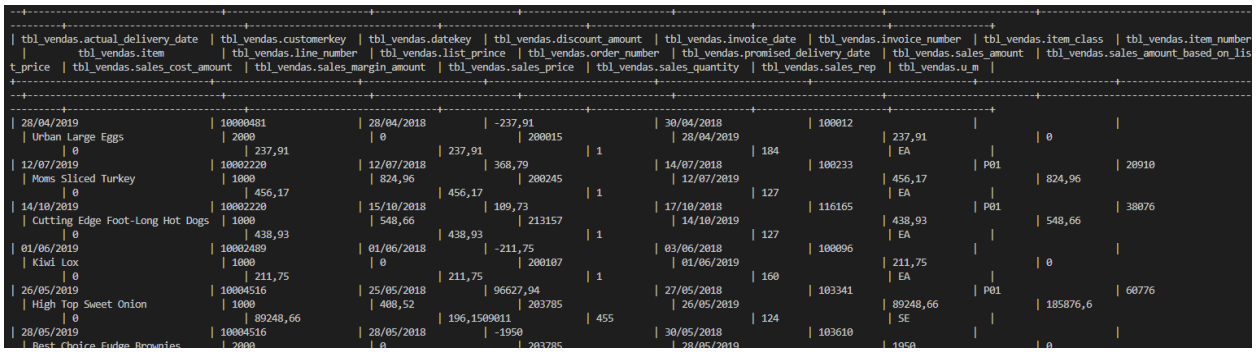
```
root@hive_server:/opt# hdfs dfs -mkdir /datalake/raw/clientes
root@hive_server:/opt# hdfs dfs -mkdir /datalake/raw/divisao
root@hive_server:/opt# hdfs dfs -mkdir /datalake/raw/endereco
root@hive_server:/opt# hdfs dfs -mkdir /datalake/raw/regiao
root@hive_server:/opt# hdfs dfs -mkdir /datalake/raw/vendas
```

copiando os arquivos para o hdfs

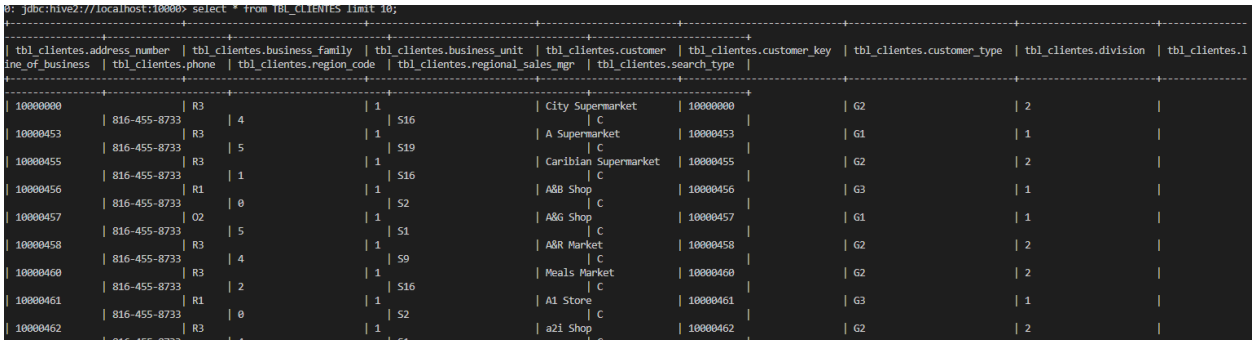
```
hdfs dfs -copyFromLocal CLIENTES.csv /datalake/raw/clientes
hdfs dfs -copyFromLocal DIVISAO.csv /datalake/raw/divisao
hdfs dfs -copyFromLocal ENDERECO.csv /datalake/raw/endereco
hdfs dfs -copyFromLocal REGIAO.csv /datalake/raw/regiao
hdfs dfs -copyFromLocal VENDAS.csv /datalake/raw/vendas
```

