

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель
должность, уч. степень, звание

подпись, дата

Н. В. Путилова
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №3
Заполнение таблиц и модификация данных

по курсу: «Проектирование баз данных»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

Z1431

подпись, дата

М.Д. Быстров
инициалы, фамилия

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.

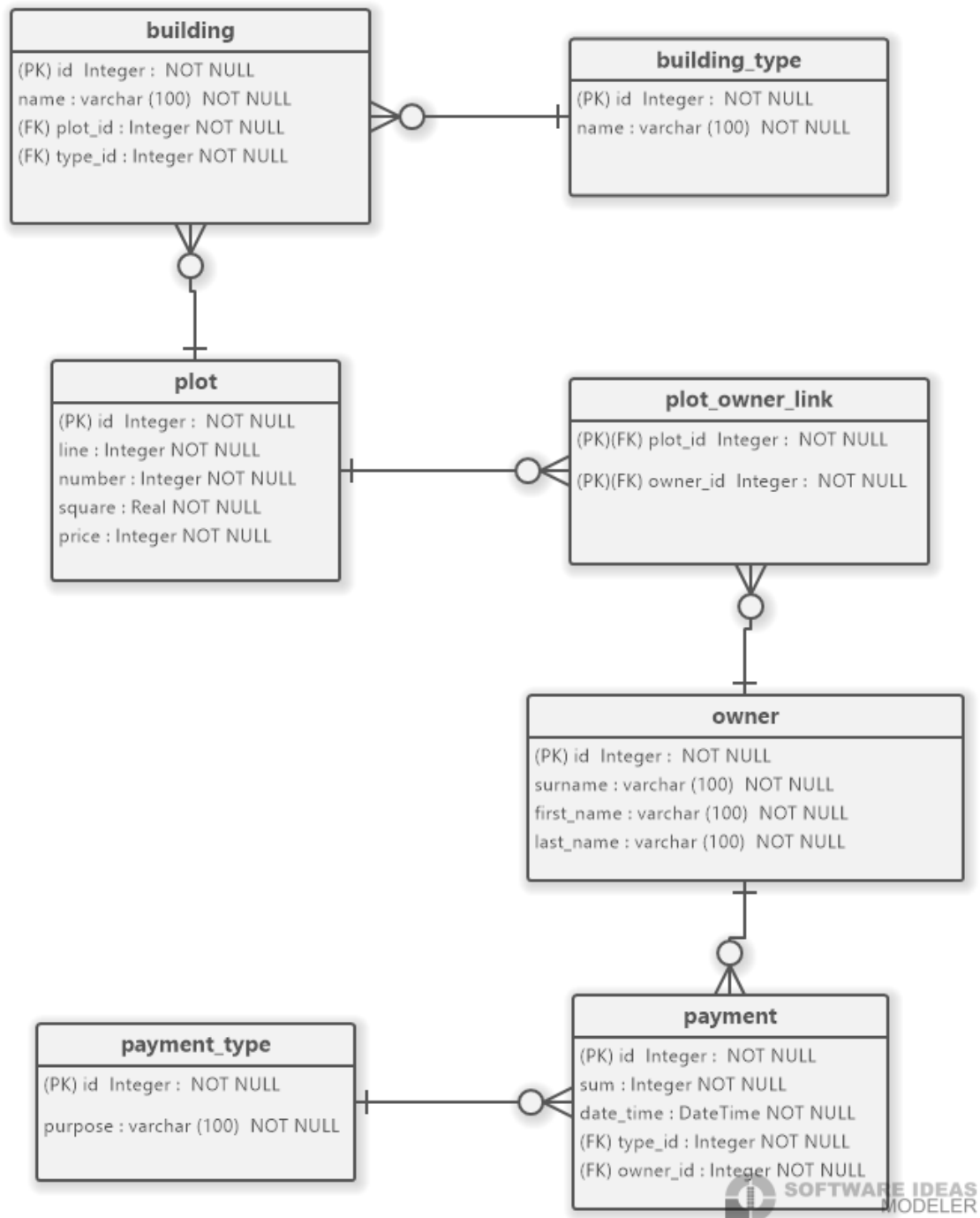
Вариант 2:

1. Садоводство: участки, владельцы с учетом совместной собственности, линии/номер участка, площадь стоимость постройки, тип построек, взносы в фонд садоводства
 - а. номера участков владельцев с отчеством, заканчивающимся на «ич», но не начинающиеся на букву «А»
 - б. участки, на которых зарегистрировано более 1 типа постройки
 - в. Тип (типы) построек, которые отсутствуют на участках
 - г. Владелец (владельцы) участка максимальной площади
 - д. Владельцы участков числом типов построек больше среднего

