Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа №4

Реализация SQL-запросов для создания базы данных

Студент: Губаревич А.В.

Преподаватель: Силич С.С.

МИНСК 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc160389953)

[1 СОЗДАНИЕ ТАБЛИЦ 5](#_Toc160389954)

[2 НАПОЛНЕНИЕ ТАБЛИЦ 5](#_Toc160389955)

[3 ДРУГИЕ SQL-КОМАНДЫ 16](#_Toc160389956)

[4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ 17](#_Toc160389957)

[5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ 18](#_Toc160389958)

[6 ЭКСПОРТ РЕЗУЛЬТАТОВ 19](#_Toc160389959)

[ЗАКЛЮЧЕНИЕ 20](#_Toc160389960)

# ВВЕДЕНИЕ

В ходе данной работы, требуется выполнить:

1. Создать в СУБД новую схему данных для хранения пользовательских объектов (см. часть 2).
2. В этой новой схеме данных с помощью скрипта с запросами на языке DDL SQL реализовать таблицы, соответствующие реляционным отношениям схемы данных полученной в лабораторной работе №2-3, с помощью одного (желательно) оператора CREATE TABLE для каждой таблицы в следующем порядке:
   * реализовать простую структуру таблиц, включающую только набор столбцов с добавлением описаний первичного ключа;
   * дополнить описание таблицы реализацией ограничений для описания внешних ключей; для внешних ключей установить свойства контроля целостности данных (каскадное удаление и обновление), если это возможно в целевой СУБД;
   * дополнить описание таблицы реализацией ограничений для описания бизнес-правил;
   * дополнить описание таблицы реализацией комментариев для значимых элементов таблицы.
3. Заполнить с помощью SQL-скрипта с использованием оператора INSERT таблицы строками данных для проверки правильного выбора первичных ключей и работоспособности ссылок между таблицами:
   * строками данных сначала заполнять мастер-таблицы (или таблицы, которые НЕ ссылаются на другие таблицы);
   * в каждую таблицу добавить N строк осмысленных данных (N – количество строк, выдаваемое преподавателем);
   * если не удается добавить данные в таблицу по причине нарушения уникальности первичного ключа, то следует перепроверить описание этого первичного ключа и его смысл для реального мира;
   * если не удается добавить данные в таблицу по причине нарушения ссылочной целостности, то следует убедиться, что целевые данные существуют, иначе перепроверить описание внешнего ключа.
4. Рассмотреть простые действия по изменению структуры таблицы (переименование столбца таблицы, добавление и удаление ограничений на столбец таблицы или всю таблицу) и реализовать их с помощью оператора ALTER TABLE.
5. Создать временную таблицу с помощью оператора CREATE TABLE и удалить ее с помощью оператора DROP TABLE.
6. Экспортировать результаты работы в SQL-скрипт (см. часть 2), сравнить полученный скрипт со скриптами, созданными на этапах 2 и 3.

# СОЗДАНИЕ ТАБЛИЦ

Создать в СУБД новую схему данных для хранения пользовательских

объектов.

CREATE TABLE IF NOT EXISTS public."Aviary"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

square numeric NOT NULL,

status character varying(15) NOT NULL,

type character varying(30) NOT NULL,

location character varying(30) NOT NULL

);

CREATE TABLE IF NOT EXISTS public."Adoptive\_parent"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

snp character varying(50) NOT NULL,

telephone character varying(20) NOT NULL,

approval\_status character varying(15) NOT NULL,

address character varying(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS public."Employee"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

snp character varying(50) NOT NULL,

telephone character varying(20) NOT NULL,

date\_of\_hiring date NOT NULL,

post character varying(30) NOT NULL

);

CREATE TABLE IF NOT EXISTS public."Medical\_procedure"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

cost integer NOT NULL,

amount integer NOT NULL,

reason character varying(50) NOT NULL, name character varying(30) NOT NULL

);

CREATE TABLE IF NOT EXISTS public."Feed\_supply"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

quantity integer NOT NULL,

delivery\_date date NOT NULL,

the\_supplier character varying(30) NOT NULL,

type\_of\_feed character varying(30) NOT NULL);

CREATE TABLE IF NOT EXISTS public."Animal"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

snp character varying(50) NOT NULL,

telephone character varying(20) NOT NULL,

email character varying(100) NOT NULL,

duty character varying(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS public."Adoptive\_animal"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

id\_adoptive bigint NOT NULL REFERENCES public."Adoptive\_parent"(id),

id\_animal bigint NOT NULL REFERENCES public."Animal"(id),

UNIQUE (id\_adoptive, id\_animal)

);

CREATE TABLE IF NOT EXISTS public."Animal\_feed\_supply"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

id\_animal bigint NOT NULL REFERENCES public."Animal"(id),

id\_feed\_supply bigint NOT NULL REFERENCES public."Feed\_supply"(id),

UNIQUE (id\_animal, id\_feed\_supply));

# 2 НАПОЛНЕНИЕ ТАБЛИЦ

Для заполнения таблиц информацией используется директива *INSERT*.

