

第一部分 《MySQL 8.0数据库 应用与开发（微课视频版）》 习题参考答案

学习导读：

本部分包括《MySQL 8.0 数据库应用与开发(微课视频版)》第 1~14 章的教学重点和难点提示,以及所有习题的参考答案,所有代码均在 MySQL 8.0.22 版本的环境下测试通过。

每章习题包括选择题、简答题和上机练习题 3 种类型。

- (1) 选择题侧重基本概念和基本操作的知识点掌握。
- (2) 简答题侧重基本概念的理解。
- (3) 上机练习题侧重基本操作的练习。

建议读者在学习过程中通过习题熟悉 MySQL 数据库的基本概念和基本操作,不要满足于看懂程序或习题答案,要多动手做练习,并且要达到一定的做题速度和正确率。另外,在此基础上借助于网络或其他工具,举一反三,解决一些比教材习题更新颖、算法更复杂的问题。

再者,本书第 3 章的习题答案之后附加的主教材和指导书常用数据库 teaching 的表结构和数据,仅供读者在学习过程中参考。读者在学习过程中,重点关注例题代码和习题代码的运行状况即可。读者还可以根据表结构的定义,在进行操作时自行添加一定数量的 MySQL 数据,以观察代码的运行状况,进一步加深自己的理解,并通过代码运行时间观察和理解算法优化的重要性。

需要指出的是,主教材例题、指导书上机习题的代码运行和上机实验测试,对数据库表中的数据进行了大量的增删查改操作,数据库 teaching 中的数据在代码运行过程中一定会发生一些变化。

第 1 章

MySQL 数据库概述

1.1 教学要求

1.1.1 基本要求

- (1) 掌握有关数据库的基本概念及其之间的关系,例如数据、信息、数据库、数据库管理系统、数据库系统和 SQL。
- (2) 掌握有关数据库的基本关系理论,掌握投影、选择和连接的内涵。
- (3) 掌握概念模型的实体、属性、联系、主键、实体集、实体型的基本概念和含义,了解概念模型的表示方法,了解数据库设计中关系模型的规范化的意义。
- (4) 了解 MySQL 数据库软件的特点和用途。
- (5) 熟练掌握 MySQL 数据库软件的下载、安装和配置过程。
- (6) 学会利用客户端工具登录 MySQL 数据库。
- (7) 了解可视化软件 MySQL Workbench 的安装和使用方法。
- (8) 积累和掌握本课程的常用英文词汇。

1.1.2 重点与难点

1. 本章重点

- (1) 数据库管理系统的功能和组成。
- (2) 关系数据库的基本理论。
- (3) 数据库系统的基本组成。
- (4) 安装、启动和配置 MySQL 的基本过程。
- (5) 利用客户端工具登录 MySQL 数据库。

2. 本章难点

- (1) MySQL 数据库软件的下载、安装和配置过程。
- (2) 利用客户端工具登录 MySQL 数据库。
- (3) 关系模型的规范化设计的用途。
- (4) 概念模型的表示方法——E-R 图的实现。

1.2 习题参考答案

1. 选择题

- (1) 数据模型的三要素不包括_____。
- A. 数据结构 B. 数据操作 C. 数据类型 D. 完整性约束
- (2) 关系运算不包括_____。
- A. 连接 B. 投影 C. 选择 D. 查询
- (3) 表 1-1(主教材中)所示的学生信息表中的主键为_____。
- A. studentno B. sex C. birthday D. sname

表 1-1 student 表

studentno	sname	sex	birthdate	phone
21125221327	何桐影	女	2004/12/4	13178978999
21131133071	崔依歌	男	2002/6/6	15556845645
...

- (4) 下面数据库产品中_____是开源数据库。
- A. Oracle B. SQL Server C. MySQL D. DB2
- (5) 在 E-R 模型中,信息的 3 种概念单元不包括_____。
- A. 实体型 B. 实体值 C. 实体属性 D. 实体间联系
- (6) E-R 图是数据库设计工具之一,一般适用于建立数据库的_____。
- A. 概念模型 B. 结构模型 C. 物理模型 D. 逻辑模型
- (7) SQL 又称_____。
- A. 结构化定义语言 B. 结构化控制语言
C. 结构化查询语言 D. 结构化操纵语言
- (8) 从 E-R 模型向关系模型转换,一个 $M:N$ 的联系转换成一个关系模式时,该关系模式的主键是_____。
- A. M 端实体的键 B. N 端实体的键
C. M 端实体键与 N 端实体键的组合 D. 重新选取其他属性
- (9) DB、DBS 和 DBMS 三者之间的关系是_____。
- A. DB 包括 DBMS 和 DBS B. DBS 包括 DB 和 DBMS
C. DBMS 包括 DB 和 DBS D. 不能相互包括

