

# Access

## **READER t.b.v. BIS**

Versie 4.0  
17 November 2016

Jorg Janssen  
Marco Engelbart  
Alice de Groot  
Misja Nabben  
Peter Cornelissen

© 2016 Hogeschool van Arnhem en Nijmegen

The material contained in this document is proprietary to the named company, the Hogeschool van Arnhem Nijmegen (HAN) and is submitted for informational purposes only. No rights in said material are transferred. This material may not be disclosed, duplicated or otherwise revealed, in whole or in part, without the prior written consent of the HAN.

## Versiehistorie

Versiehistorie			
Datum	Versie	Aangebrachte wijzigingen	Auteurs
17-11-2016	4.0	Combinatie van Access handleidingen + actualisering	CNLP
31-10-2013	3.5	Server 2012 icm MS Access 2013	NBNM
08-11-2012	3.4	Aanpassingen H 2.6; voetnoot toegevoegd ivm gebruik SQL Server 2012 icm MS Access 2010	NBNM
22-08-2012	3.3	Aanpassingen H 2.2.2	NBNM
25-11-2011	3.2	Aanpassingen H4.1 en verder voor MS Access 2010 ADP	NBNM
21-11-2011	3.1	Kleine layout aanpassingen	NBNM
01-11-2011	3.0	Aanpassingen t.b.v. MS Access 2010	GRTDA, NBNM
01-09-2008	2.0	Aanpassingen t.b.v. MS Access 2007	TRSKD, GRTDA
01-09-2006	1.0		JSNO, EGBC

## Inhoudsopgave

1	Inleiding .....	4
2	Maken eerste project .....	5
3	Instellen connectie naar de database .....	8
3.1	Werken met tabellen .....	15
3.2	Tabel ontwerp modus .....	16
3.3	Werken met formulieren .....	20
4	Eenvoudige formulieren .....	24
4.1	Enkelvoudig onderhoudsformulier .....	24
4.1.1	Recordbron .....	25
4.1.2	Besturingselementbron .....	26
4.2	List controls: Combobox en Listbox .....	26
4.2.1	Rijbron .....	27
4.2.2	Weergeven van namen en omschrijvingen .....	28
4.3	Subformulieren .....	29
4.4	Unbound formulieren en controls .....	31
4.5	Zoekmogelijkheden .....	32
4.5.1	Standaard Access zoekfunctie .....	32
4.5.2	Navigeren m.b.v. een listbox of combobox .....	32
4.5.3	Selectie van records in een subformulier .....	34
4.5.4	Records filteren .....	35
5	Formulierontwerp .....	38
5.1	Processen .....	38
5.2	Foreign Key's .....	39
5.2.1	Toevoegen foreign key record .....	41
5.3	Foreign key verwijzingen .....	42
5.4	Afronding .....	46
6	Geavanceerde formulieren .....	47
6.1	Openen van andere formulieren en rapporten .....	47
6.2	Embedded SQL .....	48
6.3	Meerdere niveau's van subformulieren .....	50
6.4	Bewaken van beperkingsregels .....	52
6.5	Keuzerondjes .....	53
6.6	Afbeeldingen .....	55
7	Rapporten .....	57
7.1	Eenvoudige rapporten .....	57
7.2	Sorteren en groeperen .....	58
7.3	Openen van een rapport vanaf een formulier .....	60
7.4	Subrapporten .....	61
8	Menu's en opstartopties .....	63
8.1	Menu's .....	63
8.1.1	Het maken van een lint via de Access Opties .....	63
8.1.2	Het maken van een lint in XML en openen vanuit VBA .....	66
8.2	Opstartopties .....	67
	Bijlage 1: Stappenplan voor het maken van een formulier .....	68
	Bijlage 2: Database-schema Muziekdatabase .....	70

# 1 Inleiding

Deze reader gaat over het ontwikkelen van applicaties. De applicaties waar het hier om gaat hebben altijd een database als onderliggend opslagsysteem. De belangrijkste functies van de applicaties betreffen de invoer en beheer van gegevens in de database. Het vakgebied van het ontwerpen en bouwen van databases en bijbehorende applicaties wordt *informatiesysteemontwikkeling* genoemd.

We beginnen deze reader met de uitleg over hoe een eerste project opgezet kan worden en hoe een connectie naar de SQL-Server gelegd kan worden. Vervolgens wordt er dieper ingegaan op het bouwen van formulieren.

De voorbeelddatabase waar in deze reader gebruik van wordt gemaakt heeft als basis de muziekdatabase, zoals beschreven in het boek *Relationele Databases & SQL* van Wiegerink c.s. Een database-schema van de complete database vind je achterin deze reader.

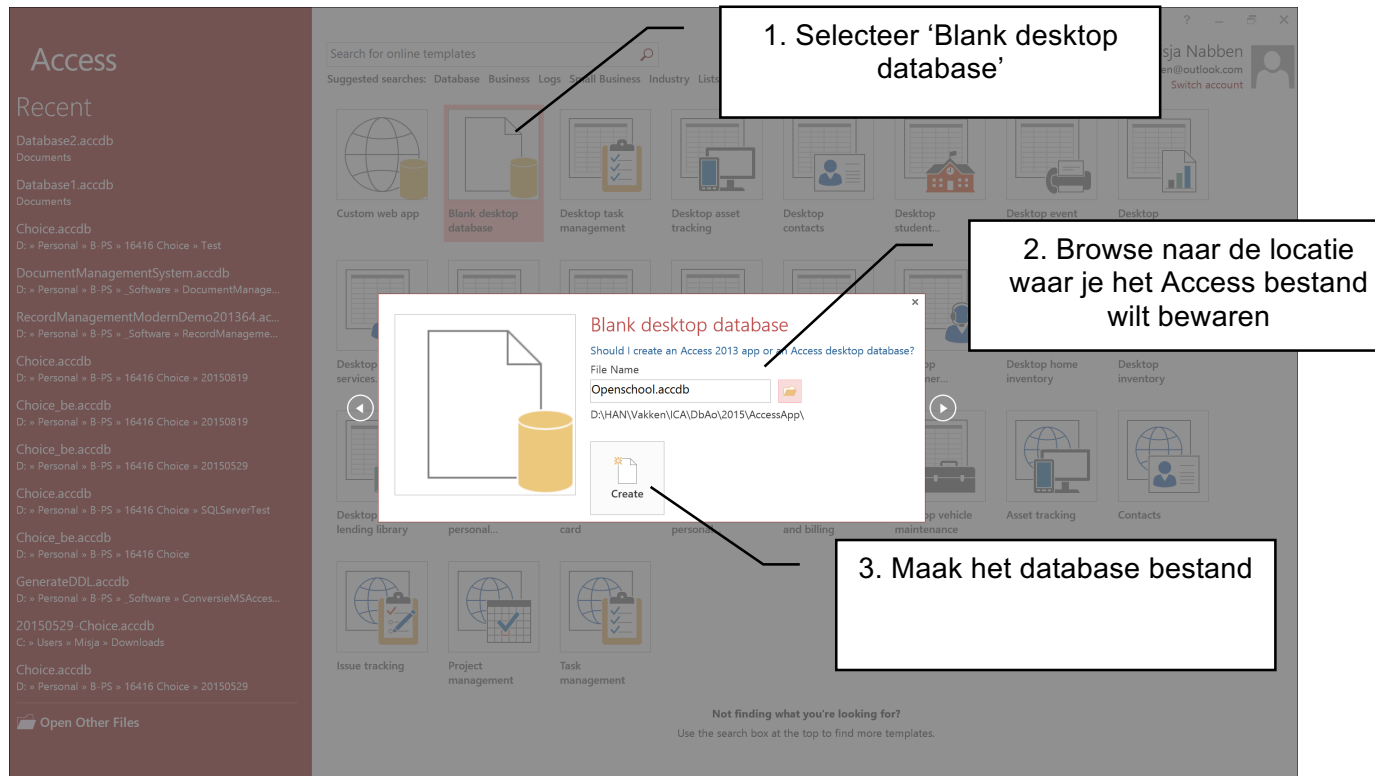
Het tool waar in deze syllabus gebruik van wordt gemaakt is Microsoft Access. De voorbeelden zijn gemaakt en getest in Microsoft Access 2013, maar zijn allemaal mogelijk in versies vanaf Access 2003.

Microsoft Access bestaat van oudsher uit twee onderdelen: een relationeel database management systeem (RDBMS), genaamd de Jet Engine, en een gedeelte om een client applicatie mee te bouwen. Deze syllabus gaat voornamelijk over het tweede onderdeel. De Jet Engine is een prima RDBMS voor kleine, eenvoudige applicaties waar door een beperkt aantal mensen tegelijk gebruik van wordt gemaakt. De Jet Engine ondersteunt geen triggers en stored procedures. Je kunt in de Jet Engine tabellen aanmaken, maar je kunt deze ook verbinding laten maken met een willekeurig ander RDBMS via ODBC. Vanaf 2002 is Access ook bruikbaar zonder de Jet Engine. Je bent dan verplicht gebruik maken van Microsoft SQL Server als RDBMS. Een applicatie met Jet Engine maak je door een nieuw ACCDB-/MDB-bestand te maken. Een applicatie zonder Jet Engine maak je door een Access Data Project, een ADP-bestand te maken. Vanaf 2013 is het echter niet meer mogelijk om een ADP project te maken. We maken dan ook gebruik van het ACCDB bestandstype. In de syllabus houden we zoveel mogelijk rekening met beide varianten, indien noodzakelijk geven we van beide mogelijkheden een voorbeeld en separate uitleg. Als er niets over wordt vermeld, gelden uitleg en voorbeeld voor beide varianten.

Deze reader is een eerste opstap in het gebruik van Access als frond-end applicatie-ontwikkelomgeving. Natuurlijk zijn er nadelen aan deze methode maar Access leent zich goed om een applicatie in korte tijd op te zetten.

## 2 Maken eerste project

Het maken van een eerste project in MS Access gaat volgens het doorlopen van een aantal stappen. Volg onderstaande stappen om het projectbestand aan te maken.

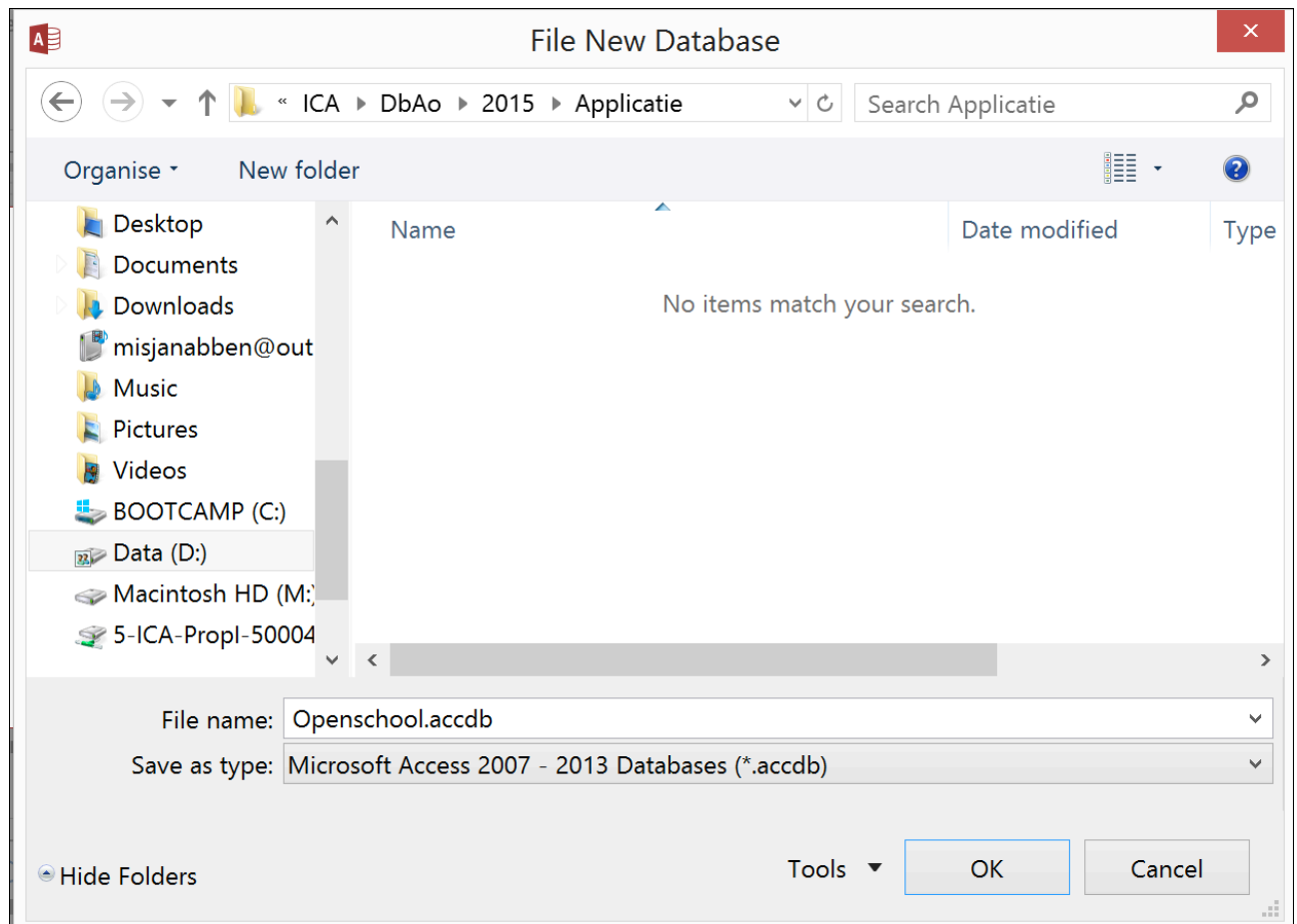


### 1. Selecteer 'Blank desktop database'

We creëren een 'Blank desktop database' project door in bovenstaand scherm de juiste optie te selecteren.

### 2. Browse naar de locatie waar je het Access bestand wilt bewaren

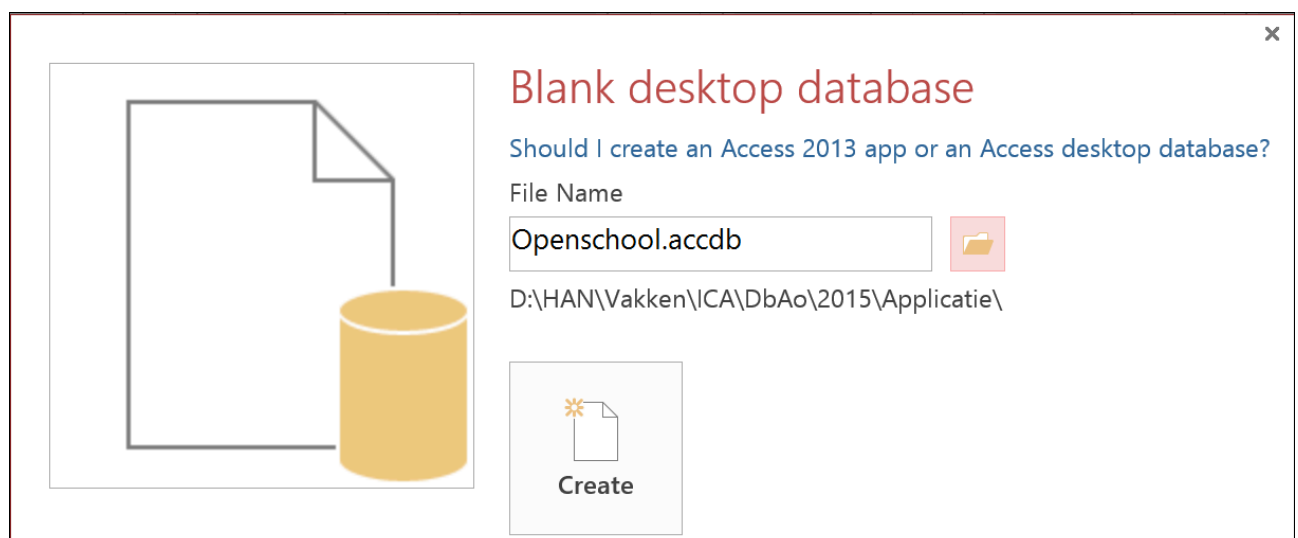
We gaan nu een leeg project (lege database) aanmaken door eerst de locatie aan te geven waar deze bewaard moet worden en selecteren tegelijkertijd het type 'Microsoft Access 2007-2013 Database (\*.accdb)' in onderstaand scherm.



Na 'Ok' komen we weer terug in het beginscherm.

### 3. Maak het database bestand

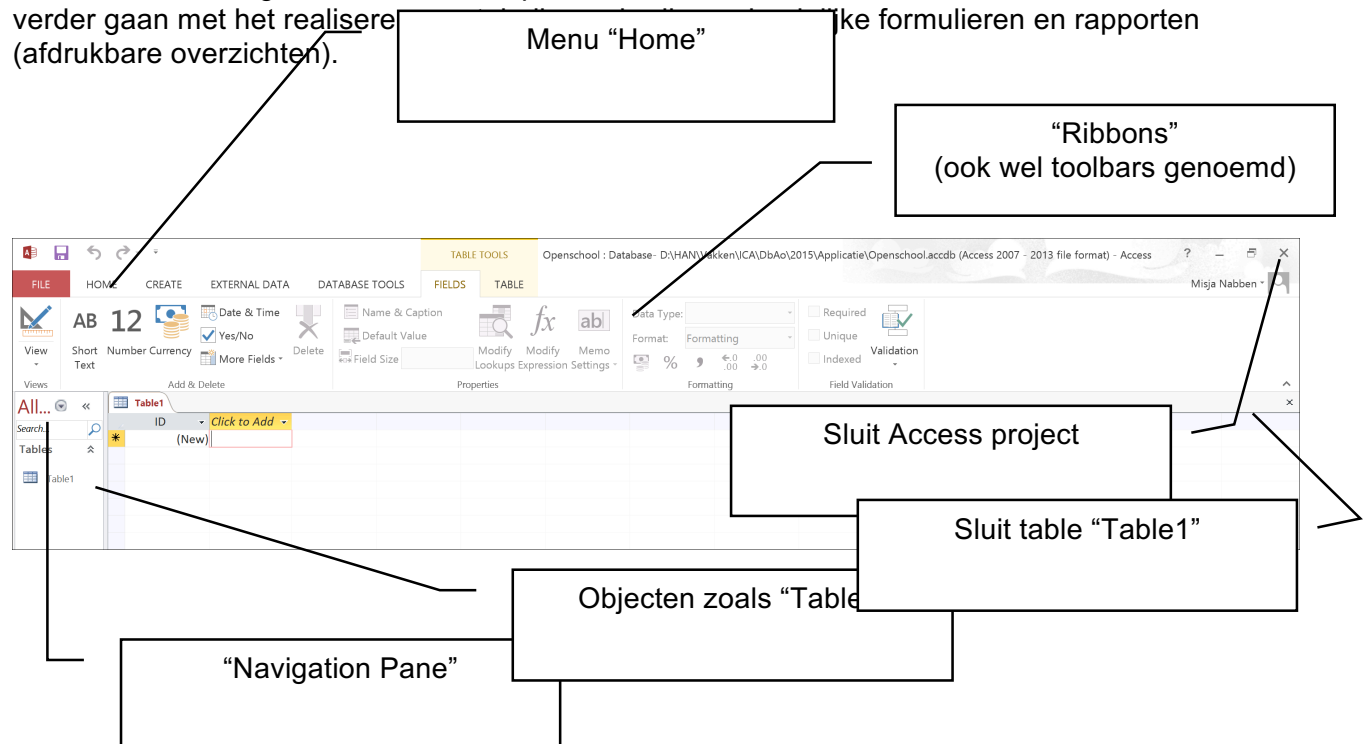
We kiezen vervolgens voor 'Create'.



Nu wordt een leeg **MS Access ACCDB** bestand aangemaakt. Met dit bestand gaan we vervolgens verder. Het gehele MS Access project, bestaande uit projecteigenschappen, formulieren, rapporten en code modules, wordt bewaard in één fysiek bestand.

**Korte introductie van het applicatie platform:**

Zodra het bestand gemaakt is komen we op het **hoofdscherm** van MS Access. Vanaf hier kunnen we verder gaan met het realiseren van de formulieren en rapporten (afdrukbare overzichten).



Het linkergedeelte, de **Navigation Pane**, worden alle Access objecten, zoals o.a. **Tables** (tabellen), **Forms** (formulieren) en **Reports** (rapporten) getoond die onderdeel uitmaken van het Access project (hier Openschool.accdb).

Bovenin beeld bevindt zich het applicatiemenu, vanaf het menu **Create** kunnen we de diverse objecten gaan ontwikkelen. Onder het menu bevindt zich de **Ribbon**. Afhankelijk van de gekozen menuoptie verschijnt er een bijbehorende Ribbon met daarop de diverse knoppen.

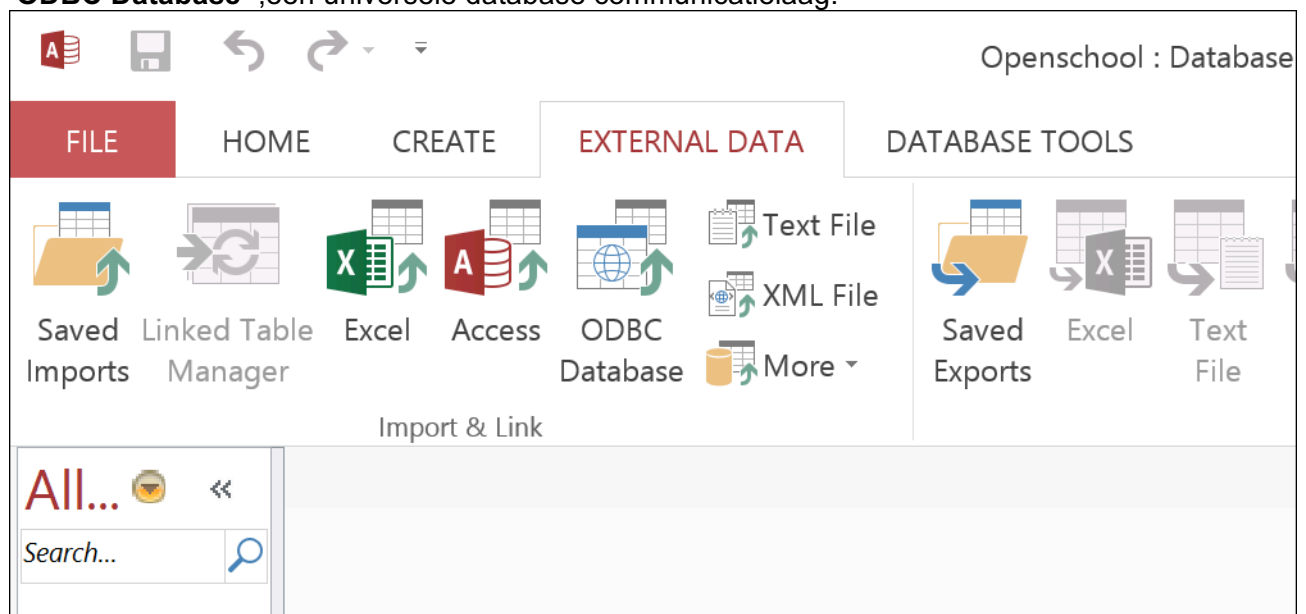
Zoals je ziet wordt er standaard 1 tabel genaamd 'Table1' geopend. Access biedt de mogelijkheid om een Access database te maken wat vergelijkbaar is met een MS SQL Server database alleen wat minder robuust. Voor eenvoudige applicaties met een klein aantal gebruikers zou dat prima kunnen werken. Binnen dit project maken we daar geen gebruik van aangezien we SQL Server als database platform gebruiken.

