实操1：

改一改代码，将最近一分钟出现字符“b”的次数统计一下：

例如：最近一分钟，出现了12次

编码思路：添加时间窗口限定其时间为1分钟即可。

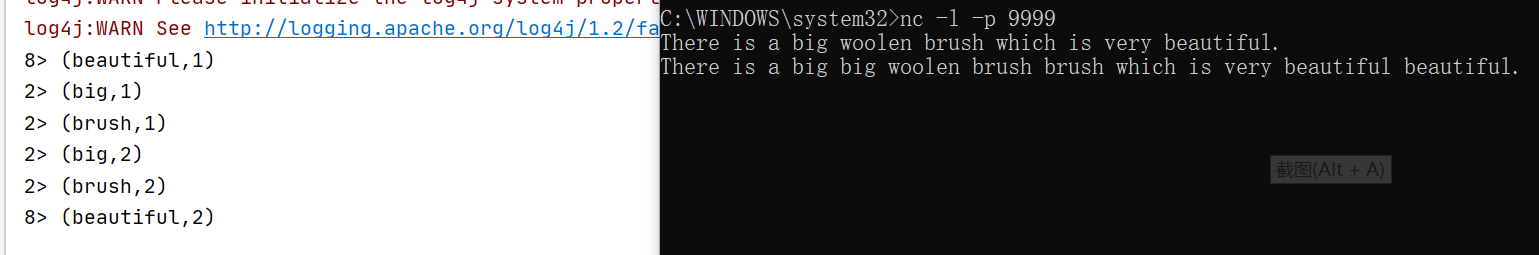
修改实操1 1\_flink模块Main.scala如下：

1. val stream = text.flatMap {
2. \_.toLowerCase.split("\\W+") filter {
3. \_.contains(target)
4. }
5. }.map{
6. (\_,1)
7. }.keyBy(0).timeWindow(Time.seconds(60)).sum(1)
8. stream.print()