参考答案: CDACB ACCB

2. 简答题

(1) 什么是数据库管理系统? 请举出日常生活中一些应用数据库的实际范例。

答: 数据库管理系统(DBMS)是位于用户和操作系统之间的一种操纵和管理数据库的软件,用于建立、使用和维护数据库。它对数据库进行统一的管理和控制,以保证数据库的安全性和完整性,是数据库系统的核心。

在日常生活中,像学生信息管理、网上购物、银行存取款、购买火车票等,其软件的后台都是由数据库作为支撑。

(2) MySQL 数据库管理系统的基本架构拥有哪 4 个模块?

答: 数据库管理系统由多种不同的程序模块组成, 数据库管理系统的基本架构一般包括以下 4 个模块。

① 存储管理: 数据库管理系统通常会自行配置磁盘空间, 将数据存入存储装置的数据库。

② 查询处理: 负责处理用户下达的查询语言命令语句, 可以再细分成多个模块, 负责检查语法、优化查询命令的处理程序。

③ 事务管理: 在事务管理中, 同名的事务管理子系统负责处理数据库的事务, 保障数据库事务的操作需要一并执行; 锁定管理也称为并发控制管理子系统, 负责资源的锁定。

④ 恢复管理: 恢复管理主要是日志管理子系统, 负责记录数据库的所有操作, 包含事务日志, 以便同名的恢复管理子系统能够执行恢复处理, 恢复数据库系统存储的数据到指定的时间点。

(3) 举例说明 3 种关系运算的特点。

答: 在关系中访问所需的数据时, 需要对关系进行一定的关系运算。关系数据库主要支持选择、投影和连接关系运算, 它们源于关系代数中的并、交、差、选择、投影和连接等运算。

① 选择关系用于查找符合条件的行, 从而形成新关系。新关系和原来关系的结构一致, 记录行一般会变少。例如, 在学生表中查询年龄在 18 岁以上的学生记录。

② 投影关系是从一个表中找出若干字段形成一个新关系的操作, 新关系与原来关系的记录行应该一样多, 但其数据列一般会变少。例如, 在学生表中输出学生的学号、姓名和出生日期。

③ 连接关系是从两个表中选择满足条件的记录, 且新关系中包含原来多个关系的列, 新关系结构中的列类型由原来关系中的列映射而来。例如, 查询学生的学号、姓名和期末成绩。3 个输出项并不是在一个表中可以搜索到, 需要进行连接操作, 即将两个或多个表中的行按一定的条件横向结合, 形成一个新的临时表, 在此临时表中进行数据查询。

3. 上机练习题

(1) 在“服务”窗口中练习 MySQL 服务器的启动和关闭方法。

答: 单击“开始”按钮, 选择“运行”命令, 输入 services. msc 后按 Enter 键, 弹出如图 1-1 所示的窗口。从中可以看到 MySQL 服务已经启动, 而且服务的启动类型是自动启动。



图 1-1 “服务”窗口

右击 MySQL80 服务项,弹出如图 1-2 所示的快捷菜单,可以通过该菜单对 MySQL80 服务器进行“启动”“停止”“暂停”“恢复”“重新启动”等操作。



图 1-2 更改服务的启动类型

(2) 在控制台中完成启动 MySQL 服务后打开控制台,在控制台中输入密码完成 MySQL 的登录,并且执行 select 的相关语句查看系统的当前日期。

答:主要操作命令和运行结果如下。

```
C:\Users\Administrator> mysql -u root -p
Enter password: *****
mysql> select curdate();
+-----+
| curdate() |
+-----+
| 2020-12-01 |
+-----+
1 row in set (0.07 sec)
```

(3) 打开 MySQL Workbench 软件,简述其可视化界面的主要功能区。

答:单击“开始”按钮,选择“所有程序”命令,然后执行 MySQL 路径下的 MySQL Workbench 8.0 CE 命令,即可进入 MySQL Workbench 连接界面,输入密码后进入如图 1-3 所示的 MySQL Workbench 软件的可视化界面。

其主要功能区如下。

① 菜单和工具栏:如图 1-4 所示,在 MySQL Workbench 主界面中可以通过 File、Edit、View、Query、Database 等下拉菜单和工具栏实现 MySQL 数据库对象的创建、修改、查询、删除等常见操作。

