

Databases

Wat zijn databases? Databases worden gebruikt om grote hoeveelheden gegevens te beheren en om te zetten in nuttige informatie. Bijvoorbeeld bij de catalogus van de bibliotheek, om de gegevens van je bankrekening te beheren en om statistische informatie verzamelen. Ze slaan de informatie op een efficiënte manier op zodat zeer grote aantallen gegevens (bijvoorbeeld de gegevens van miljoenen personen in de gemeente administratie) snel met elkaar vergeleken kunnen worden.

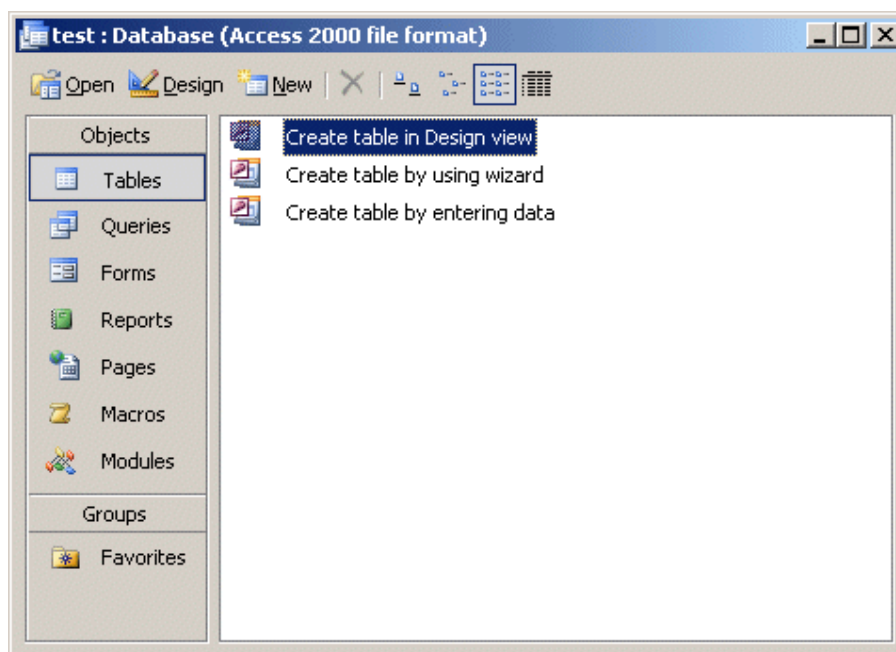
Databases bestaan meestal uit *Tabellen*. Elke tabel heeft een aantal rijen, ook wel *records* genoemd waarin één set gegevens is opgeslagen (bijvoorbeeld alle gegevens van één persoon). Per kolom (ook wel *veld* genoemd) wordt dan één soort gegeven opgeslagen (bijvoorbeeld achternaam, straat of geboortedatum). Elk tabel heeft ook een *primaire sleutel*. Dat is een veld dat gegarandeerd maar één keer voorkomt. Bij een tabel met studentgegevens zou dat bijvoorbeeld het studentnummer kunnen zijn.

■ adressen : Table							
	Voornaam	Tussenvoegsel	Achternaam	Adres	Postcode	Plaats	Geboortedatum
	Dick	van	Dijk	Toorstraat 12	6521 NR	Nijmegen	5-4-1985
	Peter		langhout	bosplein 4	7000 AA	Gouda	3-11-1979
	Sint	er	klaas	Kasteel 4	0001 ZZ	Madrid	5-12-1600
	Anne		Jansen	Stationsplein 3	6521 AA	Nijmegen	6-5-1985
	Yvonne		Vernhout	Makelaarstraat	1000 QS	Amsterdam	6-9-1975
▶							

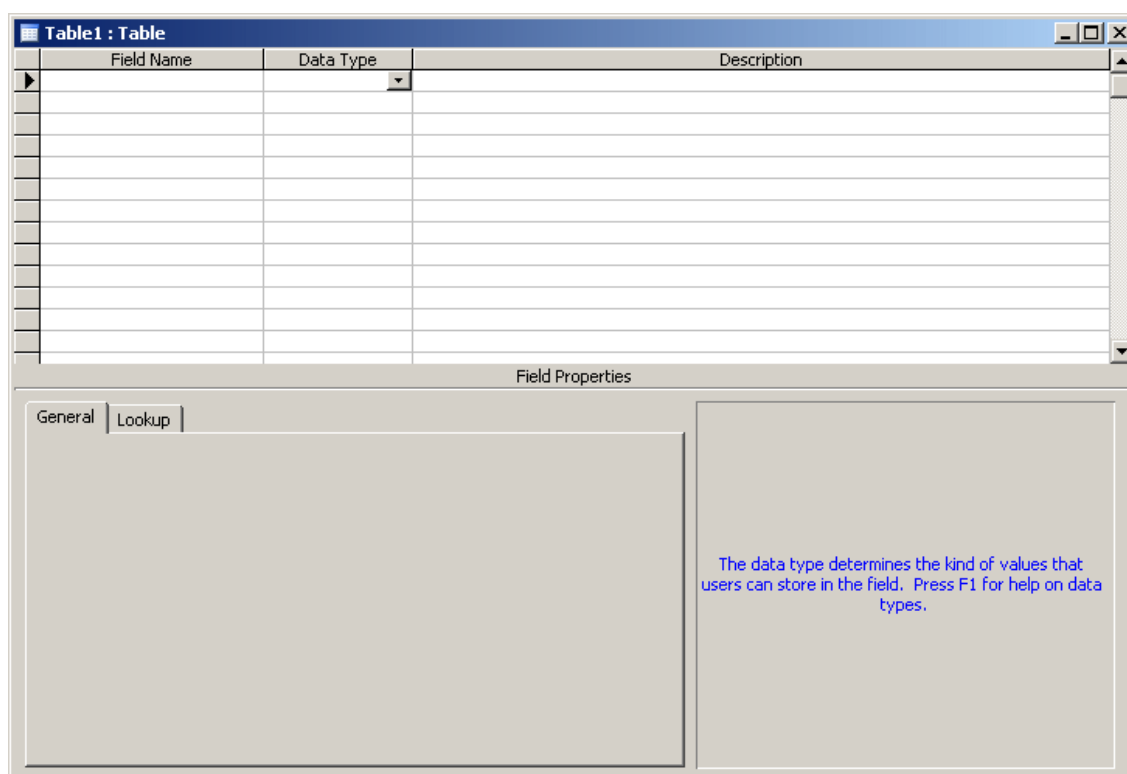
Meer complexe databases hebben meerdere tabellen. Een database van de universiteit zou bijvoorbeeld een tabel kunnen hebben met de persoonsgegevens van studenten, een tweede voor de gegevens van docenten en aparte tabellen voor elk vak (met de deelnemers, hun cijfers en de titels van hun werkstukken). Bovendien moeten al deze tabellen gecombineerd kunnen worden om bijvoorbeeld te bepalen bij welke docent de beste cijfers worden gehaald.

In deze cursus zullen we alleen werken met een meer eenvoudige database die maar uit één tabel bestaat. Voor deze cursus gebruiken we Access 2003, de database van Microsoft Office. Er zijn veel andere databaseprogramma's die elk hun eigen voordeel hebben, maar de essentie van het systeem is steeds hetzelfde.

Om te beginnen gaan we zelf een eenvoudige database maken met een adressen bestand. Allereerst moet je een nieuwe database aanmaken. Start Access en kies in de menubalk: *File* → *New* en vervolgens aan de rechterkant *Blank Database*. Je wordt nu gevraagd waar en onder welke naam je het bestand wilt opslaan. Noem het bestand 'adressen' en sla het op een plek op waar je het weer terug kunt vinden. Als dat gebeurt is heb je als het goed is het database venster voor je:



Aan de linkerkant van het venster zien we een rij met opties. De meeste gaan we niet gebruiken, alleen de tweede, *Queries*, komt later aan bod. Voorlopig gaan we eerst een tabel maken. Selecteer "Create table in Design View". Je krijgt nu een tabel te zien met drie kolommen. Hierin gaan we aangeven hoe ons tabel eruit moet gaan zien.



