Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Мегафакультет трансляционных информационных технологий

Факультет инфокоммуникационных технологий

Дисциплина: Базы Данных

**Отчет по Лабораторной работе №1**

«Создание БД в СУБД PostgreSQL. Резервное копирование и восстановление БД»

Выполнила: Микулина Алиса Романовна

Группа: K32421, 2 курс

Преподаватель: Говорова Марина Михайловна

Санкт-Петербург

2023

**Цель работы:** овладеть практическими навыками создания таблиц базы данных PostgreSQL 1Х, заполнения их рабочими данными, резервного копирования и восстановления БД.

**Практическое задание:**

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: *Primary Key, Unique, Check, Foreign Key*.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД.

*Указание:*

*Создать две резервные копии:*

* *с расширением CUSTOM для восстановления БД;*
* *с расширением PLAIN для листинга (в отчете);*
* *при создании резервных копий БД настроить параметры Dump options для Type of objects**и**Queries.*

1. Восстановить БД.

**Ход выполнения работы:**

Название БД:

БД «Автовокзал».

Схема логической модели базы данных, сгенерированная Generate ERD:

A picture containing text, diagram, parallel, number

Description automatically generated

pg\_dump (plain)

*--*

*-- PostgreSQL database dump*

*--*

*-- Dumped from database version 11.19*

*-- Dumped by pg\_dump version 11.19*

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_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;

*--*

*-- Name: bs; Type: SCHEMA; Schema: -; Owner: postgres*

*--*

CREATE SCHEMA bs;

ALTER SCHEMA bs OWNER TO postgres;

*--*

*-- Name: update\_seat\_number(); Type: FUNCTION; Schema: public; Owner: postgres*

*--*

CREATE FUNCTION public.update\_seat\_number() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

DECLARE

  seat\_number INT;

BEGIN

  SELECT num\_seat INTO seat\_number FROM bs.seat WHERE id = NEW.seat\_id;

  NEW.seat\_number := seat\_number;

  RETURN NEW;

END;

$$;

ALTER FUNCTION public.update\_seat\_number() OWNER TO postgres;

SET default\_tablespace = '';

SET default\_with\_oids = false;

*--*

*-- Name: bus; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.bus (

    number integer NOT NULL,

    type\_id integer NOT NULL,

    state\_num character(128) NOT NULL,

    manufacture\_year integer NOT NULL,

    CONSTRAINT manufacture\_year\_constraint CHECK ((manufacture\_year >= 1920))

);

ALTER TABLE bs.bus OWNER TO postgres;

*--*

*-- Name: bus\_number\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.bus ALTER COLUMN number ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.bus\_number\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: bus\_stop; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.bus\_stop (

    id integer NOT NULL,

    address character varying(256) NOT NULL,

    name character varying(128) NOT NULL

);

ALTER TABLE bs.bus\_stop OWNER TO postgres;

*--*

*-- Name: bus\_stop\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.bus\_stop ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.bus\_stop\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: bus\_type; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.bus\_type (

    id integer NOT NULL,

    size character varying(20) DEFAULT 'average'::character varying NOT NULL,

    luggage\_compartment character varying(3) NOT NULL,

    toilet character varying(3) NOT NULL,

    num\_seats integer NOT NULL,

    brand character varying(128),

    manufacturer character varying(128),

    manufacturer\_country character varying(128),

    CONSTRAINT bus\_type\_luggage\_compartment\_check CHECK (((luggage\_compartment)::text = ANY ((ARRAY['yes'::character varying, 'no'::character varying])::text[]))),

    CONSTRAINT bus\_type\_size\_check CHECK (((size)::text = ANY ((ARRAY['very small'::character varying, 'small'::character varying, 'average'::character varying, 'big'::character varying, 'very big'::character varying])::text[]))),

    CONSTRAINT bus\_type\_toilet\_check CHECK (((toilet)::text = ANY ((ARRAY['yes'::character varying, 'no'::character varying])::text[])))

