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。

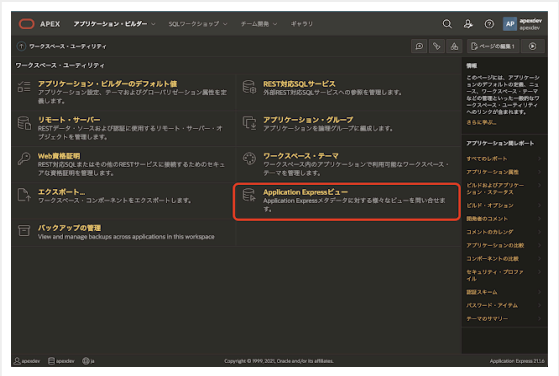
2022年1月8日土曜日

APEXのリポジトリの変更を確認する

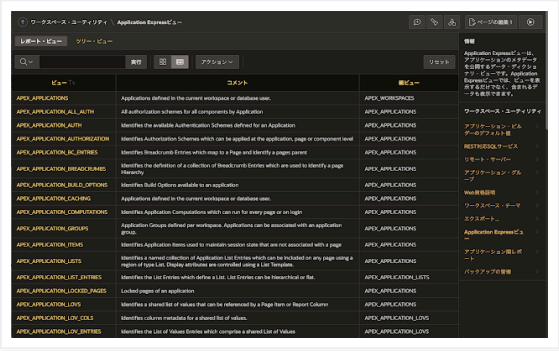
Oracle APEXのアプリケーションが変更されていないことを確認するには、どのような方法があるでしょう？と相談を受けたので、ちょっと考えてみました。

Oracle APEXのアプリケーションを構成しているページ、リージョン、ページ・アイテムなどはすべてAPEXアプリケーション・ビューを通して情報を得ることができます。

アプリケーション・ビルダーの**Application Expressビュー**から参照できます。



ビューからは、アプリケーションを構成している情報以外にも、稼働ログ、ワークスペース、登録されているユーザーやロールなども参照できます。



以下の手順で、APEXの変更を検知してみます。

1. APEXの標準ビューの内容を実表にコピーする。
2. コピーした実表とAPEXの標準ビューを比較する。

APEXの標準ビューをコピーするために、以下のPL/SQLスクリプトを書きました。

```
declare
    -- APEXのバージョンは21.1を想定している。
```

```
    C_APEX constant varchar2(12) := 'APEX_210100';
    -- スナップショットとなる表はAPEXの部分をB001に変更する。
    C_SNAPID constant varchar2(4) := 'B001';
    l_sql varchar2(4000);
    /*
     * ビューのカラムにnull component_typeのような定義が含まれている場合があり、データ型が不明になる。
     * CTASの実行でORA-1723が発生するため、そのようなカラムをスナップショットの取得から除く。
     */
    function get_column_list(
        p_view_name in varchar2
    ) return varchar2
    as
        l_column_list varchar2(4000);
    begin
        for c in (
            select column_name from all_tab_cols
            where 1=1
            and owner = C_APEX
            and table_name = p_view_name
            and not (data_type = 'VARCHAR2' and data_length = 0)
            order by column_id
        )
        loop
            if lengthb(l_column_list) > 0 then
                l_column_list := l_column_list || ',';
            end if;
            l_column_list := l_column_list || c.column_name;
        end loop;
        return l_column_list;
    end;
begin
    for t in (
        -- APEXの公開ビューはAPEXで始まるパブリック・シノニムが定義されている。
        select synonym_name, table_name from all_synonyms
        where 1=1
        and
            owner = 'PUBLIC'
        and
            table_owner = C_APEX
        and
            synonym_name like 'APEX%'
        and
            table_name in
            (
                -- APEXスキーマでアクセス可能な表またはビューに限定する。
                select table_name from all_tab_privs
                where 1=1
```

```
                and grantee ='PUBLIC'
                and privilege in ('SELECT','READ')
                and grantable = 'NO'
                and table_schema = C_APEX
                and type in ('VIEW','TABLE')
            )
        order by synonym_name
    )
    loop
        /* 比較不要のビューを排除する。
         * もっと排除できるビューはある - ログなど - が、以降の処理でエラーが発生するビューだけを除外している。
         */
        if t.synonym_name in (
          'APEX_WORKSPACE_ACTIVITY_LOG',
          'APEX_SYS_ALL_CONSTRAINTS',
          'APEX_APPL_EXPORT_COMPS',
          'APEX_COLLECTIONS',
          'APEX_SYS_ALL_OBJECTS', -- 実体はALL_OBJECTS
          'APEX_SYS_ALL_DEPENDENCIES', -- 実体はALL_DEPENDENCIES
          'APEX_SCHEDULER_JOBS', -- ALL_SCHEDULER_JOBS
          'APEX_DG_BUILTIN_COMPANY_NAMES' -- 何故か変更が通知される - 不要
        ) then
            continue;
        end if;
        -- CTAS文の生成。
        l_sql := 'create table ' || C_SNAPID || substr(t.synonym_name,5) || ' as select '
            || get_column_list(t.table_name) || ' from ' || t.synonym_name;
        begin
            execute immediate l_sql;
        exception
            when others then
                dbms_output.put_line(sqlerrm);
                dbms_output.put_line(l_sql);
        end;
    end loop;
end;
```

