

日々是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。

2023年2月21日 火曜日

APEXのプロセスで発生する暗黙コミットの注意点

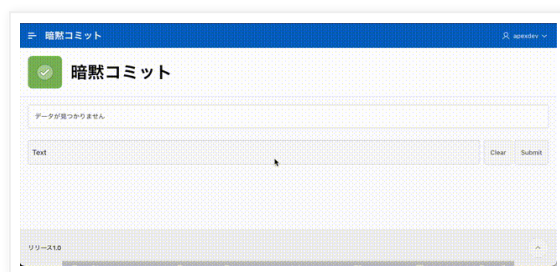
追記: APEX 23.1よりアプリケーション定義のプロパティにセッション・ステート・コミットが追加されました。以下はセッション・ステート・コミットが即時の動作になります。セッション・ステート・コミットがリクエストの終わりの場合は、暗黙コミットは発生しません。

APEX 23.1で追加されたセッション・ステート・コミットの注意点

Twitterを見ていたら、ページを送信したときにエラーが発生したけどトランザクションがロールバックしない、というツイートを見かけました。

このような事象は実際に発生します。プロセス中でページ・アイテムを操作すると、**暗黙でトランザクションがコミットされる**ことがあるためです。

テストのためのアプリケーションを作成しました。



今回はアプリケーションを作成することが目的ではないため、作り方については省略します。

以下のDDLで表TEST_MESSAGESを作成し、ページ・アイテムに記述した文字列を列MESSAGEに書き込んでいます。

表TEST_MESSAGESの内容は、クラシック・レポートで表示しています。

```
create table test_messages (  
  id          number generated by default on null as identity  
             constraint test_messages_id_pk primary key,  
  message     varchar2(4000 char)  
)  
;
```

ボタンSubmitをクリックすると、4つのプロセスを実行します。



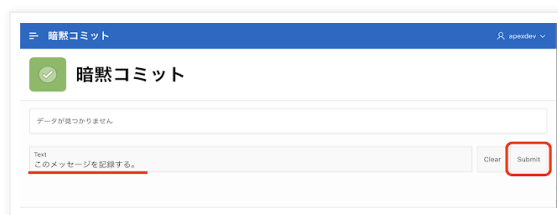
それぞれのプロセスが実行するのは、1行のコマンドです。上から順にコマンドを記載します。

プロセス**Update Page Item**にて、ページ・アイテム**P1_TEXT**に値として**処理完了**を設定します。

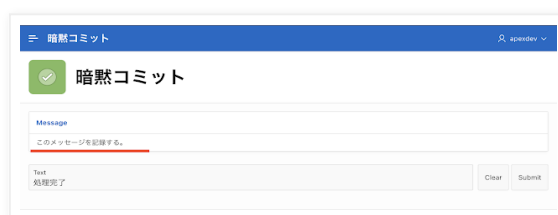
```
delete from test_messages;
insert into test_messages(message) values(:P1_TEXT);
:P1_TEXT := '処理完了';
rollback;
```

上記の記述方法でページ・アイテムに値を設定すると、この時点でトランザクションがコミットされます。より正確には、**すでにP1_TEXTに保持されている値と異なる値が設定されたときに、トランザクションがコミットされます。**

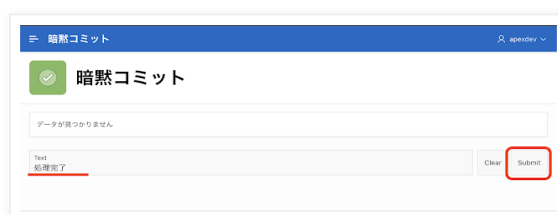
例えばページ・アイテム**P1_TEXT**に**このメッセージを記録する。**と書いてボタン**Submit**をクリックします。



rollbackは実行されますが、ページ・アイテム**P1_TEXT**に**処理完了**と書き込まれている時点でトランザクションはコミットされているため、レポートに**このメッセージを記録する。**と表示されます。



しかし、ページ・アイテム**P1_TEXT**に**処理完了**と書いて**Submit**をクリックしても、レポートには何も表示されません。



表TEST_MESSAGESには**処理完了**と書き込まれてはいますが、P1_TEXTの値に変更がないためトランザクションはコミットされません。そのため後続のrollbackで、表への書き込みもロールバックされます。

`apex_util.set_session_state`の第3引数（`p_commit`）にtrueを指定した場合も、同様の動作になります。

```
delete from test_messages;  
insert into test_messages(message) values(:P1_TEXT);  
apex_util.set_session_state('P1_TEXT','処理完了',true);  
rollback;
```

バインド変数に値を指定している場合は、同じプロセス内でrollbackを実行するとトランザクションのコミットはされません。`apex_util.set_session_state`の第3引数がtrueの場合は、同じプロセス内であってもファンクションが呼び出された時点でトランザクションがコミットされるため、その後rollbackを実行しても`apex_util.set_session_state`以前の処理はロールバックされません。

`apex_util.set_session_state`の第3引数がfalseの場合は、トランザクションはコミットされません。そのため、delete、insertの処理も含めてトランザクションがロールバックされます。

```
delete from test_messages;  
insert into test_messages(message) values(:P1_TEXT);  
apex_util.set_session_state('P1_TEXT','処理完了',false);  
rollback;
```

OracleのDan McGhanさんによるブログ記事に、より詳しい解説があります。

Implicit Commits in APEX

<http://danmcghan.blogspot.com/2012/08/implicit-commits-in-apex.html>

APEXの暗黙でコミットを実行するポイントを7つ挙げています。

1. ページのロード時、レンダリングが完了後
2. ページの送信時、ブランチが呼び出されて別ページに移る前
3. ページの送信時、検証（Validation）が失敗してページの再描画が行われる前
4. バインド変数の操作を含むプロセス（Process）の終了時
5. 計算（Computation）の終了時
6. APEX_UTIL.SET_SESSION_STATEが呼ばれたとき（この記事が書かれたときは引数p_commitはなく、常にコミット）
7. APEX_MAIL.PUSH_QUEUEが呼ばれた時

今回動作確認のために使用したAPEX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/test-implicit-commit.zip>

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

[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.
