

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 заданиях.

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):

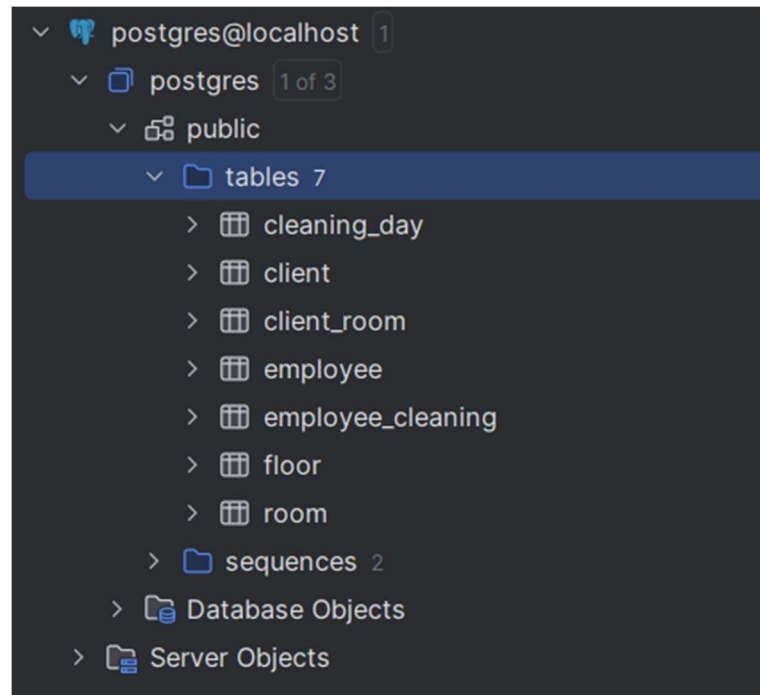
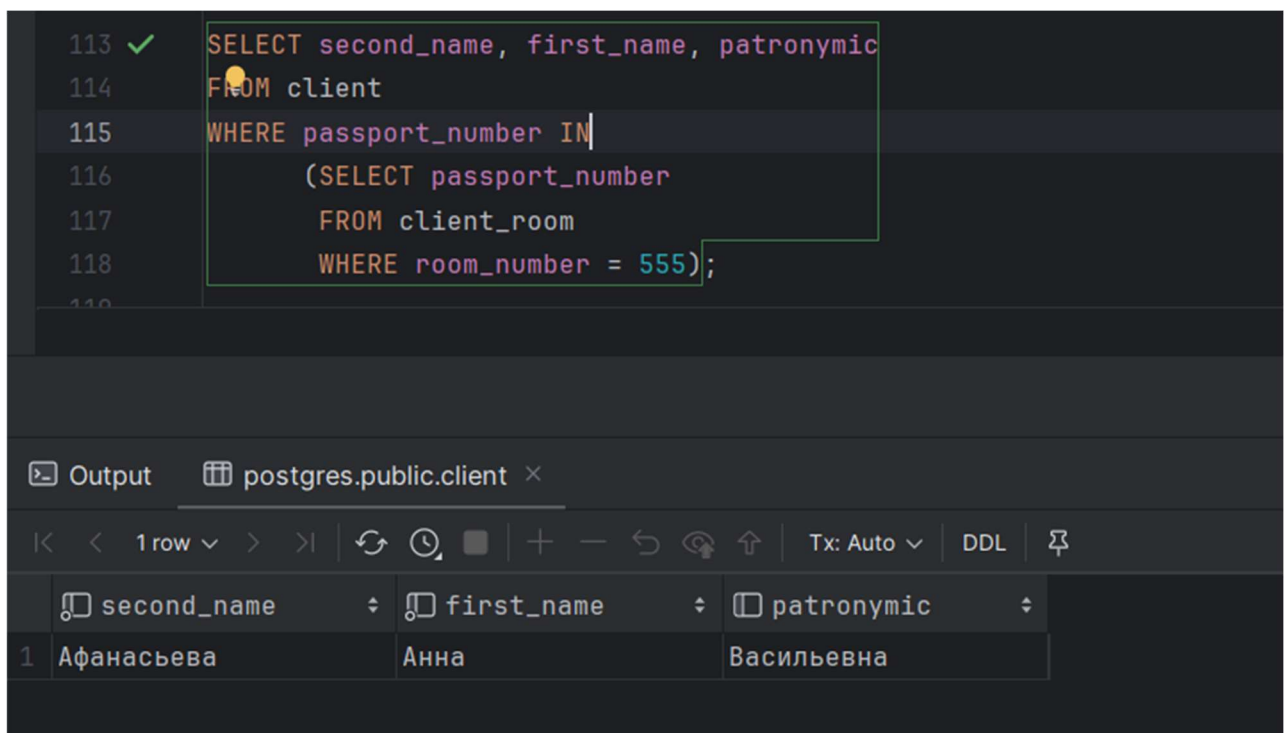


Рисунок 1 – Структура БД.

