

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31日 火曜日

表の操作ログを取得する - AUDIT_UTILを使ってみる

QuickSQLから表ディレクティブの/historyがサポートされなくなったのはなぜ？という以下のTweetに返信がありました。



こちらのスレッドでLiveSQLとQuickSQLを開発しているChristina Choさん(2018、2019年と来日されてセミナーをされているので、ご存知の方もいらっしゃるかも)からの返事があり、Connor McDonaldさんとOttmar Gobrechtさんが公開しているAUDIT_UTILの方がよりよいソリューションだから、とのことでした。



せっかくなので、AUDIT_UTILを試してみます。GitHubのリンクは[こちら](#)になります。

Autonomous Databaseへインストールする

パッケージAUDIT_UTILをAutonomous Databaseにインストールしてみます。sqlplusやSQLclで接続できないAPEXサービスに対してもインストールできるよう、データベース・アクションからインストールを行います。

GitHubよりaudit_util_setup.sql、audit_util_ps.sql、audit_util_pb.sqlをダウンロードします。データベース・ユーザーAUD_UTILを作成し、そのユーザーにパッケージAUDIT_UTILを作成します。

例えばAPEXのワークスペース・ユーザーとしてAPEXDEVがあり、そのユーザーにパッケージAUDIT_UTILを作成する場合はsame_schema以下にあるインストール・スクリプトを使用する、とのことでした。こちらのケースは試していません。READMEのSame Schema Supportのセクションを参照してください。

audit_util_setup.sqlは以下の行から始まります。Always FreeのAutonomous Databaseにインストールするので、tspaceにはdataを指定します。

```
define schema = aud_util
define tspace = data

pro
pro Have you set the SCHEMA and TSPACE variables before running this?
pro If yes, press Enter to continue, otherwise Ctrl-C to abort
pause
```

audit_util_setup.sqlは以下で終了します。Now we create the main packageのコメント以降でパッケージを作成するスクリプトを呼び出していますが、データベース・アクションからは呼び出すことができません。そのため、このコメント以降はスクリプトから除きます。

```
end;
/

--
-- Now we create the main package
--
set echo off
@@audit_util_ps.sql
@@audit_util_pb.sql

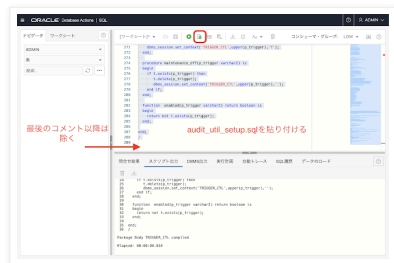
set lines 120
col object_name format a40

select object_name, object_type, status
from dba_objects
where owner = upper('&schema.');
```

今回はパーティショニング・オプションが使える前提ですが、ライセンスがない場合はaudit_util_setup.sqlの以下の部分をコメントアウトし、パッケージAUDIT_UTILの設定g_partitioningも変更する必要があります。

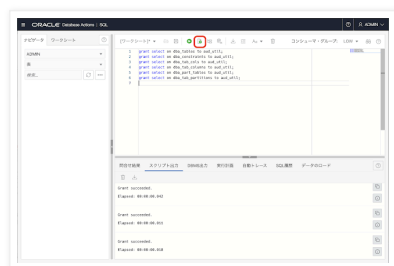
```
--
-- If you do not have a partitioning license, comment out these lines
-- and set the "g_partitioning" variable to false in the package body
--
partition by range ( aud$timestamp )
interval (numtoyminterval(1,'MONTH'))
( partition SYS_P000 values less than ( to_timestamp('20200101','yyyymmdd') )
)
pctfree 1 tablespace &tspace;
```

audit_util_setup.sqlの先頭行から最後のコメントの前までをクリップボードにコピーし、データベース・アクションのSQLに貼り付け、実行します。

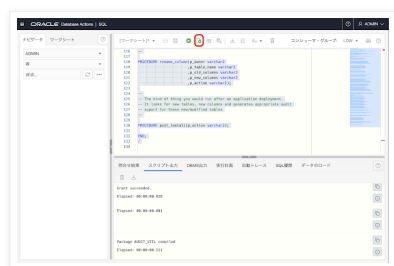


データベース・ユーザーAUD_UTILが作成されています。これからパッケージを作成しますが、その前にAUD_UTILに必要な権限を割り当てます。

```
grant select on dba_tables to aud_util;
grant select on dba_constraints to aud_util;
grant select on dba_tab_cols to aud_util;
grant select on dba_tab_columns to aud_util;
grant select on dba_part_tables to aud_util;
grant select on dba_tab_partitions to aud_util;
```



