

# SQLでTOP3ランキングをやりたい

2016年8月14日

PostgreSQL 9.5.4 で説明してありますがOracle でも可能な筈。

夏休み直前に後輩Hシー氏から振られた話題を記録しておく。

RANK関数でTOP3ランキングをやってみる

## 下準備

今回のデータは、[SQLでデータベーステーブルの縦・横変換](#)で使っていたデータを流用します。

```
$ psql -h 192.168.1.28 kouhai kouhai
psql (9.5.3, サーバ 9.5.4)
"help" でヘルプを表示します.
```

```
kouhai=> \d
リレーションの一覧
スキーマ | 名前 | 型 | 所有者
-----+-----+-----+-----
public   | 明細 | テーブル | kouhai
(1行)
```

```
kouhai=> \d 明細
テーブル "public.明細"
列 | 型 | 修飾語
-----+-----+-----
売上年 | CHARACTER VARYING(4) |
商品 | CHARACTER VARYING(10) |
売上 | NUMERIC(5,0) |
```

```
kouhai=> SELECT * FROM 明細;
```

```
売上年 | 商品 | 売上
-----+-----+-----
2004   | みかん | 40
2004   | りんご | 37
2004   | ばなな | 24
2004   | すもも | 15
2004   | ぶどう | 5
2005   | みかん | 60
2005   | ばなな | 30
2005   | りんご | 29
2005   | すもも | 10
2005   | ぶどう | 3
2006   | みかん | 90
2006   | まんごー | 88
2006   | さくらんぼ | 70
```

```

2006 | ぶどう | 10
2006 | めろん | 6
2007 | ここあ | 290
2007 | だいち | 100
(17行)

```

kouhai=>

## ベースとなるクエリ

商品毎の売上合計	売上年毎の売上合計
<pre> SELECT 商品       ,SUM(売上) AS 合計 FROM 明細 GROUP BY 商品 ORDER BY 合計 DESC ; </pre>	<pre> SELECT 売上年       ,SUM(売上) AS 合計 FROM 明細 GROUP BY 売上年 ORDER BY 合計 DESC ; </pre>
<pre> 商品   合計 -----+----- ここあ   290 みかん   190 だいち   100 まんごー   88 さくらんぼ   70 りんご   66 ばなな   54 すもも   25 ぶどう   18 めろん   6 (10行) </pre>	<pre> 売上年   合計 -----+----- 2007   390 2006   264 2005   132 2004   121 (4行) </pre>

これらのクエリがベースとなります。

## 商品毎のランキング

商品毎の結果にランキングを付与してみます。

売上トップのランキング	売上ワーストのランキング
<pre> SELECT 商品       ,SUM(売上) AS 合計       ,RANK() OVER ( ORDER BY SUM(売上) DESC ) AS ランキング FROM 明細 GROUP BY 商品 ; </pre>	<pre> SELECT 商品       ,SUM(売上) AS 合計       ,RANK() OVER ( ORDER BY SUM(売上) ) AS ランキング FROM 明細 GROUP BY 商品 ; </pre>

売上トップのランキング			売上ワーストのランキング		
商品	合計	ランキング	商品	合計	ランキング
-----+-----+-----			-----+-----+-----		
ここあ	290	1	めろん	6	1
みかん	190	2	ぶどう	18	2
だいち	100	3	すもも	25	3
まんごー	88	4	ばなな	54	4
さくらんぼ	70	5	りんご	66	5
りんご	66	6	さくらんぼ	70	6
ばなな	54	7	まんごー	88	7
すもも	25	8	だいち	100	8
ぶどう	18	9	みかん	190	9
めろん	6	10	ここあ	290	10
(10行)			(10行)		

RANK()関数はウィンドウ関数で、指定された範囲内(ウィンドウ内)での現在行の位置に対応した値を返します。ウィンドウの範囲はOVER句で指定されますが、このクエリにはPARTITION BY句による範囲指定がないので、テーブル内のレコード全体が対象になります。

ウィンドウ範囲の行並びについてはOVER句の指定にあるORDER BY句でソートをさせて決めています。この例ではSUM(売上)の結果をソートさせています。このソートはあくまでウィンドウ範囲内でのソートで、この例ではウィンドウ範囲がテーブル内のレコード全体なので、結果は合計でソートされていますが、慎重を期すならベースクエリにあった“ORDER BY 合計 DESC”の記述も行った方が良いです。

## 売上年別でランキング

次に売上年別の結果にランキングを付与してみます。

売上年別のランキング		
SELECT 売上年		
	,SUM(売上) AS 合計	
	,RANK() OVER ( ORDER BY SUM(売上) DESC) AS ランキング	
FROM 明細		
GROUP BY 売上年		
;		
売上年	合計	ランキング
-----+-----+-----		
2007	390	1
2006	264	2
2005	132	3
2004	121	4
(4行)		

