



Access cursus Hoofdstuk 3

Handleiding van Helpmij.nl

Auteur: OctaFish

April 2011

“ Dé grootste en gratis computerhelpdesk van Nederland ”

In deze aflevering gaan we beginnen met het opzetten van de database. We beginnen met het importeren van een tabel. Vervolgens gaan we de tabellen maken op basis van de geïmporteerde tabel, en gaan we tabellen aan elkaar koppelen, zodat er zinvolle relaties tussen de tabellen ontstaan. Eerst gaan we echter de leerstof uit het vorige hoofdstuk uitwerken in een praktijk voorbeeld.

Tabellen maken en normaliseren

Vaak wordt een database gemaakt op basis van een Excel-bestand dat om verschillende redenen onhandelbaar is geworden, bijvoorbeeld doordat er teveel kolommen zijn ingevoegd, of dat de limiet van het aantal beschikbare rijen in beeld komt.

Excel is prima geschikt om een enkele tabel in te maken, maar voor echte database oplossingen is het toch minder geschikt. Als voorbeeld het Excel-werkblad dat als basis dient voor onze duikclub. Dit bestand kun je downloaden, om zelf te bekijken en te importeren in de volgende oefeningen. Het plaatje hieronder laat maar een deel van de records zien.

	A	B	C	D	E	F	G	H	I
1	LidNr	Voornaam	Tussenvoegsel	Achternaam	Straatnaam	HuisNr	Huisnr_Toef	Postcode	Stad
2	71	Ed		Verstraete	W. Barendstraat	51		3572 PD	Utrecht
3	75	Ingrid		Willems	Gisbetlaan	32		3571 AK	Utrecht
4	178	Kris		Maes	Biltstraat	152		3572 BN	Utrecht
5	179	Nancy		Monfrans	Schermerlaan	99		3705 GM	Zeist
6	227	Annick		Landuyt	Stromboli	32		3524 WR	Utrecht
7	251	Luc	van	Heste	Waterstraat	25		3511 BW	Utrecht
8	256	Irene		Debondt	Rijnzichtlaan	30		3981 BV	Bunnik
9	257	Wauter	van	Liere	Adriaanstraat	59	bis	3581 SC	Utrecht
10	263	Ronny		Kerckaert	Bilderdijkstraat	3		3532 VA	Utrecht
11	264	Johan		Coppieters	Mallebaan	91		3581 CH	Utrecht
12	266	Tine		Dhoore	Brijantlaan	142		3523 CJ	Utrecht
13	71	Ed		Verstraete	W. Barendstraat	51		3572 PD	Utrecht
14	178	Kris		Maes	Biltstraat	152		3572 BN	Utrecht
15	179	Nancy		Monfrans	Schermerlaan	99		3705 GM	Zeist
16	227	Annick		Landuyt	Stromboli	32		3524 WR	Utrecht
17	257	Wauter	van	Liere	Adriaanstraat	59	bis	3581 SC	Utrecht
18	264	Johan		Coppieters	Mallebaan	91		3581 CH	Utrecht
19	266	Tine		Dhoore	Brijantlaan	142		3523 CJ	Utrecht
20	267	Kevin	van	Torre	Rotterdamsestraat	16		2586 GM	Den Haag
21	71	Ed		Verstraete	W. Barendstraat	51		3572 PD	Utrecht
22	75	Ingrid		Willems	Gisbetlaan	32		3571 AK	Utrecht
23	178	Kris		Maes	Biltstraat	152		3572 BN	Utrecht
24	179	Nancy		Monfrans	Schermerlaan	99		3705 GM	Zeist

In het voorbeeld zie je een lijst van alle mensen die een duiktraining hebben gehad. Wat gelijk op valt, is dat de naamgegevens van de verschillende personen in elke rij herhaald worden. Er zijn meer kolommen, zoals de kolom Duikplaats, waarin gegevens vaker voorkomen. In het hoofdstuk over Normaliseren hebben we gezien, dat we voor herhalende gegevens liever aparte tabellen maken. Dat werk kun je doen in Excel, of in Access. Omdat dit een cursus Access is, doen we het uiteraard allemaal in Access. Je kunt bijvoorbeeld de volgende acties uitvoeren:

- De volledige tabel importeren in Access.
- Bedenken welke tabellen je kunt maken van het werkblad. Deze tabellen maak je op basis van Tabelmaak queries.
- Aan de nieuwe tabellen sleutelvelden toevoegen, bijvoorbeeld een Autonummerveld.
- Bijwerkqueries maken, waarmee je de gegevens in de tabellen bijwerkt.

