ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Старший преподаватель		Н.В Путилова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

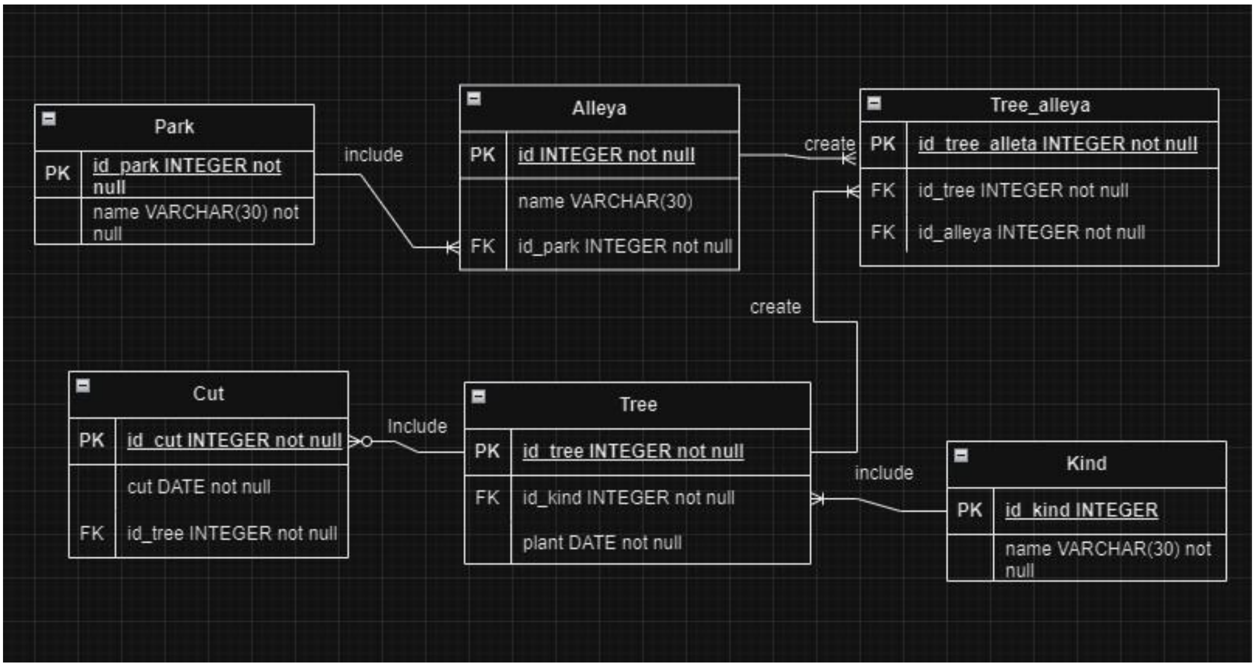
Хранимые процедуры

по дисциплине: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4134к		Костяков Н.А.
		подпись, дата	инициалы, фамилия

