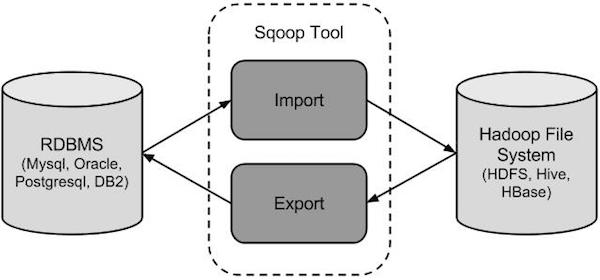
# sqoop数据迁移

## 概述

sqoop是apache旗下一款**“Hadoop和关系数据库服务器之间传送数据”**的工具。

**导入数据**：MySQL，Oracle导入数据到Hadoop的HDFS、HIVE、HBASE等数据存储系统；

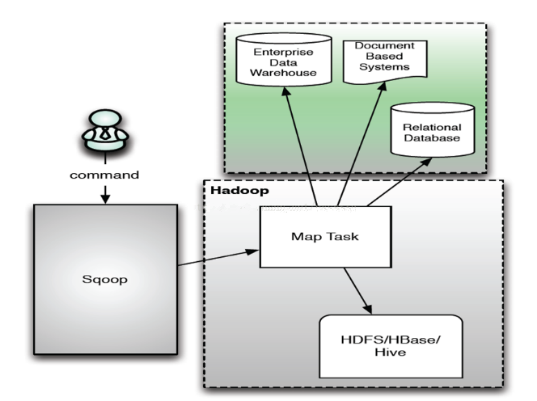
**导出数据：**从Hadoop的HDFS、HIVE中导出数据到关系数据库mysql等