We gaan deze techniek uitvoeren op de kolom [SoortTraining] uit het voorbeeldbestand Duikleden Trainingsheet.

Importeren van een Excel bestand met Kopiëren/Plakken

Allereerst importeren we een Excel werkblad in Access. In het Access forum topic dat bij deze cursus hoort vind je het bestand **DuikLeden Trainingsheet.xls**. Dit bestand gaan we importeren in een nieuwe database, en het zal dienen als basis voor de verdere cursus.

De makkelijkste manier is om het hele werkblad te selecteren, op <Kopiëren> te klikken, en in Access in het Tabelvenster op <Plakken>. Access vraagt dan of de tabel kolomkoppen heeft.

Klik op <Ja> om de kolomkoppen over te nemen, of op <Nee> als er geen kolomkoppen zijn geselecteerd. De tabel wordt nu geïmporteerd.

Importeren van een Excel bestand met de wizard Importeren

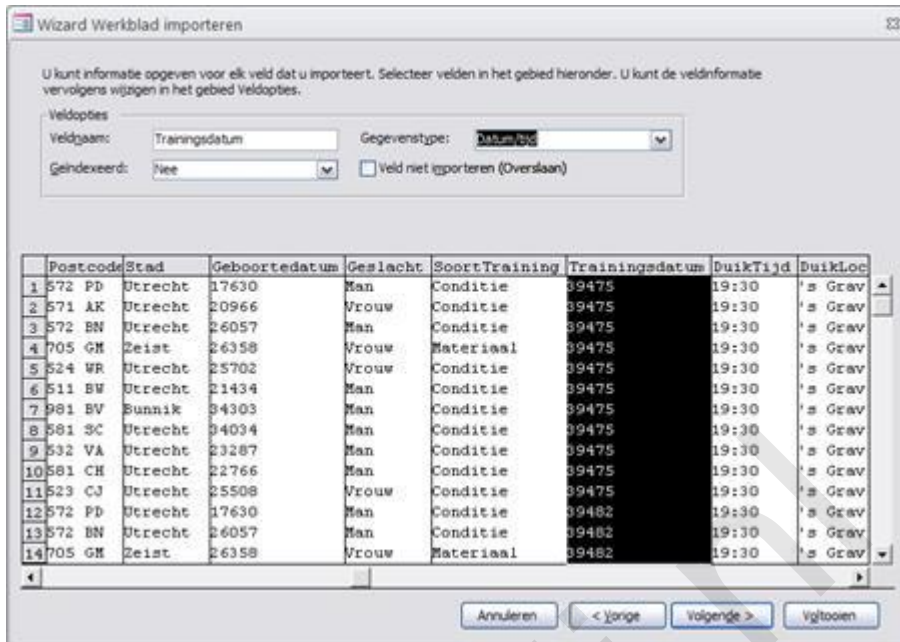
Een andere methode, die iets meer mogelijkheden biedt, is de *Wizard Importeren*. Hiermee doorloop je een aantal stappen, en kun je een aantal instellingen meegeven aan de nieuwe tabel.

1. In Office 2003:
Klik met de rechter muisknop in het tabellenvenster, en kies uit het snelmenu <Importeren...>
In Office 2007/2010:
Menu <Externe gegevens>, <Excel>, <Brongegevens importeren in een nieuwe tabel>.
2. Blader naar de map waar het bestand staat, en kies het bestandstype *Microsoft Excel*
3. Geef aan of er kolomkoppen zijn, of laat deze optie uit



4. Doorloop de wizard door een aantal keer op de knop <Volgende> te klikken, en eventueel aanpassingen te maken in de verschillende stappen. Doe dit naar eigen inzicht.

In Access 2010 kun je bijvoorbeeld de kolom *Trainingsdatum* omzetten van *Tekst* naar *Datum/Tijd*.



5. Bepaal of je een sleutelveld wilt aanmaken, of dat door Access wilt laten doen. In dit geval is nog geen sleutelveld nodig, dat maken we later
6. Noem de tabel **DuikLeden_Import**
7. Sluit de wizard af door in de laatste stap op **<Voltoeien>** te klikken

In het forum vind je onder deze  [link](#) het bestand; dit kun je downloaden om zelf om te zetten naar tabellen.

