



```
using System.Windows;
using WpfAdmin.Pages;

namespace WpfAdmin
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            // Bijvoorbeeld start je hier nog niets of navigatie naar LoginPage

            // Deze methode wordt door LoginPage aangeroepen na succesvolle login
            public void ToonDashboard()
            {
                // Toon sidebar
                NavPanel.Visibility = Visibility.Visible;

                // Navigeer naar de gewenste startpagina (bijvoorbeeld PersoonBewerken)
                MainFrame.Navigate(new PersoonBewerken());
            }

            private void BtnPersonen_Click(object sender, RoutedEventArgs e)
            {
                MainFrame.Navigate(new PersoonBewerken());
            }

            private void BtnUitloggen_Click(object sender, RoutedEventArgs e)
            {
                // Verberg sidebar en navigeer naar lege pagina of login
                NavPanel.Visibility = Visibility.Collapsed;
                MainFrame.Content = null;
            }
        }
    }
}

<Page x:Class="WpfAdmin.Pages.LoginPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="LoginPage">

    <Grid Background="White">
        <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
            <TextBlock Text="Gebruikersnaam" Margin="0,5"/>
            <TextBox x:Name="txtGebruikersnaam" Width="200" Margin="0,5"/>

            <TextBlock Text="Wachtwoord" Margin="0,5"/>
        </StackPanel>
    </Grid>
</Page>
```

```

        <PasswordBox x:Name="txtWachtwoord" Width="200" Margin="0,5"/>
        <Button x:Name="btnLogin" Content="Inloggen" Width="100" Margin="0,10"
Click="btnLogin_Click"/>
    </StackPanel>
</Grid>
</Page>
<Window x:Class="WpfAdmin.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="WPF Admin" Height="600" Width="900">

    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <!-- Sidebar -->
        <StackPanel x:Name="NavPanel" Background="#f0f0f0" Grid.Column="0" Visibility="Collapsed">
            <Button Name="btnPersonen" Content="👤 Personen" Margin="10"
Click="BtnPersonen_Click" />
            <Button Name="btnUitloggen" Content="🚪 Uitloggen" Margin="10"
Click="BtnUitloggen_Click" />
        </StackPanel>

        <!-- Main content -->
        <Frame Name="MainFrame" Grid.Column="1" NavigationUIVisibility="Hidden" />
    </Grid>
</Window>
using System.Windows;
using WpfAdmin.Pages;

namespace WpfAdmin
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            // Start met loginpagina zichtbaar en sidebar verborgen
            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }

        // Wordt aangeroepen door LoginPage na succesvolle login
        public void LoginGeslaagd()
        {
            NavPanel.Visibility = Visibility.Visible;
            MainFrame.Navigate(new PersoonBewerken()); // Ga naar overzichtpagina na login
        }
    }
}

```

```

private void BtnPersonen_Click(object sender, RoutedEventArgs e)
{
    MainFrame.Navigate(new PersoonBewerken());
}

private void BtnUitloggen_Click(object sender, RoutedEventArgs e)
{
    NavPanel.Visibility = Visibility.Collapsed;
    MainFrame.Navigate(new LoginPage(this)); // Terug naar loginpagina
}
}
}

<Page x:Class="WpfAdmin.PersoonBewerken"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Persoon Bewerken" Margin="10">

    <StackPanel Width="300" HorizontalAlignment="Center" VerticalAlignment="Top">
        <Label Content="Voornaam:"/>
        <TextBox Name="VoornaamBox" Margin="0,0,0,10"/>

        <Label Content="Achternaam:"/>
        <TextBox Name="AchternaamBox" Margin="0,0,0,10"/>

        <Label Content="Email:"/>
        <TextBox Name="EmailBox" Margin="0,0,0,10"/>

        <Label Content="Wachtwoord:"/>
        <PasswordBox Name="WachtwoordBox" Margin="0,0,0,10"/>

        <Label Content="Rol:"/>
        <ComboBox Name="RolBox" Margin="0,0,0,10">
            <ComboBoxItem Content="Admin"/>
            <ComboBoxItem Content="User"/>
            <ComboBoxItem Content="Manager"/>
        </ComboBox>

        <Button Content="Opslaan" Click="BtnOpslaan_Click" Width="100"
HorizontalAlignment="Center"/>
    </StackPanel>
</Page>

using System.Windows;
using System.Windows.Controls;
using ICCompany;

namespace WpfAdmin
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()

```

```
{
    InitializeComponent();
}

private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
{
    Person persoon = new Person
    {
        Voornaam = VoornaamBox.Text,
        Achternaam = AchternaamBox.Text,
        Email = EmailBox.Text,
        Wachtwoord = WachtwoordBox.Password,
        Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
    };

    // TODO: Opslaan in DB via class library, nu enkel MessageBox tonen
    MessageBox.Show("Persoon opgeslagen:\n" +
        $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

    // Terug naar overzicht (bedrijfs-pagina)
    this.NavigationService.Navigate(new BedrijvenOverzichtPage());
}
}
} fix mijn fouten aub zodat het werkt
```

Je hebt al een heel mooie basis, maar er zitten een paar dingen in je code die problemen kunnen veroorzaken of die ontbreken om het te laten werken. Ik ga ze stuk voor stuk voor je corrigeren en toelichten.

1. LoginPage constructor en communicatie met MainWindow

Je navigatie naar `LoginPage` stuurt `this` (MainWindow) door, maar in je XAML van LoginPage is geen constructor met parameter gedefinieerd. Je moet dus een constructor maken die `MainWindow` ontvangt zodat je daar de login kunt melden.

LoginPage.xaml.cs:

```
csharp

using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class LoginPage : Page
    {
        private MainWindow _mainWindow;

        public LoginPage(MainWindow mainWindow)
        {
            InitializeComponent();
            _mainWindow = mainWindow;
        }

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

```

{
    string gebruikersnaam = txtGebruikersnaam.Text;
    string wachtwoord = txtWachtwoord.Password;

    // Simpele validatie, vervang door echte logincheck
    if (gebruikersnaam == "admin" && wachtwoord == "wachtwoord")
    {
        _mainWindow.LoginGeslaagd();
    }
    else
    {
        MessageBox.Show("Ongeldige gebruikersnaam of wachtwoord.");
    }
}
}
}

```

Belangrijk: De `btnLogin_Click` was nog niet geïmplementeerd, die heb ik toegevoegd met een voorbeeldlogin.

2. MainWindow aanpassingen

Je hebt al 2 versies van `MainWindow` gegeven. Ik combineer ze in een correcte versie. De sidebar is eerst verborgen, en je start met de loginpagina:

```

csharp

using System.Windows;
using WpfAdmin.Pages;

namespace WpfAdmin
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }

        public void LoginGeslaagd()
        {
            NavPanel.Visibility = Visibility.Visible;
            MainFrame.Navigate(new PersoonBewerken());
        }

        private void BtnPersonen_Click(object sender, RoutedEventArgs e)
        {
            MainFrame.Navigate(new PersoonBewerken());
        }

        private void BtnUitloggen_Click(object sender, RoutedEventArgs e)
        {
            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }
    }
}

```

3. PersoonBewerken Page corrigeren

- Je moet in `PersoonBewerken` namespace aanpassen naar `WpfAdmin.Pages` (zoals in andere pages).
- `BedrijvenOverzichtPage` bestaat niet in jouw voorbeeld, dus ik verander dat naar `PersonenOverzichtPage` of laat het navigeren naar `null` of iets anders.
- `Person` class moet ergens gedefinieerd zijn, je refereert naar `CICompany` namespace. Als die er niet is, geef ik een voorbeeldklasse.

PersoonBewerken.xaml.cs:

```
csharp

using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
        {
            Person persoon = new Person
            {
                Voornaam = VoornaamBox.Text,
                Achternaam = AchternaamBox.Text,
                Email = EmailBox.Text,
                Wachtwoord = WachtwoordBox.Password,
                Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
            };

            // Nu tijdelijk MessageBox
            MessageBox.Show("Persoon opgeslagen:\n" +
                $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

            // Terug naar overzichtspagina (hier: PersonenOverzichtPage of leeg)
            // this.NavigationService.Navigate(new PersonenOverzichtPage());
            this.NavigationService.GoBack(); // Of iets anders
        }

        // Dummy Person class als voorbeeld, vervang met jouw echte model
        public class Person
        {
            public string Voornaam { get; set; }
            public string Achternaam { get; set; }
            public string Email { get; set; }
            public string Wachtwoord { get; set; }
            public string Rol { get; set; }
        }
    }
}
```

4. LoginPage XAML - toevoegen Click event handler

In je XAML `LoginPage.xaml` moet het `Click` event gekoppeld zijn aan de juiste handler. Je had al `Click="btnLogin_Click"`, dus dat is goed.

5. Namen en namespaces consistent houden

- Zorg dat alle pages in namespace ``WpfAdmin.Pages`` staan.
- MainWindow in namespace ``WpfAdmin``.
- Pages via ``MainFrame.Navigate(new LoginPage(this))`` etc.

Samenvatting van de belangrijkste wijzigingen die je moet toepassen:

1. **LoginPage.xaml.cs:** Voeg constructor met ``MainWindow`` parameter toe en login-check.
2. **MainWindow.xaml.cs:** Zorg dat je MainWindow constructor ``LoginPage(this)`` gebruikt en method ``LoginGeslaagd()`` aanroept.
3. **PersoonBewerken.xaml.cs:** Pas namespace aan, voeg dummy ``Person`` toe, en pas navigatie aan na opslaan.
4. **XAML-bestanden:** Controleren dat events correct zijn gekoppeld.

Wil je dat ik de volledige bestanden voor je genereer met alle correcte codes erin? Dan kun je die 1-op-1 overnemen.



