МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

автоматизованих систем обробки інформації та управління

(повна назва кафедри, циклової комісії)

**КУРСОВА РОБОТА**

з баз данних - 1

(назва дисципліни)

на тему: ІС «Онлайн-платформа психотерапії»

Студента II курсу, групи ІП-96

Бурятова Олексія Олексійовича

Спеціальності 121 «Інженерія програмного забезпечення»

Керівник ст. викладач Недашківський Є.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Члени комісії |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |
|  |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |

Київ-2020

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

(назва вищого навчального закладу)

## Кафедра автоматизованих систем обробки інформації і управління

Дисципліна бази даних - 1

Напрям "Програмна інженерія"

Курс 2 Група ІП-96 Семестр 3

### **ЗАВДАННЯ**

#### на курсову роботу студента

##### Бурятова Олексія Олексійовича

(прізвище, ім’я, по батькові)

1. Тема роботи

2. Строк здачі студентом закінченої роботи

3. Вихідні дані до роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

5. Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )

6. Дата видачі завдання

ЗМІСТ

[Кафедра автоматизованих систем обробки інформації і управління 2](#_Toc58867360)

[**ЗАВДАННЯ** 2](#_Toc58867361)

[**Вступ** 4](#_Toc58867362)

[**1** **ПЕРЕДМОВА** 5](#_Toc58867363)

[**2** **Теоретичні відомості** 6](#_Toc58867364)

[**3** **Опис програмного забезпечення** 8](#_Toc58867365)

[3.1. ER Діаграма 8](#_Toc58867366)

[3.2. Опис сутностей 8](#_Toc58867367)

[3.3. Фізична модель бази даних 9](#_Toc58867368)

[**4** **Інструкція користувача** 10](#_Toc58867370)

[4.1. Робота з програмою 10](#_Toc58867371)

[4.2. Системні вимоги 26](#_Toc58867372)

[**5** **Перевірка на коректність** 28](#_Toc58867373)

[**7** **Аналіз і узагальнення результатів** 30](#_Toc58867374)

[Додаток А Технічне завдання 49](#_Toc58867375)

[Додаток Б Тексти програмного коду 52](#_Toc58867376)

[Перелік посилань 94](#_Toc58867377)

# **Вступ**

Пояснювальна записка до курсової роботи: 93 сторінки, 25 рисунків, 1 таблиця, 2 посилання.

Об’єкт дослідження: дослідження робити з базами даних.

Мета роботи: розробити онлайн-платформу психотерапії, використовуючи базу даних SQL.

Вивчено синтаксис реляційної бази даних MSSQL. Приведені змістовні приклади даних у кожній з таблиць бази даних.

# **ПЕРЕДМОВА**

Дана курсова робота була призначена для поглиблення та закріплення теоретичних знань студентів, що навчаються за дисципліною «Бази даних-1». Написання курсової роботи дає студентові можливість навчитися самостійно і творчо використовувати наукові джерела та узагальнювати теоретичні положення.

# **Теоретичні відомості**

За означенням третя нормальна форма (3НФ, 3NF) вимагає, аби дані в таблиці залежали винятково від основного ключа:

* Схема бази даних повинна відповідати всім вимогам другої нормальної форми.
* Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.

Друга нормальна форма (2НФ, 2NF) вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа:

* Схема бази даних повинна відповідати вимогам першої нормальної форми.
* Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

Перша нормальна форма (1НФ, 1NF) утворює ґрунт для структурованої схеми бази даних:

* Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.
* Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.
* Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.

ER-мо За вікіпедією третя нормальна форма (3НФ, 3NF) вимагає, аби дані в таблиці залежали винятково від основного ключа:

* Схема бази даних повинна відповідати всім вимогам другої нормальної форми.
* Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.

З відти ж ми знаємо, що друга нормальна форма (2НФ, 2NF) вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа:

* Схема бази даних повинна відповідати вимогам першої нормальної форми.
* Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

І так сомо можемо дізнатись, що перша нормальна форма (1НФ, 1NF) утворює ґрунт для структурованої схеми бази даних:

* Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.
* Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.
* Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.
  1. Створення ER діаграми

ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.

Модель сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними, такі як: одна будівля може бути розділена на нуль або більше квартир, але одна квартира може бути розташована лише в одній будівлі. Сутності можуть мати різні властивості (атрибути), які характеризують їх. Діаграми, створені для представлення цих сутностей, атрибутів і зв'язків графічно, називають сутність-зв'язок діаграмами.

ER-модель зазвичай реалізується в вигляді баз даних. У разі реляційної бази даних, в якій зберігаються дані в таблицях, кожен рядок кожної таблиці являє собою один екземпляр сутності. Деякі поля даних в цих таблицях вказують на індекси в інших таблицях. Такі поля є покажчиками фізичної реалізації зв'язків між сутностями.

# **Опис програмного забезпечення**

## ER Діаграма

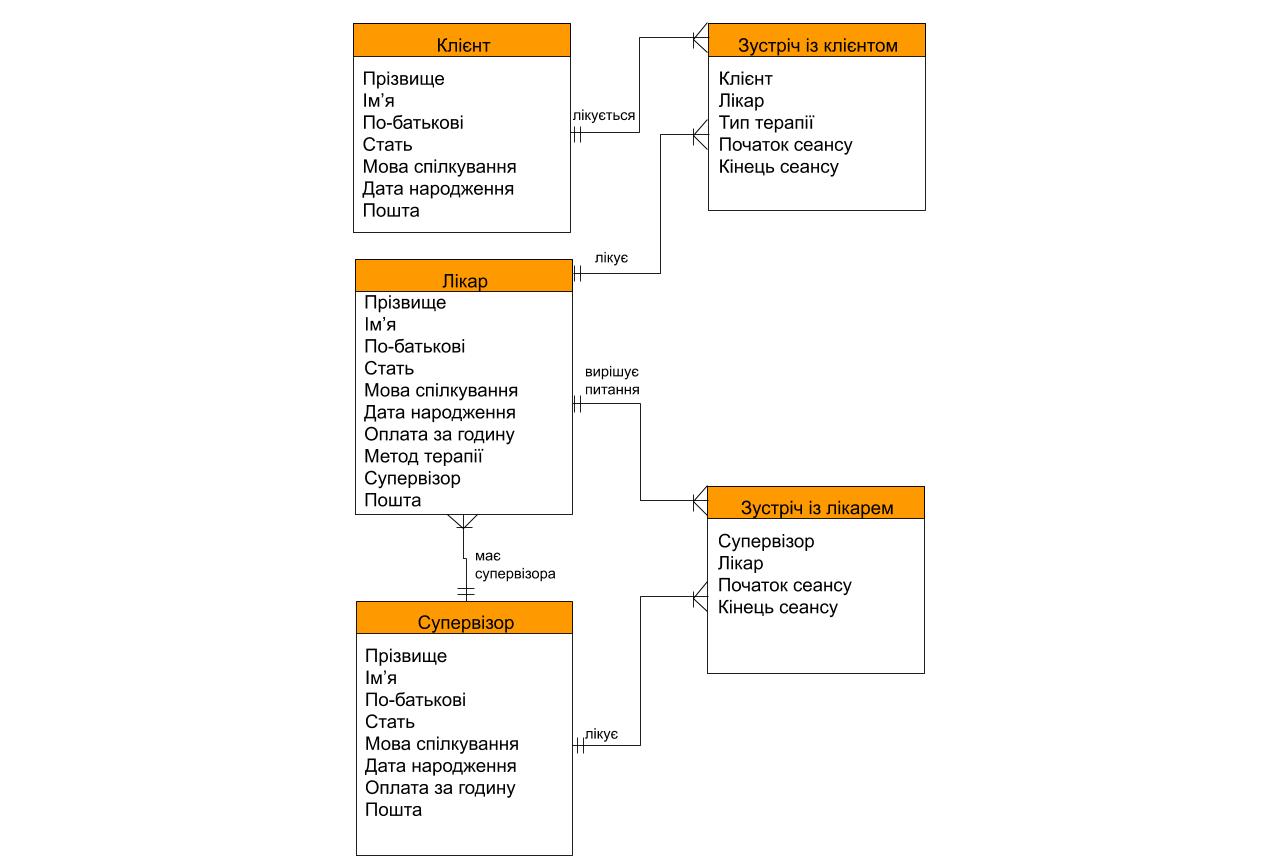


Рисунок 3.1 – ER Діаграма

## Опис сутностей

1. Клієнт
2. Лікар
3. Супервізор
4. Зустріч із клієнтом
5. Зустріч із лікарем

## Фізична модель бази даних

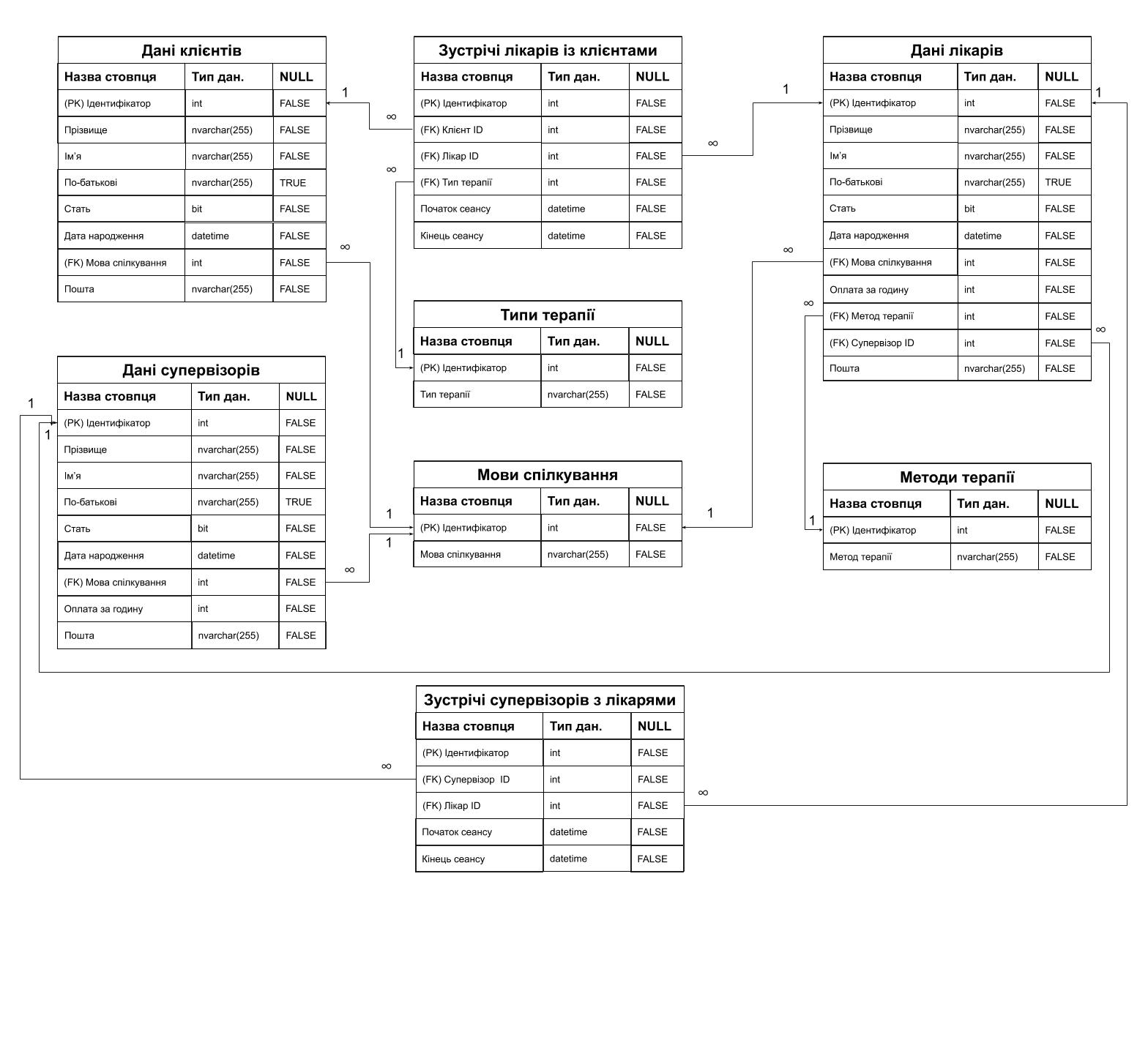


Рисунок 3.2 – Фізична модель бази даних

# **Інструкція користувача**

## Робота з програмою

Після запуску виконавчого файлу з розширенням \*.exe, відкривається головне вікно програми (Рисунок 4.1):

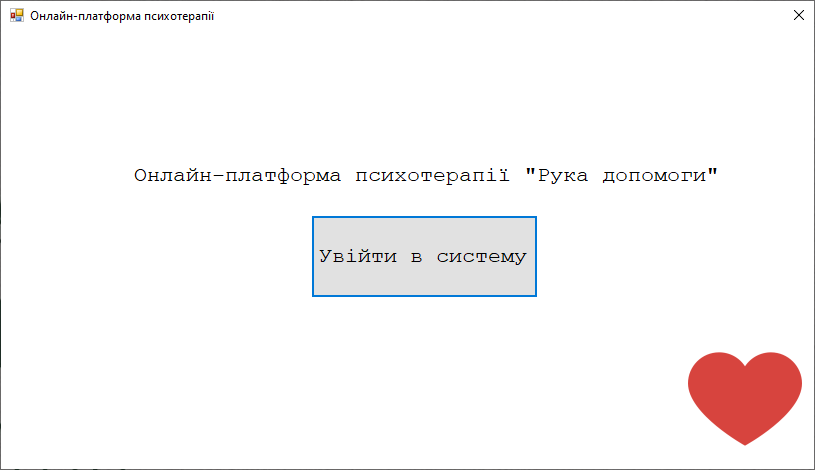


Рисунок 4.1 – Головне вікно програми

Далі користувач повинен натиснути кнопку «Увійти в систему для того, щоб розпочати роботу» (рисунок 4.2):

