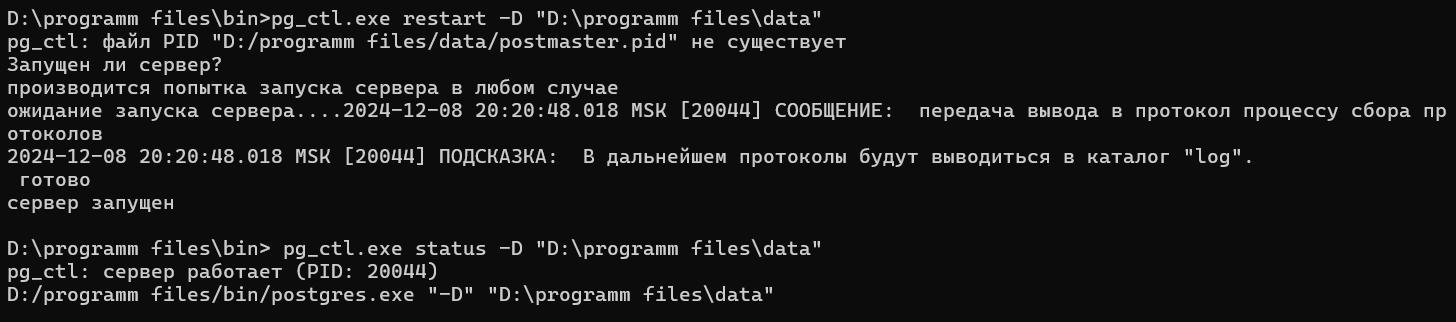
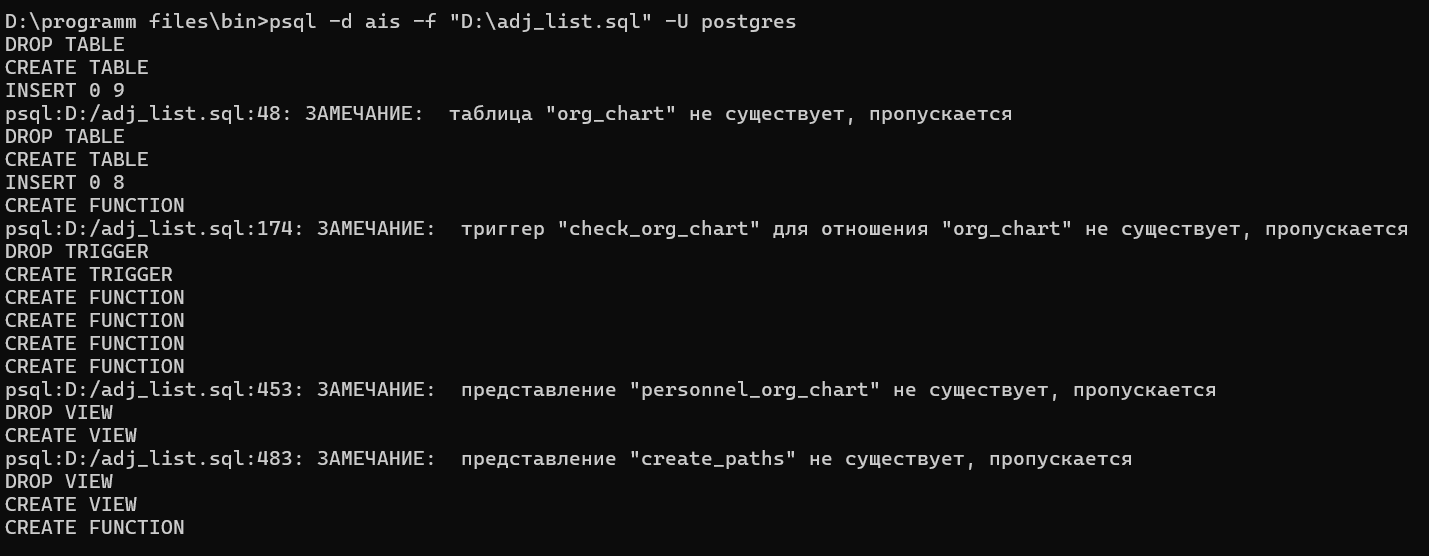
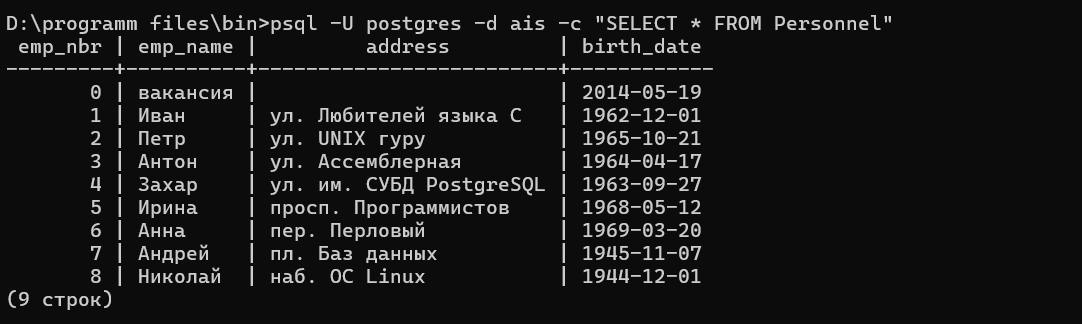
## 

