

第 5 章 创建与维护电子学校系统数据表

任务描述

在数据库中,数据表是数据库中最重要、最基本的操作对象,是数据存储的基本单位。数据表被定义为列的集合,数据在表中是按照行和列的形式来存储的。每一行代表一条唯一的记录,每一列代表记录中的一个域。

本章将详细介绍数据表的基本操作,主要包括创建数据表、查看数据表结构、修改数据表、删除数据表。通过本章的学习,能够熟练掌握数据表的基本概念,理解约束、默认和规则的含义并且学会运用;能够在图形界面模式和命令行模式下熟练地完成有关数据表的常用操作。

学习目标

- (1) 掌握创建表的方法。
- (2) 掌握表的完整性约束条件。
- (3) 掌握查看表结构的方法。
- (4) 掌握修改表的方法。
- (5) 掌握删除数据表的方法。

学习导航

本任务主要讲解数据库应用系统开发中数据表的建立技术。依据对数据库系统进行设计、创建、使用、优化、管理及维护这一操作流程,本任务属于对数据信息的创建阶段。学习如何根据数据库应用系统的功能需求,使用 CREATE、ALTER 等语句实现对数据表的建立、修改等操作,为数据库应用系统的开发与使用奠定坚实的基础,数据表操作学习导航如图 5.1 所示。

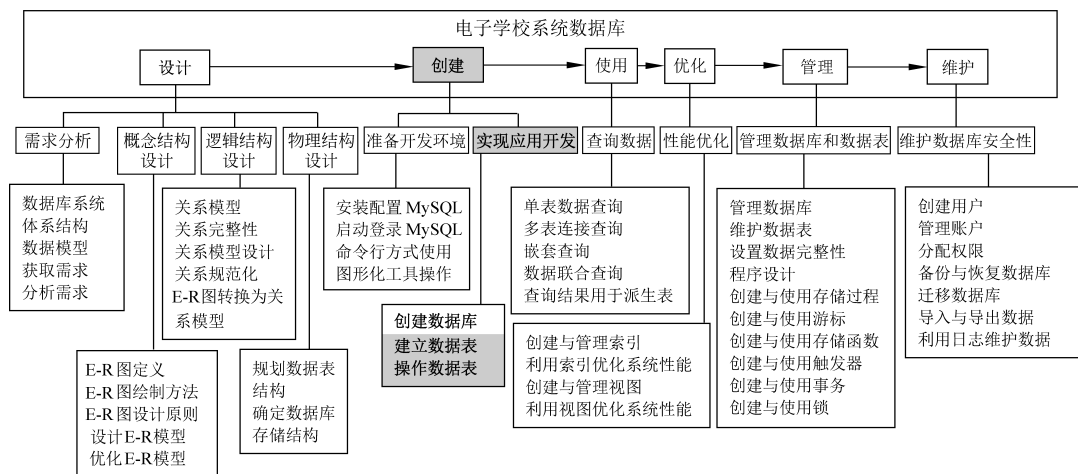


图 5.1 数据表操作学习导航

任务 5.1 规划与设计数据表

任务说明：数据类型是数据的特征之一，决定数据的存储格式，代表不同的信息类型。每一列、变量、表达式和参数都有各自的数据类型。

5.1.1 数据表的基本概念

表是数据库中用户存储所有数据的对象，是关系模型中表示实体的方式，是组成数据库的基本元素。可以说没有表，也就没有数据库。在一个关系数据库中，可以包含多张表，所有数据存储在表中。

