

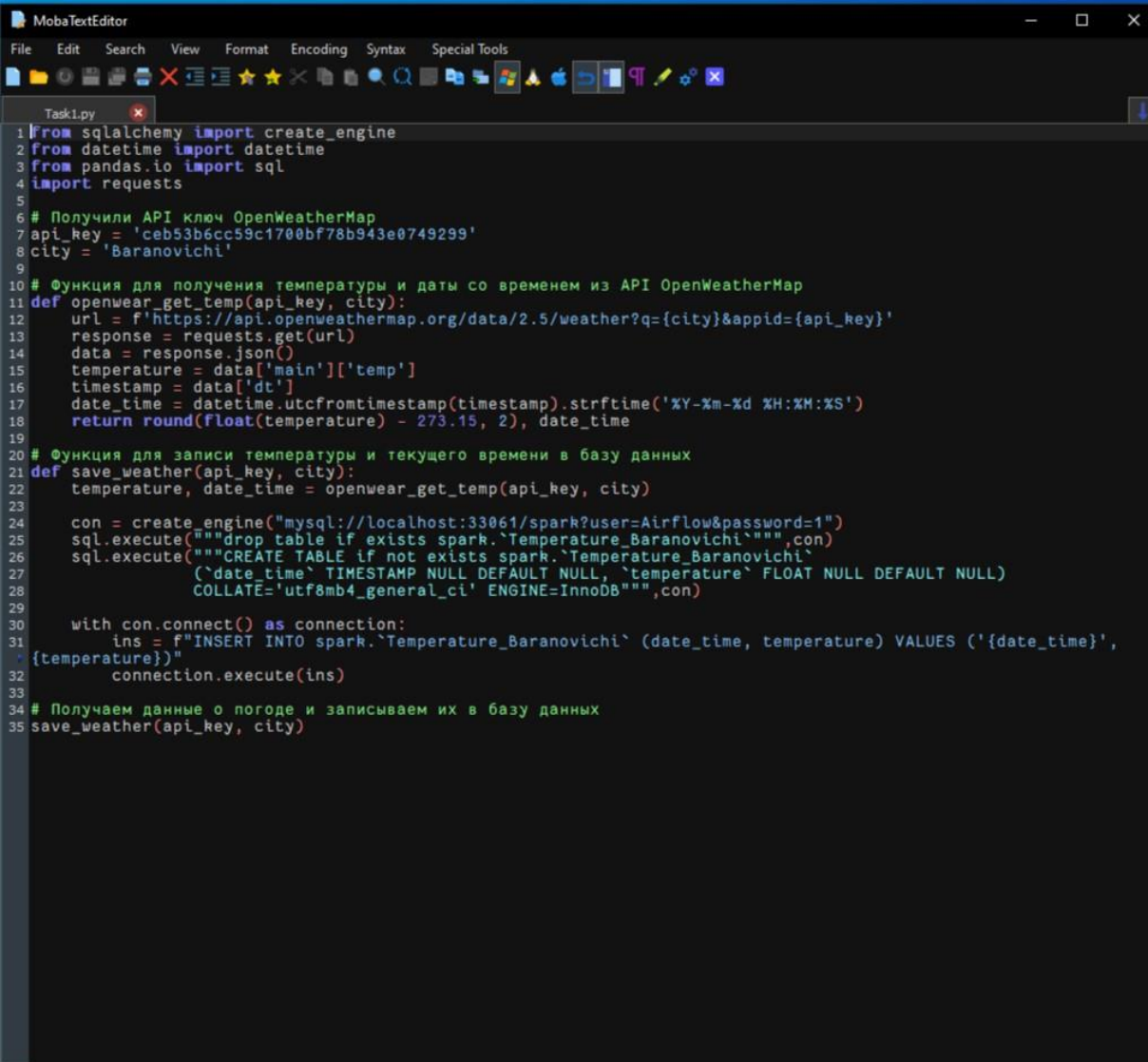
# ETL: автоматизация подготовки данных

## Урок 7. Построение пайплайнов и визуализация потоков данных в Airflow

### Задание

1. Ваша задача с использование пандас, записать полученную температуру в таблицу mysql.

