1. (A)

Which SELECT statement should you use if you want to display unique combinations of the POSITION and MANAGER values from the EMPLOYEE table?

SELECT DISTINCT position, manager FROM employee;

SELECT position, manager DISTINCT FROM employee;

SELECT position, manager FROM employee;

SELECT position, DISTINCT manager FROM employee;

Explanation:

To display a unique values in the result you can use the DISTINCT key word this will eliminate the duplicate values from the result of the query.

Question: 2. (A)

Which two are attributes of /SQL\*Plus? (Choose two)

/SQL\*Plus commands cannot be abbreviated.

/SQL\*Plus commands are accesses from a browser.

/SQL\*Plus commands are used to manipulate data in tables.

/SQL\*Plus commands manipulate table definitions in the database.

/SQL\*Plus is the Oracle proprietary interface for executing SQL statements.

Explanation:

SQL\*Plus commands can be used to manipulate data in tables and SQL\*Plus commands manipulate table definitions in the database.

Question: 3. (A)

You need to perform certain data manipulation operations through a view called EMP\_DEPT\_VU, which you previously created.

You want to look at the definition of the view (the SELECT statement on which the view was create.)

How do you obtain the definition of the view?

Use the DESCRIBE command in the EMP\_DEPT VU view.

Use the DEFINE VIEW command on the EMP\_DEPT VU view.

Use the DESCRIBE VIEW command on the EMP\_DEPT VU view.

Query the USER\_VIEWS data dictionary view to search for the EMP\_DEPT\_VU view.

Query the USER\_SOURCE data dictionary view to search for the EMP\_DEPT\_VU view.

Query the USER\_OBJECTS data dictionary view to search for the EMP\_DEPT\_VU view.

Explanation:

To look on the view definition you need to query the USER\_VIEWS data dictionary view and search for the EMP\_DEPT\_VU view.

Question: 4. (A)

Which is an /SQL\*Plus command?

INSERT

UPDATE

SELECT

DESCRIBE

DELETE

RENAME

Explanation:

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

Question: 5. (A)

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

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

Which SELECT statement accomplishes this task?

SELECT\*

FROM customers;

SELECT name, address FROM customers;

SELECT id, name, address, phone FROM customers;

SELECT cust\_name, cust\_address FROM customers;

SELECT cust\_id, cust\_name, cust\_address, cust\_phone FROM customers;.

Explanation:

This answer provides correct list of columns for the output.

Question: 6. (A)

Evaluate this SQL statement:

SELECT e.EMPLOYEE\_ID,e.LAST\_NAME,e.DEPARTMENT\_ID, d.DEPARTMENT\_NAME. FROM EMP e, DEPARTMENT d

WHERE e.DEPARTMENT\_ID = d.DEPARTMENT\_ID;

In the statement, which capabilities of a SELECT statement are performed?

Selection, projection, join

Difference, projection, join

Selection, intersection, join

Intersection, projection, join

Difference, projection, product

Question: 7. (A)

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

The sort is in ascending by order by default.

The sort is in descending order by default.

The ORDER BY clause must precede the WHERE clause.

The ORDER BY clause is executed on the client side.

The ORDER BY clause comes last in the SELECT statement.

The ORDER BY clause is executed first in the query execution.

Question: 8. (A)

From SQL\*Plus, you issue this SELECT statement: SELECT\*

From orders;

You use this statement to retrieve data from a data table for \_\_\_\_\_\_\_\_\_\_. (Choose all that apply)

Updating

Viewing

Deleting

Inserting

Truncating

Explanation:

You can use SELECT statement to display and to insert data into different table.

Question: 9. (A)

Evaluate this SQL\*Plus command:

START delaccount

Which task will this command accomplish?

It executes the DELACCOUNT PL/SQL routine.

It runs the DELACCOUNT.SQL script file.

It creates the DELACCOUNT file using the default file extension.

It invokes the editor to edit the contents of the DELACCOUNT file.

Explanation:

START is SQL\*Plus command which is used to run the already stored script. It will not display the actual command in the script.

Question: 10. (A)

Which SQL SELECT statement performs a projection, a selection, and join when executed?

SELECT p.id\_number, m.manufacturer\_id, m.city FROM product p, manufacturer m

WHERE p.manufacturer\_id = m.manufacturer\_id AND m.manufacturer\_id = 'NF10032';

SELECT id\_number, manufacturer\_id FROM product

ORDER BY manufacturer\_id, id\_number;

SELECT id\_number, manufacturer\_id FROM product

WHERE manufacturer\_id = 'NF10032';

SELECT manufacturer\_id, city FROM manufacturer

AND manufacturer\_id = 'NF10032' ORDER BY city;

Explanation:

PROJECTION will select the whole column values of the table while SELECTION will gives you rows of the table and JOIN is joining the two tables on a same column. To get all these task done in one statement you can use this command

SELECT p.id\_number, m.manufacturer\_id, m.city

FROM product p, manufacturer m

WHERE p.manufacturer\_id = m.manufacturer\_id

AND m.manufacturer\_id = 'NF10032';

Question: 11. (A)

The CUSTOMERS table has these columns:

|  |  |  |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(4) | NOT NULL |
| CUSTOMER\_NAME | VARCHAR2(100) | NOT NULL |
| STREET\_ADDRESS | VARCHAR2(150) |  |
| CITY\_ADDRESS | VARCHAR2(50) |  |
| STATE\_ADDRESS | VARCHAR2(50) |  |
| PROVINCE\_ADDRESS | VARCHAR2(50) |  |
| COUNTRY\_ADDRESS | VARCHAR2(50) |  |
| POSTAL\_CODE | VARCHAR2(12) |  |
| CUSTOMER\_PHONE | VARCHAR2(20) |  |

Which statement finds the rows in the CUSTOMERS table that do not have a postal code?

SELECT customer\_id, customer\_name FROM customers

WHERE postal\_code CONTAINS NULL;

SELECT customer\_id, customer\_name FROM customers

WHERE postal\_code = '\_\_\_\_\_\_\_\_';

SELECT customer\_id, customer\_name FROM customers

WHERE postal\_code IS NULL;

SELECT customer\_id, customer\_name FROM customers

WHERE postal code IS NVL;

SELECT customer\_id, customer\_name

FROM customers

WHERE postal\_code = NULL;

Explanation:

This statement returns the rows in the CUSTOMERS table that do not have a postal code. The correct syntax to check NULL values is usage of “IS NULL” clause.

Question: 12. (A)

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?

The two statements produce identical results.

The second statement returns a syntax error.

There is no need to specify DESC because the results are sorted in descending order by default.

The two statements can be made to produce identical results by adding a column alias for the salary column in the second SQL statement.

Explanation:

These two statements produce identical results, because it is possible even to use numbers to indicate the column position where Oracle should order the output from a statement.

Question: 13. (A)

Evaluate the set of SQL statements:

CREATE TABLE dept (deptno NUMBER(2), dname VARCNAR2(14), loc VARCNAR2(13)); ROLLBACK; DESCRIBE DEPT

What is true about the set?

The DESCRIBE DEPT statement displays the structure of the DEPT table.

The ROLLBACK statement frees the storage space occupies by the DEPT table.

The DESCRIBE DEPT statement returns an error ORA-04043: object DEPT does not exist.

The DESCRIBE DEPT statement displays the structure of the DEPT table only if the us a COMMIT statement introduced before the ROLLBACK statement..

Explanation:

The structure of the DEPT table will be displayed because the CREATE TABLE statement is DDL operation and it cannot be rolled back because implicit *commit* occurs on the database when a user exits SQL\*Plus or issues a data-definition language (DDL) command such as a *create table*

statement, user to create a database object, or an *alter table* statement, used to alter a database object.

Question: 14. (A)

Examine the data of the EMPLOYEES table.

EMPLOYEES (EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | HR\_MGR | 5000 |
| 106 | Bryan | 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 |

Evaluate this SQL statement:

SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary, m.employee\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m

WHERE e.mgr\_id = m.employee\_id AND e.salary > 4000;

What is its output?

A.

EMP\_id EMPLOYEE SALARY Mgr\_id Manager

------- ---------- --------- ------------- --------------

110 Bob 8000 Bob

120 Ravi 6500 110 Ravi

108 Jennifer 6500 110 Jennifer

103 Chris 4200 120 Chris

105 Diana 5000 108 Diana

B.

EMP\_id EMPLOYEE SALARY Mgr\_id Manager

------- ---------- --------- ------------- --------------

120 Ravi 6500 110 Bob

108 Jennifer 6500 110 Bob

103 Chris 4200 120 Ravi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 105 | Diana 500 108 Jennifer | |  |  |
| C. |  |  |  |  |
| EMP\_id EMPLOYEE SALARY Mgr\_id Manager | | |  |  |
| ------- ---------- --------- ------------- -------------- | | |  |  |
| 110 Bob 800 | |  |  |  |
| 120 | Ravi 6500 110 Bob | |  |  |
| 108 | Jennifer 6500 110 Bob | |  |  |
| 103 | Chris 4200 120 Ravi | |  |  |
| 105 | Diana 5000 108 Jennifer | |  |  |
| D |  |  |  |  |
| EMP\_id EMPLOYEE SALARY Mgr\_id Manager | | |  |  |
| ------- ---------- --------- ------------- -------------- | | |  |  |
| 110 Bob 8000 110 Bob | | |  |  |
| 120 | Ravi 6500 120 Ravi | |  |  |
| 108 | Jennifer 6500 108 Jennifer | |  |  |
| 103 | Chris 4200 103 Chris | |  |  |
| 105 | Diana 5000 105 Dina | |  |  |

E. The SQL statement produces an error.

Explanation:

This statement lists the ID, name, and salary of the employee, and the ID and name of the employee's manager, for all the employees who have a manager and earn more than 4000

Question: 15. (A)

Which /SQL\*Plus feature can be used to replace values in the WHERE clause?

Substitution variables

Replacement variables

Prompt variables

Instead-of variables

This feature cannot be implemented through /SQL\*Plus.

Explanation:

Lexical substitution variables can be used to replace values in the WHERE clause.

Question: 16. (A)

You are formulating queries in a SQL\*Plus. Which of the following statement correctly describes how to specify a column alias?

Place the alias at the beginning of the statement to describe the table.

Place the alias after each column separated by a space to describe the column.

Place the alias after each column separated by a comma to describe the column.

Place the alias at the end of the statement to describe the table.

Explanation:

Aliases do not describe the tables they describe columns so the alias should be place at the end of each column and separated by a space to describe the column.

Question: 17. (A)

You want to use a function in you column clause of a SQL statement. The NVL function accomplishes which of the following tasks?

Assists in the distribution of output across multiple columns.

Enables you to specify alternate output for non-NULL column values.

Enables you to specify alternated out for NULL column values.

Nullifies the value of the column out put.

Explanation:

NVL function is simple if\_then operation that test column values out to see whether it is NULL and if it find it is null then NVL substitutes the specified default value for the NULL value.

Question: 18. (A)

You want to use SQL\*Plus to connect to the oracle database. Which of the following choices does not indicate a component you must specify when logging into the oracle?

The SQL\*Plus Keyword.

The username

The password.

The database name.

Explanation:

When connecting to the database you don’t need to specify the name of the database and when you are not specifying the name of the database then you will be connected to the local database.

Question: 19. (A)

The EMPLOYEE\_HISTORY table contains these columns:

EMPLOYEE\_ID NUMBER

LAST\_NAME VARCHAR2(25)

FIRST\_NAME VARCHAR2(25)

DEPARTMENT\_ID NUMBER

POSITION VARCHAR2(30)

SALARY NUMBER(6,2)

HIRE\_DATE DATE

DEPART\_DATE DATE

The EMPLOYEE\_HISTORY table contains only former employees.

You need to create a report to display all former employees that were hired on or after January 1, 1996. The data should display in this format:

Former Employee Term of Employment

---------------------------- ----------------------------------

14837 - SMITH 10-MAY-92 / 01-JUN-01

Which SELECT statement could you use?

SELECT employee\_id||' - '||last\_name AS Former Employee, hire\_date||' / '||depart\_date AS Term of Employment

FROM employee\_history WHERE hire\_date > '31-DEC-95';

SELECT employee\_id||' - '||last\_name "AS Former Employee", hire\_date||' / '||depart\_date "AS Term of Employment"

FROM employee\_history

WHERE hire\_date > '31-DEC-95';

C. SELECT employee\_id||' - '||last\_name 'Former Employee', hire\_date||' / '||depart\_date 'Term of Employment'

FROM employee\_history WHERE hire\_date > '31-DEC-95' AND depart\_date > NULL;

D. SELECT employee\_id||' - '||last\_name "Former Employee", hire\_date||' / '||depart\_date "Term of Employment"

FROM employee\_history WHERE hire\_date > '31-DEC-95' AND depart\_date <> NULL;

E. SELECT employee\_id||' - '||last\_name "Former Employee", hire\_date||' / '||depart\_date "Term of Employment"

FROM employee\_history WHERE hire\_date > '31-DEC-95' AND depart\_date IS NOT NULL;

Question: 20. (A)

The EMPloyee table contains these columns: Empno Number(4)

Ename Varchar2(10) job varchar2(10) sal Varchar2(10)

You need to display the employees information by using this query.

How many columns are presented after executing this query:

SELECT Empno||','||Ename||','||Job "Employee Information" FROM employee;

1

2

3

0

4

.

Explanation

When we used Concatenation operator between diferent no of columns so then one Column as a resultant column is to be apperead.

Question: 21.(B)

Examine the data of the EMPLOYEES table.

EMPLOYEES (EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | HR\_MGR | 5000 |
| 106 | Bryan | 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 |

Which statement lists the ID, name, and salary of the employee, and the ID and name of the employee's manager, for all the employees who have a manager and earn more than 4000?

SELECT employee\_id "Emp\_id", emp\_name "Employee", salary,

employee\_id "Mgr\_id", emp\_name "Manager" FROM employees

WHERE salary > 4000;

SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary,

m.employee\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m

WHERE e.mgr\_id = m.mgr\_id AND e.salary > 4000;

SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary,

m.employee\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m

WHERE e.mgr\_id = m.employee\_id AND e.salary > 4000;

SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary,

m.mgr\_id "Mgr\_id", m.emp\_name "manager" FROM employees e, employees m

WHERE e.mgr\_id = m.employee\_id AND e.salary > 4000;

SELECT e.employee\_id "Emp\_id", e.emp\_name "Employee", e.salary,

m.mgr\_id "Mgr\_id", m.emp\_name "Manager" FROM employees e, employees m

WHERE e.employee\_id = m.employee\_id AND e.salary > 4000;.

Explanation:

This statement lists the ID, name, and salary of the employee, and the ID and name of the employee's manager, for all the employees who have a manager and earn more than 4000

Question: 22. (B)

You need to display the last names of those employees who have the letter “A” as the second character in their names.

Which SQL statement displays the required results?

SELECT last\_name FROM EMP

WHERE last\_name LIKE ‘\_A%’;

SELECT last\_name FROM EMP

WHERE last name =’\*A%’

SELECT last\_name FROM EMP

WHERE last name =’\_A%’;

SELECT last\_name FROM EMP

WHERE last name LIKE ‘\*A%’

Explanation:

Statement in this answer will show correct results because usage of operator LIKE and format mask ‘\_A%’ extract the last names of those employees who have the letter “A” as the second character in their names. Symbol ‘\_’ in format mask substitute exactly one symbol and cannot be NULL.

Question: 23. (B)

In which scenario would TOP N analysis be the best solution?

You want to identify the most senior employee in the company.

You want to find the manager supervising the largest number of employees.

You want to identify the person who makes the highest salary for all employees.

You want to rank the top three sales representatives who have sold the maximum number of products.

Explanation:

If you want to rank the top three sales representatives who have sold the maximum number of products TOP-N query will be the best solution. TON-N queries use inline views and are handy for displaying a short list of table data, based on “greatest” or “least” criteria.

Question: 24. (B)

Evaluate this SQL statement:

SELECT c.customer\_id, o.order\_id, o.order\_date, p.product\_name

FROM customer c, curr\_order o, product p

WHERE customer.customer\_id = curr\_order.customer\_id

AND o.product\_id = p.product\_id

ORDER BY o.order\_amount;

This statement fails when executed. Which change will correct the problem?

Include the ORDER\_AMOUNT column in the SELECT list.

Use the table name in the ORDER BY clause.

Remove the table aliases from the WHERE clause.

Use the table aliases instead of the table names in the WHERE clause.

Remove the table alias from the ORDER BY clause and use only the column name.

Explanation:

When an alias is define for a table name in join then you cannot use the table name instead of alias in the FROM clause while using alias in the SELECT list. An alias should be used in the WHERE clause also.

Question: 25. (B)

You want to display the titles of books that meet these criteria:

Purchased before January 21, 2001

Price is less then $500 or greater than $900

You want to sort the results by their data of purchase, starting with the most recently bought book.

Which statement should you use?

A. SELECT book\_title

FROM books

WHERE price between 500 and 900

AND purchase\_date < ’21-JAN-2001’

ORDER BY purchase\_date;

WHERE price IN (500,900)

AND purchase\_date < ’21-JAN-2001’ ORDER BY purchase date ASC;

SELECT book\_title

FROM books

WHERE price < 500 or > 900

AND purchase\_date < ’21-JAN-2001’

ORDER BY purchase date DESC;

D. SELECT book\_title

FROM books

WHERE (price < 500 OR price > 900)

AND purchase\_date < ’21-JAN-2001’

ORDER BY purchase date DESC;

Explanation:

This statement provides required results.

Question: 26. (B)

For which task would you use the WHERE clause in a SELECT statement?

to designate the ORDER table location

to compare PRODUCT\_ID values to 7382

to display only unique PRODUCT\_ID values

to restrict the rows returned by a GROUP BY clause

Explanation:

You can use the WHERE clause in the SELECT statement to implement the condition on the statement by comparing values.

Question: 27. (B)

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

The registrar has requested a report listing the students' grade point averages (GPA), sorted from highest grade point average to lowest within each semester, starting from the earliest date. Which statement accomplishes this?

SELECT student\_id, semester\_end, gpa FROM student\_grades

ORDER BY semester\_end DESC, gpa DESC;

SELECT student\_id, semester\_end, gpa FROM student\_grades

ORDER BY semester\_end ASC, gpa ASC;

SELECT student\_id, semester\_end, gpa FROM student\_grades

ORDER BY semester\_end, gpa DESC;

SELECT student\_id, semester\_end, gpa FROM student\_grades

ORDER BY gpa DESC, semester\_end DESC;

SELECT student\_id, semester\_end, gpa FROM student\_grades

ORDER BY gpa DESC, semester\_end ASC;.

Explanation:

This answer shows correct syntax and semantics to receive desired result.

Question: 28. (B)

The ORDERS table has these columns:

|  |  |  |
| --- | --- | --- |
| ORDER\_ID | NUMBER(4) | NOT NULL |
| CUSTOMER\_ID | NUMBER(12) | NOT NULL |
| ORDER\_TOTAL | NUMBER(10,2) |  |

The ORDERS table tracks the Order nnmher, the order total, and the customer to whom the Order belongs. Which two statements retrieve orders with an inclusive total that ranges between 100.00 and 2000.00 dollars? (Choose two.)

SELECT customer\_id, order\_id, order\_total FROM orders

RANGE ON order\_total (100 AND 2000) INCLUSIVE;

SELECT customer\_id, order\_id, order\_total FROM orders

HAVING order\_total BETWEEN 100 and 2000;

SELECT customer\_id, order\_id, order\_total FROM orders

WHERE order\_total BETWEEN 100 and 2000;

SELECT customer\_id, order\_id, order\_total FROM orders

WHERE order\_total >= 100 and <= 2000;

SELECT customer\_id, order\_id, order\_total FROM orders

WHERE order\_total >= 100 and order\_total <= 2000;

Explanation:

Answers C and E provide correct results to show. You can use BETWEEN or comparison operations to retrieve data.

Question: 29. (B)

Examine the structure of the PRODUCT table.

PRODUCT Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRODUCT \_ID | | | NUMBER | | NOT NULL, Primary Key | | | |
| PRODUCT\_NAME | | | VARCHAR2 | |  | | | |
|  | | | (25) | |  | | | |
| SUPPLIER\_ID | | | NUMBER | | Foreign key to SUPPLIER\_ID of the | | | |
|  | | |  | | SUPPLIER table | | | |
| CATERORY\_ID | | | NUMBER | |  | | | |
| QTY\_PER\_UNIT | | | NUMBER | |  | | | |
|  |  | |  | |  |  |  | |  |
| LIST\_RRICE |  | | NUMBER (5,2) | |  |  |  | |  |
| COST |  | | NUMBER (5,2) | |  |  |  | |  |

You want to display all product identification numbers of products for which there are 500 or more available for immediate sale. You want the product numbers displayed alphabetically by supplier, then by product number from lowest to highest. Which statement should you use to achieve the required results?

SELECT product\_id FROM product

WHERE qty\_per\_unit >= 500 ORDER BY supplier\_id, product\_id;

SELECT product\_id FROM product

WHERE qty\_per\_unit >= 500 SORT BY supplier\_id, product\_id;

SELECT product\_id FROM product

WHERE qty\_per\_unit >= 500

ORDER BY supplier\_id, product\_id DESC;

SELECT product\_id FROM product

WHERE qty\_per\_unit > 500 SORT BY supplier\_id, product\_id;

Explanation:

SELECT product\_id

FROM product

WHERE qty\_per\_unit >= 500

ORDER BY supplier\_id, product\_id;

This statement will give the product\_id from product table where qty\_per\_unit will be equal to and greater than 500 and it will sort it in ascending order by default.

Question: 30. (B)

Examine the data in TEACHER table.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | LAST\_NAME | FIRST\_NAME | SUBJECT\_ID |
| 88 | Tsu | Ming | HST AMER |
| 70 | Smith | Ellen | HST INDIA |
| 56 | Jones | Karen | HST\_REVOL |
| 58 | Hann | Jeff | HST CURR |
| 63 | Hopewell | Mary Elizabetn | HST\_RELIG |

Which query should you use to return the following values from the TEACHER table? Name Subject

------------------------------------- -------------------

Jones, Karen HST\_REVOL

Hopewell, Mary Elizabeth HST\_RELIG

SELECT last\_name||', '||first\_name "Name", subject\_id "Subject" FROM teacher

WHERE subject\_id LIKE 'HST\\_%' ESCAPE '\';

SELECT last\_name||', '||first\_name "Name", subject\_id "Subject" FROM teacher

WHERE subject\_id = 'HST\\_R%';

SELECT last\_name||', '||first\_name "Name", subject\_id "Subject" FROM teacher

WHERE subject\_id LIKE '%HST\\_R%' ESC '\';

SELECT last\_name||', '||first\_name "Name", subject\_id "Subject" FROM teacher

WHERE subject\_id LIKE 'HST\_%';

Explanation:

When you need to have an exact match for the actual ‘%’ and ‘\_’ characters use the ESCAPE option. This option specifies what the ESCAPE character is.

Question: 31. (B)

You query the database with this SQL statement:

SELECT bonus

FROM salary

WHERE bonus BETWEEN 1 AND 250

OR (bonus IN(190, 500, 600)

AND bonus BETWEEN 250 AND 500);

Which value could the statement return?

100

260

400

600

Explanation:

For the above statement the value of bonus return will be 100. After resolving the above written operators the result return would be 100.

Question: 32. (B)

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?

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;

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;

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);

SELECT student\_id, marks, ROWNUM "Rank” FROM (SELECT student\_id, marks

FROM students

ORDER BY marks DESC) WHERE ROWNUM <= 10

AND finish\_date BETWEEN '01-JAN-99' AND '31-DEC-99' AND course\_id = 'INT\_SQL';

Explanation:

This statement using inline sub-query will provide correct results to show 10 students who achieved the highest ranking in the course INT SQL and who completed the course in the year 1999.

Question: 33. (B)

Examine the structure of the LINE\_ITEM table.

|  |  |  |
| --- | --- | --- |
| LINE\_ITEM\_ID | NUMBER(9) | NOT NULL, Primary Key |
| ORDER\_ID | NUMBER(9) | NOT NULL, Primary Key, Foreign Key to |
|  |  | ORDER\_ID column of the CURR\_ORDER table |
| PRODUCT\_ID | NUMBER(9) | NOT\_NULL, Foreign Key to PRODUCT\_ID column |
|  |  | of the PRODUCT table |
| QUANTITY | NUMBER(9) |  |

You must display the order number, line item number, product identification number, and quantity of each item where the quantity ranges from 10 through 100. The order numbers must be in the range of 1500 through 1575. The results must be sorted by order number from lowest to highest and then further sorted by quantity from highest to lowest.

Which statement should you use to display the desired result?

A. SELECT order\_id, line\_item\_id, product\_id, quantity

FROM line\_item

WHERE quantity BETWEEN 9 AND 101

AND order\_id BETWEEN 1500 AND 1575

ORDER BY order\_id DESC, quantity DESC;

SELECT order\_id, line\_item\_id, product\_id, quantity FROM line\_item

WHERE (quantity > 10 AND quantity < 100) AND order\_id BETWEEN 1500 AND 1575 ORDER BY order\_id ASC, quantity;

SELECT order\_id, line\_item\_id, product\_id, quantity FROM line\_item

WHERE (quantity > 9 OR quantity < 101) AND order\_id BETWEEN 1500 AND 1575 ORDER BY order\_id, quantity;

SELECT order\_id, line\_item\_id, product\_id, quantity FROM line\_item

WHERE quantity BETWEEN 10 AND 100 AND order\_id BETWEEN 1500 AND 1575 ORDER BY order\_id, quantity DESC;

Explanation:

Following query will give all the desire result by using to condition with ORDER BY clause with order\_id by default will be sorted in ascending order and quantity will be sorted in descending order

SELECT order\_id, line\_item\_id, product\_id, quantity FROM line\_item

WHERE quantity BETWEEN 10 AND 100 AND order\_id BETWEEN 1500 AND 1575 ORDER BY order\_id, quantity DESC;

Question: 34. (B)

The ITEM table contains these columns:

ITEM\_ID NUMBER(9)

COST NUMBER(7,2)

RETAIL NUMBER(7,2)

You need to create a report that displays the cost, the retail price, and the profit for item number 783920. To calculate the profit, subtract the cost of the item from its retail price, and then deduct an administrative fee of 25 percent of this derived value.

Which SELECT statement produces the desired results?

SELECT cost, retail, (retail - cost) - ((retail - cost) \* .25) "Profit" FROM item

WHERE item\_id = 783920;

SELECT cost, retail, (retail - cost) - retail - (cost \* .25) "Profit" FROM item

WHERE item\_id = 783920;

SELECT cost, retail, (retail - cost - retail - cost) \* .25 "Profit" FROM item

WHERE item\_id = 783920;

SELECT cost, retail, retail - cost - retail - cost \* .25 "Profit"

FROM item

WHERE item\_id = 783920;

Explanation:

To get a complete desired report you can use the following query with expressions

SELECT cost, retail, (retail - cost) - ((retail - cost) \* .25) "Profit"

FROM item

WHERE item\_id = 783920;

Question: 35. (B)

The ITEM table contains these columns:

ITEM\_ID NUMBER(9)

COST NUMBER(7,2)

RETAIL NUMBER(7,2)

The RETAIL and COST columns contain values greater than zero.

Evaluate these two SQL statements:

SELECT item\_id, (retail \* 1.25) + 5.00 - (cost \* 1.10) - (cost \* .10) AS Calculated Profit FROM item;

SELECT item\_id, retail \* 1.25 + 5.00 - cost \* 1.10 - cost \* .10 "Calculated Profit" FROM item;

What will be the result?

Statement 1 will display the 'Calculated Profit' column heading.

Statement 1 and statement 2 will return the same value.

Statement 1 will return a higher value than statement 2.

One of the statements will NOT execute.

Explanation:

For the above written statements the first statement will not execute because the column alias contain spaces so it should be place in the a double quotation marks for the statement to execute successful.

Question: 36. (B)

The EMP table contains these columns:

LAST NAME VARCHAR2(25)

SALARY NUMBER(6,2)

DEPARTMENT\_ID NUMBER(6)

You need to display the employees who have not been assigned to any department. You write the SELECT statement:

SELECT LAST\_NAME, SALARY, DEPARTMENT\_ID

FROM EMP

WHERE DEPARTMENT\_ID = NULL;

What is true about this SQL statement?

The SQL statement displays the desired results.

The column in the WHERE clause should be changed to display the desired results.

The operator in the WHERE clause should be changed to display the desired results.

The WHERE clause should be changed to use an outer join to display the desired results.

Explanation:

The operator in the WHERE clause should be changed to display the desired results. There are times when you want to substitute a value in place of NULL. Oracle provides this functionality with a special function, called NVL(). You cannot use operation equal with NULL, but you can achieve desired results using NVL() function after the WHERE clause.

Question: 37. (B)

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

A WHERE clause can be used to restrict both rows and groups.

A WHERE clause can be used to restrict rows only.

A HAVING clause can be used to restrict both rows and groups.

A HAVING clause can be used to restrict groups only.

A WHERE clause CANNOT be used in a query of the query uses a HAVING clause.

A HAVING clause CANNOT be used in subqueries.

, C

Explanation :

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

rows are grouped

the group function is applied to the group

the group that match the criteria in the Having clause are displayed.

WHERE clause cannot be use to restrict groups

HAVING clause use to restrict groups

WHERE clause cannot be use when there is group functions.

Question: 38. (B)

You are sorting data in a table in you SELECT statement in descending order. The column you are sorting on contains NULL records, where will the NULL record appears?

At the beginning of the list.

At the end of the list.

In the middle of the list.

At the same location they are listed in the unordered table.

