



链滴

Hive 分析数据过程

作者: [sq8852161](#)

原文链接: <https://ld246.com/article/1566974665651>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



当ClouderaManager平台搭建好之后，如何分析数据。

以下是一个使用Hive来处理的过程。以供参考

业务太阳模型：

维度：

事实：

登录根用户

评估分析模型：

在hive终端：

```
CREATE DATABASE IF NOT EXISTS transformdb;
```

建立维度表

```
CREATE TABLE IF NOT EXISTS transformdb.dimperson(  
personkey BIGINT,  
firstname STRING,  
lastname STRING  
)  
CLUSTERED BY (firstname,lastname,personkey) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional'='true','orc.compress'='ZLIB','orc.create.index'='true');
```

建立transformdb fct事实表

```
CREATE TABLE IF NOT EXISTS transformdb.fctpersonaccount(  
personaccountkey BIGINT,
```

```
personkey BIGINT,  
accountkey BIGINT,  
balance DECIMAL(18,9)  
)  
CLUSTERED BY (personkey,accountkey) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES ('transactional'='true','orc.compress'='ZLIB','orc.create.index'='true');
```

建立新数据库:

```
CREATE DATABASE IF NOT EXISTS reportdb;  
CREATE DATABASE IF NOT EXISTS organisedb;
```

创建表 account 维度表

```
CREATE TABLE IF NOT EXISTS transformdb.dimaccount(  
accountkey BIGINT,  
accountnumber INT  
)  
CLUSTERED BY (accountnumber,accountkey) INTO 1 BUCKETS  
  
STORED AS orc  
TBLPROPERTIES('transactional'='true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
CREATE TABLE IF NOT EXISTS organisedb.dimaccount LIKE transformdb.dimaccount;
```

创建 organisedb fct 事实表

```
CREATE TABLE IF NOT EXISTS organisedb.fctpersonaccount(  
personaccountkey BIGINT,  
personkey BIGINT,  
accountkey BIGINT,  
balance DECIMAL(18,9)  
)  
CLUSTERED BY (personkey,accountkey) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES ('transactional'='true','orc.compress'='ZLIB','orc.create.index'='true');
```

创建表 organisedb.dimperson

```
CREATE TABLE IF NOT EXISTS organisedb.dimperson(  
personkey BIGINT,  
firstname STRING,  
lastname STRING  
)  
CLUSTERED BY (firstname,lastname,personkey) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES ('transactional'='true','orc.compress'='ZLIB','orc.create.index'='true');
```

创建表reportdb.report001

```
CREATE TABLE IF NOT EXISTS reportdb.report001(  
  firstname STRING,  
  lastname STRING,  
  accountnumber INT,  
  balance DECIMAL(18,9)  
)  
CLUSTERED BY (firstname,lastname) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES ('transactional'='true','orc.compress'='ZLIB','orc.create.index'='true');
```

建立评估模型

```
INSERT INTO TABLE reportdb.report001  
SELECT  
  dimperson.firstname,dimperson.lastname,  
  dimaccount.accountnumber,  
  fctpersonaccount.balance  
FROM  
  organisedb.fctpersonaccount  
JOIN  
  organisedb.dimperson  
ON  
  fctpersonaccount.personkey = dimperson.personkey  
JOIN  
  organisedb.dimaccount  
ON  
  fctpersonaccount.accountkey = dimaccount.accountkey
```

检索数据库:

创建数据库:

```
CREATE DATABASE IF NOT EXISTS retrievedb;
```

将示例数据拷贝到master机器硬盘

```
scp -r 下载/00rawdata/ root@10.23.0.136:/tmp
```

建立hdfs文件夹

```
hadoop fs -mkdir /user/retrieve/
```

从master硬盘拷贝文件到hdfs文件系统

```
hadoop fs -put /tmp/00rawdata/ /user/retrieve
```

将文件置为可读可写可执行

```
hadoop fs -chmod 777 /user/retrieve/00rawdata/*.*
```

创建各数据文件表,并加载数据

```
rawfirstname
CREATE TABLE IF NOT EXISTS retrievedb.rawfirstname (
  firstnameid  string,
  firstname    string,
  sex          string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;'
```

加载数据

第一种,从HDFS文件系统加载

```
LOAD DATA INPATH '/user/retrieve/00rawdata/rawfirstname.csv' OVERWRITE INTO TABLE re
rievedb.rawfirstname;
```

第二种,从本地文件系统加载

```
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawfirstname.csv' OVERWRITE INTO TABL
retrievedb.rawfirstname;
```

```
rawlastname
CREATE TABLE IF NOT EXISTS retrievedb.rawlastname (
  lastnameid  string,
  lastname    string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;'
```

```
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawlastname.csv' OVERWRITE INTO TABLE
retrievedb.rawlastname;
```

```
rawperson
CREATE TABLE IF NOT EXISTS retrievedb.rawperson (
  persid      string,
  firstnameid string,
  lastnameid  string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;'
```

```
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawperson.csv' OVERWRITE INTO TABLE r
trievedb.rawperson;
```

```
rawdatetime
CREATE TABLE IF NOT EXISTS retrievedb.rawdatetime (
  id          string,
  datetimes   string,
  monthname   string,
  yearnumber  string,
  monthnumber string,
  daynumber   string,
  hournumber  string,
  minutenumbe string,
  ampm        string
```

```
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;';
```

```
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawdatetime.csv' OVERWRITE INTO TABLE  
retrievedb.rawdatetime;
```

```
rawaddress  
CREATE TABLE IF NOT EXISTS retrievedb.rawaddress (  
  id      string,  
  Postcode string,  
  Latitude string,  
  Longitude string,  
  Easting string,  
  Northing string,  
  GridRef string,  
  District string,  
  Ward string,  
  DistrictCode string,  
  WardCode string,  
  Country string,  
  CountyCode string,  
  Constituency string,  
  TypeArea string  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;';
```

```
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawaddress.csv' OVERWRITE INTO TABLE  
etrievedb.rawaddress;
```

```
rawaddresshistory  
CREATE TABLE IF NOT EXISTS retrievedb.rawaddresshistory(  
  id      string,  
  pid     string,  
  aid     string,  
  did1    string,  
  did2    string  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;';
```

```
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawaddresshistory.csv' OVERWRITE INTO  
ABLE retrievedb.rawaddresshistory;
```

```
rawaccount  
CREATE TABLE IF NOT EXISTS retrievedb.rawaccount (  
  id      string,  
  pid     string,  
  accountno string,  
  balance string  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ';;';  
LOAD DATA LOCAL INPATH 'file:///tmp/00rawdata/rawaccount.csv' OVERWRITE INTO TABLE
```

```
etrievedb.rawaccount;
```

评估数据库:

创建评估数据库

```
CREATE DATABASE IF NOT EXISTS assessdb;
```

创建 firstname001 临时表 存储去除标题的数据

```
CREATE TABLE IF NOT EXISTS assessdb.firstname001 (  
  firstnameid  string,  
  firstname    string,  
  sex          string  
)  
CLUSTERED BY (firstnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

清除表中数据

