МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра САПР

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Распределенные базы данных и базы знаний»

Тема: Разработка баз данных

|  |  |  |
| --- | --- | --- |
| Студент гр. 0302 |  | Хаматов В.Р. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2022

ЗАДАНИЕ НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студент Хаматов В.Р.

Группа 0302

Тема работы: Проектирование базы данных

Исходные данные:

Разработать базу данных собак

Содержание пояснительной записки: «Содержание», «Введение», «Задание», «Краткое описание предметной области», «Проектирование базы данных», «Создание базы данных», «Создание таблиц», «Заполнение таблиц данными», «Разработка объектов промежуточного слоя», «Разработка стратегии резервного копирования», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

40 страниц

Дата выдачи задания: 10.10.2022

Дата сдачи задания:

Дата защиты задания:

|  |  |  |
| --- | --- | --- |
| Студент гр. 0302 |  | Хаматов В.Р. |
| Преподаватель |  | Новакова Н.Е. |

АННОТАЦИЯ

Целью данного индивидуального домашнего задания является проектирование и создание базы данных в рамках выбранной предметной области. Также частью проекта является разработка объектов промежуточного слоя (хранимые процедуры, UDF, представления). Наконец, в рамках работы предлагается стратегия резервного копирования базы данных.

SUMMARY

The purpose of this individual homework is a designing and creation a database within the chosen subject area. Second part of the project is the development of intermediate layer objects (stored procedures, UDFs, views). Final part of the work is suggesting a database backup strategy.

СОДЕРЖАНИЕ

[1 Введение 5](#_Toc122014172)

[2 Задание 5](#_Toc122014173)

[Цель работы: 5](#_Toc122014174)

[Задачи 5](#_Toc122014175)

[3 Краткое описание предметной области 5](#_Toc122014176)

[4 Проектирование базы данных 7](#_Toc122014177)

[5 Создание базы данных 8](#_Toc122014178)

[6 Создание таблиц 9](#_Toc122014179)

[7 Заполнение таблиц данными 17](#_Toc122014180)

[8 Разработка объектов промежуточного слоя 21](#_Toc122014181)

[8.1 Хранимые процедуры 21](#_Toc122014182)

[8.2 UDF 26](#_Toc122014183)

[8.3 Представления 33](#_Toc122014184)

[9 Разработка стратегии резервного копирования 35](#_Toc122014185)

[10 Заключение 39](#_Toc122014186)

[11 Список использованных источников 40](#_Toc122014187)

# Введение

В ходе выполнения данного индивидуального домашнего задания необходимо создать базу данных, предназначенную для отслеживания ухода за собаками. Для каждой собаки в БД хранятся сведения, о том, какой идентификатор у этой собаки, к какой клинике она закреплена, кличка, порода, возраст, идентификатор ветеринара, у которого собака обследуется, паспорт хозяина.

# Задание

### Цель работы:

Создать базу данных собак, предназначенную для отслеживания ухода за ними.

### Задачи

* Проектирование БД (структуры данных)
* Создание БД
* Создание таблиц и ограничений целостности
* Заполнение таблиц данными
* Разработка объектов промежуточного слоя (представлений, хранимых процедур, UDF-ов)
* Разработка стратегии резервного копирования

# Краткое описание предметной области

Предметная область в данной работе – база данных собак, с помощью которой можно будет легко узнать возраст, кличку, хозяина собаки, информацию о наличии прививок у конкретной собаки. База данных разрабатывается на основе следующей теоретической информации:

Собака - домашнее животное, одно из наиболее популярных (наряду с кошкой) животных-компаньонов.

Но помимо того, что собаки - верные друзья человека, при неправильном уходе за ними они могут наносить вред окружающим. Поэтому очень неплохо было бы собирать и анализировать информацию о них.

Вакцинация собак — проявление заботы не только о животных, но и о людях, контактирующих с ними. Вовремя сделанные прививки уберегут собаку от инфекций, паразитов, а также защитят хозяина от заражения некоторыми «собачьими» болезнями. Залог успешного и безболезненного проведения таких процедур — правильная подготовка питомца и соблюдение календаря.

Стандартный порядок прививок:

1. Первая прививка от инфекций (наиболее заразных заболеваний: от чумки, гепатита, энтеритов, парагриппа, лептоспироза.)
2. Вторая от инфекций - через 3 – 4 недели после первой прививки.
3. Прививка от бешенства – через 3 – 4 недели после первой прививки.
4. Третья прививка от инфекций - через 3 – 4 недели после второй.
5. Вторая прививка от бешенства – через 3 – 4 недели после второй прививки от бешенства.
6. Ежегодная — каждый год ревакцинация от всех болезней.

Реализация каждой задачи производилась с помощью SQL-запросов. Процесс разработки базы данных включал в себя создание и заполнение таблиц данными, написание хранимых процедур, пользовательских функций и представлений для удобной работы с базой данных.

Выполнение задания позволит лучше познакомиться с непосредственным процессом проектирования баз данных, а также закрепить на практике навыки, полученные в ходе основания теоретической и практической частей дисциплины «Базы данных».

# Проектирование базы данных

Исходя из предметной области выделим сущности в виде отдельных таблиц:

1. Veterinarian — ветеринар. Таблица содержит идентификтор ветеринара, название клиники, в которой он работает, его имя, фамилия, отчество и номер контактного телефона. Первичный ключ — veterinarian\_id.
2. Clinic — клиника. Таблица содержит названия клиники, адрес клиники. Первичный ключ — clinic\_name.
3. Vaccination —вакцины. Таблица содержит идентификатор собаки, дату прохождения: первой вакцины от инфекций, второй вакцины от инфекций, вакцины от бешенства, третьей вакцины от инфекций, второй вакцины от бешенства, ежегодной вакцины. Первичный ключ — dog\_id.
4. Owner — хозяин. Таблица содержит паспорт владельца, имя владельца, фамилия владельца, отчество владельца, номер телефона владельца. Первичный ключ — owner\_passport.
5. Dog — собаки. Таблица содержит идентификаторы собак, паспорт владельца собаки, кличка собаки, возраст собаки, порода собаки, название клиники, к которой относится собака, идентификатор ветеринара, за которым закреплена собака. Первичный ключ – dog\_id.

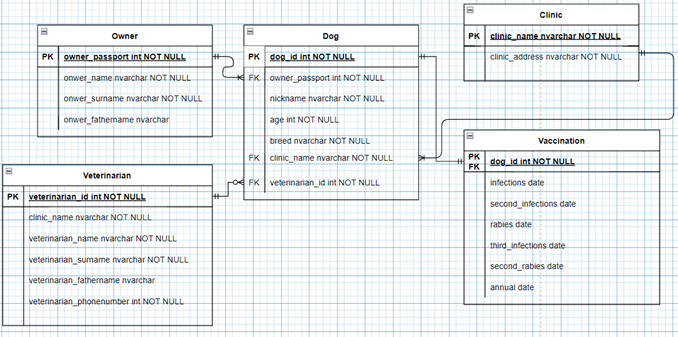


Рис. 4.1 – ER диаграмма

# Создание базы данных

Создадим базу данных:

CREATE DATABASE dogDb ON

(

NAME = dogDb\_dat,

FILENAME = 'C:\database\dogDbDat.mdf',

SIZE = 5MB, MAXSIZE = 50MB, FILEGROWTH = 10%

)

LOG ON

(

NAME = dogDb\_log,

FILENAME = 'C:\database\dogDbLog.ldf',

SIZE = 5MB, MAXSIZE = 50MB, FILEGROWTH = 10%

);

Результат выполнения запроса (Рис. 5.1):

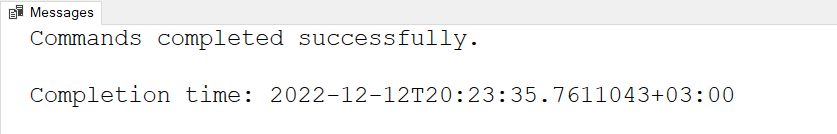


Рис. 5.1 – результат выполнения запроса

Создадим отдельную схему IDZ:

USE dogDb

GO

CREATE SCHEMA IDZ AUTHORIZATION dbo;

Результат выполнения запроса (Рис. 5.2):

