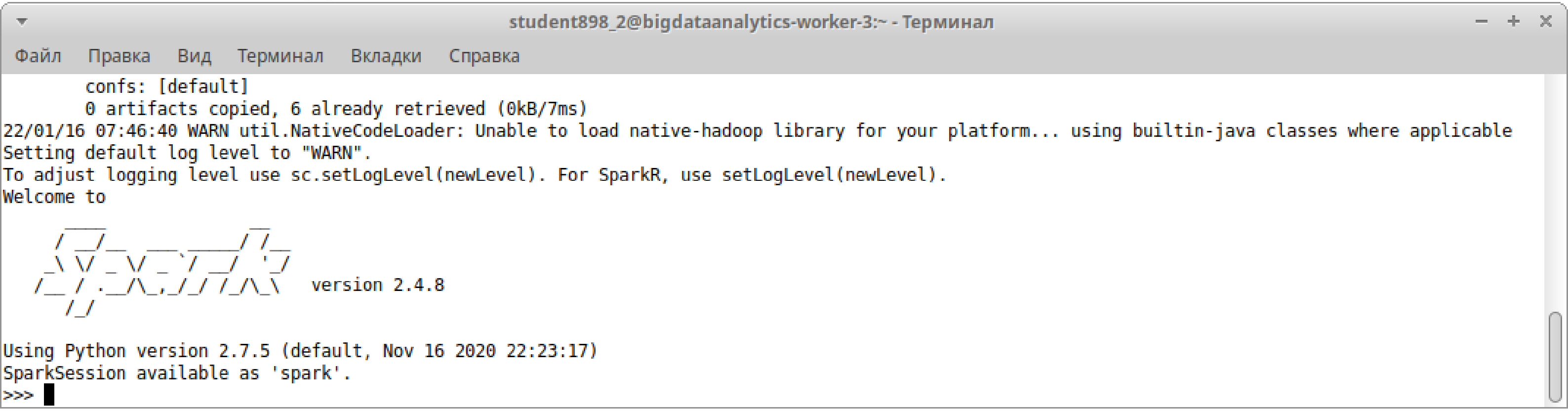
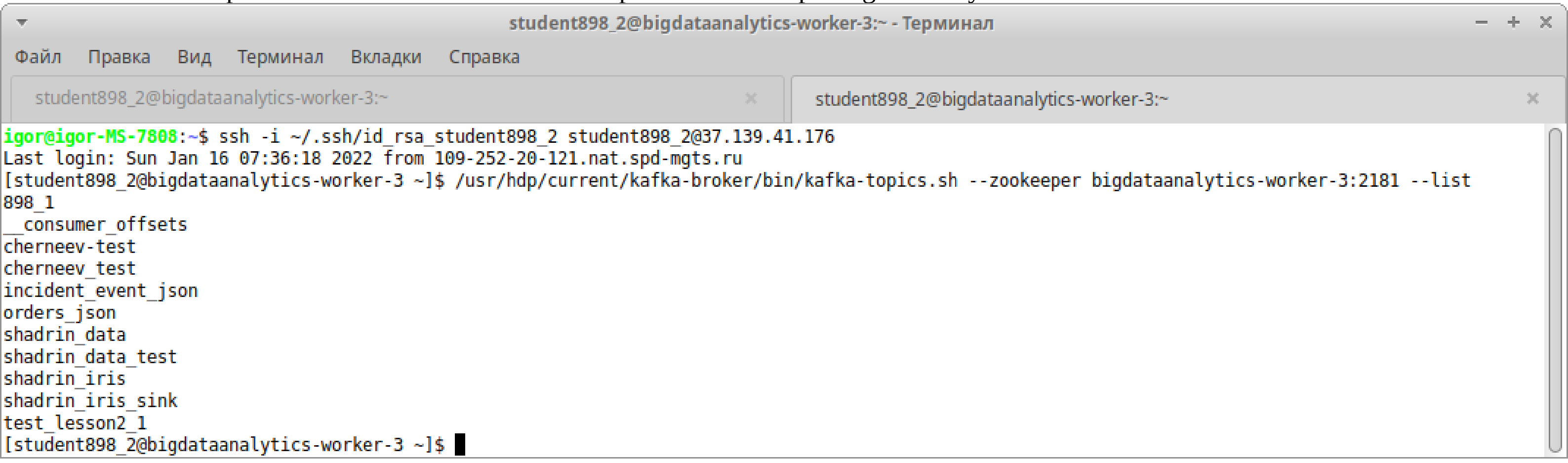
# 4. Spark Streaming. Sinks

ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
export SPARK_KAFKA_VERSION=0.10
/opt/spark-2.4.8/bin/pyspark --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 --driver-memory 512m --master local[1]



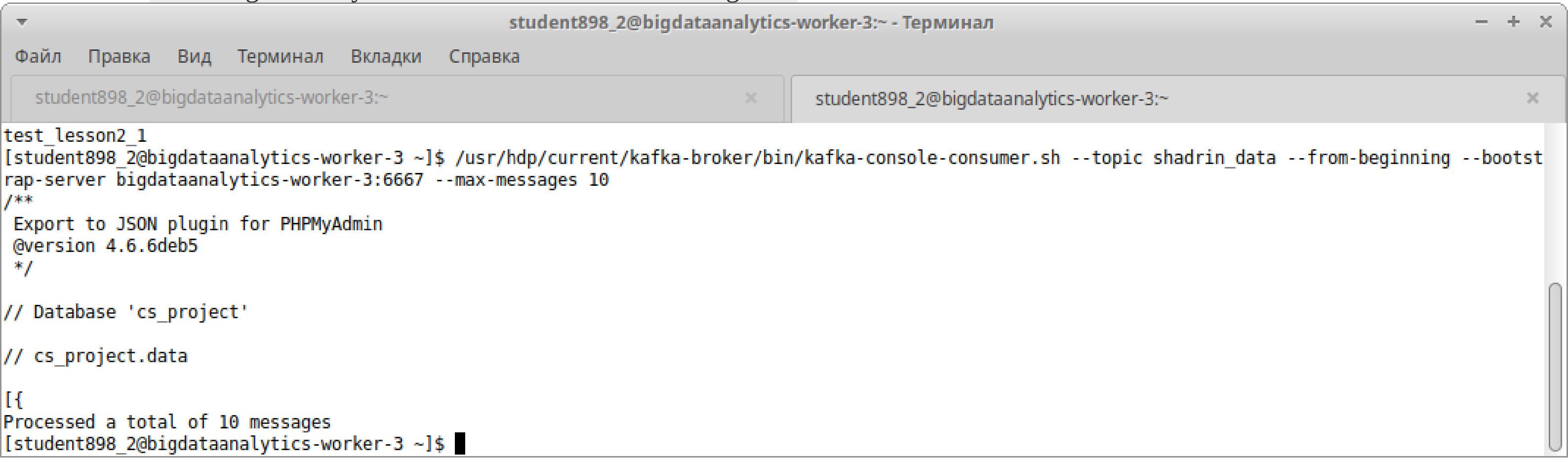
```
        confs: [default]
        0 artifacts copied, 6 already retrieved (0kB/7ms)
22/01/16 07:46:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.8
      /_/

Using Python version 2.7.5 (default, Nov 16 2020 22:23:17)
SparkSession available as 'spark'.
>>> 
```

Подключены все зависимости. Форич-бач в той версии не работал.
В другом терминале
ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper bigdataanalytics-worker-3:2181 --list


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
igor@igor-MS-7808:~$ ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
Last login: Sun Jan 16 07:36:18 2022 from 109-252-20-121.nat.spd-mgts.ru
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper bigdataanalytics-worker-3:2181 --list
898_1
__consumer_offsets
cherneev-test
cherneev_test
incident_event_json
orders_json
shadrin_data
shadrin_data_test
shadrin_iris
shadrin_iris_sink
test_lesson2_1
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

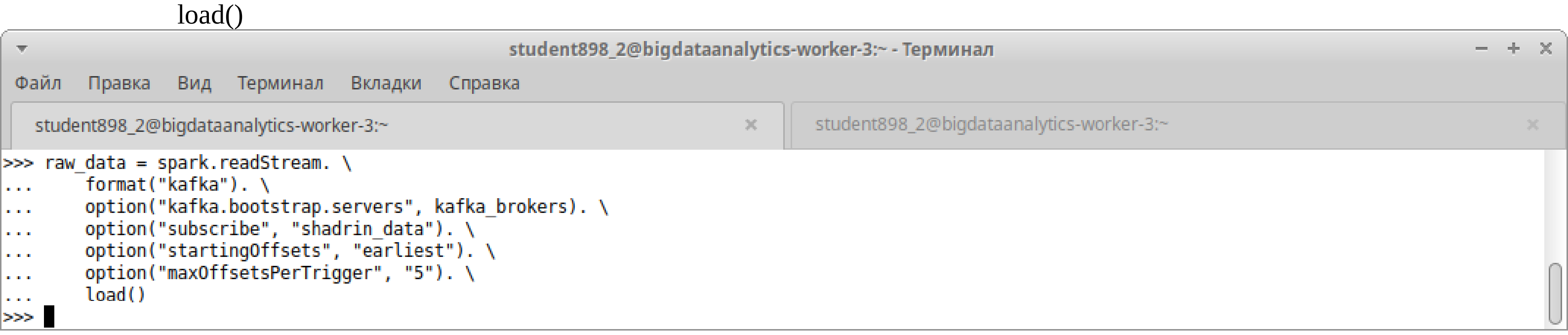
Прочитать топик  shadrin_data
/usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadrin_data --from-beginning --bootstrap-server bigdataanalytics-worker-3:6667 --max-messages 10


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
test_lesson2_1
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadrin_data --from-beginning --bootst
rap-server bigdataanalytics-worker-3:6667 --max-messages 10
/**
 Export to JSON plugin for PHPMyAdmin
 @version 4.6.6deb5
 */

// Database 'cs_project'

// cs_project.data

[{
Processed a total of 10 messages
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

В терминале со спарк
from pyspark.sql import functions as F

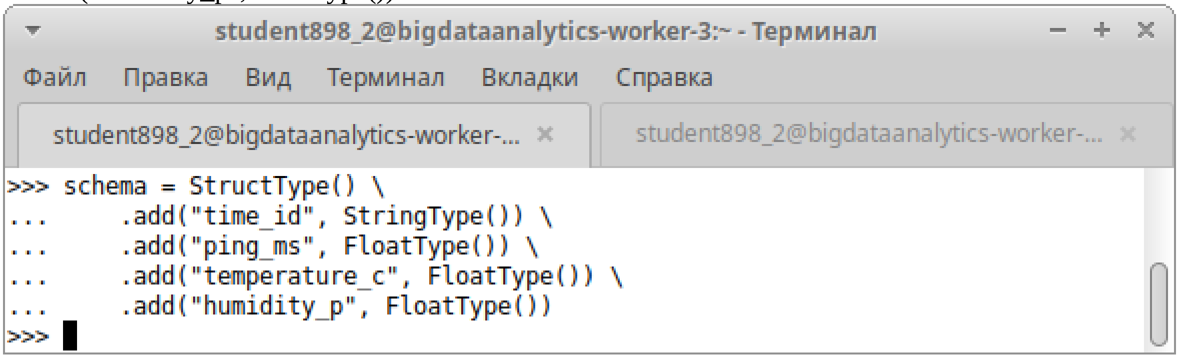
```
from pyspark.sql.types import StructType, StringType, FloatType
kafka_brokers = "bigdataanalytics-worker-3:6667"

raw_data = spark.readStream. \
    format("kafka"). \
    option("kafka.bootstrap.servers", kafka_brokers). \
    option("subscribe", "shadrin_data"). \
    option("startingOffsets", "earliest"). \
    option("maxOffsetsPerTrigger", "5"). \
    load()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
>>> raw_data = spark.readStream. \
...     format("kafka"). \
...     option("kafka.bootstrap.servers", kafka_brokers). \
...     option("subscribe", "shadrin_data"). \
...     option("startingOffsets", "earliest"). \
...     option("maxOffsetsPerTrigger", "5"). \
...     load()
>>> 
```

