

# 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。

2021年2月2日 火曜日

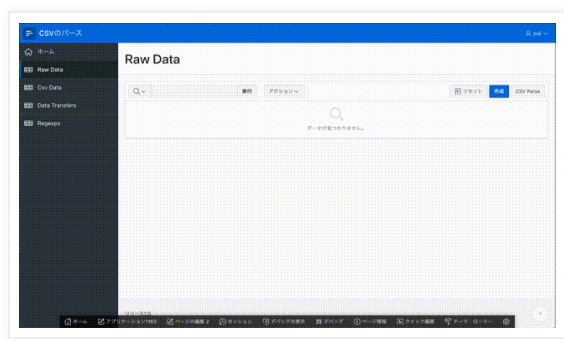
## CSVに含まれる任意の文字列のパーズ

CSVに含まれる一行の中に、説明書きとして以下のようながあります。

1.20 from tokyo to london

これをパーズしてデータベースに保存したいんだけど、と同僚より相談を受けました。JavaScriptだとできるんだけど、PL/SQLは書いたことがない、とのことでした。

それだと、何から始めて良いのかわからないので、ざくっとOracle APEXでサンプルを作って、こうやるのだよ、と伝えました。その作業記録です。



まず、スキーマをクイックSQLで定義しました。

```
# prefix: szx
# semantics: default
raw_data
  body file

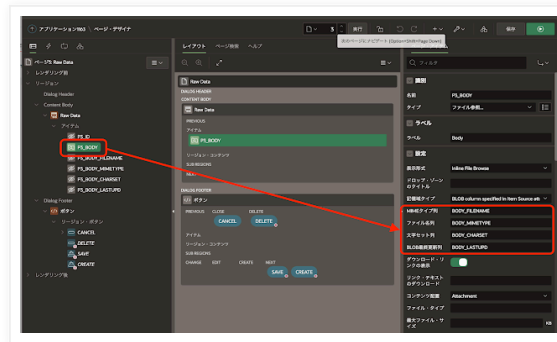
csv_data
  raw_data_id /fk raw_data
  category vc4000
  description vc4000

data_transfers
  csv_data_id /fk csv_data
  fare num
  from_place vc255
  to_place vc255

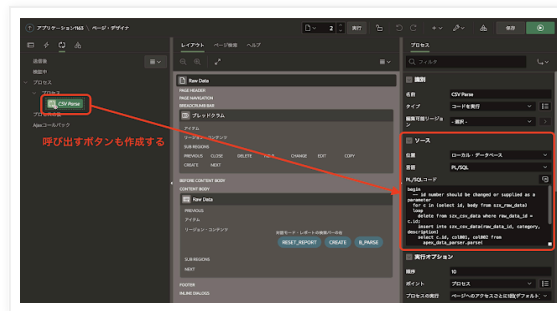
regexps
  category vc80
  regexp_str vc4000
  insert_str vc4000
  var_num num
  seq num
```

CSVファイルはSZX\_ROW\_DATA表のBLOBにアップロードします。このクイックSQLの定義から、特別な指定をせずにスキーマの生成からアプリケーションの作成までを行います。

CSVファイルをアップロードする際に、ファイル名やファイルのタイプなどもデータベースにきちんと登録させるために、フォームに含まれるBLOBのページ・アイテムに、MIMETYPE列、ファイル名列、文字セット列、BLOB最終更新列を設定します。



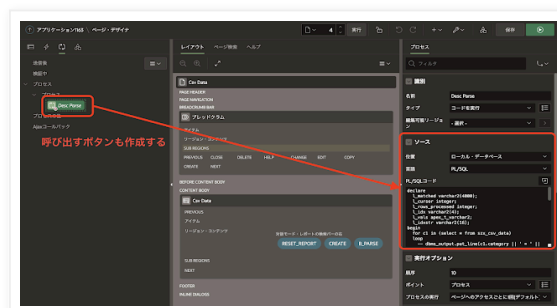
CSVファイルをアップロードしたのちに、カンマで分割する処理をアップロードしたファイルの一覧ページに追加します。



PL/SQLコードでは、APEX\_DATA\_PARSER.PARSEを使用しています。

```
begin
-- id number should be changed or supplied as a parameter
for c in (select id, body from szx_raw_data)
loop
delete from szx_csv_data where raw_data_id = c.id;
insert into szx_csv_data(raw_data_id, category, description)
select c.id, col001, col002 from
apex_data_parser.parse(
p_content => c.body,
p_file_name => 'data.csv'
);
end loop;
end;
```

次にDESCRIPTIONの列をパースする処理を、CSVファイルのパース結果の一覧画面に追加します。



PL/SQLコードでは、regexp\_replaceを使用しています。

```
declare
l_matched varchar2(4000);
l_cursor integer;
l_rows_processed integer;
l_idx varchar2(4);
l_vals apex_t_varchar2;
l_idxstr varchar2(16);
begin
for c1 in (select * from szx_csv_data)
loop
-- dbms_output.put_line(c1.category || ' = ' || c1.description);
for c2 in (select * from szx_regexps where category = c1.category order by seq)
loop
l_idxstr := '';
for i in 1..c2.var_num
loop
l_idxstr := l_idxstr || '\' || i || ':';
end loop;
-- dbms_output.put_line(c2.regexp_str || ' = ' || c2.insert_str || ' = ' || l_idxstr);
l_matched := regexp_replace(c1.description, c2.regexp_str, l_idxstr);
if l_matched <> c1.description then
-- dbms_output.put_line('MATCHED!!!!');
l_cursor := dbms_sql.open_cursor;
dbms_sql.parse(l_cursor, c2.insert_str, dbms_sql.native);
dbms_sql.bind_variable(l_cursor, ':1', c1.id);
l_vals := apex_string.split(l_matched, ':');
for i in 2..c2.var_num
loop
l_idx := ':' || i;
dbms_sql.bind_variable(l_cursor, l_idx, l_vals(i-1));
end loop;
l_rows_processed := dbms_sql.execute(l_cursor);
dbms_sql.close_cursor(l_cursor);
end if;
end loop;
end loop;
commit;
end;
```

正規表現と、それが一致した時に実行するINSERT文を表に登録します。



Category	Insert Str
fare	insert into szx_data_transfers(csv_data_id, fare, from_place, to_place) values(:1, :2, :3, :4)
fare	insert into szx_data_transfers(csv_data_id, fare, from_place, to_place) values(:1, :2, :3, :4)

登録する正規表現その 1

カテゴリ: fare

正規表現: ([0-9\.]+)\s+from\s+([A-Za-z ]+)\s+to\s+([A-Za-z ]+)

INSERT文: insert into szx\_data\_transfers(csv\_data\_id, fare, from\_place, to\_place) values(:1, :2, :3, :4)

バインド変数の数: 4

登録する正規表現その 2

カテゴリ: fare

正規表現: ([0-9\.]+)\s+to\s+([A-Za-z ]+)\s+from\s+([A-Za-z ]+)

INSERT文: insert into szx\_data\_transfers(csv\_data\_id, fare, from\_place, to\_place) values(:1, :2, :4, :3)

バインド変数の数: 4

以上です。テストには以下のCSVファイルをしました。

```
fare, 1.200      from tokyo  to  london
fare, 3200.00    from beijing to tokyo
fare, 4500.00    to  london from new york
fare, 900.00     to  new york from beijing
```

アプリケーションのエクスポートを、こちらに置いています。

<https://github.com/ujnak/apexapps/blob/master/exports/csvparser.sql>

完

Yuji N. 時刻: 14:11

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](#).