

1Z0-051 Dumps

Oracle Database: SQL Fundamentals I

<https://www.certleader.com/1Z0-051-dumps.html>



NEW QUESTION 1

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

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

Answer: AD

Explanation:

Gaps in the Sequence

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

Another event that can cause gaps in the sequence is a system crash. If the sequence caches values in memory, those values are lost if the system crashes.

Because sequences are not tied directly to tables, the same sequence can be used for multiple tables.

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

Modifying a Sequence

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

```
DROP SEQUENCE dept_deptid_seq;
```

NEW QUESTION 2

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.

NEW QUESTION 3

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

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

Answer: AB

Explanation:

Working with Dates The Oracle Database stores dates in an internal numeric format, representing the century, year, month, day, hours, minutes, and seconds. The default display and input format for any date is DD-MON-RR. RR Date Format The RR date format is similar to the YY element, but you can use it to specify different centuries. Use the RR date format element instead of YY so that the century of the return value varies according to the specified two digit year and the last two digits of the current year. The table in the slide summarizes the behavior of the RR element.

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

untitled Note the values shown in the last two rows of the above table. As we approach the middle of the century, then the RR behavior is probably not what you want. This data is stored internally as follows: CENTURY YEAR MONTH DAY HOUR MINUTE SECOND 19 87 06 17 17 10 43

NEW QUESTION 4

Evaluate the following SQL statements:

DELETE FROM sales;

There are no other uncommitted transactions on the SALES table.

Which statement is true about the DELETE statement?

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

Answer: C

NEW QUESTION 5

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:

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

NEW QUESTION 6

Examine the structure of the EMPLOYEES table:

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

Which UPDATE statement is valid?

- A. UPDATE employees SET first_name = 'John' SET last_name = 'Smith' WHERE employee_id = 180;
- B. UPDATE employees SET first_name = 'John', SET last_name = 'Smith' WHERE employee_id = 180;
- C. UPDATE employee SET first_name = 'John' AND last_name = 'Smith' WHERE employee_id = 180;
- D. UPDATE employee SET first_name = 'John', last_name = 'Smith' WHERE employee_id = 180;

Answer: D

NEW QUESTION 7

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

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

Answer: CDE

NEW QUESTION 8

The STUDENT_GRADES table has these columns:

STUDENT_IDNUMBER(12)
SEMESTER_ENDDATE
GPANUMBER(4,3)

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

Which statement accomplish this?

- A. SELECT student_id, semester_end, gpa FROM student_grades ORDER BY semester_end DESC, gpa DESC;
- B. SELECT student_id, semester_end, gpa FROM student_grades ORDER BY semester_end, gpa ASC
- C. SELECT student_id, semester_end, gpa FROM student_grades ORDER BY gpa DESC, semester_end ASC;
- D. SELECT student_id, semester_end, gpa FROM student_grades ORDER BY gpa DESC, semester_end DESC;
- E. SELECT student_id, semester_end, gpa FROM student_grades ORDER BY gpa DESC, semester_end ASC;
- F. SELECT student_id, semester_end, gpa FROM student_grades ORDER BY semester_end, gpa DESC

Answer: F

NEW QUESTION 9

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 punctuations marks as well as spaces. However, neither quoted nor nonquoted identifiers can contain double quotation marks or the null character (\0). Within a namespace, no two objects can have the same name. Nonquoted identifiers are not case sensitive. Oracle interprets them as uppercase. Quoted identifiers are case sensitive. By enclosing names in double quotation marks, you can give the following names to different objects in the same namespace: employees "employees" "Employees" "EMPLOYEES"

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

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

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

NEW QUESTION 10

View the Exhibit and examine the description for the CUSTOMERS table.

| Table CUSTOMERS | | |
|---------------------|----------|---------------|
| Name | Null? | Type |
| CUST_ID | NOT NULL | NUMBER |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2 (40) |
| CUST_GENDER | NOT NULL | CHAR (1) |
| CUST_YEAR_OF_BIRTH | NOT NULL | NUMBER (4) |
| CUST_MARITAL_STATUS | | VARCHAR2 (20) |
| CUST_STREET_ADDRESS | NOT NULL | VARCHAR2 (40) |
| CUST_POSTAL_CODE | NOT NULL | VARCHAR2 (10) |
| CUST_CITY | NOT NULL | VARCHAR2 (30) |
| CUST_STATE_PROVINCE | NOT NULL | VARCHAR2 (40) |
| COUNTRY_ID | NOT NULL | NUMBER |
| CUST_INCOME_LEVEL | | VARCHAR2 (30) |
| CUST_CREDIT_LIMIT | | NUMBER |
| CUST_EMAIL | | VARCHAR2 (30) |

You want to update the CUST_CREDIT_LIMIT column to NULL for all the customers, where CUST_INCOME_LEVEL has NULL in the CUSTOMERS table. Which SQL statement will accomplish the task?

- A. UPDATE customers SET cust_credit_limit = NULL WHERE CUST_INCOME_LEVEL = NULL;
- B. UPDATE customers SET cust_credit_limit = NULL WHERE cust_income_level IS NULL;
- C. UPDATE customers SET cust_credit_limit = TO_NUMBER(NULL) WHERE cust_income_level = TO_NUMBER(NULL);
- D. UPDATE customers SET cust_credit_limit = TO_NUMBER(' ',9999) WHERE cust_income_level IS NULL;

Answer: B

NEW QUESTION 10

Evaluate these two SQL statements:

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

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

What is true about them?

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

Answer: A

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

Incorrect Answer: B there is no syntax error. Results are sorted in ascending order by default. ORDER BY 2 will take the second column as sorting column. Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-22

NEW QUESTION 12

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.

NEW QUESTION 13

Exhibit contains the structure of PRODUCTS table:

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

Evaluate the following query:

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

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

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

Answer: B

NEW QUESTION 15

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?

