案例一：公司新闻舆情分析

背景

公司希望分析行业内各个公司的企业新闻数据，为各类新闻打标签（正面、中性、负面），然后分析行业负面新闻情况。新闻数据字段如下：

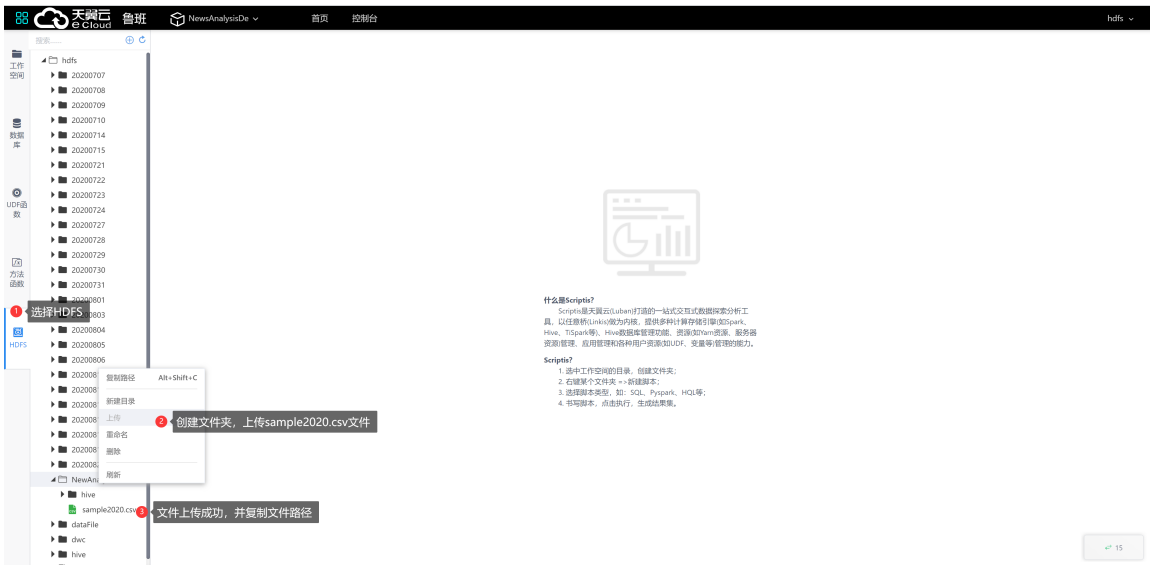
记录 ID	新闻标题	新闻日期	企业名字	新闻编号	新闻标签

需求为生成某个时间段的企业负面新闻总数数据：

日期	企业名字				往前	第 N 周	负面	新闻	总数

操作步骤

Step 1：上传数据至鲁班平台

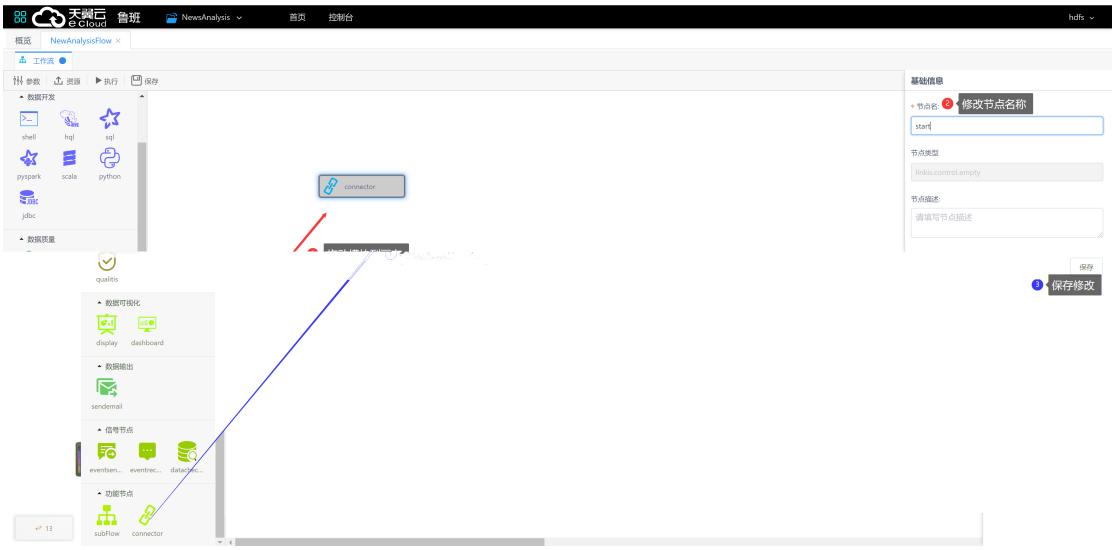


1. *工作空间—>数据分析—> scrpits->选择 HDFS*
2. *右键 hdfs 根目录，创建文件夹 NewAnalysis*
3. *右键 NewAnalysis 上传文件*
4. *右键文件，复制文件路径，右键文件夹，复制文件夹路径*

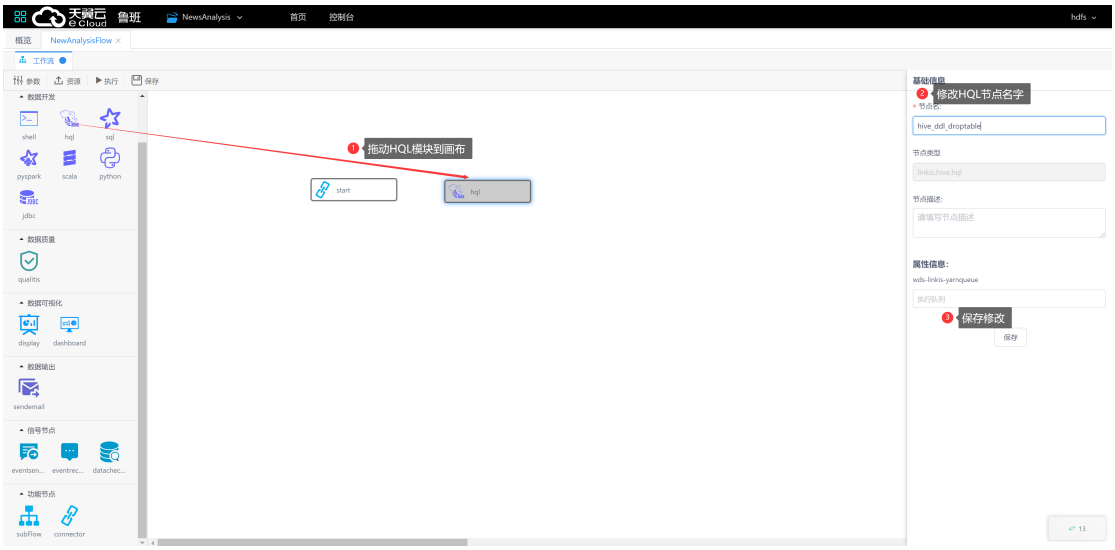
Step 2：**创建工作流**

在工作空间中，点击【应用开发】标签，进入【工作流开发】页面，点击【创建工程】按钮，输入工程信息后，点击【创建工作流】，输入工作流信息，成功后点击刚创建的工作流，进入工作流开发页面

