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 БАЗЫ ДАННЫХ POSTGRESQL. ЗАПОЛНЕНИЕ ТАБЛИЦ БД РАБОЧИМИ ДАННЫМИ.**

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

|  |  |
| --- | --- |
| Проверил:  Говорова М.М. \_\_\_\_\_\_\_\_\_\_  Дата: «\_\_» \_\_\_\_\_\_\_ 20\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил(и):  студент(ы) группы К3242  Луцкович А.А. |

Санкт-Петербург 2020/2021

# Цель работы

Создание таблиц базы данных PostgreSQL 1Х, заполнение их рабочими данными, осуществление 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 .

7. Восстановить БД.

# Индивидуальное задание

Вариант 8, БД «Аэропорт»

# Выполнение

### Название БД

«Аэропорт»

### Схема инфологической модели данных БД

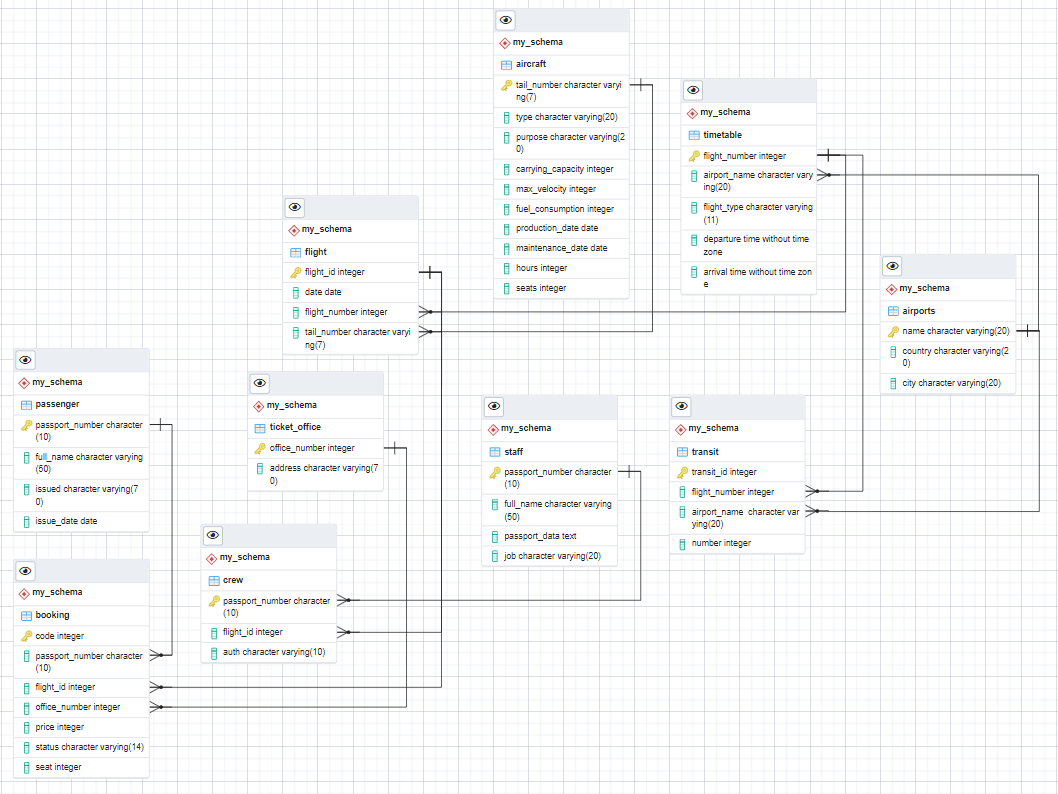


Рисунок 1 – Схема инфологической модели БД, сгенерированная в Generate ERD

### Plain dump

1. Создание базы данных (схемы в бд):

CREATE SCHEMA my\_schema;

ALTER SCHEMA my\_schema OWNER TO postgres;

1. Создание таблицы Самолёт и определение ограничений:

CREATE TABLE my\_schema.aircraft (

tail\_number character varying(7) NOT NULL,

type character varying(20) NOT NULL,

purpose character varying(20) NOT NULL,

carrying\_capacity integer NOT NULL,

max\_velocity integer NOT NULL,

fuel\_consumption integer NOT NULL,

production\_date date NOT NULL,

maintenance\_date date NOT NULL,

hours integer NOT NULL,

seats integer NOT NULL,

CONSTRAINT rule1\_aircraft CHECK (((purpose)::text = ANY ((ARRAY['грузовой'::character varying, 'пассажирский'::character varying])::text[])))