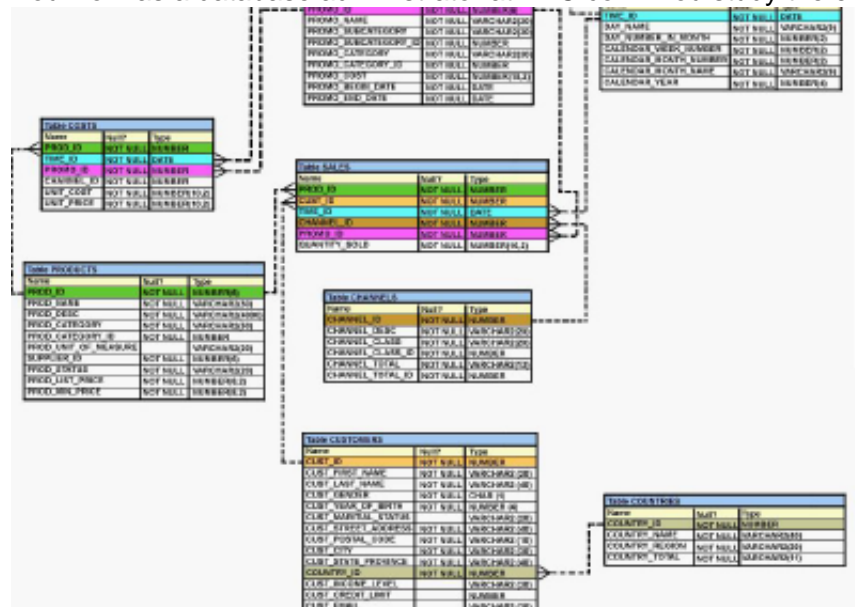
| Table CUSTOMERS | | |
|---------------------|----------|---------------|
| Name | Null? | Type |
| CUST_ID | NOT NULL | NUMBER |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2 (40) |
| CUST_GENDER | NOT NULL | CHAR (1) |
| CUST_YEAR_OF_BIRTH | NOT NULL | NUMBER (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 value
- D. It produces an error because the AND operator cannot be used to combine multiple BETWEEN clause

Answer: A

NEW QUESTION 16

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



Evaluate the following SQL statement: Exhibit:

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

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

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

Answer: A

NEW QUESTION 20

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

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

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

| Table CUSTOMERS | | |
|---------------------|----------|---------------|
| Name | Null? | Type |
| CUST_ID | NOT NULL | NUMBER |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2 (40) |
| CUST_GENDER | NOT NULL | CHAR (1) |
| CUST_YEAR_OF_BIRTH | NOT NULL | NUMBER (4) |
| CUST_MARITAL_STATUS | | VARCHAR2 (20) |
| CUST_STREET_ADDRESS | NOT NULL | VARCHAR2 (40) |
| CUST_POSTAL_CODE | NOT NULL | VARCHAR2 (10) |
| CUST_CITY | NOT NULL | VARCHAR2 (30) |
| CUST_STATE_PROVINCE | NOT NULL | VARCHAR2 (40) |
| COUNTRY_ID | NOT NULL | NUMBER |
| CUST_INCOME_LEVEL | | VARCHAR2 (30) |
| CUST_CREDIT_LIMIT | | NUMBER |
| CUST_EMAIL | | VARCHAR2 (30) |

You issue the following query:

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

Which statement is true regarding the outcome of this query?

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

Answer: B

Explanation:

Creating Joins with the USING Clause

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

The Natural JOIN USING Clause

The format of the syntax for the natural JOIN USING clause is as follows: SELECT table1.column, table2.column FROM table1 JOIN table2 USING (join_column1, join_column2...); While the pure natural join contains the NATURAL keyword in its syntax, the JOIN...USING syntax does not. An error is raised if the keywords NATURAL and USING occur in the same join clause. The JOIN...USING clause allows one or more equijoin columns to be explicitly specified in brackets after the USING keyword. This avoids the shortcomings associated with the pure natural join. Many situations demand that tables be joined only on certain columns, and this format caters to this requirement.

NEW QUESTION 23

Which one is a system privilege?

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

E. CREATE TABLE

Answer: E

NEW QUESTION 27

Which is a valid CREATE TABLE statement?

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

Answer: A

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

Incorrect Answer:

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

D Table name must begin with a letter.

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

NEW QUESTION 32

The CUSTOMERS table has the following structure: Exhibit:

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

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

*

Display the first name and tax amount of the customers. Tax is 5% of their credit limit

*

Only those customers whose income level has a value should be considered

*

Customers whose tax amount is null should not be considered

Which statement accomplishes all the required tasks?

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

Answer: B

NEW QUESTION 36

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

| 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 |

Exhibit:

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

- A. UPDATE promotions SET promo_cost = promo_cost+ 100 WHERE TO_CHAR(promo_end_date, 'yyyy') > '2000';
- B. SELECT promo_begin_date FROM promotions WHERE TO_CHAR(promo_begin_date,'mon dd yy')='jul 01 98';
- C. UPDATE promotions SET promo_cost = promo_cost+ 100 WHERE promo_end_date > TO_DATE(SUBSTR('01-JAN-2000',8));
- D. SELECT TO_CHAR(promo_begin_date,'dd/month') FROM promotions WHERE promo_begin_date IN (TO_DATE('JUN 01 98'), TO_DATE('JUL 01 98'));

Answer: AB

NEW 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. Winch two queries would achieve the required result? (Choose two.)

- A. SELECT cust_last_nam

B. cust_credit_limit FROM customers ORDER BY cust_credit_limit DESC:
C. SELECT cust_last_nam
D. cust_credit_limit FROM customers ORDER BY cust_credit_limit:
E. SELECT cust_last_nam
F. cust_credit_limit FROM customers ORDER BY cust_credit_limit NULLS LAST:
G. SELECT cust_last_nam
H. cust_credit_limit FROM customers ORDER BY cust_last_nam
I. cust_credit_limit NULLS LAST:

Answer: BC

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.

NEW QUESTION 38

You are currently located in Singapore and have connected to a remote database in Chicago.

You issue the following command:

Exhibit:

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

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

What is the outcome?

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

Answer: D

NEW QUESTION 39

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 constraint
- 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: BD

Explanation:

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

D: The NOT NULL constraint ensures that null values are not permitted for the column

Incorrect Answer: A statement is not true C statement is not true

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

NEW QUESTION 44

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

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

Answer: ABD

NEW QUESTION 46

Examine the structure of the EMPLOYEES and NEW_EMPLOYEES tables:

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

NEW_EMPLOYEES
EMPLOYEE_ID NUMBER Primary Key
NAME VARCHAR2(60)

Which MERGE statement is valid?

