

AtividadeSQL

November 6, 2020

1 Atividade Spark SQL

Considerando o dataset detalhado a seguir, extraia o conjunto de informações solicitadas.

1.0.1 Dataset dados da BOVESPA

- Arquivo disponível em /home/dados/bovespa/bovespa.csv
- Dados relativos a bovespa, a bolsa de valores no Brasil
- ~1.3GB
- 8.1M de instâncias

#	Nome do campo	Descrição
0	RegisterType	Fixo '1'
1	TradingDate	Data do pregão
2	BDICode	Utilizado para classificar os papéis na emissão do boletim diário de informações
3	NegotiationCode	Código de negociação do papel
4	MarketType	Cód. Do mercado em que o papel está cadastrado
5	TradeName	Nome resumido da empresa emissora do papel
6	Specification	Especificação do Papel
7	ForwardMarketTermInDays	Prazo em dias do mercado a termo
8	Currency	Moeda de referência
9	OpeningPrice	Preço de abertura do papel no pregão
10	MaxPrice	Preço máximo do papel no pregão
11	MinPrice	Preço mínimo do papel no pregão
12	MeanPrice	Preço médio do papel no pregão
13	LastTradePrice	Preço do último negócio do papel no pregão
14	BestPurshaseOrderPrice	Preço da melhor oferta de compra do papel no mercado
15	BestPurshaseSalePrice	Preço da melhor oferta de venda do papel no mercado
16	NumborOfTrades	Número de negócios efetuados com o papel no pregão
17	NumberOfTradedStocks	Quantidade total de títulos negociados neste papel

#	Nome do campo	Descrição
18	VolumeOfTradedStocks	Volume total de títulos negociados neste papel
19	PriceForOptionsMarketOrSecondaryTermMarket	Preço de exercício para o mercado de opções ou valor do contrato para o mercado
20	PriceCorrectionsForOptionsMarketOrSecondaryTermMarket	Implicado ou correção de preços de exercícios ou valores de contrato
21	DueDateForOptionsMarketOrSecondaryTermMarket	Data de vencimento para os mercados de opções
22	FactorOfPaperQuotation	Fator de cotação do papel
23	PointsInPriceForOptionsMarketReferencedInDollarOrSecondaryTermMarket	Pontos para opções referenciadas em dólar
24	ISINOrInternCode	Código do papel no sistema ISIN
25	DistributionNumber	Número de distribuição do papel

Informações a serem extraídas:

1. Quantidade de dias com operações da PETR4 (NegociationCode)
2. Maior valor (MaxPrice) histórico por ação (NegociationCode)
3. Maior valor (MaxPrice) histórico da PETR4 (NegociationCode)
4. Dia ('TradingDate') com a maior quantidade de papeis (NegociationCode) operados
5. Dia ('TradingDate') da semana com a maior quantidade de papeis (NegociationCode) operados
6. Maior lucro histórico de um papel (NegociationCode) na bovespa (MaxPrice - OpeningPrice)
7. Maior prejuízo histórico de um papel (NegociationCode) na bovespa (OpeningPrice - Last-TradePrice)
8. Moeda (Currency) com mais operações
9. Papel (NegociationCode) operado em CZ (Currency) com maior quantidade de operações
10. Papel (NegociationCode) operado em CZ (Currency) com maior valor médio das operações (MeanPrice)
11. Media do preço médio (MeanPrice), mínimo (MinPrice) e máximo (MaxPrice) anual (TradingDate) das ações da PETR4 (NegociationCode)
12. Preço médio (MeanPrice) anual (TradingDate) das ações da PETR4 (NegociationCode)
13. Preço médio (MeanPrice) anual (TradingDate) das 10 ações (NegociationCode) com mais operações na bovespa
14. Desvio Padrão anual do preço médio (MeanPrice) da ação da PETR4 (NegociationCode)
15. Desvio Padrão anual do preço médio (MeanPrice) das 10 ações (NegociationCode) com mais operações na bovespa
16. Preço médio (MeanPrice) anual (TradingDate) das ações (NegociationCode) com a maior quantidade de operações de acordo com a moeda (Currency)

