

5. Spark Streaming. Stateful streams

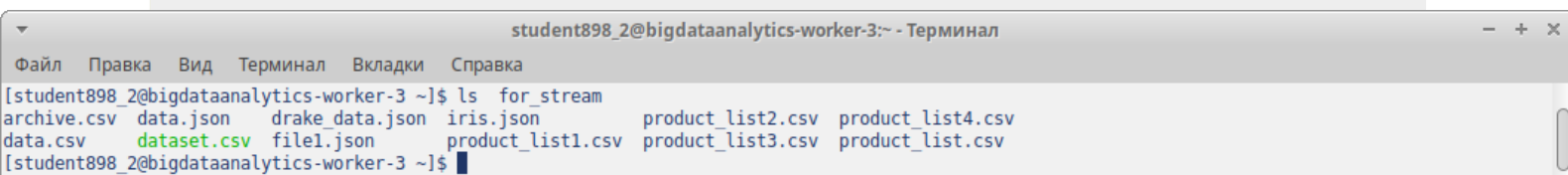
Загрузить в топик kafka свои данные, прочитать их в потоке, применить watermark и window. Повторить шаги выполненные на занятии.

Дополнительно, объединить статичный и динамичный потоки. Задание на повышенный бал: Написать скрипт на python для конвертации файла csv в json.

Подключаемся и проверяем, что файл **data**.csv загрузился.

```
ssh -i ~/.ssh/id_rsa_student898_2 student898\_2@37.139.41.176
```

```
ls for_stream
```

A terminal window titled 'student898_2@bigdataanalytics-worker-3:~ - Терминал' showing the output of the 'ls for_stream' command. The output lists several files: archive.csv, data.json, drake_data.json, iris.json, product_list2.csv, product_list4.csv, data.csv, dataset.csv, file1.json, product_list1.csv, product_list3.csv, and product_list.csv.

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
[student898_2@bigdataanalytics-worker-3 ~]$ ls for_stream
archive.csv  data.json    drake_data.json  iris.json      product_list2.csv  product_list4.csv
data.csv     dataset.csv  file1.json       product_list1.csv  product_list3.csv  product_list.csv
[student898_2@bigdataanalytics-worker-3 ~]$
```

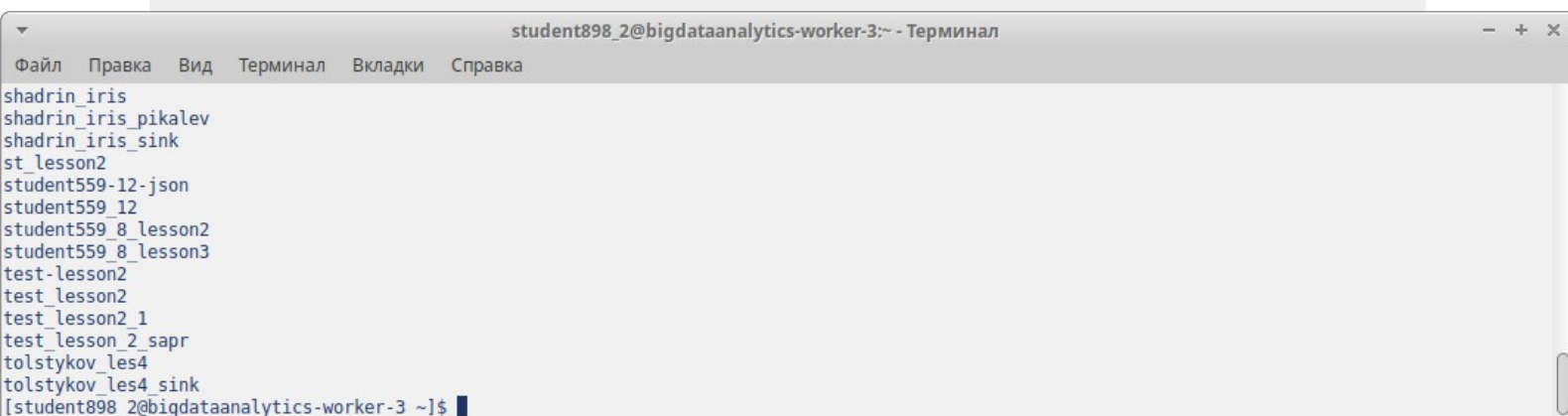
```
less for_stream/data.csv
```

A terminal window titled 'student898_2@bigdataanalytics-worker-3:~ - Терминал' showing the output of the 'less for_stream/data.csv' command. The output displays the first few lines of a CSV file with columns: time_id, ping_ms, temperature_c, humidity_p.

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
"time_id","ping_ms","temperature_c","humidity_p"
"2021-09-30 21:08:02","17.28","25","35"
"2021-09-30 21:09:02","17.73","23","40"
"2021-09-30 21:10:01","18.59","22","41"
"2021-09-30 21:12:02","16.73","22","42"
"2021-09-30 21:13:02","18.12","22","42"
"2021-09-30 21:14:01","18.21","22","43"
"2021-09-30 21:15:01","17.92","22","43"
"2021-09-30 21:16:02","17.2","22","43"
:
[student898_2@bigdataanalytics-worker-3 ~]$
```

