分布式数据仓库 HIVE

一、实验环境

已经配置好的 Hadoop 伪分布式或完全分布式环境

二、实验内容

以下操作均在 hadoop 用户下进行

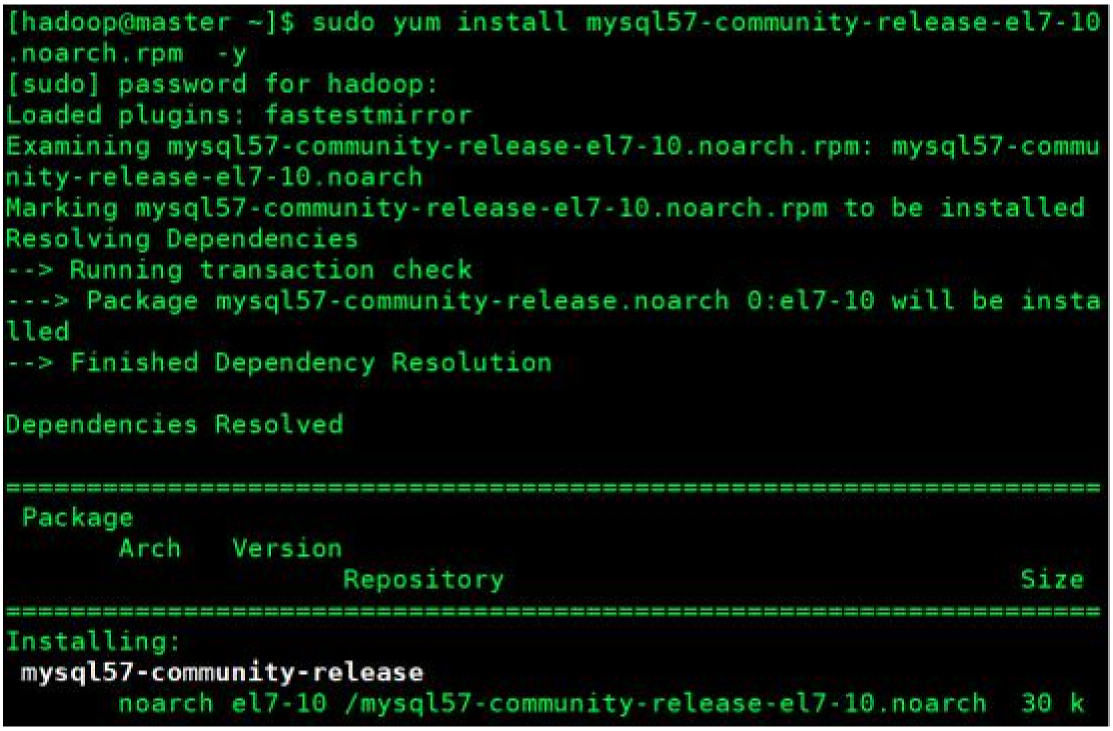
1. 安装 mysql

1） 下载 mysql5.7 的 yum 源

```
[hadoop@master ~]$ wget -i -c
http://dev.mysql.com/get/mysql57-community-release-el7-10.noarch.rpm
```

2） 安装 mysql 官方的源

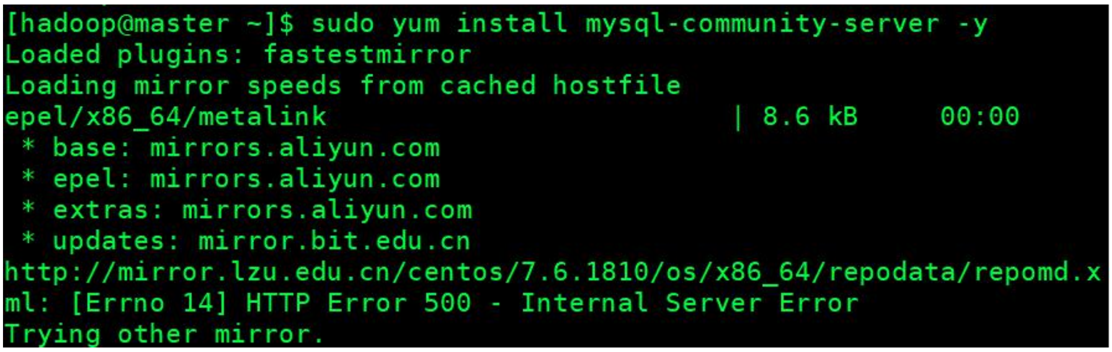
```
[hadoop@master ~]$ sudo yum install mysql57-community-release-el7-10.noarch.rpm
-y
```



```
[hadoop@master ~]$ sudo yum install mysql57-community-release-el7-10
.noarch.rpm  -y
[sudo] password for hadoop:
Loaded plugins: fastestmirror
Examining mysql57-community-release-el7-10.noarch.rpm: mysql57-commu
nity-release-el7-10.noarch
Marking mysql57-community-release-el7-10.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package mysql57-community-release.noarch 0:el7-10 will be insta
lled
--> Finished Dependency Resolution

Dependencies Resolved

====================================================================
 Package
       Arch   Version
                     Repository                                  Size
====================================================================
Installing:
 mysql57-community-release
       noarch el7-10 /mysql57-community-release-el7-10.noarch    30 k
```

3） 安装 mysql-server

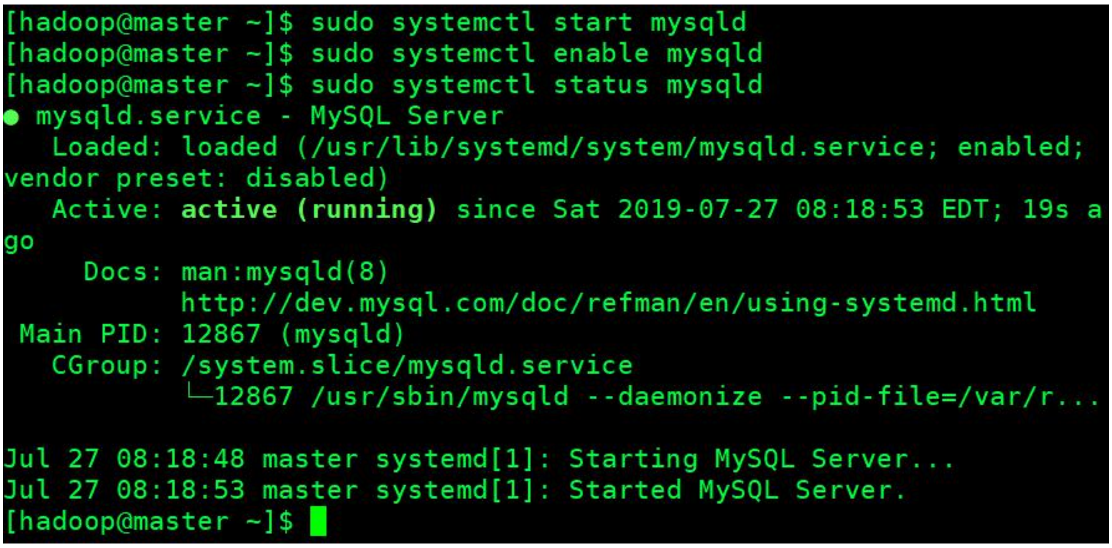
```
[hadoop@master ~]$ sudo yum install mysql-community-server -y
```



```
[hadoop@master ~]$ sudo yum install mysql-community-server -y
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
epel/x86_64/metalink                                     | 8.6 kB     00:00
 * base: mirrors.aliyun.com
 * epel: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirror.bit.edu.cn
http://mirror.lzu.edu.cn/centos/7.6.1810/os/x86_64/repodata/repomd.x
ml: [Errno 14] HTTP Error 500 - Internal Server Error
Trying other mirror.
```

