Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Военный факультет

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

Тема «Сеть клубов карате»

Лабораторная работа №3

Реализация SQL-запросов для создания базы данных

Студент: Р.С. Кочеров

Преподаватель: Д.В. Куприянова

МИНСК 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc189596220)

[1 СОЗДАНИЕ ТАБЛИЦ 4](#_Toc189596221)

[2 НАПОЛНЕНИЕ ТАБЛИЦ 8](#_Toc189596222)

[3 ДРУГИЕ SQL-КОМАНДЫ 17](#_Toc189596223)

[4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ 18](#_Toc189596224)

[5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ 19](#_Toc189596225)

[6 ЭКСПОРТ РЕЗУЛЬТАТОВ 21](#_Toc189596226)

[ЗАКЛЮЧЕНИЕ 45](#_Toc189596227)

# ВВЕДЕНИЕ

Данная лабораторная работа направлена на углубленное изучение проектирования, создания и управления реляционными базами данных с использованием языка SQL. В ходе работы вы получите опыт работы с основами архитектуры баз данных, включая разработку схемы данных, проектирование таблиц и определение их структуры. Особое внимание будет уделено установлению связей между таблицами с помощью первичных и внешних ключей, а также внедрению бизнес-правил, обеспечивающих целостность и корректность данных.

Работа включает практические задания по заполнению таблиц данными как с помощью SQL-скриптов, так и через интерфейс системы управления базами данных (СУБД). Вы также освоите использование оператора ALTER TABLE для внесения изменений в структуру таблиц, создания и удаления временных таблиц, а также экспорта результатов в SQL-скрипт для дальнейшего анализа и сравнения. Успешное выполнение всех этапов работы позволит закрепить теоретические знания и развить практические навыки, необходимые для работы с реляционными базами данных в реальных условиях.

# СОЗДАНИЕ ТАБЛИЦ

Ниже приведены SQL-запросы для создания всех таблиц по варианту:

BEGIN;

CREATE TABLE IF NOT EXISTS public.club

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

name\_club text COLLATE pg\_catalog."default",

foundation\_data text COLLATE pg\_catalog."default",

email\_club text COLLATE pg\_catalog."default",

karate\_style text COLLATE pg\_catalog."default",

CONSTRAINT club\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.club

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.coach

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

"fullName\_coach" text COLLATE pg\_catalog."default",

salary numeric,

experience integer,

phone\_number text COLLATE pg\_catalog."default",

mastery\_level\_coach text COLLATE pg\_catalog."default",

club\_id\_coach bigint,

CONSTRAINT coach\_pkey PRIMARY KEY (id),

CONSTRAINT club\_fkey\_coach FOREIGN KEY (club\_id\_coach)

REFERENCES public.club (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.coach

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.event

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

" name\_event" text COLLATE pg\_catalog."default",

type\_event text COLLATE pg\_catalog."default",

date\_event date,

level text COLLATE pg\_catalog."default",

sponsor text COLLATE pg\_catalog."default",

club\_id\_event bigint,

evenlocation\_id\_event bigint,

CONSTRAINT event\_pkey PRIMARY KEY (id),

CONSTRAINT club\_fkey\_event FOREIGN KEY (club\_id\_event)

REFERENCES public.club (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT eventlocation\_fkey\_event FOREIGN KEY (evenlocation\_id\_event)

REFERENCES public.event\_location (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.event

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.event\_location

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

price numeric,

address text COLLATE pg\_catalog."default",

capacity integer,

area numeric,

CONSTRAINT event\_location\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.event\_location

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.payment

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

email\_payment text COLLATE pg\_catalog."default",

amount numeric,

type\_payment numeric,

erip\_number text COLLATE pg\_catalog."default",

student\_id\_payment bigint,

CONSTRAINT payment\_pkey PRIMARY KEY (id),

CONSTRAINT student\_fkey\_payment FOREIGN KEY (student\_id\_payment)

REFERENCES public.student (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.payment

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.student

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

"fullName\_stydent" text COLLATE pg\_catalog."default",

age integer,

weight numeric,

mastery\_level\_student text COLLATE pg\_catalog."default",

coach\_id\_student bigint,

CONSTRAINT student\_pkey PRIMARY KEY (id),

CONSTRAINT coah\_fkey FOREIGN KEY (coach\_id\_student)

REFERENCES public.coach (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.student

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.student\_event

(

student\_key bigint NOT NULL,

event\_key bigint NOT NULL,

CONSTRAINT student\_event\_pkey PRIMARY KEY (student\_key, event\_key),

CONSTRAINT "event\_fkey\_tableSE" FOREIGN KEY (event\_key)

REFERENCES public.event (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "student\_fkey\_tableSE" FOREIGN KEY (student\_key)

REFERENCES public.student (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.student\_event

OWNER to postgres;

CREATE TABLE IF NOT EXISTS public.student\_workout

(

student\_key bigint NOT NULL,

workout\_key bigint NOT NULL,

CONSTRAINT student\_workout\_pkey PRIMARY KEY (student\_key, workout\_key),

CONSTRAINT "student\_fkey\_tableSW" FOREIGN KEY (student\_key)

REFERENCES public.student (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "workout\_fkey\_tableSW" FOREIGN KEY (workout\_key)

REFERENCES public.workout (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.student\_workout

OWNER to postgres;

END;

# НАПОЛНЕНИЕ ТАБЛИЦ

Для заполнения таблиц информацией используется директива INSERT. Ниже приведены SQL-запросы и скриншоты заполненных таблиц:

На рисунке 2.1 представлен функционал открытия запросника.

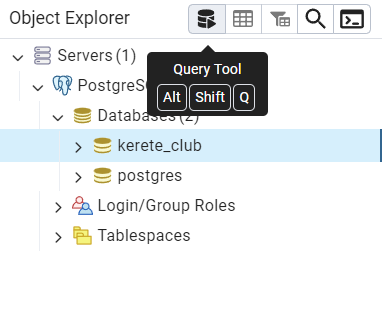


Рисунок 2.2 – Открытие запросника

На рисунке 2.2 представлен функционал выполнения написанного в запроснике SQL-скрипта.

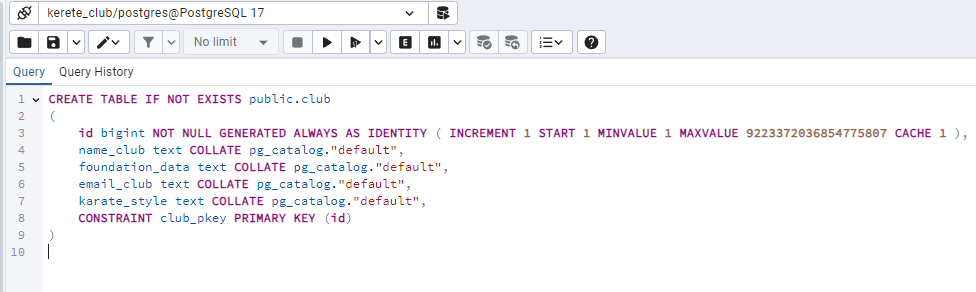


Рисунок 2.2 – Выполнение SQL-скрипта

На рисунке 2.3 представлен SQL-скрипт для заполнения таблицы club.



