

项目 3

数据库的创建与操作

数据库是存储数据库对象的容器。通过项目 1 和项目 2 的学习,在安装 MySQL 数据库管理系统、配置好环境,完成需求分析、概念结构设计和数据库逻辑结构的设计后,就需要利用这个环境表达用户的要求,建立数据库及其应用系统。

本项目主要介绍在 MySQL 数据库管理系统的支持下,利用 Navicat 和 SQL 语句创建查看、修改和删除数据库等操作。

项目主要任务:

- 任务 1 创建数据库;
- 任务 2 打开查看数据库;
- 任务 3 修改数据库;
- 任务 4 删除数据库。

能力培养目标:

- 具备创建数据库的能力;
- 具备维护数据库中数据的能力。

素质培养目标:

- 培养学生解决实际问题的独立思考能力;
- 培养学生团队合作的精神;
- 培养学生的创新思维;
- 培养学生数据库设计人员的职业素养。

3.1 项目导入和需求分析

1. 项目导入

通过数据收集、设计 E-R 图、转换关系模式、对关系模式进行规范化处理,得到了学生信息管理系统数据库的逻辑结构,开发人员下一步工作就是在 MySQL 数据库管理系统支持下创建和维护学生信息管理数据库。在数据库创建好以后,还会面临数据库的修改、删除、查看等。本项目着重讲解 MySQL 中创建、查看、修改、删除数据库的方法。

2. 需求分析

后台数据库是对前台页面显示的管理,设计人员在了解 MySQL 数据库管理系统、熟悉其工作环境、掌握 MySQL 数据库的相关概念后,接下来的工作是在 MySQL 数据库管理系统支持下利用 Navicat 和 SQL 语句创建并维护 XSGL 数据库。

3.2 相关知识

3.2.1 认识 SQL 语言

1. SQL 语言概述

结构化查询语言 SQL(Structured Query Language)是由美国国家标准协会 ANSI(American National Standards Institute)和国际标准化组织 ISO(International Standards Organization)定义的标准。

SQL 标准自 1986 年以来不断演化发展,有数种版本。从 1992 年发布的 SQL-92 标准,1999 年发布的 SQL:1999 标准,以及当前最新的 SQL:2008 标准。MySQL 致力于支持套 ANSI/ISO SQL 标准,但不会以牺牲代码的速度和质量为代价。

2. SQL 语言的特点

SQL 语言有以下 4 个特点。

- 一体化:集数据定义语言、数据操纵语言、数据控制语言元素为一体。
- 使用方式:有两种使用方式,即交互使用方式和嵌入高级语言中的使用方式。
- 非过程化语言:只需要提出“干什么”,不需要指出“如何干”,语句的操作过程由系统自动完成。
- 人性化:符合人们的思维方式,容易理解和掌握。

3. SQL 语言的分类

在 MySQL 系统中,根据 SQL 语言的执行功能特点,可以将 SQL 语言分为 3 种类型:数据定义语言、数据操纵语言和数据控制语言。

- 数据定义语言 DDL(Data Definition Language)。数据定义语言是最基础的 SQL 语言类型,其用来创建数据库和创建、修改、删除数据库中的各种对象,为其他语言的操作提供对象。只有在创建数据库和数据库中的各种对象之后,数据库中的各种其他操作才有意义。例如,数据库、表、触发器、存储过程、视图、索引、函数及用户等都是

数据库中的对象,都需要通过定义才能使用。最常用的 DDL 语句是 CREATE、DROP 和 ALTER。

- 数据操纵语言 DML(Date Manipulation Language)。用于完成数据和数据更新操作,其中数据更新是指对数据进行插入、删除和修改操作。最常使用的 DML 语句是 SELECT、INSERT、UPDATE 和 DELETE。
- 数据控制语言 DCL(Date Control Language)。数据控制语言是用来设置或更改数据库用户或角色权限的语句,主要包括 GRANT 语句和 REVOKE 语句。GRANT 语句可以将指定的安全对象的权限授予相应的主体,REVOKE 语句则删除授予的权限。

