Hive QL笔记

一、hadoop(分布式计算平台)子项目

1.Core/Common

2.Arvo :用于数据序列化的系统

3.MapReduce :是一种编程模型, 用于大规模数据集(>1TB)的并行运算

4.HDFS :分布式文件系统

5.Chukwa :开源的数据收集系统, 用于监控和分析大型分布式系统的数据

6.Hive :基于Hadoop文件系统上的数据仓库架构

7.HBase :一个分布式的、面向列的开源数据库

(1)适合于存储非结构化数据的数据库

(2)基于列而不是基于行的模式

8.Pig :是一个对大型数据集进行分析和评估的平台

二、HDFS

1.NameNode和DataNode

NameNode管理文件系统的命名空间, 维护整个文件系统的文件目录树及这个文件的索引目录, 管理客户端对文件的访问.

DataNode存储文件块, 被客户端和NameNode调用, 通过心跳定时向NameNode发送所存储的文件块信息.

2.副本存放与读取策略

默认副本数为3?, HDFS的存放策略是将一个副本存放在本地机架的节点上, 另一个副本放在同一机架的另一个节点上, 最后一个副本放在不同机架的节点上.

在读取数据时, HDFS会尽量让读取程序读取离客户端最近的副本.

四、Hive

1.Hive是一个基于Hadoop文件系统上的数据仓库架构, 它为数据仓库的管理提供了许多功能: 数据ETL工具、数据存储管理和大量数据集的查询与分析能力.

2.Hadoop是批量处理系统, 任务是高延迟性的, 在任务提交和处理过程中会消耗一些时间成本. Hive不提供数据排序和查询cache功能, 也不提供在线事务处理, 不提供实时的查询和记录级的更新, 但Hiv额能更好地处理不变的大规模数据集(如: 网络日志)上的批量任务. Hive最大的价值是可扩展性、可延展性, 并拥有良好的容错性和第约束的数据输入格式.

3.Hive有位图索引.

4.Hive主要包含4类数据模型: 表(Table)、外部表(External Table)、分区(Partition)和桶(Bucket).

5.每个表在Hive中都有一个对应的存储目录. 例:datawarehouse/htable/ds=20100301/city=Beijing/part-0000

datawarehouse: 数据仓库目录, 由${hive.metastore.warehouse.dir}指定, 所有的表数据(外部表除外)都保存在这个目录中

htable: 表名

ds=20100301: 分区

city=Beijing: 分区

part-0000: 表示哈希值为0的桶. 桶对指定列进行hash计算时, 根据哈希值切分数据, 每个桶对应一个文件.

6.Hive将元数据存储在RDBMS中, 一般为derby或mysql.

5.Hive QL

5.1.DDL(数据定义):

5.1.1.Create/Drop/Alter/Use Database

5.1.1.1.Create Database创建数据库:

create(database|schema) [if not exists] db\_name --DATABASE和SCHEMA是一样的

[comment db\_comment]

[location hdfs\_path]

[with dbproperties(pro\_name=pro\_value, ...)];

5.1.1.2.Drop Database删除数据库:

drop(database|schema) [if exists] db\_name [restrict|cascade]

默认是RESTRICT, 当db不为空时语句会失败, 想要删除db的同时删除其目录下的tables, 用CASCADE.

5.1.1.3.Alter Database:

alter (database|schema) db\_name set dbproperties(pro\_name=pro\_value, ...)

alter (database|schema) db\_name set owner[user|role] user\_or\_role

alter (database|schema) db\_name set location hdfs\_path --不会改变旧的表、数据的目录

5.1.1.4.Use Database:

use db\_name;

use default

selsct current\_database() --查看当前所使用的db

5.1.2.Create/Drop/Truncate Table

5.1.2.1.Create Table新建表:

create [temporary] [external] table [if not exists] [db\_name.]table\_name

[(col\_name data\_type [comment col\_comment], ... [constraint\_specification])] --默认是创建managed table

[comment table\_comment] --临时表不能创建分区, 索引

[partitioned by (col\_name data\_type [comment col\_comment], ...)]

