

SQL 数据库系统 实验指导书

前言

本指导书基于 MySQL 8.0 版本，围绕数据库管理、表操作、数据查询、安全控制等核心技能，设计 12 个递进式实验，覆盖从基础工具使用到高级事务处理的全流程。通过理论与实操结合，培养学生的数据库操作能力、问题解决能力及职业素养，同时融入课程思政要求，强化数据伦理、安全意识与社会责任感。

实验 1: MySQL 工具的使用

一、实验目的

1. 掌握 MySQL 8.0 社区版的下载、安装、配置及卸载流程。
2. 熟练使用命令行及图形化工具（Workbench、Navicat）登录 MySQL 数据库。

二、实验环境

- 硬件：PC 机（Windows 7/10 系统）
- 软件：MySQL 8.0、Navicat Premium 16、MySQL Workbench

三、实验内容与步骤

任务 1: 下载 MySQL 8.0 社区版

1. 通过浏览器访问 MySQL 官网（<https://www.mysql.com/>），导航至“Downloads”页面。
2. 选择“MySQL Community Server 8.0”，根据操作系统选择对应安装包下载。

任务 2: 安装 MySQL 8.0

1. 双击安装包，选择“Developer Default”安装模式，按向导完成安装。
2. 配置 root 用户密码（需符合强密码策略：包含大小写字母、数字及特殊符号）。

任务 3: 配置环境变量

3. 进入“控制面板→系统→高级系统设置→环境变量”，在“系统变量”中找到“Path”。

2. 添加 MySQL 安装目录的“bin”路径（如 C:\Program Files\MySQL\MySQL Server 8.0\bin）。
3. 验证：打开命令提示符，输入 `mysql --version`，显示版本信息即为配置成功。

任务 4：命令行登录 MySQL

1. 打开命令提示符，输入登录命令：
`mysql -u root -p`
2. 输入 root 密码，出现 `mysql>`提示符即登录成功。

任务 5：使用 Workbench 登录

1. 打开 MySQL Workbench，点击“Local Instance MySQL80”，输入 root 密码完成连接。

任务 6：使用 Navicat 登录

1. 打开 Navicat，点击“连接→MySQL”，输入连接名、主机名（localhost）、端口（3306）、用户名（root）及密码，测试连接成功后登录。

四、课程思政要求

1. **信息安全意识**：从官方渠道下载软件，避免恶意代码风险；设置强密码，保护账户安全。
2. **团队协作**：遇安装问题时，与组员共同分析排查，培养互助精神与责任担当。

实验 2：MySQL 数据库管理

一、实验目的

1. 掌握数据库的创建、删除、修改及切换操作。
2. 学会使用图形化工具管理数据库。

二、实验内容与步骤

任务 1：创建数据库 db_study

1. 登录 MySQL 后，执行命令（指定字符集为 gbk）：
`CREATE DATABASE db_study CHARACTER SET gbk;`

任务 2：使用数据库 **db_study**

1. 切换至目标数据库：

```
USE db_study;
```

任务 3：修改数据库字符集

1. 将字符集修改为 `utf8mb4`（支持生僻字及特殊符号）：

```
ALTER DATABASE db_study CHARACTER SET utf8mb4 COLLATE  
utf8mb4_general_ci;
```

2. 验证修改结果：

```
SHOW CREATE DATABASE db_study;
```

任务 4：删除数据库 **db_study**

1. 执行删除命令（谨慎操作，建议先备份）：

```
DROP DATABASE IF EXISTS db_study;
```

任务 5：图形化工具管理

1. Workbench：在左侧“Schemas”面板右键“Create Schema”，配置名称及字符集。
2. Navicat：右键连接→“新建数据库”，设置名称、字符集及排序规则。

三、课程思政要求

1. **规范操作**：创建数据库必须指定字符集，避免数据乱码；删除前需确认，防止误操作。
2. **责任担当**：小组内相互检查命令，避免因个人失误破坏集体实验环境。