② Navigator 导航区:如图 1-5 所示,其主要包括 MANAGEMENT(管理)下的服务器状态、客户端连接、用户权限、数据的导出等,以及实例(INSTANCE)和性能(PERFORMANCE)管理。

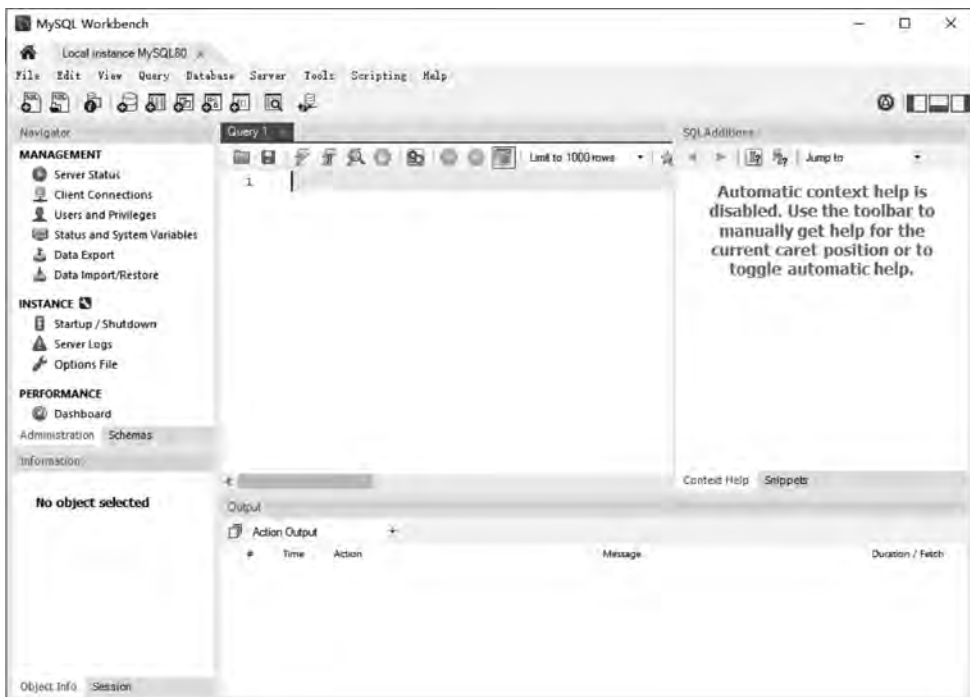


图 1-3 MySQL Workbench 软件的可视化界面



图 1-4 菜单和工具栏

③ Schemas 数据库：如图 1-6 所示，在数据库操作界面的 Schemas 区域可以看到用户创建的数据库以及 MySQL 中的系统数据库，通过展开数据库文件夹可以看到数据库对象。右击数据库对象，可以实现对数据库对象的创建、修改、查询和删除等操作。



图 1-5 导航区



图 1-6 Schemas 数据库


④ 命令的编辑和执行: 通过对数据库对象的右键快捷菜单操作或直接输入命令,可以在 Query 区进行命令的编辑和执行,并将结果输出到 Result 区。如图 1-7 所示,输入一个表达式和函数,单击“执行”按钮 ,结果就显示出来。同时,Output 区中会显示操作日志,如图 1-8 所示。



图 1-7 查询命令的编辑和执行

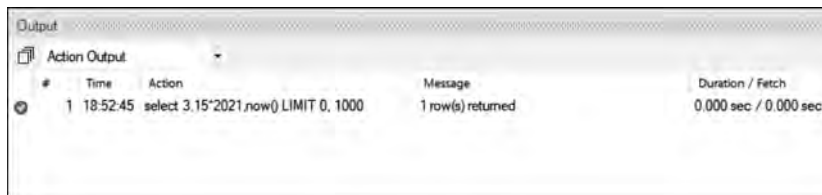


图 1-8 操作日志

另外,在 SQL additions 工作区中通过 Jump to 下拉框选择 select、update、delete 等命令,则会显示相关命令的帮助信息,例如命令格式等。

第2章

MySQL 8.0语言基础

2.1 教学要求

2.1.1 基本要求

- (1) 了解字符集和字符序的基本概念和用途。
- (2) 掌握标识符、关键字、常量、变量等 MySQL 8.0 基本语法要素的概念和应用。
- (3) 熟练掌握常用字符型、数值型、日期时间型等数据类型的用途。
- (4) 熟练掌握算术运算符、比较运算符、逻辑运算符及位运算的用途,了解运算符的优先级及其作用。
- (5) 熟练掌握常用字符型函数、日期时间型函数、数值型函数的格式和用途。
- (6) 掌握各种值类型表达式的写法和运算类型。
- (7) 学会利用 select 语句对常量、变量、函数、表达式等进行数据输出的操作。

