Solr集群版部署文档

目录

[Solr集群版部署文档 1](#_Toc505342363)

[1 前言 1](#_Toc505342364)

[2 SOLR集群安装部署 1](#_Toc505342365)

[2.1 部署安装软件 1](#_Toc505342366)

[2.1.1 第一步: 将安装包上传到服务器 1](#_Toc505342367)

[2.1.2 第二步：将安装包解压到系统根目录下 1](#_Toc505342368)

[2.2 第三步：修改配置文件 2](#_Toc505342369)

[2.2.1 修改solr.xml配置文件 2](#_Toc505342370)

[2.2.2 修改JVM堆栈内存及Zookeeper连接地址 2](#_Toc505342371)

[2.2.3 查看Zookeeper连接地址： 3](#_Toc505342372)

[2.2.4 第四步：将配置好的应用分发到其他的服务器 5](#_Toc505342373)

[2.2.5 第五步：修改被分发的服务器solr.xml配置文件的主机名 5](#_Toc505342374)

[2.2.6 第六步：将Solr配置文件上传到Zookeeper 5](#_Toc505342375)

[2.2.7 第七步：启动Solr 5](#_Toc505342376)

[2.2.8 浏览器查看是否启动成功 6](#_Toc505342377)

[3 部署定时脚本 7](#_Toc505342378)

[3.1 第一步：修改脚本配置 7](#_Toc505342379)

[3.1.1 上传脚本到Linux 7](#_Toc505342380)

[3.1.2 修改脚本配置 7](#_Toc505342381)

[3.2 第二步：手动执行脚本（必须） 8](#_Toc505342382)

[3.3 第三步：脚本添加定时任务 9](#_Toc505342383)

[4 部署易搜 10](#_Toc505342384)

[4.1 部署yisou 10](#_Toc505342385)

[4.1.1 上传tomcat-yisou.zip到Linux 10](#_Toc505342386)

[4.1.2 解压到/solrCloud目录下 10](#_Toc505342387)

[4.1.3 修改yisou配置文件 10](#_Toc505342388)

[4.2 启动tomcat-yisou 11](#_Toc505342389)

[5 创建hbase表 13](#_Toc505342390)

[5.1 第一步：创建HBase表 13](#_Toc505342391)

[5.2 第二步：创建Hive映射表 14](#_Toc505342392)

[5.3 第三步:初始化Impala连接 14](#_Toc505342393)

[6 数据导入 16](#_Toc505342394)

[6.1 部署前的准备： 16](#_Toc505342395)

[6.1.1 压缩包说明 16](#_Toc505342396)

[6.1.2 修改配置文件的一般步骤 16](#_Toc505342397)

[6.1.3 第二步：修改application.properties配置文件 18](#_Toc505342398)

[6.1.4 第三步：替换core-site.xml配置文件 19](#_Toc505342399)

[6.1.5 第四步：修改columns.json配置文件 19](#_Toc505342400)

[6.1.6 第五步：ClouderaManager集群关闭HDFS服务的权限检查 20](#_Toc505342401)

[6.2 部署应用 23](#_Toc505342402)

[6.2.1 第一步：创建目录 23](#_Toc505342403)

[6.2.2 第二步：解压压缩包 23](#_Toc505342404)

[6.2.3 第三步：替换修改过的jar包 23](#_Toc505342405)

[6.2.4 第四步：从Oracle导入历史数据 23](#_Toc505342406)

[6.2.5 第五步：从BCP导入实时数据 23](#_Toc505342407)

[6.2.6 第六步：验证数据是否导入到Solr 24](#_Toc505342408)

[6.2.7 第七步：验证数据是否导入到HBase 26](#_Toc505342409)

[7 附件1：Hive映射HBase外部表sql 28](#_Toc505342410)

# 前言

搭建之前先确认CHD5的环境是正常的

# SOLR集群安装部署

## 部署安装软件

### 第一步: 将安装包上传到服务器

### 第二步：将安装包解压到系统根目录下

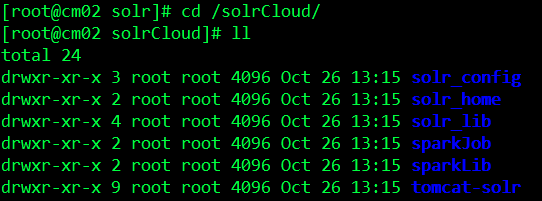
1. 给安装包添加可执行权限

chmod u+x solrCloud.zip

1. 解压到根目录

unzip solrCloud.zip -d /

解压后的solrCloud目录下的文件为:



文件说明:

solr\_config ：Solr配置文件目录

solr\_home ：Solr数据主目录

solr\_lib ：Solr依赖jar包目录

sparkJob ：Spark任务目录

sparkLib ：Spark依赖的jar包目录

tomcat-solr ：Solr应用

## 第三步：修改配置文件

### 修改solr.xml配置文件

vim /solrCloud/solr\_home/solr.xml

1. 修改host标签为你的主机名

<str name="host">${host:cm02.spark.com}</str>



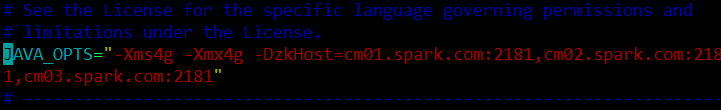
### 修改JVM堆栈内存及Zookeeper连接地址

执行编辑命令：

vim /solrCloud/tomcat-solr/bin/catalina.sh

1. 修改JVM的堆栈内存，根据情况修改内存大小，这里默认是4G，看情况修改Solr的内存，生产上至少调到10G，然后看Solr的运行情况，如果查询慢的话可以往上加，最大不要加到总内存的一半
2. 修改Zookeeper的连接地址，只需要修改为你自己的主机名就可以了，端口号不变

JAVA\_OPTS="-Xms4g -Xmx4g -DzkHost=cm01.spark.com:2181,cm02.spark.com:2181,cm03.spark.com:2181"



### 查看Zookeeper连接地址：

Cloudera Manager管理界面查看Zookeeper连接地址：

浏览器打开Cloudera Manager管理界面：<http://cm02.spark.com:7180>  






将这几个主机名以逗号分隔就是Zookeeper的连接地址（还有一种写法是主机名加上端口号，两种方式都可以）示例如下：

方式1：

cm01.spark.com,cm02.spark.com,cm03.spark.com

方式2：

cm01.spark.com:2181,cm02.spark.com:2181,cm03.spark.com:2181

### 第四步：将配置好的应用分发到其他的服务器

执行分发命令：

scp -r /solrCloud/ root@cm01:/

每一个服务器各执行一次，cm01是主机名为cm01.spark.com的服务器的简写，是在hosts文件里配置的，分发到不同的服务器注意修改“cm01”为对应的服务器，另外，自己的服务器就不用分发了。