系统的需求分析反映了用户的需求,我们要将其转化为信息世界的结构,这就是概念结构设计阶段主要要完成的内容。

3.2.2 MySQL 数据库简介

1. MySQL 数据库文件介绍

数据库管理的核心任务包括创建、操作和支持数据库。在 MySQL 中,每个数据库都对应存放在一个与数据库同名的文件夹中。MySQL 数据库文件有“. FRM”“. MYD”和“. MYI” 3 种文件格式,其中“. FRM”是描述表结构的文件,“. MYD”是表的数据文件,“. MYI”是表数据文件中的索引文件。它们都存放在与数据库同名的文件夹中。数据库的默认存放位置 C:\Documents and Settings\AllUsers\ApplicationData\MySQL\MySQL Server 5.5\Data。MySQL 安装完成之后,将会在其 Data 目录下自动创建 information_schema、mysql、performance_schema、sys4 个数据库。

这 4 个数据库的作用如表 3-1 所示。不要随意删除系统自带的数据库,否则会使 MySQL 不能正常运行。

表 3-1 MySQL 自动创建的数据库作用

数据库名称	数据库作用
mysql	描述用户访问权限
Information_schema	保存关于 MySQL 服务器所维护的所有其他数据库的信息,如数据库名,数据库的表、表栏的数据类型与访问权限等
Performance_schema	主要用于收集数据库服务器性能参数
sys	用户利用该数据库进行测试工作

2. MySQL 存储引擎

数据库存储引擎是数据库底层软件组件,数据库管理系统(DBMS)使用数据引擎进行创建、查询、更新和删除数据等操作。不同的存储引擎提供不同的存储机制、索引 技巧、锁定水平等功能,使用不同的存储引擎可以获得特定的功能。现在的数据库管理系统都支持多种不同的数据引擎,MySQL 的核心就是存储引擎。

MySQL 提供了多个不同的存储引擎,包括处理事务安全表的引擎和处理非事务安全表的引擎。在 MySQL 中,不需要在整个服务器中使用同种存储引擎,针对具体的要求,可以对每一个表使用不同的存储引擎。MySQL 支持的存储引擎有 InnoDB、MyISAM、Memory、

Merge、Archive、Federated、CSV、BLACKHOLE 等。可以使用 SHOW ENGINES 语句查看系统所支持的引擎类型。

(1) InnoDB 存储引擎。InnoDB 是事务型数据库的首选引擎,既支持事务安全表 (ACID),也支持行锁定和外键。MySQL 5.5.5 之后,InnoDB 作为默认存储引擎,其主要特性有以下几种。

- InnoDB 给 MySQL 提供了具有提交、回滚和崩溃恢复能力的事务安全 (ACID 兼容) 存储引擎。InnoDB 锁定在行级并在 SELECT 语句中提供一个类似 Oracle 的非锁定读,这些功能增加了多用户部署和性能。在 SQL 查询中,可以自由地将 InnoDB 类型与其他 MySQL 类型的表混合起来,甚至在同一个查询中也可以混合。
- InnoDB 是为处理巨大数据量时的最大性能设计,它的 CPU 效率可能是任何其他基于磁盘的关系数据库引擎所不能匹敌的。
- InnoDB 存储引擎完全与 MySQL 服务器整合,它为在主内存中缓存数据和索引而维持自己的缓冲池。InnoDB 将它的表和索引存在一个逻辑表空间,表空间可以包含数个文件(或原始磁盘分区)。这与 MyISAM 表不同,在 MyISAM 表中,每个表都被保存在分离的文件中。InnoDB 表可以是任意尺寸,即使在文件尺寸被限制为 2GB 的操作系统上。
- InnoDB 支持外键完整性约束 (Foreign Key)。
- 存储表中的数据时,每张表的存储都按主键顺序存放,如果在表定义时没有显示指定主键,InnoDB 会为每一行生成一个 6B 的 ROWID,并以此作为主键。
- InnoDB 被用在众多需要高性能的大型数据库站点上。InnoDB 不创建目录,使用 InnoDB 时,MySQL 将在 MySQL 数据目录下创建一个名称为 ibdata1 的 18MB 的自动扩展数据文件,以及两个名称为 ib_logfile0 和 ib_logfile1 的 5MB 的日志文件。

(2) MyISAM 存储引擎。MyISAM 基于 ISAM 的存储引擎,并对其进行扩展,它是在 Web、数据存储和其他应用环境下最常使用的存储引擎之一。MyISAM 拥有较高的插入、查询速度,但不支持事务。在 MySQL 5.5.5 之前的版本中,MyISAM 是默认存储引擎。MyISAM 的主要特性有以下几种。

