

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 БД PostgreSQL в pgAdmin
по дисциплине: Проектирование и реализация баз данных

Специальность:

09.03.03 Мобильные и сетевые технологии

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверил:

Говорова М.М. _____

Дата: «__» _____ 20__ г.

Оценка _____

Выполнил:

студент группы К3240

Чернов Е. К.

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

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

Вариант 1. БД «Отель»

Описание предметной области: Отели 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).

ВЫПОЛНЕНИЕ

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию)
2. Создать схему в составе базы данных
3. Создать таблицы базы данных

Была создана база данных “Hotel” согласно варианту 1. Создана схема и созданы таблицы (hotel, room, room_type, promotions, price, registration, roomer, worker):

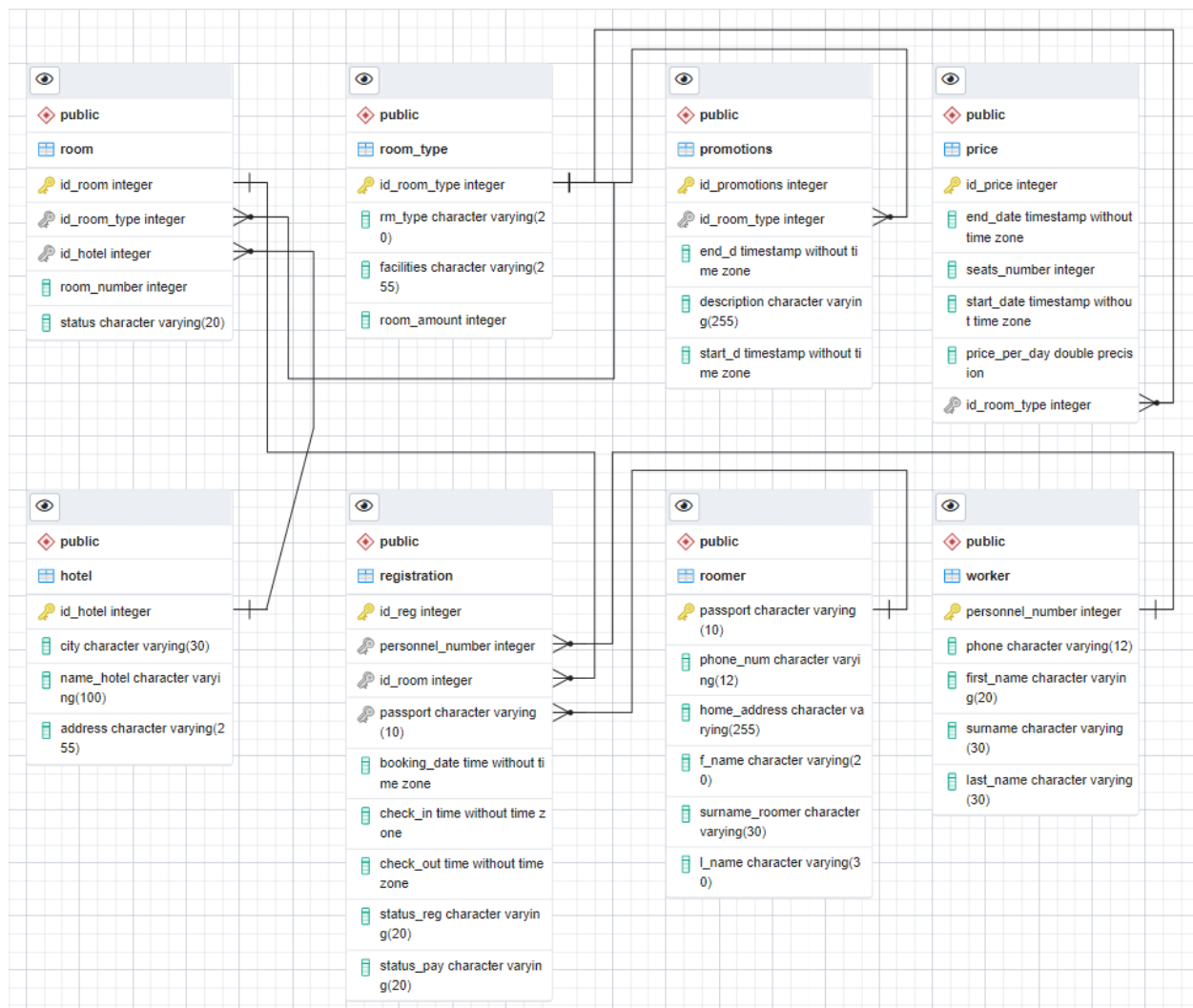


Рисунок 1 - Схема логической модели базы данных, сгенерированная в Generate ERD

Для таблиц room_type, promotions и price были добавлены суррогатные ключи (соответственно: id_room_type, id_promotions, id_price).