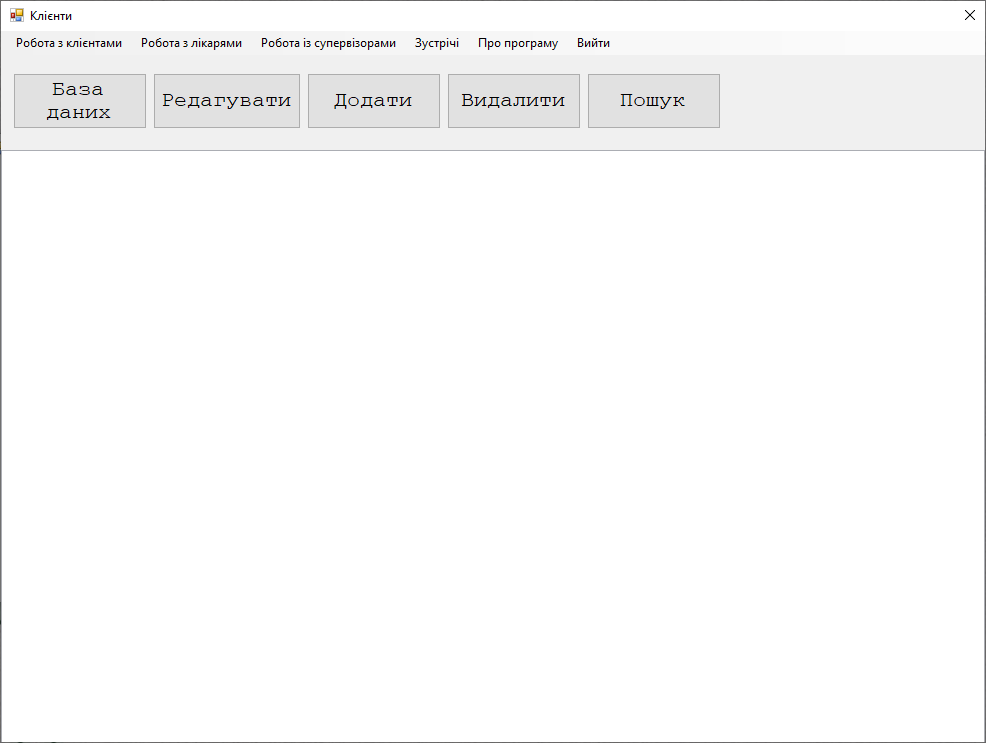


Рисунок 4.2 – Основне меню програми

Користувачу надається можливість обрати подальші дії: на панелі керування можна обрати роботу з клієнтами (ця форма є за замовченням), з лікарями, супервізорами, а також переглянути/створити зустріч з клієнтом/супервізором та невеликий опис програми. Обираємо роботу з клієнтами та завантажуємо базу даних(рисунок 4.3):

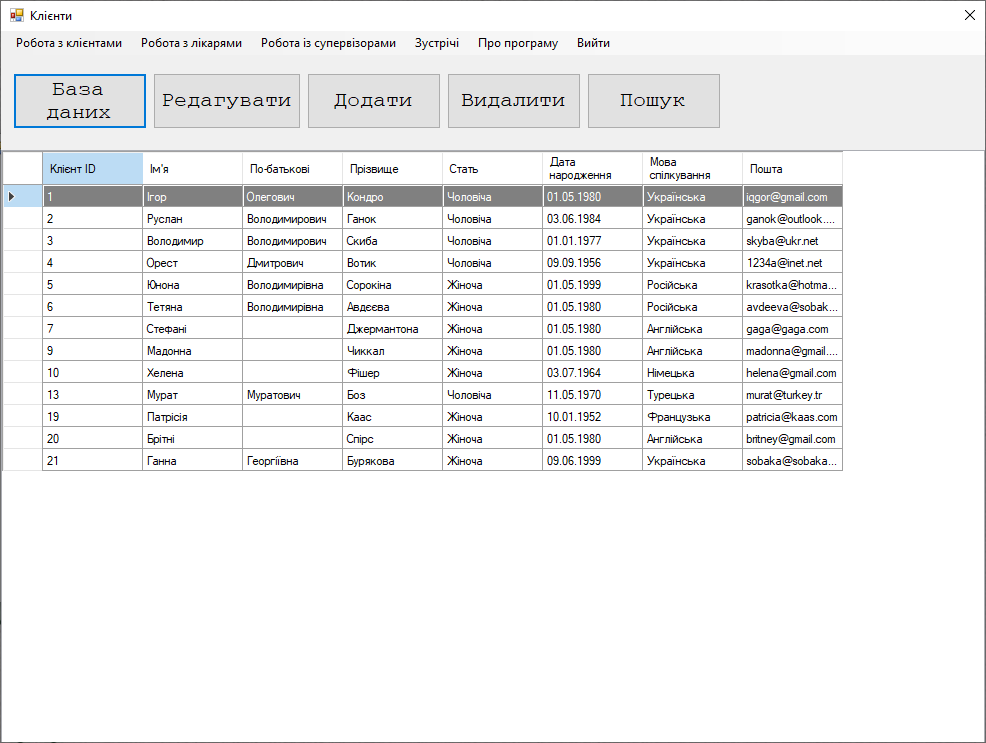


Рисунок 4.3 – Завантажена база даних

Користувач може: редагувати базу, додавати записи, видаляти дані та шукати за ключем клієнтів (рисунки 4.4, 4.5, 4.6, 4.7):