```
TRUNCATE TABLE assessdb.firstname001;
```

清除表中标题

```
INSERT INTO TABLE assessdb.firstname001  
SELECT firstnameid, firstname, sex  
FROM retrievdb.rawfirstname  
WHERE firstnameid <> "id";
```

建立临时表firstname002 存储删除空格的数据

```
CREATE TABLE IF NOT EXISTS assessdb.firstname002 (  
  firstnameid  string,  
  firstname    string,  
  sex          string  
)  
CLUSTERED BY (firstnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
TRUNCATE TABLE assessdb.firstname002;
```

使用函数去除字段空格

```
INSERT INTO TABLE assessdb.firstname002  
SELECT firstnameid, rtrim(ltrim(firstname)), rtrim(ltrim(sex))  
FROM assessdb.firstname001;
```

建立 firstname003 转换firstname中的数据类型

```
CREATE TABLE IF NOT EXISTS assessdb.firstname003 (  
  firstnameid  int,  
  firstname    string,  
  sex          string  
)  
CLUSTERED BY (firstnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');  
  
TRUNCATE TABLE assessdb.firstname003;  
  
INSERT INTO TABLE assessdb.firstname003  
SELECT CAST(firstnameid as INT), SUBSTRING(firstname,2,LENGTH(firstname)-2), SUBSTRING(  
ex,2,LENGTH(sex)-2)  
FROM assessdb.firstname002;
```

创建fistname表,将处理后的数据存储在表里

```
CREATE TABLE IF NOT EXISTS assessdb.firstname (  
  firstnameid  int,  
  firstname    string,  
  sex          string  
)  
CLUSTERED BY (firstnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');  
  
TRUNCATE TABLE assessdb.firstname;
```

将处理后的数据迁移到firstname表中

```
INSERT INTO TABLE assessdb.firstname  
SELECT firstnameid, firstname, sex  
FROM assessdb.firstname003  
ORDER BY firstnameid;
```

查询数据

```
SELECT firstnameid, firstname, sex from assessdb.firstname SORT BY firstname LIMIT 10;
```

相同过程处理lastname

```
CREATE TABLE IF NOT EXISTS assessdb.lastname001 (  
  lastnameid  string,  
  lastname    string  
)  
CLUSTERED BY (lastnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```



```
TRUNCATE TABLE assessdb.lastname001;
```

```
INSERT INTO TABLE assessdb.lastname001  
SELECT lastnameid, lastname  
FROM retrievedb.rawlastname  
WHERE lastnameid <> "id";
```

```
CREATE TABLE IF NOT EXISTS assessdb.lastname002 (  
  lastnameid  string,  
  lastname    string  
)  
CLUSTERED BY (lastnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.lastname002  
SELECT lastnameid, rtrim(ltrim(lastname))  
FROM assessdb.lastname001;
```

```
CREATE TABLE IF NOT EXISTS assessdb.lastname003 (  
  lastnameid  int,  
  lastname    string  
)  
CLUSTERED BY (lastnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.lastname003  
SELECT CAST(lastnameid as INT), SUBSTRING(lastname,2,LENGTH(lastname)-2)  
FROM assessdb.lastname002;
```

```
CREATE TABLE IF NOT EXISTS assessdb.lastname (  
  lastnameid  int,  
  lastname    string  
)  
CLUSTERED BY (lastnameid) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.lastname  
SELECT lastnameid, lastname  
FROM assessdb.lastname003  
ORDER BY lastnameid;
```

相同过程处理person

```
CREATE TABLE IF NOT EXISTS assessdb.person001 (  
  persid      string,  
  firstnameid string,  
  lastnameid  string  
)  
CLUSTERED BY (persid) INTO 1 BUCKETS
```

```
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.person001
SELECT persid, firstnameid, lastnameid
FROM retrievedb.rawperson
WHERE persid <> "id";
```

```
CREATE TABLE IF NOT EXISTS assessdb.person002 (
  persid      int,
  firstnameid int,
  lastnameid  int
)
CLUSTERED BY (persid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.person002
SELECT CAST(persid as INT), CAST(firstnameid as INT), CAST(lastnameid as INT)
FROM assessdb.person001;
```

```
CREATE TABLE IF NOT EXISTS assessdb.person (
  persid      int,
  firstnameid int,
  lastnameid  int
)
CLUSTERED BY (persid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.person
SELECT persid, firstnameid, lastnameid
FROM assessdb.person002;
```

处理,创建组合表personfull,将各个表数据组合起来

```
CREATE TABLE IF NOT EXISTS assessdb.personfull(
  persid      int,
  firstnameid int,
  firstname   string,
  lastnameid  int,
  lastname    string,
  sex         string
)
CLUSTERED BY (persid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.personfull
SELECT person.persid, person.firstnameid, firstname.firstname, person.lastnameid, lastname.la
tname, firstname.sex
FROM assessdb.firstname
JOIN assessdb.person
ON firstname.firstnameid = person.firstnameid
```

```
JOIN assessdb.lastname
ON lastname.lastnameid = person.lastnameid;
```

清理临时表,空出空间

```
DROP TABLE assessdb.firstname001;
DROP TABLE assessdb.firstname002;
DROP TABLE assessdb.firstname003;
DROP TABLE assessdb.lastname001;
DROP TABLE assessdb.lastname002;
DROP TABLE assessdb.lastname003;
DROP TABLE assessdb.person001;
DROP TABLE assessdb.person002;
```

评估datetime表

```
CREATE TABLE IF NOT EXISTS assessdb.datetime001 (
  id      string,
  datetimes string,
  monthname string,
  yearnumber string,
  monthnumber string,
  daynumber string,
  hournumber string,
  minutenum string,
  ampm     string
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.datetime001
SELECT
  id,
  datetimes,
  monthname,
  yearnumber,
  monthnumber,
  daynumber,
  hournumber,
  minutenum,
  ampm
FROM retrievedb.rawdatetime
WHERE id <> "id";
```

```
CREATE TABLE IF NOT EXISTS assessdb.datetime002 (
  id      string,
  datetimes string,
  monthname string,
  yearnumber string,
  monthnumber string,
  daynumber string,
  hournumber string,
```

```

    minutenumber string,
    ampm          string
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```

```

INSERT INTO TABLE assessdb.datetime002
SELECT
    id,
    rtrim(ltrim(datetimes)),
    rtrim(ltrim(monthname)),
    rtrim(ltrim(yearnumber)),
    rtrim(ltrim(monthnumber)),
    rtrim(ltrim(daynumber)),
    rtrim(ltrim(hournumber)),
    rtrim(ltrim(minutenumber)),
    rtrim(ltrim(ampm))
FROM assessdb.datetime001;

```

```

CREATE TABLE IF NOT EXISTS assessdb.datetime003 (
    id          int,
    datetimes  string,
    monthname  string,
    yearnumber int,
    monthnumber int,
    daynumber  int,
    hournumber int,
    minutenumber int,
    ampm       string
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```

```

INSERT INTO TABLE assessdb.datetime003
SELECT
    CAST(id as INT),
    SUBSTRING(datetimes,2,LENGTH(datetimes)-2),
    SUBSTRING(monthname,2,LENGTH(monthname)-2),
    CAST(yearnumber as INT),
    CAST(monthnumber as INT),
    CAST(daynumber as INT),
    CAST(hournumber as INT),
    CAST(minutenumber as INT),
    SUBSTRING(ampm,2,LENGTH(ampm)-2)
FROM assessdb.datetime002;

