

学生管理数据库数据的操作

任务描述

(1)虽然表已创建好,但它是空的,那么如何往表里添加数据?添加数据之后,又如何进行管理?

- (2) 在向表中操纵数据时,如何避免输入无效的数据?
- (3) 数据完整性有哪些类型? 它们分别可以避免哪些类型的无效数据?
- (4) 如何在实际操作时实现数据完整性?

- (1) 掌握: 表数据的添加,数据完整性的实现。
- (2) 理解:数据完整性的概念和类型,约束的类型。

知识准备



5.1 数据完整性概述

数据库中的数据是从外界输入的,在向数据库中添加、修改和删除数据时,难免会因手 工输入而产生各种错误。如何保证和维护数据的正确性、一致性和可靠性,成为数据库系统 关注的问题。利用约束、默认和规则来维护数据的完整性,可以避免大部分无效数据的 产生。

5.1.1 数据完整性的概念

数据完整性用于保证数据库中数据的正确性、一致性和可靠性,防止数据库中存在不符 合语义规定的数据以及因错误信息的输入输出导致无效操作或产生错误信息。

5.1.2 数据完整性的类型

1. 实体完整性

实体完整性又称为行完整性,规定表的每一行在表中是唯一的实体。实体完整性通过 索引、PRIMARY KEY 约束、UNIQUE 约束或 IDENTITY 属性实现。如 student 表中 sno (学号)的取值必须唯一,它唯一标识了相应记录所代表的学生。学生的姓名不能作为主键, 因为完全可能存在两名学生同名同姓的情况。

2. 域完整性

域完整性又称为列完整性,保证指定列的数据具有正确的数据类型、格式和有效的数据 范围。域完整性通过 FOREIGN KEY 约束、CHECK 约束、DEFAULT 约束、NOT NULL 定义和规则实现。如学生的考试成绩必须为 0~100,性别只能是"男"或"女"。

3. 参照完整性

参照完整性又称为引用完整性,是指两个表的主键和外键的数据应对应一致。它确保 了有主关键字的表中对应其他表的外关键字的行存在,即保证了表之间的数据的一致性。 参照完整性是建立在外关键字和主关键字之间或外关键字和唯一性关键字之间的关系上 的,包含外关键字的表称为从表,被从表引用或参照的表称为主表。参照完整性的作用体现 在几个方面:若主表中无关联的记录时,则不能将记录添加或更改到相关表中;若可能导 致相关表中生成孤立记录时,则不能更改主表中的该值;若存在与某记录匹配的相关记录 时,则不能从主表中删除该记录。例如学生学习课程的课程号必须是有效的课程号,score (成绩)表的外键 cno(课程号)将参考 course(课程)表中主键 cno(课程号)以实现数据完 整性。

5.2 实现约束

约束是强制数据完整性的首选方法。约束是通过限制列中数据、行中数据以及表之间 数据取值从而实现数据完整性的方法。定义约束可以在创建表时设置,也可以在修改表时 添加约束。

5.2.1 PRIMARY KEY(主键)约束

PRIMARY KEY 约束在表中定义一个主键,唯一地标识表中的行。一个表应有一个 PRIMARY KEY 约束,且只能有一个 PRIMARY KEY 约束。PRIMARY KEY 约束中的 列不能接受空值和重复值。

若已有 PRIMARY KEY 约束,要将新列作为主键,则必须先删除现有的 PRIMARY KEY 约束,然后再创建新的主键;当 PRIMARY KEY 约束由另一个表的 FOREIGN KEY 约束引用时,不能删除被引用的 PRIMARY KEY 约束,要删除它,必须先删除引用的 FOREIGN KEY 约束。主键可以是一列,也可以是多列组合的复合主键。

5.2.2 DEFAULT(默认值)约束

DEFAULT 约束是在用户未提供某些列的数据时,数据库系统为用户提供的默认值。

表的每一列都可以包含一个 DEFAULT 定义。可以修改或删除现有的 DEFAULT 定义,但必须先删除已有的 DEFAULT 定义,然后通过新定义重新创建。默认值必须与 DEFAULT 定义适用的列的数据类型一致,每一列只能定义一个默认值。

5.2.3 CHECK 约束

CHECK 约束是限制用户输入某一列的数据取值,即该列只能输入一定范围的数据。



也就是只有符合 CHECK 约束条件的数据才能输入。CHECK 约束可以作为表定义的一部 分在创建表时创建,也可以添加到现有表中。在一个表中可以创建多个 CHECK 约束,在 一列上也可以创建多个 CHECK 约束。

5.2.4 UNIQUE 约束

由于一个表只能定义一个主键,而在实际应用中,表中可能有多列的值需要是唯一的,可以使用 UNIQUE 约束确保在非主键列中不输入重复值。要强制一列或多列组合(不是 主键)的唯一性时应使用 UNIQUE 约束。与 PRIMARY KEY 约束不同的是,一个表可以 定义多个 UNIQUE 约束,允许列为空值,但空值只能出现一次。

5.2.5 NOT NULL 约束

在设计表时,表中的列可以定义为允许或不允许空值。如果允许某列可以不输入数据,则该列定义为 NULL 约束;如果某列必须输入数据,则该列定义为 NOT NULL 约束。默认情况下,列允许为 NULL。NULL 通常表示值未知或未定义。NULL 不同于零、空白或长度为零的字符串,NULL 表示用户还没有为该列输入值。

5.2.6 FOREIGN KEY 约束

FOREIGN KEY 约束用于强制实现参照完整性,保证了数据库中表数据的一致性和 正确性。FOREIGN KEY 约束可以规定表中的某列参照同一个表或另外一个表中已有 的 PRIMARY KEY 约束或 UNIQUE 约束的列。FOREIGN KEY 约束可以在创建表时创 建,也可以向现有表添加 FOREIGN KEY 约束。一个表可以有多个 FOREIGN KEY 约束。