1. Таблица Aviary:

INSERT INTO public."Aviary"(square, status, type, location) VALUES

(25.5, 'Active', 'Outdoor', 'North Wing'),

(15.0, 'Maintenance', 'Indoor', 'South Wing'),

(30.0, 'Active', 'Outdoor', 'East Wing'),

(12.5, 'Cleaning', 'Indoor', 'West Wing'),

(20.0, 'Active', 'Mixed', 'Central Area'),

(18.0, 'Quarantine', 'Isolation', 'Medical Wing'),

(22.0, 'Active', 'Outdoor', 'Garden Area'),

(14.0, 'Renovation', 'Indoor', 'North Wing'),

(28.0, 'Active', 'Outdoor', 'Playground'),

(16.5, 'Active', 'Indoor', 'South Wing');

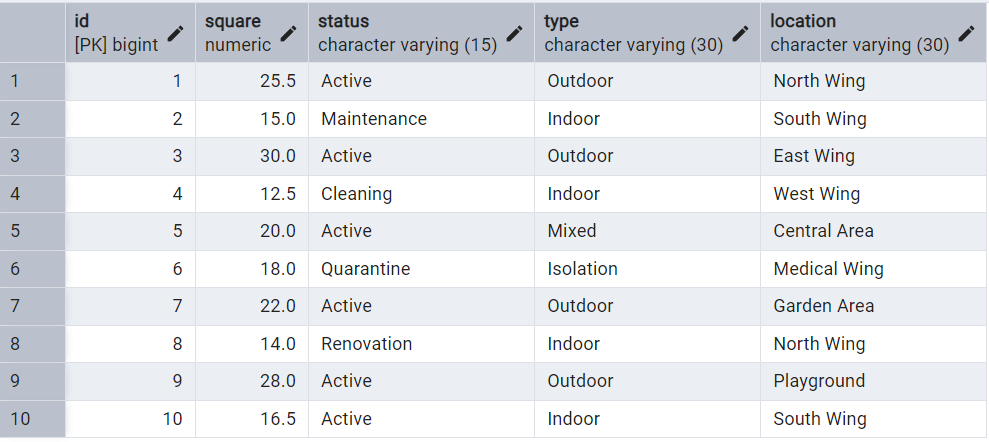


Рисунок 2.1 – Строки таблицы «Aviary»

1. Таблица Adoptive\_parent:

INSERT INTO public."Adoptive\_parent"(snp, telephone, approval\_status, address) VALUES

('Иванов Иван Иванович', '+79161234567', 'Approved', 'ул. Ленина, д. 10, кв. 5'),

('Петрова Мария Сергеевна', '+79169876543', 'Pending', 'ул. Пушкина, д. 25, кв. 12'),

('Сидоров Алексей Петрович', '+79167778899', 'Approved', 'пр. Мира, д. 15, кв. 8'),

('Кузнецова Елена Викторовна', '+79165554433', 'Rejected', 'ул. Садовая, д. 7, кв. 3'),

('Николаев Дмитрий Олегович', '+79162223344', 'Approved', 'ул. Центральная, д. 30, кв. 15'),

('Орлова Анна Михайловна', '+79163332211', 'Pending', 'пр. Победы, д. 42, кв. 9'),

('Васнецов Сергей Александрович', '+79164445566', 'Approved', 'ул. Лесная, д. 18, кв. 6'),

('Зайцева Ольга Дмитриевна', '+79168889900', 'Approved', 'ул. Школьная, д. 5, кв. 11'),

('Федоров Максим Игоревич', '+79169990011', 'Pending', 'пр. Строителей, д. 22, кв. 4'),

('Смирнова Татьяна Владимировна', '+79161112233', 'Approved', 'ул. Молодежная, д. 14, кв. 7');



Рисунок 2.2 – Строки таблицы «Adoptive\_parent»

1. Таблица Employee:

INSERT INTO public."Employee"(snp, telephone, date\_of\_hiring, post) VALUES

('Волков Андрей Николаевич', '+79160001122', '2020-03-15', 'Ветеринар'),

('Лебедева Ирина Петровна', '+79161113344', '2019-06-20', 'Зоотехник'),

('Соколов Павел Михайлович', '+79162224455', '2021-01-10', 'Смотритель'),

('Громова Екатерина Александровна', '+79163335566', '2018-09-05', 'Администратор'),

('Белов Денис Сергеевич', '+79164446677', '2022-02-28', 'Волонтер'),

('Романова Наталья Игоревна', '+79165557788', '2020-11-12', 'Ветеринар'),

('Крылов Артем Викторович', '+79166668899', '2019-04-18', 'Смотритель'),

('Давыдова Марина Олеговна', '+79167779900', '2021-07-23', 'Зоотехник'),

('Ершов Виктор Дмитриевич', '+79168880011', '2018-12-30', 'Администратор'),

('Тихонова Светлана Алексеевна', '+79169991122', '2022-05-14', 'Волонтер');

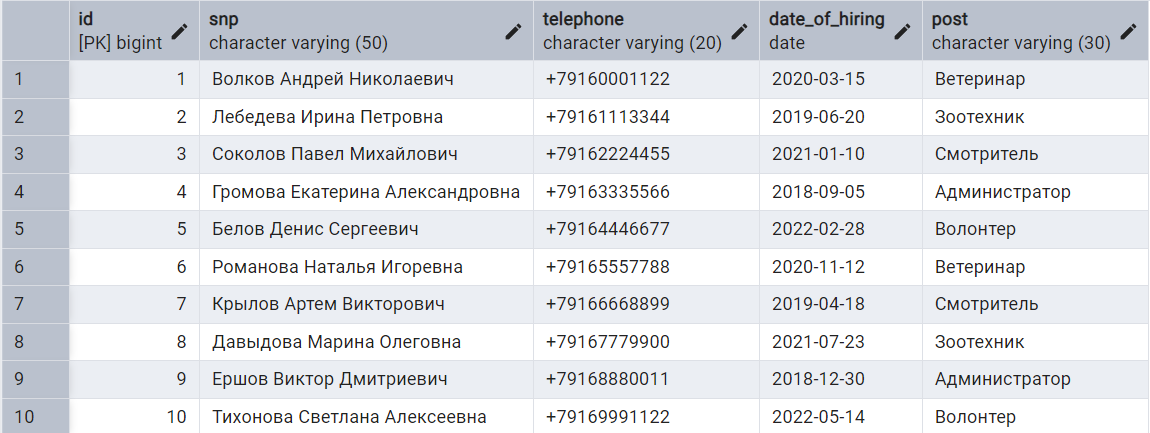


Рисунок 2.3 – Строки таблицы «Employee»

1. Таблица Medical\_procedure:

INSERT INTO public."Medical\_procedure"(cost, amount, reason, name) VALUES

(1500, 1, 'Ежегодный осмотр', 'Общий осмотр'),

(3000, 1, 'Вакцинация', 'Прививка от бешенства'),

(2500, 1, 'Стерилизация', 'Кастрация'),

(2800, 1, 'Стерилизация', 'Стерилизация'),

(1200, 2, 'Обработка от паразитов', 'Обработка от блох'),

(5000, 1, 'Хирургическая операция', 'Перелом лапы'),

(800, 1, 'Диагностика', 'Анализ крови'),

(2000, 1, 'Лечение', 'Лечение ушной инфекции'),

(3500, 1, 'Стоматология', 'Чистка зубов'),

(1800, 1, 'Диагностика', 'УЗИ брюшной полости');



Рисунок 2.4 – Строки таблицы «Medical\_procedure»

1. Таблица Feed\_supply:

INSERT INTO public."Feed\_supply"(quantity, delivery\_date, the\_supplier, type\_of\_feed) VALUES

(100, '2024-01-15', 'ЗооМир', 'Сухой корм для собак'),

(50, '2024-01-16', 'ПетШоп', 'Влажный корм для кошек'),

(75, '2024-01-17', 'АгроКорм', 'Зерновой корм для птиц'),

(30, '2024-01-18', 'ЗооЛюкс', 'Корм для грызунов'),

(60, '2024-01-19', 'Фермер', 'Сено для кроликов'),

(40, '2024-01-20', 'АкваМир', 'Корм для рыб'),

(90, '2024-01-21', 'ЗооМир', 'Сухой корм для кошек'),

(25, '2024-01-22', 'ПетШоп', 'Лакомства для собак'),

(35, '2024-01-23', 'АгроКорм', 'Корм для рептилий'),

(55, '2024-01-24', 'ЗооЛюкс', 'Влажный корм для собак');

