

T
R
O
Y
T
E
C
H
E
C
H
O
N
O
G
I
E
S



Oracle Database 11g : SQL
Fundamentals I Exam
Exam: 1Z0-051

Edition: 6.0

© 2013 - 2014 Test Killer, LTD All Rights Reserved

QUESTION: 1

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

Answer: A

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: 2

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

Answer: C

QUESTION: 3

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.

Answer: A

QUESTION: 4

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.

Answer: B, D

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: statement is not true statement is not true Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-9

QUESTION: 5

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.

Answer: C, D

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:

EMP_ID_SEQ sequence can be used to populate JOB_ID

EMP_ID_SEQ sequence will not be invalidated when column in EMPLOYEE_ID is modified. EMP_ID_SEQ sequence will be dropped automatically when you drop the EMPLOYEES table. 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: 6

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

Answer: B, E

QUESTION: 7

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.

Answer: C

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: 8

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);
```

Answer: A

QUESTION: 9

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;

Answer: C

Explanation:

The general syntax to create a synonym is: CREATE [PUBLIC] SYNONYM synonym FOR object;

QUESTION: 10

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
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, 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;

```

Answer: A

QUESTION: 11

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?

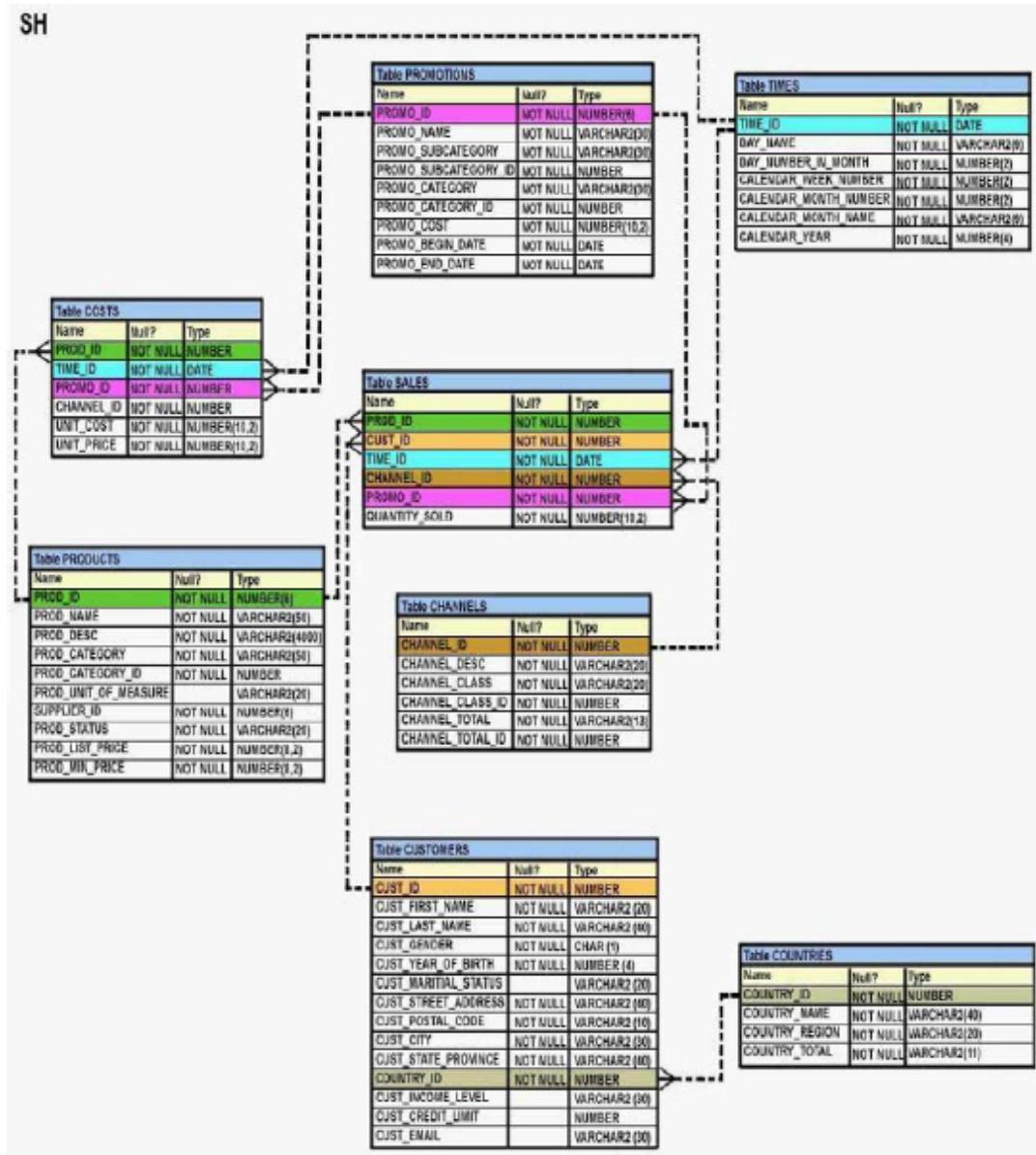
- A. It executes successfully and gives the required result.
- B. It executes successfully but does not give the required result.
- C. It produces an error because the join order of the tables is incorrect.
- D. It produces an error because equijoin and nonequijoin conditions cannot be used in the same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