- A. MERGE INTO new_employees c USING employees e ON (c.employee_id = e.employee_id) WHEN MATCHED THEN UPDATE SET
- B. name = e.first_name || ',' || e.last_name WHEN NOT MATCHED THEN INSERT value S(e.employee_id, e.first_name || ', ' || e.last_name);
- C. MERGE new_employees c USING employees e ON (c.employee_id = e.employee_id) WHEN EXISTS THEN UPDATE SET
- D. name = e.first_name || ',' || e.last_name WHEN NOT MATCHED THEN INSERT valueS(e.employee_id, e.first_name || ', ' || e.last_name);
- E. MERGE INTO new_employees c USING employees e ON (c.employee_id = e.employee_id) WHEN EXISTS THEN UPDATE SET
- F. name = e.first_name || ',' || e.last_name WHEN NOT MATCHED THEN INSERT value S(e.employee_id, e.first_name || ', ' || e.last_name);
- G. MERGE new_employees c FROM employees e ON (c.employee_id = e.employee_id) WHEN MATCHED THEN UPDATE SET
- H. name = e.first_name || ',' || e.last_name WHEN NOT MATCHED THEN INSERT INTO new_employees valueS(e.employee_id, e.first_name || ', ' || e.last_name);

Answer: A

Explanation: Explanation: this is the correct MERGE statement syntax

Incorrect Answer: Bit should MERGE INTO table_name Cit should be WHEN MATCHED THEN Dit should MERGE INTO table_name Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 8-29

NEW QUESTION 51

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(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_name, promo_begin_date FROM promotions WHERE promo_begiii_date > '01-JAN-01' ORDER BY 2 DESC;
- B. SELECT promo_nam
- C. promo_begiii_date FROM promotions WHERE promo_begin_date > '01-JAN-01' ORDER BY promo_name DESC:
- D. SELECT promo_nam
- E. promo_begin_date FROM promotions WHERE promo_begin_date > '01-JAN-01' ORDER BY 1DESC:
- F. SELECT promo_name, promo_begin_date "START DATE" FROM promotions WHERE promo_begin_date > '01-JAN-01' ORDER BY "START DATE" DESC;

Answer: AD

NEW QUESTION 52

Examine the structure and data of the CUSTJTRANS table:

CUSTJRANS

Name Null? Type

CUSTNO NOT NULL CHAR(2) TRANSDATE DATE TRANSAMT NUMBER(6,2) CUSTNO TRANSDATE TRANSAMT

11 01-JAN-07 1000

22 01-FEB-07 2000

33 01-MAR-07 3000

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

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

Answer: ACD

NEW QUESTION 57

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

NEW QUESTION 58

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

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

SELECT * FROM ord;

A. CREATE SYNONYM ord FOR orders; This command is issued by O

B. CREATE PUBLIC SYNONYM ord FOR orders; This command is issued by O

C. CREATE SYNONYM ord FOR oe.orders; This command is issued by the database administrator

D. CREATE PUBLIC SYNONYM ord FOR oe.orders; This command is issued by the database administrator

Answer: D

Explanation:

Creating a Synonym for an Object To refer to a table that is owned by another user, you need to prefix the table name with the name of the user who created it, followed by a period. Creating a synonym eliminates the need to qualify the object name with the schema and provides you with an alternative name for a table, view, sequence, procedure, or other objects. This method can be especially useful with lengthy object names, such as views. In the syntax: PUBLIC Creates a synonym that is accessible to all users synonym Is the name of the synonym to be created object Identifies the object for which the synonym is created Guidelines The object cannot be contained in a package. A private synonym name must be distinct from all other objects that are owned by the same user. If you try to execute the following command (alternative B, issued by OE):

NEW QUESTION 59

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

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

B. Sub queries can contain GROUP BY and ORDER BY clauses

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

D. Main query and sub query can get data from different tables

E. Main query and sub query must get data from the same tables

F. Sub queries can contain ORDER BY but not the GROUP BY clause

Answer: ABD

NEW QUESTION 62

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

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

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

You issue the following SQL statements:

Exhibit:


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

What would be the outcome?

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

Answer: B

Explanation:

CASE Expression Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement: CASE expr WHEN comparison_expr1 THEN return_expr1 [WHEN comparison_expr2 THEN return_expr2 WHEN comparison_exprn THEN return_exprn ELSE else_expr] END

NEW QUESTION 63

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: AD

NEW QUESTION 65

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.

NEW QUESTION 67

Examine the structure of the INVOICE table: Exhibit:

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

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

- A. SELECT inv_no,NVL2(inv_date,'Pending','Incomplete') FROM invoice;

- B. SELECT inv_no,NVL2(inv_amt,inv_date,'Not Available') FROM invoice;
C. SELECT inv_no,NVL2(inv_date,sysdate-inv_date,sysdate) FROM invoice;
D. SELECT inv_no,NVL2(inv_amt,inv_amt*.25,'Not Available') FROM invoice;

Answer: AC

Explanation:

The NVL2 Function

The NVL2 function provides an enhancement to NVL but serves a very similar purpose. It evaluates whether a column or expression of any data type is null or not. 5-6 The NVL function\ If the first term is not null, the second parameter is returned, else the third parameter is returned. Recall that the NVL function is different since it returns the original term if it is not null. The NVL2 function takes three mandatory parameters. Its syntax is NVL2(original, ifnotnull, ifnull), where original represents the term being tested. Ifnotnull is returned if original is not null, and ifnull is returned if original is null. The data types of the ifnotnull and ifnull parameters must be compatible, and they cannot be of type LONG. They must either be of the same type, or it must be possible to convert ifnull to the type of the ifnotnull parameter. The data type returned by the NVL2 function is the same as that of the ifnotnull parameter.

NEW QUESTION 68

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

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

You need to display all promo categories that do not have 'discount' in their subcategory.

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

- A. SELECT promo_category FROM promotions MINUS SELECT promo_category FROM promotions WHERE promo_subcategory = 'discount';
B. SELECT promo_category FROM promotions INTERSECT SELECT promo_category FROM promotions WHERE promo_subcategory = 'discount';
C. SELECT promo_category FROM promotions MINUS SELECT promo_category FROM promotions WHERE promo_subcategory <> 'discount';
D. SELECT promo_category FROM promotions INTERSECT SELECT promo_category FROM promotions WHERE promo_subcategory <> 'discount';

Answer: AD

NEW QUESTION 72

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

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

Answer: AE

NEW QUESTION 76

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

- A. SELECT TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'fmDdspth "of" Month YYYY fmHH:MI:SS AM') NEW_DATE FROM dual;
B. SELECT TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'Ddspth "of" Month YYYY fmHH:MI:SS AM') NEW_DATE FROM dual;
C. SELECT TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'fmDdspth "of" Month YYYY HH:MI:SS AM') NEW_DATE FROM dual;
D. SELECT TO_CHAR(TO_DATE('19-Mar-2001', 'DD-Mon-YYYY'), 'fmDdspth "of" Month YYYYfmtHH:HI:SS AM') NEW_DATE FROM dual;

Answer: A

NEW QUESTION 80

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

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

Answer: AD

NEW QUESTION 82

Which statement is true regarding sub queries?

