

DATABASES

We hebben geleerd dat de gegevens 'de grondstof' vormen bij de informatieproductie.

Om de benodigde gegevens voor een bepaalde informatieproductie op een gemakkelijke manier te kunnen terugvinden, zal men de gegevens niet zo maar lukraak opslaan. Men zal ze groeperen, al naar gelang het doel waarvoor ze worden bewaard.

Een bestand, gegevensverzameling of *file* is dan ook een verzameling van bij elkaar behorende gegevens (*data*) die voor een of meer doeleinden bijeen zijn gebracht. Deze worden in een bestand opgeslagen met de bedoeling er op een later tijdstip gebruik van te maken. Met het oog hierop dienen de gegevens zodanig toegankelijk te zijn, dat ze op een eenvoudige manier kunnen worden geraadpleegd of verder verwerkt. De gegevens van een bestand kunnen op allerlei manieren worden bijgehouden, bijvoorbeeld op een lijst, in een ordner of in een kaartenbak.

De naam van het bestand duidt meestal aan, welk soort gegevens in dit bestand wordt bijgehouden, bijvoorbeeld ledenbestand, artikelbestand, klantenbestand enz.

In de huidige maatschappij is door de explosieve ontwikkeling op allerlei terreinen een bijna onuitputtelijke behoefte aan informatie ontstaan. Deze informatiebehoefte heeft op zichzelf weer geleid tot een nog groter aanbod van gegevens. Hierdoor en door de snelheid waarmee de informatie moet worden geproduceerd, is het onvermijdelijk dat de informatieverzorging steeds meer geautomatiseerd wordt, waarbij de gegevens door een computersysteem worden beheerd.

Onder een *computerbestand* wordt verstaan: een met behulp van een computersysteem beheerde gegevensverzameling. De gegevensverzamelingen dienen uiteraard zo vastgelegd te zijn, dat zij voor de computer met behulp van randapparatuur leesbaar en toegankelijk zijn.

Voor we een bepaald bestand samenstellen, zullen we ons moeten afvragen waarvoor we het willen gebruiken. Afhankelijk hiervan stellen we vast welke gegevens er in dit bestand opgenomen worden. Indien een artikelbestand alleen nodig is voor de facturering aan de klanten, moet van elk artikel het nummer, de naam en de verkoopprijs worden opgenomen. Willen we daarnaast ook regelmatig een voorraadoverzicht hebben, dan zullen tevens van elk artikel de voorraad in stuks en de kostprijs in dit bestand opgenomen worden.

Vervolgens bepalen we welke gegevens in dit bestand op hetzelfde onderwerp betrekking hebben. In het voorbeeld van het artikelbestand behoren bij elk artikel het nummer, de naam en de verkoopprijs. Een groep gegevens die bij elkaar hoort en betrekking heeft op hetzelfde onderwerp, noemt men een *record*. In ons voorbeeld bevat het artikelrecord de relevante gegevens van één artikel. Zo zullen alle relevante gegevens van één klant in een klantenrecord staan.

Records met dezelfde soort gegevens voegt men tot een bestand samen. Zo vormen alle artikelrecords samen het artikelbestand en alle klantenrecords het klantenbestand.

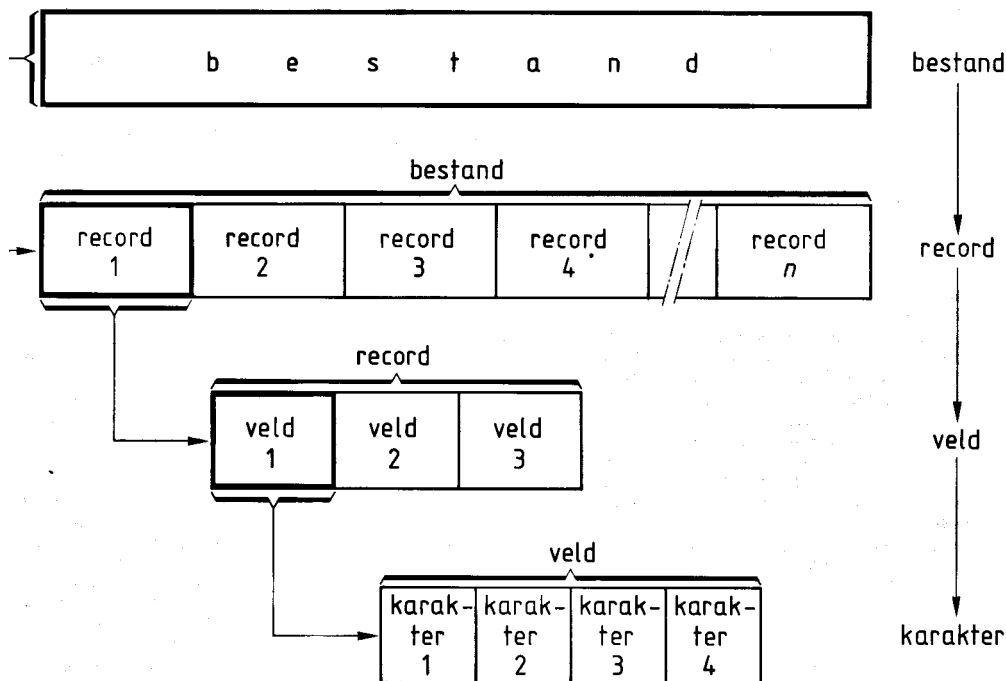
Elk record bestaat op zijn beurt uit een of meer velden of items. Elk artikelrecord in het voorbeeld bevat een nummerveld, een naamveld en een verkoopprijsveld. Elk veld is weer opgebouwd uit een of meer karakters, dat wil zeggen uit letters, cijfers en leestekens. Indien het artikelnummer uit vier karakters bestaat, vormen deze vier karakters het nummerveld.

