Oracle 层次查询功能的剖析

The Anatomy of Functions of Hierarchical Queries Based on Oracle

刘建波（河北医科大学流行病学与统计学教研室 050017）
王荣杰 李献军 吕铁山（国家邮政局石家庄培训中心 050021）

摘要：软件项目中存在着大量的组织结构业务，Oracle 数据库中的层次查询功能给程序员带来了极大方便，本文对其使用方法和灵活性进行了深入探讨。

关键词：Oracle 树形结构 层次查询

1 引言

在软件项目中有许多"层次结构"方面的业务，它们往往使用树形控件表达。Oracle 数据库系统在标准 SQL 语言基础上进行了扩展，引入了层次查询(Hierarchical Queries)，该功能给程序员带来了极大方便，本文对此进行深入探讨。

2 基本知识

2.1 层次结构

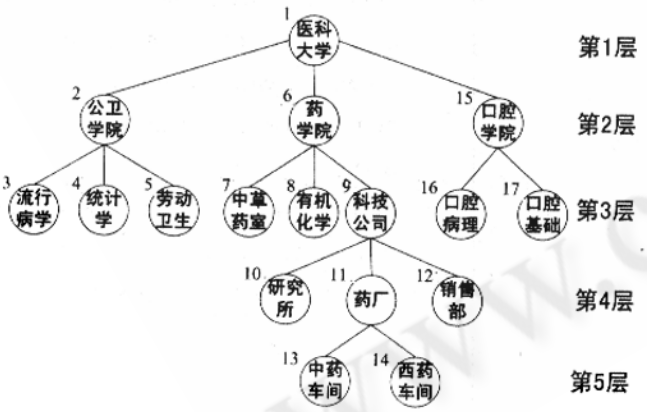


图 1 医科大学组织结构

层次结构与树形结构相同，也称为分级结构或等级结构，它是由具备等级属性的数据组织在一起所形成的表现形式。如计算机的磁盘文件和目录管理是层次结构。图 1 是一个虚拟的医科大学树形结构(节点左上角括弧内数字稍后详述)。医科大学是根节点，没有父节点，它有三个子节点(公卫学院、药学院、口腔学院)。中草药室、有机化学、科技公司是药学院的子节点，是医科大学的孙节点。流行病学、中草药室、中药车间等，虽然位居的层次高低不同，但他们均是叶节点，没有子孙。

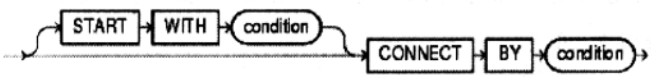
在 Oracle 中建立一张数据表 Hierarchical_Data 用于存储图 1 中的数据，数据表的内容如表 1 所示。

表 1 层次数据原始存储次序

ID	NODE	P_ID	P_NODE
A	医科大学		
B	公卫学院	A	医科大学
C	药学院	A	医科大学
D	口腔学院	A	医科大学
E	流行病学	B	公卫学院
F	统计学	B	公卫学院
G	劳动卫生	B	公卫学院
H	中草药室	C	药学院
I	有机化学	C	药学院
J	科技公司	C	药学院
J	科技公司	C	药学院
K	口腔病理	D	口腔学院
L	口腔基础	D	口腔学院
M	研究所	J	科技公司
N	药厂	J	科技公司
O	销售部	J	科技公司
P	中药车间	N	药厂
Q	西药车间	N	药厂

注：节点代码 ID，名称 NODE

2.2 Oracle 层次查询语法



START WITH：指定层次数据的"根"所在行的条件。

CONNECT BY：指定层次数据中父节点所在行与子节点所在行的关系。

CONDITION 语句：这是一般性查询的 WHERE 语句，并

非专门为层次查询设立。

2.3 查询步骤

Oracle 层次查询提取数据过程有 5 个步骤：

① 从数据表中提取满足 START WITH 条件的数据行作为根节点数据，可能是一行，也可能是多行；

② 根据 CONNECT BY 指定的条件选择每一个根数据行的子行，也就是选择子节点数据行；

③ 将第二步中选择的子节点数据行作为根节点数据行，选择出子节点数据行。然后再将这些子节点数据行作为根节点数据行并选择其子行，依此类推；

④ 如果查询语句中有 WHERE 语句，将从结果集的层次数据行中剔除不满足选择条件的数据行。Oracle 对层次数据中的每一行进行检查以确定是否剔除该行，而不会剔除不满足 WHERE 条件的行所衍生的子行数据；

