

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин

Лабораторная работа №4
Реализация SQL-запросов для создания базы данных

Студент:
Преподаватель:

Губаревич А.В.
Силич С.С.

МИНСК 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 СОЗДАНИЕ ТАБЛИЦ	5
2 НАПОЛНЕНИЕ ТАБЛИЦ	Ошибка! Закладка не определена.
3 ДРУГИЕ SQL-КОМАНДЫ	16
4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ.....	17
5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ	18
6 ЭКСПОРТ РЕЗУЛЬТАТОВ.....	19
ЗАКЛЮЧЕНИЕ	42

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 удастся 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 удастся 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.

1 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_s
upply"(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');
```

	id [PK] bigint	square numeric	status character varying (15)	type character varying (30)	location character varying (30)
1	1	25.5	Active	Outdoor	North Wing
2	2	15.0	Maintenance	Indoor	South Wing
3	3	30.0	Active	Outdoor	East Wing
4	4	12.5	Cleaning	Indoor	West Wing
5	5	20.0	Active	Mixed	Central Area
6	6	18.0	Quarantine	Isolation	Medical Wing
7	7	22.0	Active	Outdoor	Garden Area
8	8	14.0	Renovation	Indoor	North Wing
9	9	28.0	Active	Outdoor	Playground
10	10	16.5	Active	Indoor	South Wing

Рисунок 2.1 – Строки таблицы «Aviary»

2. 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');

```

	id [PK] bigint	snp character varying (50)	telephone character varying (20)	approval_status character varying (15)	address character varying (50)
1	1	Иванов Иван Иванович	+79161234567	Approved	ул. Ленина, д. 10, кв. 5
2	2	Петрова Мария Сергеевна	+79169876543	Pending	ул. Пушкина, д. 25, кв. 12
3	3	Сидоров Алексей Петрович	+79167778899	Approved	пр. Мира, д. 15, кв. 8
4	4	Кузнецова Елена Викторовна	+79165554433	Rejected	ул. Садовая, д. 7, кв. 3
5	5	Николаев Дмитрий Олегович	+79162223344	Approved	ул. Центральная, д. 30, кв. 15
6	6	Орлова Анна Михайловна	+79163332211	Pending	пр. Победы, д. 42, кв. 9
7	7	Васнецов Сергей Александрович	+79164445566	Approved	ул. Лесная, д. 18, кв. 6
8	8	Зайцева Ольга Дмитриевна	+79168889900	Approved	ул. Школьная, д. 5, кв. 11
9	9	Федоров Максим Игоревич	+79169990011	Pending	пр. Строителей, д. 22, кв. 4
10	10	Смирнова Татьяна Владимировна	+79161112233	Approved	ул. Молодежная, д. 14, кв. 7

Рисунок 2.2 – Строки таблицы «Adoptive_parent»

3. 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');

```

	id [PK] bigint	snp character varying (50)	telephone character varying (20)	date_of_hiring date	post character varying (30)
1	1	Волков Андрей Николаевич	+79160001122	2020-03-15	Ветеринар
2	2	Лебедева Ирина Петровна	+79161113344	2019-06-20	Зоотехник
3	3	Соколов Павел Михайлович	+79162224455	2021-01-10	Смотритель
4	4	Громова Екатерина Александровна	+79163335566	2018-09-05	Администратор
5	5	Белов Денис Сергеевич	+79164446677	2022-02-28	Волонтер
6	6	Романова Наталья Игоревна	+79165557788	2020-11-12	Ветеринар
7	7	Крылов Артем Викторович	+79166668899	2019-04-18	Смотритель
8	8	Давыдова Марина Олеговна	+79167779900	2021-07-23	Зоотехник
9	9	Ершов Виктор Дмитриевич	+79168880011	2018-12-30	Администратор
10	10	Тихонова Светлана Алексеевна	+79169991122	2022-05-14	Волонтер

Рисунок 2.3 – Строки таблицы «Employee»

4. 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');
```

	id [PK] bigint	cost integer	amount integer	reason character varying (50)	name character varying (30)
1	1	1500	1	Ежегодный осмотр	Общий осмотр
2	2	3000	1	Вакцинация	Прививка от бешенства
3	3	2500	1	Стерилизация	Кастрация
4	4	2800	1	Стерилизация	Стерилизация
5	5	1200	2	Обработка от паразитов	Обработка от блох
6	6	5000	1	Хирургическая операция	Перелом лапы
7	7	800	1	Диагностика	Анализ крови
8	8	2000	1	Лечение	Лечение ушной инфекции
9	9	3500	1	Стоматология	Чистка зубов
10	10	1800	1	Диагностика	УЗИ брюшной полости

Рисунок 2.4 – Строки таблицы «Medical_procedure»

5. 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');
```

	id [PK] bigint	quantity integer	delivery_date date	the_supplier character varying (30)	type_of_feed character varying (30)
1	1	100	2024-01-15	ЗооМир	Сухой корм для собак
2	2	50	2024-01-16	ПетШоп	Влажный корм для кош...
3	3	75	2024-01-17	АгроКорм	Зерновой корм для птиц
4	4	30	2024-01-18	ЗооЛюкс	Корм для грызунов
5	5	60	2024-01-19	Фермер	Сено для кроликов
6	6	40	2024-01-20	АкваМир	Корм для рыб
7	7	90	2024-01-21	ЗооМир	Сухой корм для кошек
8	8	25	2024-01-22	ПетШоп	Лакомства для собак
9	9	35	2024-01-23	АгроКорм	Корм для рептилий
10	10	55	2024-01-24	ЗооЛюкс	Влажный корм для собак

Рисунок 2.5 – Строки таблицы «Feed_supply»

6. 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»

	id [PK] bigint	breed character varying (30)	date_of_receipt date	state_of_health character varying (30)	type character varying (30)	id_aviary bigint	id_employee bigint
1	1	Британская короткошерстная	2024-01-15	Здоров	Кошка	1	1
2	2	Немецкая овчарка	2024-01-16	Здоров	Собака	2	2
3	3	Сиамская	2024-01-17	Лечение	Кошка	3	3
4	4	Дворняга	2024-01-18	Здоров	Собака	4	4
5	5	Волнистый попугай	2024-01-19	Здоров	Птица	5	5
6	6	Хомяк сирийский	2024-01-20	Здоров	Грызун	6	6
7	7	Мейн-кун	2024-01-21	Карантин	Кошка	7	7
8	8	Лабрадор	2024-01-22	Здоров	Собака	8	8
9	9	Неразлучники	2024-01-23	Здоров	Птица	9	9
10	10	Шиншилла	2024-01-24	Лечение	Грызун	10	10

Рисунок 2.6 – Строки таблицы «Animal»

7. 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');
```