2.1.2 重点与难点

1. 本章重点

- (1) 字符集和字符序的基本概念。
- (2) 标识符、关键字、常量、变量等 MySQL 8.0 基本语法要素的概念和应用。
- (3) 常用字符型、数值型、日期时间型等数据类型的用途。
- (4) 算术运算符、比较运算符、逻辑运算符及位运算的用途。
- (5) 常用字符型函数、日期时间型函数、数值型函数的格式和用途。

2. 本章难点

- (1) 各种数据类型的变量的定义和使用方法。
- (2) 常用字符型函数、日期时间型函数、数值型函数的格式。
- (3) 算术运算符、比较运算符、逻辑运算符的使用方法。

2.2 习题参考答案

1. 选择题

- (1) 以下_____命令是 DML 语句。
- A. create database B. alter table C. select D. alter database

- (2) 以下关于 MySQL 的说法中错误的是_____。
- A. MySQL 是一种关系型数据库管理系统
 B. MySQL 软件是一种开放源代码的软件
 C. MySQL 服务器工作在客户端/服务器端模式下或嵌入式系统中
 D. MySQL 中的 MySQL 语句区分大小写
- (3) 在控制台中执行_____语句时可以退出 MySQL。
- A. exit B. go 或 quit C. go 或 exit D. exit 或 quit
- (4) 关于 MySQL 数据库的说法,下列选项_____的说法是错误的。
- A. MySQL 数据库不仅开放源代码,而且能够跨平台使用
 B. MySQL 数据库启动服务时有两种方式,如果服务已经启动,可以在任务管理器中查找 mysqld. exe 程序,如果该进程存在,则表示正在运行
 C. 手动更改 MySQL 的配置文件 my. ini 时只能更改与客户端有关的配置,不能更改与服务器端相关的配置信息
 D. 登录 MySQL 数据库成功后,直接输入“help;”语句,按 Enter 键可以查看帮助信息
- (5) 下列_____类型不是 MySQL 中常用的数据类型。
- A. int B. var C. time D. char
- (6) 在 MySQL 中会话变量前面的字符为_____。
- A. 空格 B. # C. @@ D. @
- (7) 设置表的默认字符集的关键字是_____。
- A. default character B. default set
 C. default D. default character set

参考答案: CDDCB DD

2. 简答题

(1) 简述字符集 utf8mb4 与 UTF-8 的区别。

答: utf8mb4 是 MySQL 8.0 的默认字符编码,能够实现 UTF-8 Unicode 机制的编码,其中 mb4 表示 most bytes 4,最多占用 4 字节。UTF-8 编码是一种变长的编码机制,可以用 1~4 字节存储字符。MySQL 8.0 默认的排序字符序是 utf8mb4_0900_ai_ci。

UTF-8 是针对 Unicode 字符的一种变长字符编码。该字符集对英文使用 1 字节来编码,对中文使用 3 字节来编码。UTF-8 包含全世界所有国家需要用到的字符,是通用国际编码。UTF-8 编码的文字可以在各国支持 UTF-8 字符集的浏览器上显示。

(2) datetime 类型和 timestamp 类型有什么相同点和不同点?

答: 日期与时间类型是为了方便在数据库中存储日期和时间而设计的,datetime 和 timestamp 表示日期和时间。其中每种类型都有其取值范围,如果赋予它一个不合法的,将会被 0 代替。datetime 的取值范围为 1000-01-01 00:00:00~9999-12-31 23:59:59,而 timestamp 的取值范围是 1970-01-01 00:00:00 至 2037 年的某个时间。

(3) MySQL 支持的数据类型主要分成哪 3 类? 15 属于什么类型? 17 属于什么类型?

