

# 第 1 部分 MySQL 基础

第 1 章 初识 MySQL

第 2 章 数据库设计

第 3 章 MySQL 数据类型

# 第 1 章

## 初识 MySQL

### 本章内容：

- MySQL 的介绍
- 在 Windows 上安装 MySQL
- 在 Linux 上安装 MySQL
- 在 Docker 上安装 MySQL



本章将讲解 MySQL 在 Web 业务场景下的意义和优势，介绍 MySQL 8 的特性和作用，并讲述在多种环境下安装 MySQL 的步骤，以便初学者可以快速开始 MySQL 的学习。

### 1.1 MySQL 是什么

MySQL 是当下流行的关系数据库管理系统（Relational Database Management System, RDBMS），使用 C 和 C++ 语言编写而成，因而具有源码级的可移植性。MySQL 可以在 Linux、macOS、Novell NetWare、OpenBSD、Solaris、Windows 等多种操作系统中运行。在 Web 应用方面，MySQL 是非常好的关系数据库管理系统应用软件之一。作为一种关系数据库管理系统，它将数据保存在不同的数据表中。MySQL 所使用的 SQL 语言是用于访问数据库的常用标准化语言。由于 MySQL 体积小、速度快、总拥有成本低，尤其是具有开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。由于其社区版性能卓越，因此搭配各种后端语言可以组成良好的开发环境。此外，MySQL 既可以嵌入应用程序中，也可以独立支持数据仓库、全文索引和高可用的冗余系统、在线事务处理系统。另外，MySQL 还具有如下特点：

- 为多种编程语言提供了 API，这些编程语言包括 C、C++、Python、Java、Perl、PHP、Ruby、.NET 等。
- 支持多线程，可以充分利用 CPU 资源。
- 具有优化的 SQL 查询算法，有效地提高了查询速度。
- 提供 TCP/IP、ODBC 和 JDBC 等多种数据库连接途径。
- 支持可以处理上千万条记录的大型数据库。
- 支持多种存储引擎，比如 InnoDB、MyISAM 等。
- MySQL 是可以定制的，它采用了 GPL 协议，因而用户可以通过修改源码来开发自己的 MySQL 系统。
- 提供多语言支持，常见的编码如中文的 GB2312、UTF-8 等都可以用作数据表名和数据列名。
- 提供用于管理、检查、优化数据库操作的管理工具。
- 具有在线 DDL 更改功能，数据架构支持动态应用程序，提高了开发人员操作数据表的灵活性。
- 复制无崩溃从机，可以提高可用性。
- 复制多线程从机，可以提高性能。

## 1.2 MySQL 8 安装和配置

### 1.2.1 安装版本的选择

在安装 MySQL 时，应选择自己所需的版本和相应的文件格式。MySQL 有开发版本和通用版本。开发版本具有最新功能，但不建议用于生产用途。通用版本可用于生产用途，因此建议安装最新的版本。

MySQL 8.0 的版本号由三个数字和一个可选后缀组成，例如 MySQL-8.0.3.[x]，x 可选，如果有，则会随着版本的发布而自增。版本号中的数字解释如下：

- 第一个数字（8）是主版本号。
- 第二个数字（0）是次版本号。主、次版本号一起构成发布的序列号，序列号表示包含稳定的功能集。
- 第三个数字（3）是发布系列中的版本号。每个新的错误修订版的版本号中都会增加此值。一般情况下建议安装最新的版本。

### 1.2.2 在 Windows 平台下安装和配置 MySQL

MySQL 只能安装在 Microsoft Windows 64 位操作系统中，如果想要安装 MySQL 8.0 Server（服务器版），那么还需要在系统中安装 Microsoft Visual C++ 2015 Redistributable Package 和 Microsoft .NET Framework 4.5.2 或更高版本。在安装服务器版本之前，应确保在本地系统中已

下载好软件安装包。下面将演示在 Windows 系统中安装 MySQL 数据库的两种方式。

### 1. MySQL Installer（利用安装程序安装）

**步骤 01** 进入 MySQL 官网下载相关安装包，如图 1-1 所示。第一个安装包用于在线安装，在网络畅通的情况下，可以选用这种安装方式。第二个是可以离线安装的软件包，一般建议采用这种安装方式。

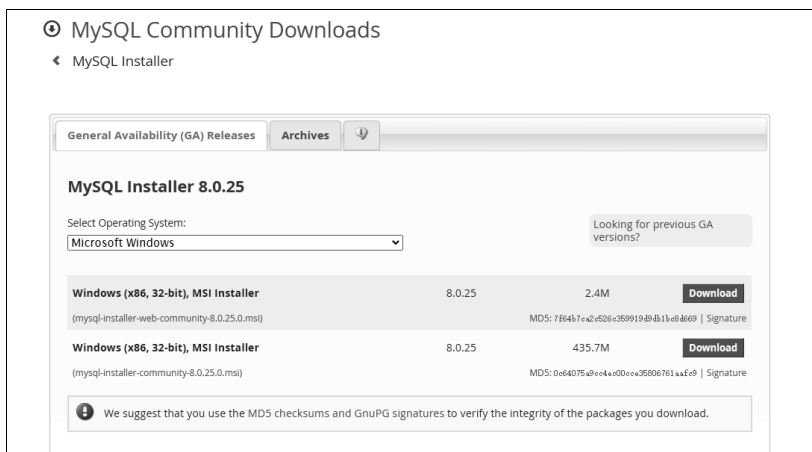


图 1-1 下载安装包

**步骤 02** 首次下载 MySQL 安装程序时，安装向导会引导我们完成 MySQL 产品的初始安装。首次安装需要进行初始设置，MySQL 安装程序在初始设置期间会检测主机上安装的现有 MySQL 产品，并将它们添加到要管理的产品列表中。如图 1-2 所示，当我们打开 MSI 文件之后，会提示我们选择安装的类型。

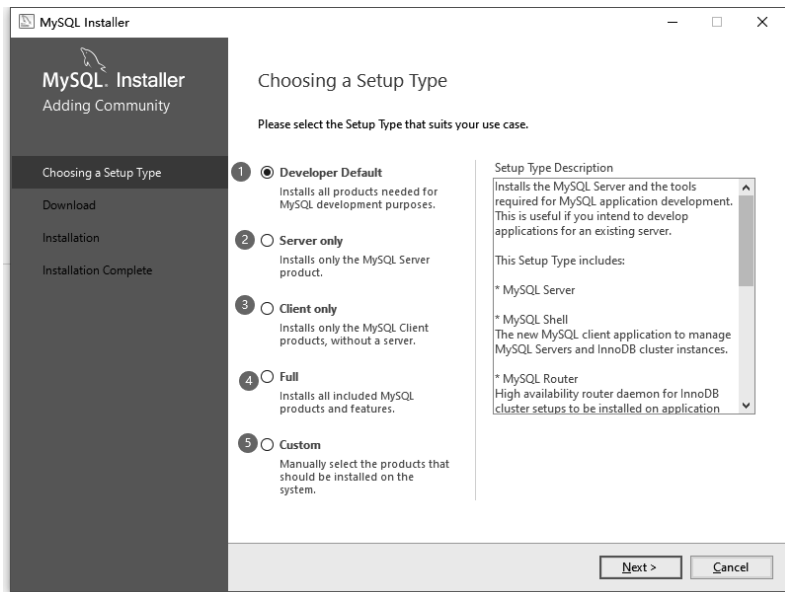


图 1-2 选择安装的类型

如图 1-2 所示，总共有 5 种安装类型。

① Developer Default(默认安装): 用于MySQL应用程序的开发。如果我们想要利用 MySQL 来进行应用程序的开发，可以选择此安装类型。

② Server only (仅安装服务器): 仅安装MySQL服务器。此安装类型在下载MySQL服务器安装包时会让用户选择安装GA版(一般可用性,即稳定版)或开发版。该安装类型使用默认的安装路径和数据存储路径。

