实验3-2 hive

@author owen

1.Hive安装配置

> hive默认使用derby数据库存储其相关元数据，也可以改为使用mysql来存储hive的元数据信息，这里因为机器上刚好装了mysql，所以使用mysql

1.安装mysql

```
apt-get install mysql-server
apt-get install mysql-client
```

2.下载解压hive-3.1.2

3.下载mysql connector

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.48.tar.gz
解压以后放到hive-3.1.2/lib/ 下面
```

4.配置hive/conf

```
HADOOP_HOME=/home/Hadoop/hadoop-3.2.1
export HIVE_CONF_DIR=/home/Hadoop/hive-3.1.2/conf
export HIVE_AUX_JARS_PATH=/home/Hadoop/hive-3.1.2/lib
```

配置细节参见https://blog.csdn.net/weixin_43824520/article/details/100580557和https://blog.csdn.net/aguang_vip/article/details/81583661

（为了以后可能的调试方便，可以再配置一个hive的logs文件夹

```
配置日志，复制一个模板
cp hive-log4j2.properties.template hive-log4j2.properties
vi hive-log4j2.properties
配置property.hive.log.dir
property.hive.log.dir = /root/hive-3.1.0/logs（注意：logs需要自己创建，在hive目录下mkdir logs)
```

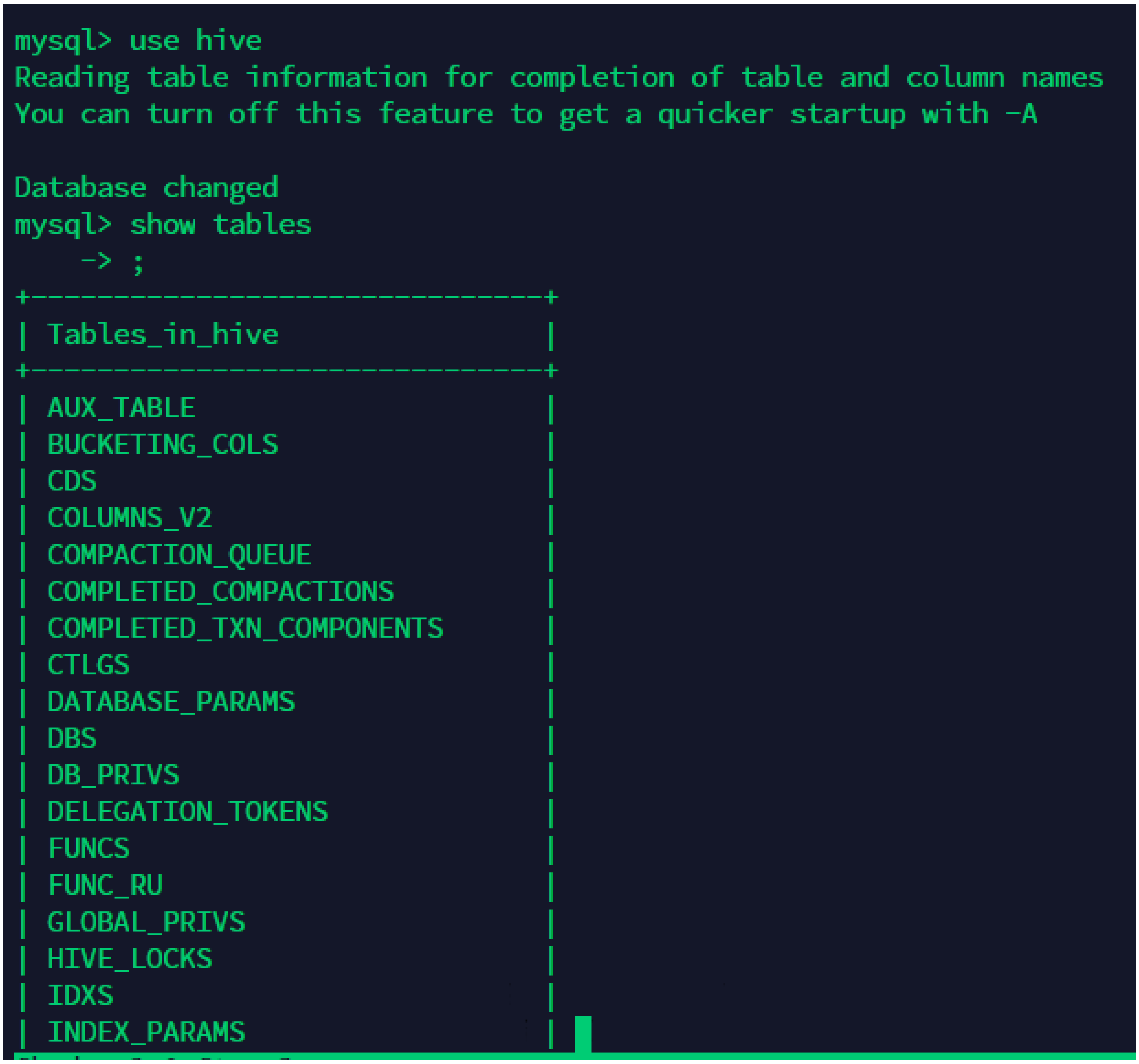
在hive配置连接mysql的时候遇到一个问题

1. 格式化hive时，报错communication link failure，同时发现登录mysql使用root用户时 不需要密码

   select user, host, plugin from mysql.user 发现root 的plugin为 'auth_sock', 需要修改为'mysql_native_password'. 同时记得用update mysql.user set authentication_string = pasword('设置的密码') where user='root' 这样登录mysql的时候就需要密码了

5.格式化hive

命令见上面网站，成功后，可以看到在mysql中生成了相关的表