```

```

CREATE TABLE IF NOT EXISTS assessdb.dates (
    id          int,
    datetimes  string,
    monthname  string,
    yearnumber int,
    monthnumber int,

```

```

    daynumber    int,
    hournumber   int,
    minutenumbe int,
    ampm         string
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

INSERT INTO TABLE assessdb.dates
SELECT
    id,
    datetimes,
    monthname,
    yearnumber,
    monthnumber,
    daynumber,
    hournumber,
    minutenumbe,
    ampm
FROM assessdb.datetime003;
DROP TABLE assessdb.datetime001;
DROP TABLE assessdb.datetime002;
DROP TABLE assessdb.datetime003;

```

评估,过滤 address表数据

```

CREATE TABLE IF NOT EXISTS assessdb.address001 (
    id          STRING,
    postcode    STRING,
    latitude    STRING,
    longitude   STRING,
    easting     STRING,
    northing    STRING,
    gridref     STRING,
    district    STRING,
    ward        STRING,
    districtcode STRING,
    wardcode    STRING,
    country     STRING,
    countycode  STRING,
    constituency STRING,
    typearea    STRING
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

INSERT INTO TABLE assessdb.address001
SELECT
    id,
    postcode,
    latitude,
    longitude,

```

```
easting,  
northing,  
gridref,  
district,  
ward,  
districtcode,  
wardcode,  
country,  
countycode,  
constituency,  
typearea  
FROM retrievedb.rawaddress  
WHERE id <> "id";
```

```
CREATE TABLE IF NOT EXISTS assessdb.address002 (  
  id      STRING,  
  postcode  STRING,  
  latitude  STRING,  
  longitude  STRING,  
  easting   STRING,  
  northing  STRING,  
  gridref   STRING,  
  district  STRING,  
  ward      STRING,  
  districtcode STRING,  
  wardcode  STRING,  
  country   STRING,  
  countycode STRING,  
  constituency STRING,  
  typearea  STRING  
)  
CLUSTERED BY (id) INTO 1 BUCKETS  
STORED AS orc  
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.address002  
SELECT  
  id,  
  rtrim(ltrim(postcode)),  
  rtrim(ltrim(latitude)),  
  rtrim(ltrim(longitude)),  
  rtrim(ltrim(easting)),  
  rtrim(ltrim(northing)),  
  rtrim(ltrim(gridref)),  
  rtrim(ltrim(district)),  
  rtrim(ltrim(ward)),  
  rtrim(ltrim(districtcode)),  
  rtrim(ltrim(wardcode)),  
  rtrim(ltrim(country)),  
  rtrim(ltrim(countycode)),  
  rtrim(ltrim(constituency)),  
  rtrim(ltrim(typearea))  
FROM assessdb.address001;
```

```

CREATE TABLE IF NOT EXISTS assessdb.address003 (
  id      INT,
  postcode  STRING,
  latitude  DECIMAL(18, 9),
  longitude  DECIMAL(18, 9),
  easting   INT,
  northing  INT,
  gridref   STRING,
  district  STRING,
  ward      STRING,
  districtcode STRING,
  wardcode  STRING,
  country   STRING,
  countycode STRING,
  constituency STRING,
  typearea  STRING
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```

```

INSERT INTO TABLE assessdb.address003
SELECT
  CAST(id as INT),
  SUBSTRING(postcode,2,LENGTH(postcode)-2),
  CAST(latitude as DECIMAL(18, 9)),
  CAST(longitude as DECIMAL(18, 9)),
  CAST(easting as INT),
  CAST(northing as INT),
  SUBSTRING(gridref,2,LENGTH(gridref)-2),
  SUBSTRING(district,2,LENGTH(district)-2),
  SUBSTRING(ward,2,LENGTH(ward)-2),
  SUBSTRING(districtcode,2,LENGTH(districtcode)-2),
  SUBSTRING(wardcode,2,LENGTH(wardcode)-2),
  SUBSTRING(country,2,LENGTH(country)-2),
  SUBSTRING(countycode,2,LENGTH(countycode)-2),
  SUBSTRING(constituency,2,LENGTH(constituency)-2),
  SUBSTRING(typearea,2,LENGTH(typearea)-2)
FROM assessdb.address002;

```

```

CREATE TABLE IF NOT EXISTS assessdb.postaddress (
  id      INT,
  postcode  STRING,
  latitude  DECIMAL(18, 9),
  longitude  DECIMAL(18, 9),
  easting   INT,
  northing  INT,
  gridref   STRING,
  district  STRING,
  ward      STRING,
  districtcode STRING,
  wardcode  STRING,
  country   STRING,

```

```
    countycode  STRING,
    constituency STRING,
    typearea    STRING
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.postaddress
SELECT
    id,
    postcode,
    latitude,
    longitude,
    easting,
    northing,
    gridref,
    district,
    ward,
    districtcode,
    wardcode,
    country,
    countycode,
    constituency,
    typearea
FROM
    assessdb.address003;
```

```
CREATE TABLE IF NOT EXISTS assessdb.addresshistory001 (
    id      STRING,
    pid     STRING,
    aid     STRING,
    did1    STRING,
    did2    STRING
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
```

```
INSERT INTO TABLE assessdb.addresshistory001
SELECT
    id,
    pid,
    aid,
    did1,
    did2
FROM
    retrievedb.rawaddresshistory
```

```
WHERE id <> "id";
```

```
CREATE TABLE IF NOT EXISTS assessdb.addresshistory002 (
    id      INT,
    pid     INT,
    aid     INT,
    did1    INT,
    did2    INT
```



```

)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE assessdb.addresshistory002
SELECT
  CAST(id as INT),
  CAST(pid as INT),
  CAST(aid as INT),
  CAST(did1 as INT),
  CAST(did2 as INT)
FROM
  assessdb.addresshistory001;

```

```

CREATE TABLE IF NOT EXISTS assessdb.addresshistory (
  id      INT,
  pid     INT,
  aid     INT,
  did1    INT,
  did2    INT
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE assessdb.addresshistory
SELECT
  id,
  pid,
  aid ,
  did1,
  did2
FROM
  assessdb.addresshistory002;
DROP TABLE assessdb.address001;
DROP TABLE assessdb.address002;
DROP TABLE assessdb.address003;

```

```

DROP TABLE assessdb.addresshistory001;
DROP TABLE assessdb.addresshistory002;
SELECT
  addresshistory.id,
  addresshistory.pid,
  personfull.firstname,
  personfull.lastname,
  addresshistory.aid,
  postaddress.postcode,
  addresshistory.did1,
  dates1.datetimes as startdate,
  addresshistory.did2,
  dates2.datetimes as enddate
FROM
  assessdb.addresshistory
JOIN
  assessdb.personfull

```

