

第一章 初识MySQL

章节内容

- 数据库基础知识 熟悉
- 安装 MySQL 数据库 熟悉
- SQL 分类 重点
- 数据库的创建、修改和删除 重点
- 列类型 重点
- 数据表的创建、修改和删除 重点

章节目标

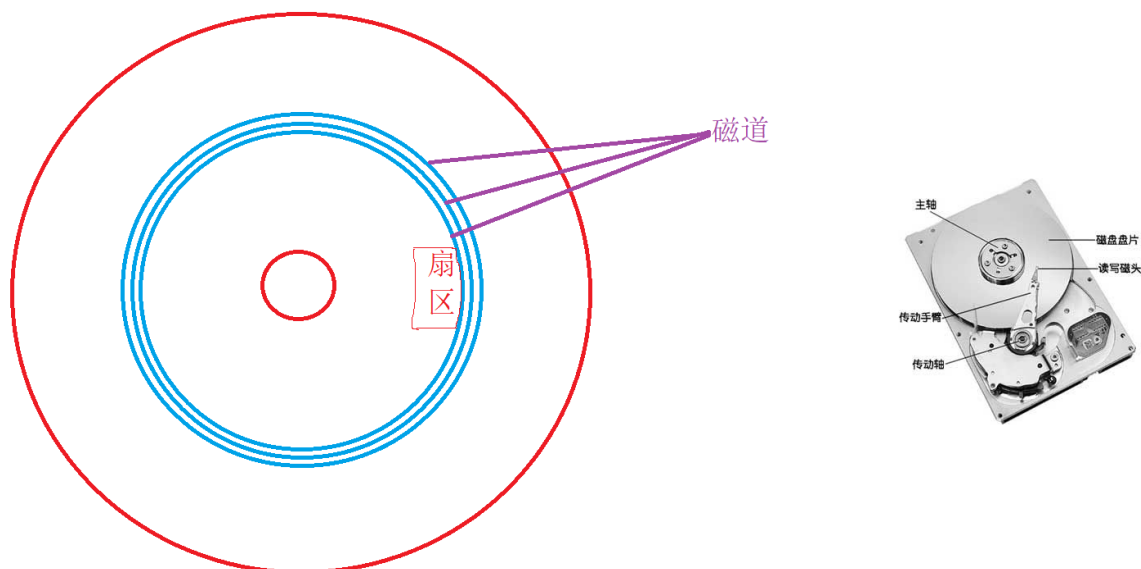
- 熟悉 MySQL 数据库基础知识
- 熟练安装 MySQL 数据库
- 掌握数据库的创建、修改和删除
- 掌握常用的列类型
- 掌握数据表的创建、修改和删除

第一节 数据库基础知识

1. 数据库

1.1 概念

数据库，英文名称为Database，简称DB。从字面意思来看就是存储数据的仓库；从专业角度解释为存储在计算机磁盘上的有组织、可共享的大量数据的集合。



1.2 数据库的类型

数据库分为关系型数据库和非关系型数据库两大类。常见的关系型数据库有 MySQL、Oracle、SQL Server、SQLite、DB2 等。常见的非关系型数据库有 Redis、MongoDB 等。

1.3 数据库管理系统

数据库管理系统，英文名称为Database Management System，简称DBMS。主要用于科学组织和存储数据、高效的获取和维护数据。

Navicat

2. MySQL 介绍和安装

2.1 MySQL 介绍

MySQL 是目前最流行的开源的、免费的关系型数据库，适用于中小型甚至大型互联网应用，能够在 windows 和 linux 平台上部署

MySQL Oracle 属于Oracle公司

2.2 MySQL 安装

建议初始密码设置为root

演示

2.3 连接数据库

找到 MySQL 安装目录下的 bin 目录，然后打开命令窗口，在命令窗口中按如下语法输入命令：

```
mysql -h MySQL数据库服务器的IP地址 -u 用户名 -p
```

然后按下回车键，输入密码即可

第二节 结构化查询语言

1. SQL 分类

结构化查询语句，英文名称为Structured Query Language，简称 SQL。结构化查询语句分为数据定义语言、数据操作语言、数据查询语言和数据控制语言四大类。

名称	描述	命令
数据定义语言 (DDL)	数据库、数据表的创建、修改和删除	CREATE、ALTER、DROP
数据操作语言 (DML)	数据的增加、修改和删除	INSERT、UPDATE、DELETE
数据查询语言 (DQL)	数据的查询	SELECT
数据控制语言 (DCL)	用户授权、事务的提交和回滚	GRANT、COMMIT、ROLLBACK

