SSQL 测试文档

目录

一、测试简介.....3
二、测试内容.....3
2.1 创建功能测试.....3
2.1.1 正确创建.....3
2.1.2 语法错误.....3
2.1.3 插入重复列.....4
2.1.4 指定多个主值.....4
2.1.5 指定主值不存在.....5
2.1.6 创建已存在表格.....5
2.2 插入功能测试.....5
2.2.1 正确插入.....5
2.2.2 语法错误.....6
2.2.3 插入不存在表格.....6
2.2.4 插入重复列.....6
2.2.5 列不存在.....7
2.2.6 列值数量不对应.....7
2.2.7 主值已存在.....8
2.3 删除功能测试.....8
2.3.1 正确删除.....8
2.3.2 语法错误.....9
2.3.3 表格不存在.....9
2.3.4 属性不存在.....9
2.4 查询功能测试.....10
2.4.1 正确查询.....10
2.4.2 语法错误.....11
2.4.3 表格不存在.....11
2.4.4 返回属性不存在.....12
2.4.5 查询属性不存在.....12
2.5 布尔表达式功能测试.....13
2.5.1 逻辑与.....13
2.5.2 逻辑或.....15
2.5.3 逻辑非.....15
2.6 算术表达式功能测试.....17
2.6.1 加法.....17
2.6.2 减法.....18
2.6.3 乘法.....18
2.6.4 除法.....19
三、总结.....20

一、测试简介

解释器除了实现基本的 SSQL 增删查改功能外，还实现了布尔表达式、逻辑表达式功能。这样，解释器可以分为六个子功能：创建表格、插入列、删除列、查询表格、布尔表达式、逻辑表达式。

测试针对这六个子功能分别进行测试。对于每一个子功能，首先测试正确语句，接着根据该子功能容易出错的几项内容分别进行测试。比如，对于创建一个表格常见的错误有：语法错误、插入重复列、指定多个主值、指定的主值不存在、创建已存在的表格。在测试创建功能项时，我们针对性地对这几项常见错误进行测试。

二、测试内容

2.1 创建功能

本项主要测试 SSQL 的创建表格功能，测试内容有正确创建语句、语法错误、插入重复列、指定多个主值、指定的主值不存在、创建已存在的表格。

2.1.1 正确创建

1.测试语句：

CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));

预期结果：创建表格成功

测试结果：成功创建表格

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
Create table 'Student' succeed!
```

2.测试语句：不指定 DETAULT 值

CREATE TABLE Teacher(sid INT, age INT, PRIMARY KEY(sid));

预期结果：创建表格成功

测试结果：成功创建表格

```
>> CREATE TABLE Teacher(sid INT, age INT, PRIMARY KEY(sid));
Create table 'Teacher' succeed!
```

2.1.2 语法错误

1.测试语句：

CREATE TABLE Student(sid INT, age INT DEFAULT, PRIMARY KEY(sid));

预期结果：语法错误，DEFAULT 未指定值

测试结果：语法错误

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT, PRIMARY KEY(sid));
Grammar Error :
CREATE TABLE Student(sid INT, age INT DEFAULT, PRIMARY KEY(sid));
                                              ^
```

2.测试语句：

CREATE Student (sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid, age));

预期结果：语法错误

测试结果：语法错误

```
>> CREATE Student (sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid, age));
Grammar Error :
CREATE Student (sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid, age));
       ^^^^^^^
```

2.1.2 插入重复列

1.测试语句：

CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid), age INT);

预期结果：错误，age 列重复

测试结果：错误，age 列重复

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid), age INT
);
Error :  The table has the same fields called 'age' when you create it!
```

2.测试语句：

CREATE TABLE Student(sid INT, sid INT DEFAULT = 18, PRIMARY KEY(sid));

预期结果：错误，sid 列重复

测试结果：错误，sid 列重复

```
>> CREATE TABLE Student(sid INT, sid INT DEFAULT = 18, PRIMARY KEY(sid));
Error :  The table has the same fields called 'sid' when you create it!
```

2.1.3 指定多个主值

1.测试语句：

CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid), PRIMARY KEY(age));

预期结果：错误，重复主值

测试结果：错误，重复主值

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid), PRIMARY KEY(age));
Error :  The statment has two primary key declarations
```