смотрим лист топигов

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

A terminal window titled 'student898_2@bigdataanalytics-worker-3:~ - Терминал' showing the output of the 'kafka-topics.sh --zookeeper bigdataanalytics-worker-3:2181 --list' command. The output lists several Kafka topics.

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
shadrin_iris
shadrin_iris_pikalev
shadrin_iris_sink
st_lesson2
student559-12-json
student559_12
student559_8_lesson2
student559_8_lesson3
test-lesson2
test_lesson2
test_lesson2_1
test_lesson_2_sapr
tolstykov_les4
tolstykov_les4_sink
[student898_2@bigdataanalytics-worker-3 ~]$
```

Создаю топик tolstykov_**les5**

```
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic tolstykov_les5 --zookeeper
bigdataanalytics-worker-3:2181 --partitions 1 --replication-factor 1
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic tolstykov_les5 --zookeeper bigdataanalytics-worker-3:2181 --partitions 1 --replication-factor 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 "tolstykov_les5".
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper bigdataanalytics-worker-3:2181 --list
```

Загрузить файл в топик

```
/usr/hdp/current/kafka-broker/bin/kafka-console-producer.sh --broker-list bigdataanalytics-worker-3:6667 --topic tolstikov les5 < for stream/data.csv
```

[illegible]

Прочитать топик [tolstykov_les5](#)

```
/usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic tolstykov_les5 --from-  
beginning --bootstrap-server bigdataanalytics-worker-3:6667 --max-messages 10
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
"time_id","ping_ms","temperature_c","humidity_p"
"2021-09-30 21:08:02","17.28","25","35"
"2021-09-30 21:09:02","17.73","23","40"
"2021-09-30 21:10:01","18.59","22","41"
"2021-09-30 21:12:02","16.73","22","42"
"2021-09-30 21:13:02","18.12","22","42"
"2021-09-30 21:14:01","18.21","22","43"
"2021-09-30 21:15:01","17.92","22","43"
"2021-09-30 21:16:02","17.2","22","43"
"2021-09-30 21:17:02","18.16","22","43"
Processed a total of 10 messages
[student898 2@bigdataanalytics-worker-3 ~]$
```

Запускаем `rusepark`

```
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]
```

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

```
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", "tolstykov_les5"). \

    option("startingOffsets", "earliest"). \

    option("maxOffsetsPerTrigger", "6"). \

    load()

Определяем схему данных нашего исходного датасета.

schema = StructType() \

    .add("time_id", StringType()) \

    .add("ping_ms", StringType()) \

    .add("temperature_c", StringType()) \

    .add("humidity_p", StringType())

def console_output(df, freq):

    return df.writeStream \
```

```

.format("console") \

.trigger(processingTime='%s seconds' % freq ) \

.options(truncate=False) \

.start()

```

```

student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> 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", "tolstykov_les5"). \
...     .option("startingOffsets", "earliest"). \
...     .option("maxOffsetsPerTrigger", "6"). \
...     .load()
>>> schema = StructType() \
...     .add("time_id", StringType()) \
...     .add("ping_ms", StringType()) \
...     .add("temperature_c", StringType()) \
...     .add("humidity_p", StringType())
>>> def console_output(df, freq):
...     return df.writeStream \
...         .format("console") \
...         .trigger(processingTime='%s seconds' % freq ) \
...         .options(truncate=False) \
...         .start()
>>>

```

Сделаем преобразование в плоскую структуру

