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 №5

Разработка SQL запросов: запросы с подзапросами

по курсу: «Проектирование баз данных»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № Z1431 М.Д. Быстров

подпись, дата инициалы, фамилия

Санкт-Петербург 2024

**ЗАДАНИЕ**

**Лабораторная работа №5 Разработка SQL запросов:**

**запросы с подзапросами**

Цель работы: Получение навыков создания запросов с подзапросами.

**Задание и последовательность выполнения работы**

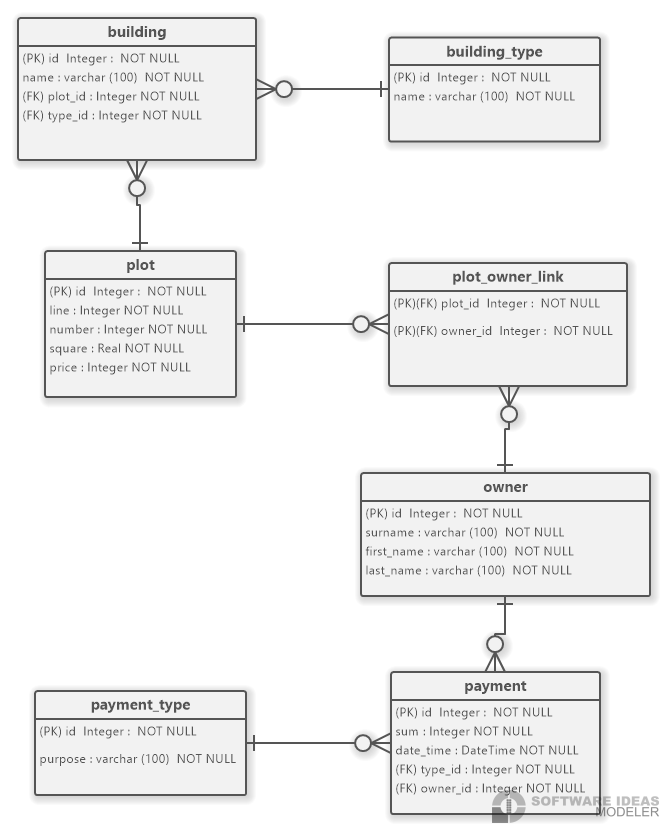
По аналогии с примерами, приведенными выше реализовать запросы г) .. ж), указанные в варианте задания. Один из запросов на максимум/минимум реализовать с помощью директивы *all*. Запрос на «все» (реляционное деление) реализовать с помощью 2 *not exists* и с помощью агрегатной функции. Запросы на разность реализовать в 3 вариантах: *Not in, except* (*except* MySQL не поддерживает, поэтому только синтаксис), с

использованием левого/правого соединения

**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



**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 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 | |

# Результаты работы

|  |  |  |
| --- | --- | --- |
| запрос | текст запросов на SQL; | наборы данных, возвращаемые запросами  (скриншоты); |
| Владелец (владельцы) участка максимальной площади | -- all  **select** **distinct** *o*.\* **from** "owner" *o*  **join** plot\_owner\_link *link* **ON** *link*.owner\_id = *o*.id  **join** plot **as** *p1* **on** *p1*.id = *link*.plot\_id  **where** *p1*.square >= **all**  (**select** *p2*.square **from** plot *p2*);  -- max  **select** **distinct** *o*.\* **from** "owner" *o*  **join** plot\_owner\_link *link* **ON** *link*.owner\_id = *o*.id  **join** plot **as** *p1* **on** *p1*.id = *link*.plot\_id  **where** *p1*.square =  (**select** **max**(*p2*.square) **from** plot *p2*); |  |
| Владельцы участков числом типов построек больше среднего | **select** **distinct** *o*.\* **from** "owner" *o*  **join** plot\_owner\_link *link* **ON** *link*.owner\_id = *o*.id  **join** plot **as** *p1* **on** *p1*.id = *link*.plot\_id  **join** building *b* **on** *b*.plot\_id = *p1*.id  **group** **by** *o*.id, *p1*.id  **having** **count**(**distinct** *b*.type\_id) > (  **select** **avg**(*cnt*) **from** (  **select** **count**(**distinct** *b*.type\_id) **as** *cnt* **from** plot *p1*  **left** **join** building *b* **on** *b*.plot\_id = *p1*.id  **group** **by** *p1*.id) *q*); |  |
| Владельцы, оплатившие в 2023 году , все типы взносов | -- not exists  **select** *o*.\* **from** "owner" *o*  **where** **not** **exists** (  **select** \* **from** payment\_type *pt*  **where** **not** **exists**(  **select** \* **from** payment *p*  **where** *p*.owner\_id = *o*.id  **and** *p*.type\_id = *pt*.id  **and** **date\_part**(**'year'**, *p*.date\_time)=2023))  -- агр функция  **select** *o*.\*, **count**(**distinct** *p*.type\_id) **from** "owner" *o*  **join** payment *p*  **on** *p*.owner\_id = *o*.id  **where** **date\_part**(**'year'**, *p*.date\_time)=2023  **group** **by** *o*.id  **having** **count**(**distinct** *p*.type\_id) =  (**select** **count**(*pt*.id) **from** payment\_type *pt*); |  |
| Участки, на которых нет беседок, но есть туалеты или бани | -- not in вариант  **select** *p*.\* **from** plot *p*  **join** building *b*  **on** *b*.plot\_id = *p*.id  **join** building\_type *bt*  **on** *bt*.id = *b*.type\_id  **where** *bt*."name" **in** (**'Туалет'**, **'Баня'**)  **and** *p*.id **not** **in** (  **select** *p*.id **from** plot *p*  **join** building *b*  **on** *b*.plot\_id = *p*.id  **join** building\_type *bt*  **on** *bt*.id = *b*.type\_id  **where** *bt*."name" **in** (**'Беседка'**));  -- except вариант  **select** *p*.\* **from** plot *p*  **join** building *b*  **on** *b*.plot\_id = *p*.id  **join** building\_type *bt*  **on** *bt*.id = *b*.type\_id  **where** *bt*."name" **in** (**'Туалет'**, **'Баня'**)  **except**  **select** *p*.\* **from** plot *p*  **join** building *b*  **on** *b*.plot\_id = *p*.id  **join** building\_type *bt*  **on** *bt*.id = *b*.type\_id  **where** *bt*."name" **in** (**'Беседка'**);  -- left join вариант  **select** *p1*.\* **from** plot *p1*  **join** building *b1*  **on** *b1*.plot\_id = *p1*.id  **join** building\_type *bt1*  **on** *bt1*.id = *b1*.type\_id  **left** **join** (**select** *p2*.\* **from** plot *p2*  **join** building *b2*  **on** *b2*.plot\_id = *p2*.id  **join** building\_type *bt2*  **on** *bt2*.id = *b2*.type\_id  **and** *bt2*."name" = **'Беседка'**) *q*  **on** *q*.id = *p1*.id  **where** *bt1*."name" **in** (**'Туалет'**, **'Баня'**)  **and** *q*.id **is** **null**; |  |

**Выводы об особенностях реализации простых запросов на выборку.**

Необходимые запросы реализованы для СУБД Postgresql. Использованы аргегатные функции, not exists, except, all, not in, left join. Результаты запросов соответствуют ожидаемым.