Рисунок 2.3 – SQL-скрипт для таблицы club

На рисунке 2.4 представлен результат заполнения таблицы club с помощью SQL-скрипта.



Рисунок 2.4 – Результат заполнения таблицы club

На рисунке 2.5 представлен SQL-скрипт для заполнения таблицы coach.



Рисунок 2.5 – SQL-скрипт для таблицы coach

На рисунке 2.6 представлен результат заполнения таблицы coach с помощью SQL-скрипта.



Рисунок 2.6 – Результат заполнения таблицы coach

На рисунке 2.7 представлен SQL-скрипт для заполнения таблицы event\_location.



Рисунок 2.7 – SQL-скрипт для таблицы event\_location

На рисунке 2.8 представлен результат заполнения таблицы event\_location с помощью SQL-скрипта.

Изображение выглядит как текст, меню, снимок экрана, документ

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2.8 – Результат заполнения таблицы event\_location

На рисунке 2.9 представлен SQL-скрипт для заполнения таблицы event.



Рисунок 2.9 – SQL-скрипт для таблицы event

На рисунке 2.10 представлен результат заполнения таблицы event с помощью SQL-скрипта.

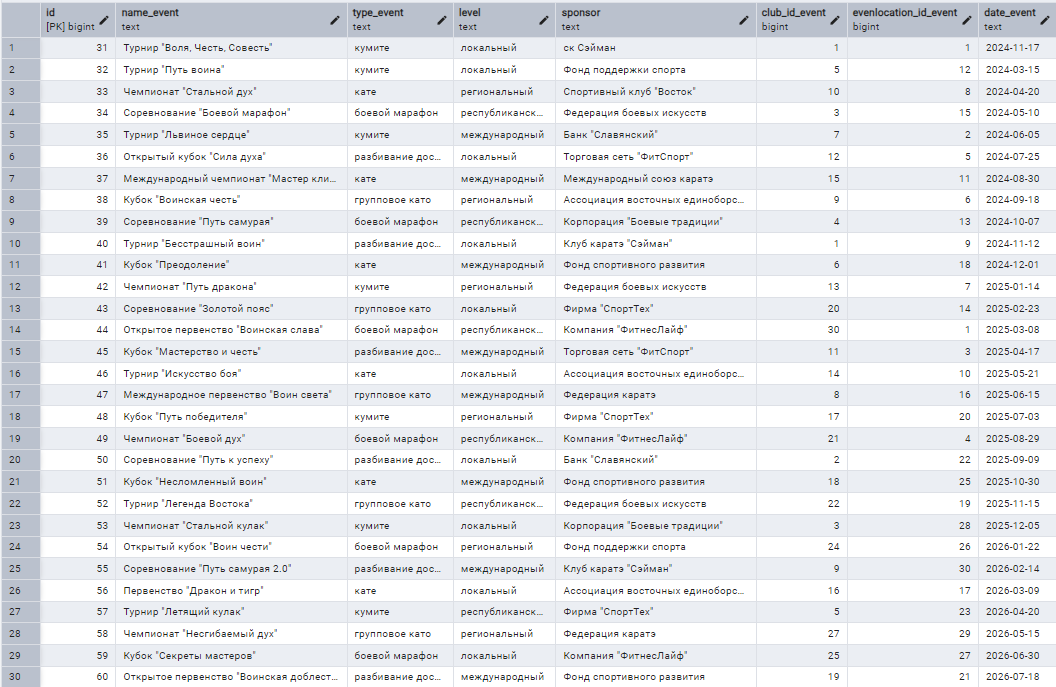


Рисунок 2.10 – Результат заполнения таблицы event

На рисунке 2.11 представлен SQL-скрипт для заполнения таблицы student.

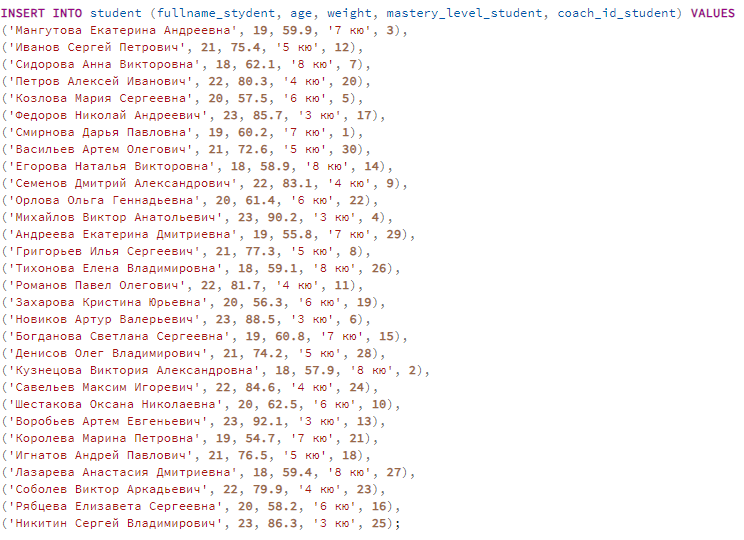


Рисунок 2.11 – SQL-скрипт для таблицы student

На рисунке 2.12 представлен результат заполнения таблицы student с помощью SQL-скрипта.

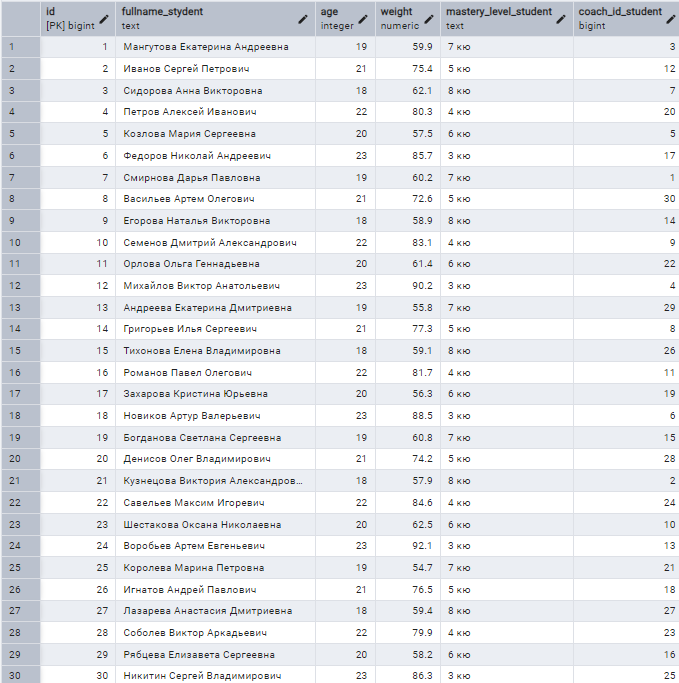


Рисунок 2.12 – Результат заполнения таблицы student

На рисунке 2.13 представлен SQL-скрипт для заполнения таблицы payment.



Рисунок 2.13 – SQL-скрипт для таблицы payment

На рисунке 2.14 представлен результат заполнения таблицы payment с помощью SQL-скрипта.

