

Handleiding

Access 2003/2007

(Versie 6.5)

behorende bij het vak OGO 1.1: Van Model tot Systeem (2IO05)

Opleiding Technische Informatica

Inhoudsopgave:

1.	INLEIDING MICROSOFT ACCESS	4
1.1.	Opstarten van Access	4
1.2.	Definiëren van tabellen	5
1.2.1.	Het aanmaken van een nieuwe tabeldefinitie	5
1.2.2.	De structuur veranderen van een tabel	7
1.2.3.	Toekennen van eigenschappen aan een veld	7
1.2.4.	Het importeren van de Vraag tabel	9
1.2.5.	Relaties tussen verschillende tabellen	9
1.3.	Het invoeren en wijzigen van gegevens in tabellen	10
1.3.1.	Het invoeren van gegevens in een tabel	10
1.3.2.	Het wijzigen van gegevens in een tabel	12
1.3.3.	Het verwijderen van gegevens uit een tabel	12
1.4.	Het formuleren van query's	13
1.4.1.	Het formuleren van een query op één tabel	13
1.4.2.	Het aanmaken van een query op meerdere tabellen	14
1.4.3.	Het veranderen van een Select query in een Make-Table query	15
1.4.4.	Het wijzigen van het uitvoerformaat van een bestaande query	15
1.4.5.	Het opstellen van een query die een tabel wijzigt	16
1.5.	Formulieren	17
1.5.1.	Het aanmaken van een nieuw formulier op één tabel	17
1.5.2.	Het aanmaken van een nieuw formulier op een query	20
1.5.3.	Het aanmaken van een nieuw formulier op meerdere tabellen	21
1.5.4.	Het wijzigen van tabellen door middel van een formulier	21
1.6.	Rapporten	22
1.6.1.	Het aanmaken van een nieuw rapport voor één tabel	22
1.6.2.	Het aanmaken van een nieuw rapport voor meerdere tabellen	23
1.6.3.	Het aanmaken van een nieuw rapport op een query in plaats van een tabel	23
2.	INLEIDING DATABASETOEPASSINGEN	24
2.1.	Introductie	24
2.2.	Een eenvoudig startformulier	24
2.2.1.	Toevoegen van een knop aan een formulier	25
2.2.2.	Toevoegen van een label aan een formulier	26
2.2.3.	Instellen van eigenschappen: Property sheets	28
2.3.	Een eerste versie van de databasetoepassing	30
2.3.1.	Opstarten van een databasetoepassing	30
2.3.2.	Intermezzo: de werking van knoppen	31
2.3.3.	Een met de Wizard gemaakte knop aanpassen	34
3.	ENIGE UITBREIDINGEN	37
3.1.	Het koppelen van besturingselementen aan een gegevensbron	37
3.1.1.	Wijzigingen in de tabelstructuur verwerken in een formulier	37
3.1.2.	Toevoegen van comboboxen	39
3.1.3.	Informatie doorgeven tussen formulieren	40
3.2.	Default formulieren en componenten	43
3.3.	Rapporten	44

4.	EEN GROTER VOORBEELD	46
4.1.	Vooruitblik	46
4.2.	Het voorbeeld	46
4.3.	Tabelstructuur	47
4.4.	Functionaliteit	48
4.5.	Procedure voor het afnemen van een enquête	49
5.	FORMULIEREN	51
5.1.	frmAnswerIdsToElts, frmMainMenu en frmAnswerIds	51
5.2.	frmEnterQuestions	53
5.3.	frmEnterProjectMembers	58
5.4.	frmStartInterview	58
5.5.	frmQuestionnaire	61
6.	BEVEILIGING	64
7.	DATABASE WIZARD	67
BIJLAGE A.	DE TOOLBOX	70
BIJLAGE B.	PROPERTY SHEETS	71
a.	Formulieren	71
b.	Controls	72
BIJLAGE C.	PROGRAMMEREN IN VISUAL BASIC FOR APPLICATIONS	74
a.	Visual Basic	74
b.	Access specifieke uitbreidingen	76


1. Inleiding Microsoft Access

1.1. Opstarten van Access

*Doel: Na afloop moet Access for Windows opgestart zijn en dient een nieuwe database met de naam **OGO Enquete** aangemaakt te zijn.*

Access kan op verschillende manier vanuit Windows worden opgestart. Dit aantal is afhankelijk van de wijze waarop het programma is geïnstalleerd. Twee mogelijke manieren zijn:

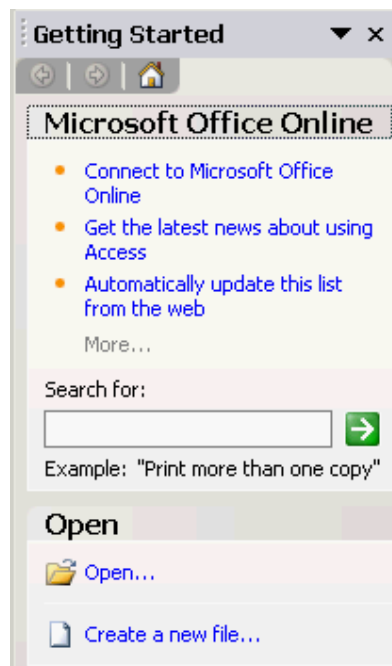
1. Via het startmenu  :

Druk op dit icoon en ga in het menu met de cursor naar 'P'rograms'. Vervolgens zal er een nieuw menu te voorschijn komen dat ook weer submenu's kent. In één van deze menu's zal de icoon  Microsoft Office Access 2003 (of 2007) zich bevinden. Druk hier met de **Select**-toets op en het programma wordt opgestart.

2. Via de zogenaamde snelkoppeling of 'short-cut':

Wanneer dit icoon zich bevindt op het opstartscherm van Windows, volstaat het om Access op te starten door twee keer snel achter elkaar, wat 'dubbelklikken' wordt genoemd, op de **Select**-toets te drukken terwijl de cursor op de afbeelding staat.

Als het programma is opgestart verschijnt er aan de zijkant van het Access window het volgende venster op de monitor:



Klik vervolgens "Create a new file" en dan 'Blank Database' aan. Vervolgens verschijnt een venster waarin men een directory kan selecteren en waar men in het veld achter 'File name:' de naam van de nieuwe database dient in te vullen. Ga naar een geschikte directory, typ **OGO Enquete** in en druk vervolgens op **Create**.

In ons voorbeeld gaan we een enquêtedatabase samenstellen waarmee de enquête die aan het eind van OGO1.1 gehouden wordt ook in elektronische vorm kan worden afgenomen. De eerste versie van deze enquête is heel eenvoudig, met alleen getallen als antwoorden. In het tweede deel van de cursus wordt de database uitgebreid met onder andere meer-keuze vragen, waarbij de antwoorden uit een vooraf gedefinieerde verzameling gekozen moeten worden.

We gebruiken het Access 2000 file formaat (default ingesteld). Als je deze instelling veranderd hebt, stel dan weer het Access 2000 file formaat in via tools menu – kies options en stel het file formaat in op het “advanced” tabblad.

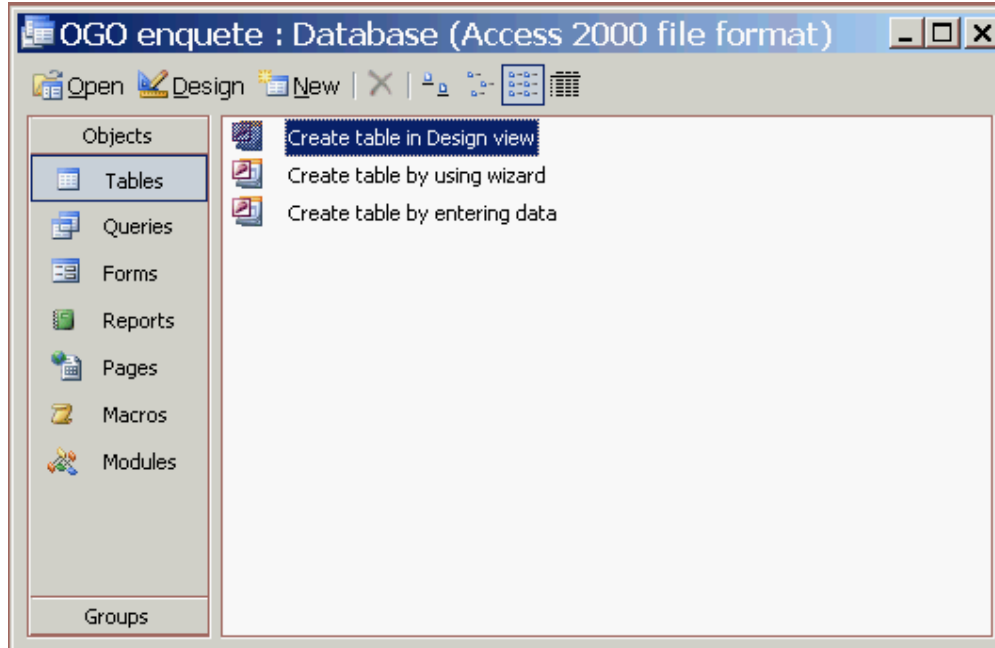
1.2. Definiëren van tabellen


*Doel: Na afloop moeten de tabellen **Enquete**, **Student** en **Antwoord** zijn gedefinieerd met enkele relaties daartussen. De tabel **Vraag** moet zijn geïmporteerd uit de **OGO Enquete start** database.*


1.2.1. Het aanmaken van een nieuwe tabeldefinitie

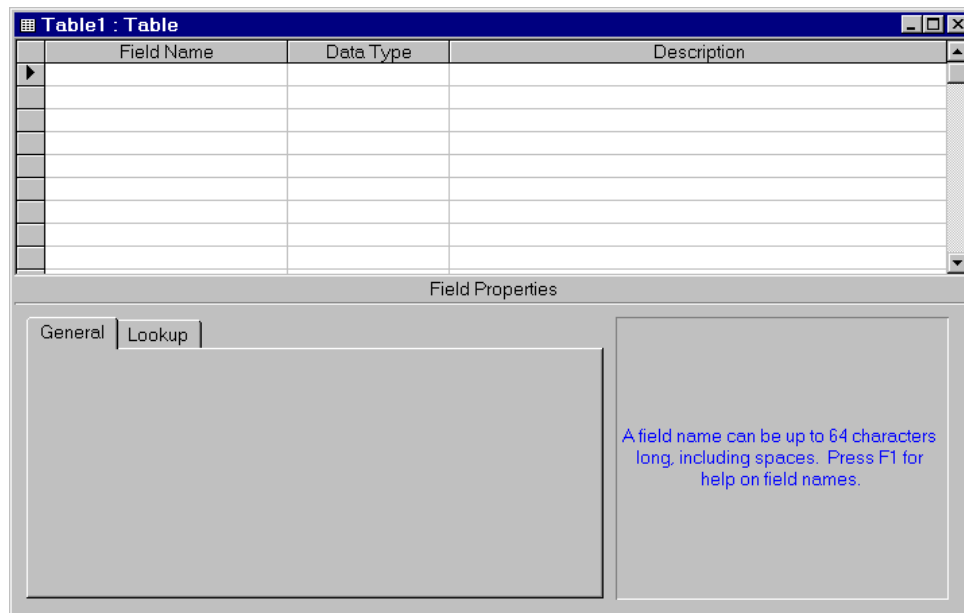
Doel: Na afloop moet tabel **Enquete** aangemaakt zijn met het veld **EnqueteNr** dat als primaire sleutel dient.

Nadat de nieuwe database is aangemaakt verschijnt het volgende venster op het scherm. Dit venster zullen we het ‘Database venster’ noemen.



In dit venster kunnen 7 groepen objecten (Tables, Queries, Forms, Reports, Pages, Macros en Modules) geselecteerd worden, waarvan de ‘Tables’ groep geopend is. Druk op . Selecteer de optie ‘Design View’ en druk op **OK**. De structuur van de nieuwe tabel verschijnt.

De tabel kan ook aangemaakt worden door te dubbelklikken op  **Create table in Design view**.



- Maak het structuurvenster schermvullend met de maximaliseerknop rechts boven in de titelbalk van het venster.

Bovenaan dit venster zijn drie kolommen, namelijk:

1. 'Field name': De namen van de velden in de tabel.
2. 'Data Type': Het datatype van de veldnamen.
3. 'Description': Informatie over een veld; mag leeg worden gelaten.

Invoeren eerste veld:

- Typ *EnqueteNr* in het eerste veld van de kolom 'Field name'.
- Druk op <Tab> of op de rechter cursortoets om de cursor naar de volgende kolom te verplaatsen.
- ▾ • Klik op de afrolknop in dit veld om de beschikbare veldtypes te bekijken en selecteer hieruit 'AutoNumber'. Dat wil zeggen dat Access voor dit veld zelf een waarde kiest als er een nieuw record wordt aangemaakt.
- Typ als beschrijving *Uniek nummer van een enquete*.

Nu moeten we nog aan het veld *EnqueteNr* de primaire sleutel-eigenschap toekennen.

- Ga met de cursor van je muis in de grijze kolom staan, links van de 'Field Name' kolom ter hoogte van *EnqueteNr*. Druk op de **Select**-muisknop om het hele veld te selecteren.

	Field Name	Data Type	Description
▶	EnqueteNr	AutoNumber	Uniek nummer van een enquete

- Druk vervolgens op de icoon 'Primary Key' in de werkbalk en de sleutel is toegekend.

Het toekennen van de sleutel-eigenschap aan meerdere velden gebeurt door eerst de desbetreffende velden te selecteren en vervolgens op de 'Primary Key' icoon te drukken. Om snel meer dan één veld selecteren, selecteer je eerst één veld. Vervolgens houd je <Ctrl> ingedrukt, terwijl je met de **Select**-muisknop de andere velden selecteert.

Wegschrijven en afsluiten:

- Kies uit het menu **File** de optie **Save As...** Het venster 'Save As' verschijnt.
- Ken aan de gedefinieerde tabel de naam **Enquete** toe en druk op **OK**.
- Kies **File, Close**.

De eerste tabel is nu weggeschreven.



Men kan ook tussentijds de tabelstructuur opslaan. Gebruik daarvoor de knop 'Save' in de werkbalk (of 'Toolbar').



In plaats van de combinatie **File, Close** kan men ook op sluitknop drukken die rechtsboven in het structuurvenster zit.

1.2.2. De structuur veranderen van een tabel

*Doel: Na afloop dienen de velden **StudentId** en **Datum** te zijn toegevoegd aan tabel **Enquete**.*

- Selecteer de tabel **Enquete** door erop te klikken (niet dubbelklikken!) als deze nog niet geselecteerd is en druk vervolgens op **Design**. 
- Het eerste veld in de eerste regel is geselecteerd. Als je nu deze naam zou willen aanpassen dan kun je op <F2> drukken en kun je de naam aanpassen. Een andere manier is te klikken met je muis op de plaats waar je de naam wilt gaan veranderen. Zou je de naam in zijn geheel willen veranderen dan kan volstaan worden door meteen de nieuwe naam in te tikken.
- We willen een nieuw veld invoeren. Ga met je cursortoetsen 1 veld naar beneden. Dit kan ook door 3 keer op je Tab-toets te drukken of door met je muis op dit veld te klikken.
- Tik hier `StudentId` in en ken het type `Text` toe (default waarde).
- Maak op dezelfde manier nog een tweede veld met naam `Datum` en type `Date/Time`.
- Schrijf de nieuwe tabelopmaak weg en sluit het venster.

Terzijde: We hebben de tabel **Enquete** feitelijk niet nodig in het eerste deel van de cursus. Deze tabel maakt het mogelijk meerdere enquetes met dezelfde student te houden, functionaliteit die in de laatste hoofdstukken wordt gerealiseerd.

1.2.3. Toekennen van eigenschappen aan een veld

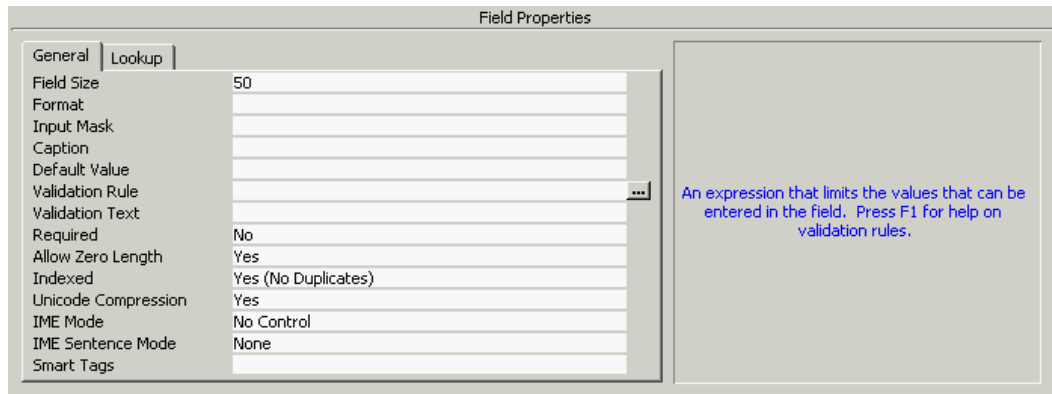
*Doel: Na afloop moet de tabel **Student** zijn aangemaakt. In deze tabel dienen de mogelijke waarden van het veld **Geslacht** beperkt te zijn en dient de invoer van het veld **StudentId** aan een bepaalde syntax te voldoen. Ook moet de tabel **Antwoord** aangemaakt zijn.*

We maken een tabel aan die enige informatie over studenten bevat. Uiteraard is in dit voorbeeld de hoeveelheid informatie zeer beperkt en dient slechts ter illustratie.

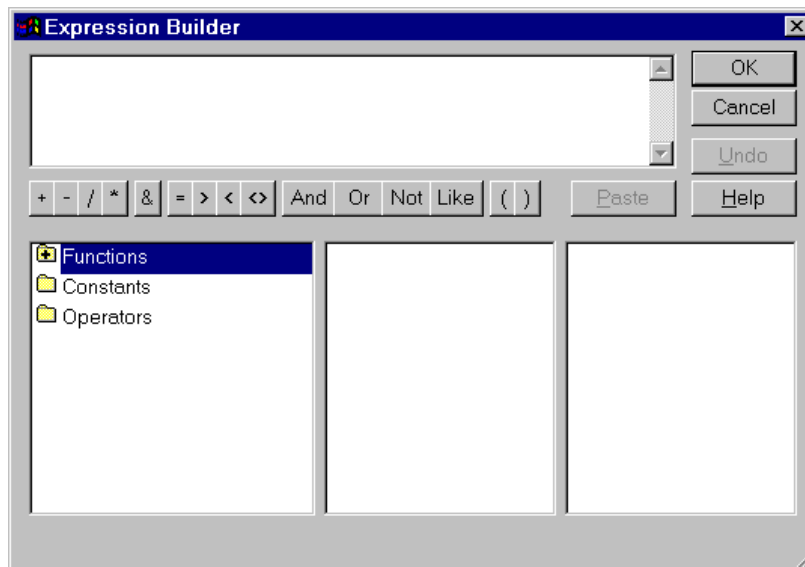
Maak een nieuwe tabel aan en voer de onderstaande informatie in (maak `StudentId` de primaire sleutel):

Veldnaam	Datatype	Omschrijving
StudentId	Text	Sleutel
Naam	Text	
Geslacht	Text	

Voor het geslacht zijn maar twee mogelijkheden, M en V. We willen deze beperking ook invoeren in de database. Ga naar het veld `Geslacht`. De 'Field Properties' zien er als volgt uit:



In bovenstaand scherm is de cursor (onzichtbaar) op het veld 'Validation Rule' gezet, waarna de hiernaast afgebeelde knop achter het veld verschijnt. Doe dit ook en druk vervolgens op deze knop. Het venster van de "Expression Builder" verschijnt:



Tik nu in het bovenste venster "M" of "V". Klik vervolgens op **OK**.

Daarnaast willen we ook dat de waarden altijd in hoofdletters in de tabel komen te staan. Zet daarom een > in het veld achter **Format**. Hierdoor wordt bijvoorbeeld 'm' automatisch omgezet in 'M'.

Het veld *StudentId* kent een bepaald formaat. Het bestaat namelijk uit één letter (S) gevolgd door 6 cijfers, waarvan de letter als hoofdletter wordt weggeschreven. Ga naar de 'Field Properties' van het veld *StudentId* en zet in het venster achter 'Input Mask' >L000000. De 'L' geeft aan dat de eerste invoer een letter moet zijn en de nullen geven aan dat de letter gevolgd moeten worden door 6 cijfers. Schrijf deze tabel weg als **Student** en sluit het venster. Breng dezelfde wijziging aan in de **Enquete** tabel.

Om snel te zien wat de betekenis van bepaalde **Field Properties** is, kan de F1 Help functie gebruikt worden. Door in de betreffende **Property** te klikken, en vervolgens de F1 knop in te drukken verschijnt direct het relevante Help scherm.

Maak ook een nieuwe tabel aan met de volgende structuur:

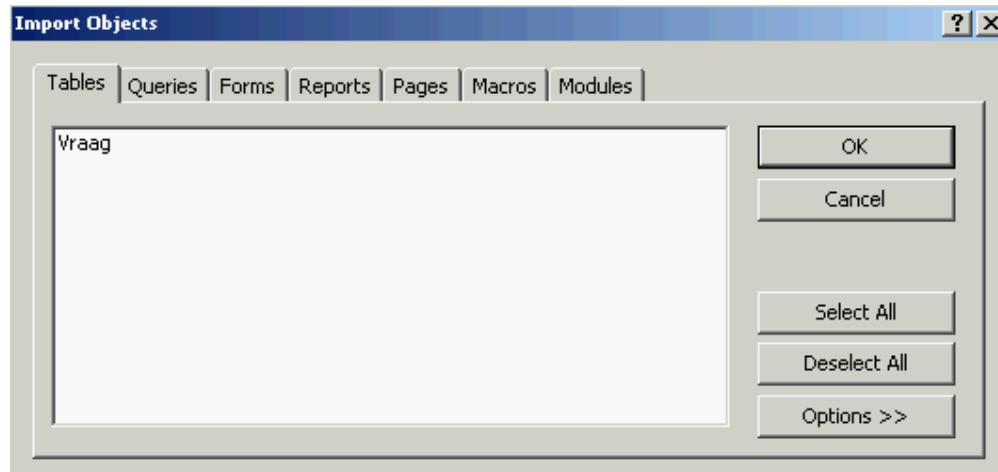
Veldnaam	Datatype	Omschrijving
VraagNr	Number	Sleutel
EnqueteNr	Number	Sleutel
Antwoord	Number	

Maak van de combinatie *VraagNr* en *EnqueteNr* de primaire sleutel: selecteer beide velden en druk op het "primaire sleutel" icoon. Schrijf de tabel weg als **Antwoord**. Sluit het venster.

1.2.4. Het importeren van de Vraag tabel

*Doel: na afloop dient de tabel **Vraag** uit de **OGO Enquete Start** database te zijn geïmporteerd.*

Selecteer in het **File** menu 'Get external data' en selecteer 'Import'. Navigeer naar de directory waar de **OGO Enquete Start** database staat en selecteer deze. Het **Import Objects** venster verschijnt:



Selecteer **Vraag** en druk op OK. Tabel **Vraag** is nu opgenomen in de database. Hierin zijn de vragen al opgenomen.

1.2.5. Relaties tussen verschillende tabellen

Doel: Na afloop dienen de relaties tussen de verschillende tabellen gespecificeerd te zijn.

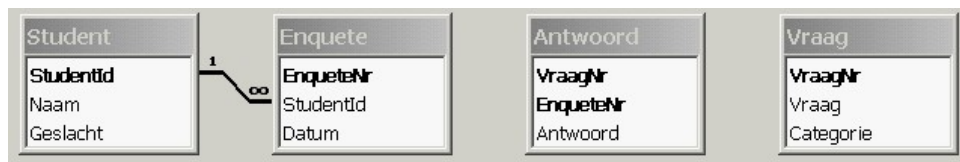
Vaak mogen gegevens niet zonder meer aan een tabel toegevoegd worden. Wanneer er bijvoorbeeld een nieuwe enquête aan de tabel **Enquete** wordt toegevoegd, hoort daar een verwijzing naar de geënqueteerde student bij. Deze student dient al te zijn opgenomen in de database. Dit kan opgezocht worden in de tabel **Student**. Een soortgelijk verhaal kan verteld worden over een antwoord op een bepaalde vraag. De betreffende enquête dient ook te bestaan in de tabel **Enquete**. We noemen zo'n samenhang tussen verschillende tabellen *Referentiële Integriteit*.

We gaan nu dergelijke relaties tussen de verschillende tabellen definiëren.



- Klik op de 'Relationships' icoon in de 'Toolbar'.
- Voeg vervolgens de tabellen **Student**, **Enquete**, **Antwoord** en **Vraag** met **Add** toe en druk op **Close**.
- Ga met de cursor naar de tabel **Student** en selecteer het veld *StudentId*. Houd de **Select**-muisknop ingedrukt en beweeg de cursor naar het *StudentId*-veld van de tabel **Enquete**. Als je op dit veld staat kun je de muisknop weer los laten. Het volgende venster dient nu te zijn geopend:

- Druk op 'Enforce Referential Integrity' zodat er een vinkje komt te staan.
- Zorg er voor dat er ook een vinkje komt te staan in de twee 'Cascade' velden. Dit zorgt er onder andere voor dat wanneer het *StudentId* van een student verandert, dit ook meteen wordt doorgevoerd in de tabel **Enquete**. Later zal hier uitgebreider op worden terug gekomen.
- Druk vervolgens op **Create**. Het 'Relationships'-venster ziet er nu zo uit:



De eerste relatie is nu gelegd. Doe hetzelfde met de *EnqueteNr*-velden van **Enquete** en **Antwoord** en met de *VraagNr* velden van **Antwoord** en **Vraag**. Merk op dat er nu oneindig-symbolen aan weerszijden van de **Antwoord** tabel staan: de **1** staat aan de kant waar een waarde maar een keer mag voorkomen. Sluit het venster en antwoord 'Yes' op de vraag of je de layout wilt opslaan.

1.3. Het invoeren en wijzigen van gegevens in tabellen

Doel: Na afloop bevatten alle tabellen gegevens en is bekend hoe men deze gegevens kan wijzigen.

1.3.1. Het invoeren van gegevens in een tabel

Doel: Na afloop moeten alle tabellen gegevens bevatten, genoeg om in het vervolg van het practicum vooruit te kunnen.

- Open de tabel **Student** vanuit het 'database-venster' door de tabel te selecteren en op **Open** te drukken, of door dubbel te klikken op de tabel.
- De cursor staat op het eerste veld; tik S100100 en ga naar het volgende veld; typ daar Reinders in en typ M in het derde veld. Ga met de tab-toets, de cursortoetsen of de muis naar het eerste veld van de tweede regel.
- Voer vanuit hier de volgende informatie in:

<i>StudentId</i>	<i>Naam</i>	<i>Geslacht</i>
S101101	Versteeg	M
S102102	Lammers	V
S113113	Bongers	M

Enkele opmerkingen:

- 1) Linksonder in het tabelvenster bevinden zich de navigatie-knoppen:



Bovenstaand voorbeeld wil zeggen dat de tabel 4 'records' of rijen bevat en dat de cursor zich bij 'record' 2 bevindt. De werking van de navigatieknoppen is als volgt:

Knop	Betekenis
	Ga naar het eerste record van de tabel
	Ga één record terug
	Ga één record vooruit
	Ga naar het laatste record in de tabel
	Ga naar een nieuw (leeg) record. Daar kan men nieuwe gegevens toevoegen



- 2) Men kan rechtstreeks teruggaan naar het structuurvenster door op de 'Design View' icoon te drukken dat linksboven in de 'Toolbar' zit. Weer terugspringen naar de gegevens kan door op de 'Data Sheet View' icoon te drukken dat op de oude plaats van de 'Design View' icoon zit.