使用说明：

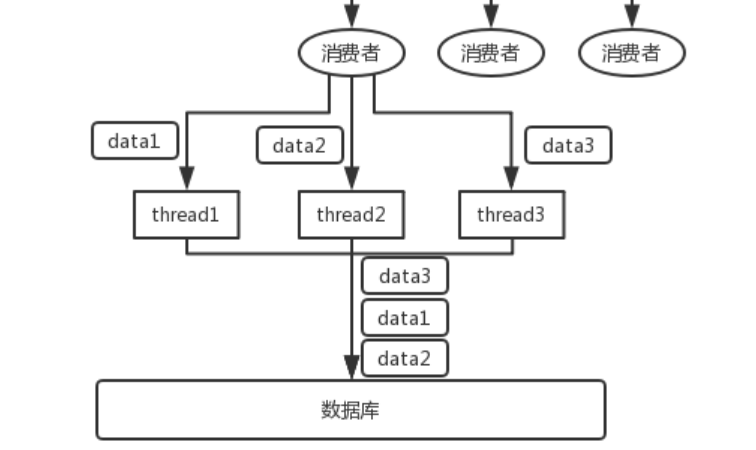
运行修改后的Main.scala

可在控制台看见每分钟输入的英文句子所筛选出含b的单词以及其出现次数。



实操2：

题目1：为什么kafka接入的数据，返回字段buy\_time并没有先后顺序？



消费者在读取数据时，多线程上的数据汇入同一个流，因此不存在先后顺序。

题目2：统计乘客到达数前5的城市

编码思路：

修改实操2代码，将项目修改为maven项目，在pom.xml中添加依赖，重写process方法选择每10秒钟窗口处理的数据排名前五的城市。

使用说明：

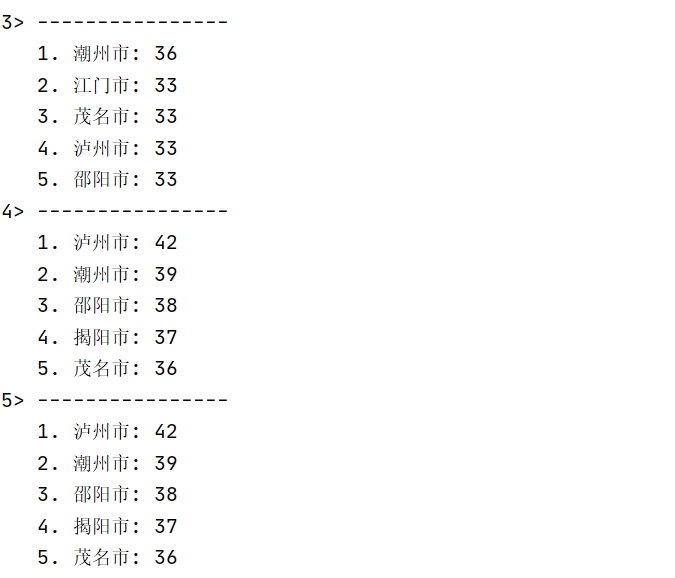
pom.xml中需要添加的依赖：

1. **<dependency>**
2. **<groupId>**com.alibaba**</groupId>**
3. **<artifactId>**fastjson**</artifactId>**
4. **<version>**1.2.62**</version>**
5. **</dependency>**

运行top5.scala

1. val JSONStream = inputKafkaStream.map(x =>JSON.parseObject(x))
2. val destinationStream = JSONStream.map(x =>(x.getString("destination"),1))
3. destinationStream.keyBy(0).window(SlidingProcessingTimeWindows.of(Time.seconds(60L), Time.seconds(10L)))
4. // 计算top5城市
5. .sum(1)
6. .windowAll(TumblingProcessingTimeWindows.of(Time.seconds(10L)))
7. .process(**new** ProcessAllWindowFunction[(String, Int), String, TimeWindow] {
8. override def process(context: Context, elements: Iterable[(String, Int)], out: Collector[String]): Unit = {
9. val top5 = elements.toSeq
10. .sortBy(-\_.\_2)
11. .take(5)
12. .zipWithIndex
13. .map { **case** ((item, sum), idx) => s"   ${idx + 1}. $item: $sum" }
14. .mkString("\n")
15. out.collect(("-" \* 16) + "\n" + top5)
16. }
17. })
18. .print()

