



Access voor Beginners H4

Handleiding van Helpmij.nl

Auteur: Octafish

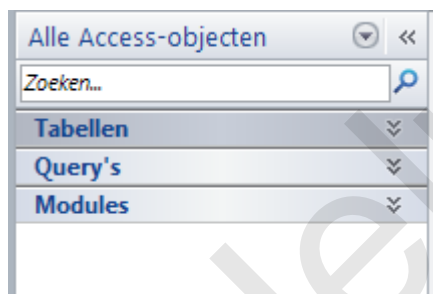
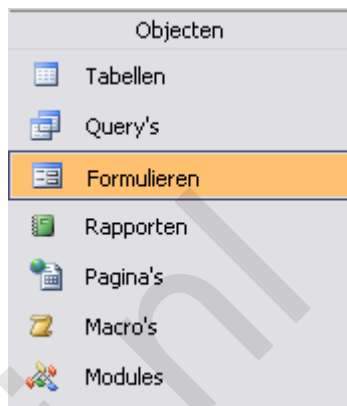
Mei 2011

“ Dé grootste en gratis computerhelpdesk van Nederland ”

In deze aflevering gaan we beginnen met het maken van de Frontend en Backend database. We beginnen daarbij eerst met het koppelen van de twee databases via de Import tool. Ook zul je lezen hoe je een database zelf kunt opsplitsen. Later in de cursus gaan we de frontend via VBA koppelen aan de backend.

De Backend-Frontend database

Je kunt er bij een database voor kiezen om alle objecten en elementen die je in de database nodig hebt in één bestand te houden. Access is hierin begijnsel ook op gebouwd, met een verdeling van alle componenten in verschillende objecten. Deze objecten vind je in Access 2003 terug in het navigatievenster. Kies je een categorie, dan verschijnen in het grote venster rechts alle objecten die binnen die categorie zijn aangemaakt. Zo kun je heel snel al je objecten terugvinden.



In Access 2007 en Access 2010 werk je met een Navigatiebalk. Deze bevat alle objecten in één smalle balk. De verschillende categorieën kun je naar wens in- of uitklappen door op de knopjes met de dubbele pijl te klikken, zodat je de verschillende objecten per categorie kunt bekijken. Ook kun je de objecten op verschillende manieren groeperen. Omdat de navigatiebalk veel minder ruimte heeft als de vorige Access-versies, kun je met behulp van de navigatiebalk <Zoeken> de lijst filteren, zodat je nog een enigszins overzichtelijke lijst overhoudt.

In de cursus maken we gebruik van de groepen voor Tabellen, Queries, Formulieren, Rapporten en Modules. We gaan het dus niet hebben over Pagina's en Macro's.

Voordelen en nadelen van volledige database

Wat zijn de voordelen en nadelen van een gesplitste database, en wanneer moet je het toepassen?

Eigenschap	Volledige database	Gesplitste database
Aantal bestanden	1	2 of meer
Geschikt voor	Enkele gebruiker	Meerdere gebruikers
Onderhouden van db	Relatief simpel; alles staat in één bestand.	Als je verschillende Frontends hebt gekoppeld aan een Backend, dan kan het zijn dat je een aanpassing een aantal keer moet herhalen in de Frontends. Dat kan meer werk opleveren.
Effect onderhoud op gebruiker en nieuwe objecten ontwikkelen.	Als de database in onderhoud is, kan er niemand anders in werken. Het werk ligt dan dus stil.	Voor nieuwe objecten of onderhoud kun je een kopie van een frontend maken, daar de (nieuwe) objecten in ontwikkelen, en kopieën van de nieuwe Frontend uitzetten bij de gebruikers.
Veiligheidskopieën maken	Een back-up hoeft maar van één bestand gemaakt te worden, wat vrij simpel is.	Afhankelijk van het aantal verschillende Frontends, moet je van meer bestanden een back-up maken.
Opstartopties	Voor verschillende gebruikers(groepen) moet er een uitgebreide rechtenstructuur worden ontworpen. Het maken en onderhouden hiervan kan lastig zijn.	Elke specifieke gebruiker(sgroep) kan een aparte Frontend krijgen. Hier zitten alleen de formulieren en rechten in die ze voor de specifieke eigen taken nodig hebben.

