

本章要点

- (1) MySQL 数据库简介。
- (2) 数据定义。
- (3) 存储引擎。

数据库是一个存储数据对象的容器,数据对象包含表、视图、索引、存储过程、触发器等,我们必须首先创建数据库,然后才能创建存放于数据库中的数据对象。本章介绍 MySQL 数据库概述,MySQL 数据库的创建、选择、修改和删除,存储引擎等内容。

本书很多例题都是基于学生信息数据库 `stusys`,`stusys` 是本书的重要数据库,参见附录 B 学生信息数据库 `stusys` 的表结构和样本数据。

3.1 MySQL 数据库简介

安装 MySQL 数据库时,生成了系统使用的数据库,包括 `mysql`、`information_schema`、`performance_schema` 和 `sys` 等,MySQL 把有关数据库管理系统自身的管理信息都保存在这几个数据库中,如果删除了它们,MySQL 将不能正常工作,操作时要十分小心。

可以使用 `SHOW DATABASES` 命令查看已有的数据库。

【例 3.1】 查看 MySQL 服务器中的已有数据库。

在 MySQL 命令行客户端输入如下语句:

```
mysql > SHOW DATABASES;
```

执行结果:

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| sys               |
+-----+
4 rows in set (0.00 sec)
```

这几个系统使用的数据库如果被删除了,MySQL 将无法正常工作,操作时务必注意,其作用分别介绍如下。

(1) `information_schem`: 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。

- (2) mysql: 描述用户访问权限。
- (3) performance_schema: 主要用于收集数据库服务器性能参数。
- (4) sys: 该数据库里面包含了一系列的存储过程、自定义函数以及视图,存储了许多系统的元数据信息。

3.2 定义数据库

数据定义语言用于定义数据库和定义表、视图等,定义数据库包括创建数据库、选择数据库、修改数据库和删除数据库等操作,下面分别介绍。

3.2.1 创建数据库

在使用数据库以前,首先需要创建数据库。在学生成绩管理系统中,以创建名称为 stusys 的学生信息数据库为例,说明创建数据库使用的 SQL 语句。

创建数据库使用 CREATE DATABASE 语句。

语法格式:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS]db_name
[[DEFAULT] CHARACTER SET charset_name]
[[DEFAULT] COLLATE collation_name];
```

说明:

(1) 语句中“[]”为可选语法项,“{ }”为必选语法项,“|”分隔括号或大括号中的语法项,只能选择其中一项。

(2) db_name: 数据库名称。

(3) IF NOT EXISTS: 在创建数据库前进行判断,只有该数据库目前尚不存在时才执行 CREATE DATABASE 操作。

(4) CHARACTER SET: 指定数据库字符集。

(5) COLLATE: 指定字符集的校对规则。

(6) DEFAULT: 指定默认值。

【例 3.2】 创建名称为 stusys 的学生信息数据库,该数据库是本书的重要数据库。

在 MySQL 命令行客户端输入如下 SQL 语句:

```
mysql > CREATE DATABASE stusys;
```

执行结果:

```
Query OK, 1 row affected (0.06 sec)
```

查看已有数据库的语句如下:

```
mysql > SHOW DATABASES;
```

显示结果:

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| stusys |
| sys |
+-----+
5 rows in set (0.36 sec)
```

可以看出,数据库列表中包含了刚创建的数据库 stusys。

3.2.2 选择数据库

用 CREATE DATABASE 语句创建了数据库之后,该数据库不会自动成为当前数据库,需要用 USE 语句指定其为当前数据库。

语法格式:

```
USE db_name;
```

【例 3.3】 选择 stusys 为当前数据库。

```
mysql> USE stusys;
```

执行结果:

```
Database changed
```

3.2.3 修改数据库

数据库创建后,如果需要修改数据库的参数,可以使用 ALTER DATABASE 语句。

语法格式:

```
ALTER{DATABASE | SCHEMA} [db_name]
[DEFAULT] CHARACTER SET charset_name
[DEFAULT] COLLATE collation_name;
```

说明:

- (1) 数据库名称可以省略,表示修改当前(默认)数据库。
- (2) 选项 CHARACTER SET 和 COLLATE 与创建数据库语句相同。

【例 3.4】 修改数据库 stusys 的默认字符集和校对规则。

```
mysql> ALTER DATABASE stusys
- < DEFAULT CHARACTER SET gb2312
- < DEFAULT COLLATE gb2312_chinese_ci;
```

执行结果:

```
Query OK, 1 row affected (0.31 sec)
```

3.2.4 删除数据库

删除数据库使用 DROP DATABASE 语句。

语法格式：

