Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №3 «Создание БД в СУБД PostgreSQL. Резервное копирование и восстановление БД»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Пиотуховский А.А.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы

Овладеть практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4, создания таблиц базы данных PostgreSQL 1X, назначение ограничений на данные, заполнения таблицх рабочими данными, резервного копирования и восстановления БД.

Практическое задание

1. Установить СУБД PostgreSQL, pgAdmin4.
2. Создать базу данных с использованием pgAdmin4.
3. Создать схему, таблицы в базе данных.
4. Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД.
7. Восстановить резервную копию БД.

Вариант 10. БД «Автовокзал»

Описание предметной области:

С автовокзала ежедневно отправляется несколько междугородных/международных автобусных рейсов. Номер рейса определяется маршрутом и временем отправления. По всем промежуточным остановкам на маршруте известны название, тип населенного пункта, время прибытия, отправления, время стоянки.

Автобусы курсируют по расписанию, но могут назначаться дополнительные рейсы на заданный период или определенные даты.

Билеты могут продаваться предварительно, но не ранее чем за 10 суток. В билете указывается номер места в автобусе. На каждый рейс может продаваться не более 10 билетов без места, цена на которые снижается на 10%. Пунктами отправления и назначения, согласно билету, могут быть промежуточные остановки.

Билеты могут продаваться в кассе автовокзала или онлайн.

Необходимо учитывать, что местом посадки и высадки пассажира могут быть промежуточные остановки согласно купленному билету.

На каждый рейс формируется экипаж из двух водителей.

БД должна содержать следующий минимальный набор сведений: Номер рейса. Номер водителя. Номер автобуса. Паспортные данные водителя. Пункт отправления. Пункт назначения. Промежуточные остановки. Дата отправления. Время отправления. Время в

пути. Тип автобуса. Количество мест в автобусе. Страна. Производитель. Год выпуска. Номер билета. Номер места в автобусе (при наличии). Цена билета. ФИО пассажира. Паспортные данные пассажира.

