Subject: Re: LINK2 Pattern Problem
Posted by AJM on Thu, 18 Jun 2009 13:36:39 GMT
View Forum Message <> Reply to Message

Ever since you brought this problem with the Oracle database to my attention via private email I have been seeking a proper solution. After playing around with tasks which use either the GROUP BY or HAVING clause I have made changes to the 'dml.oracle.php4/5.class.inc' files so that they can construct valid SQL statements using the information provided.

The problem is that each database vendor uses a different interpretation of the SQL standard, so what works in MySQL may not work in Oracle.

Take, for example, the GROUP BY clause. Oracle insists that every column in the SELECT list is also in the GROUP BY clause whereas MySQL is less strict. Although the Oracle implemetation was true in the SQL standard of 1991 this was changed in 1999 so that any column in the GROUP BY clause which is functionally dependent on another column in the GROUP BY clause does *NOT* have to be included. According to relational theory any non-key column on a table is functionally dependent on the primary key of that table, so if the GROUP BY clause contains the primary key then any non-key columns do not have to be specified.

The HAVING clause is used when the result set needs to be filtered by an aggregated column. This cannot be referenced in the WHERE clause as this is used before the results of the aggregation are known, so it has to be moved to the HAVING clause. MySQL is intelligent in that where the SELECT list contains an entry such as '<expression> AS aliasname' (where <expression> is an aggregation, a function or a subquery) it will allow the HAVING clause to reference that entry by 'aliasname' instead of '<expression>'. This is intelligent because it prevents <expression> from having to be defined and evaluated again. Oracle, on the other hand, is not so intelligent. You cannot use 'aliasname' in the HAVING clause, it has to be '<expression>', but ONLY if it is an aggregation, and not a function or a subquery. The only way around this is to put the SQL statement (without the HAVING clause) in a subquery, then enclose this in an outer query with the HAVING clause switched to the WHERE clause.

The attached file contains updates which work in MySQL, and the 'dml.oracle.php4/5.class.inc' file will automaically make changes to the GROUP BY and HAVING clauses which are required for Oracle.

This means that any changes which you made to either 'dict_table.class.inc' or 'std.table.class.inc' can be reversed out as they are no longer needed.

Try it out and let me know how you get on.

File Attachments

1) radicore-2009-06-18-update.zip, downloaded 2118 times