Sprint 2 Review

Connecting to Quandl

Jon Honda

December 7, 2017

devleague

What is...

Quandl

The world's most powerful data lives on Quandl.

The premier source for financial, economic, and alternative datasets, serving investment professionals. Quandl's platform is used by over 250,000 people, including analysts from the world's top hedge funds, asset managers and investment banks. More on what we do.

Alternative Data

We bring undiscovered data from non-traditional publishers to investors seeking unique, predictive insights. We leverage exclusive relationships to deliver these alpha-generating datasets to our customers.

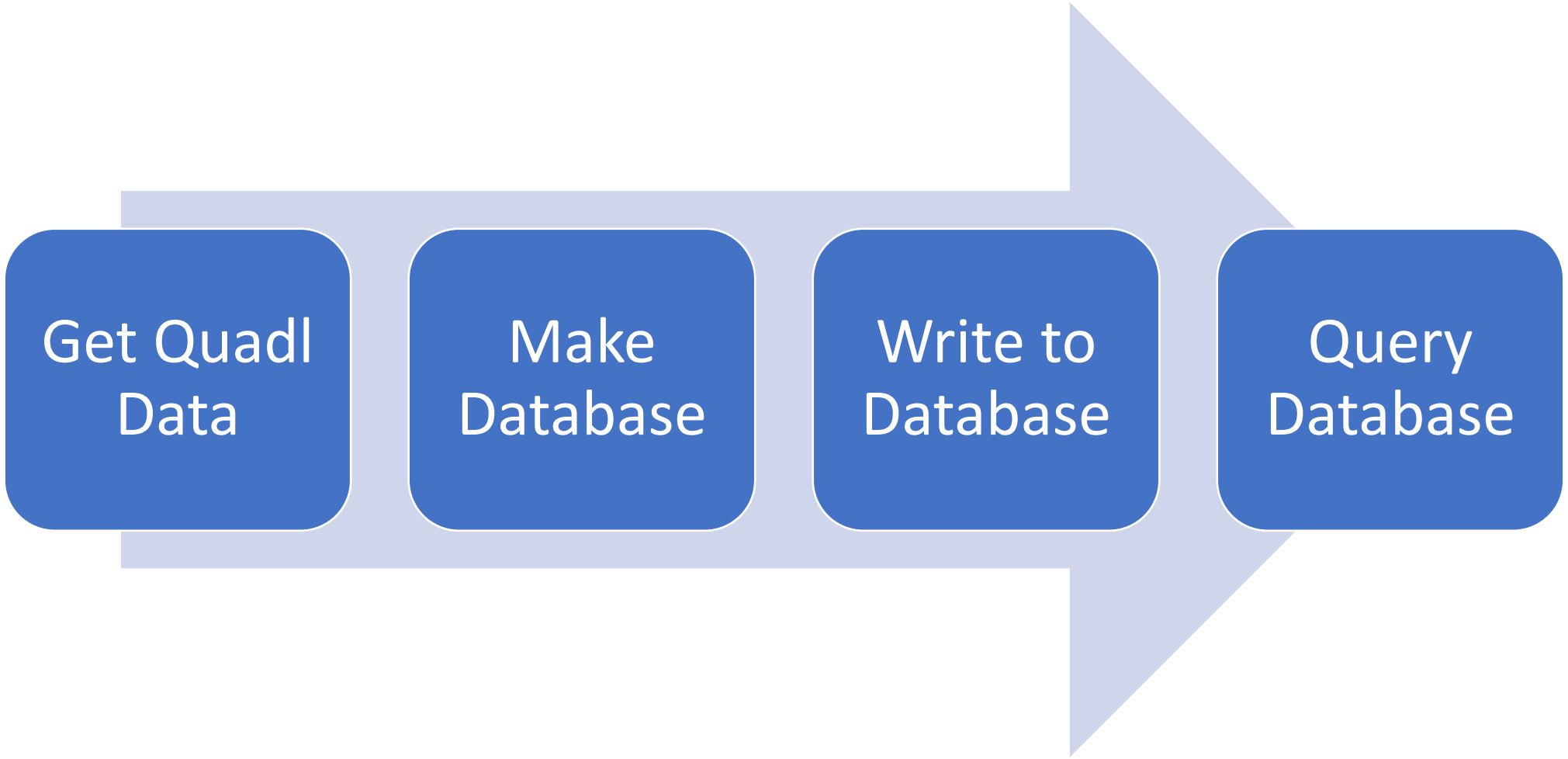
LEARN MORE →

Core Financial Data

Quandl delivers market data from hundreds of sources via API, or directly into Python, R, Excel and many other tools. Get the data you need in the format you want.

SEARCH DATA →

Sprint Objectives



Get Quandl Data

Invoke quandl library

Write to variable

```
####get quandl data:
import quandl
print("getting quandl data....")
quandl.ApiConfig.api_key= "L8zt1AhNiQi4gufsXy4g"
mydata = quandl.get("LME/PR_CO")
```

Use Quandl API to
get cobalt price data set

devleague

Get Quadl Data → Make Database → Write to Database → Query Database

Python code…

```
#get quandl data:
import quandl
print("getting quandl data....")
quandl.ApiConfig.api_key= "L8zt1AhNiQi4gufsXy4g"
mydata = quandl.get("LME/PR_CO")
```

Date	Value
1947-01-01	243.080
1947-04-01	246.267
1947-07-01	250.115