Выполнение

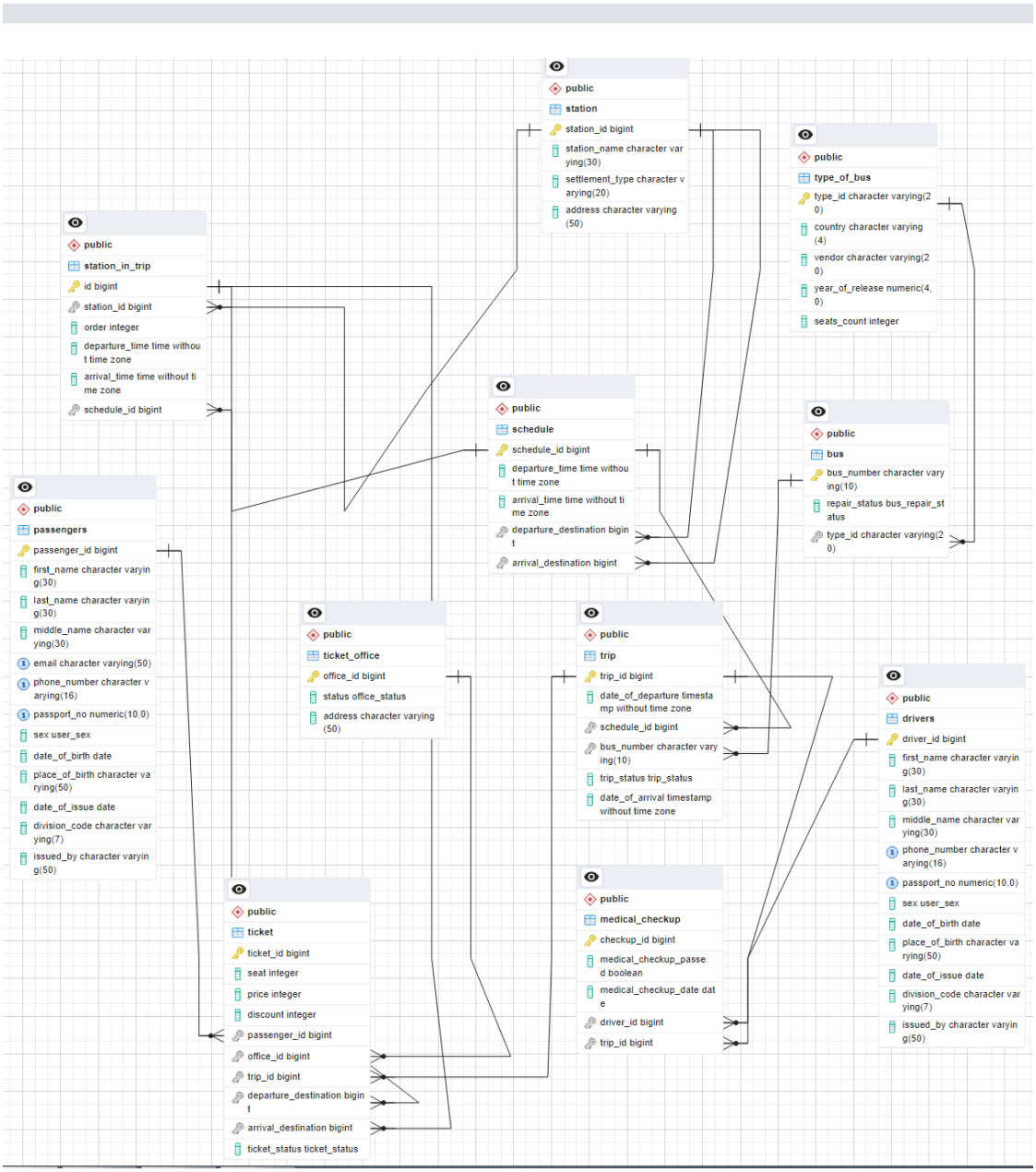

Рисунок 1 – Схема логической модели базы данных в pgAdmin4.

ERD в pgAdmin 4 строит верные, но запутанные связи. Без пересоздания всех связей собрать таблицы в аккуратном порядке не получилось. На рисунке 2 представлена эта же логическая схема, но выполненная в IDEA от JetBrains (Powered by yFiles).