答: MySQL 支持的数据类型主要分成数值型、字符型和日期时间型。15 属于数值型中的整型数据。17 属于字符型数据。

(4) 简述系统变量、全局变量和会话变量的关系。

答：系统变量包括全局系统变量和会话系统变量两种类型。

全局变量在 MySQL 启动时由服务器自动初始化为默认值，主要影响整个 MySQL 实例的全局设置，大部分全局变量都是作为 MySQL 的服务器调节参数存在。对全局变量的修改会影响到整个服务器。

会话变量在每次建立一个新的连接时由 MySQL 来初始化。对会话变量的修改只会影响到当前的会话，也就是当前的数据库连接。会话变量可以随时定义，必须在名称前加@符号。

(5) 简述聚合函数的特点和用途。

答：聚合函数用于对一组值进行计算并返回一个单一的值。除 count() 函数之外，其他聚合函数忽略空值。聚合函数经常与 select 语句的 group by 子句一起使用。聚合函数的作用是在结果集中通过对被选列值的收集处理，返回一个数值型的计算结果。

3. 上机练习题

(1) 创建文本文件“D:\my717.txt”，查询系统当前日期、当前时间以及到 2025 年 1 月 1 日还有多少天，然后通过 MySQL 命令执行文本文件中的内容。

文本文件的内容如下：

```
select now();           -- 查询当前日期
select curtime();      -- 输出当前时间
select datediff('2025-1-1',now()); -- 到 2025 年 1 月 1 日的天数
```

答：在 DOS 窗口中执行 MySQL 命令，指定文本文件并执行该文件中的语句。

输出结果如下：

```
C:\Users\Administrator> mysql <D:\my717.txt -u root -p
Enter password: *****
now()
2021-03-18 10:58:28
curtime()
10:58:28
datediff('2025-1-1',now())
1385
C:\Users\Administrator>
```

(2) 利用随机函数输出 20~90 的任意两个数(含两位小数)。

答：命令和运行结果如下。

```
mysql> select 20 + round(70 * rand(),2), 20 + round(70 * rand(),2);
+-----+-----+
| 20 + round(70 * rand(),2) | 20 + round(70 * rand(),2) |
+-----+-----+
| 45.89 | 20.68 |
+-----+-----+
1 row in set (0.02 sec)
```

(3) 计算 1000 天后的日期和 3000 分钟后的时间。

答：命令和运行结果如下。

```
mysql > select date_add(now(), interval 1000 day);
```

```
+-----+
| date_add(now(), interval 1000 day) |
+-----+
| 2023 - 12 - 12 13:48:37           |
+-----+
1 row in set (0.00 sec)
```

```
mysql > select date_add(now(), interval 3000 minute);
```

```
+-----+
| date_add(now(), interval 3000 minute) |
+-----+
| 2021 - 03 - 19 13:48:37             |
+-----+
1 row in set (0.00 sec)
```

第3章

MySQL 8.0数据库 和表的基本操作

3.1 教学要求

3.1.1 基本要求

- (1) 掌握概念模型的设计过程和利用 E-R 图描述概念模型的方法。
- (2) 掌握 MySQL 数据库的创建、打开和存储方法。
- (3) 学会运用 MySQL Workbench 可视化软件维护数据库。
- (4) 掌握创建数据库表的命令格式,以及表中各字段的数据类型和取值范围。
- (5) 熟练掌握数据表的创建、查看、修改和删除等操作。
- (6) 熟练掌握数据表记录的添加、修改和删除操作。
- (7) 掌握数据表中各种约束的创建和管理操作。了解表的数据完整性规则与数据的正确性、精确性和可靠性的关系。
- (8) 了解临时表的用途和管理操作。

3.1.2 重点与难点

1. 本章重点

- (1) 利用 E-R 图描述概念模型的方法。
- (2) 设置和修改 MySQL 数据库存储引擎的操作。
- (3) MySQL 数据库的创建、打开、删除和存储方法。
- (4) 创建数据库表的命令格式。
- (5) 数据表的创建、查看、修改和删除等操作。
- (6) 表记录的添加、修改和删除操作。
- (7) 利用 MySQL Workbench 管理表的基本过程。
- (8) 数据表中各种约束的创建和管理操作。

2. 本章难点

- (1) 利用 E-R 图描述概念模型的方法。
- (2) 创建数据库表的命令格式。
- (3) 数据表的创建、查看、修改和删除等操作。
- (4) 表的数据完整性的用途。

(5) 数据表中各种约束的创建和管理操作。

3.2 习题参考答案

1. 选择题

(1) 在 MySQL 数据库中,通常使用_____语句来指定一个已有数据库作为当前工作数据库。

- A. using B. used C. uses D. use

(2) _____命令用于删除一个数据库。

- A. create database B. drop database
C. alter database D. use InnoDB

(3) 在使用 insert 语句插入记录时,使用_____关键字会忽略导致重复关键字的错误记录。

- A. no same B. ignore C. repeat D. unique

(4) 删除列的命令是_____。

- A. alter table...delete... B. alter table...delete column...
C. alter table...drop... D. alter table...drop column...

