第 4 章 DDL 数据定义

4.1 创建数据库

1）创建一个数据库，数据库在 HDFS 上的默认存储路径是/user/hive/warehouse/*.db。

```
hive (default)> create database db_hive;
```

2）避免要创建的数据库已经存在错误，增加 if not exists 判断。（标准写法）

```
hive> create database db_hive;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask.
Database db_hive already exi
Sts
hive (default)> create database if not exists db_hive;
```

3）创建一个数据库，指定数据库在 HDFS 上存放的位置

```
hive (default)> create database db_hive2 location '/db_hive2.db';
```

4.2 修改数据库

用户可以使用 ALTER DATABASE 命令为某个数据库的 DBPROPERTIES 设置键-值对属性值，来描述这个数据库的属性信息。数据库的其他元数据信息都是不可更改的，包括数据库名和数据库所在的目录位置。

```
hive (default)> alter database db_hive set dbproperties('createtime'='20170830');
```

在 mysql 中查看修改结果

```
hive> desc database extended db_hive;
```

db_name comment location owner_name owner_type parameters

db_hive hdfs://hadoop102:8020/user/hive/warehouse/db_hive.db atguigu USER {createtime=20170830}

4.3 查询数据库

4.3.1 显示数据库

1）显示数据库

```
hive> show databases;
```

2）过滤显示查询的数据库

```
hive> show databases like 'db_hive*';
OK
db_hive
db_hive_1
```

4.3.2 查看数据库详情

1）显示数据库信息

```
hive> desc database db_hive;
OK
db_hive         hdfs://hadoop102:8020/user/hive/warehouse/db_hive.db      atguiguUSER
```

2）显示数据库详细信息，extended

```
hive> desc database extended db_hive;
OK
db_hive         hdfs://hadoop102:8020/user/hive/warehouse/db_hive.db      atguiguUSER
```

4.3.3 切换当前数据库

```
hive (default)> use db_hive;
```

4.4 删除数据库

1）删除空数据库

```
hive>drop database db_hive2;
```

2）如果删除的数据库不存在，最好采用 if exists 判断数据库是否存在

```
hive> drop database db_hive2;
FAILED: SemanticException [Error 10072]: Database does not exist: db_hive
```

```
hive> drop database if exists db_hive2;
```

3）如果数据库不为空，可以采用 cascade 命令，强制删除

```
hive> drop database db_hive;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask.
InvalidOperationException(message:Database db_hive is not empty. One or more tables
```

```
exist.)
```

```
hive> drop database db_hive cascade;
```

4.5 创建表

1）建表语法

```
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] table_name
    [(col_name data_type [COMMENT col_comment], ...)]
    [COMMENT table_comment]
    [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]
    [ROW FORMAT row_format]
    [STORED AS file_format]
    [LOCATION hdfs_path]
```

2）字段解释说明：

（1）CREATE TABLE 创建一个指定名字的表。如果相同名字的表已经存在，则抛出异常；用户可以用 IF NOT EXISTS 选项来忽略这个异常。

（2）EXTERNAL 关键字可以让用户创建一个外部表，在建表的同时指定一个指向实际数据的路径（LOCATION），Hive 创建内部表时，会将数据移动到数据仓库指向的路径；若创建外部表，仅记录数据所在的路径，不对数据的位置做任何改变。在删除表的时候，内部表的元数据和数据会被一起删除，而外部表只删除元数据，不删除数据。

（3）COMMENT：为表和列添加注释。

（4）PARTITIONED BY 创建分区表

（5）CLUSTERED BY 创建分桶表

（6）SORTED BY 不常用

（7）ROW FORMAT

DELIMITED [FIELDS TERMINATED BY char] [COLLECTION ITEMS TERMINATED BY char]

[MAP KEYS TERMINATED BY char] [LINES TERMINATED BY char]

| SERDE serde_name [WITH SERDEPROPERTIES (property_name=property_value, property_name=property_value, ...)]

用户在建表的时候可以自定义 SerDe 或者使用自带的 SerDe。如果没有指定 ROW FORMAT 或者 ROW FORMAT DELIMITED，将会使用自带的 SerDe。在建表的时候，用户

还需要为表指定列，用户在指定表的列的同时也会指定自定义的 SerDe，Hive 通过 SerDe 确定表的具体的列的数据。

（8）STORED AS 指定存储文件类型

常用的存储文件类型：SEQUENCEFILE（二进制序列文件）、TEXTFILE（文本）、RCFILE（列式存储格式文件）

如果文件数据是纯文本，可以使用 STORED AS TEXTFILE。如果数据需要压缩，使用 STORED AS SEQUENCEFILE。

（9）LOCATION ：指定表在 HDFS 上的存储位置。

（10）LIKE 允许用户复制现有的表结构，但是不复制数据。

4.5.1 管理表

1）理论

默认创建的表都是所谓的管理表，有时也被称为内部表。因为这种表，Hive 会（或多或少地）控制着数据的生命周期。Hive 默认情况下会将这些表的数据存储在由配置项 hive.metastore.warehouse.dir(例如，/user/hive/warehouse)所定义的目录的子目录下。当我们删除一个管理表时，Hive 也会删除这个表中数据。管理表不适合和其他工具共享数据。

