

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Базы данных

Тема «Сеть клубов карате»
Лабораторная работа №2
Создание реляционной схемы данных

Студент:
Преподаватель:

Р.С. Кочеров
Д.В. Куприянова

Минск 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ «СЕТЬ КЛУБОВ КАРАТЕ»	4
1.1 Описание таблицы и сущностей	4
1.2 Типы связей.....	4
2 СХЕМА ER-МОДЕЛИ	6
3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ	10
ЗАКЛЮЧЕНИЕ.....	14

ВВЕДЕНИЕ

В современном бизнесе эффективное управление данными является ключевым аспектом для обеспечения конкурентоспособности и принятия решений на основе точной информации. Информационные системы, построенные на основе реляционных баз данных, позволяют структурировать, хранить и обрабатывать большие объемы данных, что особенно актуально для компаний розничной торговли. PostgreSQL, как одна из наиболее популярных и надежных систем управления базами данных, предоставляет широкие возможности для реализации таких систем.

Целью данной лабораторной работы является преобразование ER-модели «Сеть клубов карате», разработанной ранее, в реляционную схему данных с использованием PostgreSQL. В процессе работы будут выполнены следующие задачи: преобразование сущностей и атрибутов ER-диаграммы в таблицы и столбцы, определение первичных и внешних ключей, реализация связей между таблицами, а также нормализация данных для устранения избыточности и обеспечения целостности.

Результатом работы станет реляционная схема данных, которая может быть использована для создания базы данных «Сеть клубов карате» в PostgreSQL. Эта схема будет включать таблицы для клубов, тренеров, учеников, тренировок, мероприятий и мест проведения мероприятий, а также промежуточные таблицы для реализации связей многие-ко-многим.

1 ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ «СЕТЬ КЛУБОВ КАРАТЕ»

Исходное задание: создание реляционной схемы данных, преобразовать ER-диаграмму в реляционную модель данных.

1.1 Описание таблицы и сущностей

В рамках реляционной модели «Сеть клубов карате» предусмотрены следующие таблицы и соответствующие им сущности.

Таблица Клуб соответствует сущности клуба и включает поля: номер клуба (уникальный идентификатор), название, дата основания, почта, стиль карате.

Таблица Тренер соответствует сущности тренера и включает поля: номер тренера (уникальный идентификатор), ФИО, заработная плата, стаж работы, контактный телефон, уровень мастерства.

Таблица Оплата соответствует сущности оплаты и включает поля: номер оплаты (уникальный идентификатор), почта, сумма, тип оплаты, номер ЕРИП.

Таблица Мероприятие соответствует сущности мероприятия и включает поля: номер мероприятия (уникальный идентификатор), название, тип, дата, уровень, спонсор.

Таблица Место проведения мероприятия соответствует сущности места проведения мероприятия и включает поля: номер места (уникальный идентификатор), цена, адрес, вместительность, площадь.

Таблица Ученик соответствует сущности ученика и включает поля: номер ученика (уникальный идентификатор), ФИО, возраст, вес, уровень мастерства.

Таблица Тренировка соответствует сущности тренировки и включает поля: номер тренировки (уникальный идентификатор), тип, адрес зала, дата и время, группа.

Для реализации связи многие-ко-многим предусмотрена промежуточная таблица Тренировка ученика. Она содержит поля: номер ученика (ссылка на ученика, как внешний ключ) и номер тренировки (ссылка на тренировку, как внешний ключ).

Для реализации связи многие-ко-многим предусмотрена промежуточная таблица Мероприятие ученика. Она содержит поля: номер ученика (ссылка на ученика, как внешний ключ) и номер мероприятия (ссылка на мероприятие, как внешний ключ).

1.2 Типы связей

Для описания взаимосвязей между объектами были выделены следующие связи.