4） 启动数据库以及设置开机自启动，查看 mysql-server 的状态

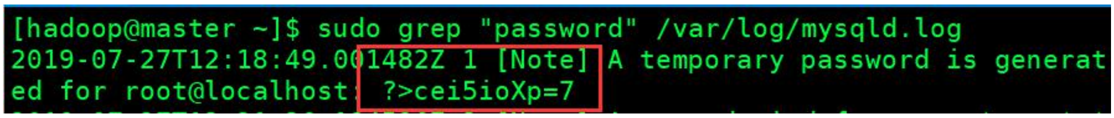
```
[hadoop@master ~]$ sudo systemctl start mysqld

[hadoop@master ~]$ sudo systemctl enable mysqld

[hadoop@master ~]$ sudo systemctl status mysqld
```



```
[hadoop@master ~]$ sudo systemctl start mysqld
[hadoop@master ~]$ sudo systemctl enable mysqld
[hadoop@master ~]$ sudo systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled;
vendor preset: disabled)
   Active: active (running) since Sat 2019-07-27 08:18:53 EDT; 19s a
go
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
 Main PID: 12867 (mysqld)
   CGroup: /system.slice/mysqld.service
           └─12867 /usr/sbin/mysqld --daemonize --pid-file=/var/r...

Jul 27 08:18:48 master systemd[1]: Starting MySQL Server...
Jul 27 08:18:53 master systemd[1]: Started MySQL Server.
[hadoop@master ~]$ █
```

5） 查看初始化密码

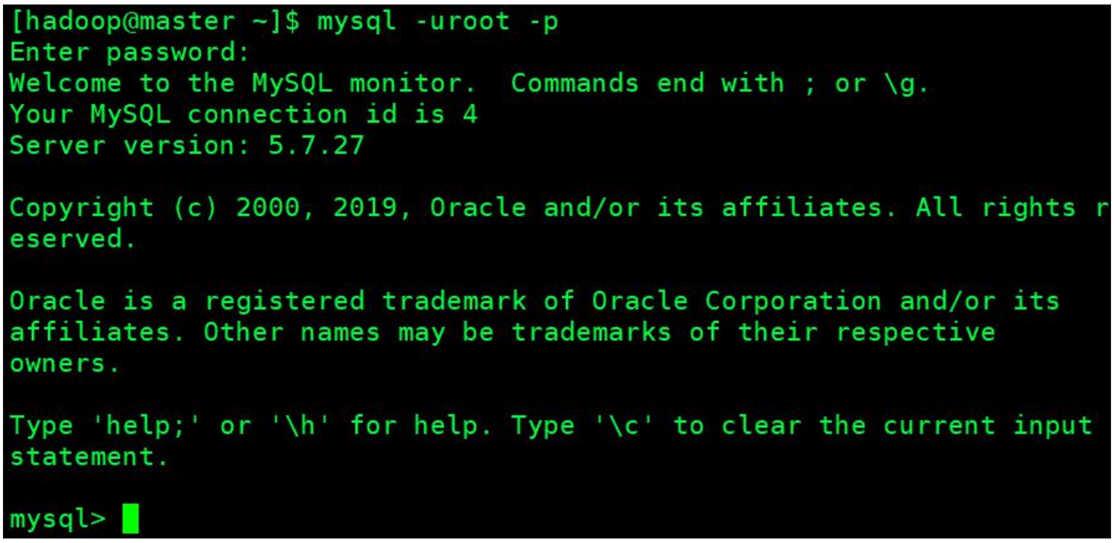
```
[hadoop@master ~]$ sudo grep "password" /var/log/mysqld.log
```



```
[hadoop@master ~]$ sudo grep "password" /var/log/mysqld.log
2019-07-27T12:18:49.001482Z 1 [Note] A temporary password is generat
ed for root@localhost: ?>cei5ioXp=7
```

6） 登陆 mysql 数据库

```
[hadoop@master ~]$ mysql -uroot -p
```

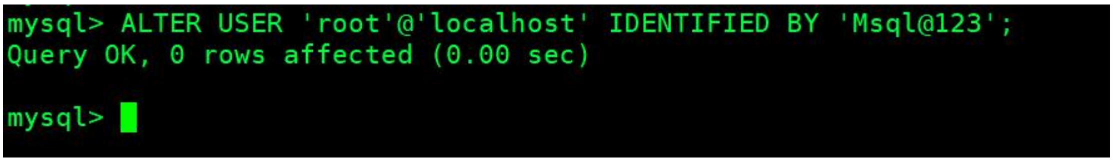


```
[hadoop@master ~]$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.27

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights r
eserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> 
```

7） 修改 root 账户密码

密码必须要是大写加小写字母加特殊字符加数字

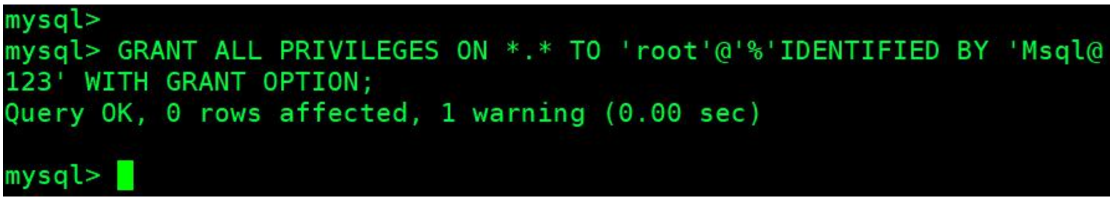
```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new password';
```



```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Msql@123';
Query OK, 0 rows affected (0.00 sec)

mysql> 
```

8） 授予 root 用户远程登陆权限

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%'IDENTIFIED BY 'new password' WITH
GRANT OPTION;
```

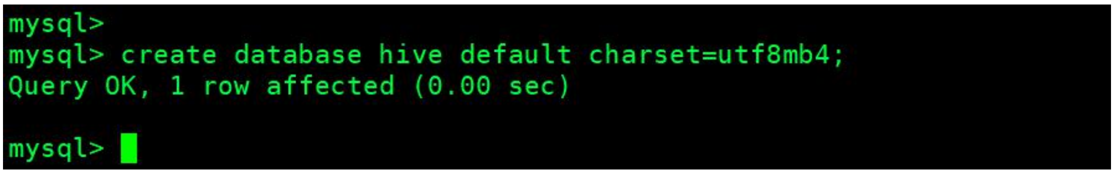


```
mysql>
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%'IDENTIFIED BY 'Msql@
123' WITH GRANT OPTION;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> 
```

9） 创建 hive 数据库

待会 hive 的元数据就存放在这个库里面

```
mysql> create database hive default charset=utf8mb4;
```



```
mysql>
mysql> create database hive default charset=utf8mb4;
Query OK, 1 row affected (0.00 sec)

mysql> 
```

2. 安装 hive

1）解压 Hive

```
sudo tar -zxvf /home/package/apache-hive-3.1.0-bin.tar.gz  -C /usr/
```

2）重命名安装路径

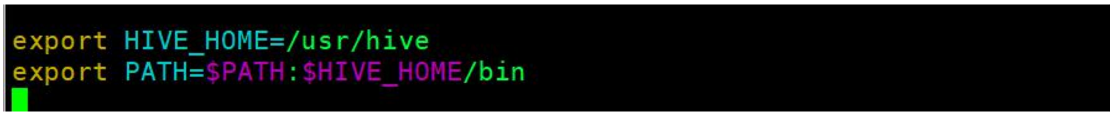
```
[hadoop@master ~]$ sudo mv /usr/apache-hive-3.1.0-bin/ /usr/hive
```

3）配置 hive 环境变量