Таблица должна содержать как минимум текущее время и температуру (т.е. два поля). Таблицу не удаляем, используем append.



```
1 from sqlalchemy import create_engine
2 from datetime import datetime
3 from pandas.io import sql
4 import requests
5
6 # Получили API ключ OpenWeatherMap
7 api_key = 'ceb53b6cc59c1700bf78b943e0749299'
8 city = 'Baranovichi'
9
10 # Функция для получения температуры и даты со временем из API OpenWeatherMap
11 def openwear_get_temp(api_key, city):
12     url = f'https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}'
13     response = requests.get(url)
14     data = response.json()
15     temperature = data['main']['temp']
16     timestamp = data['dt']
17     date_time = datetime.utcfromtimestamp(timestamp).strftime('%Y-%m-%d %H:%M:%S')
18     return round(float(temperature) - 273.15, 2), date_time
19
20 # Функция для записи температуры и текущего времени в базу данных
21 def save_weather(api_key, city):
22     temperature, date_time = openwear_get_temp(api_key, city)
23
24     con = create_engine("mysql://localhost:33061/spark?user=Airflow&password=1")
25     sql.execute("""drop table if exists spark.`Temperature_Baranovichi`""", con)
26     sql.execute("""CREATE TABLE if not exists spark.`Temperature_Baranovichi`
27                 (`date_time` TIMESTAMP NULL DEFAULT NULL, `temperature` FLOAT NULL DEFAULT NULL)
28                 COLLATE='utf8mb4_general_ci' ENGINE=InnoDB""", con)
29
30     with con.connect() as connection:
31         ins = f"INSERT INTO spark.`Temperature_Baranovichi` (date_time, temperature) VALUES ('{date_time}',
32         {temperature})"
33         connection.execute(ins)
34
35 # Получаем данные о погоде и записываем их в базу данных
36 save_weather(api_key, city)
```

```

68 #DAGS
69 dag3 = DAG('HomeWork7_Task1',
70 default_args=default_args,
71 description="HomeWork7_1",
72 catchup=False,
73 schedule_interval='0 7 * * *')
74 task51 = BashOperator(
75 task_id='Step_Work_7_1'
76 bash_command='export SPARK_HOME=/home/spark && export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin && python3
- /home/lera/Homework/HW7/Task1.py',
77 dag=dag3)
78 #DAG6
79 from datetime import datetime
80 from airflow import DAG
81 from airflow.operators.bash import BashOperator
82 from airflow.operators.python import PythonOperator, BranchPythonOperator
83 from datetime import datetime, timedelta
84 from airflow.operators.python import PythonOperator
85 from datetime import datetime
86 import requests
87 import os
88
89 def openwear_get_temp(**kwargs):
90
91     ti = kwargs['ti']
92     city = "Baranovichi"
93     api_key = "ceb53b6cc59c1700bf78b943e0749299"
94     url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"
95     payload = {}
96     headers = {}
97     response = requests.request("GET", url, headers=headers, data=payload)
98     return round(float(response.json()['main']['temp'])-273.15, 2)
99
100
101 def openwear_check_temp(ti):
102     temp = int(ti.xcom_pull(task_ids='Baranovichi_get_temperature'))
103     print(f'Temperature now is {temp}')
104     if temp >= 15:
105         return 'Baranovichi_Temp_warm'
106     else:
107         return 'Baranovichi_Temp_cold'
108
109 with DAG(
110     'Baranovichi_check_temperature_warm_or_cold',
111     start_date=datetime(2024, 11, 18),
112     catchup=False,
113     tags=['HomeWork7_weather'],
114 ) as dag:
115     Baranovichi_get_temperature = PythonOperator(
116         task_id='Baranovichi_get_temperature',
117         python_callable=openwear_get_temp,
118     )
119
120     Baranovichi_check_temperature = BranchPythonOperator(
121         task_id='Baranovichi_check_temperature',
122         python_callable=openwear_check_temp,
123     )
124
125     Baranovichi_Temp_warm = BashOperator(
126         task_id='Baranovichi_Temp_warm',
127         bash_command='echo "It is warm"',
128     )
129
130     Baranovichi_Temp_cold = BashOperator(
131         task_id='Baranovichi_Temp_cold',
132         bash_command='echo "It is cold"',
133     )
134
135 Baranovichi_get_temperature >> Baranovichi_check_temperature >> [Baranovichi_Temp_warm,
136 Baranovichi_Temp_cold]
137