Men onderscheidt de volgende typen velden:

1. numerieke velden;
2. alfanumerieke velden;
3. logische ofwel boolean velden
4. datumveld : Wanneer een datum (dag, maand, jaar) in een dergelijk veldtype is vastgelegd, kan het verschil in dagen met een andere datum snel worden bepaald.
5. currency

Samengevat:

Een bestand is opgebouwd uit een aantal gelijksoortige records; elk record bestaat op zijn beurt uit een of meer velden en elk veld weer uit een of meer karakters; zie figuur hieronder.



Willen we een bepaald record in een bestand op een eenvoudige manier terug kunnen vinden, dan zal het zich op een bepaalde manier van de overige records moeten onderscheiden. Dit houdt in, dat we elk record van een uniek gegeven voorzien, waarmee het ondubbelzinnig kan worden geïdentificeerd. Een gegeven waarmee een record binnen een bestand uniek wordt gemaakt, noemen we recordsleutel, recordkey of kortweg key. Een sleutel bestaat meestal uit één veld, soms uit een combinatie van enkele velden uit een record.

In een klantenbestand zou je als recordsleutel de achternaam van de klant kunnen kiezen. Het zal evenwel duidelijk zijn dat dit problemen kan opleveren, omdat er in het klantenbestand meer personen kunnen voorkomen met dezelfde achternaam.

Dit is de reden waarom men in de praktijk meestal zelf een sleutelgegeven aan elk record toevoegt. Hiervoor gebruikt men vaak een nummer. Elk record heeft dan zijn eigen nummer, zodat vergissingen uitgesloten zijn. Zo zal in een artikelbestand het artikelnummerveld als key fungeren, in het klantenbestand het klantnummerveld en in een telefoonabonneebestand het netnummerveld plus het abonneenummerveld.

De keuze van een goede sleutel is een belangrijke zaak, omdat alleen met de juiste sleutelwaarde het gewenste record in een bestand teruggevonden kan worden. We kunnen in dit geval dus werkelijk van 'sleutel' spreken. Zonder kennis van de juiste sleutel blijft de toegang tot een bepaald gegeven in het bestand afgesloten.

9011 018397	1	1	RAATS	WERKBOEK LOTUS 1,2,3 HL	SOFTW 5,25	25.00
9011 885856	3	1	RAATS	WERKBOEK LOTUS 1,2,3 HL	SOFTW 3,50	25.00
9011 885848	3	1	RAATS	MECHANICA, TOEPASSINGEN IN DE BOUW	UW	23.75
9011 011783	1	2	RATELBAND	ZIE VINK		
9011 446569	2	2	RAUWERDA	MACHINALE HOUTBEWERKING		91.25

Een gedeelte uit het titelbestand van Educaboek

De bestanden zijn als volgt te onderscheiden:

- 1 *Permanente bestanden* (= stambestand, historisch bestand, bewaarbestand, masterfile of systeembestand) zijn bestanden die men regelmatig voor de gegevensverwerking nodig heeft. Om zo'n bestand actueel (= up to date) te houden, zal men het regelmatig bijwerken (= muteren). Voorbeelden hiervan zijn het artikelbestand en het klantenbestand.
- 2 *Tijdelijke bestanden* zijn bestanden waarvan het niet de bedoeling is dat ze penmanent in het systeem aanwezig blijven. Voorbeelden daarvan zijn:
 - a. *invoerbestanden*, hierin zijn gegevens vastgelegd die later nog in een gegevensverwerkend systeem moeten worden verwerkt, bijvoorbeeld een bestand met betalingsopdrachten zoals dat door grote bedrijven wordt aangemaakt en aan de bankgirocentrale wordt aangeboden. Wanneer in een

invoerbestand wijzigingen (mutaties) worden verzameld ter verwerking in een permanent bestand, noemt men het ook wel mutatiebestand.

- b. *uitvoerbestanden*, deze worden aangemaakt als de geproduceerde informatie niet gelijk op printer of beeldscherm wordt gepresenteerd. Redenen daarvoor kunnen zijn, dat een uitvoerapparaat tijdelijk niet beschikbaar is of dat het afdrukken van de informatie op een ander systeem moet plaatsvinden.

Verder kent men nog de tijdelijke hulpbestanden die door systeem- en toepassingsprogramma's zelf worden aangemaakt. Meestal hebben deze een beveiligingsfunctie of worden ze tijdelijk aangemaakt om een snellere werking van het systeem te verkrijgen.

De volgende bewerkingen met een bestand zijn mogelijk:

- 1 aanmaken of creëren (van nieuw bestand);
- 2 onderhouden of muteren
- 3 raadplegen (retrieval);
- 4 kopiëren;
- 5 sorteren;
- 6 indexeren.

Het aanmaken of creëren van een nieuw bestand

Voor er sprake kan zijn van een bestand, zal er een bestand gecreëerd moeten worden. Dit zal in veel gevallen plaatsvinden door invoer van gegevens via het toetsenbord aan de hand van basis- of invoerdocumenten. De overige mogelijkheden van gegevensinvoer zijn beschreven in hoofdstuk 4.

