

Hand-out

Programmeren

Periode 2

05. Databases

1. Inleiding

Een database, gegevensbank of databank is een digitaal opgeslagen archief, ingericht met het oog op flexibele raadpleging en gebruik. Voor de programmeur is een database iets waarin gegevens kunnen worden opgeslagen met behulp van programma's.

Er zijn verschillende soorten databases, maar wij beperken ons in dit deel tot de grootste groep databases: de relationele databases. Deze databases maken allemaal gebruik van de taal SQL (Structured Query Language).

2. Relationele databases

Even ter opfrissing van de theorie, de volgende regels met betrekking tot relationele databases:

- Een relationele database bestaat uit tabellen.
- Een tabel bestaat uit rijen (regels) en kolommen.
- Iedere regel in een tabel heeft dezelfde opbouw.
- Ieder tabel heeft een sleutelveld.
- Tabellen worden gekoppeld op basis van sleutelvelden en kolommen.
- Voor een n-op-m relatie heb je altijd een extra tabel nodig.

Een voorbeeld :

...

In dit geval bestaat de database uit drie tabellen, Games, Characters en GamesCharacters.

De tabel Games kent drie kolommen, Id, Name en Creator. Daarbij is Id (ai-die) het unieke identificatienummer dat ook wel sleutel wordt genoemd. Om deze kolommen te onderscheiden van gelijknamige kolommen van andere tabellen schrijven we in dit deel de namen van de kolommen altijd met de tabelnaam ervoor, dus niet Id maar Games.Id en Characters.Id.

Games.Id is de sleutelkolom voor de tabel Games en Characters.Id is de sleutelkolom voor de tabel Characters. De kolommen GamesCharacters.GameId en GamesCharacters.CharacterId vormen samen de sleutel van de tabel GamesCharacters. Dit noemt men een koppeltabel.

Als een kolom sleutel is van de tabel betekent dit dat een waarde maar een keer mag voorkomen in deze kolom.

Waarom deze structuur ? Een game, bijvoorbeeld Super Mario Galaxy, kent meerdere personages, zoals Mario, Luigi en Peach. Maar mario komt in meerdere games voor, bijvoorbeeld ook in Super Mario Kart. Deze structuur zorgt ervoor dat je al deze informatie kunt opslaan in tabellen zonder een conflict te krijgen met de regels voor sleutels :

...

Door de sleutels te combineren kun je erachter komen welk personage voorkomt in welk spel. In het spel met Games.Id 1 (Super mario Galaxy) komen de personages voor met Characters.Id's 1, 2 en 3 (dus Mario, Luigi en Peach).

3. **Microsoft Access**

Dit is een zogenaamde desktop database die deel uitmaakt van de Microsoft Office suite. Access kan goed ingezet worden voor prototyping (het snel in elkaar zetten van een applicatie als voorbeeld om iets uit te proberen) en in kleine (desktop) omgevingen. Visual Studio werkt goed samen met Access.

4. **MySQL**

MySQL is een vreemde eend in de bijt: geen Microsoft product, niet standaard geïntegreerd in Visual Studio en een Open Source programma. Toch behandelen we ook deze database, en wel om de volgende redenen:

- Je leer binnen Visual Studio werken met niet-Microsoft-producten.
- Je leer werken met een van de grote (gratis) databases die veel gebruikt worden op het internet.

Om inn Visual Studio met MySQL te kunnen werken hebben we een zogenaamde connector nodig, die kun je vinden op het volgende adres : <https://www.mysql.com/products/connector/> en kies dan voor de ADO.NET connector.

Deze connector kun je downloaden en in je project implementeren. Vervolgens moet je in de code een connection string aanmaken en kun je aan de slag.

5. **Microsoft SQL-Server Express**

Microsoft SQL Server Express is het kleine broertje van SQL Server (de grote database van Microsoft). Ook deze producten zijn goed geïntegreerd in Visual Studio. Omdat je leert echte programma's te maken voor een bedrijfsmatige omgeving, is het belangrijk dat je ook weet hoe je met deze omgeving verbinding maakt.

6. **DataSets**

In C# werken we nooit rechtstreeks met een database. Een database binvindt zich net zoals andere bronnen van data (DataSources of gegevensbronnen), altijd buiten het programma. Binnen het programma werken we met een DataSet, en die ziet er net zo uit als de tabellen van een relationele database.

Een DataSet onderhoudt de verbindingen met de DataSources. Een DataSource kan een database zijn maar ook een XML bestand, een spreadsheet of een willekeurig ander bestand.

Hiermee is meteen duidelijk waarom men een DataSet bedacht heeft: je kunt op deze manier gegevens van allerlei DataSources eenvoudig in een programma combineren.

Een DataSet bestaat uit DataTables. Een DataTable werkt net zoals een tabel in een database. Alleen bevindt een tabel zich in een database en een DataTable staat in een DataSet.

Een DataSet bestaat alleen in het geheugen en dat betekent zodra je het programma beëindigt alle gegevens weg zijn. Je moet er zelf voor zorgen dat de gegevens vanuit een DataSet weer opgeslagen worden in de oorspronkelijke DataSources.

7. Demo

In dit stuk gaan we klassikaal een demo maken. Het is de bedoeling dat je mee kijkt en zo stapsgewijs houvast krijgt op het onderwerp, maak daar waar nodig is aantekeningen in je schrift of in je kladblok. Als je klaar bent ga je zelf aan de slag dus let goed op!

8. Opdracht

In deze opdracht ga je aan de slag met het maken van het eerste deel van Boter Kaas en Eieren. De bedoeling is dat je een Grid gebruikt om het speelveld in op te maken. Dit doe je door gebruik te maken van grid en ColumnDefinitions. Vervolgens kun je controls gebruiken op de plaats van de cirkels en kruisjes, bijvoorbeeld een Image Control maar dit kan ook een Button of een Label zijn.

Het is op dit moment nog niet de bedoeling dat je het spel helemaal werkend krijgt maar wel het speelveld opmaakt en er voor zorgt dat wanneer er ergens geklikt wordt er een kruisje op het scherm komt te staan. De volgende les gaan we verder aan deze opdracht en maken we de logica achter het spel op.