```

[DAGs](#)
[Cluster Activity](#)
[Datasets](#)
[Security](#)
[Browse](#)
[Admin](#)
[Docs](#)
19:24 UTC

## DAGs

Active 5
Paused 0
Running 0
Failed 0


Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
AGanshin003	Valerik	<span>1</span>	0 8 * * *	2024-11-17, 19:24:38	2024-11-18, 05:00:00	<span>1</span>	<span>▶</span> <span>🗑️</span>	...
Baranovichi_check_temperature_warm_or_cold	airflow	<span>1</span>	1 day, 0:00:00	2024-11-18, 18:41:45	2024-11-18, 00:00:00	<span>1</span>	<span>▶</span> <span>🗑️</span>	...
HomeWork6_Task2	Valerik	<span>1</span>	0 7 * * *	2024-11-18, 19:16:26	2024-11-18, 04:00:00	<span>1</span>	<span>▶</span> <span>🗑️</span>	...
HomeWork7_Task1	Valerik	<span>1</span>	0 7 * * *	2024-11-18, 19:04:14	2024-11-18, 04:00:00	<span>1</span>	<span>▶</span> <span>🗑️</span>	...
hello_world	airflow	<span>2</span>	0 12 * * *	2024-11-17, 12:00:00	2024-11-18, 12:00:00	<span>2</span>	<span>▶</span> <span>🗑️</span>	...

Showing 1-5 of 5 DAGs

Version: v2.7.3  
Git Version: f1242537838516b8bb8156130bc001595bfbbeb01

**Airflow** DAGs Cluster Activity Datasets Security Browse Admin Docs 19:06 UTC

**DAG: HomeWork7\_Task1** HomeWork7\_1

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code Audit Log

18.11.2024 19:04:23 25 All Run Types All Run States Clear Filters

Press **Shift** + **/** for Shortcuts

**HomeWork7\_Task1** Run 2024-11-18, 04:00:00 UTC / **Step\_Work\_7\_1** Task

Details Graph Gantt <> Code Logs

(by attempts)

