

# 指定時間幅でまとめる

2024/03/19 おまけ追加  
2024/03/18 ヒントだけ

## サンプルデータ

こういうデータがあったとします。

クエリ クエリの履歴

```

1 SELECT * FROM public.log
2

```

データ出力 メッセージ 通知

	time timestamp without time zone	devid character varying (10)	actionid character varying (10)
1	2024-03-18 19:56:36.22647	PC-A	TRAN-A
2	2024-03-18 19:56:36.826	PC-A	TRAN-B
3	2024-03-18 19:59:12.814435	PC-B	TRAN-A
4	2024-03-18 19:59:13.00125	PC-B	TRAN-B
5	2024-03-18 19:59:14	PC-B	TRAN-C
6	2024-03-18 20:00:36.393073	PC-A	TRAN-B
7	2024-03-18 20:01:39.613909	PC-C	TRAN-X
8	2024-03-18 20:01:39.62	PC-C	TRAN-Y
9	2024-03-18 20:01:39.623	PC-C	TRAN-Z
10	2024-03-18 20:00:36.393073	PC-A	TRAN-C
11	2024-03-18 20:00:36.393073	PC-A	TRAN-C

## 連続した2秒以内の塊を得る

項目DEVID毎に、連続した2秒以内のレコード群を一つのグループに纏めたいという要求がありました。以下のクエリでとりあえずグループの識別はできました。

項目MARKが'T'のレコードがグループの先頭レコードになるんじゃないかな、と。

クエリ クエリの履歴

```

1 WITH V AS (
2   SELECT *,LAG(TIME,1) OVER(PARTITION BY DEVID ORDER BY TIME) as TIME1PRE
3   FROM LOG
4 ),
5 VV AS (
6   SELECT TIME
7     , DEVID
8     , ACTIONID
9     , TIME - TIME1PRE AS CALC
10    , CASE WHEN '00:00:02' >= (TIME - TIME1PRE) THEN 'Y' ELSE 'T' END AS MARK
11   FROM V
12 )
13 SELECT TIME, DEVID, ACTIONID, CALC, MARK
14 FROM VV
15 ORDER BY TIME
16 ;
17

```

データ出力 メッセージ 通知



	time timestamp without time zone	devid character varying (10)	actionid character varying (10)	calc interval	mark text
1	2024-03-18 19:56:36.22647	PC-A	TRAN-A	[null]	T
2	2024-03-18 19:56:36.826	PC-A	TRAN-B	00:00:00.59953	Y
3	2024-03-18 19:56:37	PC-A	TRAN-C	00:00:00.174	Y
4	2024-03-18 19:59:12.814435	PC-B	TRAN-A	[null]	T
5	2024-03-18 19:59:13.00125	PC-B	TRAN-B	00:00:00.186815	Y
6	2024-03-18 19:59:14	PC-B	TRAN-C	00:00:00.99875	Y
7	2024-03-18 20:00:36.393073	PC-A	TRAN-B	00:03:59.393073	T
8	2024-03-18 20:00:36.393075	PC-A	TRAN-C	00:00:00.000002	Y
9	2024-03-18 20:01:39.613909	PC-C	TRAN-X	[null]	T
10	2024-03-18 20:01:39.62	PC-C	TRAN-Y	00:00:00.006091	Y
11	2024-03-18 20:01:39.623	PC-C	TRAN-Z	00:00:00.003	Y

## グルーピングはできない

グループ化するキーが無いので、グルーピングはできません。精々グループの先頭レコードの識別ぐらいしかできません。

疑似的なグループ化であれば、項目MARKが'T'のレコードだけ抽出すればいい。

クエリ クエリの履歴

```

1 WITH V AS (
2   SELECT *, LAG(TIME,1) OVER(PARTITION BY DEVID ORDER BY TIME) as TIME1PRE
3   FROM LOG
4 ),
5 VV AS (
6   SELECT TIME
7     , DEVID
8     , ACTIONID
9     , TIME - TIME1PRE AS CALC
10    , CASE WHEN '00:00:02' >= (TIME - TIME1PRE) THEN 'Y' ELSE 'T' END AS MARK
11   FROM V
12 )
13 SELECT TIME, DEVID, ACTIONID, CALC, MARK
14 FROM VV
15 WHERE MARK='T'
16 ORDER BY TIME
17 ;
18