1947-10-01	260.309
1948-01-01	266.173
1948-04-01	272.897
1948-07-01	279.497
1948-10-01	280.656

Pandas Data Type

data as “tables” of rows & columns

Data Types

```
####get quandl data:
import quandl
print("getting quandl data....")
quandl.ApiConfig.api_key= "L8zt1AhNiQi4gufsXy4g"
mydata = quandl.get("LME/PR_CO")
```

Pandas

Date	Value
1947-01-01	243.080
1947-04-01	246.267
1947-07-01	250.115
1947-10-01	260.309
1948-01-01	266.173
1948-04-01	272.897
1948-07-01	279.497
1948-10-01	280.656

datetime

64 bit float

*Different Data Types = Different Abilities

Python code…

```
import sqlite3
import os
db = sqlite3.connect(':memory:') #### allocate memory
os.remove('_jonhonda_dat\quandla') #### remove existing db if it exists
db = sqlite3.connect('_jonhonda_dat\quandlala')  #### make a database
 #### build table:
cursor = db.cursor()
cursor.execute ('''CREATE TABLE COPrices(id INTEGER PRIMARY KEY, Date TEXT, Val FLOAT)''')
db.commit()
```

Directory

DB File

Makes the database table.
The language is SQL

devleague

Get Quadl Data | **Make DB** | Write to DB | Query DB

Make Database

```
import sqlite3
import os
db = sqlite3.connect(':memory:') #### allocate memory
os.remove('_jonhonda_dat\quandlala') #### remove existing db if it exists
db = sqlite3.connect('_jonhonda_dat\quandlala')  #### make a database
```

```
#### build table:
cursor = db.cursor()
cursor.execute ('''CREATE TABLE COPrices(id INTEGER PRIMARY KEY, Date TEXT, Val FLOAT)''')
db.commit()
```

