

It's working :-)

It's working :-)

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.queries/2008-06/msg00490.html>

- *From:* Eka1618 <Eka1618@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 11 Jun 2008 04:50:02 -0700
-

So I feel bad that I wasted your time reading my long story about my DB, but it is giving me the correct results..... I did some further testing this morning and everything is just as I need it to be. I have alot of data to look at and compare constantly. By the end of my work day everthing looks the same to me, so maybe that what happened yesterday.

Thank you again for all the help, I really appreciate it!

~Erica~

"Eka1618" wrote:

Hi Ken,

I didn't realize I left the single quotes in there, it was just something I tried. The query results that I get is without using the single quotes.

I understand what you are saying about using LIKE and =, That's why I do not understand why the results that come back from the query are as if I used a wildcard.

I am creating a test database for engineering technicians to store test data in and for others to request tests and then report on test results.

Each test REQUEST has a Lock, Key and Pattern Combination (sometimes several). I have broken my tables like this:

1. tblREQUEST can have many LOCKS (tblLock), many KEYS (tblKey) and many PATTERNS (tblPattern).
2. tblRequest can also have many TESTS (tblTest)

It's working :-)

It's working :-)

3. Each test is sent to a test queue (tblQueue) which has many RESULTS (tblResults)

4. Each test Queue also can have many REVIEWS (tblReview)

I have seperated tblTest and tblQueue because issuing a test doesn't mean that it should be automatically queued to be conducted. there is other information about test queue here in this table that cannot be anywhere else so I seperated the two.

The reason why tblLock, tblKey and tblPattern are seperated is because there are times when there are many locks that associate with only one key and one pattern; Times when there are many keys associated with one lock and one pattern and times where there are many patterns associated with one lock and one key. There is also a situation where there are many one lock/one key/many pattern combinations in which case I needed an additional two tables to enable the database to do so, and these table are only used in that situation.

I have gone through the table design over and over and have actually redone it 3 times because it is so complex. I cannot imagine another way to relate these tables to one another. I work independently as an intern on this assignment, no one here knows how to program...at all. Unless someone looks at this, I have to use my best judgement. Everything is working as it should though, I just do not understand these query results...

Hopefully this explained it more for you. My bosses want to launch this in the next couple weeks (I've been working on this since Jan), so I do not have much time to redesign. Thank you for all the help so far! If you can think of anything else, let me know :-)

~Erica~

"Ken Sheridan" wrote:

Erica:

One thing I notice is that you've enclosed the reference to each control in single quotes in the first part of each OR operation, e.g.

```
L_EWO_NO = '[forms].[frmCustomResultReport].[cmbLEWO]'
```

This is actually looking for rows which contain the string '[forms].[frmCustomResultReport].[cmbLEWO]' in the L_EWO_NO column, rather than the value selected in the combo box. It should simply be:

It's working :-)

It's working :-)

```
L_EWO_NO = [forms].[frmCustomResultReport].[cmbLEWO]
```

The reference to the control is actually a parameter.

However, I don't really see how this would explain the behaviour you are getting. In fact it should result in your getting no rows returned as we can presumably rule out the L_MAT_TYPE column containing the string '[forms].[frmCustomResultReport].[cmbLMatType]'. Using the LIKE operator

rather than the = operator would only produce the results you are getting if used with a wildcard character, e.g.

```
L_MAT_TYPE LIKE [forms].[frmCustomResultReport].[cmbLMatType] &
"%"
```

Used without the & "%" it would give the same results as the equality operator.

I doubt you need concern yourself with the 'null thing'. I suspect your combo boxes' RowSources have only the one column each, so it doesn't apply.

See what happens when you remove the single quotes around the parameters. If it still doesn't produce the desired results then I would think the problem probably stems from the underlying logical model, of which I have to admit my understanding is far from clear. It would perhaps help if you could explain what the three tables represent in terms of real world entities and how they are related to each other.

Ken Sheridan
Stafford, England

"Eka1618" wrote:

Hi Ken,

I have the query working like it should. But now I am getting results as if

I used the word "LIKE" as I had initially. I changed it a while ago and it is

still acting as if I have it in there. Here is my updated version of the query:

```
SELECT tblResults.Q_INC, tblResults.SAMPLE_NO,
tblResults.TENSION,
tblResults.NOTES, tblResults.OBSERVATIONS,
tblResults.L_PART_NO,
```

