
Handling ORA-01436 CONNECT BY PRIOR Errors when Extracting

A Net 2000 Ltd. Technical Note

Abstract

This technical note describes the cause of the *ORA-01436: CONNECT BY loop in user data* error and how to handle it.

For more information, technical notes and white papers please visit the DataBee web site at <http://www.DataBee.com> For a full discussion the many issues which must be handled when creating referentially correct subsets please download the Test Database Generation and Management white paper at: <http://www.DataBee.com/TestDBGenWhitePaper.pdf>

Implementation Details

The DataBee software is designed to create referentially correct subsets of large Oracle databases. Unlike our white papers, which are generic discussions and do not require the use of the DataBee software, this document is a technical note. It is necessarily a non-generic discussion of the issue and the DataBee software is required to implement the technique discussed below.

Net 2000 Ltd.
Info@Net2000Ltd.com
<http://www.Net2000Ltd.com>

Handling ORA-01436 CONNECT BY PRIOR Errors when Extracting

It often happens that data must be extracted from a table via a self-referential relationship. This relationship is either required by a foreign key or is implemented as a DataBee Table-To-Table extraction rule designed to enforce a logical relationship.

By default, the DataBee software will use a CONNECT BY PRIOR type SQL statement to extract the required data. This method is much faster than a using a standard SQL inner join. Lets look at some typical data and then a simple select statement which uses a CONNECT BY PRIOR.

| EMPNO | ENAME | JOB | MGR |
|-------|--------|-----------|------|
| 7369 | SMITH | CLERK | 7902 |
| 7499 | ALLEN | SALESMAN | 7698 |
| 7521 | WARD | SALESMAN | 7698 |
| 7566 | JONES | MANAGER | 7839 |
| 7654 | MARTIN | SALESMAN | 7698 |
| 7698 | BLAKE | MANAGER | 7839 |
| 7782 | CLARK | MANAGER | 7839 |
| 7788 | SCOTT | ANALYST | 7566 |
| 7839 | KING | PRESIDENT | |
| 7844 | TURNER | SALESMAN | 7698 |
| 7876 | ADAMS | CLERK | 7788 |
| 7900 | JAMES | CLERK | 7698 |
| 7902 | FORD | ANALYST | 7566 |
| 7934 | MILLER | CLERK | 7782 |

The data in each row is quite clearly related to the data in other rows. In this case the relationship is MGR to EMPNO. In other words, to remain referentially correct, if a row with a specific EMPNO is extracted then the row for that employees manager must be extracted as well. This would be expressed as a DataBee Table-To-Table rule like the one below...

```
EMP.MGR -> EMP.EMPNO
```

... which states that “for every row extracted for the EMP table, also extract the rows in the EMP table where the MGR column of the original extracted row matches the EMPNO column of the new row”. The Oracle CONNECT BY PRIOR type SQL statement is designed for exactly this purpose. Assume we have previously extracted a single row for the table and this row has an EMPNO of 7369. Executing a statement like ...

```
SELECT ENAME,EMPNO,MGR
FROM EMP
START WITH EMPNO=7369 CONNECT BY EMPNO = PRIOR MGR;
```

... quickly fetches all of the required rows in a single execution of the statement. The data would be returned like:

| ENAME | EMPNO | MGR |
|-------|-------|--------|
| SMITH | 7369 | 7902 |
| FORD | 7902 | 7566 |
| JONES | 7566 | 7839 |
| KING | 7839 | (null) |

Notice that the relationship chain is followed all the way up to the president KING who has a MGR value of (null). The CONNECT BY PRIOR relationship stops at that record because there is no further onward chain.

The ORA-01436: CONNECT BY Loop Error

The ORA-01436: CONNECT BY loop in user data error is generated when there is a circular path in the relationships. For example assume the data in the previous example was erroneously updated to have a MGR value of 7902 instead of (null) in the data for KING.

| ENAME | EMPNO | MGR |
|-------|-------|------|
| SMITH | 7369 | 7902 |
| FORD | 7902 | 7566 |
| JONES | 7566 | 7839 |
| KING | 7839 | 7902 |

<<< used to be (null), now says FORD is KING's MGR

This means that when the CONNECT BY PRIOR statement pulled the data for the KING record it would then have to go and fetch the data for the FORD row. However, since the FORD record has already been part of the result set, it means an endless loop would certainly develop. To prevent this the Oracle database issues an *ORA-01436: CONNECT BY loop in user data* error and does not return any rows.

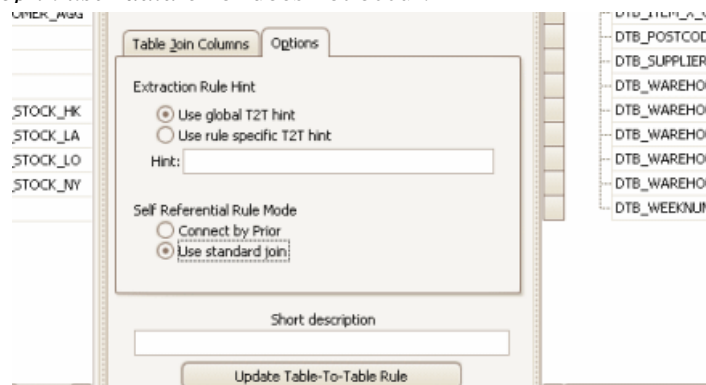
So, in summary, the ORA-01436 error simply means that it is not possible to use a CONNECT BY PRIOR SQL type statement on the data because there are infinite loops in the way the data reference each other.

Fixing the ORA-01436: CONNECT BY Loop Error in DataBee

The DataBee software has an alternate method of extracting data in situations such as described above. In simple terms what it does is fetch each row one by one carefully following the chain up to its conclusion. When it detects the duplicate (the KING record in the above example) it terminates the fetch. This ensures that the required referentially correct data is extracted. However, it is much slower because it requires one rule activation (a pass) for each link in the chain. DataBee is optimized to pursue multiple chains in parallel so the number of effective rule passes will be the length of the longest chain.

The default method of operation is CONNECT BY PRIOR since it is much faster than a standard join. However if the data in the table cannot support a CONNECT BY PRIOR relationship the standard join can be enabled.

To enable standard joins on a self referential extraction rule, simply open up the rule for editing in the Set Designer application (double click on it) and go to the Options tab. The Options tab is just behind the Table Join Columns tab in the center of the form. Select the option for "Use Standard Join" and save the rule. This will cause the rule to use the alternate extraction method and ensure the *ORA-01436: CONNECT BY loop in user data* error does not occur.



The Options Tab of a Table-To-Table Rule showing the Use Standard Join selection