

日々是Oracle APEX

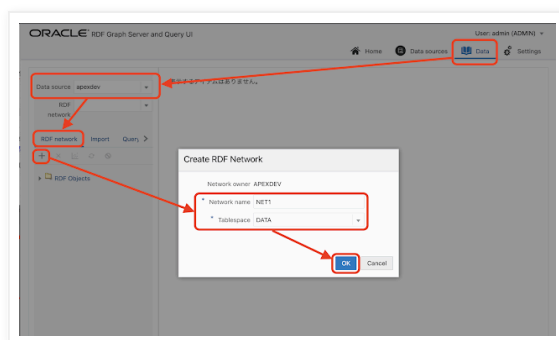
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年12月8日 水曜日

Oracle RDF Graph ServerをAutonomous Databaseで使用する(2) - SPARQLを実行する

先の記事で構築した環境を使って、マニュアルに記載されている雑誌記事の例を実行してみます。

最初にネットワークを作成します。Dataタブを開きます。Data Sourceにapexdevを選択し、RDF networkのプラス(+)をクリックします。Create RDF Networkのダイアログが開くので、Network nameとしてNET1、TablespaceとしてDATAを選択しOKをクリックします。

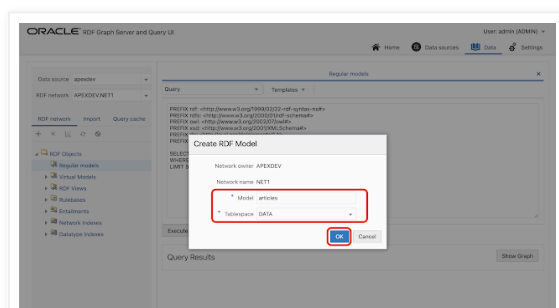


RDFセマンティック・ネットワークとしてNET1が作成されたら、続けてモデルを作成します。

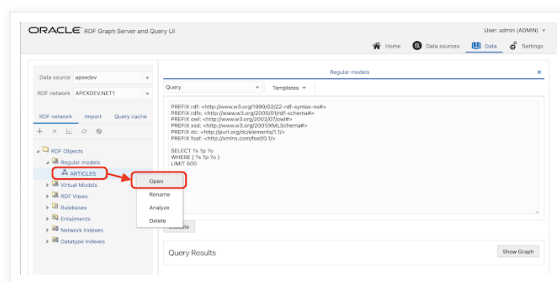
RDF networkとしてAPEXDEV.NET1を選択します。RDF networkのRDF Objectsを開き、Regular modelsの上で右クリックしてメニューを表示させ、Createを実行します。



Modelとしてarticles、TablespaceとしてDATAを選択し、OKをクリックします。



モデル**ARTICLES**が作成されたら、**Regular models**を開いて選択します。右クリックでメニューを表示させ、**Open**を実行します。



モデルのデータを投入します。与えるテキストとして**SPARQL**を選択し、**Update**を選びます。以下の記述を行い、**Execute**をクリックします。

```
PREFIX nature: <http://nature.example.com/>
PREFIX      dc: <http://purl.org/dc/elements/1.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>

INSERT DATA {

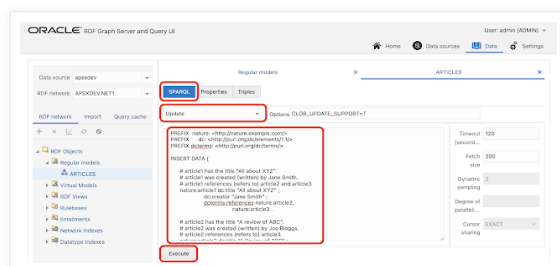
    # article1 has the title "All about XYZ".
    # article1 was created (written) by Jane Smith.
    # article1 references (refers to) article2 and article3
    nature:article1 dc:title "All about XYZ" ;
                      dc:creator "Jane Smith" ;
                      dcterms:references nature:article2,
                      nature:article3 .

    # article2 has the title "A review of ABC".
    # article2 was created (written) by Joe Bloggs.
    # article2 references (refers to) article3.
    nature:article2 dc:title "A Review of ABC" ;
                      dc:creator "Joe Bloggs" ;
                      dcterms:references nature:article3 .

}
```

articles.srx hosted with ❤ by GitHub

[view raw](#)