	id [PK] bigint	snp character varying (50)	telephone character varying (20)	email character varying (100)	duty character varying (50)
1	1	Петров Алексей Иванович	+79161234567	petrov@example.com	Выгул собак
2	2	Сидорова Мария Петровна	+79169876543	sidorova@example.com	Уход за кошками
3	3	Кузнецов Дмитрий Сергеевич	+79167778899	kuznetsov@example.com	Кормление животных
4	4	Николаева Елена Викторовна	+79165554433	nikolaeva@example.com	Уборка вольеров
5	5	Васнецов Игорь Александрович	+79162223344	vasnecov@example.com	Социализация животных
6	6	Орлова Анна Михайловна	+79163332211	orlova@example.com	Помощь ветеринару
7	7	Лебедев Сергей Николаевич	+79164445566	lebedev@example.com	Транспортировка
8	8	Зайцева Ольга Дмитриевна	+79168889900	zayceva@example.com	Обучение волонтеров
9	9	Федоров Максим Игоревич	+79169990011	fedorov@example.com	Фандрайзинг
10	10	Смирнова Татьяна Владимировна	+79161112233	smirnova@example.com	Работа с документами

Рисунок 2.7 – Строки таблицы «Volunteer»

Для случаев, когда требуется точно добавить или изменить данные, в pgAdmin реализована возможность ручного редактирования содержимого таблиц через визуальный редактор. Этот способ исключает необходимость написания SQL-кода и идеально подходит для разовых операций или работы с небольшими объемами информации.

Процесс ручного заполнения состоит из нескольких интуитивно понятных шагов:

Шаг 1: Первым делом необходимо подключиться в pgAdmin к нужному серверу и развернуть в левой панели навигации дерево объектов до целевой таблицы, следуя по пути: Schemas -> public -> Tables.

Шаг 2: Чтобы открыть таблицу для просмотра и редактирования, следует кликнуть на ней правой кнопкой мыши и в контекстном меню выбрать опцию View/Edit Data -> All Rows.

Шаг 3: Для создания новой записи в открывшемся окне-редакторе нужно нажать на кнопку с изображением плюса («+») или ссылку «Добавить строку». В нижней части интерфейса появится форма для ввода, где можно заполнить значения для каждого поля будущей записи.

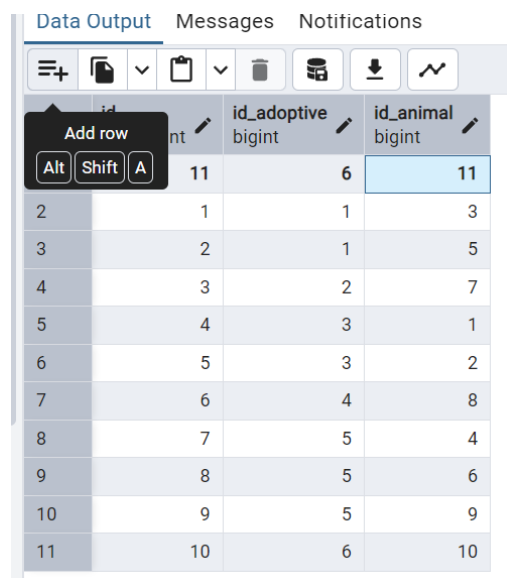


Рисунок 2.8 – Окно добавления новой записи в таблицу

Шаг 4 (Фиксация): Важно помнить, что все внесенные изменения носят временный характер до момента их явного сохранения в БД. Чтобы записать новую строку окончательно, необходимо нажать кнопку «Сохранить изменения» (команда Commit) или использовать быструю клавишу Ctrl + Enter.

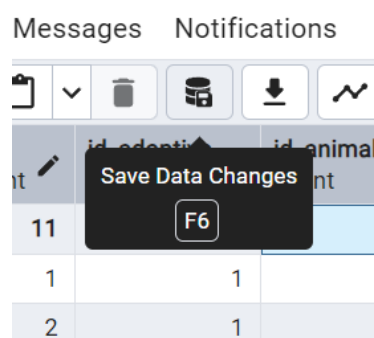


Рисунок 2.9 – Сохранение в базу данных

8. Таблица Adoptive_animal. Заполнение таблицы вручную.

	id [PK] bigint	id_adoptive bigint	id_animal bigint
1	1	1	3
2	2	1	5
3	3	2	7
4	4	3	1
5	5	3	2
6	6	4	8
7	7	5	4
8	8	5	6
9	9	5	9
10	10	6	10

Рисунок 2.10 – Строки таблицы «Adoptive_animal»

9. Таблица Animal_feed_supply. Заполнение таблицы вручную.

	id [PK] bigint	id_animal bigint	id_feed_supply bigint
1	1	1	2
2	2	2	1
3	3	3	3
4	4	4	1
5	5	5	4
6	6	6	5
7	7	7	2
8	8	8	6
9	9	9	7
10	10	10	8

Рисунок 2.11 – Строки таблицы «Adoptive_feed_supply»

10. Таблица Animal_volunteer. Заполнение таблицы вручную.

	id [PK] bigint	id_volunteer bigint	id_animal bigint
1	1	1	1
2	2	1	2
3	3	2	3
4	4	2	4
5	5	3	5
6	6	3	6
7	7	4	7
8	8	4	8
9	9	5	9
10	10	5	10

