用户手册

主要描述ThssDB基本功能及进阶功能的使用，以及结果的展示

代码配置

配置`src/main/java/cn/edu/thssdb/Config.java`文件

```
public static final Boolean CLOSE_LOG = true; // 是否关闭日志
public static final Boolean USE_GZIP = false; // 是否使用压缩
public static final String ISOLATION_LEVEL = "READ_COMMITTED";
//  public static final String ISOLATION_LEVEL = "SERIALIZATION";
```

- 切换隔离级别：将ISOLATION_LEVEL设置为"READ_COMMITTED"，即为"READ_COMMITTED"隔离级别，设置为"SERIALIZATION"即为"SERIALIZATION"隔离级别
- 是否开启日志功能：CLOSE_LOG为true则关闭日志功能，为false则开启日志
  - 提交的压缩包里read_committed.jar为关闭日志的版本
  - 提交的压缩包里**read_committed_log.jar**为开启日志的版本
- 是否开启对数据文件的压缩：USE_GZIP为true则开启压缩功能，为false则关闭压缩

在进行一切操作之前，需要先启动ThssDB服务端，再启动Client客户端，输入以下命令进行连接：

```
connect root root;
```

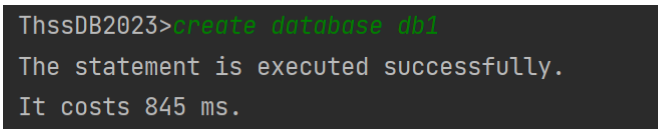
1 元数据管理模块

1.1 基本功能

1.1.1 实现数据库的创建、删除、切换

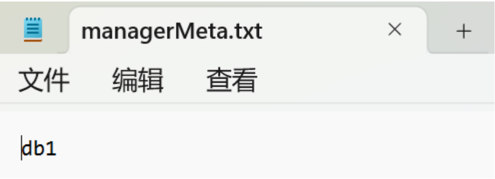
数据库的创建

```
create database dbName;
```

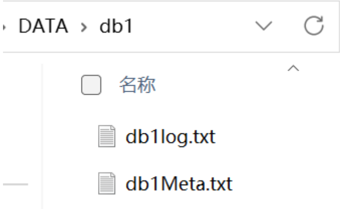


```
ThssDB2023>create database db1
The statement is executed successfully.
It costs 845 ms.
```

运行后，managerMeta.txt中出现db1数据库名称，DATA文件夹中出现新的db1文件夹，里面有db1的meta文件与log文件

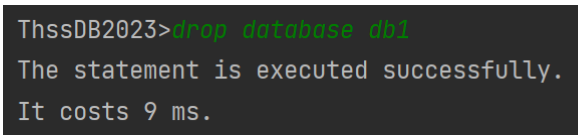


db1



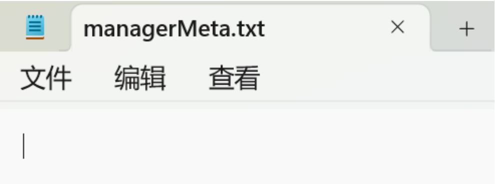
数据库的删除

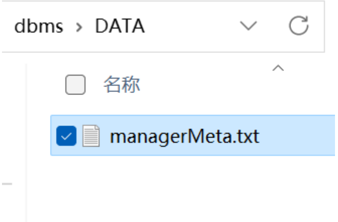
```
drop database dbName;
```



```
ThssDB2023>drop database db1
The statement is executed successfully.
It costs 9 ms.
```

运行后，managerMeta.txt中的db1没有了，DATA中的db1文件夹被整个删除

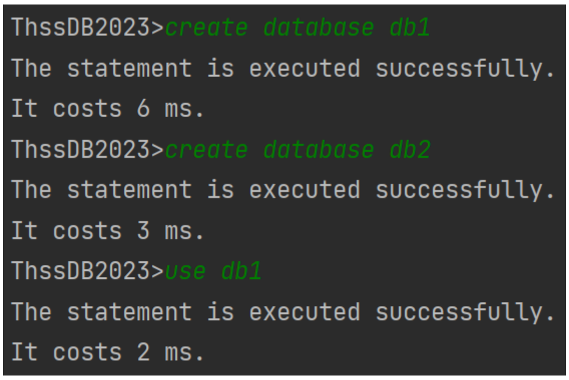




数据库的切换

注意一定要显式指明所使用的数据库

```
use dbName;
```



```
ThssDB2023>create database db1
The statement is executed successfully.
It costs 6 ms.
ThssDB2023>create database db2
The statement is executed successfully.
It costs 3 ms.
ThssDB2023>use db1
The statement is executed successfully.
It costs 2 ms.
```

运行后，即可在db1数据库中进行之后的操作

1.1.2 实现表的创建、删除

表的创建

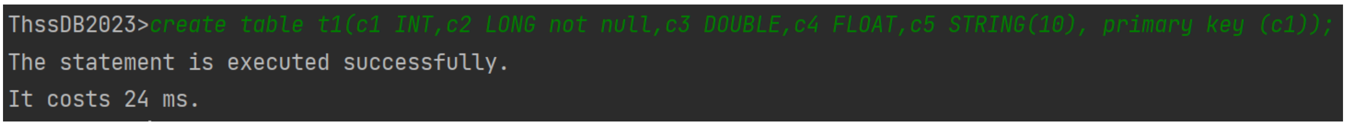
实现了“NOT NULL”和“PRIMARY KEY”两个关键字，Type为Int，Long，Float，Double，String（必须给出长度）之一。

格式：

```
CREATE TABLE tableName(
    attrName1 Type1,
    attrName2 Type2,
    attrNameN TypeN NOT NULL,
    …,
    PRIMARY KEY(attrName1, attrName2, ...)
);
```

