("). If you name a schema object using a quoted identifier, then you must use the double quotation marks whenever you refer to that object.

A nonquoted identifier is not surrounded by any punctuation. The following list of rules applies to both quoted and nonquoted identifiers unless otherwise indicated:

Names must be from 1 to 30 bytes long with these exceptions:

Names of databases are limited to 8 bytes.

Names of database links can be as long as 128 bytes.

If an identifier includes multiple parts separated by periods, then each attribute can be up to 30 bytes long.

Each period separator, as well as any surrounding double quotation marks, counts as one byte.

For example, suppose you identify a column like this:



"schema"."table"."column"

Nonquoted identifiers cannot be Oracle Database reserved words (ANSWER D). Quoted identifiers can be reserved words, although this is not recommended. Depending on the Oracle product you plan to use to access a database object, names might be further restricted by other product-specific reserved words. The Oracle SQL language contains other words that have special meanings. These words include datatypes, schema names, function names, the dummy system table DUAL, and keywords (the uppercase words in SQL statements, such as DIMENSION, SEGMENT, ALLOCATE, DISABLE, and so forth). These words are not reserved. However, Oracle uses them internally in specific ways. Therefore, if you use these words as names for objects and object parts, then your SQL statements may be more difficult to read and may lead to unpredictable results. In particular, do not use words beginning with SYS\_ as schema object names, and do not use the names of SQL built-in functions for the names of schema objects or user-defined functions. You should use ASCII characters in database names, global database names, and database link names, because ASCII characters provide optimal compatibility across different platforms and operating systems.

Nonquoted identifiers must begin with an alphabetic character (ANSWER B - begins with 9) from your database character set. Quoted identifiers can begin with any character. Nonquoted identifiers can contain only alphanumeric characters from your database character set and the underscore (\_), dollar sign (\$), and pound sign (#). Database links can also contain periods (.) and "at" signs (@). Oracle strongly discourages you from using \$ and # in nonquoted identifiers.

Quoted identifiers can contain any characters and punctuations marks as well as spaces. However, neither quoted nor nonquoted identifiers can contain double quotation marks or the null character (\0).

Within a namespace, no two objects can have the same name. Nonquoted identifiers are not case sensitive.

Oracle interprets them as uppercase. Quoted identifiers are case sensitive. By enclosing names in double quotation marks, you can give the following names to different objects in the same namespace:

employees

"employees"

"Employees"

"EMPLOYEES"

Note that Oracle interprets the following names the same, so they cannot be used for different objects in the same namespace:

employees

EMPLOYEES

"EMPLOYEES"

Columns in the same table or view cannot have the same name. However, columns in different tables or views can have the same name.

Procedures or functions contained in the same package can have the same name, if their arguments are not of the same number and datatypes. Creating multiple procedures or functions with the same name in the same package with different arguments is called overloading the procedure or function.

### QUESTION 126

View the Exhibit to examine the description for the SALES table. Which views can have all DML operations performed on it? (Choose all that apply.)

| Table SALES   |          |              |
|---------------|----------|--------------|
| Name          | Null?    | Type         |
| PROD_ID       | NOT NULL | NUMBER       |
| CUST_ID       | NOT NULL | NUMBER       |
| TIME_ID       | NOT NULL | DATE         |
| CHANNEL_ID    | NOT NULL | NUMBER       |
| PROMO_ID      | NOT NULL | NUMBER       |
| QUANTITY_SOLD | NOT NULL | NUMBER(10,2) |

- A. CREATE VIEW v3  
AS SELECT \* FROM SALES  
WHERE cust\_id = 2034

- WITH CHECK OPTION;
- B. CREATE VIEW v1  
AS SELECT \* FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
WITH CHECK OPTION;
- C. CREATE VIEW v2  
AS SELECT prod\_id, cust\_id, time\_id FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
WITH CHECK OPTION;
- D. CREATE VIEW v4  
AS SELECT prod\_id, cust\_id, SUM(quantity\_sold) FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
GROUP BY prod\_id, cust\_id  
WITH CHECK OPTION;

**Correct Answer:** AB

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Creating a View

You can create a view by embedding a subquery in the CREATE VIEW statement.

In the syntax:

CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view

[(alias[, alias]...)]

AS subquery

[WITH CHECK OPTION [CONSTRAINT constraint]]

[WITH READ ONLY [CONSTRAINT constraint]];

OR REPLACE Re-creates the view if it already exists

FORCE Creates the view regardless of whether or not the base tables exist NOFORCE Creates the view only if the base tables exist (This is the default.) View Is the name of the view alias Specifies names for the expressions selected by the view's query (The number of aliases must match the number of expressions selected by the view.) subquery Is a complete SELECT statement (You can use aliases for the columns in the SELECT list.)

WITH CHECK OPTION Specifies that only those rows that are accessible to the view can be inserted or updated ANSWER D constraint Is the name assigned to the CHECK OPTION constraint

WITH READ ONLY Ensures that no DML operations can be performed on this view Rules for Performing DML Operations on a View

You cannot add data through a view if the view includes:

Group functions

A GROUP BY clause

The DISTINCT keyword

The pseudocolumn ROWNUM keyword

Columns defined by expressions

NOT NULL columns in the base tables that are not selected by the view ANSWER C

**QUESTION 127**

View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables. There is only one customer with the cus\_last\_name column having value Roberts. Which INSERT statement should be used to add a row into the ORDERS table for the customer whose CUST\_LAST\_NAME is Roberts and CREDIT\_LIMIT is 600?

| ORDERS      |          |               |
|-------------|----------|---------------|
| Name        | Null?    | Type          |
| ORDER_ID    | NOT NULL | NUMBER (4)    |
| ORDER_DATE  | NOT NULL | DATE          |
| ORDER_MODE  |          | VARCHAR2 (8)  |
| CUSTOMER_ID | NOT NULL | NUMBER (6)    |
| ORDER_TOTAL |          | NUMBER (8, 2) |

| CUSTOMERS       |          |               |
|-----------------|----------|---------------|
| Name            | Null?    | Type          |
| CUSTOMER_ID     | NOT NULL | NUMBER (6)    |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME  | NOT NULL | VARCHAR2 (20) |
| CREDIT_LIMIT    |          | NUMBER (9, 2) |
| CUST_ADDRESS    |          | VARCHAR2 (40) |

- A. INSERT INTO orders VALUES (l.'10-mar-2007\ 'direct'. (SELECT customerid FROM customers WHERE cust\_last\_iame='Roberts' AND credit\_limit=600). 1000);
- B. INSERT INTO orders (order\_id.order\_date.order\_mode. (SELECT customer id FROM customers WHERE cust\_last\_iame='Roberts' AND redit\_limit=600).order\_total) VALUES(L'10-mar-2007'. 'direct', &&customer\_id, 1000):
- C. INSERT INTO(SELECT o.order\_id. o.order\_date.o.order\_modex.customer\_id. ordertotal FROM orders o. customers c WHERE o.customer\_id = c.customerid AND c.cust\_la\$t\_name-RoberTs' ANDc.credit\_liinit=600) VALUES (L'10-mar-2007\ 'direct'.( SELECT customer\_id FROM customers WHERE cust\_last\_iame='Roberts' AND credit\_limit=600). 1000);
- E. INSERT INTO orders (order\_id.order\_date.order\_mode. (SELECT customer\_id FROM customers WHERE cust\_last\_iame='Roberts' AND credit\_limit=600).order\_total) VALUES(l.'10-mar-2007\ 'direct'. &customer\_id. 1000):

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 128

What is true regarding sub queries?

- A. The inner query always sorts the results of the outer query
- B. The outer query always sorts the results of the inner query
- C. The outer query must return a value to the outer query
- D. The inner query returns a value to the outer query
- E. The inner query must always return a value or the outer query will give an error

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: The inner query returns a value to the outer query. If the inner query does not return a value, the outer query does not return a result

#### QUESTION 129

View the Exhibit and examine the structure of the CUSTOMERS table. Evaluate the following SQL statement:

```
SQL> SELECT cust_city, COUNT(cust_last_name)
FROM customers
WHERE cust_credit_limit > 1000
GROUP BY cust_city
HAVING AVG(cust_credit_limit) BETWEEN 5000 AND 6000;
```

Which statement is true regarding the outcome of the above query?

| 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_MARITAL_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) |

- A. It executes successfully.
- B. It returns an error because the BETWEEN operator cannot be used in the HAVING clause.
- C. It returns an error because WHERE and HAVING clauses cannot be used in the same SELECT statement.
- D. It returns an error because WHERE and HAVING clauses cannot be used to apply conditions on the same column.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 130**

The following data exists in the PRODUCTS table:

```
PROD_ID PROD_LIST_PRICE
-----
123456 152525.99
```

You issue the following query:

```
SQL> SELECT RPAD(( ROUND(prod_list_price)), 10,'*')
FROM products
WHERE prod_id = 123456;
```

What would be the outcome?

- A. 152526\*\*\*\*
- B. \*\*152525.99
- C. 152525\*\*
- D. an error message

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

The LPAD(string, length after padding, padding string) and RPAD(string, length after padding, padding string) functions add a padding string of characters to the left or right of a string until it reaches the specified length after padding.

**QUESTION 131**

View the Exhibit and examine the structure of the PROMOTIONS table. Examine the following two SQL statements:

**Statement 1**

```
SQL>SELECT promo_category,SUM(promo_cost)
FROM promotions
WHERE promo_end_date-promo_begin_date > 30
GROUP BY promo_category;
```

**Statement 2**

```
SQL>SELECT promo_category,sum(promo_cost)
FROM promotions
GROUP BY promo_category
HAVING MIN(promo_end_date-promo_begin_date)>30;
```

Which statement is true regarding the above two SQL statements?

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

- A. statement 1 gives an error, statement 2 executes successfully
- B. statement 2 gives an error, statement 1 executes successfully
- C. statement 1 and statement 2 execute successfully and give the same output
- D. statement 1 and statement 2 execute successfully and give a different output

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 132

Examine the structure and data of the CUSTJTRANS table:

```

CUSTJTRANS
Name Null? Type
CUSTNO NOT NULL CHAR(2)
TRANSDATE DATE
TRANSAMT NUMBER(6,2)
CUSTNO TRANSDATE
TRANSAMT
11 01-JAN-07 1000
22 01-FEB-07 2000
33 01-MAR-07 3000

```

Dates are stored in the default date format dd-mon-rr in the CUSTJTRANS table. Which three SQL statements would execute successfully? (Choose three.)

- A. SELECT transdate + '10' FROM custjrans;
- B. SELECT \* FROM custjrans WHERE transdate = '01-01-07';
- C. SELECT transamt FROM custjrans WHERE custno > '11';
- D. SELECT \* FROM custjrans WHERE transdate='01-JANUARY-07';
- E. SELECT custno - 'A' FROM custjrans WHERE transamt > 2000;

**Correct Answer: ACD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 133

View the Exhibit and examine the data in the COSTS table.

| COSTS   |          |           |            |
|---------|----------|-----------|------------|
| PROD_ID | PROMO_ID | UNIT_COST | UNIT_PRICE |
| 14      | 111      | 900       | 1129       |
| 15      | 333      | 875       | 1075       |
| 16      | 333      | 700       | 900        |
| 17      | 444      | 1000      | 1150       |

You need to generate a report that displays the IDs of all products in the COSTS table whose unit price is at least 25% more than the unit cost. The details should be displayed in the descending order of 25% of the unit cost. You issue the following query:

```
SQL>SELECT prod_id
FROM costs
WHERE unit_price >= unit_cost * 1.25
ORDER BY unit_cost * 0.25 DESC;
```

Which statement is true regarding the above query?

- A. It executes and produces the required result.
- B. It produces an error because an expression cannot be used in the ORDER BY clause.
- C. It produces an error because the DESC option cannot be used with an expression in the ORDER BY clause.
- D. It produces an error because the expression in the ORDER BY clause should also be specified in the SELECT clause.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 134

Which two statements are true regarding sub queries? (Choose two.)

- A. A sub query can retrieve zero or more rows.
- B. Only two sub queries can be placed at one level.
- C. A sub query can be used only in SQL query statements.
- D. A sub query can appear\* on either side of a comparison operator.
- E. There is no limit on the number of sub query levels in the WHERE clause of a SELECT statement.

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 135

View the Exhibit and examine the data in the PRODUCTS table. You need to display product names from the

PRODUCTS table that belong to the 'Software/Other' category with minimum prices as either \$2000 or \$4000 and no unit of measure. You issue the following query:

```
SQL>SELECT prod_name, prod_category, prod_min_price
FROM products
WHERE prod_category LIKE '%Other%' AND (prod_min_price = 2000 OR
prod_min_price = 4000) AND prod_unit_of_measure <> '';
```

Which statement is true regarding the above query?

| PRODUCTS |                             |                |                |                      |
|----------|-----------------------------|----------------|----------------|----------------------|
| PROD_ID  | PROD_NAME                   | PROD_CATEGORY  | PROD_MIN_PRICE | PROD_UNIT_OF_MEASURE |
| 101      | Envoy 256MB - 40GB          | Hardware       | 6000           | Nos.                 |
| 102      | Y Box                       | Electronics    | 9000           |                      |
| 103      | DVD-R Disc, 4.7 GB          | Software/Other | 2000           | Nos.                 |
| 104      | Documentation Set - Spanish | Software/Other | 4000           |                      |

- A. It executes successfully but returns no result.
- B. It executes successfully and returns the required result.
- C. It generates an error because the condition specified for PROD\_UNIT\_OF\_MEASURE is not valid.
- D. It generates an error because the condition specified for the PROD\_CATEGORY column is not valid.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 136

You need to calculate the number of days from 1st January 2007 till date . Dates are stored in the default format of dd-mon-rr. Which two SQL statements would give the required output? (Choose two.)

- A. SELECT SYSDATE - '01-JAN-2007' FROM DUAL;
- B. SELECT SYSDATE - TOJDATE(X)1/JANUARY/2007") FROM DUAL;
- C. SELECT SYSDATE - TOJDATE('01-JANUARY-2007') FROM DUAL;
- D. SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;
- E. SELECT TO\_DATE(SYSDATE, '\*DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL;

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**



### QUESTION 137

You need to generate a list of all customer last names with their credit limits from the CUSTOMERS table. Those customers who do not have a credit limit should appear last in the list. Which two queries would achieve the required result? (Choose two.)

- A. SELECT cust\_last\_name, cust\_credit\_limit FROM customers  
ORDER BY cust\_credit\_limit DESC;
- B. SELECT cust\_last\_name, cust\_credit\_limit FROM customers  
ORDER BY cust\_credit\_limit;
- C. SELECT cust\_last\_name, cust\_credit\_limit FROM customers  
ORDER BY cust\_credit\_limit NULLS LAST;
- D. SELECT cust\_last\_name, cust\_credit\_limit FROM customers  
ORDER BY cust\_last\_name, cust\_credit\_limit NULLS LAST;

**Correct Answer:** BC

**Section:** (none)

**Explanation**

#### Explanation/Reference:

Explanation:

If the ORDER BY clause is not used, the sort order is undefined, and the Oracle server may not fetch rows in the same order for the same query twice. Use the ORDER BY clause to display the rows in a specific order.

Note: Use the keywords NULLS FIRST or NULLS LAST to specify whether returned rows containing null values should appear first or last in the ordering sequence. ANSWER C Sorting

The default sort order is ascending:

- Numeric values are displayed with the lowest values first (for example, 1 to 999).
- Date values are displayed with the earliest value first (for example, 01-JAN-92 before 01-JAN-95).

- Character values are displayed in the alphabetical order (for example, "A" first and "Z" last).
- Null values are displayed last for ascending sequences and first for descending sequences.

- ANSWER B

- You can also sort by a column that is not in the SELECT list.

### QUESTION 138

Examine the structure and data in the PRICE\_LIST table: Name Null? Type -----

PROD\_ID NOT NULL NUMBER(3)

PROD\_PRICE VARCHAR2(10)

PROD\_ID PROD PRICE

-----  
100 \$234.55

101 \$6,509.75

102 \$1,234

in the same format as the PROD\_PRICE. Which SQL statement would give the required result?

- A. SELECT TO\_CHAR(prod\_price\*.25,'\$99.999.99') FROM PRICE\_LIST;
- B. SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\*.25,'\$99.999.00') FROM PRICE\_LIST;
- C. SELECT TO\_CHAR(TO\_NUMBER(prod\_price,'\$99.999.99')\*.25,'\$99.999.00') FROM PRICE\_LIST;
- D. SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'\$99.999.99')\*.25,'\$99.999.00') FROM PRICE\_LIST;

**Correct Answer:** C

**Section:** (none)

**Explanation**

#### Explanation/Reference:

**QUESTION 139**

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?

- A. Only statement 1 executes successfully and gives the required result.
- B. Only statement 2 executes successfully and gives the required result.
- C. Both statements 1 and 2 execute successfully and give different results.
- D. Both statements 1 and 2 execute successfully and give the same required result.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 140**

Which statement is true regarding the COALESCE function?

- A. It can have a maximum of five expressions in a list.
- B. It returns the highest NOT NULL value in the list for all rows.

- C. It requires that all expressions in the list must be of the same data type.
- D. It requires that at least one of the expressions in the list must have a NOT NULL value.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

The COALESCE Function

The COALESCE function returns the first nonnull value from its parameter list. If all its parameters are null, then null is returned.

The COALESCE function takes two mandatory parameters and any number of optional parameters. The syntax is COALESCE(expr1, expr2,...,exprn), where expr1 is returned if it is not null, else expr2 if it is not null, and so on. COALESCE is a general form of the NVL function, as the following two equations illustrate:

COALESCE(expr1,expr2) = NVL(expr1,expr2)

COALESCE(expr1,expr2,expr3) = NVL(expr1,NVL(expr2,expr3))

The data type COALESCE returns if a not null value is found is the same as that of the first not null parameter.

To avoid an "ORA-00932: inconsistent data types" error, all not null parameters must have data types compatible with the first not null parameter.

#### QUESTION 141

View the Exhibit and examine the structure of ORD and ORD\_ITEMS tables. The ORD\_NO column is PRIMARY KEY in the ORD table and the ORD\_NO and ITEM\_NO columns are composite PRIMARY KEY in the ORD\_ITEMS table.

Which two CREATE INDEX statements are valid? (Choose two.)

| ORD      |          |           |
|----------|----------|-----------|
| Name     | Null?    | Type      |
| ORD_NO   | NOT NULL | NUMBER(2) |
| ORD_DATE |          | DATE      |
| CUST_ID  |          | NUMBER(4) |

| ORD_ITEMS |          |             |
|-----------|----------|-------------|
| Name      | Null?    | Type        |
| ORD_NO    | NOT NULL | NUMBER(2)   |
| ITEM_NO   | NOT NULL | NUMBER(3)   |
| QTY       |          | NUMBER(8,2) |

- A. CREATE INDEX ord\_idx1  
ON ord(ord\_no);
- B. CREATE INDEX ord\_idx2  
ON ord\_items(ord\_no);
- C. CREATE INDEX ord\_idx3  
ON ord\_items(item\_no);
- D. CREATE INDEX ord\_idx4  
ON ord,ord\_items(ord\_no, ord\_date,qty);

**Correct Answer: BC**

**Section: (none)****Explanation****Explanation/Reference:**

Explanation:

How Are Indexes Created?

You can create two types of indexes.

Unique index: The Oracle server automatically creates this index when you define a column in a table to have a PRIMARY KEY or a UNIQUE constraint. The name of the index is the name that is given to the constraint.

Nonunique index: This is an index that a user can create. For example, you can create the FOREIGN KEY column index for a join in a query to improve the speed of retrieval. Note: You can manually create a unique index, but it is recommended that you create a unique constraint, which implicitly creates a unique index.

**QUESTION 142**

View the Exhibit and examine the structure of the PROMOTIONS table. Using the PROMOTIONS table, you need to find out the names and cost of all the promos done on 'TV' and 'internet' that ended in the time interval 15th March '00 to 15th October '00. Which two queries would give the required result? (Choose two.)

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

- A. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE promo\_category IN ('TV', 'internet') AND  
promo\_end\_date BETWEEN '15-MAR-00' AND '15-OCT-00';
- B. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE promo\_category = 'TV' OR promo\_category = 'internet' AND promo\_end\_date >='15- MAR-00' OR  
promo\_end\_date <='15-OCT-00';
- C. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE (promo\_category BETWEEN 'TV' AND 'internet') AND  
(promo\_end\_date IN ('15-MAR-00','15-OCT-00'));
- D. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE (promo\_category = 'TV' OR promo\_category = 'internet') AND (promo\_end\_date >='15-MAR-00'  
AND promo\_end\_date <='15-OCT-00');

**Correct Answer: AD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 143**

You need to display the date 11-oct-2007 in words as "Eleventh of October, Two Thousand Seven". Which SQL statement would give the required result?

- A. SELECT TO\_CHAR('11-oct-2007', 'miDdsph "of Month. Year") FROM DUAL;
- B. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'X 'miDdsph of month, year') FROM DUAL;
- C. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'miDdthsp "of\* Month. Year') FROM DUAL;
- D. SELECT TO\_DATE(TO\_CHAR('11-oct-2007'fiiiDdsph "of" Month. Year')) FROM DUAL;

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 144**

View the Exhibit and examine the structure of the PRODUCTS table.

All products have a list price.

You issue the following command to display the total price of each product after a discount of 25% and a tax of 15% are applied on it. Freight charges of \$100 have to be applied to all the products.

```
SQL>SELECT prod_name, prod_list_price -(prod_list_price*(25/100))
+(prod_list_price -(prod_list_price*(25/100))*(15/100))+100
AS "TOTAL PRICE"
FROM products;
```

What would be the outcome if all the parentheses are removed from the above statement?

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

- A. It produces a syntax error.
- B. The result remains unchanged.
- C. The total price value would be lower than the correct value.
- D. The total price value would be higher than the correct value.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 145**

You need to perform these tasks:

1. Create and assign a MANAGER role to Blake and Clark
2. Grant CREATE TABLE and CREATE VIEW privileges to Blake and Clark

Which set of SQL statements achieves the desired results?