```
<Page x:Class="WpfAdmin.PersoonBewerken"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Persoon Bewerken" Margin="10">

    <StackPanel Width="300" HorizontalAlignment="Center" VerticalAlignment="Top">
        <Label Content="Voornaam:"/>
        <TextBox Name="VoornaamBox" Margin="0,0,0,10"/>

        <Label Content="Achternaam:"/>
        <TextBox Name="AchternaamBox" Margin="0,0,0,10"/>

        <Label Content="Email:"/>
        <TextBox Name="EmailBox" Margin="0,0,0,10"/>

        <Label Content="Wachtwoord:"/>
        <PasswordBox Name="WachtwoordBox" Margin="0,0,0,10"/>

        <Label Content="Rol:"/>
        <ComboBox Name="RolBox" Margin="0,0,0,10">
            <ComboBoxItem Content="Admin"/>
            <ComboBoxItem Content="User"/>
            <ComboBoxItem Content="Manager"/>
        </ComboBox>

        <Button Content="Opslaan" Click="BtnOpslaan_Click" Width="100"
            HorizontalAlignment="Center"/>
    </StackPanel>
</Page>
```

```

</StackPanel>
</Page>
Severity Code    Description Project File Line    Suppression State
Error    CS0103 The name 'RolBox' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 21 Active
Error    CS0103 The name 'AchternaamBox' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 18 Active
Error    CS0103 The name 'EmailBox' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 19 Active
Error    CS0103 The name 'InitializeComponent' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 10 Active
Error    CS0103 The name 'VoornaamBox' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 17 Active
Error    CS0103 The name 'WachtwoordBox' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 20 Active
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
        {
            Person persoon = new Person
            {
                Voornaam = VoornaamBox.Text,
                Achternaam = AchternaamBox.Text,
                Email = EmailBox.Text,
                Wachtwoord = WachtwoordBox.Password,
                Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
            };

            // Nu tijdelijk MessageBox
            MessageBox.Show("Persoon opgeslagen:\n" +
                $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

            // Terug naar overzichtspagina (hier: PersonenOverzichtPage of leeg)
            // this.NavigationService.Navigate(new PersonenOverzichtPage());
            this.NavigationService.GoBack(); // Of iets anders

```



```

    }
}

// Dummy Person class als voorbeeld, vervang met jouw echte model
public class Person
{
    public string Voornaam { get; set; }
    public string Achternaam { get; set; }
    public string Email { get; set; }
    public string Wachtwoord { get; set; }
    public string Rol { get; set; }
}
}

```

De fouten die je krijgt wijzen erop dat je ``PersoonBewerken.xaml.cs`` en ``PersoonBewerken.xaml`` niet correct aan elkaar gekoppeld zijn door Visual Studio. De belangrijkste oorzaak is meestal dat de ``.xaml.cs`` niet als **partial class** binnen dezelfde namespace en classnaam staat, of dat de ``x:Class`` in de XAML niet overeenkomt met de namespace + class in de ``.cs``.

Stap 1: Controleer ``x:Class`` in je XAML

Je hebt dit:

```

xml

<Page x:Class="WpfAdmin.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Persoon Bewerken" Margin="10">

```

Let op: ``x:Class="WpfAdmin.PersoonBewerken"`` maar je codebehind staat in namespace ``WpfAdmin.Pages``.

Oplossing: Zet in je XAML

```

xml

x:Class="WpfAdmin.Pages.PersoonBewerken"

```

Stap 2: Controleer namespace in codebehind

In je ``PersoonBewerken.xaml.cs``:

```

csharp

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        // ...
    }
}

```

```
}
}
```

Dat klopt nu, want `WpfAdmin.Pages` is dezelfde namespace als in de XAML (nadat je die hebt aangepast).

Stap 3: Build opnieuw

Sla beide bestanden op en **clean** en **rebuild** je project in Visual Studio (menu: Build → Clean Solution en daarna Build → Build Solution).

Stap 4: Controleer of je .xaml en .xaml.cs bestanden gekoppeld zijn in Visual Studio

- Rechtsklik in Solution Explorer op `PersoonBewerken.xaml`
- Kies **Open With...** → zorg dat het opent in XAML Editor
- Zorg dat `PersoonBewerken.xaml.cs` als codebehind is gekoppeld (moet automatisch)

Samengevat

Je XAML moet bovenaan zijn:

```
xml

<Page x:Class="WpfAdmin.Pages.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Persoon Bewerken" Margin="10">
```

Je `PersoonBewerken.xaml.cs`:

```
csharp

using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
        {
            Person persoon = new Person
            {
                Voornaam = VoornaamBox.Text,
                Achternaam = AchternaamBox.Text,
                Email = EmailBox.Text,
                Wachtwoord = WachtwoordBox.Password,
                Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
            };

            MessageBox.Show("Persoon opgeslagen:\n" +
                            $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

            this.NavigationService.GoBack();
        }
    }
}
```

```

    }

    public class Person
    {
        public string Voornaam { get; set; }
        public string Achternaam { get; set; }
        public string Email { get; set; }
        public string Wachtwoord { get; set; }
        public string Rol { get; set; }
    }
}

```

Als je dit aanpast, en daarna je project herbouwt, zou `InitializeComponent()` en de namen van je controls (zoals `rolBox`) herkend moeten worden en verdwijnen de fouten.

Wil je dat ik je help met een volledig minimalistisch WPF-projectvoorbeeld waarin dit werkt? Dan kan ik dat ook voor je genereren.



pas de hele codes aan

Zeker! Hieronder vind je de volledig aangepaste en samenhangende code voor je WPF Admin-project met:

- `MainWindow.xaml` + `MainWindow.xaml.cs`
- `LoginPage.xaml` + `LoginPage.xaml.cs`
- `PersoonBewerken.xaml` + `PersoonBewerken.xaml.cs`

Alle namespace- en classnamen zijn consistent en correct gekoppeld.

1) MainWindow.xaml

```

xml

<Window x:Class="WpfAdmin.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="WPF Admin" Height="600" Width="900">

    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <!-- Sidebar -->
        <StackPanel x:Name="NavPanel" Background="#f0f0f0" Grid.Column="0" Visibility="Collapsed">
            <Button Name="btnPersonen" Content="👤 Personen" Margin="10" Click="BtnPersonen_Click" />
            <Button Name="btnUitloggen" Content="🚪 Uitleggen" Margin="10" Click="BtnUitloggen_Click" />
        </StackPanel>

        <!-- Main content -->
        <Frame Name="MainFrame" Grid.Column="1" NavigationUIVisibility="Hidden" />
    </Grid>
</Window>

```

2) MainWindow.xaml.cs

```
csharp

using System.Windows;
using WpfAdmin.Pages;

namespace WpfAdmin
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            // Start met loginpagina zichtbaar en sidebar verborgen
            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }

        // Wordt aangeroepen door LoginPage na succesvolle login
        public void LoginGeslaagd()
        {
            NavPanel.Visibility = Visibility.Visible;
            MainFrame.Navigate(new PersoonBewerken()); // Ga naar bewerken na login
        }

        private void BtnPersonen_Click(object sender, RoutedEventArgs e)
        {
            MainFrame.Navigate(new PersoonBewerken());
        }

        private void BtnUitloggen_Click(object sender, RoutedEventArgs e)
        {
            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this)); // Terug naar loginpagina
        }
    }
}
```

3) LoginPage.xaml

```
xml

<Page x:Class="WpfAdmin.Pages.LoginPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="LoginPage">

    <Grid Background="White">
        <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
            <TextBlock Text="Gebruikersnaam" Margin="0,5"/>
            <TextBox x:Name="txtGebruikersnaam" Width="200" Margin="0,5"/>

            <TextBlock Text="Wachtwoord" Margin="0,5"/>
            <PasswordBox x:Name="txtWachtwoord" Width="200" Margin="0,5"/>
            <Button x:Name="btnLogin" Content="Inloggen" Width="100" Margin="0,10" Click="btnLogin_Click"/>
        </StackPanel>
    </Grid>
</Page>
```

4) LoginPage.xaml.cs

csharp

```
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class LoginPage : Page
    {
        private readonly MainWindow _mainWindow;

        public LoginPage(MainWindow mainWindow)
        {
            InitializeComponent();
            _mainWindow = mainWindow;
        }

        private void btnLogin_Click(object sender, RoutedEventArgs e)
        {
            string gebruiker = txtGebruikersnaam.Text;
            string wachtwoord = txtWachtwoord.Password;

            // Simpele check (vervang door echte authenticatie)
            if (gebruiker == "admin" && wachtwoord == "wachtwoord")
            {
                _mainWindow.LoginGeslaagd();
            }
            else
            {
                MessageBox.Show("Onjuiste gebruikersnaam of wachtwoord.", "Fout", MessageBoxButton.OK);
            }
        }
    }
}
```

5) PersoonBewerken.xaml

xml