分发后在各个服务器上查看是否分发成功

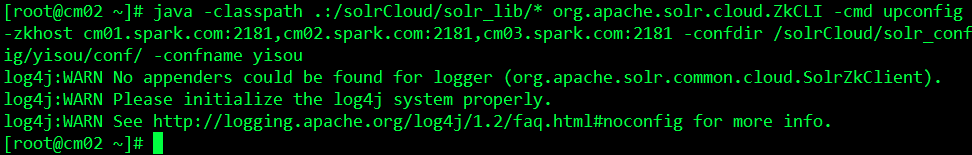
### 第五步：修改被分发的服务器solr.xml配置文件的主机名

参考第三步

### 第六步：将Solr配置文件上传到Zookeeper

注意：执行下面的命令的时候把Zookeeper的连接地址改为你自己的主机名

java -classpath .:/solrCloud/solr\_lib/\* org.apache.solr.cloud.ZkCLI -cmd upconfig -zkhost cm01.spark.com:2181,cm02.spark.com:2181,cm03.spark.com:2181 -confdir /solrCloud/solr\_config/yisou/conf/ -confname yisou



### 第七步：启动Solr

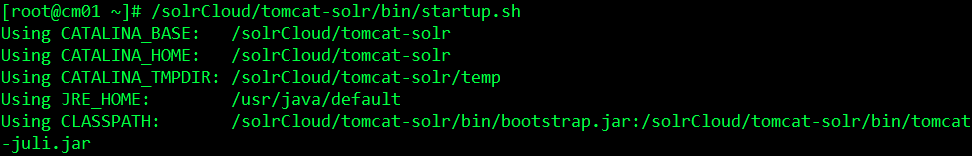
1. 给执行脚本添加可执行权限

chmod u+x /solrCloud/tomcat-solr/bin/\*.sh

每一台服务器都执行一下

1. 执行启动命令

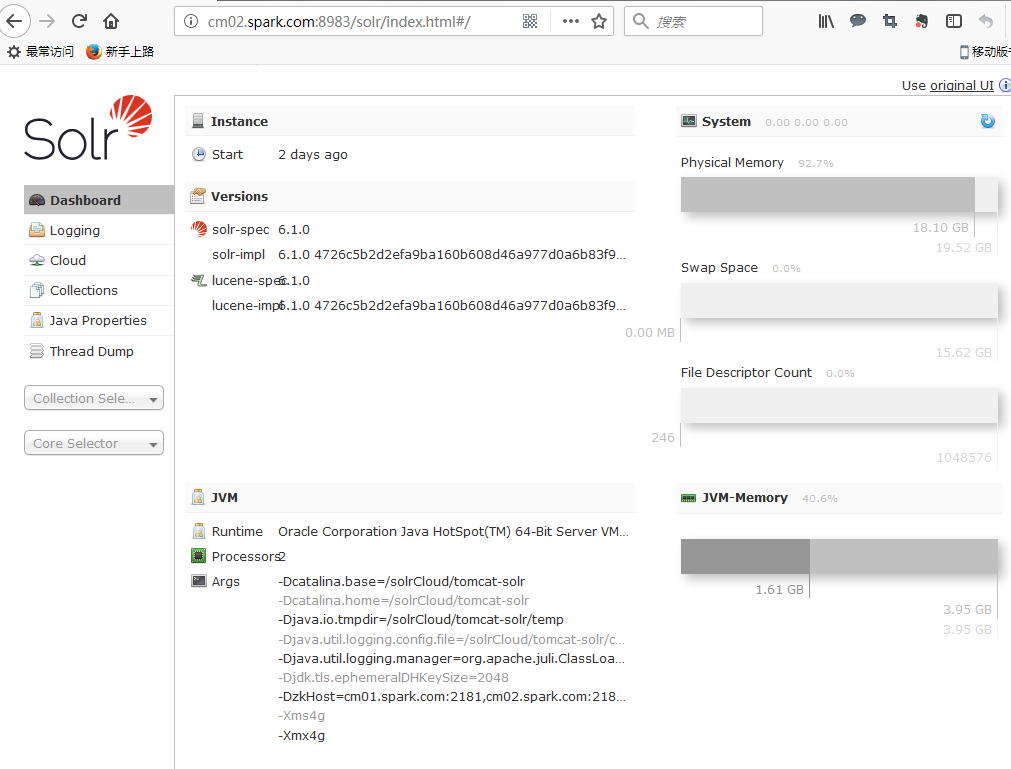
/solrCloud/tomcat-solr/bin/startup.sh



每一台服务器都执行一下

### 浏览器查看是否启动成功

<http://cm02.spark.com:8983/solr/index.html>



# 部署定时脚本

脚本名字为create\_solr\_collection.py

## 第一步：修改脚本配置

### 上传脚本到Linux

在/solrCloud目录下创建脚本目录

mkdir -p /solrCloud/script

将create\_solr\_collection.py脚本上传到该目录下

### 修改脚本配置

vim /solrCloud/script/create\_solr\_collection.py

程序的入口的脚本的最下面

修改初始化参数

init("http://cm02.spark.com", 8983, 3, 2)

第一个参数：Solr应用所在的任意一台服务器的主机名,

第二个参数：Solr应用程序的端口号，与《Solr集群安装部署》部分的端口号一样

第三个参数：Solr部署到多少台服务器上

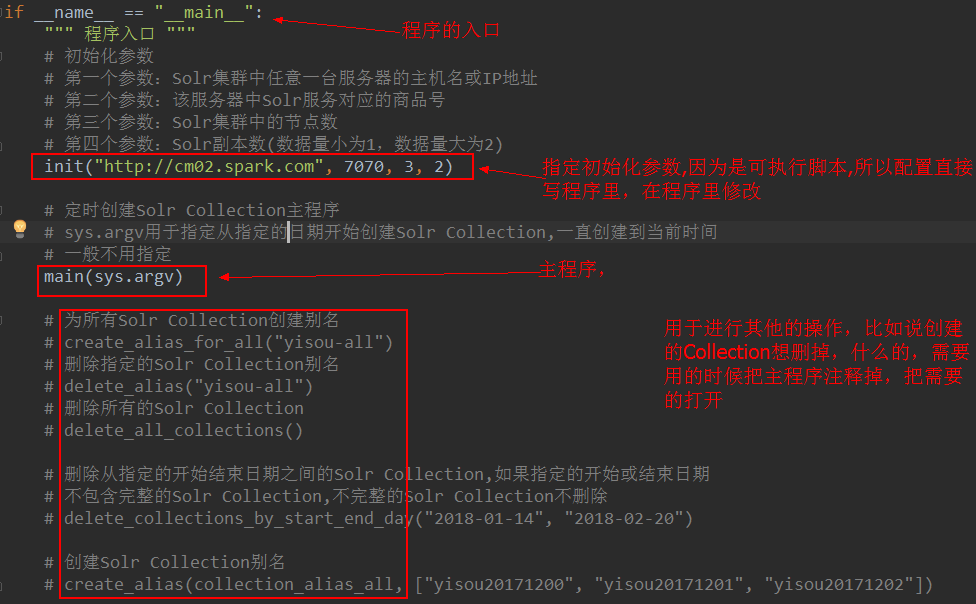
第四个参数：Solr副本数，数据量小为1，数据量大为2，北京三期项目数据量有100亿以上，为2，建议60亿以上就设为2

脚本的主程序是

main(sys.argv)

