

# ThssDB 系统实现实验报告

谭恺 2019011317

王森 2019013249

翟浩清 2019011308

张兴龙 2019013258

## 一、查询模块：

Select 语句主要将查询语句分为，join，filter，select，distinct 四个部分进行。

1. join 部分：首先将所有的表进行全连接，若查询中存在 join 语句则先会将 join 语句连接的两表进行连接，并根据 on 条件进行初步筛选。最终生成 QueryTable 对象。QueryTable 对象主要成员为 Row 与 Column，其中 row 存储所有的数据元组，Column 存储每一列的元数据（仅包含数据名称与类型）
2. filter 部分：全连接完成后，对 Row 列表中每一个元素根据 where 条件进行筛选，递归处理 where 条件子句中的 and，or 语句，并对表达式进行递归的计算，筛出符合条件的行。
3. select 部分。Filter 完成后，对数据列进行筛选。分为三种情况，\*，A.\*，a.对于\*则相当于不进行筛选，A.\*则将表 A 对应的所有列筛选出来（在创建 QueryTable 是，重命名所有列为 A.a 形式）。如果筛选的列名没有携带表名，则对该列名进行扩展，（在列名前加一个"."），即使存在两个不同的表都拥有该列的情况，最终的返回结果也只会是第一次匹配到的列。
4. distinct：在 select 完成之后，重新对 row 进行检查，并剔除掉重复元素。
5. 在完成以上四个步骤后，构造 QueryResult 对象进行返回。

### 创建表

由 column\_def()获取所有列的定义。每一列由 column\_name()获取列的名称。对于列的类型，首先获取 type\_name()，若解析后得到的字符串包含"STRING"和"("，则列的类型为"STRING"，并设置最大长度为 "(" 后的整数，否则设置列的类型为字符串对应的值。若某个列的定义中包含"PRIMARYKEY"或"NOTNULL"关键字，则对该列进行相应的设置。由 table\_constraint()获取表的主键约束，将包含的列设置为主键。

### 表插入：

实现：通过解析树构造需插入的数据项(Row)，调用 Table 的插入函数插入数据。

### 表更新：

实现：通过解析树获取 update 条件，再遍历 Table 所有元组(Row)，若满足条件，则调用 Table.update 对数据项进行更新。这里需要注意的是数据的引用问题，直接对原数据项进行修改可能造成意料之外的结果(如修改主键)；此外，原 Table 的 update 函数存在问题，导致所有插入的数据均产生主键冲突，这里修改为先删除旧数据项，再检查新数据项是否产生主键冲突，若冲突则将旧数据项恢复。

### 表删除：

实现：通过解析树获取 delete 条件，再遍历 Table 所有元组(Row)，若满足条件，则调用 Table.delete 删除数据项。

## 展示数据库信息

1. SHOW DATABASES 展示所有数据库的名称。实现方式为获取 manager 中 HashMap 类型的变量 databases，键值即为数据库的名称。
2. SHOW DATABASE databaseName 展示某个数据库中所有表的名称。实现方式为获取 manager 中相应名称的 database，然后获取该 database 中 HashMap 类型的变量 tableMap，键值即为表的名称。
3. SHOW TABLE tableName 展示某张表的模式信息。实现方式为获取当前数据库相应名称的表，遍历表中的 column，提取列名称、类型、长度等信息。

## 实现功能：

1. 实现了基本的查询语句功能（select from join on where）
2. Where 与 on 支持逻辑运算符。
3. 实现了 distinct 语句
4. 实现了表插入，删除，更新操作。
5. 实现了 show table, show database 语句。

## 实验结果

### 创建表并展示该表的模式信息

```
ThssDB>create table student(age int, name string(26), id string(26) not null, primary key(id));
create table student.
It costs 6 ms.
ThssDB>show table student;
(age,INT,0,false,-1)
(name,STRING,0,false,26)
(id,STRING,1,true,26)
It costs 15 ms.
```

### 展示某个数据库中所有表的名称

```
ThssDB>show database test;
student
It costs 2 ms.
```

### 展示所有数据库的名称

```

ThssDB>show databases;
test2
test
It costs 2 ms.
ThssDB>create database test3;
Create database test3.
It costs 5 ms.
ThssDB>show databases;
test2
test3
test
It costs 2 ms.
ThssDB>drop database test3;
Drop database test3.
It costs 2 ms.
ThssDB>show databases;
test2
test
It costs 2 ms.

