第5章

# 表的基本操作

表是数据库中存储数据的基本单位,由一个或多个字段组成,每个字段需要有对应的数据类型。MySQL数据库中表的管理涉及表的作用、类型、构成、创建、删除和修改等。

本章先讲述了表的基本概念以及 MySQL 支持的数据类型和运算符等一些基础知识;接着讲述了表的基本操作,包括创建、查看、修改、复制和删除表;最后讲述了 MySQL 的约束控制,以及如何定义和修改字段的约束条件。

# 5.1 表的基本概念

数据库与表之间的关系:数据库是由各种数据表组成的;数据表是数据库中最重要的对象,是用来存储和操作数据的逻辑结构。表由列和行组成,列是表数据的描述,行是 表数据的实例。一个表包含若干个字段或记录。表的操作包括创建新表、修改表和删除 表等,这些操作都是数据库管理中最基本、最重要的操作。

5.1.1 建表原则

为减少数据输入错误,并能使数据库高效工作,设计表时应按照一定的原则对信息进行分类。同时为确保表结构设计的合理性,通常还要对表进行规范化设计,以消除表中存在的冗余。保证一个表只围绕一个主题,并使得表容易维护。

### 5.1.2 数据库表的信息存储分类原则

(1) 每个表应该只包含关于一个主题的信息。

当每个表只包含关于一个主题的信息时,就可以独立于其他主题来维护该主题的信息。例如,应将教师基本信息保存在"教师"表中。如果将这些基本信息保存在"授课"表中,则在删除某教师的授课信息时会将其基本信息一同删除。

(2) 表中不应包含重复信息。

表间也不应有重复信息,每条信息只保存在一个表中,需要时只在一处进行更新即可,这样效率更高。例如,每个学生的学号、姓名、性别等信息只保存在"学生"表中,而"成绩"中不再保存这些信息。

表(Table)是数据库中存储数据最常见和最简单的一种形式,数据库可以将复杂的数据结构用较为简单的二维表来表示。二维表是由行和列组成的,分别都包含着数据,如表 5-1 所示。

学号	姓名	性别	年龄
2020081601	光一飞	女	20
2020081602	李忠诚	男	18
2020081603	王晴水	女	21
2020081604	赵洋办	男	19

表 5-1 学生信息表

每个表都是由若干行和列组成的,在数据库中表中的行称为记录,表中的列则称为这 些记录的字段。

记录也被称为一行数据,是表中的一行。在关系型数据库的表中,一行数据是指一条 完整的记录。

字段是表中的一列,用于保存每条记录的特定信息。如表 5-1 所示的学生信息表中 的字段包括"学号""姓名""性别"和"年龄"。数据表的一列包含了某个特定字段的全部 信息。

对于表的基本操作主要包括创建表、查看表、修改表以及删除表等,这些操作非常重要,它们是数据库操作的基础。

### 5.2 创建数据表

创建数据表指的是在已存在的数据库中建立新表。MySQL 既可以根据开发需求创 建新表,也可以根据已有的表复制相同的表结构。其中依据已有的表创建相同结构的新 表方式会在后面的章节中讲解,此处仅讲解如何根据需求创建一个简单的新表。

在创建表之前,首先要选择在哪个数据库中创建表。在 MySQL 自带的客户端软件中,可以使用如下 SQL 语句来选择数据库:

USE db name;

其中"db\_name"为选择的数据库名称。如果没有选择数据库而是直接创建表,则会 提示"No database selected"错误。如果选择数据库成功,则会提示"Database changed", 之后就可以在选择的数据库中创建新表了。操作数据表时,也可以不使用"USE 数据库" 的方式选择数据库,而直接将表名的位置改为"数据库.表名"的形式,就可以在任何数据 库下访问其他数据库中的表。

如果使用 Navicat 软件,操作可以大大简化,直接双击要选择的数据库即可切换数据 库(如果是第一次选中该数据库,那么数据库图标会由灰色变为绿色,下次再选择该数据 库时只需要单击即可)。

本节将分别讲解在 Navicat 中如何使用 SQL 语句创建表以及如何使用图形界面创 建表。

### 5.2.1 使用 SQL 语句创建表

在 Navicat 中,要使用 SQL 语句创建表,必须在选中的数据库中打开一个 SQL 语句 执行窗口(在此之前新建一个名为 jxgl 的数据库),具体操作如图 5-1 所示。

