**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет кібербезпеки, комп’ютерної та програмної інженерії**

Кафедра інженерії програмного забезпечення



*Курсова робота*

*з дисципліни*

*«Бази даних»*

*на тему: «Розробка бази даних для навчального закладу (школа)»*

Виконав: *студент гр. ПІ-327Б*

*Волошин Кіріл Ігорович  
Прийняв/ла: кандидат технічних наук, доцент* *Шибицька Наталія Миколаївна*

Захищено: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2022

**Завдання на курсову роботу**

Завданнями та метою курсової роботи є вивчення принципів побудови та обслуговування реляційних баз даних; формування у студентів навичок практичного застосування мови запитів Structured Query Language (SQL), сучасної повнофункціональної системи керування базами даних (СКБД) MS SQL Server, яка є вільно розповсюдженим програмним забезпеченням із відкритим кодом; знайомство з методами використання баз даних при розробці програмних застосунків.

# Реферат

Обсяг пояснюваної записки – 52 сторінки.

Кількість частин записки – 11 частин.

Кількість ілюстрацій – 20 ілюстрацій.

Кількість додатків – 1 додаток.

Кількість посилань на джерела інформації – 8 посилань.

Метою курсової роботи є вивчення принципів побудови та обслуговування реляційних баз даних та набуття практичних навичок з побудови бази даних, формування запитів до неї та опрацювання взаємодії бази даних та програмної системи.

Методикою вирішення поставленої задачі:

1. Аналіз предметної області
2. Постановка задачі
3. Визначення програмного забезпечення
4. Проєктування системи
5. Реалізація бази даних
6. Розробка системи
7. Написання пояснювальної записки

Результатом курсової роботи є: схема бази даних, база даних, запити до бази даних, програмне забезпечення для використання бази даних в шкільних закладах, пояснювальна записка.

Ключові слова: школа, база даних, таблиця, інформація, сутність, програмне забезпечення, предметна область.

# Зміст

[Завдання на курсову роботу 2](#_Toc119088074)

[Реферат 3](#_Toc119088075)

[Зміст 4](#_Toc119088076)

[Вступ 5](#_Toc119088077)

[1. Сучасний стан досліджуваної проблеми. База даних для школи 5](#_Toc119088078)

[2. Мета курсової роботи 6](#_Toc119088079)

[Аналіз предметної області 7](#_Toc119088080)

[Мета створення та які переваги має система управління даними для школи 7](#_Toc119088081)

[Постановка задачі. 8](#_Toc119088082)

[Проектування бази даних. 11](#_Toc119088083)

[ Концептуальна модель 11](#_Toc119088084)

[ Сутності предметної області 12](#_Toc119088085)

[1. Сутність учитель 12](#_Toc119088086)

[2. Сутність учень 12](#_Toc119088087)

[3. Сутність клас 13](#_Toc119088088)

[4. Сутність кабінет 13](#_Toc119088089)

[5. Сутність предмет 14](#_Toc119088090)

[6. Сутність оцінка 14](#_Toc119088091)

[ SQL лістинг сутностей: 15](#_Toc119088092)

[Опис програмного забезпечення 16](#_Toc119088093)

[Загальний опис програмного продукту 16](#_Toc119088094)

[Опис програми 17](#_Toc119088095)

[Початок роботи в застосунку 17](#_Toc119088096)

[Меню зміни 19](#_Toc119088097)

[Висновок 28](#_Toc119088098)

[Джерела 29](#_Toc119088099)

[Додатки. Лістинг програми: 30](#_Toc119088100)

# **Вступ**

## **Сучасний стан досліджуваної проблеми. База даних для школи**

На сучасному етапі розвитку світового суспільства освіта стає найважливішим елементом соціальної сфери держави, що забезпечує стійкеекономічне зростання, соціальну стабільність, розвиток інститутів громадянського суспільства та національну безпеку держави. Актуальною для України на даному етапі є глибока та всебічна модернізація сфери вищої освіти з виділенням необхідних для цього ресурсів та створенням механізмів їх ефективного впровадження.

Система керування базами даних (СУБД) — це інструмент для організації даних. Більшість навчальних закладів використовують електронні таблиці та програми Excel для керування та організації даних. Однак у міру зростання вони повинні мати спеціалізовані та власні системи керування даними.

Широке коло користувачів представляє базу даних за допомогою природної мови, а також за допомогою інших доступних людині засобів. Таким чином, в базах даних дані розробляються і зберігаються як єдине ціле в інтересах розв’язування всіх задач предметної області.

База даних містить інформацію, необхідну для розв'язування цілого комплексу задач даної установи, підприємства та ін. База даних може поповнюватися новими даними, а раніше введені дані можуть змінюватися або зовсім видалятися. При цьому зміни в базі даних не вимагають внесення змін у прикладні програми.

З розвитком технологій в сучасному світі, школа перетворилася на складну установу з кількома корпусами, різними відділами, кількома секціями та дуже великою кількістю здобувачів освіти.

Наразі перехід шкіл від вже традиційної паперової системи управління даними на цифрову автоматизовану систему управління є критичною потребою часу. Із зростанням відцифрування надзвичайно важливо мати цифрове автоматизоване рішення для керування даними. Це не тільки допоможе керувати даними без втручання людини, але й підключається до інших модулів програмного забезпечення для управління школою, щоб краще організувати різні завдання та функції школи.

## **Мета курсової роботи**

Метою даної курсової роботи є проектування бази даних для навчального закладу, а саме школи, і розроблення неї. У загальному випадку створення кожної програмної системи, включаючи і базу даних, зазвичай проходить складний життєвий цикл. Існує багато методологій, які описують життєвий цикл проєктування та розробки баз даних. У цьому курсовому проєкті буде використано методологію, згідно з якої життєвий цикл складається з наступних етапів:

* Розробка стратегій автоматизації предметної області;
* Проведення системного аналізу предметної області;
* Концептуальне моделювання предметної області;
* Логічне та фізичне проєктування;

Головною ціллю курсового проєкту є проєктування бази даних навчального закладу, а саме школи. У базі буде зберігатися інформація про учнів, викладачів, класи, предмети. Як вже було сказано вище, ця система може спростити та покращити роботу навчального закладу.

Для виконання даної курсової роботи було обрано:

* Засобом реалізації бази даних було обрано MS SQL Server.
* У якості інтерфейсу було обрано Windows Forms
* Мова програмування – C#
* Середовище розробки – Visual Studio 19

# **Аналіз предметної області**

Сама система управління базою даних школи – це система, яка за допомогою передових технологій, допомагає навчальним закладами зберігати, підтримувати та обробляти данні про здобувачів освіти із автоматизацією повсякденних адміністративних завдань. Це забезпечує співпрацю на основі даних між зацікавленими сторонами школи, такими як вчителі, учні, батьки та адміністративний персонал, а також це являється універсальним рішенням для інтеграції даних про здобувачів освіти із різних процесів.

Данна система спрощує відстеження інформації про студентів і прогресу, що є корисним для персоналу школи. Таким чином, на основі цієї інформації можна приймати важливі рішення, пов’язані з навчанням учнів. Крім того, він також допомагає документувати оцінки студентів і бали оцінювання, відстежувати їхню відвідуваність, створювати звіти тощо. Також така система управління допомагає школам зберігати та отримувати доступ до записів за потреби, таким чином спрощуючи процес роботи адміністрації школи.

## **Мета створення та які переваги має система управління даними для школи**

Як вже було сказано раніше мета створення системи – це спрощення процесу ведення документації в навчальних закладах початкової середньої освіти. Ціль системи – це її стан коли адміністрація школи може зберігати, підтримувати та обробляти данні про здобувачів освіти. Головна перевага системи управління даними школи – це спрощення процесу роботи адміністрації, під цим маємо на увазі:

* Зменшення часу виконання документообігу
* За допомогою бази даних можливо краще та простіше систематизувати структуру даних
* Підвищується продуктивність, без бази даних працівниками доводиться по декілька разів заповнювати одну й ту саму інформацію для різних цілей, але за допомогою системи данні автоматично будуть синхронізуватися в усіх відділах.

Недоліків в системи не так вже і багато, але вони існують та ці недоліки як правило пов’язані з її функціями. Напевно най головніше – це складний інтерфейс, він може призвести до деяких труднощів роботи з базою даних, через що всі вище сказані переваги можуть просто зникнути. Наступна та дуже вагома проблема – це відсутність хорошого інтернету, хоча це і не є на пряму недоліком системи та все одно воно являється прямою необхідністю оптимального функціонування. Також одною з проблем може стати неохота деяких викладачів переходити до використання системи так як їм зручніше використовувати традиційні методи роботи. Хоча це не є прямою проблемою самої системи однак залишається фактор того що така проблема існує, але зручне програмне забезпечення може зробити цей перехід значно зручнішим для них.

# **Постановка задачі.**

Відповідно до проведеного аналізу предметної області, який був наведений у вступі, було сформовано перелік завдань, які необхідно вирішити, щоб реалізувати систему управління базою даних школи.