## sqoop1与sqoop2架构对比

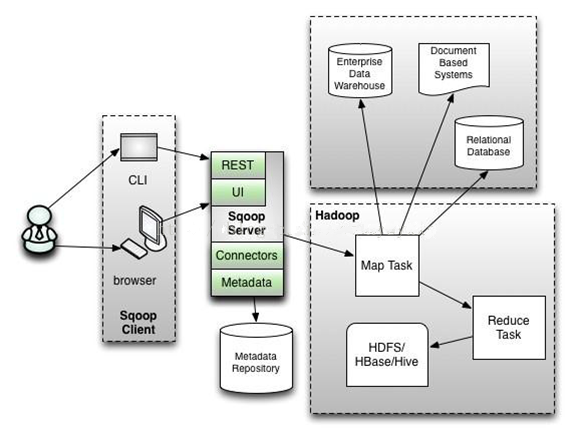
### sqoop1架构

Sqoop1以Client客户端的形式存在和运行。没有任务时是没有进程存在的。



### sqoop2架构

sqoop2是以B/S服务器的形式去运行的，始终会有Server服务端进程在运行。



## 工作机制

将导入或导出命令翻译成mapreduce程序来实现。

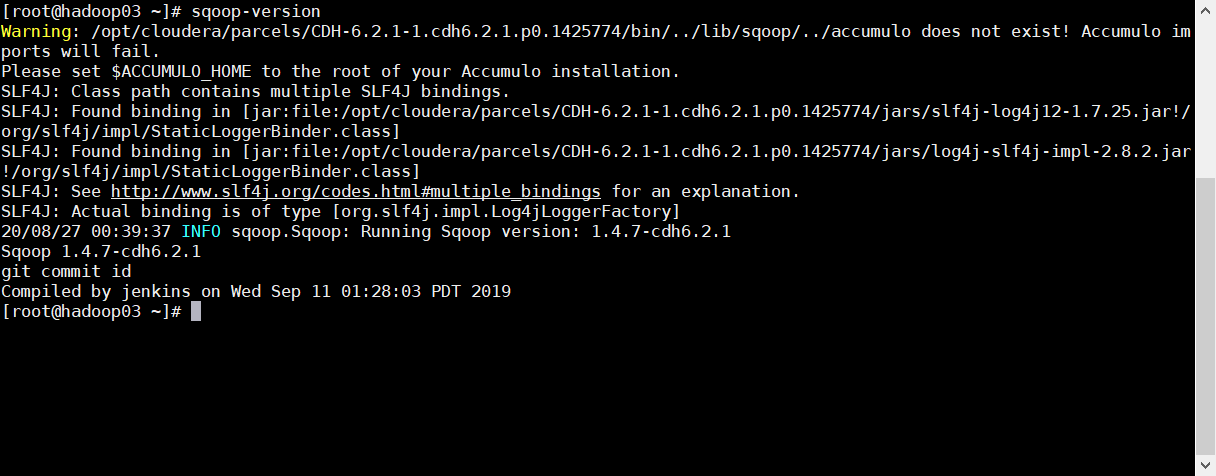
## sqoop安装

### sqoop安装

略。

### 验证启动

sqoop-version



## Sqoop抽取的两种方式

对于Mysql数据的采集，通常使用Sqoop来进行。

通过Sqoop将关系型数据库数据到Hive有两种方式，一种是原生Sqoop API，一种是使用HCatalog API。两种方式略有不同。

HCatalog方式与Sqoop方式的参数基本都是相同，只是个别不一样，都是可以实现Sqoop将数据抽取到Hive。

### 区别

**数据格式支持**

Sqoop方式支持的数据格式较少，HCatalog支持的数据格式多，包括RCFile, ORCFile, CSV, JSON和SequenceFile等格式。

**数据覆盖**

Sqoop方式允许数据覆盖，HCatalog不允许数据覆盖，每次都只是追加。

**字段名**

Sqoop方式比较随意，不要求源表和目标表字段相同(字段名称和个数都可以不相同)，它抽取的方式是将字段按顺序插入，比如目标表有3个字段，源表有一个字段，它会将数据插入到Hive表的第一个字段，其余字段为NULL。但是HCatalog不同，源表和目标表字段名需要相同，字段个数可以不相等，如果字段名不同，抽取数据的时候会报NullPointerException错误。HCatalog抽取数据时，会将字段对应到相同字段名的字段上，哪怕字段个数不相等。

### Sqoop方式

|  |
| --- |
| sqoop import \  --hive-import \  --connect 'jdbc:mysql://localhost:3306/test' \  --username 'root' \  --password '123456789' \  --query " select order\_no from driver\_action where \$CONDITIONS" \  --hive-database test \  --hive-table driver\_action \  --hive-partition-key pt \  --hive-partition-value 20190901 \  --null-string '' \  --null-non-string '' \  --num-mappers 1 \  --target-dir /tmp/test \  --delete-target-dir |

### HCatalog方式

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://localhost:3306/test\  --username 'root' \  --password 'root' \  --query "SELECT order\_no FROM driver\_action WHERE \$CONDITIONS" \  --hcatalog-database test \  --hcatalog-table driver\_action \  --hcatalog-partition-keys pt \  --hcatalog-partition-values 20200104 \  --hcatalog-storage-stanza 'stored as orcfile tblproperties ("orc.compress"="SNAPPY")' \  --num-mappers 1 |

针对不同字段名，想要使用HCatalog方式将数据插入，可以使用下面的方式：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://localhost:3306/test\  --username 'root' \  --password 'root' \  --query "SELECT order\_no\_src **as order\_no\_target** FROM driver\_action WHERE \$CONDITIONS" \  --hcatalog-database test \  --hcatalog-table driver\_action \  --hcatalog-partition-keys pt \  --hcatalog-partition-values 20200104 \  --hcatalog-storage-stanza 'stored as orc tblproperties ("orc.compress"="SNAPPY")' \  --num-mappers 1 |

## 项目选型

因为项目采用的是**ORC** File文件格式，sqoop原始方式并不支持，因此使用HCatalog方式来进行数据的导入导出。

## Sqoop的数据导入

“导入工具”导入单个表从RDBMS到HDFS。表中的每一行被视为HDFS的记录。所有记录都存储为文本文件的文本数据（或者Avro、sequence文件等二进制数据）