```
<Page x:Class="WpfAdmin.Pages.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Persoon Bewerken" Margin="10">

    <StackPanel Width="300" HorizontalAlignment="Center" VerticalAlignment="Top">
        <Label Content="Voornaam:"/>
        <TextBox Name="VoornaamBox" Margin="0,0,0,10"/>

        <Label Content="Achternaam:"/>
        <TextBox Name="AchternaamBox" Margin="0,0,0,10"/>

        <Label Content="Email:"/>
        <TextBox Name="EmailBox" Margin="0,0,0,10"/>

        <Label Content="Wachtwoord:"/>
        <PasswordBox Name="WachtwoordBox" Margin="0,0,0,10"/>

        <Label Content="Rol:"/>
        <ComboBox Name="RolBox" Margin="0,0,0,10">
            <ComboBoxItem Content="Admin"/>
            <ComboBoxItem Content="User"/>
            <ComboBoxItem Content="Manager"/>
        </ComboBox>

        <Button Content="Opslaan" Click="BtnOpslaan_Click" Width="100" HorizontalAlignment="Center",
    </StackPanel>
</Page>
```

6) PersoonBewerken.xaml.cs

```
csharp

using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
        {
            Person persoon = new Person
            {
                Voornaam = VoornaamBox.Text,
                Achternaam = AchternaamBox.Text,
                Email = EmailBox.Text,
                Wachtwoord = WachtwoordBox.Password,
                Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
            };

            MessageBox.Show("Persoon opgeslagen:\n" +
                $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

            // Ga terug naar vorige pagina (of maak hier een overzichtspagina)
            if (this.NavigationService.CanGoBack)
            {
                this.NavigationService.GoBack();
            }
        }
    }

    // Dummy Person class als voorbeeld, vervang door je eigen model / database
    public class Person
    {
        public string Voornaam { get; set; }
        public string Achternaam { get; set; }
        public string Email { get; set; }
        public string Wachtwoord { get; set; }
        public string Rol { get; set; }
    }
}
```

Extra tips

- Zorg dat je projectmap structuur klopt: `Pages` folder met `LoginPage` en `PersoonBewerken` erin.
- Build clean → build om alles te synchroniseren.
- Het `MainWindow` moet het startvenster zijn (check in project properties).

Met deze code werkt je login + navigatie naar bewerken pagina met sidebar netjes.

Als je wilt kan ik ook nog de `PersonenOverzichtPage` maken of extra functionaliteit toevoegen. Laat maar weten!



de bewerk persoon page moet pas vertoond worden als ik op de button persoon druk

Dat kan ik regelen! Je wilt dus dat:

- Bij starten zie je alleen de loginpagina.
- Na succesvolle login verschijnt de navigatie (sidebar) met de knop "Personen".
- Pas als je op die knop "Personen" klikt, wordt de `PersoonBewerken`-pagina getoond.

Dat betekent dat je na login niet meteen naar `PersoonBewerken` navigeert, maar naar een lege pagina of bijvoorbeeld een welkomspagina, en pas na klik op "Personen" de bewerkpagina laadt.

Ik pas de code van `MainWindow.xaml.cs` en `LoginPage.xaml.cs` hierop aan.

Aangepaste MainWindow.xaml.cs

```
csharp

using System.Windows;
using WpfAdmin.Pages;

namespace WpfAdmin
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }

        public void LoginGeslaagd()
        {
            NavPanel.Visibility = Visibility.Visible;

            // Na login navigeren we eerst naar een lege pagina of welkom pagina
            MainFrame.Content = null; // Of maak een welkomspagina als je wil
        }

        private void BtnPersonen_Click(object sender, RoutedEventArgs e)
        {
            MainFrame.Navigate(new PersoonBewerken());
        }

        private void BtnUitloggen_Click(object sender, RoutedEventArgs e)
        {
            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }
    }
}
```

Aangepaste LoginPage.xaml.cs

csharp

```
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class LoginPage : Page
    {
        private readonly MainWindow _mainWindow;

        public LoginPage(MainWindow mainWindow)
        {
            InitializeComponent();
            _mainWindow = mainWindow;
        }

        private void btnLogin_Click(object sender, RoutedEventArgs e)
        {
            string gebruiker = txtGebruikersnaam.Text;
            string wachtwoord = txtWachtwoord.Password;

            if (gebruiker == "admin" && wachtwoord == "wachtwoord")
            {
                _mainWindow.LoginGeslaagd();
            }
            else
            {
                MessageBox.Show("Onjuiste gebruikersnaam of wachtwoord.", "Fout", MessageBoxButton.OK);
            }
        }
    }
}
```

Wat gebeurt er nu?

- Bij starten open je loginpagina, sidebar is verborgen.
- Na login verschijnt de sidebar, maar je blijft nog op een lege pagina (of een pagina die je kan toevoegen).
- Pas als je op **Personen** klikt, wordt `PersoonBewerken` geladen.

Optioneel: Maak een lege welkomspagina om netjes te zijn

Maak een nieuwe `WelcomePage.xaml`:

```
xml

<Page x:Class="WpfAdmin.Pages.WelcomePage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Welkom">
    <Grid>
        <TextBlock Text="Welkom bij de Admin" FontSize="24"
                   HorizontalAlignment="Center" VerticalAlignment="Center"/>
    </Grid>
</Page>
```

En `WelcomePage.xaml.cs`:

csharp

```
namespace WpfAdmin.Pages
{
    public partial class WelcomePage : Page
    {
        public WelcomePage()
        {
            InitializeComponent();
        }
    }
}
```

En in `MainWindow.xaml.cs` vervang je in `LoginGeslaagd()` de regel:

csharp

```
MainFrame.Content = null;
```

door

csharp

```
MainFrame.Navigate(new WelcomePage());
```

Wil je dat ik die welkomspagina voor je maak?

Of wil je dat ik je help met een `PersonenOverzichtPage` zodat je vanuit daar op een persoon kan klikken om te bewerken?



id	name	contact	address	zip	city	country	phone	email	btw	login	password
regdate	acceptdate	lastmodified	status	language	logo	nacecode	code				
1	TechNova Solutions	Pieter Senhadji	Innovation Street 1	3000	Leuven	België	+32 16 111 222	pieter.s@technova.be	BE7777777777	psenhadji	hashedpassword 2022-05-12 00:00:00.000
2	Cyber Dynamics	Ayoub Amaadachou	Digital Boulevard 5	2000	Antwerpen	België	+32 3 222 333	ayoub.a@cyberdynamics.be	BE8888888888	aamaadachou	hashedpassword 2023-01-17 00:00:00.000
3	PixelWave Systems	Brian Kodio	Tech Park 12	3000	Gent	België	+32 9 333 444	brian.k@pixelwave.be	BE9999999999	bkodio	hashedpassword 2021-09-03 00:00:00.000
4	NetFusion BV	Sofiane Ndayisabye	Circuit Avenue 7	8500	Kortrijk	België	+32 56 444 555	sofiane.n@netfusion.be	BE1212121212	sndayisabye	hashedpassword 2022-11-20 00:00:00.000
5	CodeCrafters Inc.	Lucas Mutabaruka	Binary Road 3	4000	Luik	België	+32 4 555 666	lucas.m@codecrafters.be	BE1313131313	lmutabaruka	hashedpassword 2023-07-08 00:00:00.000
6	Finastra NV	Ayoub Alazaroae	Kapelstraat 10	1000	Brussel	België	+32 2 222 33 44	ayoub.a@finastra.be	BE1111111111	aalazaroae	hashedpassword 2023-02-14 00:00:00.000
7	EuroTrust Finance	Fadela Van Cleemput	Financieringslaan 4	2000	Antwerpen	België					