```

标准查询：

```

ThssDB>select * from t_2;
t_2.t, t_2.k
-----
1, tk
2, po
3, hihhi

```

Where 子句

```

ThssDB>select * from t_2 where t>1;
t_2.t, t_2.k
-----
2, po
3, hihhi

```

Where 支持逻辑运算符号

```
ThssDB>select * from t_2 where t<3 and k>='po';
t_2.t, t_2.k
-----
1, tk
2, po
It costs 3 ms.
```

Distinct 语句

```
ThssDB>select * from t_1;
t_1.t, t_1.k
-----
1, 1
2, 2
3, null
4, 1
5, null
It costs 3 ms.
ThssDB>select distinct k from t_1;
t_1.k
-----
1
2
null
```

Join on 语句

```
ThssDB>select * from t_1 join t_2 on t_1.t < t_2.t;
t_1.t, t_1.k, t_2.t, t_2.k
-----
1, 1, 2, po
1, 1, 3, hihhi
2, 2, 3, hihhi
```

复杂语句综合：

```
ThssDB>select t_1.t,t_2.k from t_1 join t_2 on t_1.t<t_2.t or t_2.t<t_1.t where t_2.k>'po' or t_2.k = 'po';
t_1.t, t_2.k
-----
1, tk
1, po
2, po
```

Insert:

全属性的插入：

```
insert into table_name values(attr1_val, attr2_val,...);
```

属性值数量必须与列数相等，同时满足主键、非 null 约束。

```
ThssDB>insert into work values('zxl', 1);
Insert into work.
It costs 9 ms.
```

部分属性的插入:

insert into table\_name (attr1.attr2,...) values(attr1\_val, attr2\_val,...);  
指定的属性数量必须与值数量相等, 同时满足主键、非 null 约束。

```
ThssDB>insert into work (name) values('tk');
Insert into work.
It costs 4 ms.
```

Update:

Update table\_name set attr1 = value1 where attr\_2 =( <, >, <=, >=, !=) value2;

```
ThssDB>update work set salary = 3 where name='zxl';
Update 1 tuple from work.
It costs 24 ms.
```

Delete:

删除所有元组:

Delete from table\_name;

```
ThssDB>delete from work where salary=3;
Delete 1 tuple from work.
It costs 3 ms.
```

删除部分元组:

Delete from table\_name where attr =( <, >, <=, >=, !=) value;

```
ThssDB>delete from work;
Delete 1 tuple from work.
It costs 3 ms.
```

## 二、事务并发与恢复模块:

### 1、使用锁实现小规模并发与 read committed 隔离级别

-1、对于每张表格, 维护两个`Arraylist`: `SlockSessions` 与 `XlockSessions`, 分别记录给该表格上共享锁与排它锁的`Session`, 并且`locklevel`记录加锁级别, 实现函数`takeSLock`,`takeXLock`,`releaseSLock`,`releaseXLock`;

- 2、对于读锁, select 语句生成`QueryTable`前申请 S 锁, 生成完后释放 S 锁, 保证了

数据查询的过程中不会被修改。

- 3、对于写锁，当收到`begin transaction`指令时，将该客户端的`session`加入`manager.currentSessions`，开始维护对应`session`的`x\_lockDict`即该`session`用 X 锁占据的表名列表。在增、删、改表格数据时，会获得相应表格的 X 锁，`commit`后，将该`session`占据的锁全部释放，这样，所有的修改就只能在`commit`之后查看，实现了 Read Committed 级别的隔离机制。

- 4、阻塞机制，当客户端无法拿取表格需要的锁时，会陷入循环，只有表格能被上锁时才能继续执行。这时就需要 server 采用多线程的`ThreadPoolServer`来避免阻塞。

## 2、实现写 log 与读 log，重启时回复记录数据

- 1、目前写 log 根据框架只能写`session == 0`的 log，
- 2、`writelog`函数，获取当前的数据库，根据命名获取.log 记录文件并将指令追加写入
- 3、`readlog`函数，获取.log 记录文件，扫描一遍命令，重新执行除在 transaction 中但未被 commit 之外的所有指令，同时相应修改.log 文件，同时为了避免重新执行时再次记录一边 log 导致 log 记录成倍增加的情况，我们新增了一个不记录 log 的函数用与 log 恢复执行。