);

ALTER TABLE bs.bus\_type OWNER TO postgres;

*--*

*-- Name: bus\_type\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.bus\_type ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.bus\_type\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: crew; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.crew (

    id integer NOT NULL,

    med\_exam\_date timestamp without time zone NOT NULL,

    admission\_status character varying(3) NOT NULL,

    driver\_id integer NOT NULL,

    CONSTRAINT crew\_admission\_status\_check CHECK (((admission\_status)::text = ANY ((ARRAY['yes'::character varying, 'no'::character varying])::text[])))

);

ALTER TABLE bs.crew OWNER TO postgres;

*--*

*-- Name: crew\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.crew ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.crew\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: driver; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.driver (

    id integer NOT NULL,

    full\_name character(128) NOT NULL,

    passport character(256) NOT NULL,

    phone\_number bigint,

    e\_mail character(128)

);

ALTER TABLE bs.driver OWNER TO postgres;

*--*

*-- Name: driver\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.driver ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.driver\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: flight; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.flight (

    number integer NOT NULL,

    date timestamp without time zone NOT NULL,

    departure\_time time without time zone NOT NULL,

    arrival\_time time without time zone NOT NULL,

    status character(128),

    bus\_num integer NOT NULL,

    crew\_id integer NOT NULL,

    route\_id integer NOT NULL,

    CONSTRAINT arrival\_departure\_time\_constraint CHECK ((departure\_time < arrival\_time)),

    CONSTRAINT flight\_date\_check CHECK ((date >= '2010-01-01 00:00:00'::timestamp without time zone))

);

ALTER TABLE bs.flight OWNER TO postgres;

*--*

*-- Name: flight\_number\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.flight ALTER COLUMN number ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.flight\_number\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: passenger; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.passenger (

    id integer NOT NULL,

    full\_name character(128) NOT NULL,

    passport character(256) NOT NULL,

    phone\_number bigint,

    e\_mail character(128)

);

ALTER TABLE bs.passenger OWNER TO postgres;

*--*

*-- Name: passenger\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.passenger ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.passenger\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: route; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.route (

    id integer NOT NULL,

    weekdays text[] NOT NULL,

    distance\_status character varying(13) NOT NULL,

    departure\_time time without time zone NOT NULL,

    travel\_time interval NOT NULL,

    departure\_point character varying(128) NOT NULL,

    arrival\_point character varying(128) NOT NULL,

    distance integer NOT NULL,

    CONSTRAINT distance\_constraint CHECK ((distance > 0)),

    CONSTRAINT route\_departure\_time\_check CHECK (((departure\_time >= '00:00:00'::time without time zone) AND (departure\_time <= '23:59:59'::time without time zone))),

    CONSTRAINT route\_distance\_status\_check CHECK (((distance\_status)::text = ANY ((ARRAY['urban'::character varying, 'suburban'::character varying, 'intercity'::character varying, 'international'::character varying])::text[]))),

    CONSTRAINT travel\_time\_constraint CHECK ((travel\_time > '00:00:00'::interval))

);

ALTER TABLE bs.route OWNER TO postgres;

*--*

*-- Name: route\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.route ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.route\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: seat; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.seat (

    id integer NOT NULL,

    num\_seat integer NOT NULL,

    taken\_status character varying(3) NOT NULL,

    flight\_id integer NOT NULL,

    CONSTRAINT seat\_taken\_status\_check CHECK (((taken\_status)::text = ANY ((ARRAY['yes'::character varying, 'no'::character varying])::text[])))

);

ALTER TABLE bs.seat OWNER TO postgres;