* Які дані потрібно зберігати для оптимальної роботи
* Визначити функції які система автоматизує
* Визначити користувачів системи

Інформація яку потрібно зберігати:

* Інформація про учнів
* Інформація про викладачів
* Інформація про класи
* Інформація про предмети
* Інформація про оцінки
* Класи
* Розклад

Функції системи:

* Додавання інформації про учнів
* Додавання інфомрації про викладачів
* Додавати класи
* Перегляд предметів
* Перегляд списку учнів
* Перегляд списку класів
* Перегляд списку викладачів
* Редагувати інформацію про учнів
* Редагувати інформацію про викладачів
* Редагувати інформацію про класи
* Видаляти інформацію про учнів
* Видаляти інформацію про викладачів
* Видаляти класи

Запити на які повинна давати відповіді система:

* Які оцінки в учня
* Який предмет відповідає якому викладачу
* Який розклад в потрібний день на неділі
* Скільки учнів в класі

Звіти, які має формувати система:

* Звіт про успішність учнів (яка оцінка в якого учня)
* Звіт про розклад учнів
* Звіт класи учнів
* Звіт про список викладачів
* Звіт про список предметів

Завдання автоматизації:

* Система повинна автоматизувати документообіг. Для цього викладачі повинні вводити інформацію про учнів в систему. Далі система сама формує списки документів та дає можливість редагувати та переглядати її тім в кого є до цього доступ

Користувачі системи:

Користувачами системи є учні, викладачі, адміністрація школи та батьки учнів. Є декілька рівнів доступу переглядач, редактор та адміністратор.

Рівень переглядача є в учнів та їх батьків. З цим рівнем доступу вони можуть переглядати список вчителів, список предметів, список занять та список учнів.

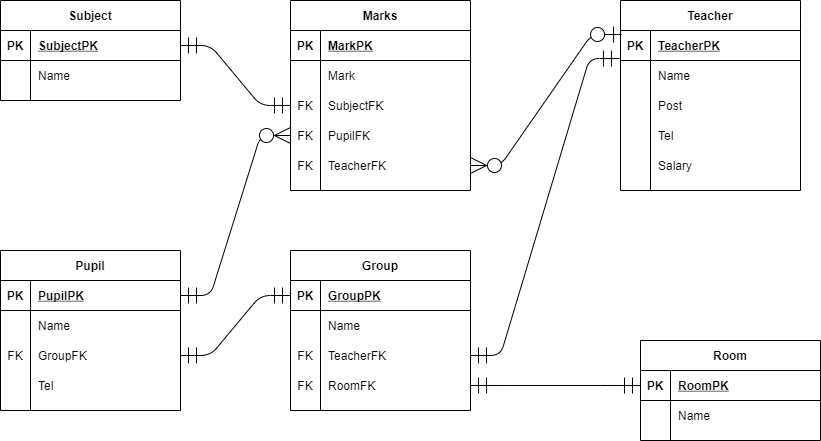
Рівень редактору належить викладачам в якому вони можуть додавати та змінювати інформацію про успішність учнів, а також мають усі права рівня переглядача

Рівень адміністратора належить адміністрації школи, з ним вони можуть редагувати будь яку інформацію, додавати та видаляти. А також мають всі права нижніх рівнів

# **Проектування бази даних.**

## **Концептуальна модель**

Визначивши сутності та провівши аналіз предметної області була сформована концептуальна модель системи даних навчального закладу за допомогою мови ER-моделювання



Пояснення концептуальної моделі:

SubjectPK – SubjectFK (Subject – Marks): Багато до багатьох, може бути декілька оцінок з декілької предметів предмет

PupilPK – PupilFK (Pupil – Marks): Один до багатьох, у одного учня може бути декілька оцінок

TeacherFK – TeacherPK (Marks – Teacher): Багато до одного, може бути декілька оцінок виставлених одиним викалдачем

GroupFK – GroupPK (Pupil – Group): Один до одної, один учень може бути в одній групі

TeacherFK – TeacherPK (Group – Teacher): Один до одного, у одної групи може бути лише один класний керівник(вчитель)

RoomFK – RoomPK (Group – Room): Один клас може бути закріплений за однім кабінетом

# **Сутності предметної області**

## **Сутність учитель**

Короткий опис сутності: Учитель, людина яка вчить учнів, та може бути класним керівником

Атрибути. Сутність характеризується наступними артибутами:

* Пізвище, ім’я
* Номер
* Заробітня плата
* Предмет
* Дата

Зв’язки:

* Учитель може відповідати лише одному класу

Бізнес правила:

* Заробітня плата не може бути нульовою

## **Сутність учень**

Короткий опис сутності: Учень, здобувач освіти

Атрибути. Сутність характеризується наступними артибутами:

* Пізвище, ім’я
* Номер
* Клас

Зв’язки:

* Учень може відповідати лише одному класу

Бізнес правила:

* Учень повинен бути закріплений за класом

## **Сутність клас**

Короткий опис сутності: Клас набір учнів за якими закріплений класний керівник

Атрибути. Сутність характеризується наступними артибутами:

* Назва
* Викладач
* Кабінет
* Оцінки

Зв’язки:

* Клас може мати лише одного класного керівника
* Клас може відповідати лише одиному кабінету

Бізнес правила:

* Клас не може немати кабінету
* Клас повинен мати унікальну назву

## **Сутність кабінет**

Короткий опис сутності: Учитель, людина яка вчить учнів, та може бути класним керівником

Атрибути. Сутність характеризується наступними артибутами:

* Номер кабінету

Зв’язки:

* Кабінет може бути закріплений лише за одним класом

Бізнес правила:

* Кабінет повинен мати унікальний номер

## **Сутність предмет**

Короткий опис сутності: Предмет якому вчать

Атрибути. Сутність характеризується наступними артибутами:

* Назва
* Вчитель

Зв’язки:

* Предмет повинен бути закріплений за вчителелм

Бізнес правила:

* Предмет повинен мати унікальну назву

## **Сутність оцінка**

Короткий опис сутності: Оцінка отримана учнем за який небудь предмет

Атрибути. Сутність характеризується наступними артибутами:

* Оцінка
* Предмет
* Учень
* Група
* Вчитель

Зв’язки:

* Оцінка повинна бути закріплена та учнем і його групою
* Оцінка повинна бути закріплена за вчителем який її виставив

Бізнес правила:

* Оцінка не повинна бути меньше 0 та більше за 12

## **SQL лістинг сутностей:**

CREATE TABLE [dbo].[Teacher]

(

[TeacherPK] INT NOT NULL PRIMARY KEY,

[Name] VARCHAR(50) NOT NULL,

[Post] VARCHAR(50) NOT NULL,

[Salary] INT NOT NULL,

[Tel] VARCHAR(50) NOT NULL,

)

CREATE TABLE [dbo].[room] (

[RoomPK] INT NOT NULL,

[Name] VARCHAR(45) NOT NULL,

PRIMARY KEY ([RoomPK]));

CREATE TABLE [dbo].[group] (

[GroupPK] INT NOT NULL,

[Name] VARCHAR (45) NOT NULL,

[TeacherFK] INT NOT NULL,

[RoomFK] INT NOT NULL,

PRIMARY KEY CLUSTERED ([GroupPK] ASC),

CONSTRAINT [RoomFK] FOREIGN KEY ([RoomFK]) REFERENCES [dbo].[room] ([RoomPK]),

CONSTRAINT [TeacherFK] FOREIGN KEY ([TeacherFK]) REFERENCES [dbo].[Teacher] ([TeacherPK])

);

CREATE TABLE [dbo].[pupil] (

[PupilPK] INT NOT NULL,

[Name] VARCHAR(50) NOT NULL,

[GroupFK] INT NOT NULL,

[Tel] VARCHAR(50) NOT NULL,

PRIMARY KEY ([PupilPK]),

CONSTRAINT [GroupFK] FOREIGN KEY ([GroupFK]) REFERENCES [dbo].[group] ([GroupPK])

);

CREATE TABLE [dbo].[subject] (

[SubjectPK] INT NOT NULL,

[Name] VARCHAR(45) NOT NULL,

PRIMARY KEY ([SubjectPK]));

CREATE TABLE [dbo].[marks] (

[MarkPK] INT NOT NULL,

[Mark] VARCHAR(45) NOT NULL,

[SubjectFK] INT NOT NULL,

[PupilFK] INT NOT NULL,

[TeacherFK] INT NOT NULL,

PRIMARY KEY ([MarkPK]),

CONSTRAINT [SubjectFK] FOREIGN KEY ([SubjectFK]) REFERENCES [dbo].[subject] ([SubjectPK])

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT [PupilFK] FOREIGN KEY ([PupilFK]) REFERENCES [dbo].[pupil] ([PupilPK])

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT [TeacherFK1] FOREIGN KEY ([TeacherFK]) REFERENCES [dbo].[teacher] ([TeacherPK])

ON DELETE NO ACTION

ON UPDATE NO ACTION);

# **Опис програмного забезпечення**

## **Загальний опис програмного продукту**

Даний програмний продукт був написаний на мові програмування C#. Використовувалося стередовище розроки Visual Studio 19

