**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. 1303 |  | Андреева Е.А. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

**Цель работы.**

Развернуть локально PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы в задания.

**Текст задания**

Вариант 1

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.

**Выполнение работы**

Создание и заполнение таблиц (DataGrip):

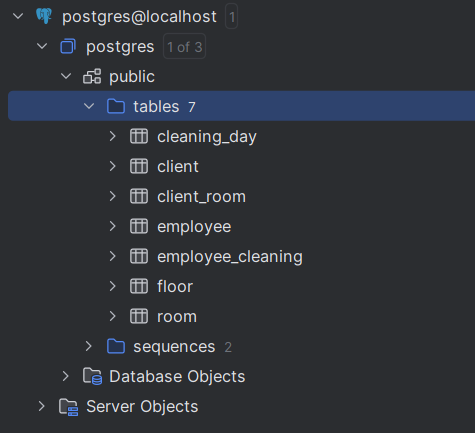
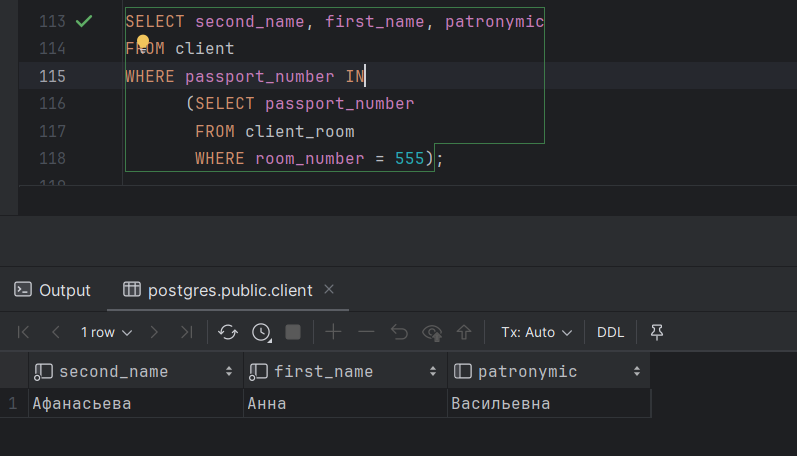


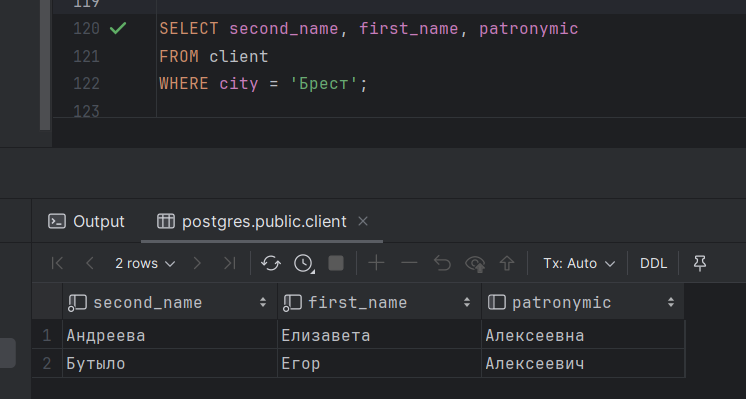
Рисунок 1 – Структура БД.

Выполним запросы к БД:

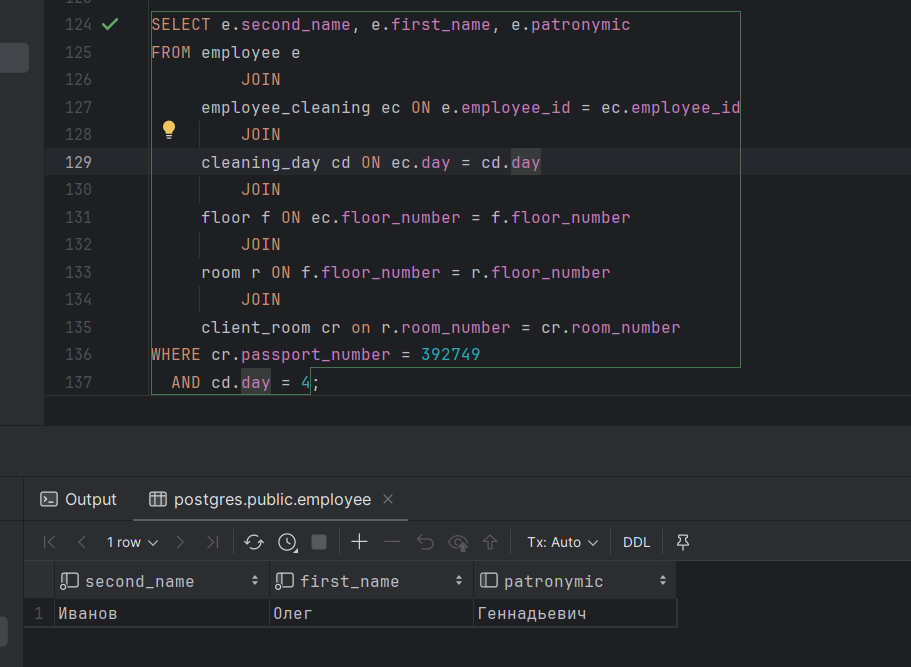
1. Получить информацию о клиентах, проживающих в заданном номере