- 3) Merk op dat kleine letters in de *StudentId*- en in *Geslacht*-velden meteen in hoofdletters worden omgezet. Dit geldt niet voor de kleine letters in het *Naam* veld.

Voer in de tabel **Enquete** de volgende gegevens in en sluit de tabel vervolgens af:

<i>EnqueteNr</i>	<i>StudentId</i>	<i>Datum</i>
1	S100100	20/08/2005
2	S101101	20/08/2005
3	S102102	20/08/2005
4	S113113	20/08/2005

NB: Afhankelijk van de instelling van de PC kan de datum 11 maart 2005 als 3/11/2005 of als 11/3/2005 moeten worden ingevoerd. Door een datum als 31/1/2005 in te voeren is dit snel te achterhalen. De settings zijn te veranderen m.b.v. de Regional Settings.

Probeer ook eens waarden voor *StudentId* en *EnqueteNr* in te voeren die niet in de andere tabellen zitten om te kijken wat er gebeurt. Als je een record hebt toegevoegd maar het programma keurt je nieuwe toevoeging niet goed, omdat bijvoorbeeld de student niet bestaat, dan kun je 'eruit' springen door op <Esc> te drukken. Het nieuwe en foute record wordt verwijderd.

Voer in de tabel **Antwoord** de volgende gegevens in:

VraagNr	EnqueteNr	Antwoord
4	1	1
4	2	4
4	3	5
4	4	4
25	1	1
25	2	5
25	3	4
25	4	3

Probeer in tabel **Antwoord** eens een combinatie *VraagNr*, *EnqueteNr* opnieuw te gebruiken.

Sluit alle tabellen af. De waarden worden automatisch weggeschreven. Er hoeft hiervoor geen aparte opdracht gegeven te worden.

1.3.2. Het wijzigen van gegevens in een tabel

*Doel: Na afloop dient de waarde van **StudentId** van student 'Bongers' in S103103 veranderd te zijn.*

Er is een fout gemaakt bij het invoeren. Student Bongers heeft namelijk niet identiteitsnummer S113113 maar S103103.

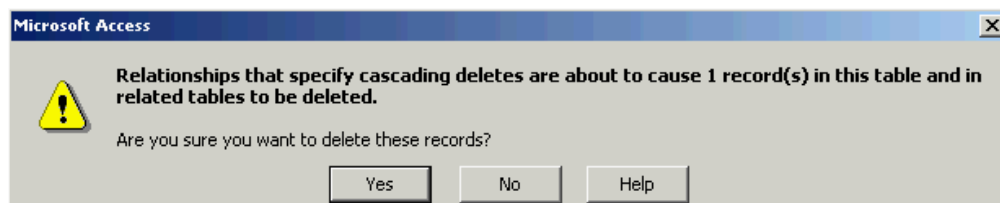
- Open de tabel **Student**.
- Ga naar het *StudentId*-veld van student Bongers.
- Verander de waarde in S103103.
- Sluit het venster.
- Controleer of in de tabel **Enquete** het identiteitsnummer dat eerst de waarde S113113 had, ook veranderd is in S103103. Dit dient namelijk gebeurd te zijn vanwege de Referentiële integriteit en het aanzetten van de 'Cascade Update Related Fields' optie.

1.3.3. Het verwijderen van gegevens uit een tabel

*Doel: Na afloop is de student Reinders uit de tabel **Student** verwijderd.*

Student Reinders heeft besloten om zich uit te schrijven bij de universiteit. Ten gevolge hiervan wordt hij uit de database verwijderd.

- Open de tabel **Student**.
- Selecteer het gehele record van student Reinders.
- Druk nu op de **Menu**-muisknop en selecteer uit het menu dat verschijnt **Cut** of **Delete Record** of druk op de <Delete>-toets. De volgende boodschap verschijnt op het scherm:



- Druk op **Yes**.

- Controleer of in tabellen **Enquete** en **Antwoord** de enquête van de student met identiteitsnummer S100100 verwijderd is. Dit dient namelijk gebeurd te zijn vanwege de Referentiële integriteit en het aanzetten van de 'Cascade Delete Related Records' optie.

1.4. Het formuleren van query's

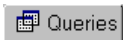
*Doel: Na afloop moeten er een query gebaseerd op een tabel, een query op meerdere tabellen, een **Make-Table** query en een **Update** query gemaakt zijn.*

Voor het opvragen van informatie uit een database, maar ook voor het toevoegen en wijzigen van informatie, wordt veelal gebruik gemaakt van *query's*, vragen aan de Database.

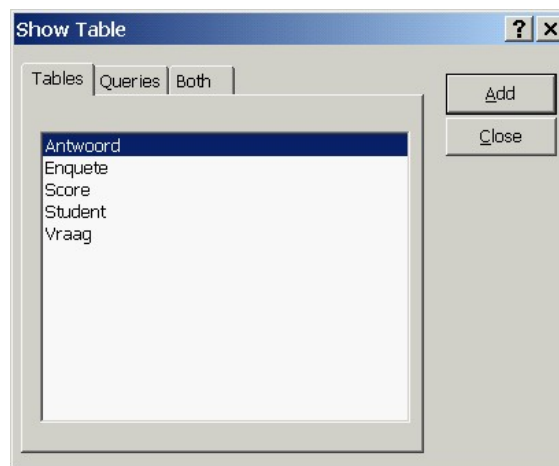
1.4.1. Het formuleren van een query op één tabel

*Doel: Na afloop moet de query **qryVraag25** zijn geformuleerd op tabel **Antwoord** die alle antwoorden op een bepaalde vraag (25) oplevert.*

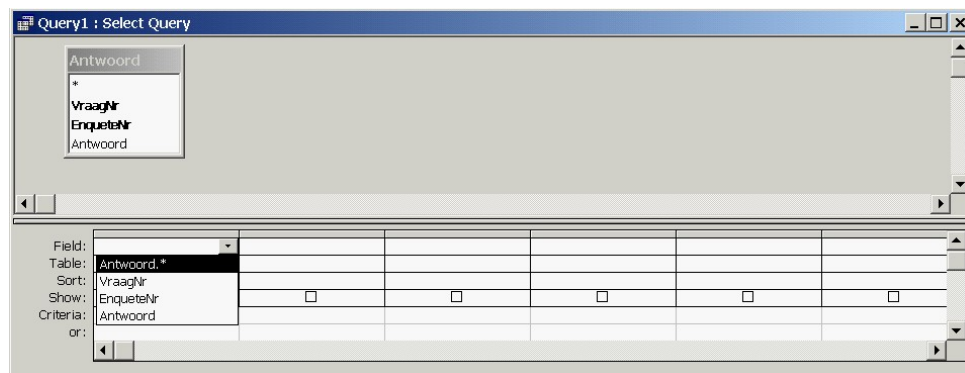
We willen een query gaan opstellen die alle records uit de tabel **Antwoord** haalt waarvan de waarde van *VraagNr* gelijk is aan 25.



- Druk met de cursor op de query-knop van het database-venster. De query object-groep wordt geopend.
- Druk vervolgens op **New**.
- Selecteer 'Design View' en klik op **OK**. Het volgende scherm verschijnt:



- Selecteer de tabel **Antwoord**.
- Klik op **Add** en vervolgens op **Close**.
- Klik op de afrolknop die zich in de eerste kolom van het rooster bevindt. Het volgende venster bevindt zich nu op je scherm:



- Selecteer in het afrolmenu **Antwoord**.* Hiermee geven we aan dat we alle velden van de tabel **Antwoord** willen gebruiken in de query. In de kolom bevindt zich nu ook een vinkje. Daarmee wordt aangegeven dat de velden die in deze kolom staan ook in het resultaat van de query komen te staan. Dat willen we ook zo houden, immers we willen van een record uit de tabel **Antwoord** alle velden hebben en niet slechts enkele velden.
- ✓ Plaats in de volgende kolom op de eerste regel het veld *VraagNr*.
- ✓ Klik op het vinkje op de 'Show'-regel in deze kolom. Het vinkje verdwijnt. Hierdoor wordt deze kolom niet getoond in het eindresultaat (anders zou *VraagNr* twee keer verschijnen). Deze kolom wordt alleen gebruikt om goede records uit de tabel **Antwoord** te filteren.
- Vul in de 'Criteria'-regel 25 in.
- Plaats in de volgende kolom op de eerste regel het veld *EnqueteNr* en klik het bijbehorende vinkje weg.
- Klik in de tweede kolom op de regel 'Sort'. Er verschijnt een afrolknop.
- Selecteer in het bijbehorende afrolmenu de optie 'Ascending' om oplopend te sorteren.
- ! Klik op de knop 'Run' uit de 'Toolbar' om de query uit te voeren.

Op het scherm verschijnen nu alle antwoorden op vraag 25 in oplopende volgorde van *EnqueteNr*.



Druk op de 'Save' icoon en schrijf de query weg als **qryVraag25**.



- Druk op de 'Design View' icoon. Het query opmaakscherm komt weer tevoorschijn.
- Zet het weergave vinkje in de tweede kolom weer aan en voer de query uit. Vergelijk het resultaat.
- Ga weer terug naar het query opmaakscherm en vervang **Ascending** door **Descending**.
- Voer de query opnieuw uit en vergelijk het resultaat.
- Sluit het venster af.

Opmerking:

Je kunt het queryschema ook invullen door de veldnamen vanuit het kader bovenin het venster naar de juiste cel te slepen. Wanneer je dubbelklikt op een veldnaam wordt deze automatisch in de eerste lege kolom van het rooster geplaatst.

1.4.2. Het aanmaken van een query op meerdere tabellen

*Doel: Na afloop moet de query **qryScore** zijn geformuleerd op de tabellen **Vraag** en **Antwoord**, die voor iedere vraag laat zien hoe de studenten gemiddeld verschillende aspecten van OGO1.1 beoordeeld hebben.*

In het resultaat van de query willen we *VraagNr*, *Categorie*, *Vraag* en *Antwoord* zien. We maken ons er niet druk over dat het gemiddelde van bepaalde vragen (zoals vraag 1) geen betekenis heeft.

- Open een nieuwe query
- Selecteer met 'Add' nu de tabellen **Vraag** en **Antwoord**.
- Het query opmaakscherm verschijnt en in het bovenste gedeelte staan de twee tabellen en de onderlinge relatie.
- Vul het rooster als volgt in:

Field:	VraagNr	Categorie	Vraag	Antwoord
Table:	Vraag	Vraag	Vraag	Antwoord
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Dit zijn de velden die we in het resultaat van deze query terug willen zien.
N.B. VraagNr komt zowel in de Vraag als in de Antwoord tabel voor. Het ligt meer voor de hand VraagNr uit de Vraag tabel te kiezen, maakt voor het resultaat is dit niet van belang.



- Druk nu op de 'Totals' knop uit de 'Toolbar'.
- Selecteer in de 'Total'-regel van het veld *Antwoord* uit het afrolmenu: 'Avg'. 'Avg' staat voor 'average' wat gemiddelde betekent. De oude waarde, 'Group By', wordt hierdoor vervangen.

Met deze opdracht wordt aangegeven, dat er in het resultaat van de query in plaats van alle scores gegroepeerd per enquetevraag, wat wordt aangegeven door 'Group by', alleen het gemiddelde van alle scores voor iedere vraag komt te staan.




- Voer de query uit.
- Verander de opmaak van de uitvoer zodat de vragen leesbaar zijn, b.v. door de hoogte van de rijen aan te passen. Wanneer je de cursor aan de linkerkant van het uitvoerresultaat in het grijze gebied tussen de twee rijen plaatst, verandert de cursor van vorm en kun je de rij hoger maken.
- Schrijf de query weg onder de naam **qryScore**.

1.4.3. Het veranderen van een Select query in een Make-Table query

*Doel: Na afloop is het resultaat van de query **qryScore** weggeschreven naar de tabel **Score**.*

We willen de uitkomsten van **qryScore** expliciet wegschrijven als een tabel.



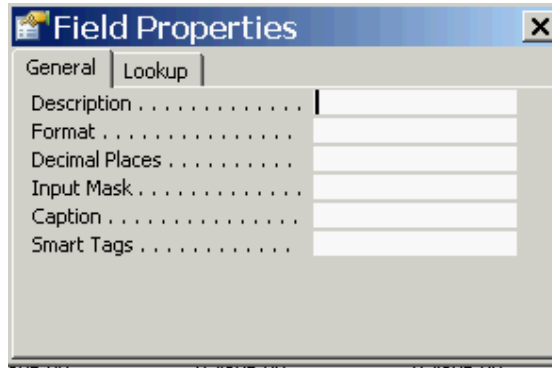
- Open **qryScore**, wanneer deze na het vorige onderdeel is afgesloten.
- Ga naar het query opmaakscherm met de 'Design View' knop.
- Druk op het driehoekje naast de 'Query Type'-knop in de 'Toolbar'.
- Selecteer uit het verschenen menu 'Make-Table Query...': 
- Geef als naam voor de tabel op: **Score** en sluit het venster af door op **OK** te drukken.
- Voer de query uit en de uitkomsten worden na een waarschuwing naar een tabel weggeschreven.
- Sluit het venster.
- Open de tabel **Score**. Wat valt je op?

1.4.4. Het wijzigen van het uitvoerformaat van een bestaande query

*Doel: Na afloop hebben alle gemiddelden van de query **qryScore** precies 1 decimaal.*

- Selecteer **qryScore** en druk op **Design**.
- Druk weer op het driehoekje naast de 'Query Type'-knop uit de 'Toolbar'. Omdat deze knop het type van de geselecteerde query toont, zie je hier nu de Make-Table icoon. De knop zit nog wel op dezelfde plaats. Stel het query-type nu in als 'Select Query'. De query zal nu bij uitvoering weer een lijst met resultaten op het scherm gaan zetten en deze niet meer naar een tabel wegschrijven.
- Druk op de **Menu**-muisknop, terwijl de muis zich in de kolom van *Antwoord* bevindt.
- Klik op 'Properties'. Het volgende scherm verschijnt:






- Selecteer uit het afrolmenu bij 'Format' 'Fixed' en vul bij 'Decimal Places' 1 in.
- Voer de query uit en vergelijk het resultaat met het vorige resultaat.
- Sluit het venster af.

1.4.5. **Het opstellen van een query die een tabel wijzigt**



*Doel: Na afloop moet de query **qryTelBij** zijn gedefinieerd op de tabel **Antwoord** die voor student S103103 bij de score van alle antwoorden een punt bijtelt.*

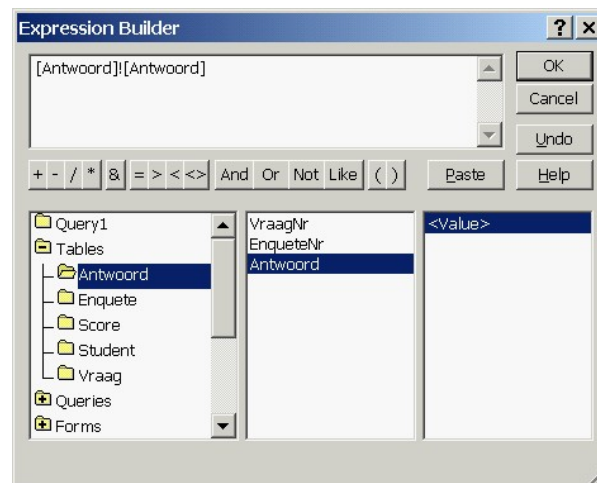
Student S103103 had als goede informaticastudent aangenomen dat de antwoorden vanaf nul werden genummerd. Al zijn antwoorden moeten dus met 1 punt verhoogd worden.



- Open een nieuwe query met tabellen **Antwoord** en **Enquete**.
- Druk weer op het driehoekje naast 'Query Type', maar selecteer nu 'Update Query'
 -  Update Query
- Vul het queryrooster op de volgende manier in:

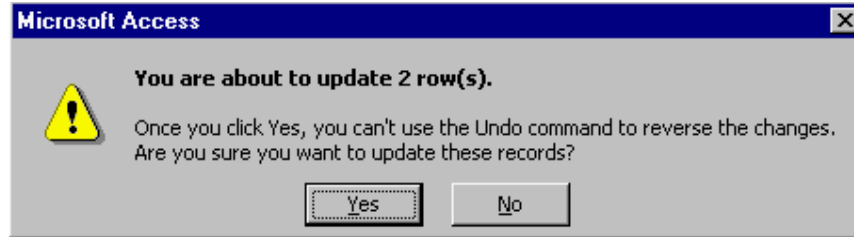
Field:	Antwoord	StudentId
Table:	Antwoord	Enquete
Update To:		
Criteria:		"S103103"

- Ga op het 'Update To' veld staan in de eerste kolom en druk op de **Menu**-muisknop.
- Selecteer uit het menu **Build....**
- Dubbelklik op  Tables en druk vervolgens op  Antwoord .
- Klik vervolgens dubbel op *Antwoord* in het middelste veld. Het venster ziet er nu als volgt uit:





- Typ nu in +1.
- Druk nu op **OK**.
- Voer de query uit.
- De volgende waarschuwing kan op het scherm verschijnen. In dit geval dient u op **Yes** te drukken:



- Schrijf de query weg als **qryTelBij** en sluit het venster.
- Controleer in de **Antwoord** tabel of de enquêteresultaten van student S103103 aangepast zijn.

Deze update-query twee keer uitvoeren heeft natuurlijk het effect dat de score van alle antwoorden van student S103103 met twee punten worden opgehoogd. Probeer dit uit. Wijzig daarna **qryTelBij** zodat de teveel toegekende punten weer gecompenseerd worden.

1.5. Formulieren

*Doel: Na afloop dienen de formulieren **frmStudent**, **frmScore** en **frmSamen** te zijn aangemaakt.*

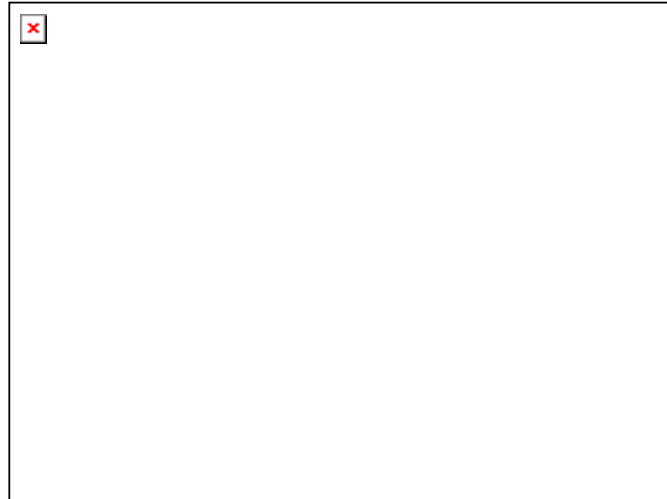
Een formulier wordt gebruikt om in een tabel gegevens toe te voegen, te verwijderen en te veranderen. Om deze acties uit te voeren hebben we in principe geen formulieren nodig, zoals we tot nu toe gezien hebben, maar formulieren maken deze acties makkelijker en overzichtelijker. Dit laatste is belangrijk voor databasegebruikers die geen Informatica-achtergrond hebben.

1.5.1. Het aanmaken van een nieuw formulier op één tabel

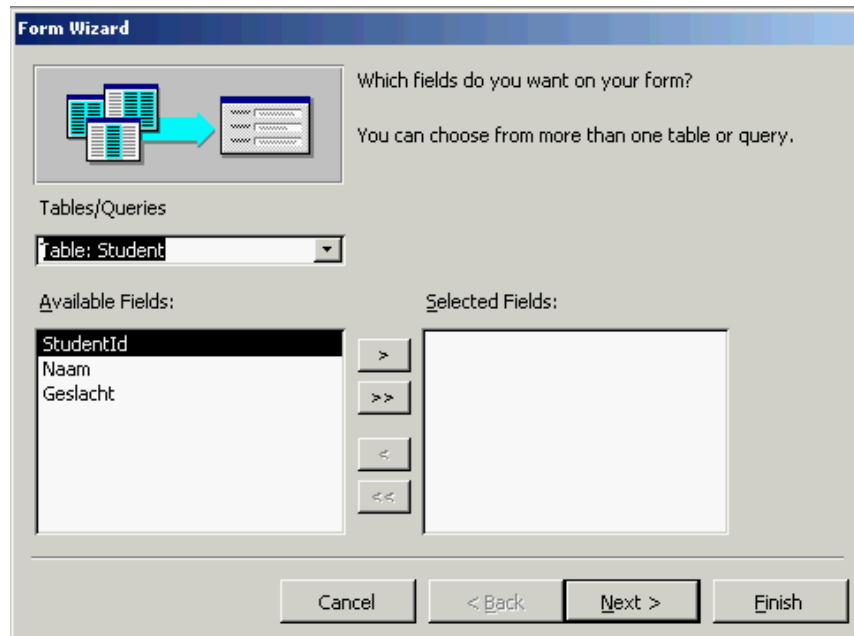
*Doel: Na afloop moet het formulier **frmStudent** zijn gedefinieerd op tabel **Student** waarin de gegevens per student kunnen worden gewijzigd.*



- Druk op de Forms-knop van het database-venster. De Forms object-groep wordt geopend.
- Druk vervolgens op **New**. Het volgende venster verschijnt:

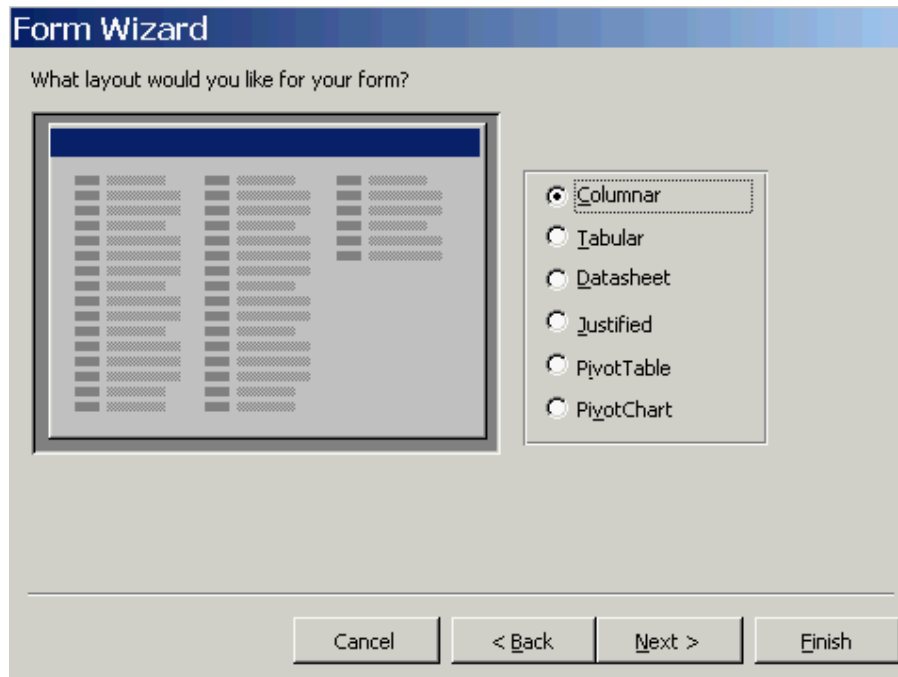


- Selecteer de 'Form Wizard' en druk op de afrolknop die in het venster zit en selecteer hieruit 'Student'. (In plaats van deze twee stappen was het ook mogelijk geweest om te dubbelklikken op  Create form by using wizard)
- Druk op **OK**.
- Het volgende scherm verschijnt:

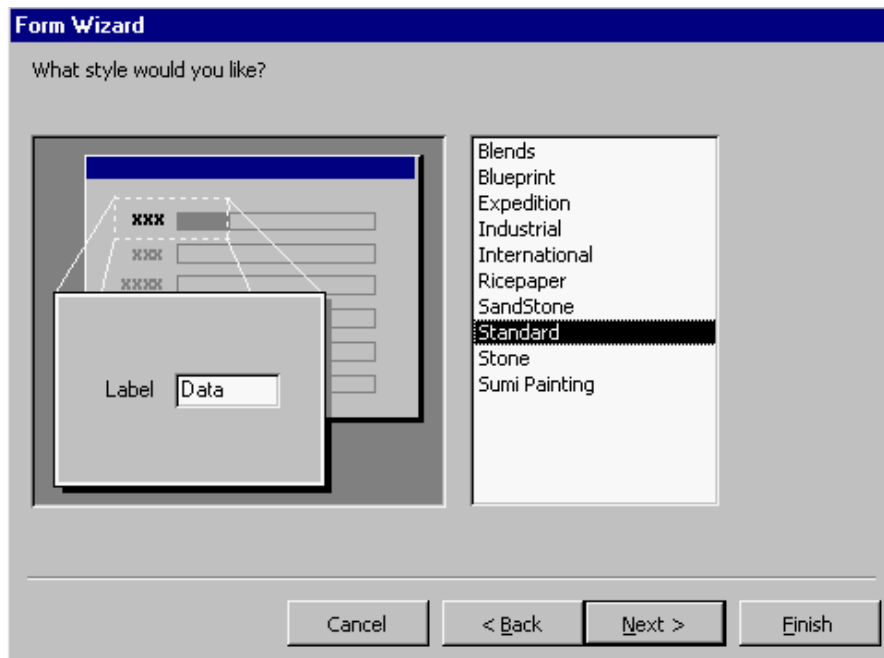


>>

- In dit scherm kunnen we aangeven welke velden we in het formulier willen hebben. In ons geval willen we alle velden hebben. Daarvoor dient op de knop met de twee 'groter dan' symbolen gedrukt te worden. Doe dit. Wil je een beperkt aantal velden hebben dan moet je ieder veld dat je nodig hebt eerst in het linkervenster selecteren en daarna met de knop met het enkele 'groter dan' teken overhalen naar het rechter venster.
- Druk op **Next >**.
- Het volgende scherm verschijnt:



- In dit scherm kunnen we aangegeven hoe de opmaak van het formulier moet zijn. Selecteer 'Columnar' en druk op **Next >**.
- Het volgende scherm ziet er zo uit:



- Er zijn verschillende mogelijkheden voor de achtergrond. Selecteer enkele en bekijk de verschillen. Selecteer 'Standard' en druk op **Next >**.
- Het volgende scherm verschijnt:

- In dit scherm kun je de laatste informatie invoeren over het formulier. In de bovenste invoerregel dien je het formulier een titel te geven. Standaard staat hier de naam van de tabel die je gebruikt hebt. In dit voorbeeld is dit 'Student'. Verander deze naam niet. Het scherm geeft nog meer mogelijkheden maar die zijn voor ons niet van belang. Druk op **Finish**.

We zien nu een record uit onze tabel **Student** in het formulier verschijnen. Met behulp van de navigatieknoppen kunnen we andere records zichtbaar maken.

- Sluit het formulier af.
- De wizard geeft het formulier standaard dezelfde naam als de gekozen titel. Wijzig de naam van het formulier in het Forms-venster in **frmStudent**. Merk op, dat hierdoor de formuliertitel "Student" niet veranderd wordt.

1.5.2. Het aanmaken van een nieuw formulier op een query

*Doel: Na afloop moet het formulier **frmScore** zijn aangemaakt op basis van de query **qryScore** en moet duidelijk zijn dat deze query bij het openen van het formulier uitgevoerd wordt.*

Maak een formulier aan op **qryScore**. Dit gaat op dezelfde wijze als het aanmaken van een formulier op een tabel, alleen moet nu een query geselecteerd worden als basisobject. Geef het formulier de titel **Scores** en sluit het venster af. Merk op dat de records die getoond worden de actuele resultaten zijn van **qryScore**.

