

第 1 篇

基础知识

本篇通过对数据库基础、初识 MySQL、使用 MySQL 图形化管理工具、数据库操作、存储引擎及数据类型和数据表操作等内容的介绍，并结合大量的图示、举例和视频，帮助读者快速掌握 MySQL，为学习后面的知识奠定坚实的基础。



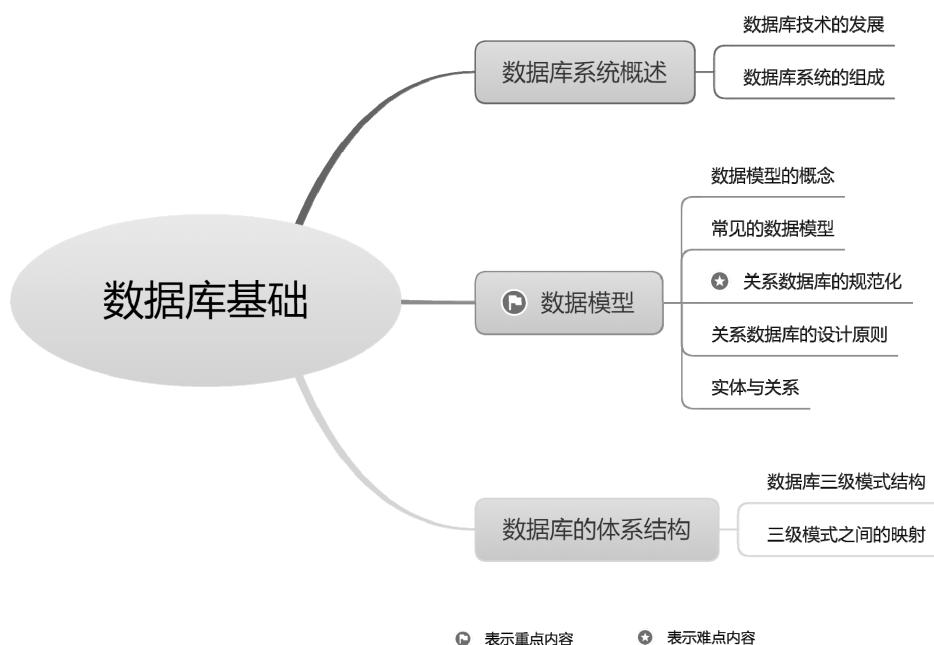
第 1 章



数据库基础

本章主要介绍数据库的相关概念，包括数据库系统概述、数据模型和数据库的体系结构等内容。通过本章的学习，读者应该掌握数据库系统、数据模型、数据库三级模式结构以及数据库规范化等概念。

本章知识架构及重难点如下：



1.1 数据库系统概述



1.1.1 数据库技术的发展

数据库技术是为了响应数据管理任务的需求而产生的，随着计算机技术的发展，对数据管理技术的要求不断提高。数据库技术先后经历了人工管理、文件系统和数据库系统 3 个阶段，下面分别进行介绍。

1. 人工管理阶段

20 世纪 50 年代中期以前, 计算机主要用于科学计算。当时硬件设备和软件技术都很落后, 数据基本依赖于人工管理。人工管理阶段具有如下特点。

- (1) 数据不能被保存。
- (2) 数据没有专门的软件进行管理。
- (3) 不能共享数据。
- (4) 数据不具有独立性。

2. 文件系统阶段

20 世纪 50 年代后期到 20 世纪 60 年代中期, 硬件设备和软件技术都有了进一步发展, 有了磁盘等存储设备和专门的数据管理软件 (即文件系统)。该阶段具有如下特点。

- (1) 数据可以长期被保存。
- (2) 由文件系统管理数据。
- (3) 数据共享性差且冗余大。
- (4) 数据独立性差。

3. 数据库系统阶段

20 世纪 60 年代后期以来, 计算机应用于管理系统, 而且规模越来越大, 应用越来越广泛, 数据量急剧增长, 对共享功能的要求越来越强烈, 使用文件系统管理数据已经不能满足要求, 于是出现了数据库系统来统一管理数据。数据库系统的出现满足了多用户、多应用共享数据的需求, 比文件系统具有明显的优点, 标志着数据管理技术的飞跃。