1. Связь «Клуб – Тренер» (один-ко-многим): каждый клуб может иметь нескольких тренеров, но каждый тренер прикреплен только к одному клубу.
2. Связь «Клуб – Мероприятие» (один-ко-многим): в каждом клубе могут проводиться различные мероприятия, но каждое мероприятие относится только к одному клубу.
3. Связь «Тренер – Тренировки» (один-ко-многим): каждый тренер может проводить несколько тренировок, но каждая тренировка проводится только одним тренером.
4. Связь «Тренер – Ученик» (один-ко-многим): у тренера может быть несколько учеников, но у каждого ученика может быть только один основной тренер.
5. Связь «Мероприятие – Ученик» (многие-ко-многим): ученик может участвовать в нескольких мероприятиях, а в одном мероприятии могут участвовать сразу несколько учеников.
6. Связь «Мероприятие – Место проведения мероприятия» (многие-к-одному): одно место может использоваться для проведения нескольких мероприятий, но каждое мероприятие проходит в одном конкретном месте.
7. Связь «Ученик – Тренировки» (многие-ко-многим): ученик может посещать разные тренировки, а на одной тренировке могут присутствовать сразу несколько учеников.
8. Связь «Ученик – Оплата» (один-к-одному): за каждым учеником закреплена своя информация об оплате занятий.

2 СХЕМА ER-МОДЕЛИ

Схема ER-диаграммы представлена на рисунке 2.1.

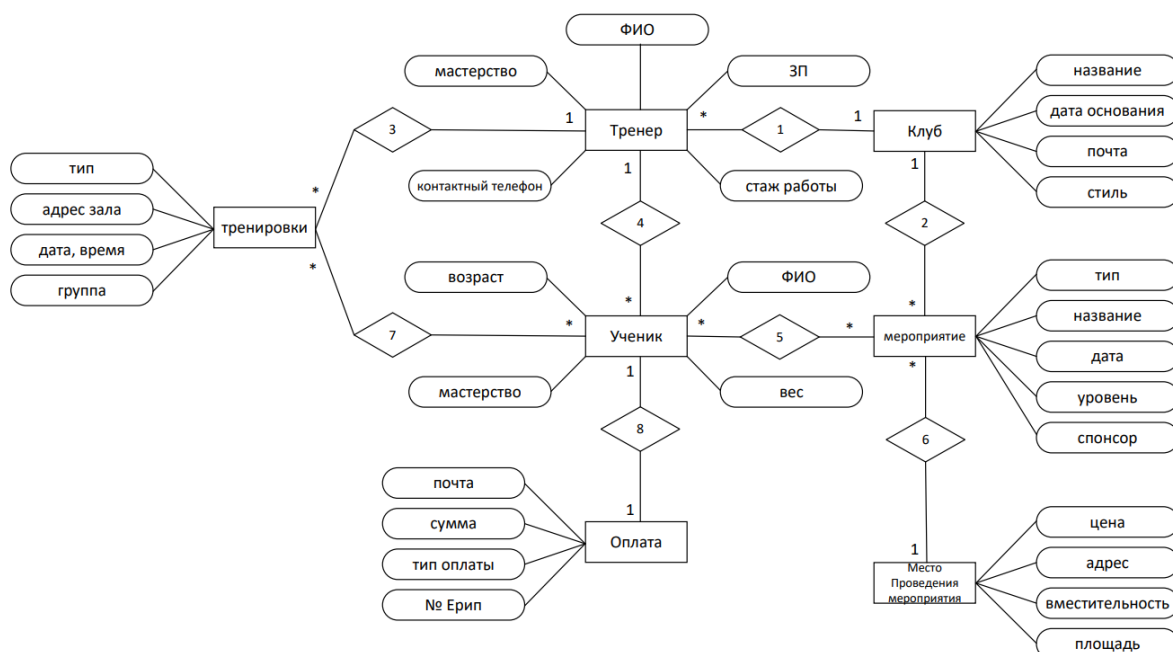


Рисунок 2.1 – ER-диаграмма

Порядок перевода ER-модели в реляционную модель выполняется с помощью алгоритма, состоящего из пяти шагов:

1. Каждый объект на ER-диаграмме превращается в реляционное отношение (далее для краткости – таблицу), имя объекта становится именем таблицы. Можно выделить семь таблиц со следующими именами: «Клуб», «Тренер», «Ученик», «Тренировки», «Оплата», «Мероприятие», «Место проведения мероприятия».

2. Каждый атрибут объекта становится столбцом с тем же именем.

3. Уникальные атрибуты объекта превращаются в первичный ключ таблицы. Таким образом были добавлены следующие первичные ключи: номер клуба, номер тренера, номер ученика, номер тренировки, номер оплаты, номер мероприятия и номер места проведения мероприятия. Алгоритм выполнения шагов 1-3 представлено на рисунке 2.2.