③ Client only (仅安装客户端): 仅安装最新的MySQL应用程序和MySQL连接器。此安装类型类似于默认安装类型,不同之处在于它不包括MySQL服务器或通常与服务器捆绑的客户端程序,例如mysql或mysqldadmin。

④ Full (完整): 安装所有可用的 MySQL 产品。如果是初学者,那么可以选择此安装类型。

⑤ Custom (自定义): 此安装类型可以从 MySQL 安装程序目录中筛选想安装的各个 MySQL 产品。

**步骤 03** 选择好想要的安装类型,单击 Next 按钮,就会进入安装需求检查对话框,如图 1-3 所示。

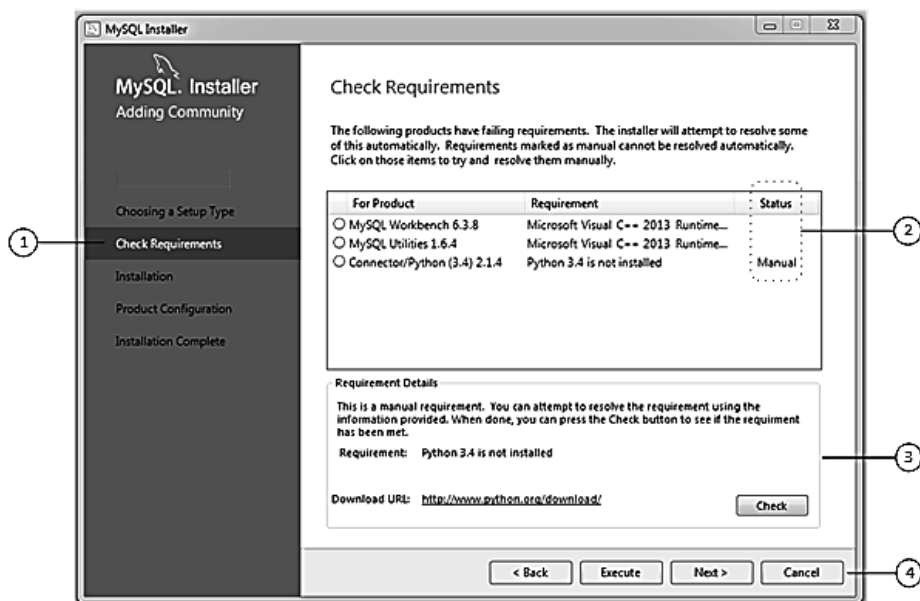


图 1-3 安装需求检查对话框

在安装需求检查对话框中包括如下内容:

① 显示初始设置中的当前步骤。读者在此列表中看到的步骤可能会略有不同,具体步骤取决于主机上已安装的产品、必备软件的可用性以及读者想要安装在主机上的产品。

② 按产品列出所有待确认的安装需求。

③ 安装需求的详细说明可以帮助我们解决安装前需要解决的问题。如果需要安装必备的软件，则该界面会提供下载用的网址（URL）。在下载并安装所需的软件后，单击 Check 按钮来验证是否已经满足安装需求。

④ 提供以下辅助安装的操作：

- Back: 返回上一步。此操作可以让我们回到上一步重新选择安装类型。
- Execute: 让 MySQL 安装程序尝试为所有选择的安装项目下载和安装必备的软件。
- Next: 不进行安装需求检查而继续安装产品，其中不包括未通过安装需求检查的产品。
- Cancel: 取消安装 MySQL 产品。

**步骤 04** MySQL 网络和身份认证配置，如图 1-4 所示。

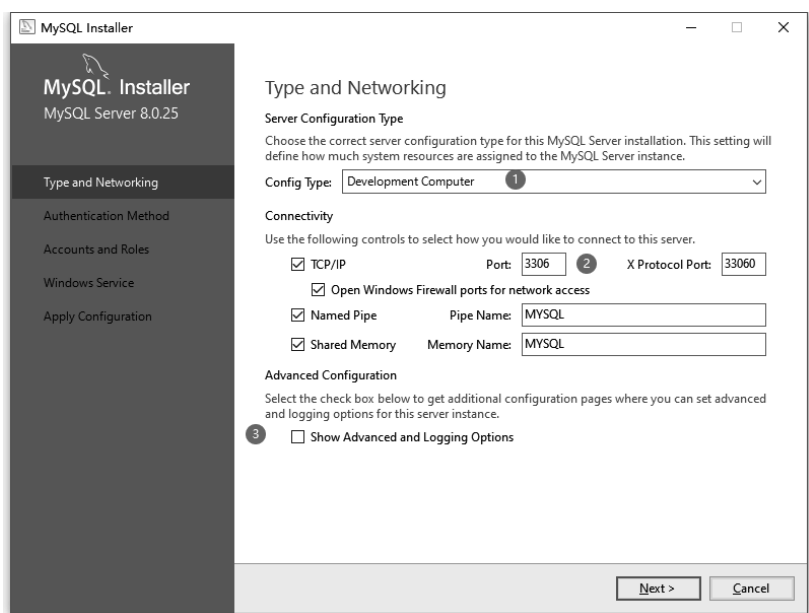


图 1-4 MySQL 网络配置

在该对话框中有以下选项：

- ① 选择需要配置的服务类型。
- ② 默认的服务端口号，①和②这两个配置可以不做修改。
- ③ 建议勾选 Show Advanced and Logging Options 复选框，以便在后续步骤中设置自定义日志记录和高级选项。

