**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №3.2 «Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими данными»

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

Автор: Тюленев А.С.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

[Цель работы 3](#_Toc148889485)

[Практическое задание 3](#_Toc148889487)

[Вариант 6. БД «Пассажир 3](#_Toc148889495)

[Рисунок 1 – Схема логической модели базы данных. 4](#_Toc148889496)

[Листинг дампа 5](#_Toc148889497)

[Вывод 13](#_Toc148889498)

# Цель работы

# Овладеть практическими навыками создания таблиц базы данных PostgreSQL 1Х, заполнения их рабочими данными, резервного копирования и восстановления БД.

# Практическое задание

# Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).

# Создать схему в составе базы данных.

# Создать таблицы базы данных.

# Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.

# Заполнить таблицы БД рабочими данными.

# Создать резервную копию БД.

# Указание:

# Создать две резервные копии:

# - с расширением CUSTOM для восстановления БД;

# - с расширением PLAIN для листинга (в отчете);

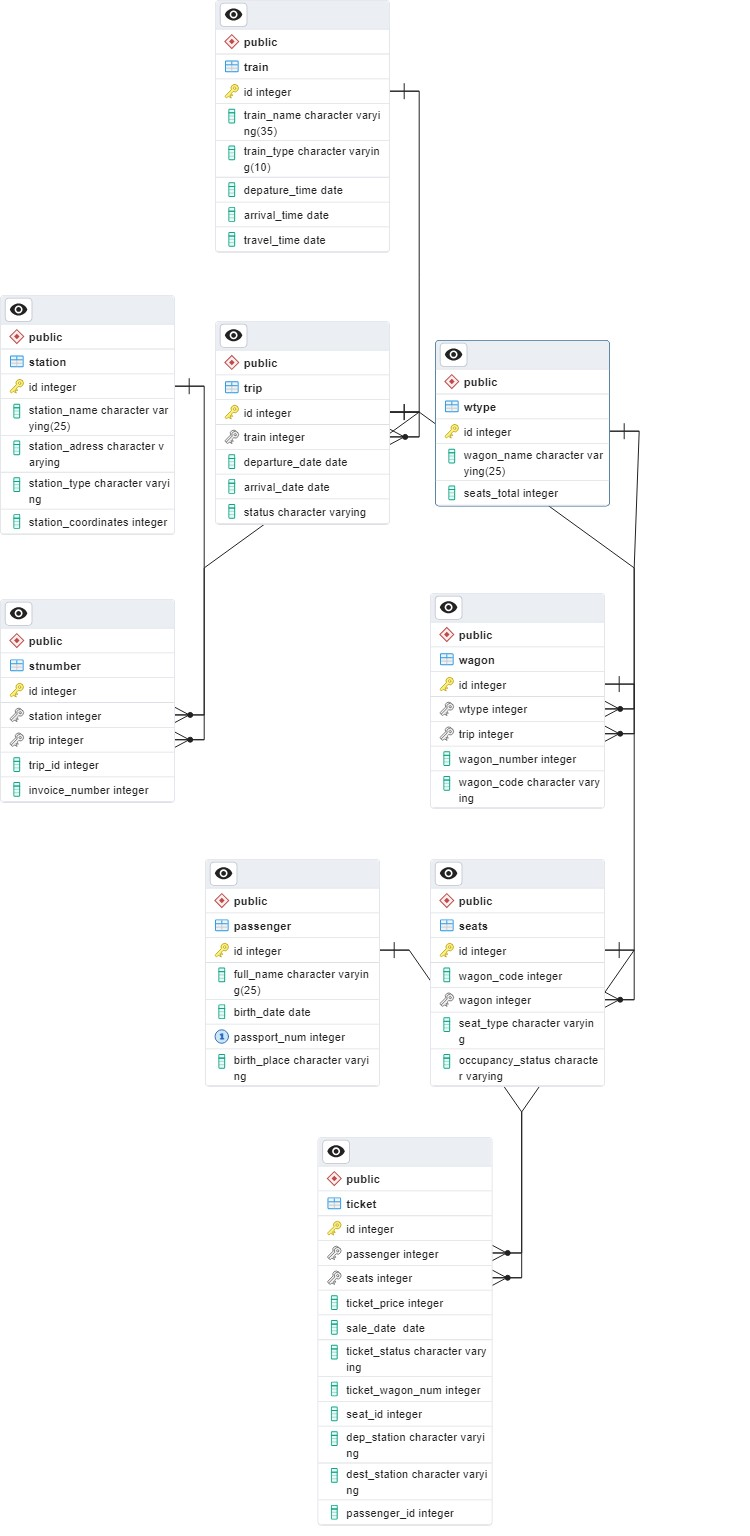
# - при создании резервных копий БД настроить параметры Dump options для Type of objects и Queries.

# Восстановить БД.

# Вариант 6. БД «Пассажир»

**Описание предметной области:**

Схема логической модели базы данных, сгенерированная в Generate ERD указана на рисунке 1.



