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8月16日月曜日

データベース・セキュリティの活用(7) - 仮想プライベート・データベース

テスト用に作成したAPEXアプリケーションは、意図的にSQLインジェクションに脆弱にしています。仮想プライベート・データベースを構成することにより、APEXアプリケーションへ変更を変更せずに（認証スキームは変更します）SQLインジェクションを防いでみます。

認証スキームの変更

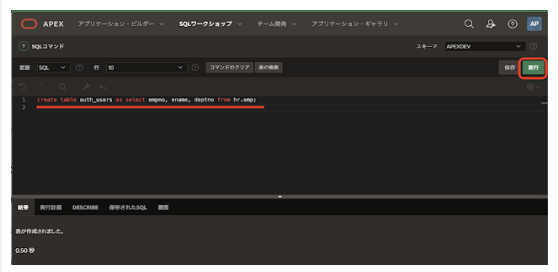
仮想プライベート・データベースによる表HR.EMPの保護を実装するにあたり、作成済みのAPEXアプリケーションを従業員によって認証するように変更します。

```
create table auth_users as select empno, ename, deptno from hr.emp;
```

seminar200825-create_auth_users.sql hosted with ❤ by GitHub　view raw

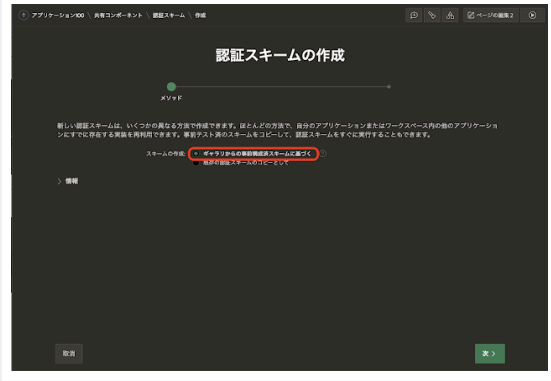
最初に認証に使用する表**AUTH_USERS**を表HR.USERSより作成します。

APEXの**SQLワークショップ**の**SQLコマンド**より実行します。



テスト用アプリケーションの**共有コンポーネント**の**認証スキーム**を開き、**作成**を実行します。

スキームの作成として、**ギャラリからの事前構成済スキームに基づく**を選択し、**次**に進みます。



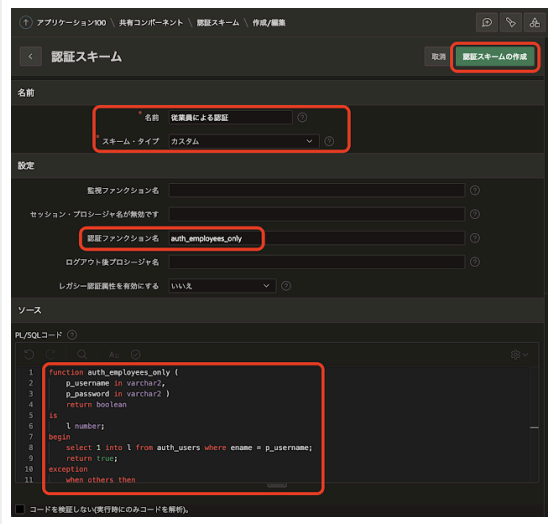
名前を**従業員による認証**とし、**スキーム・タイプ**として**カスタム**を選択します。**ソース**の**PL/SQLコード**に、下記のファンクションauth_employees_onlyを記述します。

```
function auth_employees_only (
    p_username in varchar2,
    p_password in varchar2 )
    return boolean
is
    l number;
begin
    select 1 into l from auth_users where ename = p_username;
    return true;
exception
    when others then
        return false;
end;
```

seminar200825-auth_employees_only.sql hosted with ❤️ by GitHub view raw

認証ファンクション名に**auth_employees_only**を指定します。ユーザー名が従業員名に一致しているとサインインに成功します。パスワードの検証は行いません。従業員名は大文字なので、ユーザー名も大文字で入力する必要があります。

認証スキームの作成をクリックします。



作成された認証スキーム**従業員による認証**は、作成後より**カレント・スキーム**になります。

アプリケーションを実行し、サインインの確認を行います。ユーザーAPEXDEVでサインインしたままの場合は、一旦サインアウトします。

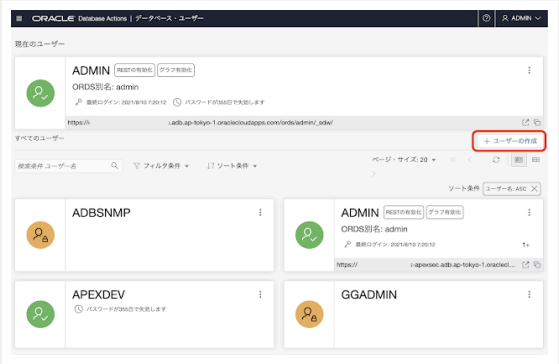
ユーザー名として大文字で**SCOTT**と入力し、**サインイン**をクリックします。パスワードは検証に使用されません。

