

日々は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。

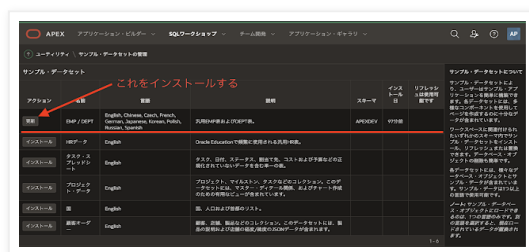
2021年8月27日 金曜日

表の操作ログを取得する

表の操作ログをトリガーで取得したい、と相談を受けました。ちょっと方法について考えてみました。

準備

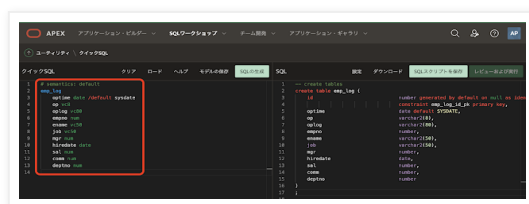
SQLワークショップのユーティリティのサンプル・データセットからEMP/DEPTをインストールします。表EMPを操作するアプリケーションを作成することにより、動作の確認を行います。



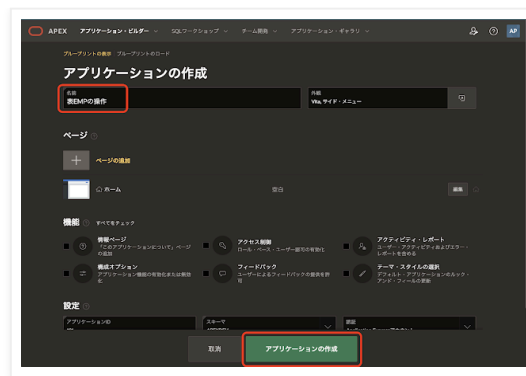
SQLワークショップのユーティリティのクイックSQLより表EMPの操作ログを保存する表EMP_LOGを作成します。クイックSQLの定義は以下になります。

```
# semantics: default
emp_log
  optime date /default sysdate
  op vc8
  oplog vc80
  empno num
  ename vc50
  job vc50
  mgr num
  hiredate date
  sal num
  comm num
  deptno num
```

SQLの生成、SQLスクリプトを保存、レビューおよび実行を順次実行して、表EMP_LOGを作成します。アプリケーションの作成は行いません。



テストに使用する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、空のアプリケーションを作成します。名前は表EMPの操作とします。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、フォームによる編集を行うページを作成します。**ページの作成**を実行します。



フォームを選択します。



フォーム付きレポートを選択します。



ページ名は任意です。ここでは**レポート・ページ名をレポートEMP、フォーム・ページ名をフォームEMP**としています。**フォーム・ページ・モードはモーダル・ダイアログ**にします。フォームのページに作成されるページ・アイテムの名前にはページ番号が含まれるので、**フォーム・ページ番号は3**にしてください。**次へ**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ます。次へ進みます。

データ・ソースの表/ビューの名前にEMP(表)を指定します。次へ進みます。

フォームが扱う主キー型として、データベースで管理(ROWID)を選択し、作成をクリックします。

以上で表EMPの対話モード・レポートと編集フォームのページが作成されました。

同様の手順で対話グリッドのページを作成します。ページの作成で編集可能対話グリッドを選択します。ページ名をグリッドEMPとし、レポート・ソースの表/ビューの名前にEMP(表)、主キー型としてROWIDを選択します。作成をクリックします。



これで対話グリッドのページも作成されました。

これから、作成したアプリケーションを使った表EMPの操作を、ログとして記録する実装を行なっていきます。

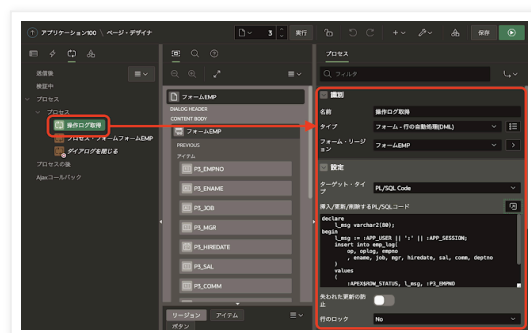
フォームの操作ログの記録

Oracle APEXのアプリケーションにプロセスを追加して、操作のログを取得します。

フォームのページ（ページ番号3番）をページ・デザイナーにて開きます。右ペインにプロセス・ビューを開き、**プロセスの作成**を実行します。作成するプロセスのコードは以下になります。

```
declare
    l_msg varchar2(80);
begin
    l_msg := :APP_USER || ':' || :APP_SESSION;
    insert into emp_log(
        op, oplog, empno
        , ename, job, mgr, hiredate, sal, comm, deptno
    )
    values
    (
        :APEX$ROW_STATUS, l_msg, :P3_EMPNO
        , :P3_ENAME, :P3_JOB, :P3_MGR, :P3_HIREDATE, :P3_SAL, :P3_COMM, :P3_DEPTNO
    );
end;
```

