

课堂实验

一、基本信息

文档编号		文档版本	1.0
实验名称	Hyperbase 基本用法		
所属课程	第 14 讲 分布式 NewSQL 数据库 Hyperbase	认证等级	数据工程师（初级）
授课形式	上机实验	实验批次	第 10 次 / 共 11 次

二、实验目的

- 掌握 Hyperbase 增删改查等基本操作。
- 掌握 SQLBulkload 数据导入的基本方法。

三、实验准备

- 下载并安装 TDH Client。
- 下载并安装 Waterdrop, SQL 语句在 Waterdrop 中运行。
- 实验目录与命名规划
 - (1) 本地目录
工作目录: /mnt/disk1/{student_name}
 - (2) HDFS 目录
工作目录: /tmp/{student_name}
Inceptor 外表目录: /tmp/{student_name}/hyper_data
 - (3) Inceptor 数据库名: {db_student_name}
 - (4) {student_name} 为变量, 代表学员姓名全拼

四、实验内容

1、进入 Hyperbase 命令行

- 任务: 启动 TDH Client, 用 hbase shell 命令进入 Hyperbase 命令行。
- 步骤

Linux:

```
// 执行 init.sh 脚本, 启动 TDH Client
1. source {TDH_Client_install_dir}/init.sh
// 将 Hadoop 当前用户设置为 hbase, 进行访问授权
2. export HADOOP_USER_NAME=hbase
// 进入 Hyperbase 命令行
3. hbase shell
```

2、创建表, 并插入数据

- 任务: 创建一张表, 并使用 put 命令向表中插入数据。
- 步骤

HBase Shell:

```
// 创建表
1. create '{table_student_name}', 'cf'
// 列出表
2. list
// 插入数据
3. put '{table_student_name}', 'row1', 'cf:a', 'value1'
4. put '{table_student_name}', 'row2', 'cf:b', 'value2'
5. put '{table_student_name}', 'row3', 'cf:c', 'value3'
```

3、读取数据

- 任务: 使用 get、scan 命令读取数据, 使用 count 命令统计行数。
- 步骤

HBase Shell:

```
// 读取 Cell
1. get '{table_student_name}', 'row1'
// 读取前 10 行
2. scan '{table_student_name}', {LIMIT=>10}
// 统计行数
3. count '{table_student_name}'
```

4、修改表属性

- 任务: 下线表, 更改表属性, 上线表, 查看更改结果。
- 步骤

HBase Shell:

```
// 将表下线
1. disable '{table_student_name}'
```

```
// 更改表属性
2. alter '{table_student_name}',{NAME=>'cf',COMPRESSION => 'snappy',
   DATA_BLOCK_ENCODING => 'PREFIX'}
// 将表上线
3. enable '{table_student_name}'
// 查看表属性
4. desc '{table_student_name}'
```

5、分裂和压缩表

- 任务：将基于衍生流的统计结果写入 Inceptor 结果表，该操作启动流任务，触发流处理过程。
- 步骤

HBase Shell:

```
// 插入数据
1. put '{table_student_name}','row4','cf:a','value4'
2. put '{table_student_name}','row5','cf:b','value5'
3. put '{table_student_name}','row6','cf:c','value6'
// 将数据溢写磁盘
4. flush '{table_student_name}'
// 分裂表
5. split '{table_student_name}','row3'
// 压缩表
6. compact '{table_student_name}'
7. major_compact '{table_student_name}'
```

6、导出表属性到本地文件

- 任务：利用 describeInJson 命令将表属性导出到本地 Json 文件，并查看表属性。
- 步骤

HBase Shell:

```
// 将表属性导出到本地 Json 文件
1. describeInJson '{table_student_name}','true','/mnt/disk1/{student_name}/hbase.json'
2. exit
```

Linux:

```
// 查看表属性文件
1. cat /mnt/disk1/{student_name}/hbase.json
```

7、查看 META 表

- 任务：查看 META 表，理解 Hyperbase 元数据。

- 步骤

HBase Shell:

1. scan 'hbase:meta'

8、删除表

- 任务：先将表下线，再删除。

- 步骤

HBase Shell:

1. disable '{table_student_name}'
2. drop '{table_student_name}'

9、SQLBulkload 数据导入

- 任务：创建 Hyperdrive 表，使用 SQLBulkload 批量导入数据。

- 步骤

(1) 在 Inceptor 中创建 Hyperbase 表

SQL:

1. use {db_student_name};
2. drop table if exists hyper_table;
3. create table hyper_table(
 rowkey string,
 num int,
 country int,
 rd string)
 stored as hyperdrive
 tblproperties
 ('hyperdrive.table.splitkey'='"110000000","120000000","130000000","140000000","150000000","160000000","170000000","180000000","190000000"');

(2) 将实验数据上传到 HDFS

Linux:

1. hadoop fs -mkdir -p /tmp/{student_name}/hyper_data
2. hadoop fs -put {data.csv_file_path} /tmp/{student_name}/hyper_data

(3) 基于 HDFS 实验数据，在 Inceptor 中创建外表

SQL:

1. drop table if exists hyper_external_table;
2. create external table hyper_external_table(
 rowkey string,
 num int,
 country int,

```
rd int)
row format delimited fields terminated by ','
location '/tmp/{student_name}/hyper_data';
```

(4) 将 Inceptor 外表中的数据用 SQLBulkLoad 导入 Hyperbase 表

SQL:

1. insert overwrite table hyper_table select /*+USE_BULKLOAD*/ rowkey,num,country,rd from hyper_external_table order by rowkey;
2. select * from hyper_table limit 10;

10、基于全局索引进行查询

- 任务：先为 Hyperbase 表创建全局索引，再基于索引进行查询。
- 步骤

(1) 在 Inceptor 中创建全局索引

SQL:

1. use {db_student_name};
2. create global index index_num on hyper_table(num);

(2) 在 HBase Shell 中生成索引数据

HBase Shell:

- ```
// 对于非默认数据库下创建的 Hyperdrive 表，创建全局索引时，表前要加数据库前缀
1. rebuild_global_index '{db_student_name}:hyper_table', 'index_num'
```

(3) 在 Inceptor 中基于索引进行查询

SQL:

- ```
// Inceptor 默认运行在 Cluster Mode 下，查询时不基于索引
1. select * from hyper_table where num=503;
// 在 Cluster Mode 下，只有明确声明，Inceptor 才基于索引进行查询
2. select /*+USE_INDEX(e USING index_num)*/ * from hyper_table e where num =503;
// 切换为 Local Mode，Inceptor 自动匹配合适的索引进行查询
3. set ngmr.exec.mode=local;
4. set hyperbase.integer.transform=true;
5. select * from hyper_table where num =503;
// 将 Inceptor 切换为 Cluster Mode
6. set ngmr.exec.mode=cluster;
```