集計単位を商品から売上年に切り替えただけで、結果は予想通りですね。こちらも、慎重を期すならベースクエリにあった“ORDER BY 合計 DESC”の記述も行った方が良いです。

## 売上年別の商品のランキング

売上年別の商品売上ランキングを出してみます。

**売上年別の商品売上ランキング**

```

SELECT 売上年
       , 商品
       , SUM(売上) AS 合計
       , RANK() OVER ( PARTITION BY 売上年 ORDER BY SUM(売上) DESC ) AS ランキング
FROM 明細
GROUP BY 売上年, 商品
;

```

売上年	商品	合計	ランキング
2004	みかん	40	1
2004	りんご	37	2
2004	ばなな	24	3
2004	すもも	15	4
2004	ぶどう	5	5
2005	みかん	60	1
2005	ばなな	30	2
2005	りんご	29	3
2005	すもも	10	4
2005	ぶどう	3	5
2006	みかん	90	1
2006	まんごー	88	2
2006	さくらんぼ	70	3
2006	ぶどう	10	4
2006	めろん	6	5
2007	ここあ	290	1
2007	だいず	100	2

(17行)

ORVER句で指定しているPARTITION BY句で、RANK()関数が算出する行位置の範囲を指定します。ここでは 売上年 を指定するので、売上年毎にSUM(売上)でソートが行われRANK()関数は売上年が変わると改めて行位置を1から計算して返すこととなります。

**TOP3を出すにはどうするのさ？**

もうお分かりの通り、RANK()関数の結果でフィルターすれば良いのです。

例えば売上年別の商品売上ランキングTOP3なら以下のようなクエリがあります。

サブクエリ版	WITH句利用版																																																																																																
<pre>SELECT M.* FROM (SELECT 売上年       ,商品       ,SUM(売上) AS 合計       ,RANK() OVER (                 PARTITION BY 売上                 ORDER BY SUM(売 上) DESC) AS ランキング FROM 明細 GROUP BY 売上年,商品 ) AS M WHERE M.ランキング &lt; 4 ;</pre>	<pre>WITH M AS (   SELECT 売上年         ,商品         ,SUM(売上) AS 合計         ,RANK() OVER (                 PARTITION BY 売上年                 ORDER BY SUM(売上) DESC) AS ランキング FROM 明細 GROUP BY 売上年,商品 ) SELECT * FROM M WHERE M.ランキング &lt; 4 ;</pre>																																																																																																
<table border="1"> <thead> <tr> <th>売上年</th> <th>商品</th> <th>合計</th> <th>ランキング</th> </tr> </thead> <tbody> <tr><td>2004</td><td>みかん</td><td>40</td><td>1</td></tr> <tr><td>2004</td><td>りんご</td><td>37</td><td>2</td></tr> <tr><td>2004</td><td>ばなな</td><td>24</td><td>3</td></tr> <tr><td>2005</td><td>みかん</td><td>60</td><td>1</td></tr> <tr><td>2005</td><td>ばなな</td><td>30</td><td>2</td></tr> <tr><td>2005</td><td>りんご</td><td>29</td><td>3</td></tr> <tr><td>2006</td><td>みかん</td><td>90</td><td>1</td></tr> <tr><td>2006</td><td>まんごー</td><td>88</td><td>2</td></tr> <tr><td>2006</td><td>さくらんぼ</td><td>70</td><td>3</td></tr> <tr><td>2007</td><td>ここあ</td><td>290</td><td>1</td></tr> <tr><td>2007</td><td>だいず</td><td>100</td><td>2</td></tr> </tbody> </table>	売上年	商品	合計	ランキング	2004	みかん	40	1	2004	りんご	37	2	2004	ばなな	24	3	2005	みかん	60	1	2005	ばなな	30	2	2005	りんご	29	3	2006	みかん	90	1	2006	まんごー	88	2	2006	さくらんぼ	70	3	2007	ここあ	290	1	2007	だいず	100	2	<table border="1"> <thead> <tr> <th>売上年</th> <th>商品</th> <th>合計</th> <th>ランキング</th> </tr> </thead> <tbody> <tr><td>2004</td><td>みかん</td><td>40</td><td>1</td></tr> <tr><td>2004</td><td>りんご</td><td>37</td><td>2</td></tr> <tr><td>2004</td><td>ばなな</td><td>24</td><td>3</td></tr> <tr><td>2005</td><td>みかん</td><td>60</td><td>1</td></tr> <tr><td>2005</td><td>ばなな</td><td>30</td><td>2</td></tr> <tr><td>2005</td><td>りんご</td><td>29</td><td>3</td></tr> <tr><td>2006</td><td>みかん</td><td>90</td><td>1</td></tr> <tr><td>2006</td><td>まんごー</td><td>88</td><td>2</td></tr> <tr><td>2006</td><td>さくらんぼ</td><td>70</td><td>3</td></tr> <tr><td>2007</td><td>ここあ</td><td>290</td><td>1</td></tr> <tr><td>2007</td><td>だいず</td><td>100</td><td>2</td></tr> </tbody> </table>	売上年	商品	合計	ランキング	2004	みかん	40	1	2004	りんご	37	2	2004	ばなな	24	3	2005	みかん	60	1	2005	ばなな	30	2	2005	りんご	29	3	2006	みかん	90	1	2006	まんごー	88	2	2006	さくらんぼ	70	3	2007	ここあ	290	1	2007	だいず	100	2
売上年	商品	合計	ランキング																																																																																														
2004	みかん	40	1																																																																																														
2004	りんご	37	2																																																																																														
2004	ばなな	24	3																																																																																														
2005	みかん	60	1																																																																																														
2005	ばなな	30	2																																																																																														
2005	りんご	29	3																																																																																														
2006	みかん	90	1																																																																																														
2006	まんごー	88	2																																																																																														
2006	さくらんぼ	70	3																																																																																														
2007	ここあ	290	1																																																																																														
2007	だいず	100	2																																																																																														
売上年	商品	合計	ランキング																																																																																														
2004	みかん	40	1																																																																																														
2004	りんご	37	2																																																																																														
2004	ばなな	24	3																																																																																														
2005	みかん	60	1																																																																																														
2005	ばなな	30	2																																																																																														
2005	りんご	29	3																																																																																														
2006	みかん	90	1																																																																																														
2006	まんごー	88	2																																																																																														
2006	さくらんぼ	70	3																																																																																														
2007	ここあ	290	1																																																																																														
2007	だいず	100	2																																																																																														