```
mysql> use hive
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> show tables
    -> ;
+-------------------------------+
| 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                  |
```

可能的报错：java.lang.NoSuchMethodError: com.google.common.base.Preconditions.checkArgument

https://blog.csdn.net/GQB1226/article/details/102555820

ps：Hive使用默认derby保存元数据见教程

https://www.cnblogs.com/raphael5200/p/5177457.html

https://www.jianshu.com/p/6bfad788ab09

2. Hive操作

创建表格

```
create table tUser ( user_id int, item_id int, cat_id int, merchant_id int, brand_id int,
 month int, day int, action int, age_range int, gender int,  province string ) row format
delimited fields terminated by ',';

 -- row format delimited fields terminated by ','; 规定了文件中列的分隔符
 -- 如果不指定分隔符，导入数据在表中全部为ULL
```

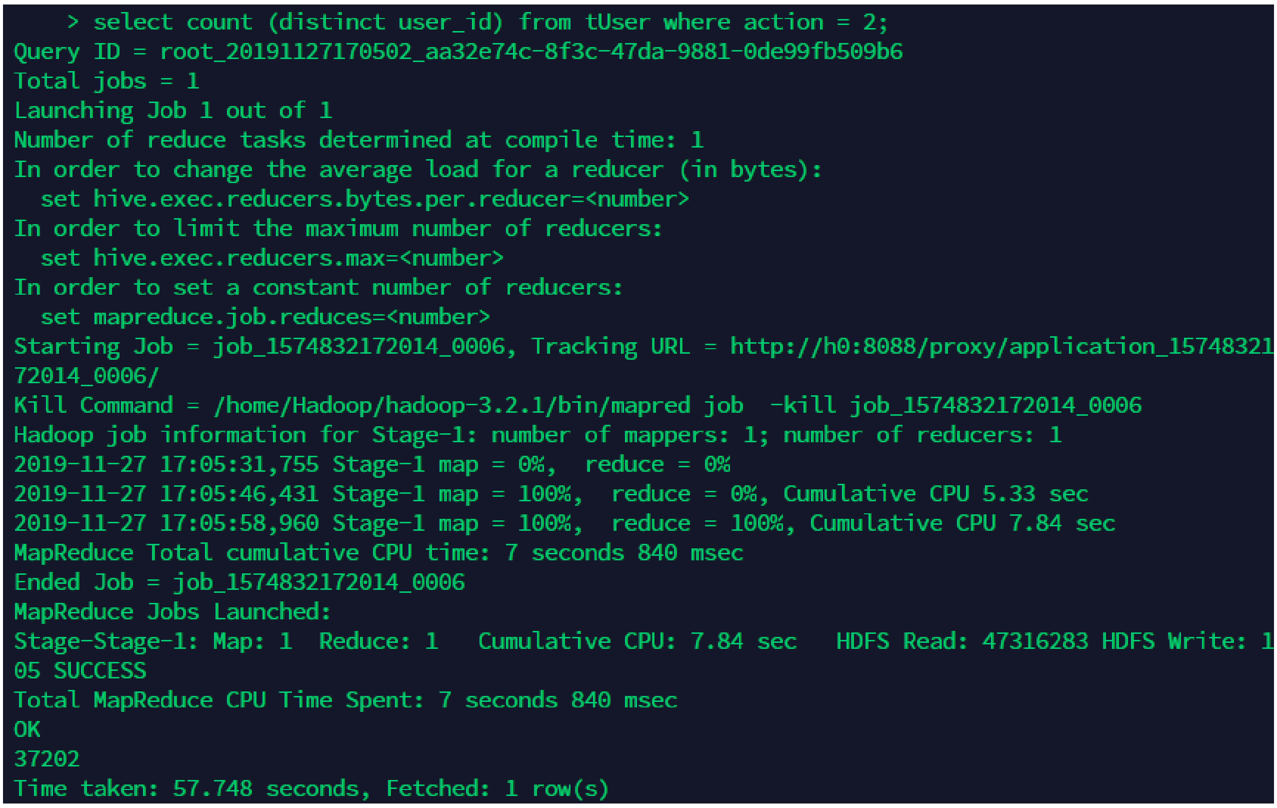
导入数据

