

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»  
Факультет инфокоммуникационных технологий

**ОТЧЕТ**  
**О ЛАБОРАТОРНОЙ РАБОТЕ № 3**

по теме:

«СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ POSTGRESQL.  
ЗАПОЛНЕНИЕ ТАБЛИЦ РАБОЧИМИ ДАННЫМИ»

по дисциплине: Проектирование и реализация баз данных

Специальность:  
09.03.03 Прикладная информатика

Проверила:  
Говорова М.М. \_\_\_\_\_  
Дата: «\_\_» \_\_\_\_\_ 20\_\_ г.  
Оценка \_\_\_\_\_

Выполнил:  
студент группы: K3241  
Иконенко Д.А.

Санкт-Петербург 2020/2021

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

**Практическое задание:**

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. заданию).
3. Создать схему в составе базы данных.
4. Создать таблицы базы данных.
5. Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.
6. Заполнить таблицы БД рабочими данными.
7. 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.
8. Восстановить БД.

## Выполнение работы

### База данных «Железная дорога» (вариант 6 в лабораторной работе 2)

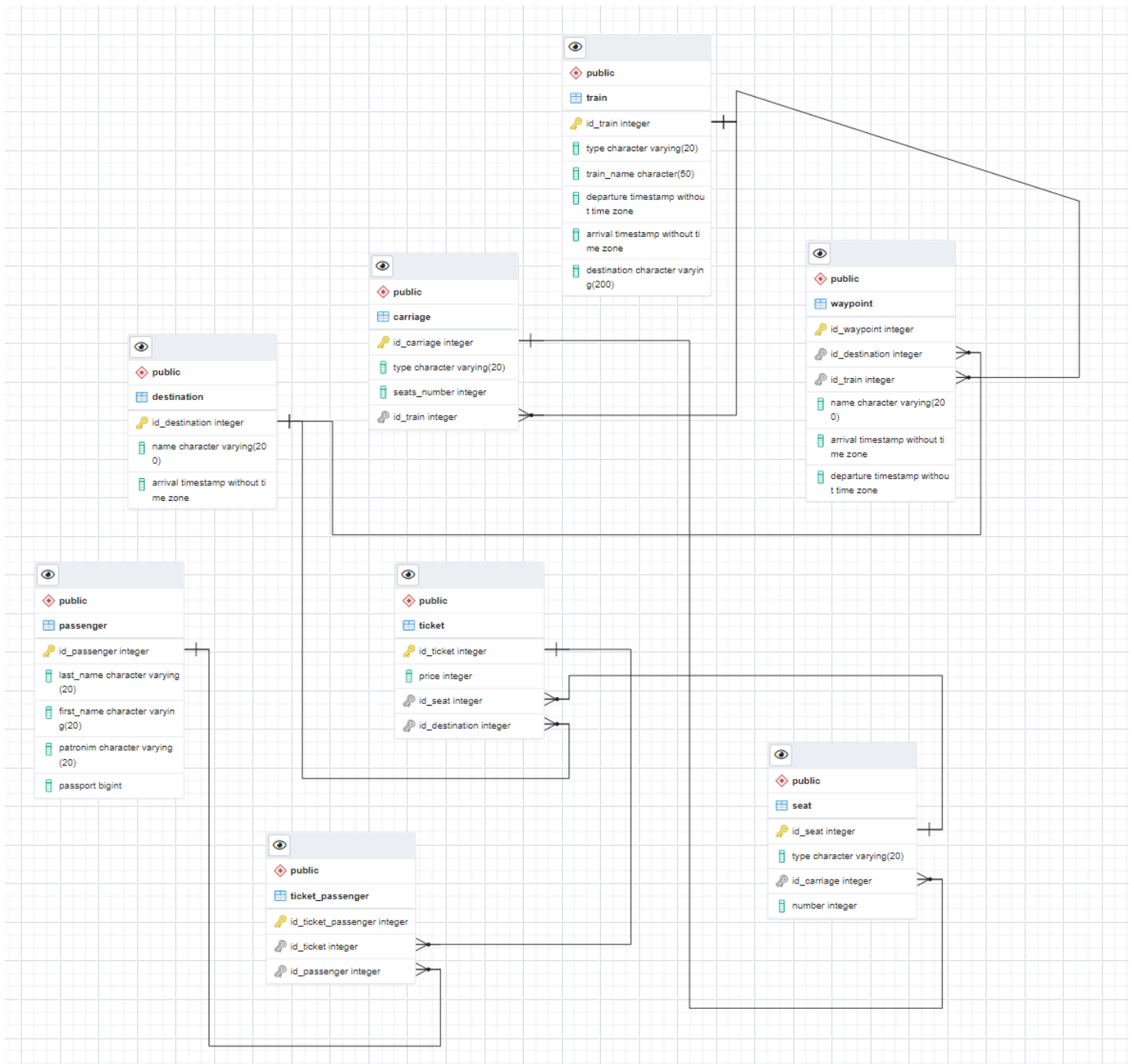


Рисунок 1 – Схема логической модели базы данных, сгенерированная в Generate ERD

## Dump базы данных

--

-- PostgreSQL database dump

--

-- Dumped from database version 13.2

-- Dumped by pg\_dump version 13.2

-- Started on 2021-05-20 16:22:13

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;

--

-- TOC entry 3 (class 2615 OID 2200)

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

--

CREATE SCHEMA public;

ALTER SCHEMA public OWNER TO postgres;

```
--  
-- TOC entry 3056 (class 0 OID 0)  
-- Dependencies: 3  
-- Name: SCHEMA public; Type: COMMENT; Schema: -; Owner: postgres  
--
```

```
COMMENT ON SCHEMA public IS 'standard public schema';
```

```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

```
--  
-- TOC entry 202 (class 1259 OID 16428)  
-- Name: carriage; Type: TABLE; Schema: public; Owner: postgres  
--
```

### Таблица «Вагон»

```
CREATE TABLE public.carriage (  
    id_carriage integer NOT NULL,  
    type character varying(20) NOT NULL,  
    seats_number integer NOT NULL,  
    id_train integer NOT NULL,  
