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 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «__» _____ 20__г.
Оценка _____

Выполнил:
студент группы К3240
Козлов И.Д.

Санкт-Петербург 2022

ЦЕЛЬ РАБОТЫ

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

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Оборудование: компьютерный класс.
Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.
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 .*

1. Восстановить БД.

Вариант 1. БД «Отель»

Описание предметной области: Отели сети находятся в разных городах. Цены на номера одного типа во всех отелях одинаковы и зависят от типа номера и количества мест. Номер может быть забронирован, занят или свободен. При заезде в отель постояльцы проходят регистрацию. Информация о регистрации постояльцев отеля (выехавших из отеля) хранится в течение года и 1 января удаляется в архив.

БД должна содержать следующий минимальный набор сведений: Адрес отеля. Название отеля. Номер комнаты. Тип комнаты. Количество мест. Цена комнаты за сутки проживания. Имя постояльца. Фамилия постояльца. Отчество постояльца. Адрес постоянного проживания. Дата заезда. Дата отъезда.

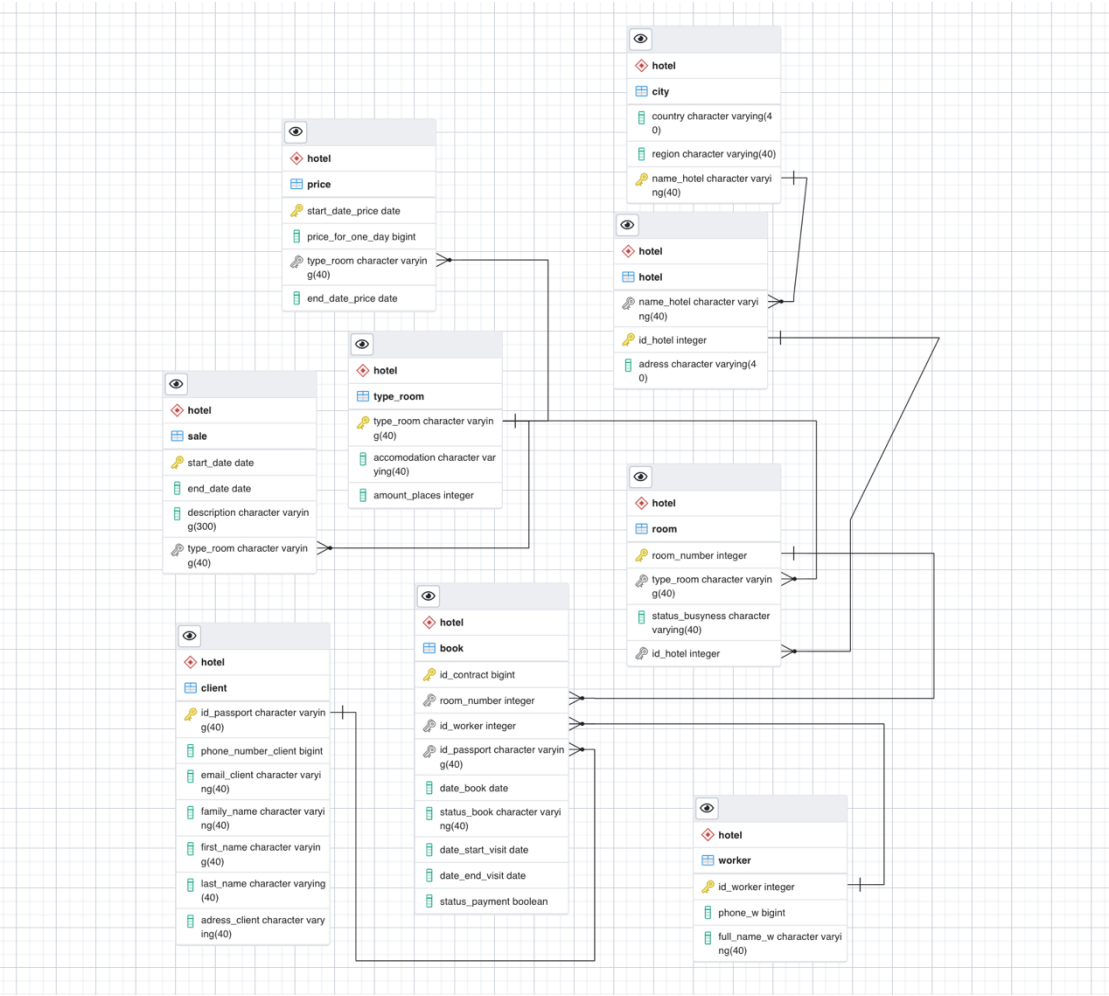
Дополнить исходные данные информацией: по бронированию комнаты; по сотруднику, который регистрирует постояльца в отеле в день заезда; по оплате проживания; по составу удобств в комнате; по акциям, доступным при бронировании (скидки).