*--*

*-- Name: seat\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.seat ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.seat\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: ticket; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.ticket (

    number integer NOT NULL,

    departure\_time time without time zone NOT NULL,

    arrival\_time time without time zone NOT NULL,

    cancellation\_status character(3) NOT NULL,

    purchase\_type character(7) NOT NULL,

    flight\_num integer NOT NULL,

    passenger\_id integer NOT NULL,

    departure\_stop\_id integer NOT NULL,

    arrival\_stop\_id integer NOT NULL,

    cost integer NOT NULL,

    seat\_id integer NOT NULL,

    seat\_number integer NOT NULL,

    CONSTRAINT departure\_arrival\_stops\_constraint CHECK ((departure\_stop\_id <> arrival\_stop\_id)),

    CONSTRAINT ticket\_cancellation\_status\_check CHECK ((cancellation\_status = ANY (ARRAY['yes'::bpchar, 'no'::bpchar]))),

    CONSTRAINT ticket\_purchase\_type\_check CHECK ((purchase\_type = ANY (ARRAY['online'::bpchar, 'offline'::bpchar]))),

    CONSTRAINT time\_constraint CHECK ((departure\_time < arrival\_time))

);

ALTER TABLE bs.ticket OWNER TO postgres;

*--*

*-- Name: ticket\_number\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.ticket ALTER COLUMN number ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.ticket\_number\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Name: way\_stop; Type: TABLE; Schema: bs; Owner: postgres*

*--*

CREATE TABLE bs.way\_stop (

    id integer NOT NULL,

    global\_id integer NOT NULL,

    route\_id integer NOT NULL,

    departure\_time time without time zone NOT NULL,

    arrival\_time time without time zone NOT NULL,

    duration interval NOT NULL,

    CONSTRAINT arrival\_departure\_constraint CHECK ((arrival\_time <= departure\_time)),

    CONSTRAINT duration\_constraint CHECK ((duration >= '00:00:00'::interval)),

    CONSTRAINT way\_stop\_departure\_time\_check CHECK (((departure\_time >= '00:00:00'::time without time zone) AND (departure\_time <= '23:59:59'::time without time zone))),

    CONSTRAINT way\_stop\_departure\_time\_check1 CHECK (((departure\_time >= '00:00:00'::time without time zone) AND (departure\_time <= '23:59:59'::time without time zone)))

);

ALTER TABLE bs.way\_stop OWNER TO postgres;

*--*

*-- Name: way\_stop\_id\_seq; Type: SEQUENCE; Schema: bs; Owner: postgres*

*--*

ALTER TABLE bs.way\_stop ALTER COLUMN id ADD GENERATED ALWAYS AS IDENTITY (

    SEQUENCE NAME bs.way\_stop\_id\_seq

    START WITH 1

    INCREMENT BY 1

    NO MINVALUE

    NO MAXVALUE

    CACHE 1

);

*--*

*-- Data for Name: bus; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (1, 1, 'ABC123', 2015);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (2, 3, 'XYZ789', 2018);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (3, 4, 'GHI456', 2020);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (4, 5, 'JKL321', 2016);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (5, 6, 'MNO654', 2019);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (6, 7, 'PQR987', 2017);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (7, 8, 'STU246', 2014);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (8, 9, 'VWX369', 2013);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (9, 10, 'YUI852', 2012);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (10, 11, 'ZXC741', 2011);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (11, 12, 'POI098', 2010);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (12, 13, 'LKJ765', 2009);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (13, 14, 'HGF432', 2008);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (14, 15, 'EDC159', 2007);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (15, 16, 'QAZ246', 2006);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (16, 17, 'WSX369', 2005);

INSERT INTO bs.bus (number, type\_id, state\_num, manufacture\_year) OVERRIDING SYSTEM VALUE VALUES (17, 18, 'RFV852', 2004);

