MySQL 数据库入门

"Zero to Hero !"

井明
数据科学与计算机学院
jingming@sdu.edu.cn
2025年3月14日

山东女子学院
Shandong Women's University

MySQL 安装

官方文档：

- MySQL 安装包下载地址： MySQL Community Downloads
- MySQL 安装教程： MySQL 安装教程

课程安装包：

- MySQL 安装包: 百度云盘 (提取码: sdwu)

MySQL 安装步骤

1. 解压安装包到指定目录，如：`D:\mysql-8.0.41-winx64`
2. 运行 `cmd`，进入解压目录下的 `bin` 目录
3. 执行命令：`mysqld --initialize-insecure --console` 或 `mysqld -I` ，初始化数据库。
4. 执行命令：`mysqld --install`，安装 MySQL 服务。如果安装失败，可以尝试使用管理员权限运行 `cmd`，再次执行命令。
5. 执行命令：`net start mysql`，启动 MySQL 服务
6. 执行命令：`mysql -u root -p`，登录 MySQL 数据库
7. 输入密码：`root`，登录成功。如果密码不正确，可以尝试使用 `mysql -u root -p --skip-password` 登录。

MySQL 安装步骤（课程方案）

1. 解压安装包到指定目录，如：`D:\mysql-8.0.41-winx64`
2. 运行 `cmd`，进入解压目录下的 `bin` 目录
3. 执行命令：`mysqld --initialize-insecure --console` 或 `mysqld -I` ，初始化数据库。
4. 在解压目录下打开`data`文件夹，打到.err文件，查看随机生成的密码。
5. 再次执行命令：`mysqld`，启动MySQL数据库。
6. 执行命令：`mysql -u root -p`，登录 MySQL 数据库
7. 输入密码：查看.err文件中的密码，登录成功。

navicat17 premium lite 安装 (2选1)

- 安装包下载地址： navicat17 premium lite
- 百度网盘 链接: https://pan.baidu.com/s/1m25PETmtzQCj-uS7LnPTGg?pwd=sdwu 提取码: sdwu

MySQL常用命令

以下是 20 条常用的 MySQL 命令及其示例，涵盖数据库管理、表操作、数据操作和用户管理等方面：

1. 连接 MySQL

```
mysql -u root -p
```

说明：使用 root 用户连接 MySQL，系统会提示输入密码。

2. 显示所有数据库

```
SHOW DATABASES;
```

说明：列出 MySQL 服务器上的所有数据库。

3. 创建数据库

```
CREATE DATABASE mydb;
```

说明：创建名为 mydb 的数据库。

4. 使用数据库

```
USE mydb;
```

说明：切换到 mydb 数据库进行操作。

5. 删除数据库

```
DROP DATABASE mydb;
```

说明：删除 mydb 数据库，操作不可撤销。

6. 显示所有表

```
SHOW TABLES;
```

说明：列出当前数据库中的所有表。

7. 创建表

```
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    age INT,
    email VARCHAR(100)
);
```

说明：创建 users 表，包含 id（主键）、name、age 和 email 字段。

8. 查看表结构

```
DESC users;
```

说明：显示 users 表的结构，包括字段、类型、主键等信息。

9. 删除表

```
DROP TABLE users;
```

说明：删除 users 表，操作不可撤销。

10. 插入数据

```
INSERT INTO users (name, age, email) VALUES ('Alice', 25, 'alice@example.com');
```

说明：向 users 表中插入一条数据。

11. 查询数据

```
SELECT * FROM users;
```

说明：查询 users 表中的所有数据。

12. 条件查询

```
SELECT * FROM users WHERE age > 20;
```

说明：查询 users 表中 age 大于 20 的所有用户。

13. 更新数据

```
UPDATE users SET age = 30 WHERE name = 'Alice';
```

说明：将 Alice 的 age 修改为 30。

14. 删除数据

```
DELETE FROM users WHERE name = 'Alice';
```

说明：删除 users 表中 name 为 Alice 的用户。

15. 计数查询

```
SELECT COUNT(*) FROM users;
```

说明：统计 users 表中的数据条数。

16. 排序查询

```
SELECT * FROM users ORDER BY age DESC;
```

说明：按 age 降序排列 users 表中的数据。

17. 分组查询

```
SELECT age, COUNT(*) FROM users GROUP BY age;
```

说明：按 age 分组，并统计每个年龄的用户数量。

18. 创建用户

```
CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'password123';
```

说明：创建一个 testuser 用户，密码为 password123。

19. 授权用户权限

```
GRANT ALL PRIVILEGES ON mydb.* TO 'testuser'@'localhost';
```

说明：授予 testuser 对 mydb 数据库的所有权限。

20. 刷新权限

```
FLUSH PRIVILEGES;
```

说明：使权限修改立即生效。

数据库概念(以MySQL为例)