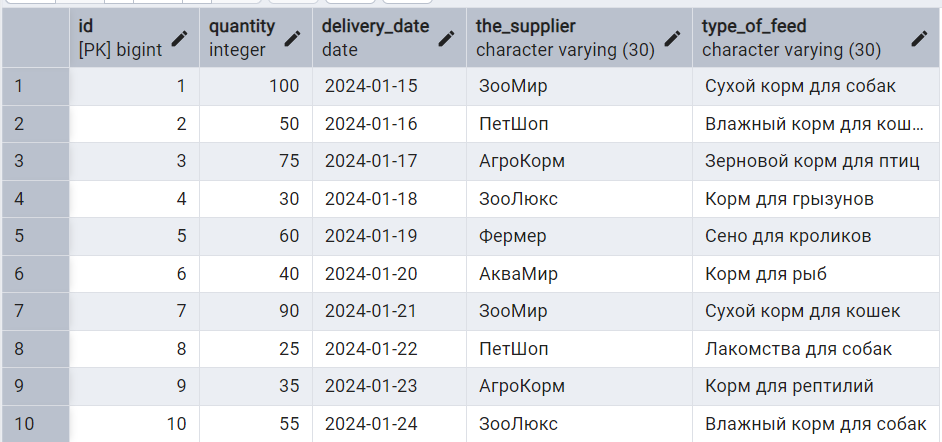


Рисунок 2.5 – Строки таблицы «Feed\_supply»

1. Таблица Animal:

INSERT INTO public."Animal"(breed, date\_of\_receipt, state\_of\_health, type, id\_aviary, id\_employee) VALUES

('Британская короткошерстная', '2024-01-15', 'Здоров', 'Кошка', 1, 1),

('Немецкая овчарка', '2024-01-16', 'Здоров', 'Собака', 2, 2),

('Сиамская', '2024-01-17', 'Лечение', 'Кошка', 3, 3),

('Дворняга', '2024-01-18', 'Здоров', 'Собака', 4, 4),

('Волнистый попугай', '2024-01-19', 'Здоров', 'Птица', 5, 5),

('Хомяк сирийский', '2024-01-20', 'Здоров', 'Грызун', 6, 6),

('Мейн-кун', '2024-01-21', 'Карантин', 'Кошка', 7, 7),

('Лабрадор', '2024-01-22', 'Здоров', 'Собака', 8, 8),

('Неразлучники', '2024-01-23', 'Здоров', 'Птица', 9, 9),

('Шиншилла', '2024-01-24', 'Лечение', 'Грызун', 10, 10);

Рисунок 2.6 – Строки таблицы «Animal»

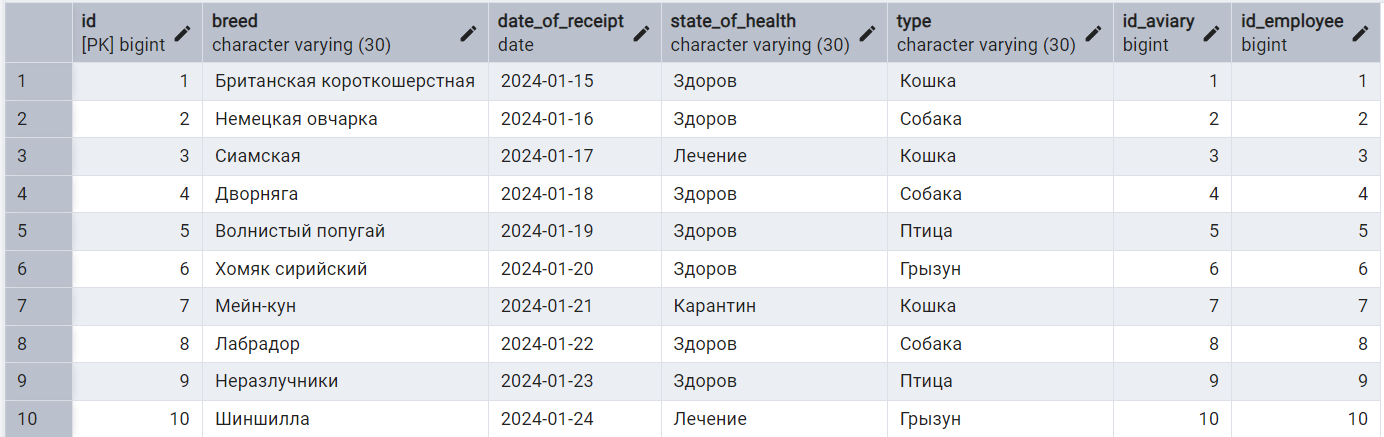


Рисунок 2.6 – Строки таблицы «Animal»