Step 3：**编辑工作流**



1. *添加 connector 节点（拖动 connector 模块到画布）*
2. *右键 connector 节点，弹出节点参数设置栏，输入节点名：start，点击保存*



3. *添加 hql 节点（拖动 hql 模块到画布）*
4. *单击 hql 节点，弹出节点参数设置栏，输入节点名：hive_ddl_droptable，点击保存*

5. *双击节点进入编辑界面，粘贴如下代码*

```
drop table if exists ODS_T_NEWS;

drop table if exists DWS_T_NEWS;

drop table if exists DWA_T_NEW_NEGATIVE;

drop table if exists DWA_T_NEW_POSTIVE;
```

6. *点击 保存，保存脚本后，点击 运行，调试脚本*

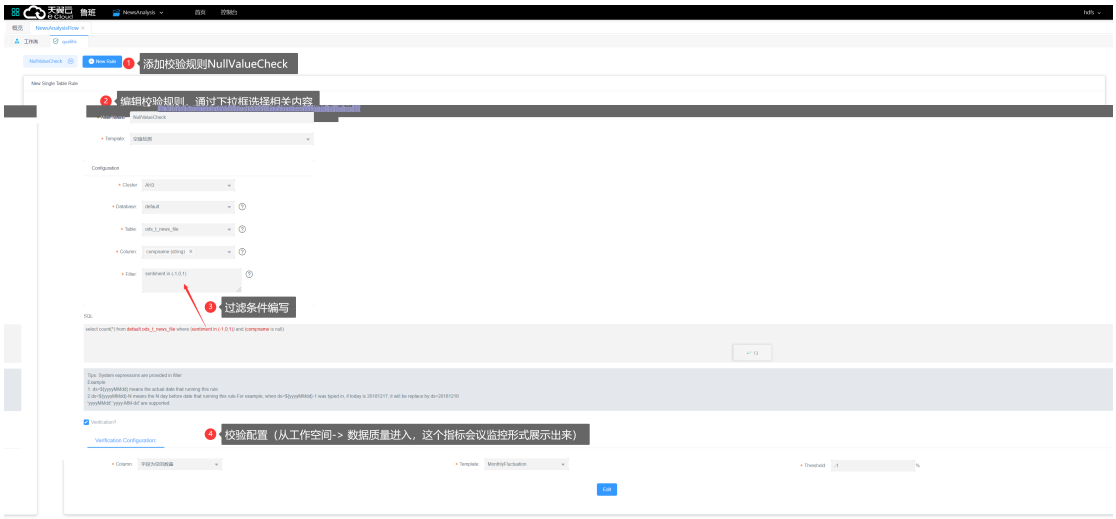
后续按照以上步骤分别添加节点（各节点代码见附录）：

1. *1 个 Hql 节点，命名为 hive_ddl_createtable*
2. *4 个 Scala 节点:，命名为 genData_1、NewByDay、NegNewByWeek、hiveToMysql*
3. *1 个 jdbc 节点，命名为 mysql_ddl*
4. *1 个 qualitis，命名为 qualitis*

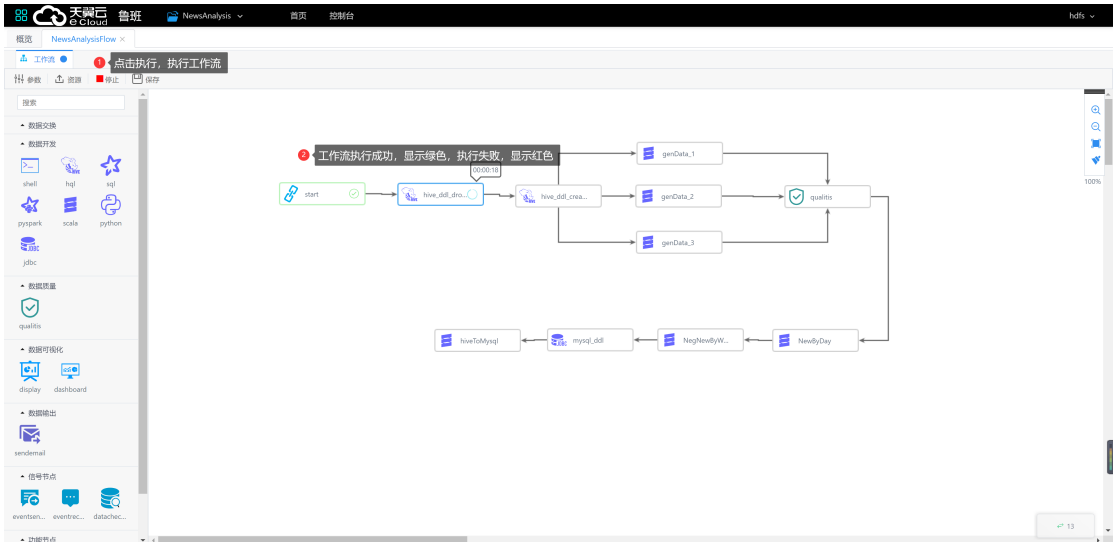
随后，点击 genData_1 节点，直接在画布上复制出 2 份，分别命名为 genData_2 和 genData_3

