

MySQL 初始使用指南（从零到实战 | 指令全覆盖版）

本文目标：

- 每一类 SQL 指令都有明确说明
- 每一条常用指令都配有可直接运行的样例
- 每一种数据类型都有使用场景 + 示例
- 适合作为：长期查阅手册 / 博客系列第一篇 / 面试复习底稿

一、MySQL 与 SQL 的基础认知

1.1 MySQL 是什么

MySQL 是一个 **关系型数据库管理系统 (RDBMS)**，特点是：

- 使用「表 (Table)」存储数据
- 使用「SQL」与数据库交互
- 强结构、强约束、强一致性

1.2 SQL 是什么

SQL (Structured Query Language) 是操作关系型数据库的标准语言。

🔗 SQL 本身不是某个数据库独有，但不同数据库会有扩展语法。

1.3 SQL 指令的四大分类（必背）

分类	全称	作用	是否常用
DDL	Data Definition Language	定义结构	★★★★★
DML	Data Manipulation Language	操作数据	★★★★★★
DQL	Data Query Language	查询数据	★★★★★★
DCL	Data Control Language	权限控制	★★

二、数据库级指令（DDL）【附完整示例】

2.1 查看当前 MySQL 中的所有数据库

代码块

```
1 SHOW DATABASES;
```

📌 说明：

- 仅展示你有权限访问的数据库

2.2 创建数据库

代码块

```
1 CREATE DATABASE demo_db;
```

推荐写法（避免中文、表情乱码）：

代码块

```
1 CREATE DATABASE demo_db
2 DEFAULT CHARSET = utf8mb4
3 COLLATE = utf8mb4_general_ci;
```

2.3 使用数据库（非常重要）

代码块

```
1 USE demo_db;
```

🔗 后续所有表操作都基于当前数据库

2.4 删除数据库（危险指令）

代码块

```
1 DROP DATABASE demo_db;
```

⚠ 数据不可恢复

三、表级指令（DDL）【结构设计核心】

3.1 创建表（完整标准示例）

代码块

```
1 CREATE TABLE user (  
2   id INT PRIMARY KEY AUTO_INCREMENT COMMENT '用户ID',  
3   username VARCHAR(50) NOT NULL COMMENT '用户名',  
4   age INT COMMENT '年龄',  
5   email VARCHAR(100) UNIQUE COMMENT '邮箱',  
6   created_at DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间'  
7 );
```

🔗 字段说明：

字段	含义
PRIMARY KEY	主键，唯一标识一行
AUTO_INCREMENT	自增
NOT NULL	不允许为空
UNIQUE	唯一约束
COMMENT	字段注释

3.2 查看表结构

代码块

```
1 DESC user;
```

或:

代码块

```
1 SHOW CREATE TABLE user;
```

3.3 查看当前数据库的所有表

代码块

```
1 SHOW TABLES;
```

3.4 删除表

代码块

```
1 DROP TABLE user;
```

四、MySQL 常见数据类型（分类 + 示例）

4.1 整数类型

类型	字节	示例	说明
INT	4	age INT	最常用
BIGINT	8	id BIGINT	用户量极大

代码块

```
1 age INT COMMENT '年龄'
```

4.2 字符串类型（重点）

类型	示例	使用场景
CHAR	CHAR(10)	定长，如性别
VARCHAR	VARCHAR(255)	用户名、标题
TEXT	TEXT	长文本

代码块

```
1 username VARCHAR(50)
```

4.3 时间类型

类型	示例	说明
DATE	2025/1/1	仅日期
DATETIME	2025/1/1 12:00	常用
TIMESTAMP	自动更新	与时区相关

代码块

```
1 created_at DATETIME DEFAULT CURRENT_TIMESTAMP
```

五、数据操作指令（DML）【必会】

5.1 插入数据（INSERT）

代码块

```
1 INSERT INTO user (username, age, email)
2 VALUES ('Tom', 18, 'tom@qq.com');
```

插入多条：

代码块

```
1 INSERT INTO user (username, age)
2 VALUES ('Alice', 20), ('Bob', 22);
```

5.2 修改数据（UPDATE）

代码块

```
1 UPDATE user
2 SET age = 19
3 WHERE username = 'Tom';
```

⚠ 没有 WHERE 会更新整张表

5.3 删除数据（DELETE）

代码块

```
1 DELETE FROM user WHERE id = 1;
```

六、查询指令（DQL）【使用频率最高】

6.1 查询所有数据

代码块

```
1 SELECT * FROM user;
```

6.2 指定列查询

代码块

```
1 SELECT username, age FROM user;
```

6.3 条件查询 (WHERE)