*--*

*-- Data for Name: bus\_stop; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (1, '123 Main St.', 'Main St. and 1st Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (2, '456 Elm St.', 'Elm St. and 2nd Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (3, '789 Oak St.', 'Oak St. and 3rd Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (4, '1011 Maple St.', 'Maple St. and 4th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (5, '1213 Pine St.', 'Pine St. and 5th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (6, '1415 Cedar St.', 'Cedar St. and 6th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (7, '1617 Birch St.', 'Birch St. and 7th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (8, '1819 Walnut St.', 'Walnut St. and 8th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (9, '2021 Cherry St.', 'Cherry St. and 9th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (10, '2223 Poplar St.', 'Poplar St. and 10th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (11, '2425 Chestnut St.', 'Chestnut St. and 11th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (12, '2627 Spruce St.', 'Spruce St. and 12th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (13, '2829 Sycamore St.', 'Sycamore St. and 13th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (14, '3031 Pine St.', 'Pine St. and 14th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (15, '3233 Maple St.', 'Maple St. and 15th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (16, '3435 Oak St.', 'Oak St. and 16th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (17, '3637 Elm St.', 'Elm St. and 17th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (18, '3839 Main St.', 'Main St. and 18th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (19, '4041 Walnut St.', 'Walnut St. and 19th Ave.');

INSERT INTO bs.bus\_stop (id, address, name) OVERRIDING SYSTEM VALUE VALUES (20, '4243 Cherry St.', 'Cherry St. and 20th Ave.');

*--*

*-- Data for Name: bus\_type; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (1, 'big', 'yes', 'yes', 50, 'Mercedes', 'Daimler AG', 'Germany');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (3, 'average', 'yes', 'yes', 30, 'Volvo', 'Volvo Buses', 'Sweden');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (4, 'small', 'no', 'no', 20, 'Volkswagen', 'Volkswagen Commercial Vehicles', 'Germany');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (5, 'very big', 'yes', 'yes', 75, 'Scania', 'Scania AB', 'Sweden');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (6, 'average', 'no', 'no', 40, 'MAN', 'MAN Truck & Bus', 'Germany');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (7, 'big', 'yes', 'yes', 55, 'Iveco', 'Iveco Bus', 'Italy');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (8, 'average', 'yes', 'no', 35, 'Neoplan', 'Neoplan Bus GmbH', 'Germany');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (9, 'small', 'no', 'no', 25, 'Opel', 'Opel/Vauxhall', 'Germany');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (10, 'very small', 'no', 'no', 15, 'Peugeot', 'Peugeot S.A.', 'France');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (11, 'big', 'yes', 'yes', 60, 'Setra', 'Setra Buses and Coaches', 'Germany');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (12, 'average', 'yes', 'yes', 30, 'Van Hool', 'Van Hool N.V.', 'Belgium');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (13, 'big', 'yes', 'yes', 65, 'Dennis', 'Dennis Specialist Vehicles', 'United Kingdom');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (14, 'average', 'no', 'no', 40, 'Ashok Leyland', 'Ashok Leyland Limited', 'India');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (15, 'very big', 'yes', 'yes', 80, 'MCI', 'Motor Coach Industries', 'Canada');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (16, 'average', 'yes', 'yes', 35, 'Orion', 'Orion Bus Industries', 'Canada');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (17, 'very small', 'no', 'no', 12, 'Isuzu', 'Isuzu Motors', 'Japan');

INSERT INTO bs.bus\_type (id, size, luggage\_compartment, toilet, num\_seats, brand, manufacturer, manufacturer\_country) OVERRIDING SYSTEM VALUE VALUES (18, 'big', 'yes', 'yes', 70, 'Bova', 'Bova', 'Netherlands');

*--*

*-- Data for Name: crew; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (1, '2022-01-01 10:00:00', 'yes', 1);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (2, '2022-01-02 11:00:00', 'no', 2);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (3, '2022-01-03 12:00:00', 'yes', 3);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (4, '2022-01-04 13:00:00', 'yes', 4);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (5, '2022-01-05 14:00:00', 'no', 5);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (6, '2022-01-06 15:00:00', 'no', 6);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (7, '2022-01-07 16:00:00', 'yes', 7);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (8, '2022-01-08 17:00:00', 'yes', 8);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (9, '2022-01-09 18:00:00', 'yes', 9);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (10, '2022-01-10 19:00:00', 'no', 10);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (11, '2022-01-11 20:00:00', 'yes', 11);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (12, '2022-01-12 21:00:00', 'no', 12);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (13, '2022-01-13 22:00:00', 'yes', 13);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (14, '2022-01-14 23:00:00', 'yes', 14);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (15, '2022-01-15 10:00:00', 'no', 15);

