記事
Tomoko Furuzono · 2020年9月10日 3m read

日付範囲クエリのSQLパフォーマンスを改善する vol2

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？

日付範囲のサブクエリをまだご覧になっていない場合は、前回の投稿をご確認ください。
https://jp.community.intersystems.com/post/日付範囲クエリのsqlパフォーマンスを改善する

なぜ、こうも日付クエリに注目しているのでしょうか？ それは、日付クエリが重要だからです。 それは報告であり、統計であり、自分の素晴らしい仕事を上司に証明するための数字です（もちろん、あなたが実際にそうしていればの話ですが)。では、前回と同じようなテーブルを見てみましょう。ただし、実際には MAXLEN と MINLEN を次のように適切に定義します。

```
Class User.DateQ extends %Persistent
{
Property DateSubmitted as %Date;

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

Index DateIdx on DateSubmitted;
}
```

では、先月のすべてのデータを取得したい場合を見てみましょう。
次のようなクエリを書き、「良い仕事」を考えます。

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

これは非常に合理的なクエリです（また、非常に見やすいです）！ただし、クエリプランを見ると次のことがわかります。

クエリプラン
相対コスト = 787073 <===== カイルのメモ: 絶対にこの数字は見ないでください!!!!! • Call module B, which populates bitmap temp-file A. • Read bitmap temp-file A, looping on ID. • For each row: - Read master map SQLUser.DateQ.IDKEY, using the given idkey value. - Output the row.
Module: B
• Read index map SQLUser.DateQ.DateIdx, looping on DateSubmitted and ID. • For each row: - Add ID bit to bitmap temp-file A.

皆さんはこう思っていることでしょう。「DateSubmitted でループ！？
範囲を指定したのに！」 そう、そのとおりです！ ただし、フィールドを TODATE
関数に入れたため、どのようにデータを整形すべきかは分かりません。 技術的には、意図的に誤った形式（奇妙であっても有効になってしまう）を入力することで、さまざまなデータを取得することができます。

では、どうしますか？ そう、TODATE を削除するのです！
正しい日付形式を使用している限り、クエリは正しい結果を返し、パフォーマンスが大幅に向上します。

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

プラン：

クエリプラン
相対コスト = 487364 <==== ちょっと待ってください！これを見ないようにと言ったはずです!!! • Call module B, which populates bitmap temp-file A. • Read bitmap temp-file A, looping on ID. • For each row: - Read master map SQLUser.DateQ.IDKEY, using the given idkey value. - 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.

これで、日付インデックスで範囲条件が使用されていることが分かりました！ このトリックは、このテーブルが肥大化し、常にデータのサブセットのみを表示したくなった場合に特に重要です。

TODATE は日付カラムには適さないということを覚えておいてください。
現在選択しているモードに適した形式を使用してください。

ご質問はありますか？ コメントはありますか？ 以下にご記入ください！

#Code Snippet #SQL #Caché #InterSystems IRIS

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-vol2