

# 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。

2022年3月28日月曜日

## GitHubにあるdb-sample-schemasをAutonomous Databaseにインストールする

GitHubにある[db-sample-schemas](#)のsales-historyをAlways FreeのAutonomous Databaseにインストールするつもりで準備作業をしました。準備が完了してsales-historyのサンプルをインストールしてみたところ、Autonomous Databaseには最初からスキーマSHがインストールされていて、db-sample-schemasのスクリプトを実行する必要はありませんでした。その他のサンプルのインストールも確認したところ、以下の結果になっています。

**bus\_intelligence:** OEおよびSH、特にSHに依存しているためADBではインストールでエラーが発生する。

**customer\_orders:** インストールできる。

**human\_resource:** インストールできる。

**info\_exchange:** Streamsに関連したエラーが発生する。

**order\_entry:** ディレクトリを作るところ(CREATE DIRECTORY)と、XDB関連でエラーが発生する。

**product\_media:** ディレクトリを作るところ(CREATE DIRECTORY)でエラーが発生する。

**sales\_history:** スキーマSHが存在するため、インストールはできない。

**shipping:** OE(order\_entry)に依存しているが、スキーマOEが存在していればエラーは発生しない。

以前に[Real Application Securityの動作確認](#)を行った際にdb-sample-schemasのhuman\_resourceをインストールしていたので、他のサンプルもインストールできるかと思ったのですが、そうではなかったです。

以前の記事は1年以上前の作業です。今回の作業では、プラットフォームもOracle Linux 8（シェイプはAlways FreeのVM.Standard.E2.1.Micro）を使い、実行しているコマンドも異なることから、改めて作業を記録します。また、Autonomous Databaseの接続に使用するウォレットは、ociコマンドを使って生成します。

Always Freeのコンピュータ・インスタンスを、作成した直後の状態から作業を始めます。

今回は新規にユーザーを作成せず、ユーザーopcで作業をしています。opcは特別な権限を持つユーザーなので、opc以外のユーザーを作成して作業した方が良いでしょう。

### ociコマンドのインストール

ユーザーopcで接続して、ociコマンドをインストールします。これでよろしいですか？と聞かれたら、yを入力します。

```
sudo dnf install python36-oci-cli
```

```
[opc@dwcp2 ~]$ sudo dnf install python36-oci-cli
Ksplice for Oracle Linux 8 (x86_64) 5.7 MB/s | 708 kB 00:00
MySQL 8.0 for Oracle Linux 8 (x86_64) 15 MB/s | 2.2 MB 00:00
MySQL 8.0 Tools Community for Oracle Linux 8 (x86_64) 2.4 MB/s | 249 kB 00:00
MySQL 8.0 Connectors Community for Oracle Linux 8 (x86_64) 200 kB/s | 20 kB 00:00
Oracle Software for OCI users on Oracle Linux 8 (x86_64) 29 MB/s | 29 MB 00:00
Oracle Linux 8 BaseOS Latest (x86_64) 28 MB/s | 43 MB 00:01
Oracle Linux 8 Application Stream (x86_64) 24 MB/s | 32 MB 00:01
Oracle Linux 8 Addons (x86_64) 7.8 MB/s | 2.9 MB 00:00
Latest Unbreakable Enterprise Kernel Release 6 for Oracle Linux 8 (x86_64) 24 MB/s | 40 MB 00:01
```

依存関係が解決しました。

パッケージ	Arch	バージョン	リポジトリ	サイズ
-------	------	-------	-------	-----

インストール:

python36-oci-cli	noarch	3.6.2-1.el8	ol8_addons	13 M
------------------	--------	-------------	------------	------

アップグレード:

python36-oci-sdk	x86_64	2.61.0-1.el8	ol8_addons	20 M
------------------	--------	--------------	------------	------

