ГУАП

КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель Н. В. Путилова

должность, уч. степень, звание подпись, дата инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №3

Заполнение таблиц и модификация данных

|  |  |  |  |
| --- | --- | --- | --- |
| РАБОТУ ВЫПОЛНИЛ |  |  |  |
| СТУДЕНТ гр. № | Z1431 |  | М.Д. Быстров |
|  |  | подпись, дата | инициалы, фамилия |

по курсу: «Проектирование баз данных»

Санкт-Петербург 2024

**ЗАДАНИЕ**

**Лабораторная работа №3 Заполнение таблиц и модификация данных** Цель работы: Получение умений и навыков манипулирования данными в реляционной базе данных

**Задание и последовательность выполнения работы**

1)Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 2.

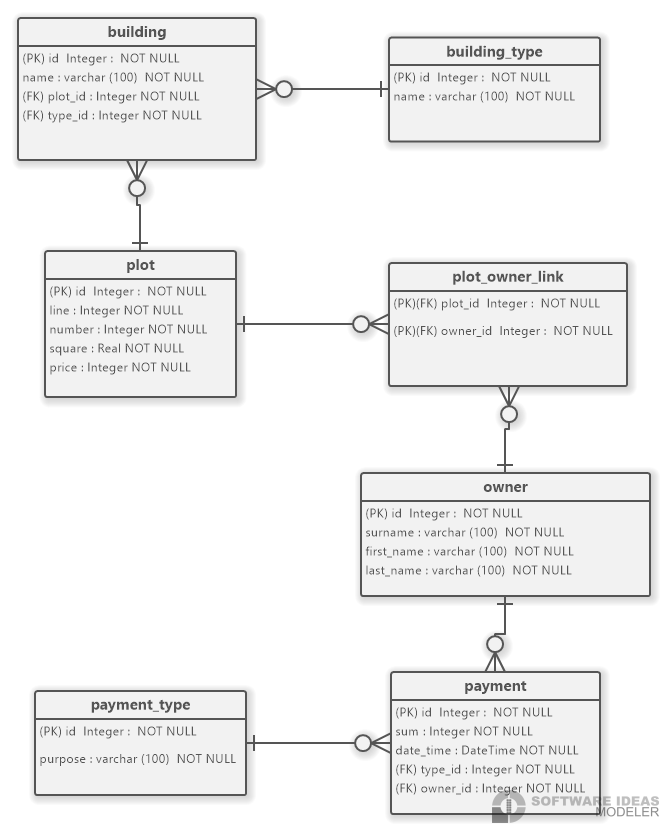
В строках, вставляемых в таблицы, должны быть данные как удовлетворяющие, так и не удовлетворяющие условиям запросов, приведенных в варианте задания. (Для демонстрации этого необходимо в отчете создать таблицу, где будет указано задание на запрос, данные удовлетворяющие условиям запроса, данные не удовлетворяющие условиям запроса)

2)Необходимо привести свои примеры использования операторов *update* и *delete* и *merge* с описанием их назначения.

**Вариант** 2:

1. Садоводство: участки, владельцы с учетом совместной собственности, линии/номер участка, площадь стоимость постройки, тип построек, взносы в фонд садоводства
2. номера участков владельцев с отчеством, заканчивающимся на «ич», но не начинающиеся на букву «А»
3. участки, на которых зарегистрировано более 1 типа постройки
4. Тип (типы) построек, которые отсутствуют на участках
5. Владелец (владельцы) участка максимальной площади
6. Владельцы участков числом типов построек больше среднего
7. Владельцы, оплатившие в 2023 году , все типы взносов
8. Участки, на которых нет беседок, но есть туалеты или бани

# Физическая модель БД для СУБД Postgresql



# Данные в таблицах БД

|  |  |
| --- | --- |
| Имя таблицы | Данные |
| plot |  |
| building\_type |  |
| building |  |
| owner |  |
| plot\_owner\_link |  |
| payment\_type |  |
| payment |  |

# Скрипт полного заполнения базы

INSERT INTO public.building\_type(name) VALUES ('Жилой дом');

INSERT INTO public.building\_type(name) VALUES ('Туалет');

INSERT INTO public.building\_type(name) VALUES ('Баня');

INSERT INTO public.building\_type(name) VALUES ('Сарай');

INSERT INTO public.building\_type(name) VALUES ('Беседка');

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Главный дом', 1, 1);

INSERT INTO public.building-(name, plot\_id, owner\_id) VALUES ('Туалет', 1, 2);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Финская баня', 1, 3);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Сарай у дороги', 1, 4);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Открытая беседка', 1, 5);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Дом', 2, 1);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Русская баня', 2, 3);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Дом у дороги', 3, 1);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Туалет', 3, 2);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Дом из бруса', 4, 1);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Баня', 5, 3);

INSERT INTO public.building(name, plot\_id, owner\_id) VALUES ('Дом кирпичный', 5, 1);

INSERT INTO public.payment\_type(purpose) VALUES ('Ежегодный взнос');

