

# Expand SQL Command



```
SqlCommand comm = new SqlCommand("DELETE FROM Deelname WHERE persoon_id = @parID",
conn);
```

Expand this command.

Where persoon\_id = parID AND where deelname.activiteit\_id in select Activiteit WHERE organisator\_id = @parID



Sure, I can help you expand the SQL command to include both conditions. Here is the expanded SQL command:

csharp

```
SqlCommand comm = new SqlCommand(
    "DELETE FROM Deelname " +
    "WHERE persoon_id = @parID " +
    "AND activiteit_id IN (SELECT activiteit_id FROM Activiteit WHERE organisator_id = @parID)",
    conn
);
```

In this command:

- The first condition `persoon\_id = @parID` ensures that the `persoon\_id` matches the parameter `@parID`.
- The second condition `activiteit\_id IN (SELECT activiteit\_id FROM Activiteit WHERE organisator\_id = @parID)` ensures that the `activiteit\_id` is in the list of `activiteit\_id`s for which the `organisator\_id` matches the parameter `@parID`.

Make sure you add the `@parID` parameter to the command before executing it. Here's how you do that:

csharp

```
comm.Parameters.AddWithValue("@parID", parID);
```

Putting it all together, your code should look something like this:

csharp

```
using (SqlCommand comm = new SqlCommand(
    "DELETE FROM Deelname " +
    "WHERE persoon_id = @parID " +
```

```

"AND activiteit_id IN (SELECT activiteit_id FROM Activiteit WHERE organisator_id =
@parID)",
conn))
{
    comm.Parameters.AddWithValue("@parID", parID);

    // Open the connection if it's not already open
    if (conn.State != ConnectionState.Open)
    {
        conn.Open();
    }

    // Execute the command
    int rowsAffected = comm.ExecuteNonQuery();

    // Optionally, handle the result
    Console.WriteLine($"{rowsAffected} rows deleted.");
}

```

This command ensures that the deletion from `Deelnemers` table only happens for the rows where both conditions are met.



```

<Border Background="PaleGoldenrod"
    Width="250" Height="125"
    BorderThickness="2" BorderBrush="Black">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition Height="0.5*"/>
            <RowDefinition/>
        </Grid.RowDefinitions>

        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="1.2*"/>
                <ColumnDefinition Width="3*"/>
                <ColumnDefinition Width="Auto"/>
            </Grid.ColumnDefinitions>
            <Image x:Name="ImgActiviteitIcon"/>
            <StackPanel Grid.Column="1">
                <Label Name="LblDatum" Padding="0"/>
                <Label Name="LblOrganiser" Padding="0"/>
                <Label Name="LblDeelnemers" Padding="0"/>
            </StackPanel>
            <StackPanel Grid.Column="2">
                <Button Content="📄" Width="25px" Margin="0 0 0 5" Background="Transparent"
BorderBrush="Black" BorderThickness="1"/>
                <Button Content="🗑️" Width="25px" Background="Transparent" BorderBrush="Black"
BorderThickness="1"/>
            </StackPanel>
        </Grid>

        <Label Grid.Row="1" Name="LblTitel"/>

```

```
<TextBlock TextWrapping="Wrap" IsEnabled="False" Grid.Row="2" Name="TxtBeschrijving"/>
</Grid>
</Border>
```

Ik heb dit gemaakt in .xaml

Schrijf me de .cs versie van dit, Ik wil een functie -> private void AddBorder(Activiteit activiteit)

die enkel de border genereert 1:1 met deze .xaml, en toevoegt in WrapActiviteiten.Children



Certainly! Below is the C# code to create a `Border` similar to the one defined in your XAML and a function `AddBorder` that takes an `Activiteit` object and adds the `Border` to `WrapActiviteiten.Children`.

First, let's define the `Activiteit` class if it doesn't exist already:

csharp

```
public class Activiteit
{
    public string IconSource { get; set; }
    public string Datum { get; set; }
    public string Organisator { get; set; }
    public string Deelnemers { get; set; }
    public string Titel { get; set; }
    public string Beschrijving { get; set; }
}
```

Now, let's implement the `AddBorder` function:

csharp

```
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Media.Imaging;

private void AddBorder(Activiteit activiteit)
{
    // Create Border
    Border border = new Border
    {
        Background = Brushes.PaleGoldenrod,
        Width = 250,
        Height = 125,
        BorderThickness = new Thickness(2),
        BorderBrush = Brushes.Black,
    };

    // Create Grid
    Grid grid = new Grid();
    grid.RowDefinitions.Add(new RowDefinition());
    grid.RowDefinitions.Add(new RowDefinition { Height = new GridLength(0.5, GridUnitType.Star) });
    grid.RowDefinitions.Add(new RowDefinition());

    // Create Inner Grid
    Grid innerGrid = new Grid();
    innerGrid.ColumnDefinitions.Add(new ColumnDefinition { Width = new GridLength(1.2,
```

```

GridUnitType.Star) });
    innerGrid.ColumnDefinitions.Add(new ColumnDefinition { Width = new GridLength(3,
GridUnitType.Star) });
    innerGrid.ColumnDefinitions.Add(new ColumnDefinition { Width = GridLength.Auto });

    // Create Image
    Image imgActiviteitIcon = new Image();
    if (!string.IsNullOrEmpty(activiteit.IconSource))
    {
        imgActiviteitIcon.Source = new BitmapImage(new Uri(activiteit.IconSource,
UriKind.RelativeOrAbsolute));
    }
    innerGrid.Children.Add(imgActiviteitIcon);

    // Create StackPanel for Labels
    StackPanel stackPanel1 = new StackPanel();
    Grid.SetColumn(stackPanel1, 1);
    Label lblDatum = new Label { Name = "LblDatum", Padding = new Thickness(0), Content =
activiteit.Datum };
    Label lblOrganiser = new Label { Name = "LblOrganiser", Padding = new Thickness(0),
Content = activiteit.Organiser };
    Label lblDeelnemers = new Label { Name = "LblDeelnemers", Padding = new Thickness(0),
Content = activiteit.Deelnemers };
    stackPanel1.Children.Add(lblDatum);
    stackPanel1.Children.Add(lblOrganiser);
    stackPanel1.Children.Add(lblDeelnemers);
    innerGrid.Children.Add(stackPanel1);

    // Create StackPanel for Buttons
    StackPanel stackPanel2 = new StackPanel();
    Grid.SetColumn(stackPanel2, 2);
    Button btnDetails = new Button { Content = "📄", Width = 25, Margin = new Thickness(0,
0, 0, 5), Background = Brushes.Transparent, BorderBrush = Brushes.Black, BorderThickness =
new Thickness(1) };
    Button btnDelete = new Button { Content = "🗑️", Width = 25, Background =
Brushes.Transparent, BorderBrush = Brushes.Black, BorderThickness = new Thickness(1) };
    stackPanel2.Children.Add(btnDetails);
    stackPanel2.Children.Add(btnDelete);
    innerGrid.Children.Add(stackPanel2);

    // Add Inner Grid to Main Grid
    grid.Children.Add(innerGrid);

    // Create and Add Title Label
    Label lblTitel = new Label { Name = "LblTitel", Content = activiteit.Titel };
    Grid.SetRow(lblTitel, 1);
    grid.Children.Add(lblTitel);

    // Create and Add Description TextBlock
    TextBlock txtBeschrijving = new TextBlock { Name = "TxtBeschrijving", Text =
activiteit.Beschrijving, TextWrapping = TextWrapping.Wrap, IsEnabled = false };
    Grid.SetRow(txtBeschrijving, 2);
    grid.Children.Add(txtBeschrijving);

    // Add Grid to Border
    border.Child = grid;

    // Add Border to WrapActiviteiten
    WrapActiviteiten.Children.Add(border);
}