- A. CREATE ROLE manager;  
GRANT create table, create view  
TO manager;  
GRANT manager TO BLAKE,CLARK;
- B. CREATE ROLE manager;  
GRANT create table, create voew  
TO manager;  
GRANT manager ROLE TO BLAKE,CLARK;
- C. GRANT manager ROLE TO BLAKE,CLARK;  
GRANT create table, create voew  
TO BLAKE CLARK;  
\*\*\*MISSING\*\*\*

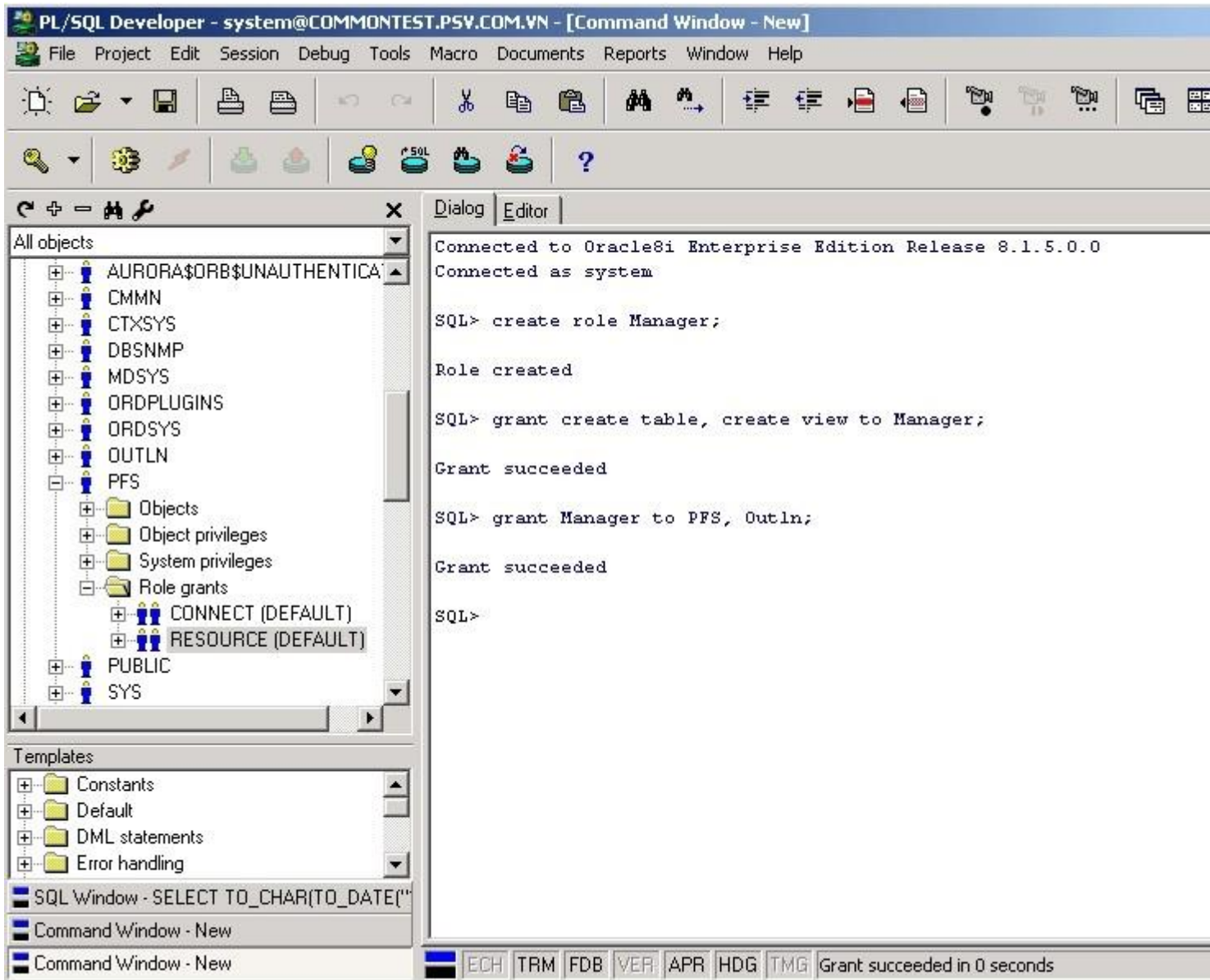
**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Result of commands:



#### QUESTION 146

Evaluate the following query:

```

SQL> SELECT promo_name || q{'s start date was }' || promo_begin_date
AS "Promotion Launches"
FROM promotions;

```

What would be the outcome of the above query?

- A. It produces an error because flower braces have been used.
- B. It produces an error because the data types are not matching.
- C. It executes successfully and introduces an 's at the end of each promo\_name in the output.
- D. It executes successfully and displays the literal "{'s start date was} " for each row in the output.

**Correct Answer: C**



**Section: (none)****Explanation****Explanation/Reference:**

Explanation:

So, how are words that contain single quotation marks dealt with? There are essentially two mechanisms available. The most popular of these is to add an additional single quotation mark next to each naturally occurring single quotation mark in the character string Oracle offers a neat way to deal with this type of character literal in the form of the alternative quote (q) operator. Notice that the problem is that Oracle chose the single quote characters as the special pair of symbols that enclose or wrap any other character literal. These character-enclosing symbols could have been anything other than single quotation marks. Bearing this in mind, consider the alternative quote (q) operator. The q operator enables you to choose from a set of possible pairs of wrapping symbols for character literals as alternatives to the single quote symbols. The options are any single-byte or multibyte character or the four brackets: (round brackets), {curly braces}, [squarebrackets], or <angle brackets>. Using the q operator, the character delimiter can effectively be changed from a single quotation mark to any other character

The syntax of the alternative quote operator is as follows:

q'delimiter'character literal which may include the single quotes delimiter' where delimiter can be any character or bracket.

Alternative Quote (q) Operator

Specify your own quotation mark delimiter.

Select any delimiter.

Increase readability and usability.

```
SELECT department_name || q'[ Department's Manager Id: ]'
```

```
|| manager_id
```

```
AS "Department and Manager"
```

```
FROM departments;
```

Alternative Quote (q) Operator

Many SQL statements use character literals in expressions or conditions. If the literal itself contains a single quotation mark, you can use the quote (q) operator and select your own quotation mark delimiter.

You can choose any convenient delimiter, single-byte or multibyte, or any of the following character pairs: [ ], { }, ( ), or < >.

In the example shown, the string contains a single quotation mark, which is normally interpreted as a delimiter of a character string. By using the q operator, however, brackets [] are used as the quotation mark delimiters. The string between the brackets delimiters is interpreted as a literal character string.

**QUESTION 147**

View the Exhibit and examine the structure of the PROMOTIONS table. Using the PROMOTIONS table, you need to display the names of all promos done after January 1, 2001, starting with the latest promo. Which query would give the required result? (Choose all that apply.)

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |



- A. SELECT promo\_name, promo\_begin\_date FROM promotions WHERE promo\_begin\_date > '01-JAN-01' ORDER BY 2 DESC;
- B. SELECT promo\_name, promo\_begin\_date FROM promotions WHERE promo\_begin\_date > '01-JAN-01' ORDER BY promo\_name DESC;
- C. SELECT promo\_name, promo\_begin\_date FROM promotions WHERE promo\_begin\_date > '01-JAN-01' ORDER BY 1DESC;
- D. SELECT promo\_name, promo\_begin\_date "START DATE" FROM promotions WHERE promo\_begin\_date > '01-JAN-01' ORDER BY "START DATE" DESC;

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 148

Evaluate the following SQL commands:

```
SQL>CREATE SEQUENCE ord_seq
INCREMENT BY 10
START WITH 120
MAXVALUE 9999
NOCYCLE;
```

```
SQL>CREATE TABLE ord_items
(ord_no NUMBER(4)DEFAULT ord_seq.NEXTVAL NOT NULL,
item_no NUMBER(3),
qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200),
expiry_date date CHECK (expiry_date > SYSDATE),
CONSTRAINT it_pk PRIMARY KEY (ord_no,item_no),
CONSTRAINT ord_fk FOREIGN KEY(ord_no) REFERENCES orders(ord_no));
```

The command to create a table fails. Identify the reason for the SQL statement failure? (Choose all that apply.)

- A. You cannot use SYSDATE in the condition of a CHECK constraint.
- B. You cannot use the BETWEEN clause in the condition of a CHECK constraint.
- C. You cannot use the NEXTVAL sequence value as a DEFAULT value for a column.
- D. You cannot use ORD\_NO and ITEM\_NO columns as a composite primary key because ORD NO is also the FOREIGN KEY.

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

CHECK Constraint

The CHECK constraint defines a condition that each row must satisfy. The condition can use the same constructs as the query conditions, with the following exceptions:

References to the CURRVAL, NEXTVAL, LEVEL, and ROWNUM pseudocolumns  
Calls to SYSDATE, UID, USER, and USERENV functions

Queries that refer to other values in other rows

A single column can have multiple CHECK constraints that refer to the column in its definition. There is no limit

to the number of CHECK constraints that you can define on a column. CHECK constraints can be defined at the column level or table level.

CREATE TABLE employees

(...

salary NUMBER(8,2) CONSTRAINT emp\_salary\_min

CHECK (salary > 0),

#### **QUESTION 149**

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

- A. addition
- B. subtraction
- C. raising to a power
- D. finding the quotient
- E. finding the lowest value

**Correct Answer:** ACE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 150**

View the Exhibit and examine the structure of the PROMOTIONS table. You need to generate a report of all promos from the PROMOTIONS table based on the following conditions:

1. The promo name should not begin with 'T' or 'N'.
2. The promo should cost more than \$20000.
3. The promo should have ended after 1st January 2001.

Which WHERE clause would give the required result?

| Table PROMOTIONS     |          |          |
|----------------------|----------|----------|
| Name                 | Null?    | Type     |
| PROMO_ID             | NOT NULL | NUMBER   |
| PROMO_NAME           | NOT NULL | VARCHAR2 |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2 |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER   |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2 |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER   |
| PROMO_COST           | NOT NULL | NUMBER   |
| PROMO_BEGIN_DATE     | NOT NULL | DATE     |
| PROMO_END_DATE       | NOT NULL | DATE     |

- A. WHERE promo\_name NOT LIKE 'T%' OR promo\_name NOT LIKE 'N%' AND promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01'
- B. WHERE (promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%')OR promo\_cost > 20000 OR promo\_end\_date > '1-JAN-01'
- C. WHERE promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%' AND promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01'
- D. WHERE (promo\_name NOT LIKE '%T%' OR promo\_name NOT LIKE '%N%') AND(promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01')

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 151

View the Exhibit and examine the structure of the SALES table. The following query is written to retrieve all those product IDs from the SALES table that have more than 55000 sold and have been ordered more than 10

times.

```
SQL> SELECT prod_id  
FROM sales  
WHERE quantity_sold > 55000 AND COUNT(*)>10  
GROUP BY prod_id  
HAVING COUNT(*)>10;
```

Which statement is true regarding this SQL statement?

| Table SALES   |          |        |
|---------------|----------|--------|
| Name          | Null?    | Type   |
| PROD_ID       | NOT NULL | NUMBER |
| CUST_ID       | NOT NULL | NUMBER |
| TIME_ID       | NOT NULL | DATE   |
| CHANNEL_ID    | NOT NULL | NUMBER |
| PROMO_ID      | NOT NULL | NUMBER |
| QUANTITY_SOLD | NOT NULL | NUMBER |

- A. It executes successfully and generates the required result.
- B. It produces an error because COUNT(\*) should be specified in the SELECT clause also.
- C. It produces an error because COUNT(\*) should be only in the HAVING clause and not in the WHERE clause.
- D. It executes successfully but produces no result because COUNT(prod\_id) should be used instead of COUNT(\*).

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Restricting Group Results with the HAVING Clause

You use the HAVING clause to specify the groups that are to be displayed, thus further restricting the groups on the basis of aggregate information. In the syntax, group\_condition restricts the groups of rows returned to those groups for which the specified condition is true.

The Oracle server performs the following steps when you use the HAVING clause:

1. Rows are grouped.
2. The group function is applied to the group.
3. The groups that match the criteria in the HAVING clause are displayed. The HAVING clause can precede the GROUP BY clause, but it is recommended that you place the GROUP BY clause first because it is more logical. Groups are formed and group functions are calculated before the HAVING clause is applied to the groups in the SELECT list. Note: The WHERE clause restricts rows, whereas the HAVING clause restricts groups.

#### QUESTION 152

You need to create a table with the following column specifications:

1. Employee ID (numeric data type) for each employee
2. Employee Name (character data type) that stores the employee name
3. Hire date, which stores the date of joining the organization for each employee
4. Status (character data type), that contains the value 'ACTIVE' if no data is entered
5. Resume (character large object [CLOB] data type), which contains the resume submitted by the employee

Which is the correct syntax to create this table?

- A. 

```
CREATE TABLE EMP_1
(emp_id NUMBER(4),
emp_name VARCHAR2(25),
start_date DATE,
e_status VARCHAR2(10) DEFAULT 'ACTIVE',
resume CLOB(200));
```
- B. 

```
CREATE TABLE 1_EMP
(emp_id NUMBER(4),
emp_name VARCHAR2(25),
start_date DATE,
emp_status VARCHAR2(10) DEFAULT 'ACTIVE',
resume CLOB);
```
- C. 

```
CREATE TABLE EMP_1
(emp_id NUMBER(4),
emp_name VARCHAR2(25),
start_date DATE,
emp_status VARCHAR2(10) DEFAULT "ACTIVE",
resume CLOB);
```
- D. 

```
CREATE TABLE EMP_1
(emp_id NUMBER,
emp_name VARCHAR2(25),
start_date DATE,
emp_status VARCHAR2(10) DEFAULT 'ACTIVE',
resume CLOB);
```

**Correct Answer:** D

**Section:** (none)

**Explanation**

#### Explanation/Reference:

Explanation:

CLOB Character data (up to 4 GB)

NUMBER [(p,s)] Number having precision p and scale s (Precision is the total number of decimal digits and scale is the number of digits to the right of the decimal point; precision can range from 1 to 38, and scale can range from 84 to 127.)

#### QUESTION 153

You need to extract details of those products in the SALES table where the PROD\_ID column contains the string '\_D123'. Which WHERE clause could be used in the SELECT statement to get the required output?

- A. WHERE prod\_id LIKE '%\_D123%' ESCAPE ' \_ '
- B. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\'
- C. WHERE prod\_id LIKE '%\_D123%' ESCAPE '%\_ '
- D. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\\_ '

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

A naturally occurring underscore character may be escaped (or treated as a regular nonspecial symbol) using the ESCAPE identifier in conjunction with an ESCAPE character. The second example in Figure 3-12 shows the SQL statement that retrieves the JOBS table records with JOB\_ID values equal to SA\_MAN and SA\_REP and which conforms to the original requirement:

```
select job_id from jobs
where job_id like 'SA\_%' escape '\';
```

#### QUESTION 154

Which three statements are true about multiple-row sub queries? (Choose three.)

- A. They can contain a subquery within a sub query.
- B. They can return multiple columns as well as rows.
- C. They cannot contain a sub query within a sub query.
- D. They can return only one column but multiple rows.
- E. They can contain group functions and GROUP BY and HAVING clauses.
- F. They can contain group functions and the GROUP BY clause, but not the HAVING clause.

**Correct Answer:** ABE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 155

Which two statements are true regarding the ORDER BY clause? (Choose two.)

- A. It is executed first in the query execution.
- B. It must be the last clause in the SELECT statement.
- C. It cannot be used in a SELECT statement containing a HAVING clause.
- D. You cannot specify a column name followed by an expression in this clause.
- E. You can specify a combination of numeric positions and column names in this clause.

**Correct Answer:** BE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 156

Which two statements are true regarding the USING clause in table joins?(Choose two.)

- A. It can be used to join a maximum of three tables.

- B. It can be used to restrict the number of columns used in a NATURAL join.
- C. It can be used to access data from tables through equijoins as well as nonequijoins.
- D. It can be used to join tables that have columns with the same name and compatible data types.

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

NATURAL JOIN operation

A NATURAL JOIN is a JOIN operation that creates an implicit join clause for you based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.

If the SELECT statement in which the NATURAL JOIN operation appears has an asterisk (\*) in the select list, the asterisk will be expanded to the following list of columns (in this order):

All the common columns

Every column in the first (left) table that is not a common column Every column in the second (right) table that is not a common column An asterisk qualified by a table name (for example, COUNTRIES.\*) will be expanded to every column of that table that is not a common column.

If a common column is referenced without being qualified by a table name, the column reference points to the column in the first (left) table if the join is an INNER JOIN or a LEFT OUTER JOIN. If it is a RIGHT OUTER JOIN, unqualified references to a common column point to the column in the second (right) table.

Syntax

TableExpression NATURAL [ { LEFT | RIGHT } [ OUTER ] | INNER ] JOIN { TableViewOrFunctionExpression | ( TableExpression ) }

Examples

If the tables COUNTRIES and CITIES have two common columns named COUNTRY and COUNTRY\_ISO\_CODE, the following two SELECT statements are equivalent:

```
SELECT * FROM COUNTRIES NATURAL JOIN CITIES
SELECT * FROM COUNTRIES JOIN CITIES
USING (COUNTRY, COUNTRY_ISO_CODE)
```

Topic 2, Volume B

Use these to reinforce the exam concepts.

### QUESTION 157

Which constraint can be defined only at the column level?

- A. UNIQUE
- B. NOT NULL
- C. CHECK
- D. PRIMARY KEY
- E. FOREIGN KEY

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The NOT NULL constraint can be defined only at the column level. It enforces that a value must be defined for this column such that the column may not be NULL for any row.

Incorrect Answers

A: The UNIQUE constraint enforces uniqueness on values in the constrained column. It can be defined not only at the column level.

C: The CHECK constraint enforces that values added to the constrained column must be present in a static list of values permitted for the column.

D: The PRIMARY KEY constraint stipulates that values in the constrained column(s) must be unique and not NULL. If the primary key applies to multiple columns, then the combination of values in the columns must be unique and not NULL.

E: The FOREIGN KEY constraint enforces that only values in the primary key of a parent table may be included as values in the constrained column(s) of the child table.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 227-232 Chapter 5: Creating Oracle Database Objects

**QUESTION 158**

What is true about the WITH GRANT OPTION clause?

- A. It allows a grantee DBA privileges.
- B. It is required syntax for object privileges.
- C. It allows privileges on specified columns of tables.
- D. It is used to grant an object privilege on a foreign key column.
- E. It allows the grantee to grant object privileges to other users and roles.

**Correct Answer: E**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

The GRANT command with the WITH GRANT OPTION clause allows the grantee to grant object privileges to other users and roles.

Incorrect Answers

A: The WITH GRANT OPTION does not allow a grantee DBA privileges.

B: It is not required syntax for object privileges. It is optional clause of GRANT command.

C: GRANT command does not allow privileges on columns of tables.

D: It is not used to grant an object privilege on a foreign key column.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 356-365 Chapter 8: User Access in Oracle

**QUESTION 159**

Which substitution variable would you use if you want to reuse the variable without prompting the user each time?

- A. &
- B. ACCEPT
- C. PROMPT
- D. &&

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

To reuse the variable without prompting the user each time you can use && substitution variable.



Incorrect Answers

A: This substitution variable will prompt the user each time.

B: ACCEPT is command, not substitution variable. It used to define more accurate or specific prompt or when you want more output to display as the values are defined.

C: PROMPT is part of the ACCEPT command, it is not a variable.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 165-173 Chapter 4: Sub queries

#### **QUESTION 160**

The EMPLOYEES table has these columns:

LAST NAME VARCHAR2(35)

SALARY NUMBER(8,2)

HIRE\_DATE DATE

Management wants to add a default value to the SALARY column. You plan to alter the table by using this SQL statement:

```
ALTER TABLE EMPLOYEES  
MODIFY (SALARY DEFAULT 5000);
```

What is true about your ALTER statement?

A. Column definitions cannot be altered to add DEFAULT values.

B. A change to the DEFAULT value affects only subsequent insertions to the table.

C. Column definitions cannot be altered to add DEFAULT values for columns with a NUMBER data type.

D. All the rows that have a NULL value for the SALARY column will be updated with the value 5000.

**Correct Answer: B**

**Section: (none)**

**Explanation**

#### **Explanation/Reference:**

Explanation:

A change to the DEFAULT value affects only subsequent insertions to the table. Existing rows will not be affected.

Incorrect Answers

A: Column definitions can be altered to add DEFAULT values.

C: Column definitions can be altered to add DEFAULT values. It works for columns with a NUMBER data type also.

D: A change to the DEFAULT value affects only subsequent insertions to the table. Existing rows will not be affected.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 219-224 Chapter 5: Creating Oracle Database Objects

#### **QUESTION 161**

The DBA issues this SQL command:

```
CREATE USER Scott  
IDENTIFIED BY tiger;
```

What privileges does the user Scott have at this point?

A. No privileges.

B. Only the SELECT privilege.

- C. Only the CONNECT privilege.
- D. All the privileges of a default user.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

There are no privileges for the user Scott at this point. They are not added themselves to the user immediately after creation. The DBA needs to grant all privileges explicitly.

Incorrect Answers

B: There are no privileges for the user Scott at this point. SELECT privilege needs to be added to the user Scott.

C: There are no privileges for the user Scott at this point. CONNECT privilege needs to be added to the user Scott.

D: There is no default user in Oracle.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 348-351 Chapter 8: User Access in Oracle

#### **QUESTION 162**

Which two statements complete a transaction? (Choose two)

- A. DELETE employees;
- B. DESCRIBE employees;
- C. ROLLBACK TO SAVEPOINT C;
- D. GRANT SELECT ON employees TO SCOTT;
- E. ALTER TABLE employees  
SET UNUSED COLUMN sal;
- F. Select MAX(sal)  
FROM employees  
WHERE department\_id = 20;

**Correct Answer:** DE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

D: GRANT is a DML operation which will cause an implicit commit

E: It is important to understand that an implicit COMMIT occurs on the database when a user exits SQL\*Plus or issues a data-definition language (DDL) command such as a CREATE TABLE statement, used to create a database object, or an ALTER TABLE statement, used to alter a database object.

Incorrect Answers

A: The DELETE command is data-manipulation language (DML) command and it does not complete a transaction.

B: The DESCRIBE command is internal SQL\*Plus command and it has nothing to do with completion a transaction.

C: ROLLBACK is not used to commit or complete a transaction, it is used to undo a transaction

F: SELECT command is used to retrieve data. It does not complete a transaction.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 281-282 Chapter 3: Advanced Data Selection in Oracle

**QUESTION 163**

You need to produce a report for mailing labels for all customers. The mailing label must have only the customer name and address. The CUSTOMERS table has these columns:

CUST\_ID NUMBER(4) NOT NULL  
CUST\_NAME VARCHAR2(100) NOT NULL  
CUST\_ADDRESS VARCHAR2(150)  
CUST\_PHONE VARCHAR2(20)

Which SELECT statement accomplishes this task?

- A. SELECT \* FROM customers
- B. SELECT name, address FROM customers;
- C. SELECT id, name, address, phone FROM customers;
- D. SELECT cust\_name, cust\_address FROM customers;
- E. SELECT cust\_id, cust\_name, cust\_address, cust\_phone FROM customers;

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

This answer provides correct list of columns for the output.

Incorrect Answers

A: This answer does not provide correct list of columns for the output. It is not required to show all columns of the table. Symbol "\*" is used in the SELECT command to substitute a list of all columns of the table.

B: This answer does not provide correct list of columns for the output. There are not NAME and ADDRESS columns in the CUSTOMERS table.

C: This answer does not provide correct list of columns for the output. There are not ID, NAME, ADDRESS or PHONE columns in the CUSTOMERS table.

E: This answer does not provide correct list of columns for the output. It is not required to show all columns of the table.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 20-24 Chapter 1: Overview of Oracle Databases

**QUESTION 164**

Which statement describes the ROWID data type?

- A. Binary data up to 4 gigabytes.
- B. Character data up to 4 gigabytes.
- C. Raw binary data of variable length up to 2 gigabytes.
- D. Binary data stored in an external file, up to 4 gigabytes.
- E. A hexadecimal string representing the unique address of a row in its table.

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The ROWID datatype stores information related to the disk location of table rows. They also uniquely identify the rows in your table. The ROWID datatype is stored as a hexadecimal string.

Incorrect Answers

A: It is not a binary data. The ROWID datatype is a hexadecimal string.