Het werken met een enkel databasebestand heeft dus voor- en nadelen:

Voordelen

- Het voordeel van een volledige database is, dat je alles in één bestand hebt staan. Je hoeft dus niet te zoeken naar deelbestanden, en als je besluit om de db te verplaatsen, kun je dat probleemloos doen. Je zult misschien een snelkoppeling moeten aanpassen, maar dat is het dan ook wel. De database zal doorgaans altijd werken;
- Het onderhoud van de database is relatief simpel, omdat je in één bestand alle noodzakelijke objecten zoals formulieren kunt aanpassen;
- Het maken van een kopie van de database is ook simpel, want je hoeft maar één bestand te kopiëren.

Nadelen

Nadelen zijn er uiteraard ook:

- Als je met meerdere personen tegelijk in de database werkt, kunnen er problemen ontstaan met de snelheid van de verwerking;

- Twee personen kunnen tegelijkertijd een formulier openen, waarbij er een Lock kan ontstaan op het formulier;
- Als de database crasht, moet de hele database gerepareerd worden, als dat al mogelijk is. Het bedrijfsrisico is dus vrij hoog;
- Omdat alle objecten in één bestand staan, wordt dit bestand snel groter. Dit kan de snelheid gaan beïnvloeden;
- Het instellen van rechten voor verschillende gebruikers is relatief lastig. Je zult met een tabel met gebruikersrechten moeten werken om de gebruikersrechten op de formulieren etc. goed te kunnen instellen;
- Voor onderhoudswerkzaamheden aan de database mogen er geen gebruikers in de database werken.

Voordelen en nadelen van een gesplitste database

Het werken met een gescheiden databasebestand heeft ook voor- en nadelen:

Voordelen

- Er kunnen veel meer gebruikers tegelijkertijd in de database werken. De belasting op de database is vrij laag, omdat iedere gebruiker een eigen applicatie gebruikt;
- Je hebt niet alles in één bestand staan. Elke gebruiker(sgroep) heeft zijn eigen Frontend, waarin de voor die persoon of groep noodzakelijke formulieren etc. staan;
- Het is relatief eenvoudig om voor elke gebruikersgroep rechten op formulieren en rapporten in te stellen; je neemt alleen die formulieren op die de gebruikersgroep nodig heeft;
- Voor onderhoudswerkzaamheden aan de database kunnen de gebruikers in de Frontend database doorwerken. Deze doe je ofwel in de Backend, of in een aparte kopie van de frontend.

Nadelen

Nadelen zijn er uiteraard ook:

- Er moeten van meer bestanden back-ups worden gemaakt;
- Als je de database wilt verplaatsen, moeten de tabellen opnieuw gekoppeld worden. Dit is ook nodig als je aan de tabellen nieuwe velden toevoegt, omdat die anders niet zichtbaar zijn in de Frontend.

Database splitsen met wizard

De makkelijkste manier om een Frontend en Backend te maken is met behulp van de Wizard. Om de volgende opdrachten uit te voeren, kun je een eigen database gebruiken, of de cursus database downloaden van het speciale Helpmij.nl-topic dat aan de cursus is gewijd. Het adres ervan vind je [hier](#).

In Access 2003:

- Kies <Extra>, <Database Hulpprogramma's>, <Database splitsen>
De wizard *Database splitsen* wordt gestart.

In Access 2007/2010:

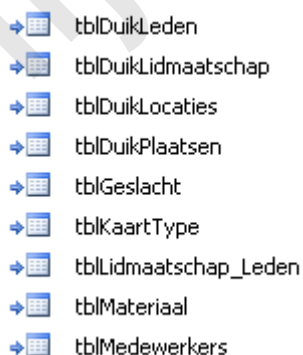
- Klik op de werkbalk <Databasehulpmiddelen>, in de groep <Gegevens verplaatsen>, op <Access-database>. De wizard *Database splitsen* wordt gestart.