```

Make sure you have a `WrapPanel` named `WrapActiviteiten` defined in your XAML or code-behind where this `Border` will be added. Here is an example:

xaml

```
<WrapPanel x:Name="WrapActiviteiten"/>
```

Now, you can call the `AddBorder`` function and pass an `Activiteit`` object to add a new border to your `WrapPanel``.



Hallo. ik heb GMap.NET.Core en GMap.NET.WinPresentation geïnstalleerd, maar er is heel weinig documentatie. Kan jij me helpen met het instellen in mijn .xaml? C# WPF project



Natuurlijk! Ik kan je helpen met het instellen van GMap.NET in een WPF-project. GMap.NET is een krachtige bibliotheek voor het weergeven van kaarten in je WPF-applicatie. Hieronder zal ik je door de stappen leiden om GMap.NET in je project te integreren en een basiskaart weer te geven.

## Stap 1: Installeer GMap.NET via NuGet

Zorg ervoor dat je de benodigde pakketten hebt geïnstalleerd. Je hebt de volgende twee pakketten nodig:

- `GMap.NET.Core``
- `GMap.NET.WindowsPresentation``

Je kunt deze pakketten installeren via de NuGet Package Manager Console:

powershell

```
Install-Package GMap.NET.Core
Install-Package GMap.NET.WindowsPresentation
```

## Stap 2: Voeg GMapControl toe aan je XAML

In je `MainWindow.xaml`` (of een andere XAML-bestand waar je de kaart wilt weergeven), voeg je de `GMapControl`` toe. Zorg ervoor dat je de juiste namespace toevoegt voor de GMap.NET controls.

xml

```
<Window x:Class="YourNamespace.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:gmap="clr-namespace:GMap.NET.WindowsPresentation;assembly=GMap.NET.WindowsPresentation"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid>
        <gmap:GMapControl x:Name="MainMap"
                        HorizontalAlignment="Stretch"
                        VerticalAlignment="Stretch"
                        Margin="10"/>
    </Grid>
</Window>
```

## Stap 3: Configureer de kaart in je code-behind

In de `MainWindow.xaml.cs` configureer je de kaart en stel je de initiële instellingen in.

csharp

```
using System.Windows;
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsPresentation;

namespace YourNamespace
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            InitializeMap();
        }

        private void InitializeMap()
        {
            // Basisinstellingen voor de kaart
            MainMap.MapProvider = GMapProviders.GoogleMap;
            MainMap.Position = new PointLatLng(52.379189, 4.899431); // Amsterdam,
            // Nederland
            MainMap.MinZoom = 2;
            MainMap.MaxZoom = 17;
            MainMap.Zoom = 10;

            // Extra instellingen
            MainMap.CanDragMap = true;
            MainMap.DragButton = MouseButton.Left;
            MainMap.ShowCenter = false; // Verberg het kruis in het midden
        }
    }
}
```

## Toelichting

- **MapProvider:** Stel de kaartprovider in. Hier gebruiken we Google Maps, maar GMap.NET ondersteunt ook andere providers zoals Bing Maps, OpenStreetMap, enz.
- **Position:** De startpositie van de kaart, ingesteld op de coördinaten van Amsterdam in dit voorbeeld.
- **MinZoom en MaxZoom:** Stelt de minimale en maximale zoomniveaus in.
- **Zoom:** Stelt het initiële zoomniveau in.
- **CanDragMap:** Hiermee kun je de kaart slepen met de muis.
- **DragButton:** Hiermee kun je instellen welke muisknop wordt gebruikt om de kaart te slepen.
- **ShowCenter:** Hiermee kun je het kruis in het midden van de kaart verbergen.

Nu heb je een basis GMap.NET-kaart geïntegreerd in je WPF-project. Je kunt de instellingen en functionaliteit verder aanpassen aan je eigen behoeften.



```
foreach (var activiteit in activiteiten)
{
```

```
}
```

Voeg voor elke activiteit een punt op het map met `activiteit.Longitude`, `activiteit.Latitude` en `activiteit.Icoon`



Om voor elke `activiteit`` een punt op de kaart toe te voegen met de coördinaten (`activiteit.Longitude``, `activiteit.Latitude``) en een bijbehorend icoon, kun je markers aan de `GMapControl`` toevoegen. Hiervoor moet je enkele stappen volgen:

1. Maak een `GMapMarker`` voor elke activiteit.
2. Stel de positie en het icoon van de marker in.
3. Voeg de marker toe aan de `GMapControl``.

Hier is een voorbeeld van hoe je dit kunt doen. Zorg ervoor dat je `activiteit``-objecten de eigenschappen `Longitude``, `Latitude`` en `Icoon`` bevatten.

## Voorbeeld Activiteit Klasse

csharp

```
public class Activiteit
{
    public double Latitude { get; set; }
    public double Longitude { get; set; }
    public string Icoon { get; set; } // Path naar het icoon
}
```

## Voeg markers toe in je `MainWindow.xaml.cs``

Hier is de aangepaste `MainWindow.xaml.cs`` om markers toe te voegen voor elke activiteit:

csharp

```
using System.Windows;
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsPresentation;
using System.Collections.Generic;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

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

            // Voeg hier je activiteiten toe
            List<Activiteit> activiteiten
```



icoon is een byte[]



Dank je voor de verduidelijking. Als het icoon een byte-array (`byte[]`) is, kunnen we dit omzetten naar een afbeelding die we als marker op de kaart kunnen weergeven.