INSERT INTO bs.crew (id, med\_exam\_date, admission\_status, driver\_id) OVERRIDING SYSTEM VALUE VALUES (16, '2022-01-16 01:00:00', 'yes', 16);

*--*

*-- Data for Name: driver; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (1, 'Ivanov Matvey Sergeevich', '0987 654321, given in Saratov oblast', 89198358347, 'mrjoulin@yandex.ru');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (2, 'Ivan Ivanov', '1234567890', 79123456789, 'ivanov@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (3, 'Dmitry Petrov', '0987654321', 79234567890, 'petrov@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (4, 'Sergey Sidorov', '1111111111', 79345678901, 'sidorov@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (5, 'Maria Ivanova', '2222222222', 79456789012, 'ivanova@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (6, 'Anna Petrova', '3333333333', 79567890123, 'petrova@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (7, 'Olga Sidorova', '4444444444', 79678901234, 'sidorova@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (8, 'Pavel Ivanov', '5555555555', 79789012345, 'pavel@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (9, 'Maxim Petrov', '6666666666', 79890123456, 'maxim@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (10, 'Irina Sidorova', '7777777777', 79901234567, 'irina@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (11, 'Elena Ivanova', '8888888888', 70012345678, 'elena@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (12, 'Nikita Petrov', '9999999999', 70123456789, 'nikita@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (13, 'Vladimir Sidorov', '0000000001', 70234567890, 'vladimir@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (14, 'Alexey Ivanov', '0000000002', 70345678901, 'alexey@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (15, 'Svetlana Petrova', '0000000003', 70456789012, 'svetlana@mail.com');

INSERT INTO bs.driver (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (16, 'Ekaterina Sidorova', '0000000004', 70567890123, 'ekaterina@mail.com');

*--*

*-- Data for Name: flight; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (8, '2022-01-01 00:00:00', '10:00:00', '12:00:00', 'scheduled', 1, 1, 1);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (9, '2022-01-02 00:00:00', '12:00:00', '14:00:00', 'scheduled', 2, 2, 2);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (10, '2022-01-03 00:00:00', '14:00:00', '16:00:00', 'scheduled', 3, 3, 3);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (11, '2022-01-04 00:00:00', '16:00:00', '18:00:00', 'scheduled', 4, 4, 4);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (12, '2022-01-05 00:00:00', '18:00:00', '20:00:00', 'scheduled', 5, 5, 5);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (13, '2022-01-06 00:00:00', '20:00:00', '22:00:00', 'scheduled', 6, 6, 6);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (14, '2022-01-07 00:00:00', '22:00:00', '23:00:00', 'scheduled', 7, 7, 7);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (15, '2022-01-08 00:00:00', '00:00:00', '02:00:00', 'scheduled', 8, 8, 1);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (16, '2022-01-09 00:00:00', '02:00:00', '04:00:00', 'scheduled', 9, 9, 2);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (17, '2022-01-10 00:00:00', '04:00:00', '06:00:00', 'scheduled', 10, 10, 3);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (18, '2022-01-11 00:00:00', '06:00:00', '08:00:00', 'scheduled', 11, 11, 4);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (19, '2022-01-12 00:00:00', '08:00:00', '10:00:00', 'scheduled', 12, 12, 5);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (20, '2022-01-13 00:00:00', '10:00:00', '12:00:00', 'scheduled', 13, 13, 6);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (21, '2022-01-14 00:00:00', '12:00:00', '14:00:00', 'scheduled', 14, 14, 7);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (22, '2022-01-15 00:00:00', '14:00:00', '16:00:00', 'scheduled', 15, 15, 1);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (23, '2022-01-16 00:00:00', '16:00:00', '18:00:00', 'scheduled', 16, 16, 2);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (24, '2022-01-17 00:00:00', '18:00:00', '20:00:00', 'scheduled', 17, 1, 3);