Een tabel maken vanuit de Import tabel

Zoals al eerder geconstateerd, zitten er in het importbestand kolommen met herhalende gegevens. Deze gaan we uit de tabel halen, en in een eigen tabel zetten. Als voorbeeld een simpele tabel (want er zit maar één kolom in), de kolom [*SoortTraining*].

We voeren de volgende stappen uit:

1. Maak een nieuwe query aan.
 In Acces 2003: groep Query's, knop <Nieuw>, <Ontwerpweergave>
 In Access 2010: menu <Maken>, knop <Query-ontwerp>
2. Selecteer de tabel <DuikLeden_Import> en klik op <Sluiten>

- Dubbelklik op het veld [SoortTraining], of sleep het veld naar het queryraster.



- Rechtsklik in het query venster naast de tabel en kies uit het snelmenu de optie <Eigenschappen>
- Kies bij de optie <Unieke waarden> de optie <Ja>
- In Access 2003:
Kies in het menu <Query> de optie <Tabelmaak query> (kan ook via het snelmenu)
Klik in Access 2007/2010 op de knop <Tabel maken>
- Geef de nieuwe tabel de naam <tblSoortTraining> en klik op de knop <Uitvoeren>
De tabel wordt aangemaakt, en kun je zien in de navigatiebalk in Access 2007/2010.
In Access 2003 moet je eerst naar het menu <Venster>, optie 1 om de tabellen te zien.
- Klik met de rechter muisknop op de tabel tblSoortTraining en kies <Ontwerpweergave>
- Maak een nieuw veld aan met de naam [TrainingSoortID], en kies het Gegevenstype <Autonummering>. Klik op de knop <Primaire sleutel> om het nieuwe veld als sleutel te definiëren.
- Sluit de tabel en sla de wijzigingen op.

Nu we een tabel hebben met unieke trainingstypen, kunnen we verder gaan met het vervangen van het tekstveld [SoortTraining] in de tabel DuikLeden_Import door een nieuw

veld, [*TrainingSoortID*]. Dit wordt logischerwijs een numeriek veld, want we hebben in de tabel *tblSoortTraining* hetzelfde veld aangemaakt als Autonummerveld.

1. Open de tabel *DuikLeden_Import* in Ontwerpweergave en voeg een numeriek veld toe met de naam [*TrainingSoortID*]. Let er op dat het veldtype *Lange Integer* is
2. Maak een nieuwe Query aan van het type *Bijwerken*
Access 2003: Menu <Query>, optie <Bijwerkquery>
Access 2007/2010: knop <Bijwerken>
3. Voeg de tabellen *Duikleden_Import* en *tblSoortTraining* toe aan de query
4. Sleep het veld **SoortTraining** uit *tblSoortTraining* naar het veld **SoortTraining** in de tabel *Duikleden_Import*.
5. Sleep het veld *TrainingSoortID* uit *Duikleden_Import* naar het queryraster. Typ in de regel <Wijzigen in> [**tblSoortTraining**].[**TrainingSoortID**]
6. Klik op de knop <Uitvoeren> om de records bij te werken
7. Hernoem de tabel **Duikleden_Import** naar **tblTraining**
8. De laatste stap is nu het verwijderen van het veld *SoortTraining* uit de tabel **tblTraining**. Dat is nu tenslotte niet meer nodig.

Opdracht

- Splits de tabel *Duikleden_Import* op in andere brontabellen. Bedenk hierbij dat je uiteraard ook groepen van gegevens naar een nieuwe tabel kunt verplaatsen. Zo kunnen alle herhalende persoonsgegevens in *Duikleden_Import* vervangen worden door één veld *LidNr*. De persoonsgegevens gaan dan bijvoorbeeld naar een tabel *tblDuikleden*.
- Importeer het Excel bestand **Duikleden_Betalingen** en splits die ook op in brontabellen. Hernoem de geïmporteerde tabel naar **tblBetalingen**

Tabellen koppelen

Een hulpmiddel om de gegevensintegriteit in de database goed te kunnen bewaken, is het leggen van Relaties tussen de tabellen. Het doel daarvan is, om ervoor te zorgen dat we in onze werktabellen geen records kunnen aanmaken met niet-bestaande gegevens. Het heeft bijvoorbeeld weinig zin om een artikel te verhuren aan een niet-bestaande klant; niet alleen kun je geen factuur sturen, omdat je geen adres hebt, maar als het artikel nooit wordt teruggebracht kun je de klant ook niet meer traceren.