[clustered by(col\_name, col\_name, ...) [sorted by(col\_name [asc|desc], ...)] into num\_buckets buckets]

--do not affect how data is inserted into a table, only how it is read

[skewed by (col\_name, col\_name, ...) ] --倾斜表

on((col\_value, col\_value, ...), (col\_value, col\_value, ...), ...)

[sorted as directories] --会建立相应目录

[

[row format row\_format] --FIELDS, COLLECTION ITEMS, MAP KEYS, LINES

[sorted as file\_format] --TEXTFILE(默认), SEQUENCEFILE, RCFILE, ORC, PARQUET, AVRO

| sorted by 'storage.handler.class.name' [with serdeproperties (...)]

]

[locationhdfs\_path]

[tblproperties (property\_name=property\_value, ...)]

[as select\_statement] --AS的表不能是partitioned table、external table、list bucketing table.

create[temporary] [external] table [if not exists] [db\_name.]table\_name

like existing\_table\_or\_view\_name

[location hdfs\_path];

5.1.2.1.1.Partitioned Tables

example:

create table page\_view(viewtime int, userid bigint,

page\_url string, referrer\_url string,

ip string comment 'ip address of the user')

comment 'this is the page view table'

partitioned by(dt string, country string)

row format delimited

fields terminated by '\001'

stored as sequencefile;

5.1.2.1.2.External Tables

example:

create external table page\_view(viewtime int, userid bigint,

page\_url string, referrer\_url string,

ip string comment 'ip address of the user',

country string comment 'country of origination')

comment 'this is the staging page view table'

row format delimited fields terminated by '\054'

stored as textfile

location '<hdfs\_location>';

5.1.2.1.3.Create Table As Select (CTAS)

example:

create table new\_key\_value\_store

row format serde "org.apache.hadoop.hive.serde2.columnar.columnarserde"

stored as rcfile

as

select (key % 1024) new\_key, concat(key, value) key\_value\_pair

from key\_value\_store

sort by new\_key, key\_value\_pair;

5.1.2.1.4.Create Table Like

create table empty\_key\_value\_store

like key\_value\_store [tblproperties (property\_name=property\_value, ...)];

example:

create table page\_view(viewtime int, userid bigint,

page\_url string, referrer\_url string,

ip string comment 'ip address of the user')

comment 'this is the page view table'

partitioned by(dt string, country string)

clustered by(userid) sorted by(viewtime) into 32 buckets

row format delimited

fields terminated by '\001'

collection items terminated by '\002'

map keys terminated by '\003'

stored as sequencefile;

5.1.2.1.5.Skewed Tables

example:

create table list\_bucket\_multiple (col1 string, col2 int, col3 string)

skewed by (col1, col2) on (('s1',1), ('s3',3), ('s13',13), ('s78',78)) [stored as directories];

5.1.2.1.6.Temporary Tables

example:

create temporary table list\_bucket\_multiple (col1 string, col2 int, col3 string);

5.1.2.1.7.Create Transactional Tables:

example:

create transactional table transactional\_table\_test(key string, value string)

partitioned by(ds string) sorted as orc;

5.1.2.1.8.Constraints创建有约束的表:

xample:

create table pk(id1 integer, id2 integer,

primary key(id1, id2) disable novalidate);

create table fk(id1 integer, id2 integer,

constraint c1 foreign key(id1, id2) references pk(id2, id1) disable novalidate);

5.1.2.2.Drop Table删除表:

drop table [if exists] table\_name [purge]

5.1.2.3.Truncate Table删除表数据:

truncate table table\_name [partition(partition\_column = partition\_col\_value, ...)];

5.1.3.Alter Table/Partition/Column

5.1.3.1.Alter Table:

5.1.3.1.1.Rename Table重命名:

alter table table\_name rename to new\_table\_name --当建表语句中没有location时移动表

5.1.3.1.2.Alter Table Properties更改表属性:

alter table table\_name set tblproperties (property\_name = property\_value, ... )

5.1.3.1.2.1.Alter Table Comment:

alter table table\_name set tblproperties ('comment' = new\_comment);

5.1.3.1.3.Add SerDe Properties:

alter table table\_name [partition partition\_column = partition\_col\_value, ...)] set serde

serde\_class\_name [with serdeproperties (property\_name = property\_value, ... )];

alter table table\_name [partition partition\_column = partition\_col\_value, ...)] set serdeproperties

(property\_name = property\_value, ... )];

5.1.3.1.4.Alter Table Storage Properties:

alter table table\_name clustered by (col\_name, ...) [sorted by (col\_name, ...)] into num\_buckets buckets;

5.1.3.1.5.Alter Table Skewed or Stored as Directories

5.1.3.1.5.1.Alter Table Skewed:

alter table table\_name skewed by (col\_name1, col\_name2, ...)

on([(col\_name1\_value, col\_name2\_value, ...) [, (col\_name1\_value, col\_name2\_value), ...]

