

数据表管理

表是用来存储数据的，也是 MySQL 数据库中最重要数据对象。

下面的内容介绍 MySQL 数据库中表的一些基本概念，数据完整性概念及其应用，在 Navicat for MySQL 中建立用户表、修改用户表的结构、表中约束的建立、表之间的关系建立以及表的删除操作等内容。

一、数据表介绍

数据库中的数据表与我们平常所说的表类似，也是由行和列组成的二维结构。表定义的是列的集合，每一行代表一条记录，每一列代表一个属性，称为字段。例如表 4-1，记录着教工的基本信息，表中每一行代表一个教职工的具体信息，而每一列是教工某一个方面的信息，例如教工号、姓名、性别等。

表 4-1 教工信息表

教工号	姓名	性别	部门	职称	年龄
1	张平乐	男	教务处	副研究员	52
2	李明明	男	科研处	研究员	48
3	彭小玲	女	计算机系	副教授	46
4	陈东浩	男	经管系	讲师	38
5	吴恒	男	外语系	副教授	40
6	刘涵珏	女	外语系	讲师	28
7	李先锋	男	工程系	讲师	30
8	陈忠实	女	计算机系	助教	26

1. 数据表的分类

按照表的用途来分类，可以分为系统表和用户表两类。

系统表是维护 MySQL 服务器和数据库正常工作的数据表，每个数据库下都会建立一些系统表，用户一般不要对系统表进行修改等操作，而是由 DBMS 系统自行维护。在前面我们已经知道，安装好 MySQL 后，会自动创建系统数据库，在这些系统数据库中的表绝大部分都是系统表。

用户表是指在用户自定义的数据表。

2. 表中的列

我们即将介绍数据库中表的建立方法，实际上表的建立就是对表中列的定义，需要为每一列定义列名（字段名）、说明列中的数据的取值范围（数据类型）以及一些其它的内容，比如是否允许为空（Null）等。

表中每一列需要定义列名，即字段名，这些名称需要在建立表时定义，名称要符合标识符的命名规则。

字段的数据类型规定了该列的取值范围和列中数据能够进行的运算。例如姓名应该是字符型的，而年龄应该是数值型的等。

表中的列还需要说明列中的值是否为空（Null），如果不能为空，则这列的数据是必填内容，不能为空。数据表中的内容尽量不要为空，这样计算或统计的结果就不会有误。

3. 用户表的主要内容

用户表是由用户建立的、用于某种实际用途的表。

我们根据实际需要设计数据库时，应该考虑好需要什么样的表，各表中应该保存哪些数据及表之间的关系等内容。在创建表时最有效的方法是将用户表中所需信息一次定义完成，

包括一些约束条件（后面介绍）和附加信息等。

用户表的主要内容有：

- (1) 表的名字，每个表都需要一个好读好记忆的名字；
- (2) 表所包含的基本数据类型及自定义的数据类型；
- (3) 表的各列的名字及每一列的数据类型（有必要的还需说明列的宽度等信息）；
- (4) 表的主码和外码信息；
- (5) 表中哪些列允许空值；
- (6) 哪些列需要索引以及索引的类型；
- (7) 是否要使用以及何时使用约束、默认设置；

用户表的内容很多，我们在下面的内容中首先说明建立用户表框架的基本信息，再逐步完善用户表，建立表间的关联，然后再往表中添加数据，对数据进行必要的维护操作，如插入、修改和删除等。

二、MySQL 的基本数据类型

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。

1. 基本数据类型

MySQL 支持包括整数类型、浮点数类型和定点数类型、字符串类型和日期和时间类型在内的多种基本数据类型。

(1) 整数数据类型是最常用的数据类型之一。MySQL 支持的整数又分为 TINYINT、SMALLINT、MEDIUMINT、INT 和 BIGINT 五种，如表 4-2 所示。

表 4-2 整数数据类型

类型名称	字节数	无符号数的取值范围	有符号数的取值范围
TINYINT	1	0~255	-128~127
SMALLINT	2	0~65535	-32768~32767
MEDIUMINT	3	0~ $2^{24}-1$	$-2^{23}~2^{23}-1$
INT	4	0~ $2^{32}-1$	$-2^{31}~2^{31}-1$
BIGINT	8	0~ $2^{64}-1$	$-2^{63}~2^{63}-1$

从表 4-2 中可以看出，不同整数类型所占用的字节数和取值范围都是不同的，其中占用字节数最小的是 TINYINT，占用字节数最大的是 BIGINT。需要注意的是，不同整数类型的取值范围可以根据字节数计算出来。

(2) 在 MySQL 数据库中，存储的小数都是使用浮点数和定点数来表示的。浮点数的类型有两种，分别是单精度浮点数类型（FLOAT）和双精度浮点数类型（DOUBLE）。而定点数类型只有 DECIMAL 类型，如表 4-3 所示。