Primeira tabela TBL_VENDAS criada



tbl_vendas.actual_delivery_date	tbl_vendas.customerkey	tbl_vendas.datekey	tbl_vendas.discount_amount	tbl_vendas.invoice_date	tbl_vendas.invoice_number	tbl_vendas.item_class	tbl_vendas.item_number	tbl_vendas.item	tbl_vendas.line_number	tbl_vendas.list_prince	tbl_vendas.order_number	tbl_vendas.promised_delivery_date	tbl_vendas.sales_amount	tbl_vendas.sales_amount_based_on_list_price	tbl_vendas.sales_cost_amount	tbl_vendas.sales_margin_amount	tbl_vendas.sales_price	tbl_vendas.sales_quantity	tbl_vendas.sales_rep	tbl_vendas.u_m
28/04/2019	10000481	28/04/2018	-237,91	30/04/2018	100012			Urban Large Eggs	2000	0	200015	28/04/2019	237,91	0	0	237,91	237,91	1	184	EA
12/07/2019	10002220	12/07/2018	368,79	14/07/2018	100233	P01	20910	Moms Sliced Turkey	1000	824,96	200245	12/07/2019	456,17	824,96	0	456,17	456,17	1	127	EA
14/10/2019	10002220	15/10/2018	109,73	17/10/2018	116165	P01	38076	Cutting Edge Foot-Long Hot Dogs	1000	548,66	213157	14/10/2019	438,93	548,66	0	438,93	438,93	1	127	EA
01/06/2019	10002489	01/06/2018	-211,75	03/06/2018	100096			Kiwi Lox	1000	0	200107	01/06/2019	211,75	0	0	211,75	211,75	1	160	EA
26/05/2019	10004516	25/05/2018	96627,94	27/05/2018	103341	P01	60776	High Top Sweet Onion	1000	408,52	203785	26/05/2019	89248,66	185876,6	0	89248,66	196,1509011	455	124	SE
28/05/2019	10004516	28/05/2018	-1950	30/05/2018	103610			Best Choice Fudge Brownies	2000	0	203785	28/05/2019	1950	0						

Tabela TBL_CLIENTES



```
0: jdbc:hive2://localhost:10000> select * from tbl_CLIENTES limit 10;
```

tbl_clientes.address_number	tbl_clientes.business_family	tbl_clientes.business_unit	tbl_clientes.customer	tbl_clientes.customer_key	tbl_clientes.customer_type	tbl_clientes.division	tbl_clientes.line_of_business	tbl_clientes.phone	tbl_clientes.region_code	tbl_clientes.regional_sales_mgr	tbl_clientes.search_type
10000000	R3	1	City Supermarket	10000000	G2	2		816-455-8733	4	S16	C
10000453	R3	1	A Supermarket	10000453	G1	1		816-455-8733	5	S19	C
10000455	R3	1	Caribian Supermarket	10000455	G2	2		816-455-8733	1	S16	C
10000456	R1	1	A&B Shop	10000456	G3	1		816-455-8733	0	S2	C
10000457	O2	1	A&G Shop	10000457	G1	1		816-455-8733	5	S1	C
10000458	R3	1	A&R Market	10000458	G2	2		816-455-8733	4	S9	C
10000460	R3	1	Meals Market	10000460	G2	2		816-455-8733	2	S16	C
10000461	R1	1	A1 Store	10000461	G3	1		816-455-8733	0	S2	C
10000462	R3	1	a2i Shop	10000462	G2	2					

Tabela TBL_ENDERECO

```
0: jdbc:hive2://localhost:10000> select * from TBL_ENDERECO limit 10;
```

tbl_endereco.address_number	tbl_endereco.city	tbl_endereco.country	tbl_endereco.customer_address_1	tbl_endereco.customer_address_2	tbl_endereco.customer_address_3	tbl_endereco.customer_address_4	tbl_endereco.state	tbl_endereco.zip_code
10000000	Akron	US	PO Box 6258				OH	44312
10000453		UK						
10000455	Huntington Beach	US	7392 Count Circle				CA	92647
10000456	Edmonton	CA	8151 Wagner Road				AB	T6E 4N6
10000458	Saginaw	US	PO Box 840				MI	48606
10000460	Goodlettsville	US	709 Rivergate Parkway				TN	37072
10000461	Boucherville	CA	1391 Gay Lussac				QU	J4B 7K1
10000462	Hazelwood	US	6311 North Lindbergh Boulevard				MO	63042
10000466	North Highlands	US	3213 Orange Grove Avenue					