任务实施

5.3 使用图形化管理工具操作学生管理数据库表数据

5.3.1 插入学生管理数据库表数据



【例 5-1】 假设学生管理数据库 studb 存在,其中的 student 表中的结构如图 5-1 所示, 先建立 student 表,使用 Navicat 图形化管理工具向 student 表中插入如表 5-1 所示的数据 (假设 student 表已创建好)。

	名	类型	长度	小数点	不是 null	
•	stuid	char	12	0		P1
	stuname	char	10	0		
	sex	char	2	0		
	age	int	2	0		
	major	char	50	0		

图 5-1 student 表结构

stuid	stuname	sex	age	major
2210001	张白	男	19	计算机网络技术
2210002	李小红	女	18	软件技术
2210003	王文	用	19	计算机应用技术

表 5-1 student 表中插入的数据

操作步骤如下。

(1) 启动 Navicat 图形化管理工具,连接 mytest 服务器,双击 studb 数据库,使其处于 打开状态,在 studb 数据库下单击"表"节点,在对象窗口中选中 student 表,如图 5-2 所示。



图 5-2 选中 student 表

(2) 单击工具栏中的"打开表"按钮,打开编辑窗口,将表 5-1 里的数据添加到 student 表中,如图 5-3 所示。

No.	=	5. 开始事务	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	± • ▼筛	选 🛃 排序	□ 导入 □ 导出
	stuid 🔺	stuname	sex	age	major	
	2210001	张白	男	19	计算机网络技	床
	2210002	李小红	女	18	软件技术	
I	2210003	王文	男	19	计算机应用技	术

图 5-3 插入数据

(3) 当数据录入完毕后,单击状态栏中的"√"按钮完成保存。

5.3.2 删除学生管理数据库表数据

在使用过程中,表中的一些数据可能不再需要,这时可以将其删除。

【例 5-2】 使用 Navicat 图形化管理工具删除 student 表中学号为 2210003 同学的信息。 操作步骤如下。

(1) 启动 Navicat 图形化管理工具,连接 mytest 服务器,双击 studb 数据库,使其处于打

开状态,在 studb 数据库下单击"表"节点,在对象窗口中双击 student 表,打开 student 表。

(2) 定位学号为 2210003 的记录行,单击该行最前面的黑色箭头,选择该行后,右击,在 弹出的快捷菜单中选择"删除记录"命令,如图 5-4 所示。

stui	d 🔺	stuname	sex	age	major	
221	0001	张白	男	19	计算机网络技术	
221	0002	李小红	女	18	软件技术	
221	0003	Ŧ文	奥	19	计算机应用技术	
	设置设置	为空白字符串 为 NULL				
囁	复制复制	为	•			
i.	粘贴					
	記録	记录				
~	排序		•			
	筛选		- >			
	移除	全部排序及简	选			
	显示		•			
	刷新					

图 5-4 选择"删除记录"命令

(3) 弹出一个"确认删除"对话框,单击"删除一条记录"按钮,删除所选行。

5.3.3 修改学生管理数据库表数据

【例 5-3】 使用 Navicat 图形化管理工具修改 student 表中学号为 2210002 同学的信息,将 age 修改为 19。

操作步骤如下。

(1) 启动 Navicat 图形化管理工具,连接 mytest 服务器,双击 studb 数据库,使其处于 打开状态,在 studb 数据库下单击"表"节点,在对象窗口中双击 student 表,打开 student 表。

(2) 直接在学号为 2210002 同学的 age 字段中修改,将 18 修改为 19,如图 5-5 所示。

1000	=	开始事务	日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日	· • ▼ 徐	选 计排序 📑 导入 📑 导出
	stuid 🔺	stuname	sex	age	major
	2210001	张白	男	19	计算机网络技术
1	2210002	李小红	女	19	软件技术

图 5-5 修改表中数据

(3) 单击状态栏中的"√"按钮完成保存。

5.4 使用语句操作学生管理数据库表数据

对表数据的操作除了使用 Navicat 图形化管理工具外,还可以使用 SQL 语句。

5.4.1 插入学生管理数据库表数据

1. 使用 INSERT 语句向表中插入数据

通过 INSERT 语句可以向表中添加一行或多行数据,语法格式如下:

INSERT INTO 表名 [(字段列表)] VALUES (值列表 1)[(值列表 2), …(值列表 n)];

语法说明如下。

(1) 表名:指定插入新数据的表的名称。

(2)字段列表:指定数据表的列名,必须用圆括号将字段列表括起来,当指定多个列时,各列之间用逗号隔开;当向表中的所有字段插入数据时,字段列表可以省略。

(3) 值列表:指定插入的新数据值。[(值列表 2),…(值列表 n)]为可选项,表示多条 记录对应的数据。每个值列表都必须用圆括号括起来,列表间用逗号分隔。

重要提示:

在插入数据时要注意:

- 数据值的数量和顺序必须与字段名列表中的数量和顺序一样。
- 数据值的数据类型必须与表的列中的数据类型匹配,否则插入失败。
- 数据值如果采用默认值则写 DEFAULT; 如果是空值则写 NULL。
- 插入数据类型如果是字符型、日期型,则必须用单引号。

1) 向表中的部分列插入数据

【例 5-4】 使用 SQL 语句向学生管理数据库 studb 的 student 表中插入一条新数据,其中 stuid 为 2210004, stuname 为"白林", sex 为"女"。



55

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
INSERT INTO student(stuid,stuname,sex)
VALUES('2210004','白林','女');
```

执行结果如图 5-6 所示。



图 5-6 添加一条数据的执行结果

2) 向表中的所有列插入数据。

【例 5-5】 使用 SQL 语句向学生管理数据库 studb 的 student 表中插入一条新数据,其中 stuid 为 2210005, stuname 为"杨辰", sex 为"男", age 为 19, major 为 NULL。

打开 MySQL 8.0 Command Line Client,输入以下语句:

INSERT INTO student VALUES('2210005','杨辰','男',19,NULL);

执行结果如图 5-7 所示。



图 5-7 向表中的所有列添加数据

3) 向表中插入多条数据

【例 5-6】 使用 SQL 语句向学生管理数据库 studb 的 student 表的部分列插入三条新数据。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
INSERT INTO student(stuid,stuname,sex,age)
VALUES('2210006','张鹏','男',19),
('2210007','刘刚','男',20),
('2210008','苏宏','女',18);
```