INSERT INTO bs.flight (number, date, departure\_time, arrival\_time, status, bus\_num, crew\_id, route\_id) OVERRIDING SYSTEM VALUE VALUES (25, '2022-01-18 00:00:00', '20:00:00', '22:00:00', 'scheduled', 1, 2, 4);

*--*

*-- Data for Name: passenger; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.passenger (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (1, 'Mikulina Alice Romanovna', '6117 002651, given in Ryazan oblast', 89036417779, 'alisa.mikulina@yandex.ru ');

INSERT INTO bs.passenger (id, full\_name, passport, phone\_number, e\_mail) OVERRIDING SYSTEM VALUE VALUES (2, 'Pankova Christina Sergeevna', '1234 567890, given in Moscow oblast', 89772852541, 'punkris@yandex.ru');

*--*

*-- Data for Name: route; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (1, '{Monday,Wednesday,Friday}', 'urban', '08:00:00', '01:30:00', 'City Center', 'Suburbia', 30);

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (2, '{Tuesday,Thursday}', 'suburban', '09:30:00', '02:00:00', 'Suburbia', 'City Center', 40);

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (3, '{Monday,Wednesday,Friday,Sunday}', 'intercity', '13:00:00', '03:30:00', 'City A', 'City B', 200);

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (4, '{Tuesday,Thursday,Saturday}', 'intercity', '10:00:00', '04:00:00', 'City B', 'City C', 250);

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (5, '{Monday,Wednesday,Friday,Sunday}', 'international', '06:00:00', '10:00:00', 'City A', 'City D', 1000);

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (6, '{Tuesday,Thursday,Saturday}', 'international', '12:00:00', '12:00:00', 'City C', 'City D', 1500);

INSERT INTO bs.route (id, weekdays, distance\_status, departure\_time, travel\_time, departure\_point, arrival\_point, distance) OVERRIDING SYSTEM VALUE VALUES (7, '{Monday}', 'urban', '07:30:00', '00:45:00', 'City Center', 'Business District', 10);

*--*

*-- Data for Name: seat; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (1, 1, 'yes', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (2, 2, 'no', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (3, 3, 'yes', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (4, 4, 'no', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (5, 5, 'yes', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (6, 6, 'no', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (7, 7, 'yes', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (8, 8, 'no', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (9, 9, 'yes', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (10, 10, 'no', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (11, 11, 'yes', 23);

INSERT INTO bs.seat (id, num\_seat, taken\_status, flight\_id) OVERRIDING SYSTEM VALUE VALUES (12, 12, 'no', 23);

*--*

*-- Data for Name: ticket; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.ticket (number, departure\_time, arrival\_time, cancellation\_status, purchase\_type, flight\_num, passenger\_id, departure\_stop\_id, arrival\_stop\_id, cost, seat\_id, seat\_number) OVERRIDING SYSTEM VALUE VALUES (1, '09:30:00', '11:30:00', 'no ', 'online ', 23, 1, 3, 16, 400, 1, 1);

INSERT INTO bs.ticket (number, departure\_time, arrival\_time, cancellation\_status, purchase\_type, flight\_num, passenger\_id, departure\_stop\_id, arrival\_stop\_id, cost, seat\_id, seat\_number) OVERRIDING SYSTEM VALUE VALUES (2, '09:30:00', '11:30:00', 'no ', 'online ', 23, 1, 3, 16, 400, 3, 3);

INSERT INTO bs.ticket (number, departure\_time, arrival\_time, cancellation\_status, purchase\_type, flight\_num, passenger\_id, departure\_stop\_id, arrival\_stop\_id, cost, seat\_id, seat\_number) OVERRIDING SYSTEM VALUE VALUES (3, '09:30:00', '11:30:00', 'no ', 'offline', 23, 1, 3, 16, 400, 5, 5);