其中，qualitis 配置如下：

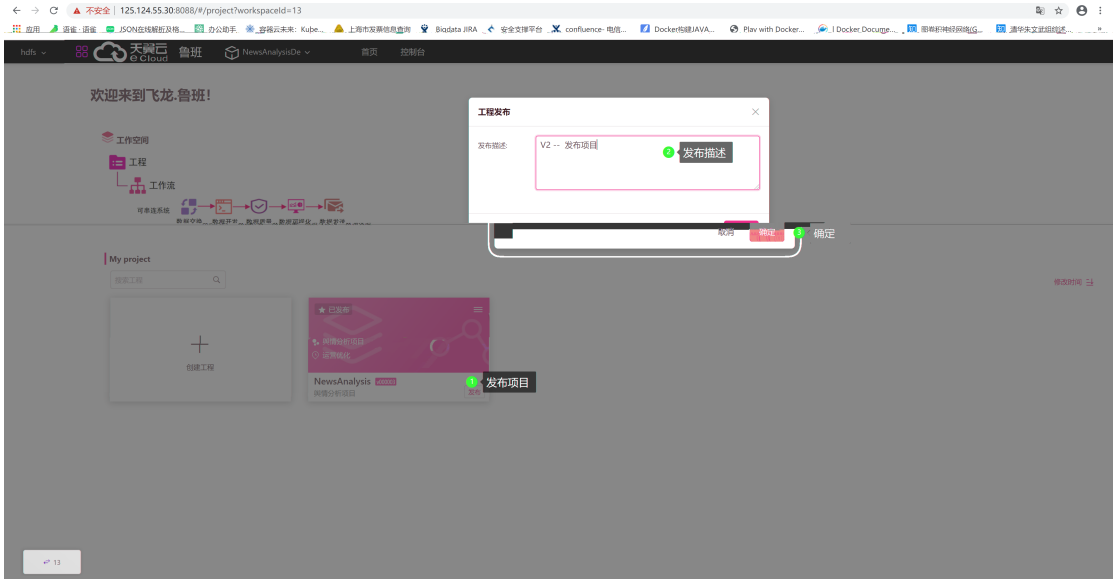
1. *拖动 qualitis 图标至画布*
2. *双击 qualitis 节点，进入数据质量开发界面，并配置如下*