In de eerste kolom vul je de namen in van de velden die je wilt hebben. Voor ons adressenbestand willen we bijvoorbeeld de velden: voornaam, tussenvoegsel, achternaam, adres, postcode, plaats en geboortedatum hebben.

In de tweede kolom kun je aangeven wat voor soort gegevens er in dat veld gebruikt worden, zoals tekst, datum of nummer. Dit is van belang als we later de gegevens willen gebruiken om mee te rekenen. Als je op een vak in deze kolom klikt verschijnt er een grijze knop met een pijl naar beneden. Klik hierop en selecteer de juiste optie. De belangrijkste zijn:

Text – tekst

Number – getallen

Date/time – datums

Yes/No – logisch veld, waar/niet-waar

Memo – zeer lange teksten

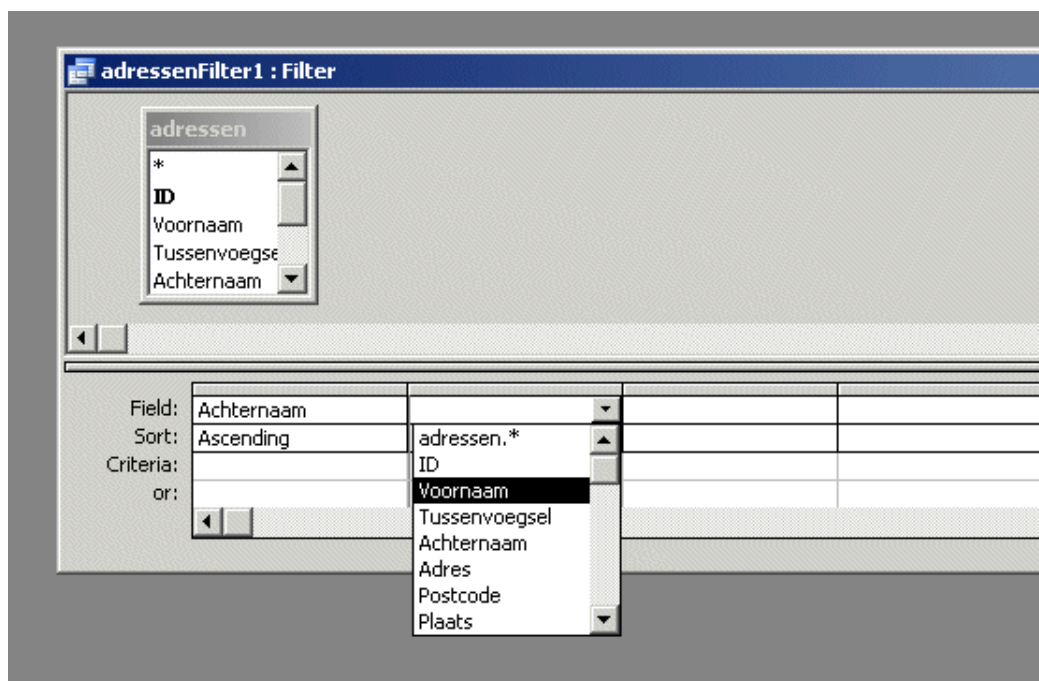
De laatste kolom geeft je de gelegenheid om uit te leggen waar het veld voor dient. In dit eenvoudige voorbeeld is dit niet echt nodig omdat de namen van de velden duidelijk genoeg zijn, maar als je meer complexe gegevens gebruikt kan het van belang zijn om uitleg te geven.

Opdracht: Vul de tabel in met de zeven genoemde velden en de bijbehorende datatypes.

Table1 : Table		
Field Name	Data Type	
Voornaam	Text	
Tussenvoegsel	Text	
Achternaam	Text	
Adres	Text	
Postcode	Text	
Plaats	Text	
Geboortedatum	Date/Time	
	Text	
	Memo	
	Number	
	Date/Time	
	Currency	
	AutoNumber	
	Yes/No	
	OLE Object	
	Hyperlink	
	Lookup Wizard...	

Daarna selecteer je in de menubalk *View* → *Datasheet view*. Het programma vraagt dan of je het tabel wilt opslaan. Zeg ja en geef het de naam *adressen*. Access waarschuwt je nu dat er nog geen primaire sleutel bestaat. De gegevens die we gekozen hebben zijn geen van allen uniek. Het is best mogelijk dat twee mensen dezelfde naam hebben, of dezelfde geboortedatum en we hebben geen student- of sofinummer. Gelukkig biedt Access zelf aan om een primaire sleutel aan te maken. Daarvoor maakt het een nieuw veld aan met een uniek nummer voor elke rij. Kies dus ja en je komt in de *datasheet view* terecht.

In beeld zie je nu een leeg tabel. Klik het witte vak onder 'Voornaam' aan en begin te typen. Er verschijnt een nieuwe regel met in de eerste kolom een 1. Op deze manier kun je één voor één alle gegevens in voeren en zorgt Access er zelf voor dat al die regels ene apart nummer hebben.



Opdracht: Stel het filter zo in dat het eerst op Plaats en dan op Achternaam sorteert.

Je kunt met filter niet alleen sorteren maar ook filteren. Stel bijvoorbeeld dat je alleen mensen wilt zien die uit Utrecht komen. Dan kun je in het sorteerfilter wat we net gebruikt hebben het volgende doen: Kies bij *Field*: "Plaats" en bij *Criteria*: "Utrecht". Als je het filter nu uitvoert is het resultaat alleen die mensen die in Utrecht wonen. Je kunt dit ook uitbreiden door op de volgende regel (bij *Or*) een extra mogelijkheid in te voeren. Bijvoorbeeld Rotterdam. Dan krijg je alle personen die in Utrecht óf in Rotterdam wonen. Door met behulp van de pijltjestoets verder te gaan in de regels eronder kun je nog meer opties geven.

Opdracht: Maak een filter dat personen met drie verschillende geboortedata selecteert.

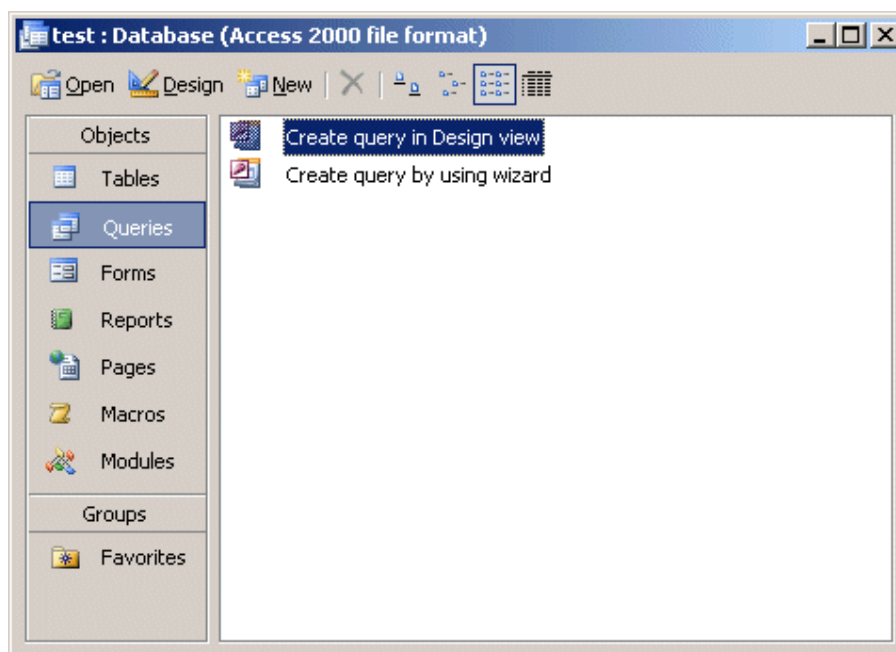
Je kunt een filter verwijderen via *Records > Remove Filter/Sort*

Queries

Als je filtert krijg je als resultaat alle velden (kolommen dus) die er in een tabel zijn. Vaak wil je die niet allemaal hebben maar slechts een klein gedeelte ervan. Bovendien wil je misschien alleen iets weten over de tabel. Bijvoorbeeld hoeveel mensen er in elke stad wonen. Voor dat soort vragen zijn er *Queries*. Dit zijn vragen die je aan de database kunt stellen waarna deze een nieuwe (tijdelijke) tabel voor je maakt met de gevraagde informatie.