Explanation:

When sorting a column with null values in ascending order then the oracle places the Null values at the end of the list if the sorting is in descending order the oracle places the null values at the start of the list.

Question: 39. (B)

The ACCOUNT table contains these columns:

ACCOUNT\_ID NUMBER(12)

PREVIOUS\_BALANCE NUMBER(7,2)

PAYMENTS NUMBER(7,2)

NEW\_PURCHASES NUMBER(7,2)

CREDIT\_LIMIT NUMBER(7)

You need to display the account number, finance charge, and current balance for accounts 1500 through 2000 with a current balance greater than the account's credit limit.

The finance charge is .9 percent (.009) of the previous balance. Adding the previous balance value,

new purchases value, and finance charge value, and then subtracting the payments value yields the current balance value.

Evaluate this statement:

SELECT account\_id, NVL(previous\_balance, 0) \* .009 finance\_charge, NVL(new\_purchases, 0) + (NVL(previous\_balance, 0) \* 1.009) - NVL(payments, 0) current balance

FROM account

WHERE (new\_purchases + (previous\_balance \* 1.009)) - payments > credit\_limit AND account\_id BETWEEN 1500 AND 2000;

Which statement about this SELECT statement is true?

The statement calculates the finance charge incorrectly.

The statement calculates the current balance incorrectly.

The statement returns only accounts that have NO previous balance.

The statement returns only accounts that have new purchases, previous balance, and payments values.

Ans. D

Question: 40.(C)

Examine the description of the EMPLOYEES table

|  |  |  |
| --- | --- | --- |
| EMP\_ID | NUMBER(4) | NOT NULL |
| LAST\_NAME | VARCHAR2(30) | NOT NULL |
| FIRST\_NAME | VARCHAR2(30) |  |
| DEPT\_ID | NUMBER(2) |  |
| JOB\_CAT | VARCHARD2(30) |  |
| SALARY | NUMBER(8,2) |  |

Which statement shows the maximum salary paid in each job category of each department?

SELECT dept\_id, job\_cat, MAX(salary) FROM employees

WHERE salary > MAX(salary);

SELECT dept\_id, job\_cat, MAX(salary) FROM employees

GROUP BY dept\_id, job\_cat;

SELECT dept\_id, job\_cat, MAX(salary) FROM employees;

SELECT dept\_id, job\_cat, MAX(salary) FROM employees

GROUP BY dept\_id;

SELECT dept\_id, job\_cat, MAX(salary) FROM employees

GROUP BY dept\_id, job\_cat, salary;

Explanation:

This answer provides correct syntax and semantics to show the maximum salary paid in each job category of each department.

Question: 41. (C)

Management has asked you to calculate the value 12\*salary\* comossion\_pct for all the employees in the EMP table. The EMP table contains these columns:

|  |  |  |
| --- | --- | --- |
| LAST NAME | VARCNAR2(35) | NOT NULL |
| SALARY | NUMBER(9,2) | NOT NULL |
| COMMISION\_PCT | NUMBER(4,2) |  |

Which statement ensures that a value is displayed in the calculated columns for all employees?

SELECT last\_name, 12\*salary\*commison\_pct FROM emp;

SELECT last\_name, 12\*salary\* (commission\_pct,0) FROM emp;

SELECT last\_name, 12\*salary\*(nvl(commission\_pct,0)) FROM emp;

SELECT last\_name, 12\*salary\*(decode(commission\_pct,0)) FROM emp;

Explanation:

This SELECT statement provides correct usage of NVL function to calculate columns for all employees. Oracle give you possibility to substitute a value in place of NULL. The basic syntax for NVL() is NVL(*column\_name, value\_if\_null*). Notice that the column specified in NVL() contains an actual value. That value is what Oracle returns; when the column is NULL, the special string is eturned. The value specified to be returned if the column value is NULL must be the same datatype as the column specified.

Question: 42. (C)

Examine the description of the STUDENTS table:

|  |  |
| --- | --- |
| STD\_ID | NUMBER(4) |
| COURSE\_ID | VARCHARD2(10) |
| START\_DATE | DATE |
| END\_DATE | DATE. |

Which two aggregate functions are valid on the START\_DATE column? (Choose two)

SUM(start\_date)

AVG(start\_date)

COUNT(start\_date)

AVG(start\_date, end\_date)

MIN(start\_date)

MAXIMUM(start\_date)

& E

Explanation:

It is possible to apply COUNT() and MIN() functions on the column with DATE data type.

Question: 43. (C)

The EMPLOYEE tables has these columns:

|  |  |
| --- | --- |
| LAST\_NAME | VARCNAR2(35) |
| SALARY | NUMBER(8,2) |
| COMMISSION\_PCT | NUMBER(5,2) |

You want to display the name and annual salary multiplied by the commission\_pct for all employees. For records that have a NULL commission\_pct, a zero must be displayed against the calculated column. Which SQL statement displays the desired results?

SELECT last\_name, (salary \* 12) \* commission\_pct FROM EMPLOYEES;

SELECT last\_name, (salary \* 12) \* IFNULL(commission\_pct, 0)

FROM EMPLOYEES;

SELECT last\_name, (salary \* 12) \* NVL2(commission\_pct, 0) FROM EMPLOYEES;

SELECT last\_name, (salary \* 12) \* NVL(commission\_pct, 0) FROM EMPLOYEES;

Explanation:

This SELECT statement provides correct usage of NVL function to calculate columns for all employees. Oracle give you possibility to substitute a value in place of NULL. The basic syntax for NVL() is NVL(*column\_name, value\_if\_null*). Notice that the column specified in NVL() contains an actual value. That value is what Oracle returns; when the column is NULL, the special string isreturned. The value specified to be returned if the column value is NULL must be the same datatype as the column specified.

Question: 44. (C)

You would like to display the system date in the format "Monday, 01 June, 2001". Which SELECT statement should you use?

SELECT TO\_DATE(SYSDATE, 'FMDAY, DD Month, YYYY') FROM dual;

SELECT TO\_CHAR(SYSDATE, 'FMDD, DY Month, 'YYY') FROM dual;

SELECT TO\_CHAR(SYSDATE, 'FMDay, DD Month, YYYY') FROM dual;

SELECT TO\_CHAR(SYSDATE, 'FMDY, DDD Month, YYYY')

FROM dual;

SELECT TO\_DATE(SYSDATE, 'FMDY, DDD Month, YYYY') FROM dual;

Explanation:

This answer is correct: “Day” shows the day spelled out, “DD” shows the two-digit date, “Month” provides the month spelled out, “YYYY” shows the four -digit year. “FMDay” is special format mask to suppresses the extra spaces between the name of the day and the number of the date.

Question: 45. (C)

Evaluate the SQL statement:

SELECT ROUND(TRUNC(MOD(1600,10),-1),2)

FROM dual;

What will be displayed?

0

1

0.00

An error statement

Explanation:

Result will be 0. MOD(x,y) function calculates the modulus of x, defined in long division as the integer remainder when x is divided by y until no further whole number can be produced. TRUNC() function truncates x to the decimal precision of y. ROUND(x,y) rounds x to the decimal precision of y.

Question: 46. (C)

Examine the description of the MARKS table:

|  |  |
| --- | --- |
| STD\_ID | NUMBER(4) |
| STUDENT\_NAME | VARCHAR2(30) |
| SUBJ1 | NUMBER(3) |
| SUBJ2 | NUMBER(3) |

SUBJ1 and SUBJ2 indicate the marks obtained by a student in two subjects. Examine this SELECT statement based on the MARKS table:

SELECT subj1+subj2 total\_marks, std\_id

FROM marks

WHERE subj1 > AVG(subj1) AND subj2 > AVG(subj2)

ORDER BY total marks;

What is the result of the SELECT statement?

The statement executes successfully and returns the student ID and sum of all marks for each student who obtained more than the average mark in each subject.

The statement returns an error at the SELECT clause.

The statement returns an error at the WHERE clause.

The statement returns an error at the ORDER BY clause.

Explanation:

The statement returns an error at the WHERE clause because group function AVG() cannot be used in the WHERE clause. Group functions can be used in SELECT clause and GROUP BY clause. They allow you to perform data operations on several values in a column of data as though the column were one collective group of data.

Question: 47. (C)

Which three SELECT statements displays 2000 in the format “$2,000.00”? (Choose three)

SELECT TO\_CHAR (2000, ‘$#,###.##’) FROM dual;

SELECT TO\_CHAR (2000, ‘$0,000.00’) FROM dual;

SELECT TO\_CHAR (2000, ‘$9,999.00’) FROM dual;

SELECT TO\_CHAR (2000, ‘$9,999.99’) FROM dual;

SELECT TO\_CHAR (2000, ‘$2,000.00’) FROM dual;

SELECT TO\_CHAR (2000, ‘$N,NNN.NN’) FROM dual;

Explanation:

Only queries in answers B, C and D will show result as in the format “$2,000.00”.

Question: 48. (C)

Examine the description of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMP\_ID | NUMBER(4) | NOT NULL |
| LAST\_NAME | VARCHAR2(30) | NOT NULL |
|  |  |  |
| FIRST\_NAME | VARCHAR2(30). |  |
| DEPT\_ID | NUMBER(2) |  |
| JOB\_CAT | VARCHAR2(30) |  |
| SALARY | NUMBER(8,2) |  |
|  |  |  |

Which statement shows the department ID, minimum salary, and maximum salary paid in that department, only of the minimum salary is less then 5000 and the maximum salary is more than 15000?

SELECT dept\_id, MIN(salary(, MAX(salary) FROM employees

WHERE MIN(salary) < 5000 AND MAX(salary) > 15000;

SELECT dept\_id, MIN(salary), MAX(salary) FROM employees

WHERE MIN(salary) < 5000 AND MAX(salary) > 15000 GROUP BY dept\_id;

SELECT dept\_id, MIN(salary), MAX(salary) FROM employees

HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;

SELECT dept\_id, MIN(salary), MAX(salary) FROM employees

GROUP BY dept\_id

HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;

SELECT dept\_id, MIN(salary), MAX(salary) FROM employees

GROUP BY dept\_id, salary

HAVING MIN(salary) < 5000 AND MAX(salary) > 15000;

Explanation:

This SELECT statement shows correct result.

Incorrect Answers

A: To provide correct data statement needs also GROUP BY clause.

B: This statement will not provide correct results.

C: HAVING clause can be used only in conjunction with GROUP BY clause. E: You need only grouping by department, not by salary.

Question: 49. (C)

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

You can use aggregate functions in any clause of a SELECT statement.

You can use aggregate functions only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.

You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns.

You can pass column names, expressions, constants, or functions as parameters to an aggregate function.

You can use aggregate functions on a table, only by grouping the whole table as one single group.

You cannot group the rows of a table by more than one column while using aggregate functions.

Explanation:

It is possible to mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns. Also it is acceptable to pass column names, expressions, constraints, or other functions as parameters to an aggregate function.

Question: 50. (C)

Which four statements correctly describe functions that are available in SQL? (Choose four)

INSTR returns the numeric position of a named character.

NVL2 returns the first non-null expression in the expression list.

TRUNCATE rounds the column, expression, or value to n decimal places.

DECODE translates an expression after comparing it to each search value.

TRIM trims the heading of trailing characters (or both) from a character string.

NVL compares two expressions and returns null if they are equal, or the first expression of they are not equal.

NULLIF compares two expressions and returns null if they are equal, or the first expression if they are not equal.

Explanation:

INSTR returns the numeric position of a named character. DECODE translates an expression after comparing it to each search value. TRIM trims the heading of trailing characters (or both)

from a character string. NULLIF compares twp expressions and returns null if they are equal, or the first expression if they are not equal.

Question: 51. (C)

Examine the structures of the PATIENT, PHYSICIAN, and ADMISSION tables.

PATIENT Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PATIENT\_ID | NUMBER |  | NOT NULL, Primary Key |  |
| LAST\_NAME | VARCHAR2 (30) | | NOT NULL |  |
| FIRST\_NAME | VARCHAR2 (25) | | NOT NULL |  |
| DOB | DATE |  |  |  |
| INS\_CODE | NUMBER |  |  |  |
| PHYSICIAN Table |  |  |  |  |
| PHYSICIAN \_ID | NUMBER |  | NOT NULL, Primary Key |  |
| LAST\_NAME | VARCHAR2 (30) | | NOT NULL |  |
| FIRST\_NAME | VARCHAR2 (25) | | NOT NULL |  |
| LICENSE\_NO | NUMBER (7) | | NOT NULL |  |
| HIRE\_DTAE | DATE |  |  |  |
| ADMISSION Table |  |  |  |  |
| PATIENT\_ID | NUMBER | NOT NULL, Primary Key, References PATIENT\_ID column of | | |
|  |  | the PATIENT table | | |
| PHYSICIAN\_ID | NUMBER | NOT NULL, Primary Key, References PHYSICIAN\_ID column | | |
|  |  | of the PHYSICIA table | | |
| ADMIT\_DATE | DATE |  |  |  |
| DISCHG\_DATE | DATE |  |  |  |
| ROOM\_ID | NUMBER | Foreign key to ROOM\_ID of the ROOM table | | |

Which SQL statement will produce a list of all patients who have more than one physician?

SELECT p.patient\_id FROM patient p

WHERE p.patient\_id IN (SELECT patient\_id FROM admission

GROUP BY patient\_id HAVING COUNT(\*) > 1);

SELECT DISTINCT a.patient\_id FROM admission a, admission a2 WHERE a.patient\_id = a2.patient\_id

AND a.physician\_id <> a2.physician\_id;

SELECT patient\_id

FROM admission

WHERE COUNT(physician\_id) > 1;

D. SELECT patient\_id

FROM patient FULL OUTER JOIN physician;

Explanation:

Self join can be used to find the above desired result to list all patients who have more than one physician as

SELECT DISTINCT a.patient\_id

FROM admission a, admission a2

WHERE a.patient\_id = a2.patient\_id

AND a.physician\_id <> a2.physician\_id;

Question: 52. (C)

Which clause should you use to exclude group results?

WHERE

HAVING

RESTRICT

GROUP BY

ORDER BY

Explanation:

HAVING clause is used to weed out unwanted data once the data is grouped using the GROUP BY statement.

Question: 53. (C)

In a SELECT statement that includes a WHERE clause, where is the GROUP BY clause placed in the SELECT statement?

Immediately after the SELECT clause

Before the WHERE clause

Before the FROM clause

After the ORDER BY clause

After the WHERE clause

Explanation:

The GROUP BY clause can be place only after the WHERE clause, or after FROM clause if there is no the WHERE clause in the statement.

Question: 54. (C)

Which two are character manipulation functions? (Choose two.)

TRIM

REPLACE

TRUNC

TO\_DATE

MOD

CASE

Explanation:

TRIM() and REPLACE() are character manipulation functions.

Question: 55. (C)

The EMPLOYEES table contains these columns:

|  |  |
| --- | --- |
| LAST\_NAME | VARCHAR2 (25) |
| SALARY | NUMBER (6,2) |
| COMMISSION\_PCT | NUMBER (6) |

You need to write a query that will produce these results:

Display the salary multiplied by the commission\_pct.

Exclude employees with a zero commission\_pct.

Display a zero for employees with a null commission value.

Evaluate the SQL statement:

SELECT LAST\_NAME, SALARY\*COMMISSION\_PCT

FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NOT NULL;

What does the statement provide?

All of the desired results

Two of the desired results

One of the desired results

An error statement

Explanation:

This statement will provide only one of the desired results: display the salary multiplied by the commission\_pct. It will not exclude employees with a zero commission\_pct and display a zero for employees with a null commission value.

Question: 56. (C)

Examine the structures of the EMPLOYEE and CURR\_ORDER tables as shown below:

EMPLOYEE Table

|  |  |  |  |
| --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key | |
| LAST\_NAME | VARCHAR2 (25) |  |  |
| FIRST NAME | VARCHAR2 (25) |  |  |
| DEPT\_ID | NUMBER | Foreign key to DEPT\_ID column of the | |
|  |  | DEPARTMENTS table | |
| JOB\_ID | NUMBER | Foreign key to JOB\_ID column of the JOBS table | |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column | |
| SALARY | NUMBER (9,2) |  |  |
| COMMISSION | NUMBER (7,2) |  |  |
| HIRE\_DATE | DATE |  |  |
| CURR\_ORDER Table | |  |  |
|  |  |  |  |
| ORDER\_ID | NUMBER |  | NOT NULL, Primary Key |
| CUSTOMER\_ID | NUMBER |  | Foreign key to CUSTOMER\_ID column of the |
|  |  |  | COSTOMERS table |
| EMPLOYEE\_ID | NUMBER |  | Foreign key to EMP\_ID column of the EMPLOYEES |
|  |  |  | table |
| ORDER\_DATE | DATE |  |  |
| ORDER\_AMT | NUMBER (7,2) |  |  |
| SHIP\_METHOD | NUMBER (5) |  |  |

You queried the database with this SQL statement:

SELECT a.last\_name, a.first\_name, a.job\_id, NVL(a.commission, 0), b.avgcomm

FROM employee a, (SELECT job\_id, AVG(commission) AVGCOMM

FROM employee

WHERE commission IS NOT NULL

GROUP BY job\_id) b

WHERE a.job\_id = b.job\_id

AND a.commission < b.avgcomm;

Which is a result of this query?

The AVG function's DISTINCT keyword must be used in the inner query or the statement will fail when executed.

The employee information displayed will be for employees that have a commission that is less than the average commission of all employees who have the same job.

The employee information displayed will be displayed in numeric order by the JOB\_ID and in alphabetical order by the LAST\_NAME where the JOB\_IDs are the same.

A self join CANNOT be used in an outer query when the inner query is an inline view. The self join must be placed in the inner query for the statement to execute successfully.

Explanation:

For the above query the information of only those employee will be displayed that have a commission that is less than the average commission of all employees who have the same job.

Question: 57. (C)

Examine the structures of the EMPLOYEE and CURR\_ORDER tables.

EMPLOYEE Table

|  |  |  |  |
| --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key | |
| LAST\_NAME | VARCHAR2 (25) |  |  |
| FIRST NAME | VARCHAR2 (25) |  |  |
| DEPT\_ID | NUMBER | Foreign key to DEPT\_ID column of the | |
|  |  | DEPARTMENTS table | |
| JOB\_ID | NUMBER | Foreign key to JOB\_ID column of the JOBS table | |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column | |
| SALARY | NUMBER (9,2) |  |  |
| COMMISSION | NUMBER (7,2) |  |  |
| HIRE\_DATE | DATE |  |  |
| CURR\_ORDER Table | |  |  |
| ORDER\_ID | NUMBER |  | NOT NULL, Primary Key |
| CUSTOMER\_ID | NUMBER |  | Foreign key to CUSTOMER\_ID column of the |
|  |  |  | COSTOMERS table |
| EMPLOYEE\_ID | NUMBER |  | Foreign key to EMP\_ID column of the EMPLOYEES |
|  |  |  | table |
| ORDER\_DATE | DATE |  |  |
| ORDER\_AMT | NUMBER (7, 2) |  |  |
| SHIP\_METHOD | NUMBER (5) |  |  |

To keep your top sales representatives motivated, your company plans to increase the bonuses of employees. You need to create a SELECT statement that returns the name, bonus, and maximum order amount associated with each employee for all employees whose bonus is less than 8 percent of their maximum order amount.

Which SELECT statement should you use?

SELECT e.last\_name, e.first\_name, e.commission, o.maxamt

FROM employee e, (SELECT employee\_id, MAX(order\_amt) MAXAMT FROM curr\_order

GROUP BY employee\_id) o

WHERE e.employee\_id = o.employee\_id AND e.commission < .08 \* o.maxamt;

SELECT e.last\_name, e.first\_name, e.commission, MAX(o.order\_amount) MAXAMT FROM employee e, curr\_order o

WHERE e.employee\_id = o.employee\_id AND e.commission < .08 \* o.maxamt

GROUP BY e.last\_name, e.first\_name, e.commission;

SELECT e.last\_name, e.first\_name, e.commission, MAX(o.order\_amt) FROM employee e, curr\_order

WHERE e.employee\_id = o.employee\_id

AND e.commission < .08 \* (SELECT employee\_id, MAX(order\_amt) MAXAMT FROM curr\_order

GROUP BY employee\_id);

SELECT last\_name, first\_name, commission, max\_amt

FROM employee, (SELECT employee\_id, MAX(order\_amt) MAXAMT FROM curr\_order

GROUP BY employee\_id)

WHERE employee.employee\_id = curr\_order.employee\_id AND commission < .08 \* maxamt;

Explanation:

Ans A contain the complete statement that will give you all the desire result needed from the query.

Question: 58. (C)

The PRODUCT table contains these columns:

PRODUCT\_ID NUMBER(9)

PRODUCT\_NAME VARCHAR2(25)

COST NUMBER(5,2)

LIST\_PRICE NUMBER(5,2)

SUPPLIER\_ID NUMBER(9)

You need to display product names, costs, supplier ids, and average list prices for all the products that cost more than the average cost of products provided by the same supplier. Which SELECT statement will achieve these results?

SELECT product\_name, cost, supplier\_id, AVG(list\_price) FROM product p, product a

WHERE p.supplier\_id = a.supplier\_id GROUP BY product\_name, cost, supplier\_id;

SELECT product\_name, cost, p.supplier\_id, AVG(list\_price) FROM product p, (SELECT supplier\_id, AVG(cost) avg\_cost FROM product

GROUP BY supplier\_id) a WHERE p.cost > a.avg\_cost

GROUP BY product\_name, cost, p.supplier\_id;

SELECT product\_name, cost, supplier\_id, AVG(list\_price) FROM product

WHERE supplier\_id IN (SELECT supplier\_id, AVG(cost) avg\_cost FROM product

GROUP BY supplier\_id)

GROUP BY product\_name, cost, supplier\_id;

SELECT product\_name, cost, p.supplier\_id, AVG(list\_price) FROM product p, (SELECT supplier\_id, AVG(cost) avg\_cost FROM product

GROUP BY supplier\_id) a

WHERE p.supplier\_id = a.supplier\_id AND p.cost > a.avg\_cost

GROUP BY product\_name, cost, p.supplier\_id;

Explanation:

For the above desired result the subquery will first find the average cost group by supplier\_id and then making a join this will give the cost of all products that cost more than the average cost as SELECT product\_name, cost, p.supplier\_id, AVG(list\_price)

FROM product p, (SELECT supplier\_id, AVG(cost) avg\_cost FROM product

GROUP BY supplier\_id) a

WHERE p.supplier\_id = a.supplier\_id AND p.cost > a.avg\_cost

GROUP BY product\_name, cost, p.supplier\_id;

Question: 60. (C)

The CUSTOMERS table has these columns:

|  |  |  |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(4) | NOT NULL |
| CUSTOMER\_NAME | VARCHAR2(100) | NOT NULL |
| STREET\_ADDRESS | VARCHAR2(150) |  |
| CITY\_ADDRESS | VARCHAR2(50) |  |
| STATE\_ADDRESS | VARCHAR2(50) |  |
| PROVINCE\_ADDRESS | VARCHAR2(50) |  |
| COUNTRY\_ADDRESS | VARCHAR2(50) |  |
| POSTAL\_CODE | VARCHAR2(12) |  |
| CUSTOMER\_PHONE | VARCHAR2(20) |  |

The CUSTOMER\_ID column is the primary key for the table.

Which two statements find the number of customers? (Choose two.)

SELECT TOTAL(\*) FROM customers;

SELECT COUNT(\*) FROM customers;

SELECT TOTAL(customer\_id) FROM customers;

SELECT COUNT(customer\_id) FROM customers;

SELECT COUNT(customers) FROM customers;

SELECT TOTAL(customer\_name) FROM customers;

Explanation:

These statements provide correct syntax and semantics to show the number of customers. Function COUNT() can be used with substitution symbol of all columns “\*” or just with one column name. Last query will be processed a little bit faster.

Question: 61. (C)

Examine the structures of the EMPLOYEES and TAX tables.

EMPLOYEES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPLOYEE\_ID |  | NUMBER |  | NOT NULL, Primary Key | |
| EMP\_NAME |  | VARCHAR2 (30) | |  |  |
| JOB\_ID |  | VARCHAR2 (20) | |  |  |
| SALARY |  | NUMBER |  |  |  |
| MGR\_ID |  | NUMBER |  | References EMPLOYEE\_ID column | |
| DEPARTMENT\_ID |  | NUMBER |  | Foreign key to DEPARTMENT\_ID column of the | |
|  |  |  |  | DEPARTMENTS table | |
| TAX |  |  |  |  |  |
| MIN\_SALARY | NUMBER | |  |  |  |
| MAX\_SALARY | NUMBER | |  |  |  |
| TAX\_PERCENT | NUMBER | | Percentage tax for given salary range | |  |

You need to find the percentage tax applicable for each employee. Which SQL statement would you use?

SELECT employee\_id, salary, tax\_percent FROM employees e, tax t

WHERE e.salary BETWEEN t.min\_salary AND t.max\_salary;

SELECT employee\_id, salary, tax\_percent FROM employees e, tax t.

WHERE e.salary > t.min\_salary, tax\_percent

SELECT employee\_id, salary, tax\_percent

FROM employees e, tax t

WHERE MIN(e.salary) = t.min\_salary AND MAX(e.salary) = t.max\_salary

D. You cannot find the information because there is no common column between the two tables.

Explanation:

You can find the percentage tax applicable for each employee by using SQL statement in answer A.

Question: 62. (C)

Examine the data in the WORKORDER table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WO\_ID | CUST\_ID | REQUIRED\_DT | COMPLE\_DT | AMOUNT |
| 1 | 1 | 04-DEC-2001 | 02-DEC-01 | 520.32 |
| 2 | 1 | 02-JAN-2002 |  |  |
| 3 | 2 | 17-JAN-2002 |  |  |
| 4 | 2 | 20-JAN-2002 | 05-JAN-2002 | 274.11 |
| 6 | 3 | 14-JAN-2001 | 13-JAN-2002 | 400.00 |
| 7 | 3 | 04-FEB -2002 |  |  |
| 8 | 4 | 01-FEB -2002 |  |  |
| 9 | 5 |  | 14-JAN-2002 |  |

The WORKORDER table contains these columns: WO\_ID NUMBER PK

CUST\_ID NUMBER REQUIRED\_DT DATE COMPL\_DT DATE AMOUNT NUMBER(7,2)

Which statement regarding the use of aggregate functions on the WORKORDER table is true?

Using the SUM aggregate function with the AMOUNT column is allowed in any portion of a SELECT statement.

Using the AVG aggregate function with any column in the table is allowed.

Using the SUM aggregate function on the AMOUNT column will result in erroneous results because the column contains null values.

Grouping on the REQUIRED\_DT and COMPL\_DT columns is NOT allowed.

Using the AVG aggregate function on the AMOUNT column ignores null values.

Using the MIN aggregate function on the COMPL\_DT column will return a null value.

Explanation:

All group functions except COUNT ignore null values. Using AVG aggregate function on the AMOUNT column ignores null values.

Question: 63.

The INVENTORY table contains these columns:

ID\_NUMBER NUMBER PK

CATEGORY VARCHAR2(10)

LOCATION NUMBER

DESCRIPTION VARCHAR2(30)

PRICE NUMBER(7,2)

QUANTITY NUMBER

You want to return the total of the extended amounts for each item category and location, including only those inventory items that have a price greater than $100.00. The extended amount of each item equals the quantity multiplied by the price. Which SQL statement will return the desired result?

SELECT category, SUM(price \* quantity) TOTAL, location FROM inventory

WHERE price > 100.00 GROUP BY category;

SELECT category, location, SUM(price) FROM inventory

WHERE price > 100.00 GROUP BY category, location;

SELECT category, SUM(price \* quantity) TOTAL, location FROM inventory

WHERE price > 100.00;

SELECT category, SUM(price \* quantity) TOTAL, location FROM inventory

WHERE price > 100.00 GROUP BY category, location;

Explanation:

You can find total extended amount by using SUM group by function and it can be grouped by catergory and then location with GROUP BY clause.

Question: 64. (C)

The EVENT table contains these columns:

EVENT\_ID NUMBER

EVENT\_NAME VARCHAR2(30)

EVENT\_DESC VARCHAR2(100)

EVENT\_TYPE NUMBER

LOCATION\_ID NUMBER

You have been asked to provide a report of the number of different event types at each location. Which SELECT statement will produce the desired result?

SELECT UNIQUE(location\_id), COUNT(event\_type) FROM event

GROUP BY location\_id;

SELECT COUNT(\*), DISTINCT(location\_id) FROM event;

SELECT DISTINCT (event\_type) FROM event

GROUP BY location\_id;

SELECT location\_id, COUNT(DISTINCT event\_type) FROM event

GROUP BY location\_id;

SELECT location\_id, MAX(DISTINCT event\_type) FROM event

GROUP BY location\_id;

Ans:d

Explanation:

Unique number of event types at each location can be find by using COUNT to count the number of events and then GROUP BY clause can be used with locatio\_id column to make it location wise.

Question: 65. (C)

Which two statements about the evaluation of clauses in a SELECT statement are true? (Choose two.)

The Oracle Server will evaluate a HAVING clause before a WHERE clause.

The Oracle Server will evaluate a WHERE clause before a GROUP BY clause.

The Oracle Server will evaluate a GROUP BY clause before a HAVING clause.

The Oracle Server will evaluate an ORDER BY clause before a WHERE clause.

The Oracle Server will evaluate an ORDER BY clause before a HAVING clause.

Explanation:

In SELECT statement WHERE clause comes after the FROM clause and before GROUP By clause while HAVING clause use to restrict the data group wise and it comes after the GROUP BY clause and ORDER BY clause comes after at end of SELECT statement.

