Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

Тема «Репетиционная база»

Лабораторная работа №3

Реализация SQL-запросов для создания базы данных

Студент: А.С. Бригадир

Преподаватель: Д.В. Куприянова

МИНСК 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc191342528)

[1 СОЗДАНИЕ ТАБЛИЦ 4](#_Toc191342529)

[2 НАПОЛНЕНИЕ ТАБЛИЦ 7](#_Toc191342530)

[3 ДРУГИЕ SQL-КОМАНДЫ 13](#_Toc191342531)

[4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ 14](#_Toc191342532)

[5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ 15](#_Toc191342533)

[6 ЭКСПОРТ РЕЗУЛЬТАТОВ 16](#_Toc191342534)

[ЗАКЛЮЧЕНИЕ 29](#_Toc191342535)

# ВВЕДЕНИЕ

Данная лабораторная работа направлена на детальное изучение процесса проектирования, создания и управления реляционными базами данных с использованием языка SQL. В рамках выполнения заданий предполагается освоить основы работы с архитектурой баз данных, включая разработку схем данных, создание таблиц и определение их структуры. Особый акцент будет сделан на установлении связей между таблицами с использованием первичных и внешних ключей, а также на реализации бизнес-правил, обеспечивающих целостность и корректность данных.

Работа предполагает выполнение практических заданий по внесению данных в таблицы как с помощью SQL-запросов, так и через встроенные инструменты пользовательского интерфейса системы управления базами данных (СУБД). Также будут изучены операции с командой ALTER TABLE для изменения структуры таблиц, создание и удаление временных таблиц, а также экспорт данных в виде SQL-скриптов для их последующего анализа и проверки. Выполнение всех этапов лабораторной работы позволит не только закрепить теоретические знания, но и развить практические умения, необходимые для эффективной работы с реляционными базами данных в профессиональной среде.

# 1 СОЗДАНИЕ ТАБЛИЦ

Ниже приведены SQL-запросы для создания всех таблиц.

CREATE TABLE IF NOT EXISTS main.rehearsal\_points (

id SERIAL PRIMARY KEY,

rating REAL,

contact\_number VARCHAR(15) NOT NULL,

schedule JSON,

name TEXT NOT NULL,

address TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS main.rooms (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

air\_conditioner BOOLEAN NOT NULL DEFAULT FALSE,

price INT NOT NULL,

recording\_support BOOLEAN NOT NULL DEFAULT FALSE,

area INT NOT NULL,

id\_rehearsal\_point INT

);

CREATE TABLE IF NOT EXISTS main.service (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

price INT NOT NULL,

type TEXT NOT NULL,

requirements TEXT,

id\_rehearsal\_point INT

);

CREATE TABLE IF NOT EXISTS main.equipment (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

type TEXT NOT NULL,

brand TEXT NOT NULL,

model TEXT NOT NULL,

condition TEXT NOT NULL,

id\_rehearsal\_point INT

);

CREATE TABLE IF NOT EXISTS main.staff (

id SERIAL PRIMARY KEY,

full\_name TEXT NOT NULL,

address TEXT,

experience INT,

phone VARCHAR(15) NOT NULL,

age INT NOT NULL,

id\_rehearsal\_point INT

);

CREATE TABLE IF NOT EXISTS main.users (

id SERIAL PRIMARY KEY,

full\_name TEXT NOT NULL,

phone VARCHAR(15) NOT NULL,

email TEXT NOT NULL,

registration\_date TIMESTAMP NOT NULL

);

CREATE TABLE IF NOT EXISTS main.booking (

id SERIAL PRIMARY KEY,

time TIMESTAMP NOT NULL,

duration INT,

cost INT NOT NULL,

creation\_date TIMESTAMP NOT NULL,

status VARCHAR(50) NOT NULL,

number\_of\_people INT NOT NULL,

id\_room INT,

id\_user INT

);

CREATE TABLE IF NOT EXISTS main.service\_booking (

id\_service INT NOT NULL,

id\_booking INT NOT NULL,

PRIMARY KEY (id\_service, id\_booking)

);

CREATE TABLE IF NOT EXISTS main.equipment\_booking (

id\_equipment INT NOT NULL,

id\_booking INT NOT NULL,

PRIMARY KEY (id\_equipment, id\_booking)

);

ALTER TABLE IF EXISTS main.rooms

ADD CONSTRAINT fk\_rooms\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point)

REFERENCES main.rehearsal\_points(id) ON DELETE CASCADE;

ALTER TABLE IF EXISTS main.service

ADD CONSTRAINT fk\_service\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point)

REFERENCES main.rehearsal\_points(id) ON DELETE CASCADE;

ALTER TABLE IF EXISTS main.equipment

ADD CONSTRAINT fk\_equipment\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point)

REFERENCES main.rehearsal\_points(id) ON DELETE CASCADE;

ALTER TABLE IF EXISTS main.staff

ADD CONSTRAINT fk\_staff\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point)

REFERENCES main.rehearsal\_points(id) ON DELETE SET NULL;

ALTER TABLE IF EXISTS main.booking

ADD CONSTRAINT fk\_booking\_room FOREIGN KEY (id\_room)

REFERENCES main.rooms(id) ON DELETE SET NULL;

ALTER TABLE IF EXISTS main.booking

ADD CONSTRAINT fk\_booking\_user FOREIGN KEY (id\_user)

REFERENCES main.users(id) ON DELETE SET NULL;

ALTER TABLE IF EXISTS main.service\_booking

ADD CONSTRAINT fk\_service\_booking\_service FOREIGN KEY (id\_service)

REFERENCES main.service(id) ON DELETE SET NULL;

ALTER TABLE IF EXISTS main.service\_booking

ADD CONSTRAINT fk\_service\_booking\_booking FOREIGN KEY (id\_booking)

REFERENCES main.booking(id) ON DELETE SET NULL;

ALTER TABLE IF EXISTS main.equipment\_booking

ADD CONSTRAINT fk\_equipment\_booking\_equipment FOREIGN KEY (id\_equipment)

REFERENCES main.equipment(id) ON DELETE SET NULL;

ALTER TABLE IF EXISTS main.equipment\_booking

ADD CONSTRAINT fk\_equipment\_booking\_booking FOREIGN KEY (id\_booking)

REFERENCES main.booking(id) ON DELETE SET NULL;

# 2 НАПОЛНЕНИЕ ТАБЛИЦ

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

Таблица rehearsal\_points: заполнение вручную.

На рисунке 2.1 представлен функционал добавления строки данных в таблицу, на рисунке 2.2 функционал сохранения изменений в базе данных.

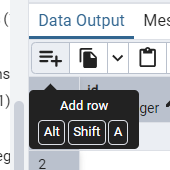


Рисунок 2.1 – Добавление данных

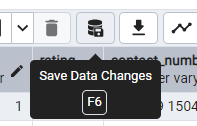


Рисунок 2.2 – Сохранение в базу данных

На рисунке 2.3 представлено содержание таблицы rehearsal\_points.