### 3 Instellen connectie naar de database

De volgende stap is het verbinden met de juiste SQL Server database zodat we gebruik kunnen maken van de tabellen van MS SQL Server. We kunnen daarvoor de tabel 'Table1' negeren en deze sluiten.

ACCDB projecten zijn uitermate geschikt voor het bouwen van een client-server oplossing. De MS Access applicatie wordt de nieuw te bouwen client-applicatie. De server bestaat uit een MS SQL Server 2016 database.

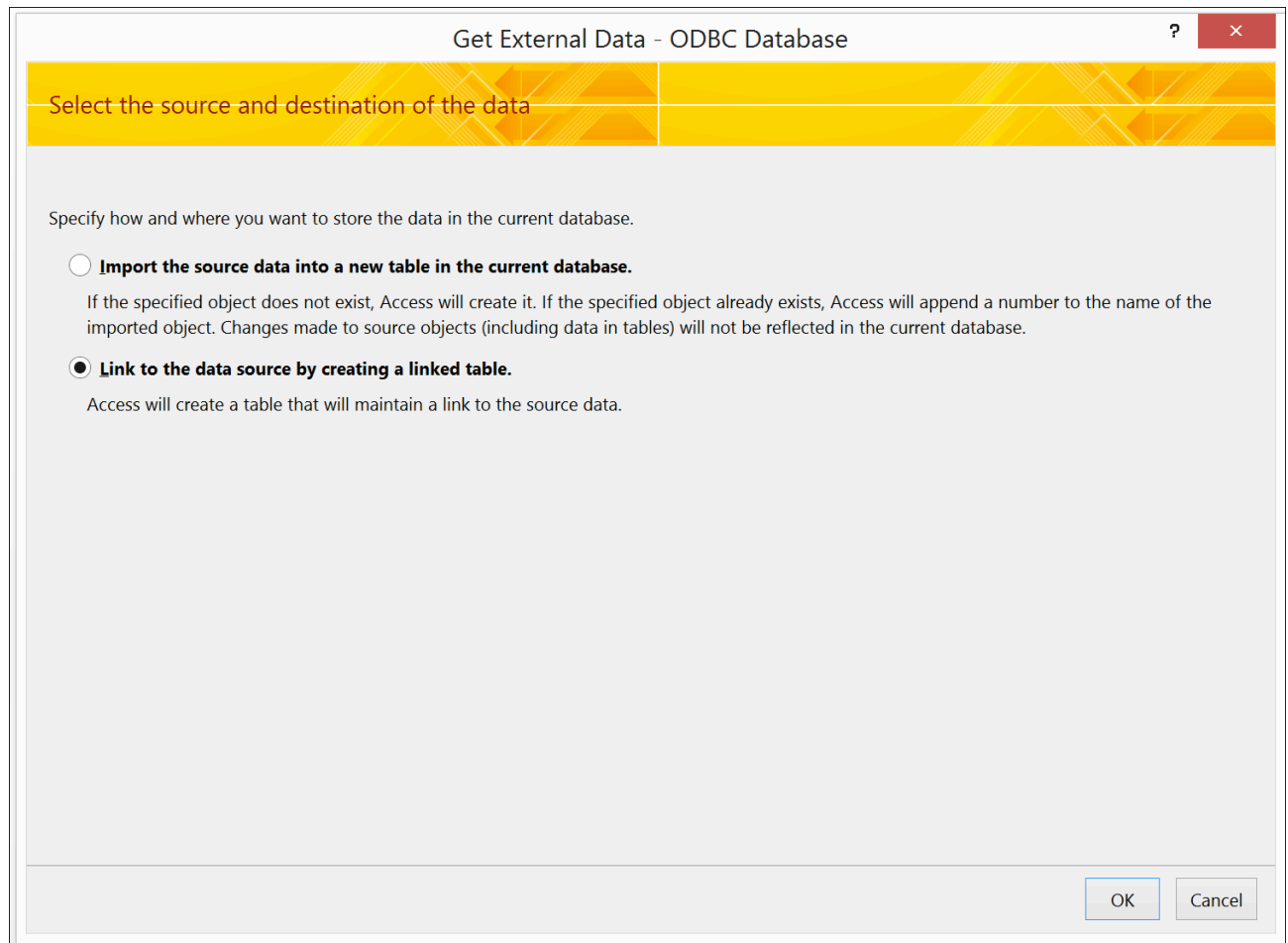
Voor het maken van een verbinding kiezen we in hoofdmenu onder menu '**External Data**' voor optie '**ODBC Database**'<sup>1</sup>, een universele database communicatielaag.



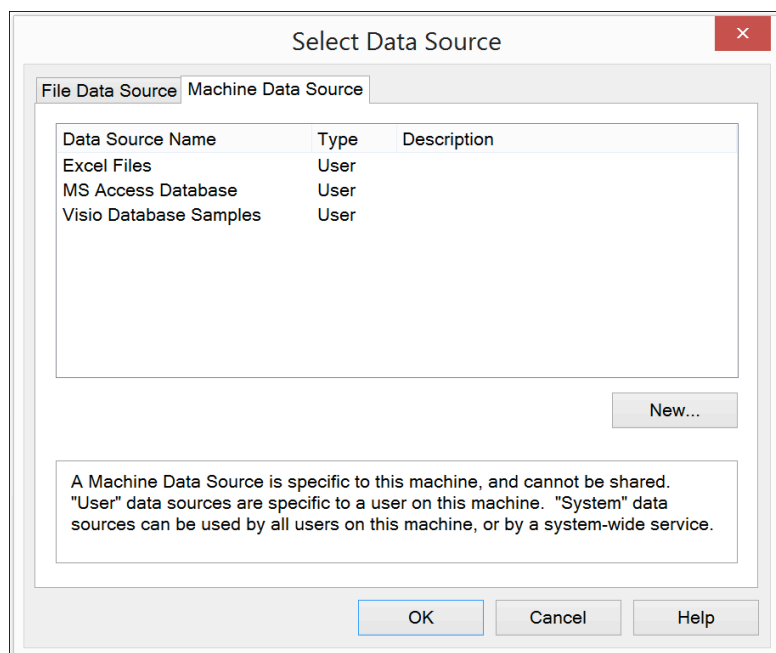
Vervolgens gaan we ervoor kiezen naar de database te linken met de optie '**Link to the data source by creating a linked table**'. Dit zorgt ervoor dat we de beschikking krijgen over een link naar de tabellen in SQL Server en dus geen kopie van de tabellen (de bovenste optie).

<sup>1</sup> Voor meer uitleg over ODBC zie MSDN – ODBC ([http://msdn.microsoft.com/en-us/library/aa266544\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa266544(v=vs.60).aspx) )

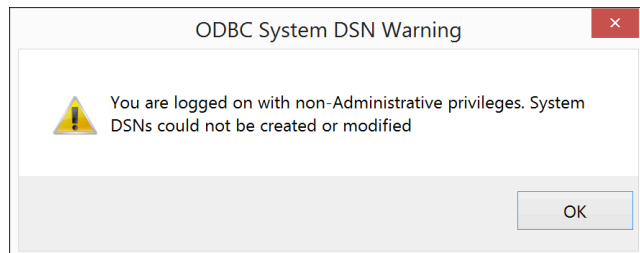




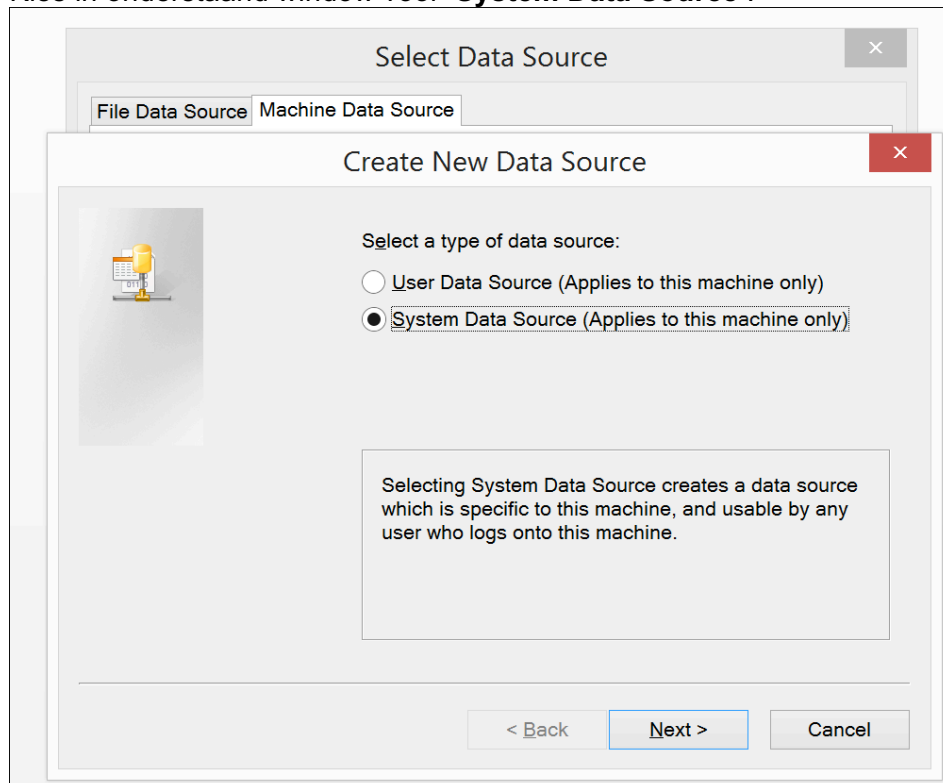
Kies vervolgens voor '**Machine Data Source**' en de optie '**New**'.



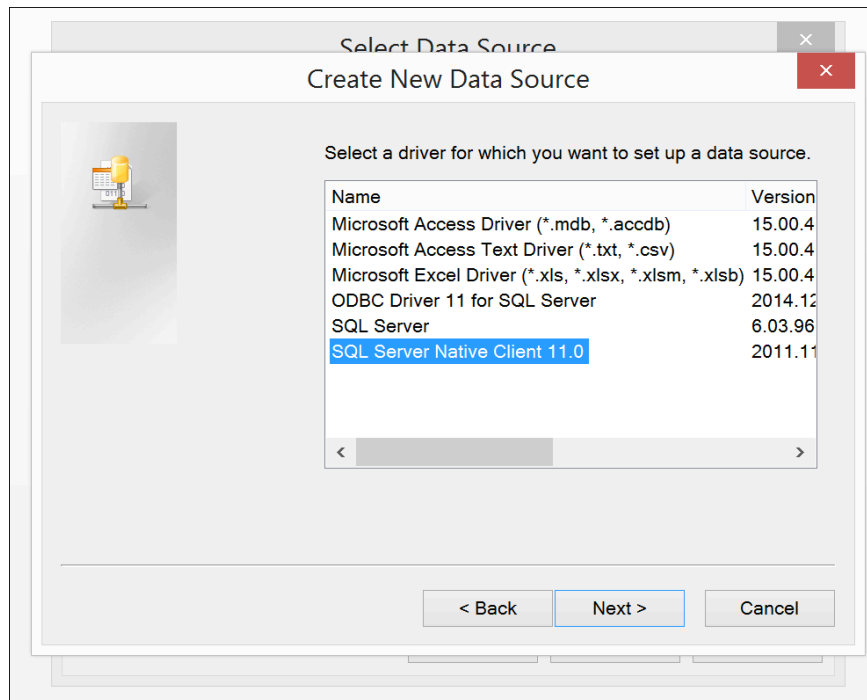
*Let op: indien je onderstaande melding krijgt moet je MS Access opnieuw als administrator opstarten.*



Kies in onderstaand window voor '**System Data Source**'.

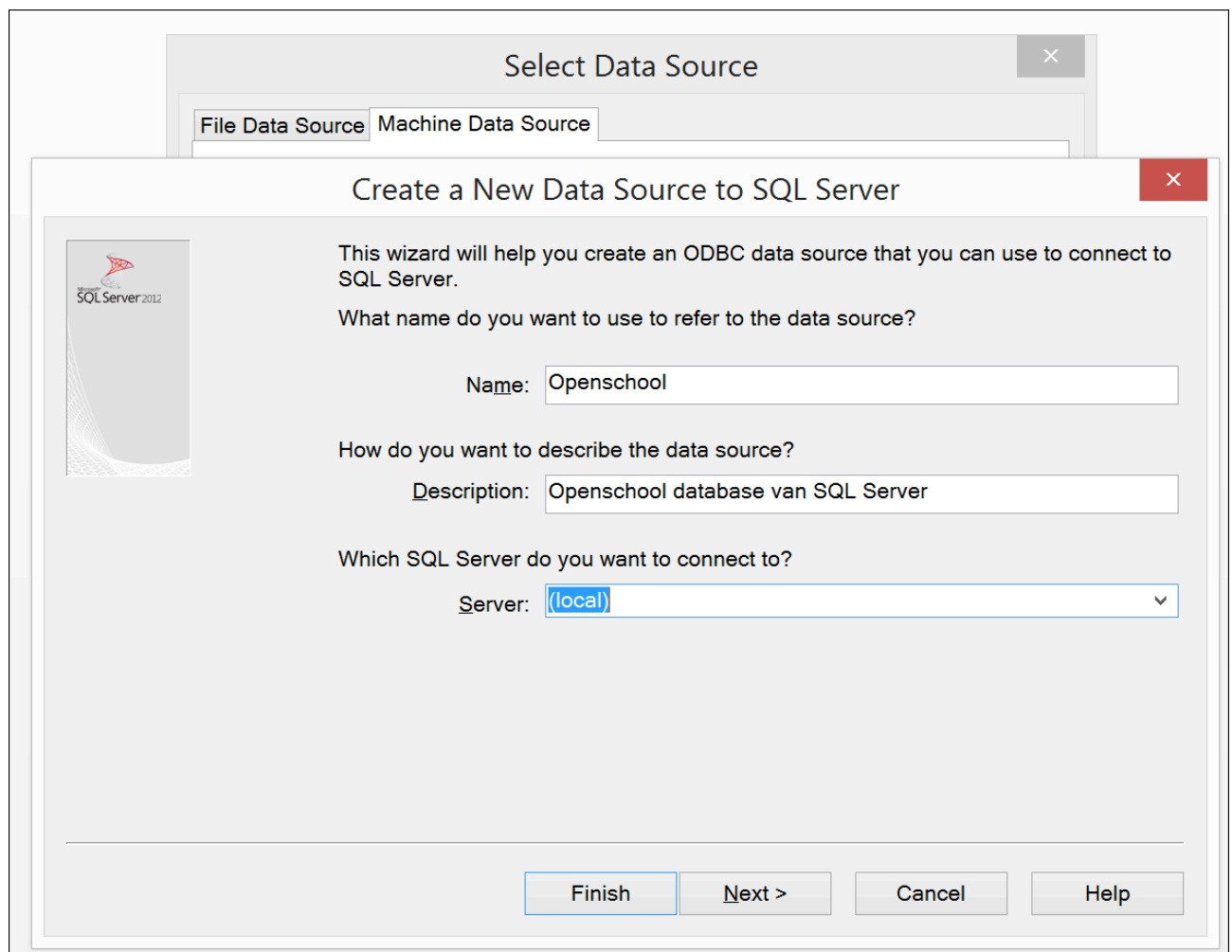


Gevolgd door '**SQL Server Native Client**', bij voorkeur de hoogste versie (hier 11.0).



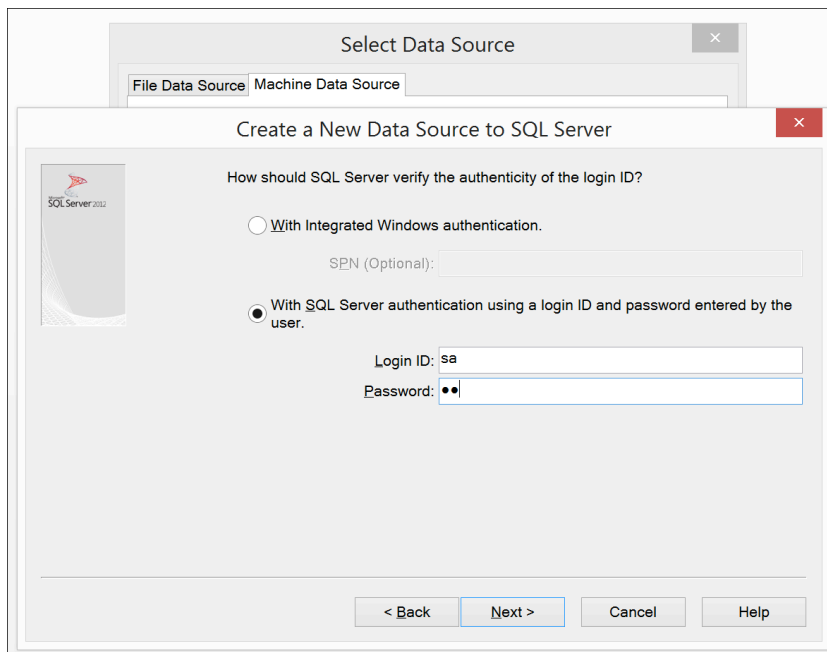
Druk Next en Finish.

Nu gaan we de gegevens van de connectie instellen:



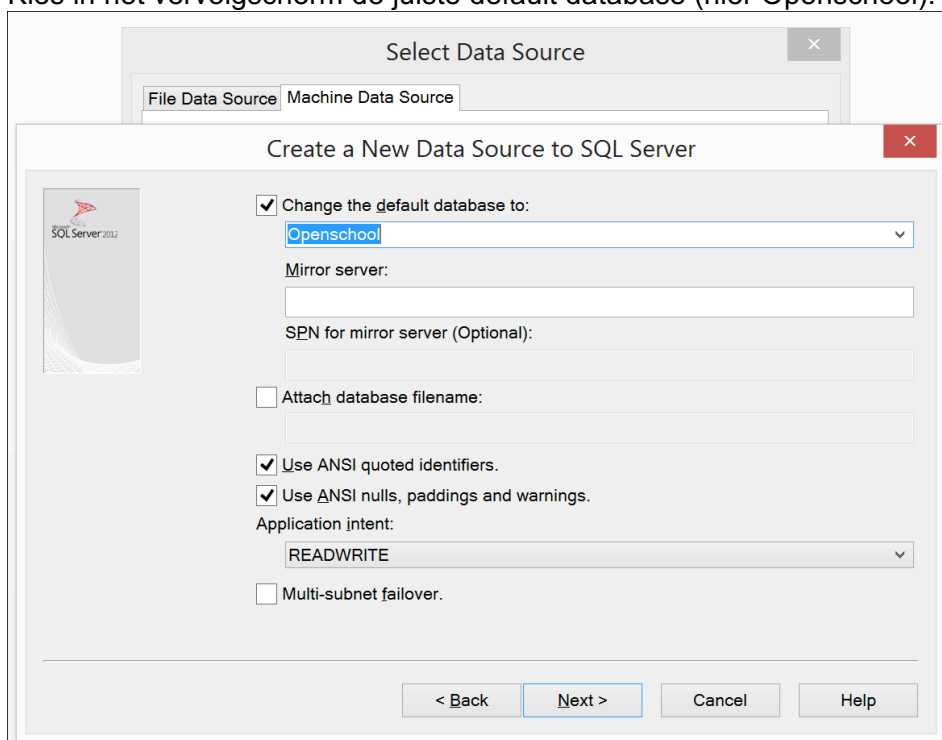
Hier vullende we de naam (zelf te bepalen) van de data source connectie in en de omschrijving en server naam.

We kiezen nu voor **server '(local)'** maar je machinenaam werkt doorgaans ook.  
Let op: voor SQLEXPRESS editie is de server name zeer waarschijnlijk **'(local)\SQLEXPRESS'**,  
check eventueel de verbinding zoals je die in SQL Server Management Studio gebruikt.  
Klik vervolgens op Next.

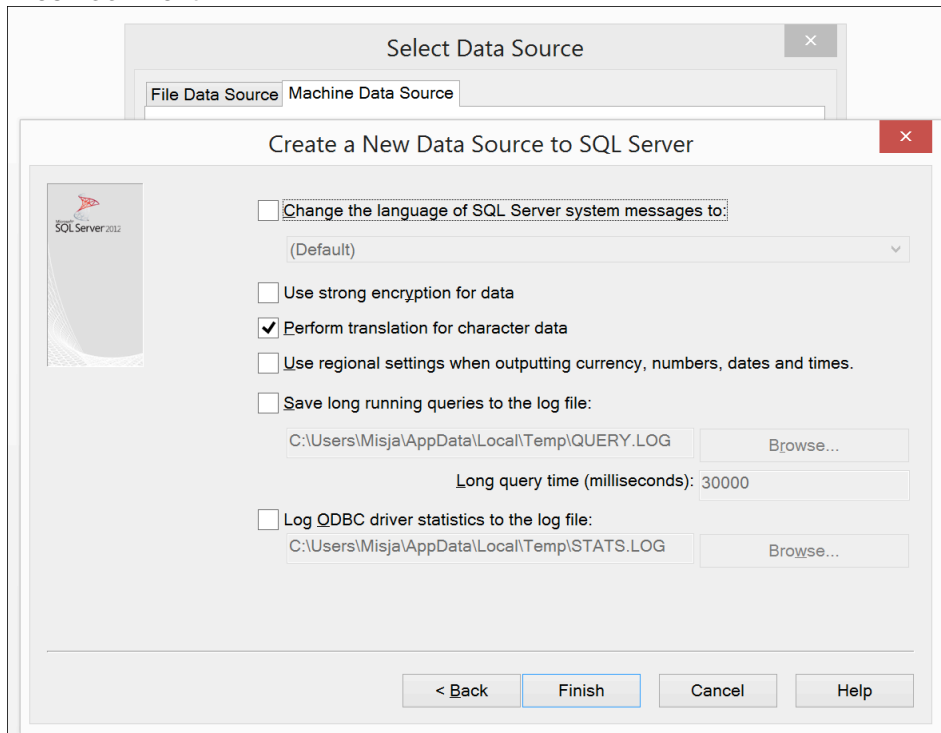


Gebruik hier een manier van inloggen, vergelijkbaar met de wijze van een database connectie maken in SQL Server Management Studio.  
Kies voor Next.