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

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.

NEW QUESTION 84

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

CUSTOMERS

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

GRADES

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

You need to display names and grades of customers who have the highest credit limit.

Which two SQL statements would accomplish the task? (Choose two.)

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

Answer: BC

NEW QUESTION 88

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_i
- B. DECODE(NVL(promo_cost.O), promo_cost * 0.25, 100) "Discount" FROM promotions;
- C. SELECT promo i
- D. DECODE(promo_cos
- E. 10000. DECODE(promo_categor
- F. 'G\ promo_cost * 25. NULL). NULL) "Catcost" FROM promotions;
- G. SELECT promo_i
- H. DECODE(NULLIF(promo_cos
- I. 10000). NUL
- J. promo_cost*.25, *N/A') "Catcost" FROM promotions;

K. SELECT promo_i
L. DECODE(promo_cos
M. >10000. 'High'. <10000. 'Low') "Range"FROM promotions;

Answer: AB

Explanation:

Note: there are some syntax issues in this question.

NEW QUESTION 89

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;

NEW QUESTION 91

Examine the structure of the STUDENTS table:

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

You need to create a report of the 10 students who achieved the highest ranking in the course INT SQL and who completed the course in the year 1999. Which SQL statement accomplishes this task?

- A. SELECT student_id, marks, ROWNUM "Rank" FROM students WHERE ROWNUM <= 10 AND finish_date BETWEEN '01-JAN-99' AND '31-DEC-99' AND course_id = 'INT_SQL' ORDER BY marks DESC;
- B. SELECT student_id, marks, ROWID "Rank" FROM students WHERE ROWID <= 10 AND finish_date BETWEEN '01-JAN-99' AND '31-DEC-99' AND course_id = 'INT_SQL' ORDER BY marks;
- C. SELECT student_id, marks, ROWNUM "Rank" FROM (SELECT student_id, marks FROM students WHERE ROWNUM <= 10 AND finish_date BETWEEN '01-JAN-99' AND '31-DEC-99' AND course_id = 'INT_SQL' ORDER BY marks DESC);
- D. SELECT student_id, marks, ROWNUM "Rank" FROM (SELECT student_id, marks FROM students WHERE (finish_date BETWEEN '01-JAN-99' AND '31-DEC-99' AND course_id = 'INT_SQL' ORDER BY marks DESC) WHERE ROWNUM <= 10 ;
- E. SELECT student_id, marks, ROWNUM "Rank" FROM (SELECT student_id, marks FROM students ORDER BY marks) WHERE ROWNUM <= 10 AND finish date BETWEEN '01-JAN-99' AND '31-DEC-99' AND course_id = 'INT_SQL';

Answer: D

NEW QUESTION 96

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

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

Answer: BD

Explanation:

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

NEW QUESTION 100

A SELECT statement can be used to perform these three functions:
Choose rows from a table.

Choose columns from a table
Bring together data that is stored in different tables by creating a link between them.
Which set of keywords describes these capabilities?

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

Answer: B

Explanation: Explanation: choose rows from a table is SELECTION,
Choose column from a table is PROJECTION
Bring together data in different table by creating a link between them is JOIN.
Incorrect Answer:
Aanswer should have SELECTION, PROJECTION and JOIN.
Canswer should have SELECTION, PROJECTION and JOIN.
Danswer should have SELECTION, PROJECTION and JOIN.
Eanswer should have SELECTION, PROJECTION and JOIN.
Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 1-6

NEW QUESTION 104

Examine the structure of the EMPLOYEES table:

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

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

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

Which INSERT statement meets the above requirements?

- A. INSERT INTO employees VALUES (emp_id_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- B. INSERT INTO employees VALUES (emp_id_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did IN (20,50));
- C. INSERT INTO (SELECT * FROM employees WHERE department_id IN (20,50)) VALUES (emp_id_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- D. INSERT INTO (SELECT * FROM employees WHERE department_id IN (20,50) WITH CHECK OPTION) VALUES (emp_id_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);
- E. INSERT INTO (SELECT * FROM employees WHERE (department_id = 20 AND department_id = 50) WITH CHECK OPTION) VALUES (emp_id_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did);

Answer: D

NEW QUESTION 107

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

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

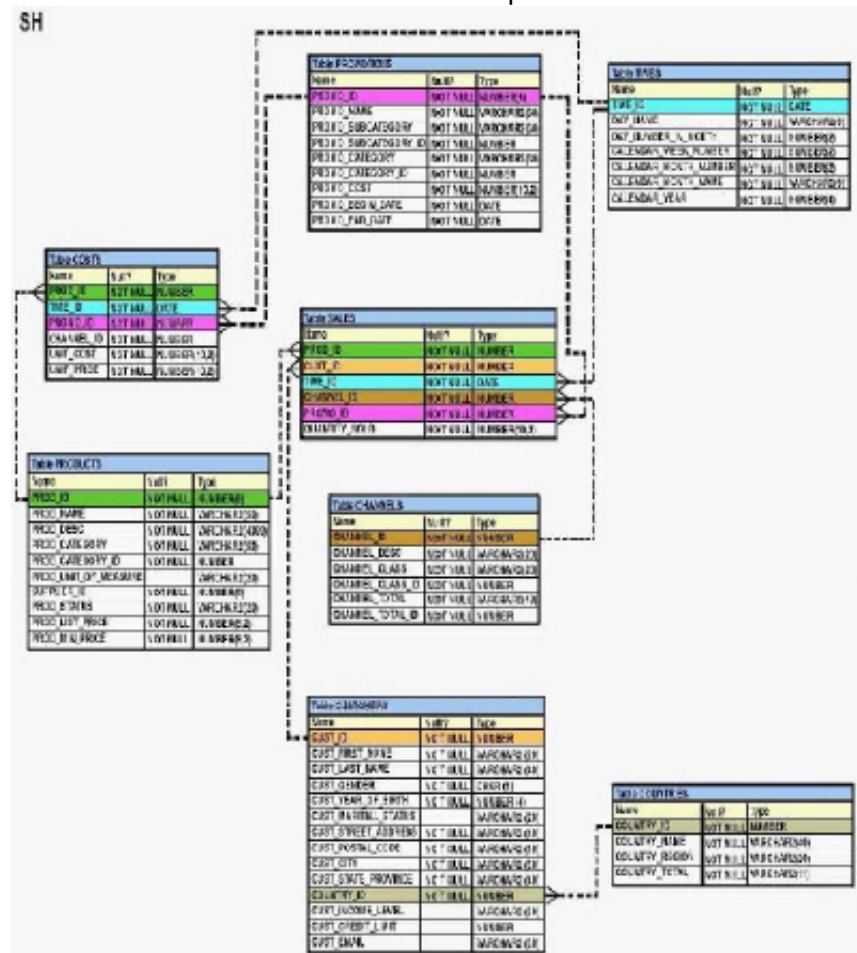
Answer: ABCF

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.