Выполним запросы к БД:

1. Получить информацию о клиентах, проживающих в заданном номере



2. Получить информацию о клиентах, прибывших из заданного города

```
119
120 ✓ SELECT second_name, first_name, patronymic
121 FROM client
122 WHERE city = 'Брест';
123
```

Output postgres.public.client x

2 rows

	second_name	first_name	patronymic
1	Андреева	Елизавета	Алексеевна
2	Бутыло	Егор	Алексеевич

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

```
124 ✓ SELECT e.second_name, e.first_name, e.patronymic
125 FROM employee e
126 JOIN
127 employee_cleaning ec ON e.employee_id = ec.employee_id
128 JOIN
129 cleaning_day cd ON ec.day = cd.day
130 JOIN
131 floor f ON ec.floor_number = f.floor_number
132 JOIN
133 room r ON f.floor_number = r.floor_number
134 JOIN
135 client_room cr on r.room_number = cr.room_number
136 WHERE cr.passport_number = 392749
137 AND cd.day = 4;
```

Output postgres.public.employee x

1 row

	second_name	first_name	patronymic
1	Иванов	Олег	Геннадьевич

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

```
139
140 ✓ SELECT COUNT(*) AS свободные_номера
141 FROM room r
142 WHERE r.room_number NOT IN (SELECT cr.room_number
143                               FROM client_room cr
144                               WHERE cr.check_in_date <= CURRENT_DATE
145                                     AND cr.check_out_date >= CURRENT_DATE);
146
r
```

Output свободные_номера:bigint x

1 row

свободные_номера
5

Администратор должен иметь возможность выполнить следующие операции:

5. принять на работу или уволить служащего гостиницы

```
148 ✓ INSERT INTO employee
149       (second_name, first_name, patronymic)
150 VALUES ('Петрова', 'Александра', 'Ивановна');
151
152
153 ✓ DELETE
154 FROM employee
155 WHERE second_name = 'Осипова'
156        AND first_name = 'Алина'
157        AND patronymic = 'Сергеевна';
158 ✓ SELECT * FROM employee;
159
160
161 UPDATE
```

Output postgres.public.employee x

6 rows

	employee_id	second_name	first_name	patronymic
1	1	Алексеев	Сергей	Андреевич
2	3	Сергеева	Татьяна	Олеговна
3	4	Иванов	Олег	Геннадьевич
4	5	Сомова	Елена	Павловна
5	6	Петрова	Александра	Ивановна

6. изменить расписание работы служащего

```

161 ✓ UPDATE employee_cleaning
162 SET day = 3
163 WHERE employee_id IN
164     (SELECT employee_id
165      FROM employee
166      WHERE second_name = 'Сомова'
167            AND first_name = 'Елена'
168            AND patronymic = 'Павловна');
169 ✓ SELECT * FROM employee_cleaning;
170

```

Output postgres.public.employee_cleaning

	employee_cleaning_id	floor_number	day	employee_id
1	1	1	1	1
2	3	3	3	3
3	4	4	4	4
4	5	5	3	5

7. поселить или выселить клиента

```

172 ✓ INSERT INTO client_room
173     (passport_number, room_number, check_in_date, check_out_date)
174     VALUES (652035, 539, '2023-10-20', '2023-10-25');
175 ✓ SELECT * FROM client_room;
176

```

Result 8-17 свободные_номера:bigint employee employee_cleaning client_room

	passport_number	room_number	check_in_date	check_out_date
1	234115	121	2020-01-08	2020-01-18
2	127891	223	2021-11-20	2021-11-23
3	458720	343	2023-10-15	2023-10-20
4	392749	457	2021-04-08	2021-04-20
5	478302	555	2021-01-28	2021-02-01
6	652035	539	2023-10-20	2023-10-25

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;
```