继续往下执行，会出现设置密码的对话框，如图 1-5 所示。

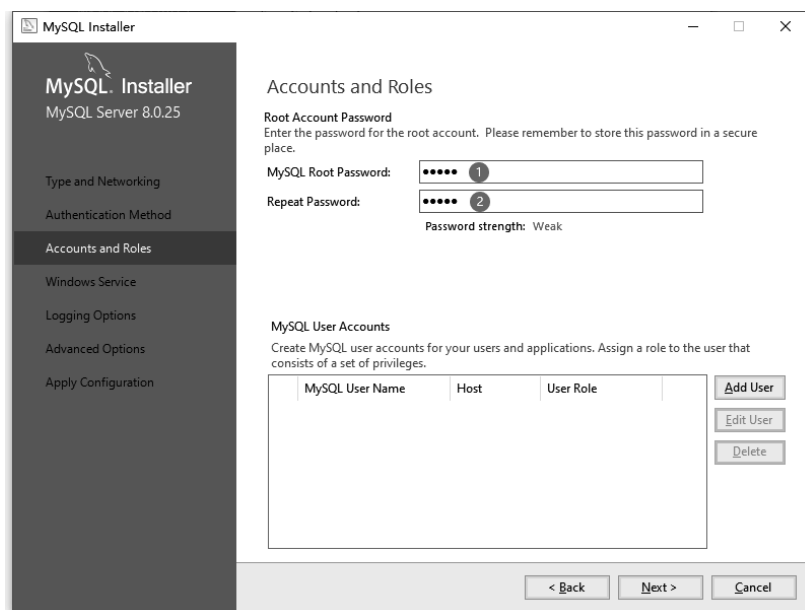


图 1-5 账户密码设置

该对话框包含以下内容：

- ① 给服务设置密码。
- ② 重复输入一次以确认密码，应当确保这两次输入的密码一致。

**步骤 05** MySQL 服务名称设置，如图 1-6 所示。

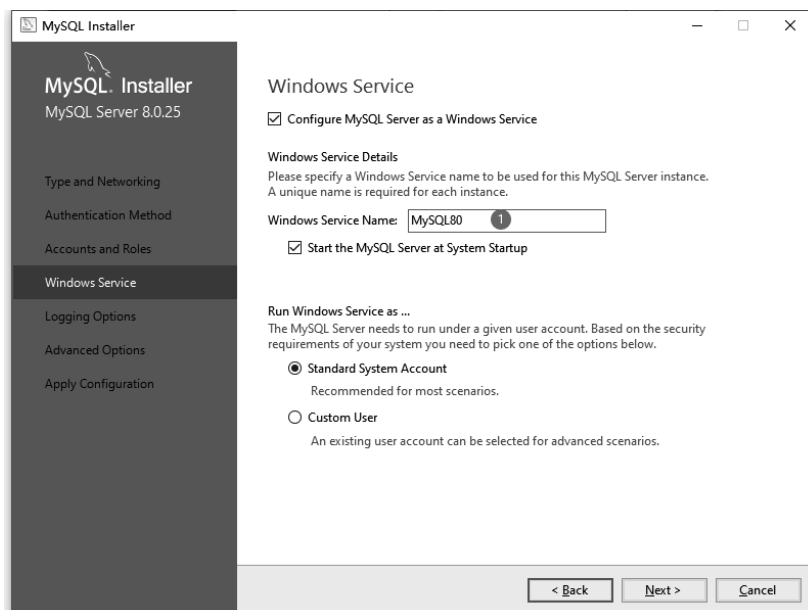


图 1-6 MySQL 服务名称设置

在该对话框中输入 MySQL 服务的名称，建议使用默认值。

**步骤 06** 设置 MySQL 服务相关的输出日志存储的路径，如图 1-7 所示。

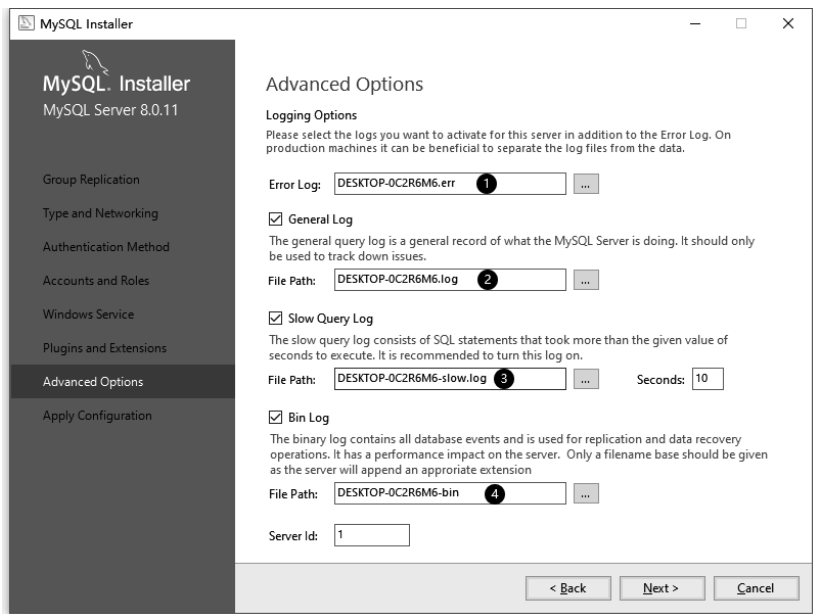


图 1-7 设置 MySQL 服务输出日志存储的路径

在该对话框中包含以下内容：

- ① 设置 MySQL 服务的错误日志输出路径。
- ② 设置一般的日志输出路径。
- ③ 设置查询日志的输出路径。
- ④ 设置二进制日志的输出路径。

**步骤 07** 验证是否安装成功。如图 1-8 所示，按住键盘上的 Windows 键，选择“运行”，然后在弹出的对话框中输入 services.msc，并按回车键。

接下来会出现如图 1-9 所示的服务窗口。如果可以看到 MySQL 8.0 这个服务，就表示 MySQL 数据库安装成功。



图 1-8 进入“运行”对话框



图 1-9 系统服务窗口

在服务成功启动之后，可以使用界面化工具 Navicat 连接数据库，当出现如图 1-10 所示的对话框时，输入主机名、端口号以及安装时设置的密码。在单击“测试连接”按钮之后，若出现连接成功的字样，则说明安装与配置都正确。



图 1-10 通过界面化工具测试连接

## 2. 在 Windows 上默认安装 MySQL 服务

表 1-1 所示是以默认安装类型安装 MySQL 服务后生成的相关目录，当我们需要查看相关信息或者数据日志时可能会用到。

表 1-1 默认安装 MySQL 服务之后生成的相关目录

目 录	目 录 内 容	备 注
bin	mysqld 服务器、客户端和应用程序	
%PROGRAMDATA%\MySQL\MySQL Server 8.0\	日志文件、数据库文件	Windows 系统变量%PROGRAMDATA%默认为 C:\ProgramData
docs	发布文档	使用 MySQL 安装程序，使用 Modify 操作可选择此文件夹
lib	服务相关的程序集	
share	字符集文件示例配置文件,用于数据库安装的 SQL	

### 1.2.3 在 Linux 平台下安装和配置 MySQL

使用 MySQL Yum 存储库安装最新的 MySQL GA 版本的步骤如下。

#### 1. 添加 MySQL Yum 存储库

要将 MySQL Yum 存储库添加到系统的存储库列表中，可以使用 RPM 命令来完成，具体步骤如下：

首先选择并下载适用于目标平台的安装包（发布包）。

我们可以查看目标系统的平台版本（操作系统版本序列号）。如图 1-11 所示，该平台的版本序列号为 7。

```
[root@localhost home]# cat /etc/redhat-release
CentOS Linux release 7.5.1804 (Core)
[root@localhost home]#
```

图 1-11 查看目标系统的操作系统版本序列号

然后下载工具 wget，下载完成之后用下面的命令进行安装：

```
yum install wget -y
```

接着下载对应版本的 RPM 文件，因为目前的平台是 7 系列，所以选择安装 7.3 版本的 RPM 文件。执行如下命令进行下载：

```
wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
```

当出现如图 1-12 所示的界面时，说明下载成功。

```
[root@localhost home]# wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
--2021-06-02 22:09:32-- https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 137.254.60.11
Connecting to dev.mysql.com (dev.mysql.com)|137.254.60.11|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://repo.mysql.com//mysql80-community-release-el7-3.noarch.rpm [following]
--2021-06-02 22:09:34-- https://repo.mysql.com//mysql80-community-release-el7-3.noarch.rpm
Resolving repo.mysql.com (repo.mysql.com)... 104.75.165.42
Connecting to repo.mysql.com (repo.mysql.com)|104.75.165.42|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26024 (25K) [application/x-redhat-package-manager]
Saving to: 'mysql80-community-release-el7-3.noarch.rpm.1'

100%[=====] 26,024  --.-K/s  in 0.1s

2021-06-02 22:09:34 (223 KB/s) - 'mysql80-community-release-el7-3.noarch.rpm.1' saved [26024/26024]
```

图 1-12 下载 RPM 包

## 2. 安装下载的发布包

安装命令如下：

```
yum install platform-and-version-specific-package-name.rpm
```

需要注意的是，`platform-and-version-specific-package-name` 为下载的 RPM 包的名称。而后执行如下命令：

```
yum install mysql80-community-release-el7-3.noarch.rpm -y
```

如果出现如图 1-13 所示的 Complete 说明，就表示发布包安装完成。

```
[root@worker1 ~]# yum install mysql80-community-release-el7-3.noarch.rpm -y
Loaded plugins: fastestmirror
Examining mysql80-community-release-el7-3.noarch.rpm: mysql80-community-release-el7-3.noarch
Marking mysql80-community-release-el7-3.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package mysql80-community-release.noarch 0:el7-3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================================
Package Arch Version Repository Size
=====================================================================================================================================
Installing:
mysql80-community-release noarch el7-3 /mysql80-community-release-el7-3.noarch 31 k
Transaction Summary
=====================================================================================================================================
Install 1 Package

Total size: 31 k
Installed size: 31 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql80-community-release-el7-3.noarch 1/1
  Verifying : mysql80-community-release-el7-3.noarch 1/1

Installed:
mysql80-community-release.noarch 0:el7-3

Complete!
```

