Access 2010 – De Doe-Maar's en Niet-Doen's

Handleiding van Helpmij.nl

Auteur: OctaFish

Juni 2012

Access 2010 – De Doe-Maar's en Niet-Doen's

Of, zoals de Engelsen zeggen: the Do's And Don'ts

Mensen die mijn bijdragen aan het Access-forum met enige regelmaat lezen, zullen misschien weten dat ik een fervent gebruiker ben van Access 2003, en de nieuwe(re) versies met enige regelmaat afraad. In dit artikel zal ik de nieuwe features van Access 2007/2010 bespreken die ik een verbetering/aanvulling vind op de oudere versies, en ook wat in mijn ogen de nadelen van de nieuwe versies zijn.

Het Lint in Office 2007/2010

Met de introductie van Office 2007 is Microsoft afgestapt van de structuur van Menubalken in combinatie met Knoppenbalken. De bediening van het pakket verloopt nu via het Lint (Ribbon in het Engels). En die overgang heeft tot gevolg gehad dat de totale interface is aangepast. Gebruikers die ooit met Office 6 zijn begonnen, en tot en met Office 2003 redelijk probleemloos konden overstappen op de nieuwe versies, zagen zich ineens geconfronteerd met een pakket met een totaal ander uiterlijk (wat op zich niet slecht hoeft te zijn) waarin bijna geen enkele opdracht meer zonder zoeken is terug te vinden (wat ik wel een kwalijke zaak vind).

In de oude versies was een duidelijke tweedeling in functionaliteit in het programma terug te vinden: alle mogelijke opdrachten die een gebruiker kon uitvoeren, kon je terugvinden in de menustructuur. Voor handelingen/acties die je vaak gebruikte, kon je een knoppenbalk aanzetten. Op die manier had je snel toegang tot veel gebruikte handelingen (knoppen) en kon je de minder gebruikte opdrachten terugvinden in de menu's. Een extra voordeel van deze structuur was, dat je de interface geheel naar eigen inzichten kon aanpassen: je kon eenvoudig eigen menu's maken, eigen knoppenbalken etc. In de Lint-versies is dat een stuk lastiger.

Een stevig nadeel van het Lint vind ik de grootte ervan: doordat er zoveel knoppen op moeten staan, is het een behoorlijk grote sta-in-de-weg. De menustructuur in de vorige versies was veel kleiner, en je kon vrij makkelijk werkbalken uitzetten als je geen knoppenbalken wilde gebruiken. Die grootte wordt alleen nog maar vervelender als je bedenkt dat een computer monitor vroeger het formaat 4:3 had, en tegenwoordig meestal 16:9 is. Dus het scherm is breder geworden, waardoor de nuttige werkruimte in de hoogte is afgenomen! OK, je krijgt er in een Word document aan de linker en rechterkant grote lege vlakken bij, maar zit iemand daar op te wachten? Gelukkig kun je het Lint wel minimaliseren, zodat je weer voldoende ruimte op het scherm overhoudt.

In de vorige versies kon je de menubalken nog uitgebreid aanpassen naar je eigen wensen. In Office 2007 is dat nog een hele klus, die in Office 2010 gelukkig weer wat beter werkt. Het is nu ongeveer net zo simpel om de snelbalk en het lint aan te passen als in 2003. Met één kanttekening: in Office 2003 kon je ook de breedtes van keuzelijsten aanpassen, wat nog wel eens handig was, en dat kan in de huidige versies niet. Tenzij je in XML-code gaat rommelen, en dat zul je niet zo snel doen.

Ontwikkelingen in Access

Alle ontwikkelingen die Microsoft in de Office-versies heeft gestoken, vind je uiteraard ook terug in het database pakket Access. Laten we de specifieke nieuwe mogelijkheden er maar eens per 'productgroep' bij pakken! Sommige veranderingen zijn niet gelijk opvallend, maar hebben toch invloed op de manier van werken.

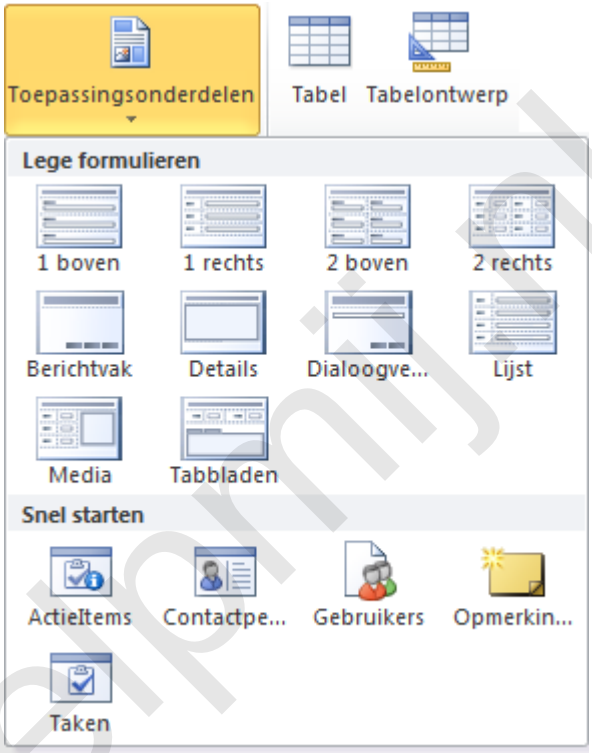
Eigenschappenvenster

Het Eigenschappenvenster bijvoorbeeld, lijkt nu een venster te zijn dat niet meer afhankelijk is van het object dat je op dat moment aan het bewerken bent. Als je in een formulierscherm op de knop <Eigenschappen> klikt, en je gaat vervolgens naar een query venster, dan staat het Eigenschappenvenster nog gewoon open. Dat kan behoorlijk irritant zijn; in het formulier wil ik de eigenschappen wèl zien, maar in een queryvenster meestal niet. Het venster wordt nu niet alleen

gebruikt voor Eigenschappen, maar ook voor de <Lijst met Velden>. In Access 2003 waren dat twee aparte vensters, die je aan of uit kunt zetten, en naast elkaar kon gebruiken. Nu is het één venster, waar dus òf de <Lijst met velden> zichtbaar is, of de <Eigenschappen>.