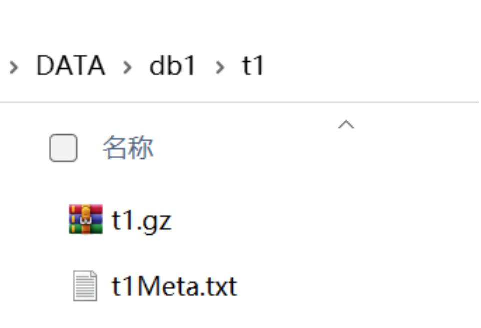
测试代码：

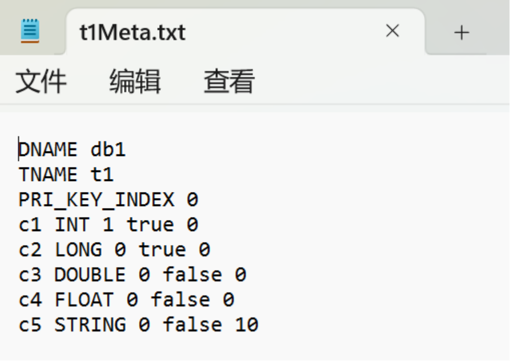
```
create table t1(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10), primary key
(c1));
```



```
ThssDB2023>create table t1(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10), primary key (c1));
The statement is executed successfully.
It costs 24 ms.
```

运行后db1文件夹下出现t1文件夹，t1Meta.txt记录了t1表的信息：



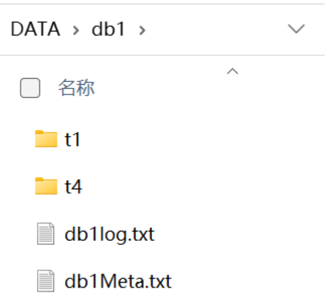


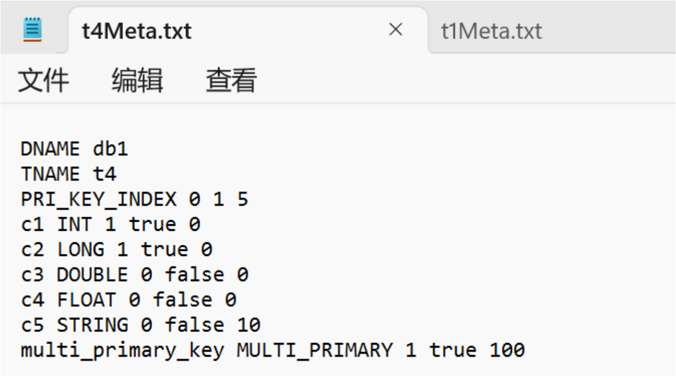
```
DNAME db1
TNAME t1
PRI_KEY_INDEX 0
c1 INT 1 true 0
c2 LONG 0 true 0
c3 DOUBLE 0 false 0
c4 FLOAT 0 false 0
c5 STRING 0 false 10
```

我们也支持多列主键：

```
create table t4(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key
(c1,c2));
```

运行后db1文件夹下出现t4文件夹，t4Meta.txt记录了t4表的信息，并标明了c1与c2都是主键





```
DNAME db1
TNAME t4
PRI_KEY_INDEX 0 1 5
c1 INT 1 true 0
c2 LONG 1 true 0
c3 DOUBLE 0 false 0
c4 FLOAT 0 false 0
c5 STRING 0 false 10
multi_primary_key MULTI_PRIMARY 1 true 100
```

表的删除

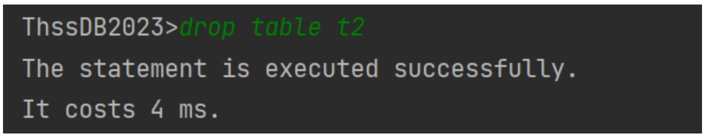
```
DROP TABLE tableName;
```

先再创建一个表t2：

```
create table t2(c1 INT,c2 LONG not null,c3 INT, c4 STRING(10), primary key (c1));
```

然后运行以下命令删除表t2：

```
drop table t2;
```



```
ThssDB2023>drop table t2
The statement is executed successfully.
It costs 4 ms.
```

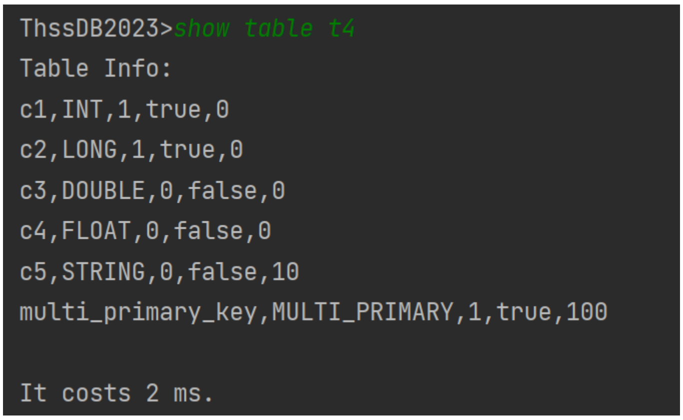
表的查找

```
SHOW TABLE tableName;
```

例如：

```
show table t4;
```

会显示表中每列的信息：

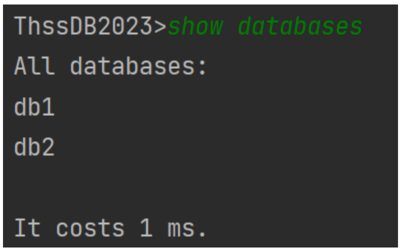


```
ThssDB2023>show table t4
Table Info:
c1,INT,1,true,0
c2,LONG,1,true,0
c3,DOUBLE,0,false,0
c4,FLOAT,0,false,0
c5,STRING,0,false,10
multi_primary_key,MULTI_PRIMARY,1,true,100

It costs 2 ms.
```

