

Міністерство науки і освіти України
Житомирський державний технологічний університет

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА

з дисципліни: «Бази даних»

на тему: Бази даних “Страхова компанія”

Студента II курсу ІПЗ-22-3 групи
спеціальності 121 «Інженерія програмного
забезпечення»

Леус Вадим Олександрович

(прізвище та ініціали)

Керівник: _____

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

І.І. Сугоняк

С.М. Кравченко

О.В. Чижмотря

м. Житомир – 2023

Зміст курсової роботи

ВСТУП.....	2
ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ РОБОТИ СТРАХОВОЇ КОМПАНІЇ.....	4
1.1. Аналіз особливостей процесу формування полісів страхування	4
1.2. Обґрунтування вибору засобів реалізації.	4
ПРОЕКТУВАННЯ БАЗИ ДАНИХ СТРАХОВОЇ КОМПАНІЇ.....	6
2.1. Аналіз структури інформаційних процесів страхової компанії	6
2.2. Проектування бази даних страхової компанії	8
РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ СТРАХОВОЇ КОМПАНІЇ.....	12
3.1 Проектування інтерфейсу системи страхової компанії	12
3.2. Реалізації операцій обробки даних в БД.....	12
3.3. Організація звітності системи	14
АДМІНІСТРУВАННЯ БАЗ ДАНИХ	21
4.1. Розробка заходів захисту інформації в БД	21
ЛІТЕРАТУРА.....	22
ВИСНОВКИ	23
ДОДАТКИ.....	24
Додаток А.....	25
Додаток Б	31

					ДУ «Житомирська політехніка».23.121.17.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Леус В.О.			База даних “Страхова компанія”	Літ.	Арк.	Аркушів
Перевір.							4	51
Керівник		Кравченко С.М.				ФІКТ, Гр. ІПЗ-22-3[1]		
Н. контр.								
Зав. каф.								

ВСТУП

Актуальність теми.

У сучасному світі, де страхові послуги відіграють важливу роль у фінансовій стабільності та соціальній захищеності, використання ефективної бази даних стає критичною складовою успішної діяльності страхових компаній. Розробка та оптимізація бази даних спрямовані на підвищення ефективності обробки інформації, забезпечення надійності зберігання даних клієнтів та оптимізацію управління ризиками.

Сучасні тенденції в галузі страхування вимагають від компаній швидкого доступу до великого обсягу даних для аналізу ризиків, прийняття обґрунтованих рішень та надання якісних послуг клієнтам. Отже, дослідження та вдосконалення баз даних страхових компаній є практичною необхідністю для їхнього успішного функціонування в умовах постійно зростаючого обсягу інформації та конкуренції на ринку.

Ця робота спрямована на вивчення сучасних технологій управління базами даних та їхнє впровадження в діяльність страхових компаній з метою підвищення їхньої конкурентоспроможності та забезпечення стабільного розвитку в умовах динамічного ринкового середовища.

Мета роботи: спроектувати і реалізувати програмне забезпечення з використанням бази даних.

Основні етапи виконання розрахункового завдання:

1. Проектування бази даних.
2. Реалізація бази даних.
3. Проектування програмного забезпечення.
4. Реалізація програмного забезпечення.
5. Тестування програмного забезпечення.
6. Складання пояснювальної записки.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

Умова завдання:

Програмне забезпечення обліків полісів.

У програмі повинні бути передбачені наступні функції:

1. Облік страхових послуг;
2. Облік укладених страхових договорів;
3. Додавання інформації про клієнтів та співробітників

Предметом дослідження є можливості застосування концепції БД та СУБД для забезпечення інформаційних потреб страхової компанії. Об'єктом дослідження є методи та засоби проектування баз даних за визначеними предметними областями.

В процесі виконання даної курсової роботи буде спроектовано і реалізовано базу даних згідно «Страхова компанія». Діяльність компанії організована в такий спосіб: до агентів звертаються різні особи з метою укладення договору про страхування. Залежно від прийнятих на страхування об'єктів і застрахованих ризиків, договір укладається за певним видом страхування (наприклад, страхування авто, страхування домівки, страхування життя).

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ РОБОТИ СТРАХОВОЇ КОМПАНІЇ.

1.1. Аналіз особливостей процесу формування полісів страхування

На початку аналізу слід звернутися до процесу збору інформації від страховика. Важливо розглянути, яким чином здійснюється збір та обробка особистих даних клієнтів, які є об'єктом страхування. Важливим етапом буде вивчення технічних та юридичних аспектів цього процесу, зокрема, дотримання вимог захисту персональних даних.

Далі важливо розглянути процес оцінки ризику та визначення страхових тарифів. Аналіз включає дослідження методів визначення ризику, використовуваних у формулюванні полісів, а також взаємозв'язок між цими факторами та визначенням оптимальних тарифів для різних категорій страхованих об'єктів.

Крім того, слід проаналізувати процес формування умов та винятків у полісі. Особлива увага має бути приділена чіткості та зрозумілості викладення умов страхового покриття, а також ідентифікації потенційних ризиків та винятків, які можуть виникнути під час виконання угоди.

Загальний аналіз особливостей процесу формування полісів страхування повинен враховувати як кількісні, так і якісні аспекти, надавати рекомендації щодо оптимізації та вдосконалення цього процесу з метою підвищення ефективності та задоволення потреб клієнтів.

1.2. Обґрунтування вибору засобів реалізації.

На вибір у мене було 3 СУБД: Microsoft SQL, Server Oracle і PostgreSQL:

Таблиця 1.1.

Функціональне порівняння СУБД

Характеристика	Microsoft SQL Server	PostgreSQL	Oracle
Адміністративне керування	Добре	Відмінно	Відмінно

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Графічні інструменти	Відмінно	Добре	Добре
Простота обслуговування	Відмінно	Відмінно	Відмінно
Механізм даних	Добре	Відмінно	Відмінно
Робота с декількома ЦП	Задовільно	Відмінно	Відмінно
Функції з'єднання і вибір індексів	Відмінно	Відмінно	Відмінно
Одночасний доступ декількох користувачів	Добре	Відмінно	Відмінно
Обробка даних мультимедіа	Погано	Добре	Відмінно
Підключення до Web	Задовільно	Відмінно	Відмінно
Повнотекстовий пошук	Добре	Відмінно	Відмінно
Функціональна сумісність	Добре	Відмінно	Добре
Інтеграція з іншими СУБД	Добре	Відмінно	Добре
Єдина реєстрація	Добре	Добре	Добре
Робота під керівництвом ОС	Задовільно	Задовільно	Добре
Можливості програмування	Задовільно	Задовільно	Відмінно
Процедури, що зберігаються та тригери	Добре	Добре	Відмінно
Вбудована мова програмування	Задовільно	Задовільно	Відмінно
Побудова БД	Добре	Добре	Відмінно
Мова SQL	Відмінно	Відмінно	Відмінно
Підтримка об'єктно-орієнтованої парадигми	Задовільно	Відмінно	Відмінно
Робота у режимі віддаленого доступу	Відмінно	Відмінно	Відмінно
Тиражування	Відмінно	Відмінно	Відмінно
Розподілена обробка транзакцій	Відмінно	Добре	Відмінно
Дистанційне адміністрування	Добре	Відмінно	Відмінно
Організація сховищ даних і підготовка звітів	Відмінно	Відмінно	Відмінно
Засоби завантаження	Відмінно	Відмінно	Відмінно
Засоби аналізу	Відмінно	Відмінно	Відмінно

ПРОЕКТУВАННЯ БАЗИ ДАНИХ СТРАХОВОЇ КОМПАНІЇ.

2.1. Аналіз структури інформаційних процесів страхової компанії

Проаналізувавши умову і ознайомившись з діяльністю страхової компанії, можна виділити основні завдання автоматизованої системи обліку страхових договорів і страхових випадків.

Для задоволення всіх вимог страхових агентів і клієнтів, автоматизована система повинна володіти наступним функціоналом:

1. Оформлення договору:
 - 1.1. Вибір типу договору;
 - 1.2. Внесення даних про клієнта;
 - 1.3. Внесення даних про предмет страхування;
2. Видалення договору;
3. Перегляд активних договорів;
4. Робота з полісами:
 - 4.1. Зміна умов;
5. Додавання нового клієнта;
6. Видалення клієнта.

Діаграма варіантів використання предметної області «Страхова компанія»:

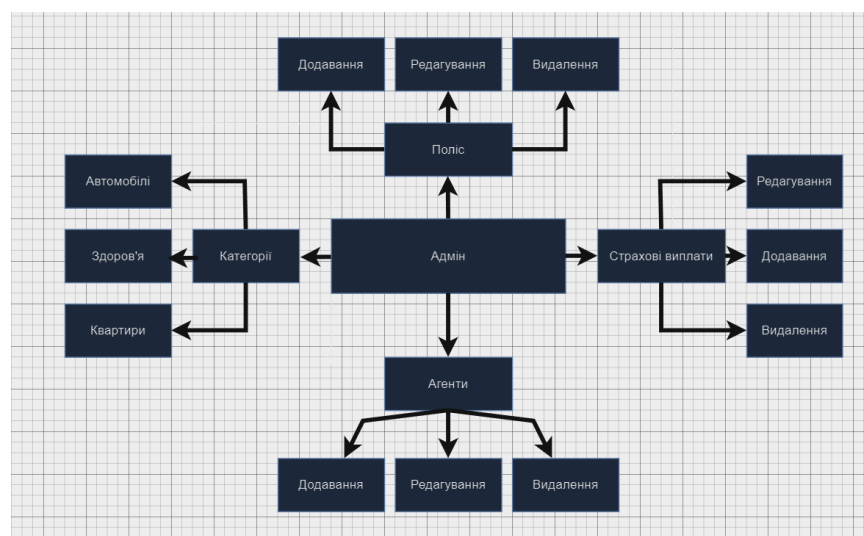


Рис. 2.1. Схема роботи системи

В результаті аналізу функцій майбутнього ПЗ були виділені дані, які будуть зберігатися в БД: дані про клієнтів, співробітників, договори, страхові послуги, що надаються компанією.

Автоматизація процесу обчислення необхідної суми договору за допомогою множення суми щомісячного платежу, кількості місяців в терміні договору.

Також система забезпечить контроль за витіканням термінів дії договору - моделі предметної області оператор буде бачити список прострочених договорів.

Для побудови ER-моделі предметної області необхідно виділити сутності предметної області. В результаті аналізу предметної області були виділені сутності :

1. Поліси - моделі предметної області містить інформацію про клієнта, страхового агента, наданої послуги, суми договору і терміни дії.
2. Страхування житла - вид оформлюваних договорів, містить дані про адресу та оціночну вартість нерухомості.
3. Страхування життя - вид оформлюваних договорів, містить дані про людину, чиє життя є об'єктом страхування.
4. Страхування автомобіля - містить дані про номер, рік випуску і оціночну вартість засобу пересування.
5. Співробітник - містить інформацію про співробітників компанії і їх посадах. Таблиця необхідна для оформлення договору.
6. Клієнт - містить інформацію про клієнтів компанії.

Для відображення виділених сутностей використовувалась ER- діаграма. Її реалізація представлена на Рисунку 2.2.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

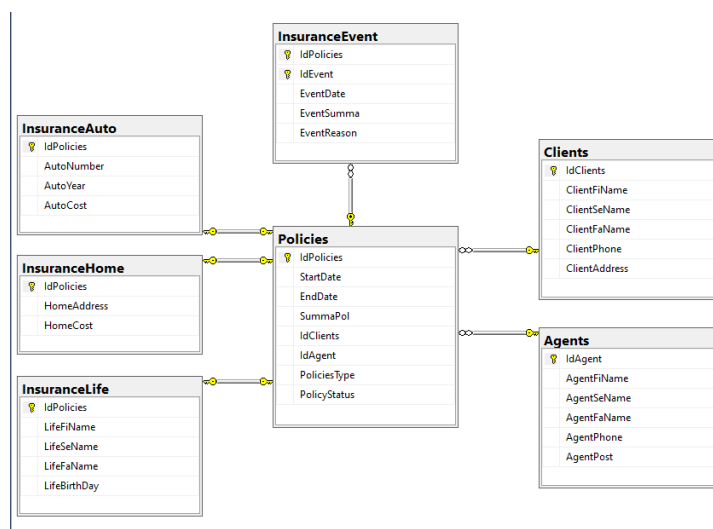


Рис. 2.2. ER- діаграма бд “Страхова компанія”

На цьому етапі проектування бази даних було створено ER-діаграму страхової компанії.

На цьому етапі проектування бази даних було створено ER-діаграму страхової компанії.

2.2. Проектування бази даних страхової компанії

На етапі логічного проектування БД необхідно на основі побудованої ER-діаграми страхової компанії отримати схему реляційної бази даних.

Внаслідок проектування до бази даних включено таблиці:

- ❖ dbo. Agents
- ❖ dbo.Clients
- ❖ dbo.InsuranceAuto
- ❖ dbo.InsuranceEvent
- ❖ dbo.InsuranceHome
- ❖ dbo.InsuranceLife
- ❖ dbo.Policies

Для збереження даних про поліси була призначена таблиця "dbo.Policies". До них належать відомості ID поліса, про початок та кінець страхування, сума

страховки, тип поліса, ID клієнта та агента пов'язаного з цим полісом, а також статус самого поліса. Структура таблиці наведена нижче:

Таблиця 2.1.

Типи даних для стовпців таблиці Policies

Policies				
<i>Поле</i>	<i>Тип поля</i>	<i>ПК</i>	<i>ЗК</i>	<i>Опис</i>
IdPolicies	INT	+	-	Код полісу
StartDate	DATE	-	-	Дата початку дії страховки
EndDate	DATE	-	-	Дата закінчення дії страховки
SummaPol	DECIMAL	-	-	Сума страховки
PoliciesType	VARCHAR(50)	-	-	Тип поліса
IdClients	INT	-	+	Код клієнта
IdAgent	INT	-	+	Код агента
PolicyStatus	VARCHAR(50)	-	-	Статус полісу

Для збереження даних про агентів була призначена таблиця "dbo. Agents". До них належать відомості про ID агента, ім'я, прізвище, ім'я по-батькові, номер телефону та пост агента. Структура таблиці наведена нижче:

Таблиця 2.2.

Типи даних для стовпців таблиці Agents

Agents				
<i>Поле</i>	<i>Тип поля</i>	<i>ПК</i>	<i>ЗК</i>	<i>Опис</i>
IdAgent	INT	+	-	Код агента
AgentFiName	VARCHAR(50)	-	-	Ім'я агента
AgentSeName	VARCHAR(50)	-	-	Прізвище агента
AgentFaName	VARCHAR(50)	-	-	Ім'я по-батькові агента
AgentPhone	VARCHAR(50)	-	-	Номер телефону агента
AgentPost	VARCHAR(50)	-	-	Посада агента

Для збереження даних про клієнтів була призначена таблиця "dbo. Clients". До них належать відомості про ID клієнта, ім'я, прізвище, ім'я по-батькові, номер телефону та адреса клієнтів. Структура таблиці наведена нижче:

Таблиця 2.3.