```

ON
  addresshistory.pid = personfull.persid
JOIN
  assessdb.postaddress
ON
  addresshistory.aid = postaddress.id
JOIN
  assessdb.dates as dates1
ON
  addresshistory.did1 = dates1.id
JOIN
  assessdb.dates as dates2
ON
  addresshistory.did2 = dates2.id
LIMIT 10;

```

查询时遇到了 Execution Error, return code 3 from org.apache.hadoop.hive.ql.exec.mr.MapredL
calTask

这是一个坑.

set hive.auto.convert.join=false

是mapjoin的问题.优化mapjoin的时候,触发了内存不足的阈值,但是内存可能够用

评估account表

```

CREATE TABLE IF NOT EXISTS assessdb.account001 (
  id      string,
  pid     string,
  accountno string,
  balance string
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE assessdb.account001
SELECT
  id,
  pid,
  accountno,
  balance
FROM retrievedb.rawaccount
WHERE id <> "id";
CREATE TABLE IF NOT EXISTS assessdb.account002 (
  id      string,
  pid     string,
  accountno string,
  balance string
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE assessdb.account002
SELECT
  id,
  pid,
  rtrim(ltrim(accountno)),

```

```

balance
FROM assessdb.account001;
CREATE TABLE IF NOT EXISTS assessdb.account003 (
  id      INT,
  pid     INT,
  accountid INT,
  accountno string,
  balance DECIMAL(18, 9)
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE assessdb.account003
SELECT
  CAST(id as INT),
  CAST(pid as INT),
  CAST(accountno as INT),
  CONCAT('AC',accountno),
  CAST(balance as DECIMAL(18, 9))
FROM assessdb.account002;

```

```

CREATE TABLE IF NOT EXISTS assessdb.account (
  id      INT,
  pid     INT,
  accountid INT,
  accountno string,
  balance DECIMAL(18, 9)
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE assessdb.account
SELECT
  id,
  pid,
  accountid,
  accountno,
  balance
FROM
  assessdb.account003;
DROP TABLE assessdb.account001;
DROP TABLE assessdb.account002;
DROP TABLE assessdb.account003;

```

过程数据库:

```

CREATE DATABASE IF NOT EXISTS processdb;
CREATE TABLE IF NOT EXISTS processdb.personhub (
  id      INT,
  keyid   STRING,
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (id) INTO 1 BUCKETS

```

```

STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.personhub001 (
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (firstname, lastname) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personhub001
SELECT DISTINCT
  firstname,
  lastname
FROM
  assessdb.personfull;
CREATE TABLE IF NOT EXISTS processdb.personhub002 (
  rid      BIGINT,
  tid      BIGINT,
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (rid, tid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personhub002
SELECT
  ROW_NUMBER() OVER (ORDER BY firstname, lastname), --计算每一行数据在结果集的行数
  unix_timestamp(), --时间戳
  firstname,
  lastname
FROM
  processdb.personhub001;
CREATE TABLE IF NOT EXISTS processdb.personhub003 (
  keyid  STRING,
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (keyid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personhub003
SELECT
  CONCAT(tid, '/', rid),
  firstname,
  lastname
FROM
  processdb.personhub002;

CREATE TABLE IF NOT EXISTS processdb.personhub004 (
  keyid  STRING,
  firstname STRING,
  lastname STRING,
  CDC    STRING
)

```

```

CLUSTERED BY (keyid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personhub004 --这一步是去除在目标表中已存在的数据,防止重复
据
SELECT
  A.keyid,
  A.firstname,
  A.lastname,
  B.keyid
FROM
  processdb.personhub003 AS A
LEFT JOIN
  processdb.personhub AS B
ON
  A.firstname = B.firstname AND A.lastname = B.lastname;
CREATE TABLE IF NOT EXISTS processdb.personhub005 (
  keyid  STRING,
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (keyid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personhub005 --将符合条件的数据放入005表
SELECT
  keyid,
  firstname,
  lastname
FROM
  processdb.personhub004
WHERE CDC IS NULL;

INSERT INTO TABLE processdb.personhub005
SELECT
  keyid,
  firstname,
  lastname
FROM
  processdb.personhub;

INSERT INTO TABLE processdb.personhub --给数据增加排序keyid
SELECT
  ROW_NUMBER() OVER (ORDER BY keyid),
  keyid,
  firstname,
  lastname
FROM
  processdb.personhub005;
DROP TABLE processdb.personhub001;
DROP TABLE processdb.personhub002;
DROP TABLE processdb.personhub003;
DROP TABLE processdb.personhub004;

```

```

CREATE TABLE IF NOT EXISTS processdb.personsexsatellite001 (
  keyid  STRING,
  sex    STRING
)
CLUSTERED BY (keyid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

INSERT INTO TABLE processdb.personsexsatellite001
SELECT DISTINCT
  A.keyid,
  B.sex
FROM
  processdb.personhub005 as A
JOIN
  assessdb.personfull AS B
ON
  A.firstname = B.firstname AND A.lastname = B.lastname;
CREATE TABLE IF NOT EXISTS processdb.personsexsatellite (
  id      INT,
  keyid   STRING,
  sex     STRING,
  timestmp BIGINT
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personsexsatellite
SELECT
  ROW_NUMBER() OVER (ORDER BY keyid),
  keyid,
  sex,
  unix_timestamp()
FROM
  processdb.personsexsatellite001;
DROP TABLE processdb.personsexsatellite001;

CREATE TABLE IF NOT EXISTS processdb.objecthub (
  id      int,
  objecttype string,
  objectname string,
  objectid int
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.objecthub001 (
  objecttype string,
  objectname string,
  objectid int
)
CLUSTERED BY (objecttype, objectname,objectid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```

```

INSERT INTO TABLE processdb.objecthub001
SELECT DISTINCT
  'intangible',
  'bankaccount',
  accountid
FROM
  assessdb.account;
TRUNCATE TABLE processdb.objecthub;
INSERT INTO TABLE processdb.objecthub
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY objecttype,objectname,objectid),
  objecttype,
  objectname,
  objectid
FROM
  processdb.objecthub001;

CREATE TABLE IF NOT EXISTS processdb.objectbankaccountsatellite0001 (
  accountid      int,
  transactionid  int,
  balance        DECIMAL(18, 9)
)
CLUSTERED BY (accountid,transactionid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.objectbankaccountsatellite0001
SELECT
  accountid,
  id as transactionid,
  balance
FROM
  assessdb.account;
CREATE TABLE IF NOT EXISTS processdb.objectbankaccountsatellite (
  id            int,
  accountid     int,
  transactionid int,
  balance       DECIMAL(18, 9),
  timestmp     bigint
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.objectbankaccountsatellite
SELECT
  ROW_NUMBER() OVER (ORDER BY accountid,transactionid),
  accountid,
  transactionid,
  balance,
  unix_timestamp()
FROM
  processdb.objectbankaccountsatellite0001;

DROP TABLE processdb.objectbankaccountsatellite0001;
DROP TABLE processdb.objecthub001;

