KillTest

质量更高　服务更好

学习资料

http://www.killtest.cn

半年免费更新服务

Exam : **1Z0-061**

Title : Oracle Database 12c: SQL Fundamentals

Version : DEMO

1.Topic 1, Main Questions

Evaluate the following SQL statement:

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

Which statement is true regarding the outcome of the above query?
A. It executes successfully and displays rows in the descending order of PROMO_CATEGORY.
B. It produces an error because positional notation cannot be used in the order by clause with set operators.
C. It executes successfully but ignores the order by clause because it is not located at the end of the compound statement.
D. It produces an error because the order by clause should appear only at the end of a compound query-that is, with the last select statement.
Answer: D

2.View the Exhibit and examine the structure of the product, component, and PDT_COMP tables.
In product table, PDTNO is the primary key.
In component table, COMPNO is the primary key.
In PDT_COMP table, <PDTNO, COMPNO) is the primary key, PDTNO is the foreign key referencing PDTNO in product table and COMPNO is the foreign key referencing the COMPNO in component table.
You want to generate a report listing the product names and their corresponding component names, if the component names and product names exist.
Evaluate the following query:
SQL>SELECT pdtno, pdtname, compno, compname
FROM product _____ pdt_comp
USING (pdtno) _____ component USING (compno)
WHERE compname IS NOT NULL;
Which combination of joins used in the blanks in the above query gives the correct output?

PRODUCT

Name	Null?	Type
PDTNO	NOT NULL	NUMBER(3)
PDTNAME		VARCHAR2(25)
QTY		NUMBER(6,2)

COMPONENT

Name	Null?	Type
COMPNO	NOT NULL	NUMBER(4)
COMPNAME		VARCHAR2(25)
QTY		NUMBER(6,2)

PDT_COMP

Name	Null?	Type
PDTNO	NOT NULL	NUMBER(2)
COMPNO	NOT NULL	NUMBER(3)

A. JOIN; JOIN
B. FULL OUTER JOIN; FULL OUTER JOIN
C. RIGHT OUTER JOIN; LEFT OUTER JOIN
D. LEFT OUTER JOIN; RIGHT OUTER JOIN
Answer: C

3.View the Exhibit for the structure of the student and faculty tables.

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

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

You need to display the faculty name followed by the number of students handled by the faculty at the base location.

Examine the following two SQL statements:

Statement 1

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

Statement 2

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

Which statement is true regarding the outcome?

A. Only statement 1 executes successfully and gives the required result.

B. Only statement 2 executes successfully and gives the required result.

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

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

Answer: D

4.View the Exhibits and examine products and sales tables.

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

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

You issue the following query to display product name and the number of times the product has been sold:

```
SQL>SELECT p.prod_name, i.item_cnt
     FROM (SELECT  prod_id, COUNT(*) item_cnt
                 FROM sales
                 GROUP BY prod_id) i RIGHT OUTER JOIN  products p
     ON  i.prod_id  = p.prod_id;
```

What happens when the above statement is executed?

A. The statement executes successfully and produces the required output.

B. The statement produces an error because item_cnt cannot be displayed in the outer query.

C. The statement produces an error because a subquery in the from clause and outer-joins cannot be used together.

D. The statement produces an error because the group by clause cannot be used in a subquery in the from clause.

Answer: A

5.You want to create a table employees in which the values of columns EMPLOYEES_ID and LOGIN_ID must be unique and not null.
Which two SQL statements would create the required table?

A)
```
CREATE TABLE employees(
employee_id NUMBER,
login_id NUMBER,
employee_name VARCHAR2(25),
hire_date DATE,
CONSTRAINT emp_id_pk PRIMARY KEY (employee_id, login_id));
```

B)
```
CREATE TABLE employees(
employee_id NUMBER CONSTRAINT emp_id_pk PRIMARY KEY,
login_id NUMBER UNIQUE,
employee_name VARCHAR2(25),
hire_date DATE);
```

C)
```
CREATE TABLE employees(
employee_id NUMBER,
login_id NUMBER,
employee_name VARCHAR2(100),
hire_date DATE,
CONSTRAINT emp_id_uk UNIQUE (employee_id, login_id));
```

D)
```
CREATE TABLE employees(
employee_id NUMBER,
login_id NUMBER,
employee_name VARCHAR2(100),
hire_date DATE,
CONSTRAINT emp_id_uk UNIQUE (employee_id, login_id),
CONSTRAINT emp_id_nn NOT NULL (employee_id, login_id));
```

E)
```
CREATE TABLE employees(
employee_id NUMBER,
login_id NUMBER,
employee_name VARCHAR2(100),
hire_date DATE,
CONSTRAINT emp_id_uk UNIQUE (employee_id, login_id),
CONSTRAINT emp_id_nn NOT NULL (employee_id, login_id));
```

F)
```
CREATE TABLE employees(
employee_id NUMBER CONSTRAINT emp_id_nn NOT NULL,
login_id NUMBER CONSTRAINT login_id_nn NOT NULL,
employee_name VARCHAR2(100),
hire_date DATE,
CONSTRAINT emp_num_id_uk UNIQUE (employee_id, login_id));
```

A. Option A

B. Option B
C. Option C
D. Option D
E. Option E
F. Option F
Answer: A,F