依存関係のインストール:

python3-arrow	noarch	1.1.1-1.el8	ol8_oci_included	119 k
python3-jmespath	noarch	0.10.0-1.el8	ol8_oci_included	48 k
python3-terminaltables	noarch	3.1.0-1.0.1.el8	ol8_oci_included	31 k
python3-typing-extensions	noarch	3.7.4.2-1.el8	ol8_oci_included	47 k

トランザクションの概要

インストール 5 パッケージ

アップグレード 1 パッケージ

ダウンロードサイズの合計: 33 M

これでよろしいですか? [y/N]: y

パッケージのダウンロード:

```
(1/6): python3-terminaltables-3.1.0-1.0.1.el8.noarch.rpm 425 kB/s | 31 kB 00:00
(2/6): python3-typing-extensions-3.7.4.2-1.el8.noarch.rpm 1.4 MB/s | 47 kB 00:00
(3/6): python3-jmespath-0.10.0-1.el8.noarch.rpm 251 kB/s | 48 kB 00:00
(4/6): python3-arrow-1.1.1-1.el8.noarch.rpm 553 kB/s | 119 kB 00:00
(5/6): python36-oci-cli-3.6.2-1.el8.noarch.rpm 18 MB/s | 13 MB 00:00
(6/6): python36-oci-sdk-2.61.0-1.el8.x86_64.rpm 17 MB/s | 20 MB 00:01
```

合計 25 MB/s | 33 MB 00:01

トランザクションの確認を実行中

トランザクションの確認に成功しました。

トランザクションのテストを実行中

トランザクションのテストに成功しました。

トランザクションを実行中

準備	:	1/1
アップグレード中	: python36-oci-sdk-2.61.0-1.el8.x86_64	1/7
インストール中	: python3-typing-extensions-3.7.4.2-1.el8.noarch	2/7
インストール中	: python3-arrow-1.1.1-1.el8.noarch	3/7
インストール中	: python3-terminaltables-3.1.0-1.0.1.el8.noarch	4/7
インストール中	: python3-jmespath-0.10.0-1.el8.noarch	5/7
インストール中	: python36-oci-cli-3.6.2-1.el8.noarch	6/7
整理	: python36-oci-sdk-2.57.0-1.el8.x86_64	7/7
scriptletの実行中	: python36-oci-sdk-2.57.0-1.el8.x86_64	7/7
検証	: python3-arrow-1.1.1-1.el8.noarch	1/7
検証	: python3-jmespath-0.10.0-1.el8.noarch	2/7
検証	: python3-terminaltables-3.1.0-1.0.1.el8.noarch	3/7
検証	: python3-typing-extensions-3.7.4.2-1.el8.noarch	4/7

検証	: python36-oci-cli-3.6.2-1.el8.noarch	5/7
検証	: python36-oci-sdk-2.61.0-1.el8.x86_64	6/7
検証	: python36-oci-sdk-2.57.0-1.el8.x86_64	7/7

アップグレード済み:

python36-oci-sdk-2.61.0-1.el8.x86\_64

インストール済み:

python3-arrow-1.1.1-1.el8.noarch

python3-jmespath-0.10.0-1.el8.noarch

python3-terminaltables-3.1.0-1.0.1.el8.noarch

python3-typing-extensions-3.7.4.2-1.el8.noarch

python36-oci-cli-3.6.2-1.el8.noarch

完了しました!

[opc@dwcp2 ~]\$

## テナンシとユーザーのOCIDの確認

OCIのコンソールにサインインし、右上の**プロファイル**を開きます。



テナンシを開いて、**OCID**をコピーします。ociコマンドのセットアップに使用するので、どこかにペーストしておきます。



**ユーザー設定**を開いて、**OCID**をコピーします。ociコマンドのセットアップに使用するので、どこかにペーストしておきます。



APIキーの作成は、oci setup configコマンドを実行して生成した公開鍵を使って行います。

## oci setup configの実行

コンピュータ・インスタンスに戻り、ociコマンドを構成します。

### oci setup config

**Enter a location for your config**は作成される構成ファイルの名前です。デフォルトのまま変更しません。

**Enter a user OCID**は、ユーザのOCIDです。コピー済みのユーザのOCIDを貼り付けます。

**Enter a tenancy OCID**は、テナンシのOCIDです。コピー済みのテナンシのOCIDを貼り付けます。

**Enter a region by index or name**は、リージョンを数値か名前で指定します。今回の例では**34**、**us-ashburn-1**を指定しています。東京リージョンであれば、**13**または**ap-tokyo-1**を指定します。