### 列举出所有的数据库

/usr/bin/sqoop help

命令行查看帮助

/usr/bin/sqoop list-databases --help

列出主机所有的数据库

/usr/bin/sqoop list-databases --connect jdbc:mysql://192.168.52.150:3306/ --username root --password 123456

查看某一个数据库下面的所有数据表

/usr/bin/sqoop list-tables --connect jdbc:mysql://192.168.52.150:3306/hive --username root --password 123456

### 完整数据导入

#### 表数据

在mysql中有一个库test中三个表：emp, emp\_add和emp\_conn。

测试数据sql在【Home\讲义\第2章 数据仓库\sqoop\mysql数据\】目录中，可以使用SQLyog等mysql客户端进行导入。

表emp:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **deg** | **salary** | **dept** |
| 1201 | gopal | manager | 50,000 | TP |
| 1202 | manisha | Proof reader | 50,000 | TP |
| 1203 | khalil | php dev | 30,000 | AC |
| 1204 | prasanth | php dev | 30,000 | AC |
| 1205 | kranthi | admin | 20,000 | TP |

表emp\_add:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **hno** | **street** | **city** |
| 1201 | 288A | vgiri | jublee |
| 1202 | 108I | aoc | sec-bad |
| 1203 | 144Z | pgutta | hyd |
| 1204 | 78B | old city | sec-bad |
| 1205 | 720X | hitec | sec-bad |

表emp\_conn:

|  |  |  |
| --- | --- | --- |
| **id** | **phno** | **email** |
| 1201 | 2356742 | gopal@tp.com |
| 1202 | 1661663 | manisha@tp.com |
| 1203 | 8887776 | khalil@ac.com |
| 1204 | 9988774 | prasanth@ac.com |
| 1205 | 1231231 | kranthi@tp.com |

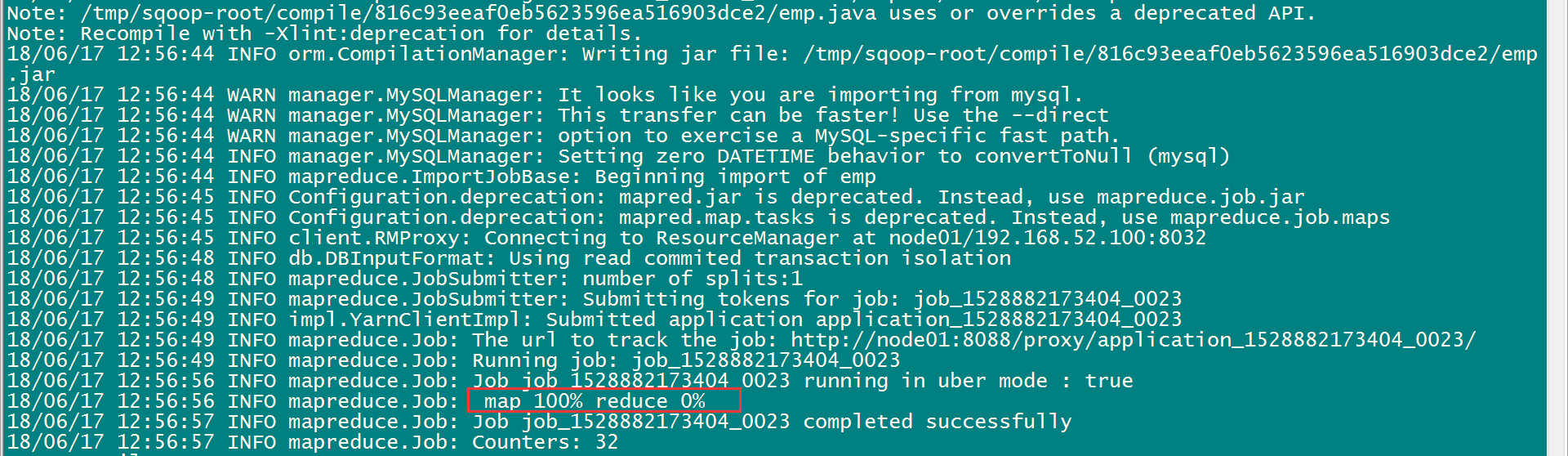
#### 导入数据库表数据到HDFS

下面的命令用于从MySQL数据库服务器中的emp表导入HDFS。

/usr/bin/sqoop import --connect jdbc:mysql://192.168.52.150:3306/test --password 123456 --username root --table emp --m 1