It's working :-)

```
tblResults.K_PART_NO, tblResults.PAT_NO,
tblResults.PAT_SIZE
FROM tblResults
WHERE ((tblResults.L_ID
In (select L_ID from tblLock
Where (L_EWO_NO =
'[forms].[frmCustomResultReport].[cmbLEWO]' OR
[forms].[frmCustomResultReport].[cmbLEWO] = "<all>")
AND (L_PART_NO =
'[forms].[frmCustomResultReport].[cmbLPart]' OR
[forms].[frmCustomResultReport].[cmbLPart] = "<all>")
AND (L_SKID_NO =
'[forms].[frmCustomResultReport].[cmbLSKID]' OR
[forms].[frmCustomResultReport].[cmbLSKID] = "<all>")
AND (L_MAT_TYPE =
'[forms].[frmCustomResultReport].[cmbLMatType]' OR
[forms].[frmCustomResultReport].[cmbLMatType] =
"<all>") AND (L_MAT_HEAT =
'[forms].[frmCustomResultReport].[cmbLMatHeat]' OR
[forms].[frmCustomResultReport].[cmbLMatHeat] =
"<all>") AND (L_HARD =
'[forms].[frmCustomResultReport].[cmbLHard]' OR
[forms].[frmCustomResultReport].[cmbLHard] = "<all>")
AND (L_PLAT =
'[forms].[frmCustomResultReport].[cmbLPlat]' OR
[forms].[frmCustomResultReport].[cmbLPlat] = "<all>")
AND (L_TOP_COAT =
'[forms].[frmCustomResultReport].[cmbLCoat]' OR
[forms].[frmCustomResultReport].[cmbLCoat] = "<all>")
AND (L_THREAD =
'[forms].[frmCustomResultReport].[cmbLThread]' OR
[forms].[frmCustomResultReport].[cmbLThread] = "<all>")
)
AND (tblResults.K_ID) IN (Select K_ID from tblKey
Where( K_EWO_NO =
'[forms].[frmCustomResultReport].[cmbKEWO]' OR
[forms].[frmCustomResultReport].[cmbKEWO] = "<all>")
AND (K_PART_NO =
'[forms].[frmCustomResultReport].[cmbKPart]' OR
[forms].[frmCustomResultReport].[cmbKPart] = "<all>")
AND (K_SKID_NO =
'[forms].[frmCustomResultReport].[cmbKSKID]' OR
[forms].[frmCustomResultReport].[cmbKSKID] = "<all>")
AND (K_MAT_TYPE =
'[forms].[frmCustomResultReport].[cmbKMatType]' OR
[forms].[frmCustomResultReport].[cmbKMatType] =
"<all>") AND (K_MAT_HEAT =
'[forms].[frmCustomResultReport].[cmbKMatHeat]' OR
[forms].[frmCustomResultReport].[cmbKMatHeat] =
"<all>") AND (K_HARD =
'[forms].[frmCustomResultReport].[cmbKHard]' OR
```

It's working :-)

It's working :-)

```
[forms].[frmCustomResultReport].[cmbKHard] = "<all>")
AND (K_PLAT =
'[forms].[frmCustomResultReport].[cmbKPlat]' OR
[forms].[frmCustomResultReport].[cmbKPlat] = "<all>")
AND (K_TOP_COAT =
'[forms].[frmCustomResultReport].[cmbKCoat]' OR
[forms].[frmCustomResultReport].[cmbKCoat] = "<all>") )
);
```

If I say show all results where L_MAT_TYPE = 5, I get results where the L_MAT_TYPE = 56...

I tried doing the null thing, but I didn't understand it much, but what I have seems to work. Please let me know what you think i should do. Thank you!

~Erica~

"Ken Sheridan" wrote:

Erica:

Let's hope so, but if you do need any further help post back.

Ken Sheridan
Stafford, England

"Eka1618" wrote:

Now that I think about it, I think I still may need to track the lock by the L_ID...

If I say where L_ID in

It's working :-)

It's working :-)

(Select L_ID where)
then I will get each L_ID
where L_PART_NO = this
and/or L_MAT_Heat = that.
Then the out put would be
each result using a L_ID
listed in the query's results.

Maybe I just solved it
myself, we'll see...lol

"Eka1618" wrote:

Ken,

I rearranged
the query to
how you
wrote it and
I think it
might have
done
the same
thing as
what I
ended up
with
yesterday
using the iif
statement
(this was is
simpler
though).

I Believe
the query
works the
way it is
written to,
however it's
still not
producing

It's working :-)

It's working :-)

the results i
need. Let
me explain
tblLock:

table lock
consists of
L_ID (PK)
that is an
AutoNumber
field. so
there may
be times
where you
will have
the same
L_PART_NO,
but different
variations
of
it, hence the
need for
L_ID.

In my
results table
I wanted to
track which
lock they
were using,
but I
tracked it by
the
L_PART_NO
instead of
L_ID
because the
users want
to see
this
information.

I am trying
to establish
a way to
track the
results that
have a
L_PART_NO
of
this or a

It's working :-)

It's working :-)

MAT_HEAT
of that. It
may not be
necessary to
have
L_PART_NO
in the
results
table, but I
do not
know how
to link these
two tables
together
because the
are not
directly
connected
in table
relationships
for other
reasons.