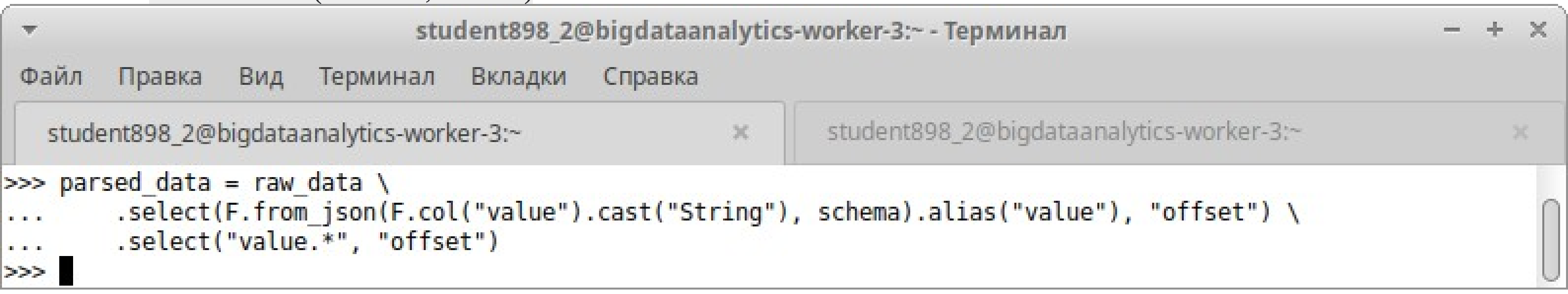
Определяем схему данных нашего исходного датасета.

```
schema = StructType() \
    .add("time_id", StringType()) \
    .add("ping_ms", FloatType()) \
    .add("temperature_c", FloatType()) \
    .add("humidity_p", FloatType())
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-...
student898_2@bigdataanalytics-worker-...

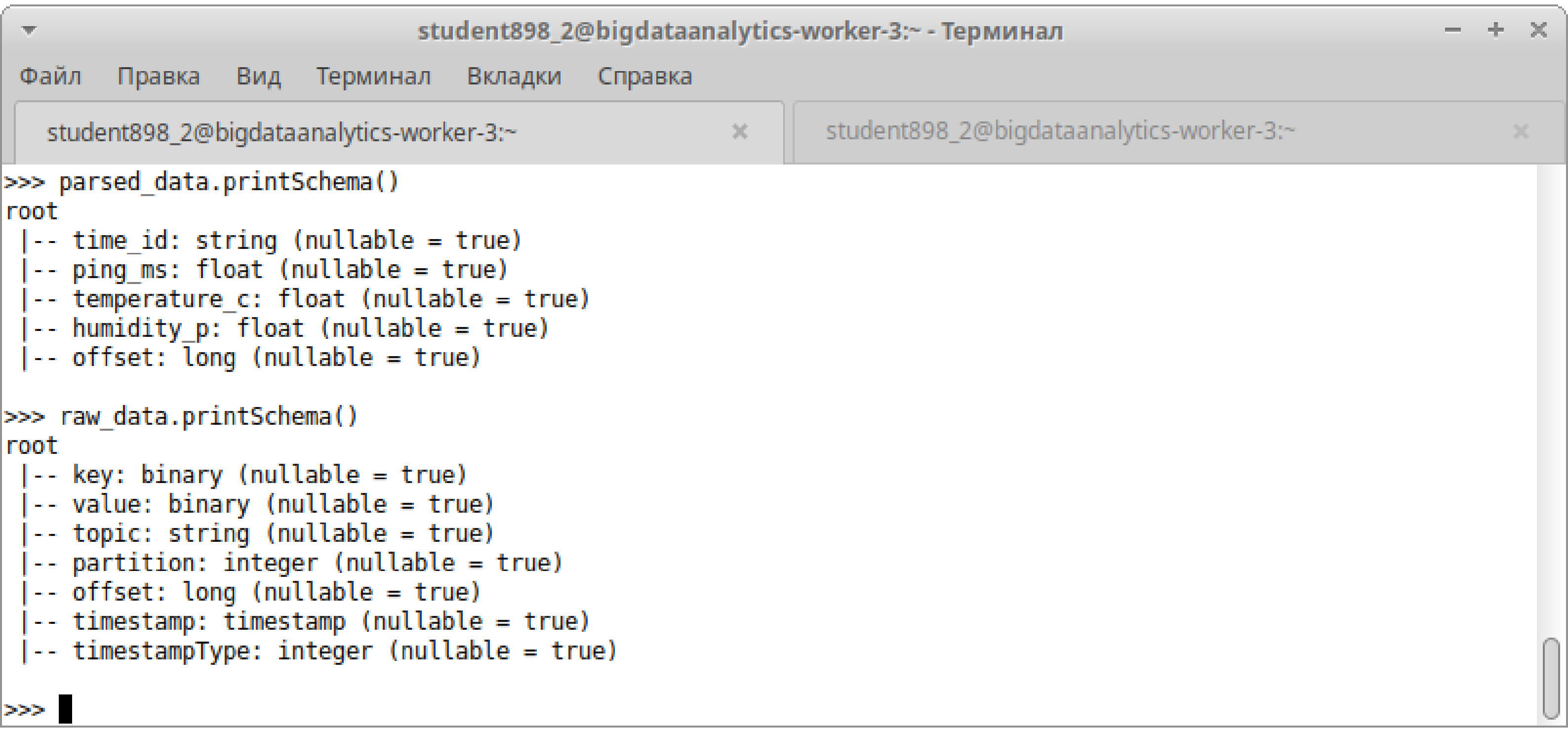
```
>>> schema = StructType() \
...     .add("time_id", StringType()) \
...     .add("ping_ms", FloatType()) \
...     .add("temperature_c", FloatType()) \
...     .add("humidity_p", FloatType())
>>> 
```

```
parsed_data = raw_data \
        .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset") \
        .select("value.*", "offset")
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
>>> parsed_data = raw_data \
...     .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset") \
...     .select("value.*", "offset")
>>> 
```

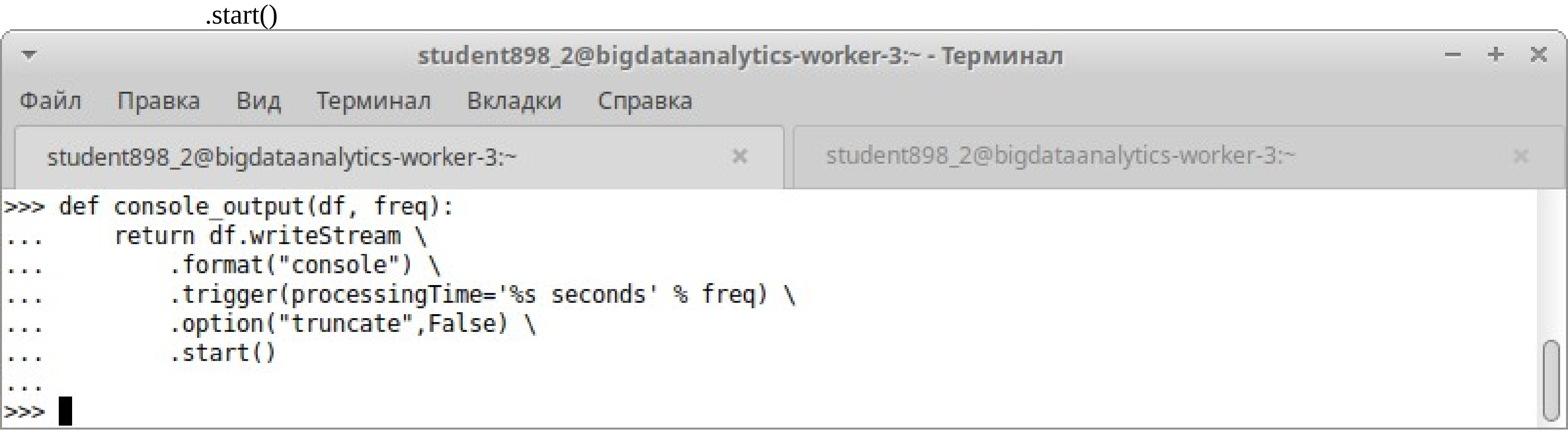
```
parsed_data.printSchema()
raw_data.printSchema()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
>>> parsed_data.printSchema()
root
 |-- time_id: string (nullable = true)
 |-- ping_ms: float (nullable = true)
 |-- temperature_c: float (nullable = true)
 |-- humidity_p: float (nullable = true)
 |-- offset: long (nullable = true)

>>> raw_data.printSchema()
root
 |-- key: binary (nullable = true)
 |-- value: binary (nullable = true)
 |-- topic: string (nullable = true)
 |-- partition: integer (nullable = true)
 |-- offset: long (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- timestampType: integer (nullable = true)