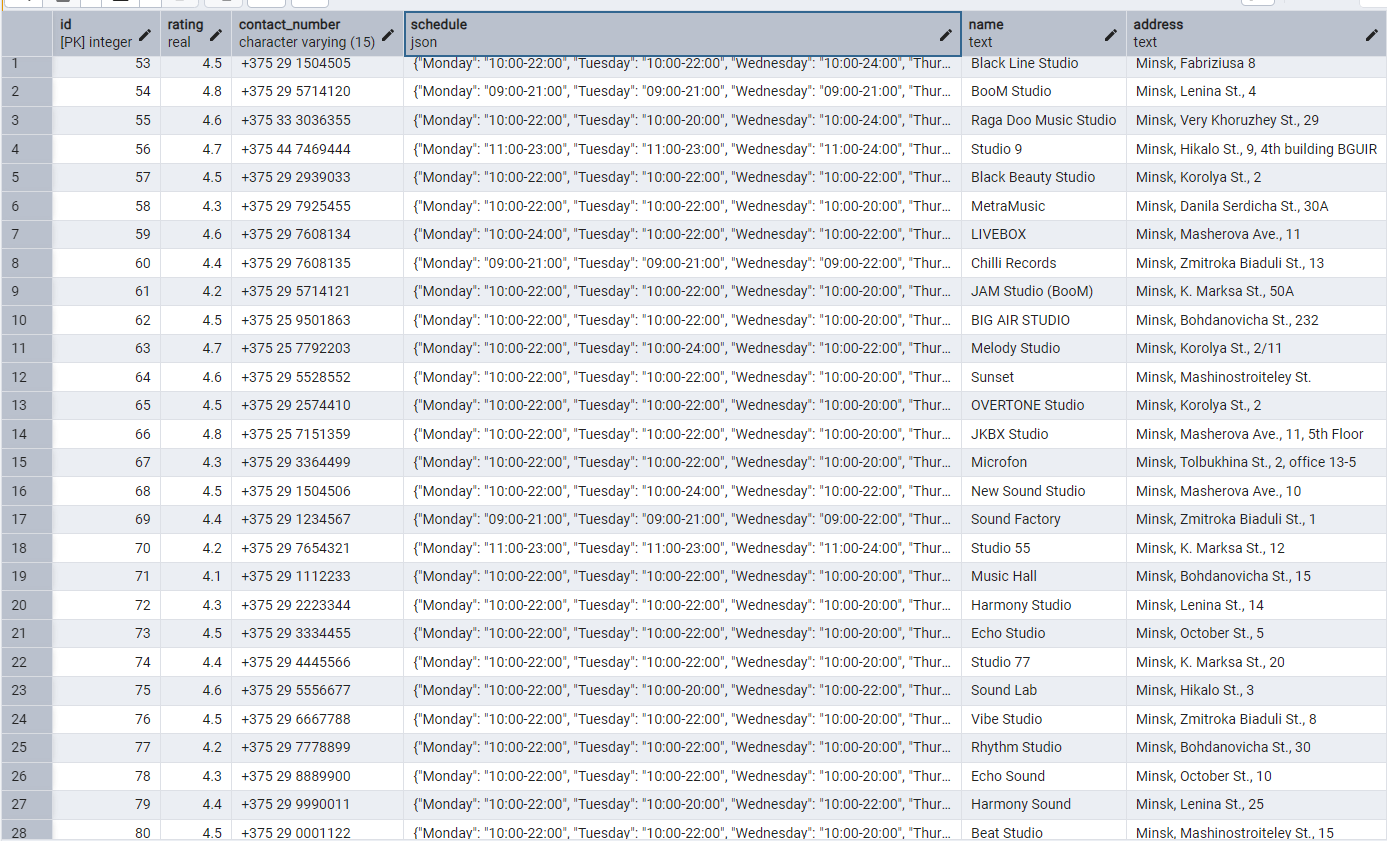


Рисунок 2.3 – Содержание таблицы «rehearsal\_points»

Таблица rooms: заполнение вручную.

Заполнение таблицы происходит способом, аналогичным с таблицей rehearsal\_points, отображенным на рисунках 2.1 и 2.2.

Содержание таблицы rooms отображено на рисунке 2.4.

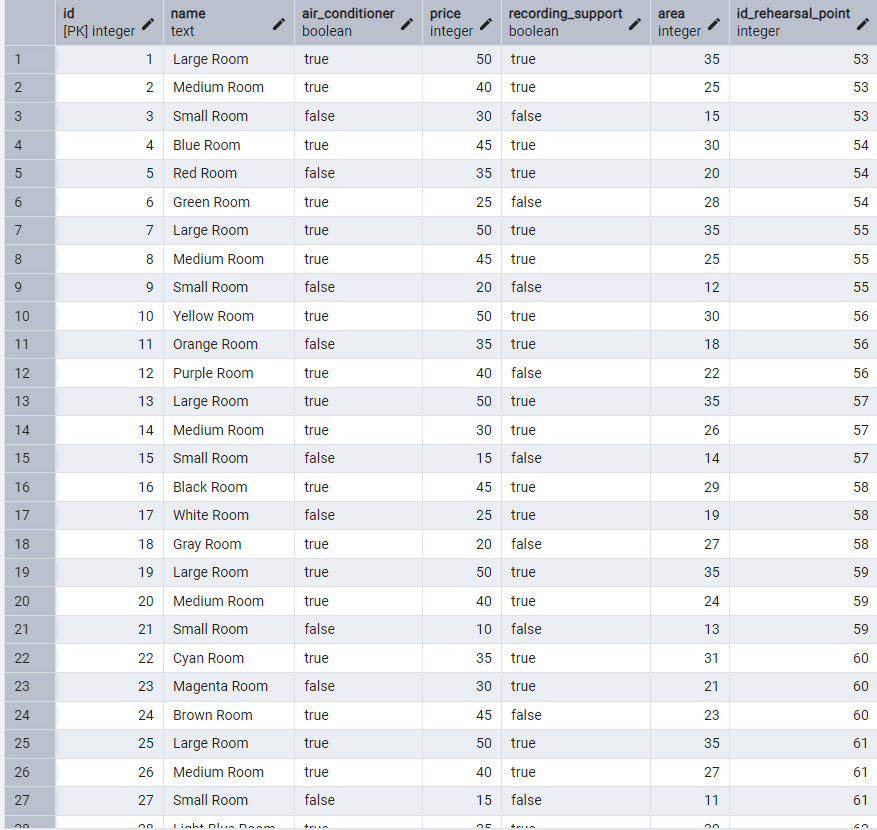


Рисунок 2.4 – Содержание таблицы «rooms»

Таблица service: заполнение вручную.

Заполнение таблицы происходит способом, аналогичным с таблицей rehearsal\_points, отображенным на рисунках 2.1 и 2.2.

Содержание таблицы service отображено на рисунке 2.5.

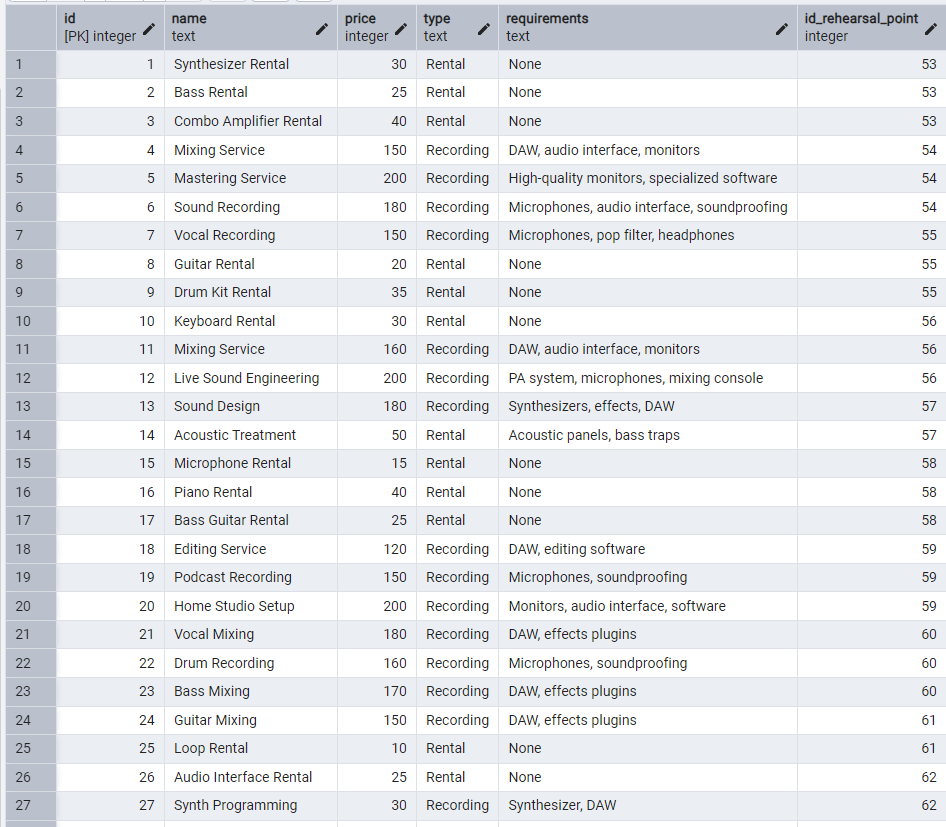


Рисунок 2.5 – Содержание таблицы «service»

Таблица equipment: заполнение вручную.

Заполнение таблицы происходит способом, аналогичным с таблицей rehearsal\_points, отображенным на рисунках 2.1 и 2.2.

Содержание таблицы equipment отображено на рисунке 2.6.



Рисунок 2.6 – Содержание таблицы «equipment»

Таблица staff: заполнение вручную.

Заполнение таблицы происходит способом, аналогичным с таблицей rehearsal\_points, отображенным на рисунках 2.1 и 2.2.

Содержание таблицы staff отображено на рисунке 2.7.

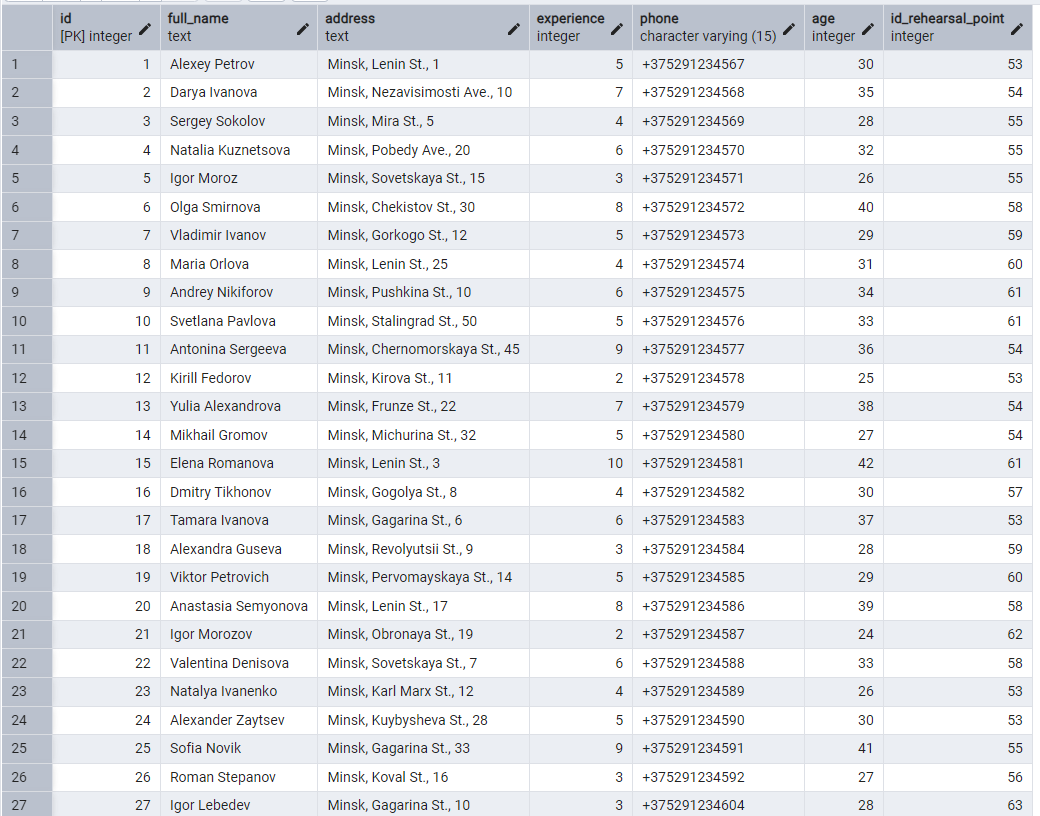


Рисунок 2.7 – Содержание таблицы «staff»

Таблица users: заполнение вручную.

