

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。

2020年4月9日 木曜日

APEXでOCIオブジェクト・ストレージ上のデータセットを使用する

OCIオブジェクト・ストレージ上のデータセット(CSVファイル)より、ファセット検索ページを作成するという[ブログエントリ](#)がありました。こういう要望は多いのかな、と思ったので自分でも実践してみます。元のブログ・エントリではブラジルにてオープン・データとして提供されている、労働災害と労働者の年齢区分のデータを使用しています。この記事は、その代わりに、日本で提供されているオープン・データを使用します。

使用するオープン・データ

東京都によって提供されている[都内の最新感染動向](#)のサイトより、

- 東京都_新型コロナウイルス陽性患者発表詳細 - [ここからダウンロード](#)
- 東京都_新型コロナコールセンター相談件数 - [ここからダウンロード](#)
- 東京都_新型コロナ受診相談窓口相談件数 - [ここからダウンロード](#)

CSVファイルをオブジェクト・ストレージへ配置する

まず、オブジェクト・ストレージにバケットを作成します。そのあと、取得した3つのCSVファイルをアップロードします。

Oracle Cloudのコンソールより、**オブジェクト・ストレージ**を開きます。



バケットの作成をクリックします。どのコンパートメントに作成してもかまいません。



バケット名を**opendata**に設定し、**バケットの生成**をクリックします。



opendataというバケットが作成されたことを確認します。バケット名**opendata**をクリックし、バケットを開きます。



オブジェクトのアップロードをクリックし、アップロードするファイルを指定するフォームを開きます。



ファイルを選択をクリックし、ダウンロード済みのファイルを選択します。3つのファイルすべてを選択します。すべて選択した後、オブジェクトのアップロードをクリックします。



CSVファイルがアップロードされていることを確認します。



DBMS_CLOUDパッケージを使用するための準備

DBMS_CLOUDパッケージを使用するために、以下の4つの準備を行います。

1. DBMS_CLOUDパッケージの実行権限をOracle APEXに与えます。
2. DATA_PUMP_DIRの読み書き許可をOracle APEXに与えます。
3. オブジェクト・ストレージにアクセスするための認証トークンを生成します。
4. アップロードしたCSVファイルへのURLパスを取得します。

DBMS_CLOUDパッケージの実行権限、DATA_PUMP_DIRの読み書き許可をOracle APEXに与える

SQL Developer Webから管理者アカウントでログインし、GRANT文を実行するのが手順として容易でしょう。

Oracle APEXがホストされているAutonomous Databaseのサービス・コンソールの開発から、SQL Developer Webを開きます。



ユーザー名としてADMIN、パスワードはデータベース・インスタンスの管理者パスワードを入力します。(管理者パスワードはインスタンス作成時に指定します。または、サービス・コンソールの管理から、管理者パスワードの設定を呼び出して更新します。)



ワークシートに以下のSQLを入力し文の実行をクリックすることにより、DBMS_CLOUDパッケージの実行権限をOracle APEXに割り当てます。ワークスペース名は、それぞれ適切な値に置き換えてください。画面の例では、MYWORKSPACEをワークスペース名として与え、DBMS_CLOUDへの実行権限の割り当てを行っています。スクリーンショットは載せませんが、この後に、DATA_PUMP_DIRの読み書き権限の付与も実行します。

GRANT EXECUTE ON DBMS_CLOUD TO ワークスペース名;

GRANT READ,WRITE ON directory DATA_PUMP_DIR TO ワークスペース名;



認証トークンを生成する

Oracle Cloudのコンソールから、**ユーザー設定**を呼び出します。



ユーザー設定のリソースより**認証トークン**を選択し、**トークンの生成**をクリックします。



ダイアログが開き、説明の入力を求められます。**説明**を入力し、**トークンの生成**をクリックします。



トークンが生成されたことを通知する画面になります。**コピー**のリンクをクリックし、認証トークンをコピーします。コピーしたトークンは無くさないようにします。コピーした後、この画面は閉じます。