数据表是 MySQL 数据库对象，在数据表中，数据以行和列的形式存储在规范化的二维表格中。MySQL 数据表主要由行和列构成。表类似于电子表格软件的工作表，但更规范。MySQL 中每张表都有一个名字，以标识该表。如图 5.2 所示数据表的名字是 student。下面说明一些与表有关的术语。

	stu_no	stu_name	stu_sex	stu_politicalstatus	stu_birthday	stu_identitycard	stu_speciality	stu_address	stu_postcode
☐	201803010001	靳锦东	男	预备党员	1997-09-27	130100199709273457	数字艺术	河北省石家庄	050000
☐	201901010001	张晓辉	男	共青团员	2000-11-15	120101200011156679	软件技术	天津市和平区	300041
☐	201901010002	李红丽	女	共青团员	2001-03-07	120106200103074566	软件技术	天津市河东区	300171
☐	201901010003	孙云	男	共青团员	2000-08-12	330100200008126677	软件技术	浙江省杭州市	310000
☐	201901020001	赵辉	男	共青团员	1999-11-09	120022199911018879	信息应用	天津市西青区	300380
☐	201902010001	王强	男	共青团员	1999-05-18	120103199905183557	网络技术	天津市河西区	300202
☐	201902010002	刘晓薇	女	预备党员	1998-10-23	120101199810233566	网络技术	天津市和平区	300041
☐	201902020001	李立	男	共青团员	1999-07-13	120106199907136349	网络安全	天津市河东区	300171
☐	201903020001	王丽莉	女	共青团员	2000-09-12	120101200009123466	视觉传达	天津市和平区	300041
☐	201903020002	刘梅梅	女	共青团员	1999-04-28	110101199904289786	视觉传达	北京市东城区	100010

图 5.2 student

(1) 表结构：每个数据库包含若干张表。每张表具有一定的结构，称为“表型”。所谓表型是指组成表的名称及数据类型，也就是通常表格的“栏目信息”。

(2) 表：表是由定义的列数和可变的行数组成的逻辑结构。

(3) 列：用来保存对象的某一类属性。每列又称为一个字段，每列的标题称为字段名。

(4) 行：用来保存一条记录，是数据对象的一个实例，包括若干信息项。

(5) 记录：每张表包含了若干行数据，它们是表的“值”，表中的一行称为一个记录，每一行都是实体的一个完整描述。个体可以是人也可以是物，甚至可以是一个概念。因此，表是记录的有限集合。

(6) 字段：每个记录由若干个数据项构成，将构成记录的每个数据项称为字段。

(7) 关键字：在学生信息表 student 中，若不加以限制，每个记录的姓名(stu_name)、性别(stu_sex)、系名(stu_speciality)、出生日期(stu_birthday)和邮编(stu_postcode)这 5 个字段的值都有可能相同，但是学号字段的值对表中所有记录来说一定不同，学号是关键字，也就是说通过“学号”字段可以将表中的不同记录区分开。

在 MySQL 数据库系统中，可以按照不同的标准对表进行分类。

1. 按照表的用途分类

(1) 系统表：用于维护 MySQL 服务器和数据库正常工作的数据表。例如，系统数据库 MySQL 中就存在若干系统表。

(2) 用户表：由用户自己创建的、用于各种数据库应用系统开发的表。

(3) 分区表：分区表是将数据水平划分为多个单元的表，这些单元可以分布到数据库中的多个文件组中。在维护整个集合的完整性时，使用分区可以快速而有效地访问或管理数据子集，从而使大型表或索引更易于管理。

2. 按照表的存储时间分类

(1) 永久表：包括 MySQL 的系统表 and 用户数据库中创建的数据表，该类表除非人工删除，否则一直存储在介质中。

(2) 临时表：临时表只有该表的用户在创建该表时是可见的。如果服务器关闭，则所有临时表会被清空、关闭。

5.1.2 MySQL 数据类型的含义与选用原则

人们都要将现实世界的各类数据抽象后放入数据库中，然而各类信息以什么格式、多大的存储空间进行组织和存储，这就有赖于人们事先的规定。例如，把 2020-02-19 规定为日期格式，就能正常地识别这组字符串的含义，否则就只是一堆无意义的数字。这就是进行数据类型定义的意义。

数据库存储的对象主要是数据，现实中存在着各种不同类型的数据，数据类型就是以数据的表现方式和存储方式来划分的数据种类。有了数据类型就能对数据进行分类，并且对不同类型的数据操作进行定义，进一步赋予该类数据的存储和操作规则。