Question: 66. (C)

Examine the structures of the EMPLOYEE and DEPARTMENT tables: EMPLOYEE

------------------

EMP\_ID NUMBER NOT NULL PK

NAME VARCHAR(30) NOT NULL

FNAME VARCHAR(25) NOT NULL DEPT\_NO NUMBER

TITLE VARCHAR2(25)

DEPARTMENT

------------------------

DEPT\_ID NUMBER NOT NULL PK DEPT\_NAME VARCHAR2(25)

You need to produce a list of departments, including the department names, that have more than three administrative assistants. Which SELECT statement will produce the desired result?

SELECT dept\_name

FROM employee JOIN department

ON employee.dept\_id = department.dept\_id

WHERE UPPER(title) = 'ADMINISTRATIVE ASSISTANT' GROUP BY dept\_name

HAVING emp\_id > 3;

SELECT dept\_name FROM employee GROUP BY dept\_no

HAVING LOWER(title) = 'administrative assistant' AND COUNT(\*) > 3;

SELECT dept\_name

FROM employee NATURAL JOIN department WHERE LOWER(title) = 'administrative assistant'

GROUP BY dept\_name

HAVING COUNT(emp\_id) > 3;

SELECT dept\_name

FROM employee e JOIN department d ON (e.dept\_no = d.dept\_id)

WHERE LOWER(title) = 'administrative assistant' AND COUNT(\*) > 3;

SELECT d.dept\_name

FROM employee e JOIN department d ON (e.dept\_no = d.dept\_id)

WHERE LOWER(title) = 'administrative assistant' GROUP BY dept\_name

HAVING COUNT(emp\_id) > 3;

SELECT d.dept\_name

FROM e.employee JOIN d.department ON (e.dept\_no = d.dept\_id)

WHERE LOWER(title) = 'administrative assistant' GROUP BY dept\_name

HAVING COUNT(emp\_id) > 3;

Explanation:

To obtain the above desired result LOWER function is used to used with the TITILE column value to find it whether values stored is in the lower case or in upper case it will convert it in to lower and compare it. COUNT function will count the whole values and restricted the result to three values by using HAVING clause.

Question: 67. (C)

Which two tasks can you perform by using the TO\_CHAR function? (Choose two)

Convert 10 to ‘TEN’

Convert ‘10’ to 10

Convert ‘10’ to ‘10’

Convert ‘TEN’ to 10

Convert a date to a character expression

Convert a character expression to a date

Explanation:

TO\_CHAR(x) function is used to convert the value x to a character or converts a date to a character string using formatting conventions.

Question: 68. (C)

Examine the data in the LINE\_ITEM table.

|  |  |  |  |
| --- | --- | --- | --- |
| LINE\_ITEM\_ID | ORDER\_ID | PRODUCT\_ID | QUANTITY |
| 2 | 1494 | A-2356 | 7 |
| 3 | 1533 | A-7849 | 18 |
| 6 | 1589 | C-589 | 33 |
| 1 | 1533 | A-3209 | 100 |
| 2 | 1533 | A-3210 | 1 |
| 4 | 1494 | Z-78 | 1 |
| 10 | 1588 | C-555 | 250 |
| 3 | 1494 | Z-79 | 5 |

You query the database and return the value 23. Which script did you use?

SELECT SUBSTR(product\_id, 3) FROM line\_item

WHERE line\_item\_id = 2 AND order\_id = 1494;

SELECT SUBSTR(product\_id, 3, -2) FROM line\_item

WHERE line\_item\_id = 2 AND order\_id = 1494;

SELECT SUBSTR(product\_id, -3, 2) FROM line\_item

WHERE line\_item\_id = 2 AND order\_id = 1494;

SELECT SUBSTR(product\_id, 3, 2) FROM line\_item

WHERE line\_item\_id = 2 AND order\_id = 1494;

Explanation:

A SUBSTR function can be used statement to extract a string of determined length.

Question: 69. (C)

Which SQL statement generates the alias Annual Salary for the calculated column SALARY\*12?

SELECT ename, salary\*12 ‘Annual Salary’ FROM employees;

SELECT ename, salary\*12 “Annual Salary” FROM employees;

SELECT ename, salary\*12 AS Annual Salary FROM employees;

SELECT ename, salary\*12 AS INITCAP(“ANNUAL SALARY”) FROM employees

Explanation:

This SQL statement provides correct syntax to generate the alias Annual Salary for the calculated column SALARY\*12.

Question: 70. (C)

Examine the structure of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | Primary Key |
| FIRST\_NAME | VARCHAR2(25) |  |
| LAST\_NAME | VARCHAR2(25) |  |
| DEPARTMENT\_ | ID NUMBER |  |
| SALARY | NUMBER |  |

What is the correct syntax for an inline view?

SELECT a.last\_name, a.salary, a.department\_id, b.maxsal FROM employees a,

(SELECT department\_id, max(salary)maxsal FROM employees

GROUP BY department\_id) b

WHERE a.department\_id = b.department\_id AND a.salary < b.maxsal;

SELECT a.last name, a.salary, a.department\_id FROM employees a

WHERE a.department\_id IN (SELECT department\_id FROM employees b

GROUP BY department\_id having salary = (SELECT max(salary) from employees))

SELECT a.last\_name, a.salary, a.department\_id FROM employees a

WHERE a.salary = (SELECT max(salary) FROM employees b

WHERE a.department\_id = b.department\_id);

SELECT a.last\_name, a.salary, a.department\_id FROM employees a

WHERE (a.department\_id, a.salary) IN (SELECT department\_id, a.salary) IN (SELECT department\_id max(salary) FROM employees b

GROUP BY department\_id ORDER BY department\_id);

Explanation:

This SQL statement shows correct syntax to build inline views. You must enclose the query text for the inline view in parentheses and also give a label for the inline view so that columns in it can be referenced later. In answer A inline view is marked as B.

Question: 71. (C)

Which clause would you use in a SELECT statement to limit the display to those employees whose salary is greater then 5000?

ORDER BY SALARY > 5000.

GROUP BY SALARY > 5000

HAVING SALARY > 5000

WHERE SALARY > 5000

Explanation:

You need to use the WHERE clause to limit the display to those employees whose salary is greater then 5000.

Question: 72. (C)

You need to calculate the total of all salaries in the accounting department. Which group function should you use?

MAX

MIN

SUM

COUNT

TOTAL

LARGEST

Explanation:

Function SUM(x) calculates the sum of all values in column x in all rows returned by the SELECT statement.

Question: 73. (C)

Which SELECT statement will the result ‘elloworld’ from the string ‘HelloWorld’?

SELECT SUBSTR( ‘HelloWorld’,1) FROM dual;

SELECT INITCAP(TRIM (‘HelloWorld’, 1,1)) FROM dual;

SELECT LOWER(SUBSTR(‘HellowWorld’, 1, 1) FROM dual;

SELECT LOWER(SUBSTR(‘HelloWorld’, 2, 1) FROM dual;

SELECT LOWER(TRIM (‘H’ FROM ‘HelloWorld’)) FROM dual;

Explanation:

TRIM function accept a string describing the data you would like to trim from a column value. It can trim from both side of column value i.e. left and right. In the following statement this function will trim as

SELECT LOWER(TRIM (‘H’ FROM ‘HelloWorld’)) FROM dual;

From the above statement trim function will remove the character ‘H’ from ‘HelloWorld’ and LOWER function will convert the remaining character to lower case.

Question: 74. (C)

Evaluate this SQL statement:

SELECT e.employee\_id, (.15\* e.salary) + (.5 \* e.commission\_pct) + (s.sales amount \* (.35 \* e.bonus)) AS CALC\_VALUE

FROM employees e, sales s WHERE e.employee\_id = s.emp\_id;

What will happen if you remove all the parentheses from the calculation?

The value displayed in the CALC\_VALUE column will be lower.

The value displayed in the CALC\_VALUE column will be higher.

There will be no difference in the value displayed in the CALC\_VALUE column.

An error will be reported.

Explanation:

There will be no difference in the value displayed in the CALC\_VALUE column because not arithmetic operations or usage of alias for the calculated expression in the SELECT clause will not cause change the value appearance.

Question: 75. (C)

Which script displays '01-JAN-02' when the ENROLL\_DATE value is '01-JUL-01'?

SELECT ROUND(enroll\_date, 'DAY') FROM student;

SELECT ROUND(enroll\_date, 'YEAR') FROM student;

SELECT ROUND(enroll\_date, 'MONTH') FROM student;

SELECT ROUND(TO \_CHAR(enroll\_date, 'YYYY')) FROM student;

Explanation:

ROUND function will round a value to the next higher value. In the above given scenario the ENROLL\_DATE will be round to the next year as the enroll\_date is higher than 30-jun-01 so it will be rounded to the next year by specifying the YEAR option with the ROUND function.

Question: 76. (C)

Which three functions can be used to manipulate character, number, or date column values? (Choose three.)

CONCAT

ROUND

TRUNC

RPAD

INSTR

Explanation:

CONCAT , ROUND and INSTR are three single row function that can be used to manipulate character, number and data column values. INSTR finds numeric position of named character CONCAT function it concatenate the first character value to the second value and ROUND fucnion will round the value to n decimal.

Question: 77. (C)

A new standard has been adopted in your department that all codes that contain only 3 characters must have a dash (-) and two character values appended to them. Which function can be used in your query to restrict the data displayed to only those codes containing 3 characters?

REPLACE

SUBSTR

LENGTH

RPAD

Answer: C

Explanation:

LENGTH function returns the number of characters in the value so you can restrict the data displayed to specified character by using LENGTH function.

Question: 78. (C)

Which statement concerning SQL functions is true?

Character functions can return character or number values.

Conversion functions convert a column definition from one data type to another data type.

Single-row functions can only be used in SELECT and WHERE clauses.

All date functions return DATE data type values.

Explanation:

Single row character functions accept character data as input and can return both character and number values. Character functions are case conversion function and character manipulation functions

Question: 79. (C)

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

The registrar has asked for a report on the average grade point average (GPA) for students enrolled during semesters that end in the year 2000. Which statement accomplish this?

SELECT AVERAGE(gpa) FROM student\_grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < 31-DEC-2000’;

SELECT COUNT(gpa) FROM student grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < ’31-DEC-2000’;

SELECT MID(gpa) FROM student grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < ’31-DEC-2000’;.

SELECT AVG(gpa) FROM student\_grades

WHERE semester\_end BETWEEN ’01-JAN-2000’ and ’31.DEC.2000’;

SELECT SUM(gpa) FROM student grades

WHERE semester\_end > ’01-JAN-2000’ and semester end < ’31-DEC-2000’;

SELECT MEDIAN(gpa) FROM student\_grades

WHERE semester end > ’01-JAN-2000’ and semester end < ’31-DEC-2000’;

Explanation:

Statement in this answer will show correct result, using function AVG(). This function takes the values for a single column on all rows returned by the query and calculates the average value for that column.

Question: 80. (C)

Evaluate the SQL statement:

SELECT a.emp\_name, a.sal, a.dept\_id, b.maxsal

FROM employees a,

(SELECT dept\_id, MAX(sal) maxsal

4. FROM employees

GROUP BY dept\_id) b

WHERE a.dept\_id = b.dept\_id

AND a.sal < b.maxsal;

What is the result of the statement?

The statement produces an error at line 1.

The statement produces an error at line 3.

The statement produces an error at line 6.

The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all departments that pay less salary then the maximum salary paid in the company.

The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all employees who earn less than the maximum salary in their department.

Explanation:

The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all employees who earn less than the maximum salary in their department. This query is example of an *inline view* which is the sub-query in the FROM clause of the main query. The sub-query can be a SELECT statement that utilizes joins, the GROUP BY clause, or the ORDER BY clause.

Question: 81. (C)

The EMP table has these columns:

ENAME VARCHAR2(35)

SALARY NUMBER(8,2)

HIRE\_DATE DATE

Management wants a list of names of employees who have been with the company for more than five years. Which SQL statement displays the required results?

SELECT ENAME FROM EMP

WHERE SYSDATE-HIRE\_DATE > 5;

SELECT ENAME FROM EMP

WHERE HIRE\_DATE -SYSDATE > 5;

SELECT ENAME FROM EMP

WHERE (SYSDATE -HIRE\_DATE)/365 > 5;

SELECT ENAME FROM EMP

WHERE (SYSDATE -HIRE\_DATE)\* 365 > 5;

Explanation:

Expression SYSDATE-HIRE\_DATE will show number of days after date of hiring employee, so you need to divide result of expression on 365 and this compare result with 5.

Question: 82. (C)

Which SELECT statement will the result ‘ello world’ from the string ‘Hello World’?

SELECT SUBSTR( ‘Hello World’,1) FROM dual;

SELECT INITCAP(TRIM (‘Hello World’, 1,1)) FROM dual;

SELECT LOWER(SUBSTR(‘Hello World’, 1, 1) FROM dual;

SELECT LOWER(SUBSTR(‘Hello World’, 2, 1) FROM dual;

SELECT LOWER(TRIM (‘H’ FROM ‘Hello World’)) FROM dual;

Explanation:

This statement will return correct result because function TRIM() will trim letter ‘H’ in the ‘Hello World’ and function LOWER() will return data in string in lowercase.

Question: 83. (C)

Evaluate the SQL statement:

SELECT ROUND(45.953, -1), TRUNC(45.936, 2)

FROM dual;

Which values are displayed?

46 and 45

46 and 45.93

50 and 45.93

50 and 45.9

45 and 45.93

45.95 and 45.93

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

Question: 84. (C)

The CUSTOMERS table has these columns:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CUSTOMER\_ID | | NUMBER (4) | | NOT NULL | |
| CUSTOMER\_NAME | | VARCHAR2 (100) | | NOT NULL | |
| STREET\_ADDRESS | | VARCHAR2 (150) | |  | |
| CITY\_ADDRESS | |  | |  | |
| STATE\_ADDRESS | | | | VARCHAR2 (50) | |  | |  |
| PROVINCE\_ADDRESS | | | | VARCHAR2 (50) | |  | |  |
| COUNTRY\_ADDRESS | | | | VARCHAR2 (50) | |  | |  |
| POSTAL\_CODE | |  | | VARCHAR2 (12) | |  | |  |
| CUSTOMER\_PHONE | | | | VARCHAR2 (20) | |  | |  |

The CUSTOMER\_ID column is the primary key for the table. You need to determine how dispersed your customer base is. Which expression finds the number of different countries represented in the CUSTOMERS table?

COUNT(UPPER(country\_address))

COUNT(DIFF(UPPER(country\_address)))

COUNT(UNIQUE(UPPER(country\_address)))

COUNT DISTINTC UPPER(country\_address)

COUNT(DISTINTC (UPPER(country\_address)))

Question: 85. (C)

You want to use single row function in your SQL statements which three of the following are number functions? (Choose three).

SINH

TO\_NUMBER.

SQRT.

ROUND.

Explanation:

Only To\_number is non numeric functions as this is a conversion function. SINH, SQRT & ROUND are all numeric functions.

Question: 86. (C)

You are using single row function in a SELECT statement which function can best be catergorized as similar in function to an IF-THEN-ELSE statement?

SQRT

DECODE

NEW\_TIME

ROWIDTOCHAR.

Explanation:

DECODE function acts like an IF-THEN-ELSE clause in your sql statement.

Question: 87. (C)

Which of the following functions are available in SQL? (Choose four)

INSTR.

NVL2.

TRUNCATE.

DECODE.

TRIM.

NVL.

NULLIF.

Explanation:

INSTR is a single row function that returns the numeric position of character in a given value DECODE function enables a powerful transformation of data from one value to another. This function works on the same principle as the if-then-else statement does in PL/SQL or in other common programming language.

TRIM function can trim character from left or right side of character or from both.

NULIFF compares two expressions and return null if they are equal or the fi rst expression if they are not equal.

Question: 88.(D)

Examine the structure of the EMPLOYEES and DEPARTMENTS tables:

EMPLOYEES

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Remarks |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| EMP\_NAME | VARCHAR2 (30) |  |
| JOB\_ID | VARCHAR2 (20) |  |
| SALARY | NUMBER |  |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID COLUMN |
| DEPARTMENT ID | NUMBER | Foreign key to DEPARTMENT ID column |
|  |  | of the DEPARTMENTS table |

DEPARTMENTS

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Remarks |
| DEPARTMENT\_ID | NUMBER | NOT NULL, Primary Key |
| DEPARTMENT\_NAME | VARCHAR2(30) |  |
| MGR\_ID | NUMBER | References MGR\_ID column of the EMPLOYEES table |

Evaluate this SQL statement:

SELECT employee\_id, e.department\_id, department\_name, salary

FROM employees e, departments d WHERE e.department\_id = d.department\_id;

Which SQL statement is equivalent to the above SQL statement?

SELECT employee\_id, department\_id, department\_name, salary.

FROM employees

WHERE department\_id IN (SELECT department\_id FROM departments);

SELECT employee\_id, department\_id, department\_name, salary

FROM employees

NATURAL JOIN departments;

SELECT employee\_id, d.department\_id, department\_name, salary

FROM employees e JOIN departments d

ON e.department\_id = d.department\_id;

SELECT employee\_id, department\_id, department\_name, Salary

FROM employees JOIN departments

USING (e.department\_id, d.department\_id);

Explanation:

This query shows correct JOIN ON clause syntax and provides equivalent to the above SQL statement.

Question: 89. (D)

Examine the data in the EMPLOYEES and DEPARTMENTS tables.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | EMPLOYEES | | |  |  |  |  |  |  |  |  |
|  | LAST\_NAME | DEPARTMENT\_ID | | | | | SALARY | |  |  |  |
|  | Getz |  | | 10 |  |  | 3000 |  |  |  |  |
|  | Davis |  | | 20 |  |  | 1500 |  |  |  |  |
|  | King |  | | 20 |  |  | 2200 |  |  |  |  |
|  | Davis |  | | 30 |  |  | 5000 |  |  |  |  |
|  | Kochhar |  | |  |  |  | 5000 |  |  |  |  |
|  | DEPARTMENTS | | |  |  |  |  |  |  |  |  |
|  | DEPARTMENT\_ID | | | |  | DEPARTMENT\_NAME | |  | |  |  |
|  | 10 |  |  |  |  | Sales | |  | |  |  |
|  | 20 |  |  |  |  | Marketing | |  | |  |  |
|  | 30 |  |  |  |  | Accounts | |  | |  |  |
|  | 40 |  |  |  |  | Administration | |  | |  |  |

You want to retrieve all employees, whether or not they have matching departments in the departments table. Which query would you use?

SELECT last\_name, department\_name FROM employees , departments(+);

SELECT last\_name, department\_name FROM employees JOIN departments (+);

SELECT last\_name, department\_name FROM employees(+) e JOIN departments d ON (e.department\_id = d.department\_id);

SELECT last\_name, department\_name FROM employees e

RIGHT OUTER JOIN departments d ON (e.department\_id = d.department\_id);

SELECT last\_name, department\_name FROM employees(+) , departments

ON (e.department\_id = d.department\_id);

SELECT last\_name, department\_name FROM employees e LEFT OUTER

JOIN departments d ON (e.department\_id = d.department\_id);

Explanation:

Answer F is correct. This query shows correct syntax to retrieve all employees, whether or not they have matching departments in the department table. Oracle9i extends its compliance with ANSI/ISO by supporting that standard’s requirements for outer join syntax and semantics.

Question: 90. (D)

In which case would you use a FULL OUTER JOIN?

Both tables have NULL values.

You want all unmatched data from one table.

You want all matched data from both tables.

You want all unmatched data from both tables.

One of the tables has more data than the other.

You want all matched and unmatched data from only one table.

Explanation:

Oracle9i also makes it possible for you to easily execute a full outer join, including all records from the tables that would have been displayed if you had used both LEFT OUTER JOIN or RIGTH OUTER JOIN clauses.

Question: 91. (D)

Which three are true regarding the use of outer joins? (Choose three.)

You cannot use IN operator in a condition that involves an outerjoin.

You use (+) on both sides of the WHERE condition to perform an outerjoin.

You use (\*) on both sides of the WHERE condition to perform an outerjoin.

You use an outerjoin to see only the rows that do not meet the join condition.

In the WHERE condition, you use (+) following the name of the column in the table without matching rows, to perform an outerjoin.

You cannot link a condition that is involved in an outerjoin to another condition by using the OR operator.

Explanation:

You can use an outerjoin to see only the rows that do not meet the join condition. In the WHERE condition, you use (+) following the name of the column in the table without matching rows, to perform an outerjoin. You cannot link a condition that is involved in an outerjoin to another condition by using the OR operator.

Incorrect Answers

You can use IN operator in a condition that involves an outerjoin.

You use (+) following the name of the column in the table without matching rows, but not on both sides of the WHERE condition to perform an outerjoin.

You don’t use (\*) to define outerjoin operation.

Question: 92. (D)

What is true about joining tables through an equijoin?

You can join a maximum of two tables through an equijoin.

You can join a maximum of two columns through an equijoin.

You specify an equijoin condition in the SELECT or FROM clauses of a SELECT statement.

To join two tables through an equijoin, the columns in the join condition must be primary key and foreign key columns.

You can join n tables (all having single column primary keys) in a SQL statement by specifying a minimum of n-1 join conditions.

Explanation:

For N joined tables using Oracle or ANSI/ISO syntax for table joins, you need at least N-1 equijoin conditions in the WHERE clause of your SELECT statement or N-1 JOIN *table\_name* ON *join\_condition* clauses in order to avoid a Cartesian product, respectively.

Question: 93. (D)

Examine the structure of the EMPLOYEES, DEPARTMENTS, and TAX tables.

EMPLOYEES

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | | | NOT NULL, Primary Key | | |
| EMP\_NAME | VARCHAR2 (30) | | |  |  |  |
| JOB\_ID | VARCHAR2 (20) | | |  |  |  |
| SALARY | NUMBER | | |  |  |  |
| MGR\_ID | NUMBER | | | References EMPLOYEE\_ID column | | |
| DEPARTMENT\_ID | NUMBER | | | Foreign key to DEPARTMENT\_ID column of | | |
|  |  |  |  | the DEPARTMENTS table | | |
| DEPARTMENTS |  |  |  |  |  |  |
| DEPARTMENT\_ID |  | NUMBER | |  | NOT NULL, Primary Key |  |
| DEPARTMENT\_NAME |  | VARCHAR2 |30| | |  |  |  |
| MGR\_ID |  | NUMBER | |  | References MGR\_ID column of the |  |
|  |  |  |  |  | EMPLOYEES table |  |
| TAX |  |  |  |  |  |  |
| MIN\_SALARY | NUMBER | |  |  |  |  |
| MAX\_SALARY | NUMBER | |  |  |  |  |
| TAX\_PERCENT | NUMBER | |  |  |  |  |

For which situation would you use a nonequijoin query?

To find the tax percentage for each of the employees.

To list the name, job id, and manager name for all the employees.

To find the name, salary, and department name of employees who are not working with Smith.

To find the number of employees working for the Administrative department and earning less then 4000.

To display name, salary, manager ID, and department name of all the employees, even if the employees do not have a department ID assigned.

Explanation:

You will use a nonequijoin query to find the tax percentage for each of the employees because you will check range of salaries without usage of equality operations in a comparison joining data from two tables.

Question: 94. (D)

Examine the structure of the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables.

EMPLOYEES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER NOT | |  | NOT NULL, Primary Key |
| EMP\_NAME | VARCHAR2 (30) | | |  |
| JOB\_ID | VARCHAR2 (20) | | |  |
| SALARY | NUMBER | |  |  |
| MGR\_ID | NUMBER | |  | References EMPLOYEE\_ID column |
| DEPARTMENT\_ID | NUMBER | |  | Foreign key to DEPARTMENT\_ID |
|  |  |  |  | column of the DEPARTMENTS table |
| DEPARTMENTS |  |  |  |  |
| DEPARTMENT\_ID | NUMBER |  | NOT NULL, Primary Key | |
| DEPARTMENT\_NAME | VARCHAR2 |  |  |  |
|  | (30) |  |  |  |
| MGR\_ID | NUMBER |  | References NGR\_ID column of the EMPLOYEES | |
|  |  |  | table | |
|  |  |  |  | |
| LOCATION\_ID | NUMBER |  | Foreign key to LOCATION\_ID column of | |
|  |  |  | theLOCATIONS table | |
|  |  |  |  |  |

LOCATIONS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | |  |  |
| LOCATION\_ID |  | NUMBER |  | NOT NULL, Primary Key | |  |  |
| CITY |  | VARCHAR2 |30) |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Which two SQL statements produce the name, department name, and the city of all the employees who earn more then 10000? (Choose two)

SELECT emp\_name, department\_name, city FROM employees e

JOIN departments d USING (department\_id) JOIN locations 1 USING (location\_id) WHERE salary > 10000;

SELECT emp\_name, department\_name, city FROM employees e, departments d, locations 1 JOIN ON (e.department\_id = d.department id) AND (d.location\_id =1.location\_id)

AND salary > 10000;

SELECT emp\_name, department\_name, city FROM employees e, departments d, locations 1 WHEREsalary > 10000;

SELECT emp\_name, department\_name, city FROM employees e, departments d, locations 1 WHEREe.department\_id = d.department\_id AND d.location\_id = 1.location\_id

AND salary > 10000;

SELECT emp\_name, department\_name, city FROM employees e

NATURAL JOIN departments, locations WHEREsalary > 10000;

Explanation:

These statements show correct syntax and semantics to receive correct results.

Question: 95. (D)

Examine the data from the CLASS and INSTRUCTOR tables.

CLASS Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CLASS\_I | CLASS\_NAME | | HOURS\_CREDI | | INSTRUCTOR\_ID | |
| D |  | | T | |  | |
| 1 | Introduction to | | 3 | | 4 | |
|  |  |  |  |  |  |  |  |  |
|  |  | Accounting |  |  |  |  |  |  |
| 2 |  | Computer Basics |  | 3 |  | 1 |  |  |
| 3 |  | Tax Accounting |  | 3 |  | 4 |  |  |
|  |  | Principles |  |  |  |  |  |  |
| 4 |  | American History |  | 3 |  | 2 |  |  |
| 5 |  | Basic Engineering |  | 3 |  |  |  |  |

INSTRUCTOR Table

|  |  |  |
| --- | --- | --- |
| INSTRUCTOR\_ID | LAST\_NAME | FIRST\_NAME |
| 1 | Chao | Ling |
| 2 | Vanderbilt | Herbert |
| 3 | Wigley | Martha |
| 4 | Page | Albert |

You have been asked to produce a report of all instructors, including the classes taught by each instructor. All instructors must be included on the report, even if they are not currently assigned to teach classes. Which two SELECT statements could you use? (Choose two.)

SELECT i.last\_name, i.first\_name, c.class\_name FROM instructor i NATURAL JOIN class c

ON (i.instructor\_id = c.instructor\_id);

SELECT i.last\_name, i.first\_name, c.class\_name FROM instructor i, class c;

SELECT i.last\_name, i.first\_name, c.class\_name

FROM class c LEFT OUTER JOIN instructor i

ON (i.instructor\_id = c.instructor\_id)

ORDER BY i.instructor\_id;

SELECT i.last\_name, i.first\_name, c.class\_name FROM instructor i, class c

WHERE i.instructor\_id = c.instructor\_id (+) ORDER BY i.instructor\_id;

SELECT i.last\_name, i.first\_name, c.class\_name FROM instructor i LEFT OUTER JOIN class c ON (i.instructor\_id = c.instructor\_id)

ORDER BY i.instructor\_id;

SELECT i.last\_name, i.first\_name, c.class\_name

FROM instructor i, class c

WHERE i.instructor\_id (+) = c.instructor\_id

ORDER BY i.instructor\_id;

& E

Explanation:

Two methods can be used using outer join to find the class with no instructor you can use a plus sign (+) with the with the side that has information missing and LEFT OUTER JOIN is also used to find the missing information.

Question: 96. (D)

Which two operators can be used in an outer join condition? (Choose two.)

=

OR

IN

AND

& D

Explanation:

Outer Join can be used to also see rows that do not meet the join condition.

For an outer join you can use the equality (=) operator with AND operator in the join condition.

Question: 97. (D)

Examine the data from the CLASS and INSTRUCTOR tables.

CLASS Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CLASS\_ID | CLASS\_NAME | | HOURS\_CREDIT | | INSTRUCTOR\_ID |
| 1 | Introduction to | | 3 |  | 4 |
|  | Accounting | |  |  |  |
| 2 | Computer Basics | | 3 |  | 1 |
| 3 | Tax Accounting | | 3 |  | 4 |
|  | Principles | |  |  |  |
| 4 | American History | | 3 |  | 2 |
| 5 | Basic Engineering | |  |  |  |
| INSTRUCTOR | Table | |  |  |  |
| INSTRUCTOR\_ID | | LAST\_NAME | FIRST\_NAME |  |  |
| 1 |  | Chao | Ling |  |  |
| 2 |  | Vanderbilt | Herbert |  |  |
| 3 |  | Wigley | Martha |  |  |
| 4 |  | Page | Albert |  |  |

You want to retrieve a list of all classes, including those with no instructor currently assigned. Which SELECT statement should you use?

SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName FROM instructor i, class c

WHERE i.instructor\_id = c.instructor\_id (+);

SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName FROM instructor I JOIN class c

WHERE i.instructor\_id (+) = c.instructor\_id;

SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName FROM instructor i RIGHT OUTER JOIN class c

ON (i.instructor\_id = c.instructor\_id);

SELECT class\_name, last\_name || ', ' || first\_name InstructorName FROM instructor RIGHT OUTER JOIN class;

SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName FROM instructor i RIGHT OUTER JOIN class c

USING (instructor\_id);

Explanation:

Oracle 9i allows the use of one of two directives, left outer join and right outer join. To find all a list of all classes that do not have any instructor assigned can be retrieved as

SELECT c.class\_name, i.last\_name || ', ' || i.first\_name InstructorName

FROM instructor i RIGHT OUTER JOIN class c

ON (i.instructor\_id = c.instructor\_id);

Question: 98. (D)

To produce a meaningful result set without any cartesian products, what is the minimum number of conditions that should appear in the WHERE clause of a four-table join?