Рисунок 2.12 – Строки таблицы « Animal_volunteer»

11. Таблица Animal_medical_procedure. Заполнение таблицы вручную.

	id [PK] bigint	id_animal bigint	id_medical bigint
1	1	1	1
2	2	1	2
3	3	2	1
4	4	2	5
5	5	3	3
6	6	4	1
7	7	4	2
8	8	5	4
9	9	6	6
10	10	7	7

Рисунок 2.13 – Строки таблицы « Animal_medical_procedure »

3 ДРУГИЕ SQL-КОМАНДЫ

С помощью команды *ALTER TABLE* осуществляется переименование атрибутов, преобразование типов данных в столбцах, управление целостностью данных путем добавления или снятия ограничений, а также множество других операций, направленных на эволюцию реляционной модели. 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;
```

id [PK] bigint	snp character varying (50)	telephone character varying (20)	hire_date date	post character varying (30)
-------------------	-------------------------------	-------------------------------------	-------------------	--------------------------------

Рисунок 3.1 – Переименованный столбец в таблице «Employee»

2. Добавление новых столбцов:

```
ALTER TABLE public."Medical_procedure" ADD COLUMN descript  
ion TEXT;
```

id [PK] bigint	cost integer	amount integer	reason character varying (50)	name character varying (30)	description text
-------------------	-----------------	-------------------	----------------------------------	--------------------------------	---------------------

Рисунок 3.2 – Добавленный столбец в таблице «Medical_procedure»

3. Удаление столбцов:

```
ALTER TABLE public."Medical_procedure" DROP COLUMN IF EXISTS  
description;
```

id [PK] bigint	cost integer	amount integer	reason character varying (50)	name character varying (30)
-------------------	-----------------	-------------------	----------------------------------	--------------------------------

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

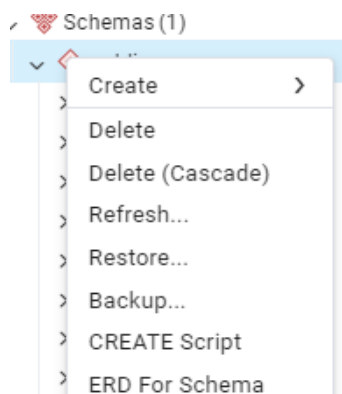


Рисунок 5.1 – Выгрузка диаграммы

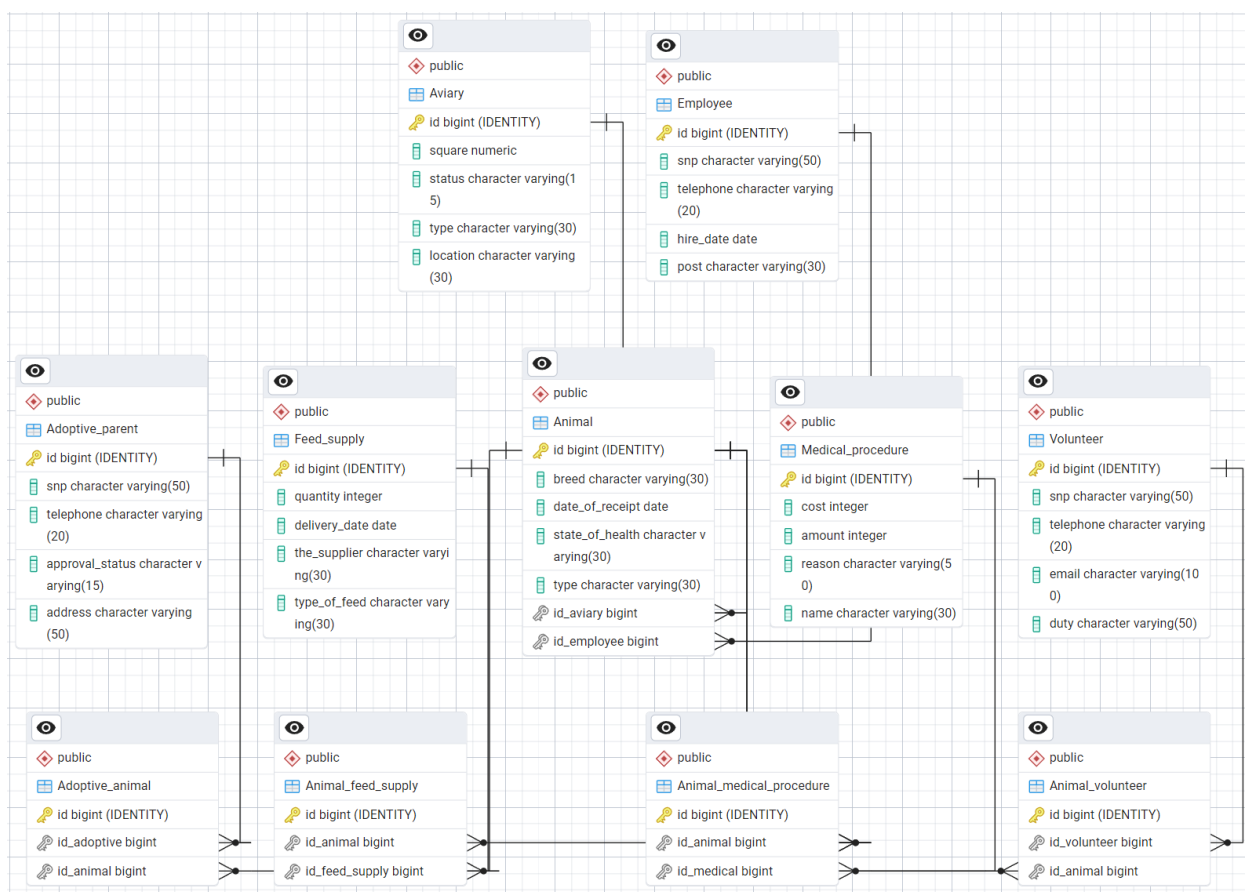


Рисунок 5.2 – Полученная EDR-диаграмма

6 ЭКСПОРТ РЕЗУЛЬТАТОВ

В целях обеспечения бесперебойной работы и минимизации рисков потери данных, рекомендуется регулярно создавать резервные копии структуры базы данных. Данная процедура необходима, например, для восстановления БД после сбоя или для развертывания ее точной копии на другом сервере.

Для выполнения этой задачи в среде управления PostgreSQL pgAdmin предусмотрен специальный механизм. Вам необходимо перейти к нужной базе данных, затем в контекстном меню выбрать пункт Schemas, а после — Backup. В открывшемся диалоговом окне требуется указать путь и имя для сохраняемого файла, а также в разделе «Format» обязательно выбрать вариант Plain. Это обеспечит создание SQL-скрипта, содержащего точные определения всех таблиц, индексов, последовательностей и других объектов схемы.

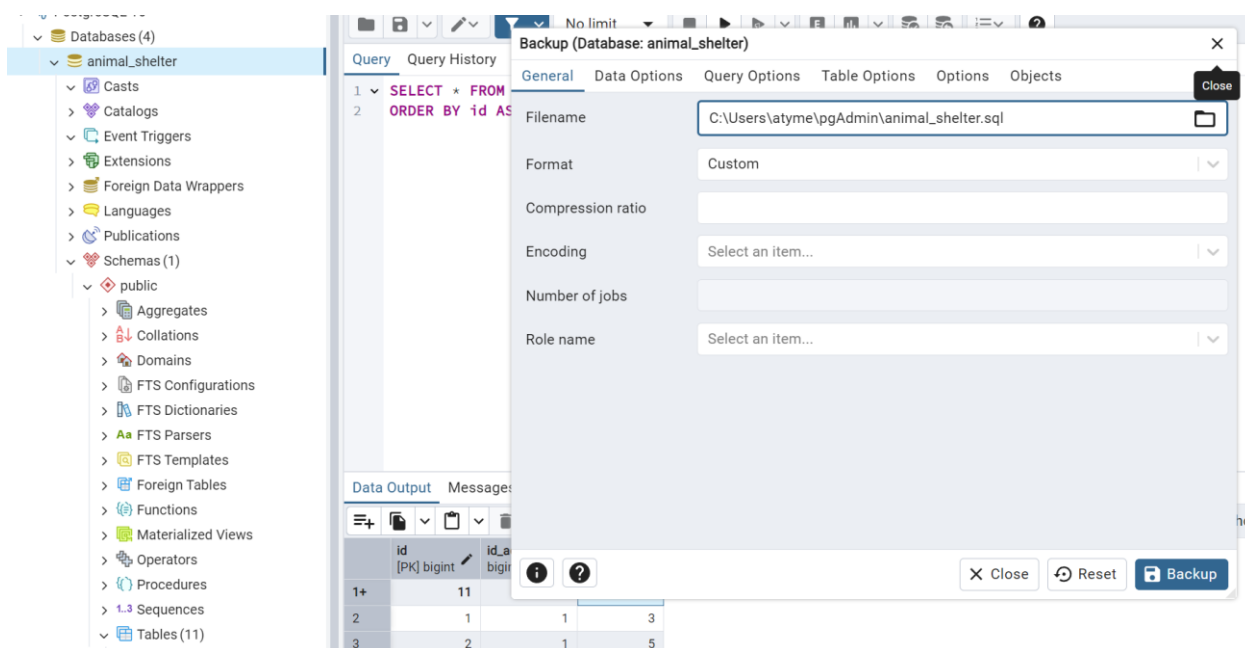


Рисунок 6.1 – Backup

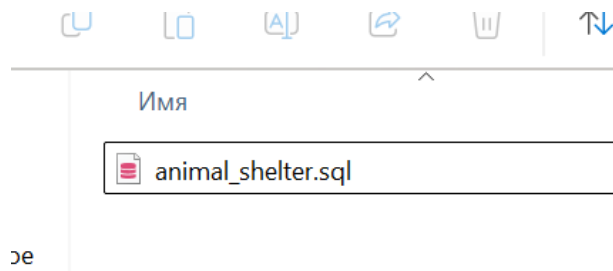


Рисунок 6.2 – резервный скрипт

Скрипт для восстановления базы данных, который создал Backup:

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

-- Dumped from database version 16.7
-- Dumped by pg_dump version 17.1

-- Started on 2025-10-17 16:41:08

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET transaction_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;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- TOC entry 228 (class 1259 OID 16438)
-- Name: Adoptive_animal; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Adoptive_animal" (
    id bigint NOT NULL,
    id_adoptive bigint NOT NULL,
    id_animal bigint NOT NULL
);
```

```

ALTER TABLE public."Adoptive_animal" OWNER TO postgres;

--
-- TOC entry 227 (class 1259 OID 16437)
-- Name: Adoptive_animal_id_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

ALTER TABLE public."Adoptive_animal" ALTER COLUMN id ADD
GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Adoptive_animal_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

--
-- TOC entry 220 (class 1259 OID 16408)
-- Name: Adoptive_parent; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Adoptive_parent" (
    id bigint NOT NULL,
    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
);

ALTER TABLE public."Adoptive_parent" OWNER TO postgres;

--
-- TOC entry 219 (class 1259 OID 16407)
-- Name: Adoptive_parent_id_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

ALTER TABLE public."Adoptive_parent" ALTER COLUMN id ADD
GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Adoptive_parent_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

```

```

);

--
-- TOC entry 232 (class 1259 OID 16474)
-- Name: Animal; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Animal" (
    id bigint NOT NULL,
    breed character varying(30) NOT NULL,
    date_of_receipt date NOT NULL,
    state_of_health character varying(30) NOT NULL,
    type character varying(30) NOT NULL,
    id_avairy bigint NOT NULL,
    id_employee bigint NOT NULL
);

ALTER TABLE public."Animal" OWNER TO postgres;

--
-- TOC entry 230 (class 1259 OID 16456)
-- Name: Animal_feed_supply; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Animal_feed_supply" (
    id bigint NOT NULL,
    id_animal bigint NOT NULL,
    id_feed_supply bigint NOT NULL
);

ALTER TABLE public."Animal_feed_supply" OWNER TO postgres;

--
-- TOC entry 229 (class 1259 OID 16455)
-- Name: Animal_feed_supply_id_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

ALTER TABLE public."Animal_feed_supply" ALTER COLUMN id ADD
GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Animal_feed_supply_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1