パッケージAUDIT_UTILの定義部を作成します。audit_util_ps.sqlの内容をデータベース・アクションのSQLに貼り付け実行します。



AUDIT_UTILのパッケージ本体を作成します。audit_util_pb.sqlの先頭行のtspaceの指定をdataに変更します。パッケージ本体の先頭には、AUDIT_UTILで設定可能なオプションが定数として定義されています。GitHubのページのComplete list of Settingsのセクションに説明があるので、そちらを参照した上で変更を加えます。今回はすべてデフォルトのままにします。

```
define schema = aud_util
define tspace = data

CREATE OR REPLACE PACKAGE BODY &&schema..AUDIT_UTIL
IS

    TYPE t_grantee_tab IS TABLE OF VARCHAR2(128);

    --
    -- global vars for new audit table support
    --

    g_aud_schema      constant varchar2(30) := upper('&&schema');
    g_aud_tspace       constant varchar2(30) := '&&tspace';

    -- prefix for all audit table names
```

```

--
g_aud_prefix      constant varchar2(10)  := '';

-- 1= dbms_output, 2=maint table, 3=both
--
g_log_level       constant int           := 3;

-- by default, just updates/deletes
--
g_inserts_audited constant boolean       := false;

--
-- even if inserts are off, do we keep the header
--
g_always_log_header constant boolean     := false;

-- sometimes an update is really a logical deleted. If you set a column
-- named as per below to 'Y', we'll audit it as a logical delete, not an update
--
g_logical_del_col constant varchar2(100) := 'DELETED_IND';

-- whether we want to capture OLD images for updates as well as NEW
--
g_capture_new_updates constant boolean := false;

-- if you want an automated scheduler job to look at auto-renaming
-- partitions, there is the name it gets
--
g_job_name        constant varchar2(80) := 'AUDIT_PARTITION_NAME_TIDY_UP';

--
-- where we should create the trigger (true=audit schema, false=table owning schema)
--
g_trigger_in_audit_schema constant boolean := true;

--
-- should we use partitioning
--
g_partitioning  constant boolean  := true;

--
-- should we use bulk binding (aka, are you expecting batch DML regularly)
--
g_bulk_bind constant boolean  := true;
g_bulk_bind_limit  int       := 500;

--
-- should we use a context/WHEN clause or a plsql call for trigger maintenance
--
g_use_context constant boolean := true;

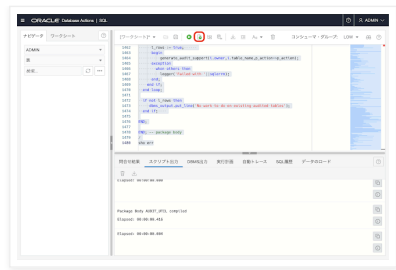
--
-- should we log CLOB/BLOB if unchanged in an update
--
g_audit_lob_on_update_always constant boolean := false;

--
-- NOTE: In terms of other naming conventions, check the routines
--       audit_table_name
--       audit_package_name
--       audit_trigger_name
--       They are the single points for controlling naming standard for each audit object type
--
--
--
-- Optional:
-- If you want the audit tables to be queryable from certain schemas/rolers, you can add them
-- here.
--
-- eg g_table_grantees t_grantee_tab := t_grantee_tab('SCOTT','HR','SYSTEM');
--
g_table_grantees t_grantee_tab := t_grantee_tab();

ex_non_existent_user exception;
pragma exception_init (ex_non_existent_user, -1917); -- non-existent user or role exception

