Interactive - 42 - foreign data wrapper

PostgreSQL can access data outside of the database. This is called a foreign data wrapper (FDW). There are all sorts of foreign data wrappers. You can access data in other databases like Oracle or MySQL. You can access data in NoSQL databases like MongoDB. For a full list (of over 30 different kinds of data you can access) https://wiki.postgresql.org/wiki/Foreign_data_wrappers .

Also you can use files like comma separated value (CSV) files through the FDW. There is a file_fdw data wrapper!

To set this up you need to create the extension in PostgreSQL.

```
CREATE EXTENSION if not exists file_fdw;
CREATE SERVER import FOREIGN DATA WRAPPER file_fdw;
```

Then we can use data on the PostgreSQL server and access it. There is a file with data on our server at `/home/uw4820/hw42_data.csv`.

The data in the file is:

```
abc,def
ghi,jkl
abc,jkl
```

We can create a table that matches with this and imports the data.

```
DROP FOREIGN TABLE if exits table1_import ;

CREATE FOREIGN TABLE table1_import (
    col1 text,
    col2 text
) SERVER import OPTIONS ( filename '/home/uw4820/hw42_03.csv', format 'csv' );
```

Now we can use standard selects to access the data.

```
SELECT * from table1_import;

SELECT * from table1_import where col1 = 'abc';
```

The limitation on the selects and the table is that it will do a full table scan (slow) each time you access the data. If you want to create indexes on it you need to use a materialized view or create a new table with the data in it. Then create indexes and other data objects based on that table.

You can use FDW to access other PostgreSQL databases and other databases in different machines.

Another way to access a different database is to use `dblink` in a query. This is more efficient, but can only be used to access database in the same running instance of PotreSQL.

This is an example of using `dblink` - you don't need to run it. It is just an example.

```
SELECT *
FROM   table1 as t1
LEFT   JOIN (
   SELECT *
   FROM dblink('dbname=database2','SELECT t3.id, t3.code FROM table2 as t3') AS t2(id int, c
) AS t4 ON t4.column = t1.column
;
```