(5) 下列关于 truncate table 的描述不正确的是_____。

- A. truncate 将删除表中的所有数据
B. 表中包含 auto_increment 列,使用 truncate table 可以重置序列值为该列的初始值
C. truncate 操作比 delete 操作占用的资源多
D. 用 truncate table 删除表,然后重新构建表

(6) 在创建表时,可以使用_____关键字使当前建立的表为临时表。

- A. ignore B. temporary C. temptable D. truncate

参考答案: DBB DCB

2. 简答题

(1) 简述在 MySQL Workbench 窗口中创建数据库的步骤。

答: 在 MySQL Workbench 窗口中使用可视化界面可以创建数据库。使用 MySQL Workbench 创建数据库的步骤如下:

① 进入 MySQL Workbench 主界面进行 MySQL 数据库的连接,单击 Create Schemas 命令(创建数据库)。

② 进入设置数据库参数界面,输入数据库名,选择字符集和排序规则,然后单击 Apply 按钮。

③ 打开创建数据库脚本显示对话框,单击 Apply 按钮,即可完成数据库的创建。

④ 在数据库操作界面的 Schemas 区域可以看到刚才创建的数据库和 MySQL 中的系统数据库。

(2) 简述创建表时各类约束对表中数据的作用。

答: 在定义表结构时还可以定义与该表相关的完整性约束条件,包括实体完整性、参照完整性和用户定义完整性。这些约束条件主要包括 not null(非空约束)、primary key(主键

约束)、unique(唯一性约束)、foreign key(外键约束)以及 check(检查约束)。各种约束对表中数据的作用如下:

- ① 非空约束限制在表的一个列中必须输入一个值。
- ② 主键约束确保一个表中的每一行可以被指定列或指定列的组合唯一地标识。一个表只能有一个主键,但可以创建多个唯一性约束。
- ③ 唯一性约束确保一列或列的组合没有重复值。
- ④ 外键约束强制一个列只能取一个被引用表中存在的值。
- ⑤ 检查约束限制一列的允许值。

(3) MySQL 支持的数据完整性有哪几类? 各有什么作用?

答: 数据完整性可以根据其作用的数据库对象和范围不同分为 4 类,具体如下。

① 域完整性: 要求列的数值具有正确的类型、格式和有效值范围,并确定是否允许有空值。通常使用有效性检查强制域完整性,也可以通过限定列中允许的数据类型、格式或有效值范围来强制数据完整性。

② 实体完整性: 要求表中的每一行具有唯一的标识。例如 student 表中的列 studentno 被设为主键,则会保证每个学生只有一个学号,而且是唯一的。

③ 引用完整性: 指两个表的主键与外键之间定义的数据完整性,将确保主键和外键的关系。引用完整性可以保证两个引用表间数据的一致性。

④ 用户定义完整性: 用户可以根据其应用环境的不同对数据库设置一些特殊的约束条件,反映某一具体应用所涉及的数据必须满足的语句要求。用户定义完整性的实现机制有规则(Rule)、触发器(Trigger)和存储过程(Stored Procedure)以及创建数据表时的所有约束(Constraint)。

(4) 简述在 MySQL Workbench 中创建含有主键的表的步骤。

答: 创建 MySQL 数据表,可以利用 MySQL Workbench 实现可视化方式。下面介绍利用可视化方式创建表的过程。

- ① 打开 MySQL Workbench 工具,在 Schemas 区域找到数据库列表。
- ② 选择一个数据库作为当前数据库,即右击该数据库,在弹出的快捷菜单中选择 Set as Default Schema 命令,则该数据库就变成当前数据库。
- ③ 选择 tables 并右击,在弹出的快捷菜单中选择 Create Tables 命令,进入创建数据表的初始界面。
- ④ 输入数据库表名,选择默认数据库引擎 InnoDB,参照设计好的表结构分别输入列名,选择数据类型,输入不同类型的参数或默认值。
- ⑤ 单击 Apply 按钮,打开脚本审核对话框。单击 Show Logs 按钮,可以进一步查看脚本。单击 Finish 按钮,则完成数据表的创建。