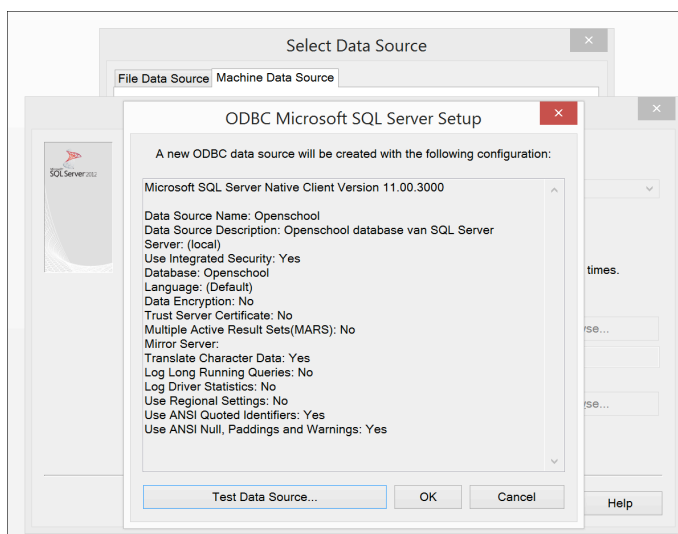
Kies in het vervolgscherm de juiste default database (hier Openschool):



Kies voor Next.



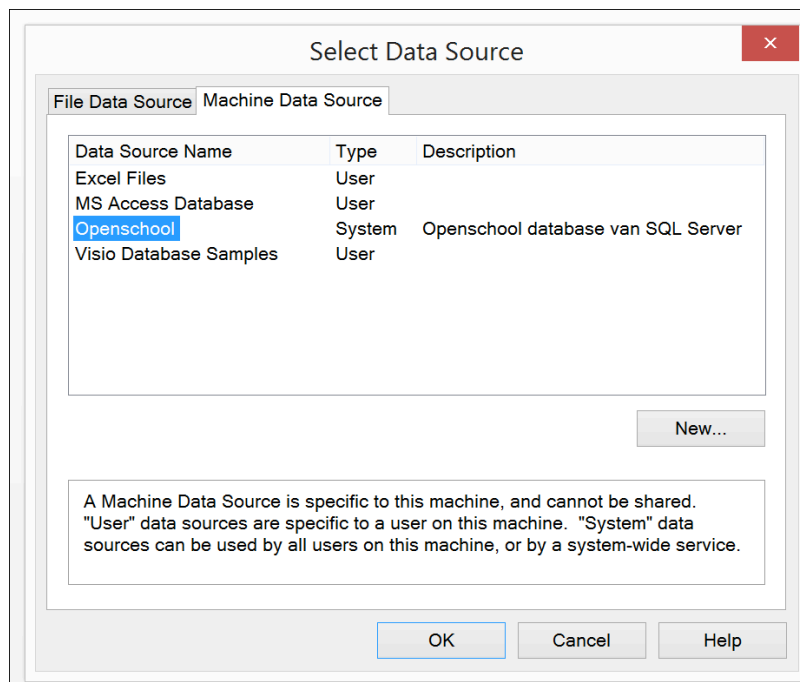
Laat alle standaardinstellingen staan en kies voor Finish.



Test de connectie met behulp van de 'Test Data Source' knop.

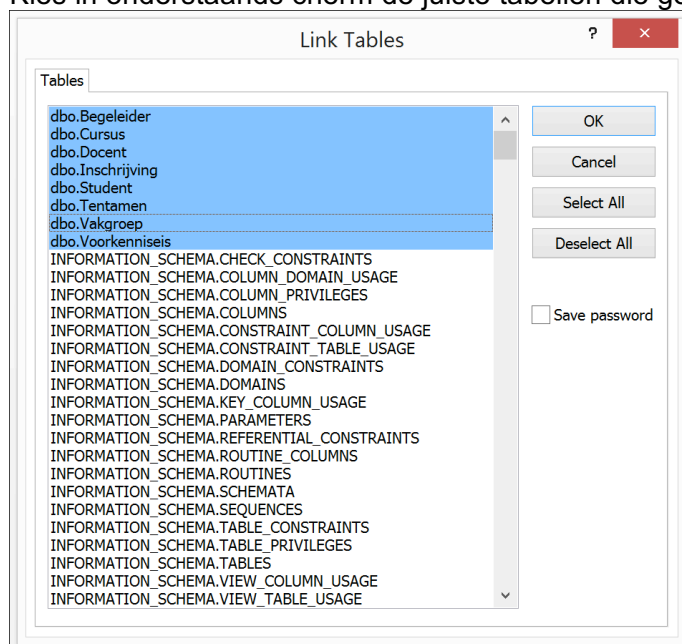
Merk op dat de database wel moet bestaan! Deze hebben we al eerder gecreëerd, zie eerdere workshops. Kies voor Ok, gevolgd door Ok.

Na het succesvol connecten van de database kom je in onderstaand scherm.



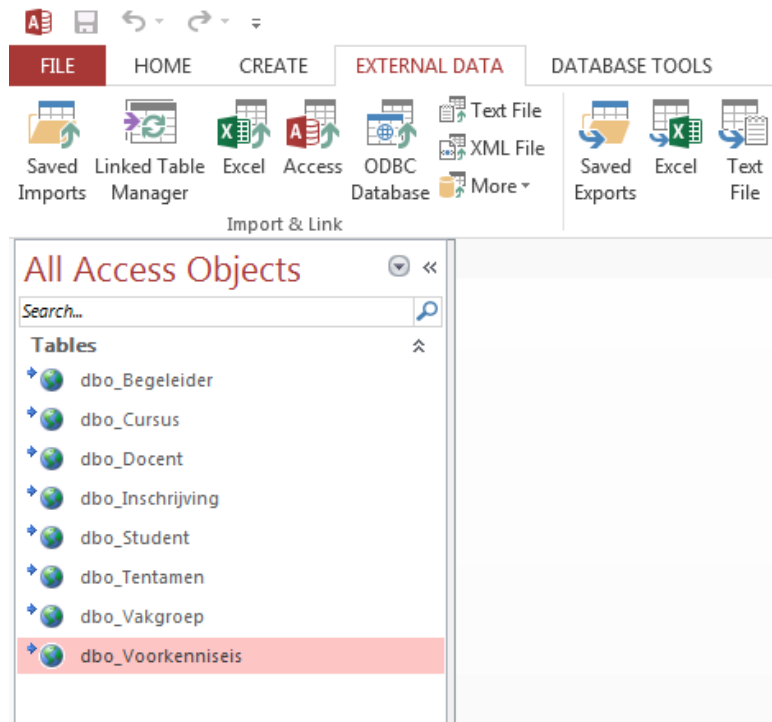
De data source (met daarin alle connectie instellingen) naar de juiste database is nu gemaakt, deze kiezen we dus ook en vervolgens voor Ok.

Kies in onderstaands scherm de juiste tabellen die gelinkt moeten worden:



Kies voor Ok.

Als het goed is verschijnt nu onderstaand scherm:



De connectie met de SQL Server database is nu geslaagd!

*Mocht je nu later de connectie opnieuw in willen stellen kan dat middels de menu-actie 'External Data' gevolgd Linked table Manager'.*

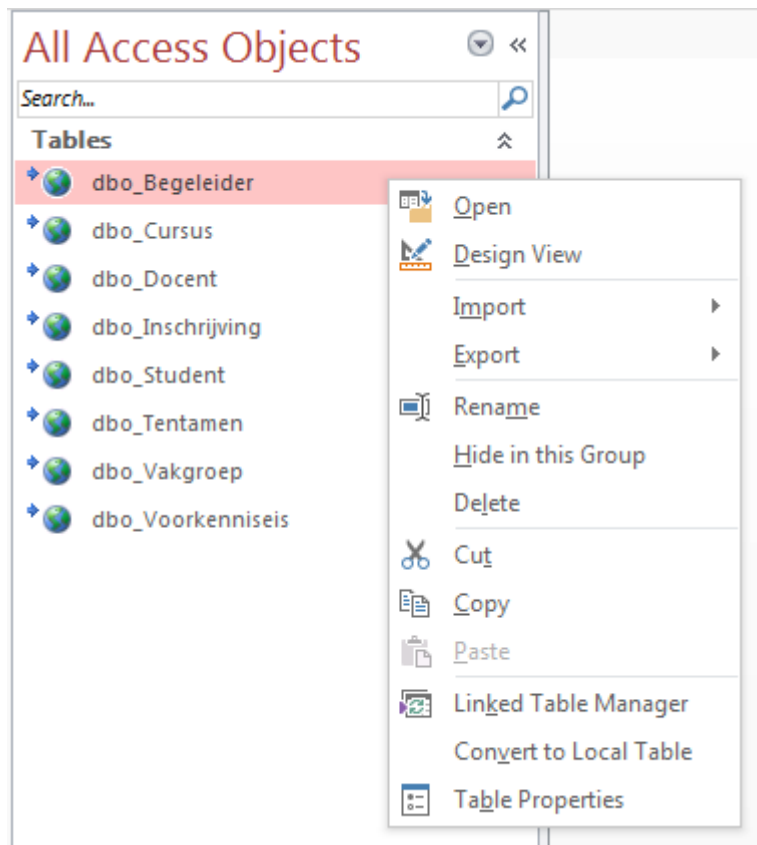
### 3.1 Werken met tabellen

We zien nu alle tabellen zoals deze aangemaakt zijn in de SQL Server database, in ons geval de Openschool. Let wel, deze tabellen zijn dus géén kopie van de tabellen uit de database maar het zijn de originele tabellen. Vergelijkbaar met zoals je in SQL Server Management Studio de tabellen kunt bekijken.

Het aanmaken, wijzigen of verwijderen van tabellen gaat nu niet via MS Access. Zodra we de database van structuur moeten wijzigen doen we dat met SQL Server Management Studio.

Allereerst gaan we zorgen dat de naam van iedere linked table hernoemen naar de originele tabel. Zo moet dbo\_Begeleider worden hernoemd in Begeleider. Dit scheelt ons later nogal wat typewerk.

Via rechtermuis komen we bij de optie om de naam van de link te wijzgen via Rename:



**Doe dit vervolgens voor iedere tabel.** Aan het einde houd je de volgende tabelnamen over.

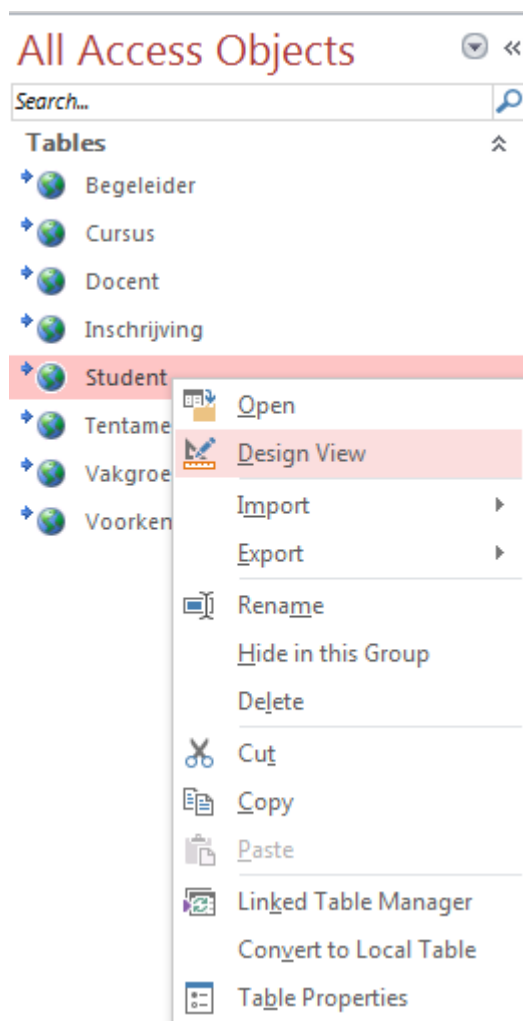


Nogmaals; we wijzigen hiermee alléén de naam van de link en niet de uiteindelijke tabelnaam in SQL Server.

### 3.2 Tabel ontwerp modus

Voor het bekijken van de structuur (oftewel definitie) van een tabel is het soms wel gemakkelijk om dat even snel in MS Access te kunnen bekijken. Dat kan door het selecteren van een tabel, vervolgens een rechtermuis-klik en dan te kiezen voor 'Design View'.





Daarna zien we de tabeldefinitie zoals deze tabel reeds in MS SQL Server aangemaakt is.

### Field Properties

This property cannot be modified in linked tables.

Indexes: Student

Index Name	Field Name	Sort Order
PK_Student	nr	Ascending

Index Properties

Primary	Yes
Unique	Yes
Ignore Nulls	No

This property cannot be modified in linked tables.

Nogmaals, wijzigen van de tabeldefinitie doen we niet middels deze gebruikersvriendelijke schermen (grafische interface) maar altijd via DDL scripts binnen Management Studio!

We sluiten de openstaande schermen weer om vervolgens eens naar de inhoud van de tabellen te kijken.

### Tabel datasheet

Dubbelklik nu op de tabel 'Student. We krijgen nu een datasheet weergave van de tabel 'Student' met daarin alle records uit deze tabel. Zie onderstaande afbeelding.

Student		
nr	naam	mentor
1	Berk	DAT
2	Tack	DAT
3	Bos	COD
4	Eik	
*		

Hiermee kunnen we dus heel snel kijken welke gegevens er in de tabel staan.

Ook kunnen we in deze datasheet records wijzigen, nieuwe records aanmaken of bestaande records verwijderen.

De onderste lege regel, beginnend met een \*, is gereserveerd voor het aanmaken van een nieuw record.

*		
---	--	--


Hier gaan we nu eens een record toevoegen. Vul wat voorbeeld populatie in, zoals bijvoorbeeld in onderstaand scherm.

Student		
nr	naam	mentor
1	Berk	DAT
2	Tack	DAT
3	Bos	COD
4	Eik	
5	Nabben	
*		

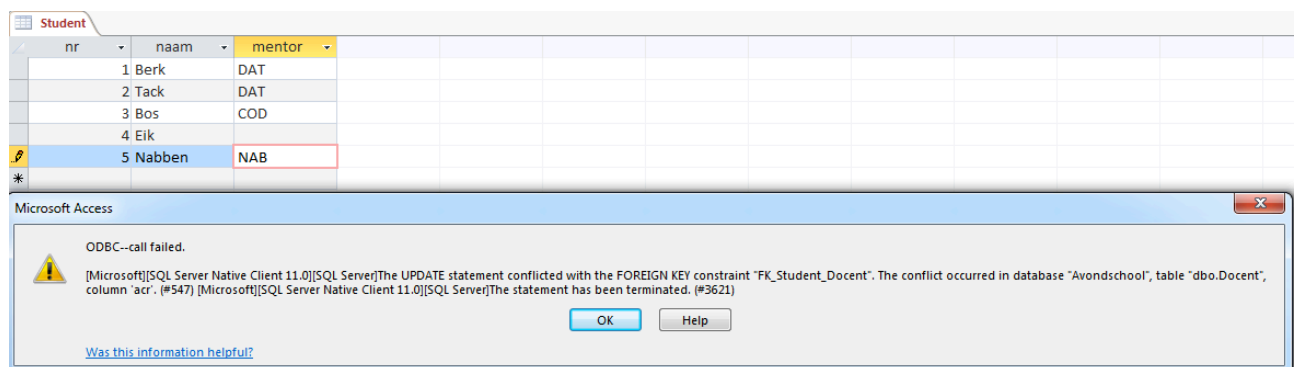
Tijdens het invoeren of bewerken van een record zien we dat er een potlood-icoontje verschijnt. Dit betekent dat het record door de gebruiker gewijzigd wordt. Op dat moment is de wijziging nog niet opgeslagen. Zodra het record verlaten wordt, door bijvoorbeeld te klikken op het vorige record, wordt de wijziging automatisch doorgevoerd. Merk op dat we dus niet expliciet voor bewaren hoeven te kiezen! MS Access zal aan SQL Server doorgeven dat er recordwijziging moet plaatsvinden.

Het verwijderen van een record gaat door middel van het klikken op het grijze blokje aan het begin van het record. Dit is de zogenaamde 'record selector'.

Student		
nr	naam	mentor
1	Berk	DAT
2	Tack	DAT
3	Bos	COD
4	Eik	
5	Nabben	
*		

Daarna kiezen we voor de knop 'Delete' , of voor het indrukken van de 'Delete' knop op het toetsenbord. In beide gevallen wordt om bevestiging gevraagd van de actie.

Je hebt wellicht gemerkt dat het invoeren of wijzigen van records in deze datasheet niet altijd even gemakkelijk gaat. Zo moeten we voor het wijzigen van bijvoorbeeld de mentor een correcte waarde in typen in het betreffende tekstvak. Doen we dat niet dan volgt onderstaande foutmelding.



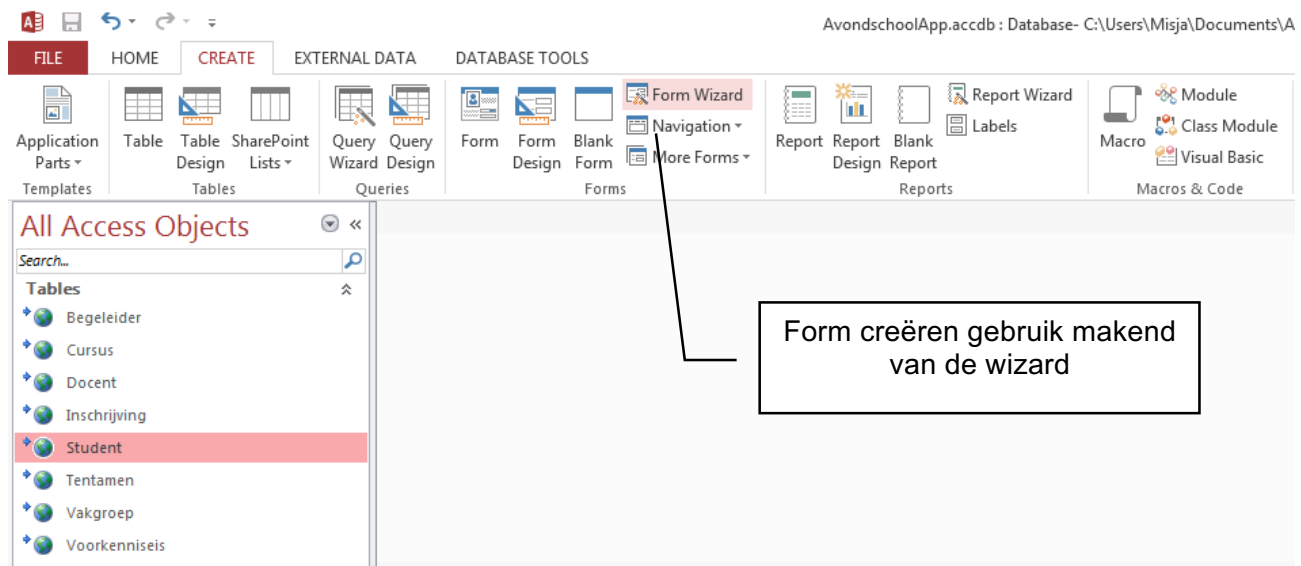
We zouden hier graag een keuze hebben uit de mogelijk te kiezen waarden zodat we geen typfouten kunne maken. Dit gaan we later met behulp van formulieren gebruikersvriendelijker maken.

### 3.3 Werken met formulieren

Het werken in een datasheet is vrij beperkt. Zo hebben we maar de beschikking over een beperkt aantal besturingselementen, ook wel controls genoemd.

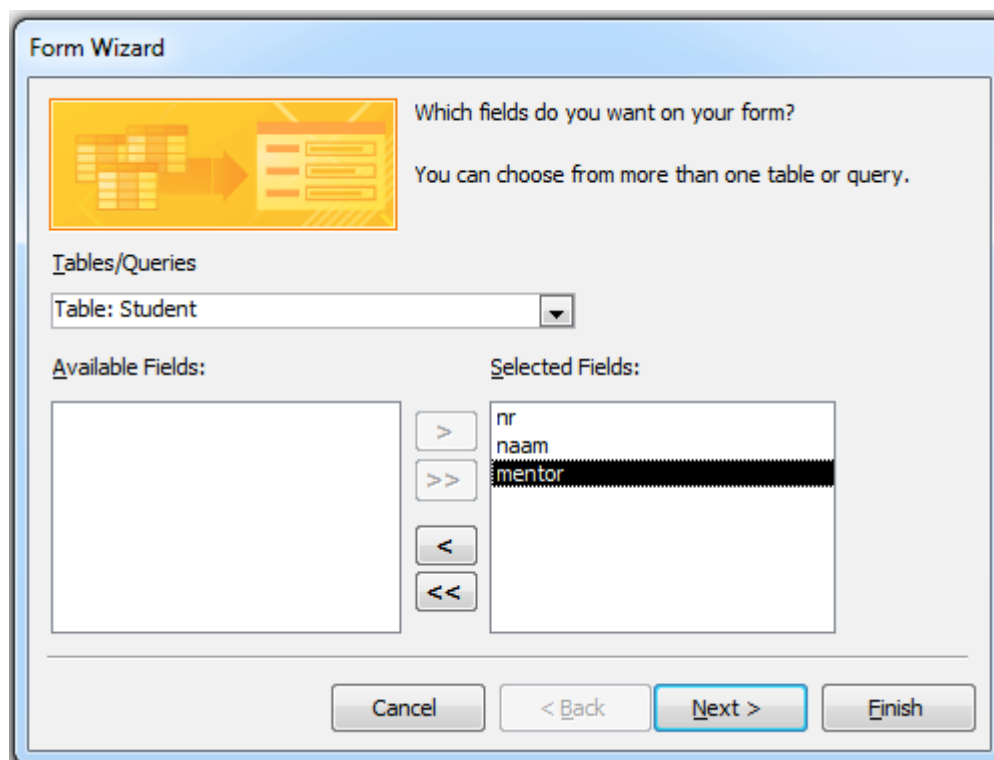
Voor het gebruikersvriendelijk kunnen bewerken van gegevens zijn formulieren beter geschikt. We gaan nu een formulier ontwikkelen voor het bewerken van de tabel 'Student'. Hiervoor zullen we gebruik maken van de Wizard die MS Access ons biedt om zeer snel een eerste versie van een formulier te realiseren.

