

Re: Equilevant of Oracle for optional records

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.queries/2006-03/msg02742.html>

- *From:* "Michel Walsh" <vanderghast@VirusAreFunnierThanSpam>
 - *Date:* Fri, 31 Mar 2006 08:33:57 -0500
-

Hi,

I previously missed the point, indeed.

```
SELECT whatever
FROM t1 LEFT JOIN (SELECT * FROM t2 WHERE type = "ta") As ta
ON t1.id=ta.id
LEFT JOIN (SELECT * FROM t2 WHERE type="tb") As tb
ON t1.id = tb.id
```

and so on (may need to add parentheses). If there is only one, or less, record in ta such that ta.id=t1.id, and one, or less, tb.id=t1.id, there should be no duplication in the result.

Sure, if you have 26 columns to consider, that makes 26 joins, but do they appear in the SELECT clause? If so, I suspect you "flattened" t2 too soon, and a normalized t2 could be more useful, to build the query, and once this query is done, then flatten the result (with a Crosstab, probably) ?

----- related to the problem, but not "directly"-----

Note that Jet translates:

```
FROM tablex LEFT JOIN tabley ON tablex.f1=tabley.g1 AND tablex.f2=cte
```

into:

```
FROM tablex LEFT JOIN tabley ON tablex.f1=tabley.g1
WHERE tablex.f2=cte
```

which IS NOT accordingly to the standard, but more intuitive... As example, the Iqaluit Trauma:

```
SELECT a.city
FROM authors As a LEFT JOIN authorsBooks As b
ON a.authorID=b.authorID AND a.city = 'Iqaluit'
```

Re: Equilevant of Oracle for optional records

returns all records, in MS SQL Server (and probably Oracle), even those where a.city <> 'Iqaluit', but Jet returns no record, since no author lives in city='Iqaluit' (as known in database pubs).

Vanderghast, Access MVP

"Maurice W. Darr" <mdarr@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:ur%23wbKCVGHA.4608@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

All,

I really appreciate the responses but this appears to be trickier to explain than I thought. Bear with me while I elaborate.

The sample data structure is

T1 with Pk Id.

T2 with Pk Id,type so we can be sure than 0 or 1 records exist in T2 for each id,Type.

What I am looking for is a solution that returns:

- 1) One row per Id
- 2) A set of columns from tables TA,TB,...,Tx in the one row.
- 3) Each table in TA,TB,...,Tx is identified with a specific record type in the physical T2 table
- 4) If there is a correct type for the Id it will return the values in T2 as Tx where type='Tx'
- 5) If there is no record with the correct type will return nulls in the column set

The solutions offered so far are creative but the ones that work:

- 1) Produce multiple rows per Id
- 2) Become quickly unmanagable when joining using more than one Type, i.e. TA,TB,TC, etc.

For the relational database people I am denormalizing the table structure and turning multiple records in T2 into a repeating group in the result set. You don't want to store information in repeating groups but they are sometimes perfect for humans who need to do something with the data.

I am asking because the structure of T2 is a great tool to let users create custom attributes for something represented by T1 rows without re-coding the UI but I need to be able to write queries that treat the rows in T2 as if they were columns in T1.

Oracle handles this data structure perfectly but I suspect it might be ported to Sql Server someday and I already know cases where it would be

Re: Equilevant of Oracle for optional records

nice to query this structure in MS Access. Right now I flatten it prior to moving to Access.

Maurice

"Michel Walsh" <vanderghast@VirusAreFunnierThanSpam> wrote in message news:u8FrSzBVGHA.2704@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hi,

```
SELECT x.*, y.*
FROM ( SELECT * FROM t1 WHERE type="A") AS x
LEFT JOIN
( SELECT * FROM t2 WHERE type="B") AS y
ON x.id = y.id
```

is what I guess you want. NO record with t1.type <> "A" will appear, but those with t1.type="A" would either be associated with their counter part in t2 (sharing a common id) where type="B", either will be associated with NULLs.

Sure, you can replace the * with explicit lists.

Hoping it may help,
Vanderghast, Access MVP

"Maurice W. Darr" <mdarr@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:uAFaj94UGHA.5828@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Any or all,

The Oracle syntax for outer joins makes it possible to join to a table by record type but I cannot find an equilevant in Access. Here's what I mean.

I have two tables
T1 (id)
T2 (id,type,data)
I want a join that will return several records from T2 joined to T1 but with different Type values
In Oracle this is straightforward.

```
Select t1.id,ta.data a_data,tb.data b_data
```

Re: Equilevant of Oracle for optional records

```
from t1, t2 ta,t2 tb
where ta.id+=t1.id
and ta.type+="A"
and tb.id+=t1.id
and tb.type+="B"
```

This query will return all the records in T1. If there is a corresponding record in T2 with a type of 'A' it will return a record

otherwise it will return null in a_data. The same is true of b_data.

Either a null if there are no records with a 'B' or a value.

In Access I cannot find a way to do this in Access.

It would be nice to use

```
T1 LEFT JOIN T2 AS Ta ON T1.ID = Ta.CAPID and
ta.type='A'
```

This does not work.

A straight left join with a filter by type will not return any records

if there is at least one alternate type in T2 for the join key.

Any ideas?

TIA, maurice