*Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана  (национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

ДИСЦИПЛИНА Языки программирования для работы с большими данными

**Отчет**

**по лабораторной работе № 10**

**Вариант 5**

Студент гр. ИУ6-21М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_**Дейкин И. И.\_\_\_

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

*Условие*

Исследовать предложенный набор данных при помощи языка Scala и фреймворка Apache Spark. Для изучения был выбран набор Russian\_demography с ресурса Kaggle. Вид набора данных приведен на рисунке 1.

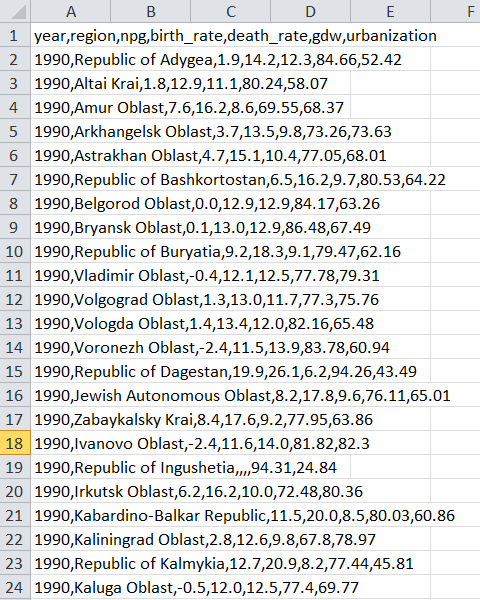


Рисунок 1 – Набор данных

*Решение*

Код главного объекта Scala приведен ниже:

|  |
| --- |
| import org.apache.spark.sql.SparkSession  import org.apache.spark.{SparkConf, SparkContext}  object Smth {  def main(args: Array[String]): Unit = {  //val logFile = "C:/Spark/readme.md"  val conf = new SparkConf().setAppName("CounterDemo").setMaster("local[\*]")  val sc = new SparkContext(conf)  val spark = SparkSession.builder().appName(name = "TestApp").getOrCreate()  val dataFile = spark.read.format("com.databricks.spark.csv")  .option("header", true).load("C:\\Users\\ideyk\\IdeaProjects\\Lab10\\russian\_demography.csv")  dataFile.createOrReplaceTempView("Demography")  // Выводим самые плодородные годы/регионы, показываются верхние 20  spark.sql("SELECT year, region FROM Demography ORDER BY birth\_rate DESC").show()  // Статистика рождаемости/смертности по Москве  spark.sql("SELECT year, birth\_rate, death\_rate FROM Demography WHERE region='Moscow'").show()  // Средняя смертность за 1997 год  spark.sql("SELECT AVG(death\_rate) FROM Demography WHERE year=1997").show()  spark.stop()  }  } |

*Результаты работы*

Результат работы первого запроса приведен на рисунке 2. Здесь выведены 20 самых плодородных пар регион+год за весь охват данных.



Рисунок 2 – Самые плодородные регионы и годы

Второй запрос (рисунок 3) выводит статистику рождаемости/смертности в Москве по годам.

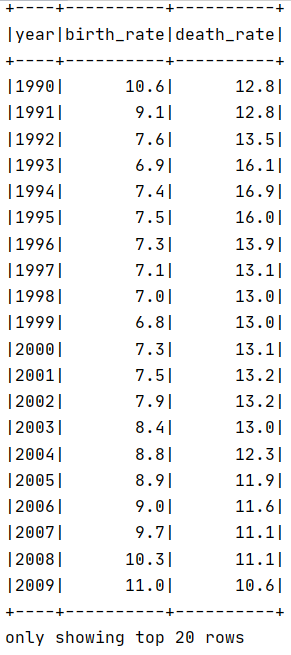


Рисунок 3 – Статистика рождаемости/смертности по Москве

В третьем запросе определяется средняя смертность по стране за 1997 год (рисунок 4).

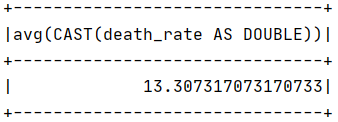


Рисунок 4 – Средняя смертность по РФ в 1997 году

*Вывод*

В ходе выполнения данной работы было настроено взаимодействие Apache Spark и языка Scala на базе среды IntelliJ IDEA. Полученные результаты выполнения SQL-запросов свидетельствуют о верной настройке системы.