2.1.4 指定主值不存在

1.测试语句：

CREATE TABLE Student(sid INT, age INT DEFAULT = 18,

PRIMARY KEY(height));
预期结果：错误，指定主值不存在
测试结果：错误，指定主值不存在

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(height));
Error :  The primary key called 'height' doesn't in the table.
```

2.测试语句：
CREATE TABLE Student(sid INT, sid INT DEFAULT = 18, PRIMARY KEY(sid, height));
预期结果：错误，主值 height 不存在
测试结果：错误，主值 height 不存在

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid,height));
Error :  The primary key called 'height' doesn't in the table.
```

2.1.5 创建已存在表格

1.测试语句：
CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
预期结果：第二句错误，Student 表格已存在
测试结果：第二句错误，Student 表格已存在

```
>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
Create table 'Student' succeed!

>> CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
Error :  The table has been created before!
```

2.2 插入功能

本项主要测试 SSQL 的插入列功能，测试内容有正确插入语句、语法错误、插入不存在表格、重复列、列不存在，列值数量不对应,主值已存在。首先使用
CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
创建了 Student 表格。

2.2.1 正确插入

1.测试语句：
INSERT INTO Student(sid, age) VALUES(1111, 18);
预期结果：插入成功
测试结果：成功插入一列

```
>> INSERT INTO Student(sid, age) VALUES(1111, 18);
insert value to table 'Student' succeed!
```

2.2.2 语法错误

1.测试语句：

INSERT INTO Student Teacher(sid, age) VALUES(1111, 18);

预期结果：语法错误

测试结果：语法错误

```
>> INSERT INTO Student Teacher(sid, age) VALUES(1111, 18);
Grammar Error :
INSERT INTO Student Teacher(sid, age) VALUES(1111, 18);
                    ^^^^^^^
```

2.测试语句：

INSERT INTO Student (sid, age) VALUES(coco, 18);

预期结果：语法错误

测试结果：语法错误

```
>> INSERT INTO Student(sid, age) VALUES(coco, 18);
Grammar Error :
INSERT INTO Student(sid, age) VALUES(coco, 18);
                                     ^^^^
```

2.2.3 插入不存在的表格

1.测试语句：

INSERT INTO Teacher(sid, age) VALUES(1111, 18);

预期结果：错误，Teacher 表格不存在

测试结果：错误，Teacher 表格不存在

```
>> INSERT INTO Teacher(sid, age) VALUES(1111, 18);
Error :  The table called 'Teacher' has not been created yet!
```

2.2.4 插入重复列

1.测试语句：

INSERT INTO Student(sid, age, age) VALUES(1111, 18, 19);

预期结果：错误，age 列重复

测试结果：错误，age 列重复

```
>> INSERT INTO Student(sid, age, age) VALUES(1111, 18, 19);
Error :  The table has the same fields called 'age' when you create it!
```

2.测试语句：

INSERT INTO Student(sid, sid, age) VALUES(1111, 1111, 19);

预期结果：错误，sid 列重复

测试结果：错误，sid 列重复

```
>> INSERT INTO Student(sid, sid, age) VALUES(1111, 1111, 19);
Error :  The table has the same fields called 'sid' when you create it!
```

2.2.5 列不存在

1.测试语句：

INSERT INTO Student(sid, height) VALUES(1111, 18);

预期结果：错误，height 列不存在

测试结果：错误，height 列不存在

```
>> INSERT INTO Student(sid, height) VALUES(1111, 18);
Error :  The field called 'height' doesn't in this table!
```

2.测试语句：

INSERT INTO Student(height) VALUES(1111);

预期结果：错误，height 列不存在

测试结果：错误，height 列不存在

```
>> INSERT INTO Student(height) VALUES(1111);
Error :  The field called 'height' doesn't in this table!
```

2.2.6 列值数量不对应

1.测试语句：

INSERT INTO Student(sid) VALUES(1111, 18);

预期结果：错误，列值数量不对应

测试结果：错误，列值数量不对应

```
>> INSERT INTO Student(sid) VALUES(1111, 18);
Error :  The size of fields and the size of value don't match.
```

2.测试语句：

INSERT INTO Student(sid, age) VALUES(1111);

预期结果：错误，列值数量不对应

测试结果：错误，列值数量不对应

```
>> INSERT INTO Student(sid, age) VALUES(1111);
Error :  The size of fields and the size of value don't match.
```