```
load data local inpath '/home/Hadoop/share-files/million_user_log.csv' into table tUser;
```

- 查询双11那天有多少人购买了商品

  ```
  select count (distinct user_id) from tUser where action = 2;
  ```

  结果共37202人购买了商品

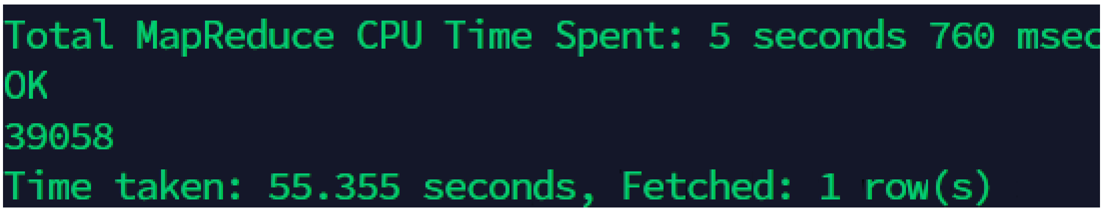
  

  ```
      > select count (distinct user_id) from tUser where action = 2;
  Query ID = root_20191127170502_aa32e74c-8f3c-47da-9881-0de99fb509b6
  Total jobs = 1
  Launching Job 1 out of 1
  Number of reduce tasks determined at compile time: 1
  In order to change the average load for a reducer (in bytes):
    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>
  Starting Job = job_1574832172014_0006, Tracking URL = http://h0:8088/proxy/application_15748321
  72014_0006/
  Kill Command = /home/Hadoop/hadoop-3.2.1/bin/mapred job  -kill job_1574832172014_0006
  Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
  2019-11-27 17:05:31,755 Stage-1 map = 0%,  reduce = 0%
  2019-11-27 17:05:46,431 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.33 sec
  2019-11-27 17:05:58,960 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.84 sec
  MapReduce Total cumulative CPU time: 7 seconds 840 msec
  Ended Job = job_1574832172014_0006
  MapReduce Jobs Launched:
  Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.84 sec   HDFS Read: 47316283 HDFS Write: 1
  05 SUCCESS
  Total MapReduce CPU Time Spent: 7 seconds 840 msec
  OK
  37202
  Time taken: 57.748 seconds, Fetched: 1 row(s)
  ```

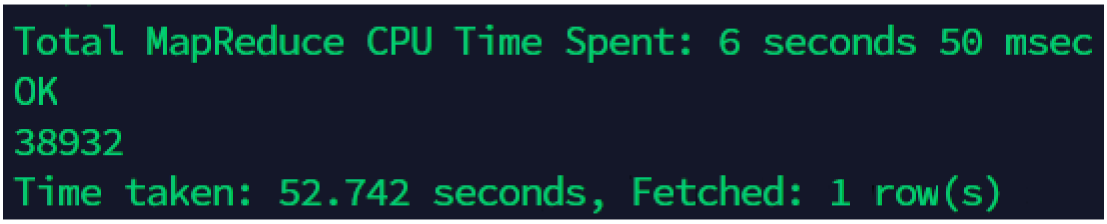
- 查询双11那天男女买家购买商品的比例