Answer: B

QUESTION: 12

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

SH



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 );

```

Answer: C

QUESTION: 13

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';`

Answer: A

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: 14

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.

Answer: A

QUESTION: 15

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;

Answer: C

Explanation:

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: 16

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,0).promo_cost * 0.25, 100) "Discount" FROM promotions;
- B. SELECT promo_id. DECODE(promo_cost, 10000, DECODE(promo_category, 'GL', 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;

Answer: A, B

Explanation:

Note: there are some syntax issues in this question.

QUESTION: 17

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.

Answer: C

QUESTION: 18

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;

Answer: C, D

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: 19

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.

Answer: D

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: 20

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

Answer: C

QUESTION: 21

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.

Answer: D

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: 22

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(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 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';
```

Answer: A, C

QUESTION: 23

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);

Answer: B

QUESTION: 24

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"

Answer: A, C, E

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: 25

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);

Answer: A

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 punctuation 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: 26

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;

```

Answer: A, B

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: 27

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_iiname='Roberts' AND credit_limit=600). 1000);
- B. INSERT INTO orders (order_id.order_date.order_mode. (SELECT customer id FROM customers WHERE cust_last_iiname='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. o.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_iiname='Roberts' AND credit_limit=600). 1000);
- D. INSERT INTO orders (order_id.order_date.order_mode. (SELECT customer_id FROM customers WHERE cust_last_iiname='Roberts' AND credit_limit=600).order_total) VALUES(l.'10-mar-2007'\ 'direct'. &customer_id. 1000);

Answer: A

QUESTION: 28

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

Answer: D

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: 29

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.

Answer: A

QUESTION: 30

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

Answer: A

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: 31

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

Answer: D

QUESTION: 32

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

Answer: A, C, D

QUESTION: 33

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.

Answer: A

QUESTION: 34

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.

Answer: A, D

QUESTION: 35

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/Other1 category with minimum prices as either \$2000 or \$4000 and no unit of measure. You issue thej 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	V Box	Electronics	9000	
103	DVD-R Disc, 4.7 GB	Software/Other	2000	Nos.
104	Documentation Set - Spanish	Softwares/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.

Answer: A

QUESTION: 36

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;

Answer: B, C

QUESTION: 37

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;

Answer: B, C

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: 38

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 PRICELIST;

- 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;

Answer: C

QUESTION: 39

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

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

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.

Answer: D

QUESTION: 40

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.

Answer: C

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:
 $\text{COALESCE(expr1,expr2)} = \text{NVL(expr1,expr2)}$ $\text{COALESCE(expr1,expr2,expr3)} = \text{NVL(expr1,NVL(expr2,expr3))}$ 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: 41

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);

Answer: B. C

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: 42

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');

Answer: A. D

QUESTION: 43

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'fiiDdsph "of" Month. Year')) FROM DUAL;

Answer: C

QUESTION: 44

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.

Answer: B

QUESTION: 45

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 view TO manager;
GRANT manager ROLE TO BLAKE,CLARK;
- C. GRANT manager ROLE TO BLAKE,CLARK; GRANT create table, create view
TO BLAKE CLARK;

Answer: A

Explanation:

Result of commands:

The screenshot shows the PL/SQL Developer interface with the following details:

- Title Bar:** PL/SQL Developer - system@COMMONTTEST.PSV.COM.VN - [Command Window - New]
- Menu Bar:** File, Project, Edit, Session, Debug, Tools, Macro, Documents, Reports, Window, Help
- Toolbar:** Standard toolbar icons.
- Search Bar:** Search icon.
- Toolbox:** Database-related icons.
- Left Panel:**
 - All objects tree view showing nodes like AURORA\$ORB\$UNAUTHENTICATED, CMMN, CTXSYS, DBSNMP, MDSYS, ORDPLUGINS, ORDSYS, OUTLN, PFS, PUBLIC, and SYS.
 - Templates section with items: Constants, Default, DML statements, Error handling, SQL Window - SELECT TO_CHAR(TO_DATE("
- Central Area:**
 - Connected to Oracle8i Enterprise Edition Release 8.1.5.0.0
 - Connected as system
 - SQL prompt: SQL> create role Manager;
 - Output: Role created
 - SQL prompt: SQL> grant create table, create view to Manager;
 - Output: Grant succeeded
 - SQL prompt: SQL> grant Manager to PFS, OUTLN;
 - Output: Grant succeeded
 - SQL prompt: SQL>
- Bottom Status Bar:** ECH TRM FDB VER APR HDG TMB Grant succeeded in 0 seconds

QUESTION: 46

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.

Answer: C

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: 47

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_begiii_date > '01-JAN-01' ORDER BY 2 DESC;
- B. SELECT promo_name, promo_begiii_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;

Answer: A. D

QUESTION: 48

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.

Answer: A. C

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: 49

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

Answer: A, C, E

QUESTION: 50

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(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 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')

Answer: C

QUESTION: 51

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(10,2)

- 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(*).

Answer: C

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: 52

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);

Answer: D

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: 53

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 '\\'

Answer: B

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: 54

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.

Answer: A, B, E

QUESTION: 55

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.

Answer: B, E

QUESTION: 56

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.

Answer: B, D

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)

Use these to reinforce the exam concepts.

QUESTION: 57

Which constraint can be defined only at the column level?

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

Answer: B

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 The UNIQUE constraint enforces uniqueness on values in the constrained column. It can be defined not only at the column level. The CHECK constraint enforces that values added to the constrained column must be present in a static list of values permitted for the column. 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. 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: 58

