

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
СРЕДСТВАМИ PGADMIN»
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:
студентка группы
К3243
Абрамова А.М.

Санкт-Петербург 2020/2021

Цель работы: 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. 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:

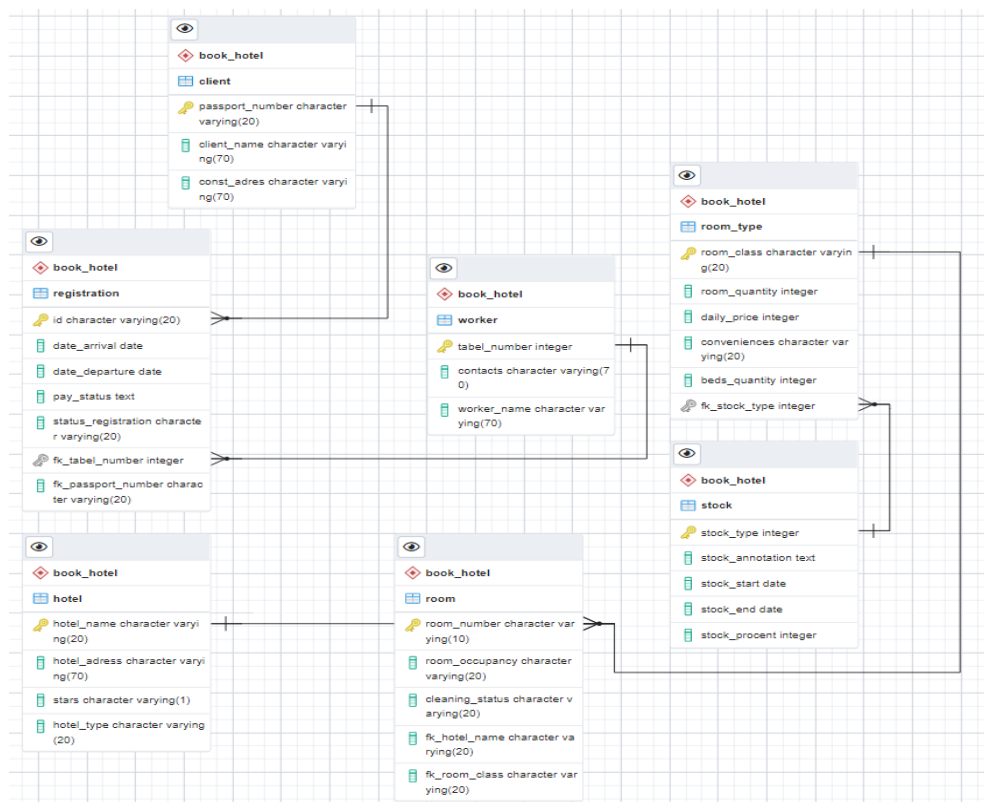
Создать две резервные копии:

- 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.

Выполнение:

Наименование: БД book_hotel

Схема логической модели базы данных, сгенерированная в Generate ERD



3. Dump, содержащий скрипты работы с БД.

```
/Создание базы данных/
--
-- PostgreSQL database dump
--

-- Dumped from database version 13.2
-- Dumped by pg_dump version 13.2

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;

/Создание схемы book_hotel/
--
-- Name: book_hotel; Type: SCHEMA; Schema: -; Owner: postgres
--

CREATE SCHEMA book_hotel;

ALTER SCHEMA book_hotel OWNER TO postgres;

--
-- Name: adminpack; Type: EXTENSION; Schema: -; Owner: -
--

CREATE EXTENSION IF NOT EXISTS adminpack WITH SCHEMA pg_catalog;

--
-- Name: EXTENSION adminpack; Type: COMMENT; Schema: -; Owner:
--

COMMENT ON EXTENSION adminpack IS 'administrative functions for PostgreSQL';

SET default_tablespace = '';

SET default_table_access_method = heap;
```

