МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

“Московский авиационный институт”

(национальный исследовательский университет)

Факультет №3 «Системы управления, информатика и электроэнергетика»

Отчет по домашней работе №10

по курсу «**Управление базами данных**»

Выполнили:

студенты группы 3О-218М-19:

Пономарев Роман

Принял:

Доцент, к.т.н. Моргунов Е.П.

Москва, 2020

1. 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"

Не забудьте, что если не указан параметр -U, то утилита psql подключается к базе данных от имени пользователя базы данных, имя которого совпадает с именем пользователя операционной системы. Поэтому возможно, что вам придется использовать параметр -U, если в базе данных не создана учетная запись такого пользователя.

```
postgres@brain:/home/WORK/Databases/Admin_DB/UTF-8$ psql -d ais -c "SELECT * FROM Personnel"
 emp_nbr | emp_name |          address          | birth_date
---------+----------+---------------------------+------------
       0 | вакансия |                           | 2014-05-19
       1 | Иван     | ул. Любителей языка C     | 1962-12-01
       2 | Петр     | ул. UNIX гуру             | 1965-10-21
       3 | Антон    | ул. Ассемблерная          | 1964-04-17
       4 | Захар    | ул. им. СУБД PostgreSQL   | 1963-09-27
       5 | Ирина    | просп. Программистов      | 1968-05-12
       6 | Анна     | пер. Перловый             | 1969-03-20
       7 | Андрей   | пл. Баз данных            | 1945-11-07
       8 | Николай  | наб. ОС Linux             | 1944-12-01
(9 строк)
```

Рисунок 1. Код запроса и его выполнение.

```
postgres@brain:/home/WORK/Databases/Admin_DB/UTF-8$ psql -d ais -c "SELECT * FROM Org_chart"
      job_title      | emp_nbr | boss_emp_nbr |  salary
---------------------+---------+--------------+-----------
 Президент           |       1 |              | 1000.0000
 Вице-президент 1    |       2 |            1 |  900.0000
 Вице-президент 2    |       3 |            1 |  800.0000
 Архитектор          |       4 |            3 |  700.0000
 Ведущий программист |       5 |            3 |  600.0000
 Программист C       |       6 |            3 |  500.0000
 Программист Perl    |       7 |            5 |  450.0000
 Оператор            |       8 |            5 |  400.0000
(8 строк)
```

Рисунок 2. Код запроса и его выполнение.

```
postgres@brain:/home/WORK/Databases/Admin_DB/UTF-8$ psql -d ais -c "SELECT * FROM Personnel_org_chart"
 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 строк)
```

Рисунок 3. Код запроса и его выполнение.

```
postgres@brain:/home/WORK/Databases/Admin_DB/UTF-8$ psql -d ais -c "SELECT * FROM Create_paths"
 level1 | level2 | level3 | level4
--------+--------+--------+---------
 Иван   | Антон  | Ирина  | Андрей
 Иван   | Антон  | Ирина  | Николай
 Иван   | Петр   |        |
 Иван   | Антон  | Захар  |
 Иван   | Антон  | Анна   |
(5 строк)
```

Рисунок 4. Код запроса и его выполнение.

2. Задание №13

Выполните проверку структуры дерева на предмет отсутствия циклов с помощью функции tree_test().

SELECT * FROM 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().

```
ais=# UPDATE Org_chart
SET boss_emp_nbr = 8
WHERE job_title = 'Ведущий программист';
UPDATE 1
ais=# select * from org_chart;
      job_title      | emp_nbr | boss_emp_nbr |  salary
---------------------+---------+--------------+-----------
 Президент           |       1 |              | 1000.0000
 Вице-президент 1    |       2 |            1 |  900.0000
 Архитектор          |       4 |            3 |  700.0000
 Программист C       |       6 |            3 |  500.0000
 Программист Perl    |       7 |            5 |  450.0000
 Оператор            |       8 |            5 |  400.0000
 Вице-президент 2    |       3 |            1 |  800.0000
 Ведущий программист |       5 |            8 |  600.0000
(8 строк)

ais=# SELECT * FROM tree_test();
 tree_test
-----------
 Cycles
(1 строка)
```

Рисунок 5. Код запроса и его выполнение.

```
ais=# UPDATE Org_chart
SET boss_emp_nbr = 8
WHERE job_title = 'Вице-президент 2';
UPDATE 1
ais=# select * from org_chart;
      job_title      | emp_nbr | boss_emp_nbr |  salary
---------------------+---------+--------------+-----------
 Президент           |       1 |              | 1000.0000
 Вице-президент 1    |       2 |            1 |  900.0000
 Архитектор          |       4 |            3 |  700.0000
 Программист C       |       6 |            3 |  500.0000
 Программист Perl    |       7 |            5 |  450.0000
 Оператор            |       8 |            5 |  400.0000
 Ведущий программист |       5 |            3 |  600.0000
 Вице-президент 2    |       3 |            8 |  800.0000
(8 строк)

ais=# SELECT * FROM tree_test();
 tree_test
-----------
 Cycles
(1 строка)
```

Рисунок 6. Код запроса и его выполнение.