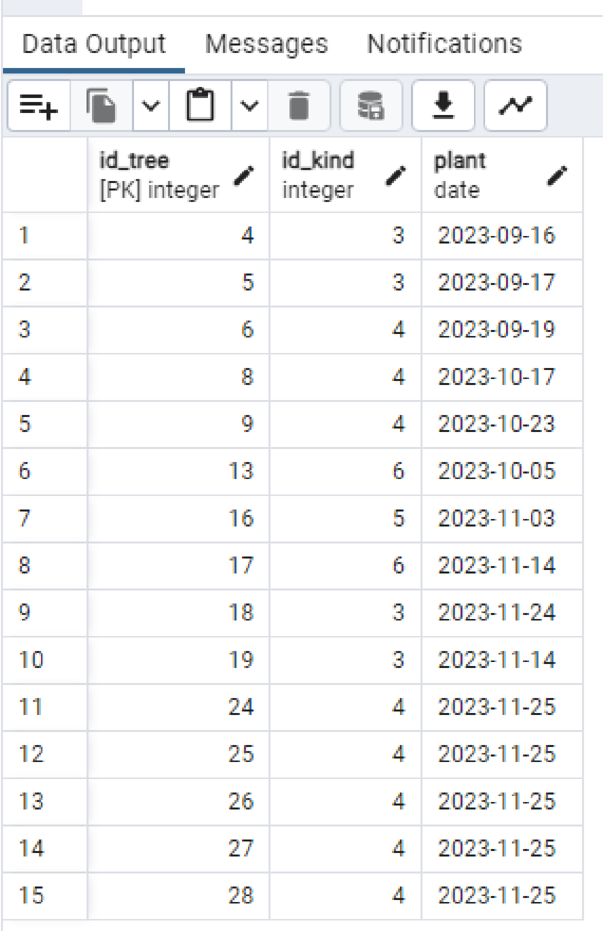
Санкт-Петербург
2023



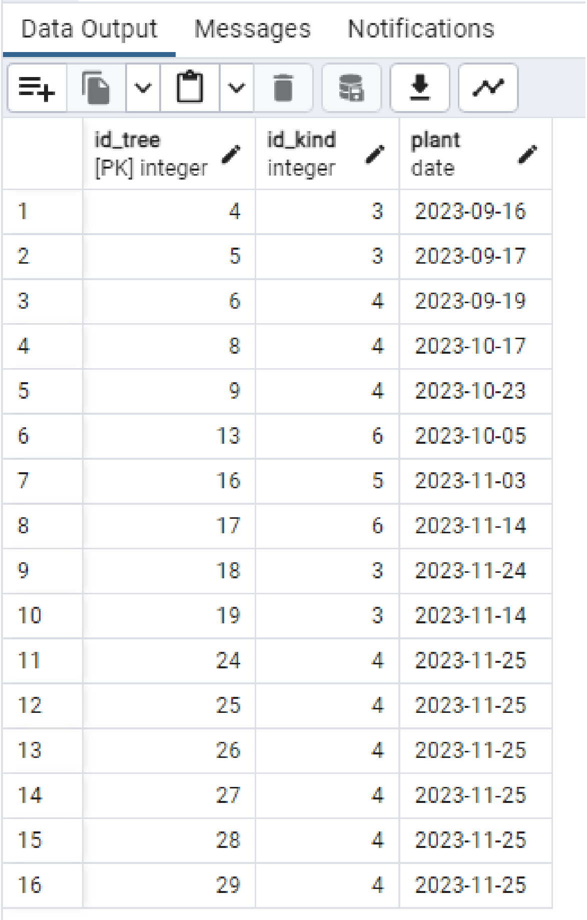
Процедура вставки с пополнением

```
CREATE OR REPLACE PROCEDURE ins_tree(ins_kind VARCHAR(128), ins_plant DATE)
LANGUAGE plpgsql
AS $$
DECLARE
    id_all_new INT;
    id_park_new INT;
    id_kind_new INT;
    id_t_a INT;
    id_tree_new INT;
BEGIN


    -- Блок с видами деревьев

        IF EXISTS (SELECT 1 FROM kind WHERE kind.name = ins_kind) THEN
            SELECT kind.id_kind INTO id_kind_new FROM kind WHERE kind.name =
ins_kind;
        ELSE
            id_kind_new := (NULLIF((SELECT MAX(id_kind) FROM kind), 0) + 1);
            INSERT INTO kind(id_kind, name) VALUES (id_kind_new, ins_kind);
        END IF;


    -- Блок с деревьями и аллеями

    id_tree_new := (SELECT COALESCE(MAX(id_tree), 0) + 1 FROM tree);
    INSERT INTO tree(id_tree, id_kind, plant) VALUES (id_tree_new, id_kind_new,
ins_plant);

END;
$$;

CALL ins_tree('Сакура', '2023-12-11');
```