/Создание таблицы stock/

```
--  
-- Name: stock; Type: TABLE; Schema: book_hotel; Owner: postgres  
--
```

```
CREATE TABLE book_hotel." stock" (  
    stock_type integer NOT NULL,  
    stock_annotation text,  
    stock_start date,  
    stock_end date,  
    stock_procent integer  
);
```

```
ALTER TABLE book_hotel." stock" OWNER TO postgres;
```

/Создание таблицы client/

```
--  
-- Name: client; Type: TABLE; Schema: book_hotel; Owner: postgres  
--
```

```
CREATE TABLE book_hotel.client (  
    passport_number character varying(20) NOT NULL,  
    client_name character varying(70),  
    const_adres character varying(70)  
);
```

```
ALTER TABLE book_hotel.client OWNER TO postgres;
```

/Создание таблицы hotel/

```
--  
-- Name: hotel; Type: TABLE; Schema: book_hotel; Owner: postgres  
--
```

```
CREATE TABLE book_hotel.hotel (  
    hotel_name character varying(20) NOT NULL,  
    hotel_adres character varying(70),  
    stars character varying(1),  
    hotel_type character varying(20),  
    CONSTRAINT chk_stars CHECK (((stars)::text = ANY (ARRAY[('1'::character varying)::text, ('2'::character varying)::text, ('3'::character varying)::text, ('4'::character varying)::text, ('5'::character varying)::text])))  
);
```

```
ALTER TABLE book_hotel.hotel OWNER TO postgres;
```

/Создание таблицы registration/

```
--  
-- Name: registration; Type: TABLE; Schema: book_hotel; Owner: postgres  
--
```

```
CREATE TABLE book_hotel.registration (  
    id character varying(20) NOT NULL,  
    date_arrival date,  
    date_departure date,  
    pay_status text,  
    status_registration character varying(20),  
    fk_tabel_number integer,  
    fk_passport_number character varying(20)  
);
```

```
ALTER TABLE book_hotel.registration OWNER TO postgres;
```

/Создание таблицы room/

```
--  
-- Name: room; Type: TABLE; Schema: book_hotel; Owner: postgres  
--
```

```
CREATE TABLE book_hotel.room (  
    room_number character varying(10) NOT NULL,  
    room_occupancy character varying(20),  
    cleaning_status character varying(20),  
    fk_hotel_name character varying(20),  
    fk_room_class character varying(20)  
);
```

```
ALTER TABLE book_hotel.room OWNER TO postgres;
```

/Создание таблицы room_type/

```
--  
-- Name: room_type; Type: TABLE; Schema: book_hotel; Owner: postgres  
--
```

```
CREATE TABLE book_hotel.room_type (  
    room_class character varying(20) NOT NULL,  
    room_quantity integer,  
    daily_price integer,  
    conveniences character varying(20),  
    beds_quantity integer,  
    fk_stock_type integer  
);
```

```
ALTER TABLE book_hotel.room_type OWNER TO postgres;
```

```

/Создание таблицы worker/
--
-- Name: worker; Type: TABLE; Schema: book_hotel; Owner: postgres
--

CREATE TABLE book_hotel.worker (
    tabel_number integer NOT NULL,
    contacts character varying(70),
    worker_name character varying(70)
);

ALTER TABLE book_hotel.worker OWNER TO postgres;

--
-- Data for Name: stock; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--
/Загрузка данных/
COPY book_hotel." stock" (stock_type, stock_annotation, stock_start, stock_end, s
tock_procent) FROM stdin;
1 скидка студентам 5% 2020-05-14 2023-05-14 5
2 на майские праздники дешевле на 3 % 2021-05-01 2021-05-10 3
\.

--
-- Data for Name: client; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--

COPY book_hotel.client (passport_number, client_name, const_adres) FROM stdin;
5623 456789 Иванов Иван Иванович г. Пенза, ул. Ленина, д. 10
5444 456789\n Сидоров Сидор Сидорович г. Санкт-
Перербург, Вяземский переулок 5/7
1234 567890 Котов Кот Котович г. Воронеж, ул. Батонная, д. 10
\.

--
-- Data for Name: hotel; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--

COPY book_hotel.hotel (hotel_name, hotel_adress, stars, hotel_type) FROM stdin;
GREENFLOW г. Сочи, Красная поляна, д. 6 4 горнолыжный отель
У Маруси г. Пенза, ул. Московская, д.5 1 апарт-отель
Астра г. Владивосток, ул. Калинина, д. 67 3 бутик-отель
\.

--