```

+32 3 333 44 55 f.vancleemput@eurotrust.be BE2222222222 f.vancleemput
hashedpassword 2022-09-05 00:00:00.000 2022-09-12 00:00:00.000 2023-10-18 00:00:00.000
active nl NULL 64.19
8 BelBank Solutions Tarik Daneau Zuidstationstraat 8 3000 Leuven België +32 16
444 55 66 tdaneau@belbank.be BE3333333333 tdaneau hashedpassword 2021-11-22
00:00:00.000 2021-12-01 00:00:00.000 2023-09-09 00:00:00.000 active nl NULL 64.19
9 CapitalCore BV Rania Kharbouch Kustlaan 20 8300 Knokke-Heist België +32 50 555
66 77 rania.k@capitalcore.be BE4444444444 rkharbouch hashedpassword 2022-03-30
00:00:00.000 2022-04-08 00:00:00.000 2024-01-21 00:00:00.000 active nl NULL 64.19
10 OptiFin Group Mehmet Ali El Abbouti Ringweg 1 3500 Hasselt België +32 11 666
77 88 mehmet.ea@optifin.be BE5555555555 maelabbouti hashedpassword 2023-05-17
00:00:00.000 2023-05-25 00:00:00.000 2024-02-13 00:00:00.000 active nl NULL 64.19
11 BlueRate Finance Yassin Huwaert IJzerlaan 17 9000 Gent België +32 9 777 88 99
yhuwaert@bluerate.be BE6666666666 yhuwaert hashedpassword 2022-07-02 00:00:00.000
2022-07-10 00:00:00.000 2023-12-04 00:00:00.000 active nl NULL 64.19
12 PharmaNova NV Tarik El Abbouti Industrieweg 12 1000 Brussel België +32 2 123 45 67
tarik.ea@pharmanova.be BE0123456789 telabbouti hashedpassword 2022-06-21 00:00:00.000
2022-07-01 00:00:00.000 2023-10-14 00:00:00.000 active nl NULL 21.20
13 Medix Solutions BV Gabriel El Barnoussi Wetenschapspark 3 9000 Gent België
+32 9 234 56 78 gabriel.eb@medixsolutions.be BE9876543210 gelbarnoussi hashedpassword
2023-01-10 00:00:00.000 2023-01-18 00:00:00.000 2024-03-05 00:00:00.000 active nl NULL
21.20
14 HealTech SA Inès Bouzaya Avenue de la Recherche 5 4000 Luik België +32 4 123 45 67
ines.b@healtech.be BE1234567890 ibouzaya hashedpassword 2021-09-14 00:00:00.000
2021-09-20 00:00:00.000 2023-07-30 00:00:00.000 active nl NULL 21.10
15 BioPharm BVBA Semih Hakan Tafahi Zuidlaan 45 2000 Antwerpen België +32 3 456
78 90 semih.t@biopharm.be BE2345678901 stafahi hashedpassword 2022-02-05
00:00:00.000 2022-02-12 00:00:00.000 2023-11-22 00:00:00.000 active nl NULL 21.20
16 NovaMed NV Rania Chakirwa Biluge Nieuwstraat 99 3000 Leuven België +32 16
123 456 rania.cb@novamed.be BE3456789012 rchakirwabiluge hashedpassword 2022-10-15
00:00:00.000 2022-10-22 00:00:00.000 2024-01-18 00:00:00.000 active nl NULL 21.10
17 CureLife Group Mahir Senhadji Ringlaan 7 3500 Hasselt België +32 11 654 321
mahir.s@curelife.be BE4567890123 msenhadji hashedpassword 2023-04-04 00:00:00.000
2023-04-10 00:00:00.000 2024-02-20 00:00:00.000 active nl NULL 21.20
18 TheraPharm Amira Mortier Pharmalaan 2 8000 Brugge België +32 50 987 654
amira.m@therapharm.be BE5678901234 amortier hashedpassword 2022-05-19 00:00:00.000
2022-05-27 00:00:00.000 2023-12-03 00:00:00.000 active nl NULL 21.10
19 GlobalMedics Yasine Ghomrari Internationaleweg 11 8500 Kortrijk België +32 56 321
987 yasine.g@globalmedics.be BE6789012345 yghomrari hashedpassword 2021-12-01
00:00:00.000 2021-12-09 00:00:00.000 2023-09-14 00:00:00.000 active nl NULL
21.20SELECT TOP (1000) [id]
,[name]
,[contact]
,[address]
,[zip]
,[city]
,[country]
,[phone]
,[email]
,[btw]
,[login]

```

```
,[password]
,[regdate]
,[acceptdate]
,[lastmodified]
,[status]
,[language]
,[logo]
,[nacecode_code]
FROM [BenchmarkDB].[dbo].[Companies]
-- Yasser El Aouati
-- Create Database EP3_q1720602;

CREATE DATABASE EP3_q1720602;
USE EP3_q1720602;

--Aanmaken van tabel ActiviteitTypes

CREATE TABLE ActiviteitTypes (
    id INT PRIMARY KEY IDENTITY,
    type VARCHAR(255) NOT NULL UNIQUE
);
--Aanmaken van tabel Personen

CREATE TABLE Personen (
    id INT PRIMARY KEY IDENTITY,
    voornaam VARCHAR(255) NULL,
    familienaam VARCHAR(255) NOT NULL,
    islid BIT NOT NULL DEFAULT 0
);
--Aanmaken van tabel Activiteiten

CREATE TABLE Activiteiten (
    id INT PRIMARY KEY IDENTITY,
    naam VARCHAR(255) NOT NULL,
    idtype INT NOT NULL,
    prijs DECIMAL(10, 2) NULL CHECK (prijs > 0),
    prijslid DECIMAL(10, 2) NULL CHECK (prijslid < prijs OR prijslid IS NULL),
    FOREIGN KEY (idtype) REFERENCES ActiviteitTypes(id),
    datum DATE NOT NULL,
    maxdeelnemers INT NULL,
    CHECK (datum > GETDATE()),
    CHECK (maxdeelnemers IS NULL OR maxdeelnemers >= 10)
);
--Aanmaken van tabel Inschrijvingen

CREATE TABLE Inschrijvingen (
    id INT PRIMARY KEY IDENTITY,
    idpersoon INT NOT NULL,
    idactiviteit INT NOT NULL,
    deelnemers INT NOT NULL CHECK (deelnemers > 0),
```

```

datum DATE NOT NULL DEFAULT GETDATE(),
status CHAR(1) NOT NULL CHECK (status IN ('I', 'W')),
FOREIGN KEY (idpersoon) REFERENCES Personen(id),
FOREIGN KEY (idactiviteit) REFERENCES Activiteiten(id)
);

-- Mockdata plaatsen in tabel personen

INSERT INTO Personen (voornaam, familienaam, islid)
VALUES ('Yasser', 'El Aouati', 1),
       ('Adam', 'Amri', 0),
       ('Matheo', 'De Bakker', 0);

-- Mockdata plaatsen in tabel Activiteittypes

INSERT INTO ActiviteitTypes (type)
VALUES ('Film'),
       ('Wandeling'),
       ('Vergadering'),
       ('Cursus'),
       ('Fietstocht');
GO

-- aanmaken van de functie udfTeBetalen

CREATE FUNCTION udfTeBetalen (@inschrijvingID INT)
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @prijs DECIMAL(10, 2);
    DECLARE @prijslid DECIMAL(10, 2);
    DECLARE @deelnemers INT;
    DECLARE @islid BIT;

    SELECT @prijs = A.prijs, @prijslid = A.prijslid, @deelnemers = I.deelnemers, @islid = P.islid
    FROM Inschrijvingen I
    JOIN Activiteiten A ON I.idactiviteit = A.id
    JOIN Personen P ON I.idpersoon = P.id
    WHERE I.id = @inschrijvingID;

    IF @prijs IS NULL
        RETURN 0;
    IF @inschrijvingID IS NULL
        RETURN 1;
    IF @islid = 1
        RETURN @prijslid * @deelnemers;
    ELSE
        RETURN @prijs * @deelnemers;
END;
GO

```

```
-- aanmaken van de functie udfInschrijvingAantal

CREATE FUNCTION udfInschrijvingAantal (@activiteitID INT, @type INT)
RETURNS INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Activiteiten WHERE id = @activiteitID)
        RETURN -1;

    IF @type = 0
        RETURN (SELECT SUM(deelnemers) FROM Inschrijvingen WHERE idactiviteit = @activiteitID AND
status = 'I');
    ELSE IF @type = 1
        RETURN (SELECT maxdeelnemers - SUM(deelnemers)
        FROM Activiteiten A
        JOIN Inschrijvingen I ON A.id = I.idactiviteit
        WHERE A.id = @activiteitID AND I.status = 'I');
    RETURN NULL;
END;
GO

-- Aanmaken van view vewTeBetalen

CREATE VIEW vewTeBetalen AS
SELECT I.id, N.naam, F.familienaam, dbo.udfTeBetalen(I.id) AS teBetalen
FROM Inschrijvingen I
JOIN Activiteiten A ON I.idactiviteit = A.id
JOIN Personen P ON I.idpersoon = P.id;
GO

-- Aanmaken van procedure sprActiviteitToevoegen

CREATE PROCEDURE sprActiviteitToevoegen
    @naam VARCHAR(255),
    @idtype INT,
    @datum DATE,
    @maxdeelnemers INT,
    @prijs DECIMAL(10, 2),
    @prijslid DECIMAL(10, 2)
AS
BEGIN
    IF @naam IS NULL OR LEN(@naam) = 0 RETURN 1;
    IF NOT EXISTS (SELECT 1 FROM ActiviteitTypes WHERE id = @idtype) RETURN 2;
    IF @datum <= GETDATE() RETURN 3;
    IF @maxdeelnemers IS NOT NULL AND @maxdeelnemers < 10 RETURN 4;
    IF @prijs IS NOT NULL AND @prijs < 1 RETURN 5;
    IF @prijslid IS NOT NULL AND @prijslid >= @prijs RETURN 6;

    INSERT INTO Activiteiten (naam, idtype, datum, maxdeelnemers, prijs, prijslid)
    VALUES (@naam, @idtype, @datum, @maxdeelnemers, @prijs, @prijslid);
END;
```

```

GO
-- Aanmaken van procedure sprInschrijvingToevoegen

CREATE PROCEDURE sprInschrijvingToevoegen
    @idpersoon INT,
    @idactiviteit INT,
    @deelnemers INT
AS
BEGIN
    DECLARE @maxdeelnemers INT;
    DECLARE @inschrijvingen INT;

    SELECT @maxdeelnemers = maxdeelnemers FROM Activiteiten WHERE id = @idactiviteit;
    SELECT @inschrijvingen = dbo.udfInschrijvingAantal(@idactiviteit, 1);

