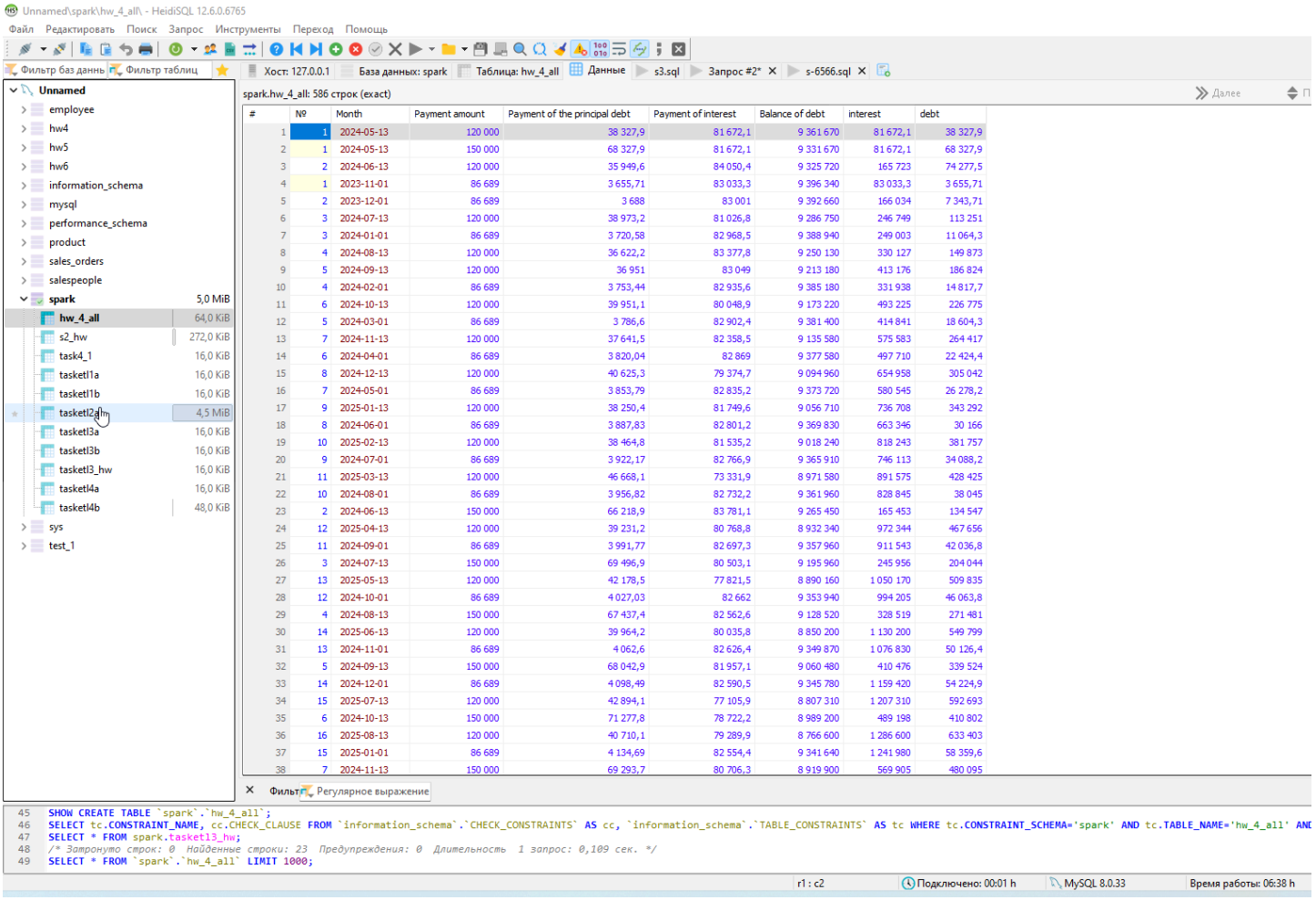
```
import pyspark,time,platform,sys,os
from datetime import datetime
from pyspark.sql.session import SparkSession
from pyspark.sql.functions import col,lit,current_timestamp
import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import inspect,create_engine
from pandas.io import sql
import warnings,matplotlib
warnings.filterwarnings("ignore")
t0=time.time()
con=create_engine("mysql://root:@localhost/spark")
os.environ['PYSPARK_PYTHON'] = sys.executable
os.environ['PYSPARK_DRIVER_PYTHON'] = sys.executable
spark=SparkSession.builder.appName("Hi").getOrCreate()

sql.execute("""drop table if exists spark.`HW_4_all`""",con)
sql.execute("""CREATE TABLE if not exists spark.`HW_4_all` (
            `№` 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("№").rowsBetween(Window.unboundedPreceding, Window.currentRow)
df1 = spark.read.format("com.crealytics.spark.excel")\
      .option("dataAddress", "'sheet_sem'!A1")\
      .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("D:/Geekbrains_ETL/s4_2_HW.xlsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df2 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'sheet_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("D:/Geekbrains_ETL/s4_2_HW.xlsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))


df3 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'sheet_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("D:/Geekbrains_ETL/s4_2_HW.xlsx").limit(1000)\
        .withColumn("interest", sum1(col("Payment of interest")).over(w))\
        .withColumn("debt", sum1(col("Payment of the principal debt")).over(w))

df_combined = df1.union(df2).union(df3)
```