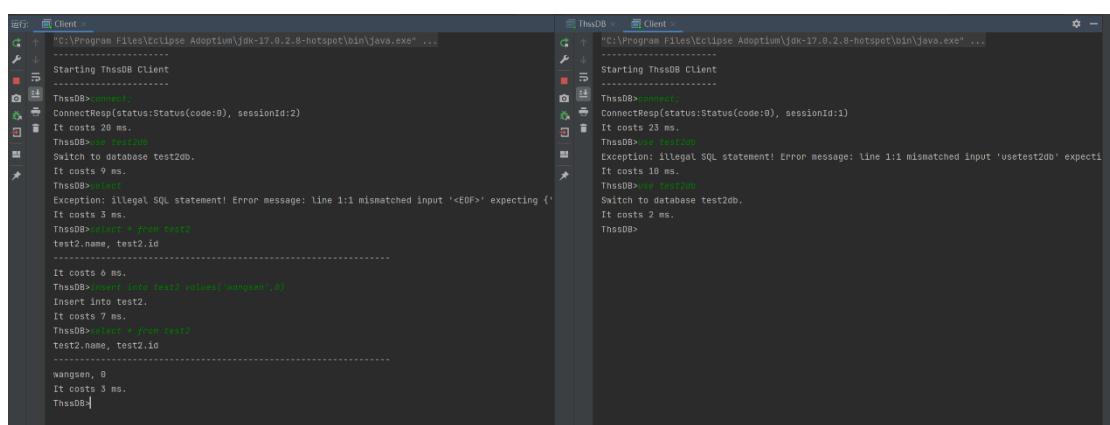
实现功能：

1. 实现 begin transaction 与 commit 语句，利用普通锁协议实现了 read committed 隔离级别。
2. 实现了写 log 与读 log，能够在重启时恢复数据。
3. 部分实现了多事务并发。

实验结果：

### 一、事务并发

原情况：



```
Client
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:2)
It costs 20 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 9 ms.
ThssDB> insert into test2
Exception: illegal SQL statement! Error message: line 1:1 mismatched input '<EOF>' expecting '('
It costs 3 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 8
It costs 6 ms.
ThssDB> insert into test2
Insert into test2.
It costs 7 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 8
It costs 3 ms.
ThssDB>

ThssDB
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:1)
It costs 23 ms.
ThssDB> use test2db
Exception: illegal SQL statement! Error message: line 1:1 mismatched input 'usetest2db' expecti
It costs 10 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 2 ms.
ThssDB>
```

事物未并发时，可以正常查询

```
Client
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:2)
It costs 20 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 9 ms.
ThssDB> insert
Exception: illegal SQL statement! Error message: line 1:1 mismatched input '<EOF>' expecting '('
It costs 3 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
It costs 6 ms.
ThssDB> insert values(wangsen, 0)
Insert into test2.
It costs 7 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
It costs 3 ms.
ThssDB>

ThssDB
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:1)
It costs 23 ms.
ThssDB> use test2db
Exception: illegal SQL statement! Error message: line 1:1 mismatched input 'usetest2db' expecti
It costs 19 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 2 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
It costs 3 ms.
ThssDB>
```

并发处理事物时，`session2`事物未提交的情况下，`session1`无法查询修改表后的信息

```
Client
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:2)
It costs 20 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 9 ms.
ThssDB> insert
Exception: illegal SQL statement! Error message: line 1:1 mismatched input '<EOF>' expecting '('
It costs 3 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
It costs 6 ms.
ThssDB> insert values(wangsen, 0)
Insert into test2.
It costs 7 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
It costs 3 ms.
ThssDB> start transaction.
start transaction.
It costs 1 ms.
ThssDB> insert values(wangsen, 0)
Insert into test2.
It costs 1 ms.
ThssDB>

ThssDB
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:1)
It costs 23 ms.
ThssDB> use test2db
Exception: illegal SQL statement! Error message: line 1:1 mismatched input 'usetest2db' expecti
It costs 19 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 2 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
It costs 3 ms.
ThssDB> select * from test2
|
```