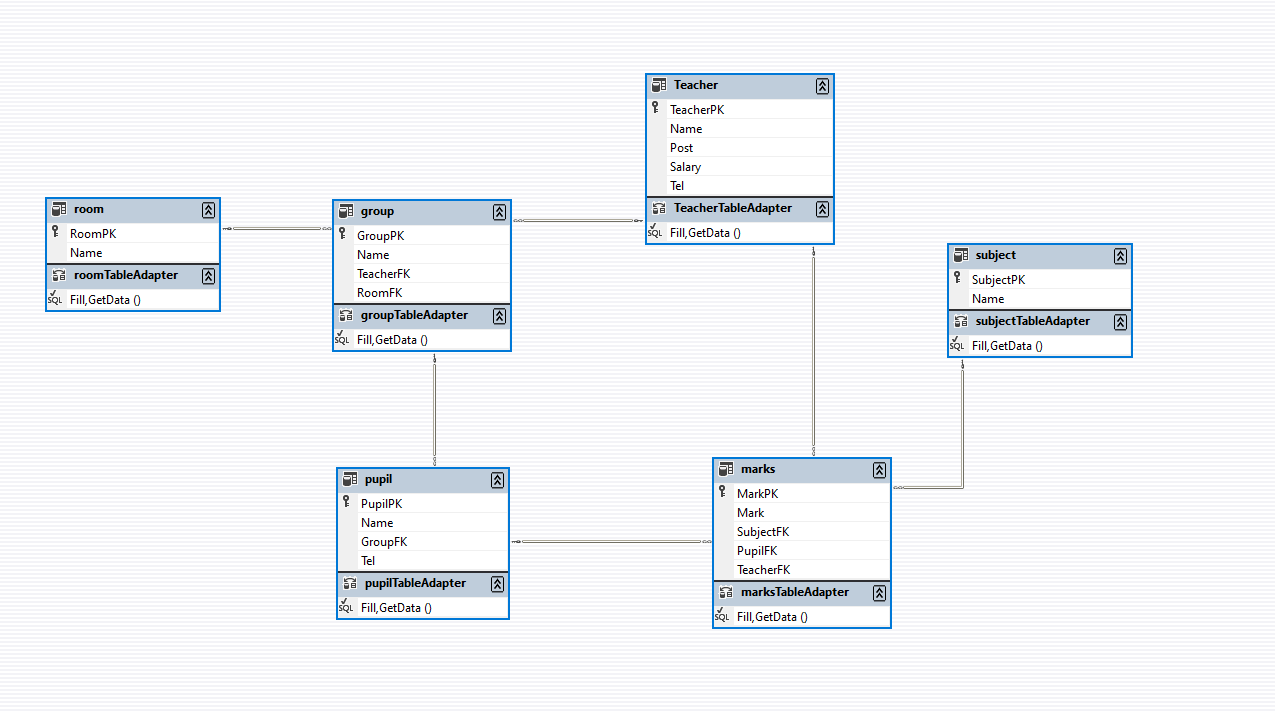
C# одна з найпотужніших мов, що швидко розвиваються і затребуваних в ІТ-галузі. Зараз на ньому пишуться різні програми: від невеликих десктопних програм до великих веб-порталів і веб-сервісів, що обслуговують щодня мільйони користувачів.

Visual Studio — це інтегроване середовище розробки (IDE) від Microsoft. Він використовується для розробки комп'ютерних програм, включаючи веб-сайти , веб-програми, веб-сервіси та мобільні програми. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation, Windows Store і Microsoft Silverlight. Він може створити як рідний, так і керований код .

Visual Studio містить редактор коду, що підтримує IntelliSense ( компонент завершення коду), а також рефакторинг коду. Вбудований налагоджувач працює як налагоджувач на рівні джерела, так і на рівні машини. Інші вбудовані інструменти включають профайлер коду, конструктор для створення програм із графічним інтерфейсом користувача, веб-дизайнер, конструктор класів і конструктор схем бази даних. Він приймає плагіни, які розширюють функціональні можливості майже на кожному рівні, включно з додаванням підтримки систем керування джерелами (наприклад, Subversion і Git ).) і додавання нових наборів інструментів, таких як редактори та візуальні дизайнери для доменних мов або наборів інструментів для інших аспектів життєвого циклу розробки програмного забезпечення (наприклад, клієнт Azure DevOps: Team Explorer).

Microsoft SQL Server - Система управління реляційними базами даних (РСУБД) , розроблена корпорацією Microsoft. Основна мова запитів - Transact-SQL, створений спільно Microsoft і Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO із структурованої мови запитів (SQL) з розширеннями. Використовується до роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД у цьому сегменті ринку.

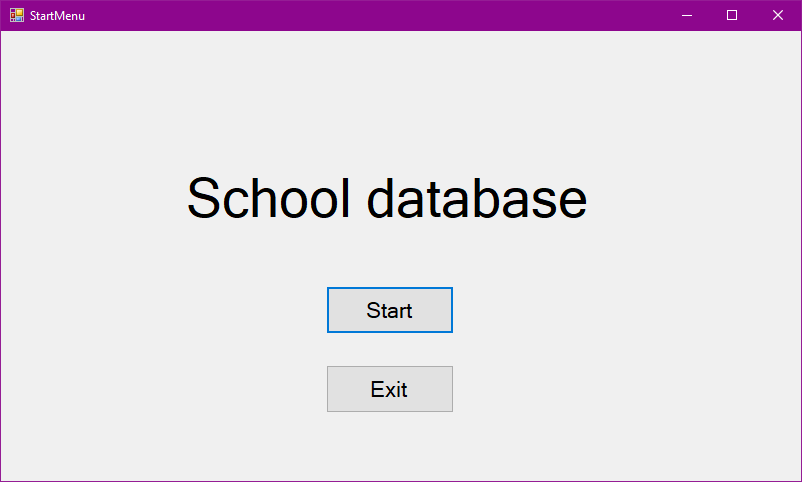
Діаграма класів бази даних:



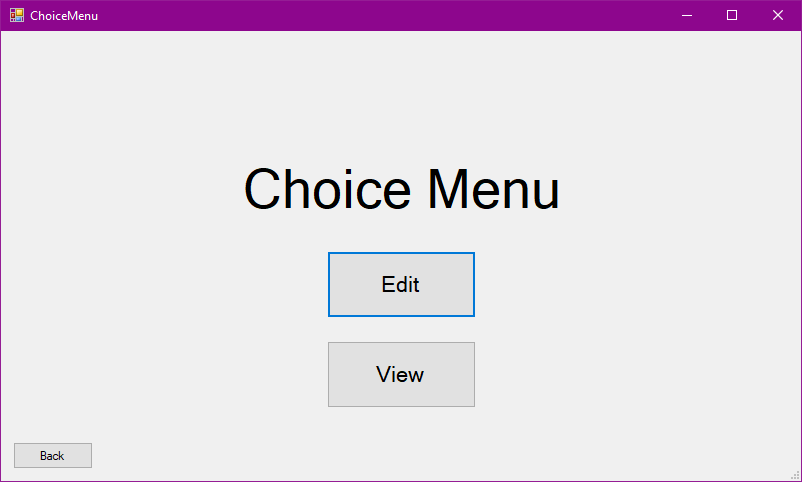
## Опис програми

## Початок роботи в застосунку

Із запуском програми вона буде зустрічати вас з меню з якого можна буде розпочати роботу з застосунком. В якій буде дві кнопки Start – розпочати роботу та Exit – вийти з застосунку.

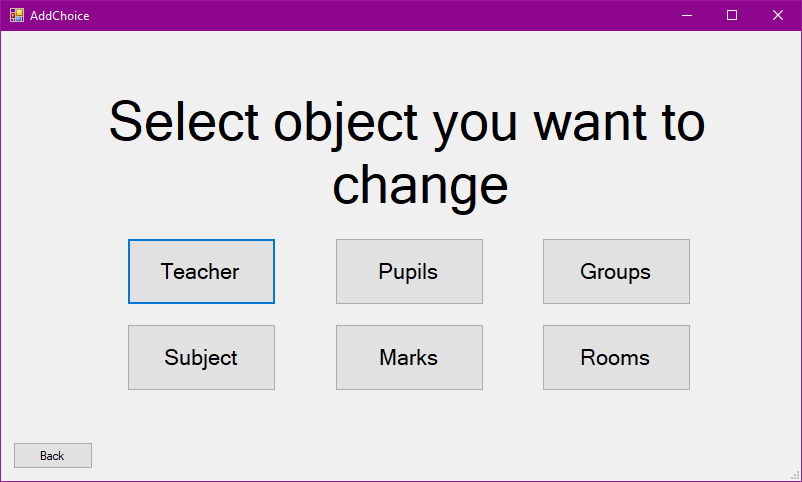


Після початку роботи застосунок перекине вас до головного меню в якому можна буде вибрати функцію яка вам потрібна а саме View – перегляд чи Edit – зміну.