1.1.2 数据库系统的组成

数据库系统 (database system, DBS) 是采用数据库技术的计算机系统, 是由数据库 (数据)、数据库管理系统、数据库管理员 (database administrator, DBA)、支持数据库系统的硬件和软件 (应用开发工具、应用系统等) 以及用户 5 个部分构成的运行实体, 如图 1.1 所示。其中, 数据库管理员是对数据库进行规划、设计、维护和监视等的专业管理人员, 在数据库系统中起着非常重要的作用。

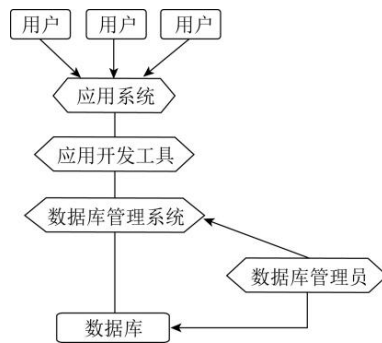


图 1.1 数据库系统的组成

1.2 数据模型



1.2.1 数据模型的概念

数据模型是数据库系统的核心与基础。它是关于描述数据与数据之间的联系、数据的语义、数据

一致性约束的概念性工具的集合。

数据模型通常由数据结构、数据操作和完整性约束 3 部分组成，分别介绍如下。

(1) 数据结构：它是对系统静态特征的描述，它的描述对象包括数据的类型、内容、性质和数据之间的相互关系。

(2) 数据操作：它是对系统动态特征的描述，也是对数据库各种对象实例的操作。

(3) 完整性约束：它是完整性规则的集合，它定义了给定数据模型中数据及其联系所具有的制约和依存规则。

1.2.2 常见的数据模型

常见的数据库数据模型主要有层次模型、网状模型和关系模型。

1. 层次模型

用树状结构表示实体类型及实体间联系的数据模型被称为层次模型，如图 1.2 所示。其特点如下。

(1) 每棵树有且仅有一个无双亲节点，称为根。

(2) 树中除根外所有节点有且仅有一个双亲。

2. 网状模型

用有向图结构表示实体类型及实体间联系的数据模型被称为网状模型，如图 1.3 所示。用网状模型编写应用程序极其复杂，数据的独立性较差。

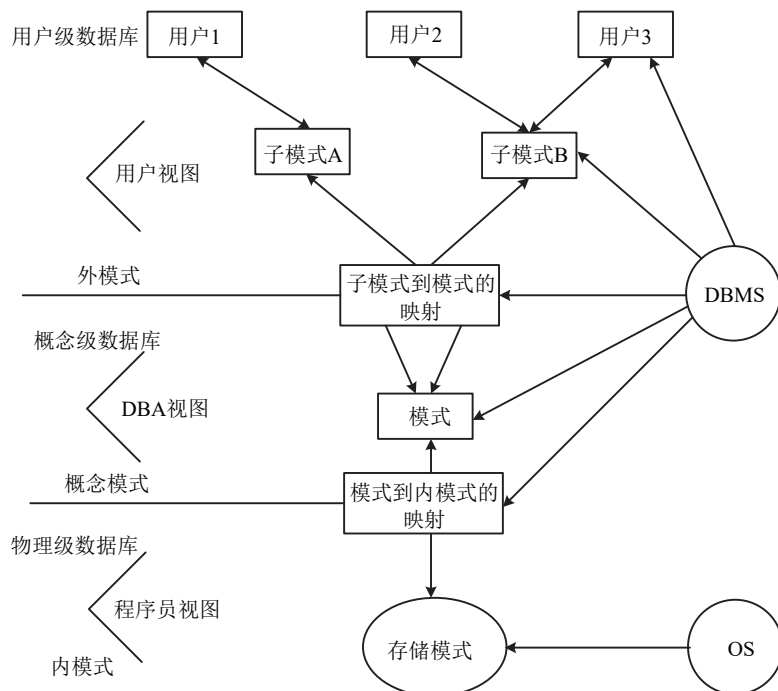


图 1.2 层次模型

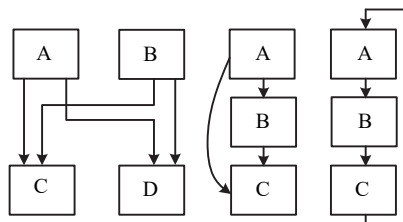


图 1.3 网状模型

3. 关系模型

关系模型以二维表来描述数据。在关系模型中，每个表都有多个字段列和记录行，而每个字段列都有固定的属性（数字、字符、日期等），如图 1.4 所示。关系模型数据结构简单、清晰，具有很高的数据独立性，是目前主流的数据库数据模型。关系模型的基本术语如下。

