Hive总结1 基本操作

数据库的基本操作

创建数据库

```
create database [if not exists] myhive;
```

说明一下： Hive表存放位置模式是由hive-site.xml当中的一个属性指定的：

```
<property>
    <name>hive.metastore.warehouse.dir</name>
<value>/user/hive/warehouse</value>
</property>
```

创建数据库并指定HDFS的存储位置

```
create database myhive2 location '/myhive2';
```

删除数据库

```
drop database myhive2;
```

只能对空数据库使用该命令，如果数据库下面有数据表，那么就会报错。

强制删除数据库

```
drop database myhive cascade;
```

包含数据库下面的表一起删除，不要执行，十分危险

查看数据库

```
show databases;
```

查看详细信息

```
# 查看数据库基本信息
desc database myhive2;
# 查看数据库更多详细信息
desc database extended myhive2;
```

数据库切换

```
use myhive (数据库名);
```

修改数据库

数据库的元数据信息是不可更改的，**包括数据库的名称以及数据库所在的位置**,但我们可以使用**alter database** 命令来修改数据库的一些属性。

```
# 修改数据库的创建日期
alter  database  myhive2  set  dbproperties('createtime'='20880611');
```

Hive创建表的字段类型

基本数据类型

数据类型	长度	例子
TINYINT	1byte有符号整数	20
SMALINT	2byte有符号整数	20
INT	4byte有符号整数	20
BIGINT	8byte有符号整数	20
BOOLEAN	布尔类型	TRUE
FLOAT	单精度浮点数	3.14159
DOUBLE	双精度浮点数	3.14.59
STRING	字符序列。可以指定字符集	'hello'
TIMESTAMP	整数，浮点数或者字符串	12312;1231.1232;'2012-03-03'
BINARY	字节数组	

集合数据类型

数据类型	描述	字面语法示例
STRUCT	跟对象类似，可以通过点访问	struct('John','Doe)
MAP	MAP键值对	map('first','JOIN','last','Doe')
ARRAY	ARRAY相同数组集合	Array('John','Doe')

数据表的基本操作

创建基本数据表

```
CREATE TABLE tableName(
    字段名称 字段类型,
    字段名称 字段类型
    )
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY "\t"
    STORED AS TEXTFILE;
```

创建外部数据表

```
CREATE EXTERNAL TABLE tableName2(
    字段名称 字段类型,
    字段名称 字段类型
    )
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ','
    STORED AS TEXTFILE
    LOCATION '/user/hive/external/fz_external_table';
```

从本地文件系统向表中加载数据

```
load data local inpath '文件路径' into table 表名
```

加载文件并覆盖已有的数据

```
load data local inpath '文件路径' overwrite into table 表名
```

从HDFS文件系统向表中加载数据

```
load data inpath '文件路径' into table 表名;
```

内部表和外部表的区别

- 在删除内部表时：内部表删除将表的元数据和数据同时删除。
- 在删除外部表时：外部表的元数据被删除，数据本身不删除。

分区

创建分区表

企业分区常见的分区规则：按天进行分区（一天一个分区）

创建分区表语法

```
create table score(
    s_id string,
    c_id string,
    s_score int)
    partitioned by (month string)
    row format delimited
    fields terminated by '\t';
```

创建一个表带多个分区

```
create table score2 (
    s_id string,
    c_id string,
    s_score int)
    partitioned by (
    year string,
    month string,
    day string)
    row format delimited
    fields terminated by '\t';
```

加载数据到分区表中

```
load data local inpath '数据路径' into table score partition (month='201806');
```

加载数据到多分区表中去

```
load data local inpath '数据路径' into table score2
partition(year='2018',month='06',day='01');
```

查看分区

```
show partitions score;
```

添加一个分区

```
alter table score add partition(month='201805');
```

添加多个分区

```
alter table score add partition(month='201804')   partition(month = '201803');
```

注意：添加分区之后就可以在hdfs文件系统当中看到表下面多了一个文件夹

删除分区

```
alter table score drop partition(month = '201806');
```

注意：

分区字段绝对不能出现在数据库表已有的字段中!

分区的作用

将数据按区域划分开，查询时不用扫描无关的数据，加快查询速度。

分桶

是在已有的表结构之上新添加了特殊的结构。将数据按照指定的字段进行分成多个桶中去，说白了就是将数据按照字段进行划分，可以将数据按照字段划分到多个文件当中去

开启hive的桶表功能

```
set hive.enforce.bucketing=true;
```

设置桶reduce的个数

```
set mapreduce.job.reduces=3;
```

创建桶表

```
create table course (
    c_id string,
    c_name string,
    t_id string)
    clustered by(c_id) into 3 buckets
    row format delimited
    fields terminated by '\t';
```

注意

桶表的数据加载，只能通过insert overwrite，hdfs dfs -put文件或者通过load data无法加载。所以只能先创建普通表,并通过insert overwrite的方式将普通表的数据通过查询的方式加载到桶表当中去

创建普通表

```
create table course_common (
    c_id string,
    c_name string,
    t_id string)
    row format delimited
    fields terminated by '\t';
```

普通表中加载数据

```
load data local inpath '数据路径' into table course_common;
```

通过insert overwrite给桶表中加载数据

```
insert overwrite table course select * from course_common cluster by(c_id);
```

强调

分桶字段必须是表中的字段。

分桶逻辑
对分桶字段求哈希值,用哈希值与分桶的数量取余,余几,这个数据就放在哪个桶内。