日日是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7日水曜日

擬似ヒントとページネーションの動作について

そもそのOracle APEXの対話モード・レポートなどのページネーション（ページ送り）がどのような仕組みで行われているのか、紹介してみようと思い立ちました。元ネタはOracle APEXの開発者Carsten Czarskiさんのブログ記事Application Express 18 and Report Paginationです。開発の経緯や仕組みについて解説されています。ぜひ、ご一読を。

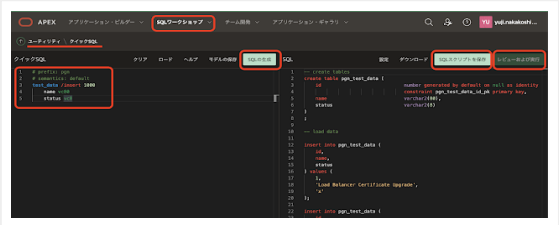
この記事では単に以下のケースでの動作を確認します。

1. 現在の動き(ROW_NUMBERを使う)
2. 擬似ヒントAPEX$USE_ROWNUM_PAGINATIONを指定したときの動き
3. 擬似ヒントAPEX$USE_NO_PAGINATIONを指定したときの動き

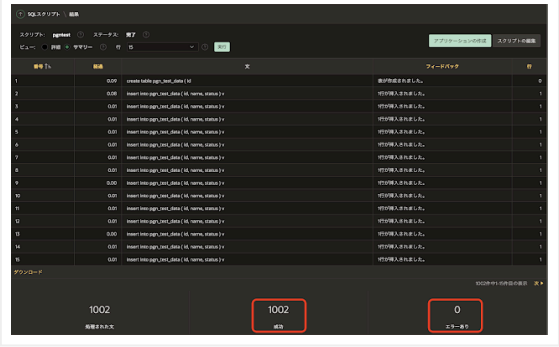
まず、テストに使う表とデータを準備します。クイックSQLで以下のモデルを定義します。

```
# prefix: pgn
# semantics: default
test_data /insert 1000
    name vc80
    status vc8
```

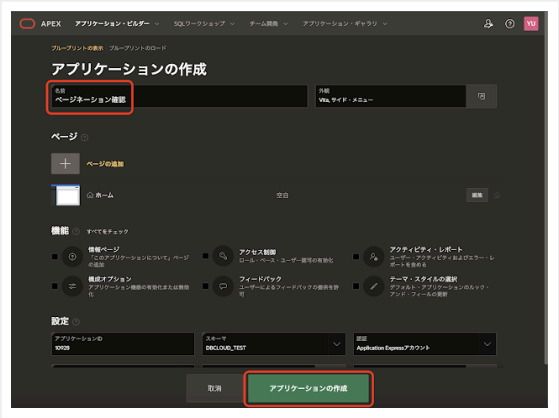
SQLワークショップから**ユーティリティ**の**クイックSQL**を実行し、上記のモデルを入力します。**SQLの生成**、**SQLスクリプトを保存**、そして**レビューおよび実行**を行います。



アプリケーションの作成は行いません。表PGN_TEST_DATAが作成され、初期のデータが1000行挿入されます。



動作確認を行うためのアプリケーションを作成します。アプリケーション作成ウィザードを起動し、**新規アプリケーション**として**名前**を**ページネーション確認**、それ以外はデフォルトのままで**アプリケーションの作成**を実行します。



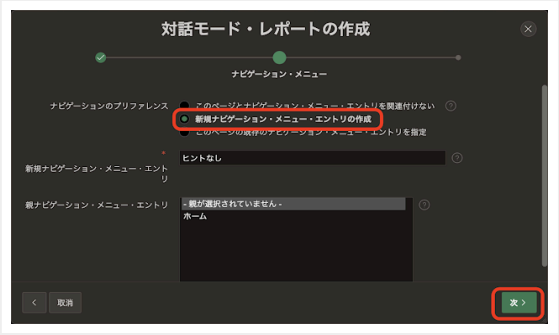
アプリケーションが作成されたら、対話モード・レポートのページを追加します。**ページの作成**をクリックします。