注意，mysql地址必须为服务器IP，不能是localhost或者机器名。

如果成功执行，那么会得到下面的输出。



为了验证在HDFS导入的数据，请使用以下命令查看导入的数据

hdfs dfs -ls /user/root/emp

#### 导入到HDFS指定目录

在导入表数据到HDFS时，使用Sqoop导入工具，我们可以指定目标目录。

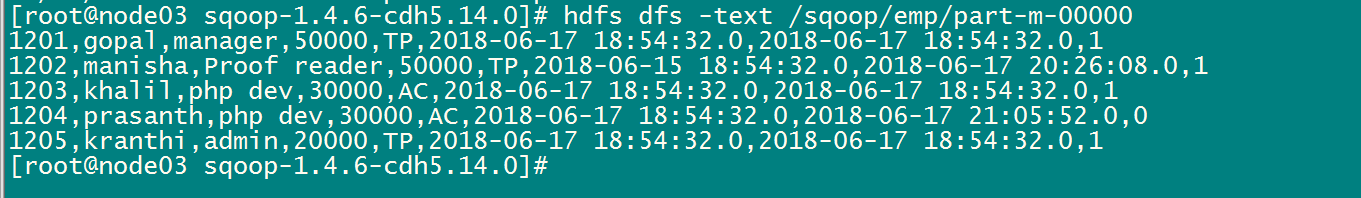
使用参数 --target-dir来指定导出目的地，

使用参数--delete-target-dir来判断导出目录是否已存在，如果存在就删掉

/usr/bin/sqoop import --connect jdbc:mysql://192.168.52.150:3306/test --username root --password 123456 --delete-target-dir --table emp --target-dir /sqoop/emp --m 1

查看导出的数据

hdfs dfs -text /sqoop/emp/part-m-00000



它会用逗号（，）分隔emp\_add表的数据和字段。

1201,gopal,manager,50000,TP

1202,manisha,Proof reader,50000,TP

1203,khalil,php dev,30000,AC

1204,prasanth,php dev,30000,AC

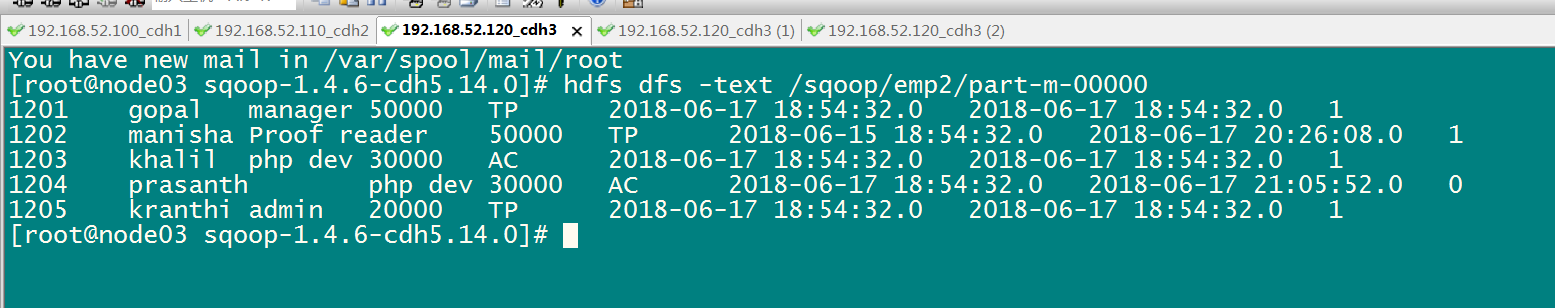
1205,kranthi,admin,20000,TP

#### 导入到hdfs指定目录并指定字段之间的分隔符

/usr/bin/sqoop import --connect jdbc:mysql://192.168.52.150:3306/test --username root --password 123456 --delete-target-dir --table emp --target-dir /sqoop/emp2 --m 1 --fields-terminated-by '\t'

