题目一操作指南

实验概述

本实验指南涵盖了 PostgreSQL 数据库的完整操作流程，包括环境准备、Patch 文件管理、数据库服务控制、编译安装以及测试验证等关键步骤。

一、环境准备

1.1 检查系统环境

首先确保 CentOS 7 系统已安装必要的工具：

```
# 检查 patch 命令是否已安装
which patch

# 如果未安装，使用 yum 安装
sudo yum install -y patch

# 验证安装
patch --version
```

1.2 准备工作目录

```
# 创建工作目录
mkdir -p ~/postgres-patch-work
cd ~/postgres-patch-work

# 确保在 PostgreSQL 源码目录中
cd /root/db/postgresql-15.15
```

二、Patch 文件创建与管理

2.1 使用 git diff 创建统一格式 patch

```
# 查看当前修改状态
git status

# 创建包含二进制文件和权限变化的完整 patch
git diff --binary > delete_all_complete.patch
```

2.2 查看生成的 patch 文件内容

```
# 查看 patch 文件内容
cat delete_all_complete.patch
```

三、数据库服务管理

3.1 停止数据库服务

```
# 切换到数据库用户
sudo -i -u uxdb

# 如果 pg_ctl 命令在 PATH 中
pg_ctl -D uxdb stop
```

3.2 启动数据库服务

```
# 确保在 PostgreSQL 源码目录中
cd /root/db/postgresql-15.15
sudo -i -u uxdb

# 启动数据库服务
/db/uxdb/bin/pg_ctl -D uxdb -l logfile start
```

四、编译与安装

4.1 清理编译文件

```
# 在源码目录下
cd /root/db/postgresql-15.15
make clean
```

4.2 重新编译和安装

```
# 完整重新编译并安装到目标目录
sudo make clean && sudo make && sudo make install
```

4.3 重新安装插件

```
cd contrib/pageinspect/
sudo make clean && sudo make && sudo make install
```

五、数据库测试

5.1 执行测试脚本

```
# 连接数据库
psql -U uxdb

# 设置输出重定向
\o /db/result.txt

# 执行测试脚本
```

```
\i /db/uxdb/test_delete_all.sql

# 停止输出重定向
\o

# 退出数据库
\q
```

六、test_delete_all.sql

```
-- PostgreSQL DELETE ALL 语法完整测试脚本
-- 包含正确用法和错误用法的完整演示

-- 创建基础测试表
CREATE TABLE test_basic (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    age INTEGER
);

-- 插入测试数据
INSERT INTO test_basic (name, age) VALUES
    ('Alice', 25),
    ('Bob', 30),
    ('Charlie', 35);

-- 测试1: 正确的 DELETE ALL 语法
SELECT '=== 测试1: 正确的 DELETE ALL 语法 ===' as test_info;
DELETE ALL FROM test_basic;

-- 验证结果
SELECT 'DELETE ALL 后的记录数: ' as info, COUNT(*) as record_count FROM test_basic;

-- 重新插入数据
INSERT INTO test_basic (name, age) VALUES
    ('David', 40),
    ('Eve', 28),
    ('Frank', 32);

-- 测试2: DELETE ALL with RETURNING
SELECT '=== 测试2: DELETE ALL with RETURNING ===' as test_info;
DELETE ALL FROM test_basic
RETURNING name, age;

-- 创建多表测试的表
CREATE TABLE departments (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    status VARCHAR(20)
);

CREATE TABLE employees (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    dept_id INTEGER,
```

```
    salary DECIMAL(10,2)
);

-- 插入多表测试数据
INSERT INTO departments (name, status) VALUES
    ('技术部', 'active'),
    ('销售部', 'inactive'),
    ('人事部', 'active');

INSERT INTO employees (name, dept_id, salary) VALUES
    ('张三', 1, 8000.00),
    ('李四', 2, 6000.00),
    ('王五', 3, 5000.00);

-- 测试3: DELETE ALL with USING
SELECT '=== 测试3: DELETE ALL with USING 子句 ===' as test_info;
DELETE ALL FROM employees e
USING departments d
WHERE e.dept_id = d.id AND d.status = 'inactive';

-- 验证多表删除结果
SELECT '多表删除后剩余员工数: ' as info, COUNT(*) as record_count FROM employees;

-- 重新准备测试数据
INSERT INTO test_basic (name, age) VALUES
    ('Test1', 20),
    ('Test2', 25);

SELECT '=== 测试4: 错误用法演示（以下语句应该报错）===' as test_info;

-- 错误1: DELETE ALL with WHERE（这应该报语法错误）
-- 注意: 实际执行时会报错，我们用 try-catch 方式捕获错误
DO $$
BEGIN
    -- DELETE ALL FROM test_basic WHERE age > 20;
    -- 语法错误在 "ALL" 附近
    RAISE NOTICE '错误1: DELETE ALL 不支持 WHERE 子句 - 语法错误会在执行时检测到';
EXCEPTION WHEN syntax_error THEN
    RAISE NOTICE '捕获到预期的语法错误: DELETE ALL 不能与 WHERE 子句一起使用';
END $$;

-- 错误2: DELETE ALL with WHERE and RETURNING（也是语法错误）
DO $$
BEGIN
    -- DELETE ALL FROM test_basic WHERE name = 'Test1' RETURNING *;
    -- 语法错误在 "ALL" 附近
    RAISE NOTICE '错误2: DELETE ALL with WHERE and RETURNING - 也是语法错误';
EXCEPTION WHEN syntax_error THEN
    RAISE NOTICE '捕获到预期的语法错误: DELETE ALL 不能同时使用 WHERE 和 RETURNING';
END $$;

SELECT '=== 测试5: 标准 DELETE 语句对比 ===' as test_info;

-- 标准 DELETE with WHERE（正确）
SELECT '标准 DELETE with WHERE: ' as info;
DELETE FROM test_basic WHERE age > 20;
```

```sql
-- 标准 DELETE without WHERE（相当于 DELETE ALL）
SELECT '标准 DELETE without WHERE: ' as info;
DELETE FROM test_basic;

-- 测试空表的 DELETE ALL
SELECT '=== 测试7: 边界条件 ===' as test_info;
SELECT '删除空表: ' as info;
DELETE ALL FROM test_basic;

-- 测试单条记录的 DELETE ALL
INSERT INTO test_basic (name, age) VALUES ('Single', 99);
SELECT '删除单条记录: ' as info;
DELETE ALL FROM test_basic;


-- 插入测试数据
INSERT INTO test_basic (name, age) VALUES
    ('Rollback1', 30),
    ('Rollback2', 35);

SELECT '=== 测试8: 事务回滚 ===' as test_info;

BEGIN;
DELETE ALL FROM test_basic;
SELECT '事务内的记录数: ' as info, COUNT(*) as record_count FROM test_basic;
ROLLBACK;

-- 验证回滚结果
SELECT '回滚后的记录数: ' as info, COUNT(*) as record_count FROM test_basic;


-- 清理所有测试表
DROP TABLE IF EXISTS test_basic;
DROP TABLE IF EXISTS departments;
DROP TABLE IF EXISTS employees;

-- 输出测试完成信息
SELECT '=== DELETE ALL 语法完整测试结束 ===' as final_result;
```