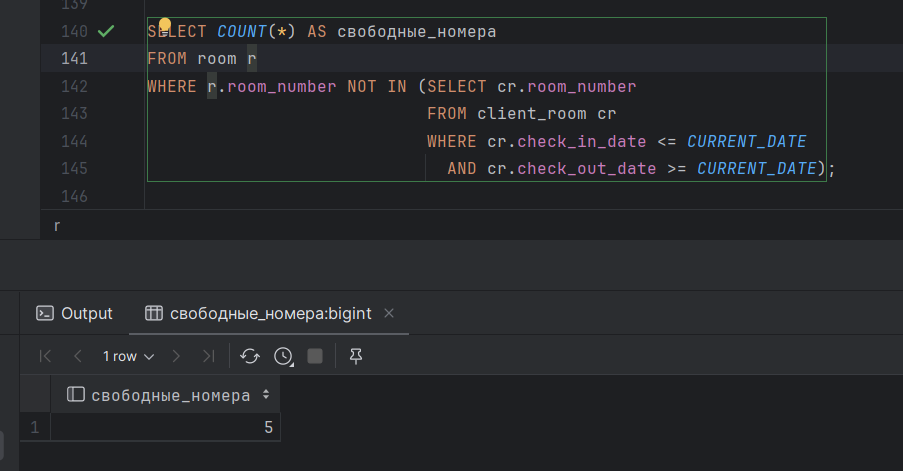
1. Получить информацию о клиентах, прибывших из заданного города



1. Получить информацию о том, кто из служащих убирал номер указанного клиента в заданный день недели

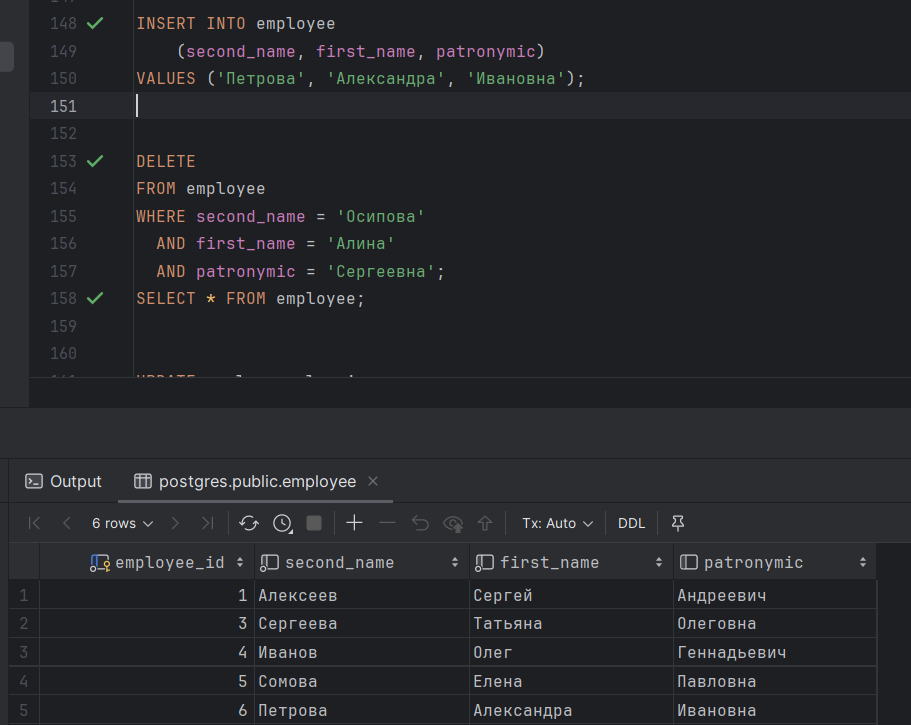


1. Узнать есть ли в гостинице свободные места и свободные номера и, если есть, то сколько

