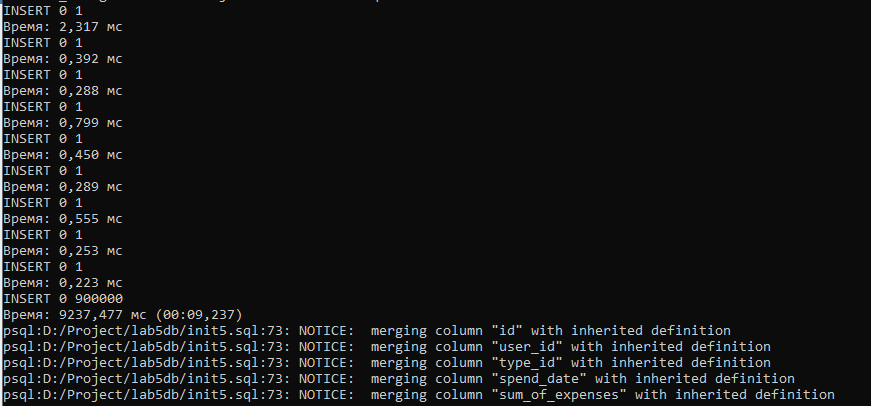
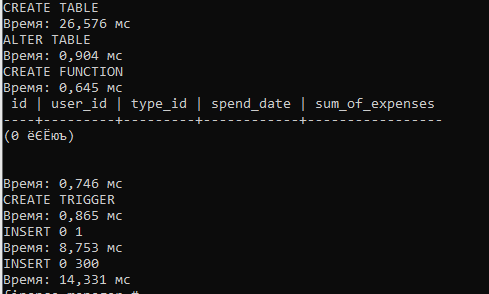
**Задание 1.** Для Вашей представленной базы данных выбрать 1 или несколько таблиц для секционирования. Обосновать выбор. Выбрать ключ и выполнить разделение на секции.

CREATE TABLE expenses\_2020\_11 (like expenses including all) INHERITS (expenses);

ALTER TABLE expenses\_2020\_11 add check ( spend\_date between date'2020-11-01' and date'2020-12-20'-1);  
  
Выберем таблицу expenses, т.к. её можно секционировать по датам для более удобной навигации и просмотра расходов в заданном промежутке. Т.к. в основном работа будет вестись с датами и промежутками времени за которые пользователь совершил ряд расходов, то наиболее важное для удобства и быстроты будет секционирование данной таблицы.

**Задание 2.** Написать триггер, который будет вставлять новые записи в соответствующие секции. Продемонстрировать работу секционирования, используя генерацию тестовых данных.





CREATE OR REPLACE FUNCTION expenses\_select\_part()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.spend\_date between date'2020-11-01' and date'2020-12-20'-1 THEN

INSERT INTO expenses\_2020\_11 VALUES (NEW.\*);

else

RAISE EXCEPTION 'this date not in you partitions %',NEW.id;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

SELECT \* from expenses\_2020\_11;

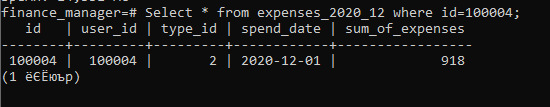
CREATE TRIGGER check\_date\_expenses

BEFORE INSERT ON expenses

FOR EACH ROW EXECUTE PROCEDURE expenses\_select\_part();  
INSERT INTO expenses(id,type\_id,spend\_date,sum\_of\_expenses,user\_id)values(1001,2,'2020-12-05',1123,100001);

**Задание 3.** Написать несколько запросов  SELECT для демонстрации выборки данных из Вашей секционированной таблицы.

Select \* from expenses\_2020\_12 where id=100004;



Select \* from expenses\_2020\_12 where spend\_date between ‘2020-12-01’ and ‘2020-12-06’;