# Рисунок 1 – Схема логической модели базы данных.

# Листинг дампа:

--

-- PostgreSQL database dump

--

-- Dumped from database version 16.0

-- Dumped by pg\_dump version 16.0

-- Started on 2023-10-26 06:53:41

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;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

-- TOC entry 222 (class 1259 OID 16420)

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

--

-- Создаём таблицу для пассажиров

CREATE TABLE public.passenger (

id integer NOT NULL,

full\_name character varying(25) NOT NULL,

birth\_date date NOT NULL,

passport\_num integer NOT NULL,

birth\_place character varying NOT NULL

);

ALTER TABLE public.passenger OWNER TO postgres;

--

-- TOC entry 221 (class 1259 OID 16417)

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

--

-- Создаём таблицу для мест

CREATE TABLE public.seats (

id integer NOT NULL,

wagon\_code integer NOT NULL,

wagon integer NOT NULL,

seat\_type character varying NOT NULL,

occupancy\_status character varying NOT NULL

);

ALTER TABLE public.seats OWNER TO postgres;

--

-- TOC entry 219 (class 1259 OID 16411)

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

--

--Создаём таюлицу для станций

CREATE TABLE public.station (

id integer NOT NULL,

station\_name character varying(25) NOT NULL,

station\_adress character varying NOT NULL,

station\_type character varying NOT NULL,

station\_coordinates integer NOT NULL

);

ALTER TABLE public.station OWNER TO postgres;

--

-- TOC entry 218 (class 1259 OID 16408)

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

--

-- Создаём таблицы для номеров станций

CREATE TABLE public.stnumber (

id integer NOT NULL,

station integer NOT NULL,

trip integer NOT NULL,

trip\_id integer NOT NULL,

invoice\_number integer NOT NULL

);

ALTER TABLE public.stnumber OWNER TO postgres;

--

-- TOC entry 220 (class 1259 OID 16414)

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

--

-- Создаём таблицу для билетов

CREATE TABLE public.ticket (

id integer NOT NULL,

passenger integer NOT NULL,

seats integer NOT NULL,

ticket\_price integer NOT NULL,

sale\_date date NOT NULL,

ticket\_status character varying NOT NULL,

ticket\_wagon\_num integer NOT NULL,

seat\_id integer NOT NULL,

dep\_station character varying NOT NULL,

dest\_station character varying NOT NULL,

passenger\_id integer NOT NULL

);

ALTER TABLE public.ticket OWNER TO postgres;

--

-- TOC entry 215 (class 1259 OID 16399)

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

--

-- Создаём таблицу для поездов

CREATE TABLE public.train (

id integer NOT NULL,

train\_name character varying(35) NOT NULL,

train\_type character varying(10) NOT NULL,

depature\_time date NOT NULL,

arrival\_time date NOT NULL,

travel\_time date NOT NULL

);

ALTER TABLE public.train OWNER TO postgres;

--

-- TOC entry 216 (class 1259 OID 16402)

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

--

-- Создаём таблицу для рейса

CREATE TABLE public.trip (

id integer NOT NULL,

train integer NOT NULL,

departure\_date date NOT NULL,

arrival\_date date NOT NULL,

status character varying

);

ALTER TABLE public.trip OWNER TO postgres;

--

-- TOC entry 217 (class 1259 OID 16405)

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

--

-- Создаём таблицу для вагонов

CREATE TABLE public.wagon (

id integer NOT NULL,

wtype integer NOT NULL,

trip integer NOT NULL,

wagon\_number integer NOT NULL,

wagon\_code character varying NOT NULL

);

ALTER TABLE public.wagon OWNER TO postgres;

--

-- TOC entry 223 (class 1259 OID 16423)

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

--

-- Создаём таблицу для типов вагона

CREATE TABLE public.wtype (

id integer NOT NULL,

wagon\_name character varying(25) NOT NULL,

seats\_total integer NOT NULL

);

ALTER TABLE public.wtype OWNER TO postgres;

--

-- TOC entry 4898 (class 0 OID 16420)

-- Dependencies: 222

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

--

COPY public.passenger (id, full\_name, birth\_date, passport\_num, birth\_place) FROM stdin;

\.

--

-- TOC entry 4897 (class 0 OID 16417)

-- Dependencies: 221

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

--

COPY public.seats (id, wagon\_code, wagon, seat\_type, occupancy\_status) FROM stdin;

\.

--

-- TOC entry 4895 (class 0 OID 16411)

-- Dependencies: 219

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

--

COPY public.station (id, station\_name, station\_adress, station\_type, station\_coordinates) FROM stdin;

\.

--

-- TOC entry 4894 (class 0 OID 16408)

-- Dependencies: 218

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

--

COPY public.stnumber (id, station, trip, trip\_id, invoice\_number) FROM stdin;

\.

--

-- TOC entry 4896 (class 0 OID 16414)

-- Dependencies: 220

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

--

COPY public.ticket (id, passenger, seats, ticket\_price, "sale\_date ", ticket\_status, ticket\_wagon\_num, seat\_id, dep\_station, dest\_station, passenger\_id) FROM stdin;

\.

