大数据技术基础课程实验报告

期末实验二：基于华为云的数据实时处理与展示综合实践

付容天

学号 2020211616

班级 2020211310

计算机学院（国家示范性软件学院）

2023 年 5 月 24 日

1. 基础配置部分

假设平台已经将每个商品的订单信息实时写入 Kafka 中，为了实现对采集到的数据进行实时处理并做关联数据库的查询展示，我们需要先进行一些基础配置，主要包括：

（1） 购买虚拟私有云 VPC；

（2） 开通数据接入服务（DMS Kafka）；

（3） 创建数据库服务（MySQL）；

（4） 创建数据表；

（5） 开通数据探索服务（DLI）。

下面我结合得分点给出这部分的试验记录和分析。

根据实验指导书，我创建了符合要求的 VPC，如下所示：

图 1：创建的 VPC 截图

接着，我按照实验指导书创建了 Kafka 实例，截图如下：

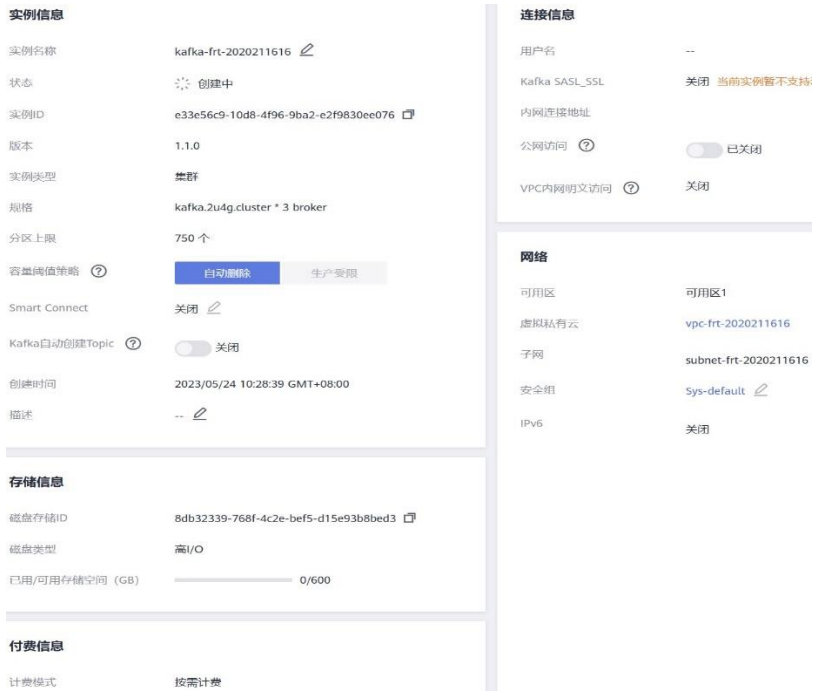