左上のメニューを確認します。サインインしたユーザー名**scott**が小文字で表示されています。

以上で認証スキームは作成は完了です。

ユーザーVPDADMINの作成

仮想プライベート・データベースを構成するユーザー**VPDADMIN**を作成します。**データベース・アクション**より**データベース・ユーザー**を開きます。**ユーザーの作成**をクリックします。



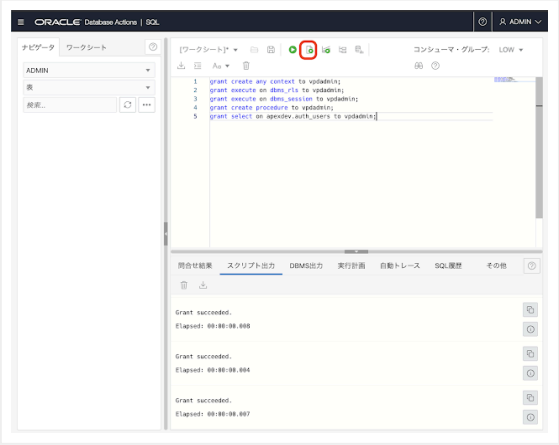
ユーザー名として**VPDADMIN**を指定します。**パスワード**を設定し、**Webアクセス**を**ON**に設定します。**ユーザーの作成**をクリックします。

ユーザーVPDADMINが作成されます。WebアクセスをONとしているため、データベース・アクションにVPDADMINとして接続することができます。

ユーザーVPDADMINが仮想プライベート・データベースを構成するために必要な権限を割り当てます。**データベース・アクション**の**SQL**を開き、以下を実行します。

```
grant create any context to vpdadmin;
grant execute on dbms_rls to vpdadmin;
grant execute on dbms_session to vpdadmin;
grant create procedure to vpdadmin;
grant select on apexdev.auth_users to vpdadmin;
```

seminar210825-vpdadmin-privs.sql hosted with ❤️ by GitHub view raw



ユーザーVPDADMINの作成は以上で完了です。データベース・アクションからサインアウトします。

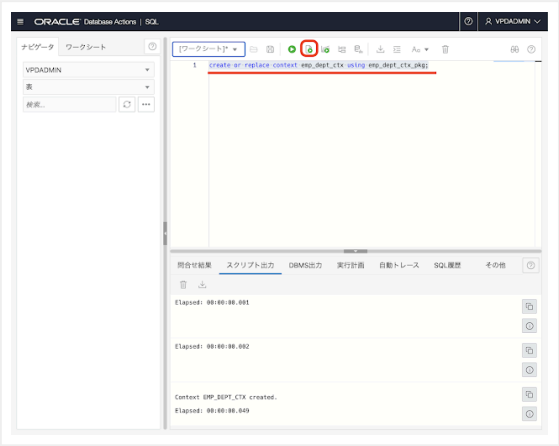
仮想プライベート・データベースの構成

仮想プライベート・データベースの構成作業を行います。**データベース・アクション**にユーザー**VPDADMIN**にてサインインします。**開発**の**SQL**を開きます。

最初にアプリケーション・コンテキスト**EMP_DEPT_CTX**を作成します。このアプリケーション・コンテキストを操作するパッケージは**EMP_DEPT_CTX_PKG**とします。

```
create or replace context emp_dept_ctx using emp_dept_ctx_pkg;
```

seminar210825-create_app_context.sql hosted with ❤ by GitHub view raw



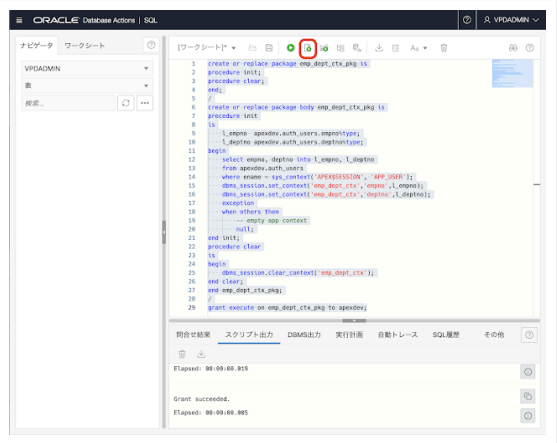
作成されたアプリケーション・コンテキストはビューDBA_CONTEXTより参照できます。(ビューの参照権限が必要です。)