8

2

3

4

5

There is no such criteria

Explanation:

When we join different tables in where clause so n-1 conditions are used and n is the number of tables which we are join.

Question: 99.(E)

Examine the data in the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | DEPARTMENT\_ID | SALARY |
| Getz | 10 | 3000 |
| Davis | 20 | 1500 |
| King | 20 | 2200 |
| Davis | 30 | 5000 |
| … |  |  |

Which three subqueries work? (Choose three)

SELECT \*

FROM employees

where salary > (SELECT MIN(salary) FROM employees

GROUP BY department.id);

SELECT \*

FROM employees

WHERE salary = (SELECT AVG(salary) FROM employees

GROUP BY department\_id);

SELECT distinct department\_id FROM employees

Where salary > ANY (SELECT AVG(salary) FROM employees

GROUP BY department\_id);

SELECT department\_id FROM employees

WHERE SALARY > ALL (SELECT AVG(salary) FROM employees.

GROUP BY department\_id);

SELECT last\_name FROM employees

Where salary > ANY (SELECT MAX(salary) FROM employees

GROUP BY department\_id);

SELECT department\_id FROM employees

WHERE salary > ALL (SELECT AVG(salary) FROM employees

GROUP BY AVG(SALARY));

Explanation:

These answers show correct syntax, because they use ANY and ALL keywords for convert multi-row output of sub-query to one-row result.

Question: 100. (E)

Examine the data from the ORDERS and CUSTOMERS table.

ORDERS

|  |  |  |  |
| --- | --- | --- | --- |
| ORD\_ID | ORD\_DATE | CUST\_ID | ORD\_TOTAÖ |
| 100 | 12-JAN-2000 | 15 | 10000 |
| 101 | 09-MAR-2000 | 40 | 8000 |
| 102 | 09-MAR-2000 | 35 | 12500 |
| 103 | 15-MAR-2000 | 15 | 12000 |
| 104 | 25-JUN-2000 | 15 | 6000 |
| 105 | 18-JUL-2000 | 20 | 5000 |
| 106 | 18-JUL-2000 | 35 | 7000 |
| 107 | 21-JUL-2000 | 20 | 6500 |
| 108 | 04-AUG -2000 | 10 | 8000 |

CUSTOMERS

|  |  |  |
| --- | --- | --- |
| CUST\_ID | CUST\_NAME | City |
| 10 | Smith | LosAngeles |
| 15 | Bob | San Francisco |
| 20 | Martin | Chicago |
| 25 | Mary | New York |
| 30 | Rina | Chicago |
| 35 | Smith | New York |
| 40 | Linda | New York |

Which SQL statement retrieves the order ID, customer ID, and order total for the orders that are placed on the same day that Martin places his orders?

SELECT ord\_id, cust\_id, ord\_total FROM orders, customers WHERE cust\_name=’Mating’

AND ord\_date IN (’18-JUL-2000’,’21-JUL-2000’);

SELECT ord\_id, cust\_id, ord\_total FROM orders

Where ord\_date IN (SELECT ord\_date FROM orders

WHERE cust\_id = (SELECT cust\_id FROM customers

WHERE cust\_name = ‘Martin’));

SELECT ord\_id, cust\_id, ord\_total FROM orders

Where ord\_date IN (SELECT ord\_date FROM orders, customers

Where cust\_name = ‘Martin’);

SELECT ord\_id, cust\_id, ord\_total FROM orders

WHERE cust\_id IN (SELECT cust\_id FROM customers

WHERE cust name = ‘Martin’);.

Explanation:

This query will return the order ID, customer ID, and order total for the orders that are placed on the same day that Martin places his orders.

Question: 101. (E)

Examine the data from the ORDERS and CUSTOMERS tables.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ORDERS | |  |  |  |  |  |  |  |  |  |  |  |
| ORD\_ID | |  |  | ORD\_DATE |  |  | CUST\_ID | ORD\_TOTAÖ | | |  |  |
| 100 |  |  |  | 12-JAN-2000 |  | 15 | | 10000 |  |  |  |  |
| 101 |  |  |  | 09-MAR-2000 |  | 40 | | 8000 |  |  |  |  |
| 102 |  |  |  | 09-MAR-2000 |  | 35 | | 12500 |  |  |  |  |
| 103 |  |  |  | 15-MAR-2000 |  | 15 | | 12000. |  |  |  |  |
| 104 |  |  |  | 25-JUN-2000 |  | 15 | | 6000 |  |  |  |  |
| 105 |  |  |  | 18-JUL-2000 |  | 20 | | 5000 |  |  |  |  |
| 106 |  |  |  | 18-JUL-2000 |  | 35 | | 7000 |  |  |  |  |
| 107 |  |  |  | 21-JUL-2000 |  | 20 | | 6500 |  |  |  |  |
| 109 |  |  |  | 04-AUG -2000 |  | 10 | | 8000 |  |  |  |  |
| CUSTOMERS | | | |  |  |  |  |  |  |  |  |  |
| CUST\_ID | |  |  | CUST\_NAME | CITY | | |  |  |  |  |  |
| 10 |  |  |  | Smith | Los Angeles | | |  |  |  |  |  |
| 15 |  |  |  | Bob | San Francisco | | |  |  |  |  |  |
| 20 |  |  |  | Martin | Chicago | | |  |  |  |  |  |
| 25 |  |  |  | Mary | New York | | |  |  |  |  |  |
| 30 |  |  |  | Rina | Chicago | | |  |  |  |  |  |
| 35 |  |  |  | Smith | New York | | |  |  |  |  |  |
| 40 |  |  |  | Lind | New York | | |  |  |  |  |  |
| Evaluate the SQL statement: | | | | |  |  |  |  |  |  |  |  |
| SELECT \* | |  |  |  |  |  |  |  |  |  |  |  |
| FROM orders | | | |  |  |  |  |  |  |  |  |  |
| WHERE cust\_id = (SELECT cust\_id | | | | | | | |  |  |  |  |  |
| FROM customers | | | |  |  |  |  |  |  |  |  |  |
| WHERE cust\_name = 'Smith'); | | | | |  |  |  |  |  |  |  |  |
| What is the result when the query is executed? | | | | | | | |  |  |  |  |  |
| A. | |  |  |  |  |  |  |  |  |  |  |  |
| ORD\_ID | ORD\_DATE | | | |  |  | CUST\_ID | ORD\_TOTAL | |  | |  |
| 102 |  | 09-MAR-2000 | | |  | | 35 | 12500 |  |  |  |  |
| 106 |  | 18-JUL-2000 | | |  | | 35 | 7000 |  |  |  |  |
| 108 |  | 04-AUG-2000 | | |  | | 10 | 8000 |  |  |  |  |

B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORD\_ID | ORD\_DATE | CUST\_ID | ORD\_TOTAL | |
| 102 | 09-MAR-2000 | 35 | 12500 | |
| 106 | 18-JUL-2000 | 35 | 7000 | |
| C. |  |  |  |  |
| ORD\_ID | ORD\_DATE | CUST\_ID |  | ORD\_TOTAL |
| 108 | 04-AUG -2000 | 10 |  | 8000 |

The query fails because the subquery returns more than one row.

The query fails because the outer query and the inner query are using different tables.

Explanation:

The query fails because the sub-query returns more than one row: it is possible situation when there are some customers with name Smith.

Question: 102. (E)

Evaluate this SQL statement:

SELECT product\_id, product\_name, price FROM product

WHERE supplier\_id IN (SELECT supplier\_id FROM product WHERE price > 120

OR qty\_in\_stock > 100);

Which values will be displayed?

The PRODUCT\_ID, PRODUCT\_NAME, and PRICE of products that are priced greater than $120.00 and have a QTY\_IN\_STOCK value greater than 100.

The PRODUCT\_ID , PRODUCT\_NAME, and PRICE of products that are priced greater than $120.00 or that have a QTY\_IN\_STOCK value greater than 100.

The PRODUCT\_ID, PRODUCT\_NAME, and PRICE of products that are priced greater than $120.00 or that have a QTY\_IN\_STOCK value greater than 100 and have a supplier.

The PRODUCT\_ID, PRODUCT\_NAME, and PRICE of products supplied by a supplier with products that are priced greater than $120.00 or with products that have a QTY\_IN\_STOCK value greater than 100.

Explanation:

Ans D is correct for the above written query as only the product\_id, product\_name and price of all those product those have price greater than $120 or have QTY\_IN\_STOCK is greater than 100. this has been done by using subquery you can also get same result by omitting subquery and use the conditions directly in the WHERE clause of the main query.

Question: 103. (E)

Which statement regarding subqueries is true?

A subquery CANNOT reference a table that is not included in the outer query's FROM clause.

Subqueries can be nested up to 5 levels.

A subquery must be placed on the right side of the comparison operator.

Subqueries can return multiple columns.

Explanation:

Subquery is a SELECT statement which is embedded in a clause of another SELECT statement retrieve data base on an unknown condition. You can build a powerful statement out of simple

one by using subqueries. A single row subqueries can return only one row but many columns while a multiple row subqueries can return multiple rows and multiple columns

Question: 104. (E)

Evaluate this SELECT statement:

SELECT employee\_id, name

FROM employee

WHERE employee\_id NOT IN (SELECT employee\_id

FROM employee

WHERE department\_id = 30

AND job = 'CLERK');

What would happen if the inner query returned a NULL value?

No rows would be selected from the EMPLOYEE table.

All the EMPLOYEE\_ID and NAME values in the EMPLOYEE table would be displayed.

Only the rows with EMPLOYEE\_ID values equal to NULL would be included in the results.

A syntax error would be returned.

Explanation:

If a subquery return a NULL value to the main query then the no rows would be select from the employee table.

Question: 105. (E)

Examine the data from the DONATION table (PLEDGE\_ID is the primary key).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PLEDGE\_I | DONOR\_I | PLEDGE\_D | AMOUNT\_PLED | AMOUNT\_P | PAYMENT\_DT |
| N | D | T | GED | AID |  |
| 1 | 1 | 10-SEP- | 1000 | 1000 | 02-OCT-2001 |
|  |  | 2001 |  |  |  |
| 2 | 1 | 22-EFB - | 1000 |  |  |
|  |  | 2002 |  |  |  |
| 3 | 2 | 08-OCT- | 10 | 10 | 28-OCT-2001 |
|  |  | 2001 |  |  |  |
| 4 | 2 | 10-DEC- | 50 |  |  |
|  |  | 2001 |  |  |  |
| 5 | 3 | 02-NOV- | 10000 | 9000 | 28-DEC-2001 |
|  |  | 2001 |  |  |  |
| 6 | 3 | 05-JAN- | 1000 | 1000 | 31-JAN-2002 |
|  |  | 2002 |  |  |  |
| 7 | 4 | 09-NOV- | 2100 | 2100 | 15-DEC-2001 |
|  |  | 2001 |  |  |  |
| 8 | 5 | 09-DEC- | 110 | 110 | 29-DEC-2001 |
|  |  | 2001 |  |  |  |

This statement fails when executed: SELECT amount\_pledged, amount\_paid FROM donation

WHERE donor\_id = (SELECT donor\_id FROM donation

WHERE amount\_pledged = 1000.00

OR pledge\_dt = '05-JAN-2002');

Which two changes could correct the problem? (Choose two.)

Remove the subquery WHERE clause.

Change the outer query WHERE clause to 'WHERE donor\_id IN'.

Change the outer query WHERE clause to 'WHERE donor\_id LIKE'.

Remove the single quotes around the date value in the inner query WHERE clause.

Change the subquery WHERE clause to 'WHERE amount\_pledged = 1000.00 AND pledge\_dt = '05-JAN-2002' '.

Include the DONOR\_ID column in the select list of the outer query.

Explanation:

When using multiple row subquery then you must use multi row operator with the subquery as in the above written statement equality operator (=) use with the WHERE clause should be replaced with IN operator to execute the query successfully. For the above query you can also remove the subquery and make the condtion in the where clause of main query which is used in the subquery.

Question: 106. (E)

A subquery can be used to \_\_\_\_\_\_\_\_\_.

Create groups of data.

Sort data in a specific order

Convert data to a different format

Retrieve data based on an unknown condition

Explanation:

A sub-query can be used to retrieve data based on an unknown condition

Question: 107. (E)

Which three statements about subqueries are true? (Choose three)

A single row subquery can retrieve only one column and one row.

A single row subquery can retrieve only one row but many columns.

A multiple row subquery can retrieve multiple rows and multiple columns.

A multiple row subquery can be compared by using the “>” operator.

A single row subquery can use the IN operator.

A multiple row subquery can use the “=” operator.

Answer: B, C & D

Explanation:

A single row sub-query can retrieve only one row but many columns. A multiple row subquery can retrieve one row or multiple rows and multiple columns. A multiple row sub-query can be compared by using the “>” operator.

Question: 108. (E)

You define a multiple-row subquery in the WHERE clause of an SQL query with a comparison operator "=".

What happens when the main query is executed?

The main query executes with the first value returned by the subquery.

The main query executes with the last value returned by the subquery.

The main query executes with all the values returned by the subquery.

The main query fails because the multiple-row subquery cannot be used with the comparison operator

You cannot define a multiple-row subquery in the WHERE clause of a SQL query.

Explanation:

The main query fails because the multiple-row sub-query cannot be used with the comparison operator. Only single-row query can use comparison operators, like =, <, >, <=, >, and <>.

Question: 109. (E)

Which operator can be used with a multiple-row subquery?

=

LIKE

BETWEEN

NOT IN

IS

<>

Explanation:

Only NOT IN operator can be used with a multi-row sub-query. All others may be used with single-row sub-query only.

Question: 110. (E)

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

A single row subquery can retrieve data from only one table.

A SQL query statement cannot display data from table B that is referred to in its subquery, unless table B is included in the main query's FROM clause.

A SQL query statement can display data from table B that is referred to in its subquery, without including table B in its own FROM clause.

D A single row subquery can retrieve data from more than one table.

A single row subquery cannot be used in a condition where the LIKE operator is used for comparison.

A multiple-row subquery cannot be used in a condition where the LIKE operator is used for comparison.

Explanation:

A SQL query statement cannot display data from table B that is referred to in its sub-query, funless table B is included in the main query's FROM clause. And a single row sub-query can retrieve data from more than one table.

Question: 111. (E)

Examine the data from the EMP table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | |  | |  |
| 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 | 300 | |  | |  | | |  |

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

Which three tasks would require subqueries or joins in order to perform in a single step? (Choose three)

Deleting the records of employees who do not earn commission.

Increasing the commission of employee 3 by the average commission earned in department 20.

Finding the cumber of employees who do NOT earn commission and are working for department 20.

Inserting into the table a new employee 10 who works for department 20 and earns a commission that is equal to the commission earned by employee 3.

Creating a table called COMMISSION that has the same structure and data as the columns EMP\_ID and COMMISSIONS of the EMP table.

Decreasing the commission by 150 for the employees who are working in department 30 and earning a commission of more then 800.

Explanation:

Increasing the commission of employee 3 by the average commission earned in department 20 will require sub-queries or joins in order to perform in a single step. Inserting into the table a new employee 10 who works for department 20 and earns a commission that is equal to the commission earned by employee 3 is correct answer also. To create a table that has the same structure as subset of columns from another table it’s required to use a subquery in order to perform in a single step.

Question: 112. (E)

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

&

ACCEPT

PROMPT

&&

Explanation:

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

Question: 113. (E)

Which SQL statement accepts user input for the columns to be displayed, the table name, and WHERE condition?

SELECT &1, "&2" FROM &3

WHERE last\_name = '&4';

SELECT &1, '&2' FROM &3

WHERE '&last\_name = '&4' ';

SELECT &1, &2 FROM &3

WHERE last\_name = '&4';

SELECT &1, '&2' FROM EMP

WHERE last\_name = '&4';

Explanation :

In a WHERE clause, date and characters values must be enclosed within single quotation marks. Sample of the correct syntax

SELECT EMPLOYEE\_ID, &COLUMN\_NAME FROM EMPLOYEES

Question: 114. (E)

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?

DELETE FROM employees

WHERE employee\_id = (SELECT employee\_id FROM employees);

DELETE \* FROM employees

WHERE employee\_id = (SELECT employee\_id FROM new\_employees);

DELETE FROM employees

WHERE employee\_id IN (SELECT employee\_id

FROM new\_employees

WHERE name = 'Carrey');

D. DELETE \* FROM employees

WHERE employee\_id IN (SELECT employee\_id

FROM new\_employees

WHERE last\_name = 'Carrey');

Explanation :

The correct syntax for DELETE statement DELETE [ FROM ] table

[ WHERE condition ];

Question: 115. (E)

Evaluate this SQL\*Plus command:

COLUMN teacher\_name HEADING 'Teacher' FORMAT A25

Which two tasks will this command accomplish? (Choose two.)

It will set the TEACHER\_NAME column heading to 'Teacher'.

It will center the column heading of the TEACHER\_NAME column.

It will limit the TEACHER\_NAME column heading to 25 characters.

It will display the current settings for the TEACHER\_NAME column.

It will set the display width of the TEACHER\_NAME column to 25.

Explanation:

COLUMN command is used to change the display format of the column result value. As for the above statement the heading for TEACHER\_NAME will become ‘Teacher’ and the width for the this column will be 25.

Question: 116. (E)

The STUDENT table contains these columns:

ID NUMBER(9) PK

LAST\_NAME VARCHAR2(25)

FIRST\_NAME VARCHAR2(25)

STU\_TYPE VARCHAR2(1)

ENROLL\_DATE DATE

Evaluate this iSQL\*Plus script:

SELECT id, last\_name, first\_name, enroll\_date FROM student

WHERE stu\_type = '&stype'

AND enroll\_date between &&bdate and &&edate

/

What happens when this script is run a second time in the same session?

The user is prompted for all three substitution variables.

An error occurs because the STYPE substitution variable is NOT correctly specified.

The values of BDATE and EDATE from the previous execution are used.

An error occurs because the BDATE and EDATE substitution variables are NOT correctly specified.

Explanation:

Use the double ampersand (&&) substitution variable if you want to reuse the variable value with out prompting the user each time. The user will see the prompt for the value only once and the values will be used from the previous execution. In the above statement the values of BDATE and EDATE from the previous execution are used.

Question: 117. (E)

Evaluate this SELECT statement:

SELECT order\_num, &order\_date

FROM &&ordertbl

WHERE order\_date = '&order\_date';

Which statement regarding the execution of this statement is true?

The user will be prompted for all values in the select list each time the statement is executed in a session.

The user will be prompted for the table name each time the statement is executed in a session.

An error will occur when executing this statement because substitution variables are NOT allowed in a WHERE clause.

The user will be prompted for the table name only the first time the statement is executed in a session.

An error will occur when executing this statement because substitution variables must be unique within a SELECT statement.

Explanation:

When double ampersand (&&) substitution variable is used then user will only be prompted for once and the same value will be used again it will not ask you for a value for substitution variable again.

Question: 118. (E)

The INVENTORY table contains these columns:

ID\_NUMBER NUMBER PK

DESCRIPTION VARCHAR2(30)

SUPPLIER\_ID NUMBER

You want to create a query that for each session allows the user to input a value for DESCRIPTION each time the query runs. While the DESCRIPTION column is stored in upper case, you want the query to retrieve matching values regardless of the case used when inputting the substitution variable value.

Which SELECT statement should you use?

SELECT id\_number, supplier\_id FROM inventory

WHERE description = UPPER(&description);

SELECT id\_number, supplier\_id FROM inventory

WHERE LOWER(description) = LOWER('&description');

SELECT id\_number, supplier\_id FROM inventory

WHERE LOWER(description) = '&description';

SELECT id\_number, supplier\_id FROM inventory

WHERE description = UPPER('&&description');

Explanation:

To retrieve the matching values regardless of the case used when inputting the substitution variable value use the function LOWER which will change the case of stored value and substitution variable value to compare them.

Question: 119. (E)

In which clauses of a SELECT statement can substitution variables be used?

the SELECT, WHERE, GROUP BY, and ORDER BY clauses, but NOT the FROM clause

the SELECT, FROM, WHERE, and GROUP BY clauses, but NOT the ORDER BY clause

the SELECT and FROM clauses, but NOT the WHERE clause

the SELECT, FROM, and WHERE clauses only

the SELECT, FROM, WHERE, GROUP BY, ORDER BY, and HAVING clauses

Explanation:

Substitution variable can be used in the SELECT, FROM, WHERE, GROUP BY, ORDER BY and HAVING clause.

Question: 120. (E)

What is the default character for specifying substitution variable in select statement?

Ampersand.

Ellipses.

Quotations marks.

Asterik

Explanation:

The ampersand (&) character is used by default to define runtime variable in SQL\*Plus.

Question: 121.(F)

Which four are correct guidelines for naming database tables? (Choose four)

Must begin with either a number or a letter.

Must be 1-30 characters long.

Should not be an Oracle Server reserved word.

Must contain only A-Z, a-z, 0-+, \_, \*, and #.

Must contain only A-Z, a-z, 0-9, \_, $, and #.

Must begin with a letter.

Explanation:

Oracle database object must begin with a letter and can usually be between 1 and 30 characters long, except for databases (which have a maximum of eight characters) and database links (with a maximum of 128 characters). Name cannot be an Oracle Server reserved word. Name must contain only A-Z, a-z, 0-9, \_, $, and #.

Question: 122. (F)

Examine the structure of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Remarks |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| LAST\_NAME | VARCNAR2(30) |  |
| FIRST\_NAME | VARCNAR2(30) |  |
| JOB\_ID | NUMBER |  |
| SAL | NUMBER |  |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column |
| DEPARTMENT\_ID | NUMBER |  |

You need to create an index called NAME\_IDX on the first name and last name fields of the EMPLOYEES table. Which SQL statement would you use to perform this task?

CREATE INDEX NAME\_IDX (first\_name, last\_name);

CREATE INDEX NAME\_IDX (first\_name AND last\_name);

CREATE INDEX NAME\_IDX ON (first\_name, last\_name);

CREATE INDEX NAME\_IDX

ON employees (first\_name AND last\_name);

CREATE INDEX NAME\_IDX

ON employees(first\_name, last\_name);

CREATE INDEX NAME\_IDX

FOR employees(first\_name, last\_name);

Explanation:

Answer E provides correct syntax to create index: CREATE INDEX *index\_name* ON *table\_name*(*list of columns*).

Question: 123. (F)

Which three are DATETIME data types that can be used when specifying column definitions? (Choose three.)

TIMESTAMP

INTERVAL MONTH TO DAY

INTERVAL DAY TO SECOND

INTERVAL YEAR TO MONTH

TIMESTAMP WITH DATABASE TIMEZONE

Explanation:

TIMESTAMP, INTERVAL DAY TO SECOND and INTERVAL YEAR TO MONTH can be used to specify column definition.

Question: 124. (F)

What does the TRUNCATE statement do?

Removes the table

Removes all rows from a table

Shortens the table to 10 rows

Removes all columns from a table

Removes foreign keys from a table

Explanation:

Command TRUNCATE is used to remove all row data from the table, while leaving the definition of the table intact, including the definition of constraints and any associated database objects as indexes, constraints, and triggers on the table.

Question: 125. (F)

Which statement about a table is true?

A table can have up to 10,000 columns.

The size of a table does NOT need to be specified.

A table CANNOT be created while users are using the database.

The structure of a table CANNOT be modified while the table is online.

Explanation:

When creating a table there is no need to specify the table size. Table will get the default settings for it. Tables can be created when the users are using the database and table structure can be modified online.

Question: 126. (F)

The ACCOUNT table contains these columns:

ACCOUNT\_ID NUMBER(12)

FINANCE\_CHARGE NUMBER(7,2)

PREV\_BALANCE NUMBER(7,2)

PAYMENTS NUMBER(7,2)

NEW\_PURCHASES NUMBER(7,2)

You created the ACCOUNT\_ID\_SEQ sequence to generate sequential values for the ACCOUNT\_ID column.

You issue this statement:

ALTER TABLE account

MODIFY (finance\_charge NUMBER(8,2));

Which statement about the ACCOUNT\_ID\_SEQ sequence is true?

The sequence is dropped.

The precision of the sequence is changed.

The sequence is reverted to its minimum value.

The sequence is unchanged.

Explanation:

When the sequence is created then the ALTER TABLE command have no effect on the state of sequence and the sequence will remain unchanged.

Question: 127. (F)

Examine the structure of the EMPLOYEE table.

EMPLOYEE Table

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| EMP\_LNAME | VARCHAR2 (25) |  |
| EMP\_FNAME | VARCHAR2 (25) |  |
| DEPT\_ID | NUMBER | Foreign key to DEPT\_ID column of the DEPARTMENT table |
| JOB\_ID | NUMBER | Foreign key to JOB\_ID column of the JOB table |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column |
| SALARY | NUMBER (9,2) |  |
| HIRE\_DATE | DATE |  |
| DPB | DATE |  |

Which statement should you use to increase the EMP\_LNAME column length to 25 if the column currently contains 3000 records?

You CANNOT increase the width of the EMP\_LNAME column.

ALTER TABLE employee

MODIFY emp\_lname VARCHAR2(25);

ALTER TABLE employee

RENAME emp\_lname VARCHAR2(25);

ALTER employee TABLE

MODIFY COLUMN emp\_lname VARCHAR2(25);

ALTER employee TABLE

MODIFY COLUMN (emp\_lname VARCHAR2(25));

Explanation:

You can change the size of the column but this change will have effect only on the future inserts. To change the size of the column datatype use the command

ALTER TABLE employee

MODIFY emp\_lname VARCHAR2(25);

Question: 128. (F)

Examine the structure of the PO\_DETAIL table.

PO\_DETAIL Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PO-NUM | PO\_LINE\_IN | PRODUCT\_ID | QUANTITY | UNIT\_PRICE |
| 10052 | 1 | 1 | 100 | 10.30 |
| 10052 | 2 | 2 | 100 | 10.00 |
| 10054 | 1 | 1 | 50 | 72.00 |
| 10054 | 2 | 2 | 10 | 10.00 |
| 10054 | 3 | 3 | 10 | 10.00 |

Which statement will permanently remove all the data in, the indexes on, and the structure of the PO\_DETAIL table?

DELETE TABLE po\_detail;

TRUNCATE TABLE po\_detail;

ALTER TABLE po\_detail

SET UNUSED (po\_num, po\_line\_id, product\_id, quantity, unit\_price);

DROP TABLE po\_detail;

Explanation:

DROP TABLE command will completely remove the table structure, its all data all the indexes on the table and constraints will also be removed. Only the view and synonym for the table will remain but they are invalid.

Question: 129.

Examine the structure of the EMPLOYEE table.

EMPLOYEE Table

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| EMP\_LNAME | VARCHAR2 (25) |  |
| EMP\_FNAME | VARCHAR2 (25) |  |
| DEPT\_ID | NUMBER | Foreign key to DEPT\_ID column of the DEPARTMENT |
|  |  | table |
| JOB\_ID | NUMBER | Foreign key to JOB\_ID column of the JOB table |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column |
| SALARY | NUMBER (9,2) |  |
| HIRE\_DATE | DATE |  |
| DPB | DATE |  |

The MGR\_ID column currently contains employee identification numbers, and you need to allow users to include text characters in the identification values. Which statement should you use to implement this?

ALTER employee

MODIFY (mgr\_id VARCHAR2(15));

ALTER TABLE employee

MODIFY (mgr\_id VARCHAR2(15));

ALTER employee TABLE

MODIFY COLUMN (mgr\_id VARCHAR2(15));

ALTER TABLE employee

REPLACE (mgr\_id VARCHAR2(15));

You CANNOT modify the data type of the MGR\_ID column.

Explanation:

Using MODIFY command you can change the size of the column but you cannot change the datatype of a column.

Question: 130. (F)

Which CREATE TABLE statements will fail? (Choose all that apply.)

A. CREATE TABLE time1 (time1 NUMBER(9));

CREATE TABLE date (time\_id NUMBER(9));

CREATE TABLE time (time\_id NUMBER(9));

CREATE TABLE time\* (time\_id NUMBER(9));

CREATE TABLE $time (time\_id NUMBER(9));

CREATE TABLE datetime (time\_id NUMBER(9));

Explanation:

When creating table the name of table must start with character. It can use the A-Z, a-z, 0-9, \_, $,

# for the table name but name must start with character. Also the oracle database reserved cannot be used in the table name.

Question: 131. (F)

Evaluate this statement:

SELECT \*

FROM greg.customer;

If as user Chan you execute this statement successfully, which statement must be true?

The CUSTOMER table exists in user Greg's schema.

The CUSTOMER table was created in the DBA schema.

The CUSTOMER table was created in the public schema.

The CUSTOMER table is accessible to all users.

Explanation:

For the above statement to execute successful you must have access privileges on the table and the table must exist in the greg user schema.

Question: 132. (F)

Logged on as user Ann, you issue these SQL statements:

CREATE TABLE customer ( customer\_id NUMBER, company\_id VARCHAR2(30), contact\_name VARCHAR2(30), contact\_title VARCHAR2(20), address VARCHAR2(30),

city VARCHAR2(25), region VARCHAR2(10),

postal\_code VARCHAR2(20), country\_id NUMBER,

phone VARCHAR2(20), fax VARCHAR2(20),

credit\_limit NUMBER (7,2));

CREATE PUBLIC SYNONYM cust

FOR customer;

GRANT SELECT ON customer TO chan;

Which SELECT statements could user Chan issue to access the CUSTOMER table? (Choose all that apply.)