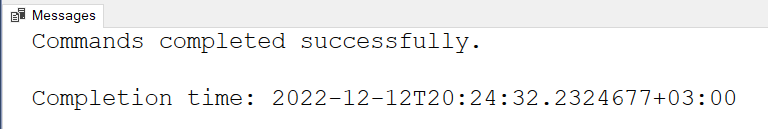


Рис. 5.2 – результат выполнения запрос

# Создание таблиц

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Cписок ветеринаров** | | | | Имя таблицы:  **Veterinarian** | | | |
| Дата разработки:  12.12.2022 | | |
| Порядковый номер таблицы:  **1** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор | | veterinarian\_id | int | | P | NOT NULL | |  |
|  | 2 | Название клиники | | clinic\_name | Nvarchar(50) | |  | NOT NULL,  clinic\_name != '' | |  |
|  | 3 | Имя ветеринара | | veterinarian\_name | Nvarchar(50) | |  | NOT NULL  veterinarian\_name != ' ' | |  |
|  | 4 | Фамилия ветеринара | | veterinarian\_surname | Nvarchar(50) | |  | NOT NULL,  Veterinarian\_surname != ' ' | |  |
|  | 5 | Отчество ветеринара | | veterinarian\_fathername | Nvarchar(50) | |  | veterinarian\_fathername != ' ' | |  |
|  | 6 | Номер телефона ветеринара | | veterinarian\_phonenumber | Nvarchar(11) | |  | NOT NULL, CHECK(veterinarian\_phonenumber LIKE '8[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  ); | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Список клиник** | | | | Имя таблицы:  **Clinic** | | | |
| Дата разработки:  12.12.2022 | | |
| Порядковый номер таблицы:  **2** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Название клиники | | clinic\_name | Nvarchar(50) | | P | NOT NULL,  clinic\_name != ' ' | |  |
|  | 2 | Адресс клиники | | clinic\_address | Nvarchar(50) | |  | NOT NULL,  clinic\_address!= ' ' | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Список вакцин, сделанной собаке** | | | | Имя таблицы:  **Vaccination** | | | |
| Дата разработки:  12.12.2022 | | |
| Порядковый номер таблицы:  **3** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор | | dog\_id | int | | P | NOT NULL | |  |
|  | 2 | Первая вакцина от инфекций | | infections | date | |  |  | |  |
|  | 3 | Вторая вакцина от инфекций | | second\_infections | date | |  |  | |  |
|  | 4 | Вакцина от бешенства | | rabies | date | |  |  | |  |
|  | 5 | Третья вакцина от инфекций | | third\_infections | date | |  |  | |  |
|  | 6 | Вторая вакцина от бешенства | | second\_rabies | date | |  |  | |  |
|  | 7 | Ежегодная вакцина | | annual | date | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Список владельцев собак** | | | | Имя таблицы:  **Owner** | | | |
| Дата разработки:  12.12.2022 | | |
| Порядковый номер таблицы:  **4** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Паспорт владельца | | owner\_passport | Nvarchar(10) | | P | NOT NULL | |  |
|  | 2 | Имя владельца | | owner\_name | Nvarchar(50) | |  | NOT NULL,owner\_name != ' ' | |  |
|  | 3 | Фамилия владельца | | owner\_surname | Nvarchar(50) | |  | NOT NULL, owner\_surname != ' ' | |  |
|  | 4 | Отчество владельца | | owner\_fathername | Nvarchar(50) | |  | owner\_fathername != ' ' | |  |
|  | 5 | Номер телефона владельца | | owner\_phonenumber | Nvarchar(12) | |  | NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Список собак** | | | | Имя таблицы:  **Dog** | | | |
| Дата разработки:  12.12.2022 | | |
| Порядковый номер таблицы:  **5** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор | | dog\_id | int | | P | NOT NULL | |  |
|  | 2 | Паспорт владельца собаки | | owner\_passport | Nvarchar(50) | | F | NOT NULL | |  |
|  | 3 | Кличка собаки | | nickname | Nvarchar(50) | |  | NOT NULL,  nickname != ' ' | |  |
|  | 4 | Возраст собаки | | age | Nvarchar(50) | |  | NOT NULL | |  |
|  | 5 | Порода собаки | | breed | Nvarchar(50) | |  | NOT NULL,  breed != ' ' | |  |
|  | 6 | Имя клиники, к которой относится собака | | clinic\_name | Nvarchar(50) | | F | NOT NULL | |  |
|  | 7 | Идентификатор ветеринара, за которым закреплена собака | | veterinarian\_id | Nvarchar(50) | |  | NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

В качестве ограничения целостности в основном используется NOT NULL, но также для возраста собаки предусмотрено ограничение:

CONSTRAINT CK\_Dog\_age CHECK(age > 0),

Создание таблицы «Veterinarian»:

USE dogDb

GO

CREATE TABLE IDZ.Veterinarian(

veterinarian\_id int NOT NULL PRIMARY KEY IDENTITY (1,1),

clinic\_name nvarchar(50) NOT NULL ,

CONSTRAINT CK\_Not\_clinic\_nickname CHECK(clinic\_name != ''),

veterinarian\_name nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_veterinarian\_name CHECK(veterinarian\_name != ''),

veterinarian\_surname nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_veterinarian\_surname CHECK(veterinarian\_surname != ''),

veterinarian\_fathername nvarchar(50),

CONSTRAINT CK\_Not\_veterinarian\_fathername CHECK(veterinarian\_fathername != ''),

veterinarian\_phonenumber nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_veterinarian\_phonenumber CHECK(veterinarian\_phonenumber LIKE '8[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

);

Результат выполнения запроса (Рис. 6.1):

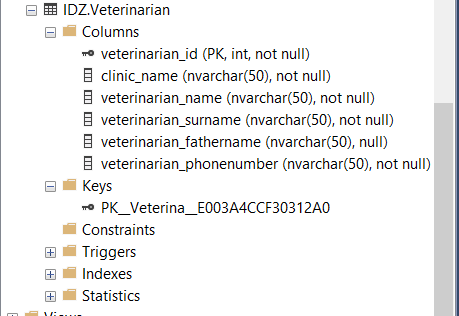


Рис. 6.1 – результат выполнения запроса

Создание таблицы «Clinic»:

USE dogDb

GO

CREATE TABLE IDZ.Clinic(

clinic\_name nvarchar(50) NOT NULL PRIMARY KEY,

clinic\_address nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_empty\_clinic\_address CHECK(clinic\_address != '')

);

Результат выполнения запроса (Рис. 6.2):

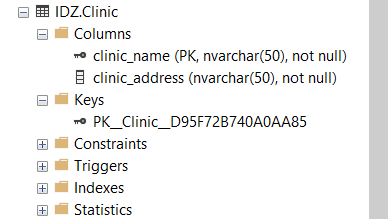


Рис. 6.2 – результат выполнения запроса

Создание таблицы «Owner»:

USE dogDb

GO

CREATE TABLE IDZ.Owner(

owner\_passport nvarchar(10) NOT NULL PRIMARY KEY,

CONSTRAINT CK\_Not\_empty\_owner\_passport CHECK(owner\_passport != ''),

owner\_name nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_empty\_owner\_name CHECK(owner\_name != ''),

owner\_surname nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_empty\_clinic\_surname CHECK(owner\_surname != ''),

owner\_fathername nvarchar(50),

CONSTRAINT CK\_Not\_empty\_owner\_father CHECK(owner\_fathername != ''),

owner\_phonenumber nvarchar(12) NOT NULL,

CONSTRAINT CK\_Not\_empty\_owner\_phonenumber CHECK(owner\_phonenumber != ''),

);

Результат выполнения запроса (Рис. 6.3):

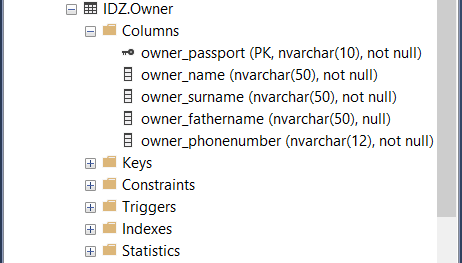


Рис. 6.3 – результат выполнения запроса

Создание таблицы «Dog»:

USE dogDb

GO

CREATE TABLE IDZ.Dog(

dog\_id int NOT NULL PRIMARY KEY IDENTITY(1,1),

owner\_passport nvarchar(10) REFERENCES IDZ.owner(owner\_passport),

nickname nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_empty\_nickname CHECK(nickname != ''),

age int NOT NULL,

CONSTRAINT CK\_Dog\_age CHECK(age > 0),

breed nvarchar(50) NOT NULL,

CONSTRAINT CK\_Not\_empty\_breed CHECK(breed != ''),

clinic\_name nvarchar(50) REFERENCES IDZ.clinic(clinic\_name),

veterinarian\_id int REFERENCES IDZ.Veterinarian(veterinarian\_id)

);

Результат выполнения запроса (Рис. 6.4):

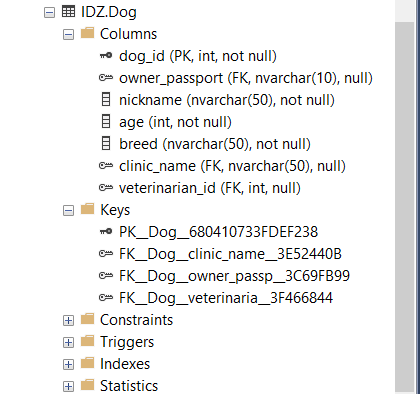


Рис. 6.4 – результат выполнения запроса

Создание таблицы «Vaccination»:

USE dogDb

GO

CREATE TABLE IDZ.Vaccination(

dog\_id int NOT NULL PRIMARY KEY REFERENCES IDZ.dog(dog\_id),

infections date,

second\_infections date,

rabies date,

third\_infections date,

second\_rabies date,

annual date

);

Результат выполнения запроса (Рис. 6.5):

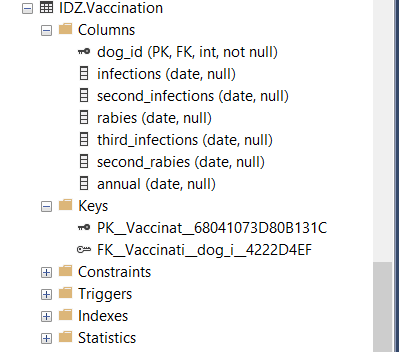


Рис. 6.5 – результат выполнения запроса

# Заполнение таблиц данными

Заполнение таблицы «Owner»:

INSERT INTO IDZ.Owner

VALUES (8015215425,'Вадим','Хаматов', 'Рустамович','89378482007'),

(8011765234,'Андрей','Блюдин', 'Игоревич','88008228007'),

(8015065712,'Дмитрий','Горюнов', 'Кириллович','88003236078'),

(8025665841,'Семен','Григорьев', 'Михайлович','89378432067'),

(8045665091,'Захар','Пархомчук', 'Михайлович','89278434007'),

(8055365124,'Андрей','Зонов', 'Русланович','89374412007'),

(8065265152,'Софья','Усольцева', 'Григорьевна','89348182017');

Результат выполнения запроса (Рис. 7.1), (Рис. 7.2):

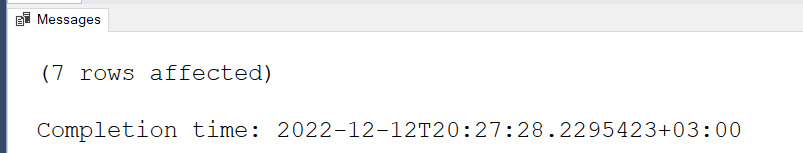


Рис. 7.1 – результат выполнения запроса

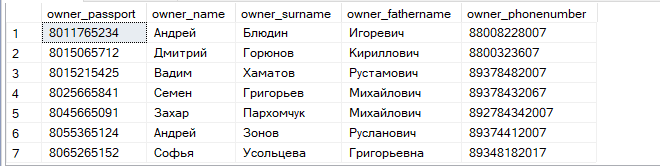


Рис. 7.2 – результат заполнения таблицы

Заполнение таблицы «Clinic»:

INSERT INTO IDZ.Clinic

VALUES ('Котонай', 'ул. Костюшко 38'),

('Мой питомец', 'ул. Московский проспект 175'),

('Айболит', 'Невский проспект 192');

Результат выполнения запроса (Рис. 7.3), (Рис. 7.4):

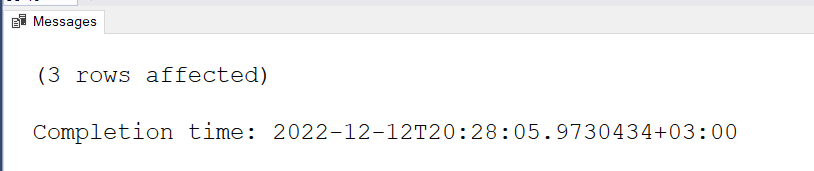


Рис. 7.3 – результат выполнения запроса

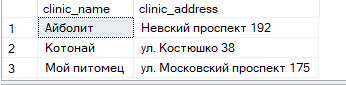


Рис. 7.4 – результат заполнения таблицы

Заполнение таблицы «Veterinarian»:

INSERT INTO IDZ.Veterinarian

VALUES ('Мой питомец','Максим','Киншаров','Олегович',89378482007),

('Айболит','Кирилл','Дементьев','Владимирович',89273231743),

('Айболит','Родион','Ахмадуллин','Эдуардович',89113232202),

('Котонай','Арсений','Бубновский','Олегович',89138482123);

Результат выполнения запроса (Рис. 7.5), (Рис. 7.6):

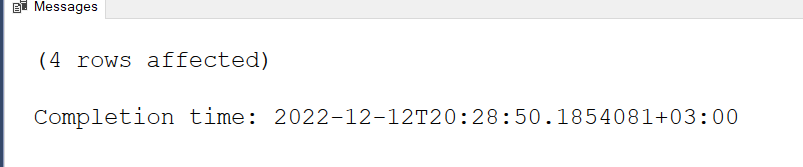


Рис. 7.5 – результат выполнения запроса

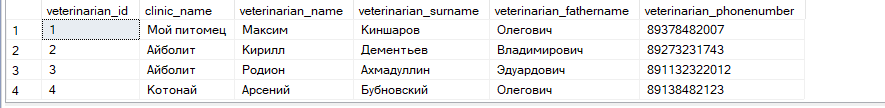


Рис. 7.6 – результат заполнения таблицы

Заполнение таблицы «Dog»:

INSERT INTO IDZ.Dog

VALUES (8015215425,'Шарик',10,'Такса', 'Мой питомец',1),

(8011765234,'Бобик',10,'Хаски', 'Мой питомец',2),

(8015065712,'Тузик',10,'Хаски', 'Мой питомец',2),

(8025665841,'Тихон',10,'Овчарка','Айболит',2),

(8045665091,'Рэкс',10,'Ламбрадор','Айболит',3),

(8045665091,'Мухтар',10,'Алабай','Котонай',4);

Результат выполнения запроса (Рис. 7.7), (Рис. 7.8):

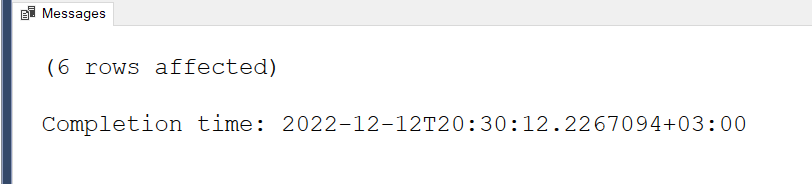


Рис. 7.7 – результат выполнения запроса

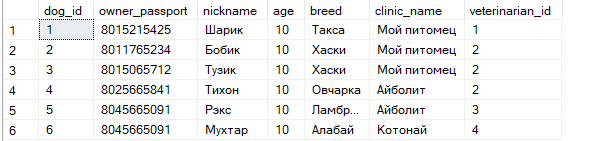


Рис. 7.8 – результат заполнения таблицы

Заполнение таблицы «Vaccination»:

INSERT INTO IDZ.Vaccination

VALUES (1,'2022-02-03','2022-02-24','2022-02-24','2022-03-17','2022-03-17','2022-04-01'),

(2,'2020-12-01','2020-12-22','2020-12-22','2021-01-12','2021-01-12','2021-02-14'),

(3,'2019-08-01','2019-08-22','2019-08-22','2019-09-12','2019-09-12','2019-10-05'),

(4,'2022-12-12',NULL, NULL, NULL, NULL, NULL),

(5,'2022-02-03','2022-02-21','2022-02-22', NULL,NULL, NULL),

(6,'2022-04-07','2022-04-28','2022-04-28','2022-05-19','2022-05-19','2022-06-03');

Результат выполнения запроса (Рис. 7.9), (Рис. 7.10):

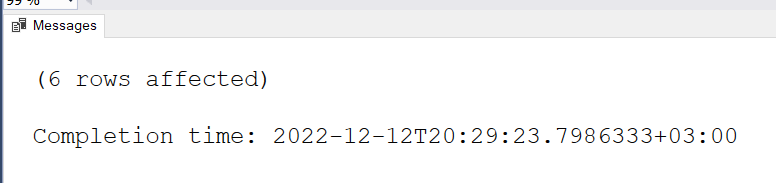


Рис. 7.9 – результат выполнения запроса

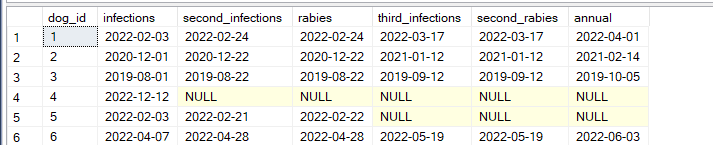


Рис. 7.10 – результат заполнения таблицы

# Разработка объектов промежуточного слоя

## Хранимые процедуры

Создание хранимой процедуры, которая добавит в таблицу с информацией о вакцинах новую запись (вместо NULL), в поле следующей по очереди вакцине.