- 大文件(达 63 位文件长度)在支持大文件的文件系统和操作系统上被支持。
- 当删除、更新及插入操作混合使用时,动态尺寸的行产生更少碎片,这要通过合并相邻被删除的块,以及若下一个块被删除,就扩展到下一块来自动完成。
- 每个 MyISAM 表最大索引数是 64,这可以通过重新编译来改变。每个索引最大的列数是 16 个。
- 最大的键长度是 1000B,这也可以通过编译来改变。对于键长度超过 250B 的情况,一个超过 1024B 的键将被用上。
- BLOB 和 TEXT 列可以被索引。
- NULL 值被允许在索引的列中。这个值占每个键的一个字节。
- 所有数字键值以高字节优先被存储,以允许一个更高的索引压缩。
- 每表一个 AUTOINCREMENT 列的内部处理。MyISAM 为 INSERT 和 UPDATE 操作自动更新这一列,这使得 AUTO INCREMENT 列更快(至少提高 10%),在序列顶的值被删除之后就不能再利用。

- 可以把数据文件和索引文件放在不同目录中。
- 每个字符列可以有不同的字符集。
- 有 VARCHAR 的表可以固定或动态记录长度。
- VARCHAR 和 CHAR 列可以多达 64KB。

使用 MyISAM 引擎创建数据库,将生成 3 个文件。文件的名称以表的名称开始,扩展名指出文件类型: Frm 文件存储表定义,数据文件的扩展名为. MYD(MYData),索引文件的扩展名为. MYI (MYIndex)。

(3) MEMORY 存储引擎。MEMORY 存储引擎将表中的数据存储在内存中,为查询和引用其他表数据提供快速访问。MEMORY 的主要特性有以下几种。

- MEMORY 表的每个表可以有多达 32 个索引,每个索引 16 列,以及 500B 的最大键长度。
- MEMORY 存储引擎执行 HASH 和 BTRE 索引。
- 可以在一个 MEMORY 表中有非唯一键值。
- MEMORY 表使用一个固定的记录长度格式。
- MEMORY 不支持 BLOB 或 TEXT 列。
- MEMORY 支持 AUTOINCREMENT 列和对可包含 NULL 值的列的索引。
- MEMORY 表在所有客户端之间共享(就像其他任何非 TEMPORARY 表)。
- MEMORY 表内容被保存在内存中,内存是 MEMORY 表和服务器在查询处理的空闲中创建的内部表共享。
- 当不再需要 MEMORY 表的内容时,要释放被 MEMORY 表使用的内存,应该执行 DELETE FROM 或 TRUNCATE TABLE,或者删除整个表(使用 DROP TABLE)。

(4) 存储引擎选择。不同存储引擎都有各自的特点,以适应不同的需求。在实际工作中,选择一个合适的存储引擎是一个比较复杂的问题。每种存储引擎都有自己的优缺点,不能笼统地说谁比谁好。

- InnoDB: 支持事务处理,支持外键,支持崩溃修复能力和并发控制。如果对事务的完整性要求比较高(如银行),要求实现并发控制(如售票),那么选择 InnoDB 有很大的优势。如果需要频繁更新、删除操作的数据库,就可以选择 InnoDB,因为它支持事务的提交(Commit)和回滚(Rollback)。
- MyISAM: 插入数据快,空间和内存使用率比较低。如果表主要用于插入新记录和读出记录,那么选择 MyISAM 能实现高效率处理。如果应用的完整性、并发性要求比较低,也可以使用。
- MEMORY: 所有的数据都在内存中,数据的处理速度虽然快,但是安全性不高。如果需要很快的读写速度,对数据的安全性要求较低,可以选择 MEMORY。但它对表的大小有要求,不能建立太大的表。所以,这类数据库只能使用在相对较小的数据库表中。

注意: 同一个数据库也可以使用多种存储引擎的表。如果一个表对事务处理的要求比较高,可以选择 InnoDB。在这个数据库中可以选择 MyISAM 存储查询要求比较高的表。如果该数据库需要一个用于查询的临时表,那么可以选择 MEMORY 存储引擎。