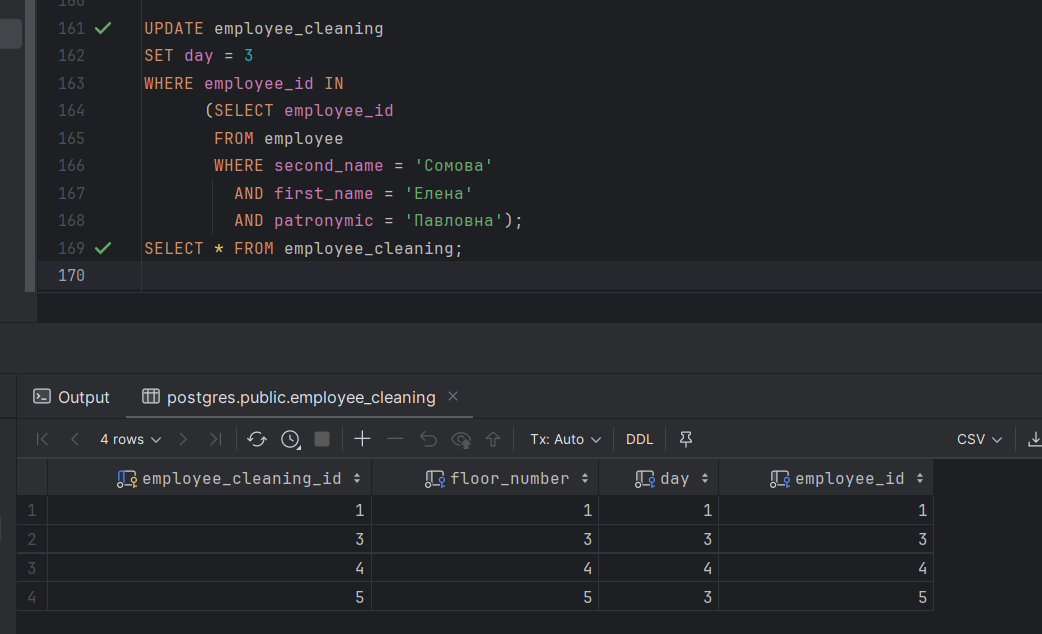


Администратор должен иметь возможность выполнить следующие операции:

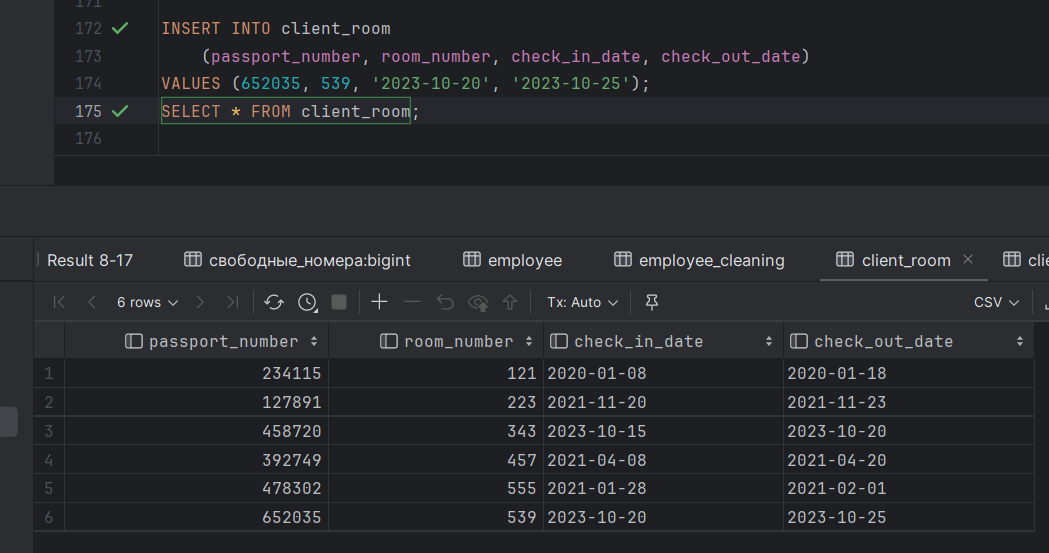
1. принять на работу или уволить служащего гостиницы