Типи даних для стовпців таблиці Clients

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Clients				
Поле	Тип поля	ПК	ЗК	Опис
IdClients	INT	+	-	Код клієнта
ClientFiName	VARCHAR(50)	-	-	Ім'я клієнта
ClientSeName	VARCHAR(50)	-	-	Прізвище клієнта
ClientFaName	VARCHAR(50)	-	-	Ім'я по-батькові клієнта
ClientPhone	VARCHAR(50)	-	-	Номер телефону клієнта
ClientAddress	VARCHAR(50)	-	-	Адреса клієнта

Для збереження даних про автомобілі була призначена таблиця "dbo.InsuranceAuto". До них належать відомості про номер авто, рік випуску авто та ціна. Структура таблиці наведена нижче:

Таблиця 2.4.

Типи даних для стовпців таблиці InsuranceAuto

InsuranceAuto				
Поле	Тип поля	ПК	ЗК	Опис
IdPolicies	INT	-	+	Код поліса
AutoNumber	VARCHAR(50)	-	-	Номер автомобіля
AutoYear	INT	-	-	Рік випуску автомобіля
AutoCost	DECIMAL	-	-	Ціна автомобіля

Для збереження даних про квартири була призначена таблиця "dbo.InsuranceHome". До них належать відомості про адресу та ціну квартири. Структура таблиці наведена нижче:

Таблиця 2.5.

Типи даних для стовпців таблиці InsuranceHome

InsuranceHome				
Поле	Тип поля	ПК	ЗК	Опис
IdPolicies	INT	-	+	Код поліса
HomeAddress	VARCHAR(50)	-	-	Адреса квартири
HomeCost	DECIMAL	-	-	Ціна квартири

Для збереження даних про застрахованих людей була призначена таблиця "dbo.InsuranceLife". До них належать відомості про ім'я, прізвище, ім'я по батькові та дата народження. Структура таблиці наведена нижче:

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.6.

Типи даних для стовпців таблиці InsuranceLife

InsuranceLife				
<i>Поле</i>	<i>Тип поля</i>	<i>ПК</i>	<i>ЗК</i>	<i>Опис</i>
IdPolicies	INT	-	+	Код полісу
LifeFiName	VARCHAR(50)	-	-	Ім'я людини
LifeSeName	VARCHAR(50)	-	-	Прізвище людини
LifeFaName	VARCHAR(50)	-	-	Ім'я по-батькові людини
LifeBirthDay	DATE	-	-	День народження людини

Для збереження даних про страхові виплати була призначена таблиця "dbo.InsuranceEvent". До них належать відомості про ID випадка, дати події, сума та причина виплати. Структура таблиці наведена нижче:

Таблиця 2.7.

Типи даних для стовпців таблиці InsuranceEvent

InsuranceEvent				
<i>Поле</i>	<i>Тип поля</i>	<i>ПК</i>	<i>ЗК</i>	<i>Опис</i>
IdPolicies	INT	-	+	Код поліса
IdEvent	INT	+	-	Код події
EventDate	VARCHAR(50)	-	-	Дата випадку
EventSumma	DECIMAL	-	-	Сума виплати
EventReason	VARCHAR(100)	-	-	Причина виплати

РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ СТРАХОВОЇ КОМПАНІЇ

3.1 Проектування інтерфейсу системи страхової компанії

Структура інтерфейсу наводиться в рис.3.1.

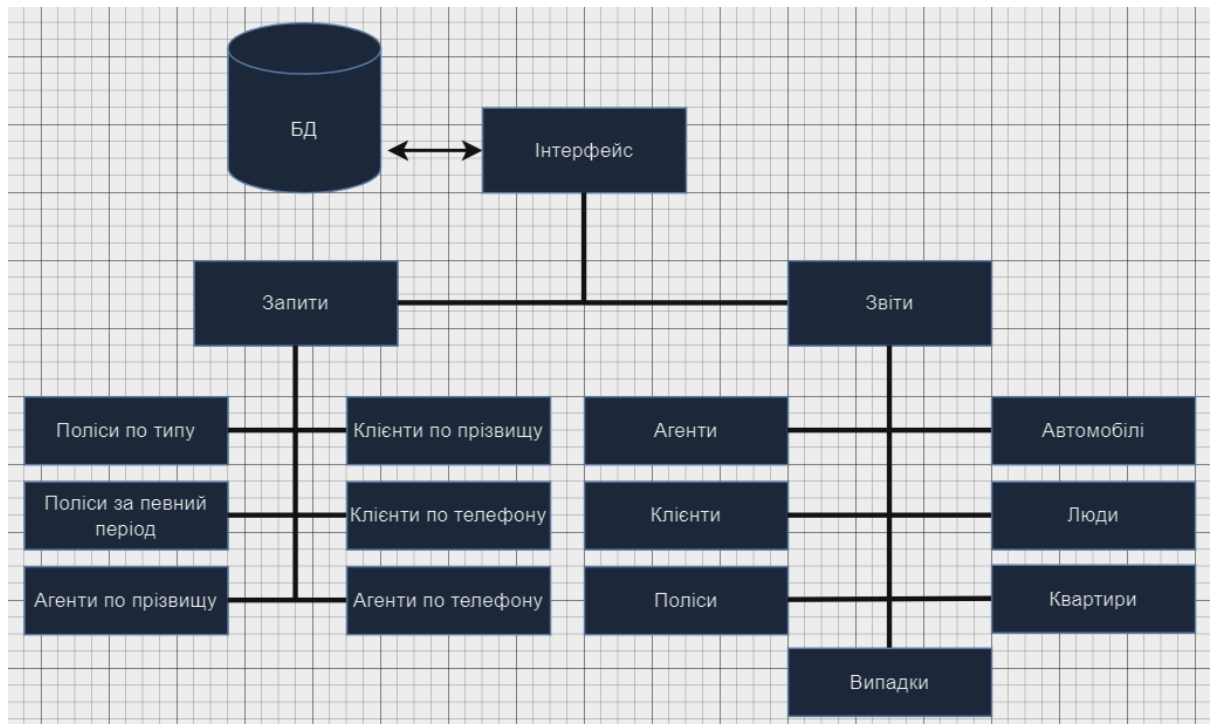


Рис.3.1. Структура інтерфейсу страхової компанії

3.2. Реалізації операцій обробки даних в БД

В якості СУБД була обрана SQL Server. Дана СУБД має такі переваги:

1. Масштабованість
2. Чудова продуктивність
3. Простота використання
4. Сховища даних

Отриманий SQL скрипт для створення таблиць бази даних:

```
CREATE DATABASE SIP; -- "Сфера захисту страхування" SIP Sphere_Insurance_Protection
GO
```

```
-- Створення таблиці "Клієнти"
CREATE TABLE Clients (
    IdClients INT PRIMARY KEY,
    ClientFiName VARCHAR(50),
    ClientSeName VARCHAR(50),
    ClientFaName VARCHAR(50),
    ClientPhone VARCHAR(20),
    ClientAddress VARCHAR(100)
);
```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

-- Створення таблиці "Агенти"
CREATE TABLE Agents (
    IdAgent INT PRIMARY KEY,
    AgentFiName VARCHAR(50),
    AgentSeName VARCHAR(50),
    AgentFaName VARCHAR(50),
    AgentPhone VARCHAR(20),
    AgentPost VARCHAR(50)
);

-- Створення таблиці "Поліси"
CREATE TABLE Policies (
    IdPolicies INT PRIMARY KEY,
    StartDate DATE,
    EndDate DATE,
    SummaPol DECIMAL(10, 2),
    PoliciesType VARCHAR(50),
    IdClients INT,
    IdAgent INT,
    PolicyStatus VARCHAR(20),
    FOREIGN KEY (IdClients) REFERENCES Clients(IdClients),
    FOREIGN KEY (IdAgent) REFERENCES Agents(IdAgent)
);

-- Створення таблиці "Страхування автомобіля"
CREATE TABLE InsuranceAuto (
    IdPolicies INT PRIMARY KEY,
    AutoNumber VARCHAR(20),
    AutoYear INT,
    AutoCost DECIMAL(10, 2),
    FOREIGN KEY (IdPolicies) REFERENCES Policies(IdPolicies)
);

-- Створення таблиці "Страхування житла"
CREATE TABLE InsuranceHome (
    IdPolicies INT PRIMARY KEY,
    HomeAddress VARCHAR(100),
    HomeCost DECIMAL(10, 2),
    FOREIGN KEY (IdPolicies) REFERENCES Policies(IdPolicies)
);

-- Створення таблиці "Страхування життя"
CREATE TABLE InsuranceLife (
    IdPolicies INT PRIMARY KEY,
    LifeFiName VARCHAR(50),
    LifeSeName VARCHAR(50),
    LifeFaName VARCHAR(50),
    LifeBirthDay DATE,
    FOREIGN KEY (IdPolicies) REFERENCES Policies(IdPolicies)
);

-- Створення таблиці "Страховий випадок"
CREATE TABLE InsuranceEvent (
    IdPolicies INT,
    IdEvent INT,
    EventDate DATE,
    EventSumma DECIMAL(10, 2),
    EventReason VARCHAR(100),
    PRIMARY KEY (IdPolicies, IdEvent),
    FOREIGN KEY (IdPolicies) REFERENCES Policies(IdPolicies)
);

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Сформовані таблиці в обраній СУБД зображені на рисунку 3.2.

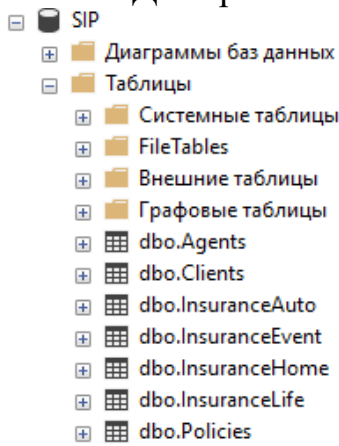


Рис. 3.2. Сформовані таблиці в СУБД

3.3. Організація звітності системи

Для роботи з базою даних була створена форма. Для зручності взаємодії з різними таблицями форма була розділена на 3 основні вкладки, зображені на рис.3.3.

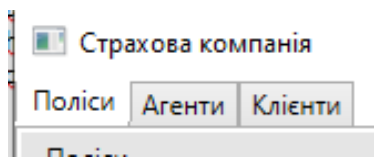


Рис. 3.3. Основні вкладки

The image shows a detailed view of the 'Поліси' (Policies) form. It includes a table with columns: IdPolicies, StartDate, EndDate, SummaPol, IdClients, IdAgent, PoliciesType. Below the table is a section for 'Додавання нового полісу:' (Adding a new policy) with input fields for ID, SummaPol, ID агента, and dates. To the right, there are sections for 'Застраховані автомобілі:' (Insured vehicles) and 'Фільтрація по датам:' (Filter by date) and 'Фільтрація по типу:' (Filter by type).

Рис. 3.4. Вкладка з інформацією про поліси

Поліси

IdPolicies	StartDate	EndDate	SummaPol	IdClients	IdAgent	PoliciesType
1	12/13/2023 12:00:00 AM	12/14/2024 12:00:00 AM	4299.00	1	1	Страховання а
2	9/1/2024 12:00:00 AM	9/2/2025 12:00:00 AM	3750.00	2	2	Страховання ж
3	3/11/2024 12:00:00 AM	3/12/2025 12:00:00 AM	5300.00	3	4	Страховання ж
4	12/17/2023 12:00:00 AM	12/18/2025 12:00:00 AM	2999.00	4	3	Страховання ж
5	2/5/2024 12:00:00 AM	2/6/2025 12:00:00 AM	4760.00	5	1	Страховання ж

Зміна даних

?

Хочете змінити дані цього поліса?

Да

Ніт

Додавання нового полісу:

ID:

1

Дата початку:

13.12.2023

15

Сума полісу:

4299,00

Дата кінця:

14.12.2024

15

ID агента:

1

ID клієнта:

1

Змінити поліс

Видалити поліс

Рис. 3.8. Запит на вставку даних з таблиці Policies

Страхова компанія

ПолісиАгентиКлієнти

Поліси

IdPolicies	StartDate	EndDate	SummaPol	IdClients	IdAgent	PoliciesType
1	12/13/2023 12:00:00 AM	12/14/2024 12:00:00 AM	4299.00	1	1	Страховання а
2	9/1/2024 12:00:00 AM	9/2/2025 12:00:00 AM	3750.00	2	2	Страховання ж
3	3/11/2024 12:00:00 AM	3/12/2025 12:00:00 AM	5300.00	3	4	Страховання ж
4	12/17/2023 12:00:00 AM	12/18/2025 12:00:00 AM	2999.00	4	3	Страховання ж
5	2/5/2024 12:00:00 AM	2/6/2025 12:00:00 AM	4760.00	5	1	Страховання ж

Дані поліса успішно змінені!

OK

Додавання нового полісу:

ID:

5

Дата початку:

05.02.2024

15

Сума полісу:

4760,00

Дата кінця:

06.02.2025

15

ID агента:

1

ID клієнта:

5

Змінити поліс

Видалити поліс

Рис. 3.9. Зміна даних у полісі

Підтвердження видалення

Ви впевнені, що хочете видалити цей поліс?