NEW QUESTION 109

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



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

- A. DELETE FROM products WHERE prod_id = (SELECT prod_id FROM sales WHERE time_id - 3*365 = SYSDATE);
- B. DELETE FROM products WHERE prod_id = (SELECT prod_id FROM sales WHERE SYSDATE >= time_id - 3*365);
- C. DELETE FROM products WHERE prod_id IN (SELECT prod_id FROM sales WHERE SYSDATE - 3*365 >= time_id);
- D. DELETE FROM products WHERE prod_id IN (SELECT prod_id FROM sales WHERE time_id >= SYSDATE - 3*365);

Answer: C

NEW QUESTION 110

Which CREATE TABLE statement is valid?

- A. CREATE TABLE ord_details (ord_no NUMBER(2) PRIMARY KEY, item_no NUMBER(3) PRIMARY KEY, ord_date DATE NOT NULL);
- B. CREATE TABLE ord_details (ord_no NUMBER(2) UNIQUE, NOT NULL, item_no NUMBER(3), ord_date DATE DEFAULT SYSDATE NOT NULL);
- C. CREATE TABLE ord_details (ord_no NUMBER(2) , item_no NUMBER(3), ord_date DATE DEFAULT NOT NULL, CONSTRAINT ord_uq UNIQUE (ord_no), CONSTRAINT ord_pk PRIMARY KEY (ord_no));
- D. CREATE TABLE ord_details (ord_no NUMBER(2), item_no NUMBER(3), ord_date DATE DEFAULT SYSDATE NOT NULL, CONSTRAINT ord_pk PRIMARY KEY (ord_no, item_no));

Answer: D

Explanation:

PRIMARY KEY Constraint

A PRIMARY KEY constraint creates a primary key for the table. Only one primary key can be created for each table. The PRIMARY KEY constraint is a column or a set of columns that uniquely identifies each row in a table. This constraint enforces the uniqueness of the column or column combination and ensures that no column that is part of the primary key can contain a null value. Note: Because uniqueness is part of the primary key constraint definition, the Oracle server enforces the uniqueness by implicitly creating a unique index on the primary key column or columns.

NEW QUESTION 113

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 row
- 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

NEW QUESTION 114

Which statement correctly describes SQL and /SQL*Plus?

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

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

Answer: A

NEW QUESTION 116

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

| Table CUSTOMERS | | |
|---------------------|----------|---------------|
| Name | Null? | Type |
| CUST_ID | NOT NULL | NUMBER |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2 (20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2 (40) |
| CUST_GENDER | NOT NULL | CHAR (1) |
| CUST_YEAR_OF_BIRTH | NOT NULL | NUMBER (4) |
| CUST_MARITAL_STATUS | | VARCHAR2 (20) |
| CUST_STREET_ADDRESS | NOT NULL | VARCHAR2 (40) |
| CUST_POSTAL_CODE | NOT NULL | VARCHAR2 (10) |
| CUST_CITY | NOT NULL | VARCHAR2 (30) |
| CUST_STATE_PROVINCE | NOT NULL | VARCHAR2 (40) |
| COUNTRY_ID | NOT NULL | NUMBER |
| CUST_INCOME_LEVEL | | VARCHAR2 (30) |
| CUST_CREDIT_LIMIT | | NUMBER |
| CUST_EMAIL | | VARCHAR2 (30) |

Using the CUSTOMERS table, you need to generate a report that shows the average credit limit for customers in WASHINGTON and NEW YORK. Which SQL statement would produce the required result?

- A. SELECT cust_city, AVG(cust_credit_limit) FROM customers WHERE cust_city IN ('WASHINGTON','NEW YORK') GROUP BY cust_credit_limit, cust_city;
B. SELECT cust_city, AVG(cust_credit_limit) FROM customers WHERE cust_city IN ('WASHINGTON','NEW YORK') GROUP BY cust_city, cust_credit_limit;
C. SELECT cust_city, AVG(cust_credit_limit) FROM customers WHERE cust_city IN ('WASHINGTON','NEW YORK') GROUP BY cust_city;
D. SELECT cust_city, AVG(NVL(cust_credit_limit,0)) FROM customers WHERE cust_city IN ('WASHINGTON','NEW YORK');

Answer: C

Explanation:

Creating Groups of Data: GROUP BY Clause Syntax You can use the GROUP BY clause to divide the rows in a table into groups. You can then use the group functions to return summary information for each group. In the syntax: group_by_expression Specifies the columns whose values determine the basis for grouping rows Guidelines

If you include a group function in a SELECT clause, you cannot select individual results as well, unless the individual column appears in the GROUP BY clause. You receive an error message if you fail to include the column list in the GROUP BY clause.

Using a WHERE clause, you can exclude rows before dividing them into groups.

You must include the columns in the GROUP BY clause.

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

NEW QUESTION 121

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

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

Answer: BCD

NEW QUESTION 126

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

| EMP_ID | DEPT_ID | COMMISSION |
|--------|---------|------------|
| 1 | 10 | 500 |
| 2 | 20 | 1000 |
| 3 | 10 | |
| 4 | 10 | 600 |
| 5 | 30 | 800 |
| 6 | 30 | 200 |
| 7 | 10 | |
| 8 | 20 | 3000 |

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

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

Answer: AC

NEW QUESTION 127

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

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

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

You need to find those customers who have never changed their address.

Which SET operator would you use to get the required output?

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

Answer: C

NEW QUESTION 129

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 row
- 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:

COALESCE(expr1,expr2) = NVL(expr1,expr2) COALESCE(expr1,expr2,expr3) = NVL(expr1,NVL(expr2,expr3)) The data type COALESCE returns if a not null value is found is the same as that of the first not null parameter. To avoid an "ORA-00932: inconsistent data types" error, all not null parameters must have data types compatible with the first not null parameter.