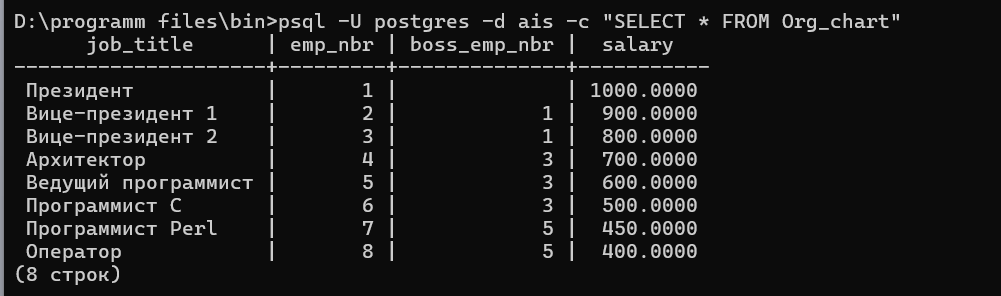
## Запускаем сервер

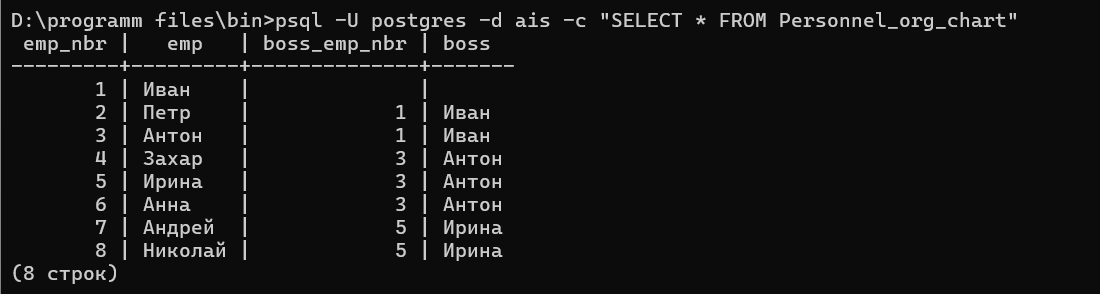


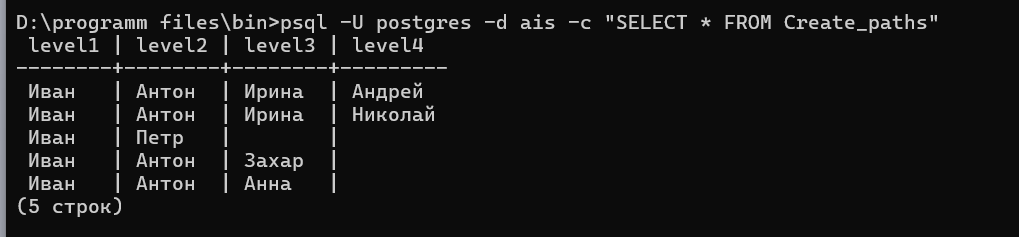


## задание 12

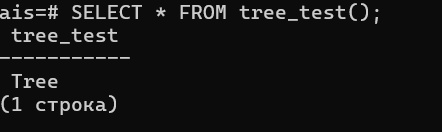




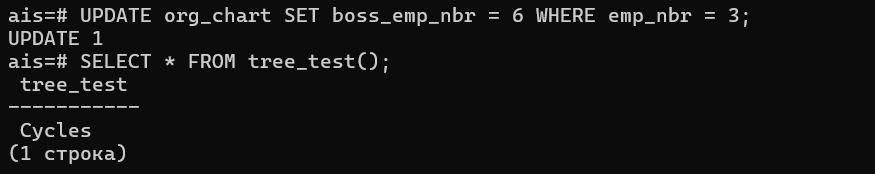




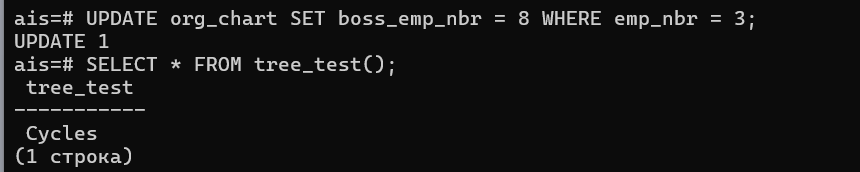
## задание 13



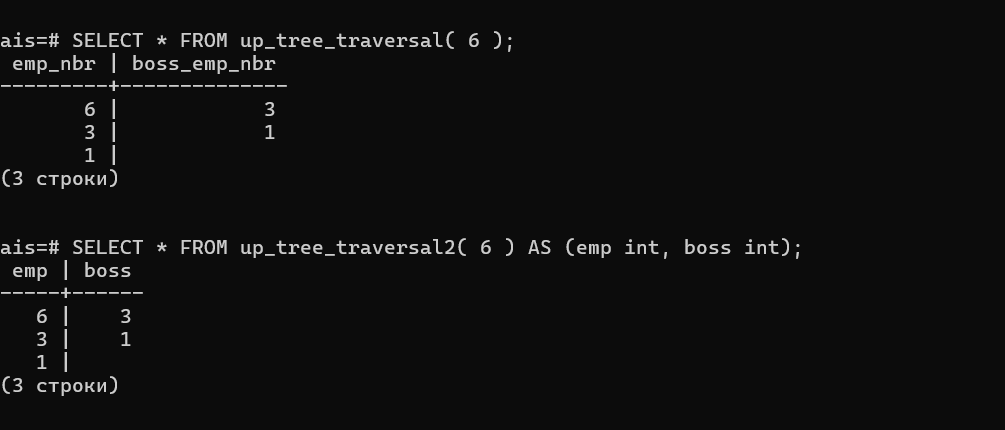
