

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6月15日 木曜日

APEX_SESSION.SET_TENANT_IDを活用する

最近、プロシージャAPEX_SESSION.SET_TENANT_IDを使ってAPEXアプリケーションをマルチテナントに対応させよう、という記事を読みました。Jon Dixonさんが彼のブログより公開しています。

Building Multi-Tenant APEX Apps

<https://blog.cloudnueva.com/multi-tenant-apex-apps>

イタリアのRoberto Capancioniさんも記事を書いています。

Multitenant App with APEX

<https://www.capancioni.com/multitenant-app-with-apex/>

気になったので、自分でも使ってみようと思います。

Oracle APEXはアプリケーション・コンテキストのネームスペースとしてAPEX\$SESSIONを用意しています。SYS_CONTEXT関数を呼び出して、ネーム・スペースAPEX\$SESSIONよりAPP_ID、APP_SESSION、APP_USERといった値を参照できます。

SYS_CONTEXT関数から参照できる、APEXの組み込み置換文字列については[マニュアルに記載](#)があります。アプリケーション・コンテキストとして置換文字列の値を参照することには、いくつかの利点があります。

1. V関数よりも高速である。V関数はPL/SQLによって実装されており、SELECT文にV関数が含まれていると、SQLとPL/SQLの実行エンジン間でのコンテキスト・スイッチが発生する。
2. バインド変数と異なり、ビューを定義するSELECT文に含めることができる。

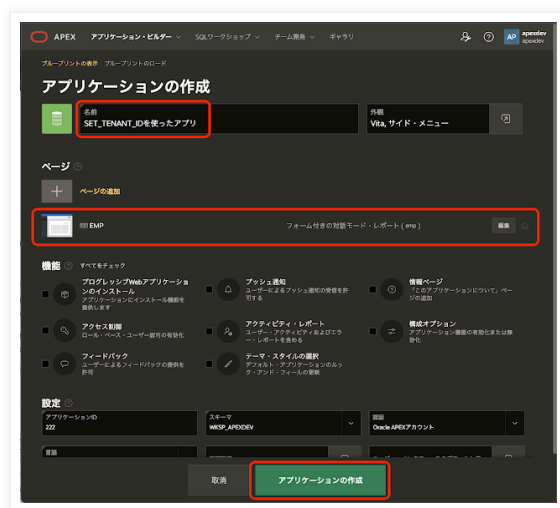
仮想プライベート・データベースにより同様の実装が可能ですが、その際は自前でネームスペースを作成する必要があります。APEX_SESSION.SET_TENANT_IDは、テナントの分離にパラメータを1つしか使わず、仮想プライベート・データベースまでは実装したくない、といった場合に手軽に活用できます。

簡単に言うと、APEX_SESSION.SET_TENANT_ID(p_tenant_id)を呼び出して設定した文字列は、SYS_CONTEXT('APEX\$SESSION','APP_TENANT_ID')から参照できます。

サンプル・データセットのEMP/DEPTに含まれる表EMPを使って、APEX_SESSION.SET_TENANT_IDを使ったアプリケーションを作ってみます。

テナントを分離するパラメータとして列DEPTNOを使用します。列ENAMEの名前でアプリケーションにサインインすると、所属部門が同じ従業員（列DEPTNOの値が同じ）に限り参照できるようにします。

アプリケーション作成ウィザードを起動し、表EMPをソースとした対話モード・レポートのページを追加します。



追加する対話モード・レポートのページ名はEMPとします。表またはビュー、対話モード・レポートを選択し、表またはビューとしてEMPを選択します。フォームを含めるにチェックを入れます。



以上の設定でアプリケーションを作成します。

APEX_SESSION.SET_TENANT_IDは、サインイン時に実行される**認証スキームの認証後のプロシージャ**内で呼び出します。サインインの処理が完了した後にAPEX_SESSION.SET_TENANT_IDを呼び出し、APP TENANT IDを変更することはできません。

表EMPの列ENAMEの値でサインインする**認証スキーム**を作成します。その認証スキームの**認証後のプロシージャ**よりAPEX_SESSION.SET_TENANT_IDを呼び出し、APP_TENANT_IDを設定します。