Defines Field

Table Name

Field Name

Data Type

devleague

Get Quadl Data → **Make DB** → Write to DB → Query DB

Incompatible Data Types!

```
#### write to DB:
import sqlite3
import os
db = sqlite3.connect(':memory:') #### allocate memory
os.remove('_jonhonda_dat\quandlala') #### remove existing db if it exists
db = sqlite3.connect('_jonhonda_dat\quandlala')  #### make a database
 #### build table:
cursor = db.cursor()
cursor.execute ('''CREATE TABLE COPrices(id INTEGER PRIMARY KEY, Date TEXT, Val FLOAT)''')
db.commit()
```

datetime

64 bit float

Date	Value
1947-01-01	243.080
1947-04-01	246.267
1947-07-01	250.115
1947-10-01	260.309
1948-01-01	266.173
1948-04-01	272.897
1948-07-01	279.497
1948-10-01	280.656

devleague

Get Quadl Data → Make DB → **Write to DB** → Query DB

Solve Incompatibility

mydata

Date	Value
1947-01-01	243.080
1947-04-01	246.267
1947-07-01	250.115
1947-10-01	260.309
1948-01-01	266.173
1948-04-01	272.897
1948-07-01	
1948-10-01	280.656

rowi → 1948-07-01 → Change type → 1948-07-01

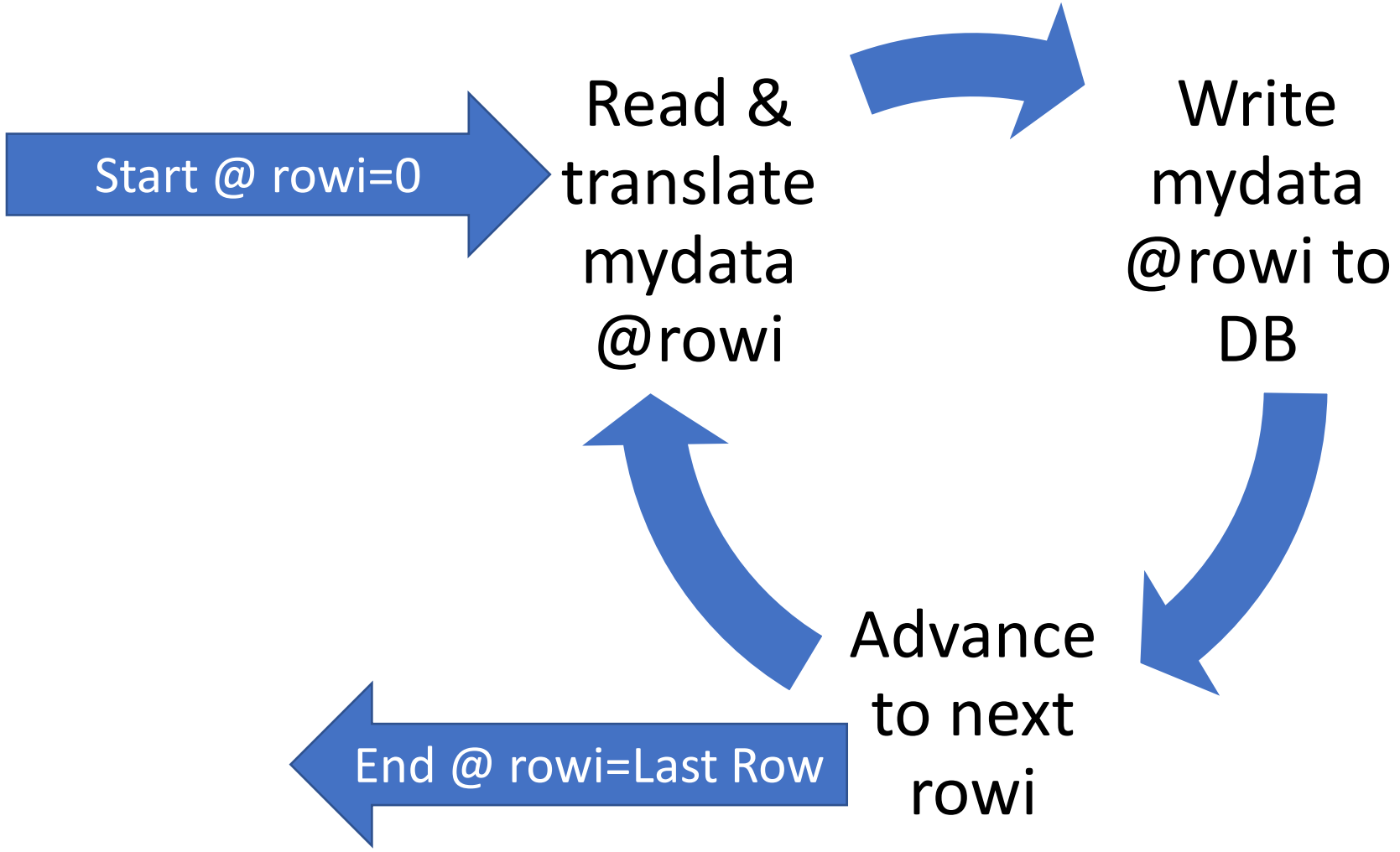
Looks the same, but computer reads it differently!

```
myDate = mydata.index[rowi].strftime('%m %d %Y')
```