Dicas: - Crie uma célula (Insert -> Insert Cell Below) para cada informação solicitada - A análise deve ser feita sobre os dados do HDFS - Inicialize o seu cluster executando o script em: Desktop/ambientes/spark/inicializar.sh - Acesse o seu cluster executando o script em: Desktop/ambientes/spark/abrir_navegador.sh

```
[1]: import os
os.environ['PYSPARK_PYTHON'] = '/usr/bin/python3'

from pyspark.sql import SparkSession

sc = SparkSession \
    .builder \
    .master('spark://spark-master:7077') \
    .config('spark.executor.memory', '1g') \
    .getOrCreate()
```

```
[2]: df = sc.read \
    .option('delimiter', ',') \
    .option('header', 'true') \
    .option('inferSchema', 'true') \
    .csv('hdfs://namenode:9000/bovespa.csv')
```

```
[3]: df.printSchema()
```

```
root
|-- RegisterType: integer (nullable = true)
|-- TradingDate: integer (nullable = true)
|-- BDICode: double (nullable = true)
|-- NegotiationCode: string (nullable = true)
|-- MarketType: integer (nullable = true)
|-- TradeName: string (nullable = true)
|-- Specification: string (nullable = true)
|-- ForwardMarketTermInDays: string (nullable = true)
|-- Currency: string (nullable = true)
|-- OpeningPrice: double (nullable = true)
|-- MaxPrice: double (nullable = true)
|-- MinPrice: double (nullable = true)
|-- MeanPrice: double (nullable = true)
|-- LastTradePrice: double (nullable = true)
|-- BestPurshaseOrderPrice: double (nullable = true)
|-- BestPurshaseSalePrice: double (nullable = true)
|-- NumborOfTrades: double (nullable = true)
|-- NumberOfTradedStocks: double (nullable = true)
|-- VolumeOfTradedStocks: double (nullable = true)
|-- PriceForOptionsMarketOrSecondaryTermMarket: double (nullable = true)
|-- PriceCorrectionsForOptionsMarketOrSecondaryTermMarket: double (nullable =
true)
|-- DueDateForOptionsMarketOrSecondaryTermMarket: double (nullable = true)
|-- FactorOfPaperQuotatuion: double (nullable = true)
|-- PointsInPriceForOptionsMarketReferencedInDollarOrSecondaryTerm: double
(nullable = true)
|-- ISINOrInternCode: string (nullable = true)
```

```
-- DistributionNumber: double (nullable = true)
```

```
[4]: df.createOrReplaceTempView('bovespa')
```

```
[5]: sc.sql('select NegociationCode from Bovespa').show(4)
```

```
+-----+
|NegociationCode|
+-----+
|          ACE 2|
|          AVI 2|
|          CRA 2|
|          AGR 11|
+-----+
only showing top 4 rows
```

```
[6]: sc.sql('select NegociationCode, max(MaxPrice) as maximo from bovespa group by
↳NegociationCode').show(10)
```

```
+-----+-----+
|NegociationCode|  maximo|
+-----+-----+
|          FGO 4| 240000.0|
|          VAG 3|5395600.0|
|          SUL 4| 310000.0|
|          OPM 82|    64.0|
|          OTC 6|1000000.0|
|          VSP 4|   106.0|
|          ROS 3|  40000.0|
|          OTV 85| 19791.0|
|          OTC 93| 15000.0|
|          OTC 96| 10100.0|
+-----+-----+
only showing top 10 rows
```

```
[7]: sc.sql('select TradingDate, NegociationCode, MaxPrice from bovespa where
↳NegociationCode == \'PETR4\']").show(5)
```

```
+-----+-----+-----+
|TradingDate|NegociationCode|MaxPrice|
+-----+-----+-----+
| 19980316|          PETR4| 27600.0|
| 19980317|          PETR4| 27800.0|
| 19980318|          PETR4| 27900.0|
| 19980319|          PETR4| 27700.0|
```