```

parsed_data = raw_data \

    .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset") \

    .select("value.*", "offset")

out = console_output(raw_data, 10)

out.stop()

```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
+-----+
[null][22 74 69 6D 65 5F 69 64 22 2C 22 70 69 6E 67 5F 6D 73 22 2C 22 74 65 6D 70 65 72 61 74 75 72 65 5F 63 22 2C 22 68 75 6D 69 64 69 74 79 5F 70 22
]|tolstikov_les5|0|0|2022-01-24 18:24:41.358|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 30 38 3A 30 32 22 2C 22 31 37 2E 32 38 22 2C 22 32 35 22 2C 22 33 35 22]
]|tolstikov_les5|0|1|2022-01-24 18:24:41.364|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 30 39 3A 30 32 22 2C 22 31 37 2E 37 33 22 2C 22 32 33 22 2C 22 34 30 22]
]|tolstikov_les5|0|2|2022-01-24 18:24:41.364|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 30 3A 30 31 22 2C 22 31 38 2E 35 39 22 2C 22 32 32 22 2C 22 34 31 22]
]|tolstikov_les5|0|3|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 32 3A 30 32 22 2C 22 31 36 2E 37 33 22 2C 22 32 32 22 2C 22 34 32 22]
]|tolstikov_les5|0|4|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 33 3A 30 32 22 2C 22 31 38 2E 31 32 22 2C 22 32 32 22 2C 22 34 32 22]
]|tolstikov_les5|0|5|2022-01-24 18:24:41.365|0|
+-----+

Batch: 1
+-----+
+-----+
|key|value|topic|partition|
offset|timestamp|timestampType|
+-----+
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 34 3A 30 31 22 2C 22 31 38 2E 32 31 22 2C 22 32 32 22 2C 22 34 33 22]|tolstikov_les5|0|
6|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 35 3A 30 31 22 2C 22 31 37 2E 39 32 22 2C 22 32 32 22 2C 22 34 33 22]|tolstikov_les5|0|
7|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 36 3A 30 32 22 2C 22 31 37 2E 32 22 2C 22 32 32 22 2C 22 34 33 22]|tolstikov_les5|0|
8|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 37 3A 30 32 22 2C 22 31 38 2E 31 36 22 2C 22 32 32 22 2C 22 34 33 22]|tolstikov_les5|0|
9|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 38 3A 30 32 22 2C 22 32 31 2E 33 35 22 2C 22 32 32 22 2C 22 34 32 22]|tolstikov_les5|0|
10|2022-01-24 18:24:41.365|0|
[null][22 32 30 32 31 2D 30 39 2D 33 30 20 32 31 3A 31 39 3A 30 31 22 2C 22 31 37 2E 33 31 22 2C 22 32 32 22 2C 22 34 33 22]|tolstikov_les5|0|
11|2022-01-24 18:24:41.365|0|
+-----+

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

```
extended_data = raw_data \

    .select(F.from_json(F.col("value").cast("String"), schema).alias("value"),
"offset") \

    .select("value.*", "offset") \

    .withColumn("receive_time", F.current_timestamp())

extended_data.printSchema()

def console_output(df, freq):

    return df.writeStream \

        .format("console") \

        .trigger(processingTime='%s seconds' % freq ) \

        .option("checkpointLocation", "checkpoints/duplicates_console_chk") \
```

```
.options(truncate=False) \

.start()
```

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

>>> extended_data = raw_data \
...     .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset") \
...     .select("value.*", "offset") \
...     .withColumn("receive_time", F.current_timestamp())
>>> extended_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)

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

Подключаемся в другом окне, удаляем **checkpoints**