認証トークン自体が作成されていることはユーザー設定の画面に一覧されますが、認証トークン自体を再度取り出すことはできません。



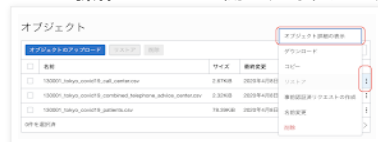
これで、認証トークンの取得は完了です。

CSVファイルへのURLパスを取得する

先ほど作成したバケットopendataを開き、**リソース**から**オブジェクト**を選んで、含まれるオブジェクトの一覧を表示します。



オブジェクトの詳細を表示するため、オブジェクトの**操作メニュー**を開き、**オブジェクト詳細の表示**を実行します。



それぞれ3つのCSVファイルについて、**URLパス(URI)**をコピーして、後で使用できるよう、どこかに保存しておきます。

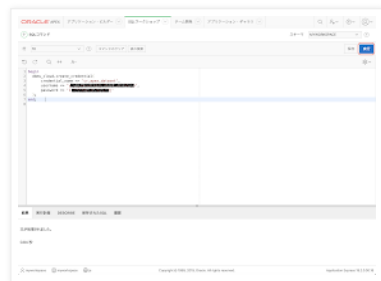


これでDBMS_CLOUDパッケージを使って外部表を作成するために必要な情報がすべて揃いました。

オブジェクト・ストレージにアクセスするための認証情報をデータベースに登録する

最初にオブジェクト・ストレージのアクセスに使用する認証情報(クリデンシャル) を登録します。以下のコマンドをOracle APEXのSQLワークショップから実行します。画面上の例では、 クリデンシャル名を**cr_apex_dataset**としています。

```
begin
  dbms_cloud.create_credential(
    credential_name => 'クリデンシャル名',
    username => 'OCIユーザー名',
    password => '認証トークン'
  );
end;
```



ここで私はちょっと引っ掛かったのですが、ユーザー名に**oracleidentitycloudservice/**で始まるものと、それが除かれているものの2種類登録されているはずです。oracleidentitycloudservice/で始まるユーザーに認証トークンを登録した(このブログの順に従うと、oracleidentitycloudservice/で始まるユーザーに認証トークンをつけています)のであれば、OCIユーザー名もその名前を指定する必要があります。

次のSQLを実行することで、登録したクリデンシャルを確認することができます。

```
select * from user_credentials;
```

今回の手順は演習なのでユーザー登録はこのようにしていますが、ログインに使うアカウントをこのような用途で使うのは心配です。以前にオブジェクト・ストレージの操作について書いた**記事**で、API呼び出しのためのユーザーのセットアップについて書いています。この方法で登録したユーザーをDBMS_CLOUD.CREATE_CREDENTIALによってクリデンシャルとして登録する方法が、**CREATE_CREDENTIAL Procedure (OCI Signing Key Credentials)**として説明されているので、よりシリアスな用途のときは、そちらを検討した方が良いでしょう。