图 1-13 发布包安装成功

如果使用的是其他系列的版本，那么可以根据实际情况选择安装相应的发布包。

对于基于 EL6 的系统，命令的形式为：

```
yum install mysql80-community-release-el6-{version-number}.noarch.rpm
```

对于基于 EL7 的系统，命令的形式为：

```
yum install mysql80-community-release-el7-{version-number}.noarch.rpm
```

对于基于 EL8 的系统，命令的形式为：

```
yum install mysql80-community-release-el8-{version-number}.noarch.rpm
```

对于 Fedora 34，命令的形式为：

```
dnf install mysql80-community-release-fc34-{version-number}.noarch.rpm
```

对于 Fedora 33，命令的形式为：

```
dnf install mysql80-community-release-fc33-{version-number}.noarch.rpm
```

### 3. 安装 MySQL

通过以下命令安装 MySQL：

```
yum install mysql-community-server -y
```

这条命令将安装 MySQL 服务器包（mysql-community-server）以及运行该服务器所需组件的包，包括客户端包（mysql-community-client）、客户端和服务器的常见错误消息和字符集（mysql-community-common）以及共享客户端库（mysql-community-libs）。如果出现如图 1-14 所示的信息，就表示安装成功。

```
Installed:
mysql-community-libs.x86_64 0:8.0.25-1.el7      mysql-community-libs-compat.x86_64 0:8.0.25-1.el7      mysql-community-server.x86_64 0:8.0.25-1.el7

Dependency Installed:
mysql-community-client.x86_64 0:8.0.25-1.el7      mysql-community-client-plugins.x86_64 0:8.0.25-1.el7      mysql-community-common.x86_64 0:8.0.25-1.el7
net-tools.x86_64 0:2.0-0.25.20131004git.el7        perl.x86_64 4:5.16.3-299.el7_9                          perl-Carp.noarch 0:1.26-244.el7
perl-Encode.x86_64 0:2.51-7.el7                  perl-Exporter.noarch 0:5.68-3.el7                          perl-File-Path.noarch 0:2.09-2.el7
perl-File-Temp.noarch 0:0.23.01-3.el7            perl-Filter.x86_64 0:1.49-3.el7                          perl-File-Path.noarch 0:2.09-2.el7
perl-HTTP-Tiny.noarch 0:0.033-3.el7              perl-Filter.x86_64 0:1.49-3.el7                          perl-Getopt-Long.noarch 0:2.40-3.el7
perl-Pod-Perldoc.noarch 0:3.20-4.el7            perl-PathTools.x86_64 0:3.40-5.el7                        perl-Pod-Escapes.noarch 1:1.04-299.el7_9
perl-Scalar-List-Utils.x86_64 0:1.27-248.el7     perl-Pod-Simple.noarch 1:3.28-4.el7                       perl-Pod-Usage.noarch 0:1.63-3.el7
perl-Text-ParseWords.noarch 0:3.29-4.el7         perl-Socket.x86_64 0:2.010-5.el7                         perl-Storable.x86_64 0:2.45-3.el7
perl-Text-ParseWords.noarch 0:3.29-4.el7         perl-Time-HiRes.x86_64 4:1.9725-3.el7                    perl-Time-Local.noarch 0:1.2300-2.el7
perl-constant.noarch 0:1.27-2.el7               perl-Libs.x86_64 4:5.16.3-299.el7_9                      perl-macros.x86_64 4:5.16.3-299.el7_9
perl-parent.noarch 1:0.225-244.el7              perl-podlators.noarch 0:2.5.1-3.el7                      perl-threads.x86_64 0:1.87-4.el7
perl-threads-shared.x86_64 0:1.43-6.el7

Dependency Updated:
postfix.x86_64 2:2.10.1-9.el7

Replaced:
mariadb-libs.x86_64 1:5.5.56-2.el7

Complete!
[root@localhost ~]#
```

图 1-14 安装 MySQL

### 4. 启动 MySQL 服务器

使用以下命令启动 MySQL 服务器：

```
systemctl start mysqld
```

使用以下命令检查 MySQL 服务器的状态：

```
systemctl status mysqld
```

查询服务的状态，显示 running 时表示服务已经启动，如图 1-15 所示。

```
[root@localhost ~]# systemctl status mysqld
● mysqld.service - MySQL Server
  Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
  Active: active (running) since Mon 2021-05-31 23:51:45 EDT; 8s ago
    Docs: man:mysqld(8)
          http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 4502 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
 Main PID: 4631 (mysqld)
  Status: "Server is operational"
   Tasks: 38
  Memory: 470.1M
   CGroup: /system.slice/mysqld.service
           └─4631 /usr/sbin/mysqld

May 31 23:51:36 localhost.localdomain systemd[1]: Starting MySQL Server...
May 31 23:51:45 localhost.localdomain systemd[1]: Started MySQL Server.
[root@localhost ~]#
```

图 1-15 MySQL 服务状态

需要注意的是，如果操作系统已启用 `systemd`，则应使用标准 `systemctl`（或 `service`）命令（例如 `stop`、`start`、`status` 和 `restart`）来管理 MySQL 服务器的服务。`mysqld` 服务默认是启用的（在系统重新启动时启用了）。

## 5. 登录 MySQL

在安装 MySQL 服务时系统创建了一个超级用户账户（`root`）。超级用户的密码已设置并存储在错误日志文件中，如果我们想要知道密码，可以使用如下命令进行查询：

```
grep 'temporary password' /var/log/mysqld.log
```

如图 1-16 所示，最后输出的字符串就是想要查看的密码。

```
[root@localhost ~]# grep 'temporary password' /var/log/mysqld.log
2021-06-01T03:51:40.448706Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: T<kB=RL_05f
[root@localhost ~]#
```

图 1-16 想要查看的密码

我们使用生成的临时密码登录并为超级用户账户设置自定义密码，以便尽快更改 `root` 密码。首先执行下面的命令进行登录：

```
mysql -uroot -p
```

如图 1-17 所示，在登录过程中需要输入密码，可以输入之前查询到的密码进行登录。

```
[root@localhost ~]# mysql -uroot -p
Enter password: 
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.25

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

图 1-17 MySQL 登录

然后执行如下命令修改密码:

```
alter user 'root'@'localhost' identified by 'NewPass';
```

在修改密码时注意密码要尽量复杂, 需要包含数字、大小写字母和标记符号, 否则系统会出现如图 1-18 所示的提示, 提示密码修改出错。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPass';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPass123456';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPass!';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPass!234567';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

图 1-18 修改 MySQL 账户的密码

## 1.2.4 在 Docker 平台下安装和配置 MySQL

Docker 是一个用于开发、交付和运行应用程序的开放平台。Docker 能够将应用程序与基础架构分开, 从而快速交付软件。借助 Docker, 可以采用与管理应用程序相同的方式来管理基础架构。

Docker 是当下热门的容器, 为了省去 Linux 下安装配置程序烦琐且易于出错的步骤, 笔者增加了本节内容, 安装 Docker 环境后, 就能在其下快速地安装 Redis 服务。

### 1. 在 CentOS 下安装 Docker

安装 Docker 的系统需求是必须具备一个 CentOS 7 或者 CentOS 8 的维护版本, 不支持之前的旧版本。

#### 1) 卸载旧版本

较旧的 Docker 版本称为 docker 或 docker-engine。如果安装过这些程序, 请卸载它们及其相关的依赖项。

```
$ sudo yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine
```

#### 2) 更新系统 yum 工具包

需要安装 yum-utils 软件包:

