-- Lab3 - Advanced DML Operations (Simplified Version)  
  
  
-- ================================  
-- Part A: Table Setup  
-- ================================  
  
DROP TABLE IF EXISTS employees CASCADE;  
DROP TABLE IF EXISTS departments CASCADE;  
DROP TABLE IF EXISTS projects CASCADE;  
DROP TABLE IF EXISTS employee\_archive CASCADE;  
DROP TABLE IF EXISTS temp\_employees CASCADE;  
  
CREATE TABLE employees (  
 emp\_id SERIAL PRIMARY KEY,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 department VARCHAR(50),  
 salary INTEGER,  
 hire\_date DATE,  
 status VARCHAR(20) DEFAULT 'Active'  
);  
  
CREATE TABLE departments (  
 dept\_id SERIAL PRIMARY KEY,  
 dept\_name VARCHAR(50),  
 budget INTEGER,  
 manager\_id INTEGER  
);  
  
CREATE TABLE projects (  
 project\_id SERIAL PRIMARY KEY,  
 project\_name VARCHAR(100),  
 dept\_id INTEGER,  
 start\_date DATE,  
 end\_date DATE,  
 budget INTEGER  
);  
  
-- ================================  
-- Part B: INSERT Operations  
-- ================================  
  
INSERT INTO employees (first\_name, last\_name, department)  
VALUES ('Alice', 'Johnson', 'IT');  
  
INSERT INTO employees (first\_name, last\_name)  
VALUES ('Bob', 'Smith');  
  
INSERT INTO departments (dept\_name, budget, manager\_id)  
VALUES  
('HR', 100000, 1),  
('IT', 200000, 2),  
('Sales', 150000, 3);  
  
INSERT INTO employees (first\_name, last\_name, hire\_date, salary)  
VALUES ('Charlie', 'Brown', *CURRENT\_DATE*, 50000 \* 1.1);  
  
CREATE TEMP TABLE temp\_employees AS  
SELECT \* FROM employees WHERE department = 'IT';  
  
-- ================================  
-- Part C: UPDATE Operations  
-- ================================  
  
UPDATE employees  
SET salary = salary \* 1.10;  
  
UPDATE employees  
SET status = 'Senior'  
WHERE salary > 60000 AND hire\_date < '2020-01-01';  
  
UPDATE employees  
SET department = CASE  
 WHEN salary > 80000 THEN 'Management'  
 WHEN salary BETWEEN 50000 AND 80000 THEN 'Senior'  
 ELSE 'Junior'  
END;  
  
ALTER TABLE employees ALTER COLUMN department SET DEFAULT 'General';  
UPDATE employees SET department = DEFAULT WHERE status = 'Inactive';  
  
UPDATE departments d  
SET budget = (SELECT *AVG*(salary) \* 1.2  
 FROM employees e  
 WHERE e.department = d.dept\_name);  
  
UPDATE employees  
SET salary = salary \* 1.15,  
 status = 'Promoted'  
WHERE department = 'Sales';  
  
-- ================================  
-- Part D: DELETE Operations  
-- ================================  
  
-- 13. Простое удаление  
DELETE FROM employees  
WHERE status = 'Terminated';  
  
-- 14. Сложное условие  
DELETE FROM employees  
WHERE salary < 40000  
 AND hire\_date > '2023-01-01'  
 AND department IS NULL;  
  
-- 15. DELETE с подзапросом (исправленный вариант)  
-- employees.department = departments.dept\_name (оба VARCHAR)  
DELETE FROM departments d  
WHERE NOT *EXISTS* (  
 SELECT 1  
 FROM employees e  
 WHERE e.department = d.dept\_name  
);  
  
-- 16. DELETE + RETURNING  
DELETE FROM projects  
WHERE end\_date < '2023-01-01'  
RETURNING \*;  
  
-- ================================  
-- Part E: NULL Handling  
-- ================================  
  
INSERT INTO employees (first\_name, last\_name, salary, department)  
VALUES ('Diana', 'Lee', NULL, NULL);  
  
UPDATE employees  
SET department = 'Unassigned'  
WHERE department IS NULL;  
  
DELETE FROM employees  
WHERE salary IS NULL OR department IS NULL;  
  
-- ================================  
-- Part F: RETURNING Clause  
-- ================================  
  
INSERT INTO employees (first\_name, last\_name, salary)  
VALUES ('Ethan', 'Clark', 70000)  
RETURNING emp\_id, first\_name || ' ' || last\_name AS full\_name;  
  
UPDATE employees  
SET salary = salary + 5000  
WHERE department = 'IT'  
RETURNING emp\_id, salary - 5000 AS old\_salary, salary AS new\_salary;  
  
DELETE FROM employees  
WHERE hire\_date < '2020-01-01'  
RETURNING \*;  
  
-- ================================  
-- Part G: Advanced Patterns  
-- ================================  
  
INSERT INTO employees (first\_name, last\_name)  
SELECT 'Frank', 'Miller'  
WHERE NOT *EXISTS* (  
 SELECT 1 FROM employees WHERE first\_name='Frank' AND last\_name='Miller'  
);  
  
UPDATE employees e  
SET salary = salary \* CASE  
 WHEN (SELECT budget FROM departments d WHERE d.dept\_name = e.department) > 100000  
 THEN 1.10  
 ELSE 1.05  
END;  
  
INSERT INTO employees (first\_name, last\_name, department, salary)  
VALUES  
('Anna','K','Sales',50000),  
('Max','L','IT',60000),  
('John','P','IT',55000),  
('Sara','M','HR',45000),  
('Tom','Q','Sales',52000);  
  
UPDATE employees  
SET salary = salary \* 1.10  
WHERE first\_name IN ('Anna','Max','John','Sara','Tom');  
  
CREATE TABLE employee\_archive AS TABLE employees WITH NO DATA;  
INSERT INTO employee\_archive  
SELECT \* FROM employees WHERE status = 'Inactive';  
DELETE FROM employees WHERE status = 'Inactive';  
  
UPDATE projects p  
SET end\_date = end\_date + INTERVAL '30 days'  
WHERE budget > 50000  
 AND (SELECT *COUNT*(\*) FROM employees e WHERE e.department = p.dept\_id) > 3;