Слева – было. справа - стало



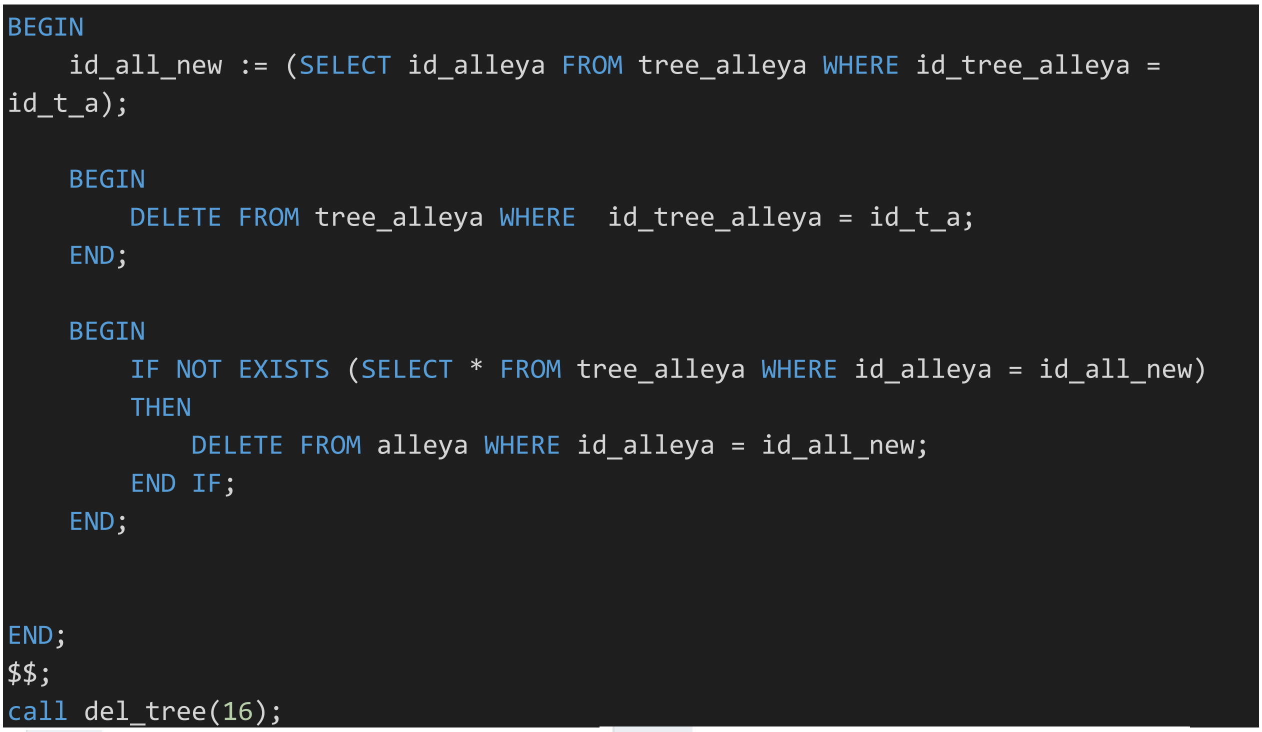
	id_tree [PK] integer	id_kind integer	plant date
1	4	3	2023-09-16
2	5	3	2023-09-17
3	6	4	2023-09-19
4	8	4	2023-10-17
5	9	4	2023-10-23
6	13	6	2023-10-05
7	16	5	2023-11-03
8	17	6	2023-11-14
9	18	3	2023-11-24
10	19	3	2023-11-14
11	24	4	2023-11-25
12	25	4	2023-11-25
13	26	4	2023-11-25
14	27	4	2023-11-25
15	28	4	2023-11-25