**Do you want to generate a new API Signing RSA key pair?**にYを入力し、APIキーとして登録するRSAのキーペア（APIキーとして登録するのは公開鍵）を生成します。

**Enter a directory for your keys to be created**は、RSAのキーペアが作成されるディレクトリを指定します。デフォルトのまま変更しません。

**Enter a name for your key**は、RSAのキーペアを保存するファイル名の一部になります。デフォルトのまま変更しません。公開鍵はこの識別子に**\_public.pem**が追加されます。秘密鍵は**.pem**が追加されます。

以上でセットアップでの質問への入力は完了です。構成ファイルが作成されます。

**Public key written to**として表示されるファイル名を覚えておきます。このファイルの内容を、ユーザのAPIキーを作成するときに使用します。

```
[opc@dwcp2 ~]$ oci setup config
This command provides a walkthrough of creating a valid CLI config file.

The following links explain where to find the information required by this
script:

User API Signing Key, OCID and Tenancy OCID:

    https://docs.cloud.oracle.com/Content/API/Concepts/apisigningkey.htm#Other

Region:

    https://docs.cloud.oracle.com/Content/General/Concepts/regions.htm

General config documentation:

    https://docs.cloud.oracle.com/Content/API/Concepts/sdkconfig.htm
```

Enter a location for your config [/home/opc/.oci/config]:

Enter a user OCID: コピー済みのテナンシのOCID

Enter a tenancy OCID: コピー済みのユーザのOCID

Enter a region by index or name(e.g.

1: af-johannesburg-1, 2: ap-chiyoda-1, 3: ap-chuncheon-1, 4: ap-dcc-canberra-1, 5: ap-hyderabad-1,  
6: ap-ibarak-1, 7: ap-melbourne-1, 8: ap-mumbai-1, 9: ap-osaka-1, 10: ap-seoul-1,  
11: ap-singapore-1, 12: ap-sydney-1, 13: ap-tokyo-1, 14: ca-montreal-1, 15: ca-toronto-1,  
16: eu-amsterdam-1, 17: eu-frankfurt-1, 18: eu-marseille-1, 19: eu-milan-1, 20: eu-stockholm-1,

```
21: eu-zurich-1, 22: il-jerusalem-1, 23: me-abudhabi-1, 24: me-dcc-muscat-1, 25: me-dubai-1,
26: me-jeddah-1, 27: sa-santiago-1, 28: sa-saopaulo-1, 29: sa-vinhedo-1, 30: uk-cardiff-1,
31: uk-gov-cardiff-1, 32: uk-gov-london-1, 33: uk-london-1, 34: us-ashburn-1, 35: us-gov-ashburn-1,
36: us-gov-chicago-1, 37: us-gov-phoenix-1, 38: us-langley-1, 39: us-luke-1, 40: us-phoenix-1,
41: us-sanjose-1): 34
```

Do you want to generate a new API Signing RSA key pair? (If you decline you will be asked to supply the path to an existing key.) [Y/n]: Y

Enter a directory for your keys to be created [/home/opc/.oci]:

Enter a name for your key [oci\_api\_key]:

Public key written to: /home/opc/.oci/oci\_api\_key\_public.pem

Enter a passphrase for your private key (empty for no passphrase):

Private key written to: /home/opc/.oci/oci\_api\_key.pem

Fingerprint: fc:45:ac:d7:4a:fa:c3:d1:60:94:c2:8d:de:90:97:ac

Config written to /home/opc/.oci/config

If you haven't already uploaded your API Signing public key through the console, follow the instructions on the page linked below in the section 'How to upload the public key':