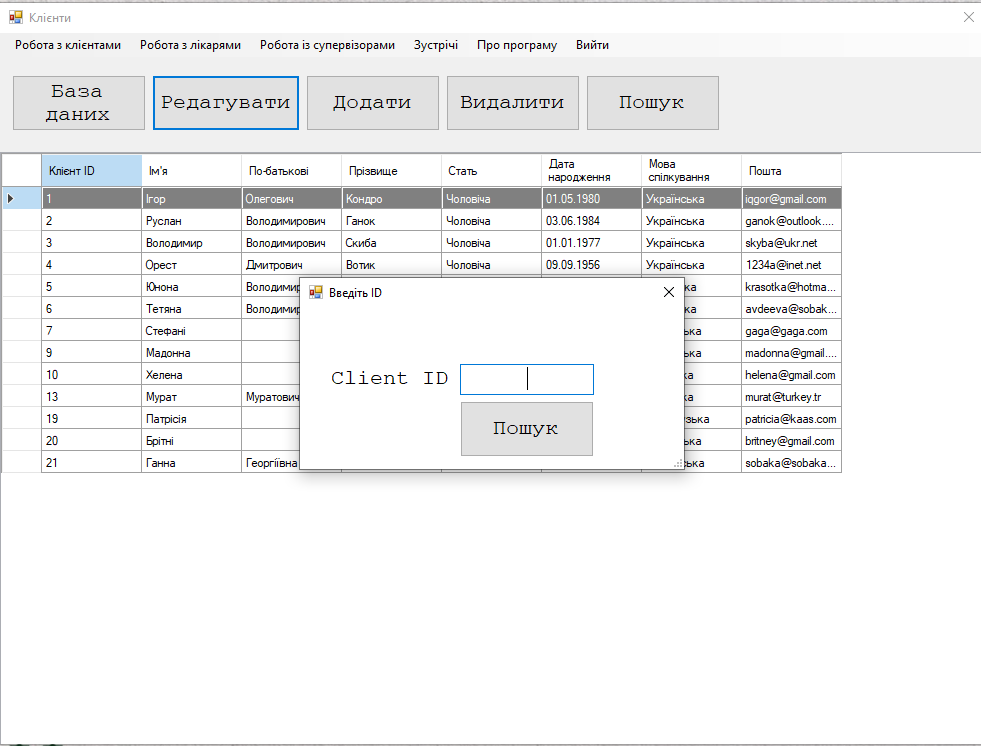


Рисунок 4.4 – Пошук клієнта за ідентифікатором для редагування

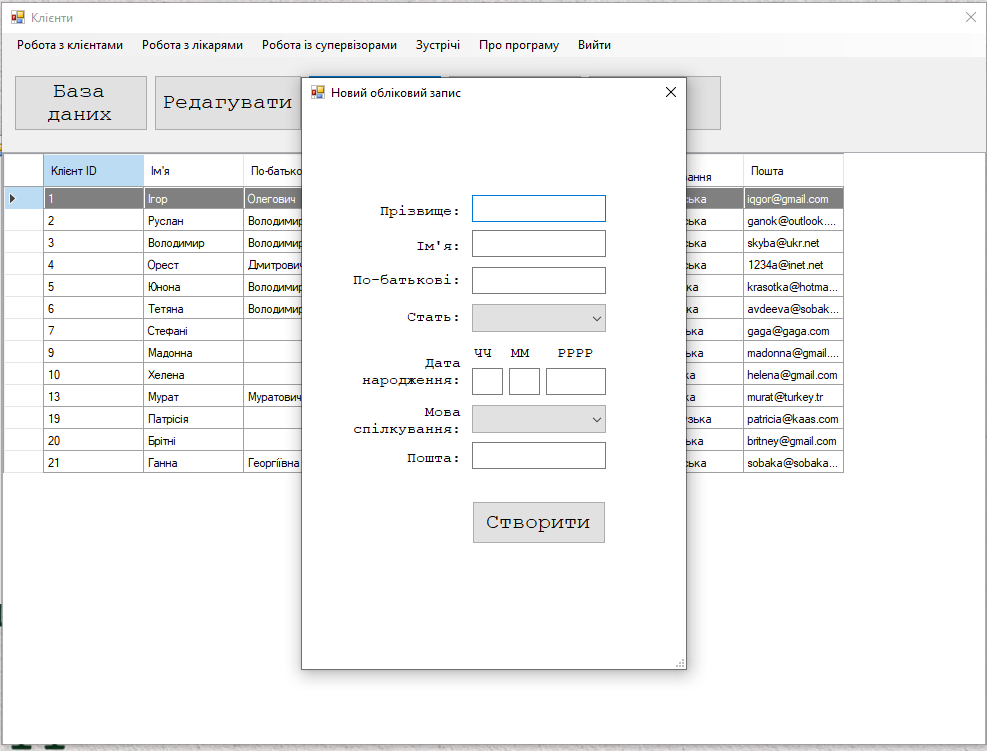


Рисунок 4.5 – Створення нового облікового запису

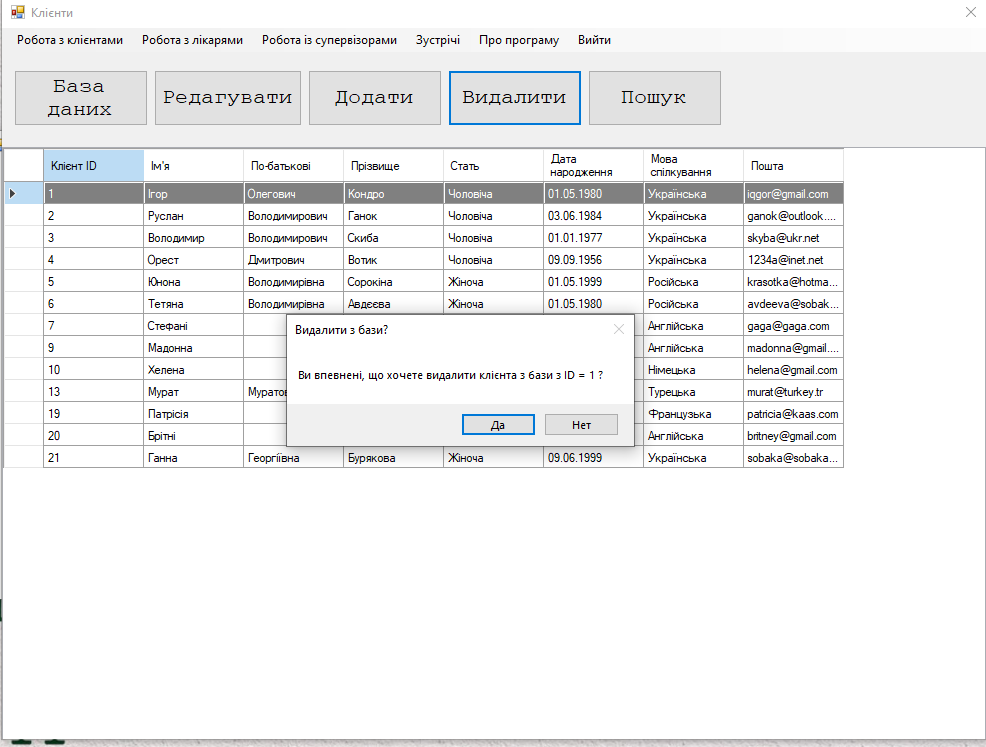


Рисунок 4.6 – Видалення облікового запису з бази даних

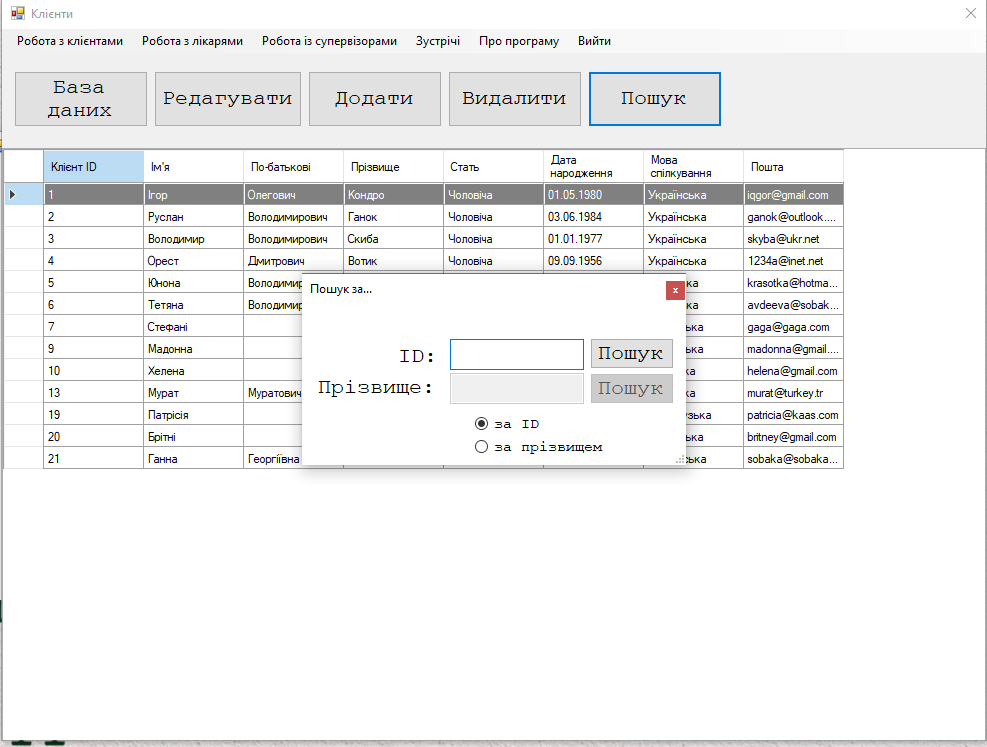


Рисунок 4.7 –Пошук клієнта за ключем: ідентифікатора або прізвищем

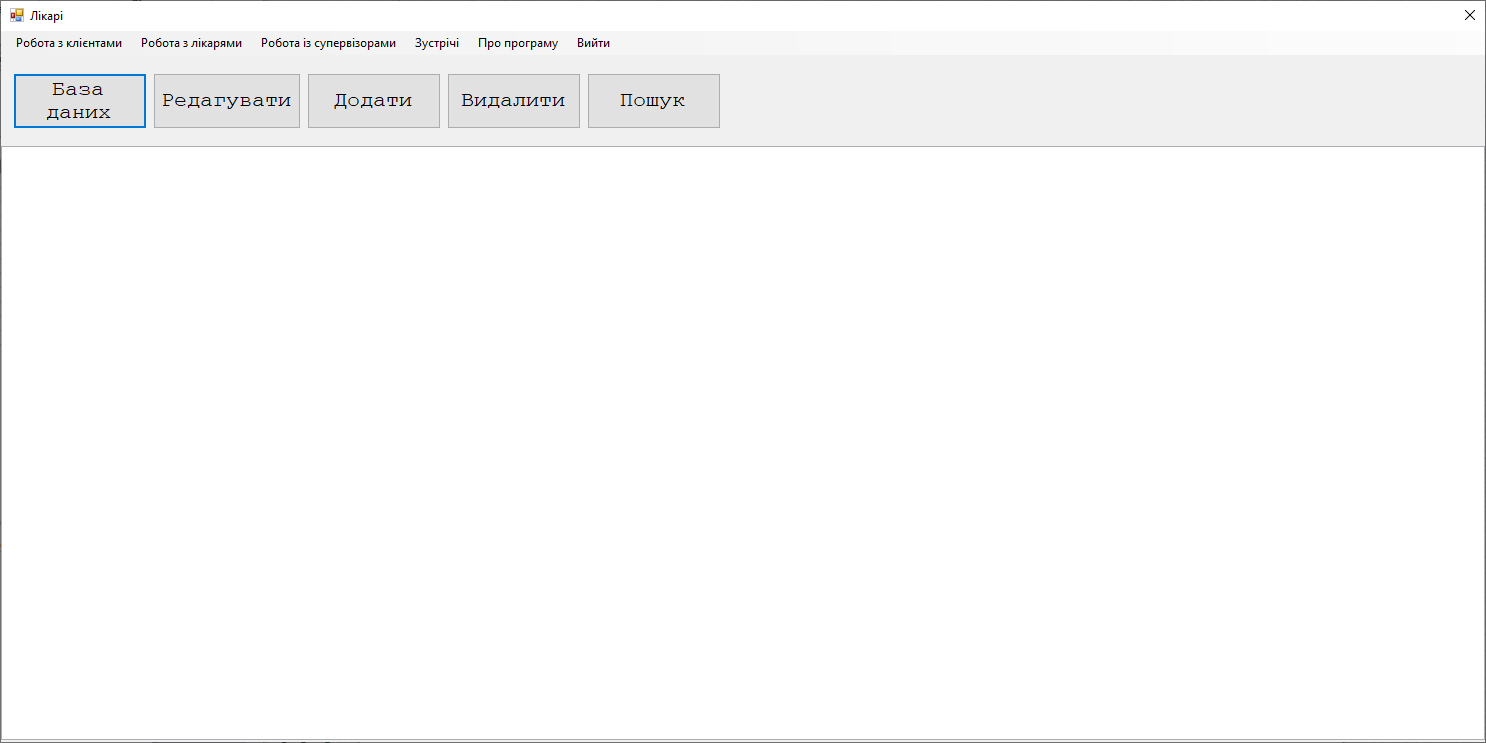
Перейдемо до роботи з лікарями. (рисунок 5.7):

Рисунок 4.8 Форма для роботи з лікарями

Аналогічно до бази даних з клієнтами, лікар має ті ж самі атрибути для роботи, а саме: редагування, додавання, видалення, пошук. Проте сама база даних має інші дані. Користувач може: редагувати базу, додавати записи, видаляти дані та шукати за ключем клієнтів (рисунки 4.9, 4.10, 4.11, 4.12):

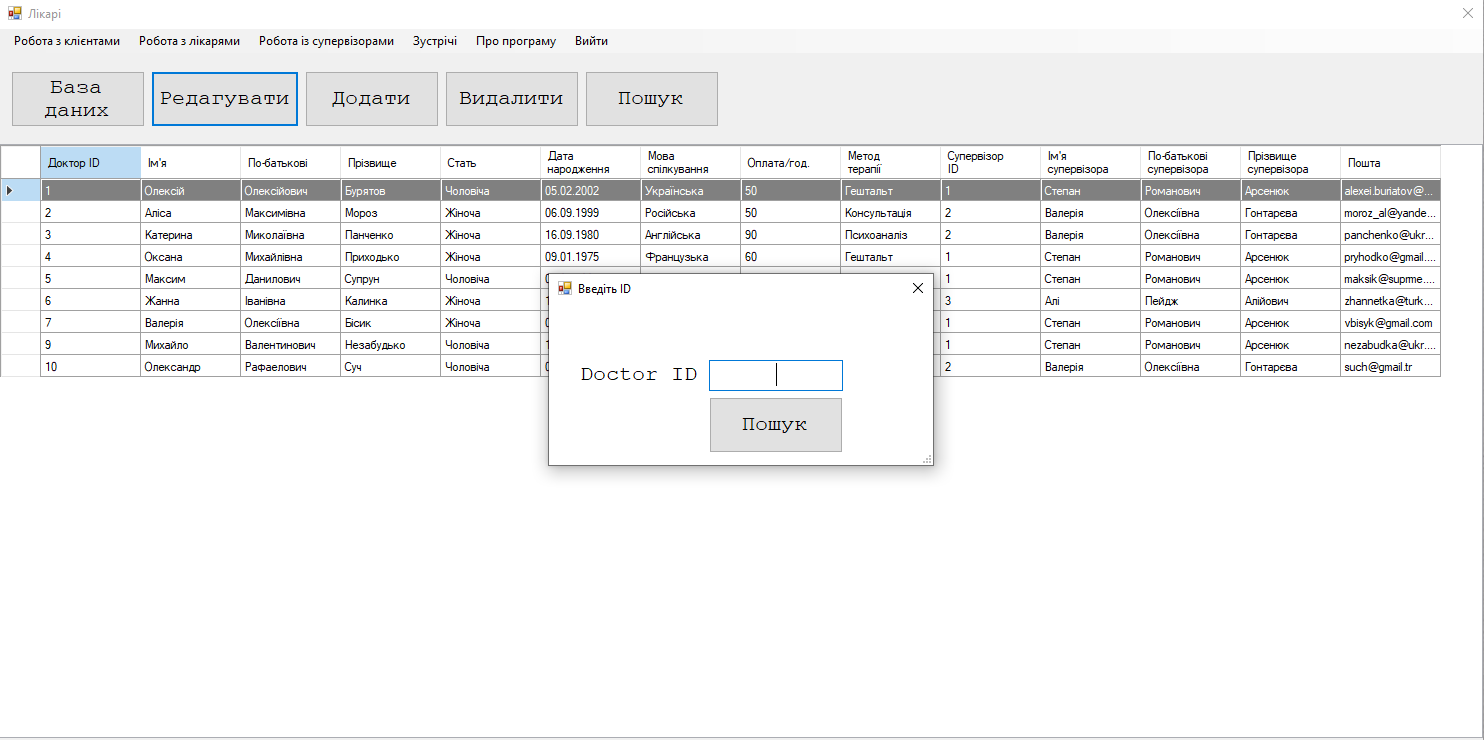


Рисунок 4.9 – Редагування обраного облікового запису лікаря

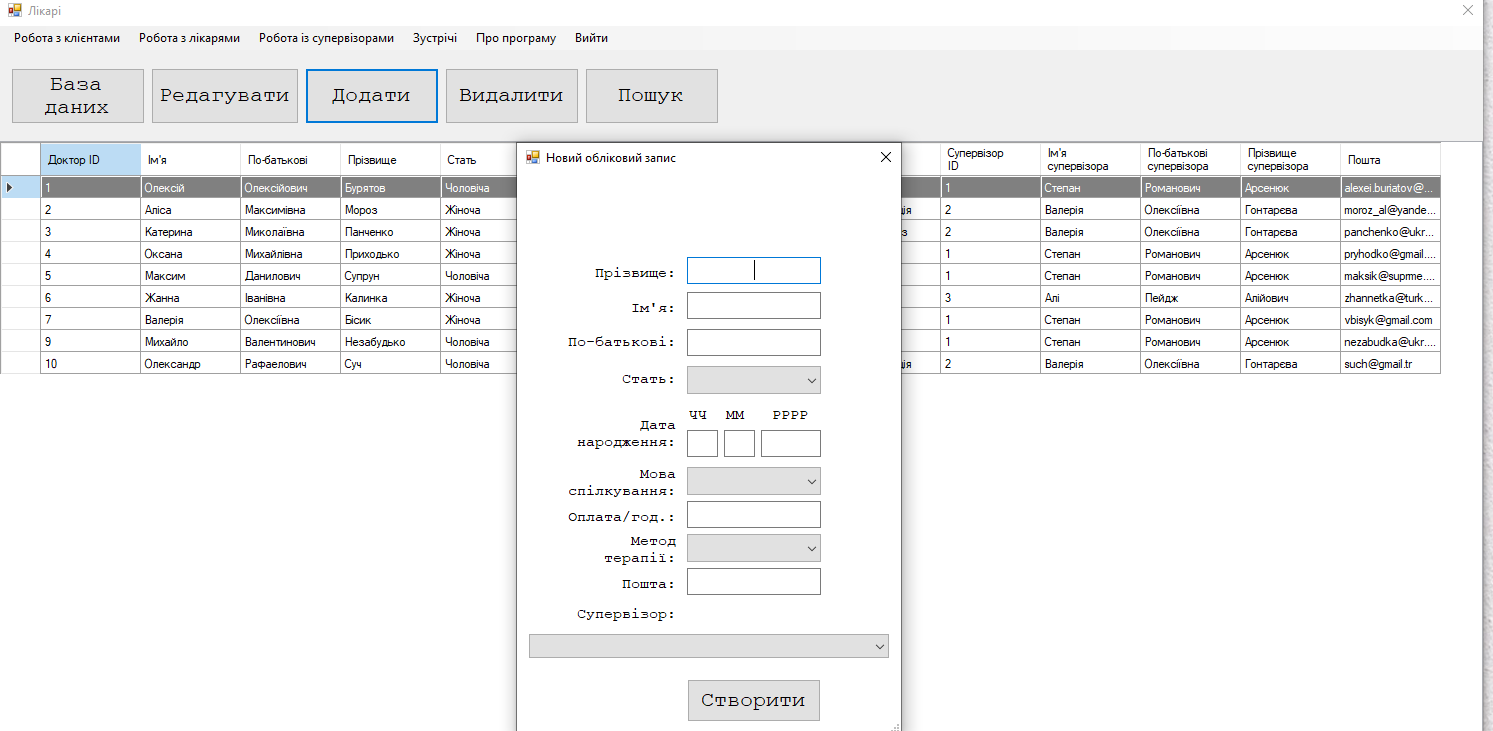
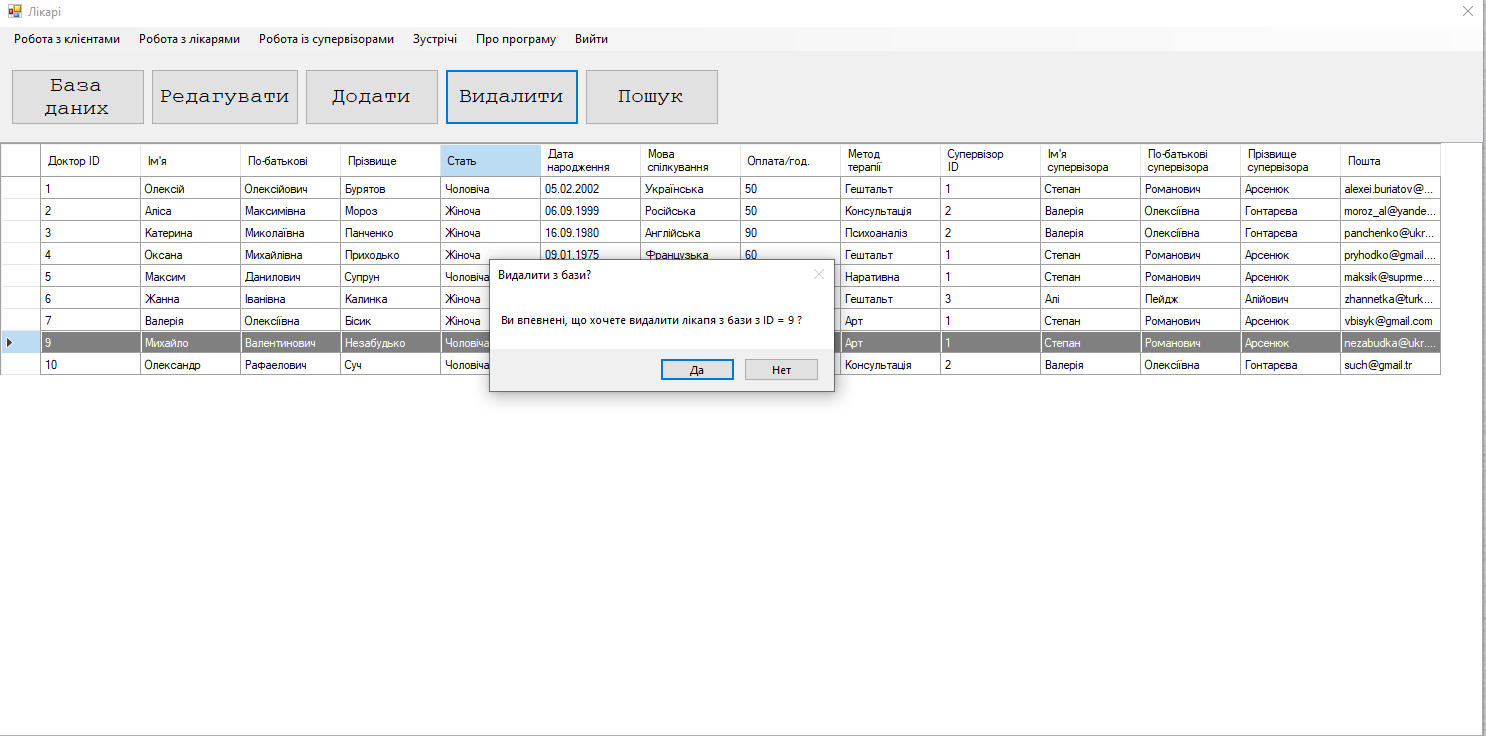