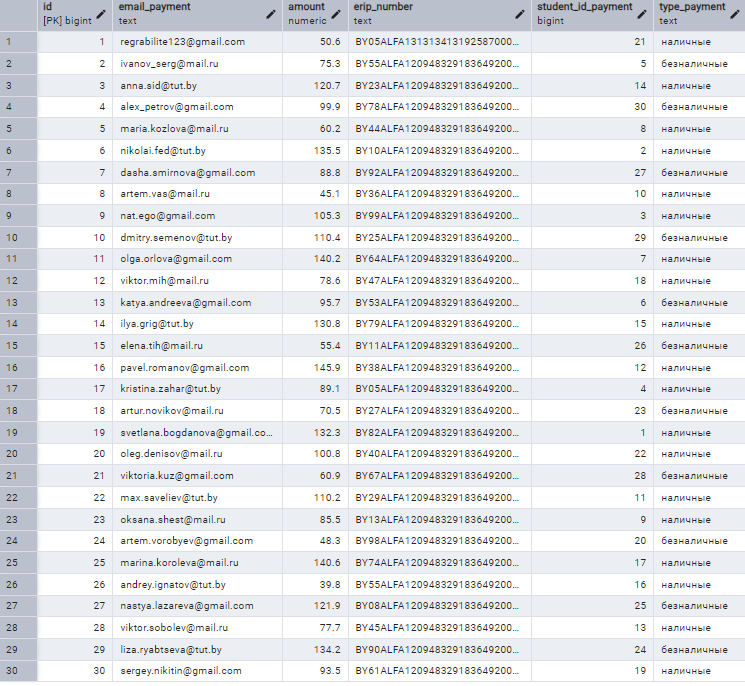


Рисунок 2.14 – Результат заполнения таблицы payment

На рисунке 2.15 представлен SQL-скрипт для заполнения таблицы workout.

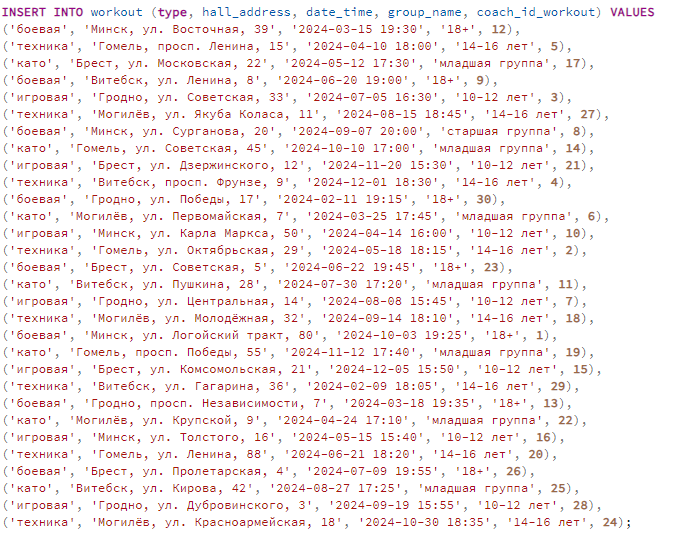


Рисунок 2.15 – SQL-скрипт для таблицы workout

На рисунке 2.16 представлен результат заполнения таблицы workout с помощью SQL-скрипта.



Рисунок 2.16 – Результат заполнения таблицы workout

На рисунке 2.17 представлен результат заполнения таблицы student\_workout с помощью SQL-скрипта.

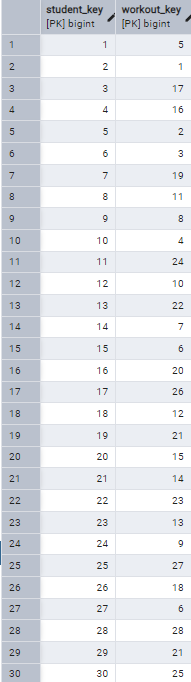


Рисунок 2.17 – SQL-скрипт для таблицы student\_workout

На рисунке 2.18 представлен результат заполнения таблицы student\_event в ручную.

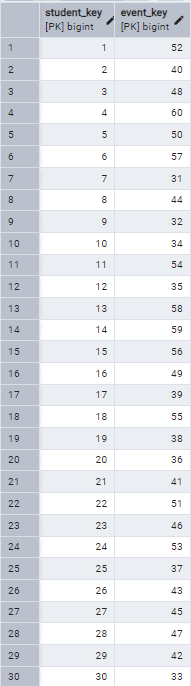


Рисунок 2.18 – Результат заполнения таблицы student\_event

# ДРУГИЕ SQL-КОМАНДЫ

В процессе работы с базами данных часто возникает необходимость изменять структуру таблиц без потери данных. Для этого используется оператор ALTER TABLE, который позволяет вносить изменения в существующую таблицу, такие как переименование столбцов, изменение типов данных, добавление и удаление ограничений. Дальше будут рассмотрены основные операции, выполняемые с помощью ALTER TABLE.

Для переименования столбца в таблице используется следующая команда:

ALTER TABLE coach

RENAME COLUMN fullname\_coach TO name\_coach;

Для изменения типа данных столбца используется следующая команда:

ALTER TABLE coach

ALTER COLUMN mastery\_level\_coach TYPE varchar (10);

Ограничения (CONSTRAINTS) в SQL обеспечивают целостность данных и позволяют задавать дополнительные правила для столбцов. К основным ограничениям относятся PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE, NOT NULL и DEFAULT.

Добавить ограничение можно с помощью следующей команды:

ALTER TABLE coach

ADD CONSTRAINT salary\_check CHECK (salary >= 0);

Удалить ограничение можно с помощью следующей команды:

# ALTER TABLE coach

# DROP CONSTRAINT salary\_check;СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ

Создадим временную таблицу coach\_event которая будет хранить поля coach\_id, event\_id, rank:

CREATE TEMP TABLE coach\_event (coach\_id bigint, event\_id bigint, rank text);

Добавим ограничение PRIMARY\_KEY:

ALTER TABLE coach\_event

ADD CONSTRAINT coach\_event\_pkey PRIMARY KEY (coach\_id, event\_id);

Вставим во временную таблицу тестовые данные:

INSERT INTO coach\_event VALUES

(1, 1, '1'),

(2, 2, '2'),

(3, 3, '3'),

(4, 4, 'Проигрыш'),

(5, 5, '1');

Выведем всю таблицу:

SELECT \* FROM coach\_event;

На рисунке 4.1 представлена временная таблица coach\_event.

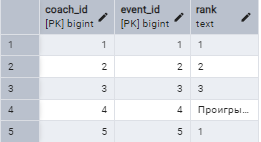


Рисунок 4.1 – Временная таблица coach\_event

Удалим временную таблицу:

DROP TABLE coach\_event;

# ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ

После создания всех таблиц можно экспортировать их структуру в виде UML-диаграммы и сравнить ее с UML-диаграммой, созданной во второй лабораторной работе. Для этого воспользуемся функцией ERD For Database в меню. На рисунке 5.1 представлен функционал генерации диаграммы.

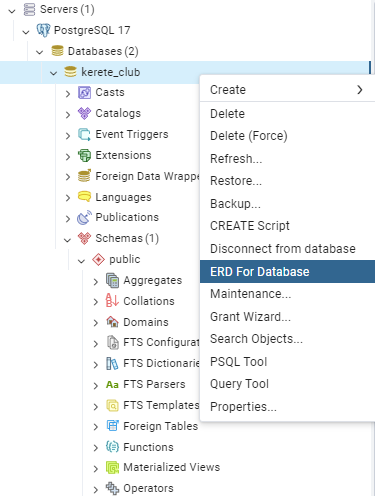


Рисунок 5.1 – Выгрузка диаграммы

На рисунке 5.2 представлен результат генерации диаграммы.

