



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ

по лабораторной работе № 3

Название: Создание БД для аналитики

Дисциплина: Базы данных

Студент

ИУ6-33Б
(Группа)

19.11.2023

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

М.А. Скворцова

(И.О. Фамилия)

Москва, 2023

Цель

Сформировать у студента понимание особенностей создания аналитических баз данных и умение их настраивать и поддерживать.

Ход работы

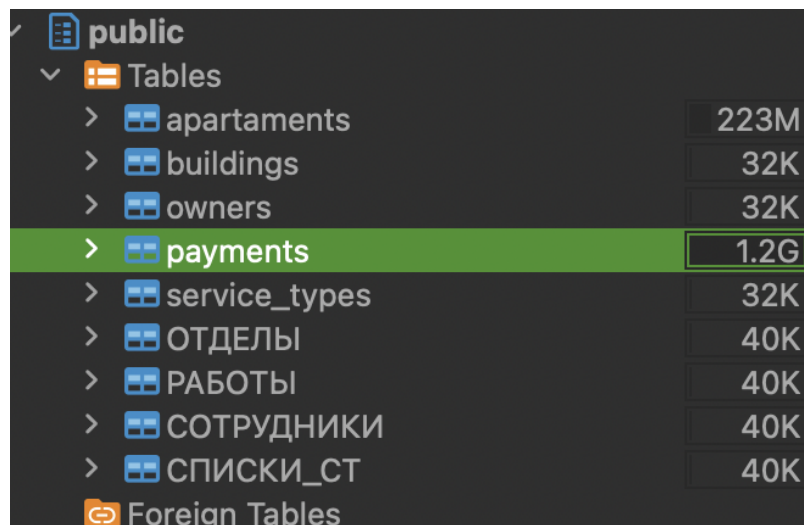
Часть 1. Проектирование схемы базы данных

Изменим схему путем денормализации добавления jsonb атрибутов и массивов.

```
1  CREATE TABLE IF NOT EXISTS public.buildings (  
2      id uuid PRIMARY KEY,  
3      street varchar NOT NULL,  
4      number integer NOT NULL  
5  );  
6  
7  CREATE TABLE IF NOT EXISTS public.owners (  
8      id uuid PRIMARY KEY,  
9      name varchar NOT NULL,  
10     last_name varchar NOT NULL  
11 );  
12  
13 CREATE TABLE IF NOT EXISTS public.apartaments (  
14     id uuid PRIMARY KEY,  
15     number integer NOT NULL,  
16     full_address jsonb NOT NULL,  
17     building_id uuid NOT NULL REFERENCES buildings ON DELETE CASCADE,  
18     owner_id uuid NOT NULL REFERENCES owners ON DELETE CASCADE  
19 );  
20  
21 CREATE TABLE IF NOT EXISTS public.service_types (  
22     id uuid PRIMARY KEY,  
23     description varchar NOT NULL,  
24     has_counter bool NOT NULL,  
25     price_per_month integer NOT NULL  
26 );  
27  
28 CREATE TABLE IF NOT EXISTS public.payments (  
29     id uuid PRIMARY KEY,  
30     period date NOT NULL,  
31     payment_date date NOT NULL,  
32     price integer NOT NULL,  
33     service_type_id uuid[] NOT NULL,  
34     apartament_id uuid NOT NULL REFERENCES apartaments ON DELETE CASCADE  
35 );
```