```
[hadoop@master ~]$ sudo vim /etc/profile
```

在/etc/profile 文件中添加以下环境变量

```
export HIVE_HOME=/usr/hive

export PATH=$PATH:$HIVE_HOME/bin
```



```
export HIVE_HOME=/usr/hive
export PATH=$PATH:$HIVE_HOME/bin
```

4）使环境变量生效

```
[hadoop@master ~]$ source /etc/profile
```

5）配置 hive-site.xml 配置文件

因为 hive-site.xml 配置文件是不存在的，我们从默认配置文件复制一份

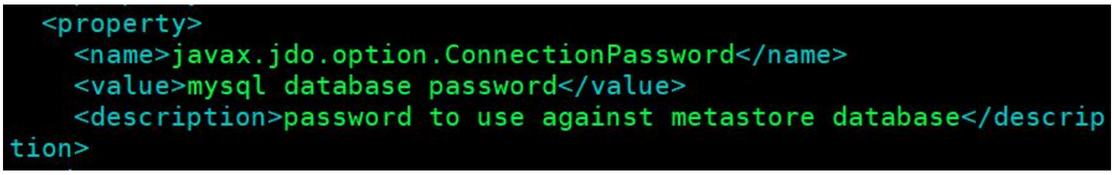
```
[hadoop@master ~]$ sudo cp  $HIVE_HOME/conf/hive-default.xml.template
$HIVE_HOME/conf/hive-site.xml
```

修改配置文件

```
[hadoop@master ~]$ sudo vim $HIVE_HOME/conf/hive-site.xml
```

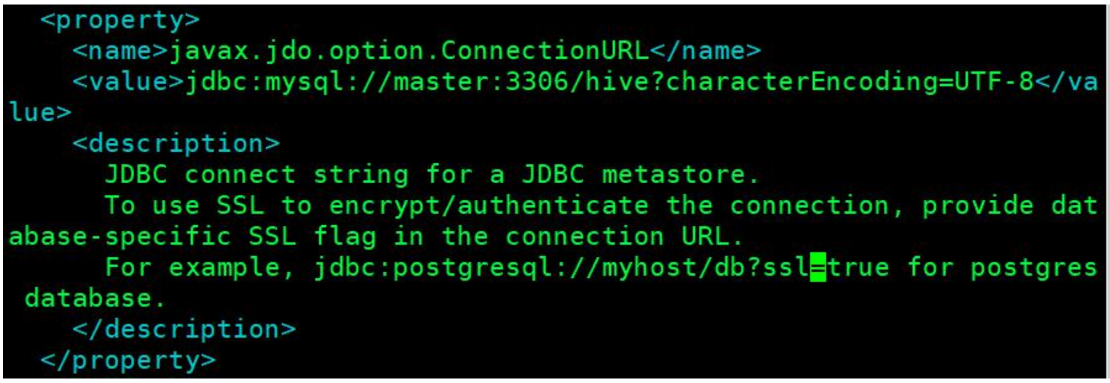
修改以下参数的值

```
<name>javax.jdo.option.ConnectionPassword</name>

<value>mysql database password</value>
```



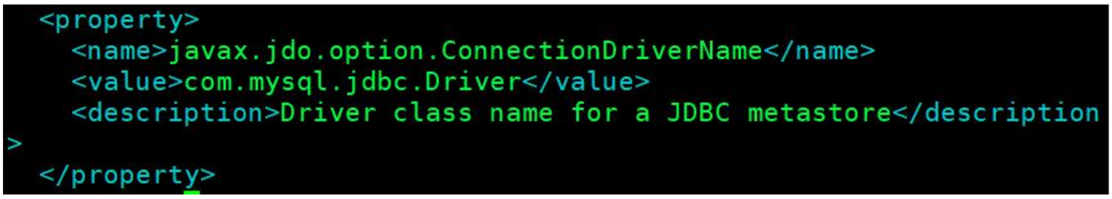
```
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>mysql database password</value>
    <description>password to use against metastore database</descrip
tion>
```

```
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://master:3306/hive?characterEncoding=UTF-8</value>
```



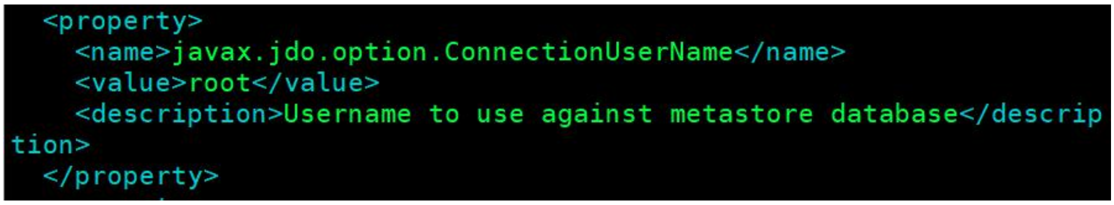
```
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://master:3306/hive?characterEncoding=UTF-8</va
lue>
    <description>
      JDBC connect string for a JDBC metastore.
      To use SSL to encrypt/authenticate the connection, provide dat
abase-specific SSL flag in the connection URL.
      For example, jdbc:postgresql://myhost/db?ssl=true for postgres
 database.
    </description>
  </property>
```

```
<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>
```



```
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description
>
  </property>
```

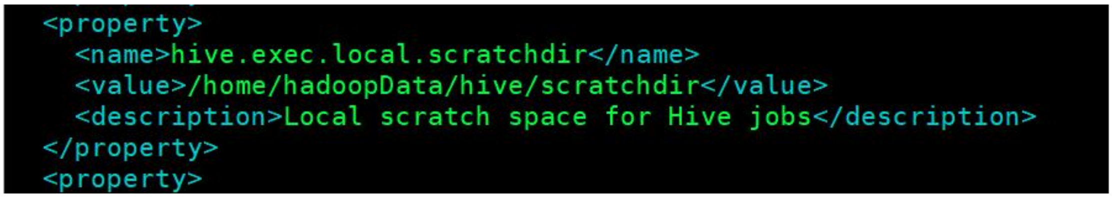
```
<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>
```



```
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
    <description>Username to use against metastore database</descrip
tion>
  </property>
```

Hive 作业的本地临时空间

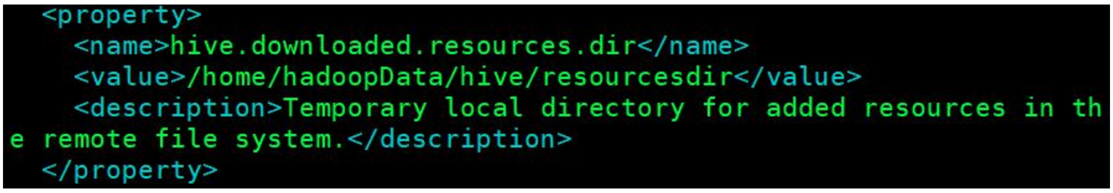
```
<name>hive.exec.local.scratchdir</name>

<value>/home/hadoopData/hive/scratchdir</value>
```



```
  <property>
    <name>hive.exec.local.scratchdir</name>
    <value>/home/hadoopData/hive/scratchdir</value>
    <description>Local scratch space for Hive jobs</description>
  </property>
  <property>
```

用于在远程文件系统中添加资源的临时本地目录。

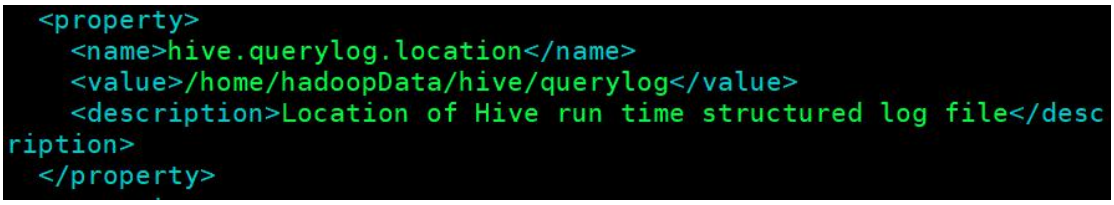
```
<name>hive.downloaded.resources.dir</name>
<value>/home/hadoopData/hive/resourcesdir</value>
```