```

```

);

--
-- TOC entry 231 (class 1259 OID 16473)
-- Name: Animal_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public."Animal" ALTER COLUMN id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME public."Animal_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

--
-- TOC entry 238 (class 1259 OID 16514)
-- Name: Animal_medical_procedure; Type: TABLE; Schema: public;
Owner: postgres
--

CREATE TABLE public."Animal_medical_procedure" (
    id bigint NOT NULL,
    id_animal bigint NOT NULL,
    id_medical bigint NOT NULL
);

ALTER TABLE public."Animal_medical_procedure" OWNER TO postgres;

--
-- TOC entry 237 (class 1259 OID 16513)
-- Name: Animal_medical_procedure_id_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

ALTER TABLE public."Animal_medical_procedure" ALTER COLUMN id
ADD GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Animal_medical_procedure_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

```

```

--
-- TOC entry 236 (class 1259 OID 16496)
-- Name: Animal_volunteer; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Animal_volunteer" (
    id bigint NOT NULL,
    id_volunteer bigint NOT NULL,
    id_animal bigint NOT NULL
);

ALTER TABLE public."Animal_volunteer" OWNER TO postgres;

--
-- TOC entry 235 (class 1259 OID 16495)
-- Name: Animal_volunteer_id_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

ALTER TABLE public."Animal_volunteer" ALTER COLUMN id ADD
GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Animal_volunteer_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

--
-- TOC entry 218 (class 1259 OID 16400)
-- Name: Aviary; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Aviary" (
    id bigint NOT NULL,
    square numeric NOT NULL,
    status character varying(15) NOT NULL,
    type character varying(30) NOT NULL,
    location character varying(30) NOT NULL
);

ALTER TABLE public."Aviary" OWNER TO postgres;

```

```

--
-- TOC entry 217 (class 1259 OID 16399)
-- Name: Aviary_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public."Aviary" ALTER COLUMN id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME public."Aviary_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

--
-- TOC entry 222 (class 1259 OID 16414)
-- Name: Employee; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Employee" (
    id bigint NOT NULL,
    snp character varying(50) NOT NULL,
    telephone character varying(20) NOT NULL,
    hire_date date NOT NULL,
    post character varying(30) NOT NULL
);

ALTER TABLE public."Employee" OWNER TO postgres;

--
-- TOC entry 221 (class 1259 OID 16413)
-- Name: Employee_id_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

ALTER TABLE public."Employee" ALTER COLUMN id ADD GENERATED
ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Employee_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

```



```

--
-- TOC entry 226 (class 1259 OID 16426)
-- Name: Feed_supply; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Feed_supply" (
    id bigint NOT NULL,
    quantity integer NOT NULL,
    delivery_date date NOT NULL,
    the_supplier character varying(30) NOT NULL,
    type_of_feed character varying(30) NOT NULL
);

ALTER TABLE public."Feed_supply" OWNER TO postgres;

--
-- TOC entry 225 (class 1259 OID 16425)
-- Name: Feed_supply_id_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

ALTER TABLE public."Feed_supply" ALTER COLUMN id ADD GENERATED
ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Feed_supply_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

--
-- TOC entry 224 (class 1259 OID 16420)
-- Name: Medical_procedure; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Medical_procedure" (
    id bigint NOT NULL,
    cost integer NOT NULL,
    amount integer NOT NULL,
    reason character varying(50) NOT NULL,
    name character varying(30) NOT NULL,
    CONSTRAINT chk_positive_cost CHECK ((cost > 0))
);

```

```

ALTER TABLE public."Medical_procedure" OWNER TO postgres;

--
-- TOC entry 223 (class 1259 OID 16419)
-- Name: Medical_procedure_id_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

ALTER TABLE public."Medical_procedure" ALTER COLUMN id ADD
GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Medical_procedure_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

--
-- TOC entry 234 (class 1259 OID 16490)
-- Name: Volunteer; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Volunteer" (
    id bigint NOT NULL,
    snp character varying(50) NOT NULL,
    telephone character varying(20) NOT NULL,
    email character varying(100) NOT NULL,
    duty character varying(50) NOT NULL
);

ALTER TABLE public."Volunteer" OWNER TO postgres;

--
-- TOC entry 233 (class 1259 OID 16489)
-- Name: Volunteer_id_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

ALTER TABLE public."Volunteer" ALTER COLUMN id ADD GENERATED
ALWAYS AS IDENTITY (
    SEQUENCE NAME public."Volunteer_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