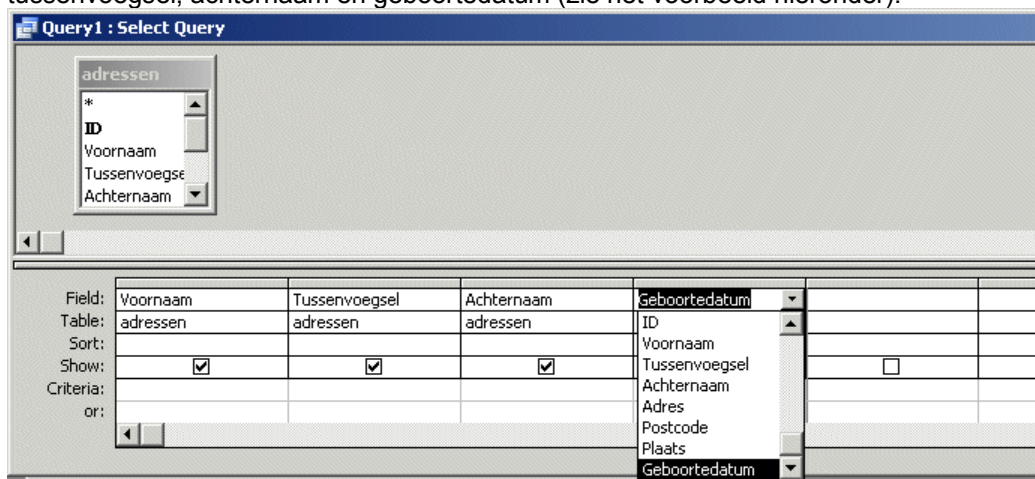
In de meest eenvoudige vorm kun je query gebruiken om de kolommen te selecteren die je wilt zien. Neem de database die je eerder hebt gemaakt. Als je alleen de naam en de geboortedatum wilt weten heb je de overige gegevens niet nodig. Je kunt dan een query maken die je alleen die gegevens geeft. Een extra voordeel van een query is dat je deze, in tegenstelling tot een filter, kunt opslaan.

Sluit je tabel af en ga in het databasevenster naar *Queries* en selecteer *Create query in Design view*.



Je krijgt een venster voor je dat lijkt op het *filter* venster, en een tweede venster *Show Table*. Allereerst moet je in dat tweede venster kiezen uit welke tabel je de gegevens gaat gebruiken. We hebben maar één tabel dus klik eerst 'Add' en vervolgens op 'close'.

We gaan eerst een eenvoudige query maken die alleen de naam en de geboortedatum oplevert. Omdat je kunt kiezen welke kolommen je wilt hebben, moet je elk veld dat je in je query wilt hebben apart aangeven. Dus selecteer in de eerste 4 vakken bij *Field* de velden Voornaam, tussenvoegsel, achternaam en geboortedatum (zie het voorbeeld hieronder).



Als je vervolgens op de menubalk *View* → *Datasheet view* selecteert krijg je een tabel zonder adres gegevens. Mocht je de query weer willen veranderen kies je op dezelfde manier *View* → *Design view*.

Als je dit venster sluit zal Access je vragen of je deze query wilt opslaan. Bevestig dit en geef hem de naam 'geboortedata'. Deze is nu zichtbaar in het query venster.

Jokers

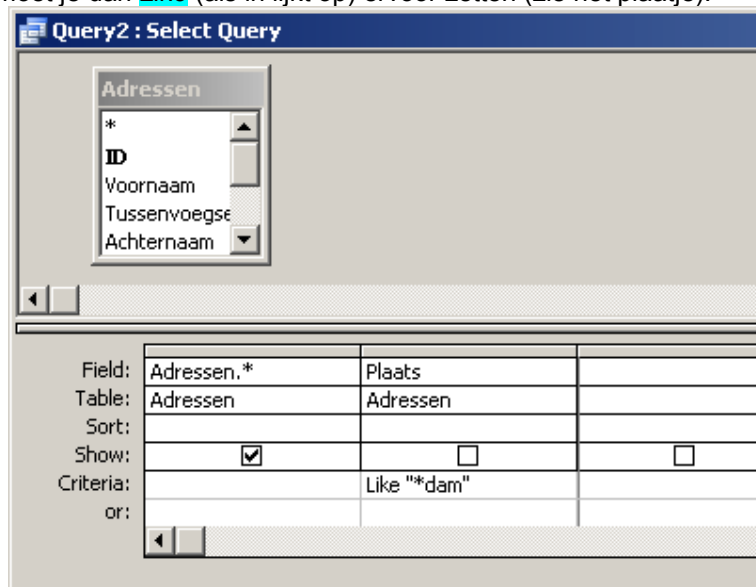
Soms wil je een selectie maken die veel verschillende mogelijkheden dekt. Stel bijvoorbeeld dat je alle namen wilt selecteren die beginnen met E. Dan kun je alle mogelijke E-namen wel

invoeren en kijken of ze aanwezig zijn, maar veel efficiënter is om een joker of wildcard te gebruiken. Deze geeft je dan alle namen die met E beginnen.

Het eerste jokerteken is een sterretje *. Dat staat voor alle mogelijke aantallen tekens (inclusief geen). Dus als we bijvoorbeeld een lijst met namen doorzoeken met "E*", krijgen we alle namen die met E beginnen, hoe lang of kort ze verder ook zijn (Dus als resultaat krijg je zowel *Ed* als *Elisabeth* als *E*).

De tweede joker is het vraagteken ?. Deze kun je gebruiken als je precies weet hoeveel letters (of cijfers) er moeten zijn maar je niet weet welke. Als je bijvoorbeeld weet dat een voornaam op -ill eindigt kun je zoeken op "?ill". Je krijgt als resultaat dan zowel *Will* als *Bill*, maar niet *Phill*.

Jokers gebruik je op de plaats van criteria, net zoals je eerder filterde op plaatsnamen. Je kunt jokers echter niet simpelweg in een criterium gebruiken. Om aan te geven dat er een joker gebruikt wordt moet je dan *Like* (als in lijkt op) ervoor zetten (zie het plaatje).



Maak een nieuwe Query ("Create Query in Design View").

- Selecteer weer de tabel Adressen en kies in de eerste kolom bij field Adressen.* Dat betekent dat je alle kolommen van de tabel Adressen wilt laten zien.
- In de tweede kolom selecteer je bij field Plaats en vul je als criterium in *Like "*dam"*. Daarmee filter je alleen die personen eruit die uit een plaats komen die eindigt op dam.
- Ten slotte haal je in de tweede kolom het vinkje weg bij de regel *Show*. Als je dit niet doet krijg je per persoon twee keer de plaats te zien. Die wordt namelijk al een keer getoond doordat je in de eerste kolom Adressen.* hebt geselecteerd.
- Ga naar de datasheet view om het resultaat te zien

In Access is het vaak nodig om de velden die je wilt laten zien (met vinkje Show), en de velden waarmee je een bewerking wilt uitvoeren (zonder vinkje Show), te scheiden.

Opdracht: Maak een query die je alle records oplevert waarvan de tweede letter in het veld Adres een o is.

Meer selectie

Eerder hebben we geselecteerd met specifieke waarden ("*Utrecht*") en met jokers. Je kunt ook selecteren op bepaalde reeksen cijfers of data. Hiervoor gebruiken we Operatoren. Dit zijn in principe drie tekens: het is-teken =, het groter dan teken > en het kleiner dan teken <. Als je bijvoorbeeld een tabel hebt met de cijfergegevens van een vak kun je als criterium bij het veld

cijfers **<5.5** aangeven. Dan krijg je als resultaat alleen degenen die een onvoldoende hebben behaald (alle cijfers *kleiner dan* 5,5). Als je alle voldoende wilt zien moet het iets anders doen. Als het criterium **>5.5** wordt, krijg je alleen degenen die *meer* dan een 5,5 hebben. Om ook de haken over de sloot studenten te krijgen moet je het criterium **>=5.5** (*is groter of gelijk* aan 5,5) geven.