Заполнение таблицы происходит способом, аналогичным с таблицей rehearsal\_points, отображенным на рисунках 2.1 и 2.2.

Содержание таблицы users отображено на рисунке 2.8.

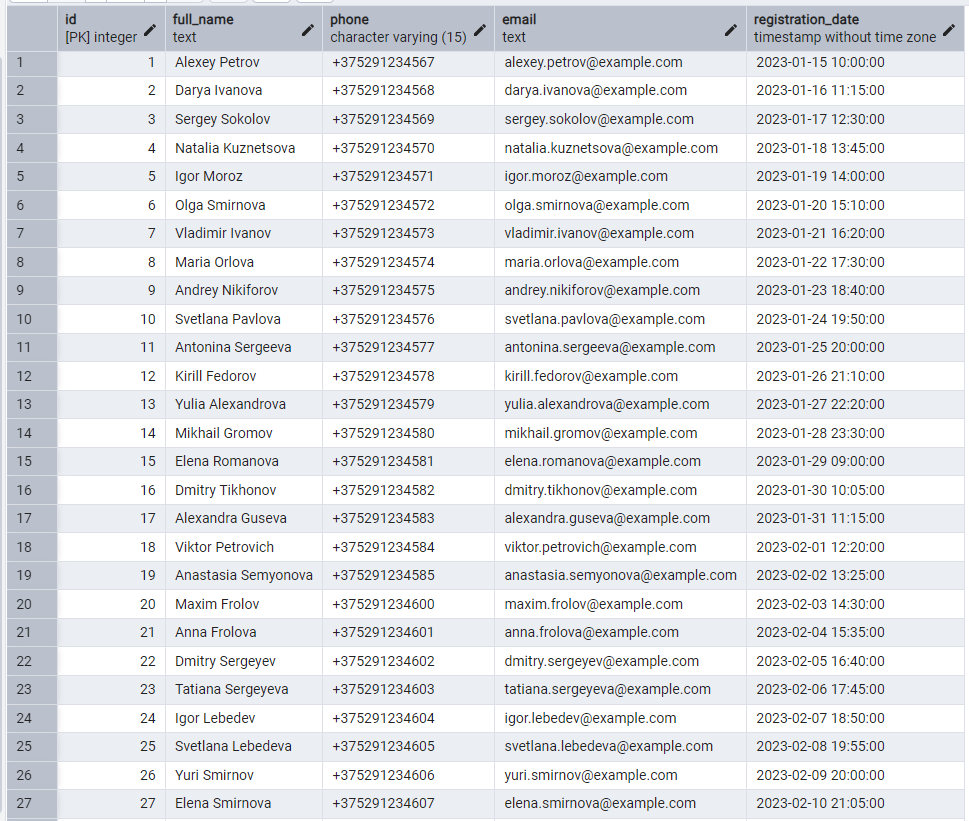


Рисунок 2.8 – Содержание таблицы «users»

Таблица booking: заполнение вручную.

Заполнение таблицы происходит способом, аналогичным с таблицей rehearsal\_points, отображенным на рисунках 2.1 и 2.2.

Содержание таблицы booking отображено на рисунке 2.9.

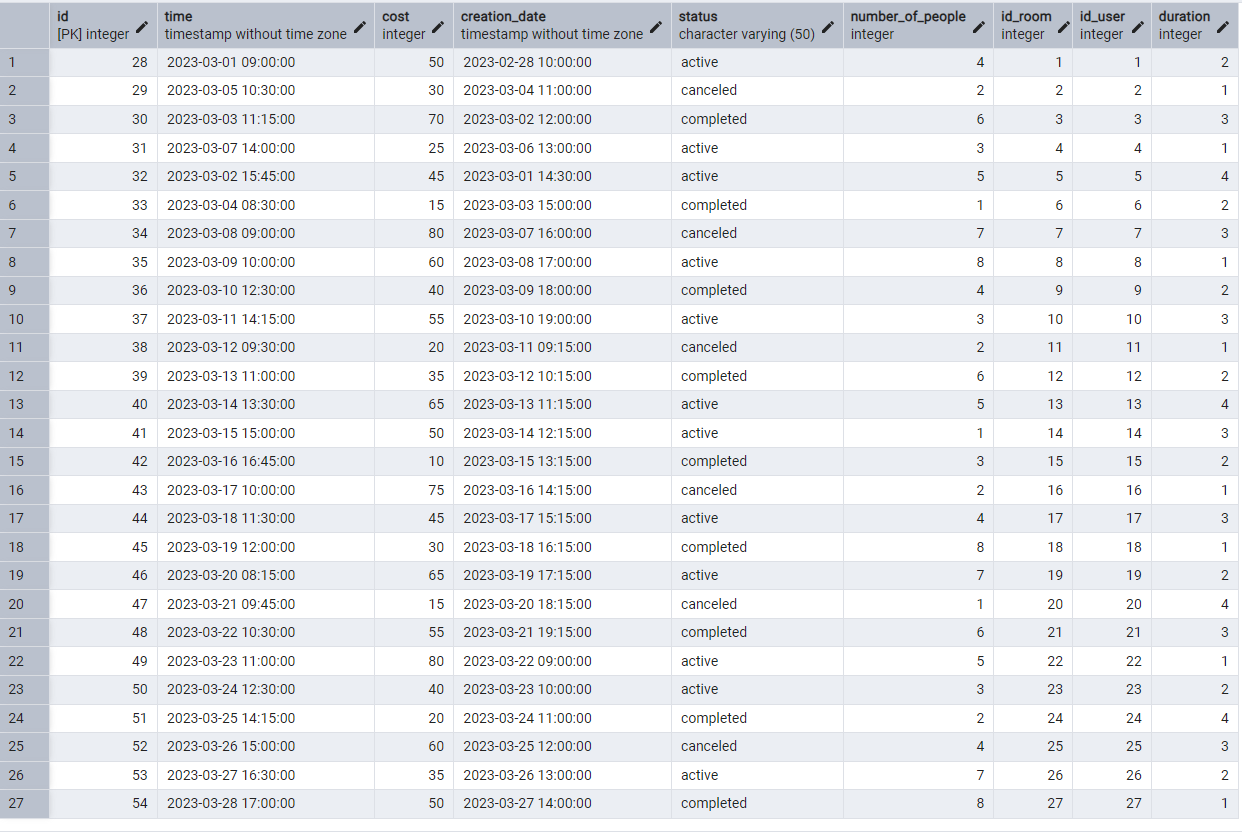


Рисунок 2.9 – Содержание таблицы «booking»

Таблица service\_booking:

INSERT INTO main.service\_booking (id\_service, id\_booking) VALUES (1, 28), (2, 29), (3, 30), (4, 31), (5, 32), (6, 33), (7, 34), (8, 35), (9, 36), (10, 37), (11, 38), (12, 39), (13, 40), (14, 41), (15, 42), (16, 43), (17, 44), (18, 45), (19, 46), (20, 47), (21, 48), (22, 49), (23, 50);

Содержание таблицы service\_booking отображено на рисунке 2.10.

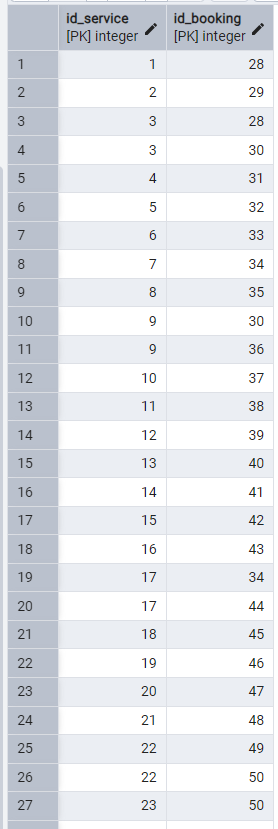


Рисунок 2.10 – Содержание таблицы «service\_booking»

Таблица equipment\_booking:

INSERT INTO main.equipment\_booking (id\_equipment, id\_booking) VALUES (1, 28), (2, 29), (3, 30), (4, 31), (5, 32), (6, 33), (7, 34), (8, 35), (9, 36), (10, 37), (11, 38), (12, 39), (13, 40), (14, 41), (15, 42), (16, 43), (17, 44), (18, 45), (19, 46), (20, 47), (21, 48), (22, 49), (23, 50);

Содержание таблицы equipment\_booking отображено на рисунке 2.11.

