МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

ІНСТИТУТ ФІЗИКО-ТЕХНІЧНИХ ТА КОМПЮТЕРНИХ НАУК

КАФЕДРА КОМПЮТЕРНИХ НАУК

**ЗВІТ**

**з обчислювальної практики**

**на тему:**

**«Розробка застосунку для страхової компанії»**

Керівник практики від навчального закладу,

доцент кафедри комп’ютерних наук к.ф.-м.н. Довгунь А.Я.

Практикант

студент 144 групи

Приутеса Артем Едуардович

**Чернівці - 2025**

**Зміст**

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Опис предметної області

1.2. Технічне завдання на розробку програмного продукту

1.3. Опис інструментів розробки

2. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Проектування бази даних

2.2. Створення прототипу інтерфейсу

2.3. Реалізація програмного продукту

2.4. Інструкція користувача

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. **Аналіз предметної області**
   1. **Опис предметної області**

Страхова компанія — це фінансова установа, основною діяльністю якої є надання страхових послуг фізичним та юридичним особам. Предметна область охоплює сукупність процесів, пов’язаних із прийняттям страхових ризиків, укладанням договорів, обліком страхових випадків, нарахуванням та здійсненням страхових виплат.

Основна мета страхової компанії полягає в забезпеченні фінансового захисту клієнтів від можливих збитків, пов’язаних із ризиками, які можуть виникнути у повсякденному житті або внаслідок господарської діяльності.

Діяльність страхової компанії також передбачає взаємодію з агентами, які здійснюють пошук клієнтів та продаж страхових продуктів, а також з державними регуляторами, які контролюють дотримання чинного законодавства у сфері страхування.

Основний напрямок діяльності є надання страхових послуг фізичним та юридичним особам шляхом укладання договору з метою захисту особи від ризику. Компанія бере на себе зобов’язання виплатити страхове відшкодування або страхову суму при настанні страхового випадку, а клієнт зобов’язується сплачувати страхові внески відповідно до умов договору.

Основний напрям включає в себе: розробку та реалізації страховий продуктів, оцінку страхових ризиків, укладання та обслуговування страхових договорів, ведення обліку клієнтів та полісів, прийом заяв про страхові випадки, прийняття рішень щодо виплат страхового відшкодування.

Додаткові напрями діяльності включають в себе: інвестиційну діяльність, перестрахування, інформаційно-консультаційні послуги, агентська діяльність, освітня та соціальна діяльність.

**Організаційна структура компанії**

**Рис. 1.1**

**Генеральний директор:** несе відповідальність за загальне стратегічне управління підприємством, приймає ключові управлінські рішення, затверджує плани розвитку, контролює фінансові показники та представляє компанію у взаємодії з зовнішніми організаціями — державними органами, партнерами, інвесторами.

**Заступник директора**: курує окремими напрямками діяльності, зокрема страхування, фінанси, правові питання, інформаційні технології, маркетинг та інші.

**Страховий департамент:** відповідає за розробку страхових продуктів, оцінку ризиків, підготовку договорів та загальну організацію процесу страхування. У його складі можуть діяти окремі сектори за напрямами: страхування життя, майна, здоров’я, автомобілів тощо. Працівники цього департаменту проводять розрахунок страхових тарифів, аналізують ринок страхових послуг, формують умови страхування відповідно до чинного законодавства.

**Юридичний відділ:** забезпечує правовий супровід діяльності компанії. Його працівники перевіряють правильність укладення договорів, проводять юридичну експертизу внутрішніх документів, консультують інших працівників з правових питань, супроводжують судові справи, у яких фігурує компанія, а також слідкують за відповідністю роботи страхової організації чинному законодавству України.

**Відділ по роботі з клієнтами**: Його завданням є консультування потенційних і діючих клієнтів, супровід під час оформлення договорів, надання допомоги при продовженні чи зміні полісів, а також розгляд звернень, скарг і запитів. Робота цього відділу безпосередньо впливає на репутацію компанії та рівень довіри з боку клієнтів.

**Фінансово-економічний відділ:** займається всіма аспектами бухгалтерського та фінансового обліку в компанії. Він контролює рух коштів, веде облік сплачених страхових внесків і здійснених виплат, формує фінансову звітність, складає бюджети та забезпечує своєчасне виконання зобов’язань перед клієнтами, державними органами і партнерами. Крім того, цей підрозділ відповідає за створення та облік страхових резервів, що є критично важливими для стабільності компанії.

