Sqoop

实验目的

熟练使用sqoop在mysql、hive、hbase、hdfs之间传输数据。

实验内容

安装，配置Sqoop

```
#!/bin/bash
if [ -d '/apps/sqoop' ];then
    sudo rm -rf /apps/sqoop
fi
# 将Sqoop的安装包复制到/apps下，并解压
sudo cp ~/big_data_tools/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz /apps/
tar xzvf /apps/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz -C /apps/
# 删除压缩包
sudo rm -rf /apps/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
# 修改名称
mv /apps/sqoop-1.4.7.bin__hadoop-2.6.0 /apps/sqoop
# 删除有关sqoop_home的环境变量
sed -i '/SQOOP_HOME/d' ~/.bashrc
# 添加环境变量
echo 'export SQOOP_HOME=/apps/sqoop' >> ~/.bashrc
echo 'export PATH=$SQOOP_HOME/bin:$PATH' >> ~/.bashrc
# 在导数据时，会涉及连接mysql，将mysql驱jar包导入导Lib目录下
cp ~/big_data_tools/mysql-connector-java-5.1.46-bin.jar /apps/sqoop/lib/
# 重命令
mv /apps/sqoop/conf/sqoop-env-template.sh sqoop-env.sh
# 在/apps/sqoop/conf/sqoop-env.sh中添加环境变量的相关信息
echo 'HADOOP_COMMON_HOME=/apps/hadoop' >> /apps/sqoop/conf/sqoop-env.sh
echo 'HADOOP_MAPRED_HOME=/apps/hadoop' >> /apps/sqoop/conf/sqoop-env.sh
echo 'HBASE_HOME=/apps/hbase' >> /apps/sqoop/conf/sqoop-env.sh
echo 'HIVE_HOME=/apps/hive' >> /apps/sqoop/conf/sqoop-env.sh
# 注释掉/apps/sqoop/bin/configure-sqoop128到147行的内容
sed -i "128,147s/^/#/g" /apps/sqoop/bin/configure-sqoop
# 重启，使环境变量生效
sudo reboot
```

版本信息：

```
chen@ubuntu:/apps/sqoop/bin$ sqoop version
/apps/hadoop/libexec/hadoop-functions.sh: line 2326: HADOOP_ORG.APACHE.SQOOP.SQOOP_US
ER: bad substitution
/apps/hadoop/libexec/hadoop-functions.sh: line 2421: HADOOP_ORG.APACHE.SQOOP.SQOOP_OP
TS: bad substitution
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/apps/hbase/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2020-11-10 19:30:45,012 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
Sqoop 1.4.7
git commit id 2328971411f57f0cb683dfb79d19d4d19d185dd8
Compiled by maugli on Thu Dec 21 15:59:58 STD 2017
```

使用 sqoop help 查看支持的命令

```
chen@ubuntu:/apps/sqoop/bin$ sqoop help
/apps/hadoop/libexec/hadoop-functions.sh: line 2326: HADOOP_ORG.APACHE.SQOOP.SQOOP_US
ER: bad substitution
/apps/hadoop/libexec/hadoop-functions.sh: line 2421: HADOOP_ORG.APACHE.SQOOP.SQOOP_OP
TS: bad substitution
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/apps/hbase/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2020-11-10 19:31:59,836 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
usage: sqoop COMMAND [ARGS]

Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  import-mainframe   Import datasets from a mainframe server to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information

See 'sqoop help COMMAND' for information on a specific command.
```

使用 sqoop help COMMAND 显示具体命令的信息

```
chen@ubuntu:/apps/sqoop/bin$ sqoop help import
/apps/hadoop/libexec/hadoop-functions.sh: line 2326: HADOOP_ORG.APACHE.SQOOP.SQOOP_US
ER: bad substitution
/apps/hadoop/libexec/hadoop-functions.sh: line 2421: HADOOP_ORG.APACHE.SQOOP.SQOOP_OP
TS: bad substitution
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/apps/hbase/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2020-11-10 19:33:00,242 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
usage: sqoop import [GENERIC-ARGS] [TOOL-ARGS]

Common arguments:
   --connect <jdbc-uri>                                     Specify JDBC
                                                            connect
                                                            string
   --connection-manager <class-name>                        Specify
                                                            connection
                                                            manager
                                                            class name
```