2）案例实操

（1）普通创建表

```
create table if not exists student2(
id int, name string
)
row format delimited fields terminated by '\t'
stored as textfile
location '/user/hive/warehouse/student2';
```

（2）根据查询结果创建表（查询的结果会添加到新创建的表中）

```
create table if not exists student3
as select id, name from student;
```

（3）根据已经存在的表结构创建表

```
create table if not exists student4 like student;
```

（4）查询表的类型

```
hive (default)> desc formatted student2;
```

Table Type: MANAGED_TABLE

4.6 分区表

分区表实际上就是对应一个 HDFS 文件系统上的独立的文件夹，该文件夹下是该分区所有的数据文件。Hive 中的分区就是分目录，把一个大的数据集根据业务需要分割成小的数据集。在查询时通过 WHERE 子句中的表达式选择查询所需要的指定的分区，这样的查询效率会提高很多。

4.5.2 外部表

1）理论

因为表是外部表，所以 Hive 并非认为其完全拥有这份数据。删除该表并不会删除掉这份数据，不过描述表的元数据信息会被删除掉。

2）管理表和外部表的使用场景：

每天将收集到的网站日志定期流入 HDFS 文本文件。在外部表（原始日志表）的基础上做大量的统计分析，用到的中间表、结果表使用内部表存储，数据通过 SELECT+INSERT 进入内部表。

3）案例实操

分别创建部门和员工外部表，并向表中导入数据。

（1）原始数据


dept.txt


emp.txt

（2）建表语句

创建部门表

```
create external table if not exists default.dept(
deptno int,
dname string,
loc int
)
row format delimited fields terminated by '\t';
```

创建员工表

```
create external table if not exists default.emp(
empno int,
ename string,
job string,
mgr int,
```

```
hiredate string,
sal double,
comm double,
deptno int)
row format delimited fields terminated by '\t';
```

（3）查看创建的表

```
hive (default)> show tables;
OK
tab_name
dept
emp
```

（4）向外部表中导入数据

导入数据

```
hive (default)> load data local inpath '/opt/module/datas/dept.txt' into table default.dept;
hive (default)> load data local inpath '/opt/module/datas/emp.txt' into table default.emp;
```

查询结果

```
hive (default)> select * from emp;
hive (default)> select * from dept;
```

（5）查看表格式化数据

```
hive (default)> desc formatted dept;
Table Type:                EXTERNAL_TABLE
```

4.6.1 分区表基本操作

1）引入分区表（需要根据日期对日志进行管理）

/user/hive/warehouse/log_partition/20170702/20170702.log

/user/hive/warehouse/log_partition/20170703/20170703.log

/user/hive/warehouse/log_partition/20170704/20170704.log

2）创建分区表语法

```
hive (default)> create table dept_partition(
                deptno int, dname string, loc string
                )
```

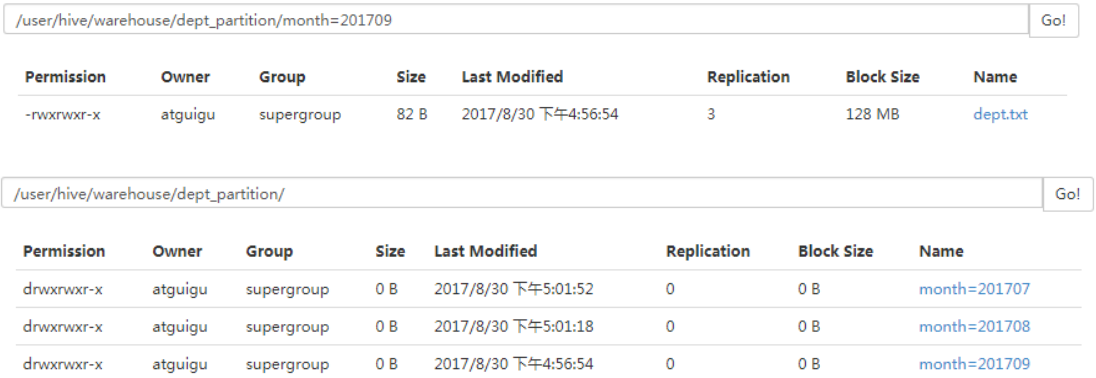
```
partitioned by (month string)
row format delimited fields terminated by '\t';
```

3）加载数据到分区表中

hive (default)> load data local inpath '/opt/module/datas/dept.txt' into table default.dept_partition partition(month='201709');

hive (default)> load data local inpath '/opt/module/datas/dept.txt' into table default.dept_partition partition(month='201708');

hive (default)> load data local inpath '/opt/module/datas/dept.txt' into table default.dept_partition partition(month='201707');



4）查询分区表中数据

单分区查询

hive (default)> select * from dept_partition where month='201709';