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

Ограничения для таблицы hotel:

- id_hotel integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- city character varying(30) COLLATE pg_catalog."default" NOT NULL,
- name_hotel character varying(100) COLLATE pg_catalog."default" NOT NULL,
- address character varying(255) COLLATE pg_catalog."default" NOT NULL,
- CONSTRAINT hotel_pkey PRIMARY KEY (id_hotel)

Ограничения для таблицы room:

- id_room integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- id_room_type integer NOT NULL,
- id_hotel integer NOT NULL,
- room_number integer NOT NULL,
- status character varying(20) COLLATE pg_catalog."default" NOT NULL,
- CONSTRAINT room_pkey PRIMARY KEY (id_room),
- CONSTRAINT room_id_hotel_fkey FOREIGN KEY (id_hotel) REFERENCES public.hotel (id_hotel) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
- CONSTRAINT room_id_room_type_fkey FOREIGN KEY (id_room_type) REFERENCES public.room_type (id_room_type) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
- CONSTRAINT room_status_check CHECK (status::text = 'свободен'::text OR status::text = 'занят'::text OR status::text = 'забронирован'::text),

- CONSTRAINT room_room_number_check CHECK (room_number > 0)

Ограничения для таблицы room_type:

- id_room_type integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- rm_type character varying(20) COLLATE pg_catalog."default" NOT NULL,
- facilities character varying(255) COLLATE pg_catalog."default",
- room_amount integer NOT NULL,
- CONSTRAINT room_type_pkey PRIMARY KEY (id_room_type),
- CONSTRAINT room_type_rm_type_check CHECK (rm_type::text = 'econom'::text OR rm_type::text = 'standart'::text OR rm_type::text = 'vip'::text),
- CONSTRAINT room_type_room_amount_check CHECK (room_amount = 0 OR room_amount = 1)

Ограничения для таблицы promotions:

- id_promotions integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- id_room_type integer NOT NULL,
- end_d timestamp without time zone NOT NULL,
- description character varying(255) COLLATE pg_catalog."default",
- start_d timestamp without time zone NOT NULL,
- CONSTRAINT promotions_pkey PRIMARY KEY (id_promotions),
- CONSTRAINT promotions_id_room_type_fkey FOREIGN KEY (id_room_type) REFERENCES public.room_type (id_room_type) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
- CONSTRAINT promotions_check CHECK (start_d < end_d)

Ограничения для таблицы price:

- id_price integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- end_date timestamp without time zone NOT NULL,
- seats_number integer NOT NULL,
- start_date timestamp without time zone NOT NULL,
- price_per_day double precision NOT NULL,
- id_room_type integer NOT NULL,
- CONSTRAINT price_pkey PRIMARY KEY (id_price),
- CONSTRAINT price_id_room_type_fkey FOREIGN KEY (id_room_type) REFERENCES public.room_type (id_room_type) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION NOT VALID,
- CONSTRAINT price_seats_number_check CHECK (seats_number >= 0),
- CONSTRAINT price_check CHECK (start_date < end_date),
- CONSTRAINT price_price_per_day_check CHECK (price_per_day > 0::double precision)

Ограничения для таблицы registration:

- id_reg integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- personnel_number integer NOT NULL,
- id_room integer NOT NULL,
- passport character varying(10) COLLATE pg_catalog."default" NOT NULL,
- status_reg character varying(20) COLLATE pg_catalog."default" NOT NULL,
- status_pay character varying(20) COLLATE pg_catalog."default" NOT NULL,
- booking_date timestamp without time zone NOT NULL,