е. Владельцы, оплатившие в 2023 году , все типы взносов

ж. Участки, на которых нет беседок, но есть туалеты или бани

Физическая модель БД для СУБД Postgresql



Данные в таблицах БД

Имя таблицы	Данные																																				
plot	<div><div>plot</div><div>Введите SQL выражение чтобы отфильтровать результаты</div><table><tr><td></td><td>123 id</td><td>123 line</td><td>123 number</td><td>123 square</td><td>123 price</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1 000</td><td>100 000 000</td></tr><tr><td>2</td><td>2</td><td>2</td><td>2</td><td>1 500</td><td>150 000 000</td></tr><tr><td>3</td><td>3</td><td>1</td><td>3</td><td>800</td><td>80 000 000</td></tr><tr><td>4</td><td>4</td><td>2</td><td>4</td><td>1 100</td><td>110 000 000</td></tr><tr><td>5</td><td>5</td><td>2</td><td>5</td><td>1 120</td><td>112 000 000</td></tr></table></div>		123 id	123 line	123 number	123 square	123 price	1	1	1	1	1 000	100 000 000	2	2	2	2	1 500	150 000 000	3	3	1	3	800	80 000 000	4	4	2	4	1 100	110 000 000	5	5	2	5	1 120	112 000 000
	123 id	123 line	123 number	123 square	123 price																																
1	1	1	1	1 000	100 000 000																																
2	2	2	2	1 500	150 000 000																																
3	3	1	3	800	80 000 000																																
4	4	2	4	1 100	110 000 000																																
5	5	2	5	1 120	112 000 000																																
building_type	<div><div>building_type</div><div>Введите SQL вы</div><table><tr><td></td><td>123 id</td><td>abc name</td></tr><tr><td>1</td><td>1</td><td>Жилой дом</td></tr><tr><td>2</td><td>2</td><td>Туалет</td></tr><tr><td>3</td><td>3</td><td>Баня</td></tr><tr><td>4</td><td>4</td><td>Сарай</td></tr><tr><td>5</td><td>5</td><td>Беседка</td></tr></table></div>		123 id	abc name	1	1	Жилой дом	2	2	Туалет	3	3	Баня	4	4	Сарай	5	5	Беседка																		
	123 id	abc name																																			
1	1	Жилой дом																																			
2	2	Туалет																																			
3	3	Баня																																			
4	4	Сарай																																			
5	5	Беседка																																			

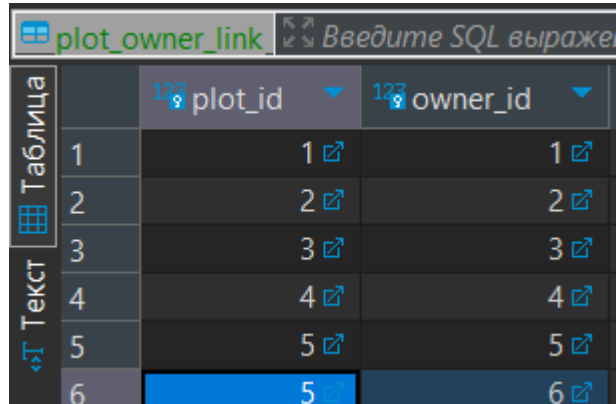
building

building Введите SQL выражение чтобы отфильтровать результаты				
	123 id	ABC name	123 plot_id	123 type_id
1	1	Главный дом	1	1
2	2	Туалет	1	2
3	3	Финская баня	1	3
4	5	Открытая беседка	1	5
5	6	Дом	2	1
6	7	Русская баня	2	3
7	8	Дом у дороги	3	1
8	9	Туалет	3	2
9	10	Дом из бруса	4	1
10	11	Баня	5	3
11	12	Дом кирпичный	5	1

owner

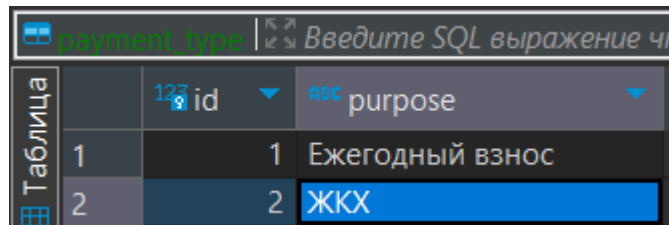
owner Введите SQL выражение чтобы отфильтровать результаты				
	123 id	ABC surname	ABC first_name	ABC last_name
1	1	Иванов	Дмитрий	Петрович
2	2	Петров	Иван	Дмитриевич
3	3	Дмитриева	Анастасия	Ивановна
4	4	Дмитриев	Иван	Алексеевич
5	5	Иванов	Петр	Дмитриевич
6	6	Петров	Дмитрий	Петрович

plot_owner_link



	plot_id	owner_id
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	5	6

payment_type



	id	purpose
1	1	Ежегодный взнос
2	2	ЖКХ

payment

payment Введите SQL выражение чтобы отфильтровать результаты					
Таблица	id	sum	date_time	type_id	owner_id
1	1	500 000	2023-05-16 13:00:00.000	1	1
2	2	500 000	2023-04-16 13:00:00.000	1	2
3	3	500 000	2023-03-16 13:00:00.000	1	3
4	4	500 000	2023-05-16 13:00:00.000	1	4
5	5	500 000	2023-05-16 13:00:00.000	1	6
6	6	500 000	2024-05-16 13:00:00.000	2	6

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 запросов	Данные, не удовлетворяющие 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 Петров Иван Дмитриевич	owner(id) 1 Иванов Дмитрий Петрович 3 Дмитриева Анастасия Ивановна 4 Дмитриев Иван Алексеевич 5 Иванов Петр Дмитриевич 6 Петров Дмитрий Петрович
Владельцы участков числом типов построек больше среднего	owner(id) 1 Иванов Дмитрий Петрович	owner(id) 2 Петров Иван Дмитриевич 3 Дмитриева Анастасия Ивановна 4 Дмитриев Иван Алексеевич 5 Иванов Петр Дмитриевич 6 Петров Дмитрий Петрович
Владельцы, оплатившие в 2023 году , все типы взносов	owner(id) 6 Петров Дмитрий Петрович	owner(id) 1 Иванов Дмитрий Петрович 2 Петров Иван Дмитриевич 3 Дмитриева Анастасия Ивановна 4 Дмитриев Иван Алексеевич 5 Иванов Петр Дмитриевич
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);
```

2. **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;
```

3. **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 SOURCE, однако этот оператор присутствует.