Sjablonen

Er is behoorlijk gesleuteld aan de ingebouwde sjablonen; zo kun je nu eenvoudig standaardonderdelen maken voor bepaalde toepassingen. In bijgaande afbeelding zie je daar een aantal voorbeelden van. Je kunt zelfs ook eigen Toepassingsonderdelen maken voor toekomstig gebruik.

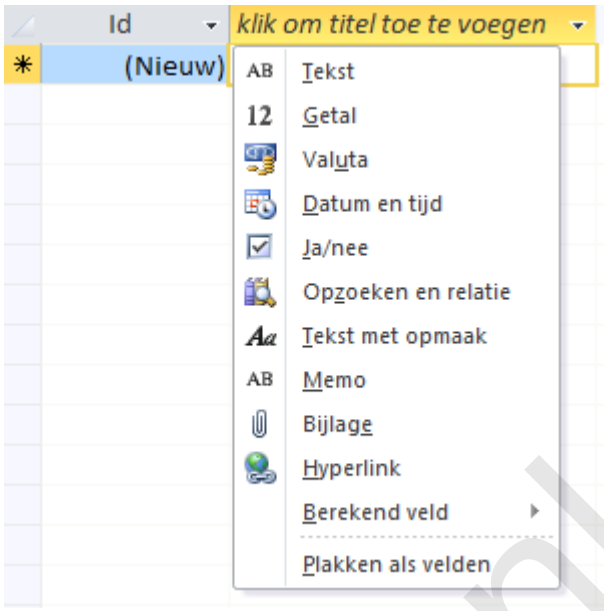


In het blok <Snel starten> zie je een aantal tabellen sjablonen staan die in één keer een aantal objecten aanmaken. Kies je Contactpersonen bijvoorbeeld, dan krijg je een tabel [Contactpersonen], een query, een aantal formulieren en een aantal rapporten die je dan weer kunt aanpassen. Deze werkwijze is vooral bedoeld (en geschikt) voor minder ervaren Access-gebruikers, die op deze manier snel aan de slag kunnen met een database. Ervaren gebruikers zullen toch wel enige tijd kwijt zijn aan het aanpassen van de door Access gemaakte objecten.

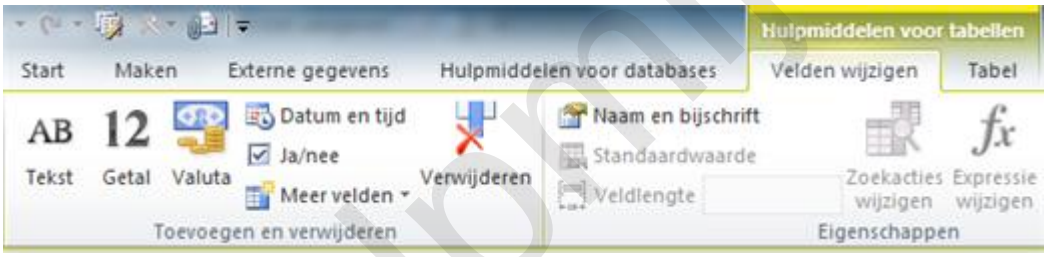
Tabellen

In de structuur van tabellen zou je zo op het eerste gezicht niet zo heel veel meer kunnen bedenken; een tabel is immers niet veel meer dan een vergaarbak van gegevens, die per record zijn gegroepeerd. Toch heeft Microsoft wat nieuwe objecten bedacht: de <keuzelijst met meerdere waarden>, en het <Berekend veld>. Daarvan is er één bruikbaar, en de ander niet. Ik zal dat later toelichten.

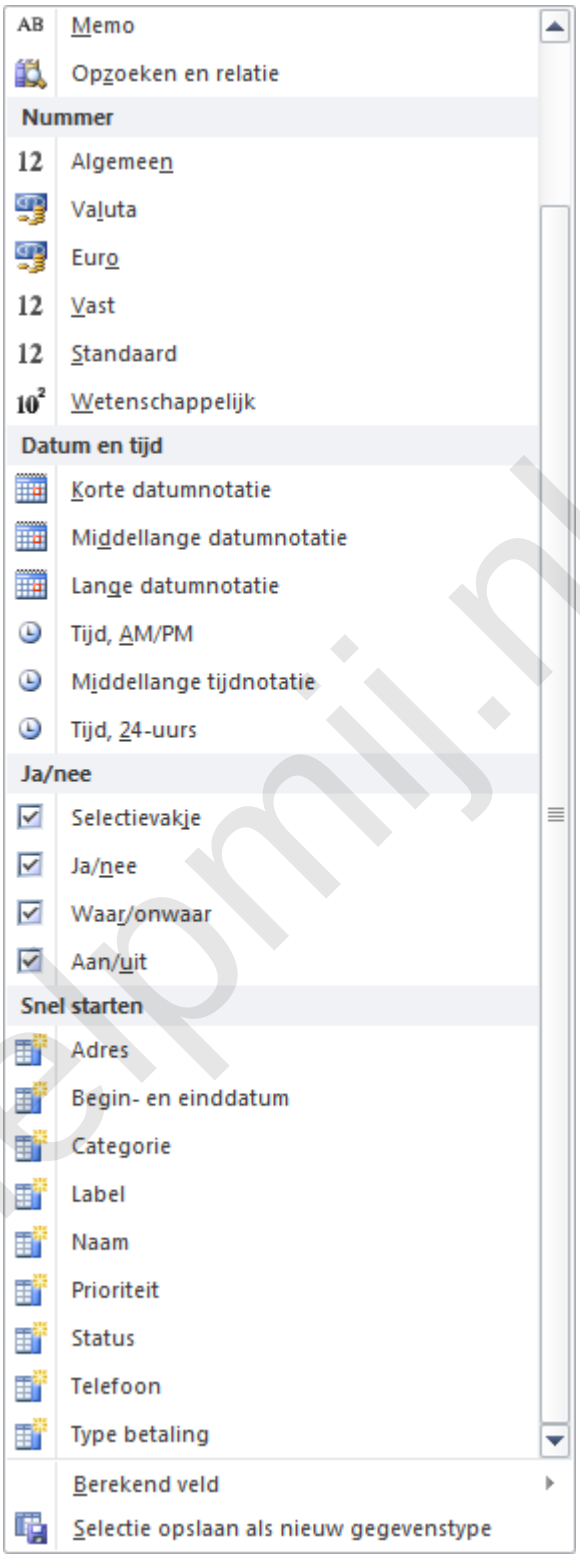
Een andere opvallende verandering is de manier waarop je nu velden kunt toevoegen aan een tabel. Wil je gebruik maken van de nieuwe werkwijze, dan maak je een tabel niet meer in de <Ontwerpweergave>, wat overigens nog steeds kan, maar klik je in het Lint <Maken> op de knop <Tabel>. Je kunt nu een tabel maken door velden te kiezen uit het blok met voorgedefinieerde velden, of je kunt in het tabelvenster een veldnaam intypen en een veldtype kiezen.