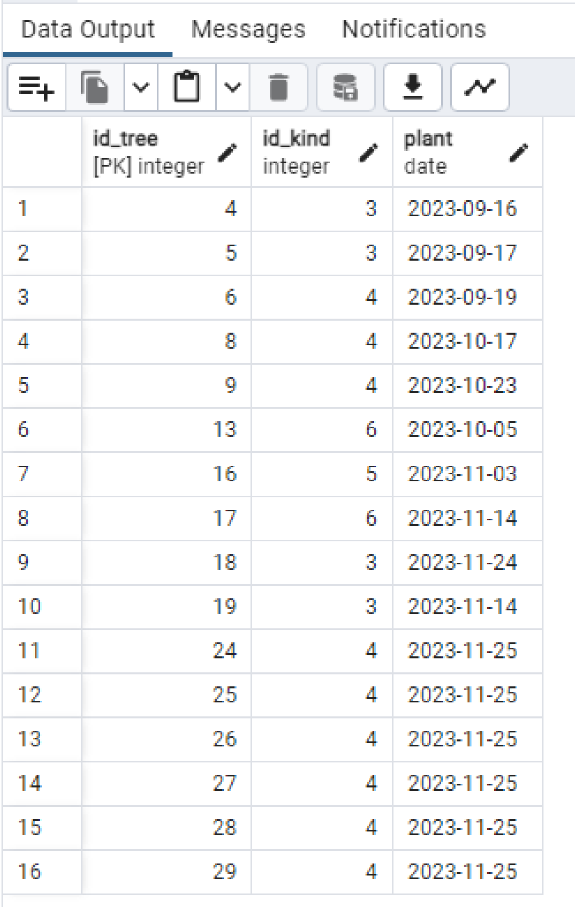
	id_tree [PK] integer	id_kind integer	plant date
1	4	3	2023-09-16
2	5	3	2023-09-17
3	6	4	2023-09-19
4	8	4	2023-10-17
5	9	4	2023-10-23
6	13	6	2023-10-05
7	16	5	2023-11-03
8	17	6	2023-11-14
9	18	3	2023-11-24
10	19	3	2023-11-14
11	24	4	2023-11-25
12	25	4	2023-11-25
13	26	4	2023-11-25
14	27	4	2023-11-25
15	28	4	2023-11-25
16	29	4	2023-11-25

--удаление с очисткой справочника

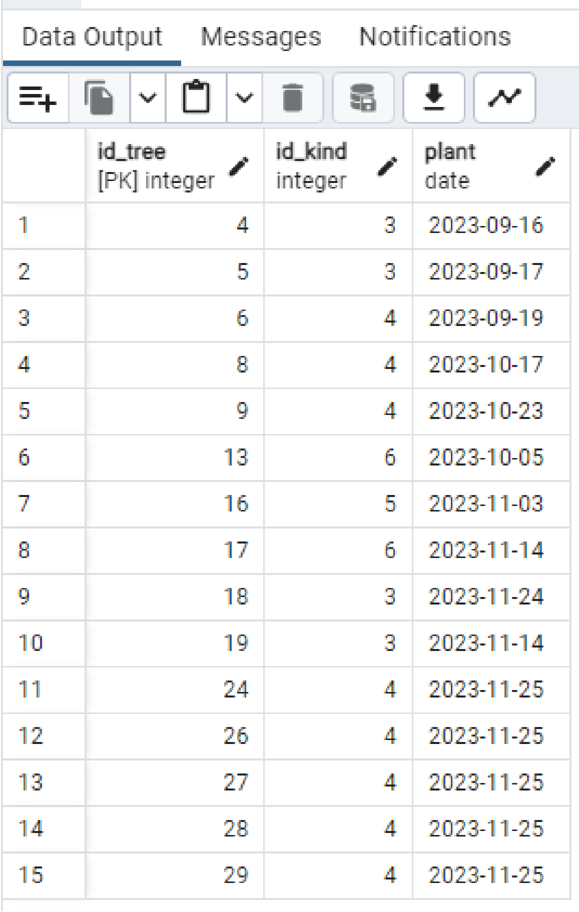
```
CREATE OR REPLACE PROCEDURE del_tree(id_t_a INT)
LANGUAGE plpgsql
AS $$
DECLARE
    id_all_new INT;
```



```
BEGIN
    id_all_new := (SELECT id_alleya FROM tree_alleya WHERE id_tree_alleya = 
id_t_a);

    BEGIN
        DELETE FROM tree_alleya WHERE  id_tree_alleya = id_t_a;
    END;

    BEGIN
        IF NOT EXISTS (SELECT * FROM tree_alleya WHERE id_alleya = id_all_new)
        THEN
            DELETE FROM alleya WHERE id_alleya = id_all_new;
        END IF;
    END;


END;
$$;
call del_tree(16);
```