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.

Answer: E

Explanation:

The GRANT command with the WITH GRANT OPTION clause allows the grantee to grant object privileges to other users and roles. Incorrect Answers The WITH GRANT OPTION does not allow a grantee DBA privileges. It is not required syntax for object privileges. It is optional clause of GRANT command. GRANT command does not allow privileges on columns of tables. 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: 59

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. &&

Answer: D

Explanation:

To reuse the variable without prompting the user each time you can use && substitution variable. Incorrect Answers This substitution variable will prompt the user each time. 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. 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: 60

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 at 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.

Answer: B

Explanation:

A change to the DEFAULT value affects only subsequent insertions to the table. Existing rows will not be affected. Incorrect Answers Column definitions can be altered to add DEFAULT values. Column definitions can be altered to add DEFAULT values. It works for columns with a NUMBER data type also. 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: 61

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.

Answer: A

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 There are no privileges for the user Scott at this point. SELECT privilege needs to be added to the user Scott. There are no privileges for the user Scott at this point. CONNECT privilege needs to be added to the user Scott. 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: 62

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;

Answer: D, E

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 The DELETE command is data-manipulation language (DML) command and it does not complete a transaction. 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 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: 63

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_INUMBER(4) NOT NULL CUST_NAMVARCHAR2(100)
NOT NULL CUST_ADDRESS VARCHAR2(150)
CUST_PHONVARCHAR2(20) Which SELECT statement accomplishes this task?

- A. SELECT*
- B. FROM customers;
- C. SELECT name, address
- D. FROM customers;
- E. SELECT id, name, address, phone
- F. FROM customers;
- G. SELECT cust_name, cust_address
- H. FROM customers;
- I. SELECT cust_id, cust_name, cust_address, cust_phone
- J. FROM customers;

Answer: D**Explanation:**

This answer provides correct list of columns for the output. Incorrect Answers 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. This answer does not provide correct list of columns for the output. There are not NAME and ADDRESS columns in the CUSTOMERS table. 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. 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: 64

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.

Answer: E

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 It is not a binary data. The ROWID datatype is a hexadecimal string. It is not a character data. The ROWID datatype is a hexadecimal string. It is not a raw binary data. The ROWID datatype is a hexadecimal string. 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: 65

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.

Answer: A, B, D, E

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 All views based on the DEPT table become invalid, but they are not deleted. 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. 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: 66

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 if it is based on tables that you created.

Answer: C

Explanation:

You can create a view in your own schema only if you are granted the CREATE VIEW privilege. Incorrect Answers You can create a view in your own schema only. You can create a view in your own schema only, not in any schema. There is no sequence view in Oracle. You cannot create a view that is accessible by everyone. You will need specially grant SELECT privileges on this view for everyone. 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: 67

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

- A. Constraint names must start with SYS_C.
- B. All constraints must be defined 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.

Answer: C, D

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 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. Not all constraints must be defines at the column level. Only NOT NULL constraint must be. 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: 68

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 if the query uses a HAVING clause.
- F. A HAVING clause CANNOT be used in sub queries.

Answer: B, D

Explanation:

B: WHERE clause cannot be used to restrict groups WHERE clause cannot be used when there is group functions. D: A HAVING clause can only be 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 used to restrict groups
 - C. A HAVING clause can only be used to restrict GROUPS.

E. WHERE clause cannot be used when there is group function, instead HAVING is to be used. 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: 69

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.

Answer: D

Explanation:

'=' is used in the statement and sub query will return more than one row. Employees table has 2 rows 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_it='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: 70

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';

Answer: C

Explanation:

In a WHERE clause, date and character 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 used in the statement.

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Producing Readable Output with iSQL*PLUS, p. 7-8

QUESTION: 71

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

Answer: C

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
- C. Does not meet truncate functions
- D. Does not meet round 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: 72

The CUSTOMERS table has these columns:

CUSTOMER_ID	NUMBER (4)	NOT NULL
CUSTOMER_NAME	VARCHAR2 (100)	NOT NULL
STREET_ADDRESS	VARCHAR2 (150)	
CITY_ADDRESS	VARHCAR2 (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)))

Answer: E

QUESTION: 73

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');

Answer: C

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 Oracle-01427: single-row sub query returns more than one row.
- B. Incorrect DELETE statement
- C. Incorrect DELETE statement

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Manipulating Data, p. 8-19

QUESTION: 74

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.

Answer: E

Explanation:

None of the statements produce an error. Incorrect answer: Statement will not cause error Statement will not cause error Statement will not cause error Statement will not cause error

QUESTION: 75

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.

Answer: C

Explanation:

You can remove existing rows from a table by using the DELETE statement. DELETE [FROM] table [WHENcondition]; Incorrect answer: Statement will not cause error Delete statement will not delete the table structure Statement will not cause error Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-19

QUESTION: 76

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);

Answer: A

Explanation:

The correct statement for MERGE is MERGE INTO table_name Incorrect answer: Wrong statement with the keyword EXISTS Wrong statement with the keyword EXISTS Wrong statement on the MERGE new_employees Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-29

QUESTION: 77

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';

Answer: A

Explanation:

INSTR is a character function return the numeric position of a named string.
 INSTR(NAMED,'a') Incorrect answer: Did not return a numeric position for 'a'. Did not return a numeric position for 'a'. Did not return a numeric position for 'a'. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 3-8

QUESTION: 78

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

Answer: B, D, E, G

Explanation:

Oracle constraint type is Not Null, Check, Primary Key, Foreign Key and Unique