take_apex_snapshot.sql hosted with ❤️ by GitHub view raw

実行すると、APEX_ で始まるビューを、B001_で始まる表にコピーします。CTAS(CREATE TABLE AS SELECT)を実行しています。

続いて作成した表とAPEX標準ビューを比較します。以下のPL/SQLスクリプトを書きました。

```
declare
    C_SNAPID constant varchar2(5) := 'B001%';
    l_sql1 varchar2(32767);
    l_sql2 varchar2(32767);
```

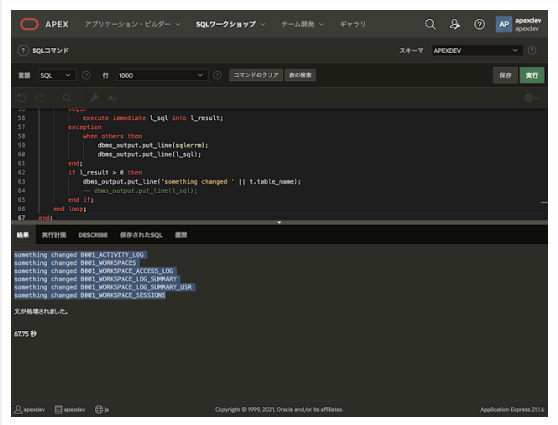
```
    l_sql  varchar2(32767);
    l_column_list varchar2(32767);
    l_result number;
begin
    -- スナップショットとして作成されている表を比較対象とする。
    for t in (select table_name from user_tables where table_name like C_SNAPID)
    loop
        l_column_list := '';
        for c in
        (
            select column_name, data_type from user_tab_cols
            where table_name = t.table_name order by column_id
        )
        loop
            if lengthb(l_column_list) > 0 then
                l_column_list := l_column_list || ',';
            end if;
            -- 比較はハッシュをとって行う。
            if c.data_type in ('VARCHAR2') then
                l_column_list := l_column_list || 'ora_hash(' || c.column_name || ')';
            elsif c.data_type in ('CLOB','NCLOB','BLOB') then
                -- 4000バイトを超えるとora_hashは使えない。重い処理だが、SHA256(引数4)のハッシュを取る。
                l_column_list := l_column_list || 'case when ' || c.column_name || ' is null th
            else
                -- それ以外は、生データで比較する。
                l_column_list := l_column_list || c.column_name;
            end if;
        end loop;
        /*
         * 以下のSQLを比較のために生成し、実行する。count(*)が0であれば、表は完全に一致。
         *
           select count(*) from
           (
             (
                 select ora_hash(col1), case when col2 is null then null else dbms_crypto.hash(
                 from snapshot_table
                 minus
                 select ora_hash(col1), case when col2 is null then null else dbms_crypto.hash(
                 from apex_view
             )
             union all
             (
                 select ora_hash(col1), case when col2 is null then null else dbms_crypto.hash(
                 from apex_view
                 minus
                 select ora_hash(col1), case when col2 is null then null else dbms_crypto.hash(
                 from snapshot_table
```

```
            )
        )
     */
    l_sql1 := 'select ' || l_column_list || ' from ' || t.table_name;
    l_sql2 := 'select ' || l_column_list || ' from ' || 'APEX' || substr(t.table_name,5);
    l_sql := 'select count(*) from ((' || l_sql1 || ' minus ' || l_sql2 || ') union all ('
    -- dbms_output.put_line(l_sql);
    begin
        execute immediate l_sql into l_result;
    exception
        when others then
            dbms_output.put_line(sqlerrm);
            dbms_output.put_line(l_sql);
    end;
    if l_result > 0 then
        dbms_output.put_line('something changed ' || t.table_name);
        -- dbms_output.put_line(l_sql);
    end if;
  end loop;
end;
```