Hier is een voorbeeld van hoe je dit kunt doen:

### Voorbeeld `Activiteit` Klasse

csharp

```
public class Activiteit
{
    public double Latitude { get; set; }
    public double Longitude { get; set; }
    public byte[] Icoon { get; set; }
}
```

### Voeg markers toe in je `MainWindow.xaml.cs`

Hier is de aangepaste `MainWindow.xaml.cs` om markers toe te voegen voor elke activiteit:

csharp

```
using System.Collections.Generic;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsPresentation;
using System.Windows.Shapes;

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

            // Voeg hier je activiteiten toe
            List<Activiteit> activiteiten = new List<Activiteit>
            {
                new Activiteit { Latitude = 52.379189, Longitude = 4.899431, Icoon =
LoadIcon("path_to_icon1.png") },
                new Activiteit { Latitude = 51.924420, Longitude = 4.477733, Icoon =
LoadIcon("path_to_icon2.png") },
            };
            // Voeg meer activiteiten toe zoals nodig

            // Voeg markers toe voor elke activiteit
            foreach (var activiteit in activiteiten)
            {
                AddMarker(activiteit);
            }
        }
    }
}
```



```

private void InitializeMap()
{
    // Basisinstellingen voor de kaart
    MainMap.MapProvider = GMapProviders.GoogleMap;
    MainMap.Position = new PointLatLng(52.379189, 4.899431); // Amsterdam,
    Nederland
    MainMap.MinZoom = 2;
    MainMap.MaxZoom = 17;
    MainMap.Zoom = 10;

    // Extra instellingen
    MainMap.CanDragMap = true;
    MainMap.DragButton = MouseButton.Left;
    MainMap.ShowCenter = false; // Verberg het kruis in het midden
}

private void AddMarker(Activiteit activiteit)
{
    // Maak een nieuwe marker
    GMapMarker marker = new GMapMarker(new PointLatLng(activiteit.Latitude,
    activiteit.Longitude));

    // Creëer een image element voor het icoon
    Image icon = new Image();
    icon.Width = 24; // Pas de grootte van het icoon aan zoals nodig
    icon.Height = 24;
    icon.Source = ByteArrayToImage(activiteit.Icoon);

    // Voeg de image toe aan de marker
    marker.Shape = icon;

    // Voeg de marker toe aan de kaart
    MainMap.Markers.Add(marker);
}

private BitmapImage ByteArrayToImage(byte[] byteArray)
{
    BitmapImage bitmapImage = new BitmapImage();
    using (MemoryStream ms = new MemoryStream(byteArray))
    {
        bitmapImage.BeginInit();
        bitmapImage.StreamSource = ms;
        bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
        bitmapImage.EndInit();
    }
    return bitmapImage;
}

private byte[] LoadIcon(string path)
{
    // Laad het icoon van een bestandspad naar een byte array
    return File.ReadAllBytes(path);
}
}

```

## Toelichting

1. **InitializeMap:** Deze functie configureert de initiële instellingen van de kaart.
2. **AddMarker:** Deze functie voegt een marker toe aan de kaart voor elke `Activiteit`. Het maakt een `GMapMarker`, zet de positie in, converteert de byte-array van het icoon naar een `BitmapImage`, en stelt dit in als het `Shape` van de marker.
3. **ByteArrayToImage:** Deze functie converteert een `byte[]` naar een `BitmapImage`, wat gebruikt wordt voor het icoon van de marker.
4. **LoadIcon:** Deze helperfunctie leest een icoon van een bestandspad en retourneert dit als een `byte[]`. Dit is optioneel, maar handig als je iconen van schijf wilt laden voor testdoeleinden.

Met deze stappen kun je markers met aangepaste iconen toevoegen aan je kaart in een WPF-project met behulp van GMap.NET.



latitude en longitude zijn decimals in mijn model



Geen probleem! We kunnen de `decimal`-typen voor `Latitude` en `Longitude` in je `Activiteit`-model behouden en omzetten naar `double` wanneer we ze gebruiken om de positie van de markers op de kaart te bepalen. Hier is de bijgewerkte versie van de code:

### Voorbeeld `Activiteit` Klasse

csharp

```
public class Activiteit
{
    public decimal Latitude { get; set; }
    public decimal Longitude { get; set; }
    public byte[] Icoon { get; set; }
}
```

### Voeg markers toe in je `MainWindow.xaml.cs`

Hier is de aangepaste `MainWindow.xaml.cs` om markers toe te voegen voor elke activiteit:

csharp

```
using System.Collections.Generic;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsPresentation;
using System.Windows.Input;

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

            // Voeg hier je activiteiten toe
            List<Activiteit> activiteiten = new List<Activiteit>
            {
                new Activiteit { Latitude = 52.379189m, Longitude = 4.899431m, Icoon =
LoadIcon("path_to_icon1.png") },
                new Activiteit { Latitude = 51.924420m, Longitude = 4.477733m, Icoon =
LoadIcon("path_to_icon2.png") },
            };
            // Voeg meer activiteiten toe zoals nodig
        }
    }
}
```

```

        // Voeg markers toe voor elke activiteit
        foreach (var activiteit in activiteiten)
        {
            AddMarker(activiteit);
        }
    }

    private void InitializeMap()
    {
        // Basisinstellingen voor de kaart
        MainMap.MapProvider = GMapProviders.GoogleMap;
        MainMap.Position = new PointLatLng(52.379189, 4.899431); // Amsterdam,
        Nederland
        MainMap.MinZoom = 2;
        MainMap.MaxZoom = 17;
        MainMap.Zoom = 10;

        // Extra instellingen
        MainMap.CanDragMap = true;
        MainMap.DragButton = MouseButton.Left;
        MainMap.ShowCenter = false; // Verberg het kruis in het midden
    }

    private void AddMarker(Activiteit activiteit)
    {
        // Maak een nieuwe marker
        GMapMarker marker = new GMapMarker(new
        PointLatLng((double)activiteit.Latitude, (double)activiteit.Longitude));

        // Creëer een image element voor het icoon
        Image icon = new Image
        {
            Width = 24, // Pas de grootte van het icoon aan zoals nodig
            Height = 24,
            Source = ByteArrayToImage(activiteit.Icoon)
        };

        // Voeg de image toe aan de marker
        marker.Shape = icon;

        // Voeg de marker toe aan de kaart
        MainMap.Markers.Add(marker);
    }

    private BitmapImage ByteArrayToImage(byte[] byteArray)
    {
        BitmapImage bitmapImage = new BitmapImage();
        using (MemoryStream ms = new MemoryStream(byteArray))
        {
            bitmapImage.BeginInit();
            bitmapImage.StreamSource = ms;
            bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
            bitmapImage.EndInit();
        }
        return bitmapImage;
    }

    private byte[] LoadIcon(string path)
    {
        // Laad het icoon van een bestandspad naar een byte array
        return File.ReadAllBytes(path);
    }
}

