

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №10
по курсу «Проектирование баз данных»

Выполнила: Прудникова А. А.
Группа: М8О-114СВ-24
Преподаватель: Моргунов Е. П.

Москва, 2024

Задание 12

Сделайте выборки данных из таблиц «Персонал» и «Организационная структура», а также реконструируйте организационную структуру с помощью двух представлений (view). Команды можно выполнять не только в среде интерактивного терминала psql, но также и из командной строки операционной системы. Выполните эти команды в командной строке операционной системы:

```
psql -d ais -c "SELECT * FROM Personnel"
psql -d ais -c "SELECT * FROM Org_chart"
psql -d ais -c "SELECT * FROM Personnel_org_chart"
psql -d ais -c "SELECT * FROM Create_paths"
```

Запрос

```
psql -h localhost -p 5432 -U postgres -d ais -c "SELECT
* FROM Personnel"
psql -h localhost -p 5432 -U postgres -d ais -c "SELECT * FROM
Org_chart"
psql -h localhost -p 5432 -U postgres -d ais -c "SELECT * FROM
Personnel_org_chart"
psql -h localhost -p 5432 -U postgres -d ais -c "SELECT * FROM
Create_paths"
```

Результат

emp_nbr	emp_name	address	birth_date	job_title	emp_nbr	boss_emp_nbr	salary
0	вакансия		2014-05-19	Президент	1		1000.0000
1	Иван	ул. Любителей языка С	1962-12-01	Вице-президент 1	2	1	900.0000
2	Петр	ул. UNIX гуру	1965-10-21	Вице-президент 2	3	1	800.0000
3	Антон	ул. Ассемблерная	1964-04-17	Архитектор	4	3	700.0000
4	Захар	ул. им. СУБД PostgreSQL	1963-09-27	Ведущий программист	5	3	600.0000
5	Ирина	просп. Программистов	1968-05-12	Программист С	6	3	500.0000
6	Анна	пер. Перловый	1969-03-20	Программист Perl	7	5	450.0000
7	Андрей	пл. Баз данных	1945-11-07	Оператор	8	5	400.0000
8	Николай	наб. ОС Linux	1944-12-01				

(9 rows)

(8 rows)

emp_nbr	emp	boss_emp_nbr	boss
1	Иван		
2	Петр	1	Иван
3	Антон	1	Иван
4	Захар	3	Антон
5	Ирина	3	Антон
6	Анна	3	Антон
7	Андрей	5	Ирина
8	Николай	5	Ирина

(8 rows)

level1	level2	level3	level4
Иван	Антон	Ирина	Андрей
Иван	Антон	Ирина	Николай
Иван	Петр		
Иван	Антон	Захар	
Иван	Антон	Анна	

(5 rows)

Задание 13

Выполните проверку структуры дерева на предмет отсутствия циклов с помощью функции `tree_test()`. Если вы еще не вносили изменения в таблицу «Организационная структура», то функция покажет отсутствие нарушения структуры дерева. Теперь создайте в таблице «Организационная структура» сначала короткий цикл, а затем длинный цикл. Для каждого из указанных циклов выполните проверку с помощью функции `tree_test()`.

Запрос

```
SELECT * FROM tree_test();
```

```
-- Иван — начальник Петра
UPDATE Org_chart
SET boss_emp_nbr = 1
WHERE emp_nbr = 2;
```

```
-- Петр — начальник Ивана
UPDATE Org_chart
SET boss_emp_nbr = 2
WHERE emp_nbr = 1;
```

Результат

	tree_test
1	Tree

```
[P0001] ERROR: Bad orgchart structure
Where: PL/pgSQL function check_org_chart() line 15 at RAISE
```

Решение

Сработал триггер. Отключим его и посмотрим, что будет со структурами.

Запрос

```
ALTER TABLE Org_chart DISABLE TRIGGER check_org_chart;
```

```
-- Иван — начальник Петра
UPDATE Org_chart
SET boss_emp_nbr = 1
WHERE emp_nbr = 2;
```

```
-- Петр — начальник Ивана
```

```
UPDATE Org_chart
SET boss_emp_nbr = 2
WHERE emp_nbr = 1;

SELECT * FROM tree_test();
```

Результат

	tree_test
1	Cycles

Решение

Проделаем теперь то же самое для длинного цикла.