⑤ Oracle 层次查询的结果将以特定的次序返回数据行，以保证显示的次序不乱。公卫学院和药学院虽是兄弟节点关系，但是在显示的时候首先将公卫学院及其所有子孙节点显示完毕之后才显示药学院及其子孙。图 1 每个节点左上角括弧内数字表达的是返回数据行的次序号码。

表 2　层次查询结果显示次序

ID	NODE	P_ID	P_NODE
A	医科大学		
B	公卫学院	A	医科大学
E	流行病学	B	公卫学院
F	统计学	B	公卫学院
G	劳动卫生	B	公卫学院
C	药学院	A	医科大学
H	中草药室	C	药学院
I	有机化学	C	药学院
J	科技公司	C	药学院
M	研究所	J	科技公司
N	药厂	J	科技公司
P	中药车间	N	药厂
Q	西药车间	N	药厂
O	销售部	J	科技公司
D	口腔学院	A	医科大学
K	口腔病理	D	口腔学院
L	口腔基础	D	口腔学院

父节点代码 P_ID，名称 P_NODE

2.4 查询限制

① 不能存在数据表之间的 JOIN 连接，也不能从有 JOIN 连接的视图中获取数据；

② 如果查询语句中有 ORDER BY 语句，返回结果的次序将遵循 ORDER BY 从句，查询步骤中第⑤步的作用被忽略；

③ CONNECT BY 不能包含子查询；

④ CONNECT BY 中的 CONDITION 从句必须包含"... PRIOR EXPR = EXPR"或者"... EXPR = PRIOR EXPR"来表达父子节点关系的数据行。

2.5 伪列

伪列不是数据表中真正意义上的字段，从语法上讲，它的处理方式与数据表中的列很相似，并在查询集中也有返回值。伪列给程序员提供了更多工具，可在编程中更加灵活运用。

3 层次查询的使用和注意事项

3.1 选择所有子孙节点

```
select * from Hierarchical_Data Start with ID = 'A' connect by prior ID=P_ID
```

该条查询语句从图 1 选择医科大学的所有子孙节点记录，返回的结果如表 2 所示。通过表 1 和表 2 比较，我们可以体会到查询步骤中第 5 步叙述的层次数据显示次序。如果存在 ORDER BY 语句，则遵从查询限制中的第 2 点；如果没有 START WITH 语句，Oracle 将把表 1 中的所有行视为根节点。START WITH 语句中可以包含子查询。

如果 CONNECT BY 语句返回的结果中有一个循环，Oracle 将弹出错误信息。循环的发生是因为某一行数据是另一特定行的祖先行，又是他的子孙行，也就是说，层次数据的树形结构被破坏而形成了环型结构。

3.2 选择任意节点之下的子孙

药学院是医科大学的一个子节点，下面的 SQL 语句选择了药学院及其下属单位：

```
select * from Hierarchical_Data Start with ID = 'C' connect by P_ID = prior ID
```

该语句返回的结果是表 2 中的药学院开始的一个子集，共有 9 行记录。这个任意节点由 START WITH 语句确定，并成为树形结构的根节点。

这里需要注意与 3.1 中所述的 Select 语句进行比较。CONNECT BY 语句中的 PRIOR 关键字位置，它可以在等号左侧，也可以在右侧，关键问题是 PRIOR 后面是什么字段。

3.3 WHERE 语句的使用

选择药学院及其所有子孙节点，并要求所有数据行符合节点代码在 D 和 P 之间。

```
select * from Hierarchical_Data where (ID between 'D' and 'P') Start with ID = 'C' connect by P_ID = prior ID
```

该语句查询结构见表 3。

表 3

ID	NODE	P_ID	P_NODE
H	中草药室	C	药学院
I	有机化学	C	药学院
J	科技公司	C	药学院
M	研究所	J	科技公司
N	药厂	J	科技公司
P	中药车间	N	药厂
O	销售部	J	科技公司