3. 上机练习题

(1) 登录数据库系统以后,创建 student 数据库,数据库创建成功后,查看数据库系统中还存在哪些数据库。

答: 命令和运行结果如下。

```

C:\Users\Administrator>mysql -u root -p
Enter password: *****
mysql> create database if not exists student;
Query OK, 1 row affected (0.02 sec)
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| mytest |
| performance_schema |
| sakila |
| student |
| sys |
| teaching |
| world |
+-----+
9 rows in set (0.00 sec)

```

(2) 创建 booksmgt 数据库,在 booksmgt 数据库中使用 MySQL 语句创建 book 表和 author 表,结构如下:

```

book(bookid char(6),bookname varchar(30), price float(5,2))
author(authorid char(6),authorname varchar(10), bookid char(6), phone varchar(15))

```

设置 book 表中的 bookid 为主键,author 表中的 bookid 为外键。

答: ① 创建 book 表的 MySQL 语句。

```

mysql> create database booksmgt;
Query OK, 1 row affected (0.19 sec)
mysql> use booksmgt;
Database changed
mysql> create table book (
-> bookid char(6) not null,
-> bookname varchar(30) null,
-> price float(5, 2) null,
-> constraint PKbook primary key (bookid asc)
-> );
Query OK, 0 rows affected (0.78 sec)

```

② 创建 author 表的 MySQL 语句。

```

mysql> create table author(
-> authorid char(6) not null,
-> authorname varchar(10) null,
-> bookid char(6) null,
-> phone varchar(15) null,
-> );
Query OK, 0 rows affected (0.18 sec)

```


③ 设置 book 表中的 bookid 为主键,author 表中的 bookid 为外键。

```
mysql > alter table author
  -> add constraint fk_book_author
  -> foreign key(bookid) references book(bookid);
      Query OK, 0 rows affected (0.42 sec)
      Records: 0 Duplicates: 0 Warnings: 0
```

(3) 在数据库 booksmgt 中利用 MySQL 语句创建一个图书销售表 booksales,结构如下:

```
booksales(bookid nchar(6), sellnum int, selldate datetime)
```

分别利用 insert、delete、update 语句添加、删除和更新数据。

答: ① 利用 MySQL 语句创建 booksales 表。

```
mysql > create table booksales(
  -> bookid nchar(6) not null,
  -> sellnum int not null,
  -> selldate datetime not null
  -> );
      Query OK, 0 rows affected (0.14 sec)
```

② 利用 insert 语句为 booksales 表添加数据。

```
mysql > insert into booksales values ('m00011',7, '2019-12-20');
      Query OK, 1 row affected (0.12 sec)
mysql > insert into booksales(bookid,sellnum,selldate)
  -> values ('m00018',3, '2019-11-16');
      Query OK, 1 row affected (0.07 sec)
```

③ 利用 update 语句为 booksales 表更新数据。

```
mysql > update booksales set sellnum = 11
  -> where bookid = 'm00011';
      Query OK, 1 row affected (0.09 sec)
```

④ 利用 delete 语句删除 booksales 表的数据。

```
mysql > delete from booksales
  -> where bookid = 'm00011';
      Query OK, 1 row affected (0.08 sec)
```

(4) 利用 MySQL 语句先删除 booksales 表中销售时间在 2019 年 12 月以前的记录,再删除全部记录,然后删除该表。

答: 利用 MySQL 语句删除 booksales 表的数据。

```
mysql > delete from booksales
  -> where selldate <'2019-12-01';
      Query OK, 1 row affected (0.04 sec)
mysql > truncate table booksales; -- 删除所有数据
      Query OK, 0 rows affected (0.16 sec)
mysql > drop table booksales;
      Query OK, 0 rows affected (0.07 sec)
```

附：本书中常用的数据库 teaching 中的表结构和表记录。

(1) 学生信息表 student。

student 表结构

```
create table if not exists student (
  studentno char(11) not null comment'学号',
  sname char(8) not null comment'姓名',
  sex enum('男', '女') default '男' comment'性别',
  birthdate date not null comment'出生日期',
  entrance int(3) null comment'入学成绩',
  phone varchar(12) not null comment'电话',
  Email varchar(20) not null comment'电子信箱',
  primary key (studentno)
);
```

student 表的常用数据

studentno	sname	sex	birthdate	entrance	phone	Email
20112100072	许东方	男	2002/2/4	658	12545678998	su12@163.com
20112111208	韩吟秋	女	2002/2/14	666	15878945612	han@163.com
20120203567	封白玫	女	2003/9/9	898	13245674564	feng@126.com
20120210009	崔舟帆	男	2002/11/5	789	13623456778	cui@163.com
20123567897	赵雨思	女	2003/8/4	879	13175689345	pinan@163.com
20125121109	梁一苇	女	2002/9/3	777	13145678921	bing@126.com
20126113307	姚扶媚	女	2003/9/7	787	13245678543	zhu@163.com
21125111109	敬秉辰	男	2004/3/1	789	15678945623	jing@sina.com
21125221327	何桐影	女	2004/12/4	879	13178978999	he@sina.com
21131133071	崔依歌	男	2002/6/6	787	15556845645	cui@126.com
21135222201	夏文斐	女	2005/10/6	867	15978945645	xia@163.com
21137221508	赵临江	男	2005/2/13	789	12367823453	ping@163.com