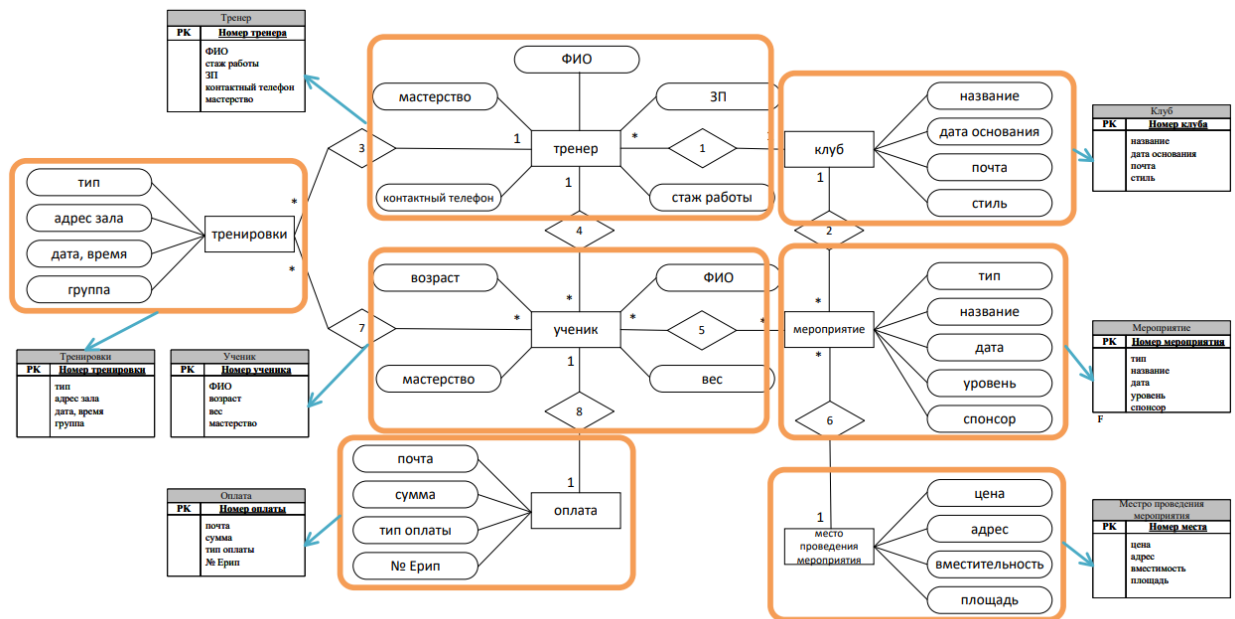


Рисунок 2.2 – Выполнение шагов 1-3

4. Связи «один-ко-многим» становятся ссылками в уже существующих таблицах, при этом внешний ключ добавляется в виде столбца в таблицу, соответствующую объекту со стороны «многие» связи. Внешние ключи ссылаются на первичные ключи целевых таблиц. Пример связи «один-ко-многим» представлен на рисунке 2.3.