Daarnaast kun je via de groep <Toevoegen en verwijderen> in het lint <Hulpmiddelen voor Tabellen> ook een veldtype kiezen en toevoegen aan de tabel.



Als je op deze manier bijvoorbeeld een Valutaveld toevoegt, hoef je niet meer apart de veldingstellingen aan te passen.
Onder de knop <Ja/nee> zie je de knop met keuzelijst <Meer velden>. Deze bevat een uitgebreide keuzelijst met meer opties.



Met name de groep <Snel starten> is hier interessant. Zo voeg je met de optie <Adres> in één keer 5 velden toe: <Adres>, <Plaats>, <Provincie>, <Postcode> en <LandRegio>. Over de wijsheid van deze keuze van Microsoft kun je nog twisten: persoonlijk zou ik in Nederland niet zo snel een veld <Provincie> gebruiken, en het veld <Adres> opsplitsen in <Straatnaam> en <Huisnummer>, of misschien zelfs wel in <Straatnaam> en <Huisnummer> en <Huisnummer_Toev> om selecties te kunnen maken op basis van straatnamen bijvoorbeeld. Heb je in één veld zowel straatnaam als huisnummer, dan is je tabel eigenlijk al niet meer genormaliseerd, en dat is toch waar we naar streven!

Maar zoals al gezegd: met de opties uit <Snel starten> heb je wel op een vlotte manier een aantal velden in je tabel staan, die je dan vervolgens makkelijk kunt aanpassen.

Het Multi Value Field (Meervoudige Waarde Veld)

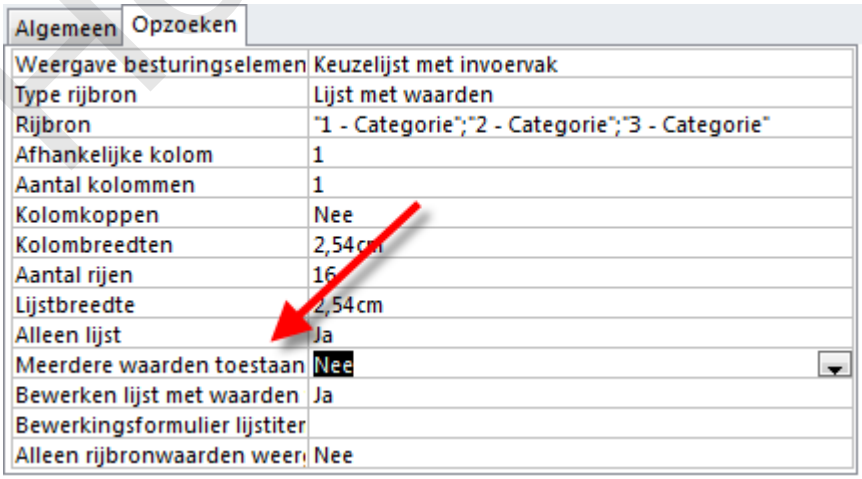
Een interessante nieuwe optie in Access is de *Meervoudige keuzelijst*. Toen ik er voor het eerst van hoorde was ik bepaald niet enthousiast, en een hoop andere Access-gebruikers ook niet. Wat is namelijk het geval? Als ontwerper probeer je een database te maken die aan bepaalde eisen voldoet. Eén van die eisen is, dat we in één veld van een record maar één waarde opslaan. Heb je voor een bepaalde toepassing meerdere keuzes op te slaan, dan maak je daarvoor een aparte tabel die je koppelt aan de hoofdtabel. Een voorbeeldje van zo'n constructie is een tabel <Bedrijven>, waaraan je <Contactpersonen> of <Medewerkers> hangt. Het is redelijk onzinnig om in je tabel <Bedrijven> één record voor het bedrijf te maken met daarin een tekst- of memoveld waarin je alle namen van contactpersonen typt!

En nu komt Microsoft dus met een veldtype waarin je wèl meerdere waarden kunt opslaan! Ik was dan ook bepaald niet blij met deze 'verbetering' en was er dan ook een felle tegenstander van. Nu ik mij iets meer in deze optie heb verdiept, heb ik mijn mening echter bijgesteld, en zie ik toch wel voordelen van een multi-keuzelijst. En dan met name voor keuzelijsten die een <Lijst met Waarden> bevatten die je intypt, al kun je ook een aparte tabel gebruiken die je koppelt aan de MVF.

De reden van mijn omslag is dat Access de opgeslagen waarden niet als één string opslaat, wat ik dus eerst dacht, maar in een interne onzichtbare tabel. Access maakt a.h.w. zelf een één-op-veel koppeling tussen de hoofdtabel en de tabel waarin de verschillende keuzes worden opgeslagen. Alleen zie je die tabel niet, en is het ook lastig om daar bij te komen. Omdat de geselecteerde waarden worden opgeslagen in een aparte tabel, is het ook mogelijk om in een query de inhoud ervan te splitsen. En als je dat kunt doen, dan heb je toch een vorm van een genormaliseerde tabel, die voldoet aan de 1e Normaalvorm.