图 2：创建的 Kafka 实例截图

然后，我按照实验指导书要求创建了数据库服务，截图如下：

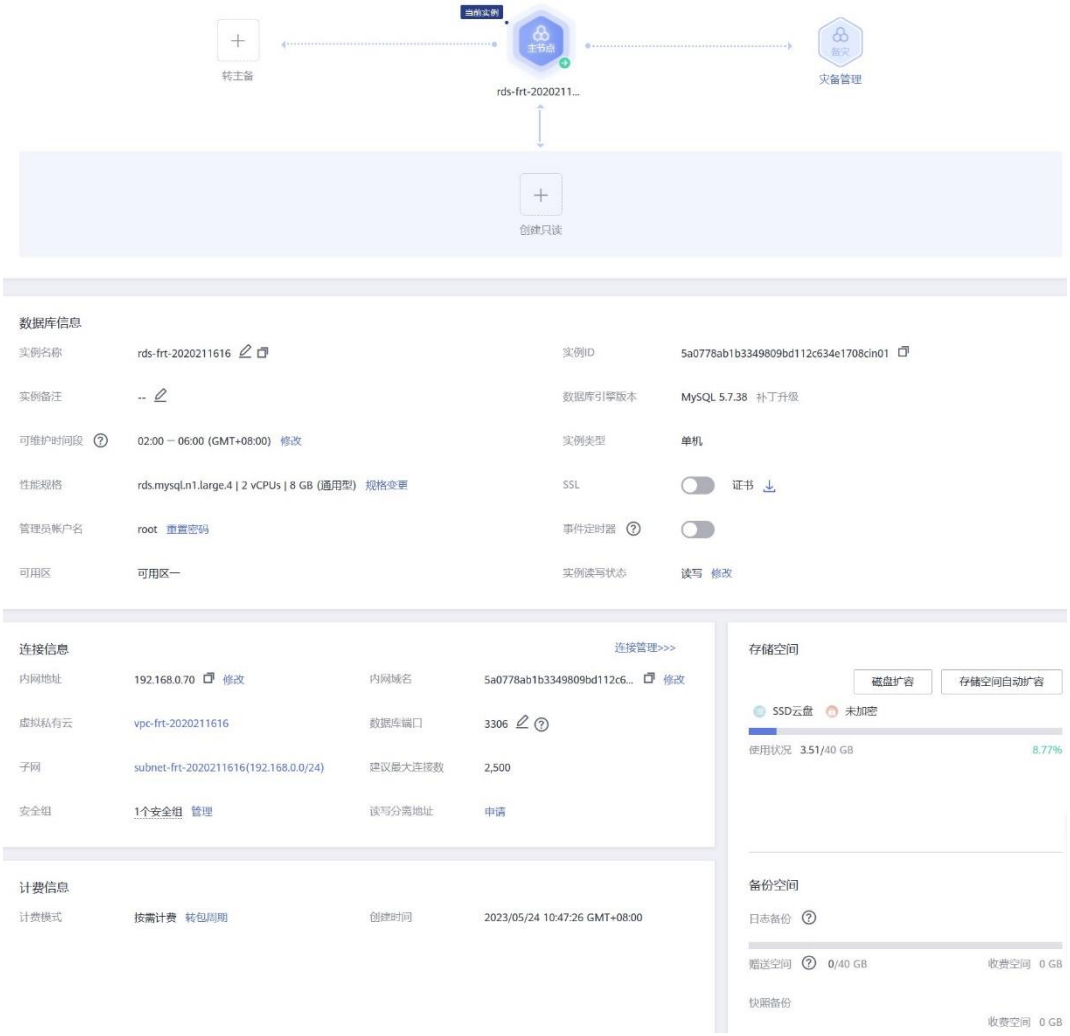

图 3：MySQL 数据库截图

在此基础上，我登录刚刚创建的数据库，创建了如下图所示的数据表：

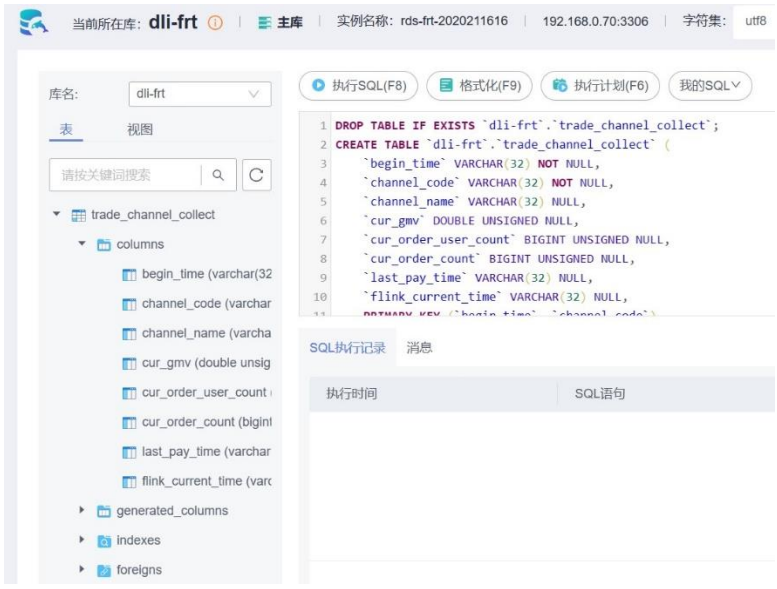

图 4：新建表格截图

最后，我在“数据湖探索 DLI”页面中开通了数据探索服务，并按照实验指导书的要求新建了相应的队列，命名为 dll_cce_queue_frt，如下所示：

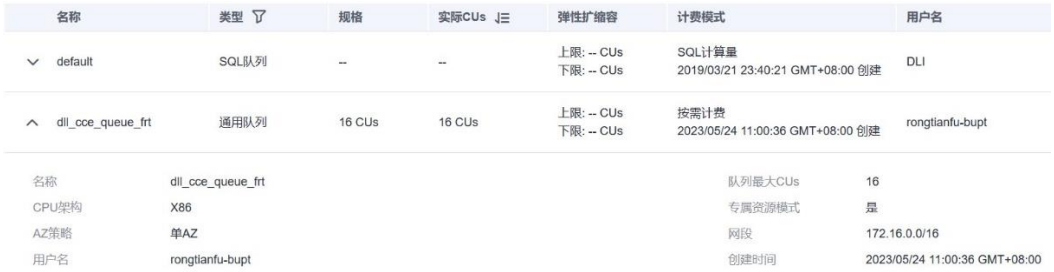

图 5：新建队列截图

并且，在数据湖页面配置跨源链接 connect_vpc_frt，并选择创建好的队列 dll_cce_queue_frt、选择虚拟私有云为 vpc_frt_2020211616，最后得到如下图所示的已激活的跨源连接截图：