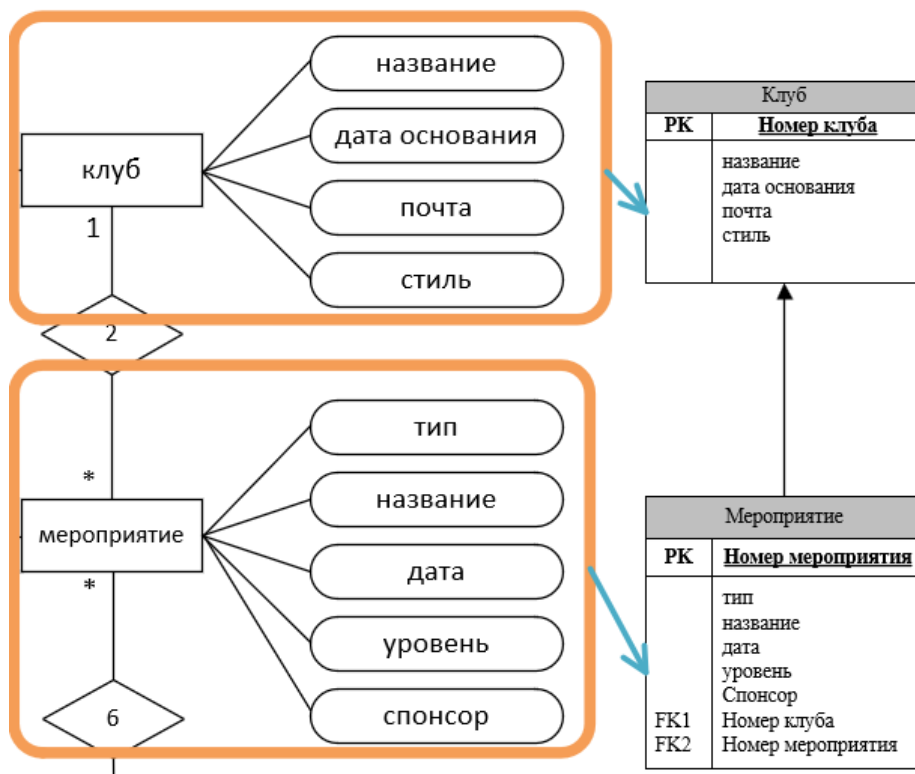


Рисунок 2.3 – Связь «один-ко-многим»

5. Связи «многие-ко-многим» реализуются через отдельную таблицу.

Была создана таблица «Тренировка ученика», в которой находятся два поля внешних ключей: «номер ученика» и «номер тренировки».

Была создана таблица «Мероприятие ученика», в которой находятся два поля внешних ключей: «номер ученика» и «номер мероприятия».

Пример связи «многие-ко-многим» представлен на рисунке 2.4.