显示全部数据库

```
show databases
```



```
ThssDB2023>show databases
All databases:
db1
db2

It costs 1 ms.
```

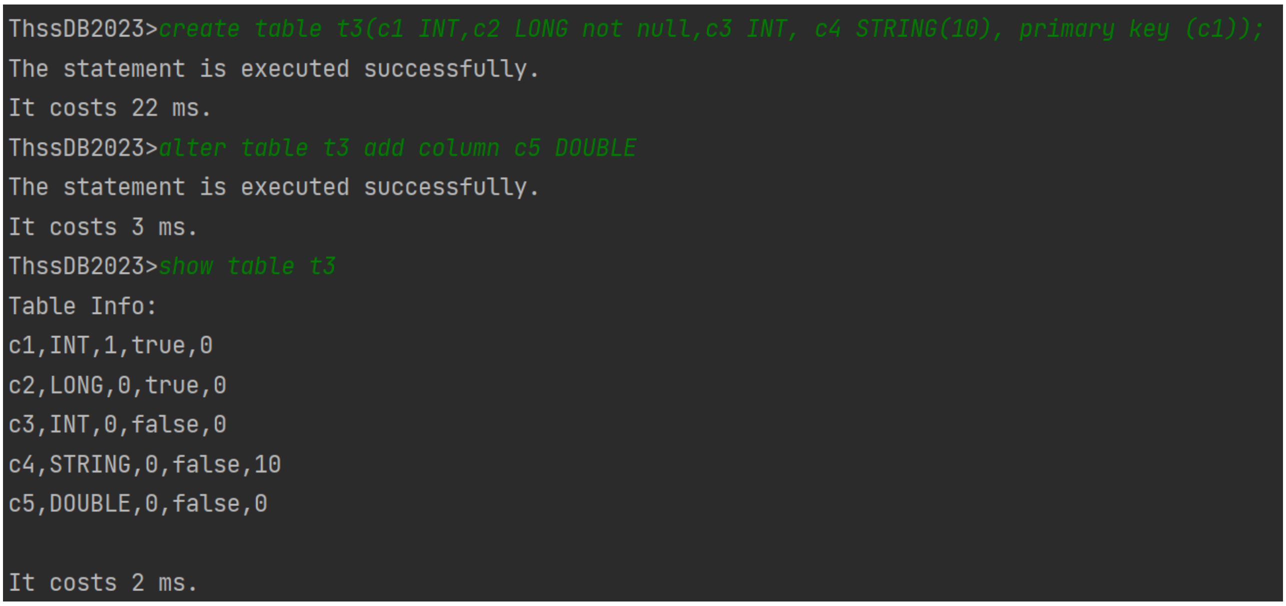
1.2 进阶功能

1.2.1 实现表结构或表约束的修改

```
create table t3(c1 INT,c2 LONG not null,c3 INT, c4 STRING(10), primary key (c1));
```

- **添加、删除、修改列**

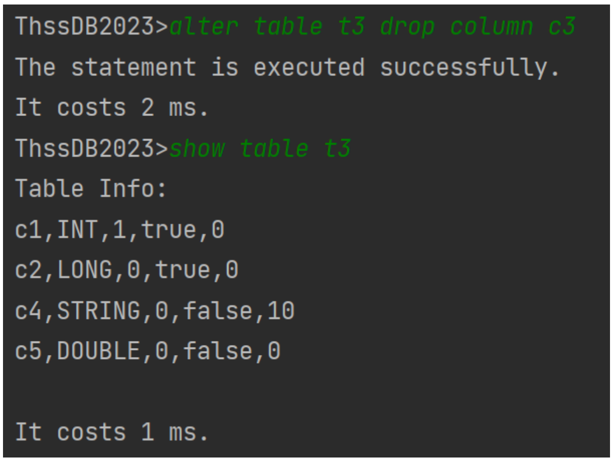
```
# 添加列
alter table t3 add column c5 DOUBLE
show table t3
```



```
ThssDB2023>create table t3(c1 INT,c2 LONG not null,c3 INT, c4 STRING(10), primary key (c1));
The statement is executed successfully.
It costs 22 ms.
ThssDB2023>alter table t3 add column c5 DOUBLE
The statement is executed successfully.
It costs 3 ms.
ThssDB2023>show table t3
Table Info:
c1,INT,1,true,0
c2,LONG,0,true,0
c3,INT,0,false,0
c4,STRING,0,false,10
c5,DOUBLE,0,false,0

It costs 2 ms.
```

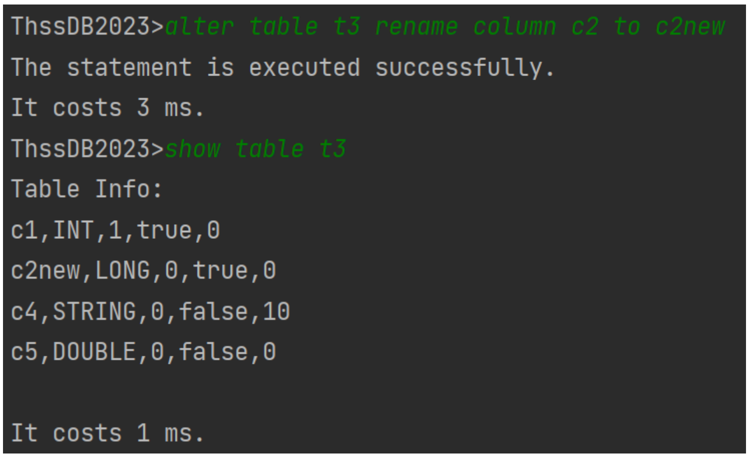
```
# 删除列
alter table t3 drop column c3
show table t3
```



```
ThssDB2023>alter table t3 drop column c3
The statement is executed successfully.
It costs 2 ms.
ThssDB2023>show table t3
Table Info:
c1,INT,1,true,0
c2,LONG,0,true,0
c4,STRING,0,false,10
c5,DOUBLE,0,false,0

It costs 1 ms.
```

```
# 重命名列
alter table t3 rename column c2 to c2new
show table t3
```