Het onderhouden of muteren van een bestand

Om juiste informatie te kunnen verstrekken, is het noodzakelijk dat de bestanden actuele en betrouwbare gegevens bevatten. De gegevens zullen daarom regelmatig bijgewerkt moeten worden.

We kunnen voor het actueel houden van bestanden drie soorten mutaties onderscheiden:

- 1 het toevoegen van nieuwe records (inserting), bijvoorbeeld het opnemen van een nieuw artikel in het artikelbestand;
- 2 het veranderen van bestaande records (updating), bijvoorbeeld het wijzigen van de prijs van een artikel in het artikelbestand;
- 3 het verwijderen van bestaande records (deleting), bijvoorbeeld het uit het assortiment nemen van een artikel.

Bij het wijzigen van bestanden maakt men gebruik van het sleutelgegeven. Daarnaast zal men, om er zeker van te zijn dat het juiste record gewijzigd wordt, in een aantal gevallen gebruik maken van een extra gegeven uit het record. Zo zal voor het verwijderen van een artikelrecord naast het sleutelgegeven de artikelnaam als controlegegeven kunnen worden gebruikt.

Het raadplegen van bestanden

Het hoofddoel van een bestand is het verstrekken van gegevens voor de informatievoorziening. Dit raadplegen kan inhouden:

- 1 het zoeken naar een bepaald record (searching), bijvoorbeeld het tonen van de artikelgegevens van een bepaald artikelnummer op het beeldscherm;
- 2 het selecteren van records aan de hand van een of meer selectiekenmerken, bijvoorbeeld het afdrukken op een lijst van de artikelen waarvan de voorraad lager is dan 50 stuks.

Het kopiëren van bestanden (back-up file)

Het kopiëren van bestanden gebeurt uit veiligheidsoverwegingen, om bij calamiteiten het verloren gegane bestand te kunnen reconstrueren.

Het sorteren van bestanden

Onder sorteren wordt verstaan het rangschikken van de records in een bepaalde volgorde op grond van een sorteerkennmerk, bijvoorbeeld het alfabetisch rangschikken van de klantenrecords of het opklimmend rangschikken van de artikelrecords op artikelnummer.

Het indexeren van bestanden

Indexeren is het aanmaken van een extra bestand om met behulp daarvan snel toegang te krijgen tot de ongesorteerde gegevens. Naar de index worden alleen de velden gekopieerd waarin snel gezocht moet kunnen worden, aangevuld met verwijzingen naar de oorspronkelijke records. Vervolgens wordt de index gesorteerd.

Methoden voor de bewerkingen met bestanden kunnen zijn:

- 1 real-time verwerking;
- 2 batch-verwerking.

Real-time verwerking wordt ook wel postgewijze of transactiegewijze verwerking genoemd. Bij deze methode wordt de gewenste computerverwerking onmiddellijk, althans binnen enkele seconden, uitgevoerd. Deze verwerking kan betrekking hebben op het direct aanbrengen van een wijziging in óf op het direct verkrijgen van informatie uit een bestand.

Bij het reserveren van vliegtuigplaatsen bijvoorbeeld is het van belang, dat men direct geïnformeerd wordt of er voor een bepaalde vlucht nog plaats is. Zo ja, dan dient voor de desbetreffende reiziger deze plaats ook direct in het bestand te worden gereserveerd.

Real-time verwerking vindt uiteraard altijd on-line plaats. Het randapparaat, bijvoorbeeld de terminal, staat via een datalijn rechtstreeks in verbinding met de centrale eenheid.

Real-time verwerking wordt toegepast als:

- de gegevens in de bestanden op elk moment up to date dienen te zijn;
- men op elk gewenst moment moet kunnen beschikken over de gegevens uit de bestanden.

Een ander voorbeeld is het gebruik van balie-terminals bij banken, waarmee men kasstortingen of opnamen direct kan verwerken; in het laatste geval kan men ook controleren of er nog voldoende geld op de rekening staat.

Batch-verwerking heet ook wel stapelgewijze of groepsgewijze verwerking. In een aantal gevallen is het niet nodig dat de gewenste computerverwerking direct plaatsvindt. De gegevens in het bestand hoeven niet op elk moment up to date te zijn. De mutaties worden in dit geval over een bepaalde periode opgespaard tot een stapel of groep en na afloop van die periode verwerkt. Deze periode kan variëren van een uur tot een jaar. Batch-verwerking wordt vooral toegepast daar waar de gegevens van een bestand periodiek worden geraadpleegd. Indien bijvoorbeeld eenmaal per maand de salarissen worden berekend, kan men de mutaties opsparen en het bestand juist vóór de uitbetaling bijwerken.

Gegevensbank (databank of database)

In dit hoofdstuk hebben we tot nu toe alleen gesproken over aparte bestanden, waarbij elk bestand ontworpen is voor een specifiek onderdeel in het totale informatiesysteem. Zo zal het onderdeel 'magazijnbeheer' gebruik maken van het artikelvoorraadbestand, het onderdeel 'inkoopadministratie' van het inkoopartikelbestand en het onderdeel 'facturering' van het orderbestand, het klantenbestand en het artikelbestand.