Dit werkt ook met data. Maak een nieuwe query en vul als criterium bij geboortedatum in **> 1-1-1970**. Je krijgt dan de personen die na 1 januari 1970 geboren zijn. Merk op dat Access de spaties vanzelf opvult met het teken #.

Opdracht: Pas de query aan zodat hij alle personen die na 1970 en voor 1980 geboren zijn. Dit zijn eigenlijk twee criteria die op hetzelfde geboortedatumveld betrekking hebben; je kunt ze samenvoegen door het woord **And** te gebruiken, als **criterium1 And criterium2**

Samenvoegen

Je kunt informatie uit velden ook samenvoegen. Stel bijvoorbeeld dat je de hele naam wilt hebben zonder aparte velden voor voor- en achternaam. Dat kan. Maak een nieuwe query aan en voeg in het eerste Field in: **[voornaam] & [Achternaam]**
Als je deze query uitvoert krijg je het volgende resultaat:



Expr1
JanJansen
AnneJong
MariekeDijk
ErikVries
ArjanWittegem
PeterLanghout
GeertVries
NinaBrink
*

De **rechte haken** rond de titel van het veld Voornaam en Achternaam geven aan Access door dat het om **velden** gaat. Het verbindingsteken **&** zorgt ervoor dat de inhoud van beide velden verbonden wordt.

Dat is mooi, maar eigenlijk willen we een spatie tussen de namen hebben. Ga weer naar de *Design View* en vervang de tekst in het Field met: **[voornaam] & ' ' & [Achternaam]** (let op: je moet het rechte aanhalingsteken zijn, niet het schuine; het dubbele aanhalingsteken mag ook)

Het resultaat is deze keer:



Expr1
Jan Jansen
Anne Jong
Marieke Dijk
Erik Vries
Arjan Wittgem
Peter Langhout
Geert Vries
Nina Brink
*

Het gedeelte tussen aanhalingstekens wordt op de plek tussen de twee velden geplakt. Omdat er alleen een spatie tussen staat komt er dus een spatie tussen de voor- en achternaam.

Opdracht: Verander de query zodat ook de tussenvoegsels op de juiste plek staan. Negeer dat er bij namen zonder toevoegsels dan twee spaties komen te staan (dat is ingewikkelder om te voorkomen).

Rekenen

Tot nu toe hebben we alleen informatie uit de tabel gehaald die je zelf ook meteen kunt zien. Maar je kunt Queries ook gebruiken om nieuwe gegevens te berekenen. Als een winkel een database heeft met de gegevens en inkooprijzen van artikelen kan er een query gemaakt worden die in een extra kolom de hoeveelheid BTW aangeeft. Of uit een tabel met de persoonsgegevens van klanten kan de gemiddelde leeftijd berekend worden.

Ga naar www.let.uu.nl/~Gerrit.Bloothoofthoof/personal/onderwijs/naamkunde/restricted/vn10000.mdb
Sla het bestand op en open het vanuit Access (*File* → *Open*). Je hebt nu een database van de tienduizend meest populaire voornamen.

VN10000 : Table														
	VNC	SEXE	TOTAAL	CLUSTER	STAM	SUFFIX	N8589	N9094	N9599	N0004	P8589	P9094	P9599	P0004
►	Thomas	M	25056	MThomas	Thomas		4856	6559	6412	5897	8972	11424	11760	11324
	Johannes	M	23438	MJohannes	Johan	es	7618	5719	3860	2671	14075	9961	7079	5129
	Kevin	M	22171	MKevin	Kevin		5449	7165	5353	3211	10068	12479	9818	6166
	Tim	M	22063	MTim	Tim		4657	5367	5781	5155	8605	9348	10602	9899
	Jeroen	M	20954	MThomas	Jeroen		6551	6174	3839	1717	12104	10753	7041	3297
	Sanne	V	20840	VSanne	San	e	4836	4365	5432	5090	9349	7988	10478	10267
	Maria	V	20691	MJohannes	Mari	a	6866	5098	3263	2382	13274	9329	6294	4804

Open de tabel, getiteld VN10000, met veertien kolommen. Ga naar de *design view* en je ziet bij de kolom *description* voor elk veld een uitleg over wat erin staat. De belangrijkste zijn de eerste met de betreffende voornaam erin en de laatste acht. Die acht kun je opdelen in twee groepen: de kolommen met een N en de kolommen met een P. De vier N kolommen geven het totaal aantal keren dat een naam in een bepaalde periode gebruikt is. Er zijn dus bijvoorbeeld 4.856 kinderen Thomas genoemd in de periode 1985-1999. De P kolommen geven het percentage dat die naam van het totaal was, maal tienduizend. Van alle kinderen werd dus bijna 1 procent (0,8972%) Thomas genoemd tussen 1985 en 1999.

Sluit de tabel, ga naar Queries en maak een nieuwe query.

Allereerst gaan we een eenvoudige som maken. De kolom TOTAAL geeft aan het totale aantal kinderen dat van 1983 tot 2004 Thomas is genoemd. Om te kijken of dat overeenkomt met de kolommen per vijf jaren periode willen we die optellen. Maak een query die het veld VNC en Totaal selecteert. In het derde vak vullen we in: *Totaal2: [n8589]+[n9094]+[n9599]+[n0004]*

Field:	VNC	TOTAAL	Totaal2: [n8589]+[n9094]+[n9599]+[n0004]
Table:	VN10000	VN10000	
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

Het gedeelte *Totaal2:* geeft aan dat de nieuwe kolom Totaal2 moet heten. Alles vóór de dubbele punt komt in het resultaat als titel van het veld te staan. *[n8589]+[n9094]+[n9599]+[n0004]* betekent dat we de verschillende kolommen bij elkaar optellen. Je kunt alle vier standaard berekeningen, plus (+), min(-), maal(*) en gedeeld door(/), uitvoeren in een query. Als we deze query uitvoeren krijgen we dit resultaat:

Query1 : Select Query			
	VNC	TOTAAL	Totaal2
►	Thomas	25056	23724
	Johannes	23438	19868
	Kevin	22171	21178
	Tim	22063	20960
	Jeroen	20954	18281
	Sanne	20840	19723
	Maria	20691	17609
	Laura	19792	18513
	Jan	19473	17078
	Anne	19261	18387
	Dennis	18772	16273
	Mark	18205	15443
	Kim	18066	15750
	Niels	16982	15593
Record: 1 of 10000			

We zien dat de twee totalen niet helemaal hetzelfde zijn. Dat komt doordat voor het veld totaal de jaren 1983 en 1984 ook meegenomen zijn.

Opdracht: Verander de query zodat je een vierde kolom krijgt met daarin het verschil tussen de twee totalen.

Berekeningen over een kolom

In het vorige gedeelte hebben we een berekening voor elke rij laten uitvoeren. Je kunt de database ook laten rekenen over de hele lengte van een kolom. Stel bijvoorbeeld dat je wilt weten hoe vaak de top tienduizend namen gemiddeld gegeven worden. Maak een query aan en selecteer in het eerste vak bij *Field* Totaal. We willen nu iets over een kolom gaan doen en dus

hebben we een extra veld nodig. Op de werkbalk bovenaan zit een knop met een Σ teken. Als je deze inklikt krijg je in de query een extra regel genaamd *Total:*. Selecteer hieruit *Avg* (voor gemiddelde). Wanneer je naar de Datasheet view gaat zie slechts één vakje met daarin het gemiddelde (423,5 per naam).