レポートをクリックします。

対話モード・レポートをクリックします。

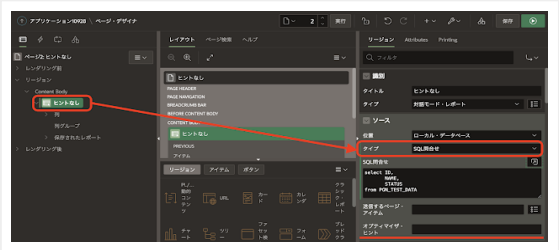
ページ名を**ヒントなし**として、**次**に進みます。

ナビゲーションのプリファレンスとして、**新規ナビゲーション・メニュー・エントリの作成**を選択し、**次**に進みます。



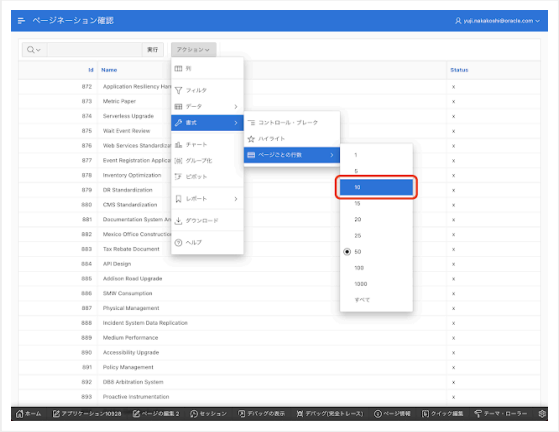
データ・ソースの**表/ビューの名前**に**PGN_TEST_DATA(表)**を指定し、**作成**を実行します。

対話モード・レポートのページが作成されたら、対話モード・レポートの**ソース**の**タイプ**を**SQL問合せ**に変更します。**オプティマイザ・ヒント**のプロパティが現れます。最初はここを無指定のままで、実行されるSQLを確認します。

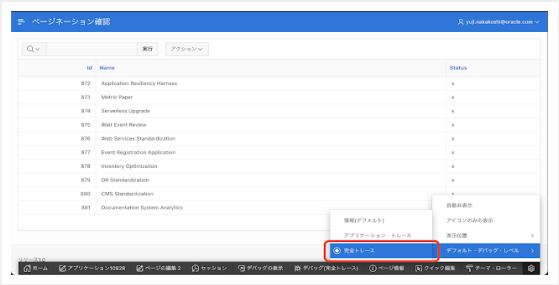


ページを実行します。

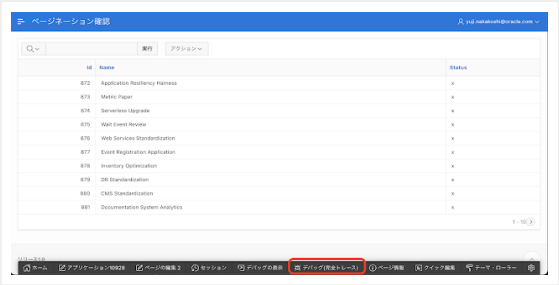
この作業は不要ですが、見やすくなるように（そして、スクリーンショットのサイズが小さくなるように）１ページの表示を10行にしました。



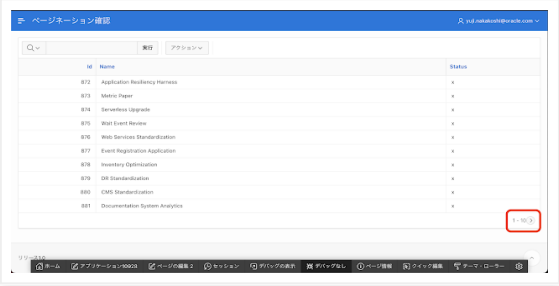
開発者ツールバーの右端、**歯車**のアイコンより**デフォルト・デバッグ・レベル**を**完全トレース**に変更します。実行されているSQLと実行計画をデバッグ・ログより取得します。