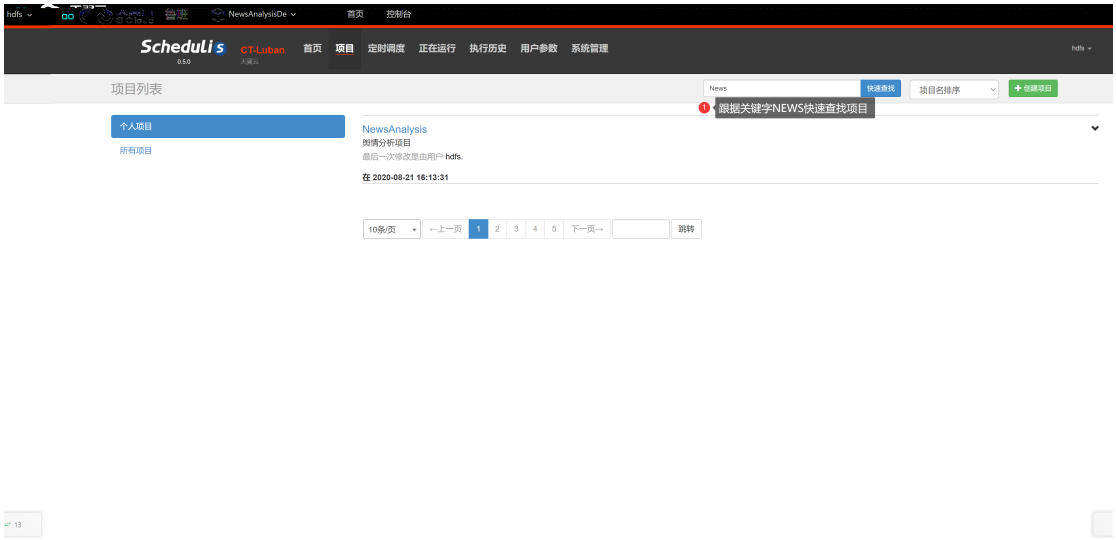
最后，添加节点的依赖关系，如下图，只需要使用鼠标连线即可：



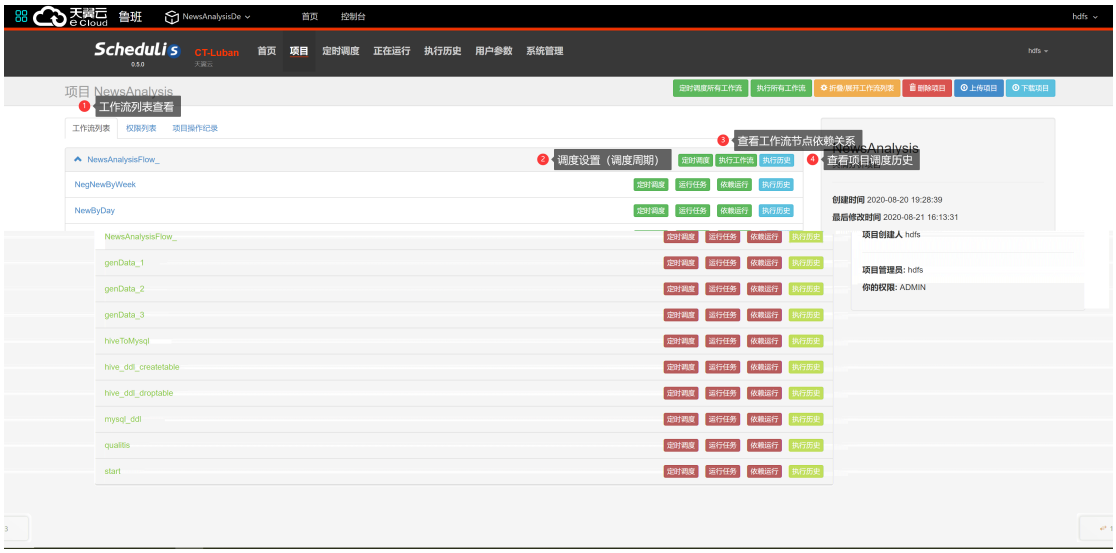
Step 4：发布项目至调度系统



1. *工作空间—>应用开发—> 工作流开发->选择 HDFS*
2. *点击发布，发布项目*

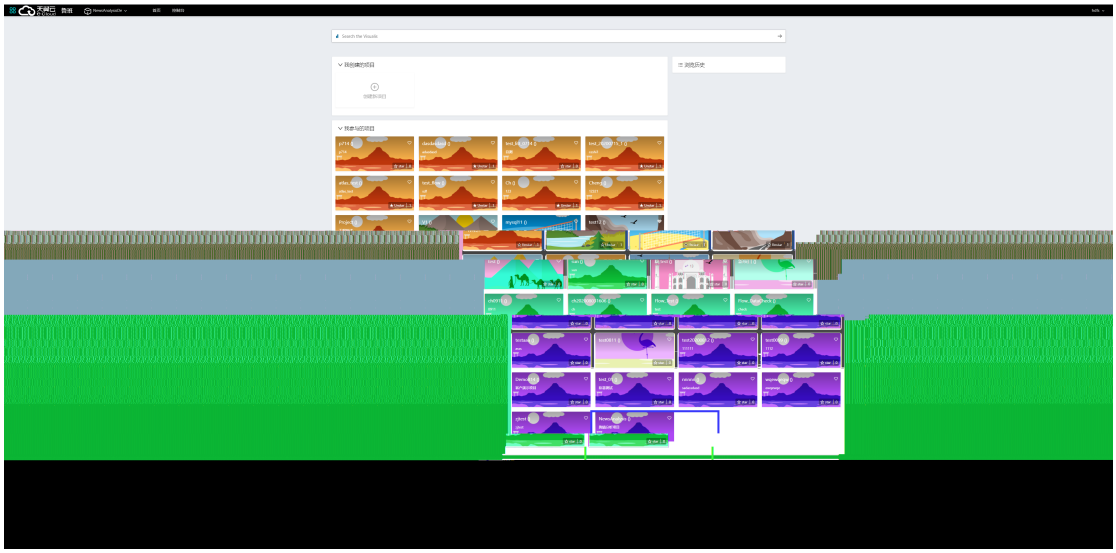


3. *进入当前【工作空间】—>【生产运维】—> 【schedulis】功能中，找到该项目*

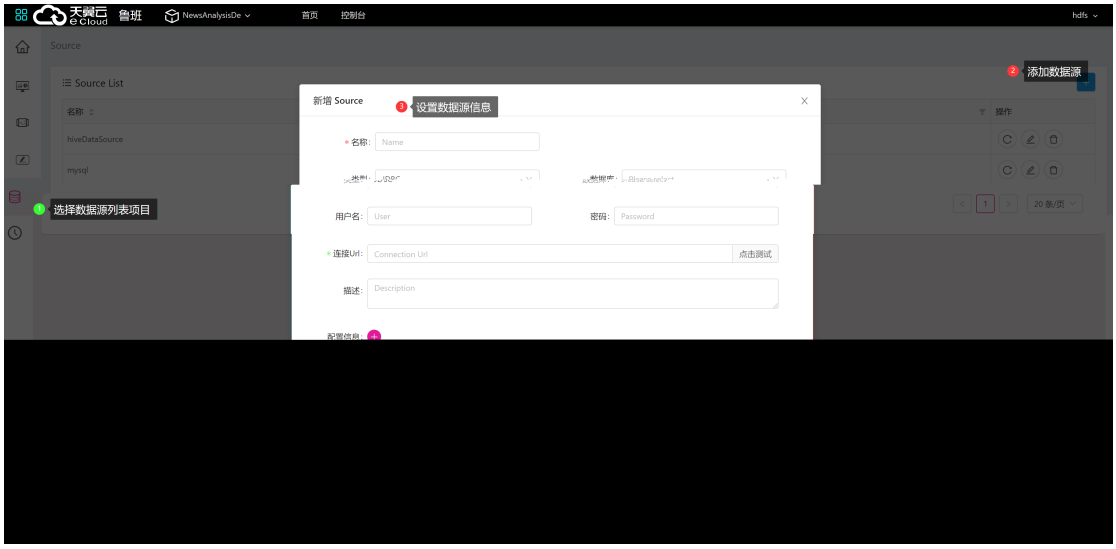


4. *点击【执行工作流】，开始运行当前工作流*

Step 5：**定制报表**

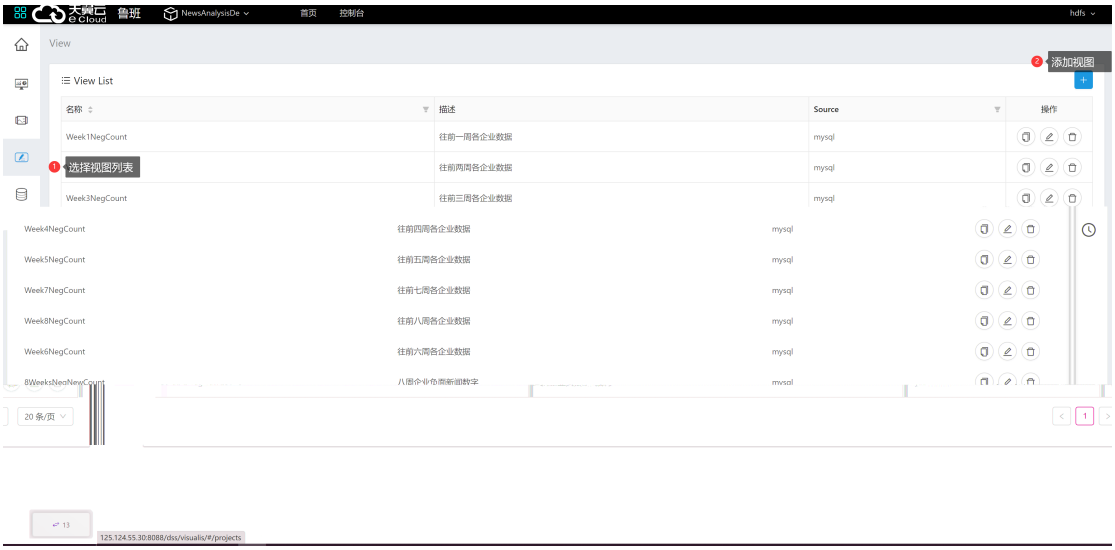