    IF @maxdeelnemers IS NULL OR @inschrijvingen IS NULL OR @inschrijvingen >= @deelnemers
    BEGIN
        INSERT INTO Inschrijvingen (idpersoon, idactiviteit, deelnemers, status)
        VALUES (@idpersoon, @idactiviteit, @deelnemers, 'I');
        RETURN 0;
    END
    ELSE
    BEGIN
        INSERT INTO Inschrijvingen (idpersoon, idactiviteit, deelnemers, status)
        VALUES (@idpersoon, @idactiviteit, @deelnemers, 'W');
        RETURN 1;
    END
END;
GO
-- aanmaken van de trigger udfTeBetalen

CREATE TRIGGER trgUpdateWachtrij
ON Inschrijvingen
AFTER DELETE
AS
BEGIN
    DECLARE @activiteitID INT, @vrijPlaatsen INT, @idWachtrij INT, @deelnemers INT;

    SELECT @activiteitID = deleted.idactiviteit FROM deleted;
    IF (SELECT datum FROM Activiteiten WHERE id = @activiteitID) <= GETDATE()
        RETURN;

    SELECT @vrijPlaatsen = dbo.udfInschrijvingAantal(@activiteitID, 1);

    DECLARE cWachtrij CURSOR FOR
        SELECT id, deelnemers FROM Inschrijvingen WHERE idactiviteit = @activiteitID AND status = 'W'
    ORDER BY datum;

    END;
GO ik wil dat de gegevens van de sql vertoond worden boven de voornaam bij persoonbewerken en

```

als ik op een naam klik dat ik de gegevens kan aanpassen binnen in het tabel en als ik gegevens toevoeg dat ze worden behouden in de tabel

Bedankt voor de uitgebreide data en SQL-schema. Voor je WPF-applicatie wil je dat de **Bewerk Persoon** **Page** pas wordt geladen en weergegeven wanneer je op een specifieke button ("Persoon") klikt. Dit is een typische navigatievraag.

Ik neem aan dat je een `Frame` in je `MainWindow` hebt waar de pagina's worden geladen, bijvoorbeeld een `Frame` genaamd `MainFrame`.

Simpele WPF-navigatie setup

1. Je hebt een `Button` in je `MainWindow.xaml` (of in de pagina waar je lijst met personen staat):

```
xml

<Button Content="Persoon bewerken" Click="PersoonBewerken_Click" />
```

2. In de code-behind (`MainWindow.xaml.cs` of de juiste `.cs`):

```
csharp

private void PersoonBewerken_Click(object sender, RoutedEventArgs e)
{
    // Navigeer naar de Bewerken pagina
    MainFrame.Navigate(new PersoonBewerkPage());
}
```

Wat als je van de PersonenOverzichtPage naar Bewerken wilt navigeren?

Stel je hebt in `PersonenOverzichtPage.xaml` een lijst en een knop "Bewerken". Je wil dat wanneer je die knop klikt, de `MainWindow` zijn Frame naar de bewerk pagina navigeert met de juiste persoon.

Een goede aanpak:

- Maak een event of gebruik een callback om aan `MainWindow` te melden dat er genavigeerd moet worden.

Voorbeeld via event:

In `PersonenOverzichtPage.xaml.cs`:

```
csharp

public event Action<int> BewerkenRequested; // int = id van de persoon

private void ButtonBewerken_Click(object sender, RoutedEventArgs e)
{
    if (SelectedPersoonId != null)
    {
        BewerkenRequested?.Invoke(SelectedPersoonId);
    }
}
```

In `MainWindow.xaml.cs` waar je de `PersonenOverzichtPage` laadt:

csharp

```
var personenPage = new PersonenOverzichtPage();
personenPage.BewerkenRequested += (id) =>
{
    // Navigeer met id door naar bewerk pagina
    MainFrame.Navigate(new PersoonBewerkPage(id));
};
MainFrame.Navigate(personenPage);
```

In `PersoonBewerkPage`` voeg je een constructor toe die het id accepteert:

csharp

```
public PersoonBewerkPage(int persoonId)
{
    InitializeComponent();
    LoadPersoonData(persoonId);
}

private void LoadPersoonData(int id)
{
    // haal persoon uit DB en vul velden in
}
```

Samenvatting

- Gebruik een `Frame`` in je `MainWindow`` om pagina's te tonen.
- Navigeer naar de Bewerken pagina pas na knopklik met `MainFrame.Navigate(...)``.
- Gebruik events of methoden om de `id`` van de persoon door te geven.
- Laad de gegevens van die persoon in de bewerk pagina op basis van dat id.

Heb je al een `Frame`` en pagina's opgezet? Kan ik je helpen met een volledig voorbeeld van `MainWindow.xaml`` + `MainWindow.xaml.cs`` + de pagina's?

Of wil je de code om personen te laden in een lijst en daaruit navigeren?



geef me de gehele code

Hier is de volledige, correcte SQL-code die je kunt gebruiken voor je database, inclusief het schema, functies, views, procedures en voorbeelddata.

sql

```
-- Database aanmaken en gebruiken
CREATE DATABASE EP3_q1720602;
GO
USE EP3_q1720602;
GO

-- Tabel ActiviteitTypes
CREATE TABLE ActiviteitTypes (
    id INT PRIMARY KEY IDENTITY,
    type VARCHAR(255) NOT NULL UNIQUE
);
```



```

GO

-- Tabel Personen
CREATE TABLE Personen (
    id INT PRIMARY KEY IDENTITY,
    voornaam VARCHAR(255) NULL,
    familienaam VARCHAR(255) NOT NULL,
    islid BIT NOT NULL DEFAULT 0
);
GO

-- Tabel Activiteiten
CREATE TABLE Activiteiten (
    id INT PRIMARY KEY IDENTITY,
    naam VARCHAR(255) NOT NULL,
    idtype INT NOT NULL,
    prijs DECIMAL(10, 2) NULL CHECK (prijs > 0),
    prijslid DECIMAL(10, 2) NULL CHECK (prijslid < prijs OR prijslid IS NULL),
    datum DATE NOT NULL,
    maxdeelnemers INT NULL CHECK (maxdeelnemers >= 10 OR maxdeelnemers IS NULL),
    FOREIGN KEY (idtype) REFERENCES ActiviteitTypes(id),
    CHECK (datum > GETDATE())
);
GO

-- Tabel Inschrijvingen
CREATE TABLE Inschrijvingen (
    id INT PRIMARY KEY IDENTITY,
    idpersoon INT NOT NULL,
    idactiviteit INT NOT NULL,
    deelnemers INT NOT NULL CHECK (deelnemers > 0),
    datum DATE NOT NULL DEFAULT GETDATE(),
    status CHAR(1) NOT NULL CHECK (status IN ('I', 'W')),
    FOREIGN KEY (idpersoon) REFERENCES Personen(id),
    FOREIGN KEY (idactiviteit) REFERENCES Activiteiten(id)
);
GO

-- Mockdata toevoegen aan Personen
INSERT INTO Personen (voornaam, familienaam, islid)
VALUES
    ('Yasser', 'El Aouati', 1),
    ('Adam', 'Amri', 0),
    ('Mattheo', 'De Bakker', 0);
GO

-- Mockdata toevoegen aan ActiviteitTypes
INSERT INTO ActiviteitTypes (type)
VALUES
    ('Film'),
    ('Wandeling'),
    ('Vergadering'),
    ('Cursus'),
    ('Fietstocht');
GO

-- Functie udfTeBetalen
CREATE FUNCTION udfTeBetalen (@inschrijvingID INT)
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @prijs DECIMAL(10, 2);
    DECLARE @prijslid DECIMAL(10, 2);
    DECLARE @deelnemers INT;
    DECLARE @islid BIT;

    SELECT @prijs = A.prijs,
           @prijslid = A.prijslid,
           @deelnemers = I.deelnemers,
           @islid = P.islid
    FROM Inschrijvingen I
    JOIN Activiteiten A ON I.idactiviteit = A.id
    JOIN Personen P ON I.idpersoon = P.id
    WHERE I.id = @inschrijvingID;

```

```

IF @prijs IS NULL
    RETURN 0;
IF @inschrijvingID IS NULL
    RETURN 1;
IF @islid = 1
    RETURN @prijslid * @deelnemers;
ELSE
    RETURN @prijs * @deelnemers;
END;
GO

-- Functie udfInschrijvingAantal
CREATE FUNCTION udfInschrijvingAantal (@activiteitID INT, @type INT)
RETURNS INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Activiteiten WHERE id = @activiteitID)
        RETURN -1;

    IF @type = 0
        RETURN (SELECT ISNULL(SUM(deelnemers), 0) FROM Inschrijvingen WHERE idactiviteit = @activiteitID);
    ELSE IF @type = 1
        RETURN (SELECT maxdeelnemers - ISNULL(SUM(deelnemers), 0)
                FROM Activiteiten A
                LEFT JOIN Inschrijvingen I ON A.id = I.idactiviteit AND I.status = 'I'
                WHERE A.id = @activiteitID);
    RETURN NULL;
END;
GO

-- View vewTeBetalen
CREATE VIEW vewTeBetalen AS
SELECT I.id,
       A.naam,
       P.familienaam,
       dbo.udfTeBetalen(I.id) AS teBetalen
FROM Inschrijvingen I
JOIN Activiteiten A ON I.idactiviteit = A.id
JOIN Personen P ON I.idpersoon = P.id;
GO

