記事
Megumi Kakechi · 2023年2月12日 3m read

%TimeStamp型プロパティを使用した範囲指定のクエリが遅い場合の対処方法

これは InterSystems FAQ サイトの記事です。

日時検索で、TimeStamp型のクエリのパフォーマンスが出ない場合の対処法をご紹介します。

%TimeStamp データ型形式 (yyyy-mm-dd hh:mm:ss.ffff)は、人が読めることを目的とした ODBC
日付形式の文字列として格納されます。
そのため、どうしてもデータサイズが大きくなりクエリの実行に時間がかかってしまいます。
%TimeStamp型のプロパティにインデックスを作成している場合にも、クエリオプティマイザはそのインデックスを優先して最適化するようにはなっておりません。

IRISでは、POSIX
時刻(※)をサポートしているため、TimeStamp値を表すのに %Library.PosixTime データ型形式を使用できます。
こちらは、Integer型で保存され、**%Timestampの高性能な代替法**となります。

※POSIX 時間は、協定世界時 (UTC) 1970年1月1日
00:00:00（UNIXエポック）からの経過秒数として表されます。
1970-01-01 00:00:00より前の日付は、負の論理値で表されます。

%PosixTime データ型形式（エンコードされた 64 ビットの符号付き整数）は、%TimeStamp
データ型よりも少ないディスク容量とメモリ使用で済むため、%TimeStamp
よりも優れたパフォーマンスが期待されます。
%PosixTime でサポートされる最も古い日時は、0001-01-01 00:00:00 で、論理値は -6979664624441081856
です。
そして、サポートされる最後の日時は 9999-12-31 23:59:59.999999 で、論理値は 1406323805406846975 です。

現在日時を%PosixTime型で出力したり、%TimeStamp データ型形式にしたい場合は以下のように行えます。

```
USER>set ptime = ##Class(%Library.PosixTime).CurrentTimeStamp()

USER>write ptime
1154596073773251031
USER>write ##Class(%Library.PosixTime).LogicalToTimeStamp(ptime)
2023-01-24 14:06:06.404055
USER>write ##Class(%Library.PosixTime).TimeStampToLogical(
"2023-01-24 14:06:06.404055")
1154596073773251031
```

詳細はクラスリファレンスをご覧ください。
%Library.PosixTime

***

Cachéをお使いのお客様は、残念ながら%PosixTime データ型形式は使用できません。
「それでも何とか TimeStamp型のクエリのパフォーマンスを向上させたい！」

場合は、代替案として新しい計算フィールド(Calculated/SqlComputed)をテーブルに追加し、それらにインデックスを作成する方法があります。

こちらを使用する場合、タイムスタンプを $H 形式の文字列で保存するため、インデックスのデータサイズが小さくなった分だけのパフォーマンスが向上します。

例：
以下のような計算プロパティを追加します。

```
Property DataTS As %TimeStamp;
Property DataTS2 As %String
 [ Calculated, SqlComputeCode = { Set {*}=$ZDTH({DataTS},3
)}, SqlComputed ];  // ?????
```

インデックスは、"2022-09-21 11:31:00" の代わりに "66373,41452" で作成するようになります。

この場合、アプリで使用しているクエリ自体も変更する必要があります。
日付の比較は以下のように行います。

```
SELECT *
FROM
TEST.TABLE1
WHERE
     tochar(DataTS2, 'YYYY-MM-DD')||' '||tochar(substring(DataTS2,7,5), 'HH24:MI:SS')
     BETWEEN
     '2000-01-01 00:00:00' and '2022-12-31 23:59:59'

     /// $H ?????? 'YYYY-MM-DD HH24:MI:SS' ???????
     /// tochar(DataTS2, 'YYYY-MM-
DD')||' '||tochar(substring(DataTS2,7,5), 'HH24:MI:SS')
```

【関連】
TIMESTAMP型のフォーマットについて
日付範囲クエリのSQLパフォーマンスを改善する
TIMESTAMP型の項目に対して、TOCHAR() や TODATE() を用いた SELECT を実行するとエラーになります

#SQL #ヒントとコツ #InterSystems IRIS #InterSystems IRIS for Health

---

ソースURL:
https://jp.community.intersystems.com/post/timestamp%E5%9E%8B%E3%83%97%E3%83%AD%E3%83%91%E3%83%86%E3%82%A3%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%9F%E7%AF%84%E5%9B%B2%E6%8C%87%E5%AE%9A%E3%81%AE%E3%82%AF%E3%82%A8%E3%83%AA%E3%81%8C%E9%81%85%E3%81%84%E5%A0%B4%E5%90%88%E3%81%AE%E5%AF%BE%E5%87%A6%E6%96%B9%E6%B3%95