We sluiten openstaande schermen en kiezen voor de menuoptie 'Create'. In de getoonde ribbon kiezen we voor de knop 'Form Wizard'.

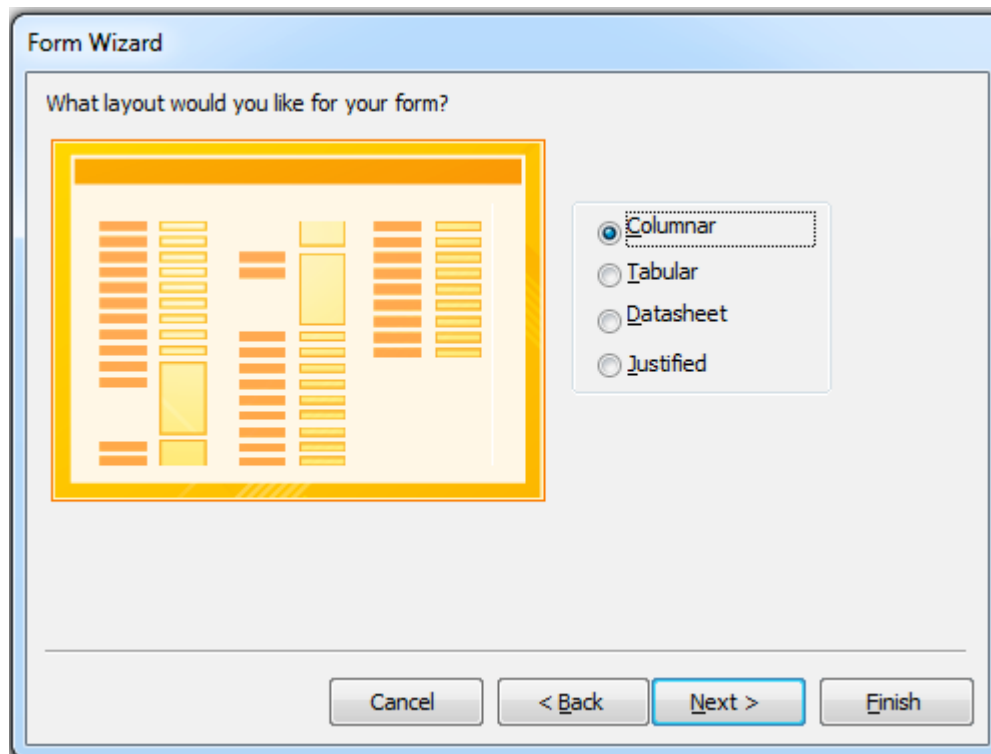


We doorlopen de wizard voor het maken van een gebruikersvriendelijk formulier voor het bewerken van de gegevens van de tabel 'Student'.

Geef de volgende instellingen:

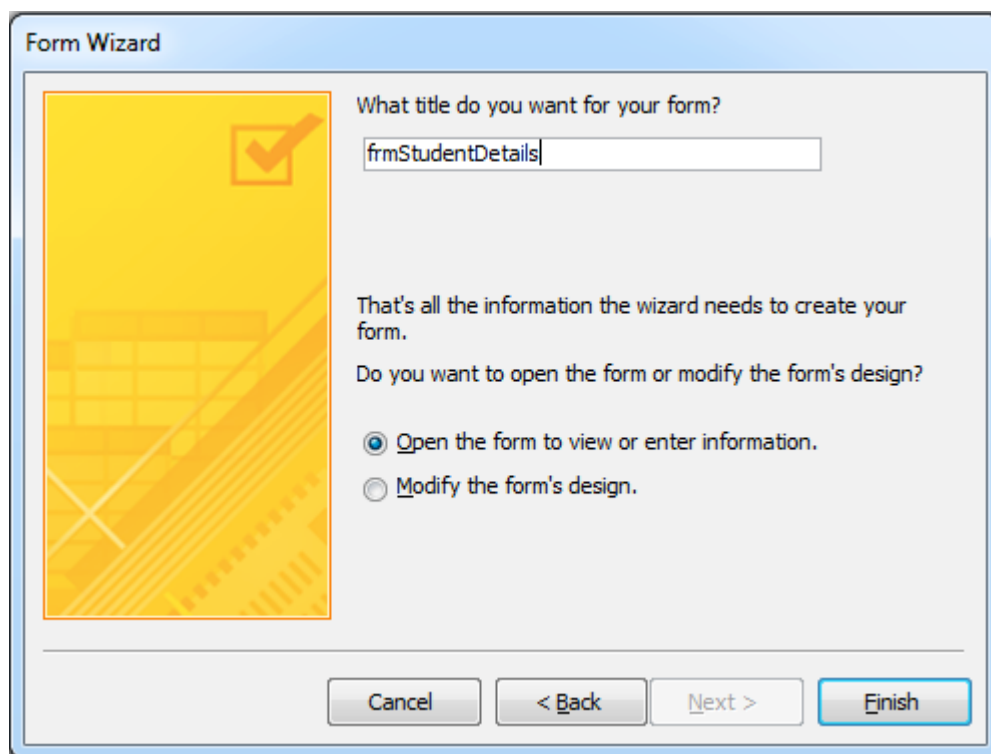


Kies voor 'Next'.



The 'Form Wizard' dialog box is shown with the title 'Form Wizard'. The main question is 'What layout would you like for your form?'. On the left, there is a preview of a form layout with orange and yellow rectangular fields. On the right, there are four radio button options: 'Columnar' (selected), 'Tabular', 'Datasheet', and 'Justified'. At the bottom, there are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

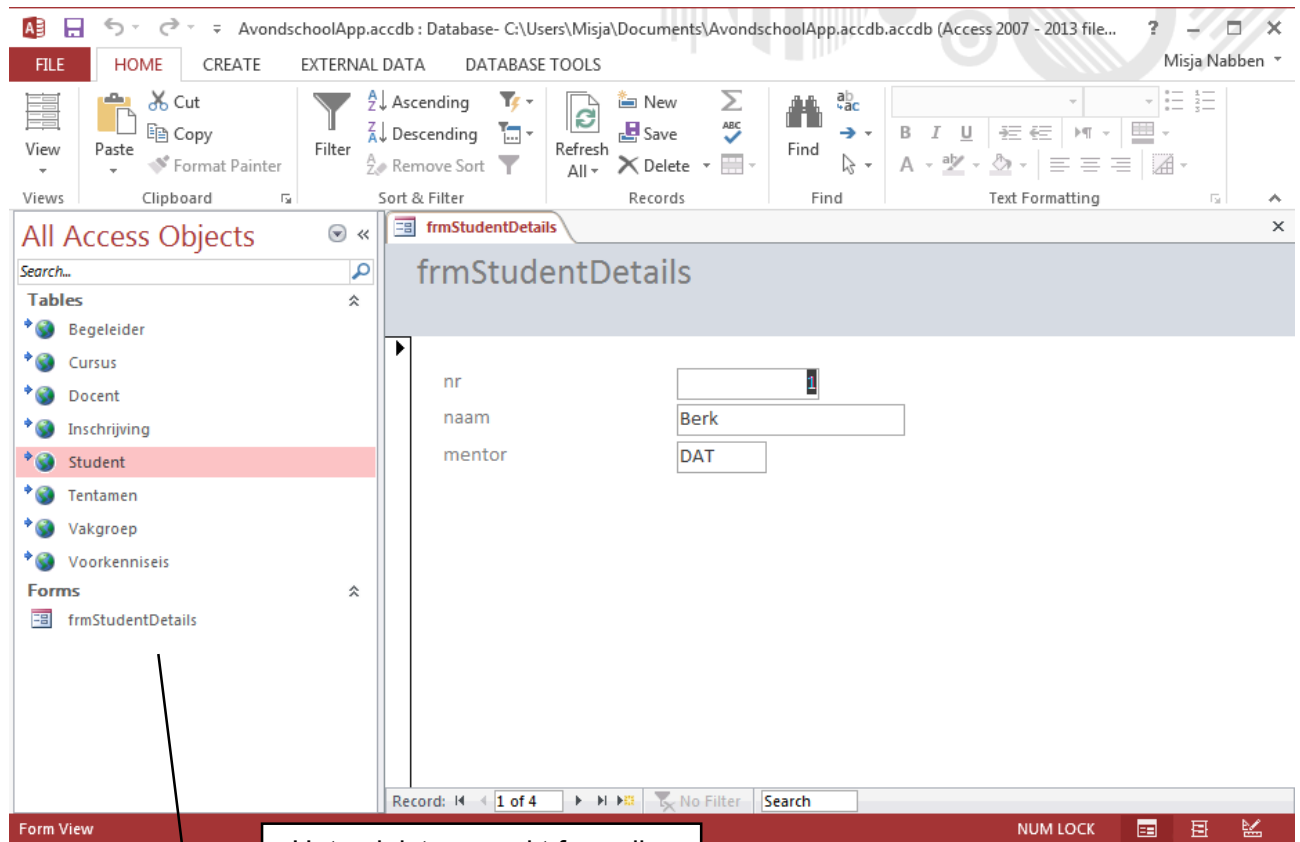
Als opmaakvorm kiezen we voor het 'Columnar' tonen van de kolommen uit de gekozen tabel. Ook in het laatste scherm kiezen we de standaard instellingen: Kies voor 'Next'.



The 'Form Wizard' dialog box is shown with the title 'Form Wizard'. The main question is 'What title do you want for your form?'. On the left, there is a preview of a form layout with a yellow background and a checkmark icon. On the right, there is a text input field containing 'frmStudentDetails'. Below this, there is a message: 'That's all the information the wizard needs to create your form.' followed by the question 'Do you want to open the form or modify the form's design?'. There are two radio button options: 'Open the form to view or enter information.' (selected) and 'Modify the form's design.'. At the bottom, there are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

Kies voor 'Finish'.

We hebben nu razendsnel een werkend formulier gemaakt voor het kunnen bewerken van Student gegevens.



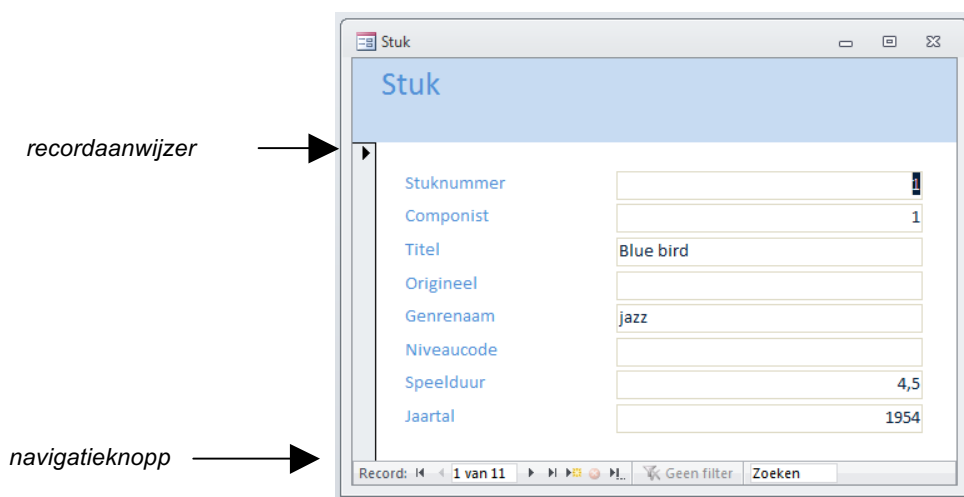
De student gegevens **Het zojuist gemaakt formulier verschijnt hier** delijk formulier gewoon te wijzigen. Tijdens de course gaan we de **gebruik te maken van de wizard**, ontwikkelen.

## 4 Eenvoudige formulieren

Formulieren zijn de schermen waarmee gebruikers interacteren met de applicatie. Ze vormen de interface tussen de gebruiker en de database. De meeste formulieren (in het Engels *forms*) dienen voor het onderhouden van gegevens in de database. Bij het ontwikkelen van onderhoudsformulieren kun je vaak een eerste versie maken m.b.v. de wizards. Meestal voldoet zo'n eerste versie nog niet aan alle wensen en dien je het gegenereerde formulier aan te passen.

### 4.1 Enkelvoudig onderhoudsformulier

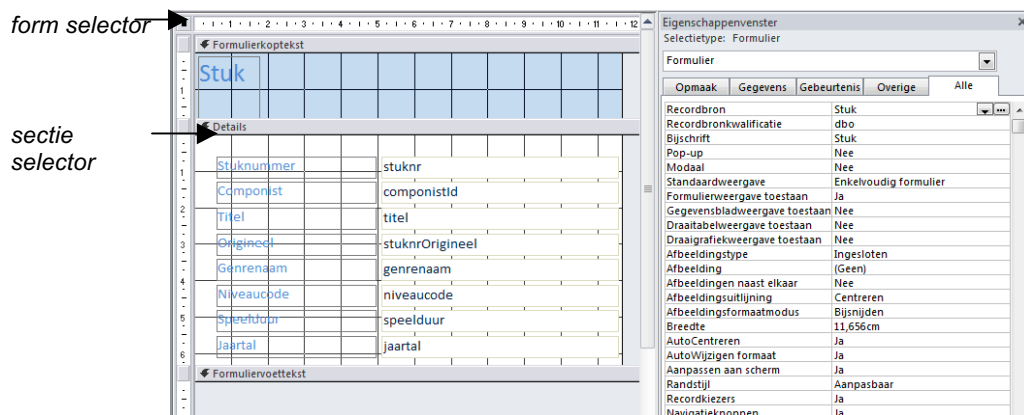
Als voorbeeld bekijken we een formulier voor het onderhouden van de gegevens van muziekstukken. In dit geval is het handig om eerst een formulier met de wizard te ontwikkelen en dat vervolgens aan te passen. Het formulier dat op die manier ontstaat noemen we **frmStuk** en dat ziet er in de formulierweergave als volgt uit:



Formulier frmStuk

Het is een formulier in *enkelvoudige weergave*, hetgeen betekent dat je één record tegelijk op het scherm ziet en dat je met de *navigatieknoppen* onderin het formulieren tussen records kunt bladeren. Een formulier in doorlopende weergave toont meerdere records in een tabel.

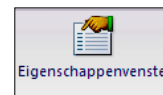
Hetzelfde formulier ziet er in de ontwerpweergave zo uit:



Ontwerpweergave frmStuk



Het formulier bevat een serie *besturingselementen* (in het engels *controls*): *labels* en *tekstvakken* (*textboxes*). Het formulier en zijn besturingselementen hebben *eigenschappen* (*properties*) die kunnen worden ingesteld; als je gebruik maakt van de wizard gebeurt dit deels automatisch.



Een formulier heeft onder meer de volgende eigenschappen:

Recordbron:	Tabel, query of view die de bron vormt van de gegevens voor dit formulier.
Bewerken toestaan:	Geeft aan of de records van de recordbron kunnen worden gewijzigd.
Verwijderingen toestaan:	Geeft aan of de records van uit recordbron kunnen worden verwijderd.
Toevoegingen toestaan:	Geeft aan of er nieuwe records kunnen worden toegevoegd aan de recordbron.
Gegevensinvoer:	Geeft aan of het formulier alleen bedoeld is voor het toevoegen van nieuwe records.
Standaardweergave:	Wijze waarop de records uit de recordbron worden weergegeven; één record tegelijk (Enkelvoudig formulier) of meerdere records in één keer (Doorlopend formulier).
Navigatieknoppen:	Geeft aan of de navigatieknoppen zichtbaar zijn.
Recordkiezers	Geeft aan of de verticale recordaanwijzer balk getoond moet worden.

Bij het formulier **frmStuk** zijn deze eigenschappen als volgt ingesteld:

Recordbron:	Stuk
Bewerken toestaan:	Ja
Verwijderingen toestaan:	Ja
Toevoegingen toestaan:	Ja
Gegevensinvoer:	Nee
Standaardweergave:	Enkelvoudig formulier
Navigatieknoppen:	Ja
Recordkiezers	Ja

#### 4.1.1 Recordbron

De *recordbron* (in het Engels *record source*) is een van de belangrijkste eigenschappen van een formulier in Access. Je geeft er mee aan waar de records die in het formulier staan vandaan komen. Er zijn verschillende mogelijkheden:

Tabel:	Kies één van de tabellen.
Directe SQL:	Je kunt als recordbron direct een SQL

	SELECT opdracht intikken of opbouwen met de query-builder, bijvoorbeeld SELECT * FROM Stuk ORDER BY stuknr.
Access Query:	Alleen in accdb's en mdb's. Kies één van de queries die je gemaakt hebt in het query-gedeelte van Access.
MS SQL View:	Alleen in adp's. Kies één van de views (weergaven) die je gemaakt hebt op de SQL Server.
MS SQL Stored Procedure:	Alleen in adp's. Kies één van de stored procedures die je gemaakt hebt op de SQL Server. Deze stored procedure moet een recordset retourneren.

Je kunt de eigenschap recordbron ook leeg laten. Het formulier is dan niet meer gebonden aan records. Er worden geen gegevens meer opgehaald uit de database en de tekstvakken blijven leeg. Ook de navigatieknoppen hebben geen functie meer. Het formulier heet dan *niet-afhankelijk* of in het Engels *unbound*.

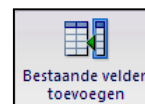
#### 4.1.2 Besturingselementbron

De tekstvakken van het formulier **frmStuk** zijn gekoppeld aan kolommen van de recordbron. De bijbehorende eigenschap heet *besturingselementbron*. Het tekstvak **txtStuknr** heeft de volgende instelling:

Besturingselementbron: stuknr (de kolom stuknr van de recordbron)

De besturingselementbron (in het Engels *control source*) is een van de belangrijkste eigenschappen van een textbox in Access. De combinatie van recordbron van het formulier en besturingselementbron van de tekstvakken zorgt ervoor dat gebruikers gegevens uit de database kunnen zien. Wijzigingen die gebruikers maken in de tekstvakken worden door Access automatisch opgeslagen in de database (mits de nieuwe waarden aan alle constraints voldoen). Access zorgt dus geheel zelfstandig voor het opbouwen en uitvoeren van een SQL UPDATE opdracht. Ook het toevoegen en verwijderen van records zijn vanzelf mogelijk met een formulier. Access bouwt ook hiervoor zelfstandig de benodigde SQL INSERT en DELETE opdrachten op en voert ze uit. Het toevoegen van records is trouwens alleen mogelijk als alle niet-optionele kolommen van een tabel in de recordbron zitten.

Je kunt heel makkelijk textboxes met een besturingselementbron aanmaken door kolommen te slepen uit de **Lijst met velden**. Hierin vind je alle kolommen van de recordbron van het formulier.

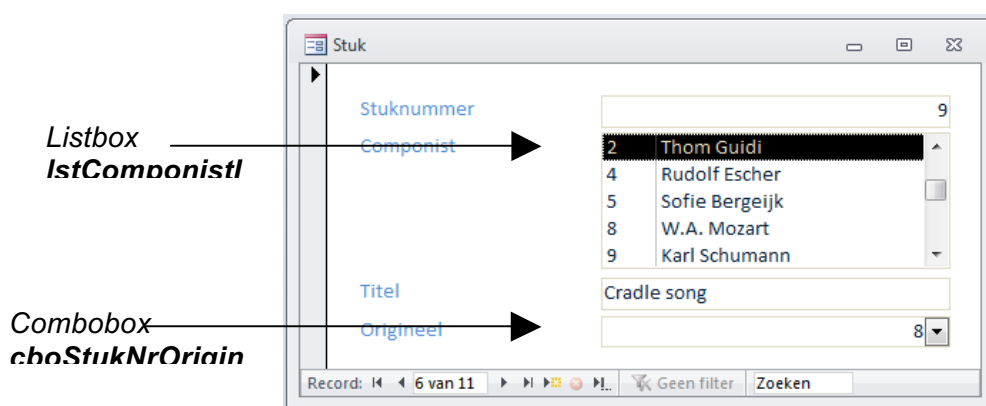


Een tekstvak waarbij de eigenschap besturingselementbron leeg is gelaten heet een niet-afhankelijk tekstvak (Engels: *unbound*). Hierin wordt standaard niks weergegeven en wat een gebruiker er in typt wordt niet opgeslagen in de database. Meer hierover lees je in paragraaf 2.4

## 4.2 List controls: Combobox en Listbox

Een *combobox* (Nederlands: *keuzelijst met invoervak*) en een *listbox* (Nederlands: *keuzelijst*) gebruik je als er een beperkte reeks van waarden mogelijk is in een control. De gebruiker heeft met een list control de mogelijkheid één van de waarden uit de reeks te kiezen. Een list control heeft vier belangrijke eigenschappen, waarvan we de besturingselementbron al kennen:

- Besturingselementbron: De kolom van de recordbron waar de in de lijst **geselecteerde waarde** vandaan komt respectievelijk in wordt opgeslagen.
- Type Rijbron: Afhankelijk van deze instelling kan bij Rijbron gekozen worden voor een lijst van vaste waarden of kunnen de waarden opgehaald worden uit de database.
- Rijbron: De reeks van waarden die getoond wordt en waar de gebruiker uit kan kiezen.
- Afhankelijke kolom: Het volgnummer van de kolom van de rijbron waarin de waarden staan die worden opgeslagen.



*frmStuk met list- en combobox*

In bovenstaand voorbeeld zijn de eigenschappen van de getoonde listbox **lstComponistID** als volgt ingesteld:

- Besturingselementbron: componistId
- Type Rijbron: Tabel/weergave/opgeslagen procedure
- Rijbron: SELECT componistId, naam FROM Componist ORDER BY componistId
- Afhankelijke kolom: 1

#### 4.2.1 Rijbron

De eigenschap *rijbron* (engels: *row source*) kan op verschillende manieren ingevuld worden.

- Vaste lijst van waarden: Een puntkomma-gescheiden vaste lijst met waarden.
- Tabel: De waarden in de lijst worden opgehaald uit een tabel.
- Directe SQL: Je kunt als rijbron direct een SQL SELECT opdracht intikken of opbouwen met de query-builder, bijvoorbeeld SELECT \* FROM Stuk ORDER BY stuknr.
- Access Query: Alleen in accdb's en mdb's. Kies één

MS SQL View:	van de queries die je gemaakt hebt in het query-gedeelte van Access.
MS SQL Stored Procedure:	Alleen in adp's. Kies één van de views (weergaven) die je gemaakt hebt op de SQL Server.
MS SQL Stored Procedure:	Alleen in adp's. Kies één van de stored procedures die je gemaakt hebt op de SQL Server. Deze stored procedure moet een recordset retourneren.