2. 数据库操作

2.1 创建数据库的语法

```
CREATE DATABASE [IF NOT EXISTS] 数据库名称 DEFAULT CHARACTER SET 字符集 COLLATE 排序规则;
```

示例：创建数据库lesson，并指定字符集为 GBK，排序规则为 GBK_CHINESE_CI

```
CREATE DATABASE IF NOT EXISTS lesson DEFAULT CHARACTER SET GBK COLLATE GBK_CHINESE_CI;
```

2.2 修改数据库的语法

```
ALTER DATABASE 数据库名称 CHARACTER SET 字符集 COLLATE 排序规则;
```

示例：修改数据库lesson的字符集为 UTF8，排序规则为 UTF8_GENERAL_CI

```
ALTER DATABASE lesson CHARACTER SET UTF8 COLLATE UTF8_GENERAL_CI;
```

2.3 删除数据库的语法

```
DROP DATABASE [IF EXISTS] 数据库名称;
```

示例：删除数据库lesson

```
DROP DATABASE IF EXISTS lesson;
```

2.4 查看数据库语法

```
SHOW DATABASES;
```

2.5 使用数据库的语法

```
USE 数据库名称;
```

示例：使用数据库lesson

```
USE lesson;
```

3. 列类型

在 MySQL 中，常用列类型主要分为数值类型、日期时间类型、字符串类型。

3.1 数值类型

类型	说明	取值范围	存储需求
<code>tinyint</code>	非常小的数据	有符号值: $-2^7 \sim 2^7-1$ 无符号值: $0 \sim 2^8-1$	1字节
<code>smallint</code>	较小的数据	有符号值: $-2^{15} \sim 2^{15}-1$ 无符号值: $0 \sim 2^{16}-1$	2字节
<code>mediumint</code>	中等大小的数据	有符号值: $-2^{23} \sim 2^{23}-1$ 无符号值: $0 \sim 2^{24}-1$	3字节
<code>int</code>	标准整数	有符号值: $-2^{31} \sim 2^{31}-1$ 无符号值: $0 \sim 2^{32}-1$	4字节
<code>bigint</code>	较大的整数	有符号值: $-2^{63} \sim 2^{63}-1$ 无符号值: $0 \sim 2^{64}-1$	8字节
<code>float</code>	单精度浮点数	无符号值: $1.1754351 * 10^{-38} \sim 3.402823466 * 10^{38}$	4字节
<code>double</code>	双精度浮点数	无符号值: $2.22507385 * 10^{-308} \sim 1.79769313 * 10^{308}$	8字节
<code>decimal</code>	字符串形式的浮点数	<code>decimal(m, d)</code>	m个字节

3.2 日期时间类型

类型	说明	取值范围
<code>DATE</code>	<code>YYYY-MM-dd</code> , 日期格式	<code>1000-01-01</code> ~ <code>9999-12-31</code>
<code>TIME</code>	<code>HH:mm:ss</code> , 时间格式	<code>-838:59:59.000000</code> ~ <code>838:59:59.000000</code>
<code>DATETIME</code>	<code>YY-MM-dd HH:mm:ss</code>	<code>1000-01-01 00:00:00.000000</code> ~ <code>9999-12-31 23:59:59.999999</code>
<code>TIMESTAMP</code>	<code>YYYY-MM-dd HH:mm:ss</code> 格式表示的时间戳	<code>1970-01-01 00:00:01.000000</code> ~ <code>2038-01-19 03:14:07.999999</code>
<code>YEAR</code>	<code>YYYY</code> 格式的年份值	1901~2155

3.3 字符串类型

类型	说明	最大长度
<code>char [(M)]</code>	固定长字符串, 检索快但费空间, $0 \leq M \leq 255$	M字符
<code>varchar [(M)]</code>	可变字符串 $0 \leq M \leq 65535$	变长度
<code>text</code>	文本串	$2^{16}-1$ 字节