- （1）关系：一个二维表就是一个关系。
- （2）元组：二维表中的一行，即表中的记录。
- （3）属性：二维表中的一列，用类型和值表示。
- （4）域：每个属性取值的变化范围，如性别的域为{男，女}。

关系中的数据约束如下。

- （1）实体完整性约束：约束关系的主键中属性值不能为空值。
- （2）参照完整性约束：关系之间的基本约束。
- （3）用户定义的完整性约束：反映了具体应用中数据的语义要求。

学生姓名	年级	家庭住址
张三	2000	成都
李四	2000	北京
王五	2000	上海

学生姓名	课程	成绩
张三	数学	100
张三	物理	95
张三	社会	90
李四	数学	85
李四	社会	90
王五	数学	80
王五	物理	75

图 1.4 关系模型

1.2.3 关系数据库的规范化

关系数据库的规范化理论是，关系数据库中的每一个关系都要满足一定的规范。根据满足规范的条件不同，可以分为 5 个等级：第一范式（1NF）、第二范式（2NF）……第五范式（5NF）。其中，NF 是 normal form 的缩写。一般情况下，只要把数据规范到第三范式标准就可以满足需要。下面举例介绍前 3 种范式。

1. 第一范式

第一范式是指在一个关系中，消除重复字段，且各字段都是最小的逻辑存储单位。第一范式是第二范式和第三范式的基础，是最基本的范式。第一范式包括下列指导原则。

- （1）数据组的每个属性只可以包含一个值。
- （2）关系中的每个数组必须包含相同数量的值。
- （3）关系中的每个数组一定不能相同。

在任何一个关系数据库中，第一范式是对关系模式的基本要求，不满足第一范式的数据库就不是关系型数据库。

如果数据表中的每一列都是不可再分割的基本数据项，即同一列中不能有多值，那么就称此数据表符合第一范式，由此可见第一范式具有不可再分解的原子特性。

在第一范式中，数据表的每一行只包含一个实体的信息，并且每一行的每一列只能存放实体的一个属性。例如，对于学生信息，不可以将学生实体的两个或多个或所有属性信息（如学号、姓名、性别、年龄、班级等）都放在一个列中予以显示，也就是说，一个列中应只存放学生实体的一个属性信息。

如果数据表中的列信息都符合第一范式，那么在数据表中的字段都是单一的、不可再分的。如

表 1.1 就是不符合第一范式的学生信息表，因为“班级”列中包含“系别”和“班级”两个属性信息，这样“班级”列中的信息就不是单一的，是可以再分的；而表 1.2 就是符合第一范式的学生信息表，它将原“班级”列的信息拆分到“系别”列和“班级”列中。

表 1.1 不符合第一范式的学生信息表

学 号	姓 名	性 别	年 龄	班 级
9527	东*方	男	20	计算机系 3 班

表 1.2 符合第一范式的学生信息表

学 号	姓 名	性 别	年 龄	系 别	班 级
9527	东*方	男	20	计算机	3 班

2. 第二范式

第二范式是在第一范式的基础上建立起来的，即满足第二范式必先满足第一范式。第二范式要求数据库表中的每个实体（即各个记录行）必须可以被唯一地区分。为实现区分各行记录，通常需要为表设置一个“区分列”，用以存储各个实体的唯一标识。在学生信息表中，设置了“学号”列，由于每个学生的编号都是唯一的，因此每个学生可以被唯一地区分（即使学生存在重名的情况），这个唯一属性列被称为主关键字或主键。

第二范式要求实体的属性完全依赖于主关键字，即不能存在仅依赖于主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该被分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。