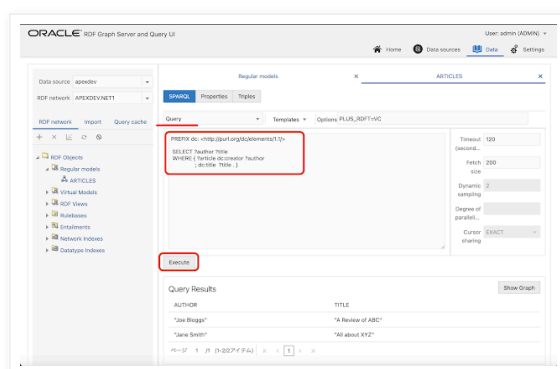
モデル**ARTICLES**を検索して、投入されたデータを確認します。Updateの代わりに**Query**を選択し、マニュアルに記載されている以下のSPARQL問い合わせを実行します。すべての記事の作者とタイトルをリストします。

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT ?author ?title
WHERE { ?article dc:creator ?author
        ; dc:title ?title . }
```

articlequery1.srx hosted with ❤ by GitHub

[view raw](#)



マニュアルの例に含まれる、もうひとつのSPARQL問い合わせを実行します。Article1に参照されている記事の一覧です。

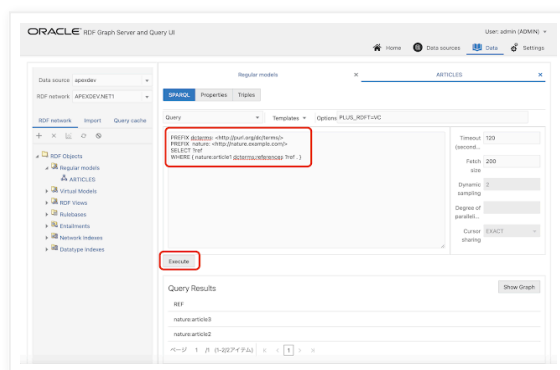
```
PREFIX dcterms: <http://purl.org/dc/terms/>
```