3.4 列类型修饰属性

属性名	说明	示例
UNSIGNED	无符号，只能用来修饰数值类型，表明该列数据不能出现负数	INT(4) UNSIGNED，表示只能为4位大于等于0的整数
ZEROFILL	不足的位数使用0来填充	INT(4) ZEROFILL，如果给定的值为10，此时只有2位，而该列需要4位，不足的2位由0来填充，最终值为0010
NOT NULL	表示该列类型的值不能为空	VARCHAR(20) NOT NULL，表示该列数据不能为空值
DEFAULT	表示设置默认值	INT(4) DEFAULT 0，表示该列不赋值时默认为0
AUTO_INCREMENT	表示自增长，只能应用于数值列类型，该列类型必须为键，且不能为空	INT(11) AUTO_INCREMENT NOT NULL PRIMARY KEY。第一次为该列中插入值时为1，第二次为2

4. 数据表操作

4.1 数据表类型

MySQL 中的数据表类型有许多，如 MyISAM、InnoDB、HEAP、BLOB、CSV 等。其中最常用的就是 MyISAM 和 InnoDB

4.2 MyISAM 与 InnoDB 的区别

名称	MyISAM	InnoDB
事务处理	不支持	支持
数据行锁定	不支持	支持
外键约束	不支持	支持
全文索引	支持	不支持
表空间大小	较小	较大,约2倍

事务：涉及的所有操作是一个整体，要么都执行，要么都不执行。

数据行锁定：一行数据，当一个用户在修改该数据时，可以直接将该条数据锁定。

如何选择数据表的类型？

当涉及的业务操作以查询居多，修改和删除较少时，可以使用 MyISAM。当涉及的业务操作经常会有修改和删除操作时，使用 InnoDB。

4.3 创建数据表

```
CREATE TABLE [IF NOT EXISTS] 数据表名称(
    字段名1 列类型(长度) [修饰属性] [键/索引] [注释],
    字段名2 列类型(长度) [修饰属性] [键/索引] [注释],
    字段名3 列类型(长度) [修饰属性] [键/索引] [注释],
    .....
    字段名n 列类型(长度) [修饰属性] [键/索引] [注释]
) [ENGINE = 数据表类型] [CHARSET=字符集编码] [COMMENT=注释];
```

示例：创建学生表，表中有字段学号、姓名、性别、年龄和成绩

```
CREATE TABLE IF NOT EXISTS student(
    `number` VARCHAR(30) NOT NULL PRIMARY KEY COMMENT '学号，主键',
    name VARCHAR(30) NOT NULL COMMENT '姓名',
    sex TINYINT(1) UNSIGNED DEFAULT 0 COMMENT '性别：0-男 1-女 2-其他',
    age TINYINT(3) UNSIGNED DEFAULT 0 COMMENT '年龄',
    score DOUBLE(5, 2) UNSIGNED COMMENT '成绩'
) ENGINE=InnoDB CHARSET=utf8 COMMENT='学生表';
```

4.4 修改数据表

- 修改表名

```
ALTER TABLE 表名 RENAME AS 新表名;
```

示例：将student表名称修改为 stu

```
ALTER TABLE student RENAME AS stu;
```

- 增加字段

```
ALTER TABLE 表名 ADD 字段名 列类型(长度) [修饰属性] [键/索引] [注释];
```

示例：在 stu 表中添加字段联系电话(phone)，类型为字符串，长度为11，非空

```
ALTER TABLE stu ADD phone VARCHAR(11) NOT NULL COMMENT '联系电话';
```

- 查看表结构

```
DESC 表名; -- 查看表结构
```

- 修改字段

```
-- MODIFY 只能修改字段的修饰属性
ALTER TABLE 表名 MODIFY 字段名 列类型(长度) [修饰属性] [键/索引] [注释];
-- CHANGE 可以修改字段的名字以及修饰属性
ALTER TABLE 表名 CHANGE 字段名 新字段名 列类型(长度) [修饰属性] [键/索引] [注释];
```

示例：将 `stu` 表中的 `sex` 字段的类型设置为 `VARCHAR`，长度为2，默认值为'男'，注释为 "性别，男，女，其他"

```
ALTER TABLE stu MODIFY sex VARCHAR(2) DEFAULT '男' COMMENT '性别：男，女，其他';
```

示例：将 `stu` 表中 `phone` 字段修改为 `mobile`，属性保持不变

```
ALTER TABLE stu CHANGE phone mobile VARCHAR(11) NOT NULL COMMENT '联系电话';
```

- **删除字段**

```
ALTER TABLE 表名 DROP 字段名;
```

示例：将 `stu` 表中的 `mobile` 字段删除

```
ALTER TABLE stu DROP mobile;
```

4.5 删除数据表

```
DROP TABLE [IF EXISTS] 表名;
```

示例：删除数据表 `stu`

```
DROP TABLE IF EXISTS stu;
```