NEW QUESTION 132

Examine the structure and data in the PRICE_LIST table:

PROD_ID NOT NULL NUMBER(3)

PROD_PRICE VARCHAR2(10)

PROD_ID PROD PRICE

100 \$234.55

101 \$6,509.75

102 \$1,234

in the same format as the PROD_PRICE. Which SQL statement would give the required result?

- A. SELECT TO_CHAR(prod_price* .25,'\$99.999.99') FROM PRICE_LIST;
- B. SELECT TO_CHAR(TO_NUMBER(prod_price)* .25,'\$99.999.00') FROM PRICE_LIST;
- C. SELECT TO_CHAR(TO_NUMBER(prod_price,'\$99.999.99')* .25,'\$99.999.00') FROM PRICE_LIST;
- D. SELECT TO_NUMBER(TO_NUMBER(prod_price,'\$99.999.99')* .25/'\$99.999.00') FROM PRICE_LIST;

Answer: C

NEW QUESTION 133

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

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

Answer: ACD

NEW QUESTION 137

Evaluate the following SQL statements: Exhibit:

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

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

Exhibit:

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

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

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

Answer: BD

NEW QUESTION 140

Evaluate the following SQL statements: Exhibit:

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

Exhibit:

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

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

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

Answer: B

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),

NEW QUESTION 141

Examine the structure of the EMPLOYEES and NEW_EMPLOYEES tables:

EMPLOYEES

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

NEW_EMPLOYEES

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

Which DELETE statement is valid?

- A. DELETE FROM employees WHERE employee_id = (SELECT employee_id FROM employees);
- B. DELETE * FROM employees WHERE employee_id=(SELECT employee_id FROM new_employees);
- C. DELETE FROM employees WHERE employee_id IN (SELECT employee_id FROM new_employees WHERE name = 'Carrey');
- D. DELETE * FROM employees WHERE employee_id IN (SELECT employee_id FROM new_employees WHERE name = 'Carrey');

Answer: C

NEW QUESTION 144

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

NEW QUESTION 149

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 successful

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

NEW QUESTION 150

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

You want to display only those product names with their list prices where the list price is at least double the minimum price. The report should start with the product

name having the maximum list price satisfying this

condition.

Evaluate the following SQL statement:

SQL>SELECT prod_name,prod_list_price FROM products WHERE prod_list_price >= 2 * prod_min_price

Which ORDER BY clauses can be added to the above SQL statement to get the correct output?

(Choose all that apply.)

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

A. ORDER BY prod_list_price DESC, prod_name;

B. ORDER BY (2*prod_min_price)DESC, prod_name;

C. ORDER BY prod_name, (2*prod_min_price)DESC;

D. ORDER BY prod_name DESC, prod_list_price DESC;

E. ORDER BY prod_list_price DESC, prod_name DESC;

Answer: AE

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.

NEW QUESTION 155

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

- A. A simple view in which column aliases have been used cannot be update
- 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: BD

NEW QUESTION 156

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.

NEW QUESTION 157

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: AC

NEW QUESTION 159

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: AC

NEW QUESTION 164

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: AF

Explanation: Index exist for UNIQUE and PRIMARY KEY constraints

Incorrect Answer: BORDER_ID is neither UNIQUE nor PRIMARY KEY CSTATUS is neither UNIQUE nor PRIMARY KEY DPROD_ID is neither UNIQUE nor PRIMARY KEY EORD_TOTAL is neither UNIQUE nor PRIMARY KEY

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 10-15

NEW QUESTION 169

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: ADE

NEW QUESTION 172

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: BDEG

Explanation:

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

NEW QUESTION 175

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.

NEW QUESTION 180

View the Exhibits and examine PRODUCTS and SALES 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) |

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

NEW QUESTION 182

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: BCDF

Explanation:

manipulate data items, act on each row returned, return one result per row, and accept arguments that can be a column or expression.

Incorrect Answer: A is not single row attributes E functions can accept more than one argument, e.g NVL2

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 3-5

NEW QUESTION 186

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: BE

Explanation:

GRANT ROLE to ROLE/USER

Incorrect Answer: A A role can be granted to a user C Create table privilege is not granted D Execute privilege is not granted F Delete privilege is not granted

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-15

NEW QUESTION 189

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 statement
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: AD

Explanation:

Using a Subquery to Solve a Problem Suppose you want to write a query to find out who earns a salary greater than Abel's salary. To solve this problem, you need two queries: one to find how much Abel earns, and a second query to find who earns more than that amount. You can solve this problem by combining the two queries, placing one query inside the other query. The inner query (or subquery) returns a value that is used by the outer query (or main query). Using a subquery is equivalent to performing two sequential queries and using the result of the first query as the search value in the second query. Subquery Syntax A subquery is a SELECT statement that is embedded in the clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries. They can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself. You can place the subquery in a number of SQL clauses, including the following: WHERE clause HAVING clause FROM clause In the syntax: operator includes a comparison condition such as >, =, or IN Note: Comparison conditions fall into two classes: single-row operators (>, =, >=, <, <>, <=) and multiple-row operators (IN, ANY, ALL, EXISTS). The subquery is often referred to as a nested SELECT, sub-SELECT, or inner SELECT statement. The subquery generally executes first, and its output is used to complete the query condition for the main (or outer) query. Guidelines for Using Subqueries Enclose subqueries in parentheses. Place subqueries on the right side of the comparison condition for readability. (However, the subquery can appear on either side of the comparison operator.) Use single-row operators with single-row subqueries and multiple-row operators with multiple-row subqueries. Subqueries can be nested to an unlimited depth in a FROM clause but to "only" 255 levels in a WHERE clause. They can be used in the SELECT list and in the FROM, WHERE, and HAVING clauses of a query.

NEW QUESTION 191

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

NEW QUESTION 193

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 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

NEW QUESTION 196

Examine the structure of the EMPLOYEES and DEPARTMENTS tables:

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: A there is no join B invalid syntax C does not involve the join in the primary and foreign key

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 4-8

NEW QUESTION 199

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

NEW QUESTION 201

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 value
- B. A change to the DEFAULT value affects only subsequent insertions to the table
- C. Column definitions cannot be altered to add DEFAULT values for columns with a NUMBER data type
- D. All the rows that have a NULL value for the SALARY column will be updated with the value 5000.

Answer: B

Explanation:

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

Incorrect Answers

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

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

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

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

NEW QUESTION 206

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.

