

## Отчет по лабораторной работе № 3

```

46 ---
47 version: '3.8'
48 x-airflow-common:
49   &airflow-common
50   # In order to add custom dependencies or upgrade provider packages you can use your extended image.
51   # Comment the image line, place your Dockerfile in the directory where you placed the docker-compose.yml
52   # and uncomment the "build" line below, Then run 'docker-compose build' to build the images.
53   image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.6.0}
54   # build: .
55   environment:
56     &airflow-common-env
57     AIRFLOW__CORE__EXECUTOR: CeleryExecutor
58     AIRFLOW__DATABASE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow
59     # For backward compatibility, with Airflow <2.3
60     AIRFLOW__CORE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow
61     AIRFLOW__CELERY__RESULT_BACKEND: db+postgresql://airflow:airflow@postgres/airflow
62     AIRFLOW__CELERY__BROKER_URL: redis://:@redis:6379/0
63     AIRFLOW__CORE__FERNET_KEY: ''
64     AIRFLOW__CORE__DAGS_ARE_PAUSED_AT_CREATION: 'true'
65     AIRFLOW__CORE__LOAD_EXAMPLES: 'true'
66     AIRFLOW__API__AUTH_BACKENDS: 'airflow.api.auth.backend.basic_auth,airflow.api.auth.backend.session'
67     # yamllint disable rule:line-length
68     # Use simple http server on scheduler for health checks
69     # See https://airflow.apache.org/docs/apache-airflow/stable/administration-and-deployment/logging-monitoring/check-
70     # yamllint enable rule:line-length
71     AIRFLOW__SCHEDULER__ENABLE_HEALTH_CHECK: 'true'
72     # WARNING: Use _PIP_ADDITIONAL_REQUIREMENTS option ONLY for a quick checks
73     # for other purpose (development, test and especially production usage) build/extend Airflow image.
74     _PIP_ADDITIONAL_REQUIREMENTS: ${_PIP_ADDITIONAL_REQUIREMENTS:-}
75   volumes:
76     - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
77     - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
78     - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
79   user: "${AIRFLOW_UID:-50000}:0"
80   depends_on:
81     &airflow-common-depends-on
82     redis:
83       condition: service_healthy
84     postgres:
85       condition: service_healthy

```

Рис. 1 – конфигурация файла docker-compose.yml (взята дефолтная конфигурация с официального сайта Apache Airflow)

```

/users/iratewarrior/misis/big_data/lab33 > docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
49e93b43056a   apache/airflow:2.6.0               "/usr/bin/dumb-init ..." About an hour ago Up About an hour (healthy)       8080/tcp      lab33-airflow-scheduler-1
a21e994433cb   apache/airflow:2.6.0               "/usr/bin/dumb-init ..." About an hour ago Up About an hour (healthy)       0.0.0.0:8080->8080/tcp lab33-airflow-webserver-1
4e6e4035da7f   apache/airflow:2.6.0               "/usr/bin/dumb-init ..." About an hour ago Up About an hour (healthy)       8080/tcp      lab33-airflow-worker-1
2472ae9868c3   apache/airflow:2.6.0               "/usr/bin/dumb-init ..." About an hour ago Up About an hour (healthy)       8080/tcp      lab33-airflow-triggerer-1
bd849758d43a   apache/airflow:2.6.0               "/bin/bash -c 'funct..." About an hour ago Exited (0) About an hour ago                                lab33-airflow-init-1
dcacedb39324   postgres:13                         "docker-entrypoint.s..." About an hour ago Up About an hour (healthy)       0.0.0.0:5433->5432/tcp lab33-postgres-1
d74b90cb868e   redis:latest                       "docker-entrypoint.s..." About an hour ago Up About an hour (healthy)       6379/tcp      lab33-redis-1