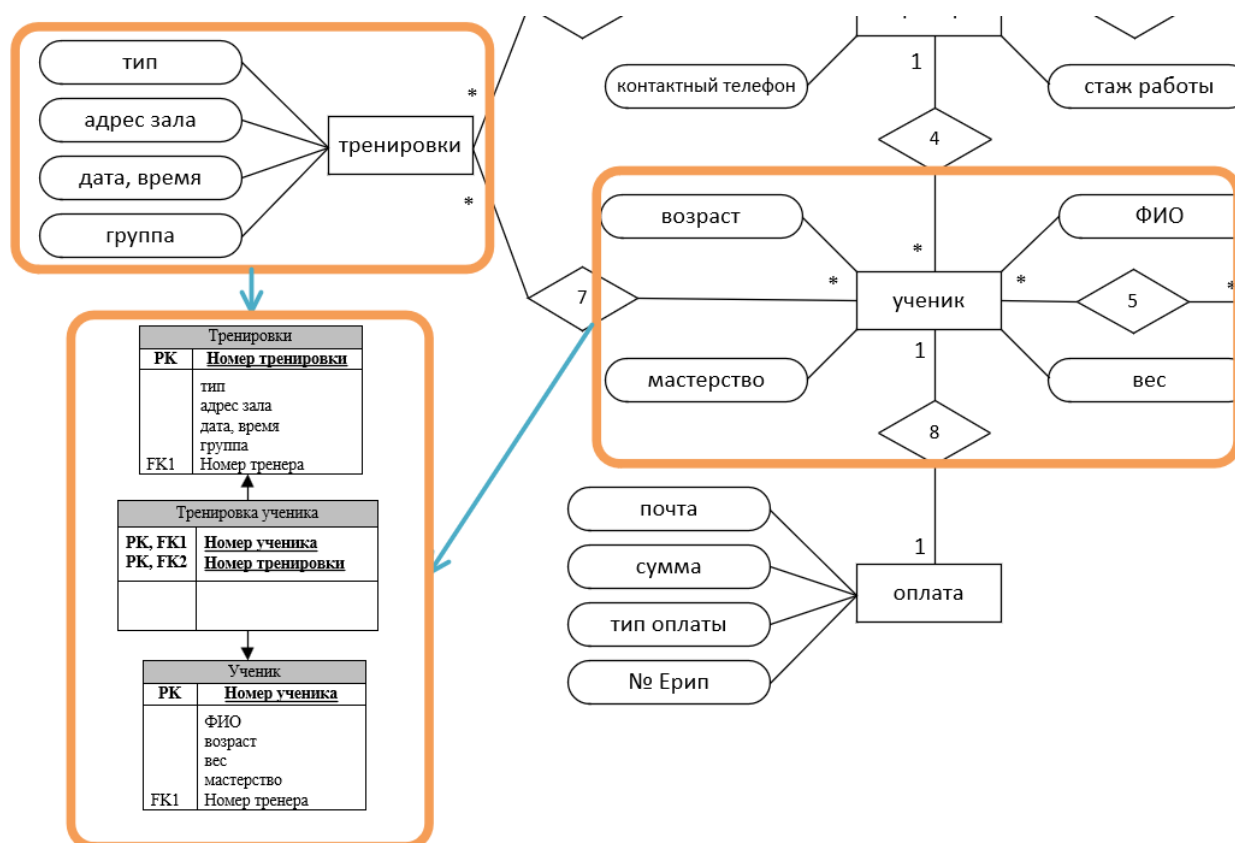


Рисунок 2.4 – Связь «многие-ко-многим»

Данные 5 шагов позволяют создать UML-диаграмму – реляционную схему данных «бумажного» преобразования, которая представлена на рисунке 2.5.

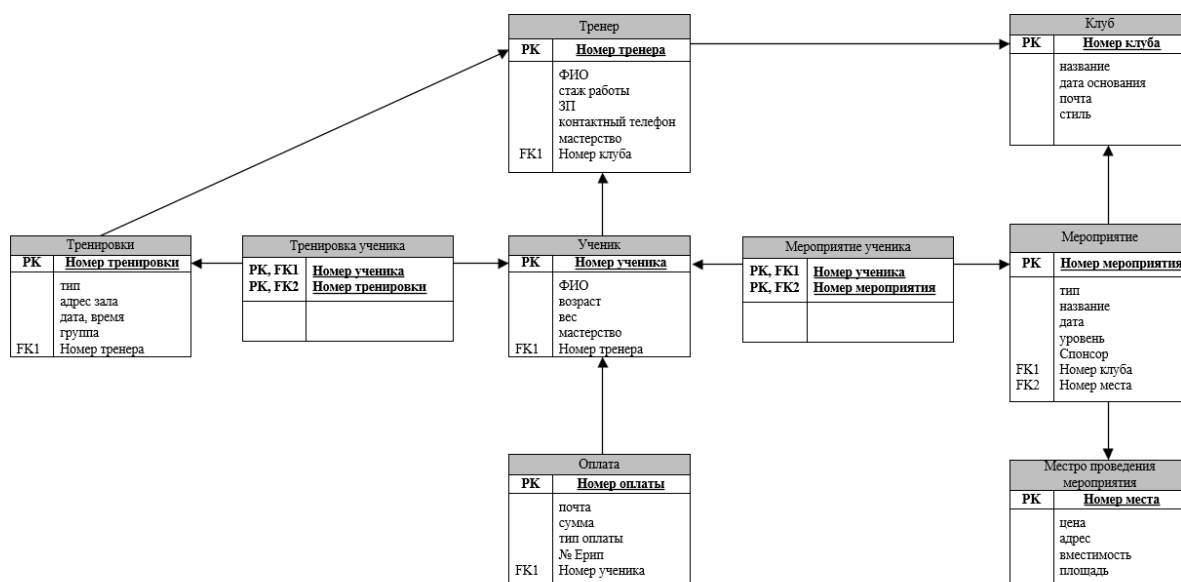


Рисунок 2.5 – UML-диаграмма

3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ

Для перевода ER-диаграммы в реляционную диаграмму используется pgAdmin4. Для проведения операций были выполнены следующие шаги:

1. Открыть pgAdmin4.
2. Создать новую базу данных. Создание базы данных представлено на рисунке 3.1.

