

更に上のクオリティ  
更に上のサービス!

問題集

**ITEXAMPASS**

<https://www.itexampass.jp>



1年で無料進級することに提供する

**Exam** : **1Z0-061J**

**Title** : Oracle Database 12c: SQL  
Fundamentals

**Version** : DEMO

## 1. トピック 1、主な質問

次の SQL 文を評価します。

```
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';
```

上記のクエリの結果に関して、どちらのステートメントが正しいですか？

- A.正常に実行され、PROMO\_CATEGORY の降順で行が表示されます。
- B.集合演算子の order by 節で位置表記を使用できないため、エラーが発生します。
- C.これは正常に実行されますが、複合文の末尾にはないため、order by 句は無視されます。
- D. order by 句は複合クエリの最後、つまり最後の select 文でのみ表示されるため、エラーが発生します。

**Answer: D**

## 2. 展示を見て、製品、コンポーネント、および PDT\_COMP テーブルの構造を調べます。

製品表では、PDTNO が主キーです。

コンポーネント表では、COMPNO が主キーです。

PDT\_COMP 表では、<PDTNO、COMPNO> は主キー、PDTNO は製品表の PDTNO を参照する外部キー、COMPNO はコンポーネント表の COMPNO を参照する外部キーです。

コンポーネント名と製品名が存在する場合は、製品名とそれに対応するコンポーネント名をリストしたレポートを生成します。

次のクエリを評価します。

```
SQL> SELECT pdtno, pdtname, compno, compname
FROM product _____ pdt_comp
USING (pdtno) _____コンポーネント USING (compno)
WHERE compname は NULL ではありません。
```

上記のクエリーでブランクに使用された結合の組み合わせによって、正しい出力が得られますか？

**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. ジョイン; ジョイン
- B. フル・アウト・ジョイン; フル・アウター・ジョイン
- C. 右外側のジョイン。 LEFT OUTER JOIN
- D. LEFT OUTER JOIN; 右外側のジョイン

**Answer: C**

- 3. 学生と教員のテーブルの構造については展示をご覧ください。

**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)

教員名を表示し、それに続いて教員が扱う生徒の数をベースロケーションに表示する必要があります。  
次の2つのSQL文を調べます。

**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;
```

結果については、どのステートメントが正しいのですか？

- A.文1のみが正常に実行され、必要な結果が得られます。
- B.文2のみが正常に実行され、必要な結果が得られます。
- C.ステートメント1と2の両方が正常に実行され、異なる結果が得られます。
- D.文1と2はともに正常に実行され、同じ結果が得られます。

**Answer: D**

- 4. 展示品を見て、製品と販売テーブルを調べます。

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)

次の問合せを発行して、製品名および製品が販売された回数を表示します:

```
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;
```

上記のステートメントが実行されるとどうなりますか?

- A.文は正常に実行され、必要な出力が生成されます。
- B.外側の照会で item\_cnt を表示できないため、ステートメントでエラーが発生します。
- C. from 句と外部結合のサブクエリを一緒に使用できないため、文でエラーが発生します。
- D. group 句を from 句のサブクエリで使用できないため、文でエラーが発生します。

**Answer: A**

5. 列 EMPLOYEES\_ID および LOGIN\_ID の値が一意であり、NULL でない必要がある表 employees を作成するとします。

どの2つのSQL文が必要なテーブルを作成しますか？

- 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. オプション A  
B. オプション B  
C. オプション C  
D. オプション D

E. オプション E

F. オプション F

**Answer: A,F**