INSERT INTO bs.ticket (number, departure\_time, arrival\_time, cancellation\_status, purchase\_type, flight\_num, passenger\_id, departure\_stop\_id, arrival\_stop\_id, cost, seat\_id, seat\_number) OVERRIDING SYSTEM VALUE VALUES (4, '09:30:00', '11:30:00', 'no ', 'online ', 23, 1, 3, 16, 400, 7, 7);

INSERT INTO bs.ticket (number, departure\_time, arrival\_time, cancellation\_status, purchase\_type, flight\_num, passenger\_id, departure\_stop\_id, arrival\_stop\_id, cost, seat\_id, seat\_number) OVERRIDING SYSTEM VALUE VALUES (5, '09:30:00', '11:30:00', 'no ', 'offline', 23, 1, 3, 16, 400, 9, 9);

INSERT INTO bs.ticket (number, departure\_time, arrival\_time, cancellation\_status, purchase\_type, flight\_num, passenger\_id, departure\_stop\_id, arrival\_stop\_id, cost, seat\_id, seat\_number) OVERRIDING SYSTEM VALUE VALUES (6, '09:30:00', '11:30:00', 'no ', 'online ', 23, 1, 3, 16, 400, 11, 11);

*--*

*-- Data for Name: way\_stop; Type: TABLE DATA; Schema: bs; Owner: postgres*

*--*

INSERT INTO bs.way\_stop (id, global\_id, route\_id, departure\_time, arrival\_time, duration) OVERRIDING SYSTEM VALUE VALUES (1, 1, 1, '08:12:00', '08:10:00', '00:02:00');

INSERT INTO bs.way\_stop (id, global\_id, route\_id, departure\_time, arrival\_time, duration) OVERRIDING SYSTEM VALUE VALUES (3, 5, 6, '14:40:00', '14:20:00', '00:20:00');

*-- pg\_catalog*

*/\**

*SELECT pg\_catalog.setval('bs.bus\_number\_seq', 17, true);*

*SELECT pg\_catalog.setval('bs.bus\_stop\_id\_seq', 20, true);*

*SELECT pg\_catalog.setval('bs.bus\_type\_id\_seq', 18, true);*

*SELECT pg\_catalog.setval('bs.crew\_id\_seq', 16, true);*

*SELECT pg\_catalog.setval('bs.driver\_id\_seq', 16, true);*

*SELECT pg\_catalog.setval('bs.flight\_number\_seq', 25, true);*

*SELECT pg\_catalog.setval('bs.passenger\_id\_seq', 2, true);*

*SELECT pg\_catalog.setval('bs.route\_id\_seq', 7, true);*

*SELECT pg\_catalog.setval('bs.seat\_id\_seq', 12, true);*

*SELECT pg\_catalog.setval('bs.ticket\_number\_seq', 6, true);*

*SELECT pg\_catalog.setval('bs.way\_stop\_id\_seq', 3, true); \*/*

*-- ограничения целостности*

ALTER TABLE ONLY bs.bus

    ADD CONSTRAINT bus\_number\_pkey PRIMARY KEY (number);

ALTER TABLE ONLY bs.bus\_stop

    ADD CONSTRAINT bus\_stop\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.bus\_type

    ADD CONSTRAINT bus\_type\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.crew

    ADD CONSTRAINT crew\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.driver

    ADD CONSTRAINT driver\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.flight

    ADD CONSTRAINT flight\_pkey PRIMARY KEY (number);

ALTER TABLE ONLY bs.passenger

    ADD CONSTRAINT passenger\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.route

    ADD CONSTRAINT route\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.seat

    ADD CONSTRAINT seat\_pkey PRIMARY KEY (id);

ALTER TABLE ONLY bs.ticket

    ADD CONSTRAINT ticket\_pkey PRIMARY KEY (number);