```

Рис. 2 – после поднятия docker compose список работающих контейнеров

```
docker compose up

lab33-airflow-webserver-1 | 172.24.0.1 - - [05/May/2023:22:28:14 +0000] "GET /static/dist/tiLog.60be340a0fd851aa6e3f.js HTTP/1.1" 304 0 "http://localhost:8080/log?d
k_id=visualize_data&execution_date=2023-05-05T22%3A27%3A52.813200%2B00%3A00&map_index=-1" Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML
.0 Safari/537.36"
lab33-airflow-webserver-1 | 172.24.0.1 - - [05/May/2023:22:28:14 +0000] "GET /object/next_run_datasets/data_transfer_dag HTTP/1.1" 200 2 "http://localhost:8080/log?d
sk_id=visualize_data&execution_date=2023-05-05T22%3A27%3A52.813200%2B00%3A00&map_index=-1" Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML
.0 Safari/537.36"
lab33-airflow-webserver-1 | 172.24.0.1 - - [05/May/2023:22:28:14 +0000] "GET /get_logs_with_metadata?dag_id=data_transfer_dag&task_id=visualize_data&map_index=-1&ex
3A27%3A52.813200%2B00%3A00&try_number=1&metadata=null HTTP/1.1" 200 2752 "http://localhost:8080/log?dag_id=data_transfer_dag&task_id=visualize_data&execution_date=20
0%2B00%3A00&map_index=-1" Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:28:15 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:28:46 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:29:16 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:28:34.508+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:29:46 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:30:16 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:29:34.681+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:29:46 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:30:16 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:30:34.835+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:30:46 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:31:16 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:31:35.000+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:31:46 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:32:17 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-scheduler-1 | [2023-05-05T22:32:25.920+0000] {scheduler_job_runner.py:1576} INFO - Resetting orphaned tasks for active dag runs
lab33-airflow-triggerer-1 | [2023-05-05T22:32:35.168+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:32:47 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:33:17 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:33:35.336+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:33:47 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-redis-1 | 1:M 05 May 2023 22:34:17.290 * 100 changes in 300 seconds. Saving...
lab33-redis-1 | 1:M 05 May 2023 22:34:17.295 * Background saving started by pid 2800
lab33-redis-1 | 2800:C 05 May 2023 22:34:17.301 * DB saved on disk
lab33-redis-1 | 2800:C 05 May 2023 22:34:17.304 * Fork CoW for RDB: current 0 MB, peak 0 MB, average 0 MB
lab33-redis-1 | 1:M 05 May 2023 22:34:17.397 * Background saving terminated with success
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:34:17 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:34:35.491+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:34:47 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:35:18 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:35:35.671+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:35:48 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:36:18 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
lab33-airflow-triggerer-1 | [2023-05-05T22:36:35.850+0000] {triggerer_job_runner.py:485} INFO - 0 triggers currently running
lab33-airflow-webserver-1 | 127.0.0.1 - - [05/May/2023:22:36:48 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
```

Рис. 3 – работа Airflow в консоли

```
docker-compose.yaml  new_dag.py  data24.csv

1  from airflow.models import DAG
2  from airflow.operators.python_operator import PythonOperator
3  from datetime import datetime
4  import pandas as pd
5  import psycopg2
6
7  # Функция для чтения данных из CSV файла
8  def read_csv_data():
9      df = pd.read_csv('/opt/airflow/dags/data24.csv')
10     return df
11
12  # Функция для записи данных в базу данных Postgres
13  def write_to_postgres():
14      conn = psycopg2.connect(
15          host='172.24.0.3',
16          port='5432',
17          database='airflow',
18          user='airflow',
19          password='airflow'
20      )
21      cursor = conn.cursor()
22
23      # Прочитать данные из CSV файла
24      df = read_csv_data()
25
26      for i in df.values:
27          cursor.execute("INSERT INTO public.lab34 (id, name, surname, age, height, weight, run_100m) VALUES (%s, %s, %s, %s, %s, %s, %s);",
28              [i[0], i[1], i[2], i[3], i[4], i[5], i[6]])
29
30      conn.commit()
31      cursor.close()
32      conn.close()
```