    CONSTRAINT carriage_id_carriage_check CHECK ((id_carriage >= 0)),  
    CONSTRAINT carriage_seats_number_check CHECK ((seats_number >= 0)),  
    CONSTRAINT carriage_type_check CHECK (((type)::text = 'сидячий'::text) OR  
((type)::text = 'плацкарт'::text) OR ((type)::text = 'купе'::text) OR ((type)::text =  
'СВ'::text) OR ((type)::text = 'люкс'::text)))  
);
```

```
ALTER TABLE public.carriage OWNER TO postgres;
```

```
--
```

```
-- TOC entry 204 (class 1259 OID 16448)
```

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

```
--
```

#### Таблица «Пункт назначения»

```
CREATE TABLE public.destination (  
    id_destination integer NOT NULL,  
    name character varying(200) NOT NULL,  
    arrival timestamp without time zone NOT NULL,  
    CONSTRAINT destination_id_destination_check CHECK ((id_destination >= 0))  
);
```

```
ALTER TABLE public.destination OWNER TO postgres;
```

```
--
```

```
-- TOC entry 200 (class 1259 OID 16420)
```

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

```
--
```

#### Таблица «Пассажир»

```
CREATE TABLE public.passenger (  
    id_passenger integer NOT NULL,  
    last_name character varying(20) NOT NULL,  
    first_name character varying(20) NOT NULL,  
    patronim character varying(20),  
    passport bigint NOT NULL,  
    CONSTRAINT passenger_passport_check CHECK (((passport >= 1000000000)  
AND (passport <= '9999999999'::bigint)))
```

```
);
```

```
ALTER TABLE public.passenger OWNER TO postgres;
```

```
--
```

```
-- TOC entry 203 (class 1259 OID 16438)
```

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

```
--
```

### Таблица «Место»

```
CREATE TABLE public.seat (
```

```
    id_seat integer NOT NULL,
```

```
    type character varying(20) NOT NULL,
```

```
    id_carriage integer NOT NULL,
```

```
    number integer NOT NULL,
```

```
    CONSTRAINT seat_id_seat_check CHECK ((id_seat >= 0)),
```

```
    CONSTRAINT seat_number_check CHECK ((number >= 0)),
```

```
    CONSTRAINT seat_type_check CHECK (((type)::text = 'сидячее'::text) OR  
((type)::text = 'верхнее'::text) OR ((type)::text = 'нижнее'::text) OR ((type)::text =  
'боковое'::text)))
```