查看文件内容

hdfs dfs -text /sqoop/emp2/part-m-00000



#### 导入关系表到HIVE

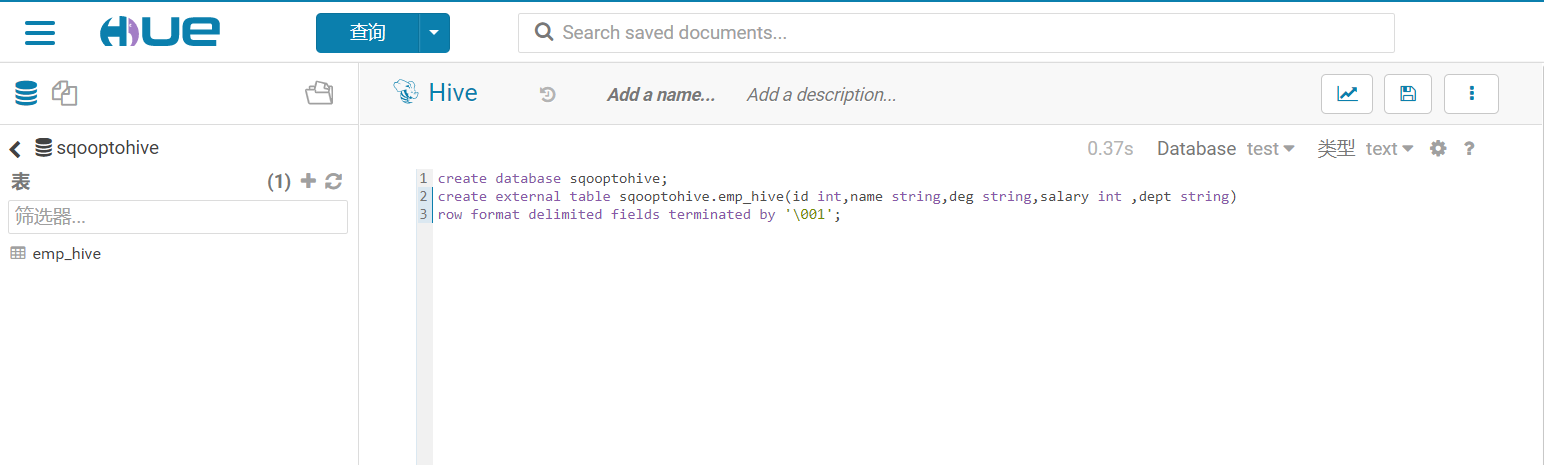
##### 第一步：准备hive数据库与表

将我们mysql当中的数据导入到hive表当中来

hive (default)> create database sqooptohive;

hive (default)> use sqooptohive;

hive (sqooptohive)> create table sqooptohive.emp\_hive(id int,name string,deg string,salary int ,dept string) row format delimited fields terminated by '\t' stored as orc;