表 4-3 浮点数和定点数类型

类型名称	字节数	有符号的取值范围	无符号的取值范围
FLOAT	4	$-3.40E+38~-1.17E-38$	0 和 $1.17E-38~3.40E+38$
DOUBLE	8	$-1.79E+308~2.22E-308$	0 和 $2.22E-308~1.79E+308$
DECIMAL(M,D)	M+2	$-1.79E+308~2.22E-308$	0 和 $2.22E-308~1.79E+308$

从表 4-3 中可以看出，DECIMAL 类型的取值范围与 DOUBLE 类型相同。需要注意的是 DECIMAL 类型的有效取值范围是由 M 和 D 决定的，其中，M 表示的是数据的长度，D 表示的是小数点后的长度。比如，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。

(3) 二进制数据类型用于存储二进制数据，又分为 BINARY、VARBINARY 和 BLOB 三种，如表 4-4 所示。

表 4-4 二进制数据类型

类型名称	取值范围和说明
BINARY	BINARY (n), 表示定长的二进制数据, n 为长度, 占 n+4 个字节。如果数据长度不足 n, 将在数据的后面用“\0”补齐。
VARBINARY	VARBINARY (n), 表示变长的二进制数据, n 为长度, 占实际数据长度+4 个字节。
BOLB	可以用来存储数据量很大的二进制数据, 如图片、PDF 文档等, 根据二进制编码进行比较和排序。

(4) 字符串类型是使用最多的数据类型, 可以用它来存储各种字母、数字符号、特殊符号。一般情况下, 使用字符串类型数据时须在其前后加上单引号。字符串数据类型又细分为 CHAR 和 VARCHAR 两种, 如表 4-5 所示。

表 4-5 字符串类型

类型名称	取值范围和说明
CHAR	定义形式为: char[(n)], 表示长度为 n 个字节的固定长度且非 Unicode 的字符数据, n 必须是一个介于 1 和 8000 之间的数值, 存储大小为 n 个字节。
VARCHAR	定义形式为: varchar[(n)], 表示长度为 n 个字节的可变长度且非 Unicode 的字符数据。n 必须是一个介于 1 和 8000 之间的数值。存储大小为输入数据的字节的实际长度+1。

(5) 大文本数据类型用于存储大量的字符, 例如文章内容、评论等。TEXT 类型分为四种, 如表 4-6 所示。

表 4-6 文本数据类型

类型名称	取值范围
TINYTEXT	0~255 字节
TEXT	0~65535 字节
MEDIUMTEXT	0~16777215 字节
LONGTEXT	0~4294967295 字节

(6) 日期和时间数据类型用于存储日期和时间, 又细分为 YEAR、DATE、TIME、DATETIME 和 TIMESTAMP 五种, 如表 4-7 所示。

表 4-7 日期和时间数据类型

类型名称	字节数	取值范围	日期格式
YEAR	1	1901~2155	YYYY
DATE	4	1000-01-01~9999-12-31	YYYY-MM-DD
TIME	3	-838:59:59~838:59:59	HH:MM:SS
DATETIME	8	1000-01-01 00:00:00 ~9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS
TIMESTAMP	4	1970-01-01 00:00:01 ~2038-01-19 03:14:07	YYYY-MM-DD HH:MM:SS

(7) 其它数据类型: 除上面介绍的数据类型之外还有 ENUM、SET、BIT 三种数据类型, 其中 ENUM 类型又称为枚举类型, 定义 ENUM 类型的数据格式如下所示:

ENUM('值 1','值 2','值 3',..., '值 n')

在上述格式中, ('值 1','值 2','值 3',..., '值 n')称为枚举列表, ENUM 类型的数据只能从枚举列表中取, 并且只能取一个。需要注意的是, 枚举列表中的每个值都有一个顺序编号, MySQL 中存入的就是这个顺序编号, 而不是列表中的值。

SET 类型用于表示字符串对象，它的值可以有零个或多个，SET 类型数据的定义格式与 ENUM 类型类似，具体语法格式如下所示：

SET('值 1','值 2','值 3',..., '值 n')

与 ENUM 类型相同，('值 1','值 2','值 3',..., '值 n')列表中的每个值有一个顺序编号，MySQL 中存入的也是这个顺序编号，而不是列表中的值。

BIT 类型用于表示二进制数据。定义 BIT 类型的基本语法格式如下所示：

BIT(M)

在上述格式中，M 用于表示每个值的位数，范围为 1~64。需要注意的是，如果分配的 BIT(M)类型的数据长度小于 M，将在数据的左边用 0 补齐。例如，为 BIT(6)分配值 b'110'的效果与分配 b'000110'相同。

2. NULL 的含义

在现实世界中我们填写某些表格时，某些表项的内容不能确定或者没有必要说明时可以不填写。在 MySQL 中，我们用 NULL 表示数值未知的空值。要注意的是，空值不是“空白”或者“0”，没有两个空值是相等的，当把两个空值进行比较或将空值与任意数值进行比较时均返回未知的空值 NULL。因此，NULL 表示未知、不可用或将在以后添加的数据。