除了主程序外还有好几个被注释掉的方法，用以进行其他的操作

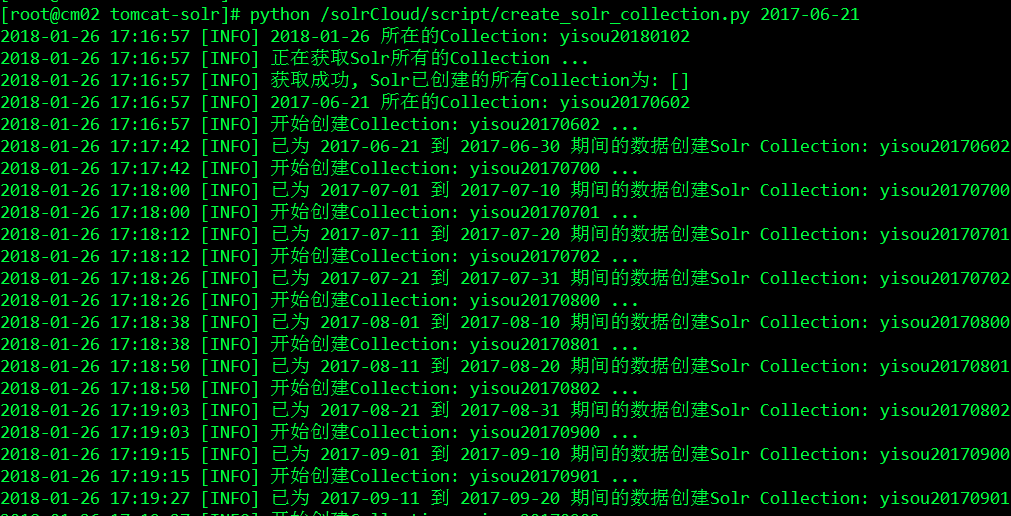
程序写得很详细，不懂的话可以看



## 第二步：手动执行脚本（必须）

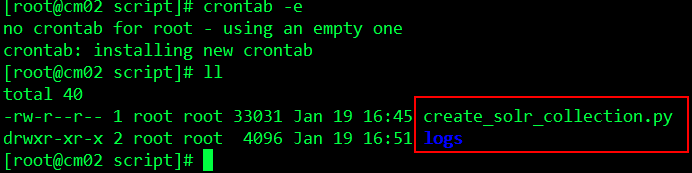
**python** **/**solrCloud**/**script**/**create\_solr\_collection.py 2017**-**06**-**21

确定要导入的最早的数据是什么时候，根据是capture\_time，比如说是2017-06-21



首次运行生成当前需要用的Collection，后面定时执行就不用管了

除了控制台打印的，还会在可执行脚本所在的目录下生成logs目录用以存放执行日志



## 第三步：脚本添加定时任务

添加定时任务：

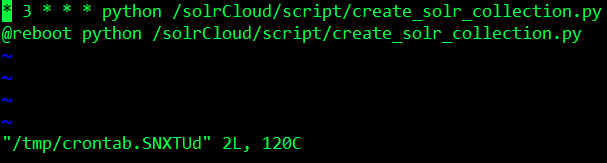
**crontab** –e

添加以下内容：

## 添加以下内容

**\*** 3 **\*** **\*** **\*** python **/**solrCloud**/**script**/**create\_solr\_collection.py

**@**reboot python **/**solrCloud**/**script**/**create\_solr\_collection.py



# 部署易搜

## 部署yisou

仅在一台服务器上部署就可以了，我这里部署在cm02.spark.com服务器上

### 上传tomcat-yisou.zip到Linux

### 解压到/solrCloud目录下

添加可执行权限

chmod u+x tomcat-yisou.zip

解压

unzip tomcat-yisou.zip –d /solrCloud/

### 修改yisou配置文件

vim /solrCloud/tomcat-yisou/webapps/yisou/WEB-INF/classes/solr.properties

# Java API 通过HTTP Client连接Solr

solr.url=http://cm02.spark.com:8983/solr/yisou-all

solr.maxRetries=1

solr.connectionTimeout=1000

# JavaAPI通过Zookeeper连接Solr

solrCloud.zkHost=cm01.spark.com:2181,cm02.spark.com:2181,cm03.spark.com:2181

# Solr Collection名

solrCloud.defaultCollection=yisou-all

只需要修改solrCloud.zkHost和solr.url即可，红色的标识是需要修改的部分。

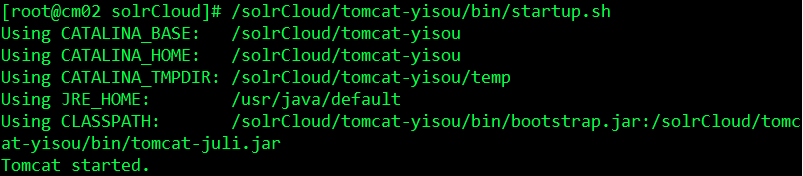
将solr.url值里的cm02.spark.com修改成你Solr集群中中任意一台服务器的主机名或者IP

将solrCloud.zkHost值里的主机名修改成Solr集群中所有的主机名



## 启动tomcat-yisou

/solrCloud/tomcat-yisou/bin/startup.sh



yisou在这里只是一个搜索引擎，网页无法访问，只有把前端应用部署上去的时候才能看到

前端调用需要的配置:

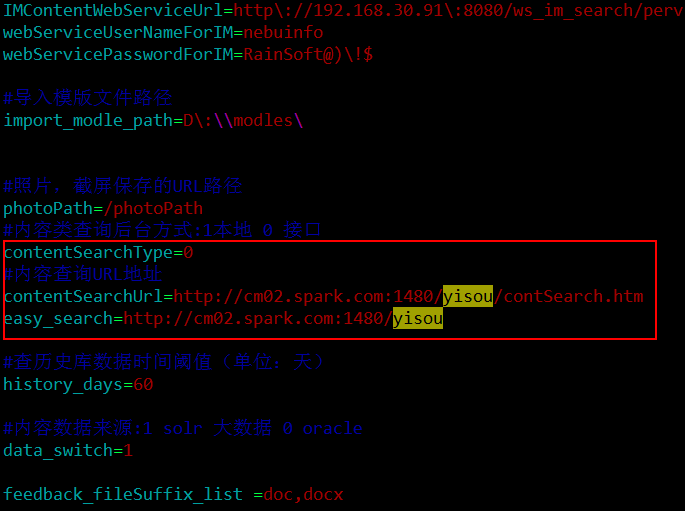
contentSearchUrl=http://cm02.spark.com:1480/yisou/contSearch.htm

easy\_search=http://cm02.spark.com:1480/yisou

一般是在

${tomcat\_home}webapps/case/WEB-INF/classes/config/sysconfig.properties

里配置的



# 创建hbase表

## 第一步：创建HBase表

在大数据集群任意服务器上

hbase shell

create 'H\_REG\_CONTENT\_EMAIL','INFO'

create 'H\_REG\_CONTENT\_WEIBO','INFO'

create 'H\_REG\_CONTENT\_BBS','INFO'

create 'H\_REG\_CONTENT\_SEARCH','INFO'

create 'H\_REG\_VID\_INFO','INFO'

create 'H\_REG\_REALID\_INFO','INFO'

create 'H\_SERVICE\_INFO','INFO'

create 'H\_REG\_CONTENT\_IM\_CHAT','INFO'

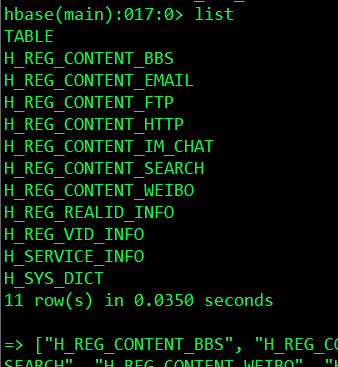
create 'H\_REG\_CONTENT\_HTTP','INFO'

create 'H\_REG\_CONTENT\_FTP','INFO'

create 'H\_SYS\_DICT', 'INFO'