 Incorrect answer: Is not Oracle constraint Is not Oracle constraint Is not Oracle constraint Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-3

QUESTION: 79

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;

Answer: D

Explanation:

DROP VIEW viewname; Incorrect answer: Not a valid drop view statement
 Refer: Introduction to Oracle9i: SQL,
 Oracle University Study Guide, 11-20

QUESTION: 80

Which is an iSQL*Plus command?

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

Answer: D**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:

INSERT is not a SQL*PLUS command UPDATE is not a SQL*PLUS command

SELECT is not a SQL*PLUS command DELETE is not a SQL*PLUS command

RENAME is not a SQL*PLUS command

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

QUESTION: 81

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';

Answer: D

QUESTION: 82

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

Answer: C, E

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: ROWID is not require GROUP BY clause is not require Must have inline view and outer query. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-23

QUESTION: 83

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;`

Answer: C

QUESTION: 84

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;

Answer: B

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: Multiplication and division has priority over addition and subtraction in Operator precedence. Give wrong results Wrong syntax Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 1-11

QUESTION: 85

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;

Answer: D

Explanation:

this is the correct solution for the answer. GRANT role_name to users; Incorrect answer: there is no such MODIFY keyword invalid CREATE command, there is no such NEW keyword invalid GRANT command, there is no such ROLE keyword invalid GRANT command, there is no such CHANGE keyword Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-10

QUESTION: 86

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);

Answer: D

Explanation:

ALTER TABLE table_nameADD [CONSTRAINT constraint] type (column);
 Incorrect answer: wrong syntax wrong syntax wrong syntax no such MODIFY keyword Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-17

QUESTION: 87

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;

Answer: F

Explanation:

sorted by highest to lowest is DESCENDING order Incorrect answer: result in ascending order wrong syntax with SORT keyword wrong syntax with SORT keyword default value for ORDER by is in ascending order wrong syntax with SORT keyword Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-22

QUESTION: 88

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.

Answer: C

Explanation:

sorted by highest to lowest is DESCENDING order Incorrect answer: grant the table privilege to PUBLIC login as sysoper login as DBA or sysdba no such option is allow. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 9-9

QUESTION: 89

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

Answer: B, C, D, F

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: is not single row attributes functions can accept more than one argument, e.g NVL2 Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 3-5

QUESTION: 90

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);

Answer: B

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: 91

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', '');

Answer: C, E, F

Explanation:

EMPLOYEE_ID is a primary key. Incorrect answer: EMPLOYEE_ID cannot be null
 EMPLOYEE_ID cannot be null mismatch of field_name with datatype Refer:
 Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-11

QUESTION: 92

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

Answer: C

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: the correct answer should be Alice, Reena and Timber the correct answer should be Alice, Reena and Timber the correct answer should be Alice, Reena and Timber Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-17

QUESTION: 93

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_';

Answer: A

Explanation:

ESCAPE identifier to search for the _ symbol Incorrect answer: ESCAPE identifier must be use wrong syntax wrong syntax Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-13

QUESTION: 94

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;

Answer: D

Explanation:

Concatenation operator to create a resultant column that is a character expression. Incorrect answer: no such dear customer column invalid syntax invalid syntax invalid syntax invalid syntax Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 1-18

QUESTION: 95

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.

Answer: A

QUESTION: 96

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

Answer: D

Explanation:

Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE. Incorrect answer: Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE Object privilege on VIEW is DELETE, INSERT, REFERENCES, SELECT and UPDATE Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-12

QUESTION: 97

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

Answer: A, F

Explanation:

Index exist for UNIQUE and PRIMARY KEY constraints Incorrect answer: ORDER_ID is neither UNIQUE nor PRIMARY KEY STATUS is neither UNIQUE nor PRIMARY KEY PROD_ID is neither UNIQUE nor PRIMARY KEY ORD_TOTAL is neither UNIQUE nor PRIMARY KEY Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-15

QUESTION: 98

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.

Answer: A

Explanation:

group functions on column ignore NULL values Incorrect answer: group functions on column ignore NULL values group functions on column ignore NULL values NVL function can be use for column with NULL values no such INC_NULLS keyword Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-12

QUESTION: 99

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;

Answer: C

Explanation:

For highest gpa value MAX function is needed, for result with per semester GROUP BY clause is needed Incorrect answer: per semester condition is not included result would not display the highest gpa value invalid syntax error invalid syntax error Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-7

QUESTION: 100

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

Answer: A, B, D, E

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: sub query cannot be used F: is incorrect. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 6-5

QUESTION: 101

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.

Answer: C

Explanation:

use the double-ampersand if you want to reuse the variable value without prompting the user each time. Incorrect answer: is not an error && will not prompt user for second time && will not ask the user for new value Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7-13

QUESTION: 102

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;

Answer: C

Explanation:

DATE value subtract DATE value will return numeric value. Incorrect answer: does not return numeric value does not return numeric value does not return numeric value Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 9-13

QUESTION: 103

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.

Answer: B, C

Explanation:

Create a synonyms when the names of the tables are too long or the table names are difficult. Incorrect answers: 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: 104

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.

Answer: A

QUESTION: 105

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;

Answer: B

Explanation:

function MAX(column_name) Incorrect answer: invalid statement inner query return more than one line column maxsal does not exists. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-7

QUESTION: 106

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%

Answer: B

Explanation:

WHERE lower(country_address)=’france’ Incorrect answer: invalid use of symbol “” invalid use of IS keyword invalid use of % in condition invalid use of condition Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-12

QUESTION: 107

Which are iSQL*Plus commands? (Choose all that apply.)

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

Answer: D

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:

INSERT is not a SQL*PLUS command UPDATE is not a SQL*PLUS command
 SELECT is not a SQL*PLUS command DELETE is not a SQL*PLUS command
