

开飞机的老张

写给未来的信

首页 归档 标签 关于

PostgreSQL的行转列和列转行

· 2019-07-03 · # 数据库





Oracle中的行转列和列转行分别有pivot和unpivot方法。

在PostgreSQL中,行列互转的方法也有很多,在这里介绍常用的两种。

crosstab行转列

有某平均温度数据(data_avg_temp):

name	month	avg_temp	
康山	jan	5	
康山	apr	16.3	
康山	july	28.8	
康山	oct	19.2	
棠荫	jan	6	

name	month	avg_temp
棠荫	apr	17.6
棠荫	july	29.7
棠荫	oct	20.3

要转换成如下格式:

name	jan	apr	july	oct
康山	5	16.3	28.8	19.2
棠荫	6	17.6	29.7	20.3

使用crosstab方法实现行转列:

1. 使用crosstab方法,需要安装扩展模块tablefunc:

CREATE EXTENSION tablefunc; -- 第一次使用crosstab前执行,后续无需再执行

2. 实现代码:

```
SELECT * FROM crosstab
(
    'SELECT name, month, avg_temp FROM data_avg_temp ORDER BY 1,2',
    $$values('jan'::text),('apr'::text),('july'::text),('oct'::text)$$
)
AS data_avg_temp_cross
(name text, jan numeric, apr numeric, july numeric, oct numeric);
```

或:

```
SELECT * FROM crosstab
(
    'SELECT name, month, avg_temp FROM data_avg_temp ORDER BY 1,2',
    'SELECT DISTINCT month FROM data_avg_temp ORDER BY 1'
)
AS data_avg_temp_cross
(name text, jan numeric, apr numeric, july numeric, oct numeric);
```

3. 转换原理

crosstab内的第一个参数,是行转列的源表数据:

```
SELECT name, month, avg_temp FROM data_avg_temp ORDER BY 1,2',
```

crosstab内的第二个参数,是行转列的那一列数据,有两种形式:

• values形式:

```
$$values('jan'::text), ('apr'::text), ('july'::text), ('oct'::text)$$
```

• DISTINCT形式:

'SELECT DISTINCT month FROM data_avg_temp ORDER BY 1'

最后要定义转换后的表结构:

AS data_avg_temp_cross (name text, jan numeric, apr numeric, july numeric, oct numeric)

jsonb列转行

有某平均温度数据 (data_avg_temp2) :

name	jan	apr	july	oct
康山	5	16.3	28.8	19.2
棠荫	6	17.6	29.7	20.3

要转换成如下格式:

name	month	avg_temp
康山	jan	5
康山	apr	16.3
康山	july	28.8
康山	oct	19.2
棠荫	jan	6
棠荫	apr	17.6
棠荫	july	29.7

name	month	avg_temp
棠荫	oct	20.3

使用jsonb方法实现列转行:

1. 实现代码:

```
SELECT name, (b.rec).key as month, (b.rec).value FROM
(
    SELECT name, jsonb_each(row_to_json(a)::jsonb-'name'::varchar) as rec FF
    (SELECT name, jan, apr, july, oct FROM data_avg_temp2) a
) b
```

2. 转换原理

关键点在于:

```
jsonb_each(row_to_json(a)::jsonb-'name'::varchar)
```

row_to_json(a)::jsonb把行数据转为JSON,例如:

name	jan	apr	july	oct
康山	5	16.3	28.8	19.2

转为:

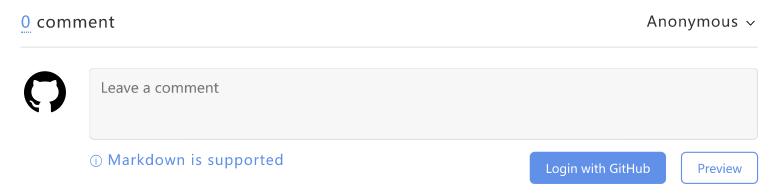
```
{"name":"康山","jan":"5","apr":"16.3","july":"28.8","oct":"19.2"}
```

-'name'::varchar把JSON中的name键值去掉:

jsonb_each再转换为JSON对象rec,最后在外层SELECT中用(b.rec).key和(b.rec).value分别取出月份和平均温度。

下一篇

Oracle和PostgreSQL的自动备份及定期删除脚本



Be the first person to leave a comment!



Email | RSS