<https://docs.cloud.oracle.com/Content/API/Concepts/apisigningkey.htm#How2>

```
[opc@dwcp2 ~]$
```

## APIキーの追加

**oci setup config**を実行して生成された**公開鍵**を、クリップボードに**コピー**します。

```
[opc@dwcp2 ~]$ cat .oci/oci_api_key_public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAzi1QIn/kVIju/edTp+W0
VVEgXuBQo10F/8YUla00BjjLH1VMpHLnFdp96/2g3gY23INuYvcqy+PGOG8e6vwb
ipVvN05Uq9ZLqwvLSJzIMjSqxmHudJXYYSheep+rjhYnObfXpRMLk1NwL5v60s8/
FXY3R3wCmQSU8vISeBL3dLLJkJbg85SECLTf5e7E3tTAaQ9t13tbDL1FA+RUOSTe
hFJCeWJVUtNK4L1qwW07JJr5y2rR8hGYgv18CZkAodPpzy2wgSMIS0CA4YUnYYXR
AH1iCM2yhK0dHJUCb9ueMCIgtJN+M7Leo1cKp42Quqo7e/mHX2CznMu6TxIRgc+acwIDAQAB
-----END PUBLIC KEY-----
[opc@dwcp2 ~]$
```

OCIコンソールの**プロファイル**から**ユーザー設定**を開き、**リソース**から**APIキー**を選択します。

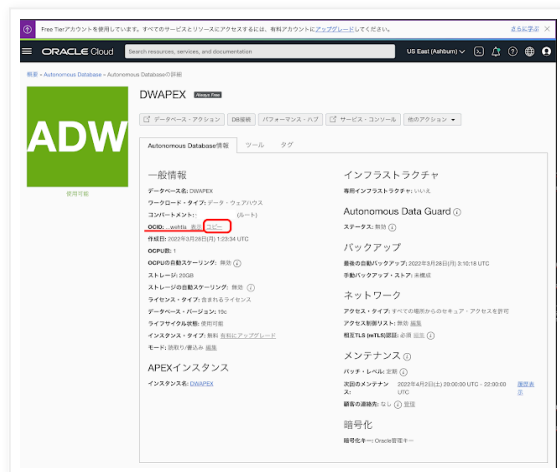
**APIキーの追加**をクリックします。ダイアログが開くので、**公開キーの貼付け**を選択し、先ほどコピーした公開鍵を**公開キー**にペーストします。

以上を行い、**追加**をクリックします。



OCIコンソールより接続先となるAutonomous Databaseを開き、OCIDを確認します。

OCIDをクリップボードにコピーします。



コンピュータ・インスタンスに移り、ウォレットを生成するコマンドを実行します。

`oci db autonomous-database generate-wallet --autonomous-database-id ADBのOCID --file ウォレット・ファイル名 --password "ウォレットのパスワード"`

以下ではウォレットのファイル名として、**Wallet\_DWAPEX.zip**を指定しています。ウォレットの生成コマンドは、成功したときにはメッセージを表示しないようです。

```
[opc@dwcp2 ~]$ oci db autonomous-database generate-wallet --autonomous-database-id ADBのOCID --file Wallet_DWAPEX.zip --password "パスワード"
[opc@dwcp2 ~]$ ls -l Wallet_DWAPEX.zip
-rw-rw-r--. 1 opc opc 21627  3月 28 04:28 Wallet_DWAPEX.zip
[opc@dwcp2 ~]$
```