确保mysql正常运行

```
chen@ubuntu:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled
   Active: active (running) since Tue 2020-11-10 19:04:40 PST; 30min ago
  Process: 848 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p
  Process: 722 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, st
 Main PID: 850 (mysqld)
    Tasks: 28 (limit: 4633)
   CGroup: /system.slice/mysql.service
           └─850 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Nov 10 19:04:31 ubuntu systemd[1]: Starting MySQL Community Server...
Nov 10 19:04:40 ubuntu systemd[1]: Started MySQL Community Server.
```

查询mysql中的数据库。

```
sqoop list-databases \
--connect jdbc:mysql://localhost:3306/ \
--username root \
--password 123456
```

```
chen@ubuntu:~$ sqoop list-databases \
> --connect jdbc:mysql://localhost:3306/ \
> --username root \
> --password 123456
/apps/hadoop/libexec/hadoop-functions.sh: line 2326: HADOOP_ORG.APACHE.SQOOP.SQOOP_US
ER: bad substitution
/apps/hadoop/libexec/hadoop-functions.sh: line 2421: HADOOP_ORG.APACHE.SQOOP.SQOOP_OP
TS: bad substitution
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/apps/hbase/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2020-11-10 19:38:23,255 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2020-11-10 19:38:23,330 WARN tool.BaseSqoopTool: Setting your password on the command
-line is insecure. Consider using -P instead.
2020-11-10 19:38:23,567 INFO manager.MySQLManager: Preparing to use a MySQL streaming
 resultset.
Tue Nov 10 19:38:23 PST 2020 WARN: Establishing SSL connection without server's ident
ity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ r
equirements SSL connection must be established by default if explicit option isn't se
t. For compliance with existing applications not using SSL the verifyServerCertificat
e property is set to 'false'. You need either to explicitly disable SSL by setting us
eSSL=false, or set useSSL=true and provide truststore for server certificate verifica
tion.
information_schema
hive
mysql
performance_schema
sys
```

