

SQL Error Troubleshooting Guide



```
Msg 102, Level 15, State 6, Line 71
Incorrect syntax near 'ACCELERATED_DATABASE_RECOVERY'.
Msg 102, Level 15, State 1, Line 75
Incorrect syntax near 'WAIT_STATS_CAPTURE_MODE'.
Msg 155, Level 15, State 1, Line 102
'OPTIMIZE_FOR_SEQUENTIAL_KEY' is not a recognized CREATE TABLE option.
Msg 155, Level 15, State 1, Line 117
'OPTIMIZE_FOR_SEQUENTIAL_KEY' is not a recognized CREATE TABLE option.
Msg 155, Level 15, State 1, Line 137
'OPTIMIZE_FOR_SEQUENTIAL_KEY' is not a recognized CREATE TABLE option.
Msg 4902, Level 16, State 1, Line 140
Cannot find the object "dbo.Activiteit" because it does not exist or you do not have permissions.
Msg 4902, Level 16, State 1, Line 142
Cannot find the object "dbo.Activiteit" because it does not exist or you do not have permissions.
Msg 4902, Level 16, State 1, Line 144
Cannot find the object "dbo.Activiteit" because it does not exist or you do not have permissions.
Msg 4902, Level 16, State 1, Line 147
Cannot find the object "dbo.Activiteit" because it does not exist or you do not have permissions.
```



The error messages you are encountering in SQL Server indicate a variety of syntax issues and potential problems with object existence or permissions. Let's address each one:

1. Incorrect Syntax

plaintext

```
Msg 102, Level 15, State 6, Line 71
Incorrect syntax near 'ACCELERATED_DATABASE_RECOVERY'.
Msg 102, Level 15, State 1, Line 75
Incorrect syntax near 'WAIT_STATS_CAPTURE_MODE'.
```

These errors suggest there is incorrect syntax near 'ACCELERATED_DATABASE_RECOVERY' and 'WAIT_STATS_CAPTURE_MODE'. These might be commands or options that are not recognized in your SQL Server version. Make sure you are using a version that supports these features.

Solution:

- Verify the SQL Server version and check the compatibility of these options.
- Ensure that the syntax for enabling these features is correct and supported by your version.

2. Unrecognized CREATE TABLE Option

```
plaintext
```

```
Msg 155, Level 15, State 1, Line 102
'OPTIMIZE_FOR_SEQUENTIAL_KEY' is not a recognized CREATE TABLE option.
```

The option 'OPTIMIZE_FOR_SEQUENTIAL_KEY' is not recognized in the context of the `CREATE TABLE` statement. This option might be specific to certain SQL Server versions or configurations.

Solution:

- Check if 'OPTIMIZE_FOR_SEQUENTIAL_KEY' is supported in your SQL Server version.
- If not, remove or replace this option with a valid one.

3. Object Not Found or Permission Issues

```
plaintext
```

```
Msg 4902, Level 16, State 1, Line 140
Cannot find the object "dbo.Activiteit" because it does not exist or you do not have
permissions.
```

These errors indicate that the object `dbo.Activiteit` does not exist or there are insufficient permissions to access it.

Solution:

- Verify that the object `dbo.Activiteit` exists in the database.
- Ensure you have the necessary permissions to access the object.
- Check if there are any typos in the object name.

General Steps for Troubleshooting:

1. Check SQL Server Version:

Make sure your SQL Server version supports the features and options you are trying to use.

2. Syntax Validation:

Double-check the syntax for any new features or options. Refer to the official SQL Server documentation for the correct syntax.

3. Permissions:

Ensure that you have the appropriate permissions to perform operations on the database objects.

4. Object Existence:

Verify that all objects you are referencing exist in the database. You can query the system catalog views to check for object existence, e.g.:

```
sql
```

```
SELECT * FROM sys.objects WHERE name = 'Activiteit';
```

5. Error Line Reference:

The line numbers provided in the error messages indicate where the issue occurs in your script. Focus on these lines to identify and correct the issues.

By following these steps, you should be able to diagnose and resolve the errors in your SQL script. If you provide more context or specific code snippets, I can offer more targeted assistance.



SQL Server 13.0.4001



Given that you are using SQL Server 2016 (version 13.0.4001), let's address the issues specifically considering this version:

1. Incorrect Syntax

plaintext

```
Msg 102, Level 15, State 6, Line 71
Incorrect syntax near 'ACCELERATED_DATABASE_RECOVERY'.
Msg 102, Level 15, State 1, Line 75
Incorrect syntax near 'WAIT_STATS_CAPTURE_MODE'.
```

Solution:

- **ACCELERATED_DATABASE_RECOVERY** and **WAIT_STATS_CAPTURE_MODE** are features introduced in later versions of SQL Server. For instance, Accelerated Database Recovery is available from SQL Server 2019 (version 15.x) onwards. You need to remove or replace these features as they are not supported in SQL Server 2016.

2. Unrecognized CREATE TABLE Option

plaintext

```
Msg 155, Level 15, State 1, Line 102
'OPTIMIZE_FOR_SEQUENTIAL_KEY' is not a recognized CREATE TABLE option.
```

Solution:

- The **OPTIMIZE_FOR_SEQUENTIAL_KEY** option is also introduced in SQL Server 2019. Remove this option from your `CREATE TABLE` statement, as it is not supported in SQL Server 2016.`

3. Object Not Found or Permission Issues

plaintext

```
Msg 4902, Level 16, State 1, Line 140
Cannot find the object "dbo.Activiteit" because it does not exist or you do not have
permissions.
```

Solution:

- Verify the existence of the `dbo.Activiteit` object:

sql

```
SELECT * FROM sys.objects WHERE name = 'Activiteit';
```

- If the object does not exist, you need to create it before referencing it.
- Ensure that you have the necessary permissions to access the object:

sql

```
SELECT * FROM sys.database_principals WHERE name = USER_NAME();
```

Adjusted SQL Script

Here's a general outline of how you might adjust your script to be compatible with SQL Server 2016:

sql

```
-- Assuming you need to create a table without the unsupported options
CREATE TABLE dbo>YourTable (
    ID INT PRIMARY KEY,
    Name NVARCHAR(100)
    -- Other columns as needed
);

-- Ensure you have the necessary permissions
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo>YourTable TO [YourUser];

-- Verify if the table exists before performing operations
IF OBJECT_ID('dbo>YourTable', 'U') IS NOT NULL
BEGIN
    -- Your subsequent SQL operations
END

-- If you need to perform actions on 'dbo.Activiteit', ensure it exists
IF OBJECT_ID('dbo.Activiteit', 'U') IS NOT NULL
BEGIN
    -- Your SQL operations on dbo.Activiteit
END
ELSE
BEGIN
    PRINT 'The object dbo.Activiteit does not exist.';
END
```

By following these guidelines and removing or replacing the unsupported options, your SQL script should run without the errors you're encountering. If you need further specific adjustments, please

provide more context or specific parts of your SQL script for a more tailored solution.



how can i update version



Upgrading SQL Server to a newer version involves several steps, including planning, preparing, and performing the upgrade. Here are the general steps to upgrade SQL Server from version 2016 to a newer version like SQL Server 2019 or SQL Server 2022:

1. Planning

A. Choose the Version to Upgrade To:

- Decide which version of SQL Server you want to upgrade to. The latest versions as of now are SQL Server 2019 and SQL Server 2022.

B. Check Compatibility:

- Ensure that your applications and scripts are compatible with the new version.
- Review the [SQL Server upgrade documentation](#).

2. Preparation

A. Backup:

- Perform a full backup of all databases.
- Backup the SQL Server configuration settings and any custom scripts or jobs.

B. Test the Upgrade:

- Ideally, set up a test environment that mimics your production environment.
- Perform a test upgrade and validate that all applications function as expected.

C. Review Upgrade Checklist:

- Review the official [SQL Server upgrade checklist](#).

3. Performing the Upgrade

A. Download the Installation Media:

- Download the ISO file for the new version from the Microsoft website or the [Microsoft Volume Licensing Service Center](#).

B. Launch the Installation:

- Run the SQL Server setup executable (`setup.exe`).
- Select the option to upgrade from a previous version of SQL Server.

C. Follow the Upgrade Wizard:

- Follow the prompts in the SQL Server Installation Wizard to upgrade your SQL Server instance.
- Choose the instance you want to upgrade.
- Review the upgrade rules and fix any issues reported.
- Proceed with the upgrade process.

D. Post-Upgrade Tasks:

- Apply the latest cumulative updates for the new SQL Server version.
- Update database compatibility levels if necessary.
- Run DBCC CHECKDB on all databases to ensure integrity.
- Monitor the server closely for any issues.

Example: Upgrading SQL Server 2016 to SQL Server 2019

- 1. Download and Mount SQL Server 2019 ISO:**
 - Download from the [SQL Server 2019 download page](#).
- 2. Run Setup:**
 - Run `setup.exe` from the mounted ISO or extracted files.
- 3. Upgrade Wizard:**
 - Select "Upgrade from a previous version of SQL Server".
 - Follow the prompts, selecting your SQL Server 2016 instance to upgrade.
 - Complete the upgrade process.
- 4. Post-Upgrade:**
 - Apply any cumulative updates.
 - Validate functionality by running tests on your applications.
 - Update compatibility levels and perform DBCC CHECKDB.

Additional Resources:

- [SQL Server Upgrade Advisor](#)
- [SQL Server 2019 Release Notes](#)
- [SQL Server 2022 Release Notes](#)

By carefully following these steps and utilizing the provided resources, you can upgrade your SQL Server instance to a newer version smoothly.



Fill in my list with the following data from my script: My Persoon object looks like this:

```
public int Id { get; set; }
public string Voornaam { get; set; }
public string Achternaam { get; set; }
public string Login { get; set; }
public string Paswoord { get; set; }
public byte[] Profielfoto { get; set; }
public DateTime RegDatum { get; set; }
public bool IsAdmin { get; set; }
```

Profielfoto can be null for now

```
private static List<Persoon> list = new List<Persoon>
```

```
{
```

```
};
```

```
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (1, N'Rogier', N'van der Linde', N'rogier',
N'1b4f0e9851971998e732078544c96b36c3d01cedf7caa332359d6f1d83567014', , CAST(N'2020-12-
23' AS Date), 1)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (2, N'Frank ', N'Salliau', N'frank',
N'60303ae22b998861bce3b28f33eec1be758a213c86c93c076dbe9f558c11c752', , CAST(N'2021-03-
03' AS Date), 1)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (3, N'Abdelkarim', N'Chouikh', N'abdel',
N'fd61a03af4f77d870fc21e05e7e80678095c92d808cfb3b5c279ee04c74aca13', , CAST(N'2023-11-
11' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (4, N'Alexandre', N'Kamnerdsiri', N'alex',
N'a4e624d686e03ed2767c0abd85c14426b0b1157d2ce81d27bb4fe4f6f01d688a', , CAST(N'2023-06-
17' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (5, N'Ayoub', N'Aymen', N'ayoub',
N'a140c0c1eda2def2b830363ba362aa4d7d255c262960544821f556e16661b6ff', , CAST(N'2022-12-
12' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (6, N'Brian', N'Sutanto', N'brian',
N'ed0cb90bdfa4f93981a7d03cff99213a86aa96a6cbcf89ec5e8889871f088727', , CAST(N'2023-09-
17' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (7, N'Marcel', N'De Schrijver', N'marcel',
N'bd7c911264aae15b66d4291b6850829aa96986b1d3ead34d1fdbfef27056c112', , CAST(N'2023-
09-17' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (8, N'Miranda', N'Sousa', N'mira',
N'1f9bfef15fee8a10c4d0711c7eb0c083962123e1918e461b6a508e7146c189b2', , CAST(N'2024-01-
11' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (9, N'Nelly', N'Lihmedi', N'nelly',
N'b4451034d3b6590060ce9484a28b88dd332a80a22ae8e39c9c5cb7357ab26c9f', , CAST(N'2023-
10-09' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (10, N'Sandra', N'Lauer', N'sandra',
N'ec2738feb2bbb0bc783eb4667903391416372ba6ed8b8dddbebbdb37e5102473', , CAST(N'2024-
02-13' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (11, N'Soo', N'Wang', N'soo',
N'744ea9ec6fa0a83e9764b4e323d5be6b55a5accfc7fe4c08eab6a8de1fca4855', , CAST(N'2023-04-
21' AS Date), 0)
INSERT [dbo].[Person] ([id], [firstname], [lastname], [login], [password], [regdate], [isadmin])
VALUES (12, N'Sophie', N'Laethem', N'sophie',
```

```
N'a98ec5c5044800c88e862f007b98d89815fc40ca155d6ce7909530d792e909ce', , CAST(N'2023-08-08' AS Date), 0)
```



Let's translate your SQL `INSERT` statements into `Persoon` objects and add them to the `list`. Note that the `Profielfoto` property will be set to `null` for now.

Here is the complete code with the populated list:

csharp

```
public class Persoon
{
    public int Id { get; set; }
    public string Voornaam { get; set; }
    public string Achternaam { get; set; }
    public string Login { get; set; }
    public string Paswoord { get; set; }
    public byte[] Profielfoto { get; set; }
    public DateTime RegDatum { get; set; }
    public bool IsAdmin { get; set; }
}

private static List<Persoon> list = new List<Persoon>
{
    new Persoon { Id = 1, Voornaam = "Rogier", Achternaam = "van der Linde", Login = "rogier", Paswoord = "1b4f0e9851971998e732078544c96b36c3d01cedf7caa332359d6f1d83567014", Profielfoto = null, RegDatum = new DateTime(2020, 12, 23), IsAdmin = true },
    new Persoon { Id = 2, Voornaam = "Frank", Achternaam = "Salliau", Login = "frank", Paswoord = "60303ae22b998861bce3b28f33eec1be758a213c86c93c076dbe9f558c11c752", Profielfoto = null, RegDatum = new DateTime(2021, 3, 3), IsAdmin = true },
    new Persoon { Id = 3, Voornaam = "Abdelkarim", Achternaam = "Chouikh", Login = "abdel", Paswoord = "fd61a03af4f77d870fc21e05e7e80678095c92d808cfb3b5c279ee04c74aca13", Profielfoto = null, RegDatum = new DateTime(2023, 11, 11), IsAdmin = false },
    new Persoon { Id = 4, Voornaam = "Alexandre", Achternaam = "Kamnerdsiri", Login = "alex", Paswoord = "a4e624d686e03ed2767c0abd85c14426b0b1157d2ce81d27bb4fe4f6f01d688a", Profielfoto = null, RegDatum = new DateTime(2023, 6, 17), IsAdmin = false },
    new Persoon { Id = 5, Voornaam = "Ayoub", Achternaam = "Aymen", Login = "ayoub", Paswoord = "a140c0cleda2def2b830363ba362aa4d7d255c262960544821f556e16661b6ff", Profielfoto = null, RegDatum = new DateTime(2022, 12, 12), IsAdmin = false },
    new Persoon { Id = 6, Voornaam = "Brian", Achternaam = "Sutanto", Login = "brian", Paswoord = "ed0cb90bdfa4f93981a7d03cff99213a86aa96a6cbcf89ec5e8889871f088727", Profielfoto = null, RegDatum = new DateTime(2023, 9, 17), IsAdmin = false },
    new Persoon { Id = 7, Voornaam = "Marcel", Achternaam = "De Schrijver", Login = "marcel", Paswoord = "bd7c911264aae15b66d4291b6850829aa96986b1d3ead34d1fdbfef27056c112", Profielfoto = null, RegDatum = new DateTime(2023, 9, 17), IsAdmin = false },
    new Persoon { Id = 8, Voornaam = "Miranda", Achternaam = "Sousa", Login = "mira", Paswoord = "1f9bfbef15fee8a10c4d0711c7eb0c083962123e1918e461b6a508e7146c189b2", Profielfoto = null, RegDatum = new DateTime(2024, 1, 11), IsAdmin = false },
    new Persoon { Id = 9, Voornaam = "Nelly", Achternaam = "Lihmedi", Login = "nelly", Paswoord = "b4451034d3b6590060ce9484a28b88dd332a80a22ae8e39c9c5cb7357ab26c9f", Profielfoto = null, RegDatum = new DateTime(2023, 10, 9), IsAdmin = false },
    new Persoon { Id = 10, Voornaam = "Sandra", Achternaam = "Lauer", Login = "sandra", Paswoord = "ec2738feb2bbb0bc783eb4667903391416372ba6ed8b8dddbbebbdb37e5102473", Profielfoto = null, RegDatum = new DateTime(2024, 2, 13), IsAdmin = false },
    new Persoon { Id = 11, Voornaam = "Soo", Achternaam = "Wang", Login = "soo", Paswoord = "744ea9ec6fa0a83e9764b4e323d5be6b55a5accfc7fe4c08eab6a8de1fca4855", Profielfoto = null, RegDatum = new DateTime(2023, 4, 21), IsAdmin = false },
    new Persoon { Id = 12, Voornaam = "Sophie", Achternaam = "Laethem", Login = "sophie", Paswoord = "a98ec5c5044800c88e862f007b98d89815fc40ca155d6ce7909530d792e909ce", Profielfoto = null, RegDatum = new DateTime(2023, 8, 8), IsAdmin = false }
};
```

This code snippet defines the `Persoon` class and populates the `list` with the provided data. Each `Profielfoto` property is set to `null` as specified.


```
public class DataInitializerService
{
    private static string connString =
    ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;

    private static List<Persoon> list = new List<Persoon>
    {
        new Persoon { Id = 1, Voornaam = "Rogier", Achternaam = "van der Linde", Login =
"rogier", Paswoord =
"1b4f0e9851971998e732078544c96b36c3d01cedf7caa332359d6f1d83567014", Profielfoto = null,
RegDatum = new DateTime(2020, 12, 23), IsAdmin = true },
        new Persoon { Id = 2, Voornaam = "Frank", Achternaam = "Salliau", Login = "frank",
Paswoord = "60303ae22b998861bce3b28f33eec1be758a213c86c93c076dbe9f558c11c752",
Profielfoto = null, RegDatum = new DateTime(2021, 3, 3), IsAdmin = true },
        new Persoon { Id = 3, Voornaam = "Abdelkarim", Achternaam = "Chouikh", Login = "abdel",
Paswoord = "fd61a03af4f77d870fc21e05e7e80678095c92d808cfb3b5c279ee04c74aca13",
Profielfoto = null, RegDatum = new DateTime(2023, 11, 11), IsAdmin = false },
        new Persoon { Id = 4, Voornaam = "Alexandre", Achternaam = "Kamnerdsiri", Login = "alex",
Paswoord = "a4e624d686e03ed2767c0abd85c14426b0b1157d2ce81d27bb4fe4f6f01d688a",
Profielfoto = null, RegDatum = new DateTime(2023, 6, 17), IsAdmin = false },
        new Persoon { Id = 5, Voornaam = "Ayoub", Achternaam = "Aymen", Login = "ayoub",
Paswoord = "a140c0c1eda2def2b830363ba362aa4d7d255c262960544821f556e16661b6ff",
Profielfoto = null, RegDatum = new DateTime(2022, 12, 12), IsAdmin = false },
        new Persoon { Id = 6, Voornaam = "Brian", Achternaam = "Sutanto", Login = "brian",
Paswoord = "ed0cb90bdfa4f93981a7d03cff99213a86aa96a6cbcf89ec5e8889871f088727",
Profielfoto = null, RegDatum = new DateTime(2023, 9, 17), IsAdmin = false },
        new Persoon { Id = 7, Voornaam = "Marcel", Achternaam = "De Schrijver", Login = "marcel",
Paswoord = "bd7c911264aae15b66d4291b6850829aa96986b1d3ead34d1fdbfef27056c112",
Profielfoto = null, RegDatum = new DateTime(2023, 9, 17), IsAdmin = false },
        new Persoon { Id = 8, Voornaam = "Miranda", Achternaam = "Sousa", Login = "mira",
Paswoord = "1f9bfeb15fee8a10c4d0711c7eb0c083962123e1918e461b6a508e7146c189b2",
Profielfoto = null, RegDatum = new DateTime(2024, 1, 11), IsAdmin = false },
        new Persoon { Id = 9, Voornaam = "Nelly", Achternaam = "Lihmedi", Login = "nelly",
Paswoord = "b4451034d3b6590060ce9484a28b88dd332a80a22ae8e39c9c5cb7357ab26c9f",
Profielfoto = null, RegDatum = new DateTime(2023, 10, 9), IsAdmin = false },
        new Persoon { Id = 10, Voornaam = "Sandra", Achternaam = "Lauer", Login = "sandra",
Paswoord = "ec2738feb2bbb0bc783eb4667903391416372ba6ed8b8dddbebbdb37e5102473",
Profielfoto = null, RegDatum = new DateTime(2024, 2, 13), IsAdmin = false },
        new Persoon { Id = 11, Voornaam = "Soo", Achternaam = "Wang", Login = "soo", Paswoord
= "744ea9ec6fa0a83e9764b4e323d5be6b55a5accfc7fe4c08eab6a8de1fca4855", Profielfoto = null,
RegDatum = new DateTime(2023, 4, 21), IsAdmin = false },
        new Persoon { Id = 12, Voornaam = "Sophie", Achternaam = "Laethem", Login = "sophie",
Paswoord = "a98ec5c5044800c88e862f007b98d89815fc40ca155d6ce7909530d792e909ce",
Profielfoto = null, RegDatum = new DateTime(2023, 8, 8), IsAdmin = false }
    };

    public static void InitializeData()
    {
        using (SqlConnection conn = new SqlConnection(connString))
        {

```

```
// open connectie
conn.Open();

// voer SQL commando uit
SqlCommand comm = new SqlCommand("SELECT * FROM Person", conn);

SqlDataReader reader = comm.ExecuteReader();
if (!reader.Read())
{
    FillPersons();
}
}

private static void FillPersons()
{
    throw new NotImplementedException();
}
}
```

Create FillPersons For me that inserts all the people.

I added images. i got 5 female images that go like this: f1.jpg, f2.jpg, ...
and 11 male images that go the same way. m1.jpg, m2.jpg, ...
Assets/Profielfotos/f1.png
Assets/Profielfotos/m1.png



Certainly! Let's create the `FillPersons` method to insert the list of `Persoon` objects into the `Person` table in your database. This method will also handle reading the profile pictures from the given file paths and converting them to a byte array to store them in the database. Here's the updated code with the `FillPersons` method:

csharp

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;

public class Persoon
{
    public int Id { get; set; }
    public string Voornaam { get; set; }
    public string Achternaam { get; set; }
    public string Login { get; set; }
    public string Paswoord { get; set; }
    public byte[] Profielfoto { get; set; }
    public DateTime RegDatum { get; set; }
    public bool IsAdmin { get; set; }
}

public class DataInitializerService
{
}
```

```

private static string connString =
ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;

private static List<Persoon> list = new List<Persoon>
{
    new Persoon { Id = 1, Voornaam = "Rogier", Achternaam = "van der Linde", Login =
"rogier", Paswoord = "1b4f0e9851971998e732078544c96b36c3d01cedf7caa332359d6f1d83567014",
Profielfoto = LoadImage("Assets/Profielfotos/m1.png"), RegDatum = new DateTime(2020, 12,
23), IsAdmin = true },
    new Persoon { Id = 2, Voornaam = "Frank", Achternaam = "Salliau", Login = "frank",
Paswoord = "60303ae22b998861bce3b28f33eec1be758a213c86c93c076dbe9f558c11c752", Profielfoto
= LoadImage("Assets/Profielfotos/m2.png"), RegDatum = new DateTime(2021, 3, 3), IsAdmin =
true },
    new Persoon { Id = 3, Voornaam = "Abdelkarim", Achternaam = "Chouikh", Login =
"abdel", Paswoord = "fd61a03af4f77d870fc21e05e7e80678095c92d808cfb3b5c279ee04c74aca13",
Profielfoto = LoadImage("Assets/Profielfotos/m3.png"), RegDatum = new DateTime(2023, 11,
11), IsAdmin = false },
    new Persoon { Id = 4, Voornaam = "Alexandre", Achternaam = "Kamnerdsiri", Login =
"alex", Paswoord = "a4e624d686e03ed2767c0abd85c14426b0b1157d2ce81d27bb4fe4f6f01d688a",
Profielfoto = LoadImage("Assets/Profielfotos/m4.png"), RegDatum = new DateTime(2023, 6,
17), IsAdmin = false },
    new Persoon { Id = 5, Voornaam = "Ayoub", Achternaam = "Aymen", Login = "ayoub",
Paswoord = "a140c0cleda2def2b830363ba362aa4d7d255c262960544821f556e16661b6ff", Profielfoto
= LoadImage("Assets/Profielfotos/m5.png"), RegDatum = new DateTime(2022, 12, 12), IsAdmin
= false },
    new Persoon { Id = 6, Voornaam = "Brian", Achternaam = "Sutanto", Login = "brian",
Paswoord = "ed0cb90bdfa4f93981a7d03cff99213a86aa96a6cbcf89ec5e8889871f088727", Profielfoto
= LoadImage("Assets/Profielfotos/m6.png"), RegDatum = new DateTime(2023, 9, 17), IsAdmin =
false },
    new Persoon { Id = 7, Voornaam = "Marcel", Achternaam = "De Schrijver", Login =
"marcel", Paswoord = "bd7c911264aae15b66d4291b6850829aa96986b1d3ead34d1fdbfdef27056c112",
Profielfoto = LoadImage("Assets/Profielfotos/m7.png"), RegDatum = new DateTime(2023, 9,
17), IsAdmin = false },
    new Persoon { Id = 8, Voornaam = "Miranda", Achternaam = "Sousa", Login = "mira",
Paswoord = "1f9bfeb15fee8a10c4d0711c7eb0c083962123e1918e461b6a508e7146c189b2", Profielfoto
= LoadImage("Assets/Profielfotos/f1.png"), RegDatum = new DateTime(2024, 1, 11), IsAdmin =
false },
    new Persoon { Id = 9, Voornaam = "Nelly", Achternaam = "Lihmedi", Login = "nelly",
Paswoord = "b4451034d3b6590060ce9484a28b88dd332a80a22ae8e39c9c5cb7357ab26c9f", Profielfoto
= LoadImage("Assets/Profielfotos/f2.png"), RegDatum = new DateTime(2023, 10, 9), IsAdmin =
false },
    new Persoon { Id = 10, Voornaam = "Sandra", Achternaam = "Lauer", Login =
"sandra", Paswoord = "ec2738feb2bbb0bc783eb4667903391416372ba6ed8b8dddbbebbdb37e5102473",
Profielfoto = LoadImage("Assets/Profielfotos/f3.png"), RegDatum = new DateTime(2024, 2,
13), IsAdmin = false },
    new Persoon { Id = 11, Voornaam = "Soo", Achternaam = "Wang", Login = "soo",
Paswoord = "744ea9ec6fa0a83e9764b4e323d5be6b55a5accfc7fe4c08eab6a8del1fca4855", Profielfoto
= LoadImage("Assets/Profielfotos/f4.png"), RegDatum = new DateTime(2023, 4, 21), IsAdmin =
false },
    new Persoon { Id = 12, Voornaam = "Sophie", Achternaam = "Laethem", Login =
"sophie", Paswoord = "a98ec5c5044800c88e862f007b98d89815fc40ca155d6ce7909530d792e909ce",
Profielfoto = LoadImage("Assets/Profielfotos/f5.png"), RegDatum = new DateTime(2023, 8,
8), IsAdmin = false }
};

public static void InitializeData()
{
    using (SqlConnection conn = new SqlConnection(connString))
    {
        // open connectie
        conn.Open();

        // voer SQL commando uit
        SqlCommand comm = new SqlCommand("SELECT * FROM Person", conn);

        SqlDataReader reader = comm.ExecuteReader();
        if (!reader.Read())
        {
            FillPersons(conn);
        }
    }
}

private static void FillPersons(SqlConnection conn)
{

```

```
foreach (var persoon in list)
{
    using (SqlCommand cmd = new SqlCommand("INSERT INTO Person (id, firstname,
lastname, login, password, regdate, isadmin, profielfoto) VALUES (@Id, @Voornaam,
@Achternaam, @Login, @Paswoord, @RegDatum, @IsAdmin, @Profielfoto)", conn))
    {
        cmd.Parameters.AddWithValue("@Id", persoon.Id);
        cmd.Parameters.AddWithValue("@Voornaam", persoon.Voornaam);
        cmd.Parameters.AddWithValue("@Achternaam", persoon.Achternaam);
        cmd.Parameters.AddWithValue("@Login", persoon.Login);
        cmd.Parameters.AddWithValue("@Paswoord", persoon.Paswoord);
        cmd.Parameters.AddWithValue("@RegDatum", persoon.RegDatum);
        cmd.Parameters.AddWithValue("@IsAdmin", persoon.IsAdmin);
        cmd.Parameters.AddWithValue("@Profielfoto", persoon.Profielfoto ??
(object)DBNull.Value);

        cmd.ExecuteNonQuery();
    }
}

private static byte[] LoadImage(string filePath)
{
    return File.Exists(filePath) ? File.ReadAllBytes(filePath) : null;
}
```

In this code:

- `LoadImage` is a helper method that reads an image from a given file path and returns it as a byte array.
- The `Profielfoto` properties in the `Persoon` objects are set to the appropriate image files.
- The `FillPersons` method inserts the `Persoon` objects into the `Person` table, including the profile pictures.

Make sure the paths to the image files (`Assets/Profielfotos/f1.png`, etc.) are correct and the images exist in those locations.



```
private static byte[] LoadImage(string filePath)
{
    return File.Exists(filePath) ? File.ReadAllBytes(filePath) : null;
}
```

Does not work



If `LoadImage` is not working as expected, let's enhance the method with additional checks and logging to ensure it reads the image files correctly. Here's the improved code with better error handling and a few changes to ensure the images are correctly loaded:

1. Enhanced `LoadImage` Method:

- Include more detailed error handling and logging to understand why the image loading might be failing.

2. Ensure Correct Image Paths:

- Verify that the image paths are correct relative to the application's working directory.

Here is the updated `LoadImage` method and `DataInitializerService` class with improved error handling and logging:

csharp

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.IO;

public class Persoon
{
    public int Id { get; set; }
    public string Voornaam { get; set; }
    public string Achternaam { get; set; }
    public string Login { get; set; }
    public string Paswoord { get; set; }
    public byte[] Profielfoto { get; set; }
    public DateTime RegDatum { get; set; }
    public bool IsAdmin { get; set; }
}

public class DataInitializerService
{
    private static string connString =
    ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;

    private static List<Persoon> list = new List<Persoon>
    {
        new Persoon { Id = 1, Voornaam = "Rogier", Achternaam = "van der Linde", Login =
"rogier", Paswoord = "1b4f0e9851971998e732078544c96b36c3d01cedf7caa332359d6f1d83567014",
Profielfoto = LoadImage("Assets/Profielfotos/m1.png"), RegDatum = new DateTime(2020, 12,
23), IsAdmin = true },
        new Persoon { Id = 2, Voornaam = "Frank", Achternaam = "Salliau", Login = "frank",
Paswoord = "60303ae22b998861bce3b28f33eec1be758a213c86c93c076dbe9f558c11c752", Profielfoto =
LoadImage("Assets/Profielfotos/m2.png"), RegDatum = new DateTime(2021, 3, 3), IsAdmin =
true },
        new Persoon { Id = 3, Voornaam = "Abdelkarim", Achternaam = "Chouikh", Login =
"abdel", Paswoord = "fd61a03af4f77d870fc21e05e7e80678095c92d808cfb3b5c279ee04c74aca13",
Profielfoto = LoadImage("Assets/Profielfotos/m3.png"), RegDatum = new DateTime(2023, 11,
11), IsAdmin = false },
        new Persoon { Id = 4, Voornaam = "Alexandre", Achternaam = "Kamnerdsiri", Login =
"alex", Paswoord = "a4e624d686e03ed2767c0abd85c14426b0b1157d2ce81d27bb4fe4f6f01d688a",
Profielfoto = LoadImage("Assets/Profielfotos/m4.png"), RegDatum = new DateTime(2023, 6,
17), IsAdmin = false },
        new Persoon { Id = 5, Voornaam = "Ayoub", Achternaam = "Aymen", Login = "ayoub",
Paswoord = "a140c0cleda2def2b830363ba362aa4d7d255c262960544821f556e16661b6ff", Profielfoto =
LoadImage("Assets/Profielfotos/m5.png"), RegDatum = new DateTime(2022, 12, 12), IsAdmin =
false },
        new Persoon { Id = 6, Voornaam = "Brian", Achternaam = "Sutanto", Login = "brian",
Paswoord = "ed0cb90bdafa4f93981a7d03cff99213a86aa96a6cbcf89ec5e8889871f088727", Profielfoto =
LoadImage("Assets/Profielfotos/m6.png"), RegDatum = new DateTime(2023, 9, 17), IsAdmin =
false },
        new Persoon { Id = 7, Voornaam = "Marcel", Achternaam = "De Schrijver", Login =
"marcel", Paswoord = "bd7c911264aae15b66d4291b6850829aa96986b1d3ead34d1fdbfef27056c112",
Profielfoto = LoadImage("Assets/Profielfotos/m7.png"), RegDatum = new DateTime(2023, 9,
17), IsAdmin = false },
        new Persoon { Id = 8, Voornaam = "Miranda", Achternaam = "Sousa", Login = "mira",
Paswoord = "1f9bfef15fee8a10c4d0711c7eb0c083962123e1918e461b6a508e7146c189b2", Profielfoto =
LoadImage("Assets/Profielfotos/f1.png"), RegDatum = new DateTime(2024, 1, 11), IsAdmin =
false },
        new Persoon { Id = 9, Voornaam = "Nelly", Achternaam = "Lihmedi", Login = "nelly",
Paswoord = "b4451034d3b6590060ce9484a28b88dd332a80a22ae8e39c9c5cb7357ab26c9f", Profielfoto =
LoadImage("Assets/Profielfotos/f2.png"), RegDatum = new DateTime(2023, 10, 9), IsAdmin =
false },
        new Persoon { Id = 10, Voornaam = "Sandra", Achternaam = "Lauer", Login =
"sandra", Paswoord = "ec2738feb2bbb0bc783eb4667903391416372ba6ed8b8dddbebbdb37e5102473",
Profielfoto = LoadImage("Assets/Profielfotos/f3.png"), RegDatum = new DateTime(2024, 2,
13), IsAdmin = false },
        new Persoon { Id = 11, Voornaam = "Soo", Achternaam = "Wang", Login = "soo",
Paswoord = "744ea9ec6fa0a83e9764b4e323d5be6b55a5accfc7fe4c08eab6a8delfca4855", Profielfoto
```

```

= LoadImage("Assets/Profielfotos/f4.png"), RegDatum = new DateTime(2023, 4, 21), IsAdmin =
false },
    new Persoon { Id = 12, Voornaam = "Sophie", Achternaam = "Laethem", Login =
"sophie", Paswoord = "a98ec5c5044800c88e862f007b98d89815fc40ca155d6ce7909530d792e909ce",
Profielfoto = LoadImage("Assets/Profielfotos/f5.png"), RegDatum = new DateTime(2023, 8,
8), IsAdmin = false }
};

public static void InitializeData()
{
    using (SqlConnection conn = new SqlConnection(connString))
    {
        // open connectie
        conn.Open();

        // voer SQL commando uit
        SqlCommand comm = new SqlCommand("SELECT * FROM Person", conn);

        SqlDataReader reader = comm.ExecuteReader();
        if (!reader.Read())
        {
            reader.Close(); // Close the reader before inserting new data
            FillPersons(conn);
        }
    }
}

private static void FillPersons(SqlConnection conn)
{
    foreach (var persoon in list)
    {
        using (SqlCommand cmd = new SqlCommand("INSERT INTO Person (id, firstname,
lastname, login, password, regdate, isadmin, profielfoto) VALUES (@Id, @Voornaam,
@Achternaam, @Login, @Paswoord, @RegDatum, @IsAdmin, @Profielfoto)", conn))
        {
            cmd.Parameters.AddWithValue("@Id", persoon.Id);
            cmd.Parameters.AddWithValue("@Voornaam", persoon.Voornaam);
            cmd.Parameters.AddWithValue("@Achternaam", persoon.Achternaam);
            cmd.Parameters.AddWithValue("@Login", persoon.Login);
            cmd.Parameters.AddWithValue("@Paswoord", persoon.Paswoord);
            cmd.Parameters.AddWithValue("@RegDatum", persoon.RegDatum);
            cmd.Parameters.AddWithValue("@IsAdmin", persoon.IsAdmin);
            cmd.Parameters.AddWithValue("@Profielfoto", persoon.Profielfoto ??
(object)DBNull.Value);

            cmd.ExecuteNonQuery();
        }
    }
}

private static byte[] LoadImage(string filePath)
{
    try
    {
        if (File.Exists(filePath))
        {
            return File.ReadAllBytes(filePath);
        }
        else
        {
            Console.WriteLine($"File not found: {filePath}");
            return null;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error loading image from {filePath}: {ex.Message}");
        return null;
    }
}
}

```


Key Points:

1. Error Handling in `LoadImage`:

- The method now catches exceptions and logs them.
- It logs a message if the file does not exist.

2. Closing the Reader:

- The `SqlDataReader` is closed before attempting to insert new data.

3. File Paths:

- Ensure that the file paths to the images are correct relative to the application's working directory.

By adding these improvements, you should be able to diagnose why `LoadImage` might not be working correctly and ensure that images are loaded and inserted into the database.



file.Exists gives always false while image exists in Assets/Profielfotos the extension is .jpg



If `File.Exists` is always returning false even though the files exist, it's likely an issue with the file path. Here are some steps to troubleshoot and fix this:

1. Check Current Directory:

- Print the current working directory to verify where the application is looking for the files.
- Adjust the file paths relative to this directory.

2. Correct File Paths:

- Ensure the paths are correctly formed based on the current directory.

Here's an updated version of the `LoadImage` method and some logging to print the current directory:

csharp

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.IO;

public class Persoon
{
    public int Id { get; set; }
    public string Voornaam { get; set; }
    public string Achternaam { get; set; }
    public string Login { get; set; }
    public string Paswoord { get; set; }
    public byte[] Profielfoto { get; set; }
    public DateTime RegDatum { get; set; }
    public bool IsAdmin { get; set; }
}

public class DataInitializerService
{
    private static string connString =
        ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;

    private static List<Persoon> list = new List<Persoon>
    {
```

```

        new Persoon { Id = 1, Voornaam = "Rogier", Achternaam = "van der Linde", Login =
"rogier", Paswoord = "1b4f0e9851971998e732078544c96b36c3d01cedf7caa332359d6f1d83567014",
Profielfoto = LoadImage("Assets/Profielfotos/m1.jpg"), RegDatum = new DateTime(2020, 12,
23), IsAdmin = true },
        new Persoon { Id = 2, Voornaam = "Frank", Achternaam = "Salliau", Login = "frank",
Paswoord = "60303ae22b998861bce3b28f33eec1be758a213c86c93c076dbe9f558c11c752", Profielfoto
= LoadImage("Assets/Profielfotos/m2.jpg"), RegDatum = new DateTime(2021, 3, 3), IsAdmin =
true },
        new Persoon { Id = 3, Voornaam = "Abdelkarim", Achternaam = "Chouikh", Login =
"abdel", Paswoord = "fd61a03af4f77d870fc21e05e7e80678095c92d808cfb3b5c279ee04c74aca13",
Profielfoto = LoadImage("Assets/Profielfotos/m3.jpg"), RegDatum = new DateTime(2023, 11,
11), IsAdmin = false },
        new Persoon { Id = 4, Voornaam = "Alexandre", Achternaam = "Kammerdsiri", Login =
"alex", Paswoord = "a4e624d686e03ed2767c0abd85c14426b0b1157d2ce81d27bb4fe4f6f01d688a",
Profielfoto = LoadImage("Assets/Profielfotos/m4.jpg"), RegDatum = new DateTime(2023, 6,
17), IsAdmin = false },
        new Persoon { Id = 5, Voornaam = "Ayoub", Achternaam = "Aymen", Login = "ayoub",
Paswoord = "a140c0c1eda2def2b830363ba362aa4d7d255c262960544821f556e16661b6ff", Profielfoto
= LoadImage("Assets/Profielfotos/m5.jpg"), RegDatum = new DateTime(2022, 12, 12), IsAdmin
= false },
        new Persoon { Id = 6, Voornaam = "Brian", Achternaam = "Sutanto", Login = "brian",
Paswoord = "ed0cb90bdfa4f93981a7d03cff99213a86aa96a6cbcf89ec5e889871f088727", Profielfoto
= LoadImage("Assets/Profielfotos/m6.jpg"), RegDatum = new DateTime(2023, 9, 17), IsAdmin =
false },
        new Persoon { Id = 7, Voornaam = "Marcel", Achternaam = "De Schrijver", Login =
"marcel", Paswoord = "bd7c911264aae15b66d4291b6850829aa96986b1d3ead34d1fdbfefd27056c112",
Profielfoto = LoadImage("Assets/Profielfotos/m7.jpg"), RegDatum = new DateTime(2023, 9,
17), IsAdmin = false },
        new Persoon { Id = 8, Voornaam = "Miranda", Achternaam = "Sousa", Login = "mira",
Paswoord = "1f9bfeb15fee8a10c4d0711c7eb0c083962123e1918e461b6a508e7146c189b2", Profielfoto
= LoadImage("Assets/Profielfotos/f1.jpg"), RegDatum = new DateTime(2024, 1, 11), IsAdmin =
false },
        new Persoon { Id = 9, Voornaam = "Nelly", Achternaam = "Lihmedi", Login = "nelly",
Paswoord = "b4451034d3b6590060ce9484a28b88dd332a80a22ae8e39c9c5cb7357ab26c9f", Profielfoto
= LoadImage("Assets/Profielfotos/f2.jpg"), RegDatum = new DateTime(2023, 10, 9), IsAdmin =
false },
        new Persoon { Id = 10, Voornaam = "Sandra", Achternaam = "Lauer", Login =
"sandra", Paswoord = "ec2738feb2bbb0bc783eb4667903391416372ba6ed8b8dddbebbdb37e5102473",
Profielfoto = LoadImage("Assets/Profielfotos/f3.jpg"), RegDatum = new DateTime(2024, 2,
13), IsAdmin = false },
        new Persoon { Id = 11, Voornaam = "Soo", Achternaam = "Wang", Login = "soo",
Paswoord = "744ea9ec6fa0a83e9764b4e323d5be6b55a5accfc7fe4c08eab6a8delfca4855", Profielfoto
= LoadImage("Assets/Profielfotos/f4.jpg"), RegDatum = new DateTime(2023, 4, 21), IsAdmin =
false },
        new Persoon { Id = 12, Voornaam = "Sophie", Achternaam = "Laethem", Login =
"sophie", Paswoord = "a98ec5c5044800c88e862f007b98d89815fc40ca155d6ce7909530d792e909ce",
Profielfoto = LoadImage("Assets/Profielfotos/f5.jpg"), RegDatum = new DateTime(2023, 8,
8), IsAdmin = false }
    };

    public static void InitializeData()
    {
        using (SqlConnection conn = new SqlConnection(connString))
        {
            // open connectie
            conn.Open();

            // voer SQL commando uit
            SqlCommand comm = new SqlCommand("SELECT * FROM Person", conn);

            SqlDataReader reader = comm.ExecuteReader();
            if (!reader.Read())
            {
                reader.Close(); // Close the reader before inserting new data
                FillPersons(conn);
            }
        }
    }

    private static void FillPersons(SqlConnection conn)
    {
        foreach (var persoon in list)
        {
            using (SqlCommand cmd = new SqlCommand("INSERT INTO Person (id, firstname,
lastname, login, password, regdate, isadmin, profielfoto) VALUES (@Id, @Voornaam,

```



```
@Achternaam, @Login, @Paswoord, @RegDatum, @IsAdmin, @Profielfoto)", conn))
{
    cmd.Parameters.AddWithValue("@Id", persoon.Id);
    cmd.Parameters.AddWithValue("@Voornaam", persoon.Voornaam);
    cmd.Parameters.AddWithValue("@Achternaam", persoon.Achternaam);
    cmd.Parameters.AddWithValue("@Login", persoon.Login);
    cmd.Parameters.AddWithValue("@Paswoord", persoon.Paswoord);
    cmd.Parameters.AddWithValue("@RegDatum", persoon.RegDatum);
    cmd.Parameters.AddWithValue("@IsAdmin", persoon.IsAdmin);
    cmd.Parameters.AddWithValue("@Profielfoto", persoon.Profielfoto ??
(object)DBNull.Value);

    cmd.ExecuteNonQuery();
}
}
}

private static byte[] LoadImage(string filePath)
{
    try
    {
        string currentDir = Directory.GetCurrentDirectory();
        Console.WriteLine($"Current Directory: {currentDir}");
        string fullPath = Path.Combine(currentDir, filePath);

        if (File.Exists(fullPath))
        {
            return File.ReadAllBytes(fullPath);
        }
        else
        {
            Console.WriteLine($"File not found: {fullPath}");
            return null;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error loading image from {filePath}: {ex.Message}");
        return null;
    }
}
```

Key Changes:

1. **Logging the Current Directory:** Prints the current working directory to ensure the path is correct.
2. **Combining Paths:** Uses `Path.Combine` to create the full path to the image files.

Run the application and check the console output to verify the current directory and see if the paths to your image files are correctly formed. This should help identify why `File.Exists` is returning false.



```
private static byte[] LoadImage(string filePath)
{
    var uriSource = new Uri(filePath, UriKind.RelativeOrAbsolute);

    if (File.Exists())
    {
    }

    return File.Exists(filePath) ? File.ReadAllBytes(filePath) : null;
}
```

```
}
}
```

Use urisource



To use `Uri` for resolving the file paths, you'll need to ensure that the path is correctly interpreted as a relative path to your project directory. The `Uri` object can help create an absolute path from a relative one, which you can then check with `File.Exists` and read the file if it exists. Here's how you can modify the `LoadImage` method to use `Uri`:

csharp

```
private static byte[] LoadImage(string filePath)
{
    try
    {
        Uri uriSource = new Uri(filePath, UriKind.RelativeOrAbsolute);

        // If the path is relative, combine it with the current directory
        if (!uriSource.IsAbsoluteUri)
        {
            string currentDir = Directory.GetCurrentDirectory();
            string absolutePath = Path.Combine(currentDir, uriSource.ToString());
            uriSource = new Uri(absolutePath, UriKind.Absolute);
        }

        string absoluteFilePath = uriSource.LocalPath;

        // Check if the file exists at the absolute path
        if (File.Exists(absoluteFilePath))
        {
            return File.ReadAllBytes(absoluteFilePath);
        }
        else
        {
            Console.WriteLine($"File not found: {absoluteFilePath}");
            return null;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error loading image from {filePath}: {ex.Message}");
        return null;
    }
}
```

Explanation:

1. **Uri Creation:** The `Uri` object is created with `UriKind.RelativeOrAbsolute`, so it will interpret the path as either relative or absolute.
2. **Handling Relative Paths:** If the `Uri` is not absolute, it combines the current directory with the provided path to create an absolute path.
3. **File Existence Check:** It uses the `LocalPath` property of the `Uri` object to get the absolute file path and checks if the file exists.
4. **Reading the File:** If the file exists, it reads and returns the file bytes; otherwise, it logs an error.

Ensure your image files are correctly located relative to your project's current directory when running the application. This code will help resolve the file paths and load the images as needed.



Tabel Activiteit: voorzie minstens 30 activiteiten van verschillende soorten (sport, cultuur en hobby); beperk je om het verbeteren makkelijker te maken tot activiteiten in augustus 2024 en verspreid binnen het Brussels Gewest

```
public class CultuurActiviteit : Activiteit
{
    public ActiviteitSector Sector { get; set; }
}
```

```
public enum ActiviteitSector
{
    Nvt,
    Muziek,
    Theater,
    Dans,
    Andere
}
```

```
namespace CLActiBuddy
{
    public abstract class Activiteit
    {
        public int Id { get; set; }
        public string Titel { get; set; }
        public string Beschrijving { get; set; }
        public DateTime DatumTijd { get; set; }
        public byte[] Icoon { get; set; }
        public decimal Longitude { get; set; }
        public decimal Latitude { get; set; }
        public int MaxPersonen { get; set; }
        public ActiviteitSoort Soort { get; set; }
        public int Leeftijdsgroep { get; set; }
        public int OrganisatorId { get; set; }
    }
}
```

```
public enum ActiviteitSoort
{
    Sport = 1,
    Cultuur = 2,
    Hobby = 3
}
```

```
namespace CLActiBuddy
{
```

```
public class HobbyActiviteit : Activiteit
{
    public ActiviteitNiveau Niveau { get; set; }
}

public enum ActiviteitNiveau
{
    Nvt,
    Beginner,
    Halfgevorderd,
    Gevorderd
}

namespace CLActiBuddy
{
    public class SportActiviteit : Activiteit
    {
        public ActiviteitMoeilijkheid Moeilijkheid { get; set; }
    }

    public enum ActiviteitMoeilijkheid
    {
        Nvt,
        Makkelijk,
        Gemiddeld,
        Zwaar
    }
}
```

Geef me een List<Activiteiten>



Hier is een lijst van 30 activiteiten in augustus 2024, verdeeld over de drie types: sport, cultuur, en hobby. Deze activiteiten vinden plaats in het Brussels Gewest. De details zijn fictief, maar geven een representatieve spreiding en variëteit binnen de gevraagde parameters.

Lijst van Activiteiten

csharp

```
using System;
using System.Collections.Generic;

namespace CLActiBuddy
{
    public static class ActiviteitenInitializer
    {
        public static List<Activiteit> GetActiviteiten()
        {
            return new List<Activiteit>
```

```

{
    // Sportactiviteiten
    new SportActiviteit
    {
        Id = 1,
        Titel = "Zomerse Hardloopwedstrijd",
        Beschrijving = "Een hardloopwedstrijd van 10 km door het park.",
        DatumTijd = new DateTime(2024, 8, 1, 9, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport1.png"),
        Longitude = 4.3517m,
        Latitude = 50.8467m,
        MaxPersonen = 100,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 18,
        OrganisatorId = 1,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Id = 2,
        Titel = "Fietsrit in Brussel",
        Beschrijving = "Fietsrit van 20 km langs de mooiste plekken in
Brussel.",
        DatumTijd = new DateTime(2024, 8, 5, 10, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport2.png"),
        Longitude = 4.3488m,
        Latitude = 50.8467m,
        MaxPersonen = 50,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 15,
        OrganisatorId = 2,
        Moeilijkheid = ActiviteitMoeilijkheid.Makkelijk
    },
    new SportActiviteit
    {
        Id = 3,
        Titel = "Vriendschappelijke Voetbalwedstrijd",
        Beschrijving = "Een voetbalwedstrijd tussen lokale teams.",
        DatumTijd = new DateTime(2024, 8, 8, 17, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport3.png"),
        Longitude = 4.3556m,
        Latitude = 50.8494m,
        MaxPersonen = 200,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 12,
        OrganisatorId = 3,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Id = 4,
        Titel = "Yoga in het Park",
        Beschrijving = "Gratis yoga sessie in het stadspark.",
        DatumTijd = new DateTime(2024, 8, 12, 8, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport4.png"),
        Longitude = 4.3489m,
        Latitude = 50.8460m,
        MaxPersonen = 30,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 18,
        OrganisatorId = 4,
        Moeilijkheid = ActiviteitMoeilijkheid.Makkelijk
    },
    new SportActiviteit
    {
        Id = 5,
        Titel = "Basketbaltoernooi",
        Beschrijving = "Toernooi voor basketbalteams van verschillende
niveaus.",
        DatumTijd = new DateTime(2024, 8, 15, 14, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport5.png"),
        Longitude = 4.3484m,
        Latitude = 50.8461m,
        MaxPersonen = 50,
        Soort = ActiviteitSoort.Sport,
    }
}

```

```

        Leeftijdsgroep = 16,
        OrganisatorId = 5,
        Moeilijkheid = ActiviteitMoeilijkheid.Zwaar
    },

    // Cultuuractiviteiten
    new CultuurActiviteit
    {
        Id = 6,
        Titel = "Brusselse Muziekavond",
        Beschrijving = "Een avond met lokale muzikanten in de
stadsschouwburg.",
        DatumTijd = new DateTime(2024, 8, 2, 19, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur1.png"),
        Longitude = 4.3499m,
        Latitude = 50.8466m,
        MaxPersonen = 150,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 18,
        OrganisatorId = 6,
        Sector = ActiviteitSector.Muziek
    },
    new CultuurActiviteit
    {
        Id = 7,
        Titel = "Theaterfestival",
        Beschrijving = "Een festival met verschillende toneelstukken in de
open lucht.",
        DatumTijd = new DateTime(2024, 8, 6, 20, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur2.png"),
        Longitude = 4.3480m,
        Latitude = 50.8440m,
        MaxPersonen = 200,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        OrganisatorId = 7,
        Sector = ActiviteitSector.Theater
    },
    new CultuurActiviteit
    {
        Id = 8,
        Titel = "Dansvoorstelling",
        Beschrijving = "Professionele dansgroepen tonen hun beste nummers.",
        DatumTijd = new DateTime(2024, 8, 10, 18, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur3.png"),
        Longitude = 4.3556m,
        Latitude = 50.8494m,
        MaxPersonen = 100,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 16,
        OrganisatorId = 8,
        Sector = ActiviteitSector.Dans
    },
    new CultuurActiviteit
    {
        Id = 9,
        Titel = "Kunstexpositie",
        Beschrijving = "Expositie van lokale kunstenaars in het
stadscentrum.",
        DatumTijd = new DateTime(2024, 8, 13, 10, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur4.png"),
        Longitude = 4.3550m,
        Latitude = 50.8470m,
        MaxPersonen = 80,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        OrganisatorId = 9,
        Sector = ActiviteitSector.Ander
    },
    new CultuurActiviteit
    {
        Id = 10,
        Titel = "Filmavond in het Park",
        Beschrijving = "Buitenfilmvertoning met een klassieker in het
stadspark.",

```

```

        DatumTijd = new DateTime(2024, 8, 18, 21, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur5.png"),
        Longitude = 4.3545m,
        Latitude = 50.8480m,
        MaxPersonen = 300,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 0,
        OrganisatorId = 10,
        Sector = ActiviteitSector.Ander
    },

    // Hobbyactiviteiten
    new HobbyActiviteit
    {
        Id = 11,
        Titel = "Schilderworkshop",
        Beschrijving = "Workshop voor beginners en gevorderden.",
        DatumTijd = new DateTime(2024, 8, 3, 14, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby1.png"),
        Longitude = 4.3491m,
        Latitude = 50.8468m,
        MaxPersonen = 25,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 15,
        OrganisatorId = 11,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Id = 12,
        Titel = "Fotografie Cursus",
        Beschrijving = "Leer de basisprincipes van fotografie.",
        DatumTijd = new DateTime(2024, 8, 7, 11, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby2.png"),
        Longitude = 4.3502m,
        Latitude = 50.8471m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 18,
        OrganisatorId = 12,
        Niveau = ActiviteitNiveau.Halfgevoorderd
    },
    new HobbyActiviteit
    {
        Id = 13,
        Titel = "Tuinaanleg Workshop",
        Beschrijving = "Leer de basis van tuinonderhoud en -ontwerp.",
        DatumTijd = new DateTime(2024, 8, 9, 10, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby3.png"),
        Longitude = 4.3512m,
        Latitude = 50.8462m,
        MaxPersonen = 15,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 20,
        OrganisatorId = 13,
        Niveau = ActiviteitNiveau.Gevorderd
    },
    new HobbyActiviteit
    {
        Id = 14,
        Titel = "Creatieve Schrijfavond",
        Beschrijving = "Workshop voor creatieve schrijvers.",
        DatumTijd = new DateTime(2024, 8, 11, 18, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby4.png"),
        Longitude = 4.3522m,
        Latitude = 50.8463m,
        MaxPersonen = 30,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 18,
        OrganisatorId = 14,
        Niveau = ActiviteitNiveau.Halfgevoorderd
    },
    new HobbyActiviteit
    {
        Id = 15,

```

```

        Titel = "Kookcursus",
        Beschrijving = "Leer de kunst van het koken met een chef-kok.",
        DatumTijd = new DateTime(2024, 8, 20, 16, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby5.png"),
        Longitude = 4.3532m,
        Latitude = 50.8474m,
        MaxPersonen = 12,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 21,
        OrganisatorId = 15,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Id = 16,
        Titel = "Breien voor Beginners",
        Beschrijving = "Leer breien met eenvoudige patronen.",
        DatumTijd = new DateTime(2024, 8, 22, 13, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby6.png"),
        Longitude = 4.3504m,
        Latitude = 50.8465m,
        MaxPersonen = 10,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 15,
        OrganisatorId = 16,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Id = 17,
        Titel = "Handgemaakte Kaarten",
        Beschrijving = "Workshop voor het maken van handgemaakte kaarten.",
        DatumTijd = new DateTime(2024, 8, 25, 11, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby7.png"),
        Longitude = 4.3516m,
        Latitude = 50.8476m,
        MaxPersonen = 15,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 16,
        OrganisatorId = 17,
        Niveau = ActiviteitNiveau.Halfgevoorderd
    },
    new HobbyActiviteit
    {
        Id = 18,
        Titel = "Muziekproductie Workshop",
        Beschrijving = "Leer muziek maken met digitale tools.",
        DatumTijd = new DateTime(2024, 8, 28, 15, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby8.png"),
        Longitude = 4.3520m,
        Latitude = 50.8469m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 18,
        OrganisatorId = 18,
        Niveau = ActiviteitNiveau.Gevorderd
    },
    new HobbyActiviteit
    {
        Id = 19,
        Titel = "Origami Workshop",
        Beschrijving = "Leer de kunst van origami met diverse ontwerpen.",
        DatumTijd = new DateTime(2024, 8, 30, 14, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby9.png"),
        Longitude = 4.3530m,
        Latitude = 50.8478m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 12,
        OrganisatorId = 19,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Id = 20,

```



```

        Titel = "Creatief Schrijven",
        Beschrijving = "Workshop voor het verbeteren van je
schrijfvaardigheden.",
        DatumTijd = new DateTime(2024, 8, 31, 16, 0, 0),
        Icoon = LoadImage("Assets/Icons/hobby10.png"),
        Longitude = 4.3545m,
        Latitude = 50.8480m,
        MaxPersonen = 25,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 18,
        OrganisatorId = 20,
        Niveau = ActiviteitNiveau.Halfgevorderd
    },

    // Meer Sportactiviteiten
    new SportActiviteit
    {
        Id = 21,
        Titel = "Zwemwedstrijd",
        Beschrijving = "Competitie voor verschillende zwemstijlen.",
        DatumTijd = new DateTime(2024, 8, 4, 11, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport6.png"),
        Longitude = 4.3495m,
        Latitude = 50.8473m,
        MaxPersonen = 80,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 14,
        OrganisatorId = 21,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Id = 22,
        Titel = "Tafeltennis Toernooi",
        Beschrijving = "Toernooi voor amateur tafeltennissers.",
        DatumTijd = new DateTime(2024, 8, 7, 12, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport7.png"),
        Longitude = 4.3507m,
        Latitude = 50.8464m,
        MaxPersonen = 40,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 16,
        OrganisatorId = 22,
        Moeilijkheid = ActiviteitMoeilijkheid.Makkelijk
    },
    new SportActiviteit
    {
        Id = 23,
        Titel = "Wielervedstrijd",
        Beschrijving = "Een spannende wielervedstrijd door het centrum van
Brussel.",
        DatumTijd = new DateTime(2024, 8, 11, 9, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport8.png"),
        Longitude = 4.3489m,
        Latitude = 50.8462m,
        MaxPersonen = 60,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 18,
        OrganisatorId = 23,
        Moeilijkheid = ActiviteitMoeilijkheid.Zwaar
    },
    new SportActiviteit
    {
        Id = 24,
        Titel = "Klimwedstrijd",
        Beschrijving = "Klimcompetitie op een klimwand.",
        DatumTijd = new DateTime(2024, 8, 15, 13, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport9.png"),
        Longitude = 4.3521m,
        Latitude = 50.8472m,
        MaxPersonen = 30,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 16,
        OrganisatorId = 24,
        Moeilijkheid = ActiviteitMoeilijkheid.Gevorderd
    }
}

```

```

    },
    new SportActiviteit
    {
        Id = 25,
        Titel = "Tennis Toernooi",
        Beschrijving = "Toernooi voor tennisfanaten.",
        DatumTijd = new DateTime(2024, 8, 20, 10, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport10.png"),
        Longitude = 4.3515m,
        Latitude = 50.8469m,
        MaxPersonen = 50,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 16,
        OrganisatorId = 25,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },

    // Meer Cultuuractiviteiten
    new CultuurActiviteit
    {
        Id = 26,
        Titel = "Jazz Festival",
        Beschrijving = "Een festival met jazzmuziek van lokale en
internationale artiesten.",
        DatumTijd = new DateTime(2024, 8, 3, 20, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur6.png"),
        Longitude = 4.3550m,
        Latitude = 50.8471m,
        MaxPersonen = 150,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 18,
        OrganisatorId = 26,
        Sector = ActiviteitSector.Muziek
    },
    new CultuurActiviteit
    {
        Id = 27,
        Titel = "Openlucht Theater",
        Beschrijving = "Een openlucht theaterstuk in de tuin van het museum.",
        DatumTijd = new DateTime(2024, 8, 9, 19, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur7.png"),
        Longitude = 4.3560m,
        Latitude = 50.8465m,
        MaxPersonen = 100,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        OrganisatorId = 27,
        Sector = ActiviteitSector.Theater
    },
    new CultuurActiviteit
    {
        Id = 28,
        Titel = "Moderne Dans Workshop",
        Beschrijving = "Workshop moderne dans voor alle niveaus.",
        DatumTijd = new DateTime(2024, 8, 12, 15, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur8.png"),
        Longitude = 4.3545m,
        Latitude = 50.8468m,
        MaxPersonen = 25,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 16,
        OrganisatorId = 28,
        Sector = ActiviteitSector.Dans
    },
    new CultuurActiviteit
    {
        Id = 29,
        Titel = "Kunstmarkt",
        Beschrijving = "Markt met kunst en ambachten van lokale kunstenaars.",
        DatumTijd = new DateTime(2024, 8, 18, 10, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur9.png"),
        Longitude = 4.3557m,
        Latitude = 50.8472m,
        MaxPersonen = 200,
        Soort = ActiviteitSoort.Cultuur,
    }
}

```

```

        Leeftijdsgroep = 0,
        OrganisatorId = 29,
        Sector = ActiviteitSector.Ander
    },
    new CultuurActiviteit
    {
        Id = 30,
        Titel = "Theaterimprovisatie",
        Beschrijving = "Improviseer en speel mee in een theaterstuk.",
        DatumTijd = new DateTime(2024, 8, 25, 17, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur10.png"),
        Longitude = 4.3567m,
        Latitude = 50.8475m,
        MaxPersonen = 50,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        OrganisatorId = 30,
        Sector = ActiviteitSector.Theater
    }
};

private static byte[] LoadImage(string filePath)
{
    // Deze methode zou de afbeelding laden van het bestandssysteem, maar is hier
    een placeholder.
    return new byte[0]; // Placeholder; vervang dit met de echte implementatie.
}
}

```

Toelichting

1. **Activiteiten Types:** De lijst bevat activiteiten verdeeld over `Sport`, `Cultuur`, en `Hobby`.
2. **DatumTijd:** De activiteiten zijn gespreid over augustus 2024.
3. **Locaties:** De coördinaten zijn representatief voor locaties in het Brussels Gewest.
4. **Icoon:** `LoadImage` is een placeholder en zou in een echte implementatie de iconen voor de activiteiten laden.

Pas de coördinaten en iconen aan volgens de specifieke locaties en bronnen die je hebt.



Maak nu een FillActiviteiten netals FillPerosns

```

private static void FillActiviteiten()
{
    throw new NotImplementedException();
}

private static void FillPersons()
{
    foreach (var persoon in personen)
    {
        using (SqlConnection conn = new SqlConnection(connString))
        {
            using (SqlCommand cmd = new SqlCommand("INSERT INTO Person (id, firstname,
            lastname, login, password, regdate, isadmin, profielfoto) VALUES (@Id, @Voornaam,
            @Achternaam, @Login, @Paswoord, @RegDatum, @IsAdmin, @Profielfoto)", conn))

```

```

        {
            cmd.Parameters.AddWithValue("@Id", persoon.Id);
            cmd.Parameters.AddWithValue("@Voornaam", persoon.Voornaam);
            cmd.Parameters.AddWithValue("@Achternaam", persoon.Achternaam);
            cmd.Parameters.AddWithValue("@Login", persoon.Login);
            cmd.Parameters.AddWithValue("@Paswoord", persoon.Paswoord);
            cmd.Parameters.AddWithValue("@RegDatum", persoon.RegDatum);
            cmd.Parameters.AddWithValue("@IsAdmin", persoon.IsAdmin);
            cmd.Parameters.AddWithValue("@Profielfoto", persoon.Profielfoto ??
(object)DBNull.Value);

            cmd.ExecuteNonQuery();
        }
    }
}

```



Zeker! Hier is hoe je de `FillActiviteiten` methode kunt implementeren, gebaseerd op de structuur van `FillPersons` die je hebt gegeven:

Implementatie van `FillActiviteiten`

1. **Verbind met de Database:** Maak een verbinding met de database, net zoals in `FillPersons`.
2. **Voeg Activiteiten Toe:** Gebruik een `SqlCommand` om activiteiten toe te voegen aan de database.
3. **Parameter Binding:** Bind de parameters voor elke activiteit om SQL-injecties te voorkomen.

Hier is de complete implementatie:

csharp

```

private static void FillActiviteiten()
{
    // Verzameling van activiteiten, dit zou normaal vanuit een database of een andere
    bron worden geladen.
    var activiteiten = new List<Activiteit>
    {
        new CultuurActiviteit
        {
            Id = 26,
            Titel = "Jazz Festival",
            Beschrijving = "Een festival met jazzmuziek van lokale en internationale
artiesten.",
            DatumTijd = new DateTime(2024, 8, 3, 20, 0, 0),
            Icoon = LoadImage("Assets/Icons/cultuur6.png"),
            Longitude = 4.3550m,
            Latitude = 50.8471m,
            MaxPersonen = 150,
            Soort = ActiviteitSoort.Cultuur,
            Leeftijdsgroep = 18,
            OrganisatorId = 26,
            Sector = ActiviteitSector.Muziek
        },
        new CultuurActiviteit
        {
            Id = 27,
            Titel = "Openlucht Theater",
            Beschrijving = "Een openlucht theaterstuk in de tuin van het museum.",

```

```

        DatumTijd = new DateTime(2024, 8, 9, 19, 0, 0),
        Icoon = LoadImage("Assets/Icons/cultuur7.png"),
        Longitude = 4.3560m,
        Latitude = 50.8465m,
        MaxPersonen = 100,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        OrganisatorId = 27,
        Sector = ActiviteitSector.Theater
    },
    // Voeg hier de andere activiteiten toe...
    // Hier een paar extra om het geheel te voltooien
    new SportActiviteit
    {
        Id = 21,
        Titel = "Zwemwedstrijd",
        Beschrijving = "Competitie voor verschillende zwemstijlen.",
        DatumTijd = new DateTime(2024, 8, 4, 11, 0, 0),
        Icoon = LoadImage("Assets/Icons/sport6.png"),
        Longitude = 4.3495m,
        Latitude = 50.8473m,
        MaxPersonen = 80,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 14,
        OrganisatorId = 21,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    // Voeg hier nog meer activiteiten toe indien nodig
};

foreach (var activiteit in activiteiten)
{
    using (SqlConnection conn = new SqlConnection(connString))
    {
        conn.Open();

        // Bepaal het juiste SQL-commando op basis van het type activiteit
        string sql = activiteit is CultuurActiviteit ?
            "INSERT INTO CultuurActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Sector) VALUES (@Id, @Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Sector)" :
            activiteit is SportActiviteit ?
            "INSERT INTO SportActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Moeilijkheid) VALUES (@Id, @Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Moeilijkheid)" :
            "INSERT INTO HobbyActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Niveau) VALUES (@Id, @Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Niveau)";

        using (SqlCommand cmd = new SqlCommand(sql, conn))
        {
            cmd.Parameters.AddWithValue("@Id", activiteit.Id);
            cmd.Parameters.AddWithValue("@Titel", activiteit.Titel);
            cmd.Parameters.AddWithValue("@Beschrijving", activiteit.Beschrijving);
            cmd.Parameters.AddWithValue("@DatumTijd", activiteit.DatumTijd);
            cmd.Parameters.AddWithValue("@Icoon", activiteit.Icoon ??
(object)DBNull.Value);
            cmd.Parameters.AddWithValue("@Longitude", activiteit.Longitude);
            cmd.Parameters.AddWithValue("@Latitude", activiteit.Latitude);
            cmd.Parameters.AddWithValue("@MaxPersonen", activiteit.MaxPersonen);
            cmd.Parameters.AddWithValue("@Soort", activiteit.Soort);
            cmd.Parameters.AddWithValue("@Leeftijdsgroep", activiteit.Leeftijdsgroep);
            cmd.Parameters.AddWithValue("@OrganisatorId", activiteit.OrganisatorId);

            // Voeg type-specifieke parameters toe
            if (activiteit is CultuurActiviteit cultuurActiviteit)
            {
                cmd.Parameters.AddWithValue("@Sector", cultuurActiviteit.Sector);
            }
            else if (activiteit is SportActiviteit sportActiviteit)
            {
                cmd.Parameters.AddWithValue("@Moeilijkheid",

```

```

sportActiviteit.Moeilijkheid);
    }
    else if (activiteit is HobbyActiviteit hobbyActiviteit)
    {
        cmd.Parameters.AddWithValue("@Niveau", hobbyActiviteit.Niveau);
    }

    cmd.ExecuteNonQuery();
}
}
}
}
}

```

Toelichting

1. **Verbindingsbeheer:** Open een nieuwe verbinding voor elke activiteit.
2. **SQL-Commando:** Gebruik een `switch`-achtige structuur om de SQL-commando's aan te passen op basis van het type activiteit.
3. **Parameter Binding:** Voeg parameters toe voor alle gemeenschappelijke velden en specifieke velden op basis van het type activiteit.

Opmerking: Zorg ervoor dat de tabellen `CultuurActiviteit`, `SportActiviteit`, en `HobbyActiviteit` in de database bestaan met de juiste kolommen voor de bijbehorende eigenschappen van de activiteiten. Als deze tabellen nog niet bestaan, moet je ze mogelijk eerst maken of de SQL-commando's aanpassen op basis van je database-schema.



```

string sql = activiteit is CultuurActiviteit ?
    "INSERT INTO CultuurActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude,
    MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Sector) VALUES (@Id, @Titel, @Beschrijving,
    @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep,
    @OrganisatorId, @Sector)" :
    activiteit is SportActiviteit ?
    "INSERT INTO SportActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude,
    MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Moeilijkheid) VALUES (@Id, @Titel,
    @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort,
    @Leeftijdsgroep, @OrganisatorId, @Moeilijkheid)" :
    "INSERT INTO HobbyActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude,
    MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Niveau) VALUES (@Id, @Titel, @Beschrijving,
    @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep,
    @OrganisatorId, @Niveau)";

```

Maak hier een if else van



Zeker, als je liever `if-else` statements gebruikt om het SQL-commando te bepalen, kan de `FillActiviteiten` methode worden herschreven met de `if-else` structuur. Dit is de aangepaste versie:

csharp

```
private static void FillActiviteiten()
{
    var activiteiten = new List<Activiteit>
    {
        // Voeg hier je activiteiten toe, zoals eerder gedefinieerd
    };

    foreach (var activiteit in activiteiten)
    {
        using (SqlConnection conn = new SqlConnection(connString))
        {
            conn.Open();

            string sql = "";
            if (activiteit is CultuurActiviteit)
            {
                sql = "INSERT INTO CultuurActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Sector) VALUES (@Id, @Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Sector)";
            }
            else if (activiteit is SportActiviteit)
            {
                sql = "INSERT INTO SportActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Moeilijkheid) VALUES (@Id, @Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Moeilijkheid)";
            }
            else if (activiteit is HobbyActiviteit)
            {
                sql = "INSERT INTO HobbyActiviteit (Id, Titel, Beschrijving, DatumTijd, Icoon, Longitude, Latitude, MaxPersonen, Soort, Leeftijdsgroep, OrganisatorId, Niveau) VALUES (@Id, @Titel, @Beschrijving, @DatumTijd, @Icoon, @Longitude, @Latitude, @MaxPersonen, @Soort, @Leeftijdsgroep, @OrganisatorId, @Niveau)";
            }
            else
            {
                throw new InvalidOperationException("Onbekend type activiteit.");
            }

            using (SqlCommand cmd = new SqlCommand(sql, conn))
            {
                cmd.Parameters.AddWithValue("@Id", activiteit.Id);
                cmd.Parameters.AddWithValue("@Titel", activiteit.Titel);
                cmd.Parameters.AddWithValue("@Beschrijving", activiteit.Beschrijving);
                cmd.Parameters.AddWithValue("@DatumTijd", activiteit.DatumTijd);
                cmd.Parameters.AddWithValue("@Icoon", activiteit.Icoon ?? (object)DBNull.Value);
                cmd.Parameters.AddWithValue("@Longitude", activiteit.Longitude);
                cmd.Parameters.AddWithValue("@Latitude", activiteit.Latitude);
                cmd.Parameters.AddWithValue("@MaxPersonen", activiteit.MaxPersonen);
                cmd.Parameters.AddWithValue("@Soort", activiteit.Soort);
                cmd.Parameters.AddWithValue("@Leeftijdsgroep", activiteit.Leeftijdsgroep);
                cmd.Parameters.AddWithValue("@OrganisatorId", activiteit.OrganisatorId);

                // Voeg type-specifieke parameters toe
                if (activiteit is CultuurActiviteit cultuurActiviteit)
                {
                    cmd.Parameters.AddWithValue("@Sector", cultuurActiviteit.Sector);
                }
                else if (activiteit is SportActiviteit sportActiviteit)
                {
                    cmd.Parameters.AddWithValue("@Moeilijkheid", sportActiviteit.Moeilijkheid);
                }
                else if (activiteit is HobbyActiviteit hobbyActiviteit)
                {
                    cmd.Parameters.AddWithValue("@Niveau", hobbyActiviteit.Niveau);
                }

                cmd.ExecuteNonQuery();
            }
        }
    }
}
```

```

    }
  }
}

```

Toelichting

1. **`if-else` Structuur:** De **`if-else`** structuur bepaalt welk SQL-commando gebruikt moet worden op basis van het type activiteit (**`CultuurActiviteit`**, **`SportActiviteit`**, of **`HobbyActiviteit`**).
2. **Parameter Binding:** Parameters worden toegevoegd aan het **`SqlCommand`** object op basis van de activiteit. Dit zorgt ervoor dat alle gegevens correct worden ingevoerd in de database.
3. **Foutafhandeling:** Er is een extra controle toegevoegd voor onbekende activiteitentypes, wat helpt om eventuele problemen te identificeren.

Extra Opmerkingen

- Zorg ervoor dat de tabellen **`CultuurActiviteit`**, **`SportActiviteit`**, en **`HobbyActiviteit`** correct zijn gedefinieerd in je database en overeenkomen met de kolommen die je in de **`INSERT`**-commando's hebt gespecificeerd.
- **Database Schema:** Als je problemen ondervindt bij het uitvoeren van deze commando's, controleer dan of je database schema de juiste kolommen en types bevat. Het is ook handig om je database schema te vergelijken met de klassen die je in je code gebruikt.



Tabel Deelname: minstens 50 ingeschreven deelnemers op verschillende activiteiten

```

public class Deelname
{
    public int Id { get; set; }
    public int PersoonId { get; set; }
    public int ActiviteitId { get; set; }
}

```

Maak ook een Fill Deelnamen met een randomizer



Voor het genereren van een **`FillDeelnamen`** methode die deelnemers willekeurig aan activiteiten koppelt, kun je de volgende stappen volgen. We gebruiken een randomizer om willekeurige combinaties van personen en activiteiten te maken, en voegen vervolgens deze combinaties in de **`Deelname`** tabel in.

Hier is een voorbeeld hoe je dit zou kunnen implementeren:

csharp

```

private static void FillDeelnamen()
{
    // Lijst van alle personen en activiteiten ophalen
    var personen = GetPersonen(); // Methode om alle personen uit de database te halen
}

```



```

var activiteiten = GetActiviteiten(); // Methode om alle activiteiten uit de database
te halen

// Randomizer initialiseren
Random random = new Random();

// Houdt bij welke combinaties al zijn toegevoegd (voor het geval je geen dubbele
deelnames wilt)
HashSet<int persoonId, int activiteitId> bestaandeDeelnamen = new HashSet<int,
int>();

using (SqlConnection conn = new SqlConnection(connString))
{
    conn.Open();

    for (int i = 0; i < 50; i++) // Aantal deelnames
    {
        int persoonId;
        int activiteitId;

        // Kies willekeurig een persoon en een activiteit
        do
        {
            persoonId = personen[random.Next(personen.Count)].Id;
            activiteitId = activiteiten[random.Next(activiteiten.Count)].Id;
        }
        while (bestaandeDeelnamen.Contains((persoonId, activiteitId))); // Voorkom
dubbele deelnames

        bestaandeDeelnamen.Add((persoonId, activiteitId));

        // SQL Command voor het toevoegen van een deelname
        using (SqlCommand cmd = new SqlCommand("INSERT INTO Deelname (Id, PersoonId,
ActiviteitId) VALUES (@Id, @PersoonId, @ActiviteitId)", conn))
        {
            cmd.Parameters.AddWithValue("@Id", i + 1); // Je kunt hier ook een unieke
ID-generatie logica implementeren
            cmd.Parameters.AddWithValue("@PersoonId", persoonId);
            cmd.Parameters.AddWithValue("@ActiviteitId", activiteitId);

            cmd.ExecuteNonQuery();
        }
    }
}

// Methode om alle personen op te halen uit de database
private static List<Persoon> GetPersonen()
{
    var personen = new List<Persoon>();
    using (SqlConnection conn = new SqlConnection(connString))
    {
        conn.Open();
        using (SqlCommand cmd = new SqlCommand("SELECT Id FROM Person", conn))
        {
            using (SqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    personen.Add(new Persoon { Id = reader.GetInt32(0) });
                }
            }
        }
    }
    return personen;
}

// Methode om alle activiteiten op te halen uit de database
private static List<Activiteit> GetActiviteiten()
{
    var activiteiten = new List<Activiteit>();
    using (SqlConnection conn = new SqlConnection(connString))
    {
        conn.Open();
        using (SqlCommand cmd = new SqlCommand("SELECT Id FROM Activiteit", conn))
    }
}

```

```

    {
        using (SqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                activiteiten.Add(new Activiteit { Id = reader.GetInt32(0) });
            }
        }
    }
    return activiteiten;
}

```

Uitleg

1. **Lijst van Personen en Activiteiten:** De methoden `GetPersonen` en `GetActiviteiten` worden gebruikt om lijsten van personen en activiteiten op te halen uit de database.
2. **Randomizer:** We gebruiken een `Random` object om willekeurige keuzes te maken uit de lijsten van personen en activiteiten.
3. **Duplicate Check:** Met behulp van een `HashSet` controleren we of een combinatie van persoon en activiteit al bestaat om dubbele deelnames te voorkomen.
4. **Insert Statement:** Voor elke willekeurige combinatie van persoon en activiteit wordt een record toegevoegd aan de `DeelName` tabel.
5. **Unique ID Generation:** In dit voorbeeld gebruiken we een simpele incrementale ID voor de deelname, maar je kunt ook een andere logica voor unieke ID-generatie implementeren als dat nodig is.

Aandachtspunten

- **Database Schema:** Zorg ervoor dat de `DeelName` tabel correct is gedefinieerd in je database met de juiste kolommen en types.
- **Foutafhandeling:** Overweeg om foutafhandelingsmechanismen toe te voegen om eventuele problemen tijdens het database-insertieproces te beheren.

Met deze aanpak kun je 50 willekeurige deelnames aan je activiteiten toevoegen, met een zekere mate van variatie en zonder dubbele deelnames.



```

new CultuurActiviteit
{
    Titel = "Kunstmarkt",
    Beschrijving = "Markt met kunst en ambachten van lokale kunstenaars.",
    DatumTijd = new DateTime(2024, 8, 18, 10, 0, 0),
    Icoon = LoadImage("Assets/Iconen/cultuur.png"),
    Longitude = 4.3557m,
    Latitude = 50.8472m,
    MaxPersonen = 200,
    Soort = ActiviteitSoort.Cultuur,
    Leeftijdsgroep = 0,
    Sector = ActiviteitSector.Andere
},

```

```

new CultuurActiviteit
{
    Titel = "Theaterimprovisatie",
    Beschrijving = "Improviseer en speel mee in een theaterstuk.",
    DatumTijd = new DateTime(2024, 8, 25, 17, 0, 0),
    Icoon = LoadImage("Assets/Iconen/cultuur.png"),
    Longitude = 4.3567m,
    Latitude = 50.8475m,
    MaxPersonen = 50,
    Soort = ActiviteitSoort.Cultuur,
    Leeftijdsgroep = 12,
    Sector = ActiviteitSector.Theater
}

```

Geef er nog 20



Hier is een lijst van 20 extra activiteiten in augustus 2024, verspreid over verschillende soorten en sectoren, zoals sport, cultuur en hobby. Deze activiteiten zijn ook gespreid binnen het Brussels Gewest.

Cultuur Activiteiten

1. Kunsttentoonstelling

- **Beschrijving:** Expositie van moderne kunstwerken.
- **DatumTijd:** 2024-08-01 11:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3500m
- **Latitude:** 50.8500m
- **MaxPersonen:** 100
- **Leeftijdsgroep:** 0
- **Sector:** Muziek

2. Straatmuziekfestival

- **Beschrijving:** Festival met live straatmuzikanten.
- **DatumTijd:** 2024-08-02 14:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3485m
- **Latitude:** 50.8490m
- **MaxPersonen:** 300
- **Leeftijdsgroep:** 0
- **Sector:** Muziek

3. Fotografie Workshop

- **Beschrijving:** Leer de basis van fotografie in een workshop.
- **DatumTijd:** 2024-08-03 09:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3540m
- **Latitude:** 50.8530m
- **MaxPersonen:** 20
- **Leeftijdsgroep:** 16

- **Sector:** Andere

4. Muzikale Avond

- **Beschrijving:** Geniet van live muziek van lokale bands.
- **DatumTijd:** 2024-08-04 19:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3520m
- **Latitude:** 50.8540m
- **MaxPersonen:** 150
- **Leeftijdsgroep:** 0
- **Sector:** Muziek

5. Dansworkshop

- **Beschrijving:** Leer nieuwe dansmoves in een workshop.
- **DatumTijd:** 2024-08-05 18:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3570m
- **Latitude:** 50.8520m
- **MaxPersonen:** 30
- **Leeftijdsgroep:** 12
- **Sector:** Dans

6. Poëzie Lezing

- **Beschrijving:** Beleef een avond van poëzie door lokale dichters.
- **DatumTijd:** 2024-08-06 20:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3550m
- **Latitude:** 50.8510m
- **MaxPersonen:** 80
- **Leeftijdsgroep:** 0
- **Sector:** Andere

7. Theaterwandeling

- **Beschrijving:** Een interactieve theaterwandeling door de stad.
- **DatumTijd:** 2024-08-07 16:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3560m
- **Latitude:** 50.8505m
- **MaxPersonen:** 40
- **Leeftijdsgroep:** 12
- **Sector:** Theater

8. Kunstworkshop voor Kinderen

- **Beschrijving:** Creatieve kunstactiviteiten voor kinderen.
- **DatumTijd:** 2024-08-08 10:00
- **Icoon:** `Assets/Iconen/cultuur.png`
- **Longitude:** 4.3580m
- **Latitude:** 50.8525m
- **MaxPersonen:** 25
- **Leeftijdsgroep:** 6
- **Sector:** Andere

Sport Activiteiten

9. Marathon van Brussel

- **Beschrijving:** Een marathon door de stad Brussel.
- **DatumTijd:** 2024-08-09 07:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3550m
- **Latitude:** 50.8500m
- **MaxPersonen:** 1000
- **Leeftijdsgroep:** 18
- **Moeilijkheid:** Zwaar

10. Zumba les in het Park

- **Beschrijving:** Zumba les in een lokaal park.
- **DatumTijd:** 2024-08-10 17:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3515m
- **Latitude:** 50.8495m
- **MaxPersonen:** 50
- **Leeftijdsgroep:** 12
- **Moeilijkheid:** Makkelijk

11. Fietstocht door Brussel

- **Beschrijving:** Een georganiseerde fietstocht door de stad.
- **DatumTijd:** 2024-08-11 09:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3525m
- **Latitude:** 50.8505m
- **MaxPersonen:** 60
- **Leeftijdsgroep:** 12
- **Moeilijkheid:** Gemiddeld

12. Basketbaltoernooi

- **Beschrijving:** Toernooi voor lokale basketbalteams.
- **DatumTijd:** 2024-08-12 14:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3535m
- **Latitude:** 50.8520m
- **MaxPersonen:** 100
- **Leeftijdsgroep:** 16
- **Moeilijkheid:** Gemiddeld

13. Voetbalwedstrijd

- **Beschrijving:** Een spannende voetbalwedstrijd tussen teams.
- **DatumTijd:** 2024-08-13 15:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3565m
- **Latitude:** 50.8515m
- **MaxPersonen:** 200
- **Leeftijdsgroep:** 12
- **Moeilijkheid:** Gemiddeld

14. Yoga in het Park

- **Beschrijving:** Yoga sessie in het lokale park.
- **DatumTijd:** 2024-08-14 08:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3555m
- **Latitude:** 50.8525m

- **MaxPersonen:** 30
- **Leeftijdsgroep:** 12
- **Moeilijkheid:** Makkelijk

15. Hardloopwedstrijd

- **Beschrijving:** Wedstrijd voor hardlopers van alle niveaus.
- **DatumTijd:** 2024-08-15 10:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3575m
- **Latitude:** 50.8530m
- **MaxPersonen:** 150
- **Leeftijdsgroep:** 18
- **Moeilijkheid:** Gemiddeld

16. Tafeltennistoernooi

- **Beschrijving:** Toernooi voor tafeltennis liefhebbers.
- **DatumTijd:** 2024-08-16 13:00
- **Icoon:** `Assets/Iconen/sport.png`
- **Longitude:** 4.3545m
- **Latitude:** 50.8520m
- **MaxPersonen:** 40
- **Leeftijdsgroep:** 12
- **Moeilijkheid:** Makkelijk

Hobby Activiteiten

17. Kookworkshop

- **Beschrijving:** Leer nieuwe kooktechnieken in deze workshop.
- **DatumTijd:** 2024-08-17 11:00
- **Icoon:** `Assets/Iconen/hobby.png`
- **Longitude:** 4.3555m
- **Latitude:** 50.8515m
- **



geeft ze in formaat dat ik je heb doorgestuurd. zodat ik het gelijk bij mijn lijst kan toevoegen na de ,



Hier is de lijst van 20 extra activiteiten in hetzelfde formaat als de voorbeelden die je eerder hebt gegeven. Je kunt deze activiteiten eenvoudig toevoegen aan je lijst.

csharp

```
new CultuurActiviteit
{
    Titel = "Kunsttentoonstelling",
    Beschrijving = "Expositie van moderne kunstwerken.",
    DatumTijd = new DateTime(2024, 8, 1, 11, 0, 0),
    Icoon = LoadImage("Assets/Iconen/cultuur.png"),
    Longitude = 4.3500m,
    Latitude = 50.8500m,
    MaxPersonen = 100,
```

```

        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 0,
        Sector = ActiviteitSector.Muziek
    },
    new CultuurActiviteit
    {
        Titel = "Straatmuziekfestival",
        Beschrijving = "Festival met live straatmuzikanten.",
        DatumTijd = new DateTime(2024, 8, 2, 14, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),
        Longitude = 4.3485m,
        Latitude = 50.8490m,
        MaxPersonen = 300,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 0,
        Sector = ActiviteitSector.Muziek
    },
    new CultuurActiviteit
    {
        Titel = "Fotografie Workshop",
        Beschrijving = "Leer de basis van fotografie in een workshop.",
        DatumTijd = new DateTime(2024, 8, 3, 09, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),
        Longitude = 4.3540m,
        Latitude = 50.8530m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 16,
        Sector = ActiviteitSector.Andere
    },
    new CultuurActiviteit
    {
        Titel = "Muzikale Avond",
        Beschrijving = "Geniet van live muziek van lokale bands.",
        DatumTijd = new DateTime(2024, 8, 4, 19, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),
        Longitude = 4.3520m,
        Latitude = 50.8540m,
        MaxPersonen = 150,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 0,
        Sector = ActiviteitSector.Muziek
    },
    new CultuurActiviteit
    {
        Titel = "Dansworkshop",
        Beschrijving = "Leer nieuwe dansmoves in een workshop.",
        DatumTijd = new DateTime(2024, 8, 5, 18, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),
        Longitude = 4.3570m,
        Latitude = 50.8520m,
        MaxPersonen = 30,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        Sector = ActiviteitSector.Dans
    },
    new CultuurActiviteit
    {
        Titel = "Poëzie Lezing",
        Beschrijving = "Beleef een avond van poëzie door lokale dichters.",
        DatumTijd = new DateTime(2024, 8, 6, 20, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),
        Longitude = 4.3550m,
        Latitude = 50.8510m,
        MaxPersonen = 80,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 0,
        Sector = ActiviteitSector.Andere
    },
    new CultuurActiviteit
    {
        Titel = "Theaterwandeling",
        Beschrijving = "Een interactieve theaterwandeling door de stad.",
        DatumTijd = new DateTime(2024, 8, 7, 16, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),

```

```

        Longitude = 4.3560m,
        Latitude = 50.8505m,
        MaxPersonen = 40,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 12,
        Sector = ActiviteitSector.Theater
    },
    new CultuurActiviteit
    {
        Titel = "Kunstworkshop voor Kinderen",
        Beschrijving = "Creatieve kunstactiviteiten voor kinderen.",
        DatumTijd = new DateTime(2024, 8, 8, 10, 0, 0),
        Icoon = LoadImage("Assets/Iconen/cultuur.png"),
        Longitude = 4.3580m,
        Latitude = 50.8525m,
        MaxPersonen = 25,
        Soort = ActiviteitSoort.Cultuur,
        Leeftijdsgroep = 6,
        Sector = ActiviteitSector.Andere
    },
    new SportActiviteit
    {
        Titel = "Marathon van Brussel",
        Beschrijving = "Een marathon door de stad Brussel.",
        DatumTijd = new DateTime(2024, 8, 9, 07, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3550m,
        Latitude = 50.8500m,
        MaxPersonen = 1000,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 18,
        Moeilijkheid = ActiviteitMoeilijkheid.Zwaar
    },
    new SportActiviteit
    {
        Titel = "Zumba les in het Park",
        Beschrijving = "Zumba les in een lokaal park.",
        DatumTijd = new DateTime(2024, 8, 10, 17, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3515m,
        Latitude = 50.8495m,
        MaxPersonen = 50,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 12,
        Moeilijkheid = ActiviteitMoeilijkheid.Makkelijk
    },
    new SportActiviteit
    {
        Titel = "Fietstocht door Brussel",
        Beschrijving = "Een georganiseerde fietstocht door de stad.",
        DatumTijd = new DateTime(2024, 8, 11, 09, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3525m,
        Latitude = 50.8505m,
        MaxPersonen = 60,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 12,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Titel = "Basketbaltoernooi",
        Beschrijving = "Toernooi voor lokale basketbalteams.",
        DatumTijd = new DateTime(2024, 8, 12, 14, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3535m,
        Latitude = 50.8520m,
        MaxPersonen = 100,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 16,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Titel = "Voetbalwedstrijd",

```



```

        Beschrijving = "Een spannende voetbalwedstrijd tussen teams.",
        DatumTijd = new DateTime(2024, 8, 13, 15, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3565m,
        Latitude = 50.8515m,
        MaxPersonen = 200,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 12,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Titel = "Yoga in het Park",
        Beschrijving = "Yoga sessie in het lokale park.",
        DatumTijd = new DateTime(2024, 8, 14, 08, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3555m,
        Latitude = 50.8525m,
        MaxPersonen = 30,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 12,
        Moeilijkheid = ActiviteitMoeilijkheid.Makkelijk
    },
    new SportActiviteit
    {
        Titel = "Hardloopwedstrijd",
        Beschrijving = "Wedstrijd voor hardlopers van alle niveaus.",
        DatumTijd = new DateTime(2024, 8, 15, 10, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3575m,
        Latitude = 50.8530m,
        MaxPersonen = 150,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 18,
        Moeilijkheid = ActiviteitMoeilijkheid.Gemiddeld
    },
    new SportActiviteit
    {
        Titel = "Tafeltennistoernooi",
        Beschrijving = "Toernooi voor tafeltennis liefhebbers.",
        DatumTijd = new DateTime(2024, 8, 16, 13, 0, 0),
        Icoon = LoadImage("Assets/Iconen/sport.png"),
        Longitude = 4.3545m,
        Latitude = 50.8520m,
        MaxPersonen = 40,
        Soort = ActiviteitSoort.Sport,
        Leeftijdsgroep = 12,
        Moeilijkheid = ActiviteitMoeilijkheid.Makkelijk
    },
    new HobbyActiviteit
    {
        Titel = "Kookworkshop",
        Beschrijving = "Leer nieuwe kooktechnieken in deze workshop.",
        DatumTijd = new DateTime(2024, 8, 17, 11, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3555m,
        Latitude = 50.8515m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 18,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Titel = "Sieraden Maken",
        Beschrijving = "Workshop over het maken van je eigen sieraden.",
        DatumTijd = new DateTime(2024, 8, 18, 14, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3560m,
        Latitude = 50.8500m,
        MaxPersonen = 15,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 16,
        Niveau = ActiviteitNiveau.Halfgevoorderd
    },
    },

```

```

new HobbyActiviteit
{
    Titel = "Schilderworkshop",
    Beschrijving = "Leer schildertechnieken in een creatieve workshop.",
    DatumTijd = new DateTime(2024, 8, 19, 10, 0, 0),
    Icoon = LoadImage("Assets/Iconen/hobby.png"),
    Longitude = 4.3575m,
    Latitude = 50.8525m,
    MaxPersonen = 25,
    Soort = ActiviteitSoort.Hobby,
    Leeftijdsgroep = 18,
    Niveau = ActiviteitNiveau.Halfgevorderd
},
new HobbyActiviteit
{
    Titel = "Modelbouwdag",
    Beschrijving = "Dag om je eigen modelbouwprojecten te maken.",
    DatumTijd = new DateTime(2024, 8, 20, 13, 0, 0),
    Icoon = LoadImage("Assets/Iconen/hobby.png"),
    Longitude = 4.3550m,
    Latitude = 50.8530m,
    MaxPersonen = 30,
    Soort = ActiviteitSoort.Hobby,
    Leeftijdsgroep = 12,
    Niveau = ActiviteitNiveau.Beginner
},
new HobbyActiviteit
{
    Titel = "Lezing over Fotografie",
    Beschrijving = "Leer meer over de kunst van fotografie.",
    DatumTijd = new DateTime(2024, 8, 21, 19, 0, 0),
    Icoon = LoadImage("Assets/Iconen/hobby.png"),
    Longitude = 4.3540m,
    Latitude = 50.8505m,
    MaxPersonen = 50,
    Soort = ActiviteitSoort.Hobby,
    Leeftijdsgroep = 0,
    Niveau = ActiviteitNiveau.Gevorderd
},
new HobbyActiviteit
{
    Titel = "Creatieve Schrijfworkshop",
    Beschrijving = "Workshop voor het ontwikkelen van je schrijfvaardigheden.",
    DatumTijd = new DateTime(2024, 8, 22, 16, 0, 0),
    Icoon = LoadImage("Assets/Iconen/hobby.png"),
    Longitude = 4.3555m,
    Latitude = 50.8510m,
    MaxPersonen = 20,
    Soort = ActiviteitSoort.Hobby,
    Leeftijdsgroep = 16,
    Niveau = ActiviteitNiveau.Beginner
},
new HobbyActiviteit
{
    Titel = "Kunst en Ambachten Fair",
    Beschrijving = "Bezoek een fair met kunst en ambachten.",
    DatumTijd = new DateTime(2024, 8, 23, 10, 0, 0),
    Icoon = LoadImage("Assets/Iconen/hobby.png"),
    Longitude = 4.3565m,
    Latitude = 50.8535m,
    MaxPersonen = 60,
    Soort = ActiviteitSoort.Hobby,
    Leeftijdsgroep = 0,
    Niveau = ActiviteitNiveau.Nvt
},
new HobbyActiviteit
{
    Titel = "Yoga en Meditatie",
    Beschrijving = "Workshop met yoga en meditatie sessies.",
    DatumTijd = new DateTime(2024, 8, 24, 09, 0, 0),
    Icoon = LoadImage("Assets/Iconen/hobby.png"),
    Longitude = 4.3555m,
    Latitude = 50.8520m,
    MaxPersonen = 25,
    Soort = ActiviteitSoort.Hobby,

```

```

        Leeftijdsgroep = 12,
        Niveau = ActiviteitNiveau.Halfgevorderd
    },
    new HobbyActiviteit
    {
        Titel = "Creatieve Fotografie",
        Beschrijving = "Workshop over creatieve technieken in fotografie.",
        DatumTijd = new DateTime(2024, 8, 25, 11, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3570m,
        Latitude = 50.8505m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 16,
        Niveau = ActiviteitNiveau.Halfgevorderd
    },
    new HobbyActiviteit
    {
        Titel = "Handwerken en Borduren",
        Beschrijving = "Leer handwerken en borduren in een gezellige sfeer.",
        DatumTijd = new DateTime(2024, 8, 26, 14, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3580m,
        Latitude = 50.8525m,
        MaxPersonen = 15,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 12,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Titel = "Bloemschikken Workshop",
        Beschrijving = "Maak je eigen bloemstukken in een workshop.",
        DatumTijd = new DateTime(2024, 8, 27, 10, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3550m,
        Latitude = 50.8510m,
        MaxPersonen = 20,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 18,
        Niveau = ActiviteitNiveau.Beginner
    },
    new HobbyActiviteit
    {
        Titel = "DIY Decoratie",
        Beschrijving = "Workshop voor het maken van je eigen decoraties.",
        DatumTijd = new DateTime(2024, 8, 28, 15, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3560m,
        Latitude = 50.8530m,
        MaxPersonen = 25,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 12,
        Niveau = ActiviteitNiveau.Halfgevorderd
    },
    new HobbyActiviteit
    {
        Titel = "Digitaal Tekenen",
        Beschrijving = "Leer digitaal tekenen met behulp van speciale software.",
        DatumTijd = new DateTime(2024, 8, 29, 11, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3575m,
        Latitude = 50.8520m,
        MaxPersonen = 15,
        Soort = ActiviteitSoort.Hobby,
        Leeftijdsgroep = 16,
        Niveau = ActiviteitNiveau.Gevorderd
    },
    new HobbyActiviteit
    {
        Titel = "Muziek en Geluid Effecten",
        Beschrijving = "Leer over muziek en geluidseffecten in een interactieve workshop.",
        DatumTijd = new DateTime(2024, 8, 30, 13, 0, 0),
        Icoon = LoadImage("Assets/Iconen/hobby.png"),
        Longitude = 4.3555m,

```

```
Latitude = 50.8505m,  
MaxPersonen = 20,  
Soort = ActiviteitSoort.Hobby,  
Leeftijdsgroep = 12,  
Niveau = ActiviteitNiveau.Halfgevoerd  
}
```

Kopieer en plak deze code in je lijst na de reeds bestaande activiteiten om ze toe te voegen. Let erop dat de `LoadImage` methode het pad naar het icoon goed verwerkt en dat je ervoor zorgt dat de gegevens zoals `Longitude` en `Latitude` overeenkomen met de werkelijke locaties van de activiteiten.

ChatGPT kan fouten maken. Controleer belangrijke informatie.