続いてパッケージEMP_DEPT_CTX_PKGとそのパッケージ本体を作成します。次のスクリプトを実行します。アプリケーション・コンテキストにはサインインしたユーザーの従業員番号empnoと部門番号deptnoが保持されます。作成されたパッケージはユーザーAPEXDEVから実行できるよう、実行権限を与えます。

```
create or replace package emp_dept_ctx_pkg is
```

```
procedure init;
procedure clear;
end;
/
create or replace package body emp_dept_ctx_pkg is
procedure init
is
    l_empno  apexdev.auth_users.empno%type;
    l_deptno apexdev.auth_users.deptno%type;
begin
    select empno, deptno into l_empno, l_deptno
    from apexdev.auth_users
    where ename = sys_context('APEX$SESSION', 'APP_USER');
    dbms_session.set_context('emp_dept_ctx','empno',l_empno);
    dbms_session.set_context('emp_dept_ctx','deptno',l_deptno);
    exception
    when others then
        -- empty app context
        null;
end init;
procedure clear
is
begin
    dbms_session.clear_context('emp_dept_ctx');
end clear;
end emp_dept_ctx_pkg;
/
grant execute on emp_dept_ctx_pkg to apexdev;
```

seminar210825-create_context_package.sql hosted with ❤️ by GitHub view raw



仮想プライベート・データベースのポリシーを作成します。

最初に作成するポリシーは、表EMPの検索範囲をサインインしたユーザーと同じ部門に制限するポリシー**emp_in_same_deptno**です。

検索条件を生成するファンクション**pred_emp_in_same_deptno**を作成します。

```
CREATE OR REPLACE FUNCTION pred_emp_in_same_deptno
(
    schema_p IN VARCHAR2,
    table_p IN VARCHAR2
)
RETURN VARCHAR2
AS
    pred VARCHAR2(80);
BEGIN
    pred := q'~deptno = SYS_CONTEXT('emp_dept_ctx','deptno')~';
RETURN pred;
END;
/
```

seminar200825-pred_emp_in_same_deptno.sql hosted with ❤ by GitHub view raw

続けてポリシー**emp_in_same_deptno**を作成します。プロシージャDBMS_RLS.ADD_POLICYを呼び出します。

```
begin
    dbms_rls.add_policy(
        object_schema => 'hr'
      , object_name => 'emp'
      , policy_name => 'emp_in_same_deptno'
      , function_schema => 'vpdadmin'
      , policy_function => 'pred_emp_in_same_deptno'
      , statement_types => 'select'
      , policy_type => DBMS_RLS.CONTEXT_SENSITIVE
      , namespace => 'emp_dept_ctx'
      , attribute => 'deptno'
    );
end;
/
```

seminar200825-emp_in_same_deptno.sql hosted with ❤ by GitHub view raw

もうひとつ作成するポリシーは、自分自身がマネージャである従業員の列SALとCOMMのみ参照できるように制限するポリシー**emp_is_mgr**です。

条件を生成するファンクションは**pred_emp_is_mgr**です。ファンクションを作成した後にポリシー**emp_is_mgr**を作成します。

```
CREATE OR REPLACE FUNCTION pred_emp_is_mgr
(
    schema_p IN VARCHAR2,
    table_p IN VARCHAR2
```

```
)
RETURN VARCHAR2
AS
    pred VARCHAR2(80);
BEGIN
    pred := q'~mgr = SYS_CONTEXT('emp_dept_ctx','empno')~';
RETURN pred;
END;
/
begin
    dbms_rls.add_policy(
        object_schema => 'hr'
      , object_name => 'emp'
      , policy_name => 'emp_is_mgr'
      , function_schema => 'vpdadmin'
      , policy_function => 'pred_emp_is_mgr'
      , statement_types => 'select'
      , policy_type => DBMS_RLS.CONTEXT_SENSITIVE
      , sec_relevant_cols => 'sal,comm'
      , sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS
      , namespace => 'emp_dept_ctx'
      , attribute => 'empno'
    );
end;
/
```