короткий



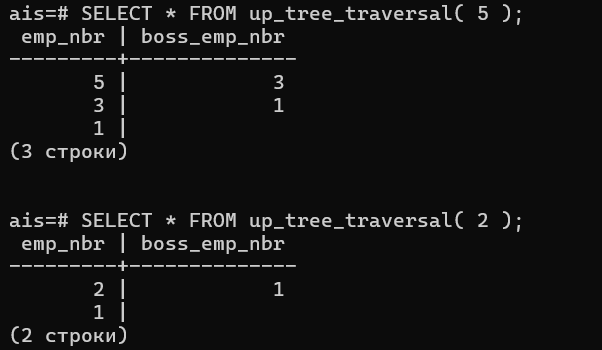
длинный



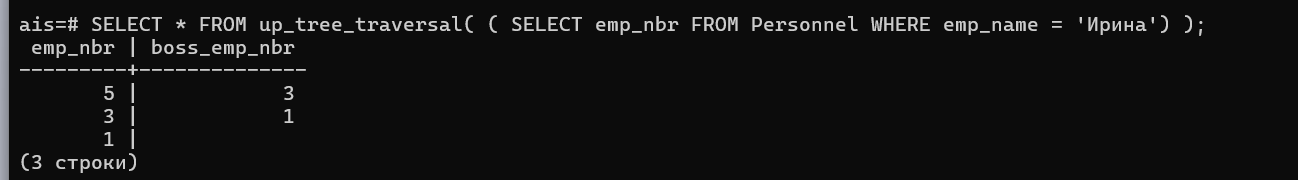
## задание 14



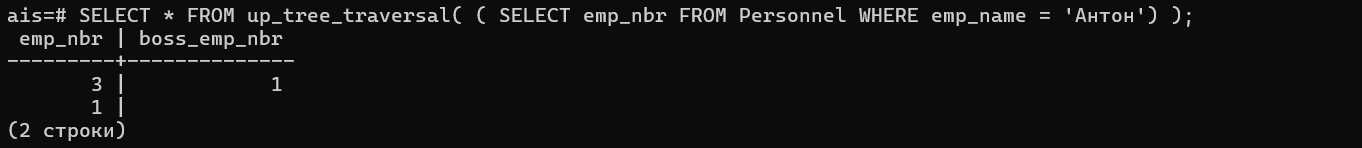
меняем номер



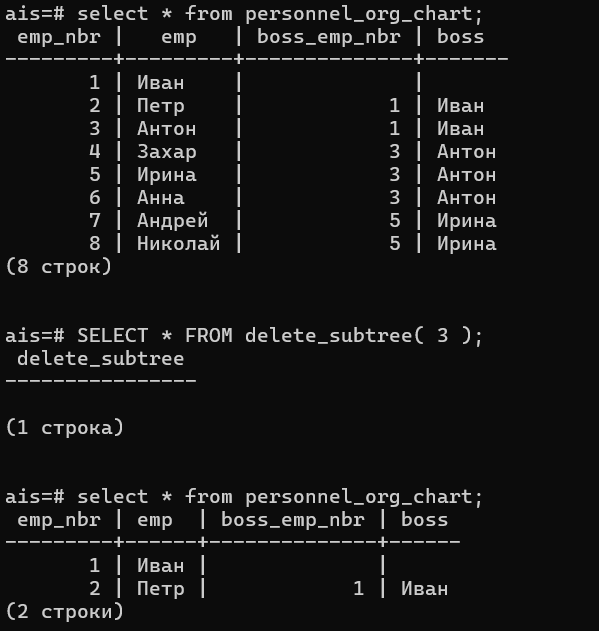
5- Ирина



3 - Антон

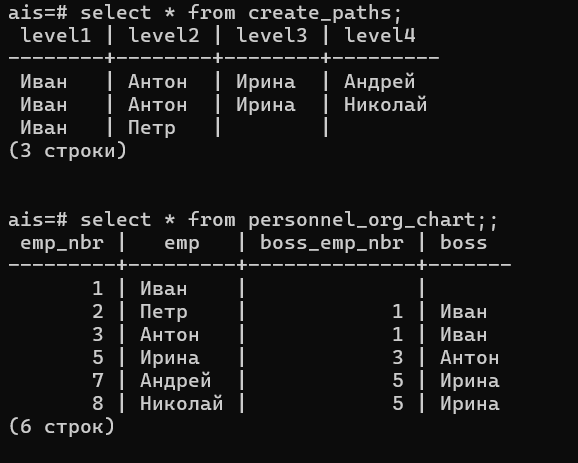


## Задание 15

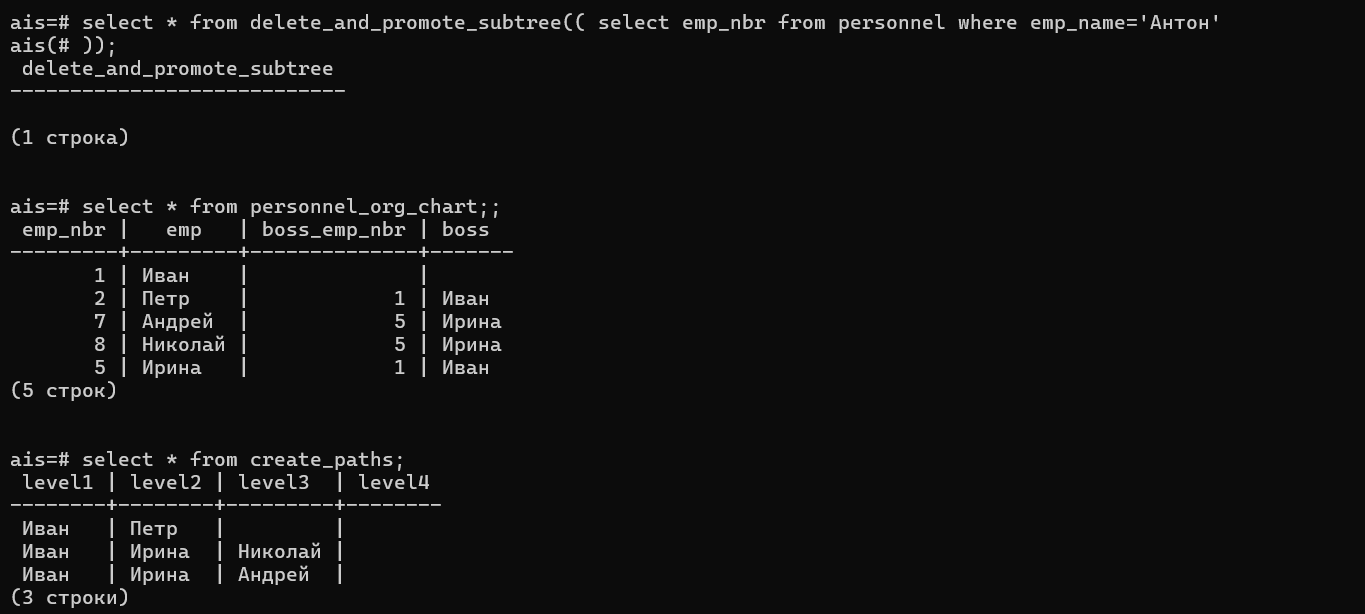


## Задание 16

до:



после:



## 

## Задание 17

Создаем новое представление:

CREATE VIEW Create\_paths ( level1, level2, level3, level4, level5 ) AS

SELECT O1.emp AS e1, O2.emp AS e2, O3.emp AS e3, O4.emp AS e4, O5.emp AS e5

FROM Personnel\_org\_chart AS O1

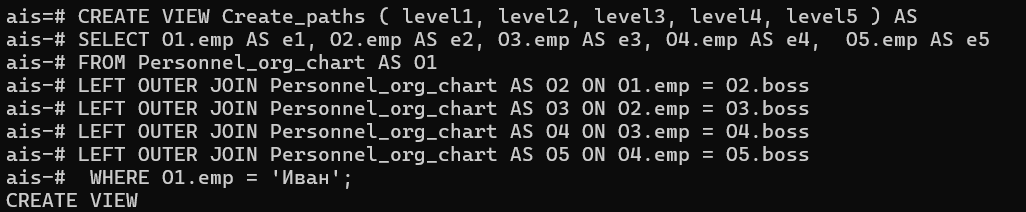
LEFT OUTER JOIN Personnel\_org\_chart AS O2 ON O1.emp = O2.boss

LEFT OUTER JOIN Personnel\_org\_chart AS O3 ON O2.emp = O3.boss

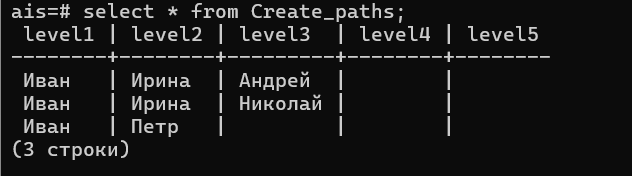
LEFT OUTER JOIN Personnel\_org\_chart AS O4 ON O3.emp = O4.boss

LEFT OUTER JOIN Personnel\_org\_chart AS O5 ON O4.emp = O5.boss

WHERE O1.emp = 'Иван';



Результат:



## 

## Задание 18

создаем функцию увеличивающую зп на определенный процент.