## データベースへ接続するツールのインストール

`sqlplus`と`sqlldr`をインストールします。

最初にOracle Instance Clientが含まれるリポジトリを追加します。

`sudo dnf -y install oracle-instantclient-release-el8`

```
[opc@dwcp2 ~]$ sudo dnf -y install oracle-instantclient-release-el8
メタデータの期限切れの最終確認: 1:14:19 時間前の 2022年03月28日 03時23分34秒 に実施しました。
依存関係が解決しました。
=====
   パッケージ                                Arch              バージョン        リポジトリ          サイズ
=====
インストール:
  oracle-instantclient-release-el8          x86_64            1.0-1.el8         ol8_baseos_latest   16 k

トランザクションの概要
=====
インストール  1 パッケージ

ダウンロードサイズの合計: 16 k
インストール後のサイズ: 18 k
パッケージのダウンロード:
oracle-instantclient-release-el8-1.0-1.el8.x86_64.rpm          313 kB/s | 16 kB      00:00
-----
合計                                                            301 kB/s | 16 kB      00:00
トランザクションの確認を実行中
トランザクションの確認に成功しました。
トランザクションのテストを実行中
トランザクションのテストに成功しました。
トランザクションを実行中
  準備                :
  インストール中      : oracle-instantclient-release-el8-1.0-1.el8.x86_64          1/1
  scriptletの実行中: oracle-instantclient-release-el8-1.0-1.el8.x86_64          1/1
  検証                : oracle-instantclient-release-el8-1.0-1.el8.x86_64          1/1

インストール済み:
```

oracle-instantclient-release-el8-1.0-1.el8.x86\_64

完了しました!

[opc@dwcp2 ~]\$

sqlplusが含まれるパッケージoracle-instantclient-sqlplus、sqlldrが含まれるパッケージoracle-instantclient-tools、それとGitHubよりdb-sample-schemasをクローンする際に使用するgitもインストールします。

sudo dnf -y install oracle-instantclient-sqlplus oracle-instantclient-tools git

[opc@dwcp2 ~]\$ sudo dnf -y install oracle-instantclient-sqlplus oracle-instantclient-tools git  
Oracle Instant Client 21 for Oracle Linux 8 (x86\_64) 79 kB/s | 9.9 kB 00:00  
依存関係が解決しました。

パッケージ	Arch	バージョン	リポジトリ	サイズ
インストール:				
git	x86_64	2.27.0-1.el8	ol8_appstream	164 k
oracle-instantclient-sqlplus	x86_64	21.5.0.0.0-1	ol8_oracle_instantclient21	710 k
oracle-instantclient-tools	x86_64	21.5.0.0.0-1	ol8_oracle_instantclient21	864 k
依存関係のインストール:				
git-core	x86_64	2.27.0-1.el8	ol8_appstream	5.7 M
git-core-doc	noarch	2.27.0-1.el8	ol8_appstream	2.5 M
oracle-instantclient-basic	x86_64	21.5.0.0.0-1	ol8_oracle_instantclient21	53 M
perl-Error	noarch	1:0.17025-2.el8	ol8_appstream	46 k
perl-Git	noarch	2.27.0-1.el8	ol8_appstream	78 k
perl-TermReadKey	x86_64	2.37-7.el8	ol8_appstream	40 k

トランザクションの概要

インストール 9 パッケージ

ダウンロードサイズの合計: 63 M

インストール後のサイズ: 290 M

パッケージのダウンロード:

(1/9): oracle-instantclient-sqlplus-21.5.0.0.0-1.x86_64.rpm	4.8 MB/s   710 kB	00:00
(2/9): oracle-instantclient-tools-21.5.0.0.0-1.x86_64.rpm	3.1 MB/s   864 kB	00:00
(3/9): git-2.27.0-1.el8.x86_64.rpm	1.2 MB/s   164 kB	00:00
(4/9): git-core-2.27.0-1.el8.x86_64.rpm	16 MB/s   5.7 MB	00:00
(5/9): git-core-doc-2.27.0-1.el8.noarch.rpm	5.2 MB/s   2.5 MB	00:00
(6/9): perl-Error-0.17025-2.el8.noarch.rpm	312 kB/s   46 kB	00:00
(7/9): perl-TermReadKey-2.37-7.el8.x86_64.rpm	474 kB/s   40 kB	00:00
(8/9): perl-Git-2.27.0-1.el8.noarch.rpm	841 kB/s   78 kB	00:00
(9/9): oracle-instantclient-basic-21.5.0.0.0-1.x86_64.rpm	27 MB/s   53 MB	00:01