Да Нет

Додавання нового полісу

ID: 5 Дата початку: 05.02.2024 15

Сума полісу: 4760,00 Дата кінця: 06.02.2025 15

ID агента: 1 ID клієнта: 5

Змінити поліс Видалити поліс

Рис. 3.11. Запит на видалення поліса та того, що застраховано

Автомобілі Люди Квартири Страхові випадки

IdPc	IdEvent	EventDate	EventSun	EventReason
1	1	12/16/2023 12:00:00 AM	50000.00	Машина потрапила в аварію

ID поліса: 1 Сума виплати: 50000

ID випадка: 1 Причина: Машина потрапила в аварію

Дата випадку: 16.12.2023 15

Додати випадок Змінити випадок Видалити випадок

Рис. 3.11. Запит на додавання страхового випадку

Поліси

Агенти

Клієнти

Поліси

IdPo	StartDate	EndDate	Summa	IdC	Id	PoliciesType	PolicyStatus
3	3/11/2024 12:00:00 AM	3/12/2025 12:00:00 AM	5300.00	3	4	Страхування життя	Неактивний
5	2/5/2024 12:00:00 AM	2/6/2025 12:00:00 AM	4760.00	5	1	Страхування життя	Неактивний

Додавання нового полісу:

ID:

3

Сума полісу:

5300,00

ID агента:

4

Дата початку:

11.03.2024

Дата кінця:

12.03.2025

ID клієнта:

3

Змінити поліс

Видалити поліс

Автомобілі

Люди

Квартири

Страхові випадки

IdPolicies	IdEvent	EventDate	EventSumma	EventReason
1	1	12/16/2023 12:00:00 AM	50000.00	Машина потрапила в аварію

ID поліса:

1

Сума виплати:

50000

ID випадка:

1

Причина:

Машина потрапила в аварію

Дата випадку:

16.12.2023

15

Додати випадок

Змінити випадок

Видалити випадок

Фільтрація по датам:

Початок:

16.12.2023

Кінець:

17.12.2027

Зняти фільтрацію

Фільтрація по типу:

Фільтрація по типу:

☒

Страхування життя

☐

Страхування життя

☐

Страхування автомобіля

Зняти фільтрацію

Рис. 3.12.Запит на фільтрацію полісів по типу і датам

Поліси

Агенти

Клієнти

Агенти:

IdAgent	AgentFiName	AgentSeName	AgentFaName	AgentPhone	AgentPost
1	Борис	Приставако	Юрійович	+380 63 876 5432	Старший агент
2	Антон	Прокопенко	Олегович	+380 68 387 8488	Молодший агент
3	Петро	Іванов	Петрович	+380 95 789 4561	Агент з підтримки
4	Олена	Корпенко	Анатоліївна	+380 65 334 5025	Агент з поширення

Зміна даних

?

Хочете змінити дані цього робітника?

Да

Нет

Рис. 3.13.Запит вставки даних з таблиці Agents

Додання видалення редагування працівників:

ID:

1

Видалити агента

(Для видалення треба ID)

Ім'я:

Борис

Прізвище:

Приставако

По-батькові:

Юрійович

Додати агента

Номер телефону:

+380 63 876 5432

Посада:

Старший агент

Змінити дані

Дані агента успішно змінені!

OK

Рис. 3.14.Зміна даних в таблиці Agents

Агенти:

IdAgent	AgentFName	AgentSeName	AgentFaName	AgentPhone	AgentPost
1	Борис	Пристава	Юрійович	+380 63 876 5432	Старший агент

Поліси:

IdPolicies	StartDate	EndDate	SummaPol	IdClients	IdAgent
1	12/13/2023 12:00:00 AM	12/14/2024 12:00:00 AM	4299.00	1	1
5	2/5/2024 12:00:00 AM	2/6/2025 12:00:00 AM	4760.00	5	1

☒ Прізвище
 ☐ Номер телефону

Введіть дані для фільтрації:

Зняти фільтрацію

Пристава

Додавання видалення редагування працівників:

ID:

Ім'я:

Прізвище:

По-батькові:

Номер телефону:

Посада:

Видалити агента

(Для видалення треба ID)

Додати агента

Змінити дані

Рис. 3.15. Фільтрація агентів по прізвищу та полісів пов'язаних з ними

Агент успішно доданий!

OK

Додавання видалення редагування працівників:

ID:

Ім'я:

Прізвище:

По-батькові:

Номер телефону:

Посада:

Видалити агента

(Для видалення треба ID)

Додати агента

Змінити дані

Рис. 3.16. Запит на додавання агентів

Страхова компанія

Поліси

Агенти

Клієнти

Клієнти:

IdClients	ClientFName	ClientSeName	ClientFaName	ClientPhone	ClientAddress
1	Анна	Сидорова	Іванівна	+380 67 123 4567	вул. Шевченка, 13, 45
2	Петро	Стерненко	Геннадійович	+380 66 987 6543	вул. Київська, 21, 245
3	Олена	Коваленко	Анатолієвна	+380 99 555 1122	вул. Гагаріна, 5, 12
4	Володимир	Зеленов	Олегович	+380 94 223 4325	вул. Київська, 13, 45
5	Борис	Джонсонок	Якович	+380 97 234 4745	вул. Центральна, 13, 45

Поліси:

IdPolicies	StartDate	EndDate	SummaPol	IdClients	IdAgent
1	12/13/2023 12:00:00 AM	12/14/2024 12:00:00 AM	4299.00	1	1
2	9/1/2024 12:00:00 AM	9/2/2025 12:00:00 AM	3750.00	2	2
3	3/11/2024 12:00:00 AM	3/12/2025 12:00:00 AM	5300.00	3	4
4	12/17/2023 12:00:00 AM	12/18/2025 12:00:00 AM	2999.00	4	3
5	2/5/2024 12:00:00 AM	2/6/2025 12:00:00 AM	4760.00	5	1

Зміна даних

Хочете змінити дані цього клієнта?

Да

Нет

☐ Прізвище
 ☐ Номер телефону

Введіть дані для фільтрації:

Зняти фільтрацію

Додавання видалення редагування клієнтів

ID:

Ім'я:

Прізвище:

По-батькові:

Номер телефону:

Адреса:

Видалити агента

(Для видалення треба ID)

Додати клієнта

Змінити дані

Рис. 3.17. Запит вставку даних з таблиці Clients

×

Клієнт успішно доданий!

OK

Додавання видалення редагування клієнтів

ID: 6

Ім'я: Гена

Прізвище: Генюк

По-батькові: Генадійович

Номер телефону: +380 99 373 5435

Адреса: вул. Вокзальна, 5, 476

Видалити агента
(Для видалення треба ID)

Додати клієнта

Змінити дані

Рис. 3.18. Додавання нового клієнта

Поліси

Агенти

Клієнти

Клієнти:

IdClients	ClientFiName	ClientSeName	ClientFaName	ClientPhone	ClientAddress
1	Анна	Сидорова	Іванівна	+380 67 123 4567	вул. Шевченка, 13, 45

☐ Прізвище
 ☒ Номер телефону

Введіть дані для фільтрації:

Зняти фільтрацію

+380 67

Поліси:

IdPolicies	StartDate	EndDate	SummaPol	IdClients	IdAger
1	12/13/2023 12:00:00 AM	12/14/2024 12:00:00 AM	4299.00	1	1

Додавання видалення редагування клієнтів

ID: 6

Ім'я:

Прізвище:

По-батькові:

Номер телефону:

Адреса:

Видалити агента
(Для видалення треба ID)

Додати клієнта

Змінити дані

Рис. 3.19. Фільтрація клієнтів по номеру телефону та полісів пов'язаних з ними

АДМІНІСТРУВАННЯ БАЗ ДАНИХ

4.1. Розробка заходів захисту інформації в БД

В даній базі даних було створено 2 ролі “Бухгалтерія” (рис.4.1.) та ”Агенти” (рис.4.2.). Бухгалтери мають доступ тільки до таблиці з полісами та виплатами, а ось агенти мають доступ до всієї інформації, яка тільки є.

```
-- Створення ролі для бухгалтерії
CREATE ROLE Buchalteriya;
-- Надання доступу до таблиці Policies
GRANT SELECT ON Policies TO Buchalteriya;
-- Надання доступу до таблиці InsuranceEvent
GRANT SELECT ON InsuranceEvent TO Buchalteriya;

-- Створення користувачів та надання їм ролі "Бухгалтерія"
CREATE LOGIN Bugalter1 WITH PASSWORD = '1234';
CREATE USER Bugalter1 FOR LOGIN Bugalter1;
ALTER ROLE Buchalteriya ADD MEMBER Bugalter1;

CREATE LOGIN Bugalter2 WITH PASSWORD = '4321';
CREATE USER Bugalter2 FOR LOGIN Bugalter2;
ALTER ROLE Buchalteriya ADD MEMBER Bugalter2;
```

Рис. 4.1. Створення ролі Buchalteriya та 2 користувачів з цією роллю

```
-- Створення ролі для агентів
CREATE ROLE AgentRole;

-- Надання доступу до всіх таблиць (Clients, Agents, Policies, InsuranceAuto, InsuranceHome, InsuranceLife, InsuranceEvent)
GRANT SELECT ON Clients TO AgentRole;
GRANT SELECT ON Agents TO AgentRole;
GRANT SELECT ON Policies TO AgentRole;
GRANT SELECT ON InsuranceAuto TO AgentRole;
GRANT SELECT ON InsuranceHome TO AgentRole;
GRANT SELECT ON InsuranceLife TO AgentRole;
GRANT SELECT ON InsuranceEvent TO AgentRole;

-- Створення користувачів та надання їм ролі "Агент"
CREATE LOGIN AgentR1 WITH PASSWORD = '1234';
CREATE USER AgentR1 FOR LOGIN AgentR1;
ALTER ROLE AgentRole ADD MEMBER AgentR1;

CREATE LOGIN AgentR2 WITH PASSWORD = '4321';
CREATE USER AgentR2 FOR LOGIN AgentR2;
ALTER ROLE AgentRole ADD MEMBER AgentR2;
```

Рис. 4.2. Створення ролі AgentRole та 2 користувачів з цією роллю

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

ЛІТЕРАТУРА

1. Microsoft Visual C#. Докладне керівництво. 8-е видання..
2. C# 6.0 in a Nutshell. The Definitive Reference 6th Edition.
3. Гультяїв А.К., Машин В.А. Проектування та дизайн інтерфейсу користувача.
4. Карієв Ч.А. Технологія Microsoft ADO .NET, 2-моделі предметної області вид.
5. Конноллі Т. Бази даних. Проектування, реалізація та супровід.
6. Теорія та практика.
7. Мейєр Д. Теорія реляційних баз даних.
8. Праг К.М. Access 2003. Біблія користувача.
9. Пасічник В.В. Організація баз даних та знань: Підручник. / Пасічник В.В., Резніченко В.А. - К.: ВHV, 2006. - 384с.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

ВИСНОВКИ

В процесі виконання даної курсової роботи я спроектував базу даних “Страхова компанія”, а саме логічну та фізичну структуру. Наступним кроком було виконано реалізацію бази даних у СУБД Microsoft SQL Server Management Studio, а саме: створення бази даних, заповнення даними та реалізація функцій, процедур та тригерів.

Для зручної роботи з створеною базою даних було створене програмне забезпечення на WPF, що реалізує роботу з спроектованою базою даних. Для цього було спроектовано форму та реалізовано відповідний функціонал для неї.

Тестування та демонстрація програмного забезпечення показало, що всі поставлені завдання були реалізовані та працюють без відхилень.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

ДОДАТКИ

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Технічне завдання**1. Загальне положення****1.1. Найменування програмного засобу**

Повне найменування програмної системи: "База даних Страхова компанія" (надалі "програма"). Коротка назва програмної системи - "Страховка"

1.2. Призначення розробки та область застосування

Програмна система "База даних страхової компанії" призначена для ефективного управління даними, пов'язаними з полісами страхування, клієнтами та страховими випадками. Система дозволяє збирати, обробляти та відображати інформацію про страхові угоди, клієнтів та страхові випадки. Крім того, база даних підтримує функції аналізу та звітності для полегшення управління страховими операціями.

Ця система призначена для використання страховими компаніями будь-якого розміру. Вона розрахована на роботу з великою кількістю даних та може обслуговувати базу страхових полісів та клієнтів з досягненням до 100000 записів.

1.3. Мета

Програмна система "База даних Страхової компанії" дозволить підвищити ефективність, повноту та швидкість роботи працівників страхової компанії щодо обробки даних про клієнтів та управління страховими полісами

1.4. Найменування розробника та замовника.

Розробник даного продукту - студент групи ІПЗ-22-3 Леус Вадим Олександрович (надалі "розробник").

Замовник програмного продукту - кафедра програмного забезпечення обчислюваної техніки Житомирського державного технологічного університету в межах виконання курсової з дисципліни «Бази даних» (надалі "замовник").

2. Підстава для розробки**2.1. Документ на підставі якого ведеться розробка**

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Робота ведеться на підставі навчального плану за напрямом 6.050201 «Програмна інженерія» наказу про закріплення тем курсових робіт за студентами.

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик.

3.1.1. Загальні вимоги

Програмна система має забезпечувати:

- можливість дистанційної роботи з робочих станцій локальної та глобальної мережі підприємства;
- інтерфейс користувача, що не залежить від операційної системи;
- постійний доступ користувачів до БД;
- оптимальне збереження даних (за обсягом та структурою);
- аутентифікацію користувачів та захист інформації від несанкціонованого доступу;
- надійне збереження даних та можливість відновлення даних у випадку непередбачуваних збоїв системи;
- створення архівів даних, що не використовуються протягом визначеного терміну;
- можливість модернізації системи через зміну функціональних потреб користувача або модернізацію обладнання;

3.1.1. Склад виконуваних функцій

Для моєї курсової роботи на тему "База даних Страхова компанія" потрібно розробити базу даних, яка підтримує виконання наступних операцій:

- Реєстрація клієнтів страхової компанії.
- Збереження даних про страхові поліси клієнтів та їх страхові історії.
- Облік виплат страхових відшкодувань та взаєморозрахунків.
- Ведення статистики та аналіз даних щодо страхових подій.
- Спільний доступ до системи для співробітників страхової компанії та їх керівництва.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

- Захист і збереження конфіденційної інформації клієнтів.

Ця база даних допоможе страховій компанії ефективно ведення її діяльності, включаючи реєстрацію клієнтів, ведення обліку страхових полісів та виплат, аналіз подій та надання доступу для внутрішнього використання. Також буде надано заходи безпеки для збереження конфіденційної інформації клієнтів.

3.1.2. Організація вхідних і вихідних даних

Вхідні дані: Інформація про клієнтів, страхові поліси, страхові випадки та інші важливі дані.

Організація даних: Дані повинні відповідати стандартам та вводитися через діалогові форми на основі затверджених документів.

Звіти та виведення: Забезпечення можливості паперового представлення та друку звітів.

3.1.3. Часові характеристики і розмір пам'яті, необхідної для роботи програми.

Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с.

Час виконання команд меню не більше 1 с.

Відображення масивів даних за запитом не більше 3 хвилин.

Доступність БД – 90% цілодобово.

Операції з'єднання з БД не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи програми не менше 1Гб.

Дисковий простір, необхідний для збереження програми і файлів даних не більше 300 Мбайт для робочої станції та 20 ГБайт.

Інсталяційний пакет програми, що містить у складі БД не повинні перевищувати 100 Мбайт.

3.2. Вимоги до надійності.

3.2.1. Вимоги до надійного функціонування

Програма повинна нормально функціонувати при безперебійній роботі ПК. Доступність БД 90% при одночасному доступі 30 користувачів.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

При апаратних збоях, відновлення нормальної роботи програми повинне виконуватися після:

а) апаратні збої сервера - перезавантаження ОС сервера, запуск сервера БД (запуску резервного сервера, використання технологій RAID для збереження даних);

б) апаратні збої робочої станції – перезавантаження ОС ПК, запуск виконуваного файлу програми.