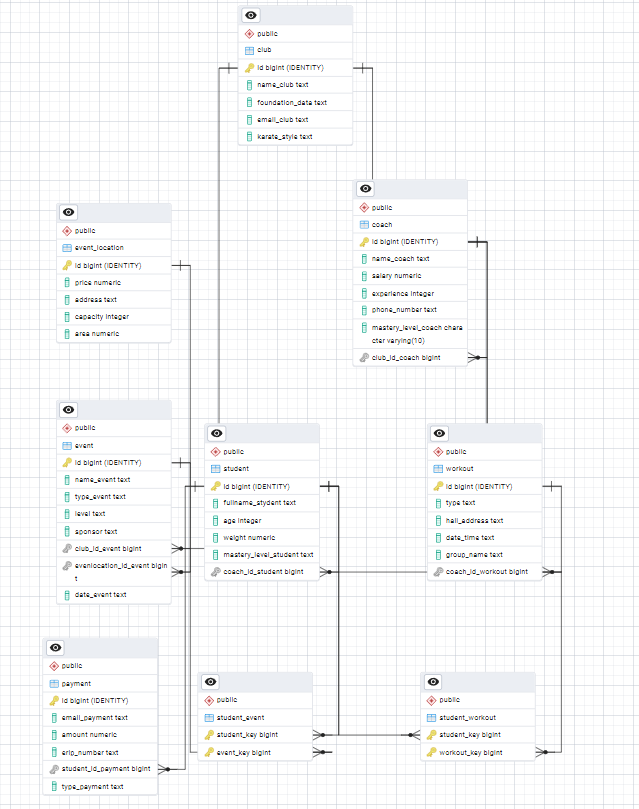


Рисунок 5.2 – Полученная EDR-диаграмма

На рисунке 5.3 полученная UML-диаграмма из предыдущей работы

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5.3 – Полученная UML -диаграмма из предыдущей работы

# ЭКСПОРТ РЕЗУЛЬТАТОВ

Для восстановления базы данных или создания ее копии необходимо сохранить точную структуру таблиц. В pgAdmin 4 для этого предусмотрено специальное меню: База данных (karate\_club) → Backup, где следует указать имя файла и выбрать формат Plain. На рисунке 6.1 представлен функционал для резервного копирования БД karate\_club.

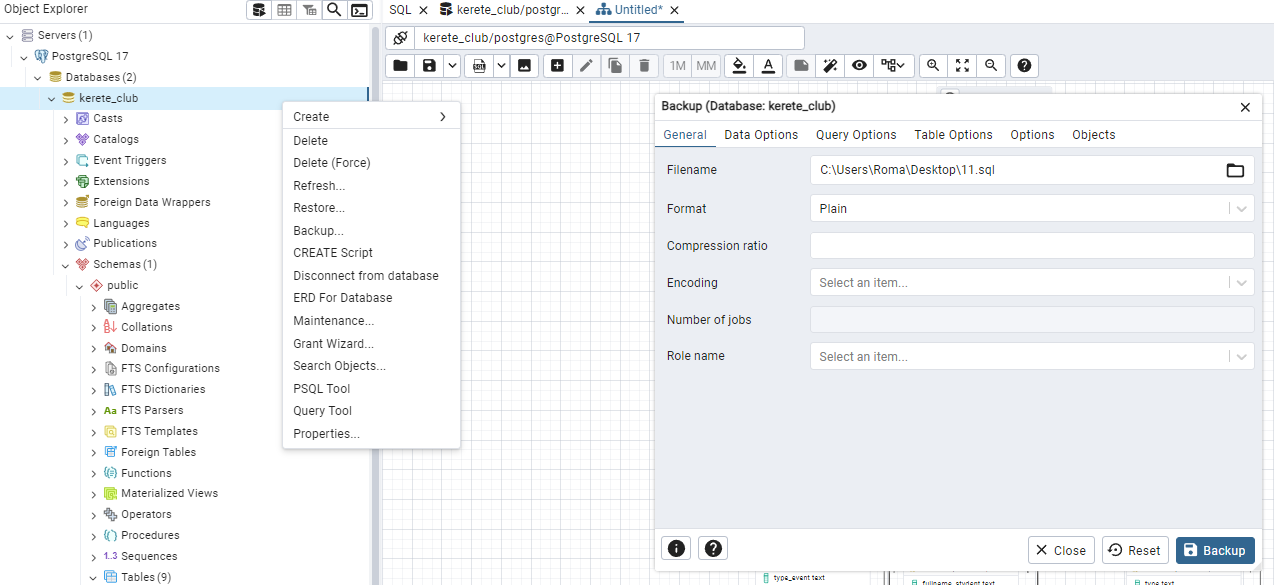


Рисунок 6.1 – Функционал для резервного копирования

базы данных karate\_club

Скрипт для восстановления базы данных airfield (файл karate\_club.sql):

--

-- PostgreSQL database dump

--

-- Dumped from database version 17.2

-- Dumped by pg\_dump version 17.2

-- Started on 2025-02-27 16:45:24

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET transaction\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

-- TOC entry 219 (class 1259 OID 32768)

-- Name: club; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.club (

id bigint NOT NULL,

name\_club text,

foundation\_data text,

email\_club text,

karate\_style text

);

ALTER TABLE public.club OWNER TO postgres;

--

-- TOC entry 4929 (class 0 OID 0)

-- Dependencies: 219

-- Name: TABLE club; Type: COMMENT; Schema: public; Owner: postgres

--

COMMENT ON TABLE public.club IS 'клубы по карате';

--

-- TOC entry 226 (class 1259 OID 32817)

-- Name: club\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.club ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.club\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 220 (class 1259 OID 32771)

-- Name: coach; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.coach (

id bigint NOT NULL,

name\_coach text,

salary numeric,

experience integer,

phone\_number text,

mastery\_level\_coach character varying(10),

club\_id\_coach bigint

);

ALTER TABLE public.coach OWNER TO postgres;

--

-- TOC entry 227 (class 1259 OID 32818)

-- Name: coach\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.coach ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.coach\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 223 (class 1259 OID 32780)

-- Name: event; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.event (

id bigint NOT NULL,

name\_event text,

type\_event text,

level text,

sponsor text,

club\_id\_event bigint,

evenlocation\_id\_event bigint,

date\_event text

);

ALTER TABLE public.event OWNER TO postgres;

--

-- TOC entry 228 (class 1259 OID 32819)

-- Name: event\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.event ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.event\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 224 (class 1259 OID 32783)

-- Name: event\_location; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.event\_location (

id bigint NOT NULL,

price numeric,

address text,

capacity integer,

area numeric

);

ALTER TABLE public.event\_location OWNER TO postgres;

--

-- TOC entry 229 (class 1259 OID 32820)

-- Name: event\_location\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.event\_location ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.event\_location\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 225 (class 1259 OID 32786)

-- Name: payment; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.payment (

id bigint NOT NULL,

email\_payment text,

amount numeric,

erip\_number text,

student\_id\_payment bigint,

type\_payment text

);