 RENAME is not a SQL*PLUS command Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7

QUESTION: 108

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 Francisco');
- B. SELECT city_address, COUNT (*) FROM customers

WHERE city_address IN ('Los Angeles', 'San Francisco') GROUP BY city_address;
 C. SELECT city_address, COUNT(customer_id)
 FROM customers
 WHERE city_address IN ('Los Angeles', 'San Francisco') GROUP BY
 city_address, customer_id;
 D. SELECT city_address, COUNT(customer_id) FROM customers
 GROUP BY city_address IN ('Los Angeles', 'San Francisco');

Answer: B

Explanation:

Not C: The customer ID in the GROUP BY clause is wrong

QUESTION: 109

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

Answer: D

Explanation:

create a view regardless of whether or not the base tables exist. Incorrect answer: the option is not valid the option is not valid the option is not valid Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-3

QUESTION: 110

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

Answer: C

Explanation:

modifies the data but not the structure of a table

Incorrect answer:

DML does not complete a transaction

DDL modifies the structure and data in the table

DML does not modify table structure.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-3

QUESTION: 111

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

Answer: D

QUESTION: 112

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

Answer: A

Explanation:

when a user is created, by default no privilege is granted Incorrect answer: SELECT is not grant CONNECT is not grant default profile is grant by default not privilege.

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-6

QUESTION: 113

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.

Answer: B, E

Explanation:

GRANT ROLE to ROLE/USER Incorrect answer: Role can be grant to user Create table privilege is not granted Execute privilege is not granted Delete privilege is not granted Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-15

QUESTION: 114

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

Answer: D

Explanation:

a sub query that is part of the FROM clause of another query Incorrect answer: is not a schema object sub query can contain GROUP BY clause as well. does not necessary contains group functions Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-21

QUESTION: 115

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;

Answer: D

Explanation:

Equijoins are also called simple joins or inner joins. Equijoin involve primary key and foreign key. Incorrect answer: there is no join B invalid syntax does not involve the join in the primary and foreign key Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 4-8

QUESTION: 116

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.

Answer: D

Explanation:

the result is sort by price which is numeric and follow by product_name which is alphabetically. Incorrect answer: the results are sorted the results are sorted with alphabetically as well the results are sorted with numerically as well Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-2

QUESTION: 117

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.

Answer: C, E

Explanation:

You use an outer join to also see rows that do not meet the join condition. Incorrect answer: meet a join condition meet a join condition meet non join condition only does not take into consideration of primary key and foreign key relationship
Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 4-17

QUESTION: 118

Which constraint can be defined only at the column level?

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

Answer: B

Explanation:

the NOT NULL constraint can be specified only at the column level, not at the table level. Incorrect answer: UNIQUE can be define at table level CHECK can be define at table level PRIMARY KEY can be define at table level FOREIGN KEY can be define at table level Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-8 New Questions

QUESTION: 119

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;

Answer: A, D, E

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
	M1	Position of minus sign for negatives	99999M1	-3040	3040-
	FR	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 ⁿ times (n is the number of nines after V)	9999V99	3040	304000
	S	+ or - sign is prefixed	S999999	3040	+3040

QUESTION: 120

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.

Answer: C

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: 121

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

Answer: D, E

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: 122

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 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;

Answer: D

QUESTION: 123

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;`

Answer: C

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: 124

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

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?

- A. It generates an error because the alias is not valid.
- B. It executes successfully and gives the correct output.
- C. It executes successfully but does not give the correct output.
- D. It generates an error because the usage of the ROUND function in the expression is not valid.

- E. It generates an error because the concatenation operator can be used to combine only two items.

Answer: C

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: 125

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;

Answer: D

QUESTION: 126

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

Answer: A

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: 127

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.

Answer: B, C

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: 128

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.

Answer: B, C

QUESTION: 129

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.

Answer: E, F

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: 130

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.

Answer: B, D

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: 131

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.

Answer: B, D

QUESTION: 132

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.

Answer: B, E

QUESTION: 133

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.

Answer: B

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: 134

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.

Answer: B, D

QUESTION: 135

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.

Answer: B, D, F

Explanation:

Creating a Sequence (continued) CYCLE | NOCYCLE Specifies whether the sequence continues to generate values after reaching its maximum or minimum value (NO CYCLE 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: 136

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

Answer: A

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 (NO CYCLE 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: 137

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.

Answer: A, D

QUESTION: 138

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.

Answer: A

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: 139

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';

Answer: D

QUESTION: 140

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.

Answer: A

QUESTION: 141

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.

Answer: A

QUESTION: 142

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
 $\geq 2 * \text{prod_min_price}$ 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;

Answer: A, E

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: 143

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.

Answer: B**Explanation:**

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

QUESTION: 144

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

Answer: A, C, D

QUESTION: 145

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

Answer: B, D

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: 146

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

Answer: B

QUESTION: 147

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

Answer: B

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: 148

Examine the data in the CUST_NAME column of the CUSTOMERS table.
 CUST_NAME ----- Renske Ladwig Jason Mallin Samuel McCain
 Allan MCEwen Irene Mikkilineni 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%');