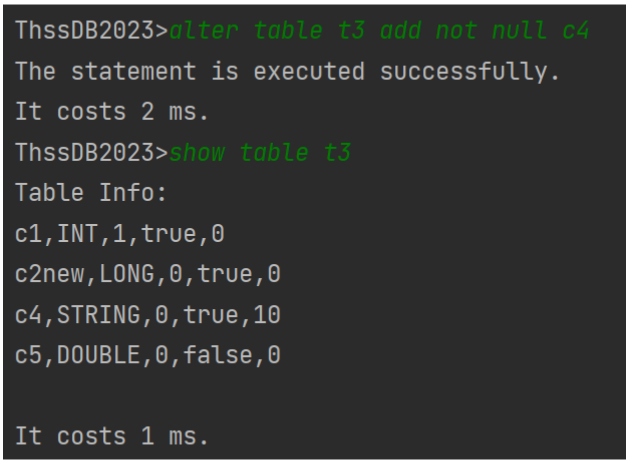


```
ThssDB2023>alter table t3 rename column c2 to c2new
The statement is executed successfully.
It costs 3 ms.
ThssDB2023>show table t3
Table Info:
c1,INT,1,true,0
c2new,LONG,0,true,0
c4,STRING,0,false,10
c5,DOUBLE,0,false,0

It costs 1 ms.
```

- **添加、删除表上的约束**

```
# 添加约束
alter table t3 add not null c4
show table t3
```

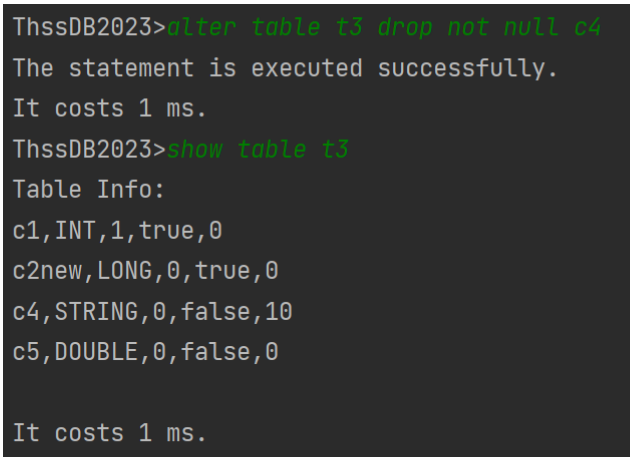


```
ThssDB2023>alter table t3 add not null c4
The statement is executed successfully.
It costs 2 ms.
ThssDB2023>show table t3
Table Info:
c1,INT,1,true,0
c2new,LONG,0,true,0
c4,STRING,0,true,10
c5,DOUBLE,0,false,0

It costs 1 ms.
```

(注：每一行倒数第二个bool值代表这一列是否not null)

```
# 删除约束
alter table t3 drop not null c4
show table t3
```



```
ThssDB2023>alter table t3 drop not null c4
The statement is executed successfully.
It costs 1 ms.
ThssDB2023>show table t3
Table Info:
c1,INT,1,true,0
c2new,LONG,0,true,0
c4,STRING,0,false,10
c5,DOUBLE,0,false,0

It costs 1 ms.
```

2 存储模块

2.1 基本功能

2.1.1 数据写入

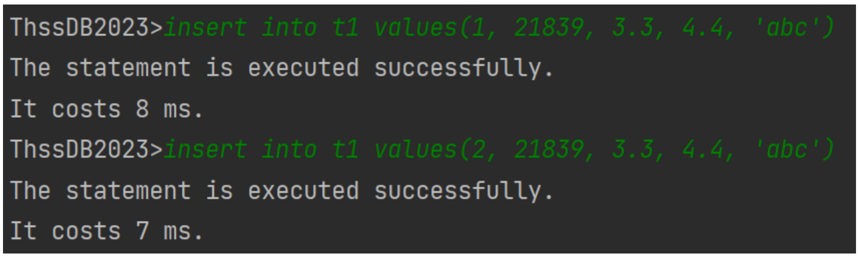
字符串需要用单引号包围，无需考虑负数

格式：

```
INSERT INTO [tableName(attrName1, attrName2,…, attrNameN)] VALUES (attrValue1,
attrValue2,…, attrValueN);
```

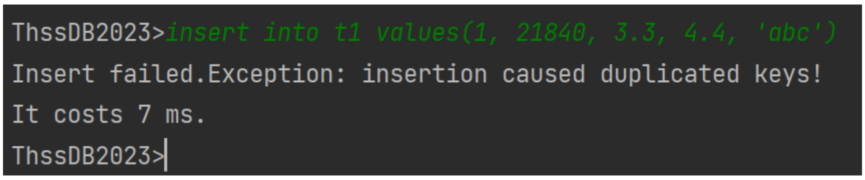
测试代码：

```
create database db3;
use db3;
create table t1(c1 INT,c2 LONG not null,c3 FLOAT, c4 DOUBLE, c5 STRING(10), primary key
(c1));
insert into t1 values(1, 21839, 3.3, 4.4, 'abc')
insert into t1 values(2, 21839, 3.3, 4.4, 'abc')
```



```
ThssDB2023>insert into t1 values(1, 21839, 3.3, 4.4, 'abc')
The statement is executed successfully.
It costs 8 ms.
ThssDB2023>insert into t1 values(2, 21839, 3.3, 4.4, 'abc')
The statement is executed successfully.
It costs 7 ms.
```