Een voorbeeldje.

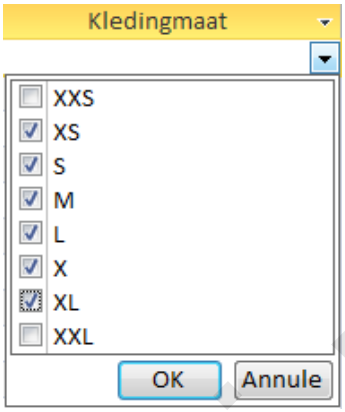
Als je een keuzelijst maakt (bijvoorbeeld via <Snel starten>, <Categorie> dan krijg je in eerste instantie een 'ouderwetse' keuzelijst, die één waarde kan bevatten. Je kunt in het tabelontwerp nu aangeven dat je van deze keuzelijst een <keuzelijst met meerdere waarden> wilt maken.



Dubbelklik je op <Nee> bij de regel *Meerdere waarden toestaan*, of kies je uit de keuzelijst <Ja>, dan vraagt Microsoft eerst of je dat zeker weet. Je kunt de keuzelijst daarna namelijk niet meer terugzetten naar een Enkelvoudige lijst; mocht je dat toch willen, dan zal je een nieuwe keuzelijst moeten maken.

Stel dat we in een tabel [Artikelen] kledingstukken opslaan met een artikelnummer, en je wilt in je tabel bijhouden in welke maten dat kledingstuk leverbaar is. Dan kunnen we een keuzelijst maken voor kledingmaten. Zo'n lijst loopt vaak van XXS via M naar XXL. Al die mogelijkheden kun je in een tabel

zetten, maar een keuzelijst is hiervoor ook prima geschikt. Als je één waarde kunt invullen in het veld [Kledingmaat], dan moeten we voor elke combinatie van een kledingstuk van een bepaald artikelnummer met een bepaalde maat een apart record maken. Doe je dat voor veel kledingstukken, dan loopt de tabel Artikelen snel vol. Bovendien is het dan lastig om overzichten te maken op basis van een artikelgroep. We kunnen daarvoor dus ook een keuzelijst met meervoudige keuze maken; dat is een stuk overzichtelijker omdat je nu voor elk kledingstuk maar één record hoeft te maken; in het veld [Kledingmaat] selecteer je nu alle mogelijke maten waarin het kledingstuk is te krijgen.



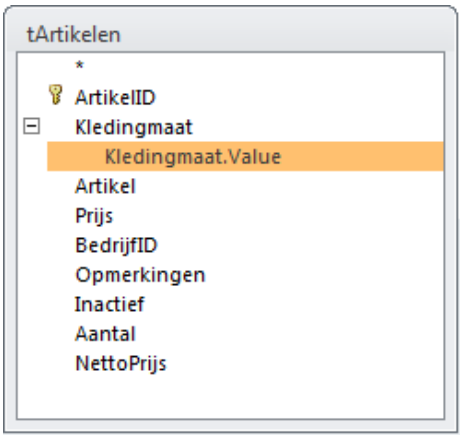
In de tabel wordt vervolgens deze tekst opgeslagen:

tArtikelen
Kledingmaat
L; M; S; X; XL; XS

Zoals ik eerder al zei: de opgeslagen string deed mij dus het ergste vrezen voor de integriteit van de database, omdat de tekst de indruk wekt dat de 1e normaalvorm met voeten wordt getreden. In een normale query ziet het resultaat er dan ook uit zoals ik dus vreesde!

qArtikelen - Niet gesplitst		
ArtikelID	**Kledingmaat**	**Artikel**
0601-1	L; M; S; X; XL; XS	Overall

Maar door een kleine aanpassing in het query ontwerp, is het wel degelijk mogelijk om de gesplitste waarden te zien. De truc daarvoor is gelukkig heel simpel. In het query-ontwerp, kun je kiezen welke velden je wilt zien. Dit is een relatief simpel proces van slepen dan wel selecteren van de juiste velden. Bij een meervoudige keuzelijst is de keuze iets uitgebreider.

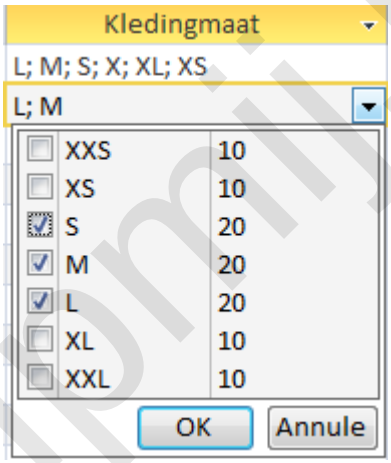


Je ziet dat de keuzelijst <Kledingmaat> een extra regel heeft. Onder het 'Hoofdveld' zie je de regel *Kledingmaat.Value* staan. Als je het veld gebruikt (dus de hoofdregel), dan zie je het resultaat uit de

query [qArtikelen - Niet gesplitst]. Gebruik je echter het veld *Kledingmaat.Value*, dan krijg je de query uit [qArtikelen - Gesplitst].

qArtikelen - Gesplitst		
ArtikelID	**Kledingmaat.Value**	**Artikel**
0601-1	XS	Overall
0601-1	S	Overall
0601-1	M	Overall
0601-1	L	Overall
0601-1	X	Overall
0601-1	XL	Overall

Of deze nieuwe optie handig is of niet, valt overigens te bezien, want hoe houd je nu de voorraden bij van de verschillende maten per artikel? En wat doe je met minimumvoorraden? De meervoudige keuzelijst kun je nog wel gebruiken om minimum voorraden te laten zien. Je kunt je voorstellen dat je van een kledingstukken voor de meest verkochte maten een hogere minimum voorraad wilt dan voor de extremere maten.



Je kunt de tweede kolom echter niet gebruiken in queries. Wil je naast de maten ook nog de minimumvoorraad terugzien in een query, dan is het beter om een aparte tabel te gebruiken. Hier kun je dan uiteraard voor elke kledingmaat de eigen gegevens vastleggen. Deze tabel koppel je dan in de query op basis van het sleutelveld uit de tabel (bijvoorbeeld het veld MaatID) en het veld [Kledingmaat.Value] uit de tabel [Artikelen].