```

```
--
-- TOC entry 4934 (class 0 OID 16438)
-- Dependencies: 228
-- Data for Name: Adoptive_animal; Type: TABLE DATA; Schema:
public; Owner: postgres
--

COPY public."Adoptive_animal" (id, id_adoptive, id_animal) FROM
stdin;
1    1    3
2    1    5
3    2    7
4    3    1
5    3    2
6    4    8
7    5    4
8    5    6
9    5    9
10   6    10
\.
```

```
--
-- TOC entry 4926 (class 0 OID 16408)
-- Dependencies: 220
-- Data for Name: Adoptive_parent; Type: TABLE DATA; Schema:
public; Owner: postgres
--

COPY public."Adoptive_parent" (id, snp, telephone,
approval_status, address) FROM stdin;
1    Иванов Иван Иванович      +79161234567      Approved      ул.
Ленина, д. 10, кв. 5
2    Петрова Мария Сергеевна +79169876543      Pending ул. Пушкина,
д. 25, кв. 12
3    Сидоров Алексей Петрович  +79167778899      Approved      пр.
Мира, д. 15, кв. 8
4    Кузнецова Елена Викторовна +79165554433      Rejected      ул.
Садовая, д. 7, кв. 3
5    Николаев Дмитрий Олегович  +79162223344      Approved      ул.
Центральная, д. 30, кв. 15
6    Орлова Анна Михайловна     +79163332211      Pending пр. Победы,
д. 42, кв. 9
7    Васнецов Сергей
Александрович +79164445566      Approved      ул. Лесная, д. 18,
кв. 6
8    Зайцева Ольга Дмитриевна  +79168889900      Approved      ул.
Школьная, д. 5, кв. 11
```

```

9   Федоров Максим Игоревич +79169990011    Pending пр.
Строителей, д. 22, кв. 4
10  Смирнова Татьяна
Владимировна    +79161112233    Approved    ул. Молодежная, д.
14, кв. 7
\.
```

```

--
-- TOC entry 4938 (class 0 OID 16474)
-- Dependencies: 232
-- Data for Name: Animal; Type: TABLE DATA; Schema: public;
Owner: postgres
--
```

```

COPY public."Animal" (id, breed, date_of_receipt,
state_of_health, type, id_avairy, id_employee) FROM stdin;
1   Британская короткошерстная 2024-01-
15 Здоров Кошка 1 1
2   Немецкая овчарка 2024-01-16 Здоров Собака 2 2
3   Сиамская 2024-01-17 Лечение Кошка 3 3
4   Дворняга 2024-01-18 Здоров Собака 4 4
5   Волнистый попугай 2024-01-19 Здоров Птица 5 5
6   Хомяк сирийский 2024-01-20 Здоров Грызун 6 6
7   Мейн-кун 2024-01-21 Карантин Кошка 7 7
8   Лабрадор 2024-01-22 Здоров Собака 8 8
9   Неразлучники 2024-01-23 Здоров Птица 9 9
10  Шиншилла 2024-01-24 Лечение Грызун 10 10
\.
```

```

--
-- TOC entry 4936 (class 0 OID 16456)
-- Dependencies: 230
-- Data for Name: Animal_feed_supply; Type: TABLE DATA; Schema:
public; Owner: postgres
--
```

```

COPY public."Animal_feed_supply" (id, id_animal, id_feed_supply)
FROM stdin;
1   1 2
2   2 1
3   3 3
4   4 1
5   5 4
6   6 5
7   7 2
8   8 6
9   9 7
10  10 8
```

```
\.
```

```
--  
-- TOC entry 4944 (class 0 OID 16514)  
-- Dependencies: 238  
-- Data for Name: Animal_medical_procedure; Type: TABLE DATA;  
Schema: public; Owner: postgres  
--
```

```
COPY public."Animal_medical_procedure" (id, id_animal,  
id_medical) FROM stdin;
```

```
1 1 1  
2 1 2  
3 2 1  
4 2 5  
5 3 3  
6 4 1  
7 4 2  
8 5 4  
9 6 6  
10 7 7  
11 8 8  
12 9 9  
13 10 10
```

```
\.
```

```
--  
-- TOC entry 4942 (class 0 OID 16496)  
-- Dependencies: 236  
-- Data for Name: Animal_volunteer; Type: TABLE DATA; Schema:  
public; Owner: postgres  
--
```

```
COPY public."Animal_volunteer" (id, id_volunteer, id_animal)  
FROM stdin;
```

```
1 1 1  
2 1 2  
3 2 3  
4 2 4  
5 3 5  
6 3 6  
7 4 7  
8 4 8  
9 5 9  
10 5 10
```

```
\.
```

```
--
-- TOC entry 4924 (class 0 OID 16400)
-- Dependencies: 218
-- Data for Name: Aviary; Type: TABLE DATA; Schema: public;
Owner: postgres
--
```

```
COPY public."Aviary" (id, square, status, type, location) FROM
stdin;
```

```
1  25.5    Active  Outdoor North Wing
2  15.0    Maintenance Indoor  South Wing
3  30.0    Active  Outdoor East Wing
4  12.5    Cleaning   Indoor  West Wing
5  20.0    Active  Mixed    Central Area
6  18.0    Quarantine Isolation Medical Wing
7  22.0    Active  Outdoor Garden Area
8  14.0    Renovation Indoor  North Wing
9  28.0    Active  Outdoor Playground
10 16.5    Active  Indoor  South Wing
\.
```

```
--
-- TOC entry 4928 (class 0 OID 16414)
-- Dependencies: 222
-- Data for Name: Employee; Type: TABLE DATA; Schema: public;
Owner: postgres
--
```

```
COPY public."Employee" (id, snp, telephone, hire_date, post)
FROM stdin;
```

```
1  Волков Андрей Николаевич      +79160001122      2020-03-
15 Ветеринар
2  Лебедева Ирина Петровна +79161113344      2019-06-
20 Зоотехник
3  Соколов Павел Михайлович      +79162224455      2021-01-
10 Смотритель
4  Громова Екатерина Александровна +79163335566      2018-09-
05 Администратор
5  Белов Денис Сергеевич      +79164446677      2022-02-28 Волонтер
6  Романова Наталья Игоревна    +79165557788      2020-11-
12 Ветеринар
7  Крылов Артем Викторович +79166668899      2019-04-
18 Смотритель
8  Давыдова Марина Олеговна      +79167779900      2021-07-
23 Зоотехник
9  Ершов Виктор Дмитриевич +79168880011      2018-12-
30 Администратор
10 Тихонова Светлана Алексеевна    +79169991122      2022-05-
14 Волонтер
```