[stored as directories];

5.1.3.1.5.2.Alter Table Not Skewed:

alter table table\_name not skewed;

5.1.3.1.5.3.Alter Table Not Stored as Directories:

alter table table\_name not stored as directories;

5.1.3.1.5.4.Alter Table Set Skewed Location

5.1.3.1.6.Alter Table Constraints:

alter table table\_name add constraint constraint\_name primary key (column, ...) disable novalidate;

alter table table\_name add constraint constraint\_name foreign key (column, ...) references

table\_name(column, ...) disable novalidate rely;

alter table table\_name drop constraint constraint\_name;

5.1.3.1.7.Additional Alter Table Statements

5.1.3.2.Alter Partition:

5.1.3.2.1.Add Partitions:

alter table table\_name add [if not exists] partition (partition\_column = partition\_col\_value, ...)

[location 'location'][, partition partition\_spec [location 'location'], ...];

5.1.3.2.2.Rename Partition:

alter table table\_name partition partition\_spec rename to partition partition\_spec;

5.1.3.2.3.Exchange Partition:

例: alter table table\_name\_2 exchange partition (partition\_spec, partition\_spec2, ...) with table table\_name\_1

--交换多个分区

5.1.3.2.4.Recover Partitions (MSCK REPAIR TABLE):

alter table table\_name recover partitions;

5.1.3.2.5.Drop Partitions删除分区:

alter table table\_name drop [if exists] partition partition\_spec[, partition partition\_spec, ...] [purge];

purge: 删除之后不可恢复.

5.1.3.2.6.(Un)Archive Partition:

alter table table\_name archive partition partition\_spec;

alter table table\_name unarchive partition partition\_spec;

5.1.3.3.Alter Either Table or Partition

5.1.3.3.1.Alter Table/Partition File Format:

alter table table\_name [partition partition\_spec] set fileformat file\_format;

5.1.3.3.2.Alter Table/Partition Location:

alter TABLE table\_name [partition partition\_spec] set location"new location";

5.1.3.3.3.Alter Table/Partition Touch:

alter TABLE table\_name touch [partition partition\_spec];

5.1.3.3.4.Alter Table/Partition Protections

5.1.3.3.5.Alter Table/Partition Compact

alter table table\_name [partition (partition\_key = 'partition\_value' [, ...])] compact 'compaction\_type'[and wait]

[with overwrite tblproperties ("property"="value" [, ...])];

5.1.3.3.6.Alter Table/Partition Concatenate

alter table table\_name [partition (partition\_key = 'partition\_value' [, ...])] concatenate;

5.1.3.3.7.Alter Table/Partition Update columns

alter table table\_name [partition (partition\_key = 'partition\_value' [, ...])] update columns;

5.1.3.4.Alter Column:

5.1.3.4.1.Change Column Name/Type/Position/Comment:

alter table table\_name [partition partition\_spec] change [column] col\_old\_name col\_new\_name column\_type

[comment col\_comment] [first|after column\_name] [cascade|restrict];

--ALTER TABLE CHANGE COLUMN CASCADE子句将覆盖表分区的列元数据，而不管表或分区的保护模式如何。

example:

create table test\_change (a int, b int, c int);

alter table test\_change change a a1 int; --first change column a's name to a1.

alter table test\_change change a1 a2 string after b; --change a1's name to a2, data type to string, and after b.

--the new table's structure is: b int, a2 string, c int.

alter table test\_change change c c1 int first; --change column c's name to c1, and put it as the first column.

--the new table's structure is: c1 int, b int, a2 string.

alter table test\_change change a1 a1 int comment 'this is column a1'; --add a comment to column a1

5.1.3.4.2.Add/Replace Columns

alter table table\_name

[partition partition\_spec]

add|replace columns (col\_name data\_type [comment col\_comment], ...)