- Geeft het formulier nu de naam **frmScore**.
- Wijzig de tabel **Antwoord** door van een vraag alle antwoorden met 1 punt te verlagen.
- Open **frmScore**.

Het formulier **frmScore** toont nu het nieuwe gemiddelde voor de vraag. Merk op dat we de query nergens expliciet hebben uitgevoerd. Access heeft deze query uitgevoerd op het moment dat **frmScore** werd geopend.

- Sluit het formulier.

1.5.3. Het aanmaken van een nieuw formulier op meerdere tabellen

*Doel: Na afloop moet het formulier **frmSamen** zijn gedefinieerd op de tabellen **Enquete** en **Student** waarin per student gegevens over studenten en hun enquêtes kunnen worden ingevuld en getoond.*

- Open een nieuw formulier (Druk op **New**).
- Selecteer weer de 'Form Wizard' maar selecteer geen tabel.
- Druk op **OK**.



- Selecteer in het venster waar je de tabellen en de queries moet opgeven de tabel **Student**.
- Druk op de button met de twee 'groter dan' symbolen om alle velden over te halen.
- Selecteer nu de tabel **Enquete**.
- Breng nu met behulp van de knop met het 'groter dan'-teken de velden *EnqueteNr* en *Datum* naar de andere kant.
- Druk op **Next >**. Doordat we in het 'relations' model de relaties tussen de verschillende tabellen hebben vastgelegd weet de 'Form Wizard' de samenhang tussen de tabellen.
- Het nieuwe venster vraagt ons hoe we records willen bekijken. Selecteer 'by Student' en 'Form with subforms' en druk op **Next >**.
- Selecteer in het volgende venster 'Datashet' en druk op **Next >**.
- Selecteer de 'Standard' stijl en druk op **Next >**.
- Geef het formulier de titel **Samen** en het "subform" de titel **Samen Subform**
- Druk op **Finish**

Het formulier verschijnt. Met behulp van de navigatietoetsen kan door de verschillende records worden gelopen.


- Sluit het venster af.
- Hernoem Samen tot frmSamen en Samen Subform tot frmSamen Subform.

1.5.4. Het wijzigen van tabellen door middel van een formulier

*Doel: Na afloop moeten d.m.v. het formulier **frmSamen** tabel **Student** en tabel **Enquete** gewijzigd zijn.*

- Open het formulier **frmSamen**.

We gaan een nieuwe student invoeren die als *StudentId* S108108 heeft en als naam 'Verhagen'. Daarnaast gaat deze student vandaag de enquête invullen.

-  Ga naar een nieuw (leeg) record met behulp van de navigatie-knoppen. *Let erop dat je 'buitenste' navigatieknoppen gebruikt.* De binnenste navigatieknoppen zijn om door de enqueteresultaten te lopen.
- Vul de velden op dezelfde manier in zoals dat in de onderstaande figuur gedaan is (vul wel de juiste datum in):

Samen

StudentId: S108108

Naam: Verhagen

Geslacht: M

EnqueteNr	Datum
5	20-3-2005

Record: 2 of 2

Record: 4 of 4

- Sluit het venster af.
- Controleer of in de tabel **Student** de nieuwe student is toegevoegd en controleer of de enquête voor deze student ook in de tabel **Enquete** is toegevoegd.

1.6. Rapporten

*Doel: Na afloop moeten de rapporten **rptStudent**, **rptSamen** en **rptScore** zijn aangemaakt.*

Wanneer men de inhoud van een database bijvoorbeeld aan derden wil tonen, dan gebeurt dat meestal via een afdruk op papier. Nu kan men een tabel afdrukken, maar vaak wil men een opmaak hebben die er netter en overzichtelijker uitziet. Hiervoor worden rapporten gebruikt.

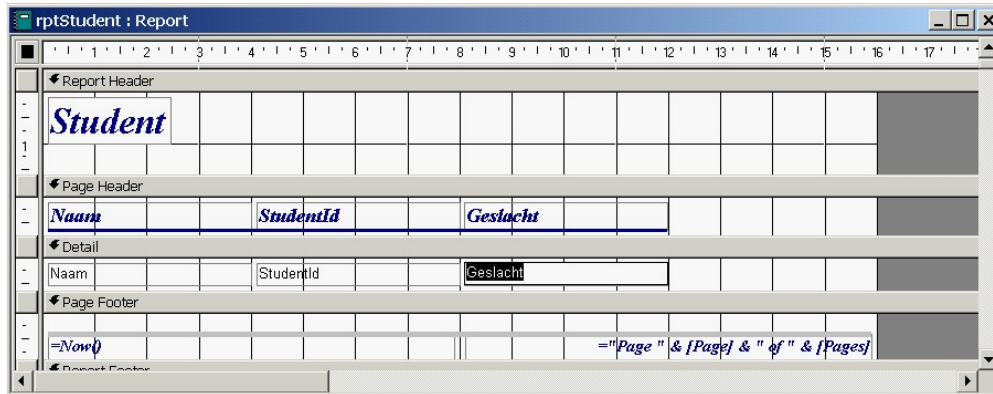
- Het maken en wijzigen van een rapport werkt op nagenoeg dezelfde manier als bij de formulieren.
- Druk op de Reports-knop in het databasevenster om de Reports-groep te openen.

Het verdient de aanbeveling de database nu eerst op te slaan om je resultaten veilig te stellen. Om een rapport in preview modus te kunnen bekijken, heeft Access de details van een printer nodig. Heb je geen enkele printer geïnstalleerd, dan zal het rapport niet in preview modus worden geopend. Heb je een netwerk printer geïnstalleerd, dan kan het zijn dat Access deze printer op het netwerk gaat zoeken. Dit kan tot 5 minuten duren, dus als dit optreedt is het verstandig Access te stoppen. In beide gevallen is de oplossing om een *willekeurige lokale* printer als **default** printer te installeren.

1.6.1. Het aanmaken van een nieuw rapport voor één tabel

*Doel: Na afloop moet rapport **rptStudent** gemaakt zijn voor de tabel **Student**.*

- Maak rapport **rptStudent**, analoog aan het formulier **frmStudent**, met de Report Wizard. Zorg dat de velden in de volgorde *Naam*, *StudentId* en *Geslacht* staan (dit moet direct bij de selectie van de velden gebeuren) en dat de namen alfabetisch gesorteerd worden.
- Open (na hernoemen van het rapport) het rapport **rptStudent** in de Design-view:



1.6.2. Het aanmaken van een nieuw rapport voor meerdere tabellen

*Doel: Na afloop moet het rapport **rptSamen** zijn gedefinieerd op de tabel **Student** en de tabel **Enquete**, waarin de enquêtes die zijn gedefinieerd getoond worden.*

- Maak het rapport **rptSamen** aan, analoog aan het formulier **frmSamen**.

Merk op dat er geen 'sub rapport' geproduceerd wordt. Merk ook op dat Access op *StudentId* sorteert. We zullen in een volgend hoofdstuk zien hoe dit veranderd kan worden.

1.6.3. Het aanmaken van een nieuw rapport op een query in plaats van een tabel

*Doel: Na afloop moet het rapport **rptScore** zijn aangemaakt op de query **qryScore**.*

- Maak het rapport **rptScore** aan, analoog aan het formulier **frmScore**. Sorteert op *VraagNr*.

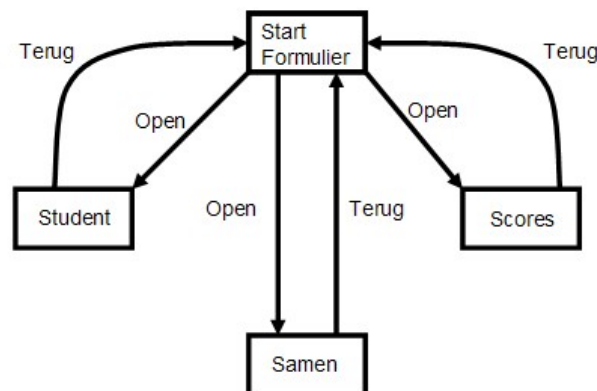
Opmerking: Getallen ('Numbers') worden tegen de rechterkant van een kolom geplaatst en de tekst tegen de linkerkant. Dit komt de opmaak van dit rapport niet ten goede. De vragen en ook de headers zijn niet geheel zichtbaar. In een volgend hoofdstuk wordt uitgelegd hoe dit veranderd kan worden.

2. Inleiding Databasetoepassingen

2.1. *Introductie*

Doel: een korte vooruitblik geven op databasetoepassingen

In het voorgaande hoofdstuk hebben we kennis gemaakt met de databaseobjecten van het type tabel, query, formulier en rapport en gezien hoe we deze konden realiseren binnen MS-Access. Dit leverde een database op, die geschikt is voor incidenteel gebruik, zoals het uitproberen van query's of het bijhouden van een eenvoudige administratie. In de praktijk is dat vaak niet voldoende. Bij een database, die intensief gebruikt wordt, is het belangrijk dat gegevens met een minimum aan handelingen in de goede volgorde ingevoerd kunnen worden. Dit vereist bijvoorbeeld dat een formulier "met één druk op de knop" vanuit een openstaand formulier kan worden geopend en van de juiste startinformatie voorzien kan worden. Een dergelijke functionaliteit spaart tijd en voorkomt veel fouten ten gevolge van verkeerd overnemen van gegevens of het verder werken met een verkeerd formulier. Tot nu toe hebben we alleen losse formulieren en dito rapporten gezien, die met behulp van een Wizard zijn gegenereerd op basis van een of meer tabellen of query's. Dit is echter al een goed begin. We zullen in dit hoofdstuk kennismaken met technieken voor het verder uitbouwen van onze collectie objecten tot een geïntegreerde databasetoepassing. Na afloop van het hoofdstuk zijn de verschillende formulieren als volgt met elkaar verbonden:



In dit diagram geven de rechthoeken de formulieren weer, de pijlen de overgangen tussen de formulieren en het label naast de pijl de soort van overgang.

2.2. *Een eenvoudig startformulier*

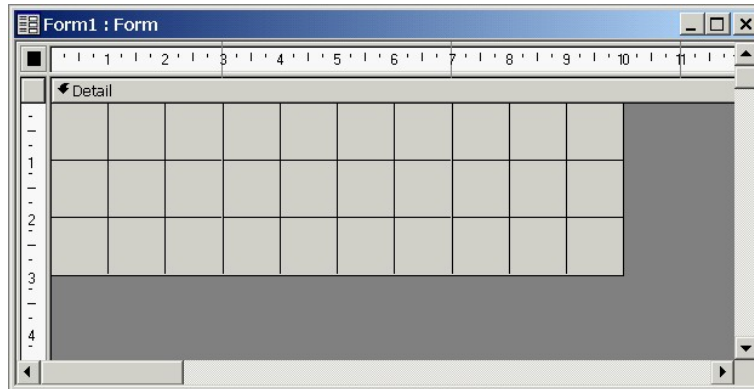
Doel: Na afloop is het formulier frmStartFormulier gemaakt dat de gebruiker toegang biedt tot de al aanwezige formulieren.

We gaan in dit hoofdstuk een eenvoudig keuzemenu maken op basis van een formulier. Dit menuformulier zal het eerste formulier worden, dat de gebruiker bij het starten van de databasetoepassing te zien krijgt. Het zal een keuze bieden uit alle beschikbare formulieren. Dit bevordert de overzichtelijkheid van de toepassing.

We hebben een nieuw formulier nodig, dat niet gebaseerd is op een tabel. Op dit formulier zullen we knoppen plaatsen, die elk een ander formulier starten.



- Open een nieuw formulier door in het Forms-venster op **New** te drukken en vervolgens voor **Design View** te kiezen. Het volgende scherm verschijnt:



Het ruitjesvel onder het kopje **Detail** wordt het **Design Grid** genoemd.

- Sla het formulier op als **frmStartFormulier**.

2.2.1. Toevoegen van een knop aan een formulier


*Doel: na afloop is een opdrachtknop geplaatst op het formulier **frmStartFormulier**.*

Om objecten, zoals knoppen en labels op het formulier te plaatsen hebben we de **toolbox** (werkset) nodig. De toolbox is de gereedschapskist van de ontwerper en bevat een aantal gereedschappen, **controls** (besturingselementen) genaamd, die voorgesteld worden door knoppen. We kunnen een control op het formulier plaatsen door de bijbehorende knop in de toolbox in te drukken en met de cursor een positie op het formulier te selecteren. Het bijbehorende besturingselement wordt nu op het formulier geplaatst en is beschikbaar voor verdere bewerking. We volstaan hier met het bespreken van de elementen die we gebruiken. Een korte uitleg over de overige controls is te vinden in Bijlage A.



- Druk op de **toolbox** icoon op de werkbalk, wanneer het werkset-venster nog niet te zien is.



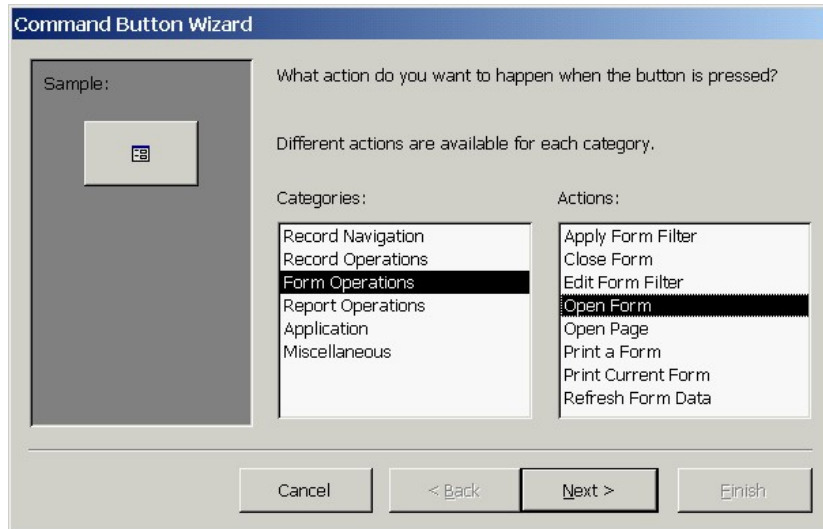
De twee knoppen boven de horizontale streep dienen ingedrukt te zijn. De rechter daarvan , de **Control Wizard** knop, activeert een Wizard voor sommige van de knoppen er onder. De linker knop, de **object selector**, maakt het mogelijk om objecten op het formulier te selecteren en te wijzigen.

We gaan nu een opdrachtknop aan **frmStartFormulier** toevoegen voor het openen van **frmStudent** (zorg dat de Control Wizard knop is ingedrukt):



- Druk de knop **Command Button** (Opdrachtknop) in de toolbox in.
- Wanneer de muis-cursor boven het formulier gezet wordt, verandert de vorm van de cursor in een + teken, met daaronder de icoon voor een knop. Wijs een geschikte plaats op het

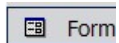
formulier aan en druk op de **select**-muisknop. De **Command Button Wizard** wordt nu gestart.



- Blader even door de categorieën voor een indruk van de mogelijkheden.
- Selecteer **Form Operations** uit de lijst met categorieën.
- Selecteer daarna **Open Form** uit de lijst met acties en druk op **Next**.
- Kies **frmStudent** uit de lijst met formulieren en druk op **Next**.
- Kies ervoor om het formulier te openen en alle records weer te geven. Druk op **Next**.
- Kies ervoor om de tekst "Open" op de knop te zetten. Druk op **Next**.
- Geef de knop de naam **cmdStudentOpen** en druk op **Finish**.
- Geef de knop een geschikte grootte.



De afmetingen van de knop kunnen in de Design View gewijzigd worden, wanneer de knop geselecteerd is. Als met de Select-knop in de knop wordt geklikt, verschijnen kleine vierkantjes op de hoeken en de middens van de zijden van de knop. Door de cursor op deze vierkantjes te plaatsen en vervolgens te klikken en te slepen kun je de afmetingen wijzigen.



- Probeer de knop uit in de formulierweergave.

Merk op, dat we de knop een naam gegeven hebben, die de functie van de knop aanduidt. Dit is straks nuttig wanneer we in de programmacode de functie van de knop willen aanpassen of naar de knop willen verwijzen.

De **Command Button Wizard** biedt een eenvoudige manier om veel voorkomende functionaliteit aan een formulier toe te voegen, die geactiveerd kan worden door het drukken op een knop. Sommige van deze functies, zoals die uit de categorie **Record Operations**, veronderstellen dat het formulier gebaseerd is op een gegevensbron, zoals een tabel of een query. Andere (**Open Form**, **Close Application**) zijn bewerkingen op formulier- of toepassingsniveau.

2.2.2. Toevoegen van een label aan een formulier

*Doel: na afloop is een label met verklarende tekst naast de knop op **frmStartFormulier** geplaatst.*

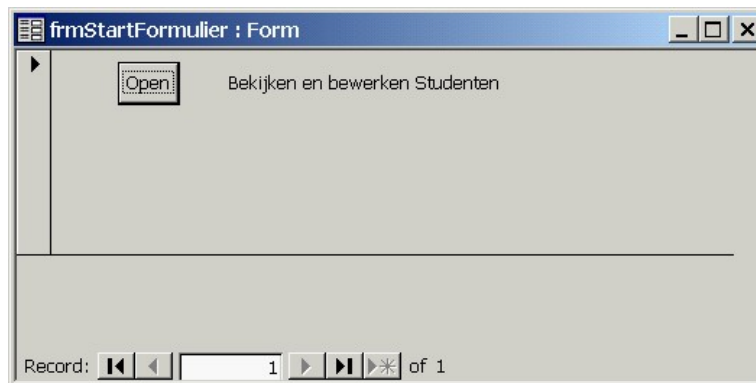
Omdat het opschrift **Open** op alle formulieren van toepassing kan zijn, zetten we tekst naast de knop, die een aanduiding geeft van de functie van de knop.



- Druk de knop **Label** in in de toolbox.
- Wanneer de cursor boven het formulier komt, verandert deze in een + teken, nu met de letter A eronder. Zet het plusteken op de plaats op het formulier waar de linkerbovenhoek van het label moet komen en sleep de muiscursor naar de plaats voor de rechteronderhoek.
- Laat de muisknop los. Voeg de tekst “Bekijken en bewerken Studenten” toe.

De tekst die in het label staat kan gewijzigd worden door in die tekst te klikken. Daarna kun je over de tekst heentikken. Om uit deze modus te komen: klik op een willekeurige plek buiten het label op het design grid.

- Het resultaat ziet er in de Form View als volgt uit:



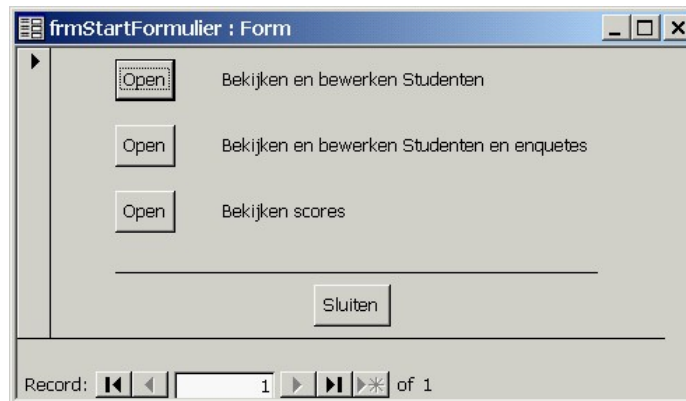
Merk op, dat we de label tekst ook als opschrift voor de knop hadden kunnen gebruiken. Het nadeel hiervan is, dat we dan brede knoppen (balken) nodig hebben, die mogelijk ook nog van verschillende grootte zijn. Nu kunnen we alle knoppen dezelfde grootte geven.

- Breid **frmStartFormulier** uit met knoppen voor het starten van **frmSamen** en **frmScore**, een knop voor het sluiten van de toepassing (uit de categorie Toepassing) en een horizontale scheidingslijn.

Het kan zijn dat er op het formulier geen ruimte is om een extra control te plaatsen. We moeten dan het zichtbare deel van het formulier vergroten. Plaats in dat geval de cursor op de rand van het design grid¹. De cursor verandert van vorm (een lijn met pijlen naar boven en naar beneden). Zodra de cursor van vorm is veranderd, kan door op de Select-knop te klikken en te slepen het design grid vergroot worden. Door de cursor rechts op de rand van het design grid te plaatsen kan het formulier breder worden gemaakt.

- Dit levert het volgende resultaat op:

¹ Er kan een **Form header** en **footer** aanwezig zijn. In dat geval krijg je hetzelfde effect als je de cursor aan de bovenkant van de balk onderaan positioneert. Header en footer zijn te verwijderen en toe te voegen door de **detail** sectie rechts aan te klikken en header/footer te (de)selecteren.



- Probeer drie "Open"-knoppen uit. Waarom kunnen de scores niet gewijzigd worden?
- Probeer de "Sluiten"-knop uit. Wat is het verschil met het indrukken van de knop rechtsboven aan het formulier?

Er zijn nog een paar zaken in het bovenstaande formulier die aangepast moeten worden. Een formulier wordt standaard voorzien van een Record-navigatiebalk (onderaan het formulier) en een record-selector (de linkerkolom van het formulier), ook al is het formulier niet gebaseerd op een tabel of query. Dit zijn de standaardinstellingen van het formulier, die eenvoudig te veranderen zijn.

2.2.3. Instellen van eigenschappen: Property sheets

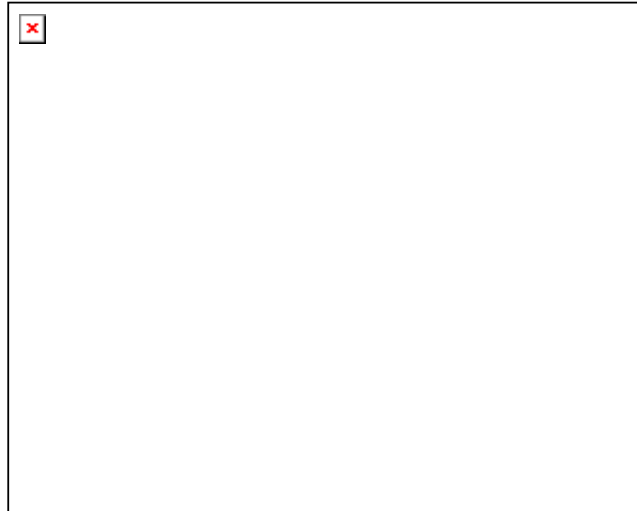
*Doel: na afloop zijn enige instellingen van **frmStartFormulier** aangepast.*

Het wijzigen van instellingen gebeurt in het **Property sheet** (eigenschappenblad) van het formulier. Alle besturingselementen en andere objecten, zoals formulieren en rapporten, hebben een eigenschappenblad. Dit blad is op te roepen, door met de menu-knop op de component te drukken. Dit kan het beste in de **Design View** gebeuren.

- Open frmStartFormulier in de Design View.
- Druk met de menuknop op verschillende componenten op het formulier om hun eigenschappenblad op te roepen. Selecteer **Proprieties (Eigenschappen)** uit het menu. Bekijk deze, maar verander niets.
- Druk, om de eigenschappen van het formulier op te roepen, met de menuknop op het zwarte vierkant in de linkerbovenhoek en selecteer weer properties:

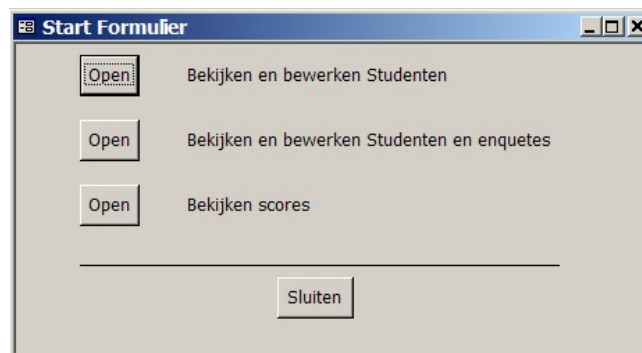


- Het volgende scherm verschijnt:



We zien vijf tabbladen in het venster, waarvan de eerste vier instellingen groeperen die te maken hebben met respectievelijk de opmaak van het formulier, de onderliggende gegevensbron, de interactiviteit van het formulier en met overige eigenschappen, zoals Naam. Het vijfde tabblad geeft alle eigenschappen samen weer. Meer uitleg over deze eigenschappen is te vinden in Bijlage B.

- Open het tabblad **Format**.
- Om de navigatiebalk op het formulier te verbergen, moet de betreffende eigenschap de waarde "No" krijgen. Selecteer de eigenschap **Navigation Buttons** en kies de waarde "No" uit het afrol-menu.
- Stel ook de eigenschappen **Record Selectors** en **Dividing Lines** in op "No".
- Geef de eigenschap **Caption** (Bijchrift) de waarde "Start Formulier". Deze tekst wordt in de titelbalk van het formulier geplaatst.
- Sluit de property sheets.
- Het formulier ziet er nu in de formulierweergave uit als volgt:



Opmerking: wanneer we binnen Access werken, kunnen de property sheets ook in de **Form View** opgeroepen worden. Dit maakt het werk van de applicatieontwikkelaar wat eenvoudiger.

Het startformulier biedt nu vier keuzes. Het is eenvoudig uit te breiden door knoppen toe te voegen. We kunnen bijvoorbeeld een tweede kolom knoppen toevoegen voor het openen van beschikbare rapporten.

Een meer geavanceerde versie van het menuformulier is het zogenaamde **Switchboard** (schakelpaneel), dat een onderdeel is van de voorbeeldtoepassingen, die met de **Database Wizard** gegenereerd worden (zie hoofdstuk 7).

Andere vormen van menu's, die we uit de Windows omgeving kennen, kunnen gerealiseerd worden met behulp van formulier specifieke menubalken of met behulp van de **Tab control** die in de **Toolbox** te vinden is. Het gebruik van tabbladen voor het organiseren van lijsten met items wordt goed geïllustreerd door de **Property sheets**.

2.3. Een eerste versie van de databasetoepassing

Doel: Na afloop is er een eerste versie van een toepassing die de gebruiker toegang biedt tot de aangemaakte formulieren en afschermt van de onderliggende database omgeving.

Een eindgebruiker van een databasetoepassing wil in het algemeen zijn werk kunnen doen, zonder geconfronteerd te worden met de achterliggende techniek. We kunnen dit bereiken door de juiste opties voor het opstarten van de database in te stellen.

2.3.1. Opstarten van een databasetoepassing

Doel: na afloop kan de OGO Enquête database zo ingesteld worden, dat alleen de vier formulieren beschikbaar zijn.

We gaan er nu voor zorgen, dat bij het starten van de database het startformulier als eerste (en enige) formulier getoond wordt.

- Maak eerst een reservekopie van het bestand **OGO Enquete.mdb**.
- Kies nu de optie "**Startup...**" uit het **Tools**-menu van de Access-werkbalk. Het volgende scherm verschijnt:



- Vul als `Application Title` "OGO Enquete" in.
- Kies als startpagina bij `Display Form/Page` **frmStartFormulier**. Dit formulier wordt nu automatisch gestart bij het open van de database.
- Haal alle vinkjes weg, behalve die bij Speciale Access-toetsen. Hierdoor blijft slechts een beperkt aantal menu-items beschikbaar en wordt het databasevenster verborgen.
- Druk op "OK", sla de veranderingen op en sluit Access af.
- Open **OGO Enquete.mdb**. Wat valt je op?
- Sluit de toepassing andermaal af
- Open **OGO Enquete.mdb**, terwijl je de <Shift>-toets ingedrukt houdt. De werkbalken zijn nu weer beschikbaar.