🕙 Navicat Premium			- <b>X</b>
文件 编辑 查看 收藏夹	工具 窗口 帮助		登录
💰 . 🖥 🔳	$f_{(x)}$	<u>2</u> <u>#</u>	. <b>.</b> *
连接 新建查询 表	视图 函数	用户 其它	查询
🔺 📉 DBconn	对象 邮*无标题	- 查询	() m
■ information_schema ▲ ■ jxgl	E 打开表 IP 设计表 E	新建表 🕺 🤉 🤇	
田表			0表
▷ EB 视图 ▷ f <sub>z</sub> 函数			
■ 查询			字符集
▷ 🧕 备份			utf8
🛢 mysql			排序规则
performance_schema			utf8_general_ci
Sys Sys			-
🛢 test			
0表	🔪 DBconn 🚦	jxgl	

图 5-1 在 jxgl 数据库中创建 SQL 语句执行窗口

按照图 5-1 操作完成后,会看到如图 5-2 所示的界面,如果界面中数据库一栏中显示为 jxgl,则表示操作成功(此处单击下拉按钮"▼",可以切换数据库)。



图 5-2 选择数据库成功界面

接下来就可以在 SQL 语句执行窗口中输入创建表的 SQL 语句,其语法格式如下。

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] 表名

字段名 1,数据类型[完整性约束条件], 字段名 2,数据类型[完整性约束条件],

(

```
字段名 n, 数据类型[完整性约束条件]
```

)

注意:语法格式中的"[]"表示可选的。

在上述语法格式中,CREATE TABLE 为创建表的固定语法格式。

可选项 TEMPORARY 表示临时表,临时表指的是一种仅在当前会话中可见,并在 当前会话关闭时自动删除的数据表,它主要用于临时存储数据。临时表的语法很简单,只 需在 CREATE 与 TABLE 关键字中间添加 TEMPORARY 即可。创建临时表时指定的 数据库可以是 MySQL 服务器中存在的数据库,也可以是不存在的数据库。若数据库不 存在,则操作临时表时必须使用"数据库.临时表名"指定临时表所在的数据库。除此之 外,临时表中数据的操作与普通表相同,都可以进行 SELECT、INSERT、UPDATE 和 DELETE 操作,这里不再演示。需要注意的是,SHOW TABLES 不能查看指定数据库下 有哪些临时表,并且临时表的表名必须使用 ALTER TABLE 修改,而不能使用 RENAME TABLE…TO 修改。

"表名"指的是创建的数据表名称(为了避免数据表重复,通常为数据表添加前缀用于 区分不同的项目。前缀一般选取数据库的前几个字母,并添加一个下画线,比如 jxgl\_ student)。表名不能与数据库的关键字同名,如 CREATE、DATABASE、TABLE 等。

"字段名"指的是数据表的列名,为二维表中每一列的列名。不同字段之间的定义使用","隔开,但最后一个字段没有","。

"数据类型"为该字段所存储的数据的数据类型。

"完整性约束条件"指的是字段的某些特殊约束条件,关于表的约束,将在后续章节进 行详细讲解。

注意:在 MySQL 中,如果使用的数据库名、表名或字段名等与保留字冲突,需使用 撇号(')括起来。在 MySQL 自动生成的代码中,表名或字段名等全部使用撇号括起来。

以创建名为 student 的表为例进行演示,该表用来存储学生信息,具体信息及数据类型的选择如表 5-2 所示,创建 student 表的 SQL 语句如例 5-1 所示。

字段名	数 据 类 型	描述
sno	VARCHAR(10)	学号,主键
sname	VARCHAR(20)	姓名,非空
ssex	ENUM	'男', '女',默认为'男'
sbirth	DATE	出生日期
zno	CHAR(4)	专业号
sclass	VARCHAR(10)	所在班级

表 5-2 student 表相关信息



#### 【例 5-1】 使用 SQL 语句创建 student 表。

```
CREATE TABLE student (
sno VARCHAR(10) NOT NULL COMMENT '学号',
sname VARCHAR(20) NOT NULL COMMENT '姓名',
ssex ENUM('男', '女') NOT NULL DEFAULT '男' COMMENT '性别',
sbirth DATE NOT NULL COMMENT '出生日期',
zno VARCHAR(4) NULL COMMENT '古生号',
sclass VARCHAR(10) NULL COMMENT '班级',
PRIMARY KEY ('sno')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8 bin;
```