SELECT \*

FROM ann.customer;

SELECT customer\_id FROM customer;

SELECT customer\_name, phone FROM cust;

SELECT \* FROM ann.cust;

SELECT COUNT(city) FROM chan.cust;

Explanation:

Simplify access to the objects by creating a synonym which is another name for an object. For the above statement user Chan can access ann customer table using synonym as

SELECT customer\_name, phone

FROM cust;

As the user ann grant SELECT privileges to the user chan so chan will also access the table using following command

SELECT customer\_name, phone

FROM cust;

Question: 133. (F)

Evaluate this CREATE TABLE statement:

CREATE TABLE curr\_order (

id NUMBER,

customer\_id NUMBER,

emp\_id NUMBER,

order\_dt TIMESTAMP WITH LOCAL TIME ZONE,

order\_amt NUMBER(7,2),

ship\_method VARCHAR2(5));

Which statement about the ORDER\_DT column is true?

Data will be normalized to the database time zone.

Data will include a time zone displacement in its value.

Data stored in the column will be returned in the server's local time zone.

Data will be stored using a fractional seconds precision of 3.

Explanation:

TIMESTAMP WITH LOCAL TIME ZONE is another variant of TIMESTAMP that includes a time zone displacement in its value. It differs from TIMESTAMP WITH TIME ZONE in that data stored in the database is normalized to the database time zone, and the time zone displacement is not stored as part of the column data. When users retrieve the data, Oracle returns it in the users' local ession time zone.

Question: 134. (F)

Examine the structure of the PRODUCT table.

|  |  |  |
| --- | --- | --- |
| PRODUCT\_ID | NUMBER | NOT NULL, Primary Key |
| PRODUCT\_NAME | VARCHAR2 (25) |  |
| SUPPLIER\_ID | NUMBER | Foreign key to SUPPLIER\_ID of the SUPPLIER table |
| LIST\_PRICE | NUMBER (7,2) |  |
| COST | NUMBER (7,2) |  |
| QTY\_IN\_STOCK | NUMBER |  |
| QTY\_ON\_ORDER | NUMBER |  |
| REORDER\_LEVEL | NUMBER |  |
| REORDER\_QTY | NUMBER |  |

You need to reduce the LIST\_PRICE column precision to 6 with a scale of 2 and ensure that when inserting a row into the PRODUCT table without a value for the LIST\_PRICE column, a price of $5.00 will automatically be inserted. The PRODUCT table currently contains no records.

Which statement should you use?

ALTER TABLE product

ADD OR REPLACE (list\_price NUMBER(8,2) DEFAULT 5);

ALTER TABLE product

MODIFY (list\_price NUMBER(6,2) DEFAULT 5);

ALTER TABLE product

MODIFY COLUMN (list\_price NUMBER(6,2) DEFAULT '$5.00');

ALTER TABLE product

MODIFY (list\_price NUMBER(8,2) DEFAULT 5) REPLACE COLUMN (list\_price NUMBER(6,2);

You CANNOT reduce the size of the LIST\_PRICE column.

Explanation:

You can reduce the precision of column only if the column do not have any record in it. To do change the size and also set a default value for the column use the command as.

ALTER TABLE product

MODIFY (list\_price NUMBER(6,2) DEFAULT 5);

Question: 135. (F)

Which statements about data types are true? (Choose all that apply.)

The TIMESTAMP data type is an extension of the VARCHAR2 data type.

The BLOB data type stores character data up to four gigabytes.

The VARCHAR2 data type requires that a minimum size be specified when defining a column of this type.

The CHAR datatype should be used for fixed-length character data.

The INTERVAL YEAR TO MONTH data type allows time to be stored as an interval of years and months.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Explanation:

The CHAR datatype specifies a fixed-length character string. The default length for a CHAR column is 1 byte and the maximum allowed is 2000 bytes.

Question: 136. (F)

Which ALTER TABLE statement should you use to add a PRIMARY KEY constraint on the MANUFACTURER\_ID column of the INVENTORY table?

ALTER TABLE inventory

ADD CONSTRAINT manufacturer\_id PRIMARY KEY;

ALTER TABLE inventory

ADD PRIMARY KEY (manufacturer\_id);

ALTER TABLE inventory

MODIFY manufacturer\_id CONSTRAINT PRIMARY KEY;

ALTER TABLE inventory

MODIFY CONSTRAINT PRIMARY KEY manufacturer\_id;

Explanation:

To add a primary key constraint in the table use the following command

ALTER TABLE inventory

ADD PRIMARY KEY (manufacturer\_id);

Question: 137. (F)

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?

Column definitions cannot be altered to add DEFAULT values.

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

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

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

Explanation:

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

Question: 138. (F)

You need to change the definition of an existing table. The COMMERCIALS table needs its DESCRIPTION column changed to hold varying length characters up to 2000 bytes. The column can currently hold 1000 bytes per value. The table contains 20000 rows.

Which statement is valid?

ALTER TABLE commercials MODIFY (description CHAR2(2000));

ALTER TABLE commercials CHANGE (description CHAR2(2000));

ALTER TABLE commercials

CHANGE (description VARCHAR2(2000));

ALTER TABLE commercials

MODIFY (description VARCHAR2(2000));

You cannot increase the size of a column if the table has rows.

Question: 139. (F)

Evaluate the SQL statement

DROP TABLE DEPT:

Which four statements are true of the SQL statement? (Choose four)

You cannot roll back this statement.

All pending transactions are committed.

Al views based on the DEPT table are deleted.

All indexes based on the DEPT table are dropped.

All data in the table is deleted, and the table structure is also deleted.

All data in the table is deleted, but the structure of the table is retained.

All synonyms based on the DEPT table are deleted.

Explanation:

You cannot roll back DROP TABLE statement. All pending transactions related on this table are committed. If the table is dropped, Oracle automatically drops any index, trigger and constraint associated with the table as well. All data in the table is deleted, and the table structure is also deleted.

Question: 140. (F)

Which statement describes the ROWID data type?

Binary data up to 4 gigabytes.

Character data up to 4 gigabytes.

Raw binary data of variable length up to 2 gigabytes.

Binary data stored in an external file, up to 4 gigabytes.

A hexadecimal string representing the unique address of a row in its table.

Explanation:

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

Question: 141. (F)

You just issued the following statement.

Alter table marketing

Drop column profit;

Which of the following choices identified when the column will actually be removed from database.

Immediately following statement execution.

After the Alter table drop unused columns command is issued.

After the Alter table set unused column command is issued.

After the Alter table modify command is issued.

Explanation:

When drop column command is issued then column will be immediately drop.

Question: 142. (F)

Which of the following can be a valid column name?

Number

1966\_Invoices

Catch\_#22

#Invoices

None of the above

Explanation

Column name must be start with an alphabet.But not the reserved word of the oracle

Question: 143.(G)

You need to modify the STUDENTS table to add a primary key on the STUDENT\_ID column. The table is currently empty.

Which statement accomplishes this task?

ALTER TABLE students

ADD PRIMARY KEY student\_id;

ALTER TABLE students

ADD CONSTRAINT PRIMARY KEY (student\_id);

ALTER TABLE students

ADD CONSTRAINT stud\_id\_pk PRIMARY KEY student\_id;

ALTER TABLE students

ADD CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

ALTER TABLE students

MODIFY CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

Explanation:

This statement provides correct syntax to add a primary key on the STUDENT\_ID column of the STUDENT table.

Question: 144. (G)

Which SQL statement defines the FOREIGN KEY constraint on the DEPTNO column of the EMP table?

CREATE TABLE EMP (empno NUMBER(4), ename VARCNAR2(35),

deptno NUMBER(7,2) NOT NULL

CONSTRAINT emp\_deptno\_fk FOREIGN KEY deptno REFERENCES dept deptno);

CREATE TABLE EMP (empno NUMBER(4), ename VARCNAR2(35), deptno NUMBER(7,2)

CONSTRAINT emp\_deptno\_fk REFERENCES dept (deptno));

CREATE TABLE EMP (empno NUMBER(4) ename VARCHAR2(35),

deptno NUMBER(7,2) NOT NULL,

CONSTRAINT emp\_deptno\_fk REFERENCES dept (deptno) FOREIGN KEY (deptno));

CREATE TABLE EMP (empno NUMBER(4), ename VARCNAR2(35),

deptno NUMBER(7,2) FOREIGN KEY

CONSTRAINT emp deptno fk REFERENCES dept (deptno));

Explanation:

This statement provides correct syntax to define the FOREIGN KEY constraint on the DEPTNO column of the EMP table.

Question: 145. (G)

Which three statements correctly describe the functions and use of constraints? (Choose three.)

Constraints provide data independence.

Constraints make complex queries easy.

Constraints enforce rules at the view level.

Constraints enforce rules at the table level.

Constraints prevent the deletion of a table if there are dependencies.

F. Constraints prevent the deletion of an index if there are dependencies.

Explanation:

Constraints have functions to enforce rules at the view and table levels and to prevent the deletion of data if dependencies exist between tables.

Question: 146. (G)

Which two statements about NOT NULL constraints are true? (Choose two.)

NOT NULL constraints can only be defined at the column level.

You CANNOT define a NOT NULL column if the column does NOT have a non-null value for every row.

You CANNOT add a NOT NULL constraint to an existing column using the ALTER TABLE statement.

You can modify the structure of a NOT NULL constraint using the ALTER TABLE statement.

A NOT NULL constraint is stored in the data dictionary as a UNIQUE constraint.

Explanation:

NOT NULL constraints cannot be define at table level they can be define at column level. To define a NOT NULL constraint for a column the column must not have any null values in it.

Question: 147. (G)

The PO\_DETAIL table contains these columns:

PO\_NUM NUMBER NOT NULL, Primary Key PO\_LINE\_ID NUMBER NOT NULL, Primary Key

PRODUCT\_ID NUMBER Foreign Key to PRODUCT\_ID column of the PRODUCT table QUANTITY NUMBER

UNIT\_PRICE NUMBER(5,2)

Evaluate this statement:

ALTER TABLE po\_detail

ENABLE CONSTRAINT po\_num\_pk;

For which task would you issue this statement?

to drop and recreate the PRIMARY KEY constraint on the PO\_NUM column

to activate the previously disabled constraint on the PO\_NUM column while creating a PRIMARY KEY index

to create a new PRIMARY KEY constraint on the PO\_NUM column

to enable any previously disabled FOREIGN KEY constraints that are dependent on the PO\_NUM column

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Explanation:

To activate the previously disable constraints you can use the ENABLE clause in the ALTER TABLE command.

ALTER TABLE po\_detail

ENABLE CONSTRAINT po\_num\_pk;

Question: 148. (G)

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

Constraints only enforce rules at the table level.

Constraints prevent a table with dependencies from being deleted.

Constraints can be created at the same time as the table or after the table is created.

You must provide a name for each constraint at the time of its creation.

Constraint names are NOT required to follow the standard object-naming rules.

Explanation:

Constraints can be created at the table level and column level. They can be created at the time of table creation or after the table is created. You cannot delete data from a table if there are dependencies on the table. If you do not provide a name to the constraint then the oracle will assign the name to constraint..

Question: 149. (G)

Examine the structure of the EMPLOYEE table.

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| EMP\_LNAME | VARCHAR2 |  |
|  | (25) |  |
| EMP\_FNAME | VARCHAR2 |  |
|  | (25) |  |
| DEPT\_ID | NUMBER | Foreign key to DEPT\_ID column of the |
|  |  | DEPARTMENT table |
| JOB\_ID | NUMBER | Foreign key to JOB\_ID column of the JOB table |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column |
| SALARY | NUMBER (9,2) |  |
| HIRE\_DATE | DATE |  |

Which CREATE TABLE statement should you use to create the EMPLOYEE table?

CREATE TABLE employee ( employee\_id NUMBER, emp\_lname VARCHAR2(25), emp\_fname VARCHAR2(25), dept\_id NUMBER,

job\_id NUMBER, mgr\_id NUMBER, salary NUMBER(9,2), hire\_date DATE,

CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),

CONSTRAINT dept\_id\_fk FOREIGN KEY(dept\_id) REFERENCES department(dept\_id), CONSTRAINT job\_id\_fk FOREIGN KEY(job\_id) REFERENCES job(job\_id), CONSTRAINT mgr\_id\_fk FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id));

CREATE TABLE employee ( employee\_id NUMBER, emp\_lname VARCHAR2(25), emp\_fname VARCHAR2(25), dept\_id NUMBER,

job\_id NUMBER, mgr\_id NUMBER, salary NUMBER(9,2), hire\_date DATE,

CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id));

CREATE TABLE employee ( employee\_id NUMBER, emp\_lname VARCHAR2(25), emp\_fname VARCHAR2(25), dept\_id NUMBER,

job\_id NUMBER, mgr\_id NUMBER, salary NUMBER(9,2), hire\_date DATE,

CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),

CONSTRAINT dept\_id\_fk FOREIGN KEY(dept\_id) REFERENCES department(dept\_id), CONSTRAINT job\_id\_fk FOREIGN KEY(job\_id) REFERENCES job(job\_id));

CREATE TABLE employee ( employee\_id NUMBER, emp\_lname VARCHAR2(25), emp\_fname VARCHAR2(25), dept\_id NUMBER,

job\_id NUMBER, mgr\_id NUMBER, salary NUMBER(9,2), hire\_date DATE,

CONS TRAINT employee\_id\_pk PRIMARY KEY(employee\_id),

CONSTRAINT mgr\_id\_fk FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id));

Explanation:

Employee Table will all constraints can be created by the following command. CREATE TABLE employee (

employee\_id NUMBER, emp\_lname VARCHAR2(25), emp\_fname VARCHAR2(25), dept\_id NUMBER,

job\_id NUMBER, mgr\_id NUMBER, salary NUMBER(9,2), hire\_date DATE,

CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),

CONSTRAINT dept\_id\_fk FOREIGN KEY(dept\_id) REFERENCES department(dept\_id), CONSTRAINT job\_id\_fk FOREIGN KEY(job\_id) REFERENCES job(job\_id), CONSTRAINT mgr\_id\_fk FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id));

Question: 150. (G)

Which syntax turns an existing constraint on?

ALTER TABLE table\_name ENABLE constraint\_name;

ALTER TABLE table\_name

STATUS = ENABLE CONSTRAINT constraint\_name;

ALTER TABLE table\_name

ENABLE CONSTRAINT constraint\_name;

ALTER TABLE table\_name

STATUS ENABLE CONSTRAINT constraint\_name;

ALTER TABLE table\_name

TURN ON CONS TRAINT constraint\_name;

ALTER TABLE table\_name

TURN ON CONSTRAINT constraint\_name;

Explanation:

ALTER TABLE statement with ENABLE CONSTRAINT keywords is correct answer to enable an existing constraint.

Question: 151. (G)

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

Constraint names must start with SYS\_C.

All constraints must be defines at the column level.

Constraints can be created after the table is created.

Constraints can be created at the same time the table is created.

Information about constraints is found in the VIEW\_CONSTRAINTS dictionary view.

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).

Question: 152. (G)

Which constraint can be defines only at the column level?

UNIQUE

NOT NULL

CHECK

PRIMARY KEY

FOREIGN KEY

Explanation:

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

Question: 153. (G)

Which statement explicitly names a constraint?

ALTER TABLE student\_grades ADD

FOREIGN KEY (student\_id) REFERENCES students(student\_id);

ALTER TABLE student\_grades

ADD CONSTRAINT NAME = student\_id\_fk

FOREIGN KEY (student\_id) REFERENCES students(student\_id);

ALTER TABLE student\_grades ADD CONSTRAINT student\_id\_fk

FOREIGN KEY (student\_id) REFERENCES students(student\_id);

ALTER TABLE student grades

ADD NAMED CONSTRAINT student\_id\_fk

FOREIGN KEY (student\_id) REFERENCES students(student\_id);

ALTER TABLE student grades ADD NAME student\_id\_fk

FOREIGN KEY (student\_id) REFERENCES students(student\_id);

Explanation:

This statement provides correct syntax to add a foreign key constraint to the existing table.

Question: 154. (G)

Examine the SQL statements that creates ORDERS table:

CREATE TABLE orders (SER\_NO NUMBER UNIQUE, ORDER\_ID NUMBER, ORDER\_DATE DATE NOT NULL STATUS VARCHARD2(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)

SER\_NO

ORDER\_ID

STATUS

PROD\_ID.

ORD\_TOTAL

Composite index on ORDER\_ID and ORDER\_DATE

Explanation:

Indexes are created automatically by Oracle to support integrity constraints that enforce uniqueness. The two types of integrity constraints that enforce uniqueness are PRIMARY KEY and UNIQUE constraints. When the primary key or UNIQUE constraint is declared, a unique index to support the column’s uniqueness is also created, and all values in all columns that were defined as part of the primary key or UNIQUE constraint are placed into the index.

Question: 155. (G)

For which two constraints does the Oracle Server implicitly create a unique index? (Choose two.)

NOT NULL.

PRIMARY KEY

FOREIGN KEY

CHECK

UNIQUE

Explanation:

Indexes are created automatically by Oracle to support integrity constraints that enforce uniqueness. The two types of integrity constraints that enforce uniqueness are PRIMARY KEY and UNIQUE constraints. When the primary key or UNIQUE constraint is declared, a unique index to support the column’s uniqueness is also created, and all values in all columns that were defined as part of the primary key or UNIQUE constraint are placed into the index.

Question: 156. (G)

Your attempt to disable a constraints result in the following error

Ora:02297: cannot disable constraint – dependencies exist.

Which of the following types of the constraints is likely causing interference with your disablement of this one?

Check constraints.

Not NULL constraints.

Foreign key Constraints.

Unique Constraints.

Explanation:

When you attempt to disable a primary key when existing foreign keys depend on that primary key. Then you will get this above shown error. To disable the column with primary key you have to use an additional option CASCADE within the command.

Question: 157.(H)

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

EMPLOYEES

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | Primary Key |
| FIRST\_NAME | VARCHARD2(25) |  |
| LAST\_NAME | VARCHARD2(25) |  |
| HIRE\_DATE | DATE |  |

NEW EMPLOYEES

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | |  |  |
| EMPLOYEE\_ID |  | NUMBER | Primary Key | |  |  |
| NAME |  | VARCHAR2(60) |  |  |  |  |

Which UPDATE statement is valid?

UPDATE new\_employees SET name = (Select last\_name||. first\_name

FROM employees

Where employee\_id =180) WHERE employee\_id =180;

UPDATE new\_employees SET name = (SELECT last\_name||first\_name

FROM employees) WHERE employee\_id =180;

UPDATE new\_employees SET name = (SELECT last\_name|| first\_name

FROM employees

WHERE employee\_id =180)

WHERE employee\_id =(SELECT employee\_id FROM new employees);

UPDATE new\_employees SET name = (SELECT last name|| first\_name

FROM employees WHERE employee\_id= (SELECT employee\_id FROM new\_employees))

WHERE employee\_id =180;

Explanation:

Sub-query in this answer will return one row value, concatenated first and last name for the employee with ID 180, so update will be successful. When sub-queries are linked to the parent by equality comparisons, the parent query expects only one row of data from the subquery.

Question: 158. (H)

Examine the structure if 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?

MERGE INTO new\_employees c USING employees e

ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN

UPDATE SET

c.name = e.first\_name ||’,’|| e.last\_name WHEN NOT MATCHED THEN

INSERT VALUES(e.employee\_id, e.first\_name ||’, ‘||e.last\_name);

MERGE new\_employees c USING employees e

ON (c.employee\_id = e.employee\_id) WHEN EXIST THEN

UPDATE SET

c.name = e.first\_name ||’,’|| e.last\_name WHEN NOT MATCHED THEN

INSERT VALUES(e.employee\_id, e.first\_name ||’,. ‘||e.last\_name);

MERGE INTO new employees c USING employees e

ON (c.employee\_id = e.employee\_id) WHEN EXISTS THEN

UPDATE SET

c.name = e.first\_name ||’,’|| e.last\_name WHEN NOT MATCHED THEN

INSERT VALUES(e.employee\_id, e.first\_name ||’, ‘||e.last\_name);

MERGE new\_employees c FROM employees e

ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN

UPDATE SET

c.name = e.first\_name ||’,’|| e.last\_name WHEN NOT MATCHED THEN

INSERT INTO new\_employees VALUES(e.employee\_id, e.first\_name ||’.’||e.last\_name);

Explanation:

Correct syntax for the MERGE command is MERGE INTO *table1* USING *table2* on (*join\_condition*) WHEN MATCHED UPDATE SET col1 = *value* WHEN NOT MATCHED INSERT (*column\_list*) values (*column\_values*).

Question: 159. (H)

Examine the data in the EMPLOYEES and EMP\_HIST tables:

EMPLOYEES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | 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 |

EMP\_HIST

|  |  |  |  |
| --- | --- | --- | --- |
| EMPLOYEE\_ID | NAME | JOB\_ID | SALARY |
| 101 | Smith | SA\_CLERK | 2000 |
| 103 | Chris | IT\_CLERK | 2200 |
| 104 | John | HR\_CLERK | 2000 |
| 106 | Smith | AD\_ASST | 3000 |
| 108 | Jennifer | HR\_MGR | 4500 |

The EMP\_HIST table is updated at the end of every year. The employee ID, name, job ID, and salary of each existing employee are modified with the latest data. New employee details are added to the table.

Which statement accomplishes this task?

UPDA TE emp\_hist

SET employee\_id, name, job\_id, salary = (SELECT employee\_id, name, job\_id, salary FROM employees)

WHERE employee\_id IN (SELECT employee\_id FROM employees);

MERGE INTO emp\_hist eh USING employees e

ON (eh.employee\_id = e.employee\_id) WHEN MATCHED THEN

UPDATE SET eh.name = e.name, eh.job\_id = e.job\_id,

eh.salary = e.salary

WHEN NOT MATCHED THEN

INSERT VALUES (e.employee id, e.name, e.job id, e.salary);

MERGE INTO emp\_hist eh USING employees e

ON (eh.employee\_id = e.employee\_id) WHEN MATCHED THEN

UPDATE emp hist SET eh.name = e.name, eh.job\_id = e.job\_id,

eh.salary = e.salary

WHEN NOT MATCHED THEN INSERT INTO emp\_hist

VALUES (e.employee\_id, e.name, e.job\_id, e.salary);

MERGE INTO emp\_hist eh USING employees e. WHEN MATCHED THEN

UPDATE emp\_hist SET eh.name = e.name, eh.job\_id = e.job\_id,

eh.salary = e.salary

WHEN NOT MATCHED THEN INSERT INTO emp\_hist

VALUES (e.employee\_id, e.name, e.job\_id, e.salary);

Answer: B

Explanation:

This task can be done using the MERGE command. Correct syntax for the MERGE command is MERGE INTO *table1* USING *table2* on (*join\_condition*) WHEN MATCHED UPDATE

SET col1 = *value* WHEN NOT MATCHED INSERT (*column\_list*) values (*column\_values*).

Question: 161. (H)

Examine the data in the EMPLOYEES and DEPARTMENTS tables:

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 | 110 | EX\_DIR | 8000 |
| 120 | Ravi | 20 | 110 | SA\*DIR | 6500 |

DEPARTMENTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| DEPARTMENT\_ID | | DEPARTMENT\_NAME |  |  |  |
| 10 |  | Admin |  |  |  |
| 20 |  | Education |  |  |  |
| 30 |  | IT |  |  |  |
| 40 |  | Human Resources |  |  |  |

Also examine the SQL statements that create the EMPLOYEES and DEPARTMENTS tables:

CREATE TABLE departments (department\_id NUMBER PRIMARY KEY, department\_name VARCHAR2(30)); CREATE TABLE employees (EMPLOYEE\_ID NUMBER PRIMARY KEY, EMP\_NAME VARCHAR2(20),

DEPT\_ID NUMBER REFERENCES departments(department\_id), MGR\_ID NUMBER REFERENCES employees(employee id),

MGR\_ID NUMBER REFERENCES employees(employee id),

JOB\_ID VARCHAR2(15). SALARY NUMBER);

ON the EMPLOYEES,

On the EMPLOYEES table, EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID.

DEPT\_ID is foreign key to DEPARTMENT\_ID column of the DEPARTMENTS table. On the DEPARTMENTS table, DEPARTMENT\_ID is the primary key.

Examine this DELETE statement:

DELETE

FROM departments

WHERE department id = 40;

What happens when you execute the DELETE statement?

Only the row with department ID 40 is deleted in the DEPARTMENTS table.

The statement fails because there are child records in the EMPLOYEES table with department ID 40.

The row with department ID 40 is deleted in the DEPARTMENTS table. Also the rows with employee IDs 110 and 106 are deleted from the EMPLOYEES table.

The row with department ID 40 is deleted in the DEPARTMENTS table. Also the rows with employee IDs 106 and 110 and the employees working under employee 110 are deleted from the EMPLOYEES table.

The row with department ID 40 is deleted in the DEPARTMENTS table. Also all the rows in the EMPLOYEES table are deleted.

The statement fails because there are no columns specifies in the DELETE clause of the DELETE statement.

Explanation:

It will be error generated because there are 2 child records in the EMPLOYEES table with department number you try to delete from the DEPARTMENTS table.

Question: 162. (H)

You maintain two tables, CUSTOMER and PROSPECT, that have identical structures but different data. You want to synchronize these two tables by inserting records from the PROSPECT table into the CUSTOMER table, if they do not exist. If the customer already exists in the CUSTOMER table, you want to update customer data.

Which DML statement should you use to perform this task?

INSERT

UPDATE

SYNC

MERGE

You CANNOT perform this task with one DML operation.

Answer: D

Explanation:

MERGE command can be use to synchronize the two tables. when this command is used then if the data already exist then it will be updated and if does not exist then these two tables will be synchronize by inserting the data from one table to the other.

Question: 163. (H)

Examine the structure of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | NOT NULL |
| EMP\_NAME | VARCHAR2(30) |  |
| JOB\_ID | VARCHAR2(20) | DEFAULT 'SA\_REP' |
| SAL | NUMBER |  |
| COMM\_PCT | NUMBER |  |
| MGR\_ID | NUMBER. |  |
| DEPARTMENT\_ID | NUMBER |  |

You need to update the records of employees 103 and 115. The UPDATE statement you specify should update the rows with the values specified below:

JOB\_ID: Default value specified for this column definition.

SAL: Maximum salary earned for the job ID SA\_REP.

COMM\_PCT: Default value specified for this commission percentage column, if any.

If no default value is specified for the column, the value should be NULL.

DEPARTMENT\_ID: Supplied by the user during run time through substitution variable.

Which UPDATE statement meets the requirements?

UPDATE employees SET job\_id = DEFAULT

AND Sal = (SELECT MAX(sal) FROM employees

WHERE job\_id = 'SA\_REP' AND comm\_pct = DEFAULT AND department\_id = &did

WHERE employee\_id IN (103,115);

UPDATE employees SET job\_id = DEFAULT AND Sal = MAX(sal)

AND comm\_pct = DEFAULT OR NULL AND department\_id = &did

WHERE employee\_id IN (103,115) AND job\_id = 'SA\_REP';

UPDATE employees SET job\_id = DEFAULT, Sal = (SELECT MAX(sal) FROM employees

WHERE job\_id = 'SA\_REP'), comm\_pct = DEFAULT, department\_id = &did

WHERE employee\_id IN (103,115);

UPDATE employees SET job\_id = DEFAULT, Sal = MAX(sal), comm\_pct = DEFAULT, department\_id = &did

WHERE employee\_id IN (103,115) AND job\_id = 'SA\_REP';

UPDATE employees. SET job\_id = DEFAULT, Sal = (SELECT MAX(sal) FROM employees

WHERE job\_id = 'SA\_REP') comm\_pct = DEFAULT OR NULL, department\_id = &did

WHERE employee\_id IN (103,115);

Answer: C

Explanation:

This UPDATE statement is correct to receive desired results. Correct syntax is UPDATE *table\_name* SET *column\_name1* = value, *column\_name2* = value2. You can also use theDEFAULT keyword to set a column value to its specified default value in update statements as well.

Question: 164. (H)

You added a PHONE-NUMBER column of NUMBER data type to an existing EMPLOYEES table. The EMPLOYEES table already contains records of 100 employees. Now, you want to enter the phone numbers of each of the 100 employees into the table.

Some of the employees may not have a phone number available.

Which data manipulation operation do you perform?

MERGE

INSERT

UPDATE

ADD

ENTER

You cannot enter the phone numbers for the existing employee records.

Answer: C

Explanation:

To update information you need to use UPDATE command.

Question: 165. (H)

Which two statements complete a transaction? (Choose two)

DELETE employees;

DESCRIBE employees;

ROLLBACK TO SAVEPOINT C;

GRANT SELECT ON employees TO SCOTT;

ALTER TABLE employees SET UNUSED COLUMN sal;

Select MAX(sal) FROM employees

WHERE department\_id = 20;

, E

Explanation:

It is important to understand that an implicit COMMIT occurs on the database when a user exits SQL\*Plus or issues a data-definition language (DDL) command such as a CREATE TABLE statement, used to create a database object, or an ALTER TABLE statement, used to alter a database object. Also after the ROLLBACK command is issued, a new transaction is started implicitly by the database session.

Question: 166. (H)

Examine the data in the EMPLOYEES table.

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 |  | EK\_DIR | 8000 |
| 120 | Revi | 20 | 110 | SA\_DIR | 6500 |

On the EMPLOYEES table, EMPLOYEE\_ID is the primary key. MGR\_ID is the ID of managers and refers to the EMPLOYEE\_ID. The JOB\_ID column is a NOT NULL column.

Evaluate this DELETE statement:

DELETE employee\_id, salary, job\_id

FROM employees

WHERE dept\_id = 90;

Why does the DELETE statement fail when you execute it?