Рисунок 4.10 – Створення нового облікового запису лікаря

Рисунок 4.11 – Видалення наявного облікового запису лікаря

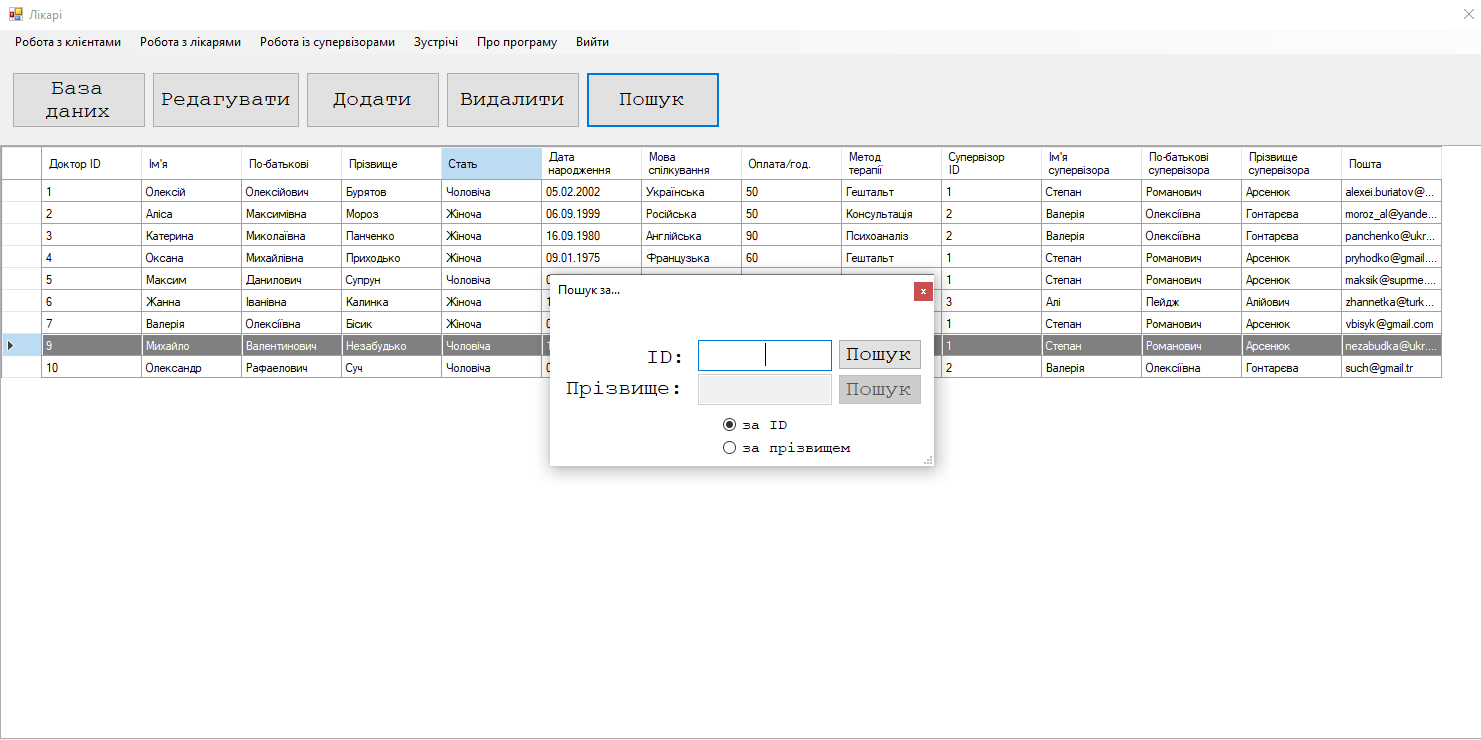


Рисунок 4.12 – Пошук облікового запису за ключовими полями: прізвище/ID

Перейдемо до роботи із супервізорами. (рисунок 4.13):

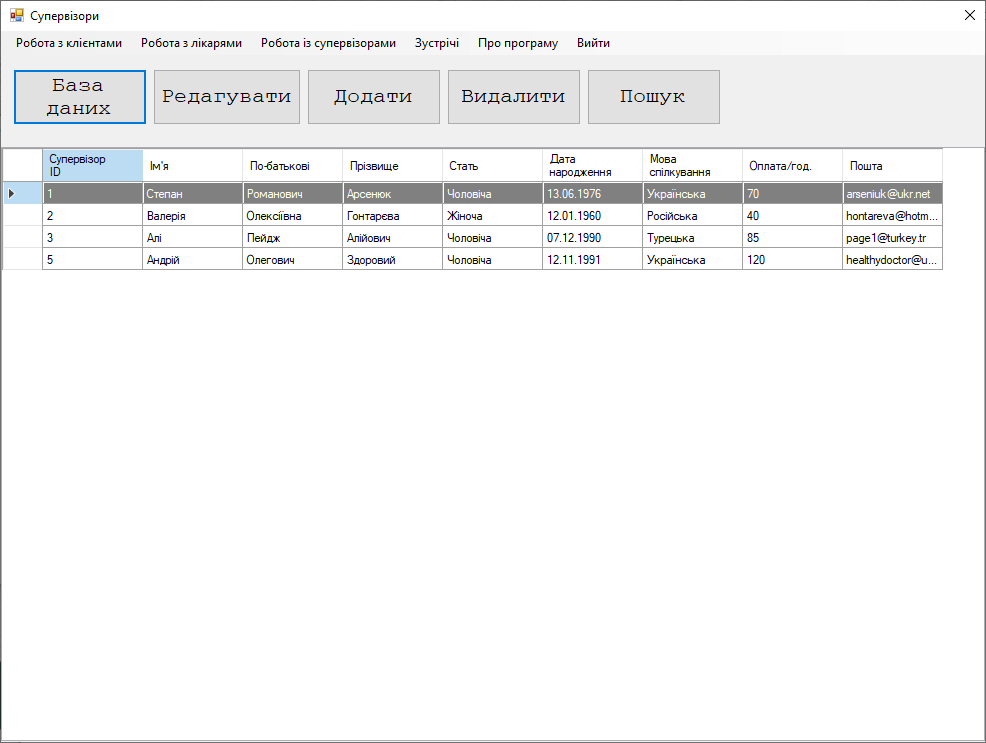


Рисунок 4.13 – Робота із супервізорами. Завантажена база даних

Аналогічно до бази даних з клієнтами та лікарями супервізор має ті ж самі атрибути для роботи, а саме: редагування, додавання, видалення, пошук. Проте сама база даних має інші дані. Користувач може: редагувати базу, додавати записи, видаляти дані та шукати за ключем клієнтів (рисунки 4.14, 4.15, 4.16, 4.17):

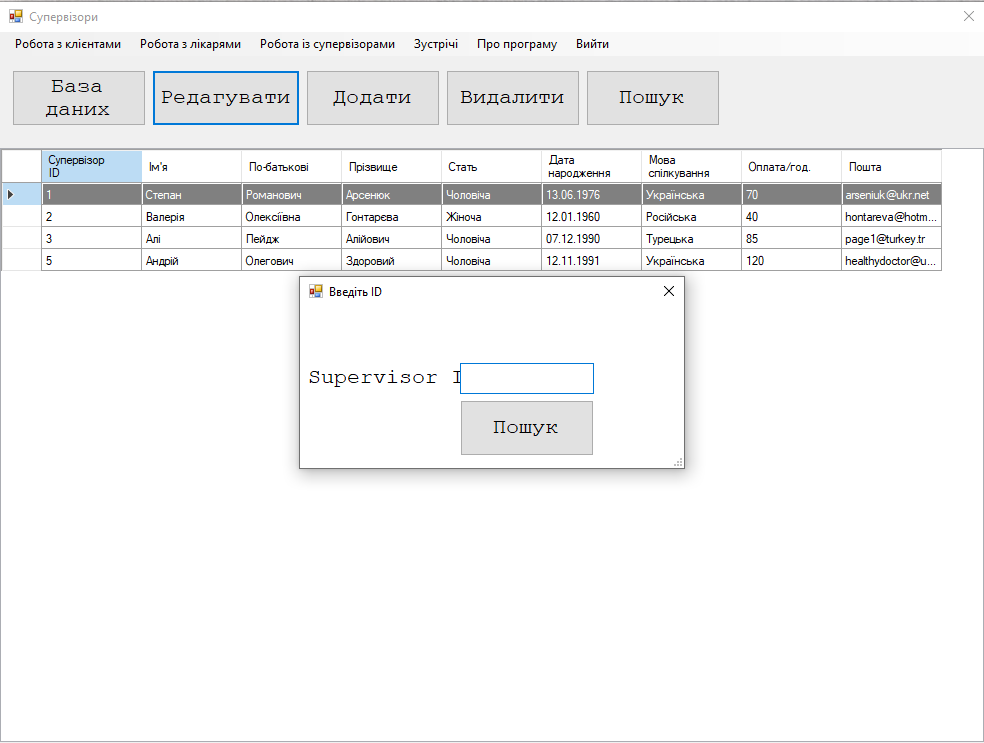


Рисунок 4.14 – Пошук обліку для редагування

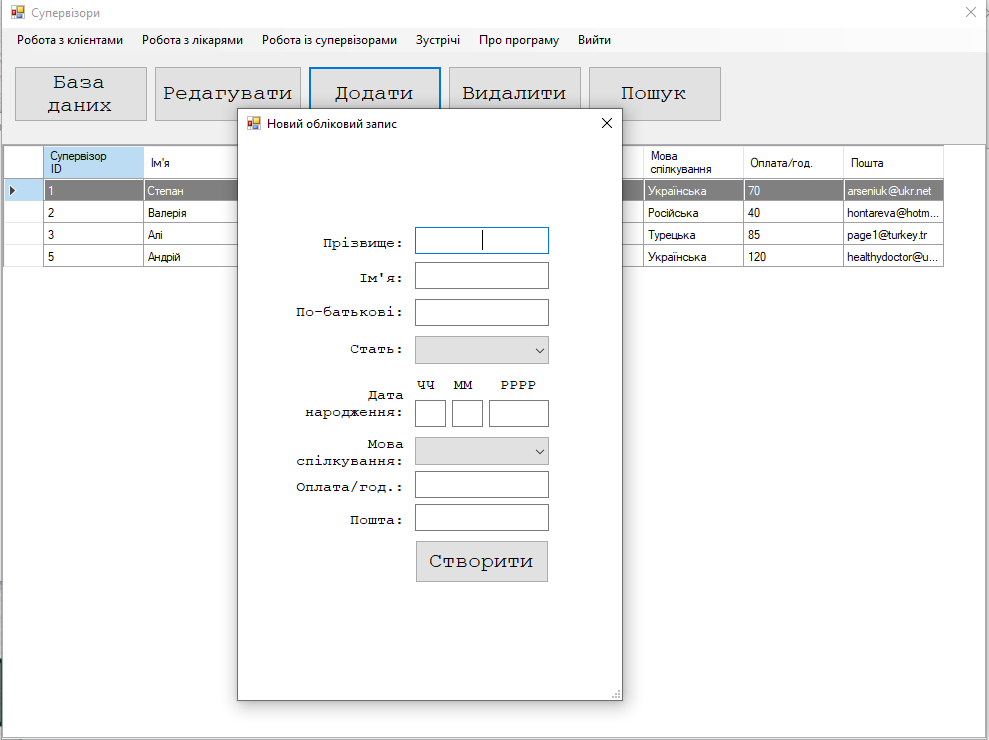


Рисунок 4.15 – Створення нового облікового запису

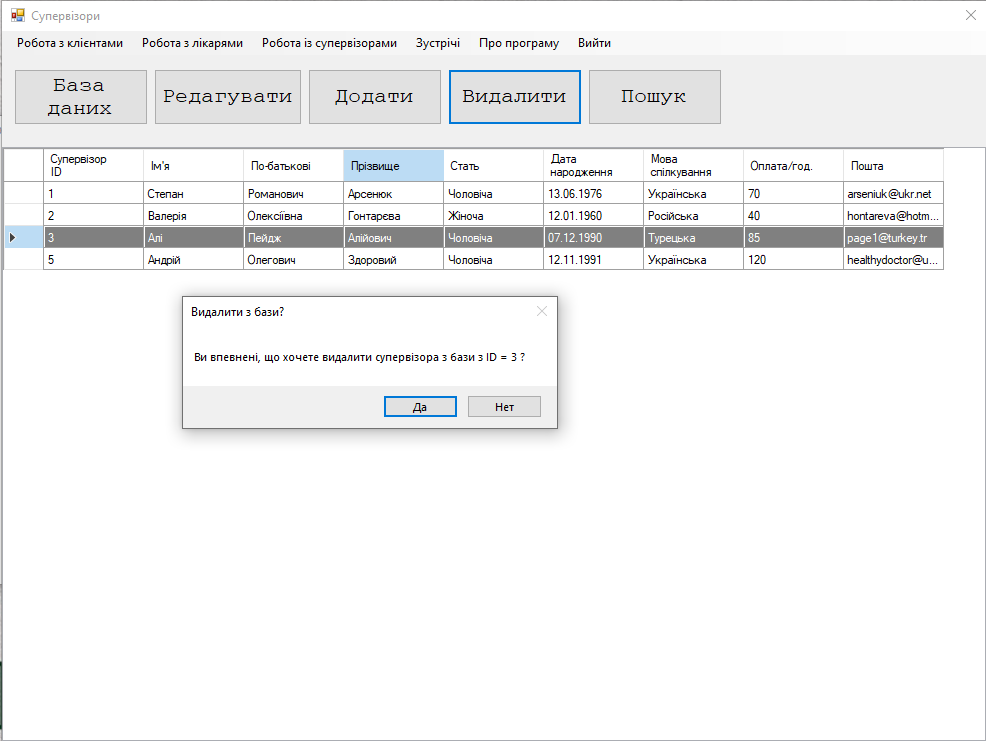


Рисунок 4.16 – Видалення обраного облікового запису

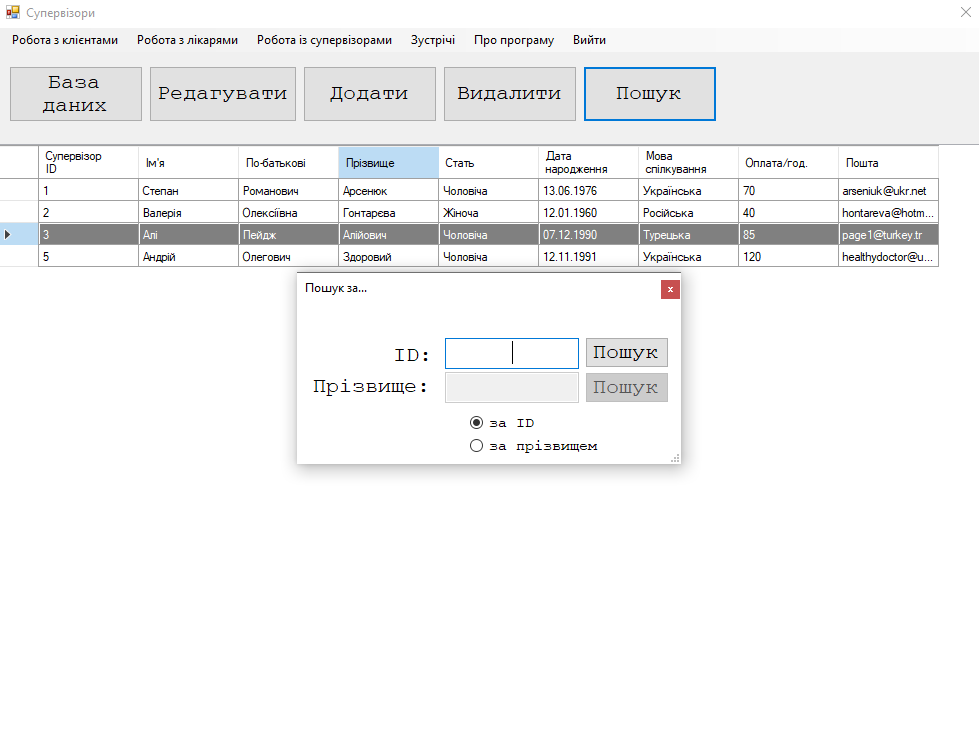


Рисунок 4.17– Пошук облікового запису супервізора за полями: прізвище/ID

Перейдемо до створення записів про зустрічі з лікарями та клієнтами, а також перегляд попередніх зустрічей.