共有コンポーネントの認証スキームを開きます。

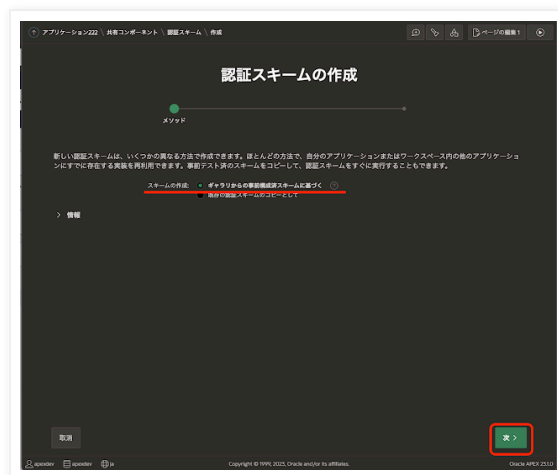


作成済みの**認証スキーム**が一覧されます。**作成**をクリックします。



スキームの作成はギャラリーからの**事前構成済スキームに基づく**を選択します。

次に進みます。



作成する認証スキームの**名前**は任意ですが、今回は**ENAMEでサインイン**としました。**スキーム・タイプ**として**公開資格証明**を選択します。これは主にテストに使用する、ユーザー名だけでサインインする認証スキームです。



認証スキームが作成され、カレントの認証スキームになります。

ソースのPL/SQLコードに以下を記述します。サインインしたユーザ名で表EMPを検索し、見つかった従業員の部門コード(列DEPTNOの値)をAPP_TENANT_IDに設定しています。

```

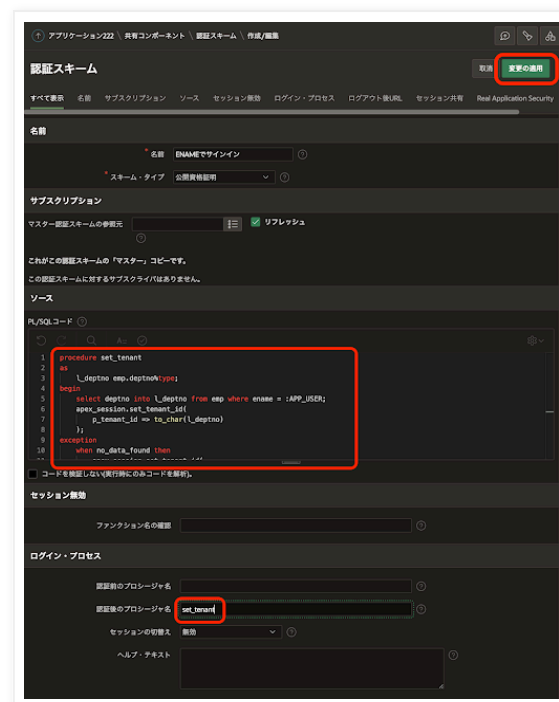
procedure set_tenant
as
    l_deptno emp.deptno%type;
begin
    select deptno into l_deptno from emp where ename = :APP_USER;
    apex_session.set_tenant_id(
        p_tenant_id => to_char(l_deptno)
    );
exception
    when no_data_found then
        null;
end;

```

set_tenant.sql hosted with ❤ by GitHub

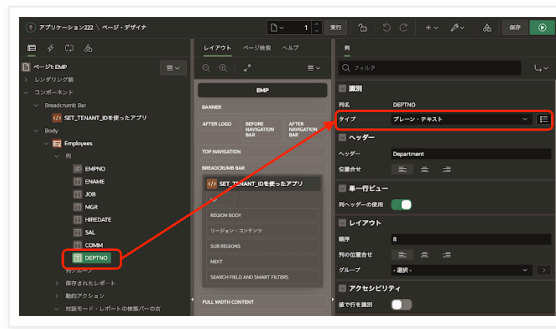
[view raw](#)

プロシージャset_tenantをログイン・プロセスの認証後のプロシージャ名に記述します。



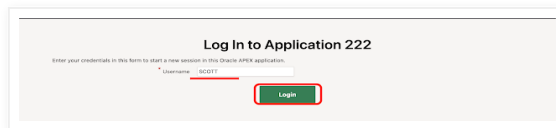
以上でアプリケーション・コンテキストAPP_TENANT_IDの設定は完了です。