Als je de vorige oefeningen hebt uitgevoerd, heb je een aantal brontabellen gemaakt waarvan het sleutelveld terugkomt in de tabellen *tblTraining* en *tblBetalingen*. Er is bijvoorbeeld nu een relatie tussen de tabel *tblSoortTraining* en *tblTraining*. Één van de basisprincipes van een goede database behelst dat je in gerelateerde tabellen geen gegevens opslaat die niet voorkomen in de brontabellen. In het bovengenoemde voorbeeld hebben we in de tabel *tblSoortTraining* 4 verschillende typen trainingssoorten. Deze zijn genummerd van 1-4. In de tabel *tblTraining* mogen we dus alleen maar de waarden 1, 2, 3 en 4 opnemen. Vul je in het veld *TrainingSoortID* de waarde 5 in, dan is dat een ongeldige waarde; we hebben namelijk geen trainingssoort met het ID-waarde 5.

Voorkomen dat ongeldige waarden worden ingevuld noemen we in de database <Referentiële Integriteit afdwingen>. We regelen dat in het venster *Relaties*.

Relaties maken tussen tabellen

Koppelingen leggen tussen tabellen doe je, net als met queries met meerdere tabellen, door velden naar elkaar te slepen. Deze velden moeten van hetzelfde type zijn, anders kun je ze niet koppelen.

In Access 2003 open je het venster *Relaties* met de knop **Relaties** in de knoppenbalk, of via het menu <Extra>, <Relaties>. In Access 2007/2010 klik je op de knop **Relaties** in de werkbalk <Hulpmiddelen voor databases>.

Je moet eerst tabellen toevoegen aan het venster. Als er nog geen tabellen staan, zal Access het venster <Tabellen weergeven> laten zien. Hier selecteer je de tabellen die je nodig hebt. Als je later extra tabellen wilt toevoegen klik je op de knop <Tabellen weergeven>.

De volgende stap is het slepen van het koppelveld uit de eerste tabel naar het overeenkomende veld uit de andere tabel. Het maakt niet uit vanaf welke tabel je begint met slepen; zolang je de juiste velden sleept, kan Access de goede koppeling maken.

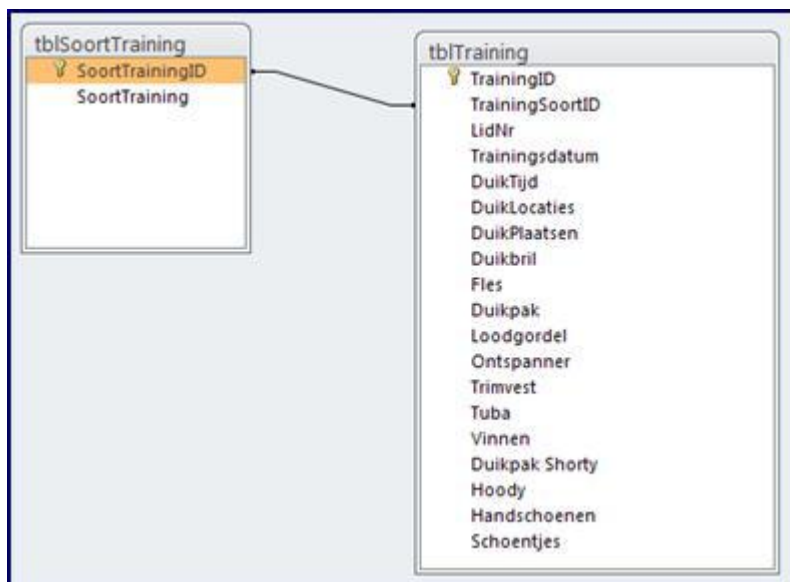
Relatie zonder <Referentiële integriteit afdwingen>

In het voorbeeld gaan we het veld *SoortTrainingID* koppelen aan *TrainingSoortID*.



Je ziet, om te beginnen, de twee velden die moeten worden gekoppeld. Erboven zie je de namen van de tabellen waar ze uit afkomstig zijn. Als je het (sleutel)veld per ongeluk op een verkeerd veld hebt losgelaten, kun je alsnog het juiste veld selecteren uit de keuzelijst(en).