オブジェクト・ストレージ上のCSVファイルを参照する外部表を作成する

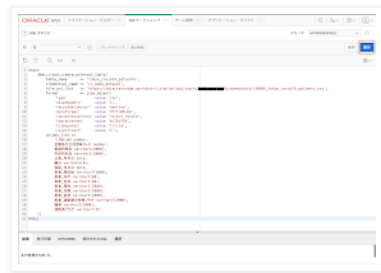
オブジェクト・ストレージ上のCSVファイルを参照する外部表**TOKYO_COVID19_PATIENTS**を作成します。元になっているデータは**東京都_新型コロナウイルス陽性患者発表詳細**です。SQLワークショップのSQLコマンドから実行します。

```
begin
  dbms_cloud.create_external_table(
    table_name => 'tokyo_covid19_patients',
    credential_name => 'cr_apex_dataset',
    file_uri_list => 'https://objectstorage.ap-tokyo-1.oraclecloud.com/n/xxxxxxx/b/opendata/o/130001_tokyo_covid19_patients.csv',
    format => json_object(
      'type' value 'csv',
      'skipheaders' value '1',
      'recorddelimiter' value 'newline',
      'dateformat' value 'YYYY-MM-DD',
      'conversionerrors' value 'reject_record',
      'characterset' value 'AL32UTF8',
      'trimspaces' value 'ltrim',
      'rejectlimit' value '1'),
    column_list =>
      'LINE_NO number,
      全国地方公共団体コード number,
      都道府県名 varchar2(2000),
      市区町村名 varchar2(2000),
      公表_年月日 date,
      曜日 varchar2(8),
      発症_年月日 date,
```

```

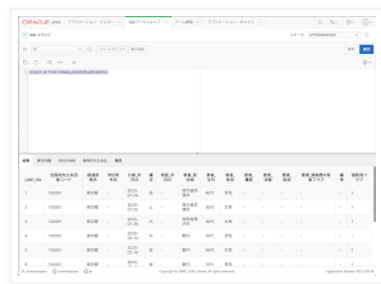
患者_居住地 varchar2(2000),
患者_年代 varchar2(80),
患者_性別 varchar2(80),
患者_属性 varchar2(2000),
患者_状態 varchar2(2000),
患者_症状 varchar2(2000),
患者_渡航歴の有無フラグ varchar2(2000),
備考 varchar2(2000),
退院済フラグ varchar2(8)'
);
end;

```



表が作成されたら、内容を確認してみましょう。以下の単純なSELECT文を実行します。

```
select * from tokyo_covid19_patients;
```



次に東京都 新型コロナコールセンター相談件数の外部表TOKYO_COVID19_CALL_CENTERを作成します。

```

begin
  dbms_cloud.create_external_table(
    table_name => 'tokyo_covid19_call_center',
    credential_name => 'cr_apex_dataset',
    file_uri_list => 'https://objectstorage.ap-tokyo-1.oraclecloud.com/n/xxxxxxx/b/opendata/o/130001_tokyo_covid19_call_center.csv',
    format => json_object(
      'type' value 'csv',
      'skipheaders' value '1',
      'recorddelimiter' value 'newline',
      'dateformat' value 'YYYY-MM-DD',
      'conversionerrors' value 'reject_record',
      'characteraset' value 'AL32UTF8',
      'trimspaces' value 'ltrim',
      'rejectlimit' value '1'),
    column_list =>
      '全国地方公共団体コード number,
      都道府県名 varchar2(2000),
      市区町村名 varchar2(2000),
      受付_年月日 date,
      曜日 varchar2(8),
      相談件数 number'
  );
end;

```

最後に東京都_新型コロナ受診相談窓口相談件数の外部表TOKYO_COVID19_COMBINED_TELPHONE_ADVICE_CENTERを作成します。

```

begin
  dbms_cloud.create_external_table(
    table_name => 'tokyo_covid19_combined_telephone_advice_center',
    credential_name => 'cr_apex_dataset',
    file_uri_list => 'https://objectstorage.ap-tokyo-1.oraclecloud.com/n/xxxxxxx/b/opendata/o/130001_tokyo_covid19_combined_telephone_advice_center.csv',
    format => json_object(
      'type' value 'csv',
      'skipheaders' value '1',
      'recorddelimiter' value 'newline',
      'dateformat' value 'YYYY-MM-DD',
      'conversionerrors' value 'reject_record',
      'characteraset' value 'AL32UTF8',
      'trimspaces' value 'ltrim',
      'rejectlimit' value '1'),
    column_list =>
      '全国地方公共団体コード number,
      都道府県名 varchar2(2000),
      市区町村名 varchar2(2000),
      受付_年月日 date,
      曜日 varchar2(8),
      相談件数 number'
  );
end;

```

```
);  
end;
```

3つの外部表が登録できました。これ以降は、Oracle APEXのアプリケーション作成になります。参照だけが可能な表ですが、表を元にしたアプリケーション作成ですので、Oracle APEXとしては一般的なアプリケーション作成になります。