图 6：跨源连接截图

紧接着测试 DLI 队列与 RDS 的联通性，下图 7 展示了 RDS 的地址和端口，图 8 展示了测试结果：

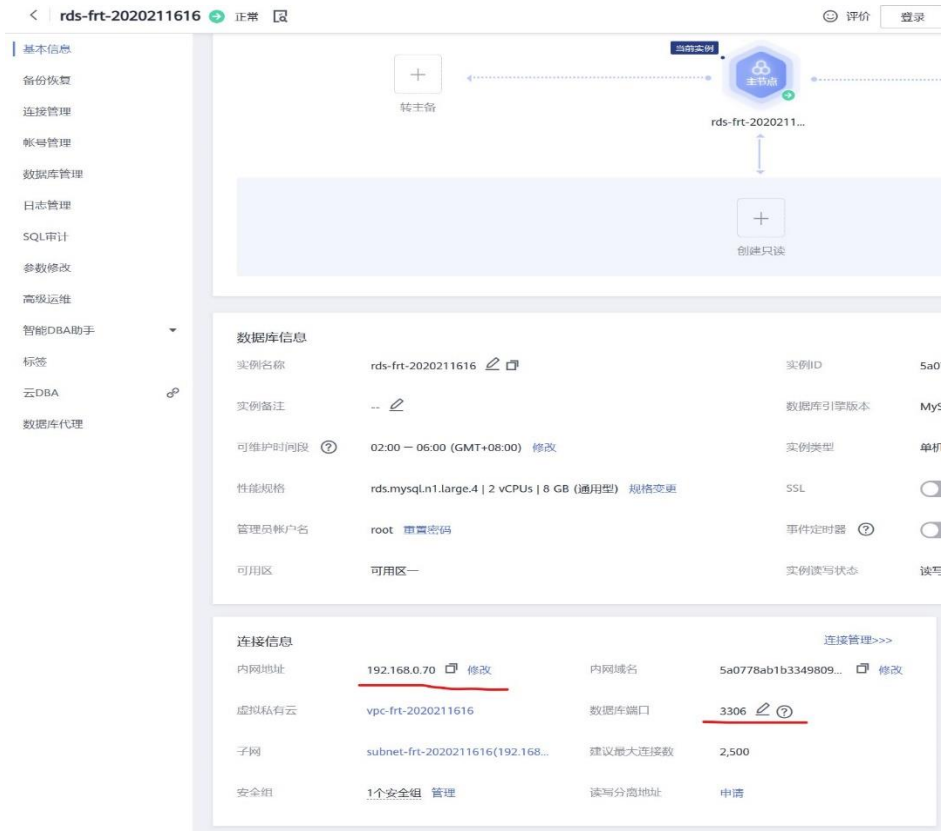

图 7：RDS 地址与端口信息（图中标红处）

图 8：RDS 联通性测试

之后测试 DLI 与 Kafka 实例的联通性，下图 9 展示了 Kafka 实例的地址与端口信息，图 10 展示了联通性测试结果：

图 9：Kafka 实例地址与端口信息

图 10：联通性测试结果（可达）

2. 新建作业、运行并查看

在这部分的实验中，我主要是完成了数据统计作业创建和运行，并在完成了必要配置之后输入了相关数据，最后查看结果，具体而言，先后完成了：

（1） 新建 Flink 作业；

（2） 新建云主机服务（ECS）并安装 Kafka 客户端；

（3） 向服务器发送数据；

（4） 从 MySQL 中查看接收到的数据。

下面我将结合得分点给出这部分实验的记录与分析。

首先，我根据实验指导书在数据湖页面中创建了新的 Flink 作业，在进行相应的配置与编辑后提交并启动 Flink 作业，得到如下所示的结果：

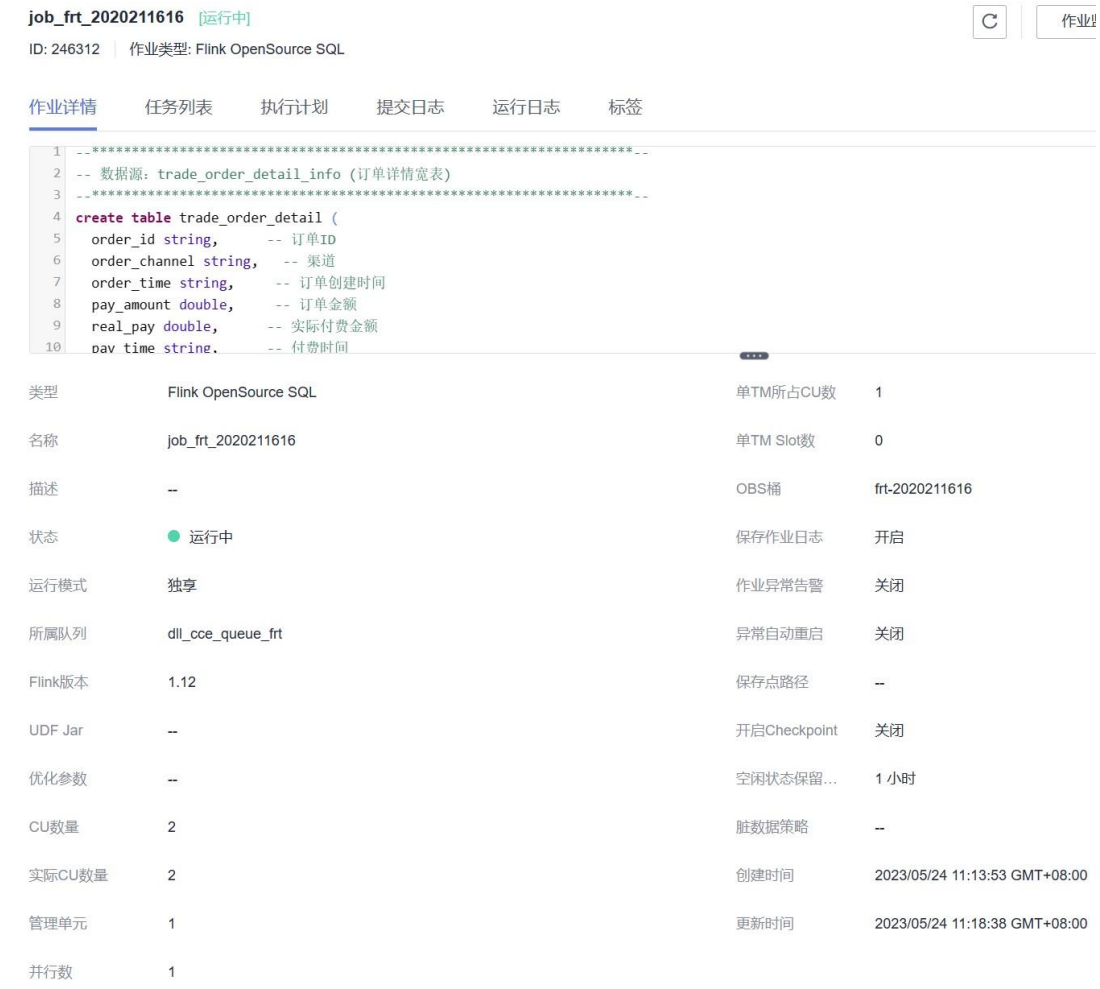

图 11：Flink 作业提交成功截图

然后，我购买了弹性云服务器 ECS，并在进行相应配置之后启动机器，在本地通过远程连接工具登录到服务器上，在安装完 Java 之后，输入命令查看相应状态，并截图如下：