```
);
```

```
ALTER TABLE public.seat OWNER TO postgres;
```

```
--
```

```
-- TOC entry 205 (class 1259 OID 16453)
```

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

```
--
```

### Таблица «Билет»

```
CREATE TABLE public.ticket (  
    id_ticket integer NOT NULL,  
    price integer NOT NULL,  
    id_seat integer NOT NULL,  
    id_destination integer NOT NULL,  
    CONSTRAINT ticket_id_ticket_check CHECK ((id_ticket >= 0)),  
    CONSTRAINT ticket_price_check CHECK ((price >= 0))  
);
```

```
ALTER TABLE public.ticket OWNER TO postgres;
```

```
--  
-- TOC entry 206 (class 1259 OID 16488)  
-- Name: ticket_passenger; Type: TABLE; Schema: public; Owner: postgres  
--
```

#### Таблица «Продажа билета»

```
CREATE TABLE public.ticket_passenger (  
    id_ticket_passenger integer NOT NULL,  
    id_ticket integer NOT NULL,  
    id_passenger integer NOT NULL  
);
```

```
ALTER TABLE public.ticket_passenger OWNER TO postgres;
```

```
--  
-- TOC entry 201 (class 1259 OID 16423)  
-- Name: train; Type: TABLE; Schema: public; Owner: postgres  
--
```



### Таблица «Поезд»

```
CREATE TABLE public.train (  
    id_train integer NOT NULL,  
    type character varying(20) NOT NULL,  
    train_name character(50) NOT NULL,  
    departure timestamp without time zone NOT NULL,  
    arrival timestamp without time zone NOT NULL,  
    destination character varying(200) NOT NULL,  
    CONSTRAINT train_id_train_check CHECK ((id_train >= 0)),  
    CONSTRAINT train_type_check CHECK (((type)::text = 'скоростной'::text) OR  
((type)::text = 'фирменный'::text) OR ((type)::text = 'пассажирский'::text) OR  
((type)::text = 'скорый'::text)))  
);
```

```
ALTER TABLE public.train OWNER TO postgres;
```

```
--  
-- TOC entry 207 (class 1259 OID 16510)  
-- Name: waypoint; Type: TABLE; Schema: public; Owner: postgres  
--
```

### Таблица «Точка в пути»

```
CREATE TABLE public.waypoint (  
    id_waypoint integer NOT NULL,  
    id_destination integer NOT NULL,  
    id_train integer NOT NULL,  
    name character varying(200) NOT NULL,  
    arrival timestamp without time zone NOT NULL,  
    departure timestamp without time zone NOT NULL,  
    CONSTRAINT waypoint_id_waypoint_check CHECK ((id_waypoint >= 0))  
);
```

```
ALTER TABLE public.waypoint OWNER TO postgres;
```

```
--
```

```
-- TOC entry 3045 (class 0 OID 16428)
```

```
-- Dependencies: 202
```

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

```
--
```

#### Внесение данных: вагон

```
COPY public.carriage (id_carriage, type, seats_number, id_train) FROM stdin;
```

```
1    плацкарт    60    2
```

```
2    сидячий     75    1
```

```
3    купе    40    3
```

```
\.
```

```
--
```

```
-- TOC entry 3047 (class 0 OID 16448)
```

```
-- Dependencies: 204
```

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

```
--
```

#### Внесение данных: пункт назначения

```
COPY public.destination (id_destination, name, arrival) FROM stdin;
```

```
1    Москва      2021-05-20 19:00:00м
```

```
2    Нижний Новгород 2021-05-22 09:52:00
```

```
3    Саратов 2021-05-22 06:29:00
```

```
\.
```

```
--  
-- TOC entry 3043 (class 0 OID 16420)  
-- Dependencies: 200  
-- Data for Name: passenger; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

#### Внесение данных: пассажир

```
COPY public.passenger (id_passenger, last_name, first_name, patronim, passport)  
FROM stdin;
```

```
1    Мельникова Ксения Витальевна      3235418149  
2    Иванова София Ивановна           8125151917  
3    Буракшаева Юлия Сергеевна         5528819336
```

```
\\.
```

```
--  
-- TOC entry 3046 (class 0 OID 16438)  
-- Dependencies: 203  
-- Data for Name: seat; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

#### Внесение данных: место