Het gevolg hiervan is, dat veel werk dat eenmalig kan gebeuren, in een aantal gevallen onnodig wordt herhaald, omdat in de afzonderlijke bestanden dezelfde gegevens worden bijgehouden. Zo zal het artikelvoorraadbestand en het inkoopartikelbestand gedeeltelijk dezelfde gegevens bevatten. Er is in zo'n geval sprake van redundantie (overtolligheid van gegevens). Dit betekent méér dan alleen verspilling van externe geheugenruimte. Wanneer namelijk deze bestanden regelmatig gemuteerd worden, is het denkbaar dat de bestanden onderling niet meer met elkaar in overeenstemming zijn. Iedere beheerder van een onderdeel van het informatiesysteem is immers zelf verantwoordelijk voor de verwerking van de gegevens in deze bestanden.

Om deze nadelen zoveel mogelijk op te heffen, kan men de afzonderlijke bestanden integreren tot één gegevensbank. De gegevens in zo'n gegevensbank moeten naar verschillende gezichtspunten toegankelijk zijn, zodat elk programma op basis van zijn eigen zoekkenmerk ervan gebruik kan maken.

De gegevens worden in principe eenmalig opgeslagen. Hierdoor wordt de redundantie tot een minimum beperkt, met alle voordelen van dien. Wanneer nu een bepaalde wijziging plaatsvindt, worden alle hierop betrekking hebbende gegevens in de databank dienovereenkomstig aangepast, zodat alle gebruikers dezelfde consistente gegevens tot hun beschikking hebben.

Een gegevensbank (*database*) is een verzameling van alle logisch bij elkaar behorende gegevens voor een bepaald informatiesysteem, zodanig beheerd dat ze vanuit verschillende gezichtspunten en door verschillende toepassingsprogramma's (tegelijk) toegankelijk zijn.

Bij het beheer van een gegevensbank wordt gebruik gemaakt van speciaal daarvoor ontwikkelde programmatuur, een *DataBase Management System* of DBMS. In een dergelijk systeem worden in een *data dictionary* (DD) een groot aantal kenmerken van de te beheren gegevens beschreven, o.a. veldnaam, lengte, gegevenstype en relatie met andere gegevens. Men zegt wel, dat in de data dictionary de gegevens over de gegevens zijn vastgelegd.

De wijze waarop gegevens in verschillende bestanden of tabellen zijn samengebracht en de verbanden tussen de bestanden, vormen de structuur van de database. Deze structuur wordt binnen een DBMS (netwerkmodel) beschreven in DDL (*data description language*).

Voor het bewerken (invoeren, wijzigen, opvragen etc.) van gegevens in een database gebruikt men DML (*data manipulation language*). Bij het schrijven van toepassingsprogramma's hebben programmeurs voor de toegang tot de database via het DBMS een aparte set opdrachten tot hun beschikking.

Databasemodellen

Voor het aanbrengen van structuur in een database zijn verschillende modellen in gebruik. Men kent het hiërarchische model, het netwerkmodel en het relationele model. Wetenschappers onderzoeken of het semantische model voor de praktijk geschikt kan worden gemaakt.

Omdat het relationele model in de praktijk het meest wordt toegepast, gaan we alleen op dat model iets verder in.

Relationele databases

Binnen een relationele database zijn de gegevens vastgelegd in tabellen (kolommen en rijen). De wijze waarop de gegevens in tabellen worden samengevoegd, vindt men door de vast te leggen gegevens te normaliseren (zie volgend hoofdstuk). Tussen de tabellen worden relaties aangebracht met behulp van extra kolommen met verwijzingen, of met behulp van extra tabellen met verwijzingen. De samenhang tussen de gegevens wordt uitsluitend in tabellen vastgelegd.

Voor het omgaan met gegevens in een relationele database wordt veel gebruik gemaakt van SQL. Een van de mogelijkheden van deze taal is het creëren van een view, dat is een niet echt aanwezige (virtuele) tabel die voor de gebruiker tijdelijk wordt samengesteld uit één of meer tabellen binnen de database. Een view presenteert zich naar de gebruiker als een gewone tabel waarop de normale SQL-opdrachten voor het selecteren en presenteren van gegevens kunnen worden uitgevoerd.

Binnen veel standaardprogrammatuur voor bestandsbeheer (zoals dBASE IV) en ook in het volgende hoofdstuk wordt de tabel uit de theorie van de relationele databases aangeduid als een bestand. De veldnaam is dan een tabelkolom en een record is synoniem met een rij uit de tabel.

NORMALISEREN

Bij het ontwikkelen van een informatiesysteem wordt onder meer de informatiebehoefte van de toekomstige gebruikers vastgesteld. Dit is het begin van de informatieanalyse. Het resultaat van de informatie-analyse is, dat bekend is welke gegevens moeten worden verzameld.

Door middel van gegevensanalyse wordt hierna bepaald hoe die gegevens het beste kunnen worden gegroepeerd en welke onderlinge verbanden er tussen de gegevens bestaan.

De groepen gegevens met hun onderlinge verbanden worden bij de gegevensanalyse beschreven in een logische gegevensstructuur. In de huidige toepassingen wordt de logische gegevensstructuur gebruikt bij het vastleggen en opvragen van de gegevens door de gebruiker.

Hierna wordt gekeken wat de meest geschikte fysieke gegevensstructuur of opslagstructuur in de computer is. Daarbij gaat het vooral om:

- het opslagmedium (magneetband, magneetschijf enz.);
- de organisatievorm (sequentieel, direct, relationele database enz.).

De fysieke gegevensstructuur zal bijna altijd afwijken van de logische structuur. De programmatuur zorgt hierbij voor de noodzakelijke onderlinge afstemming.

