实验 3-1 Hive 加载电商源数据

建议课时：60 分钟

一、 实验目的

 掌握hive 建库、建表语句；

 掌握hive 加载数据的几种方式；

 熟练hive 表数据的常用查询语句；

二、 实验环境

Dsight 智慧实验室中的 hadoop 环境

三、 实验步骤

本节实验所做内容如下红色标注：

本节实验主要是通过Hive 加载本地源数据到 hdfs 分布式文件系统。

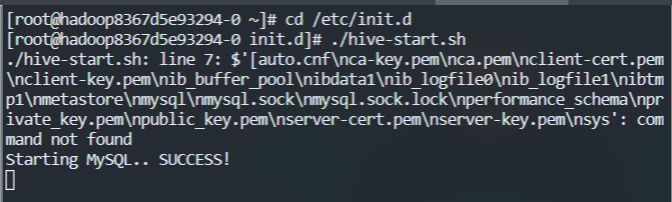
具体实验步骤如下：

1. 进入实验室，打开hadoop 环境

2. 启动hive 进程

第一次进入实验环境时启动方式：cd /etc/init.d运 行./hive-start.sh

之后进入实验室环境时输入hive，回车即可。



3. 创建数据库

创建自己的数据库zhaoteng，以队长名字命名：create database zhaoteng;

语句：

create database zhaoteng;

截图：

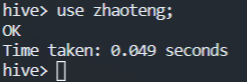


4. 使用数据库

语句：

use zhaoteng;

截图：



5. 创建表

创建用户基本信息表、手机销售信息表、用户评价表、用户行为表。

**①用户基本信息表：**

创建语句：

create table userInfo(

user\_id string, ##用户ID

user\_name string, ##用户名

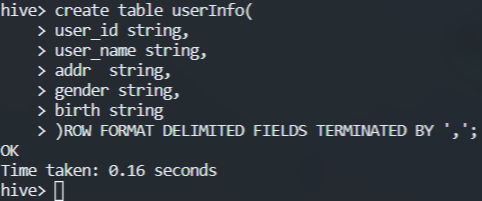
addr string, ##所在地

gender string, ##性别

birth string ##生日

)ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

截图：



**②手机销售信息表：**

建表语句：

create table iphone(

goods\_id string, ##商品ID

goods\_name string, ##商品名称

goods\_property string, ##商品基本属性

store\_name string, ##店铺名称

store\_id string, ##店铺ID

goods\_url string, ##商品链接

goods\_price float, ##价格

keyword string, ##搜索关键词

sail\_count int, ##评论数

good\_rate int, ##好评率

brand string, ##品牌

model string, ##型号

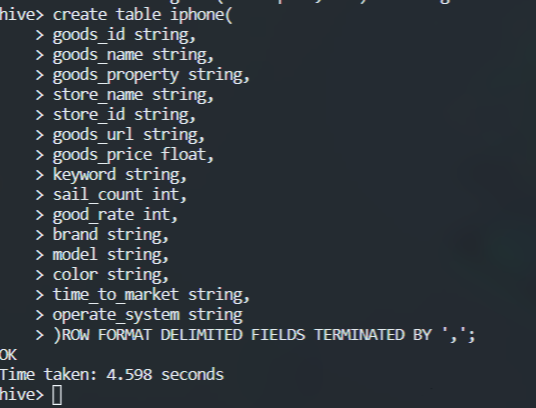
color string, ##机身颜色

time\_to\_market string, ##上市时间

operate\_system string ##操作系统

)ROW FORMAT DELIMITED FIELDS TERMINATED BY ','; ##各字段间以','号分割

截图：



**③用户评价表**

建表语句：

create table comment(

goods\_id string, ##商品ID

comment\_text int, ##评论正文

comment\_time string, ##发布时间

user\_id string, ##用户ID

user\_name string, ##用户名

user\_level string, ##用户等级

color string, ##商品颜色

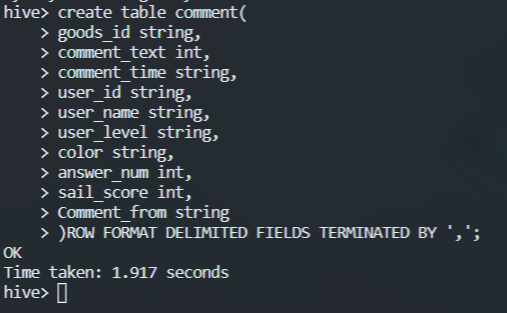
answer\_num int, ##回复数

sail\_score int, ##评价分数

Comment\_from string ##来源

)ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

截图：



**④用户行为表**

建表语句：

create table action(

user\_id string, ##用户ID

goods\_id string, ##商品ID

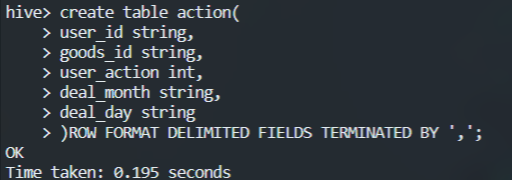
user\_action int, ##用户行为

deal\_month string, ##交易月份

deal\_day string ##交易日

)ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

截图：



1. 加载数据

通过hive 加载本地数据的方式依次加载上述表中的源数据。

①加载用户信息数据：

load data local inpath "/home/hadoop/文档/Day09/userinfo.csv" into table userinfo;

②加载用户行为数据：

load data local inpath "/home/hadoop/文档/Day09/action.csv" into table action;

③加载用户评价表：

load data local inpath "/home/hadoop/文档/Day09/comment.csv" into table comment;

④加载手机销售信息表：

load data local inpath "/home/hadoop/文档/Day09/iphone.csv" into table iphone;

7. 查询加载的数据

出现了中文乱码的情况，数据库建的时候默认为utf-8，在windows系统中中创建的文件，默认编码是GBK，所以将文件重新保存为utf-8，即可成功导入正确的数据。

8. 处理手机销售信息表数据

由于爬虫过程是通过关键词搜索获取的，所获取数据不仅包含手机的销售数据，此外，关于手机的部分外设（充电宝、数据线、手机膜、保护壳、耳机等）销售数据，需要将这部分数据清洗掉。观察表中数据可看出，所有有关手机的操作系统的列字段又有信息，所以可根据手机销售信息表中的操作系统字段进行筛选即，使用操作系统非空和商品name的内容包括‘手机’字段合并查询。

筛选手机的销售数据存储到新表goods\_sail\_info 中。

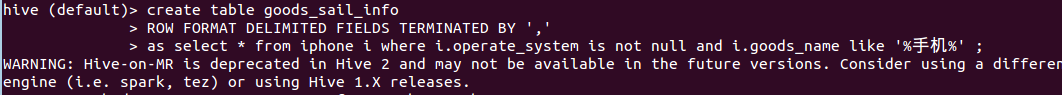
语句：

create table goods\_sail\_info

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

as select \* from iphone i where i.operate\_system is not null and i.goods\_name like '%手机%' ;

截图：



9. 处理用户行为表数据

**要求：**将用户行为源表中交易月份与交易日拼接为新字段存储到新表。

**思想：**使用concat\_ws函数以"-"拼接交易月份和交易日这两个字段，生成新的字段deal\_time

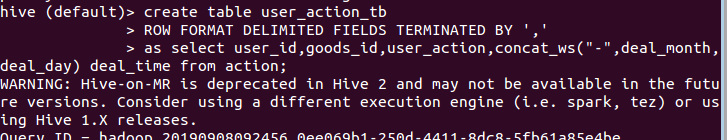
**新的用户行为表**

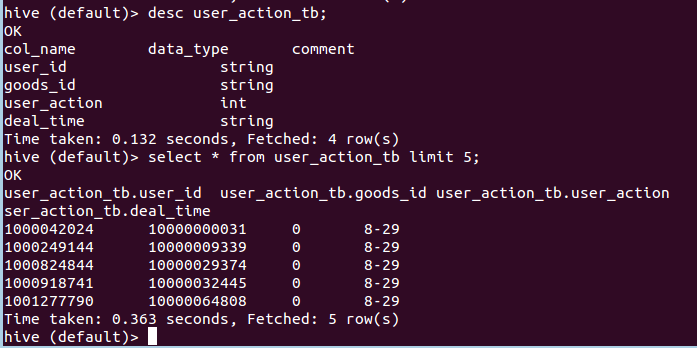
create table user\_action\_tb

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

as select user\_id,goods\_id,user\_action,concat\_ws("-",deal\_month,deal\_day) deal\_time from action;

**截图：**





10.处理用户信息表数据

要求：关联用户评论表和用户信息表将用户等级、用户年龄段划分后存

入新表。

1表示年龄<18

2表示年龄在[18,24]

3表示年龄在[25,29]

4表示年龄在[30,34]

5表示年龄在[35,39]

6表示年龄在[40,49]

7表示年龄>=50

新表数据字段说明：

**（user\_id、user\_name、addr、gender为原用户信息表的用户id、用户名，所在地，性别，生日另外user\_grade为评论表中的用户等级，所以需要以相同的用户id连接用户信息表和评论表的信息实现。该表中还添加了新的两列：age\_region表示用户年龄等级，用户年龄段的信息，通过用户信息表中的生日列获取用户出生的年份，然后获取到当前系统时间，使用字符串截取函数取出年份，两者差值即用户的年龄，然后使用case then 语句针对年龄字段进行不同条件的设置,得到新列的数值）注意：用户生日中还存在字段为null和无年份的情况：（10月20日），所以需要创建新表时注意过滤这类数据，可通过设置算出的年龄是否>=0来判断：**

**语句实现：**

create table user\_info\_new\_tb

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

as select u.user\_id user\_id ,u.user\_name user\_name,u.addr addr,u.gender gender,

case when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<18 then 1

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=18 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=24 then 2

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=25 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=29 then 3

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=30 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=34 then 4

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=35 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=39 then 5

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=40 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=49 then 6

else 7 end age\_region ,

case when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<18 then '18岁以下'

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=18 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=24 then '18-24岁'

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=25 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=29 then '25-29岁'

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=30 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=34 then '30-34岁'

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=35 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=39 then '35-39岁'

when (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=40 and (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))<=49 then '40-49岁'

else '50岁以上'

end age\_region\_alias ,

c.user\_level user\_grade from userinfo u join comment c on u.user\_id=c.user\_id where (cast(substring(current\_date,0,4) as int) - cast(substring(u.birth,0,4) as int))>=0;

截图：

