1、进入CDH环境

2、导入过程：

a、选择需要导入hive的sql表,将其表结构导出,修改为hive支持的数据类型,将表结构sql文件放入集群本地文件。(需要注意表结构和数据要一一对应)

b、使用命令hive -f "xx.sql"运行sql文件,创建表。

c、使用命令hive,进入hive。

d、选择需要导入hive的sql表,将其数据导出为TXT结构,再将文件放入集群本地文件。

(注意导出的数据可能带有“”,需要对应正确的字段类型或者批量修改txt文件内容)

e、在hive中使用命令load data local inpath "xx.txt" into table "xx";

3、查看数据,可行则进行下一步。

4、数据清洗,有些脏数据会影响排名结果。

5、选择排行的某一项,进行排名

select createShopUserName,count(\*) as np from ods\_shopbill group by createShopUserName order by np desc limit 10;

create table query\_result as select shopGoodsName as goodsName,sum(amount) as number from shopbillitem group by shopGoodsName order by number DESC limit 20;

得出排名结果。

6、导出数据：

INSERT OVERWRITE DIRECTORY '/zwy' ROW FORMAT DELIMITED FIELDS TERMINATED by ',' select \* from ljd.month8rank;

hadoop fs -get '/zwy/000000\_0' '/home/ljdoutput'

7、使用sqoop将数据导入mysql：

a、hive中建立临时表shopbillitem\_temp

b、选中数据并导入临时表

c、sqoop连接导入mysql

sudo -uhdfs sqoop export --connect jdbc:mysql://112.74.172.76:3306/tobacoo?characterEncoding=UTF-8 --username --password --table export --export-dir '/user/hiv

e/warehouse/tobacoo\_dw.db/sqoop' --fields-terminated-by '\001';

MYSQL相关：

数据查询语句：

今日：select \*FROM ods\_shopbillitem WHERE to\_days(checkOutTime) = to\_days(now());

昨天：select \*FROM ods\_shopbillitem WHERE TO\_DAYS( NOW( ) ) - TO\_DAYS( checkOutTime) <= 1;

7天内: select \*FROM ods\_shopbillitem WHERE DATE\_SUB(CURDATE(), INTERVAL 7 DAY) <= date(checkOutTime);

上周：SELECT name,submittime FROM enterprise WHERE YEARWEEK(date\_format(submittime,'%Y-%m-%d')) = YEARWEEK(now())-1;

本月：select \* from user where DATE\_FORMAT(pudate,'%Y%m') = DATE\_FORMAT(CURDATE(),'%Y%m') ;

时间段查询：select \* from tobacoo\_dw.ods\_shopbillitem where checkouttime >='2019-10-08 00:00:00' and checkOutTime < '2019-10-15 00:00:00';

上月：SELECT \* FROM 表名 WHERE PERIOD\_DIFF( date\_format( now( ) , '%Y%m' ) , date\_format( 时间字段名, '%Y%m' ) ) =1

删除某一年数据：DELETE FROM week\_shop WHERE date\_format(checkOutTime,'%Y')='2034'

直接rank排名:

set @x=0;

CREATE TABLE day\_result\_rank AS(SELECT a.goods\_name,a.number,@x:=@x+1 as ranking from(select shopGoodsName as goods\_name,sum(amount) as number from day\_shop group by shopGoodsName order by number DESC limit 20) a);

今日增长数：

CREATE TABLE day21\_growth(

SELECT day21\_result\_rank.goods\_name,(day21\_result\_rank.ranking-day20\_result\_rank.ranking) AS growth FROM day21\_result\_rank,day20\_result\_rank WHERE day21\_result\_rank.goods\_name = day20\_result\_rank.goods\_name;)

销量内增长排名：

SET @x=0;

CREATE table day21\_growth\_rank as(SELECT a.goods\_name,a.growth,@x:=@x+1 as ranking FROM(SELECT goods\_name as goods\_name,growth AS growth FROM day21\_growth ORDER BY growth DESC limit 20)a);

汇总统计：

CREATE TABLE day21\_growth\_rank SELECT day21\_result\_rank.goods\_name,day21\_result\_rank.number,day21\_result\_rank.ranking,(-day21\_result\_rank.ranking+day20\_result\_rank.ranking) AS growth,(-day21\_result\_rank.number+day20\_result\_rank.number) AS growth\_number FROM day21\_result\_rank,day20\_result\_rank WHERE day21\_result\_rank.goods\_name = day20\_result\_rank.goods\_name;

求增长率：

CREATE TABLE temp SELECT \*,thismonth\_growth\_rank.number/(thismonth\_growth\_rank.number-thismonth\_growth\_rank.growth\_number) AS percent FROM thismonth\_growth\_rank;

update temp set percent=percent-1 ;

drop table thismonth\_growth\_rank;

RENAME table temp to thismonth\_growth\_rank;

hive查询6月份数据：select \* from tobacoo\_dw.shopbillitem where checkouttime like concat('%6/2019%');

修改字段名：ALTER TABLE tobacoo\_dw.shopbillitem CHANGE COLUMN checkouttime checkouttime DATE;

查看库：show databases;

查看表：show tables;

查看表前100条数据：select \*from "xx" limit 100;

删除表：drop table "xx";

删除表数据：truncate table xx;s

复制表结构：CREATE TABLE A.new\_table LIKE B.old\_table;

查看表内有多少条数据：select count(\*) from ods\_shopbill;

创建临时表：create table query\_result as select shopGoodsName as goodsName,sum(amount) as number from shopbillitem group by shopGoodsName order by number DESC limit 20;

建表：create table shopbillitem(id bigint,shopId bigint,billItemCode string,billCode string,shopGoodsId bigint,shopGoodsName string,

shopGoodsSecondCategoryId bigint,amountOrder double,amount double,price double,memberPrice double,totalGoodsPrice double,

totalGoodsConsume double,totalDiscount double,paymentModule int,costPrice double,profit double,createShopUserId bigint,

createShopUserName string,createTime string,isDiscount tinyint,discountRate double,enableAreaDiscount tinyint,enableMemberDiscount tinyint,

goodsType string,checkOutTime string,timeline string)

row format delimited

fields terminated by '\t'

stored as textfile;

create table ljd.shopbillitem(shopid bigint,shopgoodsid bigint,shopgoodsname string,amount double,price double,totalgoodsconsume double,costprice double,profit double,discoutrate double

,barcode string,checkouttime date)row format delimited fields terminated by '\t' stored as textfile;

hive支持的基本数据类型：

TINYINT

SMALLINT

INT

BIGINT

BOOLEAN

FLOAT

DOUBLE

DOUBLE PRECISION

STRING

BINARY

TIMESTAMP

DECIMAL

DECIMAL(precision, scale)

DATE

VARCHAR

CHAR

参考ods\_shopbill表的信息：

MapReduce Jobs Launched:

Stage-Stage-1: Map: 14 Reduce: 1 Cumulative CPU: 49.44 sec HDFS Read: 3620754598 HDFS Write: 8 SUCCESS

Total MapReduce CPU Time Spent: 49 seconds 440 msec

OK

9028264