插入重复主键会有提示：



```
ThssDB2023>insert into t1 values(1, 21840, 3.3, 4.4, 'abc')
Insert failed.Exception: insertion caused duplicated keys!
It costs 7 ms.
ThssDB2023>
```

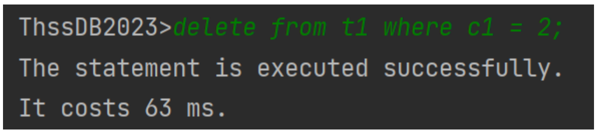
2.1.2 数据删除

格式：

```
DELETE FROM tableName WHERE attrName = attValue;
```

测试代码：

```
delete from t1 where c1 = 2;
```



```
ThssDB2023>delete from t1 where c1 = 2;
The statement is executed successfully.
It costs 63 ms.
```

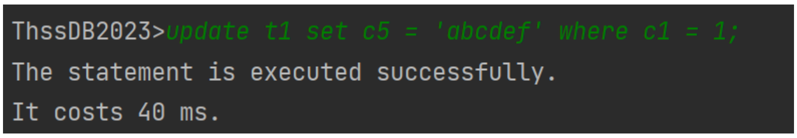
2.1.3 数据更新

格式：

```
UPDATE tableName SET attrName=attrValue WHERE attrName=attrValue;
```

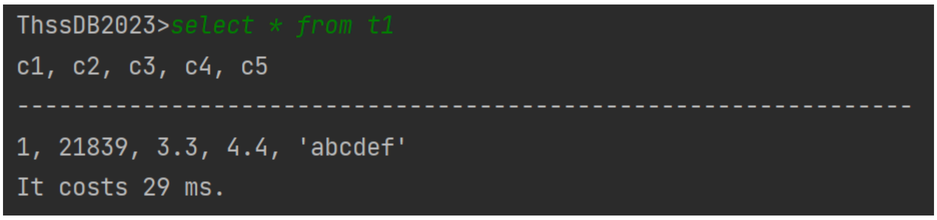
测试代码：

```
update t1 set c5 = 'abcdef' where c1 = 1;
```



```
ThssDB2023>update t1 set c5 = 'abcdef' where c1 = 1;
The statement is executed successfully.
It costs 40 ms.
```

验证：



```
ThssDB2023>select * from t1
c1, c2, c3, c4, c5
----------------------------------------------------------------
1, 21839, 3.3, 4.4, 'abcdef'
It costs 29 ms.
```

2.2 进阶功能

可以开启数据文件的压缩，在配置文件`src/main/java/cn/edu/thssdb/Config.java`里指定

```
public static final Boolean USE_GZIP = true; // 是否使用压缩
```