Het is aan te raden om eerst een back-up te maken van de database, voor het geval er iets mis gaat. Dat kan ook via hetzelfde menu <Database Hulpprogramma's>.

- Kies <Extra>, <Database Hulpprogramma's>, <Database splitsen> en klik in het vervolgscherm op de knop <Database splitsen>. Access stelt een bestandsnaam voor, die je eventueel kunt veranderen. Het is aan te raden om het voorstel, de huidige bestandsnaam + de toevoeging *_BE*, te accepteren. Het is algemeen gebruik om een Backend database aan te duiden met deze toevoeging.
- Klik vervolgens op **OK**. De database wordt nu gesplitst.

Als Access klaar is met het proces, zit je in de Frontend van de database, de database waarin de formulieren komen dus.

Je ziet nog steeds dezelfde tabellen als er eerst in de database aanwezig waren. Alleen zijn alle tabellen nu voorzien van een pijl naast het tabelpictogram: de tabellen zijn nu *koppelingen* naar de tabellen in de Backend database. Het is nu ook niet meer mogelijk om de *Eigenschappen* van de tabellen te veranderen. Als je nu rechtsklikt op een tabel, en voor *Ontwerpweergave* kiest, word je gewaarschuwd dat je een gekoppelde tabel probeert te openen. Dit lukt op zich nog wel, maar je kunt wijzigingen niet meer op slaan.



Database handmatig splitsen

Het is uiteraard ook mogelijk om de database zelf te splitsen, zonder wizard. Hierbij maak je eerst een kopie van de database, en hernoemt die dan door er bijvoorbeeld *_BE* achter te zetten. De originele db krijgt dan de naam met *_FE* er achter. Zo weet je altijd wat de Backend en wat de Frontend is.

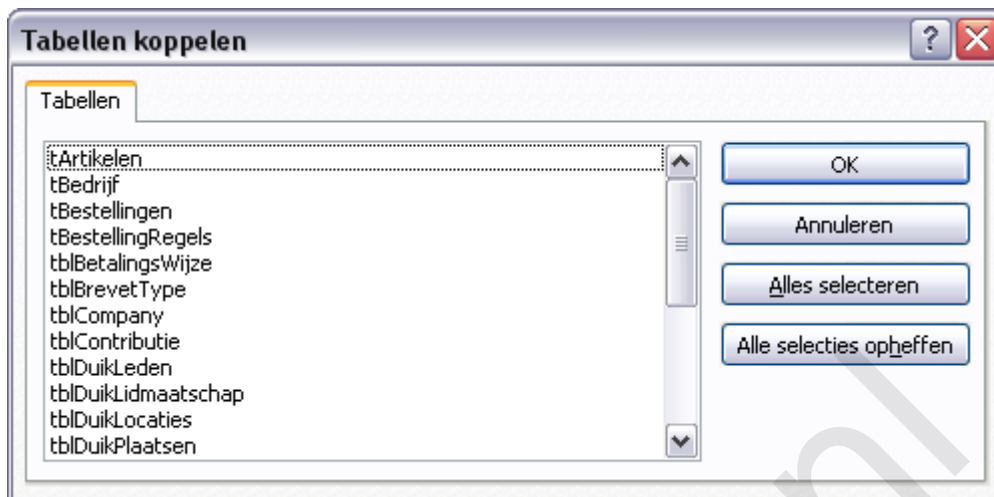
In de Frontend verwijder je vervolgens alle tabellen. Als je al koppelingen hebt gelegd tussen verschillende tabellen, vraagt Access of je die eerst wilt verwijderen. Klik op <Ja> om de koppelingen te verwijderen.

Als alle tabellen zijn verwijderd, kun je de tabellen uit de nieuwe backend database koppelen.

In Access 2003:

- Klik met de rechtermuis knop in het (nu lege) tabellen venster. Kies vervolgens <Tabellen koppelen> uit het snelmenu

- Selecteer de Backend database die je zo-even hebt gemaakt.
Het dialoogvenster *Tabellen koppelen* verschijnt.