```

## Toelichting

1. **Decimal naar Double conversie:** In de `AddMarker`-functie wordt `(double)activiteit.Latitude` en `(double)activiteit.Longitude` gebruikt om de `decimal` waarden naar `double` om te zetten, aangezien de `PointLatLng`-constructor doubles vereist.

2. **Initialisatie en Gebruik van GMapControl:** Dit blijft hetzelfde als voorheen, met de nodige aanpassingen om de markers toe te voegen.
3. **BitmapImage van byte[]:** De `ByteArrayToImage`-functie converteert een `byte[]` naar een `BitmapImage`, wat gebruikt wordt voor het icoon van de marker.
4. **LoadIcon Helperfunctie:** Deze leest een icoon van een bestandspad en retourneert dit als een `byte[]`, handig voor testdoeleinden.

Met deze aanpak kun je eenvoudig markers met aangepaste iconen toevoegen aan je kaart in een WPF-project met GMap.NET, zelfs wanneer de coördinaten als `decimal` worden opgeslagen in je model.



kunnen we ook een click event toevoegen voor de marker? Zodat ik de aangeklikte activiteit kan achterhalen



Zeker, we kunnen een klikgebeurtenis toevoegen aan de marker, zodat je kunt achterhalen welke activiteit is aangeklikt. Om dit te doen, moeten we een `MouseLeftButtonUp`-eventhandler toevoegen aan het `Shape`-element van de marker (de `Image` in dit geval) en de bijbehorende activiteit opslaan in de marker.

Hier is een voorbeeld van hoe je dit kunt doen:

## Voeg een klikgebeurtenis toe in `MainWindow.xaml.cs`

Hier is de bijgewerkte `MainWindow.xaml.cs` om een klikgebeurtenis toe te voegen aan elke marker:

csharp

```
using System.Collections.Generic;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsPresentation;
using System.Windows.Input;

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

            // Voeg hier je activiteiten toe
            List<Activiteit> activiteiten = new List<Activiteit>
            {
                new Activiteit { Latitude = 52.379189m, Longitude = 4.899431m, Icoon =
LoadIcon("path_to_icon1.png") },
                new Activiteit { Latitude = 51.924420m, Longitude = 4.477733m, Icoon =
LoadIcon("path_to_icon2.png") },
            };
            // Voeg meer activiteiten toe zoals nodig
        }
    }
}
```

```

        // Voeg markers toe voor elke activiteit
        foreach (var activiteit in activiteiten)
        {
            AddMarker(activiteit);
        }
    }

    private void InitializeMap()
    {
        // Basisinstellingen voor de kaart
        MainMap.MapProvider = GMapProviders.GoogleMap;
        MainMap.Position = new PointLatLng(52.379189, 4.899431); // Amsterdam,
Nederland
        MainMap.MinZoom = 2;
        MainMap.MaxZoom = 17;
        MainMap.Zoom = 10;

        // Extra instellingen
        MainMap.CanDragMap = true;
        MainMap.DragButton = MouseButton.Left;
        MainMap.ShowCenter = false; // Verberg het kruis in het midden
    }

    private void AddMarker(Activiteit activiteit)
    {
        // Maak een nieuwe marker
        GMapMarker marker = new GMapMarker(new
PointLatLng((double)activiteit.Latitude, (double)activiteit.Longitude));

        // Creëer een image element voor het icoon
        Image icon = new Image
        {
            Width = 24, // Pas de grootte van het icoon aan zoals nodig
            Height = 24,
            Source = ByteArrayToImage(activiteit.Icoon)
        };

        // Voeg klikgebeurtenis toe
        icon.MouseLeftButtonUp += (sender, e) =>
        {
            OnMarkerClick(activiteit);
        };

        // Voeg de image toe aan de marker
        marker.Shape = icon;

        // Voeg de marker toe aan de kaart
        MainMap.Markers.Add(marker);
    }

    private void OnMarkerClick(Activiteit activiteit)
    {
        MessageBox.Show($"Geklikte activiteit: {activiteit.Latitude},
{activiteit.Longitude}");
        // Voeg hier de logica toe om de aangeklikte activiteit te verwerken
    }

    private BitmapImage ByteArrayToImage(byte[] byteArray)
    {
        BitmapImage bitmapImage = new BitmapImage();
        using (MemoryStream ms = new MemoryStream(byteArray))
        {
            bitmapImage.BeginInit();
            bitmapImage.StreamSource = ms;
            bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
            bitmapImage.EndInit();
        }
        return bitmapImage;
    }

    private byte[] LoadIcon(string path)
    {
        // Laad het icoon van een bestandspad naar een byte array
        return File.ReadAllBytes(path);
    }

```

```
}  
}
```

## Toelichting

1. **MouseLeftButtonUp Event:** We voegen een `MouseLeftButtonUp`-eventhandler toe aan het `Image`-element dat als marker wordt gebruikt. In deze eventhandler roepen we de `OnMarkerClick`-methode aan.
2. **OnMarkerClick Methode:** Deze methode wordt aangeroepen wanneer een marker wordt aangeklikt. Hierin kun je de logica implementeren om de aangeklikte activiteit te verwerken. In dit voorbeeld wordt een eenvoudige `MessageBox` getoond met de coördinaten van de aangeklikte activiteit.
3. **AddMarker Methode:** De `AddMarker`-methode voegt de marker toe aan de kaart en koppelt de klikgebeurtenis aan het `Image`-element.