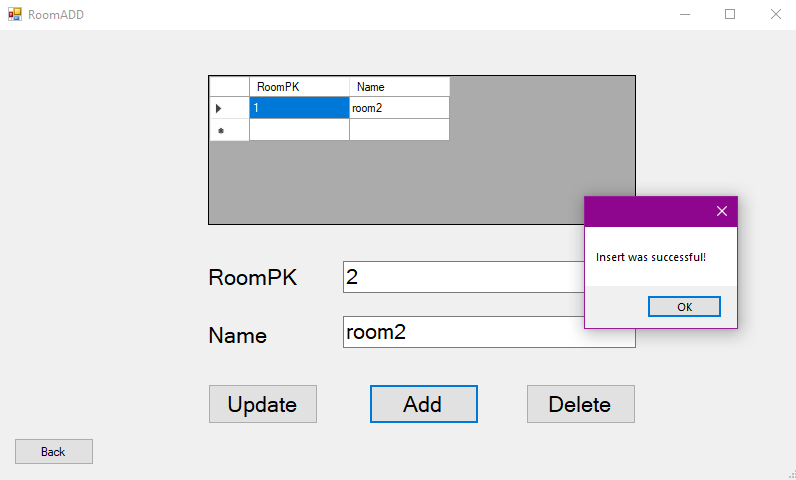


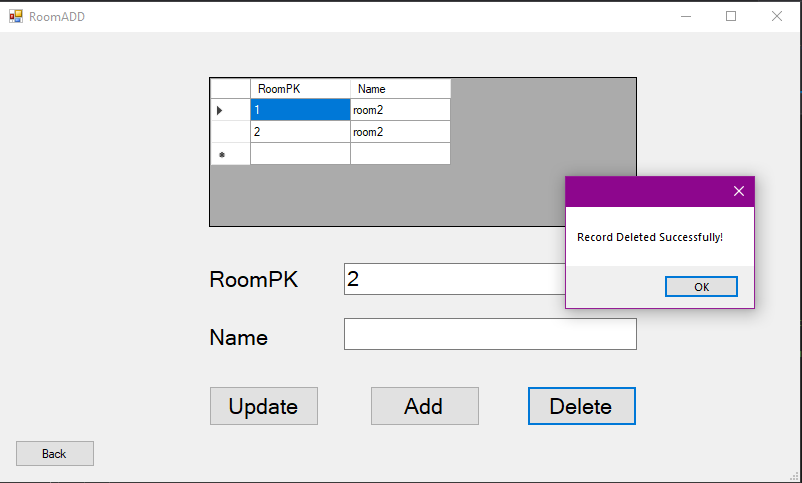
## Меню зміни

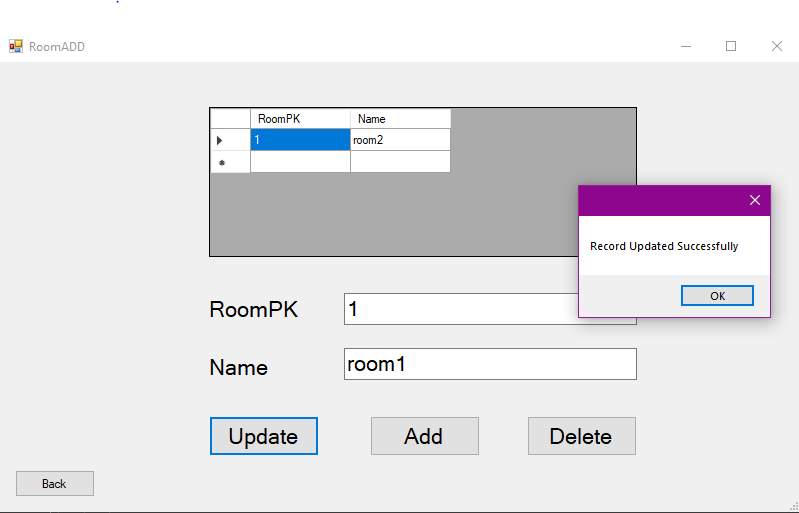
У цьому меню ви маєте можливість вибрати одну із сутностей та додати її оновити або видалити. Кожна дія буде супроводжуватися маленьким віконцем з повідомленням про успішність виконання функції.



Для наглядності я виконаю всі доступні в данному вікні фунції, щоб побачити повідомлення які з’являються

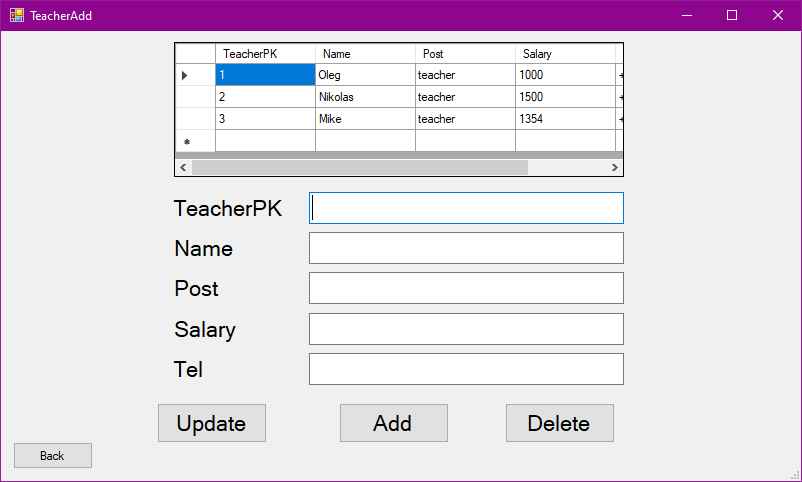


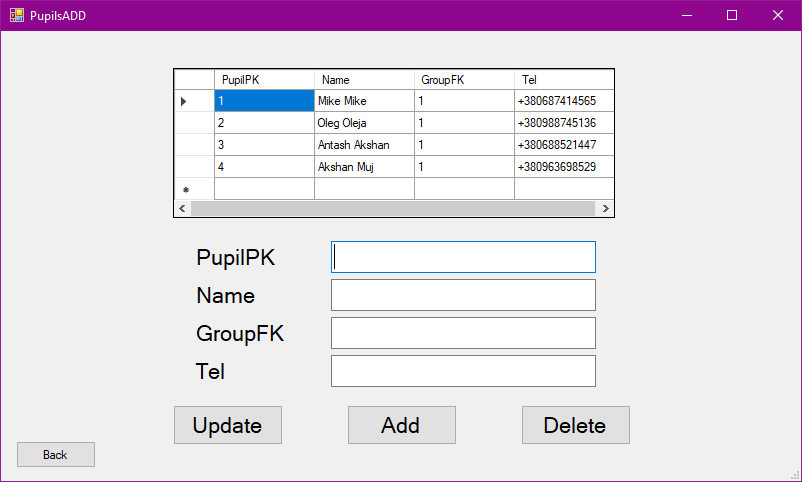
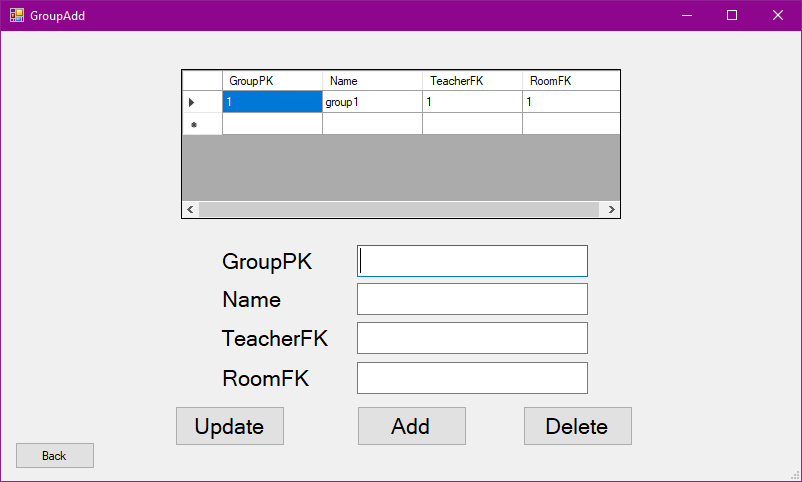


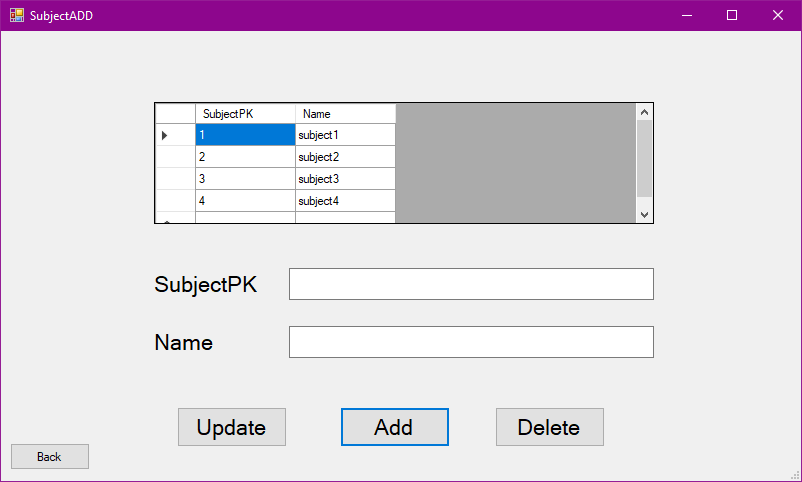


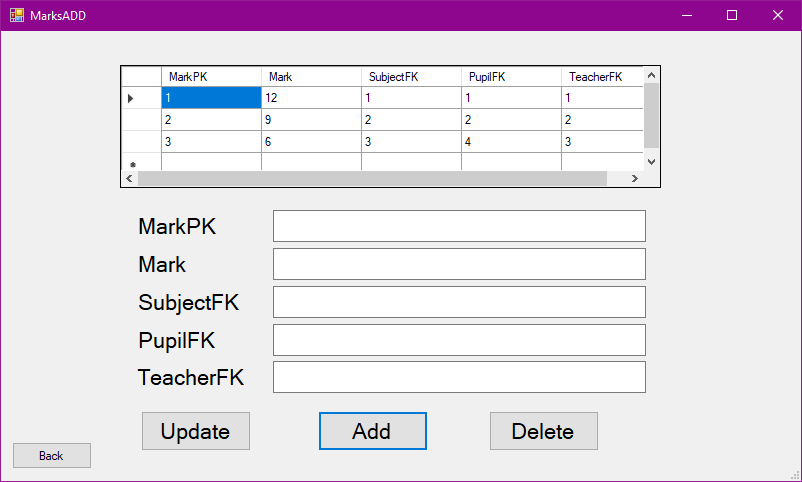
Та за допомогою кнопки Back можна повернутися до минулого меню