**Відділ врегулювання збитків:** виконує розгляд страхових випадків. Саме сюди надходять заяви клієнтів про настання страхових подій. Спеціалісти цього відділу здійснюють перевірку фактів, проводять оцінку збитків (іноді спільно з незалежними експертами), приймають рішення про доцільність та розмір виплат, а також готують документи для виплати страхового відшкодування. У випадку порушення умов договору або недостовірності заявлених даних, відділ готує обґрунтовану відмову у виплаті.

Темою для проходження практики є розробка застосунку для надання послуг страхування компанії «Save».

**1.2. Технічне завдання на розробку програмного продукту**

1. Загальні відомості.

Інформаційна система для надання страхових послуг призначена для автоматизації діяльності страхової компанії. Система покликана полегшити процеси обслуговування клієнтів, укладення договорів, обробки страхових випадків, здійснення виплат та формування звітності.

2. Цілі та призначення.

Застосунок має досягти таких цілей:

* зменшення витрат часу на обробку інформації;
* централізоване зберігання даних у безпечному середовищі з архітектурою «клієнт-сервер»;
* можливість швидкого доступу до даних клієнтів, договорів, платежів, страхових випадків;
* зручне управління користувачами через систему ролей(клієнт, адміністратор);
* розширені можливості формування звітів;
* адміністрування користувачів і бази даних.

3.Характеристика об’єкта автоматизації.

Об’єктом автоматизації є страхова компанія, що надає послуги з добровільного та обов’язкового страхування (життя, майна, авто, здоров’я тощо).

4.Вимоги до застосунку.

Система повинна забезпечувати такі функціональні можливості:

* авторизація користувачів та контроль доступу за ролями ;
* реєстрація клієнтів та зберігання інформації про них;
* укладання та супровід договорів страхування;
* реєстрація страхових випадків і облік виплат;
* ведення платежів за договорами;
* формування звітів;
* безпечне зберігання даних;

5.Перелік робіт зі створення застосунку.

Розглянемо перелік робіт, спрямованих на створення застосунку для страхової компанії.

* розробка технічного завдання;
* проектування логічної моделі бази даних;
* створення прототипу інтерфейсу користувача;
* реалізація програмної логіки застосунку;
* тестування функціоналу на тестових даних;
* підготовка інструкції користувача.

6.Порядок розробки застосунку.

1. На основі технічного завдання виконується проектування бази даних.
2. Далі створюється макет інтерфейсу.
3. Реалізується прикладна логіка (введення/редагування клієнтів, договорів, виплат).
4. Додаються допоміжні модулі — формування звітів, захист доступу, журнал операцій.
5. Здійснюється повне тестування системи.

7.Порядок контролю та приймання застосунку.

Контроль та приймання застосунку виконується комісією у складі представника керівництва страхової компанії, IT-фахівця компанії(або простого користувача) та розробника.

Вимоги до введення застосунку в дію.

Для повноцінного запуску ІС потрібно:

* провести навчання персоналу роботі з ІС;
* забезпечити доступ до локальної бази даних;
* протестувати систему в реальному середовищі з обмеженим обсягом даних;
* за потреби коригування на основі результатів пробного запуску.

9.Вимоги до документування.

Супровідна документація повинна включати:

* опис предметної області (страхова компанія);
* схему структури бази даних;
* пояснення до архітектури застосунку;
* інструкції користувача;

10.Джерела розробки:

* завдання на проходження обчислювальної практики;
* технічне завдання;

11.Для того, щоб впровадити застосунок для класичних ігор у роботу, апаратне забезпечення повинно відповідати таким мінімальним нормам:

* процесор Intel Core i3 (1.4 GHz) або аналог
* Оперативна пам’ять 4 GB
* Жорсткий диск 250 GB

Програмне забезпечення:

* Операційна система: Windows 7 або новіша
* Платформа розробки: С++ Builder
* СУБД: Microsoft Access
  1. **Опис інструментів розробки**

C++ Builder — це середовище швидкої розробки (RAD, Rapid Application Development), що дозволяє створювати настільні застосунки для Windows із використанням мови програмування C++. Розроблений компанією Embarcadero, цей інструмент поєднує у собі можливості візуального проектування інтерфейсу та потужного компілятора C++.