2.2.7 主值已存在

1.测试语句：

INSERT INTO Student(sid, age) VALUES(1111, 18);
INSERT INTO Student(sid, age) VALUES(1111, 18);

预期结果：第二句错误，主值已存在

测试结果：第二句错误，主值已存在

```
>> INSERT INTO Student(sid, age) VALUES(1111, 18);
Error :  The primary key 'sid' has the same value '1111' after you insert!
```

2.3 删除功能

本项主要测试 SSQL 的删除列功能，测试内容有正确插入语句、语法错误、表格不存在、属性不存在。每项使用

CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
INSERT INTO Student(sid, age) VALUES(1111, 18);
INSERT INTO Student(sid, age) VALUES(1112, 19);
INSERT INTO Student(sid, age) VALUES(1113, 23);

创建了 Student 表格,里面含有 3 行。

2.3.1 正确删除

1.测试语句：

DELETE FROM Student WHERE age > 10 && age < 20;

预期结果：删除成功

测试结果：删除成功

```
>> DELETE FROM Student WHERE age > 10 && age < 20;
Delete OK
```

```
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
```

2.测试语句：

DELETE FROM Student ;

预期结果：删除成功

测试结果：删除成功

```
>> DELETE FROM Student;
Delete OK
```

```
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
```

2.3.2 语法错误

1.测试语句：

DELETE FROM Student age > 10;

预期结果：语法错误

测试结果：语法错误

```
>> DELETE FROM Student age > 10;
Grammar Error :
DELETE FROM Student age > 10;
                    ^^^
```

2.测试语句：

DELETE Student WHERE age > 10;

预期结果：语法错误

测试结果：语法错误

```
>> DELETE Student age > 10;
Grammar Error :
DELETE Student age > 10;
       ^^^^^^^
```

2.3.3 表格不存在

1.测试语句：

DELETE FROM Teacher WHERE age > 10;

预期结果：表格不存在

测试结果：表格不存在

```
>> DELETE FROM Teacher WHERE age > 10;
Error :  The table called 'Teacher' has not been created yet!
```

1.测试语句：

DELETE FROM Teacher;

预期结果：表格不存在

测试结果：表格不存在

```
>> DELETE FROM Student WHERE age > 10 && age < 20;
Delete OK
```

2.3.4 属性不存在

1.测试语句：

DELETE FROM Student WHERE height > 10;

预期结果：height 属性不存在

测试结果：height 属性不存在

```
>> DELETE FROM Student WHERE height > 10;
Error :  The field in conditions called height doesn't in this table!
```

1.测试语句：

DELETE FROM Student WHERE age > 10 && height > 10;

预期结果：height 属性不存在

测试结果：height 属性不存在

```
>> DELETE FROM Student WHERE age > 10 && height > 10;
Error :  The field in conditions called height doesn't in this table!
```

2.4 查询功能

本项主要测试 SSQL 的删除列功能，测试内容有正确查询语句、语法错误、表格不存在、返回属性不存在，查询属性不存再。首先使用

CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));

INSERT INTO Student(sid, age) VALUES(1111, 18);

INSERT INTO Student(sid, age) VALUES(1112, 19);

INSERT INTO Student(sid, age) VALUES(1113, 23);

创建了 Student 表格,里面含有 3 行。

2.4.1 正确查询

1.测试语句：

SELECT sid FROM Student WHERE age < 20;

预期结果：查询成功

测试结果：查询成功

```
>> SELECT sid FROM Student WHERE age < 20;
|---------------|
|       sid     |
|---------------|
|       1112    |
|---------------|
|       1111    |
|---------------|
Query OK
```

2.测试语句：

SELECT * FROM Student ;

预期结果：查询成功

测试结果：查询成功

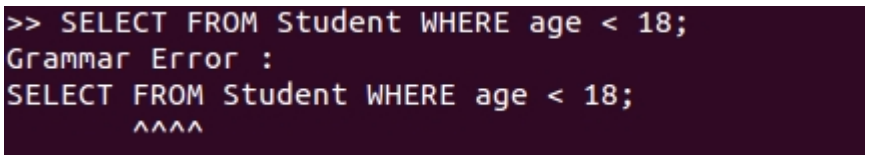
2.4.2 语法错误

1.测试语句：

SELECT FROM Student WHERE age < 18;