- B: It is not a character data. The ROWID datatype is a hexadecimal string.  
C: It is not a raw binary data. The ROWID datatype is a hexadecimal string.  
D: It is not binary data stored in an external file. The ROWID datatype is a hexadecimal string.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 216 Chapter 5: Creating Oracle Database Objects

#### **QUESTION 165**

Evaluate the SQL statement  
DROP TABLE DEPT:

Which four statements are true of the SQL statement? (Choose four)

- A. You cannot roll back this statement.
- B. All pending transactions are committed.
- C. All views based on the DEPT table are deleted.
- D. All indexes based on the DEPT table are dropped.
- E. All data in the table is deleted, and the table structure is also deleted.
- F. All data in the table is deleted, but the structure of the table is retained.
- G. All synonyms based on the DEPT table are deleted.

**Correct Answer:** ABDE

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

You cannot roll back DROP TABLE statement. All pending transactions related on this table are committed. If the table is dropped, Oracle automatically drops any index, trigger and constraint associated with the table as well. All data in the table is deleted, and the table structure is also deleted.

Incorrect Answers

- C: All views based on the DEPT table become invalid, but they are not deleted.  
F: All data in the table is deleted, and the table structure is also deleted. Command TRUNCATE deletes all data in the table, but does not delete the structure of the table.  
G: All synonyms based on the DEPT table are not deleted after dropping the table. OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 225 Chapter 5: Creating Oracle Database Objects

#### **QUESTION 166**

You are granted the CREATE VIEW privilege. What does this allow you to do?

- A. Create a table view.
- B. Create a view in any schema.
- C. Create a view in your schema.
- D. Create a sequence view in any schema.
- E. Create a view that is accessible by everyone.
- F. Create a view only of it is based on tables that you created.

**Correct Answer:** C

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

You can create a view in your own schema only if you are granted the CREATE VIEW privilege.

#### Incorrect Answers

A: You can create a view in your own schema only.

B: You can create a view in your own schema only, not in any schema.

D: There is no sequence view in Oracle.

E: You cannot create a view that is accessible by everyone. You will need specially grant SELECT privileges on this view for everyone.

F: You can create a view in your own schema, but not only for tables in your schema. You can use object from other users schemas if you have privileges to retrieve data from them.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 292-301 Chapter 7: Creating Other Database Objects in Oracle

#### QUESTION 167

Which two statements about creating constraints are true? (Choose two)

A. Constraint names must start with SYS\_C.

B. All constraints must be defines at the column level.

C. Constraints can be created after the table is created.

D. Constraints can be created at the same time the table is created.

E. Information about constraints is found in the VIEW\_CONSTRAINTS dictionary view.

**Correct Answer:** CD

**Section:** (none)

**Explanation**

#### Explanation/Reference:

Explanation:

Constraints can be created after the table is created. Use ALTER TABLE command for that. Constraints can be created at the same time the table is created (CREATE TABLE command).

#### Incorrect Answers

A: There is no requirements in Oracle that constraint names must start with SYS\_C. Oracle can use prefix "SYS" to build indexes for UNIQUE and NOT NULL constraints, but it is not required for user to follow this naming rule.

B: Not all constraints must be defines at the column level. Only NOT NULL constraint must be.

E: There is no VIEW\_CONSTRAINTS dictionary view in Oracle.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 227-232 Chapter 5: Creating Oracle Database Objects

#### QUESTION 168

Which two statements are true about WHERE and HAVING clauses? (Choose two)

A. A WHERE clause can be used to restrict both rows and groups.

B. A WHERE clause can be used to restrict rows only.

C. A HAVING clause can be used to restrict both rows and groups.

D. A HAVING clause can be used to restrict groups only.

E. A WHERE clause CANNOT be used in a query of the query uses a HAVING clause.

F. A HAVING clause CANNOT be used in sub queries.

**Correct Answer:** BD

**Section:** (none)

**Explanation**

#### Explanation/Reference:

Explanation:

B: WHERE clause cannot be use to restrict groups

WHERE clause cannot be use when there is group functions.

D: A HAVING clause can only e used to restrict GROUPS.

Note: HAVING clause to specify which groups are to be displayed and thus further restrict the groups on the basis of aggregate information. The Oracle server performs the following steps when you use the Having clause

1. rows are grouped
2. the group function is applied to the group
3. the group that match the criteria in the Having clause are displayed.

Incorrect Answers :

A. Where clause cannot be use to restrict groups

C. A HAVING clause can only e used to restrict GROUPS.

E. WHERE clause cannot be use when there is group function, instead HAVING is to be use. F. There is no constraint to use HAVING clause in a sub queries.

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Aggregating Data using Group Functions, p. 5-20

#### QUESTION 169

EMPLOYEES and DEPARTMENTS data:

EMPLOYEES

| EMPLOYEE_ID | EMP_NAME | DEPT_ID | MGR_ID | JOB_ID   | SALARY |
|-------------|----------|---------|--------|----------|--------|
| 101         | Smith    | 20      | 120    | SA_REP   | 4000   |
| 102         | Martin   | 10      | 105    | CLERK    | 2500   |
| 103         | Chris    | 20      | 120    | IT_ADMIN | 4200   |
| 104         | John     | 30      | 108    | HR_CLERK | 2500   |
| 105         | Diana    | 30      | 108    | IT_ADMIN | 5000   |
| 106         | Smith    | 40      | 110    | AD_ASST  | 3000   |
| 108         | Jennifer | 30      | 110    | HR_DIR   | 6500   |
| 110         | Bob      | 40      |        | EX_DIR   | 8000   |
| 120         | Ravi     | 20      | 110    | SA_DIR   | 6500   |

DEPARTMENTS

| DEPARTMENT_ID | DEPARTMENT_NAME |
|---------------|-----------------|
| 10            | Admin           |
| 20            | Education       |
| 30            | IT              |
| 40            | Human Resources |

On the EMPLOYEES table, EMPLOYEE\_ID is the primary key. MGR\_ID is the ID managers and refers to the EMPLOYEE\_ID.

On the DEPARTMENTS table DEPARTMENT\_ID is the primary key.

Evaluate this UPDATE statement.

```

UPDATE employees
SET mgr_id =
(SELECT mgr_id
FROM employees
WHERE dept_id=
(SELECT department_id
FROM departments
WHERE department_name = 'Administration')),
Salary = (SELECT salary
FROM employees
WHERE emp_name = 'Smith')
WHERE job_id = 'IT_ADMIN';

```

What happens when the statement is executed?

- A. The statement executes successfully, leaves the manager ID as the existing value, and changes the salary to 4000 for the employees with ID 103 and 105.
- B. The statement executes successfully, changes the manager ID to NULL, and changes the salary to 4000 for the employees with ID 103 and 105.
- C. The statement executes successfully, changes the manager ID to NULL, and changes the salary to 3000 for the employees with ID 103 and 105.
- D. The statement fails because there is more than one row matching the employee name Smith.
- E. The statement fails because there is more than one row matching the IT\_ADMIN job ID in the EMPLOYEES table.
- F. The statement fails because there is no 'Administration' department in the DEPARTMENTS table.

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

'=' is use in the statement and sub query will return more than one row. Employees table has 2 row matching the employee name Smith.

The update statement will fail.

Incorrect Answers :

- A. The Update statement will fail no update was done.
- B. The update statement will fail no update was done.
- C. The update statement will fail no update was done.
- E. The update statement will fail but not due to job\_id='IT\_ADMIN' F. The update statement will fail but not due to department\_id='Administration'

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Sub queries, p. 6-12

### QUESTION 170

Which SQL statement accepts user input for the columns to be displayed, the table name, and WHERE condition?

- A. SELECT &1, "&2"  
FROM &3  
WHERE last\_name = '&4';
- B. SELECT &1, '&2'  
FROM &3  
WHERE '&last\_name = '&4' ';
- C. SELECT &1, &2  
FROM &3

```
WHERE last_name = '&4';  
D. SELECT &1, '&2'  
FROM EMP  
WHERE last_name = '&4';
```

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

In a WHERE clause, date and characters values must be enclosed within single quotation marks.

Sample of the correct syntax

```
SELECT EMPLOYEE_ID, &COLUMN_NAME  
FROM EMPLOYEES
```

Incorrect Answers :

- A. Incorrect use of " symbol
- B. Incorrect use of ' symbol
- D. No input for table name as EMP has been use in the statement.

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Producing Readable Output with iSQL\*PLUS, p. 7-8

#### **QUESTION 171**

Evaluate the SQL statement:

```
SELECT ROUND(45.953, -1), TRUNC(45.936, 2)  
FROM dual;
```

Which values are displayed?

- A. 46 and 45
- B. 46 and 45.93
- C. 50 and 45.93
- D. 50 and 45.9
- E. 45 and 45.93
- F. 45.95 and 45.93

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

ROUND (45.953,-1) will round value to 1 decimal places to the left.

TRUNC (45.936,2) will truncate value to 2 decimal

The answer will be 50 and 45.93

Incorrect Answers :

- A. Does not meet round and truncate functions
- B. Does not meet round functions
- D. Does not meet truncate functions
- E. Does not meet round functions
- F. Does not meet round functions



Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Single-Row functions, p. 3-13

#### QUESTION 172

The CUSTOMERS table has these columns:

|                  |                |          |
|------------------|----------------|----------|
| CUSTOMER_ID      | NUMBER (4)     | NOT NULL |
| CUSTOMER_NAME    | VARCHAR2 (100) | NOT NULL |
| STREET_ADDRESS   | VARCHAR2 (150) |          |
| CITY_ADDRESS     | VARCHAR2 (50)  |          |
| STATE_ADDRESS    | VARCHAR2 (50)  |          |
| PROVINCE_ADDRESS | VARCHAR2 (50)  |          |
| COUNTRY_ADDRESS  | VARCHAR2 (50)  |          |
| POSTAL_CODE      | VARCHAR2 (12)  |          |
| CUSTOMER_PHONE   | VARCHAR2 (20)  |          |

The CUSTOMER\_ID column is the primary key for the table.

You need to determine how dispersed your customer base is. Which expression finds the number of different countries represented in the CUSTOMERS table?

- A. COUNT(UPPER(country\_address))
- B. COUNT(DIFF(UPPER(country\_address)))
- C. COUNT(UNIQUE(UPPER(country\_address)))
- D. COUNT DISTINCT UPPER(country\_address)
- E. COUNT(DISTINCT (UPPER(country\_address)))

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 173

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

### EMPLOYEES

|             |               |             |
|-------------|---------------|-------------|
| EMPLOYEE_ID | NUMBER        | Primary Key |
| FIRST_NAME  | VARCHAR2 (25) |             |
| LAST_NAME   | VARCHAR2 (25) |             |
| HIRE_DATE   | DATE          |             |

### NEW EMPLOYEES

|             |               |             |
|-------------|---------------|-------------|
| EMPLOYEE_ID | NUMBER        | Primary Key |
| NAME        | VARCHAR2 (60) |             |

Which DELETE statement is valid?

- A. DELETE FROM employees  
WHERE employee\_id = (SELECT employee\_id  
FROM employees);
- B. DELETE \* FROM employees  
WHERE employee\_id = (SELECT employee\_id  
FROM new\_employees);
- C. DELETE FROM employees  
WHERE employee\_id IN (SELECT employee\_id  
FROM new\_employees  
WHERE name = 'Carrey');
- D. DELETE \* FROM employees  
WHERE employee\_id IN (SELECT employee\_id  
FROM new\_employees  
WHERE last\_name = 'Carrey');

**Correct Answer: C**

**Section: (none)**

**Explanation**

#### **Explanation/Reference:**

Explanation:

The correct syntax for DELETE statement

DELETE [ FROM ] table

[ WHERE condition ];

Incorrect Answers :

A. '=' is use in the statement and sub query will return more than one row. Error Ora-01427: single-row sub query returns more than one row.

B. Incorrect DELETE statement

D. Incorrect DELETE statement

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Manipulating Data, p. 8-

#### **QUESTION 174**

Examine the structure of the EMP\_DEPT\_VU view:

| Column Name   | Type         | Remarks                    |
|---------------|--------------|----------------------------|
| EMPLOYEE_ID   | NUMBER       | From the EMPLOYEES table   |
| EMP_NAME      | VARCHAR2(30) | From the EMPLOYEES table   |
| JOB_ID        | VARCHAR2(20) | From the EMPLOYEES table   |
| SALARY        | NUMBER       | From the EMPLOYEES table   |
| DEPARTMENT_ID | NUMBER       | From the DEPARTMENTS table |
| DEPT_NAME     | VARCHAR2(30) | From the DEPARTMENTS table |

Which SQL statement produces an error?

- A. SELECT \*  
FROM emp\_dept\_vu;
- B. SELECT department\_id, SUM(salary)  
FROM emp\_dept\_vu  
GROUP BY department\_id;
- C. SELECT department\_id, job\_id, AVG(salary)  
FROM emp\_dept\_vu  
GROUP BY department\_id, job\_id;
- D. SELECT job\_id, SUM(salary)  
FROM emp\_dept\_vu  
WHERE department\_id IN (10,20)  
GROUP BY job\_id  
HAVING SUM(salary) > 20000;
- E. None of the statements produce an error; all are valid.

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: None of the statements produce an error.

Incorrect answer:

- A Statement will not cause error
- B Statement will not cause error
- C Statement will not cause error
- D Statement will not cause error

#### **QUESTION 175**

You own a table called EMPLOYEES with this table structure:

```
EMPLOYEE_ID NUMBER Primary Key
FIRST_NAME VARCHAR2(25)
LAST_NAME VARCHAR2(25)
HIRE_DATE DATE
```

What happens when you execute this DELETE statement?

```
DELETE employees;
```

- A. You get an error because of a primary key violation.
- B. The data and structure of the EMPLOYEES table are deleted.
- C. The data in the EMPLOYEES table is deleted but not the structure.

D. You get an error because the statement is not syntactically correct.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: You can remove existing rows from a table by using the DELETE statement.

DELETE [FROM] table

[WHERE condition];

Incorrect answer:

A Statement will not cause error

B Delete statement will not delete the table structure

D Statement will not cause error

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-19

### QUESTION 176

Examine the structure of the EMPLOYEES and NEW\_EMPLOYEES tables:

EMPLOYEES

EMPLOYEE\_ID NUMBER Primary Key

FIRST\_NAME VARCHAR2(25)

LAST\_NAME VARCHAR2(25)

HIRE\_DATE DATE

NEW\_EMPLOYEES

EMPLOYEE\_ID NUMBER Primary Key

NAME VARCHAR2(60)

Which MERGE statement is valid?

- A. MERGE INTO new\_employees c USING employees e ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN UPDATE SET c.name = e.first\_name ||', '|| e.last\_name WHEN NOT MATCHED THEN INSERT VALUES (e.employee\_id, e.first\_name ||', '||e.last\_name);
- B. MERGE new\_employees c USING employees e ON (c.employee\_id = e.employee\_id) WHEN EXISTS THEN UPDATE SET c.name = e.first\_name ||', '|| e.last\_name WHEN NOT MATCHED THEN INSERT VALUES (e.employee\_id, e.first\_name ||', '||e.last\_name);
- C. MERGE INTO new\_employees c USING employees e ON (c.employee\_id = e.employee\_id) WHEN EXISTS THEN UPDATE SET c.name = e.first\_name ||', '|| e.last\_name WHEN NOT MATCHED THEN INSERT VALUES(e.employee\_id, e.first\_name ||', '||e.last\_name);
- D. MERGE new\_employees c FROM employees e ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN UPDATE SET c.name = e.first\_name ||', '|| e.last\_name WHEN NOT MATCHED THEN INSERT INTO new\_employees VALUES (e.employee\_id, e.first\_name ||', '||
- E. last\_name);

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The correct statement for MERGE is

MERGE INTO table\_name

Incorrect answer:

B Wrong statement with the keyword EXISTS

C Wrong statement with the keyword EXISTS

D Wrong statement on the MERGE new\_employees

**QUESTION 177**

The EMPLOYEES table contains these columns:

EMPLOYEE\_ID NUMBER(4)  
ENAME VARCHAR2 (25)  
JOB\_ID VARCHAR2(10)

Which SQL statement will return the ENAME, length of the ENAME, and the numeric position of the letter "a" in the ENAME column, for those employees whose ENAME ends with a the letter "n"?

- A. SELECT ENAME, LENGTH(ENAME), INSTR(ENAME, 'a') FROM EMPLOYEES WHERE SUBSTR(ENAME, -1, 1) = 'n';
- B. SELECT ENAME, LENGTH(ENAME), INSTR(ENAME, , -1, 1) FROM EMPLOYEES WHERE SUBSTR(ENAME, -1, 1) = 'n';
- C. SELECT ENAME, LENGTH(ENAME), SUBSTR(ENAME, -1, 1) FROM EMPLOYEES WHERE INSTR(ENAME, 1, 1) = 'n';
- D. SELECT ENAME, LENGTH(ENAME), SUBSTR(ENAME, -1, 1) FROM EMPLOYEES WHERE INSTR(ENAME, -1, 1) = 'n';

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

INSTR is a character function return the numeric position of a named string.

INSTR(NAMED, 'a')

Incorrect answer:

B Did not return a numeric position for 'a'.

C Did not return a numeric position for 'a'.

D Did not return a numeric position for 'a'.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 3-8

**QUESTION 178**

Which four are valid Oracle constraint types? (Choose four.)

- A. CASCADE
- B. UNIQUE
- C. NONUNIQUE
- D. CHECK
- E. PRIMARY KEY
- F. CONSTANT
- G. NOT NULL

**Correct Answer:** BDEG

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Oracle constraint type is Not Null, Check, Primary Key, Foreign Key and Unique

Incorrect answer:

A Is not Oracle constraint

C Is not Oracle constraint

F Is not Oracle constraint

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-3

#### **QUESTION 179**

Which SQL statement would you use to remove a view called EMP\_DEPT\_VU from your schema?

- A. DROP emp\_dept\_vu;
- B. DELETE emp\_dept\_vu;
- C. REMOVE emp\_dept\_vu;
- D. DROP VIEW emp\_dept\_vu;
- E. DELETE VIEW emp\_dept\_vu;
- F. REMOVE VIEW emp\_dept\_vu;

**Correct Answer:** D

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

DROP VIEW viewname;

Incorrect answer:

A Not a valid drop view statement

B Not a valid drop view statement

C Not a valid drop view statement

E Not a valid drop view statement

F Not a valid drop view statement

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-20

#### **QUESTION 180**

Which is an iSQL\*Plus command?

- A. INSERT
- B. UPDATE
- C. SELECT
- D. DESCRIBE
- E. DELETE
- F. RENAME

**Correct Answer:** D

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation: The only SQL\*Plus command in this list : DESCRIBE. It cannot be used as SQL command. This command returns a description of table name, including all columns in that table, the datatype for each column and an indication of whether the column permits storage of NULL values.

Incorrect answer:

A INSERT is not a SQL\*PLUS command

B UPDATE is not a SQL\*PLUS command

C SELECT is not a SQL\*PLUS command

E DELETE is not a SQL\*PLUS command

F RENAME is not a SQL\*PLUS command

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7

**QUESTION 181**

Examine the description of the EMPLOYEES table:

```
EMP_ID NUMBER(4) NOT NULL
LAST_NAME VARCHAR2(30) NOT NULL
FIRST_NAME VARCHAR2(30)
DEPT_ID NUMBER(2)
```

Which statement produces the number of different departments that have employees with last name Smith?

- A. SELECT COUNT(\*) FROM employees WHERE last\_name='Smith';
- B. SELECT COUNT (dept\_id) FROM employees WHERE last\_name='Smith';
- C. SELECT DISTINCT(COUNT(dept\_id)) FROM employees WHERE last\_name='Smith';
- D. SELECT COUNT(DISTINCT dept\_id) FROM employees WHERE last\_name='Smith';
- E. SELECT UNIQUE(dept\_id) FROM employees WHERE last\_name='Smith';

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 182**

Top N analysis requires \_\_\_\_\_ and \_\_\_\_\_. (Choose two.)

- A. the use of rowid
- B. a GROUP BY clause
- C. an ORDER BY clause
- D. only an inline view
- E. an inline view and an outer query

**Correct Answer:** CE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The correct statement for Top-N Analysis

```
SELECT [column_list], ROWNUM
```

```
FROM (SELECT [column_list]
```

```
FROM table
```

```
ORDER BY Top-N_column)
```

```
WHERE ROWNUM <= N;
```

Incorrect answer:

A ROWID is not require

B GROUP BY clause is not require

D Must have inline view and outer query.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-23

**QUESTION 183**

Which statement adds a constraint that ensures the CUSTOMER\_NAME column of the CUSTOMERS table holds a value?

- A. ALTER TABLE customers ADD CONSTRAINT cust\_name\_nn CHECK customer\_name IS NOT NULL;
- B. ALTER TABLE customers MODIFY CONSTRAINT cust\_name\_nn CHECK customer\_name IS NOT NULL;
- C. ALTER TABLE customers MODIFY customer\_name CONSTRAINT cust\_name\_nn NOT NULL;
- D. ALTER TABLE customers MODIFY customer\_name CONSTRAINT cust\_name\_nn IS NOT NULL;
- E. ALTER TABLE customers MODIFY name CONSTRAINT cust\_name\_nn NOT NULL;
- F. ALTER TABLE customers ADD CONSTRAINT cust\_name\_nn CHECK customer\_name NOT NULL;

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 184

Evaluate this SQL statement:

```
SELECT ename, sal, 12*sal+100 FROM emp;
```

The SAL column stores the monthly salary of the employee. Which change must be made to the above syntax to calculate the annual compensation as "monthly salary plus a monthly bonus of \$100, multiplied by 12"?

- A. No change is required to achieve the desired results.
- B. SELECT ename, sal, 12\*(sal+100) FROM emp;
- C. SELECT ename, sal, (12\*sal)+100 FROM emp;
- D. SELECT ename, sal+100,\*12 FROM emp;

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

to achieve the result you must add 100 to sal before multiply with 12.

Select ename, sal, 12\*(sal+100) from EMP;

Incorrect answer:

A Multiplication and division has priority over addition and subtraction in Operator precedence.

C Give wrong results

D Wrong syntax

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 1-11

#### QUESTION 185

You are the DBA for an academic database. You need to create a role that allows a group of users to modify existing rows in the STUDENT\_GRADES table.

Which set of statements accomplishes this?

- A. CREATE ROLE registrar; GRANT MODIFY ON student\_grades TO registrar; GRANT registrar to user1, user2, user3
- B. CREATE NEW ROLE registrar; GRANT ALL ON student\_grades TO registrar; GRANT registrar to user1, user2, user3
- C. CREATE ROLE registrar; GRANT UPDATE ON student\_grades TO registrar; GRANT ROLE registrar to user1, user2, user3
- D. CREATE ROLE registrar; GRANT UPDATE ON student\_grades TO registrar; GRANT registrar to user1, user2, user3;



E. CREATE registrar; GRANT CHANGE ON student\_grades TO registrar; GRANT registrar;

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

this is the correct solution for the answer.

GRANT role\_name to users;

Incorrect answer:

A there is no such MODIFY keyword

B invalid CREATE command, there is no such NEW keyword

C invalid GRANT command, there is no such ROLE keyword

E invalid GRANT command, there is no such CHANGE keyword

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-10

#### **QUESTION 186**

You need to modify the STUDENTS table to add a primary key on the STUDENT\_ID column.  
The table is currently empty.