De eigenschappen *afhankelijke kolom* (Engels: *bound column*) en besturingselementbron vormen samen de koppeling met de database.

Naast de bovengenoemde eigenschappen heeft een list control nog enkele interessante eigenschappen:

Aantal kolommen:	De kolommen van de rijbron die getoond worden in de combobox. Als je hier 1 invult wordt alleen de eerste kolom getoond, als je 2 invult worden kolom 1 en 2 getoond etc.
Kolombreedten:	De breedte van de kolommen in de combobox (in centimeters of inches), puntkomma gescheiden. Als je kolom 1 van de rijbron niet wil laten zien, zet dan de eerste waarde op 0 cm.
Kolomkoppen	Als je hier Ja invult worden de namen van de kolommen getoond.

Voor de listbox in bovenstaand voorbeeld zijn deze eigenschappen als volgt ingesteld:

Aantal kolommen:	2
Kolombreedten:	1cm;4cm
Kolomkoppen	Nee

#### 4.2.2 Weergeven van namen en omschrijvingen

Je ziet in onderstaand voorbeeld bij Componist dat een combobox als-ie niet is uitgeklaapt alleen de afhankelijke kolom weergeeft. Soms zal hier iets zinvol staan, zoals bij Genre, maar heel vaak zal dit slechts een weinig zeggend nummer of een code zijn. Er zijn twee oplossingen om toch iets zinvollers aan de gebruiker te tonen. Beide oplossingen zie je in onderstaand voorbeeld:

The screenshot shows a Microsoft Access form titled 'Stuk'. It contains several text boxes and two dropdown menus. The fields are: 'Stuknummer' (text box), 'Titel' (text box), 'Originele stuk' (text box), 'Speelduur' (text box), 'Jaartal' (text box), 'Genrenaam' (dropdown menu), 'Niveaucode' (dropdown menu), and 'Componist' (dropdown menu). The 'Componist' dropdown menu is open, showing 'Thom Guidi'. The status bar at the bottom shows 'Record: 6 van 11' and 'Geen filter'.

*frmStuk met weergave naam en omschrijving*

Bij Niveau is de Rijbron van de combobox **cboNiveau** aangepast zodat de getoonde kolom zowel de code als de omschrijving weergeeft:

```
SELECT niveaucode,
       niveaucode + ' - ' + omschrijving  2
FROM   Niveau
```

De belangrijkste overige eigenschappen zijn als volgt ingesteld:

Afhankelijke kolom:	1
Aantal kolommen:	2
Kolombreedten:	0cm;3cm
Alleen lijst:	Ja

Bij Componist is voor een andere oplossing gekozen; hier wordt de naam van de componist in een aparte textbox getoond. Dit is het eenvoudigst te maken door in plaats van de tabel Stuk als recordbron van het formulier een opgeslagen query (in een mdb), een view (in een adp) te gebruiken. Je kunt de benodigde SQL natuurlijk ook direct als recordbron intikken. Het gaat om de volgende OUTER JOIN:

```
SELECT Stuk.*, Componist.naam
FROM   Stuk LEFT OUTER JOIN Componist
       ON   Stuk.componistId = Componist.componistId
```

Of met haakjes <sup>3</sup>:

```
SELECT Stuk.*, Componist.naam
FROM   (Stuk LEFT OUTER JOIN Componist
       ON   Stuk.componistId = Componist.componistId)
```

Bij complexe queries (in een adp) kan het noodzakelijk zijn om de eigenschap *Unieke tabel* van het formulier in te stellen zodat de records van de basistabel van het formulier bewerkbaar blijven.

Nu kun je een textbox maken met als besturingselementbron de kolom Naam van de tabel Componist. Vanwege de joinconditie zal altijd automatisch de correcte naam getoond worden. Omdat dit scherm bedoeld is voor het beheren van stukken en niet voor het beheren van componisten, is de textbox met de naam uitgeschakeld, zodat de naam hier niet gewijzigd kan worden. De interessante eigenschappen van deze textbox:

Besturingselementbron:	Componist.naam
Ingeschakeld:	Nee

### 4.3 Subformulieren

Bij mutaties van gegevens in tabellen van een database (invoeren nieuwe gegevens, wijzigen en verwijderen van gegevens), moet rekening worden gehouden met de verwijzingen tussen de tabellen. Het is daarom aan te raden bij het maken van onderhoudsformulieren de samenhang tussen de tabellen te implementeren m.b.v. hoofd- en subformulieren.

<sup>2</sup> De Jet Engine gebruikt & als concatenatieteken, vanaf Access 2013 werkt ook +, MS SQL Server gebruikt +.

<sup>3</sup> Bij andere projecttype dan ADP wordt de query door de Jet Engine uitgevoerd. Deze verwacht rond ieder join extra haakjes.

Sub-formulier

Record: 1 van 8 | Geen filter | Zoeken

*frmStukEnBezetting*

Als voorbeeld van het gebruik van een hoofd- en een subformulier bekijken we een formulier voor het onderhouden van stuk- en bezettingsgegevens. Dit formulier genaamd **frmStukEnBezetting** is hierboven afgebeeld. Merk op dat het subformulier op *Doorlopende Weergave* is ingesteld. Hieronder zie je het formulier in de ontwerpweergave:

subform-control

Ontwerpweergave  
*frmStukEnBezetting*

Dit formulier bevat een subform-besturingselement met de naam **subBezetting**. Dit subform-besturingselement kun je zien als een venster waardoor het subformulier **frmsubBezettingBijStuk** zichtbaar wordt gemaakt.

Een subformulier is een normaal formulier, het enige verschil is dat het nu in een subform-besturingselement is geplaatst. Het heeft dan ook dezelfde eigenschappen als ieder ander formulier. De recordbron van het hier getoonde subformulier is de tabel **Bezettingsregel**.

Het subformulier loopt automatisch mee met het hoofdformulier, dat wil zeggen dat alleen de bezettingsregels getoond worden van het stuk dat nu actief is in het hoofdformulier. Dit is te bewerkstelligen door enkele eigenschappen van het subform-besturingselement in te stellen:

Bronobject:	frmsubBezettingBijStuk
Subvelden	stuknr
koppelen:	
Hoofd velden	stuknr
koppelen:	

Deze drie eigenschappen verdienen enige toelichting. De eigenschap *Bronobject* geeft aan welk formulier zichtbaar wordt gemaakt in het subformulier-besturingselement. De eigenschappen *Subvelden koppelen* en *Hoofd velden koppelen* beschrijven de koppeling tussen hoofd- en subformulier. In de bovenstaande situatie: in het subformulier worden alleen die records getoond waarvan het stuknummer correspondeert met het stuknummer van het actieve record op het hoofdformulier.

Vaak gaat het hier om een verwijzing van vreemde sleutel naar primaire sleutel. In het hier besproken voorbeeld is dat de verwijzing:

Bezettingsregel (stuknr) → Stuk (stuknr)

Merk op dat het stuknummer *niet* getoond wordt op het subformulier. Als de koppeling goed werkt staat hier immers altijd hetzelfde stuknummer als op het hoofdformulier, en dat zou dubbelop zijn.

Ook van het subformulier **frmsubBezettingBijStuk** geven we de belangrijkste eigenschappen:

Recordbron:	Bezettingsregel
Standaardweergave	Doorlopend formulier
Navigatieknoppen:	Nee
Scrollbalken:	Alleen verticaal

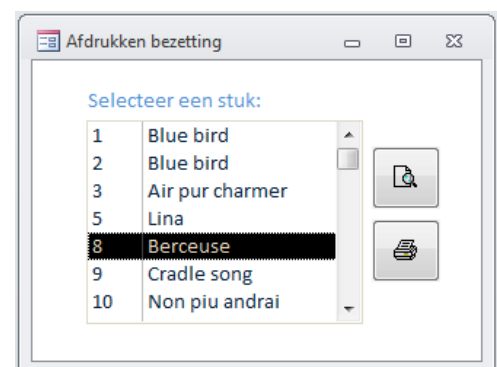
#### 4.4 Unbound formulieren en controls

Een formulier waarbij de eigenschap *recordbron* is ingevuld noem je een *bound form*, of in het Nederlands een *afhankelijk formulier*. Een formulier zonder *recordbron* heet *unbound*, of *niet-afhankelijk*.

Een control waarbij de eigenschap *besturingselementbron* is ingevuld noem je een *bound control*, of in het Nederlands een *afhankelijk besturingselement*. Een control zonder *besturingselementbron* heet *unbound*, of *niet-afhankelijk*.

Bij het openen van een bound formulier worden direct gegevens opgehaald uit de database, volgens de instelling van de *recordbron* en eventuele filters. De gegevens die worden opgehaald worden automatisch in de bound controls geplaatst en je kunt met de navigatieknoppen door de records bladeren. Als je een wijziging maakt in een van de bound controls, wordt de wijziging automatisch in de database opgeslagen zodra je naar een ander record navigeert. Ook het maken van een nieuw record kan, zonder dat je daar iets voor hoeft te programmeren. Voor bound formulieren genereert Access dus zelf SQL SELECT-, UPDATE-, INSERT- en DELETE-opdrachten.

Als je een formulier maakt dat niet gebruikt wordt om gegevens in de database te wijzigen of om nieuwe gegevens te maken en er hoeft ook niet met navigatieknoppen door records te worden gebladerd, dan is het in de meeste gevallen het handigst een formulier zonder *recordbron* te maken. Een formulier zonder *recordbron* kan nog steeds gegevens uit de database *tonen*, bijvoorbeeld in een listbox of een combobox. Maar aanpassen of toevoegen van gegevens kan niet zonder meer met zo'n unbound formulier. Omdat er niet te



navigeren valt in een unbound formulier, is het fraaiër om de navigatieknoppen uit te zetten.

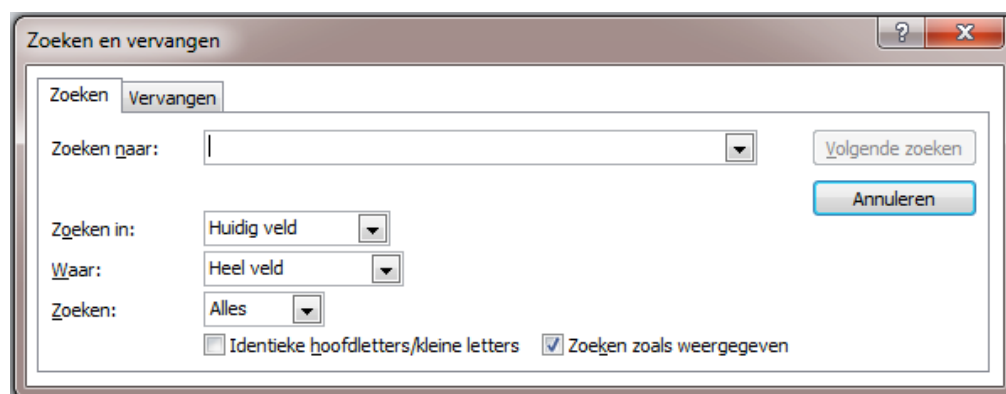
Controls op een unbound formulier zijn altijd unbound, er is immers geen recordbron dus ook geen velden om als besturingselementbron te gebruiken. Let op dat listboxen en comboboxen nog wel een *rijbron* kunnen hebben, ze kunnen nog wel lijstjes tonen met gegevens uit de database. Ook op een bound formulier kunnen unbound controls staan. Sleep bijvoorbeeld een listbox uit de toolbar op het formulier en laat de besturingselementbron leeg. Unbound controls komen van pas als je een gebruiker een waarde wil laten selecteren uit een lijstje, maar deze keuze niet wil opslaan in de database. Je kunt dit bijvoorbeeld gebruiken om een gebruiker eenvoudig een bepaald record te laten vinden. Of om een selectie te maken van records waar een afdruk voor moet worden gemaakt. De volgende paragraaf gaat over het zoeken naar records. Er wordt veelvuldig gebruik gemaakt van unbound controls. Hoofdstuk 5 gaat over het maken van afdrukken.

## 4.5 Zoekmogelijkheden

Tot nu toe hebben we bound formulieren gemaakt waar alle records van de recordbron in worden getoond. De gebruiker kan met de navigatieknoppen of een scrollbar in de records bladeren. Als er een wijziging in een record moet worden gemaakt, bladert de gebruiker naar het juiste record en voert de wijziging in. Dit werkt prima als er in de recordbron een beperkt aantal records aanwezig is. Je kunt je echter voorstellen dat het bladeren door heel veel records niet gebruiksvriendelijk is. In deze paragraaf enkele voorbeelden van handige manieren om het juiste record, of de juiste records, te vinden.

### 4.5.1 Standaard Access zoekfunctie

In het lint Start is de functie *Zoeken* (of *Find*) te vinden, die een standaard zoekscherm opent. Met deze zoekfunctie is het mogelijk om in records te zoeken, in de kolom van het actieve veld (het veld waar de cursor in staat) of in alle kolommen. Deze standaard zoekfunctie kun je ook met behulp van een knop op het formulier laten aanroepen. De wizard die verschijnt na het op het formulier slepen van een knop kan de daarvoor benodigde vba-code opbouwen.



### 4.5.2 Navigeren m.b.v. een listbox of combobox

Naast de hierboven getoonde standaard zoekfunctie kun je ook je eigen, fraaiere, zoek- en navigatiemogelijkheden bouwen. Een voorbeeld zie je in onderstaand formulier. In dit voorbeeld zie je een listbox, maar in plaats daarvan zou je ook een combobox kunnen gebruiken, die werkt op precies dezelfde manier.



*frmStukZoeken*

Aan de linkerkant van dit bound formulier zie je een unbound keuzelijst genaamd **IstStukken**. Als je in deze keuzelijst een stuk aanklikt, spring je direct naar het corresponderende record.

De belangrijkste eigenschappen van **IstStukken**:

Besturingselementbron:	-
Type rijbron:	Tabel/Query
Rijbron:	SELECT Stuknr, Titel FROM STUK ORDER BY Stuknr
Afhankelijke kolom:	1
Aantal kolommen:	2
Kolombreedten:	1cm;3cm
Alleen lijst:	Ja

Om deze functionaliteit correct te laten werken hebben we voor het eerst nu VBA<sup>4</sup> code nodig.

De 'Na bijwerken'-gebeurtenisprocedure van de listbox IstStukken ziet er dan als volgt uit:

```
Private Sub IstStukken_AfterUpdate()

    Dim rst As DAO.Recordset
    Set rst = Me.RecordsetClone

    rst.FindFirst "[STUKNR] = " & Me.IstStukken.Value
    Me.Bookmark = rst.Bookmark

    Set rst = Nothing

End Sub
```

Verderop in deze reader wordt meer uitgelegd over het gebruik van VBA code.

<sup>4</sup> VBA: Visual Basic for Applications, de programmeertaal beschikbaar in nagenoeg alle MS Office onderdelen..

### 4.5.3 Selectie van records in een subformulier

In paragraaf 2.3 zie je op het formulier **frmStukEnBezetting** de bezettingsregels per stuk in beeld gebracht. Daarbij is gebruik gemaakt van een hoofd- en een subformulier. Als wel de bezettingsregels moeten kunnen worden aangepast, maar niet de gegevens van de stukken, dan zou je het volgende voorbeeld kunnen gebruiken.

	instrumentnaam	toonhoogte	aantal
	drums		1
	piano		1
	saxofoon	alt	2
	saxofoon	tenor	1
*			

*frmBezettingbijStuk*

Het hoofdformulier is nu unbound. Met behulp van de combobox **cboStuknr** kun je een stuk selecteren. De titel wordt in de unbound textbox getoond en in het subformulier worden de bezettingsregels van het geselecteerde stuk weergegeven.

Bij het maken van dit formulier kun je het best als volgt te werk gaan:

- Ontwikkel eerst het subformulier **frmsubBezettingBijStuk** apart (als dat nog niet bestaat); de recordbron van het subformulier is de tabel **Bezettingsregel**;
- Ontwikkel vervolgens het hoofdformulier (zonder recordbron) en plaats daarop een combobox **cboStuknr** en een niet-afhankelijk tekstvak **txtTitel**.
- De eigenschappen van de combobox **cboStuknr** zijn als volgt:

Besturingselementbron: -  
 Type rijbron: Tabel/Query  
 Rijbron: SELECT Stuknr, Titel FROM STUK  
 ORDER BY Stuknr  
 Afhankelijke kolom: 1  
 Aantal kolommen: 2  
 Kolombreedten: 2cm;5cm  
 Lijstbreedte: 7cm  
 Alleen lijst Ja

- Plaats op het hoofdformulier een subform-besturingselement **subBezettingVanStuk** en kies als bronobject het formulier **frmsubBezettingBijStuk**.
- Breng de koppeling tussen hoofd- en subformulier tot stand door bij het subform-besturingselement de volgende eigenschappen in te vullen:

Subvelden

Stuknr

koppelen:  
Hoofd velden                      cboStuknr  
koppelen:

Laat de titel van het stuk zien in het tekstvak **txtTitel** door de 'Na Bijwerken'-gebeurtenisprocedure van de combobox **cboStuknr** als volgt in te vullen:

```
Private Sub cboStuknr_AfterUpdate()  
  
    Me.txtTitel = Me.cboStuknr.Column(1)  
  
End Sub
```

De uitdrukking Column(1) verwijst naar de tweede kolom (en dat is de titel) in de combobox; Column(0) verwijst naar de eerste kolom (dat is in dit geval het stuknummer).

#### 4.5.4 Records filteren

De laatste zoekmethode die we beschrijving werkt door middel van het instellen van de eigenschap *filter* van een formulier. In die eigenschap kun je een conditie opgeven die als een soort SQL WHERE-conditie op de recordbron wordt toegepast. Natuurlijk stel je deze eigenschap niet als applicatieprogrammeur éénmalig in maar maak je het mogelijk dat de gebruiker er zelf voor kan zorgen dat deze eigenschap wordt ingesteld. Dat zou er als volgt uit kunnen zien (let op de navigatiebalk):

*frmStukZoeken*

De textbox waar het nummer ingetikt kan worden om naar te zoeken heet **txtZoekStuk** en is een unboud textbox. Achter de knoppen **cmdZoek** en **cmdReset** zit de volgende code:

```
Private Sub cmdZoek_Click()  
  
    If Not IsNull(Me.txtZoekStuk) Then  
  
        Me.Filter = "Stuknr = " & Me.txtZoekStuk.Value  
        Me.FilterOn = True  
  
        Me.txtZoekStuk.SetFocus  
  
        Me.cmdZoek.Enabled = False  
        Me.cmdReset.Enabled = True  
  
    End If  
  
End Sub  
  
Private Sub cmdReset_Click()  
  
    Me.FilterOn = False  
    Me.txtZoekStuk.Value = Null  
  
    Me.txtZoekStuk.SetFocus  
  
    Me.cmdReset.Enabled = False  
    Me.cmdZoek.Enabled = True  
  
End Sub
```

Het gaat in het eerste stuk code om de eigenschap *Filter* van het formulier (*Me*) die wordt ingesteld en vervolgens wordt geactiveerd (*FilterOn = True*). In het tweede stuk zie je dat het filter weer wordt gedeactiveerd. De rest van de code is voor de nette afwerking: het in- en uitschakelen van de twee knoppen, het verplaatsen van de cursor (*SetFocus*) en het leegmaken van het zoekveld.

## 5 Formulierontwerp

In het vorige hoofdstuk is de basiskennis beschreven die je nodig hebt om formulieren te maken in Access. In dit hoofdstuk geven we enkele tips die je in overweging kunt nemen bij het bepalen van welke formulieren je gaat maken voor je applicatie, en hoe deze formulieren eruit moeten zien.

### 5.1 Processen

Een formulier wordt altijd gemaakt om een (deel van) een *bedrijfsproces* te ondersteunen. Naast een database(-model) zul je voordat je de applicatie gaat maken een model hebben gemaakt van de bedrijfsprocessen. Hiervoor bestaan veel verschillende technieken. Bij de meeste technieken geef je naast het verloop van het proces een meer of minder uitgebreide beschrijving van de gegevens die in het proces van belang zijn. Je beschrijft bijvoorbeeld welke zoekcriteria er gebruikt kunnen worden en welke gegevens op het scherm worden getoond.

Hieronder staat een voorbeeld van een procesbeschrijving van een van de processen die bij de muziekdatabase horen, namelijk het proces *Beheren Stukken*.

De gegevensbeheerder wil een nieuw stuk invoeren, de gegevens van een bestaand stuk wijzigen of een stuk verwijderen.

Als een bestaand stuk moet worden gewijzigd of verwijderd dan wordt het betreffende stuk gevonden op basis van stuknummer of titel.

Als de componist van het stuk een componist is die nog niet bekend is dan moeten de gegevens van deze nieuwe componist meteen worden ingevoerd.

Je kunt van de beschrijving gebruik maken bij het ontwerpen van je formulier. Probeer in eerste instantie de recordbron op te stellen aan de hand van de beschrijving. Het spreekt vanzelf dat de basis van deze recordbron gevormd wordt door de tabel **Stuk**. Er zijn in de beschrijving geen aanwijzingen dat sommige kolommen van deze tabel niet ingevuld moeten worden. We gaan er daarom vanuit dat alle kolommen zijn in te vullen of aan te passen. In eerste instantie is de recordbron daarom:

```
SELECT * FROM Stuk
```

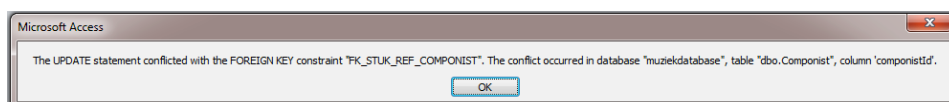
Op basis van deze recordbron kun je met behulp van een wizard of met de hand een begin van formulier maken. Je zult een keuze moeten maken tussen een formulier in doorlopende weergave, gegevensbladweergave of enkelvoudige weergave. De keuze hiertussen hangt voornamelijk af van het aantal kolommen in de recordbron. Als dit er veel zijn kun je beter kiezen voor doorlopende weergave of gegevensbladweergave. Als het om slechts enkele kolommen gaat kun je beter voor enkelvoudige weergave kiezen. Ook als je controls wil gebruiken die nogal veel ruimte in beslag nemen, zoals een listbox, een radiobutton group of een calendar, is enkelvoudige weergave handiger. Voor het genoemde voorbeeld ziet het formulier zo uit:

Merk op dat de labels met de hand aangepast zijn; de wizard baseerde ze op de kolomnamen maar die zijn niet erg fraai.