プロセスの**タイプ**として**フォーム - 行の自動処理(DML)**を選択します。**フォーム・リージョン**は**フォーム EMP**です。**設定のターゲット・タイプ**として**PL/SQL Code**を選択し、上記のコードを記載します。実際に編集している表の操作ではなく、別表に書き込むだけなので、（対象が元表である）**失われた更新の防止**は**OFF**、**行のロック**は**No**とします。



これでフォームを使った表EMPのINSERT、UPDATE、DELETEの操作が表EMP_LOGに記載されます。

テスト用アプリケーションを実行し表EMPの操作を行なった後、表EMP_LOGを確認してみます。SQLコマンドから以下のSQLを実行します。

```
select * from emp_log order by optime desc;
```



The screenshot shows the APEX SQL Workshop interface. The SQL command `select * from emp_log order by optime desc;` has been executed. The results are displayed in a table with the following columns: ID, OPTIME, OP, OPLG, EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO. The table contains three rows of data, all from the same user (APEXDEV) and session (80005685995361) at 21-08-27 15:00:00. The operations performed are insert (I), update (U), and delete (D).

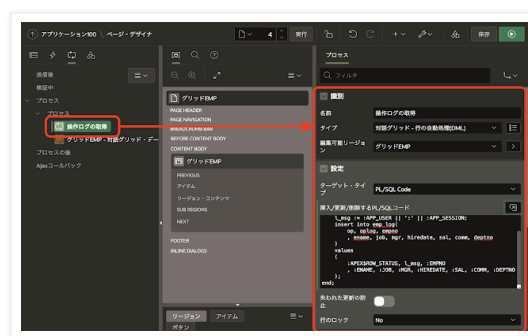
ID	OPTIME	OP	OPLG	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
27	21-08-27	D	APEXDEV:80005685995361	8000	YLIJ	MANAGER	7700	21-08-15	4000	40	40
26	21-08-27	U	APEXDEV:80005685995361	8000	YLIJ	MANAGER	7700	21-08-15	4000	40	40
25	21-08-27	I	APEXDEV:80005685995361	8000	YLIJ	MANAGER	7700	21-08-15	4000	40	20

対話グリッドの操作ログの記録

対話グリッドのページ（ページ番号4番）をページ・デザイナーにて開きます。右ペインにプロセス・ビューを開き、**プロセスの作成**を実行します。作成するプロセスのコードは以下になります。ほとんどフォームと同じです。

```
declare
    l_msg varchar2(80);
begin
    l_msg := :APP_USER || ':' || :APP_SESSION;
    insert into emp_log(
        op, oplog, empno
        , ename, job, mgr, hiredate, sal, comm, deptno
    )
    values
    (
        :APEX$ROW_STATUS, l_msg, :EMPNO
        , :ENAME, :JOB, :MGR, :HIREDATE, :SAL, :COMM, :DEPTNO
    );
end;
```

プロセスのプロパティの設定についても、フォームのときと同じです。



これで、対話グリッドの操作ログを取得するプロセスも作成できました。

テスト用アプリケーションから表EMPを操作し、その操作ログを確認します。

ID	OP	OPLOG	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
20	D	APEXDEV100055682953M	8000	YUJI	MANAGER	7788	21-08-21	4000	80	20
21	U	APEXDEV100055682953M	8000	YUJI	MANAGER	7788	21-08-19	4000	80	20
22	C	APEXDEV100055682953M	8000	YUJI	MANAGER	7788	21-08-01	4000	20	20
23	D	APEXDEV100055682953M	8000	YUJI	MANAGER	7788	21-08-15	4000	40	40
24	U	APEXDEV100055682953M	8000	YUJI	MANAGER	7788	21-08-15	4000	40	40
25	C	APEXDEV100055682953M	8000	YUJI	MANAGER	7788	21-08-15	4000	40	20

トリガーによる記録

APEXのアプリケーションではなく、データベース・トリガーを使って操作ログを取得してみます。

INSERTトリガーは以下になります。

```
create or replace trigger tgr_emp_ins
  before insert on emp
  for each row
declare
  l_msg varchar2(80);
begin
  l_msg := sys_context('APEX$SESSION','APP_USER') || ':' || sys_context('APEX$SESSION','APP_SESSION');
  insert into emp_log(
    op, oplog, empno
    , ename, job, mgr, hiredate, sal, comm, deptno
  )
  values
  (
    'INSERT', l_msg, :new.empno
    , :new.ename, :new.job, :new.mgr, :new.hiredate, :new.sal, :new.comm, :new.deptno
  );
end;
```