ХОД РАБОТЫ

1) **наименование БД:**

- Hotel

2) **схема логической модели:**



3) **Dumb**

```
/*Создание базы данных*/

CREATE DATABASE hotel
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'C'
    LC_CTYPE = 'C'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;

/*Создание схемы*/

CREATE SCHEMA IF NOT EXISTS hotel
    AUTHORIZATION postgres;
```

/*Создание таблицы book и определение ограничений*/

```
CREATE TABLE IF NOT EXISTS hotel.book
(
    id_contract bigint NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),
    room_number integer NOT NULL,
    id_worker integer NOT NULL,
    id_passport character varying(40) COLLATE pg_catalog."default" NOT NULL,
    date_book date,
    status_book character varying(40) COLLATE pg_catalog."default",
    date_start_visit date NOT NULL,
    date_end_visit date NOT NULL,
    status_payment boolean NOT NULL,
    CONSTRAINT book_pkey PRIMARY KEY (id_contract),
    CONSTRAINT book_id_contract_id_contract1_key UNIQUE (id_contract)
        INCLUDE(id_contract),
    CONSTRAINT book_id_passport_fkey FOREIGN KEY (id_passport)
        REFERENCES hotel.client (id_passport) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT book_id_worker_fkey FOREIGN KEY (id_worker)
        REFERENCES hotel.worker (id_worker) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT book_room_number_fkey FOREIGN KEY (room_number)
        REFERENCES hotel.room (room_number) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT book_check CHECK (date_start_visit < date_end_visit) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.book
    OWNER to postgres;
```

/*Создание таблицы city и определение ограничений */

```
CREATE TABLE IF NOT EXISTS hotel.city
(
    country character varying(40) COLLATE pg_catalog."default" NOT NULL,
    region character varying(40) COLLATE pg_catalog."default" NOT NULL,
    name_hotel character varying(40) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT city_pkey PRIMARY KEY (name_hotel),
    CONSTRAINT city_name_hotel_name_hotel1_key UNIQUE (name_hotel)
        INCLUDE(name_hotel)
)
```

```
TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.city
    OWNER to postgres;
```

/*Создание таблицы client и определение ограничений */

```
CREATE TABLE IF NOT EXISTS hotel.client
(
    id_passport character varying(40) COLLATE pg_catalog."default" NOT NULL,
    email_client character varying(40) COLLATE pg_catalog."default",
    family_name character varying(40) COLLATE pg_catalog."default" NOT NULL,
    first_name character varying(40) COLLATE pg_catalog."default" NOT NULL,
    last_name character varying(40) COLLATE pg_catalog."default" NOT NULL,
    adress_client character varying(40) COLLATE pg_catalog."default" NOT NULL,
    phone_number_client character varying(15) COLLATE pg_catalog."default",
    CONSTRAINT client_pkey PRIMARY KEY (id_passport),
    CONSTRAINT client_email_client_email_client1_key UNIQUE (email_client)
        INCLUDE(email_client),
    CONSTRAINT client_id_passport_id_passport1_key UNIQUE (id_passport)
        INCLUDE(id_passport),
    CONSTRAINT client_phone_number_client_key UNIQUE (phone_number_client),
    CONSTRAINT client_phone_number_client_check CHECK (phone_number_client::text
~~ '+%'::text) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.client
    OWNER to postgres;
```

