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

по курсу: «Проектирование баз данных»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № Z1431		М. Д. Быстров
	подпись, дата	инициалы, фамилия

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

   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

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

Физическая модель БД для СУБД Sqlite

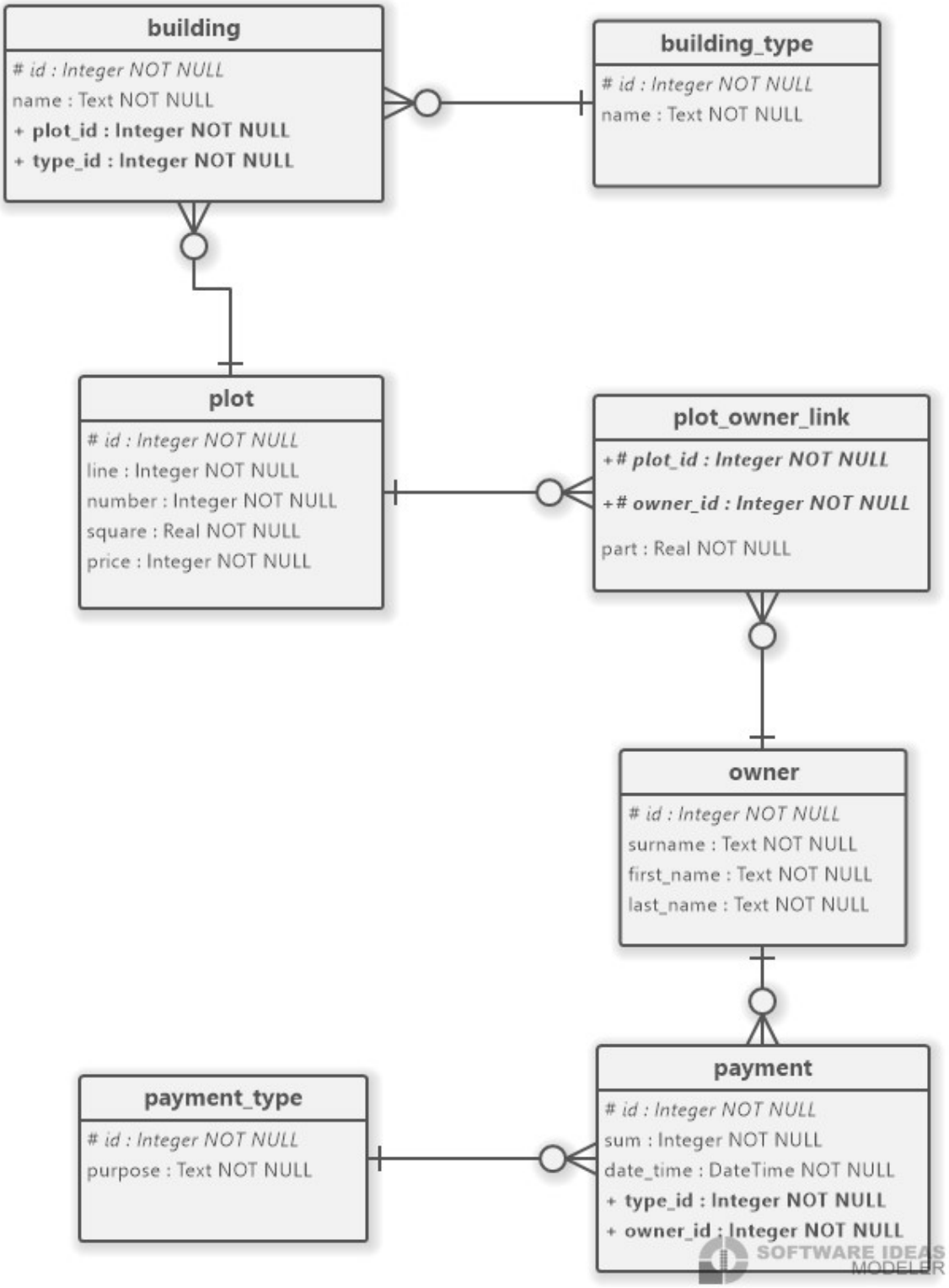

Таблица с описанием ссылочной целостности

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 owner_id)	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 owner_id)	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;
```

2. Создание таблиц:

```
CREATE TABLE public.plot (
	id int GENERATED ALWAYS AS IDENTITY NOT NULL,
	line int NOT NULL,
	"number" int NOT NULL,
	square real NOT NULL,
	price int NOT NULL,
	CONSTRAINT plot_pk PRIMARY KEY (id)
);

CREATE TABLE public.building (
	id int GENERATED ALWAYS AS IDENTITY NOT NULL,
	"name" text NOT NULL,
	plot_id int NOT NULL,
	type_id int NOT NULL,
	CONSTRAINT building_pk PRIMARY KEY (id)
);

CREATE TABLE public.building_type (
	id int GENERATED ALWAYS AS IDENTITY NOT NULL,
	"name" text NOT NULL,
```

```
    CONSTRAINT building_type_pk PRIMARY KEY (id)
);

CREATE TABLE public."owner" (
    id int GENERATED ALWAYS AS IDENTITY NOT NULL,
    surname text NOT NULL,
    first_name text NOT NULL,
    last_name text NOT NULL,
    CONSTRAINT owner_pk PRIMARY KEY (id)
);

CREATE TABLE public.plot_owner_link (
    plot_id int NOT NULL,
    owner_id int NOT NULL,
    part real NOT NULL,
    CONSTRAINT plot_owner_link_pk PRIMARY KEY (plot_id,owner_id)
);

CREATE TABLE public.payment (
    id int GENERATED ALWAYS AS IDENTITY NOT NULL,
    sum int NOT NULL,
    date_time timestamp NOT NULL,
    type_id int NOT NULL,
    owner_id int NOT NULL,
    CONSTRAINT payment_pk PRIMARY KEY (id)
```

```
);


CREATE TABLE public.payment_type (
	id int GENERATED ALWAYS AS IDENTITY NOT NULL,
	purpose text NOT NULL,
	CONSTRAINT payment_type_pk PRIMARY KEY (id)
);
```

3. Создание внешних ограничений

```
ALTER TABLE public.building ADD CONSTRAINT building_plot_fk FOREIGN KEY (plot_id) REFERENCES public.plot(id) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE public.building ADD CONSTRAINT building_building_type_fk FOREIGN KEY (type_id) REFERENCES public.building_type(id) ON DELETE RESTRICT ON UPDATE CASCADE;


ALTER TABLE public.plot_owner_link ADD CONSTRAINT plot_owner_link_plot_fk FOREIGN KEY (plot_id) REFERENCES public.plot(id) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE public.plot_owner_link ADD CONSTRAINT plot_owner_link_owner_fk FOREIGN KEY (owner_id) REFERENCES public."owner"(id) ON DELETE RESTRICT ON UPDATE CASCADE;


ALTER TABLE public.payment ADD CONSTRAINT payment_payment_type_fk FOREIGN KEY (type_id) REFERENCES public.payment_type(id) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE public.payment ADD CONSTRAINT payment_owner_fk FOREIGN KEY (owner_id) REFERENCES public."owner"(id) ON DELETE RESTRICT ON UPDATE CASCADE;
```

4. Удаление и добавление столбца командой ALTER TABLE

ALTER TABLE public.payment_type DROP COLUMN purpose;

ALTER TABLE public.payment_type ADD purpose text 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.