в) БД повинна повертатись в найближчий несуперечний стан – передбачити точки відновлення.

При збоях програмного забезпечення:

а) система повинна забезпечувати можливість відновлення даних та фіксацію і «відкат» транзакцій.

б) в системі має бути реалізована коректна обробка виняткових ситуацій.

3.2.2. Контроль вхідної і вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора:

- Перевірка відповідності даних доменам інформаційних атрибутів;
- Використанням механізму авто заповнення та вибору за переліком для зв'язаних даних;
- Захист від помилок оператора (залипань, випадкових символів тощо).

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування. Для вихідної інформації передбачити:

- відсутність блокування даних через багато користувальницький доступ;
- постійне оновлення даних у відображених на екрані звітних формах.
- повідомлення про використання режиму монопольного доступу супервайзерами БД.

3.2.3. Час відновлення після відмови.

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезавантаження операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.3. Умови експлуатації і збереження

Програма використовується у багато користувальницькому середовищі. Регламенті операції проводити за графіком:

- створення резервних копій даних: 1 раз на тиждень;
- збереження резервної копії журналу транзакцій – 1 раз на тиждень;
- архівування даних – 1 раз на місяць;
- обслуговуючі функції (прибирання “сміття”, де фрагментація тощо) - щоденне;
- оновлення системи ідентифікації та аутентифікації користувачів – 1 раз на місяць та за потребою.

Збереження архівних копій - на визначених носіях. Збереження резервних копій – на визначеному диску без перезапису. Знищення копій за регламентом.

3.4. Вимоги до інформаційної і програмної сумісності.

3.4.1. Вимоги до інформаційних структур на вході і виході

Формат відображення даних має дозволяти імпорт даних в додатки MS Office для редагування та перегляду.

3.4.2. Вимоги до методів рішення і мов програмування

Вибір методів рішення здійснюється розробникам без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

3.4.3. Вимоги до системи програмних засобів.

Вимоги до програмного забезпечення сервера:

ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником).

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

Вимоги до програмного забезпечення робочої станції:

ОС - родина Windows не нижче Windows XP.

Драйвери периферійних пристроїв - введення/виводу визначаються та встановлюються при встановленні ОС в залежності від конфігурації робочої станції.

Пакет додатків MS Office.

Робоча станція є клієнтом мереж Microsoft

3.5. Вимоги до складу і параметрів технічних засобів.

Вимоги до складу технічних засобів:

1.Сервер:

- сервер у базовій конфігурації із підтримкою RAID;
- з'ємний запам'ятовуючий пристрій для архівування даних;
- принтер для друку;
- засоби для під'єднання до локальної мережі.

2.Робоча станція:

- ПК на базі процесорів Intel, AMD у стандартній комплектації;
- периферійні пристрої друку;
- оперативна пам'ять - 128Мбайт;
- об'єм дискової пам'яті - 20Гбайт.

4.Вимоги до програмної документації

Програмна документація повинна включати наступні відомості:

1. "Інструкція по інсталяції (встановленню) програми", складається з опису інсталяційного пакету, переліку етапів інсталяції та їх послідовності, опису встановлених програмних компонентів та режимів їх роботи після інсталяції. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться у 4 розділі.

2. "Керівництво користувача" складається з опису послідовності завантаження програми, основних режимів роботи, опису основних екранних форм, переліку виняткових ситуацій та реакції користувача на них, порядку виконання завдань

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

в системі. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 3 розділі курсової роботи.

3. "Керівництво адміністратора баз даних", складається з опису складу таблиць бази даних та правил доступів до них, опису та послідовності робіт по обслуговуванню бази даних (архівування, резервне копіювання, з вказівкою періодичності виконання та засобів, що для цього використовуються), порядку налаштування серверу та клієнтських додатків.

Додаток Б

Файл MainWindow.xaml.cs:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
using System.Windows;
using System.Windows.Controls;
namespace CourseWork_DB_SIP_WPF
{
    public partial class MainWindow : Window
    {
        private string connectionString = "Data Source=DESKTOP-GGI31RV\\MSSQLSERVERNEW;Initial
Catalog=SIP;Integrated Security=True";
        public MainWindow()
        {
            InitializeComponent();
            LoadData();
        }
        //Завантаження даних в таблиці та оновлення даних в них
        private void LoadData()
        {
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                try
                {
                    connection.Open();

                    string homeQuery = "SELECT * FROM InsuranceHome";
                    SqlDataAdapter homeAdapter = new SqlDataAdapter(homeQuery, connection);
                    DataTable homeTable = new DataTable();
                    homeAdapter.Fill(homeTable);
                    HomeTBL1.ItemsSource = homeTable.DefaultView;

                    string lifeQuery = "SELECT * FROM InsuranceLife";
                    SqlDataAdapter lifeAdapter = new SqlDataAdapter(lifeQuery, connection);
                    DataTable lifeTable = new DataTable();
                    lifeAdapter.Fill(lifeTable);
                    LifeTBL1.ItemsSource = lifeTable.DefaultView;
```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				31
Змн.	Арк.	№ докум.	Підпис	Дата		


```

string autoQuery = "SELECT * FROM InsuranceAuto";
SqlDataAdapter autoAdapter = new SqlDataAdapter(autoQuery, connection);
DataTable autoTable = new DataTable();
autoAdapter.Fill(autoTable);
AutoTBL1.ItemsSource = autoTable.DefaultView;

string agentsQuery = "SELECT * FROM Agents";
SqlDataAdapter agentsAdapter = new SqlDataAdapter(agentsQuery, connection);
DataTable agentsTable = new DataTable();
agentsAdapter.Fill(agentsTable);
AgentTBL1.ItemsSource = agentsTable.DefaultView;

string policiesQuery = "SELECT * FROM Policies";
SqlDataAdapter policiesAdapter = new SqlDataAdapter(policiesQuery, connection);
DataTable policiesTable = new DataTable();
policiesAdapter.Fill(policiesTable);

PoliciesTBL1.ItemsSource = policiesTable.DefaultView;
PoliciesTBL2.ItemsSource = policiesTable.DefaultView;
PoliciesTBL3.ItemsSource = policiesTable.DefaultView;

string clientsQuery = "SELECT * FROM Clients";
SqlDataAdapter clientsAdapter = new SqlDataAdapter(clientsQuery, connection);
DataTable clientsTable = new DataTable();
clientsAdapter.Fill(clientsTable);
ClientTBL1.ItemsSource = clientsTable.DefaultView;

string eventQuery = "SELECT * FROM InsuranceEvent";
SqlDataAdapter eventAdapter = new SqlDataAdapter(eventQuery, connection);
DataTable eventTable = new DataTable();
eventAdapter.Fill(eventTable);
InsuranceEventTBL.ItemsSource = eventTable.DefaultView;
}
catch (Exception ex)
{
    MessageBox.Show("Помилка завантаження даних: " + ex.Message);
}
finally
{
    connection.Close();
}
}
}

// Фільтр полісів по типам
private void InsuranceAutoRB_Checked(object sender, RoutedEventArgs e)
{
    ApplyFilterByType("Страхування автомобіля");
}
private void InsuranceHomeRB_Checked(object sender, RoutedEventArgs e)
{
    ApplyFilterByType("Страхування житла");
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

```

    }
    private void InsuranceLifeRB_Checked(object sender, RoutedEventArgs e)
    {
        ApplyFilterByType("Страхування життя");
    }
    private void ClearFilterBTN_Click(object sender, RoutedEventArgs e)
    {
        InsuranceAutoRB.IsChecked = false;
        InsuranceLifeRB.IsChecked = false;
        InsuranceHomeRB.IsChecked = false;

        ApplyFilterByType("");
        ApplyFilterByDates(null, null);
        LoadData();
    }
    private void ApplyFilterByType(string filter)
    {
        string queryPolicies = "SELECT * FROM Policies";

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            try
            {
                connection.Open();

                if (!string.IsNullOrEmpty(filter))
                {
                    queryPolicies += " WHERE PoliciesType = @Filter";
                }

                SqlDataAdapter adapterPolicies = new SqlDataAdapter(queryPolicies, connection);

                if (!string.IsNullOrEmpty(filter))
                {
                    adapterPolicies.SelectCommand.Parameters.AddWithValue("@Filter", filter);
                }

                DataTable dataTablePolicies = new DataTable();
                adapterPolicies.Fill(dataTablePolicies);
                PoliciesTBL1.ItemsSource = dataTablePolicies.DefaultView;
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка завантаження даних: " + ex.Message);
            }
            finally
            {
                connection.Close();
            }
        }
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

// Фільтр полісів за датами
private void DatePicker1_SelectedDateChanged(object sender, SelectionChangedEventArgs e)
{
    ApplyFilterByDates(DatePicker1.SelectedDate, DatePicker2.SelectedDate);
}
private void DatePicker2_SelectedDateChanged(object sender, SelectionChangedEventArgs e)
{
    ApplyFilterByDates(DatePicker1.SelectedDate, DatePicker2.SelectedDate);
}
private void ClearFilterDataBTN_Click(object sender, RoutedEventArgs e)
{
    DatePicker1.SelectedDate = null;
    DatePicker2.SelectedDate = null;

    ApplyFilterByType("");
    ApplyFilterByDates(null, null);
    LoadData();
}
private void ApplyFilterByDates(DateTime? startDate, DateTime? endDate)
{
    string queryPolicies = "SELECT * FROM Policies WHERE 1=1";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();

            if (startDate != null && endDate != null)
            {
                queryPolicies += " AND StartDate >= @StartDate AND EndDate <= @EndDate";
            }

            SqlDataAdapter adapterPolicies = new SqlDataAdapter(queryPolicies, connection);

            if (startDate != null && endDate != null)
            {
                adapterPolicies.SelectCommand.Parameters.AddWithValue("@StartDate", startDate);
                adapterPolicies.SelectCommand.Parameters.AddWithValue("@EndDate", endDate);
            }

            DataTable dataTablePolicies = new DataTable();
            adapterPolicies.Fill(dataTablePolicies);
            PoliciesTBL1.ItemsSource = dataTablePolicies.DefaultView;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка завантаження даних: " + ex.Message);
        }
        finally
        {
            connection.Close();
        }
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

```

// Основні взаємодії з табл агентів
private void SaveAgentBTN_Click(object sender, RoutedEventArgs e)
{
    string agentFiName = AgentFiTB.Text;
    string agentSeName = AgentSeTB.Text;
    string agentFaName = AgentFaTB.Text;
    string agentPhone = AgentPhoTB.Text;
    string agentPost = AgentPostTB.Text;
    string agentID = AgentIDTB.Text;

    if (string.IsNullOrEmpty(agentFiName) || string.IsNullOrEmpty(agentSeName) ||
        string.IsNullOrEmpty(agentFaName) || string.IsNullOrEmpty(agentPhone) ||
        string.IsNullOrEmpty(agentPost) || string.IsNullOrEmpty(agentID))
    {
        MessageBox.Show("Будь ласка, заповніть всі поля.");
        return;
    }

    if (!Regex.IsMatch(agentPhone, @"^\+\d{3}\s\d{2}\s\d{3}\s\d{4}$"))
    {
        MessageBox.Show("Невірний формат номера телефону. Використовуйте формат: +XXX XX XXX XXXX");
        return;
    }

    if (!IsUpperCase(agentFiName) || !IsUpperCase(agentSeName) || !IsUpperCase(agentFaName))
    {
        MessageBox.Show("Ім'я, по-батькові або прізвище мають бути написані з великої літери.");
        return;
    }

    if (IsAgentIDExists(agentID) || IsAgentPhoneExists(agentPhone))
    {
        MessageBox.Show("Агент з таким ID або номером телефона вже існує!");
        return;
    }

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            string query = "INSERT INTO Agents (IdAgent, AgentFiName, AgentSeName, AgentFaName, AgentPhone, AgentPost) " +
                "VALUES (@IdAgent, @AgentFiName, @AgentSeName, @AgentFaName, @AgentPhone, @AgentPost)";
            SqlCommand cmd = new SqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@IdAgent", agentID);
            cmd.Parameters.AddWithValue("@AgentFiName", agentFiName);
            cmd.Parameters.AddWithValue("@AgentSeName", agentSeName);
            cmd.Parameters.AddWithValue("@AgentFaName", agentFaName);

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

```

cmd.Parameters.AddWithValue("@AgentPhone", agentPhone);
cmd.Parameters.AddWithValue("@AgentPost", agentPost);

cmd.ExecuteNonQuery();
MessageBox.Show("Агент успішно доданий!");
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при додаванні агента: " + ex.Message);
}
finally
{
    connection.Close();
}
}
LoadData();
}
private void AsseptChangeAgentBTN_Click(object sender, RoutedEventArgs e)
{
    string agentID = AgentIDTB.Text;
    string agentFiName = AgentFiTB.Text;
    string agentSeName = AgentSeTB.Text;
    string agentFaName = AgentFaTB.Text;
    string agentPhone = AgentPhoTB.Text;
    string agentPost = AgentPostTB.Text;

    if (string.IsNullOrEmpty(agentFiName) || string.IsNullOrEmpty(agentSeName) ||
string.IsNullOrEmpty(agentFaName) ||
        string.IsNullOrEmpty(agentPhone) || string.IsNullOrEmpty(agentPost) ||
string.IsNullOrEmpty(agentID))
    {
        MessageBox.Show("Будь ласка, заповніть всі поля.");
        return;
    }

    if (!Regex.IsMatch(agentPhone, @"^\+\d{3}\s\d{2}\s\d{3}\s\d{4}$"))
    {
        MessageBox.Show("Невірний формат номера телефону. Використовуйте формат: +XXX XX XXX
XXXX");
        return;
    }

    if (!IsUpperCase(agentFiName) || !IsUpperCase(agentSeName) || !IsUpperCase(agentFaName))
    {
        MessageBox.Show("Ім'я, по-батькові або прізвище мають бути написані з великої літери.");
        return;
    }

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

```

        string query = "UPDATE Agents SET AgentFiName = @AgentFiName, AgentSeName =
@AgentSeName, " +
        "AgentFaName = @AgentFaName, AgentPhone = @AgentPhone, AgentPost = @AgentPost "
+
        "WHERE IdAgent = @AgentID";

SqlCommand cmd = new SqlCommand(query, connection);
cmd.Parameters.AddWithValue("@AgentFiName", agentFiName);
cmd.Parameters.AddWithValue("@AgentSeName", agentSeName);
cmd.Parameters.AddWithValue("@AgentFaName", agentFaName);
cmd.Parameters.AddWithValue("@AgentPhone", agentPhone);
cmd.Parameters.AddWithValue("@AgentPost", agentPost);
cmd.Parameters.AddWithValue("@AgentID", agentID);