1

All Levels

```
[2024-11-18, 18:48:20 UTC] (taskinstance.py:1159) INFO - Dependencies all met for dep_co
[2024-11-18, 18:48:20 UTC] (taskinstance.py:1159) INFO - Dependencies all met for dep_co
[2024-11-18, 18:48:20 UTC] (taskinstance.py:1361) INFO - Starting attempt 1 of 1
[2024-11-18, 18:48:22 UTC] (taskinstance.py:1382) INFO - executing <task(BashOperator): S
[2024-11-18, 18:48:22 UTC] (standard_task_runner.py:57) INFO - started process 795 to run
[2024-11-18, 18:48:22 UTC] (standard_task_runner.py:64) INFO - Running: ["airflow", "task
[2024-11-18, 18:48:23 UTC] (standard_task_runner.py:85) INFO - Job 220: Subtask Step_Work_
[2024-11-18, 18:48:25 UTC] (task_command.py:416) INFO - Running <taskinstance: HomeWork7_
[2024-11-18, 18:48:29 UTC] (taskinstance.py:1662) INFO - Exporting env vars: AIRFLOW_CTX
[2024-11-18, 18:48:29 UTC] (subprocess.py:63) INFO - Temp dir root location: /tmp
[2024-11-18, 18:48:29 UTC] (subprocess.py:75) INFO - Running command: ['/usr/bin/bash',
[2024-11-18, 18:48:29 UTC] (subprocess.py:86) INFO - Output:
[2024-11-18, 18:48:30 UTC] (subprocess.py:93) INFO - File "/home/lera/homeWork7/Task1
[2024-11-18, 18:48:30 UTC] (subprocess.py:93) INFO - SyntaxError: Non-UTF-8 code starting
[2024-11-18, 18:48:30 UTC] (subprocess.py:97) INFO - Command exited with return code 1
[2024-11-18, 18:48:30 UTC] (taskinstance.py:1937) ERROR - Task failed with exception
Traceback (most recent call last):
  File "/home/lera/airflow-venv/lib/python3.8/site-packages/airflow/operators/bash.py", l
    raise AirflowException(
airflow.exceptions.AirflowException: Bash command failed. The command returned a non-zero
[2024-11-18, 18:48:30 UTC] (taskinstance.py:1440) INFO - Marking task as FAILED. dag_id=
[2024-11-18, 18:48:30 UTC] (standard_task_runner.py:104) ERROR - Failed to execute job 22
[2024-11-18, 18:48:30 UTC] (local_task_runner.py:128) INFO - Task exited with return
[2024-11-18, 18:48:30 UTC] (taskinstance.py:2778) INFO - 0 downstream tasks scheduled fr
```

Unamed-1 (spark) Temperature\_Baranovich - HeidiSQL 12.8.0.6908

Unamed-1

1.9 MB

spark:Temperature >> Далее Показать все Сортировка Столбцы (2/2) Фильтр

#	date_time	temperature
1	2024-11-18 19:04:20	2.25

39 SELECT \* FROM "spark"."Temperature\_Baranovich" LIMIT 1000;

r1: c2 Подключен MySQL 8.0.40 Время работы: 00:30 h Серверное в Ожидание.

**Airflow** DAGs Cluster Activity Datasets Security Browse Admin Docs 18:43 UTC

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code Audit Log

18.11.2024 18:41:56 25 All Run Types All Run States Clear Filters

Press **Shift** + **/** for Shortcuts

deferred failed queued removed restarting running scheduled skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

**Baranovich\_check\_temperature\_warm\_or\_cold** Run 2024-11-18, 18:41:45 UTC / **Baranovich\_Temp\_cold** Task

Clear task Mark state as... Filter Tasks

Details Graph Gantt <> Code Logs

Layout: Left -> Right

Baranovich\_get\_temperature PythonOperator success

Baranovich\_check\_temper... BranchPythonOperator success

Baranovich\_Temp\_cold BashOperator success

Baranovich\_Temp\_warm BashOperator skipped

Airflow

DAGs

Cluster Activity

Datasets

Security

Browse

Admin

Docs

18:43 UTC

AA

Grid

Graph

Calendar

Task Duration

Task Tries

Landing Times

Gantt

Details

<> Code

Audit Log

18.11.2024 18:41:56

25

All Run Types

All Run States

Clear Filters

Auto-refresh

Press **shift** + **/** for Shortcuts

deferred

failed

queued

removed

restarting

running

scheduled

skipped

success

up\_for\_reschedule

up\_for\_retry

upstream\_failed

no\_status

DAG

Baranovich\_check\_temperature\_warm\_or\_cold

Task

2024-11-18, 18:41:45 UTC / Baranovich\_Temp\_cold

Clear task

Mark state as...