19980320	PETR4	28000.0
----------	-------	---------

only showing top 5 rows

```
[8]: df.select('TradingDate', 'NegociationCode', 'MaxPrice')\
      .filter(df.NegociationCode == 'PETR4')\
      .orderBy(df.MaxPrice)\
      .show(5)
```

TradingDate	NegociationCode	MaxPrice
20160211	PETR4	427.0
20160126	PETR4	431.0
20160210	PETR4	441.0
20160120	PETR4	450.0
20160212	PETR4	451.0

only showing top 5 rows

```
[9]: import pyspark.sql.functions as func
      from pyspark.sql.functions import udf
      from pyspark.sql.types import StringType

      udfNegociation2 = udf(lambda negotiation: negotiation + negotiation,
                             StringType())

      df.select(func.col('NegociationCode'), func.col('MaxPrice'))\
        .filter(func.col('NegociationCode')== 'PETR4')\
        .groupBy(func.col('NegociationCode'))\
        .agg(func.max(func.col('MaxPrice')).alias('ValorMaximo'),
              func.count(func.col('MaxPrice')).alias('qt'))\
        .withColumn('negociation2', udfNegociation2('NegociationCode'))\
        .show(3)
```

NegociationCode	ValorMaximo	count(MaxPrice AS `qt`)	negociation2
PETR4	52100.0	5391	PETR4PETR4

```
[ ]:
```

```
[10]: #informacao 1
sc.sql('select Count(NegotiationCode) as QNTDdias  from bovespa where
      ↳NegociationCode==\'PETR4\' ').show(5)
```

```
+-----+
|QNTDdias|
+-----+
|    5391|
+-----+
```

```
[11]: #informacao 2
sc.sql('select NegociationCode, max(MaxPrice) as PrecoMaximo from bovespa group
      ↳by NegociationCode order by Negociationcode').show(10)
```

```
+-----+-----+
|NegociationCode|PrecoMaximo|
+-----+-----+
|      A1LX34|    22597.0|
|      A1LX34F|    22597.0|
|      AALC11B|     3717.0|
|      AALC34|    10058.0|
|      AALC34F|    10058.0|
|      AALL34|    18737.0|
|      AALL34F|    19389.0|
|      AALR3|     1976.0|
|      AALR3F|     1999.0|
|      AALR3T|     1956.0|
+-----+-----+
```

only showing top 10 rows

```
[12]: #informacao 3
sc.sql('select NegociationCode, max(MaxPrice) as PrecoMaximo from bovespa
      ↳where NegociationCode==\'PETR4\' group by NegociationCode').show()
```

```
+-----+-----+
|NegociationCode|PrecoMaximo|
+-----+-----+
|      PETR4|    52100.0|
+-----+-----+
```

```
[13]: #informacao 4
sc.sql('select TradingDate, count(NegotiationCode) as NPapers from bovespa
      ↳group by TradingDate order by NPapers desc').show(10)
```

```
+-----+-----+
```

TradingDate	NPapers
20191216	4652
20191212	4241
20191213	4223
20191118	4208
20190819	4135
20191114	4131
20191021	3975
20190916	3954
20191113	3930
20190715	3916

only showing top 10 rows

```
[14]: #informacao 5
import datetime
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

def pegaDiaSemana(x):
    try:
        data = str(x)
        ano, mes, dia = data[0:4], data[4:6], data[6:8]
        return datetime.datetime(int(ano), int(mes), int(dia)).weekday()
    except:
        return '0'

udfPegaDiaSemana = udf(pegaDiaSemana, StringType())

df.select('TradingDate')\
    .withColumn('diaSemana', udfPegaDiaSemana('TradingDate'))\
    .groupBy(func.col('diaSemana'))\
    .count()\
    .show()
```

diaSemana	count
3	1597353
0	1626494
5	3
1	1613747
4	1599650
2	1688444

```
[15]: #informacao 6
sc.sql('select NegociationCode, max(MaxPrice - OpeningPrice) as lucro from
↳bovespa group by NegociationCode order by lucro desc').show(10)
```