表 4

ID	NODE	P_ID	P_NODE
C	药学院	A	医科大学
H	中草药室	C	药学院
I	有机化学	C	药学院
J	科技公司	C	药学院
M	研究所	J	科技公司
N	药厂	J	科技公司
P	中药车间	N	药厂
O	销售部	J	科技公司

表 5

L	HIERARCHY	3	研究所
1	药学院	3	药厂
2	中草药室	4	中药车间
2	有机化学	4	西药车间
2	科技公司	3	销售部

如果将 WHERE 语句放到 CONNECT BY 中，则变成了下面的 SQL 查询：

select * from Hierarchical_Data Start with ID = 'C' connect by P_ID = prior ID and (ID between 'D' and 'P')，查询结果见表 4。这条语句只针对子孙节点进行过滤。请读者多写几条类似的语句进行深入领会。

3.4 LEVEL 伪列的使用

Oracle 使用 LEVEL 伪列来显示层次数据的层次号码，LEVEL 对于选定的根节点返回 1，根节点的子节点返回 2，依次类推，如图 1 中右侧表达层次的数字。LEVEL 在层次查询中会受到内存影响。

select Level L, lpad('',2 * (level - 1))||Node Hierarchy from Hierarchical_Data Start with ID in ('C') connect by P_ID = prior ID，该语句返回药学院所有子孙节点(表 5)。结合图 1 和表 5 会发现，查询语句利用 LEVEL 伪列将层次数据之间的层次关系使用缩进形式清晰地表达。程序员利用缩进非常方便地将层次数据赋给树形控件快速展现在用户界面。该查询使用了 START WITH ID IN ('C')，如果写成 START WITH ID IN ('C', 'B')，查询结果中将包括公卫学院及其子孙。

如果在 Connect by 语句后面添加 Level <= 3，查询语句将返回 3 层以内(包括根节点)的子孙节点，表 5 中的中、西药车间将不存在。如果我们要查找药学院的孙节点(包括药学院在内的第三层数据)，WHERE 语句必须在 START WITH 语句之前，不能在 CONNECT BY 之后添加组合限制。例如：select lpad('',2 * (level - 1))||Node Hierarchy from Hierarchical_Data where Level = 3 Start with ID in ('C') connect by P_ID = prior ID，查询结果只返回研究所、药厂、销售部三个孙代节点，如果将 WHERE 语句去掉，在 CONNECT BY 后面添加 LEVEL = 3 AND，其查询结果不能达到目的。

3.5 PRIOR 后面的字段与寻找方向

以上所有的查询语句均将节点代码 ID 放在了 PRIOR 后面，它是一个从 Start with 语句指定的节点开始寻找子孙节点的过程。如果将 PRIOR 放在父节点代码字段 P_ID 之前，它的含义是从指定节点开始寻找父节点的过程。读者可以自行调整 PRIOR 的位置，深入体会 Oracle 的查询过程，并注意层次关系在表现形式方面有无变化。下面的语句返回中药车间的第三代祖先(包括自身节点在内的第 4 层数据)：select lpad('',2 * (level - 1))||Node Hierarchy from Hierarchical_Data where Level = 4 Start with ID in ('P') connect by prior P_ID = ID，返回结果是药学院。

4 结束语

本文以虚拟的医科大学组织结构探讨了 Oracle 的层次查询功能。在软件项目中，涉及层次数据的业务有许多方面。如大学的班级维护，层次繁多，变化频繁，可以充分利用 Oracle 的层次查询功能进行灵活地配置、方便地上溯和下溯查询；财务方面，即可以将某一费用归结到某一个大类，也可以根据某一科目查找任意子孙级别的科目组成。利用 LEVEL、LPAD 产生查询结果缩进功能，可以利用树形控件方便、快速地展现到用户界面。

总之，领会和掌握 Oracle 的层次查询功能，会极大提高程序员的日常工作效率。

参考文献

ORACLE. Oracle9i SQL Reference, Release 2 (9.2). October 2002.