```
  <property>
    <name>hive.downloaded.resources.dir</name>
    <value>/home/hadoopData/hive/resourcesdir</value>
    <description>Temporary local directory for added resources in th
e remote file system.</description>
  </property>
```

Hive 运行时结构化日志文件的位置

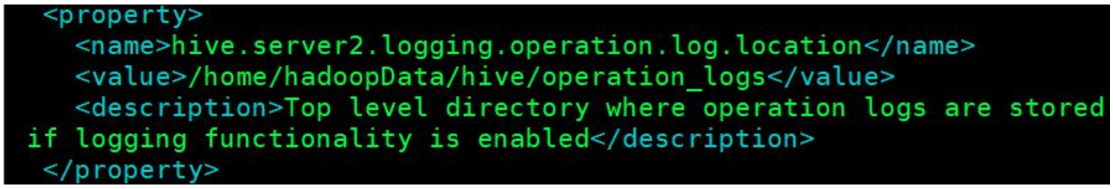
```
<name>hive.querylog.location</name>

<value>/home/hadoopData/hive/querylog</value>
```



```
  <property>
    <name>hive.querylog.location</name>
    <value>/home/hadoopData/hive/querylog</value>
    <description>Location of Hive run time structured log file</desc
ription>
  </property>
```

存储操作日志的顶级目录

```
<name>hive.server2.logging.operation.log.location</name>
<value>/home/hadoopData/hive/operation_logs</value>
```



```
  <property>
    <name>hive.server2.logging.operation.log.location</name>
    <value>/home/hadoopData/hive/operation_logs</value>
    <description>Top level directory where operation logs are stored
 if logging functionality is enabled</description>
  </property>
```

6）配置 hive-env.sh 配置文件

因为 hive-env.sh 配置文件也是不存在的，我们可以模板文件复制一份

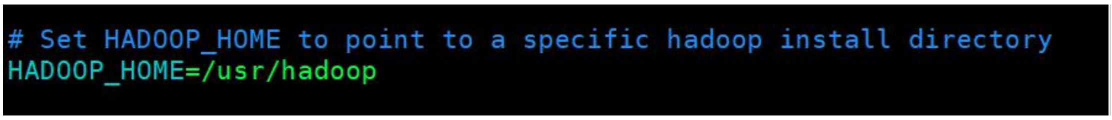
```
[hadoop@master ~]$ sudo cp  $HIVE_HOME/conf/hive-env.sh.template
$HIVE_HOME/conf/hive-env.sh
```

编辑 hive-env.sh 文件

```
[hadoop@master ~]$ sudo vim $HIVE_HOME/conf/hive-env.sh
```

修改配置文件中的 HADOOP_HOME

```
HADOOP_HOME=/usr/hadoop
```



```
# Set HADOOP_HOME to point to a specific hadoop install directory
HADOOP_HOME=/usr/hadoop
```

3. 配置驱动文件

将 mysql 驱动文件放到$HIVE_HOME/lib 目录下

```
[hadoop@master ~]$ sudo cp mysql-connector-java.jar $HIVE_HOME/lib
```

将 jline-2.12.jar 放到$HADOOP_HOME/lib 目录下

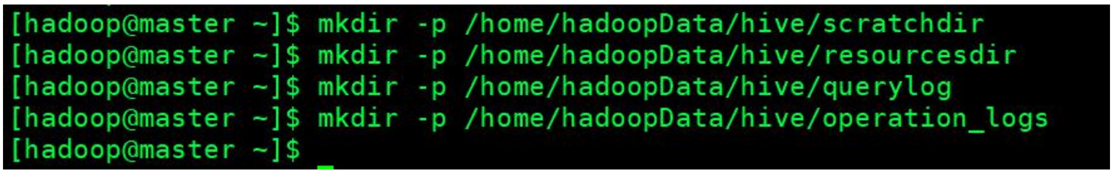
```
[hadoop@master ~]$ sudo cp $HIVE_HOME/lib/jline-2.12.jar $HADOOP_HOME/lib
```

将$HIVE_HOME/lib/log4j-slf4j-impl-2.10.0.jar 删除

```
[hadoop@master ~]$ sudo rm $HIVE_HOME/lib/log4j-slf4j-impl-2.10.0.jar
```

4. 创建 hive 数据存放相关目录

```
[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/scratchdir

[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/resourcesdir

[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/querylog

[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/operation_logs
```



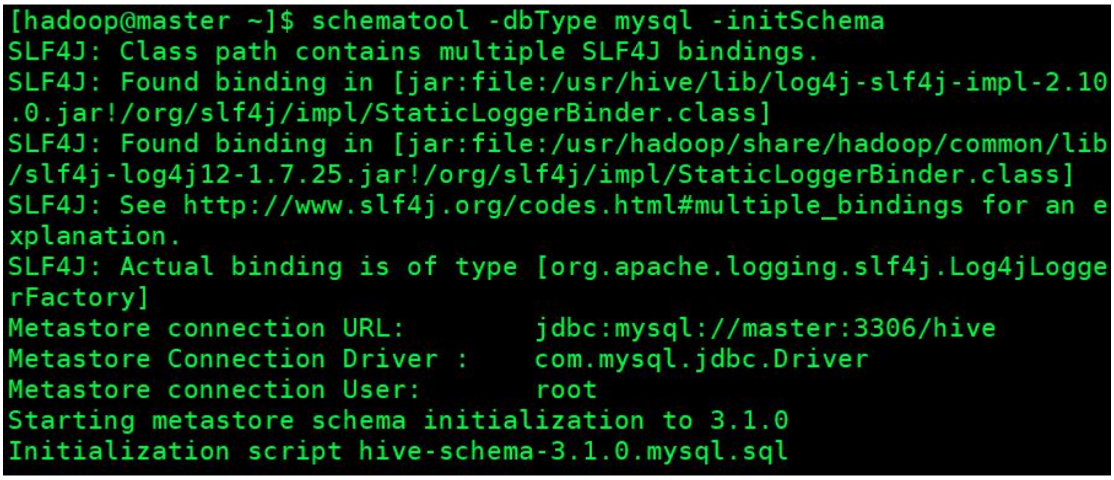
```
[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/scratchdir
[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/resourcesdir
[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/querylog
[hadoop@master ~]$ mkdir -p /home/hadoopData/hive/operation_logs
[hadoop@master ~]$ 
```

5. 修改 hive 的属主权限