ALTER TABLE ONLY bs.driver

    ADD CONSTRAINT unique\_d\_email UNIQUE (e\_mail);

ALTER TABLE ONLY bs.driver

    ADD CONSTRAINT unique\_d\_passport UNIQUE (passport);

ALTER TABLE ONLY bs.driver

    ADD CONSTRAINT unique\_d\_phone UNIQUE (phone\_number);

ALTER TABLE ONLY bs.passenger

    ADD CONSTRAINT unique\_p\_email UNIQUE (e\_mail);

ALTER TABLE ONLY bs.passenger

    ADD CONSTRAINT unique\_p\_passport UNIQUE (passport);

ALTER TABLE ONLY bs.passenger

    ADD CONSTRAINT unique\_p\_phone UNIQUE (phone\_number);

ALTER TABLE ONLY bs.way\_stop

    ADD CONSTRAINT way\_stop\_pkey PRIMARY KEY (id);

CREATE TRIGGER seatnumbertrigger BEFORE INSERT ON bs.ticket FOR EACH ROW EXECUTE PROCEDURE public.update\_seat\_number();

ALTER TABLE ONLY bs.bus

    ADD CONSTRAINT bus\_type\_id\_fkey FOREIGN KEY (type\_id) REFERENCES bs.bus\_type(id);

ALTER TABLE ONLY bs.crew

    ADD CONSTRAINT crew\_driver\_id\_fkey FOREIGN KEY (driver\_id) REFERENCES bs.driver(id);

ALTER TABLE ONLY bs.flight

    ADD CONSTRAINT flight\_bus\_num\_fkey FOREIGN KEY (bus\_num) REFERENCES bs.bus(number);

ALTER TABLE ONLY bs.flight

    ADD CONSTRAINT flight\_crew\_id\_fkey FOREIGN KEY (crew\_id) REFERENCES bs.crew(id);

ALTER TABLE ONLY bs.flight

    ADD CONSTRAINT flight\_route\_id\_fkey FOREIGN KEY (route\_id) REFERENCES bs.route(id);

ALTER TABLE ONLY bs.seat

    ADD CONSTRAINT seat\_flight\_id\_fkey FOREIGN KEY (flight\_id) REFERENCES bs.flight(number);

ALTER TABLE ONLY bs.ticket

    ADD CONSTRAINT ticket\_arrival\_stop\_id\_fkey FOREIGN KEY (arrival\_stop\_id) REFERENCES bs.bus\_stop(id);

ALTER TABLE ONLY bs.ticket

    ADD CONSTRAINT ticket\_departure\_stop\_id\_fkey FOREIGN KEY (departure\_stop\_id) REFERENCES bs.bus\_stop(id);

ALTER TABLE ONLY bs.ticket

    ADD CONSTRAINT ticket\_flight\_num\_fkey FOREIGN KEY (flight\_num) REFERENCES bs.flight(number);

ALTER TABLE ONLY bs.ticket

    ADD CONSTRAINT ticket\_passenger\_id\_fkey FOREIGN KEY (passenger\_id) REFERENCES bs.passenger(id);

ALTER TABLE ONLY bs.ticket

    ADD CONSTRAINT ticket\_seat\_id\_fkey FOREIGN KEY (seat\_id) REFERENCES bs.seat(id);

ALTER TABLE ONLY bs.way\_stop

    ADD CONSTRAINT way\_stop\_global\_id\_fkey FOREIGN KEY (global\_id) REFERENCES bs.bus\_stop(id);

ALTER TABLE ONLY bs.way\_stop

    ADD CONSTRAINT way\_stop\_route\_id\_fkey FOREIGN KEY (route\_id) REFERENCES bs.route(id);

*--*

*-- PostgreSQL database dump complete*

*--*

Вывод:

В процессе выполнения лабораторной работы я овладела навыками создания таблиц баз данных с помощью PgAdmin 4 и PostgreSQL, заполнила их рабочими данными, сделала пезервные копии базы и восстановила ее с помощью pg\_restore.