In dit hoofdstuk gaat het om het vaststellen van de logische structuur.

Voordat verder gegaan wordt met de gegevensanalyse, lichten we een aantal begrippen toe die hierbij worden gehanteerd.

Object: een zaak waarin we geïnteresseerd zijn en waarover informatie nodig is, bijvoorbeeld de klant bij een handelsonderneming.

Entiteit: de beschrijving van een object in een gegevensverzameling, bijvoorbeeld van elke klant naam, adres, woonplaats en telefoonnummer.

Attribuut: de algemene aanduiding van één bepaald kenmerk of eigenschap van een object of een entiteit, bijvoorbeeld naam, adres, woonplaats en telefoonnummer.

Attribuutwaarde: de waarde die een attribuut kan aannemen, bijvoorbeeld Pietersen, Jansen, Clarijs als naam van een klant.

Groep: een aantal bij elkaar behorende kenmerken, die samen een deel van een object of entiteit vormen, bijvoorbeeld de NAW-gegevens van een klant.

Occurrence: de beschrijving van een bepaald object of een entiteit; hierbij hebben alle attributen behorende tot dat object of die entiteit een bepaalde waarde, bijvoorbeeld: Jansen, Vleerakker 9, Wouw, 3456.

De volgende series begrippen zijn nagenoeg synoniemen van elkaar:

1. entiteitenverzameling, tabel en bestand;
2. object, entiteit, occurrence, tabelrij en record;
3. kenmerk, attribuut, eigenschap, rubriek, element, tabelkolom en veld.

VOORBEELD

KLANT

Naam	Adres	Woonplaats	Telefoon
Jansen	Vleerakker 9	Wouw	3456
Pietersen	Markt 14	Roosendaal	56789

Van het object KLANT worden de kenmerken (attributen) naam, adres, woonplaats en telefoonnummer belangrijk geacht. Twee klanten (entiteiten) worden in twee occurrences met vier attribuutwaarden per occurrence beschreven. Men kan ook zeggen dat tabel KLANT bestaat uit 2 rijen en 4 kolommen.

In dit hoofdstuk wordt hierna, om vertwarring te voorkomen, gebruik gemaakt van de volgende begrippen:

- 1 bestand;
- 2 record;
- 3 veld.

In elke organisatie worden bestanden door verschillende personen en/of afdelingen gebruikt. Daarom moeten er afspraken gemaakt worden over de schrijfwijze, de grootte, het aantal, het type, het doel enz.

Daarnaast moet voorkomen worden dat dezelfde gegevens op verschillende plaatsen in de organisatie worden bijgehouden. Anders zou het kunnen gebeuren dat klantgegevens zowel bij de Inkoopafdeling als bij de afdeling Magazijn en de afdeling Administratie worden bijgehouden. Hierdoor ontstaat een zekere overtuilingheid ofwel *redundantie* aan gegevens.

Onder redundantie wordt hier verstaan: het meermalen voorkomen van dezelfde gegevens binnen één informatiesysteem.

VOORBEELD

De afdeling Verkoop heeft een bestand KLANTEN met daarin o.a. de volgende velden: KLANTNAAM; KLANTADRES; KLANTPOSTC; KLANTPLAATS.

De afdeling Magazijn heeft een bestand AFNEMERS met daarin o.a. de volgende velden AFNNAAM; AFNADRES; AFNPOSTC; AFNPLAATS.

Beide afdelingen houden in deze verschillende bestanden gegevens bij over dezelfde klanten.

De moeilijkheid hierbij is niet alleen dat er een grote verspilling optreedt, maar dat zich ook problemen gaan vormen bij het onderhoud van de bestanden. Bij één mutatie op een bepaald gegeven moeten nu op verschillende plaatsen de bestanden worden bijgewerkt. Buiten het extra werk zijn er ook meer kansen op het maken van fouten bij het afzonderlijk inbrengen van de mutatie.

Bovendien is het te verwachten dat er een tijdsverschil zal optreden tussen het aanbrengen van de mutaties in de bestanden. Hierdoor is het mogelijk dat gegevens in de afzonderlijke bestanden gedurende een bepaalde periode niet aan elkaar gelijk zijn. Er ontstaan dan tegenstrijdigheden binnen dezelfde gegevensverzameling. Men spreekt in dit verband dan van *inconsistente* bestanden. Het is duidelijk dat inconsistentie hoe dan ook tegengegaan moet worden.

VOORBEELD

Bij verhuizing van een klant kan de adreswijziging bij het magazijn al doorgevoerd zijn, terwijl deze bij de afdeling Inkoop nog niet verwerkt is.

Het doel van de gegevensanalyse is het voorkomen van voornoemde problemen.

Voordat de gegevensanalyse kan plaatsvinden, zal eerst bepaald moeten worden welke gegevens er geanalyseerd moeten worden. De te creëren bestanden moeten dienen om aan de informatie-behoef-ten van de organisatie te voldoen. Onderzoek naar de informatiebehoefte leveren een ongestructureerde verzameling benodigde gegevens op. Door analyse kan hierin een zodanige structuur aangebracht worden, dat redundantie en inconsistentie zich niet zullen voordoen.

Een van de toe te passen technieken om gegevens te structureren is de normalisatietechniek van Codd. Met behulp hiervan worden de gegevens zodanig in bestanden ondergebracht, dat er geen onregelmatigheden optreden bij het onderhoud van de bestanden.