在 MySQL 设计表的各个字段（列）时，我们应该尽量避免 NULL 值，因为在数据统计等操作时有 NULL 数据的列可能会出错。我们用 NOT NULL 表示数据列不允许空值，这样就可以确保数据列必须包含有意义的数据，从而确保数据的完整性。

创建用户数据表

一、用户表实例说明

我们在“学生成绩管理”数据库下创建三个用户表，分别是学生表、课程表和成绩表。三个表的基本内容如下表 4-10、表 4-11 和表 4-12 所示。

表 4-10 学生表

列名	类型	长度	说明
学号	char	12	关键字（主键）
姓名	char	8	NOT NULL
性别	char	2	NOT NULL
出生年月	datetime	4	
所在系	varchar	30	

表 4-11 课程表

列名	类型	长度	说明
课程号	char	8	关键字（主键）
课程名	varchar	30	NOT NULL
学分	smallint	2	NOT NULL
任课教师	char	8	

表 4-12 成绩表

列名	类型	长度	说明
学号	char	12	关键字（主键）
课程号	char	8	
成绩	int	4	

二、创建用户表过程说明

下面介绍使用“Navicat for MySQL”工具交互式创建用户表的步骤

(1) 启动“Navicat for MySQL”，双击连接名“localhost_3306”建立连接，在“连接”面板中双击“学生成绩管理”数据库，鼠标右键单击其下的“表”节点，在打开的菜单中选择“新建表”菜单项，如图 4-1 所示，打开“新建表”面板。

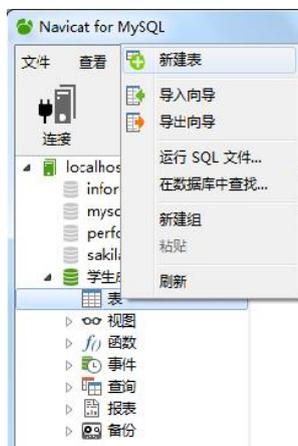


图 4-1 选择“新建表”命令

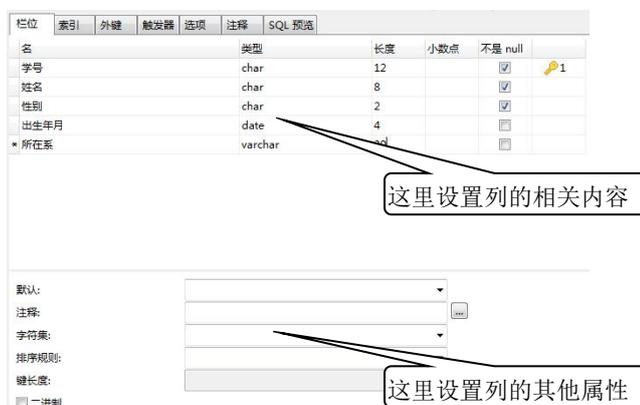


图 4-2 “新建表”面板

(2) 在打开的“新建表”面板中，在其“栏位”选项卡的“名”列下输入表中各字段列的名称，在“类型”列下拉列表框中选择字段列对应的数据类型，在“长度”列下输入字段的长度，在“小数点”列指定小数的位数，在“不是 null”列下单击复选框以确定该列中的数据是否允许为空值 NULL，如图 4-2 所示。

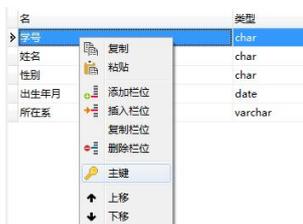


图 4-3 设置主键

名	类型	长度	小数点	不是 null	
学号	char	12	0	<input checked="" type="checkbox"/>	1
姓名	char	8	0	<input checked="" type="checkbox"/>	
性别	char	2	0	<input checked="" type="checkbox"/>	
出生年月	date	0	0	<input type="checkbox"/>	
所在系	varchar	30	0	<input type="checkbox"/>	

图 4-4 定义“学生表”

(3) 依次定义表中各列，定义完毕后可为表指定主键。选择表的某一列（复合主键按顺序依次操作），单击鼠标右键，在打开的菜单中选择“主键”命令，如图 4-3 所示。也可使用工具栏中的主键工具“ 主键”来设置表的主键。“学生表”的所有列定义完毕后，如图 4-4 所示。

(4) 表的所有列定义完毕后可在表的属性面板中定义表的名称和所有者，也可单击工具栏中的“保存”工具，在弹出的“表名”对话框中输入表名称“学生表”后单击“确定”按钮，即保存表的设置，如图 4-5 所示。