代码块

```
1 SELECT * FROM user WHERE age >= 18;
```

常用条件:

代码块

```
1 age > 18
2 age BETWEEN 18 AND 25
3 username LIKE 'T%'
4 id IN (1,2,3)
```

6.4 排序 (ORDER BY)

代码块

```
1 SELECT * FROM user ORDER BY age DESC;
```

6.5 分页 (LIMIT)

代码块

```
1 SELECT * FROM user LIMIT 0, 10;
```

七、索引 (Index) 【性能关键】

7.1 创建索引

代码块

```
1 CREATE INDEX idx_username ON user(username);
```

7.2 查看索引

代码块

```
1 SHOW INDEX FROM user;
```

📌 索引适合:

- WHERE
- ORDER BY
- JOIN

八、表关系 (入门理解)

8.1 一对多设计示例

代码块

```
1 CREATE TABLE class (  
2     id INT PRIMARY KEY AUTO_INCREMENT,  
3     name VARCHAR(50)  
4 );  
5  
6 CREATE TABLE student (  
7     id INT PRIMARY KEY AUTO_INCREMENT,  
8     name VARCHAR(50),  
9     class_id INT  
10 );
```

九、新手常见错误总结

- 忘记 WHERE
- 表设计随意
- 不加索引
- 字符集不是 utf8mb4

十、学习路线建议

- 1 熟练 CRUD + 查询
- 2 学会索引和表设计
- 3 掌握 JOIN / 事务
- 4 面试专项训练

十一、力扣训练计划

1. 阶段 0：找回「最基本操作感」

目标：SELECT / WHERE / ORDER BY / LIMIT 不再手生

👉 必刷（8 题左右）

题号	题名	你练什么
1757	可回收且低脂的产品	SELECT + WHERE
584	寻找用户推荐人	WHERE + !=
595	大的国家	多条件 AND / OR
1148	文章浏览 I	去重 DISTINCT
1683	无效推文	LENGTH()
1873	计算特殊奖金	CASE WHEN
196	删除重复的电子邮箱	子查询 + 删除思路
627	变更性别	CASE WHEN

👉 这一阶段要求

- 不查语法
- 能一眼写出 `SELECT ... FROM ... WHERE ...`

2. 阶段 1: 过滤 + 去重 + 排序

目标: 恢复真实查询场景

👉 必刷 (6 题)

题号	题名	重点
607	销售员	NOT IN
610	判断三角形	CASE WHEN
1050	合作过至少三次的演员和导演	GROUP BY
1068	产品销售分析 I	JOIN
1587	银行账户概要 II	HAVING
197	上升的温度	自连接

👉 这一阶段你会重新想起

- GROUP BY \neq WHERE
- HAVING 是干嘛的
- JOIN 的基本写法

3. 阶段 2: 聚合函数 (面试高频)

目标: COUNT / SUM / AVG / MAX / MIN 条件统计

👉 核心必刷 (8 题)

题号	题名	面试含金量
175	组合两个表	★ JOIN 基础
176	第二高的薪水	★ 子查询
177	第N高的薪水	★★
178	分数排名	★★
180	连续出现的数字	★★
184	部门工资最高的员工	★★
185	部门工资前三高的所有员工	★★★★
550	游戏玩法分析 IV	★★★★

👉 这阶段 = 面试 SQL 及格线

4. 阶段 3：子查询 & 复杂逻辑（进阶）

目标：遇到复杂表关系不慌

👉 进阶刷（选做）

题号	题名	技能
262	行程和用户	多表 + 条件统计
569	员工薪水中位数	子查询
570	至少有5名直接下属的经理	自连接
571	给定数字的频率查询中位数	思维题
574	当选者	聚合 + 排序

5. 如果你时间紧（极简版路线）

💡 10 题把 SQL 拉回面试水平

代码块

```
1 1757
2 595
3 1148
4 175
5 176
6 177
7 178
8 184
9 185
10 550
```

6. 刷题建议（非常重要）

结合你现在是 C++ / 后端方向，我强烈建议：

1. 每道题写完都想一句话

1. “这题考的是 JOIN + GROUP BY + HAVING”

2. 不要追求一次写对

- SQL 本来就是靠“改出来的”

3. 把高频模板记住

代码块

```
1 SELECT ...  
2 FROM A  
3 JOIN B ON ...  
4 GROUP BY ...  
5 HAVING ...
```