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)
{
    MessageBox.Show("Дані агента успішно змінені!");
    FilterAgentDG();
    ClearAgentFields();
}
else
{
    MessageBox.Show("Не вдалося змінити дані агента.");
}
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при зміні даних агента: " + ex.Message);
}
finally
{
    connection.Close();
}
}

private void DeleteAgentBTN_Click(object sender, RoutedEventArgs e)
{
    string agentIDToDelete = AgentIDTB.Text;
    if (string.IsNullOrEmpty(agentIDToDelete))
    {
        MessageBox.Show("Введіть ID агента для видалення.");
        return;
    }

    MessageBoxResult result = MessageBox.Show("Ви впевнені, що хочете видалити агента з цим ID?",
"Видалення агента", MessageBoxButton.YesNo, MessageBoxImage.Question);

    if (result == MessageBoxResult.Yes)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            SqlTransaction transaction = null;

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

```

try
{
    connection.Open();
    transaction = connection.BeginTransaction();

    string deletePoliciesQuery = "DELETE FROM Policies WHERE IdAgent = @AgentID";
    SqlCommand deletePoliciesCmd = new SqlCommand(deletePoliciesQuery, connection, transaction);
    deletePoliciesCmd.Parameters.AddWithValue("@AgentID", agentIDToDelete);
    deletePoliciesCmd.ExecuteNonQuery();

    string deleteAgentQuery = "DELETE FROM Agents WHERE IdAgent = @AgentID";
    SqlCommand deleteAgentCmd = new SqlCommand(deleteAgentQuery, connection, transaction);
    deleteAgentCmd.Parameters.AddWithValue("@AgentID", agentIDToDelete);
    int rowsAffected = deleteAgentCmd.ExecuteNonQuery();

    transaction.Commit();

    if (rowsAffected > 0)
    {
        MessageBox.Show("Агент і пов'язані з ним поліси були видалені.");
        LoadData();
    }
    else
    {
        MessageBox.Show("Агент з таким ID не знайдений.");
    }
}
catch (Exception ex)
{
    transaction?.Rollback();
    MessageBox.Show("Помилка при видаленні агента та його полісів " + ex.Message);
}
finally
{
    connection.Close();
}
}

private bool IsUpperCase(string input)
{
    return char.IsUpper(input[0]);
}

private bool IsAgentIDExists(string agentID)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        SqlCommand command = new SqlCommand("SELECT COUNT(*) FROM Agents WHERE IdAgent = @AgentID", connection);
        command.Parameters.AddWithValue("@AgentID", agentID);

        int count = (int)command.ExecuteScalar();
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return count > 0;
    }
}
private bool IsAgentPhoneExists(string agentPhone)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        SqlCommand command = new SqlCommand("SELECT COUNT(*) FROM Agents WHERE AgentPhone
= @AgentPhone", connection);
        command.Parameters.AddWithValue("@AgentPhone", agentPhone);

        int count = (int)command.ExecuteScalar();

        return count > 0;
    }
}
private void AgentTBL2_SelectionChanged(object sender,
System.Windows.Controls.SelectionChangedEventArgs e)
{
    if (AgentTBL1.SelectedItem != null && AgentTBL1.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)AgentTBL1.SelectedItem;

        MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього робітника?", "Зміна
даних", MessageBoxButton.YesNo, MessageBoxImage.Question);

        if (result == MessageBoxResult.Yes)
        {
            AgentIDTB.Text = row["IdAgent"].ToString();
            AgentFiTB.Text = row["AgentFiName"].ToString();
            AgentSeTB.Text = row["AgentSeName"].ToString();
            AgentFaTB.Text = row["AgentFaName"].ToString();
            AgentPhoTB.Text = row["AgentPhone"].ToString();
            AgentPostTB.Text = row["AgentPost"].ToString();
        }
    }
    else
    {
        MessageBox.Show("Тут ще немає робітника.");
    }
}

// Фільтр даних агентів по прізвищу або номеру
private void FilterAgentDG()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            string query = "SELECT * FROM Agents";
            if (AgentPhoneRB.IsChecked == true)
            {

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		39


```

        query += " WHERE AgentPhone LIKE @SearchText";
    }
    else if (AgentSeRB.IsChecked == true)
    {
        query += " WHERE AgentSeName LIKE @SearchText";
    }

    SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
    adapter.SelectCommand.Parameters.AddWithValue("@SearchText", "%" + SeachAgentTB.Text +
"%");

    DataTable dataTable = new DataTable();
    adapter.Fill(dataTable);

    AgentTBL1.ItemsSource = dataTable.DefaultView;
    FilterPoliciesAGEDG();
}
catch (Exception ex)
{
    MessageBox.Show("Помилка оновлень даних агентів: " + ex.Message);
}
finally
{
    connection.Close();
}
}
}
private void FilterPoliciesAGEDG()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            string query = "SELECT Policies.* FROM Policies INNER JOIN Agents ON Policies.IdAgent =
Agents.IdAgent";
            if (AgentPhoneRB.IsChecked == true)
            {
                query += " WHERE Agents.AgentPhone LIKE @SearchText";
            }
            else if (AgentSeRB.IsChecked == true)
            {
                query += " WHERE Agents.AgentSeName LIKE @SearchText";
            }

            SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
            adapter.SelectCommand.Parameters.AddWithValue("@SearchText", "%" + SeachAgentTB.Text +
"%");

            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            PoliciesTBL2.ItemsSource = dataTable.DefaultView;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка оновлення даних полісів: " + ex.Message);
        }
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    finally
    {
        connection.Close();
    }
}

// Фільтр агентів рб і тб
private void AgentSeRB_Checked(object sender, RoutedEventArgs e)
{
    FilterAgentDG();
    FilterPoliciesAGEDG();
}
private void AgentPhoneRB_Checked(object sender, RoutedEventArgs e)
{
    FilterAgentDG();
    FilterPoliciesAGEDG();
}
private void ClearFilterAgentBTN_Click(object sender, RoutedEventArgs e)
{
    AgentPhoneRB.IsChecked = false;
    AgentSeRB.IsChecked = false;
    FilterAgentDG();
    FilterPoliciesAGEDG();
}
private void SeachAgentTB_TextChanged(object sender, System.Windows.Controls.TextChangedEventArgs e)
{
    FilterAgentDG();
    FilterPoliciesAGEDG();
}
private void ClearAgentFields()
{
    AgentIDTB.Text = "";
    AgentFiTB.Text = "";
    AgentSeTB.Text = "";
    AgentFaTB.Text = "";
    AgentPhoTB.Text = "";
    AgentPostTB.Text = "";
}

// Основні взаємодії з табл клієнтів
private void SaveClientBTN_Click(object sender, RoutedEventArgs e)
{
    string idClients = ClientIDTB.Text;
    string clientFiName = ClientFiTB.Text;
    string clientSeName = ClientSeTB.Text;
    string clientFaName = ClientFaTB.Text;
    string clientPhone = ClientPhoTB.Text;
    string clientAddress = ClientAddressTB.Text;

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

```

if (!Regex.IsMatch(clientPhone, @"^\+\d{3}\s\d{2}\s\d{3}\s\d{4}$"))
{
    MessageBox.Show("Невірний формат номера телефону. Використовуйте формат: +XXX XX XXX
XXXX");
    return;
}
if (!Regex.IsMatch(clientAddress, @"вул.\s\w+,\s\d+,\s\d+"))
{
    MessageBox.Show("Невірний формат адреса. Використовуйте формат: вул. XXXX, NN, YYY");
    return;
}

using (SqlConnection connection = new SqlConnection(connectionString))
{
    try
    {
        connection.Open();
        string checkExistingQuery = "SELECT COUNT(*) FROM Clients WHERE IdClients = @IdClients
OR ClientPhone = @ClientPhone";
        SqlCommand checkExistingCmd = new SqlCommand(checkExistingQuery, connection);
        checkExistingCmd.Parameters.AddWithValue("@IdClients", idClients);
        checkExistingCmd.Parameters.AddWithValue("@ClientPhone", clientPhone);

        int existingCount = (int)checkExistingCmd.ExecuteScalar();

        if (existingCount > 0)
        {
            MessageBox.Show("Клієнт з таким же ID або номером телефона вже існує! Змініть дані.");
            return;
        }
        string insertQuery = "INSERT INTO Clients (IdClients, ClientFiName, ClientSeName, ClientFaName,
ClientPhone, ClientAddress) " +
            "VALUES (@IdClients, @ClientFiName, @ClientSeName, @ClientFaName,
@ClientPhone, @ClientAddress)";

        SqlCommand cmd = new SqlCommand(insertQuery, connection);
        cmd.Parameters.AddWithValue("@IdClients", idClients);
        cmd.Parameters.AddWithValue("@ClientFiName", clientFiName);
        cmd.Parameters.AddWithValue("@ClientSeName", clientSeName);
        cmd.Parameters.AddWithValue("@ClientFaName", clientFaName);
        cmd.Parameters.AddWithValue("@ClientPhone", clientPhone);
        cmd.Parameters.AddWithValue("@ClientAddress", clientAddress);

        cmd.ExecuteNonQuery();
        MessageBox.Show("Клієнт успішно доданий!");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка при додаванні клієнта: " + ex.Message);
    }
    finally
    {
        connection.Close();
    }
    FilterClientDG();
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

```

        ClearClientFields();
    }
}
private void AsseptChangeClientBTN_Click(object sender, RoutedEventArgs e)
{
    string IdClients = ClientIDTB.Text;
    string clientFiName = ClientFiTB.Text;
    string clientSeName = ClientSeTB.Text;
    string clientFaName = ClientFaTB.Text;
    string clientPhone = ClientPhoTB.Text;
    string clientAddress = ClientAddressTB.Text;

    if (!Regex.IsMatch(clientPhone, @"^\+\d{3}\s\d{2}\s\d{3}\s\d{4}$"))
    {
        MessageBox.Show("Невірний формат номера телефону. Використовуйте формат: +XXX XX XXX XXXX");
        return;
    }
    if (!Regex.IsMatch(clientAddress, @"^вул.\s\w+,\s\d+,\s\d+$"))
    {
        MessageBox.Show("Невірний формат адреса. Використовуйте формат: вул. XXXX, NN, YYY");
        return;
    }
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();

            string query = "UPDATE Clients SET ClientFiName = @ClientFiName, ClientSeName = @ClientSeName, " +
                "ClientFaName = @ClientFaName, ClientPhone = @ClientPhone, ClientAddress = @ClientAddress " +
                "WHERE IdClients = @IdClients";

            SqlCommand cmd = new SqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@IdClients", IdClients);
            cmd.Parameters.AddWithValue("@ClientFiName", clientFiName);
            cmd.Parameters.AddWithValue("@ClientSeName", clientSeName);
            cmd.Parameters.AddWithValue("@ClientFaName", clientFaName);
            cmd.Parameters.AddWithValue("@ClientPhone", clientPhone);
            cmd.Parameters.AddWithValue("@ClientAddress", clientAddress);