>>> 
```

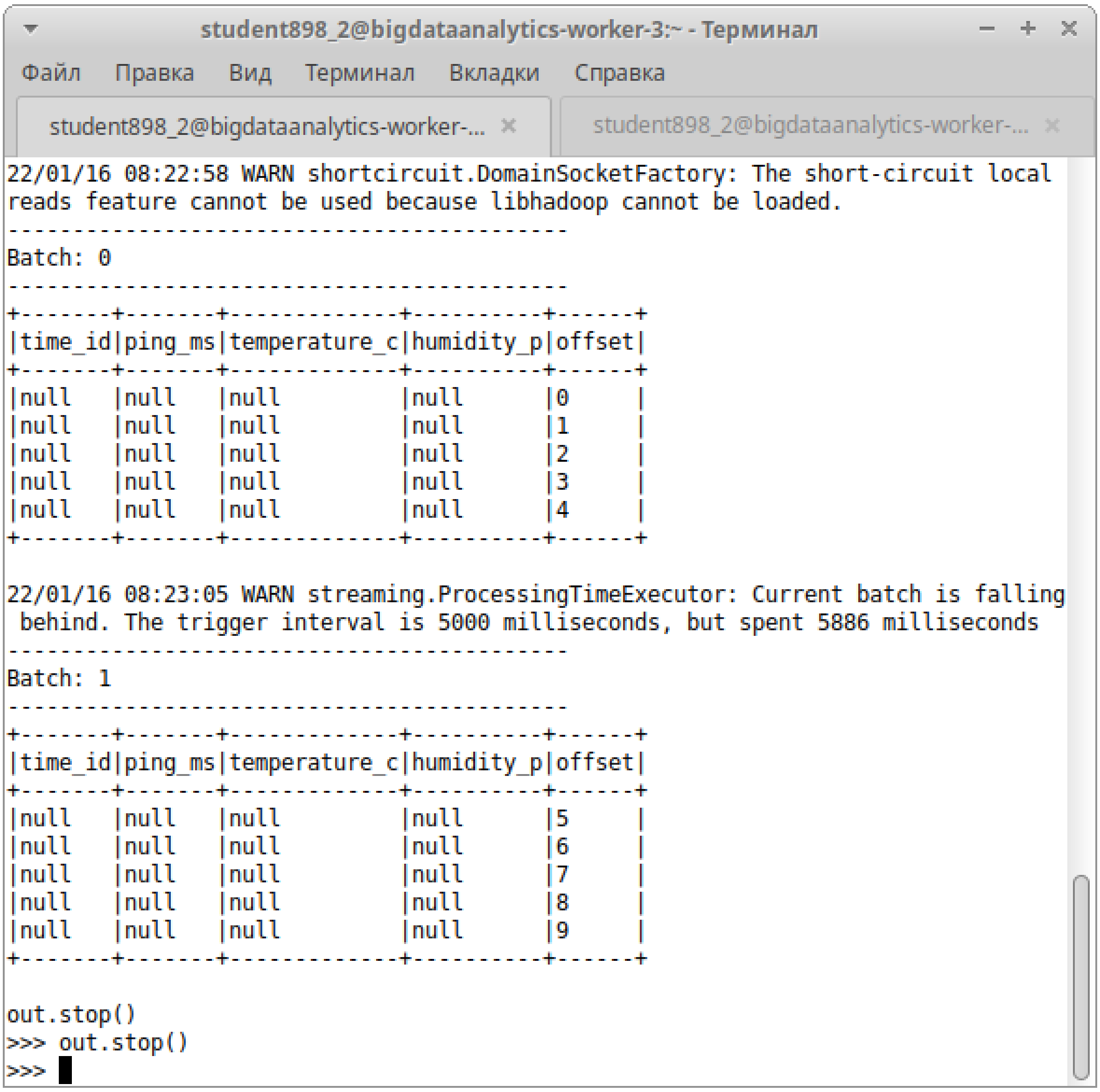
Чекпоинт

```
def console_output(df, freq):
    return df.writeStream \
        .format("console") \
        .trigger(processingTime='%s seconds' % freq) \
        .option("truncate",False) \
        .start()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
>>> def console_output(df, freq):
...     return df.writeStream \
...         .format("console") \
...         .trigger(processingTime='%s seconds' % freq) \
...         .option("truncate",False) \
...         .start()
... 
>>> 
```

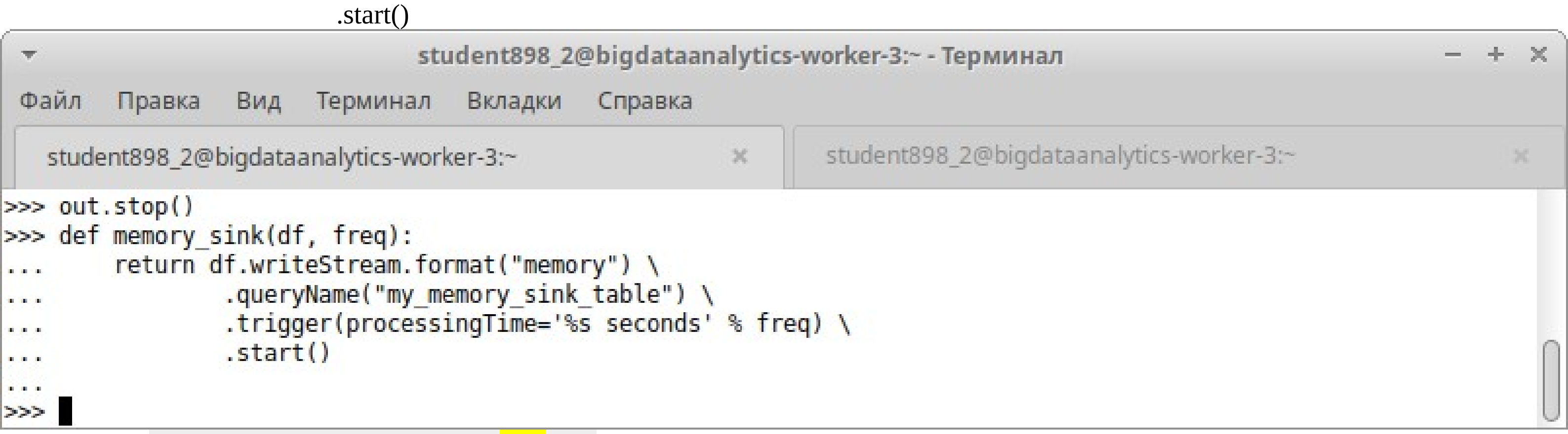
```
out = console_output(parsed_data, 5)
out.stop()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
student898_2@bigdataanalytics-worker-...
student898_2@bigdataanalytics-worker-...

```
22/01/16 08:22:58 WARN shortcircuit.DomainSocketFactory: The short-circuit local
reads feature cannot be used because libhadoop cannot be loaded.
-------------------------------------------
Batch: 0
-------------------------------------------
+-------+-------+-------------+----------+------+
|time_id|ping_ms|temperature_c|humidity_p|offset|
+-------+-------+-------------+----------+------+
|null   |null   |null         |null      |0     |
|null   |null   |null         |null      |1     |
|null   |null   |null         |null      |2     |
|null   |null   |null         |null      |3     |
|null   |null   |null         |null      |4     |
+-------+-------+-------------+----------+------+

22/01/16 08:23:05 WARN streaming.ProcessingTimeExecutor: Current batch is falling
 behind. The trigger interval is 5000 milliseconds, but spent 5886 milliseconds
-------------------------------------------
Batch: 1
-------------------------------------------
+-------+-------+-------------+----------+------+
|time_id|ping_ms|temperature_c|humidity_p|offset|
+-------+-------+-------------+----------+------+
|null   |null   |null         |null      |5     |
|null   |null   |null         |null      |6     |
|null   |null   |null         |null      |7     |
|null   |null   |null         |null      |8     |
|null   |null   |null         |null      |9     |
+-------+-------+-------------+----------+------+

out.stop()
>>> out.stop()
>>> 
```

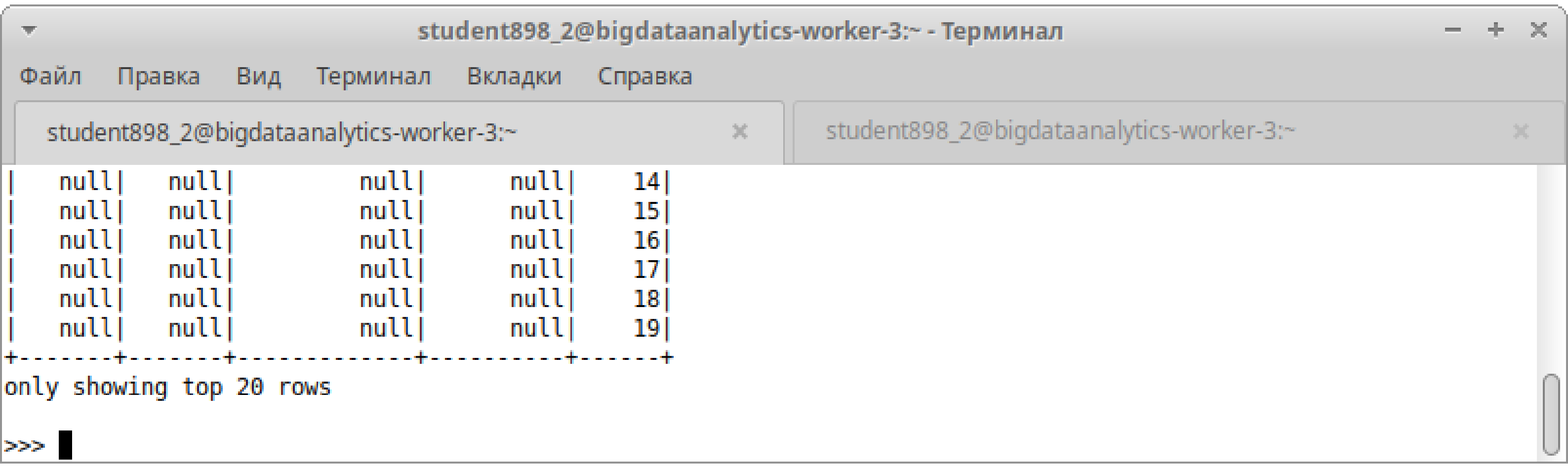
Запись потока в память

```
def memory_sink(df, freq):
        return df.writeStream.format("memory") \
                .queryName("my_memory_sink_table") \
                .trigger(processingTime='%s seconds' % freq) \
                .start()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

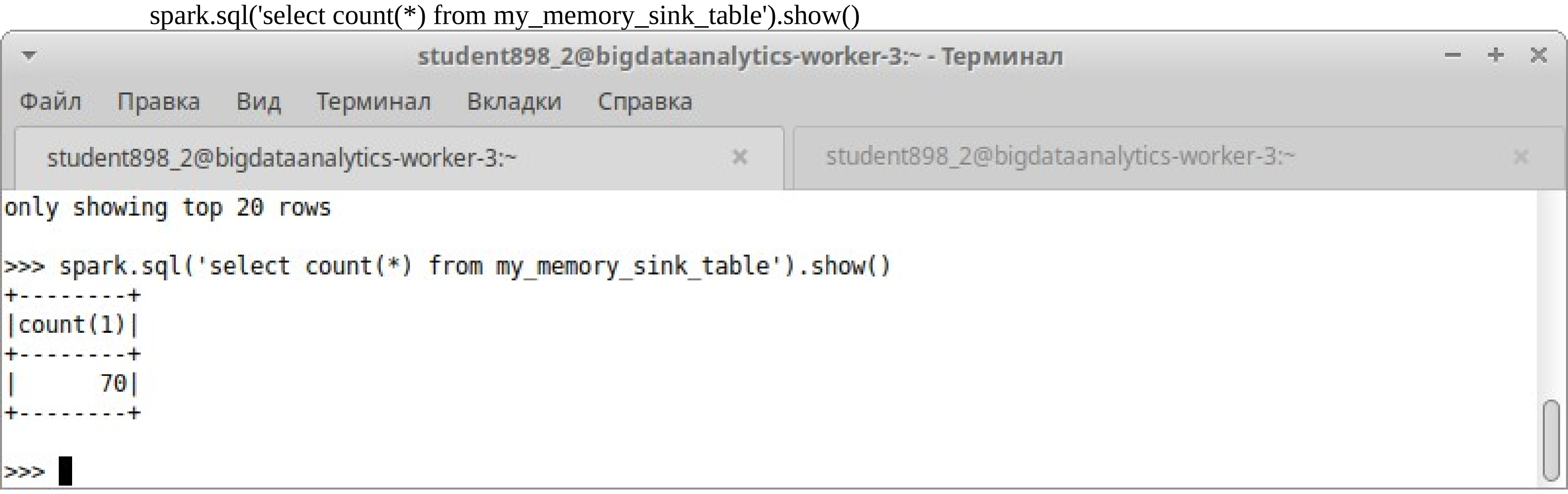
```
>>> out.stop()
>>> def memory_sink(df, freq):
...     return df.writeStream.format("memory") \
...         .queryName("my_memory_sink_table") \
...         .trigger(processingTime='%s seconds' % freq) \
...         .start()
...
>>> 
```

```
stream = memory_sink(parsed_data, 10)
spark.sql("select * from my_memory_sink_table").show()
```

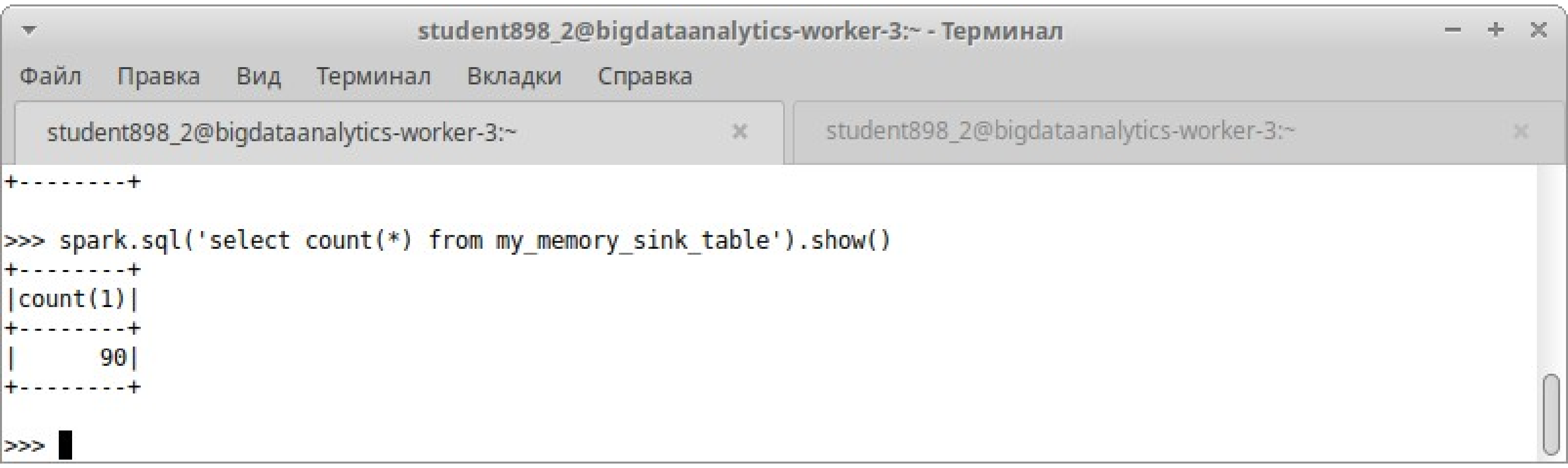

student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
|   null|   null|         null|      null|    14|
|   null|   null|         null|      null|    15|
|   null|   null|         null|      null|    16|
|   null|   null|         null|      null|    17|
|   null|   null|         null|      null|    18|
|   null|   null|         null|      null|    19|
+-------+-------+-------------+----------+------+
only showing top 20 rows

>>> 
```

spark.sql('select count(*) from my_memory_sink_table').show()