```
COPY public.seat (id_seat, type, id_carriage, number) FROM stdin;
```

```
1    боковое      1    36  
2    сидячее      2    11  
3    нижнее       3    22
```

```
\\.
```

```
--  
-- TOC entry 3048 (class 0 OID 16453)  
-- Dependencies: 205
```

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

--

### Внесение данных: билет

COPY public.ticket (id\_ticket, price, id\_seat, id\_destination) FROM stdin;

1 1800 1 2

2 3100 2 1

3 2300 3 3

\\.

--

-- TOC entry 3049 (class 0 OID 16488)

-- Dependencies: 206

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

--

### Внесение данных: продажа билета

COPY public.ticket\_passenger (id\_ticket\_passenger, id\_ticket, id\_passenger) FROM stdin;

1 1 3

2 2 1

3 3 2

\\.

--

-- TOC entry 3044 (class 0 OID 16423)

-- Dependencies: 201

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

--

### Внесение данных: поезд

COPY public.train (id\_train, type, train\_name, departure, arrival, destination) FROM stdin;

1 скоростной Сапсан 2021-05-20 15:00:00 2021-05-20 2021-05-20 19:00:00 Москва

2 фирменный Волга 2021-05-21 18:00:00 2021-05-21 09:52:00 Нижний Новгород

3 пассажирский СПб-Саратов 2021-05-21 15:22:00 2021-05-21 06:29:00 Саратов

\\.

--

-- TOC entry 3050 (class 0 OID 16510)

-- Dependencies: 207

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

--

### Внесение данных: точка в пути

COPY public.waypoint (id\_waypoint, id\_destination, id\_train, name, arrival, departure) FROM stdin;

1 1 1 Тверь 2021-05-20 2021-05-20 16:45:00 2021-05-20 16:49:00

2 2 2 Малая Вишера 2021-05-21 22:35:00 2021-05-21 22:47:00

3 3 3 Кирсанов 2021-05-22 01:51:00 2021-05-22 01:53:00

\\.

--

-- TOC entry 2894 (class 2606 OID 16432)

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

--

### Добавление ограничений

#### Вагон: первичный ключ

```
ALTER TABLE ONLY public.carriage
```

```
    ADD CONSTRAINT "Carriage_pkey" PRIMARY KEY (id_carriage);
```

```
--
```

```
-- TOC entry 2898 (class 2606 OID 16452)
```

```
-- Name: destination Destination_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

#### Пункт назначения: первичный ключ

```
ALTER TABLE ONLY public.destination
```

```
    ADD CONSTRAINT "Destination_pkey" PRIMARY KEY (id_destination);
```

```
--
```

```
-- TOC entry 2890 (class 2606 OID 16504)
```

```
-- Name: passenger Passenger_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

#### Пассажир: первичный ключ

```
ALTER TABLE ONLY public.passenger
```

```
    ADD CONSTRAINT "Passenger_pkey" PRIMARY KEY (id_passenger);
```

```
--
```

```
-- TOC entry 2896 (class 2606 OID 16442)
```

```
-- Name: seat Seat_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

#### Место: первичный ключ

```
ALTER TABLE ONLY public.seat
```

```
    ADD CONSTRAINT "Seat_pkey" PRIMARY KEY (id_seat);
```

```
--
```

```
-- TOC entry 2902 (class 2606 OID 16492)
```

```
-- Name: ticket_passenger Ticket_Passenger_pkey; Type: CONSTRAINT; Schema:  
public; Owner: postgres
```

```
--
```

#### Билет: первичный ключ

```
ALTER TABLE ONLY public.ticket_passenger
```

```
    ADD CONSTRAINT "Ticket_Passenger_pkey" PRIMARY KEY (id_ticket_passenger);
```

```
--
```

```
-- TOC entry 2900 (class 2606 OID 16457)
```

```
-- Name: ticket Ticket_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

#### Продажа билета: первичный ключ

```
ALTER TABLE ONLY public.ticket
```

```
    ADD CONSTRAINT "Ticket_pkey" PRIMARY KEY (id_ticket);
```

```
--
```

```
-- TOC entry 2892 (class 2606 OID 16427)
```

```
-- Name: train Train_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

### Поезд: первичный ключ

```
ALTER TABLE ONLY public.train
```

```
    ADD CONSTRAINT "Train_pkey" PRIMARY KEY (id_train);
```

```
--
```

```
-- TOC entry 2904 (class 2606 OID 16514)
```

```
-- Name: waypoint Waypoint_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

### Точка в пути: первичный ключ