NegociationCode	lucro
ANT 3	4.411E8
ARN 2	1.5E8
TEL 4	1.0000008E8
RHE 4	1.0E8
ARN 4	9.0E7
LIGH11F	8.005E7
BDL 4	5.0E7
ANT 4	4.0E7
CCC 2	3.322E7
POL 3	3.0E7

only showing top 10 rows

```
[16]: #informacao 7
sc.sql('select NegociationCode, max(OpeningPrice - LastTradePrice) as prejuizo
↳from bovespa group by NegociationCode order by prejuizo desc').show(10)
```

NegociationCode	prejuizo
ANT 3	1.45E8
SLE 2	1.0E8
RHE 4	5.0E7
ARN 4	4.0E7
ANT 4	3.5E7
BDL 4	3.0E7
LAM 3	2.7E7
CCC 2	2.5E7
BDL 2	2.2E7
BMK 3	1.9999999E7

only showing top 10 rows

```
[17]: #informacao 8
sc.sql('select Currency, Count(Currency) as QNTD from bovespa group by
↳Currency').show()
```

Currency	QNTD
----------	------

CZ\$	458842	
NCZ\$	172625	
R\$	6995662	
CR\$	498562	

```
[18]: #informacao 9
sc.sql('select Currency, NegociationCode, Count(NegociationCode) as QNTD from
↳bovespa where Currency==\'CZ$\' group by Currency, NegociationCode order by
↳QNTD desc').show(10)
```

Currency	NegociationCode	QNTD
CZ\$	PMA 2	2002
CZ\$	PET 2	1993
CZ\$	SHA 2	1755
CZ\$	BB 2	1751
CZ\$	BES 2	1749
CZ\$	FNV 6	1730
CZ\$	BBD 4	1720
CZ\$	TRB 2	1679
CZ\$	AVI 2	1648
CZ\$	VAG 2	1643

only showing top 10 rows

```
[19]: #informacao 10
sc.sql('select Currency, NegociationCode, max(MeanPrice) as MaiorValorMedio
↳from bovespa where Currency==\'CZ$\' group by Currency, NegociationCode
↳order by MaiorValorMedio desc').show(10)
```