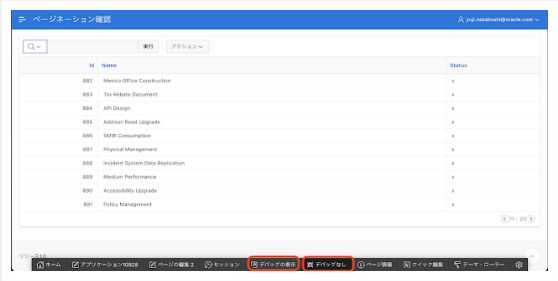
開発者ツールバーの**デバッグ(完全トレース)**をクリックし、デバッグを有効にします。



開発者ツールバーのデバッグ機能については、**デバッグなし**に変わります。ページ送りを実行します。

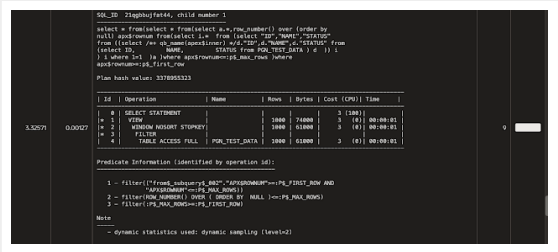


11行目から20行目までの10行が表示されます。**デバッグなし**をクリックしてデバッグ出力を止め、**デバッグの表示**をクリックします。



対話モード・レポートのページで発行された、直近のajax pluginがページネーションの処理です。

デバッグ・ログを開き、実行されたSQLを見つけ、実行計画を確認します。



実行されているSQLを、見やすいように書き直します。

```
select * from(
    select * from(
        select a.*, row_number() over (order by null) apx$rownum from(
            select i.* from (
                select "ID","NAME","STATUS" from((
                    select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
                        select  ID,
                                NAME,
                                STATUS
                        from PGN_TEST_DATA
                    ) d
                )) i
            ) i where 1=1
        ) a
    ) where apx$rownum<=:p$_max_rows
) where apx$rownum>=:p$_first_row
```

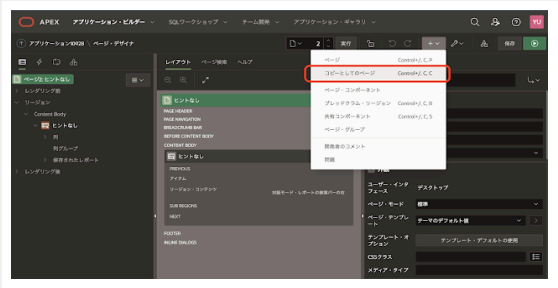
今回の問合せでは、**p$_max_rows**は21、**p$_first_row**には11が渡されています。分析関数のROW_NUMBERを使い、一画面に表示される行に限定して、SELECT文を実行していることが分かります。無駄にサブクエリが多く見えるかもしれませんが、サブクエリに対してフィルタなどの条件が追加されます。ページ送りをする度にSELECT文が実行されていることも分かります。

一旦データをコンポーネント側で全件取得して、ページ送りをコンポーネント側で実施するといった実装にはなっていません。行が大量にある表が対象だと、そのような実装では処理できないでしょう。最初の表示に長時間かかることが予想されます。また、行の取得時に行のチェックサムが生