```
ALTER TABLE ONLY public.waypoint
```

```
    ADD CONSTRAINT "Waypoint_pkey" PRIMARY KEY (id_waypoint);
```

```
--
```

```
-- TOC entry 2905 (class 2606 OID 16433)
```

```
-- Name: carriage Carriage_id_train_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

### Вагон: внешний ключ

```
ALTER TABLE ONLY public.carriage
```

```
    ADD CONSTRAINT "Carriage_id_train_fkey" FOREIGN KEY (id_train) REFERENCES public.train(id_train);
```

```
--
```

```
-- TOC entry 2906 (class 2606 OID 16443)
```

```
-- Name: seat Seat_id_carriage_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```



#### Место: внешний ключ

```
ALTER TABLE ONLY public.seat
```

```
    ADD CONSTRAINT "Seat_id_carriage_fkey" FOREIGN KEY (id_carriage)
REFERENCES public.carriage(id_carriage);
```

```
--
```

```
-- TOC entry 2910 (class 2606 OID 16505)
```

```
-- Name: ticket_passenger Ticket_Passenger_id_passenger_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

#### Продажа билета: внешний ключ пассажир

```
ALTER TABLE ONLY public.ticket_passenger
```

```
    ADD CONSTRAINT "Ticket_Passenger_id_passenger_fkey" FOREIGN KEY
(id_passenger) REFERENCES public.passenger(id_passenger) NOT VALID;
```

```
--
```

```
-- TOC entry 2909 (class 2606 OID 16493)
```

```
-- Name: ticket_passenger Ticket_Passenger_id_ticket_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
```

```
--
```

#### Продажа билета: внешний ключ билет

```
ALTER TABLE ONLY public.ticket_passenger
```

```
    ADD CONSTRAINT "Ticket_Passenger_id_ticket_fkey" FOREIGN KEY (id_ticket)
REFERENCES public.ticket(id_ticket);
```

```
--
```

```
-- TOC entry 2908 (class 2606 OID 16463)
```

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

```
--
```

#### Билет: внешний ключ пункт назначения

```
ALTER TABLE ONLY public.ticket
```

```
    ADD CONSTRAINT "Ticket_id_destination_fkey" FOREIGN KEY (id_destination)  
REFERENCES public.destination(id_destination);
```

```
--
```

```
-- TOC entry 2907 (class 2606 OID 16458)
```

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

```
--
```

#### Билет: внешний ключ место

```
ALTER TABLE ONLY public.ticket
```

```
    ADD CONSTRAINT "Ticket_id_seat_fkey" FOREIGN KEY (id_seat) REFERENCES  
public.seat(id_seat);
```

```
--
```

```
-- TOC entry 2911 (class 2606 OID 16515)
```

```
-- Name: waypoint Waypoint_id_destination_fkey; Type: FK CONSTRAINT; Schema:  
public; Owner: postgres
```

```
--
```

#### Точка в пути: внешний ключ пункт назначения

```
ALTER TABLE ONLY public.waypoint
```

```
    ADD CONSTRAINT "Waypoint_id_destination_fkey" FOREIGN KEY (id_destination)  
REFERENCES public.destination(id_destination);
```

```
--  
-- TOC entry 2912 (class 2606 OID 16520)  
-- Name: waypoint Waypoint_id_train_fkey; Type: FK CONSTRAINT; Schema: public;  
Owner: postgres  
--
```

#### Точка в пути: внешний ключ поезд

```
ALTER TABLE ONLY public.waypoint  
    ADD CONSTRAINT "Waypoint_id_train_fkey" FOREIGN KEY (id_train)  
    REFERENCES public.train(id_train);
```

```
-- Completed on 2021-05-20 16:22:13
```

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

## Выводы

В ходе выполнения работы освоено программное средство pgadmin 4 и команды языка SQL для создания таблиц, задания ограничений на атрибуты и добавления данных. Создана база данных по заданию, в каждую таблицу добавлено несколько строк, проведены запросы для проверки работы ограничений.