devleague

Get Quadl Data | Make DB | **Write to DB** | Query DB

Write all mydata to Database

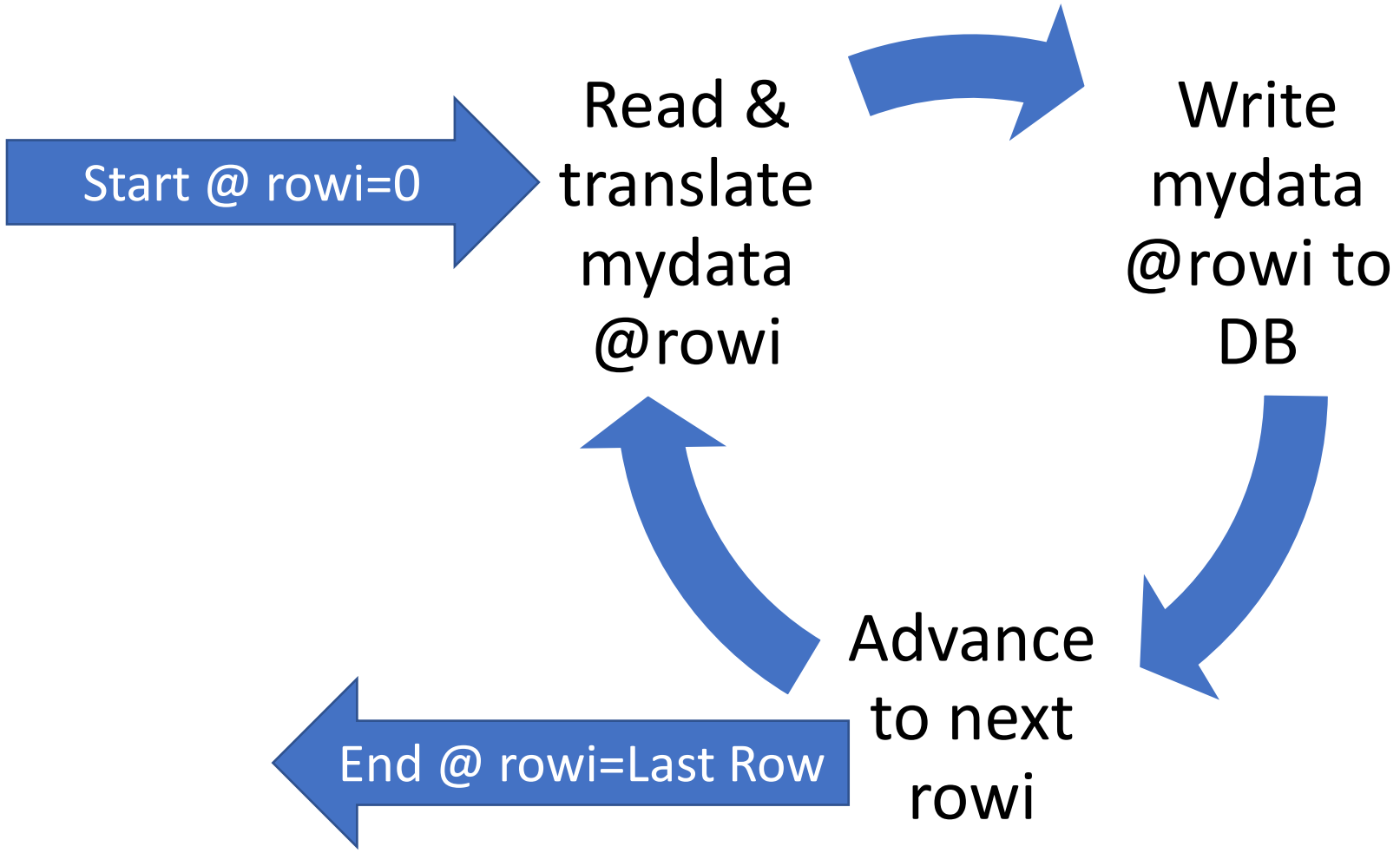


rows

mydata

	Date	Value
0	1947-01-01	243.080
1	1947-04-01	246.267
2	1947-07-01	250.115
3	1947-10-01	260.309
4	1948-01-01	266.173
5	1948-04-01	272.897
6	1948-07-01	279.497
7	1948-10-01	280.656

Write all mydata to Database



rows

mydata

	Date	Value
0	1947-01-01	243.080
1	1947-04-01	246.267
2	1947-07-01	250.115
3	1947-10-01	260.309
4	1948-01-01	266.173