Запрос

```
UPDATE Org_chart
  SET boss_emp_nbr = 1
  WHERE emp_nbr = 2;

UPDATE Org_chart
  SET boss_emp_nbr = 2
  WHERE emp_nbr = 3;

UPDATE Org_chart
  SET boss_emp_nbr = 3
  WHERE emp_nbr = 1;

SELECT * FROM tree_test();
```

Результат

	tree_test
1	Cycles

Решение

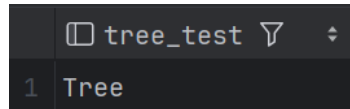
Вернем структуру дерева для дальнейшей работы.

Запрос

```
UPDATE Org_chart
  SET boss_emp_nbr = NULL
  WHERE emp_nbr = 1;
```

```
SELECT * FROM tree_test();
```

Результат



tree_test
1 Tree

Задание 14

Выполните обход дерева организационной структуры снизу вверх, начиная с конкретного узла, можно с помощью функции `up_tree_traversal()` либо функции `up_tree_traversal2()`. Сначала сделайте это с помощью первой из функций:

```
SELECT * FROM up_tree_traversal(6);
```

Параметром этих функций является код работника. Измените код работника и повторите команду. Теперь воспользуйтесь второй функцией. Учтите, что она возвращает `SETOF RECORD`, поэтому команда будет более сложной:

```
SELECT * FROM up_tree_traversal2(6) AS (emp int, boss int);
```

Очевидно, что для использования числового кода работника нужно знать этот код. Удобнее иметь дело с именем работника. Поэтому можно в качестве параметра этих функций использовать подзапрос, возвращающий код работника в качестве своего результата. Не забудьте, что текст подзапроса заключается в скобки, поэтому появляются двойные скобки:

```
SELECT * FROM up_tree_traversal((SELECT ... FROM Personnel WHERE ...));
```

Завершите эту команду и выполните ее с различными именами работников.

Запрос

```
SELECT * FROM up_tree_traversal(6);
```

```
SELECT * FROM up_tree_traversal(2);
```

```
SELECT * FROM up_tree_traversal2(6) AS (emp int, boss int);
```

```
SELECT * FROM up_tree_traversal((SELECT emp_nbr FROM Personnel WHERE emp_name = 'Анна'));
```

```
SELECT * FROM up_tree_traversal((SELECT emp_nbr FROM Personnel WHERE emp_name = 'Иван'));
```

Результат

	emp_nbr	boss_emp_nbr
1	6	3
2	3	2
3	2	1
4	1	<null>

	emp_nbr	boss_emp_nbr
1	2	1
2	1	<null>

	emp	boss
1	6	3
2	3	2
3	2	1
4	1	<null>

	emp_nbr	boss_emp_nbr
1	6	3
2	3	2
3	2	1
4	1	<null>

	emp_nbr	boss_emp_nbr
1	1	<null>

Задание 15

Выполните операцию удаления поддерева с помощью функции `delete_subtree()`. Параметром функции является код работника:

```
SELECT * FROM delete_subtree(6);
```

Аналогично работе с функцией `up_tree_traversal()` используйте подзапрос для получения кода работника по его имени. После удаления поддерева посмотрите, что стало с организационной структурой с помощью двух представлений `Personnel_org_chart` и `Create_paths`.

Запрос

```
SELECT * FROM delete_subtree(6);
```

```
SELECT * FROM delete_subtree((SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Николай'));
```

```
SELECT * FROM Personnel_org_chart;
```

```
SELECT * FROM Create_paths;
```

Результат

	emp_nbr	emp	boss_emp_nbr	boss
1	4	Захар	3	Антон
2	5	Ирина	3	Антон
3	7	Андрей	5	Ирина
4	2	Петр	1	Иван
5	3	Антон	2	Петр
6	1	Иван	<null>	<null>

	level1	level2	level3	level4
1	Иван	Петр	Антон	Захар
2	Иван	Петр	Антон	Ирина

Задание 16

Если в таблице «Организационная структура» осталось мало данных, дополните ее и выполните удаление элемента иерархии с продвижением дочерних элементов на один уровень вверх (т. е. к «бабушке»):

```
SELECT * FROM delete_and_promote_subtree(5);
```

Аналогично работе с функцией `up_tree_traversal()`, используйте подзапрос для получения кода работника по его имени. После удаления элемента иерархии проверьте, что произошло с организационной структурой, с помощью двух представлений: `Personnel_org_chart` и `Create_paths`.