対話モード・レポートの列DEPTNOが数値で表示されるよう、識別のタイプをプレーン・テキストに変更します。



アプリケーションを実行します。

公開資格証明のサインインが要求されるので、**SCOTT**でサインインします。



今の所、アプリケーション・コンテキストのAPP_TENANT_IDは設定されていますが、アプリケーション側に何も制限はかかっていません。そのため表EMPのすべての行が表示されます。

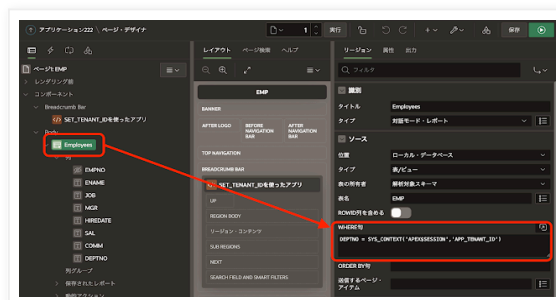
SCOTTのDEPTNOは40で、同じ部門の従業員はMARTINのみです。

Employee Name	Job	Manager	Hired	Salary	Commission	Department
ADAMS	CLERK	SCOTT	1983/01/12	1,100		20
ALLEN	SALESMAN	BLAKE	1981/02/09	1,600	300	30
BLAKE	MANAGER	KING	1981/05/01	2,850		30
CLARK	MANAGER	KING	1981/06/09	2,450		10
FORD	ANALYST	JONES	1981/12/03	3,000		20
JAMES	CLERK	BLAKE	1981/12/03	950		30
JONES	MANAGER	KING	1981/04/02	2,975		30
KING	PRESIDENT		1981/11/17	5,000		10
MARTIN	CLERK	BLAKE	1981/03/28	1,200	1400	40
MILLER	CLERK	CLARK	1982/07/09	1,300		10
SCOTT	PRESIDENT	JONES	1987/12/09	3,000		40
SMITH	CLERK	FORD	1980/12/07	800		20
TURNER	SALESMAN	BLAKE	1981/09/08	1,500	0	30
WARD	SALESMAN	BLAKE	1981/02/22	1,200	500	30

対話モード・レポートにAPP_TENANT_IDを使った制限を加えます。

ソースのWHERE句に以下の条件を記述します。

DEPTNO = SYS_CONTEXT('APEX\$SESSION','APP_TENANT_ID')



ページを実行すると、SCOTTとMARTINに表示が限定されることが確認できます。

SET_TENANT_IDを使ったアプリ

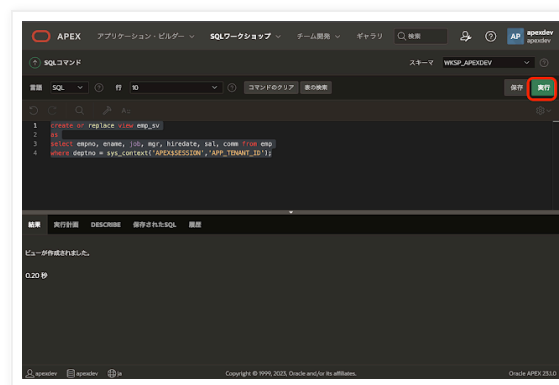
SET_TENANT_IDを使ったアプリ

Employee Name	Job	Manager	Hired	Salary	Commission	Department
MARTIN	CLERK	BLAKE	1963/09/28	1,200	1400	40
SCOTT	PRESIDENT	JONES	1962/07/09	5,000		40

APP_TENANT_IDを使って検索結果を制限することができましたが、表EMPがデータ・ソースに現れる度にWHERE句を追加するのは大変です。

表EMP（列DEPTNOを持つ表）を扱う際には、常にAPP_TENANT_IDで制限されるようにビューEMP_SVを作成します。そして表EMPの代わりにビューEMP_SVを指定することにより、アプリケーションのどこでも所属している部門のデータだけが扱えるように制限します。

```
create or replace view emp_sv
as
select empno, ename, job, mgr, hiredate, sal, comm from emp
where deptno = sys_context('APEX$SESSION', 'APP_TENANT_ID');
```