##### 第三步：开始导入

/usr/bin/sqoop import \

--connect jdbc:mysql://192.168.52.150:3306/test \

--username root \

--password 123456 \

--table emp \

--fields-terminated-by '\t' \

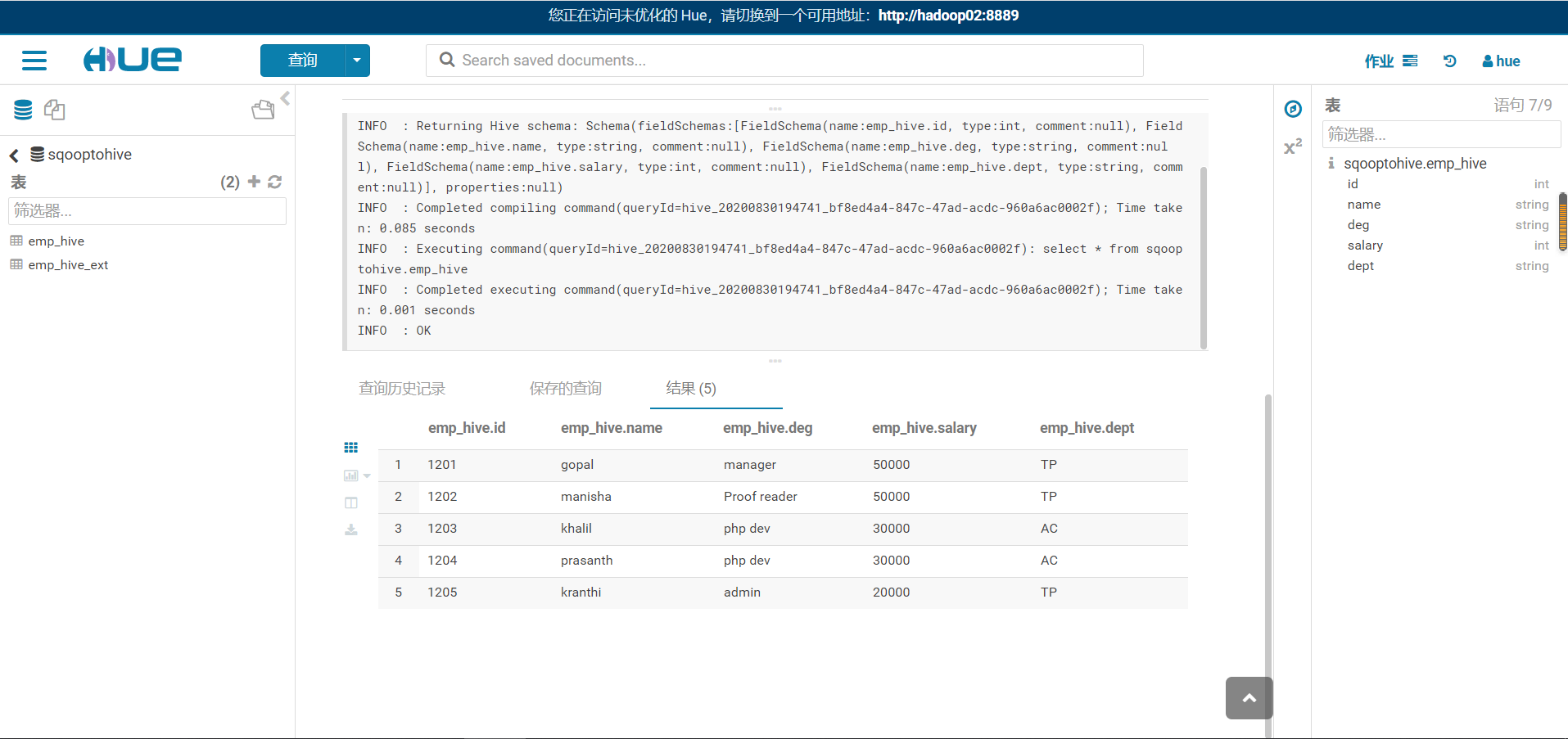
--hcatalog-database sqooptohive \

--hcatalog-table emp\_hive \

-m 1

##### 第四步：hive表数据查看

select \* from sqooptohive.emp\_hive;



### 条件部分导入

#### where导入到HDFS

我们可以导入表时使用Sqoop导入工具，"where"子句的一个子集。它执行在各自的数据库服务器相应的SQL查询，并将结果存储在HDFS的目标目录。

where子句的语法如下。

|  |
| --- |
| --where <condition> |

按照条件进行查找，通过—where参数来查找表emp\_add当中city字段的值为sec-bad的所有数据导入到hdfs上面去

/usr/bin/sqoop import \

--connect jdbc:mysql://192.168.52.150:3306/test \

--username root --password 123456 --table emp\_add \

--target-dir /sqoop/*emp\_add* -m 1 --delete-target-dir \

--where "city = 'sec-bad'"

#### sql语句查找导入hdfs

我们还可以通过 –query参数来指定我们的sql语句，通过sql语句来过滤我们的数据进行导入

/usr/bin/sqoop import \

--connect jdbc:mysql://192.168.52.150:3306/test --username root --password 123456 \