```

```

SHOW TABLES;
CREATE TABLE IF NOT EXISTS processdb.locationhub (
  id      INT,
  locationtype STRING,
  locationname STRING,
  locationid INT
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.locationhub001 (
  locationtype STRING,
  locationname STRING,
  locationid INT
)
CLUSTERED BY (locationtype, locationname,locationid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.locationhub001
SELECT DISTINCT
  'intangible',
  'geospace',
  id as locationid
FROM
  assessdb.postaddress;
INSERT INTO TABLE processdb.locationhub
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY locationtype,locationname,locationid),
  locationtype,
  locationname,
  locationid
FROM
  processdb.locationhub001;
CREATE TABLE IF NOT EXISTS processdb.locationgeospacesatellite0001 (
  locationid INT,
  postcode STRING,
  latitude DECIMAL(18, 9),
  longitude DECIMAL(18, 9),
  easting INT,
  northing INT,
  gridref STRING,
  district STRING,
  ward STRING,
  districtcode STRING,
  wardcode STRING,
  country STRING,
  countycode STRING,
  constituency STRING,
  typearea STRING
)
CLUSTERED BY (locationid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```



```

INSERT INTO TABLE processdb.locationgeospacesatellite0001
SELECT
  id as locationid,
  postcode,
  latitude,
  longitude,
  easting,
  northing,
  gridref,
  district,
  ward,
  districtcode,
  wardcode,
  country,
  countycode,
  constituency,
  typearea
FROM
  assessdb.postaddress;
CREATE TABLE IF NOT EXISTS processdb.locationgeospace1satellite (
  id          INT,
  locationid  INT,
  postcode    STRING,
  timestmp   BIGINT
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

INSERT INTO TABLE processdb.locationgeospace1satellite
SELECT
  ROW_NUMBER() OVER (ORDER BY locationid),
  locationid,
  postcode,
  unix_timestamp()
FROM
  processdb.locationgeospacesatellite0001
ORDER BY locationid;

CREATE TABLE IF NOT EXISTS processdb.locationgeospace2satellite (
  id          INT,
  locationid  INT,
  latitude    DECIMAL(18, 9),
  longitude   DECIMAL(18, 9),
  timestmp   BIGINT
)
CLUSTERED BY (id, locationid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.locationgeospace2satellite
SELECT
  ROW_NUMBER() OVER (ORDER BY locationid),
  locationid,
  latitude,

```

```

longitude,
unix_timestamp()
FROM
processdb.locationgeospacesatellite0001;
CREATE TABLE IF NOT EXISTS processdb.locationgeospace3satellite (
id      INT,
locationid INT,
easting  INT,
northing INT,
timestmp  BIGINT
)
CLUSTERED BY (id, locationid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.locationgeospace3satellite
SELECT
ROW_NUMBER() OVER (ORDER BY locationid),
locationid,
easting,
northing,
unix_timestamp()
FROM
processdb.locationgeospacesatellite0001;
CREATE TABLE IF NOT EXISTS processdb.locationgeospace4satellite (
id      INT,
locationid INT,
postcode  STRING,
latitude  DECIMAL(18, 9),
longitude DECIMAL(18, 9),
easting  INT,
northing INT,
gridref  STRING,
district STRING,
ward     STRING,
districtcode STRING,
wardcode  STRING,
country  STRING,
countycode STRING,
constituency STRING,
typearea  STRING,
timestmp  BIGINT
)
CLUSTERED BY (id, locationid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.locationgeospace4satellite
SELECT
ROW_NUMBER() OVER (ORDER BY locationid),
locationid,
postcode,
latitude,
longitude,
easting,
northing,

```

```

gridref,
district,
ward,
districtcode,
wardcode,
country,
countycode,
constituency,
typearea,
unix_timestamp()
FROM
  processdb.locationgeospacesatellite0001;
DROP TABLE processdb.locationgeospacesatellite0001;
DROP TABLE processdb.locationhub001;

CREATE TABLE IF NOT EXISTS processdb.eventhub (
  id      int,
  eventtype string,
  eventname string,
  eventid int
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.eventhub001 (
  eventtype string,
  eventname string,
  eventid int
)
CLUSTERED BY (eventtype, eventname,eventid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.eventhub001
SELECT DISTINCT
  'intangible',
  'banktransaction',
  id as eventid
FROM
  assessdb.account;
INSERT INTO TABLE processdb.eventhub
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY eventtype,eventname,eventid),
  eventtype,
  eventname,
  eventid
FROM
  processdb.eventhub001;
CREATE TABLE IF NOT EXISTS processdb.eventbanktransactionsatellite0001 (
  accountid      int,
  transactionid  int,
  balance        DECIMAL(18, 9)
)
CLUSTERED BY (accountid,transactionid) INTO 1 BUCKETS
STORED AS orc

```

```

TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.eventbanktransactionsatellite0001
SELECT
  accountid,
  id as transactionid,
  balance
FROM
  assessdb.account;
CREATE TABLE IF NOT EXISTS processdb.eventbanktransactionsatellite (
  id          int,
  accountid   int,
  transactionid int,
  balance     DECIMAL(18, 9),
  timestmp    bigint
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.eventbanktransactionsatellite
SELECT
  ROW_NUMBER() OVER (ORDER BY accountid,transactionid),
  accountid,
  transactionid,
  balance,
  unix_timestamp()
FROM
  processdb.eventbanktransactionsatellite0001;
DROP TABLE processdb.eventbanktransactionsatellite0001;
DROP TABLE processdb.eventhub001;

CREATE TABLE IF NOT EXISTS processdb.timehub (
  id      INT,
  timeid  INT
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.timehub001 (
  timeid  INT
)
CLUSTERED BY (timeid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.timehub001
SELECT DISTINCT
  id as timeid
FROM
  assessdb.dates
WHERE yearnumber = 2015;
INSERT INTO TABLE processdb.timehub001
SELECT DISTINCT
  id as timeid
FROM
  assessdb.dates

```

```

WHERE yearnumber = 2016;
INSERT INTO TABLE processdb.timehub
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY timeid),
  timeid
FROM
  processdb.timehub001;
CREATE TABLE IF NOT EXISTS processdb.timesatellite0001 (
  timeid INT,
  datetimes string
)
CLUSTERED BY (timeid) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.timesatellite0001
SELECT
  id as timeid,
  datetimes
FROM
  assessdb.dates
WHERE yearnumber = 2015;
INSERT INTO TABLE processdb.timesatellite0001
SELECT
  id as timeid,
  datetimes
FROM
  assessdb.dates
WHERE yearnumber = 2016;
CREATE TABLE IF NOT EXISTS processdb.time1satellite (
  id INT,
  timeid INT,
  datetimes STRING,
  timestmp BIGINT
)
CLUSTERED BY (id) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.time1satellite
SELECT
  ROW_NUMBER() OVER (ORDER BY timeid),
  timeid,
  datetimes,
  unix_timestamp()
FROM
  processdb.timesatellite0001
ORDER BY timeid;
DROP TABLE processdb.timesatellite0001;
DROP TABLE processdb.timehub001;

CREATE TABLE IF NOT EXISTS processdb.person_person_link(
  id INT,
  personid1 INT,
  personid2 INT
)

```