);

ALTER TABLE my\_schema.aircraft OWNER TO postgres;

1. Создание таблицы Аэропорты и определение ограничений:

CREATE TABLE my\_schema.airports (

name character varying(20) NOT NULL,

country character varying(20) NOT NULL,

city character varying(20) NOT NULL

);

ALTER TABLE my\_schema.airports OWNER TO postgres;

1. Создание таблицы Бронирование и определение ограничений:

CREATE TABLE my\_schema.booking (

code integer NOT NULL,

passport\_number character(10) NOT NULL,

flight\_id integer NOT NULL,

office\_number integer NOT NULL,

price integer NOT NULL,

status character varying(14) NOT NULL,

seat integer NOT NULL,

CONSTRAINT rule1\_booking CHECK ((price > 0)),

CONSTRAINT rule3\_booking CHECK (((status)::text = ANY ((ARRAY['зарезервирован'::character varying, 'оплачен'::character varying, 'аннулирован'::character varying])::text[])))

);

ALTER TABLE my\_schema.booking OWNER TO postgres;

CREATE SEQUENCE my\_schema.booking\_code\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE my\_schema.booking\_code\_seq OWNER TO postgres;

ALTER SEQUENCE my\_schema.booking\_code\_seq OWNED BY my\_schema.booking.code;

1. Создание таблицы Экипаж и определение ограничений:

CREATE TABLE my\_schema.crew (

passport\_number character(10) NOT NULL,

flight\_id integer NOT NULL,

auth character varying(10) NOT NULL,

CONSTRAINT rule1\_crew CHECK (((auth)::text = ANY ((ARRAY['допущен'::character varying, 'не допущен'::character varying])::text[])))

);

ALTER TABLE my\_schema.crew OWNER TO postgres;

1. Создание таблицы Выполненный рейс и определение ограничений:

CREATE TABLE my\_schema.flight (

flight\_id integer NOT NULL,

date date NOT NULL,

flight\_number integer NOT NULL,

tail\_number character varying(7) NOT NULL,

CONSTRAINT rule1\_flight CHECK ((date > '1900-01-01'::date))

);

ALTER TABLE my\_schema.flight OWNER TO postgres;

CREATE SEQUENCE my\_schema.flight\_flight\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE my\_schema.flight\_flight\_id\_seq OWNER TO postgres;

ALTER SEQUENCE my\_schema.flight\_flight\_id\_seq OWNED BY my\_schema.flight.flight\_id;

1. Создание таблицы Пассажир и определение ограничений:

CREATE TABLE my\_schema.passenger (

passport\_number character(10) NOT NULL,

full\_name character varying(50) NOT NULL,

issued character varying(70) NOT NULL,

issue\_date date NOT NULL

);

ALTER TABLE my\_schema.passenger OWNER TO postgres;

1. Создание таблицы Сотрудники и определение ограничений:

CREATE TABLE my\_schema.staff (

passport\_number character(10) NOT NULL,

full\_name character varying(50) NOT NULL,

passport\_data text NOT NULL,

job character varying(20) NOT NULL,

CONSTRAINT rule1 CHECK (((job)::text = ANY ((ARRAY['первый пилот'::character varying, 'второй пилот'::character varying, 'стюард'::character varying, 'стюардесса'::character varying])::text[])))

);

ALTER TABLE my\_schema.staff OWNER TO postgres;

1. Создание таблицы Кассы и определение ограничений:

CREATE TABLE my\_schema.ticket\_office (

office\_number integer NOT NULL,

address character varying(70) NOT NULL

);

ALTER TABLE my\_schema.ticket\_office OWNER TO postgres;

CREATE SEQUENCE my\_schema.ticket\_office\_office\_number\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE my\_schema.ticket\_office\_office\_number\_seq OWNER TO postgres;

ALTER SEQUENCE my\_schema.ticket\_office\_office\_number\_seq OWNED BY my\_schema.ticket\_office.office\_number;

1. Создание таблицы Расписание и определение ограничений:

CREATE TABLE my\_schema.timetable (

flight\_number integer NOT NULL,

airport\_name character varying(20) NOT NULL,

flight\_type character varying(11) NOT NULL,

departure time without time zone NOT NULL,

arrival time without time zone NOT NULL

);

ALTER TABLE my\_schema.timetable OWNER TO postgres;

1. Создание таблицы Транзиты и определение ограничений:

CREATE TABLE my\_schema.transit (

transit\_id integer NOT NULL,

flight\_number integer NOT NULL,

"airport\_name " character varying(20) NOT NULL,

number integer NOT NULL

);