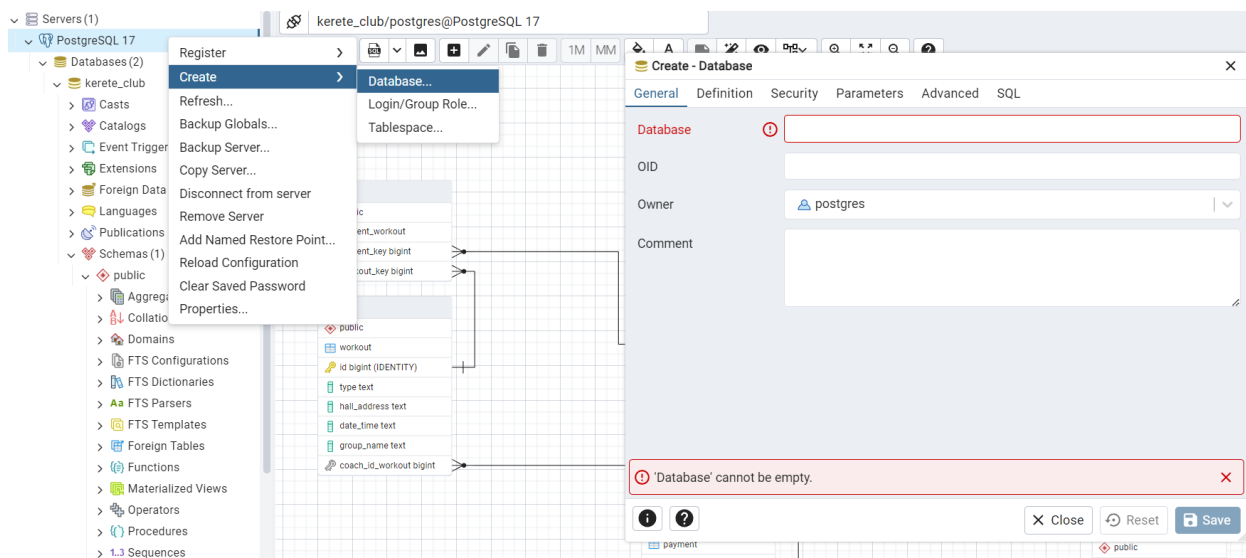


Рисунок 3.1 – Меню создание базы данных

3. Создать таблицы, в которых отсутствуют внешние ключи. Создание таблицы представлено на рисунке 3.2.

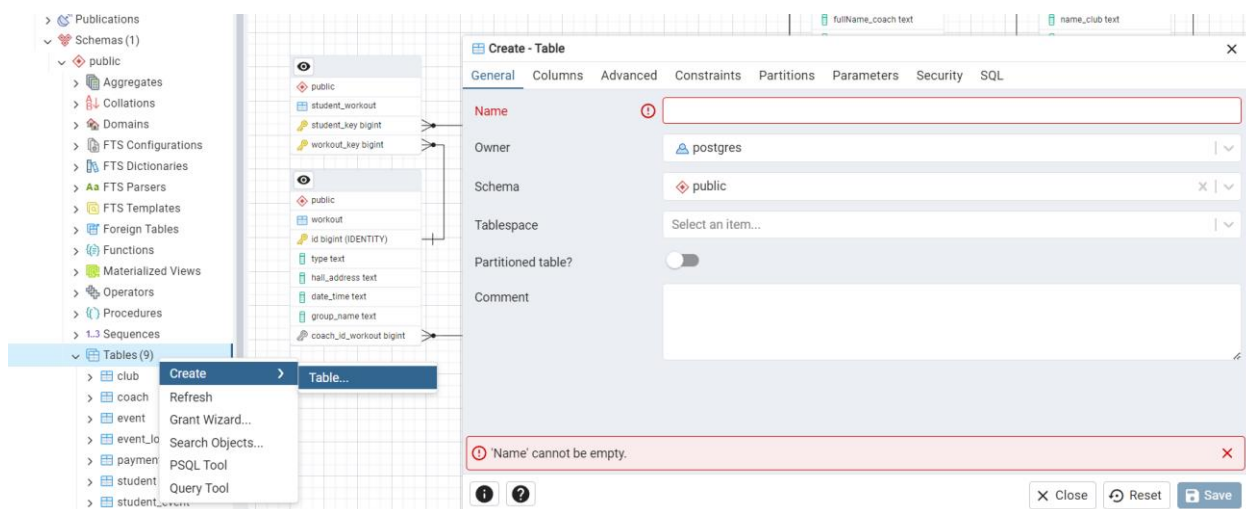


Рисунок 3.2 – Меню создание таблицы

4. Заполнить колонны таблиц. Заполнение колонны таблицы представлено на рисунке 3.3.