Tabela TBL_REGIAO

```
0: jdbc:hive2://localhost:10000> select * from TBL_REGIAO limit 10;
```

tbl_regiao.region_code	tbl_regiao.region_name
0	Canada
1	Western
2	Southern
3	Northeast
4	Central
5	International

TABELA TBL_DIVISAO

```
0: jdbc:hive2://localhost:10000> select * from TBL_DIVISAO;
```

tbl_divisao.division	tbl_divisao.division_name
1	International
2	Domestic

In [51]:
```
from pyspark.sql import SparkSession, dataframe
from pyspark.sql.functions import when, col, sum, count, isnan, round
from pyspark.sql.functions import regexp_replace, concat_ws, sha2, rtrim, substring
from pyspark.sql.functions import unix_timestamp, from_unixtime, to_date
from pyspark.sql.types import StructType, StructField
from pyspark.sql.types import DoubleType, IntegerType, StringType
from pyspark.sql import HiveContext

import os
import re

from pyspark.sql.functions import regexp_replace
from pyspark.sql.functions import when

spark = SparkSession.builder.master("local[*]")\
    .enableHiveSupport()\
    .getOrCreate()
```

In [ ]:
```
```

In [58]:
```
df_vendas = spark.sql("select * from desafio_curso.TBL_VENDAS")
df_clientes = spark.sql("select * from desafio_curso.TBL_CLIENTES")
df_endereco = spark.sql("select * from desafio_curso.TBL_ENDERECO")
df_regiao = spark.sql("select * from desafio_curso.TBL_REGIAO")
df_divisao = spark.sql("select * from desafio_curso.TBL_DIVISAO")
```

In [60]:
```
df_vendas.columns
```

In [61]:
```
df_vendas.createOrReplaceTempView("dim_vendas")
```

In [65]:
```
df_regiao.columns
```

Out[65]:
```
['region_code', 'region_name']
```

In [69]:
```
df_endereco.columns
```

Out[69]:
```
['address_number',
 'city',
 'country',
 'customer_address_1',
 'customer_address_2',
 'customer_address_3',
```

In [71]:
```
df_endereco.createOrReplaceTempView("dim_localidade")
```

In [ ]:

In [72]:
```
dim_loc = df_endereco = spark.sql("""
   select address_number, city, country , customer_address_1, customer_address_2, customer_address_3, customer_address_4,
   state, zip_code
   from dim_localidade



""")
```

In [74]:
```
dim_loc.columns
```

Out[74]:
```
['address_number',
 'city',
 'country',
 'customer_address_1',
 'customer_address_2',
 'customer_address_3',
 'customer_address_4',
 'state',
 'zip_code']
```

In [75]:
```
salvar_df(df_endereco, 'dim_loc')
```

```
hdfs dfs -get /datalake/gold/dim_loc/part-* /input/desafio_hive/gold/dim_loc.csv
```

In [ ]:

In [ ]:

In [ ]:

In [63]:
```
dim_temp = df_vendas = spark.sql("""
select actual_delivery_date, datekey, invoice_date, promised_delivery_date
from dim_vendas


""")
```

In [64]:
```
dim_temp.show(3)
```

...

In [68]:
```
salvar_df(df_vendas,file='dim_temp')
```

```
hdfs dfs -get /datalake/gold/dim_temp/part-* /input/desafio_hive/gold/dim_temp.csv
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [37]:
```
#df = df_pedido.join(df_item_pedido,df_pedido.id_pedido ==  df_item_pedido.id_pedido,"inner")
df_regiao.columns
```

Out[37]:
```
['region_code', 'region_name']
```

In [28]:
```
df_1 = df_clientes.join(df_endereco,df_clientes.address_number == df_endereco.address_number,"left")
```

In [29]:
```
df_1.columns
```

...

In [30]:
```
df_1.createOrReplaceTempView("dim_cli")
```

In [53]:
```
df_clientess = df_1 = spark.sql("""
select  business_family, business_unity, customer, customerkey, customer_type,
   line_of_business, phone, search_type
   from dim_cli


