第3章 操作数据库和表

数据库的功能是管理数据,这些数据必须被存放在数据库中才能够对它们进行管理。 而表是存储数据的数据库对象。数据库中有很多种类型的对象,如表、视图、存储过程、 触发器等。

数据库和表是数据管理的基础,本章介绍数据库和表的相关操作,包括数据库和表的概念、创建和对数据库和表的管理等。

本章学习要点:

- □ 理解数据库和数据表的概念
- □ 掌握数据库的两种创建方式
- □ 掌握表的两种创建方式
- □ 熟悉数据库相关查询
- □ 掌握数据库的修改和删除
- □ 掌握数据表相关查询
- □ 掌握数据表的删除
- □ 掌握字段的添加
- □ 掌握字段的修改
- □ 掌握字段的删除

3.1 数据库和表概述

首先要了解什么是数据库和表。虽然通过本书的前两章,读者对数据库有了一些了 解,本章将补充介绍数据库和表的概念。

_--3.1.1 数据库概述 --、

本书前两章已经介绍了数据库系统。而 MySQL 数据库有着一套字符集和字符校对 规则,不同的字符集用于处理不同的数据。

一个数据库服务器可以有多个数据库,不同的是可以具有不同的字符集和校对规则。 因此在创建数据库时,需要为数据库选择字符集和校对规则。

字符集是一套符号和编码,校对规则是在字符集内用于比较字符的一套规则。 MySQL 能够支持多种字符集,可以通过执行如下语句查看可用的字符集:

SHOW CHARACTER SET;

上述代码的执行效果如图 3-1 所示。

- MySQL 数据库管理与开发实践教程

Result Grid 🛛 🛄	Filter Rows:	Export: 🔛 Wrap	Cell Content: 1A	
Charset	Description	Default collation	Maxlen	
big5	Big5 Traditional Chinese	big5_chinese_ci	2	
dec8	DEC West European	dec8_swedish_ci	1	
cp850	DOS West European	cp850_general_ci	1	
hp8	HP West European	hp8_english_ci	1	
koi8r	KOI8-R Relcom Russian	koi8r_general_ci	1	
latin 1	cp1252 West European	latin1_swedish_ci	1	
latin2	ISO 8859-2 Central European	latin2_general_ci	1	
swe7	7bit Swedish	swe7_swedish_ci	1	
uery Output	Result 9 ×			Read Or

● 图 3-1 当前可用的字符集

在图 3-1 中, Charset 表示字符集, Description 表示描述内容, Default collation 表示 默认的校对规则。任何一个给定的字符集至少有一个校对规则,它还可能有几个校对规则。可以使用以下语句查看校对规则,语句如下:

SHOW COLLATION [LIKE 'pattern'];

直接使用上述语句 SHOW COLLATION 时查看校对规则列表,如果想要查看 latin1 (西欧 ISO-8859-1)字符集的校对规则,可以在 SHOW COLLATION 后添加 LIKE 查询。 查看校对规则列表的效果如图 3-2 所示。

esult Grid 📗	Filter Rows:		Ex	port: 📙 Wri	ap Cell Content	. <u>∓⊼</u> [
Collation	Charset	Id	Default	Compiled	Sortlen	
big5_chine	big5	1	Yes	Yes	1	
big5_bin	big5	84		Yes	1	
dec8_swe	dec8	3	Yes	Yes	1	
dec8_bin	dec8	69		Yes	1	
cp850_gen	cp850	4	Yes	Yes	1	
cp850_bin	cp850	80		Yes	1	
hp8_englis	hp8	6	Yes	Yes	1	
hp8_bin	hp8	72		Yes	1	
ery Output	Result 10 >	<				Read Only

● 图 3-2 校对规则列表

校对规则不同,它所代表的含义也有所不同。所有的校对规则都有一些共同的特征,如下所示。

(1) 两个不同的字符集不能有相同的校对规则。

(2)每个字符集有一个默认校对规则。

存在校对规则命名约定,它们以其相关的字符集名开始,通常包括一个语言名,并 且以_ci(大小写不敏感)、_cs(大小写敏感)或_bin(二元)结束。

●--3.1.2 表概述 -->

表是数据库最基本的组成对象,是数据库的实体,用来组织和存储数据。生活中人 们接触到的有各种表,如面试时填写的基本信息表,在考勤时候有考勤表,在发工资时 有工资表等。

数据库中的表与生活中的表是一样的,有着表头和数据。不过数据库中的表,表头

被定义为字段,是表的列;每一行存储一条记录。如学生信息表如表 3-1 所示。

	表 3-1	_ 学生信息表
--	-------	---------

姓名	性别	年龄	籍贯	
段林	男	12	河南郑州	
王淼	女	12	河南开封	

如表 3-1 所示,这是生活中常见的表,有着表头和数据。而将其定义为数据库中的 表,那么表头的姓名、性别、年龄和籍贯将用作表的列(字段);两个学生的信息是表的 行,该表中有两条数据。