查看是否创建成功：

list



## 第二步：创建Hive映射表

在大数据集群的任意一台服务器输入命令：

hive

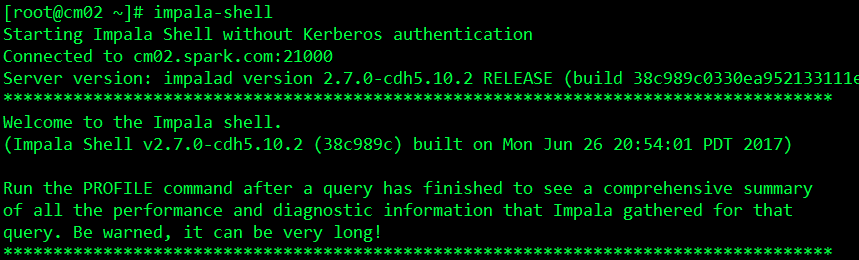
要创建的数据仓库实例及表参见：

附件1：Hive映射HBase外部表sql

## 第三步:初始化Impala连接

在大数据集群的任意一台服务器输入命令：

Impala-shell



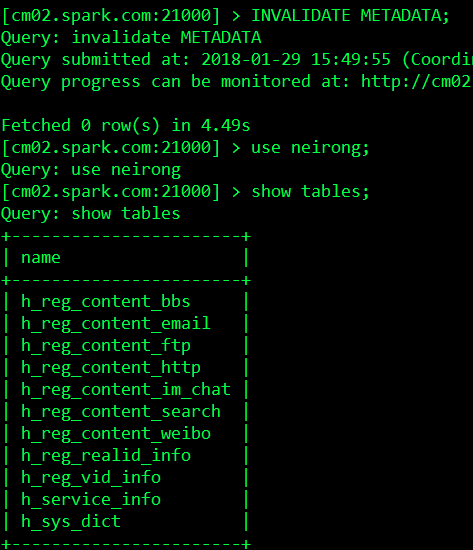
同步Hive元数据

INVALIDATE METADATA;

查看是否同步成功：

use neirong;

show tables;



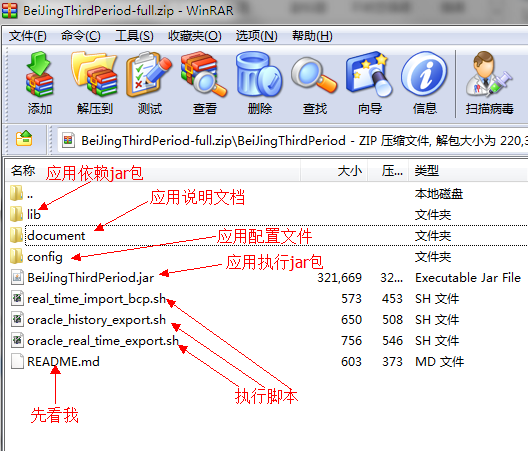
# 数据导入

压缩包名：BeiJingThirdPeriod-full.zip

## 部署前的准备：

### 压缩包说明

项目压缩包名：BeiJingThirdPeriod-full.zip



可执行脚本说明：

oracle\_real\_time\_export.sh Oracle数据实时导入脚本

real\_time\_import\_bcp.sh Bcp文件实时导入脚本

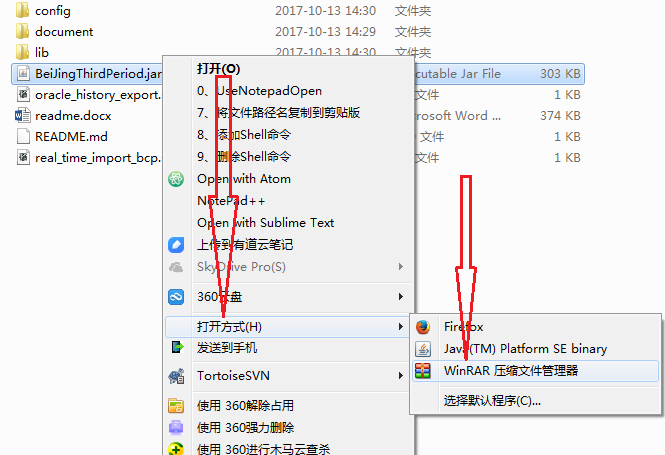
oracle\_history\_export.sh Oracle历史数据导入脚本

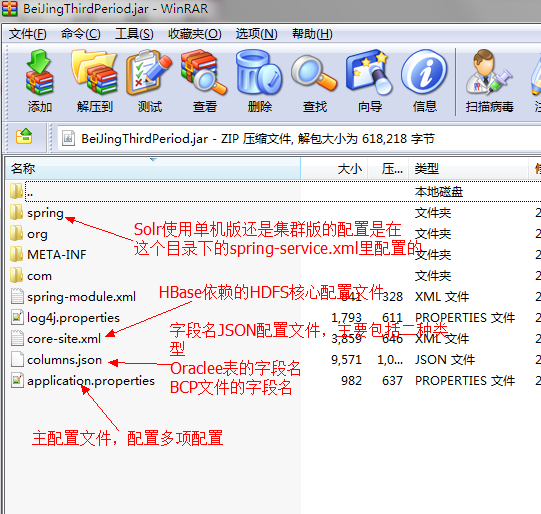
这里使用红色标识的两个

### 修改配置文件的一般步骤

将jar包从zip包里拖出来

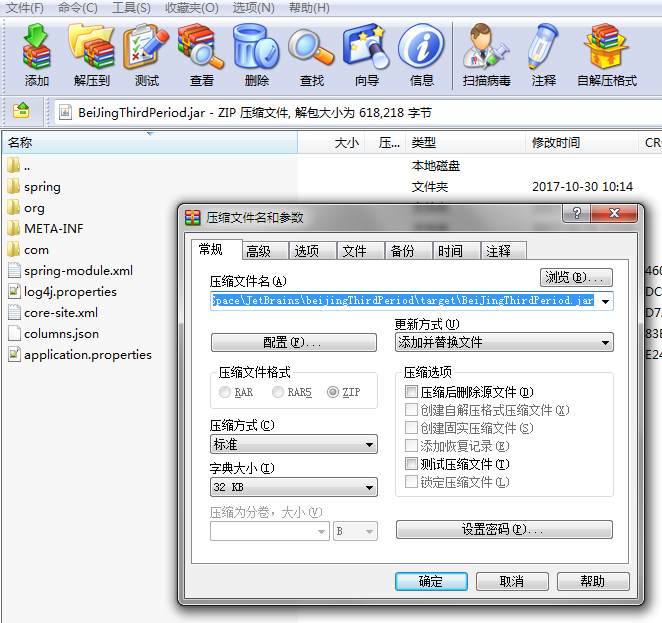
**第一步：用解压缩软件打开Jar包**





**由于要修改好几个配置文件，这里先说下修改的步骤**

1. 保存修改的配置文件（具体哪个配置文件怎么修改下面讲）
2. 把修改后的配置文件拖回BeiJingThirdPeriod.jar压缩包中覆盖原来的配置文件
3. 拖回压缩包后会提示是否覆盖，点击确定，提示框如下:

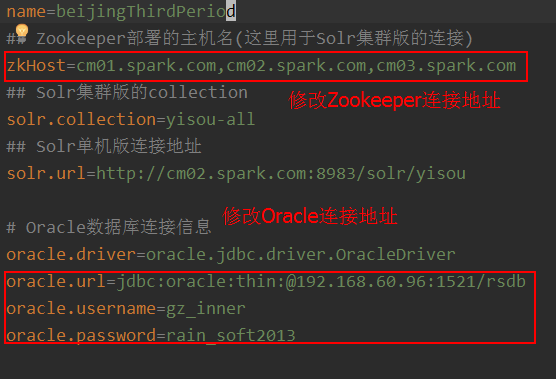


### 第二步：修改application.properties配置文件

1. 将application.properties从jar包中拖出来

2. 编辑application.properties配置文件

application.properties配置文件截图



需要修改两处配置：

第一处：修改Oracle连接信息

第二处：修改Zookeeper的连接地址，

3. 保存修改

4. 将修改后的配置文件拖回jar包中覆盖原来的文件

### 第三步：替换core-site.xml配置文件

大数据集群上此文件夹的位置一般为:

/etc/hadoop/conf /core-site.xml

每一个服务器上的这个目录下都有

### 第四步：修改columns.json配置文件

操作参考上一步的操作

Columns.json配置文件是基于管综的，如果是Isec的，一定要核对一下，另外有Isec的BCP文件转管综的BCP文件的中间组件，这个可以问下田彦军老师

此文档写于2018年1月26日，如果在这之后Oracle表结构没有变，并且BCP文件结构也没有变则不用修改

如果修改请继续

这个配置文件是配置Oracle表、BCP文件、Solr字段

这是一个json文件

命名格式是：

关键字:[字段1，字段2，字段3.。。。]

关键字中：

“bcp”开头的是BCP文件字段结构

“filter\_bcp”开头的表示BCP文件数据导入的数据需要对这些字段进行过滤

“reg”开关的是Oracle里对应表的结构，包括service\_info

“solr”开头的是Solr配置的配置文件

“solr\_date\_field\_type”表示Solr里日期类型的字段

“solr\_exclusion\_fields”表示不需要导入到Solr里的字段

根据情况修改对应的配置

### 第五步：ClouderaManager集群关闭HDFS服务的权限检查

1. 登陆到Cloudera Manager管理界面

<http://cm02.spark.com:7180>

1. 进入HDFS配置界面







1. 重启使配置生效



## 部署应用

将BeiJingThirdPeriod-full.zip上传到服务器

### 第一步：创建目录

**mkdir** -p **/**opt**/**modules

### 第二步：解压压缩包

**unzip** BeiJingThirdPeriod-full.zip -d **/**opt**/**modules/

### 第三步：替换修改过的jar包

将你在Windows平台下修改过配置文件的jar包替换刚刚解压的jar包

### 第四步：从Oracle导入历史数据

大数据版一般是先从Oracle导入数据，然后再从BCP文件导入数据

如果不需要从Oracle导入数据的话这一步可以直接跳过

后台启动命令

**nohup** sh oracle\_history\_export.sh start "${开始时间}" **&**

开始时间是根据Bcp文件最早生成的时间来确定的，比如说最早的Bcp文件是从2017-11-01 00:00:00开始生成的，截止时间就应该是:” 2017-11-01 00:00:00”。注意一定要加上英文的双引号,不然会被认为是两个参数

这是后台任务，如果需要查看日志的话执行命令:

**tail -f** /opt/modules/BeiJingThirdPeriod/log/log.log

### 第五步：从BCP导入实时数据

执行Oracle实时任务脚本:

后台启动命令

**nohup** sh oracle\_real\_time\_export.sh start **&**

如果需要停止的话：

**sh** oracle\_real\_time\_export.sh stop

### 第六步：验证数据是否导入到Solr

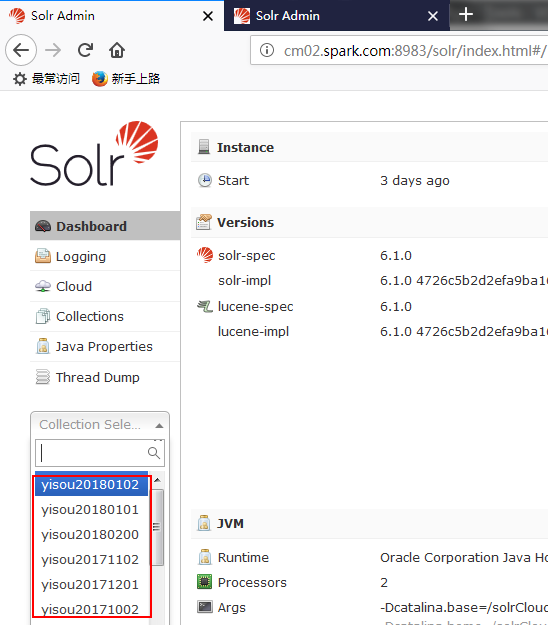
**Solr集合的划分：**

Solr按时间10天划分一个集合，比如说2017年12月，前10天是一个集合命名为yisou20171200，中间10天是一个集合命名为yisou20171201，剩下的时间是一个集合,命名为yisou20171202。以此类推

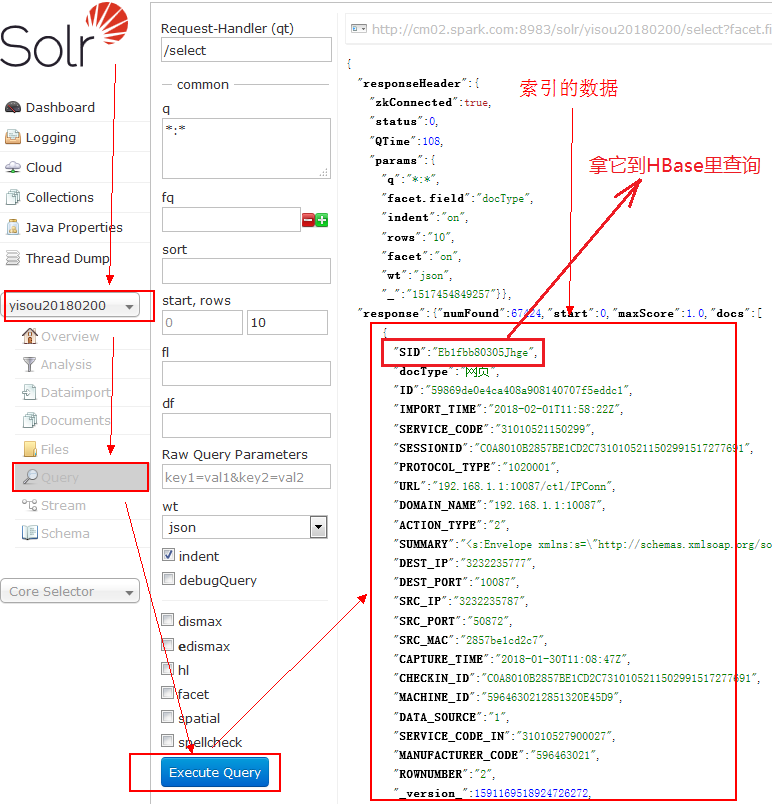
数据是基于时间（Oracle是基于capture\_time，BCP文件中基于import\_time，这两个时间字段相差不大）索引到Solr里不同的集合的，去日志文件里看当时数据是索引到哪个集合，然后到Solr管理界面对应的集合查看数据是否导入。