```
hdfs dfs -rm -f -r checkpoints
```

```
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 ~]$ hdfs dfs -rm -f -r checkpoints
22/01/24 18:35:58 INFO fs.TrashPolicyDefault: Moved: 'hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/checkpoints' to trash at: hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/.Trash/Current/user/student898_2/checkpoints
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -ls
Found 9 items
drwx----- student898_2 student898_2 0 2022-01-24 18:35 .Trash
drwxr-xr-x student898_2 student898_2 0 2022-01-20 19:25 .sparkStaging
drwxr-xr-x student898_2 student898_2 0 2021-12-15 22:13 for_stream
drwxr-xr-x student898_2 student898_2 0 2022-01-22 22:34 input_csv_for_stream
drwxr-xr-x student898_2 student898_2 0 2022-01-23 19:15 my_parquet_sink
drwxr-xr-x student898_2 student898_2 0 2022-01-23 19:13 shadrin_iris_file_checkpoint
drwxr-xr-x student898_2 student898_2 0 2022-01-23 19:36 shadrin_iris_kafka_checkpoint
drwxr-xr-x student898_2 student898_2 0 2022-01-22 22:56 tolstykov_les4_file_checkpoint
drwxr-xr-x student898_2 student898_2 0 2022-01-22 23:03 tolstykov_les4_kafka_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$
```

В первом окне запускаем

```
stream = console_output(extended_data , 5)
```

```
stream.stop()
```

Во втором окне наблюдаем

```
hdfs dfs -du -h checkpoints/duplicates_console_chk
```


Файл Правка Вид Терминал Вкладки Справка

```
drwxr-xr-x - student898_2 student898_2 0 2022-01-22 22:56 tolstykov_les4_file_checkpoint
drwxr-xr-x - student898_2 student898_2 0 2022-01-22 23:03 tolstykov_les4_kafka_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -du -h checkpoints/duplicates_console_chk
87      174 checkpoints/duplicates_console_chk/commits
45      90 checkpoints/duplicates_console_chk/metadata
1.3 K   2.5 K checkpoints/duplicates_console_chk/offsets
30      60 checkpoints/duplicates_console_chk/sources
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -du -h checkpoints/duplicates_console_chk
174     348 checkpoints/duplicates_console_chk/commits
45      90 checkpoints/duplicates_console_chk/metadata
2.5 K   5.1 K checkpoints/duplicates_console_chk/offsets
30      60 checkpoints/duplicates_console_chk/sources
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -du -h checkpoints/duplicates_console_chk
203     406 checkpoints/duplicates_console_chk/commits
45      90 checkpoints/duplicates_console_chk/metadata
3.4 K   6.8 K checkpoints/duplicates_console_chk/offsets
30      60 checkpoints/duplicates_console_chk/sources
[student898_2@bigdataanalytics-worker-3 ~]$
```

Файл Правка Вид Терминал Вкладки Справка

```
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+-----+
|null    |null    |null         |null      |42    |2022-01-24 18:39:10.004|
|null    |null    |null         |null      |43    |2022-01-24 18:39:10.004|
|null    |null    |null         |null      |44    |2022-01-24 18:39:10.004|
|null    |null    |null         |null      |45    |2022-01-24 18:39:10.004|
|null    |null    |null         |null      |46    |2022-01-24 18:39:10.004|
|null    |null    |null         |null      |47    |2022-01-24 18:39:10.004|
+-----+-----+-----+-----+-----+-----+
```

Batch: 8

```
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+-----+
|null    |null    |null         |null      |48    |2022-01-24 18:39:15.004|
|null    |null    |null         |null      |49    |2022-01-24 18:39:15.004|
|null    |null    |null         |null      |50    |2022-01-24 18:39:15.004|
|null    |null    |null         |null      |51    |2022-01-24 18:39:15.004|
|null    |null    |null         |null      |52    |2022-01-24 18:39:15.004|
|null    |null    |null         |null      |53    |2022-01-24 18:39:15.004|
+-----+-----+-----+-----+-----+-----+
```

```
stream = c.stop()
>>> stream.stop()
>>> □
```

Данные не читаются

Задаём воте́рмарку, которая должна очищать чекпоинт. Первый параметр - название колонки, на которую смотрит воте́рмарка, второй параметр - гарантированное время жизни информации о сообщении в чекпоинте. Именно для этого мы добавляли столбец `receive_time`.