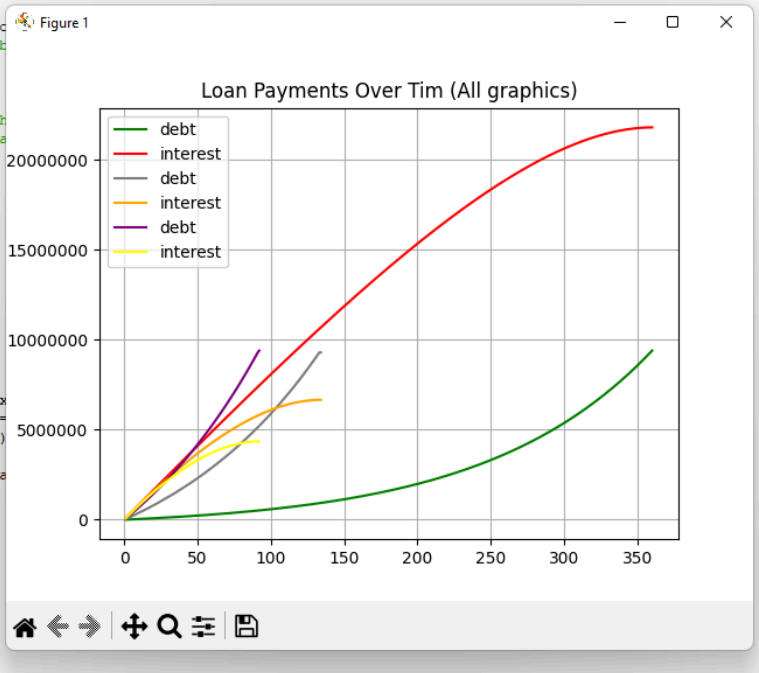
```
df_combined.write.format("jdbc").option("url","jdbc:mysql://localhost:3306/spark?user=root&password=")\
      .option("driver", "com.mysql.cj.jdbc.Driver").option("dbtable", "HW_4_all")\
      .mode("append").save()

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

df_pandas1 = df1.toPandas()
df_pandas2 = df2.toPandas()
df_pandas3 = df3.toPandas()

# Get current axis
ax = plt.gca()
ax.ticklabel_format(style='plain')

# bar plot
df_pandas1.plot(kind='line', x='№', y='debt', color='green', ax=ax)
df_pandas1.plot(kind='line', x='№', y='interest', color='red', ax=ax)
df_pandas2.plot(kind='line', x='№', y='debt', color='grey', ax=ax) # ежемесячный платеж 120 000
df_pandas2.plot(kind='line', x='№', y='interest', color='orange', ax=ax)  # ежемесячный платеж 120 000
df_pandas3.plot(kind='line', x='№', y='debt', color='purple', ax=ax)  # ежемесячный платеж 150 000
df_pandas3.plot(kind='line', x='№', y='interest', color='yellow', ax=ax)  # ежемесячный платеж 150 000

# set the title
plt.title('Loan Payments Over Tim (All graphics)')
plt.grid ( True )
ax.set(xlabel=None)

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


Unnamed\spark\hw_4_all\ - HeidiSQL 12.6.0.6765
Файл Редактировать Поиск Запрос Инструменты Переход Помощь
Фильтр баз данны Фильтр таблиц
Хост: 127.0.0.1 База данных: spark Таблица: hw_4_all Данные s3.sql Запрос #2* s-6566.sql
Unnamed
employee
hw4
hw5
hw6
information_schema
mysql
performance_schema
product
sales_orders
salespeople
spark 5,0 MiB
hw_4_all 64,0 KiB
s2_hw 272,0 KiB
task4_1 16,0 KiB
tasketl1a 16,0 KiB
tasketl1b 16,0 KiB
tasketl2hw 4,5 MiB
tasketl3a 16,0 KiB
tasketl3b 16,0 KiB
tasketl3_hw 16,0 KiB
tasketl4a 16,0 KiB
tasketl4b 48,0 KiB
sys
test_1
spark.hw_4_all: 586 строк (exact)
Далее
№ Month Payment amount Payment of the principal debt Payment of interest Balance of debt interest debt
1 1 2024-05-13 120 000 38 327,9 81 672,1 9 361 670 81 672,1 38 327,9
2 1 2024-05-13 150 000 68 327,9 81 672,1 9 331 670 81 672,1 68 327,9
3 2 2024-06-13 120 000 35 949,6 84 050,4 9 325 720 165 723 74 277,5
4 1 2023-11-01 86 689 3 655,71 83 033,3 9 396 340 83 033,3 3 655,71
5 2 2023-12-01 86 689 3 688 83 001 9 392 660 166 034 7 343,71
6 3 2024-07-13 120 000 38 973,2 81 026,8 9 286 750 246 749 113 251
7 3 2024-01-01 86 689 3 720,58 82 968,5 9 388 940 249 003 11 064,3
8 4 2024-08-13 120 000 36 622,2 83 377,8 9 250 130 330 127 149 873
9 5 2024-09-13 120 000 36 951 83 049 9 213 180 413 176 186 824