Hoewel het niet mijn gewoonte is om mensen iets verkeerd aan te leren, maak ik voor deze keer toch een uitzondering. We klikken dus op de knop <OK>. In het venster *Relaties* ziet de koppeling er zo uit:

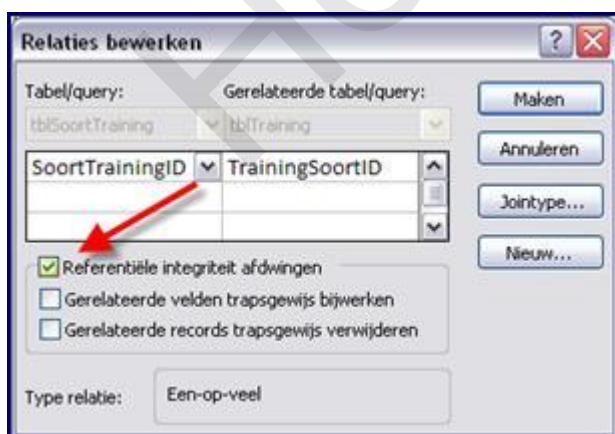


Tussen de velden loopt een dunne lijn. Deze koppeling is volkomen onbruikbaar. Doordat er geen *Referentiële integriteit* is tussen de tabellen, kun je net zo goed SoortTrainingID koppelen aan het veld Duikbril. Er wordt toch niet gecontroleerd of de gegevens juist zijn. En dat is nu net wèl de bedoeling van de koppeling...

Als je in een database dit soort koppelingen tegenkomt, weet je dus dat er iets verkeerd is. In dit geval kun je dubbelklikken op de verbindinglijn, en de koppeling alsnog correct instellen.

Relatie met <Referentiële integriteit afdwingen>

Het belangrijkste onderdeel in het venster *Relaties bewerken* is dan ook het selectievakje <Referentiële integriteit afdwingen>. Hiermee stellen we in dat er in de tabel **tblTraining** geen waarden mogen worden opgeslagen die niet voorkomen in de tabel **tblSoortTraining**.



Zoals hiervoor al is uitgelegd, heeft het ook niet zoveel zin om waarden op te slaan van niet-bestaande trainingssoorten. Als je behoefte hebt aan meer soorten, dan voeg je die eerst toe aan de tabel **tblSoortTraining**, waarna je ze kunt selecteren in de tabel **tblTraining**. We zullen later zien hoe we hier gebruik van gaan maken op onze formulieren.

In bijgaand voorbeeld is de optie <Referentiële integriteit afdwingen> wèl aangevinkt. Tegelijk met het aanvinken worden de twee overige opties in het menu actief. Deze zijn, afhankelijk van het doel van de tabellen, soms bruikbaar, en soms minder.

