6.6 使用Spark SQL读写数据库

Spark SQL可以支持Parquet、JSON、Hive等数据源，并且可以通过JDBC连接外部数据源

6.6.1 准备工作
6.6.2 读取MySQL数据库中的数据
6.6.3 向MySQL数据库写入数据

6.6.1 准备工作



- 在Linux中启动MySQL数据库

```
$ service mysql start
$ mysql -u root -p
#屏幕会提示你输入密码
```

- 输入下面SQL语句完成数据库和表的创建：

```
mysql> create database spark;
mysql> use spark;
mysql> create table student (id int(4), name char(20), gender char(4), age int(4));
mysql> insert into student values(1,'Xueqian','F',23);
mysql> insert into student values(2,'Weiliang','M',24);
mysql> select * from student;
```

6.6.1 准备工作



- 下载MySQL的JDBC驱动程序，比如mysql-connector-java-5.1.40.tar.gz
- 把该驱动程序拷贝到spark的安装目录” /usr/local/spark/jars”下
- 启动pyspark

```
$ cd /usr/local/spark
$ ./bin/pyspark
```

6.6.2 读取MySQL数据库中的数据



执行以下命令连接数据库，读取数据，并显示：

```
from pyspark.sql import SparkSession
if __name__ == '__main__':
    spark = SparkSession \
        .builder \
        .master("local[*]") \
        .appName("SparkReadMySQL") \
        .getOrCreate()
    jdbcDF=spark.read.\
        format("jdbc").\
        option("driver","com.mysql.cj.jdbc.Driver").\
        option("url","jdbc:mysql://localhost:3306/spark?useSSL=false").\
        option("dbtable","students").\
        option("user","root").\
        option("password","123456").\
        load()
    jdbcDF.show()
    spark.stop()
```

```
+---+----------+------+---+
| id|      name|gender|age|
+---+----------+------+---+
|  1|   Xueqian|     F| 23|
|  2|   Xhangyi|     M| 19|
|  3|     ywlee|     M| 66|
|  4|YoonSangOk|     M| 56|
+---+----------+------+---+
```

6.6.2 读取MySQL数据库中的数据



参数名称	参数值	含义
url	jdbc:mysql://localhost:3306/spark?useSSL=false	数据库连接地址
driver	com.mysql.cj.jdbc.Driver	JDBC驱动程序
dbtable	students	需要访问的表
user	root	数据库用户名
password	123456	数据库密码

6.6.3 向MySQL数据库写入数据



现在开始编写程序，往spark.students表中插入两条记录

```
from pyspark.sql import Row
from pyspark.sql.types import *
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
if __name__ == '__main__':
    spark = SparkSession \
        .builder \
        .config(conf=SparkConf()) \
        .getOrCreate()

    # 下面设置模式信息
    schema=StructType([StructField("id",IntegerType(),True), \
                       StructField("name",StringType(),True), \
                       StructField("gender",StringType(),True), \
                       StructField("age",IntegerType(),True)])
```

6.6.3 向MySQL数据库写入数据



```
# 下面设置两条数据，表示两个学生的信息
studentRDD=spark \
              .sparkContext \
              .parallelize(["3 RongCheng M 26","4 GuanHua M 27"]) \
              .map(lambda x:x.split(" "))

# 下面创建Row对象，每个Row对象都是rowRDD中的一行
# 对第1列去除空白后转整数，# 对第2列去除空白，# 对第3列去除空白，  # 对第4列去除空白后转整数
rowRDD=studentRDD.map(lambda p:Row(int(p[0].strip()),p[1].strip(),p[2].strip(),int(p[3].strip())))
# 建立起Row对象和模式之间的对应关系，也就是把数据和模式对应起来
studentDF=spark.createDataFrame(rowRDD,schema)

# 写入数据库
prop = {
    'user': 'root',
    'password': '123456',
    'driver': "com.mysql.cj.jdbc.Driver"
}

studentDF.write.jdbc("jdbc:mysql://localhost:3306/spark?useSSL=false",'students','append',prop)
spark.stop()
```

```
mysql> select * from students;
+------+-----------+--------+------+
| id   | name      | gender | age  |
+------+-----------+--------+------+
|    1 | Xueqian   | F      |   23 |
|    2 | Xhangyi   | M      |   19 |
|    3 | RongCheng | M      |   26 |
|    4 | GuanHua   | M      |   27 |
+------+-----------+--------+------+
4 rows in set (0.00 sec)
```

看看MySQL数据库中的spark.students表发生了什么变化

6.6.3 向MySQL数据库写入数据