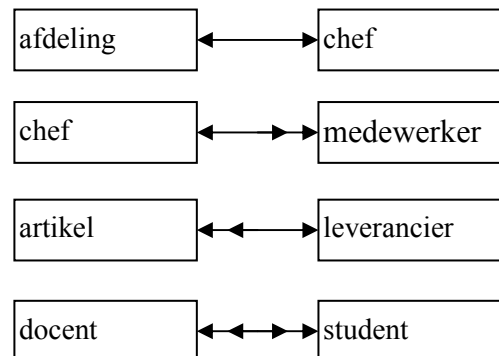
De relatie, het verband of de koppeling tussen de bestanden wordt afgebeeld in de vorm van een:

- 1 Bachman-diagram
- 2 strokendiagram.

In een Bachman-diagram worden de relaties tussen de bestanden, zonder vermelding van de record-indeling, als volgt grafisch weergegeven: elk bestand wordt afgebeeld in de vorm van een rechthoek. De relatie tussen de bestanden wordt aangegeven met een of meer pijlen.

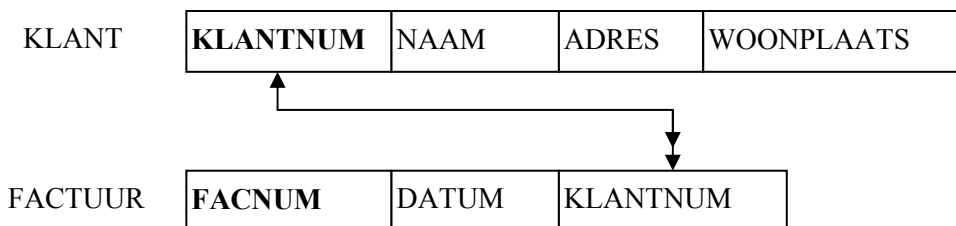
De volgende drie relaties worden hierbij onderscheiden:

- 1 de één-op-één-relatie; bij elke afdeling hoort precies één chef, bij elke chef hoort precies één afdeling
- 2a de één-op-n-relatie; bij elke chef hoort een aantal van 0, 1 of meer medewerkers, bij elke medewerker hoort precies één chef; let hierbij op het aantal pijlpunten.
- 2b de n-op-één-relatie; bij elk artikel hoort precies één leverancier; bij elke leverancier hoort een aantal van 0, 1 of meer artikelen
- 3 de n-op-m-relatie; bij elke docent hoort een aantal van 0, 1 of meer studenten; bij elke student hoort een aantal van 0, 1 of meer docenten



In een strokendiagram worden de relaties tussen de bestanden als volgt afgebeeld:

- 1 Van elk bestand worden de velden van het record achter elkaar in de vorm van rechthoeken weergegeven.
- 2 Het sleutelveld of de sleutelvelden worden hierbij onderstreept
- 3 De relatie tussen de bestanden wordt met een of meer pijlpunten aangegeven.



Het normaliseren volgens de methode van Codd bestaat meestal uit verschillende stappen. Hierbij worden stap voor stap de afzonderlijke bestanden met de bijbehorende record-indelingen bepaald. Elke record-indeling bestaat hierbij uit een of meer velden, zowel met sleutelgegevens als zonder sleutelgegevens.

Voordat met het eigenlijke normaliseren wordt begonnen, dient de volgende stap gedaan te worden:

- Schrijf alle benodigde gegevens op uit de informatiebehoefte. Geef hierbij elk gegeven een unieke veldnaam
- verwijder de procesgegevens uit de gegevensverzameling. Een procesgegeven is een gegeven dat direct of indirect uit andere gegevens is af te leiden.

Hierna gaan we pas 'echt' normaliseren:

- Breng de gegevens in de eerste normaalvorm: Verwijder uit het oorspronkelijke bestand de gegevens die meermalen (kunnen) voorkomen (repeating group).
- Breng de gegevens in de tweede normaalvorm: Verwijder de gegevens die niet geheel afhankelijk zijn van de volledig samengestelde sleutel van het bestand.
- Breng de gegevens in de derde normaalvorm: Verwijder de gegevens die ook afhankelijk zijn van andere gegevens, die niet tot de sleutel van het bestand behoren.

VOORBEELD 1

Een administratiekantoor verricht allerlei soorten diensten ten behoeve van haar klanten, zoals het opstellen van belastingaangiften, het controleren van de administratie, het samenstellen van jaarrekeningen en rapporten en het geven van organisatie-adviezen.

Elke te verrichten soort dienst vereist een specialistische kennis en is daarom ondergebracht in een taakgroep. Elke taakgroep bestaat uit één of meer medewerkers en elke medewerker behoort tot één taakgroep.

Van elke voor een klant te verrichten opdracht wordt een budgetbedrag vastgesteld. Voor een opdracht kunnen een of meer taakgroepen ingeschakeld worden.

Op het einde van elke maand wenst de directie van het administratiekantoor een overzicht met van elke klant het budgetbedrag en de specificatie van de werkelijke personeelskosten. Het uurtarief is afhankelijk van de medewerker die de werkzaamheden uitvoert.