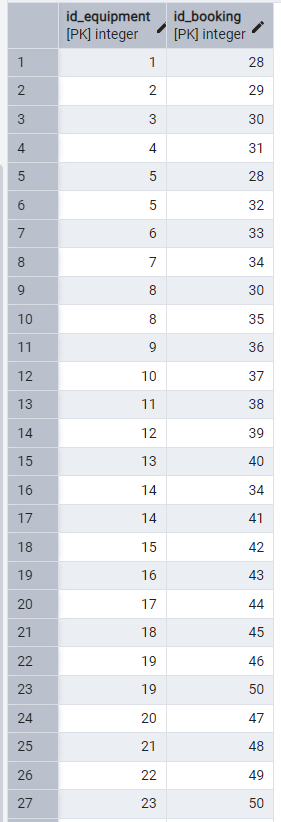
****

Рисунок 2.11 – Содержание таблицы «equipment\_booking»

# 3 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. Далее будут рассмотрены основные операции, выполняемые с помощью ALTER TABLE.

Для добавления столбца в таблицу используется следующая команда:

ALTER TABLE main.booking ADD COLUMN duration INT;

Для переименования столбца в таблице используется следующая команда:

ALTER TABLE main.booking RENAME COLUMN time TO booking\_time;

Для изменения типа данных столбца используется следующая команда:

ALTER TABLE main.booking ALTER COLUMN cost TYPE DECIMAL(10, 2);

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

ALTER TABLE IF EXISTS main.booking

ADD CONSTRAINT fk\_booking\_room FOREIGN KEY (id\_room)

REFERENCES main.rooms(id) ON DELETE SET NULL;

Удалить ограничение можно с помощью следующей команды:

ALTER TABLE main.booking DROP CONSTRAINT fk\_booking\_room;

Для удаления столбца используется следующая команда:

ALTER TABLE main.booking DROP COLUMN duration;

# 4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ

Создадим временную таблицу music\_events которая будет хранить поля name, event\_date:  CREATE TEMP TABLE music\_events ( name TEXT NOT NULL, event\_date DATE NOT NULL );

Вставим во временную таблицу тестовые данные:

INSERT INTO music\_events (name, event\_date) VALUES

('Concert Linkin Park', '2025-03-15'),

('Concert Papin Olimpos', '2025-04-20'),

('Concert Megadeth', '2025-05-10');

Сделаем выборку: SELECT \* FROM music\_events;

Результат представлен на рисунке 4.1.

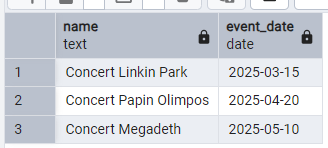


Рисунок 4.1 – Выборка данных из временной таблицы

Удалим временную таблицу: DROP TABLE music\_events;

# 5 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 Schema в меню. На рисунке 5.1 представлен функционал генерации диаграммы. На рисунке 5.2 представлен результат генерации диаграммы.

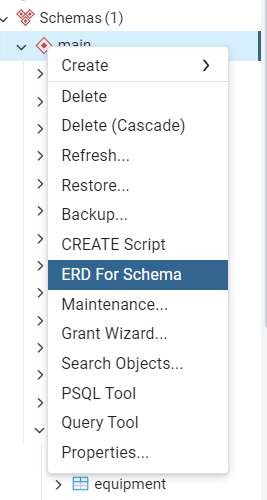


Рисунок 5.1 – Выгрузка диаграммы

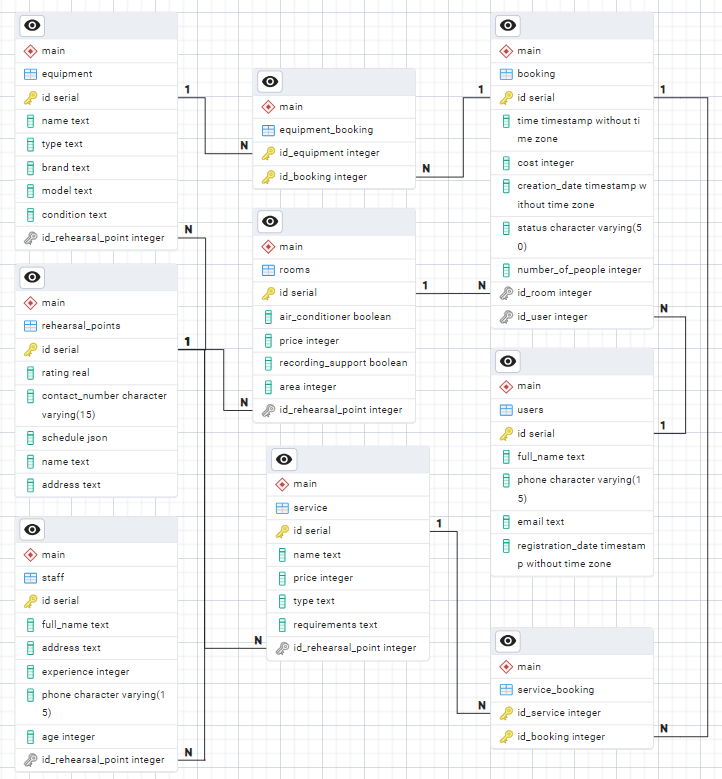


Рисунок 5.2 – Полученная ERD-диаграмма

# 6 ЭКСПОРТ РЕЗУЛЬТАТОВ

Для восстановления базы данных или создания ее копии необходимо сохранить точную структуру таблиц. В pgAdmin для этого предусмотрено специальное меню: Schema → Backup, где следует указать имя файла и выбрать формат Plain. На рисунке 6.1 представлен функционал создания резервной копии БД.

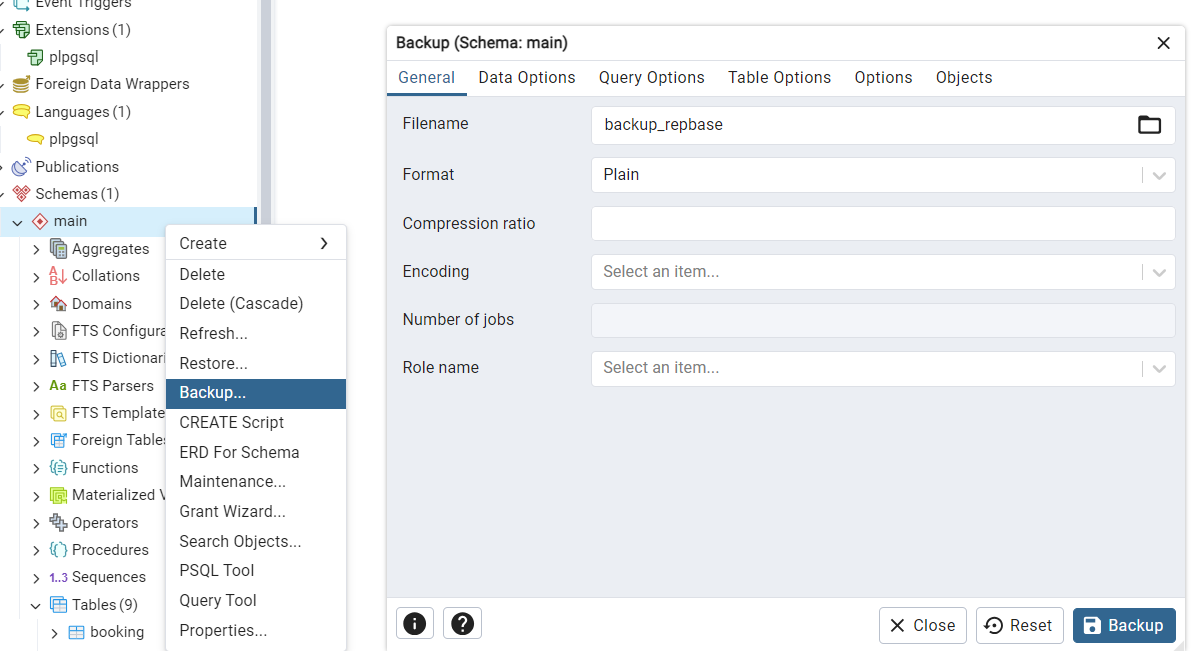


Рисунок 6.1 – Backup

Скрипт для восстановления базы данных:

--

-- PostgreSQL database dump

--

-- Dumped from database version 17.2

-- Dumped by pg\_dump version 17.2

-- Started on 2025-02-25 02:12:58

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;

--

-- TOC entry 5 (class 2615 OID 16927)

-- Name: main; Type: SCHEMA; Schema: -; Owner: postgres

--

CREATE SCHEMA main;

ALTER SCHEMA main OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

-- TOC entry 234 (class 1259 OID 16985)

-- Name: booking; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.booking (

    id integer NOT NULL,

    "time" timestamp without time zone NOT NULL,

    cost integer NOT NULL,

    creation\_date timestamp without time zone NOT NULL,

    status character varying(50) NOT NULL,

    number\_of\_people integer NOT NULL,

    id\_room integer,

    id\_user integer,

    duration integer

);