```
$ sudo yum install -y yum-utils
```

### 3) 设置稳定的存储库 (可选择其中一个)

可以根据自己的网络情况选择存储库地址, 推荐使用阿里云和清华大学源, 如果设置官方源, 那么需要单独设置好网络才可以进行正常操作。

使用官方源地址:

```
$ sudo yum-config-manager \
  --add-repo \
  https:// download.docker.com/linux/centos/docker-ce.repo
```

使用阿里云:

```
$ sudo yum-config-manager \
  --add-repo \
  http:// mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

使用清华大学源:

```
$ sudo yum-config-manager \
  --add-repo \
  https:// mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/docker-ce.repo
```

### 4) 安装 Docker Engine-Community

安装社区版 Docker 包:

```
$ sudo yum install docker-ce docker-ce-cli containerd.io
```

### 5) 启动 Docker

启动 Docker 很简单:

```
$ sudo systemctl start docker
```

### 6) 验证是否正确安装

可以使用如下命令来验证 Docker 是否正确安装:

```
$ sudo docker run hello-world
```

## 2. 在 Windows 10 下安装 Docker

Docker Desktop 是 Docker 在 Windows 10 和 macOS 操作系统上的官方安装方式, 这种方式依然是先在虚拟机中安装 Linux, 再安装 Docker。用户可以从网上搜索 Docker Desktop 的安装软件 `docker-ce-desktop-windows` (此方法仅适用于 Windows 10 操作系统专业版、企业版、教育版和部分家庭版)。

### 1) 安装 Hyper-V 虚拟机

Hyper-V 是微软开发的虚拟机, 类似于 VMWare 或 VirtualBox, 只是 Hyper-V 仅适用于 Windows 10。这是 Docker Desktop for Windows 所使用的虚拟机 (这个虚拟机一旦启用, VirtualBox、VMWare、Workstation 15 及以下版本将无法使用。如果必须在计算机上使用其他虚拟机, 则不要在 Windows 中启动 Hyper-V)。

## 2) 启动 Hyper-V

启动 Hyper-V 的操作步骤如下:

**步骤 01** 在 Windows 中右击“开始”按钮，在弹出的快捷菜单中选择“应用和功能”命令，如图 1-19 所示。

**步骤 02** 打开“程序和功能”窗口，单击“启用或关闭 Windows 功能”链接，如图 1-20 所示。



图 1-19 选择“应用和功能”命令

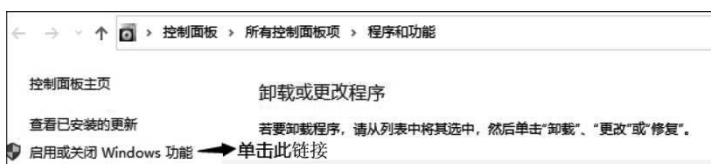


图 1-20 “程序和功能”窗口

**步骤 03** 在打开的“Windows 功能”对话框中勾选“Hyper-V”复选框，然后单击“确定”按钮，如图 1-21 所示。

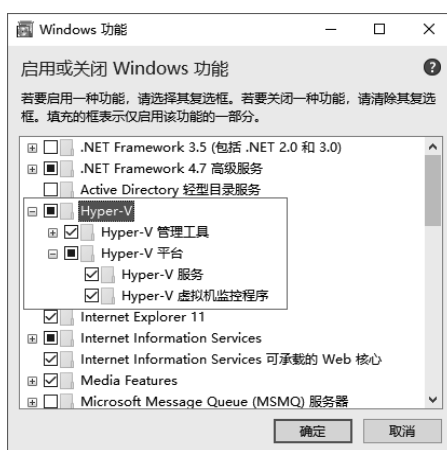



图 1-21 勾选“Hyper-V”复选框

### 3) 安装 Docker

接下来双击下载的安装 Docker 的 EXE 文件，在打开的对话框中单击 Next 按钮，最后单击 Finish 按钮完成安装。

安装完成后，Docker 会自动启动，通知栏上会出现图标，表示 Docker 正在运行，然后按 Win+R 快捷键，在弹出的“运行”窗口中输入 PowerShell。

用户还可以执行命令 `docker run hello-world` 来检查是否安装成功。

## 3. 利用 Docker 安装 MySQL

### 1) 下载 MySQL 镜像

执行 `docker pull mysql/mysql-server:8.0` 下载 MySQL 镜像，如图 1-22 所示。

```
[root@localhost ~]# docker pull mysql/mysql-server:8.0
8.0: Pulling from mysql/mysql-server
ee9d4d67fb68: Pull complete
151f7d1c9a3e: Pull complete
b01941651e65: Pull complete
64eb680a58e2: Pull complete
05c4ac61c33c: Pull complete
1c44925848bc: Pull complete
fff02a04d6b8: Pull complete
Digest: sha256:56ec3d7509327c66e4b8b22c72ecd56572ae1f87c91ef806c80fa09c7707c845
Status: Downloaded newer image for mysql/mysql-server:8.0
docker.io/mysql/mysql-server:8.0
```

图 1-22 下载镜像

### 2) 启动 MySQL 服务

执行如下命令启动 MySQL 服务：

```
docker run --name=mysql11 --restart on-failure -d mysql/mysql-server:8.0
```

### 3) 其他操作

删除容器的命令如下：

```
docker rm -f mysql11
```

停止容器的命令如下：

```
docker stop mysql11
```

### 4) 查看启动的日志

查看启动日志的命令如下：

```
docker logs mysql11
```

查看 MySQL 服务的密码的命令如下：

```
docker logs mysql11 2>&1 | grep GENERATED
```

查询到的密码如图 1-23 所示。

```
[root@localhost ~]# docker logs mysql11 2>&1 | grep GENERATED
[Entrypoint] GENERATED ROOT PASSWORD: W;3FK_r+DW92r^^62w4L,ZA^mq=1BJX7
[root@localhost ~]#
```

图 1-23 查询到的密码

### 5) 登录 MySQL 服务

登录 MySQL 服务的命令如下:

```
docker exec -it mysql11 mysql -uroot -p
```

当系统询问时, 输入生成的 root 密码。由于该 `MYSQL_ONETIME_PASSWORD` 选项默认为 `true`, 因此在将 MySQL 客户端连接到服务器后, 必须通过以下语句来重置服务器 root 账户的密码:

```
alter user 'root'@'localhost' identified by 'password';
```

或者执行下面的命令进行安装:

```
docker run -p 3306:3306 --name mysql \  
-v /root/mysql/log:/var/log/mysql \  
-v /root/mysql/data:/var/lib/mysql \  
-v /root/mysql/conf:/etc/mysql \  
-e MYSQL_ROOT_PASSWORD=123456 \  
-d mysql/mysql-server:8.0
```

以上命令挂载了服务的日志文件和 lib 文件, 已经设置好了密码。需要注意的是, 在执行这条命令之前, 要在宿主机的 `/root/mysql/conf` 目录下创建 `my.cnf` 文件, 这个文件是服务启动的配置文件, 该配置文件至少应该有如下几行内容:

```
[mysqld]  
init_connect='set collation_connection = utf8_general_ci'  
init_connect='set names utf8'  
secure_file_priv=
```

## 1.3 如何学习 MySQL

在学习 MySQL 数据库之前, 很多读者会问, 如何才能更好地学习 MySQL 的相关技能呢? 下面就讲述学习 MySQL 的方法。

### 1. 培养兴趣

兴趣是最好的老师, 不论学习什么知识, 兴趣都可以极大地提高学习效率, 学习 MySQL 也不例外。

### 2. 扎实基础

随着技术的不断深入, 只有有着扎实的基础功底, 才能在技术道路上走得更远。对于 MySQL 的学习而言, SQL 语句是其中最基础的部分, 很多功能都通过 SQL 语句来实现, 所以读者在学习的过程中要多编写 SQL 语句, 最好能对同一功能使用不同的 SQL 语句来实现。