ALTER TABLE public.payment OWNER TO postgres;

--

-- TOC entry 230 (class 1259 OID 32821)

-- Name: payment\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.payment ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.payment\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 221 (class 1259 OID 32774)

-- Name: student; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.student (

id bigint NOT NULL,

fullname\_stydent text,

age integer,

weight numeric,

mastery\_level\_student text,

coach\_id\_student bigint

);

ALTER TABLE public.student OWNER TO postgres;

--

-- TOC entry 234 (class 1259 OID 32869)

-- Name: student\_event; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.student\_event (

student\_key bigint NOT NULL,

event\_key bigint NOT NULL

);

ALTER TABLE public.student\_event OWNER TO postgres;

--

-- TOC entry 231 (class 1259 OID 32822)

-- Name: student\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.student ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.student\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 233 (class 1259 OID 32854)

-- Name: student\_workout; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.student\_workout (

student\_key bigint NOT NULL,

workout\_key bigint NOT NULL

);

ALTER TABLE public.student\_workout OWNER TO postgres;

--

-- TOC entry 222 (class 1259 OID 32777)

-- Name: workout; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.workout (

id bigint NOT NULL,

type text,

hall\_address text,

date\_time text,

group\_name text,

coach\_id\_workout bigint

);

ALTER TABLE public.workout OWNER TO postgres;

--

-- TOC entry 232 (class 1259 OID 32823)

-- Name: workout\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.workout ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.workout\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 4908 (class 0 OID 32768)

-- Dependencies: 219

-- Data for Name: club; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.club (id, name\_club, foundation\_data, email\_club, karate\_style) FROM stdin;

1 Сэйман 2005 seiman@gmail.com кёкусинкай

2 Кёкусин Россия 1998 kyokushin.ru@gmail.com кёкусинкай

3 Бусидо 2010 busido@gmail.com шотокан

4 Доджо Ямато 2003 yamato.dojo@gmail.com годзю-рю

5 Тигр 2015 tiger.karate@gmail.com вадорю

6 Каратэ Киокушин 2008 kiokushin.club@gmail.com кёкусинкай

7 Фудошин 2002 fudoshin.karate@gmail.com годзю-рю

8 Шотокан Академия 2011 shotokan.academy@gmail.com шотокан

9 Доджо Сакура 2006 sakura.dojo@gmail.com шотокан

10 Восточный Дракон 2013 dragon.east@gmail.com годзю-рю

11 Самурай Спирит 1999 samurai.spirit@gmail.com шотокан

12 Кёкусин Воины 2007 warriors.kyokushin@gmail.com кёкусинкай

13 Боец 2001 boec.karate@gmail.com шотокан

14 Окинава Карате 2014 okinawa.karate@gmail.com годзю-рю

15 Каратэ-Дзен 2009 karate.zen@gmail.com вадорю

16 Спартанцы 2016 spartans.karate@gmail.com кёкусинкай

17 Доджо Бушидо 2004 bushido.dojo@gmail.com шотокан

18 Путь Воина 2000 way.of.warrior@gmail.com годзю-рю

19 Тенгу 2012 tengu.karate@gmail.com шотокан

20 Сэйкан 1997 seikan.karate@gmail.com кёкусинкай

21 Хонбу Доджо 1995 honbu.dojo@gmail.com годзю-рю

22 Сакура Карате 2017 sakura.karate@gmail.com шотокан

23 Муген 2003 mugen.karate@gmail.com вадорю

24 Ямато Карате 2005 yamato.karate@gmail.com шотокан

25 Восточный Путь 2018 east.way@gmail.com годзю-рю

26 Каратэ-Форс 2010 karate.force@gmail.com кёкусинкай

27 Сёгун 2006 shogun.karate@gmail.com шотокан

28 Львиное Сердце 2019 lion.heart@gmail.com вадорю

29 Тэнсин 2020 tenshin.karate@gmail.com годзю-рю

30 Самурай 2000 samyrai.@gmail.com кёкусинкай

\.

--

-- TOC entry 4909 (class 0 OID 32771)

-- Dependencies: 220

-- Data for Name: coach; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.coach (id, name\_coach, salary, experience, phone\_number, mastery\_level\_coach, club\_id\_coach) FROM stdin;

1 Алейник Иван Владимирович 2450 12 +375291112233 2 дан 5

2 Волков Павел Алексеевич 3100 8 +375292223344 2 дан 17

3 Кочеров Роман Сергеевич 2850 15 +375293334455 4 дан 12

4 Кройч Иван Андрисович 2150 6 +375294445566 4 дан 9

5 Лазовский Илья Александрович 3200 22 +375295556677 5 дан 25

6 Лейбук Елена Владимировна 2600 7 +375296667788 1 дан 3

7 Лукашевич Евгений Андреевич 2750 10 +375297778899 7 дан 14

8 Мысько Никита Андреевич 2900 18 +375298889900 2 дан 21

9 Некраш Максим Александрович 2300 5 +375299990011 3 дан 7

10 Никитенко Роман Александрович 3350 25 +375291122233 4 дан 28

11 Олехнович Никита Витальевич 2200 9 +375292233344 5 дан 2

12 Перещук Владислав Владимирович 2750 14 +375293344455 2 дан 6

13 Пучков Егор Сергеевич 3050 20 +375294455566 3 дан 11

14 Чернявский Владислав Кириллович 2950 17 +375295566677 4 дан 15

15 Чернявский Арсений Андреевич 2150 7 +375296677788 6 дан 22

16 Шустик Алексей Олегович 2800 19 +375297788899 6 дан 4

17 Шутко Александр Павлович 2500 11 +375298899900 1 дан 20

18 Белоусов Олег Викторович 3000 21 +375299911122 2 дан 8

19 Гаврилов Виталий Сергеевич 3100 23 +375291122334 3 дан 16

20 Давыдов Максим Олегович 2250 8 +375292233445 4 дан 29

21 Егоров Николай Анатольевич 2900 15 +375293344556 3 дан 1

22 Жуков Артём Иванович 2350 6 +375294455667 2 дан 18

23 Зайцев Кирилл Владимирович 3200 24 +375295566778 2 дан 13

24 Иванов Сергей Александрович 2750 12 +375296677889 4 дан 26

25 Козлов Виктор Аркадьевич 2500 10 +375297788990 1 дан 30

26 Лебедев Павел Егорович 3350 28 +375298899001 2 дан 19

27 Морозов Андрей Вячеславович 2000 4 +375299900112 1 дан 23

28 Новиков Денис Алексеевич 3100 22 +375291011223 3 дан 10

29 Орлов Тимур Евгеньевич 2850 16 +375292122334 4 дан 27

30 Петров Артём Константинович 2550 9 +375293233445 5 дан 24

\.