ALTER TABLE my\_schema.transit OWNER TO postgres;

CREATE SEQUENCE my\_schema.transit\_number\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE my\_schema.transit\_number\_seq OWNER TO postgres;

ALTER SEQUENCE my\_schema.transit\_number\_seq OWNED BY my\_schema.transit.transit\_id;

1. Внесение изменений в уже созданные таблицы:

ALTER TABLE ONLY my\_schema.booking ALTER COLUMN code SET DEFAULT nextval('my\_schema.booking\_code\_seq'::regclass);

ALTER TABLE ONLY my\_schema.flight ALTER COLUMN flight\_id SET DEFAULT nextval('my\_schema.flight\_flight\_id\_seq'::regclass);

ALTER TABLE ONLY my\_schema.ticket\_office ALTER COLUMN office\_number SET DEFAULT nextval('my\_schema.ticket\_office\_office\_number\_seq'::regclass);

ALTER TABLE ONLY my\_schema.transit ALTER COLUMN transit\_id SET DEFAULT nextval('my\_schema.transit\_number\_seq'::regclass);

1. Заполнение таблиц рабочими данными:

COPY my\_schema.aircraft (tail\_number, type, purpose, carrying\_capacity, max\_velocity, fuel\_consumption, production\_date, maintenance\_date, hours, seats) FROM stdin;

GB-5246 Airbus грузовой 578 603 145 1975-12-17 2019-11-24 257257 6

VT-B462 Boeing пассажирский 450 990 200 1998-02-19 2020-09-26 32451 50

\.

COPY my\_schema.airports (name, country, city) FROM stdin;

домодедово россия москва

внуково россия москва

схипхол нидерланды амстердам

каструп дания копенгаген

\.

COPY my\_schema.booking (code, passport\_number, flight\_id, office\_number, price, status, seat) FROM stdin;

1 8067482519 1 2 3600 оплачен 34

2 8098372164 2 3 26000 зарезервирован 16

\.

COPY my\_schema.crew (passport\_number, flight\_id, auth) FROM stdin;

\.

COPY my\_schema.flight (flight\_id, date, flight\_number, tail\_number) FROM stdin;

1 2021-05-16 6413 GB-5246

2 2021-06-04 4562 VT-B462

\.

COPY my\_schema.passenger (passport\_number, full\_name, issued, issue\_date) FROM stdin;

8094543523 Гром Игорь Константинович выдан МВД города Санкт-Петербурга 1994-12-05

8098372164 Пчелкина Юлия Евгеньевна выдан отделом ОМС Фрунзенского района города Санкт-Петербурга 2005-09-02

8067482519 Волков Олег Кириллович выдан МВД города Москвы 1999-02-17

\.

COPY my\_schema.staff (passport\_number, full\_name, passport\_data, job) FROM stdin;

8057648274 Леонова Александра Кирилловна дата рождения, кем выдан паспорт, когда выдан стюардесса

8095736417 Лаврентьев Артём Арсенович дата рождения, кем выдан паспорт, когда выдан первый пилот

8096354971 Ковалев Тимофей Максимович дата рождения, кем выдан паспорт, когда выдан второй пилот

\.

COPY my\_schema.ticket\_office (office\_number, address) FROM stdin;

1 Московский проспект 174

2 Садовая 45

3 Вознесенская 103

\.

COPY my\_schema.timetable (flight\_number, airport\_name, flight\_type, departure, arrival) FROM stdin;

6413 домодедово чартерный 05:30:00 07:20:00

3812 каструп регулярный 17:30:00 22:10:00

4562 схипхол трансферный 23:20:00 06:45:00

\.

COPY my\_schema.transit (transit\_id, flight\_number, "airport\_name ", number) FROM stdin;

1 4562 внуково 1

2 4562 каструп 2

\.