Met deze wijzigingen kun je nu op markers op de kaart klikken en de bijbehorende activiteit achterhalen en verwerken.



vul mijn combobox CboType met mijn Enum ActiviteitSoort



Natuurlijk! Om je `ComboBox` (`CboType`) te vullen met waarden van je `ActiviteitSoort`-enum, kun je de `Enum.GetValues`-methode gebruiken om alle waarden van de enum op te halen en deze toe te voegen aan de `ComboBox`. Hieronder vind je een voorbeeld van hoe je dit kunt doen.

## Stap 1: Definieer de Enum

Definieer je `ActiviteitSoort`-enum ergens in je project:

```
csharp  
  
public enum ActiviteitSoort  
{  
    Sport,  
    Muziek,  
    Kunst,  
    Educatie  
    // Voeg meer soorten toe zoals nodig  
}
```

## Stap 2: Voeg de ComboBox toe aan je XAML

Voeg een `ComboBox` toe aan je `MainWindow.xaml`:

```
xml  
  
<Window x:Class="YourNamespace.MainWindow"  
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800">
<Grid>
<ComboBox x:Name="CboType" Width="200" Height="30" VerticalAlignment="Top"
Margin="10"/>
<gmap:GMapControl x:Name="MainMap" HorizontalAlignment="Stretch"
VerticalAlignment="Stretch" Margin="10" Grid.Row="1"/>
</Grid>
</Window>

```

### Stap 3: Vul de ComboBox met Enum-waarden in de code-behind

In je `MainWindow.xaml.cs` kun je de `ComboBox` vullen met de waarden van de enum in de `MainWindow`-constructor of in een aparte methode die je aanroept tijdens de initialisatie:

csharp

```

using System;
using System.Collections.Generic;
using System.Windows;
using System.Windows.Controls;
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsPresentation;

namespace YourNamespace
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            InitializeMap();
            InitializeComboBox();

            // Voeg hier je activiteiten toe
            List<Activiteit> activiteiten = new List<Activiteit>
            {
                new Activiteit { Latitude = 52.379189m, Longitude = 4.899431m, Icoon =
LoadIcon("path_to_icon1.png") },
                new Activiteit { Latitude = 51.924420m, Longitude = 4.477733m, Icoon =
LoadIcon("path_to_icon2.png") },
            };
            // Voeg meer activiteiten toe zoals nodig

            // Voeg markers toe voor elke activiteit
            foreach (var activiteit in activiteiten)
            {
                AddMarker(activiteit);
            }
        }

        private void InitializeMap()
        {
            // Basisinstellingen voor de kaart
            MainMap.MapProvider = GMapProviders.GoogleMap;
            MainMap.Position = new PointLatLng(52.379189, 4.899431); // Amsterdam,
Nederland

            MainMap.MinZoom = 2;
            MainMap.MaxZoom = 17;
            MainMap.Zoom = 10;

            // Extra instellingen
            MainMap.CanDragMap = true;
            MainMap.DragButton = MouseButton.Left;
            MainMap.ShowCenter = false; // Verberg het kruis in het midden
        }
    }
}

```

```

    }

    private void InitializeComboBox()
    {
        // Vul de ComboBox met de waarden van de enum
        CboType.ItemsSource = Enum.GetValues(typeof(ActiviteitSoort));
        CboType.SelectedIndex = 0; // Optioneel: Stel de standaard geselecteerde index
    }

    private void AddMarker(Activiteit activiteit)
    {
        // Maak een nieuwe marker
        GMapMarker marker = new GMapMarker(new
        PointLatLng((double)activiteit.Latitude, (double)activiteit.Longitude));

        // Creëer een image element voor het icoon
        Image icon = new Image
        {
            Width = 24, // Pas de grootte van het icoon aan zoals nodig
            Height = 24,
            Source = ByteArrayToImage(activiteit.Icoon)
        };

        // Voeg klikgebeurtenis toe
        icon.MouseLeftButtonUp += (sender, e) =>
        {
            OnMarkerClick(activiteit);
        };

        // Voeg de image toe aan de marker
        marker.Shape = icon;

        // Voeg de marker toe aan de kaart
        MainMap.Markers.Add(marker);
    }

    private void OnMarkerClick(Activiteit activiteit)
    {
        MessageBox.Show($"Geklikte activiteit: {activiteit.Latitude},
        {activiteit.Longitude}");
        // Voeg hier de logica toe om de aangeklikte activiteit te verwerken
    }

    private BitmapImage ByteArrayToImage(byte[] byteArray)
    {
        BitmapImage bitmapImage = new BitmapImage();
        using (MemoryStream ms = new MemoryStream(byteArray))
        {
            bitmapImage.BeginInit();
            bitmapImage.StreamSource = ms;
            bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
            bitmapImage.EndInit();
        }
        return bitmapImage;
    }

    private byte[] LoadIcon(string path)
    {
        // Laad het icoon van een bestandspad naar een byte array
        return File.ReadAllBytes(path);
    }

    public class Activiteit
    {
        public decimal Latitude { get; set; }
        public decimal Longitude { get; set; }
        public byte[] Icoon { get; set; }
    }

    public enum ActiviteitSoort
    {
        Sport,
        Muziek,

```



```
Kunst,
Educatie
// Voeg meer soorten toe zoals nodig
}
```

## Toelichting

1. **InitializeComboBox Methode:** Deze methode gebruikt `Enum.GetValues` om de waarden van `ActiviteitSoort` op te halen en stelt deze in als `ItemsSource` van de `ComboBox`. Hierdoor wordt de `ComboBox` gevuld met de enum-waarden.
2. **SelectedIndex:** De `SelectedIndex` wordt ingesteld op 0 om ervoor te zorgen dat de eerste waarde standaard is geselecteerd. Dit is optioneel.

Met deze aanpak wordt je `ComboBox` gevuld met de waarden van de `ActiviteitSoort`-enum, en kun je de geselecteerde waarde gebruiken in je applicatie.



ik ben nu bezig op een andere pagina. geef me een foreach voor CboSoort te vullen met ActiviteitSoort



Natuurlijk, je kunt de `ComboBox` handmatig vullen met de waarden van je `ActiviteitSoort`-enum met een `foreach`-lus. Hier is een voorbeeld van hoe je dit kunt doen in de code-behind van een andere pagina.