[cascade|restrict]

5.1.3.4.3.Partial Partition Specification:

Example:

alter table foo partition (ds=’2008-04-08’, hr) change column dec\_col\_name dec\_col\_name decimal(38,18);

5.1.4.Create/Drop/Alter View

5.1.4.1.Create View新建视图

create view [if not exists] [db\_name.]view\_name [(column\_name [comment column\_comment], ...) ]

[comment view\_comment]

[tblproperties (property\_name = property\_value, ...)]

as select ...;

show views: 显示数据库中的视图列表

example:

create view onion\_referrers(url comment 'url of referring page')

comment 'referrers to the onion website'

as

select distinct referrer\_url

from page\_view

where page\_url='http://www.theonion.com';

5.1.4.2.Drop View

drop view [if exists] [db\_name.]view\_name

example:

drop view onion\_referrers;

5.1.4.3.Alter View Properties

alter view [db\_name.]view\_name set tblproperties (pro\_name=pro\_value, pro\_name1=pao\_value1,...);

5.1.4.4.Alter View As Select

alter view [db\_name.]view\_name as select\_statement;

5.1.5.Create/Drop/Alter Index

5.1.5.1.Create Index

create index index\_name

on table base\_table\_name (col\_name, ...)

as index\_type

[with deferred rebuild]

[idxproperties (property\_name=property\_value, ...)]

[in table index\_table\_name]

[

[ row format ...] stored as ...

| stored by ...

]

[location hdfs\_path]

[tblproperties (...)]

[comment "index comment"];

5.1.5.2.Drop Index

drop index [if exists] index\_name on table\_name;

5.1.5.3.Alter Index

alter index index\_name on table\_name [partition partition\_spec] rebuild;

5.1.6.Create/Drop Macro

5.1.6.1.Create Temporary Macro

create temporary macro macro\_name([col\_name col\_type, ...]) expression;

examples:

create temporary macro fixed\_number() 42;

create temporary macro string\_len\_plus\_two(x string) length(x) + 2;

create temporary macro simple\_add (x int, y int) x + y;

5.1.6.2.Drop Temporary Macro

drop temporary macro [if exists] macro\_name;

5.1.7.Create/Drop/Reload Function

5.1.7.1.Create Temporary Function

create temporary function function\_name as class\_name;

5.1.7.2.Drop Temporary Function

drop temporary function [if exists] function\_name;

5.1.7.3.Create Function

create function [db\_name.]function\_name as class\_name

[using jar|file|archive 'file\_uri' [, jar|file|archive 'file\_uri'] ];

5.1.7.4.Drop Function

drop function [if exists] function\_name;

5.1.7.5.Reload Function

reload function;

5.1.8.Create/Drop/Grant/Revoke Roles and Privileges

5.1.8.1.Show Databases

show (databases|schemas) [like 'identifier\_with\_wildcards'];

5.1.8.2.Show Tables/Views/Partitions/Indexes

5.1.8.2.1.Show Tables

show tables [in database\_name] ['identifier\_with\_wildcards'];

5.1.8.2.2.Show Views

show views [in/from database\_name] [like 'pattern\_with\_wildcards'];

examples:

show views; -- show all views in the current database

show views 'test\_\*'; -- show all views that start with "test\_"

show views '\*view2'; -- show all views that end in "view2"

show views like 'test\_view1|test\_view2'; -- show views named either "test\_view1" or "test\_view2"

show views from test1; -- show views from database test1

show views in test1; -- show views from database test1 (from and in are same)

show views in test1 "test\_\*"; -- show views from database test2 that start with "test\_"

5.1.8.2.3.Show Partitions

show partitions [db\_name.]table\_name [partition(partition\_spec)];

example:

show partitions databasefoo.tablebar partition(ds='2010-03-03', hr='12');

5.1.8.2.4.Show Table/Partition Extended

5.1.8.2.5.Show Table Properties

show tblproperties tblname;

show tblproperties tblname("foo");

5.1.8.2.6.Show Create Table

show create table ([db\_name.]table\_name|view\_name);