\.

```
--
-- TOC entry 4932 (class 0 OID 16426)
-- Dependencies: 226
-- Data for Name: Feed_supply; Type: TABLE DATA; Schema: public;
Owner: postgres
--
```

```
COPY public."Feed_supply" (id, quantity, delivery_date,
the_supplier, type_of_feed) FROM stdin;
```

```
1  100 2024-01-15  ЗооМир  Сухой корм для собак
2   50 2024-01-16  ПетШоп  Влажный корм для кошек
3   75 2024-01-17  АгроКорм  Зерновой корм для птиц
4   30 2024-01-18  ЗооЛюкс  Корм для грызунов
5   60 2024-01-19  Фермер  Сено для кроликов
6   40 2024-01-20  АкваМир  Корм для рыб
7   90 2024-01-21  ЗооМир  Сухой корм для кошек
8   25 2024-01-22  ПетШоп  Лакомства для собак
9   35 2024-01-23  АгроКорм  Корм для рептилий
10  55 2024-01-24  ЗооЛюкс  Влажный корм для собак
\.
```

```
--
-- TOC entry 4930 (class 0 OID 16420)
-- Dependencies: 224
-- Data for Name: Medical_procedure; Type: TABLE DATA; Schema:
public; Owner: postgres
--
```

```
COPY public."Medical_procedure" (id, cost, amount, reason, name)
FROM stdin;
```

```
1  1500      1  Ежегодный осмотр  Общий осмотр
2  3000      1  Вакцинация  Прививка от бешенства
3  2500      1  Стерилизация  Кастрация
4  2800      1  Стерилизация  Стерилизация
5  1200      2  Обработка от паразитов  Обработка от блох
6  5000      1  Хирургическая операция  Перелом лапы
7  800 1      Диагностика Анализ крови
8  2000      1  Лечение  Лечение ушной инфекции
9  3500      1  Стоматология  Чистка зубов
10 1800      1  Диагностика УЗИ брюшной полости
\.
```

```
--
-- TOC entry 4940 (class 0 OID 16490)
-- Dependencies: 234
```

```

-- Data for Name: Volunteer; Type: TABLE DATA; Schema: public;
Owner: postgres
--

COPY public."Volunteer" (id, snp, telephone, email, duty) FROM
stdin;
1    Петров Алексей Иванович
+79161234567    petrov@example.com    Выгул собак
2    Сидорова Мария Петровна
+79169876543    sidorova@example.com    Уход за кошками
3    Кузнецов Дмитрий
Сергеевич +79167778899    kuznetsov@example.com    Кормление
животных
4    Николаева Елена
Викторовна +79165554433    nikolaeva@example.com    Уборка
вольеров
5    Васнецов Игорь
Александрович +79162223344    vasnecov@example.com    Социали
зация животных
6    Орлова Анна
Михайловна +79163332211    orlova@example.com    Помощь
ветеринару
7    Лебедев Сергей
Николаевич +79164445566    lebedev@example.com    Транспортировка
8    Зайцева Ольга
Дмитриевна +79168889900    zayceva@example.com    Обучение
волонтеров
9    Федоров Максим Игоревич +79169990011    fedorov@example.com
Фандрайзинг
10   Смирнова Татьяна
Владимировна +79161112233    smirnova@example.com    Работа с
документами
\

--
-- TOC entry 4950 (class 0 OID 0)
-- Dependencies: 227
-- Name: Adoptive_animal_id_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Adoptive_animal_id_seq"', 10,
true);

--
-- TOC entry 4951 (class 0 OID 0)
-- Dependencies: 219

```



```

-- Name: Adoptive_parent_id_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Adoptive_parent_id_seq"', 10,
true);

--
-- TOC entry 4952 (class 0 OID 0)
-- Dependencies: 229
-- Name: Animal_feed_supply_id_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Animal_feed_supply_id_seq"',
10, true);

--
-- TOC entry 4953 (class 0 OID 0)
-- Dependencies: 231
-- Name: Animal_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public."Animal_id_seq"', 10, true);

--
-- TOC entry 4954 (class 0 OID 0)
-- Dependencies: 237
-- Name: Animal_medical_procedure_id_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Animal_medical_procedure_id_seq"',
13, true);

--
-- TOC entry 4955 (class 0 OID 0)
-- Dependencies: 235
-- Name: Animal_volunteer_id_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Animal_volunteer_id_seq"', 10,
true);

```

```

--
-- TOC entry 4956 (class 0 OID 0)
-- Dependencies: 217
-- Name: Aviary_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public."Aviary_id_seq"', 10, true);

--
-- TOC entry 4957 (class 0 OID 0)
-- Dependencies: 221
-- Name: Employee_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public."Employee_id_seq"', 10, true);

--
-- TOC entry 4958 (class 0 OID 0)
-- Dependencies: 225
-- Name: Feed_supply_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public."Feed_supply_id_seq"', 10,
true);

--
-- TOC entry 4959 (class 0 OID 0)
-- Dependencies: 223
-- Name: Medical_procedure_id_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Medical_procedure_id_seq"',
10, true);

--
-- TOC entry 4960 (class 0 OID 0)
-- Dependencies: 233
-- Name: Volunteer_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

```

```

SELECT pg_catalog.setval('public."Volunteer_id_seq"', 10, true);

--
-- TOC entry 4752 (class 2606 OID 16444)
-- Name: Adoptive_animal
Adoptive_animal_id_adoptiv_id_animal_key; Type: CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Adoptive_animal"
    ADD CONSTRAINT "Adoptive_animal_id_adoptiv_id_animal_key"
    UNIQUE (id_adoptiv, id_animal);

--
-- TOC entry 4754 (class 2606 OID 16442)
-- Name: Adoptive_animal Adoptive_animal_pkey; Type: CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Adoptive_animal"
    ADD CONSTRAINT "Adoptive_animal_pkey" PRIMARY KEY (id);

--
-- TOC entry 4744 (class 2606 OID 16412)
-- Name: Adoptive_parent Adoptive_parent_pkey; Type: CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Adoptive_parent"
    ADD CONSTRAINT "Adoptive_parent_pkey" PRIMARY KEY (id);

--
-- TOC entry 4756 (class 2606 OID 16462)
-- Name: Animal_feed_supply
Animal_feed_supply_id_animal_id_feed_supply_key; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_feed_supply"
    ADD CONSTRAINT
    "Animal_feed_supply_id_animal_id_feed_supply_key" UNIQUE
    (id_animal, id_feed_supply);

--

