### pom.xml

<dependency>

<groupId>org.apache.flink</groupId>

<artifactId>flink-table\_2.11</artifactId>

<version>1.7.2</version>

</dependency>

### 2.1 register dataset

开发：

#### 例子1：csv文件源（BatchTableEnvironment）

**package** com.sql;  
  
**import** org.apache.flink.api.common.typeinfo.TypeInformation;  
**import** org.apache.flink.api.common.typeinfo.Types;  
**import** org.apache.flink.api.java.DataSet;  
**import** org.apache.flink.api.java.ExecutionEnvironment;  
**import** org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;  
**import** org.apache.flink.table.api.StreamTableEnvironment;  
**import** org.apache.flink.table.api.Table;  
**import** org.apache.flink.table.api.TableEnvironment;  
**import** org.apache.flink.table.api.java.BatchTableEnvironment;  
**import** org.apache.flink.table.sources.CsvTableSource;  
  
**public class** source3 {  
 **public static void** main(String[] args)**throws** Exception{  
 ExecutionEnvironment env = ExecutionEnvironment.*getExecutionEnvironment*();  
 BatchTableEnvironment benv = BatchTableEnvironment.*getTableEnvironment*(env);  
 CsvTableSource cv = **new** CsvTableSource(**"D:\\a.csv"**, **new** String[]{**"name"**,**"age"**}, **new** TypeInformation[]{Types.***STRING***,Types.***INT***});  
 Table table11 = benv.fromTableSource(cv);  
 benv.registerTable(**"t11"**,table11);  
 Table t12 = benv.sqlQuery(**"select name,age from t11 where age>15"**);  
 t12.printSchema();  
 DataSet<ta2> ta2d = benv.toDataSet(t12, ta2.**class**);  
 ta2d.print();  
 }  
}

**package** com.sql;  
  
**public class** ta2 {  
 **public** String **name**;  
 **public int age**;  
 **public** ta2(){  
 }  
 @Override  
 **public** String toString(){  
 **return name**+**" "**+**age**;  
 }  
}

#### 例子2： dataset（TableEnvironment）

**package** com.sql;  
  
  
**import** org.apache.flink.api.java.DataSet;  
**import** org.apache.flink.api.java.ExecutionEnvironment;  
**import** org.apache.flink.api.java.tuple.Tuple2;  
**import** org.apache.flink.table.api.Table;  
**import** org.apache.flink.table.api.TableEnvironment;  
**import** org.apache.flink.table.api.java.BatchTableEnvironment;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** source5 {  
 **public static void** main(String[] args)**throws** Exception{  
 ExecutionEnvironment env = ExecutionEnvironment.*getExecutionEnvironment*();  
 BatchTableEnvironment tenv = TableEnvironment.*getTableEnvironment*(env);  
  
 List<Tuple2<String,Long>> t1 =**new** ArrayList<Tuple2<String,Long>>();  
 t1.add(**new** Tuple2<String,Long>(**"zs"**,10L));  
 t1.add(**new** Tuple2<String,Long>(**"ls"**,10L));  
 t1.add(**new** Tuple2<String,Long>(**"ww"**,10L));  
  
 DataSet<Tuple2<String,Long>> d1 = env.fromCollection(t1);  
  
 tenv.registerDataSet(**"table123"**,d1,**"name,age"**);  
 Table t123 = tenv.sqlQuery(**"select name,age from table123"**);  
 DataSet<ta3> ta3DataSet = tenv.toDataSet(t123, ta3.**class**);  
 ta3DataSet.print();  
 }  
}

**package** com.sql;  
  
**public class** ta3 {  
 **public** String **name**;  
 **public** Long **age**;  
 **public** ta3(){  
 }  
 @Override  
 **public** String toString(){  
 **return name**+**" "**+**age**;  
 }  
}

### 2.2Registerdatastream

开发：

**package** com.sql;  
**import** org.apache.flink.api.java.tuple.Tuple1;  
**import** org.apache.flink.api.java.tuple.Tuple2;  
**import** org.apache.flink.streaming.api.datastream.DataStream;  
**import** org.apache.flink.streaming.api.datastream.DataStreamSource;  
**import** org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;  
**import** org.apache.flink.table.api.StreamTableEnvironment;  
**import** org.apache.flink.table.api.Table;  
  
**public class** source2 {  
 **public static void** main(String[] args)**throws** Exception {  
 StreamExecutionEnvironment env = StreamExecutionEnvironment.*getExecutionEnvironment*();  
 org.apache.flink.table.api.java.StreamTableEnvironment tenv = StreamTableEnvironment.*getTableEnvironment*(env);  
 DataStreamSource<Tuple1<String>> source1 = env.fromElements(Tuple1.*of*(**"aa"**),Tuple1.*of*(**"bb"**));  
 tenv.registerDataStream(**"t1"**,source1,**"name"**);  
 Table tt = tenv.sqlQuery(**"select name from t1"**);  
 tt.printSchema();  
 DataStream<Tuple2<Boolean, tat>> tatDataStream = tenv.toRetractStream(tt,tat.**class**);  
 tatDataStream.print();  
 env.execute(**"a"**);  
 }  
}

**package** com.sql;  
  
**public class** tat {  
 **public** String **name**;  
 **public** tat(){  
 }  
  
 @Override  
 **public** String toString() {  
 **return name**;  
 }  
}

### 3.explain

**package** com.sql;  
  
**import** org.apache.flink.api.java.tuple.Tuple2;  
**import** org.apache.flink.streaming.api.datastream.DataStream;  
**import** org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;  
**import** org.apache.flink.table.api.Table;  
**import** org.apache.flink.table.api.TableEnvironment;  
**import** org.apache.flink.table.api.java.StreamTableEnvironment;  
  
**public class** source4 {  
 **public static void** main(String[] args)**throws** Exception{  
 StreamExecutionEnvironment env = StreamExecutionEnvironment.*getExecutionEnvironment*();  
 StreamTableEnvironment tEnv = TableEnvironment.*getTableEnvironment*(env);  
  
 DataStream<Tuple2<Integer, String>> stream1 = env.fromElements(**new** Tuple2<>(1, **"hello"**));  
 DataStream<Tuple2<Integer, String>> stream2 = env.fromElements(**new** Tuple2<>(1, **"hello"**));  
  
 Table table1 = tEnv.fromDataStream(stream1, **"count, word"**);  
 Table table2 = tEnv.fromDataStream(stream2, **"count, word"**);  
 Table table = table1  
 .unionAll(table2);  
  
 String explanation = tEnv.explain(table);  
 System.***out***.println(explanation);  
 }  
}