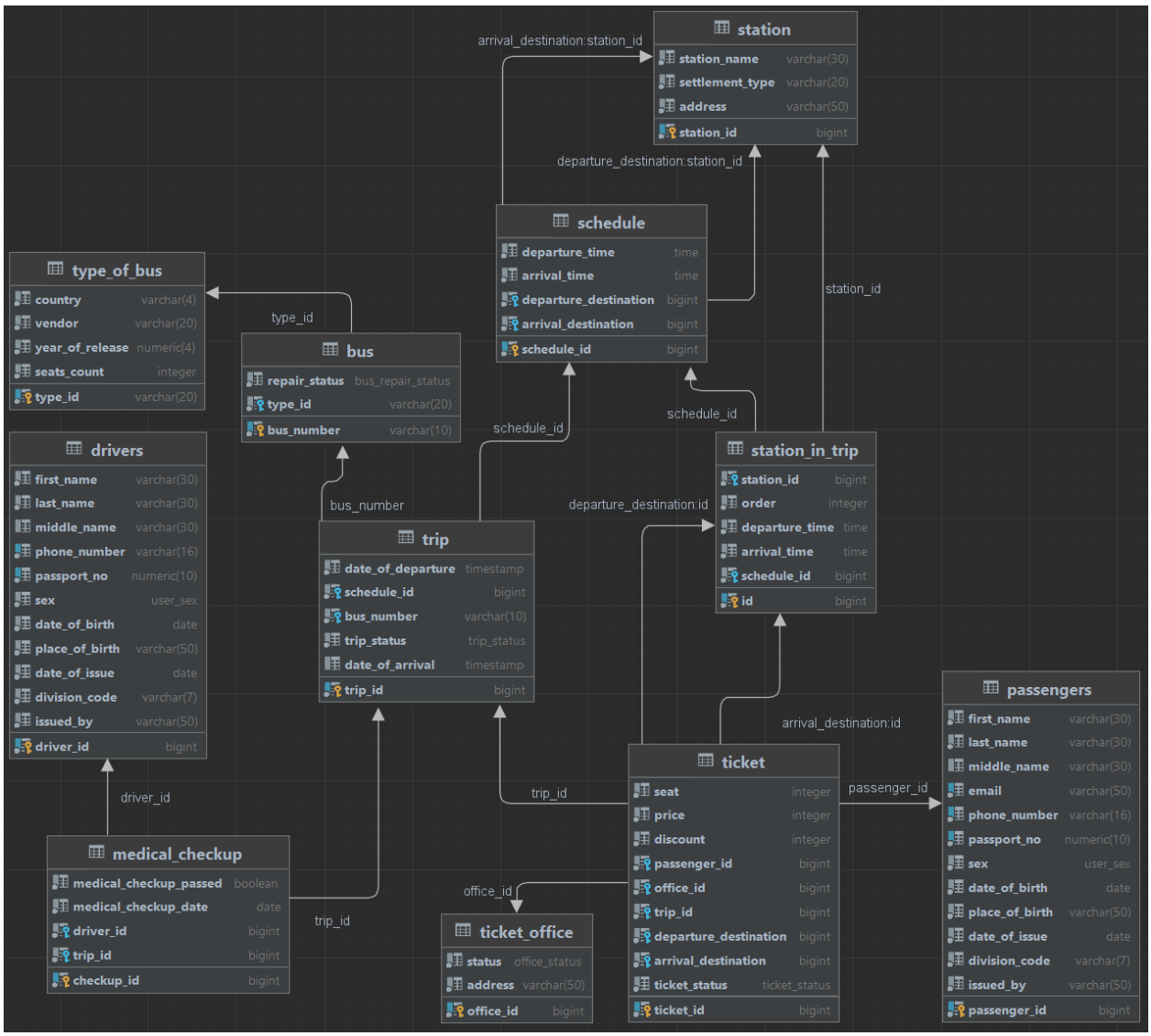

Рисунок 2 – Схема логической модели базы данных в IDEA от JetBrains.

Листинг кода дампа приведен ниже в листинге 1:

Листинг 1 – Описание атрибутов сущностей.

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 15.3
-- Dumped by pg_dump version 15.2

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: ifmo; Type: DATABASE; Schema: -; Owner: postgres
--

CREATE DATABASE ifmo WITH TEMPLATE = template0 ENCODING = 'UTF8'
LOCALE_PROVIDER = libc LOCALE = 'Russian_Russia.1251';


ALTER DATABASE ifmo OWNER TO postgres;

\connect ifmo

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: bus_repair_status; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.bus_repair_status AS ENUM (
    'good',
    'defective',
    'discarded',
    'repairing'
);


ALTER TYPE public.bus_repair_status OWNER TO postgres;

--
-- Name: office_status; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.office_status AS ENUM (
```

```
    'opened',
    'closed',
    'closed temporarily',
    'closed permanently'
);


ALTER TYPE public.office_status OWNER TO postgres;

--
-- Name: ticket_status; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.ticket_status AS ENUM (
    'pending',
    'paid',
    'canceled'
);


ALTER TYPE public.ticket_status OWNER TO postgres;

--
-- Name: trip_status; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.trip_status AS ENUM (
    'on schedule',
    'delayed',
    'canceled',
    'rescheduled'
);


ALTER TYPE public.trip_status OWNER TO postgres;

--
-- Name: user_sex; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.user_sex AS ENUM (
    'M',
    'F',
    'NB'
);


ALTER TYPE public.user_sex OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;

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

CREATE TABLE public.bus (
    bus_number character varying(10) NOT NULL,
    repair_status public.bus_repair_status NOT NULL,
    type_id character varying(20) NOT NULL,
    CONSTRAINT check_bus_number CHECK (((bus_number)::text ~ '[a-z0-
9]+'::text))
```

```
);


ALTER TABLE public.bus OWNER TO postgres;

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

CREATE TABLE public.medical_checkup (
    checkup_id bigint NOT NULL,
    medical_checkup_passed boolean NOT NULL,
    medical_checkup_date date NOT NULL,
    driver_id bigint NOT NULL,
    trip_id bigint NOT NULL
);


ALTER TABLE public.medical_checkup OWNER TO postgres;

--
-- Name: crew_crew_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

ALTER TABLE public.medical_checkup ALTER COLUMN checkup_id ADD GENERATED
ALWAYS AS IDENTITY (
    SEQUENCE NAME public.crew_crew_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


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

CREATE TABLE public.drivers (
    driver_id bigint NOT NULL,
    first_name character varying(30) NOT NULL,
    last_name character varying(30) NOT NULL,
    middle_name character varying(30),
    phone_number character varying(16) NOT NULL,
    passport_no numeric(10,0) NOT NULL,
    sex public.user_sex NOT NULL,
    date_of_birth date NOT NULL,
    place_of_birth character varying(50) NOT NULL,
    date_of_issue date NOT NULL,
    division_code character varying(7) NOT NULL,
    issued_by character varying(50) NOT NULL,
    CONSTRAINT check_date_of_birth CHECK ((date_of_birth < now())),
    CONSTRAINT check_date_of_issue CHECK (((date_of_issue > date_of_birth)
AND (date_of_issue < now()))),
    CONSTRAINT check_division_code CHECK (((division_code)::text ~ '^[0-
9]{3}\-[0-9]{3}$'::text)),
    CONSTRAINT check_first_name CHECK (((first_name)::text ~ '^[A-Яа-яа-zA-
Z]+$'::text)),
    CONSTRAINT check_issued_by CHECK (((issued_by)::text ~ '^[a-zA-Za-яA-Я
]+$'::text)),
    CONSTRAINT check_last_name CHECK (((last_name)::text ~ '^[A-Яа-яа-zA-
Z]+$'::text)),
```