5	1948-04-01	272.897
6	1948-07-01	279.497
7	1948-10-01	280.656

```
for rowi in range(0,len(mydata.index)):
    myVal = mydata.iloc[rowi][0]
    myDate = mydata.index[rowi].strftime('%m %d %Y')
    cursor.execute ('''INSERT INTO COPrices(Date, Val) VALUES(?,?)''',(myDate,myVal))
    db.commit()
```

Read & Translate

Write to DB

Write all mydata to Database

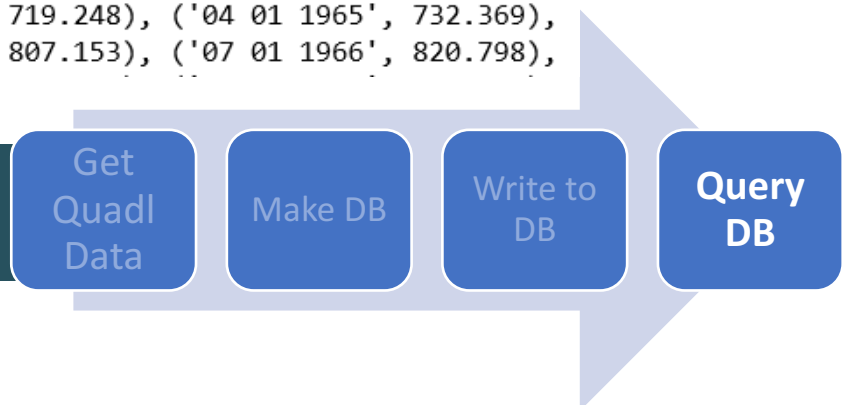
```
cursor.execute ('''INSERT INTO COPrices(Date, Val) VALUES(?,?)''',(myDate,myVal))
```