图 4-5 “表名”对话框

【自己动手试一试】

请同学们依照上面介绍的方法建立“学生表”、“课程表”和“成绩表”，其中“课程

表”和“成绩表”的定义如下图 4-6 和 4-7 所示。

名	类型	长度	小数点	不是 null	
课程号	char	8	0	<input checked="" type="checkbox"/>	1
课程名	varchar	30	0	<input checked="" type="checkbox"/>	
学分	smallint	2	0	<input checked="" type="checkbox"/>	
任课教师	char	8	0	<input type="checkbox"/>	

图 4-6 “课程表”定义

名	类型	长度	小数点	不是 null	
学号	char	12	0	<input checked="" type="checkbox"/>	1
课程号	char	8	0	<input checked="" type="checkbox"/>	2
成绩	int	4	0	<input type="checkbox"/>	

图 4-7 “成绩表”定义

三、表中设置自动递增列

MySQL 数据表中的自动递增列相当于 Access 数据表中的自动增长列。MySQL 数据表中自动递增列的设置是在表的定义时完成的，如果表中某列要定义为自动递增列，则这个列（字段）的数据类型必须是整数型的，一般设置为整型（int）或大整型（bigint），而不能是字符类型或其它类型。

默认情况下，如果某列设置为自动递增列，则该字段的值从 1 开始自增。下面介绍标识列的设置方法。

对于前面介绍的表 4-1 “教工信息表”，其中的教工号可以定义为自动增加列。

- （1）在教工信息表的设计表状态下选中“教工号”列（字段）；
- （2）勾选其下的“自动递增”复选框，如图 4-8 所示；
- （3）设置完成后单击“保存”按钮完成设置。

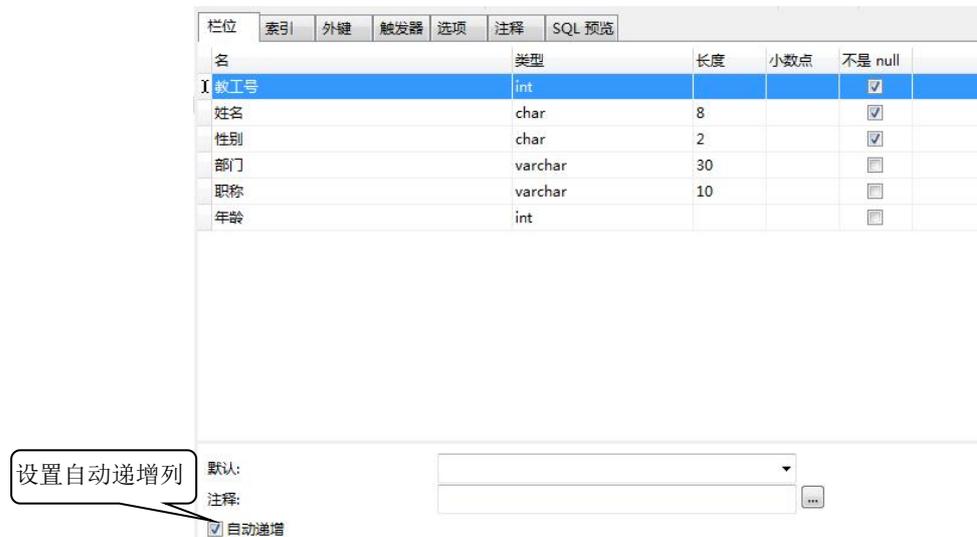


图 4-8 自动递增列的设置

标识列一般可作为主键使用，因为在同一个表中不会有相同的两个标识值存在；使用时标识列也不需要用户输入具体的值，是由系统根据标识种子和标识增量来自动产生。

四、修改用户表结构

已经建立的用户表，如果发现不符合要求，例如某列的类型不合适、列的长度需要增大或缩小、需要增加列或修改列的约束，还有某些列不再需要，想要删除等，这些都可以进行修改。我们这一节主要介绍用户表结构的修改。

我们介绍在“Navicat for MySQL”工具中修改用户表结构的基本方法。

- （1）启动“Navicat for MySQL”，双击连接名“localhost_3306”建立连接，在“连接”面板中双击“学生成绩管理”数据库，展开“表”节点，鼠标右键单击其下需要修改结构的