3 查询模块

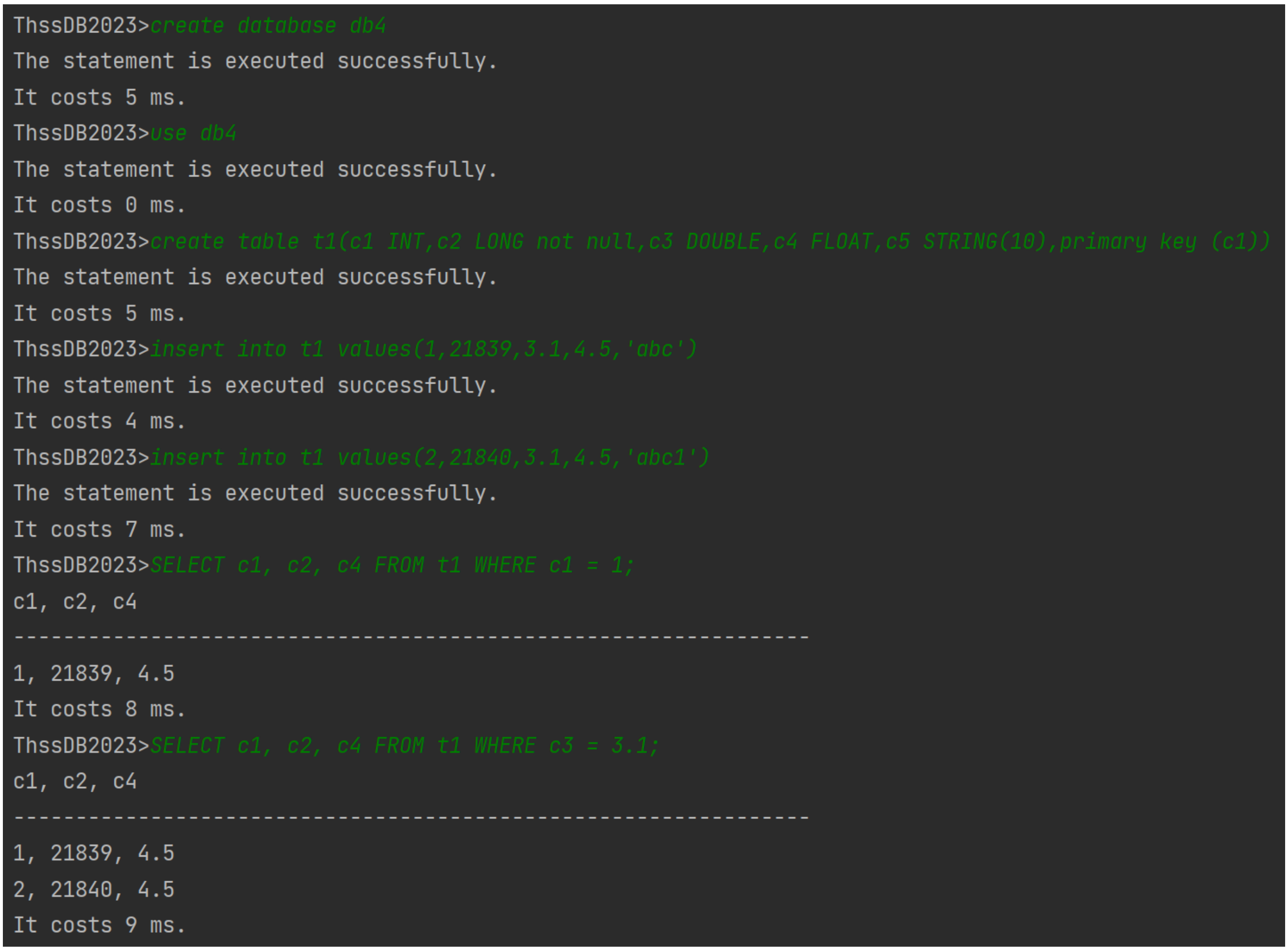
3.1 基本功能

支持下列SQL语句：

```
# 1
SELECT attrName1, attrName2, … attrNameN FROM tableName [WHERE attrName1 = attrValue];
# 2
SELECT tableName1.AttrName1, tableName1.AttrName2…, tableName2.AttrName1,
tableName2.AttrName2,… FROM tableName1 JOIN tableName2 ON
tableName1.attrName1=tableName2.attrName2 [WHERE  tableName1.attrName1 = attrValue];
```

测试代码1：

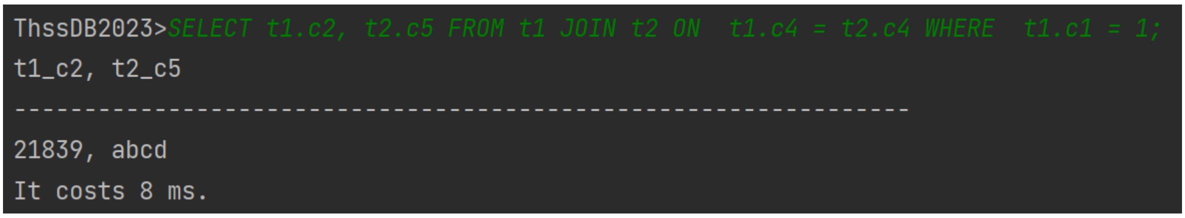
```
create database db4
use db4
create table t1(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1))
insert into t1 values(1,21839,3.1,4.5,'abc')
insert into t1 values(2,21840,3.1,4.5,'abc1')
SELECT c1, c2, c4 FROM t1 WHERE c1 = 1;
SELECT c1, c2, c4 FROM t1 WHERE c3 = 3.1;
```



```
ThssDB2023>create database db4
The statement is executed successfully.
It costs 5 ms.
ThssDB2023>use db4
The statement is executed successfully.
It costs 0 ms.
ThssDB2023>create table t1(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1))
The statement is executed successfully.
It costs 5 ms.
ThssDB2023>insert into t1 values(1,21839,3.1,4.5,'abc')
The statement is executed successfully.
It costs 4 ms.
ThssDB2023>insert into t1 values(2,21840,3.1,4.5,'abc1')
The statement is executed successfully.
It costs 7 ms.
ThssDB2023>SELECT c1, c2, c4 FROM t1 WHERE c1 = 1;
c1, c2, c4
----------------------------------------------------------------
1, 21839, 4.5
It costs 8 ms.
ThssDB2023>SELECT c1, c2, c4 FROM t1 WHERE c3 = 3.1;
c1, c2, c4
----------------------------------------------------------------
1, 21839, 4.5
2, 21840, 4.5
It costs 9 ms.
```

测试代码2：

```
create table t2(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1))
insert into t2 values(2,21839,3.1,4.5,'abcd')
insert into t2 values(3,21840,3.1,4.6,'abcd')
SELECT t1.c2, t2.c5 FROM t1 JOIN t2 ON  t1.c4 = t2.c4 WHERE  t1.c1 = 1;
```



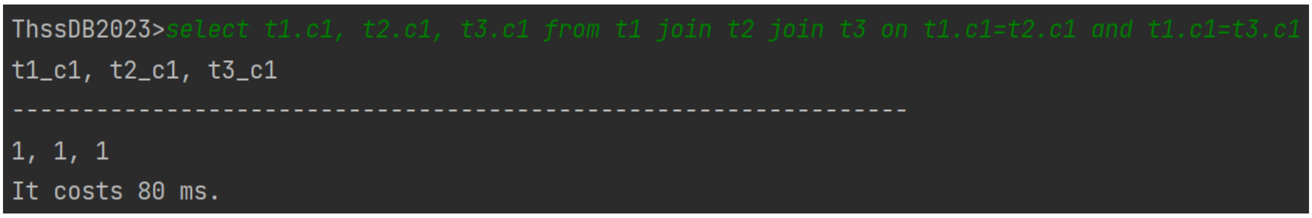
```
ThssDB2023>SELECT t1.c2, t2.c5 FROM t1 JOIN t2 ON  t1.c4 = t2.c4 WHERE  t1.c1 = 1;
t1_c2, t2_c5
----------------------------------------------------------------
21839, abcd
It costs 8 ms.
```