INSERT INTO public.payment\_type(purpose) VALUES ('ЖКХ');

INSERT INTO public.owner(surname, first\_name, last\_name) VALUES ('Иванов', 'Дмитрий', 'Петрович');

INSERT INTO public.owner(surname, first\_name, last\_name) VALUES ('Петров', 'Иван', 'Дмитриевич');

INSERT INTO public.owner(surname, first\_name, last\_name) VALUES ('Дмитриев', 'Петр', 'Иванович');

INSERT INTO public.owner(surname, first\_name, last\_name) VALUES ('Иванов', 'Петр', 'Дмитриевич');

INSERT INTO public.owner(surname, first\_name, last\_name) VALUES 'Дмитриев', 'Иван', 'Алексеевич');

INSERT INTO public.owner(surname, first\_name, last\_name) VALUES ('Петров', 'Дмитрий', 'Петрович');

INSERT INTO public.payment(sum, date\_time, type\_id, owner\_id) VALUES (500000, '2024-05-16 13:00:00', 2, 6);

INSERT INTO public.payment(sum, date\_time, type\_id, owner\_id) VALUES (500000, '2023-05-16 13:00:00', 1, 1);

INSERT INTO public.payment(sum, date\_time, type\_id, owner\_id) VALUES (500000, '2023-05-16 13:00:00', 1, 4);

INSERT INTO public.payment(sum, date\_time, type\_id, owner\_id) VALUES (500000, '2023-05-16 13:00:00', 1, 6);

INSERT INTO public.payment(sum, date\_time, type\_id, owner\_id) VALUES (500000, '2023-04-16 13:00:00', 1, 2);

INSERT INTO public.payment(sum, date\_time, type\_id, owner\_id) VALUES (500000, '2023-03-16 13:00:00', 1, 3);

INSERT INTO public.plot(line, number. square, price) VALUES (1, 1, 1000, 100000000);

INSERT INTO public.plot(line, number. square, price) VALUES (2, 2, 1500, 150000000);

INSERT INTO public.plot(line, number. square, price) VALUES (1, 3, 800, 80000000);

INSERT INTO public.plot(line, number. square, price) VALUES (2, 4, 1100, 110000000);

INSERT INTO public.plot(line, number. square, price) VALUES (2, 5, 1120, 112000000);

INSERT INTO public.plot\_owner\_link VALUES (1, 1);

INSERT INTO public.plot\_owner\_link VALUES (2, 2);

INSERT INTO public.plot\_owner\_link VALUES (3, 3);

INSERT INTO public.plot\_owner\_link VALUES (4, 4);

INSERT INTO public.plot\_owner\_link VALUES (5, 5);

INSERT INTO public.plot\_owner\_link VALUES (5, 6);

# Таблица тестовых данных

Тестовые данные

|  |  |  |  |
| --- | --- | --- | --- |
| Текст запроса | Данные, удовлетворяющие условиям | Данные, не удовлетворяющие |  |
|  | запросов | условиям запросов |  |
| номера участков владельцев с отчеством, заканчивающимся на «ич», но не начинающиеся на букву «А» | plot (id)  1, 2, 5 | plot (id)  3,4 |  |
| участки, на которых зарегистрировано более 1 типа постройки | plot(id)  1,2,3,5 | plot(id)  4 |  |
| Тип (типы) построек, которые отсутствуют на участках | building\_type(id)  4 Сарай | building\_type(id)  1 Жилой дом  2 Туалет  3 Баня  5 Беседка |  |
| Владелец (владельцы) участка максимальной площади | owner(id)  2 | owner(id)  1,3,4,5,6 | |
| Владельцы участков числом типов построек больше среднего | owner(id)  1 | owner(id)  2,3,4,5,6 | |
| Владельцы, оплатившие в 2023 году , все типы взносов | owner(id)  6 | owner(id)  1,2,3,4,5 | |
| Участки, на которых нет беседок, но есть туалеты или бани | plot(id)  2,3,5 | plot(id)  1,4 | |

# Примеры использования insert, update, merge

3. **INSERT**: Пример вставки нового платежа:

INSERT INTO public.payment VALUES (4, 500000, '2023-05-16 13:00:00', 1, 4);

1. **UPDATE**: Пример обновления информации о платеже:

UPDATE payment

SET sum=1, date\_time='2023-05-16 13:00:00', type\_id=1, owner\_id=1

WHERE id=1;

1. **MERGE**: Пример вставки нового платежа или замены существующего:

Payment1 – таблица, идентичная по структуре таблице payment.

MERGE INTO payment p1

USING payment1 p2

ON p1.id = p2.id

WHEN MATCHED THEN

UPDATE SET p1.sum = p2.sum

WHEN NOT MATCHED THEN

INSERT (sum, date\_time, type\_id, owner\_id)

VALUES (p2.sum, p2.date\_time, p2.type\_id, p2.owner\_id);

**Выводы об особенностях манипулирования данными в выбранной СУБД.**

В СУБД Postgresql для оператора MERGE нет операции NOT MATCHED BY TARGET, однако этот оператор присутствует.