浏览器打开Solr管理界面：

<http://cm02.spark.com:8983/solr/index.html>



根据日志里数据写入的Solr集合，选择对应的集合，如数据采集的时候是2018-02-01 12:00:00，对应的集合是yisou20180200



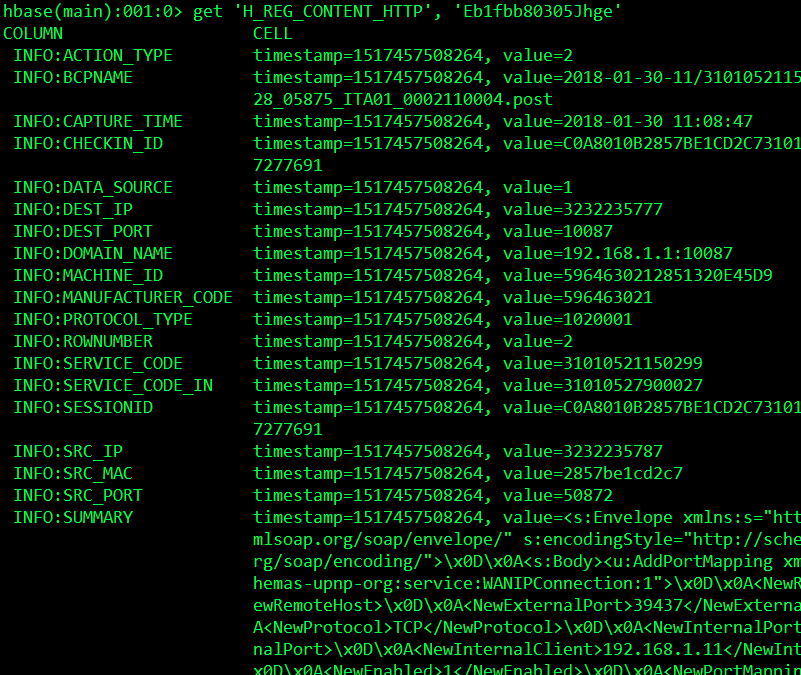
### 第七步：验证数据是否导入到HBase

在大数据集群里的任意一台服务器上打开HBase命令行：

hbase shell

然后拿Solr里的数据与HBase的数据进行对比，Solr里docType字段值是网页就到HBase查网页表的数据，把SID字段值传给HBase

get 'H\_REG\_CONTENT\_HTTP', 'Eb1fbb80305Jhge'



查看数据是否一致，需要指出的是Solr与HBase日期的格式不太一致，但是值是一样的。

# 附件1：Hive映射HBase外部表sql

*--创建数据库***create** database neirong;

*-- BBS论坛表***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_BBS (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 ROOM\_ID string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 ACCOUNT string,  
 PASSWD string,  
 URL string,  
 DOMAIN\_NAME string,  
 REF\_URL string,  
 REF\_DOMAIN string,  
 POSTING\_ID string,  
 TITLE string,  
 AUTHOR string,  
 SOURCE string,  
 ACTION\_TYPE string,  
 SUMMARY string,  
 FILE\_PATH string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 CHECKIN\_ID string,  
 DATA\_SOURCE string,  
 MACHINE\_ID string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_IN string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string  
)  
ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key, INFO:SESSIONID ,INFO:SERVICE\_CODE,INFO:ROOM\_ID,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE ,INFO:USER\_NAME,INFO:PROTOCOL\_TYPE ,INFO:ACCOUNT,INFO:PASSWD,INFO:URL,INFO:DOMAIN\_NAME,INFO:REF\_URL,INFO:REF\_DOMAIN,INFO:POSTING\_ID,INFO:TITLE,INFO:AUTHOR,INFO:SOURCE,INFO:ACTION\_TYPE,INFO:SUMMARY,INFO:FILE\_PATH,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:CHECKIN\_ID,INFO:DATA\_SOURCE,INFO:MACHINE\_ID,INFO:MANUFACTURER\_CODE,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:IMPORT\_TIME,INFO:SERVICE\_CODE\_IN,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_BBS");

*--邮件表***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_EMAIL (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 ROOM\_ID string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 ACCOUNT string,  
 PASSWD string,  
 MAILID string,  
 SEND\_TIME string,  
 MAIL\_FROM string,  
 MAIL\_TO string,  
 CC string,  
 BCC string,  
 SUBJECT string,  
 SUMMARY string,  
 ATTACHMENT string,  
 FILE\_PATH string,  
 ACTION\_TYPE string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 CHECKIN\_ID string,  
 DATA\_SOURCE string,  
 MACHINE\_ID string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_IN string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string  
 )  
 ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SESSIONID,INFO:SERVICE\_CODE,INFO:ROOM\_ID,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE,INFO:USER\_NAME,INFO:PROTOCOL\_TYPE,INFO:ACCOUNT,INFO:PASSWD,INFO:MAILID,INFO:SEND\_TIME,INFO:MAIL\_FROM,INFO:MAIL\_TO,INFO:CC,INFO:BCC,INFO:SUBJECT,INFO:SUMMARY,INFO:ATTACHMENT,INFO:FILE\_PATH,INFO:ACTION\_TYPE,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:CHECKIN\_ID,INFO:DATA\_SOURCE,INFO:MACHINE\_ID,INFO:MANUFACTURER\_CODE,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:IMPORT\_TIME,INFO:SERVICE\_CODE\_IN,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_EMAIL");  
  
*--FTP文件表***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_FTP (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 ACCOUNT string,  
 PASSWD string,  
 FILE\_NAME string,  
 FILE\_PATH string,  
 ACTION\_TYPE string,  
 IS\_COMPLETED string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 ROOM\_ID string,  
 CHECKIN\_ID string,  
 MACHINE\_ID string,  
 DATA\_SOURCE string,  
 FILE\_SIZE string,  
 FILE\_URL string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_IN string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string  
)  
 ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SESSIONID ,INFO:SERVICE\_CODE ,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE, INFO:USER\_NAME ,INFO:PROTOCOL\_TYPE,INFO:ACCOUNT,INFO:PASSWD ,INFO:FILE\_NAME,INFO:FILE\_PATH,INFO:ACTION\_TYPE,INFO:IS\_COMPLETED,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:ROOM\_ID,INFO:CHECKIN\_ID,INFO:MACHINE\_ID,INFO:DATA\_SOURCE,INFO:FILE\_SIZE,INFO:FILE\_URL,INFO:MANUFACTURER\_CODE,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:IMPORT\_TIME,INFO:SERVICE\_CODE\_IN,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_FTP");  
  
