Performance comparison between `Pandas+AWS Wrangler` versus `Pandas+PyAthena+SQLAlchemy`

In [1]:
```
import pandas as pd
import sqlalchemy
import awswrangler as wr
```

Query definition

This query returns more than 2 million rows

In [2]:
```
query="SELECT * FROM large_table"
```

Database definition

In [3]:
```
region_name="us-east-1"
schema_name="robson_valuation"
work_group="mlops"
```

Query via AWS Wrangler

In [4]:
```
%%time

aws_wrangler_df = wr.athena.read_sql_query(
    sql             = query,
    database        = schema_name,
    workgroup       = work_group,
    ctas_approach   = False
)

aws_wrangler_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2009322 entries, 0 to 2009321
Data columns (total 19 columns):
 #   Column                                Dtype
```

```
---  ------                                  -----
 0   unit_id                                 object
 1   private_area                            float64
 2   floor                                   Int32
 3   parkings                                Int64
 4   building_year                           Int32
 5   latitude                                object
 6   longitude                               object
 7   neighbourhood                           string
 8   city                                    string
 9   last_transaction_value                  float64
 10  last_transaction_date                   object
 11  adj_value                               float64
 12  adj_value_per_meter                     float64
 13  last_transaction_date_1p                datetime64[ns]
 14  last_transaction_value_1p               float32
 15  last_transaction_value_1p_per_meter     float64
 16  last_transaction_date_3p                datetime64[ns]
 17  last_transaction_value_3p               float32
 18  last_transaction_value_3p_per_meter     float64
dtypes: Int32(2), Int64(1), datetime64[ns](2), float32(2), float64(6), object(4), string(2)
memory usage: 266.4+ MB
CPU times: user 14 s, sys: 3.14 s, total: 17.2 s
Wall time: 1min 34s
```

Query via PyAthena+SQLAlchemy

In [6]:

```
%%time

conn_str = "awsathena+rest://:@athena.{region_name}.amazonaws.com:443/{schema_name}?work_group={work_group}&compressi

pyathena_sqlalchemy_conn = sqlalchemy.create_engine(
    url = conn_str.format(
        region_name  = region_name,
        schema_name  = schema_name,
        work_group   = work_group
    ),
    echo_pool = True,
)

pyathena_df = pd.read_sql_query(query,con=pyathena_sqlalchemy_conn)

pyathena_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2009322 entries, 0 to 2009321
Data columns (total 19 columns):
```

```
 #   Column                             Dtype
---  ------                             -----
 0   unit_id                            float64
 1   private_area                       float64
 2   floor                              int64
 3   parkings                           int64
 4   building_year                      int64
 5   latitude                           float64
 6   longitude                          float64
 7   neighbourhood                      object
 8   city                               object
 9   last_transaction_value             float64
 10  last_transaction_date              object
 11  adj_value                          float64
 12  adj_value_per_meter                float64
 13  last_transaction_date_1p           datetime64[ns]
 14  last_transaction_value_1p          float64
 15  last_transaction_value_1p_per_meter  float64
 16  last_transaction_date_3p           datetime64[ns]
 17  last_transaction_value_3p          float64
 18  last_transaction_value_3p_per_meter  float64
dtypes: datetime64[ns](2), float64(11), int64(3), object(3)
memory usage: 291.3+ MB
CPU times: user 2min 46s, sys: 11.7 s, total: 2min 58s
Wall time: 16min 37s
```