Voor Voorraadbeheer heb je ook een aparte tabel voor nodig, waarin je voor elke maat apart bijhoudt hoeveel exemplaren er in de winkel liggen. Ook deze tabel kun je op basis van Kledingmaat.Value per artikel en kledingmaat verder verwerken.

Conclusie:

De nieuwe optie <Meervoudige Veld waarde (in het Engels: Multi Value Field, of afgekort MVF) biedt een zekere meerwaarde. Mits op de juiste manier gebruikt, kun je het opslaan van je gegevens vereenvoudigen zonder dat je afbreuk doet aan het Normaliseringsniveau van de database.

Het Calculated Field (Berekend veld)

Een stuk minder enthousiast ben ik over het nieuwe *Berekende Veld*. Ook dit veldtype druist in tegen de normaliserings regels die we toch proberen te hanteren in een database. Deze keer gaat het over de regel: "Gegevens die we kunnen berekenen slaan we niet op in een tabel". En dat is nu precies wat het berekende veld doet: het maakt een berekening op basis van andere velden in een nieuw record. En daarbij heb ik ook al een belangrijk woord laten vallen, namelijk het woord 'nieuw'. De berekening wordt namelijk alleen uitgevoerd als je een nieuw record aanmaakt, niet als je het record muteert, en

de velden waarop de berekening zijn gebaseerd veranderen. Je zult dan zien, dat de oorspronkelijke uitkomst gewoon blijft staan! En dat je die uitkomst dus zelf moet herberekenen. Het veld is qua functionaliteit dus te vergelijken met de Standaardwaarde die je aan een veld kunt meegeven; die wordt ook alleen ingevuld bij het aanmaken van een nieuw record.

En dat gegeven alleen al maakt het gebruik ervan totaal onbruikbaar; in een database worden nu eenmaal regelmatig gegevens gemuteerd. En zoals eerder al gezegd: voor berekeningen gebruiken we Queries. Die zijn daar ook voor bedoeld. Kortom: elk woord extra voor het berekende veld is er één teveel, wat mij betreft. In mijn databases zul je ze dan ook nooit tegen komen!

Gegevensmacro's

Een behoorlijke nuttige nieuwe vinding op tabelniveau is de introductie van *Gegevensmacro's*. Dit zijn macro's die worden getriggerd op basis van een verandering in een tabel en waarmee je gegevens in een tabel kunt bewerken/muteren. In vorige versies van Access kon je uiteraard al macro's maken, maar dat waren toch hoofdzakelijk macro's om handelingen mee te automatiseren zoals Bladeren in een formulier etc. Een gegevensmacro vervult in Access ongeveer de functie die de *Stored Procedure* heeft in SQL server. De macro wordt uitgevoerd op basis van een *gegevensverandering* op recordniveau. Om het anders te zeggen: als er een veldwaarde verandert in een record, kun je met een gegevensmacro een ander veld in hetzelfde record laten bijwerken.

De verschillende triggermomenten zie je in bijgaande afbeelding:

Je kunt bijvoorbeeld aan een veld denken waarin je een status bijhoudt, bijvoorbeeld storingsmeldingen. Een nieuwe storing krijgt dan bij het aanmaken de status <Geregistreerd>. Op het moment dat een behandelaar een melding opent, verandert de status van de melding: hij is nu in behandeling. De status van de melding zou nu dus moeten veranderen van <Geregistreerd> naar <In behandeling>. Deze actie kan een behandelaar uiteraard op het formulier doen door de status in de keuzelijst te veranderen, maar omdat dat een handmatige actie is, bestaat de kans dat hij dat niet doet. Met een Gegevensmacro kan je dit proces automatiseren, zodat behandelaars hier niet meer over hoeven na te denken.

Jammer genoeg geeft Microsoft op zijn eigen website een enigszins zinloos voorbeeld van een gegevensmacro. In hun voorbeeld wordt een veld [PercentageVoltooid] bijgewerkt op basis van een verandering in het veld [ProjectStatus]. Dat laatste veld kent de waarden <Niet gestart>, <Wordt uitgevoerd> en <Voltooid>. Bij de waarden <Niet gestart> en <Voltooid> horen vaste waarden (resp. 0% en 100%) en die worden ingevuld in het veld [PercentageVoltooid] als één van die twee waarden wordt gekozen uit de keuzelijst. En daar gaat het voorbeeld dan wat mij betreft stevig in de fout, want als je een tabel gebruikt voor de project statussen, en je zet daar de percentages bij de bijbehorende keuzes, dan kun je die percentages net zo makkelijk met een query opvragen; daar heb je helemaal geen berekend veld voor nodig! Sterker nog: door een macro aan de tabel toe te voegen, wordt de tabel er alleen maar trager van. Bovendien voeg je dus een overbodig veld toe aan je tabel, want het Percentageveld is gewoon op te vragen op basis van de status. Dus een slecht voorbeeld van Tabelontwerp! De macro is a.d.h.v. de video prima na te maken, dus om er eens mee te experimenteren, kan het geen kwaad.

Je kunt de gegevensmacro maken in de Ontwerpweergave (met de keuzelijst die hierboven is weergegeven) of vanuit de Gegevensweergave van de tabel, waar je in het lint <Tabel> aparte

knoppen hebt voor de alle gebeurtenissen, aangevuld met de bovenvermelde keuzelijst.

De macro-editor is behoorlijk veranderd sinds Access 2003, maar is nog steeds een tool die ik behoorlijk onbegrijpelijk vind. En nogal inconsequent in zijn gebruik. Zo kun je bij een ALS gebeurtenis bij het typen van een veldnaam de juiste veldnaam selecteren uit een keuzelijstje die Access op het scherm tovert. Prima, zo maak je geen typfouten! Kies je daarna vervolgens de Gegevensactie <Veld instellen>, dan moet je zelf de hele veldnaam intypen, en verschijnt de keuzelijst niet! Erg verwarrend, en helemaal niet consequent dus. Ook de layout van het scherm vind ik erg verwarrend. Maar dat kan dus ook aan mij liggen.