--

-- TOC entry 4912 (class 0 OID 32780)

-- Dependencies: 223

-- Data for Name: event; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.event (id, name\_event, type\_event, level, sponsor, club\_id\_event, evenlocation\_id\_event, date\_event) FROM stdin;

31 Турнир "Воля, Честь, Совесть" кумите локальный ск Сэйман 1 1 2024-11-17

32 Турнир "Путь воина" кумите локальный Фонд поддержки спорта 5 12 2024-03-15

33 Чемпионат "Стальной дух" кате региональный Спортивный клуб "Восток" 10 8 2024-04-20

34 Соревнование "Боевой марафон" боевой марафон республиканский Федерация боевых искусств 3 15 2024-05-10

35 Турнир "Львиное сердце" кумите международный Банк "Славянский" 7 2 2024-06-05

36 Открытый кубок "Сила духа" разбивание досок локальный Торговая сеть "ФитСпорт" 12 5 2024-07-25

37 Международный чемпионат "Мастер клинка" кате международный Международный союз каратэ 15 11 2024-08-30

38 Кубок "Воинская честь" групповое като региональный Ассоциация восточных единоборств 9 6 2024-09-18

39 Соревнование "Путь самурая" боевой марафон республиканский Корпорация "Боевые традиции" 4 13 2024-10-07

40 Турнир "Бесстрашный воин" разбивание досок локальный Клуб каратэ "Сэйман" 1 9 2024-11-12

41 Кубок "Преодоление" кате международный Фонд спортивного развития 6 18 2024-12-01

42 Чемпионат "Путь дракона" кумите региональный Федерация боевых искусств 13 7 2025-01-14

43 Соревнование "Золотой пояс" групповое като локальный Фирма "СпортТех" 20 14 2025-02-23

44 Открытое первенство "Воинская слава" боевой марафон республиканский Компания "ФитнесЛайф" 30 1 2025-03-08

45 Кубок "Мастерство и честь" разбивание досок международный Торговая сеть "ФитСпорт" 11 3 2025-04-17

46 Турнир "Искусство боя" кате локальный Ассоциация восточных единоборств 14 10 2025-05-21

47 Международное первенство "Воин света" групповое като международный Федерация каратэ 8 16 2025-06-15

48 Кубок "Путь победителя" кумите региональный Фирма "СпортТех" 17 20 2025-07-03

49 Чемпионат "Боевой дух" боевой марафон республиканский Компания "ФитнесЛайф" 21 4 2025-08-29

50 Соревнование "Путь к успеху" разбивание досок локальный Банк "Славянский" 2 22 2025-09-09

51 Кубок "Несломленный воин" кате международный Фонд спортивного развития 18 25 2025-10-30

52 Турнир "Легенда Востока" групповое като республиканский Федерация боевых искусств 22 19 2025-11-15

53 Чемпионат "Стальной кулак" кумите локальный Корпорация "Боевые традиции" 3 28 2025-12-05

54 Открытый кубок "Воин чести" боевой марафон региональный Фонд поддержки спорта 24 26 2026-01-22

55 Соревнование "Путь самурая 2.0" разбивание досок международный Клуб каратэ "Сэйман" 9 30 2026-02-14

56 Первенство "Дракон и тигр" кате локальный Ассоциация восточных единоборств 16 17 2026-03-09

57 Турнир "Летящий кулак" кумите республиканский Фирма "СпортТех" 5 23 2026-04-20

58 Чемпионат "Несгибаемый дух" групповое като региональный Федерация каратэ 27 29 2026-05-15

59 Кубок "Секреты мастеров" боевой марафон локальный Компания "ФитнесЛайф" 25 27 2026-06-30

60 Открытое первенство "Воинская доблесть" разбивание досок международный Фонд спортивного развития 19 21 2026-07-18

\.

--

-- TOC entry 4913 (class 0 OID 32783)

-- Dependencies: 224

-- Data for Name: event\_location; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.event\_location (id, price, address, capacity, area) FROM stdin;

1 500 Москва, ул. Ленина, 12 150 200.5

2 700 Санкт-Петербург, просп. Мира, 25 300 320.0

3 450 Казань, ул. Кирова, 7 200 250.3

4 600 Екатеринбург, ул. Баумана, 18 350 400.2

5 550 Новосибирск, ул. Советская, 44 250 280.0

6 800 Нижний Новгород, просп. Гагарина, 19 400 500.7

7 750 Челябинск, ул. Победы, 8 600 650.4

8 620 Ростов-на-Дону, ул. Ломоносова, 10 420 470.1

9 680 Уфа, ул. Пушкина, 5 380 430.8

10 570 Воронеж, ул. Дзержинского, 13 270 310.9

11 490 Пермь, просп. Комсомольский, 33 180 220.6

12 530 Красноярск, ул. Гоголя, 22 190 260.3

13 850 Волгоград, ул. Центральная, 2 650 700.1

14 760 Самара, ул. Чехова, 45 500 550.4

15 640 Омск, ул. Карла Маркса, 9 410 460.3

16 480 Тюмень, просп. Ленина, 15 160 210.7

17 510 Иркутск, ул. Сибирская, 21 190 230.5

18 720 Барнаул, ул. Победы, 14 430 480.2

19 700 Хабаровск, ул. Советская, 7 390 440.8

20 590 Ярославль, ул. Октябрьская, 26 280 330.6

21 680 Тула, ул. Суворова, 8 420 470.5

22 620 Калининград, ул. Балтийская, 12 380 430.1

23 770 Саратов, ул. Куйбышева, 6 550 600.3

24 550 Тверь, ул. Московская, 19 250 290.8

25 580 Чебоксары, ул. Калинина, 30 260 310.2

26 690 Белгород, ул. Генерала Жукова, 11 390 440.7

27 530 Рязань, ул. Космонавтов, 16 200 240.6

28 600 Владивосток, просп. Русский, 27 350 400.4

29 820 Ижевск, ул. Гвардейская, 33 620 680.9

30 740 Пенза, ул. Лермонтова, 17 480 520.5

\.

--

-- TOC entry 4914 (class 0 OID 32786)

-- Dependencies: 225

-- Data for Name: payment; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.payment (id, email\_payment, amount, erip\_number, student\_id\_payment, type\_payment) FROM stdin;