ALTER TABLE main.booking OWNER TO postgres;

--

-- TOC entry 233 (class 1259 OID 16984)

-- Name: booking\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.booking\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.booking\_id\_seq OWNER TO postgres;

--

-- TOC entry 4987 (class 0 OID 0)

-- Dependencies: 233

-- Name: booking\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.booking\_id\_seq OWNED BY main.booking.id;

--

-- TOC entry 228 (class 1259 OID 16958)

-- Name: equipment; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.equipment (

    id integer NOT NULL,

    name text NOT NULL,

    type text NOT NULL,

    brand text NOT NULL,

    model text NOT NULL,

    condition text NOT NULL,

    id\_rehearsal\_point integer

);

ALTER TABLE main.equipment OWNER TO postgres;

--

-- TOC entry 236 (class 1259 OID 16996)

-- Name: equipment\_booking; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.equipment\_booking (

    id\_equipment integer NOT NULL,

    id\_booking integer NOT NULL

);

ALTER TABLE main.equipment\_booking OWNER TO postgres;

--

-- TOC entry 227 (class 1259 OID 16957)

-- Name: equipment\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.equipment\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.equipment\_id\_seq OWNER TO postgres;

--

-- TOC entry 4988 (class 0 OID 0)

-- Dependencies: 227

-- Name: equipment\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.equipment\_id\_seq OWNED BY main.equipment.id;

--

-- TOC entry 222 (class 1259 OID 16929)

-- Name: rehearsal\_points; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.rehearsal\_points (

    id integer NOT NULL,

    rating real,

    contact\_number character varying(15) NOT NULL,

    schedule json,

    name text NOT NULL,

    address text NOT NULL

);

ALTER TABLE main.rehearsal\_points OWNER TO postgres;

--

-- TOC entry 221 (class 1259 OID 16928)

-- Name: rehearsal\_points\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.rehearsal\_points\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.rehearsal\_points\_id\_seq OWNER TO postgres;

--

-- TOC entry 4989 (class 0 OID 0)

-- Dependencies: 221

-- Name: rehearsal\_points\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.rehearsal\_points\_id\_seq OWNED BY main.rehearsal\_points.id;

--

-- TOC entry 224 (class 1259 OID 16938)

-- Name: rooms; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.rooms (

    id integer NOT NULL,

    name text NOT NULL,

    air\_conditioner boolean DEFAULT false NOT NULL,

    price integer NOT NULL,

    recording\_support boolean DEFAULT false NOT NULL,

    area integer NOT NULL,

    id\_rehearsal\_point integer

);

ALTER TABLE main.rooms OWNER TO postgres;

--

-- TOC entry 223 (class 1259 OID 16937)

-- Name: rooms\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.rooms\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.rooms\_id\_seq OWNER TO postgres;

--

-- TOC entry 4990 (class 0 OID 0)

-- Dependencies: 223

-- Name: rooms\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.rooms\_id\_seq OWNED BY main.rooms.id;

--

-- TOC entry 226 (class 1259 OID 16949)

-- Name: service; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.service (

    id integer NOT NULL,

    name text NOT NULL,

    price integer NOT NULL,

    type text NOT NULL,

    requirements text,

    id\_rehearsal\_point integer

);

ALTER TABLE main.service OWNER TO postgres;

--

-- TOC entry 235 (class 1259 OID 16991)

-- Name: service\_booking; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.service\_booking (

    id\_service integer NOT NULL,

    id\_booking integer NOT NULL

);

ALTER TABLE main.service\_booking OWNER TO postgres;

--

-- TOC entry 225 (class 1259 OID 16948)

-- Name: service\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.service\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.service\_id\_seq OWNER TO postgres;

--

-- TOC entry 4991 (class 0 OID 0)

-- Dependencies: 225

-- Name: service\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.service\_id\_seq OWNED BY main.service.id;

--

-- TOC entry 230 (class 1259 OID 16967)

-- Name: staff; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.staff (

    id integer NOT NULL,

    full\_name text NOT NULL,

    address text,

    experience integer,

    phone character varying(15) NOT NULL,

    age integer NOT NULL,

    id\_rehearsal\_point integer

);

ALTER TABLE main.staff OWNER TO postgres;

--

-- TOC entry 229 (class 1259 OID 16966)

-- Name: staff\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.staff\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.staff\_id\_seq OWNER TO postgres;

--

-- TOC entry 4992 (class 0 OID 0)

-- Dependencies: 229

-- Name: staff\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.staff\_id\_seq OWNED BY main.staff.id;

--

-- TOC entry 232 (class 1259 OID 16976)

-- Name: users; Type: TABLE; Schema: main; Owner: postgres

--

CREATE TABLE main.users (

    id integer NOT NULL,

    full\_name text NOT NULL,

    phone character varying(15) NOT NULL,

    email text NOT NULL,

    registration\_date timestamp without time zone NOT NULL

);

ALTER TABLE main.users OWNER TO postgres;

--

-- TOC entry 231 (class 1259 OID 16975)

-- Name: users\_id\_seq; Type: SEQUENCE; Schema: main; Owner: postgres

--

CREATE SEQUENCE main.users\_id\_seq

    AS integer

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1;

ALTER SEQUENCE main.users\_id\_seq OWNER TO postgres;

--

-- TOC entry 4993 (class 0 OID 0)

-- Dependencies: 231

-- Name: users\_id\_seq; Type: SEQUENCE OWNED BY; Schema: main; Owner: postgres

--

ALTER SEQUENCE main.users\_id\_seq OWNED BY main.users.id;

--

-- TOC entry 4792 (class 2604 OID 16988)

-- Name: booking id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.booking ALTER COLUMN id SET DEFAULT nextval('main.booking\_id\_seq'::regclass);

--

-- TOC entry 4789 (class 2604 OID 16961)

-- Name: equipment id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.equipment ALTER COLUMN id SET DEFAULT nextval('main.equipment\_id\_seq'::regclass);

--

-- TOC entry 4784 (class 2604 OID 16932)

-- Name: rehearsal\_points id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.rehearsal\_points ALTER COLUMN id SET DEFAULT nextval('main.rehearsal\_points\_id\_seq'::regclass);

--

-- TOC entry 4785 (class 2604 OID 16941)

-- Name: rooms id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.rooms ALTER COLUMN id SET DEFAULT nextval('main.rooms\_id\_seq'::regclass);

--

-- TOC entry 4788 (class 2604 OID 16952)

-- Name: service id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.service ALTER COLUMN id SET DEFAULT nextval('main.service\_id\_seq'::regclass);

--

-- TOC entry 4790 (class 2604 OID 16970)

-- Name: staff id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.staff ALTER COLUMN id SET DEFAULT nextval('main.staff\_id\_seq'::regclass);

--

-- TOC entry 4791 (class 2604 OID 16979)

-- Name: users id; Type: DEFAULT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.users ALTER COLUMN id SET DEFAULT nextval('main.users\_id\_seq'::regclass);

--

-- TOC entry 4979 (class 0 OID 16985)

-- Dependencies: 234

-- Data for Name: booking; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.booking (id, "time", cost, creation\_date, status, number\_of\_people, id\_room, id\_user, duration) FROM stdin;

28  2023-03-01 09:00:00 50  2023-02-28 10:00:00 active  4   1   1   2

29  2023-03-05 10:30:00 30  2023-03-04 11:00:00 canceled    2   2   2   1

30  2023-03-03 11:15:00 70  2023-03-02 12:00:00 completed   6   3   3   3

31  2023-03-07 14:00:00 25  2023-03-06 13:00:00 active  3   4   4   1

32  2023-03-02 15:45:00 45  2023-03-01 14:30:00 active  5   5   5   4

33  2023-03-04 08:30:00 15  2023-03-03 15:00:00 completed   1   6   6   2

34  2023-03-08 09:00:00 80  2023-03-07 16:00:00 canceled    7   7   7   3

35  2023-03-09 10:00:00 60  2023-03-08 17:00:00 active  8   8   8   1

36  2023-03-10 12:30:00 40  2023-03-09 18:00:00 completed   4   9   9   2

37  2023-03-11 14:15:00 55  2023-03-10 19:00:00 active  3   10  10  3

38  2023-03-12 09:30:00 20  2023-03-11 09:15:00 canceled    2   11  11  1

39  2023-03-13 11:00:00 35  2023-03-12 10:15:00 completed   6   12  12  2

40  2023-03-14 13:30:00 65  2023-03-13 11:15:00 active  5   13  13  4

41  2023-03-15 15:00:00 50  2023-03-14 12:15:00 active  1   14  14  3

42  2023-03-16 16:45:00 10  2023-03-15 13:15:00 completed   3   15  15  2

43  2023-03-17 10:00:00 75  2023-03-16 14:15:00 canceled    2   16  16  1

44  2023-03-18 11:30:00 45  2023-03-17 15:15:00 active  4   17  17  3

45  2023-03-19 12:00:00 30  2023-03-18 16:15:00 completed   8   18  18  1

46  2023-03-20 08:15:00 65  2023-03-19 17:15:00 active  7   19  19  2

47  2023-03-21 09:45:00 15  2023-03-20 18:15:00 canceled    1   20  20  4

48  2023-03-22 10:30:00 55  2023-03-21 19:15:00 completed   6   21  21  3

49  2023-03-23 11:00:00 80  2023-03-22 09:00:00 active  5   22  22  1

50  2023-03-24 12:30:00 40  2023-03-23 10:00:00 active  3   23  23  2

51  2023-03-25 14:15:00 20  2023-03-24 11:00:00 completed   2   24  24  4

52  2023-03-26 15:00:00 60  2023-03-25 12:00:00 canceled    4   25  25  3

53  2023-03-27 16:30:00 35  2023-03-26 13:00:00 active  7   26  26  2

54  2023-03-28 17:00:00 50  2023-03-27 14:00:00 completed   8   27  27  1

\.