1. Таблица Volunteer:

INSERT INTO public."Volunteer"(snp, telephone, email, duty) VALUES

('Петров Алексей Иванович', '+79161234567', 'petrov@example.com', 'Выгул собак'),

('Сидорова Мария Петровна', '+79169876543', 'sidorova@example.com', 'Уход за кошками'),

('Кузнецов Дмитрий Сергеевич', '+79167778899', 'kuznetsov@example.com', 'Кормление животных'),

('Николаева Елена Викторовна', '+79165554433', 'nikolaeva@example.com', 'Уборка вольеров'),

('Васнецов Игорь Александрович', '+79162223344', 'vasnecov@example.com', 'Социализация животных'),

('Орлова Анна Михайловна', '+79163332211', 'orlova@example.com', 'Помощь ветеринару'),

('Лебедев Сергей Николаевич', '+79164445566', 'lebedev@example.com', 'Транспортировка'),

('Зайцева Ольга Дмитриевна', '+79168889900', 'zayceva@example.com', 'Обучение волонтеров'),

('Федоров Максим Игоревич', '+79169990011', 'fedorov@example.com', 'Фандрайзинг'),

('Смирнова Татьяна Владимировна', '+79161112233', 'smirnova@example.com', 'Работа с документами');

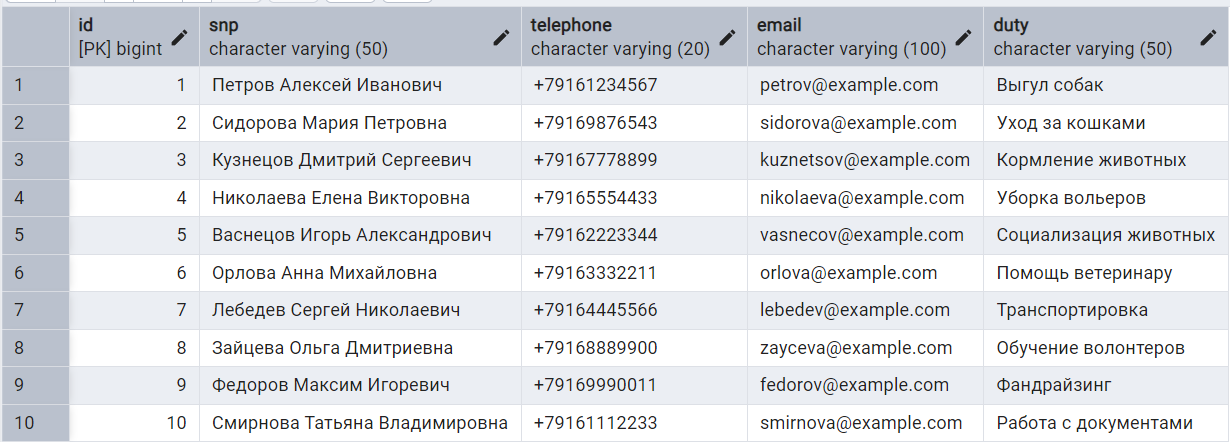


Рисунок 2.7 – Строки таблицы «Volunteer»

1. Таблица Adoptive\_animal:

INSERT INTO public."Adoptive\_animal"(id\_adoptive, id\_animal) VALUES

(1, 3),

(1, 5),

(2, 7),

(3, 1),

(3, 2),

(4, 8),

(5, 4),

(5, 6),

(5, 9),

(6, 10);

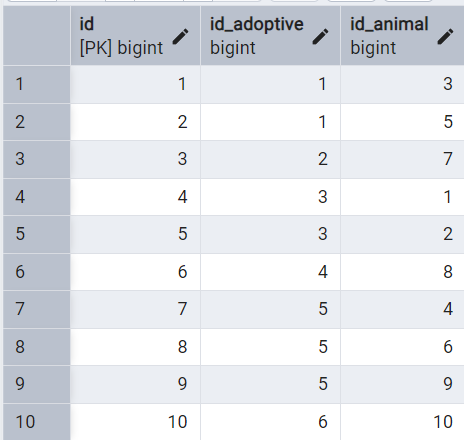


Рисунок 2.8 – Строки таблицы «Adoptive\_animal»