  ```
  select count (item_id) from tUser where action = 2 and gender = 0;
  select count (item_id) from tUser where action = 2 and gender = 1;
  ```

  男女比为 $\frac{男}{女} = \frac{38932}{39058} = 0.996774$

  ```
  Total MapReduce CPU Time Spent: 5 seconds 760 msec
  OK
  39058
  Time taken: 55.355 seconds, Fetched: 1 row(s)
  ```

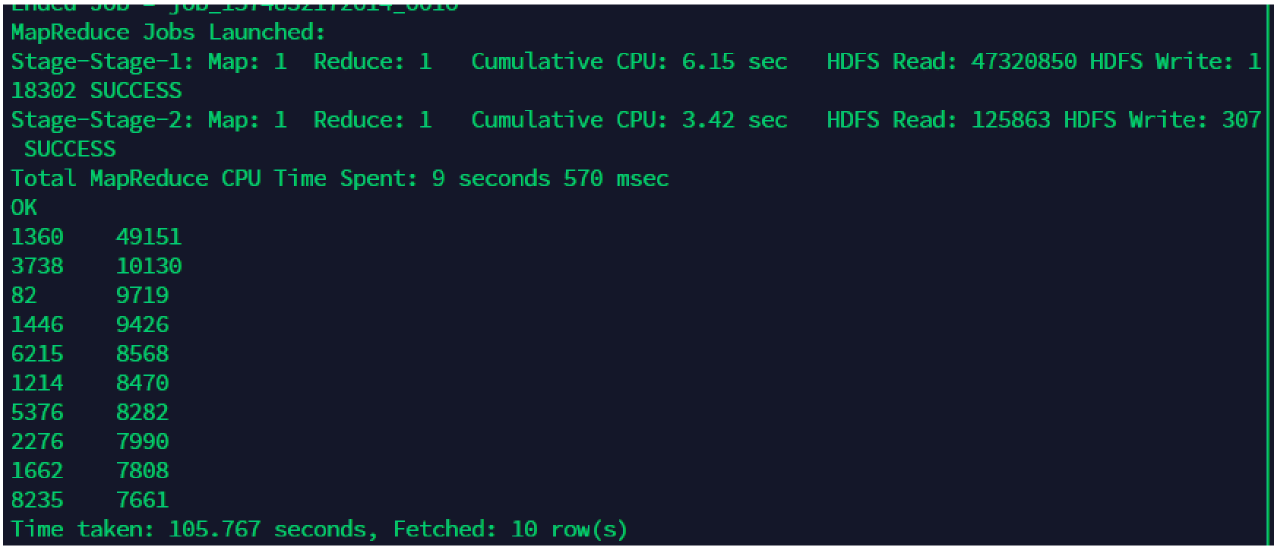
  

  ```
  Total MapReduce CPU Time Spent: 6 seconds 50 msec
  OK
  38932
  Time taken: 52.742 seconds, Fetched: 1 row(s)
  ```

- 查询双11那天浏览次数前十的品牌

```
select brand_id, count ( action ) actions from tUser where action = 0 group by brand_id
order by actions desc limit 10;
```

下图中 第一列是brand_id 第二列是这个品牌总点击量



```
Ended Job = job_1574852172014_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.15 sec   HDFS Read: 47320850 HDFS Write: 1
18302 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.42 sec   HDFS Read: 125863 HDFS Write: 307
 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 570 msec
OK
1360	49151
3738	10130
82	9719
1446	9426
6215	8568
1214	8470
5376	8282
2276	7990
1662	7808
8235	7661
Time taken: 105.767 seconds, Fetched: 10 row(s)
```