使用Sqoop

```
mysql -uroot -p123456
```

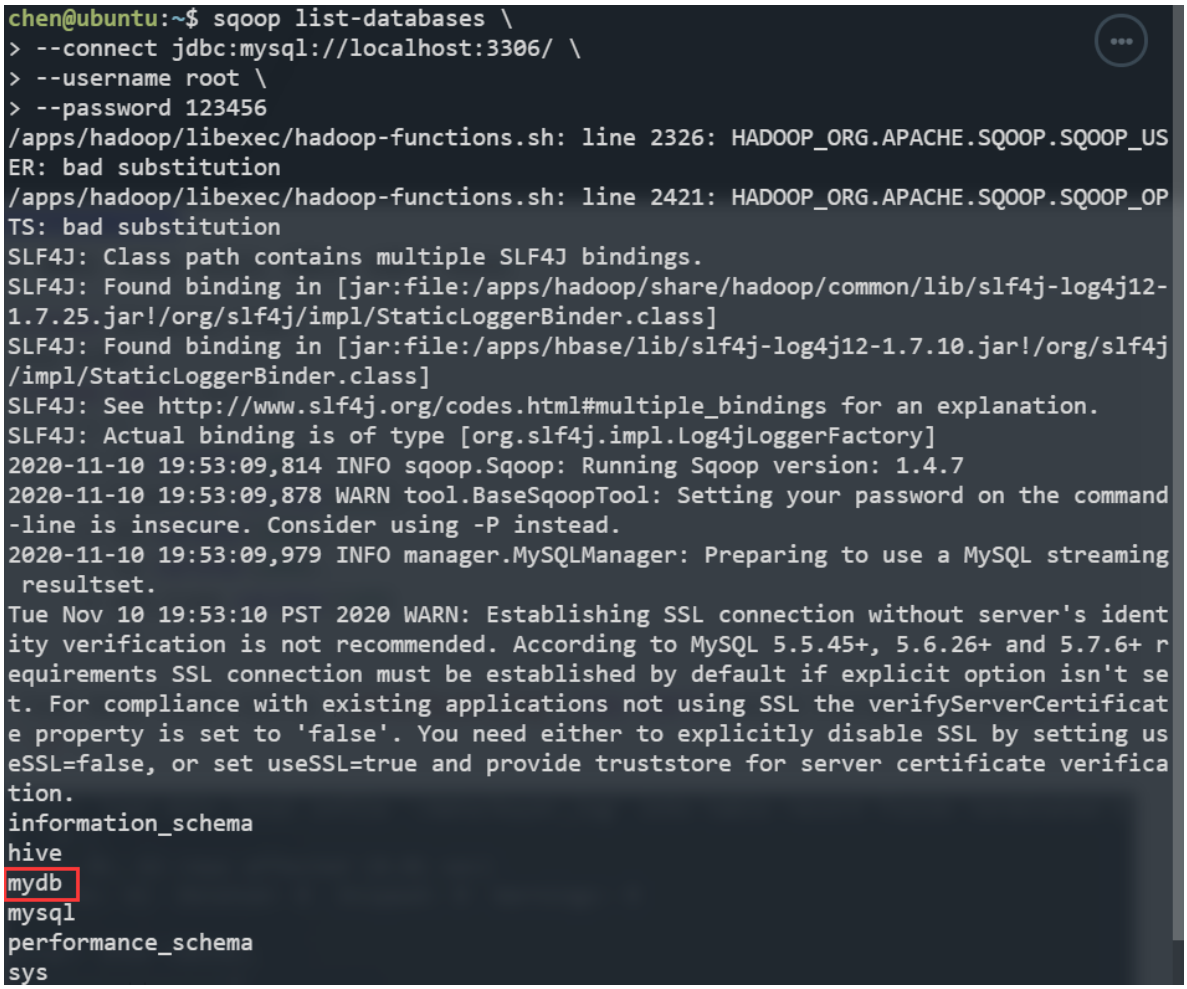
```
create database mydb;
use mydb;
create table record
    (
    id varchar(100),
    buyer_id varchar(100),
    dt varchar(100),
    ip varchar(100),
    opt_type varchar(100)
    );

load data local infile '/data/buyer_log' into table record fields terminated by
'\t';
```

```
mysql> load data local infile '/data/buyer_log' into table record fields terminated b
y '\t';
Query OK, 62 rows affected (0.01 sec)
Records: 62  Deleted: 0  Skipped: 0  Warnings: 0

mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| record         |
+----------------+
1 row in set (0.00 sec)
```

```
sqoop list-databases \
--connect jdbc:mysql://localhost:3306/ \
--username root \
--password 123456
```




```
sqoop list-tables \
--connect jdbc:mysql://localhost:3306/mydb \
--username root \
--password 123456
```

```
chen@ubuntu:~$ sqoop list-tables \
> --connect jdbc:mysql://localhost:3306/mydb \
> --username root \
> --password 123456
/apps/hadoop/libexec/hadoop-functions.sh: line 2326: HADOOP_ORG.APACHE.SQOOP.SQOOP_US
ER: bad substitution
/apps/hadoop/libexec/hadoop-functions.sh: line 2421: HADOOP_ORG.APACHE.SQOOP.SQOOP_OP
TS: bad substitution
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/apps/hbase/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2020-11-10 19:54:49,696 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2020-11-10 19:54:49,759 WARN tool.BaseSqoopTool: Setting your password on the command
-line is insecure. Consider using -P instead.
2020-11-10 19:54:49,847 INFO manager.MySQLManager: Preparing to use a MySQL streaming
 resultset.
Tue Nov 10 19:54:49 PST 2020 WARN: Establishing SSL connection without server's ident
ity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ r
equirements SSL connection must be established by default if explicit option isn't se
t. For compliance with existing applications not using SSL the verifyServerCertificat
e property is set to 'false'. You need either to explicitly disable SSL by setting us
eSSL=false, or set useSSL=true and provide truststore for server certificate verifica
tion.
record
chen@ubuntu:~$
```

