**МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВОСИБИРСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«НОВОСИБИРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

СПЕЦИАЛЬНОСТЬ 09.02.03

ПРОГРАММИРОВАНИЕ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Курсовая работа

по Проектированию базы данных

тема: Проектирование базы данных для кинотеатра

Выполнил: Лебедев Никита

Группа: 120 ПКС

Руководитель: Кондюрин В. А.

г. Новосибирск 2023 г

Оглавление

[Введение 3](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179020)

[Разработка схемы базы данных 4](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179021)

[2.Создание таблиц 5](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179022)

[2.1. Таблица «Point» 5](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179023)

[2.2. Таблица «Meet» 6](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179024)

[2.3. Таблица «Events» 7](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179025)

[2.4. Таблица «PlanToday» 8](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179026)

[2.5. Таблица «Client» 9](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179027)

[3. Создание ER-диаграммы 10](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179028)

[4. Создание представлений 11](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179029)

[4.1. Представление «Meet-client» 11](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179030)

[4.2. Представление «PlanToday-Meet-client» 11](file:///C:\Users\nik-l\Desktop\Курсовая%20БД%20Надеин%20120ПКС.docx#_Toc132179031)

# Введение

Базы данных являются неотъемлемой частью современных информационных технологий и играют важную роль в управлении большим объемом информации в различных сферах деятельности. Web приложение - это одна из таких сфер, где базы данных могут значительно улучшить и оптимизировать его работу.

Основные понятия:

* **SQL -** декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.
* **Реляционная база данных -** это набор данных с предопределенными связями между ними.
* **PostgreSQL -** свободная объектно-реляционная система управления базами данных.
* **База данных -** совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

**Цель курсовой:** разработка базы данных для приложения, которая позволит хранить и обрабатывать информацию о встречах, событиях, планах и долгосрочных целях.

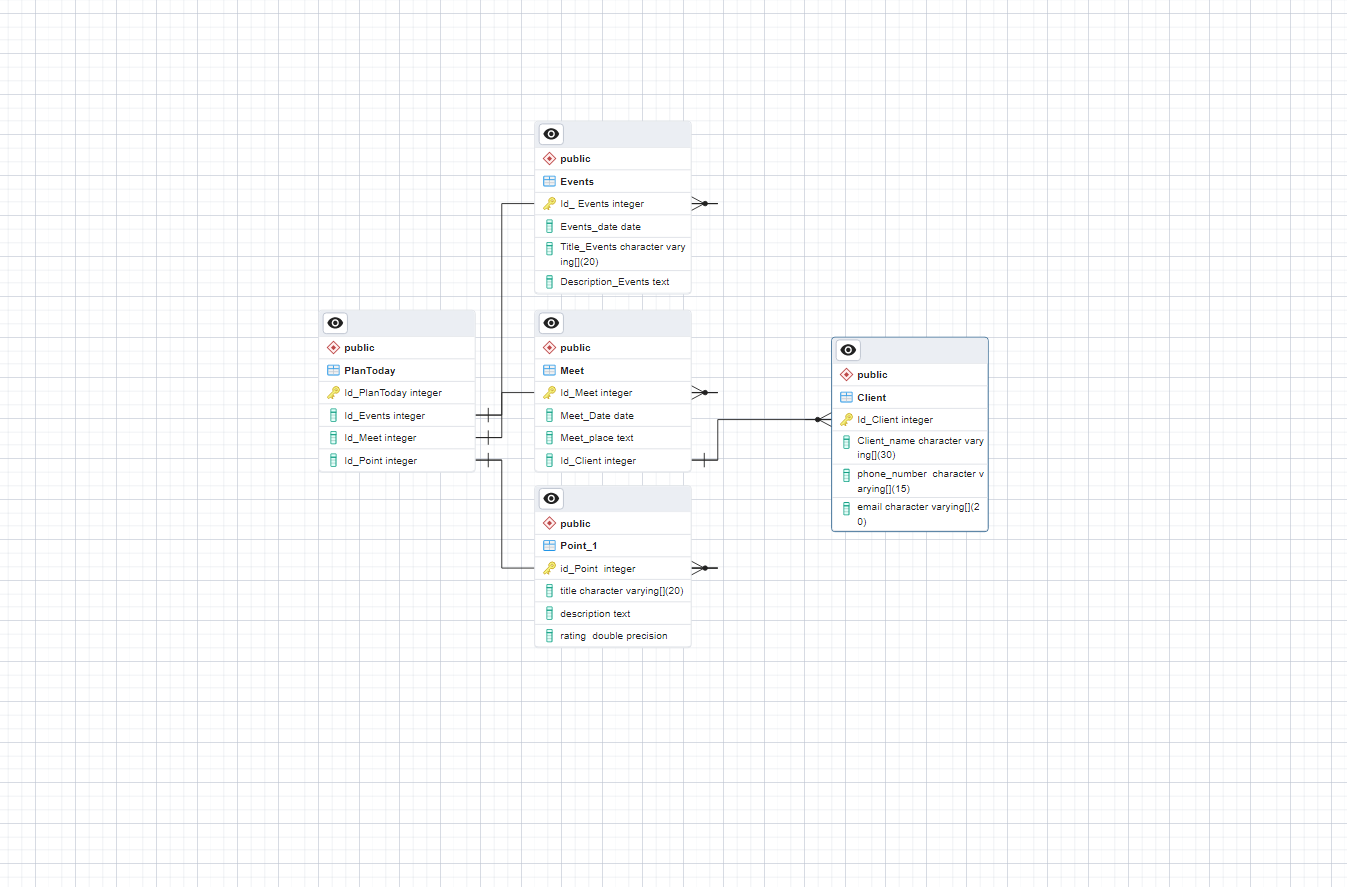
Для достижения поставленной цели необходимо решить следующие задачи:

* Разработать предварительную схему БД;
* Создание таблиц на основе разработанной схеме;
* Создание ER-диаграммы БД;
* Выполнить несколько запросов для получения представлений;

# Разработка схемы базы данных

Схема разрабатывалась с помощью *PostgreSQL*. В схеме описаны сущности и столбцы, которые их описывают, также отмечены основные (первичные) и внешние ключи, типы данных. Преимущественно в базе данных использовались связи один к одному и один ко многим. С помощью этого можно будет создать таблицы и связи между ними.

*Рис.1(схема базы данных)*



# 2.Создание таблиц

# 2.1. Таблица «Point»

Таблица предназначена для хранения информация о долгосрочных целях

Таблица 1. Таблица « Point »

|  |  |
| --- | --- |
| Id\_Point | Идентификатор |
| title | Краткое название |
| description | Описание |
| rating | Степень важности |

Скрипт:

CREATE TABLE IF NOT EXISTS public."Point"

(

id\_Point integer NOT NULL,

title character varying(20) COLLATE pg\_catalog."default",

description text COLLATE pg\_catalog."default",

rating double precision,

CONSTRAINT " Point\_pkey" PRIMARY KEY (id\_Point)

)

ALTER TABLE IF EXISTS public." Point "

OWNER to postgres;

# 

# 2.2. Таблица «Meet»

Таблица предназначена для хранения информации о запланированных встречах с клиентом

Таблица 2. Таблица « Meet »

|  |  |
| --- | --- |
| Id\_Meet | Идентификатор |
| Meet\_Date | Дата встречи |
| Meet\_place | Место встречи |
| Id\_Client | Идентификатор |

**Скрипт:** CREATE TABLE IF NOT EXISTS public." Meet "

(

Id\_ Meet integer NOT NULL,

Meet\_Date date,

Meet\_place text COLLATE pg\_catalog."default",

id\_Client integer,

CONSTRAINT " Meet\_pkey" PRIMARY KEY (id\_ Meet)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public." Meet "

OWNER to postgres;

# 2.3. Таблица «Events»

Таблица предназначена для хранения информации о запланированных событиях

Таблица 3. Таблица «Events»

|  |  |
| --- | --- |
| Id\_ Events | Идентификатор |
| Events\_date | Дата сеанса |
| Title\_Events | Краткое название |
| Description\_Events | Описание |

**Скрипт:** CREATE TABLE IF NOT EXISTS public." Events "

(

id\_Events integer NOT NULL,

Events\_date date,

title\_Events character varying(20) COLLATE pg\_catalog."default",

description\_Events text COLLATE pg\_catalog."default",

CONSTRAINT " Events\_pkey " PRIMARY KEY (id\_Events)

)

TABLESPACE pg\_default;

# 2.4. Таблица «PlanToday»

Таблица предназначена для хранения информации о запланированых мероприятиях, целях и встречах на сегодня.

Таблица 4. Таблица « PlanToday »

|  |  |
| --- | --- |
| Id\_PlanToday | Идентификатор |
| Id\_Events | Идентификатор |
| Id\_Meet | Идентификатор |
| Id\_Point | Идентификатор |

**Скрипт:** CREATE TABLE IF NOT EXISTS public." PlanToday "

(

id\_PlanToday integer NOT NULL,

id\_Events integer,

id\_Meet integer,

id\_Point integer,

CONSTRAINT " PlanToday\_pkey" PRIMARY KEY (id\_PlanToday)

)

TABLESPACE pg\_default;

# 

# 2.5. Таблица «Client»

Таблица предназначена для хранения информации о клиентах.

Таблица 5. Таблица « Client »

|  |  |
| --- | --- |
| Id\_Client | Идентификатор |
| Client\_name | ФИО клинта |
| phone\_number | Номер телефона |
| email | Электронная почта |

**Скрипт:** CREATE TABLE IF NOT EXISTS public." Client "

(

id\_ Client integer NOT NULL,

Client\_name character varying(30) COLLATE pg\_catalog."default",

phone\_number character varying(15) COLLATE pg\_catalog."default",

email character varying(20) COLLATE pg\_catalog."default",

CONSTRAINT " Client\_pkey" PRIMARY KEY (id\_Client)