## Stap 1: Definieer de Enum (als nog niet gedefinieerd)

Zorg ervoor dat je `ActiviteitSoort`-enum ergens in je project is gedefinieerd:

csharp

```
public enum ActiviteitSoort
{
    Sport,
    Muziek,
    Kunst,
    Educatie
    // Voeg meer soorten toe zoals nodig
}
```

## Stap 2: Voeg de ComboBox toe aan je XAML

Voeg een `ComboBox` toe aan je pagina's XAML-bestand:

xml

```
<Page x:Class="YourNamespace.YourPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008">
```

```
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d"
Title="YourPage">

<Grid>
  <ComboBox x:Name="CboSoort" Width="200" Height="30" VerticalAlignment="Top"
Margin="10"/>
</Grid>
</Page>
```

### Stap 3: Vul de ComboBox in de code-behind

In de code-behind van je pagina (`YourPage.xaml.cs`), kun je de `ComboBox` vullen met de enum-waarden:

csharp

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

namespace YourNamespace
{
    public partial class YourPage : Page
    {
        public YourPage()
        {
            InitializeComponent();
            FillComboBoxWithEnumValues();
        }

        private void FillComboBoxWithEnumValues()
        {
            // Haal de waarden van de enum op
            var enumValues = Enum.GetValues(typeof(ActiviteitSoort));

            // Voeg elke enum waarde toe aan de ComboBox
            foreach (var value in enumValues)
            {
                CboSoort.Items.Add(value);
            }

            // Optioneel: Stel de standaard geselecteerde index in
            CboSoort.SelectedIndex = 0;
        }
    }
}
```

### Toelichting

1. **Enum.GetValues:** De `Enum.GetValues`-methode haalt alle waarden van de `ActiviteitSoort`-enum op.
2. **Foreach-lus:** De `foreach`-lus gaat door elke waarde van de enum en voegt deze toe aan de `CboSoort`-ComboBox.
3. **SelectedIndex:** De `SelectedIndex` wordt optioneel ingesteld op 0 om ervoor te zorgen dat de eerste waarde standaard is geselecteerd. Dit is optioneel en kan worden weggelaten als je geen standaard geselecteerde waarde wilt.

Met deze aanpak kun je eenvoudig je `ComboBox` vullen met de waarden van de `ActiviteitSoort`-enum in een andere pagina van je WPF-project.



```
private void CboSoort_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}
}
```

Haal hier enkel de Enum welke geselecteerd is. niet meer dan dit geven ik ga het zelf programmeren. enkel sender converteren naar geselecteerde enum



Natuurlijk, hier is hoe je de geselecteerde enum-waarde kunt ophalen in de `CboSoort\_SelectionChanged`-eventhandler door de `sender` te converteren naar de geselecteerde enum-waarde:

csharp

```
private void CboSoort_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (sender is ComboBox comboBox && comboBox.SelectedItem is ActiviteitSoort
        selectedSoort)
    {
        // selectedSoort bevat nu de geselecteerde enum-waarde
    }
}
```

## Toelichting

1. **Casting van Sender:** `sender` wordt eerst gecast naar een `ComboBox`.
2. **SelectedItem:** Vervolgens wordt gecontroleerd of `SelectedItem` een `ActiviteitSoort` is.
3. **Gebruik van de Geselecteerde Enum-Waarde:** `selectedSoort` bevat de geselecteerde enum-waarde.

Met deze aanpak kun je de geselecteerde enum-waarde gemakkelijk verkrijgen en gebruiken in je verdere logica.



```
if (int.TryParse(specifiekText, out int specifiek))
{
    ActiviteitMoeilijkheid moeilijkheid =
}
```

try Parse int to enum



Om een integer-waarde om te zetten naar een enum-waarde in C#, kun je de `Enum.IsDefined`-methode gebruiken om te controleren of de integer-waarde geldig is voor de enum, en daarna kun je de `Enum.ToObject`-methode gebruiken om de integer-waarde naar de bijbehorende enum-waarde te converteren.

Hier is een voorbeeld van hoe je dit kunt doen:

## Stap 1: Definieer de Enum

Zorg ervoor dat je `ActiviteitMoeilijkheid`-enum ergens in je project is gedefinieerd:

```
csharp

public enum ActiviteitMoeilijkheid
{
    Gemakkelijk = 1,
    Gemiddeld = 2,
    Moeilijk = 3
    // Voeg meer moeilijkheidsniveaus toe zoals nodig
}
```

## Stap 2: Integer naar Enum Converteren

In je code kun je de integer-waarde naar een enum-waarde converteren:

```
csharp

if (int.TryParse(specifiekText, out int specifiek))
{
    if (Enum.IsDefined(typeof(ActiviteitMoeilijkheid), specifiek))
    {
        ActiviteitMoeilijkheid moeilijkheid = (ActiviteitMoeilijkheid)specifiek;
        // Nu heb je de specifiek omgezet naar de enum-waarde moeilijkheid
    }
    else
    {
        // Specifiek is geen geldige waarde voor de ActiviteitMoeilijkheid enum
    }
}
else
{
    // specifiekText kon niet worden geparsed naar een integer
}
```

## Toelichting

1. **Int.TryParse:** Probeer de `specifiekText` te parsen naar een integer. Als dit lukt, wordt de integer opgeslagen in `specifiek`.
2. **Enum.IsDefined:** Controleer of de integer-waarde geldig is voor de `ActiviteitMoeilijkheid`-enum.
3. **Enum Casten:** Als de integer-waarde geldig is, cast deze naar de `ActiviteitMoeilijkheid`-enum.
4. **Foutenafhandeling:** Voeg logica toe voor het geval de integer-waarde niet geldig is of de parsing mislukt.

Met deze aanpak kun je veilig een integer-waarde omzetten naar een bijbehorende enum-waarde, waarbij je controleert of de integer-waarde geldig is binnen de context van de enum.



System.Data.SqlClient.SqlException: 'Arithmetic overflow error converting numeric to data type numeric.'

The statement has been terminated.'

at

