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6日火曜日

対話グリッドで特定時刻のデータを扱う

表題の件を聞かれたので、実装できるかどうかを確認しました。

対話グリッドにはロストアップデートの保護が実装されているので、本来はページのフェッチごと に他のセッションによる更新データが現れたとしても気にする必要はありません。

ロストアップデートの保護については、こちらの記事やこちらの記事で紹介しています。

まずテストデータを作ります。クイックSQLで以下のモデルを定義しました。

prefix: grd
semantics: default
paging_data /insert 1000
 name vc80

表GRD_PAGING_DATAが作成されます。表には主キー列IDと列NAMEが定義されています。サンプルデータとして1000行あらかじめ挿入します。**SQLの生成**をした後、**SQLスクリプトを保存**して**レビューおよび実行**をします。



空のアプリケーションを作成します。**名前を特定時刻での対話グリッド操作**とします。**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、ページ・デザイナでホーム・ページを開きます。Content Body に対話グリッドのリージョンを作成します。識別のタイトルを対話グリッド、タイプを対話グリッ

ド、ソースの位置はローカル・データベースとし、タイプとしてSQL問合せを選択します。

SQL問合せとして、以下のSELECT文を記述します。特定時刻のデータを取得するために、フラッシュバック・クエリを実行します。送信するページ・アイテムにP1_SCNを設定します。まだ、ページ・アイテムは作成していないため選択リストには現れません。直接記述します。

SELECT ID, NAME
FROM GRD_PAGING_DATA
AS OF SCN : P1_SCN



SCNを保持するページ・アイテムP1_SCNを作成します。**識別の名前をP1_SCN、タイプ**はテキスト・フィールドとし、ラベルをSCNとします。

ソースのフォーム・リージョンは、対象リージョンがフォームではないので、- 選択 - のままにし、タイプは式を選択します。言語はPL/SQLで、PL/SQL式に以下を指定します。

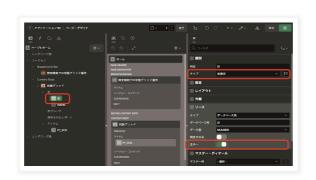
DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER

使用には**セッション・ステートの既存の値を常に置換**を選択し、**セッション・ステートの保持**として**リクエストごと(メモリーのみ)**を選択します。

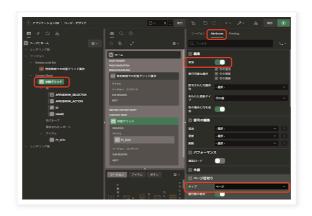
これで、ページが示される度にページ・アイテムP1_SCNに、その時点でのSCNが設定されます。対話グリッドではページングが行われても、AS OF SCNで指定されているSCNは変わらないため、常にページが表示された時点でのデータが表示されます。

対話グリッドを編集可能にします。

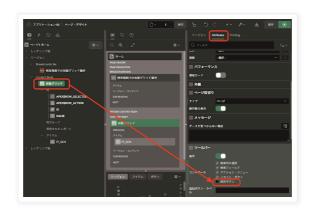
列IDを選択し、タイプを非表示、主キーをONに変更します。



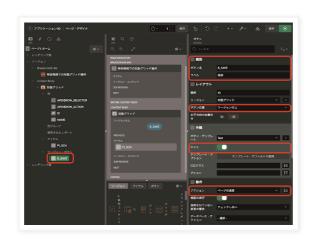
対話グリッドのリージョンを選択し、Attributesの編集の有効をONに変更します。ページ区切りはスクロールでもページでも構わないのですが、今回はページに変更します。



ツール・バーより**保存ボタン**のチェックを外します。対話グリッドの保存ボタンを使うと処理が対話グリッドで完結して、ページ・アイテムP1_SCNの操作ができません。そのため保存後に更新されたSCNをSELECT文に反映できません。



代わりにリージョンのボタンを追加します。ボタン名はB_SAVEとし、ラベルは保存とします。ボタン位置はリージョンの上に配置します。ホットはONとし、アクションはページの送信を選択します。



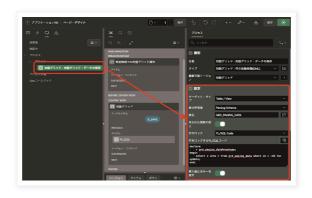
続いて、更新処理を行うプロセスを変更します。プロセスを選択し、ターゲット・タイプをRegion SourceからTable / Viewに変更し、特定時刻の検索結果ではなく元表を更新対象にします。表名にはGRD_PAGING_DATAを指定します。失われた更新の防止はON、行のロックはプロセスの設定を変更してもRegion Sourceを対象とするため、ロックかけられません。PL/SQL Codeに変更し、以下のPL/SQLコードを行をロックするPL/SQLコードとして記述します。

```
declare
```

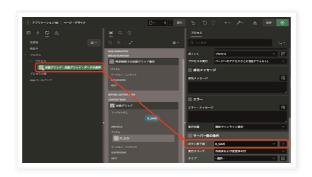
r grd_paging_data%rowtype;

begin

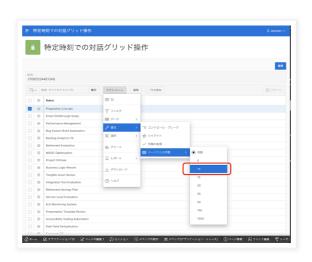
select * into r from grd_paging_data where id = :ID for update nowait;



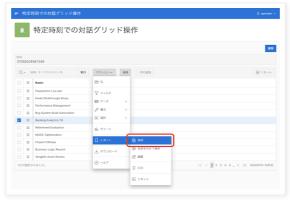
今回の例ではボタンがB_SAVEひとつしかないため必ずしも必要ではありませんが、サーバー側の条件のボタン押下時にB_SAVEを設定しておきます。



以上で試験用のアプリケーションは完成です。アプリケーションを実行してみます。ページングの効果をわかりやすくするため、表示行数を10行にします。**アクション**・メニューから**書式、ページごとの行数、10**を選びます。



レポートの保存を行います。



以上で実装は完了です。動作を確認してみます。

それぞれ別のブラウザで、アプリケーションを開きます。

最初のブラウザで2ページのデータを変更して保存します。



次のブラウザで2ページ目に移動しても、変更前のデータが見えています。ページを開いたときの SCNでデータが検索されているためです。



以上で確認作業も完了です。対話グリッドの保存ボタンが使えないため、保存を実行するとレポートの表示が先頭ページに戻ってしまうのは、致し方ないところです。

確認のために使用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/select-as-scn.sgl

Oracle APEXのアプリケーション開発の参考になれば幸いです。

★一厶)

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.