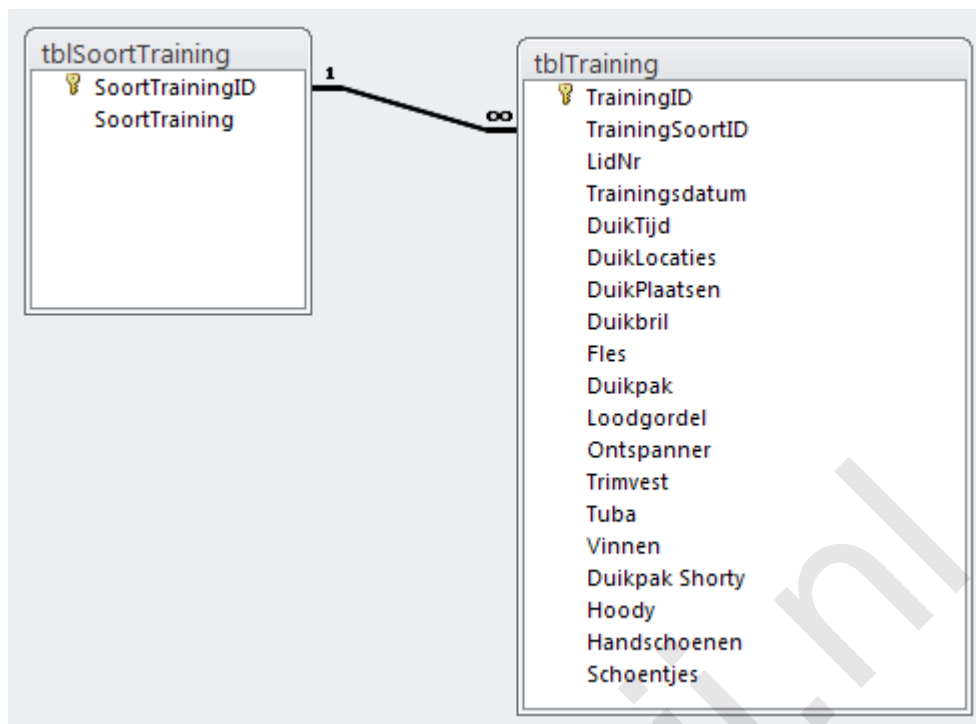
- De optie <Gerelateerde velden trapsgewijs bijwerken> is handig als je in je Brontabel (in het voorbeeld: de tabel tblSoortTraining) zelf de nummers bepaalt. Bijvoorbeeld door een ID te genereren op basis van een jaartal, maandwaarde en een volgnummer. Access kan dit soort sleutelvelden niet voor je maken, dus daar maak je meestal zelf een functie voor. Dit wordt later in de cursus dan ook behandeld.
Als je zelf dus een nummer aanmaakt, en later besluit dat het toch anders moet, omdat je bijvoorbeeld cijfers tekort komt, en het nummer wilt uitbreiden, zou je de situatie creëren dat in de gekoppelde tabel de verwijzingen naar de brontabel niet meer kloppen. In de gekoppelde tabel heb je tenslotte nog de oude nummers staan. Door dit vinkje aan te zetten zorg je ervoor dat in alle gekoppelde tabellen de nummers ook worden bijgewerkt, en dat kan uiteraard veel werk schelen.
Opmerking: als het sleutelveld van het type Autonummering is heeft het aanzetten van het vinkje geen zin, omdat Autonummering niet kan worden aangepast. Een recordID zal in dat geval dus nooit veranderen.
- De optie <Gerelateerde records trapsgewijs verwijderen> is in potentie een heel gevaarlijke instelling. Hiermee stel je namelijk in dat records uit gekoppelde tabellen worden verwijderd, als het bijbehorende bronrecord uit de brontabel wordt verwijderd. Neem bijvoorbeeld een tabel tArtikelen, met verschillende artikelen van verschillende leveranciers. Als een artikel niet meer leverbaar is, en je zou het record verwijderen met deze optie aangevinkt, dan zou je ook ineens in de tabel tBestellingen alle records van dit artikel verwijderen! En dan klopt er ineens niks meer van je administratie!
Bedenk dus, als je deze optie wilt gebruiken, of het je database niet vernielt. Er zijn overigens voldoende voorbeelden te bedenken, waar deze optie erg handig kan zijn, zeker in de testfase van je database.

Verschillen in soorten relaties

Onder in het venster zie je wel type koppeling er wordt gemaakt. Er zijn drie hoofdvarianten:

- Eén-op-veel relatie
- Eén-op-één relatie
- Veel-op-veel relatie

De meest gebruikte hiervan is de Eén-op-veel relatie. Deze relatie wordt gemaakt als je een sleutelveld van de brontabel sleept naar het overeenkomende veld in de gerelateerde tabel. In spreektaal zeg je eigenlijk: in mijn brontabel heb in allemaal unieke records, die in de gekoppelde tabel veel keer voor mogen komen. Een duiklid bijvoorbeeld, met een uniek lidnummer, maar vele keren meedoen aan trainingen. Derhalve bestaat er een één-op-veel relatie tussen tblDuiklid en de tabel tblTraining.