--

-- TOC entry 4891 (class 0 OID 16399)

-- Dependencies: 215

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

--

COPY public.train (id, train\_name, train\_type, depature\_time, arrival\_time, travel\_time) FROM stdin;

\.

--

-- TOC entry 4892 (class 0 OID 16402)

-- Dependencies: 216

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

--

COPY public.trip (id, train, departure\_date, arrival\_date, status) FROM stdin;

\.

--

-- TOC entry 4893 (class 0 OID 16405)

-- Dependencies: 217

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

--

COPY public.wagon (id, wtype, trip, wagon\_number, wagon\_code) FROM stdin;

\.

--

-- TOC entry 4899 (class 0 OID 16423)

-- Dependencies: 223

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

--

COPY public.wtype (id, wagon\_name, seats\_total) FROM stdin;

\.

--

-- TOC entry 4735 (class 2606 OID 16431)

-- Name: passenger idpassenger; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.passenger

ADD CONSTRAINT idpassenger PRIMARY KEY (id);

--

-- TOC entry 4733 (class 2606 OID 16433)

-- Name: seats idseat; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.seats

ADD CONSTRAINT idseat PRIMARY KEY (id);

--

-- TOC entry 4729 (class 2606 OID 16435)

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

--

ALTER TABLE ONLY public.station

ADD CONSTRAINT idstation PRIMARY KEY (id);

--

-- TOC entry 4727 (class 2606 OID 16437)

-- Name: stnumber idstnumber; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.stnumber

ADD CONSTRAINT idstnumber PRIMARY KEY (id);

--

-- TOC entry 4731 (class 2606 OID 16439)

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

--

ALTER TABLE ONLY public.ticket

ADD CONSTRAINT idticket PRIMARY KEY (id);

--

-- TOC entry 4721 (class 2606 OID 16429)

-- Name: train idtrain; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.train

ADD CONSTRAINT idtrain PRIMARY KEY (id);

--

-- TOC entry 4723 (class 2606 OID 16441)

-- Name: trip idtrip; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.trip

ADD CONSTRAINT idtrip PRIMARY KEY (id);

--

-- TOC entry 4725 (class 2606 OID 16443)

-- Name: wagon idwagon; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.wagon

ADD CONSTRAINT idwagon PRIMARY KEY (id);

--

-- TOC entry 4739 (class 2606 OID 16445)

-- Name: wtype idwtype; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.wtype

ADD CONSTRAINT idwtype PRIMARY KEY (id);

--

-- TOC entry 4719 (class 2606 OID 16516)

-- Name: wtype seats\_total ; Type: CHECK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE public.wtype

ADD CONSTRAINT "seats\_total " CHECK ((seats\_total <= 50)) NOT VALID;

--

-- TOC entry 4737 (class 2606 OID 16503)

-- Name: passenger uniquepassenger; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.passenger

ADD CONSTRAINT uniquepassenger UNIQUE (passport\_num);

--

-- TOC entry 4747 (class 2606 OID 16462)

-- Name: seats fkseats; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.seats

ADD CONSTRAINT fkseats FOREIGN KEY (wagon) REFERENCES public.wagon(id) NOT VALID;

--

-- TOC entry 4743 (class 2606 OID 16482)

-- Name: stnumber fkstnumber1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.stnumber

ADD CONSTRAINT fkstnumber1 FOREIGN KEY (station) REFERENCES public.station(id) NOT VALID;

--

-- TOC entry 4744 (class 2606 OID 16487)

-- Name: stnumber fkstnumber2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.stnumber

ADD CONSTRAINT fkstnumber2 FOREIGN KEY (trip) REFERENCES public.trip(id) NOT VALID;

--

-- TOC entry 4745 (class 2606 OID 16492)

-- Name: ticket fkticket1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ticket

ADD CONSTRAINT fkticket1 FOREIGN KEY (passenger) REFERENCES public.passenger(id) NOT VALID;

--

-- TOC entry 4746 (class 2606 OID 16497)

-- Name: ticket fkticket2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ticket

ADD CONSTRAINT fkticket2 FOREIGN KEY (seats) REFERENCES public.seats(id) NOT VALID;

--

-- TOC entry 4740 (class 2606 OID 16477)

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

--

ALTER TABLE ONLY public.trip

ADD CONSTRAINT fktrip FOREIGN KEY (train) REFERENCES public.train(id) NOT VALID;

--

-- TOC entry 4741 (class 2606 OID 16467)

-- Name: wagon fkwagon1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.wagon

ADD CONSTRAINT fkwagon1 FOREIGN KEY (wtype) REFERENCES public.wtype(id) NOT VALID;

--

-- TOC entry 4742 (class 2606 OID 16472)

-- Name: wagon fkwagon2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.wagon

ADD CONSTRAINT fkwagon2 FOREIGN KEY (trip) REFERENCES public.trip(id) NOT VALID;

--

-- PostgreSQL database dump complete

--

# Вывод

Во время лабораторного занятия я научился работать с базами данных в PostgreSQL используя pgAdmin 4. Оформил структуру базы, включая таблицы и ограничения, добавил в них данные, сделал бэкапы и удачно восстановил базу данных, удостоверившись в правильности выполненных действий.