-- Procedure sprActiviteitToevoegen
CREATE PROCEDURE sprActiviteitToevoegen
    @naam VARCHAR(255),
    @idtype INT,
    @datum DATE,
    @maxdeelnemers INT = NULL,
    @prijs DECIMAL(10, 2) = NULL,
    @prijslid DECIMAL(10, 2) = NULL
AS
BEGIN
    IF @naam IS NULL OR LEN(@naam) = 0
    BEGIN
        RAISERROR('Naam is verplicht.', 16, 1);
        RETURN 1;
    END

    IF NOT EXISTS (SELECT 1 FROM ActiviteitTypes WHERE id = @idtype)
    BEGIN
        RAISERROR('Ongeldig activiteitstype.', 16, 1);
        RETURN 2;
    END

    IF @datum <= CAST(GETDATE() AS DATE)
    BEGIN
        RAISERROR('Datum moet in de toekomst liggen.', 16, 1);
        RETURN 3;
    END

    IF @maxdeelnemers IS NOT NULL AND @maxdeelnemers < 10
    BEGIN
        RAISERROR('maxdeelnemers moet minstens 10 zijn.', 16, 1);
        RETURN 4;
    END
END

```

```

IF @prijs IS NOT NULL AND @prijs < 1
BEGIN
    RAISERROR('Prijs moet minstens 1 zijn.', 16, 1);
    RETURN 5;
END

IF @prijslid IS NOT NULL AND (@prijs IS NULL OR @prijslid >= @prijs)
BEGIN
    RAISERROR('Prijslid moet lager zijn dan prijs.', 16, 1);
    RETURN 6;
END

INSERT INTO Activiteiten (naam, idtype, datum, maxdeelnemers, prijs, prijslid)
VALUES (@naam, @idtype, @datum, @maxdeelnemers, @prijs, @prijslid);

RETURN 0;
END;
GO

-- Procedure sprInschrijvingToevoegen
CREATE PROCEDURE sprInschrijvingToevoegen
    @idpersoon INT,
    @idactiviteit INT,
    @deelnemers INT
AS
BEGIN
    DECLARE @maxdeelnemers INT;
    DECLARE @beschikbareplaatsen INT;

    SELECT @maxdeelnemers = maxdeelnemers FROM Activiteiten WHERE id = @idactiviteit;
    SET @beschikbareplaatsen = dbo.udfInschrijvingAantal(@idactiviteit, 1);

    IF @maxdeelnemers IS NOT NULL AND (@beschikbareplaatsen IS NULL OR @beschikbareplaatsen < @deelnemers)
    BEGIN
        RAISERROR('Niet genoeg beschikbare plaatsen.', 16, 1);
        RETURN 1;
    END

    INSERT INTO Inschrijvingen (idpersoon, idactiviteit, deelnemers, status)
    VALUES (@idpersoon, @idactiviteit, @deelnemers, 'I');

    RETURN 0;
END;
GO

```

Als je deze code uitvoert, krijg je:

- Een database `EP3_q1720602`.
- Tabellen `ActiviteitTypes`



ik wil geen sql maar enkel de gegevens van de sql in de app sql server management studio die ik heb aangegeven dat die zichtbaar zijn en ook kan aanpassen

Ah, je wil dus **geen SQL-code** maar gewoon een overzicht van de gegevens die in jouw database zitten, zodat je die makkelijk kunt bekijken en aanpassen in SQL Server Management Studio (SSMS).

Ik help je hiermee op een duidelijke manier:

Hoe de gegevens in SSMS zichtbaar en bewerkbaar maken?

1. Tabellen openen en data bekijken

- Open **SQL Server Management Studio (SSMS)**.
- Verbind met je database-server.
- Vouw in Object Explorer je database uit, bijvoorbeeld ``EP3_q1720602``.
- Vouw de map **Tables** uit.
- Zoek de tabel die je wil bekijken, bv. ``Personen``, ``Activiteiten``, ``ActiviteitTypes``, of ``Inschrijvingen``.
- Klik met de rechtermuisknop op de tabel en kies **"Edit Top 200 Rows"** om de data te zien en te bewerken.
- Of kies **"Select Top 1000 Rows"** om de data te bekijken (niet direct bewerkbaar).

2. Gegevens aanpassen

- In het venster **Edit Top 200 Rows** kan je rijen wijzigen, nieuwe rijen toevoegen of verwijderen.
- Je wijzigingen worden direct toegepast op de database zodra je uit de rij klikt of de rij opslaat.

Overzicht van de tabellen en hun belangrijkste kolommen (zoals jij ze hebt):

Tabel	Kolommen (voorbeeld)	Wat het bevat
<code>`Personen`</code>	<code>`id`</code> , <code>`voornaam`</code> , <code>`familienaam`</code> , <code>`islid`</code>	Personen met lid-status
<code>`ActiviteitTypes`</code>	<code>`id`</code> , <code>`type`</code>	Soorten activiteiten (Film, Wandeling, ...)
<code>`Activiteiten`</code>	<code>`id`</code> , <code>`naam`</code> , <code>`idtype`</code> , <code>`prijs`</code> , <code>`prijslid`</code> , <code>`datum`</code> , <code>`maxdeelnemers`</code>	Activiteiten met data en prijzen
<code>`Inschrijvingen`</code>	<code>`id`</code> , <code>`idpersoon`</code> , <code>`idactiviteit`</code> , <code>`deelnemers`</code> , <code>`datum`</code> , <code>`status`</code>	Wie is ingeschreven en hoeveel deelnemers

Tips om gegevens te bewerken en zichtbaar te houden:

- Gebruik **"Edit Top 200 Rows"** voor snelle bewerking.
- Voor meer geavanceerd beheer kan je zelf queries schrijven, bijvoorbeeld:

```
sql
SELECT * FROM Personen;
```

of

```
sql
UPDATE Personen SET voornaam = 'Jan' WHERE id = 1;
```

Wil je een overzicht van de gegevens die er nu in staan?

Dan kan ik ze ook voor je ophalen als voorbeeld:

Voorbeeld van een query die alle personen toont:

```
sql
SELECT * FROM Personen;
```

Voorbeeld van een query die alle activiteiten toont:

```
sql
SELECT * FROM Activiteiten;
```

Je kan deze queries uitvoeren in SSMS en zo je data bekijken.

Wil je dat ik jou een overzicht geef van de huidige data in jouw tabellen (zoals voorbeelddata uit je database)?

Laat het me weten, dan kan ik dat opvragen met een paar voorbeeld SELECT-statements.

Of wil je hulp met het aanpassen van data in SSMS?

Kort gezegd:

In SSMS open je gewoon je database, klik met rechts op een tabel en kies "Edit Top 200 Rows" om makkelijk gegevens te zien en te veranderen.

Als je wilt, kan ik ook stap-voor-stap uitleg geven met screenshots of een korte handleiding maken. Wat wil je?



geef me de code die dit kan doen in C# zodat ik dit kan toevoegen

Ah, je wil in C# een kleine applicatie maken waarmee je de gegevens uit jouw SQL Server database kan ophalen én aanpassen, zonder zelf SQL-queries in SSMS te moeten typen?

Dus een soort **eenvoudige CRUD-interface** in C# die met jouw bestaande SQL Server database connect en data toont + wijzigt?

Ik geef je een voorbeeld van zo'n minimalistische C# WinForms app (je kan ook WPF gebruiken, maar WinForms is iets eenvoudiger om te starten) die:

- Verbindt met je SQL Server database
- Data uit een tabel ophaalt en toont in een DataGridView
- Wijzigingen kan doorvoeren (update, insert, delete)

C# WinForms voorbeeld met SqlDataAdapter en DataGridView

csharp

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace SimpleDataEditor
{
    public partial class MainForm : Form
    {
        private SqlConnection connection;
        private SqlDataAdapter dataAdapter;
        private DataTable dataTable;
        private string connectionString = @"Server=.\SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=

        public MainForm()
        {
            InitializeComponent();

            // DataGridView toevoegen aan formulier via designer of code
            DataGridView dgv = new DataGridView
            {
                Dock = DockStyle.Fill,
                AutoGenerateColumns = true
            };
            this.Controls.Add(dgv);

            // Verbind en laad data
            connection = new SqlConnection(connectionString);
            LoadData(dgv);
        }

        private void LoadData(DataGridView dgv)
        {
            try
            {
                connection.Open();

                // Vervang hier 'Personen' door de tabel die je wil bekijken/wijzigen
                dataAdapter = new SqlDataAdapter("SELECT * FROM Personen", connection);

                // SqlCommandBuilder genereert automatisch Insert/Update/Delete commands op basis van de select query
                SqlCommandBuilder commandBuilder = new SqlCommandBuilder(dataAdapter);

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

                dgv.DataSource = dataTable;
            }
            catch (Exception ex)
            {
                MessageBox.Show("Fout bij laden data: " + ex.Message);
            }
            finally
            {
                connection.Close();
            }
        }

        // Sla wijzigingen in database op
        private void SaveChanges()
        {
            try
            {
                connection.Open();
                dataAdapter.Update(dataTable);
                MessageBox.Show("Wijzigingen opgeslagen!");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Fout bij opslaan: " + ex.Message);
            }
            finally
            {
                connection.Close();
            }
        }
    }
}
```

```

        {
            connection.Close();
        }
    }

    protected override void OnFormClosing(FormClosingEventArgs e)
    {
        // Vraag bij sluiten of je wijzigingen wilt opslaan
        if (dataTable != null && dataTable.GetChanges() != null)
        {
            var result = MessageBox.Show("Wil je wijzigingen opslaan?", "Opslaan", MessageBoxButtons.YesNo);
            if (result == DialogResult.Yes)
            {
                SaveChanges();
            }
            else if (result == DialogResult.Cancel)
            {
                e.Cancel = true; // voorkom sluiten
            }
        }
        base.OnFormClosing(e);
    }
}
}
}

```

Wat doet deze code?