Omdat we nog enige eenvoudige aanpassingen willen doen om de toepassing bruikbaar te maken, herstellen we de Access-omgeving weer naar zijn oorspronkelijke staat.

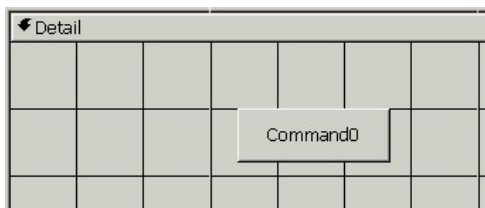
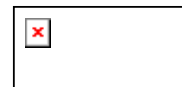
- Zet alle vinkjes weer aan in het opstartscherm en druk op **OK**.

2.3.2. *Intermezzo: de werking van knoppen*

Doel: na afloop is een knop gecreëerd en is de werking daarvan met VBA aangepast.

In grotere toepassingen zullen de menu's niet georganiseerd worden per formulier, zoals hier maar per taak. Een taak wordt meestal ondersteund door een opeenvolging van formulieren, waarbij op een van tevoren bepaalde manier tussen de formulieren gewisseld kan worden. Formulieren zullen dan ook meestal voorzien zijn van een aantal knoppen waarmee naar vervolgf formulieren gesprongen kan worden of naar eerdere formulieren teruggesprongen kan worden. We kennen deze gang van zaken van de Wizards die we al gezien hebben. Meestal wordt daarbij niet alleen het volgende formulier geopend, maar ook het actuele formulier gesloten. Het sluiten van een formulier en het openen ervan via een knop kunnen afzonderlijk met behulp van de **Command Button Wizard** gerealiseerd worden. Voor het uitvoeren van de combinatie is enig eenvoudig programmeerwerk nodig. De programmeertaal die binnen Access toepassingen gebruikt wordt is **VBA: Visual Basic for Applications**. Een overzicht van een aantal belangrijke mogelijkheden van deze taal wordt gegeven in de bijlage Bijlage C. We zullen nu eerst ingaan op de werking van een knop.

- Maak een nieuw formulier aan in de ontwerpweergave (Design View).
- Schakel in de toolbox de **Control Wizard** uit.
- Voeg nu een knop toe aan het formulier. Omdat de Command Button Wizard uit staat, verschijnt de knop zonder meer op het formulier:

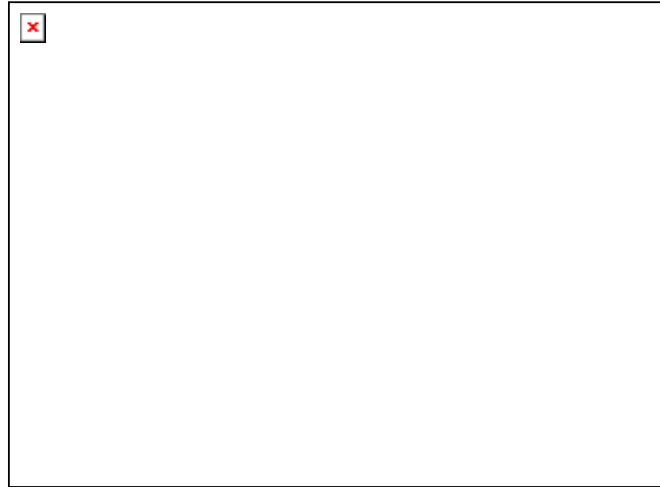


- Open de Property Sheets van de knop en geef deze het bijschrift (caption) "Aan" (tabblad **Format**)
- Geef de knop de naam "cmdWissel" (tabblad **Other**)
- Sluit het eigenschappenblad, ga naar de Form View en druk op de knop. Wat gebeurt er?

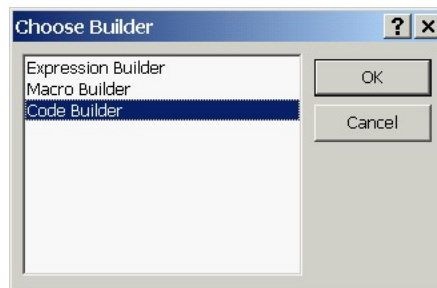
Het drukken op een knop heet binnen Access (en de meeste andere visuele programmeeromgevingen) een **event** (gebeurtenis). Andere gebeurtenissen, die binnen Access betekenis hebben zijn het indrukken van een toets, of het aanwijzen van een object binnen het toepassingsvenster. Gebeurtenissen als deze zijn het mechanisme waarmee de applicatie interessante activiteiten van de gebruiker kan signaleren. Daarnaast gebruikt Access intern gebeurtenissen om complexe processen, zoals het opstarten of afsluiten van een formulier op te delen in een aantal eenvoudiger stappen (zie Bijlage B).

Om de applicatie op een gebeurtenis te laten reageren, moet er programmacode met die gebeurtenis geassocieerd worden, die bij het optreden van het event geactiveerd kan worden. De **Command Button Wizard** plaatst daarom niet alleen een knop op het formulier en stelt een aantal eigenschappen daarvan in, maar voegt ook **VBA**-code toe aan het formulier. Deze code is ondergebracht in een **event procedure** of **event handler** (gebeurtenisprocedure). Is er geen procedure gedefinieerd voor een bepaalde gebeurtenis, dan gebeurt er niets, zoals daarnet. Het indrukken van **cmdWissel** leverde dus alleen een visueel effect op, meer niet. De event handlers voor de knop zijn te vinden via de property sheets van de knop:

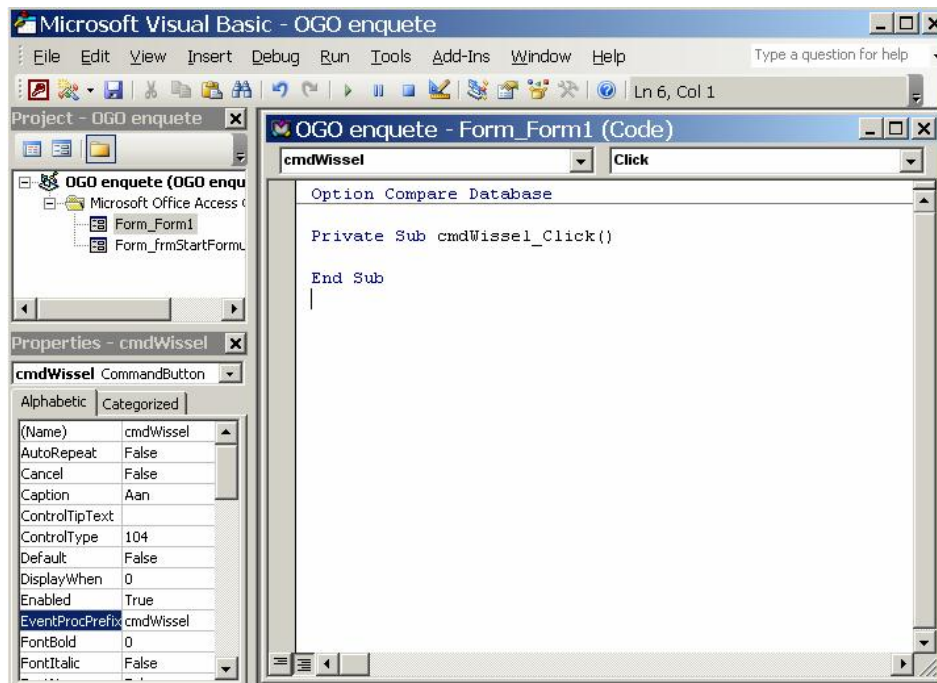
- Open de **Property Sheets** van **cmdWissel** en selecteer het **Event**-tabblad. Hier worden de mogelijke events getoond:



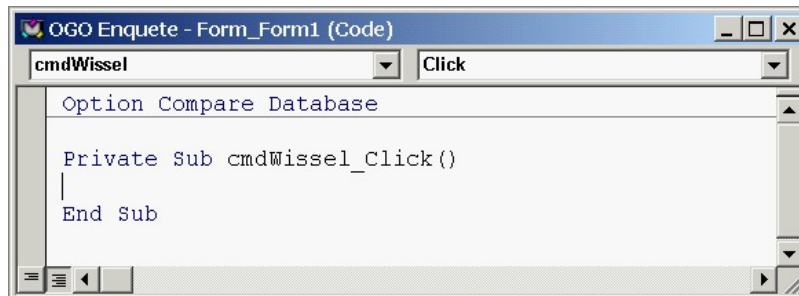
- Selecteer de **“On Click”** (bij klikken) en druk op de opbouwfunctieknop.
- Kies de **Code Builder** (opbouwfunctie voor programmacode) uit het lijstje:



Het codevenster gaat nu open:



Dit venster omvat links een **Project**-venster, waarin staat welke objecten, zoals formulieren, er beschikbaar zijn. Daaronder staat het **Property**-venster voor **cmdWissel**. Hierin staat de eigenschap **Eventprocedure** geselecteerd voor **cmdWissel**. Het grootste deel van het scherm wordt ingenomen door het codescherm voor Form1. Deze omvat één eventprocedure, genaamd **cmdWissel_Click**:



De eventprocedure is standaard een subroutine, wat aangegeven wordt door de **Private Sub** en **End Sub** sleutelwoorden. VBA voegt deze code automatisch toe aan het formulier wanneer de opbouwfunctie gekozen wordt. Wanneer in de afrolvensters een ander object en een ander event gekozen wordt, wordt daarvoor automatisch eenzelfde skelet aangemaakt en getoond.

De naam van een eventprocedure wordt gevormd door achter de naam van het object (knop, tekstvak, formulier) de naam van het event te plakken: **Objectnaam_Eventnaam()**: geen "punt" maar een "onderstreepje" tussen de twee delen, want het betreft hier een naam en niet een aanroep. De event handler voor het dubbelklik-event van de knop heet dus: `cmdWissel_DbIClick()`.

- Voeg nu de volgende code toe aan de procedure:

```
' Wissel tussen de opschriften "Aan" en "Uit".
If cmdWissel.Caption = "Aan" Then
    cmdWissel.Caption = "Uit"
Else
    cmdWissel.Caption = "Aan"
End If
```

In deze code gebeurt het volgende. Er wordt getest of de conditie `cmdWissel.Caption = "Aan"` waar is. Dit is het geval wanneer de knop **cmdWissel** op het moment van indrukken het opschrift "Aan" had. In dit geval wordt de opdracht achter het sleutelwoord **Then** uitgevoerd: het opschrift krijgt de waarde "Uit". Vermeldt het opschrift een andere tekst (de conditie is onwaar), dan wordt de opdracht achter **Else** uitgevoerd en krijgt de knop het opschrift "Aan".

Merk op, dat in VBA het "="-teken de ene keer "is gelijk aan" als betekenis heeft en de andere keer "wordt". Dit wordt bepaald door de context waarin het gebruikt wordt. Opletten dus.

We zien hier verder dat we met VBA de eigenschappen van objecten binnen Access, zoals die van een knop, kunnen veranderen. Om naar de eigenschap van een object te verwijzen wordt de "punt"-notatie gebruikt: **objectnaam.eigenschapnaam**.

De eerste regel is een commentaarregel. Binnen Access begint een commentaargedeele met een apostrophe "' ". Deze kan aan het begin van een nieuwe regel staan, of in een regel, na een opdracht. Alles wat na de apostrophe op de regel komt, wordt als commentaar beschouwd en genegeerd bij het uitvoeren van de programmacode. Het einde van de regel is ook het einde van het commentaar.

- Sluit het codevenster door op het kruisje rechtsboven te drukken of druk op de Access-icoon linksboven.
- Test de werking van de knop in de Form View: bij het drukken op de knop wisselt het opschrift tussen "Aan" en "Uit".

Bij het intikken van de code hebben we gezien, dat wanneer de naam van een bekend object ingetikt wordt, er bij het intikken van de punt een lijst met beschikbare eigenschappen getoond wordt. Tik je de naam van een object in en komt de lijst niet, dan heb je een tyfout gemaakt of is het object niet binnen de programmacontext bekend. De context biedt ook een default object. Hieraan wordt gerefereerd wanneer we de objectnaam voor de eigenschapsnaam weglaten.

- Vervang nu `cmdWissel.Caption` door `Caption` overal in de bovenstaande code en test de knop nog eens. Wat is het default object?
- Bewaar het formulier als **frmKlad**.

2.3.3. Een met de Wizard gemaakte knop aanpassen

*Doel: na afloop zal het drukken op de open-knop op **frmStartFormulier** een nieuw formulier openen en **frmStartFormulier** sluiten.*

- Open **frmStartFormulier** in Design View
- Selecteer met de menu-knop de knop voor het "Bekijken en bewerken Studenten" en kies de bovenste optie: **Build Event ...**

Als er nog geen code is ingevoerd, moet je hier kiezen tussen Expression Builder, Macro Builder en Code Builder. In ons geval is er al code aangemaakt en de Code Builder wordt direct opgestart.

We zien nu de procedure, die door de Command Button Wizard gegenereerd is voor de actie Open uit de categorie Form Operations. De regel die het echte werk doet is:

```
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

De overige code in de procedure is voor foutafhandeling. De wizard voegt deze regels toe om ervoor te zorgen dat de toepassing niet stilvalt, wanneer er een fout optreedt in de procedure. Voor meer informatie over foutafhandeling en debuggen, zie Bijlage C.

`DoCmd` is de naam van een VBA-object dat veel gebruikt wordt voor interfacetaken zoals het openen van een formulier of het afsluiten van een toepassing. Andere typen van objecten die we al tegengekomen zijn, zijn formulieren en knoppen. Wanneer we in het codevenster `DoCmd` selecteren en <F1> indrukken, wordt de Visual Basic Help-pagina voor de `DoCmd`-property opgeroepen. Uitleg over het `DoCmd` object en het gebruik ervan is te bekijken via de link naar het `DoCmd` object op deze pagina.

`OpenForm` is een **methode** van `DoCmd`. Elk Access-object beschikt, naast een aantal eigenschappen, over een aantal methoden, die bij dat object gebruikt kunnen worden. Andere methoden van `DoCmd` zijn `Close` (om een formulier te sluiten) en `Quit` (om een toepassing af te sluiten). We kunnen deze methoden al bij **frmStartFormulier** aantreffen, door door het codevenster heen te scrollen. Ook hier gebruiken we de punt-notatie om een zekere methode van een bepaald object aan te geven: **objectnaam.methodenaam**.

`OpenForm` kan zeven, gedeeltelijk optionele argumenten (parameters) accepteren om zijn werk te doen. Het eerste, verplichte argument (van het type `String`) is de naam van het formulier dat geopend moet worden. Het tweede en derde argument zijn in deze context niet

nodig en worden overgeslagen. Omdat er een vierde argument wordt meegegeven moeten de tussenliggende komma's wel getypt worden. Het vierde argument is een conditie die op de gegevensbron onder het formulier toegepast kan worden (een zogenaamd **filter**). Omdat frmStartFormulier niet op een gegevensbron gebaseerd is is deze parameter (ook een String) leeg.

Samengevat zorgt deze regel er dus voor dat **frmStudent** geopend wordt. Willen we daarna frmStartFormulier sluiten, dan moet DoCmd nog eens aangeroepen worden.

- Voeg de volgende regels toe na bovenstaande regel:

```
stDocName = "frmStartFormulier"
DoCmd.Close acForm, stDocName
```

acForm is een in VBA gedefinieerde constante, die aangeeft dat het argument dat volgt een formulier is. Hieruit blijkt dat VBA geen conventie voor naamgeving afdwingt, en dat een goede, systematische naamgeving de verantwoordelijkheid van de programmeur is.

- Sla het formulier op en probeer de knop uit.
- Wat gebeurt er wanneer we de argumenten van de Close-methode weglaten?
- Wat gebeurt er wanneer we de Close-methode aanroepen voor de Open-methoden?

Wanneer we een nieuw formulier openen, wordt dat formulier het actuele formulier. In VBA-terminen: het nieuwe formulier krijgt de focus. Wanneer we DoCmd.Close zonder argumenten aanroepen, wordt het actuele formulier gesloten.

- Pas nu ook de andere **Open**-knoppen op **frmStartFormulier** op overeenkomstige wijze aan.
- Voeg aan **frmStudent** een knop toe in de **Form Footer**-sectie van het formulier, met als opschrift "Terug", die frmStartFormulier opent en het formulier zelf sluit. Gebruik weer de **Control Wizard** om een "formulier-sluiten"-knop aan te maken en vervang de opdracht DoCmd.Close door:

```
Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "frmStartFormulier"
DoCmd.OpenForm stDocName, , , stLinkCriteria

stDocName = "frmStudent"
DoCmd.Close acForm, stDocName
```

- ✘ • Stel tot slot de eigenschap **Close Button** van het formulier in op "nee".

Hiermee wordt de Windows-knop voor het sluiten van het venster buiten werking gesteld en kan **frmStudent** alleen afgesloten worden door terug te gaan naar **frmStartFormulier**. Dit biedt de programmeur de gelegenheid om de taken die met het formulier dienen te gebeuren netjes af te ronden. De gebruiker kan het werk niet abrupt beëindigen door het venster te sluiten.

- Herhaal de aanpassingen voor **frmScore** en **frmSamen**.

Tot slot nog twee opmerkingen.

Wanneer we informatie tussen twee formulieren willen uitwisselen is dat alleen mogelijk, wanneer beide formulieren open staan. Het sluiten van het ene formulier moet dan pas plaatsvinden na het kopiëren van de gewenste informatie.

We hebben gezien, dat de Control Wizard standaard code toevoegt die lokaal in een formulier te gebruiken is. Deze code wordt dan ook bij het formulier bewaard. Daarnaast is het mogelijk om procedures of functies die voor meerdere formulieren van nut zijn buiten die formulieren te bewaren in een **module**. Modules zijn, net als formulieren en tabellen te creëren vanuit het databasevenster. We zullen hierover meer zien in hoofdstuk 5.



3. Enige Uitbreidingen

In het vorige hoofdstuk hebben we een aantal technieken gezien die nuttig zijn voor het structureren van een databasetoepassing. Hierbij hebben we geen gebruik gemaakt van de inhoud of structuur van de database. In dit hoofdstuk zullen we enige technieken laten zien voor het gebruik van gegevens in een toepassing.

3.1. **Het koppelen van besturingselementen aan een gegevensbron**

Doel: na afloop hebben we kennis gemaakt met verschillende manieren om een formulierveld te koppelen aan tabellen.

3.1.1. **Wijzigingen in de tabelstructuur verwerken in een formulier**

*Doel: na afloop is de tabel **Student** uitgebreid met een **Opleiding** veld en is het formulier **frmStudent** daaraan aangepast.*

De **Form Wizard** is een goed hulpmiddel om op basis van een tabel een bijpassend formulier te genereren. Het komt echter nog al eens voor dat er daarna nog een veld aan de onderliggende tabel toegevoegd moet worden. Deze verandering dient vaak in het bijbehorende formulier doorgevoerd te worden. Veronderstel echter dat aan het formulier al veel functionaliteit is toegevoegd. Maken we een nieuwe versie aan met de Wizard, dan moet deze functionaliteit weer opnieuw toegevoegd worden. Het is dan handiger om een extra veld toe te voegen aan het formulier en dat formulier-veld zelf met de tabel te verbinden.

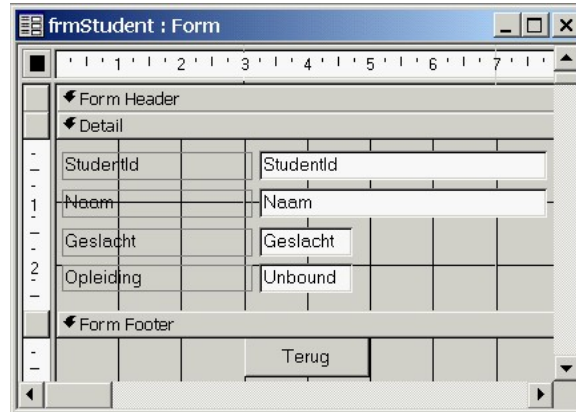
In paragraaf 1.5.1 is met de **Form Wizard** een formulier **frmStudent** gemaakt op basis van de tabel **Student**.

- Maak in de tabel **Student** een extra veld **Opleiding** van het type 'Text' aan met lengte 5. Geef dit veld de standaardwaarde "BTI".
- Open nu het formulier **frmStudent** in design mode.
- Maak de **Close Button** weer beschikbaar.
- Selecteer de **textbox**knop (tekstvak) van de toolbox en klik vervolgens op het design grid van het formulier. Op het formulier staan nu een textbox en een label.

abl

De afmetingen van het label en de textbox kunnen onafhankelijk van elkaar gewijzigd worden. Om het label en de textbox samen te verplaatsen: eerst een van beide selecteren, dan de cursor op de rand positioneren (de cursor verandert in een hand), vervolgens klikken en slepen. Om het label en de textbox afzonderlijk van elkaar te verslepen: selecteer eerst het label of de textbox en positioneer de cursor op het grotere vierkantje in de linkerbovenhoek. De cursor verandert nu in een hand waarvan de wijsvinger is opgestoken. Daarna kan de control op de gebruikelijke manier worden verplaatst.

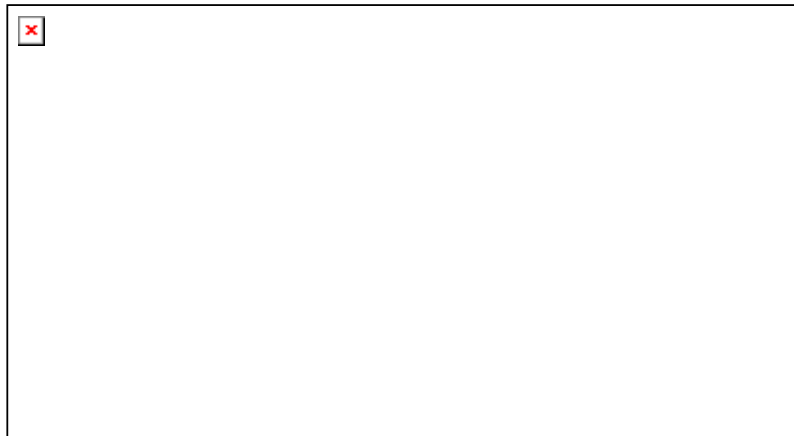
- Pas het formulier **Student** aan, zoals hieronder is aangegeven:



In de textbox staat de tekst 'unbound'. Dat betekent dat de textbox niet gekoppeld is aan de tabel **Student**. Deze koppeling kunnen we maken in de property sheets van het tekstvak.

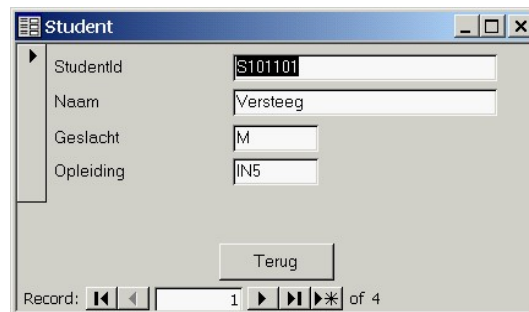


- Selecteer het tabblad **Data** en klik vervolgens in het witte vak naast 'Control Source'. Er verschijnen nu een paar knoppen. Klik op de pijlknop en er verschijnt een lijst met velden van de onderliggende tabel. Selecteer hieruit het *Opleiding* veld. Daarmee is de textbox gekoppeld aan de tabel.



- Verander op het tabblad **Other** de naam van de textbox in *txtOpleiding*.

Wanneer je de property sheets open laat en een andere control selecteert, krijg je de eigenschappen van die control te zien. Probeer je de properties voor een andere control uit het contextmenu te selecteren, terwijl de property sheets open staan, dan sluiten deze.



- Sluit het formulier en sla het op.
- Controleer dat het formulier correct werkt: typ willekeurige lettercombinaties in het *Opleiding*-veld en verifieer dit in de tabel **Student**.

3.1.2. Toevoegen van comboboxen

Doel: kennismaken met de mogelijkheden van de keuzelijst. Na afloop is het formulier StartEnquete aangemaakt met daarop twee tekstvelden en een keuzelijst.



Bij het invoeren van enquêtes willen we niet de *StudentId*s met de hand intypen: de kans op typfouten is dan vrij groot. We introduceren daarom op het formulier een combobox voor het kiezen van de gewenste waarde. De combobox is een combinatie van een keuzelijst en een tekstvak en werkt als volgt. De combobox biedt een lijst met waarden waaruit er een gekozen kan worden. Deze waarde wordt opgeslagen in het veld van de onderliggende tabel dat in de 'Control Source' van de combobox wordt aangegeven. In ons geval het veld *StudentId* van de tabel **Enquete**. De keuzelijst zelf wordt gevuld met behulp van een query (of tabel). Deze wordt gedefinieerd als 'Row Source' van de combobox. In ons geval een query die de *StudentId*'s ophaalt uit de tabel **Student**. Als extra ondersteuning zullen we niet alleen het *StudentId*, maar ook de bijbehorende *Naam* uit deze tabel ophalen. Het invoervak kunnen we gebruiken om bij een lange lijst snel naar het gewenste studentnummer te springen. Wanneer we het begin van een studentnummer intikken, toont de combobox het eerste studentnummer met hetzelfde begin. Deze ondersteuning is een gevolg van het feit, dat de instelling van de "Auto Expand"-eigenschap standaard op "Yes" staat. Kunnen we hier ook nieuwe studentnummers invoeren?

We maken nu een formulier aan dat we koppelen aan de tabel **Enquete**, zonder gebruik te maken van een Wizard. Op het formulier komt een combobox voor het *StudentId* en textboxes voor de datum en het enquêtenummer. Waarom is het eigenlijk niet nodig een textbox voor het *EnqueteNr* aan te maken?



- Open het Forms window en klik New, Design view. Selecteer **Enquete** als tabel waar de data van het formulier vandaan komen of naartoe gaan.
- Sleep *Datum* uit het venster dat gelijk met het formulier geopend wordt² op het design grid of volg dezelfde procedure als beschreven in 3.1.1. Doe dit ook met *EnqueteNr*.
- Zet de Control Wizard uit. Plaats een combobox vanuit de toolbox op de design grid.
- Verander de tekst van het label.
- Open nu de property sheets van de combobox. Kies het tabblad 'Other' en geef de combobox een geschikte naam, b.v. *cboStudentId*. Het voorvoegsel *cbo* wordt doorgaans gebruikt voor comboboxen.
- Open het Data tabblad. Klik op het vak voor de 'Control Source' en selecteer m.b.v. de pijlknop het veld *StudentId*.
- Verder zie je dat de 'Row Source Type' op 'Table/Query' staat. Klik in het vak achter 'Row Source' en vervolgens op de stippenknop. Selecteer **Student** in het 'Add Table' venster, klik op 'Add' en vervolgens op 'Close' van hetzelfde venster.
- Sleep *StudentId* en *Naam* uit de tabel in het bovenste vak van de query design grid naar respectievelijk de eerste en tweede kolom in het onderste vak en sluit de query. Sla de query op door op 'Yes' te klikken.

Het is ook mogelijk om i.p.v. de stippeltjesknop de pijlknop te gebruiken en een bestaande tabel of query te selecteren uit het afrolmenu. De SQL-tekst van de query die gegenereerd is, is te zien in het 'Row Source'-vak.

- Verander op het 'Format' tabblad 'Column Count' in 2.

Op het Data tabblad is bij 'Bound Column' de waarde 1 opgegeven. Dat wil zeggen dat de waarde van de control gekoppeld is aan de eerste kolom van de query die de waarden voor het afrolmenu ophaalt. Op het Format tabblad is bij 'Column Count' de waarde 2 opgegeven.

² Als deze 'Field List' niet verschijnt kan die via het 'View' menu te voorschijn gebracht worden

Als je een query gebruikt om de waarden van het afrolmenu te vullen kun je hiermee bepalen hoeveel kolommen van de query in het afrolmenu worden weergegeven.

- Sluit nu het formulier en geef het de naam **frmStartEnquete**.
- Open het formulier op de normale manier ('Form View').
- Voer een nieuwe enquête in (klik hiertoe op de meest rechtse record-navigatieknop).
- Sluit het formulier en open de tabel **Enquete** om te controleren dat de data inderdaad is ingevoerd.
- Open de property sheet voor de *EnqueteNr* textbox. Zet op het Data tabblad 'Enabled' op 'No'. Daardoor kan de control niet de focus krijgen en wordt de control read-only.
- Zet de "Text align"-instellingen van de invoervakken voor *Datum* en *EnqueteNr* op "Left".

Het resulterende formulier ziet er als volgt uit:

We gaan dit formulier wat opleuken.

- Open het formulier **frmStartEnquete** in design modus.
- Verander de tekst in de titelbalk van het formulier in Start Enquete.
- Maak de formulier-achtergrond geel en de letterkleur van de labels rood (selecteer alle labels tegelijkertijd)
- Geef de labels een doorlopende, zwarte rand.
- Verander de grootte van het gebruikte lettertype in 9.
- Sluit het formulier en bekijk het resultaat.

Waarom zou de kleur van de letters van het *EnqueteNr* label niet rood zijn geworden?

3.1.3. Informatie doorgeven tussen formulieren

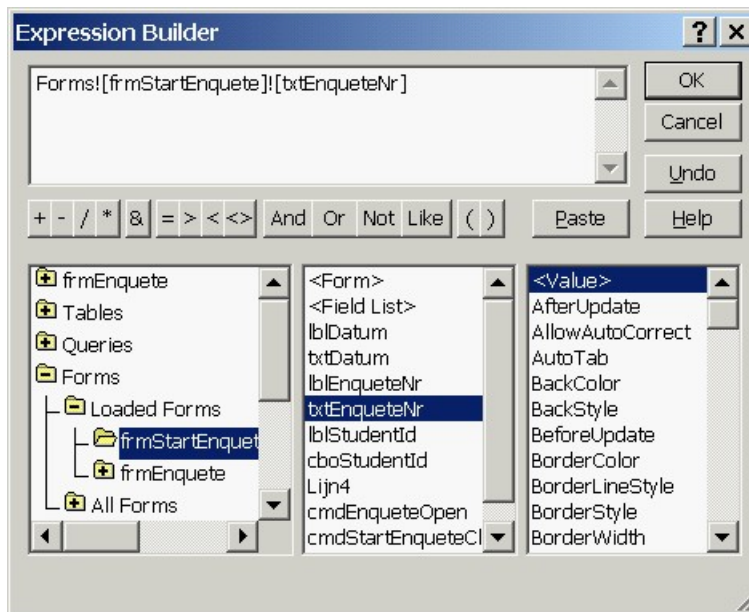
Doel: na afloop is het formulier frmEnquete gemaakt waarmee de enquête kan worden afgenomen op basis van een EnqueteNr uit frmStartEnquete.

De procedure voor het afnemen van een enquête is als volgt. We creëren een nieuwe enquête met behulp van frmStartEnquete door het ingeven van de datum en het nummer van de student die de enquête invult. De nieuwe enquête krijgt nu een nummer toegewezen. Om de vragen te beantwoorden, openen we een nieuw formulier, frmEnquete, waarop de antwoorden op de vragen ingetikt kunnen worden. Deze worden bewaard in de tabel Antwoord. Behalve

het nummer van de vraag en het antwoord is ook het nummer van de enquête nodig. Dit nummer kopiëren we uit frmStartEnquete.

De eerste taak is nu een formulier aanmaken om de enquête mee af te nemen. Het formulier is gebaseerd op de **Vraag** tabel, waarbij de **Antwoord** tabel als subformulier wordt gebruikt. Aan het formulier wordt het *EnqueteNr* toegevoegd.

- Open als voorbereiding frmStartEnquete.
- Maak met de Wizard een formulier voor de **Vraag** tabel. Zet alle velden op het formulier (let op de volgorde: *Categorie* voor *Vraag!*).
- Voeg een textbox toe met als label "EnqueteNr:"
- Geef met de Expression Builder het veld *EnqueteNr* in het Data-tabblad als standaardwaarde de overeenkomstige waarde van het veld *EnqueteNr* uit frmStartFormulier:



In het invulvak verschijnt de waarde:

```
' [Forms]![frmStartEnquete]![txtEnqueteNr] '.
```

Hier zien we meteen hoe we naar een waarde op een ander formulier kunnen verwijzen (let op de notatie met de uitroeptekens). Een voorwaarde voor het kopiëren van een waarde op deze manier is dat het formulier wel open moet staan. De waarde wordt ingevuld bij het openen van **frmEnquete**.

- Maak het design grid groter, zorg dat de Control Wizard aan staat en plaats een subformulier op de grid.
- Kies 'Use existing tables and queries', selecteer alle velden van de **Antwoord** tabel en selecteer "define My own". Koppel het VraagNr veld van formulier en subformulier.
- Koppel, nadat de wizard klaar is, het EnqueteNr-veld van het subformulier aan het EnqueteNr-veld van frmStartEnquete door als standaardwaarde

```
CInt([Forms]![frmStartEnquete]![txtEnqueteNr])
```

in te vullen. We gebruiken nu niet de letterlijke waarde (die van het type tekst is), maar de daarmee corresponderende integerwaarde door toepassing van de conversie-functie CInt(). Dit is overeenstemming met het type van het veld *EnqueteNr* in **Antwoord**.
- Geef het subformulier een redelijke naam.
- Verwijder het label dat bij het subformulier hoort of zet er een goede tekst in.
- Maak de EnqueteNr tekstboxen op formulier en subformulier read-only.

Het formulier ziet er nu als volgt uit:

Omdat het vraagnummer in het subformulier gekoppeld is aan het vraagnummer in het hoofdformulier, zal het subformulier het hoofdformulier volgen wanneer we daar door de vragen heen bladeren. We zien bij elke vraag echter ook de waarden die ingevuld zijn bij andere enquêtes (zie b.v. bij vraag 4). Dat is niet de bedoeling. We moeten de recordbron voor het subformulier aanpassen, zodat alleen voor het actuele EnqueteNr de records worden opgehaald.

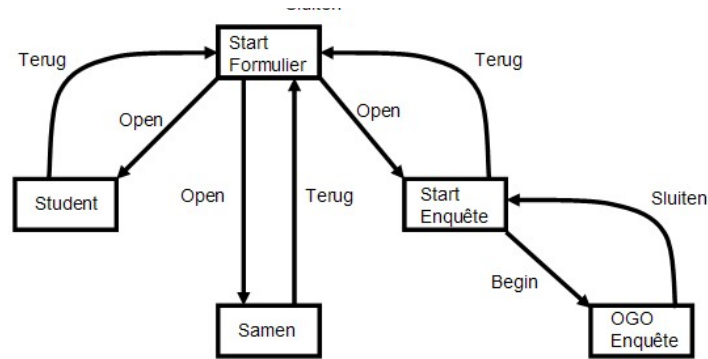
- Open frmEnquete in de Design View en ga naar het Datatabblad van het subformulier. Open deze door met de menuknop op het zwarte vierkantje in de linkerbovenhoek van het subformulier te klikken en “properties” te selecteren uit het menu. Kies de instelling “Record Source” en klik op de query builder. Het query opmaakscherm verschijnt nu.
- Voeg de bekende waarde “CInt([Forms]![frmStartEnquete]![txtEnqueteNr])” toe als selectiecriteria voor het veld EnqueteNr.

- Sla de query op en sluit uit het formulier en probeer het uit.
- Open **frmEnquete**, zonder dat **frmStartEnquete** is geopend. Wat gebeurt er?

Er rest nog een tweede taak: de verschillende formulieren in de toepassing integreren:

- Voeg nog een knop toe aan **frmStartEnquete** voor het openen van **frmEnquete**.
- Voeg een knop toe aan **frmEnquete** voor het afsluiten van dit formulier.
- Pas op **frmStartFormulier** de knop voor het openen van **frmScore** aan zodat **frmStartEnquete** wordt geopend. Pas ook het bijschrift aan.
- Voeg op **frmStartEnquete** een knop toe, waarmee dit formulier wordt gesloten en **frmStartFormulier** wordt gestart.

Het resultaat is de volgende samenhang tussen de formulieren:



Open nu **frmStartEnquete** en doe een (gedeelte van) een enquête. Probeer het subformulier zo aan te passen dat alleen het antwoord zichtbaar is en dat het nieuwe record niet zichtbaar is. Conclusie?

Terzijde: Op dit punt van de ontwikkeling hebben we een eenvoudige databasetoepassing die het een goedwillende gebruiker mogelijk maakt om studenten en enquêtes in te voeren en de resultaten daarvan te bekijken. Het is duidelijk, dat er nog de nodige verbeteringen mogelijk zijn. Het is bijvoorbeeld mogelijk om zowat alle informatie op het formulier te veranderen, terwijl alleen het antwoord op een vraag wijzigbare informatie hoeft te zijn. Verder zijn in het subformulier de velden *EnqueteNr* en *VraagNr* overbodig. Deze informatie staat al elders op het formulier. Verder kan het gebruik enigszins verwarrend zijn. Zodra we een antwoord ingevuld hebben, biedt het systeem meteen een tweede mogelijkheid om binnen dezelfde enquête op dezelfde vraag nog een antwoord te geven. Doe je dit, dan krijg je (terecht) een foutboodschap die zegt dat dit niet kan (druk op OK en <ESC> om uit deze situatie te komen). Een goede databasetoepassing zal de gebruiker van deze en andere problemen afschermen. Dit vergt echter meer programmeerwerk. Sommige aspecten daarvan zullen in hoofdstuk 5 aan de orde komen.

3.2. Default formulieren en componenten

Doel: na afloop heb je een formulier aangemaakt op basis van een zelfgemaakt formuliersjabloon.

Door defaults in te stellen voor formulieren en controls kun je je veel tijd besparen doordat je niet voor ieder formulier/control weer achter- en voorgrondkleur, font etc. hoeft in te stellen.

Defaults voor controls voor een *bepaald* formulier zijn makkelijk aan te maken: zorg dat de property's van een of andere control openstaan en klik dan in de toolbox op de control waarvoor je de defaults wilt zetten. Een alternatief is: klik in de toolbox op de control en vervolgens op de properties knop op de menubalk. Daarna kun je alle gewenste eigenschappen instellen.

- Open **frmStartEnquete** en stel de eigenschappen van de default textbox in zodat de achtergrondkleur groen is, de voorgrondkleur rood, het font een grootte 10 heeft, de hoogte 0.5cm (0.2") en breedte 3cm (1,2").
- Plaats vervolgens een textbox op het formulier.

Om voor je hele applicatie een standaard layout te krijgen moet je een default formulier aanmaken:

- Selecteer 'Options' uit het 'Tools' menu.
- Vul op het tabblad Forms/Reports bij 'Form Template' in: frmTemplate.
- Maak nu een nieuw formulier aan en stel daarop defaults in voor de achtergrondkleur van het formulier (detail sectie), voor de voor- en achtergrondkleur, afmetingen en fontgrootten van labels, textboxes en comboboxen.

- Sla vervolgens het formulier op als frmTemplate.
- Open nu een nieuw formulier zonder gebruik te maken van de Forms Wizard. Baseer het formulier op een willekeurige tabel.
- Voeg handmatig textboxes voor de velden van de onderliggende tabel toe.
- Plaats ook een extra label op het formulier.
- Doe nu hetzelfde maar gebruik dit keer de Wizard wel. Baseer het formulier weer op een bestaande tabel.

Wat is je conclusie?³

3.3. Rapporten

Doel: na afloop is het rapport AfgenomenEnquetes aangemaakt met daarin per student de afgenomen enquêtes.

Omdat rapporten (en ook Data Access Pages) zoveel overeenkomsten hebben met formulieren is een uitgebreide behandeling niet nodig. Er zijn echter enkele dingen die specifiek zijn voor rapporten. Die komen in deze sectie aan de orde aan de hand van een voorbeeld. Voeg een paar enquêtes toe aan de **Enquete** tabel, zodat er meerdere enquêtes opgenomen zijn voor een aantal studenten.

We gaan een rapport maken van de enquêtes van alle studenten.

- Maak daarvoor eerst (in design mode) een query aan waarbij de tabellen **Student** en **Enquete** gebruikt worden. Vul de kolommen van de query achtereenvolgens met *StudentId*, *Naam*, *EnqueteNr* en *Datum*. Sla de query op als **qryAfgenomenEnquetes**.
- Maak nu een nieuw rapport aan in de Design View. Stel de Record Source van het rapport in. Dit gaat net zo als bij een formulier.
- Klik nu op 'Sorting and Grouping' op de werkbalk, dan opent het Sorting and Grouping venster:



Field/Expression	Sort Order
StudentId	Ascending
Datum	Ascending

Group Properties

Group Header	Yes
Group Footer	No
Group On	Each Value
Group Interval	1
Keep Together	No

Select a field or type an expression to sort or group on

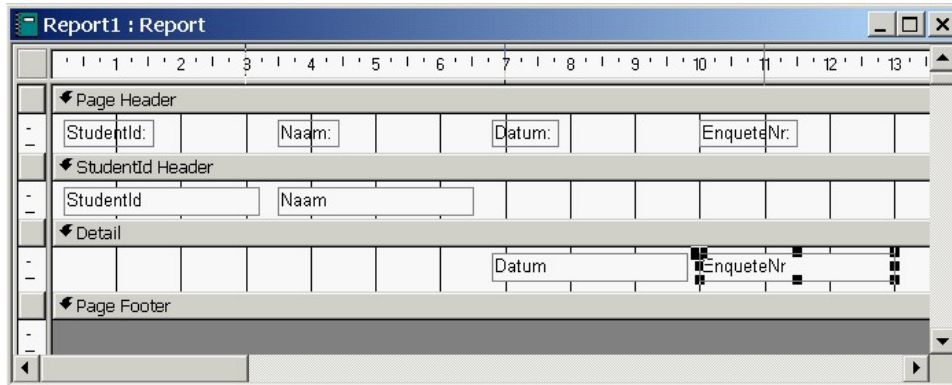
Klik in het bovenste 'Field/Expression' vak en selecteer met de pijlknop *StudentId*. Doe hetzelfde in het tweede vakje en selecteer *Datum*. We groeperen en sorteren op *StudentId* en vervolgens op 'Datum' zoals boven aangegeven. Om te voorkomen dat *Naam* en *StudentId* voor ieder vak herhaald worden, geven we op dat we een 'Group header' willen hebben voor *StudentId* (niet voor *Datum*!). Daarop verschijnt het Sorting and Grouping symbooltje voor *StudentId*. Sluit het Sorting and Grouping venster.

- Sleep de velden *StudentId* en *Naam* uit het Field List venster dat gelijktijdig is opgekomen met het rapport⁴ op de *StudentId* Header sectie van het rapport en verplaats de labels naar de Page Header sectie met cut and paste.

³ In praktische applicaties zijn form Wizards i.h.a. minder nuttig, dus weegt het extra werk om zelf de controls te plaatsen wel op tegen de tijdsbesparing van de standaard opmaak.

⁴ Verschijnt deze 'Field List' niet, zet dit dan aan m.b.v. het 'View' menu

- Zet de textboxes naast elkaar.
- Sleep dan de *Datum* en *EnqueteNr* velden naar de Detail sectie en verplaats eveneens de labels naar de Page Header sectie. Plaats de laatste twee rechts van de *Naam* textbox. Zet voor de textboxes de 'Text Align' eigenschap op 'Left'.
- Zet in de Page Header de labels boven de corresponderende textboxes en plaats er een horizontale lijn onder.
- Maak de verschillende secties zo klein mogelijk – laat een beetje ruimte over tussen de textboxes en de header en footer balk door deze laatste te verplaatsen. De witruimte in de secties wordt voor ieder record herhaald.



- Sla het rapport op als **rptEnqueteOverzicht** en open het vervolgens met 'Preview' of door dubbelklikken op het rapport.

4. Een groter voorbeeld

4.1. Vooruitblik

Doel: Overzicht van wat er nog gaat komen.

Dit deel van de cursus geeft een uitbreiding op de basiskennis van de eerste hoofdstukken. Aan de orde komen onder andere een aantal voorbeelden van wat meer gecompliceerde query's en event procedures, beveiliging en het gebruik van database wizards. Als voorbeeld gebruiken we weer een database waarmee enquêtes kunnen worden gehouden. De meeste aspecten die van belang zijn om de OGO opdracht te kunnen uitvoeren komen in dit voorbeeld aan de orde. Na het doorwerken van het materiaal moet een student in staat zijn m.b.v. de online help faciliteit in redelijk korte tijd de meest voorkomende problemen op te lossen. Als voorbereiding daarop is de toelichting bij de verschillende te realiseren onderdelen minder gedetailleerd dan in de voorgaande hoofdstukken.

In dit hoofdstuk wordt de context en de globale werking van de databasetoepassing beschreven, zodat duidelijk is met welke doelstellingen de voorbeeld database is opgezet. Daarna zullen we in hoofdstuk 5 stap voor stap het databasesysteem ontwikkelen. In hoofdstuk 6 komt vervolgens de beveiliging van het databasesysteem aan bod en in hoofdstuk 7 wordt de Database Wizard belicht.

4.2. Het voorbeeld

Doel: Beschrijving van de context van de uitgebreide casus

De voorbeelddatabase die wordt gebruikt in dit gedeelte van de cursus is een uitbreiding van de database die in het eerste gedeelte is gemaakt. De tabelstructuur is uitgebreid om meerkeuze vragen mogelijk te maken. Er is ook een afzonderlijke tabel voor de categorieën van de vragen (zie paragrafen 1.2 en 4.3). Een goede reden om dit te doen was eigenlijk al zichtbaar in het eerste gedeelte: de categorieën worden hier en daar 'misbruikt' om er een gedeelte van een vraag in te stoppen. Met de extra tabel is het gemakkelijk om een extra veld zoals Categorie-header toe te voegen. Omdat het voorbeeld gebaseerd is op een echte applicatie die in een internationale context werd gebruikt, zijn Engelse namen voor de databaseobjecten gebruikt. Een vertaling zou een te grote aanslag op de beschikbare tijd van de samenstellers hebben betekend. Het zal echter geen probleem zijn het voorbeeld van het eerste gedeelte te herkennen.

De voorbeelddatabase is ontwikkeld om een pilot project te evalueren. Deze evaluatie vond plaats d.m.v. enquêtes, waarvan de vragen en antwoorden in de database bewaard werden. In het kader van het pilot project werd in het software ontwikkeltraject een methodiek en een daarbij aansluitend hulpmiddel (tool) geïntroduceerd voor Requirements Management. Doelen van de enquête waren onder andere er achter te komen of 1) het gebruik van het tool en de methodiek een tijdsbesparing realiseerde, of 2) de gebruikers goed met de methode en met het tool uit de voeten konden en of 3) er een kwaliteitsverbetering kon worden vastgesteld. De versie van de database, die tijdens dit practicum wordt opgebouwd, omvat een aantal belangrijke onderdelen van de operationele databasetoepassing.

Verschillende deelnemers aan het project (ontwikkelaars, testers, managers) komen op verschillende manieren in aanraking met de verandering in het proces. Daarom is het nodig om aan verschillende gebruikers verschillende vragensets voor te leggen. De voorbeelddatabase zou dus ook heel geschikt zijn om enquêtes voor b.v. OGO of de Software

Engineering Projecten mee af te nemen, met aparte vragensets voor studenten, tutoren, projectmanagers etc.

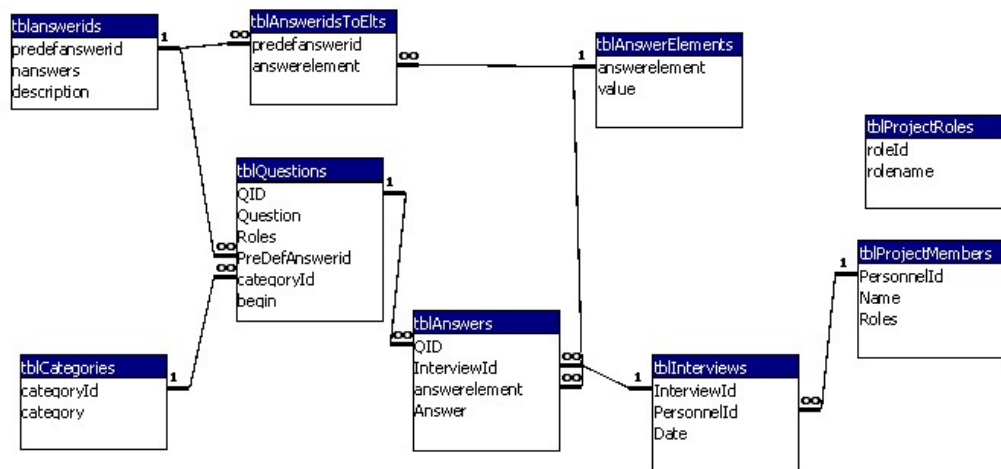
De enquête wordt in de loop van het project verscheidene keren afgenomen. Daarom is het mogelijk om bepaalde vragen alleen aan het begin van het traject, andere alleen tijdens het traject en weer andere alleen aan het eind te stellen. Bijvoorbeeld vragen over het gebruikte tool kunnen aan het eind worden gesteld (er is toch geen mogelijkheid over te stappen op een ander tool tijdens de loop van het traject), terwijl vragen over de methodiek wel tijdens de loop van het traject gesteld kunnen worden om eventueel de methodiek bij te kunnen schaven. Een van de doelstellingen van de evaluatie is om zo veel mogelijk kwantitatieve antwoorden op vragen te krijgen (dus geen open vragen), waardoor een automatische verwerking van de resultaten mogelijk is.

Een interview moet tussentijds kunnen worden onderbroken en later hervat kunnen worden. Tevens moet het mogelijk zijn een antwoord achteraf te corrigeren. Om de context van de vragen vast te leggen wordt iedere vraag in een categorie ingedeeld. Voor de beveiliging kunnen twee rollen worden onderscheiden: die van de samensteller van de enquêtes en die van de enquêteur. De eerste heeft volledig toegang tot de database, de laatste alleen tot het gedeelte dat met de beantwoording van de vragen te maken heeft. In de versie van de database die hier wordt behandeld is de verwerking van de resultaten niet opgenomen. Deze functionaliteit bleek in de praktijk uiterst nuttig om na een serie enquêtes trends snel te kunnen zien.

Behalve de tabelstructuur en de vragen zijn aanzetten voor de formulieren en een query al opgenomen, om zoveel mogelijk te kunnen concentreren op nieuwe aspecten. Er zijn echter nog wel een aantal herhalingen opgenomen om te voorkomen dat het overzicht over de applicatie verloren gaat of om het geheugen even op te frissen. In secties 4.3 t.m. 4.5 wordt een beschrijving gegeven van de voltooide applicatie. In Evaluation.mdb zijn nog niet alle onderdelen die hier beschreven worden aanwezig. Zo is bijvoorbeeld het Main Menu formulier nog niet aanwezig en missen er bepaalde knoppen. Deze worden toegevoegd in hoofdstuk 5.

4.3. Tabelstructuur

Doel: na afloop heeft de lezer een overzicht gehad van de databasestructuur van de casus.



Figuur 1 Het Access data model



- Open de database "evaluation.mdb". Klik op de relationshipsknop. Het database-diagram uit Figuur 1 wordt nu getoond.

Net als in het eerste gedeelte bestaat het centrale gedeelte van de applicatie uit de Vraag tabel (tblQuestions), de Antwoord tabel (tblAnswers), de Interview tabel (tblInterviews) en de Projectlid tabel (tblProjectMembers). De drie bovenste tabellen zijn toegevoegd om de meerkeuze antwoorden mogelijk te maken. De tblProjectRoles tabel is uitsluitend toegevoegd om gemakkelijk een rolnaam aan een getal te kunnen koppelen.

Doordat de antwoorden en de vragen in verschillende tabellen staan is het verwerken van de antwoorden op de vragen lastig: bij iedere vraag moet gekeken worden of er in het huidige interview al een antwoord is gegeven. Dit antwoord moet getoond worden, eventueel gecorrigeerd kunnen worden en daarna moet het antwoord weer worden opgeslagen. Dit laatste maakt het nodig om zelf navigatieknoppen met code te maken in plaats van de standaard door de formulier-Wizard van Access gegenereerde navigatieknoppen.

Om een geautomatiseerde verwerking van de resultaten mogelijk te maken is het nodig een tabel tblAnswerElements te hebben met antwoordelementen zoals "yes", "no" en "1". Een vraag verwijst naar een PreDefAnswerId in de tabel tblAnswerIds met vooraf gedefinieerde antwoorden. De records van deze tabel⁵ bevatten een beschrijving van het antwoord, b.v. "y/n" of "1-10", en de tabel tblAnswerIdsToElts koppelt de vooraf gedefinieerde antwoorden (PreDefAnswerId) en de antwoordelementen (AnswerElement). Behalve een verwijzing naar een antwoordelement bevat de tabel met antwoorden een veld Answer voor een tekstueel antwoord, waarmee de ondervraagden commentaar en/of een toelichting op hun antwoord kunnen geven. Zo'n mogelijkheid blijkt een goed hulpmiddel te zijn bij het interpreteren van de resultaten.

Voor ieder interview wordt behalve een verwijzing naar een projectmedewerker een datum bijgehouden: dit legt het interview uniek vast (er worden nooit twee interviews met dezelfde persoon op dezelfde dag gehouden).

Omdat er vrij veel verschillende rollen zijn, is ervoor gekozen de rollen waarvoor een bepaalde vraag wordt gesteld in een bit-string (Roles) te coderen. Een tweede reden om dit te doen is omdat hierdoor wat extra programmeerwerk noodzakelijk wordt. Het was namelijk ook mogelijk geweest om de rollen m.b.v. een Access 'option group' vast te leggen. Daarentegen zijn in de vragen tabel wel expliciet de velden "begin", "during" en "end" opgenomen. Het zal blijken wat de consequenties voor de applicatie zijn van deze verschillende manieren van aanpak.

Alle tabellen hebben hier een naam die met tbl begint. Door deze naamgeving is het altijd duidelijk dat het een tabel betreft. Formulieren krijgen een prefix frm, queries qry en reports rpt. Voor documentatie is dit handig omdat dan altijd duidelijk is over welk soort object we het hebben. Het expliciet maken van het type van een programma-object voorkomt een grote klasse van programmeerfouten.

4.4. Functionaliteit

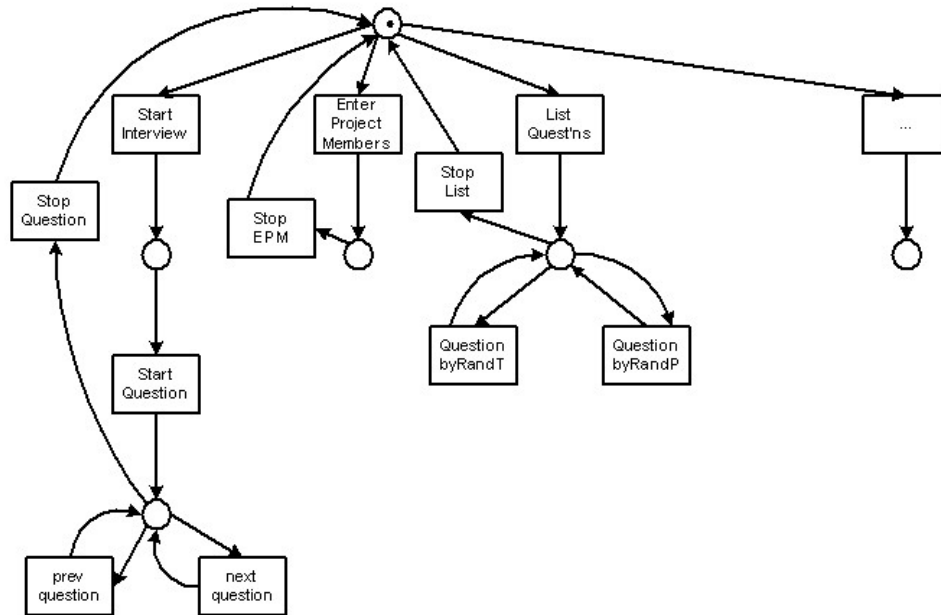
Doel: na afloop heeft de lezer een overzicht gehad van de verschillende taken die het databasesysteem moet ondersteunen.

De functionaliteit valt uiteen in twee gedeeltes: een gedeelte om de enquête op te zetten (5.1, 5.2 en 5.3) en een gedeelte om de vragen te stellen (5.4 en 5.5).

Voor het opzetten van de enquête zijn een paar formulieren echt belangrijk. Het invullen van de antwoordelementen en van de vooraf gedefinieerde antwoorden kan, gezien het aantal, eenvoudig met de hand in de tabel worden gedaan. Voor de koppeling van de antwoordelementen en de vooraf gedefinieerde antwoorden is een formulier heel handig (frmAnswerIdsToElements). Voor het aanmaken van de vragen is een formulier essentieel (frmEnterQuestions). Het is ook handig om voor het invoeren van projectmedewerkers een formulier te maken (frmEnterProjectMembers). De voornaamste reden hiervoor is dat een personeelslid vaak meerdere rollen heeft, b.v. de leider van het testteam kan zelf ook als tester actief zijn. Het aangeven van de rollen moet dus in interactie met het personeelslid plaats vinden.

⁵ Het veld *answers* (het aantal antwoorden) is niet nodig maar wordt verderop gebruikt als illustratie voor een vaak voorkomende situatie.

Het formulier frmEnterQuestions bevat een methode AddRole() en de attributen role en roles. Het attribuut roles heeft als waarde de bovengenoemde bitstring. Het attribuut role (dat geïmplementeerd wordt met een keuzelijst) is toegevoegd om makkelijk te kunnen vastleggen voor welke rollen een vraag gesteld moet worden. Door het aanroepen van de methode AddRole() (een druk op de corresponderende knop) wordt de actuele waarde van role gecombineerd met de waarde van het attribuut roles, dat vervolgens de nieuwe waarde toegekend krijgt. Dezelfde constructie gebruiken we ook voor frmEnterProjectMembers.



Figuur 2: Menustructuur

Voordat de vragen gesteld kunnen worden, moeten we eerst een interview aanmaken of we moeten doorgaan met een oud interview. Zodra dit gedaan is kan het formulier voor de enquête worden geopend.

Voor een weergave van de onderlinge samenhang tussen de verschillende formulieren gebruiken we het Petri net in Figuur 2. De menukeuzen worden voorgesteld door transities en de toestanden van het systeem door plaatsen.

Voor de implementatie van de Next, Previous en Stop overgangen kunnen we meestal de standaard Access of Windows faciliteiten gebruiken en hoeven we dus niets extra's te doen.

Er is hier gekozen voor navigatie tussen de verschillende formulieren m.b.v. knoppen. Je kunt je applicatie ook organiseren als een aantal tabbladen met behulp van een "tab" control op een formulier. Details over gebruik hiervan zijn te vinden in de online help.

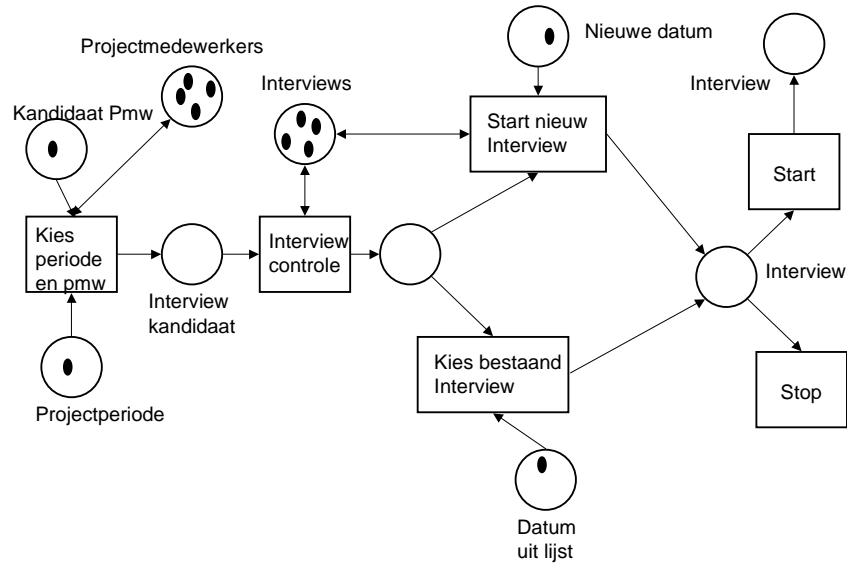
4.5. Procedure voor het afnemen van een enquête

Doel: na afloop heeft de lezer een overzicht gehad van de stappen die bij het afnemen van een enquête moeten worden gezet.

Vanuit het Main Menu wordt met de knop Questionnaire een formulier frmStartInterview geopend. Op dit formulier kan met een combobox een projectmedewerker gekozen worden. Tevens moet aangegeven worden of voor het interview "begin", "during" of "end" of een combinatie van deze moet gelden. De datum voor het interview staat standaard op de huidige dag. Bestaat het interview nog niet, dan wordt een berichtvenster (messagebox) getoond waarmee gevraagd wordt of er een nieuw interview moet worden aangemaakt. Wordt hierop met "yes" geantwoord, dan wordt een nieuw record in de tblInterviews aangemaakt. Wordt er met "no" geantwoord, dan kan de enquêteur met een combobox kiezen uit de data van bestaande interviews. Met de knop Start Interview wordt het formulier met de vragen geopend.

Hierbij moet de applicatie controleren dat er een projectperiode gekozen is, dat er een datum en projectmedewerker is gekozen en dat er een interview is aangemaakt of geselecteerd.

Deze procedure is samengevat in Figuur 3.



Figuur 3 Klassiek Petrinet m.b.t. het starten van een interview

5. Formulieren

We zullen nu stap voor stap de verschillende onderdelen van het systeem opbouwen en aan elkaar koppelen. We gaan daarbij uit van het bestand "evaluation.mdb".

5.1. frmAnswerIdsToElts, frmMainMenu en frmAnswerIds

In deze paragraaf komt een onderdeel van de toepassing aan de orde waarin de vooraf gedefinieerde antwoorden onderhouden kunnen worden. We zullen ook een menusysteem opzetten zodat er tussen formulieren gewisseld kan worden. De tabel met antwoordelementen is al gevuld, evenals de vooraf gedefinieerde antwoorden. Van het formulier dat deze aan elkaar koppelt, is de basisopzet al aanwezig.

- Open het formulier frmAnswerIdsToElts dat op de tabel tblAnswerIdsToElts is gebaseerd in design mode.

Om de koppeling gemakkelijk te kunnen maken moeten de textboxes veranderd worden in comboboxen, waarmee je records uit respectievelijk tblAnswerIds en uit tblAnswerElements kunt selecteren.

De comboboxen kunnen zo gemaakt worden, dat de leesbare versie van de elementen getoond wordt, terwijl de waarden van de sleutels van de tabellen toegekend worden aan de combobox. Dus de combobox preDefAnswerId laat het veld "description" zien aan de gebruiker, maar de combobox krijgt de waarde van het veld "preDefAnswerId" van het geselecteerde record. De controls blijven aan de velden van tblAnswerIdsToElts gekoppeld.

- Verander de predefanswerld textbox met behulp van het contextmenu (of via het Format menu) – "change to" – in een combobox. Open de eigenschappen van de combobox en klik op het tabblad Data. Laat de Control Source onveranderd. Zorg dat Row Source Type op Table/Query staat, klik rechts naast Row Source (op de stippen) en maak een query aan die uit tblAnswerIds het predefanswerld (in de eerste kolom) en de description haalt. De Bound Column moet op 1 staan. Verander vervolgens op het tabblad Format de Column Count in 2 en Column Widths in 0";2" of in 0cm;2cm (afhankelijk van je instellingen). Hierdoor wordt de "predefanswerld"-kolom verborgen en alleen de "description"-kolom getoond.
- Doe hetzelfde voor de answerelement textbox. In dit geval is het niet nodig om via de stippenknop een query aan te maken, maar kan als row source de tabel tblAnswerElements worden gekozen (waarom? – en kan het in het eerste geval ook zonder query?).

Nieuwe koppelingen tussen antwoord-elementen kunnen nu gemaakt worden door eerst met frmAnswerIds een nieuw record in de tblAnswerIds aan te maken en eventueel nieuwe antwoord-elementen in tblAnswerElements, dan het formulier te openen en voor elk antwoord-element de koppeling te maken met het antwoord-Id. Door op het Data tabblad van frmAnswerIdsToElts "Data Entry" op yes te zetten krijg je niet meer alle andere records te zien.

- Probeer voor Data Entry "yes" en "no" beide uit.
- Maak een nieuw antwoordtype 1-3 aan dat als elementen 1, 2 en 3 heeft. (Voeg eerst met frmAnswerIds het antwoord toe, open daarna frmAnswerIdsToElts om de koppeling te maken). Kies een nog niet bestaand predefanswerld.
- Maak een nieuw formulier aan als aanzet voor het hoofdmenu van de applicatie. Kies in het dialoogvenster "design view" en selecteer geen tabel of query.
- Zorg ervoor dat in de Toolbox de Control Wizard geactiveerd is.

We gaan nu een knop toevoegen aan het hoofdmenu om nieuwe meerkeuze antwoorden te maken. Normaal gesproken zouden we nu een knop maken om frmAnswerIds te openen en op dit formulier een knop om frmAnswerIdsToElts te openen. Dit leidt echter tot een technisch probleempje dat we hier willen omzeilen. Daarom openen we op het hoofdmenu



frmAnswerIdsToElts. Vanuit dat formulier moet dan voor een nieuwe meerkeuzevraag eerst een record worden aangemaakt in tblAnswerIds voordat de elementen kunnen worden toegevoegd.

- Kies uit de toolbox de command button en plaats deze op het hoofdmenu formulier. De command button wizard start nu op. Kies "form operations", "open form" en selecteer frmAnswerIdsToElts. Volg de aanwijzingen van de wizard. Kies voor "Display all elements". Kies ervoor een tekst op de knop te plaatsen i.p.v. een plaatje en kies een zinvolle naam voor de knop.
- Maak de property sheets van het formulier zichtbaar door het vierkantje links boven in het formulier rechts aan te klikken en vul op het tabblad Format bij "Caption" in: "Main Menu". Zet op hetzelfde tabblad "Navigation Buttons": "No" en zet ook de "Record Selectors" uit.
- Bekijk vervolgens de properties van de knop. Op het tabblad "Event" staat bij "On Click": "[Event Procedure]". Druk op de knop met de stippen die verschijnt als je in dat vakje klikt. Je ziet nu de Visual Basic for Applications (VBA) code die gegenereerd is door de wizard. Voor volgende stappen zal het vaak nodig zijn code toe te voegen aan de gegenereerde code.



Onder welke omstandigheden een event handler van een formulier of control wordt aangeroepen, kun je uitvinden door te klikken achter de betreffende handler op het tabblad "events" van de betreffende properties sheet en vervolgens op de F1 toets te drukken.

- Maak een knop op het formulier frmAnswerIdsToElts om frmAnswerIds te openen. Met het formulier frmAnswerIds kan een nieuw AnswerId aangemaakt worden. Voeg aan de gegenereerde code de volgende regel toe (vóór de foutafhandelingscode...), waardoor het formulier frmAnswerIdsToElts gesloten wordt:

```
DoCmd.Close acForm, "frmAnswerIdsToElts"
```

- Maak op het formulier frmAnswerIds een knop "close" waarmee frmAnswerIdsToElts weer wordt geopend. Voeg aan het eind van de gegenereerde code de volgende regel toe, waardoor frmAnswerIds inderdaad wordt gesloten:

```
DoCmd.Close acForm, "frmAnswerIds"
```

Met het DoCmd object kunnen Access "Actions" in VBA worden uitgevoerd, in bovenstaande voorbeelden om een formulier te sluiten. In deze commando's is acForm een constante die het systeem vertelt dat de actie op een formulier betrekking heeft. Met het DoCmd object kunnen ook formulieren geopend worden, records opgezocht worden, enzovoort. Het volgende voorbeeld maakt het zevende record van het formulier Employees het huidige record.

```
DoCmd.GoToRecord acDataForm, "Employees", acGoTo, 7
```

Nadere details zijn te vinden in de VBA help bij het DoCmd object, met name bij de 'Methods' link op deze pagina, waarin je de Help voor de methoden van het DoCmd object kunt vinden.

- Maak een nieuw formulier aan waarmee antwoordelementen toegevoegd kunnen worden en koppel dit formulier aan het formulier frmAnswerIdsToElts.
- Maak met de drie formulieren een nieuw AnswerId met waarden 'a', 'b' en 'c' aan.

5.2. frmEnterQuestions

Om een formulier te maken waarmee de vragen aangemaakt kunnen worden moet er enige VBA code worden geschreven. Voor degene die de vragen aanmaakt, moet het sommeren van de coderingen van een aantal rollen door de applicatie gedaan worden en verder moeten de geselecteerde rollen ook in een leesbare vorm getoond worden.

- Open het formulier frmEnterQuestions. Voor PreDefAnswerId en CategoryId zijn al comboboxen aangemaakt op dezelfde manier als boven.

Om de rollen, waarvoor een vraag gesteld moet worden, te kunnen definiëren gebruiken we het volgende: 1) een extra combobox, waarin een rol geselecteerd kan worden, 2) een tekst veld om de geselecteerde rollen weer te geven en 3) een command button om de codering van een zojuist geselecteerde rol (een macht van 2!) op te tellen bij de al eerder geselecteerde rollen (een som van machten van 2) en de leesbare vorm van de rol toe te voegen aan de tekst box.

- De combobox "role" is al aangemaakt evenals de textbox txtRoles. De combobox is niet aan een veld gebonden, bestaat uit twee kolommen ("bound column": 1) en haalt z'n waarden uit tblProjectRoles. De defaultwaarde voor het veld is nul (0).
- Zet de "Control Wizard" knop van de toolbox uit en selecteer uit de toolbox de command button en plaats deze op het formulier. Er staat nu een command button op het formulier. Wijzig m.b.v. de properties van de knop eerst de naam van de knop in b.v. "cmdAddRole": "Commandxx" is niet erg verhelderend.
- Klik in het veld achter het "On Click" event om de knop met de stippen te zien te krijgen. Klik op deze knop en kies voor "code builder". Acces heeft een procedure cmdAddRole_Click() aangemaakt. Neem eventueel de code voor foutafhandeling over (wel aanpassen natuurlijk!) van een eerder aangemaakte knop. Je kunt natuurlijk ook de wizard gebruiken, dan een (willekeurig gekozen) actie opgeven en deze achteraf verwijderen. De code zullen we dadelijk toevoegen. Sluit nu eerst de VBA editor.

Figuur 4 Layout frmEnterQuestions

We hebben nu alle basiselementen op het formulier geplaatst (zie Figuur 4). Wat over blijft is ervoor te zorgen dat de berekening en weergave van de rollen waar een vraag voor bestemd is, correct plaats vindt. Hiertoe dienen we te zorgen voor de juiste beginwaarden voor de verschillende betrokken velden.

Voeg op dezelfde manier als hiervoor een event handler toe voor het On_Current event van het formulier – het vierkantje linksboven op het formulier rechts aanklikken om de properties van het formulier te krijgen.

In de rest van dit hoofdstuk worden telkens stukjes pseudocode gegeven, waarna de VBA specifieke elementen van de oplossing worden aangereikt. Uit deze elementen kun je zelf de oplossing vinden. Uiteraard is het niet de bedoeling dat de oplossing uitsluitend met knip en plakwerk tot stand komt! Baseer je oplossing op de pseudocode en gebruik de voorbeelden voor de technische realisatie.

Pseudocode voor de AddRole event-handler

De event-handler van de AddRole knop kan informeel als volgt beschreven worden:

```

Geef de variabele intAllRoles de waarde die volgt
    uit de corresponderende control (zodanig 0)
Als Role nog niet voorkomt in intAllRoles
    intAllRoles = intAllRoles + Role6
Bereken de stringrepresentatie van intAllRoles
Zet de string en de nieuwe waarde van intAllRoles in de
    tekstvelden

```

De procedure CalculateRoles voor het uitrekenen van de string-representatie

Om de string met de leesbare vorm van de rollen op te bouwen moeten we de tabel met de rollen doorzoeken (op soortgelijke wijze kan een query worden geopend en het resultaat worden doorlopen)⁷:

```

Dim rst As New ADODB.Recordset
' open de tabel
rst.Open "tblProjectRoles", CurrentProject.Connection,
        adOpenKeySet, adLockOptimistic
' ga naar het eerste record
rst.MoveFirst
Do While Not rst.EOF          ' stop bij end of file
    ....
    ....                      'ga naar het volgende record
    rst.MoveNext
Loop

```

We gebruiken hier het ActiveX Data Objects (ADO) model om toegang te krijgen tot de database objecten.

Het statement dat de recordset opent heeft als eerste parameter de naam van de te openen tabel of query.

De waarde van de tweede parameter geeft informatie over de database waarmee verbinding is (hier: de Evaluatie database zelf, maar het is ook mogelijk een verbinding met een andere database te openen.) – de door Access gezette waarde van deze string:

```

Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Data Source=G:\OGOAccess\Evaluation uitwerking
ado.mdb;Mode=Share Deny None;Extended Properties="";Jet OLEDB:System database=C:\Documents and
Settings\eluit\Application Data\Microsoft\Access\System.mdw;Jet OLEDB:Registry
Path=Software\Microsoft\Office\11.0\Access\Jet\4.0;Jet OLEDB:Database Password="";Jet OLEDB:Engine Type=5;Jet
OLEDB:Database Locking Mode=0;Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Global Bulk Transactions=1;Jet
OLEDB:New Database Password="";Jet OLEDB:Create System Database=False;Jet OLEDB:Encrypt Database=False;Jet
OLEDB:Don't Copy Locale on Compact=False;Jet OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False

```

Als je zelf zo'n connection string moet opgeven kan het gelukkig veel eenvoudiger, iets als: "Provider=Microsoft.Jet.OLEDB.4.0;Data Source='c:\new.mdb'" is voldoende als connection string om de externe database new.mdb te openen.

⁶ Merk op dat we niet hoeven te controleren of Role <>0 is. Ga dit na.

⁷ Pas op met kopiëren uit de handleiding: voor de leesbaarheid zijn b.v. een return en tabs toegevoegd in het open statement; deze leiden tot fouten in de code!

De derde parameter zegt iets over het type cursor dat gebruikt wordt: dit is een dataelement dat m.n. bepaalt of de data aangepast kan worden (adOpenDynamic) of niet (adOpenStatic) en of veranderingen zichtbaar zijn voor andere gebruikers van de database. Bij adOpenKeySet zijn nieuwe records die door anderen worden toegevoegd niet zichtbaar (dit is wel het geval bij adOpenDynamic), records die weggelaten zijn door andere gebruikers zijn niet toegankelijk maar veranderingen in bestaande records door andere gebruikers zijn wel zichtbaar.

De vierde parameter geeft aan hoe locking gedaan wordt, in dit geval "optimistic locking" op record basis. Meer informatie kun je in de VBA help vinden.

Als voorbeeld volgt hieronder de code om de string op te bouwen, zodra een nieuw record "current" wordt. Voor iedere rol moet worden nagegaan of deze voorkomt in de waarde die de Roles textbox bevat. Hierbij wordt gebruik gemaakt van de bitwise And-operatie op twee integers, zeg a en b . De expressie a And b levert een integer c waarvan het bit op plaats n , c_n , 0 is als a_n of b_n 0 zijn en 1 als a_n en b_n beide 1 zijn (m.a.w. $c_n = a_n$ And b_n). Dus b.v. 4 And 2 (100 And 010) levert 0 op terwijl 2 And 3 (10 And 11) de waarde 2 (10) geeft. Access converteert dit naar een logische waarde, wanneer de expressie in de conditie van een If statement wordt gebruikt. De regel hiervoor is dat 0 wordt geconverteerd naar False, terwijl alle andere waarden opgevat worden als True.

Voor stringconcatenatie wordt & gebruikt. In het voorbeeld hieronder kunnen strRoles en rst!rolename zonder meer geconcateneerd worden omdat dit variabelen van het type string zijn. Om de rollen van elkaar te scheiden wordt een; en een spatie gebruikt. Door deze tussen "" te zetten maak je er ook een string van.

In de volgende programmatekst zien we de expressie Me!Roles. "Me" refereert altijd aan het huidige formulier of aan de onderliggende tabel. Me!Roles representeert dus de waarde in de Roles tekstbox, een integer omdat Roles een "number" is in de tabel tblQuestions.



```
Dim rst As New ADODB.Recordset, strRoles As String
strRoles = ""
rst.Open "tblProjectRoles", CurrentProject.Connection,
        adOpenKeySet, adLockOptimistic
rst.MoveFirst      ' check the roles specified by Roles
Do While Not rst.EOF
    If rst!roleId8 And Me!Roles Then
        ' And: bitwise and van de operanden
        strRoles = strRoles & rst!rolename & ";"
    End If
    rst.MoveNext
Loop
rst.Close      ' sluit de recordset (table)
```

Het verdient de aanbeveling om voor het uitrekenen van de string-representatie een procedure

```
Sub CalculateRoles(intAllRoles As Integer, strRoles As String)
```

te maken in de "Modules" object groep. Deze procedure kan dan ook aangeroepen worden vanuit de On Current event handler van het formulier, zodat de txtRoles textbox direct bij het openen van een nieuw record wordt gevuld. Bovendien kan deze procedure dan ook gebruikt worden vanuit andere formulieren, b.v. die waarmee projectmedewerkers ingevoerd kunnen worden. Vanuit een Module mag het keyword *Me* niet gebruikt worden, vandaar dat de bitrepresentatie als parameter moet worden doorgegeven. De string-representatie moet als parameter worden geretourneerd.

⁸ De velden van een record kunnen m.b.v. rst!field of met rst("field") benaderd worden

- Open de Modules groep in het database window  en druk op 'New' 
- Voer de code voor de procedure in. Neem bovenstaande code niet klakkeloos over: deze is geschreven alsof het een deel van de AddRole event-handler zelf is. Behalve de toevoeging van de procedure header en end zijn twee wijzigingen in bovenstaande code nodig (welke?). Let op: een parameter van een procedure mag niet ook nog met een Dim statement worden gedeclareerd!

De module wordt pas zichtbaar bij de modules groep nadat die gesaved is en kan ook pas daarna gebruikt worden:

- Save de database via het file menu of via de werkbalk. Daarbij wordt gevraagd of je de module wilt opslaan. Geef de module een zinvolle naam. Hier wordt de naam "EvaluationFunctions" gebruikt.

Implementatie van de AddRole event-handler

De implementatie van het stukje pseudocode

```
Geef de variabele intAllRoles de waarde die volgt
uit de corresponderende control (zonodig 0)
```

gaat als volgt:

Als er nog geen rollen voor de vraag zijn geselecteerd, heeft de textbox "Roles" de waarde "Null". De variabele intAllRoles moet dan de waarde 0 krijgen en anders de huidige waarde van het Roles veld:

```
Dim intAllRoles As Integer
' initialiseer AllRoles
If IsNull(Me!Roles) Then ' waarde Roles control v.h. formulier
    intAllRoles = 0
Else
    intAllRoles = Me!Roles
End If
```

Het stukje pseudocode

```
Als Role nog niet voorkomt in intAllRoles
```

resulteert in

```
Dim isIncluded As Boolean
IsIncluded = intAllRoles And Me!Role
If Not IsIncluded Then
    .....
End If
```

Aan het stukje pseudocode

```
intAllRoles = intAllRoles + Role
```

hoeven we uiteraard niets te veranderen. Voor

Bereken de stringrepresentatie van `intAllRoles`

Hebben we de procedure `CalculateRoles` gemaakt. De procedure kan worden aangeroepen met

```
Call CalculateRoles(intAllRoles, strRoles)
```

Of met

```
CalculateRoles intAllRoles, strRoles
```

Tenslotte hebben we nog het laatste stukje van de informele specificatie:

```
Zet de string en de nieuwe waarde van intAllRoles in de
tekstvelden
```

De vertaling van dit stukje pseudocode in VBA ligt voor de hand.

- Voeg de implementatie van deze stukjes pseudocode toe aan de `On_Click` event handler van de knop "Add Role". Er is ook nog een extra `Dim` statement nodig.

De implementatie van de `On-Current` handler van het formulier ligt nu ook voor de hand. Ga eerst na wat de verschillen met de event handler van de `Add-Role` knop zijn.

- Implementeer de `On-Current` event handler
- Voeg een knop toe aan het hoofdmenu om het formulier te openen
- Ga na of voor bepaalde controls nog goede default waarden moeten worden gekozen. Overweeg ook of de defaultwaarden door de `On-Current` handler opnieuw gezet moeten worden (dit geldt ook voor paragraaf 5.3!).

Het ophalen van waarden uit de kolommen van een combobox

Het komt in de praktijk regelmatig voor dat de waarden van de kolommen van een tekstbox van belang zijn. In onderstaand voorbeeld wordt getoond hoe deze waarden toegankelijk zijn. In dit voorbeeld wordt de waarde van de stringrepresentatie van een rol opgehaald uit de tweede kolom van de combobox in plaats van uit de tabel `tblRoles`. De kolommen van een combobox worden gerepresenteerd als een attribuut van de combobox: een array `Column`. Nu is het wel nodig om te controleren of er echt een rol gekozen is, omdat het niet mogelijk is de tweede kolom van een combobox een default waarde te geven. Met deze aanpak is het nodig om te verifiëren of de rol al in de stringrepresentatie is opgenomen (waarom?).

