Creado por:

Isabel Maniega

Introducción a FugueSQL- PANDAS

Documentación: https://fugue-tutorials.readthedocs.io/tutorials/fugue_sql/index.html

In [11]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df = pd.DataFrame({"col1": [1, 2, 3, 4], "col2": ["a", "b", "c", "c"]})
df
```

Out[2]:

	col1	col2
0	1	a
1	2	b
2	3	c
3	4	c

SQL

SELECT * FROM df WHERE col2="c"

print con esa información

In [3]:
```
# pandas

df[df.col2 == "c"]
```

Out[3]:

	col1	col2
2	3	c
3	4	c

FugueSQL

Permiten combinar Python y comandos SQL

Eso da la flexibilidad de utilizarlo con Jupyter o con script Python

In [4]:
```
# pip install fugue[sql]
```

Podemos ejecutar distintas partes para ejecución de motores de busqueda (engine), podemos usarlo con Spark o Dask:

- *pip* install fugue[sql, spark]
- *pip* install fugue[sql, dask]
- *pip* install fugue[all]

FugueSQL en notebook necesitamos instalar un extensión para poder gestionar los dataframe:

In [5]:
```
from fugue_notebook import setup
setup()
```

In [6]:
```
df
```

Out[6]:

	col1	col2
0	1	a
1	2	b
2	3	c
3	4	c

In [7]:
```
%%fsql

SELECT *
FROM df
WHERE col2="c"
PRINT
```

```
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
```

	col1	col2
0	3	c
1	4	c

schema: col1:long,col2:str

AGRUPAR INFORMACIÓN

- GROUP BY

In [8]:
```
%%fsql

SELECT col2, AVG(col1) AS avg_col1
FROM df
GROUP BY col2
PRINT
```

```
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
```

	col2	avg_col1
0	a	1.0
1	b	2.0
2	c	3.5

schema: col2:str,avg_col1:double

DROP

In [9]:
```
df
```

Out[9]:

	col1	col2
0	1	a
1	2	b
2	3	c
3	4	c

In [10]:
```
%%fsql

df4 = DROP COLUMNS col2 IF EXISTS FROM df
PRINT df4
```

	col1
0	1
1	2
2	3
3	4

schema: col1:long

NULL PARAMS

In [12]:
```
null_df = pd.DataFrame({"col1": [np.nan, np.nan, 1],
                        "col2": [2, 3, np.nan]})
null_df
```

Out[12]:

	col1	col2
0	NaN	2.0
1	NaN	3.0
2	1.0	NaN

In [13]:
```
%%fsql

df1 = FILL NULLS PARAMS col1:10, col2:20 FROM null_df
PRINT df1
```

	col1	col2
0	10.0	2.0
1	10.0	3.0
2	1.0	20.0

schema: col1:double,col2:double

SAMPLE

In [14]:
```
df
```

Out[14]:

	col1	col2
0	1	a
1	2	b
2	3	c
3	4	c

In [15]:
```
%%fsql

df3 = SAMPLE 50 PERCENT SEED 1 FROM df
PRINT df3
```

	col1	col2
0	4	c
1	3	c

schema: col1:long,col2:str

In [16]:
```
%%fsql

df2 = SAMPLE 2 ROWS SEED 42 FROM df
PRINT df2
```

	col1	col2
0	2	b
1	4	c

schema: col1:long,col2:str

OTRA FORMA

In [20]:
```
from fugue_sql import fsql

input_df = pd.DataFrame({"price": [2, 1, 3],
                         "fruit": ["apple", "banana", "orange"]})

input_df
```

Out[20]:

	price	fruit
0	2	apple
1	1	banana
2	3	orange

In [23]:
```
query = """
SELECT price, fruit FROM input_df
WHERE price > 1
PRINT
"""

fsql(query).run()
```

```
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
```

	price	fruit
0	2	apple
1	3	orange

schema: price:long,fruit:str

Out[23]:
```
DataFrames()
```

Transform: llamar a una función

In [30]:
```
df_5 = pd.DataFrame({"number": [0, 1],
                     "word": ["hello", "word"]})

df_5
```

Out[30]:

	number	word
0	0	hello
1	1	word

In [33]:
```
import re
from typing import Iterable, Dict, Any

# schema: *, vowel_count:int, consonant_count:int
def letter_count(df:Iterable[Dict[str,Any]]) -> Iterable[Dict[str,Any]]:
    for row in df:
        row['vowel_count'] = len(re.findall(r'[aeiou]', row['word'], flags=re.IGNORECASE))
        space_count = len(re.findall(r'[ ]', row['word'], flags=re.IGNORECASE))
        row['consonant_count'] = len(row['word']) - row['vowel_count'] - space_count
        yield row
```

In [35]:
```
%%fsql

SELECT *
FROM df_5
WHERE number=0
TRANSFORM USING letter_count
PRINT
```

```
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
ANTLR runtime and generated code versions disagree: 4.11.1!=4.10.1
```

	number	word	vowel_count	consonant_count
0	0	hello	2	3

schema: number:long,word:str,vowel_count:int,consonant_count:int

Creado por:

Isabel Maniega