- Verbindt met jouw database (`EP3_q1720602`) via connection string — pas die aan naar jouw server en database.
- Laadt alle data van tabel `Personen` in een `DataTable`.
- Bindt die aan een `DataGridView` die je kan bewerken (toevoegen, verwijderen, wijzigen).
- Met `SqlCommandBuilder` worden automatisch de Insert, Update en Delete commando's gegenereerd.
- Bij het sluiten van het formulier vraagt hij of je de wijzigingen wil opslaan, en voert `dataAdapter.Update()` uit om de aanpassingen naar de database door te voeren.

Wat moet je zelf nog doen?

- Vervang in de query `SELECT * FROM Personen` de tabelnaam door de tabel die je wilt beheren (`Activiteiten`, `Inschrijvingen`, etc.).
- Pas de connection string aan zodat hij jouw SQL Server instantie aanspreekt.
- Maak een nieuw WinForms-project aan in Visual Studio.
- Voeg een nieuw Form toe en plak bovenstaande code in de code-behind (vervang de standaard `Form1` code).
- Voeg een DataGridView toe in de designer of via code (zoals hierboven gedaan).
- Build en run de app.

Wil je dat ik ook een kleine demo WinForms projectstructuur genereer met meerdere tabbladen om alle tabellen te beheren?

Of liever WPF?

Als je dit simpel wil houden is dit een solide start. Je hebt direct een werkbare editor zonder SQL te schrijven, gewoon door je data in het raster te wijzigen.

Laat maar weten!



komen de gegevens hierdoor te zien bij de page PersoonBewerken.xaml

Ah, je vraagt of de data die je via die C#-code (met SqlDataAdapter/DataGridView) ophaalt ook automatisch **zichtbaar wordt in jouw WPF Page `PersoonBewerken.xaml`**?

Dat is niet automatisch zo.

Belangrijk:

De WinForms-voorbeeldcode die ik gaf is een complete kleine standalone app, die een eigen formulier (Form) heeft met een DataGridView.

In een WPF-project met meerdere pages zoals `PersoonBewerken.xaml` werkt dat anders:

- WPF gebruikt `DataGrid` of andere controls.
- Data binding in WPF gebeurt via ViewModels of code-behind.
- Je moet de data expliciet ophalen in de code-behind van `PersoonBewerken.xaml` en binden aan de controls op die pagina.

Wat moet je doen om data zichtbaar te krijgen in `PersoonBewerken.xaml`?

1. Maak in `PersoonBewerken.xaml` een DataGrid of andere controls aan om data te tonen.

Bijv:

```
xml

<DataGrid x:Name="dgPersonen" AutoGenerateColumns="True" />
```

2. Haal in de code-behind (PersoonBewerken.xaml.cs) de data op uit je database (via ADO.NET of Entity Framework).

Bijv:

```
csharp

public partial class PersoonBewerken : Page
{
    private string connectionString = @"Server=. \SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=
    private DataTable personenTable;

    public PersoonBewerken()
    {
        InitializeComponent();
        LoadPersonen();
    }
}
```



```

    }

    private void LoadPersonen()
    {
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Personen", conn);
            personenTable = new DataTable();
            adapter.Fill(personenTable);
            dgPersonen.ItemsSource = personenTable.DefaultView;
        }
    }

    // Later kun je ook updates doorvoeren
}

```

3. Optioneel: voeg Save/Update functionaliteit toe met SqlCommandBuilder en dataAdapter.Update zoals in het voorbeeld.

Samengevat:

- Je kunt die WinForms-code niet één-op-één gebruiken in een WPF Page.
- Maar het concept is hetzelfde: je maakt een connection, haalt data op, zet die data in een DataTable of List<T>, en bindt die aan een DataGrid.
- Voor WPF doe je dit in de Page zelf, en je gebruikt `DataGrid` ipv `DataGridView`.

Wil je dat ik een volledige voorbeeldcode voor `PersoonBewerken.xaml` + code-behind schrijf die data ophaalt en toont, en evt. opslaat?

Ik help je graag daarmee!



Severity	Code	Description	Project	File	Line	Suppression State
Error	CS0246	The type or namespace name 'SqlDataAdapter' could not be found (are you missing a using directive or an assembly reference?)	WpfAdmin	C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs	60	Active
Error	CS0121	The call is ambiguous between the following methods or properties: 'PersoonBewerken.PersoonBewerken()' and 'PersoonBewerken.PersoonBewerken()'	WpfAdmin	C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-2425\SlnProject\WpfAdmin\MainWindow.xaml.cs	26	Active
Error	CS0103	The name 'dgPersonen' does not exist in the current context	WpfAdmin	C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs	63	Active
Error	CS0246	The type or namespace name 'SqlConnection' could not be found (are you missing a using directive or an assembly reference?)	WpfAdmin	C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs	58	Active
Error	CS0246	The type or namespace name 'SqlConnection' could not be found (are you missing a using directive or an assembly reference?)	WpfAdmin	C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs	58	Active
Error	CS0246	The type or namespace name 'SqlDataAdapter' could not be found (are you				

```
missing a using directive or an assembly reference?) WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 60 Active
Error CS0111 Type 'PersoonBewerken' already defines a member called 'PersoonBewerken' with
the same parameter types WpfAdmin C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 50 Active
```

Je hebt een aantal compileerfouten die typisch zijn bij het gebruik van SQL-classes in C# WPF. Ik help je stap voor stap dit op te lossen:

1. `SqlConnection`, `SqlDataAdapter` niet gevonden?

Deze klassen zitten in de namespace:

```
csharp
using System.Data.SqlClient;
```

En je moet er ook zeker van zijn dat je project een referentie heeft naar de assembly:

- **System.Data.dll**

In een standaard .NET Framework WPF-project zit die meestal al standaard.

In .NET Core / .NET 5+ moet je mogelijk een NuGet-package toevoegen:

- Package: **System.Data.SqlClient**

Je kan dit doen via NuGet Package Manager in Visual Studio:

```
mathematica
Install-Package System.Data.SqlClient
```

2. `dgPersonen` bestaat niet in de context

Dit betekent dat je in je XAML-bestand van `PersoonBewerken.xaml` nog geen DataGrid hebt met `x:Name="dgPersonen"`.

Zorg dat je in je XAML iets hebt zoals:

```
xml
<Page x:Class="JouwNamespace.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="PersoonBewerken">

    <Grid>
        <DataGrid x:Name="dgPersonen" AutoGenerateColumns="True" />
    </Grid>

</Page>
```

3. Dubbele constructor `PersoonBewerken()`

Je hebt in je `PersoonBewerken.xaml.cs` blijkbaar 2 constructors met exact dezelfde signatuur. Verwijder de dubbele constructor zodat er slechts één overblijft:

```
csharp

public partial class PersoonBewerken : Page
{
    public PersoonBewerken()
    {
        InitializeComponent();
        LoadPersonen();
    }

    private void LoadPersonen()
    {
        // Code om data te laden
    }
}
```

Samengevat: voorbeeld correcte `PersoonBewerken.xaml` + `.xaml.cs`

PersoonBewerken.xaml

```
xml

<Page x:Class="WpfAdmin.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="PersoonBewerken">

    <Grid Margin="10">
        <DataGrid x:Name="dgPersonen" AutoGenerateColumns="True" />
    </Grid>

</Page>
```

PersoonBewerken.xaml.cs

```
csharp

using System.Data;
using System.Data.SqlClient;
using System.Windows.Controls;

namespace WpfAdmin
{
    public partial class PersoonBewerken : Page
    {
        private string connectionString = @"Server=.\SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=yes;";
        private DataTable personenTable;

        public PersoonBewerken()
        {
            InitializeComponent();
            LoadPersonen();
        }
    }
}
```

```

    }

    private void LoadPersonen()
    {
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Personen", conn);
            personenTable = new DataTable();
            adapter.Fill(personenTable);
            dgPersonen.ItemsSource = personenTable.DefaultView;
        }
    }
}

```

Nog een laatste check

- Voeg bovenin je `.xaml.cs` deze using directives toe:

```

csharp

using System.Data;
using System.Data.SqlClient;
using System.Windows.Controls;

```

- Zorg dat je project een referentie naar **System.Data.dll** heeft (standaard bij .NET Framework).
- Bij .NET Core/5/6/7: installeer `System.Data.SqlClient` via NuGet.

Als je dit toepast, moet je zonder deze fouten kunnen bouwen en zie je de data uit je database in de `DataGrid` van `PersoonBewerken.xaml`.

