

create database [IF NOT EXISTS] userdb; 创建数据库
user userdb; 使用userdb数据库

客户端的连接:

CLI的方式

hive(jdbc方式)

web方式

jdbc方式

hive --service hiveserver2 & //先后台运行hive
beeline -u jdbc:hive2:// -n hive -p a

web方式

hive --service hwi

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]
```

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、 LIKE 允许用户复制现有的表结构，但是不复制数据。

4、 ROW FORMAT 列格式

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 确定表的具体列的数据。

5、 STORED AS 存储格式

SEQUENCEFILE|TEXTFILE|RCFILE

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。

6、 CLUSTERED BY

对于每一个表（table）或者分区，Hive可以进一步组织成桶，也就是说桶是更为细粒度的数据范围划分。Hive也是针对某一列进行桶的组织。Hive采用对列值哈希，然后除以桶的个数求余的方式决定该条记录存放在哪个桶当中。

把表（或者分区）组织成桶（Bucket）有两个理由：

（1）获得更高的查询处理效率。桶为表加上了额外的结构，Hive 在处理有些查询时能利用这个结构。具体而言，连接两个在（包含连接列的）相同列上划分了桶的表，可以使用 Map 端连接（Map-side join）高效的实现。比如JOIN操作。对于JOIN操作两个表有一个相同的列，如果在这两个表都进行了桶操作。那么将保存相同列值的桶进行JOIN操作就可以，可以大大减少JOIN的数据量。

（2）使取样（sampling）更高效。在处理大规模数据集时，在开发和修改查询的阶段，如果能在数据集的一小部分数据上试运行查询，会带来很多方便。

原文链接：<https://blog.csdn.net/l1212xiao/article/details/80432759>

一些建表例子：

```
create table student(  
stuno int comment 'xuehao',
```

```
name string comment 'student name',
course array<string>,
score map<string,int>,
address struct<province:string,city:string,zip:string>
)
row format delimited
fields terminated by '\t'
collection items terminated by ','
map keys terminated by ':'
lines terminated by '\n'
stored as textfile
location '/data/';
```

```
create table checking(
id int ,
term string,
uid int,
cid int,
checkDate date,
checkTime string,
result map<string,int>
)
partitioned by (year int,month int)
clustered by (cid) sorted by (id) into 3 buckets
row format delimited fields terminated by '\t'
collection items terminated by ','
map keys terminated by ':'
lines terminated by '\n';
```

插入数据：

```
insert into table checking partition(yeat,month) select id
,term,uid,cid,checkDate,checkTime,result,year(checkDate),month(checkDate) from
checking_temp;
```

```
load data local inpath " into table dd partition(age=24);
```

删除数据：

```
insert overwrite table dw_200_rst_advert_park_idea_place_stat_day PARTITION
(dt='2017-12-20',game_id = 'id') select * from
dw_200_rst_advert_park_idea_place_stat_day where id IS NOT NULL ;
```

-- 按分区删除:

```
ALTER TABLE test1 DROP PARTITION (dt='2016-04-29');
```

```
-- 清空表
```

```
truncate table employee;
```

<https://blog.csdn.net/zimou5581/article/details/82383906>

<https://www.cnblogs.com/linn/p/6196293.html>

添加分区：

```
alter table table_name add partition(age=8);
```