Рис. 4.1 – конфигурация DAG для Apache Airflow

```
docker-compose.yaml  new_dag.py  data24.csv

33
34 # Функция для визуализации данных в консоли
35 @page
36 def visualize_data():
37     conn = psycopg2.connect(
38         host='172.24.0.3',
39         port='5432',
40         database='airflow',
41         user='airflow',
42         password='airflow'
43     )
44     # Выберем топ-3 бегунов по результатам бега 100 метров
45     query = """
46     SELECT name, surname, run_100m
47     FROM public.lab34
48     ORDER BY run_100m
49     LIMIT 3;
50     """
51     # Заполнение данных в DataFrame
52     df = pd.read_sql(query, conn)
53
54     conn.close()
55
56     # Выводим результаты в консоль
57     print("Top 3 runners by 100m run:")
58     print(df)
59
60 # Создание DAG
61 dag = DAG(
62     'data_transfer_dag',
63     description='Перенос данных из CSV в базу данных Postgres',
64     schedule_interval=None,
65     start_date=datetime(2023, 5, 5),
66     catchup=False
67 )
```

Рис. 4.2 – конфигурация DAG для Apache Airflow

```
docker-compose.yaml  new_dag.py  data24.csv
58
59 # Создание DAG
60 dag = DAG(
61     'data_transfer_dag',
62     description='Перенос данных из CSV в базу данных Postgres',
63     schedule_interval=None,
64     start_date=datetime(2023, 5, 5),
65     catchup=False
66 )
67
68 # Оператор для чтения данных из CSV файла
69 read_csv_operator = PythonOperator(
70     task_id='read_csv_data',
71     python_callable=read_csv_data,
72     dag=dag
73 )
74
75 # Оператор для записи данных в базу данных Postgres
76 write_to_postgres_operator = PythonOperator(
77     task_id='write_to_postgres',
78     python_callable=write_to_postgres,
79     dag=dag
80 )
81
82 visualize_data = PythonOperator(
83     task_id='visualize_data',
84     python_callable=visualize_data,
85     dag=dag
86 )
87
88
89
90 # Установка порядка выполнения задач
91 read_csv_operator >> write_to_postgres_operator >> visualize_data
92
```