Wil je dat ik je help met de exacte stappen in Visual Studio om de NuGet-package toe te voegen?



```

using System.Data;
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
        {
            Person persoon = new Person
            {
                Voornaam = VoornaamBox.Text,
            }
        }
    }
}

```

```

        Achternaam = AchternaamBox.Text,
        Email = EmailBox.Text,
        Wachtwoord = WachtwoordBox.Password,
        Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
    };

    MessageBox.Show("Persoon opgeslagen:\n" +
        $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

    // Ga terug naar vorige pagina (of maak hier een overzichtspagina)
    if (this.NavigationService.CanGoBack)
    {
        this.NavigationService.GoBack();
    }
}

// Dummy Person class als voorbeeld, vervang door je eigen model / database
public class Person
{
    public string Voornaam { get; set; }
    public string Achternaam { get; set; }
    public string Email { get; set; }
    public string Wachtwoord { get; set; }
    public string Rol { get; set; }
}

public partial class PersoonBewerken : Page
{
    private string connectionString =
@"Server=.\SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=True;";
    private DataTable personenTable;

    public PersoonBewerken()
    {
        InitializeComponent();
        LoadPersonen();
    }

    private void LoadPersonen()
    {
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Personen", conn);
            personenTable = new DataTable();
            adapter.Fill(personenTable);
            dgPersonen.ItemsSource = personenTable.DefaultView;
        }
    }

    // Later kun je ook updates doorvoeren
}

```

```

}
Severity Code    Description Project File Line    Suppression State
Error    CS0246 The type or namespace name 'SqlDataAdapter' could not be found (are you
missing a using directive or an assembly reference?) WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs    60    Active
Error    CS0121 The call is ambiguous between the following methods or properties:
'PersoonBewerken.PersoonBewerken()' and 'PersoonBewerken.PersoonBewerken()' WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\MainWindow.xaml.cs    26    Active
Error    CS0103 The name 'dgPersonen' does not exist in the current context WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs    63    Active
Error    CS0246 The type or namespace name 'SqlConnection' could not be found (are you missing
a using directive or an assembly reference?) WpfAdmin C:\Users\elaou\Desktop\Github\OOAD-
YasserElAouati-2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 58    Active
Error    CS0246 The type or namespace name 'SqlConnection' could not be found (are you missing
a using directive or an assembly reference?) WpfAdmin C:\Users\elaou\Desktop\Github\OOAD-
YasserElAouati-2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs 58    Active
Error    CS0246 The type or namespace name 'SqlDataAdapter' could not be found (are you
missing a using directive or an assembly reference?) WpfAdmin
C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs    60    Active
Error    CS0111 Type 'PersoonBewerken' already defines a member called 'PersoonBewerken' with
the same parameter types WpfAdmin C:\Users\elaou\Desktop\Github\OOAD-YasserElAouati-
2425\SlnProject\WpfAdmin\PersoonBewerken.xaml.cs    50    Active

```

Je hebt een paar dingen door elkaar gehaald in je code.

Ik zal ze puntsgewijs uitleggen en meteen een correcte versie geven.

Problemen in jouw code

1. Dubbele definitie van de klasse `PersoonBewerken`:

Je hebt 2 `partial class PersoonBewerken : Page` in dezelfde file. Dat mag niet.
Combineer alles in één class.

2. Gebruik van SQL-klassen (SqlConnection, SqlDataAdapter):

Je mist de juiste `using` en waarschijnlijk ook de NuGet-package.

3. `dgPersonen` bestaat niet in de context:

Waarschijnlijk heb je in je `PersoonBewerken.xaml` geen `DataGrid` met `x:Name="dgPersonen"`.

Stappen om dit te fixen

Stap 1: Installeer NuGet package

Open in Visual Studio **Package Manager Console** en voer uit:

```
powershell
```

```
Install-Package System.Data.SqlClient
```

Step 2: Corrigeer using directives in PersoonBewerken.xaml.cs

Voeg toe bovenaan:

```
csharp
```

```
using System.Data;
using System.Data.SqlClient;
using System.Windows;
using System.Windows.Controls;
```

Step 3: Corrigeer je XAML

Zorg dat je in `PersoonBewerken.xaml` een DataGrid hebt:

```
xml
```

```
<Page x:Class="WpfAdmin.Pages.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="PersoonBewerken">

    <Grid Margin="10">
        <DataGrid x:Name="dgPersonen" AutoGenerateColumns="True" />
    </Grid>

</Page>
```

Step 4: Combineer je `PersoonBewerken.xaml.cs` code in één class

```
csharp
```

```
using System.Data;
using System.Data.SqlClient;
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        private string connectionString = @"Server=.\SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=yes;";
        private DataTable personenTable;

        public PersoonBewerken()
        {
            InitializeComponent();
            LoadPersonen();
        }

        private void LoadPersonen()
        {
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
```

```

        SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Personen", conn);
        personenTable = new DataTable();
        adapter.Fill(personenTable);
        dgPersonen.ItemsSource = personenTable.DefaultView;
    }
}

private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
{
    // Voorbeeld: je kunt hier updates toevoegen als je wilt
    MessageBox.Show("Opslaan functionaliteit nog implementeren.");
}

// Eventueel je Person-class ergens anders definiëren, niet dubbel in dezelfde file
}

```

Samengevat

- Verwijder dubbele classdefinities.
- Voeg `using System.Data.SqlClient;` toe.
- Installeer de `System.Data.SqlClient` NuGet package.
- Zorg dat je `DataGrid` in XAML staat met naam `dgPersonen`.
- Zorg dat je slechts één constructor hebt in je klasse.

Wil je dat ik je ook help met een update-functie voor opslaan of andere functionaliteiten?



er is niets zichtbaar geen gegevens

Oké, als je DataGrid leeg blijft, kunnen we stap voor stap checken wat er misloopt.

Hier is een checklist en verbeterde aanpak om te zorgen dat je data wél verschijnt in je `dgPersonen`:

1. Controleer de verbinding met je database

- Staat je SQL Server aan?
- Is de connection string correct?
Voor een lokale SQL Server Express is dit meestal:

csharp

```
private string connectionString = @"Server=.\SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=Tru
```

Je kunt ook testen met `SqlConnection` alleen:

csharp

```
using (SqlConnection conn = new SqlConnection(connectionString))
{
    conn.Open();
}
```



```

    MessageBox.Show("Verbinding OK!");
}

```

Plaats dit tijdelijk in je constructor vóór `LoadPersonen()`:

Als er een foutmelding komt, dan is er een verbindingsprobleem.

2. Controleer of er data in de tabel zit

- Open SQL Server Management Studio (SSMS).
- Voer uit:

```

sql

SELECT * FROM Personen;

```

- Zijn er rijen in deze tabel?

Als nee, dan zie je natuurlijk niks in de DataGridView.

3. Voeg error handling toe om fouten te zien

Pas je `LoadPersonen` zo aan:

```

csharp

private void LoadPersonen()
{
    try
    {
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Personen", conn);
            personenTable = new DataTable();
            adapter.Fill(personenTable);
            dgPersonen.ItemsSource = personenTable.DefaultView;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Fout bij laden data:\n" + ex.Message);
    }
}

```

4. Controleer dat `dgPersonen` in XAML staat met juiste naam

In je `PersoonBewerken.xaml`:

```

xml

<DataGrid x:Name="dgPersonen" AutoGenerateColumns="True" />

```

Zonder deze `x:Name` werkt het binden niet.

5. Voorbeeld volledige werkende code (zonder andere methodes)

PersoonBewerken.xaml:

```
xml

<Page x:Class="WpfAdmin.Pages.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="PersoonBewerken">
    <Grid Margin="10">
        <DataGrid x:Name="dgPersonen" AutoGenerateColumns="True" />
    </Grid>
</Page>
```

PersoonBewerken.xaml.cs:

```
csharp

using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        private string connectionString = @"Server=.\SQLEXPRESS;Database=EP3_q1720602;Trusted_Connection=True;";
        private DataTable personenTable;

        public PersoonBewerken()
        {
            InitializeComponent();

            try
            {
                using (SqlConnection conn = new SqlConnection(connectionString))
                {
                    conn.Open();
                    MessageBox.Show("Verbinding OK!");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Verbindingsfout:\n" + ex.Message);
                return;
            }

            LoadPersonen();
        }

        private void LoadPersonen()
        {
            try
            {
                using (SqlConnection conn = new SqlConnection(connectionString))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Personen", conn);
                    personenTable = new DataTable();
                    adapter.Fill(personenTable);
                    dgPersonen.ItemsSource = personenTable.DefaultView;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Fout bij laden data:\n" + ex.Message);
            }
        }
    }
}
```

6. Herbuild en run opnieuw

- Herbuild je project.
- Start de app.
- Als je `"Verbinding OK!"` ziet, is verbinding met DB gelukt.
- Dan moet je in je DataGridView data zien als er iets in je tabel staat.

Extra tip

Als je nog steeds niks ziet, voeg in `LoadPersonen` na `adapter.Fill` een check:

```
csharp  
  
MessageBox.Show($"Aantal rijen geladen: {personenTable.Rows.Count}");
```

Zo weet je of er data wordt opgehaald.

Als je dit doorloopt en nog problemen hebt, stuur dan:

- De exacte connection string
- Foutmeldingen die je krijgt
- De inhoud van je tabel `Personen` in SSMS (eventueel een screenshot of `SELECT * FROM Personen` resultaten)

Dan help ik je verder!