Writes data at rowi to DB
The language is SQL

Get Data from Database

```
cursor.execute('''SELECT Date, StockValue FROM StockPrices''')
print(cursor.fetchall())
```

```
[('01 01 1947', 243.08), ('01 01 1947', 243.08), ('04 01 1947', 246.267), ('07 01 1947', 250.115), ('10 01 1947', 260.309),
('01 01 1948', 266.173), ('04 01 1948', 272.897), ('07 01 1948', 279.497), ('10 01 1948', 280.656), ('01 01 1949', 275.37),
('04 01 1949', 271.692), ('07 01 1949', 273.262), ('10 01 1949', 270.984), ('01 01 1950', 281.209), ('04 01 1950', 290.735),
('07 01 1950', 308.51), ('10 01 1950', 320.32), ('01 01 1951', 336.372), ('04 01 1951', 344.455), ('07 01 1951', 351.774),
('10 01 1951', 356.579), ('01 01 1952', 360.195), ('04 01 1952', 361.414), ('07 01 1952', 368.084), ('10 01 1952', 381.241),
('01 01 1953', 388.472), ('04 01 1953', 392.259), ('07 01 1953', 391.696), ('10 01 1953', 386.521), ('01 01 1954', 385.924),
('04 01 1954', 386.716), ('07 01 1954', 391.596), ('10 01 1954', 400.348), ('01 01 1955', 413.753), ('04 01 1955', 422.226),
('07 01 1955', 430.925), ('10 01 1955', 437.787), ('01 01 1956', 440.491), ('04 01 1956', 446.771), ('07 01 1956', 451.983),
('10 01 1956', 461.278), ('01 01 1957', 470.578), ('04 01 1957', 472.835), ('07 01 1957', 480.315), ('10 01 1957', 475.681),
('01 01 1958', 468.353), ('04 01 1958', 472.786), ('07 01 1958', 486.653), ('10 01 1958', 500.38), ('01 01 1959', 511.063),
('04 01 1959', 524.241), ('07 01 1959', 525.196), ('10 01 1959', 529.322), ('01 01 1960', 543.347), ('04 01 1960', 542.697),
('07 01 1960', 546.012), ('10 01 1960', 541.063), ('01 01 1961', 545.949), ('04 01 1961', 557.43), ('07 01 1961', 568.228),
('10 01 1961', 581.624), ('01 01 1962', 595.176), ('04 01 1962', 602.58), ('07 01 1962', 609.575), ('10 01 1962', 613.132),
('01 01 1963', 622.679), ('04 01 1963', 631.835), ('07 01 1963', 644.96), ('10 01 1963', 654.84), ('01 01 1964', 671.149),
('04 01 1964', 680.757), ('07 01 1964', 692.807), ('10 01 1964', 698.424), ('01 01 1965', 719.248), ('04 01 1965', 732.369),
('07 01 1965', 750.184), ('10 01 1965', 773.104), ('01 01 1966', 797.328), ('04 01 1966', 807.153), ('07 01 1966', 820.798),
```

Sprint Objectives

Get Quadl Data

Make Database

Write to Database

Query Database

devleague

(Python)

```python
####get quandl data:
import quandl
print("getting quandl data....")
quandl.ApiConfig.api_key= "L8zt1AhNiQi4gufsXy4g"
mydata = quandl.get("LME/PR_CO")

#### write to DB:
import sqlite3
import os
db = sqlite3.connect(':memory:') #### allocate memory
os.remove('_jonhonda_dat\quandlala') #### remove existing db if it exists
db = sqlite3.connect('_jonhonda_dat\quandlala')  #### make a database
 #### build table:
cursor = db.cursor()
cursor.execute ('''CREATE TABLE COPrices(id INTEGER PRIMARY KEY, Date TEXT, Val FLOAT)''')
db.commit()

#### iterate through results writing to database:
import datetime
print ("writing to database...")
rowi = 0
for rowi in range(0,len(mydata.index)):
    myVal = mydata.iloc[rowi][0]
    myDate = mydata.index[rowi].strftime('%m %d %Y')
    cursor.execute ('''INSERT INTO COPrices(Date, Val) VALUES(?,?)''',(myDate,myVal))
    db.commit()
    if rowi%250==0:
        print("still writing data...Row: ", rowi)

#### get data from db
print ("get data from db")
cursor.execute('''SELECT * FROM COPrices''')
print(cursor.fetchall())
db.close()
```

Fin

devleague