            int rowsAffected = cmd.ExecuteNonQuery();
            if (rowsAffected > 0)
            {
                MessageBox.Show("Дані клієнта успішно змінені!");
                FilterClientDG();
                ClearClientFields();
            }
            else
            {
                MessageBox.Show("Не вдалося змінити дані клієнта.");
            }
        }
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

```

        catch (Exception ex)
        {
            MessageBox.Show("Помилка при зміні даних клієнта: " + ex.Message);
        }
        finally
        {
            connection.Close();
        }
    }
}

private void DeleteClientBTN_Click(object sender, RoutedEventArgs e)
{
    string clientIDToDelete = ClientIDTB.Text;
    if (string.IsNullOrEmpty(clientIDToDelete))
    {
        MessageBox.Show("Введіть ID клієнта для видалення.");
        return;
    }

    MessageBoxResult result = MessageBox.Show("Ви впевнені, що хочете видалити цього клієнта?",
        "Підтвердження видалення", MessageBoxButton.YesNo, MessageBoxImage.Question);

    if (result == MessageBoxResult.Yes)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            SqlTransaction transaction = null;

            try
            {
                connection.Open();
                transaction = connection.BeginTransaction();

                string deletePoliciesQuery = "DELETE FROM Policies WHERE IdClients = @IdClients";
                SqlCommand deletePoliciesCmd = new SqlCommand(deletePoliciesQuery, connection, transaction);
                deletePoliciesCmd.Parameters.AddWithValue("@IdClients", clientIDToDelete);
                deletePoliciesCmd.ExecuteNonQuery();

                string deleteClientQuery = "DELETE FROM Clients WHERE IdClients = @IdClients";
                SqlCommand deleteClientCmd = new SqlCommand(deleteClientQuery, connection, transaction);
                deleteClientCmd.Parameters.AddWithValue("@IdClients", clientIDToDelete);
                int rowsAffected = deleteClientCmd.ExecuteNonQuery();

                transaction.Commit();

                if (rowsAffected > 0)
                {
                    MessageBox.Show("Клієнт і пов'язані з ним поліси були видалені.");
                    FilterClientDG();
                }
                else
                {
                    MessageBox.Show("Клієнт з таким ID не знайдений.");
                }
            }
        }
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        catch (Exception ex)
        {
            transaction?.Rollback();
            MessageBox.Show("Помилка при видаленні клієнта та його полісів " + ex.Message);
        }
        finally
        {
            connection.Close();
        }
    }
}

private void ClientTBL_SelectionChanged(object sender,
System.Windows.Controls.SelectionChangedEventArgs e)
{
    if (ClientTBL1.SelectedItem != null && ClientTBL1.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)ClientTBL1.SelectedItem;

        MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього клієнта?", "Зміна даних",
        MessageBoxButton.YesNo, MessageBoxImage.Question);

        if (result == MessageBoxResult.Yes)
        {
            ClientIDTB.Text = row["IdClients"].ToString();
            ClientFiTB.Text = row["ClientFiName"].ToString();
            ClientSeTB.Text = row["ClientSeName"].ToString();
            ClientFaTB.Text = row["ClientFaName"].ToString();
            ClientPhoTB.Text = row["ClientPhone"].ToString();
            ClientAddressTB.Text = row["ClientAddress"].ToString();
        }
    }
    else
    {
        MessageBox.Show("Тут ще немає клієнта.");
    }
}

// Фільтр даних клієнтів по прізвищу або номеру
private void FilterClientDG()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            string query = "SELECT * FROM Clients";
            if (ClientPhoneRB.IsChecked == true)
            {
                query += " WHERE ClientPhone LIKE @SearchText";
            }
            else if (ClientSeRB.IsChecked == true)
            {
                query += " WHERE ClientSeName LIKE @SearchText";
            }
        }
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
        adapter.SelectCommand.Parameters.AddWithValue("@SearchText", "%" + SeachClientTB.Text +
"%");

        DataTable dataTable = new DataTable();
        adapter.Fill(dataTable);

        ClientTBL1.ItemsSource = dataTable.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка оновлень даних клієнтів: " + ex.Message);
    }
    finally
    {
        connection.Close();
    }
}
}
private void FilterPoliciesCLIDG()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            string query = "SELECT Policies.* FROM Policies INNER JOIN Clients ON Policies.IdClients =
Clients.IdClients";
            if (ClientPhoneRB.IsChecked == true)
            {
                query += " WHERE Clients.ClientPhone LIKE @SearchText";
            }
            else if (ClientSeRB.IsChecked == true)
            {
                query += " WHERE Clients.ClientSeName LIKE @SearchText";
            }

            SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
            adapter.SelectCommand.Parameters.AddWithValue("@SearchText", "%" + SeachClientTB.Text +
"%");

            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            PoliciesTBL3.ItemsSource = dataTable.DefaultView;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка оновлення даних полісів: " + ex.Message);
        }
        finally
        {
            connection.Close();
        }
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

    }
}

// Фільтр клієнтів рб і тб
private void ClientSeRB_Checked(object sender, RoutedEventArgs e)
{
    FilterClientDG();
    FilterPoliciesCLIDG();
}
private void ClientPhoneRB_Checked(object sender, RoutedEventArgs e)
{
    FilterClientDG();
    FilterPoliciesCLIDG();
}
private void ClearFilterClientBTN_Click(object sender, RoutedEventArgs e)
{
    ClientPhoneRB.IsChecked = false;
    ClientSeRB.IsChecked = false;
    LoadData();
}
private void SeachClientTB_TextChanged(object sender, System.Windows.Controls.TextChangedEventArgs e)
{
    FilterClientDG();
    FilterPoliciesCLIDG();
}
private void ClearClientFields()
{
    ClientFiTB.Text = "";
    ClientSeTB.Text = "";
    ClientFaTB.Text = "";
    ClientPhoTB.Text = "";
    ClientAddressTB.Text = "";
}

// Взаємодія з табл Policies
private void ChangePolBTN_Click(object sender, RoutedEventArgs e)
{
    if (!string.IsNullOrEmpty(IdPoliciesNewTB.Text) && !string.IsNullOrEmpty(SummaPolNewTB.Text)
        && !string.IsNullOrEmpty(IdAgentNewTB.Text) && !string.IsNullOrEmpty(IdClientNewTB.Text)
        && DateStartDP.SelectedDate != null && DateEndDP.SelectedDate != null)
    {
        int idPolicies, idAgent, idClient;
        decimal summaPol;
        DateTime startDate, endDate;

        if (!int.TryParse(IdPoliciesNewTB.Text, out idPolicies) || !decimal.TryParse(SummaPolNewTB.Text, out
summaPol)
            || !int.TryParse(IdAgentNewTB.Text, out idAgent) || !int.TryParse(IdClientNewTB.Text, out idClient)
            || !DateTime.TryParse(DateStartDP.Text, out startDate) || !DateTime.TryParse(DateEndDP.Text, out
endDate))
        {

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				47
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        MessageBox.Show("Невірний формат введених даних.");
        return;
    }
    if (summaPol <= 0)
    {
        MessageBox.Show("Сума поліса повина бути додатковим числом.");
        return;
    }
    if (!AgentExists(idAgent))
    {
        MessageBox.Show("Агента з таким ID не існує.");
        return;
    }
    if (!ClientExists(idClient))
    {
        MessageBox.Show("Клієнта з таким ID не існує.");
        return;
    }
    if (startDate > endDate)
    {
        MessageBox.Show("Дата початку не може бути пізніше за дату завершення.");
        return;
    }

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            string query = "UPDATE Policies SET StartDate = @StartDate, EndDate = @EndDate, SummaPol = @SummaPol, IdAgent = @IdAgent, IdClients = @IdClients WHERE IdPolicies = @IdPolicies";

            SqlCommand cmd = new SqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@StartDate", startDate);
            cmd.Parameters.AddWithValue("@EndDate", endDate);
            cmd.Parameters.AddWithValue("@SummaPol", summaPol);
            cmd.Parameters.AddWithValue("@IdAgent", idAgent);
            cmd.Parameters.AddWithValue("@IdClients", idClient);
            cmd.Parameters.AddWithValue("@IdPolicies", idPolicies);

            int rowsAffected = cmd.ExecuteNonQuery();

            if (rowsAffected > 0)
            {
                string updateStatusQuery = "UPDATE Policies SET PolicyStatus = " +
                    "CASE " +
                    "WHEN GETDATE() < StartDate THEN 'Неактивний' " +
                    "WHEN GETDATE() BETWEEN StartDate AND EndDate THEN 'Активний' "
                +
                    "ELSE 'Прострочений' " +
                    "END " +
                    "WHERE IdPolicies = @IdPolicies";

                SqlCommand updateStatusCmd = new SqlCommand(updateStatusQuery, connection);
                updateStatusCmd.Parameters.AddWithValue("@IdPolicies", idPolicies);

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

```

        updateStatusCmd.ExecuteNonQuery();

        MessageBox.Show("Дані поліса успішно змінені!");
        LoadData();
    }
    else
    {
        MessageBox.Show("Не вдалося змінити поліс.");
    }
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при зміні даних поліса: " + ex.Message);
}
finally
{
    connection.Close();
}
}
else
{
    MessageBox.Show("Заповніть усі поля для зміни даних.");
}
}
private bool AgentExists(int agentId)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        string query = "SELECT COUNT(*) FROM Agents WHERE IdAgent = @AgentId";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@AgentId", agentId);
            connection.Open();
            int count = (int)command.ExecuteScalar();
            return count > 0;
        }
    }
}
private bool ClientExists(int clientId)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        string query = "SELECT COUNT(*) FROM Clients WHERE IdClients = @ClientId";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@ClientId", clientId);
            connection.Open();
            int count = (int)command.ExecuteScalar();
            return count > 0;
        }
    }
}
private void DeletePolBTN_Click(object sender, RoutedEventArgs e)
{

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (!string.IsNullOrEmpty(IdPoliciesNewTB.Text) && int.TryParse(IdPoliciesNewTB.Text, out int
idPolicies))
        {
            MessageBoxResult result = MessageBox.Show("Ви впевнені, що хочете видалити цей поліс?",
"Підтвердження видалення", MessageBoxButton.YesNo, MessageBoxImage.Question);

            if (result == MessageBoxResult.Yes)
            {
                int errorCount = 0;

                using (SqlConnection connection = new SqlConnection(connectionString))
                {
                    try
                    {
                        connection.Open();
                        SqlCommand checkAutoCommand = new SqlCommand("SELECT IdPolicies FROM
InsuranceAuto WHERE IdPolicies = @IdPolicies;", connection);
                        checkAutoCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
                        object autoResult = checkAutoCommand.ExecuteScalar();

                        if (autoResult != null)
                        {
                            SqlCommand deleteAutoCommand = new SqlCommand("DELETE FROM InsuranceAuto
WHERE IdPolicies = @IdPolicies;", connection);
                            deleteAutoCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
                            deleteAutoCommand.ExecuteNonQuery();
                        }
                        else
                        {
                            errorCount++;
                        }

                        SqlCommand checkLifeCommand = new SqlCommand("SELECT IdPolicies FROM
InsuranceLife WHERE IdPolicies = @IdPolicies;", connection);
                        checkLifeCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
                        object lifeResult = checkLifeCommand.ExecuteScalar();

                        if (lifeResult != null)
                        {
                            SqlCommand deleteLifeCommand = new SqlCommand("DELETE FROM InsuranceLife
WHERE IdPolicies = @IdPolicies;", connection);
                            deleteLifeCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
                            deleteLifeCommand.ExecuteNonQuery();
                        }
                        else
                        {
                            errorCount++;
                        }

                        SqlCommand checkHomeCommand = new SqlCommand("SELECT IdPolicies FROM
InsuranceHome WHERE IdPolicies = @IdPolicies;", connection);
                        checkHomeCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
                        object homeResult = checkHomeCommand.ExecuteScalar();

                        if (homeResult != null)
                        {

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        SqlCommand deleteHomeCommand = new SqlCommand("DELETE FROM InsuranceHome
WHERE IdPolicies = @IdPolicies;", connection);
        deleteHomeCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
        deleteHomeCommand.ExecuteNonQuery();
    }
    else
    {
        errorCount++;
    }

    SqlCommand deletePoliciesCommand = new SqlCommand("DELETE FROM Policies WHERE
IdPolicies = @IdPolicies;", connection);
    deletePoliciesCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
    deletePoliciesCommand.ExecuteNonQuery();

    if (errorCount == 3)
    {
        MessageBox.Show("Не вдалося видалити поліс.");
    }
    else
    {
        MessageBox.Show("Дані успішно видалені.");
        LoadData();
    }
}
catch (Exception ex)
{
    MessageBox.Show("Помилка видалення даних: " + ex.Message);
}
finally
{
    connection.Close();
}
}
}
else
{
    MessageBox.Show("Введіть Id поліса для видалення.");
}
}

private void PoliciesTBL4_SelectionChanged(object sender,
System.Windows.Controls.SelectionChangedEventArgs e)
{
    if (PoliciesTBL1.SelectedItem != null && PoliciesTBL1.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)PoliciesTBL1.SelectedItem;
        if (row["IdPolicies"] != DBNull.Value && row["SummaPol"] != DBNull.Value &&
            row["IdAgent"] != DBNull.Value && row["IdClients"] != DBNull.Value &&
            row["StartDate"] != DBNull.Value && row["EndDate"] != DBNull.Value)
        {
            MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього поліса?", "Зміна
данних", MessageBoxButton.YesNo, MessageBoxImage.Question);
            if (result == MessageBoxResult.Yes)
            {

```

		Лейс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

```

        IdPoliciesNewTB.Text = row["IdPolicies"].ToString();
        SummaPolNewTB.Text = row["SummaPol"].ToString();
        IdAgentNewTB.Text = row["IdAgent"].ToString();
        IdClientNewTB.Text = row["IdClients"].ToString();
        DateStartDP.SelectedDate = (DateTime)row["StartDate"];
        DateEndDP.SelectedDate = (DateTime)row["EndDate"];
    }
}
else
{
    MessageBox.Show("Тут ще немає поліса.");
}
}
}

```

// Взаємодія з табл InsuranceAuto

```
private void AddPoliciesAutoBTN_Click(object sender, RoutedEventArgs e)
```

```

{
    if (ValidateInput())
    {
        int idPolicies, idAgent, idClient;
        decimal summaPol, autoCost;
        DateTime startDate, endDate;

        if (!int.TryParse(IdPoliciesNewTB.Text, out idPolicies) ||
            !decimal.TryParse(SummaPolNewTB.Text, out summaPol) ||
            !int.TryParse(IdAgentNewTB.Text, out idAgent) ||
            !int.TryParse(IdClientNewTB.Text, out idClient) ||
            !DateTime.TryParse(DateStartDP.Text, out startDate) ||
            !DateTime.TryParse(DateEndDP.Text, out endDate) ||
            !decimal.TryParse(AutoCostTB.Text, out autoCost))
        {
            MessageBox.Show("Невірний формат введених даних.");
            return;
        }

        string autoNumber = AutoNumberTB.Text;
        int autoYear;