CREATE PROCEDURE makeVaccin1(@current\_dog\_id int)

AS

IF ((SELECT infections FROM IDZ.Vaccination WHERE dog\_id = @current\_dog\_id) IS NULL)

begin

UPDATE IDZ.Vaccination

SET infections = GETDATE()

WHERE dog\_id = @current\_dog\_id

end

else

IF ((SELECT second\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_dog\_id) IS NULL)

begin

UPDATE IDZ.Vaccination

SET second\_infections = GETDATE()

WHERE dog\_id = @current\_dog\_id

end

else

IF ((SELECT rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_dog\_id) IS NULL)

begin

UPDATE IDZ.Vaccination

SET rabies = GETDATE()

WHERE dog\_id = @current\_dog\_id

end

else

IF ((SELECT third\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_dog\_id) IS NULL)

begin

UPDATE IDZ.Vaccination

SET third\_infections = GETDATE()

WHERE dog\_id = @current\_dog\_id

end

else

IF ((SELECT second\_rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_dog\_id) IS NULL)

begin

UPDATE IDZ.Vaccination

SET second\_rabies = GETDATE()

WHERE dog\_id = @current\_dog\_id

end

else

IF ((SELECT annual FROM IDZ.Vaccination WHERE dog\_id = @current\_dog\_id) IS NULL)

begin

UPDATE IDZ.Vaccination

SET annual = GETDATE()

WHERE dog\_id = @current\_dog\_id

end

GO

Результат выполнения запроса (Рис. 8.1.1), (Рис. 8.1.2):

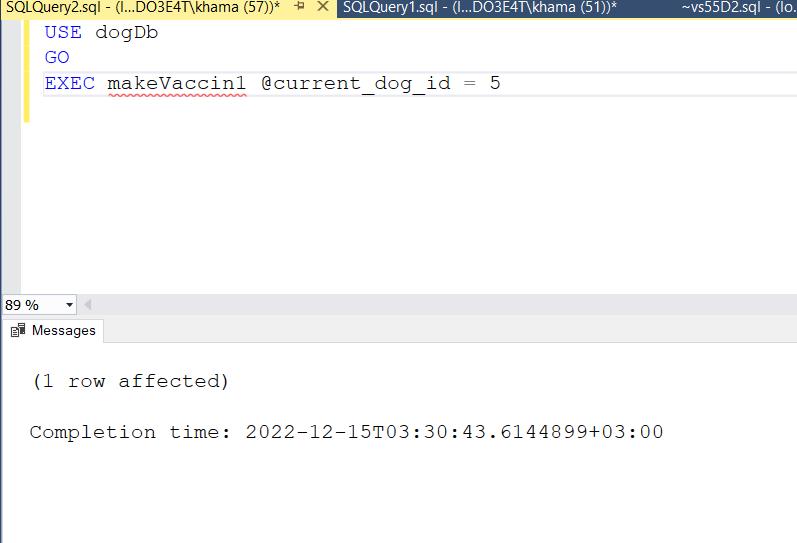


Рис. 8.1.1 – результат выполнения процедуры

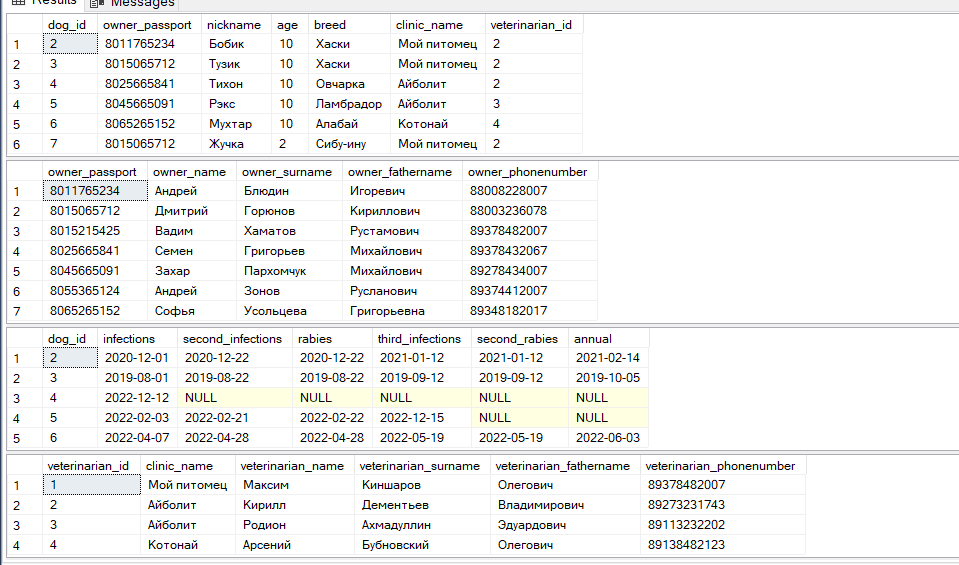


Рис. 8.1.2 – результат выполнения процедуры

Создание пользовательской процедуры, которая полностью убирает всю информации о собаке с переданном в функцию id.

CREATE PROC delete\_dog\_fromDb (@id int)

AS

DELETE FROM IDZ.Vaccination WHERE dog\_id = @id

DELETE FROM IDZ.Dog WHERE dog\_id = @id

GO

Результат выполнения запроса (Рис. 8.1.3), (Рис. 8.1.4):

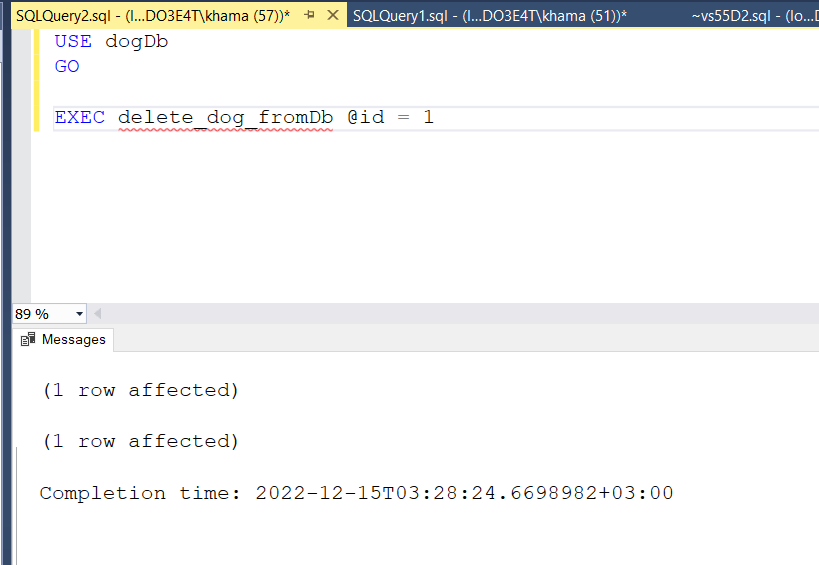


Рис. 8.1.3 – результат выполнения процедуры

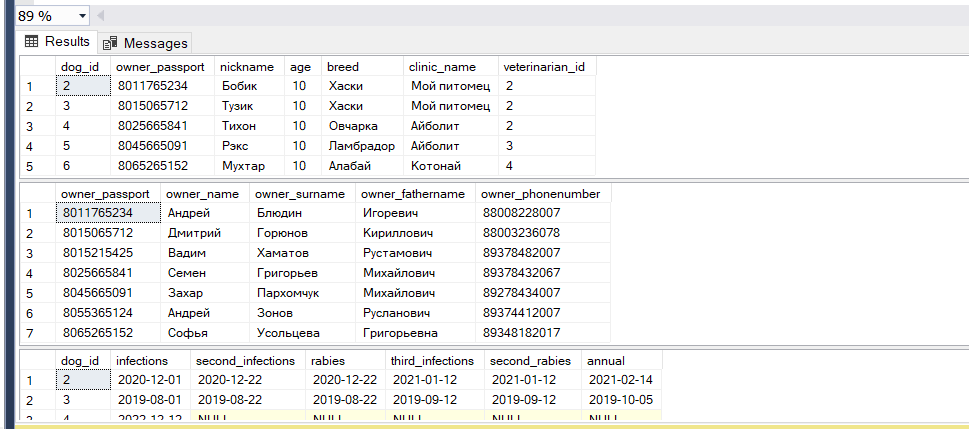


Рис. 8.1.4 – результат выполнения процедуры

Создание хранимой процедуры, которая добавит еще одну собаку владельцу, с переданным номером паспорта. Автоматически прикрепит собаку к клинике и ветеринару.