列也叫作表的字段,每个字段都有着指定的数据类型。在对数据进行操作时,数据 库系统将根据字段的数据类型对数据进行操作。数据类型在本书的第2章中已介绍过。

MySQL 数据库在安装后有着系统自带的表,如查询系统数据库 sakila 中 actor 表的数据,其效果如图 3-3 所示。

Qu	ery 1 my	rsql3* ×										
C	- 🛛 🖉	f 🖗 🕐	1 🚯 🖽 1 🥑) 🛛 📓 🚽	0	1 7						
	1 • US 2 • SE 3	E sakila; ELECT * FROM	actor;									
•		III										4
Res	sult Grid 🛛 🚺	Filter Rows:		Edit: 🔏	ð 🖬	Export/Import:	5	Wrap Ce	ell Content:	<u>Ŧ</u> Ā		
	actor_id	first_name	last_name	last_update								^
	1	PENELOPE	GUINESS	2006-02-15 0								
	2	NICK	WAHLBERG	2006-02-15 0								
	3	ED	CHASE	2006-02-15 0								
	4	JENNIFER	DAVIS	2006-02-15 0								
	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 0								
	6	BETTE	NICHOLSON	2006-02-15 0								
	7	GRACE	MOSTEL	2006-02-15 0								
	8	MATTHEW	JOHANSSON	2006-02-15 0								
	9	JOE	SWANK	2006-02-15 0								
	10	CHRISTIAN	GABLE	2006-02-15 0								
acte	or 1 x								Apply		ancel	-

▲ 图 3-3 sakila 数据库中的 actor 表

3.2 创建数据库

对数据库的使用是从数据库和表的创建开始的。本节介绍数据库的创建,有三种方式:使用 MySQL Workbench 创建数据库;新建查询窗口使用 SQL 语句创建数据库;在 控制台使用 SQL 语句创建数据库。本节介绍如何使用 MySQL Workbench 和 SQL 语句创建数据库。

●--3.2.1 MySQL Workbench 创建数据库 --->

MySQL 5.6.19 版本下, MySQL 的操作可以直接在 MySQL Workbench 中进行。首先 登录数据库进入如图 3-4 所示的界面。

MySQL Workbench				
Local instance MySQL56 ×			same a subbridge	
File Edit View Query Database Server	Tools Scripting	Help		
	1)			Ø — —
Navigator	Query 1	fruit ne	ew_table new_schema - Schem	na ×
SCHEMAS Q Filter objects	**	Name: new_	schema Rename References	The name of the schema. It is recommended to use only Refactor model, changing all references found in view,
 fruitshop information new_schem performantion sakila shop test world Schema Schemas 	· · ·	Collation: Server big5- big5- big5- big5- big5- big5- big5- big5- big5- big5- big5- big5- cessed ces	r Default r Default collation r Default collation - deca3, pin - default collation -	Specifies which charset/collations the schema's tables will
	Schema	latin1 latin2	- latin1_spanish_ci - default collation	
		latin2 latin2 latin2	- latin2_czech_cs - latin2_general_ci - latin2_hungarian_ci	Apply Revert
Object Info Session	Output :::::	latin2	- latin2_croatian_ci 🔹 🔻	

图 3-4 创建数据库

如图 3-4 所示,在右侧数据库列表中,右击任意一个数据库名称,弹出菜单如图 3-4 所示。选择 Create Schema 选项即可打开创建数据库的界面。

图 3-4 右侧有创建数据库相关数据,包括数据库名称填写和数据库校对规则的选择。 设置好数据后单击 Apply 按钮即可打开执行对话框,如图 3-5 所示。

Apply SQL Script to Database								
Review SQL Script Apply SQL Script	Review the SQL Script to be Applied on the Database							
	Online DDL Algorithm: Default Lock Type: Default CREATE SCHEMA `new_schema`; CREATE SCHEMA `new_schema`;							
	Back Apply Cancel							

图 3-5 执行数据库创建

如图 3-5 所示,该对话框中列出了在图 3-4 界面中设置的数据供用户确认。其中带 有行号的代码是执行数据库操作所使用的 SQL 语句。单击 Apply 按钮后,系统尝试执行 数据库的创建,弹出如图 3-6 所示的对话框来显示执行状态。

Apply SQL Script to Database		×
Review SQL Script Apply SQL Script	Applying SQL script to the database	
	The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.	
	Show Logs Back Finish Cancel	

● 图 3-6 操作执行状态

如图 3-6 所示。确认数据库创建已顺利执行后,在对话框中单击 Finish 按钮关闭对 话框,即可完成数据库的创建。



●--3.2.2 SQL 语句创建数据库-->

本节介绍在查询窗口中使用 SQL 语句创建数据库。由图 3-5 可以看出:在 MySQL Workbench 工具下创建数据库时,默认生成的语句是 CREATE SCHEMA。因此,也可以 直接在控制台中通过执行该语句进行创建。

CREATE SCHEMA 语句可以看作是创建架构,数据库的创建可以使用 CREATE DATABASE 语句。但 MySQL 数据库中 CREATE SCHEMA 语句和 CREATE DATABASE 语句的作用是一致的。

数据库在创建时需要设置数据库名称和字符集、校对规则。其语法格式如下:

```
CREATE DATABASE db_name
[[DEFAULT] CHARACTER SET charset_name]
[[DEFAULT] COLLATE collation name]
```

上述语法中,数据库字符集和校对规则的设置,与数据集和校对规则的分配方式如 下所示。

(1) 如果指定了 CHARACTER SET charset_name 和 COLLATE collation_name, 那么 采用字符集 charset_name 和校对规则 collation_name。

(2)如果指定了 CHARACTER SET charset_name 而没有指定 COLLATE collation_name,那么采用 CHARACTER SET charset_name 和 CHARACTER SET collation_name 的默认校对规则。

(3) 如果 CHARACTER SET charset_name 和 COLLATE collation_name 都没有指定,则采用服务器字符集和服务器校对规则。