实验 3：MySQL 表结构管理

一、实验目的

1. 掌握表的创建、字段添加 / 修改 / 删除及表删除操作。
2. 理解数据类型、约束（主键、唯一、非空）的作用。

二、实验内容与步骤

任务 1：创建用户登录表 **tb_login**

表结构定义：

字段名	数据类型	约束	描述
login_id	INT(5)	主键、自增长	序号（初始值 1，步长 1）
login_name	VARCHAR(20)	非空、唯一	用户名
login_password	VARCHAR(45)	非空	用户密码

1. 执行创建命令：

```
CREATE TABLE tb_login (  
    login_id INT(5) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    login_name VARCHAR(20) NOT NULL UNIQUE,  
    login_password VARCHAR(45) NOT NULL  
);
```

任务 2：添加备注字段 **login_remark**

1. 执行添加命令：

```
ALTER TABLE tb_login ADD login_remark VARCHAR(255);
```

任务 3：修改 **login_remark** 为 **TEXT** 类型

1. 执行修改命令：

```
ALTER TABLE tb_login MODIFY login_remark TEXT;
```

任务 4：删除 **login_remark** 字段

1. 执行删除命令：

```
ALTER TABLE tb_login DROP login_remark;
```

任务 5：删除表 **tb_login**

1. 执行删除命令：

```
DROP TABLE tb_login;
```

三、课程思政要求

1. **数据真实性**：选择合适数据类型（如 TEXT 存储长文本），避免类型不当导致数据失真。
2. **严谨性**：修改 / 删除前用 DESC tb_login 确认表结构，防止误操作。

实验 4：表记录的检索

一、实验目的

1. 熟练使用简单查询、聚合函数、多表连接及子查询。
2. 掌握条件筛选、排序及分组查询。

二、实验内容与步骤

任务 1：查询学分 4 的专业基础课

1. 假设课程表为 tb_course，执行命令：

```
SELECT * FROM tb_course  
  
WHERE course_credit = 4 AND course_type = '专业基础课';
```

任务 2：查询学生最高 / 最矮身高

1. 假设学生表为 tb_student，执行命令：

```
SELECT MAX(student_height) AS 最高身高, MIN(student_height) AS 最低身高  
  
FROM tb_student;
```

任务 3：查询计算机学院的班级数量

1. 连接系部表（tb_department）和班级表（tb_class）：

```
SELECT COUNT(*) AS 计算机学院班级数量  
  
FROM tb_class  
  
JOIN tb_department ON tb_class.department_id = tb_department.department_id  
  
WHERE department_name = '计算机学院';
```

任务 4：查询选修“数据库原理与应用”的学生姓名

1. 子查询实现：

```
SELECT student_name AS 姓名
FROM tb_student
WHERE student_id IN (
    SELECT student_id FROM tb_grade
    JOIN tb_course ON tb_grade.course_id = tb_course.course_id
    WHERE course_name = '数据库原理与应用'
);
```

三、课程思政要求

1. **技术服务民生：**理解查询技术在政务统计（如疫苗接种率）中的应用，培养用技术解决实际问题的意识。

实验 5：视图与索引

一、实验目的

1. 掌握视图的创建与使用，理解索引对查询效率的优化作用。
2. 学会创建唯一索引及普通索引。

二、实验内容与步骤

任务 1：创建视图统计计算机系人数

1. 执行创建命令：

```
CREATE VIEW num_computer AS
SELECT COUNT(tb_student.student_id) AS 计算机系人数
FROM tb_student
JOIN tb_class ON tb_student.class_id = tb_class.class_id
JOIN tb_department ON tb_class.department_id =
tb_department.department_id
WHERE tb_department.department_id = 'X01';
```

任务 2: 创建视图查询学生成绩

1. 执行创建命令:

```
CREATE VIEW grade_student AS
SELECT tb_student.student_name, tb_grade.grade_score
FROM tb_student
JOIN tb_grade ON tb_student.student_id = tb_grade.student_id;
```