CREATE PROC add\_New\_Dog\_For\_Owner(@passport nvarchar(10),@nick nvarchar(50),@current\_age int,@current\_breed nvarchar(50))

AS

DECLARE @current\_owner\_veterinarian int = (SELECT veterinarian\_id FROM IDZ.Dog WHERE owner\_passport = @passport)

DECLARE @current\_owner\_clinic nvarchar(50) = (SELECT clinic\_name FROM IDZ.Dog WHERE owner\_passport = @passport)

INSERT INTO IDZ.Dog

VALUES(@passport,@nick,@current\_age,@current\_breed,@current\_owner\_clinic,@current\_owner\_veterinarian)

GO

Результат выполнения запроса (Рис. 8.1.5), (Рис. 8.1.6):

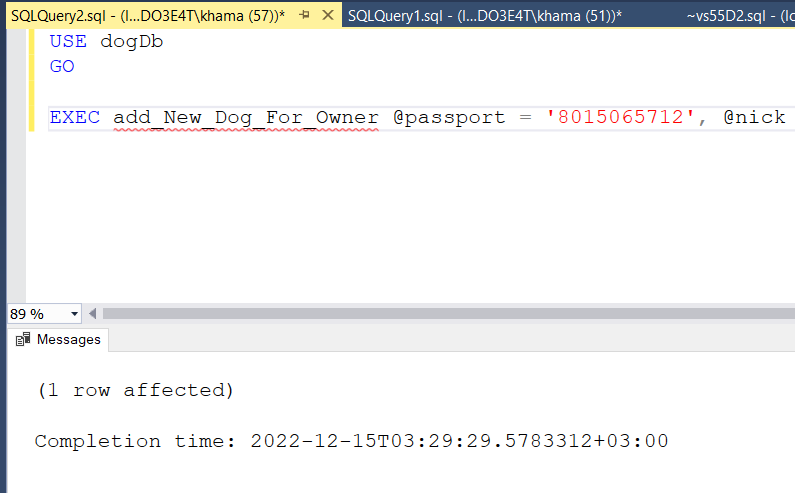


Рис. 8.1.5 – результат выполнения процедуры

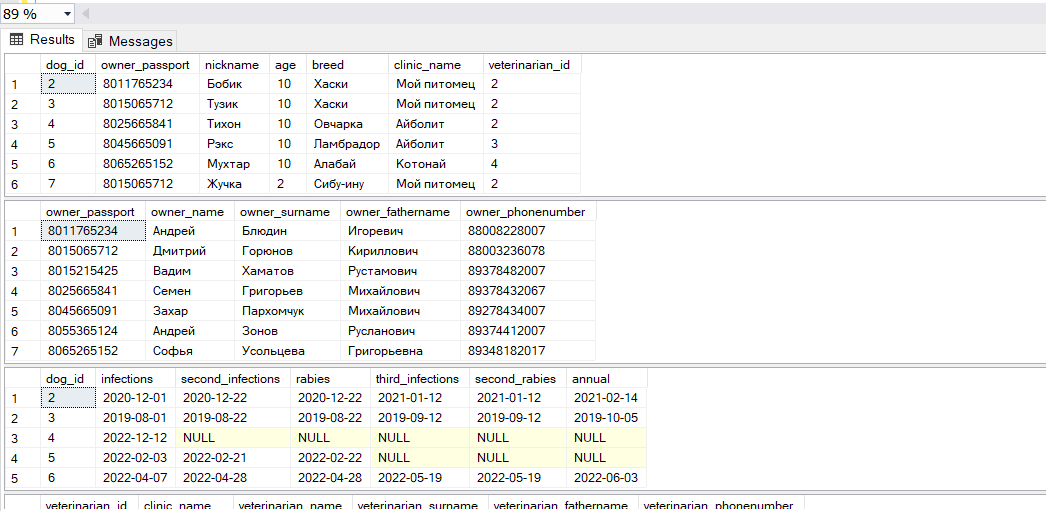


Рис. 8.1.6 – результат выполнения процедуры

## UDF

Создание пользовательской функции, которая вернет значение о том, через сколько дней нужно поставить собаке следующую прививку.

CREATE FUNCTION checkDaysForNextVaccin (@current\_id int) RETURNS INT

AS

BEGIN

DECLARE @lastVaccination date

DECLARE @dayForVaccination int

IF ((SELECT infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NULL)

BEGIN

SET @dayForVaccination = 0

RETURN @dayForVaccination

END

ELSE

BEGIN

IF

((SELECT second\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NULL)

begin

SET @lastVaccination = (SELECT infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id)

SET @dayForVaccination = DATEDIFF(day,@lastVaccination,GETDATE())

RETURN @dayForVaccination

end

IF

((SELECT rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NULL)

begin

SET @lastVaccination = (SELECT second\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id)

SET @dayForVaccination = DATEDIFF(day,@lastVaccination,GETDATE())

RETURN @dayForVaccination

end

IF

((SELECT third\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NULL)

begin

SET @lastVaccination = (SELECT rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id)

SET @dayForVaccination = DATEDIFF(day,@lastVaccination,GETDATE())

RETURN @dayForVaccination

end

IF

((SELECT second\_rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NULL)

begin

SET @lastVaccination = (SELECT third\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id)

SET @dayForVaccination = DATEDIFF(day,@lastVaccination,GETDATE())

RETURN @dayForVaccination

end

IF

((SELECT annual FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NULL)

begin

SET @lastVaccination = (SELECT second\_rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id)

SET @dayForVaccination = DATEDIFF(day,@lastVaccination,GETDATE())

end

end

RETURN @dayForVaccination

END;

GO

Результат выполнения запроса (Рис. 8.2.1), (Рис. 8.2.2):

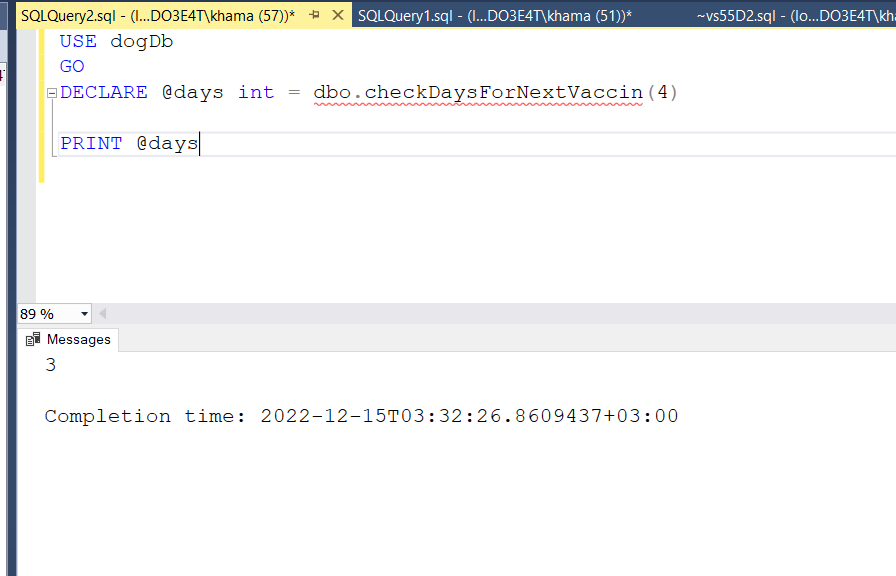


Рис. 8.2.1 – результат выполнения функции

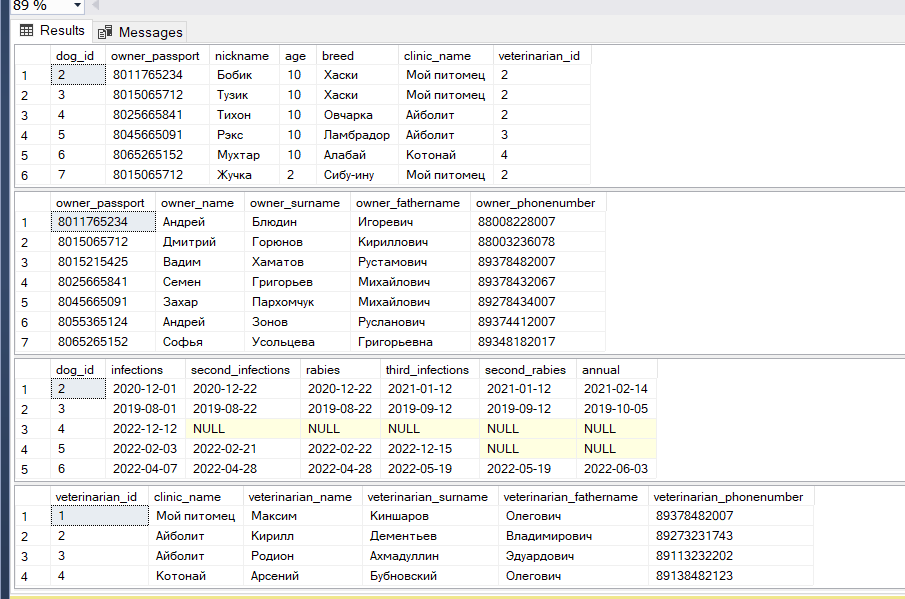


Рис. 8.2.2 – результат выполнения функции

USE dogDb

GO

CREATE FUNCTION check\_Neccecary\_Vaccin (@current\_id int ) RETURNS smallint

AS

begin

DECLARE @need\_Vaccin smallint

IF (((SELECT infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT second\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT third\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT second\_rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT annual FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL))

SET @need\_Vaccin = 0

ELSE

SET @need\_Vaccin = 1

RETURN @need\_Vaccin

end

GO

Создание пользовательской функции, которая позволит просмотреть всю информацию о поставленных прививках определенной собаке.