/*Создание таблицы hotel и определение ограничений */

```
CREATE TABLE IF NOT EXISTS hotel.hotel
(
    name_hotel character varying(40) COLLATE pg_catalog."default" NOT NULL,
    id_hotel integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1
START 0 MINVALUE 0 MAXVALUE 2147483647 CACHE 1 ),
    adress character varying(40) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT hotel_pkey PRIMARY KEY (id_hotel),
    CONSTRAINT hotel_adress_adress1_key UNIQUE (adress)
        INCLUDE(adress),
    CONSTRAINT hotel_id_hotel_id_hotel1_key UNIQUE (id_hotel)
        INCLUDE(id_hotel),
    CONSTRAINT hotel_name_hotel_fkey FOREIGN KEY (name_hotel)
        REFERENCES hotel.city (name_hotel) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

```
TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.hotel
    OWNER to postgres;
```

/*Создание таблицы price и определение ограничений*/

```
CREATE TABLE IF NOT EXISTS hotel.price
(
    price_for_one_day bigint NOT NULL,
    type_room character varying(40) COLLATE pg_catalog."default" NOT NULL,
    end_date_price date NOT NULL,
    id_price bigint NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START
1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),
    start_date_price date NOT NULL,
    CONSTRAINT price_pkey PRIMARY KEY (id_price)
        INCLUDE(id_price),
    CONSTRAINT price_price_for_one_day_price_for_one_day1_key UNIQUE
(price_for_one_day)
        INCLUDE(price_for_one_day),
    CONSTRAINT price_type_room_fkey FOREIGN KEY (type_room)
        REFERENCES hotel.type_room (type_room) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.price
    OWNER to postgres;
```

/*Создание таблицы room и определение ограничений*/

```
CREATE TABLE IF NOT EXISTS hotel.room
(
    room_number integer NOT NULL,
    type_room character varying(40) COLLATE pg_catalog."default" NOT NULL,
    status_busyness character varying(20) COLLATE pg_catalog."default" NOT NULL,
    id_hotel integer NOT NULL,
    CONSTRAINT room_pkey PRIMARY KEY (room_number),
    CONSTRAINT room_room_number_room_number1_key UNIQUE (room_number)
        INCLUDE(room_number),
    CONSTRAINT room_id_hotel_fkey FOREIGN KEY (id_hotel)
        REFERENCES hotel.hotel (id_hotel) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT room_type_room_fkey FOREIGN KEY (type_room)
        REFERENCES hotel.type_room (type_room) MATCH SIMPLE
```

```
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
  CONSTRAINT room_room_number_check CHECK (room_number > 0),
  CONSTRAINT room_status_busyness_check CHECK (status_busyness::text = ANY
(ARRAY['Занят'::character varying, 'Не занят'::character varying]::text[])) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.room
  OWNER to postgres;
```

/*Создание таблицы sale и определение ограничений */

```
CREATE TABLE IF NOT EXISTS hotel.sale
(
  end_date date,
  description character varying(300) COLLATE pg_catalog."default",
  type_room character varying(40) COLLATE pg_catalog."default" NOT NULL,
  id_sale integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1
MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
  start_date date,
  CONSTRAINT sale_pkey PRIMARY KEY (id_sale),
  CONSTRAINT sale_type_room_fkey FOREIGN KEY (type_room)
    REFERENCES hotel.type_room (type_room) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
  CONSTRAINT sale_check CHECK (start_date < end_date) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.sale
  OWNER to postgres;
```

/*Создание таблицы type_room и определение ограничений*/

```
CREATE TABLE IF NOT EXISTS hotel.type_room
(
  type_room character varying(40) COLLATE pg_catalog."default" NOT NULL,
  accomodation character varying(40) COLLATE pg_catalog."default",
  amount_places integer NOT NULL,
  CONSTRAINT type_room_pkey PRIMARY KEY (type_room),
  CONSTRAINT type_room_amount_places_check CHECK (amount_places > 0)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.type_room
```

```
    OWNER to postgres;
```

/*Создание таблицы worker и определение ограничений*/

