

简介

- 本文档主要介绍项目二需要实现的简化版SQL语句的格式与含义，建议在实现每种语句之前先阅读对应的要求。文档中存在的错误将在FAQ上订正，请保持关注。
- 文档中所有提到“给出用户提示”的地方，若没有明确要求输出的格式和内容，则可自行设计。
- 文档中“列”与“字段”含义相同。
- 项目实现的数据库没有类型系统，即每一个数据都当做字符串保存。
- 项目的入口可以是一个 `while` 循环：

```
int main()
{
    while(true)
    {
        // 接收用户输入的一行指令
        // ...

        // 处理用户输入；将修改后的数据更新到文件中；输出结果
        // ...
    }
    return 0;
}
```

- 测试样例保证
 - 所有语句的格式均与本文档相同，例如不会把 `SELECT` 拼成 `SLECT`、`SELECT` 和 `FROM` 之间一定不会出现表名
 - 文档中提到的所有“字符串”，都可以使用 `char[100]` 保存，不会溢出
 - 表格总数、单一表格中的字段数、单一表格的最大行数均不会超过100

CREATE

- 基本格式

```
CREATE TABLE [A] ([B]);
```

- 含义
创建一个表名为 `[A]` 的新表，该表的字段、主键如 `[B]` 所示
- 占位符格式与含义
 - `[A]` 可以为
 - 一个字符串，表示新表的表名
 - `[B]` 可以为
 - 使用英文逗号分隔的若干个字段名；其中某一个字段后可以带 `PRIMARY_KEY` 关键字（该字段与 `PRIMARY_KEY` 之间用空格分隔），表示主键
- 样例

```
CREATE TABLE student (id PRIMARY_KEY,name,birth_place);
CREATE TABLE test (column_one,column_two PRIMARY_KEY);
CREATE TABLE test2 (column_one);
```

- 备注
 - 若新表创建成功，需要给出提示
 - 若表名已存在，需要给出提示，并拒绝新表的创建
 - 测试样例保证 [B] 至少包含一个字段，字段名各不相同，且最多只有一个字段是主键
 - 测试样例保证 [B] 中每个字段的字段名只由26个字母（包括大小写）和下划线 _ 构成

SHOW TABLES

- 基本格式

```
SHOW TABLES;
```

- 含义

列出数据库中现存的表的数量、每个表的表名和字段名（包括 PRIMARY_KEY 关键字）

- 备注

输出格式样例：

```
2 table(s):
student(id PRIMARY_KEY,name,birth_place)
department(name)
```

和

```
0 tables(s):
```

DROP

- 基本格式

```
DROP TABLE [A];
```

- 含义

删除表名为 [A] 的表

- 占位符格式与含义

- [A] 可以为
 - 一个字符串，表示要删除的表的表名

- 样例

```
DROP TABLE student;
```

- 备注

- 若表 [A] 删除成功，需要给出用户提示
- 若表 [A] 不存在，需要给出用户提示

INSERT

- 基本格式

```
INSERT INTO [A] VALUES ([B]);
```

- 含义

向表 [A] 插入一行新数据，该行数据每一个字段的数值如 [B] 所示

- 占位符格式与含义

- [A] 可以为
 - 一个字符串，表示要插入数据的表的表名
- [B] 可以为
 - 使用英文逗号分隔的若干个字符串，每个字符串表示在某一列的数据；列数据的顺序与创建表 [A] 时的字段名顺序相同

- 样例

```
INSERT INTO student VALUES (123,张三,江苏);
```

- 备注

- 若新数据插入成功，需要给出用户提示
- 若新数据违反主键约束（即新数据与某一行旧数据在标注为主键的字段上重复），需要给出用户提示，并拒绝新数据的插入
- 若 [B] 中的列数据个数与 [A] 的字段个数不相同，需要给出用户提示
- 需要考虑 [B] 为空字符串的情况，即 [B] 中列数据个数为0，此时列数据个数显然与 [A] 的字段个数不同
- 测试用例保证 [B] 中每个列数据不包含 `,`、`(`、`)`、`;` 和空格这五种字符

SELECT

- 基本格式

```
SELECT [A] [B] FROM [C] [D];
```

- 含义

找出表 [C] 中满足 [D] 条件的所有行，并打印这些行的 [B] 字段；若 [A] 为 `DISTINCT`，还需要对输出结果去重

- 占位符格式与含义

- [A] 可以为
 - 空字符串，即不需要对结果去重
 - `DISTINCT`，即需要对结果去重
- [B] 可以为
 - 使用英文逗号分隔的若干个字段名
 - `*`，表示表 [C] 中的所有字段，顺序与创建该表时相同
- [C] 可以为
 - 一个字符串，表示要查询的表的表名
- [D] 可以为
 - 空字符串，表示没有筛选条件，[C] 表中的所有行都需要处理