CREATE FUNCTION checkFullVaccinPack (@current\_id int) RETURNS bit

AS

begin

DECLARE @need\_Vaccin bit

IF (((SELECT infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT second\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT third\_infections FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT second\_rabies FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL)

AND

((SELECT annual FROM IDZ.Vaccination WHERE dog\_id = @current\_id ) IS NOT NULL))

SET @need\_Vaccin = 1

ELSE

SET @need\_Vaccin = 0

RETURN @need\_Vaccin

end

GO

Результат выполнения запроса (Рис. 8.2.3), (Рис. 8.1.4):

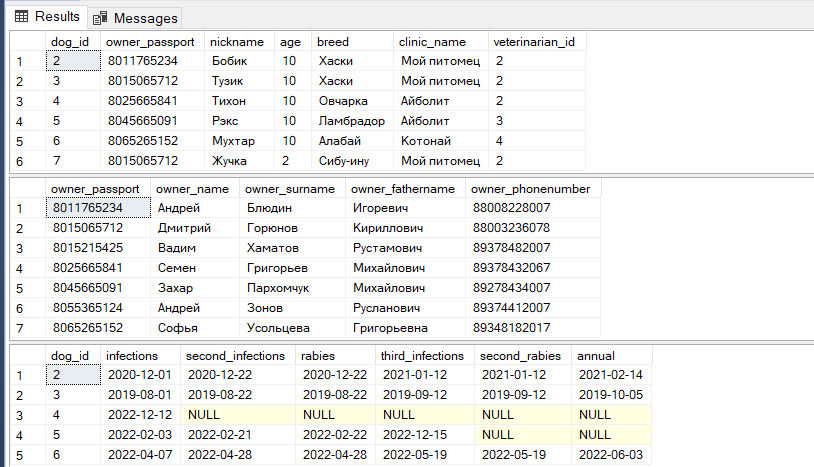


Рис. 8.2.3 – результат выполнения функции

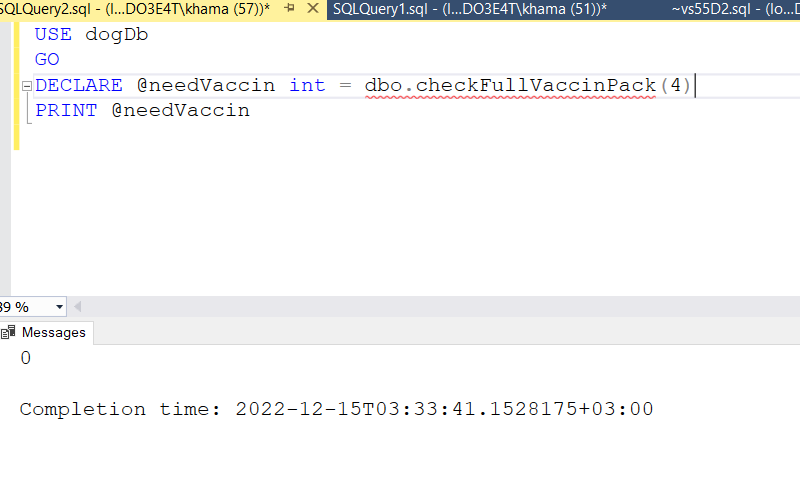


Рис. 8.2.4 – результат выполнения функции

Создание пользовательской функции, которая выведет таблицу всех собак, которые стоят на учете у ветеринара с переданным @current\_veterinarian\_id.

CREATE FUNCTION printVetsAndPats(@current\_veterinarian\_id int) RETURNS TABLE

AS

RETURN(

SELECT v.veterinarian\_name,v.veterinarian\_surname,d.nickname

FROM IDZ.Dog AS d INNER JOIN IDZ.Veterinarian as v ON d.veterinarian\_id = v.veterinarian\_id

WHERE v.veterinarian\_id = @current\_veterinarian\_id

)

GO

Результат выполнения запроса (Рис. 8.2.5), (Рис. 8.2.6):

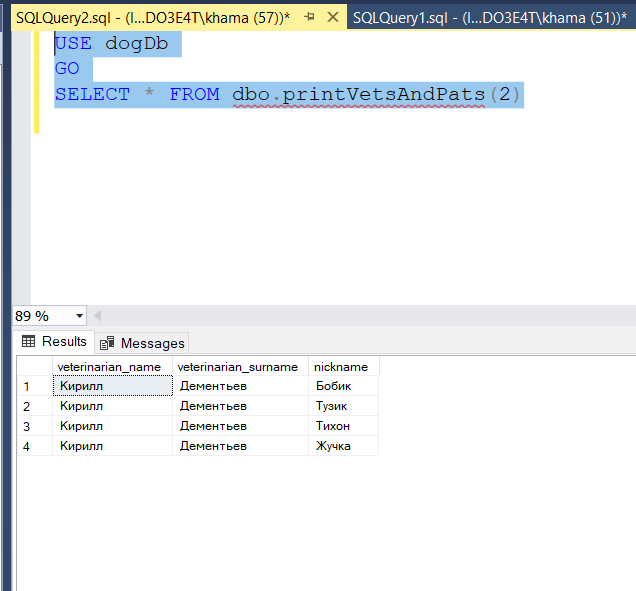


Рис. 8.2.5 – результат выполнения функции

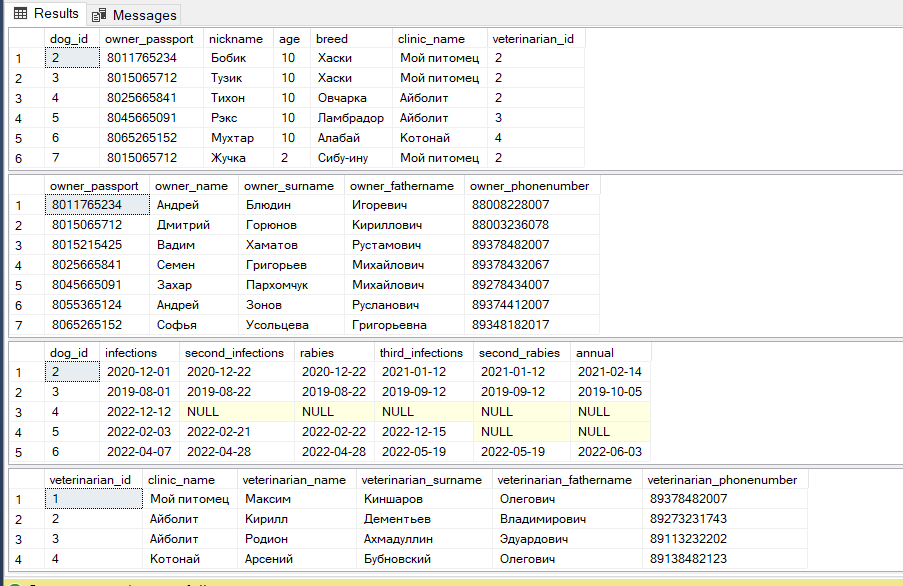


Рис. 8.2.6 – результат выполнения функции

## Представления

Создание представления, которое покажет всю информацию о параметрах собак. Данное представление помогает не выводит личную информацию о владельце.

create view DogsInfo AS

SELECT nickname,age,breed FROM IDZ.Dog

Результат выполнения запроса (Рис. 8.3.1):

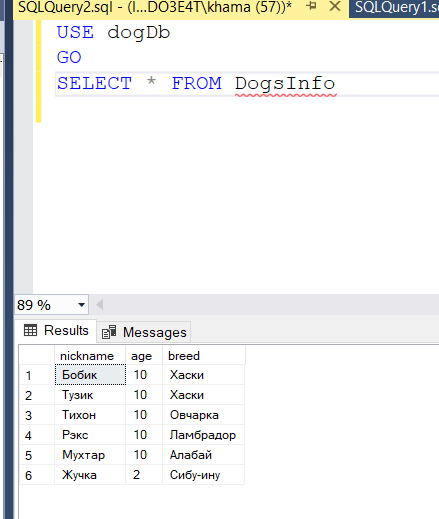


Рис. 8.3.1 – результат выполнения процедуры

Создание представления, которая покажет количество собак в базе и их средний возраст.

create view dogsCountAndAvgAge AS

SELECT count(dog\_id) AS 'Количество собак', AVG(AGE) AS 'Средний возраст собак' FROM IDZ.Dog

Результат выполнения запроса (Рис. 8.3.2):