表名“学生表”，在打开的菜单中选择“设计表”菜单项，如图 4-9 所示。

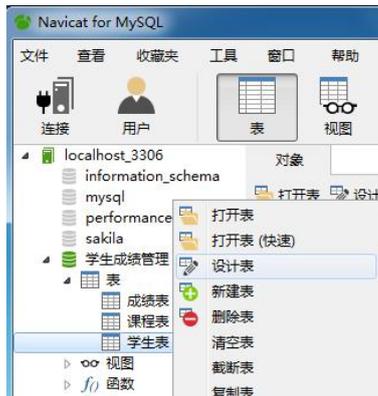


图 4-9 选择“设计表”命令

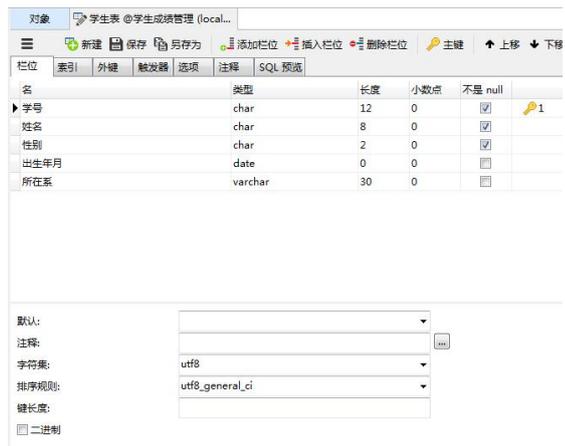


图 4-10 “学生表设计”对话框

(2) 在随后打开的学生表设计内容窗格中，可对表的各列的列名、数据类型、是否允许为空和其它属性进行修改，如图 4-10 所示；

(3) 如果想往表中插入新的列，可在学生表设计内容窗格中单击“添加栏位”命令，然后输入列名，定义列的类型、相关属性和列是否设置为空等内容；如果想删除某列，在选中的列上单击“删除栏位”命令，然后在弹出的“确认删除”对话框中选择“删除”命令即可，如图 4-11 所示。

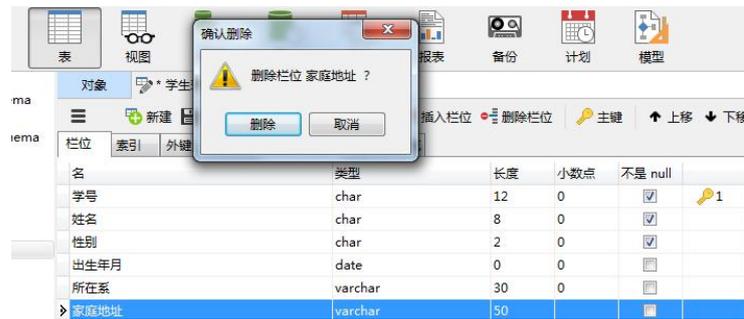


图 4-11 确认删除栏位对话框

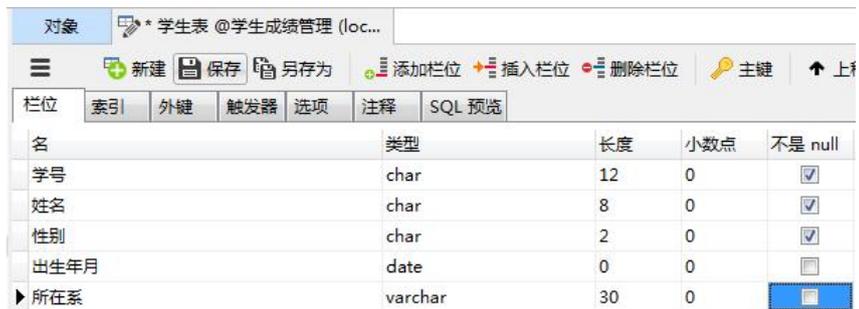


图 4-12 保存“学生表”

(4) 全部修改完成后，单击表设计内容窗格中的“保存”按钮如图 4-12 所示，完成表结构的修改。

修改表的结构很简单，但一般应该在往表中输入数据前修改表的结构，否则会影响现有数据的存储或违反相应约束规定而导致现有数据的损坏。

五、删除用户表

如果数据库中有不需要的数据表，可以将其删除，以便释放其所占有的空间。但删除的表是不能再恢复的，所以在删除表时一定要小心确认。

下面我们介绍使用“Navicat for MySQL”工具删除用户表的基本方法。

假设我们已经在“学生成绩管理”数据库中建立了一个表“学生表备份”，这个表不再需要，想从数据库中删除它，其方法如下。

1. 启动“Navicat for MySQL”，双击连接名“localhost_3306”建立连接，在“连接”面板中双击“学生成绩管理”数据库，展开“表”节点，鼠标右键单击其下的“学生表_copy”表节点，在打开的菜单中选择“删除表”菜单命令。

2. 如果表建立有外码，则必须首先删除外码，然后才能删除表。对于已删除的用户表，在“Navicat for MySQL”，工具中不再存在。另外请注意：不能删除当前正在使用的表；也不要试图删除系统表。

任务三 数据完整性与表中约束的建立

一、数据完整性基本概念

数据库中的数据（存放在表中）应该防止输入不符合语义的错误数据，而始终保持其中数据的正确性、一致性和有效性，这就是我们这节要介绍的数据完整性。数据完整性是衡量数据库质量好坏的标准之一，MySQL 数据库提供了完善的数据完整性机制。我们这一节首先介绍数据完整性的基本概念，然后用实例说明在 MySQL 中约束和默认对象等的创建和管理。