Рисунок 1. Обновленная схема для БД

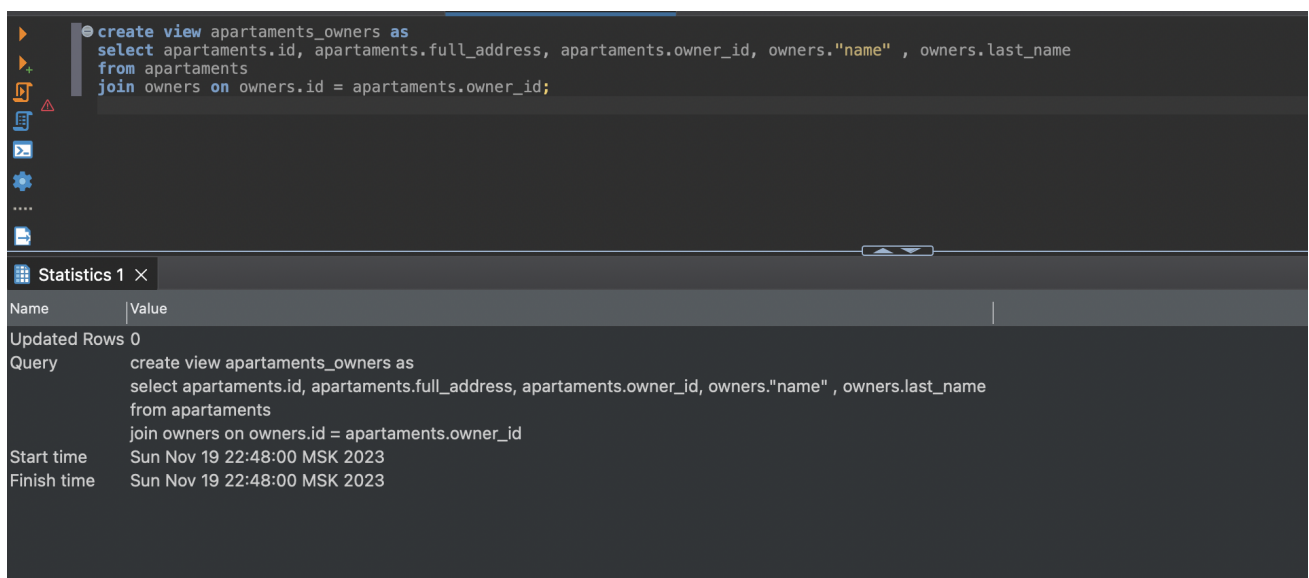
Заполним БД с более 6 миллионами записей.



public	
Tables	
> apartments	223M
> buildings	32K
> owners	32K
> payments	1.2G
> service_types	32K
> ОТДЕЛЫ	40K
> РАБОТЫ	40K
> СОТРУДНИКИ	40K
> СПИСКИ_СТ	40K
Foreign Tables	

Рисунок 2. Вес БД с 6kk+ записями

Часть 2. Создание и заполнения таблиц



```
create view apartaments_owners as
select apartaments.id, apartaments.full_address, apartaments.owner_id, owners."name" , owners.last_name
from apartaments
join owners on owners.id = apartaments.owner_id;
```

Name	Value
Updated Rows	0
Query	create view apartaments_owners as select apartaments.id, apartaments.full_address, apartaments.owner_id, owners."name" , owners.last_name from apartaments join owners on owners.id = apartaments.owner_id
Start time	Sun Nov 19 22:48:00 MSK 2023
Finish time	Sun Nov 19 22:48:00 MSK 2023