Середовище C++ Builder має наступні переваги:

* Підтримка візуального проектування форм (drag-and-drop інтерфейс).
* Висока швидкість створення прототипів.
* Інтеграція з базами даних, у тому числі Microsoft Access.
* Потужний компілятор C++, що дозволяє створювати швидкі й ефективні програми.
* Можливість створення як 32-бітних, так і 64-бітних програм.
* Вбудовані компоненти для роботи з базами даних: TTable, TQuery, TDataSource, DBGrid тощо.

Середовище C++ Builder має наступні недоліки:

* Обмежена підтримка сучасних бібліотек C++ та сторонніх фреймворків.
* Програми можуть мати більший розмір через статичне зв'язування з бібліотеками.
* Платна ліцензія.
* Менша популярність у порівнянні з Visual Studio або Qt.

Інтерфейс:

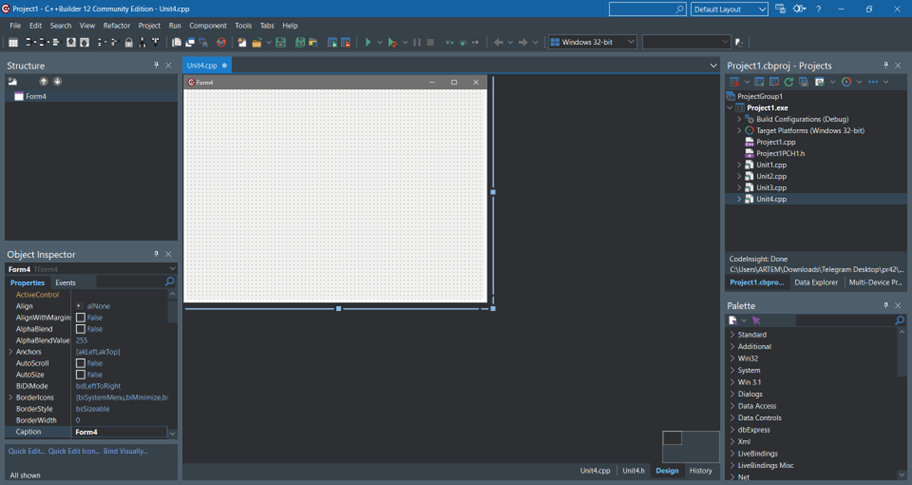


Рис. 1.2

Інтерфейс C++ Builder складається з таких основних частин:

* Form Designer - дозволяє візуально створювати вікна програми.
* Code Editor - для написання програмного коду на C++.
* Object Inspector - для налаштування властивостей об’єктів форми.
* Project Manager - для управління файлами проекту.
* Tool Palette - набір компонентів, які можна перетягувати на форму.

Microsoft Access - це настільна система управління базами даних (СУБД), яка входить до складу пакету Microsoft Office. Вона дозволяє створювати таблиці, запити, форми та звіти без потреби в написанні великої кількості SQL-коду, і добре підходить для невеликих або середніх проєктів.

Середовище Microsoft Access має наступні переваги:

* Проста у використанні навіть для користувачів без глибоких технічних знань.
* Інтеграція з іншими офісними програмами (Excel, Word).
* Швидке створення таблиць, запитів і звітів за допомогою вбудованих шаблонів.
* Можливість імпорту/експорту даних.
* Сумісність із C++ Builder через ODBC або драйвери Access Database Engine.

Середовище Microsoft Access має наступні недоліки:

* Обмежена масштабованість: підходить для невеликої кількості користувачів і записів .
* Відсутність підтримки складних транзакцій та тригерів, як у професійних СУБД (SQL Server, PostgreSQL).
* Відсутність кросплатформенності — працює лише в середовищі Windows.
* Невисока безпека, якщо не реалізувати додаткові механізми захисту.