```
If Not IsNull(Me!Role.Column(1))9 Then
    strRole = Me!Role.Column(1) & ";" " ' rol geselecteerd
Else
    strRole = ""
End If
IsIncluded10 = Role And intAllRoles
If Not IsIncluded Then
' voeg rol toe als die nog niet in Roles zit
    Me!Roles = intAllRoles + Role
    strRoles = strRoles & strRole
    Me!txtRoles = strRoles ' zet de string in de textbox
End If
```

⁹ De adressering van de kolommen begint bij nul.

¹⁰ In tenminste 1 geval bleek Access de guard `Not(Roles And intAllRoles)` verkeerd te berekenen

5.3. frmEnterProjectMembers

Completeer het formulier om de projectmedewerkers in te voeren en voeg ook hiervoor een knop aan het hoofdmenu toe. Het *personnelId* veld is read-only gemaakt door op het Data tabblad van de properties van de control "Enabled" op No te zetten (autonumber veld). Op het formulier moeten alle gegevens van de tabel worden ingevuld, de rollen moeten op dezelfde wijze als voor *frmEnterQuestions* kunnen worden toegevoegd. Je kunt hiervoor de event handler van *frmEnterQuestions* eenvoudig kopiëren. Omdat het keyword "Me" in een Module niet is toegestaan kun je deze handler helaas niet als procedure in een Module opnemen. Zorg ook dat de On-Current handler van het formulier de string representatie van de rollen vult. Merk op dat de namen van de tekst- en comboboxen gelijk zijn aan die van de corresponderende controls van *frmEnterQuestions*. Uiteindelijk moet het formulier er als volgt uit komen te zien:

- Maak enkele projectmedewerkers aan

5.4. frmStartInterview

Het volgende formulier is *frmStartInterview*. Met dit formulier wordt een nieuw interview aangemaakt of een oud interview voortgezet, waarna een enquête gestart wordt, vergelijkbaar met het formulier van het eerste gedeelte.

- Open het formulier *frmStartInterview* in design view. Geen van de controls is aan een tabel gebonden. "Navigation Buttons" zijn uitgezet.

Er zijn twee comboboxen: een (*personnelId*) om een projectmedewerker te selecteren, en een (*date*) om een datum te selecteren. De default waarde van de *Date* combobox is *Date()*. Deze functie retourneert de huidige datum (de functie *Now()* retourneert ook de tijd). Na het kiezen van een datum en een (al ingevoerde) projectmedewerker kan vastgesteld worden of het om een bestaand interview gaat (dag en medewerker vormen een unieke identificatie van een interview). Als het geen bestaand interview is moet de gebruiker gevraagd worden of het inderdaad de bedoeling is een nieuw interview aan te maken.

Open de property sheets van de combobox *personnelId*. Zet het formaat voor de combobox op "fixed". Doe je dit niet, dan wordt aangenomen dat de combobox een string bevat. Dan is de waarde van `(Me!personnelId = rst!personnelId) = ("1" = 1) = False`, wat een

type conversie nodig maakt. De typeconversie van string naar integer kan gedaan worden met de functie CInt en CInt(Me!personnelId) = rst!personnelId levert wel het goede resultaat. Zorg dat "Limit to List" aanstaat (waarom?).

- Maak als Row Source van de *Date* combobox een query aan die uit tblInterviews de data van de interviews van de in PersonnelId geselecteerde medewerker filtert. Geef in het criteriumveld bij PersonnelId op:

```
[Forms]![frmStartInterview]![PersonnelId]
```



- Laat de column count van de combobox op 1 staan, zodat de PersonnelId kolom niet zichtbaar is.
- De default waarde voor de datum moet de huidige dag zijn ("Date()"). Deze waarde komt niet altijd voor in tblInterviews. Kies als formaat "Short Date" – dit formaat "verstopt" het tijdsdeel van data. Let op dat hier "Limit to List" niet aan mag staan! (waarom?)
- Maak een After-Update event handler voor de combobox *personnelId* met het volgende statement:

```
Me!Date.Requery
```

Deze requery is nodig omdat de gegevens voor de *Date* combobox veranderen als een (andere) medewerker wordt geselecteerd.

- Geef geschikte default waarden (True/False) aan de drie checkboxes (begin, during en end). Wat is hier een goede keus *vanuit het oogpunt van de gebruiker?*
- Plaats een command button cmdStartInterview op het formulier en selecteer in de Wizard het openen van frmQuestionnaire.

De eventhandler van de knop moet eerst verifiëren of een projectmedewerker, een datum en tenminste een van Begin, During en End zijn gekozen (dit laatste met b.v. If Melduring Then...)¹¹. Als dat gebeurd is moet gezocht worden in tblInterviews of er een bestaand interview is voor de geselecteerde medewerker en datum. Als dat het geval is kan het InterviewId ingevuld worden en kan frmQuestionnaire geopend worden. Als er geen interview gevonden is moet de gebruiker gevraagd worden of het inderdaad de bedoeling is dat er een nieuw interview met de geselecteerde gegevens wordt aangemaakt. Als dit het geval is, moet er een record aan tblInterviews worden toegevoegd waarna frmQuestionnaire geopend moet worden.

We komen zo tot de volgende pseudocode voor de event handler:

```
ValidData = true
If "PersonnelId is Null" Then
    "geef foutmelding "
    "ValidData = False"
End If
" soortgelijke code voor de datum en de Begin/During/End
controls"
If ValidData Then
    "Zet InterviewId"
    If "Zet InterviewId gelukt is" Then
        Open frmQuestionnaire
    End If
End If
```

De foutmelding voor een niet ingevuld PersonnelId kan gegeven worden met

```
MsgBox ("Please select a project member")
```

Als een niet bestaand interview is geselecteerd, moet de gebruiker de keus krijgen om een nieuw interview aan te maken danwel de gegevens aan te passen. Het makkelijkst is om een functie

```
Private Function setInterviewId() As Boolean
```

¹¹ De expressie Me!Begin levert alleen een Boolese waarde op als de checkbox een goede default waarde heeft gekregen!

te maken, die True retourneert als ofwel een bestaand interview is gevonden ofwel een nieuw interview is aangemaakt. In deze gevallen moet ook de InterviewId tekstbox worden ingevuld.

Pseudocode voor de functie setInterviewId:

```

setInterviewId = False
"Open tblInterviews"
Found = False
If "de tabel is niet leeg" Then
    "Ga naar het begin van de tabel"
    Do While "record niet gevonden en niet EOF van de tabel"
        Found = "record en opgegeven waarden hetzelfde"
        If Not Found Then "volgende record"
    Loop
End If
If "Not Found" Then
    "vraag gebruiker of een nieuw interview moet worden
    gemaakt"
    If "ja" Then
        "Voeg record toe en vul PersonnelId en Date in"
        "Pas de InterviewId textbox en de tabel aan (!)"
        setInterviewId = true
    End If
Else
    "Pas de InterviewId textbox aan"
    setInterviewId = true
End If

```

Het eerste gedeelte van dit stuk pseudocode zou je zonder moeite moeten kunnen implementeren. Voor het tweede gedeelte is nog wat toelichting nodig.

Als er geen bestaand interview wordt gevonden, dan moet er een messagebox getoond worden ("vraag gebruiker of een nieuw interview moet worden gemaakt"):

```
response = MsgBox("new interview for" ..... & "?", vbYesNo)
```

De variabele *response* kan gedeclareerd worden met:

```
Dim response As Integer
```

(of alleen met `Dim response`). "vbYesNo" is een een constante die aangeeft dat de gebruiker Yes of No kan kiezen. Het antwoord kan gecontroleerd worden met

```
If response = vbYes Then
```

Op de plaats van de "..... " kunnen we

```
& Me!PersonnelId.Column(1) & " on " & Me!Date
```

invullen om de gebruiker te laten zien wat hij/zij heeft ingevuld. De eerste parameter van de MsgBox functie is hier dus een aantal strings geconcateneerd met "&". De naam van de medewerker vissen we op uit de tweede kolom van de PersonnelId combobox zoals uitgelegd in 5.2.

Als de gebruiker aangeeft dat er geen nieuw interview hoeft worden aangemaakt, heeft deze zich kennelijk vergist en hoeft er verder niets te gebeuren. De gebruiker zal dan een andere medewerker of datum selecteren.

In het geval er wel een nieuw interview aangemaakt moet worden, voeg dan een nieuw record aan tblInterviews toe ("Voeg record toe en vul PersonnelId en Date in"):

```

rst.AddNew
rst!PersonnelId = Me!PersonnelId
rst!Date = Me!Date

```

Vervolgens kan het InterviewId (autonummer veld in tblInterviews) op het formulier worden ingevuld en kan de tabel aangepast worden ("Pas de InterviewId textbox en de tabel aan (!)")

```
Me!InterviewId = rst!InterviewId
rst.Update
```

Pas als rst.Update is aangeroepen wordt de tabel daadwerkelijk aangepast. Na het aanroepen van Update is het betreffende record niet meer direct toegankelijk: als Me!InterviewId = rst!InterviewId na rst.Update wordt geplaatst krijg je een run-time fout. Als het om een bestaand interview gaat moet het InterviewId ook op het formulier worden aangepast, zoals ook in de pseudocode is aangegeven!

- Voeg een knop voor dit formulier toe aan het hoofdmenu.

5.5. frmQuestionnaire

Nu moet nog het formulier om de enquête mee af te nemen gebouwd worden. Om het formulier te maken moet er eerst een query gemaakt worden die de juiste vragenset genereert op basis van de gegevens ingevuld op frmStartInterview. Ook moet er een query gemaakt worden die de meerkeuze antwoorden bij een vraag ophaalt.

- Maak om te beginnen een functie Included aan in de Evaluationfunctions module (vergeet niet te save).

```
Function Included(ToCompare As Integer, Mask As Integer) As Boolean
    Included = ToCompare And Mask
End Function
```

Deze functie kan o.a. gebruikt worden om de controleren of de rollen waarvoor een vraag gesteld moet worden voorkomen in de rollen van de geënquêteerde.

- Open qryQuestions in design mode.
- Maak in de zesde (lege) kolom met de builder in het "field" vak de uitdrukking

```
Included([tblProjectMembers]![Roles];[tblQuestions]![Roles])
```

en vul als criterium in: True. De functie Included is in de Builder onder Functions – Enquete te vinden. Access voegt hier "Expr1:" aan toe in het Field deel van de kolom. Expr1 wordt als naam van de kolom gebruikt; dit kan veranderd worden in iets zinnigers. Door deze kolom wordt gezorgd dat de rollen waarvoor de vraag gesteld moet worden overeenkomen met de rollen van het teamlid.

- Maak in de zevende kolom met de builder de uitdrukking

```
([Forms]![frmStartInterview]![begin] And [tblQuestions]![begin])
Or ([tblQuestions]![during] And
[Forms]![frmStartInterview]![during]) Or
([Forms]![frmStartInterview]![end] And [tblQuestions]![end])
```

en geef ook hier weer als criterium True op.

- Zet het vinkje voor "show" van de laatste twee kolommen uit.
- Maak nu een query qrySelectPreDefAnswers waarin de vooraf gedefinieerde antwoorden voor een bepaalde vraag worden opgehaald. Hiervoor zijn nodig in de query design grid: tabellen tblQuestions, tblAnswerIds, tblAnswerIdsToElts en tblAnswerElements. Uit tblAnswerElements zijn nodig: answerelement en value. Uit tblQuestions is QID nodig, deze moet uiteraard gelijk zijn aan het QID van de vraag die op frmQuestionnaire getoond wordt, dus vul het criterium-veld in. Deze laatste kolom moet niet getoond worden.

Open nu frmQuestionnaire in design mode.

- Verwijder met de property sheet de navigation buttons van het formulier. Disable tevens de "close Button". We moeten namelijk bij iedere vraag eventueel bestaande antwoorden opzoeken en zo nodig aanpassen – ook bij de laatste vraag! Knoppen voor deze functionaliteit moeten we dus zelf toevoegen.
- Verander op frmQuestionnaire de combobox cmbAnswer zodat deze zijn waarden ophaalt uit qrySelectPreDefAnswers. Zet de bound column op 1, Column Count op 2 en verstopt de eerste kolom door z'n breedte op nul te zetten.

De waarde van het interviewId moet makkelijk bij de hand zijn, maar je wilt de gebruiker er niet mee lastig vallen. Maak daarom de textbox interviewId onzichtbaar en geef als default waarde

```
[Forms]![frmStartInterview]![InterviewId]
```

- Maak nu de On-Current event handler voor frmQuestionnaire. Het On-Current event wordt gegenereerd zodra een nieuw record wordt geselecteerd (ook bij het openen van het formulier). De handler moet in tblAnswers zoeken naar record met dezelfde QID en InterviewId als gegeven op frmQuestionnaire. Als er al een antwoord was gegeven, dan moeten cmbAnswer en Answer de waarden uit de tabel krijgen, anders moeten ze de waarde Null krijgen. Vergeet niet te controleren op rst.EOF en rst.BOF en vergeet ook niet de tabel weer te sluiten. Voeg in de On-Current handler nog de regel

```
Me!cmbAnswer.Requery toe. In pseudocode:
```

```
If "Tabel niet leeg" Then
    "ga naar eerste record"
    Do While "geen EOF en antwoord niet gevonden"
        If "interviewId en QID van tabel en form niet
            overeenstemmen" Then
            "ga naar volgende record"
        Loop
    End If
    If "niet gevonden" Then "geef antwoord textbox en combobox
        de waarde Null"
    Else "geef textbox en combobox de waarden uit de tabel"
        "sluit tabel"
        "Requery cmbAnswer"
```

- Waarom is de requery van cmbAnswer nodig? (Probeer eventueel het effect uit)

Het enquêteformulier is nu bijna klaar – alleen de knoppen om naar de volgende of vorige vraag te gaan moeten worden toegevoegd. Deze moeten ook de functionaliteit hebben om antwoorden toe te voegen of te wijzigen.

- De procedure EnterAnswer is al opgenomen bij de code van het formulier:

```
Private Sub EnterAnswer()
Dim Found As Boolean, rst As New ADODB.Recordset
rst.Open "tblAnswers", CurrentProject.Connection,
        adOpenKeySet, adLockOptimistic
Found = False
If Not (rst.EOF And rst.BOF) Then
rst.MoveFirst
Do While Not rst.EOF And Not Found
Found = (rst!InterviewId = Me!InterviewId) And _
        (rst!QID = Me!QID)
If Not Found Then rst.MoveNext
Loop
End If
If Not Found Then
rst.AddNew
rst!QID = Me!QID
rst!InterviewId = Me!InterviewId
End If
rst!Answer = Me!Answer
rst!answerelement = Me!cmbAnswer
rst.Update
rst.Close
End Sub
```

Als een antwoord op een vraag niet eerder is gegeven wordt er met rst.AddNew een nieuw record aangemaakt, als er wel een antwoord gevonden wordt, wordt het antwoord aangepast. Daarna worden met rst.Update de veranderingen permanent gemaakt.

Om de navigatieknoppen te maken, kun je de command button wizard gebruiken (Record Navigation – Goto Next/Previous Record).

- Maak met de Wizard ook een "Quit" knop om het laatste antwoord nog netjes af te werken (Form Operations – Close Form). Verander vervolgens de code die de wizard heeft aangemaakt: voeg een aanroep "Call EnterAnswer" toe voor DoCmd.Close resp. voor DoCmd.GoToRecord , , acPrevious/acNext .

6. Beveiliging

Doel: na afloop is de databasetoepassing beveiligd op basis van autorisaties.

Voor je aan dit hoofdstuk begint is het aan te raden een kopie van je database te maken. Het komt namelijk nogal eens voor, dat na het aanbrengen van beveiliging het password vergeten wordt en dat het bestand dat Access aanmaakt om in geval van problemen de database te kunnen herstellen, per ongeluk is weggegooid. Het opnieuw toevoegen van security kan met weinig tijd opnieuw gedaan worden, maar een ontoegankelijke database is waardeloos. Access maakt nog wel een reservekopie aan, maar neem hier het zekere voor het onzekere.

In deze applicatie moet een enquêteur alleen de vragen kunnen stellen zonder dat hij of zij de database structuren kan veranderen. Dit kunnen je voor elkaar krijgen door de database te beveiligen. Met de user-level security wizard is dit vrij eenvoudig te doen. Maar er zijn ook een paar andere mogelijkheden waardoor je het al moeilijk maakt voor de meeste gebruikers om ongewenste veranderingen aan te brengen. Zo is het heel eenvoudig om het database venster te verstoppen:

- Kies Startup uit het tools menu. Hiermee kun je ook de menu's en werkbalken verstoppen¹², een titel geven aan je applicatie en een eigen icoon opgeven. Niet onbelangrijk: je kunt ook opgeven welk database object geopend wordt als de applicatie start. Kies frmMainMenu om te openen bij het opstarten.

Met de "Advanced" optie kun je ook verhinderen dat de Access special keys werken. Op het eerste gezicht lijkt je een probleem te hebben omdat je geen menu meer hebt om de opstart opties te veranderen. Als je daarna echter opstart terwijl je de shift toets ingedrukt houdt, start Access op de normale manier op. Na het aanbrengen van de beveiliging moet je de shift toets ingedrukt houden na het invoeren van je gebruikersnaam en password. Er zijn relatief weinig mensen die dit weten, dus dit geeft al een redelijke beveiliging. Probeer e.e.a. uit.

Een serieuzer beveiliging kan bereikt worden met het instellen van user-level security. Hiermee kunnen voor iedere gebruiker of groep van gebruikers de juiste toegangsrechten voor ieder database object gezet worden. Tevens kunnen op basis van gebruikersnaam of groep controls, forms en andere database objecten (on)zichtbaar gemaakt worden m.b.v. de Visible eigenschap.

- Start, om de beveiliging in te stellen, de user-level security wizard op vanaf het tools – security menu. De wizard vraagt of je een nieuwe workgroup information file wilt aanmaken. Kies deze optie. Op het volgende scherm van de wizard wordt een naam voor de workgroup informatie file gevraagd, een WID en je naam en bedrijf. Onderaan het scherm moet je opgeven of dit security bestand het default bestand wordt, dan wel dat je een shortcut naar de beveiligde database op de desktop wilt hebben.

Voor het WID is al een gegenereerde string ingevuld – deze string heb je nodig als je security bestand ooit corrupt raakt en je wilt het opnieuw genereren. Je kunt deze string dus laten staan of vervangen door een die makkelijker te onthouden is. De wizard zorgt aan het eind voor een bestand waarin je alle benodigde informatie kunt vinden. Wat betreft de keuze voor de shortcut of het default security bestand: zonder default security bestand kun je de database niet zonder meer openen door erop te dubbelklikken. Bij het openen moet je opgeven waar de security informatie staat. Dit is natuurlijk niet nodig als je een default bestand gebruikt. Een voorbeeld van de shortcut die je nodig hebt om zonder default security bestand de beveiligde database te openen is:

```
"C:\Program Files\Microsoft Office\Office\MSACCESS.EXE"  
"D:\OGO\evaluation.mdb" /WRKGRP "D:\OGO\Secured.mdw"
```

- Kies in dit geval voor de shortcut; deze wordt op de desktop geplaatst.

¹² Er blijft wel een menubalk met sterk gereduceerde functionaliteit

- Het volgende scherm vraagt welke bestanden beveiligd moeten worden, default is alles geselecteerd. Dat kun je zo laten. Het volgende scherm vraagt welke groepen je wilt onderscheiden. Dit zijn groepen met standaard permissies. Kies er geen, verderop maak je je eigen groep aan voor de enquêteurs. Het volgende scherm vraagt om de permissies voor 'Users'. Ken deze geen rechten toe – iedereen is een user, dus permissies aan de users groep geven verzwakt de beveiliging. Vervolgens wordt een scherm getoond waarop je eigen usercode al staat. Selecteer je usercode en geef een password op. Voeg nog geen gebruiker toe – je hebt nog geen groep waar deze in moet. Op het volgende scherm wordt weergegeven in welke groepen de ingevoerde gebruikers zitten. Daar staat nu alleen je usercode in de Admins groep. Op het laatste scherm wordt een naam voor de backup van de oorspronkelijke onbeveiligde database gevraagd. De database opent nu en een rapport met alle beveiligingsinformatie staat open. Als je dit wegklikt krijg je een waarschuwing dat dit informatie is die je nodig hebt en dat Access het als een snapshot file kan wegschrijven. Doe dit. Sluit de database en open de database via de shortcut.
- Maak na het inloggen eerst een groep enqueteurs aan met Tools- security – User and Group Accounts, tabblad Groups. Voer ook een ID in naar keuze. Maak vervolgens een gebruiker enqueteur aan op het Users tabblad. Maak enqueteur lid van de enqueteurs groep. Vergeet niet later de database als enqueteur te openen (blanco password) en via het Change Logon Password tabblad een password te zetten.
- Zet nu de permissies voor de enqueteurs groep met Tools- Security- User and Group Permissions.

Voor de database heeft de enqueteur open/run permissie nodig. Verder heeft hij read data permissie op de tabellen nodig en Read, Update, Insert en Delete permissie op tblAnswers, tblProjectMembers en tblInterviews. Geef enqueteur open/run permissie op frmMainMenu, frmStartInterview, frmQuestionnaire en frmEnterProjectMembers en Read Data op de queries.

- Om knoppen die een enquêteur niet kan gebruiken onzichtbaar te maken kun je de volgende event handler voor het hoofdmenu gebruiken:

```
Private Sub Form_Open(Cancel As Integer)
    Dim cat As New ADOX.Catalog, usr As User, grp As Group
    Me!cmdAnswersToElements.Visible = False
    cat.ActiveConnection = CurrentProject.Connection
    With cat
        For Each usr In .Users
            If usr.Name = CurrentUser Then
                For Each grp In usr.Groups
                    If grp.Name = "Admins" Then
                        Me!cmdAnswersToElements.Visible = True
                    End If
                Next grp
            End If
        Next usr
    End With
End Sub
```

Door aan de ActiveConnection property een geldige open connectie toe te kennen kan het Catalog ADOX (ADO Extensions for Data Definition Language and Security) object geopend worden. Vanuit de geopende Catalog kun je de objecten daarbinnen benaderen.

Maak ook de knop om vragen in te voeren onzichtbaar voor de enquêteur. In dit stuk code worden twee constructies gebruikt die in Bijlage C niet behandeld zijn. Door gebruik van het With cat statement kan cat.Users afgekort worden tot .Users. Het gebruik van For Each zorgt ervoor dat de buitenste lus uitgevoerd wordt voor alle users in de cat.Users verzameling.

Figuur 5 toont de ADO en ADOX hiërarchieën. In ADOX is Catalog het topniveau object. Onder Catalog vinden we verschillende objecten waaronder Groups en Users. Binnen een sessie kunnen verschillende databases geopend zijn. Voor iedere user wordt bijgehouden in welke groepen deze zit in het attribuut usr.Groups en voor iedere groep worden de users

bijgehouden in het attribuut grp.Users. De naam van de huidige gebruiker is op te vragen met CurrentUser.

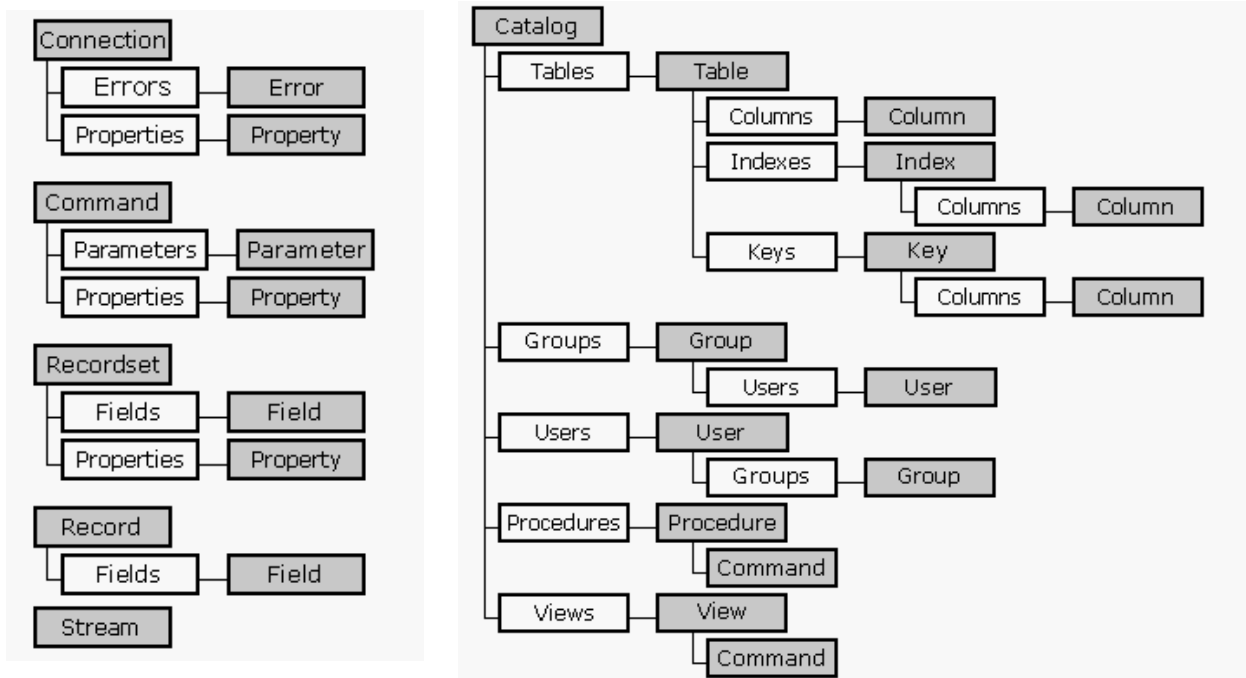
Twee eenvoudige manieren om erachter te komen of de gebruiker administrator rechten heeft (de VBA help is hierover nogal onduidelijk):

```

For Each grp In .Users(CurrentUser).Groups
    If grp.Name = "Admins" Then MsgBox ("Is Admin")
Next grp

For Each usr In .Groups("Admins").Users
    If usr.Name = CurrentUser Then MsgBox ("Is Admin")
Next usr

```



Figuur 5 ADO en ADOX object hiërarchieën

7. Database wizard

Doel: na afloop is duidelijk hoe met de database template een databasetoepassing met een schakelpaneelmenu gegenereerd en aangepast kan worden.

Bij het opstarten biedt Access de keuze uit het openen van een bestaande database of een nieuwe database te creëren. Kies je voor de laatste optie, dan krijg je de keuze tussen een blanco database en een database template. Met de laatste optie kunnen een aantal voorbeeld-databases geconstrueerd worden. Deze databases kunnen een handig uitgangspunt bieden voor een nieuwe applicatie. We zouden bijvoorbeeld de "Order Entry" database template (beschikbaar "On my Computer") kunnen gebruiken als basis voor de implementatie van de Zorgvliet Ziekenhuis database-applicatie. De gegenereerde applicaties zijn alle gestructureerd rond een centraal formulier, het "switchboard" (schakelpaneel) vanwaaruit alle schermen in de applicatie te bereiken zijn. Daarom volgt hier een korte uitleg van de werking van het "switchboard" van deze applicatie. Door gegenereerde applicaties te bekijken kun je leren hoe in Access verschillende taken gerealiseerd kunnen worden. Een database met uitsluitend het switchboard formulier en de bijbehorende tabel uit de "Order Entry" database staat bij het overige Access materiaal in <http://www.win.tue.nl/~sidorova/ogo/Access>. Bekijk eerst hoe het switchboard werkt en bekijk de tabel in datasheet en design view en het formulier in design mode.

Het switchboard is een formulier met een aantal knoppen, die verschillende functies kunnen hebben. Zo'n functie wordt in een centrale tabel vastgelegd door een commando nummer en eventuele argumenten. Een van de commando's is het omschakelen naar een ander switchboard. Ook is het mogelijk een ander switchboard het "main" switchboard te maken. Dit laatste commando wordt niet uitgelegd, evenmin als de meeste andere commando's. Uit de gedeelten van de code van deze applicatie op de volgende bladzijden is het merendeel van de mogelijke commando's weggelaten evenals alle foutafhandeling. Hier en daar is voor de opmaak een return in de code gezet, pas dus op met kopiëren uit dit document

Het switchboard formulier heeft 8 Command Buttons: Option1 tot 8. Daarvan worden een aantal zichtbaar gemaakt afhankelijk van het aantal records in de tabel die horen bij een switchboard.

Bij het openen van het formulier wordt eerst de Form_Open handler uitgevoerd.

```
Private Sub Form_Open(Cancel As Integer)
    DoCmd.SelectObject acForm, "Switchboard", True
    ' Move to the switchboard page that is marked as the default.
    Me.Filter = "[ItemNumber] = 0 AND [Argument] = 'Default' "
    Me.FilterOn = True
End Sub
```

De handler roept de SelectObject methode van het DoCmd object aan. Dit geeft het formulier de focus en maakt latere acties als GoToControl mogelijk. Vervolgens wordt de Filter eigenschap van het formulier gezet en toegepast met Me.FilterOn. Na Me.FilterOn is precies een record uit de tabel geselecteerd, n.l. die van het main switchboard.

Vervolgens wordt de On_Current handler van het formulier aangeroepen. In dit geval komt dat doordat dit de normale volgorde van aanroepen van handlers is bij het openen van een formulier (zie: sectie 0).