任务 3: 创建唯一索引

1. 为课程表 tb_course 的 course_name 字段创建唯一索引:

```
CREATE UNIQUE INDEX union_cor_name ON tb_course(course_name);
```

三、课程思政要求

1. **数据安全:** 创建视图时隐藏敏感字段（如密码），明确数据使用边界。
2. **规范操作:** 删除视图 / 索引前确认依赖关系，避免影响关联功能。

实验 6: 数据处理之增删改

一、实验目的

1. 熟练使用 INSERT、UPDATE、DELETE 语句操作数据。
2. 理解数据操作的安全性与规范性。

二、实验内容与步骤

任务 1: 插入单条课程记录

1. 执行插入命令:

```
INSERT INTO tb_course (course_id, course_name, course_credit)
VALUES ('K001', 'Python 编程', 3);
```

任务 2: 插入多条课程记录

1. 执行插入命令:

```
INSERT INTO tb_course (course_id, course_name, course_credit)
VALUES
('K002', '数据结构', 4),
('K003', '操作系统', 4);
```

任务 3: 更新课程记录

1. 补充任务 2 插入的记录（如添加课程类型）：

```
UPDATE tb_course
SET course_type = '专业基础课'
WHERE course_id IN ('K002', 'K003');
```

任务 4：删除插入的课程记录

1. 执行删除命令：

```
DELETE FROM tb_course
WHERE course_id IN ('K001', 'K002', 'K003');
```

三、课程思政要求

1. **数据准确性：**插入 / 更新时确保字段类型匹配（如避免字符串过长），防止业务混乱。
2. **安全意识：**删除前备份数据，区分 DELETE（逐行删除）与 TRUNCATE（清空表）的差异。

实验 7：存储过程与游标

一、实验目的

1. 掌握存储过程的创建、调用及流程控制（WHILE、LOOP）。
2. 理解游标的生命周期管理（声明、打开、使用、关闭）。

二、实验内容与步骤

任务 1：用 WHILE 求 1-100 累加和

1. 创建存储过程：

```
DELIMITER //
CREATE PROCEDURE s_while()
BEGIN
    DECLARE sum INT DEFAULT 0;
    DECLARE num INT DEFAULT 1;
    WHILE num <= 100 DO
        SET sum = sum + num;
        SET num = num + 1;
    END WHILE;
END //
```

```
END WHILE;  
SELECT sum;  
END //  
DELIMITER ;
```

2. 调用存储过程:

```
CALL s_while();
```

任务 2: 用 LOOP 求 1-100 累加和

1. 创建存储过程:

```
DELIMITER //  
CREATE PROCEDURE sum_loop()  
BEGIN  
    DECLARE sum INT DEFAULT 0;  
    DECLARE num INT DEFAULT 1;  
    loop_label: LOOP  
        SET sum = sum + num;  
        SET num = num + 1;  
        IF num > 100 THEN  
            LEAVE loop_label;  
        END IF;  
    END LOOP loop_label;  
    SELECT sum;  
END //  
DELIMITER ;
```

任务 3: 创建存储过程查询学生成绩

1. 带参数的存储过程:

```
DELIMITER //  
CREATE PROCEDURE student_grade(IN s_name VARCHAR(20))  
BEGIN  
    SELECT tb_grade.course_id, tb_grade.grade_score  
    FROM tb_student  
    JOIN tb_grade ON tb_student.student_id = tb_grade.student_id
```

```
WHERE tb_student.student_name = s_name;
END //
DELIMITER ;
```

三、课程思政要求

1. **权限控制**：明确存储过程的调用权限，避免数据泄露。
2. **规范流程**：游标必须按“声明 - 打开 - 使用 - 关闭”流程操作，培养严谨习惯。

实验 8：函数

一、实验目的

1. 掌握存储函数的创建与调用。
2. 学会使用系统函数（如字符串截取、日期计算）。

二、实验内容与步骤

任务 1：创建函数统计班级人数

1. 存储函数定义：

```
DELIMITER //
CREATE FUNCTION class_sum(c_name VARCHAR(20))
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total
    FROM tb_student
    JOIN tb_class ON tb_student.class_id = tb_class.class_id
    WHERE tb_class.class_name = c_name;
    RETURN total;
END //
DELIMITER ;
```