compare_apex_view_and_snapshot.sql hosted with ❤️ by GitHub view raw

APEXのアプリケーションは何も変更せずに上記のスクリプトを実行すると、以下の結果が返りました。

```
something changed B001_ACTIVITY_LOG
something changed B001_WORKSPACES
something changed B001_WORKSPACE_ACCESS_LOG
something changed B001_WORKSPACE_LOG_SUMMARY
something changed B001_WORKSPACE_LOG_SUMMARY_USR
something changed B001_WORKSPACE_SESSIONS
```



変更が検知されたのはログが書き込まれたり、利用状況によって更新される値を含んでいるビューです。変更の検知を確認するために、APEX標準ビューから表へコピーする対象に含めていましたが本来は対象から外すべきです。

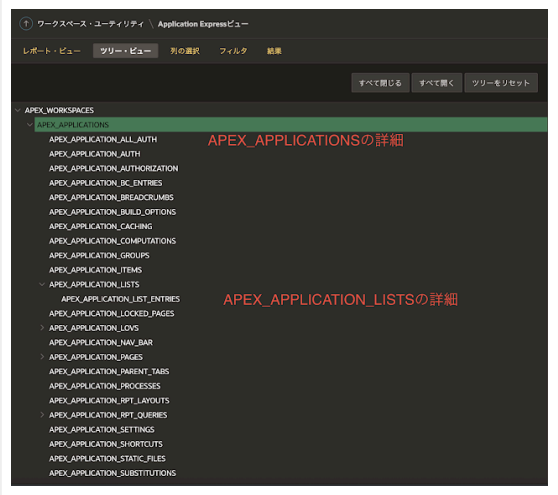
このままでは結構な負荷になるため、ログ以外にも色々なビューを比較の対象から外す必要はあります。とはいえ、APEXアプリケーションの変更を検知する実装としての方向性は正しいと思います。データの変更を検知できているためです。

作成した表を削除するために、以下のスクリプトを書きました。

```
declare
    -- 削除するスナップショット表のプレフィックス。
    C_SNAPID constant varchar2(5) := 'B001%';
begin
   for t in (select table_name from user_tables where table_name like C_SNAPID)
   loop
       execute immediate 'drop table ' || t.table_name || ' purge';
    end loop;
end;
```

drop_apex_snapshot.sql hosted with ❤️ by GitHub view raw

また、ビュー**APEX_DICTIONARY**よりビューの親子関係を確認できます。Application Expressビューのツリー・ビューの元になっている情報です。



上記の階層構造を検索しているSQLを簡略化すると、以下になります。

```
with data as (
select
    APEX_VIEW_NAME as value,
    PARENT_VIEW as parent
    from APEX_DICTIONARY
    where column_id = 0
)
select case when connect_by_isleaf = 1 then 0
            when level = 1             then 1
            else                            -1
       end as status,
```

```
       level,
      value as value
from data
start with value = 'APEX_WORKSPACES'
connect by prior value = parent
order siblings by value
```

apex_dictionary_tree.sql hosted with ❤️ by GitHub view raw

親となっているビューAPEX_APPLICATIONSへの行の追加、削除（更新は除く）については、APEX_APPLICATIONSを親としているビューに含まれている詳細情報を確認する必要はありません。それらはアプリケーションが追加されている場合は、詳細情報もすべて追加になります。アプリケーションが削除された場合は詳細情報もすべて削除されています。

データベースのディクショナリ、ビューAPEX_DICTIONARY共に、どれがユニークなカラムを示す情報がありません。そのため、挿入、削除、更新があったことは検出できますが、挿入、削除、更新のどの操作があったのか、特に更新の検出が困難です。

それぞれのビューのユニークな列を示す情報を別に用意する必要があります。

Oracle APEXを利用する際の参考になれば幸いです。

完

Yuji N. 時刻: 19:30

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