

实验三 实验 数据表的创建与修改管理实验

一、实验目的

- (1) 掌握表的基础知识。
- (2) 掌握使用 MySQL Workbench、PowerDesigner 或 SQL 语句创建表的方法。
- (3) 掌握表的修改、查看、删除等基本操作方法。
- (4) 掌握表中完整性约束的定义。
- (5) 掌握完整性约束的作用

二、实验内容

- 1、在 StudentInfo 数据库中创建 student 表和 grade 表。

(1) 表结构要求如下：

Student 表的内容

| 字段名 | 字段描述 | 数据类型 | 主键 | 非空 | 唯一 | 自增 |
|----------|------|-------------|----|----|----|----|
| num | 学号 | INT(10) | 是 | 是 | 是 | 否 |
| name | 姓名 | VARCHAR(20) | 否 | 是 | 否 | 否 |
| sex | 性别 | VARCHAR(4) | 否 | 是 | 否 | 否 |
| birthday | 出生日期 | DATETIME | 否 | 否 | 否 | 否 |
| address | 家庭住址 | VARCHAR(50) | 否 | 否 | 否 | 否 |

Grade 表的内容

| 字段名 | 字段描述 | 数据类型 | 主键 | 非空 | 唯一 | 自增 |
|--------|------|-------------|----|----|----|----|
| id | 编号 | INT(10) | 是 | 是 | 是 | 否 |
| Course | 课程名 | VARCHAR(10) | 否 | 是 | 否 | 否 |
| S_num | 学号 | INT(10) | 否 | 是 | 否 | 否 |
| grade | 成绩 | VARCHAR(4) | 否 | 否 | 否 | 否 |

(2)、相关操作

表创建成功后，查看两个表的结构。然后按照下列要求进行表操作，写出相关的命令行：

1. 将 grade 表的 course 字段的数据类型改为 VARCHAR(20)
2. 将 s_num 字段的位置改到 course 字段的前面。
3. 将 grade 字段改名为 score。
- 4.
5. 将 grade 表的存储引擎更改为 MyISAM 类型。
6. 将 student 表的 address 字段删除。
7. 在 student 表中增加名为 phone 的字段。
8. 将 grade 表改名为 gradeInfo。
9. 删除 student 表。

2、教材 P83 设计用户表

三、实验过程及结果

操作 1：在创建 student 表的时候，第一步需要选择数据库，本例中选取的是 mydb 数据库。再该数据库里面创建表格的操作时，需要使用图 1 所示的 CREATE TABLE 命令。其次定义表的各个内容，上例中包含有学号、姓名、性别、出生日期、家庭住址等五个内容，因此在建立表格的过程中，一共有五个字段建立的过程。在建立字段的时候，每一行命令都需要定义该字段的若干属性，在本例中该属性包括：是否为主键、是否非空、是否唯一、是否自增四大属性，分别通过控制该属性的关键字给出定义。创建 student 表的操作如下图 1 所示：

```
mysql> use mydb;
Database changed
mysql> CREATE TABLE student(
  -> num INT(10) NOT NULL UNIQUE PRIMARY KEY COMMENT '学号',
  -> name VARCHAR(20) NOT NULL COMMENT '姓名',
  -> sex VARCHAR(4) NOT NULL COMMENT '性别',
  -> birthday DATETIME COMMENT '出生日期',
  -> address VARCHAR(50) COMMENT '家庭住址'
  -> );
Query OK, 0 rows affected (0.02 sec)
```

图 1 创建 student 表操作

结果 1: 利用 CREATE TABLE 命令建立 student 表的结果如下图 2 所示。

```
mysql> desc student;
```

| Field | Type | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| num | int(10) | NO | PRI | NULL | |
| name | varchar(20) | NO | | NULL | |
| sex | varchar(4) | NO | | NULL | |
| birthday | datetime | YES | | NULL | |
| address | varchar(50) | YES | | NULL | |

```
5 rows in set (0.00 sec)
```