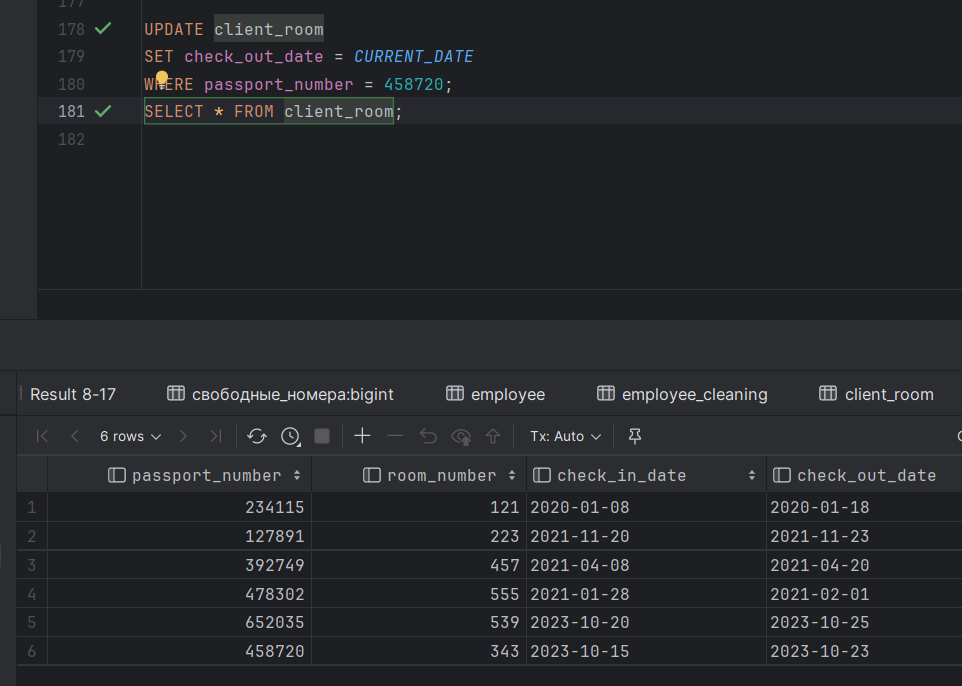


1. изменить расписание работы служащего



7. поселить или выселить клиента





**ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303/pull/27>

DB-Fiddle: <https://www.db-fiddle.com/f/eg5YTdvFZobTEQsJcfHkKQ/0>

**ПРИЛОЖЕНИЕ Б**

Файл lab2.sql:

CREATE TABLE client  
(  
 passport\_number INTEGER PRIMARY KEY,  
 second\_name TEXT not null,  
 first\_name TEXT not null,  
 patronymic TEXT,  
 city TEXT not null  
);  
  
INSERT INTO client  
 (passport\_number, second\_name, first\_name, patronymic, city)  
VALUES (234115, 'Андреева', 'Елизавета', 'Алексеевна', 'Брест'),  
 (127891, 'Козловская', 'Ольга', 'Александровна', 'Минск'),  
 (458720, 'Бутыло', 'Егор', 'Алексеевич', 'Брест'),  
 (392749, 'Григорьев', 'Вячеслав', 'Станиславович', 'Екатеринбург'),  
 (478302, 'Афанасьева', 'Анна', 'Васильевна', 'Калининград'),  
 (652035, 'Калинин', 'Григорий', 'Александрович', 'Омск');  
  
CREATE TABLE cleaning\_day  
(  
 day SMALLINT primary key  
);  
  
INSERT INTO cleaning\_day  
 (day)  
VALUES (1),  
 (2),  
 (3),  
 (4),  
 (5);  
  
CREATE TABLE employee  
(  
 employee\_id SERIAL PRIMARY KEY,  
 second\_name TEXT not null,  
 first\_name TEXT not null,  
 patronymic TEXT  
);  
  
INSERT INTO employee  
 (second\_name, first\_name, patronymic)  
VALUES ('Алексеев', 'Сергей', 'Андреевич'),  
 ('Осипова', 'Алина', 'Сергеевна'),  
 ('Сергеева', 'Татьяна', 'Олеговна'),  
 ('Иванов', 'Олег', 'Геннадьевич'),  
 ('Сомова', 'Елена', 'Павловна');  
  
  
CREATE TABLE floor  
(  
 floor\_number SMALLINT PRIMARY KEY  
);  
  
INSERT INTO floor  
 (floor\_number)  
VALUES (1),  
 (2),  
 (3),  
 (4),  
 (5);  
  