club

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	name_club	text			<input type="checkbox"/>	<input type="checkbox"/>	
	foundation_data	text			<input type="checkbox"/>	<input type="checkbox"/>	
	email_club	text			<input type="checkbox"/>	<input type="checkbox"/>	
	karateStyle	text			<input type="checkbox"/>	<input type="checkbox"/>	

Close
Reset
Save

Рисунок 3.3 – Меню заполнение колонны таблицы

4. Создать внешние ключи для связей таблиц. Создание внешних ключей представлено на рисунке 3.4.

coach

General Columns Advanced Constraints Partitions Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
club_fkey_coach	(club_id_coach) -> (id)	public.club

General Definition Columns Action

Local	Referenced	Referenced Table
club_id_coach	id	public.club

Close
Reset
Save

Рисунок 3.4 – Меню создания внешних ключей

5. В графическом редакторе выбрать базу данных, нажать на неё правой кнопкой мыши и в появившемся меню выбрать «ERD for table» (рис. 3.1). Далее pgAdmin4 представит реляционную схему данных. «Автоматическая» реляционная схема данных представлена на рисунке 3.5.

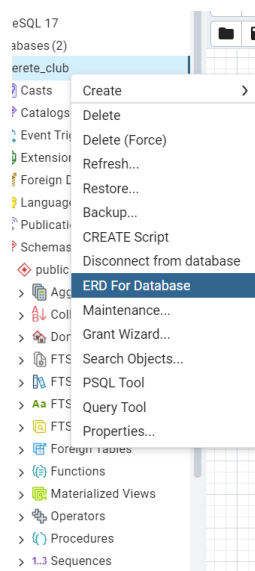


Рисунок 3.5 – Генерация реляционной схемы

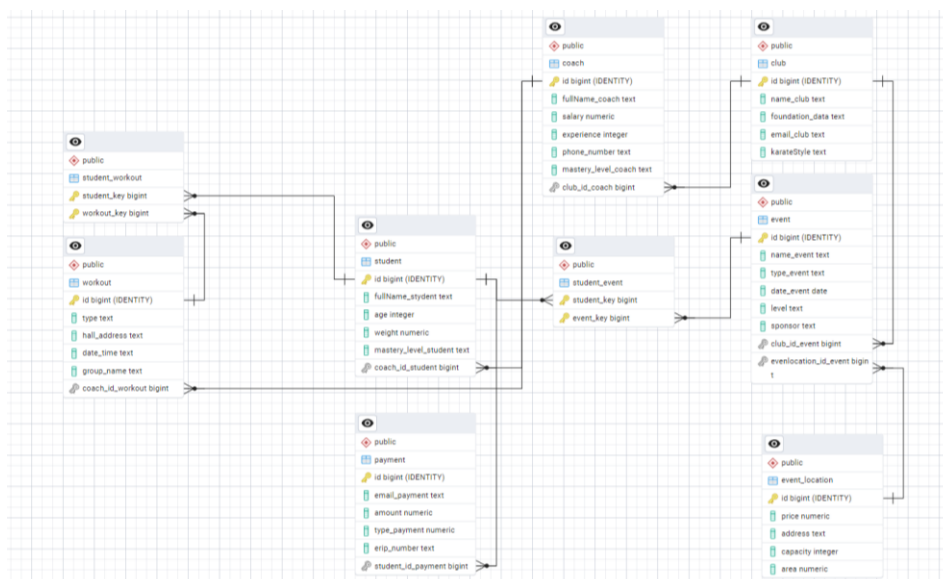


Рисунок 3.6 – ERD-диаграмма

Полученную реляционную модель можно экспортировать в файл формата .rgerd. При необходимости есть возможность восстановить код открыв этот файл в pgAdmin4 с помощью ERD Tool и сгенерировать код. Генерация кода показана на рисунке 3.7.