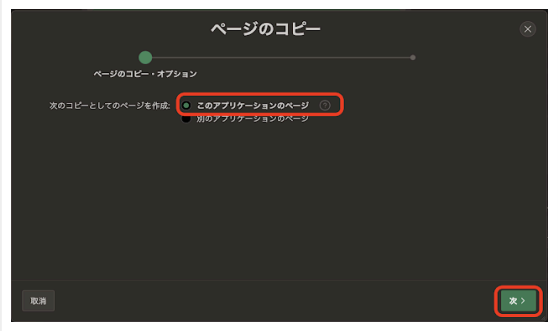
成され、ロストアップデートの保護に使用されます。ページごとに行を取得している場合、ロストアップデート保護の対象は画面に表示されているデータだけです。これが、全行取得していると、よりロストアップデートの保護違反が発生する確率が上がり、ユーザービリティが低下すると思われます。

次にヒントとしてAPEX$USE_ROWNUM_PAGINATIONを指定した場合の動作を確認します。

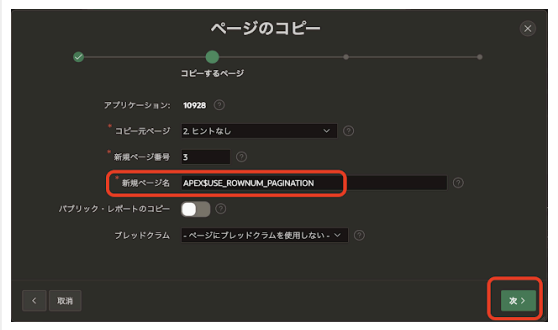
作成メニューから**コピーとしてのページ**を実行します。



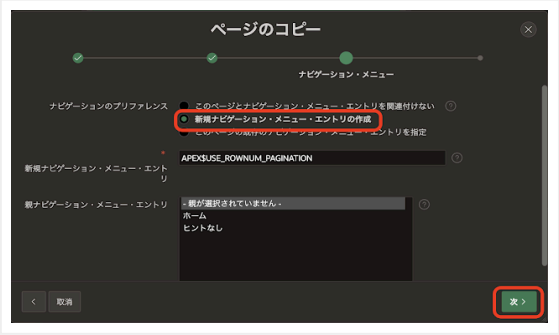
次のコピーとしてのページを作成として、**このアプリケーションのページ**を選択し、**次**に進みます。



新規ページ名として、**APEX$USE_ROWNUM_PAGINATION**を指定し、**次**に進みます。

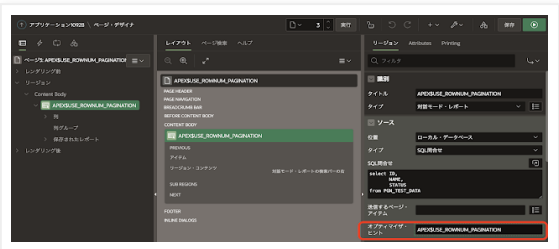


ナビゲーションのプリファレンスとして、**新規ナビゲーション・メニュー・エントリの作成**を選択し、**次**に進みます。

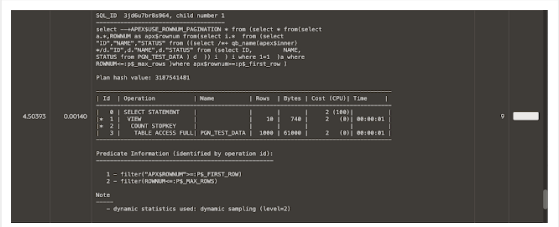


リージョンの**新しい値**をヒントなしから**APEX$USE_ROWNUM_PAGINATION**に変更し、**コピー**を実行します。

ページが作成されたら、対話モード・レポートのリージョンを選択し、**ソース**の**オプティマイザ・ヒント**に**APEX$USE_ROWNUM_PAGINATION**を指定します。



ページを実行し、ヒントなしのレポートとまったく同じ操作を行って、ページ送り時に実行されたSQLと実行計画を取得します。



実行されているSQLを、見やすいように書き直します。

```
select * from (
    select * from(
        select a.*,ROWNUM as apx$rownum from(
            select i.* from (
                select "ID","NAME","STATUS" from ((
                    select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
                        select ID,
```