Dit overzicht ziet er als volgt uit:

<i>klant- nummer</i>	<i>naam klant</i>	<i>bedrag budget</i>	<i>nummer medewerker</i>	<i>naam medewerker</i>	<i>nummer taakgroep</i>	<i>naam taakgroep</i>	<i>aantal uren</i>	<i>uurtarief per uur</i>	<i>totaal- bedrag</i>
1001	AAAA	3000	208	BOV	52	ORGAN	40	60,00	2400,00
			225	KAL	56	JAARR	20	40,00	800,00
								Totaal	3200,00
1003	DDD	2500	212	DIR	54	CONTR	20	50,00	1000,00
			225	KAL	56	JAARR	10	40,00	400,00
			231	OOM	54	CONTR	15	60,00	900,00
								Totaal	2300,00
...	...	...	...	...	..	...	..	...	...

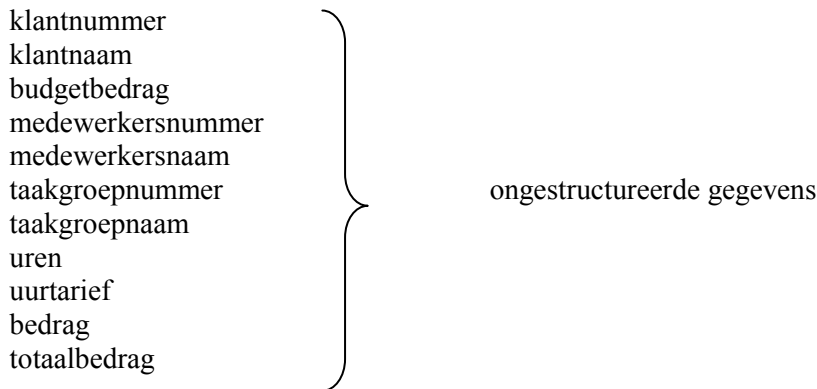
Stap 0:

Verzamel de relevante gegevens aan de hand van de informatiebehoefte.

Actie 0.1:

- a. Inventariseer uit de informatiebehoefte alle benodigde gegevens.
- b. Geef elk gegeven een unieke veldnaam.

De volgende gegevens zijn nodig om het gewenste overzicht te kunnen vervaardigen:

*Actie 0.2:*

Geef in de gegevensverzameling de procesgegevens; aan en verwijder deze.

Een *procesgegeven* is een gegeven dat direct of indirect uit andere gegevens is af te leiden; het hoeft niet te worden vastgelegd in een bestand. In dit voorbeeld staan de volgende procesgegevens:

- a bedrag (per medewerker) = aantal uren x uurtarief;
- b totaalbedrag (per klant) = som van de bedragen per medewerker.

klantnummer
 klantnaam
 budgetbedrag
 medewerkersnummer
 medewerkersnaam
 taakgroepnummer
 taakgroepnaam
 uren
 uurtarief
 bedrag → procesgegeven
 totaalbedrag → procesgegeven

Actie 0.3:

- a Geef een naam aan het bestand.
- b Bepaal de sleutel van het bestand en onderstreep deze.

Een mogelijke naam voor het bestand is KLANTEN. De mogelijke sleutel is het klantnummer. Het overzicht is namelijk ingedeeld naar klantnummer. Na stap 0 moeten de volgende gegevens worden genormaliseerd:

KLANT = naam bestand
klantnummer = sleutel
 klantnaam
 budgetbedrag
 medewerkersnummer

medewerkersnaam
 taakgroepnummer
 wakgroepnaam
 uren
 uurtarief

Stap 1:

Verwijder uit het oorspronkelijke bestand de gegevens die meermalen (kunnen) voorkomen bij dezelfde sleutel.

Actie 1.1:

Geef in het bestand de gegevens in die per klant bij elke sleutel meermalen (kunnen) voorkomen (= de *repeating group*).

klant- nummer	naam klant	bedrag budget	nummer medewerker	naam medewerker	nummer taakgroep	naam taakgroep	aantal uren	uurtarief per uur	totaal- bedrag
1001	AAAA	3000	208	BOV	52	ORGAN	40	60,00	2400,00
			225	KAL	56	JAARR	20	40,00	800,00
								Totaal	3200,00
1003	DDD	2500	212	DIR	54	CONTR	20	50,00	1000,00
			225	KAL	56	JAARR	10	40,00	400,00
			231	OOM	54	CONTR	15	60,00	900,00
								Totaal	2300,00
...	...	...	...	...	..	...	..	...	...

In het overzicht komen achter de diverse klantnummers de volgende gegevens meermalen voor:

medewerkersnummer	}	repeating group
medewerkernaam		
taakgroepnummer		
taakgroepnaam		
uren		
uurtarief		

Actie 1.2:

- Verwijder de repeating group uit het oorspronkelijke bestand en vorm hiervoor een nieuw bestand.
- Geef een naam aan het nieuw gevormde bestand. In dit voorbeeld is gekozen voor de naam WERKREGEL.
- Neem de sleutel van het oorspronkelijke bestand op in die van het nieuwe bestand. Hiermee wordt de koppeling met het oorspronkelijke bestand gehandhaafd. Indien in het oorspronkelijke bestand een samengestelde sleutel voorkomt, kan, om de koppeling met het oorspronkelijke bestand te handhaven, meestal worden volstaan met overneming van een deel van de samengestelde sleutel. In dit voorbeeld wordt het klantnummer overgenomen.
- Bepaal van het nieuwe bestand de sleutel en onderstreep deze. Een mogelijke sleutel is het medewerkersnummer. Dezelfde medewerker kan echter in dit voorbeeld meermalen in het bestand voorkomen. Neem daarom de overgenomen sleutel van het oorspronkelijke bestand op in de sleutel van het nieuwe bestand. Hiermee wordt de sleutel uniek gemaakt.