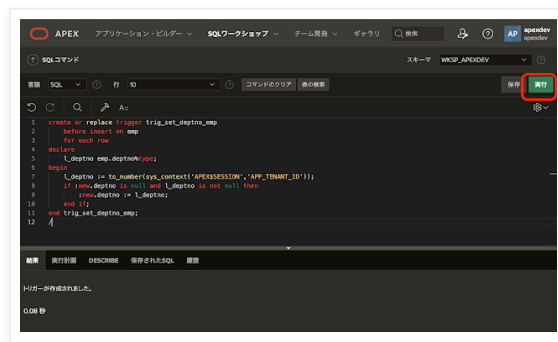


ビューEMP_SVは列DEPTNOを含んでいません。このビューに新規行を挿入すると、このままでは列DEPTNOが空白になります。列DEPTNOにAPP_TENANT_IDの値が設定されるよう、トリガーを作成します。

```
create or replace trigger trig_set_deptno_emp
before insert on emp
for each row
declare
l_deptno emp.deptno%type;
begin
l_deptno := to_number(sys_context('APEX$SESSION', 'APP_TENANT_ID'));
if :new.deptno is null and l_deptno is not null then
:new.deptno := l_deptno;
end if;
end trig_set_deptno_emp;
/
```

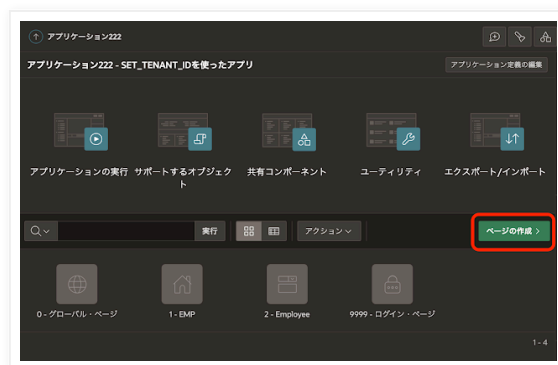
trig_set_deptno_emp.sql hosted with ❤ by GitHub

[view raw](#)



以上で準備は完了です。

ページ作成ウィザードを使って、ビューEMP_SVをデータ・ソースとした対話モード・レポートとフォームのページを作成します。



対話モード・レポートを選択します。



ページ定義のフォーム・ページを含めるにチェックを入れます。データ・ソースの表/ビューの名前としてEMP_SVを指定します。

次へ進みます。

主キー列 1 としてEMPNO(Number)を選択します。

ページの作成をクリックします。

以上の操作により、ビューEMP_SVをデータ・ソースとした対話モード・レポートとフォームのページが作成されます。

表EMPをデータ・ソースとした場合と、手順に違いはありません。

対話モード・レポートおよびフォームの操作からは、扱えるデータが制限されていることは分かりません。新規行を作成していますが、トリガーによってAPP_TENANT_IDが列DEPTNOに設定されているため、作成した行がレポートに表示されます。

Ename	Job	Mgr	Hiredate	Sal	Comm
SCOTT	PRESIDENT	7566	1982/12/16	4000	300
MARTIN	CLERK	7698	1981/09/28	1250	1400

APEXアプリケーションでのAPEX_SESSION.SET_TENANT_IDの活用は以上になります。

ビューEMP_SVに適用している条件がアクセス制限として適切なのかどうかを、APEXアプリケーションにアクセスして検証するのはあまり現実的ではありません。

APEXではコマンド・ラインの接続よりAPEXのセッションを作成するAPIとして、[APEX_SESSION.CREATE_SESSION](#)を提供しています。このAPIの引数p_call_post_authenticationにtrueを渡すことにより、今回APEX_SESSION.SET_TENANT_IDを呼び出すために記述した**認証後のプロシージャ**が実行されます。

これらの機能を使って、コマンド・ラインからビューEMP_SVを検索することができます。

```
SQL> begin
2 apex_session.create_session(
3 p_app_id => 222,
4 p_page_id => 1,
5 p_username => 'SCOTT',
6 p_call_post_authentication => true
7 );
8 end;
9* /
```

PL/SQLプロシージャが正常に完了しました。

```
SQL> select * from emp_sv;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7788	SCOTT	PRESIDENT	7566	82-12-16	4000	300
7654	MARTIN	CLERK	7698	81-09-28	1250	1400

```
SQL>
```

色々な条件のテスト・スクリプトを準備することにより、APP_TENANT_IDによるデータの分離が適切かどうかユーザー・インターフェースへアクセスせずに検証することができます。

本記事は以上になります。

今回作成した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/sample-set-tenant-id.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

Yuji N. 時刻: 14:33

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.