将Mysql数据库中的数据存入到HDFS中

- 启动hadoop 【start-all.sh】

hdfs上的mysqool不能存在

```
sqoop import \
--connect jdbc:mysql://localhost:3306/mydb \
--username root \
--password 123456 \
--table record -m 1 \
--target-dir /mysqoop
```

```
chen@ubuntu:~$ hadoop fs -ls /mysqoop
Found 2 items
-rw-r--r--   1 chen supergroup          0 2020-11-10 20:45 /mysqoop/_SUCCESS
-rw-r--r--   1 chen supergroup       2948 2020-11-10 20:45 /mysqoop/part-m-00000
```

将HDFS中的数据存入到Mysql中(先在mysql创建表结构)

```
use mydb;
create table recordfromhdfs like record;
```

```
mysql> use mydb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table recordfromhdfs like record;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| record         |
| recordfromhdfs |
+----------------+
2 rows in set (0.00 sec)

mysql> exit;
Bye
```

```
sqoop export \
--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8 \
--username root \
--password 123456 \
--table recordfromhdfs -m 1 \
--export-dir hdfs://localhost:9000/mysqoop/part-m-00000
```

检查Mysql

```
use mydb;
select * from recordfromhdfs limit 3,5;
```

```
mysql> select * from recordfromhdfs limit 5 offset 3;
+------+----------+---------------------+-----------------+----------+
| id   | buyer_id | dt                  | ip              | opt_type |
+------+----------+---------------------+-----------------+----------+
| 465  | 20002    | 2010-03-30 10:56:35 | 222.44.94.235   | 2        |
| 466  | 20002    | 2010-03-30 10:56:35 | 222.44.94.235   | 1        |
| 481  | 10181    | 2010-03-31 16:48:43 | 123.127.164.252 | 1        |
| 482  | 10181    | 2010-04-01 17:35:05 | 123.127.164.252 | 1        |
| 483  | 10181    | 2010-04-02 10:34:20 | 123.127.164.252 | 1        |
+------+----------+---------------------+-----------------+----------+
5 rows in set (0.00 sec)

mysql> select * from recordfromhdfs limit 3,5;
+------+----------+---------------------+-----------------+----------+
| id   | buyer_id | dt                  | ip              | opt_type |
+------+----------+---------------------+-----------------+----------+
| 465  | 20002    | 2010-03-30 10:56:35 | 222.44.94.235   | 2        |
| 466  | 20002    | 2010-03-30 10:56:35 | 222.44.94.235   | 1        |
| 481  | 10181    | 2010-03-31 16:48:43 | 123.127.164.252 | 1        |
| 482  | 10181    | 2010-04-01 17:35:05 | 123.127.164.252 | 1        |
| 483  | 10181    | 2010-04-02 10:34:20 | 123.127.164.252 | 1        |
+------+----------+---------------------+-----------------+----------+
5 rows in set (0.00 sec)
```

将MySQL中数据导入到HBase(先在HBase中创建表结构)

启动HBase

```
start-hbase.sh
```

进入hbase shell

```
hbase shell
```

创建名为hbaserecord，有一个列族mycf的表，用来保存数据。

```
create 'hbaserecord','mycf'
```

```
hbase(main):001:0> create 'hbaserecord','mycf'
0 row(s) in 2.5580 seconds
```

```
sqoop import \
--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8 \
--username root \
--password 123456 \
--table record \
--hbase-table hbaserecord \
--column-family mycf \
--hbase-row-key dt -m 1
```

```
scan 'hbaserecord'
```

```
hbase(main):002:0> scan 'hbaserecord'
ROW                                   COLUMN+CELL
 2010-03-26 19:55:10                  column=mycf:buyer_id, timestamp=1605153456535, value=10262
 2010-03-26 19:55:10                  column=mycf:id, timestamp=1605153456535, value=462
 2010-03-26 19:55:10                  column=mycf:ip, timestamp=1605153456535, value=123.127.164.252
 2010-03-26 19:55:10                  column=mycf:opt_type, timestamp=1605153456535, value=1
 2010-03-29 14:28:02                  column=mycf:buyer_id, timestamp=1605153456535, value=20001
 2010-03-29 14:28:02                  column=mycf:id, timestamp=1605153456535, value=464
 2010-03-29 14:28:02                  column=mycf:ip, timestamp=1605153456535, value=221.208.129.117
 2010-03-29 14:28:02                  column=mycf:opt_type, timestamp=1605153456535, value=1
 2010-03-30 10:56:35                  column=mycf:buyer_id, timestamp=1605153456535, value=20002
 2010-03-30 10:56:35                  column=mycf:id, timestamp=1605153456535, value=466
 2010-03-30 10:56:35                  column=mycf:ip, timestamp=1605153456535, value=222.44.94.235
 2010-03-30 10:56:35                  column=mycf:opt_type, timestamp=1605153456535, value=1
 2010-03-31 16:48:43                  column=mycf:buyer_id, timestamp=1605153456535, value=10181
 2010-03-31 16:48:43                  column=mycf:id, timestamp=1605153456535, value=481
 2010-03-31 16:48:43                  column=mycf:ip, timestamp=1605153456535, value=123.127.164.252
```

