

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «базы данных»**  
**Тема: Реализация базы данных в СУБД PostgreSQL**

Студентка гр. 1304

\_\_\_\_\_

Хорошкова А.С.

Преподаватель

\_\_\_\_\_

Заславский М.М.

Санкт-Петербург

2023

## **Цель работы.**

Изучение работы с базами данных на практике, создание и редактирование таблиц.

## **Текст задания (Вариант 1).**

Необходимо развернуть PostgreSQL локально:

1. Написать запросы для создания таблиц из предыдущей лабораторной работы
2. Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
3. Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
4. Исходный код выложить на [www.db-fiddle.com](http://www.db-fiddle.com) для проверки работоспособности
5. Исходный код в виде .sql файла запустить в виде PR в репо
6. В отчете описать:

- Цель
- Текст задания в соответствии с вариантом
- Скриншоты работы с СУБД PostgreSQL ([psql](#) / [DBeaver](#) / [Datagrip](#), ...)
- Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
- Исходный код в приложении
- Ссылку на исходный код [www.db-fiddle.com](http://www.db-fiddle.com) в приложении
- Ссылка на PR в приложении
- Вывод

Текст варианта работы:

Пусть требуется создать программную систему, предназначенную для администратора гостиницы. Такая система должна обеспечивать хранение сведений об имеющихся в гостинице номерах, о проживающих в гостинице клиентах и о служащих, убирающих в номерах. Пусть количество номеров в гостинице известно, и имеются номера трех типов: одноместный, двухместный

и трехместный, отличающиеся стоимостью проживания в сутки. В каждом номере есть телефон. О каждом проживающем должна храниться следующая информация: номер паспорта, фамилия, имя, отчество, город, из которого он прибыл, дата поселения в гостинице, выделенный гостиничный номер. О служащих гостиницы должна храниться информация следующего содержания: фамилия, имя, отчество, где (этаж) и когда (день недели) он убирает. Служащий гостиницы убирает все номера на одном этаже в определенные дни недели, при этом в разные дни он может убирать разные этажи. Работа с системой предполагает получение следующей информации:

- о клиентах, проживающих в заданном номере,
- о клиентах, прибывших из заданного города,
- о том, кто из служащих убирал номер указанного клиента в заданный день недели,
- есть ли в гостинице свободные места и свободные номера и, если есть, то сколько.

Администратор должен иметь возможность выполнить следующие операции:

- принять на работу или уволить служащего гостиницы.
- изменить расписание работы служащего.
- поселить или выселить клиента.

### **Ход работы.**

Для выполнения работы был использован pgAdmin - графический клиент для управления базами данных PostgreSQL.

1. Были написаны запросы для создания таблиц из предыдущей лабораторной работы.

Пример создания первых трёх таблиц приведён на Рисунок 1. Аналогично были созданы остальные таблицы.

```

hotel_db/postgres@PostgreSQL 16 (64bit)
No limit
Query Query History
1 CREATE TABLE IF NOT EXISTS cleaner
2 (
3     id_cleaner integer NOT NULL,
4     name text NOT NULL,
5     second_name text,
6     patronymic text,
7     CONSTRAINT cleaner_pkey PRIMARY KEY (id_cleaner)
8 );
9
10 CREATE TABLE IF NOT EXISTS cleaning
11 (
12     week_day text COLLATE pg_catalog."default" NOT NULL,
13     floor integer NOT NULL,
14     id_cleaner integer NOT NULL,
15     CONSTRAINT cleaning_pkey PRIMARY KEY (week_day, floor),
16     CONSTRAINT cleaning_id_cleaner_fkey FOREIGN KEY (id_cleaner)
17         REFERENCES cleaner (id_cleaner)
18 );
19
20 CREATE TABLE IF NOT EXISTS room_type
21 (
22     bed_amount integer NOT NULL,
23     price numeric NOT NULL,
24     CONSTRAINT room_type_pkey PRIMARY KEY (bed_amount)
25 );

```

Рисунок 1 — пример создания таблиц в pgAdmin

На Рисунок 2 представлены сгенерированные с помощью pgAdmin диаграммы созданных таблиц.