""")
```

In [ ]:

In [40]:
```
ft = ft_vendas = spark.sql("""
select customerkey, discount_amount, invoice_number, customerkey, item_class, item_number, item, line_number,list_prince,
order_number, sales_amount, sales_amount_based_on_list_price, sales_cost_amount, sales_margin_amount, sales_price,
sales_quantity, sales_rep
from ft_vendas

""")
```

In [ ]:
```
```

In [41]:
```
ft.columns
```

Out[41]:
```
['customerkey',
 'discount_amount',
 'invoice_number',
 'customerkey',
 'item_class',
 'item_number',
 'item',
 'line_number',
 'list_prince',
 'order_number',
 'sales_amount',
 'sales_amount_based_on_list_price',
 'sales_cost_amount',
 'sales_margin_amount',
 'sales_price',
 'sales_quantity',
 'sales_rep']
```

In [56]:
```
def salvar_df(df, file):
    output = "/input/desafio_hive/gold/" + file
    erase = "hdfs dfs -rm " + output + "/*"
    rename = "hdfs dfs -get /datalake/gold/"+file+"/part-* /input/desafio_hive/gold/"+file+".csv"
    print(rename)


    df.coalesce(1).write\
        .format("csv")\
        .option("header", True)\
        .option("delimiter", ";")\
        .mode("overwrite")\
        .save("/datalake/gold/"+file+"/")

    os.system(erase)
    os.system(rename)