CREATE TABLE employee\_cleaning  
(  
 employee\_cleaning\_id SERIAL PRIMARY KEY,  
 floor\_number INTEGER not null REFERENCES floor (floor\_number) ON DELETE CASCADE,  
 day INTEGER REFERENCES cleaning\_day (day) ON DELETE SET NULL,  
 employee\_id INTEGER REFERENCES employee (employee\_id) ON DELETE SET NULL  
);  
  
INSERT INTO employee\_cleaning (floor\_number, day, employee\_id)  
VALUES (1, 1, 1),  
 (2, 2, 2),  
 (3, 3, 3),  
 (4, 4, 4),  
 (5, 5, 5);  
  
CREATE TABLE room  
(  
 room\_number INTEGER PRIMARY KEY,  
 type TEXT not null,  
 price INTEGER not null,  
 phone\_number INTEGER,  
 floor\_number SMALLINT not null REFERENCES floor (floor\_number) ON DELETE CASCADE  
);  
  
INSERT INTO room  
 (room\_number, type, price, phone\_number, floor\_number)  
VALUES (121, 'single', 12000, 2030121, 1),  
 (223, 'double', 22000, 2030223, 2),  
 (343, 'double', 22000, 2030343, 3),  
 (457, 'triple', 32000, 2030457, 4),  
 (555, 'triple', 50000, 2030555, 5),  
 (539, 'double', 32000, 2030539, 5);  
  
CREATE TABLE client\_room  
(  
 passport\_number INTEGER not null REFERENCES client (passport\_number),  
 room\_number INTEGER not null REFERENCES room (room\_number),  
 check\_in\_date DATE not null,  
 check\_out\_date DATE not null,  
 PRIMARY KEY (passport\_number, room\_number)  
);  
  
INSERT INTO client\_room  
 (passport\_number, room\_number, check\_in\_date, check\_out\_date)  
VALUES (234115, 121, '2020-01-08', '2020-01-18'),  
 (127891, 223, '2021-11-20', '2021-11-23'),  
 (458720, 343, '2023-10-15', '2023-10-20'),  
 (392749, 457, '2021-04-08', '2021-04-20'),  
 (478302, 555, '2021-01-28', '2021-02-01');  
  
  
SELECT second\_name, first\_name, patronymic  
FROM client  
WHERE passport\_number IN  
 (SELECT passport\_number  
 FROM client\_room  
 WHERE room\_number = 555);  
  
SELECT second\_name, first\_name, patronymic  
FROM client  
WHERE city = 'Брест';  
  
SELECT e.second\_name, e.first\_name, e.patronymic  
FROM employee e  
 JOIN  
 employee\_cleaning ec ON e.employee\_id = ec.employee\_id  
 JOIN  
 cleaning\_day cd ON ec.day = cd.day  
 JOIN  
 floor f ON ec.floor\_number = f.floor\_number  
 JOIN  
 room r ON f.floor\_number = r.floor\_number  
 JOIN  
 client\_room cr on r.room\_number = cr.room\_number  
WHERE cr.passport\_number = 392749  
 AND cd.day = 4;  
  
  
SELECT *COUNT*(\*) AS свободные\_номера  
FROM room r  
WHERE r.room\_number NOT IN (SELECT cr.room\_number  
 FROM client\_room cr  
 WHERE cr.check\_in\_date <= *CURRENT\_DATE* AND cr.check\_out\_date >= *CURRENT\_DATE*);  
  
  
INSERT INTO employee  
 (second\_name, first\_name, patronymic)  
VALUES ('Петрова', 'Александра', 'Ивановна');  
  
  
DELETE  
FROM employee  
WHERE second\_name = 'Осипова'  
 AND first\_name = 'Алина'  
 AND patronymic = 'Сергеевна';  
SELECT \* FROM employee;  
  
  
UPDATE employee\_cleaning  
SET day = 3  
WHERE employee\_id IN  
 (SELECT employee\_id  
 FROM employee  
 WHERE second\_name = 'Сомова'  
 AND first\_name = 'Елена'  
 AND patronymic = 'Павловна');  
SELECT \* FROM employee\_cleaning;  
  
  
INSERT INTO client\_room  
 (passport\_number, room\_number, check\_in\_date, check\_out\_date)  
VALUES (652035, 539, '2023-10-20', '2023-10-25');  
SELECT \* FROM client\_room;  
  
  
UPDATE client\_room  
SET check\_out\_date = *CURRENT\_DATE*WHERE passport\_number = 458720;  
SELECT \* FROM client\_room;