- check_in timestamp without time zone NOT NULL,
- check_out timestamp without time zone NOT NULL,
- CONSTRAINT registration_pkey PRIMARY KEY (id_reg),
- CONSTRAINT registration_id_room_fkey FOREIGN KEY (id_room)
REFERENCES public.room (id_room) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
- CONSTRAINT registration_passport_fkey FOREIGN KEY (passport)
REFERENCES public.roomer (passport) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
- CONSTRAINT registration_personnel_number_fkey FOREIGN KEY
(personnel_number)
REFERENCES public.worker (personnel_number) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
- CONSTRAINT registration_status_reg_check CHECK (status_reg::text =
'зарегистрирован'::text OR status_reg::text = 'не зарегистрирован'::text),
- CONSTRAINT registration_check CHECK (booking_date <= check_in),
- CONSTRAINT registration_check1 CHECK (check_in < check_out)

Ограничения для таблицы roomer:

- passport character varying(10) COLLATE pg_catalog."default" NOT NULL,
- phone_num character varying(12) COLLATE pg_catalog."default" NOT NULL,
- home_address character varying(255) COLLATE pg_catalog."default" NOT NULL,
- f_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
- surname_roomer character varying(30) COLLATE pg_catalog."default" NOT NULL,
- l_name character varying(30) COLLATE pg_catalog."default",
- CONSTRAINT roomer_pkey PRIMARY KEY (passport)

Ограничения для таблицы worker:

- personnel_number integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
- phone character varying(12) COLLATE pg_catalog."default" NOT NULL,
- first_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
- surname character varying(30) COLLATE pg_catalog."default" NOT NULL,
- last_name character varying(30) COLLATE pg_catalog."default",
- CONSTRAINT worker_pkey PRIMARY KEY (personnel_number)

5. Заполнить таблицы БД рабочими данными.

Данные для таблицы hotel:

- INSERT INTO hotel (city, name_hotel, address)
VALUES ('Saint-Petersburg', 'Four Seasons', 'Вознесенский пр., 1');
- INSERT INTO hotel (city, name_hotel, address)
VALUES ('Saint-Petersburg', 'Индиго', 'ул. Чайковского, 17');
- INSERT INTO hotel (city, name_hotel, address)
VALUES ('Moscow', 'Four Seasons', 'ул. Охотный Ряд, д. 2');
- INSERT INTO hotel (city, name_hotel, address)
VALUES ('Moscow', 'Radisson Collection', '2/1 Kutuzovskiy Avenue Bld. 1');

Данные для таблицы room:

- INSERT INTO room (id_room_type, id_hotel, room_number, status)
VALUES (1, 2, 1, 'свободен');
- INSERT INTO room (id_room_type, id_hotel, room_number, status)

VALUES (2, 3, 22, 'свободен');

- INSERT INTO room (id_room_type, id_hotel, room_number, status)
VALUES (3, 4, 111, 'занят');
- INSERT INTO room (id_room_type, id_hotel, room_number, status)
VALUES (4, 5, 222, 'забронирован');

Данные для таблицы room_type:

- INSERT INTO room_type (rm_type, facilities, room_amount)
VALUES ('econom', 'Телевизор', 1);
- INSERT INTO room_type (rm_type, facilities, room_amount)
VALUES ('standart', 'Телевизор, интернет', 1);
- INSERT INTO room_type (rm_type, facilities, room_amount)
VALUES ('vip', 'Телевизор, интернет, джакузи', 0);
- INSERT INTO room_type (rm_type, facilities, room_amount)
VALUES ('vip', 'Телевизор, интернет, джакузи', 0);

Данные для таблицы promotions:

- INSERT INTO promotions (id_room_type, start_d, end_d, description)
VALUES (1, '08.03.2022', '11.03.2022', 'скидка 15%');
- INSERT INTO promotions (id_room_type, start_d, end_d, description)
VALUES (2, '08.03.2022', '10.03.2022', 'скидка 10%');
- INSERT INTO promotions (id_room_type, start_d, end_d, description)
VALUES (3, '08.03.2022', '09.03.2022', 'скидка 5%');
- INSERT INTO promotions (id_room_type, start_d, end_d, description)
VALUES (4, '08.03.2022', '09.03.2022', 'скидка 5%');

Данные для таблицы price:

- INSERT INTO price (id_room_type, start_date, end_date, seats_number, price_per_day)
VALUES (3, '07.03.2022', '11.03.2022', 1, 13500.5);
- INSERT INTO price (id_room_type, start_date, end_date, seats_number, price_per_day)
VALUES (4, '06.03.2022', '10.03.2022', 1, 12500.5);

Данные для таблицы registration:

- INSERT INTO registration (id_room, passport, personnel_number, booking_date, check_in, check_out, status_reg, status_pay)
VALUES (3, '1028763678', 1, '06.03.2022', '07.03.2022', '11.03.2022', 'зарегистрирован', 'оплачено');
- INSERT INTO registration (id_room, passport, personnel_number, booking_date, check_in, check_out, status_reg, status_pay)
VALUES (4, '1023963678', 2, '05.03.2022', '06.03.2022', '10.03.2022', 'зарегистрирован', 'не оплачено');

Данные для таблицы roomer:

- INSERT INTO roomer (passport, phone_num, home_address, f_name, surname_roomer, l_name)
VALUES ('1028763678', '+79112195365', 'ул. Пушкина, д. 5, кв. 26', 'Kirill', 'Ivanov', 'Stepanovich');
- INSERT INTO roomer (passport, phone_num, home_address, f_name, surname_roomer)
VALUES ('1023963678', '+79316573521', 'ул. Шнапса, д. 53, кв. 126', 'Stepan', 'Sidorov');
- INSERT INTO roomer (passport, phone_num, home_address, f_name, surname_roomer, l_name)
VALUES ('1029263678', '+79817345781', 'ул. Симонова, д. 53, кв. 262', 'Roman', 'Petrov', 'Egorovich');

- INSERT INTO roomer (passport, phone_num, home_address, f_name, surname_roomer)
VALUES ('1024063678', '+79528628123', 'ул. Крылова, д. 5, кв. 262', 'Gena', 'Chernov');

Данные для таблицы worker:

- INSERT INTO worker (phone, first_name, surname, last_name)
VALUES ('+79112168365', 'Kirill', 'Ivanov', 'Stepanovich');
- INSERT INTO worker (phone, first_name, surname)
VALUES ('+79316586521', 'Stepan', 'Sidorov');
- INSERT INTO worker (phone, first_name, surname, last_name)
VALUES ('+79817396781', 'Roman', 'Petrov', 'Egorovich');
- INSERT INTO worker (phone, first_name, surname)
VALUES ('+79528679123', 'Gena', 'Chernov');

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;

Резервная копия с расширением CUSTOM:

- Создаю резервную копию базы данных “Hotel” с расширением CUSTOM, определяю папку, где будет храниться копия и называю файл “hotel.backup”;
- Устанавливаю сжатие 5;
- Выбираю кодировку UTF-8;
- Устанавливаю роль postgres;

Резервная копия с расширением PLAIN:

- Создаю резервную копию базы данных “Hotel” с расширением PLAIN, определяю папку, где будет храниться копия и называю файл “hotel2.backup”;
- Устанавливаю сжатие 5;
- Выбираю кодировку UTF-8;
- Устанавливаю роль postgres;