```
[hadoop@master ~]$ sudo chown -R hadoop:hadoop /usr/hive
```

6. 初始化 hive

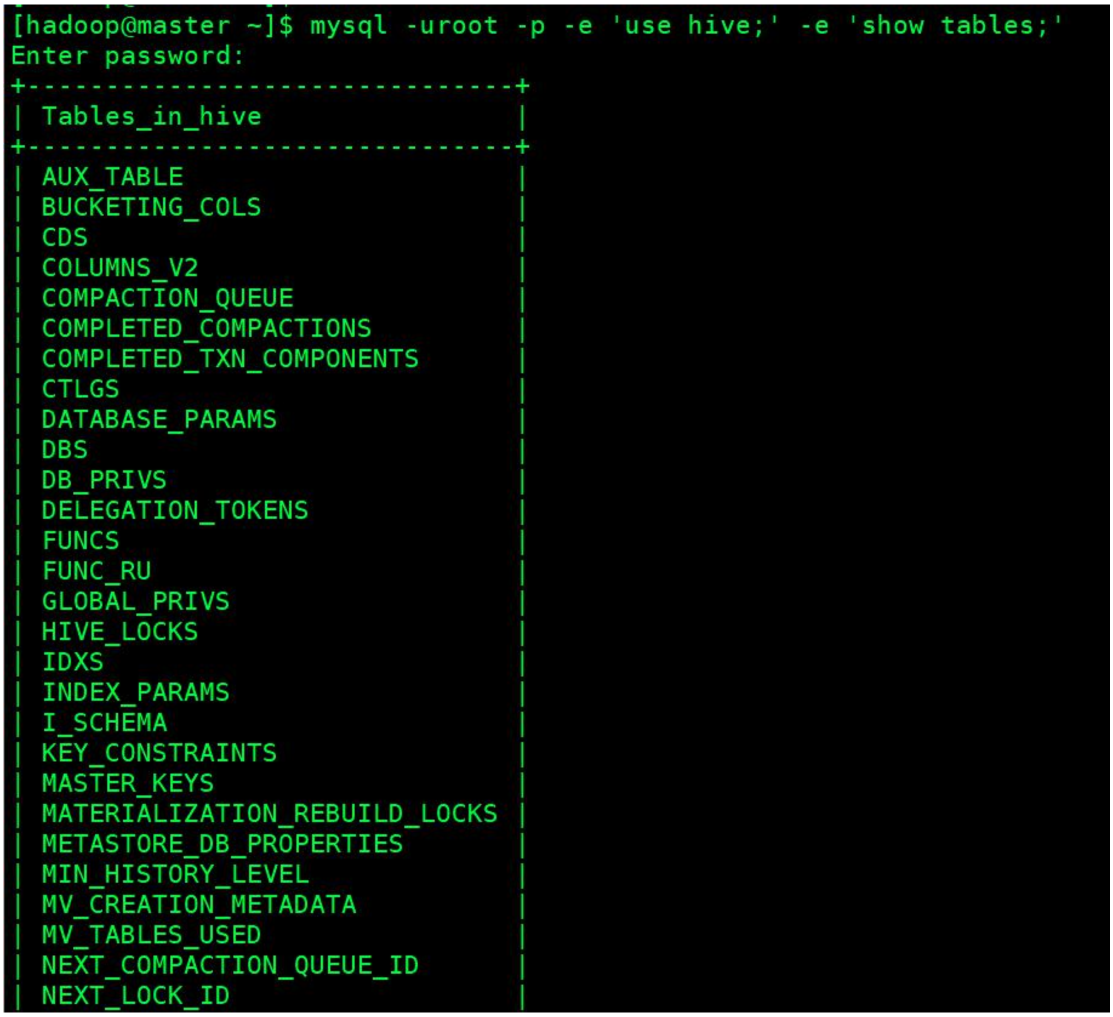
```
schematool -dbType mysql -initSchema
```



```
[hadoop@master ~]$ schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hive/lib/log4j-slf4j-impl-2.10
.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib
/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an e
xplanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLogge
rFactory]
Metastore connection URL:        jdbc:mysql://master:3306/hive
Metastore Connection Driver :    com.mysql.jdbc.Driver
Metastore connection User:       root
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.mysql.sql
```

查看 hive 数据库中的元数据表

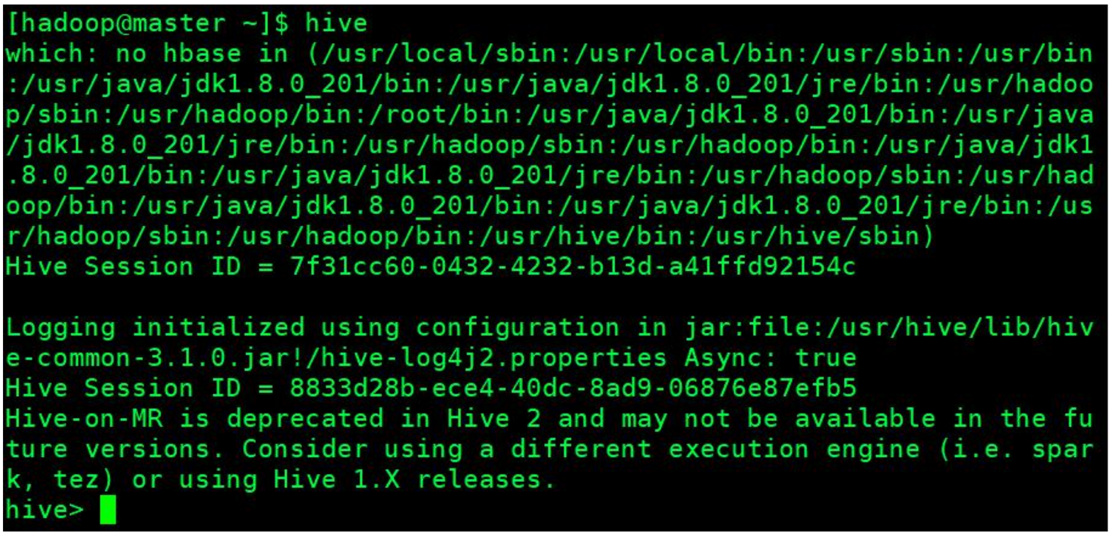
```
[hadoop@master ~]$ mysql -uroot -p -e 'use hive;' -e 'show tables;'
```



```
[hadoop@master ~]$ mysql -uroot -p -e 'use hive;' -e 'show tables;'
Enter password:
+---------------------------------+
| Tables_in_hive                  |
+---------------------------------+
| AUX_TABLE                       |
| BUCKETING_COLS                  |
| CDS                             |
| COLUMNS_V2                      |
| COMPACTION_QUEUE                |
| COMPLETED_COMPACTIONS           |
| COMPLETED_TXN_COMPONENTS        |
| CTLGS                           |
| DATABASE_PARAMS                 |
| DBS                             |
| DB_PRIVS                        |
| DELEGATION_TOKENS               |
| FUNCS                           |
| FUNC_RU                         |
| GLOBAL_PRIVS                    |
| HIVE_LOCKS                      |
| IDXS                            |
| INDEX_PARAMS                    |
| I_SCHEMA                        |
| KEY_CONSTRAINTS                 |
| MASTER_KEYS                     |
| MATERIALIZATION_REBUILD_LOCKS   |
| METASTORE_DB_PROPERTIES         |
| MIN_HISTORY_LEVEL               |
| MV_CREATION_METADATA            |
| MV_TABLES_USED                  |
| NEXT_COMPACTION_QUEUE_ID        |
| NEXT_LOCK_ID                    |
```

7. 启动 hive

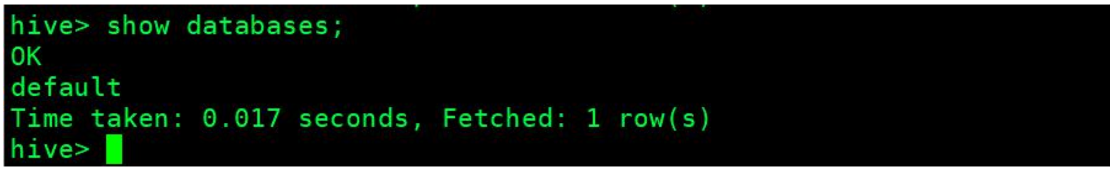
```
[hadoop@master ~]$ hive
```



```
[hadoop@master ~]$ hive
which: no hbase in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
:/usr/java/jdk1.8.0_201/bin:/usr/java/jdk1.8.0_201/jre/bin:/usr/hadoo
p/sbin:/usr/hadoop/bin:/root/bin:/usr/java/jdk1.8.0_201/bin:/usr/java
/jdk1.8.0_201/jre/bin:/usr/hadoop/sbin:/usr/hadoop/bin:/usr/java/jdk1
.8.0_201/bin:/usr/java/jdk1.8.0_201/jre/bin:/usr/hadoop/sbin:/usr/had
oop/bin:/usr/java/jdk1.8.0_201/bin:/usr/java/jdk1.8.0_201/jre/bin:/us
r/hadoop/sbin:/usr/hadoop/bin:/usr/hive/bin:/usr/hive/sbin)
Hive Session ID = 7f31cc60-0432-4232-b13d-a41ffd92154c

