

Re: Help with Cartesian or cross joins

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.queries/2006-01/msg01101.html>

- *From:* Vincent Johns <vjohns@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 13 Jan 2006 17:20:31 GMT
-

Were you able to identify the mis-matched expression in [Q_Unused]? From what you say, that's the only remaining problem. Maybe you're trying to mix text strings with numbers, or something like that. I don't have time right now to guess what went wrong there (but the version I tried appeared to work).

BTW, I suggest you avoid an expression like

```
Not IN("02","06","07","08","14","15",)
```

in your SQL on the basis that it grossly clutters the SQL and requires you to keep editing it whenever the set of numbers changes. More maintainable is putting those numbers into a Table (or computing them via a Query), so you can leave all your SQL alone once you have it working.

If you wish, you may email me a (sanitized and ZIPped) copy of your database, and I'll try to look at it in the next couple of days. (But I don't guarantee anything, and you may get better answers by posting in the newsgroup.)

```
-- Vincent Johns <vjohns@xxxxxxxxxxxxxxxxxxxxxx>  
Please feel free to quote anything I say here.
```

tmaxwell wrote:

```
I posted this back in November, but I had to stop and build a  
number of Financial Reports in VB so I put this off to the  
side. Now I really need to finish this DB. I ve tried to write  
this quite a few number of ways, but none worked all the way.  
I will post the original question and the response I got that
```

Re: Help with Cartesian or cross joins

is close.

This is the original post:

I took my main table NameAddr this file contains all of our customers with assigned customer numbers. Each of the 14 branches has a range of usage between 1 and 9999. Branch 01 only uses 1680 that leaves 8319 number assignable. I created a 1-9999 table and right joined them, this should give me all the numbers USED and all the numbers LEFT OVER

```
SELECT NAMEADDR.[co-number] AS Branch, NAMEADDR.[cust-number]
AS CustomerNumberAssinged, CustomerN.CustomerNumbers

FROM NAMEADDR RIGHT JOIN CustomerN ON NAMEADDR.[cust-number] =
CustomerN.CustomerNumbers

WHERE (((NAMEADDR.[co-number])<>"Is Not Null" And WHERE
NAMEADDR.[co-number] Is Not Null And NAMEADDR.[co-number] Not
IN("02","06","07","08","14","15",)

ORDER BY [NAMEADDR].[co-number], [NAMEADDR].[cust-number];
```

This did not work correctly, I still need the UNUSED numbers per BRANCH. I response I got (which I will list below) is close, but I get a mis-matched expression. I pulled it apart to see what was not working, but nada. The first two queries work fine, it's the Q_Unused that is still getting an error. Any suggestions would be most appreciated!

Here is the response:

Piggybacking on Marshall's clever idea of using the Cartesian product of a small Table with itself, I define a Table containing only 10 numbers:

[tbl 0-9] Table Datasheet View:

number

0

Re: Help with Cartesian or cross joins

1
2
3
4
5
6
7
8
9

I also have a Table that specifies the maximum number of accounts at each Branch. (I changed Branch "01" from 1680 to 998 to make the example work better. You should set them to whatever is correct.)

[Branches] Table Datasheet View:

```
Branch NumberOfAccounts
-----
01 998
02 997
03 500
04 74
```

...

And we have the Table of existing, and thus unavailable, account numbers at each Branch. I have included only two Branches, to make the example more concise:

[NAMEADDR] Table Datasheet View:

```
co-number cust-number
-----
01 993
01 995
01 996
02 991
02 995
```

Now we create a Query to list all 9999 possible customer numbers for any Branch, including those that may be in use:

[Q_9999] SQL:

```
SELECT [1000].number*1000+[100].number*100+[10].number*10
+[1].number AS [Number]
FROM [tbl 0-9] AS 1000, [tbl 0-9] AS 100, [tbl 0-9] AS 10,
[tbl 0-9] AS 1
WHERE
((( [1000].[number]*1000+[100].[number]*100+[10].[number]*10
+[1].[number])<0))
ORDER BY [1000].number*1000+[100].number*100
+[10].number*10+[1].number;
```

(I expect that this occupies considerably less space than a 10,000-record Table.) As you might expect, the contents are kind of boring, and I've omitted most of them here:

[Q_9999] Query Datasheet View:

```
Number
-----
```

1

Re: Help with Cartesian or cross joins

2
3
4
5
...
9997
9998
9999

Now I define a Query that lists all of the possible account numbers for each Branch, up to the limit specified in [Branches].[NumberOfAccounts]:

[Q_Combo] SQL:

```
SELECT Branches.Branch, Q_9999.Number
FROM Branches, Q_9999
WHERE (
  ((Branches.Branch)<="02") AND
  ((Q_9999.Number)<=[Branches]![NumberOfAccounts]
  And (Q_9999.Number)=990
))
```

ORDER BY Branches.Branch, Q_9999.Number;

The lines "((Branches.Branch)<="02") AND "

and "And (Q_9999.Number)=990"

are included ONLY to make the example shorter. You need to erase both lines from this SQL. This Query lists account numbers up to 998 for Branch "01" and up to 997 for Branch "02", omitting the other numbers and Branches. The complete list, now, looks like this:

[Q_Combo] Query Datasheet View:

Branch number

```
01 990
01 991
01 992
01 993
01 994
01 995
01 996
01 997
01 998
02 990
02 991
02 992
02 993
02 994
02 995
02 996
02 997
```

Now we're ready to list the unused ones. We define one more Query:

[Q_Unused] SQL:

```
SELECT Q_Combo.Branch, Q_Combo.number
```

Re: Help with Cartesian or cross joins

```
FROM Q_Combo LEFT JOIN NAMEADDR
ON (Q_Combo.Branch = NAMEADDR.[co-number])
AND (Q_Combo.number = NAMEADDR.[cust-number])
WHERE (((NAMEADDR.[co-number]) Is Null))
ORDER BY Q_Combo.Branch, Q_Combo.number;
and -- voilà -- we have what I think you asked for, a list of
all the account numbers (up to the maximum allowed for each
Branch) that are not already listed in [NAMEADDR]:
[Q_Unused] Query Datasheet View:
Branch number
```

```
-----
01 990
01 991
01 992
01 994
01 997
01 998
02 990
02 992
02 993
02 994
02 996
02 997
```

Notice that account 995 is not listed for either Branch, but 993 is listed for Branch "02".