

ダウンロード版

データベース問題 (5)

学籍番号 **氏名** **学年** **学科** _____.

※ 末尾に解答例を付けているので、各自で解答を考えた後に、末尾を見て確認しておくこと.

1. 下記の表 Product に関する問題

PName	Price	Seller

A	1200	x
B	1010	x
A	1150	y
C	800	y

(1) 下記の SQL の実行結果は何になるか

```
SELECT COUNT(*)
FROM Product
WHERE PName = 'A'
```

※ COUNT (*) は行数のカウント

(2) 下記の SQL の実行結果は何になるか

```
SELECT COUNT(DISTINCT PName)
FROM Product
```

※ DISTINCT は重複行の除去を行うので、COUNT(DISTINCT ...) は、異なる値を持つ行の行数を数える

(3) 下記の SQL の実行結果は何になるか

```
SELECT SUM(Price)
FROM Product
WHERE PName='A'
```

(4) 下記の SQL の実行結果は何になるか

```
SELECT PName, COUNT(*)
FROM Product
GROUP BY PName
```

(5) 下記の SQL の実行結果は何になるか

```
SELECT Seller, COUNT(*)
FROM Product
GROUP BY Seller
```

(6) 下記の SQL の実行結果は何になるか

```
SELECT PName, COUNT(*), SUM(Price), AVG(Price), MAX(Price), MIN(Price)
FROM Product
GROUP BY PName
```

2. 下記の2つの表に関する問題

テーブル名 SALES

PID	Price	Seller
101	1200	x
102	1010	x
101	1150	y
103	800	y

テーブル名 PRODUCT

PID	PNAME
101	A
102	B
103	C

下記の SQL の実行結果は何になるか

```
SELECT SALES.PID, SALES.Price, SALES.Seller
FROM SALES, PRODUCT
WHERE SALES.PID = PRODUCT.PID AND PNAME='A'
```

解答例

問 1 (1) 2 (2) 3 (3) 2350

(4) PName | count

A	2
B	1
C	1
(5) Seller	count

x	2				
y	2				
(6) PName	count	sum	avg	max	min
A	2	2350	1175	1200	1150
B	1	1010	1010	1010	1010
C	1	800	800	800	800

問 2

PID	Price	Seller
101	1200	x
101	1150	y

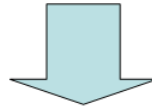
テーブルの分解と結合

A(name, teacher_name), B(name, student_name, score)

name	teacher_name
Database	K
Programming	A

name	student_name	score
Database	KK	85
Database	AA	75
Database	LL	90
Programming	KK	85
Programming	LL	75

```
SELECT A.name, A.teacher_name, B.student_name, B.score
FROM A, B
WHERE A.name = B.name
```



結合問い合わせにより元の
テーブルに戻るので、この
分解は無損失分解である。
The original table 'results' is
constructed.
The decomposition is
LOSSLESS.

name	teacher_name	student_name	score
Database	K	KK	85
Database	K	AA	75
Database	K	LL	90
Programming	A	KK	85
Programming	A	LL	75

■ SQL における DISTINCT 指定

SQL では、SELECT と選択リスト内の間に「DISTINCT」と書く場合がある。これを DISTINCT 指定という。DISTINCT 指定を行うと、問い合わせの結果に重複行があるとき 1 つを残して他を除去する。重複行というときは、行全体が同じ値だという意味である。下記の例では、同じ注文者名が別の商品番号の注文を行った場合には重複行ではない。

```
SELECT DISTINCT 商品番号, 注文者名
FROM 注文
```

DISTINCT 指定があるとき、重複行があるとき 1 つを残して他を除去するから、SQL の評価結果は多重集合にはなることは無い。必ず、リレーションになる。一方で、DISTINCT 指定がないときは、SQL の評価結果が多重集合になる場合がある。補足説明しておく。下記のテーブル results を例に説明する。

name	teacher_name	student_name	score
Database	K	KK	85
Database	K	AA	75
Database	K	LL	90
Programming	A	KK	85
Programming	A	LL	75

