

SQLで時間の集計を行う

2017年10月24日

ちょっと調べものしてしまったので自分用メモ。

処理対象データ

PostgreSQLに以下のデータがある。

```
$ psql -h dbsvr05 kamonohashi kamo
psql (9.5.9, サーバ 9.6.1)
WARNING: psql major version 9.5, server major version 9.6.
        SOME psql features might NOT WORK.
"help" でヘルプを表示します。

kamonohashi=> \d timelog
   テーブル "public.timelog"
  列 |   型   | 修飾語
-----+-----+-----
 id | INTEGER | NOT NULL
 grp | CHARACTER VARYING(3) |
 TIME | CHARACTER VARYING(5) |
インデックス:
   "timelog_pkey" PRIMARY KEY, btree (id)

kamonohashi=> SELECT * FROM timelog ORDER BY grp,id;
 id | grp | TIME
-----+-----+-----
  1 | ca1 | 08:00
  2 | ca1 | 07:30
  5 | ca1 | 07:15
  3 | ca2 | 07:55
 10 | ca2 | 07:50
 11 | ca2 | 08:10
  4 | ca3 |
  8 | ca3 | 20:70
 21 | ca3 | 27:50
 23 | ca3 | aa:bb
 15 | ca4 | 48:30
 22 | ca4 | 67:50
(12行)

kamonohashi=>
```

これを集計して以下の結果を得たい。

grp	time
ca1	22:45

grp	time
ca2	23:55
ca3	27:50
ca4	99:99

項目timeは

- HH:MMの形式で時間が記録される。
- MMは00-59、HHは0～99の範囲。
- 形式が正しくないものもある。これは計算対象外。

また、

- 集計結果が99:59を超えたら99:99の表記とする。

形式不正の確認

項目timeの形式は SIMILAR TO 演算子で検証する。最近のバージョンのSQL Serverだと、通常の Like 演算子で正規表現が使える模様、OracleだとREGEXP_LIKE()関数で正規表現が扱える。

以下エラー行は('E'の付く行は)集計対象にしない。

```
kamonohashi=> SELECT
kamonohashi->   CASE WHEN "time" SIMILAR TO '[0-9][0-9]:[0-5][0-9]' THEN
NULL ELSE 'E' END AS "err"
kamonohashi->   ,"id"
kamonohashi->   ,"grp"
kamonohashi->   ,"time"
kamonohashi-> FROM
kamonohashi->   timelog
kamonohashi-> ORDER BY grp,id
kamonohashi-> ;
err | id | grp | TIME
-----+-----+-----+-----
      | 1 | ca1 | 08:00
      | 2 | ca1 | 07:30
      | 5 | ca1 | 07:15
      | 3 | ca2 | 07:55
      |10 | ca2 | 07:50
      |11 | ca2 | 08:10
E    | 4 | ca3 |
E    | 8 | ca3 | 20:70
      |21 | ca3 | 27:50
E    |23 | ca3 | aa:bb
      |15 | ca4 | 48:30
      |22 | ca4 | 67:50
(12行)

kamonohashi=>
```

HH:MMから分単位へ変換

エラーが無いHH:MM形式の文字列で表現された時間を分単位に換算する。エラー行の時間はゼロ分にする。

HH部分は“時”なので60を乗じて“分”に換算□MM部分はそのまま。両者を足し合わせれば分単位の換算が完了。

HH × 60 □ MM □ 分単位

```
kamonohashi=> SELECT
kamonohashi->   "id"
kamonohashi->   , "grp"
kamonohashi->   , CASE WHEN "time" SIMILAR TO '[0-9][0-9]:[0-5][0-9]' THEN
CAST(LEFT("time",2) AS INTEGER) * 60 + CAST(RIGHT("time",2) AS INTEGER)
kamonohashi->                                           ELSE 0
kamonohashi->                                           END AS
"time_mins"
kamonohashi-> FROM
kamonohashi->   timelog
kamonohashi-> ;
 id | grp | time_mins
----+----+-----
  1 | ca1 |      480
  2 | ca1 |      450
  5 | ca1 |      435
 10 | ca2 |      470
 11 | ca2 |      490
  3 | ca2 |      475
 21 | ca3 |     1670
  4 | ca3 |         0
 23 | ca3 |         0
 22 | ca4 |     4070
 15 | ca4 |     2910
  8 | ca3 |         0
(12行)

kamonohashi=>
```