数据库的创建语句可参考图 3-5, 这里不再举例。一个 MySQL 服务器上能够有多个

数据库,而这些数据库的字符集和校对规则可以不同。

3.3 管理数据库

数据库的管理包括数据库查看、数据库的修改和删除。而数据库的查看又可分为查 看当前服务器中的数据库列表和查看指定数据库的详细信息。本节根据查看数据库、修 改数据库和删除数据库来介绍数据库的管理。

●--3.3.1 查看数据库 --,

数据库的查看主要通过 SQL 语句来执行。用户也可以直接在 MySQL Workbench 左 侧看到数据库的列表,如图 3-4 所示。接下来分别介绍使用 SQL 语句查看当前服务器中的数据库列表和查看指定数据库的详细信息。

1. 查看当前服务器中的数据库列表

查看当前服务器中已经存在的数据库,使用 SHOW DATABASES 语句。该语句将列 举出来多个系统数据库和用户自定义数据库,其执行效果如下所示。

```
+ ----- +
| Database |
+ ----- +
| information_schema |
| mysql |
| new_schema |
| performance_schema |
| sakila |
| test |
| world |
+ ----- +
7 rows
```

上述执行效果的最后一行,说明当前服务器中一共有7行数据库数据,即有7个数据库。

一项 若要查看当前的数据库,可使用 SELECT DATABASE()语句。系统将根据当前的操作确 定当前使用的数据库。

2. 查看数据库详细信息

同样是 SHOW DATABASES 语句,添加 CREATE 关键字和数据库名称即可查询指 定数据库的详细信息,语法如下:

SHOW CREATE DATABASE 数据库名称;

如查询 sakila 数据库的详细信息,代码如下。

SHOW CREATE DATABASE sakila;

上述代码的执行效果如下所示。

●--3.3.2 修改数据库 --,

数据库的修改包括修改数据库的名称和字符集。InnoDB存储引擎的数据库是无法修改数据库名称的,而 MyISAM存储引擎的数据库只要修改 DATA 目录下的库名文件夹就可以改变数据库名称。各种存储引擎的数据库都可以修改其字符集的校对规则,有两种方式: MySQL Workbench 工具和通过 SQL 语句。

1. 使用 MySQL Workbench 工具修改数据库

使用 MySQL Workbench 工具修改数据库信息,步骤如下。

(1) 在需要修改的数据库名称处右击,选择 Alter Schema 选项,可进入数据库修改 界面。数据库修改界面与数据库添加的界面一样,只是当存储引擎是 InnoDB 时,数据 库名称是不可编辑状态。

(2)为数据库选择新的校对规则类型,单击 Apply 按钮后,系统弹出对话框显示数 据库的修改信息;单击 Apply 按钮执行数据库修改;在弹出的对话框中单击 Finish 按钮 完成数据库的修改。

2. 使用 SQL 语句修改数据库

使用 ALTER SCHEMA 语句可以修改数据库信息,修改后的数据使用 SET 关键字设置,语法如下:

ALTER SCHEMA '数据库名称' DEFAULT CHARACTER SET 修改后的校对规则;

上述代码中,如果数据库所需要修改的校对规则是一个系列的非默认规则,那么将 使用下面的语句进行修改。

ALTER SCHEMA '数据库名称' DEFAULT CHARACTER SET 校对规则系列 DEFAULT COLLATE 校对规则;

上述两句代码中,数据库名称两端不是单引号,而是反单引号,通常用来引用数据 库名称、表的名称和字段名称。该符号在使用时可以去掉,不影响执行效果。

如将数据库 new_schema 修改为 utf16 default collatio 校对规则,代码如下。

ALTER SCHEMA 'new schema' DEFAULT CHARACTER SET utf16 ;

而将 new schema 数据库修改为 utf16 bin 校对规则,代码如下。

```
ALTER SCHEMA 'new schema' DEFAULT CHARACTER SET utf16 DEFAULT COLLATE
utf16 bin;
```

- 3.3.3 删除数据库 ---

删除数据库是指删除数据库系统中已经存在的数据库,删除数据库成功后,原来分 配的空间将被收回。数据库的删除可以在 MySQL Workbench 中进行,也可以使用 SQL 语句进行。

1. 使用 MySQL Workbench 工具删除数据库

数据库的删除与创建在 MySQL Workbench 工具下的操作略有不同。从图 3-4 可以看 出,右击数据库名称,弹出菜单中有 Drop Schema 选项。选择该选项即可进入如图 3-7 所示的对话框,可选择查看 SQL 语句或选择直接删除。

若选择 Drop Now 则直接删除该数据库,没有任何的提示。若选择 Review SQL 则显 示删除语句,如图 3-8 所示,单击 Execute 按钮即可执行数据库的删除,没有任何的提示。

MySQL V	Workbench X
1	Drop Schema Please confirm permanent deletion of schema `new_s` and all its data.
	→ Review SQL
	Drop Now
	取消

Review SC	QL Code to Execute		
Review the S	QL code to be executed.		
1	DROP DATABASE 'new_schema';		
•		Cancel	► Execute

💽 图 3-8 查看 SQL 删除语句

2. 使用 SQL 语句删除数据库

图 3-7 删除选项

DROP DATABASE 语句可以删除指定的数据库,在该语句后添加数据库名称的参 数。语法如下:

DROP DATABASE 数据库名称;