```

CLUSTERED BY (id, personid1, personid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.person_person_link002(
  personid1 INT,
  personid2 INT
)
CLUSTERED BY (personid1, personid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.personlink001(
  personid INT
)
CLUSTERED BY (personid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.personlink001
SELECT
  personhub.id as personid
FROM
  processdb.personhub
LIMIT 10;
CREATE TABLE IF NOT EXISTS processdb.object_object_link(
  id INT,
  objectid1 INT,
  objectid2 INT
)
CLUSTERED BY (id, objectid1, objectid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.object_object_link002(
  objectid1 INT,
  objectid2 INT
)
CLUSTERED BY (objectid1, objectid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.objectlink001(
  objectid INT
)
CLUSTERED BY (objectid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.objectlink001
SELECT
  objecthub.id as objectid
FROM
  processdb.objecthub
LIMIT 10;
CREATE TABLE IF NOT EXISTS processdb.location_location_link(
  id INT,
  locationid1 INT,
  locationid2 INT
)

```

```

CLUSTERED BY (id, locationid1, locationid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.location_location_link002(
  locationid1 INT,
  locationid2 INT
)
CLUSTERED BY (locationid1, locationid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.locationlink001(
  locationid INT
)
CLUSTERED BY (locationid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

CREATE TABLE IF NOT EXISTS processdb.event_event_link(
  id INT,
  eventid1 INT,
  eventid2 INT
)
CLUSTERED BY (id, eventid1, eventid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.event_event_link002(
  eventid1 INT,
  eventid2 INT
)
CLUSTERED BY (eventid1, eventid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.eventlink001(
  eventid INT
)
CLUSTERED BY (eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.eventlink001
SELECT
  eventhub.id as eventid
FROM
  processdb.eventhub
LIMIT 10;

CREATE TABLE IF NOT EXISTS processdb.time_time_link(
  id INT,
  timeid1 INT,
  timeid2 INT
)
CLUSTERED BY (id, timeid1, timeid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.time_time_link002(

```

```

    timeid1 INT,
    timeid2 INT
)
CLUSTERED BY (timeid1, timeid2) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS processdb.timelink001(
    timeid INT
)
CLUSTERED BY (timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.timelink001
SELECT
    timehub.id as timeid
FROM
    processdb.timehub
LIMIT 10;
CREATE TABLE IF NOT EXISTS processdb.person_object_link002(
    personid INT,
    objectid INT
)
CLUSTERED BY (personid, objectid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_object_link002
SELECT DISTINCT
    personlink001.personid as personid,
    objectlink001.objectid as objectid
FROM
    processdb.personlink001
cROSS JOIN
    processdb.objectlink001
LIMIT 20;
INSERT INTO TABLE processdb.person_object_link002
SELECT personhub.id, objecthub.objectid
FROM assessdb.account
JOIN
    processdb.personhub
ON account.pid = personhub.id
JOIN
    processdb.objecthub
ON account.accountid = objecthub.objectid
LIMIT 100;
CREATE TABLE IF NOT EXISTS processdb.person_object_link(
    id INT,
    personid INT,
    objectid INT
)
CLUSTERED BY (id, personid, objectid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_object_link
SELECT DISTINCT

```



```

    ROW_NUMBER() OVER (ORDER BY personid, objectid),
    personid,
    objectid
FROM
    processdb.person_object_link002;
CREATE TABLE IF NOT EXISTS processdb.person_location_link002(
    personid INT,
    locationid INT
)
CLUSTERED BY (personid, locationid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_location_link002
SELECT DISTINCT
    personlink001.personid as personid,
    locationlink001.locationid as locationid
FROM
    processdb.personlink001
CROSS JOIN
    processdb.locationlink001
LIMIT 20;
INSERT INTO TABLE processdb.person_object_link002
SELECT personhub.id, objecthub.objectid
FROM assessdb.account
JOIN
    processdb.personhub
ON account.pid = personhub.id
JOIN
    processdb.objecthub
ON account.accountid = objecthub.objectid
LIMIT 100;
CREATE TABLE IF NOT EXISTS processdb.person_object_link(
    id INT,
    personid INT,
    objectid INT
)
CLUSTERED BY (id, personid, objectid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_object_link
SELECT DISTINCT
    ROW_NUMBER() OVER (ORDER BY personid, objectid),
    personid,
    objectid
FROM
    processdb.person_object_link002;
CREATE TABLE IF NOT EXISTS processdb.person_location_link002(
    personid INT,
    locationid INT
)
CLUSTERED BY (personid, locationid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_location_link002

```

```

SELECT DISTINCT
  personlink001.personid as personid,
  locationlink001.locationid as locationid
FROM
  processdb.personlink001
CROSS JOIN
  processdb.locationlink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.person_location_link(
  id INT,
  personid INT,
  locationid INT
)
CLUSTERED BY (id, personid, locationid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_location_link
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY personid, locationid),
  personid,
  locationid
FROM
  processdb.person_location_link002;

CREATE TABLE IF NOT EXISTS processdb.person_event_link002(
  personid INT,
  eventid INT
)
CLUSTERED BY (personid, eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_event_link002
SELECT DISTINCT
  personlink001.personid as personid,
  eventlink001.eventid as eventid
FROM
  processdb.personlink001
CROSS JOIN
  processdb.eventlink001
LIMIT 20;

CREATE TABLE IF NOT EXISTS processdb.person_event_link(
  id INT,
  personid INT,
  eventid INT
)
CLUSTERED BY (id, personid, eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_event_link
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY personid, eventid),
  personid,
  eventid

```

```

FROM
  processdb.person_event_link002;

CREATE TABLE IF NOT EXISTS processdb.person_time_link002(
  personid INT,
  timeid INT
)
CLUSTERED BY (personid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_time_link002
SELECT DISTINCT
  personlink001.personid as personid,
  timelink001.timeid as timeid
FROM
  processdb.personlink001
CROSS JOIN
  processdb.timelink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.person_time_link(
  id INT,
  personid INT,
  timeid INT
)
CLUSTERED BY (id, personid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.person_time_link
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY personid, timeid),
  personid,
  timeid
FROM
  processdb.person_time_link002;
CREATE TABLE IF NOT EXISTS processdb.object_location_link002(

  objectid INT,
  locationid INT
)
CLUSTERED BY (objectid, locationid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_location_link002
SELECT DISTINCT
  objectlink001.objectid,
  locationlink001.locationid
FROM
  processdb.objectlink001
CROSS JOIN
  processdb.locationlink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.object_location_link(
  id INT,
  objectid INT,

```

```

    locationid INT
)
CLUSTERED BY (id, objectid, locationid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_location_link
SELECT DISTINCT
    ROW_NUMBER() OVER (ORDER BY objectid, locationid),
    objectid,
    locationid
FROM
    processdb.object_location_link002;
CREATE TABLE IF NOT EXISTS processdb.object_event_link002(
    objectid INT,
    eventid INT
)
CLUSTERED BY (objectid, eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_event_link002
SELECT DISTINCT
    objectlink001.objectid,
    eventlink001.eventid
FROM
    processdb.objectlink001
CROSS JOIN
    processdb.eventlink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.object_event_link(
    id INT,
    objectid INT,
    eventid INT
)
CLUSTERED BY (id, objectid, eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_event_link
SELECT DISTINCT
    ROW_NUMBER() OVER (ORDER BY objectid, eventid),
    objectid,
    eventid
FROM
    processdb.object_event_link002;
CREATE TABLE IF NOT EXISTS processdb.object_time_link002(
    objectid INT,
    timeid INT
)
CLUSTERED BY (objectid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_time_link002
SELECT DISTINCT
    objectlink001.objectid,
    timelink001.timeid

```