```
PREFIX nature: <http://nature.example.com/>
```

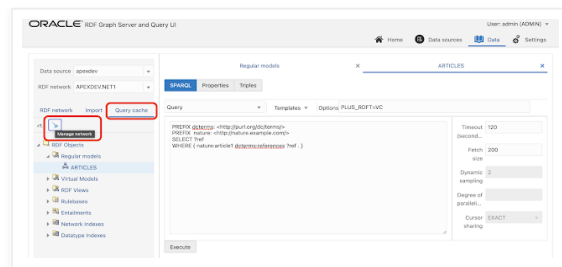
```
SELECT ?ref
WHERE { nature:article1 dcterms:references ?ref . }
```

articlequery2.srx hosted with ❤ by GitHub

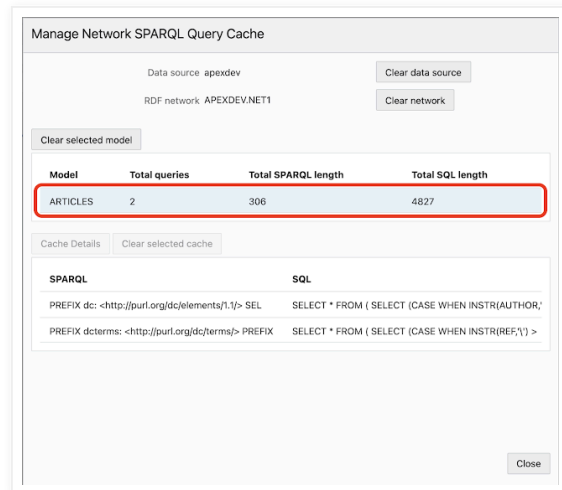
[view raw](#)



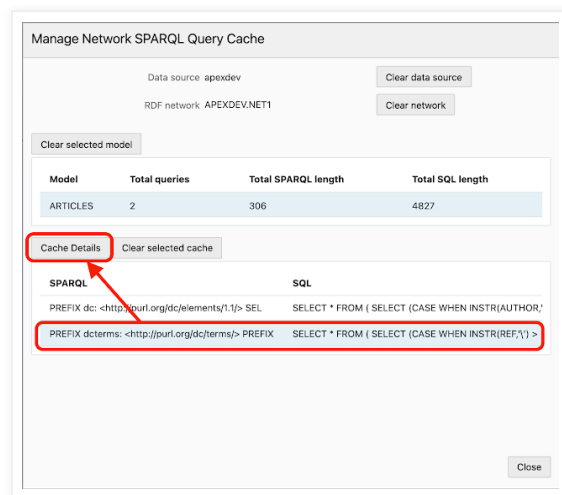
Oracle Databaseで実際に実行されているSQLを確認します。**Query cache**を選択し、**Manage network**のアイコンをクリックします。



Manage Network SPARQL Query Cacheのダイアログが開きます。ModelのARTICLESの列をクリックすると、そのモデルで実行されたSPARQLと、変換されたSQLの一覧が表示されます。



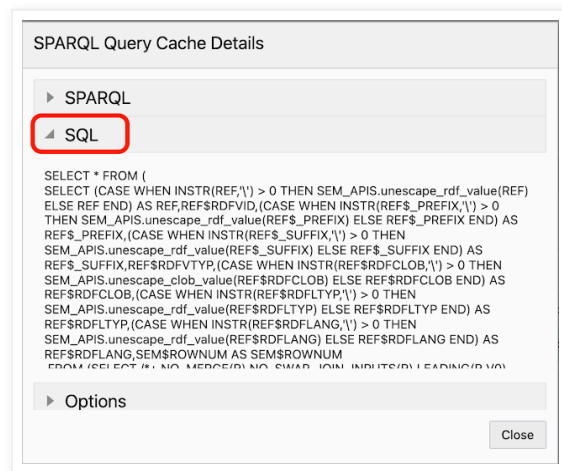
詳細を確認するSPARQL文を選択し、Cache Detailsをクリックします。



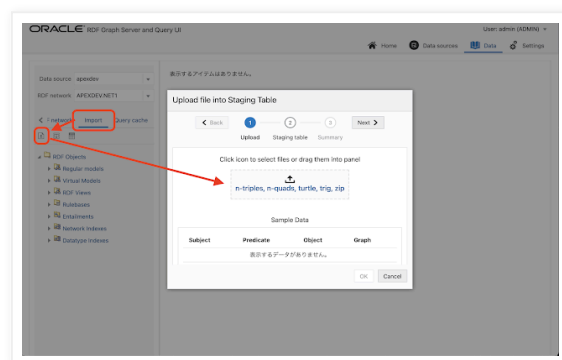
キャッシュの詳細として、SPARQLの表示がされます。



SQL文の確認もできます。このSQLはデータベースで直接実行できます。



本記事では実行しませんが、この他に**Import**の機能があります。N-Triples(.nt)、N-Quads(.nq)、Turtle(.ttl)、Trig(.trig)および、それらをZIPに固めたファイルのインポートに対応しています。



RDF Graph Serverを使用してSPARQLを実行する方法について、簡単に紹介してみました。

Oracle RDF Graph Serverについての[マニュアルの記載はこちら](#)です。

続く

[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.