## 5.2 Foreign Key's

Bij het invullen van een ComponistId mag niet zomaar ieder nummer worden ingevuld. Het moet wel het nummer zijn van een componist die in de tabel **Componist** staat. De kolom ComponistId is in het database-schema dan ook te herkennen als een foreign key. Deze foreign key wordt netjes bewaakt door de database-server. Als men toch een ongeldig nummer invult verschijnt, afhankelijk van het RDBMS, ongeveer de volgende melding:

*UPDATE statement conflicted with the FOREIGN KEY constraint 'FK\_STUK\_REF\_COMPONIST'. The conflict occurred in database 'muziekdatabase', table 'dbo.Componist', column 'componistId'. The statement has been terminated.*



Echt mis kan het dus niet gaan, maar het is niet fraai om de gebruiker deze melding voor te schotelen. Een oplossing hiervoor is het gebruik van een listbox of combobox waarmee de gebruiker alleen maar kan kiezen uit geldige nummers. De textbox ComponistId veranderen we in een combobox. In deze combobox tonen we de nummers en namen van alle componisten:

Besturingselementbron:	ComponistId
Type Rijbron:	Tabel
Rijbron:	SELECT ComponistId, Naam FROM Componist
Afhankelijke kolom:	1
Aantal kolommen:	2
Alleen lijst:	Ja
AutoUitbreiden:	Ja

Hetzelfde geldt voor het stuknummer van het origineel, de niveaucode en het genre. Dit zijn allemaal foreign keys. Het formulier gaat er nu zo uit zien:

*frmBeheerStukken*

Nadere bestudering van de foreign key's wijst uit dat er in drie van de vier een nummertje of een code staat: componistId, stuknrOrigineel en niveaucode. In de tabellen waar de foreign key's naar verwijzen zijn namen en omschrijvingen bij de nummers en de codes op te zoeken. Laten we deze namen en omschrijvingen opnemen in de recordbron van het formulier om de codes en nummers voor de gebruiker te verduidelijken:

```
SELECT Stuk.*, Componist.naam, Stuk_1.titel AS
TitelOrigineel, Niveau.omschrijving
FROM (((Stuk INNER JOIN Niveau
ON Stuk.niveaucode = Niveau.niveaucode)
LEFT OUTER JOIN Stuk Stuk_1
ON Stuk.stuknrOrigineel = Stuk_1.stuknr)
LEFT OUTER JOIN Componist ON Stuk.componistId =
Componist.componistId)
```

Nu we ook de naam van de componist, de titel van het origineel, en de omschrijving van het niveau in de lijst met velden er bij hebben, kunnen we het formulier als volgt uitbreiden:



De textboxen van de naam van de componist, de titel van het origineel, en de omschrijving van het niveau zijn op 'locked' gezet. Dit proces gaat immers over het beheer van stukken en niet over het beheer van componisten en niveau's, en terwijl stuknummer 2 beheerd wordt, moet niet de titel van het origineel daarvan aangepast worden.

### 5.2.1 Toevoegen foreign key record

In de procesbeschrijving wordt ook gezegd dat er een nieuwe componist toegevoegd moet kunnen worden als dat nodig is. Dat zou je kunnen doen door een knop achter het veld van de componist te zetten waarmee de gebruiker een nieuw formulier kan openen. Met dit nieuwe formulier is de nieuwe componist toe te voegen.

Een klik op de knop opent het volgende formulier. Dit is het formulier dat zou horen bij het proces *Beheren Componisten*. Het staat nu echter al klaar om een nieuwe componist in te voeren.

De code achter de 'Nieuw...' knop is als volgt:

```
Private Sub cmdNieuweComponist_Click()

    DoCmd.OpenForm "frmComponist", , , , acFormAdd

End Sub
```

De nieuwe componist moet wel meteen in de combobox te selecteren zijn. Om dit mogelijk te maken moet de inhoud van de combobox opnieuw opgehaald worden als het eerste formulier weer actief wordt (**Bij Activeren**):

```
Private Sub Form_Activate()

    Me.cboComponistId.Requery

End Sub
```

### 5.3 Foreign key verwijzingen

In de beschrijving van het proces staat niet duidelijk om welke gegevens van stukken het beheer precies gaat. Als we naar de database kijken dan zien we dat er twee foreign key *verwijzingen* (weet onderscheid te maken tussen foreign key's en foreign key verwijzingen) naar de tabel **Stuk**. Er is een verwijzing van de tabel **Bezettingsregel** en een verwijzing vanuit de tabel **Stuk** zelf. Dit betekent dat er meerdere bezettingsregels kunnen zijn bij één stuk, en ook kan een stuk dienen voor origineel van meerdere bewerkingen.

Omdat de procesbeschrijving niet erg duidelijk is zitten we met de vraag of bezettingsregels deel uit maken van de gegevens van een stuk. Gebaseerd op de database is het antwoord op de vraag 'Ja'. In de tabel **Bezettingsregel** maakt de kolom **Stuknr** namelijk deel uit van de primary key. Om deze reden wordt de relatie tussen **Stuk** en **Bezettingsregel** ook wel een afhankelijke relatie genoemd, of een parent-child relatie. Een bezettingsregel kan alleen bestaan bij de gratie van een stuk. Het is niet zinvol een apart formulier te maken waarmee alle bezettingsregels kunnen worden beheerd, onafhankelijk van een stuk. Als het goed is zal er in de complete processenspecificatie ook geen proces hiervoor te vinden zijn. We nemen het beheer van bezettingsregels daarom op in ons formulier voor het beheer van stukken in de vorm van een subformulier, gekoppeld op basis van beide stuknummers:

**Stuk en bezetting**

Stuknummer: 2 Genrenaam: jazz

Titel: Blue bird Niveaucode: B

Originele stuk: 1

Blue bird Componist: 2

Speelduur: 4 Thom Guidi

Jaartal: 1988

**Bezetting**

instrumentnaam	toonhoogte	aantal
drums		1
piano		1
saxofoon	alt	2
saxofoon	tenor	1
*		

Record: 1 van 8 | Zoeken

Omdat instrumentnaam een foreign key is kun je analoog aan de voorbeelden in de vorige paragraaf van dat veld een combobox maken.

Voor de andere foreign key verwijzing naar de tabel **Stuk**, die vanuit de tabel zelf, is de vraag of bewerkingen van een stuk deel uit maken van de gegevens van een stuk niet zo duidelijk te beantwoorden. De kolom **StuknrOrigineel** maakt in ieder geval geen deel uit van de primary key, en is ook niet verplicht. Het is niet vanzelfsprekend dat ieder stuk een origineel heeft. Ook andersom is er geen verplichting; een stuk hoeft niet perse een origineel te zijn van een aantal bewerkingen. Als het goed is, is er in de processenspecificatie dan ook een zelfstandig proces te vinden dat gaat over het beheren van records van de verwijzende tabel. In dit geval is zo'n beschrijving er inderdaad want het betreft *toevallig(!)* ook de tabel **Stuk** en daar gaat onze procesbeschrijving over. Het toevoegen, verwijderen en wijzigen van bewerkingen doen we dus niet als subformulier. Omdat het wel interessant is om bij een stuk zijn bewerkingen te zien implementeren we deze foreign key verwijzing als een listbox. Als er meerdere foreign key verwijzingen naar de hoofdtabel zijn kun je met tabbladen werken om alle subformulieren en listboxen overzichtelijk te houden:

*frmStukEnBezetting*

Om in de listbox niet alle stukken te tonen maar alleen stukken die een bewerking zijn van de stukken in het hoofdformulier moet de rijbron van de listbox dynamisch zijn.

Om in een mbd-applicatie een dynamische rijbron te maken volstaat het in de rijbron opnemen van een conditie met een verwijzing naar het veld **Stuknr** van het hoofdformulier:

```
SELECT *
FROM qryStuk
WHERE StuknrOrigineel = Me!Stuknr
ORDER BY Stuknr
```

In een mdb-applicatie wordt Me!Stuknr door de Jet Engine vervangen door het stuknummer dat actief is in het formulier. In een adp-applicatie wordt de SQL opdracht echter rechtstreeks naar Microsoft SQL Server gestuurd en die heeft geen idee wat-ie met Me!Stuknr aan moet. Om toch hetzelfde effect te verkrijgen zul je bij het navigeren naar een ander record in het hoofdformulier met behulp van vba-code de rijbron van de listbox moeten aanpassen. Dat kan in het event 'Bij Aanwijzen' ('On Current' in het Engels) met de volgende code:

```
Private Sub Form_Current()
```

```
    IstBewerkingen.RowSource = "" & _
        "SELECT StukNr, Titel, Componist.Naam," & _
        " Genrenaam AS Genre, Niveaucode AS Niveau," & _
        " Speelduur, Jaartal " & _
        "FROM (Stuk INNER JOIN Componist " & _
        " ON Stuk.ComponistId = Componist.ComponistID)" & _
        " WHERE StuknrOrigineel = " & Me.stuknr & _
        " ORDER BY Stuknr"
```

```
IstBewerkingen.Requery
```

```
End Sub
```

Om het helemaal af te maken zou je er voor kunnen zorgen dat de gebruiker kan dubbelklikken op een rij in de listbox en daarmee het formulier opent waarmee dat record bewerkt kan worden. Je opent dan het formulier dat hoort bij het proces voor het beheren van gegevens van de verwijzende tabel. In dit geval is dat dus grappig genoeg weer hetzelfde formulier!

```
Private Sub IstBewerkingen_DblClick()
```

```
DoCmd.OpenForm "frmStuk", , , _  
"Stuknr = " & IstBewerkingen.Value
```

```
End Sub
```

## 5.4 Afronding

In de procesbeschrijving werd vermeld dat er gezocht moet kunnen worden op stuknummer en op titel. We kiezen een van de zoekmogelijkheden die in hoofdstuk 2 werden beschreven. Tot slot voegen we een logo toe in de formulier-koptekst.

Beheer Stukken

**Vereniging Arnhemse Muziekscholen**

Zoek op stuknummer  Zoek

Zoek op titel  Reset

Stuknummer  Genrenaam

Titel  Niveaucode

Originele stuk

Componist  Nieuw...

Speelduur  Charlie Parker

Jaartal

Bezetting **Bewerkingen**

instrumentnaam	toonhoogte	aantal
<input type="text"/>	<input type="text"/>	<input type="text"/>

Record: 1 van 3 Gefilterd Zoeken

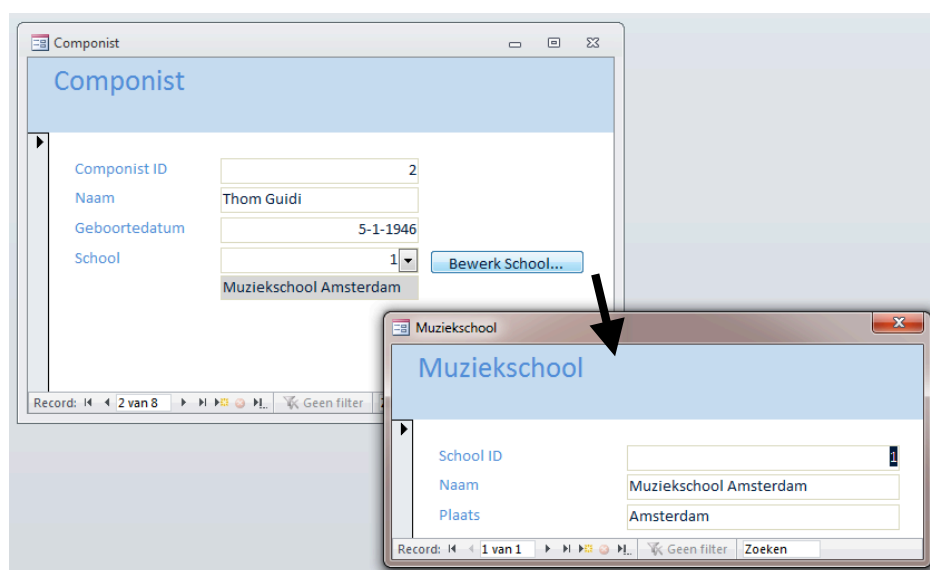
*frmStukEnBezetting*

## 6 Geavanceerde formulieren

In dit hoofdstuk bekijken wat geavanceerde formulieren. We geven enkele voorbeelden van manieren om je applicatie wat gebruiksvriendelijker te maken. In de meeste voorbeelden wordt gebruik gemaakt van Visual Basic for Applications. Het is uitdrukkelijk niet onze bedoelingen een opsomming te maken van de mogelijkheden van VBA. De VBA-code moet je zien als hulpmiddel bij enkele interface-truukjes. Het leren van VBA is voor nu geen doel op zich.

### 6.1 Openen van andere formulieren en rapporten

Het openen van een rapport of een formulier vanaf een ander formulier is in het voorgaande hoofdstuk al even aan de orde geweest, maar zonder uitleg. Bekijk onderstaand voorbeeld.



*frmStukEnBezetting + frmMuziekschool*

Een klik op de knop 'Bewerk School...' opent een nieuw formulier waarmee de betreffende school, in dit geval school 1, te bewerken is. De recordbron van dit formulier is `SELECT * FROM Muziekschool`. Toch wordt alleen school 1 weergegeven, kijk maar naar de navigatiebalk (1 van 1).

De code die achter de **Bij Klikken**-procedure zit is als volgt:

```
Private Sub cmdNieuweSchool_Click()  
    DoCmd.OpenForm "frmMuziekSchool", , , "SchoolID = " & _  
        Me.schooId, , acDialog  
End Sub
```

De opdracht `DoCmd.OpenForm` is bedoeld om formulieren te openen. De opdracht `DoCmd.OpenReport` kun je gebruiken om rapporten te openen (zie hoofdstuk 5). Je kunt een aantal parameters meegeven in de volgende volgorde:

Formname:	Naam van het te openen formulier of rapport
View:	Interessant voor rapporten: meteen afdrucken of eerst een preview weergeven
Filtername:	Niet interessant

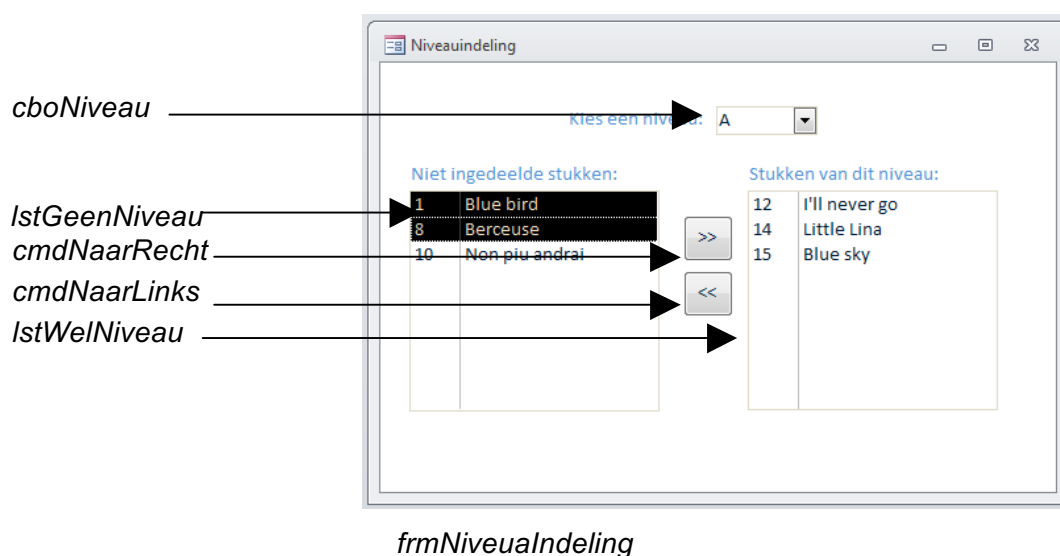
WhereCondition:	Een conditie die als WHERE-clausule op de recordbron van het formulier of rapport wordt toegepast
DataMode:	Interessant voor formulieren: alleen bewerken, alleen nieuw record toevoegen of alles mogelijk.
WindowMode:	Geeft aan in welke modus het formulier getoond moet worden. acDialog is een veelgebruikte instelling. Hierdoor blijft het scherm actief en moet het eerst gesloten worden voordat men wat anders kan doen.
OpenArgs:	Eén string die meegegeven wordt aan het te openen formulier. In dat formulier weer aan te roepen d.m.v. Me.OpenArgs.

Toelichting bij de WhereCondition: hier kun je criteria opgeven net zoals je in SQL zou doen. Je hoeft het woord 'WHERE' niet mee te geven, dat gebeurt vanzelf. In het bovenstaande voorbeeld wordt de recordbron dus `SELECT * FROM Muziekschool WHERE SchoolId = 1`. De '1' wordt er achter geplakt omdat in dit geval '1' de waarde is van het veld dat SchoolId heeft op het eerste formulier (Me).

## 6.2 Embedded SQL

Soms kan het handig zijn om in de VBA-code een SQL-opdracht te laten uitvoeren, bijvoorbeeld om even een waarde uit de database op te halen of om iets op te slaan in de database. In deze paragraaf enkele voorbeelden.

Met het onderstaande voorbeeld kan een gebruiker snel zien welke muziekstukken nog geen niveaucode hebben en welke stukken van een bepaald niveau zijn. Ook kan op een handige manier de niveaucode bij een stuk worden ingevuld of weggehaald.



De keuzelijst **IstGeenNiveau** heeft de volgende rijbron:



```
SELECT Stuknr, Titel
FROM Stuk
WHERE Niveaucode IS NULL
ORDER BY Stuknr
```

De keuzelijst **IstWelNiveau** heeft in een mdb-bestand de volgende rijbron:

```
SELECT Stuknr, Titel
FROM Stuk
WHERE Niveaucode = Me.cboNiveau.Value
ORDER BY Stuknr
```

Bij het wijzigen van de waarde in de combobox **cboNiveau** moet de inhoud van **IstWelNiveau** worden aangepast, zodat er de stukken in staan van het gekozen niveau. De benodigde code hoort bij het Na Bijwerken event van de combobox en is voor een mdb-bestand als volgt:

```
Private Sub cboNiveau_AfterUpdate()

    Me.IstWelNiveau.Requery

End Sub
```

In een accdb-bestand kan dit bijna op dezelfde wijze, echter de rijbron moet dan expliciet naar het formulier wijzen:

```
SELECT Stuk.Stuknr, Stuk.Titel
FROM Stuk
WHERE (Stuk.niveaucode=[Forms]![frmNiveauIndeling_Figuur15]![cboNiveau])
ORDER BY Stuk.Stuknr;
```

In een adp-bestand is de dynamische rijbron van IstWelNiveau niet mogelijk; je kunt in de SQL niet verwijzen naar een formulerveld. In plaats daarvan geef je de listbox initieel géén rijbron en stel je de rijbron vanuit VBA-code in op het moment dat dat nodig is, dus ook in het Na Bijwerken event van de combobox:

```
Private Sub cboNiveau_AfterUpdate()

    Me.IstWelNiveau.RowSource = "" & _
    "SELECT Stuknr, Titel " & _
    "FROM Stuk " & _
    "WHERE Niveaucode = "" & Me.cboNiveau.Value & "" " & _
    "ORDER BY Stuknr"

    Me.IstWelNiveau.Requery

End Sub
```

Dit werkt ook voor mdb en accdb bestanden.

Om meerdere rijen tegelijk te kunnen selecteren in de listboxes, dient de eigenschap *Meervoudige Selectie* op *Uitgebreid* te staan.

In het bovenstaande voorbeeld is de gebruiker bezig om de indeling van niveau A te bewerken. Als je in de linker keuzelijst één of meer stukken selecteert en vervolgens op het pijltje naar rechts klikt,

dan worden de geselecteerde stukken overgeheveld naar de keuzelijst rechts. De wezenlijke acties hiervoor zitten in het Bij Klikken event van de knop **cmdNaarRechts**:

```
Private Sub cmdNaarRechts_Click()

    Dim varlrm As Variant

    For Each varlrm In Me.lstGeenNiveau.ItemsSelected

        CurrentDB.Execute5 "UPDATE Stuk SET Niveaucode=" & _
            Me.cboNiveau.Value & " WHERE Stuknr = " & _
            Me.lstGeenNiveau.ItemData(varlrm)

    Next varlrm

    Me.lstGeenNiveau.Requery
    Me.lstWelNiveau.Requery

End Sub
```

Net zo kunnen stukken uit een niveau worden verwijderd door gebruik te maken van de knop met de pijl naar links. Het Bij Klikken event van deze knop ziet er als volgt uit:

```
Private Sub cmdNaarLinks_Click()

    Dim varlrm As Variant

    For Each varlrm In Me.lstWelNiveau.ItemsSelected

        CurrentDB.Execute "UPDATE Stuk SET Niveaucode " & _
            "= NULL WHERE Stuknr = " & _
            Me.lstWelNiveau.ItemData(varlrm)

    Next varlrm

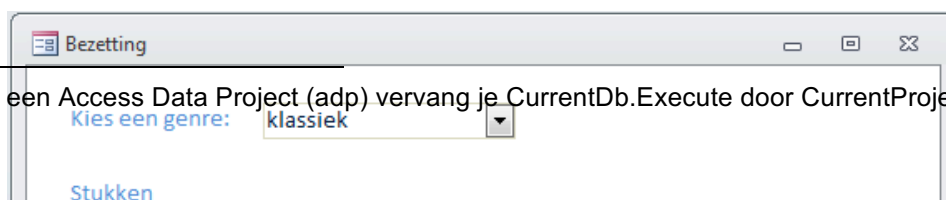
    Me.lstGeenNiveau.Requery
    Me.lstWelNiveau.Requery

End Sub
```

### 6.3 Meerdere niveau's van subformulieren

Soms komt het voor dat je twee subformulieren wil tonen, waarbij de tweede afhankelijk is van de eerste. Je moet dan niet proberen een nieuw subformulier te maken binnen het eerste subformulier, maar twee subformulieren op het hoofdformulier plaatsen.

<sup>5</sup> In een Access Data Project (adp) vervang je CurrentDb.Execute door CurrentProject.Connection.Execute



← *cboGenre*

← *subStukken*

← *subBezettingBij*

### *frmBezetting*

Het bovenste subformulier is 'gewoon' afhankelijk van het gekozen genre in de combobox **cboGenre**. Dat is gerealiseerd met de volgende eigenschappen van het bovenste subformcontrol **subStukken**:

Hoofdelden	cboGenre
koppelen:	
Subelden	Genre
koppelen:	

Het onderste subformcontrol **subBezettingVanStuk** is afhankelijk gemaakt van het bovenste met de volgende eigenschappen:

Hoofdelden	[subStukken].Form!Stuknr
koppelen:	
Subelden	Stuknr
koppelen:	

Daarbij moet je de expressie [subStukken].Form!Stuknr lezen als (van rechts naar links): de kolom (of tekstvak) Stuknr van het formulier (Form) in het subformulier-besturingselement subctrlStuk.