There is no row with dept\_id 90 in the EMPLOYEES table.

You cannot delete the JOB\_ID column because it is a NOT NULL column.

You cannot specify column names in the DELETE clause of the DELETE statement.

You cannot delete the EMPLOYEE\_ID column because it is the primary key of the table.

Answer: C

Explanation:

You cannot specify column names in the DELETE clause of the DELETE statement. Syntax of this command is: DELETE FROM *table\_name* WHERE *column\_name* = value.

Question: 167. (H)

Examine the structure of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | Primary Key |
| FIRST\_NAME | VARCNAR2(25) |  |
| LAST\_NAME | VARCNAR2(25) |  |

Which three statements inserts a row into the table? (Choose three)

INSERT INTO employees VALUES ( NULL, ‘John’,‘Smith’);

INSERT INTO employees( first\_name, last\_name) VALUES(‘John’,‘Smith’);

INSERT INTO employees VALUES (‘1000’,‘John’,NULL);

INSERT INTO employees(first\_name,last\_name, employee\_id) VALUES ( 1000, ‘John’,‘Smith’);

INSERT INTO employees (employee\_id) VALUES (1000);

INSERT INTO employees (employee\_id, first\_name, last\_name) VALUES ( 1000, ‘John’,‘’);

, E, F

Explanation:

Since EMPLOYEE\_ID column is used as primary key, it cannot be NULL, so only INSERT statements in C, E and F are correct. You can insert the row with NULL LAST\_NAME as in answer C, or only the row with EMPLOYEE\_ID as in answer E, or the row with empty LAST\_NAME column.

Question: 168. (H)

Examine the structure of the EMPLOYEES table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | | NUMBER | Primary Key | |
| FIRST\_NAME | | VARCHAR2(25) |  | |
|  |  |  |  |  |  |  |
| LAST\_NAME |  | VARCHAR2(25) |  |  |  |  |
| HIRE\_DATE |  | DATE |  |  |  |  |

You issue these statements:

CREATE table new\_emp ( employee\_id NUMBER, name VARCHAR2(30)); INSERT INTO new\_emp SELECT employee\_id , last\_name from employees;

Savepoint s1;

UPDATE new\_emp set name = UPPER(name); Savepoint s2;

Delete from new\_emp; Rollback to s2;

Delete from new\_emp where employee\_id =180; UPDATE new\_emp set name = 'James'; Rollback to s2;

UPDATE new\_emp set name = 'James' WHERE employee\_id =180; Rollback;

At the end of this transaction, what is true?

You have no rows in the table.

You have an employee with the name of James.

You cannot roll back to the same savepoint more than once.

Your last update fails to update any rows because employee ID 180 was already deleted.

Answer: A

Explanation:

At the end of this transaction you will not have rows in the table.

Question: 169. (H)

Examine the data from the CLASS and INSTRUCTOR tables.

*CLASS Table*

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS\_ID | CLASS\_NAME | HOURS\_CREDIT | INSTRUCTOR\_ID |
| 1 | Introduction to Accounting | 3 | 4 |
| 2 | Computer Basics | 3 | 1 |
| 3 | Tax Accounting Principles | 3 | 4 |
| 4 | American History | 3 | 2 |
| 5 | Basic Engineering | 3 |  |

*INSTRUCTOR Table*

|  |  |  |  |
| --- | --- | --- | --- |
| INSTRUCTOR\_ID | | LAST\_NAME | FIRST\_NAME |
| 1 | | Chao | Ling |
| 2 | | Vanderbilt | Herbert |
|  |  |  |  |  |  |
| 3 |  | Wigley | Martha |  |  |
| 4 |  | Page | Albert |  |  |

You want to delete the classes that do NOT have an instructor assigned. Which DELETE statement will accomplish the desired result?

DELETE class\_id, class\_name, hours\_credit, instructor\_id FROM class

WHERE instructor\_id IS NULL;

DELETE FROM class WHERE instructor\_id NOT IN (SELECT instructor\_id

FROM class);

DELETE FROM

instructor NATURAL JOIN class

WHERE instructor\_id IS NOT NULL;

DELETE FROM class WHERE instructor\_id IS NULL;

Answer: D

Explanation:

To delete a record from a table on the basis of null values you can use the IS NULL operator in the condition in delete statement as

DELETE FROM class WHERE instructor\_id IS NULL;

Question: 170. (H)

Which action will cause an automatic rollback?

GRANT statement

CREATE statement

System crash

exiting from iSQL\*Plus without first committing the changes

subsequent DML statement

SAVEPOINT statement

Answer: C

Explanation:

Automatic roll back occur under an abnormal termination of SQL Plus and when the system crashes or machines fails.

Question: 171. (H)

The PRODUCT table contains these columns: PRODUCT\_ID NUMBER NOT NULL PRODUCT\_ NAME VARCHAR2(25) SUPPLIER\_ID NUMBER

LIST\_PRICE NUMBER(7,2) COST NUMBER(7,2)

You need to increase the list price and cost of all products supplied by Global Imports, Inc. by 5.5 percent. The SUPPLIER\_ID for Global Imports, Inc. is 105. Which statement should you use?

UPDATE product

SET list\_price = list\_price \* 1.055 SET cost = cost \* 1.055

WHERE supplier\_id = 105;

UPDATE product

SET list\_price = list\_price \* .055 AND

cost = cost \* .055

WHERE supplier\_id = 105;

UPDATE product

SET list\_price = list\_price \* 1.055, cost = cost \* 1.055 WHERE supplier\_id = 105;

UPDATE product

SET list\_price = list\_price + (list\_price \* .055), cost = cost + (cost \* .055) WHERE supplier\_id LIKE 'Global Imports, Inc.'

OR supplier\_id = 105;

UPDATE product

SET list\_price = list\_price + (list\_price \* .055), cost = cost + (cost \* .055) WHERE supplier\_id LIKE 'Global Imports, Inc.';

Answer: C

Explanation:

Using UPDATE command you can change the data of a table. To increase the value of list price and cost of all products supplied by Global imports, Inc by 5.5 percent and the SUPPLIER\_ID for global imports. Inc is 105 use the following command

UPDATE product

SET list\_price = list\_price \* 1.055, cost = cost \* 1.055 WHERE supplier\_id = 105;

Question: 172. (H)

Which two statements would cause an implicit COMMIT to occur? (Choose two.)

GRANT

UPDATE

COMMIT

SELECT

ROLLBACK

RENAME

& F

Explanation:

All DDL commands such as CREATE, ALTER, DROP and all DCL commands such GRANT and REVOKE cause an implicit commit. RENAME command also change the definition of object so this is will also cause an implicit commit.

Question: 173. (H)

Examine the structures of the EVENT and NEW\_EVENT tables.

EVENT Table

|  |  |
| --- | --- |
| EVENT\_ID | NUMBER |
| EVENT\_NAME | VARCHAR2 (30) |
| EVENT\_DESCRIPTION | VARCHAR2 (100) |
| EVENT\_TYPE\_ID | NUMBER |
| NEW\_EVENT Table |  |
| EVENT\_ID | NUMBER |
| EVENT\_NAME | VARCHAR2 (30) |
| EVENT\_DESCRIPTION | VARCHAR2 (100) |
| EVENT\_TYPE\_ID | NUMBER |
| START\_DT | DATE |

You execute this MERGE statement:

MERGE INTO EVENT e

USING (SELECT \* FROM new\_event

WHERE event\_type\_id = 4) n ON (e.event\_id = n.event\_id) WHEN MATCHED THEN UPDATE SET

e.event\_type\_id = n.event\_type\_id, e.start\_dt = n.start\_dt

WHEN NOT MATCHED THEN

INSERT (event\_id, event\_name, event\_type\_id) VALUES(n.event\_id, n.event\_name, n.event\_type\_id);

This MERGE statement generates an error. Which statement describes the cause of the error?

A subquery CANNOT be used in the USING clause of a MERGE statement.

Table aliases CANNOT be used in a MERGE statement.

The ON clause of the statement is invalid.

The UPDATE portion of the statement is invalid.

The INSERT portion of the statement is invalid.

A MERGE statement CANNOT be used with tables that do NOT have an identical structure.

Answer: D

Explanation:

Use the MERGE statement to select rows from one table for update or insertion into another table. The decision whether to update or insert into the target table is based on a condition in the ON clause. In the above written statement the update portion contains invalid column name as no STA RT\_ID column is in the EVENT table.

Question: 174. (H)

Which statement regarding DML statement functionality is true?

DELETE can be used to delete rows or columns from a table.

MERGE will delete rows that do NOT exist in either table.

UPDATE will add rows to a table if an INTO clause is specified.

UPDATE can update multiple columns in one table.

INSERT must contain a VALUES clause.

Answer: D

Explanation:

UPDATE commands can update multiple columns in one table by using comma (,) between the columns in a single statement.

Question: 175. (H)

The STUDENT table contains these columns:

STU\_ID NUMBER(9) NOT NULL

LAST\_NAME VARCHAR2(30) NOT NULL

FIRST\_NAME VARCHAR2(25) NOT NULL

DOB DATE

STU\_TYPE\_ID VARCHAR2(1) NOT NULL

ENROLL\_DATE DATE

You create another table, named PT\_STUDENT, with an identical structure. You want to insert all part-time students, who have a STU\_TYPE\_ID value of "P", into the new table. You execute this INSERT statement:

INSERT INTO pt\_student

(SELECT stu\_id, last\_name, first\_name, dob, sysdate FROM student

WHERE UPPER(stu\_type\_id) = 'P');

What is the result of executing this INSERT statement?

All part-time students are inserted into the PT\_STUDENT table.

An error occurs because the PT\_STUDENT table already exists.

An error occurs because you CANNOT use a subquery in an INSERT statement.

An error occurs because the STU\_TYPE\_ID column is NOT included in the subquery select list.

An error occurs because both the STU\_TYPE\_ID and ENROLL\_DATE columns are NOT included in the subquery select list.

An error occurs because the INSERT statement does NOT contain a VALUES clause.

Answer: D

Explanation:

When using subquery to insert data into the table do not use the VALUES clause and the number of columns in the INSERT clause must match to number of column in the subquery. In the above written statement STU\_TYPE\_ID column is in the pt\_student table but is missing in the subquery select list so this will cause an error.

Question: 176. (H)

Examine the structures of the AR\_TRX and AR\_TRX\_HY tables.

AR\_TRX Table

|  |  |  |
| --- | --- | --- |
| TRX\_ID | NUMBER | NOT NULL, Primary Key |
| TRX\_TYPE | VARCHAR2 (5) |  |
| QUANTITY | NUMBER |  |
| UNIT\_PRICE | NUMBER (7,2) |  |
| EXT\_AMT | NUMBER (9,2) |  |
| TAX\_AMT | NUMBER (7,2) |  |
| AR\_TRX Table |  |  |
| TRX\_ID | NUMBER | NOT NULL, Primary Key |
| TRX\_TYPE | VARCHAR2 (5) |  |
| QUANTITY | NUMBER |  |
| UNIT\_PRICE | NUMBER (7,2) |  |
| EXT\_AMT | NUMBER (9,2) |  |
| TAX\_AMT | NUMBER (7,2) |  |
| GRAND\_TOTAL | NUMBER (10,2) |  |
| LOAD\_DATE | DATE |  |

You are loading historical accounts receivable data from the AR\_TRX table into the AR\_TRX\_HY table. During the load, you want to transform the data so that the GRAND\_TOTAL column of the AR\_TRX\_HY table is equal to the sum of the EXT\_AMT and TAX\_AMT columns in the AR\_TRX table. You want to set LOAD\_DATE to the current date. If the record already exists in the target table, all values, except TRX\_ID and TRX\_TYPE, should be refreshed with the most recent data.

Which MERGE statement should you execute?

MERGE INTO ar\_trx\_hy h USING ar\_trx a

ON (h.trx\_id = a.trx\_id) WHEN MATCHED THEN UPDATE SET

a.quantity = h.quantity, a.unit\_price = h.unit\_price, a.ext\_amt = h.ext\_amt, a.tax\_amt = h.tax\_amt,

a.grand\_total = h.ext\_amt + h.tax\_amt, a.load\_date = sysdate

WHEN NOT MATCHED THEN

INSERT VALUES(a.trx\_id, a.trx\_type, a.quantity, a.unit\_price, a.ext\_amt, a.tax\_amt, a.ext\_amt + a.tax\_amt, sysdate);

MERGE INTO ar\_trx\_hy USING ar\_trx

USING (trx\_id)

WHEN MATCHED THEN UPDATE SET

quantity = quantity, unit\_price = unit\_price, ext\_amt = ext\_amt, tax\_amt = tax\_amt,

grand\_total = ext\_amt + tax\_amt, load\_date = sysdate

WHEN NOT MATCHED THEN

INSERT VALUES(trx\_id, trx\_type, quantity, unit\_price, ext\_amt, tax\_amt, ext\_amt + tax\_amt, sysdate);

MERGE INTO ar\_trx\_hy h USING ar\_trx a

ON (h.trx\_id = a.trx\_id) WHEN MATCHED THEN UPDATE ar\_trx\_hy SET h.quantity = a.quantity, h.unit\_price = a.unit\_price, h.ext\_amt = a.ext\_amt, h.tax\_amt = a.tax\_amt,

h.grand\_total = a.ext\_amt + a.tax\_amt, h.load\_date = sysdate

WHEN NOT MATCHED THEN

INSERT VALUES(a.trx\_id, a.trx\_type, a.quantity, a.unit\_price, a.ext\_amt, a.tax\_amt, a.ext\_amt + a.tax\_amt, sysdate);

MERGE INTO ar\_trx\_hy h USING ar\_trx a

ON (h.trx\_id = a.trx\_id) WHEN MATCHED THEN UPDATE SET

h.quantity = a.quantity, h.unit\_price = a.unit\_price, h.ext\_amt = a.ext\_amt, h.tax\_amt = a.tax\_amt,

h.grand\_total = a.ext\_amt + a.tax\_amt, h.load\_date = sysdate

WHEN NOT MATCHED THEN

INSERT VALUES(a.trx\_id, a.trx\_type, a.quantity, a.unit\_price, a.ext\_amt, a.tax\_amt, a.ext\_amt + a.tax\_amt, sysdate);

MERGE INTO ar\_trx\_hy h USING ar\_trx a

ON (h.trx\_id = a.trx\_id) WHEN MATCHED UPDATE SET h.quantity = a.quantity,

h.unit\_price = a.unit\_price, h.ext\_amt = a.ext\_amt, h.tax\_amt = a.tax\_amt,

h.grand\_total = a.ext\_amt + a.tax\_amt, h.load\_date = sysdate

WHEN NOT MATCHED

INSERT VALUES(a.trx\_id, a.trx\_type, a.quantity, a.unit\_price, a.ext\_amt, a.tax\_amt, a.ext\_amt + a.tax\_amt, sysdate);

Answer: D

Explanation:

Ans E contain the correct syntax for MERGE statement to performs all of the above desired changes.

Question: 177. (H)

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?

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.

The statement executes successfully, changes the manager ID to NULL, and changes the salary to 4000 for the employees with ID 103 and 105.

The statement executes successfully, changes the manager ID to NULL, and changes the salary to 3000 for the employees with ID 103 and 105.

The statement fails because there is more than one row matching the employee name Smith.

The statement fails because there is more than one row matching the IT\_ADMIN job ID in the EMPLOYEES table.

The statement fails because there is no 'Administration' department in the DEPARTMENTS table.

Explanation :

'=' is use in the statement and subquery will return more than one row. Employees table has 2 row matching the employee name Smith.

The update statement will fail.

Question: 178. (H)

The transaction control which prevent more than one user from updating data in a table is which of the following?

Lock.

Commit.

Rollback.

Savepoint.

Explanation:

A LOCK is the mechanism that prevents more than one user at a time from making changes to a database. COMMIT, ROLLBACK & SAVEPOINT are the options which make end of the transaction at different stages in a transaction.

Question: 179.(I)

What is necessary for your query on an existing view to execute successfully?

The underlying tables must have data.

You need SELECT privileges on the view.

The underlying tables must be in the same schema.

You need SELECT privileges only on the underlying tables.

Answer: B

Explanation:

To query an existing view you need to have SELECT privileges on the view.

Question: 180. (I)

Examine the structure if the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Remarks |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| EMP\_NAME | VARCHAR2(30) |  |
| JOB\_ID | VARCHAR2(20) | NOT NULL |
| SAL | NUMBER |  |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column |
|  |  |  |
| DEPARTMENT\_ID | NUMBER | Foreign key to DEPARTMENT\_ID |
|  |  | column of the DEPARTMENTS table |
|  |  |  |

You need to create a view called EMP\_VU that allows the user to insert rows through the view. Which SQL statement, when used to create the EMP\_VU view, allows the user to insert rows?

CREATE VIEW emp\_Vu AS SELECT employee\_id, emp\_name, department\_id

FROM employees

WHERE mgr\_id IN (102, 120);

CREATE VIEW emp\_Vu AS

SELECT employee\_id, emp\_name, job\_id department\_id

FROM employees

WHERE mgr\_id IN (102, 120);

CREATE VIEW emp\_Vu AS

SELECT department\_id, SUM(sal) TOTALSAL FROM employees

WHERE mgr\_id IN (102, 120)

GROUP BY department\_id;

CREATE VIEW emp\_Vu AS

SELECT employee\_id, emp\_name, job\_id, DISTINCT department\_id

FROM employees;

Answer: B

Explanation:

This statement will create view that can be used to change tables in underlying table through simple views. It includes primary key, NOT NULL column and foreign key to avoid constraint restrictions.

Question: 181. (I)

You created a view called EMP\_DEPT\_VU that contains three columns from the EMPLOYEES and DEPARTMENTS tables:

EMPLOYEE\_ID, EMPLOYEE\_NAME AND DEPARTMENT\_NAME.

The DEPARTMENT\_ID column of the EMPLOYEES table is the foreign key to the primary key DEPARTMENT\_ID column of the DEPARTMENTS table.

You want to modify the view by adding a fourth column, MANAGER\_ID of NUMBER data type from the EMPLOYEES tables.How can you accomplish this task?

ALTER VIEW emp\_dept\_vu (ADD manager\_id NUMBER);

MODIFY VIEW emp\_dept\_vu (ADD manager\_id NUMBER);

ALTER VIEW emp\_dept\_vu AS SELECT employee\_id, employee\_name, department\_name, manager\_id

FROM employee e, departments d

WHERE e.department\_id = d.department\_id;

MODIFY VIEW emp\_dept\_vu AS

SELECT employee\_id, employee\_name, department\_name, manager\_id

FROM employees e, departments d WHERE e.department\_id = d.department\_id;

CREATE OR REPLACE VIEW emp\_dept\_vu AS SELECT employee\_id, employee\_name, department\_name, manager\_id

FROM employees e, departments d

WHERE e.department\_id = d.department\_id;

You must remove the existing view first, and then run the CREATE VIEW command with a new column list to modify a view.

Answer: E

Explanation:

When we want to alter the underlying data used in the definition of a view, we use the CREATE OR REPLACE VIEW statement. When a CREATE OR REPLACE VIEW statement is issued, Oracle will disregard the error that arises when it encounters the view that already exists with that name, and it will overwrite the definition for the old view with the definition for the new one.

Question: 182. (I)

Which statements concerning the creation of a view are true? (Choose all that apply.)

A constraint name must be provided when using the WITH CHECK OPTION clause or the statement will fail.

View columns that are the result of derived values must be given a column alias.

When the view already exists, using the OR REPLACE option requires the re-granting of the object privileges previously granted on the view.

A view may have column names that are different than the actual base table(s) column names by using column aliases.

Answer: B & D

Explanation:

View columns that are the result of derived values from any expressions must be given a column alias and view may have column names that are different than the base table column names by using column aliases.

Question: 183. (I)

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

Create a table view.

Create a view in any schema.

Create a view in your schema.

Create a sequence view in any schema.

Create a view that is accessible by everyone.

Create a view only of it is based on tables that you created.

Answer: C

Explanation:

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

Question: 184. (I)

Examine the structure of the CURR\_ORDER table as shown below:

CURR\_ORDER Table

|  |  |  |
| --- | --- | --- |
| ORDER\_ID | NUMBER | NOT NULL, Primary Key |
| CUSTOMER\_ID | NUMBER | NOT NULL, Foreign Key to CUSTOMER\_ID |
|  |  | column of the CUSTOMERS table |
| EMP\_ID | NUMBER | NOT NULL, Foreign Key to EMP\_ID column of |
|  |  | the EMPLOYEES table |
| ORDER\_DT | DATE | NOT NULL |
| ORDER\_AMT | NUMBER (7,2) |  |
| SHIP\_METHOD | VARCHAR2 (5) |  |

You created the ORDER\_V view selecting all rows and columns from the ORDER table where the amount of the order was over $250 and the date of the order was after January 1, 2000. The CREATE VIEW statement included the WITH CHECK OPTION clause.

Which statement will execute successfully?

INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_dt, order\_amt) VALUES (840, 292, 104, '10-OCT-2001', 318);

INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_amt, ship\_method) VALUES (936, 292, 104, 256.3, 'UPXS');

INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_dt, order\_amt, ship\_method) VALUES (164, 292, 104, '10-MAY-2001', 3.56, 'UPXS');

INSERT INTO order\_v (order\_id, customer\_id, emp\_id, order\_dt, order\_amt, ship\_method) VALUES (203, 292, 104, '10-OCT-1999', 298.4, 'UPXS');

Explanation:

Using WITH CHECK OPTION in the view creation you can ensure that all DML on the view stays with in the domain of the view. In the above scenario you cannot insert a date before January 1, 2000 and the amount of the order should also be more than $250.

Question: 185. (I)

An inline view is a SELECT statement that is given an alias and is embedded in the \_\_\_\_\_\_

clause of another SELECT statement.

FROM

WHERE

SELECT

CASE

Explanation:

Inline view is a select statement that is embedded in the FROM clause of another SELECT statement. You can use its result as a table.

Question: 186. (I)

You need to create a view EMP\_VU. The view should allow the users to manipulate the records of only the employees that are working for departments 10 or 20.

Which SQL statement would you use to create the view EMP\_VU?

CREATE VIEW emp\_vu AS SELECT \*

FROM employees

WHERE department\_id IN (10,20);

CREATE VIEW emp\_vu AS SELECT \*

FROM employees

WHERE department\_id IN (10,20) WITH READ ONLY;

CREATE VIEW emp\_vu AS SELECT \*

FROM employees

WHERE department\_id IN (10,20) WITH CHECK OPTION;

CREATE FORCE VIEW emp\_vu AS SELECT \*

FROM employees

WHERE department\_id IN (10,20);

CREATE FORCE VIEW emp\_vu AS SELECT \*

FROM employees

WHERE department\_id IN (10,20) NO UPDATE;

Answer: C

Explanation:

Tables that underlie views often have constraints that limit the data that can be added to those tables. Views cannot add data to the underlying table that would violate the table’s constraints. However, you can also define a view to restrict the user’s ability to change underlying table data even further, effectively placing a special constraint for data manipulation through the view. This additional constraint says that INSERT or UPDATE statements issued against the view are cannot create rows that the view cannot subsequently select. This constraint is configured when the view is defined by adding the WITH CHECK OPTION to the CREATE VIEW statement.

Question: 187. (I)

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

A view can be created as read only.

A view can be created as a join on two or more tables.

A view cannot have an ORDER BY clause in the SELECT statement.

A view cannot be created with a GROUP BY clause in the SELECT statement.

A view must have aliases defined for the column names in the SELECT statement.

& B

Explanation:

A view can be created as read only object. However, it is possible to change data in the underlying table(s) with some restrictions.A view also can be created as a join on two or more tables. This type of view is called complex view. Complex views provide complicated data models where many base tables are drawn together into one virtual table.

Question: 188.(J)

The database administrator of your company created a public synonym called HR for the HUMAN\_RESOURCES table of the GENERAL schema, because many users frequently use this table.

As a user of the database, you created a table called HR in your schema. What happens when you execute this query?

SELECT \*

FROM HR;

You obtain the results retrieved from the public synonym HR created by the database administrator.

You obtain the results retrieved from the HR table that belongs to your schema.

You get an error message because you cannot retrieve from a table that has the same name as a public synonym.

You obtain the results retrieved from both the public synonym HR and the HR table that belongs to your schema, as a Cartesian product.

You obtain the results retrieved from both the public synonym HR and the HR table that belongs to your schema, as a FULL JOIN.

Explanation:

By executing this query you will extract data from the HR table in your own schema, it will not work with HR synonym for the HUMAN\_RESOURCES table of the GENERAL schema.

Question: 189. (J)

Examine the structure of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Remarks |
| EMPLOYEE\_ID | NUMBER | NOT NULL, Primary Key |
| LAST\_NAME | VARCNAR2(30) |  |
| FIRST\_NAME | VARCNAR2(30) |  |
| JOB\_ID | NUMBER |  |
| SAL | NUMBER |  |
| MGR\_ID | NUMBER | References EMPLOYEE\_ID column |
| DEPARTMENT\_ID | NUMBER |  |

You need to create an index called NAME\_IDX on the first name and last name fields of the EMPLOYEES table. Which SQL statement would you use to perform this task?

CREATE INDEX NAME\_IDX (first\_name, last\_name);

CREATE INDEX NAME\_IDX (first\_name AND last\_name);

CREATE INDEX NAME\_IDX ON (first\_name, last\_name);

CREATE INDEX NAME\_IDX

ON employees (first\_name AND last\_name);

CREATE INDEX NAME\_IDX

ON employees(first\_name, last\_name);

CREATE INDEX NAME\_IDX

FOR employees(first\_name, last\_name);

Answer: E

Explanation:

Index is a schema object. It is used by the oracle server to speed up the retrieval of rows by using a pointer and can reduce disk I/O by using rapid path access method to locate the data quickly. It is independent of the table it indexes. Indexes are created for column which contain a wide range of values. It can be created on the first name and last name fields of the employees table using command as

CREATE INDEX NAME\_IDX

ON employees(first\_name, last\_name);

Question: 191. (J)

Mary has a view called EMP\_DEPT\_LOC\_VU that was created based on the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables. She granted SELECT privilege to Scott on this view. Which option enables Scott to eliminate the need to qualify the view with the name MARY

.EMP\_DEP\_LOC\_VU each time the view is referenced?

Scott can create a synonym for the EMP\_DEPT\_LOC\_VU bus using the command: CREATE PRIVATE SYNONYM EDL\_VU

FOR mary.EMP DEPT\_LOC\_VU;

then he can prefix the columns with this synonymn.

Scott can create a synonym for the EMP\_DEPT\_LOC\_VU by using the command: CREATE SYNONYM EDL\_VU

FOR mary.EMP\_DEPT\_LOC\_VU;

then he can prefix the columns with this synonym.

Scott can create a synonym for the EMP\_DEPT\_LOC\_VU by using the command: CREATE LOCAL SYNONYM EDL\_VU

FOR mary.EMP DEPT\_LOC\_VU;

then he can prefix the columns with this synonym.

Scott can create a synonym for the EMP\_DEPT\_LOC\_VU by using the command: CREATE SYNONYM EDL\_VU.

ON mary(EMP\_DEPT\_LOC\_VU);

then he can prefix the columns with this synonym.

Scott cannot create a synonym because synonyms can be created only for tables.

Scott cannot create any synonym for Mary’s view. Mary should create a private synonym for the view and grant SELECT privilege on that synonym to Scott.

Answer: B

Explanation:

Correct syntax to create a local synonym is CREATE SYNONYM *synonym\_name.* With PUBLIC keyword you can create public synonym.

Question: 192. (J)

Which two statements about sequences are true? (Choose two)

You use a NEXTVAL pseudo column to look at the next possible value that would be generated from a sequence, without actually retrieving the value.

You use a CURRVAL pseudo column to look at the current value just generated from a sequence, without affecting the further values to be generated from the sequence.

You use a NEXTVAL pseudo column to obtain the next possible value from a sequence by actually retrieving the value from the sequence.

You use a CURRVAL pseudo column to generate a value from a sequence that would be used for a specified database column.

If a sequence starting from a value 100 and incremented by 1 is used by more then one application, then all of these applications could have a value of 105 assigned to their column whose value is being generated by the sequence.

You use REUSE clause when creating a sequence to restart the sequence once it generates the maximum value defined for the sequence.

Answer: B & C

Explanation:

You use a CURRVAL pseudo column to look at the current value just generated from a sequence, without affecting the further values to be generated from the sequence. You use a NEXTVAL pseudo column to obtain the next possible value from a sequence by actually retrieving the value from the sequence.

Question: 193. (J)

Which SELECT statement will display the next value of the PARTS\_ID\_SEQ sequence by actually retrieving the value from the sequence?

SELECT NEXTVAL(parts\_id\_seq) FROM SYS.DUAL;

SELECT parts\_id\_seq.NEXTVAL FROM inventory;

SELECT parts\_id\_seq.NEXTVAL FROM SYS.DUAL;

SELECT NEXTVAL(parts\_id\_seq) FROM inventory;

SELECT parts\_id\_seq NEXTVAL FROM inventory;

Answer: C

Explanation:

To display the next value of the PARTS\_ID\_SEQ sequence you can use the following statement SELECT parts\_id\_seq.NEXTVAL

FROM SYS.DUAL;

Question: 194. (J)

You issue this statement:

CREATE PUBLIC SYNONYM part

FOR linda.product;

Which task was accomplished by this statement?

A new segment object was created.

A new object privilege was assigned.

A new system privilege was assigned.

The need to qualify an object name with its schema was eliminated.

Explanation:

You can simply the access to the database object by using a synonym an new name for an object. PUBLIC synonym is accessible to all the users of the database. After creating a synonym the need to access an object with its schema was eliminated.