salvar_df(df_clientes, 'df_clientess')
```

```
hdfs dfs -get /datalake/gold/df_clientess/part-* /input/desafio_hive/gold/df_clientess.csv
```

In [ ]:

In [ ]:

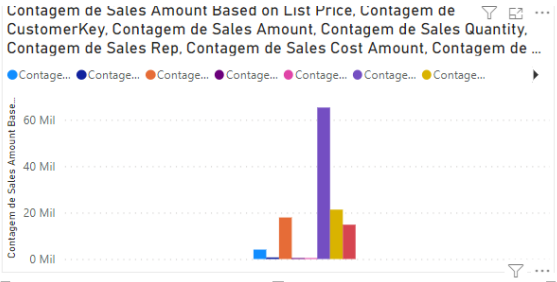
In [ ]:

In [48]:
```
salvar_df(df_clientess, 'df_clientes')
```

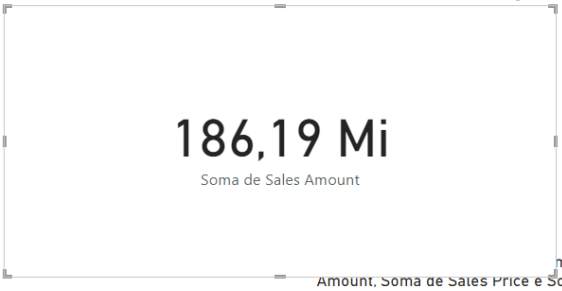
```
hdfs dfs -get /datalake/gold/df_clientes/part-* /input/desafio_hive/gold/df_clientes.csv
```

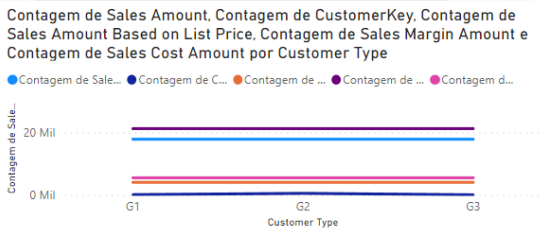
In [57]:
```
salvar_df(df_vendas, 'ft')
```

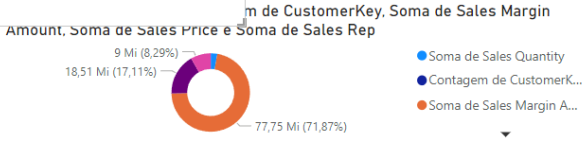
```
hdfs dfs -get /datalake/gold/ft/part-* /input/desafio_hive/gold/ft.csv
```

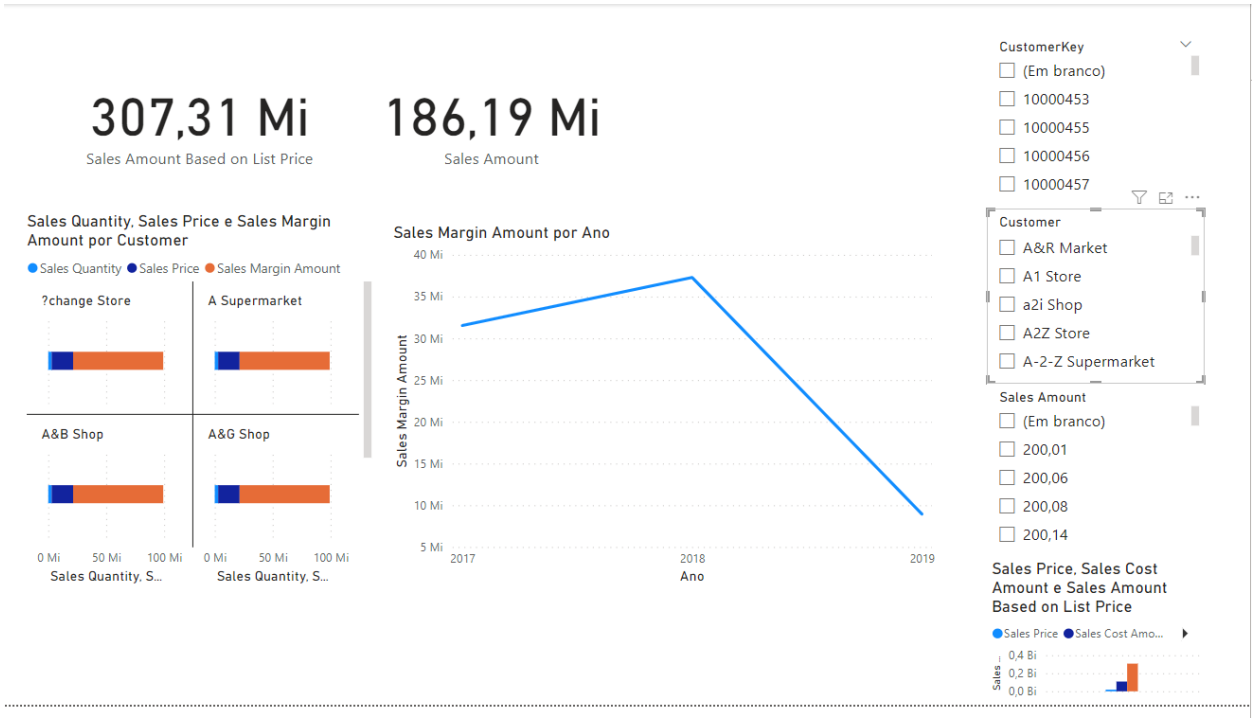


oma de Sales Amount	Contagem de List Price	Ano	Trimestre	Mês	Dia	Soma de Sa
	1					
505.832,66	41	2017	Trim 1	janeiro	12	
156.688,90	72	2017	Trim 1	janeiro	13	
356.292,63	41	2017	Trim 1	janeiro	18	
136.828,62	38	2017	Trim 1	janeiro	19	
307.447,66	41	2017	Trim 1	janeiro	20	
104.588,46	68	2017	Trim 1	janeiro	21	
186.186.769,05	**1063**					








307,31 Mi
Sales Amount Based on List Price
186,19 Mi
Sales Amount
Sales Quantity, Sales Price e Sales Margin Amount por Customer
Sales Quantity
Sales Price
Sales Margin Amount
?change Store
A Supermarket
A&B Shop
A&G Shop
0 Mi
50 Mi
100 Mi
Sales Quantity, S...
Sales Margin Amount por Ano
40 Mi
35 Mi
30 Mi
25 Mi
20 Mi
15 Mi
10 Mi
5 Mi
Sales Margin Amount
2017
2018
2019
Ano
CustomerKey
(Em branco)
10000453
10000455
10000456
10000457
Customer
A&R Market
A1 Store
a2i Shop
A2Z Store
A-2-Z Supermarket
Sales Amount
(Em branco)
200,01
200,06
200,08
200,14
Sales Price, Sales Cost Amount e Sales Amount Based on List Price
Sales Price
Sales Cost Amo...
0,4 Bi
0,2 Bi
0,0 Bi
Sales