执行结果如图 5-8 所示。

mysq1> INSE	CRT INTO student(stuid, stuname, sex, age)
-> VALU	ES('2210006','张鹏','男',19),
->	('2210007','刘刚','男',20),
_>	('2210008','苏宏','女',18);
Query OK, 3	3 rows affected (0.01 sec)
Records: 3	Duplicates: 0 Warnings: 0

图 5-8 向表中部分列添加三条数据

2. 使用 INSERT···SELECT 语句插入数据

使用 INSERT ···· SELECT 语句可以将某一个表中的数据插入另一个新数据表中,语法格式如下:

```
INSERT INTO 目标数据表名 (字段列表 1)
SELECT 字段列表 2
FROM 源数据表名
WHERE 条件表达式;
```

语法说明如下。

(1) 目标数据表名:指定要插入的新表名称。

(2) SELECT: 用于检索数据。

(3) 字段列表 2: 要检索的列表。该列与 INSERT 中的字段列表 1 的数量和顺序必须 相同,列的数据类型和长度相同或者可以进行转换。

(4) 源数据表名:表的名称。该表必须是已存在的表。

(5)条件表达式:指定插入的数据应满足的条件。

【例 5-7】 使用 SQL 语句将 student 表中性别是"男"的同学记录插入 student_copy 表中。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
CREATE TABLE student_copy
(学号 char(12)NOT NULL,
姓名 char(10),
性別 char(2)
);
```

用 INSERT INTO 语句向 student_copy 表中插入数据:

```
INSERT INTO student_copy
SELECT stuid, stuname, sex
FROM student
WHERE sex = '男';
```

执行结果如图 5-9 所示。

mysql> CREATE TABLE student_copy	
-> (字号 char(12)NOT NULL,	
\rightarrow 姓名 char(10),	
-> 1 生别 char (2)	
Query OK, 0 rows affected (0.29 sec)	
mysql> INSERT INTO student_copy	
-> SELECT stuid, stuname, sex	
-> FROM student	
-> WHERE sex='男':	
Query OK. 4 rows affected (0.06 sec)	
Records: 4 Dunlicates: 0 Warnings:	0

图 5-9 创建表并添加数据

5.4.2 修改学生管理数据库表数据

在使用过程中,根据实际情况有时需要修改表中的数据,修改数据的语法格式如下:

UPDATE 表名
SET 字段名 1 = 值 1,字段名 2 = 值 2, ...,字段名 n = 值 n
[WHERE 条件表达式];

语法说明如下。

(1) UPDATE:修改数据的关键字。

(2) 表名:指定要修改数据的表名。

(3) SET 字段名 1=值 1: 指定要更新的列及该列的新值。

(4)条件表达式:指定被更新的记录应满足的条件。

【例 5-8】 使用 SQL 语句将 student 表中学号为 2210005 的 age 由 19 修改为 18。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
UPDATE student
SET age = 18
WHERE stuid = '2210005';
```

执行结果如图 5-10 所示。

【例 5-9】 使用 SQL 语句将 student 表中所有学生的 major 更新为"计算机网络技术"。 打开 MySQL 8.0 Command Line Client,输入以下语句:

UPDATE student SET major = '计算机网络技术';

执行结果如图 5-11 所示。





mysql> UPDATE student -> SET major='计算机网络技术' -> ; Query OK, 6 rows affected (0.01 sec) Rows matched: 7 Changed: 6 Warnings: 0

图 5-11 更新数据

5.4.3 删除学生管理数据库表数据

在 MvSQL 中, 使用 DELETE 语句可以删除表中的一行或多行数据, 或者使用 TRUNCATE 语句删除表中的所有数据。

1. 使用 DELETE 语句删除数据

语法格式如下:

58

DELETE FROM 表名 [WHERE 条件表达式];

语法说明如下。

(1) 表名: 删除数据的表的名称。

(2) 条件表达式: 指定被删除的记录应满足的条件。若省略 WHERE 语句则删除表中 所有的数据。

【例 5-10】 使用 SQL 语句删除 student 表中 age 为 18 的学生数据。

打开 MySQL 8.0 Command Line Client, 输入以下语句:

DELETE FROM student WHERE age = 18;

执行结果如图 5-12 所示。

```
sql> DELETE FROM student
     WHERE age=18;
uery OK, 2 rows affected (0.02 sec)
```

重要提示:如果 DELETE 语句中没有 WHERE 子 句的限制,则表中所有数据均被删除。

2. 使用 TRUNCATE 语句删除数据

图 5-12 删除数据

使用 DELETE 语句删除记录时,若要删除表中的

所有记录,目表中记录很多时,则删除命令执行较慢。在删除表中所有数据时,使用 TRUNCATE TABLE 语句速度更快。

语法格式如下:

TRUNCATE TABLE 表名;

【例 5-11】 使用 SQL 语句删除 student 表中的所有数据。 打开 MvSQL 8.0 Command Line Client,输入以下语句:

说明: TRUNCATE TABLE 在功能上与不带 WHERE 子句的 DELETE 语句相同: TRUNCATE 语句不能带 WHERE 子句,所以它只能删除表中的

所有数据; TRUNCATE 语句删除表中的数据后,再向表中插入数据时,自动增加的字段默 认初始值重新从1开始;使用 DELETE 语句删除表中的所有数据后,再向表中插入数据 时,自动增加的字段值会从记录中该字段最大值加1开始编号; DELETE 语句每删除一行



数据都会记录在系统操作日志中,TRUNCATE 语句在删除数据时,不会在日志中记录删除的内容,无法恢复数据,所以在使用时需要慎重。

5.5 实现学生管理数据库表约束

当定义了数据完整性约束,每次更新数据时,MySQL都会检验数据内容是否符合相应 的完整性约束条件,只有符合完整性约束条件的数据才被允许更新。在定义约束前,应先确 定约束的类型,不同类型的约束强制不同类型的数据完整性。约束可以使用 Navicat 图形 化管理工具和 SQL 语句设置。