1. 什么是数据完整性

数据完整性是指数据的正确性、一致性和有效性，是指数据库中不应该存在不符合语义的数据。所谓正确性是指数据表中的数据应该是正确的，比如学生选修某门课程的成绩应该是 85 分，但不小心输入成 185 分，数据库管理系统应该能够检测出来并指出错误。一致性是指数据库中各个表中的数据应该相互照应的，比如学生表中一个学生的学号和这个学生在成绩表中的学号应该对应一致，在学生表中不存在的学号在成绩表中不应该有相应的选课记录等；而有效性是指数据应该是合法有效的，比如学生的性别应该是“男”或“女”，而不能是其它值。

数据完整性可分为四种类型：实体完整性、域完整性、参照完整性和用户定义完整性。

实体完整性是指任何一个实体（对应表中的一行或一条记录）都有区别于其它实体的特征。比如世界上没有完全相同的两个人（实体），对应到数据表中每个人对应一条记录其中的编号（或者说身份证号）应该是不同的。

域完整性是指表中每列的数据应该具有正确的数据类型、格式和有效的数值范围。

参照完整性是指在两个表的主键和外键之间数据的一致。其含义在三：一是保证被参照表和参照表之间数据的一致；二是防止数据丢失或者无意义的数；最后可以禁止在从表（例如“学生成绩管理”数据库中的“成绩表”，其“学号”列是“学生表”的外码）中插入被参照表（“学生成绩管理”数据库中的“学生表”）中不存在的关键字的记录。

用户定义完整性是用户希望定义的数据的完整性。例如电话号码是 8 位的数字码，邮政编码为 6 位数字码，对学生出生日期范围的限制等。

2. MySQL 如何实现数据完整性

MySQL 中提供的用来实施数据完整性的途径主要有创建表时的列级约束或表级约束、默认 (Default)、触发器 (Trigger)、数据类型 (Data Type)、索引 (Index) 和存储过程 (Stored Procedure) 等, 表 4-13 中列出了 MySQL 实施完整性的主要途径。

表 4-13 MySQL 实施完整性的主要途径

数据完整性类型	实施途径
实体完整性	Primary Key (主键) UNIQUE 约束 索引
域完整性	Default (默认) Foreign Key (外键) Data Type (数据类型) NOT NULL (非空)
参照完整性	Foreign Key (外键) Trigger (触发器) Stored Procedure (存储过程)
用户定义的完整性	列级约束和表级约束 (Create Table) Trigger (触发器) Stored Procedure (存储过程)

二、创建和管理约束

1. 什么是约束

约束 (Constraint) 定义了列允许的取值, 是强制完整性的标准机制, MySQL 中的约束机制包括以下 5 种。

(1) NOT NULL 约束 (非空约束): 指定数据列不接受空值 (NULL)。

(2) PRIMARY KEY 约束 (主键约束): 列或列的组合, 在一个表中不允许有两行记录包含同样的主键值。

(3) UNIQUE 约束 (惟一约束): 在列或列的组合内强制执行值的惟一性。

(4) FOREIGN KEY 约束 (外键约束): 一个表的外键指向另一个表的主键, 当一个外键没有与之对应的主键值时可阻止其插入或修改。

(5) 默认值约束: 当不给一个记录中的某列分量输入值时则采用由 <常量表达式>所提供的值, 整数、浮点数、用单引号括起来的字符串或日期都是常量表达式。

2. 创建非空约束

在 “Navicat for MySQL” 工具中创建表时建立非空约束的方法如下:

在 “Navicat for MySQL” 工具中创建表时, 在某个字段列定义时, 在其 “不是 null” 列下的复选框中单击添加 “√”, 即建立了列的 “非空约束”, 如图 4-13 所示。



图 4-13 创建表时建立非空约束

3. 创建 PRIMARY KEY 约束 (主键约束)

主键约束是用来实现实体完整性的约束, 表中定义了主键后, 就不允许有相同主键的两个记录存在。

在 “Navicat for MySQL” 工具中创建主键约束的方法如下:

在“Navicat for MySQL”工具中创建表时，选中要设为主键的列或列的组合（如果是多个列组合为主键，按顺序为每个列添加主键），单击鼠标右键，在弹出的菜单中选择“主键”命令，如图 4-14 所示。也可单击工具栏中的钥匙形状的主键工具来设置表的主键。