Logging initialized using configuration in jar:file:/usr/hive/lib/hiv
e-common-3.1.0.jar!/hive-log4j2.properties Async: true
Hive Session ID = 8833d28b-ece4-40dc-8ad9-06876e87efb5
Hive-on-MR is deprecated in Hive 2 and may not be available in the fu
ture versions. Consider using a different execution engine (i.e. spar
k, tez) or using Hive 1.X releases.
hive>
```

查看数据库：

```
hive> show databases;
```



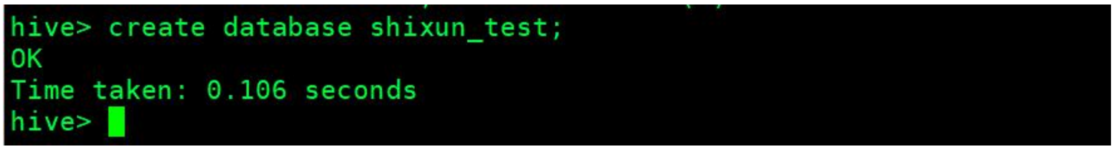
```
hive> show databases;
OK
default
Time taken: 0.017 seconds, Fetched: 1 row(s)
hive> 
```

三、hive shell 命令实操

1. 创建数据库

创建数据库 shixun_test

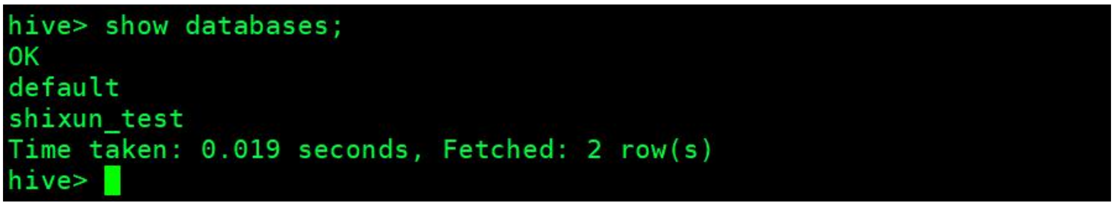
```
hive> create database shixun_test;
```



```
hive> create database shixun_test;
OK
Time taken: 0.106 seconds
hive> 
```

2. 查看数据库

```
hive> show databases;
```

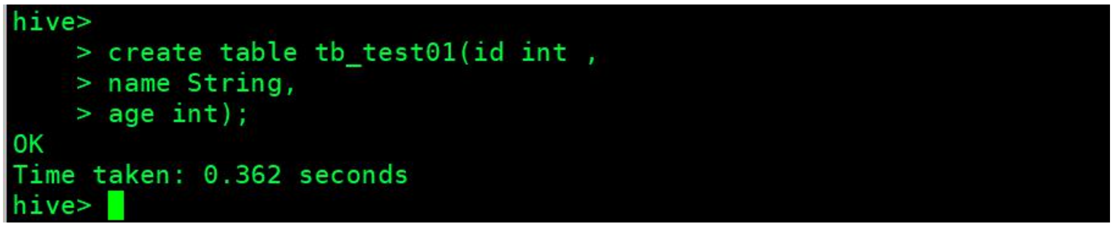


```
hive> show databases;
OK
default
shixun_test
Time taken: 0.019 seconds, Fetched: 2 row(s)
hive> 
```

3. 创建数据表

在 shixun_test 库中创建表 tb_test01

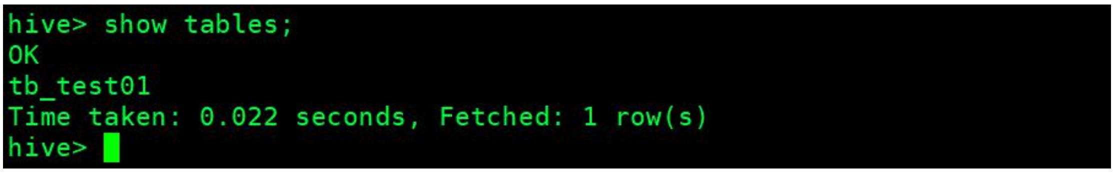
```
hive>

    > create table tb_test01(id int ,

    > name String,

    > age int);
```



```
hive>
    > create table tb_test01(id int ,
    > name String,
    > age int);
OK
Time taken: 0.362 seconds
hive>
```

4. 查看数据表

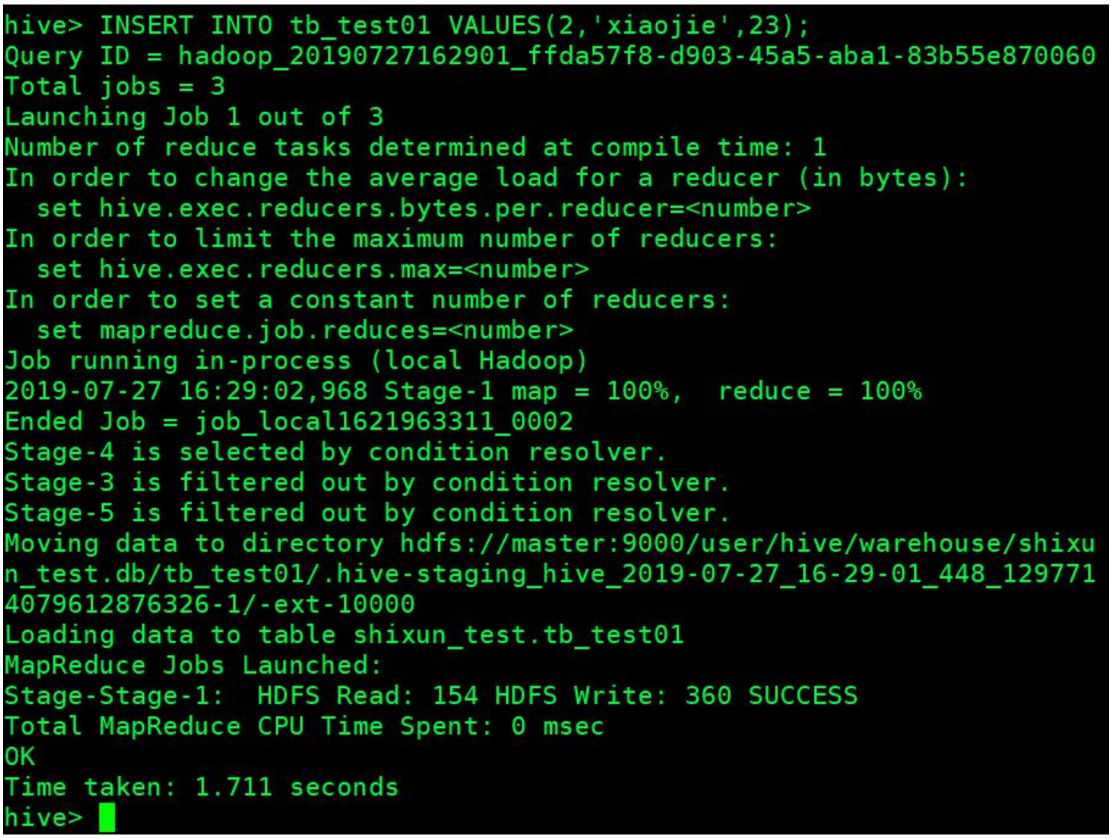
```
hive> show tables;
```



