PostgreSQL的递归查询

作者：霍琦

目录

[1. 场景描述 2](#_Toc426035383)

[1.1. 场景 2](#_Toc426035384)

[1.2. 准备表和数据 2](#_Toc426035385)

[2. 查询 2](#_Toc426035386)

[2.1. 向上递归追溯 2](#_Toc426035387)

[2.2. 向下递归追溯 2](#_Toc426035388)

[2.3. 总结 2](#_Toc426035389)

# 场景描述

## 场景

在计算模型权限逻辑中，如果计算某个模型的权限，需要当前模型和它的向上追溯到根模型的所有父模型的权限逻辑之和。实现办法可以是取出当前模型，再取出它的父模型，根据父模型再取出父模型，直到最上层的根模型。但这样做的问题就是，由于只能一层一层地向上递归，导致数据库查询次数会很多，并且不确定查询次数的上限。改进的方法是，通过一条SQL语句把某个模型的所有父模型一次全部取出，在内存中计算该模型的权限定义，这样做的好处是减少了数据库的查询交互次数。

## 准备表和数据

涉及到的模型有三种：项目、分组、工作流，他们的关系是一个项目可以包含多个分组，多个工作流，一个分组下也可以包含多个分组，多个工作流。由于分组这个物理表中就包含了父子关系（分组包含子分组），所以只使用分组这个物理表来描述。

表结构主要包含三列：pk\_wfgroup，groupname，pk\_parent，其中pk\_parent指的是父分组PK。



# 查询

## 向上递归追溯

WITH RECURSIVE rec AS

(

 SELECT

 g.pk\_wfgroup,

 g.groupname,

 g.pk\_parent,

 g.pk\_project

 FROM

 ae\_dm\_wfgroup g

 WHERE

 pk\_wfgroup='group121'

 UNION ALL

 SELECT

 gr.pk\_wfgroup,

 gr.groupname,

 gr.pk\_parent,

 gr.pk\_project

 FROM

 ae\_dm\_wfgroup gr

 INNER JOIN

 rec rec

 ON

 rec.pk\_parent = gr.pk\_wfgroup )

SELECT

 pk\_wfgroup,

 groupname,

 pk\_parent

FROM

 rec



## 向下递归追溯

WITH RECURSIVE rec AS

(

 SELECT

 g.pk\_wfgroup,

 g.groupname,

 g.pk\_parent,

 g.pk\_project

 FROM

 ae\_dm\_wfgroup g

 WHERE

 pk\_wfgroup='group1'

 UNION ALL

 SELECT

 gr.pk\_wfgroup,

 gr.groupname,

 gr.pk\_parent,

 gr.pk\_project

 FROM

 ae\_dm\_wfgroup gr

 INNER JOIN

 rec rec

 ON

 rec.pk\_wfgroup = gr.pk\_parent )

SELECT

 pk\_wfgroup,

 groupname,

 pk\_parent

FROM rec



## 总结

WITH RECURSIVE里包括两块，non-recursive-term和recursive-term，递归项和非递归项通过UNION或UNION ALL方式合并。

1. 通过向下追溯这个过程可以看到，先查询出记录group1，SELECT g.pk\_wfgroup, g.groupname, g.pk\_parent, g.pk\_project FROM ae\_dm\_wfgroup g WHERE pk\_wfgroup='group1'，这条记录做为非递归项，放到返回结果集中。

2. 接下来的任务是只要关联条件可以查询出数据，即和返回结果集做关联查询，就不断地重复：（1）根据递归语句查询出结果集，放到临时的中间表。（2）将中间表的内容加入到返回结果集中。

最终形成一个自上而下或自下而上的路径结果集。