Currency	NegociationCode	MaiorValorMedio
CZ\$	TLS 6	3.7975E7
CZ\$	LOB 2	8000000.0
CZ\$	ARN 2	8000000.0
CZ\$	CCC 2	7000000.0
CZ\$	RHE 4	6902439.0
CZ\$	EBC 4	6500000.0
CZ\$	CON 2	6500000.0
CZ\$	AHE 3	4000000.0
CZ\$	ARN 4	4000000.0
CZ\$	MES 1	3500000.0

```
+-----+-----+-----+
only showing top 10 rows
```

```
[20]: #informacao 11
sc.sql('select NegociationCode, SUBSTRING(TradingDate, 1, 4) as Year,
      ↳Avg(MinPrice), Avg(MeanPrice), Avg(MaxPrice) from bovespa where
      ↳NegociationCode==\'PETR4\' group by NegociationCode, Year order by Year
      ↳desc')..show(20)
```

```
+-----+-----+-----+-----+-----+
|NegociationCode|Year|      avg(MinPrice)|      avg(MeanPrice)|      avg(MaxPrice)|
+-----+-----+-----+-----+-----+
|      PETR4|2019|2693.6169354838707|      2724.5|2755.3669354838707|
|      PETR4|2018| 2065.718367346939|2102.4489795918366| 2140.804081632653|
|      PETR4|2017|1441.2032520325204| 1459.060975609756|1477.7642276422764|
|      PETR4|2016|1054.2610441767067|1077.4939759036145| 1101.140562248996|
|      PETR4|2015| 957.8373983739838| 980.4349593495934| 1004.719512195122|
|      PETR4|2014|1636.7701612903227|1669.7298387096773|1704.1370967741937|
|      PETR4|2013|1800.5725806451612|1825.3064516129032|1852.4879032258063|
|      PETR4|2012|2106.9065040650407| 2134.691056910569|2163.5040650406504|
|      PETR4|2011| 2344.152610441767|2369.9236947791164| 2396.610441767068|
|      PETR4|2010| 2940.068825910931| 2975.072874493927| 3011.076923076923|
|      PETR4|2009|3141.5853658536585| 3183.650406504065|3225.0243902439024|
|      PETR4|2008| 4705.016064257028| 4806.373493975903| 4906.417670682731|
|      PETR4|2007| 5357.355102040817| 5442.975510204082| 5522.302040816327|
|      PETR4|2006| 4285.987804878048|4344.0609756097565| 4400.654471544715|
|      PETR4|2005| 7993.024096385542| 8094.469879518072| 8192.240963855422|
|      PETR4|2004| 8279.132530120482| 8383.638554216868| 8488.269076305221|
|      PETR4|2003|      5547.004|      5613.908|      5681.28|
|      PETR4|2002|4727.7389558232935|4796.7389558232935| 4871.04016064257|
|      PETR4|2001| 5216.276422764227| 5297.70325203252| 5383.422764227642|
|      PETR4|2000|23769.959677419356|24186.137096774193| 24586.6935483871|
+-----+-----+-----+-----+-----+
only showing top 20 rows
```

```
[21]: #informacao 12
# Preço médio (MeanPrice) anual (TradingDate) das ações da PETR4
↳(NegociationCode)

# Vou fazer a média das entradas de PETR4, por exemplo, no ano 2019,
# como [PETR4 ANO Média] para diminuir o número de instâncias Year na tabela.
# Não ficou claro na informação se eu poderia fazer AVG() dos Mean Prices; caso
↳não seja essa
# a intenção, basta remover Avg() da query e dar group by no mean price também.
```

```
sc.sql('select NegociationCode, SUBSTRING(TradingDate, 1, 4) as Year,
↳Avg(MeanPrice) from bovespa where NegociationCode=\'PETR4\' group by
↳NegociationCode, Year order by Year desc').show(20)
```

```
+-----+-----+-----+
|NegociationCode|Year|    avg(MeanPrice)|
+-----+-----+-----+
|      PETR4|2019|          2724.5|
|      PETR4|2018|2102.4489795918366|
|      PETR4|2017| 1459.060975609756|
|      PETR4|2016|1077.4939759036145|
|      PETR4|2015| 980.4349593495934|
|      PETR4|2014|1669.7298387096773|
|      PETR4|2013|1825.3064516129032|
|      PETR4|2012| 2134.691056910569|
|      PETR4|2011|2369.9236947791164|
|      PETR4|2010| 2975.072874493927|
|      PETR4|2009| 3183.650406504065|
|      PETR4|2008| 4806.373493975903|
|      PETR4|2007| 5442.975510204082|
|      PETR4|2006|4344.0609756097565|
|      PETR4|2005| 8094.469879518072|
|      PETR4|2004| 8383.638554216868|
|      PETR4|2003|          5613.908|
|      PETR4|2002|4796.7389558232935|
|      PETR4|2001| 5297.70325203252|
|      PETR4|2000|24186.137096774193|
+-----+-----+-----+
```