Verder is het nodig om de inhoud van het onderste subformulier-besturingselement bij te werken op het moment dat een ander record in het bovenste subformulier actief wordt. Dat gebeurt in het Bij aanwijzen event van het formulier **subfrmStukken**:

```
Private Sub Form_Current()
```

Me.Parent![subBezettingVanStuk].Requery

End Sub

Dit kun je lezen als (weer van rechts naar links): bepaal opnieuw de inhoud (Requery) van het besturingselement subBezettingVanStuk op het ouderformulier (Parent) van dit formulier (Me).

## 6.4 Bewaken van beperkingsregels

Voor de muziekdatabase geldt de volgende beperkingsregel: *het jaartal van een stuk moet hoger zijn dan het geboortjaar van de componist van het stuk.*

Deze beperkingsregel (of *constraint* of *business rule*) kan niet bewaakt worden met de middelen die ons ter beschikken staan bij het maken van de tabellen. We kunnen dit echter wel in de applicatie bewaken. We bekijken het volgende formulier.

*frmBeheerStukken*

Bij het verlaten van het tekstvak **Jaartal** dienen we een controle uit te voeren. Blijkt het jaartal voor het geboortjaar van de componist te liggen, dan wordt er een messagebox met een foutmelding getoond en wordt het tekstvak niet verlaten.

De controle vindt plaats in het Voor Bijwerken event van het tekstvak Jaartal, dit event gaat af zodra de gebruiker het veld verlaat:

```
Private Sub Jaartal_BeforeUpdate(Cancel As Integer)
```

```
Dim dGeboortedatum As Date
```

```
dGeboortedatum = DLookup("Geboortedatum", _  
    "Componist", "ComponistId = " & Me.componistId)
```

```
If Me.jaartal < Year(dGeboortedatum) Then  
    MsgBox "Componist " & Me.naam & _  
        " werd geboren in " & Year(dGeboortedatum) & _  
        ", dus dit jaartal is onmogelijk", vbExclamation
```

```
Cancel = True ' verlaten tekstvak ongedaan maken
```

End If

End Sub

Let op de opdracht Cancel = True; hiermee wordt het verlaten van het veld ongedaan gemaakt, zodat de gebruiker meteen een correcte waarde moet invoeren.

In bovenstaande code zie je een functie genaamd DLookup voor het ophalen van de geboortedatum. Strict genomen is deze hier niet noodzakelijk omdat de geboortedatum al op het formulier staat en we vanuit de code daar naar zouden kunnen verwijzen.

MS Access bevat een aantal functies om snel data op te zoeken in de database. Eén van de veel gebruikte functies is de DLookup functie.

De syntax van de Dlookup functie is als volgt:

**Function DLookup(Expr As String, Domain As String, [Criteria])**

**Expr:** een expressie, meestal een kolom uit het 'domain' (maar zou ook een functie kunnen zijn)

**Domain:** de naam van een tabel of query/view

**Criteria:** een conditie, vergelijkbaar met de WHERE clause (zonder 'WHERE' construct). Dit mag dus een combinatie van condities zijn (gebruik makend van AND, OR etc)

Tijdens het uitvoeren van deze functie zal MS Access de functie vertalen in een SQL opdracht alvorens deze wordt uitgevoerd.

Dit wordt dan ongeveer als volgt vertaald:

```
SELECT      Expr
FROM        "Domain"
WHERE       ( Criteria )
```

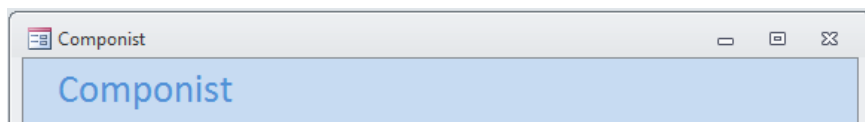
Of als het criteria ontbreekt:

```
SELECT      Expr
FROM        "Domain"
```

In een ADP project wordt vervolgens SQL opdracht naar de database server verstuurd. Voor een ACCDB/MDB project zal de Jet Engine deze uitvoeren.

## 6.5 Keuzerondjes

Als er in een veld slechts een beperkt aantal mogelijke waarden is, zou je keuzerondjes (radiobuttons) kunnen gebruiken, zoals in onderstaand voorbeeld.





### frmComponist

Om een set keuzerondjes te maken die *bound* is moet je eerst een groepsvak (ook wel option group genoemd) maken en niet losse rondjes op het formulier zetten. Daarna plaats je keuzerondjes (ook wel radio buttons of option buttons genoemd) in dit groepsvak. Tijdens het plaatsen wordt de achtergrond van het groepsvak zwart zodat duidelijk is dat het keuzerondje hoort bij het betreffende groepsvak.

Keuzerondjes hebben standaard een numerieke waarde (value), beginnend bij 1. In het voorbeeld hier boven heeft het rondje van 'Vrouw' de waarde 1 en 'Man' de waarde 2.

Dat betekent dat Access standaard de numerieke waarde van het gekozen rondje in de database probeert op te slaan. Dat klopt natuurlijk niet als er iets anders in de database behoort te komen, zoals bijvoorbeeld 'V' of 'M'. Om dat op te lossen moet je het groepsvak *unbound* maken en met behulp van programmeercode zowel het tonen als het opslaan van de juiste waarde regelen.

Om te zorgen dat bij het tonen van een record het juiste rondje ingevuld is gebruik je de volgende code in het event *Bij Aanwijzen (On Current)*.

```
Private Sub Form_Current()  
  
    If Me.GESLACHT = "V" Then  
  
        Me.grpGeslacht.Value = 1  
  
    ElseIf Me.GESLACHT = "M" Then  
  
        Me.grpGeslacht.Value = 2  
  
    End If  
  
End Sub
```

Om te zorgen dat een gekozen waarde correct wordt opgeslagen gebruik je deze code in het event *Na bijwerken (After Update)*.

```
Private Sub grpGeslacht_AfterUpdate()  
  
    If Me.grpGeslacht.Value = 1 Then  
  
        Me.GESLACHT = "V"  
  
    ElseIf Me.grpGeslacht.Value = 2 Then  
  
        Me.GESLACHT = "M"  
  
    End If  
  
End Sub
```

Merk op dat we hier geen rekening hebben gehouden met het feit dat de kolom GESLACHT wellicht NULL waarden kan bevatten.

## 6.6 Afbeeldingen

Het is mogelijk om afbeeldingen op te slaan in een relationele database. In de Jet Engine maak je daarvoor in een tabel een extra kolom aan van het type Ole-Object. In Microsoft SQL Server maak je een kolom aan van het type Image.

Op een formulier dien je een control van het type Kader voor object te plaatsen. Dit type control maakt standaard windows Ole-functies mogelijk zoals knippen, plakken en invoegen van objecten. Access zorgt automatisch voor het opslaan en ophalen van het object in en uit de database. De plaatjes moeten bij voorkeur van het type bitmap (BMP) zijn.

Instrument

Instrument

instrumentnaam drums

toonhoogte

plaatje



Record: 1 van 10 | Zoeken

*frmInstrument*



## 7 Rapporten

Rapporten zijn bedoeld om lijsten en overzichten te genereren uit de in de database opgeslagen gegevens. Ook bij het ontwikkelen van rapporten kun je gebruik maken van wizards. Vaak wil je het resultaat daarvan nog handmatig aanpassen. Daarom bekijken we hier vooral manieren om rapporten handmatig aan te maken.

### 7.1 Eenvoudige rapporten

We bekijken het rapport **rptStukkenPerNiveau** dat er in afdrukweergave als volgt uit ziet:

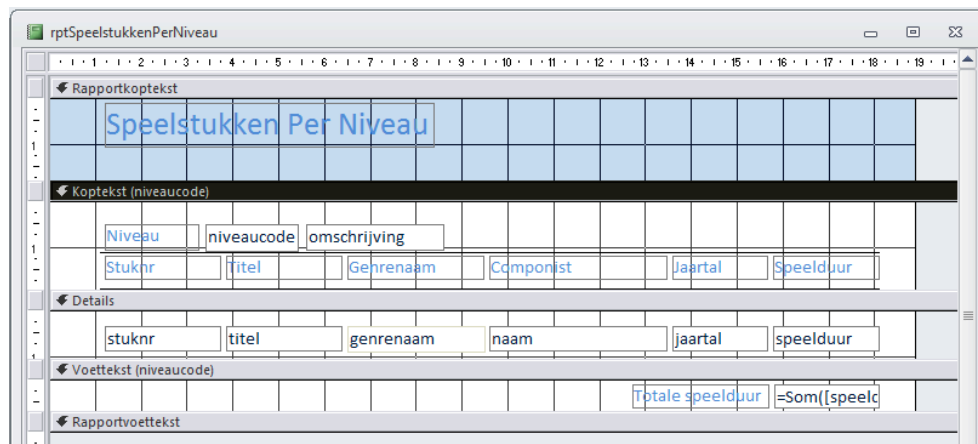
Speelstukken Per Niveau					
Niveau	A	beginners			
Stuknr	Titel	Genrenaa	Componist	Jaartal	Speelduur
12	I'll never go	pop	Karl Schumann	1996	6
14	Little Lina	klassiek	Sofie Bergeijk	1998	4,3
15	Blue sky	jazz	Jan van Maanen	1998	4
Totale speelduur					14,3
Niveau	B	gevorderden			
Stuknr	Titel	Genrenaa	Componist	Jaartal	Speelduur
2	Blue bird	jazz	Thom Guidi	1988	4
3	Air pur charme	klassiek	Rudolf Escher	1953	4,5
5	Lina		Sofie Bergeijk	1979	5
9	Cradle song	klassiek	Thom Guidi	1990	3,5
13	Swinging Lina	jazz	Jan van Maanen	1997	8
Totale speelduur					25

*rptSpeelstukkenPerNiveau*

De recordbron van dit rapport is onderstaande SQL query.

```
SELECT Stuknr, Stuk.ComponistId, Titel, Genrenaa,  
       Stuk.Niveaucode, Speelduur, Jaartal,  
       Omschrijving, Naam  
FROM   Niveau INNER JOIN Stuk  
       ON Niveau.Niveaucode = Stuk.Niveaucode  
       INNER JOIN Componist  
       ON Stuk.ComponistId = Componist.ComponistId
```

In de ontwerpweergave ziet dit rapport er als volgt uit:



Je ziet dat het rapport in diverse *secties* is opgedeeld: Rapportkopstekst, Koptekst (niveaucode), Details, Voettekst (niveaucode) en Rapportvoettekst.

De voettekst van een sectie wordt vaak gebruikt om (sub)totalen te berekenen. In dit geval wordt de totale speelduur per niveau opgeteld. Om dit te realiseren krijgt het tekstvak in de Voettekst (niveaucode) de volgende **Besturingselementbron**: =Som([speelduur]). Naast de functie **Som**, kun je ook gebruik maken van bijvoorbeeld **Aantal** en **Gemiddelde**.

## 7.2 Sorteren en groeperen

Het definiëren van secties gaat m.b.v. de optie **Groeperen en Sorteren** van de **Ontwerp** ribbon.



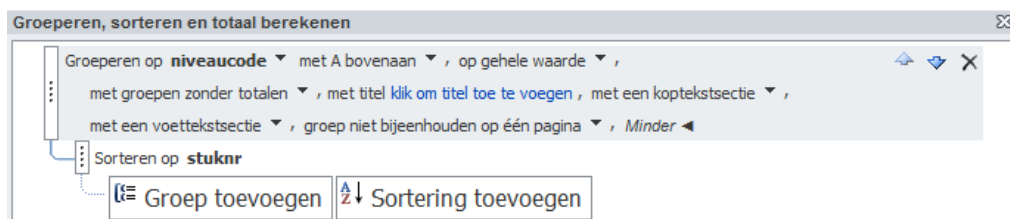
Je kunt in onderstaand venster (dat verschijnt onderin het scherm) opgeven op basis van welke kolom een nieuwe groep sectie gemaakt moet worden.



De rapportkopstekst-, rapportvoettekst- en detailsectie zijn altijd aanwezig.

Klik op **Meer** voor het bekijken van alle opties en huidige instellingen.

In dit geval wordt er gegroepeerd en gesorteerd (met A bovenaan) op **niveaucode** van de stukken en daarbinnen op **stuknr** gesorteerd.



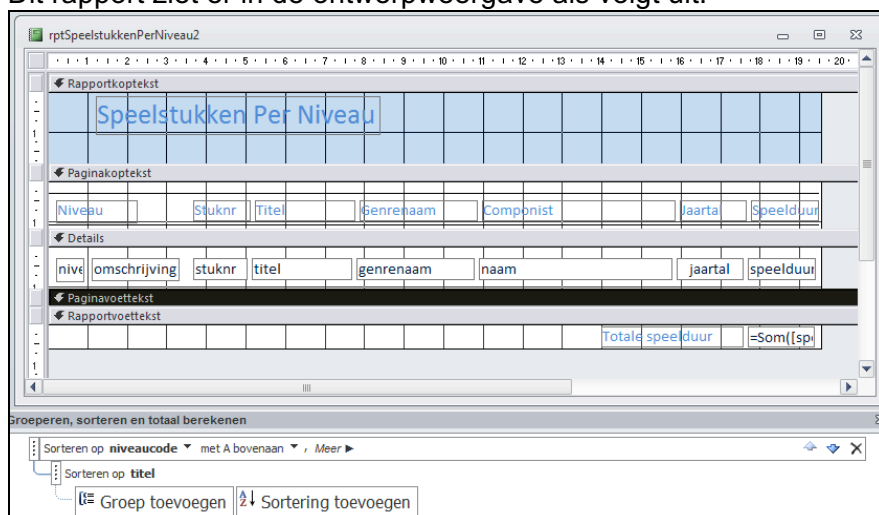
Bij niveaucode is aangegeven dat er zowel een koptekstsectie als een voettekstsectie gemaakt moet worden.

We bekijken een variant van het zojuist getoonde rapport (met dezelfde recordbron), we noemen deze variant **rptSpeelstukkenPerNiveau2**, dat het volgende overzicht geeft:

Speelstukken Per Niveau							
Niveau		Stuknr	Titel	Genrenaam	Componist	Jaartal	Speelduur
A	beginners	15	Blue sky	jazz	Jan van Maanen	1998	4
		12	I'll never go	pop	Karl Schumann	1996	6
		14	Little Lina	klassiek	Sofie Bergeijk	1998	4,3
B	gevorderden	3	Air pur charme	klassiek	Rudolf Escher	1953	4,5
		2	Blue bird	jazz	Thom Guidi	1988	4
		9	Cradle song	klassiek	Thom Guidi	1990	3,5
		5	Lina		Sofie Bergeijk	1979	5
		13	Swinging Lina	jazz	Jan van Maanen	1997	8
Totale speelduur							39,3

*rptSpeelstukkenPerNiveau2*

Dit rapport ziet er in de ontwerpweergave als volgt uit:



Enkele bijzonderheden van dit rapport:

- Er wordt gebruik gemaakt van de sectie Paginakoptekst; er wordt echter niet gegroepeerd zoals in het rapport **rptSpeelstukkenPerNiveau**;
- Voor de eerste twee tekstvakken in de Detail-sectie (niveaucode en omschrijving) is de eigenschap **Duplicaten verbergen** op **Ja** gezet. Dit heeft tot gevolg dat de gegevens, die in opeenvolgende rijen hetzelfde zijn, niet steeds worden herhaald.

- De totale speelduur staat nu in de Rapportvoettekst.
- Voor een ACCDB project zou dit ook in de Paginavoettekst gezet kunnen worden om een totaal telling per pagina te krijgen. In een ADP project (maar ook in een MDB project) is het helaas niet mogelijk om deze totaal telling per pagina te doen zoals hierboven staat. Daar zijn wel alternatieven voor, maar daar gaan we hier verder niet op in<sup>6</sup>.

### 7.3 Openen van een rapport vanaf een formulier

Stel we willen in het voorgaande rapport alleen stukken zien die in een bepaalde periode zijn gecomponeerd. Voor het invullen van de gewenste periode gebruiken we het volgende formulier **frmAfdrukkenStukken**.

Het beginjaar van de periode wordt ingevuld in het tekstvak **txtBeginjaar** en het eindjaar in het tekstvak **txtEindjaar**. Als dan vervolgens op de knop **cmdPreview** wordt geklikt, dan wordt het rapport **rptStukkenPerNiveau2** geopend en worden de stukken, die in de betreffende periode zijn gecomponeerd, getoond.

Als beide tekstvakken leeg zijn, dan moeten alle stukken worden getoond. Is alleen het tekstvak **txtBeginjaar** leeg, dan willen we alle stukken tonen, die voor of in het eindjaar gecomponeerd zijn. Is alleen het tekstvak **txtEindjaar** leeg, dan willen we alle stukken tonen, die in of na het beginjaar gecomponeerd zijn.

frmAfdrukkenStukken

De 'Bij klikken'-gebeurtenisprocedure van de knop **cmdPreview** is hieronder afgebeeld.

```
Private Sub cmdPreview_Click()

    Dim sWhereCondition As String

    If IsNull(Me.txtBeginjaar) And IsNull(Me.txtEindjaar) Then

        sWhereCondition = ""

    ElseIf IsNull(Me.txtBeginjaar) Then

        sWhereCondition = "JAARTAL <= " & Me.txtEindjaar

    ElseIf IsNull(Me.txtEindjaar) Then

        sWhereCondition = "JAARTAL >= " & Me.txtBeginjaar

    Else

        sWhereCondition = "JAARTAL BETWEEN " & _
            Me.txtBeginjaar & _
            " AND " & Me.txtEindjaar

    End If

    DoCmd.OpenReport "rptSpeelStukkenPerNiveau2", _
        acPreview, sWhereCondition

End Sub
```

<sup>6</sup> Voor een alternatief zie <http://support.microsoft.com/kb/296249/en-us>.

## 7.4 Subrapporten

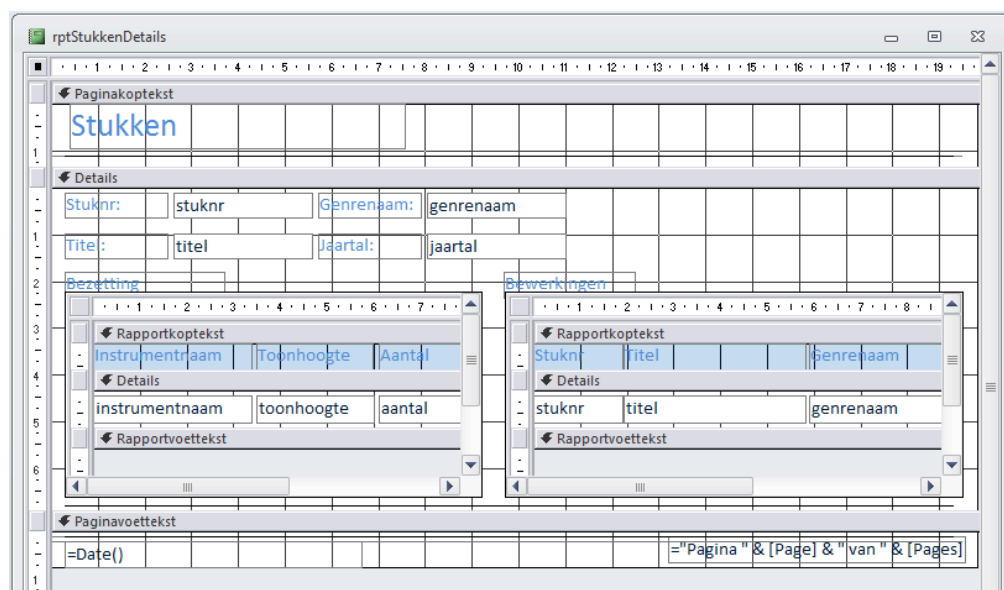
In deze paragraaf bekijken we een voorbeeld van een rapport waarbij per stuk de bezetting en de bewerkingen worden getoond. Het rapport heet **rptStukkenDetails**. Een gedeelte van het rapport is hieronder afgebeeld:

Stukken																	
Stuknr:	1	Genre naam: jazz															
Titel:	Blue bird	Jaartal: 1954															
Bezetting	<table><thead><tr><th>Stuknr</th><th>Titel</th><th>Genre naam</th></tr></thead><tbody><tr><td>2</td><td>Blue bird</td><td>jazz</td></tr><tr><td>15</td><td>Blue sky</td><td>jazz</td></tr></tbody></table>		Stuknr	Titel	Genre naam	2	Blue bird	jazz	15	Blue sky	jazz						
Stuknr	Titel	Genre naam															
2	Blue bird	jazz															
15	Blue sky	jazz															
Stuknr:	2	Genre naam: jazz															
Titel:	Blue bird	Jaartal: 1988															
Bezetting	<table><thead><tr><th>Instrument naam</th><th>Toonhoogte</th><th>Aantal</th></tr></thead><tbody><tr><td>drums</td><td></td><td>1</td></tr><tr><td>piano</td><td></td><td>1</td></tr><tr><td>saxofoon</td><td>alt</td><td>2</td></tr><tr><td>saxofoon</td><td>tenor</td><td>1</td></tr></tbody></table>		Instrument naam	Toonhoogte	Aantal	drums		1	piano		1	saxofoon	alt	2	saxofoon	tenor	1
Instrument naam	Toonhoogte	Aantal															
drums		1															
piano		1															
saxofoon	alt	2															
saxofoon	tenor	1															
Stuknr:	3	Genre naam: klassiek															
Titel:	Air pur charmer	Jaartal: 1953															
Bezetting	<table><thead><tr><th>Instrument naam</th><th>Toonhoogte</th><th>Aantal</th></tr></thead><tbody><tr><td>fluit</td><td></td><td>1</td></tr></tbody></table>		Instrument naam	Toonhoogte	Aantal	fluit		1									
Instrument naam	Toonhoogte	Aantal															
fluit		1															

woensdag 23 november 2011 Pagina 1 van 4

*rptStukkenDetails*

Een subrapport komt van pas als je twee detailsecties nodig zou hebben. In het voorbeeld zie je details over de bezetting en details over de bewerkingen van ieder stuk. Dat zou op deze manier niet lukken met alleen een hoofdrapport. De ontwerpweergave ziet er zo uit:



De recordbron van het hoofdrapport is gebaseerd op de tabel Stuk. In de details sectie van het hoofdrapport staan alle stukgegevens. De twee subrapportcontrols hebben de volgende eigenschappen:

#### rsubBezettingsregel

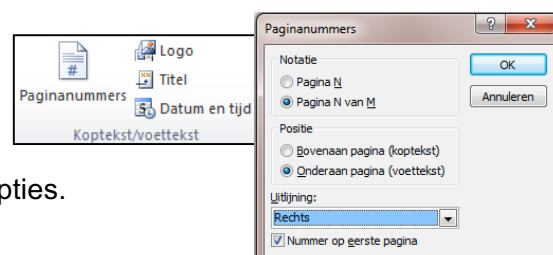
Hoofdvelden koppelen:	Stuknr
Subvelden koppelen:	Stuknr
BronObject	Rapport.rsubBezettingsregel
Te vergroten:	Ja
Te verkleinen:	Ja

#### rsubBewerkingen

Hoofdvelden koppelen:	Stuknr
Subvelden koppelen:	StuknrOrigineel
BronObject	Rapport.rsubStukken
Te vergroten:	Ja
Te verkleinen:	Ja

De recordbronnen van de subrapporten **Rapport.rsubBezettingsregel** en **Rapport.rsubStukken** zijn achtereenvolgens de tabel **Bezetting** en de tabel **Stuk**.