```
waterwarked_data = extended_data.withWatermark("receive_time", "30 seconds")
```



```
waterwarked_data.printSchema()
```

```
extended_data.printSchema()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> waterwarked_data = extended_data.withWatermark("receive_time", "30 seconds")
>>> waterwarked_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)

>>> extended_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)

>>> █
```

Схема не поменялась. Вотермарка только следит за чекпойнтом, но никак не аффецит наши данные.

Теперь данные можно проверить на наличие дубликатов. Дубли проверяем по двум колонкам: `species` и `receive_time`. Таким образом будут отсеиваться дубли по полю `species` внутри одного микробатча, так как столбец `receive_time` для всех записей внутри этого микробатча одинаковый. Для этого пишем новый датасет `deduplicated_data`

```
deduplicated_data = waterwarked_data.drop_duplicates(["humidity_p",
"receive_time"])
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> deduplicated_data = waterwarked_data.drop_duplicates(["humidity_p", "receive_time"])
>>> █
```

В другом окне удаляем папку `чекпоитс`

```
hdfs dfs -rm -f -r checkpoints
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -rm -f -r checkpoints
22/01/24 19:14:44 INFO fs.TrashPolicyDefault: Moved: 'hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/checkpoints' to trash at: hdfs://
/bigdataanalytics-head-0.mcs.local:8020/user/student898_2/.Trash/Current/user/student898_2/checkpoints1643051684243
[student898_2@bigdataanalytics-worker-3 ~]$ █
```

```
stream = console_output(deduplicated_data , 10)
```

```
stream.stop()
```

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

>>> +-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+-----+
|null   |null   |null   |null   |0     |2022-01-24 19:16:08.083|
+-----+-----+-----+-----+-----+-----+

>>> 22/01/24 19:16:15 WARN streaming.ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 10000 milliseconds, but spent 10422 milliseconds

Batch: 1
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+-----+
|null   |null   |null   |null   |6     |2022-01-24 19:16:15.442|
+-----+-----+-----+-----+-----+-----+

Batch: 2
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+-----+
|null   |null   |null   |null   |12    |2022-01-24 19:16:23.171|
+-----+-----+-----+-----+-----+-----+

Batch: 3
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|
+-----+-----+-----+-----+-----+-----+
|null   |null   |null   |null   |18    |2022-01-24 19:16:30.004|
+-----+-----+-----+-----+-----+-----+

stream.stop()
>>> stream.stop()
>>>
```

Создаём временное окно. В структуру датафрейма добавился новый столбец.

```
windowed_data = extended_data.withColumn("window_time",
F.window(F.col("receive_time"), "2 minutes"))
```

```
windowed_data.printSchema()
```

Мы добавили колонку withColumn, сделали receive_time), "2 minutes


```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> windowed_data = extended_data.withColumn("window_time", F.window(F.col("receive_time"), "2 minutes"))
>>> windowed_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)
|-- window_time: struct (nullable = false)
|   |-- start: timestamp (nullable = true)
|   |-- end: timestamp (nullable = true)
>>>
```

Ещё на это окно надо установить водтермарку

Устанавливаем водтермарку для очистки чекпоинта и удаляем дубли в каждом окне.

```
waterwarked_windowed_data = windowed_data.withWatermark("window_time", "2
minutes")
```

```
deduplicated_windowed_data = waterwarked_windowed_data \

    .drop_duplicates(["humidity_p", "window_time"])
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> waterwarked_windowed_data = windowed_data.withWatermark("window_time", "2 minutes")
>>> deduplicated_windowed_data = waterwarked_windowed_data \
...     .drop_duplicates(["humidity_p", "window_time"])
>>>
```

Сначала надо удвлить чекпинты

Проверяем как удаляются дубли из каждого окна.

```
stream = console_output(deduplicated_windowed_data , 20)

stream.stop()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
if needed...Note that this is normal for the first batch of starting query.
-----
Batch: 4
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|window_time|
+-----+-----+-----+-----+-----+-----+
|null   |null    |null         |null     |24    |2022-01-24 19:22:50.922|[2022-01-24 19:22:00, 2022-01-24 19:24:00]|
+-----+-----+-----+-----+-----+-----+

-----
Batch: 5
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|window_time|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

-----
Batch: 6
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|window_time|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

-----
Batch: 7
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|window_time|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

Traceback (most recent call last):
  File "/opt/spark-2.4.8/python/pyspark/context.py", line 270, in signal_handler
    raise KeyboardInterrupt()
KeyboardInterrupt
>>> stream.stop()
>>>
```