5.5.1 PRIMARY KEY 约束

1. 使用 Navicat 图形化管理工具设置 PRIMARY KEY 约束

【例 5-12】 使用 Navicat 图形化管理工具设置 studb 数据库中的 student 表中 stuid 字 段为主键。

操作步骤如下。

(1) 启动 Navicat 图形化管理工具,连接 mytest 服务器,双击 studb 数据库,使其处于 打开状态,在 studb 数据库下单击"表"节点,选中 student 表,在对象窗口中单击"设计表"。

(2) 选择 stuid 列,右击,从弹出的快捷菜单中选择"主键"命令,如图 5-14 所示。

ľ	名	类	型	长度	小数点	不是 null	
>	stuid	ct	nar	12	0		
	stuname	● 】	夏制	10	0		_
	sex	lis 1	钻贴	2	0		
	age	. L.	泰加栏位	0	0		
	major	⇒를 3	重入栏位	50	0		
		: •≣ ∄	夏制栏位 删除栏位				
		P	192				
		*	上移 下移				

图 5-14 设置主键

(3)此时,该行的最后一列会出现一个"钥匙"图标,单击工具栏中的"保存"按钮,关闭 窗口即可。

2. 使用 SQL 语句设置 PRIMARY KEY 约束

创建表时可通过定义 PRIMARY KEY 约束来创建主键。可以使用两种方式定义主键 来作为列或表的完整性约束。作为列的完整性约束时,只需在列定义时加上关键字 PRIMARY KEY。作为表的完整性约束时,需要在语句最后加上一条 PRIMARY KEY(列 名,…)语句。

【例 5-13】 在 studb 数据库中使用 SQL 语句创建 teacher 表,将 teacid 列定义为 PRIMARY KEY 约束。

打开 MySQL 8.0 Command Line Client,输入以下语句:



CREATE TABLE teacher(
 teacid char(10) PRIMARY KEY,
 teacname char(8),
 teacage int,
 department varchar(50));

执行结果如图 5-15 所示。

-> teacid char(10) PRIMARY KE -> teacname char(8),	ίΥ,
\rightarrow teacname char(8),	
-> teacage int.	
\rightarrow department varchar(50));	
Query OK. O rows affected (0.08 sec)	

图 5-15 成功设置 PRIMARY KEY 约束

上述 SQL 语句执行后,输入命令"DESC teacher;"可以看到 teacid 字段的 Key 值为 PRI, PRI 即主键的标识, 如图 5-16 所示。

Field	Туре	Null	Key	Default	Extra
teacid teacname teacage department	char(10) char(8) int varchar(50)	NO YES YES YES	PRI	NULL NULL NULL NULL	

图 5-16 查看 PRIMARY KEY

输入数据进行验证:

INSERT INTO teacher VALUES('18302001','张阳',40,'人工智能学院');

执行结果如图 5-17 所示。

mysql> INSERT INTO teacher VALUES('18302001','张阳',40,'人工智能学院'); Query OK, 1 row affected (0.01 sec)

图 5-17 添加一条数据

数据正常添加后,再输入同样的一条数据到 teacher 表,执行结果如图 5-18 所示。

mysql> INSERT INTO teacher VALUES('18302001','张阳',40,'人工智能学院'); ERROR 1062 (23000): Duplicate entry '18302001' for key 'teacher.PRIMARY'

图 5-18 违反 PRIMARY KEY 约束条件

添加失败,因为表中已有"18302001"这条数据,对于 PRIMARY KEY 约束来说,它是 重复添加了,违反了 PRIMARY KEY 约束。

【例 5-14】 使用 SQL 语句在 studb 数据库中创建 teac_course 表,将 teacid 和 cno 设置 为组合 PRIMARY KEY 约束。

打开 MySQL 8.0 Command Line Client,输入以下语句:

CREATE TABLE teac_course(
 teacid char(10),
 cno char(20),
 cname char(20),
 credit int,
 PRIMARY KEY (teacid, cno));

执行结果如图 5-19 所示。

mysq1>	CREATE TABLE teac_course(
\rightarrow	teacid char(10),
->	cno char(20),
\rightarrow	cname char(20),
\rightarrow	credit int,
\rightarrow	PRIMARY KEY(teacid, cno));
Query (OK, 0 rows affected (0.14 sec)

图 5-19 成功定义组合 PRIMARY KEY 约束

3. 为已存在的表设置 PRIMARY KEY 约束

语法格式如下:

ALTER TABLE 表名 MODIFY 字段名 数据类型 PRIMARY KEY;

【例 5-15】 使用 SQL 语句创建 teacher2 表,结构同 teacher 表,再将 teacid 设置为 主键。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
CREATE TABLE teacher2
(teacid char(10),
    teacname char(8),
    teacage int,
    department varchar(50)
```

);

执行结果如图 5-20 所示。



图 5-20 创建 teacher2 表

修改 teacher2 表,为 teacid 设置主键。

ALTER TABLE teacher2 MODIFY teacid char(10) PRIMARY KEY;

执行结果如图 5-21 所示。

mysql> ALTER TABLE teacher2 MODIFY teacid char(10) PRIMARY KEY; Query OK, O rows affected (0.24 sec) Records: O Duplicates: O Warnings: O

图 5-21 为 teacid 设置主键

4. 为已存在的表设置复合 PRIMARY KEY 约束 语法格式如下:

```
ALTER TABLE 表名 ADD PRIMARY KEY(字段名 1,字段名 2);
```

【例 5-16】 使用 SQL 语句在 studb 数据库中创建 teac_course1 表,结构同 teac_course 表,再将 teacid 和 cno 设置为复合 PRIMARY KEY 约束。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
CREATE TABLE teac_coursel(
teacid char(10),
cno char(20),
cname char(20),
credit int
```

);

62

执行结果如图 5-22 所示。



图 5-22 创建 teac_coursel 表

修改 teac_coursel 表,为 teacid 和 cno 设置复合 PRIMARY KEY 约束。

ALTER TABLE teac_course1 ADD PRIMARY KEY(teacid, cno);