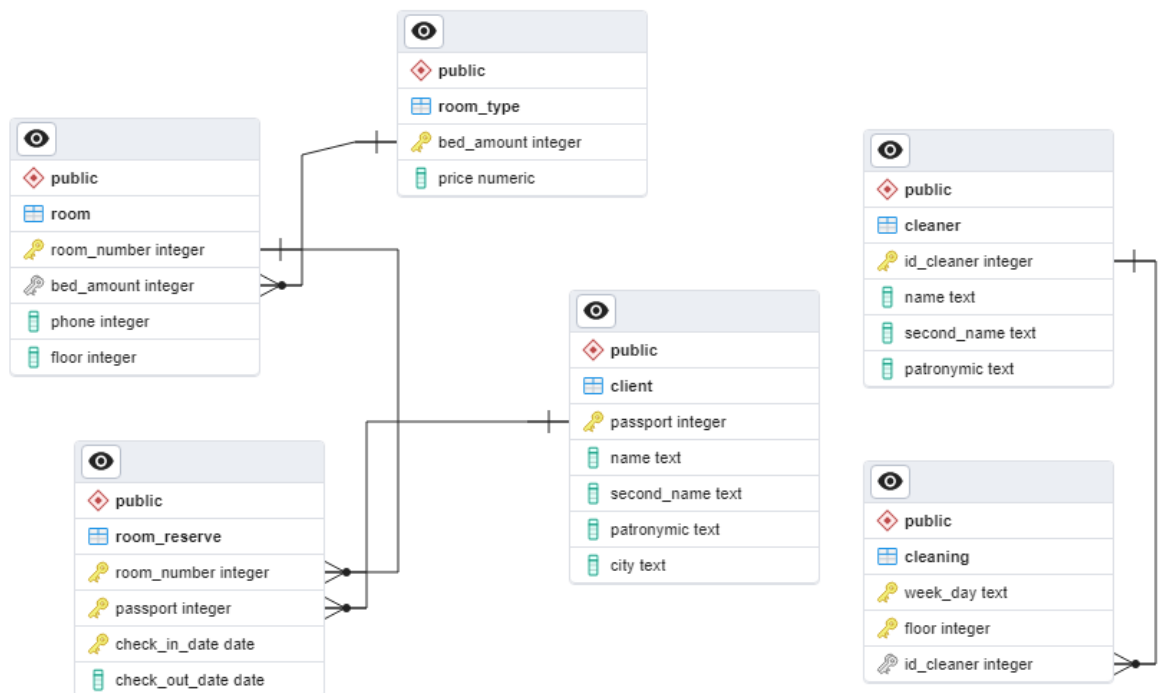


Рисунок 2 — диаграмма созданных таблиц

2. Были заполнены тестовыми данными все таблицы. Ниже представлено наполнение таблиц.

	passport [PK] integer	name text	second_name text	patronymic text	city text
1	1234123456	Саша	Привет	Викторовна	Санкт-Петербург
2	1234123457	Маша	Пока	Сергеевна	Тихвин
3	1234123458	Паша	Здравствуй	[null]	Санкт-Петербург
4	1234123459	Даша	Добрый	[null]	Санкт-Петербург
5	1234123460	Саша	Злой	[null]	Екатеринбург
6	1234123461	Александр	Нормальный	[null]	Москва
7	1234123462	Виктория	Мать	Сергеевна	Курск
8	1234123463	Вадик	Немать	Сергеевич	Санкт-Петербург

Рисунок 3 — заполненная таблица client

	bed_amount [PK] integer	price numeric
1	1	1000.00
2	2	1500.00
3	3	2500.00
4	4	3500.00
5	5	5000.00

Рисунок 4 — заполненная таблица room\_type

	room_number [PK] integer	bed_amount integer	phone integer	floor integer
1	12	1	8945647	2
2	13	2	8955647	1
3	176	1	4945647	3
4	14	3	8954647	1
5	121	1	8945697	2
6	11	2	8995647	3

Рисунок 5 — заполненная таблица room

	room_number [PK] integer	passport [PK] integer	check_in_date [PK] date	check_out_date date
1	13	1234123458	2023-03-22	2023-03-28
2	176	1234123460	2023-03-22	2023-03-26
3	13	1234123462	2023-03-22	[null]
4	14	1234123461	2023-03-22	[null]
5	12	1234123456	2023-03-22	2023-10-24
6	12	1234123470	2023-10-24	[null]

Рисунок 6 — заполненная таблица room\_reserve

	id_cleaner [PK] integer	name text	second_name text	patronymic text
1	1	Виталий	Сергеевич	Викторовна
2	2	Арина	Арина	Васильевич
3	3	Паша	Петров	[null]
4	4	Даша	Фёдоров	[null]
5	5	Саша	Иванов	[null]
6	6	Александр	Коробкин	[null]
7	7	Виктория	Крутой	Николаевич

Рисунок 7 — заполненная таблица cleaner

	week_day [PK] text	floor [PK] integer	id_cleaner integer
1	Mon	2	1
2	Wed	1	2
3	Sun	3	1
4	Sat	4	3
5	Fri	3	4
6	Thu	2	5