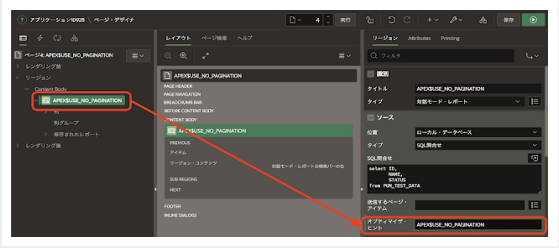
```
                                NAME,
                                STATUS
                            from PGN_TEST_DATA
                        ) d
                    )) i
                ) i where 1=1
            ) a
            where ROWNUM<=:p$_max_rows
        ) where apx$rownum>=:p$_first_row
)
```

基本的にROW_NUMBERを使った方がパフォーマンスが良いケースが多いため、そちらがデフォルトになっています。ただし、今回の例では、このヒントをつけた方がパフォーマンス面で有利なように見えます。

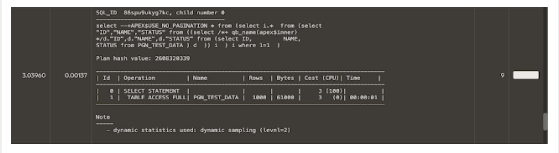
擬似ヒントを付けることで発行されるSQLが変わり、パフォーマンスに違いが出てくることを覚えておいてください。擬似ヒントを必要とするケースはあまりない（大抵は元のSQLそのものに問題がある）と考えていますが、プロパティに設定するだけなので試してみるのは簡単です。

APEX$USE_NO_PAGINATIONも確認してみます。ページのコピーやヒントの設定などの操作はAPEX$USE_ROWNUM_PAGINATIONと同じなので、手順は省略します。

対話モード・レポートの**ソース**の**オプティマイザ・ヒント**にAPEX$USE_NO_PAGINATIONを設定します。



ページを実行し、ヒントなしのレポートとまったく同じ操作を行って、ページ送り時に実行されたSQLと実行計画を取得します。



実行されているSQLを、見やすいように書き直します。

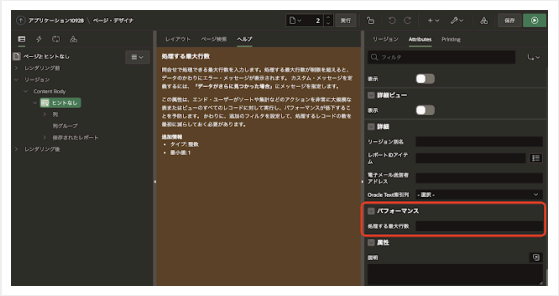
```
select i.* from (
    select "ID","NAME","STATUS" from ((
        select /*+ qb_name(apex$inner) */d."ID",d."NAME",d."STATUS" from (
            select ID,
                   NAME,
                   STATUS
            from PGN_TEST_DATA
        ) d
    )) i
) i where 1=1
```

擬似ヒント**APEX$USE_NO_PAGINATION**が指定されると、実行されるSELECT文としては全件を対象とします。そして 1 行目から表示される最初の行まで、行のフェッチを行い、表示される行について、フェッチ後にデータを取得します。表示対象の最後の行でフェッチを終了しカーソルを閉じます。

レポートの表示が 1 ページに収まる場合であれば、このヒントAPEX$USE_NO_PAGINATIONがパフォーマンス面で有利でしょう。

確認のために作成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.sql

ちなみに**対話モード・レポート**の**Attributes**には、**パーフォーマンス**の**プロパティ**が含まれます。**処理する最大行数**を指定することで、一度に取得する行数を制限することができます。



対話グリッドには**パフォーマンス**のプロパティに**遅延ロード**があります。初回表示時にデータの取得を待たずにページのレンダリングを行います。

Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 17:10

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