Je kunt de code weliswaar kopiëren naar het Kladblok, maar de XML-code die je dan krijgt helpt je vermoedelijk alleen maar van de wal midden in de sloot. Ik denk in ieder geval dat er maar weinig mensen zijn die zo'n macro liever in XML maken…

Hoewel ik nog niet met gegevensmacro's heb gewerkt, heeft het principe in mijn ogen absoluut een grote potentie, en ik ga er zeker in de cursus nog op terugkomen! Zoek, als je er mee gaat stoeien, dus vooral naar handelingen/gebeurtenissen die veldwaarden triggeren die moeten worden veranderd door een wijziging in de recordstatus; daar ligt denk ik de grote kracht van Gegevensmacro's. Zoals een recordstatus (<Geregistreerd>, <In behandeling> etc) of een behandelaar (als je <categorie A> selecteert, moet daar <Behandelaar A> aan werken, bij <categorie B> moet dat <Behandelaar B> zijn etc.)

De <F6> toets

Op zich geen onderwerp waar je een paragraaf aan zou wijden, zou je zeggen. Toch wil ik er wat woorden over kwijt. Wat is het geval? In de vorige versies kun je met één druk op de <F6> toets in de Ontwerpweergave van een tabel schakelen tussen de Velden in de tabel met hun gegevenstypen, en de *Veldeigenschappen* onderin het scherm. Inderdaad kom je, als je een veld hebt aangemaakt, met <F6> keurig in de Veldeigenschappen terecht. Maar als je daarna nog een keer op <F6> drukt, zit je ineens in het Eigenschappenvenster! Of, met een volgende klik, in de Navigatiebalk! Klik je door, dan zit je voor je het weet in het Lint, en Joost mag weten waar je nog meer terecht komt…. Het kost dus nogal wat klikken, voordat je weer bij je Velden lijst bent aangekomen. En dat vind ik dus bepaald een achteruitgang; het is eigenlijk niet meer te doen om met de <F6> toets te werken, en je wordt dus min of meer gedwongen om met de muis te werken. Een gemiste kans, wat mij betreft. Hadden ze niet een andere toets kunnen gebruiken om door de vensters te bladeren?

Ingebouwde macro's

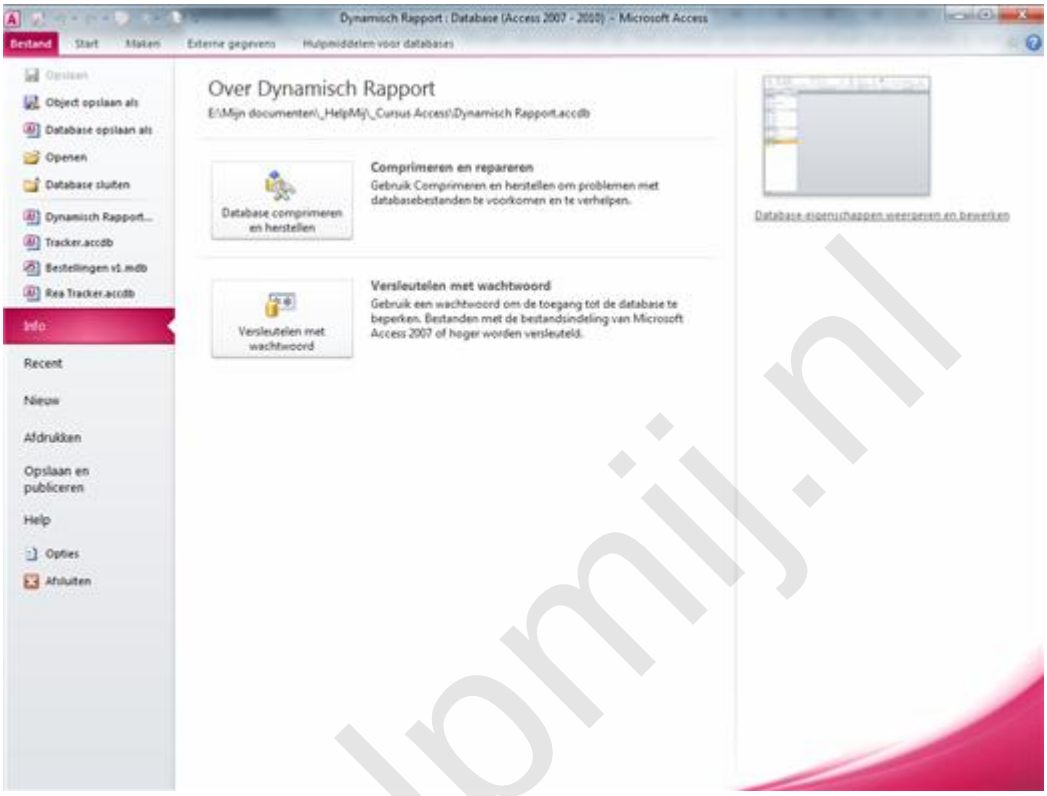
Als je een 2007/2010-database maakt (met de extensie accdb dus) krijg je niet zomaar VBA procedures als je knoppen maakt met de wizard. Meestal gebruikt Access een ingebouwde macro, die je dan nog wel kunt aanpassen, maar niet meer kunt exporteren naar een procedure, wat in de vorige versies nog wel kon. En dat is jammer, want ik vond het wel handig om met de wizard een basisknop te maken, en vervolgens in het VBA scherm de knop te modificeren. Je had dan een redelijk startpunt voor de knop voor de overige code die je wilde toevoegen. Omdat je die functies nu niet meer hebt, zit er niks anders op dan een knop gelijk in VBA te maken. En dan mis je dus die soms wel handige basiscode.