控制台输出结果：



实操3：

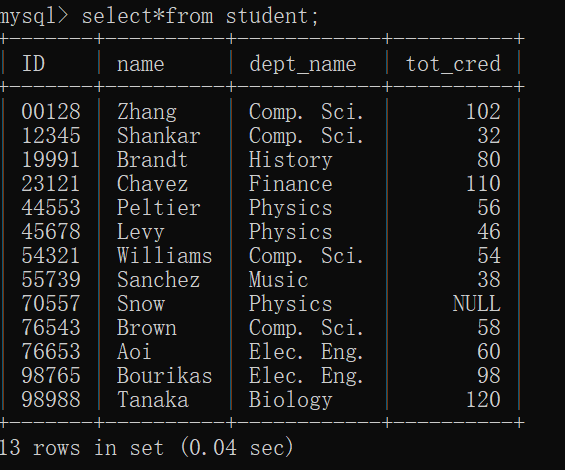
题目1：读取Mysql的数据入Kafka流

编码思路：编写mysql授权认证连接代码，以获取mysql中的数据，整合实操3的kafka生产数据的方法，完成读取Mysql的数据入Kafka流。

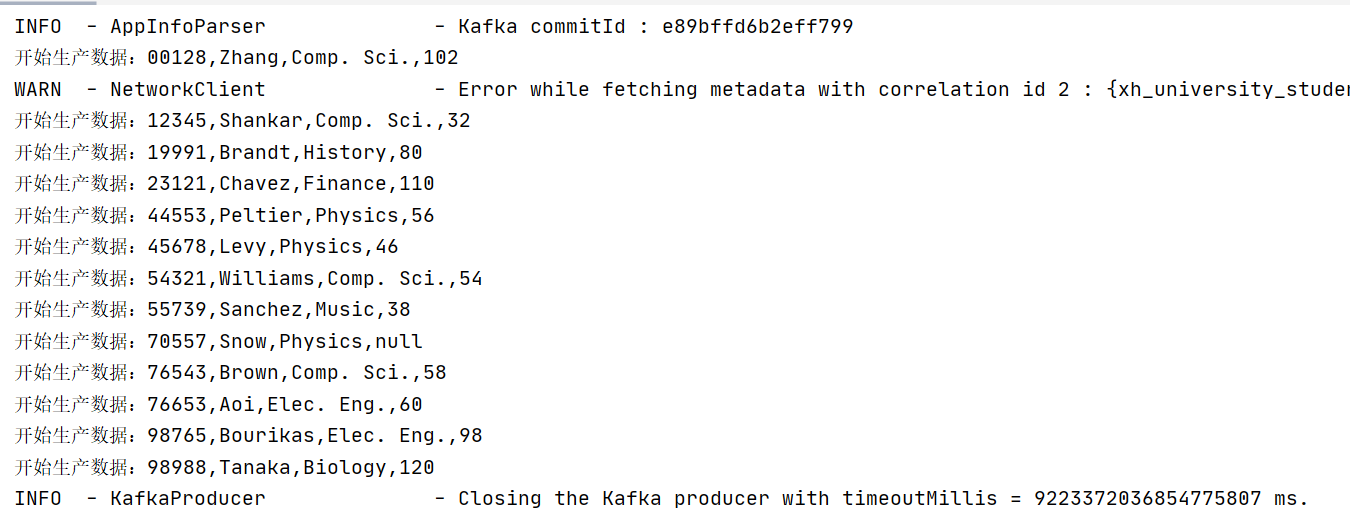
使用说明：

选择读入mysql已存在的university 数据库中的student表

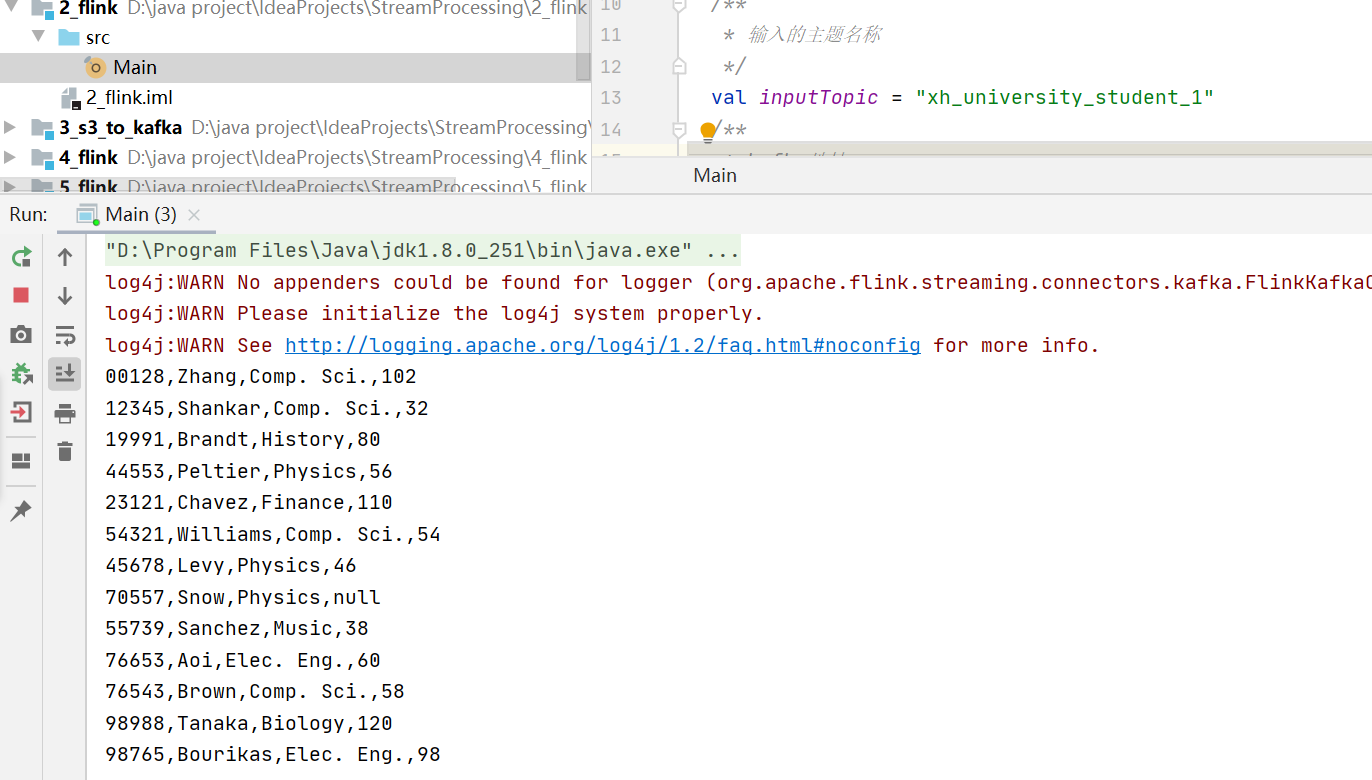
表中内容：