1. Таблица Animal\_feed\_supply:

INSERT INTO public."Animal\_feed\_supply"(id\_animal, id\_feed\_supply) VALUES

(1, 2),

(2, 1),

(3, 3),

(4, 1),

(5, 4),

(6, 5),

(7, 2),

(8, 6),

(9, 7),

(10, 8);

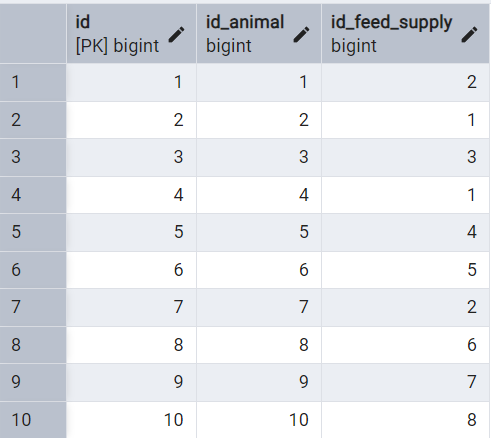


Рисунок 2.9 – Строки таблицы «Adoptive\_feed\_supply»

1. Таблица Animal\_volunteer:

INSERT INTO public."Animal\_volunteer"(id\_volunteer, id\_animal) VALUES

(1, 1),

(1, 2),

(2, 3),

(2, 4),

(3, 5),

(3, 6),

(4, 7),

(4, 8),

(5, 9),

(5, 10);

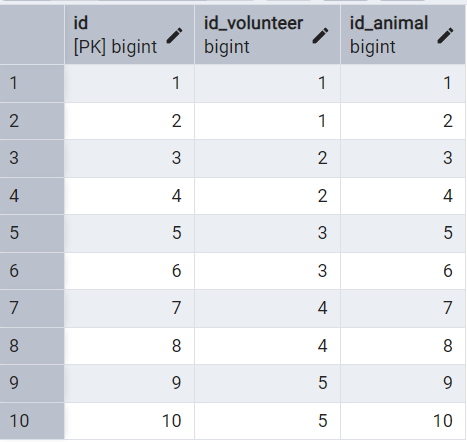


Рисунок 2.10 – Строки таблицы « Animal\_volunteer»

11. Таблица Animal\_medical\_procedure:

INSERT INTO public."Animal\_medical\_procedure"(id\_animal, id\_medical) VALUES

(1, 1),

(1, 2),

(2, 1),

(2, 5),

(3, 3),

(4, 1),

(4, 2),

(5, 4),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10);

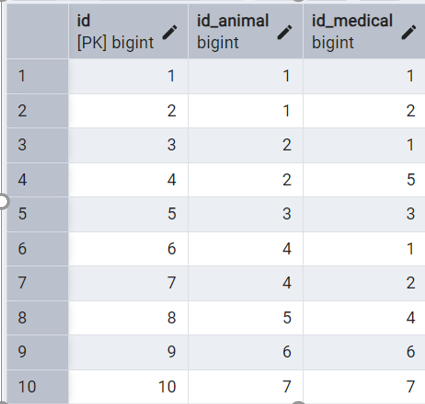


Рисунок 2.11 – Строки таблицы « Animal\_medical\_procedure »

# 3 ДРУГИЕ SQL-КОМАНДЫ

Рассмотрим простые действия по изменению структуры таблиц с помощью оператора *ALTER TABLE:*

1. Переименование столбца:

ALTER TABLE public."Employee" RENAME COLUMN date\_of\_hiring TO hire\_date;

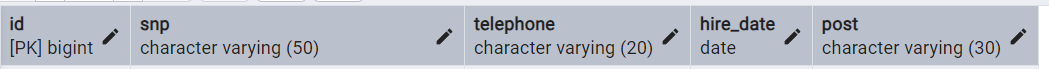


Рисунок 3.1 – Переименованный столбец в таблице «Employee»

2. Добавление новых столбцов:

ALTER TABLE public."Medical\_procedure" ADD COLUMN description TEXT;



Рисунок 3.2 – Добавленный столбец в таблице «Medical\_procedure»

3. Удаление столбцов:

ALTER TABLE public."Medical\_procedure" DROP COLUMN IF EXISTS description;



Рисунок 3.3 – Удаленный столбец в таблице «Medical\_procedure»

4. Добавление ограничений:

Ограничение CHECK для положительной стоимости.