```

audit_util_pb.sqlをデータベース・アクションのSQLにコピベし、実行します。



以上でAUDIT_UTILのインストールが完了しました。

audit_util_setup.sqlの最後に記述されている、インストール結果の確認SQLを実行します。

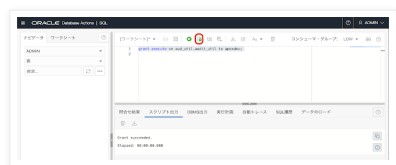
```
select object_name, object_type, status
from dba_objects
where owner = upper('aud_util');
```

以下の結果になります。

OBJECT_NAME	OBJECT_TYPE	STATUS
AUDIT_UTIL_UPDATE_TRIG	TABLE	VALID
AUDIT_UTIL_UPDATE_COLS_PK	INDEX	VALID
SCHEMA_LIST	TABLE	VALID
MAINT_SEQ	SEQUENCE	VALID
MAINT_LOG	TABLE	VALID
SEQ_AL_ID	SEQUENCE	VALID
AUDIT_HEADER	TABLE PARTITION	VALID
AUDIT_HEADER	TABLE	VALID
AUDIT_HEADER_PK	INDEX PARTITION	VALID
AUDIT_HEADER_PK	INDEX	VALID
AUDIT_PKG	PACKAGE	VALID
AUDIT_PKG	PACKAGE BODY	VALID
TRIGGER_CTL	PACKAGE	VALID
TRIGGER_CTL	PACKAGE BODY	VALID
AUDIT_UTIL	PACKAGE	VALID
AUDIT_UTIL	PACKAGE BODY	VALID

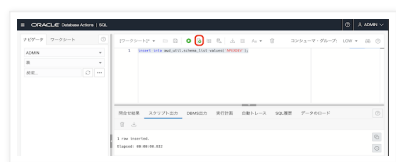
パッケージAUDIT_UTILをワークスペースAPEXDEVから呼び出せるよう、実行権限を与えます。

```
grant execute on aud_util.audit_util to apexdev;
```



AUDIT_UTILにて操作ログを取得するには、対象スキーマが表SCHEMA_LISTに登録されている必要があります。以下のINSERT文を実行します。

```
insert into aud_util.schema_list values('APEXDEV');
```

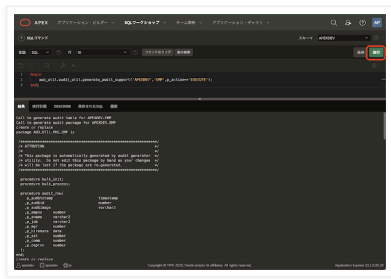


以上でAPEXのワークスペースAPEXDEVからAUDIT_UTILが使えるようになりました。

表EMPの操作ログを取得する

SQLワークショップのSQLコマンドより、AUDIT_UTIL.GENERATE_AUDIT_SUPPORTを実行します。引数のp_actionとしてEXECUTEを指定します。これで完了です。

```
begin
    aud_util.audit_util.generate_audit_support('APEXDEV','EMP',p_action=>'EXECUTE');
end;
```



