

Use Cases

Distributed Partitions

Data Loads

Migrations

Data Validations

Types of Foreign Data Wrappers

SQL

NoSQL

File

Miscellaneous

PostgreSQL

FDW provides a means to access data outside a PostgreSQL cluster. The data may be on a filesystem, on a MySQL server, or another PostgreSQL cluster. We will take a look at two examples: one accessing a file and another accessing a PostgreSQL. The steps usually involve:

1. Installing the extension.
2. Creating a server object. This provides details of the server (such as IP and port or path)
3. Creating user mapping. This involves providing the authentication information to connect to other databases. This step may or may not be necessary depending on the type of server object.
4. Creating the foreign table. This creates a table that maps to another table or file on the server object.

***** FDW for files *****

We will walk through an example:

1. Install the extension as follows:

```
CREATE EXTENSION file_fdw;
```

2. Create the server object with the following command:

```
CREATE SERVER file_server FOREIGN DATA WRAPPER file_fdw;
```

3. Create the foreign table. The syntax is very similar to the standard CREATE TABLE syntax. We have the keyword FOREIGN and need to mention the server object it points to:

```
CREATE FOREIGN TABLE file_to_tbl(  
id integer, name varchar(40),sal int  
)  
SERVER file_server  
OPTIONS (  
delimiter ',',  
filename '/tmp/emp1.txt',  
format 'csv'  
);
```

file named names.csv in /home/postgres with data that

looks like this:

1, Jayadevan
2, Steve
3, Jones
4, Anne
5, Julie

***** Postgres FDW*****

Let's create a new user for this:

```
CREATE USER myuser1 PASSWORD 'myuser';  
ALTER USER myuser1 WITH CREATEDB;
```

2. Connect to pgp (an existing database) as postgres and create the table that will be accessed using FDW:

```
CREATE TABLE prod_table (id serial, name varchar(50));  
INSERT INTO prod_table (name) VALUES ('Scott'), ('Thomson');  
GRANT SELECT on prod_table to myuser1;
```

Log in as myuser1 to PostgreSQL and create the database where the FDW will be created with the following command:

```
CREATE DATABASE reports;
```

4. Log in as postgres to the new database (reports) and install the extension:
CREATE EXTENSION postgres_fdw ;

5. Create the server object:

```
CREATE SERVER f_prod FOREIGN DATA WRAPPER postgres_fdw  
OPTIONS (  
  dbname 'pgp',  
  host '192.168.56.101',  
  port '5432',  
);
```

```
ALTER SERVER f_prod OWNER TO myuser1;
```

6. Then, create the user mapping:

```
CREATE USER MAPPING FOR myuser1 SERVER f_prod OPTIONS (  
  user 'myuser1',  
  password 'myuser'  
);
```

7. Next, log in as myuser1 to reports and create the foreign table:

```
CREATE FOREIGN TABLE prod_table (  
  id integer ,  
  name varchar(100)  
)  
SERVER f_prod;  
SELECT * FROM prod_table;
```

***** Using Mysql_FDW *****

1: Open the url :

https://yum.postgresql.org/10/redhat/rhel-7.3-x86_64/

2: Download the Mysql_fdw extension and install it in fedora:

```
mysql_fdw_94-2.0.1-1.f20.x86_64.rpm  
rpm -ivh mysql_fdw_94-2.0.1-1.f20.x86_64.rpm
```

```
3: open postgres :  
# psql  
\c warehouse_db
```

```
4: create the mysql_fdw extension:  
create extension mysql_fdw
```

```
5: create the mysql server :  
CREATE SERVER mysql_server  
    FOREIGN DATA WRAPPER mysql_fdw  
    OPTIONS (host '127.0.0.1');
```

```
6: create user mapping between postgres and mysql :
```

```
CREATE USER MAPPING FOR postgres  
SERVER mysql_server  
OPTIONS (username 'root', password 'mysql');           -- root and 'mysql' is user  
name and password of mysql server and mapped to postgres user of postgresql.
```

```
7: create the foreign table :
```

```
create foreign table a(id int)server mysql_server options(dbname 'a' ,table_name  
'a1');
```

```
           -- 'a' is the database at mysql and 'a1' is the table name which  
will be imported into table a of postgresql.
```

```
8: Match the result between them:
```

```
select * from a;           -- at postgresql
```

```
select * from a1          -- at mysql server , the result shd match.
```