## RANK()じゃ駄目な場合

売上合計が同じ値になる商品だって当然出てきます。そうした場合はRANK()関数を使うと意図した出力にならないかもしれません。

例えば、データに漏れがあったことが判明し、 “すいか”、 “もも” の情報を追加したとしましょう。

```
kouhai=> INSERT INTO 明細 VALUES(2006,'すいか',88);
INSERT 0 1
kouhai=> INSERT INTO 明細 VALUES(2006,'もも',88);
INSERT 0 1
kouhai=> WITH M AS (
kouhai(>   SELECT 売上年
kouhai(>         ,商品
kouhai(>         ,SUM(売上) AS 合計
kouhai(>         ,RANK() OVER ( PARTITION BY 売上年 ORDER BY SUM(売上) DESC) AS
ランキング
kouhai(>   FROM 明細
kouhai(>   GROUP BY 売上年,商品
kouhai(> )
kouhai-> SELECT *
kouhai-> FROM M
kouhai-> WHERE M.ランキング < 4
```

```
kouhai-> ;
売上年 | 商品 | 合計 | ランキング
-----+-----+-----+-----
2004 | みかん | 40 | 1
2004 | りんご | 37 | 2
2004 | ばなな | 24 | 3
2005 | みかん | 60 | 1
2005 | ばなな | 30 | 2
2005 | りんご | 29 | 3
2006 | みかん | 90 | 1
2006 | もも | 88 | 2
2006 | まんごー | 88 | 2
2006 | すいか | 88 | 2
2007 | ここあ | 290 | 1
2007 | だいち | 100 | 2
(12行)

kouhai=>
```

あれ？さくらんぼう.....

売上年が 2006 のものを見えます。ついでなのでDENSE\_RANK()関数ROW\_NUMBER()関数、についても確認してみます。

RANK()関数	DENSE_RANK()関数	ROW_NUMBER()関数
<pre>SELECT 売上年 ,商品 ,SUM(売上) AS 合計 ,RANK() OVER ( PARTITION BY 売上年 ORDER BY SUM(売上) DESC) AS ランキング FROM 明細 WHERE 売上年='2006' GROUP BY 売上年,商品 ;</pre>	<pre>SELECT 売上年 ,商品 ,SUM(売上) AS 合計 ,DENSE_RANK() OVER ( PARTITION BY 売上年 ORDER BY SUM(売上) DESC) AS ランキング FROM 明細 WHERE 売上年='2006' GROUP BY 売上年,商品 ;</pre>	<pre>SELECT 売上年 ,商品 ,SUM(売上) AS 合計 ,ROW_NUMBER() OVER ( PARTITION BY 売上年 ORDER BY SUM(売上) DESC) AS ランキング FROM 明細 WHERE 売上年='2006' GROUP BY 売上年,商品 ;</pre>
<pre>売上年   商品   合計   ランキング -----+-----+-----+----- 2006   みかん   90   1 2006   すいか   88   2 2006   まんごー   88   2 2006   もも   88   2 2006   さくらんぼ   70   5 2006   ぶどう   10   6 2006   めろん   6   7 (7行)  kouhai=&gt;</pre>	<pre>売上年   商品   合計   ランキング -----+-----+-----+----- 2006   みかん   90   1 2006   すいか   88   2 2006   まんごー   88   2 2006   もも   88   2 2006   さくらんぼ   70   3 2006   ぶどう   10   4 2006   めろん   6   5 (7行)  kouhai=&gt;</pre>	<pre>売上年   商品   合計   ランキング -----+-----+-----+----- 2006   みかん   90   1 2006   すいか   88   2 2006   まんごー   88   3 2006   もも   88   4 2006   さくらんぼ   70   5 2006   ぶどう   10   6 2006   めろん   6   7 (7行)  kouhai=&gt;</pre>

