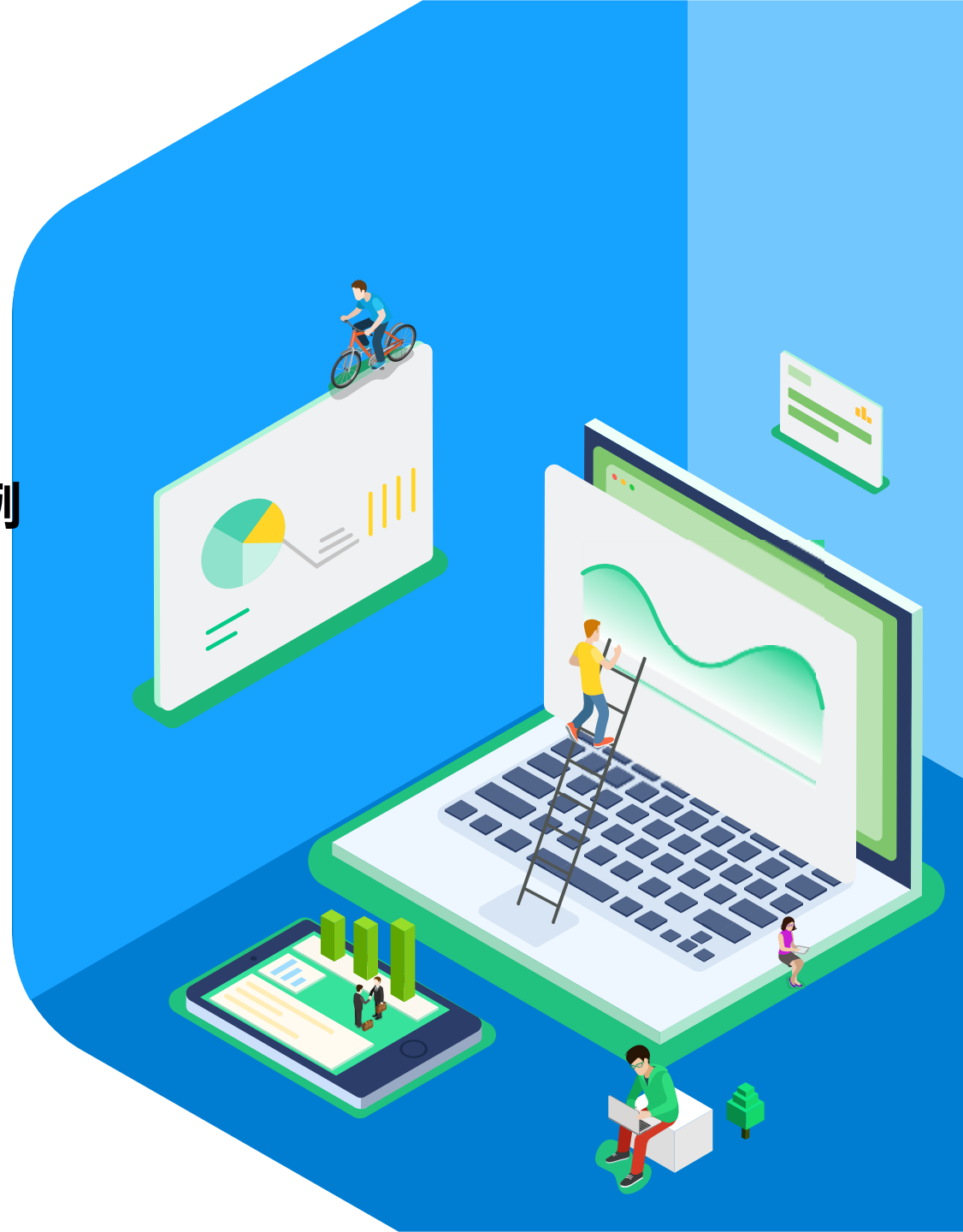
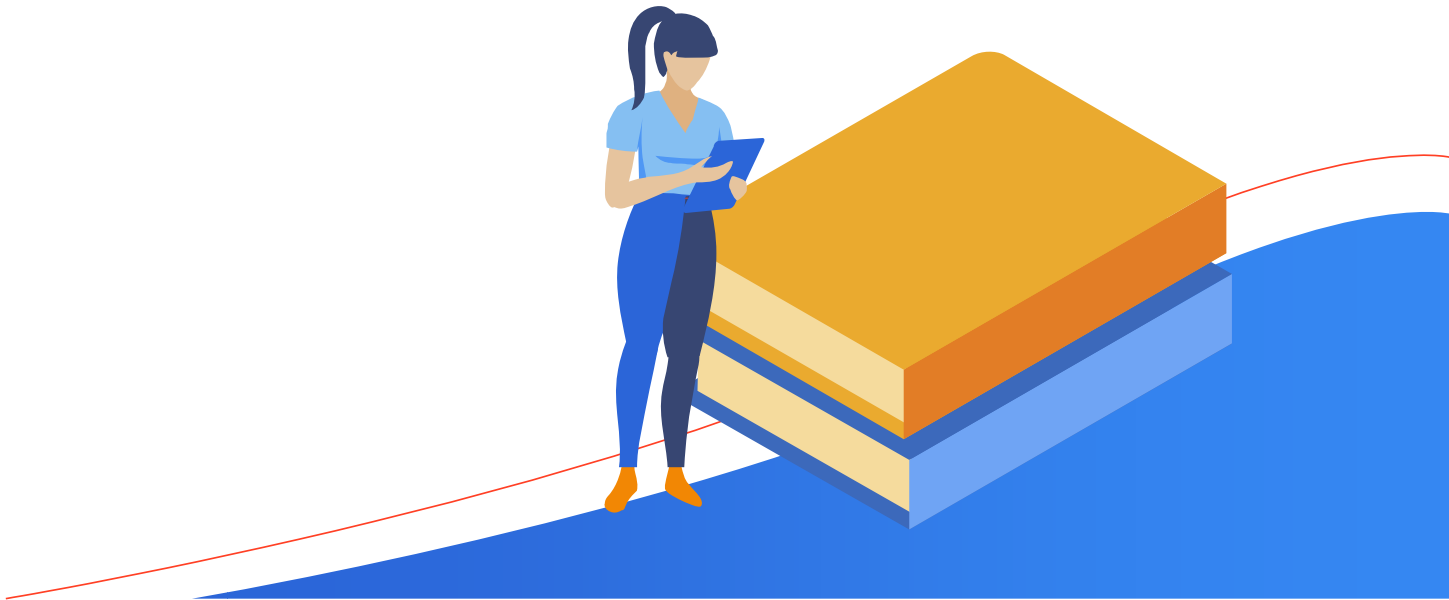
Turing AI 万维图灵 | 大数据系列课程

大数据技术-第十章：购物网站用户行为分析综合案例

Sqoop实现Hive、MySQL、HBase数据迁移

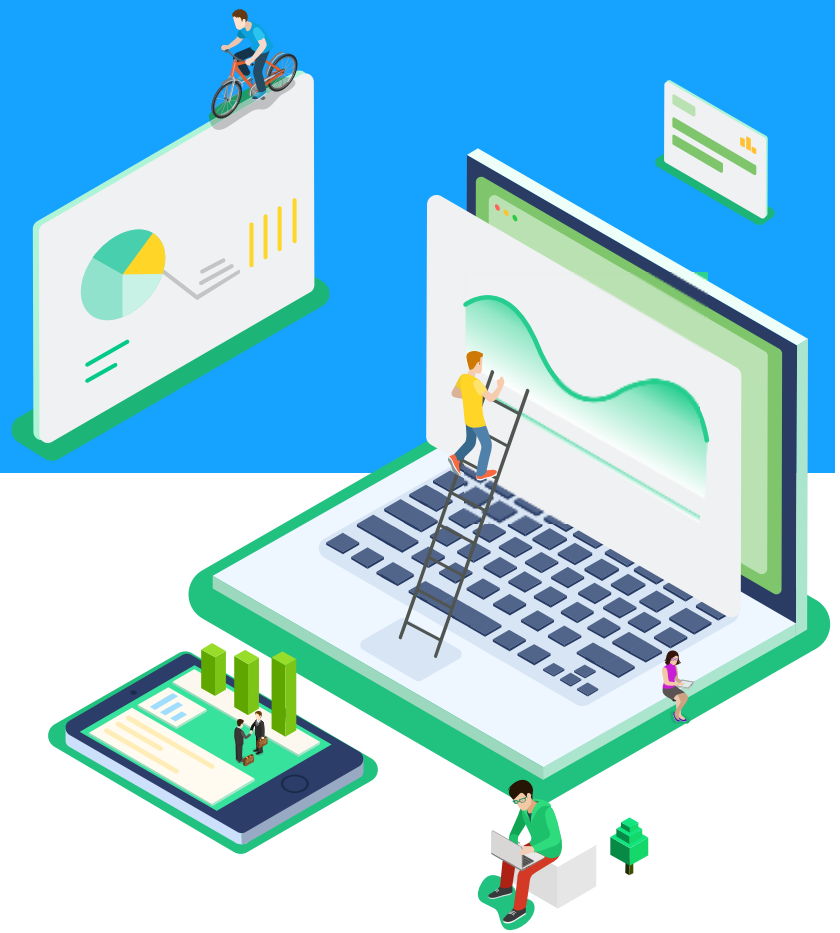


CONTENTS



01

Hive预操作



Hive预操作

Turing AI 万图 维灵
www.tringai.com

1. 创建临时表user_action

```
hive> create table dblab.user_action(id STRING,uid STRING, item_id STRING, behavior_type STRING, item_category STRING, visit_date DATE, province STRING) COMMENT 'Welcome to XMU dblab! ' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE;
```

可以新建一个终端，查看，这个数据文件在HDFS中确实已经被创建，请在新建的终端中执行下面命令：

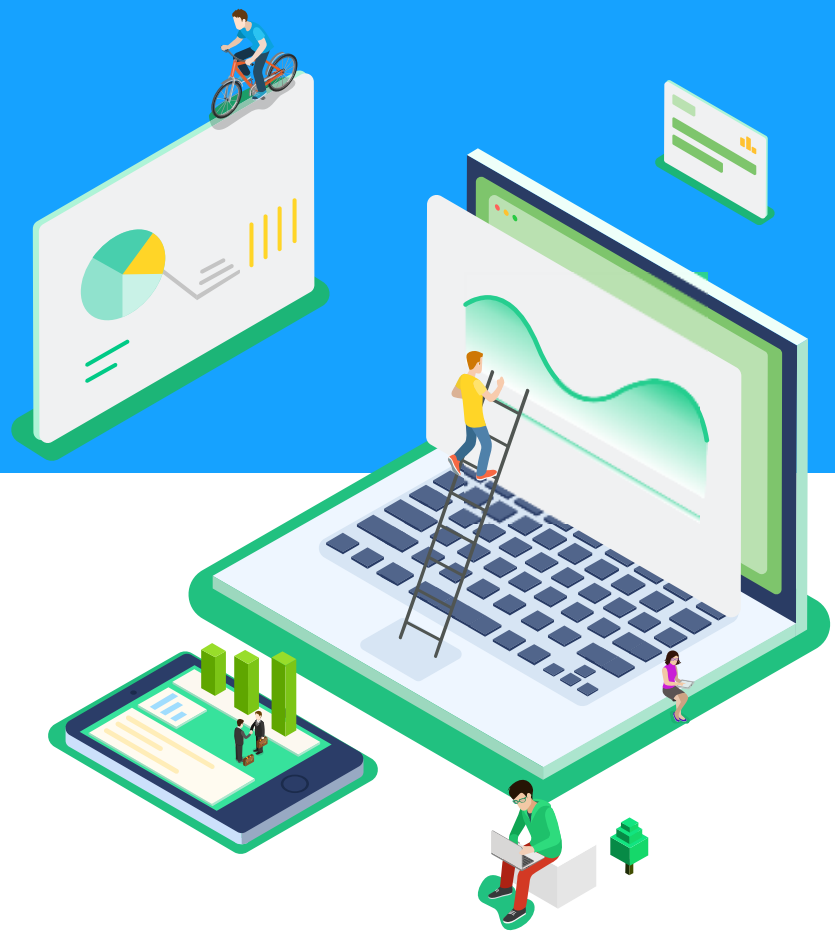
```
[hadoop@master ~]$ hdfs dfs -ls /user/hive/warehouse/dblab.db/
```

2.**将bigdata_user表中的数据插入到user_action**

```
## 把dblab.bigdata_user数据插入到dblab.user_action表中
hive> INSERT OVERWRITE TABLE dblab.user_action select * from dblab.bigdata_user;
## 查询上面的插入命令是否成功执行
hive> select * from user_action limit 10;
```

02

将Hive数据导入MySQL



将Hive数据导入MySQL

1.将前面生成的临时表数据从Hive导入到 MySQL中

(1)登录 MySQL

请在Linux系统中新建一个终端，执行下面命令：

```
$ mysql -u root -p
```

(2)创建数据库

```
mysql> show databases; #显示所有数据库
mysql> create database dblab; #创建dblab数据
mysql> use dblab; #使用数据库
```

将Hive数据导入MySQL

注意：请使用下面命令查看数据库的编码：

```
mysql>show variables like "char%";
```

```
+--------------------------+-----------------------------+
| Variable_name            | Value                       |
+--------------------------+-----------------------------+
| character_set_client     | utf8                        |
| character_set_connection | utf8                        |
| character_set_database   | latin1                      |
| character_set_filesystem | binary                      |
| character_set_results    | utf8                        |
| character_set_server     | latin1                      |
| character_set_system     | utf8                        |
| character_sets_dir       | /usr/share/mysql/charsets/  |
+--------------------------+-----------------------------+
8 rows in set (0.00 sec)
```

将Hive数据导入MySQL

修改了编码格式后，再次执行“show variables like ” char% “” 命令会得到如图所示的结果。

```
+--------------------------+-----------------------------+
| Variable_name            | Value                       |
+--------------------------+-----------------------------+
| character_set_client     | utf8                        |
| character_set_connection | utf8                        |
| character_set_database   | utf8                        |
| character_set_filesystem | binary                      |
| character_set_results    | utf8                        |
| character_set_server     | utf8                        |
| character_set_system     | utf8                        |
| character_sets_dir       | /usr/share/mysql/charsets/  |
+--------------------------+-----------------------------+
8 rows in set (0.00 sec)
```

将Hive数据导入MySQL

(3)创建表

下面在MySQL的数据库dblab中创建一个新表user_action，并设置其编码为utf-8：

```
mysql> CREATE TABLE `dblab`.`user_action` (`id`
varchar(50),`uid` varchar(50),`item_id`
varchar(50),`behavior_type` varchar(10),`item_category`
varchar(50), `visit_date` DATE,`province` varchar(20))
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

创建成功后，输入下面命令退出MySQL：

```
mysql> exit
```

将Hive数据导入MySQL

(4)通过sqoop:把hive数据，上传到MySQL。

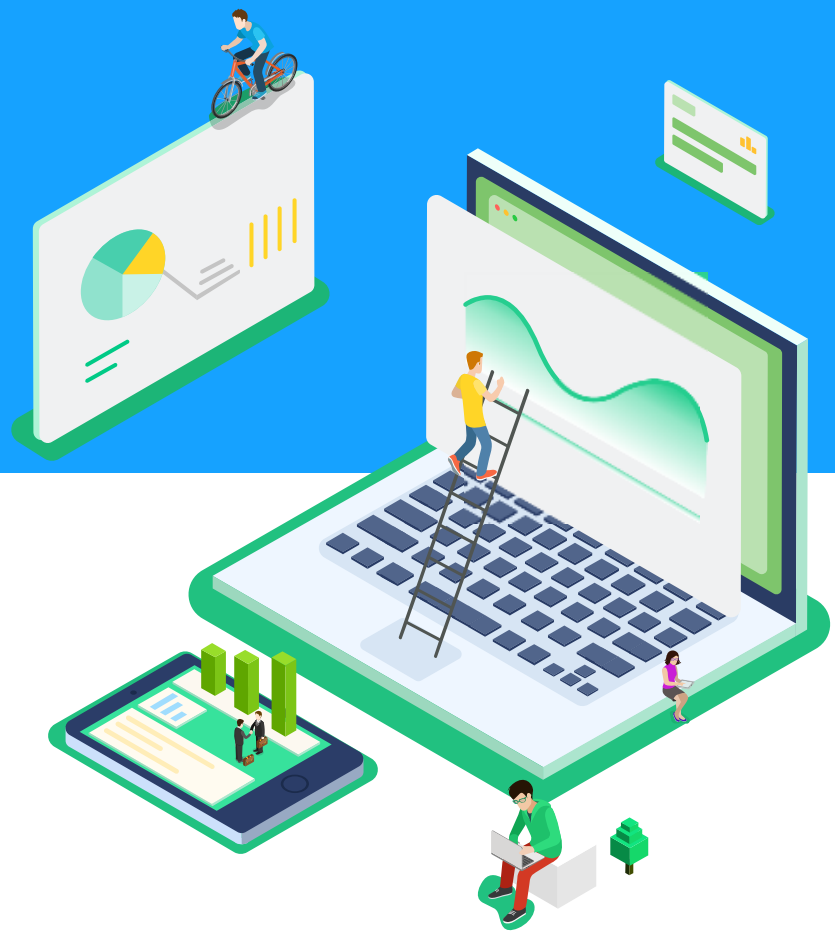
```
## 切换到Linux系统终端，执行下面命令
[hadoop@master ~]$ sqoop export -connect
jdbc:mysql://localhost:3306/dblab -username root -password
Password123$ -table user_action -export-dir
'/user/hive/warehouse/dblab.db/user_action' -fields-terminated-by '\t';
```

(5)通过查询MySQL数据。

```
## 切换到MySQL窗口，执行命令
mysql> select count(*) from user_action;
```

03

将MySQL数据导入到HBase



将MySQL数据导入到HBase

(1)启动HBase shell。

```
## 启动hbase shell
[hadoop@master ~]$  hbase shell
```

(2)创建表user_action。

在HBase中创建了一个user_action表，这个表中有一个列族f1，历史版本保留数量为5。

```
 ## 创建表user_action
hbase> create 'user_action', { NAME => 'f1',
VERSIONS => 5}
```

将MySQL数据导入到HBase

(3)将MySQL导入数据到HBase。

用sqoop工具将MySQL的表user_action的数据导入数据到HBase的user_action中。

```
 ## 启动hbase shell：
[hadoop@master ~]$ sqoop  import  --connect
jdbc:mysql://192.168.56.101:3306/dblab --username root --password Password123$
--table user_action --hbase-table user_action --column-family f1 --hbase-row-key
id --hbase-create-table -m 1
```

注意：IP部分需要与自己的MySQL库的一致。命令解释如下：

```
sqoop  import  --connect  jdbc:mysql://192.168.56.101:3306/dblab
--username  root
--password  Password123$
--table user_action
--hbase-table user_action #HBase中表名称
--column-family f1 #列簇名称
--hbase-row-key id #HBase 行键
--hbase-create-table #是否在不存在情况下创建表
-m 1 #启动 Map 数量
```

将MySQL数据导入到HBase

(4) 查看HBase中user_action表数据

现在，再次切换到HBase Shell运行的那个终端窗口，在“hbase>”命令提示符下，执行下面命令查询刚才导入的数据：

```
hbase(main):008:0> scan 'user_action',{LIMIT=>2}
ROW                   COLUMN+CELL
 1                    column=f1:behavior_type, timestamp=1614940534637, value=1
 1                    column=f1:item_category, timestamp=1614940534637, value=4076
 1                    column=f1:item_id, timestamp=1614940534637, value=285259775
 1                    column=f1:province, timestamp=1614940534637, value=\xE7\xA6\x8F\
                      xE5\xBB\xBA
 1                    column=f1:uid, timestamp=1614940534637, value=10001082
 1                    column=f1:visit_date, timestamp=1614940534637, value=2014-12-08
 10                   column=f1:behavior_type, timestamp=1614940534637, value=1
 10                   column=f1:item_category, timestamp=1614940534637, value=10894
 10                   column=f1:item_id, timestamp=1614940534637, value=323339743
 10                   column=f1:province, timestamp=1614940534637, value=\xE9\xBB\x91\
                      xE9\xBE\x99\xE6\xB1\x9F
 10                   column=f1:uid, timestamp=1614940534637, value=10001082
 10                   column=f1:visit_date, timestamp=1614940534637, value=2014-12-12
2 row(s) in 0.0330 seconds
```

注意，我们用limit2是返回HBase表中的前面2行数据，但是，上面的结果，从“行数”来看，给人一种错误，似乎不是2行，要远远多于2行。这是因为，HBase在显示数据的时候，和关系型数据库MySQL是不同的，每行显示的不是一行记录，而是一个“单元格”。

Turing AI 万维图灵 | 大数据系列课程

大数据

BIG DATA

智 / 能 / 科 / 技　　放 / 眼 / 未 / 来