3.2 进阶功能

3.2.1 支持三张表以上的join

测试代码：

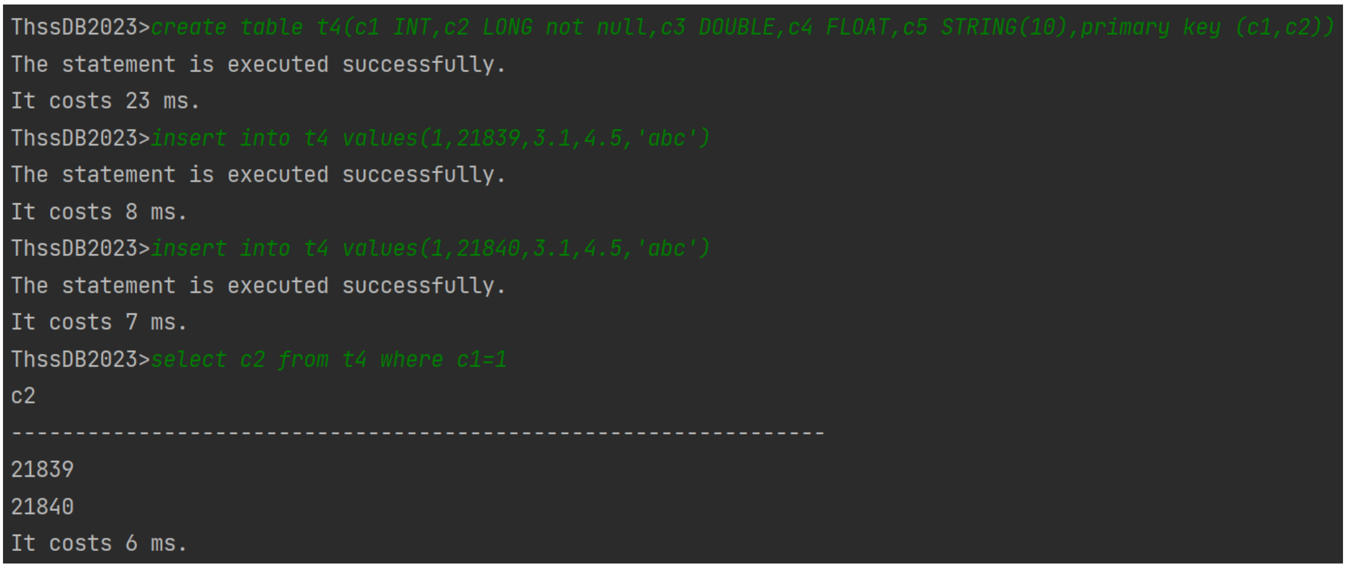
```
create database db5
use db5
create table t1(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1))
create table t2(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1))
create table t3(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1))
insert into t1 values(1,21839,3.1,4.5,'abc')
insert into t2 values(1,21839,3.1,4.5,'abc')
insert into t3 values(1,21839,3.1,4.5,'abc')
# 多表join
select t1.c1, t2.c1, t3.c1 from t1 join t2 join t3 on t1.c1=t2.c1 and t1.c1=t3.c1
```



```
ThssDB2023>select t1.c1, t2.c1, t3.c1 from t1 join t2 join t3 on t1.c1=t2.c1 and t1.c1=t3.c1
t1_c1, t2_c1, t3_c1
----------------------------------------------------------------
1, 1, 1
It costs 80 ms.
```

3.2.2 支持多列主键

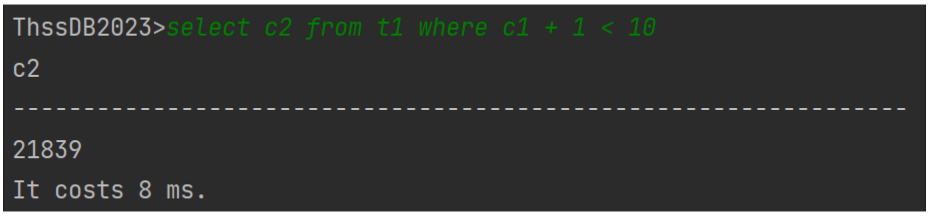
```
create table t4(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key
(c1,c2))
insert into t4 values(1,21839,3.1,4.5,'abc')
insert into t4 values(1,21840,3.1,4.5,'abc')
select c2 from t4 where c1=1
```



```
ThssDB2023>create table t4(c1 INT,c2 LONG not null,c3 DOUBLE,c4 FLOAT,c5 STRING(10),primary key (c1,c2))
The statement is executed successfully.
It costs 23 ms.
ThssDB2023>insert into t4 values(1,21839,3.1,4.5,'abc')
The statement is executed successfully.
It costs 8 ms.
ThssDB2023>insert into t4 values(1,21840,3.1,4.5,'abc')
The statement is executed successfully.
It costs 7 ms.
ThssDB2023>select c2 from t4 where c1=1
c2
----------------------------------------------------------------
21839
21840
It costs 6 ms.
```

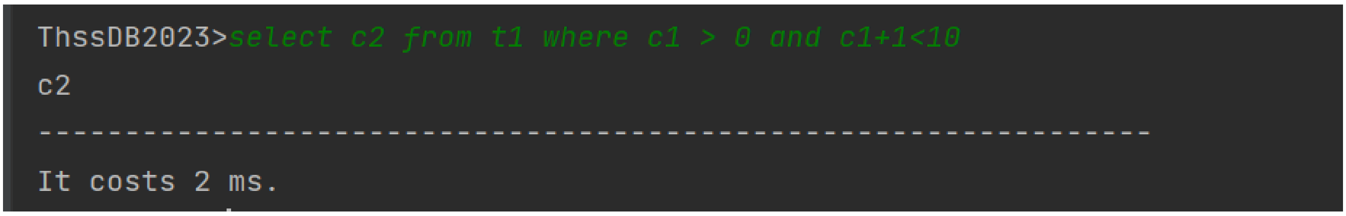
3.2.3 where支持表达式和多个条件AND/OR

```
select c2 from t1 where c1 + 1 < 10
```



```
ThssDB2023>select c2 from t1 where c1 + 1 < 10
c2
-----------------------------------------------------------------
21839
It costs 8 ms.
```

```
select c2 from t1 where c1 > 0 and c1+1<10
```

