厦门大学林子雨，郑海山，赖永炫 编著

《Spark编程基础（Python）》

教材配套

机房上机实验指南

实验4 Spark SQL编程初级实践

（版本号：2020年4月版本）

（答案）



主讲教师：林子雨

厦门大学数据库实验室

二零二零年四月

目录

[一、实验目的 1](#_Toc494985819)

[二、实验平台 1](#_Toc494985820)

[三、实验内容和要求 1](#_Toc494985821)

[1．Spark SQL基本操作 1](#_Toc494985822)

[2．编程实现将RDD转换为DataFrame 2](#_Toc494985823)

[3. 编程实现利用DataFrame读写MySQL的数据 4](#_Toc494985824)

[四、实验报告 5](#_Toc494985825)

[**附录1:任课教师介绍** 6](#_Toc494985826)

[**附录2：课程教材介绍** 6](#_Toc494985827)

[**附录3：高校大数据课程公共服务平台介绍** 7](#_Toc494985828)

厦门大学林子雨,赖永炫,陶继平 编著《Spark编程基础》

第5章 Spark SQL

教材配套机房上机实验指南

实验4 Spark SQL编程初级实践

（答案）

**主讲教师：林子雨**

E-mail: [ziyulin@xmu.edu.cn](mailto:ziyulin@xmu.edu.cn) 个人主页：<http://dblab.xmu.edu.cn/post/linziyu>

# 一、实验目的

（1）通过实验掌握Spark SQL的基本编程方法；

（2）熟悉RDD到DataFrame的转化方法；

（3）熟悉利用Spark SQL管理来自不同数据源的数据。

# 二、实验平台

操作系统： Ubuntu16.04

Spark版本：2.4.0

数据库：MySQL

Python版本：3.4.3

# 三、实验内容和要求

## 1．Spark SQL基本操作

将下列json数据复制到你的ubuntu系统/usr/local/spark下，并保存命名为employee.json。

{ "id":1 ,"name":" Ella","age":36 }

{ "id":2,"name":"Bob","age":29 }

{ "id":3 ,"name":"Jack","age":29 }

{ "id":4 ,"name":"Jim","age":28 }

{ "id":5 ,"name":"Damon" }

{ "id":5 ,"name":"Damon" }

首先为employee.json创建DataFrame，并写出Python语句完成下列操作：

创建DataFrame

答案：

>>> spark=SparkSession.builder().getOrCreate()

>>> df = spark.read.json("file:///usr/local/spark/employee.json")

1. 查询DataFrame的所有数据

答案：>>> df.show()

1. 查询所有数据，并去除重复的数据

答案：>>> df.distinct().show()

1. 查询所有数据，打印时去除id字段

答案：>>> df.drop("id").show()

1. 筛选age>20的记录

答案：>>> df.filter(df.age > 30 ).show()

1. 将数据按name分组

答案：>>> df.groupBy("name").count().show()

1. 将数据按name升序排列

答案：>>> df.sort(df.name.asc()).show()

(7)取出前3行数据

答案：>>> df.take(3) 或python> df.head(3)

(8)查询所有记录的name列，并为其取别名为username

答案：>>> df.select(df.name.alias("username")).show()

(9)查询年龄age的平均值

答案：>>> df.agg({"age": "mean"}).show()

(10)查询年龄age的最大值

答案：>>> df.agg({"age": "max"}).show()

## 2．编程实现将RDD转换为DataFrame

源文件内容如下（包含id,name,age），将数据复制保存到ubuntu系统/usr/local/spark下，命名为employee.txt，实现从RDD转换得到DataFrame，并按id:1,name:Ella,age:36的格式打印出DataFrame的所有数据，注意最后一列数据千万不要留空格，会出现越界的异常。请写出程序代码。（任选一种方法即可）

1,Ella,36

2,Bob,29

3,Jack,29

答案：

假设当前目录为/usr/local/spark/mycode/rddtodf，在当前目录下新建一个目录mkdir -p src/main/python，然后在目录/usr/local/spark/mycode/rddtodf/src/main/python下新建一个rddtodf.py，复制下面代码；（下列两种方式任选其一）