1. *进入【工作空间】-> 【数据分析】 -> 【visualis】单击进入与**项目同名**的可视化项目*

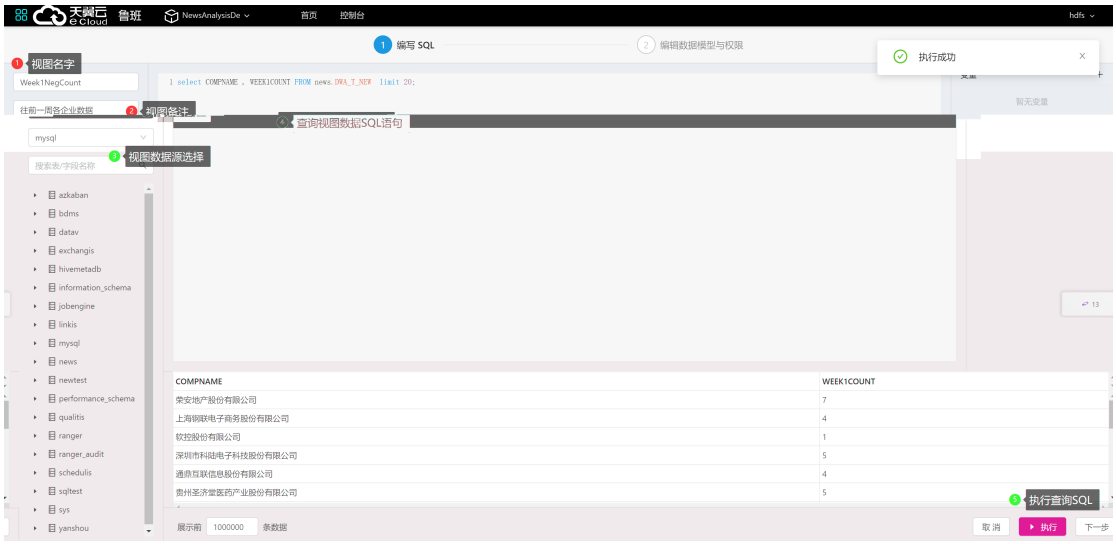


2. 添加数据源 -> 编辑数据源属性，关键信息如下

jdbc:mysql: 192.168.0.38:3306 用户名/密码： root / Bigdata2@20



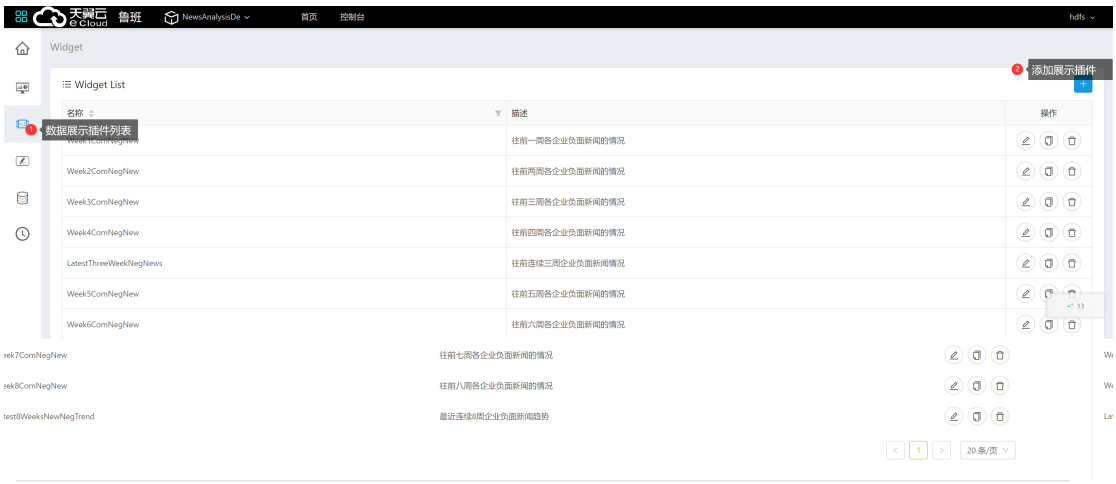
3. 选择视图列表，新增视图（如上图）

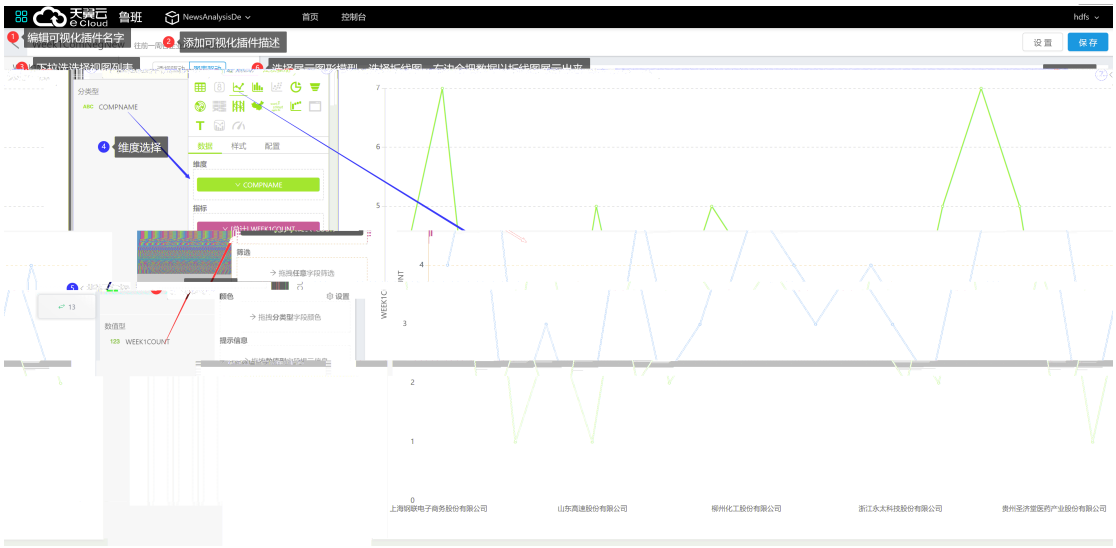


4. 点击编辑视图后，输入【视图名称】，及【SQL 语句】

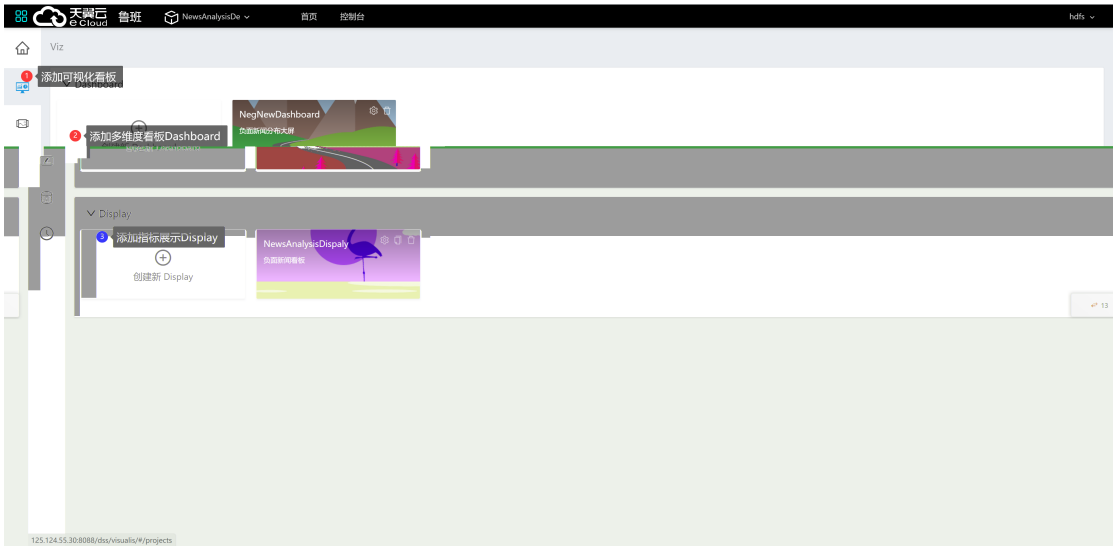
- *Week1NegCount 的 SQL 语句：select COMPNAME , WEEK1COUNT FROM news.DWA_T_NEW limit 20;*
- *8WeeksNegNewCount 的 SQL 语句：*

select COMPNAME,WEEK1COUNT ,WEEK2COUNT, WEEK3COUNT, WEEK4COUNT ,WEEK5COUNT ,WEEK6COUNT ,WEEK7COUNT ,WEEK8COUNT, (WEEK1COUNT + WEEK2COUNT+ WEEK3COUNT+ WEEK4COUNT + WEEK5COUNT + WEEK6COUNT + WEEK7COUNT + WEEK8COUNT) totalCount FROM news.DWA_T_NEW order by totalCount desc LIMIT 20;