```
else if (this is SportActiviteit sportActiviteit)
{
    sql = "INSERT INTO activiteit (Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude,
    MaxPersonen, Soort, Leeftijdsgroep, Organisator_Id, Moeilijkheid) " +
        "VALUES (@Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude,
    @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Moeilijkheid)";
    using (SqlCommand cmd = new SqlCommand(sql, conn))
    {
        cmd.Parameters.AddWithValue("@Titel", sportActiviteit.Titel);
        cmd.Parameters.AddWithValue("@Beschrijving", sportActiviteit.Beschrijving);
        cmd.Parameters.AddWithValue("@DatumTijd", sportActiviteit.DatumTijd);
        if (sportActiviteit.Icoon == null)
        {
            SqlParameter imageParameter = new SqlParameter("@Icoon", SqlDbType.Image);
            imageParameter.Value = DBNull.Value;
            cmd.Parameters.Add(imageParameter);
        }
        else
        {
            cmd.Parameters.AddWithValue("@Icoon", sportActiviteit.Icoon);
        }
        cmd.Parameters.AddWithValue("@Longitude", sportActiviteit.Longitude);
        cmd.Parameters.AddWithValue("@Latitude", sportActiviteit.Latitude);
        cmd.Parameters.AddWithValue("@MaxPersonen", sportActiviteit.MaxPersonen);
        cmd.Parameters.AddWithValue("@Soort", (int)sportActiviteit.Soort);
        cmd.Parameters.AddWithValue("@Leeftijdsgroep", sportActiviteit.Leeftijdsgroep);
        cmd.Parameters.AddWithValue("@OrganisatorId", sportActiviteit.OrganisatorId);
        cmd.Parameters.AddWithValue("@Moeilijkheid", sportActiviteit.Moeilijkheid == null ?
        DBNull.Value : (int)sportActiviteit.Moeilijkheid);
        cmd.ExecuteNonQuery();
    }
}
```



De fout `Arithmetic overflow error converting numeric to data type numeric` treedt op wanneer een waarde die je probeert in te voegen of bij te werken niet past in de kolom die is gedefinieerd in de database. Dit kan gebeuren als de waarde te groot of te klein is voor het gespecificeerde datatype van de kolom.

Om deze fout op te lossen, moeten we een paar dingen controleren:

1. **Controleer de datatypes in de database:** Zorg ervoor dat de kolommen in de `activiteit`-tabel de juiste datatypes en groottes hebben om de waarden die je probeert in te voegen te accommoderen.
2. **Controleer de waarden die je probeert in te voegen:** Zorg ervoor dat de waarden die je in de parameters stopt binnen de toegestane bereiken vallen voor de kolommen in de database.

Laten we eens kijken naar de typische datatypes en mogelijke oorzaken van problemen in jouw situatie:

- **Longitude en Latitude:** Zorg ervoor dat deze kolommen groot genoeg zijn om decimale waarden op te slaan (bijvoorbeeld `decimal(18, 10)`).
- **MaxPersonen:** Controleer het datatype, bijvoorbeeld `int` of `smallint`.
- **Soort en Moeilijkheid:** Deze zijn meestal `int` in de database, dus de waarden zouden moeten passen, maar controleer of ze geen waarden buiten het bereik hebben.
- **Icoon:** Dit lijkt in orde, gezien de null-check en het gebruik van `SqlDbType.Image`.

Hier is een geüpdatet voorbeeld van je code met enkele aanpassingen om mogelijke fouten te helpen oplossen en betere foutafhandeling toe te voegen:

csharp

```
else if (this is SportActiviteit sportActiviteit)
{
    sql = "INSERT INTO activiteit (Titel, Beschrijving, DatumTijd, Icoon, Longitude,
Latitude, MaxPersonen, Soort, Leeftijdsgroep, Organisator_Id, Moeilijkheid) " +
        "VALUES (@Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude,
@MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Moeilijkheid)";
    using (SqlCommand cmd = new SqlCommand(sql, conn))
    {
        cmd.Parameters.AddWithValue("@Titel", sportActiviteit.Titel);
        cmd.Parameters.AddWithValue("@Beschrijving", sportActiviteit.Beschrijving);
        cmd.Parameters.AddWithValue("@DatumTijd", sportActiviteit.DatumTijd);

        if (sportActiviteit.Icoon == null)
        {
            SqlParameter imageParameter = new SqlParameter("@Icoon", SqlDbType.Image);
            imageParameter.Value = DBNull.Value;
            cmd.Parameters.Add(imageParameter);
        }
        else
        {
            cmd.Parameters.AddWithValue("@Icoon", sportActiviteit.Icoon);
        }

        // Ensure the longitude and latitude are within reasonable ranges
        if (sportActiviteit.Longitude < -180 || sportActiviteit.Longitude > 180)
            throw new ArgumentOutOfRangeException("Longitude must be between -180 and
180.");
        if (sportActiviteit.Latitude < -90 || sportActiviteit.Latitude > 90)
            throw new ArgumentOutOfRangeException("Latitude must be between -90 and 90.");

        cmd.Parameters.AddWithValue("@Longitude", sportActiviteit.Longitude);
        cmd.Parameters.AddWithValue("@Latitude", sportActiviteit.Latitude);

        // Ensure MaxPersonen is within a reasonable range, depending on your database
        schema
        if (sportActiviteit.MaxPersonen < 0 || sportActiviteit.MaxPersonen > int.MaxValue)
            throw new ArgumentOutOfRangeException("MaxPersonen must be between 0 and the
maximum integer value.");

        cmd.Parameters.AddWithValue("@MaxPersonen", sportActiviteit.MaxPersonen);
        cmd.Parameters.AddWithValue("@Soort", (int)sportActiviteit.Soort);
        cmd.Parameters.AddWithValue("@Leeftijdsgroep", sportActiviteit.Leeftijdsgroep);
        cmd.Parameters.AddWithValue("@OrganisatorId", sportActiviteit.OrganisatorId);
        cmd.Parameters.AddWithValue("@Moeilijkheid", sportActiviteit.Moeilijkheid == null
? DBNull.Value : (int)sportActiviteit.Moeilijkheid);

        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (SqlException ex)
        {
            // Log or handle the exception as needed

```

```

        MessageBox.Show("Error inserting data: " + ex.Message);
    }
}

```

## Mogelijke oorzaken en oplossingen:

### 1. Longitude en Latitude:

- Zorg ervoor dat de waarden binnen de geldige bereik vallen.
- Controleer het datatype in de database. Voor lange decimale waarden, gebruik ``decimal(18, 10)`` of iets dergelijks.

### 2. MaxPersonen:

- Zorg ervoor dat deze waarde niet groter is dan het maximum dat het datatype in de database toestaat (``int`` of ``smallint``).

### 3. Soort en Moeilijkheid:

- Controleer of de waarden binnen het bereik vallen van de enum en het datatype in de database.

### 4. Algemene foutafhandeling:

- Vang ``SqlException`` op en geef een duidelijke foutmelding of log de fout voor verdere diagnose.

Controleer ook de definitie van je ``activiteit``-tabel in de database om te zorgen dat de datatypes geschikt zijn voor de gegevens die je probeert in te voegen.



i found it. 50,849400 from my textbox is being parsed as 50849400 instead of 50.849400 for longitude and latitude. How to check if the comma is read correctly? or check if the number fits in database. in SQL it is decimal(9,6)

here are my checks:

```

if (!decimal.TryParse(longitudeText, out longitude))
{
    LblError.Content = "Longitude verplicht en moet een getal zijn!";
    return;
}
else
{
}

if (!decimal.TryParse(latitudeText, out latitude))
{
    LblError.Content = "Latitude verplicht en moet een getal zijn!";
    return;
}

```



Het probleem dat je beschrijft, komt vaak voor bij het parseren van numerieke waarden uit tekstvelden, vooral wanneer de decimalen niet correct worden geïnterpreteerd, bijvoorbeeld door het

gebruik van een komma in plaats van een punt als decimaal scheidingsteken.

## Oplossingen en Voorstellen:

### 1. Controleer het decimale scheidingsteken:

- Zorg ervoor dat het decimale scheidingsteken correct wordt geïnterpreteerd. In Nederland en sommige andere landen wordt de komma (`,` ) gebruikt als decimaal scheidingsteken. Dit kan problemen veroorzaken bij het parsen als de parser verwacht dat de punt (`. `) het decimale scheidingsteken is.

### 2. Gebruik `CultureInfo` voor parsing:

- Gebruik de `CultureInfo`-klasse om expliciet het juiste decimale scheidingsteken op te geven. Dit voorkomt dat komma's worden geïnterpreteerd als duizenden scheidingstekens.

## Codevoorbeelden:

Hier is hoe je de decimalen kunt parsen met behulp van `CultureInfo`:

csharp

```
using System.Globalization;

decimal longitude;
decimal latitude;

string longitudeText = "50,849400"; // Voorbeeldwaarde met komma
string latitudeText = "5,123456";   // Voorbeeldwaarde met komma

// Gebruik CultureInfo met de juiste decimalen scheidingstekens
var culture = new CultureInfo("nl-NL");

if (!decimal.TryParse(longitudeText, NumberStyles.Any, culture, out longitude))
{
    LblError.Content = "Longitude verplicht en moet een getal zijn!";
    return;
}
if (!decimal.TryParse(latitudeText, NumberStyles.Any, culture, out latitude))
{
    LblError.Content = "Latitude verplicht en moet een getal zijn!";
    return;
}

// Nu kun je de longitude en latitude veilig gebruiken
```

## Controleer of de waarde in het bereik valt:

Voordat je de waarden invoegt in de database, kun je controleren of ze binnen het bereik vallen van het datatype `decimal(9,6)`.

csharp

```
// Check of de waarde binnen het bereik valt
if (longitude < -180 || longitude > 180)
{
    LblError.Content = "Longitude moet tussen -180 en 180 liggen.";
    return;
}

if (latitude < -90 || latitude > 90)
{
    LblError.Content = "Latitude moet tussen -90 en 90 liggen.";
    return;
}
```



```
LblError.Content = "Latitude moet tussen -90 en 90 liggen.";
return;
}
```

## Voorbeeld van de volledige invoeg-logica:

Hier is hoe je de invoeg-logica zou kunnen integreren met de juiste checks:

csharp

```
else if (this is SportActiviteit sportActiviteit)
{
    string sql = "INSERT INTO activiteit (Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, Organisator_Id, Moeilijkheid) " +
        "VALUES (@Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Moeilijkheid)";

    using (SqlCommand cmd = new SqlCommand(sql, conn))
    {
        cmd.Parameters.AddWithValue("@Titel", sportActiviteit.Titel);
        cmd.Parameters.AddWithValue("@Beschrijving", sportActiviteit.Beschrijving);
        cmd.Parameters.AddWithValue("@DatumTijd", sportActiviteit.DatumTijd);

        if (sportActiviteit.Icoon == null)
        {
            SqlParameter imageParameter = new SqlParameter("@Icoon", SqlDbType.Image);
            imageParameter.Value = DBNull.Value;
            cmd.Parameters.Add(imageParameter);
        }
        else
        {
            cmd.Parameters.AddWithValue("@Icoon", sportActiviteit.Icoon);
        }

        // Controleer decimalen met juiste CultureInfo
        var culture = new CultureInfo("nl-NL");
        if (!decimal.TryParse(longitudeText, NumberStyles.Any, culture, out decimal longitude))
        {
            LblError.Content = "Longitude verplicht en moet een getal zijn!";
            return;
        }
        if (!decimal.TryParse(latitudeText, NumberStyles.Any, culture, out decimal latitude))
        {
            LblError.Content = "Latitude verplicht en moet een getal zijn!";
            return;
        }

        // Check of de waarde binnen het bereik valt
        if (longitude < -180 || longitude > 180)
        {
            LblError.Content = "Longitude moet tussen -180 en 180 liggen.";
            return;
        }

        if (latitude < -90 || latitude > 90)
        {
            LblError.Content = "Latitude moet tussen -90 en 90 liggen.";
            return;
        }

        cmd.Parameters.AddWithValue("@Longitude", longitude);
        cmd.Parameters.AddWithValue("@Latitude", latitude);
        cmd.Parameters.AddWithValue("@MaxPersonen", sportActiviteit.MaxPersonen);
        cmd.Parameters.AddWithValue("@Soort", (int)sportActiviteit.Soort);
        cmd.Parameters.AddWithValue("@Leeftijdsgroep", sportActiviteit.Leeftijdsgroep);
        cmd.Parameters.AddWithValue("@OrganisatorId", sportActiviteit.OrganisatorId);
        cmd.Parameters.AddWithValue("@Moeilijkheid", sportActiviteit.Moeilijkheid == null
```

```
? DBNull.Value : (int)sportActiviteit.Moeilijkheid);

        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (SqlException ex)
        {
            // Log of handel de fout zoals nodig
            MessageBox.Show("Fout bij het invoegen van gegevens: " + ex.Message);
        }
    }
}
```

## Samenvatting:

- Gebruik `CultureInfo` met het juiste decimale scheidingsteken bij het parseren van string-waarden naar decimalen.
- Controleer of de waarden binnen de toegestane bereik vallen voordat je ze naar de database schrijft.
- Voeg foutafhandeling toe om eventuele fouten tijdens het invoegen te behandelen.

Met deze aanpak zou je probleem met het incorrect parseren van decimalen en de `Arithmetic overflow error` moeten worden opgelost.

ChatGPT kan fouten maken. Controleer belangrijke informatie.