```

```

-
- Data for Name: registration; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--

COPY book_hotel.registration (id, date_arrival, date_departure, pay_status, status_registration, fk_tabel_number, fk_passport_number) FROM stdin;
\.
```

```

--
-- Data for Name: room; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--

COPY book_hotel.room (room_number, room_occupancy, cleaning_status, fk_hotel_name, fk_room_class) FROM stdin;
123 занят не беспокоить \N \N
432 свободен убран \N \N
8765 забронирован требуется уборка \N \N
\.
```

```

--
-
- Data for Name: room_type; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--

COPY book_hotel.room_type (room_class, room_quantity, daily_price, conveniences, beds_quantity, fk_stock_type) FROM stdin;
стандарт 2 1234 wi-fi 2 \N
люкс 4 543234 wi-fi, мини-бар 4 \N
полулюкс 3 65434 фен 2 \N
\.
```

```

--
-- Data for Name: worker; Type: TABLE DATA; Schema: book_hotel; Owner: postgres
--

COPY book_hotel.worker (tabel_number, contacts, worker_name) FROM stdin;
12345 89874444444 Петров Петр Пктрович
76543 89213333333 Зайцев Заяц Зайцевич
\.
```

```

--
-
- Name: stock stock_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel." stock"

```

```

ADD CONSTRAINT " stock_pkey" PRIMARY KEY (stock_type);

--
-
- Name: room_type chk_beds_quantity; Type: CHECK CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE book_hotel.room_type
    ADD CONSTRAINT chk_beds_quantity CHECK ((beds_quantity > 0)) NOT VALID;

--
-
- Name: room chk_cleaning_status; Type: CHECK CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE book_hotel.room
    ADD CONSTRAINT chk_cleaning_status CHECK (((cleaning_status)::text = ANY (ARRAY[('требуется уборка'::character varying)::text, ('убран'::character varying)::text, ('не беспокоить'::character varying)::text]))) NOT VALID;

--
-
- Name: room_type chk_daily_price; Type: CHECK CONSTRAINT; Schema: book_hotel; Owner: postgres
--
/Добавление ограничений/
ALTER TABLE book_hotel.room_type
    ADD CONSTRAINT chk_daily_price CHECK ((daily_price > 0)) NOT VALID;

--
-
- Name: hotel chk_hotel_type; Type: CHECK CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE book_hotel.hotel
    ADD CONSTRAINT chk_hotel_type CHECK (((hotel_type)::text = ANY (ARRAY[('апартамент-отель'::character varying)::text, ('бутик-отель'::character varying)::text, ('горнолыжный отель'::character varying)::text]))) NOT VALID;

--

```



```

-
- Name: registration chk_pay_status; Type: CHECK CONSTRAINT; Schema: book_hotel;
Owner: postgres
--

ALTER TABLE book_hotel.registration
    ADD CONSTRAINT chk_pay_status CHECK ((pay_status = ANY (ARRAY[('оплачено'::ch
aracter varying)::text, ('не оплачено'::character varying)::text]))) NOT VALID;

--
-
- Name: room_type chk_room_class; Type: CHECK CONSTRAINT; Schema: book_hotel; Own
er: postgres
--

ALTER TABLE book_hotel.room_type
    ADD CONSTRAINT chk_room_class CHECK (((room_class)::text = ANY (ARRAY[('станд
арт'::character varying)::text, ('люкс'::character varying)::text, ('полулюкс'::c
haracter varying)::text]))) NOT VALID;

--
-
- Name: room chk_room_occupancy; Type: CHECK CONSTRAINT; Schema: book_hotel; Owne
r: postgres
--

ALTER TABLE book_hotel.room
    ADD CONSTRAINT chk_room_occupancy CHECK (((room_occupancy)::text = ANY (ARRAY
[('занят'::character varying)::text, ('забронирован'::character varying)::text, (
'свободен'::character varying)::text]))) NOT VALID;

--
-
- Name: room_type chk_room_quntity; Type: CHECK CONSTRAINT; Schema: book_hotel; O
wner: postgres
--

ALTER TABLE book_hotel.room_type
    ADD CONSTRAINT chk_room_quntity CHECK ((room_quantity > 0)) NOT VALID;

--
-
- Name: stock chk_stock_procent; Type: CHECK CONSTRAINT; Schema: book_hotel; Own
er: postgres
--

ALTER TABLE book_hotel." stock"