合計 31 MB/s | 63 MB 00:02

トランザクションの確認を実行中

トランザクションの確認に成功しました。

トランザクションのテストを実行中

トランザクションのテストに成功しました。

トランザクションを実行中

準備	:	1/1
インストール中	: git-core-2.27.0-1.el8.x86_64	1/9
インストール中	: oracle-instantclient-basic-21.5.0.0.0-1.x86_64	2/9
scriptletの実行中:	oracle-instantclient-basic-21.5.0.0.0-1.x86_64	2/9
インストール中	: git-core-doc-2.27.0-1.el8.noarch	3/9
インストール中	: perl-TermReadKey-2.37-7.el8.x86_64	4/9



```

インストール中    : perl-Error-1:0.17025-2.el8.noarch          5/9
インストール中    : perl-Git-2.27.0-1.el8.noarch             6/9
インストール中    : git-2.27.0-1.el8.x86_64                 7/9
インストール中    : oracle-instantclient-sqlplus-21.5.0.0.0-1.x86_64 8/9
scriptletの実行中: oracle-instantclient-sqlplus-21.5.0.0.0-1.x86_64 8/9
インストール中    : oracle-instantclient-tools-21.5.0.0.0-1.x86_64 9/9
scriptletの実行中: oracle-instantclient-tools-21.5.0.0.0-1.x86_64 9/9
検証              : oracle-instantclient-basic-21.5.0.0.0-1.x86_64 1/9
検証              : oracle-instantclient-sqlplus-21.5.0.0.0-1.x86_64 2/9
検証              : oracle-instantclient-tools-21.5.0.0.0-1.x86_64 3/9
検証              : git-2.27.0-1.el8.x86_64                 4/9
検証              : git-core-2.27.0-1.el8.x86_64            5/9
検証              : git-core-doc-2.27.0-1.el8.noarch         6/9
検証              : perl-Error-1:0.17025-2.el8.noarch        7/9
検証              : perl-Git-2.27.0-1.el8.noarch             8/9
検証              : perl-TermReadKey-2.37-7.el8.x86_64       9/9

```

インストール済み:

```

git-2.27.0-1.el8.x86_64          git-core-2.27.0-1.el8.x86_64
git-core-doc-2.27.0-1.el8.noarch  oracle-instantclient-basic-21.5.0.0.0-1.x86_64
oracle-instantclient-sqlplus-21.5.0.0.0-1.x86_64  oracle-instantclient-tools-21.5.0.0.0-1.x86_64
perl-Error-1:0.17025-2.el8.noarch  perl-Git-2.27.0-1.el8.noarch
perl-TermReadKey-2.37-7.el8.x86_64

```

完了しました!

```
[opc@dwcp2 ~]$
```

## Autonomous Databaseへの接続

Autonomous Databaseへ接続するために必要な準備を行います。

ユーザー**opc**の**ホーム・ディレクトリ**を**ORACLE\_HOME**にします。ORACLE\_HOME以下にディレクトリ**network/admin**を作成し、ウォレットの内容を展開しています。また、インストールされたsqlplus、sqlldrといったツールへのパスを通しています。

デフォルトの接続先として（TWO\_TASKの設定）、**dwapex\_low**を設定してます。この値はADBのインスタンス名によって変わります。

```

[opc@dwcp2 ~]$ mkdir -p network/admin
[opc@dwcp2 ~]$ unzip -d network/admin Wallet_DWAPEX.zip
Archive:  Wallet_DWAPEX.zip
  inflating: network/admin/README
  inflating: network/admin/cwallet.sso
  inflating: network/admin/tnsnames.ora
  inflating: network/admin/truststore.jks
  inflating: network/admin/ojdbc.properties
  inflating: network/admin/sqlnet.ora
  inflating: network/admin/ewallet.p12
  inflating: network/admin/keystore.jks
[opc@dwcp2 ~]$ export ORACLE_HOME=$HOME
[opc@dwcp2 ~]$ export PATH=/usr/lib/oracle/21/client64/bin:$PATH
[opc@dwcp2 ~]$ export TWO_TASK=dwapex_low
[opc@dwcp2 ~]$ export NLS_LANG=American_America.AL32UTF8