## 1.4 本章练习

1. 下载并安装 MySQL 数据库。
2. 使用配置向导配置 MySQL 为系统服务，并设置为手动启动或者关闭 MySQL 服务。

# 第 2 章

## 数据库设计

### 本章内容：

- 数据库设计理论
- 连接数据库
- 创建数据库和表



本章将讲解数据库设计理论，以简单的示例讲解三大范式以及 MySQL 中对数据库和表的基本操作（包括创建数据库、切换数据库、创建表等）。

### 2.1 关系数据库设计理论三大范式

关系模型（Relational Model）是一种基于表的数据模型，它广泛应用于数据库管理系统（DBMS）中。关系模型的核心思想是将数据组织成表，每个表包含一组相关的行和列。这些行和列之间的关系通过主键和外键来定义，从而实现数据的完整性、一致性和可维护性。关系模型中一些重要术语介绍如下：

- 属性（Attribute）：列的名称。
- 关系（Relation）：列属性之间存在的某种关联。
- 表（Table）：由多个属性以及众多元组所表示的各个实例组成。
- 键（Key）：由一个或多个属性组成，其值能唯一标识关系中的一个元组。如果某个关系 A 中的一个（组）属性是另一个关系 B 的键，则该（组）属性在 A 中被称为外键。
- 笛卡儿积（交叉连接，Cross Join）：第一个关系的每一行数据分别与第二个关系的每一行数据组合。
- 自然连接（Natural Join）：第一个关系的每一行数据与第二个关系的每一行数据进行

匹配，若得到交叉部分则合并，若无交叉部分则舍弃。

- $\theta$  连接 (Theta Join): 加上约束条件的笛卡儿积，先得到笛卡儿积，再根据约束条件删除不满足条件的元组。
- 外连接 (Outer Join): 执行自然连接后，将舍弃的部分也加入，并且把匹配失败处的属性设置为 null。

表 2-1 是一张酒店客人入住信息表。

表2-1 酒店客人入住信息表

客人编号	姓名	地址	.....	客房号	客房描述	客房类型	客房状态	床位数	价格(元)	入住人数
C1001	张三	Addr1	.....	1001	A 栋 1 层	单人间	入住	1	128.00	1
C1002	李四	Addr2	.....	2002	B 栋 2 层	标准间	入住	2	168.00	0
C1003	王五	Addr3	.....	2002	B 栋 2 层	标准间	入住	2	168.00	2
C1004	赵六	Addr4	.....	2003	B 栋 2 层	标准间	入住	2	158.00	1
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
C8006	A1	Addrm	.....	101	C 栋 3 层	总统套房	入住	3	1080.00	1
C8008	A2	Addrn	.....	101	C 栋 3 层	总统套房	空闲	3	1080.00	0

通过表 2-1 可以发现，该表存在不符合规范的设计，总共有如下几点：

- (1) 信息重复：比如客房类型和客房状态存在大量的数据重复。
- (2) 更新异常：修改了一个记录中的信息，但是另一个记录中相同的信息却没有被同步修改。
- (3) 插入异常：无法正确表示信息。
- (4) 删除异常：丢失有效信息。

下面利用三大范式对数据库进行改造。

**第一范式 (1NF)**：目标是确保每列的原子性，如果每列都是不可再分的最小数据单元（也被称为最小的原子单元），则满足第一范式。

**例 2.1** 第一范式优化表格示例。

没有经过第一范式优化的表格如表 2-2 所示。

表2-2 没有经过第一范式优化的表格

Address
中国北京
美国纽约
英国利物浦
日本东京
.....

经过第一范式优化的表格如表 2-3 所示。

表 2-3 经过第一范式优化的表格

CustID	Country	City
1	中国	北京
3	英国	利物浦
4	日本	东京
2	美国	纽约
.....	.....	.....

**第二范式 (2NF)：**要求每张表只描述一件事情。

**例 2.2** 第二范式优化表格示例。

没有经过第二范式优化的表格如表 2-4 所示。

表 2-4 没有经过第二范式优化的表格

字 段	例 子
客人编号	C1002
姓名	李四
地址	Addr1
客房号	2002
客房状态	入住
客房类型	标准间
床位数	2
入住人数	2
价 格	168
.....	.....

经过第二范式优化之后，把一张表拆解成两张表，分别为 Guest 表和 Room 表。

Guest 表如表 2-5 所示。

表 2-5 Guest 表

字 段	例 子
客人编号	C1002
姓名	李四
地址	Addr1
客房号	2002

Room 表如表 2-6 所示。

表 2-6 Room 表

字 段	例 子
客房号	2002
客房状态	入住
客房类型	标准间
床位数	2
入住人数	2

**第三范式 (3NF)**: 如果一个关系满足第二范式, 并且除了主键以外的其他列都不依赖于主键列, 则满足第三范式。

### 例 2.3 第三范式优化表格示例。

没有经过第三范式优化的 Room 表如表 2-7 所示。

表2-7 没有经过第三范式优化的Room表

字 段	例 子
客房号	2002
客房状态	入住
客房类型	标准间
床位数	2
入住人数	2
价格	168
.....	.....

经过第三范式优化之后, 拆成 3 张表, 分别是 Room 表、RoomType 表和 RoomState 表。Room 表如表 2-8 所示。

表2-8 Room表

字 段	例 子
客房号	2002
客房状态编号	1
客房类型编号	2
入住人数	2
.....	.....

RoomType 表如表 2-9 所示。

表2-9 RoomType表

字 段	例 子
客房类型编号	2
客房类型名称	标准间
床位数	2
价格	168

RoomState 表如表 2-10 所示。

表2-10 RoomState表

字 段	例 子
客房状态编号	1
客房状态名称	入住

如上所述，经过三大范式将一张存在问题的表拆分成 4 张表，最终数据库设计结构如图 2-1 所示（PK 表示主键，FK 表示外键）。

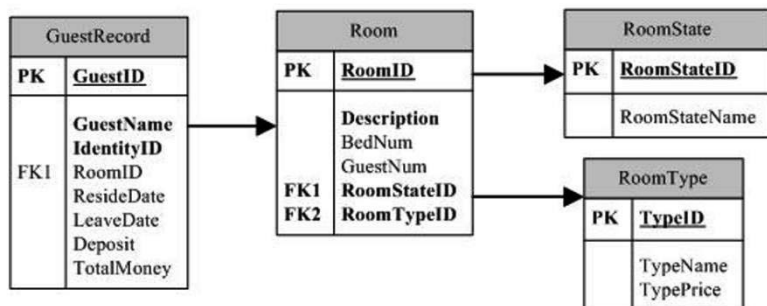


图 2-1 酒店管理系统数据库模型

## 2.2 连接数据库和基本信息查询

数据库设计好了，本节就来连接数据库并进行基本信息查询。

### 2.2.1 连接数据库

要连接到 SQL 服务器，需要在调用 MySQL 时提供用户名，并且很可能需要提供密码。如果 SQL 服务器在其他计算机上运行，还必须指定主机名。此时应联系管理员，了解该使用哪些连接参数来进行连接（即使用的主机、用户名和密码）。当知道正确的参数之后，可以执行如下命令进行连接：

```
shell> mysql -h host -u user -p
Enter password: *****
```

host 和 user 分别代表运行的 MySQL 服务器的主机名和 MySQL 账户的用户名。\*\*\*\*\* 代表密码，当 MySQL 显示“Enter password:”提示时，输入账户对应的密码。如果输入密码正确，那么会看到如下信息：

```
shell> mysql -h host -u user -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25338 to server version: 8.0.25-standard
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

出现“mysql>”提示符就表示可以输入 SQL 语句了。

如果在运行 MySQL 的同一台计算机上登录，则可以省略主机名，只需执行以下命令：