3.2.3 拓展知识 数据库管理系统：人工管理阶段、文件管理阶段和数据库管理阶段

数据管理系统从 20 世纪 50 年代开始,至今已经历了 7 个阶段,即人工管理阶段、文件管理阶段、数据库管理阶段、主/从结构数据库管理阶段、分布式结构数据库管理阶段、客户机/服务器结构(C/S 结构)数据库管理阶段和 Web 网页/服务器结构(B/S 结构)数据库管理阶段。

在本项目中,我们先介绍人工管理阶段、文件管理阶段和数据库管理阶段,在下一个项目中,我们再介绍数据库发展的主/从结构数据库管理阶段、分布式结构数据库管理阶段、客户机/服务器结构(C/S 结构)数据库管理阶段和 Web 网页/服务器结构(B/S 结构)数据库管理阶段。

1. 人工管理阶段

早期的计算机,由于内存、外存容量很小,软件中也没有数据管理部分,大部分计算机只用于数值计算,没有条件进行大批量的数据处理。程序中要处理的数据,完全靠人工管理,如用穿孔机将数据穿在纸带上输入数据,或用卡片读入机读入存储在卡片上的数据。程序员在进行程序设计和编制时,既要考虑数据的处理方法,又要管理数据的组织存储。

存取数据是根据设备的物理地址进行的,这种方式迫使程序员与物理设备打交道,即要直接管理数据存放地址,导致程序与数据紧密相关,程序高度依赖于数据,数据稍有变动,整个程序就必须全部进行修改,编程效率低,程序不灵活,且容易出错。

数据的人工管理阶段是在 20 世纪 50 年代中期以前,在这一时期,程序和数据是严格的一对一的关系,如图 3-1 所示。

人工管理阶段的特点有以下几点。

- 数据与程序不具独立性,一组数据对应于一份程序,程序依赖于数据,一旦数据发生变化,程序必须作相应的修改。
- 数据不能长期保存,数据随着程序的运行结束而结束。
- 系统中没有对数据进行管理的软件。
- 数据是独占的,不具共享性,即一份程序中的数据不能提供给其他程序使用。

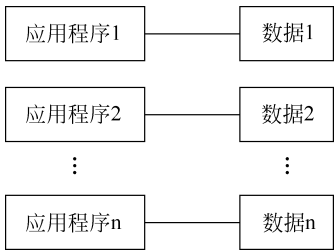


图 3-1 人工管理阶段的数据访问模式

2. 文件管理阶段

文件管理方式,是把有待加工处理的数据组织成数据文件。文件可命名,一旦命名后,程序中便可以通过文件名逻辑性地存取文件中的数据,减轻程序员直接与物理设备打交道的沉重负担,而逻辑存取与物理设备之间要经过转换,这个工作是由文件管理系统完成的,程序中不涉及任何物理设备细节。

20 世纪 50 年代中期,由于磁盘及其他直接存取装置的出现,大大推动了计算机数据处理技术的发展,使计算机应用从数据处理为主逐步转向以非数值处理为主,于是产生了专门管理数据的软件,即文件系统。

文件系统是操作系统的一部分,主要任务是管理存放在计算机外存和内存上的数据,文

件系统是数据管理及处理的主要工具,是数据库技术的基础。

在 20 世纪 50 年代后期至 20 世纪 60 年代中后期,数据管理进入文件管理阶段。在这一阶段中,程序与数据具有一定的独立性,程序与数据可分开进行存储,分别称为程序文件和数据文件,一个数据文件的数据可以被多个程序文件中的程序调用,一个程序文件中的程序也可以调用多个数据文件中的数据。在这一技术中,程序文件是通过“文件系统”与数据文件打交道的,如图 3-2 所示。

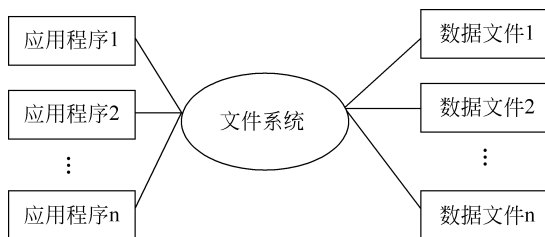


图 3-2 文件管理阶段的数据访问模式

文件管理方式存在的问题。

- 数据冗余度大。
- 数据缺乏独立性,程序与数据仍具有一定的依赖性。
- 数据无集中管理,数据的一致性得不到保证。

3. 数据库管理阶段

针对文件管理阶段的不足,出现了数据库技术。数据库技术的目标,首先是克服程序与文件相互依赖,力求数据独立;其次重在表现数据之间的联系。此外,还要克服数据冗余,解决数据安全性和完整性问题。