Наприклад, подивимось зустрічі з клієнтами(рисунок 4.18):

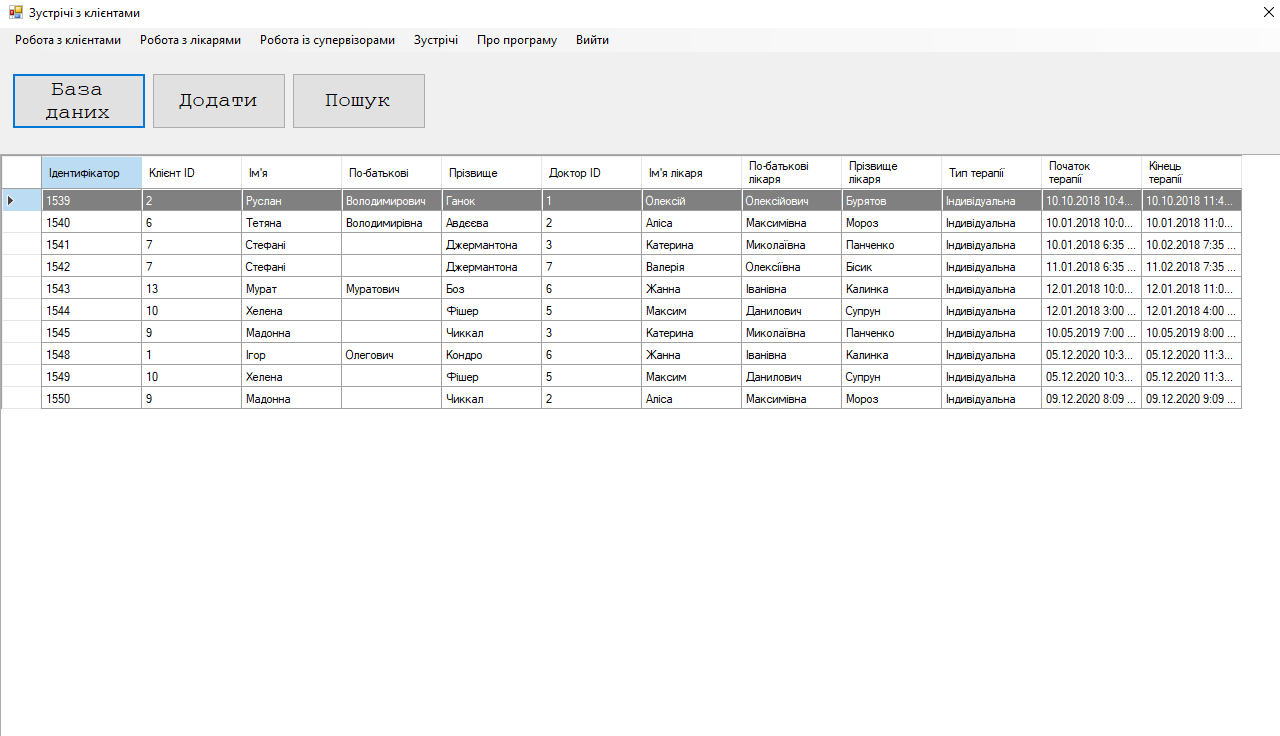


Рисунок 4.18– Завантажена база даних зустрічей з клієнтами

Для того, щоб знайти зустріч, натискаємо на кнопку «Пошук» (рисунок 4.19):

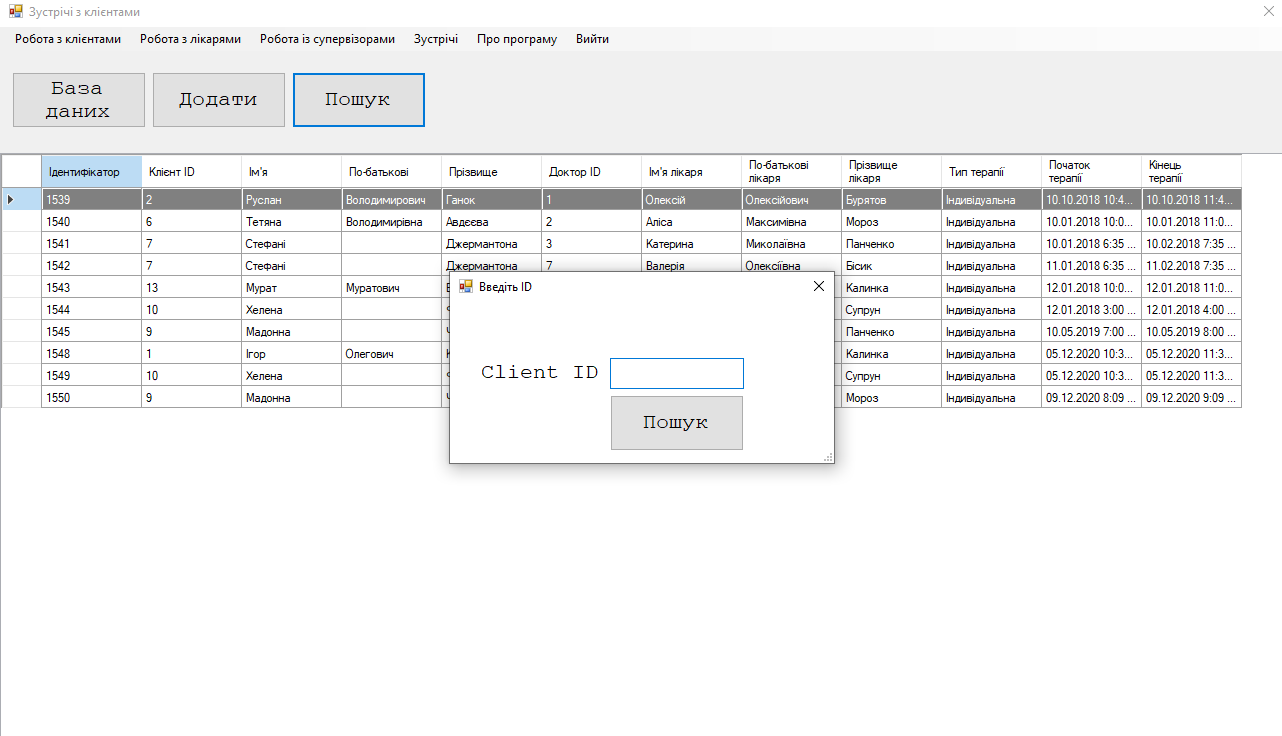


Рисунок 4.19 – Пошук за полем Client ID

А для того, щоб створити зустріч, потрібно натиснути кнопку «Додати» та ввести всі необхідні дані (рисунок 4.20):

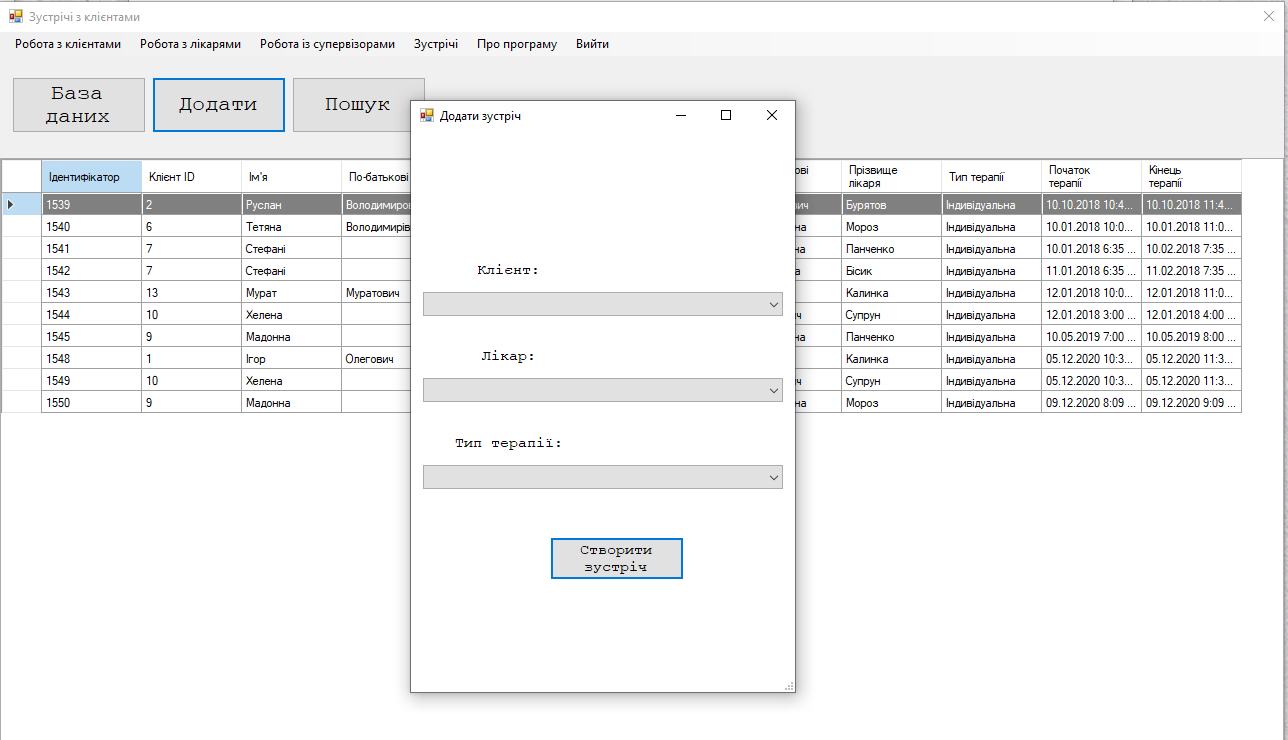


Рисунок 4.20 – Додавання зустрічі

Аналогічно працює і вкладка для лікарів (рисунок 4.21):