Ще продемонструю інші сутності, але без повтору виконання бо все буде виконуватися аналогічно до першого

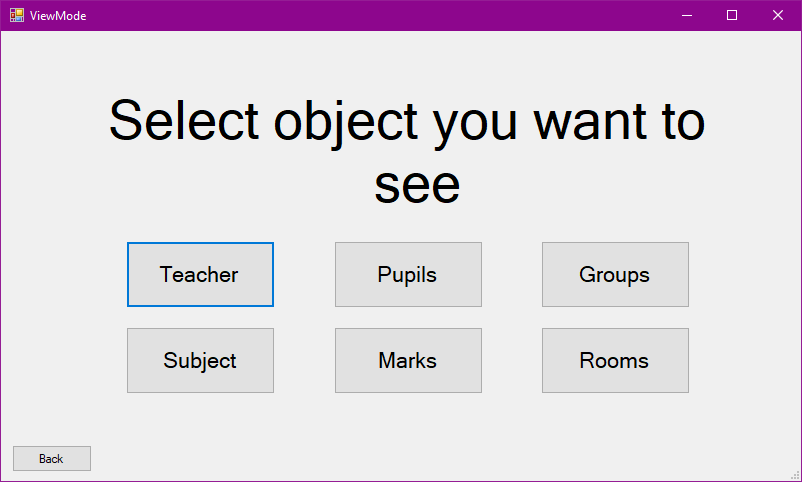


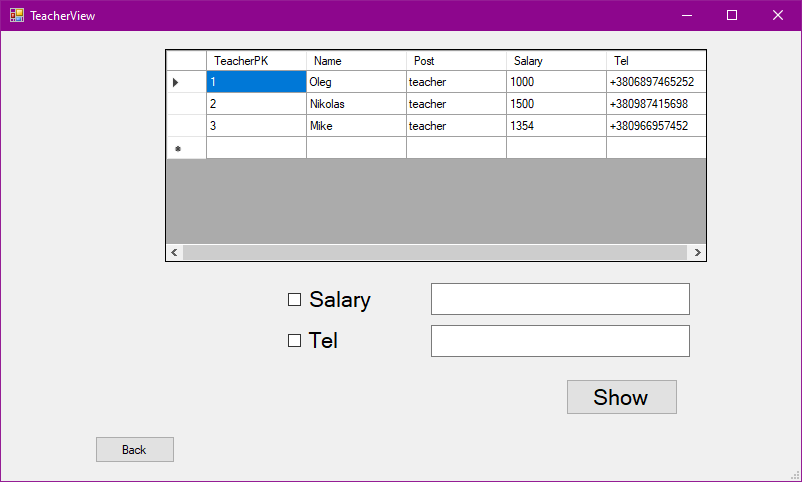
 



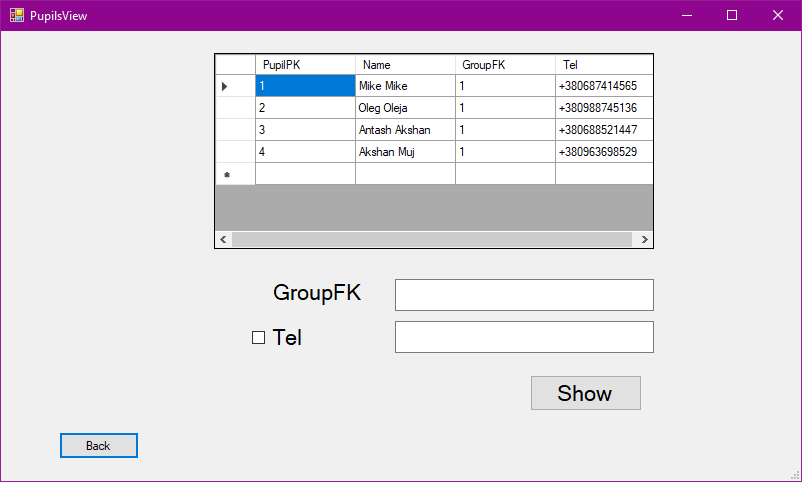


Тепер давайте перейдемо до меню перегляду

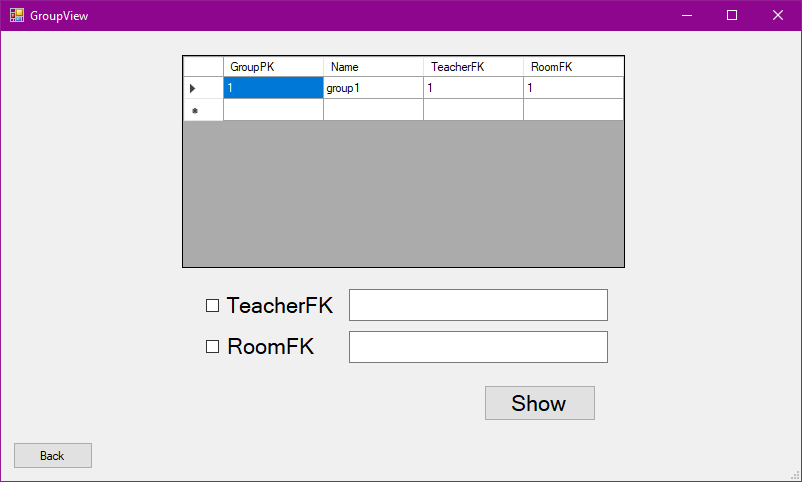




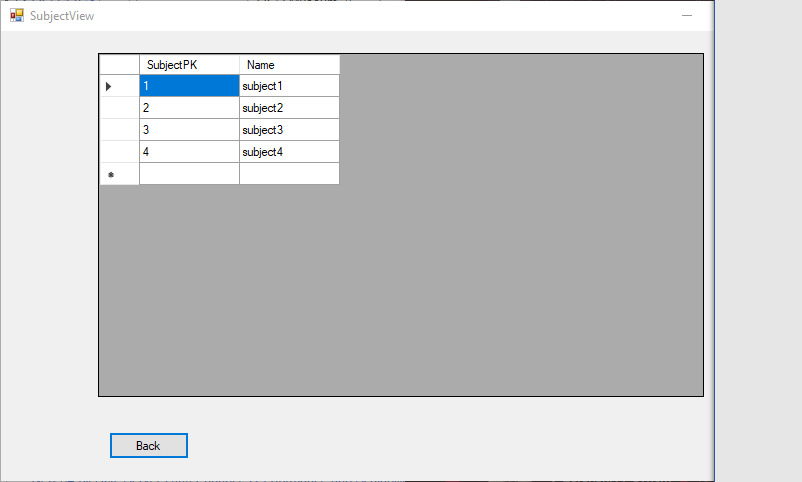
В огляді вчителів можна виставити щоб видавало вчителів без заробітноьої плати або без телефону або ні без того ні без іншого. Також можна задати конкретну заробітню плату або конткретий номер і воно видасть саме те що було вказано



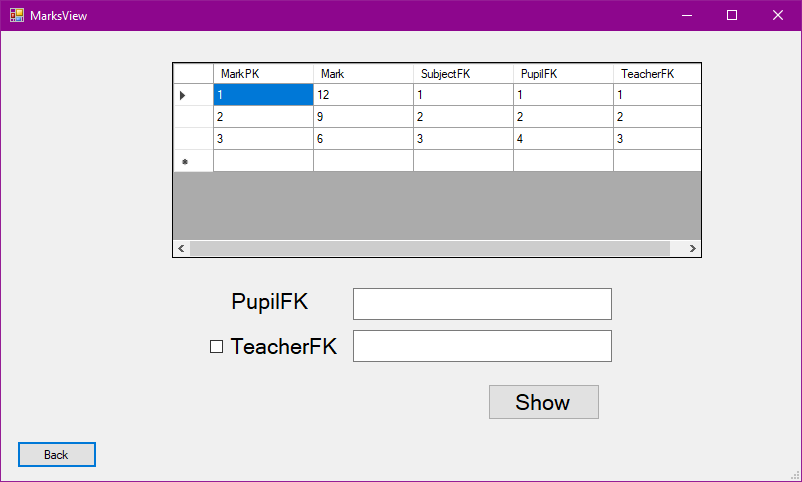
За аналогією до викладачів тут можна вказати видавати з телефоном чи без нього але можна ще вказати конкретну групу, якщо нічого не вказувати воно просто видасть без врахування пошуку