```
CREATE TABLE IF NOT EXISTS hotel.worker
(
    id_worker integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    full_name_w character varying(40) COLLATE pg_catalog."default" NOT NULL,
    phone_w character varying(15) COLLATE pg_catalog."default",
    CONSTRAINT worker_pkey PRIMARY KEY (id_worker),
    CONSTRAINT worker_id_worker_id_worker1_key UNIQUE (id_worker)
        INCLUDE(id_worker),
    CONSTRAINT worker_phone_w_key UNIQUE (phone_w),
    CONSTRAINT worker_phone_w_check CHECK (phone_w::text ~~ '+%'::text) NOT
VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS hotel.worker
    OWNER to postgres;
```

/*Заполнение таблицы book рабочими данными */

```
INSERT INTO hotel.book (
room_number, id_worker, id_passport, date_start_visit, date_end_visit,
status_payment) VALUES (
'1'::integer, '6'::integer, '12345675812'::character varying, '2022-03-02'::date, '2022-03-
13'::date, true::boolean)
 returning id_contract;

INSERT INTO hotel.book (
room_number, id_worker, id_passport, date_book, status_book, date_start_visit,
date_end_visit, status_payment) VALUES (
'2'::integer, '6'::integer, '12345675813'::character varying, '2022-03-05'::date,
'успешно'::character varying, '2022-03-05'::date, '2022-03-20'::date, true::boolean)
 returning id_contract;

INSERT INTO hotel.book (
room_number, id_worker, id_passport, date_start_visit, date_end_visit,
status_payment) VALUES (
'3'::integer, '8'::integer, '12345675814'::character varying, '2022-03-07'::date, '2022-03-
24'::date, true::boolean)
 returning id_contract;
```

/*Заполнение таблицы city рабочими данными */

```
INSERT INTO hotel.city (
country, region, name_hotel) VALUES (
'Россия'::character varying, 'Московская'::character varying, 'Мариот'::character
varying)
 returning name_hotel;

INSERT INTO hotel.city (
country, region, name_hotel) VALUES (
'Россия'::character varying, 'Иркутская'::character varying, 'Ангара'::character varying)
 returning name_hotel;

INSERT INTO hotel.city (
country, region, name_hotel) VALUES (
'Россия'::character varying, 'Ленинградская'::character varying, 'Миссия'::character
varying)
 returning name_hotel;
```

/*Заполнение таблицы client рабочими данными */

```
INSERT INTO hotel.client (
id_passport, phone_number_client, family_name, first_name, last_name, adress_client)
VALUES (
'12345675814'::character varying, '89812678912'::bigint, 'Данииловна'::character
varying, 'Алина'::character varying, 'Щипцова'::character varying, 'Маяковского
5'::character varying)
 returning id_passport;

INSERT INTO hotel.client (
id_passport, phone_number_client, email_client, family_name, first_name, last_name,
adress_client) VALUES (
'12345675813'::character varying, '89812678911'::bigint, 'maria@mail.ru'::character
varying, 'Егоровна'::character varying, 'Мария'::character varying, 'Авдеева'::character
varying, 'Лермонтова 8'::character varying)
 returning id_passport;

INSERT INTO hotel.client (
id_passport, phone_number_client, family_name, first_name, last_name, adress_client)
VALUES (
'12345675812'::character varying, '89812678910'::bigint, 'Викторович'::character
varying, 'Артем'::character varying, 'Соломахин'::character varying, 'Гоголя
12'::character varying)
 returning id_passport;
```

/*Заполнение таблицы hotel рабочими данными */

```
INSERT INTO hotel.hotel (
name_hotel, adress) VALUES (
'Миссия'::character varying, 'Биржевая линия 16'::character varying)
 returning id_hotel;
```

```
INSERT INTO hotel.hotel (
name_hotel, adress) VALUES (
'Мариот'::character varying, 'Кронверский 49'::character varying)
 returning id_hotel;

INSERT INTO hotel.hotel (
name_hotel, adress) VALUES (
'Ангара'::character varying, 'Ломоносова 9'::character varying)
 returning id_hotel;
```

/*Заполнение таблицы price рабочими данными */