NEW QUESTION 207

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 successful
- 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 constraint

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 column-level 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.

NEW QUESTION 211

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: BD

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.

NEW QUESTION 212

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 row
- 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

NEW QUESTION 217

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.

NEW QUESTION 218

You are granted the CREATE VIEW privilege. What does this allow you to do?

- A. Create a table vie
- B. Create a view in any schem
- C. Create a view in your schem
- D. Create a sequence view in any schem
- E. Create a view that is accessible by everyon
- F. Create a view only of it is based on tables that you create

Answer: C

Explanation:

You can create a view in your own schema only if you are granted the CREATE VIEW privilege.

Incorrect Answers

A:You can create a view in your own schema only.

B:You can create a view in your own schema only, not in any schema.

D:There is no sequence view in Oracle.

E:You cannot create a view that is accessible by everyone. You will need specially grant SELECT privileges on this view for everyone.

F:You can create a view in your own schema, but not only for tables in your schema. You can use object from other users schemas if you have privileges to retrieve data from them.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 292-301

Chapter 7: Creating Other Database Objects in Oracle

NEW QUESTION 219

Examine the description of the CUSTOMERS table:

| | | |
|------------------|---------------|----------|
| CUSTOMER_ID | NUMBER(4) | NOT NULL |
| CUSTOMER_NAME | VARCHAR2(100) | NOT NULL |
| STREET_ADDRESS | VARCHAR2(150) | |
| CITY_ADDRESS | VARCHAR2(50) | |
| STATE_ADDRESS | VARCHAR2(50) | |
| PROVINCE_ADDRESS | VARCHAR2(50) | |
| COUNTRY_ADDRESS | VARCHAR2(50) | |
| POSTAL_CODE | VARCHAR2(12) | |
| CUSTOMER_PHONE | VARCHAR2(20) | |

The CUSTOMER_ID column is the primary key for the table.

Which statement returns the city address and the number of customers in the cities Los Angeles or San Francisco?

- A. SELECT city_address, COUNT(*) FROM customers WHERE city_address IN ('Los Angeles', 'San Fransisco');
- B. SELECT city_address, COUNT (*) FROMcustomers WHERE city address IN ('Los Angeles', 'San Fransisco') GROUP BY city_address;
- C. SELECT city_address, COUNT(customer_id) FROMcustomers WHERE city_address IN ('Los Angeles', 'San Fransisco') GROUP BYcity_address, customer_id;
- D. SELECT city_address, COUNT (customer_id) FROM customers GROUP BY city_address IN ('Los Angeles', 'San Fransisco');

Answer: B

Explanation:

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

NEW QUESTION 223

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: A is not an error

B&& will not prompt user for second time D&& will not ask the user for new value

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 7-13

NEW QUESTION 227

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: A result in ascending order B wrong syntax with SORT keyword C wrong syntax with SORT keyword D default value for ORDER BY is in ascending order E wrong syntax with SORT keyword

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-22

NEW QUESTION 231

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

NEW QUESTION 236

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: A the option is not valid B the option is not valid C the option is not valid

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-3

NEW QUESTION 240

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

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

NEW QUESTION 242

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: AB

NEW QUESTION 247

Which is an iSQL*Plus command?

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

Answer: D

Explanation: Explanation: The only SQL*Plus command in this list : DESCRIBE. It cannot be used as SQL command. This command returns a description of table name, including all columns in that table, the datatype for each column and an indication of whether the column permits storage of NULL values. Incorrect Answer: A INSERT is not a SQL*PLUS command B UPDATE is not a SQL*PLUS command C SELECT is not a SQL*PLUS command E DELETE is not a SQL*PLUS command F RENAME is not a SQL*PLUS command

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

NEW QUESTION 251

The DBA issues this SQL command:

CREATE USER Scott

IDENTIFIED by tiger;

What privileges does the user Scott have at this point?

- A. No privilege
B. Only the SELECT privileg
C. Only the CONNECT privileg
D. All the privileges of a default use

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

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

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

D:There is no default user in Oracle.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 348-351

Chapter 8: User Access in Oracle

NEW QUESTION 253

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
- D. Does not meet truncate functions
- E. Does not meet round functions
- F. Does not meet round functions

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Single-Row functions, p. 3-13

NEW QUESTION 256

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

EMPLOYEES

| ENAME | HIREDATE | SAL | COMM |
|--------|-----------|------|------|
| SMITH | 17-DEC-00 | 800 | |
| ALLEN | 20-FEB-99 | 1600 | 300 |
| WARD | 22-FEB-95 | 1250 | 500 |
| JONES | 02-APR-98 | 2975 | |
| MARTIN | 28-SEP-99 | 1250 | 1400 |
| BLAKE | 01-MAY-97 | 2850 | |

You want to generate a report showing the total compensation paid to each employee to date.

You issue the following query:

```
SQL>SELECT ename 'joined on ' hiredate
',the total compensation paid is '
TO_CHAR(ROUND(ROUND(SYSDATE-hiredate)/365) * sal + comm)
"COMPENSATION UNTIL DATE"
FROM employees;
```

What is the outcome?

- A. It generates an error because the alias is not vali
- B. It executes successfully and gives the correct outpu
- C. It executes successfully but does not give the correct outpu
- D. It generates an error because the usage of the ROUND function in the expression is not vali
- E. It generates an error because the concatenation operator can be used to combine only two item

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.)

NEW QUESTION 261

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:

A is not a schema object

B sub query can contain GROUP BY clause as well.

C does not necessary contains group functions

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 11-21

NEW QUESTION 265

Which two statements are true about WHERE and HAVING clauses? (Choose two)

- A. A WHERE clause can be used to restrict both rows and group
- B. A WHERE clause can be used to restrict rows onl
- C. A HAVING clause can be used to restrict both rows and group
- D. A HAVING clause can be used to restrict groups onl
- E. A WHERE clause CANNOT be used in a query of the query uses a HAVING claus
- F. A HAVING clause CANNOT be used in sub querie

Answer: ABDE

Explanation: B: WHERE clause cannot be use to restrict groups

WHERE clause cannot be use when there is group functions.

D: A HAVING clause can only e used to restrict GROUPS.

Note: HAVING clause to specify which groups are to be displayed and thus further restrict the groups on the basis of aggregate information. The Oracle server performs the following steps when you use the Having clause

1.
rows are grouped
2.
the group function is applied to the group
3.
the group that match the criteria in the Having clause are displayed.