```
    CONSTRAINT check_middle_name CHECK (((middle_name)::text ~ '^[A-Яа-яа-zA-
Z]*$'::text)),
    CONSTRAINT check_passport CHECK (((passport_no)::text ~ '[0-
9]{10}'::text)),
    CONSTRAINT check_phone CHECK (((phone_number)::text ~ '^\+?[0-
9]{8,15}$'::text)),
    CONSTRAINT check_place_of_birth CHECK (((place_of_birth)::text ~ '^[a-za-
яA-ZA-Я -]+$'::text))
);


ALTER TABLE public.drivers OWNER TO postgres;

--
-- Name: drivers_driver_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.drivers ALTER COLUMN driver_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME public.drivers_driver_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


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

CREATE TABLE public.passengers (
    passenger_id bigint NOT NULL,
    first_name character varying(30) NOT NULL,
    last_name character varying(30) NOT NULL,
    middle_name character varying(30),
    email character varying(50) NOT NULL,
    phone_number character varying(16) NOT NULL,
    passport_no numeric(10,0) NOT NULL,
    sex public.user_sex NOT NULL,
    date_of_birth date NOT NULL,
    place_of_birth character varying(50) NOT NULL,
    date_of_issue date NOT NULL,
    division_code character varying(7) NOT NULL,
    issued_by character varying(50) NOT NULL,
    CONSTRAINT check_date_of_birth CHECK ((date_of_birth < now())),
    CONSTRAINT check_date_of_issue CHECK (((date_of_issue > date_of_birth)
AND (date_of_issue < now()))),
    CONSTRAINT check_division_code CHECK (((division_code)::text ~ '^[0-
9]{3}\-[0-9]{3}$'::text)),
    CONSTRAINT check_email CHECK (((email)::text ~ '^[a-z0-9._%+-]+@([a-z0-9-
]+\.)*[a-z0-9-]+\.[a-z]{2,}$'::text)),
    CONSTRAINT check_first_name CHECK (((first_name)::text ~ '^[A-Яа-яа-zA-
Z]+$'::text)),
    CONSTRAINT check_issued_by CHECK (((issued_by)::text ~ '^[a-zA-Za-яA-Я
]+$'::text)),
    CONSTRAINT check_last_name CHECK (((last_name)::text ~ '^[A-Яа-яа-zA-
Z]+$'::text)),
    CONSTRAINT check_middle_name CHECK (((middle_name)::text ~ '^[A-Яа-яа-zA-
Z]*$'::text)),
    CONSTRAINT check_passport CHECK (((passport_no)::text ~ '[0-
```

```
9]{10}'::text)),
    CONSTRAINT check_phone CHECK (((phone_number)::text ~ '^\+?[0-
9]{8,15}$'::text)),
    CONSTRAINT check_place_of_birth CHECK (((place_of_birth)::text ~ '^[a-za-
яA-ZA-Я -]+$'::text))
);


ALTER TABLE public.passengers OWNER TO postgres;

--
-- Name: passengers_passenger_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.passengers ALTER COLUMN passenger_id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME public.passengers_passenger_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


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

CREATE TABLE public.schedule (
    schedule_id bigint NOT NULL,
    departure_time time without time zone NOT NULL,
    arrival_time time without time zone NOT NULL,
    departure_destination bigint NOT NULL,
    arrival_destination bigint NOT NULL,
    CONSTRAINT check_times CHECK ((arrival_time <= departure_time))
);


ALTER TABLE public.schedule OWNER TO postgres;

--
-- Name: schedule_schedule_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.schedule ALTER COLUMN schedule_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME public.schedule_schedule_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


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

CREATE TABLE public.station (
```

```
    station_id bigint NOT NULL,
    station_name character varying(30) NOT NULL,
    settlement_type character varying(20) NOT NULL,
    address character varying(50) NOT NULL
);


ALTER TABLE public.station OWNER TO postgres;