1. Назначение первичных и внешних ключей, а также дополнительных ограничений:

ALTER TABLE ONLY my\_schema.aircraft

ADD CONSTRAINT aircraft\_pkey PRIMARY KEY (tail\_number);

ALTER TABLE ONLY my\_schema.airports

ADD CONSTRAINT airports\_pkey PRIMARY KEY (name);

ALTER TABLE ONLY my\_schema.booking

ADD CONSTRAINT booking\_pkey PRIMARY KEY (code);

ALTER TABLE ONLY my\_schema.crew

ADD CONSTRAINT crew\_pkey PRIMARY KEY (passport\_number);

ALTER TABLE ONLY my\_schema.flight

ADD CONSTRAINT flight\_pkey PRIMARY KEY (flight\_id);

ALTER TABLE ONLY my\_schema.passenger

ADD CONSTRAINT passenger\_pkey PRIMARY KEY (passport\_number);

ALTER TABLE ONLY my\_schema.transit

ADD CONSTRAINT pkey PRIMARY KEY (transit\_id);

ALTER TABLE my\_schema.passenger

ADD CONSTRAINT rule1\_passenger CHECK ((issue\_date > '1900-01-01'::date)) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule2\_aircraft CHECK (((type)::text = ANY ((ARRAY['Airbus'::character varying, 'Boeing'::character varying, 'ИЛ'::character varying, 'ТУ'::character varying])::text[]))) NOT VALID;

ALTER TABLE my\_schema.booking

ADD CONSTRAINT rule2\_booking CHECK (((seat > 0) AND (seat < 100))) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule3\_aircraft CHECK ((maintenance\_date > production\_date)) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule4\_aircraft CHECK (((max\_velocity > 399) AND (max\_velocity < 1001))) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule5\_aircraft CHECK ((max\_velocity > 0)) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule6\_aircraft CHECK ((fuel\_consumption > 0)) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule7\_aircraft CHECK ((hours > 0)) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule8\_aircraft CHECK ((seats > 0)) NOT VALID;

ALTER TABLE my\_schema.aircraft

ADD CONSTRAINT rule9\_aircraft CHECK ((production\_date > '1900-01-01'::date)) NOT VALID;

ALTER TABLE ONLY my\_schema.staff

ADD CONSTRAINT staff\_pkey PRIMARY KEY (passport\_number);

ALTER TABLE ONLY my\_schema.ticket\_office

ADD CONSTRAINT ticket\_office\_pkey PRIMARY KEY (office\_number);

ALTER TABLE ONLY my\_schema.timetable

ADD CONSTRAINT timetable\_pkey PRIMARY KEY (flight\_number);

ALTER TABLE ONLY my\_schema.flight

ADD CONSTRAINT fk\_aircraft FOREIGN KEY (tail\_number) REFERENCES my\_schema.aircraft(tail\_number) NOT VALID;

ALTER TABLE ONLY my\_schema.crew

ADD CONSTRAINT fk\_flight FOREIGN KEY (flight\_id) REFERENCES my\_schema.flight(flight\_id) NOT VALID;

ALTER TABLE ONLY my\_schema.booking

ADD CONSTRAINT fk\_flight FOREIGN KEY (flight\_id) REFERENCES my\_schema.flight(flight\_id) NOT VALID;

ALTER TABLE ONLY my\_schema.transit

ADD CONSTRAINT fk\_name FOREIGN KEY ("airport\_name ") REFERENCES my\_schema.airports(name) NOT VALID;

ALTER TABLE ONLY my\_schema.timetable

ADD CONSTRAINT fk\_name FOREIGN KEY (airport\_name) REFERENCES my\_schema.airports(name) NOT VALID;

ALTER TABLE ONLY my\_schema.transit

ADD CONSTRAINT fk\_num FOREIGN KEY (flight\_number) REFERENCES my\_schema.timetable(flight\_number) NOT VALID;

ALTER TABLE ONLY my\_schema.flight

ADD CONSTRAINT fk\_num FOREIGN KEY (flight\_number) REFERENCES my\_schema.timetable(flight\_number) NOT VALID;

ALTER TABLE ONLY my\_schema.booking

ADD CONSTRAINT fk\_office FOREIGN KEY (office\_number) REFERENCES my\_schema.ticket\_office(office\_number) NOT VALID;

ALTER TABLE ONLY my\_schema.booking

ADD CONSTRAINT fk\_passport FOREIGN KEY (passport\_number) REFERENCES my\_schema.passenger(passport\_number) NOT VALID;

ALTER TABLE ONLY my\_schema.crew

ADD CONSTRAINT fk\_staff FOREIGN KEY (passport\_number) REFERENCES my\_schema.staff(passport\_number) NOT VALID;

# выводы

В ходе выполнения лабораторной работы были созданы таблицы базы данных PostgreSQL 1Х. Были установлены ограничения на данные (первичный и внешний ключи, проверки на наличие и корректность значения). Затем таблицы были заполнены рабочими данными. Были созданы две резервные копии (в текстовом и кастомном вариантах), первая использовалась для листинга в отчете, а с помощью второй было произведено восстановление базы данных.