将Mysql的数据导入到Hive(先在Hive中创建表结构)

需要在~/.bashrc 添加以下内容。

```
export HADOOP_CLASSPATH=$HADOOP_HOME/lib:$HIVE_HOME/lib/*
export HIVE_CONF_DIR=$HIVE_HOME/conf
```

```
source ~/.bashrc
```

```
create table hiverecord (id varchar(100),buyer_id varchar(100),dt
varchar(100),ip varchar(100),opt_type varchar(100)) row format delimited fields
terminated by ',' stored as textfile;
```

```
hive> create table hiverecord (id varchar(100),buyer_id varchar(100),dt varchar(100),
ip varchar(100),opt_type varchar(100)) row format delimited fields terminated by ','
stored as textfile;
OK
Time taken: 3.979 seconds
```

```
sqoop import \
--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8 \
--username root \
--password 123456 \
--table record \
--hive-import \
--hive-table hiverecord \
--fields-terminated-by ',' -m 1
```

执行上面命令失败，需要去hdfs，把/user/chen/record删掉。

```
select * from hiverecord
```

```
538     20049   2010-04-08 14:57:31     123.127.164.252 2
539     20049   2010-04-08 14:57:31     123.127.164.252 1
540     20050   2010-04-08 14:58:23     123.127.164.252 2
541     20050   2010-04-08 14:58:23     123.127.164.252 1
542     20051   2010-04-08 15:00:33     123.127.164.252 2
543     20051   2010-04-08 15:00:33     123.127.164.252 1
544     20052   2010-04-08 15:01:36     123.127.164.252 2
545     20052   2010-04-08 15:01:36     123.127.164.252 1
546     20047   2010-04-08 15:01:50     123.127.164.252 1
Time taken: 4.583 seconds, Fetched: 62 row(s)
```

将Hive表的数据导出到Mysql中（先在mysql创建表结构）

```
use mydb;
create table recordfromhive like record;
```

导数据

```
sqoop export \
--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8 \
--username root \
--password 123456 \
--table recordfromhive \
--export-dir /user/hive/warehouse/hiverecord/part-m-00000 \
--input-fields-terminated-by ',' -m 1
```

```
drwxr-xr-x   - chen supergroup          0 2020-11-11 20:29 /user/hive/warehouse
chen@ubuntu:~$ sqoop export \
> --connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8 \
> --username root \
> --password 123456 \
> --table recordfromhive \
> --export-dir /user/hive/warehouse/hiverecord/part-m-00000 \
> --input-fields-terminated-by ',' -m 1
```

查看结果

```
use mydb;
select * from recordfromhive;
```

```
| 541  | 20050    | 2010-04-08 14:58:23 | 123.127.164.252 | 1         |
| 542  | 20051    | 2010-04-08 15:00:33 | 123.127.164.252 | 2         |
| 543  | 20051    | 2010-04-08 15:00:33 | 123.127.164.252 | 1         |
| 544  | 20052    | 2010-04-08 15:01:36 | 123.127.164.252 | 2         |
| 545  | 20052    | 2010-04-08 15:01:36 | 123.127.164.252 | 1         |
| 546  | 20047    | 2010-04-08 15:01:50 | 123.127.164.252 | 1         |
+------+----------+---------------------+-----------------+-----------+
62 rows in set (0.00 sec)
```