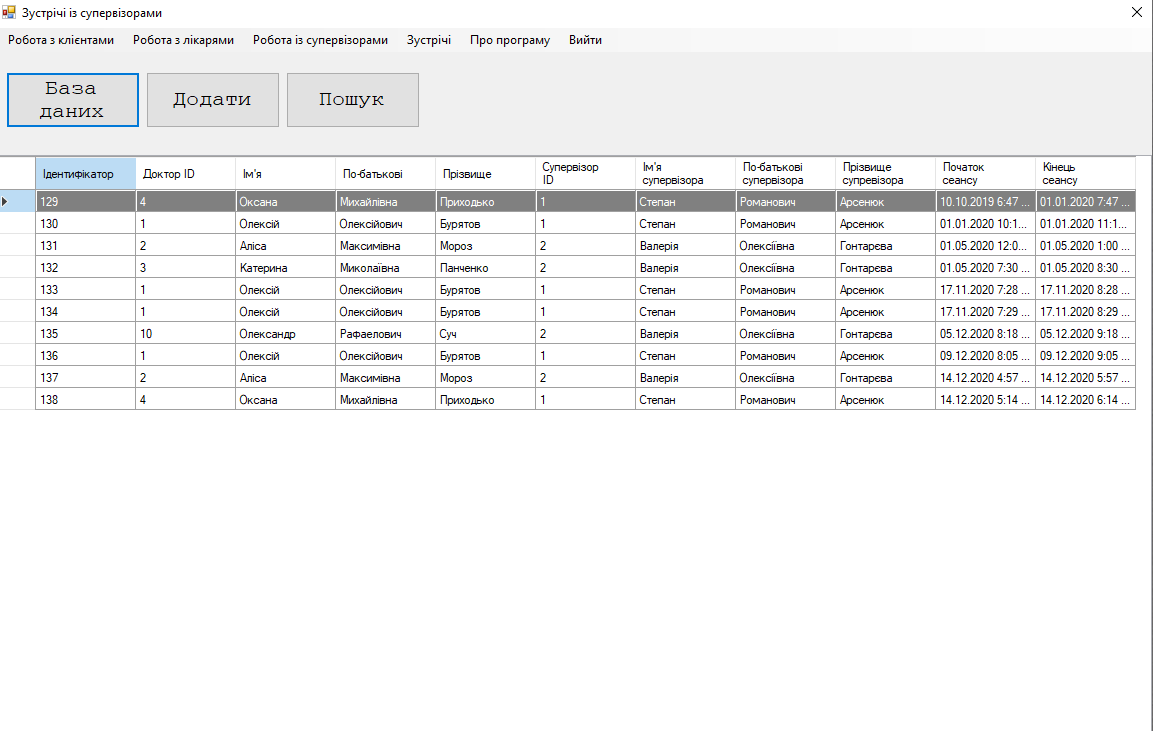


Рисунок 4.21 – Завантажена база даних зустрічей із супервізорами

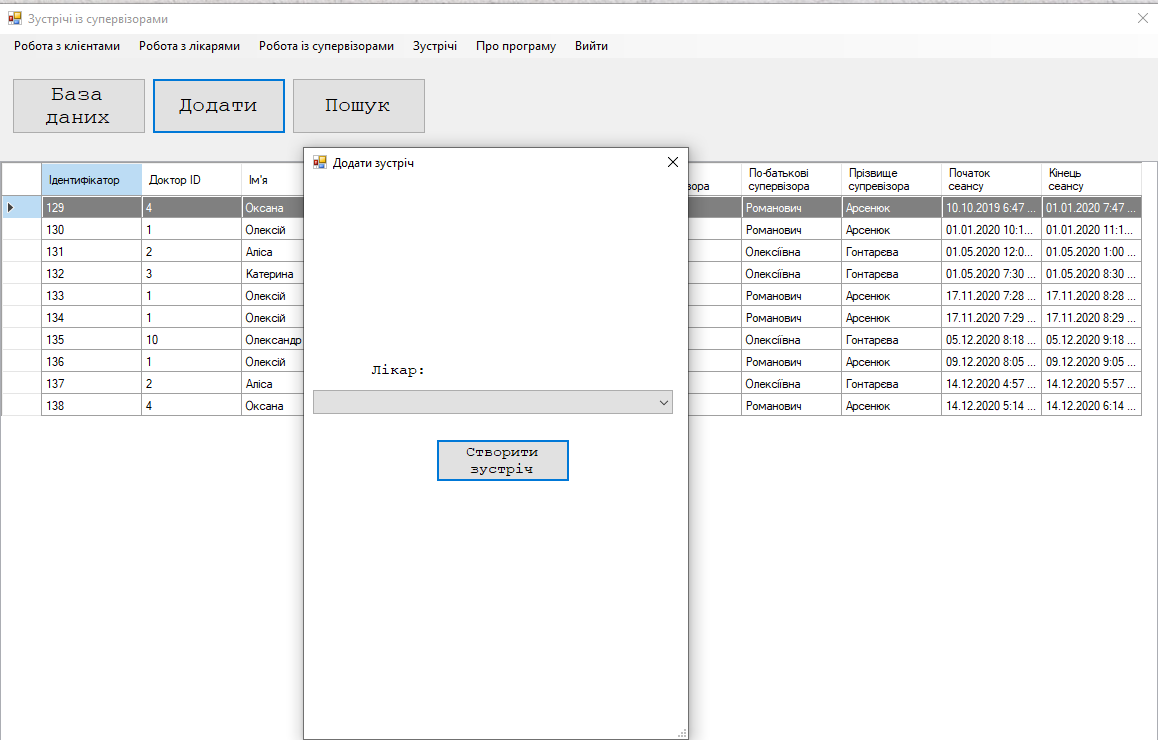


Рисунок 4.22 – Додавання нової зустрічі

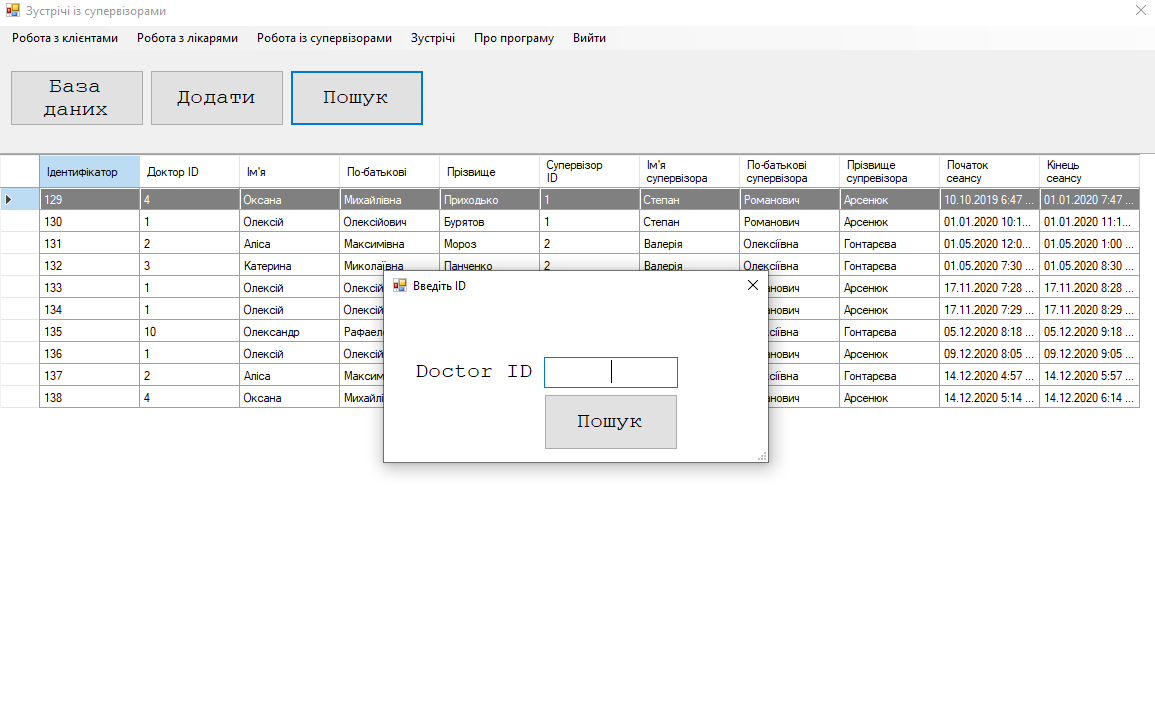


Рисунок 4.23 – Пошук зустрічі за полем DoctorID

## Системні вимоги

Системні вимоги до програмного забезпечення наведені в таблиці 4.1.

Таблиця 4.1 – Системні вимоги програмного забезпечення

|  |  |  |
| --- | --- | --- |
|  | Мінімальні | Рекомендовані |
| Операційна система | Windows® XP/Windows Vista/Windows 7/ Windows 8/Windows 10 (з останніми обновленнями) | Windows 7/ Windows 8/Windows 10  (з останніми обновленнями) |
| Процесор | Intel® Pentium® ІІІ  1.0 GHz або  AMD Athlon™ 1.0 GHz | Intel® Pentium® D або AMD Athlon™ 64 X2 |
| Оперативна пам'ять | 256 MB RAM (для Windows® XP) / 1 GB RAM (для Windows Vista/Windows 7/  Windows 8/Windows 10) | 2 GB RAM |
| Відеоадаптер | Intel GMA 950 з відеопам'яттю об'ємом не менше 64 МБ (або сумісний аналог) | |
| Дисплей | 800х600 | 1024х768 або краще |
|  | Мінімальні | Рекомендовані | |
| Прилади введення | Клавіатура, комп’ютерна миша | | |
| Додаткове програмне забезпечення | Microsoft .Net Framework 4.5.2 або вище | | |

# **Перевірка на коректність**

* 1. План тестування

Зрозуміло, що користувач може випадково (або невипадково) ввести некоректні дані. Для цього розробник повинен передбачити хід можливих дій та забезпечити максимально свій додаток від нестабільної роботи. У рамках даної курсової роботи були розроблені ряд перевірок, які сприяють покращенню роботи.

1. Тестування правильності введеної ідентифікатора
   1. Тестування при відсутності назви
   2. Тестування при відсутності правильного значення
2. Тестування правильності введених значень.
   1. Тестування при введенні замалих та завеликих значень.
   2. Приклади тестування

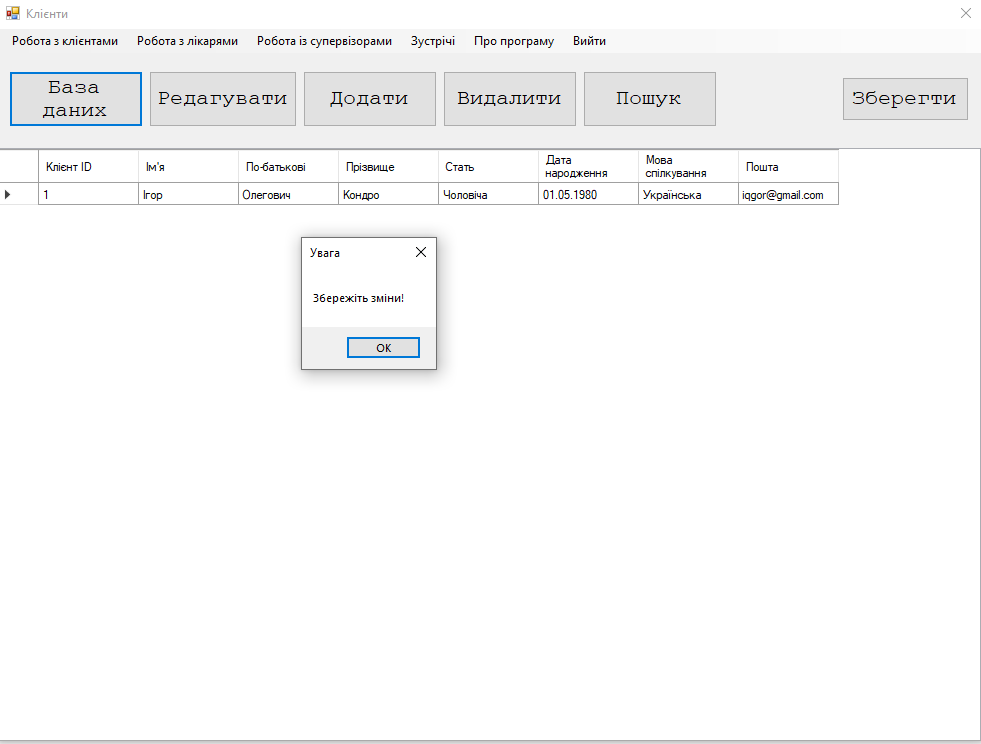
Для перевірки на коректність було створено чимало діалогових вікон, які попереджують користувача про ту чи іншу помилку. Програма не буде працювати допоки користувач не введе скориговані дані або користувач не замінить їх на інші.

Рисунок 5.1 – Попередження

# **Аналіз і узагальнення результатів**

Головною задачею курсової роботи була реалізація програми для створення онлайн-платформи психотерапії «Рука допомоги».

Критичні ситуації у роботі програми виявлені не були. Під час тестування було виявлено, що більшість помилок виникало тоді, коли користувачем вводилися замалі дані або вони були взагалі відсутні. Тому всі дані, які вводить користувач, ретельно провіряються на коректність і лише потім подаються на обробку програмі.

Упродовж виконання курсової роботи були поглиблені знання з базами даних, а саме MSSQL. Було створені сутності: клієнт, лікар, супервізор, зустрічі клієнтів із лікарями та зустрічі лікарів із супервізорами. У базі даних є такі таблиці: дані клієнтів, дані лікарів, дані супервізорів, зустрічі з клієнтами, зустрічі з супервізорами, типи терапії, методи терапії та мови спілкування.

Створений застосунок може використовуватися для будь-якої початкової психотерапевтичної служби, адже в подальшому цей застосунок буде модифікований, бо є в планах створення облікових записів у самому застосунку, а також розширення можливостей психотерапії, а саме: лікування не тільки в індивідуальному характері, а також буде передбачено групові заняття, ігрові заняття (підвид групової терапії, коли конфліктуючі сторони у формі гри пізнають одне одного, співпереживають та налагоджують відносини).

Окрім поглиблення знань мови C#, також були отримані навички проектування ER діаграм та фізичної бази даних.

Мета роботи була досягнута, всі поставлені задачі були виконані.

Додаток А Технічне завдання

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

автоматизованих систем обробки інформації та управління

Затвердив

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_ р.

Виконавець:

Студент Бурятов Олексій Олексійович

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання курсової роботи

на тему: ІС «Онлайн-платформа психотерапії»

з дисципліни:

«Бази даних-1»

Київ 2020

Додаток Б Тексти програмного коду

CREATE DATABASE PsychologyOnline

COLLATE Cyrillic\_General\_CI\_AS

GO

USE PsychologyOnline

GO

CREATE TABLE TherapyTypes

(

TherapyID int PRIMARY KEY NOT NULL IDENTITY(1,1),

TherapyType nvarchar(255) NOT NULL

);

GO

CREATE TABLE TherapyMethods

(

TherapyID int PRIMARY KEY NOT NULL IDENTITY(1,1),

TherapyMethod nvarchar(255) NOT NULL

);

GO

CREATE TABLE LanguageNames

(

LanguageID int PRIMARY KEY NOT NULL IDENTITY(1,1),

LanguageName nvarchar(35) NOT NULL

);

GO

CREATE TABLE DataClients

(

ClientID int PRIMARY KEY NOT NULL IDENTITY(1,1),

ClientLName nvarchar(255) NOT NULL,

ClientFName nvarchar(255) NOT NULL,

ClientPatronymic nvarchar(255) NULL,

ClientSex BIT NOT NULL,

ClientDateOfBirth date NOT NULL,

ClientLanguage int NOT NULL,

ClientMail nvarchar(255) NOT NULL,

CONSTRAINT FK\_DataClients\_To\_LanguageNames FOREIGN KEY (ClientLanguage)

REFERENCES LanguageNames (LanguageID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

);

GO

ALTER TABLE DataClients

ADD CONSTRAINT FK\_DataClients\_To\_LanguageNames FOREIGN KEY (ClientLanguage)

REFERENCES LanguageNames (LanguageID)

ON DELETE CASCADE

ON UPDATE NO ACTION

CREATE TABLE DataSupervs

(

SupervID int PRIMARY KEY NOT NULL IDENTITY(1,1),

SupervLName nvarchar(255) NOT NULL,

SupervFName nvarchar(255) NOT NULL,

SupervPatronymic nvarchar(255) NULL,

SupervSex BIT NOT NULL,

SupervDateOfBirth date NOT NULL,

SupervLanguage int NOT NULL,

HourlyRate int NOT NULL,

SupervMail nvarchar(255) NOT NULL,

CONSTRAINT FK\_DataSupervs\_To\_LanguageNames FOREIGN KEY (SupervLanguage)

REFERENCES LanguageNames (LanguageID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

);

GO

CREATE TABLE DataDoctors

(

DoctorID int PRIMARY KEY NOT NULL IDENTITY(1,1),

DoctorLName nvarchar(255) NOT NULL,

DoctorFName nvarchar(255) NOT NULL,

DoctorPatronymic nvarchar(255) NULL,

DoctorSex BIT NOT NULL,

DoctorDateOfBirth date NOT NULL,

DoctorLanguage int NOT NULL,

HourlyRate int NOT NULL,

TherapyMethod int NOT NULL,

SupervisorID int NOT NULL,

DoctorMail nvarchar(255) NOT NULL,

CONSTRAINT FK\_DataDoctors\_To\_LanguageNames FOREIGN KEY (DoctorLanguage)

REFERENCES LanguageNames (LanguageID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT FK\_DataDoctors\_To\_DataSupervs FOREIGN KEY (SupervisorID)

REFERENCES DataSupervs (SupervID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT FK\_DataDoctors\_To\_TherapyMethods FOREIGN KEY (TherapyMethod)

REFERENCES TherapyMethods (TherapyID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

);

GO

CREATE TABLE Meetings

(

Identifier int PRIMARY KEY NOT NULL IDENTITY(1539,1),

ClientID int NOT NULL,

DoctorID int NOT NULL,

TherapyType int NOT NULL,

SessionBegin datetime NOT NULL,

SessionEnd datetime NOT NULL,

CONSTRAINT FK\_Meetings\_To\_DataClients FOREIGN KEY (ClientID)

REFERENCES DataClients (ClientID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT FK\_Meetings\_To\_DataDoctors FOREIGN KEY (DoctorID)

REFERENCES DataDoctors (DoctorID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT FK\_Meetings\_To\_TherapyTypes FOREIGN KEY (TherapyType)

REFERENCES TherapyTypes (TherapyID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

);

GO

CREATE TABLE SupAnDocMeetings

(

Identifier int PRIMARY KEY NOT NULL IDENTITY(129,1),

SupervisorID int NOT NULL,

DoctorID int NOT NULL,

SessionBegin datetime NOT NULL,

SessionEnd datetime NOT NULL,

CONSTRAINT FK\_SupAnDocMeetings\_To\_DataSupervs FOREIGN KEY (SupervisorID)

REFERENCES DataSupervs (SupervID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT FK\_SupAnDocMeetings\_To\_DataDoctors FOREIGN KEY (DoctorID)

REFERENCES DataDoctors (DoctorID)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

);

GO

INSERT INTO LanguageNames

VALUES

('Українська'),

('Російська'),

('Англійська'),

('Французька'),

('Німецька'),

('Турецька')

;

GO

INSERT INTO TherapyTypes

VALUES

('Індивідуальна')

;

GO

INSERT INTO TherapyMethods

VALUES

('Консультація'),

('Арт'),

('Гештальт'),

('Наративна'),

('Психоаналіз')

;

GO

INSERT INTO DataSupervs

VALUES

('Арсенюк','Степан','Романович',1,'1976-06-13',1,70,'arseniuk@ukr.net'),

('Гонтарєва','Валерія','Олексіївна',0,'1960-01-12',2,40,'hontareva@hotmail.com'),

('Пейдж','Алі',NULL,1,'1990-12-07',6,80,'page@turkey.tr')

;

GO

INSERT INTO DataDoctors

VALUES

('Бурятов','Олексій','Олексійович',1,'2002-02-05',1,50,3,1,'alexei.buriatov@gmail.com'),

('Мороз','Аліса','Максимівна',0,'1999-09-06',2,50,1,2,'moroz\_al@yandex.ru'),

('Панченко','Катерина','Миколаївна',0,'1980-09-16',3,90,5,2,'panchenko@ukr.net'),