1. 整数类型

整数由正整数、负整数和 0 组成,如 39、25、-2 和 33967。在 MySQL 中,整数存储的数据类型有 tinyint、smallint、mediumint、int 和 bigint。这些类型在很大程度上是相同的,只有它们存储的值的大小不同,如表 5.1 所示。

表 5.1 整数类型

类 型	存储空间/B	最 小 值	最 大 值
tinyint	1	有符号数 $-128(-2^7)$ 无符号数 0	有符号数 $127(2^7-1)$ 无符号数 $255(2^8-1)$
smallint	2	有符号数 $-32768(-2^{15})$ 无符号数 0	有符号数 $32767(2^{15}-1)$ 无符号数 $65535(2^{16}-1)$
mediumint	3	有符号数 $-8388608(-2^{23})$ 无符号数 0	有符号数 $8388607(2^{23}-1)$ 无符号数 $16777215(2^{24}-1)$
int	4	有符号数 $-2147483648(-2^{31})$ 无符号数 0	有符号数 $2147483647(2^{31}-1)$ 无符号数 $4294967295(2^{32}-1)$
bigint	8	有符号数 $-9223372036854775808(-2^{63})$ 无符号数 0	有符号数 $9223372036854775807(2^{63}-1)$ 无符号数 $18446744073709551615(2^{64}-1)$

如果超出类型范围的操作,会给出 out of range 错误提示。为了避免此类问题发生,在选择数据类型时要根据应用的实际情况确定其取值范围,最后根据确定的结果慎重选择数据类型。

如果指定一个字段的类型为 int(10),就表示该数据类型指定的显示宽度为 10。显示宽度和数据类型的取值范围无关,显示宽度只是指明 MySQL 最大可能显示的数字个数,数值的位数小于指定的宽度时会用空格填充。如果插入了大于显示宽度的值,只要该值不超过该类型整数的取值范围,数值依然可以插入,而且能够显示出来。

2. 浮点数类型和定点数类型

MySQL 中使用浮点数和定点数来表示小数。浮点数类型有两种:单精度浮点类型(float)和双精度浮点类型(double)。定点数类型只有一种:decimal。浮点数类型和定点数类型都可以用(m,n)来表示,其中 m 称为精度,表示总共的位数; n 称为标度,表示小数的位数。例如语句 float(7,3)规定显示的值不会超过 7 位数字,小数点后面有 3 位数字。表 5.2 列举了 MySQL 中浮点数类型和定点数类型所对应的字节数及其取值范围。

表 5.2 MySQL 浮点数类型和定点数类型所对应的字节数及其取值范围

数 据 类 型	字节数/B	有符号数的取值范围	无符号数的取值范围
float	4	$-3.402823466\text{E}+38 \sim$ $1.175494351\text{E}-38$	0 和 $1.175494351\text{E}-38 \sim$ $3.402823466\text{E}+38$

续表

数据类型	字节数/B	有符号数的取值范围	无符号数的取值范围
double	8	$-1.7976931348623157E+308 \sim 2.2250738585072014E-308$	0 和 $2.2250738585072014E-308 \sim 1.7976931348623157E+308$
decimal(m,d)	m+2	$-1.7976931348623157E+308 \sim 2.2250738585072014E-308$	0 和 $2.2250738585072014E-308 \sim 1.7976931348623157E+308$

对于小数点后面的位数超过允许范围的值,MySQL 会自动将它四舍五入为最接近它的值,再插入它。例如,将数据类型为 decimal(6,2)的数据 3.1415 插入数据库后,显示的结果为 3.14。

在 MySQL 中,定点数以字符串的形式存储,在对精度要求比较高(如货币、科学数据等)时使用 decimal 类型比较好。两类浮点数进行减法和比较运算时容易出问题,所以在使用浮点数类型时需要注意,尽量避免做浮点数比较。