Which statement accomplishes this task?

A. ALTER TABLE students ADD PRIMARY KEY student\_id;

B. ALTER TABLE students ADD CONSTRAINT PRIMARY KEY (student\_id);

C. ALTER TABLE students ADD CONSTRAINT stud\_id\_pk PRIMARY KEY student\_id;

D. ALTER TABLE students ADD CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

E. ALTER TABLE students MODIFY CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

ALTER TABLE table\_name

ADD [CONSTRAINT constraint] type (column);

Incorrect answer:

A wrong syntax

B wrong syntax

C wrong syntax

E no such MODIFY keyword

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-17

#### **QUESTION 187**

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

The registrar requested a report listing the students' grade point averages (GPA) sorted from highest grade point average to lowest.

Which statement produces a report that displays the student ID and GPA in the sorted order requested by the

registrar?

- A. SELECT student\_id, gpa FROM student\_grades ORDER BY gpa ASC;
- B. SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa ASC;
- C. SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa;
- D. SELECT student\_id, gpa FROM student\_grades ORDER BY gpa;
- E. SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa DESC;
- F. SELECT student\_id, gpa FROM student\_grades ORDER BY gpa DESC;

**Correct Answer:** F

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

sorted by highest to lowest is DESCENDING order

Incorrect answer:

A result in ascending order

B wrong syntax with SORT keyword

C wrong syntax with SORT keyword

D default value for ORDER by is in ascending order

E wrong syntax with SORT keyword

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-22

#### **QUESTION 188**

Which describes the default behavior when you create a table?

- A. The table is accessible to all users.
- B. Tables are created in the public schema.
- C. Tables are created in your schema.
- D. Tables are created in the DBA schema.
- E. You must specify the schema when the table is created.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

sorted by highest to lowest is DESCENDING order

Incorrect answer:

A grant the table privilege to PUBLIC

B login as sysoper

D login as DBA or sysdba

E no such option is allow.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 9-9

#### **QUESTION 189**

Which four are attributes of single row functions? (Choose four.)

- A. cannot be nested
- B. manipulate data items

- C. act on each row returned
- D. return one result per row
- E. accept only one argument and return only one value
- F. accept arguments which can be a column or an expression

**Correct Answer:** BCDF

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

manipulate data items, act on each row returned, return one result per row, and accept arguments that can be a column or expression.

Incorrect answer:

A is not single row attributes

E functions can accept more than one argument, e.g NVL2

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 3-5

### QUESTION 190

You need to create a table named ORDERS that contain four columns:

1. an ORDER\_ID column of number data type
2. a CUSTOMER\_ID column of number data type
3. an ORDER\_STATUS column that contains a character data type
4. a DATE\_ORDERED column to contain the date the order was placed.

When a row is inserted into the table, if no value is provided when the order was placed, today's date should be used instead.

Which statement accomplishes this?

- A. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE = SYSDATE);
- B. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE DEFAULT SYSDATE);
- C. CREATE OR REPLACE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE DEFAULT SYSDATE);
- D. CREATE OR REPLACE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status VARCHAR2 (10),  
date\_ordered DATE = SYSDATE);
- E. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status NUMBER (10),  
date\_ordered DATE = SYSDATE);

F. CREATE TABLE orders (  
order\_id NUMBER (10),  
customer\_id NUMBER (8),  
order\_status NUMBER (10),  
date\_ordered DATE DEFAULT SYSDATE);

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Requirement that Order\_Status should be a character data type

Not E: Order\_status must be a character data type. There is also a syntax error.

### QUESTION 191

Examine the structure of the EMPLOYEES table:

EMPLOYEE\_ID NUMBER Primary Key  
FIRST\_NAME VARCHAR2(25)  
LAST\_NAME VARCHAR2(25)

Which three statements insert a row into the table? (Choose three.)

- A. INSERT INTO employees VALUES ( NULL, 'John', 'Smith');
- B. INSERT INTO employees( first\_name, last\_name) VALUES( 'John', 'Smith');
- C. INSERT INTO employees VALUES ( 1000, 'John', NULL);
- D. INSERT INTO employees (first\_name, last\_name, employee\_id) VALUES ( 1000, 'John', 'Smith');
- E. INSERT INTO employees (employee\_id) VALUES (1000);
- F. INSERT INTO employees (employee\_id, first\_name, last\_name) VALUES ( 1000, 'John', '');

**Correct Answer:** CEF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: EMPLOYEE\_ID is a primary key.

Incorrect answer:

A EMPLOYEE\_ID cannot be null

B EMPLOYEE\_ID cannot be null

D mismatch of field\_name with datatype

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-11

### QUESTION 192

The user Sue issues this SQL statement:

GRANT SELECT ON sue.EMP TO alice WITH GRANT OPTION;

The user Alice issues this SQL statement:

GRANT SELECT ON sue.EMP TO reena WITH GRANT OPTION;

The user Reena issues this SQL statement:

GRANT SELECT ON sue.EMP TO timber;

The user Sue issues this SQL statement:

REVOKE select on sue.EMP FROM alice;

For which users does the revoke command revoke SELECT privileges on the SUE.EMP table?

- A. Alice only

- B. Alice and Reena
- C. Alice, Reena, and Timber
- D. Sue, Alice, Reena, and Timber

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: use the REVOKE statement to revoke privileges granted to other users. Privilege granted to others through the WITH GRANT OPTION clause are also revoked.

Alice, Reena and Timber will be revoke.

Incorrect answer:

A the correct answer should be Alice, Reena and Timber

B the correct answer should be Alice, Reena and Timber

D the correct answer should be Alice, Reena and Timber

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-17

### QUESTION 193

The EMPLOYEES table contains these columns:

EMPLOYEE\_ID NUMBER(4)

LAST\_NAME VARCHAR2 (25)

JOB\_ID VARCHAR2(10)

You want to search for strings that contain 'SA\_' in the JOB\_ID column. Which SQL statement do you use?

- A. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\\_ ' ESCAPE '\';
- B. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\\_';
- C. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\\_ ' ESCAPE '"';
- D. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id = '%SA\\_';

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: ESCAPE identifier to search for the \_ symbol

Incorrect answer:

B ESCAPE identifier must be use

C wrong syntax

D wrong syntax

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-13

### QUESTION 194

The CUSTOMERS table has these columns:

CUSTOMER\_ID NUMBER(4) NOT NULL

CUSTOMER\_NAME VARCHAR2(100) NOT NULL

CUSTOMER\_ADDRESS VARCHAR2(150)

CUSTOMER\_PHONE VARCHAR2(20)

You need to produce output that states "Dear Customer customer\_name, ". The customer\_name data values come from the CUSTOMER\_NAME column in the CUSTOMERS table.

Which statement produces this output?

- A. SELECT dear customer, customer\_name, FROM customers;
- B. SELECT "Dear Customer", customer\_name || ',' FROM customers;
- C. SELECT 'Dear Customer ' || customer\_name ',' FROM customers;
- D. SELECT 'Dear Customer ' || customer\_name || ',' FROM customers;
- E. SELECT "Dear Customer " || customer\_name || "," FROM customers;
- F. SELECT 'Dear Customer ' || customer\_name || ',' || FROM customers;

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Concatenation operator to create a resultant column that is a character expression.

Incorrect answer:

A no such dear customer column

B invalid syntax

C invalid syntax

E invalid syntax

F invalid syntax

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 1-18

#### **QUESTION 195**

What is true about sequences?

- A. Once created, a sequence belongs to a specific schema.
- B. Once created, a sequence is linked to a specific table.
- C. Once created, a sequence is automatically available to all users.
- D. Only the DBA can control which sequence is used by a certain table.
- E. Once created, a sequence is automatically used in all INSERT and UPDATE statements.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 196**

Which object privileges can be granted on a view?

- A. none
- B. DELETE, INSERT, SELECT
- C. ALTER, DELETE, INSERT, SELECT
- D. DELETE, INSERT, SELECT, UPDATE

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE.

Incorrect answer:

A Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE  
B Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE  
C Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-12

#### QUESTION 197

Examine the SQL statement that creates ORDERS table:

```
CREATE TABLE orders (SER_NO NUMBER UNIQUE, ORDER_ID NUMBER, ORDER_DATE DATE NOT NULL, STATUS VARCHAR2(10) CHECK (status IN ('CREDIT', 'CASH')), PROD_ID NUMBER REFERENCES PRODUCTS(PRODUCT_ID), ORD_TOTAL NUMBER, PRIMARY KEY (order_id, order_date));
```

For which columns would an index be automatically created when you execute the above SQL statement?  
(Choose two.)

- A. SER\_NO
- B. ORDER\_ID
- C. STATUS
- D. PROD\_ID
- E. ORD\_TOTAL
- F. composite index on ORDER\_ID and ORDER\_DATE

**Correct Answer:** AF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: Index exist for UNIQUE and PRIMARY KEY constraints

Incorrect answer:

B ORDER\_ID is neither UNIQUE nor PRIMARY KEY

C STATUS is neither UNIQUE nor PRIMARY KEY

D PROD\_ID is neither UNIQUE nor PRIMARY KEY

E ORD\_TOTAL is neither UNIQUE nor PRIMARY KEY

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-15

#### QUESTION 198

What is true of using group functions on columns that contain NULL values?

- A. Group functions on columns ignore NULL values.
- B. Group functions on columns returning dates include NULL values.
- C. Group functions on columns returning numbers include NULL values.
- D. Group functions on columns cannot be accurately used on columns that contain NULL values.
- E. Group functions on columns include NULL values in calculations if you use the keyword INC\_NULLS.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation: group functions on column ignore NULL values

Incorrect answer:

B group functions on column ignore NULL values

C group functions on column ignore NULL values

D NVL function can be use for column with NULL values  
E no such INC\_NULLS keyword

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-12

#### **QUESTION 199**

The STUDENT\_GRADES table has these columns:  
STUDENT\_ID NUMBER(12)  
SEMESTER\_END DATE  
GPA NUMBER(4,3)

Which statement finds the highest grade point average (GPA) per semester?

- A. SELECT MAX(gpa) FROM student\_grades WHERE gpa IS NOT NULL;
- B. SELECT (gpa) FROM student\_grades GROUP BY semester\_end WHERE gpa IS NOT NULL;
- C. SELECT MAX(gpa) FROM student\_grades WHERE gpa IS NOT NULL GROUP BY semester\_end;
- D. SELECT MAX(gpa) GROUP BY semester\_end WHERE gpa IS NOT NULL FROM student\_grades;
- E. SELECT MAX(gpa) FROM student\_grades GROUP BY semester\_end WHERE gpa IS NOT NULL;

**Correct Answer:** C

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

For highest gpa value MAX function is needed, for result with per semester GROUP BY clause is needed

Incorrect answer:

- A per semester condition is not included
- B result would not display the highest gpa value
- D invalid syntax error
- E invalid syntax error

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-7

#### **QUESTION 200**

In which four clauses can a sub query be used? (Choose four.)

- A. in the INTO clause of an INSERT statement
- B. in the FROM clause of a SELECT statement
- C. in the GROUP BY clause of a SELECT statement
- D. in the WHERE clause of a SELECT statement
- E. in the SET clause of an UPDATE statement
- F. in the VALUES clause of an INSERT statement

**Correct Answer:** ABDE

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

Explanation:

- A: a sub query is valid on the INTO clause of an INSERT Statement
- B: a sub query can be used in the FROM clause of a SELECT statement
- D: a sub query can be used in the WHERE clause of a SELECT statement,
- E: a sub query can be used in the SET clauses of an UPDATE statement,

Incorrect answer:

- C sub query cannot be used



F: is incorrect.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 6-5

### QUESTION 201

Examine this statement:

```
SELECT student_id, gpa FROM student_grades WHERE gpa > &&value;
```

You run the statement once, and when prompted you enter a value of 2.0. A report is produced. What happens when you run the statement a second time?

- A. An error is returned.
- B. You are prompted to enter a new value.
- C. A report is produced that matches the first report produced.
- D. You are asked whether you want a new value or if you want to run the report based on the previous value.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

use the double-ampersand if you want to reuse the variable value without prompting the user each time.

Incorrect answer:

A is not an error

B && will not prompt user for second time

D && will not ask the user for new value

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7-13

### QUESTION 202

Which SQL statement returns a numeric value?

- A. 

```
SELECT ADD_MONTHS(MAX(hire_Date), 6)
FROM EMP;
```
- B. 

```
SELECT ROUND(hire_date)
FROM EMP;
```
- C. 

```
SELECT sysdate-hire_date
FROM EMP;
```
- D. 

```
SELECT TO_NUMBER(hire_date + 7)
FROM EMP;
```

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

DATE value subtract DATE value will return numeric value.

Incorrect answer:

A does not return numeric value

B does not return numeric value

D does not return numeric value

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 9-13

**QUESTION 203**

What are two reasons to create synonyms? (Choose two.)

- A. You have too many tables.
- B. Your tables names are too long.
- C. Your tables have difficult names.
- D. You want to work on your own tables.
- E. You want to use another schema's tables.
- F. You have too many columns in your tables.

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Create a synonyms when the names of the tables are too long or the table names are difficult.

Incorrect answers:

A The number of tables in a schema is not a consideration when creating a synonym.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 12-24

**QUESTION 204**

What is true about updates through a view?

- A. You cannot update a view with group functions.
- B. When you update a view group functions are automatically computed.
- C. When you update a view only the constraints on the underlying table will be in effect.
- D. When you update a view the constraints on the views always override the constraints on the underlying tables.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:****QUESTION 205**

You need to write a SQL statement that returns employee name, salary, department ID, and maximum salary earned in the department of the employee for all employees who earn less than the maximum salary in their department.

Which statement accomplishes this task?

- A. `SELECT a.emp_name, a.sal, b.dept_id, MAX(sal) FROM employees a, departments b WHERE a.dept_id = b.dept_id AND a.sal < MAX(sal) GROUP BY b.dept_id;`
- B. `SELECT a.emp_name, a.sal, a.dept_id, b.maxsal FROM employees a, (SELECT dept_id, MAX(sal) maxsal FROM employees GROUP BY dept_id) b WHERE a.dept_id = b.dept_id AND a.sal < b.maxsal;`
- C. `SELECT a.emp_name, a.sal, a.dept_id, b.maxsal FROM employees a WHERE a.sal < (SELECT MAX(sal) maxsal FROM employees b GROUP BY dept_id);`
- D. `SELECT emp_name, sal, dept_id, maxsal FROM employees, (SELECT dept_id, MAX(sal) maxsal FROM employees GROUP BY dept_id) WHERE a.sal < maxsal;`

**Correct Answer:** B

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation: function MAX(column\_name)

Incorrect answer:

A invalid statement

C inner query return more than one line

D column maxsal does not exists.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-7

**QUESTION 206**

The CUSTOMERS table has these columns:

|                  |                        |
|------------------|------------------------|
| CUSTOMER_ID      | NUMBER(4) NOT NULL     |
| CUSTOMER_NAME    | VARCHAR2(100) NOT NULL |
| STREET_ADDRESS   | VARCHAR2(150)          |
| CITY_ADDRESS     | VARCHAR2(50)           |
| STATE_ADDRESS    | VARCHAR2(50)           |
| PROVINCE_ADDRESS | VARCHAR2(50)           |
| COUNTRY_ADDRESS  | VARCHAR2(50)           |
| POSTAL_CODE      | VARCHAR2(12)           |
| CUSTOMER_PHONE   | VARCHAR2(20)           |

A promotional sale is being advertised to the customers in France. Which WHERE clause identifies customers that are located in France?

- A. WHERE lower(country\_address) = "france"
- B. WHERE lower(country\_address) = 'france'
- C. WHERE lower(country\_address) IS 'france'
- D. WHERE lower(country\_address) = '%france%'
- E. WHERE lower(country\_address) LIKE %france%

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

WHERE lower(country\_address)='france'

Incorrect answer:

A invalid use of symbol ""

C invalid use of IS keyword

D invalid use of % in condition

E invalid use of condition

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-12

**QUESTION 207**

Which are iSQL\*Plus commands? (Choose all that apply.)

- A. INSERT
- B. UPDATE
- C. SELECT
- D. DESCRIBE
- E. DELETE
- F. RENAME

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The only SQL\*Plus command in this list : DESCRIBE. It cannot be used as SQL command. This command returns a description of tablename, including all columns in that table, the datatype for each column and an indication of whether the column permits storage of NULL values.

Incorrect answer:

- A INSERT is not a SQL\*PLUS command
- B UPDATE is not a SQL\*PLUS command
- C SELECT is not a SQL\*PLUS command
- E DELETE is not a SQL\*PLUS command
- F RENAME is not a SQL\*PLUS command

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7

**QUESTION 208**

Examine the description of the CUSTOMERS table:

|                  |               |          |
|------------------|---------------|----------|
| CUSTOMER_ID      | NUMBER(4)     | NOT NULL |
| CUSTOMER_NAME    | VARCHAR2(100) | NOT NULL |
| STREET_ADDRESS   | VARCHAR2(150) |          |
| CITY_ADDRESS     | VARCHAR2(50)  |          |
| STATE_ADDRESS    | VARCHAR2(50)  |          |
| PROVINCE_ADDRESS | VARCHAR2(50)  |          |
| COUNTRY_ADDRESS  | VARCHAR2(50)  |          |
| POSTAL_CODE      | VARCHAR2(12)  |          |
| CUSTOMER_PHONE   | VARCHAR2(20)  |          |

The CUSTOMER\_ID column is the primary key for the table.

Which statement returns the city address and the number of customers in the cities Los Angeles or San Francisco?

- A. SELECT city\_address, COUNT(\*)  
FROM customers  
WHERE city\_address IN ( 'Los Angeles', 'San Fransisco');
- B. SELECT city\_address, COUNT (\*)  
FROM customers  
WHERE city address IN ( 'Los Angeles', 'San Fransisco')  
GROUP BY city\_address;
- C. SELECT city\_address, COUNT(customer\_id)

```
FROM customers
WHERE city_address IN ( 'Los Angeles', 'San Fransisco')
GROUP BY city_address, customer_id;
D. SELECT city_address, COUNT (customer_id)
FROM customers
GROUP BY city_address IN ( 'Los Angeles', 'San Fransisco');
```

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Not C: The customer ID in the GROUP BY clause is wrong

#### **QUESTION 209**

What does the FORCE option for creating a view do?

- A. creates a view with constraints
- B. creates a view even if the underlying parent table has constraints
- C. creates a view in another schema even if you don't have privileges
- D. creates a view regardless of whether or not the base tables exist

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

create a view regardless of whether or not the base tables exist.

Incorrect answer:

A the option is not valid

B the option is not valid

C the option is not valid

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-3

#### **QUESTION 210**

A data manipulation language statement \_\_\_\_\_.

- A. completes a transaction on a table
- B. modifies the structure and data in a table
- C. modifies the data but not the structure of a table
- D. modifies the structure but not the data of a table

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

modifies the data but not the structure of a table

Incorrect answer:

A DML does not complete a transaction

B DDL modifies the structure and data in the table

D DML does not modified table structure.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-3

**QUESTION 211**

Evaluate the SQL statement:

```
SELECT LPAD (salary,10,'*')  
FROM EMP  
WHERE EMP_ID = 1001;
```

If the employee with the EMP\_ID 1001 has a salary of 17000, what is displayed?

- A. 17000.00
- B. 17000\*\*\*\*\*
- C. \*\*\*\*170.00
- D. \*\*17000.00
- E. an error statement

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 212**

The DBA issues this SQL command:

```
CREATE USER scott IDENTIFIED by tiger;
```

What privileges does the user Scott have at this point?

- A. no privileges
- B. only the SELECT privilege
- C. only the CONNECT privilege
- D. all the privileges of a default user

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

when a user is created, by default no privilege is granted

Incorrect answer:

B SELECT is not grant

C CONNECT is not grant

D default profile is grant by default not privilege.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-6

**QUESTION 213**

Examine the statement:

```
GRANT select, insert, update  
ON student_grades
```

TO manager  
WITH GRANT OPTION;

Which two are true? (Choose two.)

- A. MANAGER must be a role.
- B. It allows the MANAGER to pass the specified privileges on to other users.
- C. It allows the MANAGER to create tables that refer to the STUDENT\_GRADES table.
- D. It allows the MANAGER to apply all DML statements on the STUDENT\_GRADES table.
- E. It allows the MANAGER the ability to select from, insert into, and update the STUDENT\_GRADES table.
- F. It allows the MANAGER the ability to select from, delete from, and update the STUDENT\_GRADES table.

**Correct Answer:** BE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

GRANT ROLE to ROLE/USER

Incorrect answer:

- A Role can be grant to user
- C Create table privilege is not granted
- D Execute privilege is not granted
- F Delete privilege is not granted

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-15

#### **QUESTION 214**

Which best describes an inline view?

- A. a schema object
- B. a sub query that can contain an ORDER BY clause
- C. another name for a view that contains group functions
- D. a sub query that is part of the FROM clause of another query

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

a sub query that is part of the FROM clause of another query

Incorrect answer:

- A is not a schema object
- B sub query can contain GROUP BY clause as well.
- C does not necessary contains group functions

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-21

#### **QUESTION 215**

Examine the structure of the EMPLOYEES and DEPARTMENTS tables:

|               |              |
|---------------|--------------|
| EMPLOYEES     |              |
| EMPLOYEE_ID   | NUMBER       |
| DEPARTMENT_ID | NUMBER       |
| MANAGER_ID    | NUMBER       |
| LAST_NAME     | VARCHAR2(25) |

|                 |              |
|-----------------|--------------|
| DEPARTMENTS     |              |
| DEPARTMENT_ID   | NUMBER       |
| MANAGER_ID      | NUMBER       |
| DEPARTMENT_NAME | VARCHAR2(35) |
| LOCATION_ID     | NUMBER       |

You want to create a report displaying employee last names, department names, and locations. Which query should you use to create an equi-join?

- A. SELECT last\_name, department\_name, location\_id  
FROM employees , departments ;
- B. SELECT employees.last\_name, departments.department\_name, departments.location\_id FROM  
employees e, departments D  
WHERE e.department\_id =d.department\_id;
- C. SELECT e.last\_name, d.department\_name, d.location\_id  
FROM employees e, departments D  
WHERE manager\_id =manager\_id;
- D. SELECT e.last\_name, d.department\_name, d.location\_id  
FROM employees e, departments D  
WHERE e.department\_id =d.department\_id;

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Equijoins are also called simple joins or inner joins. Equijoin involve primary key and foreign key.

Incorrect answer:

A there is no join

B invalid syntax

C does not involve the join in the primary and foreign key

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 4-8

#### **QUESTION 216**

The PRODUCTS table has these columns:

|              |              |
|--------------|--------------|
| PRODUCT_ID   | NUMBER(4)    |
| PRODUCT_NAME | VARCHAR2(45) |
| PRICE        | NUMBER(8,2)  |

Evaluate this SQL statement:



```
SELECT *  
FROM PRODUCTS  
ORDER BY price, product_name;
```

What is true about the SQL statement?

- A. The results are not sorted.
- B. The results are sorted numerically.
- C. The results are sorted alphabetically.
- D. The results are sorted numerically and then alphabetically.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

the result is sort by price which is numeric and follow by product\_name which is alphabetically.

Incorrect answer:

A the results are sorted

B the results are sorted with alphabetically as well

C the results are sorted with numerically as well

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-2

#### **QUESTION 217**

In which two cases would you use an outer join? (Choose two.)