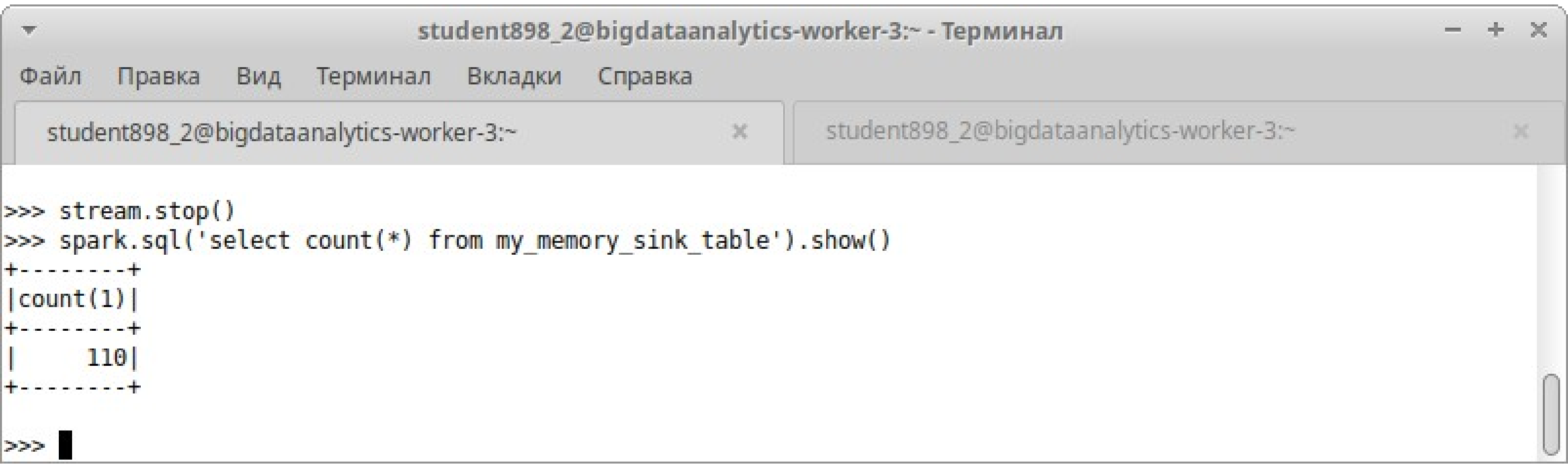

student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
only showing top 20 rows

>>> spark.sql('select count(*) from my_memory_sink_table').show()
+--------+
|count(1)|
+--------+
|      70|
+--------+

>>> 
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

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

>>> spark.sql('select count(*) from my_memory_sink_table').show()
+--------+
|count(1)|
+--------+
|      90|
+--------+

>>> 
```

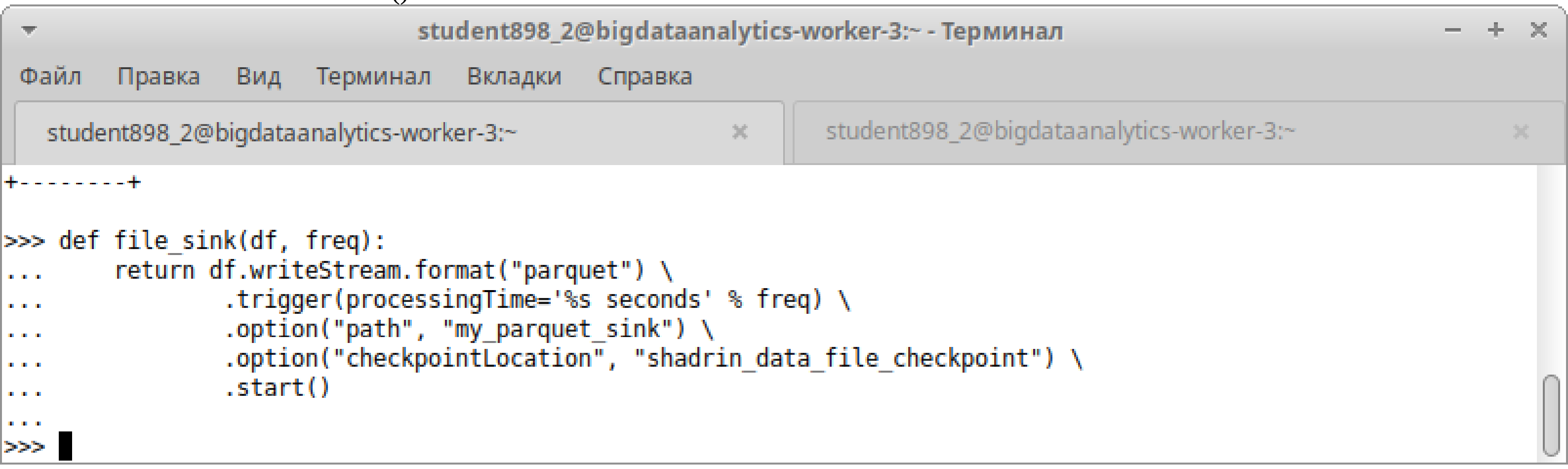
stream.stop()
spark.sql('select count(*) from my_memory_sink_table').show()


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
>>> stream.stop()
>>> spark.sql('select count(*) from my_memory_sink_table').show()
+--------+
|count(1)|
+--------+
|     110|
+--------+

>>> 
```

Запись файла в формат  parquet

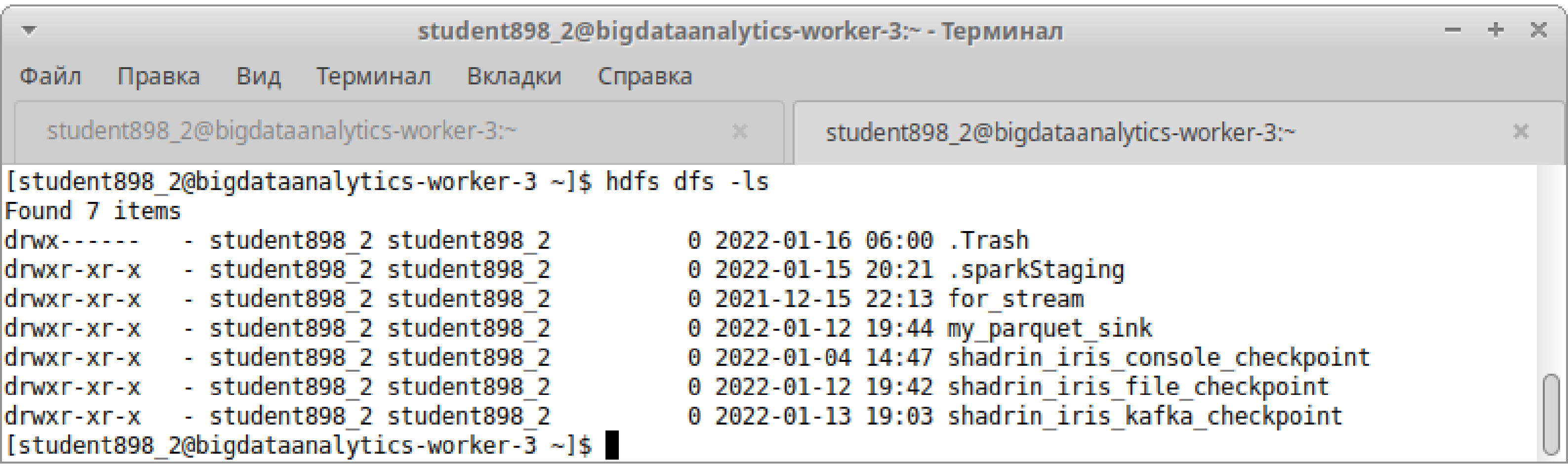
```
def file_sink(df, freq):
        return df.writeStream.format("parquet") \
                .trigger(processingTime='%s seconds' % freq) \
                .option("path", "my_parquet_sink") \
                .option("checkpointLocation", "shadrin_data_file_checkpoint") \
                .start()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

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

>>> def file_sink(df, freq):
...     return df.writeStream.format("parquet") \
...             .trigger(processingTime='%s seconds' % freq) \
...             .option("path", "my_parquet_sink") \
...             .option("checkpointLocation", "shadrin_data_file_checkpoint") \
...             .start()
...
>>> 
```

В другом терминале

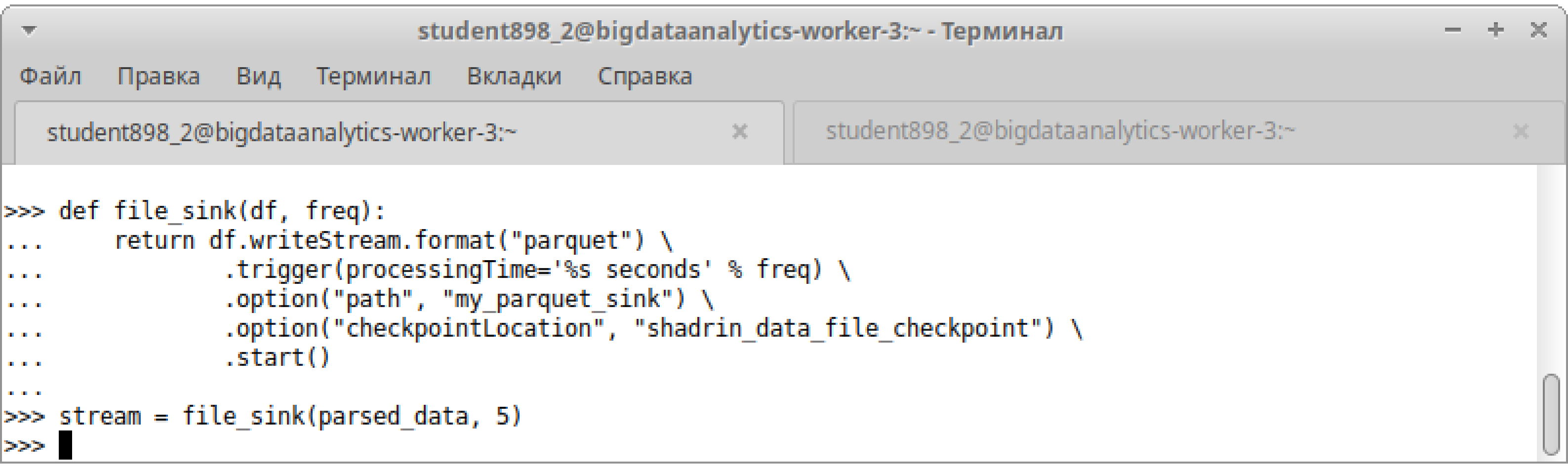
hdfs dfs -ls


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -ls
Found 7 items
drwx------   - student898_2 student898_2          0 2022-01-16 06:00 .Trash
drwxr-xr-x   - student898_2 student898_2          0 2022-01-15 20:21 .sparkStaging
drwxr-xr-x   - student898_2 student898_2          0 2021-12-15 22:13 for_stream
drwxr-xr-x   - student898_2 student898_2          0 2022-01-12 19:44 my_parquet_sink
drwxr-xr-x   - student898_2 student898_2          0 2022-01-04 14:47 shadrin_iris_console_checkpoint
drwxr-xr-x   - student898_2 student898_2          0 2022-01-12 19:42 shadrin_iris_file_checkpoint
drwxr-xr-x   - student898_2 student898_2          0 2022-01-13 19:03 shadrin_iris_kafka_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

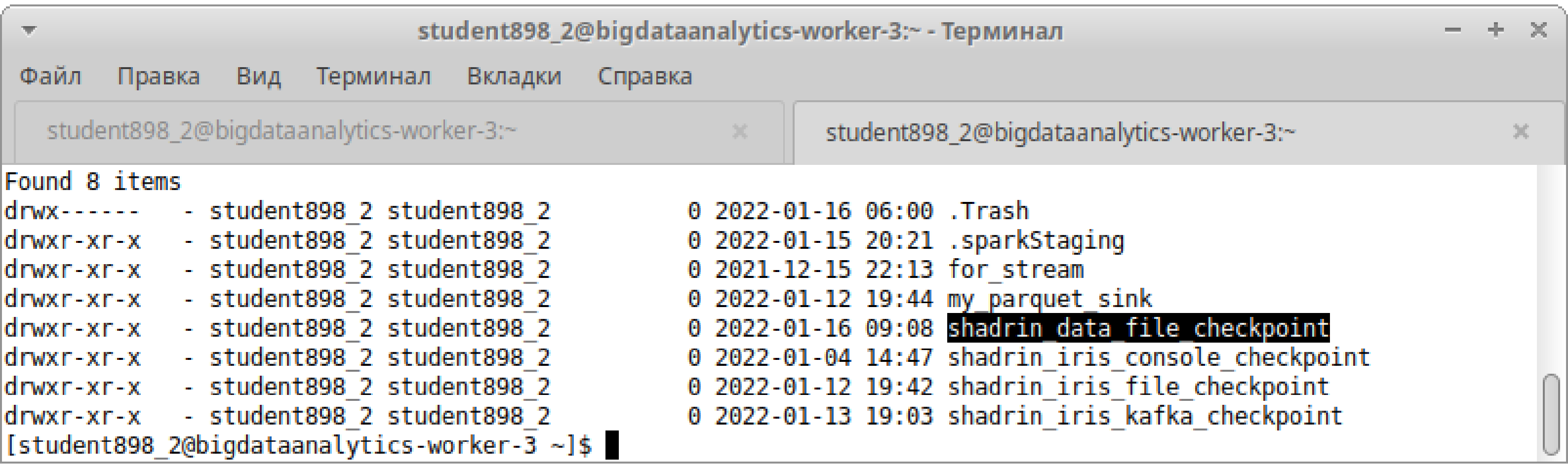
В первом терминале

stream = file_sink(parsed_data, 5)