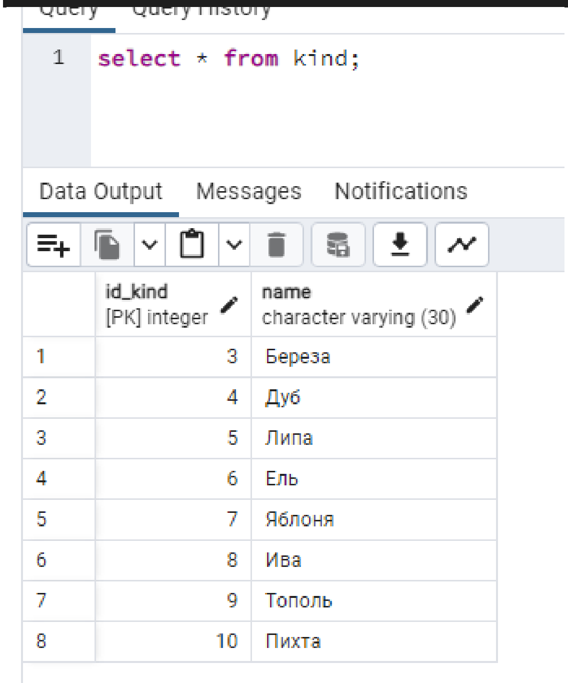
	id_tree [PK] integer	id_kind integer	plant date
1	4	3	2023-09-16
2	5	3	2023-09-17
3	6	4	2023-09-19
4	8	4	2023-10-17
5	9	4	2023-10-23
6	13	6	2023-10-05
7	16	5	2023-11-03
8	17	6	2023-11-14
9	18	3	2023-11-24
10	19	3	2023-11-14
11	24	4	2023-11-25
12	25	4	2023-11-25
13	26	4	2023-11-25
14	27	4	2023-11-25
15	28	4	2023-11-25
16	29	4	2023-11-25



	id_tree [PK] integer	id_kind integer	plant date
1	4	3	2023-09-16
2	5	3	2023-09-17
3	6	4	2023-09-19
4	8	4	2023-10-17
5	9	4	2023-10-23
6	13	6	2023-10-05
7	16	5	2023-11-03
8	17	6	2023-11-14
9	18	3	2023-11-24
10	19	3	2023-11-14
11	24	4	2023-11-25
12	26	4	2023-11-25
13	27	4	2023-11-25
14	28	4	2023-11-25
15	29	4	2023-11-25