```
hive> show tables;
OK
tb_test01
Time taken: 0.022 seconds, Fetched: 1 row(s)
hive>
```

5. 插入数据

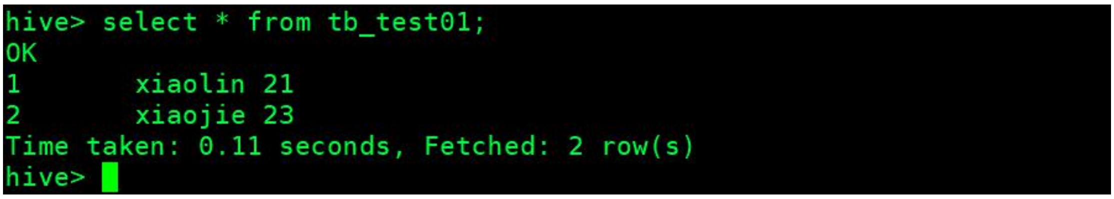
```
hive> INSERT INTO tb_test01 VALUES(2,xiaolin,21);

hive> INSERT INTO tb_test01 VALUES(2,'xiaojie',23);
```



```
hive> INSERT INTO tb_test01 VALUES(2,'xiaojie',23);
Query ID = hadoop_20190727162901_ffda57f8-d903-45a5-aba1-83b55e870060
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2019-07-27 16:29:02,968 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1621963311_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://master:9000/user/hive/warehouse/shixu
n_test.db/tb_test01/.hive-staging_hive_2019-07-27_16-29-01_448_129771
4079612876326-1/-ext-10000
Loading data to table shixun_test.tb_test01
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 154 HDFS Write: 360 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.711 seconds
hive>
```

6. 查看表数据

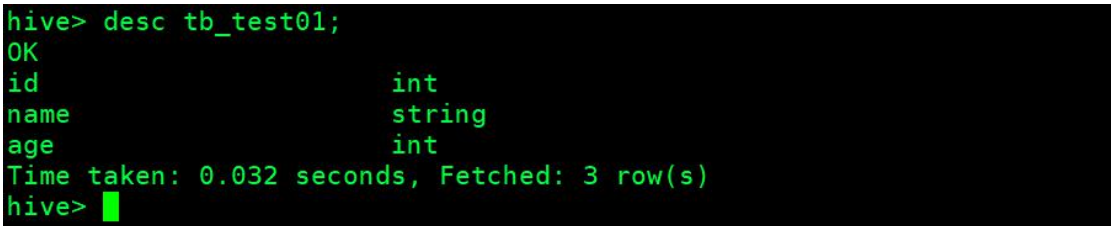
```
hive> select * from tb_test01;
```



```
hive> select * from tb_test01;
OK
1       xiaolin 21
2       xiaojie 23
Time taken: 0.11 seconds, Fetched: 2 row(s)
hive> 
```

7. 查看表结构

```
hive> desc tb_test01;
```



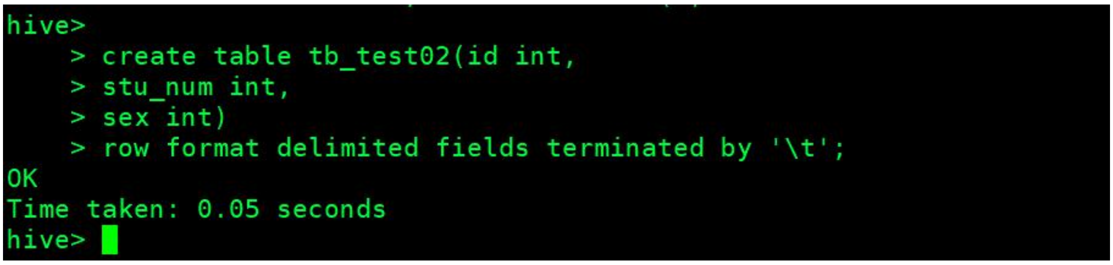
```
hive> desc tb_test01;
OK
id                      int
name                    string
age                     int
Time taken: 0.032 seconds, Fetched: 3 row(s)
hive> 
```

8. 数据导入

1）从本地文件系统中导入数据

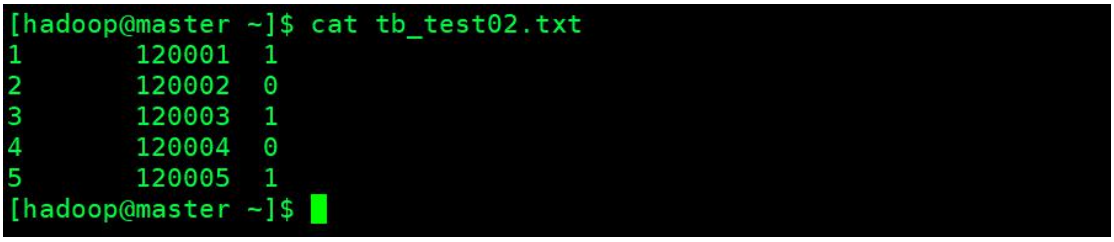
创建表 tb_test02：

```
hive> create table tb_test02(id int,
   > stu_num int,
   > sex int)
   > row format delimited fields terminated by '\t';
```



```
hive>
    > create table tb_test02(id int,
    > stu_num int,
    > sex int)
    > row format delimited fields terminated by '\t';
OK
Time taken: 0.05 seconds
hive> 
```

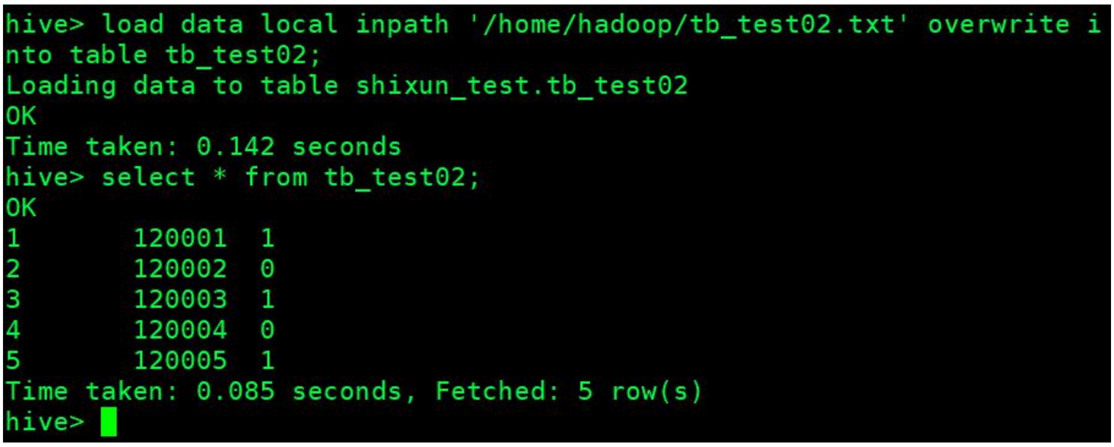
本地文件数据：



```
[hadoop@master ~]$ cat tb_test02.txt
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
[hadoop@master ~]$ 
```

加载数据：

```
hive> load data local inpath '/home/hadoop/tb_test02.txt' overwrite into table
tb_test02;
```

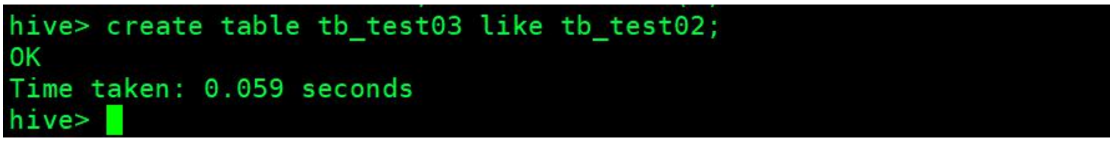