图 2 student 表的建立结果

注意: 利用 desc 命令能够查看表格的属及其所有字段和它的各个参数。这里不能用 show 命令，因为此时他还是空表。

操作 2: 在创建 gradet 表的时候，由于前面选择了 mydb 数据库，此时还是在 mydb 数据库里面的操作，因此不需要再切换数据库了。再该数据库里面创建表格的操作时，需要使用图 3 所示的 CREATE TABLE 命令。其次定义表的各个内容，上例中包含有编号、课程名、学号、成绩等四个内容，因此在建立 grade 表格的过程中，一共有四个字段建立的过程。在建立字段的时候，每一行命令都需要定义该字段的若干属性，在本例中该属性包括：是否为主键、是否非空、是否唯一、是否自增四大属性，分别通过控制该属性的关键字给出定义。创建的 grade 表操作如下图 3 所示。

```
mysql> CREATE TABLE grade(
  -> id INT(10) NOT NULL UNIQUE PRIMARY KEY COMMENT '编号',
  -> Course VARCHAR(20) NOT NULL COMMENT '课程名',
  -> S_num INT(10) NOT NULL COMMENT '学号',
  -> grade VARCHAR(4) COMMENT '成绩'
  -> );
Query OK, 0 rows affected (0.02 sec)
```

图 3 创建 grade 表操作

结果 2: 利用 CREATE TABLE 命令建立 grade 表的结果如下图 4 所示。

```
mysql> desc grade;
```

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| id | int(10) | NO | PRI | NULL | |
| Course | varchar(20) | NO | | NULL | |
| S_num | int(10) | NO | | NULL | |
| grade | varchar(4) | YES | | NULL | |

```
4 rows in set (0.00 sec)
```

图 4 student 表的建立结果

操作 3: 对表的属性字段进行修改操作一般利用 ALTER TABLE 命令和 MODIFY 关键字来实现,一般的格式为: ALTER TABLE +表名+ MODIFY + 字段名 + 字段数据类型。将 grade 表的 course 字段的数据类型改为 VARCHAR(20)结果如下图 5 所示。

```
mysql> ALTER TABLE grade
-> MODIFY Course varchar(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 5 修改字段类型操作

结果 3: 使用 desc 命令显示 grade 表格如下图 6 所示。

```
mysql> desc grade;
```

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| id | int(10) | NO | PRI | NULL | |
| Course | varchar(20) | NO | | NULL | |
| S_num | int(10) | NO | | NULL | |
| grade | varchar(4) | YES | | NULL | |

```
4 rows in set (0.00 sec)
```

图 6 修改 course 字段类型的结果

操作 4: 对表的属性字段进行移动操作一般利用 ALTER TABLE 和命令和 MODIFY 和 AFTER 关键字来实现,一般的格式为: ALTER TABLE +表名+ MODIFY + 目标字段移动位置的字段名 + 字段数据类型+目标字段名。将 s_num 字段的位置改到 course 字段的前面操作如下图 7 所示。

```
mysql> ALTER TABLE grade MODIFY Course VARCHAR(10) AFTER S_num;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 7 S_num 前置操作图

结果 4: 使用 desc 命令显示 grade 表格如下图 8 所示。

```
mysql> desc grade;
```

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| id | int(10) | NO | PRI | NULL | |
| S_num | int(10) | NO | | NULL | |
| Course | varchar(20) | YES | | NULL | |
| grade | varchar(4) | YES | | NULL | |

```
4 rows in set (0.00 sec)
```

图 8 S_num 前置结果图

操作 5: 对表的属性字段进行改名操作一般利用 ALTER TABLE 和命令和 CHANGE 关键字来实现,一般的格式为: ALTER TABLE +表名+ 目标字段名 +目标字段要修改的字段名 + 目标字段要修改的字段名的数据类型。将 grade 字段改名为 score 的操作如下图 9 所示。

```
mysql> Alter table grade change grade score varchar(4);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 9 将 grade 改名为 score

结果 5: 使用 desc 命令显示 grade 表格如下图 10 所示。

```
mysql> desc grade;
```

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| id | int(10) | NO | PRI | NULL | |
| S_num | int(10) | NO | | NULL | |
| Course | varchar(20) | YES | | NULL | |
| score | varchar(4) | YES | | NULL | |

4 rows in set (0.00 sec)

图 10 grade 改名为 score 结果图

操作 6: 对表的存储引擎类型更改操作一般利用 ALTER TABLE 和命令和 engine=关键字来实现, 一般的格式为: ALTER TABLE +表名+engine=MyISAM。将 grade 表的存储引擎更改为 MyISAM 类型的操作如下图 11 所示。