only showing top 20 rows

```
[23]: #informacao 13
# Preço médio (MeanPrice) anual (TradingDate) das 10 ações (NegociationCode)
↳com mais operações na bovespa

# Vou fazer a média das entradas de PETR4, por exemplo, no ano 2019,
# como [PETR4 ANO Média] para diminuir o número de instâncias Year na tabela.
# Não ficou claro na informação se eu poderia fazer AVG() dos Mean Prices; caso
↳não seja essa
# a intenção, basta remover Avg() da query e dar group by no mean price também.

top10Acoes = df.select('NegociationCode', 'MeanPrice', 'TradingDate')\
    .groupBy('NegociationCode')\
    .agg(func.count('NegociationCode').alias('qt'))\
    .orderBy(func.desc('qt'))\
    .limit(10)\
```

```

        .select('Negociationcode').collect()

top10Acoes = [str(i).split('=\\')[1].split('\\')[0] for i in top10Acoes] #
↳filtrando.

# Criando uma cláusula where com o array top10:
mywhere3 = ' where '
for i in range(0, len(top10Acoes)):
    mywhere3 += f' NegociationCode==\\'{top10Acoes[i]}\\' '
    if i < len(top10Acoes)-1:
        mywhere3 += ' or '

print("Top 10 ações mais negociadas: ", top10Acoes)

sc.sql('select NegociationCode, SUBSTRING(TradingDate, 1, 4) as Year,
↳Avg(MeanPrice) from bovespa'+mywhere3+' group by NegociationCode, Year order
↳by Year desc, NegociationCode').show(50)

```

Top 10 ações mais negociadas: ['PETR4T', 'VALE5T', 'USIM5T', 'BBAS3T', 'CSNA3T', 'GGBR4T', 'BBDC4T', 'ITSA4T', 'CMIG4T', 'BRKM5T']

	NegociationCode	Year	avg(MeanPrice)
	BBAS3T	2019	4924.428455284553
	BBDC4T	2019	3643.7828054298643
	BRKM5T	2019	3683.0973630831645
	CMIG4T	2019	1408.9468302658486
	CSNA3T	2019	1400.867403314917
	GGBR4T	2019	1466.2689873417721
	ITSA4T	2019	1282.3934426229507
	PETR4T	2019	2763.943482224248
	USIM5T	2019	880.4004995836802
	BBAS3T	2018	3565.7605004468273
	BBDC4T	2018	3330.918144611187
	BRKM5T	2018	5014.777493606138
	CMIG4T	2018	871.8730684326711
	CSNA3T	2018	920.0129107981221
	GGBR4T	2018	1586.2538569424964
	ITSA4T	2018	1139.4112820512821
	PETR4T	2018	2152.8916467780427
	USIM5T	2018	964.6364551863041
	BBAS3T	2017	3176.4322508398654
	BBDC4T	2017	3229.6673684210527
	BRKM5T	2017	3974.2268907563025
	CMIG4T	2017	849.0871670702179
	CSNA3T	2017	873.8371794871795
	GGBR4T	2017	1135.5294117647059

ITSA4T 2017 1007.6398713826367
PETR4T 2017 1487.8120950323973
USIM5T 2017 646.1754601226994
VALE5T 2017 2822.725641025641
BBAS3T 2016 2036.5633561643835
BBDC4T 2016 2585.122832369942
BRKM5T 2016 2440.7130044843047
CMIG4T 2016 742.6992084432718
CSNA3T 2016 879.3611738148984
GGBR4T 2016 799.6191819464034
ITSA4T 2016 806.4554896142433
PETR4T 2016 1090.7854609929077
USIM5T 2016 292.8732212160414
VALE5T 2016 1441.6931738212527
BBAS3T 2015 2082.96618852459
BBDC4T 2015 2705.7417893544734
BRKM5T 2015 1378.9717514124293
CMIG4T 2015 1002.3052805280528
CSNA3T 2015 519.1433823529412
GGBR4T 2015 767.9319371727748
ITSA4T 2015 842.9115044247787
PETR4T 2015 1002.3142565150741
USIM5T 2015 398.92831105710815
VALE5T 2015 1626.9178841309824
BBAS3T 2014 2474.987542468856
BBDC4T 2014 3285.2131147540986