预期结果：语法错误

测试结果：语法错误

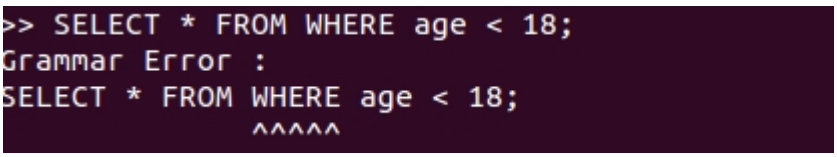


2.测试语句：

SELECT * FROM WHERE age < 18;

预期结果：语法错误

测试结果：语法错误



2.4.3 表格不存在

1.测试语句：

SELECT * FROM Teacher WHERE age > 10;

预期结果：表格不存在

测试结果：表格不存在

```
>> SELECT * FROM Teacher WHERE age > 10;
Error :  The table called 'Teacher' has not been created yet!
```

2.测试语句：

SELECT sid FROM Boy WHERE age > 10;

预期结果：表格不存在

测试结果：表格不存在

```
>> SELECT sid FROM Boy WHERE age > 10;
Error :  The table called 'Boy' has not been created yet!
```

2.4.4 返回属性不存在

1.测试语句：

SELECT sid, height FROM Student WHERE age < 18;

预期结果：返回字段不存在

测试结果：返回字段不存在

```
>> SELECT sid,height FROM Student WHERE age < 18;
Error :  The field selected called 'height' doesn't in this table!
```

2.测试语句：

SELECT height FROM Student WHERE age < 18;

预期结果：返回字段不存在

测试结果：返回字段不存在

```
>> SELECT height FROM Student WHERE age < 19;
Error :  The field selected called 'height' doesn't in this table!
```

2.4.5 查询属性不存在

1.测试语句：

SELECT * FROM Student WHERE height > 10;

预期结果：查询属性不存在

测试结果：查询属性不存在

```
>> SELECT * FROM Student WHERE height > 10;
Error :  The field in conditions called 'height' doesn't in this table!
```

2.测试语句：

SELECT * FROM Student WHERE age < 10 && height > 10;

预期结果：查询属性不存在

测试结果：查询属性不存在

```
>> SELECT * FROM Student WHERE age < 10 && height < 10;
Error :  The field in conditions called 'height' doesn't in this table!
```

2.5 布尔表达式

本项主要测试 SSQL 的布尔表达式功能，测试内容有逻辑与、逻辑或、逻辑非。首先使用

CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
INSERT INTO Student(sid, age) VALUES(1111, 18);
INSERT INTO Student(sid, age) VALUES(1112, 19);
INSERT INTO Student(sid, age) VALUES(1113, 23);
INSERT INTO Student(sid, age) VALUES(1114, 19);

创建了 Student 表格,里面含有 4 行。

2.5.1 逻辑与

1.测试语句：

SELECT * FROM Student WHERE age == 19 && sid == 1112;

预期结果：

sid	age
1112	19

测试结果：

```
>> SELECT * FROM Student WHERE age == 19 && sid == 1112;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1112    |       19      |
|---------------|---------------|
Query OK, 1 rows affected.
```

2.测试语句：

SELECT * FROM Student WHERE age > 18 && age < 20 && sid > 1112 ;

预期结果：

sid	age
1112	19

测试结果：

```
>> SELECT * FROM Student WHERE age > 18 && age < 20 && sid > 1112;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1114    |       19      |
|---------------|---------------|
Query OK, 1 rows affected.
```

3.测试语句：

SELECT * FROM Student WHERE age > 18 && age < 25 ;

预期结果：

sid	age
1112	19
1113	23
1114	19

测试结果：

```
>> SELECT * FROM Student WHERE age > 18 && age < 25;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1112    |       19      |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
|       1114    |       19      |
|---------------|---------------|
Query OK, 3 rows affected.
```

2.5.2 逻辑或

1.测试语句：

SELECT * FROM Student WHERE age == 19 || sid == 1111;

预期结果：

sid	age
1112	19
1111	18
1114	19

测试结果：

```
>> SELECT * FROM Student WHERE age == 19 || sid == 1111;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1112    |       19      |
|---------------|---------------|
|       1111    |       18      |
|---------------|---------------|
|       1114    |       19      |
|---------------|---------------|
Query OK, 3 rows affected.
```