- A. The tables being joined have NOT NULL columns.
- B. The tables being joined have only matched data.
- C. The columns being joined have NULL values.
- D. The tables being joined have only unmatched data.
- E. The tables being joined have both matched and unmatched data.
- F. Only when the tables have a primary key/foreign key relationship.

**Correct Answer:** CE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

You use an outer join to also see rows that do not meet the join condition.

Incorrect answer:

A meet a join condition

B meet a join condition

D meet non join condition only

F does not take into consideration of primary key and foreign key relationship

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 4-17

#### **QUESTION 218**

Which constraint can be defined only at the column level?

- A. UNIQUE

- B. NOT NULL
- C. CHECK
- D. PRIMARY KEY
- E. FOREIGN KEY

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

the NOT NULL constraint can be specified only at the column level, not at the table level.

Incorrect answer:

A UNIQUE can be define at table level

C CHECK can be define at table level

D PRIMARY KEY can be define at table level

E FOREIGN KEY can be define at table level

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-8

New Questions

**QUESTION 219**

Which SQL statements would display the value 1890.55 as \$1,890.55? (Choose three.)

- A. SELECT TO\_CHAR(1890.55,'\$0G000D00')  
FROM DUAL;
- B. SELECT TO\_CHAR(1890.55,'\$9,999V99')  
FROM DUAL;
- C. SELECT TO\_CHAR(1890.55,'\$99,999D99')  
FROM DUAL;
- D. SELECT TO\_CHAR(1890.55,'\$99G999D00')  
FROM DUAL;
- E. SELECT TO\_CHAR(1890.55,'\$99G999D99')  
FROM DUAL;

**Correct Answer:** ADE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

**TABLE 5-3**

Numeric  
Format Masks

| Format Element | Description of Element                                                 | Format       | Number  | Character Result                             |
|----------------|------------------------------------------------------------------------|--------------|---------|----------------------------------------------|
| 9              | Numeric width                                                          | 9999         | 12      | 12                                           |
| 0              | Displays leading zeros                                                 | 09999        | 0012    | 00012                                        |
| .              | Position of decimal point                                              | 09999.999    | 030.40  | 00030.400                                    |
| D              | Decimal separator position (period is default)                         | 09999D999    | 030.40  | 00030.400                                    |
| ,              | Position of comma symbol                                               | 09999,999    | 03040   | 00003,040                                    |
| G              | Group separator position (comma is default)                            | 09999G999    | 03040   | 00003,040                                    |
| \$             | Dollar sign                                                            | \$099999     | 03040   | \$003040                                     |
| L              | Local currency                                                         | L099999      | 03040   | GBP003040 if nls_currency is set to GBP      |
| MI             | Position of minus sign for negatives                                   | 99999MI      | -3040   | 3040-                                        |
| PR             | Wrap negatives in parentheses                                          | 99999PR      | -3040   | <3040>                                       |
| EEEE           | Scientific notation                                                    | 99.99999EEEE | 121.976 | 1.21976E+02                                  |
| U              | nls_dual_currency                                                      | U099999      | 03040   | CAD003040 if nls_dual_currency is set to CAD |
| V              | Multiplies by 10 <sup>n</sup> times (n is the number of nines after V) | 9999V99      | 3040    | 304000                                       |
| S              | + or - sign is prefixed                                                | S999999      | 3040    | +3040                                        |

**QUESTION 220**

Examine the structure of the SHIPMENTS table:

```

name Null Type
PO_ID NOT NULL NUMBER(3)
PO_DATE NOT NULL DATE
SHIPMENT_DATE NOT NULL DATE
SHIPMENT_MODE VARCHAR2(30)
SHIPMENT_COST NUMBER(8,2)

```

You want to generate a report that displays the PO\_ID and the penalty amount to be paid if the SHIPMENT\_DATE is later than one month from the PO\_DATE. The penalty is \$20 per day. Evaluate the following two queries:

```
SQL> SELECT po_id, CASE
WHEN MONTHS_BETWEEN (shipment_date,po_date)>1 THEN
TO_CHAR((shipment_date - po_date) * 20) ELSE 'No Penalty' END PENALTY
FROM shipments;
```

```
SQL>SELECT po_id, DECODE
(MONTHS_BETWEEN (po_date,shipment_date)>1,
TO_CHAR((shipment_date - po_date) * 20), 'No Penalty') PENALTY
FROM shipments;
```

Which statement is true regarding the above commands?

- A. Both execute successfully and give correct results.
- B. Only the first query executes successfully but gives a wrong result.
- C. Only the first query executes successfully and gives the correct result.
- D. Only the second query executes successfully but gives a wrong result.
- E. Only the second query executes successfully and gives the correct result.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

The MONTHS\_BETWEEN(date 1, date 2) function returns the number of months between two dates:

months\_between('01-FEB-2008','01-JAN-2008') = 1

The DECODE Function

Although its name sounds mysterious, this function is straightforward. The DECODE function implements if then-else conditional logic by testing its first two terms for equality and returns the third if they are equal and optionally returns another term if they are not. DECODE Function Facilitates conditional inquiries by doing the work of a CASE expression or an IF-THENELSE statement:

DECODE(col|expression, search1, result1

[, search2, result2,...,]

[, default])

DECODE Function

The DECODE function decodes an expression in a way similar to the IF-THEN-ELSE logic that is used in various languages. The DECODE function decodes expression after comparing it to each search value. If the expression is the same as search, result is returned. If the default value is omitted, a null value is returned where a search value does not match any of the result values.

**QUESTION 221**

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_MARITAL_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.)

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

**Correct Answer:** DE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Describe the Types of Problems That the Subqueries Can Solve There are many situations where you will need the result of one query as the input for another.

Use of a Subquery Result Set for Comparison Purposes

Which employees have a salary that is less than the average salary? This could be answered by two statements, or by a single statement with a subquery. The following example uses two statements:

select avg(salary) from employees;

select last\_name from employees where salary < result\_of\_previous\_query ; Alternatively, this example uses one statement with a subquery:

select last\_name from employees where salary < (select avg(salary) from employees); In this example, the subquery is used to substitute a value into the WHERE clause of the parent query: it is returning a single value, used for comparison with the rows retrieved by the parent query.

The subquery could return a set of rows. For example, you could use the following to find all departments that do actually have one or more employees assigned to them:

select department\_name from departments where department\_id in (select distinct(department\_id) from employees);

## QUESTION 222

View the Exhibit; examine the structure of the PROMOTIONS table.

| Table PROMOTIONS     |          |              |
|----------------------|----------|--------------|
| Name                 | Null?    | Type         |
| PROMO_ID             | NOT NULL | NUMBER(6)    |
| PROMO_NAME           | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY    | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER       |
| PROMO_CATEGORY       | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID    | NOT NULL | NUMBER       |
| PROMO_COST           | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE     | NOT NULL | DATE         |
| PROMO_END_DATE       | NOT NULL | DATE         |

Each promotion has a duration of at least seven days.

Your manager has asked you to generate a report, which provides the weekly cost for each promotion done to I date.

Which query would achieve the required result?

- A. SELECT promo\_name, promo\_cost/promo\_end\_date-promo\_begin\_date/7 FROM promotions;
- B. SELECT promo\_name,(promo\_cost/promo\_end\_date-promo\_begin\_date)/7 FROM promotions;
- C. SELECT promo\_name, promo\_cost/(promo\_end\_date-promo\_begin\_date/7) FROM promotions;
- D. SELECT promo\_name, promo\_cost/((promo\_end\_date-promo\_begin\_date)/7) FROM promotions;

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 223

You need to produce a report where each customer's credit limit has been incremented by \$1000. In the output, the customer's last name should have the heading Name and the incremented credit limit should be labeled New Credit Limit.

The column headings should have only the first letter of each word in uppercase .

Which statement would accomplish this requirement?

- A. SELECT cust\_last\_name Name, cust\_credit\_limit + 1000  
"New Credit Limit"  
FROM customers;
- B. SELECT cust\_last\_name AS Name, cust\_credit\_limit + 1000  
AS New Credit Limit  
FROM customers;
- C. SELECT cust\_last\_name AS "Name", cust\_credit\_limit + 1000  
AS "New Credit Limit"  
FROM customers;
- D. SELECT INITCAP(cust\_last\_name) "Name", cust\_credit\_limit + 1000 INITCAP("NEW CREDIT LIMIT")  
FROM customers;

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

A column alias:

- Renames a column heading
- Is useful with calculations
- Immediately follows the column name (There can also be the optional AS keyword between the column name and the alias.)
- Requires double quotation marks if it contains spaces or special characters, or if it is case sensitive.

#### QUESTION 224

View the Exhibit and examine the data in the EMPLOYEES table.

| EMPLOYEES |           |      |      |
|-----------|-----------|------|------|
| ENAME     | HIREDATE  | SAL  | COMM |
| SMITH     | 17-DEC-00 | 800  |      |
| ALLEN     | 20-FEB-99 | 1600 | 300  |
| WARD      | 22-FEB-95 | 1250 | 500  |
| JONES     | 02-APR-98 | 2975 |      |
| MARTIN    | 28-SEP-99 | 1250 | 1400 |
| BLAKE     | 01-MAY-97 | 2850 |      |

You want to generate a report showing the total compensation paid to each employee to date.  
You issue the following query:

```
SQL>SELECT ename 'joined on ' hiredate
', the total compensation paid is '
TO_CHAR(ROUND(ROUND(SYSDATE-hiredate)/365) * sal + comm)
"COMPENSATION UNTIL DATE"
FROM employees;
```

What is the outcome?

- It generates an error because the alias is not valid.
- It executes successfully and gives the correct output.
- It executes successfully but does not give the correct output.
- It generates an error because the usage of the ROUND function in the expression is not valid.
- It generates an error because the concatenation operator can be used to combine only two items.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

ROUND(column|expression, n) Rounds the column, expression, or value to n decimal places or, if n is omitted, no decimal places (If n is negative, numbers to the left of decimal point are rounded.)

#### QUESTION 225

Examine the structure of the PROMOTIONS table:

name Null Type  
PROMO\_ID NOT NULL NUMBER(6)  
PROMO\_NAME NOT NULL VARCHAR2(30)  
PROMO\_CATEGORY NOT NULL VARCHAR2(30)  
PROMO\_COST NOT NULL NUMBER(10,2)

The management wants to see a report of unique promotion costs in each promotion category. Which query would achieve the required result?

- A. SELECT DISTINCT promo\_cost, promo\_category FROM promotions;
- B. SELECT promo\_category, DISTINCT promo\_cost FROM promotions;
- C. SELECT DISTINCT promo\_cost, DISTINCT promo\_category FROM promotions;
- D. SELECT DISTINCT promo\_category, promo\_cost FROM promotions ORDER BY 1;

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 226

Evaluate the following query:

```
SELECT INTERVAL '300' MONTH,  
INTERVAL '54-2' YEAR TO MONTH,  
INTERVAL '11:12:10.1234567' HOUR TO SECOND  
FROM dual;
```

What is the correct output of the above query?

- A. +25-00 , +54-02, +00 11:12:10.123457
- B. +00-300, +54-02, +00 11:12:10.123457
- C. +25-00 , +00-650, +00 11:12:10.123457
- D. +00-300 , +00-650, +00 11:12:10.123457

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Datetime Data Types

You can use several datetime data types:

INTERVAL YEAR TO MONTH

Stored as an interval of years and months

INTERVAL DAY TO SECOND

Stored as an interval of days, hours, minutes, and seconds

#### QUESTION 227

Examine the structure proposed for the TRANSACTIONS table:



| Name              | Null     | Type         |
|-------------------|----------|--------------|
| TRANS_ID          | NOT NULL | NUMBER(6)    |
| CUST_NAME         | NOT NULL | VARCHAR2(20) |
| CUST_STATUS       | NOT NULL | CHAR         |
| TRANS_DATE        | NOT NULL | DATE         |
| TRANS_VALIDITY    |          | VARCHAR2     |
| CUST_CREDIT_LIMIT |          | NUMBER       |

Which statements are true regarding the creation and storage of data in the above table structure? (Choose all that apply.)

- A. The CUST\_STATUS column would give an error.
- B. The TRANS\_VALIDITY column would give an error.
- C. The CUST\_STATUS column would store exactly one character.
- D. The CUST\_CREDIT\_LIMIT column would not be able to store decimal values.
- E. The TRANS\_VALIDITY column would have a maximum size of one character.
- F. The TRANS\_DATE column would be able to store day, month, century, year, hour, minutes, seconds, and fractions of seconds.

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

VARCHAR2(size) Variable-length character data (A maximum size must be specified: minimum size is 1; maximum size is 4,000.)

CHAR [(size)] Fixed-length character data of length size bytes (Default and minimum size is 1; maximum size is 2,000.)

NUMBER [(p,s)] Number having precision p and scale s (Precision is the total number of decimal digits and scale is the number of digits to the right of the decimal point; precision can range from 1 to 38, and scale can range from 84 to 127.)

DATE Date and time values to the nearest second between January 1, 4712 B.C., and December 31, 9999 A.D.

**QUESTION 228**

Examine the structure proposed for the TRANSACTIONS table:

| Name                    | Null     | Type          |
|-------------------------|----------|---------------|
| TRANS_ID                | NOT NULL | NUMBER(6)     |
| CUST_NAME               | NOT NULL | VARCHAR2(20)  |
| CUST_STATUS             | NOT NULL | VARCHAR2      |
| TRANS_DATE              | NOT NULL | DATE          |
| TRANS_VALIDITY INTERVAL |          | DAY TO SECOND |
| CUST_CREDIT_VALUE       |          | NUMBER(10)    |

Which two statements are true regarding the storage of data in the above table structure? (Choose two.)

- A. The TRANS\_DATE column would allow storage of dates only in the dd-mon-yyyy format.
- B. The CUST\_CREDIT\_VALUE column would allow storage of positive and negative integers.
- C. The TRANS\_VALIDITY column would allow storage of a time interval in days, hours, minutes, and seconds.
- D. The CUST\_STATUS column would allow storage of data up to the maximum VARCHAR2 size of 4,000 characters.

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

B: The NUMBER datatype stores fixed and floating-point numbers. Numbers of virtually any magnitude can be stored and are guaranteed portable among different systems operating Oracle, up to 38 digits of precision.

The following numbers can be stored in a NUMBER column:

Positive numbers in the range  $1 \times 10^{-130}$  to  $9.99...9 \times 10^{125}$  with up to 38 significant digits  
Negative numbers from  $-1 \times 10^{-130}$  to  $9.99...99 \times 10^{125}$  with up to 38 significant digits  
Zero

Positive and negative infinity (generated only by importing from an Oracle Version 5 database)

D: The VARCHAR2 datatype stores variable-length character strings. When you create a table with a VARCHAR2 column, you specify a maximum string length (in bytes or characters) between 1 and 4000 bytes for the VARCHAR2 column.

An interval literal specifies a period of time, and Oracle supports two types of interval literals:

YEAR\_TO\_MONTH and DAY TO SECOND. For DAY TO SECOND, you can specify these differences in terms in terms of days, hours, minutes, and seconds. DAY TO SECOND contains a leading field and may contain an optional trailing field. If trailing field is specified it must be less significant than the leading field. For example, INTERVAL MINUTE TO DAY is not valid.

A DAY TO MINUTE interval considers an interval of days to the nearest minute.

Reference: Oracle Database Concepts 10g, Native Datatypes

#### QUESTION 229

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

- A. A table name can be of any length.
- B. A table can have any number of columns.
- C. A column that has a DEFAULT value cannot store null values.
- D. A table and a view can have the same name in the same schema.
- E. A table and a synonym can have the same name in the same schema.
- F. The same table name can be used in different schemas in the same database.

**Correct Answer:** EF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Synonyms

Synonyms are database objects that enable you to call a table by another name. You can create synonyms to give an alternative name to a table.

#### QUESTION 230

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

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

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Including Constraints

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.
- The following constraint types are valid:

NOT NULL

UNIQUE

PRIMARY KEY

FOREIGN KEY

CHECK

**QUESTION 231**

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

- A. A foreign key cannot contain NULL values.
- B. The column with a UNIQUE constraint can store NULLS .
- C. A constraint is enforced only for an INSERT operation on a table.
- D. You can have more than one column in a table as part of a primary key.

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 232**

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

- A. A table can have only one primary key and one foreign key.
- B. A table can have only one primary key but multiple foreign keys.
- C. Only the primary key can be defined at the column and table levels.
- D. The foreign key and parent table primary key must have the same name.
- E. Both primary key and foreign key constraints can be defined at both column and table levels.

**Correct Answer:** BE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 233**

Examine the following SQL commands:

```
SQL>CREATE TABLE products (
prod_id NUMBER(3) CONSTRAINT p_ck CHECK (prod_id > 0),
prod_name CHAR(30),
prod_qty NUMBER(6),
CONSTRAINT p_name NOT NULL,
CONSTRAINT prod_pk PRIMARY KEY (prod_id));
```

```
SQL>CREATE TABLE warehouse (
warehouse_id NUMBER(4),
roomno NUMBER(10) CONSTRAINT r_id CHECK(roomno BETWEEN 101 AND 200),
location VARCHAR2(25),
prod_id NUMBER(3),
CONSTRAINT wr_pr_pk PRIMARY KEY (warehouse_id,prod_id),
CONSTRAINT prod_fk FOREIGN KEY (prod_id) REFERENCES products(prod_id));
```

Which statement is true regarding the execution of the above SQL commands?

- A. Both commands execute successfully.
- B. The first CREATE TABLE command generates an error because the NULL constraint is not valid.
- C. The second CREATE TABLE command generates an error because the CHECK constraint is not valid.
- D. The first CREATE TABLE command generates an error because CHECK and PRIMARY KEY constraints cannot be used for the same column.
- E. The first CREATE TABLE command generates an error because the column PROD\_ID cannot be used in the PRIMARY KEY and FOREIGN KEY constraints.

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Defining Constraints

The slide gives the syntax for defining constraints when creating a table. You can create constraints at either the column level or table level. Constraints defined at the column level are included when the column is defined. Table-level constraints are defined at the end of the table definition and must refer to the column or columns on which the constraint pertains in a set of parentheses. It is mainly the syntax that differentiates the two; otherwise, functionally, a columnlevel constraint is the same as a table-level constraint.

NOT NULL constraints must be defined at the column level.

Constraints that apply to more than one column must be defined at the table level.

#### QUESTION 234

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

- A. A simple view in which column aliases have been used cannot be updated.
- B. Rows cannot be deleted through a view if the view definition contains the DISTINCT keyword.
- C. Rows added through a view are deleted from the table automatically when the view is dropped.
- D. The OR REPLACE option is used to change the definition of an existing view without dropping and recreating it.
- E. The WITH CHECK OPTION constraint can be used in a view definition to restrict the columns displayed through the view.

**Correct Answer: BD**

**Section: (none)**

## Explanation

## Explanation/Reference:

### QUESTION 235

Which three statements are true regarding views? (Choose three.)

- A. Views can be created only from tables.
- B. Views can be created from tables or other views.
- C. Only simple views can use indexes existing on the underlying tables.
- D. Both simple and complex views can use indexes existing on the underlying tables.
- E. Complex views can be created only on multiple tables that exist in the same schema.
- F. Complex views can be created on multiple tables that exist in the same or different schemas.

**Correct Answer:** BDF

**Section: (none)**

## Explanation

## Explanation/Reference:

Explanation:

Creating a Sequence (continued)

CYCLE | NOCYCLE Specifies whether the sequence continues to generate values after reaching its maximum or minimum value

(NOCYCLE is the default option.)

CACHE n | NOCACHE Specifies how many values the Oracle server preallocates and keeps in memory (By default, the Oracle server caches 20 values.)

### QUESTION 236

Evaluate the following CREATE SEQUENCE statement:

```
CREATE SEQUENCE seq1  
START WITH 100  
INCREMENT BY 10  
MAXVALUE 200  
CYCLE  
NOCACHE;
```

The SEQ1 sequence has generated numbers up to the maximum limit of 200. You issue the following SQL statement:

```
SELECT seq1.nextval FROM dual;
```

What is displayed by the SELECT statement?

- A. 1
- B. 10
- C. 100
- D. an error

**Correct Answer:** A

**Section: (none)**

## Explanation

## Explanation/Reference:

Explanation:

But why the answer is not "C" ?

Because you didn't specify the MINVALUE for the sequence. If you check the sequence definition that you

created it will have the default value of 1, which it reverts to when cycling. If you wanted to keep the minimum value you would need to specify it in the sequence creation.

sequence is the name of the sequence generator

INCREMENT BY n Specifies the interval between sequence numbers, where n is an integer (If this clause is omitted, the sequence increments by 1.)

START WITH n Specifies the first sequence number to be generated (If this clause is omitted, the sequence starts with 1.)

MAXVALUE n Specifies the maximum value the sequence can generate NOMAXVALUE Specifies a maximum value of  $10^{27}$  for an ascending sequence and 1 for a descending sequence (This is the default option.)

MINVALUE n Specifies the minimum sequence value

NOMINVALUE Specifies a minimum value of 1 for an ascending sequence and ( $10^{26}$ ) for a descending sequence (This is the default option.)

CYCLE | NOCYCLE Specifies whether the sequence continues to generate values after reaching its maximum or minimum value

(NOCYCLE is the default option.)

CACHE n | NOCACHE Specifies how many values the Oracle server preallocates and keeps in memory (By default, the Oracle server caches 20 values.)

### QUESTION 237

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

- A. They can be created on tables and clusters.
- B. They can be created on tables and simple views.
- C. You can create only one index by using the same columns.
- D. You can create more than one index by using the same columns if you specify distinctly different combinations of the columns.

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 238

SLS is a private synonym for the SH.SALES table.

The user SH issues the following command:

DROP SYNONYM sls;

Which statement is true regarding the above SQL statement?

- A. Only the synonym would be dropped.
- B. The synonym would be dropped and the corresponding table would become invalid.
- C. The synonym would be dropped and the packages referring to the synonym would be dropped.
- D. The synonym would be dropped and any PUBLIC synonym with the same name becomes invalid.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

A synonym is an alias for a table (or a view). Users can execute SQL statements against the synonym, and the database will map them into statements against the object to which the synonym points.

Private synonyms are schema objects. Either they must be in your own schema, or they must be qualified with the schema name. Public synonyms exist independently of a schema. A public synonym can be referred to by any user to whom permission has been granted to see it without the need to qualify it with a schema name. Private synonyms must be a unique name within their schema. Public synonyms can have the same name as

schema objects. When executing statements that address objects without a schema qualifier, Oracle will first look for the object in the local schema, and only if it cannot be found will it look for a public synonym.

#### QUESTION 239

View the Exhibit and examine the data in the PROMOTIONS table. PROMO\_BEGIN\_DATE is stored in the default date format, dd-mon-rr. You need to produce a report that provides the name, cost, and start date of all promos in the POST category that were launched before January 1, 2000.

Which SQL statement would you use?