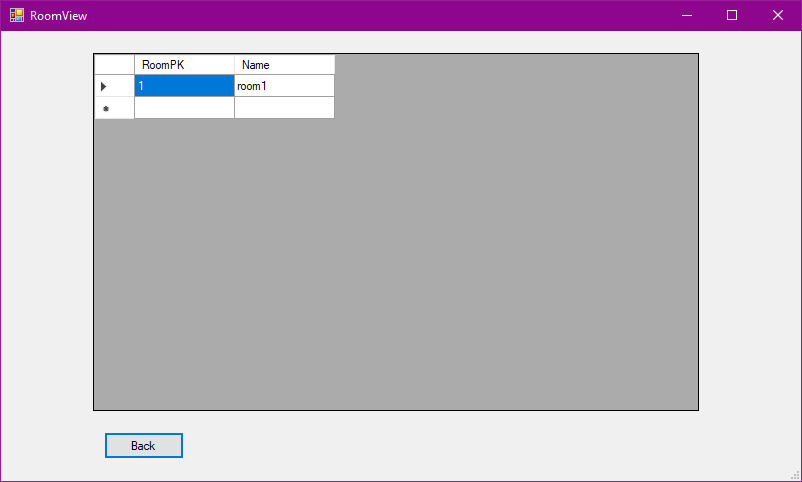
Точно так само працює і перегляд груп але тут параметри це викладач та кабінет



Пердмети одразу показуються без параметрів для сортуваня



За аналогією до інших сутностей в оцінок є параметри для сортування, а саме учень та викладач і так як параметр викладача є не обов’язковим його потрібно вмикати.



Кабінети так само як предмети видаються одразу без параметрів для сортування

# **Висновок**

Протягом виконання курсової роботи було проаналізовано освітню сферу та процес адміністрування нею. Було виведено основні проблеми та завдання. Відповідно до проведеного аналізу предметної області було наведено перелік завдань для того щоб реалізувати систему.

Досліджено які дані потрібно зберігати та які потрібні функції для роботи системи. Була спроєктована та в подальшому розроблена система для контролю бази даних в навчальному закладі, а саме в школі. Основними функціями з сутностями бази даних є:

* Додавання,
* Оновлення
* Видалення
* Перегляд

Вивчено принципи проектування та створення бази даних, методи взаємодії з даними через запити та програмне забезпечення.

Результатом курсової роботи є: схема бази даних, база даних, запити до бази даних, програмне забезпечення для використання бази даних в шкільних закладах, пояснювальна записка.

# Джерела

[https://www.talentedladiesclub.com/](https://www.talentedladiesclub.com/articles/what-are-the-benefits-of-data-management-in-the-education-sector/)

[https://buzzapp.tech/blog/](https://buzzapp.tech/blog/Advantages-and-Disadvantages-of-Student-Management-System#Disadvantages%20of%20Student%20Management%20System)

<https://fedena.com/blog/>

[https://leadschool.in/blog/](https://leadschool.in/blog/ultimate-benefits-of-using-student-database-management-system/)

[https://www.talentedladiesclub.com/](https://www.talentedladiesclub.com/articles/what-are-the-benefits-of-data-management-in-the-education-sector/)

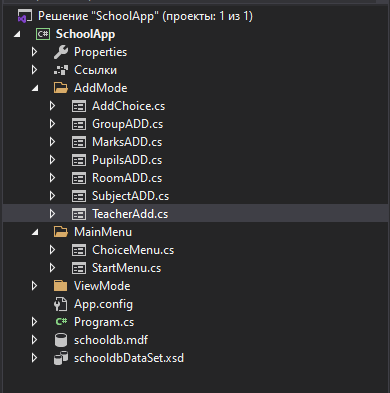
<https://en.wikipedia.org/wiki>

<https://www.quora.com/>

<https://guides.library.newschool.edu/>

# Додатки. Лістинг програми:

Структура програми



ChoiceMenu.cs

using SchoolApp.AddMode;

using SchoolApp.ViewMode;

using System;

using System.Windows.Forms;

namespace SchoolApp.MainMenu

{

public partial class ChoiceMenu : Form

{

public ChoiceMenu()

{

InitializeComponent();

}

private void Edit\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice addChoice = new AddChoice();

addChoice.ShowDialog();

}

private void View\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice viewChoice = new ViewChoice();

viewChoice.ShowDialog();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

StartMenu startMenu = new StartMenu();

startMenu.ShowDialog();

}

}

}

StartMenu.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SchoolApp.MainMenu

{

public partial class StartMenu : Form

{

public StartMenu()

{

InitializeComponent();

}

private void Start\_Click(object sender, EventArgs e)

{

this.Visible = false;

ChoiceMenu choiceMenu = new ChoiceMenu();

choiceMenu.ShowDialog();

}

private void Exit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

AddChoice.cs

using SchoolApp.MainMenu;

using System;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class AddChoice : Form

{

public AddChoice()

{

InitializeComponent();

}

private void Teacher\_Click(object sender, EventArgs e)

{

this.Visible = false;

TeacherAdd choice = new TeacherAdd();

choice.ShowDialog();

}

private void Subject\_Click(object sender, EventArgs e)

{

this.Visible = false;

SubjectADD choice = new SubjectADD();

choice.ShowDialog();

}

private void Pupils\_Click(object sender, EventArgs e)

{

this.Visible = false;

PupilsADD choice = new PupilsADD();

choice.ShowDialog();

}

private void Marks\_Click(object sender, EventArgs e)

{

this.Visible = false;

MarksADD choice = new MarksADD();

choice.ShowDialog();

}

private void Groups\_Click(object sender, EventArgs e)

{

this.Visible = false;

GroupADD choice = new GroupADD();

choice.ShowDialog();

}

private void Rooms\_Click(object sender, EventArgs e)

{

this.Visible = false;

RoomADD choice = new RoomADD();

choice.ShowDialog();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ChoiceMenu choice = new ChoiceMenu();

choice.ShowDialog();

}

private void AddChoice\_Load(object sender, EventArgs e)

{

}

}

}

GroupADD.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class GroupADD : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

private SqlCommand sqlCommand;

private SqlDataAdapter sqlDataAdapter;

public GroupADD()

{

InitializeComponent();

}

private void Add\_Click(object sender, EventArgs e)

{

con.Open();

if (con.State == System.Data.ConnectionState.Open)

{

string q = "INSERT INTO [dbo].[group] ([GroupPK], [Name], [TeacherFK], [RoomFK]) " +

"values(" + Int32.Parse(textBox1.Text) + ",'" + textBox2.Text.ToString() + "'," +

Int32.Parse(textBox3.Text) + "," + Int32.Parse(textBox4.Text) + ") ";

SqlCommand cmd = new SqlCommand(q, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Insert was successful!");

}

con.Close();

Display();

Clear();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice choice = new AddChoice();

choice.ShowDialog();

}

private void GroupADD\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.group". При необходимости она может быть перемещена или удалена.

this.groupTableAdapter.Fill(this.schooldbDataSet.group);

}

private void Update\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "")

{

sqlCommand = new SqlCommand("update group set GroupPK=@GroupPK, Name=@Name, TeacherFK=@TeacherFK, RoomFK=@RoomFK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@GroupPK", textBox1.Text);

sqlCommand.Parameters.AddWithValue("@Name", textBox2.Text);

sqlCommand.Parameters.AddWithValue("@TeacherFK", textBox3.Text);

sqlCommand.Parameters.AddWithValue("@RoomFK", textBox4.Text);

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Updated Successfully");

con.Close();

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Update");

}

}

private void Display()

{

con.Open();

DataTable dt = new DataTable();

sqlDataAdapter = new SqlDataAdapter("select \* from group", con);

sqlDataAdapter.Fill(dt);

dataGridView1.DataSource = dt;

con.Close();

}

private void Clear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

}

private void Delete\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

sqlCommand = new SqlCommand("delete group where GroupPK=@GroupPK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@GroupPK", textBox1.Text);

sqlCommand.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Deleted Successfully!");

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Delete");

}

}

}

}

MarksADD.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class MarksADD : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

private SqlDataAdapter sqlDataAdapter;

public MarksADD()

{

InitializeComponent();

}

private void Add\_Click(object sender, EventArgs e)

{

con.Open();

if (con.State == System.Data.ConnectionState.Open)

{

string q = "insert into marks(MarkPK, Mark, SubjectFK, PupilFK, TeacherFK)" +

"values('" + Int32.Parse(textBox1.Text) + "','" + textBox2.Text.ToString() + "','" +

Int32.Parse(textBox3.Text) + "','" + Int32.Parse(textBox4.Text) + "','" + Int32.Parse(textBox5.Text) + "') ";

SqlCommand cmd = new SqlCommand(q, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Insert was successful!");

}

con.Close();

Display();

Clear();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice choice = new AddChoice();

choice.ShowDialog();

}

private void MarksADD\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.marks". При необходимости она может быть перемещена или удалена.