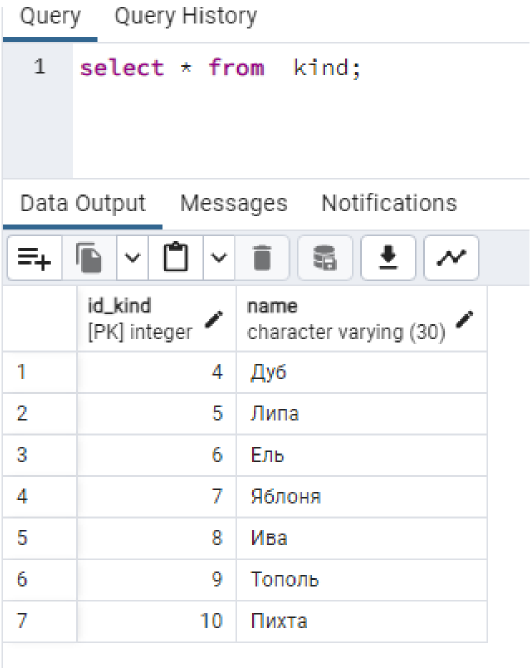
--Каскадное удаление

```
CREATE OR REPLACE PROCEDURE del_kind_cascade(id_to_del INT)
LANGUAGE plpgsql
AS $$
BEGIN
    DELETE FROM tree_alleya WHERE tree_alleya.id_tree IN
        (SELECT tree.id_tree FROM tree WHERE id_kind = id_to_del);

    DELETE FROM cut WHERE cut.id_tree IN
        (SELECT tree.id_tree FROM tree WHERE tree.id_kind = id_to_del);

    DELETE FROM tree WHERE tree.id_kind = id_to_del;
    DELETE FROM kind WHERE kind.id_kind = id_to_del;
END;
$$;
```



```
select * from kind;
```

	id_kind [PK] integer	name character varying (30)
1	3	Береза
2	4	Дуб
3	5	Липа
4	6	Ель
5	7	Яблоня
6	8	Ива
7	9	Тополь
8	10	Пихта



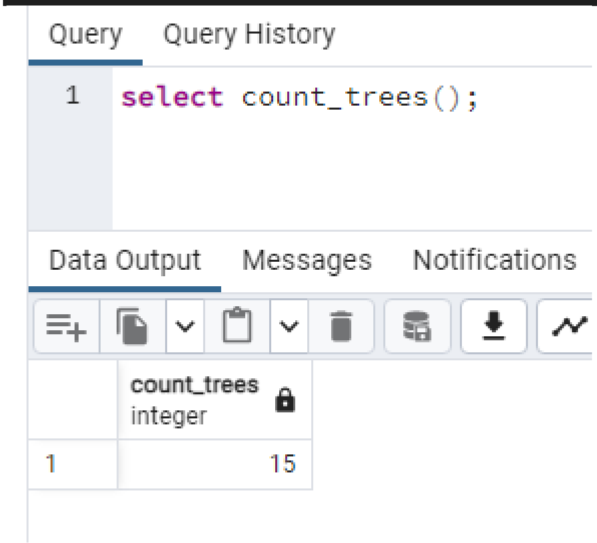
```
select * from  kind;
```

	id_kind [PK] integer	name character varying (30)
1	4	Дуб
2	5	Липа
3	6	Ель
4	7	Яблоня
5	8	Ива
6	9	Тополь
7	10	Пихта

--Процедура вычисления и возврат значения агрегатной функции

```
CREATE OR REPLACE FUNCTION count_trees() RETURNS INTEGER
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN NULLIF(COUNT(id_tree), 0) FROM tree;
END;
$$;
```

```
SELECT count_trees();
```