Інтерфейс:

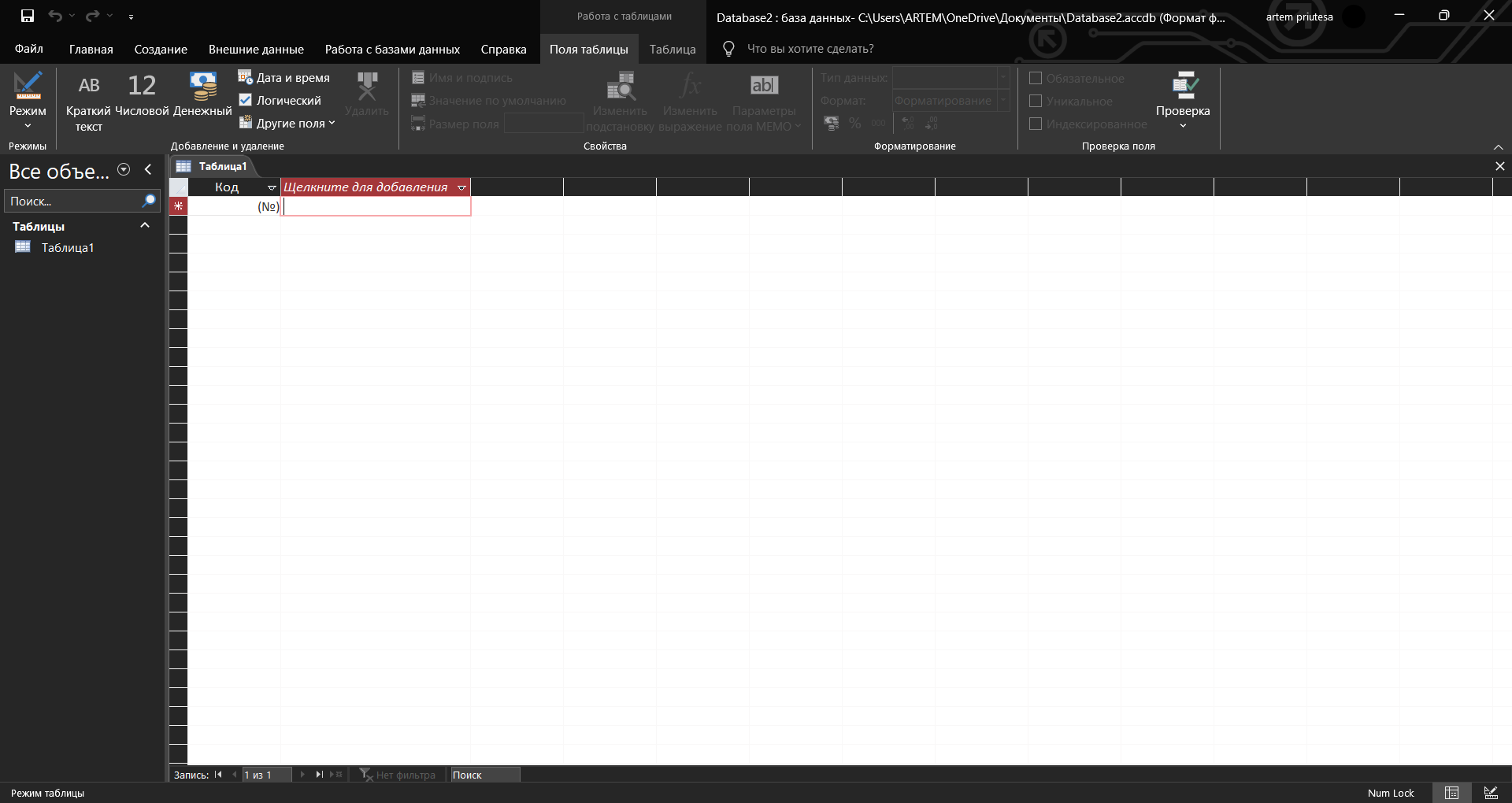
****

Рис. 1.3

Інтерфейс Microsoft Access складається з:

* Таблиць — для зберігання даних.
* Запитів — для вибірки, фільтрації та обчислення даних.
* Форм — для візуального введення/редагування даних.
* Звітів — для генерації вихідної документації.
* Навігаційної панелі — доступ до всіх об’єктів бази.