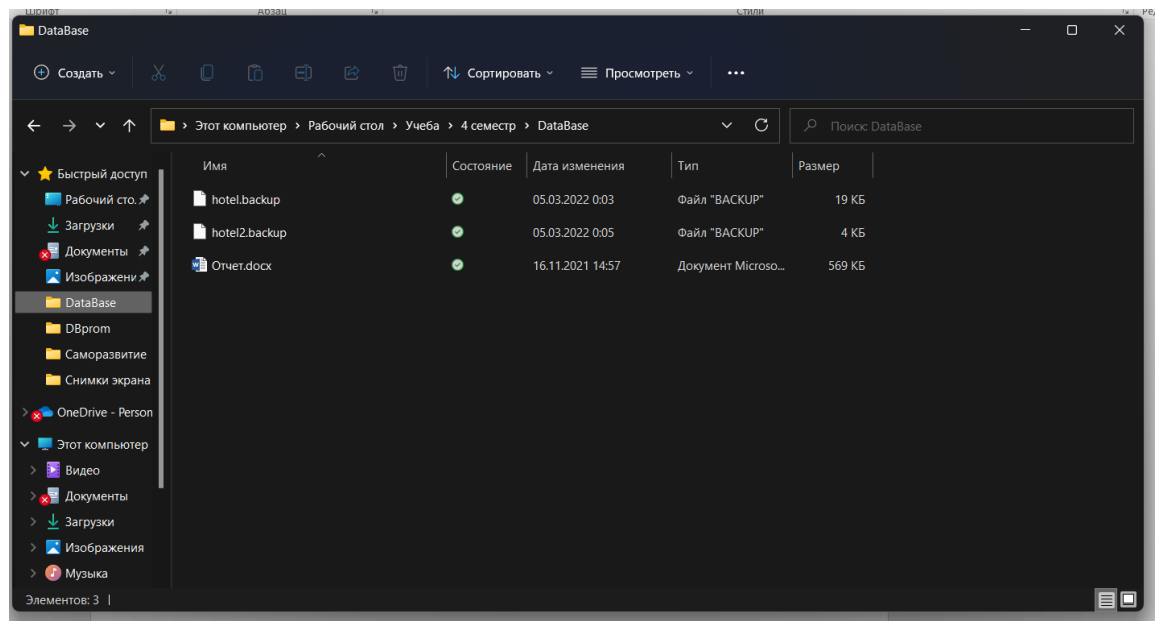


Рисунок 2 – Резервные копии базы данных

7. Восстановить БД.

Создаю пустую базу данных с названием “hotelCopy”, далее применяю к ней функцию Restore.

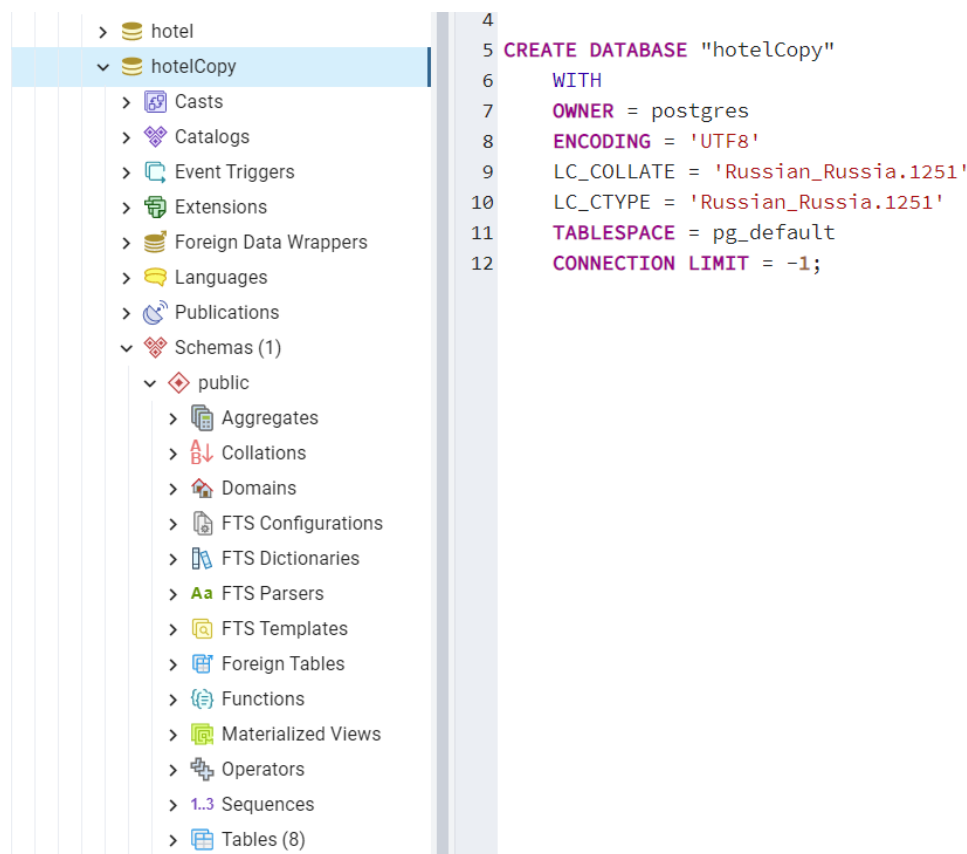


Рисунок 3 – Восстановленная копия базы данных

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. Также освоено резервное копирование и восстановление базы данных.