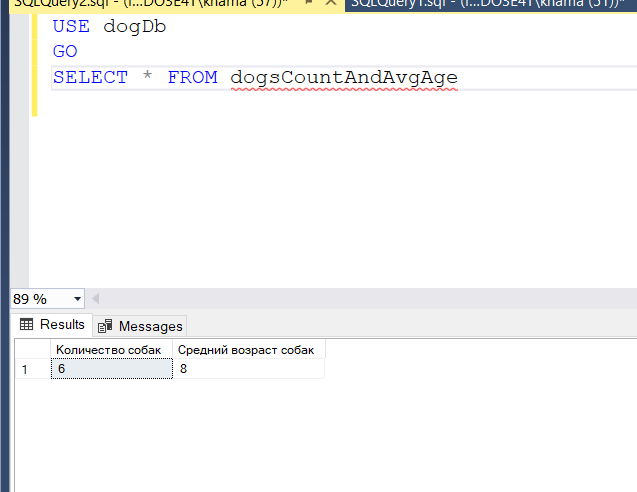


Рис. 8.3.2 – результат выполнения процедуры

Создание представления, которое покажет безопасную информацию о владельцев (без личных данных).

create view ownerSafeInfo AS

SELECT owner\_name,owner\_surname,owner\_fathername FROM IDZ.Owner

Результат выполнения запроса (Рис. 8.3.3):

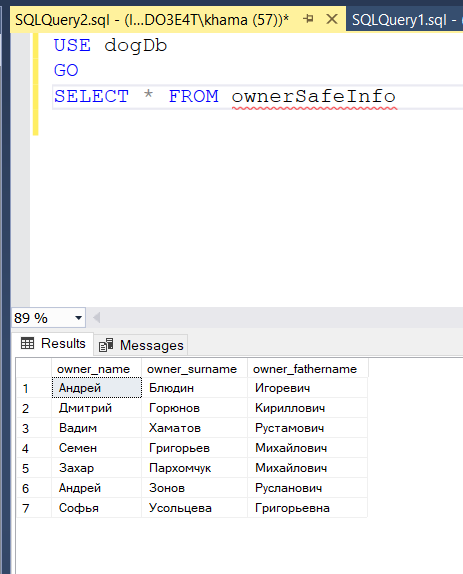


Рис. 8.3.3 – результат выполнения процедуры

# Разработка стратегии резервного копирования

В данной работе будет создана дифференциальная модель резервного копирования. Данное копирование в первый раз копирует БД полностью, а в последующие разы будут скопированы только те файлы, которые были изменены, что ускоряет работу.

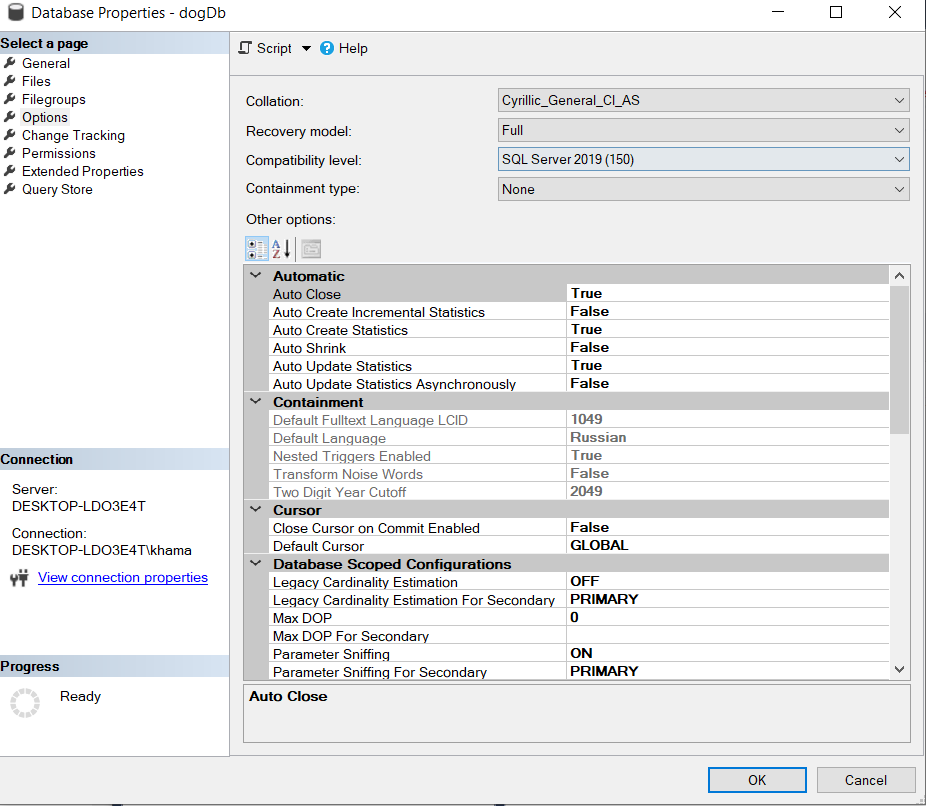
Перед началом изменим модель восстановления базы данных на «Full» (Рис.9.1).

Рис. 9.1 – Изменение Recovery model на тип Full

SQL-скрипт с созданием полной резервной копии БД (Рис. 9.2):

BACKUP DATABASE dogDb

TO DISK = 'C:\database\rersrvdogDb.bak'

WITH INIT

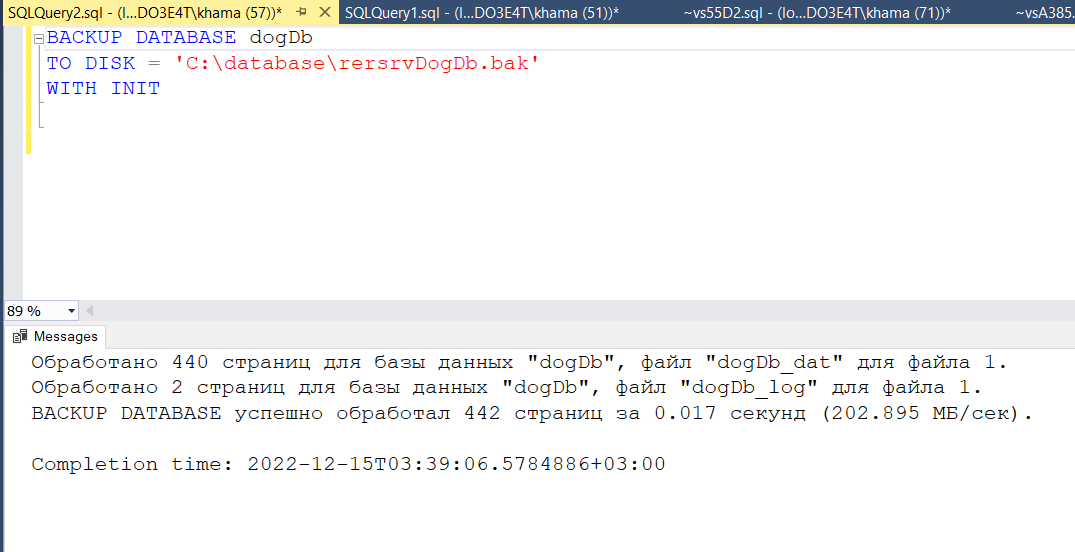


Рис. 9.2 - результат выполнения SQL-скрипта

Создадим разностную резервную копию (Рис. 9.3):

BACKUP DATABASE dogDb

TO DISK = 'C:\database\rersrvdogDbRAS.bak'