```

Answer: B

QUESTION: 149

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 ;

```

Answer: A, B

QUESTION: 150

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.

Answer: A

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: 151

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

Answer: D, E

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: 152

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.

Answer: A

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: 153

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;

Answer: C

Explanation:

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 quotes.

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: 154

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.

Question Exhibit	
PROMO_NAME	PROMO_END_DATE
post promotion #20-343	19-JUN-99
post promotion #20-274	16-JUL-99
TV promotion #12-530	23-APR-99
post promotion #17-157	29-JUN-99
TV promotion #12-481	05-JAN-00
newspaper promotion #19-4	16-AUG-99
everyday low price	01-JAN-99

OUTPUT	
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 23, 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;

Answer: C, D

QUESTION: 155

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;
```

Answer: A

QUESTION: 156

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;
```

Answer: B

QUESTION: 157

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.

Answer: C

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: 158

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;

Answer: B, C, D

QUESTION: 159

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;

Answer: D

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 incompatibility
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: 160

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)

Answer: B, D

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: 161

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;

Answer: B, D

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: 162

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.

Answer: D

QUESTION: 163

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.

Answer: B

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: 164

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.

Answer: A, C

QUESTION: 165

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.

Answer: C

QUESTION: 166

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

Answer: A, B

QUESTION: 167

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 =
d.task_id);
```

C.

```
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 =
d.task_id);
```

D.

```
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);
```

Answer: B

QUESTION: 168

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

Answer: A, E

QUESTION: 169

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);

Answer: C

QUESTION: 170

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

Answer: C

QUESTION: 171

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

Answer: C

QUESTION: 172

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

Answer: B, C

QUESTION: 173

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.

Answer: A, D

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 ($>$, $=$, \geq , $<$, \neq , \leq) 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: 174

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.

Answer: A, B, F

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: 175

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'.

Answer: C, E

QUESTION: 176

View the Exhibits and examine PRODUCTS and SALES tables.

The image shows two separate windows, each displaying a table structure. The top window is titled 'Exhibit_Products' and contains the 'PRODUCTS' table definition. The bottom window is titled 'Exhibit_Sales' and contains the 'SALES' table definition.

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)

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.

Answer: A

QUESTION: 177

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;
```

Answer: A

QUESTION: 178

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.

Answer: C

QUESTION: 179

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.

Answer: C

QUESTION: 180

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.

Answer: A

QUESTION: 181

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);

Answer: A

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: 182

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_date, promo_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.

Answer: D

QUESTION: 183

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;
```

Answer: A, C

QUESTION: 184

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.

Answer: A, C

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: 185

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.