Het toevoegen van paginanummering en datum, tijd gaat gemakkelijk via de Ontwerp ribbon. Gebruik daarvoor deze opties.



## 8 Menu's en opstartopties

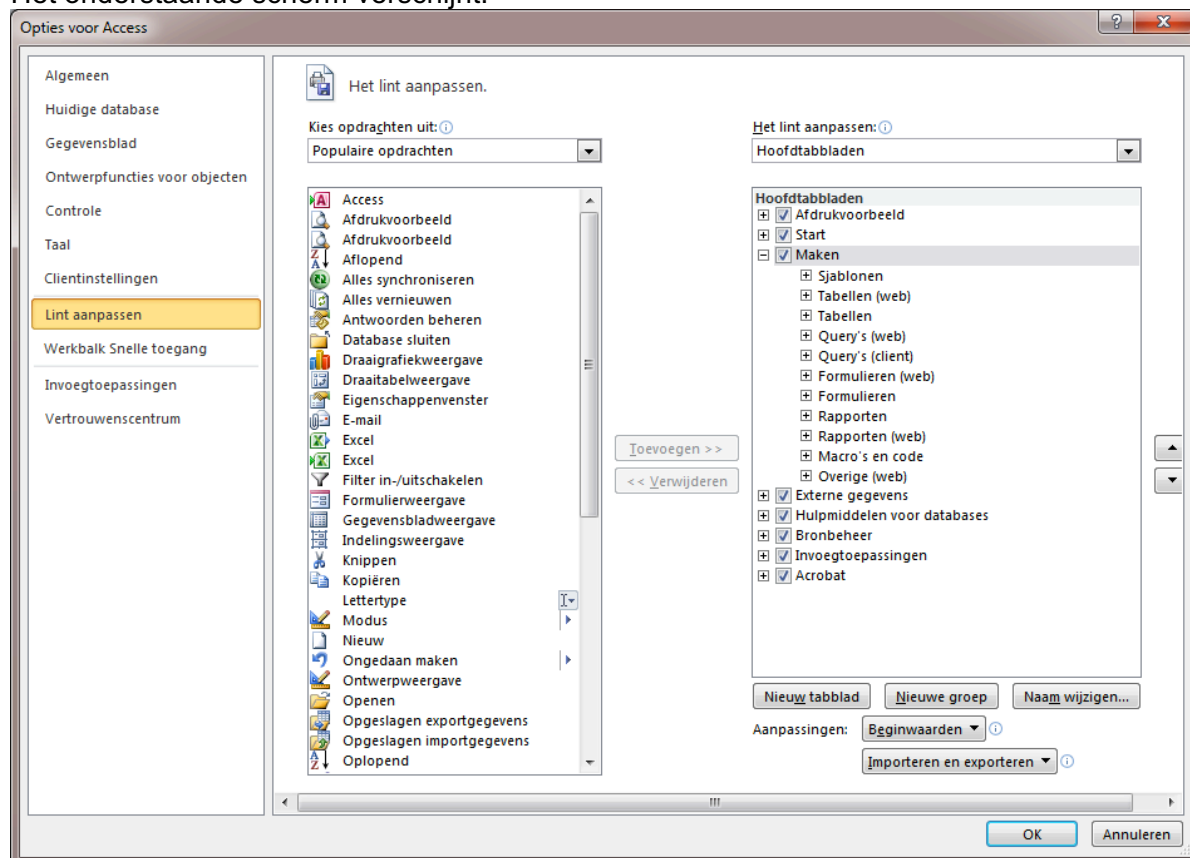
### 8.1 Menu's

Uiteindelijk dienen alle formulieren en rapporten ondergebracht te worden in een menu. Een globale beschrijving van het maken van een lint (ook wel ribbon geneemd) vind je hieronder. Het maken van een werkbalk voor snelle toegang gaat op dezelfde wijze.

#### 8.1.1 Het maken van een lint via de Access Opties

MS Access is het op meerdere manieren mogelijk om een eigengemaakt lint te definiëren. De meest gemakkelijke (maar niet meest uitgebreide) manier staat hieronder uitgelegd.

Kies **Bestand** en vervolgens **Opties** gevolgd door de optie **Lint Aanpassen**. Het onderstaande scherm verschijnt.



Hier kan men nieuwe tabbladen aanmaken, het tabblad indelen in groepen en in groepen acties definiëren.

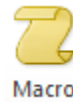


Geef het tabblad en de groep een eigen naam, door de standaard naam te wijzigen.

Het openen van zelf gemaakte formulieren of rapporten vanaf het werklint gaat middels het maken van macro's. Macro's kunnen vervolgens in een groep geplaatst worden.

Hieronder volgt een voorbeeld hoe een macro ingesteld kan worden voor het openen van een formulier:

Kies in het menu voor **Maken** en vervolgens **Macro**.



Kies als **Nieuwe actie** de optie **Formulier openen**. Het volgende scherm verschijnt:

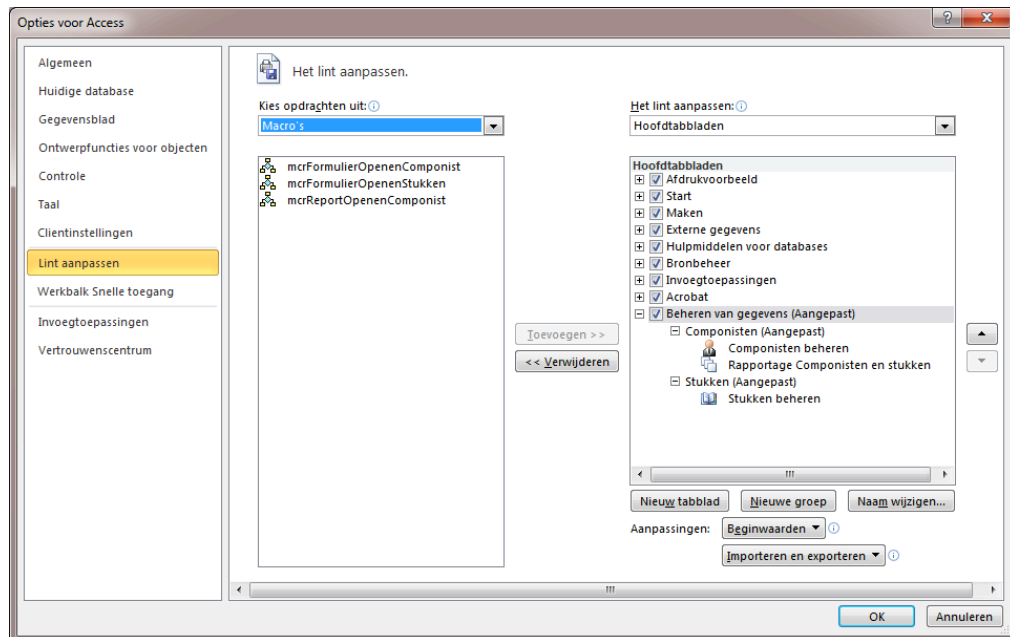
Deze stel je, bijvoorbeeld voor het openen van het formulier **frmComponist** als volgt in:

Nu de macro eenmaal gemaakt is kan deze aan het lint worden toegevoegd.

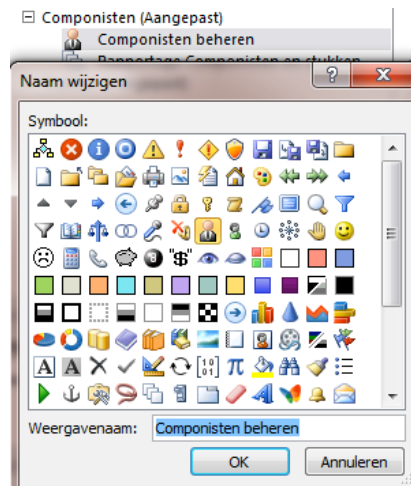
Keer terug naar **Bestand, Opties** gevolgd door de optie **Lint Aanpassen**.

Sleep vervolgens de gemaakte macro's naar de betreffende groep zoals hieronder afgebeeld,

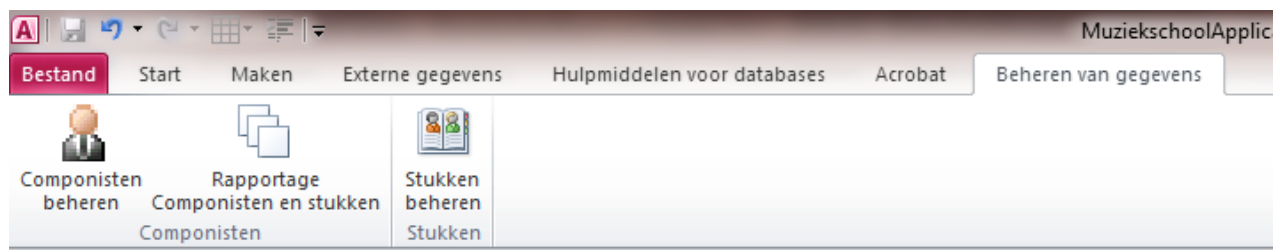




Om een gebruikersvriendelijke naam van de knop op het werklint te krijgen moet je de naam wijzigen:



Het resultaat is dan (in dit voorbeeld):



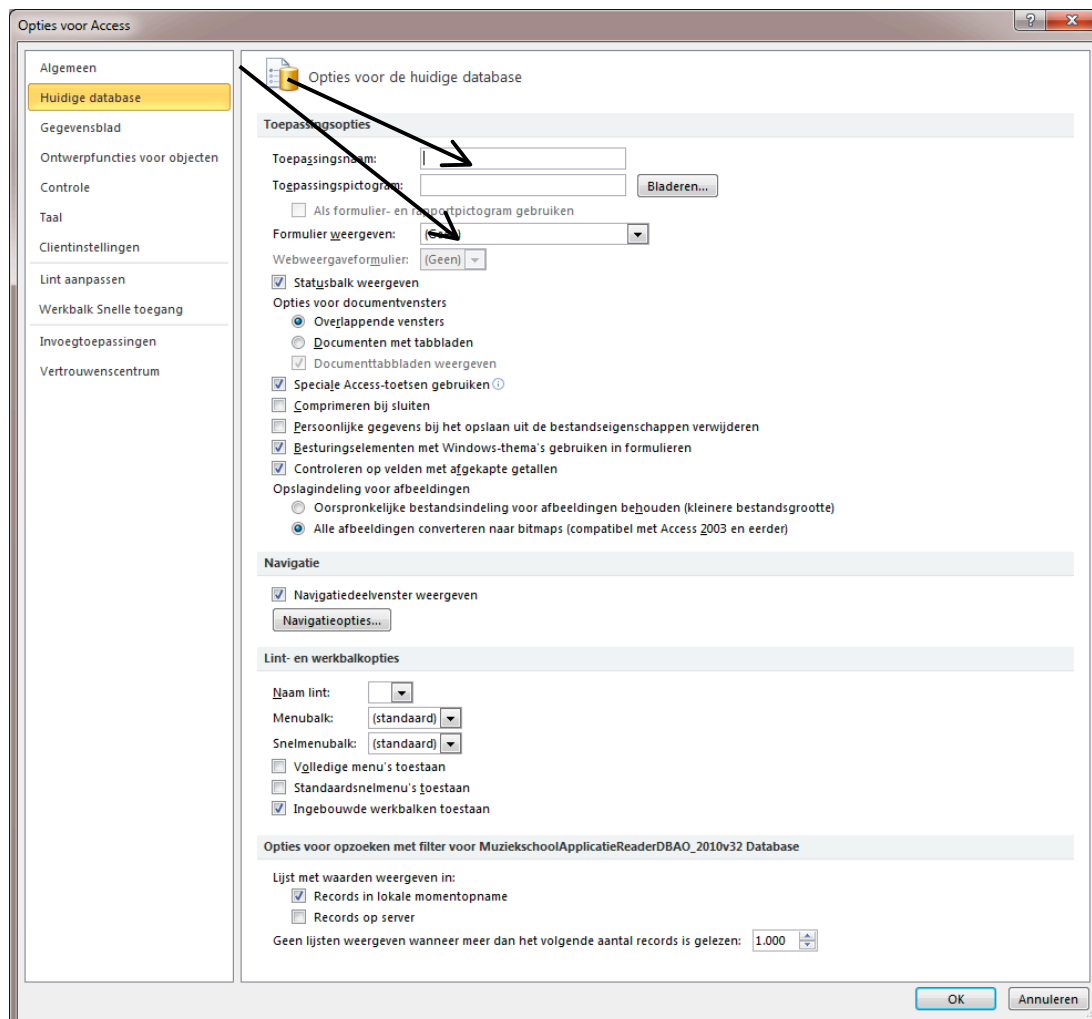
Dit lint blijft de gehele applicatie actief en is dus niet context gevoelig. Standaard staan ingebouwde werkbalken (zoals dit lint) 'aan'. Via **Bestand | Opties | Huidige Database** kan deze opties aangepast worden.

### **8.1.2 Het maken van een lint in XML en openen vanuit VBA**

Een krachtige maar tevens complexe manier van het creëren van een interactief lint is gebaseerd op het definiëren van het lint in een XML structuur. Dit wordt in deze course niet gedaan. Lees eventueel meer hierover op <http://msdn.microsoft.com/en-us/library/dd548010.aspx> of bekijk de presentaties op <http://online.han.nl> .

## 8.2 Opstartopties

In het scherm voor opstartinstellingen (**Bestand | Opties | Huidige Database**) kun je je applicatie een naam en pictogram geven en bovendien aangeven welk formulier er getoond wordt bij het opstarten.



Je kunt hier tevens de standaard menubalken uitzetten en instellen dat er geen wijzigingen meer mogen worden gemaakt in jouw eigen menu. Ook kun je instellen dat het database-venster niet getoond wordt en ook niet op te roepen is met F11.

Let op: je kunt er dan zelf alleen nog bij als je het bestand (MDB, ACCDB, ADP) opstart met de SHIFT toets ingedrukt.

## Bijlage 1: Stappenplan voor het maken van een formulier

Onderstaand stappenplan bevat enkele schermafdrucken van MS Access versie 2003. Het stappenplan is echter nog steeds up-to-date en kan derhalve ook gebruikt worden voor versies 2007, 2010, 2012 en 2016.

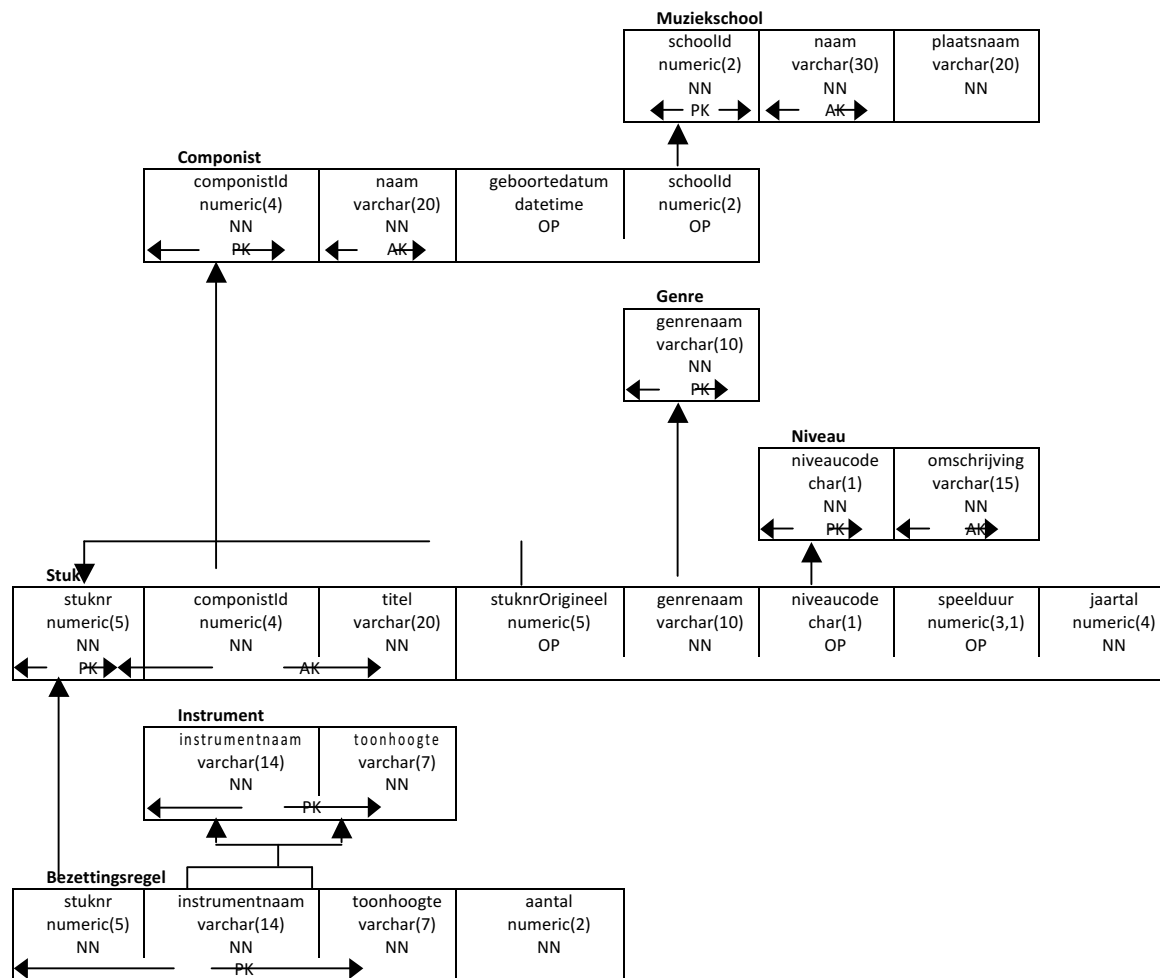
1. Bestudeer het bedrijfsproces dat ondersteund gaat worden door dit formulier.
2. Bepaal de tabel die de basis gaat vormen voor de recordsource van het formulier.
3. Bekijk welke kolommen van deze tabel foreign keys zijn.
4. Maak een query waarin je de basistabel joint met de tabellen waarnaar verwezen wordt door de foreign keys. Zorg dat in de query alle kolommen van de hoofdtabel en de omschrijvende kolommen van de gerelateerde tabellen worden geselecteerd. Als er geen foreign keys in de basistabel zitten kun je die rechtstreeks als recordsource gebruiken.
5. Bepaal of het bedrijfsproces het best gebaat is bij een formulier in enkelvoudige weergave of bij een doorlopend formulier. In het algemeen geldt: veel kolommen en weinig records, dan enkelvoudige weergave; weinig kolommen en veel records, dan doorlopende weergave.
6. Maak nu, met de wizard of met de hand, een formulier dat deze query als recordsource gebruikt. Zorg dat het formulier er netjes uit ziet.
7. De foreign key velden moeten comboboxen zijn, gevuld met de primary key waarden van de gerelateerde tabel. In MS Access kun je dit al bij het tabelontwerp opgeven!
8. Zorg dat de omschrijvende velden uit de gerelateerde tabellen disabled zijn.
9. Bepaal welke velden het handigst zijn om op te zoeken voor het bedrijfsproces.
10. Zet in de header-sectie van het formulier unbound controls voor alle zoekcriteria.
11. Hier volgt een idee voor het zoeken, er zijn vele andere mogelijkheden. In een client-server situatie is het beter om deze clientside filter niet te gebruiken\*. Zorg dat met een event (dat kan het after-update event van zo'n unbound control zijn, maar je zou ook een knop kunnen gebruiken) de eigenschap Me.Filter wordt ingesteld. Deze eigenschap bevat een conditie, vergelijkbaar met een SQL WHERE conditie. Zet vervolgens de eigenschap Me.FilterOn op True.
12. Maak ook een mogelijkheid om het filter weer uit te zetten.
13. Bekijk nu welke tabellen een foreign key verwijzing *naar* de basistabel hebben. Hiervoor gaan we subformulieren maken.
14. Maak voor ieder van deze tabellen een query waarin je deze basistabel van het subformulier joint met alle tabellen waar deze tabel een foreign key relatie naartoe heeft, behalve de basistabel van het hoofdformulier. Selecteer in de query alle kolommen van de basistabel van het subformulier en alle beschrijvende kolommen van de andere tabellen.
15. Maak nu een subformulier (met de hand of met de wizard) dat deze query als recordsource gebruikt. Gebruik voor een subformulier altijd doorlopende weergave of

gegevensbladweergave.

16. Stel de juiste koppeling tussen hoofd- en subformulier in. In de eigenschap Hoofdvelen koppelen zet je de primary key kolom van de basistabel van het hoofdformulier, in de eigenschap Subvelen koppelen zet je de foreign key kolom van de basistabel van het subformulier die verwijst naar de basistabel van het hoofdformulier.
17. De foreign key velden moeten comboboxen zijn, gevuld met de primary key waarden van de gerelateerde tabel.
18. Zorg dat de omschrijvende velden uit de gerelateerde tabellen disabled zijn.
19. Zorg dat je de foreign key kolom van de basistabel van het subformulier die verwijst naar de basistabel van het hoofdformulier niet wordt getoond.
20. Zet de navigatieknoppen van het subformulier uit.
21. Gebruik tabbladen als er meer dan één subformulier is.
- (22.) (Zet een knop op het subformulier waarmee je het formulier kunt openen waarvan de hoofdtabel dezelfde is als de basistabel van het subformulier. Zorg dat het filter is ingesteld op de primary key waarde van het eerste hoofdformulier en dat het geselecteerde record hetzelfde is als het geselecteerde record op het subformulier.)

\* Zie paragraaf *Recordset en zoekfunctionaliteit in een client/server situatie*.

## Bijlage 2: Database-schema Muziekdatabase<sup>7</sup>



<sup>7</sup> Uit: *Relationele Databases & SQL*, Leo Wiegerink e.a., Academic Service, 2<sup>e</sup> druk, ISBN 90.395.2221.9