上記を項目grpで集計すれば以下となる。

```
kamonohashi=> SELECT
kamonohashi->   "grp"
kamonohashi->   , SUM(CASE WHEN "time" SIMILAR TO '[0-9][0-9]:[0-5][0-9]' THEN
CAST(LEFT("time",2) AS INTEGER) * 60 + CAST(RIGHT("time",2) AS INTEGER)
kamonohashi(>                                           ELSE
0
kamonohashi(>
END ) AS "time_mins"
kamonohashi-> FROM
kamonohashi->   timelog
```

```
kamonohashi-> GROUP BY "grp"
kamonohashi-> ORDER BY "grp"
kamonohashi-> ;
 grp | time_mins
-----+-----
 ca1 |         1365
 ca2 |         1435
 ca3 |         1670
 ca4 |         6980
(4行)

kamonohashi=>
```

集計結果をHH:MM形式に変換

集計後、分単位を再度HH:MM形式に戻す。分単位の集計結果を MINS とすれば、

```
HH = MINS ÷ 60 の商
MM = MINS ÷ 60 の余り
```

となるので、この値を整形して文字列に落とせばいい。

5999は

```
99 × 60 □ 59 □ 5,940 □ 59 □ 5,999
```

でHH:MM形式で表現できる最大の値である5,999分の意。

PostgreSQLやSQL Serverでは整数型を除算した結果は整数型になる□Oracleは型が変換されて実数型になるかもしれない。小数点以下を切り捨てするTRUNC()関数、剰余を求めるMOD()関数、を使う必要がある。

```
kamonohashi=> SELECT
kamonohashi->   mt."grp"
kamonohashi->   ,mt."time_mins"
kamonohashi->   ,CASE WHEN 5999 < mt."time_mins" THEN '99:99'
kamonohashi->                                     ELSE to_char(mt."time_mins"
/ 60, 'FM99')||':'||to_char(mt."time_mins" % 60, 'FM99') END AS "time"
kamonohashi-> FROM
kamonohashi-> (
kamonohashi(>   SELECT
kamonohashi(>     "grp"
kamonohashi(>     ,SUM(CASE WHEN "time" SIMILAR TO '[0-9][0-9]:[0-5][0-9]'
THEN CAST(LEFT("time",2) AS INTEGER) * 60 + CAST(RIGHT("time",2) AS INTEGER)
kamonohashi(> ELSE 0
kamonohashi(> END ) AS "time_mins"
kamonohashi(> FROM
```

```
kamonohashi(>      timelog
kamonohashi(>      GROUP BY "grp"
kamonohashi(> ) mt
kamonohashi-> ORDER BY mt."grp"
kamonohashi-> ;
  grp | time_mins | TIME
-----+-----+-----
  ca1 |      1365 | 22:45
  ca2 |      1435 | 23:55
  ca3 |      1670 | 27:50
  ca4 |      6980 | 99:99
(4行)

kamonohashi=>
```

注意

分単位を INTEGER型で扱っている。集計結果がINTEGER型の最大値を超えるとエラーになるので注意が必要。

INTEGER型最大値が 2,147,483,647 だとすれば、時間の合計が 2,147,483,647分を超える際にエラーとなるだろう。

- $2,147,483,647 \div 60 = 35,791,394.11666667$ 時間
- $35,791,394.11666667 \div 24 = 1,491,308.088194444$ 日
- $1,491,308.088194444 \div 30 = 49,710.26960648148$ ヶ月

扱う時間の性格性質によっては、ここで上げた処理方法は使えない。

追記:SQL Server LocalDBで試してみる

SQL Serverの真っ当なインスタンスを作るの面倒なんでSQLServer LocalDBを使って確認する。

下準備

インスタンスを作ってデータベースを作ってテーブル作ってデータをINSERTします。

```
C:\Users\k896951>sqllocaldb CREATE kamonohashi
LocalDB インスタンス "kamonohashi" はバージョン 13.0.1601.5 で作成されました。

C:\Users\k896951>sqllocaldb START kamonohashi
LocalDB インスタンス "kamonohashi" が開始されました。

C:\Users\k896951>sqllocaldb info kamonohashi
名前:          kamonohashi
バージョン:    13.0.1601.5
共有名:
```