すいか、まんごー、もも、の売上合計が同一値のため同順タイになり、

- RANK()関数では同順タイの部分の幅を飛ばしてナンバリングするので、さくらんぼうが5位になってしまいました。
- DENSE\_RANK()関数を使った場合は同順タイがあっても飛ばさないでナンバリングするため、さくらんぼうは3位となります。
- ROW\_NUMBER()関数はレコードの並びにそのまま連番を振るので、さくらんぼうは5位になります。  
また、商品名によるソートが行われないのでROW\_NUMBER()関数の時は、すいか、まんごー、もも、の順位が変わる可能性が出てきます。

こういった結果を望むのかで関数をチョイスする必要があります。

## 追記：ソート順が変わる話

ROW\_NUMBER()の説明で「すいか、まんごー、もも、の順位が変わる可能性」のところがよう分からん、と言われたので。

先のSQL実行結果の並びをもう一度確認します。

```
kouhai=> SELECT
kouhai-> 売上年
kouhai-> ,商品
kouhai-> ,SUM(売上) AS 合計
kouhai-> ,ROW_NUMBER() OVER (
kouhai(>   PARTITION BY 売上年
kouhai(>   ORDER BY SUM(売上) DESC) AS ランキング
kouhai-> FROM 明細
kouhai-> WHERE 売上年='2006'
kouhai-> GROUP BY 売上年,商品
kouhai-> ;
売上年 | 商品 | 合計 | ランキング
-----+-----+-----+-----
2006   | みかん | 90   | 1
2006   | すいか | 88   | 2 ←
2006   | まんごー | 88   | 3 ←
2006   | もも | 88   | 4 ←
2006   | さくらんぼ | 70   | 5
2006   | ぶどう | 10   | 6
2006   | めろん | 6    | 7
(7行)

kouhai=>
```

次に、売上年でのフィルターを外して実行します。

```
kouhai=> SELECT
kouhai-> 売上年
kouhai-> ,商品
kouhai-> ,SUM(売上) AS 合計
kouhai-> ,ROW_NUMBER() OVER (
kouhai(>   PARTITION BY 売上年
kouhai(>   ORDER BY SUM(売上) DESC) AS ランキング
kouhai-> FROM 明細
kouhai-> GROUP BY 売上年,商品
kouhai-> ;
売上年 | 商品 | 合計 | ランキング
-----+-----+-----+-----
2004   | みかん | 40   | 1
2004   | りんご | 37   | 2
2004   | ばなな | 24   | 3
2004   | すもも | 15   | 4
2004   | ぶどう | 5    | 5
2005   | みかん | 60   | 1
```

2005	ばなな	30	2	
2005	りんご	29	3	
2005	すもも	10	4	
2005	ぶどう	3	5	
2006	みかん	90	1	
2006	もも	88	2	←
2006	まんごー	88	3	←
2006	すいか	88	4	←
2006	さくらんぼ	70	5	
2006	ぶどう	10	6	
2006	めろん	6	7	
2007	ここあ	290	1	
2007	だいず	100	2	

(19行)

kouhai=>

表示された 売上年=2006 の並びに差異があることがわかります。

フィルタあり	フィルタなし
2006   すいか   88   2	2006   もも   88   2
←	←
2006   まんごー   88   3	2006   まんごー   88   3
←	←
2006   もも   88   4	2006   すいか   88   4
←	←

これは表示順を明示していないので、実行するSQLによって表示順は変わる、という例です。表示順は変わらないかもしれませんが、暗黙でソートがかかっているデータベース製品もあるかもしれません。

[SQL](#), [PostgreSQL](#), [Oracle](#), [rank](#), [dense rank](#), [database](#), [技術資料](#)

From: <https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link: <https://wiki.hgotoh.jp/documents/database/sql-0008>

Last update: 2023/04/14 02:32