**2. Розробка програмного продукту**

2.1 Проектування бази даних

Характеристика бази даних

У відповідності до проаналізованої предметної області можна виділити такі сутності:

* Користувач – це людина або компанія яка користується страховими послугами.
* Договір(поліс) - це документ між клієнтом і страховою компанією.
* Платіж – це запис про те, коли і скільки клієнт заплатив за страхування.
* Страховий випадок - фіксація подій, пов’язаних із договором страхування, при яких клієнт подає заяву на відшкодування.
* Страхова виплата – грошова сума яка надається у випадку підтвердження страхового випадку.

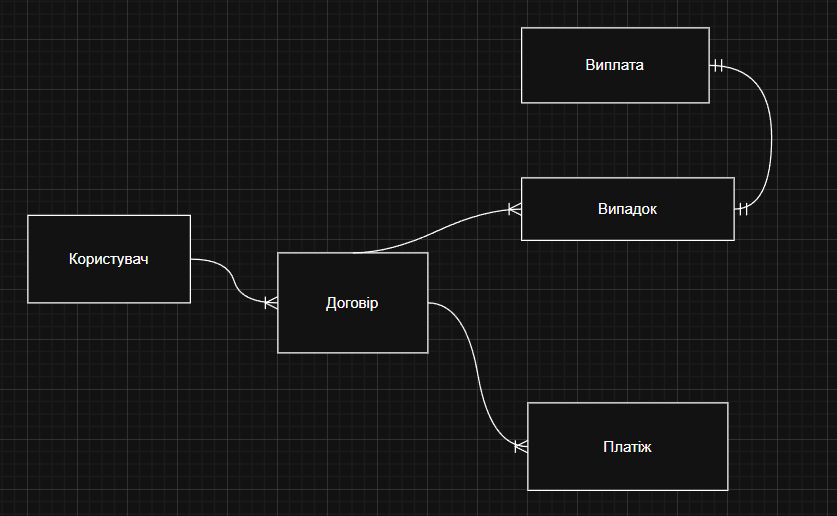


Рис.2.1 Концептуальна модель бази даних

Сутність Користувач використовує наступні реквізити:

* Код користувача(тип даних числовий ,ключове поле)
* Повне ім’я(тип даних текстовий)
* Логін(тип даних текстовий)
* Пароль(тип даних текстовий)
* Роль(тип даних текстовий)

Сутність Договір має наступні реквізити:

* Код договору(тип даних числовий, ключове поле)
* Код користувача(тип даних числовий, зв'язок з Користувач)
* Тип страхування(тип даних текстовий)
* Дата початку(тип даних числовий)
* Дата закінчення(тип даних числовий)
* Страхова сума(тип даних числовий)
* Статус договору(тип даних текстовий)

Сутність Страховий випадок має наступні реквізити:

* Код страхового випадку(тип даних числовий, ключове поле)
* Код договору(тип даних числовий, зв'язок з Договір)
* Дата випадку(тип даних числовий)
* Коментар(тип даних текстовий)
* Статус випадку(тип даних текстовий)

Сутність Виплата має наступні реквізити:

* Код страхового випадку(тип даних числовий, ключове поле, зв'язок Страховим випадком)
* Код виплати(тип даних числовий)
* Дата виплати(тип даних числовий)
* Сума виплати(тип даних числовий)

Сутність Платіж має наступні реквізити:

* Код платежу(тип даних числовий, ключове поле)
* Код договору(тип даних числовий, зв'язок з договором)
* Дата платежу(тип даних числовий)
* Сума платежу(тип даних числовий)