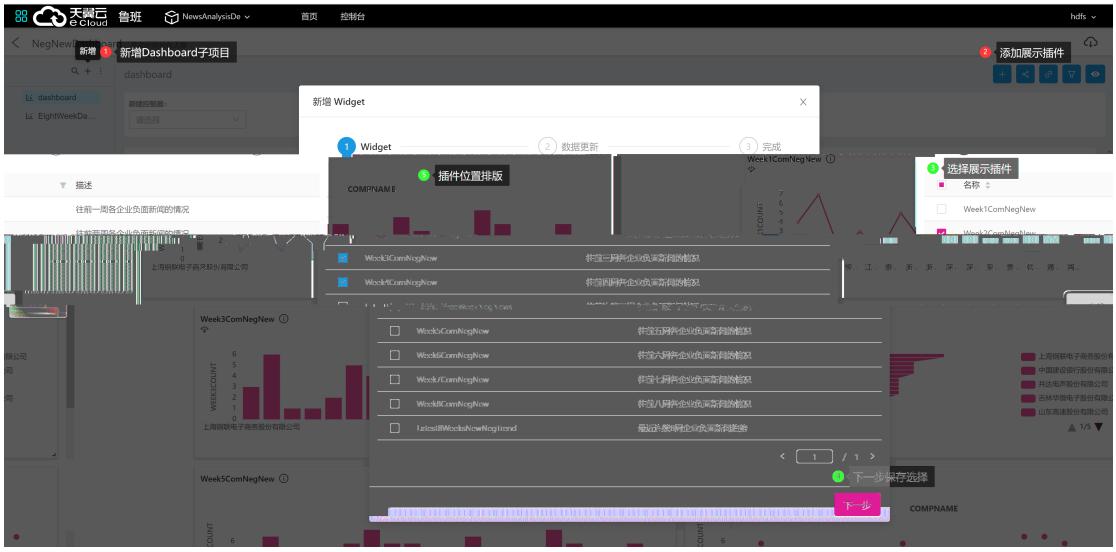


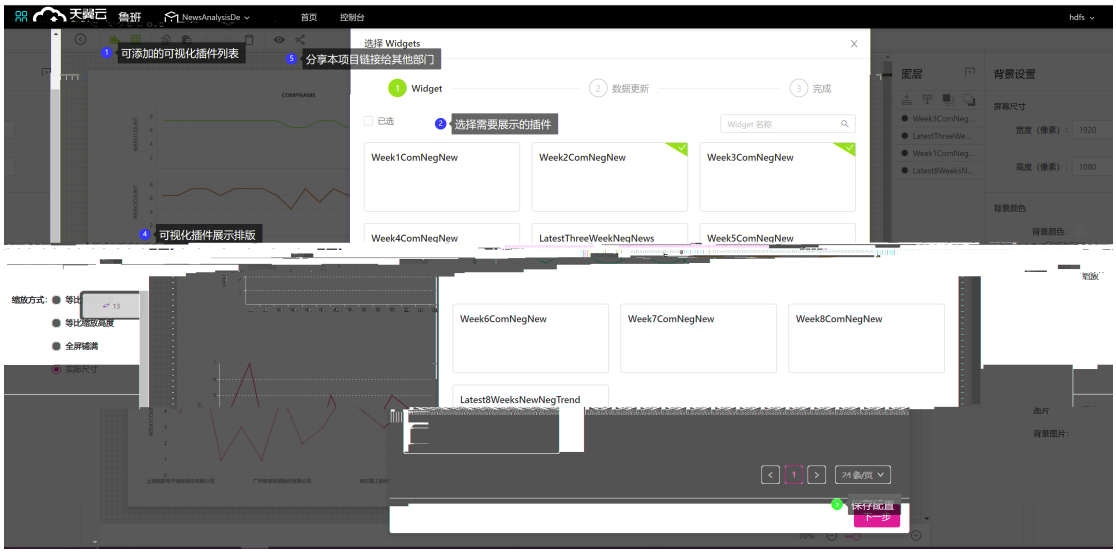


1. *点击添加展示插件，添加插件*

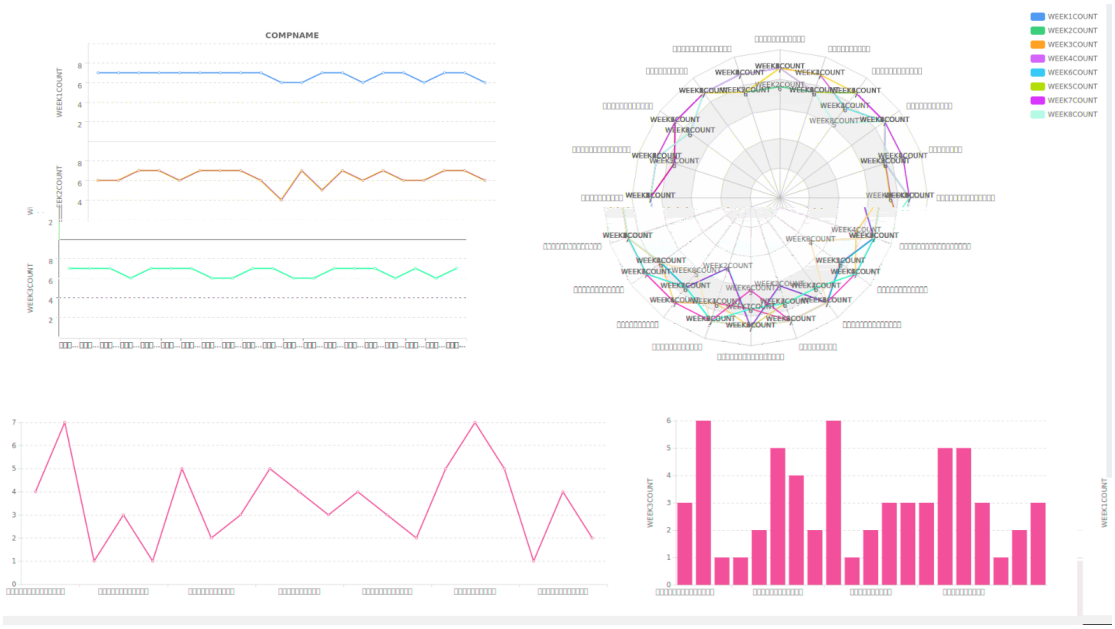


1. *添加可视化看板 Dashboard 和 display*





1. *添加可视化插件*



最终可看到如上图的分析结果

附录：新闻舆情分析案例代码

1. hive_ddl_createtable 代码

```
create table if not exists ODS_T_NEWS (
`ID` STRING comment "ID",
`TITLE` STRING comment '新闻标题',
`DATE` STRING comment '新闻日期
```

```
`WEEK2COUNT` int  comment '新闻评分正面为 1 条数' ,
`WEEK3COUNT` int  comment '新闻评分正面为 1 条数' ,
`WEEK4COUNT` int  comment '新闻评分正面为 1 条数' ,
`WEEK5COUNT` int  comment '新闻评分正面为 1 条数' ,
`WEEK6COUNT` int  comment '新闻评分正面为 1 条数' ,
`WEEK7COUNT` int  comment '新闻评分正面为 1 条数' ,
`WEEK8COUNT` int  comment '新闻评分正面为 1 条数' )
stored as ORC
location 'hdfs:///tmp/linkis/hdfs/NewAnalysis/hive/dwa_t_news_pos';
```

2. genData_1、genData_2、genData_3 代码(3 者代码相同)

```
    import java.text.SimpleDateFormat
    import java.util
    import java.util.{Calendar, Random}

    import org.apache.spark.broadcast.Broadcast
    import org.apache.spark.sql.{SQLContext, SaveMode, SparkSession}


    val sqlCtx: SQLContext   spark.sqlContext
    sqlCtx.setConf("hive.exec.dynamici.partition","true")
    sqlCtx.setConf("hive.exec.dynamic.partition.mode","nonstrict")

    val current_date   "2020-06-06"
    //数据复制倍数
  val copy_times   1

    val dtFormat   new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val date_arry   new util.ArrayList[String]();
    for(i  - 1 to 68){
      val calendar: Calendar   Calendar.getInstance()
      calendar.setTime(new SimpleDateFormat("yyyy-MM-dd").parse(current_date))
      calendar.add(Calendar.DATE,-i)
      date_arry.add(dtFormat.format(calendar.getTimeInMillis))
    }

    val schemaString   "ID,TITLE,DATE,DT,COMPNAME,NEWSCODE,SENTIMENT"

    val dtBroad: Broadcast[util.ArrayList[String]]   spark.sparkContext.broadcast(date_arry)
    val fileDF   spark.read.format("csv").option("header", true).load("hdfs:///tmp/linkis/hdfs/NewAn
alysis/sample2020.csv")   //需替换成自己上传的文件路径
```

```
    import spark.implicits._

    val ods_t_newsDF   fileDF.na.fill("unknow")
      .map(x    {
        val buffer   new StringBuffer()
        for(i  - 1 to copy_times){
          val x1   x(1).toString   new Random().nextInt()
          val index   Math.abs(new Random().nextInt(Integer.MAX_VALUE))   66
          val x2   dtBroad.value.get(index)
          var x5   x(5)
          if(x5.toString.trim.isEmpty) {
            x5   -1
          }
          buffer.append(","   x(0).toString)
          buffer.append("|"   x1)
          buffer.append("|"   x2)
          buffer.append("|"   x2.split(" ")(0))
          buffer.append("|"   x(3).toString)
          buffer.append("|"   x(4).toString)
          buffer.append("|"   x5.toString)
        }
        buffer.toString.replaceFirst(",","")
      })
      .flatMap(x    {
        val str_arr   x.split(",")
        str_arr
      })
      .filter(x    {
        val arr   x.split("\\|")
        arr.length    7 && arr.apply(3).split("-").length    3
      })
      .map(x    {
        val str_list   x.split("\\|")
        (str_list(0),str_list(1),str_list(2),str_list(3),str_list(4),str_list(5),str_list(6))
      })
      .toDF(schemaString.split(","): _*)

    //控制台输出，方便调试时观察数据
    ods_t_newsDF.show(10)

    ods_t_newsDF.write.format("hive").partitionBy("dt").mode(SaveMode.Append).saveAsTable("ODS_T_NEW
S")
```

3. NewByDay 代码

```
import org.apache.spark.sql.{SQLContext, SaveMode, SparkSession}

val sqlCtx: SQLContext   spark.sqlContext
sqlCtx.setConf("hive.exec.dynamici.partition","true")
sqlCtx.setConf("hive.exec.dynamic.partition.mode","nonstrict")
val ods_t_news_negDF   spark.sql("select DT ,COMPNAME  from ODS_T_NEWS where SENTIMENT   -1")
var schema_neg   "STATISTICDATE,COMPNAME,NEGCOUNT"
val dwt_t_news_negDF   ods_t_news_negDF.groupBy("DT", "COMPNAME").count().toDF(schema_neg.split(
","): _*)
val ods_t_news_neaDF   spark.sql("select DT ,COMPNAME  from ODS_T_NEWS where SENTIMENT   0")
var schema_nea   "STATISTICDATE,COMPNAME,NEUTRALCOUNT"
val dwt_t_news_neaDF   ods_t_news_neaDF.groupBy("DT", "COMPNAME").count().toDF(schema_nea.split(
","): _*)
val ods_t_news_posDF   spark.sql("select DT ,COMPNAME  from ODS_T_NEWS where SENTIMENT   -1")
var schema_pos   "STATISTICDATE,COMPNAME,POSCOUNT"
val dwt_t_news_posDF   ods_t_news_posDF.groupBy("DT", "COMPNAME").count().toDF(schema_pos.split(
","): _*)
val dwt_t_newsDF   dwt_t_news_negDF.join(dwt_t_news_neaDF, "STATISTICDATE,COMPNAME".split(","))
  .join(dwt_t_news_posDF, "STATISTICDATE,COMPNAME".split(","))
dwt_t_newsDF.show()
dwt_t_newsDF.write.format("hive").mode(SaveMode.Append).saveAsTable("DWS_T_NEWS")
```