Query1 : Select Query	
	GemOfTOTAAL
►	423,5596

Naast *Avg* zijn er een aantal andere berekeningen die je over een kolom kunt uitvoeren. *Sum* levert het totaal van alle cijfers in een kolom. *Min* en *Max* geven respectievelijk de laagste en hoogste waarde in een kolom. *Count* geeft het aantal regels in een kolom.

Sommige van deze berekeningen lijken misschien minder nuttig. De maximale waarde kun je bijvoorbeeld makkelijk vinden door het tabel te sorteren. Maar je kunt deze berekeningen ook in combinatie doen. Als je bijvoorbeeld je laatste Query verandert door in het tweede vak de kolom *Sexe* te selecteren en als criterium "*M*" te geven krijg je het gemiddelde voor mannen. Als je dit criterium vervolgens weglaat krijg je zelfs een opsomming per geslacht.

AvgOfTOTAAL	SEXE
528	?
479,15030541	M
376,48070175	V

Hiermee weten we ook meteen wat de optie *Group By* doet. Ze ordent, en maakt het dan mogelijk een berekening uit te voeren over unieke waarden. Hierboven is geordend over Sexe en vervolgens zijn de gemiddelden berekend voor de unieke waarden van Sexe: M, V en ?.

Ander voorbeeld: Als je alleen een vak maakt met Stam en deze met Group by sorteert krijg je een lijst met unieke stamnamen.

Opdracht: Maak een Query die het maximum én minimum aantal in totaal, per geslacht geeft. [groepeer op Sexe]

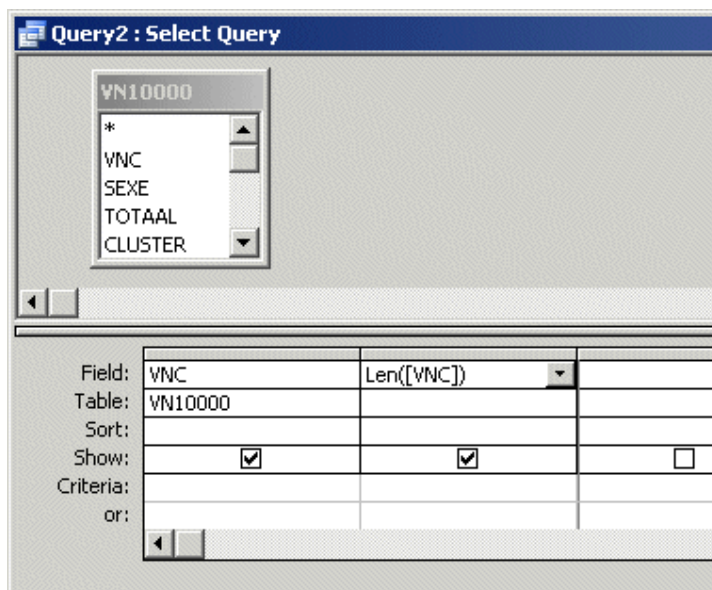
Combineren criteria

Alle mogelijkheden om Queries uit te voeren die je tot nu toe gehad hebt kunnen ook gecombineerd worden. Zo kun je bijvoorbeeld het gemiddelde vinden van alle mannelijke namen die met een P beginnen door:

- een vak dat het gemiddelde op Totaal berekent
- een vak te maken dat *Group by* uitvoert op Sexe
- een vak dat een selectie uitvoert op VNC met **Like "P*"**. In dat laatste geval moet je de *Total:* optie op *Where* instellen. *Where* veronderstelt een criterium waaraan voldaan moet worden.

(Als je daar *Group By* instelt krijg je een lijst met gemiddelden per naam, wat natuurlijk hetzelfde is als het veld Totaal).

Field:	TOTAAL	SEXE	VNC
Table:	VN10000	VN10000	VN10000
Total:	Avg	Group By	Where
Sort:			
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:		"M"	Like "P*"
or:			



En geeft als resultaat :

VNC	Expr1
Thomas	6
Johannes	8
Kevin	5
Tim	3
Jeroen	6
Sanne	5
Maria	5
Laura	5
Jan	3
Anne	4
Dennis	6
Mark	4
Kim	3
Niels	5

Iets vergelijkbaars geldt voor de andere drie functies.

Left en **Right** leveren alleen de letters die aan de linker of rechterkant van een string staan.

Hoeveel letters kun je zelf aangeven door tussen de haakjes, na een puntkomma een getal in te vullen. Dus bijvoorbeeld *Left("Plattegrond"; 4)* levert: "Plat". *Right("Plattegrond"; 5)* levert "grond". Ook hier kun je als string een kolom opgeven. Dat levert voor *Left([Vnc];5)* bijvoorbeeld op:

VNC	Expr1
Thomas	Thoma
Johannes	Johan
Kevin	Kevin
Tim	Tim
Jeroen	Jeroe
Sanne	Sanne
Maria	Maria
Laura	Laura
Jan	Jan
Anne	Anne
Dennis	Denni
Mark	Mark
Kim	Kim
Niels	Niels

Record: 1 of 10001

Mid geeft een aantal letters vanaf een bepaalde positie. `Mid("Thomas"; 3;2)` geeft *om*. Als je het laatste getal weglaat dan worden alle letters tot einde woord gegeven. `Mid("Thomas";3)` wordt dus *omas*. Merk op dat Mid iets anders doet dan Right omdat Right een vast aantal letters vanaf het einde van het woord geeft.

Opdracht: maak een query van alle voornamen die vanaf positie 3 *rr* hebben (waaronder *Gerrit*)

Instr geeft aan waar een letter of stuk tekst voorkomt in een woord. Als je bijvoorbeeld wilt weten hoeveel letters er voor "dam" zitten in Amsterdam doe je `InStr("Amsterdam"; "dam")`. Dan krijg je als resultaat: 7 want de "D" van dam is de zevende letter van Amsterdam. Dit kan ook met spaties, dus `InStr("Den Bosch"; " ")` levert op: 4.

Deze optie is ook handig om namen te filteren waar een bepaalde lettercombinatie wel of juist niet in zit. Voor de namen waar de combinatie ergens voorkomt is `Instr > 0`, voor namen waar de combinatie niet in zit is de waarde 0. Bijvoorbeeld alle namen met "maria" met de query `InStr([vnc];"maria")>0`. InStr maakt geen onderscheid tussen hoofdletters en kleine letters. Wil je alle namen waarin "maria" voorkomt maar niet vooraan, dan gebruik je `InStr([vnc];"maria")>1`.

Opdracht: maak een query met de lijst dubbele voornamen (dwz waar "-in" voorkomt).

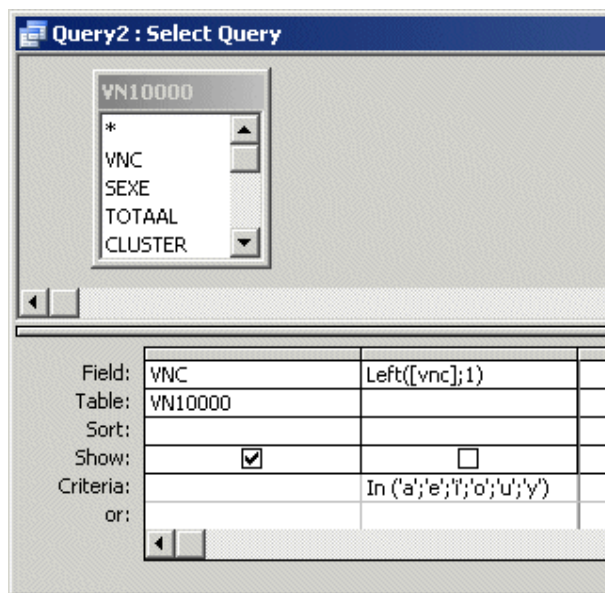
Als extra optie kun je een start positie aangeven. Stel bijvoorbeeld je wilt weten waar de eerste a in een naam voorkomt na de eerste letter. Dus je wilt bij Alexander bijvoorbeeld als resultaat niet 1 maar 5. Dan doe je: `InStr(2;"Alexander"; "a")` dan wordt de eerste letter overgeslagen (hij begint bij 2) en vindt de functie als eerste a dus de vijfde letter.