Аналогично предыдущему пункту создаём дополнительное поле `sliding_time`. В функции `F.window` первый аргумент это колонка (временная метка), по которой создаётся окно; второй аргумент - ширина окна; третий - сдвиг окна. Добавляем водермарку и указываем колонки, по которым будем исключать дубли.

```
sliding_data = extended_data.withColumn("sliding_time",
F.window(F.col("receive_time"), "1 minute", "30 seconds"))

waterarked_sliding_data = sliding_data.withWatermark("sliding_time", "2
minutes")

deduplicated_sliding_data =
waterarked_sliding_data.drop_duplicates(["humidity_p", "sliding_time"])

deduplicated_sliding_data.printSchema()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> sliding_data = extended_data.withColumn("sliding_time", F.window(F.col("receive_time"), "1 minute", "30 seconds"))
>>> watermarked_sliding_data = sliding_data.withWatermark("sliding_time", "2 minutes")
>>> deduplicated_sliding_data = watermarked_sliding_data.drop_duplicates(["humidity_p", "sliding_time"])
>>> deduplicated_sliding_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)
|-- sliding_time: struct (nullable = true)
|   |-- start: timestamp (nullable = true)
|   |-- end: timestamp (nullable = true)
>>>
```

очищаем папку чекпоинтов. Запускаем стрим.

```
stream = console_output(deduplicated_sliding_data, 5)

stream.stop()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
-----
Batch: 0
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|sliding_time|
+-----+-----+-----+-----+-----+-----+
|null   |null   |null        |null     |0     |2022-01-24 19:29:48.317|[2022-01-24 19:29:00, 2022-01-24 19:30:00]|
|null   |null   |null        |null     |0     |2022-01-24 19:29:48.317|[2022-01-24 19:29:30, 2022-01-24 19:30:30]|
+-----+-----+-----+-----+-----+-----+

22/01/24 19:29:54 WARN streaming.ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 5000 milliseconds, but spent 9133 milliseconds
-----
Batch: 1
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|sliding_time|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

Batch: 2
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|sliding_time|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

Batch: 3
-----
+-----+-----+-----+-----+-----+-----+
|time_id|ping_ms|temperature_c|humidity_p|offset|receive_time|sliding_time|
+-----+-----+-----+-----+-----+-----+
|null   |null   |null        |null     |18    |2022-01-24 19:30:02.86|[2022-01-24 19:30:00, 2022-01-24 19:31:00]|
+-----+-----+-----+-----+-----+-----+

stream.stop()
-----
Batch: 4
```

Переопределяем метод `console_output` так, чтобы можно было задавать режим вывода результата работы агрегационных функций.

```
def console_output(df, freq, out_mode):

    return df.writeStream.format("console") \

        .trigger(processingTime='%s seconds' % freq ) \

        .options(truncate=False) \

        .option("checkpointLocation", "checkpoints/watermark_console_chk2") \

        .outputMode(out_mode) \

        .start()

waterwarked_windowed_data.printSchema()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
>>> def console_output(df, freq, out_mode):
...     return df.writeStream.format("console") \
...         .trigger(processingTime='%s seconds' % freq) \
...         .options(truncate=False) \
...         .option("checkpointLocation", "checkpoints/watermark_console_chk2") \
...         .outputMode(out_mode) \
...         .start()
...
>>> waterwarked_windowed_data.printSchema()
root
|-- time_id: string (nullable = true)
|-- ping_ms: string (nullable = true)
|-- temperature_c: string (nullable = true)
|-- humidity_p: string (nullable = true)
|-- offset: long (nullable = true)
|-- receive_time: timestamp (nullable = false)
|-- window_time: struct (nullable = false)
|   |-- start: timestamp (nullable = true)
|   |-- end: timestamp (nullable = true)
>>>
```

Сделаем новый датафрейм/стрим

```
count_data = waterwarked_windowed_data.groupBy("window_time").count()
```