ALTER TABLE public."Medical\_procedure"

ADD CONSTRAINT chk\_positive\_cost CHECK (cost > 0);

5. Удаление ограничений

ALTER TABLE public."Medical\_procedure"

DROP CONSTRAINT IF EXISTS chk\_positive\_cost;

# 4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ

Создадим временную таблицу.

CREATE TEMP TABLE temp\_medical\_report AS

SELECT

a.breed,

a.type,

mp.name as procedure\_name,

mp.cost,

amp.procedure\_date

FROM public."Animal" a

JOIN public."Animal\_medical\_procedure" amp ON a.id = amp.id\_animal

JOIN public."Medical\_procedure" mp ON amp.id\_medical = mp.id;

Просмотр и удаление.

SELECT \* FROM temp\_medical\_report;

DROP TABLE temp\_medical\_report;

# 5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ

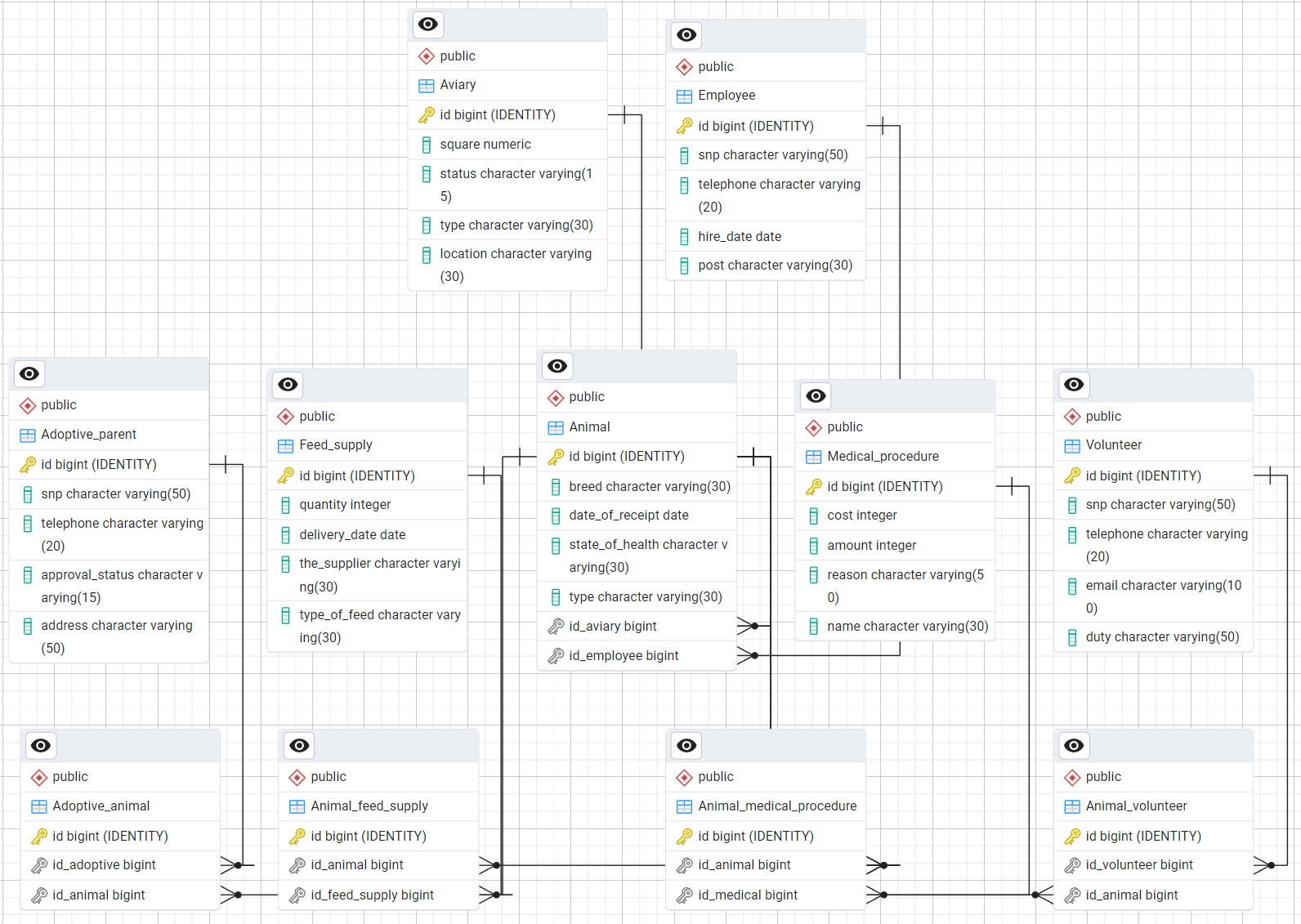


Рисунок 5.1 – Полученная EDR-диаграмма

# 6 ЭКСПОРТ РЕЗУЛЬТАТОВ

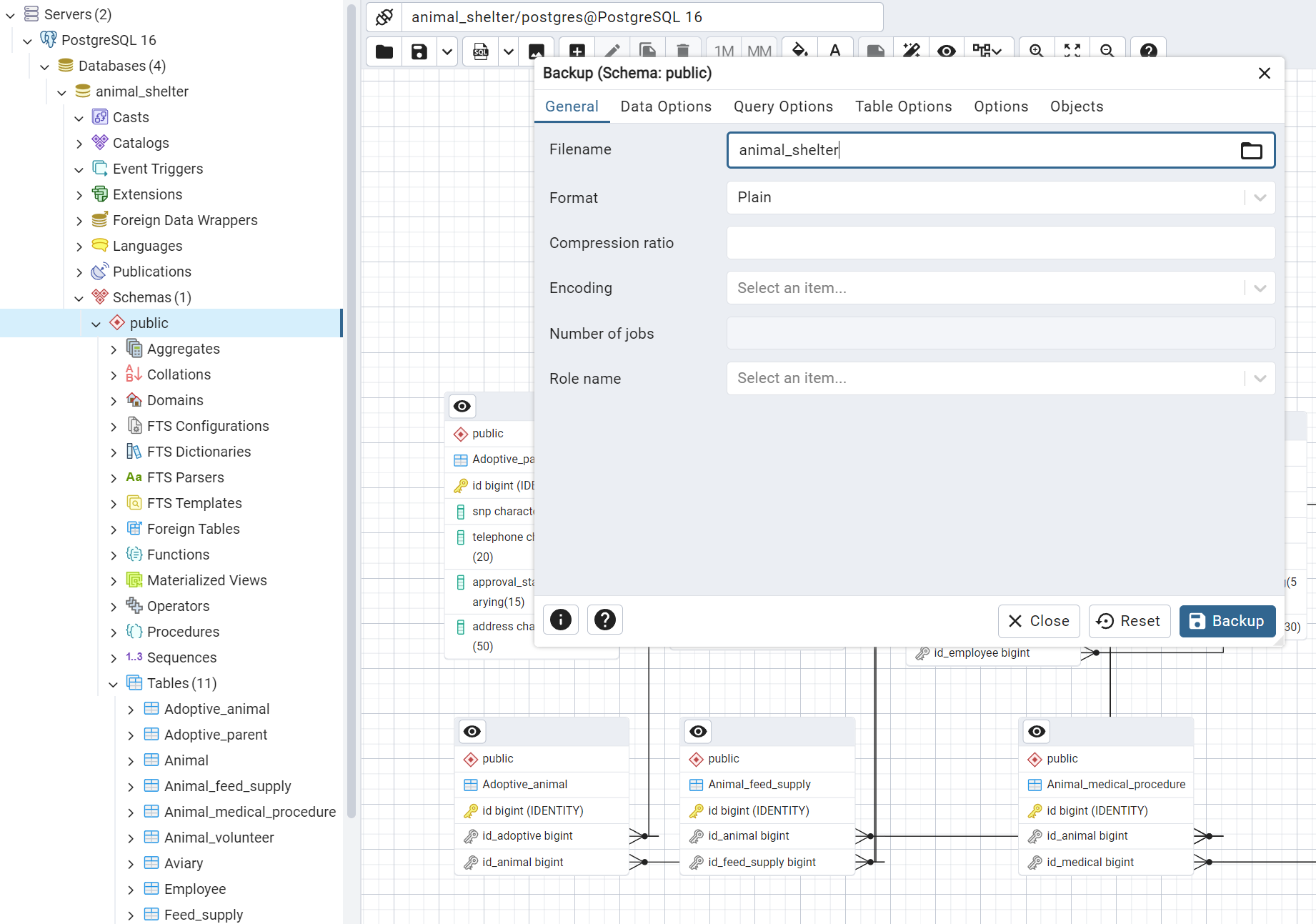


Рисунок 6.1 – Backup

# ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы освоено создание и управление базой данных в pgAdmin. Написаны SQL-запросы для создания таблиц, связей и ограничений. Проведено сравнение ERD-диаграммы с исходной моделью. Выполнен экспорт схемы в SQL-файл через Backup. Проанализированы различия между ручными скриптами и автоматически сгенерированными.