

南京邮电大学

实验报告

(2025 / 2026 学年 第二 学期)

课程名称	数据库系统基础		
实验名称	实验三：用户和权限管理		
实验时间	2026	年 6 月 18 日	
指导单位	计算机学院、软件学院、网络空间安全学院		
指导教师	彭瑶		

学生姓名	班级学号		
学院(系)	计算机学院	专 业	软件工程

实 验 报 告

实验名称	实验三：用户和权限管理		
实验类型	验证	实验学时	2
一、 实验目的和要求 （1）理解自定义函数的概念，熟练掌握使用Navicat for MySQL进行自定义函数的操作，包括自定义函数的创建方法和调用方法； （2）理解存储过程的概念，掌握使用Navicat for MySQL进行存储过程的创建、执行以及IN和OUT参数的应用方法等； （3）熟悉触发器的概念，掌握使用Navicat for MySQL进行触发器的创建以及使用过程等； （4）掌握用户和权限管理的操作方法，包括使用Navicat for MySQL进行管理用户账号、授予用户权限、查看权限、撤销权限等。			
二、 实验环境(实验设备) 硬件: 微型计算机 软件: Windows 操作系统、MySQL 5.6或更高版本、Navicat for MySQL 15或更高版本			

实验报告

三、实验原理及内容

1 自定义函数

创建一个enterprise_db数据库，并在库中创建一个职工表，命名为employee，该表包含以下字段（id, name, department, salary, year_joined），请插入不低于10条记录，进行以下程序设计。

```
CREATE DATABASE IF NOT EXISTS enterprise_db DEFAULT CHARSET utf8mb4;USE enterprise_db;CREATE TABLE employee ( id INT PRIMARY KEY, name VARCHAR(20) NOT NULL, department VARCHAR(20) NOT NULL, salary DECIMAL(10,2), year_joined INT);INSERT INTO employee VALUES (1, '陈伟', '研发部', 12000, 2018), (2, '王芳', '研发部', 11500, 2019), (3, '李娜', '市场部', 9000, 2017), (4, '张强', '研发部', 13500, 2016), (5, '赵敏', '财务部', 8800, 2020), (6, '刘洋', '市场部', 9500, 2021), (7, '孙丽', '人事部', 8200, 2019), (8, '周杰', '研发部', 10800, 2022), (9, '吴静', '财务部', 8500, 2018), (10, '郑凯', '人事部', 7800, 2023), (11, '马莉', '市场部', 10200, 2020), (12, '黄伟', '研发部', 11800, 2017);
```

查询 - enterprise_db						
1	SELECT * FROM employee;					
	id	name	department	salary	year_joined	
1	1	陈伟	研发部	12000.00	2018	
2	2	王芳	研发部	11500.00	2019	
3	3	李娜	市场部	9000.00	2017	
4	4	张强	研发部	13500.00	2016	
5	5	赵敏	财务部	8800.00	2020	
6	6	刘洋	市场部	9500.00	2021	
7	7	孙丽	人事部	8200.00	2019	
8	8	周杰	研发部	10800.00	2022	
9	9	吴静	财务部	8500.00	2018	
10	10	郑凯	人事部	7800.00	2023	
11	11	马莉	市场部	10200.00	2020	
12	12	黄伟	研发部	11800.00	2017	
12行						

(1) 请自定义一个函数，返回指定部门的所有员工工资总和，函数名称自拟并要求体现出函数的功能，给出SQL语句并输出截图：

```
DELIMITER $$CREATE FUNCTION fn_dept_total_salary(dept_name VARCHAR(50)) RETURNS DECIMAL(12,2) DETERMINISTICBEGIN DECLARE total DECIMAL(12,2); SELECT IFNULL(SUM(salary), 0) INTO total FROM employee WHERE department = dept_name; RETURN total;END$$DELIMITER ;
```