所有者: HOTATE\k896951
自動作成: いいえ
状態: 実行中
前回の開始時刻: 2017/10/24 23:20:46
インスタンス パイプ名: np:\\.\pipe\LOCALDB#\FE181F63\tsql\query

```
C:\Users\k896951>sqlcmd -S np:\\.\pipe\LOCALDB#\FE181F63\tsql\query
```

```
1> SELECT name FROM sys.databases
```

```
2> GO
```

```
name
```

```
-----  
-----  
master
```

```
tempdb
```

```
model
```

```
msdb
```

```
(4行処理されました)
```

```
1> CREATE DATABASE kamo
```

```
2> GO
```

```
1> USE kamo
```

```
2> GO
```

```
Changed DATABASE context TO 'kamo'.
```

```
1> CREATE TABLE timelog(  
2> id INT PRIMARY KEY
```

```
3> ,grp VARCHAR(3)
```

```
4> ,TIME VARCHAR(5)
```

```
5> )
```

```
6> GO
```

```
1> INSERT INTO timelog VALUES( 1,'ca1','08:00')
```

```
2> GO
```

```
(1行処理されました)
```

```
1> INSERT INTO timelog VALUES( 2,'ca1','07:30')
```

```
2> GO
```

```
(1行処理されました)
```

```
1> INSERT INTO timelog VALUES( 5,'ca1','07:15')
```

```
2> GO
```

```
(1行処理されました)
```

```
1> INSERT INTO timelog VALUES(10,'ca2','07:50')
```

```
2> GO
```

```
(1行処理されました)
```

```
1> INSERT INTO timelog VALUES(11,'ca2','08:10')
```

```
2> GO
```

```
(1行処理されました)
```

```
1> INSERT INTO timelog VALUES( 3,'ca2','07:55')
```

```
2> GO
```

(1行処理されました)

```
1> INSERT INTO timelog VALUES(21,'ca3','27:50')
2> GO
```

(1行処理されました)

```
1> INSERT INTO timelog VALUES( 4,'ca3', NULL )
2> GO
```

(1行処理されました)

```
1> INSERT INTO timelog VALUES(23,'ca3','aa:bb')
2> GO
```

(1行処理されました)

```
1> INSERT INTO timelog VALUES(22,'ca4','67:50')
2> GO
```

(1行処理されました)

```
1> INSERT INTO timelog VALUES(15,'ca4','48:30')
2> GO
```

(1行処理されました)

```
1> INSERT INTO timelog VALUES( 8,'ca3','20:70')
2> GO
```

(1行処理されました)

```
1> SELECT * FROM timelog ORDER BY grp,id
2> GO
```

id	grp	TIME
1	ca1	08:00
2	ca1	07:30
5	ca1	07:15
3	ca2	07:55
10	ca2	07:50
11	ca2	08:10
4	ca3	NULL
8	ca3	20:70
21	ca3	27:50
23	ca3	aa:bb
15	ca4	48:30
22	ca4	67:50

(12行処理されました)

```
1>
```

LIKEで正規表現が使えるか確認

LIKE演算子で正規表現が使えました。

```
1> SELECT
2>   CASE WHEN [TIME] LIKE '[0-9][0-9]:[0-5][0-9]' THEN NULL ELSE 'E' END AS
"err"
3>   ,[id]
4>   ,[grp]
5>   ,[TIME]
6> FROM
7>   timelog
8> ORDER BY [grp],[id]
9> GO
err id          grp TIME
-----
NUL          1 ca1 08:00
NUL          2 ca1 07:30
NUL          5 ca1 07:15
NUL          3 ca2 07:55
NUL         10 ca2 07:50
NUL         11 ca2 08:10
E            4 ca3 NULL
E            8 ca3 20:70
NUL         21 ca3 27:50
E           23 ca3 aa:bb
NUL         15 ca4 48:30
NUL         22 ca4 67:50

( 12 行処理されました )
1>
```

集計してみる

集計結果を “ HH:MM ” の文字列に直した後明示的に VARCHAR(5) へ CAST しているのは、表示が横長になるのを防ぐため。なくても構わない。

```
1> SELECT
2>   mt.[grp]
3>   ,mt.[time_mins]
4>   ,CAST(CASE WHEN 5999 < mt.[time_mins]
5>         THEN '99:99'
6>         ELSE format(mt.[time_mins] / 60, 'D2')+' ':'+format(mt.[time_mins]
% 60, 'D2') END AS VARCHAR(5)) AS "time"
7> FROM
8> (
9>   SELECT
10>     [grp]
11>     ,SUM(CASE WHEN [TIME] LIKE '[0-9][0-9]:[0-5][0-9]' THEN
CAST(LEFT([TIME],2) AS INTEGER) * 60 + CAST(RIGHT([TIME],2) AS INTEGER)
12>                                           ELSE 0
13>                                           END ) AS
"time_mins"
14>   FROM
```

```
15>      timelog
16>      GROUP BY [grp]
17> ) mt
18> ORDER BY mt.[grp]
19> GO
grp time_mins    TIME
-----
ca1      1365 22:45
ca2      1435 23:55
ca3      1670 27:50
ca4      6980 99:99

( 4 行処理されました )
1>
```

[database](#), [SQL](#), [時間の集計](#), [PostgreSQL](#), [SQLServer](#), [Oracle](#), [技術資料](#)

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

Permanent link:
<https://wiki.hgotoh.jp/documents/database/sql-0009>

Last update: **2023/04/14 02:32**

