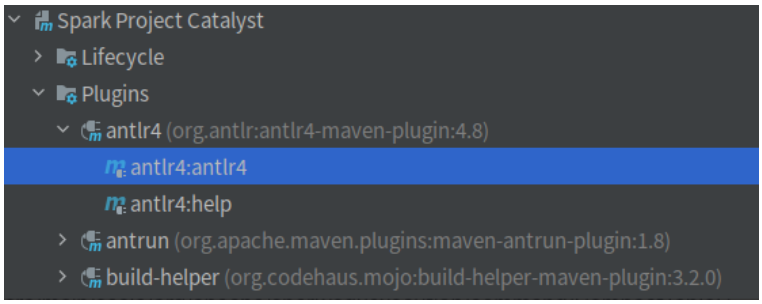
1.在SqlBase.g4中添加语法规则，路径为sql/catalyst/src/main/antlr4/org/apache/spark/sql/catalyst/parser/SqlBase.g4

```
statement
| COMPACT TABLE target=tableIdentifier partitionSpec?
    (INTO fileNum=INTEGER_VALUE FILES)?                          #compactTable
ansiNonReserved
| FILES
nonReserved
| FILES
//--SPARK-KEYWORD-LIST-START
FILES: 'FILES';
```

2.执行antlr4:antlr4插件，自动生成代码



3.SparkSqlParser.scala中添加visitCompactTable方法：

```
override def visitCompactTable(ctx: CompactTableContext): LogicalPlan = withOrigin(ctx) {
  val table: TableIdentifier = visitTableIdentifier(ctx.tableIdentifier())
  // 解析获得文件数
  val fileNum: Option[Int] = if (ctx.INTEGER_VALUE() != null) {
    Some(ctx.INTEGER_VALUE().getText.toInt)
  } else {
    None
  }
  // 解析获得partitionSpec，格式partition(key1=value1,key2=value2)
   val partition: Option[String] = if (ctx.partitionSpec() != null) {
     Some(ctx.partitionSpec().getText)
   } else {
     None
   }
   CompactTableCommand(table, fileNum, partition);
}
```

4.添加 CompactTableCommand 类的实现 在org.apache.spark.sql.execution.command包下新建CompactTableCommand类

```
package org.apache.spark.sql.execution.command

import org.apache.spark.sql.{Row, SaveMode, SparkSession}
import org.apache.spark.sql.catalyst.TableIdentifier
import org.apache.spark.sql.catalyst.expressions.{Attribute, AttributeReference}
import org.apache.spark.sql.types.StringType

case class CompactTableCommand(table: TableIdentifier,
                               fileNum: Option[Int],
                                partitionSpec: Option[String]) extends LeafRunnableCommand {

  private val defaultSize = 128 * 1024 * 1024
```

```scala

  override def output: Seq[Attribute] = Seq(
    AttributeReference("compact", StringType, nullable = false)()
  )

  override def run(sparkSession: SparkSession): Seq[Row] = {
    // 设置当前数据库
    sparkSession.catalog.setCurrentDatabase(table.database.getOrElse("default"))
    // 临时表格式: curTable_timestamp
    val tempTableName = "`" + table.identifier + "_" + System.currentTimeMillis() + "`"

    val originDataFrame = sparkSession.table(table.identifier)
    // 计算分区数，如果fileNum有效，则为fileNum，否则使用默认分区大小128M计算分区数
    val partitions = fileNum match {
      case Some(num) => num
      case None => (sparkSession.sessionState
        .executePlan(originDataFrame.queryExecution.logical)
        .optimizedPlan.stats.sizeInBytes / defaultSize).toInt + 1
    }

    if (partitionSpec.nonEmpty) {
      // 如果不更改这个参数，使用默认的static，在动态插入时，不管插入的分区是否存在，都会导致所有的分区被覆盖，数
      sparkSession.conf.set("spark.sql.sources.partitionOverwriteMode", "dynamic")
      // Dynamic partition strict mode requires at least one static partition column.
      // To turn this off set hive.exec.dynamic.partition.mode=nonstrict
      sparkSession.conf.set("hive.exec.dynamic.partition.mode", "nonstrict")

      // 当partitionSpec有值时，partition(key1=value1,key2=value2)转换为key1=value1 AND key2=value2的
      val conditionExpr = partitionSpec.get.trim
        .stripPrefix("partition(").dropRight(1)
        .replace(",", " AND ")

      // 设置分区数及where条件将数据写入临时表中
      originDataFrame
        .where(conditionExpr)
        .repartition(partitions)
        .write.mode(SaveMode.Overwrite)
        .saveAsTable(tempTableName)

      // 将临时表中的数据重新插入到原表中
      sparkSession
        .table(tempTableName)
        .write.mode(SaveMode.Overwrite)
        .insertInto(table.identifier)
    } else {
      // 当partitionSpec不存在时，设置分区数将数据读取到临时表中
      originDataFrame
        .repartition(partitions)
        .write
        .mode(SaveMode.Overwrite)
        .saveAsTable(tempTableName)

      // 读取临时表，对原表进行覆盖
```

```
66        sparkSession.table(tempTableName)
67          .write
68          .mode(SaveMode.Overwrite)
69          .saveAsTable(table.identifier)
70      }
71
72      // 删除临时表，此处不删除用于查看分区效果
73      // sparkSession.sql(s"DROP TABLE ${tempTableName}")
74
75      Seq(Row(s"compact table ${table.identifier} finished."))
76    }
77  }
```

5，编译

```
build/mvn clean package -DskipTests -Phive -Phive-thriftserver
```

6，运行

```
bin/spark-sql, COMPACT TABLE test INTO fileNum=100;
```

参考：https://gitee.com/joyjung/bigdata-homework-compact