

A Guide to SQL Usage

**Oracle, DB2,
SQL Server & MySQL**

SQL

Pocket Guide



O'REILLY®

Jonathan Gennick

Hierarchical Queries

ANSI/ISO and Oracle provide different syntax for querying data in a recursive, parent-child relationship. A stereotypical example of such a relationship is a bill of materials, in which one assembly is composed of subassemblies composed of more subassemblies, on down an indeterminate number of levels until you reach discrete parts at the bottom.

ANSI/ISO Recursive WITH (DB2)

DB2 supports the ANSI/ISO recursive use of WITH for querying hierarchical and recursive data. For example, to query the bill-of-materials structure in the `bill_of_materials` table, you could write:

```
WITH recursiveBOM
  (level, assembly_id, assembly_name,
   parent_assembly) AS
(SELECT 1,
       parent.assembly_id,
       parent.assembly_name,
       parent.parent_assembly
FROM bill_of_materials parent
WHERE parent.assembly_id=100
UNION ALL
SELECT parent.level+1,
       child.assembly_id,
       child.assembly_name,
       child.parent_assembly
FROM recursiveBOM parent, bill_of_materials child
WHERE child.parent_assembly = parent.assembly_id)
SELECT level, assembly_id,
       parent_assembly, assembly_name
FROM recursiveBOM;
```

Most of this statement consists of a subquery named `recursiveBOM` that is specified using the `WITH` clause. The subquery consists of two `SELECT`s unioned together:

- Consider the first SELECT as the union query’s starting point.
- Consider the second SELECT as defining the recursive link between parent and child rows.

The second SELECT brings in the children of the first. Because the second SELECT references the named subquery that it is part of (itself), the second SELECT recursively brings back rows returned by the second SELECT. The main SELECT kicks off all this recursion by simply selecting from the named subquery.

TIP

For a more in-depth explanation of what happens when a recursive WITH executes, read the article “Understanding the WITH Clause” at <http://gennick.com/with>.

DB2 returns recursive results in the following order, which differs from the order you’ll get from Oracle:

1. The root node
2. The root’s immediate children
3. The children of the root’s immediate children
4. And so forth.

To keep track of your depth in a hierarchy, create a LEVEL column as shown in the example query. Have the first SELECT return 1 as the value for that column. Have the second SELECT return `parent.level+1`. The root node will then have a level of 1, the root’s immediate children will be level 2, and so on, down to the bottom of the hierarchy.

CONNECT BY Syntax (Oracle)

Use the CONNECT BY syntax and related functions to write recursive queries in Oracle. While Oracle does not support ANSI’s recursive use of WITH, Oracle’s CONNECT BY feature set is arguably more expressive and easier to use.

CONNECT BY, START WITH, and PRIOR (Oracle)

To return data in a hierarchy, specify a starting node using `START WITH` and specify the parent-child relationship using `CONNECT BY`:

```
SELECT assembly_id, assembly_name, parent_assembly
FROM bill_of_materials
START WITH assembly_id = 100
CONNECT BY parent_assembly = PRIOR assembly_id;
```

ASSEMBLY_ID	ASSEMBLY_NAME	PARENT_ASSEMBLY
100	Automobile	
110	Combustion Engine	100
111	Piston	110
112	Air Filter	110
113	Spark Plug	110
114	Block	110
115	Starter System	110
116	Alternator	115
117	Battery	115
118	Starter Motor	115

...

The `START WITH` clause specifies the first row Oracle looks at. In this example, the database begins with assembly #100, the automobile. Use the `CONNECT BY` clause to specify the relationship between parent and child rows. In this bill of materials, as you navigate down the hierarchy, each child's `parent_assembly` must equal the parent's `assembly_id`.

Use the condition `START WITH parent_assembly IS NULL` to report on all assemblies and their subassemblies.

In a `CONNECT BY` query, the keyword `PRIOR` represents an operator that returns a column's value from the parent row. `PRIOR` is often used to define the recursive relationship, but you may also use `PRIOR` in `SELECT` lists, `WHERE` clauses, or anywhere else that you wish to reference a value from the current row's parent.

NOTE

CONNECT BY is not limited to hierarchical data. Any data linked in a recursive fashion is a candidate for CONNECT BY queries. See “Finding Flight Legs” at <http://gennick.com/flight>.