```

```

    ADD CONSTRAINT chk_stock_procent CHECK (((stock_procent > 0) AND (stock_procent <= 100))) NOT VALID;

--
-
- Name: worker chk_tabel_number; Type: CHECK CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE book_hotel.worker
    ADD CONSTRAINT chk_tabel_number CHECK ((tabel_number > 0)) NOT VALID;

--
-
- Name: client client_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel.client
    ADD CONSTRAINT client_pkey PRIMARY KEY (passport_number);

--
-- Name: hotel hotel_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel.hotel
    ADD CONSTRAINT hotel_pkey PRIMARY KEY (hotel_name);

--
-
- Name: registration registration_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel.registration
    ADD CONSTRAINT registration_pkey PRIMARY KEY (id);

--
-- Name: room room_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel.room
    ADD CONSTRAINT room_pkey PRIMARY KEY (room_number);

--

```

```

-
- Name: room_type room_type_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: po
stgres
--

ALTER TABLE ONLY book_hotel.room_type
    ADD CONSTRAINT room_type_pkey PRIMARY KEY (room_class);

--
-
- Name: worker worker_pkey; Type: CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel.worker
    ADD CONSTRAINT worker_pkey PRIMARY KEY (tabel_number);

--
-
- Name: registration fk_reg_to_cl; Type: FK CONSTRAINT; Schema: book_hotel; Owner
: postgres
--

ALTER TABLE ONLY book_hotel.registration
    ADD CONSTRAINT fk_reg_to_cl FOREIGN KEY (id) REFERENCES book_hotel.client(pas
sport_number) NOT VALID;

--
-
- Name: registration fk_reg_to_work; Type: FK CONSTRAINT; Schema: book_hotel; Own
er: postgres
--

ALTER TABLE ONLY book_hotel.registration
    ADD CONSTRAINT fk_reg_to_work FOREIGN KEY (fk_tabel_number) REFERENCES book_h
otel.worker(tabel_number) NOT VALID;

--
-
- Name: room fk_room_to_hot; Type: FK CONSTRAINT; Schema: book_hotel; Owner: post
gres
--

ALTER TABLE ONLY book_hotel.room
    ADD CONSTRAINT fk_room_to_hot FOREIGN KEY (room_number) REFERENCES book_hotel
.hotel(hotel_name) NOT VALID;

```

```
--
-
- Name: room_fk_room_to_rt; Type: FK CONSTRAINT; Schema: book_hotel; Owner: postgres
--

ALTER TABLE ONLY book_hotel.room
    ADD CONSTRAINT fk_room_to_rt FOREIGN KEY (room_number) REFERENCES book_hotel.
room_type(room_class) NOT VALID;

--
-
- Name: room_type_fk_rt_to_stock; Type: FK CONSTRAINT; Schema: book_hotel; Owner:
postgres
--

ALTER TABLE ONLY book_hotel.room_type
    ADD CONSTRAINT fk_rt_to_stock FOREIGN KEY (fk_stock_type) REFERENCES book_hot
el." stock"(stock_type) NOT VALID;

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

ВЫВОДЫ: в ходе проделанной работы была создана база данных, схема и таблицы, определены типы данных, ограничения, первичные и внешние ключи. В базу загружены данные. Сделана резервная копия базы данных в текстовом виде и в виде архива, также реализовано восстановление базы.