--

-- TOC entry 4973 (class 0 OID 16958)

-- Dependencies: 228

-- Data for Name: equipment; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.equipment (id, name, type, brand, model, condition, id\_rehearsal\_point) FROM stdin;

1   Combo Amplifier Amplifier   Fender  Champion 100    New 53

2   Combo Amplifier Amplifier   Fender  Champion 100    New 53

3   Combo Amplifier Amplifier   Marshall    MG30GFX Used    53

4   Drum Kit    Drum Set    Pearl   Export Series   Used    53

5   Drum Kit    Drum Set    Tama    Starclassic New 53

6   Synthesizer Synthesizer Yamaha  MX61    New 53

7   Bass Rental Instrument  Fender  Precision Bass  New 53

8   Combo Amplifier Amplifier   Roland  CUBE-40GX   New 54

9   Combo Amplifier Amplifier   Orange  Crush Pro   Used    54

10  Drum Kit    Drum Set    Yamaha  Stage Custom    New 54

11  Combo Amplifier Amplifier   Fender  Champion 100    New 55

12  Combo Amplifier Amplifier   Marshall    MG30GFX Used    55

13  Drum Kit    Drum Set    Pearl   Export Series   Used    55

14  Drum Kit    Drum Set    Tama    Starclassic New 55

15  Synthesizer Synthesizer Yamaha  MX61    New 55

16  Bass Rental Instrument  Fender  Precision Bass  New 56

17  Combo Amplifier Amplifier   Roland  CUBE-40GX   New 57

18  Combo Amplifier Amplifier   Orange  Crush Pro   Used    57

19  Drum Kit    Drum Set    Yamaha  Stage Custom    New 57

20  Drum Kit    Drum Set    Mapex   Armory  Used    58

21  Synthesizer Synthesizer Nord    Electro 5   New 58

22  Bass Rental Instrument  Epiphone    EB-3    New 58

23  Combo Amplifier Amplifier   Roland  AC30    New 59

24  Combo Amplifier Amplifier   Vox Pathfinder  Used    59

25  Drum Kit    Drum Set    Tama    Starclassic New 59

26  Drum Kit    Drum Set    Yamaha  Live Custom New 59

27  Synthesizer Synthesizer Korg    Volca FM    New 59

28  Bass Rental Instrument  Fender  Jazz Bass   New 59

29  Combo Amplifier Amplifier   Bugera  V55 New 60

30  Combo Amplifier Amplifier   Laney   Ironheart   Used    60

31  Drum Kit    Drum Set    Gretsch New Classic New 60

32  Drum Kit    Drum Set    DW  Design Series   Used    60

33  Synthesizer Synthesizer Roland  Juno-DS New 60

34  Bass Rental Instrument  Ibanez  SR300   New 60

35  Combo Amplifier Amplifier   Marshall    Code 25 New 61

36  Combo Amplifier Amplifier   Orange  Tiny Terror Used    61

37  Drum Kit    Drum Set    Ludwig  Super Classic   New 61

38  Drum Kit    Drum Set    Pearl   Vision  Used    61

39  Synthesizer Synthesizer Yamaha  Reface DX   New 61

40  Bass Rental Instrument  Squier  Affinity    New 61

41  Combo Amplifier Amplifier   Fender  Frontman    New 62

42  Combo Amplifier Amplifier   Vox AC4 Used    62

43  Drum Kit    Drum Set    Mapex   M Series    New 62

44  Drum Kit    Drum Set    Tama    S.L.P.  Used    62

45  Synthesizer Synthesizer Korg    Monologue   New 62

46  Bass Rental Instrument  Music Man   Bongo   New 62

\.

--

-- TOC entry 4981 (class 0 OID 16996)

-- Dependencies: 236

-- Data for Name: equipment\_booking; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.equipment\_booking (id\_equipment, id\_booking) FROM stdin;

1   28

2   29

3   30

4   31

5   32

6   33

7   34

8   35

9   36

10  37

11  38

12  39

13  40

14  41

15  42

16  43

17  44

18  45

19  46

20  47

21  48

22  49

23  50

24  51

25  52

26  53

27  54

5   28

8   30

14  34

19  50

\.

--

-- TOC entry 4967 (class 0 OID 16929)

-- Dependencies: 222

-- Data for Name: rehearsal\_points; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.rehearsal\_points (id, rating, contact\_number, schedule, name, address) FROM stdin;

74  4.4 +375 29 4445566 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Studio 77   Minsk, K. Marksa St., 20

75  4.6 +375 29 5556677 {"Monday": "10:00-22:00", "Tuesday": "10:00-20:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Sound Lab   Minsk, Hikalo St., 3

76  4.5 +375 29 6667788 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-24:00", "Friday": "10:00-22:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Vibe Studio Minsk, Zmitroka Biaduli St., 8

77  4.2 +375 29 7778899 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-24:00", "Friday": "10:00-22:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Rhythm Studio   Minsk, Bohdanovicha St., 30

78  4.3 +375 29 8889900 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Echo Sound  Minsk, October St., 10

79  4.4 +375 29 9990011 {"Monday": "10:00-22:00", "Tuesday": "10:00-20:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Harmony Sound   Minsk, Lenina St., 25

80  4.5 +375 29 0001122 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-24:00", "Friday": "10:00-22:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Beat Studio Minsk, Mashinostroiteley St., 15

81  4.6 +375 29 1112233 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Music Studio    Minsk, Very Khoruzhey St., 5

82  4.3 +375 29 2223344 {"Monday": "10:00-22:00", "Tuesday": "10:00-20:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Vocal Studio    Minsk, Hikalo St., 12

83  4.4 +375 29 3334455 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Sound Wave  Minsk, Mashinostroiteley St., 18

53  4.5 +375 29 1504505 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-24:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Black Line Studio   Minsk, Fabriziusa 8

54  4.8 +375 29 5714120 {"Monday": "09:00-21:00", "Tuesday": "09:00-21:00", "Wednesday": "09:00-21:00", "Thursday": "09:00-21:00", "Friday": "09:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} BooM Studio Minsk, Lenina St., 4

55  4.6 +375 33 3036355 {"Monday": "10:00-22:00", "Tuesday": "10:00-20:00", "Wednesday": "10:00-24:00", "Thursday": "10:00-22:00", "Friday": "10:00-22:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Raga Doo Music Studio   Minsk, Very Khoruzhey St., 29

56  4.7 +375 44 7469444 {"Monday": "11:00-23:00", "Tuesday": "11:00-23:00", "Wednesday": "11:00-24:00", "Thursday": "11:00-23:00", "Friday": "11:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Studio 9    Minsk, Hikalo St., 9, 4th building BGUIR

57  4.5 +375 29 2939033 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Black Beauty Studio Minsk, Korolya St., 2

58  4.3 +375 29 7925455 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} MetraMusic  Minsk, Danila Serdicha St., 30A

59  4.6 +375 29 7608134 {"Monday": "10:00-24:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} LIVEBOX Minsk, Masherova Ave., 11

60  4.4 +375 29 7608135 {"Monday": "09:00-21:00", "Tuesday": "09:00-21:00", "Wednesday": "09:00-22:00", "Thursday": "09:00-22:00", "Friday": "09:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Chilli Records  Minsk, Zmitroka Biaduli St., 13

61  4.2 +375 29 5714121 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} JAM Studio (BooM)   Minsk, K. Marksa St., 50A

62  4.5 +375 25 9501863 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} BIG AIR STUDIO  Minsk, Bohdanovicha St., 232

63  4.7 +375 25 7792203 {"Monday": "10:00-22:00", "Tuesday": "10:00-24:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Melody Studio   Minsk, Korolya St., 2/11

64  4.6 +375 29 5528552 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-24:00", "Sunday": "10:00-20:00"} Sunset  Minsk, Mashinostroiteley St.

65  4.5 +375 29 2574410 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} OVERTONE Studio Minsk, Korolya St., 2

66  4.8 +375 25 7151359 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} JKBX Studio Minsk, Masherova Ave., 11, 5th Floor

67  4.3 +375 29 3364499 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Microfon    Minsk, Tolbukhina St., 2, office 13-5

68  4.5 +375 29 1504506 {"Monday": "10:00-22:00", "Tuesday": "10:00-24:00", "Wednesday": "10:00-22:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} New Sound Studio    Minsk, Masherova Ave., 10

69  4.4 +375 29 1234567 {"Monday": "09:00-21:00", "Tuesday": "09:00-21:00", "Wednesday": "09:00-22:00", "Thursday": "09:00-22:00", "Friday": "09:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Sound Factory   Minsk, Zmitroka Biaduli St., 1

70  4.2 +375 29 7654321 {"Monday": "11:00-23:00", "Tuesday": "11:00-23:00", "Wednesday": "11:00-24:00", "Thursday": "11:00-23:00", "Friday": "11:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Studio 55   Minsk, K. Marksa St., 12

71  4.1 +375 29 1112233 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "10:00-22:00", "Sunday": "10:00-20:00"} Music Hall  Minsk, Bohdanovicha St., 15

72  4.3 +375 29 2223344 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Harmony Studio  Minsk, Lenina St., 14

73  4.5 +375 29 3334455 {"Monday": "10:00-22:00", "Tuesday": "10:00-22:00", "Wednesday": "10:00-20:00", "Thursday": "10:00-22:00", "Friday": "10:00-24:00", "Saturday": "12:00-24:00", "Sunday": "12:00-20:00"} Echo Studio Minsk, October St., 5

\.