10 4 2024-02-01 86 689 3 753,44 82 935,6 9 385 180 331 938 14 817,7
11 6 2024-10-13 120 000 39 951,1 80 048,9 9 173 220 493 225 226 775
12 5 2024-03-01 86 689 3 786,6 82 902,4 9 381 400 414 841 18 604,3
13 7 2024-11-13 120 000 37 641,5 82 358,5 9 135 580 575 583 264 417
14 6 2024-04-01 86 689 3 820,04 82 869 9 377 580 497 710 22 424,4
15 8 2024-12-13 120 000 40 625,3 79 374,7 9 094 960 654 958 305 042
16 7 2024-05-01 86 689 3 853,79 82 835,2 9 373 720 580 545 26 278,2
17 9 2025-01-13 120 000 38 250,4 81 749,6 9 056 710 736 708 343 292
18 8 2024-06-01 86 689 3 887,83 82 801,2 9 369 830 663 346 30 166
19 10 2025-02-13 120 000 38 464,8 81 535,2 9 018 240 818 243 381 757
20 9 2024-07-01 86 689 3 922,17 82 766,9 9 365 910 746 113 34 088,2
21 11 2025-03-13 120 000 46 668,1 73 331,9 8 971 580 891 575 428 425
22 10 2024-08-01 86 689 3 956,82 82 732,2 9 361 960 828 845 38 045
23 2 2024-06-13 150 000 66 218,9 83 781,1 9 265 450 165 453 134 547
24 12 2025-04-13 120 000 39 231,2 80 768,8 8 932 340 972 344 467 656
25 11 2024-09-01 86 689 3 991,77 82 697,3 9 357 960 911 543 42 036,8
26 3 2024-07-13 150 000 69 496,9 80 503,1 9 195 960 245 956 204 044
27 13 2025-05-13 120 000 42 178,5 77 821,5 8 890 160 1 050 170 509 835
28 12 2024-10-01 86 689 4 027,03 82 662 9 353 940 994 205 46 063,8
29 4 2024-08-13 150 000 67 437,4 82 562,6 9 128 520 328 519 271 481
30 14 2025-06-13 120 000 39 964,2 80 035,8 8 850 200 1 130 200 549 799
31 13 2024-11-01 86 689 4 062,6 82 626,4 9 349 870 1 076 830 50 126,4
32 5 2024-09-13 150 000 68 042,9 81 957,1 9 060 480 410 476 339 524
33 14 2024-12-01 86 689 4 098,49 82 590,5 9 345 780 1 159 420 54 224,9
34 15 2025-07-13 120 000 42 894,1 77 105,9 8 807 310 1 207 310 592 693
35 6 2024-10-13 150 000 71 277,8 78 722,2 8 989 200 489 198 410 802
36 16 2025-08-13 120 000 40 710,1 79 289,9 8 766 600 1 286 600 633 403
37 15 2025-01-01 86 689 4 134,69 82 554,4 9 341 640 1 241 980 58 359,6
38 7 2024-11-13 150 000 69 293,7 80 706,3 8 919 900 569 905 480 095
Фильтр Регулярное выражение
45 SHOW CREATE TABLE `spark`.`hw_4_all`;
46 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS tc WHERE tc.CONSTRAINT_SCHEMA='spark' AND tc.TABLE_NAME='hw_4_all' AND
47 SELECT * FROM spark.tasketl3_hw;
48 /* Затронуто строк: 0 Найденные строки: 23 Предупреждения: 0 Длительность 1 запрос: 0,109 сек. */
49 SELECT * FROM `spark`.`hw_4_all` LIMIT 1000;
r1 : c2
Подключено: 00:01 h
MySQL 8.0.33
Время работы: 06:38 h

s4_hw.py - D:\Geekbrains_ETL\s4_hw.py (3.8.10)

File Edit Format Run Options Window Help