(2) 课程信息表 course。

course 表结构

```
create table if not exists course (
  courseno char(6) not null,
  cname char(6) not null,
  type char(8) not null,
  period int(2) not null,
  exp int(2) not null,
  term int(2) not null,
  primary key (courseno)
);
```

course 表的常用数据

courseno	cname	type	period	exp	term
c05103	高等数学	必修	64	16	2
c05109	C 语言	必修	48	16	2
c05127	数据结构	必修	64	16	2
c05138	软件工程	选修	48	8	5
c06108	机械制图	必修	60	8	2
c06127	机械设计	必修	64	8	3
c06172	铸造工艺	选修	42	16	6
c08106	经济法	必修	48	0	7

c08123	金融学	必修	40	0	5
c08171	会计软件	选修	32	8	8

(3) 成绩信息表 score。

score 表结构

```
create table if not exists score(
  studentno char(11) not null,
  courseno char(6) not null,
  daily float(4,1) default 0,
  final float(4,1) default 0,
  primary key (studentno, courseno)
);
```

score 表的常用数据

studentno	courseno	daily	final
20112100072	c05103	99	92
20120203567	c05103	78	67
20120210009	c05103	65	98
20125121109	c05103	88	79
21125111109	c05103	96	97
21137221508	c05103	77	92
20112100072	c05109	95	82
20120203567	c05109	87	86
20125121109	c05109	77	82
20126113307	c05109	89	95
21125111109	c05109	87	82
21125221327	c05109	89	95
20120210009	c05138	88	89
21137221508	c05138	74	91
20112111208	c06108	77	82
20120210009	c06108	79	88
20123567897	c06108	99	99
20126113307	c06108	78	67
20112111208	c06127	85	91
20120203567	c06127	97	97
20112111208	c06172	89	95
21125221327	c06172	88	62
21131133071	c06172	78	95
21125111109	c08106	77	91
21135222201	c08106	91	77
21137221508	c08106	89	62
21131133071	c08123	78	89
21135222201	c08123	79	99
20112100072	c08171	82	69
20125121109	c08171	85	91
21131133071	c08171	88	98
21135222201	c08171	85	92

(4) 教师信息表 teacher。

teacher 表结构

```
create table if not exists teacher (
```

```

teacherno char(6) not null comment '教师编号',
tname char(8) not null comment '教师姓名',
major char(10) not null comment '专业',
prof char(10) not null comment '职称',
department char(16) not null comment '部门',
primary key (teacherno)
);

```

teacher 表的常用数据

teacherno	tname	major	prof	department
t05001	苏超然	软件工程	教授	计算机学院
t05002	常可观	会计学	助教	管理学院
t05003	孙释安	网络安全	教授	计算机学院
t05011	卢敖治	软件工程	副教授	计算机学院
t05017	茅佳峰	软件测试	讲师	计算机学院
t06011	夏期年	机械制造	教授	机械学院
t06023	卢释舟	铸造工艺	副教授	机械学院
t07019	韩庭宇	经济管理	讲师	管理学院
t08017	白成园	金融管理	副教授	管理学院
t08058	孙有存	数据科学	副教授	计算机学院

(5) 纽带表 teach_course。

teach_course 表结构

```

create table if not exists teach_course (
teacherno char(6) not null,
courseno char(6) not null,
primary key (teacherno,courseno)
);

```

teach_course 表的常用数据

teacherno	courseno
t05001	c05103
t05002	c05109
t05003	c05127
t05011	c05138
t05017	c06108
t05017	c06172
t06011	c06127
t06023	c05127
t06023	c06172
t07019	c08106
t08017	c08123
t08058	c08171

(6) 选课信息表 se_course。

se_course 表结构

```

create table se_course
(sc_no int(6) not null auto_increment,
studentno char(11) not null,
courseno char(6) not null,
teacherno char(6) not null,
score float(4,1),

```