+-----+-----+-----+

only showing top 50 rows

[24]: `#informacao 14`

```
#Desvio Padrão anual do preço médio (MeanPrice) da ação da PETR4,
↳ (NegociationCode)
sc.sql('select NegociationCode, SUBSTRING(TradingDate, 1, 4) as Year,
↳ stddev(MeanPrice) from bovespa where NegociationCode==\'PETR4\' group by
↳ NegociationCode, Year order by Year desc').show(20)
```

+-----+-----+-----+		
NegociationCode	Year	stddev_samp(MeanPrice)
+-----+-----+-----+		
PETR4 2019	164.42760488118657	
PETR4 2018	322.32991837509013	
PETR4 2017	131.33322761191312	
PETR4 2016	379.52406656319	
PETR4 2015	212.59195810379663	
PETR4 2014	319.8401226321017	
PETR4 2013	136.1536508448088	

PETR4 2012	202.65900303596553
PETR4 2011	304.10310718438325
PETR4 2010	392.22804616214563
PETR4 2009	414.3029746894998
PETR4 2008	2311.364312465641
PETR4 2007	1200.445391820076
PETR4 2006	242.19193000733335
PETR4 2005	3377.068887998981
PETR4 2004	757.8230426632923
PETR4 2003	863.8748535443933
PETR4 2002	613.8605232990916
PETR4 2001	333.4576639125791
PETR4 2000	20189.956356803512

+-----+-----+-----+-----+

only showing top 20 rows

[25]: *#informacao 15*
Desvio Padrão anual do preço médio (MeanPrice) das 10 ações (NegociationCode)
→ com mais operações na bovespa

```
top10 = df.select('NegociationCode', 'MeanPrice', 'TradingDate')\
    .groupBy('NegociationCode')\
    .agg(func.count('NegociationCode').alias('qt'))\
    .orderBy(func.desc('qt'))\
    .limit(10)\
    .select('Negociationcode').collect()

top10 = [str(i).split('=\\')[1].split('\\')[0] for i in top10] # filtrando.

# Criando uma cláusula where com o array top10:
mywhere = ' where '
for i in range(0, len(top10)):
    mywhere += f' NegociationCode=\\{top10[i]}\\'
    if i < len(top10)-1:
        mywhere += ' or '

print(top10)

sc.sql('select NegociationCode, SUBSTRING(TradingDate, 1, 4) as Year,
    →stddev(MeanPrice) from bovespa'+mywhere+' group by NegociationCode, Year
    →order by Year desc, NegociationCode').show(50)
```

['PETR4T', 'VALE5T', 'USIM5T', 'BBAS3T', 'CSNA3T', 'GGBR4T', 'BBDC4T', 'ITSA4T',
 'CMIG4T', 'BRKM5T']

+-----+-----+-----+-----+

NegociationCode	Year	stddev_samp(MeanPrice)
-----------------	------	------------------------