运行mysql\_to\_kafka项目中的Main.scala可以看见成功将mysql的student表输入到kafka topic val *topic* = "xh\_university\_student\_1"



通过实操2中的展示指定kafka topic数据可以看到mysql中的student表的确导入了val *topic* = "xh\_university\_student\_1"



实操4：

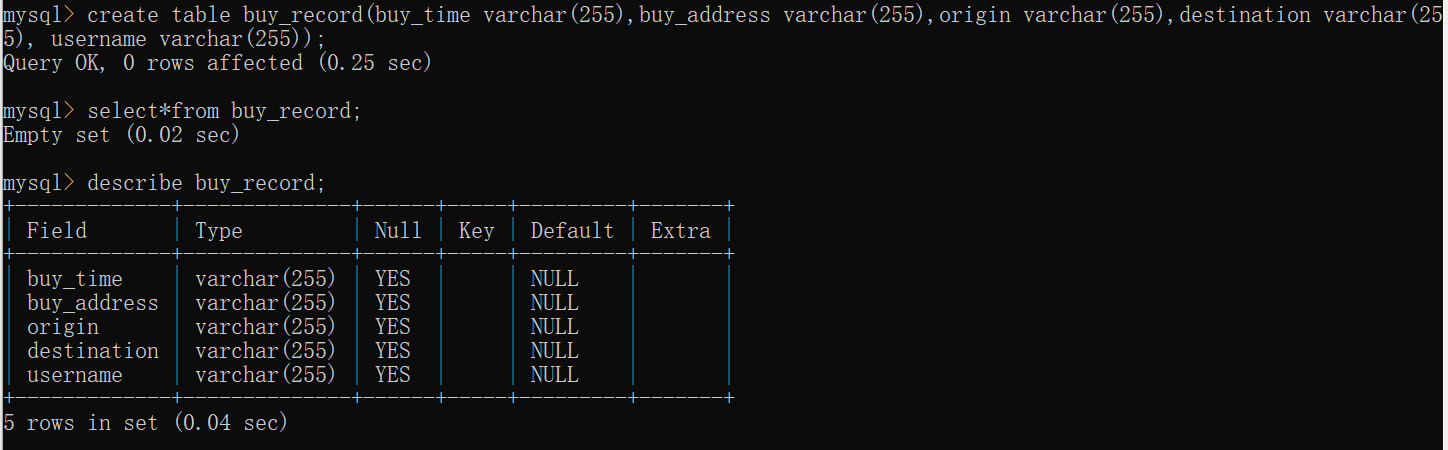
题目1：将flink流计算的结果数据入到Mysql中

编码思路：

将实操2中生产的新的kafka topic中的flink流数据计算结果输入到Mysql中，这里不进行过多的transformation，仅仅统计5秒内可以处理的流数据量，将其输入到mysql中