('Приходько','Оксана','Михайлівна',0,'1975-01-09',4,60,3,1,'pryhodko@gmail.com'),

('Супрун','Максим','Данилович',1,'2001-09-05',5,49,4,1,'maksik@suprme.com'),

('Калинка','Жанна','Іванівна',0,'1976-06-13',6,150,3,3,'zhannetka@turkey.tr'),

('Бісик','Валерія','Олексіївна',0,'1999-09-06',3,50,2,1,'bisyk@gmail.com')

;

GO

INSERT INTO DataClients

VALUES

('Кондро','Ігор','Володимирович',1,'1980-05-01',1,'igor@gmail.com'),

('Ганок','Руслан','Володимирович',1,'1984-06-03',1,'ganok@outlook.com'),

('Скиба','Володимир','Володимирович',1,'1977-01-01',1,'skyba@ukr.net'),

('Вотик','Орест','Дмитрович',1,'1956-09-09',1,'134a@inet.net'),

('Сорокіна','Юнона','Володимирівна',0,'1999-05-01',2,'krasotka@hotmail.com'),

('Авдєєва','Тетяна','Володимирівна',0,'1980-05-01',2,'avdeeva@sobaka.net'),

('Джермантона','Стефані',NULL,0,'1980-05-01',3,'gaga@gaga.com'),

('Спірс','Брітні',NULL,0,'1980-05-01',3,'britney@gmail.com'),

('Чиккал','Мадонна',NULL,0,'1980-05-01',3,'madonna@gmail.com'),

('Фішер','Хелена',NULL,0,'1964-07-03',5,'helena@gmail.com'),

('Авер','Поль','Ван',1,'1997-05-02',4,'aver@gmail.com'),

('Каас','Патрісія',NULL,0,'1952-01-10',4,'patricia@kaas.com'),

('Боз','Мурат',NULL,1,'1970-05-11',6,'murat@turkey.tr');

GO

INSERT INTO SupAnDocMeetings

VALUES

(1,4,'2019-10-10 18:47:00','2020-01-01 19:47:00'),

(1,1,'2020-01-01 10:10:00','2020-01-01 11:10:00'),

(2,2,'2020-05-01 12:00:00','2020-05-01 13:00:00'),

(2,3,'2020-05-01 19:30:00','2020-05-01 20:30:00')

;

GO

INSERT INTO Meetings

VALUES

(2,1,1,'2018-10-10 10:47:00','2018-10-10 11:47:00'),

(6,2,1,'2018-01-10 10:00:00','2018-01-10 11:00:00'),

(7,3,1,'2018-01-10 18:35:00','2018-02-10 19:35:00'),

(7,7,1,'2018-01-11 18:35:00','2018-02-11 19:35:00'),

(13,6,1,'2018-01-12 10:00:00','2018-01-12 11:00:00'),

(10,5,1,'2018-01-12 15:00:00','2018-01-12 16:00:00'),

(9,3,1,'2019-05-10 07:00:00','2019-05-10 08:00:00')

;

GO

CREATE PROCEDURE SlctDoctorsByLName

@LastN NVARCHAR(255)

AS

SELECT DISTINCT dd.DoctorID AS'Доктор ID',

dd.DoctorFName AS 'Ім''я',

dd.DoctorPatronymic AS 'По-батькові',

dd.DoctorLName AS 'Прізвище',

[dbo].gender(dd.DoctorSex) AS 'Стать',

dd.DoctorDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

dd.HourlyRate AS 'Оплата/год.',

tm.TherapyMethod AS 'Метод терапії',

ds.SupervID 'Супервізор ID',

ds.SupervFName 'Прізвище',

ds.SupervLName 'Ім''я',

ds.SupervPatronymic 'По-батькові',

dd.DoctorMail AS 'Пошта'

FROM DataDoctors dd

JOIN LanguageNames ln ON ln.LanguageID=dd.DoctorLanguage

JOIN TherapyMethods tm ON dd.TherapyMethod=tm.TherapyID

JOIN DataSupervs ds ON dd.SupervisorID= ds.SupervID

WHERE dd.DoctorLName=@LastN

;

GO

CREATE PROCEDURE SlctSupervByLName

@LastN NVARCHAR(255)

AS

SELECT ds.SupervID 'Супервізор ID',

ds.SupervLName 'Ім''я',

ds.SupervPatronymic 'По-батькові',

ds.SupervFName 'Прізвище',

[dbo].gender(ds.SupervSex) AS 'Стать',

ds.SupervDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ds.HourlyRate AS 'Оплата/год.',

ds.SupervMail AS 'Пошта'

FROM DataSupervs ds

JOIN LanguageNames ln ON ln.LanguageID=ds.SupervLanguage

WHERE ds.SupervLName=@LastN

;

GO

CREATE PROCEDURE SlctClientByLName

@LastN NVARCHAR(255)

AS

SELECT ClientID AS'Клієнт ID',

ClientFName AS 'Ім''я',

ClientPatronymic AS 'По-батькові',

ClientLName AS 'Прізвище',

[dbo].gender(ClientSex) AS 'Стать',

ClientDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ClientMail AS 'Пошта'

FROM DataClients d

JOIN LanguageNames ln ON ln.LanguageID=d.ClientLanguage

WHERE ClientLName=@LastN

;

GO

CREATE PROCEDURE SlctDoctorsByID

@Ident INT

AS

SELECT DISTINCT dd.DoctorID AS'Доктор ID',

dd.DoctorLName AS 'Прізвище',

dd.DoctorFName AS 'Ім''я',

dd.DoctorPatronymic AS 'По-батькові',

[dbo].gender(dd.DoctorSex) AS 'Стать',

dd.DoctorDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

dd.HourlyRate AS 'Оплата/год.',

tm.TherapyMethod AS 'Метод терапії',

ds.SupervID 'Супервізор ID',

ds.SupervFName 'Прізвище',

ds.SupervLName 'Ім''я',

ds.SupervPatronymic 'По-батькові',

dd.DoctorMail AS 'Пошта'

FROM DataDoctors dd

JOIN LanguageNames ln ON ln.LanguageID=dd.DoctorLanguage

JOIN TherapyMethods tm ON dd.TherapyMethod=tm.TherapyID

JOIN DataSupervs ds ON dd.SupervisorID= ds.SupervID

WHERE dd.DoctorID=@Ident

;

GO

CREATE PROCEDURE SlctSupervByID

@Ident INT

AS

SELECT ds.SupervID 'Супервізор ID',

ds.SupervFName 'Ім''я',

ds.SupervPatronymic 'По-батькові',

ds.SupervLName 'Прізвище',

[dbo].gender(ds.SupervSex) AS 'Стать',

ds.SupervDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ds.HourlyRate AS 'Оплата/год.',

ds.SupervMail AS 'Пошта'

FROM DataSupervs ds

JOIN LanguageNames ln ON ln.LanguageID=ds.SupervLanguage

WHERE ds.SupervID=@Ident

;

GO

CREATE PROCEDURE SlctSupervByName

@Ident NVARCHAR(255)

AS

SELECT ds.SupervID 'Супервізор ID',

ds.SupervFName 'Ім''я',

ds.SupervPatronymic 'По-батькові',

ds.SupervLName 'Прізвище',

[dbo].gender(ds.SupervSex) AS 'Стать',

ds.SupervDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ds.HourlyRate AS 'Оплата/год.',

ds.SupervMail AS 'Пошта'

FROM DataSupervs ds

JOIN LanguageNames ln ON ln.LanguageID=ds.SupervLanguage

WHERE ds.SupervLName=@Ident

;

GO

CREATE PROCEDURE SlctClientByID

@Ident INT

AS

SELECT ClientID AS'Клієнт ID',

ClientFName AS 'Ім''я',

ClientLName AS 'Прізвище',

ClientPatronymic AS 'По-батькові',

[dbo].gender(ClientSex) AS 'Стать',

ClientDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ClientMail AS 'Пошта'

FROM DataClients d

JOIN LanguageNames ln ON ln.LanguageID=d.ClientLanguage

WHERE ClientID=@Ident

;

GO

CREATE PROCEDURE SlctMeetingByID

@MeetingID INT

AS

SELECT \* FROM Meetings m

WHERE m.Identifier= @MeetingID

;

GO

CREATE PROCEDURE AddClient

@ClientFName NVARCHAR(35),

@ClientMName NVARCHAR(35),

@ClientLName NVARCHAR(35),

@Sex TINYINT,

@BirthD DATETIME,

@Language INT,

@Mail NVARCHAR(35)

AS

INSERT INTO DataClients

VALUES (@ClientLName, @ClientFName,@ClientMName, @Sex, @BirthD, @Language, @Mail)

;

GO

CREATE PROCEDURE AddSuperv

@FName NVARCHAR(35),

@MName NVARCHAR(35),

@LName NVARCHAR(35),

@Sex TINYINT,

@BirthD DATETIME,

@Language INT,

@HourlyRate INT,

@Mail NVARCHAR(35)

AS

INSERT INTO DataSupervs

VALUES (@LName, @FName,@MName, @Sex, @BirthD, @Language,@HourlyRate,@Mail)

;

GO

CREATE PROCEDURE AddDoctor

@DoctorFName NVARCHAR(35),

@DoctorMName NVARCHAR(35),

@DoctorLName NVARCHAR(35),

@Sex TINYINT,

@BirthD DATETIME,

@Language INT,

@HourlyRate INT,

@TherapyMethod INT,

@SupervisorID INT,

@Mail NVARCHAR(35)

AS

INSERT INTO DataDoctors

VALUES (@DoctorLName, @DoctorFName, @DoctorMName,@Sex,

@BirthD,@Language, @HourlyRate,@TherapyMethod, @SupervisorID,

@Mail

)

;

GO

CREATE PROCEDURE AddSuperv

@SupervFName NVARCHAR(35),

@SupervMName NVARCHAR(35),

@SupervLName NVARCHAR(35),

@Sex TINYINT,

@BirthD DATETIME,

@Language INT,

@HourlyRate INT,

@Mail NVARCHAR(35)

AS

INSERT INTO DataSupervs

VALUES (@SupervLName, @SupervFName, @SupervMName,@Sex,

@BirthD,@Language, @HourlyRate, @Mail

)

;

GO

CREATE PROCEDURE AddMeetingsClient

@ClientID INT,

@DoctorID INT,

@TherapyType TINYINT

AS

DECLARE @curr\_time AS DATETIME= GETDATE()

INSERT INTO Meetings(ClientID,DoctorID, TherapyType, SessionBegin, SessionEnd)

VALUES (@ClientID, @DoctorID, @TherapyType, @curr\_time, DATEADD(hh, 1, @curr\_time))

;

GO

CREATE PROCEDURE AddMeetingsSuperv

@DoctorID INT

AS

DECLARE @curr\_time AS DATETIME= GETDATE(),

@SupervID AS INT

SELECT @SupervID=SupervisorID

FROM DataDoctors

WHERE DoctorID=@DoctorID

INSERT INTO SupAnDocMeetings(SupervisorID,DoctorID, SessionBegin, SessionEnd)

VALUES (@SupervID, @DoctorID,@curr\_time, DATEADD(hh, 1, @curr\_time))

;

GO

CREATE FUNCTION gender(

@SexID BIT

)

RETURNS NVARCHAR(10)

AS

BEGIN

DECLARE

@GENDER NVARCHAR(10)

SET @GENDER='Чоловіча'

IF(@SexID=0)

SET @GENDER='Жіноча'

RETURN @GENDER

END

CREATE PROCEDURE SlctAllClients

AS

SELECT ClientID AS'Клієнт ID',

ClientFName AS 'Ім''я',

ClientPatronymic AS 'По-батькові',

ClientLName AS 'Прізвище',

[dbo].gender(ClientSex) AS 'Стать',

ClientDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ClientMail AS 'Пошта'

FROM DataClients d

JOIN LanguageNames ln ON ln.LanguageID=d.ClientLanguage

;

GO

CREATE PROCEDURE SlctAllDoct

AS

SELECT DISTINCT dd.DoctorID AS'Доктор ID',

dd.DoctorFName AS 'Ім''я',

dd.DoctorPatronymic AS 'По-батькові',

dd.DoctorLName AS 'Прізвище',

[dbo].gender(dd.DoctorSex) AS 'Стать',

dd.DoctorDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

dd.HourlyRate AS 'Оплата/год.',

tm.TherapyMethod AS 'Метод терапії',

ds.SupervID 'Супервізор ID',

ds.SupervFName 'Ім''я супервізора',

ds.SupervPatronymic 'По-батькові супервізора',

ds.SupervLName 'Прізвище супервізора',

dd.DoctorMail AS 'Пошта'

FROM DataDoctors dd

JOIN LanguageNames ln ON ln.LanguageID=dd.DoctorLanguage

JOIN TherapyMethods tm ON dd.TherapyMethod=tm.TherapyID

JOIN DataSupervs ds ON dd.SupervisorID= ds.SupervID

;

GO

CREATE PROCEDURE SlctDoctByID

@DoctorID INT

AS

SELECT DISTINCT dd.DoctorID AS'Доктор ID',

dd.DoctorFName AS 'Ім''я',

dd.DoctorPatronymic AS 'По-батькові',

dd.DoctorLName AS 'Прізвище',

[dbo].gender(dd.DoctorSex) AS 'Стать',

dd.DoctorDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

dd.HourlyRate AS 'Оплата/год.',

tm.TherapyMethod AS 'Метод терапії',

ds.SupervID 'Супервізор ID',

ds.SupervFName 'Ім''я супервізора',

ds.SupervPatronymic 'По-батькові супервізора',

ds.SupervLName 'Прізвище супервізора',

dd.DoctorMail AS 'Пошта'

FROM DataDoctors dd

JOIN LanguageNames ln ON ln.LanguageID=dd.DoctorLanguage

JOIN TherapyMethods tm ON dd.TherapyMethod=tm.TherapyID

JOIN DataSupervs ds ON dd.SupervisorID= ds.SupervID

WHERE dd.DoctorID=@DoctorID

;

GO

CREATE PROCEDURE SlctAllSuperv

AS

SELECT ds.SupervID 'Супервізор ID',

ds.SupervFName 'Ім''я',

ds.SupervPatronymic 'По-батькові',

ds.SupervLName 'Прізвище',

[dbo].gender(ds.SupervSex) AS 'Стать',

ds.SupervDateOfBirth AS 'Дата народження',

ln.LanguageName AS 'Мова спілкування',

ds.HourlyRate AS 'Оплата/год.',

ds.SupervMail AS 'Пошта'

FROM DataSupervs ds

JOIN LanguageNames ln ON ln.LanguageID=ds.SupervLanguage

;

GO

CREATE PROCEDURE SlctAllMeetings

AS

SELECT m.Identifier AS 'Ідентифікатор',

m.ClientID AS 'Клієнт ID',

dc.ClientFName AS 'Ім''я',

dc.ClientPatronymic AS 'По-батькові',

dc.ClientLName AS 'Прізвище',

dd.DoctorID AS 'Доктор ID',

dd.DoctorFName AS 'Ім''я лікаря',

dd.DoctorPatronymic AS 'По-батькові лікаря',

dd.DoctorLName AS 'Прізвище лікаря',

tt.TherapyType AS 'Тип терапії',

m.SessionBegin AS 'Початок терапії',

m.SessionEnd AS 'Кінець терапії'