結果に、実行されたSQLが出力されます。

```
Call to generate audit table for APEXDEV.EMP
create table AUD_UTIL.EMP (
  aud$tstamp    timestamp    not null,
  aud$id        number(18)   not null,
  aud$image     varchar2(3)  not null )
partition by range ( aud$tstamp )
interval (numtoyminterval(1,'MONTH'))
( partition EMP_p202109 values less than ( to_timestamp('20211001','yyyymmdd') )
) pctfree 1 tablespace data
alter table AUD_UTIL.EMP
add constraint EMP_PK primary key ( aud$tstamp, aud$id, aud$image)
using index
  (create unique index AUD_UTIL.EMP_PK
   on AUD_UTIL.EMP ( aud$tstamp, aud$id, aud$image)
   local tablespace data)
alter table AUD_UTIL.EMP add EMPNO NUMBER(4,0)
alter table AUD_UTIL.EMP add ENAME VARCHAR2(50)
alter table AUD_UTIL.EMP add JOB VARCHAR2(50)
alter table AUD_UTIL.EMP add MGR NUMBER(4,0)
alter table AUD_UTIL.EMP add HIREDATE DATE
alter table AUD_UTIL.EMP add SAL NUMBER(7,2)
alter table AUD_UTIL.EMP add COMM NUMBER(7,2)
alter table AUD_UTIL.EMP add DEPTNO NUMBER(2,0)
Call to generate audit package for APEXDEV.EMP
create or replace
package AUD_UTIL.PKG_EMP is
```

```

/*****
/* ATTENTION                               */
/*                                           */
/* This package is automatically generated by audit generator */
/* utility. Do not edit this package by hand as your changes */
/* will be lost if the package are re-generated.             */
*****/
```

```
procedure bulk_init;
procedure bulk_process;
```

```
procedure audit_row(
  p_aud$tstamp    timestamp
  ,p_aud$id        number
  ,p_aud$image     varchar2
  ,p_empno         number
  ,p_ename         varchar2
  ,p_job           varchar2
  ,p_mgr           number
  ,p_hiredate      date
  ,p_sal           number
  ,p_comm          number
  ,p_deptno       number
);
```

```
end;
create or replace
package body AUD_UTIL.PKG_EMP is
```

```

/*****
/* ATTENTION                               */
/*                                           */
/* This package is automatically generated by audit generator */
/* utility. Do not edit this package by hand as your changes */
/* will be lost if the package are re-generated.             */
*****/
```

```
type t_audit_rows is table of AUD_UTIL.EMP%rowtype
index by pls_integer;
```

```
l_audrows t_audit_rows;
```

```
procedure bulk_init is
begin
  l_audrows.delete;
end;
```

```
procedure bulk_process is
begin
```

```

forall i in 1 .. l_audrows.count
    insert into AUD_UTIL.EMP values l_audrows(i);
bulk_init;
end;

procedure audit_row(
    p_aud$tstamp      timestamp
    ,p_aud$id          number
    ,p_aud$image       varchar2
    ,p_empno           number
    ,p_ename           varchar2
    ,p_job             varchar2
    ,p_mgr             number
    ,p_hiredate         date
    ,p_sal             number
    ,p_comm            number
    ,p_deptno          number
) is
    l_idx pls_integer := l_audrows.count+1;
begin

    if l_idx > 500 then
        bulk_process;
        l_idx := 1;
    end if;

    l_audrows(l_idx).aud$tstamp := p_aud$tstamp;
    l_audrows(l_idx).aud$id      := p_aud$id;
    l_audrows(l_idx).aud$image   := p_aud$image;
    l_audrows(l_idx).empno       := p_empno;
    l_audrows(l_idx).ename       := p_ename;
    l_audrows(l_idx).job         := p_job;
    l_audrows(l_idx).mgr         := p_mgr;
    l_audrows(l_idx).hiredate    := p_hiredate;
    l_audrows(l_idx).sal         := p_sal;
    l_audrows(l_idx).comm        := p_comm;
    l_audrows(l_idx).deptno      := p_deptno;
end;

end;
grant execute on AUD_UTIL.PKG_EMP to APEXDEV
Call to generate audit trigger for APEXDEV.EMP
create or replace
trigger AUD_UTIL.AUD$EMP_APEXDEV
for insert or update or delete on APEXDEV.EMP
disable
when ( sys_context('TRIGGER_CTL','AUD$EMP_APEXDEV') is null)
compound trigger
l_dml   varchar2(1) := case when updating then 'U' when inserting then 'I' else 'D' end;
l_tstamp timestamp;
l_id    number;
l_descr  varchar2(100);

/*****
/* ATTENTION                               */
/*                                           */
/* This package is automatically generated by audit generator */
/* utility. Do not edit this package by hand as your changes */
/* will be lost if the package are re-generated.             */
*****/

before statement is
begin
    aud_util.audit_pkg.bulk_init;
    aud_util.pkg_emp.bulk_init;
end before statement;

after each row is
begin
    l_descr :=
        case
            when updating
            then 'UPDATE'
            when inserting
            then 'INSERT'
            else
            'DELETE'
        end;
end;

if updating or
deleting then
    aud_util.audit_pkg.log_header_bulk('EMP',l_dml,l_descr,l_tstamp,l_id);
end if;

if updating or
deleting then
    aud_util.pkg_emp.audit_row(
        p_aud$tstamp =>l_tstamp
        ,p_aud$id    =>l_id
        ,p_aud$image =>'OLD'

