```
## Урок 4. Партицирование данных по дате. Динамическое партицирование

1. За основу возьмите Задание 4 решенное на семинаре.
В файле s4_2 параметры кредита: Займ 9400000, срок 30 лет, ставка 10.6%.
Через https://calcus.ru/kreditnyj-kalkulyator-s-dosrochnym-pogasheniem добавьте
два листа в Excel с постоянным платежом 120 или 150 тыс. руб.
(Необязательно, но можете также сделать и для платежа 250 и 300).
Добавьте графики с досрочным погашением по этим пирометрам. Т.е. линии по
выплатам основного долга и процентов если платеж будет 120 или 150 тыс. руб. В
результате должно получиться 6 линий. Используйте разные цвета.
import pyspark,time,platform,sys,os
from datetime import datetime
from pyspark.sql.session import SparkSession
from pyspark.sql.functions import col,lit,current_timestamp
import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import inspect,create_engine
from pandas.io import sql
import warnings,matplotlib
import configparser

config = configparser.ConfigParser()
config.read('/Users/Esdesu/Desktop/JreJre/ETL/config.ini')
password = config['credentials']['password']

warnings.filterwarnings("ignore")
t0=time.time()
con=create_engine("mysql://root:" + password + "@localhost/spark")
os.environ['PYSPARK_PYTHON'] = sys.executable
os.environ['PYSPARK_DRIVER_PYTHON'] = sys.executable
spark=SparkSession.builder.appName("Home Work №4").getOrCreate()

sql.execute("""drop table if exists spark.`W4T1`""",con)
sql.execute("""CREATE TABLE if not exists spark.`W4T1` (
        `number` INT(10) NULL DEFAULT NULL,
        `Month` DATE NULL DEFAULT NULL,
        `Payment amount` FLOAT NULL DEFAULT NULL,
        `Payment of the principal debt` FLOAT NULL DEFAULT NULL,
        `Payment of interest` FLOAT NULL DEFAULT NULL,
        `Balance of debt` FLOAT NULL DEFAULT NULL,
        `interest` FLOAT NULL DEFAULT NULL,
        `debt` FLOAT NULL DEFAULT NULL
)
COLLATE='utf8mb4_general_ci'
ENGINE=InnoDB""",con)

from pyspark.sql.window import Window
from pyspark.sql.functions import sum as sum1
w =
Window.partitionBy(lit(1)).orderBy("number").rowsBetween(Window.unboundedPrecedin
g, Window.currentRow)
```

```
dfG = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'General'!A1:F361")\
        .option("useHeader", "false")\
        .option("treatEmptyValuesAsNulls", "false")\
        .option("inferSchema", "true").option("addColorColumns", "true")\
        .option("usePlainNumberFormat","true")\
        .option("startColumn", 0)\
        .option("endColumn", 99)\
        .option("timestampFormat", "MM-dd-yyyy HH:mm:ss")\
        .option("maxRowsInMemory", 20)\
        .option("excerptSize", 10)\
        .option("header", "true")\
        .format("excel")\
        .load("/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work#4/Task_1/W4T1.x
lsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df120 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'120'!A1:F135")\
        .option("useHeader", "false")\
        .option("treatEmptyValuesAsNulls", "false")\
        .option("inferSchema", "true").option("addColorColumns", "true")\
        .option("usePlainNumberFormat","true")\
        .option("startColumn", 0)\
        .option("endColumn", 99)\
        .option("timestampFormat", "MM-dd-yyyy HH:mm:ss")\
        .option("maxRowsInMemory", 20)\
        .option("excerptSize", 10)\
        .option("header", "true")\
        .format("excel")\
        .load("/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work#4/Task_1/W4T1.x
lsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df150 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'150'!A1:F93")\
        .option("useHeader", "false")\
        .option("treatEmptyValuesAsNulls", "false")\
        .option("inferSchema", "true").option("addColorColumns", "true")\
        .option("usePlainNumberFormat","true")\
        .option("startColumn", 0)\
        .option("endColumn", 99)\
        .option("timestampFormat", "MM-dd-yyyy HH:mm:ss")\
        .option("maxRowsInMemory", 20)\
        .option("excerptSize", 10)\
        .option("header", "true")\
        .format("excel")\
        .load("/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work#4/Task_1/W4T1.x
lsx").limit(1000)\
```

```
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df250 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'250'!A1:F47")\
        .option("useHeader", "false")\
        .option("treatEmptyValuesAsNulls", "false")\
        .option("inferSchema", "true").option("addColorColumns", "true")\
        .option("usePlainNumberFormat","true")\
        .option("startColumn", 0)\
        .option("endColumn", 99)\
        .option("timestampFormat", "MM-dd-yyyy HH:mm:ss")\
        .option("maxRowsInMemory", 20)\
        .option("excerptSize", 10)\
        .option("header", "true")\
        .format("excel")\
        .load("/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work#4/Task_1/W4T1.x
lsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df300 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'300'!A1:F38")\
        .option("useHeader", "false")\
        .option("treatEmptyValuesAsNulls", "false")\
        .option("inferSchema", "true").option("addColorColumns", "true")\
        .option("usePlainNumberFormat","true")\
        .option("startColumn", 0)\
        .option("endColumn", 99)\
        .option("timestampFormat", "MM-dd-yyyy HH:mm:ss")\
        .option("maxRowsInMemory", 20)\
        .option("excerptSize", 10)\
        .option("header", "true")\
        .format("excel")\
        .load("/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work#4/Task_1/W4T1.x
lsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df_combined = dfG.union(df120).union(df150).union(df250).union(df300)

df_combined.write.format("jdbc").option("url","jdbc:mysql://localhost:3306/spark?
user=root&password=" + password)\
        .option("driver", "com.mysql.cj.jdbc.Driver").option("dbtable", "W4T1")\
        .mode("append").save()

"""df_pandas = df_combined.toPandas()"""

df_pandas1 = dfG.toPandas()
df_pandas2 = df120.toPandas()
df_pandas3 = df150.toPandas()
```

```
df_pandas4 = df250.toPandas()
df_pandas5 = df300.toPandas()

ax = plt.gca()
ax.ticklabel_format(style='plain')

df_pandas1.plot(kind='line', x='number', y='debt', color='green', ax=ax,
label='Debt Genetal')
df_pandas1.plot(kind='line', x='number', y='interest', color='red', ax=ax,
label='Interest General')
df_pandas2.plot(kind='line', x='number', y='debt', color='grey', ax=ax,
label='Debt 120')
df_pandas2.plot(kind='line', x='number', y='interest', color='orange', ax=ax,
label='Interest 120')
df_pandas3.plot(kind='line', x='number', y='debt', color='purple', ax=ax,
label='Debt 150')
df_pandas3.plot(kind='line', x='number', y='interest', color='yellow', ax=ax,
label='Interest 150')
df_pandas4.plot(kind='line', x='number', y='debt', color='blue', ax=ax,
label='Debt 250')
df_pandas4.plot(kind='line', x='number', y='interest', color='brown', ax=ax,
label='Interest 250')
df_pandas5.plot(kind='line', x='number', y='debt', color='black', ax=ax,
label='Debt 300')
df_pandas5.plot(kind='line', x='number', y='interest', color='pink', ax=ax,
label='Interest 300')

plt.title('Loan Payments Over Time')
plt.grid ( True )
ax.set(xlabel=None)

plt.show()
spark.stop()
t1=time.time()
print('finished',time.strftime('%H:%M:%S',time.gmtime(round(t1-t0))))
```