1 regrabilite123@gmail.com 50.6 BY05ALFA13131341319258700000 21 наличные

2 ivanov\_serg@mail.ru 75.3 BY55ALFA12094832918364920001 5 безналичные

3 anna.sid@tut.by 120.7 BY23ALFA12094832918364920002 14 наличные

4 alex\_petrov@gmail.com 99.9 BY78ALFA12094832918364920003 30 безналичные

5 maria.kozlova@mail.ru 60.2 BY44ALFA12094832918364920004 8 наличные

6 nikolai.fed@tut.by 135.5 BY10ALFA12094832918364920005 2 наличные

7 dasha.smirnova@gmail.com 88.8 BY92ALFA12094832918364920006 27 безналичные

8 artem.vas@mail.ru 45.1 BY36ALFA12094832918364920007 10 наличные

9 nat.ego@gmail.com 105.3 BY99ALFA12094832918364920008 3 наличные

10 dmitry.semenov@tut.by 110.4 BY25ALFA12094832918364920009 29 безналичные

11 olga.orlova@gmail.com 140.2 BY64ALFA12094832918364920010 7 наличные

12 viktor.mih@mail.ru 78.6 BY47ALFA12094832918364920011 18 наличные

13 katya.andreeva@gmail.com 95.7 BY53ALFA12094832918364920012 6 безналичные

14 ilya.grig@tut.by 130.8 BY79ALFA12094832918364920013 15 наличные

15 elena.tih@mail.ru 55.4 BY11ALFA12094832918364920014 26 безналичные

16 pavel.romanov@gmail.com 145.9 BY38ALFA12094832918364920015 12 наличные

17 kristina.zahar@tut.by 89.1 BY05ALFA12094832918364920016 4 наличные

18 artur.novikov@mail.ru 70.5 BY27ALFA12094832918364920017 23 безналичные

19 svetlana.bogdanova@gmail.com 132.3 BY82ALFA12094832918364920018 1 наличные

20 oleg.denisov@mail.ru 100.8 BY40ALFA12094832918364920019 22 наличные

21 viktoria.kuz@gmail.com 60.9 BY67ALFA12094832918364920020 28 безналичные

22 max.saveliev@tut.by 110.2 BY29ALFA12094832918364920021 11 наличные

23 oksana.shest@mail.ru 85.5 BY13ALFA12094832918364920022 9 наличные

24 artem.vorobyev@gmail.com 48.3 BY98ALFA12094832918364920023 20 безналичные

25 marina.koroleva@mail.ru 140.6 BY74ALFA12094832918364920024 17 наличные

26 andrey.ignatov@tut.by 39.8 BY55ALFA12094832918364920025 16 наличные

27 nastya.lazareva@gmail.com 121.9 BY08ALFA12094832918364920026 25 безналичные

28 viktor.sobolev@mail.ru 77.7 BY45ALFA12094832918364920027 13 наличные

29 liza.ryabtseva@tut.by 134.2 BY90ALFA12094832918364920028 24 безналичные

30 sergey.nikitin@gmail.com 93.5 BY61ALFA12094832918364920029 19 наличные

\.

--

-- TOC entry 4910 (class 0 OID 32774)

-- Dependencies: 221

-- Data for Name: student; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.student (id, fullname\_stydent, age, weight, mastery\_level\_student, coach\_id\_student) FROM stdin;

1 Мангутова Екатерина Андреевна 19 59.9 7 кю 3

2 Иванов Сергей Петрович 21 75.4 5 кю 12

3 Сидорова Анна Викторовна 18 62.1 8 кю 7

4 Петров Алексей Иванович 22 80.3 4 кю 20

5 Козлова Мария Сергеевна 20 57.5 6 кю 5

6 Федоров Николай Андреевич 23 85.7 3 кю 17

7 Смирнова Дарья Павловна 19 60.2 7 кю 1

8 Васильев Артем Олегович 21 72.6 5 кю 30

9 Егорова Наталья Викторовна 18 58.9 8 кю 14

10 Семенов Дмитрий Александрович 22 83.1 4 кю 9

11 Орлова Ольга Геннадьевна 20 61.4 6 кю 22

12 Михайлов Виктор Анатольевич 23 90.2 3 кю 4

13 Андреева Екатерина Дмитриевна 19 55.8 7 кю 29

14 Григорьев Илья Сергеевич 21 77.3 5 кю 8

15 Тихонова Елена Владимировна 18 59.1 8 кю 26

16 Романов Павел Олегович 22 81.7 4 кю 11

17 Захарова Кристина Юрьевна 20 56.3 6 кю 19

18 Новиков Артур Валерьевич 23 88.5 3 кю 6

19 Богданова Светлана Сергеевна 19 60.8 7 кю 15

20 Денисов Олег Владимирович 21 74.2 5 кю 28

21 Кузнецова Виктория Александровна 18 57.9 8 кю 2

22 Савельев Максим Игоревич 22 84.6 4 кю 24

23 Шестакова Оксана Николаевна 20 62.5 6 кю 10

24 Воробьев Артем Евгеньевич 23 92.1 3 кю 13

25 Королева Марина Петровна 19 54.7 7 кю 21

26 Игнатов Андрей Павлович 21 76.5 5 кю 18

27 Лазарева Анастасия Дмитриевна 18 59.4 8 кю 27

28 Соболев Виктор Аркадьевич 22 79.9 4 кю 23

29 Рябцева Елизавета Сергеевна 20 58.2 6 кю 16

30 Никитин Сергей Владимирович 23 86.3 3 кю 25

\.

--

-- TOC entry 4923 (class 0 OID 32869)

-- Dependencies: 234

-- Data for Name: student\_event; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.student\_event (student\_key, event\_key) FROM stdin;

12 35

3 48

25 37

8 44

1 52

17 39

30 33

6 57

21 41

15 56

9 32

28 47

4 60

19 38

23 46

11 54

7 31

26 43

14 59

5 50

2 40

18 55

20 36

29 42

13 58

10 34

24 53

16 49

27 45

22 51

\.

--

-- TOC entry 4922 (class 0 OID 32854)

-- Dependencies: 233

-- Data for Name: student\_workout; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.student\_workout (student\_key, workout\_key) FROM stdin;

1 5

2 1

3 17

4 16

5 2

6 3

7 19

8 11

9 8

10 4

11 24

12 10

13 22

14 7

15 6

16 20

17 26

18 12

19 21

20 15

21 14

22 23

23 13

24 9

25 27

26 18

27 6

28 28

29 21

30 25

\.

--

-- TOC entry 4911 (class 0 OID 32777)

-- Dependencies: 222

-- Data for Name: workout; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.workout (id, type, hall\_address, date\_time, group\_name, coach\_id\_workout) FROM stdin;

1 боевая Минск, ул. Восточная, 39 2024-03-15 19:30 18+ 12

2 техника Гомель, просп. Ленина, 15 2024-04-10 18:00 14-16 лет 5

3 като Брест, ул. Московская, 22 2024-05-12 17:30 младшая группа 17

4 боевая Витебск, ул. Ленина, 8 2024-06-20 19:00 18+ 9

5 игровая Гродно, ул. Советская, 33 2024-07-05 16:30 10-12 лет 3

6 техника Могилёв, ул. Якуба Коласа, 11 2024-08-15 18:45 14-16 лет 27

7 боевая Минск, ул. Сурганова, 20 2024-09-07 20:00 старшая группа 8

8 като Гомель, ул. Советская, 45 2024-10-10 17:00 младшая группа 14

9 игровая Брест, ул. Дзержинского, 12 2024-11-20 15:30 10-12 лет 21

10 техника Витебск, просп. Фрунзе, 9 2024-12-01 18:30 14-16 лет 4

11 боевая Гродно, ул. Победы, 17 2024-02-11 19:15 18+ 30

12 като Могилёв, ул. Первомайская, 7 2024-03-25 17:45 младшая группа 6

13 игровая Минск, ул. Карла Маркса, 50 2024-04-14 16:00 10-12 лет 10

14 техника Гомель, ул. Октябрьская, 29 2024-05-18 18:15 14-16 лет 2

15 боевая Брест, ул. Советская, 5 2024-06-22 19:45 18+ 23

16 като Витебск, ул. Пушкина, 28 2024-07-30 17:20 младшая группа 11

17 игровая Гродно, ул. Центральная, 14 2024-08-08 15:45 10-12 лет 7

18 техника Могилёв, ул. Молодёжная, 32 2024-09-14 18:10 14-16 лет 18

19 боевая Минск, ул. Логойский тракт, 80 2024-10-03 19:25 18+ 1

20 като Гомель, просп. Победы, 55 2024-11-12 17:40 младшая группа 19

21 игровая Брест, ул. Комсомольская, 21 2024-12-05 15:50 10-12 лет 15

22 техника Витебск, ул. Гагарина, 36 2024-02-09 18:05 14-16 лет 29

23 боевая Гродно, просп. Независимости, 7 2024-03-18 19:35 18+ 13

24 като Могилёв, ул. Крупской, 9 2024-04-24 17:10 младшая группа 22

25 игровая Минск, ул. Толстого, 16 2024-05-15 15:40 10-12 лет 16

26 техника Гомель, ул. Ленина, 88 2024-06-21 18:20 14-16 лет 20

27 боевая Брест, ул. Пролетарская, 4 2024-07-09 19:55 18+ 26

28 като Витебск, ул. Кирова, 42 2024-08-27 17:25 младшая группа 25

29 игровая Гродно, ул. Дубровинского, 3 2024-09-19 15:55 10-12 лет 28

30 техника Могилёв, ул. Красноармейская, 18 2024-10-30 18:35 14-16 лет 24

\.

