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 №2

Создание и модификация базы данных и таблиц

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

по курсу: «Проектирование баз данных»

Санкт-Петербург 2024

**ЗАДАНИЕ**

**Лабораторная работа №2 Создание и модификация базы данных и таблиц базы данных**

Цель работы: Получение умений и навыков создания и модификации таблиц на языке SQL.

**Задание и последовательность выполнения работы**

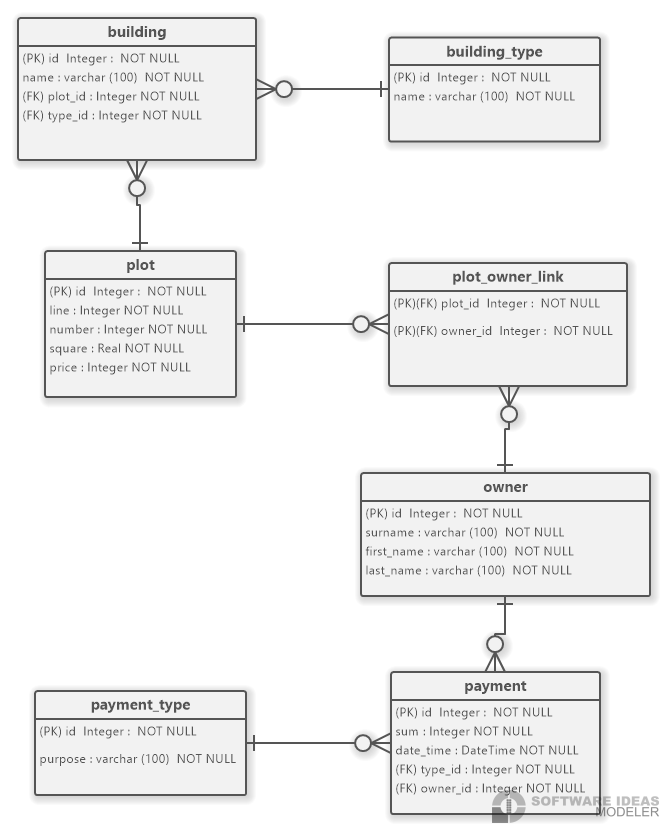
В соответствии с моделью, разработанной в предыдущей работе, создать базу данных. Продемонстрировать умение добавить и удалить столбец командой *alter table*.

**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

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

**Физическая модель БД для СУБД Sqlite**



**Таблица с описанием ссылочной целостности**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Дочерняя  таблица | Внешний  ключ | Родительская  таблица | ссылочная  целостность  при  удалении | Обоснование типа  ссылочной  целостности  при  удалении | Ссылочная  целостность  при  обновлении | Обоснование типа  ссылочной  целостности  при  обновлении |
| building | plot\_id | plot | Ограничение (Не дает удалить данные из таблицы plot, если с ними связаны какие-либо данные в таблице building) | Удаление связанных с таблицей building записей является неприемлемым ввиду возможной потери важных для работы системы данных (данных о постройках на существующих участках) | Каскадирование (При обновлении первичного ключа родительской таблицы plot обновятся внешние ключи таблицы building) | Записи дочерней таблицы building должны быть всегда связаны с единственным соответствующим объектом в родительской таблице plot, иначе происходит потеря важной для системы информации. |
| building | type\_id | building\_type | Ограничение (Не дает удалить данные из таблицы building\_type, если с ними связаны какие-либо данные в таблице building) | Удаление связанных с таблицей building записей является неприемлемым ввиду возможной потери важных для работы системы данных | Каскадирование (При обновлении первичного ключа родительской таблицы building\_type обновятся внешние ключи таблицы building) | Записи дочерней таблицы building должны быть всегда связаны с единственным соответствующим объектом в родительской таблице building\_type, иначе происходит потеря важной для системы информации. |
| plot\_owner\_link | owner\_id | owner | Ограничение (Не дает удалить данные из таблицы owner, если с ними связаны какие-либо данные в таблице plot\_owner\_link) | Удаление связанных с таблицей plot\_owner\_link записей является неприемлемым ввиду возможной потери важных для работы системы данных | Каскадирование (При обновлении первичного ключа родительской таблицы owner обновятся внешние ключи таблицы plot\_owner\_link) | Записи дочерней таблицы plot\_owner\_link должны быть всегда связаны с единственным соответствующим объектом в родительской таблице owner, иначе происходит потеря важной для системы информации. |
| plot\_owner\_link | plot\_id | plot | Ограничение (Не дает удалить данные из таблицы plot, если с ними связаны какие-либо данные в таблице plot\_owner\_link) | Удаление связанных с таблицей plot\_owner\_link записей является неприемлемым ввиду возможной потери важных для работы системы данных | Каскадирование (При обновлении первичного ключа родительской таблицы plot обновятся внешние ключи таблицы plot\_owner\_link) | Записи дочерней таблицы plot\_owner\_link должны быть всегда связаны с единственным соответствующим объектом в родительской таблице plot, иначе происходит потеря важной для системы информации. |
| payment | owner\_id | owner | Ограничение (Не дает удалить данные из таблицы owner, если с ними связаны какие-либо данные в таблице payment) | Удаление связанных с таблицей payment записей является неприемлемым ввиду возможной потери важных для работы системы данных | Каскадирование (При обновлении первичного ключа родительской таблицы owner обновятся внешние ключи таблицы payment) | Записи дочерней таблицы payment должны быть всегда связаны с единственным соответствующим объектом в родительской таблице owner, иначе происходит потеря важной для системы информации. |
| payment | type\_id | payment\_type | Ограничение (Не дает удалить данные из таблицы payment\_type, если с ними связаны какие-либо данные в таблице payment) | Удаление связанных с таблицей payment записей является неприемлемым ввиду возможной потери важных для работы системы данных | Каскадирование (При обновлении первичного ключа родительской таблицы payment\_type обновятся внешние ключи таблицы payment) | Записи дочерней таблицы payment должны быть всегда связаны с единственным соответствующим объектом в родительской таблице payment\_type, иначе происходит потеря важной для системы информации. |