```

FROM
  processdb.objectlink001
CROSS JOIN
  processdb.timelink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.object_time_link(
  id INT,
  objectid INT,
  timeid INT
)
CLUSTERED BY (id, objectid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_time_link002
SELECT DISTINCT
  objectlink001.objectid,
  timelink001.timeid
FROM
  processdb.objectlink001
CROSS JOIN
  processdb.timelink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.object_time_link(
  id INT,
  objectid INT,
  timeid INT
)
CLUSTERED BY (id, objectid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.object_time_link
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY objectid, timeid),
  objectid,
  timeid
FROM
  processdb.object_time_link002;
CREATE TABLE IF NOT EXISTS processdb.location_event_link002(
  locationid INT,
  eventid INT
)
CLUSTERED BY (locationid, eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.location_event_link002
SELECT DISTINCT
  locationlink001.locationid,
  eventlink001.eventid
FROM
  processdb.locationlink001
CROSS JOIN
  processdb.eventlink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.location_event_link(

```

```

id INT,
locationid INT,
eventid INT
)
CLUSTERED BY (id, locationid, eventid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.location_event_link
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY locationid, eventid),
  locationid,
  eventid
FROM
  processdb.location_event_link002;
CREATE TABLE IF NOT EXISTS processdb.location_time_link002(
  locationid INT,
  timeid INT
)
CLUSTERED BY (locationid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.location_time_link002
SELECT DISTINCT
  locationlink001.locationid,
  timelink001.timeid
FROM
  processdb.locationlink001
CROSS JOIN
  processdb.timelink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.location_time_link(
  id INT,
  locationid INT,
  timeid INT
)
CLUSTERED BY (id, locationid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.location_time_link
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY locationid, timeid),
  locationid,
  timeid
FROM
  processdb.location_time_link002;
CREATE TABLE IF NOT EXISTS processdb.event_time_link002(
  eventid INT,
  timeid INT
)
CLUSTERED BY (eventid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.event_time_link002
SELECT DISTINCT

```

```

    eventlink001.eventid,
    timelink001.timeid
FROM
    processdb.eventlink001
CROSS JOIN
    processdb.timelink001
LIMIT 20;
CREATE TABLE IF NOT EXISTS processdb.event_time_link(
    id INT,
    eventid INT,
    timeid INT
)
CLUSTERED BY (id, eventid, timeid) INTO 1 BUCKETS
STORED As orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE processdb.event_time_link
SELECT DISTINCT
    ROW_NUMBER() OVER (ORDER BY eventid, timeid),
    eventid,
    timeid
FROM
    processdb.event_time_link002;
DROP TABLE processdb.person_event_link002;
DROP TABLE processdb.person_location_link002;
DROP TABLE processdb.person_object_link002;
DROP TABLE processdb.person_person_link002;
DROP TABLE processdb.person_time_link002;
DROP TABLE processdb.personlink001;
DROP TABLE processdb.object_event_link002;
DROP TABLE processdb.object_location_link002;
DROP TABLE processdb.object_object_link002;
DROP TABLE processdb.object_time_link002;
DROP TABLE processdb.objectlink001;
DROP TABLE processdb.location_event_link002;
DROP TABLE processdb.location_location_link002;
DROP TABLE processdb.location_time_link002;
DROP TABLE processdb.locationlink001;
DROP TABLE processdb.event_event_link002;
DROP TABLE processdb.event_time_link002;
DROP TABLE processdb.eventlink001;
DROP TABLE processdb.time_time_link002;
DROP TABLE processdb.timelink001;

```

转换数据库:

```
DROP DATABASE transformdb CASCADE;
```

```
CREATE DATABASE IF NOT EXISTS transformdb;
```

```
CREATE TABLE IF NOT EXISTS transformdb.dimperson (
    personkey BIGINT,
    firstname STRING,
    lastname STRING

```

```

)
CLUSTERED BY (firstname, lastname, personkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true', 'orc.compress' = 'ZLIB', 'orc.create.index' = 'true');
CREATE TABLE IF NOT EXISTS transformdb.dimperson001 (
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (firstname, lastname) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true', 'orc.compress' = 'ZLIB', 'orc.create.index' = 'true');

INSERT INTO TABLE transformdb.dimperson001
SELECT DISTINCT
  firstname,
  lastname
FROM
  processdb.personhub;
CREATE TABLE IF NOT EXISTS transformdb.dimperson002 (
  personkey BIGINT,
  firstname STRING,
  lastname STRING
)
CLUSTERED BY (firstname, lastname, personkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true', 'orc.compress' = 'ZLIB', 'orc.create.index' = 'true');
INSERT INTO TABLE transformdb.dimperson002
SELECT
  ROW_NUMBER() OVER (ORDER BY firstname, lastname),
  firstname,
  lastname
FROM
  transformdb.dimperson001;
INSERT INTO TABLE transformdb.dimperson
SELECT
  personkey,
  firstname,
  lastname
FROM
  transformdb.dimperson002
ORDER BY firstname, lastname, personkey;
INSERT INTO TABLE transformdb.dimperson
VALUES
(999997, 'Ruff', 'Hond'),
(999998, 'Robbie', 'Rot'),
(999999, 'Helen', 'Kat');
DROP TABLE transformdb.dimperson001;
DROP TABLE transformdb.dimperson002;

CREATE TABLE IF NOT EXISTS transformdb.dimaccount (
  accountkey BIGINT,
  accountnumber INT
)

```



```

CLUSTERED BY (accountnumber,accountkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
CREATE TABLE IF NOT EXISTS transformdb.dimaccount001 (
  accountnumber INT
)
CLUSTERED BY (accountnumber) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE transformdb.dimaccount001
SELECT DISTINCT
  objectid
FROM
  processdb.objecthub
WHERE objecttype = 'intangible'
AND objectname = 'bankaccount';
CREATE TABLE IF NOT EXISTS transformdb.dimaccount002 (
  accountkey BIGINT,
  accountnumber INT
)
CLUSTERED BY (accountnumber,accountkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE transformdb.dimaccount002
SELECT DISTINCT
  ROW_NUMBER() OVER (ORDER BY accountnumber DESC),
  accountnumber
FROM
  transformdb.dimaccount001;

INSERT INTO TABLE transformdb.dimaccount
SELECT DISTINCT
  accountkey,
  accountnumber
FROM
  transformdb.dimaccount002
ORDER BY accountnumber;
INSERT INTO TABLE transformdb.dimaccount
VALUES
(88888887,208887),
(88888888,208888),
(88888889,208889);