执行结果如图 5-23 所示。



图 5-23 设置复合 PRIMARY KEY 约束

5. 删除 PRIMARY KEY 约束

语法格式如下:

ALTER TABLE 表名 DROP PRIMARY KEY;

【例 5-17】 使用 SQL 语句删除 teac_coursel 表的 PRIMARY KEY 约束。 打开 MySQL 8.0 Command Line Client,输入以下语句:

ALTER TABLE teac_course1 DROP PRIMARY KEY;

执行上述语句后,输入"DESC teac_course1;"语句验证,可以看到两个字段的 Key 值为 NULL, PRIMARY KEY 约束已被删除,如图 5-24 所示。

ysql> AL uery OK, ecords: ysql> DE	TER TABLE t O rows aff O Duplicat SC teac_cou	eac_cou ected (es: 0 rsel;	rsel D 0.09 s Warnin	ROP PRIMAR ec) gs: 0	Y KEY;
Field	Туре	Nu11	Key	Default	Extra
teacid cno cname credit	char(10) char(20) char(20) int	NO NO YES YES		NULL NULL NULL NULL	

图 5-24 删除 PRYIMARY KEY 约束的执行结果

5.5.2 DEFAULT 约束

1. 使用 Navicat 图形化管理工具设置 DEFAULT 约束

【例 5-18】 使用 Navicat 图形化管理工具设置 studb 数据库中 student 表里的 age 默认值为 18。

操作步骤如下。

(1) 启动 Navicat 图形化管理工具,连接 mytest 服务器,双击 studb 数据库,使其处于 打开状态,在 studb 数据库下单击"表"节点,选中 student 表,在对象窗口中单击"设计表"。

(2) 选中 age,在下方窗口中的"默认"文本框中输入"18",如图 5-25 所示。

名	类型	长度	小数点	不是 null	
stuid	char	12	0		<i>P</i> 1
stuname	char	10	0		
sex	char	2	0		
age	int	0	0		
major	varchar	50	0		
默 认:	18		~		
默认: 注释:	18		~		
默认: 注释: □ 自动递增	18		~		
默认: 注释: □ 自动递增 □ 无符号	18		~		

图 5-25 设置 DEFAULT 约束

(3) 单击工具栏中的"保存"按钮,完成 DEFAULT 约束的设置。

重要提示:设置默认值为中文时需要给该中文默认值加一对英文的双引号。

2. 使用 SQL 语句在创建表时设置 DEFAULT 约束

语法格式如下:

CREATE TABLE 表名(字段名 数据类型 数据长度 DEFAULT 默认值);

【例 5-19】 使用 SQL 语句在 studb 中创建 coursel 表,结构与 teac_coursel 表相同,并 设置 credit 默认值为 4。

打开 MySQL 8.0 Command Line Client,输入以下语句:

CREATE TABLE course1(
 teacid char(10),
 cno char(20),
 cname char(20),
 credit int DEFAULT 4
);

64

贝频讲解

执行结果如图 5-26 所示。

输入命令"DESC coursel;"进行验证, credit 的默认值(见 Default 列)已变为 4, 如图 5-27 所示。



Field	Туре	Null	Key	Default	Extra
teacid	char (10)	YES		NULL	
cno	char(20)	YES		NULL	
credit	int	YES		4	

图 5-26 设置 DEFAULT 约束

图 5-27 查看 DEFAULT 约束

3. 使用 SQL 语句为已存在的表设置 DEFAULT 约束

语法格式如下:

ALTER TABLE 表名 MODIFY 字段名 数据类型 DEFAULT 默认值;

【例 5-20】 使用 SQL 语句修改 studb 数据库中的 teacher 表,设置 teacage 默认值为 40。 打开 MySQL 8.0 Command Line Client,输入以下语句:

ALTER TABLE teacher MODIFY teacage int DEFAULT 40;

执行后输入命令"DESC teacher;"查看表结构,如图 5-28 所示。

sql> DESC t	eacher;	warning	s: 0		
Field	Туре	Nu11	Кеу	Default	Extra
teacid teacname teacage department	char(10) char(8) int varchar(50)	NO YES YES YES	PRI	NULL NULL 40 NULL	

图 5-28 设置并查看 DEFAULT 约束

4. 删除 DEFAULT 约束

语法格式如下:

ALTER TABLE 表名 MODIFY 字段名 数据类型;

【例 5-21】 使用 SQL 语句删除 teacher 表中 teacage 字段设置的默认值。 打开 MySQL 8.0 Command Line Client,输入以下语句:

```
ALTER TABLE teacher
MODIFY teacage int;
```

执行后并查看表结构,如图 5-29 所示。可以看到 teacage 字段的默认值已经没有了,表示删除成功。

Field	Туре	Null	Key	Default	Extra
teacid teacname teacage department	char (10) char (8) int varchar (50)	NO YES YES YES	PRI	NULL NULL NULL NULL	

图 5-29 删除 DEFAULT 约束

5.5.3 CHECK 约束

1. 创建表时设置 CHECK 约束

语法格式如下:

CREATE TABLE 表名(字段名 数据类型 数据长度 CHECK(表达式));

语法说明如下。

表达式:指定需要检查的条件,在更新表数据时,MySQL 会检查更新后的数据行是否 满足 CHECK 约束的条件。

【例 5-22】 使用 SQL 语句在 studb 数据库中,创建 student1 表,结构与 studb 数据库中的 student 表结构相同,sex 字段的取值范围是"男"或"女"。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
CREATE TABLE student1(stuid char(12),
stuname char(10),
sex char(2) CHECK(sex IN('男','女')),
age int,
major varchar(50));
```

执行结果如图 5-30 所示。

```
mysql> CREATE TABLE student1(stuid char(12),

-> stuname char(10),

-> sex char(2) CHECK(sex IN('男','女')),

-> age int,

-> major varchar(50));