**Ход работы**

1. Создание базы данных:

CREATE DATABASE gardening;

1. Создание таблиц:

CREATE TABLE plot (

id int4 GENERATED ALWAYS AS IDENTITY NOT NULL,

line int4 NOT NULL,

"number" int4 NOT NULL,

square float4 NOT NULL,

price int4 NOT NULL,

CONSTRAINT plot\_pk PRIMARY KEY (id)

);

CREATE TABLE building\_type (

id int4 GENERATED ALWAYS AS IDENTITY NOT NULL,

"name" varchar(100) NOT NULL,

CONSTRAINT building\_type\_pk PRIMARY KEY (id)

);

CREATE TABLE "owner" (

id int4 GENERATED ALWAYS AS IDENTITY NOT NULL,

surname varchar(100) NOT NULL,

first\_name varchar(100) NOT NULL,

last\_name varchar(100) NOT NULL,

CONSTRAINT owner\_pk PRIMARY KEY (id)

);

CREATE TABLE payment\_type (

id int4 GENERATED ALWAYS AS IDENTITY NOT NULL,

purpose varchar(100) NOT NULL,

CONSTRAINT payment\_type\_pk PRIMARY KEY (id)

);

CREATE TABLE building (

id int4 GENERATED ALWAYS AS IDENTITY NOT NULL,

"name" varchar(100) NOT NULL,

plot\_id int4 NOT NULL,

type\_id int4 NOT NULL,

CONSTRAINT building\_pk PRIMARY KEY (id),

CONSTRAINT building\_building\_type\_fk FOREIGN KEY (type\_id) REFERENCES building\_type(id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT building\_plot\_fk FOREIGN KEY (plot\_id) REFERENCES plot(id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE plot\_owner\_link (

plot\_id int4 NOT NULL,

owner\_id int4 NOT NULL,

CONSTRAINT plot\_owner\_link\_pk PRIMARY KEY (plot\_id, owner\_id),

CONSTRAINT plot\_owner\_link\_owner\_fk FOREIGN KEY (owner\_id) REFERENCES "owner"(id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT plot\_owner\_link\_plot\_fk FOREIGN KEY (plot\_id) REFERENCES plot(id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE payment (

id int4 GENERATED ALWAYS AS IDENTITY NOT NULL,

sum int4 NOT NULL,

date\_time timestamp NOT NULL,

type\_id int4 NOT NULL,

owner\_id int4 NOT NULL,

CONSTRAINT payment\_pk PRIMARY KEY (id),

CONSTRAINT payment\_owner\_fk FOREIGN KEY (owner\_id) REFERENCES "owner"(id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT payment\_payment\_type\_fk FOREIGN KEY (type\_id) REFERENCES payment\_type(id) ON DELETE RESTRICT ON UPDATE CASCADE

);

1. Удаление и добавление столбца командой ALTER TABLE

ALTER TABLE public.payment\_type DROP COLUMN purpose;

ALTER TABLE public.payment\_type ADD purpose varchar(100) NOT NULL;

**Выводы об особенностях создания таблиц разработанной модели данных в выбранной СУБД.**

В СУБД Postgresql отсутствует тип DateTime, поэтому используется тип timestamp.

Создана база данных: таблицы, внешние ограничения. Продемонстрированы навыки использования команды ALTER TABLE для удаления и добавления столбца.