使用说明：

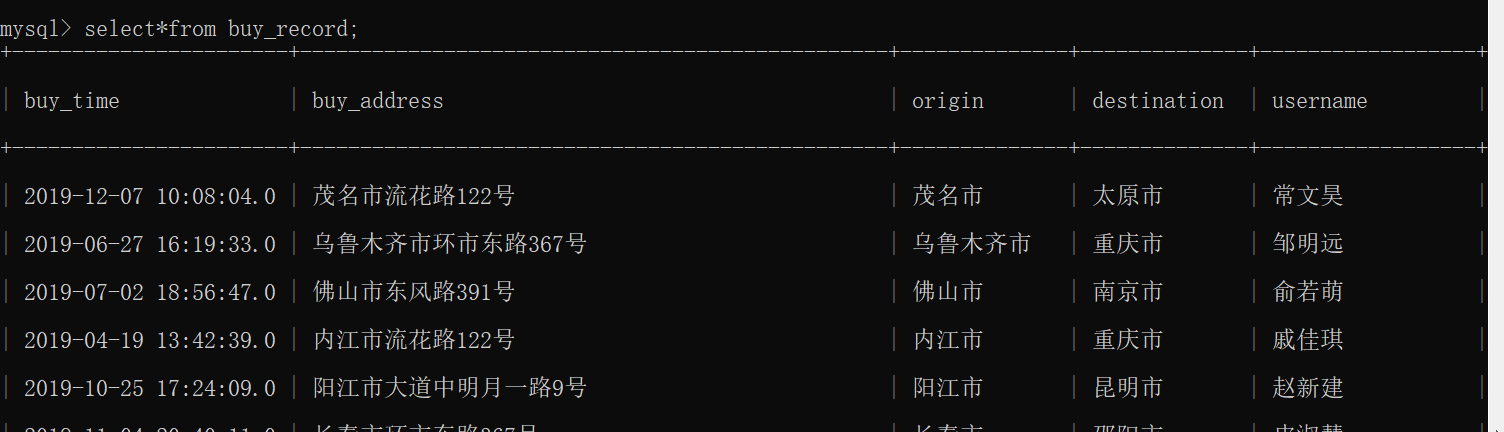
先在mysql中创建与flink流计算结果相同的表和字段

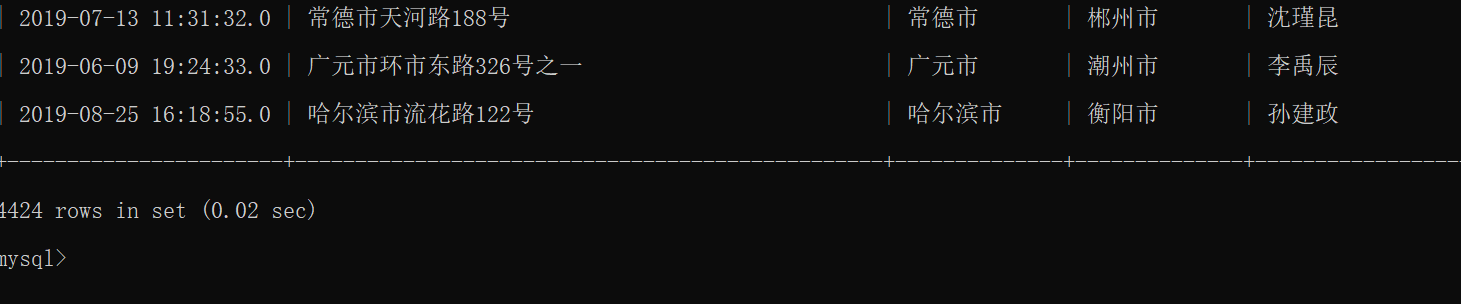


运行flink\_to\_mysql项目中的Main.java，可以看到输出了5秒内收到的流数据条数，和成功写入mysql的提示打印



到mysql中检查到buy\_record表的确导入了4424条数据。





实操5：在实操中，我们匹配关键字是写死在代码里，那么我们如何做到匹配关键字可以实时的输入？

可以通过控制台输入获取实时输入的关键字。