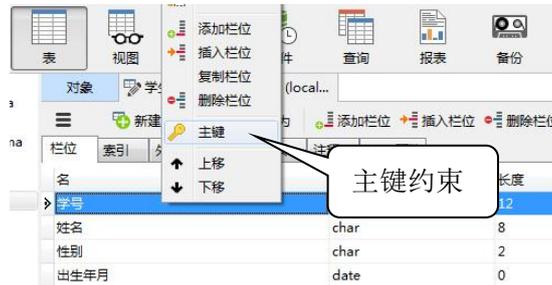


图 4-14 定义表的主键

4. 创建 UNIQUE 约束（惟一约束）

有需要时可以为非主键列创建惟一约束，以确定指定列不允许有相同的值。创建惟一约束时自动创建索引。

在“Navicat for MySQL”工具中创建惟一约束的方法如下：

在“Navicat for MySQL”工具中创建表时，选中“索引”选项卡来进行设置。

在索引内容栏中单击“名”文本框，输入索引的名称。选中“栏位”文本框，单击右边的 按钮弹出“选择栏位”对话框如图 4-15 所示，选择要创建唯一约束的列，单击“确定”按钮。在索引类型的下拉列中选择“Unique”，在索引方法的下拉列中选择“BTREE”，然后单击“保存”按钮，如图 4-16 所示。

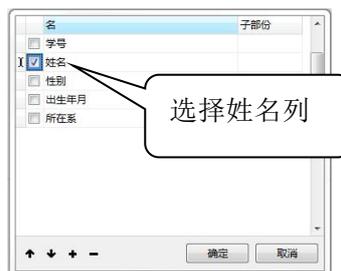


图 4-15 “选择栏位”对话框

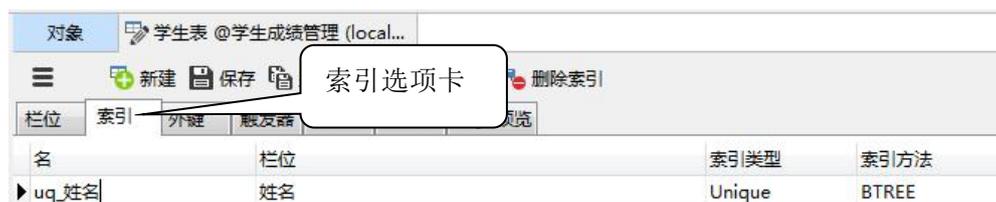


图 4-16 “索引”对话框

5. 创建 FOREIGN KEY 约束（外键约束）

在“Navicat for MySQL”工具中创建外键约束的方法如下：

在表设计中单击“外键”选项卡，打开“外键”对话框，在其中显示已有的外键关系，单击其中的“添加外键”命令按钮添加新的外键关系，单击“删除外键”按钮可删除已建立的选中的外键关系。



图 4-17 “外键”对话框

在“外键”对话框中首先定义外键名，选中“栏位”文本框，单击右边的按钮弹出“选择栏位”对话框，选择要创建外键约束的列，单击“确定”按钮。在参考数据库的下拉列中选择当前数据库名，被参考表的下拉列中设定主键表，选中“参考栏位”文本框，单击右边的按钮弹出“选择栏位”对话框，选择主键列的名字，单击“确定”按钮。在删除时的下拉列中选择“RESTRICT”，在更新时的下拉列中选择“RESTRICT”，然后单击“保存”按钮，即建立了外键约束，如图 4-17 所示。

6. 创建默认值约束

在 MySQL 中，建立用户表时可定义列的默认值。例如，将“学生表”的“性别”列定义默认值为“男”，在“Navicat for MySQL”工具中创建默认值约束方法如下：

- ① 在表设计窗格中单击选择“性别”列；
- ② 在其下面的“属性”窗格中的“默认”栏中输入字符‘男’，如图 4-18 所示；
- ③ 关闭表设计窗口，保存设置即可。



图 4-18 “默认约束”对话框

三、建立用户表之间的关系图

数据库中的用户表往往不是孤立的，而是相互关联的。前面我们已经介绍了如何在表中建立外码约束，其实外码约束就是表之间的列建立的一种联系，以保持数据之间的一致性或者说参照性。

数据库有一个称为“模型”的逻辑对象，其作用是以图形方式显示通过数据连接选择的表或表结构化对象，同时也显示它们之间的联接关系。

下面我们介绍在“学生成绩管理”数据库中使用“Navicat for MySQL”工具建立关系图的方法。

(1) 启动“Navicat for MySQL”，双击连接名“localhost_3306”建立连接，在“连接”面板中双击“学生成绩管理”数据库，鼠标右键单击“学生成绩管理”数据库，在打开的菜单中选择“逆向数据库到模型”菜单项，如图 4-19 所示，生成的模型如图 4-20 所示。

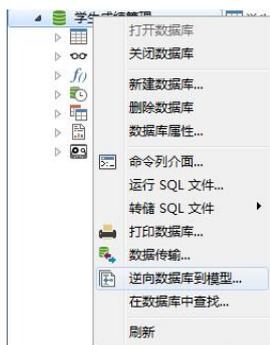


图 4-19 “逆向数据库到模型”菜单命令

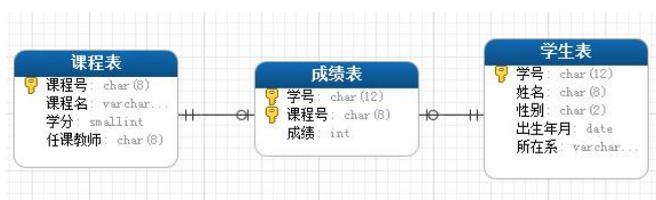


图 4-20 “学生成绩管理”关系模型

(2) 数据模型生成后，就单击“模型”对话框的“保存”按钮，弹出如图 4-21 所示的“模型名”对话框，在其中输入数据模型的名称，单击“确定”按钮，完成数据库模型的建立。