```

データ出力 メッセージ 通知

	time timestamp without time zone	devid character varying (10)	actionid character varying (10)	calc interval	mark text
1	2024-03-18 19:56:36.22647	PC-A	TRAN-A	[null]	T
2	2024-03-18 19:59:12.814435	PC-B	TRAN-A	[null]	T
3	2024-03-18 20:00:36.393073	PC-A	TRAN-B	00:03:59.393073	T
4	2024-03-18 20:01:39.613909	PC-C	TRAN-X	[null]	T

また、このクエリは2秒を越えて連続しているレコードの状態も切り分けが出来ません。 連続した2つ以上の塊の識別はできない  
 厳密に求めるなら先頭レコードのタイムスタンプからの差分を取る必要があるでしょう。

## おまけ (塊先頭のタイムスタンプをキーにする)

結果が正しいかは検証していませんが思いついたことがあったので試してみました。  
 表面上の結果はうまくいっているように見えました。

クエリ クエリの履歴

```

1 WITH V AS (
2   SELECT *, LAG(TIME,1) OVER(PARTITION BY DEVID ORDER BY TIME) as TIME1PRE FROM LOG
3 ),
4 V1 AS (
5   SELECT TIME, DEVID, ACTIONID, CASE WHEN '00:00:02' >= (TIME - TIME1PRE) THEN NULL ELSE TIME END AS MARK FROM V
6 ),
7 V2 AS (
8   SELECT TIME, DEVID, MARK FROM V1 WHERE MARK IS NOT NULL
9 ),
10 VV AS (
11   SELECT V1.TIME, V1.DEVID, V1.ACTIONID, V1.MARK, V2.MARK as GRPKEY
12   FROM V1 LEFT OUTER JOIN V2 ON(V1.DEVID=V2.DEVID AND V1.TIME BETWEEN V2.MARK AND V2.MARK + '00:00:02')
13 )
14 SELECT TIME, DEVID, ACTIONID, MARK, GRPKEY
15 FROM VV
16 ORDER BY TIME
17 ;
18

```

データ出力 メッセージ 通知



	time timestamp without time zone	devid character varying (10)	actionid character varying (10)	mark timestamp without time zone	grpkey timestamp without time zone
1	2024-03-18 19:56:36.22647	PC-A	TRAN-A	2024-03-18 19:56:36.22647	2024-03-18 19:56:36.22647
2	2024-03-18 19:56:36.826	PC-A	TRAN-B	[null]	2024-03-18 19:56:36.22647
3	2024-03-18 19:56:37	PC-A	TRAN-C	[null]	2024-03-18 19:56:36.22647
4	2024-03-18 19:59:12.814435	PC-B	TRAN-A	2024-03-18 19:59:12.814435	2024-03-18 19:59:12.814435
5	2024-03-18 19:59:13.00125	PC-B	TRAN-B	[null]	2024-03-18 19:59:12.814435
6	2024-03-18 19:59:14	PC-B	TRAN-C	[null]	2024-03-18 19:59:12.814435
7	2024-03-18 20:00:36.393073	PC-A	TRAN-B	2024-03-18 20:00:36.393073	2024-03-18 20:00:36.393073
8	2024-03-18 20:00:36.393075	PC-A	TRAN-C	[null]	2024-03-18 20:00:36.393073
9	2024-03-18 20:01:39.613909	PC-C	TRAN-X	2024-03-18 20:01:39.613909	2024-03-18 20:01:39.613909
10	2024-03-18 20:01:39.62	PC-C	TRAN-Y	[null]	2024-03-18 20:01:39.613909
11	2024-03-18 20:01:39.623	PC-C	TRAN-Z	[null]	2024-03-18 20:01:39.613909

技術資料, SQL, 時間幅での集約

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

Permanent link: <https://wiki.hgotoh.jp/documents/database/sql-0014>

Last update: 2024/03/19 02:15