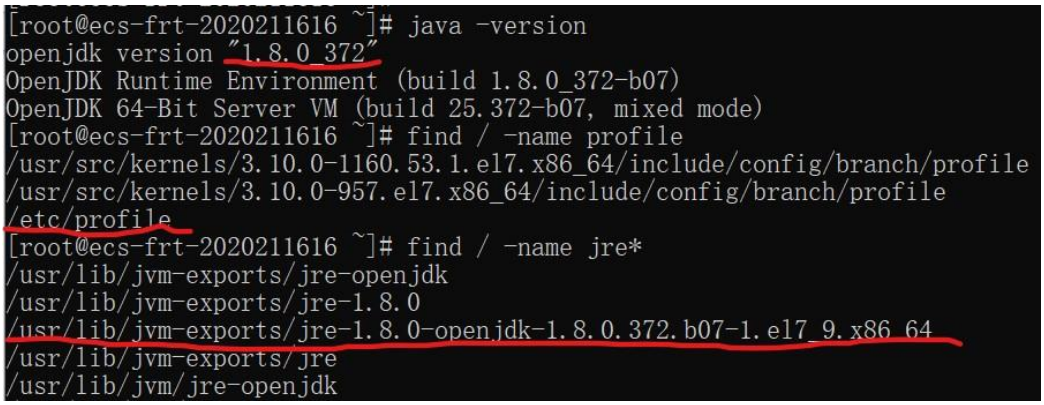

图 12：Java 安装截图

之后安装 Kafka 客户端并进入其中的 bin 目录，在此目录下启动相应脚本程序，进入数据发送页面，发送指定数据，截图如下所示：

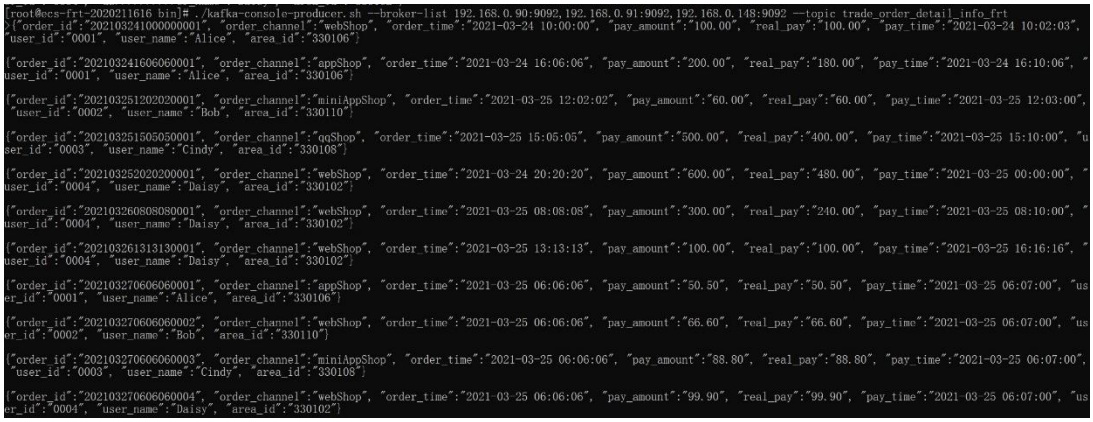

图 13：客户端发送消息至 Kafka 的截图

然后进入华为云中的 Kafka 实例内，点击消息查询，可以看到收到了如下所示的消息：

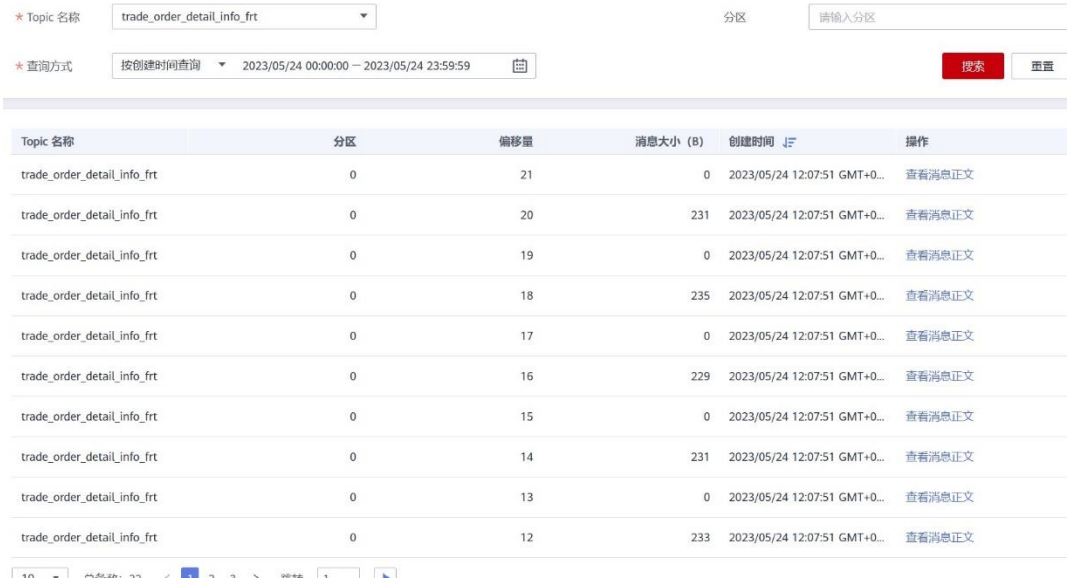

图 14：Kafka 收到消息的截图

最后，我从 MySQL 中查看接收到的数据，输入 SQL 查询语句，得到如下所示的结果：

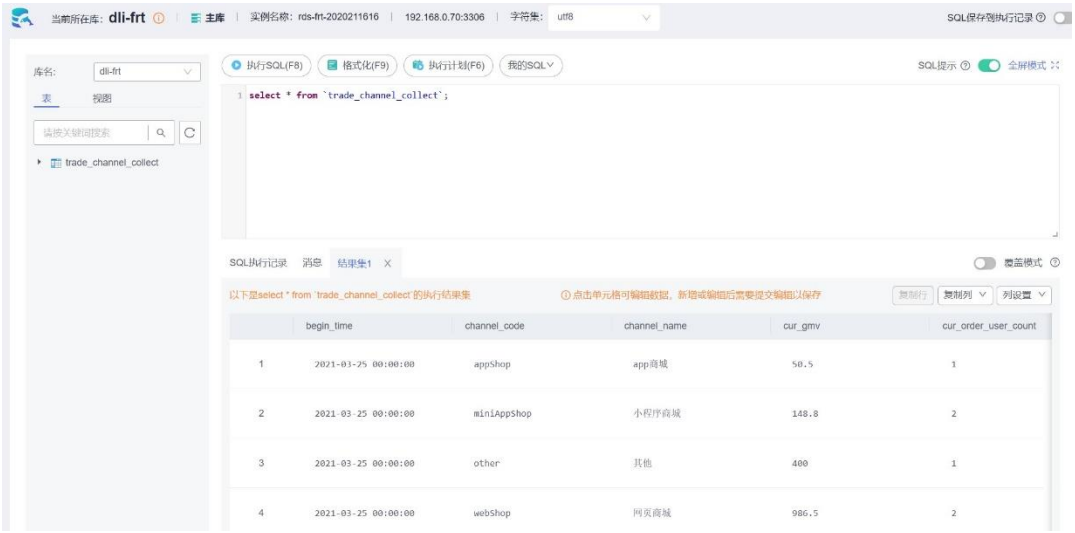

图 15：数据库查询结果截图

3. 实际应用部分

在这部分实验中，我们需要参考上面的实验流程，自行设计一个应用场景，完成数据的实时处理。我选择的应用场景是学生课程分数统计场景。

在学生课程分数统计场景中，数据集的每一行表示一个学生关于一门课的成绩，其数据结构如下所示：

学生 ID	学生班级	所在学院	课程名称	课程分数
stu_id (string)	stu_class (string)	stu_school (string)	course_name (string)	course_score (double)

我们需要实现的效果就是实时地统计出不同班级不同课程的最高分、最低分和平均分，因此结果表的数据结构设计为：

学生班级	课程名称	最高分	最低分	平均分
stu_class (string)	course_name (string)	max_score (double)	min_score (double)	avg_score (double)

