

Project_20180824

August 24, 2018

```
In [19]: import datetime
import logging
import os
from subprocess import call

import numpy as np
import pandas as pd
from sqlalchemy import create_engine

logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
logger = logging.getLogger(__name__)

# Postgres
call("bash /home/Egorov_load_project.sh", shell=True)

engine = create_engine('postgresql://postgres:@{}@postgres_host')

In [49]: # Postgres pandas
accounts = pd.read_sql('select * from public.v_accounts ', engine)
logger.info("select view ")
print(accounts.head())
logger.info("{}".format(ratings.dtypes))

2018-08-24 10:50:12,578 : INFO : select view
2018-08-24 10:50:12,589 : INFO : Series([], dtype: object)

   account_number  account_name  client_name  report_date  acc_value \
0  42301000000000000001      ..  2015-01-01    -10000.0
1  42301000000000000002      ..  2015-01-01    -50000.0
2  45201000000000000003      ..  2015-01-01     -1000.0
3  45201000000000000004      ..  2015-01-01    580000.0
4  45201000000000000005      ..  2015-01-01    400000.0

department_name
0
1
```

2
3
4

```
In [155]: #      SQL 5
```

```
account_count = accounts.groupby(['client_name']).agg({'acc_value': ['count']})#.filt
```

```
In [156]: print(account_count.head())
```

```
      acc_value  
      count  
client_name  
..         2  
..         2  
..         1
```

```
In [159]: #      - SQL 9
```

```
window_sums = accounts.assign(  
    cumsum =accounts.groupby(['client_name'])['acc_value']  
    #.agg('sum')  
    .cumsum()# .rank(method='dense', ascending=True).astype(int)  
) .query('cumsum >0').sort_values(['client_name', 'acc_value'])  
print(window_sums.head())
```

```
#pd.merge(left,right,on=['id','subject_id'])
```

```
      account_number  account_name  client_name  report_date  acc_value  \  
3  45201000000000000004      ..  2015-01-01    580000.0  
4  45201000000000000005      ..  2015-01-01    400000.0
```

```
      department_name  cumsum  
3          570000.0  
4          350000.0
```

```
In [157]: account_count.to_csv("account_count.csv", sep='\t', encoding='utf-8')
```

```
In [160]: window_sums.apply(lambda x: x).to_csv("window_sums.csv", sep='\t', encoding='utf-8')
```