--

-- TOC entry 4969 (class 0 OID 16938)

-- Dependencies: 224

-- Data for Name: rooms; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.rooms (id, name, air\_conditioner, price, recording\_support, area, id\_rehearsal\_point) FROM stdin;

1   Large Room  t   50  t   35  53

2   Medium Room t   40  t   25  53

3   Small Room  f   30  f   15  53

4   Blue Room   t   45  t   30  54

5   Red Room    f   35  t   20  54

6   Green Room  t   25  f   28  54

7   Large Room  t   50  t   35  55

8   Medium Room t   45  t   25  55

9   Small Room  f   20  f   12  55

10  Yellow Room t   50  t   30  56

11  Orange Room f   35  t   18  56

12  Purple Room t   40  f   22  56

13  Large Room  t   50  t   35  57

14  Medium Room t   30  t   26  57

15  Small Room  f   15  f   14  57

16  Black Room  t   45  t   29  58

17  White Room  f   25  t   19  58

18  Gray Room   t   20  f   27  58

19  Large Room  t   50  t   35  59

20  Medium Room t   40  t   24  59

21  Small Room  f   10  f   13  59

22  Cyan Room   t   35  t   31  60

23  Magenta Room    f   30  t   21  60

24  Brown Room  t   45  f   23  60

25  Large Room  t   50  t   35  61

26  Medium Room t   40  t   27  61

27  Small Room  f   15  f   11  61

28  Light Blue Room t   25  t   30  62

29  Light Green Room    f   20  t   20  62

30  Dark Room   t   30  f   18  62

31  Meeting Room    t   35  t   22  63

32  Rehearsal Room  f   10  f   14  63

\.

--

-- TOC entry 4971 (class 0 OID 16949)

-- Dependencies: 226

-- Data for Name: service; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.service (id, name, price, type, requirements, id\_rehearsal\_point) FROM stdin;

1   Synthesizer Rental  30  Rental  None    53

2   Bass Rental 25  Rental  None    53

3   Combo Amplifier Rental  40  Rental  None    53

4   Mixing Service  150 Recording   DAW, audio interface, monitors  54

5   Mastering Service   200 Recording   High-quality monitors, specialized software 54

6   Sound Recording 180 Recording   Microphones, audio interface, soundproofing 54

7   Vocal Recording 150 Recording   Microphones, pop filter, headphones 55

8   Guitar Rental   20  Rental  None    55

9   Drum Kit Rental 35  Rental  None    55

10  Keyboard Rental 30  Rental  None    56

11  Mixing Service  160 Recording   DAW, audio interface, monitors  56

12  Live Sound Engineering  200 Recording   PA system, microphones, mixing console  56

13  Sound Design    180 Recording   Synthesizers, effects, DAW  57

14  Acoustic Treatment  50  Rental  Acoustic panels, bass traps 57

15  Microphone Rental   15  Rental  None    58

16  Piano Rental    40  Rental  None    58

17  Bass Guitar Rental  25  Rental  None    58

18  Editing Service 120 Recording   DAW, editing software   59

19  Podcast Recording   150 Recording   Microphones, soundproofing  59

20  Home Studio Setup   200 Recording   Monitors, audio interface, software 59

21  Vocal Mixing    180 Recording   DAW, effects plugins    60

22  Drum Recording  160 Recording   Microphones, soundproofing  60

23  Bass Mixing 170 Recording   DAW, effects plugins    60

24  Guitar Mixing   150 Recording   DAW, effects plugins    61

25  Loop Rental 10  Rental  None    61

26  Audio Interface Rental  25  Rental  None    62

27  Synth Programming   30  Recording   Synthesizer, DAW    62

28  Sample Library Access   20  Rental  None    62

29  Soundtrack Composition  200 Recording   DAW, instruments    63

30  Live Streaming Setup    150 Recording   Camera, microphone, streaming software  63

\.

--

-- TOC entry 4980 (class 0 OID 16991)

-- Dependencies: 235

-- Data for Name: service\_booking; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.service\_booking (id\_service, id\_booking) FROM stdin;

1   28

2   29

3   30

4   31

5   32

6   33

7   34

8   35

9   36

10  37

11  38

12  39

13  40

14  41

15  42

16  43

17  44

18  45

19  46

20  47

21  48

22  49

23  50

24  51

25  52

26  53

27  54

3   28

9   30

17  34

22  50

\.

--

-- TOC entry 4975 (class 0 OID 16967)

-- Dependencies: 230

-- Data for Name: staff; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.staff (id, full\_name, address, experience, phone, age, id\_rehearsal\_point) FROM stdin;

1   Alexey Petrov   Minsk, Lenin St., 1 5   +375291234567   30  53

2   Darya Ivanova   Minsk, Nezavisimosti Ave., 10   7   +375291234568   35  54

3   Sergey Sokolov  Minsk, Mira St., 5  4   +375291234569   28  55

6   Olga Smirnova   Minsk, Chekistov St., 30    8   +375291234572   40  58

7   Vladimir Ivanov Minsk, Gorkogo St., 12  5   +375291234573   29  59

8   Maria Orlova    Minsk, Lenin St., 25    4   +375291234574   31  60

9   Andrey Nikiforov    Minsk, Pushkina St., 10 6   +375291234575   34  61

12  Kirill Fedorov  Minsk, Kirova St., 11   2   +375291234578   25  53

13  Yulia Alexandrova   Minsk, Frunze St., 22   7   +375291234579   38  54

16  Dmitry Tikhonov Minsk, Gogolya St., 8   4   +375291234582   30  57

18  Alexandra Guseva    Minsk, Revolyutsii St., 9   3   +375291234584   28  59

19  Viktor Petrovich    Minsk, Pervomayskaya St., 14    5   +375291234585   29  60

21  Igor Morozov    Minsk, Obronaya St., 19 2   +375291234587   24  62

23  Natalya Ivanenko    Minsk, Karl Marx St., 12    4   +375291234589   26  53

25  Sofia Novik Minsk, Gagarina St., 33 9   +375291234591   41  55

26  Roman Stepanov  Minsk, Koval St., 16    3   +375291234592   27  56

27  Igor Lebedev    Minsk, Gagarina St., 10 3   +375291234604   28  63

28  Maria Morozova  Minsk, Frunze St., 25   2   +375291234611   24  56

4   Natalia Kuznetsova  Minsk, Pobedy Ave., 20  6   +375291234570   32  55

5   Igor Moroz  Minsk, Sovetskaya St., 15   3   +375291234571   26  55

10  Svetlana Pavlova    Minsk, Stalingrad St., 50   5   +375291234576   33  61

11  Antonina Sergeeva   Minsk, Chernomorskaya St., 45   9   +375291234577   36  54

14  Mikhail Gromov  Minsk, Michurina St., 32    5   +375291234580   27  54

15  Elena Romanova  Minsk, Lenin St., 3 10  +375291234581   42  61

17  Tamara Ivanova  Minsk, Gagarina St., 6  6   +375291234583   37  53

20  Anastasia Semyonova Minsk, Lenin St., 17    8   +375291234586   39  58

22  Valentina Denisova  Minsk, Sovetskaya St., 7    6   +375291234588   33  58

24  Alexander Zaytsev   Minsk, Kuybysheva St., 28   5   +375291234590   30  53

\.