this.marksTableAdapter.Fill(this.schooldbDataSet.marks);

}

private void Update\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "")

{

sqlCommand = new SqlCommand("update marks set MarkPK=@MarkPK, Mark=@Mark, SubjectFK=@SubjectFK, PupilFK=@PupilFK, TeacherFK=@TeacherFK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@MarkPK", textBox1.Text);

sqlCommand.Parameters.AddWithValue("@Mark", textBox2.Text);

sqlCommand.Parameters.AddWithValue("@SubjectFK", textBox3.Text);

sqlCommand.Parameters.AddWithValue("@PupilFK", textBox4.Text);

sqlCommand.Parameters.AddWithValue("@TeacherFK", textBox5.Text);

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Updated Successfully");

con.Close();

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Update");

}

}

private void Display()

{

con.Open();

DataTable dt = new DataTable();

sqlDataAdapter = new SqlDataAdapter("select \* from marks", con);

sqlDataAdapter.Fill(dt);

dataGridView1.DataSource = dt;

con.Close();

}

private void Clear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

}

private void Delete\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

sqlCommand = new SqlCommand("delete marks where MarkPK=@MarkPK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@MarkPK", textBox1.Text);

sqlCommand.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Deleted Successfully!");

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Delete");

}

}

}

}

PupilsADD.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class PupilsADD : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

private SqlDataAdapter sqlDataAdapter;

public PupilsADD()

{

InitializeComponent();

}

private void Add\_Click(object sender, EventArgs e)

{

con.Open();

if (con.State == System.Data.ConnectionState.Open)

{

string q = "insert into pupil(PupilPK, Name, GroupFK, Tel)" +

"values('" + Int32.Parse(textBox1.Text) + "','" + textBox2.Text.ToString() + "','" +

Int32.Parse(textBox3.Text) + "','" + textBox4.Text.ToString() + "') ";

SqlCommand cmd = new SqlCommand(q, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Insert was successful!");

}

con.Close();

Display();

Clear();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice choice = new AddChoice();

choice.ShowDialog();

}

private void PupilsADD\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.pupil". При необходимости она может быть перемещена или удалена.

this.pupilTableAdapter.Fill(this.schooldbDataSet.pupil);

}

private void Update\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "")

{

sqlCommand = new SqlCommand("update pupil set PupilPK=@PupilPK, Name=@Name, GroupFK=@GroupFK, Tel=@Tel", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@PupilPK", textBox1.Text);

sqlCommand.Parameters.AddWithValue("@Name", textBox2.Text);

sqlCommand.Parameters.AddWithValue("@Post", textBox3.Text);

sqlCommand.Parameters.AddWithValue("@Tel", textBox4.Text);

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Updated Successfully");

con.Close();

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Update");

}

}

private void Display()

{

con.Open();

DataTable dt = new DataTable();

sqlDataAdapter = new SqlDataAdapter("select \* from pupil", con);

sqlDataAdapter.Fill(dt);

dataGridView1.DataSource = dt;

con.Close();

}

private void Clear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

}

private void Delete\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

sqlCommand = new SqlCommand("delete pupil where PupilPK=@PupilPK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@PupilPK", textBox1.Text);

sqlCommand.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Deleted Successfully!");

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Delete");

}

}

}

}

RoomADD.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class RoomADD : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

private SqlDataAdapter sqlDataAdapter;

public RoomADD()

{

InitializeComponent();

}

private void Add\_Click(object sender, EventArgs e)

{

con.Open();

if (con.State == System.Data.ConnectionState.Open)

{

string q = "insert into room(RoomPK, Name)" +

"values('" + Int32.Parse(textBox1.Text) + "','" + textBox2.Text.ToString() + "') ";

SqlCommand cmd = new SqlCommand(q, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Insert was successful!");

}

con.Close();

Display();

Clear();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice choice = new AddChoice();

choice.ShowDialog();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void RoomADD\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.room". При необходимости она может быть перемещена или удалена.

this.roomTableAdapter.Fill(this.schooldbDataSet.room);

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.subject". При необходимости она может быть перемещена или удалена.

this.subjectTableAdapter.Fill(this.schooldbDataSet.subject);

}

private void Update\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "")

{

sqlCommand = new SqlCommand("update room set RoomPK=@RoomPK, Name=@Name", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@RoomPK", textBox1.Text);

sqlCommand.Parameters.AddWithValue("@Name", textBox2.Text);

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Updated Successfully");

con.Close();

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Update");

}

}

private void Display()

{

con.Open();

DataTable dt = new DataTable();

sqlDataAdapter = new SqlDataAdapter("select \* from room", con);

sqlDataAdapter.Fill(dt);

dataGridView1.DataSource = dt;

con.Close();

}

private void Clear()

{

textBox1.Text = "";

textBox2.Text = "";

}

private void Delete\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

sqlCommand = new SqlCommand("delete room where RoomPK=@RoomPK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@RoomPK", textBox1.Text);

sqlCommand.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Deleted Successfully!");

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Delete");

}

}

}

}

SubjectADD.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class SubjectADD : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

private SqlDataAdapter sqlDataAdapter;

public SubjectADD()

{

InitializeComponent();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void Add\_Click(object sender, EventArgs e)

{

con.Open();

if (con.State == System.Data.ConnectionState.Open)

{

string q = "insert into subject(SubjectPK, Name)" +

"values('" + Int32.Parse(textBox1.Text) + "','" + textBox2.Text.ToString() + "') ";

SqlCommand cmd = new SqlCommand(q, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Insert was successful!");

}

con.Close();

Display();

Clear();

}

private void Update\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "")

{

sqlCommand = new SqlCommand("update subject set SubjectPK=@SubjectPK, Name=@Name", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@SubjectPK", textBox1.Text);

sqlCommand.Parameters.AddWithValue("@Name", textBox2.Text);

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Updated Successfully");

con.Close();

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Update");

}

}

private void Display()

{

con.Open();

DataTable dt = new DataTable();

sqlDataAdapter = new SqlDataAdapter("select \* from subject", con);

sqlDataAdapter.Fill(dt);

dataGridView1.DataSource = dt;

con.Close();

}

private void Clear()

{

textBox1.Text = "";

textBox2.Text = "";

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice choice = new AddChoice();

choice.ShowDialog();

}

private void SubjectADD\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.subject". При необходимости она может быть перемещена или удалена.

this.subjectTableAdapter.Fill(this.schooldbDataSet.subject);

}

private void Delete\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

sqlCommand = new SqlCommand("delete subject where SubjectPK=@SubjectPK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@SubjectPK", textBox1.Text);

sqlCommand.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Deleted Successfully!");

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Delete");

}

}

}

}

TeacherAdd.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.AddMode

{

public partial class TeacherAdd : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

private SqlDataAdapter sqlDataAdapter;

public TeacherAdd()

{

InitializeComponent();

}

private void Add\_Click(object sender, EventArgs e)

{

con.Open();

if (con.State == System.Data.ConnectionState.Open)

{

string q = "insert into Teacher(TeacherPK, Name, Post, Salary, Tel)" +

"values('" + Int32.Parse(textBox1.Text) + "','" + textBox2.Text.ToString() + "','" +

textBox3.Text.ToString() + "','" + textBox4.Text.ToString() + "','" + textBox5.Text.ToString() + "') ";

SqlCommand cmd = new SqlCommand(q, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Insert was successful!");

}

con.Close();

Display();

Clear();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

AddChoice choice = new AddChoice();

choice.ShowDialog();

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void TeacherAdd\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.Teacher". При необходимости она может быть перемещена или удалена.

this.teacherTableAdapter.Fill(this.schooldbDataSet.Teacher);

}

private void Update\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox5.Text != "")

{

sqlCommand = new SqlCommand("update Teacher set TeacherPK=@TeacherPK, Name=@Name, Post=@Post, Salary=@Salary, Tel=@Tel", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@TeacherPK", textBox1.Text);

sqlCommand.Parameters.AddWithValue("@Name", textBox2.Text);

sqlCommand.Parameters.AddWithValue("@Post", textBox3.Text);

sqlCommand.Parameters.AddWithValue("@Salary", textBox4.Text);

sqlCommand.Parameters.AddWithValue("@Tel", textBox5.Text);

sqlCommand.ExecuteNonQuery();

MessageBox.Show("Record Updated Successfully");

con.Close();

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Update");

}

}

private void Display()

{

con.Open();

DataTable dt = new DataTable();

sqlDataAdapter = new SqlDataAdapter("select \* from Teacher", con);

sqlDataAdapter.Fill(dt);

dataGridView1.DataSource = dt;

con.Close();

}

private void Clear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

}

private void Delete\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

sqlCommand = new SqlCommand("delete Teacher where TeacherPK=@TeacherPK", con);

con.Open();

sqlCommand.Parameters.AddWithValue("@TeacherPK", textBox1.Text);

sqlCommand.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Deleted Successfully!");

Display();

Clear();

}

else

{

MessageBox.Show("Please Select Record to Delete");

}

}

}

}

GroupView.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class GroupView : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

public GroupView()

{

InitializeComponent();

}

private void GroupView\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.group". При необходимости она может быть перемещена или удалена.