```
mysql> alter table grade engine=MyISAM;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 11 grade 表的存储引擎更改为 MyISAM 类型操作图

结果 6: 使用 SHOW TABLE STATUS LIKE 'grade'命令显示 grade 表格如下图 12 所示

```
mysql> SHOW TABLE STATUS LIKE 'grade';
```

| Name | Engine | Version | Row Format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length | Data_free | Auto_increment | Create_time | Update_time | Checksum |
|-------|--------|---------|------------|------|----------------|-------------|-----------------|--------------|-----------|----------------|---------------------|---------------------|----------|
| grade | MyISAM | 10 | Dynamic | 0 | 0 | 0 | 281474976710655 | 1024 | 0 | NULL | 2022-11-06 21:28:58 | 2022-11-06 21:28:58 | NULL |

1 row in set (0.00 sec)

图 12 使用 SHOW TABLE STATUS LIKE 'grade'命令显示 grade 表格

操作 7: 对表的属性字段进行删除操作一般利用 ALTER TABLE 和命令和 DROP 关键字来实现, 一般的格式为: ALTER TABLE +表名+ DROP +要删除的字段名。将 student 表的 address 字段删除的操作如下图 13 所示。

```
mysql> ALTER TABLE student DROP address;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 13 删除字段操作

结果 7: 使用 desc 命令显示 student 表格如下图 14 所示。

```
mysql> desc student;
```

| Field | Type | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| num | int(10) | NO | PRI | NULL | |
| name | varchar(20) | NO | | NULL | |
| sex | varchar(4) | NO | | NULL | |
| birthday | datetime | YES | | NULL | |

4 rows in set (0.00 sec)

图 14 删除字段结果

操作 8: 对表的属性字段进行增加操作一般利用 ALTER TABLE 和命令和 ADD 关键字来实现, 一般的格式为: ALTER TABLE +表名+ ADD +要增加的字段名+要增加的字段名的数据类型。在 student 表中增加名为 phone 的字段的操作如下图 15 所示。


```
mysql> alter table student add phone INT(10);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 15 增加字段操作

结果 8: 使用 desc 命令显示 student 表格如下图 16 所示。

```
mysql> desc student;
```

| Field | Type | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| num | int(10) | NO | PRI | NULL | |
| name | varchar(20) | NO | | NULL | |
| sex | varchar(4) | NO | | NULL | |
| birthday | datetime | YES | | NULL | |
| phone | int(10) | YES | | NULL | |

```
5 rows in set (0.00 sec)
```

图 16 增加字段结果

操作 9: 对表进行改名操作一般利用 ALTER TABLE 命令和 rename 关键字来实现，一般的格式为：ALTER TABLE + 表名 + rename + 修改后的表名。将 grade 表改名为 gradeInfo 的操作如下图 17 所示。

```
mysql> alter table grade rename to gradeInfo;
Query OK, 0 rows affected (0.00 sec)
```

图 17 grade 表改名操作

结果 9: 使用 desc 命令显示 student 表格如下图 18 所示。

```
mysql> desc grade;
ERROR 1146 (42S02): Table 'mydb.grade' doesn't exist
mysql> desc gradeInfo;
```

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| id | int(10) | NO | PRI | NULL | |
| S_num | int(10) | NO | | NULL | |
| Course | varchar(20) | YES | | NULL | |
| score | varchar(4) | YES | | NULL | |

```
4 rows in set (0.00 sec)
```

图 18 grade 表改名结果

操作 10: 对表进行删除操作一般利用 DROP 命令和 rename 关键字来实现，一般的格式为：DROP TABLE + 表名。删除 student 表的操作如下图 19 所示。

```
mysql> drop table student;
Query OK, 0 rows affected (0.01 sec)
```