如图 3-8 所示,界面中的 SQL 语句是删除 new schema 数据库的语句。

计									
エ 思 い	开发者。	在删除数据库	时会删除该	、 数据库 1	中所有的表	和所有数	据,因此	1, 删除数:	据库时需
要慎	重考虑。	如果确定要	删除某一个。	数据库,	可以先将	该数据库备	\ 份,然	后再进行册	刖除。

66

3.4 创建数据表

数据表是存储数据的,数据有多种类型,而一个数据库表可以存储多种不同类型的数据,因此其创建方式将比数据库复杂。本节介绍数据表的创建,分别使用 MySQL Workbench 和 SQL 语句。

●- 3.4.1 MySQL Workbench 创建数据表 ---,

数据表是在数据库的基础上创建的,因此在创建时需要指出数据表所在的数据库。 在 MySQL Workbench 中创建表,步骤如下。

(1) 展开该表所在的数据库,在 Tables 节点处右击,选择 Create Table 命令,如图 3-9 所示。



图 3-9 创建数据表

(2) 在选择了 Create Table 命令后可打开表的创建界面,如图 3-10 所示。

U	Table Name: Collation:	Schema Default		•	Schem Engine	a: ::	shop InnoDB		•	2
	Comments:								*	
Column Name		Datatype	PK	NN	UQ	BIN	UN ZF	AI Default		
📍 id		INT	1	V						
name		VARCHAR(45)								
 price 		FLOAT								
Column Name:	price			Dat	a Type	: [LOAT			
Collation:	Table Default		Ŧ	(Default	:				
Comments:						l	Primary	📃 Not Null	📃 Uniqu	e
						1	Binary	Unsigne	📃 Zero	Fill
						l	Auto Inc	rement		
olumns Inde	xes Foreign	Keys Triggers Partitio	ning	Opti	ons					

图 3-10 创建表

如图 3-10 中向表添加了三个列 (字段),对其说明如下。

① 如图 3-10 所示,上部是表的基本信息,有表的名称、所属数据库、数据库类型、 存储引擎和表的说明这几个信息,除了所属数据库以外都是可编辑状态。若此时修改数 据库信息,如数据库的校对规则,那么在执行数据表创建的同时,将修改数据库信息。

② 中间的表格是数据表的列,向表中添加新的列,并为其设置数据类型和约束(将 在第4章中详细介绍)。

③ 图 3-10 中为表添加了三个字段,分别是有着主键约束和非空约束的 id 列,该列 是整型数据;长度可变的字符串类型的、没有任何约束的 name 列;浮点类型、没有任 何约束的 price 列。

④ 下面部分是对应列的具体属性,如图 3-10 中是现实的 price 列的属性,包括列的 名称、数据类型、校对规则、默认值、说明和约束信息。

(3) 如图 3-10 所示, 在设置好表的字段信息之后, 单击 Apply 按钮执行数据表创建; 接着在弹出对话框中单击 Apply 按钮确认执行 SQL 语句;最后在弹出的对话框中单击 Finish 按钮完成数据表的创建。

_--3.4.2 使用数据表模板 ---

由图 3-9 可以看出, Create Table Like 选项下有 MySQL 中定义好的数据表模板,可直接在需要的数据库中添加类似的数据表。

数据表模板定义了常用的表的字段,如会员信息表可以用在商场、超市、KTV、各 个品牌服装店等。那么若实现定义好会员信息表模板,之后这些地方在创建数据库并需 要创建会员信息表时,可直接根据模板添加会员信息表。

本节需要介绍的有两点知识:数据表模板的创建和使用数据表模板创建数据表。

1. 数据表模板的创建

如图 3-9 所示, Create Table Like 选项下有三个系统模板和一个 Edit Templates 选项, 选择 Edit Templates 选项即可创建数据表模板, 如范例 1 所示。

【范例1】

选择 Edit Templates 选项进入如图 3-11 所示的界面。界面右上角有 New Templates 按钮,单击在左侧列表中可新建表模板。双击新建的表模板名称,可对模板进行重命名,如图 3-11 所示是将新建的模板命名为 student。

界面的下方是对表模板字段的定义,其字段的添加方式与创建表时字段的添加方式 一样,可参考图 3-10 来添加模板表中的字段,如图 3-11 所示。

图 3-11 中向表中添加了 5 个字段,单击 Close 按钮完成数据表模板的创建。接着回 到主页面,随意展开一个数据库节点,右击 Tables 节点,如图 3-12 所示,可以看到刚才 新建的模板已经可以使用。

2. 数据表模板的使用

范例1创建了学生信息表的模板,这里在数据库 shop 中创建该模板的表,步骤如下。

(1) 首先展开 shop 数据库节点,选择 Create Table Like student 命令进入如图 3-13 所示的界面。





── 图 3-11 创建数据表模板

图 3-12 找出新建模板

-0-	Collation:	Schema Default					•) En	igine:	InnoDB		•	
	Comments:											*	
olumn Name		Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default			-
id 🕴		INT	1	1									
name		VARCHAR(45)			-								
ade		INT	100	H	H		H	-	m				
native		VARCHAR(20)											
olumn Name:							c	Data T	Type:				
Collation:						Ŧ		Def	fault:				
Comments:										Primary Key	Not Null	Unique	
										Binary	Unsigned	Zero Fill	
										Auto Increment			

图 3-13 使用学生信息表模板

(2)如图 3-13 所示,范例 1 创建的学生信息表模板中的字段,都在图 3-13 中被列举出来,省去了不少的步骤。接下来对表的定义与创建表时的步骤一样,可参考 3.4.1 节,这里不再详细说明。