ОЧИСТИМ папку чекпоинтов

```
stream = console_output(count_data, 10, "update")
```

```
stream.stop()
```



```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
-----+-----+
|window_time|count|
-----+-----+
|[2022-01-24 19:34:00, 2022-01-24 19:36:00]|6|
-----+-----+

22/01/24 19:35:10 WARN streaming.ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 10000 milliseconds, but spent 10897 milliseconds
-----+-----+
Batch: 1
-----+-----+
|window_time|count|
-----+-----+
|[2022-01-24 19:34:00, 2022-01-24 19:36:00]|12|
-----+-----+

Batch: 2
-----+-----+
|window_time|count|
-----+-----+
|[2022-01-24 19:34:00, 2022-01-24 19:36:00]|18|
-----+-----+

Batch: 3
-----+-----+
|window_time|count|
-----+-----+
|[2022-01-24 19:34:00, 2022-01-24 19:36:00]|24|
-----+-----+

stream.stop()
>>> stream.stop()
>>> █
```

complete

очистим папку чекпоинтов

```
stream = console_output(count_data, 10, "complete")
```

```
stream.stop()
```

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

-----
Batch: 0
-----
+-----+
|window_time|count|
+-----+
|[2022-01-24 19:36:00, 2022-01-24 19:38:00]|6|
+-----+

-----
Batch: 1
-----
+-----+
|window_time|count|
+-----+
|[2022-01-24 19:36:00, 2022-01-24 19:38:00]|12|
+-----+

-----
Batch: 2
-----
+-----+
|window_time|count|
+-----+
|[2022-01-24 19:36:00, 2022-01-24 19:38:00]|18|
+-----+

-----
Batch: 3
-----
+-----+
|window_time|count|
+-----+
|[2022-01-24 19:36:00, 2022-01-24 19:38:00]|24|
+-----+

stream.stop()
>>> stream.stop()
>>> █
```

append

очистим папку чекпоинтов

Пишем все записи только один раз. Информация выводится один раз, когда окно заканчивается.

```
stream = console_output(count_data, 10, "append")
```

```
stream.stop()
```

выходят пустые значения, агрегирующие функции не поддерживаются

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
Batch: 0
-----+-----+
|window_time|count|
|-----+-----+
|-----+-----+
22/01/24 19:39:35 WARN streaming.ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 10000 milliseconds, but spent 16158 milliseconds
-----+-----+
Batch: 1
-----+-----+
|window_time|count|
|-----+-----+
|-----+-----+
Batch: 2
-----+-----+
|window_time|count|
|-----+-----+
|-----+-----+
[Stage 64:=====> (26 + 1) / 200]stream.stop()
22/01/24 19:39:51 ERROR v2.WriteToDataSourceV2Exec: Data source writer org.apache.spark.sql.execution.streaming.sources.MicroBatchWriter@66d7d6f6 is aborting.
22/01/24 19:39:51 ERROR v2.WriteToDataSourceV2Exec: Data source writer org.apache.spark.sql.execution.streaming.sources.MicroBatchWriter@66d7d6f6 aborted.
22/01/24 19:39:51 WARN hdfs.DFSClient: Caught exception
```

Сдвойнить стрим со статикой.

Создадим статический датафрейм, который будет расширять исходный датасет (объединение потоков)

```
static_df_schema = StructType() \
    .add("humidity_p", StringType()) \
    .add("temperature_c", StringType())
```

```
igor@igor-MS-7808:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
stream.stop()
>>> stream.stop()
>>> static_df_schema = StructType() \
...     .add("humidity_p", StringType()) \
...     .add("temperature_c", StringType())
>>> client_loop: send disconnect: Broken pipe
```

Написать скрипт на python для конвертации файла csv в json

importing the required libraries

```
import csv
```

```
import json
```

```
# defining the function to convert CSV file to JSON file

def convjson(csvFilename, jsonFilename):

    # creating a dictionary

    mydata = {}

    # reading the data from CSV file

    with open(csvFilename, encoding = 'utf-8') as csvfile:

        csvRead = csv.DictReader(csvfile)

        # Converting rows into dictionary and adding it to data

        for rows in csvRead:

            mykey = rows['S. No.']

            mydata[mykey] = rows

    # dumping the data

    with open(jsonFilename, 'w', encoding = 'utf-8') as jsonfile:

        jsonfile.write(json.dumps(mydata, indent = 4))

# filenames

csvFilename = r'mydatalist.csv'

jsonFilename = r'mydatalist.json'

# Calling the convjson function
```

```
convjson(csvFilename, jsonFilename)
```

Источник: <https://pythonpip.ru/examples/preobrazovanie-fayla-csv-v-json-v-python>