Рисунок 3. Создание view apartaments_owners

Создадим пользователя test и выдадим ему различные роли

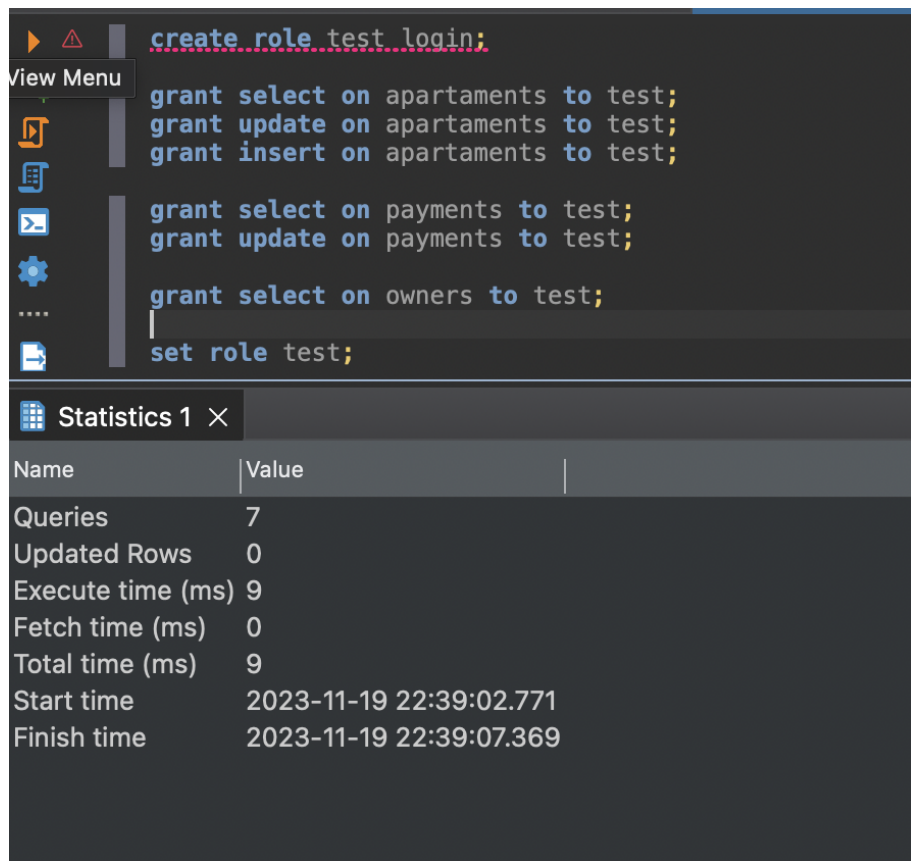


Рисунок 3. Создание роли test

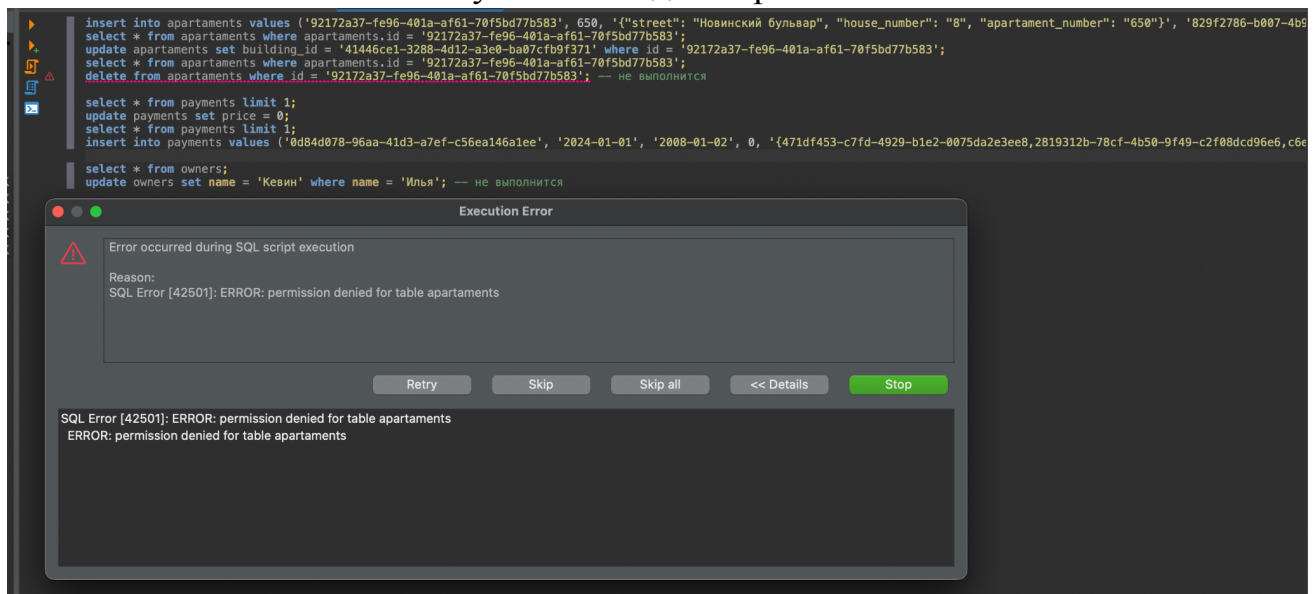


Рисунок 4. В соответствии с ролями, этот запрос не выполнился

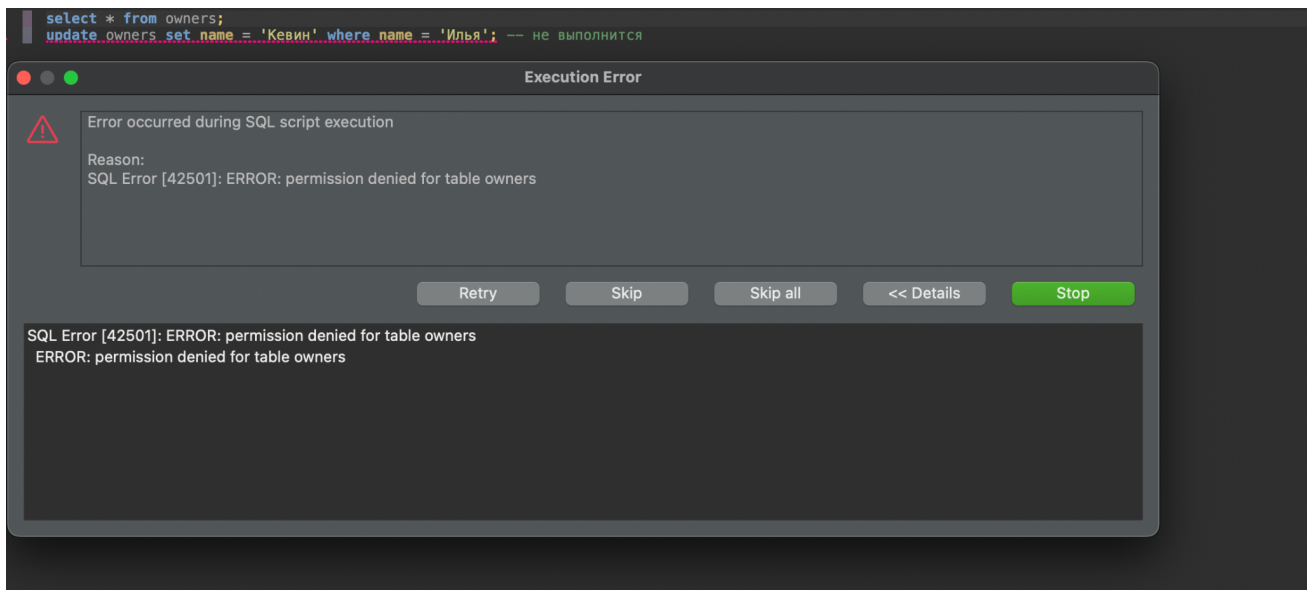


Рисунок 5. В соответствии с ролями, этот запрос также не выполнен

Вывод

Было сформировано понимание особенностей создания аналитических баз данных и умение их настраивать и поддерживать.

Вопросы

1. Для чего нужна денормализация?
 - Улучшение производительности баз данных, снижение сложности запросов.
2. Виды денормализации:
 - Простая, атрибутов, иерархии, кластера, использование материализованных представлений.
3. Методы реализации денормализации:
 - Добавление повторяющихся данных, денормализация атрибутов, иерархии, кластера, использование материализованных представлений.
4. Преимущество использования массивов:
 - Эффективное хранение и обработка коллекций данных.
5. Разница между json и jsonb:
 - `json` - хранение в текстовом формате, `jsonb` - хранение в бинарном формате, более эффективен.

6. Для чего нужны роли:

- Управление правами доступа пользователей.

7. Что такое схема:

- Логическая группировка объектов базы данных.

8. Директивы GRANT и REVOKE:

- 'GRANT' - предоставление прав, 'REVOKE' - отзыв прав.

9. Роль PUBLIC:

- Роль, которая применяется ко всем пользователям по умолчанию.

10. Добавление нового пользователя:

- 'CREATE USER имя WITH PASSWORD 'пароль';'

11. Позволить пользователю заходить на сервер:

- 'ALTER USER имя WITH LOGIN;'

12. Существующие права:

- SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, и др.

13. Смена владельца базы данных:

- 'ALTER DATABASE имя OWNER TO новый_владелец;'

14. Смена пароля для пользователя:

- 'ALTER USER имя WITH PASSWORD 'новый_пароль';'

15. Определение роли с правами:

- 'GRANT права ON объект TO роль;'

16. CHECK OPTION:

- Ограничение, которое гарантирует, что данные, добавленные через представление, соответствуют определенному условию.

17. Модификация данных через представления:

- Возможность изменять данные через представление.

18. Вставка данных через представления:

- Возможность добавлять данные через представление.

19. Полезность материализованного представления:

- Хранение предварительно вычисленных результатов запросов для

улучшения производительности.