- A. 

```
SELECT promo_name, promo_cost, promo_begin_date
FROM promotions
WHERE promo_category = 'post' AND promo_begin_date < '01-01-00';
```
- B. 

```
SELECT promo_name, promo_cost, promo_begin_date
FROM promotions
WHERE promo_cost LIKE 'post%' AND promo_begin_date < '01-01-2000';
```
- C. 

```
SELECT promo_name, promo_cost, promo_begin_date
FROM promotions
WHERE promo_category LIKE 'P%' AND promo_begin_date < '1-JANUARY-00';
```
- D. 

```
SELECT promo_name, promo_cost, promo_begin_date
FROM promotions
WHERE promo_category LIKE '%post%' AND promo_begin_date < '1-JAN-00';
```

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 240

View the Exhibit and examine the structure of CUSTOMERS table. Evaluate the following query:

```
SQL>SELECT cust_id, cust_city
FROM customers
WHERE cust_first_name NOT LIKE 'A_%g_%' AND
cust_credit_limit BETWEEN 5000 AND 15000 AND
cust_credit_limit NOT IN (7000, 11000) AND
cust_city NOT BETWEEN 'A' AND 'B';
```

Which statement is true regarding the above query?

| 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_MARITAL_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) |

- A. It executes successfully.
- B. It produces an error because the condition on the CUST\_CITY column is not valid.
- C. It produces an error because the condition on the CUST\_FIRST\_NAME column is not valid.
- D. It produces an error because conditions on the CUST\_CREDIT\_LIMIT column are not valid.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 241

View the Exhibit and examine the structure of the CUSTOMERS table. You want to generate a report showing the last names and credit limits of all customers whose last names start with A, B, or C, and credit limit is below 10,000.

Evaluate the following two queries:

```
SQL> SELECT cust_last_name, cust_credit_limit FROM customers
WHERE (UPPER(cust_last_name) LIKE 'A%' OR
UPPER(cust_last_name) LIKE 'B%' OR UPPER(cust_last_name) LIKE 'C%')
AND cust_credit_limit < 10000;
SQL> SELECT cust_last_name, cust_credit_limit FROM customers
WHERE UPPER(cust_last_name) BETWEEN 'A' AND 'C'
AND cust_credit_limit < 10000;
```

Which statement is true regarding the execution of the above queries?

| 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_MARITAL_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) |

- A. Only the first query gives the correct result.
- B. Only the second query gives the correct result.
- C. Both execute successfully and give the same result.
- D. Both execute successfully but do not give the required result.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**



### QUESTION 242

View the Exhibit and examine the structure of the PRODUCTS table. You want to display only those product names with their list prices where the list price is at least double the minimum price. The report should start with the product name having the maximum list price satisfying this condition. Evaluate the following SQL statement:

```
SQL>SELECT prod_name,prod_list_price
FROM products
WHERE prod_list_price >= 2 * prod_min_price
```

Which ORDER BY clauses can be added to the above SQL statement to get the correct output? (Choose all that apply.)

| Table PRODUCTS       |          |                |
|----------------------|----------|----------------|
| Name                 | Null?    | Type           |
| PROD_ID              | NOT NULL | NUMBER(6)      |
| PROD_NAME            | NOT NULL | VARCHAR2(50)   |
| PROD_DESC            | NOT NULL | VARCHAR2(4000) |
| PROD_CATEGORY        | NOT NULL | VARCHAR2(50)   |
| PROD_CATEGORY_ID     | NOT NULL | NUMBER         |
| PROD_UNIT_OF_MEASURE |          | VARCHAR2(20)   |
| SUPPLIER_ID          | NOT NULL | NUMBER(6)      |
| PROD_STATUS          | NOT NULL | VARCHAR2(20)   |
| PROD_LIST_PRICE      | NOT NULL | NUMBER(8,2)    |
| PROD_MIN_PRICE       | NOT NULL | NUMBER(8,2)    |

- A. ORDER BY prod\_list\_price DESC, prod\_name;
- B. ORDER BY (2\*prod\_min\_price)DESC, prod\_name;
- C. ORDER BY prod\_name, (2\*prod\_min\_price)DESC;
- D. ORDER BY prod\_name DESC, prod\_list\_price DESC;
- E. ORDER BY prod\_list\_price DESC, prod\_name DESC;

**Correct Answer:** AE

**Section:** (none)

**Explanation**

#### Explanation/Reference:

Explanation:

Using the ORDER BY Clause

The order of rows that are returned in a query result is undefined. The ORDER BY clause can be used to sort the rows. However, if you use the ORDER BY clause, it must be the last clause of the SQL statement. Further, you can specify an expression, an alias, or a column position as the sort condition.

Syntax

SELECT expr

FROM table

[WHERE condition(s)]

[ORDER BY {column, expr, numeric\_position} [ASC|DESC]];

In the syntax:

ORDER BY specifies the order in which the retrieved rows are displayed ASC orders the rows in ascending order (This is the default order.) DESC orders the rows in descending order

If the ORDER BY clause is not used, the sort order is undefined, and the Oracle server may not fetch rows in the same order for the same query twice. Use the ORDER BY clause to display the rows in a specific order.

Note: Use the keywords NULLS FIRST or NULLS LAST to specify whether returned rows containing null values should appear first or last in the ordering sequence.

### QUESTION 243

View the Exhibit and examine the data in the PROMO\_CATEGORY and PROMO\_COST columns of the PROMOTIONS table.

Evaluate the following two queries:

```
SQL>SELECT DISTINCT promo_category to_char(promo_cost)"code" FROM promotions
ORDER BY code;
SQL>SELECT DISTINCT promo_category promo_cost "code"
FROM promotions
ORDER BY 1;
```

Which statement is true regarding the execution of the above queries?

#### PROMOTIONS

| PROMO_CATEGORY | PROMO_COST |
|----------------|------------|
| radio          | 97200      |
| newspaper      | 97800      |
| TV             | 97600      |
| post           | 98000      |
| internet       | 98200      |
| TV             | 98300      |
| internet       | 98700      |
| newspaper      | 98500      |
| magazine       | 98400      |
| radio          | 99100      |
| post           | 99000      |

- A. Only the first query executes successfully.
- B. Only the second query executes successfully.
- C. Both queries execute successfully but give different results.
- D. Both queries execute successfully and give the same result.

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Note: You cannot use column alias in the WHERE clause.

#### QUESTION 244

Which tasks can be performed using SQL functions that are built into Oracle database? (Choose three.)

- A. finding the remainder of a division
- B. adding a number to a date for a resultant date value
- C. comparing two expressions to check whether they are equal
- D. checking whether a specified character exists in a given string
- E. removing trailing, leading, and embedded characters from a character string

**Correct Answer: ACD**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 245**

Which statements are true regarding single row functions? (Choose all that apply.)

- A. MOD : returns the quotient of a division
- B. TRUNC : can be used with NUMBER and DATE values
- C. CONCAT : can be used to combine any number of values
- D. SYSDATE : returns the database server current date and time
- E. INSTR : can be used to find only the first occurrence of a character in a string
- F. TRIM : can be used to remove all the occurrences of a character from a string

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

ROUND: Rounds value to a specified decimal

TRUNC: Truncates value to a specified decimal

MOD: Returns remainder of division

SYSDATE is a date function that returns the current database server date and time.

**Date-Manipulation Functions**

Date functions operate on Oracle dates. All date functions return a value of the DATE data type except MONTHS\_BETWEEN, which returns a numeric value.

MONTHS\_BETWEEN(date1, date2): Finds the number of months between date1 and date2. The result can be positive or negative. If date1 is later than date2, the result is positive; if date1 is earlier than date2, the result is negative. The noninteger part of the result represents a portion of the month.

ADD\_MONTHS(date, n): Adds n number of calendar months to date. The value of n must be an integer and can be negative.

NEXT\_DAY(date, 'char'): Finds the date of the next specified day of the week ('char') following date. The value of char may be a number representing a day or a character string. LAST\_DAY(date): Finds the date of the last day of the month that contains date. The above list is a subset of the available date functions. ROUND and TRUNC number functions can also be used to manipulate the date values as shown below:

ROUND(date[, 'fmt']): Returns date rounded to the unit that is specified by the format model fmt. If the format model fmt is omitted, date is rounded to the nearest day. TRUNC(date[, 'fmt']): Returns date with the time portion of the day truncated to the unit that is specified by the format model fmt. If the format model fmt is omitted, date is truncated to the nearest day.

**The CONCAT Function**

The CONCAT function joins two character literals, columns, or expressions to yield one larger character expression. Numeric and date literals are implicitly cast as characters when they occur as parameters to the CONCAT function. Numeric or date expressions are evaluated before being converted to strings ready to be concatenated. The CONCAT function takes two parameters. Its syntax is CONCAT(s1, s2), where s1 and s2 represent string literals, character column values, or expressions resulting in character values.

The INSTR(source string, search item, [start position],[nth occurrence of search item]) function returns a number that represents the position in the source string, beginning from the given start position, where the nth occurrence of the search item begins:

instr('http://www.domain.com',',',1,2) = 18

The TRIM function literally trims off leading or trailing (or both) character strings from a given source string:

**QUESTION 246**

In the CUSTOMERS table, the CUST\_CITY column contains the value 'Paris' for the CUST\_FIRST\_NAME 'ABIGAIL'.

Evaluate the following query:

```
SQL> SELECT INITCAP(cust_first_name || ' ' ||
  UPPER(SUBSTR(cust_city,-LENGTH(cust_city),2)))
FROM customers
WHERE cust_first_name = 'ABIGAIL';
```

What would be the outcome?

- A. Abigail PA
- B. Abigail Pa
- C. Abigail IS
- D. an error message

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 247

View the Exhibit and examine the structure of the CUSTOMERS table. In the CUSTOMERS table, the CUST\_LAST\_NAME column contains the values 'Anderson' and 'Ausson'. You issue the following query:

```
SQL> SELECT LOWER(REPLACE(TRIM('son' FROM cust_last_name),'An','O')) FROM CUSTOMERS
WHERE LOWER(cust_last_name) LIKE 'a%n';
```

What would be the outcome?

- A. 'Oder' and 'Aus'
- B. an error because the TRIM function specified is not valid
- C. an error because the LOWER function specified is not valid
- D. an error because the REPLACE function specified is not valid

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Function Purpose

ROUND(column|expression, n) Rounds the column, expression, or value to n decimal places or, if n is omitted, no decimal places (If n is negative, numbers to the left of decimal point are rounded.)

TRUNC(column|expression, n) Truncates the column, expression, or value to n decimal places or, if n is omitted, n defaults to zero

The TRIM Function

The TRIM function removes characters from the beginning or end of character literals, columns or expressions to yield one potentially shorter character item. Numeric and date literals are automatically cast as characters when they occur as parameters to the TRIM function. Numeric or date expressions are evaluated first before being converted to strings ready to be trimmed. The TRIM function takes a parameter made up of an optional and a mandatory component. Its syntax is TRIM ([trailing|leading|both] trimstring from s).

The string to be trimmed (s) is mandatory. The following points list the rules governing the use of this function: TRIM(s) removes spaces from both sides of the input string. TRIM(trailing trimstring from s) removes all occurrences of trimstring from the end of the string s if it is present.

TRIM(leading trimstring from s) removes all occurrences of trimstring from the beginning of the string s if it is

present.

TRIM(both trimstring from s) removes all occurrences of trimstring from the beginning and end of the string s if it is present.

The following queries illustrate the usage of this function:

Query 1: select trim(trailing 'e' from 1+2.14||' is pie') from dual  
Query 2: select trim(both '\*' from  
\*\*\*\*\*Hidden\*\*\*\*\*') from dual  
Query 3: select trim(1 from sysdate) from dual

ORA-30001: trim set should have only one character

30001. 00000 - "trim set should have only one character"

\*Cause: Trim set contains more or less than 1 character. This is not allowed in TRIM function.

REPLACE(text, search\_string, replacement\_string)

Searches a text expression for a character string and, if found, replaces it with a specified replacement string

#### QUESTION 248

Examine the data in the CUST\_NAME column of the CUSTOMERS table.

CUST\_NAME

-----

Renske Ladwig

Jason Mallin

Samuel McCain

Allan MCEwen

Irene Mikilineni

Julia Nayer

You need to display customers' second names where the second name starts with "Mc" or "MC." Which query gives the required output?

- A. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1)  
FROM customers  
WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1))='Mc';
- B. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1)  
FROM customers  
WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1)) LIKE 'Mc%';
- C. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1)  
FROM customers  
WHERE SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1) LIKE INITCAP('MC%');
- D. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1)  
FROM customers  
WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name, ' ')+1)) = INITCAP('MC%');

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 249

Examine the data in the CUST\_NAME column of the CUSTOMERS table.

CUST\_NAME

-----

Lex De Haan

Renske Ladwig

Jose Manuel Urman

Jason Mallin

You want to extract only those customer names that have three names and display the \* symbol in place of the first name as follows:

CUST NAME

-----  
\*\*\* De Haan

\*\*\*\* Manuel Urman

Which two queries give the required output? (Choose two.)

- A. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*') "CUST NAME"  
FROM customers  
WHERE INSTR(cust\_name, ' ',1,2)<>0;
- B. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*') "CUST NAME"  
FROM customers  
WHERE INSTR(cust\_name, ' ',-1,2)<>0;
- C. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name)- INSTR  
(cust\_name,' '),\*) "CUST NAME"  
FROM customers  
WHERE INSTR(cust\_name, ' ',-1,-2)<>0;
- D. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name)- INSTR(cust\_name,'  
' '),\*) "CUST NAME"  
FROM customers  
WHERE INSTR(cust\_name, ' ',1,2)<>0 ;

**Correct Answer:** AB

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 250

View the Exhibit and examine the structure of the EMPLOYEES table.

##### EMPLOYEES

| Name     | Null?    | Type         |
|----------|----------|--------------|
| -----    | -----    | -----        |
| EMPNO    | NOT NULL | NUMBER(4)    |
| ENAME    |          | VARCHAR2(10) |
| JOB      |          | VARCHAR2(9)  |
| HIREDATE |          | DATE         |
| SAL      |          | NUMBER(7,2)  |
| COMM     |          | NUMBER(7,2)  |
| DEPTNO   |          | NUMBER(2)    |

Examine the data in the ENAME and HIREDATE columns of the EMPLOYEES table:

ENAME HIREDATE

-----  
SMITH 17-DEC-80  
ALLEN 20-FEB-81  
WARD 22-FEB-81

You want to generate a list of user IDs as follows:

USERID

-----  
Smi17DEC80  
All20FEB81  
War22FEB81

You issue the following query:

```
SQL>SELECT CONCAT(SUBSTR(INITCAP(ename),1,3), REPLACE(hiredate,'-')) "USERID" FROM employees;
```

What is the outcome?

- A. It executes successfully and gives the correct output.
- B. It executes successfully but does not give the correct output.
- C. It generates an error because the REPLACE function is not valid.
- D. It generates an error because the SUBSTR function cannot be nested in the CONCAT function.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

REPLACE(text, search\_string,replacement\_string)

Searches a text expression for a character string and, if found, replaces it with a specified replacement string

The REPLACE Function

The REPLACE function replaces all occurrences of a search item in a source string with a replacement term and returns the modified source string. If the length of the replacement term is different from that of the search item, then the lengths of the returned and source strings will be different. If the search string is not found, the source string is returned unchanged. Numeric and date literals and expressions are evaluated before being implicitly cast as characters when they occur as parameters to the REPLACE function.

The REPLACE function takes three parameters, with the first two being mandatory. Its syntax is REPLACE (source string, search item, [replacement term]).

If the replacement term parameter is omitted, each occurrence of the search item is removed from the source string. In other words, the search item is replaced by an empty string. . The following queries illustrate the REPLACE function with numeric and date expressions:

Query 1: select replace(10000-3,'9','85') from dual

Query 2: select replace(sysdate, 'DEC','NOV') from dual

#### QUESTION 251

View the Exhibit and examine the structure and data in the INVOICE table.

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

Which statements are true regarding data type conversion in expressions used in queries? (Choose all that apply.)

- A. inv\_amt ='0255982' : requires explicit conversion
- B. inv\_date > '01-02-2008' : uses implicit conversion
- C. CONCAT(inv\_amt,inv\_date) : requires explicit conversion
- D. inv\_date = '15-february-2008' : uses implicit conversion
- E. inv\_no BETWEEN '101' AND '110' : uses implicit conversion

**Correct Answer:** DE

**Section:** (none)

## Explanation

### Explanation/Reference:

Explanation:

In some cases, the Oracle server receives data of one data type where it expects data of a different data type. When this happens, the Oracle server can automatically convert the data to the expected data type. This data type conversion can be done implicitly by the Oracle server or explicitly by the user.

Explicit data type conversions are performed by using the conversion functions. Conversion functions convert a value from one data type to another. Generally, the form of the function names follows the convention data type TO data type. The first data type is the input data type and the second data type is the output.

Note: Although implicit data type conversion is available, it is recommended that you do the explicit data type conversion to ensure the reliability of your SQL statements.

### QUESTION 252

You want to display the date for the first Monday of the next month and issue the following command:

```
SQL>SELECT TO_CHAR(NEXT_DAY(LAST_DAY(SYSDATE),'MON'), 'dd "is the first Monday for"fmmonth  
rrrr') FROM DUAL;
```

What is the outcome?

- A. It executes successfully and returns the correct result.
- B. It executes successfully but does not return the correct result.
- C. It generates an error because TO\_CHAR should be replaced with TO\_DATE.
- D. It generates an error because rrrr should be replaced by rr in the format string.
- E. It generates an error because fm and double quotation marks should not be used in the format string.

**Correct Answer: A**

**Section: (none)**

## Explanation

### Explanation/Reference:

Explanation:

· NEXT\_DAY(date, 'char'): Finds the date of the next specified day of the week ('char') following date. The value of char may be a number representing a day or a character string. · LAST\_DAY(date): Finds the date of the last day of the month that contains date. The second innermost function is evaluated next. TO\_CHAR('28-OCT-2009', 'fmMonth') converts the given date based on the Month format mask and returns the character string October. The fm modifier trims trailing blank spaces from the name of the month.

### QUESTION 253

Examine the structure and data in the PRICE\_LIST table:

Name Null Type

```
-----  
PROD_ID NOT NULL NUMBER(3)  
PROD_PRICE VARCHAR2(10)  
PROD_ID PROD_PRICE  
-----
```

```
100 $234.55  
101 $6,509.75  
102 $1,234
```

You plan to give a discount of 25% on the product price and need to display the discount amount in the same format as the PROD\_PRICE.

Which SQL statement would give the required result?

- A. SELECT TO\_CHAR(prod\_price\* .25,\$99,999.99)  
FROM PRICE\_LIST;



- B. SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\*.25,'\$99,999.00')  
FROM PRICE\_LIST;
- C. SELECT TO\_CHAR(TO\_NUMBER(prod\_price,'\$99,999.99')\*.25,'\$99,999.00') FROM PRICE\_LIST;
- D. SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'\$99,999.99')\*.25,'\$99,999.00') FROM PRICE\_LIST;

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Use TO\_NUMBER on the prod\_price column to convert from char to number to be able to multiply it with 0.25. Then use the TO\_CHAR function (with formatting '\$99,999.00') to convert the number back to char.

Incorrect:

Not C: Use the formatting '\$99,999.00' with the TO\_CHAR function, not with the TO\_NUMBER function.

Note:

\* Using the TO\_CHAR Function

The TO\_CHAR function returns an item of data type VARCHAR2. When applied to items of type NUMBER, several formatting options are available. The syntax is as follows:

TO\_CHAR(number1, [format], [nls\_parameter]),

The number1 parameter is mandatory and must be a value that either is or can be implicitly converted into a number. The optional format parameter may be used to specify numeric formatting information like width, currency symbol, the position of a decimal point, and group (or thousands) separators and must be enclosed in single

\* Syntax of Explicit Data Type Conversion

Functions

TO\_NUMBER(char1, [format mask], [nls\_parameters]) = num1

TO\_CHAR(num1, [format mask], [nls\_parameters]) = char1

TO\_DATE(char1, [format mask], [nls\_parameters]) = date1

TO\_CHAR(date1, [format mask], [nls\_parameters]) = char1

**QUESTION 254**

View the Exhibit and examine the data in the PROMO\_NAME and PROMO\_END\_DATE columns of the PROMOTIONS table, and the required output format.

| PROMO_NAME                | PROMO_END_DATE |
|---------------------------|----------------|
| post promotion #20-343    | 19-JUN-99      |
| post promotion #20-274    | 16-JUL-99      |
| TV promotion #12-530      | 13-APR-99      |
| post promotion #17-157    | 29-JUN-99      |
| TV promotion #12-481      | 05-JAN-00      |
| newspaper promotion #19-4 | 16-AUG-98      |
| everyday low price        | 01-JAN-99      |

| PROMO_NAME                | LAST_DAY                    |
|---------------------------|-----------------------------|
| post promotion #20-343    | Saturday, June 19, 1999     |
| post promotion #20-274    | Friday, July 16, 1999       |
| TV promotion #12-530      | Tuesday, April 13, 1999     |
| post promotion #17-157    | Tuesday, June 29, 1999      |
| TV promotion #12-481      | Wednesday, January 05, 2000 |
| newspaper promotion #19-4 | Sunday, August 16, 1998     |
| everyday low price        | Friday, January 01, 1999    |

Which two queries give the correct result? (Choose two.)

- A. SELECT promo\_name, TO\_CHAR(promo\_end\_date,'Day') ',' ,  
TO\_CHAR(promo\_end\_date,'Month') ',' ,  
TO\_CHAR(promo\_end\_date,'DD, YYYY') AS last\_day  
FROM promotions;
- B. SELECT promo\_name,TO\_CHAR (promo\_end\_date,'fxDay') ',' ,  
TO\_CHAR(promo\_end\_date,'fxMonth') ','

- ```
TO_CHAR(promo_end_date,'fxDD, YYYY') AS last_day
FROM promotions;
```
- C. SELECT promo\_name, TRIM(TO\_CHAR(promo\_end\_date,'Day')) ' ', TRIM(TO\_CHAR(promo\_end\_date,'Month')) ' ',  
TRIM(TO\_CHAR(promo\_end\_date,'DD, YYYY')) AS last\_day  
FROM promotions;
- D. SELECT promo\_name, TO\_CHAR(promo\_end\_date,'fmDay') ' ',  
TO\_CHAR(promo\_end\_date,'fmMonth') ' ',  
TO\_CHAR(promo\_end\_date,'fmDD, YYYY') AS last\_day  
FROM promotions;

**Correct Answer:** CD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 255

Examine the data in the PROMO\_BEGIN\_DATE column of the PROMOTIONS table:  
PROMO\_BEGIN\_DATE

```
-----
04-jan-00
10-jan-00
15-dec-99
18-oct-98
22-aug-99
```

You want to display the number of promotions started in 1999 and 2000.  
Which query gives the correct output?

- A. SELECT SUM(DECODE(SUBSTR(promo\_begin\_date,8),'00',1,0)) "2000", SUM(DECODE(SUBSTR(promo\_begin\_date,8),'99',1,0)) "1999"  
FROM promotions;
- B. SELECT SUM(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '99' THEN 1 ELSE 0 END) "1999", SUM(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '00' THEN 1 ELSE 0 END) "2000"  
FROM promotions;
- C. SELECT COUNT(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '99' THEN 1 ELSE 0 END) "1999", COUNT(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '00' THEN 1 ELSE 0 END) "2000" FROM promotions;
- D. SELECT COUNT(DECODE(SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8), '1999', 1, 0)) "1999", COUNT(DECODE(SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8), '2000', 1, 0)) "2000"  
FROM promotions;

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 256

Examine the structure of the TRANSACTIONS table:

Name Null Type

```
-----
TRANS_ID NOT NULL NUMBER(3)
```

CUST\_NAME VARCHAR2(30)  
TRANS\_DATE TIMESTAMP  
TRANS\_AMT NUMBER(10,2)

You want to display the date, time, and transaction amount of transactions that were done before 12 noon. The value zero should be displayed for transactions where the transaction amount has not been entered. Which query gives the required result?

