

Re: to optimize a select join

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2006-04/msg01936.html>

- *From:* Mladen Gogala <gogala@xxxxxxxxxxxxxx>
 - *Date:* Sun, 30 Apr 2006 04:20:43 GMT
-

On Sat, 29 Apr 2006 16:42:03 -0400, Jerry Stuckle wrote:

Maybe because the "rules of elements" don't allow all possibilities?

For instance – try this. List all records in table A, along with their matching records in table B if those records exist. Impossible without a Union, but very easy with a LEFT OUTER JOIN.

My preferred garden variety of databases (Oracle) has a syntax extension (+) for such cases. I find it to be much more understandable. In addition to that, internally the database does perform a union, it only uses the syntax to hide it.

There is minimal additional overhead to the JOIN syntax method. And much more flexibility. Additionally, you only need to learn one basic syntax for all JOIN cases.

Unfortunately, it makes the whole thing much less understandable and much less aesthetic. Here is the aesthetic problem:

```
SELECT * FROM  
emp e JOIN dept d ON e.deptno=d.deptno
```

What is the result of that join? Rows from EMP? Rows from DEPT?
Rows from both? It turns out that it is rows from both. It doesn't even save much space:

```
SELECT ename,job,dname,loc  
FROM emp e,dept d  
WHERE e.deptno=d.deptno
```

The statement above performs the same thing as the statement using the ANSI join. Saving space is especially dubious if we need an outer join.

Re: to optimize a select join

The statement

```
SELECT ename,job,dname,loc FROM  
emp e RIGHT OUTER JOIN dept d ON e.deptno=d.deptno
```

gives exactly the same result as

```
SELECT ename,job,dname,loc  
FROM emp e,dept d  
WHERE e.deptno(+)=d.deptno
```

The second SQL is much more clear, logical and pleasing to the eye.

Every database has its own syntax for outer joins. Database optimizers are finely tuned to its own syntax and usually have many more problems (and related bugs) with ANSI joins than with the usual, mathematically logical ones. In mathematical terms, "relation" is a subset of Cartesian product. Any subset. That means that we get to pick the elements of Cartesian product that we want in our relation. You can put even further conditions on relations. If the presence of pair (a,b) in the relation means that the pair (b,a) is not in the relation and the presence of pairs (a,b) and (b,c) implies the presence of the pair (a,c) then we are talking about the ordering relation. That can be refined further into strict ordering and well ordered sets (every set can be well ordered, provided we accept so called "axiom of choice", but that's slightly, just slightly, outside the scope of PHP group). That is the notation I love and understand. ANSI joins are ugly, support the dangerous illusion about "portable database applications" and make the statement much likely to encounter a bug in the database code. To make long story short, I hate ANSI joins, with passion.

--

<http://www.mgogala.com>

.

Re: to optimize a select join