●--3.4.3 SQL 语句创建数据表 --,

使用 SQL 语句创建数据表,需要指出表所在的数据库、表的名称,列举出表中每个 字段的字段名、字段的数据类型、是否为空、字段约束等,语法如下:

USE 数据库名称; CREATE TABLE 表的名称 (字段 1 的名称 字段 1 的类型 字段 1 的约束,字 段 2 的名称 字段 2 的类型 字段 2 的约束…);

或

CREATE TABLE 数据库名称.表的名称(字段1的名称字段1的类型字段1的约束,字段2的名称字段2的类型字段2的约束…);

上述第一条语句中,首先打开指定的数据库,接下来在该数据库下创建表。表的名称后面是小括号,括号内定义表的字段。每个字段的名称、类型和约束之间使用空格隔开,而两个字段之间使用逗号(,)隔开。

上述第二条语句中,直接创建指定数据库下的表,数据库与表名称之间使用圆点(.) 隔开。

如上述在 shop 数据库中创建 new_table 表,代码如下。

```
CREATE TABLE 'shop'.'new_table' (
  'id' INT NOT NULL,
  'name' VARCHAR(45) NULL,
  'price' FLOAT NULL,
  PRIMARY KEY ('id'));
```

3.5 管理数据表

表是有着结构和数据的,数据表的相关操作比数据库的操作要多,包括查看数据库 中的表、查看表的结构、添加表数据、查看表数据、修改表的定义、删除表等。本节介 绍数据表的管理。

●--3.5.1 查看数据表 --->

查看数据库包括查看指定数据库中的表和查看指定表的结构,下面从这两个方面详 细介绍表的查看。

1. 查看数据库中的表

查看数据库中的表,使用 SHOW TABLES 命令。该命令可查看指定数据库中的所有

表,如查看 sakila 数据库下的表,代码如下。

USE sakila;SHOW TABLES;

执行上述代码,其效果如图 3-14 所示。

Query 1 mysql3*	×
🗀 🖬 💋 🔗 🖞	ō. 🕑 🔀 🖽 📀 🙁 🖉 🛫 o. 👖 🖃
1 • USE sak	ila;SHOW TABLES;
•	
Result Grid 📗 Filter Ro	ows: Export: 📳 Wrap Cell Content: 🔢
Tables_in_sakila	A
actor	
actor_info	=
address	
category	
city	
country	
customer	
customer_list	
film	
Popult 2 xx	
Nesult 5 X	Read Only

图 3-14 sakila 中的表

2. 查看表结构

在控制台可以查看表的结构,即表的字段信息,其中包括:字段名、字段数据类型、 是否为主键、是否有默认值等。

查看表的结构可以使用简单查询和详细查询,分别使用 DESCRIBE/DESC 语句和 SHOW CREATE TABLE 语句。

使用 DESCRIBE/DESC 语句,语法如下:

DESCRIBE 表名;

或者简写为:

DESC 表名;

如查看 sakila 数据库中的 actor 表的结构,代码如下。

USE sakila;DESC actor;

上述代码的执行效果如图 3-15 所示。

●--3.5.2 添加表数据 -->

数据表是用来存储数据的,在字段创建之后即可向表中添加数据。向表中添加一条 数据相当于为表中对应的字段添加一个值,由于字段是有着数据类型和约束的,因此数 据的添加不能违背字段的数据类型和约束。

如一个整型字段不能添加一个字符型的数据;一个非空的字段不能够省略数据的添 加;一个允许为空的字段可以省略数据的添加;一个不能够重复的字段不能够添加重复

的值等。相关字段约束的内容将在第4章中介绍。

Resu	ult Grid 📗	Filter Rows:		Export:	Wrap Cell Content:	Ā
	Field	Туре	Null	Key	Default	Extra
•	actor_id	smallint(5) unsigned	NO	PRI	NULL	auto_increment
1	first_name	varchar(45)	NO		NULL	
	last_name	varchar(45)	NO	MUL	NULL	
1	last_update	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

图 3-15 actor 表的结构

数据添加可以在 MySQL Workbench 中进行,也可以使用 SQL 语句。

1. MySQL Workbench 中添加表数据

在 MySQL Workbench 中添加数据首先要找到需要添加数据的表,在表的名称处右击,如图 3-16 所示。

MySQL Workbench							E				
Local instance MySQL56 ×											
File Edit View Query Data	base Server Tools Scripti	ng	Help								
Navigator	Query 1	mj	/sql3* ×								
SCHEMAS	🚸 📲 📄 🔒	1 🛃	🜈 👰 🕛 😚	🗏 I 🔇		📓 🗹 🔍 🔳 🗔	2				
Q Filter objects	1 •	U	SE sakila;DESC act	or;							
► Tunctions	▲ ²										
▼ ➡ shop ▼ ➡ Tables	<							E.			
new_table	Select Bows - Limit 1000	¹ 📕	Filter Rows:		Export:	Wrap Cell Content:	<u>TA</u>				
Views Stored Procedures	Table Inspector		Туре	Null	Кеу	Default	Extra				
Functions	Com to Clinkows	-	smallint(5) unsigned	NO	PRI	NULL	auto_increment				
▼ 🗎 test	Send to SOL Editor	1e	varchar(45)	NO		NULL					
Views	County Table	e	varchar(45)	NO	MUL	NULL					
Stored Procedures	Create Table Like	ate	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIME	STAMP			
world	áltar Tabla										
Management Schemas	Table Maintenance										
T-G-marking	Drop Table										
Information	Truncate Table										
Table: new_table	Seerch Table Date										
Columns: id int(11) PK	D.C. 1 133							Read Only 🔋			
name varchar(45)	Vellezu MTT										
Object Info Session											