```
Private Sub Form_Current()
    Me.Caption = Nz(Me![ItemText], "")
    FillOptions
End Sub
```

De handler zet de tekst van de title bar van het formulier gelijk aan het ItemText veld uit de tabel. Als ItemText Null is, retourneert de Nz functie een lege string. Vervolgens wordt FillOptions aangeroepen.

```
Private Sub FillOptions()
    Const conNumButtons = 8
    Dim con As Object, rs As Object, stSql As String
    Dim intOption As Integer
    Me![Option1].SetFocus
    For intOption = 2 To conNumButtons
        Me("Option" & intOption).Visible = False
        Me("OptionLabel" & intOption).Visible = False
    Next intOption
End Sub
```

In dit eerste stukje code van FillOptions wordt de focus op de eerste knop gezet; die is dus altijd zichtbaar. Een control die de focus heeft kan niet onzichtbaar worden gemaakt. Let even op de manier waarop de string "option" en de integer intOption worden geconcateneerd om samen de naam van de control te vormen.

```
Set con = Application.CurrentProject.Connection
stSql = "SELECT * FROM [Switchboard Items]"
stSql = stSql & "WHERE [ItemNumber] > 0 AND [SwitchboardID]=" & _
    Me![SwitchboardID]
stSql = stSql & " ORDER BY [ItemNumber];"
Set rs = CreateObject("ADODB.Recordset")
rs.Open stSql, con, 1 ' 1 = adOpenKeyset
```

Hier wordt een query uitgevoerd om de items van het switchboard op te halen. Hiervoor wordt eerst de connectie die bij de huidige Access applicatie hoort opgehaald. Daarna wordt de SQL query opgebouwd, waarmee de items worden opgehaald. Het resultaat wordt in een Object rs (een recordset) opgeslagen. Zoals je ziet zijn er meerdere manieren om met recordsets te werken.

```
While (Not (rs.EOF))
    Me("Option" & rs![ItemNumber]).Visible = True
    Me("OptionLabel" & rs![ItemNumber]).Visible = True
    Me("OptionLabel" & rs![ItemNumber]).Caption =rs![ItemText]
    rs.MoveNext
Wend
rs.Close
Set rs = Nothing
Set con = Nothing
```

Voor ieder record dat in de tabel gevonden wordt, worden een knop en een label zichtbaar gemaakt. Ook hier worden de namen van de controls weer op creatieve wijze gevormd. Tenslotte worden de recordset en de database (via de connectie) gesloten.

Op de Event property sheet van knop nummer i is als On-Click handler opgegeven: = HandleButtonClick(i). In de volledige applicatie die door de Wizard wordt gegenereerd staan

meer commando's dan hieronder weergegeven. Om het geheel hier wat overzichtelijk te houden, zijn de meeste commando's weggelaten. Hoe deze commando's precies werken is echter makkelijk met de VBA help te achterhalen.

```
Private Function HandleButtonClick(intBtn As Integer)
Const conCmdGotoSwitchboard = 1, conCmdExitApplication = 6
Dim con As Object, rs As Object, stSql As String
Set con = Application.CurrentProject.Connection
Set rs = CreateObject("ADODB.Recordset")
stSql = "SELECT * FROM [Switchboard Items] "
stSql = stSql & "WHERE [SwitchboardID]=" & Me![SwitchboardID] & _
" AND [ItemNumber]=" & intBtn
rs.Open stSql, con, 1 ' 1 = adOpenKeyset
```

Op dezelfde manier als boven wordt hier een query gedaan op de tabel om het uit te voeren commando op te halen.

```
Select Case rs![Command]
Case conCmdGotoSwitchboard
Me.Filter = "[ItemNumber] = 0 AND _
[SwitchboardID]=" & rs![Argument]
```

Bij het commando om een ander switchboard te openen wordt het filter van het formulier aangepast. Dit betekent dat een ander record current wordt. Na afloop van de On-Click handler van de knop wordt dus de On_Current handler van het formulier opnieuw uitgevoerd en de knoppen voor het nieuwe switchboard worden op het formulier gezet. Dit is het centrale mechanisme van het switchboard.

```
Case conCmdExitApplication
CloseCurrentDatabase
Case Else
MsgBox "Unknown option."
End Select
```

Het laatste stuk spreekt voor zich.

Bijlage A. De Toolbox

Doel: kennismaking met de gereedschapskist van de formulier-ontwerper.



Het formulier is nu geopend in design mode en er is een toolbox verschenen waarop de verschillende controls staan die op het formulier geplaatst kunnen worden. Is de werkbalk niet verschenen, dan kan deze geopend worden door de Toolbox-icoon te selecteren uit de formulierontwerp-werkbalk, of door in het menu-onderdeel "View" de Toolbox-icoon aan te klikken.

De standaard beschikbare controls zijn van boven naar beneden (de meeste ervan worden verder niet besproken) :



- Object selector – als deze is ingedrukt kun je controls op het formulier selecteren, b.v. om die te wijzigen
- Control Wizards – als deze is ingedrukt starten voor bepaalde controls Wizards op
- Label
- Textbox – wordt altijd samen met een label geplaatst (de enige control die we tot nu toe op een formulier hebben gezien)
- Option group – om check boxen, radio buttons en toggle buttons te groeperen (bij de laatste 2 kan slechts 1 "aan" zijn; deze bepaalt de waarde van de groep)
- Toggle button
- Radio button
- Check box
- Combo box
- List box
- Command button
- Plaatje
- Object frame
- Bound object frame
- Page break
- Tab Control – maakt een control met een aantal tabbladen aan
- Subform – hiermee kan een formulier gebaseerd op een andere tabel op het formulier geplaatst worden
- Lijn – puur voor opmaak
- Rechthoek – dito
- "More Controls" – geeft een lange lijst van allerlei andere controls zoals ActiveX componenten

Het type van een component wordt zichtbaar gemaakt door deze met de cursor aan te wijzen. Als je de cursor boven een control houdt, verschijnt na korte tijd de naam van de control in een klein venster naast de cursor.

Bijlage B. Property sheets

Doel: kennismaken met eigenschappen van componenten

In deze sectie worden de voor ons belangrijkste eigenschappen van formulieren, textboxes en comboboxen besproken die met de property sheets kunnen worden ingesteld. Een aantal erg voor de hand liggende eigenschappen wordt niet genoemd. Bijna al deze eigenschappen kunnen overigens ook met Visual Basic for Applications (VBA) worden geprogrammeerd. Dat maakt het mogelijk om tijdens de executie van de applicatie b.v. knoppen te laten verschijnen en verdwijnen.

Er zijn altijd vijf property sheets. Op het eerste tabblad, Format, kunnen eigenschappen worden ingesteld die met de layout te maken hebben, zoals voor- en achtergrond kleuren, fonts, numerieke formaten en afmetingen. Op het tweede tabblad, Data, vind je alles dat met de data van het formulier of control te maken heeft, zoals 'Record Locks' van het formulier om records te kunnen locken op veld, record of tabel/query niveau en 'Enabled' van een textbox of combobox om de control read-only te maken. Op het 'Event' tabblad kunnen event handlers opgegeven worden: deze worden uitgevoerd als een bepaald event plaatsvindt, zoals het event 'Open' van een formulier dat optreedt als het formulier wordt geopend en het 'Get Focus' event van een control dat optreedt als met de muis op de control wordt geklikt. Op het 'Other' tabblad zijn nog wat andere eigenschappen in te stellen, b.v. voor het formulier de 'Cycle' eigenschap die bepaalt hoe het formulier reageert als je met de tab toets van control naar control springt en de 'Control Tip Text' waarmee je een tekst voor een 'screentip' kunt opgeven (verschijnt als je de cursor boven de control positioneert, zoals bij de controls op de Toolbox werkbalk). Op het 'All' tabblad staan eenvoudig alle eigenschappen van de eerste vier tabbladen samen.

a. Formulieren

Doel: enig overzicht geven van de verschillende keuzemogelijkheden van de verschillende eigenschappen van formulieren.

Format tabblad

Caption	De tekst die in de 'title bar' van het formulier komt
Default view	'Single form' : de representatie van het formulier zoals je die tot nu toe gezien hebt, 'Continuous forms': achter elkaar geplaatste 'single forms' voor elk record, 'Datasheet': als een tabel, zoals de subform van Samen
Scroll Bars	Bepaalt of er scroll bars aan het formulier zitten
Navigation Buttons	De knopjes onderaan het formulier om door de records heen te lopen en nieuwe records mee aan te maken.
Control Box	Het menu dat verschijnt als je links bovenin de title bar klikt
Min Max Buttons	De knoppen rechts bovenin om het formulier te minimaliseren of te maximaliseren
Close Button	Het kruis rechts bovenin om het formulier te sluiten
Picture	Plaatje voor de achtergrond van het formulier

De achtergrondkleur van het formulier stel je in op de property sheet van de 'detail' sectie van het formulier

Data tabblad

Record Source	de onderliggende tabel
Data Entry	No: toont alle records van de tabel of query met mogelijkheid om data toe te voegen, Yes: het formulier kan alleen gebruikt worden om records toe te voegen
Record Locks	Record locking op veld, record of tabel/query niveau

Event tabblad

Hier staan alle event handlers die gedefinieerd kunnen worden. De handler voor het On Current event (treedt op als een ander record wordt geselecteerd) is vaak handig. Door te klikken bij een bepaalde handler en vervolgens de F1 toets in te drukken krijg je voldoende informatie om te zien of de handler geschikt is voor het doel dat je wilt bereiken. Dit geldt overigens ook voor de andere eigenschappen.

De volgorde van de events bij openen en sluiten:

- Open - Load - Resize - Activate - Current
- Unload - Deactivate – Close

Other tabblad

Modal	Als het formulier modal gemaakt wordt, kan er in de rest van de applicatie niets meer gebeuren totdat dit formulier gesloten wordt (probeer uit!)
Cycle	Bepaalt hoe het formulier zich gedraagt als je met de tab knop tussen controls springt
Menu Bar	Hier kun je een menu bar voor het formulier opgeven

b. Controls

Doel: enig overzicht geven van de verschillende keuzemogelijkheden van de verschillende eigenschappen van controls.

Format Tabblad

Format	Voor textboxes en comboboxen legt dit het formaat van een veld vast. Als de control niet aan een veld van een tabel/query is gekoppeld kan hier het type van het veld vastgelegd worden (default: string). Als het veld wel is gekoppeld, kun je hier de representatie vastleggen, b.v. voor een numeriek veld de wetenschappelijke notatie of een percentage
Visible	Bepaalt of de control (en het bijbehorende label) zichtbaar is. Het is soms nodig om een waarde vast te leggen in een control zodat je toegang hebt tot die waarde vanuit VBA of vanuit een query. Door de control te verbergen val je de gebruiker er niet mee lastig. Het kan ook zijn dat je een volgorde van invullen van velden wilt afdwingen. In dat geval maak je een control eerst onzichtbaar en zodra b.v. een bepaalde textbox is ingevuld maak je de control zichtbaar met de After-Update handler van de textbox.
Column Count	(combobox) Het aantal kolommen van de control
Column Widths	(combobox) De breedte van de kolommen. Met een breedte van 0 verstop je een kolom
List Rows	(combobox) Het aantal rijen dat onder elkaar getoond wordt
List Width	(combobox) Breedte van de lijst

Behalve een aantal voor de hand liggende eigenschappen zijn de overige eigenschappen puur voor de opmaak. 'Back Color' (de achtergrondkleur van de control) is niet beschikbaar voor de command button – die komt alleen in standaard MS grijs. Een slimmerikje kan natuurlijk wel een manier verzinnen om toch een gekleurde knop te krijgen...

Data tabblad

Control Source	Het veld van de tabel waaraan de control is gekoppeld
Input Mask	Dwingt de gebruiker data in een bepaald formaat in te voeren, b.v. (000) 000 dwingt input als (040) 123 af
Default Value	Spreekt voor zich
Validation Rule	Maakt controle op de ingevoerde waarde mogelijk

Enabled	Control kan read-only gemaakt worden doordat het niet de focus kan krijgen
Locked	Control kan read-only gemaakt worden
Row Source Type	(combobox) Keuze uit Table/Query, Value list (in dat geval wordt bij Row Source een reeks waarden ingetikt), Field list (een reeks namen van velden van een tabel of query)
Row Source	(combobox) Naam van de tabel of query, een SQL statement, de waarden van de value list of van de field list
Bound Column	(combobox) Welke kolom van de combobox gebonden wordt aan de waarde van de control en/of aan de control source
Limit To List	(combobox) Bepaalt of er alleen waarden uit de drop-down lijst gekozen mogen worden

Event tabblad

De event handlers spreken voor zich. De After-Update handlers voor de textbox en de combobox en de On-Click handler van de command button gebruik je in de praktijk het meest. Voor de combobox is er nog een On-Not-in-List handler, die je zou kunnen gebruiken om voor een niet-standaard keuze nadere details op te vragen.

Other tabblad

De meeste van deze eigenschappen zijn aan de orde geweest. De belangrijkste is de naam. Tab Index kun je gebruiken om de volgorde vast te leggen waarin je de controls afgaat met de tab toets. Tab stop bepaalt of de control via tabs te bereiken is.

Bijlage C. Programmeren in Visual Basic for Applications

Doel: een kort overzicht geven van de programmeeromgeving van Access.

Dit hoofdstuk geeft een vereenvoudigde syntax van Visual Basic for Applications (VBA), net voldoende om de oefeningen te kunnen doen en de tekst te kunnen volgen. Nadere details kunnen in de VBA help worden opgezocht. Aangezien de betekenis van de statements niet wezenlijk verschilt van die in andere programmeertalen, wordt geen nadere uitleg gegeven. Voorbeelden van verschillende statements zijn in de code fragmenten verderop in de tekst te vinden. Een aantal statements in VBA die specifiek voor Access zijn, b.v. om een tabel te openen en te doorlopen, worden hier wat uitgebreider behandeld.

Aan een bepaalde control (b.v. een knop, tekstveld of label) kunnen in VBA *event procedures* worden gekoppeld voor verschillende events. In veel gevallen is de Command Button Wizard voldoende om de gewenste functionaliteit te realiseren, b.v. als er een formulier moet worden geopend. De Command Button Wizard creëert een procedure volgens een standaard patroon (ga dit na). In andere gevallen is het nodig zelf een stukje VBA code te schrijven en dit aan de control te koppelen.

De VBA editor kan geopend worden door op de property sheets van een control of form in het vak achter een van de event handlers te klikken en vervolgens op de stippeltjesknop te drukken en 'Code Builder' te selecteren, of via de Code knop van de Access werkbalk .



a. Visual Basic

Doel: Korte bespreking van een aantal mogelijkheden van de programmeertaal Visual Basic.

VBA is niet case-sensitive. Een commentaarregel in VBA begint met een ' en beslaat de rest van de regel. Het commentaar hoeft niet vooraan een regel te beginnen, maar kan ook na een statement gegeven worden. VBA kent geen expliciete statement separator: een nieuwe regel begint een nieuw statement. Statements kunnen vervolgd worden op een volgende regel door de eerste regel te eindigen met een spatie en een underscore.

Een procedure in VBA heeft de volgende vorm (indentatie is niet verplicht, wel aan te bevelen):

```
Sub name [(arglist)]
    [statements]
[Exit Sub]
    [statements]
End Sub
```

Het keyword *Sub* kan voorafgegaan worden door *public*, *private*, *friend* of *static*. De eerste drie keywords hebben betrekking op de mogelijkheid om de procedure vanuit andere modules aan te roepen. In een static sub behouden de lokale variabelen hun waarden tussen aanroepen van de procedure. M.b.v. het *Exit Sub* statement kan voor het standaard einde van de procedure uit de code gesprongen worden.

Een functie heeft de volgende vorm:

```
Function name [(arglist)] [As type]
    [statements]
    [name = expression]
[Exit Function]
    [statements]
    [name = expression]
```

End Function

De functieheader kan voorafgegaan worden door dezelfde keywords als een procedure. Na het keyword `As` wordt het returntype opgegeven. De structuur van de functie is verder hetzelfde als die van een procedure behalve dat de returnwaarde moet worden gezet.

Variabelen worden gedeclareerd m.b.v. het `Dim` statement:

```
Dim varname([[subscripts]]) [As type]
[,varname([[subscripts]]) [As type]]...
```

VBA kent de datatypen `Byte`, `Boolean`, `Integer`, `Long`, `Currency`, `Single`, `Double`, `Date`, `String` (voor strings van variabele lengte), `String * length` (voor strings met vaste lengte), `Variant`, zelfgedefinieerde typen (typen gedefinieerd met het `Type` statement), en het `Object` datatype.

Als het `Dim` statement refereert aan een bestaand *object*, dan moet voor het gebruik van het object het `Set` statement gebruikt worden. In ons geval is dat als we de database willen gebruiken.

```
Set objectvar = objectexpression
```

In hoofdstuk 8 zien we de volgende statements in de `HandleButtonClick` methode om de connection op te vragen en om een `Recordset` te creëren:

```
Set con = Application.CurrentProject.Connection
Set rs = CreateObject("ADODB.Recordset")
```

Door `Set objectvar = Nothing` wordt de associatie van de variabele met het object verbroken en wordt de geheugenruimte vrijgegeven die met het object geassocieerd is als de variabele de laatste is die met het object was geassocieerd. Een voorbeeld is eveneens te vinden in het hoofdstuk over de `Order Entry Database Wizard` (hoofdstuk 7).

Er zijn twee vormen van het `If`-statement – de eenvoudige vorm (past op 1 regel):

```
If condition Then [statements] [Else elsestatements]
```

En de blok vorm:

```
If condition Then
    [statements]
[ElseIf condition-n Then
    [elseifstatements] ...
[Else
    [elsestatements]]
End If
```

Naast het `If`-statement is er ook een `Case`-statement:

```
Select Case testexpression
    [Case expressionlist-n
        [statements-n]] ...
[Case Else
    [elsestatements]]
End Select
```

Het `Case`-statement wordt in deze cursus niet gebruikt; nadere informatie is in de Help te vinden.

Er zijn verschillende mogelijkheden om lussen te programmeren. Een lus met de eindigingsconditie aan het begin van de lus:

```
Do [{While | Until} condition]
    [statements]
[Exit Do]
    [statements]
Loop
```

Of met de eindigingsconditie aan het eind van de lus:

```
Do
    [statements]
[Exit Do]
    [statements]
Loop [{While | Until} condition]
```

Er is uiteraard ook een for-statement:

```
For counter = start To end [Step step]
    [statements]
[Exit For]
    [statements]
Next [counter]
```

Een tweede variant van dit statement is het **For Each** <element> **In** <Groep>. Een voorbeeld hiervan is in hoofdstuk 6 te vinden.

Met het With statement kunnen statements wat korter worden opgeschreven. Ook van het With statement is een voorbeeld in hoofdstuk 6 te vinden, samen met een korte toelichting.

b. Access specifieke uitbreidingen

Doel: een korte bespreking van een aantal Access-specifieke uitbreidingen van Visual Basic.

Er zijn zeer veel Access specifieke uitbreidingen. Access definieert een groot aantal objecten en objecthiërarchieën om de database en operaties op de database te representeren. In dit document kan daarvan maar een fractie worden behandeld. Verderop in dit document komen een aantal van deze objecten en objecthiërarchieën ter sprake, zoals het Workspace object. Over deze objecten wordt ter plekke een (zeer) korte uitleg gegeven; meer details zijn in de Access help en in online bronnen zoals de Microsoft Knowledge Base ([http://support.microsoft.com/?scid=fh;\[ln\];kbhowto](http://support.microsoft.com/?scid=fh;[ln];kbhowto)) en Microsoft Developer Network (<http://msdn.microsoft.com/library/default.asp>) te vinden.

Het is vaak nodig om in een tabel of in de resultaten van een query te zoeken. Hierboven is al getoond hoe recordset-variabelen gedeclareerd moeten worden en geopend moeten worden. Om de recordset te doorzoeken zijn een aantal functies beschikbaar. Neem aan dat de tabel, die doorzocht moet worden bestaat uit twee velden: `veld1` en `veld2`. Voor de naam van de variabele die de recordset representeert wordt net als boven `rst` gebruikt. Velden van een recordset (of controls van formulieren) kunnen worden opgevraagd door de naam van de variabele die het samengestelde object representeert (`rst`) te laten volgen door een uitroepteken en de naam van het veld. Zo geeft `rst!veld1` de waarde van `veld1` in het huidige record.

Vaak is het zo dat gezocht wordt naar een waarde die de gebruiker op het formulier heeft ingevuld in b.v. een *textbox*. Het huidige formulier (dwz het formulier van waaruit de VBA procedure werd opgestart) kan altijd aangeduid worden met *Me*. Dus `Me!gezocht` geeft de waarde van de *textbox* *gezocht*. Als een formulier op een bepaalde tabel of query is gebaseerd kan `Me!<veldnaam>` gebruikt worden, ook als er geen control op het formulier aanwezig is

voor <veldnaam>. De waarde van Me!<veldnaam> is dan eenvoudig de waarde van <veldnaam> in het huidige record.

Bij zoeken in een recordset is het altijd van belang om te controleren of de recordset niet leeg is en om te zien wanneer het begin of eind van de recordset is bereikt. Dit wordt ondersteund door de functies EOF en BOF. Om een functie aan te roepen die voor een bepaald object beschikbaar is, wordt de naam van de functie gebruikt, voorafgegaan door een punt en de naam van de objectvariabele, b.v. rst.EOF. Bij het intikken van de punt na rst geeft VBA een lijst van functies en attributen die gebruikt kunnen worden voor het betreffende objecttype.

Voor het doorlopen van de recordset zijn o.a. de functies MoveFirst, MoveLast, MoveNext en MovePrevious beschikbaar (met evidente betekenis).

De waarde van het huidige record kan eenvoudig veranderd worden door een toekenning aan een veld. De verandering wordt echter pas definitief als daarna de functie Update wordt aangeroepen of als naar een ander record wordt gegaan met b.v. MoveNext. Een nieuw record kan worden toegevoegd met de functie AddNew en een record kan worden weggegooid met de functie Delete. Ook hier geldt dat de veranderingen pas permanent worden gemaakt als de Update functie aangeroepen wordt.

Een voorbeeld: de tabel *tbl* is geopend zoals boven en we zoeken naar een waarde voor *veld1* die gegeven wordt door de textbox *gezocht*. Wordt zo'n record gevonden, dan krijgt *veld2* de waarde uit de textbox *waarde*. Zo niet, dan wordt een nieuw record toegevoegd. Neem aan dat ook een variabele *found* van het type Boolean is gedeclareerd.

```

found = False
If Not (rst.EOF And rst.BOF) Then ' tabel niet leeg!
    rst.MoveFirst
    Do While Not rst.EOF And Not found
        found = (rst!veld1 = Me!gezocht)
        If not found Then rst.MoveNext
    Loop
End If
If Not found Then
    rst.AddNew ' voeg record toe
    rst!veld1 = Me!gezocht
End If
rst!veld2 = Me!waarde
rst.Update ' maak veranderingen permanent
rst.Close ' sluit de tabel (!)

```

Twee opmerkingen:

De MoveNext instructie wordt niet meer uitgevoerd als het gezochte record gevonden is, zodat na de lus het gezochte record het 'current' record is, tenzij het gezochte record er niet is.

Als je de MoveNext instructie vergeet raakt de applicatie in een oneindige lus, en moet Access b.v. met de TaskManager worden afgebroken.

De VBA code in de rest van dit document zal alleen goed werken als de juiste bibliotheken in de juiste volgorde aan Access zijn gekoppeld. Normaal gesproken is dit het geval, maar als bepaalde bibliotheken ontbreken, kunnen bijvoorbeeld de functies om de database mee te manipuleren niet gevonden worden. De code veronderstelt ADODB; wanneer je per ongeluk de DAO bibliotheek gebruikt werkt het ook niet.



De bibliotheken zijn alleen op te geven, als de VBA editor geopend is. De VBA editor is bijvoorbeeld te openen, door op de "code"-knop te drukken. Deze is o.a. beschikbaar in de ontwerpmodus van een formulier.

- Selecteer in het VBA window “Tools > References” (of “Extra > Verwijzingen”). Je ziet nu een lijst bibliotheken waarvan de geselecteerde aangevinkt zijn; de volgorde is met de pijlknoppen in te stellen. Er zijn zes bibliotheken nodig in de volgorde:

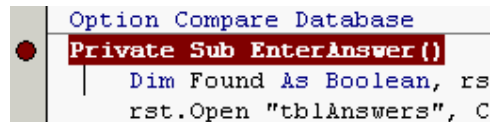
1. Visual Basic for Applications
2. Microsoft Access 9.0 Object Library
3. Microsoft ActiveX Data Objects 2.7 Library
4. Microsoft ADO Ext. 2.7 for DDL and Security
5. OLE Automation
6. Microsoft Visual Basic for Applications Extensibility 5.3.

N.B. Wanneer je zowel DAO 3.6 en ADO aangevinkt wil hebben moet je prefixen als DAO en ADODB gebruiken.

- Opdracht: creëer de genoemde tabel (geef `ve1d1` de primaire sleutel-eigenschap), creëer een formulier met twee textboxes en een knop (zet de wizard uit) waar je bovenstaande code aan verbindt en probeer de code uit.

Om problemen in je code op te sporen is de debugger uiterst handig, bijvoorbeeld wanneer je de haakjes om `rst.EOF` and `rst.BOF` vergeten bent. Je krijgt dan, wanneer je het antwoord op een bestaande vraag wil veranderen een primary key conflict: found blijft dan false waardoor je een nieuw record met de bestaande sleutel gaat aanmaken en dit probeert toe te voegen.

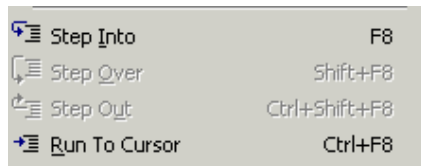
Door in de grijze balk links van de code te klikken met de select knop zet je een ‘breakpoint’. In het voorbeeld hiernaast stopt de uitvoering van de code bij de aanroep van de procedure `EnterAnswer`. Vanaf dat punt kun je b.v. de code regel voor regel laten uitvoeren (Step into), de code tot het volgende breakpoint te laten uitvoeren (Continue), of uitvoering laten stoppen bij de cursor (Run to Cursor).



```
Option Compare Database
Private Sub EnterAnswer()
    Dim Found As Boolean, rs
    rst.Open "tblAnswers", C
```



In het screenshot hiernaast zijn de knoppen van links naar rechts: Continue, Break, Reset, Toggle Breakpoint, Step Into, Step Over en Step Out.



In het debug menu kun je ook de ‘Run to Cursor’ optie selecteren.

Tijdens het debuggen kun je de cursor in de editor boven variabelen houden, waarna de waarde van de variabele zichtbaar wordt. Je kunt ook een “Watch Point” toevoegen waarmee je ingewikkelder structuren kunt bekijken, door b.v. `CurrentProject.Connection` als Watch Point op te geven.

Probeer deze dingen uit met het stukje code dat je zojuist hebt gemaakt. Haal de haakjes om `rst.EOF` en `rst.BOF` weg en geef in `gezocht` een bestaande waarde voor `ve1d1` op. Volg wat er gebeurt met de debugger. Vervang ook `rst!ve1d1` door `rst.ve1d1` en gebruik weer de debugger.