- `WHERE [E]`，表示需要筛选出满足条件 `[E]` 的行
 - `[E]` 可以为
 - `[F]`，表示满足条件 `[F]` 即可
 - `[F] AND [F]`，表示同时满足两个条件
 - `[F] OR [F]`，表示满足两个条件之一即可
 - `[F]` 可以为
 - 形如 `column_name=[G]` 的表达式，即该行在 `column_name` 列的数据与 `[G]` 相等
- 样例

```
SELECT a,b,c FROM student;
SELECT b,a,c FROM student;
SELECT DISTINCT a,b,c FROM student;
SELECT DISTINCT * FROM student;
SELECT * FROM student;
SELECT * FROM student WHERE a=张三;
SELECT * FROM student WHERE a=张三 AND b=广东;
SELECT * FROM student WHERE a=张三 OR b=广东;
```

- 备注
 - 为了方便表示，“基本格式”中没有对空格数量做严谨的表示。请注意：当 `[A]` 为空字符串时，`SELECT` 与 `[B]` 之间只有一个空格（而不是如“基本格式”所示的两个空格）；当 `[D]` 为空字符串时，`[c]` 与 `;` 之间没有空格（而不是如“基本格式”所示的一个空格）
 - 若表 `[C]` 不存在、或 `[B]` 中指定的字段不存在、或 `[D]` 中指定的字段不存在，需要给出用户提示
 - 若语句合法，需要查询数据并输出。首先输出一行字段名（若 `[B]` 为 `*`，则按照建表时的字段顺序，输出所有字段名；否则，按照 `[B]` 中指定的顺序），然后依次输出每一行的数据。行与行之间的输出顺序没有要求。字段名之间、列数据之间可以使用制表符 `\t` 分隔，测试样例保证不会出现排版错乱的问题。

- 输出格式样例：

id	name	birth_place
123	张三	江苏
124	李四	广东

- 若 `[A]` 为 `DISTINCT`，需要去重。去重的含义是：先按照没有 `DISTINCT` 关键字的方式查找数据，然后相同的行只需要输出一次。两行数据相同是指：`[B]` 所指定的所有字段的数据都相同）。

- 例如：若表为

a	b	c
1	2	3
1	3	1
1	2	4
1	2	5

则

```
SELECT DISTINCT a,b FROM table_name;
```

的输出是

```
a  b
1  2
1  3
```

UPDATE

- 基本格式

```
UPDATE [A] SET [B] [C];
```

- 含义

将表 [A] 中满足条件 [C] 的行按照 [B] 语句更新数据

- 占位符格式与含义

- [A] 可以为
 - 一个字符串，表示要更新的表的表名
- [B] 可以为
 - 形如 `column_name=xxx` 的表达式，表示将该行中名为 `column_name` 的列更新为 `xxx`
- [C] 的定义与 `SELECT` 语句中 [D] 的定义完全相同，即：可以是空字符串，表示处理所有行；也可以是形如 `WHERE ...` 的筛选表达式，表示只处理满足条件的行

- 样例

```
UPDATE student SET a=新数据;
UPDATE student SET a=新数据 WHERE b=某个数据;
UPDATE student SET a=新数据 WHERE b=某个数据 AND c=某个数据;
UPDATE student SET a=新数据 WHERE b=某个数据 OR c=某个数据;
```

- 备注

- 为了方便表示，“基本格式”中没有对空格数量做严谨的表示。请注意：若 [C] 为空字符串，则 [B] 与 ; 之间没有空格（而不是如“基本格式”所示的一个空格）
- 若表 [A] 不存在、或 [B] 中指定的字段不存在、或 [C] 中指定的字段不存在，需要给出用户提示
- 若更新成功，需要给出用户提示

DELETE

- 基本格式

```
DELETE FROM [A] [B];
```

- 含义

从表 [A] 中删除所有满足条件 [B] 的行

- 占位符格式与含义

- [A] 可以为
 - 一个字符串，表示要更新的表的表名

- [B] 的定义与 SELECT 语句中 [D] 的定义完全相同，即：可以是空字符串，表示处理所有行；也可以是形如 WHERE ... 的筛选表达式，表示只处理满足条件的行
- 样例

```
DELETE FROM student;  
DELETE FROM student WHERE a=test;  
DELETE FROM student WHERE a=test AND b=test;  
DELETE FROM student WHERE a=test OR b=test;
```

- 备注
 - 若表 [A] 不存在、或 [B] 中指定的字段不存在，需要给出用户提示
 - 若删除成功，需要给出用户提示

ALTER (拓展功能)

- 基本格式

```
ALTER TABLE [A] [B] [C];
```

- 含义
修改表 [A] 的结构，添加/删除字段 [C]
- 占位符格式与含义
 - [A] 可以为
 - 一个字符串，表示要修改的表的表名
 - [B] 可以为
 - ADD，表示新增一个字段，该字段排在原有的最后一个字段之后；同时，将数据库中所有行在该列上的数值设置为空字符串，即 ""
 - DROP，表示删除一个字段
 - [C] 可以为
 - 一个字符串，表示要添加或删除的列名
- 样例

```
ALTER TABLE student ADD gender;  
ALTER TABLE student DROP birth_place;
```

- 备注
 - 若表 [A] 不存在，需要给出用户提示
 - 若 [B] 为 ADD，且表 [A] 中已存在名为 [C] 的字段，需要给出用户提示并拒绝修改
 - 若 [B] 为 DROP，且表 [A] 中不存在名为 [C] 的字段，需要给出用户提示
 - 若修改成功，需要给出用户提示