
Article

[Kyle Baxter](#) · Aug 10, 2016 3m read

Improve SQL Performance for Date Queries, AGAIN!

Date range queries going too slow for you? SQL Performance got you down? I have ANOTHER weird trick that SQL Developers don't want you to know!*

If you haven't looked at the subquery trick for date ranges, check out my last post:

<https://community.intersystems.com/post/improve-sql-performance-date-ran...>

Why are we always looking at date queries? BECAUSE THEY'RE IMPORTANT! It's reporting, it's statistics, it's the numbers you show your boss to prove you're doing awesome work (which you are, of course!). So let's look at a similar table from last time, but with MAXLEN and MINLEN actually defined properly:

```
Class User.DateQ extends %Persistent
{
    Property DateSubmitted as %Date;
    Property Data as %String (MAXLEN=200, MINLEN=100);
    Index DateIdx on DateSubmitted;
}
```

So now you want to get all the data from the last month. You write a query like so and think "Nice job":

```
SELECT ID, DateSubmitted, Data
FROM DateQ
WHERE TO_DATE(DateSubmitted, 'MM/DD/YYYY') BETWEEN TO_DATE('07/01/2016', 'MM/DD/YYYY')
AND TO_DATE('07/31/2016', 'MM/DD/YYYY')
```

That is a very reasonable query (and such nice formatting)! However, when we look at the query plan we see:

Query Plan

- Call [module B](#), which populates bitmap temp-file A.
- Read bitmap temp-file A, looping on ID.
- For each row:

Output the row.

•

[module B](#)

- Read index map SQLUser.DateQ.DateIdx, looping on DateSubmitted and ID.
- For each row:
- Read master map SQLUser.DateQ.IDKEY, using the given idkey value.

And you're thinking, "Looping on DateSubmitted!? I gave it a range!" And right you did! However, because you put the field into a TODATE function, we don't know how to format the data. Technically you could have screwed around to get different pieces of data by purposefully putting in the wrong format (valid, if weird). So what do you do? Well, you remove TODATE! So long as you use the correct Date format, your query will return the right results and will perform much better.

The Query:

```
SELECT ID, DateSubmitted, Data
FROM DateQ
WHERE DateSubmitted BETWEEN TO_DATE('07/01/2016','MM/DD/YYYY') AND TO_DATE('07/31/2016','MM/DD/YYYY')
```

The plan:

Query Plan

Relative cost = 487364 <===== HEY! I told you not to look at this!!!

- Call [module B](#), which populates bitmap temp-file A.
- Read bitmap temp-file A, looping on ID.
- For each row:

Output the row.

•

[module B](#)

- Read index map SQLUser.DateQ.DateIdx, looping on DateSubmitted (with a range condition) and ID.
- For each row:
- Add ID bit to bitmap temp-file A.

And now we can see that we're using the range condition in the date index! This trick is especially important if this table is going to grow and we always want to only see a subset of data.

So remember: TODATE is NOT for date columns! Just use the format right for your current select mode.

Questions? Comments? Leave them below!

*Of course SQL Developers want you to know this, I'm just using the internet rule that shows that titles like this get more traffic.

[#Code Snippet](#) [#Performance](#) [#SQL](#) [#Caché](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/improve-sql-performance-date-queries-again>