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

CREATE TABLE public.station_in_trip (
    id bigint NOT NULL,
    station_id bigint NOT NULL,
    "order" integer NOT NULL,
    departure_time time without time zone NOT NULL,
    arrival_time time without time zone NOT NULL,
    schedule_id bigint NOT NULL,
    CONSTRAINT check_times CHECK ((arrival_time <= departure_time))
);


ALTER TABLE public.station_in_trip OWNER TO postgres;

--
-- Name: station_in_trip_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.station_in_trip ALTER COLUMN id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME public.station_in_trip_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


--
-- Name: station_station_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.station ALTER COLUMN station_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME public.station_station_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


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

CREATE TABLE public.ticket (
```

```
    ticket_id bigint NOT NULL,
    seat integer NOT NULL,
    price integer NOT NULL,
    discount integer DEFAULT 0 NOT NULL,
    passenger_id bigint NOT NULL,
    office_id bigint NOT NULL,
    trip_id bigint NOT NULL,
    departure_destination bigint NOT NULL,
    arrival_destination bigint NOT NULL,
    ticket_status public.ticket_status DEFAULT
'pending'::public.ticket_status NOT NULL,
    CONSTRAINT check_discount CHECK ((discount >= 0)),
    CONSTRAINT check_price CHECK ((price > 0))
);


ALTER TABLE public.ticket OWNER TO postgres;

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

CREATE TABLE public.ticket_office (
    office_id bigint NOT NULL,
    status public.office_status NOT NULL,
    address character varying(50) NOT NULL
);


ALTER TABLE public.ticket_office OWNER TO postgres;

--
-- Name: ticket_office_office_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.ticket_office ALTER COLUMN office_id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME public.ticket_office_office_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


--
-- Name: ticket_ticket_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public.ticket ALTER COLUMN ticket_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME public.ticket_ticket_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);
```

```
--
-- Name: trip; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.trip (
    trip_id bigint NOT NULL,
    date_of_departure timestamp without time zone NOT NULL,
    schedule_id bigint NOT NULL,
    bus_number character varying(10) NOT NULL,
    trip_status public.trip_status NOT NULL,
    date_of_arrival timestamp without time zone,
    CONSTRAINT check_departure_and_arrival CHECK ((date_of_departure >=
date_of_arrival))
);


ALTER TABLE public.trip OWNER TO postgres;

--
-- Name: trip_trip_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

ALTER TABLE public.trip ALTER COLUMN trip_id ADD GENERATED ALWAYS AS IDENTITY
(
    SEQUENCE NAME public.trip_trip_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


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

CREATE TABLE public.type_of_bus (
    type_id character varying(20) NOT NULL,
    country character varying(4) NOT NULL,
    vendor character varying(20) NOT NULL,
    year_of_release numeric(4,0) NOT NULL,
    seats_count integer NOT NULL,
    CONSTRAINT check_country CHECK (((country)::text ~ '[a-z]{2,4}'::text)),
    CONSTRAINT check_seats_count CHECK ((seats_count >= 0)),
    CONSTRAINT check_year_of_release CHECK (((year_of_release > (0)::numeric)
AND ((year_of_release)::double precision <= date_part('year'::text, now())))))
);


ALTER TABLE public.type_of_bus OWNER TO postgres;

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

COPY public.bus (bus_number, repair_status, type_id) FROM stdin;
e947at147    good    Лазурный
o00000777    repairing    Лазурный
t665at12    good    Лазурный
m193em47    defective    Буханка
p141hh78    good    Буханка
t765dt98    discarded    Уазик
```

```
\.


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

COPY public.drivers (driver_id, first_name, last_name, middle_name,
phone_number, passport_no, sex, date_of_birth, place_of_birth, date_of_issue,
division_code, issued_by) FROM stdin;
1   Алексей    Крутоголов Михайлович +77777777777   4878888888 М  1977-05-12
Москва 2019-07-30 780-001    ГУ МВД ПО СПБ И ЛО
3   Михаил Садко  Романович  +7908654322    7657435262 М  1978-01-02 Санкт-
Петербург    2019-07-29 780-001    ГУ МВД ПО СПБ И ЛО
4   Ярослав    Тороп  Дмитриевич +79213676543   4810876578 М  1985-12-01
Санкт-Петербург    2020-01-19 876-018    ГУ МВД ПО СПБ И ЛО
\.


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

COPY public.medical_checkup (checkup_id, medical_checkup_passed,
medical_checkup_date, driver_id, trip_id) FROM stdin;
2   t  2023-10-22 3  1
3   f  2023-10-22 1  1
4   t  2023-10-22 4  1
\.