多分区联合查询

```
hive (default)> select * from dept_partition where month='201709'
                union
                select * from dept_partition where month='201708'
                union
                select * from dept_partition where month='201707';
```

```
_u3.deptno      _u3.dname       _u3.loc _u3.month
10      ACCOUNTING      NEW YORK        201707
10      ACCOUNTING      NEW YORK        201708
10      ACCOUNTING      NEW YORK        201709
20      RESEARCH        DALLAS  201707
```

```
20      RESEARCH        DALLAS  201708
20      RESEARCH        DALLAS  201709
30      SALES   CHICAGO 201707
30      SALES   CHICAGO 201708
30      SALES   CHICAGO 201709
40      OPERATIONS      BOSTON  201707
40      OPERATIONS      BOSTON  201708
40      OPERATIONS      BOSTON  201709
```

5）增加分区

创建单个分区

```
hive (default)> alter table dept_partition add partition(month='201706') ;
```

同时创建多个分区

```
hive (default)> alter table dept_partition add partition(month='201705') partition(month='201704');
```

6）删除分区

删除单个分区

```
hive (default)> alter table dept_partition drop partition (month='201704');
```

同时删除多个分区

```
hive (default)> alter table dept_partition drop partition (month='201705'), partition (month='201706');
```

7）查看分区表有多少分区

```
hive> show partitions dept_partition;
```

8）查看分区表结构

```
hive> desc formatted dept_partition;

# Partition Information
# col_name              data_type               comment
month                   string
```

4.6.2 分区表注意事项

1）创建二级分区表

```
hive (default)> create table dept_partition2(
                deptno int, dname string, loc string
                )
                partitioned by (month string, day string)
                row format delimited fields terminated by '\t';
```

2）正常的加载数据

（1）加载数据到二级分区表中

```
hive (default)> load data local inpath '/opt/module/datas/dept.txt' into table default.dept_partition2 partition(month='201709', day='13');
```

（2）查询分区数据

```
hive (default)> select * from dept_partition2 where month='201709' and day='13';
```

3）把数据直接上传到分区目录上，让分区表和数据产生关联的两种方式

（1）方式一：上传数据后修复

上传数据

```
hive (default)> dfs -mkdir -p /user/hive/warehouse/dept_partition2/month=201709/day=12;
hive (default)> dfs -put /opt/module/datas/dept.txt /user/hive/warehouse/dept_partition2/month=201709/day=12;
```

查询数据（查询不到刚上传的数据）

```
hive (default)> select * from dept_partition2 where month='201709' and day='12';
```

执行修复命令

```
hive> msck repair table dept_partition2;
```

再次查询数据

```
hive (default)> select * from dept_partition2 where month='201709' and day='12';
```

（2）方式二：上传数据后添加分区

上传数据

```
hive (default)> dfs -mkdir -p /user/hive/warehouse/dept_partition2/month=201709/day=11;
hive (default)> dfs -put /opt/module/datas/dept.txt /user/hive/warehouse/dept_partition2/month=201709/day=11;
```

执行添加分区

```
hive (default)> alter table dept_partition2 add partition(month='201709', day='11');
```

查询数据

```
hive (default)> select * from dept_partition2 where month='201709' and day='11';
```

（3）方式三：上传数据后 load 数据到分区

创建目录

```
hive (default)> dfs -mkdir -p /user/hive/warehouse/dept_partition2/month=201709/day=10;
```

上传数据

```
hive (default)> load data local inpath '/opt/module/datas/dept.txt' into table dept_partition2 partition(month='201709',day='10');
```

查询数据

```
hive (default)> select * from dept_partition2 where month='201709' and day='10';
```

4.7 修改表

4.7.1 重命名表

（1）语法

```
ALTER TABLE table_name RENAME TO new_table_name
```

（2）实操案例

```
hive (default)> alter table dept_partition2 rename to dept_partition3;
```

4.7.2 增加、修改和删除表分区

详见 4.6.1 分区表基本操作。

4.7.3 增加/修改/替换列信息

1）语法

更新列

```
ALTER TABLE table_name CHANGE [COLUMN] col_old_name col_new_name column_type [COMMENT col_comment] [FIRST|AFTER column_name]
```

增加和替换列

```
ALTER TABLE table_name ADD|REPLACE COLUMNS (col_name data_type [COMMENT col_comment], ...)
```

注：ADD 是代表新增一字段，字段位置在所有列后面(partition 列前)，REPLACE 则是表示替换表中所有字段。

2）实操案例

（1）查询表结构

hive> desc dept_partition;

（2）添加列

hive (default)> alter table dept_partition add columns(deptdesc string);

（3）查询表结构

hive> desc dept_partition;

（4）更新列

hive (default)> alter table dept_partition change column deptdesc desc int;

（5）查询表结构

hive> desc dept_partition;

（6）替换列

hive (default)> alter table dept_partition replace columns(deptno string, dname string, loc string);

（7）查询表结构

hive> desc dept_partition;

4.8 删除表

hive (default)> drop table dept_partition;