Рис. 4.3 – конфигурация DAG для Apache Airflow

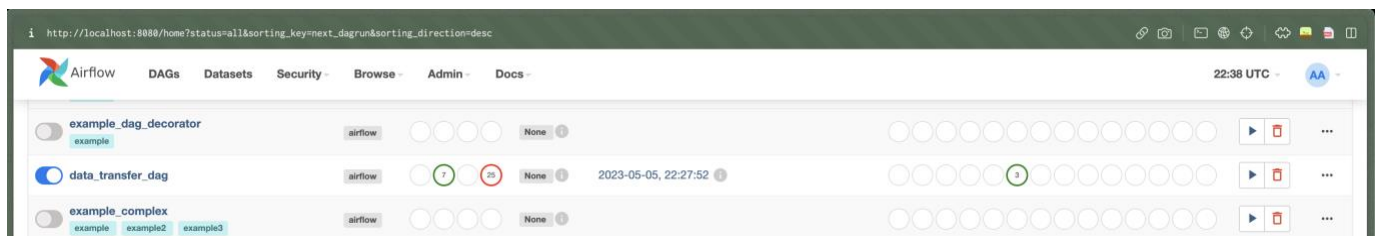


Рис. 5 – отображение DAG в UI Apache Airflow

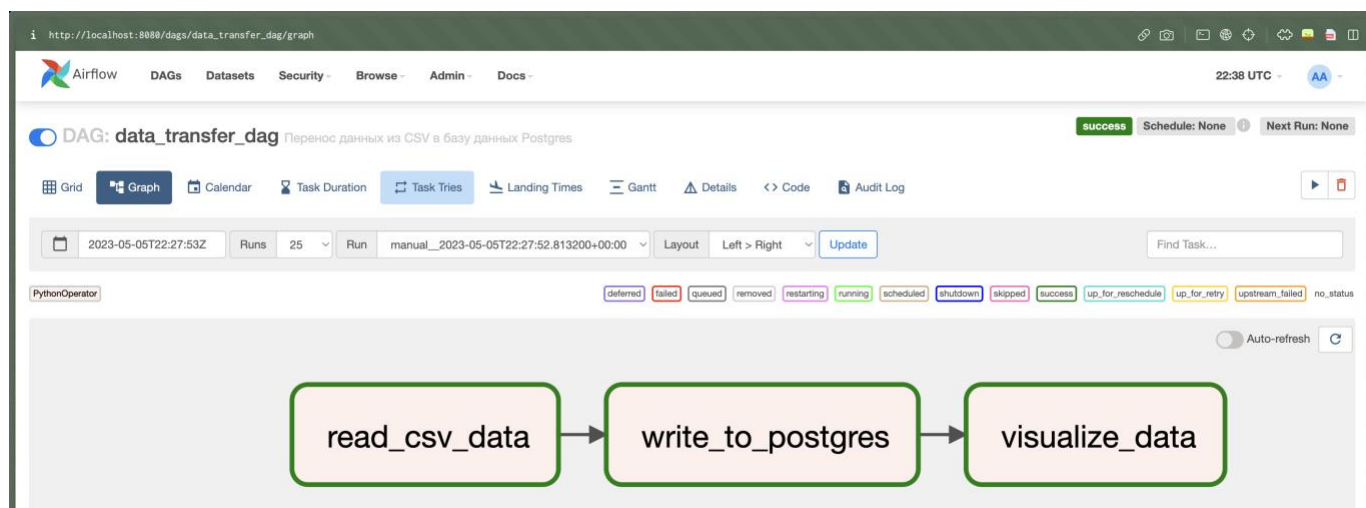


Рис. 6 – отображение графа DAG в UI Apache Airflow

```
docker-compose.yaml  new_dag.py  data24.csv x
1 id, name, surname, age, height, weight, run_100m
2 1, alex, doe, 23, 178, 60, 12
3 2, travis, scott, 25, 180, 74, 10
4 3, kendrick, lamar, 33, 170, 70, 13
5 4, donald, glover, 35, 175, 65, 15
6 5, lakeith, stanfield, 30, 185, 78, 11
```