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

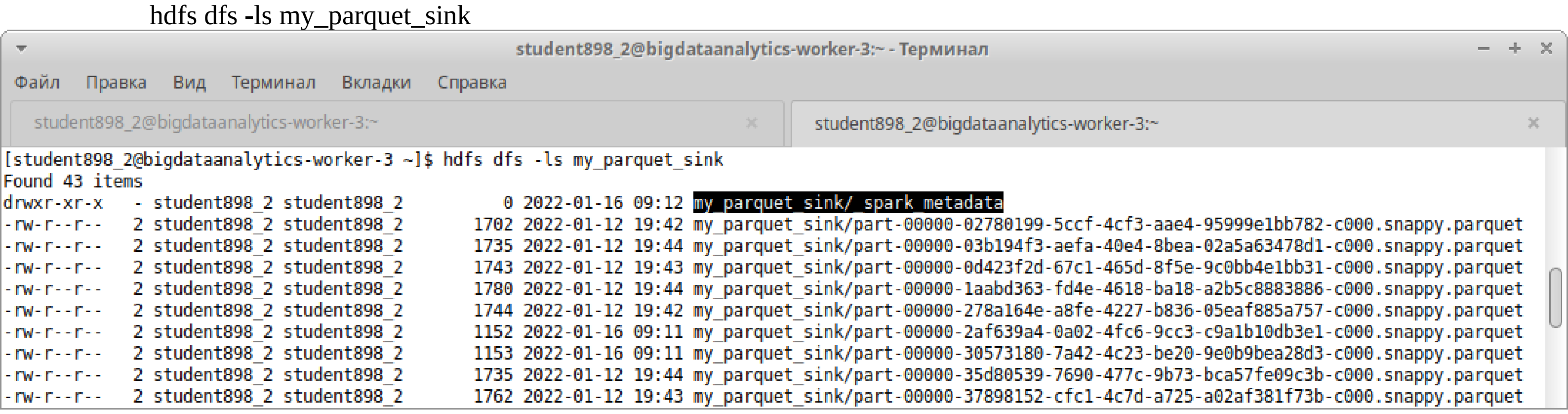
```
>>> def file_sink(df, freq):
...     return df.writeStream.format("parquet") \
...             .trigger(processingTime='%s seconds' % freq) \
...             .option("path", "my_parquet_sink") \
...             .option("checkpointLocation", "shadrin_data_file_checkpoint") \
...             .start()
...
>>> stream = file_sink(parsed_data, 5)
>>>
```

Во втором терминале
hdfs dfs -ls


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
Found 8 items
drwx------   - student898_2 student898_2          0 2022-01-16 06:00 .Trash
drwxr-xr-x   - student898_2 student898_2          0 2022-01-15 20:21 .sparkStaging
drwxr-xr-x   - student898_2 student898_2          0 2021-12-15 22:13 for_stream
drwxr-xr-x   - student898_2 student898_2          0 2022-01-12 19:44 my_parquet_sink
drwxr-xr-x   - student898_2 student898_2          0 2022-01-16 09:08 shadrin_data_file_checkpoint
drwxr-xr-x   - student898_2 student898_2          0 2022-01-04 14:47 shadrin_iris_console_checkpoint
drwxr-xr-x   - student898_2 student898_2          0 2022-01-12 19:42 shadrin_iris_file_checkpoint
drwxr-xr-x   - student898_2 student898_2          0 2022-01-13 19:03 shadrin_iris_kafka_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$
```

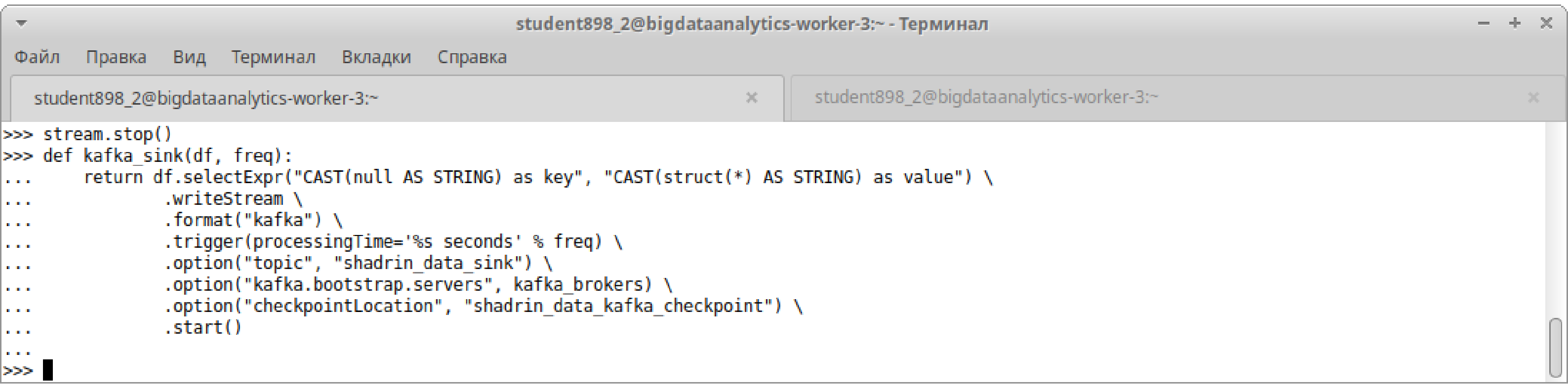
stream.stop()
hdfs dfs -ls my_parquet_sink


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -ls my_parquet_sink
Found 43 items
drwxr-xr-x   - student898_2 student898_2          0 2022-01-16 09:12 my_parquet_sink/_spark_metadata
-rw-r--r--   2 student898_2 student898_2       1702 2022-01-12 19:42 my_parquet_sink/part-00000-02780199-5ccf-4cf3-aae4-95999e1bb782-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1735 2022-01-12 19:44 my_parquet_sink/part-00000-03b194f3-aefa-40e4-8bea-02a5a63478d1-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1743 2022-01-12 19:43 my_parquet_sink/part-00000-0d423f2d-67c1-465d-8f5e-9c0bb4e1bb31-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1780 2022-01-12 19:44 my_parquet_sink/part-00000-1aabd363-fd4e-4618-ba18-a2b5c8883886-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1744 2022-01-12 19:42 my_parquet_sink/part-00000-278a164e-a8fe-4227-b836-05eaf885a757-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1152 2022-01-16 09:11 my_parquet_sink/part-00000-2af639a4-0a02-4fc6-9cc3-c9a1b10db3e1-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1153 2022-01-16 09:11 my_parquet_sink/part-00000-30573180-7a42-4c23-be20-9e0b9bea28d3-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1735 2022-01-12 19:44 my_parquet_sink/part-00000-35d80539-7690-477c-9b73-bca57fe09c3b-c000.snappy.parquet
-rw-r--r--   2 student898_2 student898_2       1762 2022-01-12 19:43 my_parquet_sink/part-00000-37898152-cfc1-4c7d-a725-a02af381f73b-c000.snappy.parquet
```

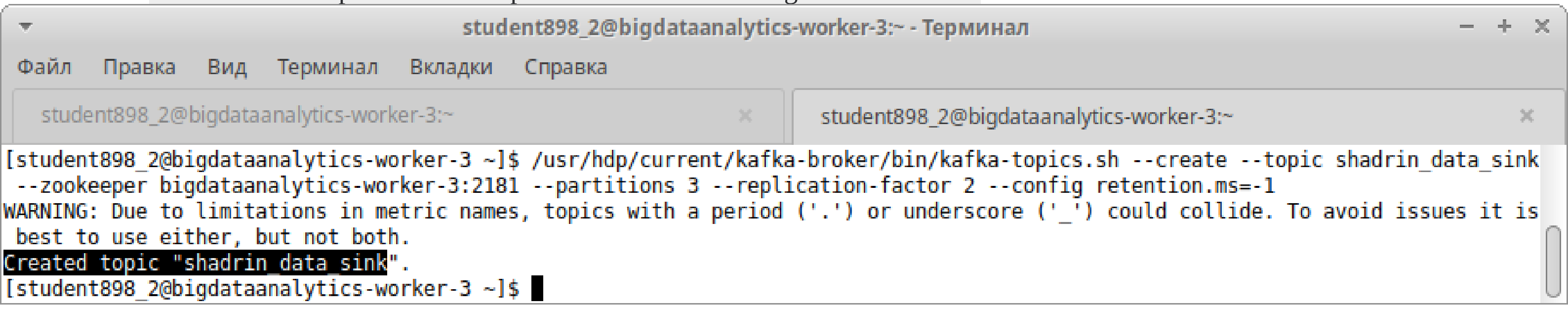
Метод записи из kafka делаем структуру key - value

```
def kafka_sink(df, freq):
        return df.selectExpr("CAST(null AS STRING) as key", "CAST(struct(*) AS STRING) as value") \
                .writeStream \
                .format("kafka") \
                .trigger(processingTime='%s seconds' % freq) \
                .option("topic", "shadrin_data_sink") \
                .option("kafka.bootstrap.servers", kafka_brokers) \
                .option("checkpointLocation", "shadrin_data_kafka_checkpoint") \
                .start()
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

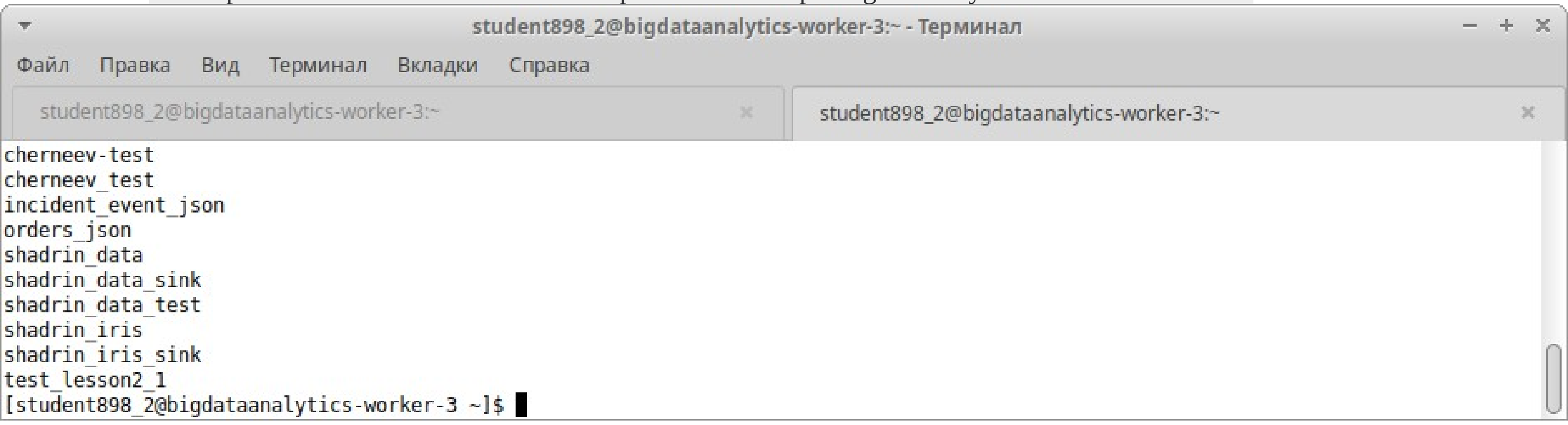
```
>>> stream.stop()
>>> def kafka_sink(df, freq):
...     return df.selectExpr("CAST(null AS STRING) as key", "CAST(struct(*) AS STRING) as value") \
...             .writeStream \
...             .format("kafka") \
...             .trigger(processingTime='%s seconds' % freq) \
...             .option("topic", "shadrin_data_sink") \
...             .option("kafka.bootstrap.servers", kafka_brokers) \
...             .option("checkpointLocation", "shadrin_data_kafka_checkpoint") \
...             .start()
...
>>> 
```

Удалил checkpointLocation
hdfs dfs -rm -f -r shadrin_iris_kafka_checkpoint
Во втором окне терминала создадим топик shadrin_data_sink
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic shadrin_data_sink --zookeeper bigdataanalytics-worker-3:2181 --partitions 3 --replication-factor 2 --config retention.ms=-1


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

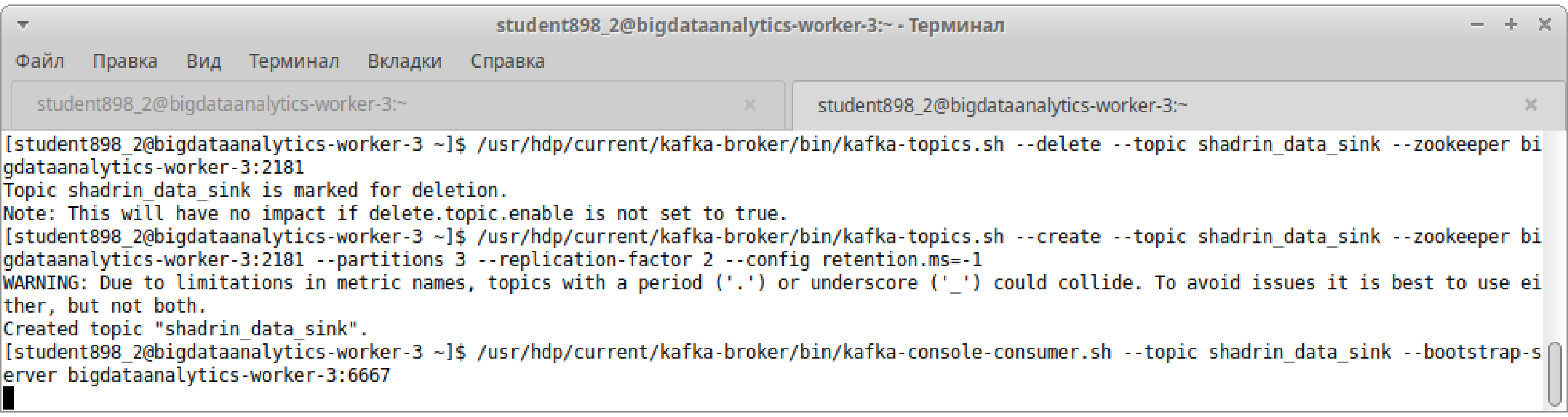
```
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic shadrin_data_sink
 --zookeeper bigdataanalytics-worker-3:2181 --partitions 3 --replication-factor 2 --config retention.ms=-1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is
 best to use either, but not both.
Created topic "shadrin_data_sink".
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper bigdataanalytics-worker-3:2181 --list


