

## 記事

[Tomoko Furuzono](#) · 2020年8月4日 3m read

# 日付範囲クエリのSQLパフォーマンスを改善する

日付範囲クエリが極端に遅くなっていませんか？ SQLのパフォーマンスが低下していませんか？ 皆様のお役に立ちそうな、変わったトリックをご紹介します！

データ追加時にタイムスタンプを記録するクラスがある場合、そのデータはIDKEYの値で順番に並べられます（具体的には、テーブル内のすべてのIDとTimeStampについてTimeStamp1 < TimeStamp2 が成立し、かつ ID1 < ID2 が成立する場合に限られています）。この知識を活かせば、TimeStampで範囲指定をする場合よりもクエリのパフォーマンスを向上させることができます。次のテーブルについて考えてみましょう。

```
Class User.TSOrder extends %Persistent
{
Property TS as %TimeStamp;

Property Data as %String (MAXLEN=100, MINLEN=200);

Index TSIdx on TS;

Index Extent [type=bitmap, extent];
}
```

このテーブルに、過去30日間の日付を含む30,000,000行のランダムなレコードを入力すると、1日あたり1,000,000レコードが出来上がります。特定日の情報を照会する場合は、次のように記述します。

```
SELECT ID, TS, Data
FROM TSOrder
WHERE
    TS >= '2016-07-01 00:00:00.00000' AND
    TS <= '2016-07-01 23:59:59.999999'
```

間違いなく合理的なクエリです。しかし、筆者のシステムでは、このクエリを処理するのに2,172,792回のグローバル参照が発生し、7.2秒かかりました。ただし、IDとTimeStampが同じ順序になっていることが分かっている場合は、TimeStampsを使用してIDの範囲を取得することができます。次のクエリについて考えてみましょう。

```
SELECT ID, TS, Data
FROM TSOrder
WHERE
    ID >= (SELECT TOP 1 ID FROM TSOrder WHERE TS >='2016-07-01
00:00:00.00000' ORDER BY TS ASC) AND
    ID <= (SELECT TOP 1 ID FROM TSOrder WHERE TS <='2016-07-01
23:59:59.999999' ORDER BY TS DESC)
```

この新しいクエリは5.1秒で完了し、999,985回のグローバル参照しか発生しませんでした\*！

この手法は、インデックス付きフィールドが多いテーブルと複数のWHERE句を持つクエリに対してより実用的に適用できます。サブクエリから生成されたIDの範囲はビットマップ形式にすることができます。これにより、複数のインデックスを持つソリューションを取得するときに驚異的な速度を得ることができます。このトリックが機能する良い例には、Ens.MessageHeaderテーブルがあります。

これはこの手法が効果的であることを示す一例です。この手法は、同じテーブル内のWHERE句に多くの条件がある場合（また、これらには明確にインデックスが付けられています）にさらに効果を発揮します！ご自身のクエリでお試してください！

\*非常に大量のレコードを返すクエリをテストする場合、管理ポータルでこれを実行すると、ほとんどの時間がデ

## 日付範囲クエリのSQLパフォーマンスを改善する

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

---

ータの表示に費やされることとなります。 テストする場合は、埋め込みSQLや動的SQLを使用して結果を出力(表示)せずに確認し、グローバル参照件数の確認にはSQLシェルを使用するのが適切です。 また、その目的でSQL Statsを使用することもできます。

[#SQL #ワールドワイド・レスポンスセンター \(WRC\)](#)

---

### ソースURL:

<https://jp.community.intersystems.com/post/%E6%97%A5%E4%BB%98%E7%AF%84%E5%9B%B2%E3%82%AF%E3%82%A8%E3%83%AA%E3%81%AEsql%E3%83%91%E3%83%95%E3%82%A9%E3%83%BC%E3%83%9E%E3%83%B3%E3%82%B9%E3%82%92%E6%94%B9%E5%96%84%E3%81%99%E3%82%8B%C2%A0>