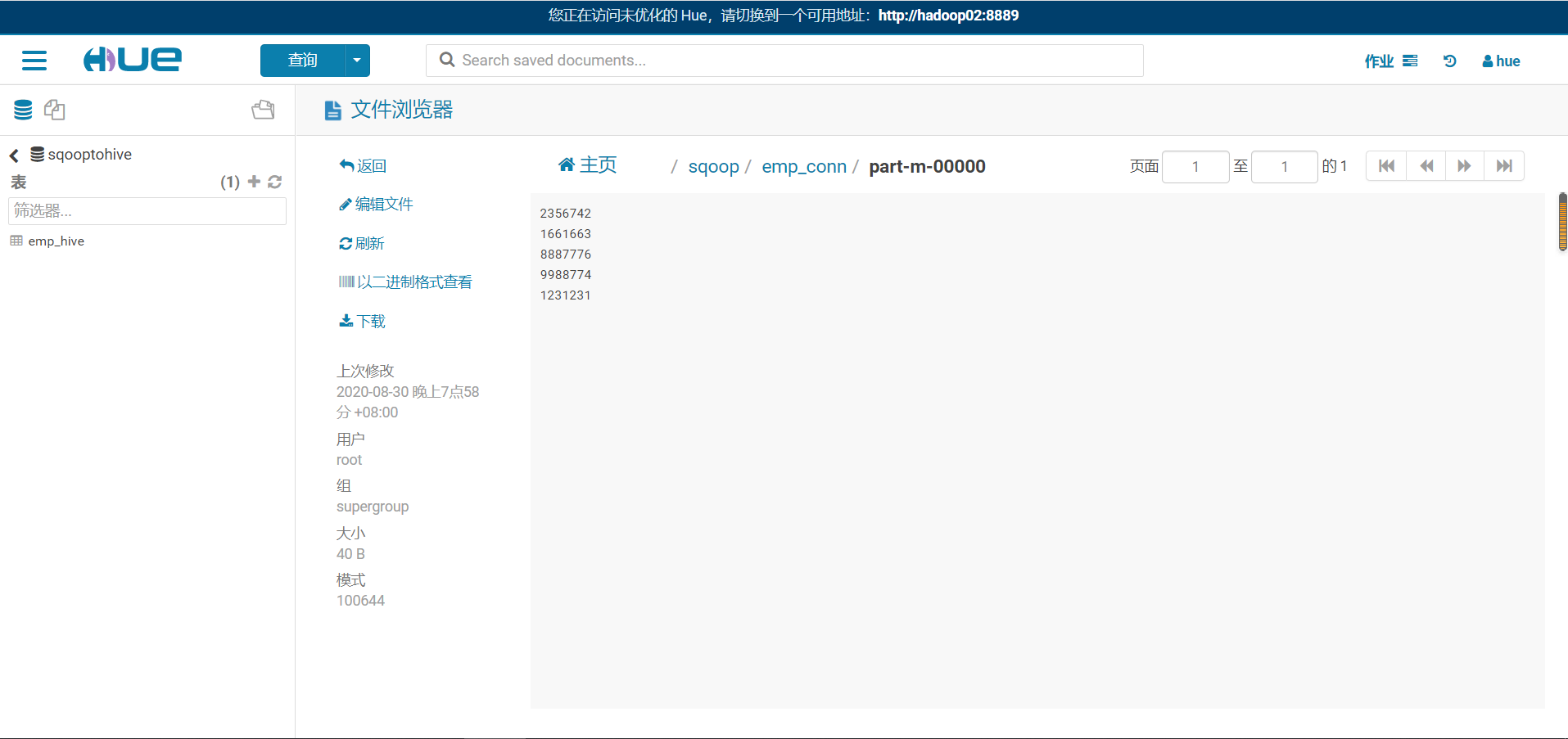
--delete-target-dir -m 1 \

--query 'select phno from *emp\_conn* where 1=1 and $CONDITIONS' \

--target-dir /sqoop/*emp\_conn*

查看hdfs数据内容

hdfs dfs -text /sqoop/emp\_conn/part\*



#### 增量导入数据到Hive表

/usr/bin/sqoop import \

--connect jdbc:mysql://192.168.52.150:3306/test --username root --password 123456 \