我们所用的数据集完整内容在本报告最后的附录中给出。

现在来看我为这个场景和需求编写的 Flink 脚本：

```
--************************************************************
-- 数据源: stu_score_detail （学生课程成绩表）
--************************************************************
create table stu_score_detail (
  stu_id string,          -- 学生ID
  stu_class string,       -- 学生班级
  stu_school string,      -- 学生所在学院
  course_name string,     -- 课程名称
  course_score double,    -- 课程分数
) with (
  "connector" = "kafka",
  "properties.bootstrap.servers" = "192.168.0.90:9092,192.168.0.91:
9092,192.168.0.148:9092",              -- Kafka连接地址
  "properties.group.id" = "stu_score", -- Kafka groupID
  "topic" = "stu_score_frt",           -- Kafka topic
  "format" = "json",
  "scan.startup.mode" = "latest-offset"
);

--************************************************************
-- 结果表: stu_score_collect （各班级不同课程数据分析表）
--************************************************************
create table stu_score_collect(
  stu_class string,       -- 学生班级
  course_name string,     -- 课程名称
  max_score double,       -- 最高成绩
  min_score double,       -- 最低成绩
  avg_score double,       -- 平均成绩
  primary key (stu_class, course_name) not enforced
) with (
  "connector" = "jdbc",
  "url" = "jdbc:mysql://192.168.0.70:3306/dli-frt",
  "table-name" = "stu_score_collect",  -- mysql表名
  "username" = "root",                 -- mysql用户名
  "password" = "frtFRT189260",         -- mysql密码
  "sink.buffer-flush.max-rows" = "1000",
  "sink.buffer-flush.interval" = "1s"
);

--************************************************************
-- 具体统计操作
--************************************************************
```

```
insert into stu_score_collect
select
      stu_class
    , course_name
    , MAX(course_score) as max_score
    , MIN(course_score) as min_score
    , AVG(course_score) as avg_score
from stu_score_detail
GROUP BY stu_class, course_name;
```

首先创建数据源表 stu_score_detail，并从 Kafka 中得到相应的数据，并通过 SQL 语法所支持的 insert 语句、select 语句、group by 结构、max 特性等功能得到每个班级每门课程的最高分、最低分和平均分，最后将这些数据输入到结果表 stu_score_collect 中。