Opdracht: maak een query die de lijst met voornamen geeft en daarachter hoeveel letters er na de eerste t komen. Dus Chantal geeft als resultaat 2 en Sebastiaan geeft 4.

Let op: In de nieuwste versie van Access moet je i.p.v. een puntkomma (;) een komma (,) gebruiken om velden binnen functie te scheiden. Dus `InStr("Amsterdam", "dam")`. Als jou versie van Access de functies niet accepteert, probeer het dan met een komma.

Een laatste nuttige functie is **IN**. Hiermee kun je bepalen of een letter, stuk tekst of getal voorkomt in een aantal alternatieven die je hebt gekozen. Je kunt daarbij meerdere

mogelijkheden op geven. Stel bijvoorbeeld dat we de namen willen hebben die met een klinker beginnen. Dan willen we dus iets weten over de eerste letter van het voornamen veld (Left([vnc];1)) en stellen we als criterium: IN ('a';'e';'i';'o';'u';'y'). Als de letter dan overeenkomt met één van de opgegeven klinkers (aeiouy) wordt de naam getoond anders niet.



Je krijgt dus een lijst met de namen als Eva, Alexander en Ingrid, maar zonder Thomas, Lisa of Stephan. Bovendien kun je kijken hoeveel resultaten er zijn:

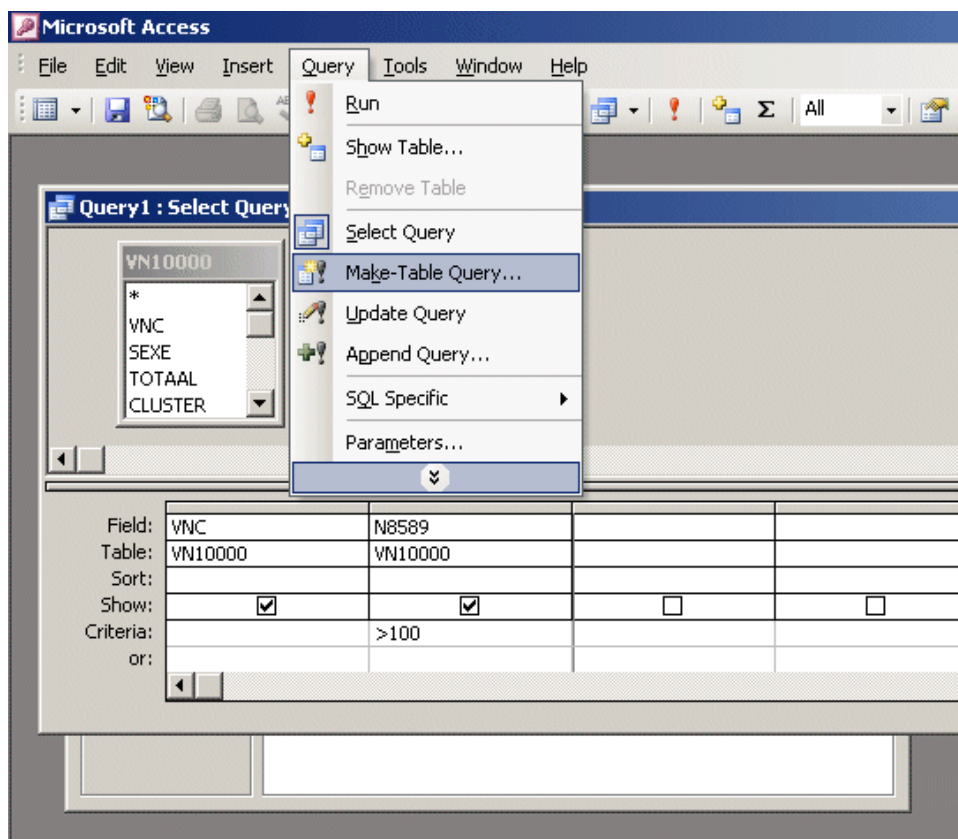


Zo weet je dus dat 1942 van de 10.000 namen met een klinker beginnen.

Opdracht: Maak een query die alle namen geeft die eindigen op 'ina' of 'ine'

Opslaan Resultaten

Als je het resultaat van een query of de informatie van een tabel ergens anders wilt opslaan, kan dat met de *Export* functie. Selecteer simpelweg een query of tabel en ga naar *File* → *Export*. Je wordt dan gevraagd waar je het wilt opslaan. Je kunt dan twee dingen doen. Je kunt het opslaan in een andere database (bijvoorbeeld een nieuw, leeg bestand) of je kunt het opslaan als Excel of tekstbestand. In het eerste geval kun je gewoon de betreffende database selecteren en op *Export* klikken. In het tweede geval moet je bij *Save as Type* voor Excel of één van de tekst mogelijkheden kiezen, het een naam geven en dan pas op *Export* klikken.

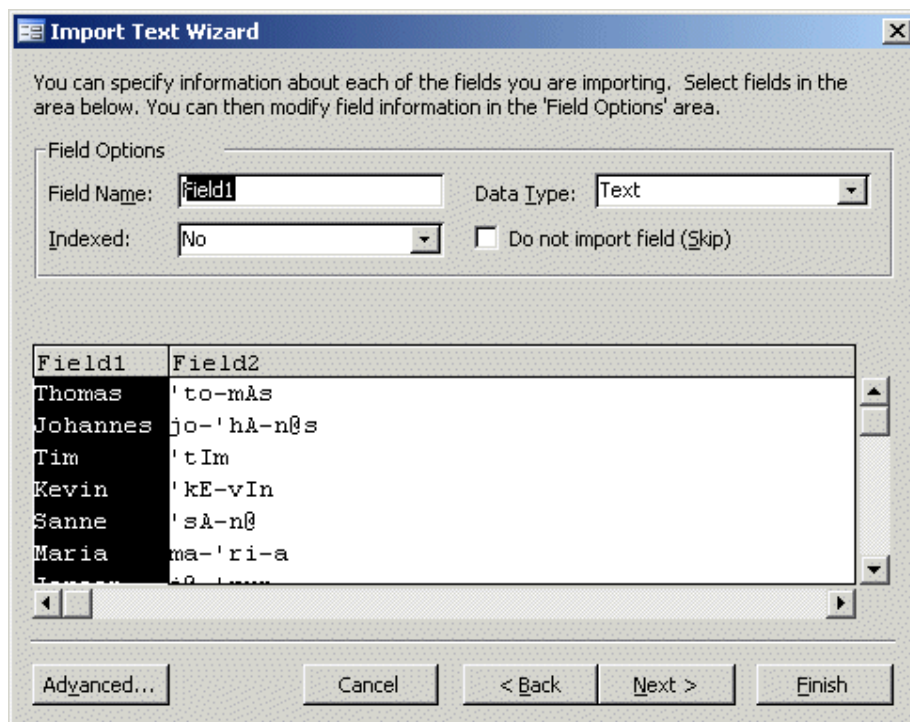


Vervolgens selecteer je in de menubalk *Query* → *Make-Table Query* (het was default *Select Query*). Hierdoor selecteert de query niet alleen, maar slaat hij het resultaat ook op als nieuwe tabel. Wanneer je op *Make-Table Query* klikt krijg je de vraag hoe de nieuw te genereren tabel moet heten en waar je hem op wilt slaan. Doe dat in deze database onder de naam "QResultaat". Deze tabel bestaat nu wel, maar is nog niet gevuld.