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

COPY public.passengers (passenger_id, first_name, last_name, middle_name,
email, phone_number, passport_no, sex, date_of_birth, place_of_birth,
date_of_issue, division_code, issued_by) FROM stdin;
9   Renaldo    Corley Никифорович    era_rowell86@yahoo.com +76475629190
7465192836 F  2004-03-21 Orange 2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
23  Justa  Brand  Андреевич  brittny43606@forestry.com  +76456292193
7847562847 F  2004-03-21 Baltimore  2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
10  Cristopher Meacham    Петров dick.leggett@banks.com +76475629194
1234556753 F  2004-03-21 Odessa 2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
8   Alayna Spears Ященов sharie_chilton04@arrest.com    +76475629191
9876543266 F  2004-03-21 Killeen    2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
21  Emanuel    Barbosa    Еленовна   leia9@scout.com    +7645629193
2047562847 F  2004-03-21 Norwalk    2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
19  Cleo   Hager  Денисовна  elainagleason@hotmail.com  +76475229193
2847562047 F  2004-03-21 Abilene    2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
11  Lorilee    Holguin    Денисович  stephenmattson@hotmail.com +76475629195
6354658672 NB 2004-03-21 Mesa   2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
22  Connie Pulliam    \N mckinleybuchanan@hotmail.com   +6475629193
9847562847 M  2004-03-21 Poughkeepsie   2018-02-02 786-009    ГУ МВД ПО СПБ И
ЛО
24  Teodora    Castleberry    Владимирович   stanford203@yahoo.com
+76475623193   6847562847 F  2004-03-21 Nashua 2018-02-02 786-009    ГУ МВД
ПО СПБ И ЛО
15  Maris  Mount  Петровна   verlene04403@hotmail.com   +76475629199
2347562847 NB 2003-03-21 Peoria 2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
17  Weldon Kirchner   \N dominiccordero@idle.com    +7647562918    2847562849
```

```
M  2004-03-21 Layton 2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
7   Jeannie    Sena   Кириллович francisca.sturgeon@yahoo.com   +76475629192
1234567890 M  2004-03-21 Pomona 2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
12  Zena   Royster    Смирнов    domonique-upshaw582@yahoo.com  +76475629196
2847562945 M  2004-03-21 Huntington Beach   2018-02-02 786-009    ГУ МВД ПО
СПБ И ЛО
14  Louis  Kroll  Иванов allegraroybal@deck.com +76475629198   2847522847 M
2005-03-21 Fayetteville   2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
27  Isreal Valerio    Олегович   ilene_lawson7@violent.definima.net
+7647562310    9999999999 M  2004-03-21 Leominster 2018-02-02 786-009    ГУ
МВД ПО СПБ И ЛО
6   Bethanie   Macmillan  Казахстанович  krystinaaugustine41@gmail.com
+76475629193   3333333333 M  2004-03-21 Norfolk    2018-02-02 786-009    ГУ
МВД ПО СПБ И ЛО
18  Cassey Hostetler  Александрович  salinawilber@departure.com +76475629
2847562840 M  2004-03-21 Westminster    2018-02-02 786-009    ГУ МВД ПО СПБ И
ЛО
13  Gregory    Cupp   Георгевич  marcy57@hotmail.com    +76475629197
2847562847 F  2004-03-21 Amarillo   2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
20  Susan  Darby  Михайлович jeanna.zimmerman@gmail.com +72475629193
2147562847 NB 2004-03-21 Orange 2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
16  Leana  Oliver Денисович  hanbernier@yahoo.com   +7647562919    2847562848
NB 2004-03-21 Lake Charles   2018-02-02 786-009    ГУ МВД ПО СПБ И ЛО
\.


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

COPY public.schedule (schedule_id, departure_time, arrival_time,
departure_destination, arrival_destination) FROM stdin;
3   14:20:00   14:13:00   2  3
2   16:30:00   10:30:00   1  4
1   12:00:00   10:00:00   1  3
\.


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

COPY public.station (station_id, station_name, settlement_type, address) FROM
stdin;
1   Ветеранов  Город  Ветеранов 13
2   Восстания  Город  Восстания 1
3   Владимирская   Город  Большая Московская 2
4   Девяткино  Деревня    Авиаторов Балтики 7
\.