Рисунок 8 — заполненная таблица cleaning

3. Были написаны примеры запросов на получение следующей информации:

- о клиентах, проживающих в заданном номере.

	passport [PK] integer	name text	second_name text	patronymic text	city text
1	1234123456	Саша	Привет	Викторовна	Санкт-Петербург

Рисунок 9 — клиенты, проживающие в номере 12

- о клиентах, прибывших из заданного города.

	passport [PK] integer	name text	second_name text	patronymic text	city text
1	1234123456	Саша	Привет	Викторовна	Санкт-Петербург
2	1234123458	Паша	Здравствуй	[null]	Санкт-Петербург
3	1234123459	Даша	Добрый	[null]	Санкт-Петербург
4	1234123463	Вадик	Немать	Сергеевич	Санкт-Петербург

Рисунок 10 — клиенты, прибывшие из Санкт-Петербурга

- о том, кто из служащих убирал номер указанного клиента в заданный день недели.

	name text	room_number integer	week_day text	id_cleaner integer	name text
1	Саша	12	Mon	1	Виталий

Рисунок 11 — работник, убирающий номер клиента Саша в понедельник

- есть ли в гостинице свободные места и свободные номера и, если есть, то сколько.

	amount bigint
1	4

Рисунок 12 — количество свободных номеров в гостинице

4. Были написаны запросы для выполнения следующих операций:

- принять на работу или уволить служащего гостиницы.

	id_cleaner [PK] integer	name text	second_name text	patronymic text
1	1	Виталий	Сергеевич	Викторовна
2	2	Арина	Арина	Васильевич
3	3	Паша	Петров	[null]
4	4	Даша	Фёдоров	[null]
5	5	Саша	Иванов	[null]
6	6	Александр	Коробкин	[null]
7	7	Виктория	Крутой	Николаевич
8	8	Василий	Кружка	Сергеевич
9	10	Вадим	Иванов	Иванович

Рисунок 13 — добавление нового сотрудника Вадима Иванова

После увольнения нового сотрудника Вадима Иванова таблица находилась в состоянии как на Рисунок 7.

- изменить расписание работы служащего.

	week_day [PK] text	floor [PK] integer	id_cleaner integer
1	Wed	1	2
2	Sun	3	1
3	Sat	4	3
4	Fri	3	4
5	Thu	2	5
6	Wed	2	1

Рисунок 14 — изменение расписание второго сотрудника с понедельника на среду

- поселить или выселить клиента.

	passport [PK] integer	name text	second_name text	patronymic text	city text
1	1234123456	Саша	Привет	Викторовна	Санкт-Петербург
2	1234123457	Маша	Пока	Сергеевна	Тихвин
3	1234123458	Паша	Здравствуй	[null]	Санкт-Петербург
4	1234123459	Даша	Добрый	[null]	Санкт-Петербург
5	1234123460	Саша	Злой	[null]	Екатеринбург
6	1234123461	Александр	Нормальный	[null]	Москва
7	1234123462	Виктория	Мать	Сергеевна	Курск
8	1234123463	Вадик	Немать	Сергеевич	Санкт-Петербург
9	1234123470	Саша	Опять	Витальевна	Санкт-Петербург

Рисунок 15 — добавление нового клиента 1234123470



	room_number [PK] integer	passport [PK] integer	check_in_date [PK] date	check_out_date date
1	13	1234123458	2023-03-22	2023-03-28
2	176	1234123460	2023-03-22	2023-03-26
3	13	1234123462	2023-03-22	[null]
4	14	1234123461	2023-03-22	[null]
5	12	1234123456	2023-03-22	2023-10-24
6	12	1234123470	2023-10-24	[null]

Рисунок 16 — заселение в номер нового клиента 1234123470

5. Исходный код был выложен на <https://www.db-fiddle.com/f/NkPS6t1GpwXi5FNWNdGri/0>
6. Ссылка на PR в приложении

### **Выводы.**

В ходе работы была создана база данных с таблицами из первой лабораторной работы, наполненными различными данными. Были написаны примеры запросов к базе данных.