Answer: B, D

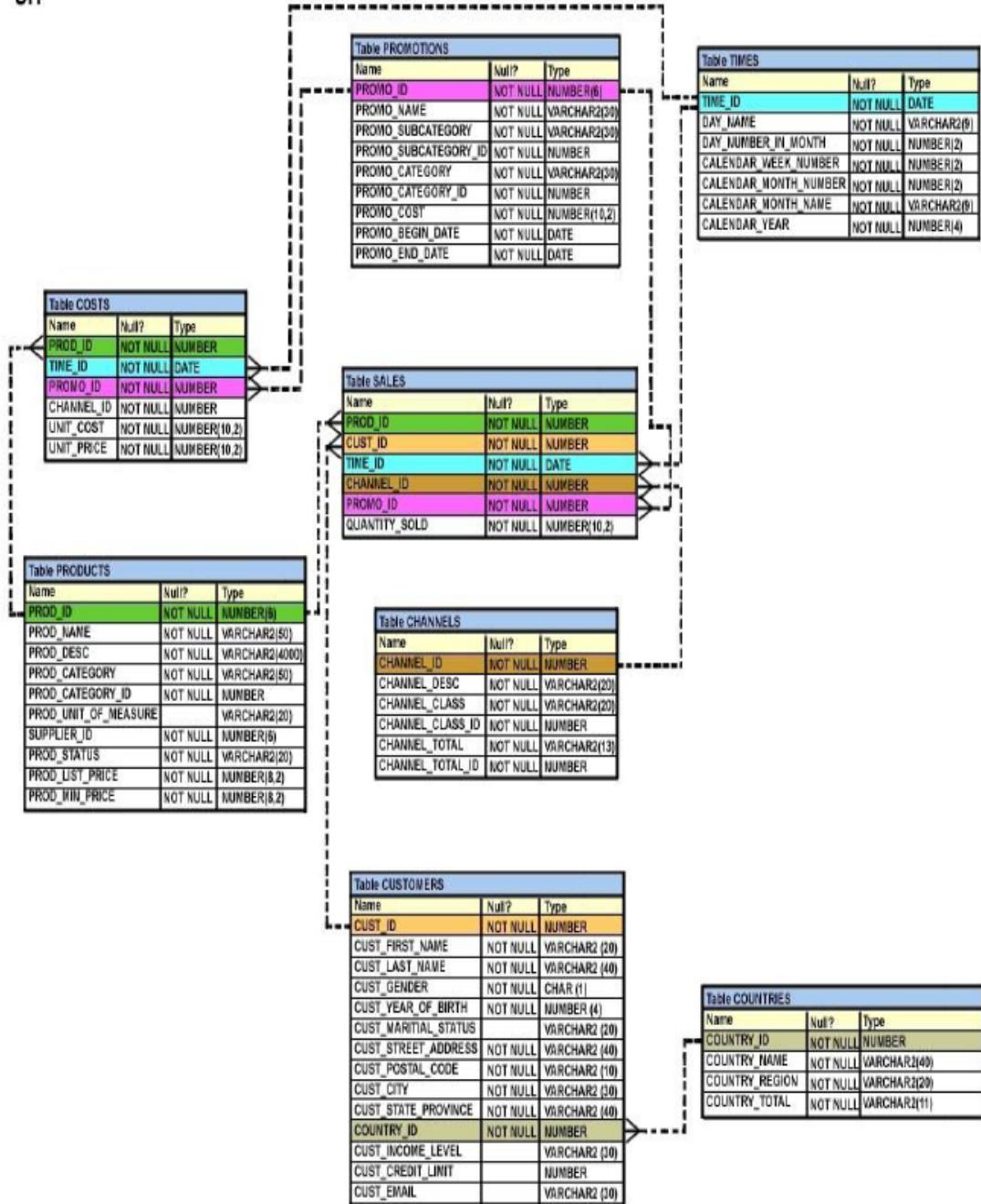
Explanation:

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: 186

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.

Answer: A

QUESTION: 187

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?