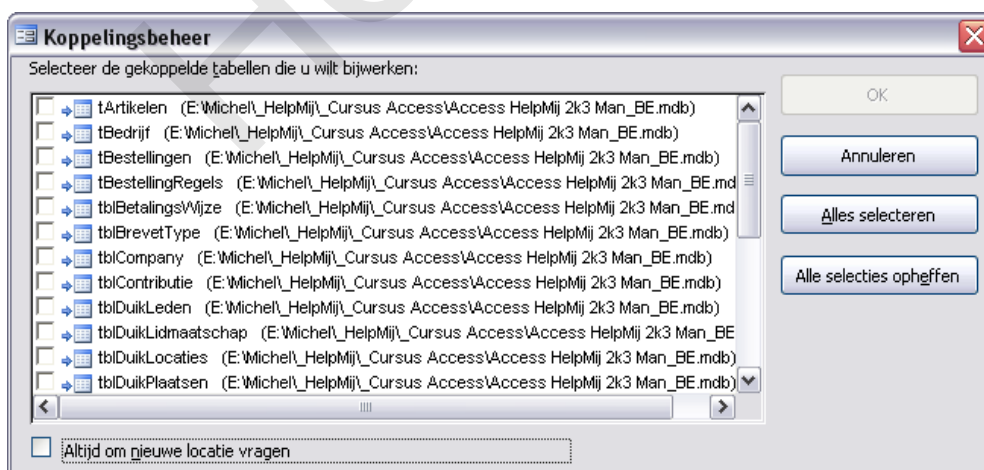
- Klik op de knop <Alles selecteren> en klik op <OK>
Alle tabellen worden nu gekoppeld; er verschijnen pijltjes naast de pictogrammen.

Tabellen opnieuw koppelen

Als je wijzigingen hebt aangebracht in de Gegevensdatabase (de Backend dus), dan moeten de koppelingen worden vernieuwd. Dat doe je met het menu *Koppelingen*.

In Access 2003:

- Kies <Extra>, <Database Hulpprogramma's>, <Koppelingsbeheer>
Het venster *Koppelingsbeheer* verschijnt



- Kies <Alles selecteren> en klik op <OK>
Access vernieuwt alle koppelingen; de voortgang van het vernieuwen kun je in de Statusbalk volgen. Als het koppelen is voltooid, geeft Access een melding dat alle tabellen zijn vernieuwd.

In *Koppelingsbeheer* kun je aanvinken of er altijd om een nieuwe locatie moet worden gevraagd. Deze optie is handig als je de Backend hebt verplaatst naar een map die verschilt van de map van de Frontend database. Access onthoudt de 'afstand' tussen de map van het gegevensbestand (de Backend) en de programmadatabase (de Frontend). Als beide databases in dezelfde map staan, dan zoekt Access bij het vernieuwen van de koppelingen de Backend in dezelfde map als de Frontend. Staat de Backend nu in een andere map, dan moet je die specifiek benoemen, voordat je de koppelingen kunt vernieuwen.

Koppelingen tussen de tabellen

Bij het splitsen van de database in een Frontend en een Backend, bevat de Backend, zoals eerder uitgelegd, de tabellen. Deze tabellen kunnen koppelingen bevatten, zoals in een vorig hoofdstuk is uitgelegd. Als je de tabellen vervolgens koppelt in een Frontend database, worden de koppelingen overgenomen uit de Backend database.

Heb je *Referentiële Integriteit* afgedwongen tussen de tabellen, wat zoals je nu inmiddels weet, eigenlijk verplicht is als je tabellen gaat koppelen, dan worden deze koppelingen ook meegenomen naar de Frontend database. Je hoeft de koppelingen dus niet opnieuw te maken. Sterker nog: je kunt in een Frontend database met gekoppelde tabellen geen koppelingen aanpassen, verwijderen of toevoegen. Dit dient allemaal te gebeuren in de Backend database.