图 3-16 MySQL Workbench 中表的操作

如图 3-16 所示,右击表的名称后有弹出的对话框,显示可在 MySQL Workbench 中执行的表的操作,包括表的查询、复制、创建、修改、删除、刷新等操作。选择第一项 Select Rows 选项可查看当前表中的数据,如图 3-17 所示。

Que	ery 1	mysql3*	new_	table ×	🛞 援 🛫	<u>s</u> []]	-	
	1 •	SELEC	T * FROM	shop.new_tab	le;				
•		111]					
Res	ult Grid	🔢 🚷	Filter Rows:		Edit: 🛃 🗮	• 🖦	Export/Import:		Wrap Cell Conten
	id	name	price						
	1	apple	2.5						
	2	banana	3.5						
ı	3	orange	5						
*	NULL	NULL	NULL						
		_							
new	/_table	1• ×						Apply	Cancel
0	图 3	8–17 <u>₹</u>	查询表数:	据					

如图 3-17 所示,查询结果是处于可编辑状态的,直接在字段名称的下面填写相应的数据,即可实现数据的添加。图 3-17 中向表 shop.new_table 添加了三条数据。

数据编辑之后单击 Apply 按钮,即可打开执行对话框,单击 Apply 按钮执行数据的添加,接着在执行状态对话框中单击 Finish 按钮完成数据的添加和保存。

2. SQL 语句添加表数据

向表中添加数据需要指出需要使用的表和添加的数据:使用 INSERT INTO 语句指出 需要添加数据的表;使用 VALUES 语句指出需要添加的数据。

添加数据的语法将在第6章中详细介绍,这里只提供添加语句的例子。如同样是向表 shop.new_table 添加了三条数据,代码如下。

```
INSERT INTO 'shop'.'new_table' ('id', 'name', 'price') VALUES ('1',
'apple', '2.5');
INSERT INTO 'shop'. 'new_table' ('id', 'name', 'price') VALUES ('2',
'banana', '3.5');
INSERT INTO 'shop'. 'new_table' ('id', 'name', 'price') VALUES ('3',
'orange', '5');
```

●--3.5.3 修改表 --,

表的名称是可以修改的,对表指定一个有意义的名称是很有利于数据操作的。修改 表的名称也有两种方式,如下所示。

1. MySQL Workbench 中修改表信息

找到需要修改的表,在其名称处右击,如图 3-16 所示。选择 Alter Table 选项可打开 表的修改界面,如图 3-18 所示。

L.S	Collation			Scrier	na: (InnoDP			_]	1
•	Colladori.	utro - default colladori		_ Lingin	e. (THIODO			•	
	Comments:								*	
Column Name		Datatype	PK N	IN UQ	BIN	UN ZF	AI Def	ault		
📍 id		INT(11)	v	/						
🔷 name		VARCHAR(45)		/						
price		FLOAT					NUL	L		
Column Name:	name			ata Tvo	e: V	ARCHAR	45)			
Column Name:	name		D	ata Type	e: V/	ARCHAR(45)			
Column Name: Collation:	name Table Default		D	ata Type Defaul	e: Vi t:	ARCHAR(45)			
Column Name: Collation: Comments:	name Table Default		D	lata Type Defaul	e: V/ t:	ARCHAR(45)	t Null	Unique	
Column Name: Collation: Comments:	name Table Default			lata Type Defaul	e: Vi	ARCHAR(] Primary] Binary	45)	t Null	Unique	:
Column Name: Collation: Comments:	name Table Default		D	Jata Type Defaul	e: V/	ARCHAR(Primary Binary Auto In	45) V No Un crement	t Null signe	Unique	:

图 3-18 修改表 goods

如图 3-18 所示,界面中显示了该表的所有信息,每一项都处于可编辑状态,可直接 在界面中进行修改。如表的名称、数据库类型、校对规则、字段信息等,除了所属数据 库以外都是可编辑状态。若此时修改数据库信息,如数据库的校对规则,那么在执行数 据表修改的同时,将修改数据库信息。

修改之后单击 Apply 按钮,即可打开执行对话框,单击 Apply 按钮执行表的修改,接着在执行状态对话框中单击 Finish 按钮完成修改。

2. SQL 语句添加表名称

表的修改涉及很多,包括表定义的修改(所属数据库、表的名称、字段信息)和表数据的修改。表数据的修改将在第6章中介绍;字段的修改将在3.6.2节中介绍,这里介绍使用 SQL 语句修改表的名称,语法如下:

ALTER TABLE <旧表名> RENAME [TO] <新表名>;

如将 shop 数据库中的 goods 表的名称修改为 goodses,代码如下。

USE shop; ALTER TABLE goods RENAME goodses;

●--3.5.4 删除表 --,

表的删除可以直接在 MySQL Workbench 中如图 3-16 所示的界面选择 Drop Table 选项,即可打开提示对话框,如图 3-19 所示。

若选择 Drop Now 则直接删除该数据表,没有任何的提示。若选择 Review SQL 则显示删除语句如图 3-20 所示,单击 Execute 按钮即可执行数据表的删除,没有任何的提示。

 MySQL Workbench

 Image: Control of the person of t

图 3-19 删除选项

⑤ 图 3-20 显示 SQL 语句

操作数据库和表

除此之外还可以使用查询窗口,编写 SQL 语句来删除。删除表使用 DROP TABLE 命令,在该语句后添加表的名称即可。语法格式如下所示:

DROP TABLE tableName;

如删除 school 数据库中 student 表,执行代码及其执行结果如下所示。

USE school; DROP TABLE student;

3.6 字段操作

字段的操作包括字段的添加(在已经存在的表中添加字段)、字段信息的修改和删除 字段。本节详细介绍字段的操作。

●--3.6.1 添加字段 --,

字段的添加可以使用 MySQL Workbench 或 SQL 语句。在 MySQL Workbench 工具下 添加字段,相当于对表的信息进行修改,可直接在如图 3-18 所示的界面中添加即可,其 操作与表的修改操作一样。

而使用 SQL 语句修改字段信息相对比较麻烦,需要有字段名、数据类型、完整性约束等信息。使用 ADD 关键字添加字段的语法如下:

ALTER TABLE <表名> ADD <新字段名> <数据类型> [约束条件] [FIRST | AFTER 已存在字段名];

新字段名为需要添加的字段的名称; FIRST 为可选参数,其作用是将新添加的字段 设置为表的第一个字段; AFTER 为可选参数,其作用是将新添加的字段添加到指定的"已 存在字段名"的后面。

【范例 2】

向 shop 数据库的 goodses 表中添加字段,名为 manager、数据类型为长度可变的字 符串,可以为空,放在 price 字段的后面,代码如下。

ALTER TABLE 'shop'. 'goodses' ADD COLUMN 'manager' VARCHAR(45) NULL AFTER 'price';

75

第 3 章

上述代码中,若没有使用 AFTER 'price'语句,将默认在最后面添加新的字段。

●--3.6.2 字段修改 --,

字段的修改包括字段位置的修改、字段名称的修改和字段信息的修改。字段信息的 修改又包括字段数据类型的修改、字段约束的修改等。本节以位置的修改、字段名称的 修改和字段信息的修改来介绍字段的修改。

1. 字段位置修改

字段位置的修改可使用 MySQL Workbench 或 SQL 语句。在 MySQL Workbench 下可 进行的字段操作有很多,首先打开表修改界面,在需要修改的字段所在行右击可打开弹 出菜单,如图 3-21 所示。

弹出菜单有字段的上移、下移、删除等操作,可直接单击需要进行的选项进行字段的操作。但是这样的操作并没有保存在数据库中,操作后的效果将直接在界面中显示, 在用户单击 Apply 按钮后打开执行对话框,接着单击 Apply 按钮执行字段的修改,最后 在执行状态对话框中单击 Finish 按钮才能完成修改。



图 3-21 字段操作

除了上述操作以外,还可以通过 ALTER TABLE 来改变表中字段的相对位置,语法格式如下:

ALTER TABLE <表名> MODIFY <字段 1> <数据类型> FIRST|AFTER <字段 2>;

"字段 1"指要修改位置的字段,"数据类型"指"字段 1"的数据类型,"FIRST"为可选参数,指将"字段 1"修改为表的第一个字段,"AFTER 字段 2"指将"字段 1"插入到"字段 2"后面。

【范例3】

将 shop 数据库 goodses 表的 manager 字段放在 name 字段之后,代码如下。

ALTER TABLE 'shop'. 'goodses' MODIFY 'manager' VARCHAR(45) AFTER 'name';

2. 字段名称修改

字段名称的修改使用 MySQL Workbench 或 SQL 语句。在 MySQL Workbench 中的操 作与表的修改操作一样;使用 SQL 语句需要用到 CHANGE COLUMN,语法如下:

ALTER TABLE 数据表 CHANGE COLUMN '原字段名' '新字段名' 字段的数据类型 字段约束;

【范例4】

将 shop 数据库 goodses 表的 manager 字段重命名为 principal,代码如下。

ALTER TABLE 'shop'. 'goodses' CHANGE COLUMN 'manager' 'principal' VARCHAR(45) NULL DEFAULT NULL ;

3. 字段类型修改

字段类型和约束的修改使用 MySQL Workbench 或 SQL 语句。在 MySQL Workbench 中的操作与表的修改操作一样;使用 SQL 语句同样需要用到 CHANGE COLUMN,语法 如下:

77

ALTER TABLE 数据表 CHANGE COLUMN '字段名' '字段名' 字段的数据类型 字段约束;

【范例 5】

将 shop 数据库 goodses 表的 price 字段的数据类型修改为长度可变的字符串,代码 如下。

USE shop;ALTER TABLE goodses CHANGE COLUMN 'price' 'price' VARCHAR(45) NOT NULL;

●--3.6.3 删除字段 --,

字段的删除使用 MySQL Workbench 或 SQL 语句。在 MySQL Workbench 中的操作与 表的修改操作一样;使用 SQL 语句语法如下:

ALTER TABLE <表名> DROP <字段名>;

【范例6】

删除 shop 数据库 goodses 表的 principal 字段,代码如下。 ALTER TABLE 'shop'.'goodses' DROP principal

3.7) 实验指导——水果数据库管理

本章详细介绍了数据库和表的相关操作,包括数据库和表的概念、创建和对数据库

和表的管理等。本节综合本章内容,创建数据库和表,实现水果数据库的管理。要求具 体实现下列操作。

- (1) 创建水果数据库名称为 fruitshop。
- (2) 创建水果表 fruit, 有字段 fid、fname、fprice、ftime。
- (3) 添加负责人字段 fmanager。
- (4) 修改 fmanager 字段名为 principal。