```
shell> mysql -u user -p
```

如果在尝试登录时收到错误提示信息，例如 ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)，则表示 MySQL 服务器守护程序（在 UNIX 操作系统中）或服务（在 Windows 操作系统中）未运行，也就是需要启动 MySQL 服务。

## 2.2.2 基本信息查询

本小节将介绍信息查询的基本原则，通过几个查询示例来熟悉 MySQL 的工作原理。

**例 2.4** 查询服务器的版本号和当前日期。在“mysql>”提示符后面输入如下命令，然后按 Enter 键：

```
mysql> select version(), current_date;
+-----+-----+
| version() | current_date |
+-----+-----+
| 8.0.25    | 2021-06-01  |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

这是一个简单的查询，说明了有关 MySQL 查询的几点信息：

(1) 查询通常由以分号结尾的 SQL 语句组成（有一些例外可以省略分号，QUIT 语句就是其中之一）。

(2) 当我们输入查询语句后，MySQL 将它发送到服务器执行并显示结果，然后显示出下一个“mysql>”提示符，表明它已准备好接收另一个查询。

(3) MySQL 以表格形式（行和列）显示查询的结果。第一行是列的标签（或称为列名），其他行是查询的结果。通常列标签是从数据库表中提取的列名称。如果要检索表达式的值而不是表列的值，那么 MySQL 会使用表达式本身来标记该列。

(4) MySQL 显示返回了多少行数据以及执行查询用了多长时间，由此我们大致可以了解服务器的性能。不过这些值是不精确的，因为它们代表的是时钟时间，并且受到服务器负载和网络延迟等因素的影响。

**例 2.5** 使用不同字母大小写进行相同内容的查询。

```
mysql> SELECT VERSION(), CURRENT_DATE;
mysql> select version(), current_date;
mysql> SeLeCt vErSiOn(), current_DATE;
```

查询结果如图 2-2 所示，在查询时，输入的 SQL 语句中的关键字可以不区分字母大小写，最后输出的

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 8.0.25    | 2021-06-01  |
+-----+-----+
1 row in set (0.00 sec)

mysql> select version(), current_date;
+-----+-----+
| version() | current_date |
+-----+-----+
| 8.0.25    | 2021-06-01  |
+-----+-----+
1 row in set (0.00 sec)

mysql> SeLeCt vErSiOn(), current_DATE;
+-----+-----+
| vErSiOn() | current_DATE |
+-----+-----+
| 8.0.25    | 2021-06-01  |
+-----+-----+
1 row in set (0.00 sec)
```

图 2-2 SQL 关键字不区分字母大小写示例

答案是一致的，也不会抛出任何异常。

### 例 2.6 把 MySQL 用作计算器。

```
mysql> select sin(pi()/5), (9+1)*5;
+-----+-----+
| sin(pi()/5) | (9+1)*5 |
+-----+-----+
| 0.5877852522924731 | 50 |
+-----+-----+
1 row in set (0.00 sec)
```

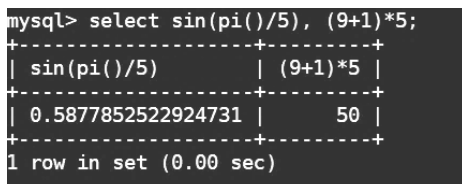
执行结果如图 2-3 所示，MySQL 给出了计算结果。

### 例 2.7 用分号分隔多条 SQL 语句。

```
mysql> select version(); select now();
+-----+
| version() |
+-----+
| 8.0.25 |
+-----+
1 row in set (0.00 sec)

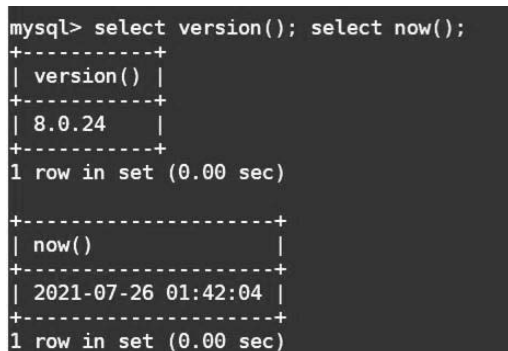
+-----+
| now() |
+-----+
| 2021-06-01 06:49:51 |
+-----+
1 row in set (0.00 sec)
```

执行结果如图 2-4 所示，可以在一行中输入多条 SQL 语句（或命令），每条 SQL 语句之间只需用分号分隔即可。



```
mysql> select sin(pi()/5), (9+1)*5;
+-----+-----+
| sin(pi()/5) | (9+1)*5 |
+-----+-----+
| 0.5877852522924731 | 50 |
+-----+-----+
1 row in set (0.00 sec)
```

图 2-3 MySQL 作为计算器



```
mysql> select version(); select now();
+-----+
| version() |
+-----+
| 8.0.24 |
+-----+
1 row in set (0.00 sec)

+-----+
| now() |
+-----+
| 2021-07-26 01:42:04 |
+-----+
1 row in set (0.00 sec)
```

图 2-4 一行执行多条 SQL 语句

### 例 2.8 MySQL 可以写成多行的形式，最后带上终止分号即可。

```
mysql> select
-> user()
-> ,
-> current_date;
```

```

+-----+-----+
| user()          | current_date |
+-----+-----+
| root@localhost | 2021-06-01   |
+-----+-----+
1 row in set (0.00 sec)

```

上述 SQL 语句的执行结果如图 2-5 所示。如果一条 SQL 查询语句冗长，在一行内写不下，那么可以写成多行的形式，最后带上终止分号即可，因为 MySQL 不是在每一行的末尾去找代表 SQL 语句结束的分号，而是在多个连续行去找分号来确定语句的结束位置。（MySQL 接收自由格式的输入，在看到分号之后才会执行当前的 SQL 查询语句），当我们不输入分号时，之前输入的命令是不会被执行的。

### 例 2.9 取消查询。

```

mysql> select
-> users
-> \c
mysql>

```

执行结果如图 2-6 所示，当输入“\c”之后，当前输入的命令就被取消执行了。

```

mysql> select
-> users
-> \c
mysql> select
-> user()
-> ,
-> current_date;
+-----+-----+
| user()          | current_date |
+-----+-----+
| root@localhost | 2021-07-26   |
+-----+-----+
1 row in set (0.00 sec)

```

图 2-5 多行查询

```

mysql> select
-> users
-> \c
mysql>

```

图 2-6 取消执行当前输入的 SQL 命令

需要注意的是，在输入取消命令“\c”之后，MySQL 回到“mysql>”提示符状态，表示已准备好接收新的查询。表 2-11 总结了 MySQL 显示不同提示符时所处的状态。

表 2-11 MySQL 所处状态的含义

提示	意义和说明
mysql>	准备就绪，等待接收新的查询
->	等待多行查询的下一行
'>	等待下一行，等待以单引号（'）开头的字符串完成
">	等待下一行，等待以双引号（"）开头的字符串完成
`>	等待下一行，等待以反引号（`）开头的标识符完成
/*>	等待下一行，等待以“/*”开头的注释完成

(1) 当以单行输入查询语句时，如果忘记以分号结尾，那么 MySQL 会一直等待我们输

入“;”：

```
mysql> select user()
->
```

如果发生这种情况(我们认为已经输入完成一条查询语句,而唯一的响应是“->”提示符),那么很可能就是 MySQL 在等待分号。输入分号之后,MySQL 才会开始执行输入的 SQL 查询语句:

```
mysql> select user()
-> ;
+-----+
| user() |
+-----+
| root@localhost |
+-----+
```

(2) “>”和“>”提示符表示当前正处于字符串的收集过程中(即表示 MySQL 正在等待用户输入表示字符串终止的配对符号)。在 MySQL 中,我们可以编写由一对“'”或一对“”作为起止的字符串,而 MySQL 允许输入跨多行的字符串。当看到“>”或“>”提示符时,表示输入了包含以“'”或“”开头的字符串,但尚未输入终止字符串的配对引号。例如:

```
mysql> select * from mytable where name = 'clay and age <18;
'>
```

如果输入此 select 语句,然后按 Enter 键并等待结果,则没有任何反应。此时我们需要注意“>”提示符提供的线索。它其实是告诉我们:MySQL 希望看到字符串的其余部分(语句中 clay 开始的字符串缺少了第二个单引号)。此时我们可以选择输入“\c”取消本次查询语句的执行:

```
mysql> select * from mytable where name = clay and age < 18;
'> '\c
mysql>
```

提示符又变回“mysql>”,表明 MySQL 再次就绪,可以接收新的查询。

## 2.3 创建数据库和表

在学习如何创建数据库和表之前,可以使用如下步骤来查询当前服务器中存在的数据库:

**步骤 01** 使用 show 语句查询当前服务器上存在哪些数据库:

```
mysql> show databases;
+-----+
| database |
+-----+
| information_schema |
| mysql |
| performance_schema |
```

```

| sys |
+-----+
4 rows in set (0.00 sec)

```

从以上信息可以了解到，当我们执行 `show databases` 语句后，服务器将查询当前存在的所有数据库。

`mysql` 数据库存储了用户访问权限等信息。`test` 数据库是系统提供的默认测试库，可以删除。上述语句显示的数据库列表可能和读者计算机上显示的数据库列表有所不同，因为不会显示当前用户没有访问权限的数据库。

**步骤 02** 如果 `mysql` 数据库存在，那么我们可以进入 `mysql` 数据库查询当前数据库中的表。比如，使用如下语句可以切换到 `mysql` 数据库：

```

mysql> use mysql
Database changed

```

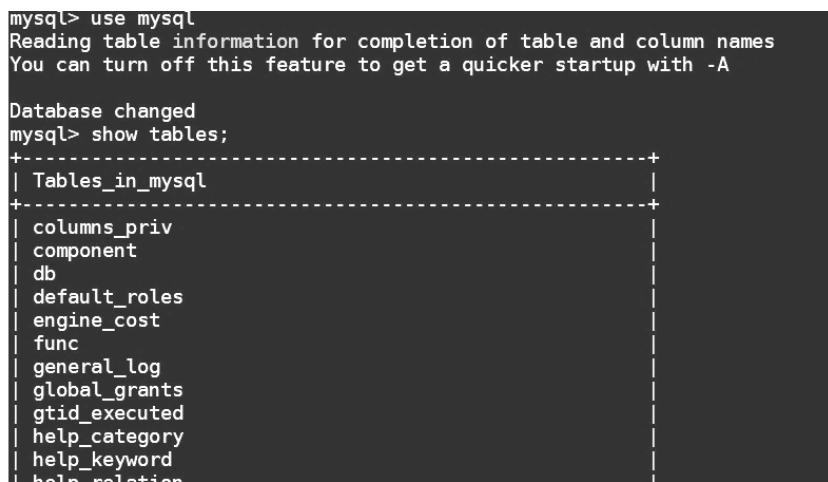
使用如下语句可以查询当前数据库中的所有表：

```

mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| component |
| db |

```

执行结果如图 2-7 所示，首先切换到 `mysql` 数据库，然后查询出当前 `mysql` 数据库中的所有表。



```

mysql> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| component |
| db |
| default_roles |
| engine_cost |
| func |
| general_log |
| global_grants |
| gtid_executed |
| help_category |
| help_keyword |
| help_relatio

```

图 2-7 查询 `mysql` 数据库中的所有表

### 2.3.1 创建数据库

如果要创建一个新的数据库，那么可以执行如下语句：

```
create database DEMO;
```

执行结果如图 2-8 所示，创建数据库成功。

在 UNIX 操作系统中，数据库的名称是区分字母大小写的（这与 SQL 关键字不区分字母大小写不同），因此在进入数据库时，必须使用数据库的名称 DEMO 来指向正确的数据库，而不能使用 Demo、demo。对于表名也是如此（注意，在 Windows 中此限制不适用）。如图 2-9 所示，切换数据库时，由于未正确使用大小写而导致切换数据库操作失败。

如图 2-10 所示，唯有数据库名称的大小写正确才能正常完成数据库的切换。

```
mysql> create database DEMO;
Query OK, 1 row affected (0.01 sec)

mysql> use demo;
ERROR 1049 (42000): Unknown database 'demo'
```

图 2-9 数据库切换操作失败

```
mysql> create database DEMO;
Query OK, 1 row affected (0.01 sec)

mysql>
```

图 2-8 创建数据库

```
mysql> use DEMO;
Database changed

mysql>
```

图 2-10 数据库切换操作成功

如果在创建数据库时出现诸如 ERROR 1044 (42000): Access denied for user clay@'localhost' 的错误，则说明用户没有创建数据库所需的权限。

## 2.3.2 创建表

如图 2-11 所示，进入 DEMO 数据库中，此时该数据库中并没有任何表。

如果要在 DEMO 数据库中创建表，那么在创建之前需要了解如下信息：

- 表名：代表数据表的名称，如果我们想要存储用户信息，则可以取名为 userinfo。
- 表字段名：表示表中有哪些属性，比如 name、age、address 等字段。

以下为创建 MySQL 数据表的通用语法：

```
create table table_name (column_name column_type);
```

**例 2.10** 在 DEMO 数据库中创建数据表 userinfo，SQL 语句如下：

```
create table if not exists `userinfo` (
  `id` int unsigned auto_increment,
  `name` varchar(100) not null,
  `age` int not null,
  `date` date,
  primary key ( `id` ))engine=innodb default charset=utf8;
```

上述语句说明如下：

- 如果不希望字段的值为 null，那么可以将字段的属性设置为 not null，如果在操作数据

```
mysql> use DEMO;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

图 2-11 切换数据库并查询所有表

库时将 null 输入到该字段，则会报错。

- auto\_increment: 把列定义为自增的属性，一般用于主键，数值会自动加 1。
- primary key: 用于把列定义为主键，可以使用多列来定义主键，列之间以逗号分隔。
- engine: 设置存储引擎。
- charset: 设置字符集的编码。
- varchar(100)与 int: 代表此字段使用的数据类型，后续章节会专门讲解每一种数据类型。

创建结果如图 2-12 所示。

```
mysql> create table if not exists `userinfo` (
->   `id` int unsigned auto_increment,
->   `name` varchar(100) not null,
->   `age` int not null,
->   `date` date,
->   primary key ( `id` ))engine=innodb default charset=utf8;
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

图 2-12 创建表

如果想查询此表的结构，可以执行如下语句：

```
describe userinfo;
```

执行结果如图 2-13 所示，可以清晰地看到此表的结构及其说明。

```
mysql> describe userinfo;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int unsigned  | NO   | PRI | NULL    | auto_increment |
| name  | varchar(100)  | NO   |     | NULL    |                |
| age   | int           | NO   |     | NULL    |                |
| date  | date          | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

图 2-13 查询表的结构及其说明

此后若再次查询当前数据库中的表，则会显示已创建表的基本信息，如图 2-14 所示。

```
mysql> show tables;
+-----+
| Tables_in_DEMO |
+-----+
| userinfo        |
+-----+
1 row in set (0.00 sec)
```

图 2-14 查询数据库中表的基本信息

## 2.4 本章练习

1. 查看当前系统中所有的数据库。
2. 创建名为 userinfo 的数据库。
3. 根据表 2-5、表 2-6、表 2-8、表 2-9、表 2-10 的信息，使用 SQL 语句在数据库中创建表。