WITH DIFFERENTIAL, INIT

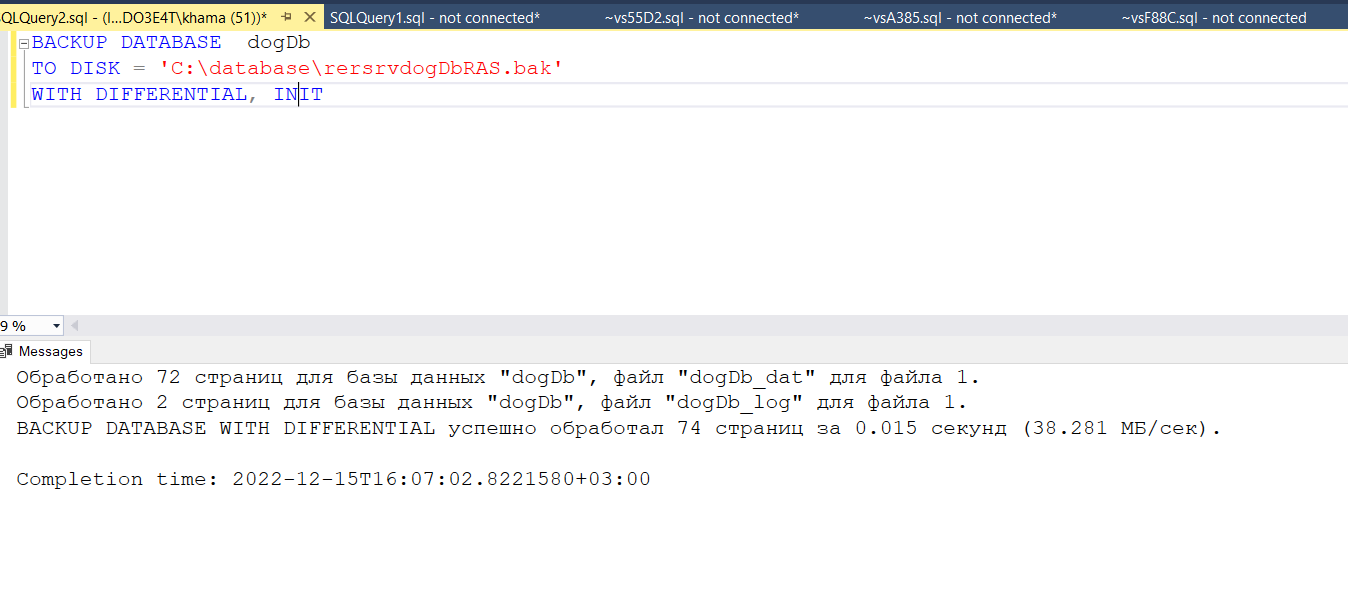
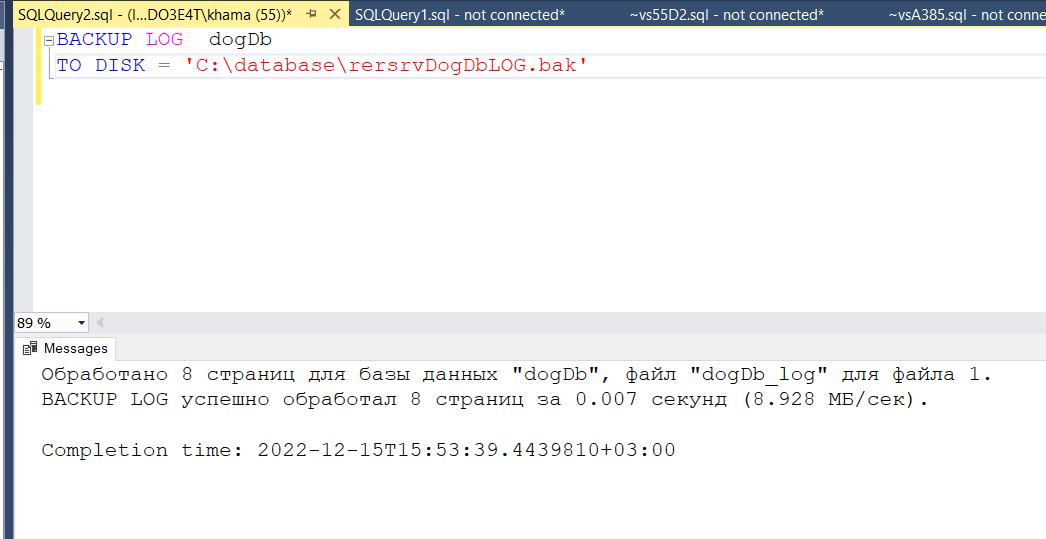


Рис. 9.3 - результат выполнения SQL-скрипта

Создадим резервную копию журнала транзакций (Рис. 9.4)

BACKUP LOG dogDb

TO DISK = 'C:\database\rersrvDogDbLOG.bak'

 Рис. 9.4 - результат выполнения SQL-скрипта

Удостоверимся, что файлы действительно создались. (Рис. 9.5)

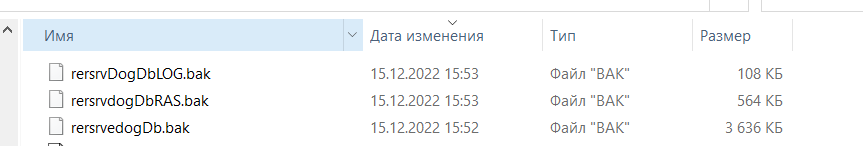


Рис. 9.5 – создание файлов backup’a

Удалим нашу БД и восстановим из созданных ранее файлов, обновим дерево с таблицами. (Рис. 9.6)

RESTORE DATABASE dogDb

FROM DISK='C:\database\rersrvedogDb.bak'

WITH NORECOVERY

RESTORE DATABASE dogDb

FROM DISK='C:\database\rersrvdogDbRAS.bak'

WITH NORECOVERY

RESTORE LOG dogDb

FROM DISK='C:\database\rersrvDogDbLOG.bak'

WITH NORECOVERY

RESTORE DATABASE dogDb

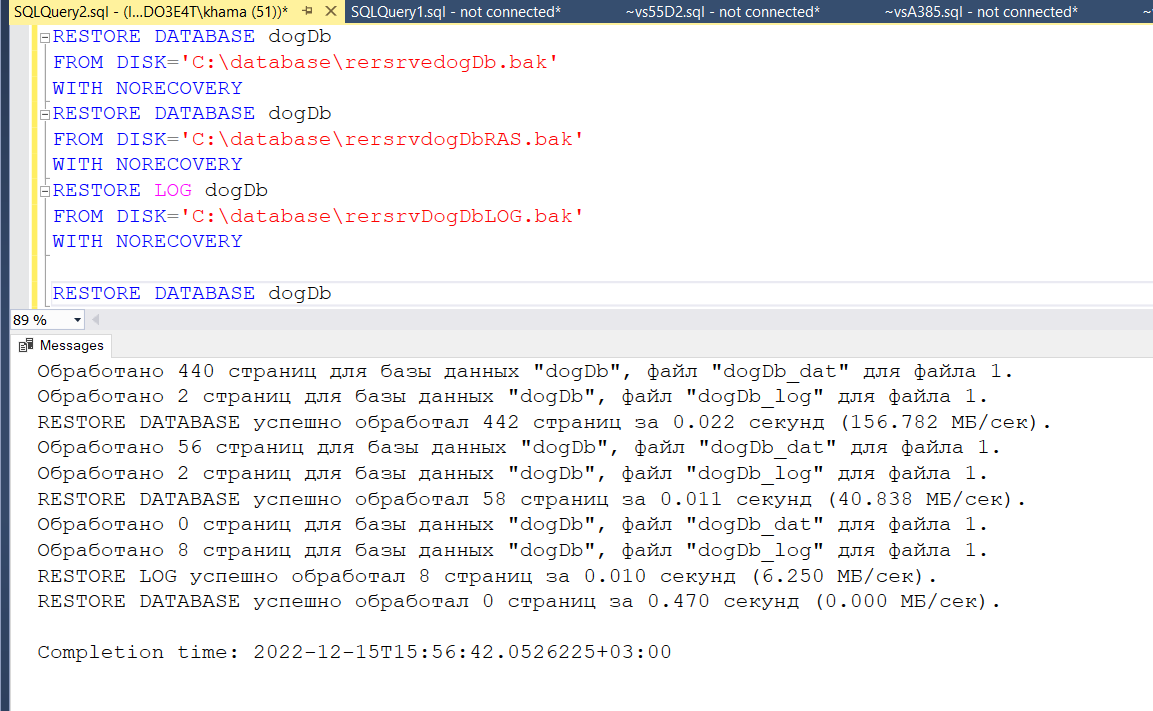


Рис. 9.6 - результат выполнения SQL-скрипта

# Заключение

В результате выполнения индивидуального домашнего задания была спроектирована и создана база данных собак, состоящая из пяти таблиц. Для полей таблиц установлены различные ограничения целостности, что позволяет проверять пользователя на правильность ввода им данных. База данных была заполнена данными с учетом краткой информации о предметной области. Также были разработаны несколько объектов промежуточного слоя:

1. Хранимые процедуры для произведения новой вакцины, полное удаление собаки из базы данных и отвязка ее от ветеринара и клиники, добавление новой собаки хозяину.

2. Функции для подсчета дней до произведения следующей вакцины собаке, информации о всех собаках, которые стоят на учете у определенного ветеринара, выводе информации о прививках, сделанных собакам.

3. Представления для безопасного вывода информации о владельцах (без личных данных), для отображения количества собак в базе и их средний возраст, для вывода информации о параметрах собаки (без личных данных владельца и ветеринара).

Так же в данной работе произведена дифференциальная модель резервного копирования. Данное копирование в первый раз копирует БД полностью, а в последующие разы будут скопированы только те файлы, которые были изменены. Данный вид модели позволяет увеличить скорость работы.

# Список использованных источников

1.Начало работы с запросами с помощью Transact-SQL. – URL: <https://learn.microsoft.com/ru-ru/training/paths/get-started-querying-with-transact-sql/> (дата обращения 13.12.2022).

2.Основные команды SQL, которые должен знать каждый программист [Электронный ресурс] Режим доступа: <https://tproger.ru/translations/sql-recap/> (дата обращения 14.12.2022)

3. Просмотр или изменение модели восстановления базы данных (SQL Server) [Электронный ресурс] Режим доступа: https://learn.microsoft.com/ru-ru/sql/relational-databases/backup-restore/view-or-change-the-recovery-model-of-a-database-sql-server?view=sql-server-ver16 (дата обращения 15.12.2022)