执行结果如图 5-3 所示。



图 5-3 使用 SQL 语句创建 student 表

从图 5-3 中可以看到"OK"字样,说明表格已经创建成功。"时间: 0.045s"表示执行 该 SQL 语句所耗费的时间。

在操作数据表时,可以不使用"USE 数据库"的方式选择数据库,而是直接将表名的 位置改为"数据库.表名"的形式,这样就可以在任何数据库下访问其他数据库中的表。创 建数据库表时,还可以设置表的存储引擎、默认字符集以及压缩类型。

(1)向 CREATE TABLE 语句末尾添加 ENGINE 选项,即设置该表的存储引擎。语法格式如下:

ENGINE=存储引擎类型

(2)向 CREATE TABLE 语句末尾添加 DEFAULT CHARSET 选项,即设置该表的字符集。语法格式如下:

DEFAULT CHARSET=字符集类型

(3)如果希望压缩索引中的关键字,使索引关键字占用更少的存储空间,可以通过设置 pack\_keys 选项实现(注意,该选项仅对 MyISAM 存储引擎的表有效)。语法格式如下:

Pack\_keys=压缩类型

注意:对于 InnoDB 存储引擎的表而言, MySQL 服务实例会在数据库目录

StudentInfo 中自动创建一个名为表名、后缀名为 frm 的表结构定义文件 Student.frm。 frm 文件记录了 Student 表的表结构定义。如果数据库表的存储引擎是 MyISAM,则 MySQL 服务实例除了会自动创建 frm 表结构定义文件外,还会自动创建一个文件名为 表名、后缀名为 MYD(即 MYData 的简写)的数据文件以及一个文件名为表名、后缀名为 MYI(即 MYIndex 的简写)的索引文件,其中,MYD 文件用于存放数据,MYI 文件用于存 放索引。

### 5.2.2 使用图形界面创建表

除了使用 SQL 语句创建表外,还可以在图形界面中创建表,这种操作对于初学者而 言更为简单。

同样以创建 student 表为例。

(1)展开 jxgl 数据库的目录结构,其中有一个名为"表"的选项,右击"表",在弹出的下拉列表中选择 "新建表",如图 5-4 所示。

(2)在打开的创建表窗口中按照要求输入字段 名、数据类型和数据长度等信息,其中数据类型可在 下拉列表中根据需求选择。输入完成后的表信息如 图 5-5 所示。

从图 5-5 中能够看到位于窗口上方的工具栏,其 中"添加字段"工具提供增加字段功能,"插入字段"工 具提供插入字段功能(在洗中字段的上方插入),"删 e performanc 新建组 e sys 粘贴 e test 刷新

新建表

导入向导...

导出向导..

运行 SOL 文件...

在数据库中查找

对象

12 打开表

#student

4 🔝 DBconn

🖌 🛢 jxgl

▷囲表

思 视图

▷ *f<sub>x</sub>* 函数

mysql

information\_schema

图 5-4 jxgl 数据库的目录结构

除字段"则用于删除选中的字段,"主键"能够将选中的字段设置为主键,"上移"和"下移" 可以上下移动选中的字段。

冒保存 💿 添加字段 📪 插入字段 📪 删除字段 👂 主键 ↑ 上移 \vee 下移											
字段 索引 外键 触发器 选项 注释 SQL 预览											
名	类	型		长度	Ē	小数点		不是 null	虚拟	键	注释
sno	va	rchar		10		0		$\checkmark$		<b>P</b> 1	学号
sname	e va	rchar		20		0		$\checkmark$			姓名
ssex	er	um						$\checkmark$			性别
sbirth	da	te		0		0		$\checkmark$			出生日期
zno	va	rchar		4		0					专业号
C sclass	va	rchar		10		0					班级

图 5-5 使用图形界面输入表信息

(3) 表信息输入完成后,单击左上方的"保存"按钮进行保存,之后便会弹出要求输入 表名的对话框,在输入框中输入 student\_1,单击"确定"按钮即完成创建表的操作,具体操 作如图 5-6 所示。