Filter Tasks

Details

Graph

Gantt

<> Code

Logs

(by attempts)

1

All Levels

All File Sources

Wrap

Download

See More

DESKTOP-KF2TB67.localdomain  
\*\*\* Found local files:  
\*\*\* + /home/airflow/logs/dag\_id=Baranovich\_check\_temperature\_warm\_or\_cold/run\_id=manual\_2024-11-18T18:41:45.997545+00:00/task\_id=Baranovich\_Temp\_cold/attempt=1.log  
[2024-11-18, 18:42:04 UTC] (taskinstance.py:1159) INFO - Dependencies all met for dep\_context=non-requeueable deps ti=<TaskInstance: Baranovich\_check\_temperature\_warm\_or\_cold.Baranovich\_Temp\_cold manual\_2024-11-18T18:41:45.997545+00:00 [queued]>  
[2024-11-18, 18:42:04 UTC] (taskinstance.py:1361) INFO - Starting attempt 1 of 1  
[2024-11-18, 18:42:04 UTC] (taskinstance.py:1362) INFO - Executing <Task(BashOperator): Baranovich\_Temp\_cold> on 2024-11-18 18:41:45.997545+00:00  
[2024-11-18, 18:42:04 UTC] (standard\_task\_runner.py:57) INFO - Started process 731 to run task  
[2024-11-18, 18:42:04 UTC] (standard\_task\_runner.py:84) INFO - Running: ['airflow', 'tasks', 'run', 'Baranovich\_check\_temperature\_warm\_or\_cold', 'Baranovich\_Temp\_cold', 'manual\_2024-\*\*\*\*\*GT\*\*\*\*g:4\*\*\*:45.997545+00:00', '--job-id', '2\*\*\*\*', '--raw', '--subd  
[2024-11-18, 18:42:04 UTC] (standard\_task\_runner.py:85) INFO - Job 2\*\*\*\*: Subtask Baranovich\_Temp\_cold  
[2024-11-18, 18:42:04 UTC] (task\_command.py:416) INFO - Running <TaskInstance: Baranovich\_check\_temperature\_warm\_or\_cold.Baranovich\_Temp\_cold manual\_2024-11-18T18:41:45.997545+00:00 [running]> on host DESKTOP-KF2TB67.localdomain  
[2024-11-18, 18:42:05 UTC] (taskinstance.py:1662) INFO - Exporting env vars: AIRFLOW\_CTX\_DAG\_OWNER='airflow' AIRFLOW\_CTX\_DAG\_ID='Baranovich\_check\_temperature\_warm\_or\_cold' AIRFLOW\_CTX\_TASK\_ID='Baranovich\_Temp\_cold' AIRFLOW\_CTX\_EXECUTION\_DATE='2024-\*\*\*\*\*GT  
[2024-11-18, 18:42:05 UTC] (subprocess.py:75) INFO - Running command: ['/usr/bin/bash', '-c', 'echo "It is cold"']  
[2024-11-18, 18:42:05 UTC] (subprocess.py:186) INFO - Output:  
[2024-11-18, 18:42:05 UTC] (subprocess.py:193) INFO - It is cold  
[2024-11-18, 18:42:05 UTC] (subprocess.py:197) INFO - Command exited with return code 0  
[2024-11-18, 18:42:05 UTC] (taskinstance.py:1400) INFO - Marking task as SUCCESS. dag\_id=Baranovich\_check\_temperature\_warm\_or\_cold, task\_id=Baranovich\_Temp\_cold, execution\_date=2024-\*\*\*\*\*GT\*\*\*\*g4\*\*\*45, start\_date=2024-\*\*\*\*\*GT\*\*\*\*g4204, end\_date=2024-\*\*\*\*\*  
[2024-11-18, 18:42:05 UTC] (local\_task\_job\_runner.py:123) INFO - Task exited with return code 0  
[2024-11-18, 18:42:05 UTC] (taskinstance.py:2778) INFO - 0 downstream tasks scheduled from follow-on schedule check

Baranovich\_get\_temperature

Baranovich\_check\_temperature

Baranovich\_Temp\_cold

Baranovich\_Temp\_warm

Duration

00:00:19

00:00:09

00:00:00

Windows taskbar with various application icons and system clock showing 21:43 on 18.11.2024.