5.1.8.2.7.Show Indexes

show [formatted] (index|indexes) on table\_with\_index [(from|in) db\_name];

5.1.8.3.Show Columns

show columns (from|in) table\_name [(from|in) db\_name] [ like 'pattern\_with\_wildcards'];

examples:

create database test\_db;

use test\_db;

create table foo(col1 int, col2 int, col3 int, cola int, colb int, colc int, a int, b int, c int);

-- show columns basic syntax

show columns from foo; -- show all column in foo

show columns from foo "\*"; -- show all column in foo

show columns in foo "col\*"; -- show columns in foo starting with "col"

show columns from foo '\*c'; -- show columns in foo ending with "c"

show columns from foo like "col1|cola"; -- show columns in foo either col1 or cola

show columns from foo from test\_db like 'col\*'; -- show columns in foo starting with "col"

show columns in foo in test\_db like 'col\*'; -- show columns in foo starting with "col" (from/in same)

5.1.8.4.Show Functions

show functions regular\_expression; --show functions “.\*” 所有函数

5.1.8.5.Show Granted Roles and Privileges

5.1.8.6.Show Locks

5.1.8.7.Show Conf

5.1.8.8.Show Transactions

5.1.8.9.Show Compactions

5.2.DML

5.2.1.Loading files into tables

load data [local] inpath 'filepath' [overwrite] into table tablename [partition (partcol1=val1, partcol2=val2 ...)]

[inputformat 'inputformat' serde 'serde']

example:

create table tab1 (col1 int, col2 int) partitioned by (col3 int) stored as orc;

load data local inpath 'filepath' into table tab1;

5.2.2.Inserting data into Hive Tables from queries

standard syntax:

insert overwrite table tablename1 [partition (partcol1=val1, partcol2=val2 ...) [if not exists]] select\_statement1 from from\_statement;

insert into table tablename1 [partition (partcol1=val1, partcol2=val2 ...)] select\_statement1 from from\_statement;

hive extension (multiple inserts):

from from\_statement

insert overwrite table tablename1 [partition (partcol1=val1, partcol2=val2 ...) [if not exists]] select\_statement1

[insert overwrite table tablename2 [partition ... [if not exists]] select\_statement2]

[insert into table tablename2 [partition ...] select\_statement2] ...;

from from\_statement

insert into table tablename1 [partition (partcol1=val1, partcol2=val2 ...)] select\_statement1

[insert into table tablename2 [partition ...] select\_statement2]

[insert overwrite table tablename2 [partition ... [if not exists]] select\_statement2] ...;

hive extension (dynamic partition inserts):

insert overwrite table tablename partition (partcol1[=val1], partcol2[=val2] ...) select\_statement from from\_statement;

insert into table tablename partition (partcol1[=val1], partcol2[=val2] ...) select\_statement from from\_statement;

example:

from page\_view\_stg pvs

insert overwrite table page\_view partition(dt='2008-06-08', country)

select pvs.viewtime, pvs.userid, pvs.page\_url, pvs.referrer\_url, null, null, pvs.ip, pvs.cnt

5.2.3.Writing data into the filesystem from queries

standard syntax:

insert overwrite [local] directory directory1

[row format row\_format] [stored as file\_format] (note: only available starting with hive 0.11.0)

select ... from ...

hive extension (multiple inserts):

from from\_statement

insert overwrite [local] directory directory1 select\_statement1

[insert overwrite [local] directory directory2 select\_statement2] ...

5.2.4.Inserting values into tables from SQL

insert into table tablename [partition (partcol1[=val1], partcol2[=val2] ...)] values values\_row [, values\_row ...]

examples:

create table students (name varchar(64), age int, gpa decimal(3, 2))

clustered by (age) into 2 buckets stored as orc;

insert into table students

values ('fred flintstone', 35, 1.28), ('barney rubble', 32, 2.32);

create table pageviews (userid varchar(64), link string, came\_from string)

partitioned by (datestamp string) clustered by (userid) into 256 buckets stored as orc;

insert into table pageviews partition (datestamp = '2014-09-23')

values ('jsmith', 'mail.com', 'sports.com'), ('jdoe', 'mail.com', null);

insert into table pageviews partition (datestamp)

values ('tjohnson', 'sports.com', 'finance.com', '2014-09-23'), ('tlee', 'finance.com', null, '2014-09-21');

insert into table pageviews

values ('tjohnson', 'sports.com', 'finance.com', '2014-09-23'), ('tlee', 'finance.com', null, '2014-09-21');