(5) 添加表数据。

- (6) 查看表的结构和数据。
- (7) 修改水果信息表的名字为 fruits。

实现上述操作,步骤如下。

(1) 创建水果数据库名称为 fruitshop, 代码如下。

CREATE SCHEMA fruitshop;

(2) 在 fruitshop 中创建水果表 fruit, 有字段 fid、fname、fprice、ftime,代码如下。

```
CREATE TABLE 'fruitshop'.'fruit' (

'fid' INT NOT NULL,

'fname' VARCHAR(45) NULL,

'fprice' VARCHAR(45) NULL,

'ftime' VARCHAR(45) NULL,

PRIMARY KEY ('fid'));
```

(3)向 fruit 表中添加负责人字段,代码如下。

ALTER TABLE 'fruitshop'. 'fruit' ADD COLUMN 'fmanager' VARCHAR(45) NULL AFTER 'ftime';

(4) 修改 fmanager 字段名为 principal, 代码如下。

ALTER TABLE 'fruitshop'. 'fruit' CHANGE COLUMN 'fmanager' 'fprincipal' VARCHAR(45) NULL DEFAULT NULL ;

(5)向 fruit 表中添加 4 条数据,代码如下。

```
INSERT INTO 'fruitshop'.'fruit'('fid','fname','fprice','ftime',
'fprincipal') VALUES ('1','orange','5','6.15','zhang');
INSERT INTO 'fruitshop'. 'fruit' ('fid', 'fname', 'fprice', 'ftime',
'fprincipal') VALUES ('2', 'apple', '2.5', '6.15', 'duan');
INSERT INTO 'fruitshop'. 'fruit' ('fid', 'fname', 'fprice', 'ftime',
'fprincipal') VALUES ('3', 'banana', '3.5', '6.18', 'he');
INSERT INTO 'fruitshop'. 'fruit' ('fid', 'fname', 'fprice', 'ftime',
'fprincipal') VALUES ('4', 'watermelon', '0.7', '6.15', 'meng');
```

(6) 查看表的结构,代码如下。

USE fruitshop;DESC fruit;

上述代码的执行效果如下所示。

+		+	+	- +	+	- +	+
	Field	Туре	Null	Key	Default	Extra	
+		+	+	- +	+	- +	+
	fid	int(11)	NO	PRI	l.	l. I	1
	fname	varchar(45) YES	l. I	l l	l. I	1
	fprice	varchar	(45) YES	I.	I.	I.	1
	ftime	varchar	(45) YES	I.	I.	I	
	fprincipa	al varcha	r(45) YE	S	I	I.	1
+		+	+	- +	+	- +	+
5	rows						

(7) 查看表中的数据,代码如下。

USE fruitshop; SELECT *FROM fruit;

上述代码的执行结果如下所示。

+		- +	+	+	+ +		+
I	fid	fname	fprice	ftime	fprincipa	al	
+		- +	+	+	+ +		+
I	1	orange	5	6.15	zhang	I	
	2	apple	2.5	6.15	duan	1	
	3	banana	3.5	6.18	he	I.	
	4	watermelo	n 0.7	6.15	meng	1	
+		- +	+	+	+ +		+
Λ	rowe						

4 rows

(8) 修改水果信息表的名字为 fruits,代码如下。

ALTER TABLE 'fruitshop'. 'fruit' RENAME TO 'fruitshop'. 'fruits';

思考与练习

一、填空题

1. 存储引擎为_____的数据库无法修 改默认数据库名称。

2. 创建表使用_____语句。

3. 查看表结构,可以使用 DESCRIBE 或 。

4. 添加字段时,可将字段放在第一位或指 定字段的_____。

二、选择题

1. 查看某一个数据库的详细信息,使用如 下关键字_____。

A. CHECK

B. SELECT

- C. SHOW
- D. CHOICE
- 查看表结构时,所显示的是_____。
 A.表的属性
 - A. 农门周庄
 - B. 表的所有字段名称
 - C. 表的完整数据
 - D. 所有字段的名称和类型等
- 3. 关于添加表数据,下列说法错误的是
 - A. 使用 SQL 语句添加表数据,允许某 些字段的数据不添加
 - B. 使用 SQL 语句添加表数据,必须根据字段顺序列举出来所有的字段
 - C. 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

- D. 若某个字段需要存储的数据在其许可范围之外, MySQL 会根据允许范围最接近它的一端截断后再进行存储
- 4. 查看数据库中的表,使用_____。
 - A. SHOW TABLES
 - B. SELECT TABLES
 - C. DESC TABLES
 - D. GET TABLES
- 5. 修改字段位置时,不能够修改为____。
 - A. 第一个字段
 - B. 最后一个字段
 - C. 指定字段的前面
 - D. 指定字段的后面
- 6. 下列说法正确的是_____。

- A. 修改字段的名称和修改表的名称都 使用 RENAME
- B. 修改字段的名称使用 RENAME;修改表的名称使用 RENAME TO
- C. 修改字段的名称使用 RENAME TO; 修改表的名称使用 RENAME
 - D. 修改字段的名称和修改字段的数据 类型都使用 CHANGE COLUMN
- 三、简答题

1. 总结可以在 MySQL Workbench 中进行的 数据库操作,并详细说明。

2. 总结可以在 MySQL Workbench 中进行的 字段操作,并详细说明。

- 简单介绍如何将最后一个字段放在第二 个字段的位置。
 - 4. 总结 SHOW 关键字可进行的操作。