- A. 

```
SELECT TO_CHAR(trans_date,'dd-mon-yyyy hh24:mi:ss'),  
       TO_CHAR(trans_amt,'$99999999D99')  
FROM transactions  
WHERE TO_NUMBER(TO_DATE(trans_date,'hh24')) < 12 AND  
       COALESCE(trans_amt,NULL)<>NULL;
```
- B. 

```
SELECT TO_CHAR(trans_date,'dd-mon-yyyy hh24:mi:ss'),  
       NVL(TO_CHAR(trans_amt,'$99999999D99'),0)  
FROM transactions  
WHERE TO_CHAR(trans_date,'hh24') < 12;
```
- C. 

```
SELECT TO_CHAR(trans_date,'dd-mon-yyyy hh24:mi:ss'),  
       COALESCE(TO_NUMBER(trans_amt,'$99999999.99'),0)  
FROM transactions  
WHERE TO_DATE(trans_date,'hh24') < 12;
```
- D. 

```
SELECT TO_DATE (trans_date,'dd-mon-yyyy hh24:mi:ss'),  
       NVL2(trans_amt,TO_NUMBER(trans_amt,'$99999999.99'), 0)  
FROM transactions  
WHERE TO_DATE(trans_date,'hh24') < 12;
```

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 257

Examine the structure of the TRANSACTIONS table:

Name Null Type  
TRANS\_ID NOT NULL NUMBER(3)  
CUST\_NAME VARCHAR2(30)  
TRANS\_DATE DATE  
TRANS\_AMT NUMBER(10,2)

You want to display the transaction date and specify whether it is a weekday or weekend. Evaluate the following two queries:

```
SQL>SELECT TRANS_DATE,CASE  
WHEN TRIM(TO_CHAR(trans_date,'DAY')) IN ('SATURDAY','SUNDAY') THEN 'weekend'  
ELSE 'weekday'  
END "Day Type"  
FROM transactions;
```

```
SQL>SELECT TRANS_DATE, CASE  
WHEN TO_CHAR(trans_date,'DAY') BETWEEN 'MONDAY' AND 'FRIDAY' THEN 'weekday'  
ELSE 'weekend'  
END "Day Type"FROM transactions;
```

Which statement is true regarding the above queries?

- A. Both give wrong results.
- B. Both give the correct result.
- C. Only the first query gives the correct result.
- D. Only the second query gives the correct result.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

Range Conditions Using the BETWEEN Operator

Use the BETWEEN operator to display rows based on a range of values:

```
SELECT last_name, salary
```

```
FROM employees
```

```
WHERE salary BETWEEN 2500 AND 3500;
```

Range Conditions Using the BETWEEN Operator

You can display rows based on a range of values using the BETWEEN operator. The range that you specify contains a lower limit and an upper limit.

The SELECT statement in the slide returns rows from the EMPLOYEES table for any employee whose salary is between \$2,500 and \$3,500.

Values that are specified with the BETWEEN operator are inclusive. However, you must specify the lower limit first.

You can also use the BETWEEN operator on character values:

```
SELECT last_name
```

```
FROM employees
```

```
WHERE last_name BETWEEN 'King' AND 'Smith';
```

**QUESTION 258**

Examine the structure of the PROMOS table:

Name	Null	Type
PROMO_ID	NOT NULL	NUMBER(3)
PROMO_NAME		VARCHAR2(30)
PROMO_START_DATE	NOT NULL	DATE
PROMO_END_DATE		DATE

You want to generate a report showing promo names and their duration (number of days). If the PROMO\_END\_DATE has not been entered, the message 'ONGOING' should be displayed. Which queries give the correct output? (Choose all that apply.)

- A. `SELECT promo_name, TO_CHAR(NVL(promo_end_date - promo_start_date, 'ONGOING')) FROM promos;`
- B. `SELECT promo_name, COALESCE(TO_CHAR(promo_end_date - promo_start_date), 'ONGOING') FROM promos;`
- C. `SELECT promo_name, NVL(TO_CHAR(promo_end_date - promo_start_date), 'ONGOING') FROM promos;`
- D. `SELECT promo_name, DECODE(promo_end_date - promo_start_date, NULL, 'ONGOING', promo_end_date - promo_start_date) FROM promos;`
- E. `SELECT promo_name, decode(coalesce(promo_end_date, promo_start_date), null, 'ONGOING', promo_end_date - promo_start_date) FROM promos;`

**Correct Answer:** BCD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 259

Examine the structure of the PROMOS table:

Name	Null	Type
PROMO_ID	NOT NULL	NUMBER(3)
PROMO_NAME		VARCHAR2(30)
PROMO_START_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

You want to display the list of promo names with the message 'Same Day' for promos that started and ended on the same day.

Which query gives the correct output?

- A. SELECT promo\_name, NVL(NULLIF(promo\_start\_date, promo\_end\_date), 'Same Day') FROM promos;
- B. SELECT promo\_name, NVL(TRUNC(promo\_end\_date - promo\_start\_date), 'Same Day') FROM promos;
- C. SELECT promo\_name, NVL2(TO\_CHAR(TRUNC(promo\_end\_date-promo\_start\_date)), NULL,'Same Day') FROM promos;
- D. SELECT promo\_name, DECODE((NULLIF(promo\_start\_date, promo\_end\_date)), NULL,'Same day') FROM promos;

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

The NULLIF Function

The NULLIF function tests two terms for equality. If they are equal the function returns a null, else it returns the first of the two terms tested.

The NULLIF function takes two mandatory parameters of any data type. The syntax is NULLIF(ifunequal, comparison\_term), where the parameters ifunequal and comparison\_term are compared. If they are identical, then NULL is returned. If they differ, the ifunequal parameter is returned ANSWER A - date and String incompatibl;a datatypes for NVL function The Date TRUNC Function

The date TRUNC function performs a truncation operation on a date value based on a specified date precision format.

The date TRUNC function takes one mandatory and one optional parameter. Its syntax is TRUNC(source date, [date precision format]). The source date parameter represents any value that can be implicitly converted into a date item. The date precision format parameter specifies the degree of truncation and is optional. If it is absent, the default degree of truncation is day. This means that any time component

#### QUESTION 260

Examine the data in the LIST\_PRICE and MIN\_PRICE columns of the PRODUCTS table:

LIST_PRICE	MIN_PRICE
10000	8000
20000	
30000	30000

Which two expressions give the same output? (Choose two.)

- A. NVL(NULLIF(list\_price, min\_price), 0)
- B. NVL(COALESCE(list\_price, min\_price), 0)
- C. NVL2(COALESCE(list\_price, min\_price), min\_price, 0)
- D. COALESCE(NVL2(list\_price, list\_price, min\_price), 0)

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the COALESCE Function

- The advantage of the COALESCE function over the NVL function is that the COALESCE function can take multiple alternate values.

- If the first expression is not null, the COALESCE function returns that expression; otherwise, it does a COALESCE of the remaining expressions.

Using the COALESCE Function

The COALESCE function returns the first non-null expression in the list.

Syntax

COALESCE (expr1, expr2, ... exprn)

In the syntax:

- expr1 returns this expression if it is not null

- expr2 returns this expression if the first expression is null and this expression is not null · exprn returns this expression if the preceding expressions are null Note that all expressions must be of the same data type.

#### QUESTION 261

View the Exhibit and examine the structure and data in the INVOICE table. Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT AVG(inv\_date )  
FROM invoice;
- B. SELECT MAX(inv\_date),MIN(cust\_id)  
FROM invoice;
- C. SELECT MAX(AVG(SYSDATE - inv\_date))  
FROM invoice;
- D. SELECT AVG( inv\_date - SYSDATE), AVG(inv\_amt)  
FROM invoice;

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the AVG and SUM Functions

You can use the AVG, SUM, MIN, and MAX functions against the columns that can store numeric data. The

example in the slide displays the average, highest, lowest, and sum of monthly salaries for all sales representatives

Using the MIN and MAX Functions

You can use the MAX and MIN functions for numeric, character, and date data types. The example in the slide displays the most junior and most senior employees.

#### QUESTION 262

Examine the structure of the ORDERS table:

Name	Null	Type
ORDER_ID	NOT NULL	NUMBER(12)
ORDER_DATE	NOT NULL	TIMESTAMP(6)
CUSTOMER_ID	NOT NULL	NUMBER(6)
ORDER_STATUS		NUMBER(2)
ORDER_TOTAL		NUMBER(8,2)

You want to find the total value of all the orders for each year and issue the following command:

```
SQL>SELECT TO_CHAR(order_date,'rr'), SUM(order_total)
FROM orders
GROUP BY TO_CHAR(order_date,'yyyy');
```

Which statement is true regarding the outcome?

- A. It executes successfully and gives the correct output.
- B. It gives an error because the TO\_CHAR function is not valid.
- C. It executes successfully but does not give the correct output.
- D. It gives an error because the data type conversion in the SELECT list does not match the data type conversion in the GROUP BY clause.

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 263

You issue the following query:

```
SQL> SELECT AVG(MAX(qty))
FROM ord_items
GROUP BY item_no
HAVING AVG(MAX(qty))>50;
```

Which statement is true regarding the outcome of this query?

- A. It executes successfully and gives the correct output.
- B. It gives an error because the HAVING clause is not valid.
- C. It executes successfully but does not give the correct output.
- D. It gives an error because the GROUP BY expression is not valid.

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

The general form of the SELECT statement is further enhanced by the addition of the HAVING clause and becomes:

SELECT column|expression|group\_function(column|expression [alias]),... FROM table

[WHERE condition(s)]

[GROUP BY {col(s)|expr}]

[HAVING group\_condition(s)]

[ORDER BY {col(s)|expr|numeric\_pos} [ASC|DESC] [NULLS FIRST|LAST]]; An important difference between the HAVING clause and the other SELECT statement clauses is that it may only be specified if a GROUP BY clause is present. This dependency is sensible since group-level rows must exist before they can be restricted. The HAVING clause can occur before the GROUP BY clause in the SELECT statement. However, it is more common to place the HAVING clause after the GROUP BY clause. All grouping is performed and group functions are executed prior to evaluating the HAVING clause.

**QUESTION 264**

Which statements are true regarding the WHERE and HAVING clauses in a SELECT statement? (Choose all that apply.)

- A. The HAVING clause can be used with aggregate functions in subqueries.
- B. The WHERE clause can be used to exclude rows after dividing them into groups.
- C. The WHERE clause can be used to exclude rows before dividing them into groups.
- D. The aggregate functions and columns used in the HAVING clause must be specified in the SELECT list of the query.
- E. The WHERE and HAVING clauses can be used in the same statement only if they are applied to different columns in the table.

**Correct Answer: AC**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 265**

Examine the data in the ORD\_ITEMS table:

ORD\_NO ITEM\_NO QTY

-----

1 111 10

1 222 20

1 333 30

2 333 30

2 444 40

3 111 40

Evaluate the following query:

```
SQL>SELECT item_no, AVG(qty)
FROM ord_items
HAVING AVG(qty) > MIN(qty) * 2
GROUP BY item_no;
```

Which statement is true regarding the outcome of the above query?



- A. It gives an error because the HAVING clause should be specified after the GROUP BY clause.
- B. It gives an error because all the aggregate functions used in the HAVING clause must be specified in the SELECT list.
- C. It displays the item nos with their average quantity where the average quantity is more than double the minimum quantity of that item in the table.
- D. It displays the item nos with their average quantity where the average quantity is more than double the overall minimum quantity of all the items in the table.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 266

Examine the structure of the CUSTOMERS table:

Name	Null Type
CUSTNO	NOT NULL NUMBER(3)
CUSTNAME	NOT NULL VARCHAR2(25)
CUSTADDRESS	VARCHAR2(35)
CUST_CREDIT_LIMIT	NUMBER(5)

CUSTNO is the PRIMARY KEY in the table. You want to find out if any customers' details have been entered more than once using different CUSTNO, by listing all the duplicate names. Which two methods can you use to get the required result? (Choose two.)

- A. self-join
- B. subquery
- C. full outer-join with self-join
- D. left outer-join with self-join
- E. right outer-join with self-join

**Correct Answer: AB**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 267

View the Exhibit and examine the data in the PROJ\_TASK\_DETAILS table.

PROJ_TASK_DETAILS				
TASK_ID	BASED_ON	TASK_IN_CHARGE	TASK_START_DATE	TASK_END_DATE
P01		KING	10-SEP-07	12-SEP-07
P02	P01	KOCHAR	13-SEP-07	14-SEP-07
P03		GREEN	14-SEP-07	18-SEP-07
P04	P03	SCOTT	19-SEP-07	20-SEP-07

The PROJ\_TASK\_DETAILS table stores information about tasks involved in a project and the relation between them.

The BASED\_ON column indicates dependencies between tasks. Some tasks do not depend on the completion of any other tasks.

You need to generate a report showing all task IDs, the corresponding task ID they are dependent on, and the name of the employee in charge of the task it depends on.

Which query would give the required result?

- A. `SELECT p.task_id, p.based_on, d.task_in_charge  
FROM proj_task_details p JOIN proj_task_details d  
ON (p.based_on = d.task_id);`
- B. `SELECT p.task_id, p.based_on, d.task_in_charge  
FROM proj_task_details p LEFT OUTER JOIN proj_task_details d ON (p.based_on =`
- C. `task_id);`
- D. `SELECT p.task_id, p.based_on, d.task_in_charge  
FROM proj_task_details p FULL OUTER JOIN proj_task_details d ON (p.based_on =`
- E. `task_id);`
- F. `SELECT p.task_id, p.based_on, d.task_in_charge  
FROM proj_task_details p JOIN proj_task_details d  
ON (p.task_id = d.task_id);`

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 268

Examine the data in the CUSTOMERS table:

CUSTNO	CUSTNAME	CITY
1	KING	SEATTLE
2	GREEN	BOSTON
3	KOCHAR	SEATTLE
4	SMITH	NEW YORK

You want to list all cities that have more than one customer along with the customer details.

Evaluate the following query:

```
SQL>SELECT c1.custname, c1.city  
FROM Customers c1 _____ Customers c2  
ON (c1.city=c2.city AND c1.custname<>c2.custname);
```

Which two JOIN options can be used in the blank in the above query to give the correct output? (Choose two.)

- A. JOIN
- B. NATURAL JOIN
- C. LEFT OUTER JOIN
- D. FULL OUTER JOIN
- E. RIGHT OUTER JOIN

**Correct Answer:** AE

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 269**

View the Exhibits and examine the structures of the CUSTOMERS, SALES, and COUNTRIES tables. You need to generate a report that shows all country names, with corresponding customers (if any) and sales details (if any), for all customers. Which FROM clause gives the required result?

- A. FROM sales JOIN customers USING (cust\_id)  
FULL OUTER JOIN countries USING (country\_id);
- B. FROM sales JOIN customers USING (cust\_id)  
RIGHT OUTER JOIN countries USING (country\_id);
- C. FROM customers LEFT OUTER JOIN sales USING (cust\_id)  
RIGHT OUTER JOIN countries USING (country\_id);
- D. FROM customers LEFT OUTER JOIN sales USING (cust\_id)  
LEFT OUTER JOIN countries USING (country\_id);

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 270**

View the Exhibit and examine the data in the EMPLOYEES table:  
You want to display all the employee names and their corresponding manager names.  
Evaluate the following query:  
SQL> SELECT e.employee\_name "EMP NAME", m.employee\_name "MGR NAME" FROM employees e  
\_\_\_\_ employees m  
ON e.manager\_id = m.employee\_id;  
Which JOIN option can be used in the blank in the above query to get the required output? Exhibit:

- A. only inner JOIN
- B. only FULL OUTER JOIN
- C. only LEFT OUTER JOIN
- D. only RIGHT OUTER JOIN

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 271**

View the Exhibit and examine the structure of the PRODUCT, COMPONENT, and PDT\_COMP tables.

PRODUCT		
Name	Null?	Type
PDTNO	NOT NULL	NUMBER(3)
PDTNAME		VARCHAR2(25)
QTY		NUMBER(6,2)
COMPONENT		
Name	Null?	Type
COMPNO	NOT NULL	NUMBER(4)
COMPNAME		VARCHAR2(25)
QTY		NUMBER(6,2)
PDT_COMP		
Name	Null?	Type
PDTNO	NOT NULL	NUMBER(2)

In PRODUCT table, PDTNO is the primary key.

In COMPONENT table, COMPNO is the primary key.

In PDT\_COMP table, (PDTNO,COMPNO) is the primary key, PDTNO is the foreign key referencing PDTNO in PRODUCT table and COMPNO is the foreign key referencing the COMPNO in COMPONENT table.

You want to generate a report listing the product names and their corresponding component names, if the component names and product names exist.

Evaluate the following query:

```
SQL>SELECT pdtno,pdtname, compno,compname
FROM product _____ pdt_comp
USING (pdtno) _____ component USING(compno)
WHERE compname IS NOT NULL;
```

Which combination of joins used in the blanks in the above query gives the correct output?

- A. JOIN; JOIN
- B. FULL OUTER JOIN; FULL OUTER JOIN
- C. RIGHT OUTER JOIN; LEFT OUTER JOIN
- D. LEFT OUTER JOIN; RIGHT OUTER JOIN

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 272

View the Exhibit and examine the structure of the SALES and PRODUCTS tables.

**SALES**

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER (3)
CUST_ID	NOT NULL	NUMBER (4)
TIME_ID		DATE
QTY_SOLD		NUMBER (10,2)

**PRODUCTS**

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER (3)
PROD_NAME		VARCHAR2 (30)
PROD_LIST_PRICE		NUMBER (8,2)

In the SALES table, PROD\_ID is the foreign key referencing PROD\_ID in the PRODUCTS table. You want to list each product ID and the number of times it has been sold. Evaluate the following query:

```
SQL>SELECT p.prod_id, COUNT(s.prod_id)
FROM products p _____ sales s
ON p.prod_id = s.prod_id
GROUP BY p.prod_id;
```

Which two JOIN options can be used in the blank in the above query to get the required output? (Choose two.)

- A. JOIN
- B. FULL OUTER JOIN
- C. LEFT OUTER JOIN
- D. RIGHT OUTER JOIN

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 273**

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

- A. A subquery can retrieve zero or more rows.
- B. Only two subqueries can be placed at one level.
- C. A subquery can be used only in SQL query statements.
- D. A subquery can appear on either side of a comparison operator.
- E. There is no limit on the number of subquery levels in the WHERE clause of a SELECT statement.

**Correct Answer:** AD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

### Using a Subquery to Solve a Problem

Suppose you want to write a query to find out who earns a salary greater than Abel's salary. To solve this problem, you need two queries: one to find how much Abel earns, and a second query to find who earns more than that amount.

You can solve this problem by combining the two queries, placing one query inside the other query. The inner query (or subquery) returns a value that is used by the outer query (or main query).

Using a subquery is equivalent to performing two sequential queries and using the result of the first query as the search value in the second query.

### Subquery Syntax

A subquery is a SELECT statement that is embedded in the clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries. They can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself.

You can place the subquery in a number of SQL clauses, including the following:

WHERE clause

HAVING clause

FROM clause

In the syntax:

operator includes a comparison condition such as >, =, or IN. Note: Comparison conditions fall into two classes: single-row operators (>, =, >=, <, <>, <=) and multiple-row operators (IN, ANY, ALL, EXISTS).

The subquery is often referred to as a nested SELECT, sub-SELECT, or inner SELECT statement. The subquery generally executes first, and its output is used to complete the query condition for the main (or outer) query.

### Guidelines for Using Subqueries

Enclose subqueries in parentheses. Place subqueries on the right side of the comparison condition for readability. (However, the subquery can appear on either side of the comparison operator.) Use single-row operators with single-row subqueries and multiple-row operators with multiple-row subqueries.

Subqueries can be nested to an unlimited depth in a FROM clause but to "only" 255 levels in a WHERE clause. They can be used in the SELECT list and in the FROM, WHERE, and HAVING clauses of a query.

### QUESTION 274

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

- A. Subqueries can contain GROUP BY and ORDER BY clauses.
- B. Main query and subquery can get data from different tables.
- C. Main query and subquery must get data from the same tables.
- D. Subqueries can contain ORDER BY but not the GROUP BY clause.
- E. Only one column or expression can be compared between the main query and subquery.
- F. Multiple columns or expressions can be compared between the main query and subquery.

**Correct Answer:** ABF

**Section:** (none)

**Explanation**

### Explanation/Reference:

Explanation:

SUBQUERIES can be used in the SELECT list and in the FROM, WHERE, and HAVING clauses of a query.

A subquery can have any of the usual clauses for selection and projection. The following are required clauses:

A SELECT list

A FROM clause

The following are optional clauses:

WHERE

GROUP BY

HAVING

The subquery (or subqueries) within a statement must be executed before the parent query that calls it, in order

that the results of the subquery can be passed to the parent.

### QUESTION 275

View the Exhibit and examine the structure of the PRODUCTS table.

Table PRODUCTS		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(6)
PROD_NAME	NOT NULL	VARCHAR2(50)
PROD_DESC	NOT NULL	VARCHAR2(4000)
PROD_CATEGORY	NOT NULL	VARCHAR2(50)
PROD_CATEGORY_ID	NOT NULL	NUMBER
PROD_UNIT_OF_MEASURE		VARCHAR2(20)
SUPPLIER_ID	NOT NULL	NUMBER(6)
PROD_STATUS	NOT NULL	VARCHAR2(20)
PROD_LIST_PRICE	NOT NULL	NUMBER(8,2)
PROD_MIN_PRICE	NOT NULL	NUMBER(8,2)

Which two tasks would require subqueries? (Choose two.)

- A. Display the minimum list price for each product status.
- B. Display all suppliers whose list price is less than 1000.
- C. Display the number of products whose list price is more than the average list price.
- D. Display the total number of products supplied by supplier 102 and have product status as 'obsolete'.
- E. Display all products whose minimum list price is more than the average list price of products and have the status 'orderable'.

**Correct Answer:** CE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 276

View the Exhibits and examine PRODUCTS and SALES tables.

Exhibit_Products		
Table PRODUCTS		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(6)
PROD_NAME	NOT NULL	VARCHAR2(50)
PROD_DESC	NOT NULL	VARCHAR2(4000)
PROD_CATEGORY	NOT NULL	VARCHAR2(50)
PROD_CATEGORY_ID	NOT NULL	NUMBER
PROD_UNIT_OF_MEASURE		VARCHAR2(20)
SUPPLIER_ID	NOT NULL	NUMBER(6)
PROD_STATUS	NOT NULL	VARCHAR2(20)
PROD_LIST_PRICE	NOT NULL	NUMBER(8,2)
PROD_MIN_PRICE	NOT NULL	NUMBER(8,2)

Exhibit_Sales		
Table SALES		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER
CUST_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
CHANNEL_ID	NOT NULL	NUMBER
PROMO_ID	NOT NULL	NUMBER
QUANTITY_SOLD	NOT NULL	NUMBER(10,2)

You issue the following query to display product name and the number of times the product has been sold:

```
SQL>SELECT p.prod_name, i.item_cnt
FROM (SELECT prod_id, COUNT(*) item_cnt
FROM sales
GROUP BY prod_id) i RIGHT OUTER JOIN products p
ON i.prod_id = p.prod_id;
```

What happens when the above statement is executed?