4. NegNewByWeek 代码

```
import java.text.SimpleDateFormat
import java.util.Calendar

import org.apache.spark.sql.{SQLContext, SaveMode, SparkSession}


def pre_n_week(date:String,n:Int)   {
  val dateFormat   new SimpleDateFormat("yyyy-MM-dd")
  val calendar   Calendar.getInstance()
  calendar.setTime(dateFormat.parse(date))
  calendar.set(Calendar.DAY_OF_WEEK,Calendar.MONDAY)

  val monday   new SimpleDateFormat("yyyy-MM-dd").format(calendar.getTime)
  calendar.setTime(dateFormat.parse(monday))
  calendar.set(Calendar.DATE, -7* (n-1) )
```

```
      val last_sun   new SimpleDateFormat("yyyy-MM-dd").format(calendar.getTime)
      calendar.setTime(dateFormat.parse(monday))
      calendar.set(Calendar.DATE,  - 7*n   1)
      val last_mon   new SimpleDateFormat("yyyy-MM-dd").format(calendar.getTime)
      (last_mon,last_sun)
    }

    val date_str   "2020-06-06"

    var week_1   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_news where STATISTICDATE    '${
pre_n_week(date_str,1)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,1)._2}'  group by COMPNAME
 "
    val week_1DF   spark.sql(week_1).toDF("COMPNAME","WEEK1COUNT")

    var week_2   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_news where STATISTICDATE    '${
pre_n_week(date_str,2)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,2)._2}'  group by COMPNAME
 "
    val week_2DF   spark.sql(week_2).toDF("COMPNAME","WEEK2COUNT")

    var week_3   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_NEWS where STATISTICDATE    '${
pre_n_week(date_str,3)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,3)._2}'  group by COMPNAME
 "
    val week_3DF   spark.sql(week_3).toDF("COMPNAME","WEEK3COUNT")

    var week_4   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_NEWS where STATISTICDATE    '${
pre_n_week(date_str,4)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,4)._2}'  group by COMPNAME
 "
    val week_4DF   spark.sql(week_4).toDF("COMPNAME","WEEK4COUNT")

    var week_5   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_NEWS where STATISTICDATE    '${
pre_n_week(date_str,5)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,5)._2}'  group by COMPNAME
 "
    val week_5DF   spark.sql(week_5).toDF("COMPNAME","WEEK5COUNT")

    var week_6   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_NEWS where STATISTICDATE    '${
pre_n_week(date_str,6)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,6)._2}'  group by COMPNAME
 "
    val week_6DF   spark.sql(week_6).toDF("COMPNAME","WEEK6COUNT")

    var week_7   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_NEWS where STATISTICDATE    '${
pre_n_week(date_str,7)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,7)._2}'  group by COMPNAME
 "
    val week_7DF   spark.sql(week_7).toDF("COMPNAME","WEEK7COUNT")
```

```
    var week_8   s"select COMPNAME,  count(1) WEEK1COUNT from  DWS_T_NEWS where STATISTICDATE    '${
pre_n_week(date_str,8)._1}'  and  STATISTICDATE    '${pre_n_week(date_str,8)._2}'  group by COMPNAME
 "
    val week_8DF   spark.sql(week_8).toDF("COMPNAME","WEEK8COUNT")

    val schema   "STATISTICDATE,COMPNAME,WEEK1COUNT,WEEK2COUNT,WEEK3COUNT,WEEK4COUNT,WEEK5COUNT,WEEK
6COUNT,WEEK7COUNT,WEEK8COUNT"


    val dt_broad   spark.sparkContext.broadcast(date_str)

    import spark.implicits._

  val joinCol   Seq("COMPNAME")

     val dwa_t_news   week_1DF
      .join(week_2DF, joinCol,"inner")
      .join(week_3DF, joinCol,"inner")
      .join(week_4DF, joinCol,"inner")
      .join(week_5DF, joinCol,"inner")
      .join(week_6DF, joinCol,"inner")
      .join(week_7DF, joinCol,"inner")
      .join(week_8DF, joinCol,"inner")
      .map(x    {
        (dt_broad.value,x(0).toString,x(1).toString.toInt,x(2).toString.toInt,x(3).toString.toInt,
          x(4).toString.toInt,x(5).toString.toInt,x(6).toString.toInt,x(7).toString.toInt,x(8).toStr
ing.toInt)
      })
      .toDF(schema.split(","):_*)

    dwa_t_news.show(10)

  dwa_t_news.write.format("hive").mode(SaveMode.Append).saveAsTable("DWA_T_NEW_NEGATIVE")
```

5. hiveToMysql 代码：

```
    import org.apache.spark.sql.{SaveMode, SparkSession}

    val date_str   "2020-06-06"
    val title   "STATISTICDATE,COMPNAME,WEEK1COUNT ,WEEK2COUNT, WEEK3COUNT, WEEK4COUNT ,WEEK5COUNT ,
WEEK6COUNT ,WEEK7COUNT ,WEEK8COUNT"

     val loadDateFromHive   s"select STATISTICDATE,COMPNAME,WEEK1COUNT ,WEEK2COUNT, WEEK3COUNT, WEEK
4COUNT ,WEEK5COUNT ,WEEK6COUNT ,WEEK7COUNT ,WEEK8COUNT from DWA_T_NEW_NEGATIVE  "
```

```
    import spark.implicits._
    val hiveDF   spark.sql(loadDateFromHive)

    hiveDF .toDF(title.split(","): _*).show(10)

    val url    "jdbc:mysql://1 2.168.0.38:3306/news?useUnicode true&characterEncoding gbk&autoReconn
ect true&failOverReadOnly false"
    val driver   "com.mysql.cj.jdbc.Driver"
    val username   "root"
    val password   "Bigdata2@20"
    val tablename   "DWA_T_NEW"

    hiveDF.write
      .format("jdbc")
      .option("driver",driver)
      .option("url",url)
      .option("user",username)
      .option("password",password)
      .option("dbtable",tablename)
      .option("numPartitions","50")
      .option("batchsize","500")
      .mode(SaveMode.Append)
      .save()
```

6. *mysql_ddl* **代码：**

```
create database if not exists news;
drop table if exists news.DWA_T_NEW;
create table if not exists news.DWA_T_NEW (
`id` int not null  AUTO_INCREMENT comment '主键',
`STATISTICDATE` varchar(32) comment '统计日期',
`COMPNAME` Text not null comment '公司名字',
`WEEK1COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK2COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK3COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK4COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK5COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK6COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK7COUNT` int  comment '新闻评分正面为-1 条数' ,
`WEEK8COUNT` int  comment '新闻评分正面为-1 条数' ,
 constraint pk_news primary key(id)
) ENGINE InnoDB AUTO_INCREMENT 10000 DEFAULT CHARSET utf8;
```