3. 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.

```
ais=# select * from up_tree_traversal(6);
 emp_nbr | boss_emp_nbr
---------+--------------
       6 |            3
       3 |            1
       1 |
(3 строки)

ais=# SELECT * FROM up_tree_traversal2( 2 ) AS (emp int, boss int);
 emp | boss
-----+------
   2 |    1
   1 |
(2 строки)

ais=#
```

Рисунок 7. Код запроса и его выполнение.

```
 emp_nbr | emp_name |         address          | birth_date
---------+----------+--------------------------+------------
       0 | вакансия |                          | 2014-05-19
       1 | Иван     | ул. Любителей языка C    | 1962-12-01
       2 | Петр     | ул. UNIX гуру            | 1965-10-21
       3 | Антон    | ул. Ассемблерная         | 1964-04-17
       4 | Захар    | ул. им. СУБД PostgreSQL  | 1963-09-27
       5 | Ирина    | просп. Программистов     | 1968-05-12
       6 | Анна     | пер. Перловый            | 1969-03-20
       7 | Андрей   | пл. Баз данных           | 1945-11-07
       8 | Николай  | наб. ОС Linux            | 1944-12-01
(9 строк)

ais=#
SELECT * FROM up_tree_traversal (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Петр' ));
 emp_nbr | boss_emp_nbr
---------+--------------
       2 |            1
       1 |
(2 строки)

ais=#
SELECT * FROM up_tree_traversal (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Николай' ));
 emp_nbr | boss_emp_nbr
---------+--------------
       8 |            5
       5 |            3
       3 |            1
       1 |
(4 строки)

ais=#
SELECT * FROM up_tree_traversal (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Анна' ));
 emp_nbr | boss_emp_nbr
---------+--------------
       6 |            3
       3 |            1
       1 |
(3 строки)
```

Рисунок 8. Код запроса и его выполнение.

4. 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.

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

(1 строка)

ais=# SELECT * FROM delete_subtree (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Ирина' ));
 delete_subtree
----------------

(1 строка)
```

Рисунок 9. Код запроса и его выполнение.

```
ais=# SELECT * FROM Personnel_org_chart;
 emp_nbr |  emp   | boss_emp_nbr | boss
---------+--------+--------------+------
       1 | Иван   |              |
       2 | Петр   |            1 | Иван
       3 | Антон  |            1 | Иван
(3 строки)

ais=# SELECT * FROM Create_paths;
 level1 | level2 | level3 | level4
--------+--------+--------+--------
 Иван   | Антон  |        |
 Иван   | Петр   |        |
(2 строки)
```

Рисунок 10. Код запроса и его выполнение.

5. Задание №16

Если в таблице «Организационная структура» осталось мало данных, то дополните ее данными и выполните удаление элемента иерархии и продвижение 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 посмотрите, что стало с организационной структурой, с помощью двух представлений Personnel_org_chart и Create_paths.

```
ais=# SELECT * FROM delete_and_promote_subtree( 5 );
 delete_and_promote_subtree
----------------------------

(1 строка)
```

Рисунок 11. Код запроса и его выполнение.

```
ais=# SELECT * FROM delete_and_promote_subtree (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Антон' ));
 delete_and_promote_subtree
----------------------------

(1 строка)

ais=# SELECT * FROM delete_and_promote_subtree (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Анна' ));
 delete_and_promote_subtree
----------------------------

(1 строка)

ais=# SELECT * FROM Personnel_org_chart;
 emp_nbr |   emp   | boss_emp_nbr | boss
---------+---------+--------------+------
       1 | Иван    |              |
       2 | Петр    |            1 | Иван
       7 | Андрей  |            1 | Иван
       8 | Николай |            1 | Иван
(4 строки)

ais=# SELECT * FROM Create_paths;
 level1 | level2  | level3 | level4
--------+---------+--------+--------
 Иван   | Андрей  |        |
 Иван   | Николай |        |
 Иван   | Петр    |        |
(3 строки)
```

Рисунок 12. Код запроса и его выполнение.

6. Задание №17

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

```
ais=# SELECT * FROM delete_and_promote_subtree (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Антон' ));
 delete_and_promote_subtree
----------------------------

(1 строка)

ais=# SELECT * FROM delete_and_promote_subtree (( SELECT emp_nbr FROM Personnel
WHERE emp_name = 'Анна' ));
 delete_and_promote_subtree
----------------------------

(1 строка)

ais=# SELECT * FROM Personnel_org_chart;
 emp_nbr |   emp   | boss_emp_nbr | boss
---------+---------+--------------+------
       1 | Иван    |              |
       2 | Петр    |            1 | Иван
       7 | Андрей  |            1 | Иван
       8 | Николай |            1 | Иван
(4 строки)

ais=# SELECT * FROM Create_paths;
 level1 | level2  | level3 | level4
--------+---------+--------+--------
 Иван   | Андрей  |        |
 Иван   | Николай |        |
 Иван   | Петр    |        |
(3 строки)
```

Рисунок 13. Код запроса и его выполнение.

7. 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.

```
ais=# CREATE FUNCTION ref_func() RETURNS refcursor AS '
DECLARE
    ref refcursor;
BEGIN
    OPEN ref FOR SELECT emp_nbr FROM personnel;
    RETURN ref;
END;
' LANGUAGE plpgsql;
CREATE FUNCTION
ais=# BEGIN;
BEGIN
ais=# SELECT ref_func();
      ref_func
--------------------
 <unnamed portal 1>
(1 строка)

ais=# FETCH ALL IN "<unnamed portal 1>";
 emp_nbr
---------
       0
       1
       2
       3
       4
       5
       6
       7
       8
(9 строк)
```

Рисунок 14. Код запроса и его выполнение.