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

COPY public.station_in_trip (id, station_id, "order", departure_time,
arrival_time, schedule_id) FROM stdin;
1   1  0  10:00:00   09:55:00   1
2   2  1  11:50:00   11:45:00   1
3   3  2  12:00:00   12:00:00   1
\.
```

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

COPY public.ticket (ticket_id, seat, price, discount, passenger_id,
office_id, trip_id, departure_destination, arrival_destination,
ticket_status) FROM stdin;
4   -1 10 0  10 2  1  3  2  pending
1   1  13 0  7  0  1  1  2  paid
2   -1 8  0  8  0  1  1  3  canceled
3   2  13 0  9  0  1  1  2  paid
\.


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

COPY public.ticket_office (office_id, status, address) FROM stdin;
2   opened Большая Морская 12
3   closed Ветеранов 11
4   closed permanently Авиаторов Балтики 7
0   opened ONLINE
\.


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

COPY public.trip (trip_id, date_of_departure, schedule_id, bus_number,
trip_status, date_of_arrival) FROM stdin;
1   2023-10-22 03:00:00    1  e947at147  delayed    2023-10-22 01:00:00
2   2023-10-22 01:00:00    2  e947at147  canceled   2023-10-22 00:00:00
3   2023-10-23 02:00:00    3  t665at12   on schedule    2023-10-22 02:00:00
\.


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

COPY public.type_of_bus (type_id, country, vendor, year_of_release,
seats_count) FROM stdin;
Буханка ru АвтоВАЗ    1995   20
Уазик   ru АвтоВАЗ    1997   2
Лазурный    ch KIA    2022   50
\.


--
-- Name: crew_crew_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
--

SELECT pg_catalog.setval('public.crew_crew_id_seq', 4, true);


--
-- Name: drivers_driver_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
```

```
postgres
--

SELECT pg_catalog.setval('public.drivers_driver_id_seq', 4, true);


--
-- Name: passengers_passenger_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public.passengers_passenger_id_seq', 27, true);


--
-- Name: schedule_schedule_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
--

SELECT pg_catalog.setval('public.schedule_schedule_id_seq', 3, true);


--
-- Name: station_in_trip_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
--

SELECT pg_catalog.setval('public.station_in_trip_id_seq', 3, true);


--
-- Name: station_station_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
--

SELECT pg_catalog.setval('public.station_station_id_seq', 4, true);


--
-- Name: ticket_office_office_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public.ticket_office_office_id_seq', 4, true);


--
-- Name: ticket_ticket_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
--

SELECT pg_catalog.setval('public.ticket_ticket_id_seq', 4, true);


--
-- Name: trip_trip_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
--

SELECT pg_catalog.setval('public.trip_trip_id_seq', 3, true);


--
```

```
-- Name: bus bus_pk; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.bus
    ADD CONSTRAINT bus_pk PRIMARY KEY (bus_number);


--
-- Name: drivers drivers_pk; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.drivers
    ADD CONSTRAINT drivers_pk PRIMARY KEY (driver_id);


--
-- Name: drivers drivers_pk2; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.drivers
    ADD CONSTRAINT drivers_pk2 UNIQUE (phone_number);


--
-- Name: drivers drivers_pk3; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.drivers
    ADD CONSTRAINT drivers_pk3 UNIQUE (passport_no);


--
-- Name: medical_checkup medical_checkup_pk; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public.medical_checkup
    ADD CONSTRAINT medical_checkup_pk PRIMARY KEY (checkup_id);


--
-- Name: passengers passengers_pk; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.passengers
    ADD CONSTRAINT passengers_pk PRIMARY KEY (passenger_id);


--
-- Name: passengers passengers_pk2; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.passengers
    ADD CONSTRAINT passengers_pk2 UNIQUE (email);


--
-- Name: passengers passengers_pk3; Type: CONSTRAINT; Schema: public; Owner:
```

```
postgres
--

ALTER TABLE ONLY public.passengers
    ADD CONSTRAINT passengers_pk3 UNIQUE (phone_number);


--
-- Name: passengers passengers_pk4; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.passengers
    ADD CONSTRAINT passengers_pk4 UNIQUE (passport_no);


--
-- Name: schedule schedule_pk; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.schedule
    ADD CONSTRAINT schedule_pk PRIMARY KEY (schedule_id);


--
-- Name: station_in_trip station_in_trip_pk; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public.station_in_trip