このテーブル results から、属性 name と teacher_name だけを抜き出して新しいテーブルを作ったとする。下記のように重複行が現れる。つまり新しく出来たテーブルは多重集合である。

name	teacher_name
Database	K
Database	K
Database	K
Programming	A
Programming	A

上記のテーブルを作るための SQL は次の通りである。

```
SELECT name, teacher_name
FROM results;
```

※ 関係演算で「射影」という演算がある。射影 results[name, student_name] といったときは、属性 name と teacher_name だけを抜き出した上で、重複行があるとき 1 つを残して他を除去することになっている。結果は次の通りである。

```
name          | teacher_name
```

```
-----  
Database      | K  
Programming   | A
```

射影を行うための SQL は、次のように DISTINCT 指定が付く。

```
SELECT DISTINCT name, teacher_name  
FROM   results;
```

■ θ -結合演算

$R(A_1, A_2, \dots, A_n)$ と $S(B_1, B_2, \dots, B_m)$ をリレーシオンとする。R の属性 A_i と S の属性 B_j 上の θ -結合演算を「 $R[A_i \theta B_j]S$ 」と書く。定義は次の通りである。（ θ は比較演算子である）。

$R[A_i \theta B_j]S$ は、R と S の直積集合 ($R \times S$) の中から「 $R.A_i \theta S.B_j$ 」を満足する要素を選んだものの θ -結合演算「 $R[A_i \theta B_j]S$ 」は、SQL を用いて次のように書き下すことができる。※ SQL の規格では、これ以外の書き方もあるが、今回は、この書き方だけを教えておく。

```
SELECT *  
FROM R, S  
WHERE <A_i  $\theta$  B_j>
```

※（参考）リレーショナル代数式を使って次のように書くことができる。

$R[A_i \theta B_j]S = \{ (t, u) \mid t \in R \wedge u \in S \wedge t[A_i] \theta u[B_j] \}$
但し、「 (t, u) 」と書いているのは、 $t = (a_1, a_2, \dots, a_n)$, $u = (b_1, b_2, \dots, b_m)$ とするときに、 $(t, u) = (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$ なる $n + m$ 項のタプルである。

■ 結合問い合わせ（結合質問ともいう）

結合問い合わせとは、問い合わせの中の「**WHERE** <expression>」のところの「<expression>」の中に、 θ -結合演算を含むような問い合わせのことである。結合問い合わせでは、SQL のテーブル参照リストに、2 つ以上のテーブル名が現れる。一方で、結合問い合わせの評価結果は、一般の問い合わせと同様に 1 つのテーブルであることに注意して欲しい。結合問い合わせでは、表参照リストに登場する表の結合が行われる。

SQL 問い合わせのプログラムは、SELECT, FROM, WHERE を使うのが基本である。文法は次のようになる。※ SQL の文法は多彩なので、ここでは、基本的な場合に説明を絞る。

```
SELECT <list-of(result-column)>  
FROM <list-of(table-name)>
```

あるいは

```
SELECT <list-of(result-column)>  
FROM <list-of(table-name)>  
WHERE <expression>
```

■ SQL における集約関数

集約関数は、COUNT, SUM, AVG, MAX, MIN などのように、値の集約を行うもの。

- ・COUNT（数え上げ）
行数を数えるもので、COUNT(*), COUNT(DISTINCT <1 個以上の属性値の並び>) のような記法が多用される。COUNT(*) は行数、COUNT(DISTINCT <1 個以上の属性値の並び>) は重複行を除去した後の行数である。
- ・SUM（総和）、AVG（平均）、MAX（最大）、MIN（最小）
特定の列について集約を行う。総和や平均や最大や最小が評価結果として得られる。

■ GROUP BY 句

GROUP BY 句は、集約の単位になるグループを指定する。GROUP BY 句が省略されている場合は、表全体が集約の単位になる（探索条件が指定されている場合には、探索結果全体が集約の単位）。

SQL で集約を行う場合、SELECT の後の「<list-of(result-column)>」のところ（「選択リスト」ともいう）には、GROUP BY で指定した列名と、集約関数だけしか書けないことに注意。