```
use yggl;
```

```
#  1. 创建存储过程p1
DELIMITER $$
CREATE PROCEDURE p1()
BEGIN
    DECLARE emp_count INT;
    SELECT COUNT(*) INTO emp_count FROM employees;

    IF emp_count < 10 THEN
        SELECT '人太少' AS '人员状态';
    ELSE
        SELECT '满员'  AS '人员状态';
    END IF;
end $$
DELIMITER ;

-- 调用示例
CALL p1();
```

```
-- 2. 创建存储过程p2
DELIMITER $$
CREATE PROCEDURE p2()
BEGIN
    -- 创建备份表
    CREATE TABLE employees_bak LIKE employees;

    -- 插入中山路员工
    INSERT INTO employees_bak
        SELECT * FROM employees
    WHERE Address LIKE '%中山路%';

    -- 查询结果
    SELECT * FROM employees_bak;

    -- 删除备份表
    DROP TABLE employees_bak;
END $$
DELIMITER ;

-- 调用示例
CALL p2();
```

```
-- 3. 创建表和存储过程p3
CREATE TABLE randnumber (
```

```
                        id INT AUTO_INCREMENT PRIMARY KEY,
                        data INT
);

DELIMITER $$
CREATE PROCEDURE p3()
BEGIN
    DECLARE i INT DEFAULT 0;
    DECLARE rand_num INT;

    loop_lable:loop
        IF i>50 THEN
            LEAVE loop_lable;
        END IF;

        SET rand_num = FLOOR(RAND() * 30) + 1;

        IF rand_num = 18 THEN
            LEAVE loop_lable;
        END IF;

        INSERT INTO randnumber (data) VALUES (rand_num);
        SET i = i + 1;
    END loop;
END;
DELIMITER ;

-- 调用示例
CALL p3();
```

```
-- 4. 创建存储过程p4
DELIMITER $$
CREATE PROCEDURE p4(IN p_name CHAR(10), OUT income DECIMAL(7,2))
BEGIN
    SELECT s.InCome - s.OutCome INTO income
    FROM Salary s
             JOIN Employees e ON s.EmployeeID = e.EmployeeID
    WHERE e.Name = p_name
    LIMIT 1;
END $$
DELIMITER ;

-- 调用示例
CALL p4('朱骏', @income);
SELECT @income;
```

```sql
-- 5. 创建存储过程p5
DELIMITER $$
CREATE PROCEDURE p5(IN edu CHAR(6), IN x DECIMAL(5,1))
BEGIN
    UPDATE Salary s
        JOIN Employees e ON s.EmployeeID = e.EmployeeID
    SET s.InCome = s.InCome * (1 + x/100)
    WHERE e.Education = edu;
END $$
DELIMITER ;

-- 调用示例（提高硕士收入10%）
CALL p5('硕士', 10.0);
```