方法一：利用反射来推断包含特定类型对象的RDD的schema，适用对已知数据结构的RDD转换；

from pyspark.conf import SparkConf

from pyspark.sql.session import SparkSession

from pyspark import SparkContext

from pyspark.sql.types import Row

if \_\_name\_\_ == "\_\_main\_\_":

sc = SparkContext("local","Simple App")

peopleRDD = sc.textFile("file:///usr/local/spark/employee.txt")

rowRDD = peopleRDD.map(lambda line : line.split(",")).map(lambda attributes : Row(int(attributes[0]),attributes[1],int(attributes[2]))).toDF()

rowRDD.createOrReplaceTempView("employee")

personsDF = spark.sql("select \* from employee")

personsDF.rdd.map(lambda t : "id:"+str(t[0])+","+"Name:"+t[1]+","+"age:"+str(t[2])).foreach(print)

方法二：使用编程接口，构造一个schema并将其应用在已知的RDD上。

from pyspark.sql.types import Row

from pyspark.sql.types import StructType

from pyspark.sql.types import StructField

from pyspark.sql.types import StringType

from pyspark.conf import SparkConf

from pyspark import SparkContext

from pyspark.sql.session import SparkSession

if \_\_name\_\_ == "\_\_main\_\_":

sc = SparkContext("local","Simple App")

peopleRDD = sc.textFile("file:///usr/local/spark/employee.txt")

schemaString = "id name age"

fields = list(map( lambda fieldName : StructField(fieldName, StringType(), nullable = True), schemaString.split(" ")))

schema = StructType(fields)

rowRDD = peopleRDD.map(lambda line : line.split(",")).map(lambda attributes : Row(int(attributes[0]),attributes[1],int(attributes[2])))

employeeDF = spark.createDataFrame(rowRDD, schema)

employeeDF.createOrReplaceTempView("employee")

results = spark.sql("SELECT \* FROM employee")

results.rdd.map(lambda t : "id:"+str(t[0])+","+"Name:"+t[1]+","+"age:"+str(t[2])).foreach(print)

|  |
| --- |
| python3 ./ rddtodf.py |

在终端即可看到输出结果。

|  |
| --- |
| id:1,Name:Ella,age:36  id:2,Name:Bob,age:29  id:3,Name:Jack,age:29 |

## 3. 编程实现利用DataFrame读写MySQL的数据

1. 在MySQL数据库中新建数据库sparktest，再建表employee，包含下列两行数据；

**表1 employee表原有数据**

|  |  |  |  |
| --- | --- | --- | --- |
| id | name | gender | age |
| 1 | Alice | F | 22 |
| 2 | John | M | 25 |

答案：

mysql> create database sparktest;

mysql> use sparktest;

mysql> create table employee (id int(4), name char(20), gender char(4), age int(4));

mysql> insert into employee values(1,'Alice','F',22);

mysql> insert into employee values(2,'John','M',25);

1. 配置Spark通过JDBC连接数据库MySQL，编程实现利用DataFrame插入下列数据到MySQL，最后打印出age的最大值和age的总和。

**表2 employee表新增数据**

|  |  |  |  |
| --- | --- | --- | --- |
| id | name | gender | age |
| 3 | Mary | F | 26 |
| 4 | Tom | M | 23 |

答案：假设当前目录为/usr/local/spark/mycode/testmysql，在当前目录下新建一个目录mkdir -p src/main/python，然后在目录/usr/local/spark/mycode/testmysql/src/main/python下新建一个testmysql.py，复制下面代码；

from pyspark import SparkContext

from pyspark.sql import SQLContext

from pyspark.sql.types import Row

from pyspark.sql.types import StructType

from pyspark.sql.types import StructField

from pyspark.sql.types import StringType

from pyspark.sql.types import IntegerType

if \_\_name\_\_ == "\_\_main\_\_":

sc = SparkContext( 'local', 'test')

spark=SQLContext(sc)

jdbcDF=spark.read.format("jdbc").option("url","jdbc:mysql://localhost:3306/sparktest").option("driver","com.mysql.jdbc.Driver").option("dbtable","employee").option("user", "root").option("password", "123").load()