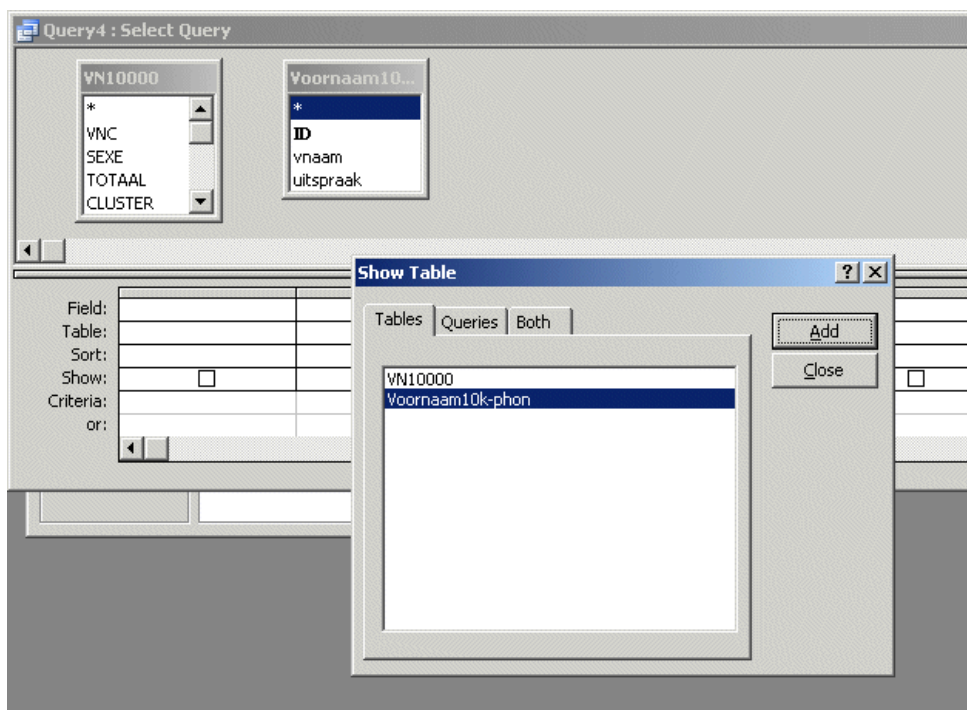
Sla de query nu eerst zelf op (noem hem "vaakvoor"), voer hem vervolgens uit (met *Run*, bovenin het query menu). Je wordt dan gevraagd of je zeker weet dat je een nieuwe tabel wilt maken en of je zeker weet dat je al die records wilt plakken in de nieuwe tabel. Klik bij beide op ja en je hebt een nieuw tabel.

Vergeet nu niet om query weer op het *Select Query* type te zetten. Klik de naam van de Query aan en klik op *Design*. Je kunt nu in de menubalk de query weer als gewone select query instellen (*Query* → *Select Query*).

Als je deze tabel Qresultaat in een nieuwe database wilt zetten zul je die database eerst moeten aanmaken. Dat kan kan via de menubalk: *File* → *New*. Kies weer *Blank database*, een naam en een plaats waar dat bestand komt te staan. Vervolgens moet je daarin het query resultaat (Qresultaat) uit VN10000.mdb importeren. Dat gaat via *File* → *Get External Data* → *Import*. Kies dan VN10000.mdb en de gewenste tabel.



Stel beide kolommen in als tekst en geef de eerste de naam *vnaam* en de tweede *uitspraak*. Klik weer op next. Nu wordt het lastiger. We moeten een primaire sleutel kiezen. Als het goed is weet je nog dat dit een veld moet zijn waarvan geen record hetzelfde is. Omdat we niet zeker weten dat de voornamen uniek zijn kiezen we *Let Access add primary key*, deze krijgt de naam *ID*. Klik op *next*, noem de tabel *uitspraak* en klik op *finish*.



Omdat je nu meerdere tabellen hebt, kunnen die ook allemaal in een query worden gebruikt. Maak een nieuwe query, en selecteer in *Show Table* de tabellen die je wilt gebruiken, dwz VN10000 en Voornaam10k-phon, één voor één en klik *add* zodat ze allebei als vak in het venster erachter verschijnen.

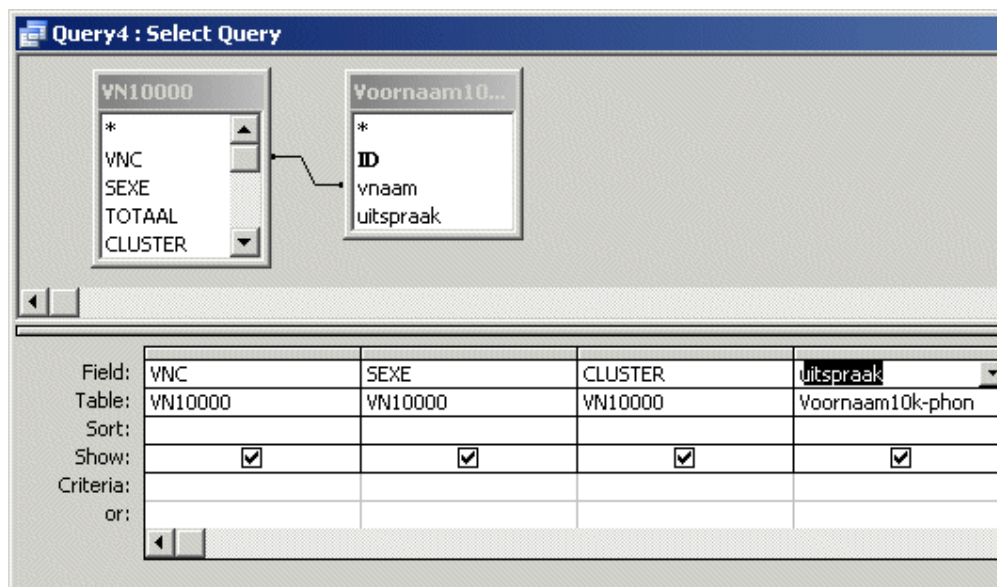
Je kunt tabellen soms ook twee keer nodig hebben, selecteer ze hier dan twee keer. Mocht je later nog een tabel aan een query willen toevoegen, dan kun je in het hoofdmenu onder Query de optie +ShowTable kiezen. Dan kom je weer in het Show Table scherm.

Mocht je de resultaten van een query willen combineren met elkaar of met een table kan dat overigens ook. Je kunt dan in *Show Table* het tabblad *Queries* selecteren en een query toevoegen.

Sluit nu het *Show Table* venster.

We hebben nu twee tabellen, VN10000 en Voornaam10k-phon. We willen een query maken met de voornaam, het geslacht, het bijbehorende cluster en de correcte uitspraak. Het cluster is de naamgroep die namen bevat waaruit bepaalde ouders vaak voor kiezen. Selecteer in de VN10000 tabel de velden VN10000.VNC, VN10000.SEXE en VN10000.CLUSTER. In de dropdown lijst zie je dat de tabelnaam met een punt aan de veldnaam is verbonden. Zo kan er bij gebruik van meerdere tabellen geen misverstand over bestaan bij welke tabel een veld hoort. Selecteer in Voornaam10k-phon het veld uitspraak.

Als we deze query zouden uitvoeren, weet Access nog niet welke uitspraak bij welke voornaam uit VN10000 hoort. Daarvoor moeten we de voornamen van de ene tabel verbinden met die van de ander. Daarvoor klikken we met de linker muisknop op *vnaam* in Voornaam10k-phon, houden we deze ingedrukt en slepen we het naar VNC in VN10000. Er verschijnt een verbindingstreepje tussen *vnaam* en VNC. We zeggen dat **de relatie** tussen VN10000 en Voornaam10k-phon is gelegd (via een gemeenschappelijk veld). Je kunt een relatie weer ongedaan maken door er met de rechter muisknop op te klikken en dan verwijderen te kiezen.



Als we nu de query opslaan als "cluster uitspraak" en hem vervolgens uitvoeren krijgen we dit resultaat:

cluster uitspraak : Select Query				
	VNC	SEXE	CLUSTER	uitspraak
▶	0	V		'nYl
	0	M		'nYl
	0	V		'nYl
	0	M		'nYl
	A	M		'a
	Aad	M		'at
	Aafje	V	MJohannes	'a-fj@
	Aafke	V	MJelle	'af-k@
	Aagje	V	MJan	'ax-j@
	Aagtje	V	MJan	'ax-tj@
	Aalbert	M	MJan	'al-bErt
	Aaldert	M	MJan	'al-d@rt
	Aaldrik	M		'al-drIk
	Aaliwah	V		a-'lF+a
Record: 1 of 10323				

Opdracht: Download het meerlingen bestand op

<http://www.let.uu.nl/~Gerrit.Bloothoofthof/personal/onderwijs/Naamkunde/restricted/MEERTEL.zip>

Laad dit als tabel "meerling" en maak een nieuwe query met vier velden: de beide namen en de daar bijhorende uitspraak. Noem deze query "meerlinguitspraak".