5.2.5.Update

update tablename set column = value [, column = value ...] [where expression]

5.2.6.Delete

delete from tablename [where expression]

5.2.7.Merge

merge into <target table> as t using <source expression/table> as s

on <boolean expression1>

when matched [and <boolean expression2>] then update set <set clause list>

when matched [and <boolean expression3>] then delete

when not matched [and <boolean expression4>] then insert values<value list>

partition\_spec:

: (partition\_column = partition\_col\_value, partition\_column = partition\_col\_value, ...)

serde\_properties:

: (property\_name = property\_value, property\_name = property\_value, ... )

创建表: CREATE [EXTERNAL] TABLE [IF NOT EXIT] table\_name

(column\_name type [COMMENT col\_comment], ...)

[COMMENT table\_comment]

[PARTITIONED BY (col\_name type, ...)]

[CLUSTERED BY (col\_name, ...) [SORTED BY (col\_name, ...)] INTO num\_buckets BUCKETS]

[ROW FORMAT row\_format] :FIELDS :字段分割; COLLECTION ITEMS :type为array时, 分割Array; MAP KEYS :type为map时, 每组K-V对内部分割; LINES :换行符分割, | SERDE serde\_name [WITH SERDEPROPERTIES (property\_na me=property\_value, property\_name=property\_value, ...)]

例: create table psn (id int, name string, hobbies ARRAY <string>, address MAP <string, string>)

row format delimited fields terminated by '\001' collection items terminated by '-'

map keys terminated by ':' lines terminated by \n;

[STORED AS file\_format] :file\_format 可以有4个选项: TEXTFILE(默认)、SEQUENCEFILE、RCFILE、ORCFILE, RCFILE由于列式存储方式, 数据加载时性能消耗较大, 但是具有较好的压缩比和查询响应.

[LOCATION hdfs\_path]

[AS select\_statement]

CREATE [EXTERNAL] TABLE [IF NOT EXIT] table\_name

LIKE exiting\_table\_name

[LOCATION hdfs\_path]

b.view

c.function

d.index

删除表 :drop table [if exits] schema\_name.table\_name :删除表的元数据和数据, 当是外部表时, 只删除元数据

修改表、分区:

增加分区 :alter table table\_name add partition (dt=’2010-08-08’, country=’US’) location ‘/path/to/us/part080 808’

partition (dt=’2010-08-09’, country=’US’) location ‘/path/to/uS/part080809’;

删除分区 :alter table table\_name drop partition (dt=’2010-08-08’, country=’US’), 元数据和数据都被删除

重命名表 :alter table table\_name rename as new\_table\_name

改变列名/类型/位置/注释 :alter table table\_name change [column] col\_old\_name col\_new\_name column\_type

[comment col\_comment] [first | after col\_name] 例 :alter table table1 change a a1 string after b

--列的改变只会更改元数据(元数据存储在RDBMS中), 而不会更改实际数据.

增加/更新列 :alter table add | replace columns (col\_name col\_type [comment col\_comment], ...)

--只有在使用native 的SerDe时才可以这么做

增加表属性 :alter table table\_name add tblproperties (property\_name=property\_value, ...)

增加SerDe属性 :alter table table\_name set serdeproperties (property\_name=property\_value, ...)

alter table table\_name set serde serde\_class\_name [with serdeproperties (property\_name=property\_value, ...)]