以下是对 外模式、内模式、模式、数据库、实体、关系、元组、属性、码 等概念的详细解释，并结合 MySQL 的示例加以说明。

1. 数据库（Database）

数据库是存储数据的容器，它组织、存储和管理数据，支持用户对数据的增、删、改、查操作。

示例

```
CREATE DATABASE mydb;
```

此命令创建了一个名为 mydb 的数据库，mydb 就是一个 数据库，可以存放多个表及其数据。

2. 模式（Schema）

模式是数据库的逻辑结构，包括 表结构、视图、索引、存储过程 等定义。它描述了数据库的组织方式，相当于数据库的“蓝图”。

示例

```sql
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    age INT,
    email VARCHAR(100)
);
```

这里定义了 users 表的结构（模式）：包含 id、name、age、email 四个字段，以及 id 作为主键的规则。模式不会存储数据，只定义数据的结构。

3. 内模式（Internal Schema）

内模式是数据库的 存储结构，即数据在物理存储设备上的组织方式。它涉及索引、存储格式、分区等，与数据库的存储引擎（如 InnoDB、MyISAM）相关。

示例

```sql
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    age INT,
    email VARCHAR(100)
) ENGINE=InnoDB;
```

ENGINE=InnoDB 指定 users 表使用 InnoDB 存储引擎，决定了数据的存储方式、索引结构等。这属于 内模式 的范畴。

4. 外模式（External Schema）

外模式是数据库的 用户视图，不同用户可以看到不同的数据库内容。例如，普通用户可能只能看到部分表或部分字段，而管理员可以看到全部数据。

示例

```
CREATE VIEW user_info AS
SELECT name, age FROM users;
```

这个视图 user_info 让用户只能看到 name 和 age，而 email 则不可见。这是 外模式，即用户所能访问的数据视图。

5. 实体（Entity）

实体是数据库中可以独立存在的对象，通常对应于现实世界的事物，可以是一个人、一辆车、一家公司等。实体在数据库中通常用 表（Table） 来表示。

示例

```sql
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT
);
```

这里，students 表表示“学生”这个 实体，其中的每一行代表一个具体的学生。

6. 关系（Relation）

关系是实体之间的联系，通常通过 外键（FOREIGN KEY） 来实现。数据库是 关系型数据库（Relational Database），就是因为表（实体）之间通过关系连接。

示例

```sql
CREATE TABLE courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(100)
);
```

```sql
CREATE TABLE enrollments (
    student_id INT,
    course_id INT,
    PRIMARY KEY (student_id, course_id),
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);
```

这里：

- students 表代表“学生”实体，
- courses 表代表“课程”实体，
- enrollments 表表示“学生选课”关系，每条记录表示某个学生选修了某门课程，student_id 和 course_id 通过外键建立关系。

7. 元组（Tuple）

元组是 表中的一行数据，即数据库中的一条记录。

示例

```
INSERT INTO users (name, age, email) VALUES ('Alice', 25, 'alice@example.com');
```

这条 INSERT 语句向 users 表中插入了一条记录：

```
id        name      age       email
1         Alice     25        alice@example.com
```

这个 数据行（记录） 就是一个 元组。

8. 属性（Attribute）

属性是 表中的字段，即每个列（Column），用于存储实体的某个特征。

示例

在 users 表中：

```
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    age INT,
    email VARCHAR(100)
);
```

- id、name、age、email 都是 属性，分别表示用户的 ID、姓名、年龄和邮箱。

9. 码（Key）

码是能唯一标识元组的属性或属性组合。常见的码有：

- 候选码（Candidate Key）：唯一标识元组的最小属性集合。
- 主码（Primary Key）：从候选码中选出的唯一标识记录的键。
- 外键（Foreign Key）：用于建立表之间的关系。

示例

```sql
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,     -- 主码 (Primary Key)
    name VARCHAR(50),
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)  -- 外键 (Foreign Key)
);
```

- emp_id 是 主码，唯一标识员工。
- dept_id 是 外键，表示员工属于哪个部门，连接 departments 表。

总结

概念	对应 MySQL 示例
数据库（Database）	`CREATE DATABASE mydb;`
模式（Schema）	`CREATE TABLE users (...);`
内模式（Internal Schema）	`ENGINE=InnoDB;`
外模式（External Schema）	`CREATE VIEW user_info AS SELECT name, age FROM users;`
实体（Entity）	`CREATE TABLE students (...);`
关系（Relation）	`FOREIGN KEY (student_id) REFERENCES students(student_id);`
元组（Tuple）	`INSERT INTO users VALUES (...);`
属性（Attribute）	`name VARCHAR(50), age INT;`
码（Key）	`PRIMARY KEY (id);` `FOREIGN KEY (dept_id);`