        if (!int.TryParse(AutoYearTB.Text, out autoYear))
        {
            MessageBox.Show("Невірний формат року виробництва авто.");
            return;
        }
        if (autoYear < 1970 || autoYear > DateTime.Now.Year)
        {
            MessageBox.Show("Рік виробництва автомобіля має бути після 1970 і не більше поточного року.");
            return;
        }
        if (startDate > endDate)
        {
            MessageBox.Show("Дата початку не може бути пізніше за дату завершення.");
            return;
        }
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

```

if (summaPol <= 0 || autoCost <= 0)
{
    MessageBox.Show("Сумма поліса та вартість авто мають бути додатними числами.");
    return;
}
string status;
DateTime currentDate = DateTime.Now;
if (currentDate < startDate)
{
    status = "Неактивний";
}
else if (currentDate >= startDate && currentDate <= endDate)
{
    status = "Активний";
}
else
{
    status = "Прострочений";
}
string insertPoliciesQuery = "INSERT INTO Policies (IdPolicies, StartDate, EndDate, SummaPol,
PoliciesType, IdClients, IdAgent, PolicyStatus) " +
"VALUES (@IdPolicies, @StartDate, @EndDate, @SummaPol, @PoliciesType,
@IdClients, @IdAgent, @PolicyStatus)";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    try
    {
        connection.Open();

        string checkPolicyQuery = "SELECT COUNT(*) FROM Policies WHERE IdPolicies =
@IdPolicies";
        SqlCommand checkPolicyCommand = new SqlCommand(checkPolicyQuery, connection);
        checkPolicyCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);

        int existingPoliciesCount = (int)checkPolicyCommand.ExecuteScalar();

        if (existingPoliciesCount > 0)
        {
            MessageBox.Show("Поліс з таким ID вже існує.");
            return;
        }

        SqlCommand insertPoliciesCommand = new SqlCommand(insertPoliciesQuery, connection);
        insertPoliciesCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
        insertPoliciesCommand.Parameters.AddWithValue("@StartDate", startDate);
        insertPoliciesCommand.Parameters.AddWithValue("@EndDate", endDate);
        insertPoliciesCommand.Parameters.AddWithValue("@SummaPol", summaPol);
        insertPoliciesCommand.Parameters.AddWithValue("@PoliciesType", "Страхування автомобіля");
        insertPoliciesCommand.Parameters.AddWithValue("@IdClients", idClient);
        insertPoliciesCommand.Parameters.AddWithValue("@IdAgent", idAgent);
        insertPoliciesCommand.Parameters.AddWithValue("@PolicyStatus", status);

        int rowsAffected = insertPoliciesCommand.ExecuteNonQuery();

        if (rowsAffected > 0)

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            string insertAutoQuery = "INSERT INTO InsuranceAuto (IdPolicies, AutoNumber, AutoYear,
AutoCost) VALUES (@IdPolicies, @AutoNumber, @AutoYear, @AutoCost)";
            SqlCommand insertAutoCommand = new SqlCommand(insertAutoQuery, connection);
            insertAutoCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
            insertAutoCommand.Parameters.AddWithValue("@AutoNumber", autoNumber);
            insertAutoCommand.Parameters.AddWithValue("@AutoYear", autoYear);
            insertAutoCommand.Parameters.AddWithValue("@AutoCost", autoCost);

            int rowsAffectedAuto = insertAutoCommand.ExecuteNonQuery();

            if (rowsAffectedAuto > 0)
            {
                MessageBox.Show("Поліс та інформація про авто успішно додані.");
                LoadData();
            }
            else
            {
                MessageBox.Show("Не вдалося додати інформацію про авто.");
            }
        }
        else
        {
            MessageBox.Show("Не вдалося додати поліс.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка при додаванні поліса та інформації про авто: " + ex.Message);
    }
    finally
    {
        connection.Close();
    }
}
else
{
    MessageBox.Show("Перевірте правильність введення даних.");
}
}
private bool ValidateInput()
{
    if (string.IsNullOrEmpty(IdPoliciesNewTB.Text) ||
        string.IsNullOrEmpty(SummaPolNewTB.Text) ||
        string.IsNullOrEmpty(IdAgentNewTB.Text) ||
        string.IsNullOrEmpty(IdClientNewTB.Text) ||
        DateStartDP.SelectedDate == null ||
        DateEndDP.SelectedDate == null ||
        string.IsNullOrEmpty(AutoNumberTB.Text) ||
        string.IsNullOrEmpty(AutoYearTB.Text) ||
        string.IsNullOrEmpty(AutoCostTB.Text))
    {
        return false;
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

        if (!int.TryParse(IdPoliciesNewTB.Text, out _) ||
            !decimal.TryParse(SummaPolNewTB.Text, out _) ||
            !int.TryParse(IdAgentNewTB.Text, out _) ||
            !int.TryParse(IdClientNewTB.Text, out _) ||
            !int.TryParse(AutoYearTB.Text, out _) ||
            !decimal.TryParse(AutoCostTB.Text, out _))
        {
            return false;
        }

        return true;
    }

    private void ChangeAutoBTN_Click(object sender, RoutedEventArgs e)
    {
        if (ValidateAutoInput())
        {
            int idPolicies;
            if (!int.TryParse(IdPoliciesNewTB.Text, out idPolicies))
            {
                MessageBox.Show("Введіть коректний Id поліса для зміни.");
                return;
            }
            string autoNumber = AutoNumberTB.Text;
            int autoYear;
            if (!int.TryParse(AutoYearTB.Text, out autoYear) || autoYear < 1970 || autoYear > DateTime.Now.Year)
            {
                MessageBox.Show("Рік виробництва автомобіля має бути після 1970 і не більше поточного року.");
                return;
            }
            decimal autoCost;
            if (!decimal.TryParse(AutoCostTB.Text, out autoCost))
            {
                MessageBox.Show("Введіть коректну вартість автомобіля.");
                return;
            }
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                try
                {
                    connection.Open();
                    string query = "UPDATE InsuranceAuto SET AutoNumber = @AutoNumber, AutoYear = @AutoYear, AutoCost = @AutoCost WHERE IdPolicies = @IdPolicies";

                    SqlCommand cmd = new SqlCommand(query, connection);
                    cmd.Parameters.AddWithValue("@AutoNumber", autoNumber);
                    cmd.Parameters.AddWithValue("@AutoYear", autoYear);
                    cmd.Parameters.AddWithValue("@AutoCost", autoCost);
                    cmd.Parameters.AddWithValue("@IdPolicies", idPolicies);

                    int rowsAffected = cmd.ExecuteNonQuery();

                    if (rowsAffected > 0)
                    {

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		55


```

        MessageBox.Show("Дані автомобіля успішно змінені!");
        LoadData();
    }
    else
    {
        MessageBox.Show("Неможливо змінити дані. Перевірте ID");
    }
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при зміні даних автомобіля: " + ex.Message);
}
finally
{
    connection.Close();
}
}
}
else
{
    MessageBox.Show("Перевірте правильність введення даних.");
}
}
private bool ValidateAutoInput()
{
    if (string.IsNullOrEmpty(IdPoliciesNewTB.Text) ||
        string.IsNullOrEmpty(AutoNumberTB.Text) ||
        string.IsNullOrEmpty(AutoYearTB.Text) ||
        string.IsNullOrEmpty(AutoCostTB.Text))
    {
        return false;
    }

    if (!int.TryParse(IdPoliciesNewTB.Text, out _) ||
        !int.TryParse(AutoYearTB.Text, out _) ||
        !decimal.TryParse(AutoCostTB.Text, out _))
    {
        return false;
    }

    return true;
}
private void AutoTBL1_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (AutoTBL1.SelectedItem != null && AutoTBL1.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)AutoTBL1.SelectedItem;

        MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього автомобіля?", "Зміна даних", MessageBoxButton.YesNo, MessageBoxImage.Question);

        if (result == MessageBoxResult.Yes)
        {
            IdPoliciesNewTB.Text = row["IdPolicies"].ToString();
            AutoNumberTB.Text = row["AutoNumber"].ToString();
        }
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

```

        AutoYearTB.Text = row["AutoYear"].ToString();
        AutoCostTB.Text = row["AutoCost"].ToString();
    }
}
else
{
    MessageBox.Show("Тут ще немає автомобіля.");
}
}

// Взаємодія з табл InsuranceLife
private void AddPoliciesHumanBTN_Click(object sender, RoutedEventArgs e)
{
    int idPolicies;
    decimal summaPol;
    int idAgent;
    int idClient;
    DateTime startDate;
    DateTime endDate;
    string lifeFiName;
    string lifeSeName;
    string lifeFaName;
    DateTime lifeBithDay;

    if (!int.TryParse(IdPoliciesNewTB.Text, out idPolicies) ||
        !decimal.TryParse(SummaPolNewTB.Text, out summaPol) ||
        !int.TryParse(IdAgentNewTB.Text, out idAgent) ||
        !int.TryParse(IdClientNewTB.Text, out idClient) ||
        !DateTime.TryParse(DateStartDP.Text, out startDate) ||
        !DateTime.TryParse(DateEndDP.Text, out endDate) ||
        !DateTime.TryParse(LifeBithDayDP.Text, out lifeBithDay) ||
        string.IsNullOrEmpty(lifeFiName = LifeFiNameTB.Text.Trim()) ||
        string.IsNullOrEmpty(lifeSeName = LifeSeNameTB.Text.Trim()) ||
        string.IsNullOrEmpty(lifeFaName = LifeFaNameTB.Text.Trim()) ||
        char.IsLower(lifeFiName[0]) || char.IsLower(lifeSeName[0]) || char.IsLower(lifeFaName[0]) ||
        lifeBithDay.Year < 1930 || lifeBithDay > DateTime.Now || summaPol <= 0)
    {
        MessageBox.Show("Перевірте правильність введення даних про поліс та людину.");
        return;
    }

    if (startDate > endDate || endDate < DateTime.Now || startDate < DateTime.Now)
    {
        MessageBox.Show("Дата початку страховки має бути меншою за дату кінця та не може бути минулою.");
        return;
    }

    string status;
    DateTime currentDate = DateTime.Now;

    if (currentDate < startDate)
    {
        status = "Неактивний";
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

```

else if (currentDate >= startDate && currentDate <= endDate)
{
    status = "Активний";
}
else
{
    status = "Прострочений";
}
string connectionString = "YourConnectionString";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    SqlTransaction transaction = null;
    try
    {
        connection.Open();
        transaction = connection.BeginTransaction();

        string checkExistingPolicyQuery = "SELECT COUNT(*) FROM Policies WHERE IdPolicies =
@IdPolicies";
        SqlCommand checkExistingPolicyCommand = new SqlCommand(checkExistingPolicyQuery,
connection, transaction);
        checkExistingPolicyCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
        int existingPolicyCount = (int)checkExistingPolicyCommand.ExecuteScalar();
        if (existingPolicyCount > 0)
        {
            MessageBox.Show("Поліс з таким ID вже існує.");
            return;
        }
        string insertPoliciesQuery = "INSERT INTO Policies (IdPolicies, SummaPol, IdAgent, IdClients,
StartDate, EndDate, PoliciesType, PolicyStatus) " +
"VALUES (@IdPolicies, @SummaPol, @IdAgent, @IdClients, @StartDate,
@EndDate, @PoliciesType, @PolicyStatus)";
        SqlCommand insertPoliciesCommand = new SqlCommand(insertPoliciesQuery, connection,
transaction);
        insertPoliciesCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
        insertPoliciesCommand.Parameters.AddWithValue("@SummaPol", summaPol);
        insertPoliciesCommand.Parameters.AddWithValue("@IdAgent", idAgent);
        insertPoliciesCommand.Parameters.AddWithValue("@IdClients", idClient);
        insertPoliciesCommand.Parameters.AddWithValue("@PoliciesType", "Страхування життя");
        insertPoliciesCommand.Parameters.AddWithValue("@StartDate", startDate);
        insertPoliciesCommand.Parameters.AddWithValue("@EndDate", endDate);
        insertPoliciesCommand.Parameters.AddWithValue("@PolicyStatus", status);
        insertPoliciesCommand.ExecuteNonQuery();

        string insertInsuranceLifeQuery = "INSERT INTO InsuranceLife (IdPolicies, LifeFiName,
LifeSeName, LifeFaName, LifeBirthDay) " +
"VALUES (@IdPolicies, @LifeFiName, @LifeSeName, @LifeFaName,
@LifeBirthDay)";
        SqlCommand insertInsuranceLifeCommand = new SqlCommand(insertInsuranceLifeQuery,
connection, transaction);
        insertInsuranceLifeCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
        insertInsuranceLifeCommand.Parameters.AddWithValue("@LifeFiName", lifeFiName);
        insertInsuranceLifeCommand.Parameters.AddWithValue("@LifeSeName", lifeSeName);
        insertInsuranceLifeCommand.Parameters.AddWithValue("@LifeFaName", lifeFaName);
        insertInsuranceLifeCommand.Parameters.AddWithValue("@LifeBirthDay", lifeBithDay);
        insertInsuranceLifeCommand.ExecuteNonQuery();
    }
    catch { }
}

```

		Лейс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

        transaction.Commit();
        MessageBox.Show("Дані успішно додані.");
        LoadData();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка при додаванні даних про поліс: " + ex.Message);
        transaction?.Rollback();
    }
    finally
    {
        connection.Close();
    }
}
}
private void ChangeHumanBTN_Click(object sender, RoutedEventArgs e)
{
    if (!string.IsNullOrEmpty(IdPoliciesNewTB.Text))
    {
        int idPolicies = int.Parse(IdPoliciesNewTB.Text);
        string lifeFiName = LifeFiNameTB.Text;
        string lifeSeName = LifeSeNameTB.Text;
        string lifeFaName = LifeFaNameTB.Text;
        DateTime LifeBirthDay = LifeBithDayDP.SelectedDate ?? DateTime.Now;

        if (!ValidateHumanInput(lifeFiName, lifeSeName, lifeFaName, LifeBirthDay))
        {
            MessageBox.Show("Перевірте правильність введення даних про людину.");
            return;
        }

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            try
            {
                connection.Open();
                string query = "UPDATE InsuranceLife SET LifeFiName = @LifeFiName, LifeSeName = @LifeSeName, LifeFaName = @LifeFaName, LifeBirthDay = @LifeBirthDay WHERE IdPolicies = @IdPolicies";

                SqlCommand cmd = new SqlCommand(query, connection);
                cmd.Parameters.AddWithValue("@LifeFiName", lifeFiName);
                cmd.Parameters.AddWithValue("@LifeSeName", lifeSeName);
                cmd.Parameters.AddWithValue("@LifeFaName", lifeFaName);
                cmd.Parameters.AddWithValue("@LifeBirthDay", LifeBirthDay);
                cmd.Parameters.AddWithValue("@IdPolicies", idPolicies);

                int rowsAffected = cmd.ExecuteNonQuery();
                if (rowsAffected > 0)
                {
                    MessageBox.Show("Дані про людину та поліс успішно змінені!");
                    LoadData();
                }
                else
                {

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

```

        MessageBox.Show("Неможливо змінити дані. Перевірте ID");
    }
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при зміні даних про людину: " + ex.Message);
}
finally
{
    connection.Close();
}
}
}
else
{
    MessageBox.Show("Введіть Id поліса для зміни.");
}
}
private bool ValidateHumanInput(string firstName, string secondName, string familyName, DateTime
birthday)
{
    if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(secondName) ||
string.IsNullOrEmpty(familyName))
    {
        return false;
    }

    if (char.IsLower(firstName[0]) || char.IsLower(secondName[0]) || char.IsLower(familyName[0]))
    {
        return false;
    }

    if (birthday < new DateTime(1930, 1, 1) || birthday > DateTime.Now)
    {
        return false;
    }

    return true;
}
private void LifeTBL1_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (LifeTBL1.SelectedItem != null && LifeTBL1.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)LifeTBL1.SelectedItem;
        MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього клієнта?", "Зміна даних",
MessageBoxButton.YesNo, MessageBoxImage.Question);

        if (result == MessageBoxResult.Yes)
        {
            IdPoliciesNewTB.Text = row["IdPolicies"].ToString();
            LifeFiNameTB.Text = row["LifeFiName"].ToString();
            LifeSeNameTB.Text = row["LifeSeName"].ToString();
            LifeFaNameTB.Text = row["LifeFaName"].ToString();
            LifeBirthDayDP.SelectedDate = row["LifeBirthDay"] != DBNull.Value
(DateTime)row["LifeBirthDay"] : DateTime.Now;

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}
else
{
    MessageBox.Show("Тут ще немає клієнта.");
}
}

// Взаємодія з табл InsuranceHome
private void AddPoliciesHomeBTN_Click(object sender, RoutedEventArgs e)
{
    int idPolicies;
    string homeAddress;
    decimal homeCost;
    int summaPol;
    int idAgent;
    int idClient;
    DateTime dateStart;
    DateTime dateEnd;

    if (!int.TryParse(IdPoliciesNewTB.Text, out idPolicies) ||
        !int.TryParse(SummaPolNewTB.Text, out summaPol) ||
        !int.TryParse(IdAgentNewTB.Text, out idAgent) ||
        !int.TryParse(IdClientNewTB.Text, out idClient) ||
        !DateTime.TryParse(DateStartDP.Text, out dateStart) ||
        !DateTime.TryParse(DateEndDP.Text, out dateEnd) ||
        string.IsNullOrEmpty(homeAddress = HomeAddressrTB.Text.Trim()) ||
        !decimal.TryParse(HomeCostTB.Text, out homeCost) ||
        !Regex.IsMatch(homeAddress, @"вул\.\s\w+,\s\d+,\s\d+") || summaPol <= 0)
    {
        MessageBox.Show("Перевірте правильність введення даних про страхування житла.");
        return;
    }

    if (dateStart > dateEnd || dateEnd < DateTime.Now || dateStart < DateTime.Now)
    {
        MessageBox.Show("Дата початку страховки має бути меншою за дату кінця та не може бути минулою.");
        return;
    }