Рис. 7 – содержание CSV файла

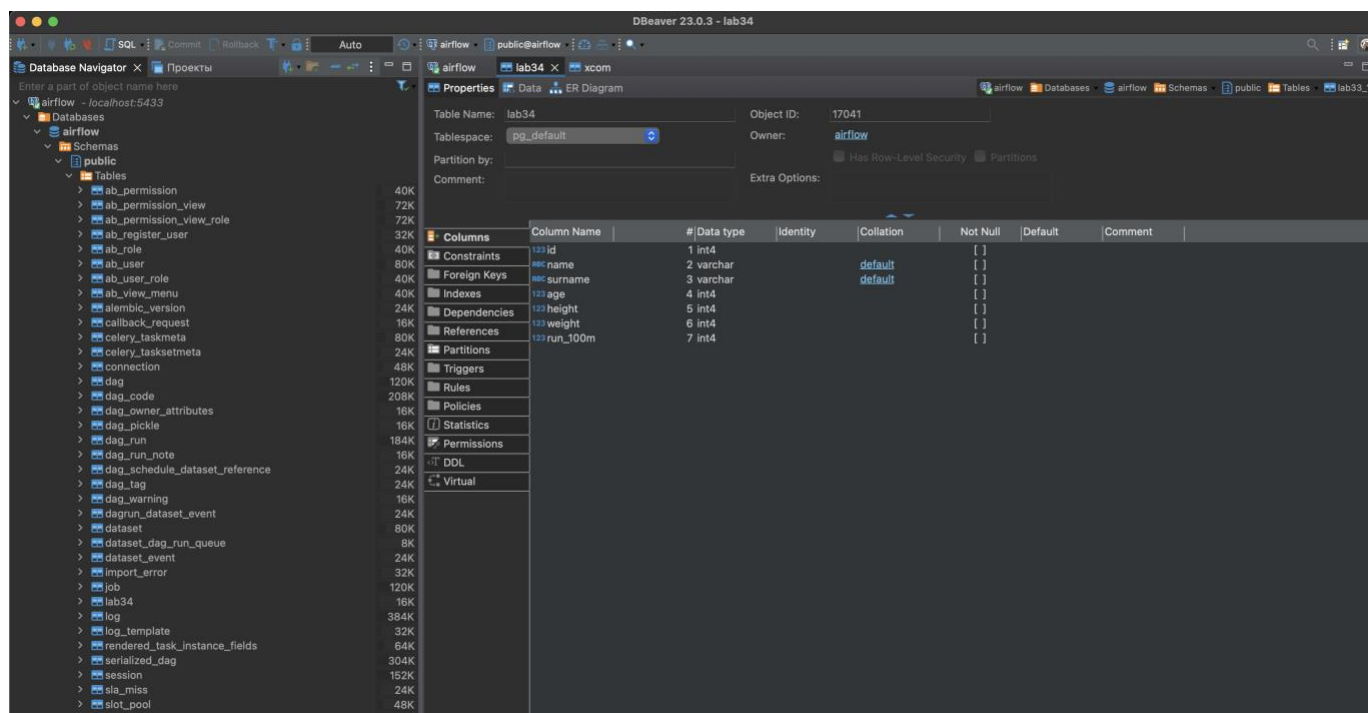


Рис. 8 – создание таблицы в БД Postgres в ПО DBeaver

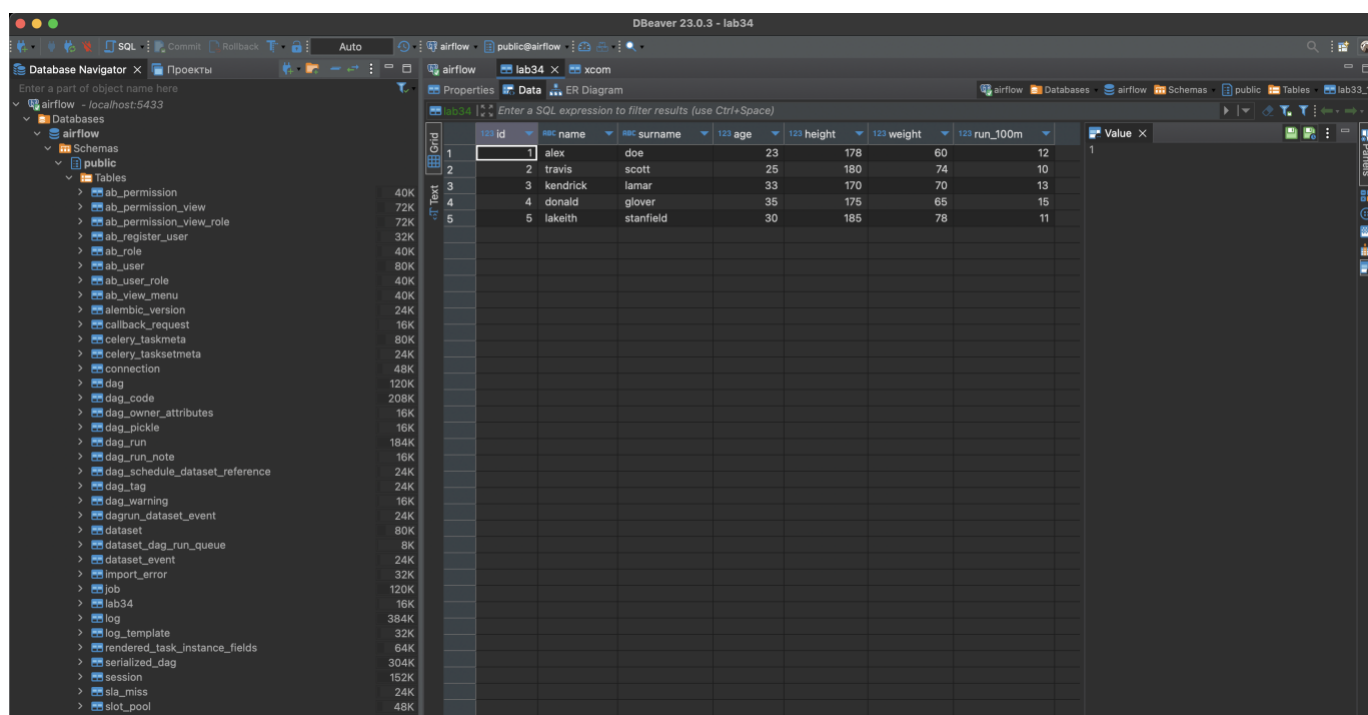


Рис. 9 – таблицы в БД Postgres заполнена данными из CSV файла после отработки DAG

http://localhost:8080/log?dag\_id=data\_transfer\_dag&task\_id=visualize\_data&execution\_date=2023-05-05T22:27:52.813200+00:00&map\_index=-1

