

《数据库原理》大作业用户文档

常珂华 2019010798 刘臣洋 2019011250 杨健辉 2019011248 张博闻 2019013267

一、运行方式

准备工作

删除根目录下 `thssdb\data` 里的所有文件。在IDEA中执行 `maven install`，并指定根目录下 `target\generated-sources` 为source root。

运行Server

在IDER中运行根目录下 `src/main/java/cn/edu/thssdb/server/ThssDB.java` 的 `main()` 函数。

运行Client

在IDER中运行根目录下 `src/main/java/cn/edu/thssdb/client/Client.java` 的 `main()` 函数。

二、SQL语句

连接服务端

```
connect;
```

断开连接

```
disconnect;
```

退出用户端

```
quit;
```

创建数据库

```
create database name;
```

删除数据库

```
drop database name;
```

创建数据表

```
create table tableName(AttrName1 AttrType1 [not null], ... ,AttrNameN AttrTypeN [not null],primary key(AttrNameX));
create table tableName(AttrName1 AttrType1 [not null][primary key], ... ,AttrNameN AttrTypeN [not null]);
```

创建数据表时，必须且只能指定一个属性作为主键。有两种方式指定主键。[] 为可选项，下同。

删除表

```
drop table tableName;
```

展示表模式信息

```
show table tableName;
```

插入语句

```
insert into tableName [(attrName1, attrName2,..., attrNameN)] values (attrValue1, attrValue2,..., attrValueN);
```

指定插入属性时，声明的属性需包含所有 `not null` 属性，且个数需与属性值个数相等。

更新语句

```
update tableName set attrName = attrValue  
[where attrName1 = attrValue1 [and/or attrName2 = attrValue2] ];
```

`where` 支持使用 `and` 或 `or` 连接的两个比较运算。比较运算包括：`<`、`>`、`<=`、`>=`、`=`、`<>`，其左运算符需为属性名，右运算符需为常元。

删除语句

```
delete from tableName [where attrName1 = attrValue1 [and/or attrName2 = attrValue2] ];
```

`where` 支持使用 `and` 或 `or` 连接的两个比较运算。比较运算包括：`<`、`>`、`<=`、`>=`、`=`、`<>`，其左运算符需为属性名，右运算符需为常元。

选择语句

```
#自然连接  
select attrName1, ... ,attrName2 from table1[,table2]  
[where attrName1 = attrValue1 [and/or attrName2 = attrValue2] ];  
#笛卡尔积并用on所接表达式筛选结果  
select * from table1 join table2 on attrName1 = attrName2  
[where attrName1 = attrValue1];
```

选择语句的属性名需写作 `table.attrname` 的形式。对自然连接语句的重复属性，在 `select` 语句中，只能用 `from` 语句的第一个表即 `table1.attrName` 来指代。

`where` 支持使用 `and` 或 `or` 连接的两个比较运算。比较运算包括：`<`、`>`、`<=`、`>=`、`=`、`<>`，其左运算符需为属性名，右运算符需为常元。

`on` 支持使用 `and` 或 `or` 连接的两个比较运算。比较运算包括：`<`、`>`、`<=`、`>=`、`=`、`<>`，其左运算符需为属性名，右运算符可为常元或属性名。

开始事务

```
begin transaction;
```

本数据库采用 `read committed` 的隔离机制。当事务A在修改某一数据表时，事务B若想访问该表，B会被阻塞直至A事务提交。

提交事务

```
commit;
```