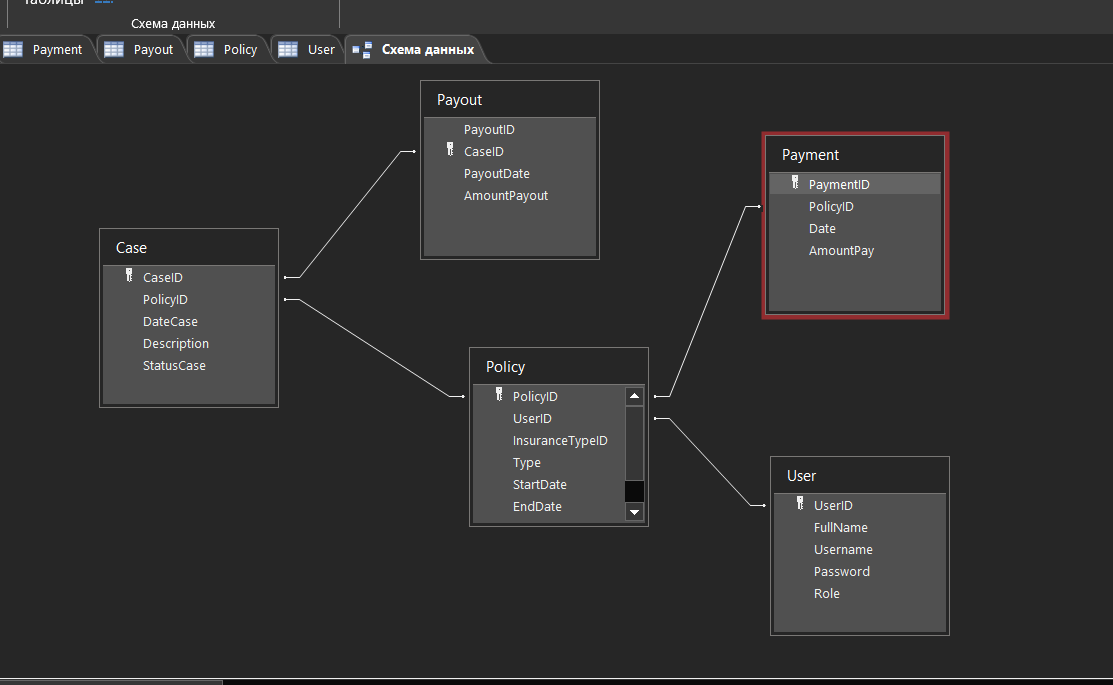


Рис.2.2 Схема бази даних

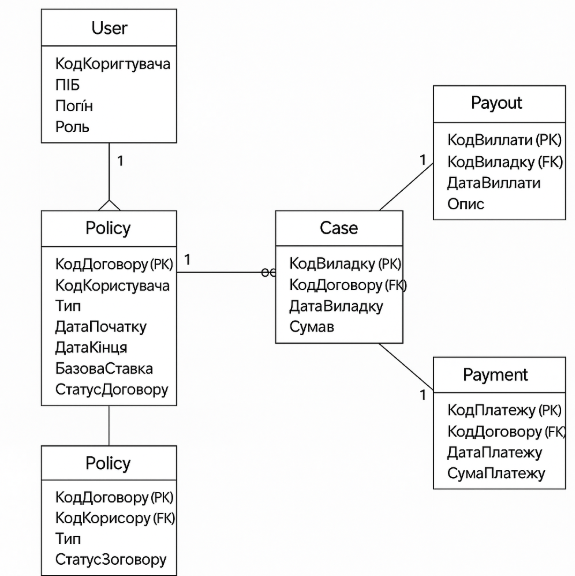


Рис.2.3 Логічна схема

Зв’язки між таблицями:

* Користувач 1—∞ Договір
* Договір 1—∞ Платіж
* Договір 1—∞ Випадок
* Випадок 1—1 Виплата

БД для страхової компанії створювалася в СУБД Accеss. Нижче вказані структури таблиць в режимі конструктора:

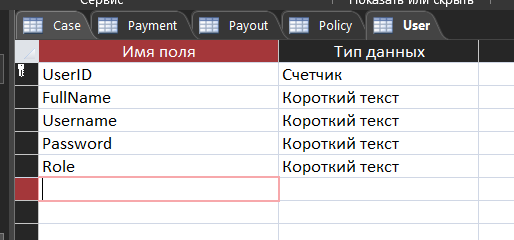


Рис 2.4 Структра таблиці Користувач

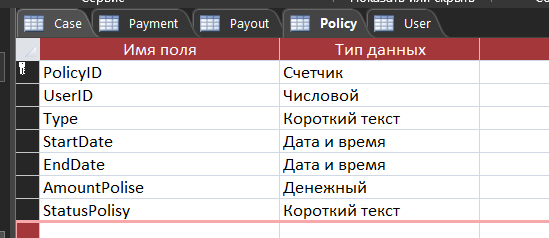


Рис 2.5 Структура таблиці Договір

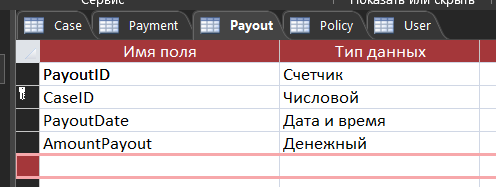


Рис 2.6 Структура таблиці Страхова виплата

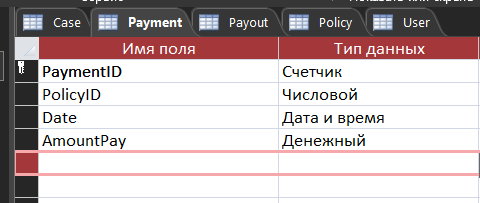


Рис 2.7 Структура таблиці Платіж

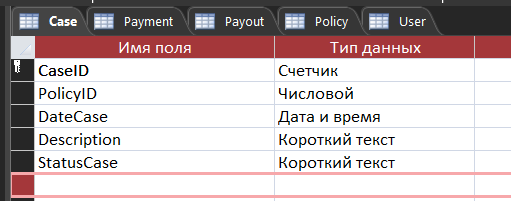


Рис 2.8 Структура таблиці Страховий випадок

Перейдемо до створення прототипу інтерфейсу.

**2.2 Створення прототипу інтерфейсу**

При створенні своєї програми, я використовував такі компоненти C++ Builder’a: Label, Edit, Button, StringGrid, ComboBox, CheckBox, RadioButton, Timer, PopupMenu, MainMenu, Image, ProgressBar.

**Label**: Текстовий напис. Ключові параметри: **Caption** (текст), **Font** (шрифт), **Alignment** (вирівнювання).

**Edit**: Поле для введення або відображення тексту. Ключові параметри: **Text** (зміст), **Font** (шрифт), **ReadOnly** (заборона редагування).

**Button**: Кнопка, що викликає дію. Ключові параметри: **Caption** (текст), **Font** (шрифт), **OnClick** (подія натискання).

**StringGrid**: Таблиця для відображення і редагування даних. Ключові параметри: **ColCount** (кількість стовпців), **RowCount** (кількість рядків), **Cells** (доступ до даних комірок).

**ComboBox**: Випадаючий список. Ключові параметри: **Items** (елементи списку), **Text** (вибраний текст), **ItemIndex** (індекс вибраного елемента).

**CheckBox**: Прапорець для вибору опції (так/ні). Ключові параметри: **Caption** (текст), **Checked** (стан: встановлено/не встановлено).

**RadioButton**: Перемикач для вибору однієї опції з групи. Ключові параметри: **Caption** (текст), **Checked** (стан: вибрано/не вибрано), **GroupIndex** (групування перемикачів).

**Timer**: Компонент для виконання дій через задані інтервали часу. Ключові параметри: **Enabled** (активний/не активний), **Interval** (інтервал в мілісекундах).

**PopupMenu**: Контекстне меню, що з'являється при правому кліку. Ключовий параметр: **Items** (пункти меню).

**MainMenu**: Головне меню програми. Ключовий параметр: **Items** (пункти меню).

**Image**: Компонент для відображення зображень. Ключовий параметр: **Picture** (власне зображення).

ProgressBar: Графічний елемент, що відображає **хід виконання процесу** у вигляді заповненої смуги. Застосовується для візуалізації прогресу завантаження, обробки даних або довготривалих операцій.  
**Ключовий параметр:** Position (поточне значення прогресу).

DateTimePicker: Елемент для **зручного вибору дати і часу** через календар або часовий список. Дозволяє уникнути помилок при ручному введенні дат.  
**Ключовий параметр:** DateTime (обране значення дати та/або часу).

**2.3 Реалізація застсунку**

* 1. Підключення бази даних.

Для підключення до існуючої бази даних я використовував компоненти ADOConnection та ADOQuery для підключення бази даних до компоненту DBGrid я використовував DataSource.

На початку запуску програми компонент ADOConnection підключається до бази даних після успішного підключення компонент ADOQuery можна використовувати для SQL-операцій. Результати запитів (SELECT) через компонент DataSource відображаються у візуальних компонентах, дозволяючи користувачеві переглядати та взаємодіяти з даними.