DROP TABLE transformdb.dimaccount001;
DROP TABLE transformdb.dimaccount002;
CREATE TABLE IF NOT EXISTS transformdb.fctpersonaccount (
  personaccountkey BIGINT,
  personkey BIGINT,
  accountkey BIGINT,
  balance DECIMAL(18, 9)
)
CLUSTERED BY (personkey,accountkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```

```

CREATE TABLE IF NOT EXISTS transformdb.fctpersonaccount001 (
  personkey      BIGINT,
  accountkey     BIGINT,
  balance        DECIMAL(18, 9)
)
CLUSTERED BY (personkey,accountkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE transformdb.fctpersonaccount001
VALUES
(999997,88888887,10.60),
(999997,88888887,400.70),
(999997,88888887,-210.90),
(999998,88888888,1000.00),
(999998,88888888,1990.60),
(999998,88888888,900.70),
(999999,88888889,160.60),
(999999,88888889,180.70),
(999999,88888889,100.60),
(999999,88888889,120.90),
(999999,88888889,180.69),
(999999,88888889,130.30);
CREATE TABLE IF NOT EXISTS transformdb.fctpersonaccount002 (
  personkey      BIGINT,
  accountkey     BIGINT,
  balance        DECIMAL(18, 9)
)
CLUSTERED BY (personkey,accountkey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE transformdb.fctpersonaccount002
SELECT
CAST(personkey AS BIGINT),
CAST(accountkey AS BIGINT),
CAST(SUM(balance) AS DECIMAL(18, 9))
FROM transformdb.fctpersonaccount001
GROUP BY personkey, accountkey;
INSERT INTO TABLE transformdb.fctpersonaccount
SELECT
ROW_NUMBER() OVER (ORDER BY personkey, accountkey),
CAST(personkey AS BIGINT),
CAST(accountkey AS BIGINT),
CAST(balance AS DECIMAL(18, 9))
FROM transformdb.fctpersonaccount002;

DROP TABLE transformdb.fctpersonaccount001;
DROP TABLE transformdb.fctpersonaccount002;
CREATE TABLE IF NOT EXISTS transformdb.dimaddress(
  addresskey     BIGINT,
  postcode       STRING
)
CLUSTERED BY (addresskey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');

```

```

INSERT INTO TABLE transformdb.dimaddress
VALUES
(1,'KA12 8RR'),
(2,'FK8 1EJ'),
(3,'EH1 2NG');

CREATE TABLE IF NOT EXISTS transformdb.dimdatetime(
  datetimekey  BIGINT,
  datetimestr  STRING
)
CLUSTERED BY (datetimekey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE transformdb.dimdatetime
VALUES
(1,'2015/08/23 16h00'),
(2,'2015/10/03 17h00'),
(3,'2015/11/12 06h00');
CREATE TABLE IF NOT EXISTS transformdb.fctpersonaddressdate(
  personaddressdatekey  BIGINT,
  personkey              BIGINT,
  addresskey             BIGINT,
  datetimekey           BIGINT
)
CLUSTERED BY (datetimekey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE transformdb.fctpersonaddressdate
VALUES
(1,999997,1,1),
(2,999998,2,2),
(3,999999,3,3);

```

组织数据库:

```

CREATE DATABASE IF NOT EXISTS organisedb;
CREATE TABLE IF NOT EXISTS organisedb.dimperson LIKE transformdb.dimperson;
INSERT INTO TABLE organisedb.dimperson
SELECT
  personkey,
  firstname,
  lastname
FROM
  transformdb.dimperson
ORDER BY firstname, lastname, personkey;
CREATE TABLE IF NOT EXISTS organisedb.dimaccount LIKE transformdb.dimaccount;
INSERT INTO TABLE organisedb.dimaccount
SELECT DISTINCT
  accountkey,
  accountnumber
FROM
  transformdb.dimaccount
ORDER BY accountnumber;

```

```

CREATE TABLE IF NOT EXISTS organisedb.fctpersonaccount LIKE transformdb.fctpersonaccou
t;
INSERT INTO TABLE organisedb.fctpersonaccount
SELECT DISTINCT
  personaccountkey,
  personkey,
  accountkey,
  balance
FROM
  transformdb.fctpersonaccount
WHERE
  personaccountkey = 1
ORDER BY personaccountkey,personkey,accountkey;
CREATE TABLE IF NOT EXISTS organisedb.dimaddress(
  addresskey  BIGINT,
  postcode   STRING
)
CLUSTERED BY (addresskey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE organisedb.dimaddress
SELECT DISTINCT
  addresskey,
  postcode
FROM
  transformdb.dimaddress
ORDER BY addresskey;
CREATE TABLE IF NOT EXISTS organisedb.dimaddress(
  addresskey  BIGINT,
  postcode   STRING
)
CLUSTERED BY (addresskey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE organisedb.dimaddress
SELECT DISTINCT
  addresskey,
  postcode
FROM
  transformdb.dimaddress
ORDER BY addresskey;
CREATE TABLE IF NOT EXISTS organisedb.fctpersonaddressdate(
  personaddressdatekey  BIGINT,
  personkey              BIGINT,
  addresskey             BIGINT,
  datetetimekey         BIGINT
)
CLUSTERED BY (datetetimekey) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE organisedb.fctpersonaddressdate
SELECT
  personaddressdatekey,

```

```

    personkey,
    addresskey,
    datetimekey
FROM
    transformdb.fctpersonaddressdate
WHERE personaddressdatekey = 1
ORDER BY
    personaddressdatekey,
    personkey,
    addresskey,
    datetimekey;

```

报表数据库:

```

CREATE DATABASE IF NOT EXISTS reportdb;
CREATE TABLE IF NOT EXISTS reportdb.report001(
    firstname    STRING,
    lastname     STRING,
    accountnumber INT,
    balance      DECIMAL(18, 9)
)
CLUSTERED BY (firstname, lastname) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE reportdb.report001
SELECT
    dimperson.firstname,
    dimperson.lastname,
    dimaccount.accountnumber,
    fctpersonaccount.balance
FROM
    organisedb.fctpersonaccount
JOIN
    organisedb.dimperson
ON
    fctpersonaccount.personkey = dimperson.personkey
JOIN
    organisedb.dimaccount
ON
    fctpersonaccount.accountkey = dimaccount.accountkey;

CREATE TABLE IF NOT EXISTS reportdb.report002(
    accountnumber INT,
    last_balance   DECIMAL(18, 9)
)
CLUSTERED BY (accountnumber, last_balance) INTO 1 BUCKETS
STORED AS orc
TBLPROPERTIES('transactional' = 'true','orc.compress'='ZLIB','orc.create.index'='true');
INSERT INTO TABLE reportdb.report002
SELECT
    dimaccount.accountnumber, sum(fctpersonaccount.balance) as last_balance
FROM organisedb.fctpersonaccount
JOIN
    organisedb.dimaccount

```

On

```
fctpersonaccount.accountkey = dimaccount.accountkey  
GROUP BY fctpersonaccount.accountkey,dimaccount.accountnumber;
```

后续几个数据库转化过程，由于代码量太大，没有进行一一的注释，需要更详细的讲解，可以看《Hiv实践》这本书。可以解决大部分hive使用中的困惑。