--

-- TOC entry 4930 (class 0 OID 0)

-- Dependencies: 226

-- Name: club\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.club\_id\_seq', 30, true);

--

-- TOC entry 4931 (class 0 OID 0)

-- Dependencies: 227

-- Name: coach\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.coach\_id\_seq', 30, true);

--

-- TOC entry 4932 (class 0 OID 0)

-- Dependencies: 228

-- Name: event\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.event\_id\_seq', 60, true);

--

-- TOC entry 4933 (class 0 OID 0)

-- Dependencies: 229

-- Name: event\_location\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.event\_location\_id\_seq', 30, true);

--

-- TOC entry 4934 (class 0 OID 0)

-- Dependencies: 230

-- Name: payment\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.payment\_id\_seq', 30, true);

--

-- TOC entry 4935 (class 0 OID 0)

-- Dependencies: 231

-- Name: student\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.student\_id\_seq', 30, true);

--

-- TOC entry 4936 (class 0 OID 0)

-- Dependencies: 232

-- Name: workout\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.workout\_id\_seq', 30, true);

--

-- TOC entry 4736 (class 2606 OID 32790)

-- Name: club club\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.club

ADD CONSTRAINT club\_pkey PRIMARY KEY (id);

--

-- TOC entry 4738 (class 2606 OID 32796)

-- Name: coach coach\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.coach

ADD CONSTRAINT coach\_pkey PRIMARY KEY (id);

--

-- TOC entry 4746 (class 2606 OID 32804)

-- Name: event\_location event\_location\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.event\_location

ADD CONSTRAINT event\_location\_pkey PRIMARY KEY (id);

--

-- TOC entry 4744 (class 2606 OID 32800)

-- Name: event event\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.event

ADD CONSTRAINT event\_pkey PRIMARY KEY (id);

--

-- TOC entry 4748 (class 2606 OID 32808)

-- Name: payment payment\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.payment

ADD CONSTRAINT payment\_pkey PRIMARY KEY (id);

--

-- TOC entry 4752 (class 2606 OID 32873)

-- Name: student\_event student\_event\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student\_event

ADD CONSTRAINT student\_event\_pkey PRIMARY KEY (student\_key, event\_key);

--

-- TOC entry 4740 (class 2606 OID 32812)

-- Name: student student\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student

ADD CONSTRAINT student\_pkey PRIMARY KEY (id);

--

-- TOC entry 4750 (class 2606 OID 32858)

-- Name: student\_workout student\_workout\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student\_workout

ADD CONSTRAINT student\_workout\_pkey PRIMARY KEY (student\_key, workout\_key);

--

-- TOC entry 4742 (class 2606 OID 32816)

-- Name: workout workout\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.workout

ADD CONSTRAINT workout\_pkey PRIMARY KEY (id);

--

-- TOC entry 4753 (class 2606 OID 32824)

-- Name: coach club\_fkey\_coach; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.coach

ADD CONSTRAINT club\_fkey\_coach FOREIGN KEY (club\_id\_coach) REFERENCES public.club(id) NOT VALID;

--

-- TOC entry 4756 (class 2606 OID 32829)

-- Name: event club\_fkey\_event; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.event

ADD CONSTRAINT club\_fkey\_event FOREIGN KEY (club\_id\_event) REFERENCES public.club(id) NOT VALID;

--

-- TOC entry 4755 (class 2606 OID 32839)

-- Name: workout coach\_fkey\_workout; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.workout

ADD CONSTRAINT coach\_fkey\_workout FOREIGN KEY (coach\_id\_workout) REFERENCES public.coach(id) NOT VALID;

--

-- TOC entry 4754 (class 2606 OID 32834)

-- Name: student coah\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student

ADD CONSTRAINT coah\_fkey FOREIGN KEY (coach\_id\_student) REFERENCES public.coach(id) NOT VALID;

--

-- TOC entry 4761 (class 2606 OID 32879)

-- Name: student\_event event\_fkey\_tableSE; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student\_event

ADD CONSTRAINT "event\_fkey\_tableSE" FOREIGN KEY (event\_key) REFERENCES public.event(id) NOT VALID;

--

-- TOC entry 4757 (class 2606 OID 32849)

-- Name: event eventlocation\_fkey\_event; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.event

ADD CONSTRAINT eventlocation\_fkey\_event FOREIGN KEY (evenlocation\_id\_event) REFERENCES public.event\_location(id) NOT VALID;

--

-- TOC entry 4758 (class 2606 OID 32844)

-- Name: payment student\_fkey\_payment; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.payment

ADD CONSTRAINT student\_fkey\_payment FOREIGN KEY (student\_id\_payment) REFERENCES public.student(id) NOT VALID;

--

-- TOC entry 4762 (class 2606 OID 32874)

-- Name: student\_event student\_fkey\_tableSE; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student\_event

ADD CONSTRAINT "student\_fkey\_tableSE" FOREIGN KEY (student\_key) REFERENCES public.student(id) NOT VALID;

--

-- TOC entry 4759 (class 2606 OID 32859)

-- Name: student\_workout student\_fkey\_tableSW; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student\_workout

ADD CONSTRAINT "student\_fkey\_tableSW" FOREIGN KEY (student\_key) REFERENCES public.student(id) NOT VALID;

--

-- TOC entry 4760 (class 2606 OID 32864)

-- Name: student\_workout workout\_fkey\_tableSW; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.student\_workout

ADD CONSTRAINT "workout\_fkey\_tableSW" FOREIGN KEY (workout\_key) REFERENCES public.workout(id) NOT VALID;

-- Completed on 2025-02-27 16:45:24

--

-- PostgreSQL database dump complete

--

# ЗАКЛЮЧЕНИЕ

В процессе выполнения лабораторной работы были успешно решены все поставленные задачи, которые включали создание новой схемы данных, проектирование и заполнение таблиц базы данных, а также проверку связей между ними. В результате были приобретены навыки работы с SQL-операторами, такими как CREATE TABLE, INSERT и ALTER TABLE, что позволило адаптировать структуру данных в соответствии с требованиями работы.

Экспорт результатов в SQL-скрипт предоставил возможность сравнить полученные данные с эталонными скриптами, что подтвердило правильность выполнения всех операций. Завершенная работа способствовала углублению понимания принципов проектирования и управления реляционными базами данных, а также развитию практических навыков написания SQL-запросов для создания, модификации и заполнения таблиц.