Question: 195. (J)

User Mark wants to eliminate the need to type the full table name when querying the TRANSACTION\_HISTORY table existing in her schema. All other database users should use the schema and full table name when referencing this table.

Which statement should user Marilyn execute?

CREATE PUBLIC SYNONYM trans\_hist FOR mark;

CREATE SYNONYM trans\_hist FOR transaction\_history;

CREATE PRIVATE SYNONYM trans\_hist FOR mark.transaction\_history;

CREATE PUBLIC trans\_hist SYNONYM FOR mark.transaction\_history;

Answer: B

Explanation:

To create a synonym in your own schema use the command

CREATE SYNONYM trans\_hist

FOR transaction\_history;

Now use Mark can access a TRANSACTION\_HISTORY table with an alternate name trans\_list in her schema.

Question: 196. (J)

The TEACHER table in your schema contains these columns:

ID NUMBER(9) NOT NULL, Primary Key

LAST\_NAME VARCHAR2(25)

FIRST\_NAME VARCHAR2(25)

SUBJECT\_ID NUMBER(9)

You execute this statement:

CREATE INDEX teacher\_name\_idx

ON teacher(first\_name, last\_name);

Which statement is true?

The statement creates a composite non-unique index.

The statement creates a composite unique index.

You must have the CREATE ANY INDEX privilege for the statement to succeed.

The statement will fail because it contains a syntax error.

Explanation:

Indexes can be unique or nonunique. Unique indexes guarantee that no two rows of a table have duplicate values in the key column (or columns). Nonunique indexes do not impose this restriction on the column values. By executing the above written statement a composite non unique index will be created.

Question: 197. (J)

The LINE\_ITEM table contains these columns:

LINE\_ITEMID NUMBER(9)

ORDER\_ID NUMBER(9)

PRODUCT\_ID VARCHAR2(9)

QUANTITY NUMBER(5)

You created a sequence called LINE\_ITEMID\_SEQ to generate sequential values for the LINE\_ITEMID column.

Evaluate this SELECT statement:

SELECT line\_itemid\_seq.CURRVAL

FROM dual;

Which task will this statement accomplish?

displays the next value of the LINE\_ITEMID\_SEQ sequence

displays the current value of the LINE\_ITEMID\_SEQ sequence

populates the LINE\_ITEMID\_SEQ sequence with the next value

increments the LINE\_ITEMID column

Answer: B

Explanation:

CURRVAL is pesudocolumn that will display the current value of the sequence that has been used.

Question: 198. (J)

Evaluate this statement:

CREATE SEQUENCE line\_item\_id\_seq

START WITH 10001

MAXVALUE 999999999

NOCYCLE;

Which statement about this CREATE SEQUENCE statement is true?

The sequence will reuse numbers and will start with 10001.

The sequence will never reuse any numbers and will increment by 1.

The sequence will continue to generate values after it reaches its maximum value.

The CREATE SEQUENCE statement will cause a syntax error because an INCREMENT BY

value is not included.

Answer: B

Explanation:

When the sequence is created with out specifying the increment by clause then it will increase by 1 which is default value and NOCYCLE option state that the sequence will not used after reaching its maximum value.

Question: 199. (J)

Which statement should you use to eliminate the need for all users to qualify Marilyn's INVENTORY table with her schema when querying the table?

CREATE SYNONYM inventory FOR inventory;

CREATE PUBLIC SYNONYM inventory FOR marilyn;

CREATE PUBLIC SYNONYM inventory FOR marilyn.invent ory;

CREATE PUBLIC inventory SYNONYM FOR marilyn.inventory;

Answer: C

Explanation:

By creating a public synonym for the inventory table then the need to qualify the table with its schema name is eliminated as it has been done by the following command.

CREATE PUBLIC SYNONYM inventory

FOR marilyn.inventory;

Question: 200. (J)

Which statement will user Barbara use to create a private synonym when referencing the EMPLOYEE table existing in user Chan's schema?

CREATE SYNONYM emp FOR chan.employee;

CREATE PUBLIC SYNONYM emp FOR chan.barbara;

CREATE PRIVATE SYNONYM emp

FOR chan.employee;

CREATE PUBLIC emp SYNONYM FOR chan.employee;

Explanation:

To create a private synonym for your own schema use the following command. CREATE SYNONYM emp

FOR chan.employee;

Question: 201. (J)

Why would you NOT create an index on a column in the CLASS\_SCHEDULE table?

to reduce disk I/O

to speed up row retrieval

to speed up queries if the table has less than 50 rows

to speed up queries that return less than 3 percent of the rows

to speed up queries that include a foreign key reference to the STUDENT table

Answer: C

Explanation:

Index are created on the large table to speed up the data retrieval or if the changes are perfomed on the table and the column is used in the WHERE clause then on that column you should create an index to speed up the DML operations. If the table is small then index should not be created as this would slow the speed of queries execution.

Question: 202.(K)

Which data dictionary table should you query to view the object privileges granted to the user on specific columns?

USER\_TAB\_PRIVS\_MADE

USER\_TAB\_PRIVS

USER\_COL\_PRIVS\_MADE

USER\_COL\_PRIVS

Explanation:

The USER\_COL\_PRIVS data dictionary view will show the object privileges granted to the user on specific columns.

Question: 203. (K)

Which two statements accurately describe a role? (Choose two.)

A role can be given to a maximum of 1000 users.

A user can have access to a maximum of 10 roles.

A role can have a maximum of 100 privileges contained in it.

Privileges are given to a role by using the CREATE ROLE statement.

A role is a named group of related privileges that can be granted to the user.

A user can have access to several roles, and several users can be assigned the same role.

& F

Explanation:

Privileges can be given to a role by using the CREATE ROLE statement. A user can have access to several roles, and several users can be assigned the same role.

Question: 204. (K)

Scott issues the SQL statements:

CREATE TABLE dept (deptno NUMBER(2), dname VARCHAR2(14), loc VARCHAR2(13)}; GRANT SELECT

ON DEPT

TO SUE;

If Sue needs to select from Scott's DEPT table, which command should she use?

SELECT \* FROM DEPT;

SELECT \*

FROM SCOTT.DEPT;

SELECT \*

FROM DBA.SCOTT.DEPT;

SELECT \*

FROM ALL\_ USERS

WHERE USER\_NAME = 'SCOTT' AND TABLE NAME = 'DEPT';

Answer: B

Explanation:

If the table doesn’t exist in your schema, you must prefix the table name with the schema information, separating the owner from the table name with a period.

Question: 205. (K)

Which statement creates a new user?

CREATE USER susan;

CREATE OR REPLACE USER susan;

CREATE NEW USER susan DEFAULT;

CREATE USER susan IDENTIFIED BY blue;

CREATE NEW USER susan IDENTIFIED by blue;

CREATE OR REPLACE USER susan IDE NTIFIED BY blue;

Answer: D

Explanation:

The correct statement to create user is: CREATE USER *user\_name* IDENTIFIED BY *password*.

Question: 206. (K)

What is true about the WITH GRANT OPTION clause?

It allows a grantee DBA privileges.

It is required syntax for object privileges.

It allows privileges on specified columns of tables.

It is used to grand an object privilege on a foreign key column.

It allows the grantee to grand object privileges to other users and roles.

Answer: E

Explanation:

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

Question: 207. (K)

You need to give the MANAGER role the ability to select from, insert into, and modify existing rows in the STUDENT\_GRADES table. Anyone given this MANAGER role should be able to pass those privileges on to others.

Which statement accomplishes this?

GRANT select, insert, update ON student\_grades

TO manager;

GRANT select, insert, update ON student\_grades

TO ROLE manager;

GRANT select, insert, modify ON student\_grades

TO manager

WITH GRANT OPTION;

GRANT select, insert, update ON student\_grades.

TO manager

WITH GRANT OPTION;

GRANT select, insert, update ON student\_grades

TO ROLE manager WITH GRANT OPTION;

F.GRANT select, insert, modify ON student\_grades

TO ROLE manager WITH GRANT OPTION;

Explanation:

This answer provides correct syntax of GRANT command to give the MANAGER role all asked privileges. Clause WITH GRANT OPTION will allow this role to pass those privileges on to others.

Question: 208. (K)

When should you create a role? (Choose two)

To simplify the process of creating new users using the CREATE USER xxx IDENTIFIED by yyy statement.

To grant a group of relate privileges to a user.

When the number of people using the database is very high.

To simplify the process of granting and revoking privileges.

To simplify profile maintenance for a user who is constantly traveling.

Answer: C & D

Explanation:

You should use roles to grant a group of privileges to a user. You grant the appropriate privileges to the role and after that grant this role to specific users. By granting to or revoking privileges from the role you can simplify procedure of users privileges maintainance: you don’t need to grant/revoke privileges to/from each user. It’s especially very helpful when you are experiencing a high number of people using the database.

Question: 209. (K)

The DBA issues this SQL command:

CREATE USER scott

IDENTIFIES by tiger;

What privileges does the user Scott have at this point?

No privileges.

Only the SELECT privilege.

Only the CONNECT privilege.

All the privileges of a default user.

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.

Question: 210. (K)

You granted user Joe the INDEX and REFERENCES privileges on the INVENTORY table. Which statement did you use?

GRANT ALL ON inventory TO joe;

GRANT ANY PRIVILEGE ON inventory

TO joe;

GRANT INDEX AND REFERENCES ON inventory

TO joe;

GRANT ALL WITH GRANT OPTION ON inventory

TO joe;

Explanation:

To grant privileges to a used on specific table use the following command

GRANT ALL

ON inventory TO joe;

This command will grant all privilege on the inventory table to the user joe

Question: 211. (K)

Which condition would allow a user to grant SELECT privileges on the CUSTOMER table to everyone using the PUBLIC keyword?

The user has SELECT privileges on the CUSTOMER table.

The user owns the CUSTOMER table.

The user has been granted the PUBLIC privilege.

The user has been granted the SELECT privilege with the PUBLIC OPTION.

Answer: B

Explanation:

To grant SELECT privilege to all users in the database you can use the PUBLIC key word. This will grant the privileges to all users in the database.

Question: 212. (K)

You issued this statement:

REVOKE REFERENCES

ON inventory

FROM joe

CASCADE CONSTRAINTS;

Which two tasks were accomplished by executing this statement? (Choose two.)

A. All the PRIMARY KEY constraints created by user Joe were removed.

All the FOREIGN KEY constraints on the INVENTORY table created by user Joe were removed.

The ability to create a PRIMARY KEY constraint was revoked from user Joe.

The ability to create a FOREIGN KEY constraint on the INVENTORY table was revoked from user Joe.

All the constraints created by user Joe were removed.

The ability to create any constraints was revoked from user Joe.

Answer: B & D

Explanation:

REVOKE REFERENCE command will revoke the ability to create a foreign key constraints on the inventory table by use joe and also all created FOREIGN KEY constraints will be removed.

Question: 213. (K)

Which privilege is an object privilege?

INDEX

DROP USER

CREATE SESSION

BACKUP ANY TABLE

Explanation:

INDEX an object privilege, which is granted to a user on a table if the INDEX privilege is revoked from a user that created an index on someone else's table, the index continues to exist after the privilege is revoked.

Question: 214. (K)

You issued this statement:

GRANT UPDATE

ON inventory

TO joe

WITH GRANT OPTION;

Which task was accomplished?

Only a system privilege was granted to user Joe.

Only an object privilege was granted to user Joe.

User Joe was granted all privileges on the INVENTORY object.

Both an object privilege and a system privilege were granted to user Joe.

Answer: B

Explanation:

By executing the above command only the UPDATE object privileges will be granted to the user joe and WITH GRANT OPTION state that the user can further grant privilege to other users.

Question: 215. (K)

Which statement is true concerning a new user that has only been granted the CREATE SESSION privilege?

The user can create a table.

The user can alter their password.

The user can select from tables they have created.

The user can grant privileges on objects that they own.

Answer: B

Explanation:

When a user has been granted the CREATE SESSION privilege then user can change their password after connecting to the database.

Question: 216. (K)

Evaluate this SQL script:

CREATE USER hr IDENTIFIED BY hr01;

CREATE ROLE hr\_manager;

CREATE ROLE hr\_clerk;

GRANT hr\_clerk TO hr\_manager;

CREATE ROLE hr\_director;

GRANT hr\_manager TO hr\_director;

GRANT hr\_director TO hr

/

How many roles will the HR user have access to?

0

1

2

3

Explanation:

In the above statmen three roles will be granted to the user HR. as the role hr\_clerk is granted to the hr\_manager and then hr\_manager is granted to the hr\_director role and then this hr\_director role is granted to the user HR have access to all three roles.

Question: 217 (A)

Examine the structure of the EMP\_DEPT\_VU view:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | | Type | | Remarks | |
| EMPLOYEE\_ID | | NUMBER | | From the EMPLOYEES table | |
| EMP\_NAME | | VARCHAR2(30) | | From the EMPLOYEES table | |
| JOB\_ID | | VARCHAR2(20) | | From the EMPLOYEES table | |
| SALARY | | NUMBER | | From the EMPLOYEES table | |
|  | | | | | | | | |
|  | |  | |  |
| DEPT\_NAME | |  | | VARCHAR2(30) | | From the DEPARTMENTS table | |  |

Which SQL statement produces an error?

SELECT \* FROM emp\_dept\_vu;

SELECT department\_id, SUM(salary) FROM emp\_dept\_vu GROUP BY department\_id;

SELECT department\_id, job\_id, AVG(salary) FROM emp\_dept\_vu GROUP BY department\_id, job\_id;

SELECT job\_id, SUM(salary) FROM emp\_dept\_vu WHERE department\_id IN (10,20) GROUP BY job\_id HAVING SUM(salary) > 20000;

None of the statements produce an error; all are valid.

Explanation :

None of the statements produce an error.

Question: 218 (H)

You own a table called EMPLOYEES with this table structure:

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | Primary Key |
| FIRST\_NAME | VARCHAR2(25) |  |
| LAST\_NAME | VARCHAR2(25) |  |
| HIRE\_DATE | DATE |  |

What happens when you execute this DELETE statement?

DELETE employees;

You get an error because of a primary key violation.

The data and structure of the EMPLOYEES table are deleted.

The data in the EMPLOYEES table is deleted but not the structure.

You get an error because the statement is not syntactically correct.

Explanation :

You can remove existing rows from a table by using the DELETE statement. DELETE [FROM] table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [WHERE condition];  Question: 219 (H) | |  |  |  |
| 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?

MERGE INTO new\_employees c USING employees e ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN UPDATE SET c.name = e.first\_name ||','|| e.last\_name WHEN NOT MATCHED THEN INSERT VALUES (e.employee\_id, e.first\_name ||', '||e.last\_name);

MERGE new\_employees c USING employees e ON (c.employee\_id = e.employee\_id) WHEN EXISTS THEN UPDATE SET c.name = e.first\_name ||','|| e.last\_name WHEN NOT MATCHED THEN INSERT VALUES (e.employee\_id, e.first\_name ||', '||e.last\_name);

MERGE INTO new\_employees c USING employees e ON (c.employee\_id = e.employee\_id) WHEN EXISTS THEN UPDATE SET c.name = e.first\_name ||', '|| e.last\_name WHEN NOT MATCHED THEN INSERT VALUES(e.employee\_id, e.first\_name ||', '||e.last\_name);

MERGE new\_employees c FROM employees e ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN UPDATE SET c.name = e.first\_name ||','|| e.last\_name WHEN

NOT MATCHED THEN INSERT INTO new\_employees VALUES (e.employee\_id, e.first\_name ||', '||e.last\_name);

Explanation :

The correct statement for MERGE is

MERGE INTO *table\_name*

Question: 220 (C)

The EMPLOYEES table contains these columns:

EMPLOYEE\_ID NUMBER(4)

ENAME VARCHAR2 (25)

JOB\_ID VARCHAR2(10)

Which SQL statement will return the ENAME, length of the ENAME, and the numeric position of the letter "a" in the ENAME column, for those employees whose ENAME ends with a the letter "n"?

A. SELECT ENAME, LENGTH(ENAME), INSTR(ENAME, 'a') FROM EMPLOYEES

WHERE SUBSTR(ENAME, -1, 1) = 'n';

SELECT ENAME, LENGTH(ENAME), INSTR(ENAME, ,-1,1) FROM EMPLOYEES WHERE SUBSTR(ENAME, -1, 1) = 'n';

SELECT ENAME, LENGTH(ENAME), SUBSTR(ENAME, -1,1) FROM EMPLOYEES WHERE INSTR(ENAME, 1, 1) = 'n';

SELECT ENAME, LENGTH(ENAME), SUBSTR(ENAME, -1,1) FROM EMPLOYEES

WHERE INSTR(ENAME, -1, 1) = 'n';

Explanation :

INSTR is a character function return the numeric position of a named string. INS TR(NAMED,’a’)

Question: 221 (G)

Which four are valid Oracle constraint types? (Choose four.)

CASCADE

UNIQUE

NONUNIQUE

CHECK

PRIMARY KEY

CONSTANT

NOT NULL

, D, E, G

Explanation :

Oracle constraint type is Not Null, Check, Primary Key, Foreign Key and Unique

Question: 222 (I)

Which SQL statement would you use to remove a view called EMP\_DEPT\_VU from your schema?

DROP emp\_dept\_vu;

DELETE emp\_dept\_vu;

REMOVE emp\_dept\_vu;

DROP VIEW emp\_dept\_vu;

DELETE VIEW emp\_dept\_vu;

F. REMOVE VIEW emp\_dept\_vu;

Explanation :

DROP VIEW viewname;

Question: 223 (C)

Examine the description of the EMPLOYEES table:

EMP\_ID NUMBER(4) NOT NULL

LAST\_NAME VARCHAR2(30) NOT NULL

FIRST\_NAME VARCHAR2(30)

DEPT\_ID NUMBER(2)

Which statement produces the number of different departments that have employees with last name Smith?

SELECT COUNT(\*) FROM employees WHERE last\_name='Smith';

SELECT COUNT (dept\_id) FROM employees WHERE last\_name='Smith';

SELECT DISTINCT(COUNT(dept\_id)) FROM employees WHERE last\_name='Smith';

SELECT COUNT(DISTINCT dept\_id) FROM employees WHERE last\_name='Smith';

SELECT UNIQUE(dept\_id) FROM employees WHERE last\_name='Smith';

Question: 224 (I)

Top N analysis requires \_\_\_\_\_ and \_\_\_\_\_. (Choose two.)

the use of rowed

a GROUP BY clause

an ORDER BY clause

only an inline view

an inline view and an outer query

, E

Explanation :

The correct statement for Top-N Analysis

SELECT [coloumn\_list], ROWNUM

FROM (SELECT [coloumn\_list]

FROM table

ORDER BY Top-N\_coloumn)

WHERE ROWNUM <= N;

Question: 225 (G)

Which statement adds a constraint that ensures the CUSTOMER\_NAME column of the CUSTOMERS table holds a value?

ALTER TABLE customers ADD CONSTRAINT cust\_name\_nn CHECK customer\_name IS NOT NULL;

ALTER TABLE customers MODIFY CONSTRAINT cust\_name\_nn CHECK

customer\_name IS NOT NULL;

ALTER TABLE customers MODIFY customer\_name CONSTRAINT cust\_name\_nn NOT NULL;

ALTER TABLE customers MODIFY customer\_name CONSTRAINT cust\_name\_nn IS NOT NULL;

ALTER TABLE customers MODIFY name CONSTRAINT cust\_name\_nn NOT NULL;

ALTER TABLE customers ADD CONSTRAINT cust\_name\_nn CHECK customer\_name NOT NULL;

Question: 226 (A)

Evaluate this SQL statement:

SELECT ename, sal, 12\*sal+100 FROM emp;

The SAL column stores the monthly salary of the employee. Which change must be made to the above syntax to calculate the annual compensation as "monthly salary plus a monthly bonus of $100, multiplied by 12"?

No change is required to achieve the desired results.

SELECT ename, sal, 12\*(sal+100) FROM emp;

SELECT ename, sal, (12\*sal)+100 FROM emp;

SELECT ename, sal+100,\*12 FROM emp;

Explanation:

To achieve the result you must add 100 to sal before multiply with 12.

Select ename, sal, 12\*(sal+100) from EMP;

Question: 227 (K)

You are the DBA for an academic database. You need to create a role that allows a group of users to modify existing rows in the STUDENT\_GRADES table.

Which set of statements accomplishes this?

CREATE ROLE registrar; GRANT MODIFY ON student\_grades TO registrar; GRANT registrar to user1, user2, user3

CREATE NEW ROLE registrar; GRANT ALL ON student\_grades TO registrar; GRANT registrar to user1, user2, user3

CREATE ROLE registrar; GRANT UPDATE ON student\_grades TO registrar; GRANT ROLE registrar to user1, user2, user3

CREATE ROLE registrar; GRANT UPDATE ON student\_grades TO registrar; GRANT

registrar to user1, user2, user3;

CREATE registrar; GRANT CHANGE ON student\_grades TO registrar; GRANT registrar;

Explanation:

This is the correct solution for the answer.

GRANT role\_name to users;

Question: 228 (G)

You need to modify the STUDENTS table to add a primary key on the STUDENT\_ID column. The table is currently empty.

Which statement accomplishes this task?

ALTER TABLE students ADD PRIMARY KEY student\_id;

ALTER TABLE students ADD CONSTRAINT PRIMARY KEY (student\_id);

ALTER TABLE students ADD CONSTRAINT stud\_id\_pk PRIMARY KEY student\_id;

ALTER TABLE students ADD CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

ALTER TABLE studentsMODIFY CONSTRAINT stud\_id\_pk PRIMARY KEY (student\_id);

Explanation :

ALTER TABLE table\_name

ADD [CONSTRAINT constraint] type (coloumn);

Introduction to Oracle9i : SQL, Oracle University Study Guide, 10-17

Question: 229 (B)

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?

SELECT student\_id, gpa FROM student\_grades ORDER BY gpa ASC;

SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa ASC;

SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa;

SELECT student\_id, gpa FROM student\_grades ORDER BY gpa;

SELECT student\_id, gpa FROM student\_grades SORT ORDER BY gpa DESC;

SELECT student\_id, gpa FROM student\_grades ORDER BY gpa DESC;

Explanation :

sorted by highest to lowest is DESCENDING order

Question: 230 (F)

Which describes the default behavior when you create a table?

The table is accessible to all users.

Tables are created in the public schema.

Tables are created in your schema.

Tables are created in the DBA schema.

You must specify the schema when the table is created.

Explanation :

sorted by highest to lowest is DESCENDING order

Question: 231 (C)

Which four are attributes of single row functions? (Choose four.)

cannot be nested

manipulate data items

act on each row returned

return one result per row

accept only one argument and return only one value

accept arguments which can be a column or an expression

, C, D, E

Explanation:

manipulate data items, act on each row returned, return one result per row, accept only one argument and return only one value.

Question: 232 (F)

You need to create a table named ORDERS that contain four columns:

an ORDER\_ID column of number data type

a CUSTOMER\_ID column of number data type

an ORDER\_STATUS column that contains a character data type

a DATE\_ORDERED column to contain the date the order was placed.

When a row is inserted into the table, if no value is provided when the order was placed, today’s date should be used instead.

Which statement accomplishes this?

CREATE TABLE orders ( order\_id NUMBER (10), customer\_id NUMBER (8), order\_status VARCHAR2 (10), date\_ordered DATE = SYSDATE);

CREATE TABLE orders ( order\_id NUMBER (10), customer\_id NUMBER (8), order\_status VARCHAR2 (10),

date\_ordered DATE DEFAULT SYSDATE);

CREATE OR REPLACE TABLE orders ( order\_id NUMBER (10),

customer\_id NUMBER (8), order\_status VARCHAR2 (10),

date\_ordered DATE DEFAULT SYSDATE); D. CREATE OR REPLACE TABLE orders (

order\_id NUMBER (10), customer\_id NUMBER (8), order\_status VARCHAR2 (10), date\_ordered DATE = SYSDATE);

CREATE TABLE orders ( order\_id NUMBER (10), customer\_id NUMBER (8), order\_status NUMBER (10), date\_ordered DATE = SYSDATE);

CREATE TABLE orders ( order\_id NUMBER (10), customer\_id NUMBER (8), order\_status NUMBER (10),

date\_ordered DATE DEFAULT SYSDATE);

Question: 233 (H)

Examine the structure of the EMPLOYEES table:

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | Primary Key |
| FIRST\_NAME | VARCHAR2(25) |  |
| LAST\_NAME | VARCHAR2(25) |  |

Which three statements insert a row into the table? (Choose three.)

INSERT INTO employees VALUES ( NULL, 'John', 'Smith');

INSERT INTO employees( first\_name, last\_name) VALUES( 'John', 'Smith');

INSERT INTO employees VALUES ( '1000', 'John', NULL);

INSERT INTO employees (first\_name, last\_name, employee\_id) VALUES ( 1000, 'John', 'Smith');

INSERT INTO employees (employee\_id) VALUES (1000);

INSERT INTO employees (employee\_id, first\_name, last\_name) VALUES ( 1000, 'John', ' ');

, E, F

Explanation :

EMPLOYEE\_ID is a primary key.

Question: 234 (K)

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?

Alice only

Alice and Reena

Alice, Reena, and Timber

Sue, Alice, Reena, and Timber

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.

Question: 235 (B)

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?

SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\\_%' ESCAPE '\';

SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\_';

SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\_' ESCAPE "\";

SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id = '%SA\_';

Explanation:

ESCAPE identifier to search for the actual % and \_ symbol

Question: 236 (A)

The CUSTOMERS table has these columns:

CUSTOMER\_ID NUMBER(4) NOT NULL

CUSTOMER\_NAME VARCHAR2(100) NOT NULL

CUSTOMER\_ADDRESS VARCHAR2(150)

CUSTOMER\_PHONE VARCHAR2(20)

You need to produce output that states "Dear Customer customer\_name, ". The customer\_name data values come from the CUSTOMER\_NAME column in the CUSTOMERS table.

Which statement produces this output?

SELECT dear customer, customer\_name, FROM customers;

SELECT "Dear Customer", customer\_name || ',' FROM customers;

SELECT 'Dear Customer ' || customer\_name ',' FROM customers;

SELECT 'Dear Customer ' || customer\_name || ',' FROM customers;

SELECT "Dear Customer " || customer\_name || "," FROM customers;

SELECT 'Dear Customer ' || customer\_name || ',' || FROM customers;

Explanation :

Concatenation operator to create a resultant column that is a character expression.

Question: 237 (J)

What is true about sequences?

Once created, a sequence belongs to a specific schema.

Once created, a sequence is linked to a specific table.

Once created, a sequence is automatically available to all users.

Only the DBA can control which sequence is used by a certain table.

Once created, a sequence is automatically used in all INSERT and UPDATE statements.

Question: 238 (K)

Which object privileges can be granted on a view?

none

DELETE, INSERT,SELECT

ALTER, DELETE, INSERT, SELECT

DELETE, INSERT, SELECT, UPDATE

Explanation :

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

Question: 239 (F)

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.)

SER\_NO

ORDER\_ID

STATUS

PROD\_ID

ORD\_TOTAL

composite index on ORDER\_ID and ORDER\_DATE

, F

Explanation:

Index exist for UNIQUE and PRIMARY KEY constraints

Question: 240 (C)

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

Group functions on columns ignore NULL values.

Group functions on columns returning dates include NULL values.

Group functions on columns returning numbers include NULL values.

Group functions on columns cannot be accurately used on columns that contain NULL values.

Group functions on columns include NULL values in calculations if you use the keyword INC\_NULLS.

Explanation :

group functions on column ignore NULL values

Question: 241 (C)

The STUDENT\_GRADES table has these columns:

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

Which statement finds the highest grade point average (GPA) per semester?

SELECT MAX(gpa) FROM student\_grades WHERE gpa IS NOT NULL;

SELECT (gpa) FROM student\_grades GROUP BY semester\_end WHERE gpa IS NOT NULL;

SELECT MAX(gpa) FROM student\_grades WHERE gpa IS NOT NULL GROUP BY

semester\_end;

SELECT MAX(gpa) GROUP BY semester\_end WHERE gpa IS NOT NULL FROM student\_grades;

SELECT MAX(gpa) FROM student\_grades GROUP BY semester\_end WHERE gpa IS NOT NULL;

Explanation :

for highest gpa value MAX function is needed,

for result with per semester GROUP BY clause is needed.

Question: 242 (E)

In which four clauses can a subquery be used? (Choose four.)

in the INTO clause of an INSERT statement

in the FROM clause of a SELECT statement

in the GROUP BY clause of a SELECT statement

in the WHERE clause of a SELECT statement

in the SET clause of an UPDATE statement

in the VALUES clause of an INSERT statement

, D, E, F

Explanation :

subquery can be use in the FROM clause of a SELECT statement, in the WHERE clause of a SELECT statement, in the SET clauses of an UPDATE statement, in the VALUES clause of an INSERT statement.

Question: 243 (H)

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?

An error is returned.

You are prompted to enter a new value.

A report is produced that matches the first report produced.

You are asked whether you want a new value or if you want to run the report based on the previous value.

Explanation :

use the double-ampersand if you want to reuse the variable value without prompting the user each time.

Question: 244 (C)

Which SQL statement returns a numeric value?

SELECT ADD\_MONTHS(MAX(hire\_Date), 6) FROM EMP;

SELECT ROUND(hire\_date)FROM EMP;

SELECT sysdate-hire\_date FROM EMP;

SELECT TO\_NUMBER(hire\_date + 7)FROM EMP;

Explanation :

DATE value subtract DATE value will return numeric value.

Question: 245 (J)

What are two reasons to create synonyms? (Choose two.)

You have too many tables.

Your tables are too long.

Your tables have difficult names.

