**Задание 5.**

**Создание Docker-контейнера с PostgreSQL и ClickHouse**

**Цель: Научиться развертывать базы данных PostgreSQL и ClickHouse в Docker с использованием Docker Compose, создавать таблицы и данные в этих базах данных, а затем использовать PySpark для чтения данных из обеих баз данных и работы с ними в рамках одного DataFrame.**

Описание задания:

* Создайте директорию для проекта и необходимые файлы.
* Создайте файлы create\_tables.sql, которые будут содержать SQL-запросы для создания таблиц и вставки данных в обе базы данных.
* В docker-compose.yml опишите конфигурацию для развертывания контейнеров с PostgreSQL и ClickHouse.
* Выполните команду для запуска контейнеров с PostgreSQL и ClickHouse.
* Создайте файл pyspark\_script.py в вашем локальном окружении (вне Docker), который будет подключаться к обеим базам данных и читать данные.
* Вы должны увидеть вывод данных из обеих баз данных (PostgreSQL и ClickHouse), считанных и обработанных PySpark.

##################### **docker-compose.yml** #####################

version: '3.8'

services:

postgres:

image: postgres:14

container\_name: postgres\_container

environment:

POSTGRES\_USER: postgres

POSTGRES\_PASSWORD: password

POSTGRES\_DB: test\_db

ports:

- "5433:5432"

volumes:

- ./create\_tables\_postgres.sql:/docker-entrypoint-initdb.d/create\_tables.sql

clickhouse:

image: yandex/clickhouse-server:latest

container\_name: clickhouse\_container

ports:

- "8123:8123"

- "9000:9000"

volumes:

- ./create\_tables\_clickhouse.sql:/docker-entrypoint-initdb.d/create\_tables.sql

##########################################

##################### **create\_tables\_postgres.sql** #####################

create table if not exists users (

id serial primary key,

name varchar(100),

age int

);

insert into users (name, age) values

('Ivan', 29),

('Dmitry', 27),

('Maxim', 19);

##################### **create\_tables\_clickhouse.sql** #####################

CREATE TABLE users (

id UInt32,

name String,

age UInt8

) ENGINE = MergeTree

ORDER BY id;

insert into users (id, name, age) values

(1, 'Denis', 16);

(2, 'Kate', 13),

(3, 'Ludmila', 52);

##########################################

**Запуск контейнеров**

docker-compose up -d

docker ps

##########################################

pip install pyspark

##################### **pyspark\_script.py**#####################

from pyspark.sql import SparkSession

# Создаем единую SparkSession

spark = SparkSession.builder \

    .appName("ClickHouse and PostgreSQL") \

    .config("spark.jars", "clickhouse-jdbc-0.7.1.jar,postgresql-42.2.23.jar") \

    .getOrCreate()

# Подключение к ClickHouse

CLICKHOUSE\_URL = "jdbc:clickhouse://localhost:8123/default"

CLICKHOUSE\_USER = "admin"

CLICKHOUSE\_PASSWORD = "password"

clickhouse\_df = spark.read \

    .format("jdbc") \

    .option("url", CLICKHOUSE\_URL) \

    .option("dbtable", "users") \

    .option("user", CLICKHOUSE\_USER) \

    .option("password", CLICKHOUSE\_PASSWORD) \

    .option("driver", "com.clickhouse.jdbc.ClickHouseDriver") \

    .load()

# Подключение к PostgreSQL

POSTGRES\_JDBC\_URL = "jdbc:postgresql://localhost:5433/test\_db"

POSTGRES\_USER = "postgres"

POSTGRES\_PASSWORD = "password"

postgres\_df = spark.read \

    .format("jdbc") \

    .option("url", POSTGRES\_JDBC\_URL) \

    .option("dbtable", "users") \

    .option("user", POSTGRES\_USER) \

    .option("password", POSTGRES\_PASSWORD) \

    .option("driver", "org.postgresql.Driver") \

    .load()

# Пример объединения данных

result\_df = clickhouse\_df.union(postgres\_df)

# Печать объединенных данных

result\_df.show()

# Остановка SparkSession

spark.stop()

##########################################

python pyspark\_script.py

docker-compose down

docker volume prune -af