例如，这里以员工工资信息表为例，若以员工编码、岗位为组合关键字（即复合主键），就会存在如下决定关系。

（员工编码、岗位）→（决定）（姓名、年龄、学历、基本工资、绩效工资、奖金）

在上面的决定关系中，还可以进一步被拆分为如下两种决定关系。

（员工编码）→（决定）（姓名、年龄、学历）

（岗位）→（决定）（基本工资）

其中，员工编码决定了员工的基本信息（包括姓名、年龄、学历等），而岗位决定了基本工资，因此这个关系表不满足第二范式。

对于上面的这种关系，可以把上述两个关系表更改为如下 3 个表。

员工信息表：EMPLOYEE（员工编码、姓名、年龄和学历）。

岗位工资表：QUARTERS（岗位和基本工资）。

员工工资表：PAY（员工编码、岗位、绩效工资和奖金）。

3. 第三范式

第三范式是在第二范式的基础上建立起来的，即满足第三范式必先满足第二范式。第三范式要求关系表不存在非关键字列对任意候选关键字列的传递函数依赖，也就是说，第三范式要求一个关系表中不包含已在其他表中包含的非主关键字信息。

所谓传递函数依赖，是指如果存在关键字段 A 决定非关键字段 B，而非关键字段 B 决定非关键字段 C，则称非关键字段 C 传递函数依赖于关键字段 A。

例如，这里以员工信息表（EMPLOYEE）为例，该表中包含员工编码、员工姓名、年龄、部门编码、部门经理等信息，该关系表的关键字为“员工编码”，因此存在如下决定关系。

(员工编码) → (决定) (员工姓名、年龄、部门编码、部门经理)

上面的这个关系表是符合第二范式的，但它不符合第三范式，因为该关系表内部隐含着如下决定关系。

(员工编码) → (决定) (部门编码) → (决定) (部门经理)

上面的关系表存在非关键字段“部门经理”对关键字段“员工编码”的传递函数依赖。对于上面的这种关系，可以把这个关系表（EMPLOYEE）更改为如下两个关系表。

员工信息表：EMPLOYEE（员工编码、员工姓名、年龄和部门编码）。

部门信息表：DEPARTMENT（部门编码和部门经理）。

对于关系型数据库的设计，理想的设计目标是按照“规范化”原则存储数据的，因为这样做能够消除数据冗余、更新异常、插入异常和删除异常。

1.2.4 关系数据库的设计原则

数据库设计是指对于一个给定的应用环境，根据用户的需求，利用数据模型和应用程序模拟现实世界中该应用环境的数据结构和处理活动的过程。

数据库设计原则如下。

(1) 数据库内数据文件的数据组织应获得最大限度的共享、最小的冗余度，消除数据及数据依赖关系中的冗余部分，使依赖于同一个数据模型的数据达到有效的分离。

(2) 保证输入、修改数据时数据的一致性与正确性。

(3) 保证数据与使用数据的应用程序之间的高度独立性。

1.2.5 实体与关系

实体是指客观存在并可相互区别的事物。实体既可以是实际的事物，也可以是抽象的概念或关系。实体之间有如下 3 种关系。

(1) 一对一关系：该关系是指表 A 中的一条记录在表 B 中有且只有一条相匹配的记录。在一对一关系中，大部分相关信息都在一个表中。

(2) 一对多关系：该关系是指表 A 中的行可以在表 B 中有许多匹配行，但是表 B 中的行只能在表 A 中有一个匹配行。

(3) 多对多关系：该关系是指关系中每个表的行在相关表中具有多个匹配行。在数据库中，多对多关系的建立是依靠第三个表（称作连接表）实现的，连接表包含相关的两个表的主键列，然后从两个相关表的主键列中分别创建与连接表中匹配列的关系。

1.3 数据库的体系结构



1.3.1 数据库三级模式结构

数据库的三级模式结构是指模式、外模式和内模式。

1. 模式

模式也被称为逻辑模式或概念模式。它是数据库中全体数据的逻辑结构和特征的描述，也是所有用户的公共数据视图。一个数据库只有一个模式。模式处于三级结构的中间层。



注意

定义模式时不仅要定义数据的逻辑结构，而且要定义数据之间的联系，定义与数据有关的安全性、完整性要求。

2. 外模式

外模式也被称为用户模式。它是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，也是数据库用户的数据视图。此外，它还是与某一应用有关的数据的逻辑表示。外模式是模式的子集，一个数据库可以有多个外模式。