UPDATEトリガーです。

```
create or replace trigger tgr_emp_upd
  before update on emp
  for each row
declare
  l_msg varchar2(80);
begin
  l_msg := sys_context('APEX$SESSION','APP_USER') || ':' || sys_context('APEX$SESSION','APP_SESSION');
  insert into emp_log(
    op, oplog, empno
    , ename, job, mgr, hiredate, sal, comm, deptno
  )
  values
  (
    'UPDATE', l_msg, :new.empno
    , :new.ename, :new.job, :new.mgr, :new.hiredate, :new.sal, :new.comm, :new.deptno
  );
end;
```

DELETEトリガーです。

```
create or replace trigger tgr_emp_del
  before delete on emp
```

```

for each row
declare
l_msg varchar2(80);
begin
l_msg := sys_context('APEX$SESSION','APP_USER') || ':' || sys_context('APEX$SESSION','APP_SESSION');
insert into emp_log(
    op, oplog, empno
    , ename, job, mgr, hiredate, sal, comm, deptno
)
values
(
    'DELETE', l_msg, :old.empno
    , :old.ename, :old.job, :old.mgr, :old.hiredate, :old.sal, :old.comm, :old.deptno
);
end;
```

上記のトリガーを設定した後、テスト用アプリケーションから表EMPを操作してみます。

その後、表EMP_LOGの内容を確認します。トリガーのログでは操作をINSERT, UPDATE, DELETEとして記載しています。Oracle APEXのプロセスではC, U, Dとしています。1つの操作について、それぞれのログが記載されていることが確認できます。

ID	OPTIME	OP	OPLOG	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
53	21-08-27	D	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-12	4000	400	20
54	21-08-27	DELETE	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-12	4000	400	20
55	21-08-27	U	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-12	4000	400	20
54	21-08-27	UPDATE	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-12	4000	400	20
56	21-08-27	C	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-12	4000	200	20
57	21-08-27	INSERT	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-12	4000	200	20
58	21-08-27	D	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-09	4000	80	20
59	21-08-27	U	APEXDEV-XXXXXX5481953M	8000	YUJ	MANAGER	7788	21-08-09	4000	80	20

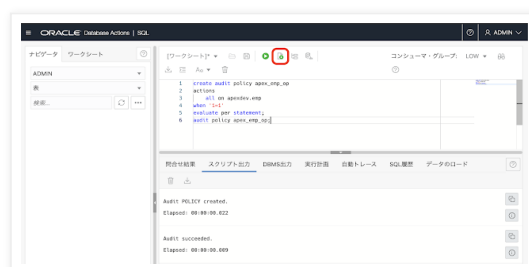
統合監査による記録

プロセスやトリガーを設定せずに、統合監査のポリシーを定義して表EMPで行われた操作を記録することができます。

ユーザーADMINにてデータベース・アクションに接続し、以下のSQLにて統合監査ポリシーapex_emp_opの作成と有効化を行います。

```

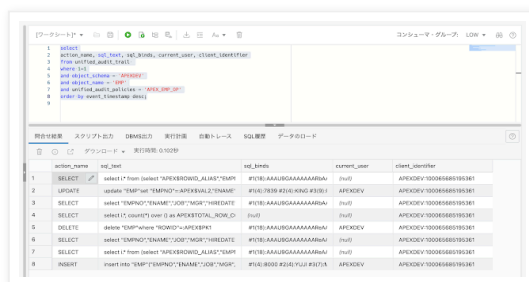
create audit policy apex_emp_op
actions
all on apexdev.emp
when '1=1'
evaluate per statement;
audit policy apex_emp_op;
```



これで表EMPへの操作が記録されます。APEXのプロセスやトリガーの実装とは異なり、SELECT文の実行も監査証跡の取得対象にすることが可能です。

テスト用のアプリケーションを使って表EMPの操作を行なった後、ビュー**UNIFIED_AUDIT_TRAIL**を検索し、監査証跡を確認します。以下のSELECT文を実行します。

```
select
action_name, sql_text, sql_binds, current_user, client_identifier
from unified_audit_trail
where 1=1
and object_schema = 'APEXDEV'
and object_name = 'EMP'
and unified_audit_policies = 'APEX_EMP_OP'
order by event_timestamp desc;
```