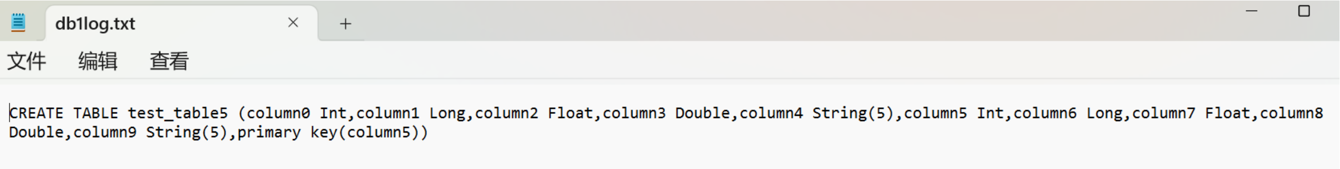


```
ThssDB2023>select c2 from t1 where c1 > 0 and c1+1<10
c2
-----------------------------------------------------------------
It costs 2 ms.
```

4 WAL演示

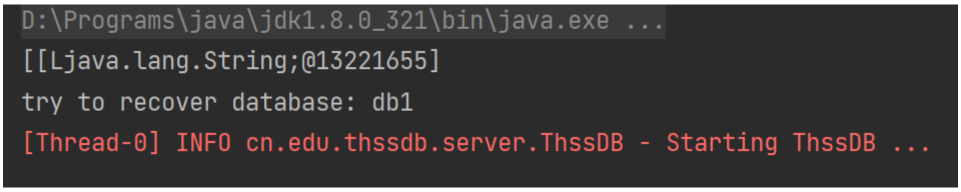
如果运行过程中，服务器关机

看到此时的日志：

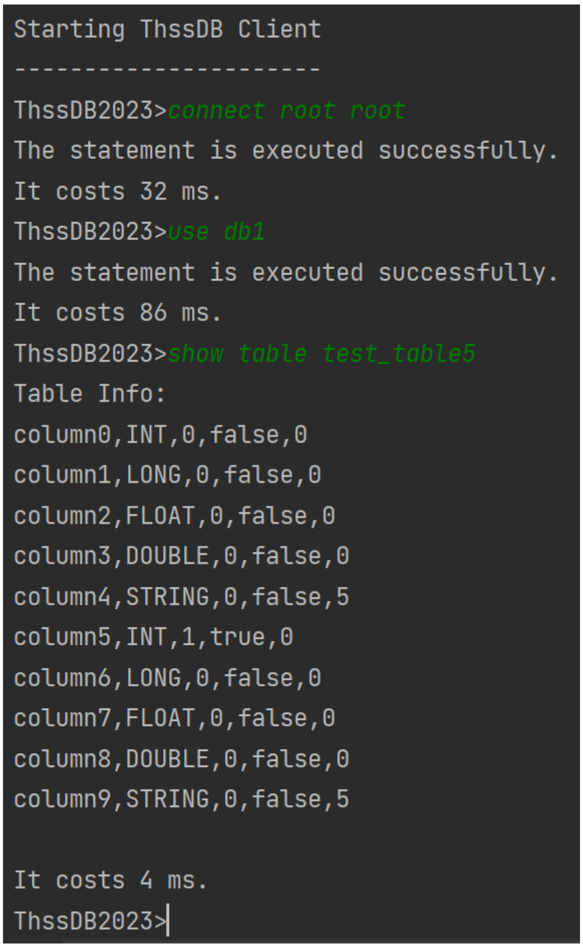


```
CREATE TABLE test_table5 (column0 Int,column1 Long,column2 Float,column3 Double,column4 String(5),column5 Int,column6 Long,column7 Float,column8
Double,column9 String(5),primary key(column5))
```

重启服务器后，重新连上客户端，发现之前操作的数据还在，说明对于已经commit未序列化存储的修改，服务器通过log recover恢复了



```
D:\Programs\java\jdk1.8.0_321\bin\java.exe ...
[[Ljava.lang.String;@13221655]
try to recover database: db1
[Thread-0] INFO cn.edu.thssdb.server.ThssDB - Starting ThssDB ...
```



```
Starting ThssDB Client
----------------------
ThssDB2023>connect root root
The statement is executed successfully.
It costs 32 ms.
ThssDB2023>use db1
The statement is executed successfully.
It costs 86 ms.
ThssDB2023>show table test_table5
Table Info:
column0,INT,0,false,0
column1,LONG,0,false,0
column2,FLOAT,0,false,0
column3,DOUBLE,0,false,0
column4,STRING,0,false,5
column5,INT,1,true,0
column6,LONG,0,false,0
column7,FLOAT,0,false,0
column8,DOUBLE,0,false,0
column9,STRING,0,false,5

It costs 4 ms.
ThssDB2023>
```

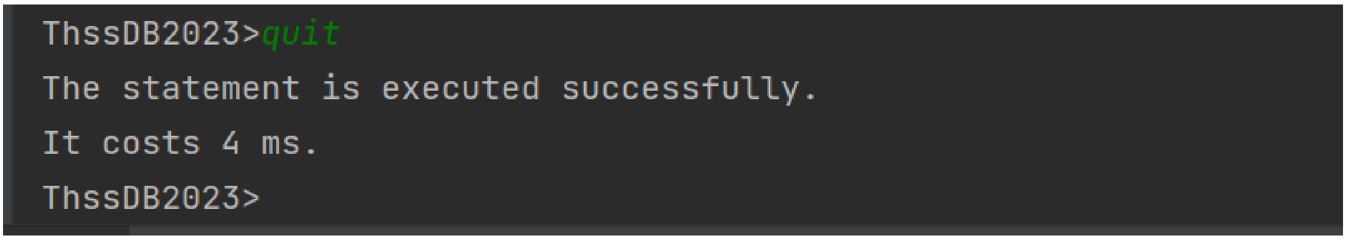
您也可以看我们录制的测试WAL的视频

https://cloud.tsinghua.edu.cn/f/2ed601cb17094b659e19/

退出

输入`quit`结束客户端连接。

```
quit
```



```
ThssDB2023>quit
The statement is executed successfully.
It costs 4 ms.
ThssDB2023>
```