```
INSERT INTO hotel.price (
price_for_one_day, type_room, end_date_price, start_date_price) VALUES (
'2000'::bigint, 'двухместный'::character varying, '2022-03-21'::date, '2022-04-10'::date)
 returning id_price;

INSERT INTO hotel.price (
price_for_one_day, type_room, end_date_price, start_date_price) VALUES (
'1300'::bigint, 'одноместный'::character varying, '2022-03-21'::date, '2022-04-10'::date)
 returning id_price;

INSERT INTO hotel.price (
price_for_one_day, type_room, end_date_price, start_date_price) VALUES (
'1900'::bigint, 'двухместный'::character varying, '2022-03-01'::date, '2022-03-20'::date)
 returning id_price;

INSERT INTO hotel.price (
price_for_one_day, type_room, end_date_price, start_date_price) VALUES (
'1200'::bigint, 'одноместный'::character varying, '2022-03-01'::date, '2022-03-20'::date)
 returning id_price;
```

/*Заполнение таблицы room рабочими данными*/

```
INSERT INTO hotel.room (
room_number, type_room, status_busyness, id_hotel) VALUES (
'3'::integer, 'двухместный'::character varying, 'занят'::character varying, '4'::integer)
 returning room_number;

INSERT INTO hotel.room (
room_number, type_room, status_busyness, id_hotel) VALUES (
'2'::integer, 'двухместный'::character varying, 'свободен'::character varying,
'3'::integer)
 returning room_number;

INSERT INTO hotel.room (
room_number, type_room, status_busyness, id_hotel) VALUES (
'1'::integer, 'одноместный'::character varying, 'занят'::character varying, '2'::integer)
```

```
 returning room_number;
```

/*Заполнение таблицы sale рабочими данными*/

```
INSERT INTO hotel.sale (
start_date, end_date, description, type_room) VALUES (
'2022-02-20'::date, '2022-02-28'::date, '20% скидка, если взять больше 5
ночей'::character varying, 'двухместный'::character varying)
 returning start_date;
```

/*Заполнение таблицы type_room рабочими данными */

```
INSERT INTO hotel.type_room (
type_room, amount_places) VALUES (
'двухместный'::character varying, '2'::integer)
 returning type_room;

INSERT INTO hotel.type_room (
type_room, amount_places) VALUES (
'одноместный'::character varying, '1'::integer)
 returning type_room;
```

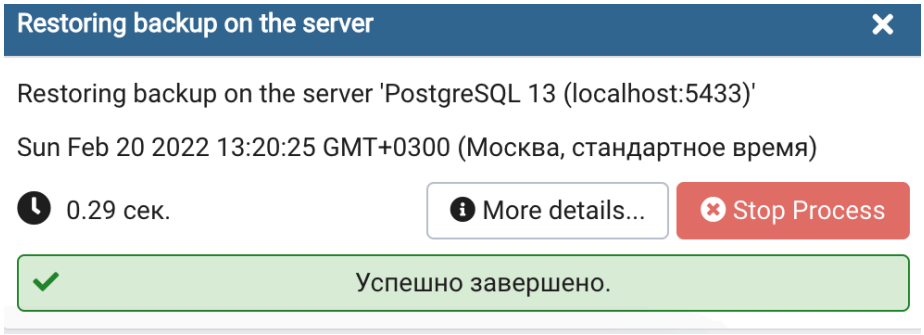
/*Заполнение таблицы worker рабочими данными */

```
INSERT INTO hotel.worker (
phone_w, full_name_w) VALUES (
'89812432574'::bigint, 'Кошман Александр Александрович'::character varying)
 returning id_worker;

INSERT INTO hotel.worker (
phone_w, full_name_w) VALUES (
'89812432573'::bigint, 'Путкин Владимир Владимирович'::character varying)
 returning id_worker;

INSERT INTO hotel.worker (
phone_w, full_name_w) VALUES (
'89812432542'::bigint, 'Иванов Иван Иванович'::character varying)
 returning id_worker;
```

/*Восстановление базы данных*/



Вывод:

В ходе выполнения работы была создана база данных в PostgreSQL, созданы таблицы и ограничения на значение столбцов, в базу данных были занесены рабочие данные, а также была создана логическая модель базы данных и dump.