You want to work on your own tables.

You want to use another schema's tables.

You have too many columns in your tables.

, C

Explanation :

create a synonyms when the table name is difficult or you have too many tables

Question: 246 (I)

What is true about updates through a view?

You cannot update a view with group functions.

When you update a view group functions are automatically computed.

When you update a view only the constraints on the underlying table will be in effect.

When you update a view the constraints on the views always override the constraints on the underlying tables.

Question: 247 (E)

You need to write a SQL statement that returns employee name, salary, department ID, and maximum salary earned in the department of the employee for all employees who earn less than the maximum salary in their department.

Which statement accomplishes this task?

SELECT a.emp\_name, a.sal, b.dept\_id, MAX(sal) FROM employees a, departments b WHERE a.dept\_id = b.dept\_id AND a.sal < MAX(sal) GROUP BY b.dept\_id;

SELECT a.emp\_name, a.sal, a.dept\_id, b.maxsal FROM employees a, (SELECT dept\_id, MAX(sal) maxsal FROM employees GROUP BY dept\_id) b WHERE a.dept\_id = b.dept\_id AND a.sal < b.maxsal;

SELECT a.emp\_name, a.sal, a.dept\_id, b.maxsal FROM employees a WHERE a.sal < (SELECT MAX(sal) maxsal FROM employees b GROUP BY dept\_id);

SELECT emp\_name, sal, dept\_id, maxsal FROM employees, (SELECT dept\_id, MAX(sal) maxsal FROM employees GROUP BY dept\_id) WHERE a.sal < maxsal;

Explanation :

function MAX(column\_name)

Question: 248 (B)

The CUSTOMERS table has these columns:

CUSTOMER\_ID NUMBER(4) NOT NULL

CUSTOMER\_NAME VARCHAR2(100) NOT NULL

STREET\_ADDRESS VARCHAR2(150)

CITY\_ADDRESS VARCHAR2(50)

STATE\_ADDRESS VARCHAR2(50)

PROVINCE\_ADDRESS VARCHAR2(50)

COUNTRY\_ADDRESS VARCHAR2(50)

POSTAL\_CODE VARCHAR2(12)

CUSTOMER\_PHONE VARCHAR2(20)

A promotional sale is being advertised to the customers in France. Which WHERE clause identifies customers that are located in France?

WHERE lower(country\_address) = "france"

WHERE lower(country\_address) = 'france'

WHERE lower(country\_address) IS 'france'

WHERE lower(country\_address) = '%france%'

WHERE lower(country\_address) LIKE %france%

Explanation :

WHERE lower(country\_address)=’france’

Explanation :

The only SQL\*Plus command in this list : DESCRIBE. It cannot be used as SQL command.

This command returns a description of tablename, including all columns in that table, the datatype for each column and an indication of whether the column permits storage of NULL values.

Question: 250 (C)

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?

SELECT city\_address, COUNT(\*) FROM customers

WHERE city\_address IN ( ‘Los Angeles’, ‘San Fransisco’);

SELECT city\_address, COUNT (\*) FROM customers

WHERE city address IN ( ‘Los Angeles’, ‘San Fransisco’) GROUP BY city\_address;

SELECT city\_address, COUNT(customer\_id) FROM customers

WHERE city\_address IN ( ‘Los Angeles’, ‘San Fransisco’)

GROUP BY city\_address, customer\_id;

SELECT city\_address, COUNT (customer\_id) FROM customers

GROUP BY city\_address IN ( ‘Los Angeles’, ‘San Fransisco’);

Question: 251 (I)

What does the FORCE option for creating a view do?

creates a view with constraints

creates a view even if the underlying parent table has constraints

creates a view in another schema even if you don't have privileges

creates a view regardless of whether or not the base tables exist

Explanation :

create a view regardless of whether or not the base tables exist.

Question: 252 (H)

A data manipulation language statement \_\_\_\_\_.

completes a transaction on a table

modifies the structure and data in a table

modifies the data but not the structure of a table

modifies the structure but not the data of a table

Explanation :

modifies the data but not the structure of a table

Question: 253 (C)

You need to perform these tasks:

Create and assign a MANAGER role to Blake and Clark

Grant CREATE TABLE and CREATE VIEW privileges to Blake and Clark

Which set of SQL statements achieves the desired results?

CREATE ROLE manager; GRANT create table, create view TO manager;

GRANT manager TO BLAKE,CLARK;

CREATE ROLE manager;

GRANT create table, create voew TO manager;

GRANT manager ROLE TO BLAKE,CLARK; C. GRANT manager ROLE TO BLAKE,CLARK;

GRANT create table, create voew TO BLAKE CLARK;

\*\*\*MISSING\*\*\*

Question: 254 (K)

The DBA issues this SQL command:

CREATE USER scott IDENTIFIED by tiger;

What privileges does the user Scott have at this point?

no privileges

only the SELECT privilege

only the CONNECT privilege

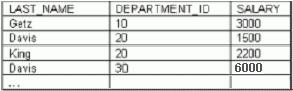
all the privileges of a default user

Explanation :

when a user is created, by default no privilege is granted

Question: 255 (E)

Exhibit:



Examine the data in the EMPLOYEES table.

Examine the subquery:

SELECT last\_name

FROM employees

WHERE salary IN (SELECT MAX(salary)

FROM employees

GROUP BY department\_id);

Which statement is true?

The SELECT statement is syntactically accurate.

The SELECT statement does not work because there is no HAVING clause.

The SELECT statement does not work because the column specified in the GROUP BY

clause is not in the SELECT list.

The SELECT statement does not work because the GROUP BY clause should be in the main query and not in the subquery.

Question: 256 (K)

Examine the statement:

GRANT select, insert, update

ON student\_grades

TO manager

WITH GRANT OPTION;

Which two are true? (Choose two.)

MANAGER must be a role.

It allows the MANAGER to pass the specified privileges on to other users.

It allows the MANAGER to create tables that refer to the STUDENT\_GRADES table.

It allows the MANAGER to apply all DML statements on the STUDENT\_GRADES table.

It allows the MANAGER the ability to select from, insert into, and update the STUDENT\_GRADES table.

It allows the MANAGER the ability to select from, delete from, and update the STUDENT\_GRADES table.

, E

Explanation :

GRANT ROLE to ROLE/USER

Question: 257 (I)

Which best describes an inline view?

a schema object

a subquery that can contain an ORDER BY clause

another name for a view that contains group functions

a subquery that is part of the FROM clause of another query

Explanation :

a subquery that is part of the FROM clause of another query

Question: 258 (D)

Examine the structure of the EMPLOYEES and DEPARTMENTS tables:

EMPLOYEES

EMPLOYEE\_ID NUMBER

DEPARTMENT\_ID NUMBER

MANAGER\_ID NUMBER

LAST\_NAME VARCHAR2(25)

DEPARTMENTS

DEPARTMENT\_ID NUMBER

MANAGER\_ID NUMBER

DEPARTMENT\_NAME VARCHAR2(35)

LOCATION\_ID NUMBER

You want to create a report displaying employee last names, department names, and locations. Which query should you use to create an equi-join?

SELECT last\_name, department\_name, location\_id FROM employees , departments ;

SELECT employees.last\_name, departments.department\_name, departments.location\_id

FROM employees e, departments D WHERE e.department\_id =d.department\_id;

SELECT e.last\_name, d.department\_name, d.location\_id FROM employees e, departments D WHERE manager\_id =manager\_id;

SELECT e.last\_name, d.department\_name, d.location\_id FROM employees e, departments D WHERE e.department\_id =d.department\_id;

Explanation :

Equijoins are also called simple joins or inner joins. Equijoin involve primary key and foreign key.

Question: 259 (B)

The PRODUCTS table has these columns:

PRODUCT\_ID NUMBER(4)

PRODUCT\_NAME VARCHAR2(45)

PRICE NUMBER(8,2)

Evaluate this SQL statement:

SELECT \* FROM PRODUCTS ORDER BY price, product\_name;

What is true about the SQL statement?

The results are not sorted.

The results are sorted numerically.

The results are sorted alphabetically.

The results are sorted numerically and then alphabetically.

Explanation :

the result is sort by price which is numeric and follow by product\_name which is alphabetically.

Question: 260 (D)

In which two cases would you use an outer join? (Choose two.)

The tables being joined have NOT NULL columns.

The tables being joined have only matched data.

The columns being joined have NULL values.

The tables being joined have only unmatched data.

The tables being joined have both matched and unmatched data.

Only when the tables have a primary key/foreign key relationship.

, E

Explanation:

You use an outer join to also see rows that do not meet the join condition.

Question: 261 (G)

Which constraint can be defined only at the column level?

UNIQUE

NOT NULL

CHECK

PRIMARY KEY

FOREIGN KEY

Explanation:

the NOT NULL constraint can be specified only at the column level, not at the table level.

|  |  |  |
| --- | --- | --- |
| Question: 262 (H) |  |  |
| 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?

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

INSERT INTO employeesVALUES (emp\_id\_seq.NEXTVAL, '&ename', '&jobid', 2000, NULL, &did IN (20,50));

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

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

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);

Question: 263 (K)

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

GRANT select ON dept TO ALL\_USERS;

GRANT select ON dept TO ALL;

GRANT QUERY ON dept TO ALL\_USERS

GRANT select ON dept TO PUBLIC;

Explanation :

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

Question: 264 (G)

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

USER\_CONSTRAINTS

USER\_OBJECTS

ALL\_CONSTRAINTS

USER\_CONS\_COLUMNS

USER\_COLUMNS

Explanation :

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

Question: 265 (K)

Examine these statements:

CREATE ROLE registrar;

GRANT UPDATE ON student\_grades TO registrar; GRANT registrar to user1, user2, user3;

What does this set of SQL statements do?

The set of statements contains an error and does not work.

It creates a role called REGISTRAR, adds the MODIFY privilege on the STUDENT\_GRADES object to the role, and gives the REGISTRAR role to three users.

It creates a role called REGISTRAR, adds the UPDATE privilege on the STUDENT\_GRADES object to the role, and gives the REGISTRAR role to three users.

It creates a role called REGISTRAR, adds the UPDATE privilege on the STUDENT\_GRADES object to the role, and creates three users with the role.

It creates a role called REGISTRAR, adds the UPDATE privilege on three users, and gives the REGISTRAR role to the STUDENT\_GRADES object.

It creates a role called STUDENT\_GRADES, adds the UPDATE privilege on three users, and gives the UPDATE role to the registrar.

Explanation:

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

Question: 266 (E)

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

A subquery should retrieve only one row.

A subquery can retrieve zero or more rows.

A subquery can be used only in SQL query statements.

Subqueries CANNOT be nested by more than two levels.

A subquery CANNOT be used in an SQL query statement that uses group functions.

When a subquery is used with an inequality comparison operator in the outer SQL statement, the column list in the SELECT clause of the subquery should contain only one column.

, F

Explanation:

subquery can retrieve zero or more rows, subquery is used with an inequality camparison operator in the outer SQL statement, the column list in the SELECT clause of the subquery should contain only one column.

Question: 267 (G)

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

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

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

CREATE TABLE student\_grades(student\_id NUMBER(12),semester\_end DATE, gpa NUMBER(4,3), student\_id\_fk FOREIGN KEY (student\_id) REFERENCES students(student\_id));

CREATE TABLE student\_grades(student\_id NUMBER(12),semester\_end DATE, gpa NUMBER(4,3), CONSTRAINT FOREIGN KEY (student\_id) REFERENCES students(student\_id));

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

Explanation :

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

Question: 268 (H)

Evaluate the SQL statement:

TRUNCATE TABLE DEPT;

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

It releases the storage space used by the table.

It does not release the storage space used by the table.

You can roll back the deletion of rows after the statement executes.

You can NOT roll back the deletion of rows after the statement executes.

An attempt to use DESCRIBE on the DEPT table after the TRUNCATE statement executes will display an error.

You must be the owner of the table or have DELETE ANY TABLE system privileges to truncate the DEPT table

, D, F

Explanation :

does not release the storage space used by the table, Can not rollback the deletion of rows after the statement executes, You must be the owner of the table or have DELETE ANY TABLE system privilege to truncate the DEPT table.

Question: 269 (G)

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

The UNIQUE constraint does not permit a null value for the column.

A UNIQUE index gets created for columns with PRIMARY KEY and UNIQUE constraints.

The PRIMARY KEY and FOREIGN KEY constraints create a UNIQUE index.

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

, D

Explanation:

the UNIQUE constraint does not permit a null value for the column The NOT NULL constraint ensure that null value are not permitted for the column

Question: 270 (H)

Which three are true? (Choose three.)

A MERGE statement is used to merge the data of one table with data from another.

A MERGE statement replaces the data of one table with that of another.

A MERGE statement can be used to insert new rows into a table.

A MERGE statement can be used to update existing rows in a table.

, B, C

Explanation:

Merge is used to merge the data of one table with data from another, replaces the data of one table with that of another, used to insert new rows into a table.

Question: 271 (F)

Which is a valid CREATE TABLE statement?

CREATE TABLE EMP9$# AS (empid number(2));

CREATE TABLE EMP\*123 AS (empid number(2));

CREATE TABLE PACKAGE AS (packid number(2));

CREATE TABLE 1EMP\_TEST AS (empid number(2));

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).

Question: 272 (A)

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?

difference, projection, join

selection, projection, join

selection, intersection, join

intersection, projection, join

difference, projection, product

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.

Question: 273 (C)

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

string

character

integer

calendar

numeric

translation

date

conversion

, E,G, H

References EMPLOYEE\_ID column

Foreign key to DEPARTMENT\_ID column of the DEPARTMENTS table

Explanation :

SQL have character, numeric, date, conversion function.

Question: 274 (J)

Examine the structure of the EMPLOYEES table:

EMPLOYEE\_ID NUMBER NOT NULL, Primary Key

EMP\_NAME VARCHAR2(30)

JOB\_ID NUMBER

SAL NUMBER

MGR\_ID NUMBER

DEPARTMENT\_ID NUMBER

You created a sequence called EMP\_ID\_SEQ in order to populate sequential values for the EMPLOYEE\_ID column of the EMPLOYEES table. Which two statements regarding the EMP\_ID\_SEQ sequence are true? (Choose two.)

You cannot use the EMP\_ID\_SEQ sequence to populate the JOB\_ID column.

The EMP\_ID\_SEQ sequence is invalidated when you modify the EMPLOYEE\_ID column.

The EMP\_ID\_SEQ sequence is not affected by modifications to the EMPLOYEES table.

Any other column of NUMBER data type in your schema can use the EMP\_ID\_SEQ sequence.

The EMP\_ID\_SEQ sequence is dropped automatically when you drop the EMPLOYEES table.

The EMP\_ID\_SEQ sequence is dropped automatically when you drop the EMPLOYEE\_ID column.

, D

Explanation :

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

Question: 275 (B)

Evaluate these two SQL statements:

SELECT last\_name, salary, hire\_dateFROM EMPLOYEES ORDER BY salary DESC; SELECT last\_name, salary, hire\_dateFROM EMPLOYEES ORDER BY 2 DESC;

What is true about them?

The two statements produce identical results.

The second statement returns a syntax error.

There is no need to specify DESC because the results are sorted in descending order by default.

The two statements can be made to produce identical results by adding a column alias for the salary column in the second SQL statement.

Explanation :

the two statement produce identical results as ORDER BY 2 will take the second column as sorting column.

Question: 276 (H)

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?

MERGE INTO new\_employees c USING employees e

ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN

UPDATE SET

C.name = e.first\_name ||','|| e.last\_name WHEN NOT MATCHED THEN

INSERT value

S(e.employee\_id, e.first\_name ||', '||e.last\_name);

MERGE new\_employees c USING employees e

ON (c.employee\_id = e.employee\_id)

WHEN EXISTS THEN

UPDATE SET

C.name = e.first\_name ||','|| e.last\_name WHEN NOT MATCHED THEN INSERT valueS(e.employee\_id, e.first\_name ||', '||e.last\_name);

C. MERGE INTO new\_employees cUSING employees e ON (c.employee\_id = e.employee\_id)

WHEN EXISTS THEN UPDATE SET

C.name = e.first\_name ||','|| e.last\_name WHEN NOT MATCHED THEN

INSERT value

S(e.employee\_id, e.first\_name ||', '||e.last\_name);

D. MERGE new\_employees c

FROM employees e ON (c.employee\_id = e.employee\_id) WHEN MATCHED THEN

UPDATE SET

c.name = e.first\_name ||','|| e.last\_name WHEN NOT MATCHED THEN INSERT INTO

new\_employees valueS(e.employee\_id, e.first\_name ||', '||e.last\_name);

Explanation :

this is the correct MERGE statement syntax

Question: 277 (C)

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

SELECT

TO\_CHAR(TO\_DATE(“19-Mar-2001’, ‘DD-Mon-YYYY’), ‘fmDdspth “of” Month YYYY fmHH:MI:SS AM’) NEW\_DATE

FROM dual;

SELECT

TO\_CHAR(TO\_DATE(’19-Mar -2001’, ‘DD-Mon -YYYY’), ‘Ddspth “of” Month YYYY fmHH:MI:SS AM’) NEW\_DATE

FROM dual;

SELECT

TO\_CHAR(TO\_DATE(’19-Mar -2001’, ‘DD-Mon -YYYY’), ‘fmDdspth “of” Month YYYY

HH:MI:SS AM’) NEW\_DATE D. SELECT

TO\_CHAR(TO\_DATE(’19-Mar -2001’, ‘DD-Mon -YYYY), ‘fmDdspth “of” Month YYYYfmtHH:HI:SS AM') NEW\_DATE FROM dual;

Question: 278 (K)

Which one is a system privilege?

SELECT

DELETE

EXECUTE

ALTER TABLE

CREATE TABLE

Question: 279 (A)

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

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

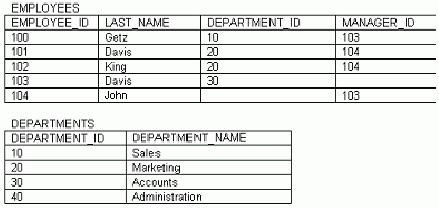
/SQL\*Plus recognizes SQL statements and sends them to the server; SQL is the Oracle proprietary interface for executing SQL statements.

/SQL\*Plus is a language for communicating with the Oracle server to access data; SQL recognizes SQL statements and sends them to the server.

SQL manipulates data and table definitions in the database; /SQL\*Plus does not allow manipulation of values in the database.

Question: 280 (D)

Exhibit



Examine the data in the EMPLOYEES and DEPARTMENTS tables.

You want to retrieve all employees’ last names, along with their manager’s last names and their department names. Which query would you use?

SELECT last\_name, manager\_id, department\_name FROM employees e

FULL OUTER JOIN departments d ON (e.department\_id = d.department\_id);

SELECT e.last\_name, m.last\_name, department\_name FROM employees e

LEFT OUTER JOIN employees m on ( e.managaer\_id = m.employee\_id) LEFT OUTER JOIN departments d ON (e.department\_id = d.department\_id);

SELECT e.last\_name, m.last\_name, department\_name FROM employees e

RIGT OUTER JOIN employees m on ( e.manager\_id = m.employee\_id) LEFT OUTER JOIN departments d ON (e.department\_id = d.department\_id);

SELECT e.last\_name, m.last\_name, department\_name FROM employees e

LEFT OUTER JOIN employees m on ( e.manager\_id = m.employee\_id) RIGT OUTER JOIN departments d ON (e.department\_id = d.department\_id);

SELECT e.last\_name, m.last\_name, department\_name FROM employees e

RIGHT OUTER JOIN employees m on ( e.manager\_id = m.employee\_id) RIGHT OUTER JOIN departments d ON (e.department\_id = d.department\_id)

SELECT last\_name, manager\_id, department\_name FROM employees e

JOIN departments d ON (e.department\_id = d.department\_id) ;

Question: 281 (H)

Examine the structure of the EMPLOYEES table:

EMPLOYEE\_ID NUMBER Primary Key

FIRST\_NAME VARCHAR2(25)

LAST\_NAME VARCHAR2(25)

HIRE\_DATE DATE

Which INSERT statement is valid?

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date) VALUES ( 1000, ‘John’, ‘Smith’, ‘01/01/01’);

INSERT INTO employees(employee\_id, first\_name, last\_name, hire\_date) VALUES ( 1000, ‘John’, ‘Smith’, ’01 January 01’);

INSERT INTO employees(employee\_id, first\_name, last\_name, Hire\_date) VALUES ( 1000, ‘John’, ‘Smith’, To\_date(‘01/01/01’));

INSERT INTO employees(employee\_id, first\_name, last\_name, hire\_date)

VALUES ( 1000, ‘John’, ‘Smith’, 01-Jan-01);

Question: 282 (I)

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?

CREATE PUBLIC SYNONYM EDL\_VU ON emp\_dept\_loc\_vu;

CREATE PUBLIC SYNONYM EDL:VU FOR mary (emp\_dept\_loc\_vu);

CREATE PUBLIC SYNONYM EDL\_VU FOR emp\_dept\_loc\_vu;

CREATE SYNONYM EDL\_VU

ON emp\_dept\_loc\_vu

FOR EACH USER;

E. CREATE SYNONYM EDL\_VU

FOR EACH USER

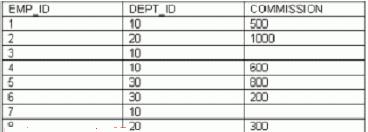
ON emp\_dept\_loc\_vu;

CREATE PUBLIC SYNONYM EDL\_VU ON emp\_dept\_loc\_vu

FOR ALL USERS;

Question: 283 (E)

Exhibit



The COMMISSION column shows the monthly commission earned by the employee. Which two tasks would require subqueries or joins in order to be performed in a single step? (Choose two.)

listing the employees who earn the same amount of commission as employee 3

finding the total commission earned by the employees in department 10

finding the number of employees who earn a commission that is higher than the average commission of the company

listing the departments whose average commission is more that 600

listing the employees who do not earn commission and who are working for department 20 in descending order of the employee ID

listing the employees whose annual commission is more than 6000

, C

Question: 284 (I)

Examine the statement:

Create synonym emp for hr.employees;

What happens when you issue the statement?

An error is generated.

You will have two identical tables in the HR schema with different names.

You create a table called employees in the HR schema based on you EMP table.

You create an alternative name for the employees table in the HR schema in your own schema.

Question: 285 (H)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?

UPDATE employees SET first\_name = ‘John’ SET last\_name = ‘Smith’

WHERE employee\_id = 180;

UPDATE employees SET first\_name = ‘John’, SET last\_name = ‘Smoth’

WHERE employee\_id = 180;

C. UPDATE employee

SET first\_name = ‘John’

AND last\_name = ‘Smith’

WHERE employee\_id = 180;

D. UPDATE employee

SET first\_name = ‘John’, last\_name = ‘Smith’

WHERE employee\_id = 180;

Question: 286 (J)

What is true about sequences?

The start value of the sequence is always 1.

A sequence always increments by 1.

The minimum value of an ascending sequence defaults to 1.

The maximum value of descending sequence defaults to 1.

Question: 287 (B)

The STUDENT\_GRADES table has these columns

STUDENT\_ID NUMBER(12)

SEMESTER\_END DATE

GPA NUMBER(4,3)

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

SELECT student\_id, gpa FROM student\_grades

WHERE semester\_end BETWEEN ’01-JAN-2001’ AND ’31-DEC-2001’ OR gpa > 3.;

SELECT student\_id, gpa FROM student\_grades

WHERE semester\_end BETWEEN ’01-JAN-2001’ AND ’31-DEC-2001’

AND gpa gt 3.0;

SELECT student\_id, gpa FROM student\_grades

WHERE semester\_end BETWEEN ’01-JAN-2001’ AND ’31-DEC-2001’

AND gpa > 3.0;

SELECT student\_id, gpa FROM student\_grades

WHERE semester\_end BETWEEN ’01-JAN-2001’ AND ’31-DEC-2001’ AND gpa > 3.0;

SELECT student\_id, gpa FROM student\_grades

WHERE semester\_end > ’01-JAN-2001’ OR semester\_end < ’31-DEC-2001’ AND gpa >= 3.0;

Question: 288 (D)

Exhibit

ORDERS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ORD\_ID |  | ORD\_DATE | CUST\_ID | ORD\_TOTAL | |
| 100 |  | 12-JAB-2000 | 15 | 10000 | |
| 101 |  | 09-MAR-2000 | 40 | 8000 | |
| 102 |  | 09-MAR-2000 | 35 | 12500 | |
| 103 |  | 15-MAR-2000 | 15 | 12500 | |
| 104 |  | 25-JUN-2000 | 15 | 6000 | |
| 105 |  | 18-JUL-2000 | 20 | 5000 | |
| 106 |  | 18-JUL-2000 | 35 | 7000 | |
| 108 |  | 04-AUG-2000 | 10 | 8000 | |
| CUSTOMERS | | |  |  |  |
| CUST\_ID | | CUST\_NAME | CITY |  |  |
| 10 |  | Smith | Los Angeles | |  |
| 15 |  | Bob | San Francisco | |  |
| 20 |  | Martin | Chicago |  |  |
| 25 |  | Mary | New York |  |  |
| 30 |  | Rina | Chicago |  |  |
| 35 |  | Smith | New York |  |  |
| 40 |  | Linda | New York |  |  |

Evaluate this SQL statement:

SELECT cust\_id, ord\_total FROM orders

WHERE ord\_total > ANY (SELECT ord\_total FROM orders

WHERE cust\_id IN (SELECT cust\_id FROM customers

WHERE city LIKE ‘New York’));

What is the result when the above query is executed?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A. | |  |  |  |  |  |
| CUST\_ID | ORD\_TOTAL | | | |  |  |  |
| 15 |  | 10000 | |  |  |  |  |
| 40 |  | 8000 | |  |  |  |  |
| 35 |  | 12500 | |  |  |  |  |
| 15 |  | 12500 | |  |  |  |  |
| 10 |  | 8000 | |  |  |  |  |
| B. | |  |  |  |  |  |  |
| CUST\_ID | ORD\_TOTAL | | |  | |  |  |
| 15 |  | 10000 | |  |  |  |  |
| 35 |  | 12500 | |  |  |  |  |
| 15 |  | 12000 | |  |  |  |  |
| C. | |  |  |  |  |  |  |
| CUST\_ID | ORD\_TOTAL | | | |  |  |  |
| 15 |  | 10000 | |  |  |  |  |
| 40 |  | 8000 | |  |  |  |  |
| 15 |  | 12000 | |  |  |  |  |
| 15 |  | 6000 | |  |  |  |  |
| 20 |  | 5000 | |  |  |  |  |
| 35 |  | 7000 | |  |  |  |  |
| 20 |  | 6500 | |  |  |  |  |
| 10 |  | 8000 | |  |  |  |  |
| D. | |  |  |  |  |  |  |
| CUST\_ID ORD\_TOTAL | | | |  | |  |  |
| 15 |  | 6000 | |  |  |  |  |
| 20 |  | 5000 | |  |  |  |  |

6500

The query returns no rows.

The query fails because ANY is not a valid operator with a subquery.

Question: 289 (E)

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

A single row subquery can retrieve data from only one table.

A SQL query statement cannot display data from table B that is referred to in its subquery, unless table B is included in the main query’s FROM clause.

A SQL query statement can display data from table B that is referred to in its subquery, without including B in its own FROM clause.

A single row subquery can retrieve data from more than one table.

A single row subquery cannot be used in a condition where the LIKE operator is used for comparison. F. A multiple-row subquery cannot be used in an INSERT statement to insert multiple rows at a time.

, F

Question: 290 (E)

What is true regarding subqueries?

The inner query always sorts the results of the outer query

The outer query always sorts the results of the inner query

The outer query must return a value to the outer query

The inner query returns a value to the outer query

The inner query must always return a value or the outer query will give an error

Question: 291 (C)

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

Convert any date literal to a date

Convert any numeric literal to a date

Convert any character literal to a date

Convert any date to a character literal

Format ’10-JAN-99’ to ‘January 10 1999’

Question: 292 (B)

EXHIBIT, Emp Table

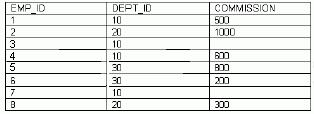


Exhibit A



Exhibit B



Examine the data from the EMP table.

Evaluate this SQL statement:

SELECT \* FROM emp

WHERE commission = (SELECT commission

FROM emp

WHERE emp\_id=3);

What is the result when the query is executed?

Exhibit A

Exhibit B

The query returns no rows

The query fails because the outer query is retrieving more than one column

The query fails because both the inner and outer queries are retrieving data from the same table.

Question: 293

Which three statements about subqueries are true? (Choose three.)

A main query can have more than one subquery.

A subquery can have more than one main query.

The subquery and main query must retrieve data from the same table.

The subquery and main query can retrieve data from different tables.

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

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

, C, E

Question: 294

Evaluate this SQL statement:

SELECT e.emp\_name, d.dept\_name

FROM employees e

JOIN departments d

USING (department\_id)

WHERE d.department\_id NOT IN (10,40)

ORDER BY dept\_name;

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

remove the ORDER BY clause

remove the table alias prefix from the WHERE clause

remove the table alias from the SELECT clause

prefix the column in the USING clause with the table alias

prefix the column in the ORDER BY clause with the table alias

replace the condition ”d.department\_id NOT IN (10,40)” in the WHERE clause with

”d.department\_id <> 10 AND d.department\_id <> 40”

Question: 295

Evaluate the SQL statement:

SELECT LPAD (salary,10,\*)

FROM EMP

WHERE EMP\_ID = 1001;

If the employee with the EMP\_ID 1001 has a salary of 17000, what is displayed?

17000.00

17000\*\*\*\*\*

\*\*\*\*170.00

\*\*17000.00

an error statement