3. 字符串类型

字符串类型是数据表中数据存储的重要类型之一。字符串类型主要用来存储字符串或文本信息。在 MySQL 数据库中,常用的字符串类型包括 char、varchar、binary、varbinary 等,如表 5.3 所示。

表 5.3 字符串类型

数据类型	取值范围	说 明
char	0~255 个字符	定长的数据存储形式是 char(n),n 代表存储的最大字符数
varchar	0~65535 个字符	变长的数据存储形式是 varchar(n),n 代表存储的最大字符数
binary	0~255B	定长的数据存储的是二进制数据,形式是 binary(n),n 代表存储的最大字节数
varbinary	0~65535B	变长的数据存储的是二进制数据,形式是 varbinary(n),n 代表存储的最大字节数

char(n)类型和 varchar(n)类型的区别是:char(n)用于存储定长字符串,如果存入的字符串少于 n 个,仍占 n 个字符的空间;而 varchar(n)用于存储长度可变的字符串,其占用的空间为实际长度加一个字符(是字符串结束符)。

text 类型被视为非二进制字符串(字符串)。text 列有一个字符集,并且根据字符集的校对规则对值进行排序和比较。在实际应用中,诸如个人简历、奖惩情况、职业说明、内容简介等信息可设定为 text 数据类型。例如,图书数据处理中的内容简介可以设定为 text 类型。

enum 类型是一种枚举类型,其值在创建时,在列上规定了一列值,语法格式是:字符名 enum(值 1,值 2,...,值 n),enum 类型的字段在取值时,只能在指定的枚举列表中取其中的一个值。因此,对于多个值选取其中一个值的,可以选择 enum 类型。例如,“性别”字段就可以定义成 enum 类型,因为只能在“男”和“女”中选一个。

set 类型是一个字符串对象,可以有 0~64 个值,其定义方式与 enum 类型类似。与 enum 类型的区别是:enum 类型的字段只能从列值中选择一个值,而 set 类型的字段可以从定义的列值中选择多个字符的组合。对于可以选取多个值的字段,可以选择 set 类型。例如,表示兴趣爱好的字段,要求提供多选项选择,可使用 set 数据类型。set('篮球','足球','音乐','电影','看书','画画','摄影'),表示可以选择“篮球”“足球”“音乐”“电影”“看书”“画画”“摄影”中的 0 项或多项。

bit(n)类型是位字段类型,其中 n 表示每个值的位数,范围为 1~64,默认为 1。如某个字段类型为 bit(6),表示该字段最多可存入 6 位二进制,即最大可存入的二进制数为 111111。

binary 类型和 varbinary 类型用于存放二进制字符串,它们之间的区别类似于 char(n)类型和 varchar(n)类型的区别。

blob,指 binary large object,即二进制大对象,是一个可以存储二进制文件的容器。在计算机中,blob 常常是数据库中用来存储二进制文件的字段类型,典型的 blob 是一幅图片或一个声音文件。blob 分为 4 种类型:tinyblob(n)、blob、mediumblob(n)和 longblob(n),它们的区别是存储的最大长度不同。

4. 日期和时间类型

表示时间值的日期和时间类型有 year、date、time、datetime 和 timestamp。

每个时间类型有一个有效值范围和一个“零”值,当指定不合法、在 MySQL 中有不能表示的值时使用“零”值。表 5.4 显示了日期和时间类型的相关特性。

表 5.4 日期和时间类型

类 型	存储长度/B	范 围	格 式	用 途
year	1	1901~2155	yyyy 或'yyyy'	年份值
date	3	1000 01 01~9999 12 31	'yyyy mm dd'或'yyyymmdd'	日期值
time	3	'-838:59:59'~'838:59:59'	hh:mm:ss	时间值或持续时间
datetime	8	1000 01 01 00:00:00~9999 12 31 23:59:59	yyyy mm dd hh:mm:ss	混合日期和时间值
timestamp	4	1970 01 01 00:00:00~2038 01 19 03:14:17	yyyy mm dd hh:mm:ss	混合日期和时间值,时间戳