我创建了相应的 topic 为 stu_score_frt，并将如上所述的 Flink 脚本提交运行，如下图所示：

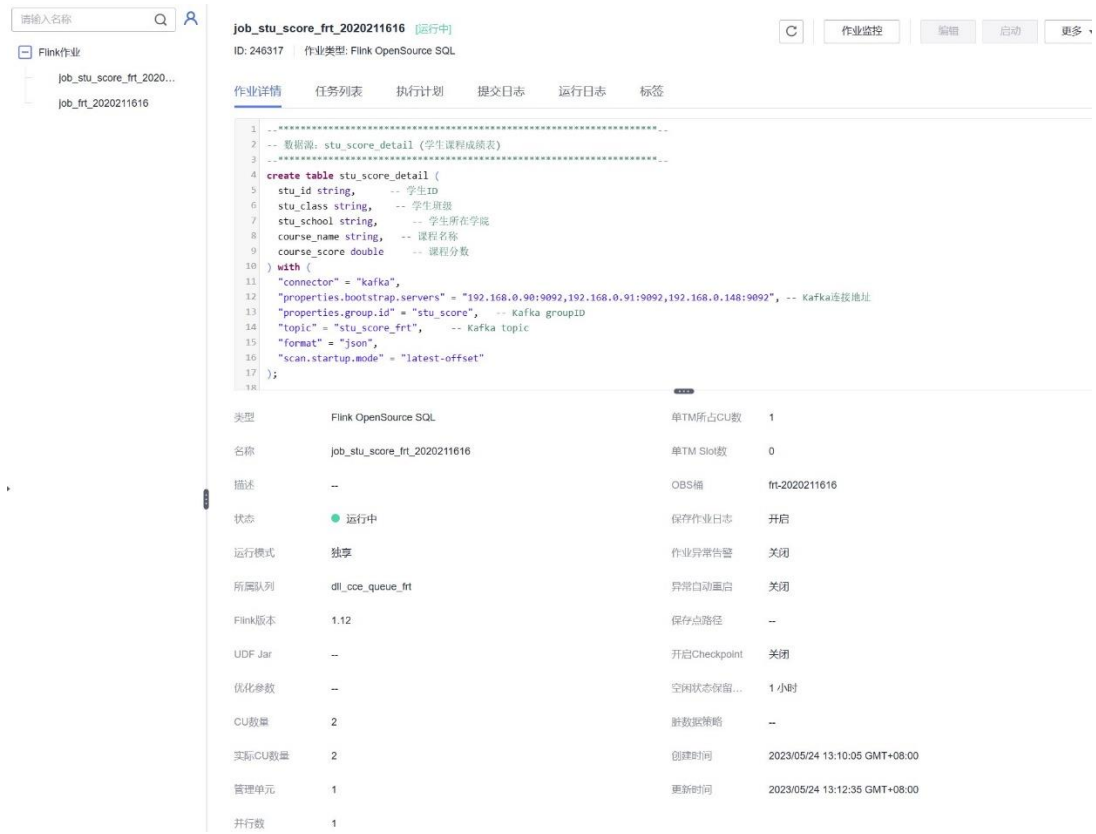

图 16：自编 Flink 脚本正常运行截图

并且，我在 MySQL 中创建了相应的表格，该表格就是前面所述的“结果表”，截图如下所示：

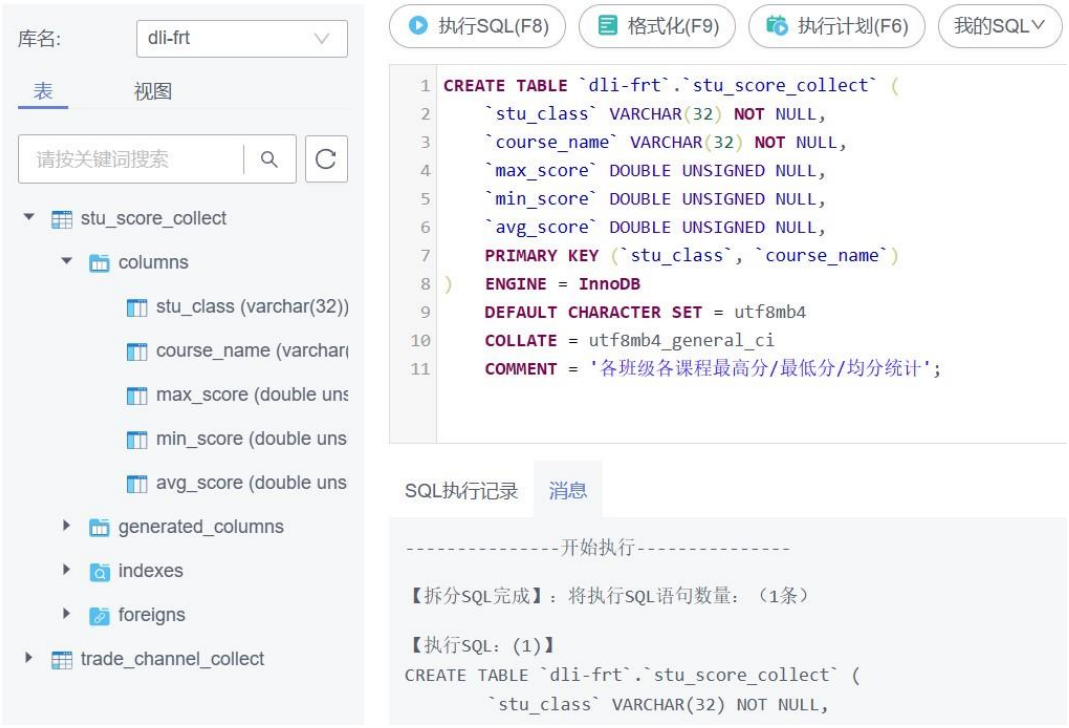

图 17：创建结果表

然后在本地通过远程连接工具连接到服务器上，并上传如附录所示的待统计数据，截图如下：

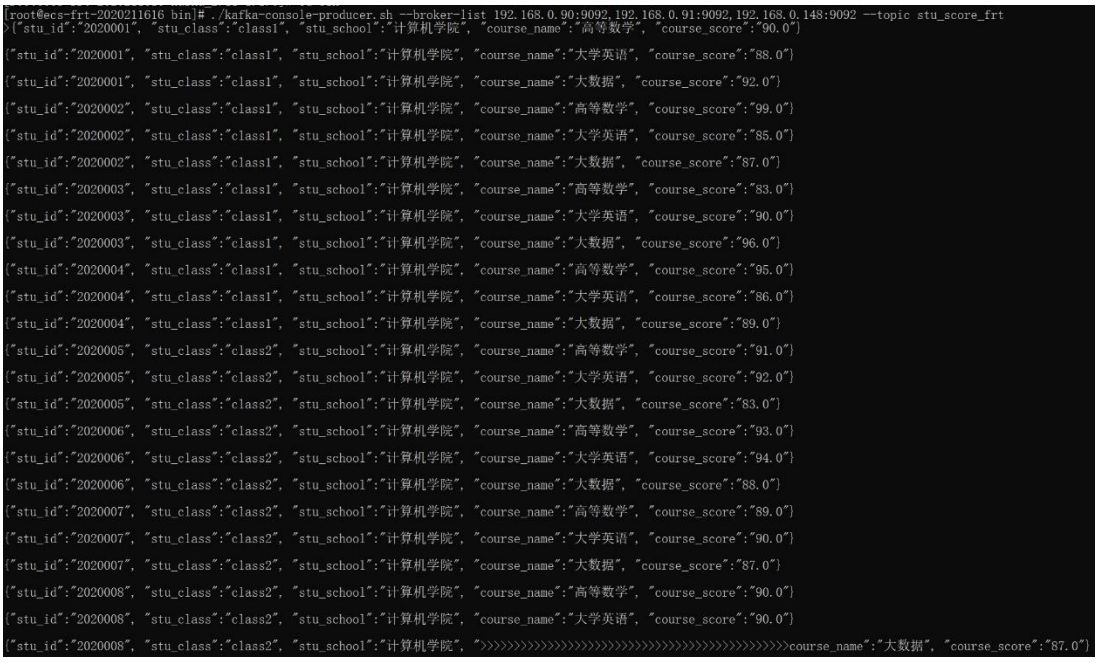

图 18：上传待处理数据

最后，我打开了 MySQL 页面，通过相应查询语句，得到了数据处理结果，每条记录表示了每个班级每门课程的最高分、最低分和平均分，查询结果截图如下所示：

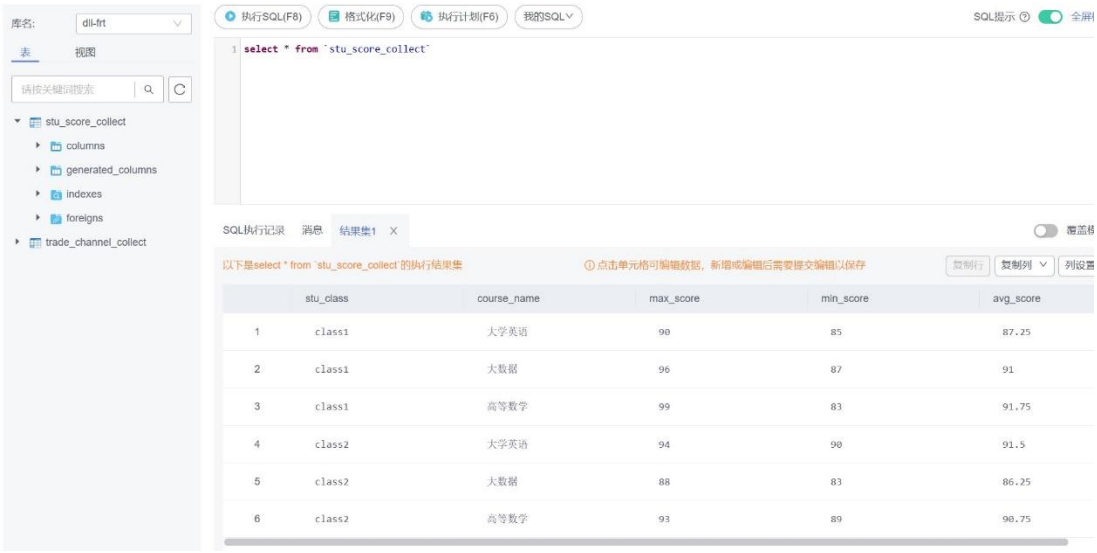

图 19：查询得到数据处理结果

4. 实验问题与实验总结

在本次实验中，我也遇到了几个问题，现记录如下：

（1） 联通性测试失败：可能是需要等待一段时间，一开始测试失败，但我耐心地多测试了几次，后面就一直是成功联通的；

（2） SQL 语句不熟：主要体在编写自己的 Flink 脚本的时候犯基本语法的错误，通过复习数据库课程知识，解决了这个问题。

本次实验是大数据技术基础课程的最后一个实验，我在之前的实验环境下结合实验指导书完成了所有的实验任务。我实现了 Flink 从 Kafka 中实时读取数据、进行统计并输出的功能。我还自行设计了基于实际情况的学生课程成绩统计实验，编写了相应的 Flink 脚本对输入的学生课程成绩进行实时统计并显示。总的来说，我在本次实验中复习了理论知识、进行了工程实践，圆满完成了实验任务，收获颇丰！

在这学期的大数据技术基础课程中，我在鄂海红老师和各位助教、同学的帮助下圆满完成了所有既定的实验任务，对当今热门的大数据技术有了基本的认识。鄂老师耐心细致的讲解、助教和同学们的热心帮助，是我能够完成这门课程的重要原因！感谢鄂老师给我们提供实验环境并进行非常到位的讲解，感谢各位助教以及所有一起参加本课程同学们！

附录：待处理数据

{"stu_id":"2020001", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"90.0"}

{"stu_id":"2020001", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"88.0"}

{"stu_id":"2020001", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"92.0"}

{"stu_id":"2020002", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"99.0"}

{"stu_id":"2020002", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"85.0"}

{"stu_id":"2020002", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"87.0"}

{"stu_id":"2020003", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"83.0"}

{"stu_id":"2020003", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"90.0"}

{"stu_id":"2020003", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"96.0"}

{"stu_id":"2020004", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"95.0"}

{"stu_id":"2020004", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"86.0"}

{"stu_id":"2020004", "stu_class":"class1", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"89.0"}

{"stu_id":"2020005", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"91.0"}

{"stu_id":"2020005", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"92.0"}

{"stu_id":"2020005", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"83.0"}

{"stu_id":"2020006", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"93.0"}

{"stu_id":"2020006", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"94.0"}

{"stu_id":"2020006", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"88.0"}

{"stu_id":"2020007", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"89.0"}

{"stu_id":"2020007", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"90.0"}

{"stu_id":"2020007", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"87.0"}

{"stu_id":"2020008", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"高等数学", "course_score":"90.0"}

{"stu_id":"2020008", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大学英语", "course_score":"90.0"}

{"stu_id":"2020008", "stu_class":"class2", "stu_school":"计算机学院", "course_name":"大数据", "course_score":"87.0"}