Airflow DAGs Datasets Security Browse Admin Docs 22:39 UTC AA

DAG: data\_transfer\_dag Перенос данных из CSV в базу данных Postgres Schedule: False

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

Task Instance: visualize\_data at 2023-05-05, 22:27:52

Task Instance Details <> Rendered Template Log XCom

Log by attempts

1

Jump To End Toggle Wrap Download

```
*** Found local files:
*** * /opt/airflow/logs/dag_id=data_transfer_dag/run_id>manual__2023-05-05T22:27:52.813200+00:00/task_id=visualize_data/attempt=1.log
[2023-05-05, 22:28:01 UTC] {taskinstance.py:1125} INFO - Dependencies all met for dep_context=non-requeueable deps ti=<TaskInstance: data_transfer_dag.visualize_data manual__2023-05-05T22:27:52.813200+00:00 [queued]>
[2023-05-05, 22:28:01 UTC] {taskinstance.py:1125} INFO - Dependencies all met for dep_context=requeueable deps ti=<TaskInstance: data_transfer_dag.visualize_data manual__2023-05-05T22:27:52.813200+00:00 [queued]>
[2023-05-05, 22:28:01 UTC] {taskinstance.py:1331} INFO - Starting attempt 1 of 1
[2023-05-05, 22:28:01 UTC] {taskinstance.py:1350} INFO - Executing <Task(PythonOperator): visualize_data> on 2023-05-05 22:27:52.813200+00:00
[2023-05-05, 22:28:01 UTC] {standard_task_runner.py:57} INFO - Started process 1434 to run task
[2023-05-05, 22:28:01 UTC] {standard_task_runner.py:84} INFO - Running: ['***', 'tasks', 'run', 'data_transfer_dag', 'visualize_data', 'manual__2023-05-05T22:27:52.813200+00:00', '--job-id', '83', '--raw', '1434']
[2023-05-05, 22:28:01 UTC] {standard_task_runner.py:85} INFO - Job 83: Subtask visualize_data
[2023-05-05, 22:28:01 UTC] {task_command.py:410} INFO - Running <TaskInstance: data_transfer_dag.visualize_data manual__2023-05-05T22:27:52.813200+00:00 [running]> on host 4e6e4035da7f
[2023-05-05, 22:28:01 UTC] {taskinstance.py:1570} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='***' AIRFLOW_CTX_DAG_ID='data_transfer_dag' AIRFLOW_CTX_TASK_ID='visualize_data' AIRFLOW_CTX_EXECUTION_DATE='2023-05-05T22:27:52.813200+00:00'
[2023-05-05, 22:28:01 UTC] {logging_mixin.py:149} INFO - Top 3 runners by 100m run:
[2023-05-05, 22:28:01 UTC] {logging_mixin.py:149} INFO -      name      surname run_100m
0   travis    scott      10
1 lakeith    stanfield  11
2   alex      doe        12
[2023-05-05, 22:28:01 UTC] {python.py:183} INFO - Done. Returned value was: None
[2023-05-05, 22:28:01 UTC] {taskinstance.py:1373} INFO - Marking task as SUCCESS. dag_id=data_transfer_dag, task_id=visualize_data, execution_date=20230505T222752, start_date=20230505T222801, end_date=20230505T222801
[2023-05-05, 22:28:01 UTC] {local_task_job_runner.py:232} INFO - Task exited with return code 0
[2023-05-05, 22:28:01 UTC] {taskinstance.py:2674} INFO - 0 downstream tasks scheduled from follow-on schedule check
```

Рис. 10 – логи DAG