任务 2：统计学生姓氏数量

1. 用 LEFT 函数截取姓氏:

```
sql
SELECT LEFT(student_name, 1) AS 姓氏, COUNT(*) AS 数量
FROM tb_student
GROUP BY 姓氏;
```

三、课程思政要求

1. **技术严谨性:** 函数创建需检查语法, 避免 “ERROR 1064” 等错误, 培养细节把控能力。

实验 9: 触发器

一、实验目的

1. 掌握触发器的创建与管理 (插入 / 更新 / 删除触发)。
2. 理解触发器在数据安全中的作用 (如防止敏感数据修改)。

二、实验内容与步骤

任务 1: 保护人文与教育学院地址不被修改

1. 创建更新前触发器:

```
CREATE TRIGGER before_noupdate_department
BEFORE UPDATE ON tb_department
FOR EACH ROW
BEGIN
    IF OLD.department_name = '人文与教育学院' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = '禁止修改人文与教育学院地址';
    END IF;
END;
```

任务 2: 记录班级操作日志

1. 创建插入后触发器:

```
CREATE TRIGGER after_insert_class
AFTER INSERT ON tb_class
```

```
FOR EACH ROW
INSERT INTO tb_class_logs (log_text)
VALUES ('添加了新的班级信息');
```

三、课程思政要求

1. **数据保护**：通过触发器监控敏感数据操作，树立“数据即责任”意识。

实验 10：事务处理

一、实验目的

1. 理解事务的 ACID 特性（原子性、一致性、隔离性、持久性）。
2. 掌握事务的提交（COMMIT）、回滚（ROLLBACK）及隔离级别设置。

二、实验内容与步骤

任务 1：提交与回滚操作

1. 提交事务：

```
BEGIN;
INSERT INTO tb_department VALUES ('X09', '人工智能学院',
'87471238', '1 栋');
COMMIT;
```

2. 回滚事务：

```
sql
BEGIN;
INSERT INTO tb_department VALUES ('X10', '马克思学院', '87471239',
'13 栋');
ROLLBACK;
```

任务 2：设置隔离级别

1. 读未提交（READ UNCOMMITTED）：

```
sql
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
```

三、课程思政要求

1. **责任担当**：事务操作直接影响数据完整性，需以严谨态度对待每一步。

实验 11：数据库安全

一、实验目的

1. 掌握用户创建、权限分配及角色管理。
2. 理解数据库安全的法律与伦理边界。

二、实验内容与步骤

任务 1：创建用户并授权

1. 创建用户 “王大明” 并授予 db_study 只读权限：

```
sql
```

```
CREATE USER '王大明'@'localhost' IDENTIFIED BY '888888';
```

```
GRANT SELECT ON db_study.* TO '王大明'@'localhost';
```

任务 2：创建学委角色

1. 角色创建与授权：

```
CREATE ROLE '学委';
```

```
GRANT SELECT ON db_study.tb_student TO '学委';
```

三、课程思政要求

1. **最小权限原则**：仅授予必要权限，避免权限滥用。
2. **法律意识**：未经授权访问数据可能触犯法律，严守技术底线。

实验 12：数据库备份与恢复

一、实验目的

1. 掌握 mysqldump 工具及图形化工具的备份 / 恢复操作。
2. 理解数据备份的重要性及合规要求。

二、实验内容与步骤

任务 1：用 mysqldump 导出数据

1. 命令提示符中执行：

```
mysqldump -u root -p db_study tb_class > D:\tb_class.xml
```

任务 2：用 Navicat 恢复数据

1. 右键数据库→“运行 SQL 文件”，选择备份文件 D:\db_study.sql。

三、课程思政要求

1. **数据资产意识**：备份数据等同于保护资产，需按规范存储与管理。
2. **合规操作**：遵守《网络安全法》，不泄露备份数据。