图 4-21 “模型名”对话框

数据库模型建立好后，数据库的表间约束将防止一些不正确数据的插入。比如在学生成绩管理数据库中成绩表中的学号和课程号必须是学生表和课程表中已经存在的学号和课程号。

建立数据库关系图一般应该在往表中输入数据之前建立，否则如果数据表中有数据违反外码约束将会出现错误，导致关系图创建失败。

任务四 查看和编辑数据表中的记录

创建好用户表结构之后，表内是空的，没有数据记录。我们在这一任务中介绍在已经建立好表结构的表中如何插入、修改、删除和查看数据记录。

一、在 Navicat for MySQL 中插入记录

启动“Navicat for MySQL”，双击连接名“localhost_3306”建立连接，在“连接”面板中双击“学生成绩管理”数据库，展开“表”节点，鼠标右键单击其下需要插入记录的表名“学生表”，在打开的菜单中选择“打开表”菜单项，如图 4-22 所示。在这里可以输入新的记录内容，如图 4-23 所示，但输入新的记录内容时要注意以下几点。

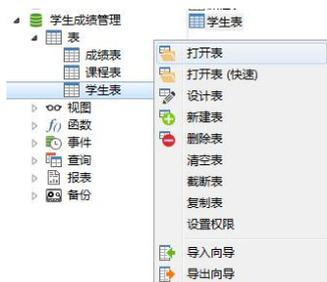


图 4-22 “打开表”菜单命令



图 4-23 在编辑状态下插入新记录

1. 输入记录中的每个字段内容时，要注意与表定义时该字段的类型相符、长度和精度

等相符，否则会出现警告框，且整个记录无效，不能保存到数据表中；

2. 如果定义表时某列的值不能为空（NULL），则必须要输入内容，否则也会出现警告框，告诉你表的哪列数据不能为空；

3. 如果字段是外键，则在主表中不存在的值不能在外键表中出现，否则也会出现警告信息，记录插入无效；

4. 如果表中某列已经定义了默认值，在输入记录时使用默认值则不要在对应的列内输入任何内容，保存这条记录时这个列自动取你设定的默认值；

5. 在输入完一条记录后，单击下方的  按钮插入下一条记录，数据插入完成后单击下方的  按钮保存符合条件的记录。

二、在 Navicat for MySQL 中更新记录

在打开的“学生表”结果窗体中用鼠标单击要更新的记录行中某个字段单元格，即可以编辑此单元格的内容，修改一个字段的值后可以继续修改本记录的其它字段值，当一行的所有修改完成后单击下方的  按钮，保存修改后的数据。

三、在 Navicat for MySQL 中删除记录

在打开的“学生表”结果窗体中用鼠标单击选中要删除的记录行，用鼠标右击该行记录，在弹出的菜单中选择“删除记录”命令菜单项，如图 4-24 所示，则会出现如图 4-25 所示的删除记录确认对话框，单击“是”则删除所选的记录，单击“否”则删除失败。

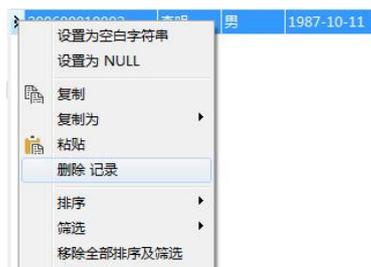


图 4-24 在编辑状态下删除记录



图 4-25 “确认删除”对话框

删除表中已有记录行时要注意以下几点：

1. 记录删除之后是不能恢复的，即删除是永久的，删除之前一定要确认无误；
2. 一次可以删除多条记录，按住“Shift”或“Ctrl”键，可以选择多条记录删除；
3. 在选好记录后，也可以按“Delete”键删除记录；

4. 如果删除的记录是其他表的外键指向，删除操作可能会影响另外一个表即外键表。例如在学生表中删除一个学生记录时，因为学生表的学号是主键，而成绩表中的学号是外键，则这个学生的所有成绩记录有可能也被删除了，这要看外键定义时的具体情况而定。

四、在 Navicat for MySQL 中查看记录

启动“Navicat for MySQL”，双击连接名“localhost_3306”建立连接，在“连接”面板中双击“学生成绩管理”数据库，展开“表”节点，鼠标右键单击其下的“学生表”，在打开的菜单中选择“打开表”菜单项，即可查看“学生表”中的所有记录。