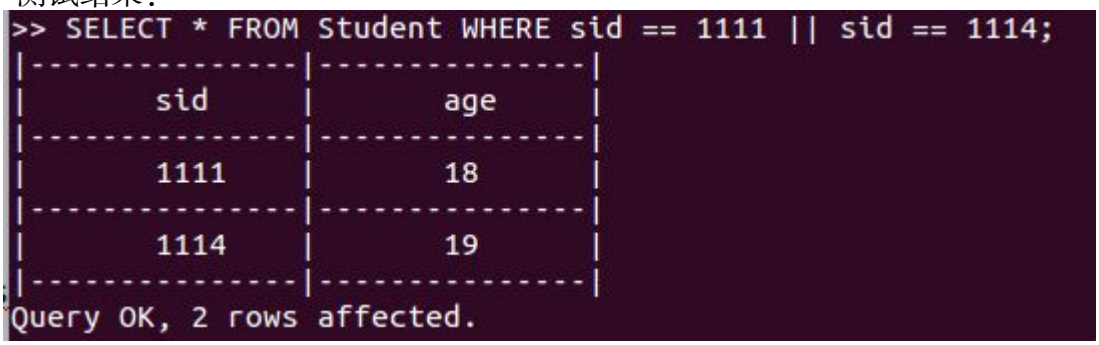
2.测试语句：

SELECT * FROM Student WHERE sid == 1111 || sid == 1113;

预期结果：

sid	age
1111	18
1113	19

测试结果：



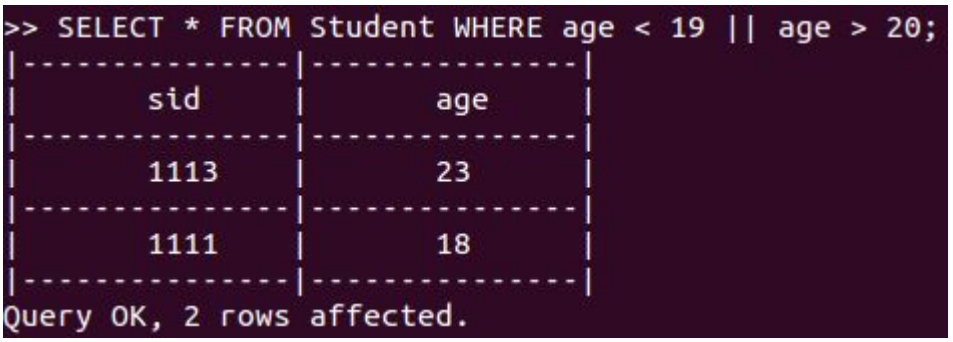
3.测试语句：

SELECT * FROM Student WHERE age < 19 || age > 20 ;

预期结果：

sid	age
1113	23
1111	18

测试结果：



2.5.3 逻辑非

1.测试语句：

SELECT * FROM Student WHERE !(age == 19);

预期结果：

sid	age
1113	23
1111	18

测试结果：

```
>> SELECT * FROM Student WHERE !(age == 19);
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
|       1111    |       18      |
|---------------|---------------|
Query OK, 2 rows affected.
```

2.测试语句：

SELECT * FROM Student WHERE !(sid == 1111 || age == 19);

预期结果：

sid	age
1113	23

测试结果：

```
>> SELECT * FROM Student WHERE !(sid == 1111 || age == 19);
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
Query OK, 1 rows affected.
```

3.测试语句：

SELECT * FROM Student WHERE !sid == 1111 && !age == 19;

预期结果：

sid	age
1113	23

测试结果：

```
>> SELECT * FROM Student WHERE !sid == 1111 && !age == 19;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
Query OK, 1 rows affected.
```

2.6 算术表达式

本项主要测试 SSQL 的布尔表达式功能，测试内容有加法运算、减法运算、乘法运算、除法运算。首先使用

```
CREATE TABLE Student(sid INT, age INT DEFAULT = 18, PRIMARY KEY(sid));
INSERT INTO Student(sid, age) VALUES(1111, 18);
INSERT INTO Student(sid, age) VALUES(1112, 19);
INSERT INTO Student(sid, age) VALUES(1113, 23);
INSERT INTO Student(sid, age) VALUES(1114, 19);
```

创建了 Student 表格,里面含有 4 行。

2.6.1 加法运算

1.测试语句：

预期结果：

sid	age

测试结果：

```
>> SELECT * FROM Student WHERE age + 5 < 18 + 1 + 2;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
Query OK, 0 rows affected.
```