```

NLS\_LANGも設定しておきます。今回のようにOracle Corporationが提供しているスクリプトを実行する場合は、言語とテリトリは**American\_America**を指定しておいた方が無難です。

sqlplusを実行し、管理ユーザーADMINにてAutonomous Databaseに接続します。

### sqlplus admin/管理ユーザーのパスワード

```
[opc@dwcp2 ~]$ sqlplus admin/管理ユーザーのパスワード

SQL*Plus: Release 21.0.0.0.0 - Production on Mon Mar 28 04:56:54 2022
Version 21.5.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Last Successful login time: Mon Mar 28 2022 02:34:38 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.15.0.1.0

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.15.0.1.0
[opc@dwcp2 ~]$
```

データベースへの接続までは完了しました。

## サンプル・スキーマのインストール手順

サンプル・スキーマのインストール手順は、GitHubのREADMEに記述されている手順に従います。

cd \$HOME

git clone https://github.com/oracle/db-sample-schemas.git

```
[opc@dwcp2 ~]$ cd $HOME
[opc@dwcp2 ~]$ git clone https://github.com/oracle/db-sample-schemas.git
Cloning into 'db-sample-schemas'...
remote: Enumerating objects: 636, done.
remote: Counting objects: 100% (67/67), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 636 (delta 43), reused 48 (delta 28), pack-reused 569
Receiving objects: 100% (636/636), 35.61 MiB | 6.86 MiB/s, done.
Resolving deltas: 100% (344/344), done.
Updating files: 100% (397/397), done.
[opc@dwcp2 ~]$
```

スクリプトに含まれているパスを、ローカルにインストールされたパスに置き換えます。

cd db-sample-schemas/

perl -p -i.bak -e 's#\_\_SUB\_\_CWD\_\_#\$(pwd)'#g' \*.sql \*/\*.sql \*/\*.dat

```
[opc@dwcp2 ~]$ cd db-sample-schemas/
[opc@dwcp2 db-sample-schemas]$ perl -p -i.bak -e 's#__SUB__CWD__#$(pwd)'#g' *.sql */*.sql */*.dat
[opc@dwcp2 db-sample-schemas]$
```

今回はsales-historyのインストールを試みました。

sales-historyのインストール・スクリプトが含まれるディレクトリへ移動します。

```
cd db-sample-schemas/sales_history/
```

```
[opc@dwcp2 ~]$ cd db-sample-schemas/sales_history/  
[opc@dwcp2 sales_history]$
```

Autonomous Databaseが接続先の場合、AS SYSDBAでの接続はできません。AS SYSDBAが指定されている接続文を探します。

```
grep -i sysdba *.sql
```

```
[opc@dwcp2 sales_history]$ grep -i sysdba *.sql  
sh_main.sql:CONNECT sys/&pass_sys@&connect_string AS SYSDBA;  
[opc@dwcp2 sales_history]$
```

sh\_main.sqlに含まれているので、sh\_main.sqlを開いて、sysをadminに変更します。また、AS SYSDBAの部分を削除します。

sh\_main.sqlの以下の部分を

```
CONNECT sys/&pass_sys@&connect_string AS SYSDBA;
```

次に変更します。

```
CONNECT admin/&pass_sys@&connect_string;
```

以上で、サンプル・スキーマのインストールを行う準備はおおむね完了です。実際にサンプル・スキーマのインストールを試みた結果は、記事の最初にまとめています。

大体のインストール・スクリプトで、ユーザーのデフォルト表領域とTEMP表領域の指定が求められます。Autonomous Databaseではそれぞれ、DATAまたはTEMPを指定します。

以上です。

残念な結果ではありますが、作業ログが何かの役にたてば幸いです。

完

Yuji N. 時刻: 15:46

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.

---