

パズル 3 1 すべての製品を購入した顧客 関係除算の応用

【問題】

次に示すような顧客、注文、注文明細（及びそのたもろもろ）の典型的なスキーマが含まれていた。

```
CREATE TABLE Customers
(customer_id INTEGER NOT NULL PRIMARY KEY,
acct_balance DECIMAL (12,2) NOT NULL,
```

Customers : 顧客テーブル customer_id : 顧客 ID acct_balance : 売掛金残高

```
CREATE TABLE Orders
(customer_id INTEGER NOT NULL,
order_id INTEGER NOT NULL PRIMARY KEY,
```

Orders : 注文テーブル customer_id : 顧客 ID order_id : 注文 ID

```
CREATE TABLE OrderDetails
(order_id INTEGER NOT NULL,
item_id INTEGER NOT NULL,
item_qty INTEGER NOT NULL,
PRIMARY KEY (order_id, item_id),
...);
```

OrderDetails : 注文明細テーブル order_id : 注文 ID item_id : 製品 ID
Item_qty : 注文数量

```
CREATE TABLE Products
(item_id INTEGER NOT NULL PRIMARY KEY,
item_qty_on_hand INTEGER NOT NULL,
...);
```

Products : 製品テーブル item_id : 製品 ID item_qty_on_hand : 現存数量

このテーブル群を使ったサンプル問題の1つに、すべての製品を購入した顧客全員の売掛金残高の平均と、すべてではないがいくつかの製品を購入した顧客全員の売掛金残高の平均を求める。

①

[クエリ] お店の製品を購入した顧客全員の売掛金残高の平均。

```
SELECT avg(acct_balance) FROM
(SELECT cust_id, acct_balance
FROM (SELECT *
      FROM Customers C, Orders O, OrderDetails OD
      WHERE O.ord_id = OD.ord_id
      AND O.cust_id = C.cust_id )
GROUP BY cust_id
HAVING count(distinct item_id) == (
SELECT count(*)
FROM Products ) ) ;
```

[説明] まず、Customers, Orders, OrderDetails を自然結合する。
そして、cust_id でグループ化をし、それぞれのグループの item_id
の数が製品の種類の数と等しければ、

お店の製品を購入した顧客が求まる。(製品の種類の
数は一番内側のサブクエリで求まる)

次に、これをサブクエリとして、お店の製品を購入した顧客の cust_id
と acct_balance のリストを返す。そして、一番外側の

SELECT はこれらの acct_balance の平均値を出力する。

[クエリ] すべてではないがいくつかの製品を購入した顧客全員の acct_balance の平均を算出する。

```
SELECT      avg (acct_balance) FROM
C SELECT    cust_id, acct_balance
FROM ( SELECT *
      FROM Customers C, Orders O, OrderDetails
      WHERE O.ord_id = OP.ord_id
      AND   O.cust_id = C.cust_id )
GROUP BY   cust_id
HAVING     count (distinct item_id) <> (
          SELECT count (*)
          FROM Products ) ) ;
```

[説明] クエリ ① と同じ"考え"だが、HAVING の条件は item_id の数が製品のすべての種類の数と等しくないものに書きかえる。