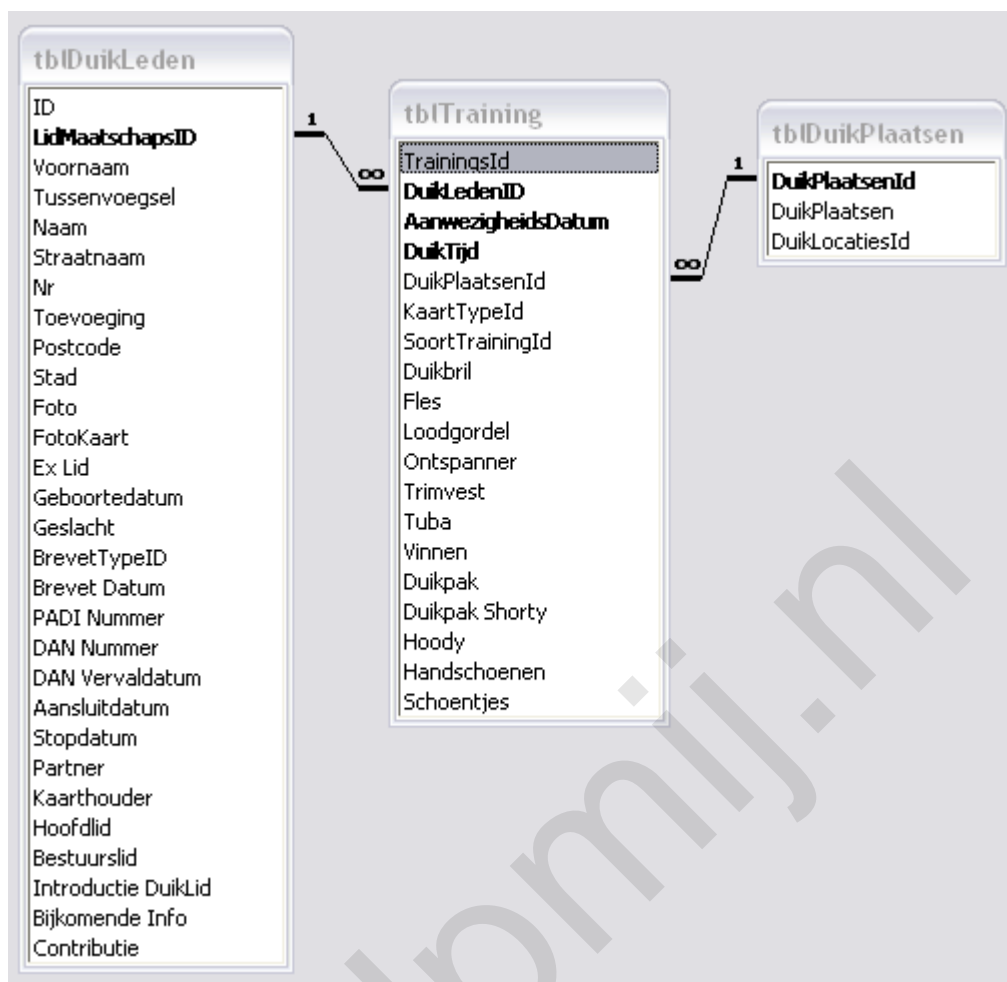


Je herkent een één-op-veel relatie aan het cijfer 1 aan de kant van de tabel met het sleutelveld, en het ∞ teken aan de veelkant.

Een één-op-één relatie maak je als je de twee sleutelvelden van de te koppelen tabellen aan elkaar koppelt. Normaal gesproken niet zo'n heel nuttige relatie, want je zegt er dus mee: er mag in beide tabellen maar één record zijn van de overeenkomende velden. Toch kan het een bruikbare relatie zijn. Denk bijvoorbeeld aan een tabel met gegevens van Bestuursleden, zoals beloning en rechten. Deze gegevens wil je niet opslaan in de Duiklid tabel, omdat iedereen daar bij zou moeten kunnen. Dus maak je een aparte tabel voor Bestuursleden. Uiteraard is elk bestuurslid één persoon, dus je neemt in deze tabel het Lidnummer op als sleutelveld. Ga je deze tabellen koppelen, dan wordt dit dus een één-op-één relatie, omdat in beide tabellen het veld Lidnummer voorkomt als sleutelveld.

De véél-op-véél relatie

De derde variant kan op twee manieren voorkomen: als je een koppeling maakt tussen twee niet-sleutelvelden, of als je een koppeling bekijkt tussen drie tabellen met tussen elk paar van twee tabellen een één-op-veel relatie. De laatste variant komt veelvuldig voor, en is daarom belangrijk voor de database. Bekijk bijvoorbeeld eens het volgende plaatje:



Tussen tblDuikLeden en tblTraining bestaat een één-op-veel relatie; elke persoon kan deelnemen aan meerdere trainingen. De andere koppeling ligt tussen tblDuikPlaatsen en tblTraining. Deze houdt in, dat elke plaats waar gedoken wordt, meerdere keren mag voorkomen in de tabel tblTraining. (verklaar dit zelf...). Tussen de tabellen tblDuikleden en tblDuikPlaatsen bestaat nu een veel-op-veel relatie: Veel duikleden kunnen duiken op veel plaatsen. Oftewel: elk duiklid kan meerdere keren meedoen aan een training, en die training kan op veel verschillende plaatsen gehouden worden.

Misschien was het nog niet opgevallen, maar de tabel tblTraining heeft een sleutel die is opgebouwd uit drie velden: DuikledenID, Aanwezigheidsdatum en Duiktijd. Oftewel: elk record is uniek op basis van Persoon, datum en trainingstijd. Je zou een sleutel kunnen maken van de velden DuikledenID en Aanwezigheidsdatum, maar daarmee beperk je het aantal trainingen dat iemand mag doen op één dag tot precies één. Door ook de trainingstijd erbij te nemen, kan een lid dus meerdere keren op een dag een training volgen, mocht dat gewenst zijn.