图 19 删除 student 表格操作

结果 10: 由于 student 表格已经被删除，使用 desc 命令显示 student 表格会报错该表格不存在，报错结果如图 20 所示。使用 SHOW TABLES 显示所有的表格会发现 student 表格已经不存在了，结果如图 21 所示。

```
mysql> desc student;
ERROR 1146 (42S02): Table 'mydb.student' doesn't exist
```

图 20 显示 student 表格报错

```
mysql> SHOW TABLES;
+-----+
| Tables_in_mydb |
+-----+
| gradeinfo       |
+-----+
1 row in set (0.00 sec)
```

图 21 student 表格不存在

三、问题思考与总结

1、关于 NOT NULL

(1) 在定义基本表语句时，NOT NULL 参数的作用是什么？

(2) 主码列修改成允许 NULL 能否操作？为什么？

2、关于主码和唯一约束

(1) 唯一约束列是否允许 NULL 值？

(2) 一张表可以设置几个主码，可以设置几个唯一约束？

解答 1: 在定义基本表语句时，not null 意思是给字段不允许为空，是一个固定的语句表示给表创建的备注不允许为空。主码列不能修改成允许 NULL，因为主键是唯一标识且不能重复，如果为空，就无法识别。

解答 2: 唯一约束在内部使用唯一索引来保证列值的唯一性。在一个列上创建唯一约束之后，该列允许插入 null 值，通常情况下，两个 null 值是不相等的，但是在唯一约束中，唯一约束把两个 null 值看作是相同的，也就是说，null 和 null 是相等的，null 和其他任何非 null 值都不相等。一张表可以设置 1 个主键，可以设置多个唯一约束。

四、实验总结（遇到什么问题？如何解决的。请列举 1-5 个问题）

问题 1: 当我登录数据库时，显示无法启动，报错截图如下图 22 所示。

```
C:\WINDOWS\system32>net start mysql
MySQL 服务正在启动 .
MySQL 服务无法启动。

服务没有报告任何错误。

请键入 NET HELPMSG 3534 以获得更多的帮助。
```

图 22 启动数据库失败

解决 1: 发生这种情况的原因是综合性、多方面的，一般情况下，通过压缩的安装包，安装 MySQL 可能会出现 MySQL 无法启动的情况，缺乏 my.ini 文件和 data 文件夹或者 data 文件夹不是通过命令创建，而是通过手动创建或者复制别人的文件夹都会导致数据库启动失败。因此出现该错误没有固定的原因，解决方案一般是缺哪些文件就创建那些文件。但是我的文件夹里面什么都不缺，但还是报错，因此我断定原因就是里面参数配置混乱出错，因此我充值了数据库，通过命令：mysqld --initialize --console 就能解决该问题。

结果 1: 通过解决 1 中的解决方案，解决了该问题，解决后的截图如图 23 所示。

```
C:\WINDOWS\system32>mysqld --initialize --console
2022-11-06T11:33:22.189975Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2022-11-06T11:33:22.190025Z 0 [Warning] 'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and 'ERROR_FOR_DIVISION_BY_ZERO' sql modes should be used with strict mode. They will be merged with strict mode in a future release.
2022-11-06T11:33:22.192225Z 0 [ERROR] --initialize specified but the data directory has files in it. Aborting.
2022-11-06T11:33:22.192684Z 0 [ERROR] Aborting
```

图 23 解决启动数据库失败的问题

问题 2: 创建表后，我认为仍然可以使用 select 命令查看，但是这张表里面只有表头，没有元素，但是结果如下图 24 所示。

```
mysql> Select * from student;  
Empty set (0.00 sec)
```

图 24 用 select 查看表格

解决 2: 利用 Select 命令来查看表的前提下是表不能为空, 如果表为空的话就会报这是个空表, 不能被选中其中的元素。因此当表中元素为空时, 不能用 select 命令来选中元素名显示在界面里面。但是在 mysql 里面支持使用 DESC 命令来显示表中的字段机器它们的属性。

结果 2: 使用 DESC 命令的结果如下图 25 所示。

```
mysql> desc student;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| num   | int(10) | NO | PRI | NULL | |  
| name  | varchar(20) | NO | | NULL | |  
| sex   | varchar(4) | NO | | NULL | |  
| birthday | datetime | YES | | NULL | |  
| address | varchar(50) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

图 25 使用 DESC 命令能获取的表