Запрос

```
SELECT * FROM delete_and_promote_subtree((SELECT emp_nbr FROM
Personnel WHERE emp_name = 'Ирина'));
```

```
SELECT * FROM Personnel_org_chart;
```

```
SELECT * FROM Create_paths;
```

Результат

	emp_nbr	emp	boss_emp_nbr	boss
1	4	Захар	3	Антон
2	2	Петр	1	Иван
3	3	Антон	2	Петр
4	1	Иван	<null>	<null>
5	7	Андрей	3	Антон

	level1 ▾	÷	level2 ▾	÷	level3 ▾	÷	level4 ▾	÷
1	Иван		Петр		Антон		Захар	
2	Иван		Петр		Антон		Андрей	

Задание 17

Представление `Create_paths` позволяет отобразить только четыре уровня иерархии. Модифицируйте его так, чтобы оно могло работать с пятью уровнями иерархии.

Решение

Для начала дополним таблицу данными.

Запрос

```
INSERT INTO personnel (emp_nbr, emp_name, address, birth_date)
VALUES
```

```
    (10, 'Вадим', 'ул. Программистов', '1980-05-15'),
    (11, 'Максим', 'ул. Кодеров', '1982-06-17'),
    (12, 'Анастасия', 'пер. Фронта', '1985-03-10'),
    (13, 'Матвей', 'ул. Бэкенда', '1987-09-22');
```

```
INSERT INTO Org_chart (emp_nbr, job_title, boss_emp_nbr,
salary)
```

```
VALUES
```

```
    (10, 'Программист', 7, 120.00),
    (11, 'HR', 7, 300.00),
    (12, 'Дизайнер интерфейса', 4, 350.00),
    (13, 'Тестировщик', 4, 500.00);
```

```
SELECT * FROM personnel_org_chart;
```

Результат

	emp_nbr ▾	÷	emp ▾	÷	boss_emp_nbr ▾	÷	boss ▾	÷
1		4	Захар			3	Антон	
2		2	Петр			1	Иван	
3		3	Антон			2	Петр	
4		1	Иван		<null>	<null>		
5		7	Андрей			3	Антон	
6		10	Вадим			7	Андрей	
7		11	Максим			7	Андрей	
8		12	Анастасия			4	Захар	
9		13	Матвей			4	Захар	

Запрос

```
DROP VIEW IF EXISTS Create_paths;

CREATE VIEW Create_paths (level1, level2, level3, level4,
level5) AS
    SELECT 01.emp AS e1, 02.emp AS e2, 03.emp AS e3,
           04.emp AS e4, 05.emp AS e5
    FROM Personnel_org_chart AS 01
    LEFT OUTER JOIN Personnel_org_chart AS 02
        ON 01.emp = 02.boss
    LEFT OUTER JOIN Personnel_org_chart AS 03
        ON 02.emp = 03.boss
    LEFT OUTER JOIN Personnel_org_chart AS 04
        ON 03.emp = 04.boss
    LEFT OUTER JOIN Personnel_org_chart AS 05 -- Изменено с
04 на 05
        ON 04.emp = 05.boss
    WHERE 01.emp = 'Иван';

SELECT * FROM Create_paths;
```

Результат

	level1	level2	level3	level4	level5
1	Иван	Петр	Антон	Андрей	Вадим
2	Иван	Петр	Антон	Андрей	Максим
3	Иван	Петр	Антон	Захар	Анастасия
4	Иван	Петр	Антон	Захар	Матвей

Задание 18

Самостоятельно ознакомьтесь с таким средством работы с таблицами базы данных, как курсоры (cursors). Воспользуйтесь технической документацией на PostgreSQL, глава «PL/pgSQL – SQL Procedural Language». Напишите небольшую функцию с применением курсора.

Запрос

```
CREATE OR REPLACE FUNCTION process_salaries()
RETURNS VOID AS $$
DECLARE
    e_name TEXT;
    emp_salary NUMERIC;
    cursor_salary CURSOR FOR
```

```

        SELECT (SELECT emp_name FROM personnel WHERE
personnel.emp_nbr = org_chart.emp_nbr), salary
        FROM org_chart
        WHERE salary > 500;
BEGIN
    OPEN cursor_salary;

    LOOP
        FETCH cursor_salary INTO e_name, emp_salary;
        EXIT WHEN NOT FOUND;
        RAISE NOTICE 'Employee: %, Salary: %', e_name,
emp_salary;
    END LOOP;

    CLOSE cursor_salary;
END;
$$ LANGUAGE plpgsql;

```

```
SELECT * FROM process_salaries();
```

Результат

```

Employee: Захар, Salary: 700.0000
Employee: Петр, Salary: 900.0000
Employee: Антон, Salary: 800.0000
Employee: Иван, Salary: 1000.0000
[2024-10-24 14:07:28] 1 row retrieved starting from 1 in 58 ms (execution: 5 ms, fetching: 53 ms)

```