2.测试语句：

```
SELECT * FROM Student WHERE sid + age < 1135;
```

预期结果：

sid	age
1112	19
1111	18
1114	19

测试结果：

```
>> SELECT * FROM Student WHERE sid + age < 1135;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1112    |       19      |
|---------------|---------------|
|       1111    |       18      |
|---------------|---------------|
|       1114    |       19      |
|---------------|---------------|
Query OK, 3 rows affected.
```

2.6.2 减法运算

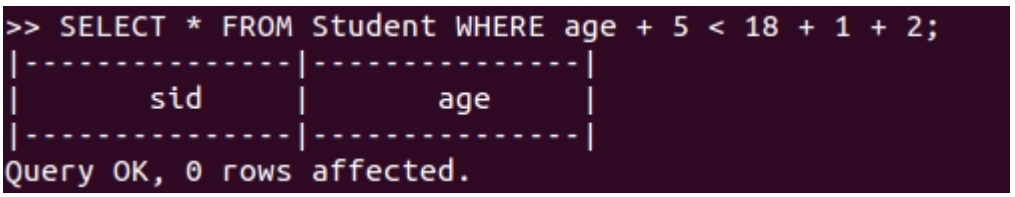
1.测试语句：

SELECT * FROM Student WHERE age + 5 < 18 +1 + 2;

预期结果：

sid	age

测试结果：



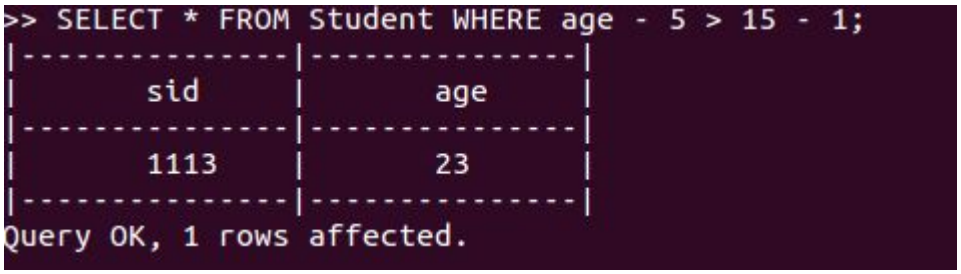
2.测试语句：

SELECT * FROM Student WHERE age - 5 > 15 - 1;

预期结果：

sid	age
1113	23

测试结果：



2.6.3 乘法运算

1.测试语句：

SELECT * FROM Student WHERE age * 5 <= 100;

预期结果：

sid	age
1112	19
1111	18
1114	19

测试结果：

```
>> SELECT * FROM Student WHERE age - 5 > 15 - 1;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
Query OK, 1 rows affected.
```

2. 测试语句：

SELECT * FROM Student WHERE age * 6 > 105 + 4;

预期结果：

sid	age
1113	23

测试结果：

```
>> SELECT * FROM Student WHERE age * 6 - 15 > 105 + 4;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
Query OK, 1 rows affected.
```

2.6.4 除法运算

1.测试语句：

SELECT * FROM Student WHERE age / 0 <= 100;

预期结果：错误，除 0 运算

测试结果：

```
>> SELECT * FROM Student WHERE age / 0 <= 100;
Error :  Number divided by 0 in the expressions
```

2. 测试语句：

SELECT * FROM Student WHERE age / 4 > 2*3 - 2;

预期结果：

sid	age
1113	23

测试结果：

```
>> SELECT * FROM Student WHERE age / 4 > 2*3-2;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
Query OK, 1 rows affected.
```

3. 测试语句：

SELECT * FROM Student WHERE age > 6*3 / 4 + 10 / 5 - 2 + 14;

预期结果：

sid	age
1114	19
1113	23
1112	19

测试结果：

```
>> SELECT * FROM Student WHERE age > 6*3 / 4 + 10 / 5 - 2 + 14;
|---------------|---------------|
|       sid     |       age     |
|---------------|---------------|
|       1114    |       19      |
|---------------|---------------|
|       1113    |       23      |
|---------------|---------------|
|       1112    |       19      |
|---------------|---------------|
Query OK, 3 rows affected.
```

三、总结

经测试，解释器实现了数据库创建、插入、删除、查询、布尔表达式运算、算术表达式运算六个子功能，测试过程中未出现功能异常。