Oracle APEXアプリケーションを作成する

作成した3つの外部表を使って、以下のような画面を作ることができます。外部表といっても、書き込みができないことを除けば通常の表と同様に扱えます。また、実際のデータはオブジェクト・ストレージ上のCSVファイルなので、それを最新のデータに置き換えるとAPEX側から見ることもできるデータも最新のものになります。



作成された表のDDLを確認してみたところ、以下でした。アクセス・ドライバはORACLE_LOADERで、オブジェクト・ストレージだからといって特別ではないようです。

```
CREATE TABLE "TOKYO_COVID19_PATIENTS"  
( "LINE_NO" NUMBER,  
  "全国地方公共団体コード" NUMBER,  
  "都道府県名" VARCHAR2(80) COLLATE "USING_NLS_COMP",  
    "市区町村名" VARCHAR2(2000) COLLATE "USING_NLS_COMP",  
  "公表_年月日" DATE,  
  "曜日" VARCHAR2(3) COLLATE "USING_NLS_COMP",  
  "発症_年月日" DATE,  
  "患者_居住地" VARCHAR2(80) COLLATE "USING_NLS_COMP",  
  "患者_年代" VARCHAR2(20) COLLATE "USING_NLS_COMP",  
  "患者_性別" VARCHAR2(20) COLLATE "USING_NLS_COMP",  
  "患者_属性" VARCHAR2(2000) COLLATE "USING_NLS_COMP",  
  "患者_状態" VARCHAR2(2000) COLLATE "USING_NLS_COMP",  
  "患者_症状" VARCHAR2(2000) COLLATE "USING_NLS_COMP",  
  "患者_渡航歴の有無フラグ" VARCHAR2(2000) COLLATE "USING_NLS_COMP",  
  "備考" VARCHAR2(2000) COLLATE "USING_NLS_COMP",  
  "退院済フラグ" VARCHAR2(8) COLLATE "USING_NLS_COMP"  
) DEFAULT COLLATION "USING_NLS_COMP"  
ORGANIZATION EXTERNAL  
( TYPE ORACLE_LOADER  
  DEFAULT DIRECTORY "DATA_PUMP_DIR"  
  ACCESS PARAMETERS  
    ( RECORDS IGNORE_HEADER=1 DELIMITED BY newline CHARACTERSET AL32UTF8 NOLOGFILE NOBADFILE NODISCARDFILE READSIZE=10000000 CREDENTIAL 'cr_apex_d'  
    FIELDS CSV WITHOUT EMBEDDED DATE_FORMAT DATE MASK 'YYYY-MM-DD' CONVERT_ERROR REJECT_RECORD LRTRIM  
  )  
  LOCATION  
    ( 'https://objectstorage.ap-tokyo-1.oraclecloud.com/n/xxxxxxxxxx/b/opendata/o/130001_tokyo_covid19_patients.csv'  
    )  
  )  
REJECT LIMIT 1  
PARALLEL  
/
```

DBMS_CLOUDパッケージに関するマニュアルは[こちら](#)になります。日本語のマニュアルは[こちら](#)です。

補足

今回のような、オープン・データであるCSVファイルを認証なしで取得できる場合であれば、Oracle APEXが提供しているプロシージャを使用して以下のようにデータを取り込むことも可能です。参考までに方法を紹介します。

東京都_新型コロナウイルス陽性患者発表詳細のデータを取り込みます。最初にデータをロードする表を作成します。

```
CREATE TABLE "TOKYO_COVID19_PATIENTS"  
( "LINE_NO" NUMBER,  
  "全国地方公共団体コード" NUMBER,  
  "都道府県名" VARCHAR2(2000),  
  "市区町村名" VARCHAR2(2000),  
  "公表_年月日" DATE,  
  "曜日" VARCHAR2(8),  
  "発症_年月日" DATE,
```

```
"患者_居住地" VARCHAR2(2000),
"患者_年代" VARCHAR2(80),
"患者_性別" VARCHAR2(80),
"患者_属性" VARCHAR2(2000),
"患者_状態" VARCHAR2(2000),
"患者_症状" VARCHAR2(2000),
"患者_渡航歴の有無フラグ" VARCHAR2(2000),
"備考" VARCHAR2(2000),
"退院済フラグ" VARCHAR2(8)
);
```