*--HTTP网页表***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_HTTP (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 URL string,  
 DOMAIN\_NAME string,  
 REF\_URL string,  
 REF\_DOMAIN string,  
 ACTION\_TYPE string,  
 SUBJECT string,  
 SUMMARY string,  
 COOKIE\_PATH string,  
 UPLOAD\_FILE string,  
 DOWNLOAD\_FILE string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 ROOM\_ID string,  
 CHECKIN\_ID string,  
 MACHINE\_ID string,  
 DATA\_SOURCE string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_IN string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string  
)  
 ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SESSIONID,INFO:SERVICE\_CODE,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE,INFO:USER\_NAME,INFO:PROTOCOL\_TYPE,INFO:URL,INFO:DOMAIN\_NAME,INFO:REF\_URL,INFO:REF\_DOMAIN,INFO:ACTION\_TYPE,INFO:SUBJECT,INFO:SUMMARY,INFO:COOKIE\_PATH,INFO:UPLOAD\_FILE,INFO:DOWNLOAD\_FILE,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:ROOM\_ID,INFO:CHECKIN\_ID,INFO:MACHINE\_ID,INFO:DATA\_SOURCE,INFO:MANUFACTURER\_CODE ,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:IMPORT\_TIME,INFO:SERVICE\_CODE\_IN,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_HTTP");  
  
*--聊天表***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_IM\_CHAT (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 ROOM\_ID string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 ACCOUNT string,  
 ACOUNT\_NAME string,  
 FRIEND\_ACCOUNT string,  
 FRIEND\_NAME string,  
 CHAT\_TYPE string,  
 SENDER\_ACCOUNT string,  
 SENDER\_NAME string,  
 CHAT\_TIME string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 MSG string,  
 CHECKIN\_ID string,  
 DATA\_SOURCE string,  
 MACHINE\_ID string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_IN string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string  
 )  
 ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SESSIONID,INFO:SERVICE\_CODE ,INFO:ROOM\_ID,INFO:CERTIFICATE\_TYPE ,INFO:CERTIFICATE\_CODE ,INFO:USER\_NAME,INFO:PROTOCOL\_TYPE,INFO:ACCOUNT,INFO:ACOUNT\_NAME,INFO:FRIEND\_ACCOUNT,INFO:FRIEND\_NAME,INFO:CHAT\_TYPE,INFO:SENDER\_ACCOUNT,INFO:SENDER\_NAME,INFO:CHAT\_TIME,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:MSG ,INFO:CHECKIN\_ID ,INFO:DATA\_SOURCE ,INFO:MACHINE\_ID ,INFO:MANUFACTURER\_CODE,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:IMPORT\_TIME,INFO:SERVICE\_CODE\_IN,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_IM\_CHAT");  
  
*--搜索***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_SEARCH (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 ROOM\_ID string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 URL string,  
 DOMAIN\_NAME string,  
 REF\_URL string,  
 REF\_DOMAIN string,  
 KEYWORD string,  
 KEYWORD\_CODE string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 CHECKIN\_ID string,  
 MACHINE\_ID string,  
 DATA\_SOURCE string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_IN string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string  
)  
 ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SESSIONID,INFO:SERVICE\_CODE,INFO:ROOM\_ID,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE,INFO:USER\_NAME ,INFO:PROTOCOL\_TYPE,INFO:URL,INFO:DOMAIN\_NAME,INFO:REF\_URL,INFO:REF\_DOMAIN,INFO:KEYWORD,INFO:KEYWORD\_CODE,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:CHECKIN\_ID,INFO:MACHINE\_ID,INFO:DATA\_SOURCE,INFO:MANUFACTURER\_CODE ,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:IMPORT\_TIME,INFO:SERVICE\_CODE\_IN,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_SEARCH");  
  
*--真实***CREATE EXTERNAL TABLE** neirong.H\_REG\_REALID\_INFO (  
 ID string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 SEX string,  
 BIRTHDAY string,  
 PEOPLE string,  
 COUNTRY string,  
 ORG\_NAME string,  
 REMARK string,  
 LAST\_LOGINTIME string,  
 LAST\_SERVICE\_CODE string,  
 USE\_NUMS string,  
 FIRST\_LOGINTIME string,  
 FIRST\_SERVICE\_CODE string,  
 FIRST\_ROOM\_ID string,  
 LAST\_ROOM\_ID string,  
 QQ\_NUMS string,  
 USER\_ADDRESS string,  
 USER\_PHONE string,  
 ID\_EXPIRETIME string  
 )  
 ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE,INFO:USER\_NAME,INFO:SEX,INFO:BIRTHDAY,INFO:PEOPLE,INFO:COUNTRY,INFO:ORG\_NAME,INFO:REMARK,INFO:LAST\_LOGINTIME,INFO:LAST\_SERVICE\_CODE,INFO:USE\_NUMS,INFO:FIRST\_LOGINTIME,INFO:FIRST\_SERVICE\_CODE,INFO:FIRST\_ROOM\_ID,INFO:LAST\_ROOM\_ID,INFO:QQ\_NUMS,INFO:USER\_ADDRESS,INFO:USER\_PHONE,INFO:ID\_EXPIRETIME")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_REALID\_INFO");  
  
*--虚拟***CREATE EXTERNAL TABLE** neirong.H\_REG\_VID\_INFO (  
 ID string,  
 PROTOCOL\_TYPE string,  
 ACCOUNT string,  
 NICK\_NAME string,  
 PASSWD string,  
 USE\_NUMS string,  
 LAST\_LOGINTIME string,  
 LAST\_SERVICE\_CODE string,  
 FIRST\_LOGINTIME string,  
 FIRST\_SERVICE\_CODE string,  
 FIRST\_ROOM\_ID string,  
 LAST\_ROOM\_ID string,  
 REALID\_NUMS string  
)  
ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:PROTOCOL\_TYPE ,INFO:ACCOUNT ,INFO:NICK\_NAME,INFO:PASSWD,INFO:USE\_NUMS,INFO:LAST\_LOGINTIME,INFO:LAST\_SERVICE\_CODE,INFO:FIRST\_LOGINTIME,INFO:FIRST\_SERVICE\_CODE,INFO:FIRST\_ROOM\_ID,INFO:LAST\_ROOM\_ID,INFO:REALID\_NUMS")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_VID\_INFO");  
  