```
hive> load data local inpath '/home/hadoop/tb_test02.txt' overwrite i
nto table tb_test02;
Loading data to table shixun_test.tb_test02
OK
Time taken: 0.142 seconds
hive> select * from tb_test02;
OK
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
Time taken: 0.085 seconds, Fetched: 5 row(s)
hive> 
```

2）从 HDFS 文件系统中导入数据

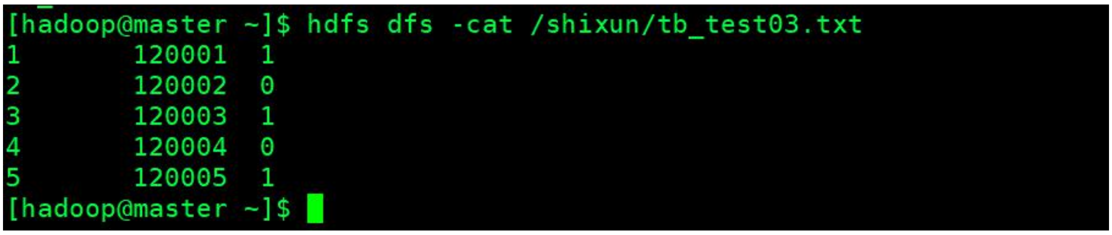
复制表结构，不复制表数据：

```
hive> create table tb_test03 like tb_test02;
```



```
hive> create table tb_test03 like tb_test02;
OK
Time taken: 0.059 seconds
hive> 
```

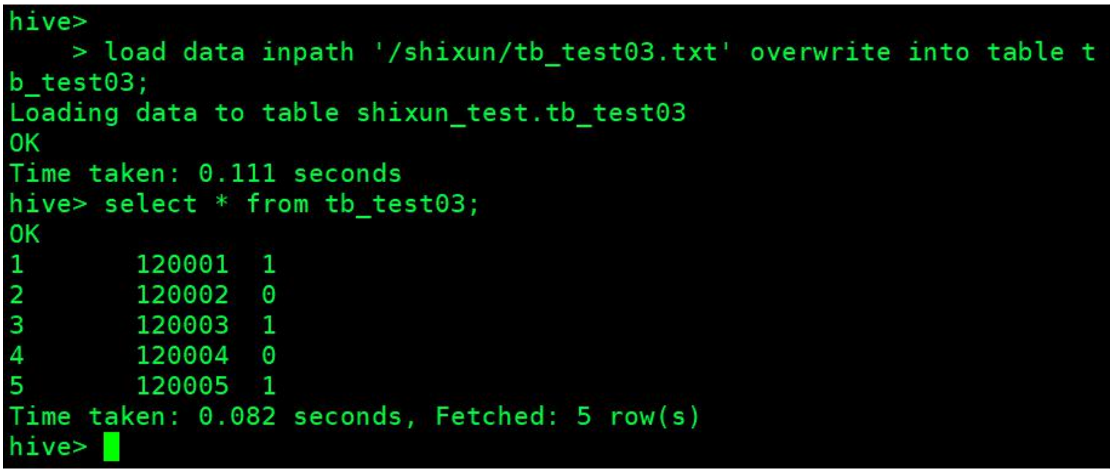
HDFS 文件系统上面 tb_test03.txt 数据



```
[hadoop@master ~]$ hdfs dfs -cat /shixun/tb_test03.txt
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
[hadoop@master ~]$ 
```

将 HDFS 文件系统上 tb_test03.txt 的数据导到 hive 表中：

```
hive>  load data inpath '/shixun/tb_test03.txt' overwrite into table tb_test03;
```

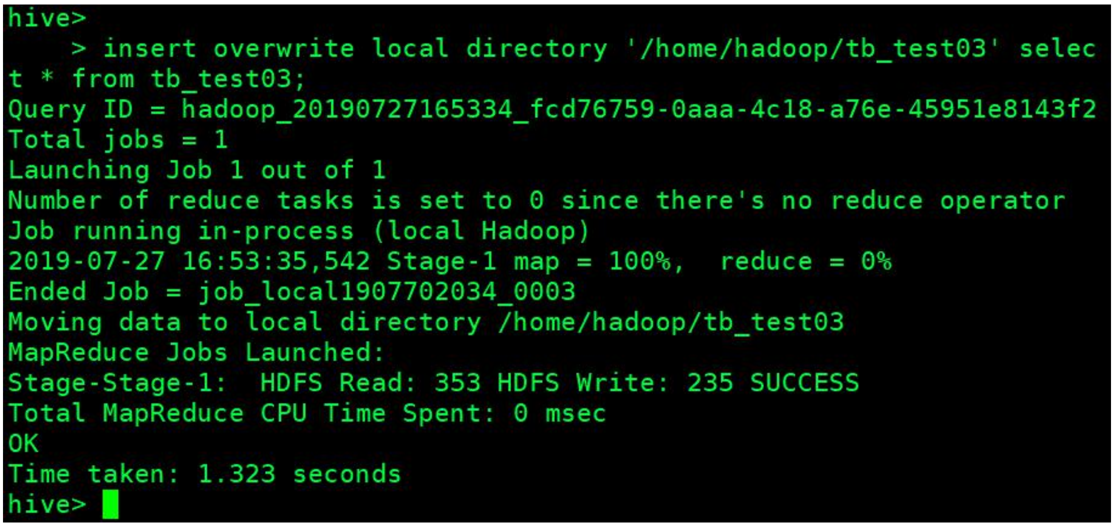


```
hive>
    > load data inpath '/shixun/tb_test03.txt' overwrite into table t
b_test03;
Loading data to table shixun_test.tb_test03
OK
Time taken: 0.111 seconds
hive> select * from tb_test03;
OK
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
Time taken: 0.082 seconds, Fetched: 5 row(s)
hive>
```

9. 数据导出

1）通过 hive 导出到本地文件系统：

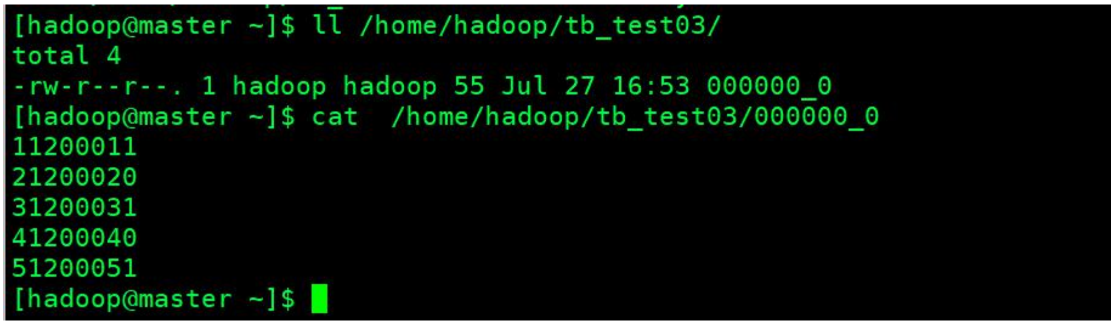
```
hive> insert overwrite local directory '/home/hadoop/tb_test03' select * from
tb_test03;
```



```
hive>
    > insert overwrite local directory '/home/hadoop/tb_test03' selec
t * from tb_test03;
Query ID = hadoop_20190727165334_fcd76759-0aaa-4c18-a76e-45951e8143f2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2019-07-27 16:53:35,542 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local1907702034_0003
Moving data to local directory /home/hadoop/tb_test03
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 353 HDFS Write: 235 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.323 seconds
hive>
```

查看 tb_test03 文件夹的数据：

```
[hadoop@master ~]$ cat /home/hadoop/tb_test03

[hadoop@master ~]$ cat  /home/hadoop/tb_test03/000000_0
```



```
[hadoop@master ~]$ ll /home/hadoop/tb_test03/
total 4
-rw-r--r--. 1 hadoop hadoop 55 Jul 27 16:53 000000_0
[hadoop@master ~]$ cat  /home/hadoop/tb_test03/000000_0
11200011
21200020
31200031
41200040
51200051
[hadoop@master ~]$
```