```
DROP{DATABASE | SCHEMA} [ IF EXISTS] db_name
```

说明：

(1) db_name：指定要删除的数据库名称。

(2) DROP DATABASE 或 DROP SCHEMA：该命令会删除指定的整个数据库，数据库中所有的表和所有数据也将被永久删除，并不给出任何提示需要确认的信息。因此，删除数据库要特别小心。

(3) IF EXISTS：使用该子句，可避免删除不存在的数据库时出现 MySQL 错误信息。

【例 3.5】 删除数据库 stusys。

```
mysql > DROP DATABASE stusys;
```

执行结果：

```
Query OK, 0 rows affected (0.23 sec)
```

查看现有数据库：

```
mysql > SHOW DATABASES;
```

显示结果：

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| sys               |
+-----+
4 rows in set (0.00 sec)
```

可以看到，由于数据库 stusys 被删除，数据库列表中已没有名称为 stusys 的数据库了。

3.3 存储引擎

存储引擎决定了表在计算机中的存储方式。存储引擎就是如何存储数据、如何为存储的数据建立索引和如何更新、查询数据等技术的实现方法。因为在关系数据库中数据的存储是以表的形式存储的，所以存储引擎简而言之就是指表的类型。

3.3.1 存储引擎概述

在 Oracle 和 SQL Server 等数据库管理系统中只有一种存储引擎，所有数据存储管理

机制都是一样的。而 MySQL 提供了多种存储引擎,用户可以根据不同的需求为表选择不同的存储引擎,也可以根据自己的需要编写自己的存储引擎,MySQL 的核心就是存储引擎。

我们可以通过 SHOW ENGINES 命令查看存储引擎。

【例 3.6】 查看存储引擎。

```
mysql > SHOW ENGINES;
```

显示结果:

```
+-----+-----+-----+-----+-----+-----+
| Engine          | Support | Comment                                     | Transactions | XA | Savepoints |
+-----+-----+-----+-----+-----+-----+
| MEMORY         | YES     | Hash based, stored in memory,             | NO           | NO | NO         |
|                |         | useful for temporary tables              |              |   |           |
| MRG_MYISAM     | YES     | Collection of identical MyISAM           | NO           | NO | NO         |
|                |         | tables                                    |              |   |           |
| CSV            | YES     | CSV storage engine                       | NO           | NO | NO         |
| FEDERATED      | NO      | Federated MySQL storage engine           | NULL         | NULL | NULL       |
| PERFORMANCE_SCHEMA | YES    | Performance Schema                       | NO           | NO | NO         |
| MyISAM         | YES     | MyISAM storage engine                    | NO           | NO | NO         |
| InnoDB         | DEFAULT | Supports transactions, row-level        | YES          | YES | YES        |
|                |         | locking, and foreign keys                |              |   |           |
| BLACKHOLE      | YES     | /dev/null storage engine (anything you  | NO           | NO | NO         |
|                |         | write to it disappears)                  |              |   |           |
| ARCHIVE        | YES     | Archive storage engine                   | NO           | NO | NO         |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.28 sec)
```

由显示结果可看出,MySQL 8.0 支持的存储引擎有: MEMORY、MRG_MYISAM、CSV、FEDERATED、PERFORMANCE_SCHEMA、MyISAM、InnoDB、BLACKHOLE、ARCHIVE 等九种,默认的存储引擎是 InnoDB。

3.3.2 常用存储引擎

下面介绍几种常用的存储引擎。

1. 存储引擎 InnoDB

InnoDB 是 MySQL 8.0 的默认存储引擎,它给 MySQL 表提供了事务、回滚、崩溃恢复能力和并发控制的事务安全。

MySQL 5.6 以后,除系统数据库外,默认存储引擎由 MyISAM 改为 InnoDB,而 MySQL 8.0 在原先的基础上进一步将系统数据库存储引擎也改为 InnoDB。

InnoDB 支持外键约束、支持自动增长列 AUTO_INCREMENT。

InnoDB 存储引擎的优势是提供了良好的事务管理,缺点是读写效率稍差,占用数据空间较大。

2. 存储引擎 MyISAM

MyISAM 存储引擎是 MySQL 中常见的存储引擎,曾经是 MySQL 的默认存储引擎。

MyISAM 存储引擎的表存储成三个文件。文件的名称与表名相同。扩展名包括 frm、myd 和 myi。其中,以 frm 为扩展名的文件,存储表的结构;以 myd 为扩展名的文件,存储数据;以 myi 为扩展名的文件,存储索引。

MyISAM 存储引擎的优势是占用空间小,处理速度快;缺点是不支持事务的完整性和并发性。

3. 存储引擎 MEMORY

MEMORY 存储引擎是 MySQL 中的一类特殊的存储引擎。它使用存储在内存中的内容来创建表,而且所有数据也放在内存中。这些特性都与 InnoDB 存储引擎、MyISAM 存储引擎不同。

每个基于 MEMORY 存储引擎的表实际对应一个磁盘文件,该文件的文件名与表名相同,类型为 frm 类型。该文件中存储表的结构,而且数据文件是存储在内存中。这样有利于对数据的快速处理,提高整个表的处理效率。

