大规模分布式系统第六次作业报告

首先要将表放入 HDFS 中，故先为表创建了 table1.csv 和 table2.csv 文件。

在 Hadoop 下创建输入文件夹

```
hadoop@ubuntu:/usr/local/Hadoop/bin$ ./hadoop fs -ls /input
```

将本地两个文件全部放入 input 文件夹下，然后观察 input 文件夹下的文件，发现已经成功执行

```
hadoop@ubuntu:/usr/local/Hadoop/bin$ ./hadoop fs -ls /input
Found 2 items
-rw-r--r--   1 hadoop supergroup        146 2019-06-10 08:22 /input/table1.csv
-rw-r--r--   1 hadoop supergroup        137 2019-06-10 08:22 /input/table2.csv
```

现在打开 pyspark，在 jupternotebook 的运行环境下进行运行测试是否能够成功读入 HDFS 的文件。

```
In [1]: data = sc.textFile("hdfs://localhost:9000/input/table1.csv")

In [2]: data.collect()

Out[2]: ['ʌ��,���ijr�,��ʌ�r�',
         '101,99,25',
         '102,56,68',
         '103,74,88',
         '104,36,75',
         '105,88,44',
         '106,65,78',
         '107,44,99',
         '108,78,65',
         '109,60,76',
         '110,100,47',
         '111,54,88']
```

验证成功

为了适应虚拟机的环境，现在将表的列全部改为英文，便于处理。

对两个表的数据导入进行观察。

```
data1 = sqlContext.read.format("com.databricks.spark.csv").options(header='true',inferschema='true').load("hdfs://localhos
```

```
data1.show()
```

```
+------+-------+----+
|number|chinese|math|
+------+-------+----+
|   101|     99|  25|
|   102|     56|  68|
|   103|     74|  88|
|   104|     36|  75|
|   105|     88|  44|
|   106|     65|  78|
|   107|     44|  99|
|   108|     78|  65|
|   109|     60|  76|
|   110|    100|  47|
|   111|     54|  88|
+------+-------+----+
```

```
In [10]: data2 = sqlContext.read.format("com.databricks.spark.csv").options(header='true',inferschema='true').load("hdfs://localhos
```

```
In [11]: data2.show()
+------+---+----+
|number|sex|tall|
+------+---+----+
|   101|  F| 180|
|   102|  M| 176|
|   103|  M| 164|
|   104|  F| 170|
|   105|  F| 158|
|   106|  F| 178|
|   107|  M| 169|
|   108|  M| 165|
|   109|  F| 176|
|   110|  F| 187|
|   111|  M| 166|
+------+---+----+
```

为了解决题目中的问题，我决定在 pyspark 中使用 sql 语句进行操作。

首先要将读入的数据注册成表。

```
+------+---+----+
```

```
In [12]: data1.registerTempTable("table1")
         data2.registerTempTable("table2")
```

```
In [13]: import pyspark.sql
```

得到男性数学得分 top3

```
In [16]: result1 = sqlContext.sql("""select math from table1 join table2 on table1.number = table2.number where sex = 'M' order by m
```

```
In [17]: result1.show(3)
+----+
|math|
+----+
|  99|
|  88|
|  88|
+----+
only showing top 3 rows
```

男性身高高于 175 且语文及格的同学总数

```
In [5]: result1 = sqlContext.sql("""select count(*) from table1 join table2 on table1.number = table2.number where sex = 'M' and ch
```

```
In [6]: result1.show()
+--------+
|count(1)|
+--------+
|       2|
+--------+
```

男性数学及格的同学中最高身高同学的学号

```
In [8]: result1 = sqlContext.sql("""select table1.number from table1 join table2 on table1.number = table2.number where sex = 'M' a
```

```
In [9]: result1.show(1)
+------+
|number|
+------+
|   102|
+------+
only showing top 1 row
```

高于 170 的女性同学的平均总分

In [11]:
```
result1 = sqlContext.sql("""select avg(math)+avg(chinese) from table1 join table2 on table1.number = table2.number where se
```

In [12]:
```
result1.show()
```

```
+-------------------------+
|(avg(math) + avg(chinese))|
+-------------------------+
|                    137.5|
+-------------------------+
```

男性同学平均总分

In [13]:
```
result1 = sqlContext.sql("""select avg(math)+avg(chinese) from table1 join table2 on table1.number = table2.number where se
```

In [14]:
```
result1.show()
```

```
+-------------------------+
|(avg(math) + avg(chinese))|
+-------------------------+
|       132.16666666666669|
+-------------------------+
```

女性同学平均总分

In [15]:
```
result1 = sqlContext.sql("""select avg(math)+avg(chinese) from table1 join table2 on table1.number = table2.number where se
```

In [16]:
```
result1.show()
```

```
+-------------------------+
|(avg(math) + avg(chinese))|
+-------------------------+
|                    142.8|
+-------------------------+
```