	action_name	sql_text	sql_binds	current_user	client_identifier
1	SELECT	select 1 from dual	#17E:AAAAAGAAAAAAABAA	#17E:AAAAAGAAAAAAABAA	APEXDEV
2	UPDATE	update EMP set EMPNO = APEXDEV.EMPNO	#17E:AAAAAGAAAAAAABAA	APEXDEV	APEXDEV
3	SELECT	select EMPNO, EMPNAME, JOB, MGR, HIREDATE	#17E:AAAAAGAAAAAAABAA	#17E:AAAAAGAAAAAAABAA	APEXDEV
4	SELECT	select 1 count(*) over (partition by EMPNO)	#17E:AAAAAGAAAAAAABAA	#17E:AAAAAGAAAAAAABAA	APEXDEV
5	DELETE	delete EMP where EMPNO = APEXDEV.EMPNO	#17E:AAAAAGAAAAAAABAA	APEXDEV	APEXDEV
6	SELECT	select EMPNO, EMPNAME, JOB, MGR, HIREDATE	#17E:AAAAAGAAAAAAABAA	#17E:AAAAAGAAAAAAABAA	APEXDEV
7	SELECT	select 1 from dual	#17E:AAAAAGAAAAAAABAA	#17E:AAAAAGAAAAAAABAA	APEXDEV
8	INSERT	insert into EMP (EMPNO, EMPNAME, JOB, MGR, HIREDATE)	#17E:AAAAAGAAAAAAABAA	APEXDEV	APEXDEV

表EMPの監査証跡が取得されていることが確認できます。

以上で表EMPの操作ログを取得する方法の紹介は終了です。

自律トランザクションについて

Oracle APEXのプロセスまたはトリガーによる操作ログの取得では、エラーが発生することにより操作のログが取得されない場合が起こり得ます。ログの書き込みを自律トランザクション（新しくトランザクションを開始し、ログを記載した時点で終了する）にて実行することにより、それぞれの障害を分離することができます。

Oracle APEXのプロセスとしての操作ログを取得する場合は、以下のようなログを表に書き込むプロシージャを作成し、そのプロシージャが自律トランザクションで動作するように記述します。APEXのページ・アイテムはデータを文字列として扱っていることを考慮して、引数の型をすべてVARCHAR2にしています。

```
create or replace procedure log_emp_op(
    p_row_status in varchar2
    , p_app_user in varchar2
    , p_app_session in varchar2
    , p_empno in varchar2
    , p_ename in varchar2
    , p_job in varchar2
    , p_mgr in varchar2
    , p_hiredate in varchar2
    , p_sal in varchar2
    , p_comm in varchar2
    , p_deptno in varchar2
)
is
    l_msg varchar2(80);
    pragma autonomous_transaction;
```



```

begin
    l_msg := p_app_user || ':' || p_app_session;
    insert into emp_log(
        op, oplog, empno
        , ename, job, mgr, hiredate, sal, comm, deptno
    )
    values
    (
        p_row_status, l_msg, p_empno
        , p_ename, p_job, p_mgr, p_hiredate, p_sal, p_comm, p_deptno
    );
    commit;
end log_emp_op;

```

操作ログを取得するプロセスのソースは以下に置き換わります。

```

log_emp_op(
    p_row_status => :APEX$ROW_STATUS
    , p_app_user => :APP_USER
    , p_app_session => :APP_SESSION
    , p_empno => :P3_EMPNO
    , p_ename => :P3_ENAME
    , p_job => :P3_JOB
    , p_mgr => :P3_MGR
    , p_hiredate => :P3_HIREDATE
    , p_sal => :P3_SAL
    , p_comm => :P3_COMM
    , p_deptno => :P3_DEPTNO
);

```

対話グリッドのプロセスのソースは以下になります。

```

log_emp_op(
    p_row_status => :APEX$ROW_STATUS
    , p_app_user => :APP_USER
    , p_app_session => :APP_SESSION
    , p_empno => :EMPNO
    , p_ename => :ENAME
    , p_job => :JOB
    , p_mgr => :MGR
    , p_hiredate => :HIREDATE
    , p_sal => :SAL
    , p_comm => :COMM
    , p_deptno => :DEPTNO
);

```

トリガーを自律トランザクションにするには、**pragma autonomous_transaction**の指定と**commit**文を追加します。INSERTトリガーの例です。他のトリガーも同様に変更します。

```

create or replace trigger tgr_emp_ins
    before insert on emp
    for each row
declare
    l_msg varchar2(80);
    pragma autonomous_transaction;
begin
    l_msg := sys_context('APEX$SESSION', 'APP_USER') || ':' || sys_context('APEX$SESSION', 'APP_SESSION');
    insert into emp_log(
        op, oplog, empno
        , ename, job, mgr, hiredate, sal, comm, deptno
    )

```

```
)  
values  
(  
    'INSERT', l_msg, :new.empno  
    , :new.ename, :new.job, :new.mgr, :new.hiredate, :new.sal, :new.comm, :new.deptno  
);  
commit;  
end;
```

今回作成したアプリケーションのエクスポートは以下に置きました。トリガーのソースもサポート・オブジェクトとして含んでいます。

<https://github.com/ujnak/apexapps/blob/master/exports/log-emp-op.sql>

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: 19:11

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.