|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«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 (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**Отчет**

**по лабораторной работе № 10**

**Название:** Обработка больших данных

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-21М |  |  | Д.Ю. Ермошин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Задание**

Скачать датасет с Kernel и выполнить несколько запросов.

**Выполнение работы**

|  |
| --- |
| package lab10  import org.apache.spark.sql.SparkSession  import org.apache.spark.{SparkConf, SparkContext}  object Lab10 {  def main(args: Array[String]): Unit = {  val sparkConfiguration = new SparkConf().setAppName("Lab10").setMaster("local[\*]")  val sparkContext = new SparkContext(sparkConfiguration);  val sparkSession = SparkSession.builder.appName("Test app").getOrCreate()  val datafile = sparkSession.read  .format("com.databricks.spark.csv")  .option("header", true)  .load("./HRR\_Scorecard.csv")  datafile.createOrReplaceTempView("HRR\_Scorecard")  // Максимальное количество необходимых мест для больных за 6 месяцев  sparkSession.sql("select `HRR`, MAX(`Hospital Beds Needed, Six Months`) as `Beds Needed` FROM HRR\_Scorecard group by `HRR` order by `Beds Needed` desc").show()  // Максимальное количество прогнозируемых зараженных COVID-19 в США старше 18 лет  sparkSession.sql("select MAX(`Projected Infected Individuals`) as max FROM HRR\_Scorecard group by `HRR`").show()  // Среднее количество пенсионеров в больницах США  sparkSession.sql("select `HRR`, AVG(`Population 65+`) as `Number of Pensioners` FROM HRR\_Scorecard group by `HRR`").show()  // Количество доступных мест в ICU в порядке убывания  sparkSession.sql("select `HRR`, `Available ICU Beds` FROM HRR\_Scorecard WHERE order by `Available ICU Beds` desc").show()  // Наибольшее количество мест в больницах Hospital (больше 100)  sparkSession.sql("select `HRR`, `Total ICU Beds` FROM HRR\_Scorecard WHERE `Total Hospital Beds` > 100").show()  sparkSession.stop()  }  } |

**Заключение**

В данной лабораторной работе мы сделали несколько запросов к Spark (c помощью SQL-API) посредством языка Scala. Тем самым мы изучили принцип обработки больших данных.