ダウンロード版

データベース問題　（4）

学籍番号　　氏名　　学年　　学科　.

学籍番号、氏名は全員記入をお願いします。

1．SQL は，集合で無く，多重集合を基礎とする体系である．では，ここでの，多重集合の定義を書きなさい．

2．リレーショナルデータモデルでは，リレーションはドメインの直積の有限部分集合であると定義されていた．「集合」なので，全く同じ属性値の並びからなる重複したタップルの存在は許されない．一方で，SQL では，集合でなく，多重集合を扱う（そのため，SQL では，「リレーション」という用語を用いずに，「表」あるいは「テーブル」という用語を用いる）．多重集合を扱う理由は，現実のデータ操作においては，表の中に，全く同じ属性値を持つ重複した行の存在を許した方が都合の良い場合があるからである．　では，具体的には，どのような場合か，例を挙げて説明しなさい．

3．下記の 2 つの SQL の相違点を説明しなさい．

※　同じ科目が，複数の教員によって開講されているような場合である．

(1) SELECT 科目名,単位数
　FROM 科目;

(2) SELECT DISTINCT 科目名,単位数
　FROM 科目;

４．下記の SQL 文を説明しなさい

(1)
```
SELECT *
FROM 商品
```

(2)
```
SELECT 商品番号
FROM 商品
```

(3)
```
SELECT 定価, 定価 + 100
FROM 商品
```

(4)
```
SELECT 科目名
FROM 科目
WHERE 単位数 >= 4
```

(5)
```
SELECT 科目番号
FROM 科目
WHERE 科目名 = 'データベース'
```

(6)
```
SELECT 被験者名
FROM 被験者
WHERE 中性脂肪 >= 100 AND 中性脂肪 <=200
```

データベース　（４）

■　リレーション (relation)　（「関係」とも呼ぶ）

D1, D2, … , Dn をドメインとするとき，D1, D2, …, Dn 上のリレーションとは，直積集合「D1×D2× … ×Dn」の任意の有限部分集合を言う．例えば，8 つの属性 id, year, month, day, customer_name, product_name, unit_price, qry があり，それぞれのドメインが次のようになっているとする．

id のドメイン: 整数
year のドメイン: {2009, 2010, ...}
month のドメイン: {1, 2, ..., 12}
day のドメイン: {1, 2, ..., 31}
customer_name のドメイン: 文字列
product_name のドメイン: 文字列
unit_price のドメイン: 0 よりも大きい実数
qty のドメイン: 0 よりも大きい整数

このとき，次の図で表される集合は 1 つのリレーションである．ここでは，ドメインの直積集合の要素を 1 つの行として書いている．

1	2009	10	26	kaneko	orange A	1.2	10
2	2009	10	26	miyamoto	Apple M	2.5	2
3	2009	10	27	kaneko	orange B	1.2	8
4	2009	19	28	miyamoto	Apple L	3	1

■ 多重集合（マルチ集合ともいう）（multi-set）

要素の重複を許すような集合のことを多重集合（マルチ集合）という．リレーショナルデータベースにおいて多重集合という概念が導入されている理由を，具体例で説明すると下記の通りである．

```
name           | score |  student_name
---------------------------------------
Database       |   80  |  KK
Database       |   95  |  AA
Database       |   80  |  LL
Programming    |   85  |  KK
Programming    |   75  |  LL
```

name = Database （科目名が「Database」）である行の score (得点) は {80, 95, 80} という多重集合である．この多重集合を見て，平均点 85，受講者 3 名ということが分かる．多重集合という概念がない場合には，name (科目名) A の得点は {80, 95} という集合になり，平均点や受講者数が分からず困ったことになる．

■　テーブル（表とも言う）(table)

リレーショナルデータベースでは，データベースをテーブルの集まり (collection of tables) として記述する． リレーショナルデータベースでのテーブルは，ヘッダと本体からなる．

・ヘッダ: 属性名が並ぶ.
・本体: 属性のドメインの直積集合の要素からなる多重集合

【テーブルの例（本体に重複値がない)】

id	year	month	day	customer name	product name	unit price	qty	created at	updated at
1	2009	10	26	kaneko	orange A	1.2	10		
2	2009	10	26	miyamoto	Apple M	2.5	2		
3	2009	10	27	kaneko	orange B	1.2	8		
4	2009	19	28	miyamoto	Apple L	3	1		

※各行が挿入された日時 ※ 各行の最終更新日時

【テーブルの例（本体に重複値がある)】

得点
80
95
80

以上のように，テーブルの本体にはリレーションを格納することができる（リレーショナルデータベースに「リレーショナル」という言葉が付いているのはそのため）し， 重複値を持つような多重集合を格納することもできる．以上のように，**テーブル（あるいは表）というときは，多重集合を基礎とする体系**になっている．

■ テーブル名 (table name)

テーブル名とは，個々のテーブルに付けられた名前のこと．

■ **属性 (attribute)**

リレーションデータベースのテーブルでは，テーブルの各列が 1 つの属性をなす． 属性名は各列に付けられた名前であり，テーブルのヘッダに書かれる． 例えば，下記のテーブルには，id, year, month, day, customer_name, product_name, unit_price, qty という名前の属性がある．

id	year	month	day	customer name	product name	unit price	qty	created at	updated at
1	2009	10	26	kaneko	orange A	1.2	10		
2	2009	10	26	miyamoto	Apple M	2.5	2		
3	2009	10	27	kaneko	orange B	1.2	8		
4	2009	19	28	miyamoto	Apple L	3	1		