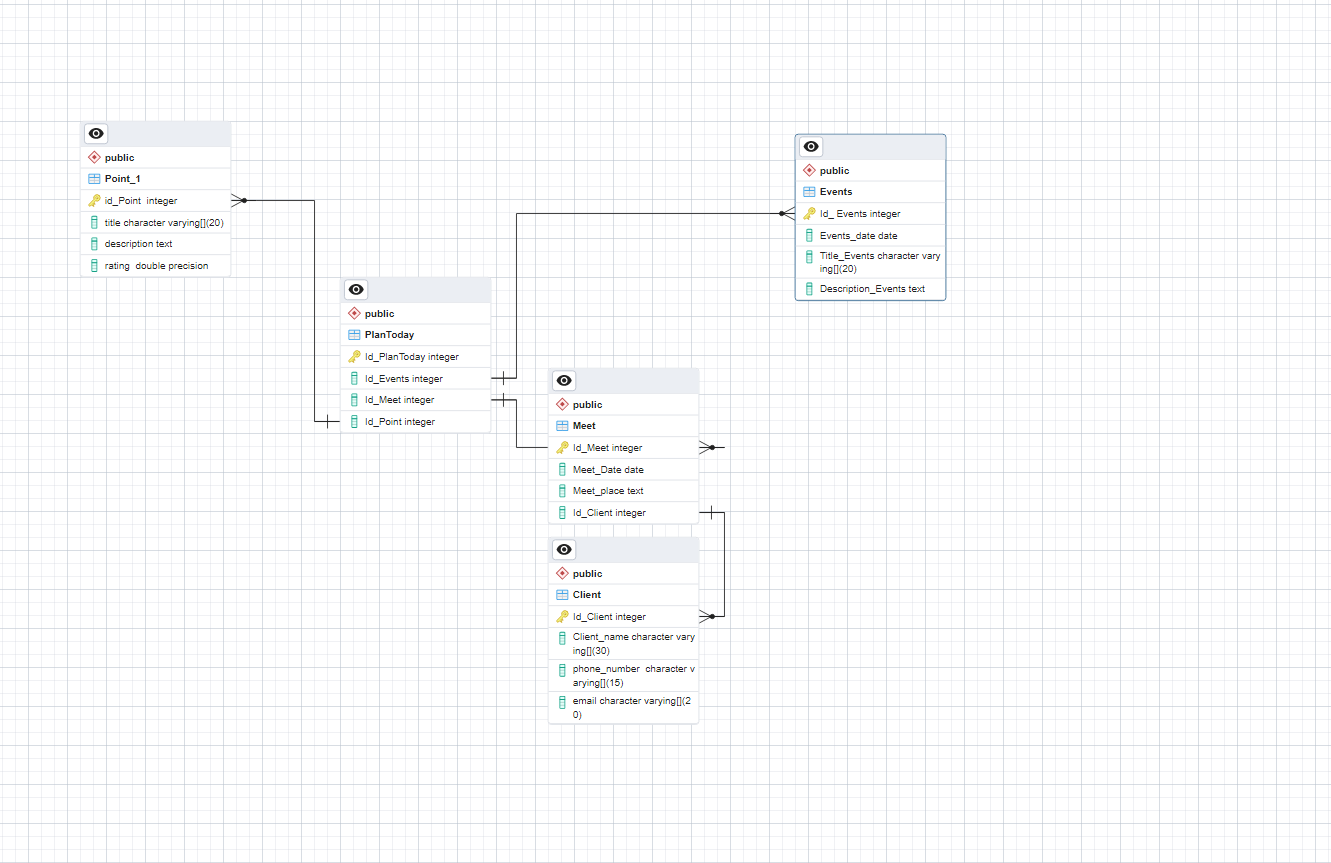
)

TABLESPACE pg\_default;

# 3. Создание ER-диаграммы

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы.

**Сущность** (таблица, отношение) — это представление набора реальных или абстрактных объектов (людей, вещей, мест, событий, идей, комбинаций и т. д.), которые можно выделить в одну группу, потому что они имеют одинаковые характеристики и могут принимать участие в похожих связях.

Рис.2(ER-диаграмма)

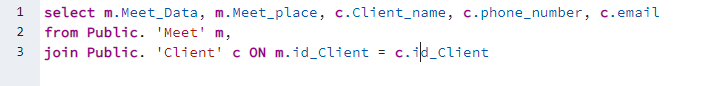
# 4. Создание представлений

***Представление в базе данных*** – это виртуальная таблица, содержимое которой определяется запросом. Как и таблица, представление состоит из ряда именованных столбцов и строк данных.

Представление выполняет функцию фильтра базовых таблиц, на которые оно ссылается. Определяющий представление запрос может быть инициирован в одной или нескольких таблицах или в других представлениях текущей или других баз данных.

# 4.1. Представление «Meet - Client»

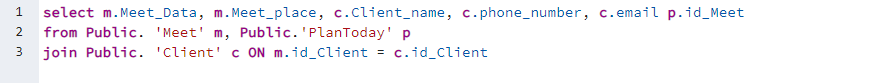
Представление показывает информацию о встрече и клиенте при вводе id клиента.



*Рис.3(Представление «meet-client»)*

# 4.2. Представление «PlanToday - Meet - Client»

Представление показывает какая встреча с каким клиентом запланирована.

*Рис.4(Представление «PlanToday-meet-client»)*