Een workaround is, dat je de database eerst maakt in 2003-format; de knoppen macro's worden dan namelijk in VBA gemaakt zoals we gewend waren. Je mist dan een paar nieuwe trucjes (zie daarvoor de betreffende hoofdstukken) maar dat hoeft op zich geen nadeel te zijn; je bent dan verzekerd van compatibiliteit met oudere Access-versies, en de database blijft gegarandeerd voldoen aan de officiële Normalisatie normen. En je kunt de database dus altijd upgraden naar SQL-server bijvoorbeeld. Ondertussen heb je wèl het voordeel van de interface van 2007/2010, dus je kunt wel wennen aan het werken in die omgeving. Een beetje 'best of both worlds' dus.

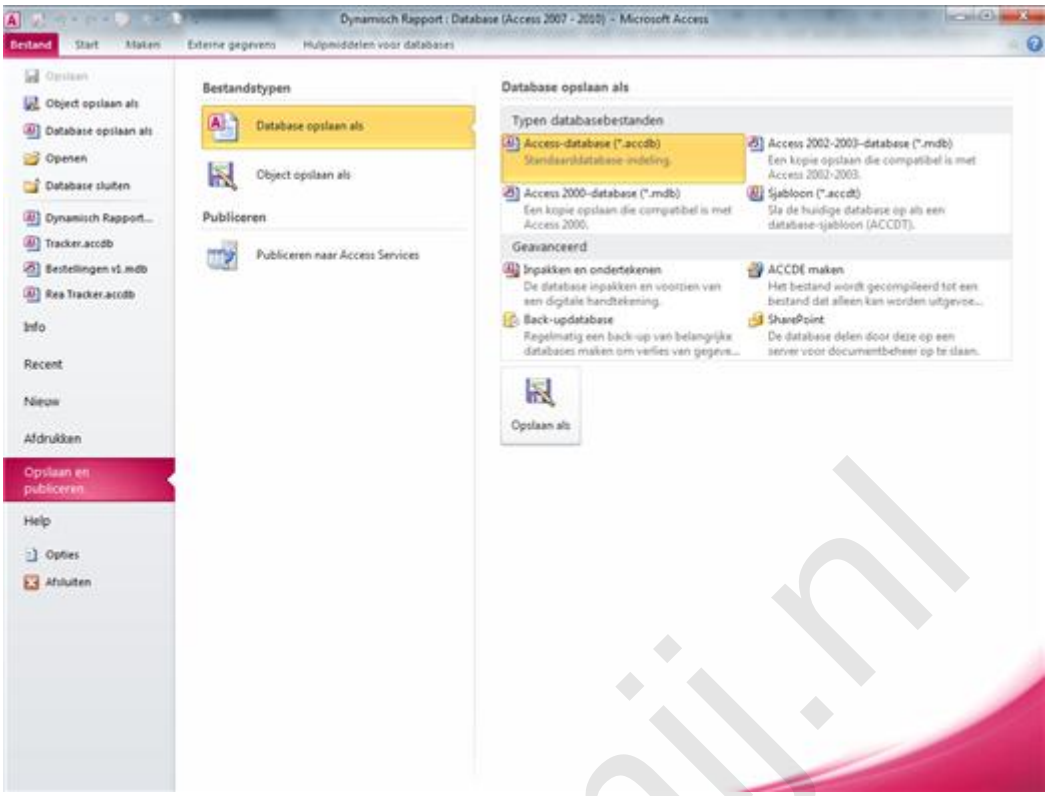
De Backstage View

Nee, niet die van Paradiso. Maar zo noemt Microsoft nu het hoofdscherm als je op de knop <Bestand>

klikt. Bij Microsoft hebben ze een bizar gevoel voor humor; om een computer uit te schakelen moet je op de <Start> knop klikken, om op de voorpagina van de database te komen, moet je op de belangrijkste knop klikken (de knop is als enige gekleurd, dus oogt nèt wat belangrijker) en dan blijk je niet op het podium te staan, maar Backstage… Maakt allemaal niet uit, het is maar een naampje! Waar het om gaat, is dat je op die pagina een veel beter overzicht hebt op allerlei instellingen m.b.t. de database.



Het scherm is doorgaans in drie secties verdeeld. In de linker kolom de opties die je kunt kiezen, in de middelste sectie knoppen om acties uit te voeren, zoals <Database comprimeren> in het voorbeeld, of <Opslaan Als> bij <Opslaan en Publiceren. En het rechtervenster bevat dan de verschillende mogelijkheden die je hebt, zoals <Database eigenschappen>.



Kortom: de Office knop uit Access 2007 is (gelukkig) verdwenen, en alles vind je nu terug op de Voorpagina (herstel: backstage) van het programma.

Formulieren

De formulieren bevatten nieuwe opties, die het ontwerpen van formulieren behoorlijk veranderen. Ik pak er een paar uit.

Indelings opties

Formulieren kun je nu ontwerpen in de <Indelingsweergave>. Deze weergave vind je ook bij Rapporten, en is een eenvoudigere manier om formulieren te ontwikkelen. Standaard worden formulieren die je baseert op een query of een tabel gemaakt met een bepaalde *Indeling*. Je kunt velden rangschikken in de indeling <Gestapeld>, waarbij labels links van de velden staan, en <Tabelvorm> waarbij de labels naar de koptekst van het formulier worden verplaatst. Met simpel klikken op deze twee knoppen kun je velden een andere indeling geven. Zijn velden eenmaal in een indeling opgemaakt, dan reageren ze als een groep. Ze worden bijvoorbeeld allemaal breder of smaller gemaakt als je de breedte van één veld verandert. Dat heeft uiteraard voor- en nadelen; soms is het helemaal niet handig als alle velden dezelfde breedte krijgen. Maar het geeft wel een bepaalde uniformiteit in het formulier. Een indeling kun je ook weer verwijderen, met de knop <Indeling verwijderen>. Als je dat doet, reageren de objecten weer zoals in Access 2003. Deze optie is niet beschikbaar voor databases in het mdb format, dus als je een oudere database opent zul je deze knoppen niet aantreffen. Om ze in oudere databases te gebruiken, moet je de db eerst publiceren als 2007/2010-database.