jdbcDF.filter(jdbcDF.age>20).collect()//检测是否连接成功

studentRDD = sc.parallelize(["3 Mary F 26","4 Tom M 23"]).map(lambda line : line.split(" "))

schema = StructType([StructField("id",IntegerType(),True),StructField("name", StringType(), True),StructField("gender", StringType(), True),StructField("age",IntegerType(), True)])

rowRDD = studentRDD.map(lambda p : Row(int(p[0]),p[1].strip(), p[2].strip(),int(p[3])))

employeeDF = spark.createDataFrame(rowRDD, schema)

prop = {}

prop['user'] = 'root'

prop['password'] = '123'

prop['driver'] = "com.mysql.jdbc.Driver"

employeeDF.write.jdbc("jdbc:mysql://localhost:3306/sparktest",'employee','append', prop)

jdbcDF.collect()

jdbcDF.agg({"age": "max"}).show()

jdbcDF.agg({"age": "sum"}).show()

然后我们，执行以下指令

|  |
| --- |
| python3 ./ rddtodf.py |

在终端下，我们就可以看到结果了。

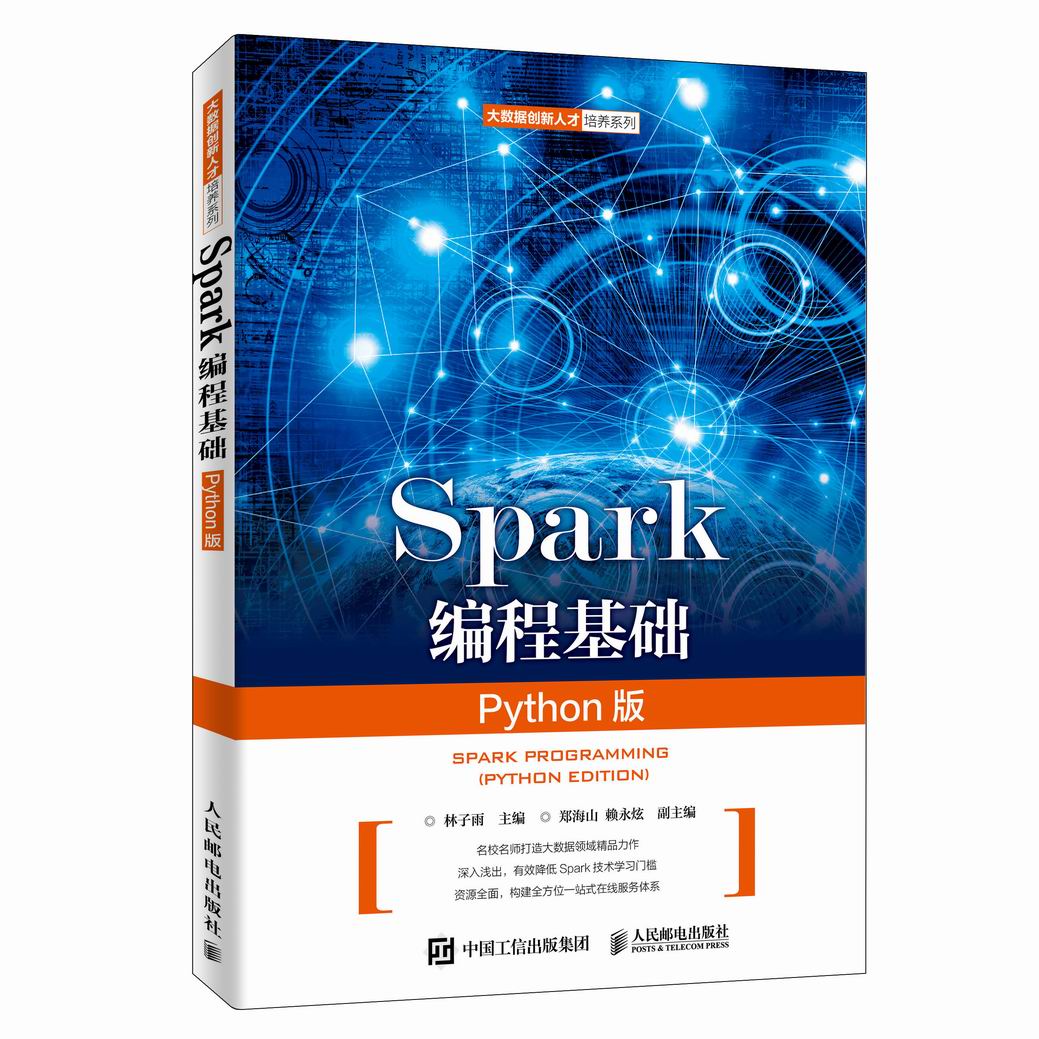
# 四、实验报告

|  |  |  |
| --- | --- | --- |
| 《Spark编程基础》实验报告 | | |
| 题目： | 姓名： | 日期： |
| 实验环境： | | |
| 实验内容与完成情况： | | |
| 出现的问题： | | |
| 解决方案（列出遇到的问题和解决办法，列出没有解决的问题）： | | |

**附录1:任课教师介绍**

|  |  |
| --- | --- |
| [http://www.cs.xmu.edu.cn/database/linziyu/images/linziyu2016.jpg](http://dblab.xmu.edu.cn/post/linziyu/) | 林子雨（1978－），男，博士，国内高校知名大数据教师，厦门大学计算机科学系副教授，厦门大学云计算与大数据研究中心创始成员，厦门大学数据库实验室负责人，中国计算机学会数据库专委会委员，中国计算机学会信息系统专委会委员，中国高校首个“数字教师”提出者和建设者。