说明

定义外模式是保证数据安全性的一个有力措施。

3. 内模式

内模式也被称为存储模式。它是数据物理结构和存储方式的描述，也是数据在数据库内部的表示方式。一个数据库只有一个内模式。

1.3.2 三级模式之间的映射

为了能够在内部实现数据库的 3 个抽象层次的联系和转换，数据库管理系统在三级模式之间提供了两层映射，分别为外模式/模式映射和模式/内模式映射。

1. 外模式/模式映射

同一个模式可以有任意多个外模式。对于每一个外模式，数据库系统都有一个外模式/模式映射。当模式发生改变时，由数据库管理员对各个外模式/模式映射做相应的改变，可以使外模式保持不变。这样，依据数据外模式编写的应用程序就不用修改，保证了数据与程序的逻辑独立性。

2. 模式/内模式映射

数据库中只有一个模式和一个内模式，因此模式/内模式映射是唯一的，它定义了数据库的全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构被改变时，数据库管理员对模式/内模式映射进行相应的改变，以保持模式不变，应用程序也会相应地发生变动。这样就保证了数据与程序的物理独立性。

1.4 小 结

本章主要介绍了数据库技术中的一些基本概念和原理，其中重点包括数据库技术的发展、数据库系统的组成、数据模型的概念、常见的数据模型、关系数据库的规范化及设计原则、实体与关系、数据库的三级模式结构，以及三级模式之间的映射等内容。其中，常见的数据模型和关系数据库的规范化及设计原则希望读者认真学习，重点掌握。