CREATE OR REPLACE FUNCTION increase\_salary(p\_percentage NUMERIC)

RETURNS INTEGER AS $$

DECLARE

s\_cursor CURSOR FOR

SELECT job\_title, salary FROM org\_chart WHERE salary IS NOT NULL;

v\_job\_title org\_chart.job\_title%TYPE;

v\_salary org\_chart .salary %TYPE;

v\_new\_salary org\_chart .salary %TYPE;

v\_count INTEGER := 0;

BEGIN

OPEN s\_cursor;

LOOP

FETCH s\_cursor INTO v\_job\_title , v\_salary ;

EXIT WHEN NOT FOUND;

v\_new\_salary := v\_salary + (v\_salary \* p\_percentage / 100);

UPDATE org\_chart

SET salary= v\_new\_salary

WHERE job\_title= v\_job\_title;

v\_count := v\_count + 1;

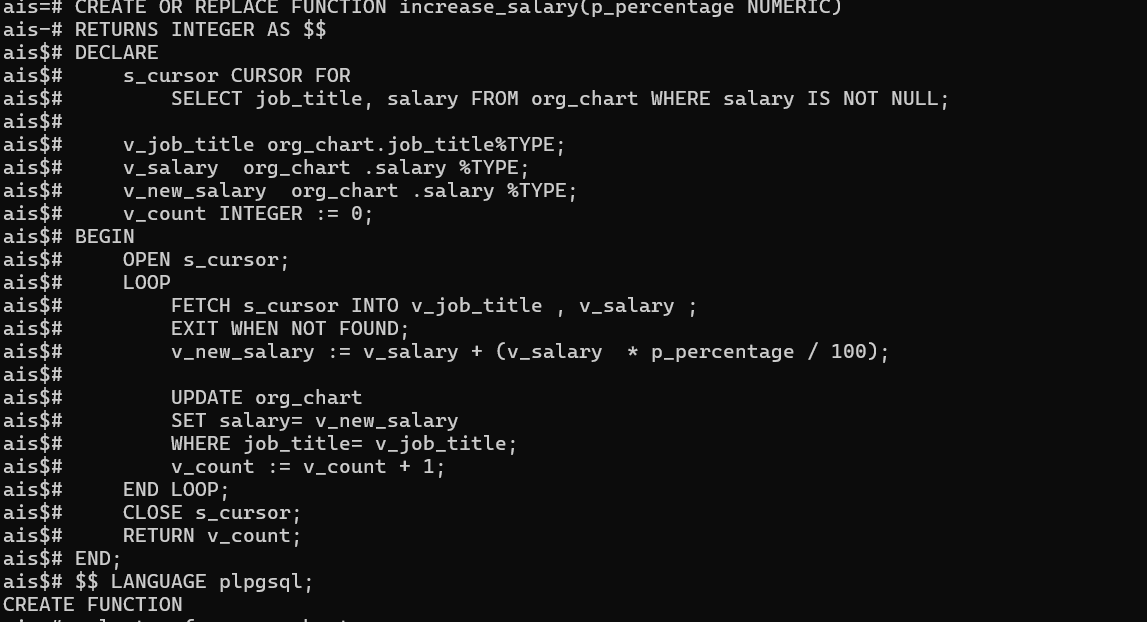
END LOOP;

CLOSE s\_cursor;

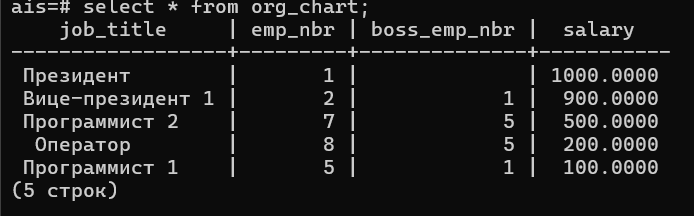
RETURN v\_count;

END;

$$ LANGUAGE plpgsql;



до запуска:



запускаем функцию, проверяем, что получилось.