+-----+-----+-----+-----+

	BBAS3T 2019	278.4108910417547
	BBDC4T 2019	364.69206520223213
	BRKM5T 2019	889.9095240917367
	CMIG4T 2019	68.05303214710867
	CSNA3T 2019	219.22622899260105
	GGBR4T 2019	143.76969001819543
	ITSA4T 2019	73.34035510641367
	PETR4T 2019	176.76398173897732
	USIM5T 2019	92.63891532765003
	BBAS3T 2018	575.051063933194
	BBDC4T 2018	432.14526038986867
	BRKM5T 2018	443.11194870134796
	CMIG4T 2018	186.25904376536047
	CSNA3T 2018	83.84440052496272
	GGBR4T 2018	99.04875552857983
	ITSA4T 2018	151.4470044266044
	PETR4T 2018	338.21257536293416
	USIM5T 2018	139.37531194931884
	BBAS3T 2017	273.274782356005
	BBDC4T 2017	269.9457573432429
	BRKM5T 2017	597.6399056312631
	CMIG4T 2017	124.77845445317237
	CSNA3T 2017	163.8048806476733
	GGBR4T 2017	112.53931672173698
	ITSA4T 2017	79.64024718269013
	PETR4T 2017	135.1036271323912
	USIM5T 2017	217.04459485048557
	VALE5T 2017	243.00362994900908
	BBAS3T 2016	470.8910488349438
	BBDC4T 2016	433.04996245749703
	BRKM5T 2016	421.8264748924745
	CMIG4T 2016	147.42872993037642
	CSNA3T 2016	236.72543717725782
	GGBR4T 2016	276.12371126963797
	ITSA4T 2016	81.81346489527289
	PETR4T 2016	400.1555589906185
	USIM5T 2016	117.55224599489586
	VALE5T 2016	510.606044374684
	BBAS3T 2015	346.33333266832733
	BBDC4T 2015	535.9770520944436
	BRKM5T 2015	295.61541026880235
	CMIG4T 2015	288.8073638718649
	CSNA3T 2015	116.06609547866248
	GGBR4T 2015	193.91744668436476
	ITSA4T 2015	111.20937042147659
	PETR4T 2015	213.90144240157417
	USIM5T 2015	118.93182722037069
	VALE5T 2015	282.324250940134

only showing top 50 rows

Top Most By Currency: [['CR\$', 'BBD 4'], ['CZ\$', 'PMA 2'], ['NCZ\$', 'PET 2'],

['R\$', 'PETR4T']]

+-----+-----+-----+		
NegotiationCode	Year	MeanPrice
+-----+-----+-----+		
PETR4T	2019	2763.943482224248
PETR4T	2018	2152.8916467780427
PETR4T	2017	1487.8120950323973
PETR4T	2016	1090.7854609929077
PETR4T	2015	1002.3142565150741
PETR4T	2014	1671.9566591422122
PETR4T	2013	1849.2040432534086
PETR4T	2012	2154.13648409894
PETR4T	2011	2456.218079493493
PETR4T	2010	2983.363447559709
PETR4T	2009	3264.6164483260554
PETR4T	2008	4897.825913089842
PETR4T	2007	5544.564305500594
PETR4T	2006	4412.334102445778
PETR4T	2005	7806.432778489117
PETR4T	2004	8550.459595959595
PETR4T	2003	5838.4639278557115
PETR4T	2002	4834.294193548387
PETR4T	2001	5357.148730350665
PETR4T	1999	26129.480662983424
PETR4T	1998	20028.90404040404
BBD 4	1998	923.0672268907563
BBD 4	1997	940.3768115942029
BBD 4	1996	908.142378559464
BBD 4	1995	776.3051724137931
BBD 4	1994	1010.6907020872866
BBD 4	1993	62409.33536585366
BBD 4	1992	16516.576846307384
PET 2	1991	122792.3268206039
BBD 4	1991	48068.76091081594
PET 2	1990	1.1893549348230913E7
BBD 4	1990	122173.04329896907
PMA 2	1990	68709.10914454277
PET 2	1989	815575.5820895522
PMA 2	1989	40317.880952380954
BBD 4	1989	23985.135396518373
PET 2	1988	68248.846666666666
BBD 4	1988	5856.22385321101
PMA 2	1988	4754.275229357798
PET 2	1987	32823.77319587629
PMA 2	1987	1507.8791018998272
BBD 4	1987	1498.091836734694
PET 2	1986	133676.46993318485
BBD 4	1986	1859.125322997416

| PMA 2|1986| 1800.8725961538462|
+-----+-----+-----+

[]:

[]: