**京东项目**

一、任务一：

1、将t\_user.csv导入MYSQL中

启动mysql和创建表

> mysql.server start

> mysql --local-infile -u root –p

mysql> create database jd\_db

mysql>--创建t\_user表（limit为mysql关键字，重新定义为limit1）

create table t\_user(

uid int(9) not null primary key,

age int(3),

sex int(1),

active\_date date,

limit1 double);

mysql>--加载数据

load data local infile '/Users/banny/Downloads/jd\_data/jdd\_dataset/t\_user.csv' into table t\_user fields terminated by ',' IGNORE 1 LINES ; --去掉第一行

2、将t\_loan\_sum导入到HDFS中

参考文献：https://blog.csdn.net/shushugood/article/details/51172140

2.1创建文本表

create table if not exists t\_loan\_sum\_txt(

uid INT,

month DATE,

loan\_sum DOUBLE)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

create table if not exists t\_click\_txt(

uid INT,

click\_time TIMESTAMP,

pid INT,

param INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

create table if not exists t\_loan\_txt(

uid INT,

loan\_time TIMESTAMP,

loan\_amount DOUBLE,

plannum INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

create table if not exists t\_order\_txt(

uid INT,

buy\_time DATE,

price DOUBLE,

qty INT,

cate\_id INT,

discount DOUBLE)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

≈

加载数据

load data local inpath '/Users/banny/Downloads/jd\_data/jdd\_dataset/t\_loan\_sum.csv' overwrite into table t\_loan\_sum\_txt;

load data local inpath '/Users/banny/Downloads/jd\_data/jdd\_dataset/t\_click.csv' overwrite into table t\_click\_txt;

load data local inpath '/Users/banny/Downloads/jd\_data/jdd\_dataset/t\_loan.csv' overwrite into table t\_loan\_txt;

load data local inpath '/Users/banny/Downloads/jd\_data/jdd\_dataset/t\_order.csv' overwrite into table t\_order\_txt;

2.2创建正式表

create table if not exists t\_loan\_sum(

uid INT,

month DATE,

loan\_sum DOUBLE)

ROW FORMAT DELIMITED

STORED AS ORC ;

导入数据

insert into t\_loan\_sum select \* from t\_loan\_sum\_txt;

create table if not exists t\_click(

uid INT,

click\_time TIMESTAMP,

pid INT,

param INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS ORC;

insert into t\_click select \* from t\_click\_txt;

create table if not exists t\_loan(

uid INT,

loan\_time TIMESTAMP,

loan\_amount DOUBLE,

plannum INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS ORC;

insert into t\_loan select \* from t\_loan\_txt;

create table if not exists t\_order(

uid INT,

buy\_time DATE,

price DOUBLE,

qty INT,

cate\_id INT,

discount DOUBLE)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS ORC;

insert into t\_order select \* from t\_order\_txt;

二、任务二

将mysql中t\_user 导入HDFS中，使用sqoop

参考文献：<https://blog.csdn.net/lusyoe/article/details/60478226>

1、使用Sqoop将mySQL中的数据导入HDFS

sqoop:000> set server --host 127.0.0.1 --port 12000 --webapp sqoop

Server is set successfully

sqoop:000> show version -all

client version:

Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb

Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016

server version:

Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb

Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016

API versions:

[v1]

sqoop:000> show connector

+------------------------+---------+------------------------------------------------------------+----------------------+

| Name | Version | Class | Supported Directions |

+------------------------+---------+------------------------------------------------------------+----------------------+

| generic-jdbc-connector | 1.99.7 | org.apache.sqoop.connector.jdbc.GenericJdbcConnector | FROM/TO |

| kite-connector | 1.99.7 | org.apache.sqoop.connector.kite.KiteConnector | FROM/TO |

| oracle-jdbc-connector | 1.99.7 | org.apache.sqoop.connector.jdbc.oracle.OracleJdbcConnector | FROM/TO |

| ftp-connector | 1.99.7 | org.apache.sqoop.connector.ftp.FtpConnector | TO |

| hdfs-connector | 1.99.7 | org.apache.sqoop.connector.hdfs.HdfsConnector | FROM/TO |

| kafka-connector | 1.99.7 | org.apache.sqoop.connector.kafka.KafkaConnector | TO |

| sftp-connector | 1.99.7 | org.apache.sqoop.connector.sftp.SftpConnector | TO |

+------------------------+---------+------------------------------------------------------------+----------------------+

sqoop:000> create link -connector generic-jdbc-connector

Creating link for connector with name generic-jdbc-connector

Please fill following values to create new link object

Name: MySQL

Database connection

Driver class: com.mysql.jdbc.Driver

Connection String: jdbc:mysql://localhost:3306/jd\_db

Username: root

Password: \*\*\*\*

Fetch Size:

Connection Properties:

There are currently 0 values in the map:

entry# protocol=tcp

There are currently 1 values in the map:

protocol = tcp

entry#

SQL Dialect

Identifier enclose:

New link was successfully created with validation status WARNING and name MySQL

sqoop:000> show link

+-------+------------------------+---------+

| Name | Connector Name | Enabled |

+-------+------------------------+---------+

| MySQL | generic-jdbc-connector | true |

+-------+------------------------+---------+

sqoop:000> create link -connector hdfs-connector

Creating link for connector with name hdfs-connector

Please fill following values to create new link object

Name: HDFS

HDFS cluster

URI: hdfs://localhost:9000

Conf directory: /Users/banny/bigdata/hadoop-2.7.3/etc/hadoop

Additional configs::

There are currently 0 values in the map:

entry#

New link was successfully created with validation status OK and name HDFS

sqoop:000> create job -f "MySQL" -t "HDFS"

Creating job for links with from name MySQL and to name HDFS

Please fill following values to create new job object

Name: mysqlTohdfs

Database source

Schema name: jd\_db

Table name: t\_user

SQL statement:

Column names:

There are currently 0 values in the list:

element#

Partition column:

Partition column nullable:

Boundary query:

Incremental read

Check column:

Last value:

Target configuration

Override null value:

Null value:

File format:

0 : TEXT\_FILE

1 : SEQUENCE\_FILE

2 : PARQUET\_FILE

Choose: 0

Compression codec:

0 : NONE

1 : DEFAULT

2 : DEFLATE

3 : GZIP

4 : BZIP2

5 : LZO

6 : LZ4

7 : SNAPPY

8 : CUSTOM

Choose: 0

Custom codec:

Output directory: hdfs://localhost:9000/data/sqoop/jd\_db/

Append mode:

Throttling resources

Extractors:

Loaders:

Classpath configuration

Extra mapper jars:

There are currently 0 values in the list:

element#

New job was successfully created with validation status OK and name mysqlTohdfs

sqoop:000> start job –n mysqlTohdfs

sqoop:000> status job -n mysqlTohdfs

Submission details

Job Name: mysqlTohdfs

Server URL: http://localhost:12000/sqoop/

Created by: banny

Creation date: 2018-03-23 15:22:59 CST

Lastly updated by: banny

External ID: job\_1521789169893\_0001

http://localhost:18088/proxy/application\_1521789169893\_0001/

2018-03-23 15:23:37 CST: RUNNING - 30.00 %

2、将HDFS上的数据加载到表t\_user中

create table if not exists t\_user\_txt(

uid INT,

age INT,

sex INT,

active DATE,

limit1 DOUBLE)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE ;

load data inpath '/data/sqoop/jd\_db/ ' overwrite into table t\_user\_txt;

create table if not exists t\_user (

uid INT,

age INT,

sex INT,

active DATE,

limit1 DOUBLE)

STORED AS ORC ;

insert into t\_user select \* from t\_user\_txt;

三、任务三

各年龄段消费者每日购买商品总价值

select age,sum(price\*qty-discount)as mn,t\_order.buy\_time from t\_user join t\_order on t\_user.uid=t\_order.uid group by age,buy\_time order by buy\_time limit 10;

--每月

select age,sum(price\*qty-discount)as mn, date\_format(buy\_time,'%Y-%m') from t\_user join t\_order on t\_user.uid=t\_order.uid group by age, date\_format(buy\_time,'%Y-%m') order by date\_format(buy\_time,'%Y-%m'),age;

男女消费者每日借贷金额

select case when sex=1 then '女' else '男' end, sum(loan\_amount) as mn,date\_format(loan\_time, '%y-%m-%d') from t\_user join t\_loan on t\_user.uid=t\_loan.uid group by sex, date\_format(loan\_time, '%y-%m-%d');

--每月

select case when sex=1 then '女' else '男' end, sum(loan\_amount) as mn,date\_format(loan\_time, '%Y-%m') from t\_user join t\_loan on t\_user.uid=t\_loan.uid group by sex, date\_format(loan\_time,'%Y-%m') order by

date\_format(loan\_time,'%Y-%m'),sex ;

四、任务四

利用Spark RDD或Spark DataFrame分析产生以下结果:

* 借款金额超过 200 且购买商品总价值超过借款总金额的用户 ID

select a.u\_id from (select t\_loan\_sum .uid as u\_id,sum(price\*qty-discount) as buy, sum(loan\_sum) as loan from t\_loan\_sum join t\_order on t\_loan\_sum.uid= t\_order.uid group by t\_loan\_sum .uid) a where a.loan > 200 and a.buy >a.loan;

* 从不买打折产品且不借款的用户 ID

select uid from t\_order group by uid having sum(discount) =0 and uid not in(select uid from t\_loan\_sum) limit 10;