- A. The statement executes successfully and produces the required output.
- B. The statement produces an error because ITEM\_CNT cannot be displayed in the outer query.
- C. The statement produces an error because a subquery in the FROM clause and outer-joins cannot be used together.
- D. The statement produces an error because the GROUP BY clause cannot be used in a subquery in the FROM clause.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 277

Examine the structure of the PRODUCTS table:



Name	Null	Type
PROD_ID	NOT NULL	NUMBER(4)
PROD_NAME		VARCHAR2(20)
PROD_STATUS		VARCHAR2(6)
QTY_IN_HAND		NUMBER(8,2)
UNIT_PRICE		NUMBER(10,2)

You want to display the names of the products that have the highest total value for UNIT\_PRICE \* QTY\_IN\_HAND.

Which SQL statement gives the required output?

- A. SELECT prod\_name  
FROM products  
WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products);
- B. SELECT prod\_name  
FROM products  
WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products  
GROUP BY prod\_name);
- C. SELECT prod\_name  
FROM products  
GROUP BY prod\_name  
HAVING MAX(unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products  
GROUP BY prod\_name);
- D. SELECT prod\_name  
FROM products  
WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(SUM(unit\_price \* qty\_in\_hand)) FROM products)  
GROUP BY prod\_name;

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 278

View the Exhibit and examine the structure of the PRODUCTS table.

Table PRODUCTS		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(6)
PROD_NAME	NOT NULL	VARCHAR2(50)
PROD_DESC	NOT NULL	VARCHAR2(4000)
PROD_CATEGORY	NOT NULL	VARCHAR2(50)
PROD_CATEGORY_ID	NOT NULL	NUMBER
PROD_UNIT_OF_MEASURE		VARCHAR2(20)
SUPPLIER_ID	NOT NULL	NUMBER(6)
PROD_STATUS	NOT NULL	VARCHAR2(20)
PROD_LIST_PRICE	NOT NULL	NUMBER(8,2)
PROD_MIN_PRICE	NOT NULL	NUMBER(8,2)

You want to display the category with the maximum number of items.  
You issue the following query:

SQL>SELECT COUNT(\*),prod\_category\_id

```

FROM products
GROUP BY prod_category_id
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM products);

```

What is the outcome?

- A. It executes successfully and gives the correct output.
- B. It executes successfully but does not give the correct output.
- C. It generates an error because the subquery does not have a GROUP BY clause.
- D. It generates an error because = is not valid and should be replaced by the IN operator.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 279

Evaluate the following SQL statement:

```

SQL> SELECT cust_id, cust_last_name
FROM customers
WHERE cust_credit_limit IN
(select cust_credit_limit
FROM customers
WHERE cust_city ='Singapore');

```

Which statement is true regarding the above query if one of the values generated by the subquery is NULL?

- A. It produces an error.
- B. It executes but returns no rows.
- C. It generates output for NULL as well as the other values produced by the subquery.
- D. It ignores the NULL value and generates output for the other values produced by the subquery.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### QUESTION 280

View the Exhibit and examine the structure of the PROMOTIONS table.

Table PROMOTIONS		
Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(6)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY_ID	NOT NULL	NUMBER
PROMO_CATEGORY	NOT NULL	VARCHAR2(30)
PROMO_CATEGORY_ID	NOT NULL	NUMBER
PROMO_COST	NOT NULL	NUMBER(10,2)
PROMO_BEGIN_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

Evaluate the following SQL statement:

```
SQL>SELECT promo_name,CASE
WHEN promo_cost >=(SELECT AVG(promo_cost)
FROM promotions
WHERE promo_category='TV')
then 'HIGH'
else 'LOW'
END COST_REMARK
FROM promotions;
```

Which statement is true regarding the outcome of the above query?

- A. It shows COST\_REMARK for all the promos in the table.
- B. It produces an error because the subquery gives an error.
- C. It shows COST\_REMARK for all the promos in the promo category 'TV'.
- D. It produces an error because subqueries cannot be used with the CASE expression.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 281

View the Exhibit and examine the structure of the PRODUCTS tables. You want to generate a report that displays the average list price of product categories where the average list price is less than half the maximum in each category.

Which query would give the correct output?

- A. 

```
SELECT prod_category,avg(prod_list_price)
FROM products
GROUP BY prod_category
HAVING avg(prod_list_price) < ALL
(SELECT max(prod_list_price)/2
FROM products
GROUP BY prod_category);
```
- B. 

```
SELECT prod_category,avg(prod_list_price)
FROM products
GROUP BY prod_category
HAVING avg(prod_list_price) > ANY
(SELECT max(prod_list_price)/2
FROM products
GROUP BY prod_category);
```
- C. 

```
SELECT prod_category,avg(prod_list_price)
FROM products
HAVING avg(prod_list_price) < ALL
(SELECT max(prod_list_price)/2
FROM products
GROUP BY prod_category);
```
- D. 

```
SELECT prod_category,avg(prod_list_price)
FROM products
GROUP BY prod_category
HAVING avg(prod_list_price) > ANY
(SELECT max(prod_list_price)/2
FROM products);
```

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the ANY Operator in Multiple-Row Subqueries

The ANY operator (and its synonym, the SOME operator) compares a value to each value returned by a subquery.

<ANY means less than the maximum.

>ANY means more than the minimum.

=ANY is equivalent to IN

Using the ALL Operator in Multiple-Row Subqueries

The ALL operator compares a value to every value returned by a subquery.

>ALL means more than the maximum and

<ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators.

### QUESTION 282

View the Exhibits and examine the structures of the COSTS and PROMOTIONS tables.

Evaluate the following SQL statement:

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

What would be the outcome of the above SQL statement?

- A. It displays prod IDs in the promo with the lowest cost.
- B. It displays prod IDs in the promos with the lowest cost in the same time interval.
- C. It displays prod IDs in the promos with the highest cost in the same time interval.
- D. It displays prod IDs in the promos with cost less than the highest cost in the same time interval.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 283

View the Exhibits and examine the structures of the PRODUCTS and SALES tables. Which two SQL statements would give the same output? (Choose two.)

- A. SELECT prod\_id FROM products  
INTERSECT  
SELECT prod\_id FROM sales;
- B. SELECT prod\_id FROM products  
MINUS  
SELECT prod\_id FROM sales;
- C. SELECT DISTINCT p.prod\_id  
FROM products p JOIN sales s  
ON p.prod\_id=s.prod\_id;
- D. SELECT DISTINCT p.prod\_id  
FROM products p JOIN sales s  
ON p.prod\_id <> s.prod\_id;

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 284

View the Exhibit and examine the structure of the ORDERS and CUSTOMERS tables.

##### ORDERS

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER (12)
ORDER_DATE		DATE
CUSTOMER_ID	NOT NULL	NUMBER (6)
ORDER_TOTAL		NUMBER (8, 2)

##### CUSTOMERS

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER (6)
CUST_NAME	NOT NULL	VARCHAR2 (20)
CUST_ADDRESS		VARCHAR2 (50)
CREDIT_LIMIT		NUMBER (9, 2)

Evaluate the following SQL command:

```
SQL> SELECT o.order_id, c.cust_name, o.order_total, c.credit_limit FROM orders o JOIN customers c
      USING (customer_id)
      WHERE o.order_total > c.credit_limit
      FOR UPDATE
      ORDER BY o.order_id;
```

Which two statements are true regarding the outcome of the above query? (Choose two.)

- A. It locks all the rows that satisfy the condition in the statement.
- B. It locks only the columns that satisfy the condition in both the tables.
- C. The locks are released only when a COMMIT or ROLLBACK is issued.
- D. The locks are released after a DML statement is executed on the locked rows.

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

FOR UPDATE Clause in a SELECT Statement

· Locks the rows in the EMPLOYEES table where job\_id is SA\_REP. · Lock is released only when you issue a ROLLBACK or a COMMIT. · If the SELECT statement attempts to lock a row that is locked by another user, the database waits until the row is available, and then returns the results of the SELECT statement

```
SELECT employee_id, salary, commission_pct, job_id
FROM employees
WHERE job_id = 'SA_REP'
```

FOR UPDATE  
ORDER BY employee\_id;

**QUESTION 285**

Which statements are true regarding the FOR UPDATE clause in a SELECT statement? (Choose all that apply.)

- A. It locks only the columns specified in the SELECT list.
- B. It locks the rows that satisfy the condition in the SELECT statement.
- C. It can be used only in SELECT statements that are based on a single table.
- D. It can be used in SELECT statements that are based on a single or multiple tables.
- E. After it is enforced by a SELECT statement, no other query can access the same rows until a COMMIT or ROLLBACK is issued.

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

FOR UPDATE Clause in a SELECT Statement

Locks the rows in the EMPLOYEES table where job\_id is SA\_REP. Lock is released only when you issue a ROLLBACK or a COMMIT. If the SELECT statement attempts to lock a row that is locked by another user, the database waits until the row is available, and then returns the results of the SELECT statement.

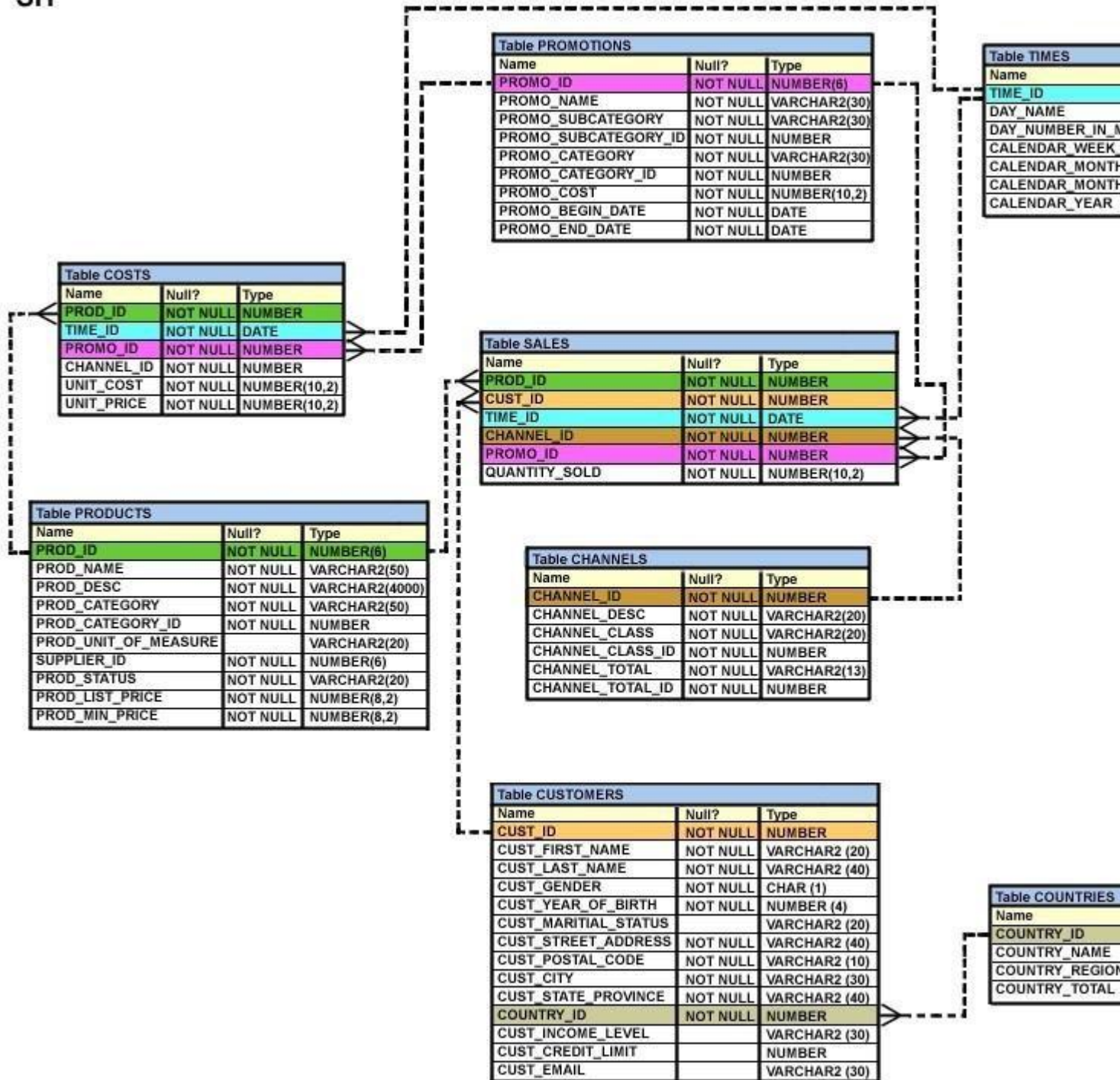
FOR UPDATE Clause in a SELECT Statement

When you issue a SELECT statement against the database to query some records, no locks are placed on the selected rows. In general, this is required because the number of records locked at any given time is (by default) kept to the absolute minimum: only those records that have been changed but not yet committed are locked. Even then, others will be able to read those records as they appeared before the change (the "before image" of the data). There are times, however, when you may want to lock a set of records even before you change them in your program. Oracle offers the FOR UPDATE clause of the SELECT statement to perform this locking. When you issue a SELECT...FOR UPDATE statement, the relational database management system (RDBMS) automatically obtains exclusive row-level locks on all the rows identified by the SELECT statement, thereby holding the records "for your changes only." No one else will be able to change any of these records until you perform a ROLLBACK or a COMMIT. You can append the optional keyword NOWAIT to the FOR UPDATE clause to tell the Oracle server not to wait if the table has been locked by another user. In this case, control will be returned immediately to your program or to your SQL Developer environment so that you can perform other work, or simply wait for a period of time before trying again. Without the NOWAIT clause, your process will block until the table is available, when the locks are released by the other user through the issue of a COMMIT or a ROLLBACK command.

**QUESTION 286**

View the exhibit and examine the description for the SALES and CHANNELS tables.

SH



You issued the following SQL statement to insert a row in the SALES table:

```
INSERT INTO sales VALUES
(23, 2300, SYSDATE, (SELECT channel_id
FROM channels
WHERE channel_desc='Direct Sales'), 12, 1, 500);
```

Which statement is true regarding the execution of the above statement?

- A. The statement will execute and the new row will be inserted in the SALES table.
- B. The statement will fail because subquery cannot be used in the VALUES clause.
- C. The statement will fail because the VALUES clause is not required with subquery.
- D. The statement will fail because subquery in the VALUES clause is not enclosed with in single quotation marks.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 287**

View the Exhibit and examine the structure of the PRODUCTS, SALES, and SALE\_SUMMARY tables.

SALE\_VW is a view created using the following command:

```
SQL>CREATE VIEW sale_vw AS  
SELECT prod_id, SUM(quantity_sold) QTY_SOLD  
FROM sales GROUP BY prod_id;
```

You issue the following command to add a row to the SALE\_SUMMARY table:

```
SQL>INSERT INTO sale_summary  
SELECT prod_id, prod_name, qty_sold FROM sale_vw JOIN products USING (prod_id) WHERE prod_id = 16;
```

What is the outcome?

- A. It executes successfully.
- B. It gives an error because a complex view cannot be used to add data into the SALE\_SUMMARY table.
- C. It gives an error because the column names in the subquery and the SALE\_SUMMARY table do not match.
- D. It gives an error because the number of columns to be inserted does not match with the number of columns in the SALE\_SUMMARY table.

**Correct Answer:** D

**Section:** (none)

**Explanation**

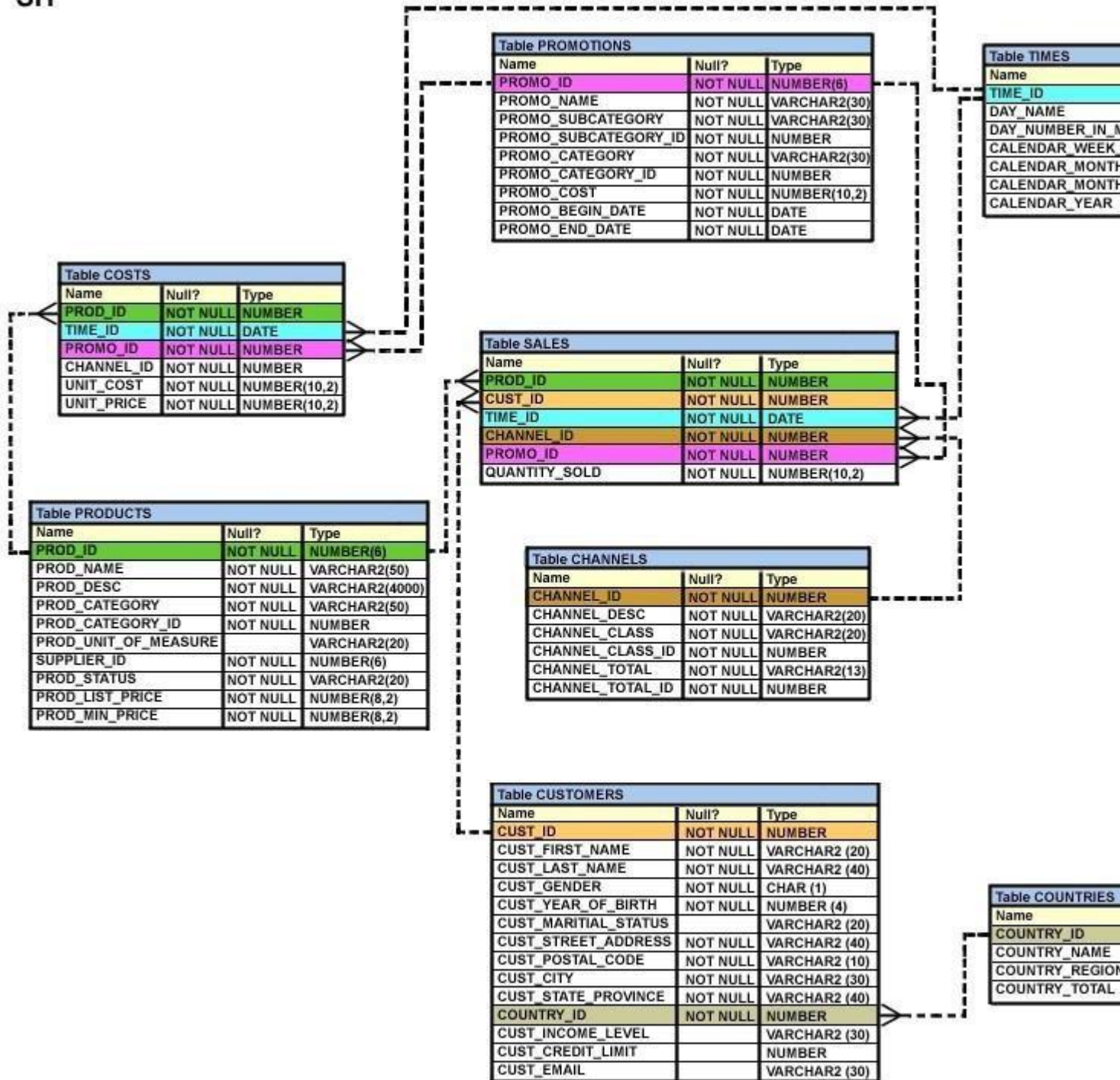
**Explanation/Reference:**

#### **QUESTION 288**

View the Exhibit and examine the structure of CUSTOMERS and SALES tables.



SH



Evaluate the following SQL statement:

```
UPDATE (SELECT prod_id, cust_id, quantity_sold, time_id
FROM sales)
SET time_id = '22-MAR-2007'
WHERE cust_id = (SELECT cust_id
FROM customers
WHERE cust_last_name = 'Roberts' AND
credit_limit = 600);
```

Which statement is true regarding the execution of the above UPDATE statement?

- A. It would not execute because two tables cannot be used in a single UPDATE statement.
- B. It would not execute because the SELECT statement cannot be used in place of the table name.
- C. It would execute and restrict modifications to only the columns specified in the SELECT statement.
- D. It would not execute because a subquery cannot be used in the WHERE clause of an UPDATE statement.

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:

One UPDATE statement can change rows in only one table, but it can change any number of rows in that table.

#### **QUESTION 289**

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

You want to update the EMPLOYEES table as follows: 4 ? 4;

- 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 the outcome?

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

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

#### **QUESTION 290**

Which two statements are true regarding the DELETE and TRUNCATE commands? (Choose two.)

- A. DELETE can be used to remove only rows from only one table at a time.
- B. DELETE can be used to remove only rows from multiple tables at a time.
- C. DELETE can be used only on a table that is a parent of a referential integrity constraint.
- D. DELETE can be used to remove data from specific columns as well as complete rows.
- E. DELETE and TRUNCATE can be used on a table that is a parent of a referential integrity constraint having ON DELETE rule.

**Correct Answer:** AE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Transactions, consisting of INSERT, UPDATE, and DELETE (or even MERGE) commands can be made permanent (with a COMMIT) or reversed (with a ROLLBACK). A TRUNCATE command, like any other DDL command, is immediately permanent: it can never be reversed.

The Transaction Control Statements

A transaction begins implicitly with the first DML statement. There is no command to explicitly start a transaction. The transaction continues through all subsequent DML statements issued by the session. These statements can be against any number of tables: a transaction is not restricted to one table. It terminates (barring any of the events listed in the previous section) when the session issues a COMMIT or ROLLBACK command. The SAVEPOINT command can be used to set markers that will stage the action of a ROLLBACK, but the same transaction remains in progress irrespective of the use of SAVEPOINT

Explicit Transaction Control Statements

You can control the logic of transactions by using the COMMIT, SAVEPOINT, and ROLLBACK statements.

Note: You cannot COMMIT to a SAVEPOINT. SAVEPOINT is not ANSI-standard SQL.

Statement	Description
COMMIT	COMMIT ends the current transaction by making all data changes permanent.
SAVEPOINT <i>name</i>	SAVEPOINT <i>name</i> marks a savepoint within the current transaction.
ROLLBACK	ROLLBACK ends the current transaction by undoing all pending data changes.
ROLLBACK TO <i>SAVEPOINT name</i>	ROLLBACK TO SAVEPOINT rolls back the current transaction to the specified savepoint, then undoes all data changes and/or savepoints that were created after the savepoint to which you are rolling back. If you use the SAVEPOINT clause, the ROLLBACK statement undoes the entire transaction. Because savepoints are used as a way to list the savepoints that you have created,

#### QUESTION 291

Which statement is true regarding transactions? (Choose all that apply.)

- A. A transaction can consist only of a set of DML and DDL statements.
- B. A part or an entire transaction can be undone by using ROLLBACK command.
- C. A transaction consists of a set of DML or DCL statements.
- D. A part or an entire transaction can be made permanent with a COMMIT.
- E. A transaction can consist of only a set of queries or DML or DDL statements.

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 292

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

- A. Savepoints are effective only for COMMIT.
- B. Savepoints may be used to ROLLBACK.
- C. Savepoints can be used for only DML statements.
- D. Savepoints are effective for both COMMIT and ROLLBACK.
- E. Savepoints can be used for both DML and DDL statements.

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**