数据库技术的特点是:数据与程序相互独立,充分体现数据间的联系,减少数据冗余度。

数据库方式与文件方式的根本区别在于:数据库面向系统,文件方式面向应用。即数据库方式综合全体应用,从整体观点组织数据,形成一个具有一定数据结构的集成化的数据集合,这个集合就是数据库。

从 20 世纪 60 年代后期,进入了数据库管理阶段。数据库(DataBASE)是通用化的相关数据的集合,它按照某种固定的存储格式组织和存储数据。因此,数据库不仅包括数据本身,而且包括关于数据之间的联系。

在数据库管理阶段中,程序文件是通过 DBMS(DataBASE Management System: 数据库管理系统)对数据文件进行访问的,如图 3-3 所示。



图 3-3 数据库管理阶段的数据访问模式

数据库管理阶段的特点如下。

- 实现了数据共享,减少了数据的冗余。
- 采用特定数据模型。
- 具有较高的数据独立性。
- 有统一的数据管理和控制功能。

3.3 任务分解

项目2中设计了学生信息管理系统的逻辑结构。本项目要求在MySQL下实现学生信息管理XSGL数据库,在建库过程中,用户会遇到诸如如何建立数据库,如何查看数据库信息,如何切换各数据库及查看和修改等问题。

3.3.1 任务1 创建数据库

1. 任务目标

- 学会使用创建数据库;
- 学会使用 T-SQL 语句创建数据库。

2. 任务描述

在MySQL中创建XSGL数据库。

3. 课堂任务

- 掌握使用MySQL创建数据库的两种方式;
- 在图形管理工具窗口中通过图形化向导创建;
- 编写SQL语句创建。

3.3.2 任务2 打开查看数据库

1. 任务目标

- 学会切换各个数据库;
- 学会使用 T-SQL 语句打开并查看数据库信息。

2. 任务描述

在MySQL中打开并查看XSGL数据库信息。

3. 课堂任务

- 掌握使用MySQL打开并查看数据库的两种方式;
- 在图形管理工具窗口中通过图形化向导打开并查看信息;
- 编写SQL语句打开并查看。

3.3.3 任务3 修改数据库

1. 任务目标

学会修改数据库中的数据。

2. 任务描述

数据库建好后,必须随时对数据库中的数据进行修改更新,才能满足人们日常工作和生

活的需要。

3. 课堂任务

掌握修改 MySQL 数据库的方法。

3.3.4 任务4 删除数据库

1. 任务目标

学会删除数据库。

2. 任务描述

在 MySQL 中删除 XSQL 数据库。

3. 课堂任务

掌握删除 MySQL 数据库的两种方式。

3.4 任务实施

3.4.1 实施任务1 创建 XSQL 数据库

1. 使用图形管理工具 Navicat 创建数据库

在图形管理工具 Navicat 窗口中使用可视化的界面通过提示来创建数据库,这是最简单也是使用最多的方式,非常适合初学者。创建数据库 XSQL 的具体步骤如下。

(1) 启动 Navicat 工具软件,打开“Navicat for MySQL”窗口,并确保与服务建立连接。

(2) 右击“连接”窗格的 mysql 服务器,在弹出的快捷菜单中选择“新建数据库”菜单命令;或双击展开 mysql 服务器,右击某一个已存在的数据库,在弹出的快捷菜单中选择“新建数据库”菜单命令,如图 3-4 所示。