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlTransaction transaction = null;

        try
        {
            connection.Open();
            transaction = connection.BeginTransaction();
            string checkIdQuery = "SELECT COUNT(*) FROM Policies WHERE IdPolicies = @IdPolicies";
            SqlCommand checkIdCommand = new SqlCommand(checkIdQuery, connection, transaction);
            checkIdCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);

            int existingRecords = (int)checkIdCommand.ExecuteScalar();
            if (existingRecords > 0)

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    MessageBox.Show("Посилка з вказаним ID вже існує.");
    return;
}
string status;
DateTime currentDate = DateTime.Now;

if (currentDate < dateStart)
{
    status = "Неактивний";
}
else if (currentDate >= dateStart && currentDate <= dateEnd)
{
    status = "Активний";
}
else
{
    status = "Прострочений";
}

string insertPoliciesQuery = "INSERT INTO Policies (IdPolicies, SummaPol, IdAgent, IdClients,
StartDate, EndDate, PoliciesType, PolicyStatus) " +
    "VALUES (@IdPolicies, @SummaPol, @IdAgent, @IdClients, @StartDate,
@EndDate, @PoliciesType, @PolicyStatus)";
SqlCommand insertPoliciesCommand = new SqlCommand(insertPoliciesQuery, connection,
transaction);
insertPoliciesCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
insertPoliciesCommand.Parameters.AddWithValue("@SummaPol", summaPol);
insertPoliciesCommand.Parameters.AddWithValue("@IdAgent", idAgent);
insertPoliciesCommand.Parameters.AddWithValue("@IdClients", idClient);
insertPoliciesCommand.Parameters.AddWithValue("@PoliciesType", "Страхування житла");
insertPoliciesCommand.Parameters.AddWithValue("@StartDate", dateStart);
insertPoliciesCommand.Parameters.AddWithValue("@EndDate", dateEnd);
insertPoliciesCommand.Parameters.AddWithValue("@PolicyStatus", status);
insertPoliciesCommand.ExecuteNonQuery();

string insertInsuranceHomeQuery = "INSERT INTO InsuranceHome (IdPolicies, HomeAddress,
HomeCost) VALUES (@IdPolicies, @HomeAddress, @HomeCost)";
SqlCommand insertInsuranceHomeCommand = new SqlCommand(insertInsuranceHomeQuery,
connection, transaction);
insertInsuranceHomeCommand.Parameters.AddWithValue("@IdPolicies", idPolicies);
insertInsuranceHomeCommand.Parameters.AddWithValue("@HomeAddress", homeAddress);
insertInsuranceHomeCommand.Parameters.AddWithValue("@HomeCost", homeCost);
insertInsuranceHomeCommand.ExecuteNonQuery();

transaction.Commit();
MessageBox.Show("Дані про страхування житла успішно додані.");
LoadData();
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при додаванні даних про страхування житла: " + ex.Message);
    transaction?.Rollback();
}
finally
{
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        connection.Close();
    }
}
}
private void ChangeHomeBTN_Click(object sender, RoutedEventArgs e)
{
    if (!string.IsNullOrEmpty(IdPoliciesNewTB.Text))
    {
        int idPolicies;
        string homeAddress;
        decimal homeCost;

        if (!int.TryParse(IdPoliciesNewTB.Text, out idPolicies) ||
            string.IsNullOrEmpty(homeAddress = HomeAddressrTB.Text.Trim()) ||
            !decimal.TryParse(HomeCostTB.Text, out homeCost) ||
            !Regex.IsMatch(homeAddress, @"^вул\.\s[A-Za-zA-Яа-я]+\.\s\d+\.\s\d+$"))
        {
            MessageBox.Show("Перевірте правильність введення даних про страхування житла.");
            return;
        }

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            try
            {
                connection.Open();
                string query = "UPDATE InsuranceHome SET HomeAddress = @HomeAddress, HomeCost = @HomeCost WHERE IdPolicies = @IdPolicies";

                SqlCommand cmd = new SqlCommand(query, connection);
                cmd.Parameters.AddWithValue("@HomeAddress", homeAddress);
                cmd.Parameters.AddWithValue("@HomeCost", homeCost);
                cmd.Parameters.AddWithValue("@IdPolicies", idPolicies);

                int rowsAffected = cmd.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    MessageBox.Show("Дані про страхування житла успішно змінені!");
                    LoadData();
                }
                else
                {
                    MessageBox.Show("Неможливо змінити дані. Перевірте ID");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка при зміні даних про страхування житла: " + ex.Message);
            }
            finally
            {
                connection.Close();
            }
        }
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		63


```

    }
    else
    {
        MessageBox.Show("Введіть Id поліса для зміни.");
    }
}
private void HomeTBL1_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (HomeTBL1.SelectedItem != null && HomeTBL1.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)HomeTBL1.SelectedItem;
        MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього об'єкта?", "Зміна даних",
        MessageBoxButton.YesNo, MessageBoxImage.Question);
        if (result == MessageBoxResult.Yes)
        {
            IdPoliciesNewTB.Text = row["IdPolicies"].ToString();
            HomeAddressrTB.Text = row["HomeAddress"].ToString();
            HomeCostTB.Text = row["HomeCost"].ToString();
        }
    }
    else
    {
        MessageBox.Show("Тут ще немає квартири.");
    }
}

```

// Взаємодія з табл InsuranceEvent

```

private void AddEventBTN_Click(object sender, RoutedEventArgs e)
{
    int idPoliciesEvent;
    int idEvent;
    decimal eventSumma;
    string eventReason;
    DateTime eventDate;

    if (!int.TryParse(IdPoliciesEventTB.Text, out idPoliciesEvent) ||
        !int.TryParse(IdEventTB.Text, out idEvent) ||
        !decimal.TryParse(EventSummaTB.Text, out eventSumma) ||
        string.IsNullOrEmpty(eventReason = EventReasonTB.Text.Trim()) ||
        !DateTime.TryParse(EventDP.Text, out eventDate) ||
        eventSumma <= 0)
    {
        MessageBox.Show("Перевірте правильність введення даних про страховий випадок.");
        return;
    }

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            string checkPolicyQuery = "SELECT COUNT(*) FROM Policies WHERE IdPolicies = @IdPolicies";
            SqlCommand checkPolicyCommand = new SqlCommand(checkPolicyQuery, connection);

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

```

        checkPolicyCommand.Parameters.AddWithValue("@IdPolicies", idPoliciesEvent);
        int policyCount = (int)checkPolicyCommand.ExecuteScalar();

        if (policyCount == 0)
        {
            MessageBox.Show("Поліс з вказаним ID не існує.");
            return;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка перевірки наявності поліса: " + ex.Message);
        return;
    }

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();

            string checkEventQuery = "SELECT COUNT(*) FROM InsuranceEvent WHERE IdPolicies = @IdPolicies AND IdEvent = @IdEvent";
            SqlCommand checkEventCommand = new SqlCommand(checkEventQuery, connection);
            checkEventCommand.Parameters.AddWithValue("@IdPolicies", idPoliciesEvent);
            checkEventCommand.Parameters.AddWithValue("@IdEvent", idEvent);

            int eventCount = (int)checkEventCommand.ExecuteScalar();

            if (eventCount > 0)
            {
                MessageBox.Show("Страховий випадок з вказаним ID вже існує.");
                return;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка перевірки наявності страхового випадку: " + ex.Message);
        return;
    }

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            string insertEventQuery = "INSERT INTO InsuranceEvent (IdPolicies, IdEvent, EventDate, EventSumma, EventReason) " +
                "VALUES (@IdPolicies, @IdEvent, @EventDate, @EventSumma, @EventReason)";
            SqlCommand insertEventCommand = new SqlCommand(insertEventQuery, connection);
            insertEventCommand.Parameters.AddWithValue("@IdPolicies", idPoliciesEvent);
            insertEventCommand.Parameters.AddWithValue("@IdEvent", idEvent);
            insertEventCommand.Parameters.AddWithValue("@EventDate", eventDate);

```

		Левс В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

```

insertEventCommand.Parameters.AddWithValue("@EventSumma", eventSumma);
insertEventCommand.Parameters.AddWithValue("@EventReason", eventReason);

int rowsAffected = insertEventCommand.ExecuteNonQuery();
if (rowsAffected > 0)
{
    MessageBox.Show("Страховий випадок успішно доданий.");
    LoadData();
}
else
{
    MessageBox.Show("Не вдалося додати страховий випадок.");
}
}
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при додаванні страхового випадку: " + ex.Message);
}
}
private void ChangeEventBTN_Click(object sender, RoutedEventArgs e)
{
    if (!string.IsNullOrEmpty(IdPoliciesEventTB.Text) && !string.IsNullOrEmpty(IdEventTB.Text)
        && !string.IsNullOrEmpty(EventSummaTB.Text) && !string.IsNullOrEmpty(EventReasonTB.Text)
        && EventDP.SelectedDate != null)
    {
        int idPolicies, idEvent;
        decimal eventSumma;
        DateTime eventDate;
        if (!int.TryParse(IdPoliciesEventTB.Text, out idPolicies) || !int.TryParse(IdEventTB.Text, out idEvent)
            || !decimal.TryParse(EventSummaTB.Text, out eventSumma) || !DateTime.TryParse(EventDP.Text, out
eventDate))
        {
            MessageBox.Show("Невірний формат введених даних.");
            return;
        }
        if (eventSumma <= 0)
        {
            MessageBox.Show("Сума випадка повина бути додатковим числом.");
            return;
        }
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            try
            {
                connection.Open();
                string checkPolicyQuery = "SELECT COUNT(*) FROM Policies WHERE IdPolicies = 
@IdPolicies";
                SqlCommand checkPolicyCmd = new SqlCommand(checkPolicyQuery, connection);
                checkPolicyCmd.Parameters.AddWithValue("@IdPolicies", idPolicies);

                int policyCount = (int)checkPolicyCmd.ExecuteScalar();

                if (policyCount == 0)
                {

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

```

        MessageBox.Show("Поліс з таким ID не існує.");
        return;
    }
    string checkEventQuery = "SELECT COUNT(*) FROM InsuranceEvent WHERE IdPolicies =
@IdPolicies AND IdEvent = @IdEvent";
    SqlCommand checkEventCmd = new SqlCommand(checkEventQuery, connection);
    checkEventCmd.Parameters.AddWithValue("@IdPolicies", idPolicies);
    checkEventCmd.Parameters.AddWithValue("@IdEvent", idEvent);

    int eventCount = (int)checkEventCmd.ExecuteScalar();
    if (eventCount > 0)
    {
        string updateEventQuery = "UPDATE InsuranceEvent SET EventDate = @EventDate,
EventSumma = @EventSumma, EventReason = @EventReason WHERE IdPolicies = @IdPolicies AND IdEvent =
@IdEvent";

        SqlCommand updateEventCmd = new SqlCommand(updateEventQuery, connection);
        updateEventCmd.Parameters.AddWithValue("@IdPolicies", idPolicies);
        updateEventCmd.Parameters.AddWithValue("@IdEvent", idEvent);
        updateEventCmd.Parameters.AddWithValue("@EventDate", eventDate);
        updateEventCmd.Parameters.AddWithValue("@EventSumma", eventSumma);
        updateEventCmd.Parameters.AddWithValue("@EventReason", EventReasonTB.Text);

        int rowsAffected = updateEventCmd.ExecuteNonQuery();
        if (rowsAffected > 0)
        {
            MessageBox.Show("Зміни страхового випадку успішно внесено.");
            LoadData();
        }
        else
        {
            MessageBox.Show("Не вдалося змінити страховий випадок.");
        }
    }
    else
    {
        MessageBox.Show("Страховий випадок з таким ID не існує.");
    }
}
catch (Exception ex)
{
    MessageBox.Show("Помилка при зміні страхового випадку: " + ex.Message);
}
finally
{
    connection.Close();
}
}
else
{
    MessageBox.Show("Заповніть усі поля для зміни страхового випадку.");
}
}
private void DeleteEventBTN_Click(object sender, RoutedEventArgs e)
{

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

```

if (!string.IsNullOrEmpty(IdEventTB.Text) && int.TryParse(IdEventTB.Text, out int idEvent))
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            connection.Open();
            SqlCommand checkEventCommand = new SqlCommand("SELECT IdEvent FROM InsuranceEvent
WHERE IdEvent = @IdEvent;", connection);
            checkEventCommand.Parameters.AddWithValue("@IdEvent", idEvent);
            object eventResult = checkEventCommand.ExecuteScalar();

            if (eventResult != null)
            {
                MessageBoxResult result = MessageBox.Show("Ви впевнені, що хочете видалити цей
страховий випадок?", "Підтвердження видалення", MessageBoxButton.YesNo, MessageBoxImage.Question);
                if (result == MessageBoxResult.Yes)
                {
                    SqlCommand deleteEventCommand = new SqlCommand("DELETE FROM InsuranceEvent
WHERE IdEvent = @IdEvent;", connection);
                    deleteEventCommand.Parameters.AddWithValue("@IdEvent", idEvent);
                    deleteEventCommand.ExecuteNonQuery();

                    MessageBox.Show("Страховий випадок успішно видалено.");
                }
            }
            else
            {
                MessageBox.Show("Страховий випадок з таким ID не існує.");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка видалення страхового випадку: " + ex.Message);
        }
        finally
        {
            connection.Close();
        }
    }
}
else
{
    MessageBox.Show("Введіть Id страхового випадку для видалення.");
}
}

private void InsuranceEventTBL_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (InsuranceEventTBL.SelectedItem != null && InsuranceEventTBL.SelectedItem is DataRowView)
    {
        DataRowView row = (DataRowView)InsuranceEventTBL.SelectedItem;
        if (row["IdPolicies"] != DBNull.Value && row["IdEvent"] != DBNull.Value &&
            row["EventSumma"] != DBNull.Value && row["EventDate"] != DBNull.Value &&
            row["EventReason"] != DBNull.Value)
        {

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        MessageBoxResult result = MessageBox.Show("Хочете змінити дані цього страхового випадку?",
"Зміна даних", MessageBoxButton.YesNo, MessageBoxImage.Question);
        if (result == MessageBoxResult.Yes)
        {
            IdPoliciesEventTB.Text = row["IdPolicies"].ToString();
            IdEventTB.Text = row["IdEvent"].ToString();
            EventSummaTB.Text = row["EventSumma"].ToString();
            EventReasonTB.Text = row["EventReason"].ToString();
            EventDP.SelectedDate = (DateTime)row["EventDate"];
        }
    }
else
{
    MessageBox.Show("Тут ще немає даних про страховий випадок.");
}
}
}
}
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		69