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

```
cherneev-test
cherneev_test
incident_event_json
orders_json
shadrin_data
shadrin_data_sink
shadrin_data_test
shadrin_iris
shadrin_iris_sink
test_lesson2_1
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

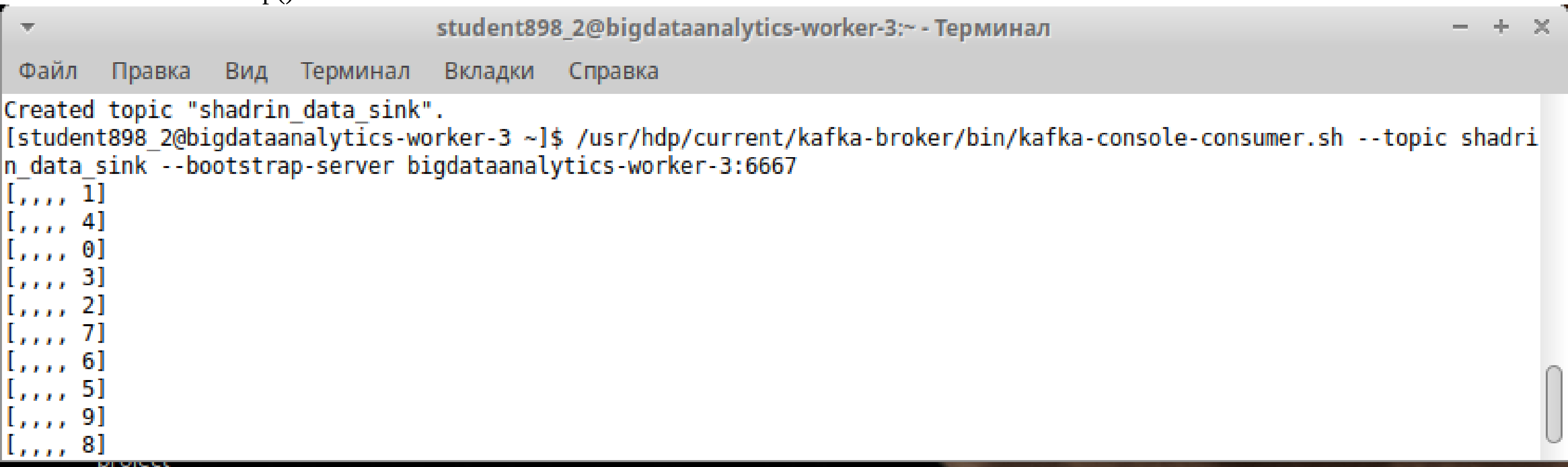
Удалим shadrin_data_sink
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_data_sink --zookeeper bigdataanalytics-worker-3:2181
Создаем shadrin_data_sink
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic shadrin_data_sink --zookeeper bigdataanalytics-worker-3:2181 --partitions 3 --replication-factor 2 --config retention.ms=-1
Подписываемся на его обновления
/usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadrin_data_sink --bootstrap-server bigdataanalytics-worker-3:6667


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл
Правка
Вид
Терминал
Вкладки
Справка
student898_2@bigdataanalytics-worker-3:~
student898_2@bigdataanalytics-worker-3:~

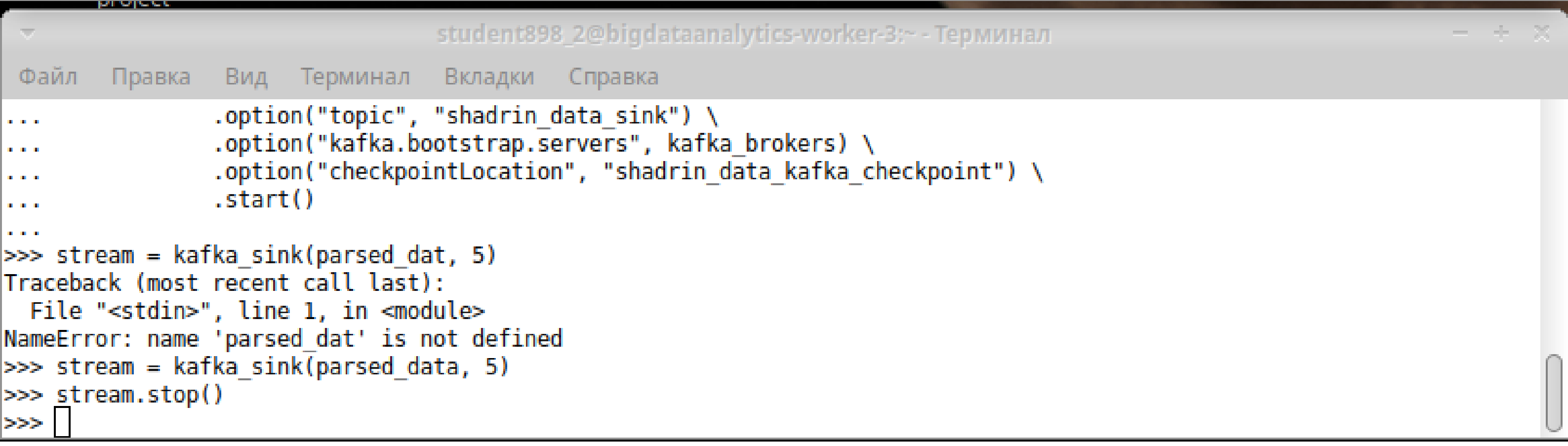
```
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_data_sink --zookeeper bi
gdataanalytics-worker-3:2181
Topic shadrin_data_sink is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic shadrin_data_sink --zookeeper bi
gdataanalytics-worker-3:2181 --partitions 3 --replication-factor 2 --config retention.ms=-1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use ei
ther, but not both.
Created topic "shadrin_data_sink".
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadrin_data_sink --bootstrap-s
erver bigdataanalytics-worker-3:6667
```

Запускаем поток в первой консоли

```
stream = kafka_sink(parsed_data, 5)
stream.stop()
```



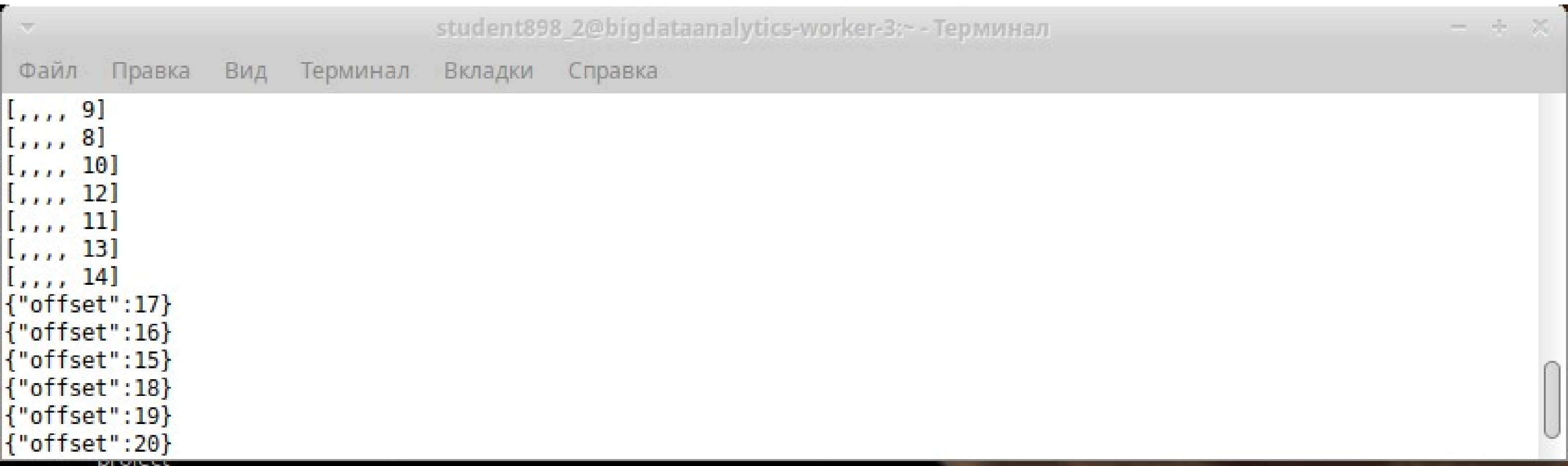
```
Created topic "shadrin_data_sink".
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadri
n_data_sink --bootstrap-server bigdataanalytics-worker-3:6667
[,,,, 1]
[,,,, 4]
[,,,, 0]
[,,,, 3]
[,,,, 2]
[,,,, 7]
[,,,, 6]
[,,,, 5]
[,,,, 9]
[,,,, 8]
```



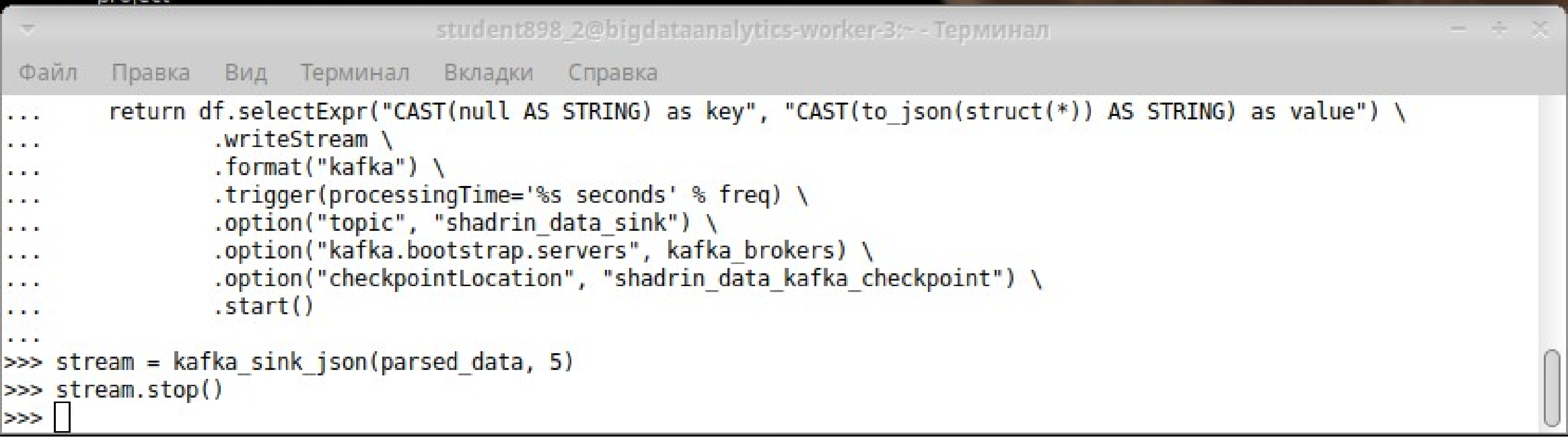
```
...             .option("topic", "shadrin_data_sink") \
...             .option("kafka.bootstrap.servers", kafka_brokers) \
...             .option("checkpointLocation", "shadrin_data_kafka_checkpoint") \
...             .start()
...
>>> stream = kafka_sink(parsed_dat, 5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'parsed_dat' is not defined
>>> stream = kafka_sink(parsed_data, 5)
>>> stream.stop()
>>> 
```

Переключимся в json