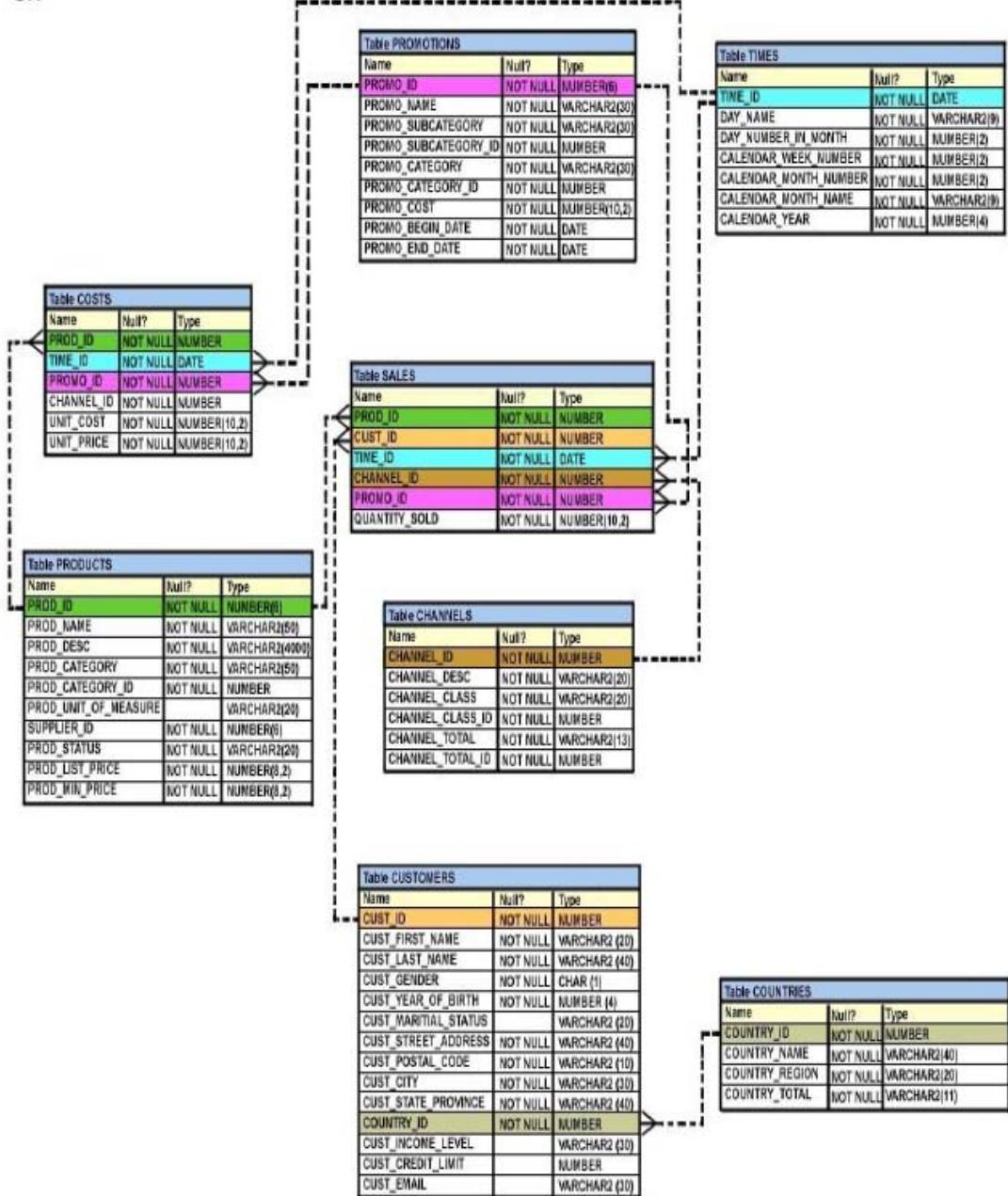
- A. It executes successfully.
- B. It gives an error because a complex view cannot be used to add data into the SALE_SUMMARY table.
- C. It gives an error because the column names in the subquery and the SALE_SUMMARY table do not match.
- D. It gives an error because the number of columns to be inserted does not match with the number of columns in the SALE_SUMMARY table.

Answer: D

QUESTION: 188

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.

Answer: C

Explanation:

One UPDATE statement can change rows in only one table, but it can change any number of rows in that table.

QUESTION: 189

View the Exhibit and examine the structures of the EMPLOYEES and DEPARTMENTS tables. You want to update the EMPLOYEES table as follows:
 -Update only those employees who work in Boston or Seattle (locations 2900 and 2700).

- Set department_id for these employees to the department_id corresponding to London (location_id 2100).
- Set the employees' salary in location_id 2100 to 1.1 times the average salary of their department.
- Set the employees' commission in location_id 2100 to 1.5 times the average commission of their department.

You issue the following command: SQL>UPDATE employees

```
SET department_id =
(SELECT department_id
FROM departments
WHERE location_id = 2100),
(salary, commission) =
(SELECT 1.1*AVG(salary), 1.5*AVG(commission) FROM employees, departments
WHERE departments.location_id IN(2900,2700,2100)) WHERE department_id IN
(SELECT department_id FROM departments WHERE location_id = 2900
OR location_id = 2700) What is 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, COMMISION) cannot be specified together in an UPDATE statement.

Answer: B

QUESTION: 190

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.

Answer: A, E

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 pending data changes permanent.
SAVEPOINT name	SAVEPOINT name marks a savepoint within the current transaction.
ROLLBACK	ROLLBACK ends the current transaction by discarding all pending data changes.
ROLLBACK TO SAVEPOINT name	ROLLBACK TO SAVEPOINT rolls back the current transaction to the specified savepoint, thereby discarding any changes and/or savepoints that were created after the savepoint to which you are rolling back. If you omit the TO SAVEPOINT clause, the ROLLBACK statement rolls back the entire transaction. Because savepoints are logical, there is no way to list the savepoints that you have created.

QUESTION: 191

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.

Answer: B, C

QUESTION: 192

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.

Answer: B, C