

日々はOracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年4月8日 木曜日

ページネーションによるSQL - 追記

偶然、レポートのSQLが遅いのはなぜか、という質問がありました。

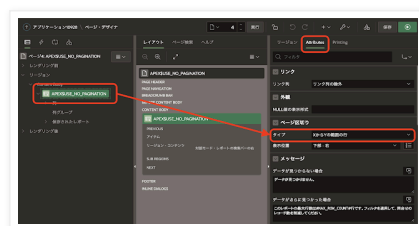
オプティマイザ・ヒントにAPEX\$USE_NO_PAGINATIONを設定したので、レポートを表示するために実行されるSELECT文は、ソースのSQL問合せで記述したSELECT文がそのまま実行されるはずでしょう？SQLコマンドなどからSELECT文を実行すると速いの、何故？

ソースに記述したSELECT文を、ほぼそのままに実行するには擬似ヒントのAPEX\$USE_NO_PAGINATIONをオプティマイザ・ヒントに設定する以外にも、いくつか解除が必要な設定があります。以下から、それらの設定を確認してみます。

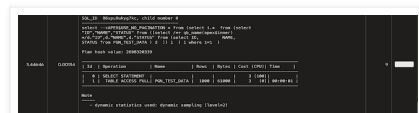
[前回の記事](#)で作成したアプリケーションを使います。

ページ区切りのタイプがXからYの範囲の行の場合

対話モード・レポートのデフォルト設定で、前回作成したAPEX\$USE_NO_PAGINATIONのページはこの設定です。



実行されるSELECT文を再掲します。



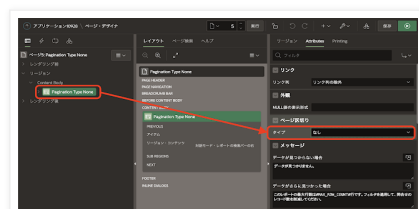
見やすくしたSELECT文は以下です。

```
select i.* from (
  select "ID","NAME","STATUS" from ((
    select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
      select ID,
             NAME,
             STATUS
      from PGN_TEST_DATA
    ) d
  )) i
) i where 1=1
```

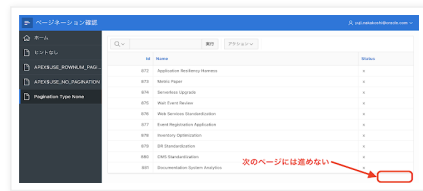
対話レポートのフィルタやその他の設定を行わなければ、ほとんどそのままSELECT文が実行されます。

ページ区切りのタイプがなしの場合

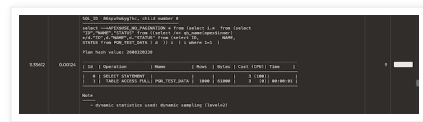
APEX\$USE_NO_PAGINATIONとして作成したページをコピーして、**ページ区切りのタイプをなし**に変更します。



ページ区切りがなしなので、次のページに進むボタンが表示されません。



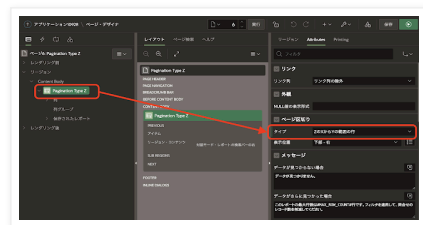
最初の表示で実行されるSELECT文を確認します。確認するログのパス情報はajax pluginではなく、**show**になります。



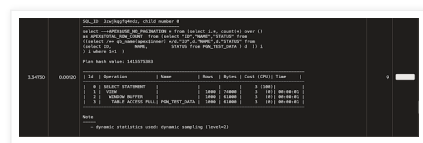
実行されるSELECT文は、**タイプがXからYの範囲の行**の場合と同じです。