Stijlen

Een optie waar veel mensen (ik ook) op hebben zitten wachten is de mogelijkheid om het uiterlijk en de kleur van knoppen te veranderen. Daar heeft Microsoft eindelijk wat op verzonnen: *Snelle Stijlen*. Hiermee kun je simpel objecten zoals knoppen een nieuw uiterlijk geven. Een voorbeeldje:

Behalve de stijl, kun je per object dan ook nog drie extra instellingen aanpassen: de *Opvulling*, de *Omtrek* en het *Effect*. Bij <Opvulling> kun je een eigen kleur kiezen, en een Kleurverloop instellen. Bij <Omtrek> praat je over (inmiddels toch wel bekende) opties als <Lijndikte> en <Lijntype>, en bij <Effecten> kun je kiezen uit:

De opties zijn veel te uitgebreid om hier allemaal te laten zien, maar een aantal zijn in het voorbeeld hierboven toegepast. Behalve dat je nu dus bijna ongelimiteerd kunt knutselen aan je knoppen, vergroot dat wel het risico dat je formulier op een kermisattractie gaat lijken; het lijkt zinvol om enige terughoudendheid te bewaren, en niet overdadig met effecten te gaan strooien! Ander puntje dat je in het achterhoofd moet houden: de knoppen zijn niet uitwisselbaar met oudere versies. Daar zie je dus nog steeds de oude grijze knoppen terug als je de db terug converteert.

Samenvatting

Microsoft doet zijn uiterste best om in nieuwe versies het gebruikersgemak zoveel mogelijk te verbeteren. Dat doen ze door bestaande functionaliteiten te verbeteren, en nieuwe elementen toe te voegen. Zoals elke software ontwikkelaar dus. Zolang de ontwikkelaar daarbij de *gebruikersgroep* in het achterhoofd houdt, is dat prima. Iedereen die professioneel met een pakket werkt, is gebaat bij verbeteringen waardoor de uitoefening van de werkzaamheden wordt verbeterd. Anders wordt het als de ontwikkelaar alle gebruikers van zijn software over één kam scheert, en blijkbaar van mening is dat de vernieuwingen die worden uitgedacht dezelfde impact hebben voor alle gebruikers van de verschillende pakketten. En op dat punt verschil ik dus toch wel van mening met Microsoft!

Als we voor het gemak twee van de uitersten van het spectrum bekijken, dan wordt het probleem hopelijk snel duidelijk: Word en Access. Daarbij is Word voor mij een heel laagdrempelig programma; je start het, en je begint te typen, en je hebt het resultaat. Daarbij kun je de tekst nog opmaken met tekstopmaak etc zodat het er fraaier uit ziet. Het verschil tussen een professionele tekstverwerker en een beginner hoef je niet noodzakelijkerwijs aan het eindresultaat te kunnen zien: de beginner kan exact hetzelfde resultaat verkrijgen als de doorgewinterde professional. Hooguit dat die laatste het werk een stuk eerder al zal hebben, bijvoorbeeld door een hogere typsnelheid.
Aanpassen van werkbalken en menustructuren zullen de aan een bepaalde versie gewende professional in eerste instantie afremmen in het werk, omdat de functies die men vaak gebruikt zijn verplaatst of zelfs verdwenen, maar het uiteindelijke doel (teksten produceren) zal er weinig last van ondervinden.

Kijken we naar Access, dan zien we een heel ander programma. Wil je een goede database kunnen maken, dan dien je over bepaalde kennis en inzichten te beschikken. Een beginner met Access zal

doorgaans niet in staat zijn een volwaardige professionele database te maken. Dat is uiteraard geen schande, want een database ontwikkelen vereist nu eenmaal die kennis. En die moet je leren. En dat zal nooit kunnen door allerlei wizards op de gebruiker los te laten die het werk uit handen nemen. Niet alleen ontneem je daarmee de nieuwe gebruiker de gelegenheid om zelf te ontdekken hoe het programma werkt, en welke handelingen nodig zijn om het gewenste resultaat te krijgen, een beginnende ontwikkelaar kan in de verleiding komen om de door Access gemaakte objecten niet kritisch genoeg te bekijken, en zo dus niet de best mogelijke database te maken. Je ziet dat m.i. ook al terug in de vragen in het Access forum, en de bestanden die de vragenstellers maken.

Terwijl nieuwe ontwikkelingen in Word dus weinig impact hebben op de bedrijfsvoering (de gebruiker typt nog steeds zijn teksten in) hebben diezelfde ontwikkelingen in Access een behoorlijk grote impact op de manier waarop een gebruiker met het pakket omgaat. Ik vind dus dat ontwikkelingen binnen Access gericht moeten zijn op de serieuze ontwikkelaar, die als doel heeft om goede professionele databases te maken, en niet op de beginnende gebruiker die af en toe eens een gegevensbestandje nodig heeft.

Ik zit niet te wachten op een knop die in één keer vijf tekstvelden aanmaakt waarvan ik de helft eigenlijk niet eens wil hebben, en die ik vervolgens allemaal moet aanpassen omdat de eigenschappen van die tekstvelden niet deugen. Daarentegen zit ik (al jaren overigens) te wachten op de versie die zelf op de achtergrond verwijzingen naar veranderde objecten aanpast. Denk daarbij aan het refereren naar een tekstveld op een formulier in een query; als je naderhand de naam van dat object verandert, werkt je query niet meer. Dat zou een verbetering zijn die ik van harte toejuich! Of een database die zelf bijhoudt waar de gekoppelde bestanden staan, zodat bij het verplaatsen van de database de gekoppelde tabellen nog steeds werken.

Dat gezegd hebbende, is Access 2010 een pakket waar na een (wat mij persoonlijk betreft best lange) periode gewenning best goed mee te werken is. Er zijn interessante nieuwe opties toegevoegd, en andere opties zijn zinvol uitgebreid. En gezien het aantal mensen dat er ondertussen mee werkt, zal Microsoft de ingeslagen weg nog wel een tijdje vol houden. En dat houdt dus in, dat ik in de cursus de komende weken wat vaker de verschillende mogelijkheden van Access 2010 zal laten zien!