Dit is in het begin misschien een beetje onhandig en zal enige gewenning nodig hebben, maar na verloop van tijd zul je deze 'eigenschap' van de Frontend-Backend constructie zeker kunnen waarderen, want het is een extra stukje beveiliging in je Frontend. Een gebruiker kan nu nooit, al dan niet per ongeluk, iets veranderen aan de structuur van je database! Dit recht behoort nu exclusief toe aan de beheerder van de Backend database.

Tabellen in de Frontend database

Je hebt gezien dat we in de Backend database alle gegevenstabellen hebben gezet. Toch zijn er bepaalde tabellen die soms zelfstandig in een Frontend database worden gezet. Dit zijn vaak tabellen die gebruikersinstellingen bevatten, die alleen voor die specifieke Frontend gelden. Dit zijn bijvoorbeeld tabellen waarin gebruikers hun eigen voorkeuren vast kunnen leggen. Je kunt bijvoorbeeld een formulier gebruiken waarin een gebruiker zijn eigen selecties kan bewaren. Deze records hebben geen betekenis voor andere gebruikers, dus die sla je dan op in een tabel in de Frontend database, zodat die gebruikers in die Frontend over hun eigen queries kunnen beschikken.

Koppelen met andere databases

Je kunt niet alleen koppelingen maken met Access-databases, maar ook met bijvoorbeeld Oracle of SQL-Server databases. Daarvoor is het vaak wel nodig om een ODBC-koppeling te maken, die je vervolgens als bron gebruikt voor de koppeling.

Een handige toepassing hiervoor is bijvoorbeeld een koppeling met Outlook, zodat je in Access kunt beschikken over het adresboek dat je in Outlook bijhoudt. Met zo'n koppeling met je contactpersonen in Outlook hoef je maar op één plek (Outlook) je gegevens bij te werken. In Access kun je deze gegevens vervolgens gebruiken voor mailings, rapporten et cetera.

Tevens kun je koppelingen maken met Excel bestanden, of tekstbestanden.

Samenvatting

Is het splitsen van een database in een Frontend en een Backend nu noodzakelijk?

Dat hangt helemaal af van het gebruiksdoel van je database. Als je de enige gebruiker bent van de database, bijvoorbeeld voor thuisgebruik, dan is het zeker niet nodig. Het kan dan zelfs een stuk lastiger worden om met de database te werken, omdat je steeds van database moet verwisselen als je iets wilt veranderen in de tabellen bijvoorbeeld.

Gebruik je de database op het werk, en moeten er meerdere gebruikers (tegelijk) in werken, dan is het zeker aan te raden. Je kunt dan overwegen om voor elke gebruiker(sgroep) een aparte Frontend te bouwen, waarin de eigen formulieren staan. Het resultaat is dan een veel stabielere database, die ook veel beter is te beveiligen.

Opdracht

In het Access-topic vind je twee voorbeelddatabases, die we voorlopig verder gaan gebruiken in de cursus: een Access 2003 variant, en een Access 2007/2010 database.

Gebruik de in dit hoofdstuk behandelde technieken om de database te splitsen in een Frontend- en een Backend database. Bekijk ook de tabellen, en de koppelingen die zijn gemaakt, en kijk ook of de structuur enigszins overeenkomt met wat je zelf hebt gemaakt.

Volgende Aflevering

In de volgende aflevering gaan we op basis van de voorbeelddatabase de eerste formulieren maken. Uiteindelijk is het de bedoeling om alle gebruikers via formulieren hun werk te laten doen; een gebruiker komt (en ziet) dus de tabellen en queries niet. Hierbij gaan we ook uitgebreid automatiseren m.b.v. VBA, want de standaardformulieren die je met de wizards maakt zijn eigenlijk alleen maar visuele lagen die bovenop de tabellen liggen. Een goed formulier is zodanig opgezet dat het gebruik ervan voor de gebruikers optimaal is geoptimaliseerd. En dat gebeurt dus meestal via VBA.

We gaan een huisstijl maken voor de formulieren, en we gaan de eerste taken automatiseren met VBA.