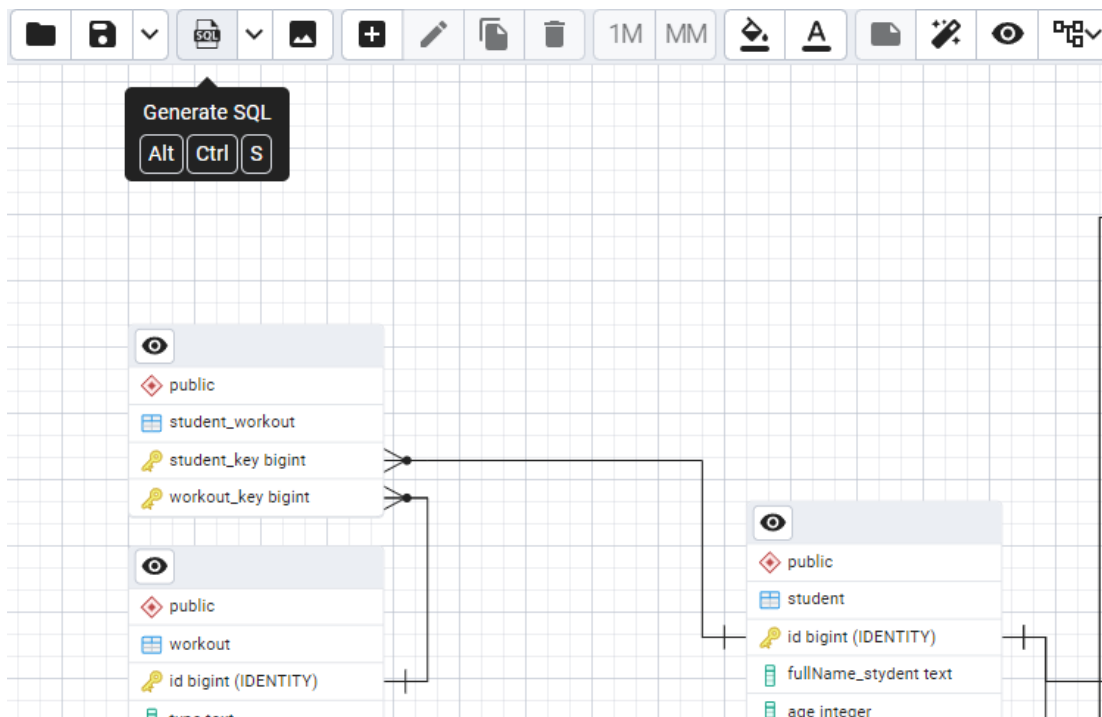


Рисунок 3.7 – Генерация кода

На рисунке 3.8 представлен SQL код для создания реляционной модели.

```

1  |-- This script was generated by the ERD tool in pgAdmin 4.
2  -- Please log an issue at https://github.com/pgadmin-org/pgadmin4/issues/new/choose if you find
3  BEGIN;
4
5
6  CREATE TABLE IF NOT EXISTS public.club
7  (
8      id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 92
9      name_club text COLLATE pg_catalog."default",
10     foundation_data text COLLATE pg_catalog."default",
11     email_club text COLLATE pg_catalog."default",
12     "karateStyle" text COLLATE pg_catalog."default",
13     CONSTRAINT club_pkey PRIMARY KEY (id)
14 );
15
16 COMMENT ON TABLE public.club
17     IS 'клубы по карате';
18
19 CREATE TABLE IF NOT EXISTS public.coach
20 (
21     id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 92
22     "fullName_coach" text COLLATE pg_catalog."default",
23     salary numeric,
24     experience integer,
25     phone_number text COLLATE pg_catalog."default",
26     mastery_level_coach text COLLATE pg_catalog."default",
27     club_id_coach bigint,
28     CONSTRAINT coach_pkey PRIMARY KEY (id)
29 );
30

```

Рисунок 3.8 – SQL код

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы была успешно преобразована ER-модель «Сеть клубов карате» в реляционную схему данных с использованием PostgreSQL. Были созданы таблицы, соответствующие сущностям ER-диаграммы, определены первичные и внешние ключи, а также реализованы связи между таблицами. В процессе работы были рассмотрены два варианта преобразования: «бумажный» и «автоматизированный».

Разработанная реляционная схема данных в дальнейшем будет использована для создания базы данных «Сеть клубов карате» в PostgreSQL. Эта база данных обеспечит эффективное управление информацией о клубе, тренерском составе, мероприятиях, тренировках и учениках, что в свою очередь позволит повысить качество обслуживания и эффективность работы сети клубов по карате.