this.groupTableAdapter.Fill(this.schooldbDataSet.group);

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice choice = new ViewChoice();

choice.ShowDialog();

}

//Teacher 1

//Room 2

private void Show\_Click(object sender, EventArgs e)

{

string SqlText = "";

if (checkBox1.Checked == true && checkBox3.Checked == false) //TeacherFK

{

SqlText += "SELECT GroupPK, Name, TeacherFK FROM [dbo].[group]";

if (checkBox1.AutoCheck == true && textBox1.Text.ToString() != "" && checkBox3.AutoCheck == false)

{

SqlText += " WHERE TeacherFK = " + textBox1.Text.ToString();

}

}

if (checkBox3.Checked == true && checkBox1.Checked == false) //RoomFK

{

SqlText += "SELECT GroupPK, Name, RoomFK FROM [dbo].[group]";

if (checkBox3.Checked == true && textBox2.Text.ToString() != "" && checkBox1.Checked == false)

{

SqlText += " WHERE RoomFK = " + textBox2.Text.ToString();

}

}

if (checkBox3.Checked == true && checkBox1.Checked == true) //All

{

SqlText += "SELECT GroupPK, Name, TeacherFK, RoomFK FROM [dbo].[group]";

if (textBox1.Text.ToString() != "" && textBox2.Text.ToString() != "")

{

SqlText += " WHERE TeacherFK = " + textBox1.Text.ToString() + " AND RoomFK = " + textBox2.Text.ToString();

}

}

if (checkBox3.Checked == false && checkBox1.Checked == false) //Nothin

{

SqlText += "SELECT GroupPK, Name FROM [dbo].[group]";

}

SqlDataAdapter da = new SqlDataAdapter(SqlText, con);

DataSet ds = new DataSet();

da.Fill(ds, "[dbo].[group]");

dataGridView1.DataSource = ds.Tables["[dbo].[group]"].DefaultView;

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

}

private void checkBox3\_CheckedChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}

MarksView.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class MarksView : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

public MarksView()

{

InitializeComponent();

}

private void MarksView\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.marks". При необходимости она может быть перемещена или удалена.

this.marksTableAdapter.Fill(this.schooldbDataSet.marks);

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

}

private void Show\_Click(object sender, EventArgs e)

{

string SqlText = "";

if (checkBox1.Checked == true) //TeacherFK

{

SqlText += "SELECT MarkPK, Mark, SubjectFK, PupilFK, TeacherFK FROM [dbo].[marks]";

if (checkBox1.AutoCheck == true && textBox2.Text.ToString() != "")

{

SqlText += " WHERE TeacherFK = " + textBox2.Text.ToString();

}

if (textBox1.Text.ToString() != "")

{

SqlText += " AND PupilFK = " + textBox1.Text.ToString();

}

}

if (textBox1.Text.ToString() != "")

{

SqlText += " WHERE PupilFK = " + textBox1.Text.ToString();

}

if (checkBox1.Checked == false && textBox1.Text.ToString() == "") //Without Teacher

{

SqlText += "SELECT MarkPK, Mark, SubjectFK, PupilFK FROM [dbo].[marks]";

}

SqlDataAdapter da = new SqlDataAdapter(SqlText, con);

DataSet ds = new DataSet();

da.Fill(ds, "[dbo].[marks]");

dataGridView1.DataSource = ds.Tables["[dbo].[marks]"].DefaultView;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice choice = new ViewChoice();

choice.ShowDialog();

}

}

}

PupilsView.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class PupilsView : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

public PupilsView()

{

InitializeComponent();

}

private void PupilsView\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.pupil". При необходимости она может быть перемещена или удалена.

this.pupilTableAdapter.Fill(this.schooldbDataSet.pupil);

}

private void Show\_Click(object sender, EventArgs e)

{

string SqlText = "";

if (checkBox1.Checked == true) //Tel

{

SqlText += "SELECT PupilPK, Name, GroupFK, Tel FROM [dbo].[pupil]";

if (checkBox1.AutoCheck == true && textBox2.Text.ToString() != "")

{

SqlText += " WHERE Tel = '" + textBox2.Text.ToString() + "'";

}

if (textBox1.Text.ToString() != "")

{

SqlText += " AND GroupFK = " + textBox1.Text.ToString();

}

}

if (checkBox1.Checked == false)

{

SqlText += "SELECT PupilPK, Name, GroupFK FROM [dbo].[pupil]";

if (textBox1.Text.ToString() != "")

{

SqlText += "WHERE GroupFK = " + textBox1.Text.ToString();

}

}

SqlDataAdapter da = new SqlDataAdapter(SqlText, con);

DataSet ds = new DataSet();

da.Fill(ds, "[dbo].[pupil]");

dataGridView1.DataSource = ds.Tables["[dbo].[pupil]"].DefaultView;

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice choice = new ViewChoice();

choice.ShowDialog();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}

RoomView.cs

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class RoomView : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

public RoomView()

{

InitializeComponent();

}

private void RoomView\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.room". При необходимости она может быть перемещена или удалена.

this.roomTableAdapter.Fill(this.schooldbDataSet.room);

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice choice = new ViewChoice();

choice.ShowDialog();

}

}

}

SubjectView.cs

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class SubjectView : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

public SubjectView()

{

InitializeComponent();

}

private void SubjectView\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.subject". При необходимости она может быть перемещена или удалена.

this.subjectTableAdapter.Fill(this.schooldbDataSet.subject);

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice choice = new ViewChoice();

choice.ShowDialog();

}

}

}

TeacherView.cs

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class TeacherView : Form

{

SqlConnection con = new SqlConnection("Data Source = (LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Visual Studio VSIAKII SHIT\\SQLDevelop\\CourseWork\\SchoolApp\\SchoolApp\\SchoolApp\\schooldb.mdf; Integrated Security = True; Connect Timeout = 30");

SqlCommand sqlCommand;

public TeacherView()

{

InitializeComponent();

}

private void TeacherView\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "schooldbDataSet.Teacher". При необходимости она может быть перемещена или удалена.

this.teacherTableAdapter.Fill(this.schooldbDataSet.Teacher);

}

private void Show\_Click(object sender, EventArgs e)

{

string SqlText = "";

if (checkBox1.Checked == true && checkBox3.Checked == false) //Salary

{

SqlText += "SELECT TeacherPK, Name, Post, Salary FROM [dbo].[Teacher]";

if (checkBox1.AutoCheck == true && textBox1.Text.ToString() != "" && checkBox3.AutoCheck == false)

{

SqlText += " WHERE Salary = " + textBox1.Text.ToString();

}

}

if (checkBox3.Checked == true && checkBox1.Checked == false) //Tel

{

SqlText += "SELECT TeacherPK, Name, Post, Tel FROM [dbo].[Teacher]";

if (checkBox3.Checked == true && textBox2.Text.ToString() != "" && checkBox1.Checked == false)

{

SqlText += " WHERE Tel = '" + textBox2.Text.ToString() + "'";

}

}

if (checkBox3.Checked == true && checkBox1.Checked == true) //All

{

SqlText += "SELECT TeacherPK, Name, Post, Salary, Tel FROM [dbo].[Teacher]";

if (textBox1.Text.ToString() != "" && textBox2.Text.ToString() != "")

{

SqlText += " WHERE Salary = " + textBox1.Text.ToString() + " AND Tel = '" + textBox2.Text.ToString() + "'";

}

}

if (checkBox3.Checked == false && checkBox1.Checked == false) //Nothin

{

SqlText += "SELECT TeacherPK, Name, Post FROM [dbo].[Teacher]";

}

SqlDataAdapter da = new SqlDataAdapter(SqlText, con);

DataSet ds = new DataSet();

da.Fill(ds, "[dbo].[Teacher]");

dataGridView1.DataSource = ds.Tables["[dbo].[Teacher]"].DefaultView;

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ViewChoice choice = new ViewChoice();

choice.ShowDialog();

}

}

}

ViewChoice.cs

using SchoolApp.MainMenu;

using System;

using System.Windows.Forms;

namespace SchoolApp.ViewMode

{

public partial class ViewChoice : Form

{

public ViewChoice()

{

InitializeComponent();

}

private void Back\_Click(object sender, EventArgs e)

{

this.Visible = false;

ChoiceMenu choice = new ChoiceMenu();

choice.ShowDialog();

}

private void Teacher\_Click(object sender, EventArgs e)

{

this.Visible = false;

TeacherView choice = new TeacherView();

choice.ShowDialog();

}

private void Subject\_Click(object sender, EventArgs e)

{

this.Visible = false;

SubjectView choice = new SubjectView();

choice.ShowDialog();

}

private void Pupils\_Click(object sender, EventArgs e)

{

this.Visible = false;

PupilsView choice = new PupilsView();

choice.ShowDialog();

}

private void Marks\_Click(object sender, EventArgs e)

{

this.Visible = false;

MarksView choice = new MarksView();

choice.ShowDialog();

}

private void Groups\_Click(object sender, EventArgs e)

{

this.Visible = false;

GroupView choice = new GroupView();

choice.ShowDialog();

}

private void Rooms\_Click(object sender, EventArgs e)

{

this.Visible = false;

RoomView choice = new RoomView();

choice.ShowDialog();

}

}

}