次に以下のSQLを実行することで、CSVファイルからのデータロードを行います。実行しているのはinsert select文で、その中でapex_data_parser.parseとapex_web_service.make_rest_request_bを使用しています。

```
insert into tokyo_covid19_patients(
  LINE_NO, "全国地方公共団体コード", "都道府県名", "市区町村名", "公表_年月日", "曜日",
  "発症_年月日", "患者_居住地", "患者_年代", "患者_性別", "患者_属性", "患者_状態",
  "患者_症状", "患者_渡航歴の有無フラグ", "備考", "退院済フラグ")
select col001, col002, col003, col004, to_date(col005, 'YYYY-MM-DD'), col006,
  to_date(col007, 'YYYY-MM-DD'), col008, col009, col010, col011,
  col012, col013, col014, col015, col016
from table(apex_data_parser.parse(
  p_content => apex_web_service.make_rest_request_b(
    'https://stopcovid19.metro.tokyo.lg.jp/data/130001_tokyo_covid19_patients.csv',
    'GET'),
  p_file_name => '130001_tokyo_covid19_patients.csv',
  p_skip_rows => 1)
);
```

東京都 新型コロナコールセンター相談件数のロードは、以下の表とINSERT文で行うことができます。表のDDLです。

```
CREATE TABLE "TOKYO_COVID19_CALL_CENTER"
( "全国地方公共団体コード" NUMBER,
"都道府県名" VARCHAR2(2000),
"市区町村名" VARCHAR2(2000),
"受付_年月日" DATE,
"曜日" VARCHAR2(8),
"相談件数" NUMBER
);
```

INSERT文は以下です。

```
insert into tokyo_covid19_call_center(
  "全国地方公共団体コード", "都道府県名", "市区町村名", "受付_年月日", "曜日", "相談件数"
)
select col001, col002, col003, to_date(col004, 'YYYY-MM-DD'), col005, col006
from table(apex_data_parser.parse(
  p_content => apex_web_service.make_rest_request_b(
    'https://stopcovid19.metro.tokyo.lg.jp/data/130001_tokyo_covid19_call_center.csv',
    'GET'),
  p_file_name => '130001_tokyo_covid19_call_center.csv',
  p_skip_rows => 1)
);
```

東京都_新型コロナ受診相談窓口相談件数のロードは、以下の表とINSERT文で行うことができます。表のDDLです。

```
CREATE TABLE "TOKYO_COVID19_COMBINED_TELEPHONE_ADVICE_CENTER"
( "全国地方公共団体コード" NUMBER,
"都道府県名" VARCHAR2(2000),
"市区町村名" VARCHAR2(2000),
"受付_年月日" DATE,
"曜日" VARCHAR2(8),
"相談件数" NUMBER
);
```

INSERT文は以下です。

```
insert into tokyo_covid19_combined_telephone_advice_center(
  "全国地方公共団体コード", "都道府県名", "市区町村名", "受付_年月日", "曜日", "相談件数"
)
select col001, col002, col003, to_date(col004, 'YYYY-MM-DD'), col005, col006
from table(apex_data_parser.parse(
  p_content => apex_web_service.make_rest_request_b(
    'https://stopcovid19.metro.tokyo.lg.jp/data/130001_tokyo_covid19_combined_telephone_advice_center.csv',
    'GET'),
  p_file_name => '130001_tokyo_covid19_combined_telephone_advice_center.csv',
  p_skip_rows => 1)
);
```

最後の最後に謝辞。Qiitaにある、こちらの記事は**じめてのAutonomous Databaseへのデータロード（Object Storage経由の場合）**も参考にさせていただきました。ありがとうございます。

完

Yuji N. 時刻: 18:27

共有

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](#)