```
def kafka_sink_json(df, freq):
        return df.selectExpr("CAST(null AS STRING) as key", "CAST(to_json(struct(*)) AS STRING) as
value") \
                .writeStream \
                .format("kafka") \
                .trigger(processingTime='%s seconds' % freq) \
                .option("topic", "shadrin_data_sink") \
                .option("kafka.bootstrap.servers", kafka_brokers) \
                .option("checkpointLocation", "shadrin_data_kafka_checkpoint") \
                .start()
stream = kafka_sink_json(parsed_data, 5)
stream.stop()
```

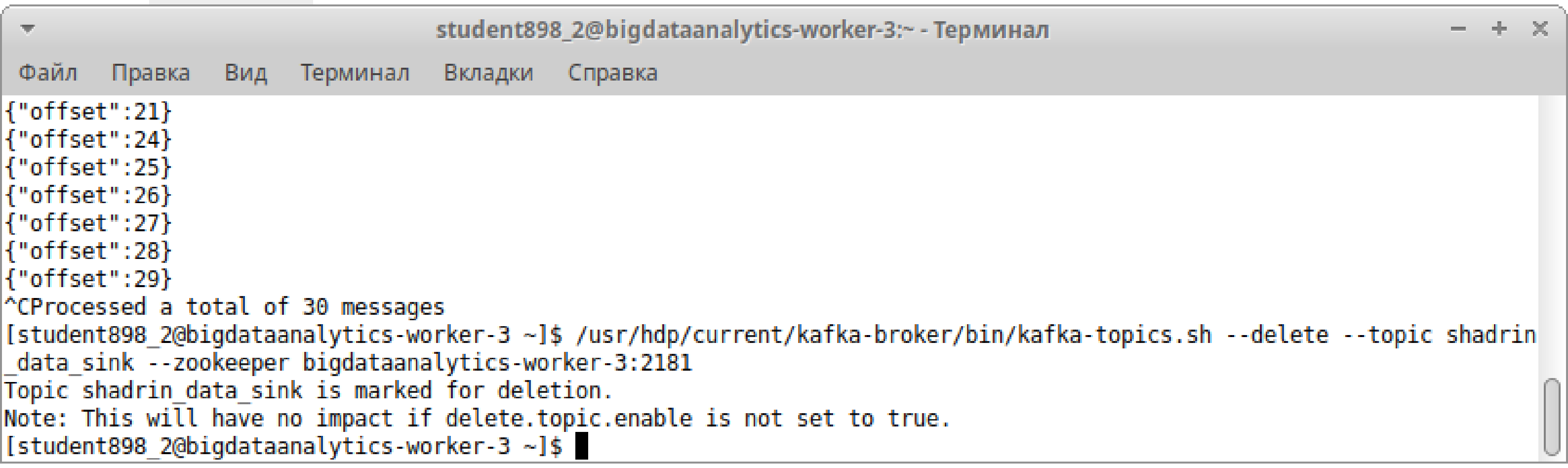

student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл Правка Вид Терминал Вкладки Справка

```
[,,,, 9]
[,,,, 8]
[,,,, 10]
[,,,, 12]
[,,,, 11]
[,,,, 13]
[,,,, 14]
{"offset":17}
{"offset":16}
{"offset":15}
{"offset":18}
{"offset":19}
{"offset":20}
```


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл Правка Вид Терминал Вкладки Справка

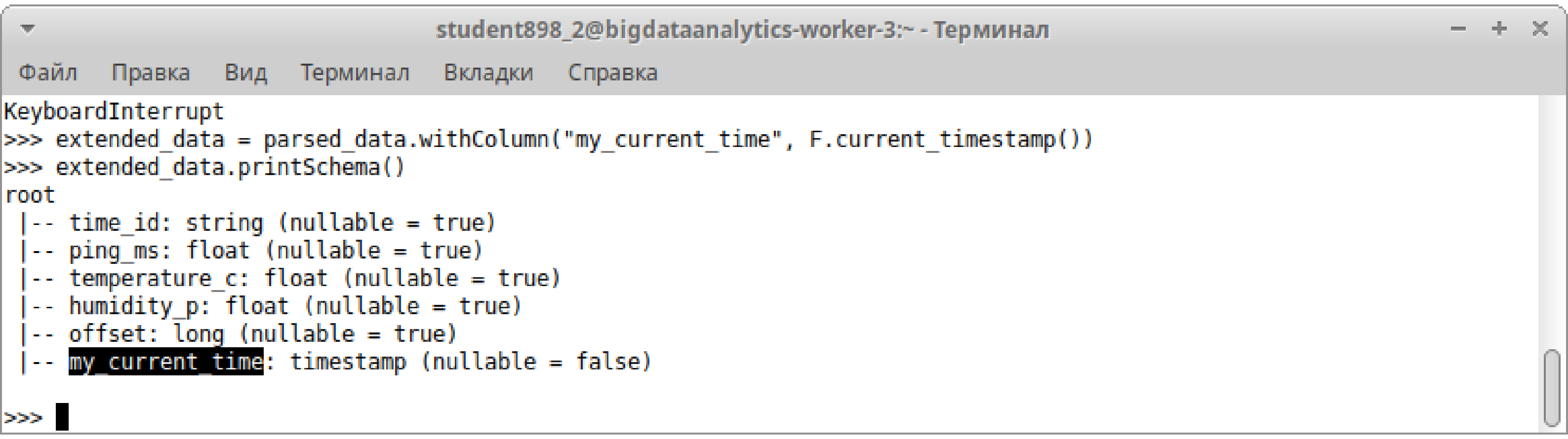
```
...     return df.selectExpr("CAST(null AS STRING) as key", "CAST(to_json(struct(*)) AS STRING) as value") \
...             .writeStream \
...             .format("kafka") \
...             .trigger(processingTime='%s seconds' % freq) \
...             .option("topic", "shadrin_data_sink") \
...             .option("kafka.bootstrap.servers", kafka_brokers) \
...             .option("checkpointLocation", "shadrin_data_kafka_checkpoint") \
...             .start()
...
>>> stream = kafka_sink_json(parsed_data, 5)
>>> stream.stop()
>>> 
```

Удалим shadrin_data_sink
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_data_sink --zookeeper bigdataanalytics-worker-3:2181


student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл Правка Вид Терминал Вкладки Справка

```
{"offset":21}
{"offset":24}
{"offset":25}
{"offset":26}
{"offset":27}
{"offset":28}
{"offset":29}
^CProcessed a total of 30 messages
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin
_data_sink --zookeeper bigdataanalytics-worker-3:2181
Topic shadrin_data_sink is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
[student898_2@bigdataanalytics-worker-3 ~]$ 
```

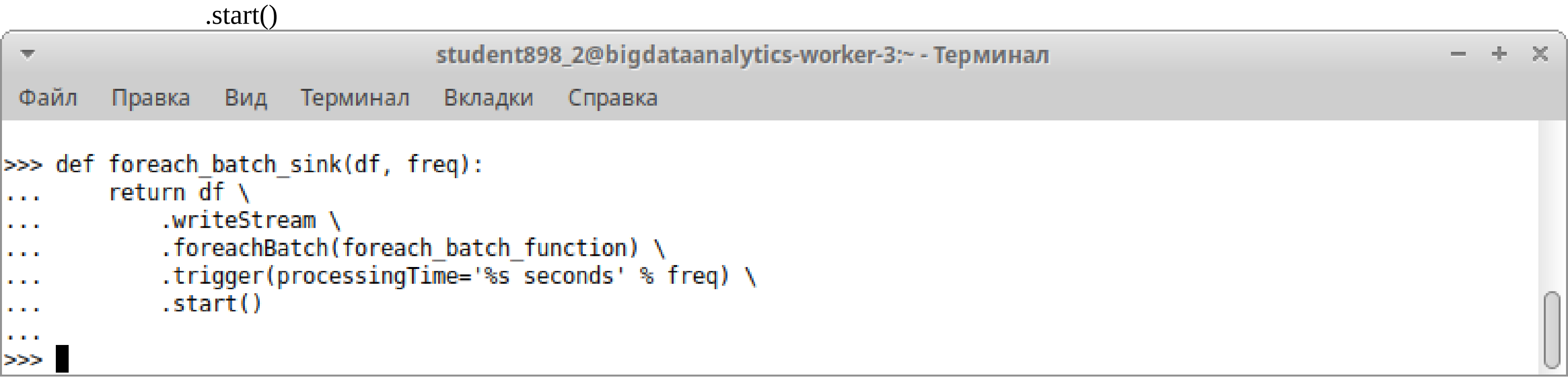
Переходим к  foreach_batch_sink
extended_data = parsed_data.withColumn("my_current_time", F.current_timestamp())
extended_data.printSchema()



```
KeyboardInterrupt
>>> extended_data = parsed_data.withColumn("my_current_time", F.current_timestamp())
>>> extended_data.printSchema()
root
 |-- time_id: string (nullable = true)
 |-- ping_ms: float (nullable = true)
 |-- temperature_c: float (nullable = true)
 |-- humidity_p: float (nullable = true)
 |-- offset: long (nullable = true)
 |-- my_current_time: timestamp (nullable = false)

>>> 
```

Определим функцию понятие формат заменяем на foreach_batch

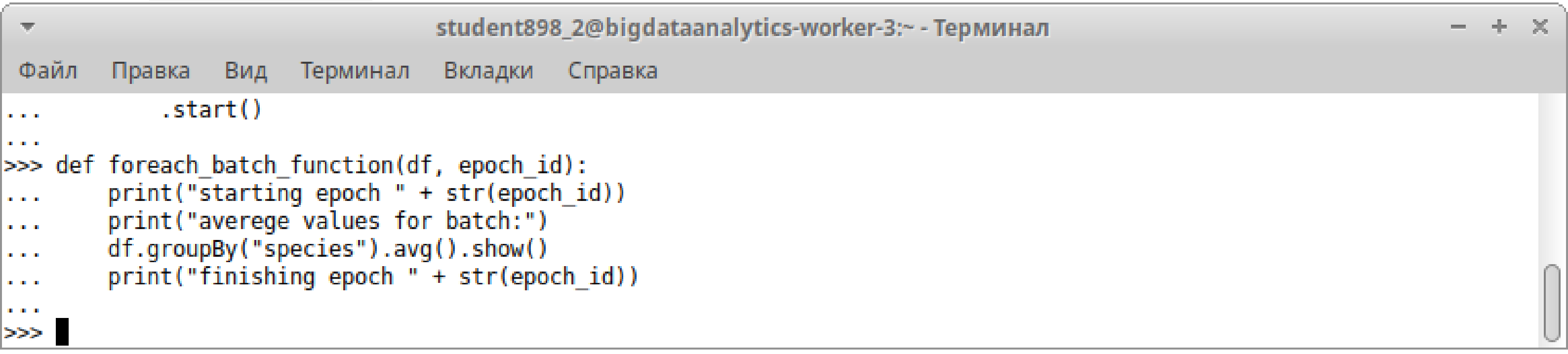
```
def foreach_batch_sink(df, freq):
    return df \
        .writeStream \
        .foreachBatch(foreach_batch_function) \
        .trigger(processingTime='%s seconds' % freq) \
        .start()
```