当只需要显示年份信息时,可以使用 year 类型,可以用 4 位数字格式或 4 位字符串格式,如输入 2020 或'2020'在表中均表示 2020 年。

date 类型用在需要显示年月日的情况,在输入时,年月日中间是否有空格均可。

time 类型用于只需要时间值的情况,取值范围为'-838:59:59'~'838:59:59',其小时部分如此大的原因是 time 类型不仅可以表示一天的时间,还可能是某个过去的时间或两个时间之间的间隔(可能大于 24 小时,甚至为负)。

datetime 用于需要显示年月日和时间的情况,年月日中的空格和时分秒中的符号“:”

是否加上都可以。

timestamp 的显示格式与 datetime 一样,只是 timestamp 的列值范围小于 datetime 类型,另外一个最大的不同是 timestamp 的值与时区有关。

在上述几种日期和时间类型中,其表示格式还有更复杂的变化,在使用过程中需要注意。

5.1.3 数据列属性的含义与设置

MySQL 中,真正约束字段的是数据类型,但是数据类型的约束太单一,需要有一些额外的约束,来更加保证数据的合法性。

MySQL 中的常用列属性有 AUTO_INCREMENT、NOT NULL、NULL、COMMENT、DEFAULT、PRIMARY KEY 以及 UNIQUE KEY 等。

1. 自增长(AUTO_INCREMENT)

设置自动增长属性,只有整型列才能设置此属性。当插入 NULL 值或 0 到一个 AUTO_INCREMENT 列中时,列被设置为 $value+1$,在这里 value 是此前表中该列的最大值。AUTO_INCREMENT 顺序从 1 开始。每张表只能有一个 AUTO_INCREMENT 列,并且它必须被索引。

2. 空属性(NOT NULL | NULL)

指定该列是否允许为空。如果不指定,则默认为 NULL。但是在实际开发过程中,尽可能保证所有的数据都不应该为 NULL,空数据没有意义,空数据没有办法参加运算。

3. 列描述(COMMENT)

实际没有什么含义,是专门用于描述字段的,会根据创建语句保存,用于帮助程序员(或者数据库管理员)进行了解的。主要用于查看创建表的语法。

4. 默认值(DEFAULT)

为列指定默认值,默认值必须为一个常数。其中,blob 和 text 列不能被赋予默认值。如果没有为列指定默认值,MySQL 自动分配一个。如果列可以取 NULL 值,默认值就是 NULL。如果列被声明为 NOT NULL,默认值取决于列类型。

(1) 对于没有声明 AUTO_INCREMENT 属性的数字类型,默认值是 0。对于一个 AUTO_INCREMENT 列,默认值是在顺序中的下一个值。

(2) 对于 timestamp 以外的类型,其默认值是该类型适当的“零”值。对于表中第一个 timestamp 列,默认值是当前的日期和时间。

(3) 对于除 enum 之外的字符串类型,默认值是空字符串。对于 enum,默认值是第一个枚举值。

5. 主键(PRIMARY KEY)

一般情况下,对主键的理解是唯一键,一张表中只能有一个字段可以使用主键,用于约束该字段里面的数据,一张表中最多有一个主键。

6. 唯一键(UNIQUE KEY)

一张表往往有很多字段需要具有唯一性,数据不能重复,这个时候用唯一键就能体现其优势,解决表中多个字段需要唯一性约束的问题。唯一键的本质与主键的性质差不多,唯一键允许字段为空,而且可以多个字段为空(空字段不参与唯一性比较)。

5.1.4 设计电子学校系统数据表结构

在电子学校系统中,系统要记录学生的相关信息,包括学生的姓名、性别、所在系部等,还需要记录该生在大学所学的所有课程,包括课程代码、课程名称、任课教师代码等。此外还会生成学生大学期间所有课程的成绩以及所有任课教师信息,这些数据都需要保存在数据库中。然而数据不能直接存放在数据库中,而是要存放到数据库的数据表中。因此,需要在 eleccollege 数据库中建立相应的数据表,分别存储不同的数据记录。