1.5 实践与练习

1. 数据库技术的发展经历了哪 3 个阶段？
2. 数据模型通常是由哪 3 个部分组成的？
3. 常用的数据库数据模型主要有哪几种？

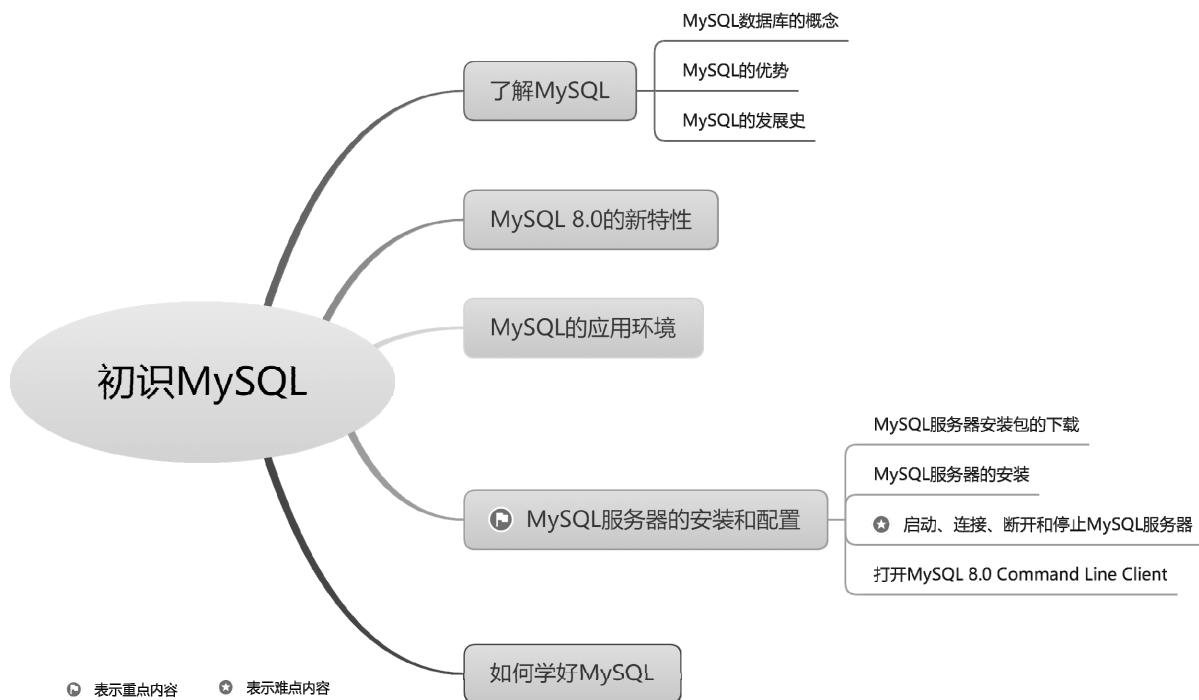
第 2 章



初识 MySQL

MySQL 数据库是目前运行速度最快的 SQL 数据库。除了具有许多其他数据库所不具备的功能和选择，MySQL 数据库还是一种完全免费的产品，用户可以直接从网上下载并安装该软件，无须支付任何费用。另外，MySQL 数据库的跨平台性也是其一大优势。本章将对 MySQL 数据库的概念、特性、应用环境，以及 MySQL 服务器的安装、配置、启动、连接、断开和停止等进行详细介绍。

本章知识架构及重难点如下：



2.1 了解 MySQL



MySQL 是目前最为流行的开放源代码的数据库管理系统之一，是完全网络化、跨平台的关系型数据库系统。它是由瑞典的 MySQL AB 公司开发的，该公司由 MySQL 的初始开发人员 David Axmark 和

Michael “Monty” Widenius 于 1995 年建立, 目前属于 Oracle 公司。MySQL 的象征符号是一只名为 Sakila 的海豚, 代表着 MySQL 数据库和团队的速度、能力、精确和优秀本质。

2.1.1 MySQL 数据库的概念

数据库 (database) 就是一个存储数据的仓库。为了方便数据的存储和管理, 数据库将数据按照特定的规律存储在磁盘上。数据库管理系统可以有效地组织和管理存储在数据库中的数据。MySQL 就是这样的一个关系型数据库管理系统 (RDBMS), 它是目前运行速度最快的 SQL 数据库管理系统。

2.1.2 MySQL 的优势

MySQL 是一款自由软件, 任何人都可以从其官方网站下载。MySQL 是一个真正的多用户、多线程 SQL 数据库服务器。它采用客户端/服务器体系结构, 由一个服务器守护程序 `mysqld` 和很多不同的客户程序及库组成。它能够快捷、有效和安全地处理大量的数据。相对于 Oracle 等数据库来说, MySQL 使用起来非常简单。MySQL 的主要目标是快捷、便捷和易用。

MySQL 被广泛地应用在 Internet 上的中小型网站中。MySQL 由于其体积小、速度快、总体拥有成本低, 尤其是开放源代码, 已成为多数中小型网站为了降低网站总体拥有成本而选择的网站数据库。

2.1.3 MySQL 的发展史

MySQL 的原开发者为瑞典的 MySQL AB 公司。MySQL 是一种完全免费的产品, 用户可以直接从网上下载并安装该软件, 无须支付任何费用。

2008 年 1 月 16 日, Sun 电脑公司 (Sun Microsystems) 正式收购 MySQL AB。

2009 年 4 月 20 日, 甲骨文公司 (Oracle) 宣布以 74 亿美元收购 Sun 电脑公司。

2013 年 6 月 18 日, 甲骨文公司修改 MySQL 授权协议, 移除了 GPL, 将 MySQL 分为社区版和商业版。社区版依然可以免费使用, 但是功能更全的商业版需要付费使用。

MySQL 从无到有, 到技术的不断更新、版本的不断升级, 经历了一个漫长的过程, 这个过程是实践的过程, 也是 MySQL 成长的过程。时至今日, MySQL 的版本已经更新到了 MySQL 8.0。

2.2 MySQL 8.0 的新特性



MySQL 是一个真正的多用户、多线程 SQL 数据库服务器。SQL (结构化查询语言) 是世界上较为流行的和标准化的数据库语言。MySQL 的特性如下。

- (1) 使用 C 和 C++ 语言编写, 并使用了多种编译器进行测试, 保证源代码的可移植性。
- (2) 支持 AIX、FreeBSD、HP-UX、Linux、Mac OS、Novell Netware、OpenBSD、OS/2 Wrap、Solaris、Windows 等多种操作系统。