seminar210825-emp_is_mgr.sql hosted with ❤️ by GitHub view raw

以上で表HR.EMPを保護するための設定が完了しました。

作成された仮想プライベート・データベースのポリシーはビューALL_POLICIESより参照できます。

APEXアプリケーションの変更

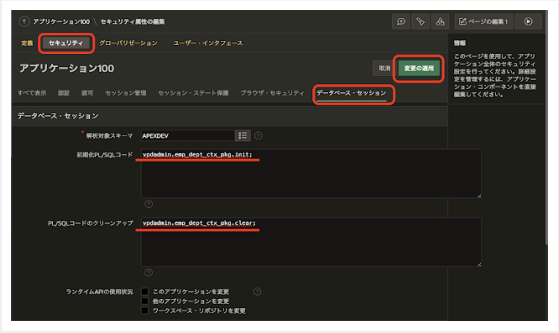
APEXアプリケーションにアプリケーション・コンテキストを操作する設定を追加します。**アプリケーション定義属性**の**セキュティ**の**データベース・セッション**の**初期化PL/SQLコード**として

vpdadmin.emp_dept_ctx_pkg.init;

PL/SQLコードのクリーンアップとして

vpdadmin.emp_dept_ctx_pkg.clear;

を指定します。



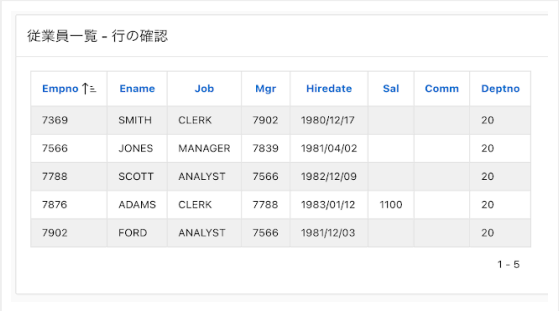
以上でアプリケーションの修正も完了です。

動作確認

アプリケーションにユーザーSCOTTでサインインし、従業員名として以下を指定してレポートを表示させます。

SCOTT' or '1' = '1

レポートのSQL問合せには変更がなくSQLインジェクションに脆弱なままですが、レポートに表示される従業員はSCOTTと同じ部門の従業員に限定されています。



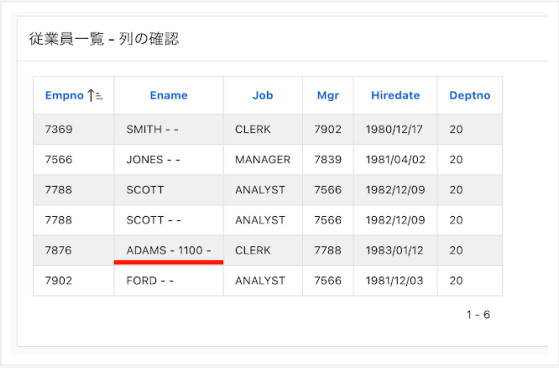
従業員一覧 - 行の確認

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	1980/12/17			20
7566	JONES	MANAGER	7839	1981/04/02			20
7788	SCOTT	ANALYST	7566	1982/12/09			20
7876	ADAMS	CLERK	7788	1983/01/12	1100		20
7902	FORD	ANALYST	7566	1981/12/03			20

1 - 5

続いて、以下を指定してレポートを表示します。

SCOTT' UNION SELECT EMPNO, ENAME || ' - ' || SAL || ' - ' || COMM ENAME, JOB, MGR, HIREDATE, DEPTNO FROM HR.EMP WHERE '1'='1



従業員一覧 - 列の確認

Empno	Ename	Job	Mgr	Hiredate	Deptno
7369	SMITH - -	CLERK	7902	1980/12/17	20
7566	JONES - -	MANAGER	7839	1981/04/02	20
7788	SCOTT	ANALYST	7566	1982/12/09	20
7788	SCOTT - -	ANALYST	7566	1982/12/09	20
7876	ADAMS - 1100 -	CLERK	7788	1983/01/12	20
7902	FORD - -	ANALYST	7566	1981/12/03	20

1 - 6

Enameに列SALの値が表示されていますが、SCOTTがマネージャーである従業員ADAMSに限定されています。

ポリシーの削除

後続の作業に影響があるため、作成した仮想プライベート・データベースのポリシーを削除します。

```
begin
    dbms_rls.drop_policy(
        object_schema => 'hr'
      , object_name => 'emp'
      , policy_name => 'emp_in_same_deptno'
    );
    dbms_rls.drop_policy(
        object_schema => 'hr'
      , object_name => 'emp'
      , policy_name => 'emp_is_mgr'
    );
end;
/
```

seminar210825-drop_vpd_policies.sql hosted with ❤️ by GitHub view raw

続く

Yuji N. 時刻: 17:50

共有

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