MEMORY 表由于其存在于内存的特性,其处理速度很快,但其数据易丢失,生命周期短。

3.3.3 选择存储引擎

在实际工作中,选择一个合适的存储引擎是一个很复杂的问题。每种存储引擎都有各自的优势,可以根据各种存储引擎的特点进行对比,给出不同情况下选择存储引擎的建议。

下面,对存储引擎 InnoDB、MyISAM 和 MEMORY 的事务安全、存储显示、空间使用、内存使用、对外键的支持、插入数据速度、批量插入速度、锁机制、数据可压缩等进行比较,如表 3.1 所示。

表 3.1 存储引擎比较

特 性	InnoDB	MyISAM	MEMORY
事务安全	支持	无	无
存储显示	64TB	有	有
空间使用	高	低	低
内存使用	高	低	高
对外键的支持	支持	无	无
插入数据速度	低	高	高
批量插入速度	低	高	高
锁机制	行锁	表锁	表锁
数据可压缩	无	支持	无

依据表 3.1 对存储引擎 InnoDB、MyISAM 和 MEMORY 特性的对比,选择时建议如下。

1. InnoDB 存储引擎

InnoDB 存储引擎支持事务处理,支持外键,支持崩溃恢复能力和并发控制,如果对事务完整性和并发控制要求比较高,选择 InnoDB 存储引擎具有优势。对于需要频繁进行更新、删除操作的数据库,也可选择 InnoDB 存储引擎。

2. MyISAM 存储引擎

MyISAM 存储引擎处理数据快,空间和内存使用低。如果表主要是用于插入记录和读出记录,选择 MyISAM 存储引擎处理效率高。对于应用的完整性、并发性要求低,也可选择 MyISAM 存储引擎。

3. MEMORY 存储引擎

MEMORY 存储引擎的数据都在内存中,数据处理速度快,但安全性不高。如果要求很快的读写速度,对数据安全性要求低,可以选择 MEMORY 存储引擎。由于 MEMORY 存储引擎对表的大小有要求,不能建较大的表,所以使用于较小的数据库表中。

MySQL 中提到的存储引擎的概念,它是 MySQL 的一个特性,可简单理解为后面要介绍的表类型。每一个表都有一个存储引擎,可在创建时指定,也可以使用 ALTER TABLE 语句修改,通过 ENGINE 关键字设置。

3.4 小 结

本章主要介绍了以下内容。

(1) 数据库是一个存储数据对象的容器,数据对象包含表、视图、索引、存储过程、触发器等。

安装 MySQL 数据库时,生成了系统使用的数据库,包括 mysql、information_schema、performance_schema 和 sys 等。

可以使用 SHOW DATABASES 命令查看已有的数据库。

(2) 在定义数据库中的语句使用如下。

创建数据库使用 CREATE DATABASE 语句。

选择数据库使用 USE 语句。

修改数据库使用 ALTER DATABASE 语句。

删除数据库使用 DROP DATABASE 语句。

(3) 存储引擎决定了表在计算机中的存储方式。存储引擎就是如何存储数据、如何为存储的数据建立索引和如何更新、查询数据等技术的实现方法。因为在关系数据库中数据的存储是以表的形式存储的,所以存储引擎简而言之就是指表的类型。

在 Oracle 和 SQL Server 等数据库管理系统中只有一种存储引擎,所有数据存储管理机制都是一样的。而 MySQL 提供了多种存储引擎,用户可以根据不同的需求为表选择不同的存储引擎,也可以根据自己的需要编写自己的存储引擎,MySQL 的核心就是存储引擎。

可以通过 SHOW ENGINES 命令查看存储引擎。

(4) MySQL 常用的存储引擎有 InnoDB、MyISAM 和 MEMORY。

① InnoDB 存储引擎。InnoDB 是 MySQL 8.0 的默认存储引擎,InnoDB 存储引擎支持事务处理,支持外键,支持崩溃恢复能力和并发控制,如果对事务完整性和并发控制要求比较高,选择 InnoDB 存储引擎具有优势。对于需要频繁进行更新、删除操作的数据库,也可选择 InnoDB 存储引擎。

② MyISAM 存储引擎。MyISAM 存储引擎处理数据快,空间和内存使用低。如果表

三、问答题

- 3.14 为什么需要系统数据库？用户可否删除系统数据库？
- 3.15 在定义数据库中,包括哪些语句？
- 3.16 什么是存储引擎？MySQL 的存储引擎与 Oracle、SQL Server 的存储引擎有何不同？
- 3.17 简述存储引擎 InnoDB、MyISAM 和 MEMORY 的特点。
- 3.18 试对比分析常用的存储引擎 InnoDB、MyISAM 和 MEMORY。