FROM Meetings m

JOIN TherapyTypes tt ON tt.TherapyID= m.TherapyType

JOIN DataClients dc ON dc.ClientID=m.ClientID

JOIN DataDoctors dd ON dd.DoctorID=m.DoctorID

;

GO

CREATE PROCEDURE SlctAllMeetingsByClientID

@ClientID INT

AS

SELECT m.Identifier AS 'Ідентифікатор',

m.ClientID AS 'Клієнт ID',

dc.ClientFName AS 'Ім''я',

dc.ClientPatronymic AS 'По-батькові',

dc.ClientLName AS 'Прізвище',

dd.DoctorID AS 'Доктор ID',

dd.DoctorFName AS 'Ім''я лікаря',

dd.DoctorPatronymic AS 'По-батькові лікаря',

dd.DoctorLName AS 'Прізвище лікаря',

tt.TherapyType AS 'Тип терапії',

m.SessionBegin AS 'Початок терапії',

m.SessionEnd AS 'Кінець терапії'

FROM Meetings m

JOIN TherapyTypes tt ON tt.TherapyID= m.TherapyType

JOIN DataClients dc ON dc.ClientID=m.ClientID

JOIN DataDoctors dd ON dd.DoctorID=m.DoctorID

WHERE m.ClientID=@ClientID

;

GO

CREATE PROCEDURE SlctAllSupAndDocMeetings

AS

SELECT m.Identifier AS 'Ідентифікатор',

dd.DoctorID AS 'Доктор ID',

dd.DoctorFName AS 'Ім''я',

dd.DoctorPatronymic AS 'По-батькові',

dd.DoctorLName AS 'Прізвище',

ds.SupervID AS 'Супервізор ID',

ds.SupervFName AS 'Ім''я супервізора',

ds.SupervPatronymic AS 'По-батькові супервізора',

ds.SupervLName AS 'Прізвище супревізора',

m.SessionBegin AS 'Початок сеансу',

m.SessionEnd AS 'Кінець сеансу'

FROM SupAnDocMeetings m

JOIN DataSupervs ds ON ds.SupervID=m.SupervisorID

JOIN DataDoctors dd ON dd.DoctorID=m.DoctorID

;

GO

CREATE PROCEDURE SlctAllSupAndDocMeetingsByDoctorID

@DoctorID INT

AS

SELECT m.Identifier AS 'Ідентифікатор',

dd.DoctorID AS 'Доктор ID',

dd.DoctorFName AS 'Ім''я',

dd.DoctorPatronymic AS 'По-батькові',

dd.DoctorLName AS 'Прізвище',

ds.SupervID AS 'Супервізор ID',

ds.SupervFName AS 'Ім''я супервізора',

ds.SupervPatronymic AS 'По-батькові супервізора',

ds.SupervLName AS 'Прізвище супревізора',

m.SessionBegin AS 'Початок сеансу',

m.SessionEnd AS 'Кінець сеансу'

FROM SupAnDocMeetings m

JOIN DataSupervs ds ON ds.SupervID=m.SupervisorID

JOIN DataDoctors dd ON dd.DoctorID=m.DoctorID

WHERE dd.DoctorID=@DoctorID

;

GO

CREATE PROCEDURE UpdateDataClient

@ClientID INT,

@FNAME NVARCHAR (255),

@Patronymic NVARCHAR (255),

@LNAME NVARCHAR(255),

@EMAIL NVARCHAR(255)

AS

UPDATE DataClients

SET ClientFName=@FNAME,

ClientLName=@LNAME,

ClientPatronymic=@Patronymic,

ClientMail=@EMAIL

WHERE ClientID=@ClientID

;

GO

CREATE PROCEDURE UpdateDataSuperv

@SupervID INT,

@FNAME NVARCHAR (255),

@Patronymic NVARCHAR (255),

@LNAME NVARCHAR(255),

@MONEY INT,

@EMAIL NVARCHAR(255)

AS

UPDATE DataSupervs

SET SupervFName=@FNAME,

SupervLName=@LNAME,

SupervPatronymic=@Patronymic,

HourlyRate=@MONEY,

SupervMail=@EMAIL

WHERE SupervID=@SupervID

;

GO

CREATE PROCEDURE UpdateDataDoct

@DoctID INT,

@FNAME NVARCHAR (255),

@Patronymic NVARCHAR (255),

@LNAME NVARCHAR(255),

@MONEY INT,

@EMAIL NVARCHAR(255)

AS

UPDATE DataDoctors

SET DoctorFName=@FNAME,

DoctorLName=@LNAME,

DoctorPatronymic=@Patronymic,

HourlyRate=@MONEY,

DoctorMail=@EMAIL

WHERE DoctorID=@DoctID

;

GO

CREATE PROCEDURE DeleteClient

@ClientID INT

AS

DELETE FROM DataClients

WHERE ClientID=@ClientID

;

GO

CREATE PROCEDURE DeleteDoctor

@DoctorID INT

AS

DELETE FROM DataDoctors

WHERE DoctorID=@DoctorID

;

GO

CREATE PROCEDURE DeleteSuperv

@SupervID INT

AS

DELETE FROM DataSupervs

WHERE SupervID=@SupervID

;

GO

CREATE PROCEDURE SlctSupervsIDName

AS

SELECT SupervID AS 'СупервізорID',

CONCAT(SupervFName,' ',SupervPatronymic,' ',SupervLName) AS 'ПІБ'

FROM DataSupervs

CREATE PROCEDURE SlctDoctorsIDName

AS

SELECT DoctorID AS 'ДокторID',

CONCAT(DoctorFName,' ',DoctorPatronymic,' ',DoctorLName) AS 'ПІБ'

FROM DataDoctors

CREATE PROCEDURE SlctClientsIDName

AS

SELECT ClientID AS 'ClientID',

CONCAT(ClientFName,' ',ClientPatronymic,' ',ClientLName) AS 'ПІБ'

FROM DataClients

CREATE PROCEDURE AddSupAndDocMeeting

@DoctorID INT

AS

INSERT INTO SupAnDocMeetings

VALUES (

(SELECT SupervisorID FROM DataDoctors

WHERE DoctorID=@DoctorID

),

@DoctorID,

GETDATE(),

DATEADD(hh, 1, current\_timestamp))

;

GO

/////////////////////////////////////////////////

CREATE VIEW AllMeetings

AS

SELECT Identifier AS 'Ідентифікатор',

ClientID AS 'PatientID',

DoctorID AS 'DoctorID',

SessionBegin AS 'Початок сесії',

SessionEnd AS 'Кінець сесії',

'Клієнт' AS 'Хто лікувався'

FROM Meetings

UNION ALL

SELECT Identifier AS 'Ідентифікатор',

DoctorID AS 'PatientID',

SupervisorID AS 'DoctorID',

SessionBegin AS 'Початок сесії',

SessionEnd AS 'Кінець сесії',

'Лікар' AS 'Хто лікувався'

FROM SupAnDocMeetings

SELECT \* FROM AllMeetings

CREATE TABLE DataClientsArchive

(

ClientID int NOT NULL,

ClientLName nvarchar(255) NOT NULL,

ClientFName nvarchar(255) NOT NULL,

ClientPatronymic nvarchar(255) NULL,

ClientSex BIT NOT NULL,

ClientDateOfBirth date NOT NULL,

ClientLanguage int NOT NULL,

ClientMail nvarchar(255) NOT NULL,

WhoDeleted nvarchar(255) NOT NULL DEFAULT(CURRENT\_USER),

WhenDeleted datetime NOT NULL DEFAULT (GETDATE())

);

CREATE TRIGGER DataClients\_DELETE ON DataClients

AFTER DELETE

AS

INSERT INTO DataClientsArchive (ClientID,ClientLName,ClientFName,ClientPatronymic,ClientSex,ClientDateOfBirth,ClientLanguage,ClientMail)

SELECT ClientID,ClientLName,ClientFName,ClientPatronymic,ClientSex,ClientDateOfBirth,ClientLanguage,ClientMail

FROM DELETED

CREATE TABLE DataDoctorsArchive

(

DoctorID int PRIMARY KEY NOT NULL IDENTITY(1,1),

DoctorLName nvarchar(255) NOT NULL,

DoctorFName nvarchar(255) NOT NULL,

DoctorPatronymic nvarchar(255) NULL,

DoctorSex BIT NOT NULL,

DoctorDateOfBirth date NOT NULL,

DoctorLanguage int NOT NULL,

HourlyRate int NOT NULL,

TherapyMethod int NOT NULL,

SupervisorID int NOT NULL,

DoctorMail nvarchar(255) NOT NULL,

WhoDeleted nvarchar(255) NOT NULL DEFAULT(CURRENT\_USER),

WhenDeleted datetime NOT NULL DEFAULT (GETDATE())

);

CREATE TRIGGER DataDoctors\_DELETE ON DataDoctors

AFTER DELETE

AS

INSERT INTO DataDoctorsArchive (DoctorID,DoctorLName,DoctorFName,DoctorPatronymic,DoctorSex,DoctorDateOfBirth,DoctorLanguage,HourlyRate,TherapyMethod,SupervisorID,DoctorMail)

SELECT DoctorID,DoctorLName,DoctorFName,DoctorPatronymic,DoctorSex,DoctorDateOfBirth,DoctorLanguage,HourlyRate,TherapyMethod,SupervisorID,DoctorMail

FROM DELETED

CREATE TABLE DataSupervArchive

(

SupervID int PRIMARY KEY NOT NULL IDENTITY(1,1),

SupervLName nvarchar(255) NOT NULL,

SupervFName nvarchar(255) NOT NULL,

SupervPatronymic nvarchar(255) NULL,

SupervSex BIT NOT NULL,

SupervDateOfBirth date NOT NULL,

SupervLanguage int NOT NULL,

HourlyRate int NOT NULL,

SupervMail nvarchar(255) NOT NULL,

WhoDeleted nvarchar(255) NOT NULL DEFAULT(CURRENT\_USER),

WhenDeleted datetime NOT NULL DEFAULT (GETDATE())

);

CREATE TRIGGER DataSuperv\_DELETE ON DataSupervs

AFTER DELETE

AS

INSERT INTO DataSupervArchive(SupervID,SupervLName,SupervFName,SupervPatronymic,SupervSex,SupervDateOfBirth,SupervLanguage,HourlyRate,SupervMail)

SELECT SupervID,SupervLName,SupervFName,SupervPatronymic,SupervSex,SupervDateOfBirth,SupervLanguage,HourlyRate,SupervMail

FROM DELETED

CREATE VIEW CurrentMeeting

AS

SELECT Ідентифікатор,

PatientID,

CASE

WHEN [Хто лікувався]='Клієнт' THEN (

SELECT CONCAT(ClientFName,' ',ClientPatronymic,' ',ClientLName) AS 'ПІБ'

FROM DataClients

WHERE ClientID=PatientID)

WHEN [Хто лікувався]='Лікар' THEN (

SELECT CONCAT(DoctorFName,' ',DoctorPatronymic,' ',DoctorLName) AS 'ПІБ'

FROM DataDoctors

WHERE DoctorID=PatientID)

END AS 'ПІБ Пацієнта',

meet.DoctorID,

CASE

WHEN [Хто лікувався]='Клієнт' THEN (

SELECT CONCAT(DoctorFName,' ',DoctorPatronymic,' ',DoctorLName) AS 'ПІБ'

FROM DataDoctors

WHERE meet.DoctorID=DoctorID)

WHEN [Хто лікувався]='Лікар' THEN (

SELECT CONCAT(SupervFName,' ',SupervPatronymic,' ',SupervLName) AS 'ПІБ'

FROM DataSupervs

WHERE meet.DoctorID=SupervID)

END AS 'ПІБ Лікаря',

[Хто лікувався] AS 'Хто лікується',

[Початок сесії],

[Кінець сесії]

FROM AllMeetings meet

WHERE [Кінець сесії]>GETDATE()

EXEC AddSupAndDocMeeting 2

SELECT \* FROM CurrentMeeting

CREATE FUNCTION ShowDoctorsByPrice (@priceFrom MONEY, @priceTo MONEY)

RETURNS TABLE

AS

RETURN

(

SELECT DISTINCT DoctorID AS 'ЛікарID',

CONCAT(DoctorFName,' ',DoctorPatronymic,' ',DoctorLName) AS 'ПІБ',

HourlyRate AS 'Погодинна оплата'

FROM DataDoctors

WHERE HourlyRate BETWEEN @priceFrom AND @priceTo

);

GO

SELECT \* FROM dbo.ShowDoctorsByPrice(10,200)

USE PsychologyOnline

CREATE ROLE accessClients

GRANT SELECT, UPDATE ON DataClients TO accessClients

GRANT SELECT, UPDATE ON Meetings TO accessClients

GRANT SELECT ON LanguageNames TO accessClients

GRANT SELECT ON TherapyTypes TO accessClients

CREATE LOGIN AlexClient WITH PASSWORD = '05fc1c06vh'

CREATE USER AlexClient FOR LOGIN AlexClient

ALTER ROLE accessClients ADD MEMBER AlexClient

CREATE ROLE accessDoctors

GRANT SELECT, UPDATE ON DataDoctors TO accessDoctors

GRANT SELECT, UPDATE ON SupAnDocMeetings TO accessDoctors

GRANT SELECT ON LanguageNames TO accessDoctors

GRANT SELECT ON TherapyTypes TO accessDoctors

GRANT SELECT ON TherapyMethods TO accessDoctors

CREATE LOGIN AlexeiDoctor WITH PASSWORD = '05fc1c06vh'

CREATE USER AlexeiDoctor FOR LOGIN AlexeiDoctor

ALTER ROLE accessDoctors ADD MEMBER AlexeiDoctor

CREATE ROLE accessSuperv

GRANT SELECT, UPDATE ON DataSupervs TO accessSuperv

GRANT SELECT, UPDATE ON SupAnDocMeetings TO accessSuperv

GRANT SELECT ON LanguageNames TO accessSuperv

GRANT SELECT ON TherapyTypes TO accessSuperv

CREATE LOGIN OleksiiSuperv WITH PASSWORD = '05fc1c06vh'

CREATE USER OleksiiSuperv FOR LOGIN OleksiiSuperv

ALTER ROLE accessSuperv ADD MEMBER OleksiiSuperv

*студента групи ІП-96 IІ курсу*

*Бурятова О.О.*

(Обсяг програми (документа), арк., Кб)

*12 арк, 339 Кб*

(Вид носія даних)

*CD-RW*

(Найменування програми (документа))

*Тексти програмного коду програмного забезпечення розробки онлайн-платформи психотерапії «Рука допомоги»*

Перелік посилань

1. SQL за 10 минут [Електронний ресурс] // Вильямс. – 2014. – Режим доступу до ресурсу: http://padabum.com/d.php?id=161361
2. Полное руководство по языку программирования С# 9.0 и платформе .NET 5 [Електронний ресурс] – Режим доступу до ресурсу: https://metanit.com/sharp/tutorial/.