1. 学生信息表 student

学生信息表 student 保存学校所有学生的信息,包括学号、姓名和身份证号等,如表 5.5 所示。

表 5.5 学生信息表 student

列 名	数 据 类 型	长 度/b	是 否 为 空	说 明
stu_no	char	12	非空	主键 学号
stu_name	char	20	非空	姓名
stu_sex	char	2	非空	性别
stu_politicalstatus	varchar	20		政治面貌
stu_birthday	date		非空	出生年月
stu_identitycard	varchar	18	非空	身份证号
stu_speciality	varchar	40	非空	所学专业
stu_address	varchar	50		家庭住址
stu_postcode	char	6		邮政编码
stu_telephone	varchar	18	非空	联系电话
stu_email	varchar	30	非空	电子邮箱
stu_resume	text			个人简介

续表

列 名	数 据 类 型	长 度 /b	是 否 为 空	说 明
stu_poor	tinyint	1	非空	是否贫困生
stu_enterscore	float		非空	入学成绩
stu_fee	int	11	非空	学费
stu_photo	blob		非空	照片

2. 课程信息表 course

课程信息表 course 保存学校各系部所开设的全部课程信息,包括课程代码、课程名称和任课教师代码等,如表 5.6 所示。

表 5.6 课程信息表 course

列 名	数 据 类 型	长 度 /b	是 否 为 空	说 明
cou_no	char	8	非空	主键 课程代码
cou_name	varchar	20	非空	课程名称
cou_teacher	char	12	非空	任课教师代码
cou_credit	decimal	3,1	非空	课程学分
cou_type	varchar	20	非空	课程性质
cou_term	tinyint	4	非空	开课学期
cou_introduction	text			课程简介

3. 教师信息表 teacher

教师信息表 teacher 保存学校全部教师的信息,包括教师代码、教师姓名、职称等信息,如表 5.7 所示。

表 5.7 教师信息表 teacher

列 名	数 据 类 型	长 度 /b	是 否 为 空	说 明
tea_no	char	12	非空	主键 教师代码
tea_name	char	20	非空	教师姓名
tea_profession	varchar	10	非空	职称
tea_department	char	12	非空	所在系部代码
tea_worktime	datetime		非空	参加工作时间
tea_appointment	varchar	50	非空	聘任岗位
tea_research	varchar	80	非空	研究领域

4. 系部信息表 department

系部信息表 department 保存学校各个系部的信息,包括系部编号、系部名称、系主任等信息,如表 5.8 所示。

表 5.8 系部信息表 department

列 名	数 据 类 型	长 度/b	是 否 为 空	说 明
dep_no	char	12	非空	主键 系部编号
dep_name	varchar	30	非空	系部名称
dep_head	char	10		系主任
dep_phone	char	12		办公电话
dep_office	varchar	30		办公室

5. 班级信息表 class

班级信息表 class 保存学校各个班级的信息,包括班级编号、班级名称、班级人数等信息,如表 5.9 所示。

表 5.9 班级信息表 class

列 名	数 据 类 型	长 度/b	是 否 为 空	说 明
class_id	char	15	非空	主键 班级编号
class_name	varchar	30	非空	班级名称
class_num	int	11	非空	班级人数
class_monitor	char	15		班长
class_teacher	char	12		班主任
class_enteryear	datetime		非空	入学年份

6. 宿舍信息表 dormitory

宿舍信息表 dormitory 保存学校各个宿舍的信息,包括宿舍 ID、宿舍楼编号、房间编号、床位号、学号等信息,如表 5.10 所示。

表 5.10 宿舍信息表 dormitory

列 名	数 据 类 型	长 度/b	是 否 为 空	说 明
dor_serialid	char	15	非空	主键 宿舍 ID
dor_floorid	char	15	非空	宿舍楼编号
dor_roomid	char	15	非空	房间编号