查询 - enterprise_db	
1	DELIMITER \$\$
2	CREATE FUNCTION fn_dept_total_salary(dept_name VARCHAR(50))
3	RETURNS DECIMAL(12,2) DETERMINISTIC
4	BEGIN

实验报告

四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

一、实验中遇到的主要问题及解决方法

问题一：题目里 employee 表的字段只给了 (id, name, department, salary, year_joined)，但第 3 题的触发器要求按男女统计——意识到这就缺一个 gender 字段。处理方式是 ALTER TABLE 增列、按 id 把已有 12 行的性别补上，再建一张 gender_stats 计数表，把初始男女基数 (M=6, F=6) 用一条 GROUP BY 灌进去，这样触发器只负责增量更新，不用每次重算全表。

问题二：第一次写自定义函数时直接 CREATE FUNCTION...BEGIN...END 报了 1064 语法错误，函数体里的分号被 MySQL 当成整条语句的结束符。后来用 DELIMITER \$\$ 把语句终止符临时换掉，函数体里随便写分号，结尾用 END\$\$ 收尾，再 DELIMITER ; 切回来，问题就解决了。存储过程那几道题也是同样的处理。

问题三：做 calculate_dept_avg 那题时 IN/OUT 参数顺序最初写反了，调用时 CALL calculate_dept_avg(@avg, '研发部') 直接报「Incorrect number of arguments」。规范的写法是 IN 在前 OUT 在后，调用时也按对应顺序传——OUT 参数用 @ 开头的会话变量接，结束后再 SELECT @avg 取出来才能看到结果。

问题四：授权那题里 GRANT SELECT(id, name) ON employee 写成 GRANT SELECT ON employee(id, name) 报错，括号位置一开始没记清楚。正确语法是 GRANT 权限名(列名) ON 表名，权限名后面跟列，不是表名后面跟列。另外 REVOKE 那条得显式写 REVOKE ALL PRIVILEGES, GRANT OPTION 才能把 WITH GRANT OPTION 授出去的转授权能力一并收回——只写 REVOKE ALL 是不够的。

（二）实验心得

这次实验把 MySQL 服务端的几块「主动逻辑」走了一遍：函数返回值、存储过程的过程式控制、触发器的事件响应、用户权限的访问控制。相比实验二的 SELECT 表达式，这次的写法更像写代码——DECLARE 变量、IF/WHILE 控制流（虽然这次没用到）、用 INTO 把查询结果塞进变量再 RETURN 出去，整体节奏跟在 Java 里写一个静态工具方法挺像的。

触发器是这次让我感受最深的一块。之前在前端做表单校验、在后端用 AOP 拦截方法都是「应用层主动检查」，触发器把这一层下沉到了数据库内部——只要有 INSERT 发生，无论是 Navicat 还是后端服务还是别的客户端，约束/统计都会被强制执行。这种「靠近数据」的约束方式比应用层更可靠，但调试起来确实更黑盒，触发器没触发时排查起来要去看 SHOW TRIGGERS、看权限、看是不是 AFTER/BEFORE 写错了，门槛比纯 SQL 高一点。

用户权限部分则更偏运维。GRANT 的列级权限、WITH GRANT OPTION 的转授权、REVOKE 的精确回收，这套东西如果以后真的做后端开发，正式环境的数据库账户肯定不能直接用 root，至少要按读、写、DDL 拆开几个最小权限账户。这次实验算是把这部分语法摸熟了，后面项目里建库时能直接拿来用。

（三）意见与建议（没有可省略）

实验报告

五.支撑毕业要求指标点

- ☒ 4.2-M能够根据实验方案，配置实验环境、开展实验，综合分析实验结果以获得合理有效的结论。
- ☒ 5.2-M 能够针对计算机及应用领域中的复杂工程问题，合理选择使用恰当的技术、资源和现代工程工具进行预测和模拟，并理解其局限性。

六、指导教师评语

评分细则	评分项	优秀	良好	中等	合格	不合格
	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	算法设计合理性					
	算法效能评价					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价（总成绩）		批阅人		日期	