



链滴

MYSQL 递归树查询的实现

作者: [boskt](#)

原文链接: <https://ld246.com/article/1537930248317>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在oracle中我们

可以使用connect by prior 函数来实现组织树表递归的查询，但是在mysql中却没有该方法，为了能够用该方法我们就得自己创建函数来实现该组织树的递归查询。以下实现了2种方式来满足子节点的递归查询与父节点的递归查询。

实现子递归查询

1.单个子递归查询的实现

oracle中使用示例代码：

```
SELECT
  T.ID
FROM
  bas_basic_tree T CONNECT BY PRIOR T.ID = T.PID
START WITH T.ID = #{id}
```

mysql中使用示例代码：

```
SELECT
  ID
FROM
  bas_basic_tree T,
  (SELECT @DATAS := getChildId_bas_basic_tree (#{id})) x
WHERE find_in_set (ID, @DATAS)
```

mysql方法的实现：

-- 在navcat中的查询下执行以下命令

```
drop function if exists getChildId_bas_basic_tree;
```

```
DELIMITER //
create function getChildId_bas_basic_tree(rootId varchar(20))
returns varchar (1000)
BEGIN
  DECLARE pTemp VARCHAR(1000);
  DECLARE cTemp VARCHAR(1000);    -- 节点ID(临时变量)

  SET pTemp = '$';
  SET cTemp =cast(rootId as CHAR); -- 把rootId强制转换为字符。

  WHILE cTemp is not null DO
    SET pTemp = concat(pTemp,',',cTemp); -- 把所有节点连接成字符串。
    SELECT group_concat(id) INTO cTemp FROM bas_basic_tree
    WHERE FIND_IN_SET(pid,cTemp)>0;
  END WHILE;
  RETURN pTemp;
END //
DELIMITER ;
```

2.多个子递归查询的实现

oracle中使用示例代码：

```

SELECT
  T.ID
FROM
  bas_basic_tree T CONNECT BY PRIOR T.ID = T.PID
  START WITH T.ID in( #{ids} )

```

-- 其中ids比如是 ('1', '2') 这种

mysql中使用示例代码:

```

SELECT
  *
FROM
  bas_basic_tree T ,
  (select @DATAK :=
    (
      SELECT
        -- 使用该方法把所有的ID用逗号分割连接成一个字符串
        group_concat(T.ID)
      FROM
        bas_basic_tree T
      WHERE
        NAME LIKE #{orgname}
      -- 括号里面查询出来的结果类似与 '1,2' 这种
    )
  ) z,
  (SELECT @DATAS := getParentId_bas_basic_tree_more (@DATAK) ) x
WHERE find_in_set (T.ID, @DATAS)

```

-- 注意这里select @DATAK 与 SELECT @DATAS 顺序不能乱, 否则会出现数据错误

mysql方法的实现:

```
drop function if exists getChildId_bas_basic_tree_more;
```

```

DELIMITER //
create function getChildId_bas_basic_tree_more(nodes varchar(1000))
returns varchar (5000)

```

```

BEGIN
  DECLARE pTemp VARCHAR(1000);
  DECLARE cTemp VARCHAR(5000);

  SET @array_content= nodes;
  SET @i=1;
  SET @count=CHAR_LENGTH(@array_content) -CHAR_LENGTH(REPLACE(@array_content,'
','')) + 1;
  SET pTemp = '$';

  WHILE @i <= @count DO
    SET cTemp = SUBSTRING_INDEX(SUBSTRING_INDEX(@array_content,',',@i),',',-1);
    WHILE cTemp is not null DO
      SET pTemp = concat(pTemp,',',cTemp);
      SELECT group_concat(id) INTO cTemp FROM bas_basic_tree

```

```

        WHERE FIND_IN_SET(pid,cTemp) > 0;
    END WHILE;
    SET @i = @i + 1;
END WHILE;

RETURN pTemp;

```

```

END //
DELIMITER ;

```

实现父递归查询

1.单个父递归查询的实现

oracle中使用示例代码:

```

SELECT
  T.ID
FROM
  bas_basic_tree T CONNECT BY PRIOR T.PID = T.ID
START WITH T.ID = #{id}

```

mysql中使用示例代码:

```

SELECT
  ID
FROM
  bas_basic_tree T,
  (SELECT @LIST := getParentId_bas_basic_tree (#{id})) x
WHERE find_in_set (ID, @LIST)

```

mysql方法的实现:

-- 在navcat中的查询下执行以下命令

```
drop function if exists getParentId_bas_basic_tree;
```

```

DELIMITER //
create function getParentId_bas_basic_tree(rootId varchar(20))
returns varchar (1000)
BEGIN
  DECLARE fid varchar(100) default "";
  DECLARE str varchar(1000) default rootId; -- 节点ID(临时变量)

  WHILE rootId is not null do
    SET fid =(SELECT pid FROM bas_basic_tree WHERE id = rootId);
    IF fid is not null THEN
      SET str = concat(str, ',', fid);
      SET rootId = fid;
    ELSE
      SET rootId = fid;
    END IF;
  END WHILE;
  return str;

```

```
END //
DELIMITER ;
```

2.多个父递归查询的实现

oracle中使用示例代码:

```
SELECT
  T.ID
FROM
  bas_basic_tree T CONNECT BY PRIOR T.PID = T.ID
  START WITH T.ID in( #{ids} )
```

-- 其中ids比如是 ('1', '2') 这种

mysql中使用示例代码:

```
SELECT
  T.*
FROM
  bas_basic_tree T ,
  (select @DATAK :=
    (
      SELECT
        -- 使用该方法把所有的ID用逗号分割连接成一个字符串
        group_concat(T.ID)
      FROM
        bas_basic_tree T
      WHERE
        NAME LIKE #{orgname}
      -- 括号里面查询出来的结果类似与 '1,2' 这种
    )
  ) zz,
  (SELECT @DATAS := getParentId_bas_basic_tree_more (@DATAK) ) x
WHERE find_in_set (T.ID, @DATAS)
```

mysql方法的实现:

-- 在navcat中的查询下执行以下命令

```
drop function if exists getParentId_bas_basic_tree_more;
```

```
DELIMITER //
create function getParentId_bas_basic_tree_more(nodes varchar(1000))
returns varchar (5000)

BEGIN
  DECLARE pTemp VARCHAR(1000);
  DECLARE cTemp VARCHAR(5000);
  DECLARE xTemp varchar(100);

  SET @array_content= nodes;
  SET @i=1;
  SET @count=CHAR_LENGTH(@array_content) -CHAR_LENGTH(REPLACE(@array_content,'','')) + 1;
```

```
SET pTemp = nodes;

WHILE @i <= @count DO
  SET cTemp=SUBSTRING_INDEX(SUBSTRING_INDEX(@array_content,',',@i),',',-1);
  WHILE cTemp is not null DO
    SET xTemp = (SELECT pid FROM bas_basic_tree WHERE id = cTemp);
    IF xTemp is not null THEN
      SET pTemp = concat(pTemp, ',', xTemp);
      set cTemp = xTemp;
    ELSE
      set cTemp = xTemp;
    END IF;
  END WHILE;
  SET @i = @i + 1;
END WHILE;
RETURN pTemp;
END //

DELIMITER ;
```