```

```

-- TOC entry 4758 (class 2606 OID 16460)
-- Name: Animal_feed_supply Animal_feed_supply_pkey; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_feed_supply"
    ADD CONSTRAINT "Animal_feed_supply_pkey" PRIMARY KEY (id);

--

-- TOC entry 4768 (class 2606 OID 16519)
-- Name: Animal_medical_procedure Animal_medical_procedure_pkey;
Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_medical_procedure"
    ADD CONSTRAINT "Animal_medical_procedure_pkey" PRIMARY KEY
(id);

--

-- TOC entry 4760 (class 2606 OID 16478)
-- Name: Animal Animal_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Animal"
    ADD CONSTRAINT "Animal_pkey" PRIMARY KEY (id);

--

-- TOC entry 4764 (class 2606 OID 16502)
-- Name: Animal_volunteer
Animal_volunteer_id_volunteer_id_animal_key; Type: CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_volunteer"
    ADD CONSTRAINT "Animal_volunteer_id_volunteer_id_animal_key"
UNIQUE (id_volunteer, id_animal);

--

-- TOC entry 4766 (class 2606 OID 16500)
-- Name: Animal_volunteer Animal_volunteer_pkey; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_volunteer"
    ADD CONSTRAINT "Animal_volunteer_pkey" PRIMARY KEY (id);

```

```

--
-- TOC entry 4742 (class 2606 OID 16406)
-- Name: Aviary Aviary_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Aviary"
    ADD CONSTRAINT "Aviary_pkey" PRIMARY KEY (id);

--
-- TOC entry 4746 (class 2606 OID 16418)
-- Name: Employee Employee_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Employee"
    ADD CONSTRAINT "Employee_pkey" PRIMARY KEY (id);

--
-- TOC entry 4750 (class 2606 OID 16430)
-- Name: Feed_supply Feed_supply_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Feed_supply"
    ADD CONSTRAINT "Feed_supply_pkey" PRIMARY KEY (id);

--
-- TOC entry 4748 (class 2606 OID 16424)
-- Name: Medical_procedure Medical_procedure_pkey; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Medical_procedure"
    ADD CONSTRAINT "Medical_procedure_pkey" PRIMARY KEY (id);

--
-- TOC entry 4762 (class 2606 OID 16494)
-- Name: Volunteer Volunteer_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Volunteer"
    ADD CONSTRAINT "Volunteer_pkey" PRIMARY KEY (id);

```

```

--
-- TOC entry 4769 (class 2606 OID 16445)
-- Name: Adoptive_animal Adoptive_animal_id_adoptive_fkey; Type:
FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Adoptive_animal"
    ADD CONSTRAINT "Adoptive_animal_id_adoptive_fkey" FOREIGN
KEY (id_adoptive) REFERENCES public."Adoptive_parent"(id);

--
-- TOC entry 4771 (class 2606 OID 16468)
-- Name: Animal_feed_supply
Animal_feed_supply_id_feed_supply_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_feed_supply"
    ADD CONSTRAINT "Animal_feed_supply_id_feed_supply_fkey"
FOREIGN KEY (id_feed_supply) REFERENCES
public."Feed_supply"(id);

--
-- TOC entry 4773 (class 2606 OID 16479)
-- Name: Animal Animal_id_aviary_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal"
    ADD CONSTRAINT "Animal_id_aviary_fkey" FOREIGN KEY
(id_aviary) REFERENCES public."Aviary"(id);

--
-- TOC entry 4774 (class 2606 OID 16484)
-- Name: Animal Animal_id_employee_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal"
    ADD CONSTRAINT "Animal_id_employee_fkey" FOREIGN KEY
(id_employee) REFERENCES public."Employee"(id);

--
-- TOC entry 4777 (class 2606 OID 16522)

```

```

-- Name: Animal_medical_procedure
Animal_medical_procedure_id_animal_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_medical_procedure"
    ADD CONSTRAINT "Animal_medical_procedure_id_animal_fkey"
FOREIGN KEY (id_animal) REFERENCES public."Animal"(id);

--

-- TOC entry 4778 (class 2606 OID 16527)
-- Name: Animal_medical_procedure
Animal_medical_procedure_id_medical_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_medical_procedure"
    ADD CONSTRAINT "Animal_medical_procedure_id_medical_fkey"
FOREIGN KEY (id_medical) REFERENCES
public."Medical_procedure"(id);

--

-- TOC entry 4775 (class 2606 OID 16508)
-- Name: Animal_volunteer Animal_volunteer_id_animal_fkey; Type:
FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_volunteer"
    ADD CONSTRAINT "Animal_volunteer_id_animal_fkey" FOREIGN KEY
(id_animal) REFERENCES public."Animal"(id);

--

-- TOC entry 4776 (class 2606 OID 16503)
-- Name: Animal_volunteer Animal_volunteer_id_volunteer_fkey;
Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_volunteer"
    ADD CONSTRAINT "Animal_volunteer_id_volunteer_fkey" FOREIGN
KEY (id_volunteer) REFERENCES public."Volunteer"(id);

--

-- TOC entry 4770 (class 2606 OID 16555)
-- Name: Adoptive_animal fk_adoptive_animal_animal; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

```

```

ALTER TABLE ONLY public."Adoptive_animal"
    ADD CONSTRAINT fk_adoptive_animal_animal FOREIGN KEY
(id_animal) REFERENCES public."Animal"(id);

--
-- TOC entry 4772 (class 2606 OID 16545)
-- Name: Animal_feed_supply fk_animal_feed_supply_animal; Type:
FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_feed_supply"
    ADD CONSTRAINT fk_animal_feed_supply_animal FOREIGN KEY
(id_animal) REFERENCES public."Animal"(id);

--
-- TOC entry 4779 (class 2606 OID 16550)
-- Name: Animal_medical_procedure
fk_animal_medical_procedure_animal; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Animal_medical_procedure"
    ADD CONSTRAINT fk_animal_medical_procedure_animal FOREIGN
KEY (id_animal) REFERENCES public."Animal"(id);

-- Completed on 2025-10-17 16:41:08

--
-- PostgreSQL database dump complete
--

```


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.