Incorrect Answers :

- A. Where clause cannot be use to restrict groups
- C. A HAVING clause can only e used to restrict GROUPS.
- E. WHERE clause cannot be use when there is group function, instead HAVING is to be use.
- F. There is no constraint to use HAVING clause in a sub queries.

Refer: Introduction to Oracle9i: SQL, Oracle University Student Guide, Aggregating Data using Group Functions, p. 5-20

NEW QUESTION 267

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 vali
- 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 claus

Answer: D

NEW QUESTION 271

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

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

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

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

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

Chapter 4: Sub queries

NEW QUESTION 272

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: BESCAPE identifier must be use Cwrong syntax Dwrong syntax

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 2-13

NEW QUESTION 277

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: Athe correct answer should be Alice, Reena and Timber Bthe correct answer should be Alice, Reena and Timber Dthe correct answer should be Alice, Reena and Timber

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 13-17

NEW QUESTION 280

What is true of using group functions on columns that contain NULL values?

- A. Group functions on columns ignore NULL value
- B. Group functions on columns returning dates include NULL value
- C. Group functions on columns returning numbers include NULL value
- D. Group functions on columns cannot be accurately used on columns that contain NULL value
- E. Group functions on columns include NULL values in calculations if you use the keyword INC_NULL

Answer: A

Explanation: group functions on column ignore NULL values

Incorrect Answer: Bgroup functions on column ignore NULL values Cgroup functions on column ignore NULL values DNVL function can be use for column with NULL values Eno such INC_NULLS keyword

Refer: Introduction to Oracle9i: SQL, Oracle University Study Guide, 5-12

NEW QUESTION 285

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

| 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) |

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 successfull
 B. It produces an error because the condition on the CUST_CITY column is not vali
 C. It produces an error because the condition on the CUST_FIRST_NAME column is not vali
 D. It produces an error because conditions on the CUST_CREDIT_LIMIT column are not vali

Answer: A

NEW QUESTION 288

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 Smit
 E. The statement fails because there is more than one row matching the IT_ADMIN job ID in the EMPLOYEES tabl
 F. The statement fails because there is no 'Administration' department in the DEPARTMENTS tabl

Answer: D

Explanation:

'=' is use in the statement and sub query will return more than one row.

Employees table has 2 row matching the employee name Smith.

The update statement will fail.

Incorrect Answers :

A. The Update statement will fail no update was done.

B. The update statement will fail no update was done.

C. The update statement will fail no update was done.

E. The update statement will fail but not due to job_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

NEW QUESTION 290

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:

ADML does not complete a transaction

BDDL modifies the structure and data in the table

DDML does not modified table structure.

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

NEW QUESTION 294

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 forma
- B. The CUST_CREDIT_VALUE column would allow storage of positive and negative integer
- C. The TRANS_VALIDITY column would allow storage of a time interval in days, hours, minutes, and second
- D. The CUST_STATUS column would allow storage of data up to the maximum VARCHAR2 size of 4,000 character

Answer: BD

Explanation:

B: The NUMBER datatype stores fixed and floating-point numbers. Numbers of virtually any magnitude can be stored and are guaranteed portable among different systems operating Oracle, up to 38 digits of precision.

The following numbers can be stored in a NUMBER column:

Positive numbers in the range 1×10^{-130} to $9.99...9 \times 10^{125}$ with up to 38 significant digits Negative numbers from -1×10^{-130} to $9.99...99 \times 10^{125}$ with up to 38 significant digits Zero Positive and negative infinity (generated only by importing from an Oracle Version 5 database)

D: The VARCHAR2 datatype stores variable-length character strings. When you create a table with a VARCHAR2 column, you specify a maximum string length (in bytes or characters) between 1 and 4000 bytes for the VARCHAR2 column. An interval literal specifies a period of time, and Oracle supports two types of interval literals: YEAR_TO_MONTH and DAY TO SECOND. For DAY TO SECOND, you can specify these differences in terms in terms of days, hours, minutes, and seconds. DAY TO SECOND contains a leading field and may contain an optional trailing field. If trailing field is specified it must be less significant than the leading field. For example, INTERVAL MINUTE TO DAY is not valid.

A DAY TO MINUTE interval considers an interval of days to the nearest minute. Reference: Oracle Database Concepts 10g, Native Datatypes

NEW QUESTION 296

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

- A. A table name can be of any lengt
- B. A table can have any number of column
- C. A column that has a DEFAULT value cannot store null value
- D. A table and a view can have the same name in the same schem
- E. A table and a synonym can have the same name in the same schem
- F. The same table name can be used in different schemas in the same databas

Answer: EF

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.

NEW QUESTION 298

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: BD

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:

NEW QUESTION 300

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

NEW QUESTION 303

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

- A. Constraint names must start with SYS_
- B. All constraints must be defines at the column leve
- C. Constraints can be created after the table is create
- D. Constraints can be created at the same time the table is create
- E. Information about constraints is found in the VIEW_CONSTRAINTS dictionary vie

Answer: CD

Explanation:

Constraints can be created after the table is created. Use ALTER TABLE command for that.

Constraints can be created at the same time the table is created (CREATE TABLE command).

Incorrect Answers

A:There is no requirements in Oracle that constraint names must start with SYS_C. Oracle can use prefix “SYS” to build indexes for UNIQUE and NOT NULL constraints, but it is not required for user to follow this naming rule.

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

E:There is no VIEW_CONSTRAINTS dictionary view in Oracle.

OCP Introduction to Oracle 9i: SQL Exam Guide, Jason Couchman, p. 227-232

Chapter 5: Creating Oracle Database Objects

NEW QUESTION 308

View the Exhibit; e xamine the structure of the PROMOTIONS table.

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

Each promotion has a duration of at least seven days.

Your manager has asked you to generate a report, which provides the weekly cost for each promotion done to I date.

Which query would achieve the required result?

- A. SELECT promo_name, promo_cost/promo_end_date-promo_begin_date/7 FROM promotions;
- B. SELECT promo_name,(promo_cost/promo_end_date-promo_begin_date)/7 FROM promotions;
- C. SELECT promo_name, promo_cost/(promo_end_date-promo_begin_date/7) FROM promotions;
- D. SELECT promo_name, promo_cost/((promo_end_date-promo_begin_date)/7) FROM promotions;

Answer: D

NEW QUESTION 311

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your 1Z0-051 Exam with Our Prep Materials Via below:

<https://www.certleader.com/1Z0-051-dumps.html>