Query OK, 0 rows affected (0.07 sec)
```

图 5-30 设置 CHECK 约束

2. 为已存在的表设置 CHECK 约束 语法格式如下:

ALTER TABLE 表名 ADD CHECK(表达式);

【例 5-23】 使用 SQL 语句设置 studb 数据库中 course 表的 credit 的值为 1~5。 打开 MySQL 8.0 Command Line Client,输入以下语句:

```
ALTER TABLE course
ADD CHECK(credit BETWEEN 1 AND 5);
```

执行结果如图 5-31 所示。

mysql> ALTER TABLE course -> ADD CHECK(credit BETWEEN 1 AND 5); Query OK, 0 rows affected (0.12 sec) Records: 0 Duplicates: 0 Warnings: 0

图 5-31 为已存在的表设置 CHECK 约束

说明:在目前的 MySQL 版本中,CHECK 约束还没有被强化,虽然可以被 MySQL 分析,但在更新数据时,CHECK 约束并不起作用。

3. 删除 CHECK 约束

语法格式如下:

ALTER TABLE 表名 DROP CHECK 检查约束名

【例 5-24】 使用 SQL 语句删除 course 表中 credit 字段的 CHECK 约束。 打开 MySQL 8.0 Command Line Client,先查看系统自动生成的约束名,输入以下语句:

SHOW CREATE TABLE course;

执行结果如图 5-32 所示,这里的 CHECK 约束名是 course_chk_1。

my +	sq1> SHOW CREATE TABLE course;
	Table Create Table
+	
)	course CREATE TABLE `course` (`teacid char(10) NOT NULL, `cno char(20) NOT NULL, `cradme` char(20) DEFAULT NULL, `cradme` char(20) DEFAULT NULL, PRIMARY KEY (`teacid`, `cno`), CONSTRAINT `course_chk_1` CHECK ((`credit` between 1 and 5)) ENGINE=InnoDB DEFAULT CHARSET=gb2312
1	row in set (0.03 sec)

图 5-32 查看 CHECK 约束名

输入以下语句删除 CHECK 约束:

ALTER TABLE course DROP CHECK course_chk_1;

输入以下语句查看 CHECK 约束:

SHOW CREATE TABLE course;

执行结果如图 5-33 所示。CHECK 约束已被删除。

my Qu Re	vsql> ALTER TABLE course DROP CHECK course_chk_1; nery OK, 0 rows affected (0.08 sec) ecords: 0 Duplicates: 0 Warnings: 0
ту +	/sql> SHOW CREATE TABLE course;
ļ	Table Create Table
) +	course CREATE TABLE 'course' ('teacid' char(10) NOT NULL, 'cno' char(20) NOT NULL, 'cname' char(20) DEFAULT NULL, 'credit' int DEFAULT NULL, PRIMARY KEY ('teacid', 'cno') ENGINE=InnoDB DEFAULT CHARSET=gb2312
1	row in set (0.00 sec)

图 5-33 删除并查看 CHECK 约束

5.5.4 UNIQUE 约束

1. 创建表时设置 UNIQUE 约束

语法格式如下:

CREATE TABLE 表名(字段名 数据类型 数据长度 UNIQUE);

【例 5-25】 使用 SQL 语句在 studb 数据库中创建 student2 表,结构与 studb 数据库中 的 student 表结构相同,将 stuname 字段设置为唯一键。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
CREATE TABLE student2(stuid char(12),
    stuname char(10) UNIQUE,
    sex char(2),
    age int,
    major varchar(50));
```

执行结果如图 5-34 所示。 输入数据验证 UNIQUE 约束。

INSERT INTO student2 VALUES('2210010','黄顺','男',19,'软件技术');

以上数据录入后,再添加一条数据到 student2 表。

68



图 5-34 创建表时设置 UNIQUE 约束

INSERT INTO student2 VALUES('2210011','黄顺','女',18,'软件技术');

执行结果如图 5-35 所示。



图 5-35 验证 UNIQUE 约束

第二条数据录入失败,因为 stuname 字段的值与上一条数据重复了,违反了 stuname 的 UNIQUE 约束。

2. 为已存在的表设置 UNIQUE 约束

语法格式如下:

ALTER TABLE 表名 MODIFY 字段名 字段类型 UNIQUE;

【例 5-26】 使用 SQL 语句为 student1 表的 stuname 字段设置 UNIQUE 约束。 打开 MySQL 8.0 Command Line Client,输入以下语句:

ALTER TABLE student1 MODIFY stuname char(10) UNIQUE;

执行结果如图 5-36 所示。

mysql> ALTER TABLE student1 MODIFY stuname char(10) UNIQUE; Query OK, 0 rows affected (0.12 sec) Records: 0 Duplicates: 0 Warnings: 0

图 5-36 为已存在的表设置 UNIQUE 约束

3. 删除 UNIQUE 约束

语法格式如下:

ALTER TABLE 表名 DROP INDEX 约束名;

【例 5-27】 使用 SQL 语句删除 student1 表中 stuname 字段的 UNIQUE 约束。

打开 MySQL 8.0 Command Line Client, 先查看系统自动生成的约束名, 输入以下语句:

SHOW CREATE TABLE student1;

执行结果如图 5-37 所示,这里检查约束名是 stuname。

mysq1> SHOV +	V CREATE TABLE student1;
1	
Table	Create Table
student1 stuid o stuname sex cha age in major UNIQUE KI CONSTRAID ENGINE=11	CREATE TABLE `student1` (char(12) DEFAULT NULL, char(10) DEFAULT NULL, ar(2) DEFAULT NULL, t DEFAULT NULL, varchar(50) DEFAULT NULL, Y `stuname` (`stuname`), NT `student1_chk_1` CHECK ((`sex` in (_gb2312'??', _gb2312'?'))) noDB DEFAULT CHARSET=gb2312
1	
l row in se	et (0.01 sec)

图 5-37 查看 UNIQUE 约束名

输入以下语句删除检查约束:

ALTER TABLE student1 DROP INDEX stuname;