```

```
,p_empno =>:old.empno
,p_ename =>:old.ename
,p_job =>:old.job
,p_mgr =>:old.mgr
,p_hiredate =>:old.hiredate
,p_sal =>:old.sal
,p_comm =>:old.comm
,p_deptno =>:old.deptno
);
end if;
end after each row;

after statement is
begin
-- log the headers
aud_util.audit_pkg.bulk_process;
-- log the details
aud_util.pkg_emp.bulk_process;
end after statement;
end;
alter trigger AUD_UTIL.AUD$EMP_APEXDEV enable
```

文が処理されました。

操作ログを書き込む表AUD_UTIL.EMPを作成し、表AUD_UTIL.EMPにデータを書き込むパッケージAUD_UTIL.PKG_EMPを作成し、最後の表EMPにトリガーを作成します。READMEの

Creating an Audit Table、Creating an Audit Packageおよび**Creating an Audit Trigger**のセクションで説明されています。GitHub上のREADMEの説明では、引数**p_action**に**OUTPUT**を指定することでコードの出力のみ行なっています。

表の定義が変更された場合も同様に、**AUDIT_UTIL.GENERATE_AUDIT_SUPPORT**を実行します。差分を適用するスクリプトが生成され、既存の履歴表やパッケージに適用されます。このときの動作については、READMEの**Schema Evolution**のセクションで説明されています。

操作ログの取得を無効化するには**AUDIT_UTIL.DROP_AUDIT_SUPPORT**を呼び出します。

```
begin
aud_util.audit_util.drop_audit_support('APEXDEV','EMP',p_action=>'EXECUTE');
end;
```

上記の実行では履歴表AUD_UTIL.EMPはそのまま残ります。履歴表も削除するには引数**p_force**に**true**を与えます。

```
begin
aud_util.audit_util.drop_audit_support('APEXDEV','EMP',p_action=>'EXECUTE',p_force=>true);
end;
```

削除についてはREADMEの**Dropping Auditing for a table**のセクションで説明されています。

ログを参照する

操作ログを保存する表としてAUD_UTIL.EMPが作成されました。(READMEではEMP_SCOTTといった形で履歴表にスキーマ名が付いています。これは異なるスキーマに同じ名前の表があるときに自動的に付加されます)。

表AUD_UTIL.EMPを検索した結果は以下になります。

```
select * from aud_util.EMP;
```

AUD\$TSTAMP	AUD\$ID	AUD\$IMAGE	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
2021-08-31T06:32:44.468Z	1	OLD	8000	YUJI	MANAGER	7788	2021/8/11 0:00:00	4000	200	20
2021-08-31T06:32:48.602Z	2	OLD	8000	YUJI	MANAGER	7788	2021/8/11 0:00:00	4000	400	20

デフォルトでは変更前の値のみが保存されます（変更後の値は表EMPに残っているため）。ただし設定の**g_capture_new_updates**を**true**にすると、変更後の値も操作ログに残ります。

表AUD_UTIL.AUDIT_HEADERを参照すると、更新されたときのDMLの種類、セッション情報のACTION、CLIENT_ID、HOST、MODULE、OS_USERも参照できます。

```
select * from aud_util.AUDIT_HEADER;
```

AUD\$TSTAMP	AUD\$ID	TABLE_NAME	DML	DESCR	ACTION	CLIENT_ID	HOST	MODULE	OS_USER
2021-08-31T06:32:44.468Z	1	EMP	U	UPDATE	Processes - point: AFTER_SUBMIT	APEXDEV:113636404757626	ordswls-2	APEXDEV/APEX:APP 102:4	oracle
2021-08-31T06:32:48.602Z	2	EMP	D	DELETE	Processes - point: AFTER_SUBMIT	APEXDEV:113636404757626	ordswls-3	APEXDEV/APEX:APP 102:4	oracle

AUDIT_UTILの紹介は以上になります。

Oracle APEXのアプリケーション開発の参考にできれば幸いです。

Yuji N. 時刻: 16:04

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**.