

Question

[Jeremy Forsyth](#) · May 10, 2019

SQL query with Count function running slow

Cache version: Cache for Windows (x86-64) 2017.2.1 (Build 8013U)

Good Afternoon,

I have a co-worker who is trying to run the below query via ODBC. The issue is that the query appears to be running extremely slow (nearly 2 hours).

```
SELECT A.RecNo, T.SDSInstID, T.TranEffectDate, COUNT(T.InternalTXID) as NoofTransactions
FROM SDSDATA.Transaction T
INNER JOIN SDSDATA.DataFeed A
on A.BankAccountBSB = T.BankAccountBSB
and A.BankAccountNo = T.BankAccountNo
WHERE T.TranEffectDate >= DATEADD(yy,-1,CURRENTDATE)
GROUP BY A.RecNo, T.SDSInstID, T.TranEffectDate
```

Below is the generated query plan

Statement Text

```
DECLARE QRS CURSOR FOR SELECT A . RecNo , T . SDSInstID , T . TranEffectDate , COUNT ( T .
InternalTXID ) AS NoofTransactions FROM SDSDATA . Transaction T INNER JOIN SDSDATA .
DataFeed A ON A . BankAccountBSB = T . BankAccountBSB AND A . BankAccountNo = T .
BankAccountNo WHERE T . TranEffectDate >= DATEADD ( yy , - ? , CURRENTDATE ) GROUP BY A .
RecNo , T . SDSInstID , T . TranEffectDate
```

Query Plan

Relative cost = 119327452

- Call [module C](#), which populates temp-file B.
- Read temp-file B, looping on the hashing subscript.
- For each row:
 - Output the row.
-

[module C](#)

- Call [module B](#), which populates bitmap temp-file A.
- Read bitmap temp-file A, looping on ID.
- For each row:
- Read master map `SDSDData.Transaction.IDKEY`, using the given idkey value.
Read index map `SDSDData.DataFeed.BSBAccountNoIndex`, using the given `%SQLUPPER(BankAccountBSB)` and `%SQLUPPER(BankAccountNo)`, and looping on ID.
For each row:
Read master map `SDSDData.DataFeed.IDKEY`, using the given idkey value.
Check distinct values for `RecNo`, `%SQLUPPER(SDSInstID)`, and `TranEffectDate` using temp-file B, subscripted by a hashing of these values.
For each distinct row:
Add a row to temp-file B, subscripted by the hashing, with node data of `InternalTXID`, `TranEffectDate`, `%SQLUPPER(SDSInstID)`, and `RecNo`.
Update the accumulated count(`InternalTXID`) in temp-file B, subscripted by the hashing.
-

[module B](#)

- Read bitmap index `SDSDData.Transaction.TranEffectDateIndex`, looping on `TranEffectDate` and bitmap chunks.
- For each bitmap chunk:
- OR the bitmap chunk into bitmap temp-file A.
-

Numbers wise there are approximately 50,611 datafeed records and 12,098,782 transaction records. The transaction table grows by approx. 50,000 records a day.

Would anyone here have any advice as to how I can increase performance for this query?

Thanks in advance,

Jeremy

[#Indexing](#) [#ODBC](#) [#Performance](#) [#SQL](#) [#Caché](#)

SQL query with Count function running slow

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Source URL: <https://community.intersystems.com/post/sql-query-count-function-running-slow>