```
>>> def foreach_batch_sink(df, freq):
...     return df \
...         .writeStream \
...         .foreachBatch(foreach_batch_function) \
...         .trigger(processingTime='%s seconds' % freq) \
...         .start()
...
>>> 
```

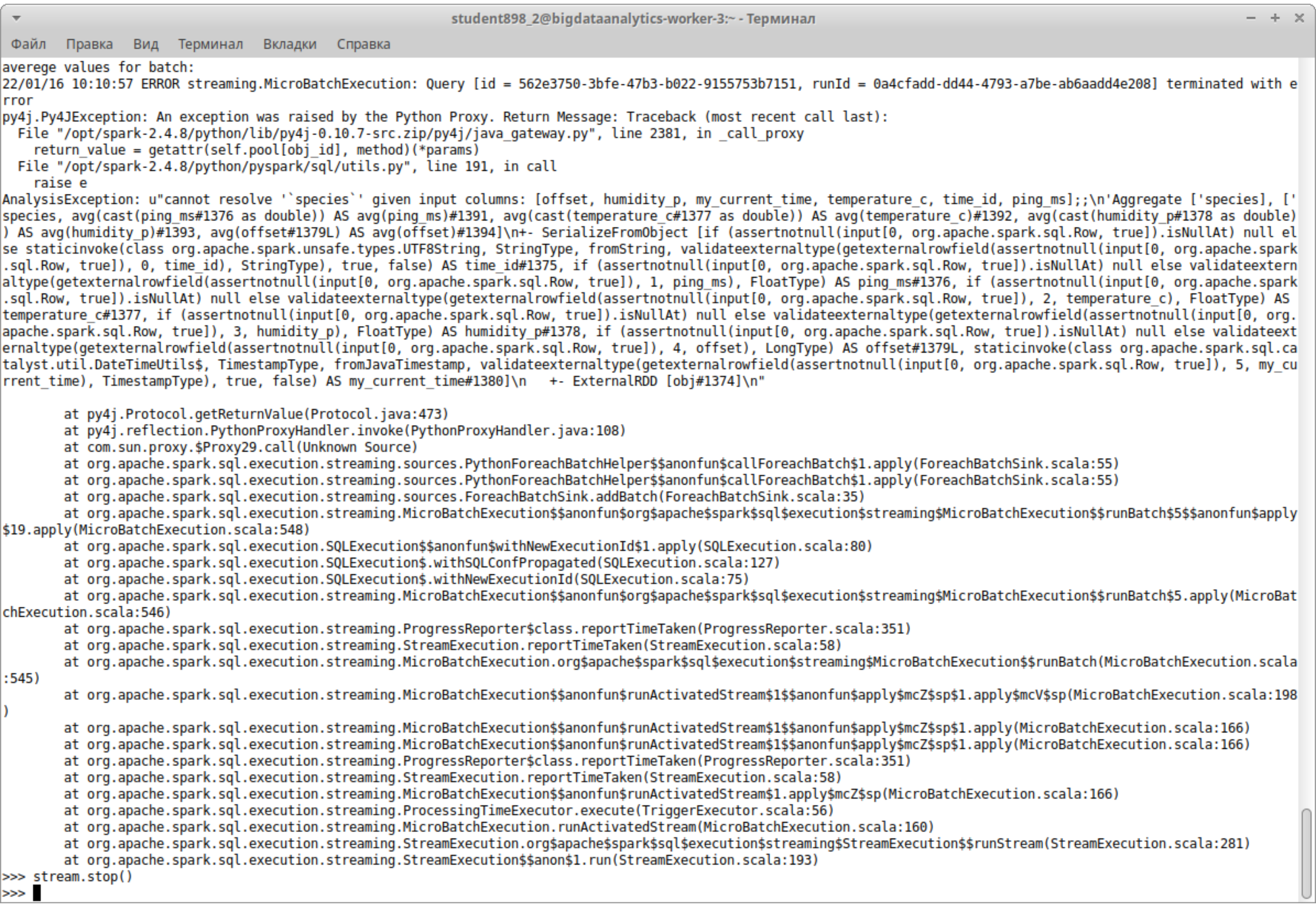
```
def foreach_batch_function(df, epoch_id):
    print("starting epoch " + str(epoch_id))
    print("averege values for batch:")
    df.groupBy("species").avg().show()
    print("finishing epoch " + str(epoch_id))
```

внутри этой функции можно работать как со статическим датасетом и порождать фильтрации, изминения, новый поток и т.д.



```
...         .start()
...
>>> def foreach_batch_function(df, epoch_id):
...     print("starting epoch " + str(epoch_id))
...     print("averege values for batch:")
...     df.groupBy("species").avg().show()
...     print("finishing epoch " + str(epoch_id))
...
>>> 
```

```
stream = foreach_batch_sink(extended_data, 5)
stream.stop()
```



```
averege values for batch:
22/01/16 10:10:57 ERROR streaming.MicroBatchExecution: Query [id = 562e3750-3bfe-47b3-b022-9155753b7151, runId = 0a4cfadd-dd44-4793-a7be-ab6aadd4e208] terminated with e
rror
py4j.Py4JException: An exception was raised by the Python Proxy. Return Message: Traceback (most recent call last):
  File "/opt/spark-2.4.8/python/lib/py4j-0.10.7-src.zip/py4j/java_gateway.py", line 2381, in _call_proxy
    return_value = getattr(self.pool[obj_id], method)(*params)
  File "/opt/spark-2.4.8/python/pyspark/sql/utils.py", line 191, in call
    raise e
AnalysisException: u"cannot resolve '`species`' given input columns: [offset, humidity_p, my_current_time, temperature_c, time_id, ping_ms];;\n'Aggregate ['species], ['
species, avg(cast(ping_ms#1376 as double)) AS avg(ping_ms)#1391, avg(cast(temperature_c#1377 as double)) AS avg(temperature_c)#1392, avg(cast(humidity_p#1378 as double)
) AS avg(humidity_p)#1393, avg(offset#1379L) AS avg(offset)#1394]\n+- SerializeFromObject [if (assertnotnull(input[0, org.apache.spark.sql.Row, true]).isNullAt) null el
se staticinvoke(class org.apache.spark.unsafe.types.UTF8String, StringType, fromString, validateexternaltype(getexternalrowfield(assertnotnull(input[0, org.apache.spark
.sql.Row, true]), 0, time_id), StringType), true, false) AS time_id#1375, if (assertnotnull(input[0, org.apache.spark.sql.Row, true]).isNullAt) null else validateextern
altype(getexternalrowfield(assertnotnull(input[0, org.apache.spark.sql.Row, true]), 1, ping_ms), FloatType) AS ping_ms#1376, if (assertnotnull(input[0, org.apache.spark
.sql.Row, true]).isNullAt) null else validateexternaltype(getexternalrowfield(assertnotnull(input[0, org.apache.spark.sql.Row, true]), 2, temperature_c), FloatType) AS
temperature_c#1377, if (assertnotnull(input[0, org.apache.spark.sql.Row, true]).isNullAt) null else validateexternaltype(getexternalrowfield(assertnotnull(input[0, org.
apache.spark.sql.Row, true]), 3, humidity_p), FloatType) AS humidity_p#1378, if (assertnotnull(input[0, org.apache.spark.sql.Row, true]).isNullAt) null else validateext
ernaltype(getexternalrowfield(assertnotnull(input[0, org.apache.spark.sql.Row, true]), 4, offset), LongType) AS offset#1379L, staticinvoke(class org.apache.spark.sql.ca
talyst.util.DateTimeUtils$, TimestampType, fromJavaTimestamp, validateexternaltype(getexternalrowfield(assertnotnull(input[0, org.apache.spark.sql.Row, true]), 5, my_cu
rrent_time), TimestampType), true, false) AS my_current_time#1380]\n   +- ExternalRDD [obj#1374]\n"

        at py4j.Protocol.getReturnValue(Protocol.java:473)
        at py4j.reflection.PythonProxyHandler.invoke(PythonProxyHandler.java:108)
        at com.sun.proxy.$Proxy29.call(Unknown Source)
        at org.apache.spark.sql.execution.streaming.sources.PythonForeachBatchHelper$$anonfun$callForeachBatch$1.apply(ForeachBatchSink.scala:55)
        at org.apache.spark.sql.execution.streaming.sources.PythonForeachBatchHelper$$anonfun$callForeachBatch$1.apply(ForeachBatchSink.scala:55)
        at org.apache.spark.sql.execution.streaming.sources.ForeachBatchSink.addBatch(ForeachBatchSink.scala:35)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution$$anonfun$org$apache$spark$sql$execution$streaming$MicroBatchExecution$$runBatch$5$$anonfun$apply
$19.apply(MicroBatchExecution.scala:548)
        at org.apache.spark.sql.execution.SQLExecution$$anonfun$withNewExecutionId$1.apply(SQLExecution.scala:80)
        at org.apache.spark.sql.execution.SQLExecution$.withSQLConfPropagated(SQLExecution.scala:127)
        at org.apache.spark.sql.execution.SQLExecution$.withNewExecutionId(SQLExecution.scala:75)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution$$anonfun$org$apache$spark$sql$execution$streaming$MicroBatchExecution$$runBatch$5.apply(MicroBat
chExecution.scala:546)
        at org.apache.spark.sql.execution.streaming.ProgressReporter$class.reportTimeTaken(ProgressReporter.scala:351)
        at org.apache.spark.sql.execution.streaming.StreamExecution.reportTimeTaken(StreamExecution.scala:58)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution.org$apache$spark$sql$execution$streaming$MicroBatchExecution$$runBatch(MicroBatchExecution.scala
:545)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution$$anonfun$runActivatedStream$1$$anonfun$apply$mcZ$sp$1.apply$mcV$sp(MicroBatchExecution.scala:198
)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution$$anonfun$runActivatedStream$1$$anonfun$apply$mcZ$sp$1.apply(MicroBatchExecution.scala:166)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution$$anonfun$runActivatedStream$1$$anonfun$apply$mcZ$sp$1.apply(MicroBatchExecution.scala:166)
        at org.apache.spark.sql.execution.streaming.ProgressReporter$class.reportTimeTaken(ProgressReporter.scala:351)
        at org.apache.spark.sql.execution.streaming.StreamExecution.reportTimeTaken(StreamExecution.scala:58)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution$$anonfun$runActivatedStream$1.apply$mcZ$sp(MicroBatchExecution.scala:166)
        at org.apache.spark.sql.execution.streaming.ProcessingTimeExecutor.execute(TriggerExecutor.scala:56)
        at org.apache.spark.sql.execution.streaming.MicroBatchExecution.runActivatedStream(MicroBatchExecution.scala:160)
        at org.apache.spark.sql.execution.streaming.StreamExecution.org$apache$spark$sql$execution$streaming$StreamExecution$$runStream(StreamExecution.scala:281)
        at org.apache.spark.sql.execution.streaming.StreamExecution$$anon$1.run(StreamExecution.scala:193)
>>> stream.stop()
>>> 
```

ДЗ - повторить действия как на уроке, только со своими данными, использовать свою схему, свой топик в кафке, попробовать как складываются файлы в паркет, в csv, изменить на json загружать в кафку, использовать другие режимы апдате или комплит, не  аппенд. Посмотреть каким ещё образом можно складывать файлы паркет, при этом остановить поток а потом запустить его ещё раз.

Запись/сохранение данных в файл

```
# CSV
data.write.csv('dataset.csv')

# JSON
data.write.save('dataset.json', format='json')

# Parquet
data.write.save('dataset.parquet', format='parquet')
```