--query "select \* from emp **where id>1203** and \$CONDITIONS" \

--fields-terminated-by '\t' \

--hcatalog-database sqooptohive \

--hcatalog-table emp\_hive \

-m 1



## Sqoop的数据导出

#### 第一步：创建mysql表

CREATE TABLE `emp\_out` (

`id` INT(11) DEFAULT NULL,

`name` VARCHAR(100) DEFAULT NULL,

`deg` VARCHAR(100) DEFAULT NULL,

`salary` INT(11) DEFAULT NULL,

`dept` VARCHAR(10) DEFAULT NULL

) ENGINE=INNODB DEFAULT CHARSET=utf8;

#### 第二步：执行导出命令

通过export来实现数据的导出，将hive的数据导出到mysql当中去

/usr/bin/sqoop export \

--connect jdbc:mysql://192.168.52.150:3306/test --username root --password 123456 \

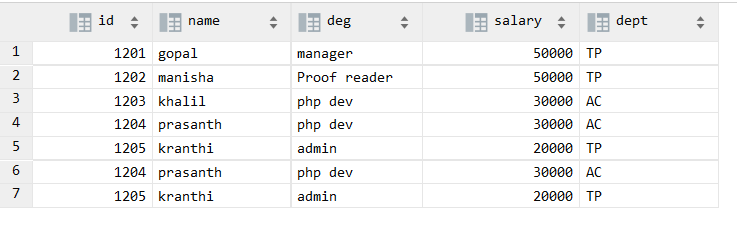
--table emp\_out \

--hcatalog-database sqooptohive \

--hcatalog-table emp\_hive \

-m 1

#### 第三步：验证mysql表数据



## Sqoop一些常用参数

|  |  |
| --- | --- |
| 参数 | 说明 |
| --connect | 连接关系型数据库的URL |
| --username | 连接数据库的用户名 |
| --password | 连接数据库的密码 |
| --driver | JDBC的driver class |
| --query或--e <statement> | 将查询结果的数据导入，使用时必须伴随参--target-dir，--hcatalog-table，如果查询中有where条件，则条件后必须加上$CONDITIONS关键字。  如果使用双引号包含sql，则$CONDITIONS前要加上\以完成转义：\$CONDITIONS |
| --hcatalog-database | 指定HCatalog表的数据库名称。如果未指定，default则使用默认数据库名称。提供 --hcatalog-database不带选项--hcatalog-table是错误的。 |
| --hcatalog-table | 此选项的参数值为HCatalog表名。该--hcatalog-table选项的存在表示导入或导出作业是使用HCatalog表完成的，并且是HCatalog作业的必需选项。 |
| --create-hcatalog-table | 此选项指定在导入数据时是否应自动创建HCatalog表。表名将与转换为小写的数据库表名相同。 |
| --hcatalog-storage-stanza 'stored as orc tblproperties ("orc.compress"="SNAPPY")' \ | 建表时追加存储格式到建表语句中，tblproperties修改表的属性，这里设置orc的压缩格式为SNAPPY |
| -m | 指定并行处理的MapReduce任务数量。  -m不为1时，需要用split-by指定分片字段进行并行导入，尽量指定int型。 |
| --split-by id | 如果指定-split by, 必须使用$CONDITIONS关键字, 双引号的查询语句还要加\ |
| --hcatalog-partition-keys  --hcatalog-partition-values | keys和values必须同时存在，相当于指定静态分区。允许将多个键和值提供为静态分区键。多个选项值之间用，（逗号）分隔。比如：  --hcatalog-partition-keys year,month,day  --hcatalog-partition-values 1999,12,31 |
| --null-string '\\N'  --null-non-string '\\N' | 指定mysql数据为空值时用什么符号存储，null-string针对string类型的NULL值处理，--null-non-string针对非string类型的NULL值处理 |
| --hive-drop-import-delims | 设置无视字符串中的分割符（hcatalog默认开启） |
| --fields-terminated-by '\t' | 设置字段分隔符 |