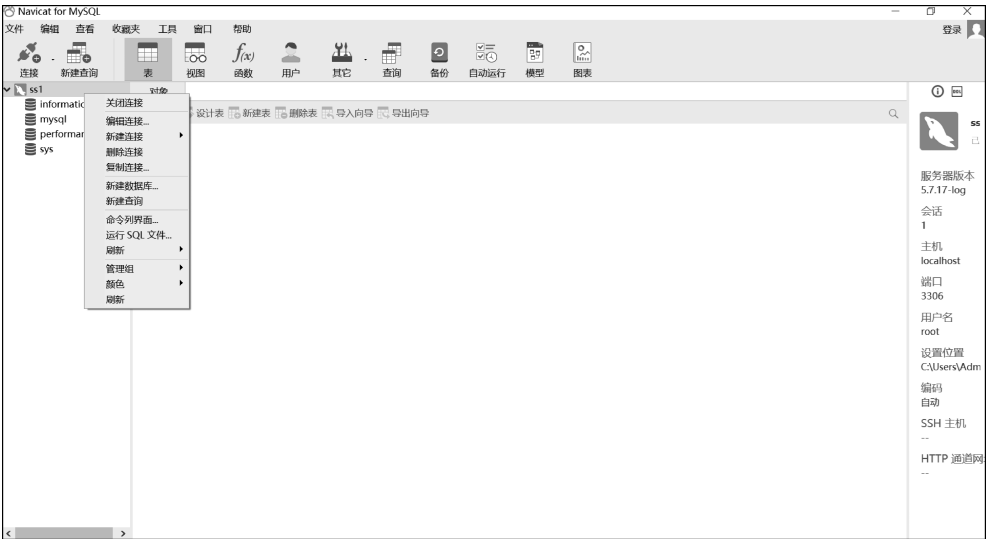


图 3-4 “新建数据库”命令菜单

(3) 在弹出“新建数据库”对话框中的相应文本框中输入数据库名,单击“确定”按钮,就完成了数据库的创建工作,如图 3-5 所示。新建好的数据库如图 3-6 所示。



图 3-5 “新建数据库”对话框

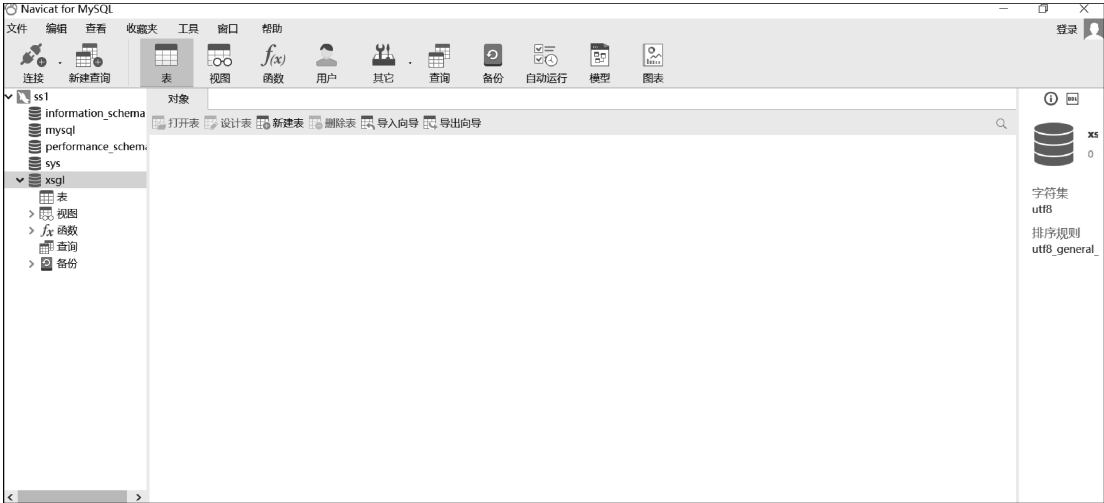


图 3-6 数据库新建完成

2. 使用 SQL 语句创建数据库

虽然使用图形管理工具是创建数据库的一种有效而又容易的方法,但在实际工作中未必总能用它创建数据库。例如,在设计一个应用程序时,开发人员会直接使用 SQL 语句在程序代码中创建数据库及其他数据库对象。MySQL 使用 CREATE DATABASE 语句同样可以完成数据库的创建操作。

用 CREATE DATABASE 命令创建数据库的语法格式如下。

```
create database database_name;
```

在该语法中 database_name 为要创建数据库的名称,该名称不能与已存在的数据库重名。

【例 3-1】 用 CREATE DATABASE 命令创建一个数据库 XSGL。

启动 Navicat 工具软件,打开 Narvicat for MySQL 窗口,并确保与服务建立连接。