`session2`提交后，`session1`能接触阻塞，查询到`session2`修改过后的信息

```
Client
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:2)
It costs 20 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 9 ms.
ThssDB> insert
Exception: illegal SQL statement! Error message: line 1:1 mismatched input '<EOF>' expecting '('
It costs 3 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
It costs 6 ms.
ThssDB> insert values(wangsen, 0)
Insert into test2.
It costs 7 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
It costs 3 ms.
ThssDB> start transaction.
start transaction.
It costs 1 ms.
ThssDB> insert values(wangsen, 0)
Insert into test2.
It costs 1 ms.
ThssDB> commit transaction.
commit transaction.
It costs 2 ms.
ThssDB>

ThssDB
"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting ThssDB Client
ThssDB> connect
ConnectResp(status:Status(code:0), sessionId:1)
It costs 23 ms.
ThssDB> use test2db
Exception: illegal SQL statement! Error message: line 1:1 mismatched input 'usetest2db' expecti
It costs 19 ms.
ThssDB> use test2db
Switch to database test2db.
It costs 2 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
It costs 3 ms.
ThssDB> select * from test2
test2.name, test2.id
-----
wangsen, 0
ws, 1
It costs 33477 ms.
ThssDB>
```

## 二、日志读写与恢复

server 关闭前的状况

```
Client
C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting Thsdb Client
-----
Thsdb>
ConnectResp(status:Status(code:0), sessionId:0)
It costs 30 ms.
Thsdb>
Switch to database test2db.
It costs 85 ms.
Thsdb> insert into test2 values('ws',0)
Insert into test2.
It costs 10 ms.
Thsdb> select * from test2
Exception: illegal SQL statement! Error message: line 1:1 mismatched input 'values' expecting I
It costs 6 ms.
Thsdb> select * from test2
test2.name, test2.id
-----
ws, 0
It costs 0 ms.
Thsdb>

Thsdb
Client
C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
??? try to recover manager
???testdb
! try to recover database testdb
Table test: (name,STRING,0,false,10) (id,INT,1,true,-1)
??? try to read log
???test2db
! try to recover database test2db
Table test2: (name,STRING,0,false,10) (id,INT,1,true,-1)
??? try to read log
[Thread-0] INFO cn.edu.thsdb.server.Thsdb - Starting Thsdb ...
session:0 use test2db
session:0 begin transaction
session:0 insert into test2 values('ws',0)
session:0 commit
session:0 begin transaction
session:0 select * from values
session:0 commit
session:0 begin transaction
session:0 select * from test2
test2.name STRING
test2.id INT
session:0 commit
已经启动, 等待数据130
```

server 关闭后重新开启，日志回复记录，client 与 server 重新连接以后，也能查询到断开之前的数据。

```
Client
C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
Starting Thsdb Client
-----
Thsdb>
ConnectResp(status:Status(code:0), sessionId:0)
It costs 30 ms.
Thsdb>
Switch to database test2db.
It costs 85 ms.
Thsdb> insert into test2 values('ws',0)
Insert into test2.
It costs 10 ms.
Thsdb> select * from test2
Exception: illegal SQL statement! Error message: line 1:1 mismatched input 'values' expecting I
It costs 6 ms.
Thsdb> select * from test2
test2.name, test2.id
-----
ws, 0
It costs 0 ms.
Thsdb>
DisconnectResp(status:Status(code:0))
It costs 22 ms.
Thsdb>
ConnectResp(status:Status(code:0), sessionId:0)
It costs 11 ms.
Thsdb>
Switch to database test2db.
It costs 14 ms.
Thsdb> select * from test2
test2.name, test2.id
-----
ws, 0
It costs 6 ms.
Thsdb>

Thsdb
Client
C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" ...
??? try to recover manager
???testdb
! try to recover database testdb
Table test: (name,STRING,0,false,10) (id,INT,1,true,-1)
??? try to read log
???test2db
! try to recover database test2db
Table test2: (name,STRING,0,false,10) (id,INT,1,true,-1)
??? try to read log
session:0 begin transaction
session:0 insert into test2 values('ws',0)
session:0 commit
session:0 begin transaction
session:0 commit
session:0 begin transaction
session:0 commit
[Thread-0] INFO cn.edu.thsdb.server.Thsdb - Starting Thsdb ...
session:0 use test2db
session:0 begin transaction
session:0 select * from test2
test2.name STRING
test2.id INT
session:0 commit
```

### 三、小组分工

谭恺：实现 select 查询及相关子句

张兴龙：实现表创建，表插入，表删除，表更新及相关子句。

翟浩清：实现 show table，show database 子句

王森：实现事务的并发与恢复功能。