    ADD CONSTRAINT station_in_trip_pk PRIMARY KEY (id);


--
-- Name: station station_pk; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.station
    ADD CONSTRAINT station_pk PRIMARY KEY (station_id);


--
-- Name: ticket_office ticket_office_pk; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.ticket_office
    ADD CONSTRAINT ticket_office_pk PRIMARY KEY (office_id);


--
-- Name: ticket ticket_pk; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ticket
    ADD CONSTRAINT ticket_pk PRIMARY KEY (ticket_id);


--
-- Name: trip trip_pk; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.trip
    ADD CONSTRAINT trip_pk PRIMARY KEY (trip_id);


--
-- Name: type_of_bus type_of_bus_pk; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.type_of_bus
    ADD CONSTRAINT type_of_bus_pk PRIMARY KEY (type_id);


--
-- Name: bus bus_type_of_bus_type_id_fk; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.bus
    ADD CONSTRAINT bus_type_of_bus_type_id_fk FOREIGN KEY (type_id)
REFERENCES public.type_of_bus(type_id);


--
-- Name: medical_checkup crew_drivers_driver_id_fk; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.medical_checkup
    ADD CONSTRAINT crew_drivers_driver_id_fk FOREIGN KEY (driver_id)
REFERENCES public.drivers(driver_id);


--
-- Name: medical_checkup medical_checkup_trip_trip_id_fk; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.medical_checkup
    ADD CONSTRAINT medical_checkup_trip_trip_id_fk FOREIGN KEY (trip_id)
REFERENCES public.trip(trip_id);


--
-- Name: schedule schedule_station_station_id_fk; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.schedule
    ADD CONSTRAINT schedule_station_station_id_fk FOREIGN KEY
(departure_destination) REFERENCES public.station(station_id);


--
-- Name: schedule schedule_station_station_id_fk2; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.schedule
    ADD CONSTRAINT schedule_station_station_id_fk2 FOREIGN KEY
(arrival_destination) REFERENCES public.station(station_id);
```

```
--
-- Name: station_in_trip station_in_trip_schedule_schedule_id_fk; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.station_in_trip
    ADD CONSTRAINT station_in_trip_schedule_schedule_id_fk FOREIGN KEY
(schedule_id) REFERENCES public.schedule(schedule_id);


--
-- Name: station_in_trip station_in_trip_station_station_id_fk; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.station_in_trip
    ADD CONSTRAINT station_in_trip_station_station_id_fk FOREIGN KEY
(station_id) REFERENCES public.station(station_id);


--
-- Name: ticket ticket_passengers_passenger_id_fk; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ticket
    ADD CONSTRAINT ticket_passengers_passenger_id_fk FOREIGN KEY
(passenger_id) REFERENCES public.passengers(passenger_id);


--
-- Name: ticket ticket_station_in_trip_id_fk; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public.ticket
    ADD CONSTRAINT ticket_station_in_trip_id_fk FOREIGN KEY
(departure_destination) REFERENCES public.station_in_trip(id);


--
-- Name: ticket ticket_station_in_trip_id_fk2; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public.ticket
    ADD CONSTRAINT ticket_station_in_trip_id_fk2 FOREIGN KEY
(arrival_destination) REFERENCES public.station_in_trip(id);


--
-- Name: ticket ticket_ticket_office_office_id_fk; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ticket
    ADD CONSTRAINT ticket_ticket_office_office_id_fk FOREIGN KEY (office_id)
REFERENCES public.ticket_office(office_id);


--
-- Name: ticket ticket_trip_trip_id_fk; Type: FK CONSTRAINT; Schema: public;
```

```
Owner: postgres
--

ALTER TABLE ONLY public.ticket
    ADD CONSTRAINT ticket_trip_trip_id_fk FOREIGN KEY (trip_id) REFERENCES
public.trip(trip_id);


--
-- Name: trip trip_bus_bus_number_fk; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.trip
    ADD CONSTRAINT trip_bus_bus_number_fk FOREIGN KEY (bus_number) REFERENCES
public.bus(bus_number);


--
-- Name: trip trip_schedule_schedule_id_fk; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public.trip
    ADD CONSTRAINT trip_schedule_schedule_id_fk FOREIGN KEY (schedule_id)
REFERENCES public.schedule(schedule_id);


--
-- PostgreSQL database dump complete
--
```

Вывод

В ходе лабораторной работы я научился создавать, заполнять и восстанавливать базы данных PostgreSQL с использованием программы pgAdmin4. В процессе лабораторной работы была создана база данных с таблицами в соответствии с заданием. Были заданы необходимые привязки и ограничения, после чего таблицы были заполнены данными. Также было создано две резервные копии, которые позволят восстановить базу данных без потерь, а также посмотреть листинг кода.