Eén knop heb ik met opzet nog niet genoemd: de knop <Jointype>. Met het venster achter deze knop kun je aangeven hoe de gegevens moeten worden gekoppeld. In Access termen heet dat: het Join type. In het scherm Relaties is het niet aan te bevelen om de standaard koppeling te veranderen; straks gaan we dat wèl doen bij het maken van queries. Op dit moment is het echter niet aan te bevelen om het Join type te veranderen.

Problemen bij het maken van relaties

Er zijn verschillende oorzaken aan te wijzen als je geen koppeling kunt maken met <Referentiële integriteit afdwingen> aangevinkt. Als je dat tegenkomt, moet je eerst het probleem verhelpen, want het heeft dus zoals eerder uitgelegd geen zin om een koppeling te maken zonder dat je Referentiële integriteit afdwingt. De twee belangrijkste redenen voor het mislukken van de koppeling zijn:

- Je probeert bijvoorbeeld een tekstveld uit de ene tabel te koppelen aan een numeriek veld uit de andere tabel. In essentie geldt: je moet alleen velden aan elkaar koppelen van hetzelfde gegevenstype. Probeer je het toch, dan zal Access waarschuwen dat de relatie niet gemaakt kan worden met Referentiële integriteit. Verander het veldtype van het niet-sleutelveld (dit is meestal de oorzaak) en kies voor dit veld hetzelfde type als het sleutelveld.
- In de te koppelen tabel staan gegevens die niet voorkomen in de brontabel. Een logisch voorbeeld is, dat je geen trainingen zou moeten vastleggen voor personen die geen lid zijn van de club. In de tabel Trainingen mag je dus alleen maar lidnummers invullen van bestaande ledennummers. De oplossing is tweeledig: vul de gegevens aan in de brontabel, of verwijder de gegevens uit de te koppelen tabel. Je kunt deze records opzoeken met de wizard *Niet-gerelateerde records*. Deze query laat bijvoorbeeld precies zien welke ledennummers geen referentie hebben naar een persoon in tblDuikLeden. Deze nummers moet je dus toevoegen, of verwijderen. Beide opties kunnen correct zijn, afhankelijk van je gegevens.

Als bovenstaande problemen op deze manier zijn verholpen, kan de relatie meestal wel worden gelegd, en is de database weer in orde.

Samenvatting

In dit hoofdstuk heb je geleerd hoe je een basistabel uit Excel kunt importeren in Access, en deze kan opsplitsen in verschillende brontabellen. Om de tabellen aan elkaar te kunnen koppelen, maak je in de nieuwe tabel sleutelvelden aan en voeg je dit veld vervolgens toe aan je gegevenstabellen. In de volgende stap ga je vervolgens de tabellen aan elkaar koppelen.

Een goede database heeft goede en correcte koppelingen tussen de diverse tabellen. Hiermee vergroot je de stabiliteit van de gegevens, en zal de kans op problemen aanzienlijk afnemen. Omgekeerd, als je verkeerde koppelingen maakt in je database, is er geen goede controle op de juistheid van je gegevens. Het resultaat is dan dat je gegevens opslaat die later niet terug te vinden zijn in je brontabellen, met (bijvoorbeeld) als gevolg dat je geen goede facturen kunt maken of op de juiste manier contributie kunt innen.

Opdracht

- Splits de tabel *Duikleden_Import* op in andere brontabellen. Bedenk hierbij dat je uiteraard ook groepen van gegevens naar een nieuwe tabel kunt verplaatsen. Zo kunnen alle herhalende persoonsgegevens in *Duikleden_Import* vervangen worden door één veld *LidNr*. De persoonsgegevens gaan dan bijvoorbeeld naar een tabel *tblDuikleden*.
- Maak een model voor de database van het Duikcentrum. Maak in ieder geval die tabellen waarvan jij vindt dat die nodig zijn in de database.
- Koppel alle brontabellen op de juiste manier aan de gegevenstabellen. Als een relatie niet te maken is, onderzoek dan wat daar de oorzaak van is, en verbeter die.
- Vul de tabellen met een aantal records, om te controleren of de gegevens op de juiste manier zijn in te vullen. Verbeter de tabellen als dat nodig is.

Volgende Aflevering

In de volgende aflevering gaan we op basis van een voorbeeld database de structuur van de database verder verfijnen, en beginnen we met het maken van een Frontend en een Backend database.

Helpmij.nl