WHERE clauses (Oracle)

CONNECT BY queries may have WHERE clauses, but they often do not because the START WITH condition tends to identify the particular tree(s) of interest.

Joins (Oracle)

CONNECT BY queries may involve joins, in which case the following order of operations applies:

1. The join is materialized first, which means that any join predicates are evaluated first.
2. The CONNECT BY processing is applied to the rows returned from the join operation.
3. Any filtering predicates from the WHERE clause are applied to the results of the CONNECT BY operation.

Be careful! Don't write joins that inadvertently eliminate nodes from the hierarchy you are querying.

Hierarchical sorts (Oracle)

Oracle's CONNECT BY syntax implies an ordering in which each parent node is followed by its immediate children, with each child followed by its own immediate children, and so on. It's rare to write a standard ORDER BY clause into a CONNECT BY query because the resulting sort destroys the hierarchical ordering of the data. However, beginning in Oracle9i you can use the new ORDER SIBLINGS BY clause to sort each level independently without destroying the hierarchy:

```
SELECT assembly_id, assembly_name, parent_assembly
FROM bill_of_materials
```

```

START WITH assembly_id = 100
CONNECT BY parent_assembly = PRIOR assembly_id
ORDER SIBLINGS BY assembly_name;

```

ASSEMBLY_ID	ASSEMBLY_NAME	PARENT_ASSEMBLY
100	Automobile	
120	Body	100
122	Left Door	120
139	Left Door Frame	122
140	Left Window	122
141	Lock	122
123	Right Door	120
144	Lock	123
142	Right Door Frame	123
143	Right Window	123
121	Roof	120

Look carefully at these results; you'll see that at the first level underneath Automobile you have Body followed by Roof. Underneath Body, you have Left Door and Right Door. Each level in the hierarchy is sorted independently, yet each parent is still followed by its immediate children—the hierarchy remains intact.

Loops in hierarchical data (Oracle)

Hierarchical data can sometimes be malformed in that a row's child may also be that row's parent or ancestor. Such a situation leads to a *loop*. To detect such problems, add `NOCYCLE` to your `CONNECT BY` clause, and the `CONNECT_BY_ISCYCLE` pseudocolumn to your `SELECT` list:

```

SELECT RPAD(' ', 2*(LEVEL-1))
       || assembly_name assembly_name,
       quantity, CONNECT_BY_ISCYCLE
FROM   bill_of_materials
START WITH assembly_id = 100
CONNECT BY NOCYCLE parent_assembly = PRIOR assembly_id;

```

`NOCYCLE` prevents Oracle from following recursive loops in the data. `CONNECT_BY_ISCYCLE` returns 1 for any row having a child that is also a parent or ancestor.

To create a loop in the sample data for this book, execute the following statement:

```
UPDATE bill_of_materials
SET parent_assembly = 113
WHERE assembly_id=100;
```

Do not COMMIT this statement. Execute the preceding SELECT to see the one row with a problem child, and then ROLLBACK to restore your data to its original state.

CONNECT BY functions and operators (Oracle)

Oracle implements a number of helpful functions and operators to use in writing CONNECT BY queries:

CONNECT_BY_ISCYCLE

Returns 1 when a row's child is also its ancestor; otherwise returns 0. Use with CONNECT BY NOCYCLE. (Oracle Database 10g and higher)

CONNECT_BY_ISLEAF

Returns 1 for leaf rows, 0 for rows having children. (Oracle Database 10g and higher)

CONNECT_BY_ROOT(*column*) or CONNECT_BY_ROOT *column*

Returns a value from the root row. See PRIOR. (Oracle Database 10g and higher)

LEVEL

Returns 0 for the root node of a hierarchy, 1 for nodes just below the root, 2 for the next level of nodes, and so forth. LEVEL is commonly used to indent hierarchical results. For example, the incantation `RPAD(' ', 2*(LEVEL-1)) || first_column` may be used to indent each level two spaces underneath the other.

PRIOR(*column*) or PRIOR *column*

Returns a value from a row's parent. See CONNECT_BY_ROOT.

`SYS_CONNECT_BY_PATH` (*column*, *delimiter*)

Returns a concatenated list of *column* values in the path from the root to the current node. Each column value is preceded by a *delimiter*. Add `SYS_CONNECT_BY_PATH(assembly_id, '/')` to the SELECT list in the preceding section, and you'll get results such as `/100`, `/100/120`, and `/100/120/122`. (Oracle9i and higher)