执行上述语句后,输入命令"DESC student1;"进行验证,可以看到 stuname 字段的 Key 值为 NULL,说明 UNIQUE 约束已被删除,如图 5-38 所示。

rsql> ALT lery OK, o ecords: O rsql> DES	ER TABLE stude 0 rows affecte Duplicates: C student1;	nti DRO d (0.04 0 Warn	P INDE sec) ings: U) stuname;	
Field	 Туре	Nu11	Key	Default	Extra
stuid stuname sex age major	char (12) char (10) char (2) int varchar (50)	YES YES YES YES YES		NULL NULL NULL NULL NULL	

图 5-38 删除 UNIQUE 约束并查看

5.5.5 NOT NULL 约束

1. 创建表时设置 NOT NULL 约束

语法格式如下:

CREATE TABLE 表名(字段名 数据类型 数据长度 NOT NULL);

【例 5-28】 使用 SQL 语句创建 student3 表,结构同 student 表,将字段 stuname 设置 成 NOT NULL 约束。

■ 前間

打开 MySQL 8.0 Command Line Client,输入以下语句:

CREATE TABLE student3(stuid char(12), stuname char(10) NOT NULL,

```
sex char(2),
age int,
major varchar(50));
```

70

执行结果如图 5-39 所示。

mysql> CREA	TE TABLE student3(stuid char(12),
\rightarrow	stuname char(10) NOT NULL,
->	sex char(2),
\rightarrow	age int,
->	major varchar(50));
Query OK, O	rows affected (0.07 sec)

图 5-39 创建表的同时设置 NOT NULL 约束

输入命令"DESC student3;"进行验证,验证结果如图 5-40 所示。

Field	Туре	Null	Key	Default	Extra
stuid	char(12)	YES		NULL	
stuname	char(10)	NO		NULL	
sex	char(2)	YES		NULL	
age	int	YES		NULL	
major	varchar(50)	YES		NULL	

图 5-40 验证结果

可以看到, stuname 字段的 Null 值为 NO, 说明该字段已被设置为 NOT NULL 约束。

2. 为已存在的表设置 NOT NULL 约束

语法格式如下:

ALTER TABLE 表名 MODIFY 字段名 数据类型 数据长度 NOT NULL;

【例 5-29】 使用 SQL 语句为 student2 表的 stuname 字段设置 NOT NULL 约束。 打开 MySQL 8.0 Command Line Client,输入以下语句:

ALTER TABLE student2 MODIFY stuname char(10) NOT NULL;

执行上述语句后,输入命令"DESC student2;"进行验证,此时 stuname 字段的 Null 值为 NO,如图 5-41 所示。

cords: 0	Duplicates:	0 Warn	ings:	0	
sq1> DESC	student2;	·	+	·	
Field	Туре	Null	Key	Default	Extra
stuid stuname sex age major	char (12) char (10) char (2) int varchar (50)	YES NO YES YES YES	PRI	NULL NULL NULL NULL NULL	

图 5-41 为已存在的表设置 NOT NULL 约束并查看

3. 删除 NOT NULL 约束

语法格式如下:

ALTER TABLE 表名 MODIFY 字段名 数据类型;

【例 5-30】 使用 SQL 语句删除 student2 表的 stuname 字段的 NOT NULL 约束。 打开 MySQL 8.0 Command Line Client,输入以下语句:

ALTER TABLE student2 MODIFY stuname char(10);

执行上述语句后,输入命令"DESC student2;"进行验证, stuname 字段的 Null 值为 YES,说明 NOT NULL 约束已被删除,如图 5-42 所示。

corus: 1	Duplicates:	0 Warn	ings: (0	
sq1> DES	C student2; 	·	I		·
Field	Туре	Null	Key	Default	Extra
stuid	char(12)	YES		NULL	
stuname	char (10)	YES	UNI	NULL	
sex	char(2)	YES		NULL	
age	int	YES		NULL	
maior	varchar(50)	YES		NULL	

图 5-42 删除 NOT NULL 约束并验证

5.5.6 FOREIGN KEY 约束

1. 使用 Navicat 图形化管理工具设置 FOREIGN KEY 约束

【例 5-31】 使用 Navicat 图形化管理工具在 studb 数据库中,设置 course 表的 cno 为 外键,参考 teac_coursel 表的 cno。

操作步骤如下。

(1) 启动 Navicat 图形化管理工具,连接 mytest 服务器,双击 studb 数据库,使其处于 打开状态,在 studb 数据库下单击"表"节点,选中 course 表,在对象窗口中单击"设计表"。

(2) 在"外键"标签中输入如图 5-43 所示的数据。

栏位	索引	外键	触发器	选项	注释	SQL	预览			
名		栏位			参考数据	库	参考表	参考栏位	删除时	更新时
I pk_re	eference	1 cno			studb		teac_course1	cno	RESTRICT	RESTRICT

图 5-43 输入数据

(3) 单击"保存"按钮,完成设置。

2. 创建表时设置 FOREIGN KEY 约束

语法格式如下:

CREATE TABLE 子表名 (字段名 字段类型 字段长度

72

```
FOREIGN KEY 字段名 [,字段名 2,…] REFERENCES <父表名> 主键列 1 [,主键列 2,…);
```

【例 5-32】 使用 SQL 语句在 studb 数据库中创建 teac_course_new 表,设置 teacid 为 外键,参考 teacher 表的 teacid 字段。

打开 MySQL 8.0 Command Line Client,输入以下语句:

```
CREATE TABLE teac_course_new(
    teacid char(10),
    teacname char(8) NOT NULL,
    cno char(20) NOT NULL,
    semester int,
    FOREIGN KEY(teacid) REFERENCES teacher(teacid));
```

执行结果如图 5-44 所示。

mysql>	CREATE TABLE teac_course_new(
->	teacid char(10),	
->	teacname char(8) NOT NULL,	
->	cno char (20) NOT NULL,	
->	semester int,	
->	FOREIGN KEY (teacid) REFERENCES	<pre>teacher(teacid));</pre>
Query (OK, O rows affected (0.07 sec)	

图 5-44 设置 FOREIGN KEY 约束

验证 FOREIGN KEY 约束,先查询 teacher 表中的数据,结果如图 5-45 所示。

teacid	teacname	teacage	department
18302001		40	人工智能学院

图 5-45 teacher 表中的数据

在 teac_course_new 表中添加一条记录,如下:

INSERT INTO teac_course_new VALUES('20302002','沈铭','202',3);

执行结果如图 5-46 所示。

ysql〉INSERT INTO teac_course_new VALUES('20302002','沈铭','202',3); RROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`studb`.`teac_course_new`, CONSTR NT teac_course_new_ibfk_1`FOREIGN KEY (`teacid`) REFERENCES` teacher` (`teacid`))

图 5-46 验证 FOREIGN KEY 约束

数据录入失败,因为违反了外键参照完整性约束,teacid 字段的取值"20302002"在 teacher 表中的 teacid 字段中不存在。

3. 为已存在的表设置 FOREIGN KEY 约束

语法格式如下:

```
ALTER TABLE 子表名 ADD FOREIGN KEY(字段名) REFERENCES 父表(字段名);
```

【例 5-33】 使用 SQL 语句在 studb 数据库中将之前创建的 teacher1 表和 teac_course1

表的 teacid 字段设置 FOREIGN KEY 约束。

打开 MySQL 8.0 Command Line Client,输入以下语句:

ALTER TABLE teac course1 ADD FOREIGN KEY(teacid) REFERENCES teacher1(teacid);

执行结果如图 5-47 所示。

mysql> ALTER TABLE teac_coursel ADD FOREIGN KEY(teacid) REFERENCES teacher1(teacid); Query OK, 0 rows affected (0.23 sec) Records: 0 Duplicates: 0 Warnings: 0

图 5-47 为已存在的表设置 FOREIGN KEY 约束

重要提示:设置外键时,被引用表(主键表)必须设置了主键或唯一键,并且数据类型和 长度必须与外键一致。

4. 删除 FOREIGN KEY 约束

语法格式如下:

ALTER TABLE 子表名 DROP FOREIGN KEY 约束名;

【例 5-34】 使用 SQL 语句删除 teac_course_new 表 teacid 字段的 FOREIGN KEY 约束。 打开 MySQL 8.0 Command Line Client,输入以下语句:

SHOW CREATE TABLE teac_course_new;

查看 FOREIGN KEY 约束名为 teac_course_new_ibfk_1,执行结果如图 5-48 所示。

mysq1> SHOW CREATE	TABLE teac_course_new;
Table	Create Table
teac_course_new teacid char(10) teacname char(20) NC semester int DF KEY teacid (tea CONSTRAINT teac ENGINE=InnoDB DEF	CREATE TABLE `teac_course_new` (DEFAULT NULL, s) NOT NULL, AT NULL, FAULT NULL, racid`), course_new_ibfk_1` FOREIGN KEY (`teacid`) REFERENCES `teacher` (`teacid`) ^AULT CHARSET=gb2312
1 row in set (0.00	sec)

图 5-48 查看 FOREIGN KEY 约束名

输入以下语句, 删除 FOREIGN KEY 约束并查看:

ALTER TABLE teac_course_new DROP FOREIGN KEY teac_course_new_ibfk_1; SHOW CREATE TABLE teac_course_new;

执行结果如图 5-49 所示。此时 FOREIGN KEY 约束已被删除,两个表之间没有了关 联关系。

74

mysql> ALTER TABLE teac_course_new DROP FOREIGN KEY teac_course_new_ibfk_1; Query OK, 0 rows affected (0.03 sec) Records: 0 Duplicates: 0 Warnings: 0 mysql> SHOW CREATE TABLE teac_course_new;	
-+ Table Create Table	
-+ teac_course_new CREATE TABLE `teac_course_new` (`teacid char(10) DEFAULT NULL, `teacname` char(8) NOT NULL, `cno` char(20) NOT NULL, `semester` int DEFAULT NULL, KEY` teacid` (`teacid`)) ENGINE=InnoDB DEFAULT CHARSET=gb2312	
-+ 1 row in set (0.00 sec)	

图 5-49 删除 FOREIGN KEY 约束并查看

任务实训营

1. 任务实训目的

(1) 掌握使用 Navicat 图形化管理工具向数据表中插入、删除和修改数据的方法。

(2) 掌握使用 SQL 语句向数据表中插入、删除和修改数据的方法。

(3) 掌握使用 Navicat 图形化管理工具和 SQL 语句两种方法来实现约束的方法。

2. 任务实训内容

(1) 使用 Navicat 图形化管理工具向 studb 学生管理数据库中的 Department 表插入数据,数据见表 5-2。

DepartID	DepartName	Chariman	Office
001	信息技术系	江波	301
002	艺术设计系	马康	302
003	外语系	李丽	303
004	体育系	赵伟	304

表 5-2 向 Department 表中插入的数据

(2) 使用 Navicat 图形化管理工具删除 Department 表 DepartID 为 001 的数据。

(3) 使用 SQL 语句向 Stu 表插入数据,数据见表 5-3。

StudentID	StudentName	Sex	Age	DepartID
22001	李明	男	20	001
22002	王芳	女	19	002
22003	赵峰	男	19	003
22004	章雪	女	20	004

表 5-3 向 Stu 表中插入的数据

(4) 使用 SQL 语句将 Stu 表中 StudentID 为 22002 学生的 Age 改为 20。

(5) 使用 SQL 语句删除 Stu 表中 StudentID 为 22001 的数据。

(6) 使用 SQL 语句为 Course 表、Grade 表添加数据,数据自定义。

(7) 为以上表添加如下约束,使用 Navicat 图形化管理工具和 SQL 语句两种方法实现。

① NOT NULL 约束: StudentName。

② PRIMARY KEY 约束: StudentID、DepartID。

③ DEFAULT 约束: Age 字段默认为 19。

④ CHECK 约束: Sex 为"男"或"女"。

(8) 插入、修改和删除数据,体会数据完整性。

(项目小结

本项目介绍数据表的定义、数据类型以及使用 Navicat 图形化管理工具和 SQL 语句向 表中插入、删除和修改数据,介绍数据完整性的概念和类型,以及如何实现约束等。