※各行が挿入された日時 ※ 各行の最終更新日時

※ テーブルのタップルが更新（例えば，テーブルに新しいタップルが挿入されたり，テーブルからタップルが削除されたり，タップルの値が変化したり）されても，属性名は不変である（変化しない）．

■ **行 (row)**

リレーショナルデータベースで「行」というときは，テーブルの本体の各行のこと．

■ **リレーショナルデータモデル**

リレーショナルデータモデルとは，次の特徴を持ったデータモデルである．

１．データベースを，テーブルの集まりとして記述

２．データ操作の体系は，リレーショナル代数演算やリレーショナル論理（リレーショナル代数演算とリレーショナル論理は等価であるといってよい）．

３．リレーショナルデータモデルの一貫性制約には，主キー制約，一意制約，非空制約，ドメイン制約，参照整合性制約などがある．

■ **SQL**

SQL は，リレーショナルデータベース言語の国際標準である．

・リレーショナル代数やリレーショナル論理を忠実にデータベース言語化したものではなく，平易な英文として，簡単に読み下せるようになっている．

・リレーショナル代数やリレーショナル論理には無い演算として，集約演算などがある．

・リレーショナル代数やリレーショナル論理がリレーショナルデータベースの操作を体系化しているのに対して，SQL は，リレーショナルデータベースのデータ操作以外にも，データベーススキーマの定義，トランザクション管理などの機能を持つ．

※ SQL では「リレーション」ではなく「表」を扱うことに注意して欲しい．「リレーション」と「表」はともに 2 次元の表の形になっているという意味では似ている．「リレーション」はドメインの直積の有限部分集合なので，同じタップルが重複して現れることはない．一方で，「表」は，同じ値をもった行が複数回現れることが許される．

■ **SQL の比較演算子**

比較演算子は 2 項述語である． SQL で，数値や文字列の比較演算子には次のようなものがある

<
<=
=
>=
>
<>

■ SQL 問い合わせ (SQL query)

SQL 問い合わせは，リレーショナルデータベースに格納された 1 つまたは複数のテーブルを使う．問い合わせの評価結果として，リレーショナルデータベース内のテーブルをそのままの形で得るということはめったになく，新しいテーブルが 1 つ作られる．

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>

以上のことを例を使って説明する．

SELDCT A1, A2, ..., An
FROM T1, T2, ,,,, Tm
WHERE <expression>

のように書くと，m 個のテーブル T1, T2, ..., Tm の直積集合から <expression> を満足する行のみを選び，そうして出来たテーブルの属性 A1, A2, ..., An を出力するという意味になる．
WHERE の部分が無い場合，例えば，

SELDCT A1, A2, ..., An
FROM T1, T2, ,,,, Tm

のように書くと，m 個のテーブル T1, T2, ..., Tm の直積集合から属性 A1, A2, ..., An を出力するという意味になる．

● SELECT <list-of(result-column)>

SELECT の後には，問い合わせの結果として得たい列名あるいは列名を含む式の，1 個以上の並びを書く． 2 個以上ある場合には半角のカンマ「,」で区切る．問い合わせ結果として得られるテーブルの中の列の並びは，SQL の SELECT に書いた列名あるいは列名を含む式の並びの順序通りになる．

SELECT の後に列名を書くとき， 「<テーブル名>.<列名>」のように，「<テーブル名>.」を前に付けねばならない場合がある． これは，FROM に指定するテーブルが 2 個以上あり，しかも，これらのテーブルが同じ属性名の属性を持つ場合である．例えば， 「科目」と「履修」という 2 つのテーブルが，同じ属性名「科目番号」を持つ場合，下記のように「科目番号」の前に「科目.」を付けるなどして，あいまいさを排除する必要がある．

```
SELECT 科目.科目番号, 科目名
FROM 科目, 履修
WHERE 科目.科目番号 = 履修.科目番号 AND 履修.学生番号 = 00001;
```

列名として「*」あるいは「<テーブル名>.*」のような書き方ができ，これは、全ての属性名を並べて書 いたのと同じ意味を持つ．

● FROM <list-of(table-name)>

FROM の後には，テーブル名の 1 個以上の並びを書く．2 個以上ある場合には半角のカンマ「,」で区切る．

テーブル参照リストの中で，タップル変数を定義することもできる． 例えば，下記の SQL 文では X と Y の 2 つのタップル変数が定義されている．

```
SELECT X.社員番号, Y.社員番号
FROM 社員 X, 社員 Y
WHERE X.部長 = Y.社員番号 AND X.給与 > Y.給与
```

● WHERE <expression>

WHERE の後には探索条件を書く．複数の探索条件を組み合わせたい場合には，（半角のカンマではなく）論理演算の AND, OR を用いる． 探索条件は，FROM で指定したテーブルの属性名や，FROM で指定したタップル変数の変数名を含む式である． 探索条件に使用できるキーワードとしては次のものがある．

比較演算（<, <=, =, >=, >, <>）
論理演算（AND, OR, NOT）
各種の述語（BETWEEN, IN, LIKE, NULL, EXIST など）
など