ページ区切りのタイプがZのXからYの範囲の行の場合

APEX\$USE_NO_PAGINATIONとして作成したページをコピーして、**ページ区切りのタイプをZのXからYの範囲の行**に変更します。



実行されるSELECT文を確認します。



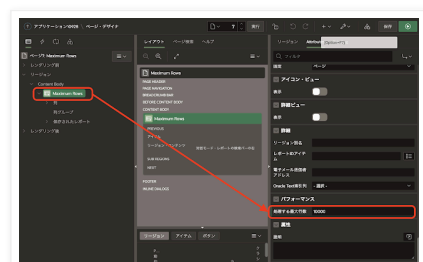
SELECT文を清書します。

```
select i.*, count(*) over () as APEX$TOTAL_ROW_COUNT from (  
  select "ID","NAME","STATUS" from (  
    select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (  
      select ID,  
             NAME,  
             STATUS  
      from PGN_TEST_DATA  
    ) d  
  ) i  
) i where 1=1
```

列にcount(*) over () as APEX\$TOTAL_ROW_COUNTが追加され、実行計画も若干変わっています。

パフォーマンスの処理する最大行数を指定

ページ区切りのタイプをZのXからYの範囲の行に設定したページをコピーして、**パフォーマンスの処理する最大行数を10000**と設定します。



実行されるSELECT文を確認します。



SELECT文を清書します

```
select i.*, count(*) over () as APEX$TOTAL_ROW_COUNT from (
  select "ID","NAME","STATUS" from ((
    select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
      select ID,
        NAME,
        STATUS
      from PGN_TEST_DATA
    ) d
  )) i
) i where 1=1 and rownum<=10001
```

検索条件にrownum<=10001が追加され、さらに実行計画が変わっています。

まとめると、レポートのSELECT文のパフォーマンスに問題があり、ページネーションが怪しいと思われる場合は、オプティマイザ・ヒントにAPEX\$USE_NO_PAGINATIONを追加するだけでなく、**ページ区切りのタイプはXからYの範囲の行**を選択し、**パフォーマンスの処理する最大行数を無指定**(ただしレポートの表示行数で制限する)にしてみましょう、ということです。

おまけ

表示列からSTATUSを除外(IDとNAMEだけを選択)し、NAME列に類似(Like)の条件を設定したときに実行されるSELECT文の例です。

```
select * from (
  select "ID","NAME" from (
    select i.* from (
      select "ID","NAME","STATUS" from ((
        select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
          select ID,
            NAME,
            STATUS
          from PGN_TEST_DATA
        ) d
      )) i
    ) i where 1=1
    and "NAME" like :apex$f1
  ) i
)
```

列NAMEで集計を行ったときに実行されたSELECT文の例です。

```
select "NAME",COUNT("ID") "APXWS_GBFC_01","APEX$TOTAL_ROW_COUNT" "AGGREGATE_ROW_COUNT",count(*)over() "APEX$TOTAL_ROW_COUNT" from (
  select i.*, (count("ID") over ()) "APEXWS_AGG_000", count(*) over () as APEX$TOTAL_ROW_COUNT from (
    select "ID","NAME","STATUS" from ((
      select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
        select ID,
          NAME,
          STATUS
        from PGN_TEST_DATA
      ) d
    )) i
  ) i where 1=1
) group by "NAME","APEX$TOTAL_ROW_COUNT"
```

レポートの条件や設定は、クライアント側のコンポーネントではなく、**サーバー側のSQLとして実行されている**ことが分かります。

以上で今回の説明は終了です。

確認のためにページを追加したアプリケーションのエクスポートを以下に置きました。
<https://github.com/ujnak/apexapps/blob/master/exports/pseudo-select-hint-aux.sql>

Oracle APEXのアプリケーション開発の参考になれば幸いです。

完

Yuji N. 時刻: 19:16

共有

<

ホーム

>

[ウェブバージョンを表示](#)

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)

Powered by Blogger.