2013年开始在厦门大学开设大数据课程，建设了国内高校首个大数据课程公共服务平台，平台累计网络访问量超过1000万次，成为全国高校大数据教学知名品牌，并荣获“2018年福建省教学成果二等奖”，主持的课程《大数据技术原理与应用》获评“2018年国家精品在线开放课程”。  E-mail: ziyulin@xmu.edu.cn  个人主页：http://dblab.xmu.edu.cn/post/linziyu  数据库实验室网站：http://dblab.xmu.edu.cn |

**附录2：课程教材介绍**



林子雨、郑海山、赖永炫编著《Spark编程基础（Python版）》

人民邮电出版社 ISBN:978-7-115-52439-3 定价：49.80元

厦门大学林子雨、郑海山、赖永炫老师编著《Spark编程基础（Python版）》，以Python作为开发Spark应用程序的编程语言，系统介绍了Spark编程的基础知识。全书共8章，内容包括大数据技术概述、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Structured Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作，以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源，包括讲义PPT、习题、源代码、软件、数据集、授课视频、上机实验指南等。

本书可以作为高等院校计算机、软件工程、数据科学与大数据技术等专业的进阶级大数据课程教材，用于指导Spark编程实践，也可供相关技术人员参考。

欢迎访问《Spark编程基础(Python版)》教材官方网站：http://dblab.xmu.edu.cn/post/spark-python/



扫一扫访问教材官网

**附录3：高校大数据课程公共服务平台介绍**



高校大数据课程公共服务平台，由中国高校首个“数字教师”的提出者和建设者——林子雨老师发起，由厦门大学数据库实验室全力打造，由厦门大学云计算与大数据研究中心、海峡云计算与大数据应用研究中心携手共建。这是国内第一个服务于高校大数据课程建设的公共服务平台，旨在促进国内高校大数据课程体系建设，提高大数据课程教学水平，降低大数据课程学习门槛，提升学生课程学习效果。平台服务对象涵盖高校、教师和学生。平台为高校开设大数据课程提供全流程辅助，为教师开展教学工作提供一站式服务，为学生学习大数据课程提供全方位辅导。平台重点打造“11个1工程”，即1本教材（含官网）、1个教师服务站、1个学生服务站、1个公益项目、1堂巡讲公开课、1个示范班级、1门在线课程、1个交流群（QQ群、微信群）、1个保障团队、1个培训基地和1个实验平台。目前平台每年访问量已经超过200万次，累计访问量超过1000万次，成为国内高校大数据教学知名品牌。

平台主页：http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/



扫一扫访问平台主页