```
        .option("excerptSize", 10)\
        .option("header", "true")\
        .format("excel")\
        .load("D:/Geekbrains_ETL/s4_2_HW.xlsx").limit(1000)\
        .withColumn("interest", suml(col("Payment of interest")).over(w))\
        .withColumn("debt", suml(col("Payment of the principal debt")).over(w))


df3 = spark.read.format("com.crealytics.spark.excel")\
        .option("dataAddress", "'sheet_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("D:/Geekbrains_ETL/s4_2_HW.xlsx").limit(1000)\
        .withColumn("interest", suml(col("Payment of interest")).c
        .withColumn("debt", suml(col("Payment of the principal deb

df_combined = df1.union(df2).union(df3)

df_combined.write.format("jdbc").option("url","jdbc:mysql://localh
        .option("driver", "com.mysql.cj.jdbc.Driver").option("dbta
        .mode("append").save()

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

df_pandas1 = df1.toPandas()
df_pandas2 = df2.toPandas()
df_pandas3 = df3.toPandas()

# Get current axis
ax = plt.gca()
ax.ticklabel_format(style='plain')

# bar plot
df_pandas1.plot(kind='line', x='№', y='debt', color='green', ax=ax
df_pandas1.plot(kind='line', x='№', y='interest', color='red', ax=
df_pandas2.plot(kind='line', x='№', y='debt', color='grey', ax=ax)
df_pandas2.plot(kind='line', x='№', y='interest', color='orange',
df_pandas3.plot(kind='line', x='№', y='debt', color='purple', ax=a
df_pandas3.plot(kind='line', x='№', y='interest', color='yellow',

# set the title
plt.title('Loan Payments Over Tim (All graphics)')
plt.grid ( True )
ax.set(xlabel=None)

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



Ln: 76 C