(4) 右击以刷新"表",则会在该节点下显示刚刚创建的 student\_1 表。

86



🖹 保存	To X	添加字段	■◆插入	字段 🝺	删除字题	役	₽主物	建	↑ 上移	♦ 下移
字段	索引	外键	触发器	选项	注释	SQL	预览			
名	类	塑 🗄	度	小数点	不是	null	虚拟	键	注釈	₽ ₽
表名									X	
输入	∖表名	1								
sti	udent_1	-								
			确?	=	取消					
			HHA		40/H					
		_	_	_	_	_	_	_	_	

图 5-6 保存创建的表

# 5.3 查看数据表

在表创建完成后,在很多情况下都会想要查看表的信息,如在插入数据之前查看数据 的类型和长度,或者查看主键和外键的设置等。本节讲述如何使用 SQL 语句来查看表的 基本结构和详细信息(利用图形界面查看表结构非常简单,读者可以自行研究一下)。

#### 5.3.1 查看所有数据表名

MySQL 中提供了专门的 SQL 语句,用于查看某数据库中存在的所有数据表、指定 模式的数据表或数据表的相关信息。

选择数据库后,可以使用 SHOW TABLES 语句来显示指定数据库中存放的所有表 名。语法格式如下:

SHOW TABLES [LIKE 匹配模式];

其中,如果省略可选项,则表示查看当前数据库中的所有数据表。如果添加可选项,则按照"匹配模式"查看数据表。

匹配模式符"%"匹配一个或多个字符,代表任意长度的字符串;匹配模式符"\_"则仅可以匹配一个字符。LIKE后的匹配模式必须使用单引号或双引号括起来。

【例 5-2】 显示数据库 jxgl 中所有的表。

SHOW TABLES;

执行结果如图 5-7 所示。

DB	conn 👻 🛢 jxgl 👻
1	SHOW TABLES;
信息	结果1 剖析 状态
Table	es_in_jxgl
▶ stude	ent
stude	ent_1

图 5-7 显示数据库 jxgl 中所有的表

### 5.3.2 查看数据表的基本结构

MySQL 提供的 DESCRIBE 语句可以查看数据表中所有字段或指定字段的信息,包括字段名和字段类型等。其中,DESCRIBE 命令可以简写成 DESC。

语法格式1:查看所有字段的信息。

DESCRIBE | DESC 数据表名;

语法格式 2: 查看指定字段的信息。

DESCRIBE | DESC 数据表名 字段名;

【例 5-3】 用 DESCRIBE 和 DESC 命令显示 jxgl 数据库中 student 表的结构。

DESCRIBE student;

或者:

DESC student;

执行结果如图 5-8 所示。

Field	Туре	Null	Key	Default	Extra
sno	varchar(10)	NO	PRI	(Null)	
sname	varchar(20)	NO		(Null)	
ssex	enum('男','女')	NO		男	
sbirth	date	NO		(Null)	
zno	varchar(4)	YES		(Null)	
sclass	varchar(20)	YES		(Null)	

#### 图 5-8 student 表结构图

通过 DESCRIBE 语句能够看到表的字段(Field)、数据类型及长度(Type)、是否允许 空值(Null)、键的设置信息(Key)、默认值(Default)以及附加信息(Extra)。但是如果想 查看关于表更详细的信息,这种方式就不可行了,需要使用一种新的 SQL 语句,如下:

SHOW CREATE TABLE;

#### 5.3.3 查看数据表的详细结构

使用 SHOW CREATE TABLE 语句,不仅可以查看表的字段、数据类型及长度、是 否允许空值、键的设置信息和默认值等,还可以查看数据库的存储引擎以及字符集等信 息。其语法格式如下:

SHOW CREATE TABLE 表名;

其中 SHOW CREATE TABLE 为查看表详细结构的固定语法格式,"表名"为要查 看的表的名称。查看 student 表详细结构的 SQL 语句如例 5-4 所示。

【例 5-4】 使用 SHOW CREATE TABLE 语句查看表的详细结构。

第5章 表的基本操作



SHOW CREATE TABLE student;

执行结果如图 5-9 所示。

信息结	果1	剖析	状态
Table	С	reate Ta	able
student	C	REATE	TABLE 'student'( 'sno' varchar(10) COLLATE utf8_bin NOT NULL C

图 5-9 查看 student 表的详细结构

由于页面显示问题,图中 CREATE TABLE 中的表信息并不能完全展现出来,为了 让大家能够清晰地看到表的详细结构,此处将信息复制并整理了一份,如下所示:

CREATE TABLE student (

```
sno VARCHAR(10) COLLATE utf8_bin NOT NULL COMMENT '学号',
sname VARCHAR(20) COLLATE utf8_bin NOT NULL COMMENT '姓名',
ssex ENUM('男','女') COLLATE utf8_bin NOT NULL DEFAULT '男' COMMENT '性别',
sbirth date NOT NULL COMMENT '出生日期',
zno VARCHAR(4) COLLATE utf8_bin DEFAULT NULL COMMENT '专业号',
sclass VARCHAR(20) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT
'班级',
PRIMARY KEY (sno)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8 bin
```

### 5.3.4 查看数据表结构

MySQL 数据库中的 SHOW COLUMNS 语句也可以查看表结构,基本语法格式如下:

```
#语法格式 1
SHOW [FULL] COLUMNS FROM 数据表名 [FROM 数据库名];
#语法格式 2
SHOW [FULL] COLUMNS FROM 数据库名.数据表名;
```

在上述语法格式中,可选项 FULL 表示显示详细内容,在不添加的情况下查询结果与 DESC 的结果相同;在添加 FULL 选项时,此语句不仅可以查看到 DESC 语句查看的信息,还可以查看到字段的权限以及 COMMENT 字段的注释信息等。在 SQL 语句中可以通过"FROM 数据库名"或"数据库名.数据表名"的方式查看任意数据库下的数据表结构信息。

【例 5-5】 查看 studentinfo 数据表结构的详细信息。

SHOW FULL COLUMNS FROM studentinfo;

执行结果如图 5-10 所示。

从上述执行结果可以看出,SHOW FULL COLUMNS 语句除了查询出与 DESC 语句相同的字段外,还会查询出 Collation(校对集)、Privileges(权限)和 Comment(注释)字段。

信	息纪	課1 剖析 为	态						
	Field	Туре	Collation	Null	Key	Default	Extra	Privileges	Comment
•	sno	varchar(10)	utf8_bin	NO	PRI	(Null)		select,insert,update,	学号
	sname	varchar(20)	utf8_bin	NO		(Null)		select,insert,update,	姓名
	ssex	enum('男','女')	utf8_bin	NO		男		select, insert, update,	性别
	sbirth	date	(Null)	NO		(Null)		select,insert,update,	出生日期
	zno	varchar(4)	utf8_bin	YES		(Null)		select,insert,update,	<b></b>
	sclass	varchar(20)	utf8_bin	YES		(Null)		select, insert, update,	班级

图 5-10 查看数据表结构

# 5.4 修改数据表

修改表是指修改数据库中已经存在的数据表的结构。表在创建完成后,可能会因为 某些原因需要对表的名称、字段名称、字段的数据类型或者字段的排列位置等进行修改。 有一种方法就是直接删除旧表,然后根据新的需求创建新表,但是如果旧表中已经存在大 量数据,那会增加额外的工作量;另外一种方法则是使用 MySQL 中提供的 ALTER TABLE 语句来修改表的相关定义,即本节要讲解的内容。

常用的修改表的操作有:修改表名、修改字段数据类型或字段名、增加和删除字段、 修改字段的排列位置、更改表的存储引擎以及删除表的外键约束等。在修改表结构时,可 以使用 DESC 命令查看修改是否成功。

在使用 Navicat 软件修改表时,要注意以下两点。

(1)如果要修改表名,只需要右击要修改的表格,然后在弹出的下拉列表中选择"重命名"选项即可。

(2)如果要修改字段的相关定义,则需右击要修改的表格,然后在弹出的下拉列表中选择"设计表"选项,进入修改表格界面,修改完成后保存即可。

5.4.1 修改表名

表名是用来区分同一个数据库中不同表的依据,因此在同一个数据库表名具有唯一性。通过 SQL 语句 ALTER TABLE 可以修改表名,其语法格式如下:

格式一:

ALTER TABLE 旧表名 RENAME [TO|AS] 新表名;

格式二:

RENAME TABLE 旧表名 1 TO 新表名 1[,旧表名 2 TO 新表名 2]…;

其中用 ALTER TABLE 修改数据表名称时,可以直接使用 RENAME,或在其后添加 TO 或 AS。而 RENAME TABLE 则必须使用 TO,另外此语法可以同时修改多个数据表的名称。

【例 5-6】 使用 ALTER TABLE 语句修改表名。