Basically, I
am just
trying to
find the
record(s) in
the lock
table that
match the
values
selected by
the user in
the combo
boxes if that
makes any
sense. so if
a person
select a
value only
for
MAT_HEAT,
I want to
see all
results
where the
Lock Part
Number's
MAT_HEAT
value

It's working :-)

It's working :-)

matches
whatever
they
selected no
matter what
the lock part
number is.

I hope this
make
sense... My
table
relationships
must stay
the way
they are.
Everything
flows the
way it's
supposed
to, so I do
see the need
in changing
it. I just
need to
figure out
how to do
some
complex
reporting. I
do not have
a big
background
with it.

Let me
know if you
have any
other
suggestions,
thank you!

~Erica~

"Ken
Sheridan"
wrote:

Erica:

It's working :-)

It's working :-)

You
need
to
evaluate
whether
the
relevant
column
in
the
table
matches
the
value
in
the
combo
box,
OR
the
combo
box
=
"<All>".
Consequently
each
of
these
Boolean
OR
operations
will
evaluate
to
TRUE
for
rows
where
there
is
a
match
for
a
selected
value,
or
for
every
row
in

It's working :-)

It's working :-)

the
case
of
the
combo
box's
value
being
the
default
<All>

.

Each
OR
operation
is
parenthesised
to
force
it
to
evaluate
independently
and
they
are
all
tacked
together
in
Boolean
AND
operations.
Rows
will
only
be
returned
therefore
where
every
parenthesised
expression
evaluates
to
TRUE
for
that
row.

It's working :-)

It's working :-)

So
the
WHERE
condition
goes
like
this:

```
WHERE  
(L_EWO_NO  
=  
[forms].[frmCustomResultReport].[cmbLEWO]  
OR  
[forms].[frmCustomResultReport].[cmbLEWO]  
=  
"<All>")  
AND  
(L_PART_NO  
=  
[forms].[frmCustomResultReport].[cmbLPart]  
OR  
[forms].[frmCustomResultReport].[cmbLPart]  
=  
"<All>")  
<  
and  
so  
on  
to>  
AND  
(L_THREAD  
=  
[forms].[frmCustomResultReport].[cmbLThread]  
OR  
[forms].[frmCustomResultReport].[cmbLThread]  
=  
"<All>")
```

NB:
only
use
the
LIKE
operator
if
you
are
matching
partial
values,
in

It's working :-)

It's working :-)

which
case
you'd
incorporate
wildcard
characters,
otherwise
use
the
normal
equality
operator
(the
equals
sign).
If
you
want
selecting
'Smith'
to
return
'Smith',
'Smithson',
'Smithereen',
'Arrowsmith'
etc
then
use
LastName
LIKE
"*"
&
cboLastName
&
"*",
but
to
simply
return
'Smith'
use
LastName
=
cboLastName.

The
SQL
above
assumes
that

It's working :-)

It's working :-)

<All>
is
in
the
bound
columns
of
the
combo
boxes'
RowSources
in
each
case,
and
not
in
a
visible
column
corresponding
to
a
hidden
bound
column
(e.g.
cases
such
as
where
a
numeric
EmployeeID
is
in
a
hidden
bound
column
but
FirstName
&
"
"
&
LastName
is
in
the
visible

It's working :-)

It's working :-)

column).
In
such
circumstances
I'd
usually
leave
the
bound
column
for
<All>
Null,
in
which
case
rather
than
=
"<All>"
I'd
use
IS
NULL
(do
not
use
'=
NULL'
note;
nothing
equals
NULL,
not
even
another
NULL).

Ken
Sheridan
Stafford,
England

"Eka1618"
wrote:

Hello,

I
am

It's working :-)

It's working :-)

trying
to
create
a
query
that
will
take
values
from
a
form
consisting
of
several
combo
boxes
as
its
criteria.

What
I
want
to
do
is
show
all
records
that
meet
the
criteria
that
was
selected.
For
instance;
if
you
have
a
form
that
consists
of
cmbA,
cmbB
and
cmbC

It's working :-)

It's working :-)

and
you
choose
a
value
in
cmbA
&
cmbB
only
the
query
will
return
results
that
match
cmbA
AND
cmbB.

right
now,
the
query
returns
results
that
match
every
cmbo
box
value
even
if
the
user
chose
<all>
(the
default
value)
I've
tried
using
OR
but
that
is
not
what

It's working :-)

It's working :-)

I
want
to
use
either.

I
am
wondering
if
this
can
even
be
done...
Here
is
my
code
for
the
query:

```
SELECT  
tblResults.Q_INC,  
tblResults.SAMPLE_NO,  
tblResults.TENSION,  
tblResults.NOTES,  
tblResults.OBSERVATIONS,  
tblResults.L_PART_NO,
```