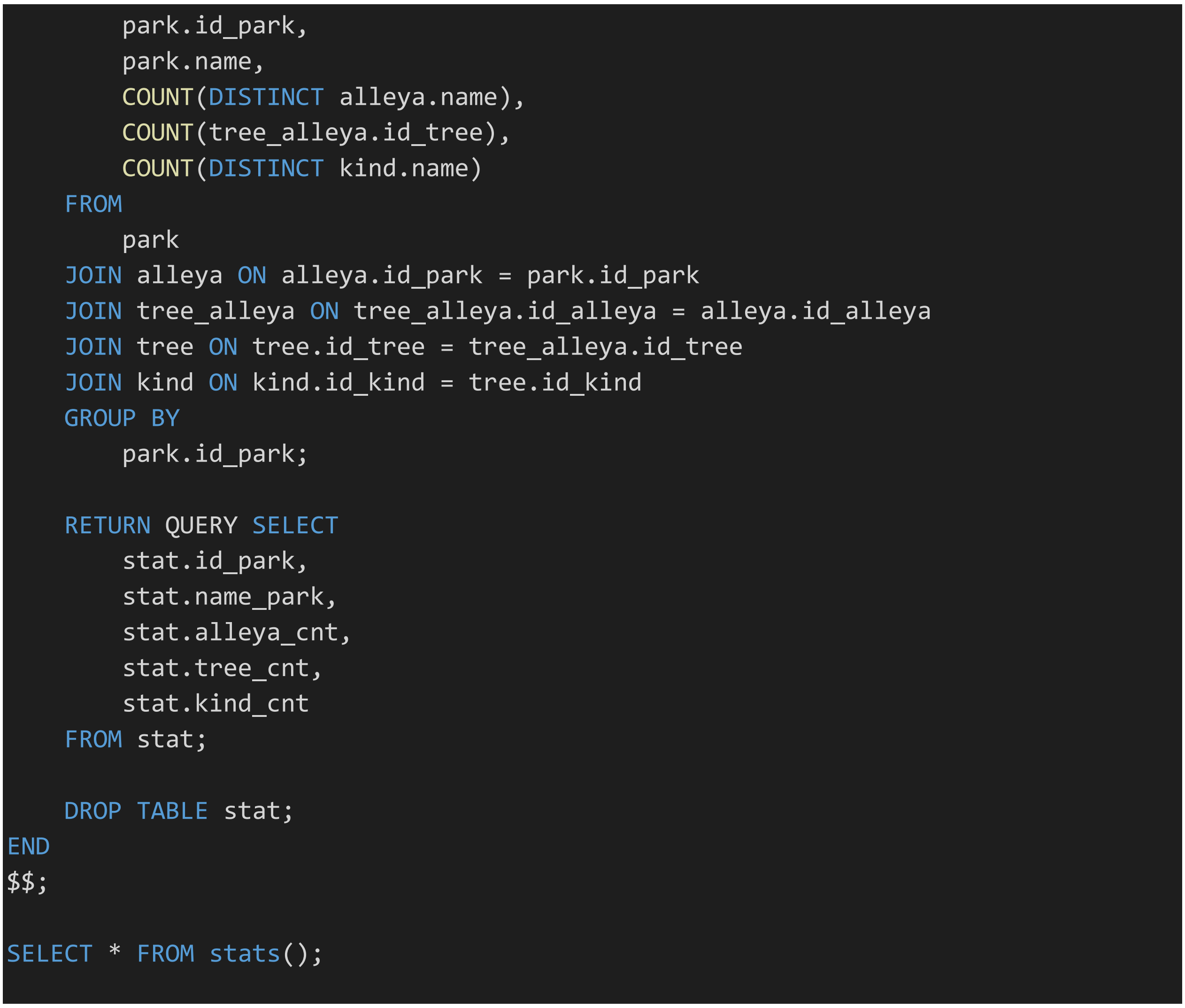


```
select count_trees();
```

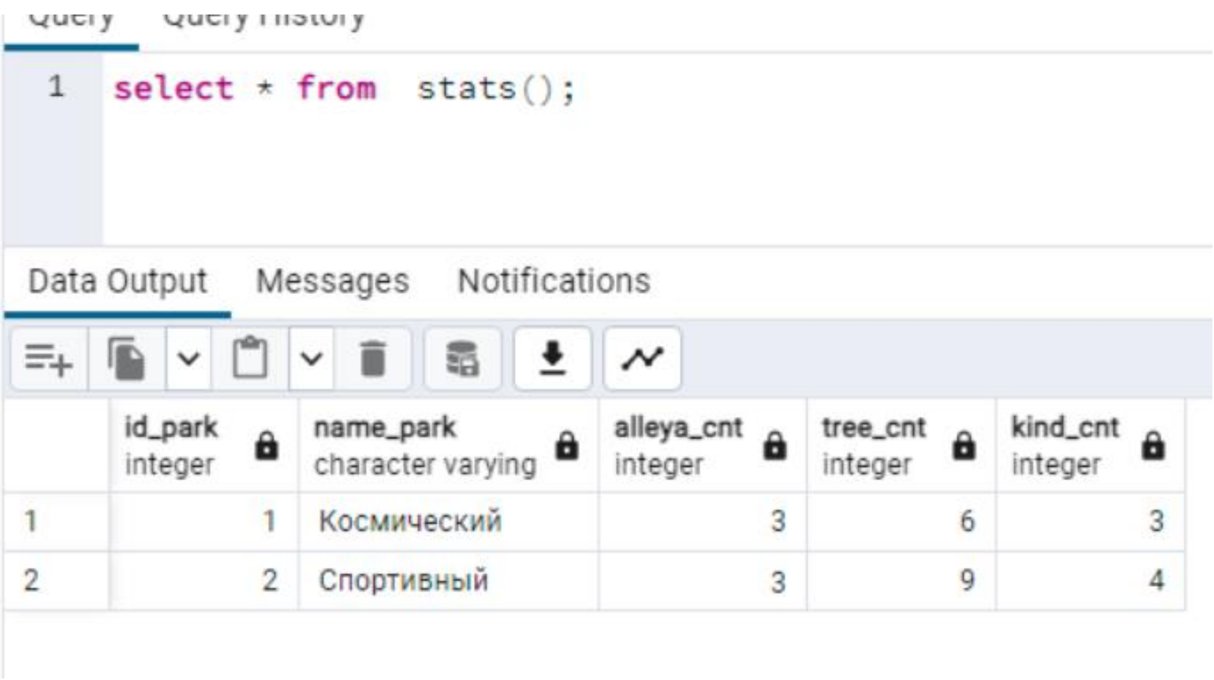
```
   count_trees
   integer
1           15
```

--Формирование статистики во временной таблице

```
CREATE OR REPLACE FUNCTION stats()
RETURNS TABLE (
    id_park INTEGER,
    name_park VARCHAR(128),
    alleya_cnt INTEGER,
    tree_cnt INTEGER,
    kind_cnt INTEGER
)
LANGUAGE plpgsql
AS $$
BEGIN
    CREATE TEMPORARY TABLE stat (
        id_park INTEGER,
        name_park VARCHAR(128),
        alleya_cnt INTEGER,
        tree_cnt INTEGER,
        kind_cnt INTEGER
    );

    INSERT INTO stat (id_park, name_park, alleya_cnt, tree_cnt, kind_cnt)
    SELECT
```



```
        park.id_park,
        park.name,
        COUNT(DISTINCT alleya.name),
        COUNT(tree_alleya.id_tree),
        COUNT(DISTINCT kind.name)
    FROM
        park
    JOIN alleya ON alleya.id_park = park.id_park
    JOIN tree_alleya ON tree_alleya.id_alleya = alleya.id_alleya
    JOIN tree ON tree.id_tree = tree_alleya.id_tree
    JOIN kind ON kind.id_kind = tree.id_kind
    GROUP BY
        park.id_park;

    RETURN QUERY SELECT
        stat.id_park,
        stat.name_park,
        stat.alleya_cnt,
        stat.tree_cnt,
        stat.kind_cnt
    FROM stat;

    DROP TABLE stat;
END
$$;

SELECT * FROM stats();
```



```
select * from  stats();
```

	id_park integer	name_park character varying	alleya_cnt integer	tree_cnt integer	kind_cnt integer
1	1	Космический	3	6	3
2	2	Спортивный	3	9	4