(Hint: Je kunt een tabel meerdere keren selecteren in *Show Table*. Merk op dat Access dan de tweede tabel **1** aan de naam toevoegt om hem van de eerste te onderscheiden. In query criteria moet je nu altijd de tabelnaam aan een veld toevoegen, bv vn10000.vnc en vn10000_1.vnc).

Natuurlijk kun je al deze mogelijkheden ook combineren. Wanneer je meerdere bewerkingen moet uitvoeren (bijvoorbeeld tabellen samenvoegen en een lastige selectie maken) kan het overzichtelijker zijn om dit in stappen te doen en de tussenoplossingen van één query in een andere te gebruiken.

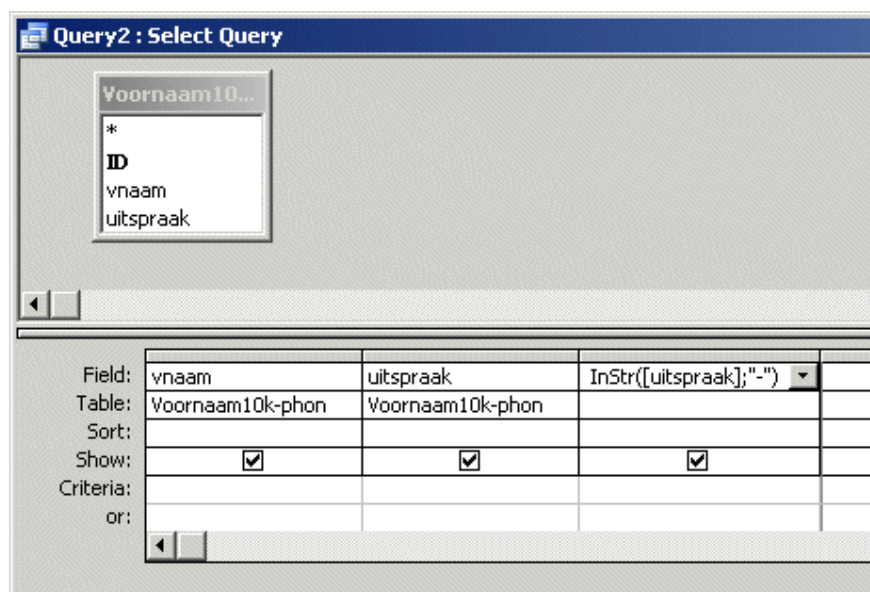
Opdracht: maak een nieuwe query die een lijst geeft van tweelingen waarvan beide namen in hetzelfde cluster vallen. Geef als velden de beide namen, de uitspraak van de namen en het cluster.

(Hint: Voeg de uitspraak pas later toe. Het cluster veld staat in VN10000. Je zult die tabel twee keer nodig hebben want je wilt **VN10000.cluster=VN10000_1.cluster** uitvoeren).

Opdracht: In *uitspraak* zijn de lettergrepen gescheiden door een verbindingsteken **~**. De klemtoon wordt aangegeven door een aanhalingsteken **'**. Maak een query die een lijst geeft van namen waarbij de klemtoon op de tweede lettergreep valt.

(Hint: Je moet eerst namen selecteren die een tweede lettergreep hebben. Gebruik vervolgens de Mid functie om het teken na **'** te selecteren;).

Bedenk bij het maken van lastige selecties, dat het handig kan zijn om het in stappen te doen die je ook als resultaat geeft. Als je bijvoorbeeld een naam wilt inkorten tot de eerste lettergreep kun je eerst een veld maken dat de lengte geeft tot het eerste verbindingsteken **~**.

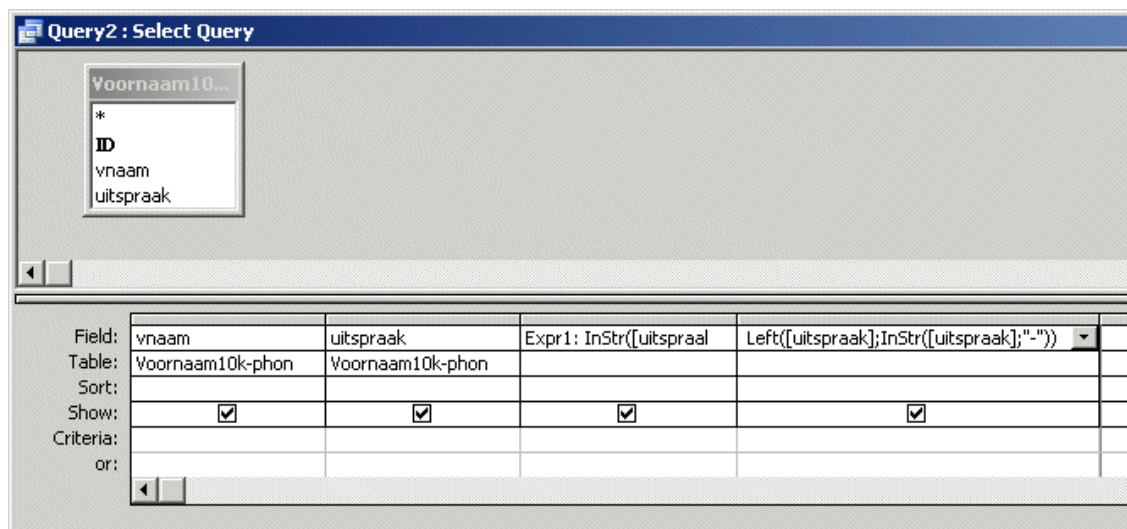


The screenshot shows the results of the query in a table with four columns: 'vnaam', 'uitspraak', 'Expr1', and a numeric column. The data is as follows:

	vnaam	uitspraak	Expr1
▶	Thomas	'to-mAs	4
	Johannes	jo-'hA-n@s	3
	Tim	'tIm	0
	Kevin	'kE-vIn	4
	Sanne	'sA-n@	4
	Maria	ma-'ri-a	3
	Jeroen	j@-'run	3
	Laura	'lA+ra	5
	Jan	'jAn	0
	Anne	'A-n@	3
	Dennis	'dE-nIs	4
	Kim	'kIm	0
	Mark	'mArk	0
	Niels	'nIls	0

Record: 1 of 9672

Als deze goed werkt, kun je dit getal weer gebruiken om het veld in te korten.



Query2 : Select Query				
	vnaam	uitspraak	Expr1	Expr2
▶	Thomas	'to-mAs	4	'to-
	Johannes	jo-'hA-n@s	3	jo-
	Tim	'tIm	0	
	Kevin	'kE-vIn	4	'kE-
	Sanne	'sA-n@	4	'sA-
	Maria	ma-'ri-a	3	ma-
	Jeroen	j@-'run	3	j@-
	Laura	'lA+-ra	5	'lA+
	Jan	'jAn	0	
	Anne	'A-n@	3	'A-
	Dennis	'dE-nIs	4	'dE-
	Kim	'kIm	0	
	Mark	'mArk	0	
	Niels	'nIs	0	

Record: 1 of 9672

Denk er dus aan dat je het resultaat van een functie in een andere functie kunt gebruiken. In het bovenstaande voorbeeld gebruiken we het resultaat voor `InStr([uitspraak]; "-")` dat dus de positie geeft waar de volgende lettergreep begint om alleen de eerste lettergreep met `Left` te pakken `Left([uitspraak]; het resultaat van InStr)`. Dat kun je zoveel doen als je wilt, je moet alleen uitkijken met haken.

Bonus opdracht: maak een query die een lijst geeft van namen waarvan de klemtoon op de laatste lettergreep valt.