改变表文件格式和组织 :alter table table\_name set fileformat file\_format

alter table table\_name clustered by (col\_name, ...)[sorted by (col\_name, ...)] into num\_buckets buckets

创建/删除视图 :create view (if not exits) view\_name [(col\_name[comment col\_com], ...)] [comment view\_comment] as select ... drop view view\_name

创建/删除函数 :create temporary function function\_name as class\_name; drop temporary function function\_name

展示描述语句:

显示表: show table identifier\_with\_wildcards: s, \*, |

显示分区: show partitions table\_name

显示表/分区拓展: show table extended [in | from db\_name] like identifier\_with\_wildcards [partition(partition\_d esc)]

显示函数: show functions ‘a.\*’

描述表/列(通常用于调试): describe [extended] table\_name[dot col\_name]

describe [extended] table\_name [dot col\_name ([dot field\_name] | [dot ‘$elem$’] | [dot ‘$keys$’] | [dot ‘$value$’] )\* ]

描述分区: describe [extend] table\_name partition\_spec

(2)DML(数据操作, Data Manipulation Language)

8.在join时, reducer会缓存join序列中除了最后一个表的所有表的记录, 再通过最后一个表将结果序列化到文件系统. 这一实现有助于在reduce端减少内存的使用量. 实践中, 应该把最大的那以表写到最后(否则浪费大量内存).

6.Hive QL实例

6.1.建数据库/删数据库

6.1.1.建数据库

create database [if not exists] userdb;

create schema userdb;

6.1.2列举数据库

show databases;

6.1.3.删除数据库

drop database if exists userdb;

drop schema userdb;

drop database if exists userdb cascade;--意味着删除数据库之前, 要全部删除相应的表

6.2.建表/删表

6.2.1.直接建表

create table if not exists employee (

eid int,

name string,

salary string,

destination string)

comment ‘employee details’

row format delimited

fields terminated by ‘\t’

lines terminated by ‘\n’

stored as textfile

--partitioned by (

--ym string comment '月份'

--);

6.2.2复制其他表的表结构

create table if not exists new\_table like old\_table;

6.2.3.从其他表选取数据创建并插入新表

create table if not exists new\_table as

select \* from old\_table

6.2.4.删表

drop table if exists employee

6.2.5.load data

load data local inpath '/home/user/sample.txt' overwrite into table employee partition (y='2016',m='12');

--overwrite 表示覆盖表中数据

6.2.6.change

alter table employee name ename string;

alter table employee salary salary double;

6.2.7.重命名表名

alter table employee rename to emp;

6.2.8.增加列

alter table employee add columns (dept string comment 'department name');

6.2.9.替换列

alter table employee replace columns (

eid int empid int,

ename string name string);

6.3.Alter更改表结构

6.3.1.添加分区

alter table employee add partition (year='2013') location '/2013/part2013';

6.3.2.删除分区

alter table employee drop if exists partition (year='1203');

6.3.3.重命名分区

alter table employee partition (year='1203') rename to partition (yoj='1203');

(4)删除列

alter table table\_name drop column id;

(5)增加列

alter table table\_name add clomuns (id string comment '代号')

(6)修改列 (此处可用于 修改字段注释)

alter table table\_name change id level string comment '层级代号'; --将id改为level

(7)替换列

alter table table\_name replace columns (id\_new string COMMENT '新字段1', level\_new string COMMENT '2');

4.insert插入

(1)插入单条数据(Hive已支持单条插入)

insert into table\_name values(‘201705’,'lol')

(2)插入分区表

insert overwrite table table\_name partition(y='2017',m='01')

select \* from table\_name\_2 where pt = concat('2017','01');

5.其他语句

(1)列举库或表

SHOW DATABASES/TABLES;

(2)根据关键字查找库或表

SHOW DATABASES/TABLES LIKE ‘\*keyword\*’

(3)列举所有函数

SHOW FUNCTIONS;

(4)查看分区

SHOW PARTITIONS test\_table;

(5)查看建表语句

SHOW CREATE TABLE table\_name;

(6)详细描述，包括建表时间、最后更新时间、行数统计等。

DESC FORMATTED test\_table;

(7)解释语句

EXPLAIN select \* from dual;

(8)清空表

truncate table table\_name;

6.视图和索引

(1)创建视图

create view emp\_30000 AS select \* from employee where salary>30000;

(2)删除视图

drop view emp\_30000

(3)创建索引

create index inedx\_salary on table employee (salary) as

'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler';

(4)删除索引

drop index inedx\_salary on table employee

7.查询语句

(1)select where

select \* from employee where salary>30000

(2)select order by

select id, name, dept from employee order by dept;

(3)select group by

select dept,count(\*) from employee group by dept;

(4)join

select c.id, c.name, c.age, o.amount from customers c join orders o on (c.id = o.customer\_id);