*--场所***CREATE EXTERNAL TABLE** neirong.H\_SERVICE\_INFO (  
 GROUP\_ID string,  
 SERVICE\_TYPE string,  
 SERVICE\_CODE string,  
 SERVICE\_NAME string,  
 ADDRESS string,  
 POSTAL\_CODE string,  
 PRINCIPAL string,  
 PRINCIPAL\_TEL string,  
 INFOR\_MAN string,  
 INFOR\_MAN\_TEL string,  
 INFOR\_MAN\_EMAIL string,  
 ISP string,  
 STATUS string,  
 ENDING\_NUMS string,  
 SERVER\_NUMS string,  
 SERVICE\_IP string,  
 NET\_TYPE string,  
 PRACTITIONER\_COUNT string,  
 NET\_MONITOR\_DEPARTMENT string,  
 NET\_MONITOR\_MAN string,  
 NET\_MONITOR\_MAN\_TEL string,  
 REMARK string,  
 DATA\_SOURCE string,  
 MACHINE\_ID string,  
 SERVICE\_NAME\_PIN\_YIN string,  
 ONLINE\_STATUS string,  
 UPDATE\_TIME string,  
 PROBE\_VERSION string,  
 TEMPLET\_VERSION string,  
 LONGITUDE\_LATITUDE string,  
 SPACE\_SIZE string,  
 LATITUDE string,  
 IP\_ADDRESS string,  
 IS\_OUTLINE\_ALERT string,  
 ELEVATION string,  
 SERVICE\_IMG string,  
 AGEN\_LAVE\_TIME string,  
 REAL\_ENDING\_NUMS string,  
 CAUSE string,  
 IS\_ALLOW\_INSERT string,  
 BUSINESS\_NATURE string,  
 LAW\_PRINCIPAL\_CERTIFICATE\_TYPE string,  
 LAW\_PRINCIPAL\_CERTIFICATE\_ID string,  
 START\_TIME string,  
 END\_TIME string,  
 IS\_READ string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 INSTALL\_TIME string,  
 POLIC\_STATION string,  
 MANUFACTURER\_CODE string,  
 AREA\_CODE string,  
 IS\_VIRTUAL string,  
 IF\_CHECK string,  
 SERVICE\_CODE\_IN string  
)  
ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SERVICE\_TYPE,INFO:SERVICE\_CODE,INFO:SERVICE\_NAME,INFO:ADDRESS,INFO:POSTAL\_CODE,INFO:PRINCIPAL,INFO:PRINCIPAL\_TEL,INFO:INFOR\_MAN,INFO:INFOR\_MAN\_TEL,INFO:INFOR\_MAN\_EMAIL,INFO:ISP,INFO:STATUS,INFO:ENDING\_NUMS,INFO:SERVER\_NUMS,INFO:SERVICE\_IP,INFO:NET\_TYPE,INFO:PRACTITIONER\_COUNT,INFO:NET\_MONITOR\_DEPARTMENT,INFO:NET\_MONITOR\_MAN,INFO:NET\_MONITOR\_MAN\_TEL,INFO:REMARK,INFO:DATA\_SOURCE,INFO:MACHINE\_ID,INFO:SERVICE\_NAME\_PIN\_YIN,INFO:ONLINE\_STATUS,INFO:UPDATE\_TIME,INFO:PROBE\_VERSION,INFO:TEMPLET\_VERSION,INFO:LONGITUDE\_LATITUDE,INFO:SPACE\_SIZE,INFO:LATITUDE,INFO:IP\_ADDRESS,INFO:IS\_OUTLINE\_ALERT,INFO:ELEVATION,INFO:SERVICE\_IMG,INFO:AGEN\_LAVE\_TIME,INFO:REAL\_ENDING\_NUMS,INFO:CAUSE,INFO:IS\_ALLOW\_INSERT,INFO:BUSINESS\_NATURE,INFO:LAW\_PRINCIPAL\_CERTIFICATE\_TYPE,INFO:LAW\_PRINCIPAL\_CERTIFICATE\_ID,INFO:START\_TIME,INFO:END\_TIME,INFO:IS\_READ,INFO:ZIPNAME,INFO:BCPNAME,INFO:ROWNUMBER,INFO:INSTALL\_TIME,INFO:POLIC\_STATION,INFO:MANUFACTURER\_CODE,INFO:AREA\_CODE,INFO:IS\_VIRTUAL,INFO:IF\_CHECK,INFO:SERVICE\_CODE\_IN")  
TBLPROPERTIES("hbase.table.name" = "H\_SERVICE\_INFO");  
  
*--微博***CREATE EXTERNAL TABLE** neirong.H\_REG\_CONTENT\_WEIBO (  
 ID string,  
 SESSIONID string,  
 SERVICE\_CODE string,  
 ROOM\_ID string,  
 CERTIFICATE\_TYPE string,  
 CERTIFICATE\_CODE string,  
 USER\_NAME string,  
 PROTOCOL\_TYPE string,  
 ACCOUNT string,  
 PASSWD string,  
 URL string,  
 DOMAIN\_NAME string,  
 REF\_URL string,  
 REF\_DOMAIN string,  
 POSTING\_ID string,  
 TITLE string,  
 AUTHOR string,  
 SOURCE string,  
 ACTION\_TYPE string,  
 SUMMARY string,  
 FILE\_PATH string,  
 DEST\_IP string,  
 DEST\_PORT string,  
 SRC\_IP string,  
 SRC\_PORT string,  
 SRC\_MAC string,  
 CAPTURE\_TIME string,  
 CHECKIN\_ID string,  
 MACHINE\_ID string,  
 DATA\_SOURCE string,  
 MANUFACTURER\_CODE string,  
 ZIPNAME string,  
 BCPNAME string,  
 ROWNUMBER string,  
 IMPORT\_TIME string,  
 SERVICE\_CODE\_OUT string,  
 TERMINAL\_LONGITUDE string,  
 TERMINAL\_LATITUDE string,  
 CERTIFICATE\_NAME string,  
 COORDINATE\_TYPE string,  
 LONGITUDE\_BAIDU string,  
 LATITUDE\_BAIDU string,  
 LONGITUDE\_HARDWARE string,  
 LATITUDE\_HARDWARE string,  
 IS\_UPDATED string  
)  
ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:SESSIONID ,INFO:SERVICE\_CODE,INFO:ROOM\_ID,INFO:CERTIFICATE\_TYPE,INFO:CERTIFICATE\_CODE,INFO:USER\_NAME,INFO:PROTOCOL\_TYPE,INFO:ACCOUNT,INFO:PASSWD,INFO:URL,INFO:DOMAIN\_NAME,INFO:REF\_URL,INFO:REF\_DOMAIN,INFO:POSTING\_ID,INFO:TITLE,INFO:AUTHOR,INFO:SOURCE,INFO:ACTION\_TYPE,INFO:SUMMARY,INFO:FILE\_PATH,INFO:DEST\_IP,INFO:DEST\_PORT,INFO:SRC\_IP,INFO:SRC\_PORT,INFO:SRC\_MAC,INFO:CAPTURE\_TIME,INFO:CHECKIN\_ID,INFO:MACHINE\_ID,INFO:DATA\_SOURCE ,INFO:MANUFACTURER\_CODE ,INFO:ZIPNAME ,INFO:BCPNAME ,INFO:ROWNUMBER,INFO:IMPORT\_TIME ,INFO:SERVICE\_CODE\_OUT,INFO:TERMINAL\_LONGITUDE,INFO:TERMINAL\_LATITUDE,INFO:CERTIFICATE\_NAME,INFO:COORDINATE\_TYPE, INFO:LONGITUDE\_BAIDU, INFO:LATITUDE\_BAIDU, INFO:LONGITUDE\_HARDWARE, INFO:LATITUDE\_HARDWARE, INFO:IS\_UPDATED")  
TBLPROPERTIES("hbase.table.name" = "H\_REG\_CONTENT\_WEIBO");  
  
**CREATE EXTERNAL TABLE** neirong.H\_SYS\_DICT (  
 ID string,  
 DICT\_CODE string,  
 DICT\_NAME string,  
 REMARK string,  
 PARENT\_ID string,  
 STATUS string,  
 RESERVE\_STATUS string  
 )  
ROW FORMAT SERDE **'org.apache.hadoop.hive.hbase.HBaseSerDe'**STORED **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH** SERDEPROPERTIES ("hbase.columns.mapping" = ":key,INFO:DICT\_CODE,INFO:DICT\_NAME,INFO:REMARK,INFO:PARENT\_ID,INFO:STATUS,INFO:RESERVE\_STATUS")  
TBLPROPERTIES("hbase.table.name" = "H\_SYS\_DICT");