## **ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД**

```
CREATE TABLE IF NOT EXISTS cleaner
(
    id_cleaner integer NOT NULL,
    name text NOT NULL,
    second_name text,
    patronymic text,
    CONSTRAINT cleaner_pkey PRIMARY KEY (id_cleaner)
);

CREATE TABLE IF NOT EXISTS cleaning
(
    week_day text COLLATE pg_catalog."default" NOT NULL,
    floor integer NOT NULL,
```

```
id_cleaner integer NOT NULL,  
CONSTRAINT cleaning_pkey PRIMARY KEY (week_day, floor),  
CONSTRAINT cleaning_id_cleaner_fkey FOREIGN KEY (id_cleaner)  
    REFERENCES cleaner (id_cleaner)  
);
```

```
CREATE TABLE IF NOT EXISTS room_type  
(  
    bed_amount integer NOT NULL,  
    price numeric NOT NULL,  
    CONSTRAINT room_type_pkey PRIMARY KEY (bed_amount)  
);
```

```
CREATE TABLE IF NOT EXISTS client  
(  
    passport integer NOT NULL,  
    name text NOT NULL,  
    second_name text,  
    patronymic text,  
    city text,  
    CONSTRAINT client_pkey PRIMARY KEY (passport)  
);
```

```
CREATE TABLE IF NOT EXISTS room  
(  
    room_number integer NOT NULL,  
    bed_amount integer NOT NULL,  
    phone integer,  
    floor integer NOT NULL,  
    CONSTRAINT room_pkey PRIMARY KEY (room_number),  
    CONSTRAINT room_bed_amount_fkey FOREIGN KEY (bed_amount)  
        REFERENCES room_type (bed_amount)  
);
```

```

CREATE TABLE IF NOT EXISTS room_reserve
(
    room_number integer NOT NULL,
    passport integer NOT NULL,
    check_in_date date NOT NULL,
    check_out_date date,
    CONSTRAINT room_reserve_pkey PRIMARY KEY (room_number,
passport, check_in_date),
    CONSTRAINT room_reserve_passport_fkey FOREIGN KEY (passport)
REFERENCES client (passport),
    CONSTRAINT room_reserve_room_number_fkey FOREIGN KEY
(room_number)
REFERENCES room (room_number)
);

```

```

INSERT INTO client(passport, name, second_name, patronymic, city)
VALUES (1234123456, 'Саша','Привет', 'Викторовна','Санкт-Петербург'),
(1234123457, 'Маша','Пока', 'Сергеевна', 'Тихвин'),
(1234123458,'Паша','Здравствуй' ,null,'Санкт-Петербург'),
(1234123459, 'Даша','Добрый' ,null,'Санкт-Петербург'),
(1234123460, 'Саша','Злой' ,null,'Екатеринбург'),
(1234123461,'Александр','Нормальный' ,null,'Москва'),
(1234123462,'Виктория','Мать' , 'Сергеевна','Курск'),
(1234123463, 'Вадик','Немать' , 'Сергеевич','Санкт-Петербург');

```

```

INSERT INTO room_type.bed_amount, price)
VALUES (1, 1000.00),
(2, 1500.00),
(3, 2500.00),
(4, 3500.00),

```

(5, 5000.00);

```
INSERT INTO room (room_number, bed_amount, phone, floor)
VALUES (12, 1, 8945647, 2),
       (13, 2, 8955647, 1),
       (176, 1, 4945647, 3),
       (14, 3, 8954647, 1),
       (121, 1, 8945697, 2),
       (11, 2, 8995647, 3);
```

```
INSERT INTO room_reserve (room_number, passport,
                           check_in_date, check_out_date)
VALUES (12, 1234123456, '2023-03-22', '2023-03-25'),
       (13, 1234123458, '2023-03-22', '2023-03-28'),
       (176, 1234123460, '2023-03-22', '2023-03-26'),
       (13, 1234123462, '2023-03-22', null),
       (14, 1234123461, '2023-03-22', null);
```

```
INSERT INTO cleaner(id_cleaner, name, second_name, patronymic)
VALUES (1, 'Виталий','Сергеевич', 'Викторовна'),
       (2, 'Арина','Арина', 'Васильевич'),
       (3,'Паша','Петров' ,null),
       (4, 'Даша','Фёдоров' ,null),
       (5, 'Саша','Иванов' ,null),
       (6,'Александр','Коробкин' ,null),
       (7,'Виктория','Крутой' , 'Николаевич'),
       (8, 'Василий','Кружка' , 'Сергеевич');
```

```
INSERT INTO cleaning(week_day, floor, id_cleaner)
VALUES ('Mon', 2, 1),
       ('Wed', 1, 2),
       ('Sun', 3, 1),
       ('Sat', 4, 3),
```

('Fri', 3, 4),  
( 'Thu', 2, 5);

**ПРИЛОЖЕНИЕ В**  
**ССЫЛКА НА PR В GITHUB**

<https://github.com/moevm/sql-2023-1304/pull/42>