--

-- TOC entry 4977 (class 0 OID 16976)

-- Dependencies: 232

-- Data for Name: users; Type: TABLE DATA; Schema: main; Owner: postgres

--

COPY main.users (id, full\_name, phone, email, registration\_date) FROM stdin;

1   Alexey Petrov   +375291234567   alexey.petrov@example.com   2023-01-15 10:00:00

2   Darya Ivanova   +375291234568   darya.ivanova@example.com   2023-01-16 11:15:00

3   Sergey Sokolov  +375291234569   sergey.sokolov@example.com  2023-01-17 12:30:00

4   Natalia Kuznetsova  +375291234570   natalia.kuznetsova@example.com  2023-01-18 13:45:00

5   Igor Moroz  +375291234571   igor.moroz@example.com  2023-01-19 14:00:00

6   Olga Smirnova   +375291234572   olga.smirnova@example.com   2023-01-20 15:10:00

7   Vladimir Ivanov +375291234573   vladimir.ivanov@example.com 2023-01-21 16:20:00

8   Maria Orlova    +375291234574   maria.orlova@example.com    2023-01-22 17:30:00

9   Andrey Nikiforov    +375291234575   andrey.nikiforov@example.com    2023-01-23 18:40:00

10  Svetlana Pavlova    +375291234576   svetlana.pavlova@example.com    2023-01-24 19:50:00

11  Antonina Sergeeva   +375291234577   antonina.sergeeva@example.com   2023-01-25 20:00:00

12  Kirill Fedorov  +375291234578   kirill.fedorov@example.com  2023-01-26 21:10:00

13  Yulia Alexandrova   +375291234579   yulia.alexandrova@example.com   2023-01-27 22:20:00

14  Mikhail Gromov  +375291234580   mikhail.gromov@example.com  2023-01-28 23:30:00

15  Elena Romanova  +375291234581   elena.romanova@example.com  2023-01-29 09:00:00

16  Dmitry Tikhonov +375291234582   dmitry.tikhonov@example.com 2023-01-30 10:05:00

17  Alexandra Guseva    +375291234583   alexandra.guseva@example.com    2023-01-31 11:15:00

18  Viktor Petrovich    +375291234584   viktor.petrovich@example.com    2023-02-01 12:20:00

19  Anastasia Semyonova +375291234585   anastasia.semyonova@example.com 2023-02-02 13:25:00

20  Maxim Frolov    +375291234600   maxim.frolov@example.com    2023-02-03 14:30:00

21  Anna Frolova    +375291234601   anna.frolova@example.com    2023-02-04 15:35:00

22  Dmitry Sergeyev +375291234602   dmitry.sergeyev@example.com 2023-02-05 16:40:00

23  Tatiana Sergeyeva   +375291234603   tatiana.sergeyeva@example.com   2023-02-06 17:45:00

24  Igor Lebedev    +375291234604   igor.lebedev@example.com    2023-02-07 18:50:00

25  Svetlana Lebedeva   +375291234605   svetlana.lebedeva@example.com   2023-02-08 19:55:00

26  Yuri Smirnov    +375291234606   yuri.smirnov@example.com    2023-02-09 20:00:00

27  Elena Smirnova  +375291234607   elena.smirnova@example.com  2023-02-10 21:05:00

28  Vladimir Petrov +375291234608   vladimir.petrov@example.com 2023-02-11 22:10:00

29  Olga Petrova    +375291234609   olga.petrova@example.com    2023-02-12 23:15:00

30  Andrey Morozov  +375291234610   andrey.morozov@example.com  2023-02-13 09:20:00

31  Maria Morozova  +375291234611   maria.morozova@example.com  2023-02-14 10:25:00

\.

--

-- TOC entry 4994 (class 0 OID 0)

-- Dependencies: 233

-- Name: booking\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.booking\_id\_seq', 54, true);

--

-- TOC entry 4995 (class 0 OID 0)

-- Dependencies: 227

-- Name: equipment\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.equipment\_id\_seq', 48, true);

--

-- TOC entry 4996 (class 0 OID 0)

-- Dependencies: 221

-- Name: rehearsal\_points\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.rehearsal\_points\_id\_seq', 83, true);

--

-- TOC entry 4997 (class 0 OID 0)

-- Dependencies: 223

-- Name: rooms\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.rooms\_id\_seq', 32, true);

--

-- TOC entry 4998 (class 0 OID 0)

-- Dependencies: 225

-- Name: service\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.service\_id\_seq', 30, true);

--

-- TOC entry 4999 (class 0 OID 0)

-- Dependencies: 229

-- Name: staff\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.staff\_id\_seq', 28, true);

--

-- TOC entry 5000 (class 0 OID 0)

-- Dependencies: 231

-- Name: users\_id\_seq; Type: SEQUENCE SET; Schema: main; Owner: postgres

--

SELECT pg\_catalog.setval('main.users\_id\_seq', 31, true);

--

-- TOC entry 4806 (class 2606 OID 16990)

-- Name: booking booking\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.booking

    ADD CONSTRAINT booking\_pkey PRIMARY KEY (id);

--

-- TOC entry 4810 (class 2606 OID 17000)

-- Name: equipment\_booking equipment\_booking\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.equipment\_booking

    ADD CONSTRAINT equipment\_booking\_pkey PRIMARY KEY (id\_equipment, id\_booking);

--

-- TOC entry 4800 (class 2606 OID 16965)

-- Name: equipment equipment\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.equipment

    ADD CONSTRAINT equipment\_pkey PRIMARY KEY (id);

--

-- TOC entry 4794 (class 2606 OID 16936)

-- Name: rehearsal\_points rehearsal\_points\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.rehearsal\_points

    ADD CONSTRAINT rehearsal\_points\_pkey PRIMARY KEY (id);

--

-- TOC entry 4796 (class 2606 OID 16947)

-- Name: rooms rooms\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.rooms

    ADD CONSTRAINT rooms\_pkey PRIMARY KEY (id);

--

-- TOC entry 4808 (class 2606 OID 16995)

-- Name: service\_booking service\_booking\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.service\_booking

    ADD CONSTRAINT service\_booking\_pkey PRIMARY KEY (id\_service, id\_booking);

--

-- TOC entry 4798 (class 2606 OID 16956)

-- Name: service service\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.service

    ADD CONSTRAINT service\_pkey PRIMARY KEY (id);

--

-- TOC entry 4802 (class 2606 OID 16974)

-- Name: staff staff\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.staff

    ADD CONSTRAINT staff\_pkey PRIMARY KEY (id);

--

-- TOC entry 4804 (class 2606 OID 16983)

-- Name: users users\_pkey; Type: CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.users

    ADD CONSTRAINT users\_pkey PRIMARY KEY (id);

--

-- TOC entry 4815 (class 2606 OID 17021)

-- Name: booking fk\_booking\_room; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.booking

    ADD CONSTRAINT fk\_booking\_room FOREIGN KEY (id\_room) REFERENCES main.rooms(id) ON DELETE SET NULL;

--

-- TOC entry 4816 (class 2606 OID 17026)

-- Name: booking fk\_booking\_user; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.booking

    ADD CONSTRAINT fk\_booking\_user FOREIGN KEY (id\_user) REFERENCES main.users(id) ON DELETE SET NULL;

--

-- TOC entry 4819 (class 2606 OID 17046)

-- Name: equipment\_booking fk\_equipment\_booking\_booking; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.equipment\_booking

    ADD CONSTRAINT fk\_equipment\_booking\_booking FOREIGN KEY (id\_booking) REFERENCES main.booking(id) ON DELETE SET NULL

--

-- TOC entry 4820 (class 2606 OID 17041)

-- Name: equipment\_booking fk\_equipment\_booking\_equipment; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.equipment\_booking

    ADD CONSTRAINT fk\_equipment\_booking\_equipment FOREIGN KEY (id\_equipment) REFERENCES main.equipment(id) ON DELETE SET NULL;

--

-- TOC entry 4813 (class 2606 OID 17011)

-- Name: equipment fk\_equipment\_rehearsal\_point; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.equipment

    ADD CONSTRAINT fk\_equipment\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point) REFERENCES main.rehearsal\_points(id) ON DELETE CASCADE;

--

-- TOC entry 4811 (class 2606 OID 17001)

-- Name: rooms fk\_rooms\_rehearsal\_point; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.rooms

    ADD CONSTRAINT fk\_rooms\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point) REFERENCES main.rehearsal\_points(id) ON DELETE CASCADE;

--

-- TOC entry 4817 (class 2606 OID 17036)

-- Name: service\_booking fk\_service\_booking\_booking; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.service\_booking

    ADD CONSTRAINT fk\_service\_booking\_booking FOREIGN KEY (id\_booking) REFERENCES main.booking(id) ON DELETE SET NULL;

--

-- TOC entry 4818 (class 2606 OID 17031)

-- Name: service\_booking fk\_service\_booking\_service; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.service\_booking

    ADD CONSTRAINT fk\_service\_booking\_service FOREIGN KEY (id\_service) REFERENCES main.service(id) ON DELETE SET NULL;

--

-- TOC entry 4812 (class 2606 OID 17006)

-- Name: service fk\_service\_rehearsal\_point; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.service

    ADD CONSTRAINT fk\_service\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point) REFERENCES main.rehearsal\_points(id) ON DELETE CASCADE;

--

-- TOC entry 4814 (class 2606 OID 17016)

-- Name: staff fk\_staff\_rehearsal\_point; Type: FK CONSTRAINT; Schema: main; Owner: postgres

--

ALTER TABLE ONLY main.staff

    ADD CONSTRAINT fk\_staff\_rehearsal\_point FOREIGN KEY (id\_rehearsal\_point) REFERENCES main.rehearsal\_points(id) ON DELETE SET NULL;

-- Completed on 2025-02-25 02:12:59

--

-- PostgreSQL database dump complete

# ЗАКЛЮЧЕНИЕ

В процессе выполнения лабораторной работы были успешно решены все поставленные задачи, включая разработку новой схемы данных, проектирование и наполнение таблиц базы данных, а также проверку их взаимосвязей. Получены навыки работы с SQL-операторами, такими как CREATE TABLE, INSERT и ALTER TABLE, что позволило изменять структуру данных в соответствии с требованиями задания. Экспорт результатов в SQL-скрипт позволил сравнить полученные данные с эталонными, что подтвердило правильность выполнения всех операций. Работа помогла углубить понимание принципов проектирования и управления реляционными базами данных, а также улучшить навыки написания SQL-запросов для создания, модификации и наполнения таблиц.