In dit voorbeeld ontstaat bestand WERKREGEL. In bestand WERKREGEL wordt de samengestelde sleutel gevormd door klantnummer en medewerkersnummer.

Na actie 1. 1 en actie 1.2 zijn de volgende bestanden gevormd:

KLANTEN	WERKREGEL	naam bestand
<u>klantnummer</u>	<u>klantnummer</u>	} record-indeling
klantnaam	<u>medewerkersnummer</u>	
budgetbedrag	medewerkersnaam	
	taakgroepnummer	
	taakgroepnaam	
	uren	
	uurmrief	

Actie 1.3:

Herhaal actie 1. 1 en actie 1. 2 voor het nieuw gevormde bestand.

In dit voorbeeld bevat het nieuw gevormde bestand geen gegevens meer die herhaald kunnen voorkomen. In de eerste normaalvorm zijn nu de bestanden KLANTEN en WERKREGEL ontstaan.

Stap 2:

Verwijder de gegevens die niet geheel afhankelijk zijn van de volledig samengestelde sleutel van het bestand.

Deze stap kan alleen toegepast worden op de bestanden die een samengestelde sleutel hebben. Alleen bij deze bestanden kan een gegeven gedeeltelijk afhankelijk zijn van de volledige sleutel.

Actie 2. 1

Zet achter elk gegeven van een bestand met een samengestelde sleutel, van welke sleutel of sleutels dit gegeven afhankelijk is.

klantnummer	
medewerkernummer	
medewerkersnaam	- afhankelijk van medewerkersnummer
taakgroepnummer	- afhankelijk van medewerkersnummer
taakgroepnaam	- afhankelijk van medewerkersnummer
uren	- afhankelijk van klantnr. + medewerkersnr.
uurtarief	- afhankelijk van medewerkersnummer

Actie 2.2:

- Formeer een nieuw bestand of bestanden van de gegevens die slechts van één en dezelfde sleutel afhankelijk zijn en verwijder deze gegevens uit het oorspronkelijke bestand.
- Geef een naam aan het nieuw gevormde bestand.
- Bepaal van de nieuwe bestand(en) de sleutel en onderstreep deze.

In dit voorbeeld ontstaat het nieuwe bestand MEDEWERKER. Al deze gegevens zijn afhankelijk van het medewerkersnummer, dat tevens de sleutel van dit bestand is. Bestand MEDEWERKER bestaat uit de volgende gegevens:

medewerkernummer
medewerkersnaam
taakgroepnummer
taakgroepnaam
uurtarief

Opmerkingen:

- 1 Bij het verwijderen van de gegevens uit het oorspronkelijke bestand moet ervoor gezorgd worden, dat de sleutel van dit bestand volledig intact blijft.
- 2 Om de koppeling tussen de bestanden te handhaven, dient de sleutel van het nieuwe bestand ook in het oorspronkelijke bestand voor te komen. Daarom wordt in het nieuwe bestand de al bestaande sleutel van het oorspronkelijke bestand overgenomen.

In de tweede normaalvorm zijn de volgende bestanden gevormd:

KLANTEN	WERKREGEL	MEDEWERKER
<u>klantnummer</u>	<u>klantnummer</u>	<u>medewerkersnummer</u>
klantnaam	<u>medewerkersnummer</u>	medewerkersnaam
budgetbedrag	uren	taakgroepnummer
		taakgroepnaam
		uurtarief

Stap 3:

Verwijder de gegevens die ook afhankelijk zijn van andere gegevens die niet tot de sleutel van het bestand behoren. Deze stap kan alleen worden uitgevoerd op bestanden die uit twee of meer gegevens bestaan.

Actie 3.1:

Ga van eik bestand na van welke andere, gegevens binnen hetzelfde bestand elk gegeven ook afhankelijk is.

Bij bestand KLANTEN zijn de twee gegevens klantnaam en budgetbedrag niet van elkaar afhankelijk. Bestand WERKREGEL bestaat slechts uit één gegeven en komt dus niet in aanmerking. Bij bestand MEDEWERKER heeft elk taakgroepnummer dezelfde naam. Het taakgroepnummer en de taakgroepnaam zijn van elkaar afhankelijk. De overige gegevens zijn niet van een ander gegeven afhankelijk.

taakgroepnummer taakgroepnaam

Actie 3.2:

- a. Formeer nieuwe bestanden van de gegevens die afhankelijk van elkaar zijn en verwijder deze gegevens uit het oorspronkelijke bestand.
- b. Bepaal van elk nieuw gevormd bestand de sleutel en onderstreep deze. Het sleutelgegeven dient ook voor te komen in het oorspronkelijke bestand om de koppeling tussen de bestanden te handhaven.

In de derde normaalvorm zijn de volgende bestanden gevormd en is het normaliseren voltooid:

KLANTEN	WERKREGEL	MEDEWERKER	TAAKGROEP
<u>klantnummer</u>	<u>klantnummer</u>	<u>medewerkers-</u>	<u>taakgroepnummer</u>
klantnaam	<u>medewerkers-</u>	<u>nummer</u>	taakgroepnaam
budgetbedrag	<u>nummer</u>	medewerkersnaam	
	uren	taakgroepnummer	
		uurtarief	

De genormaliseerde logische gegevensstructuur van de bestanden wordt als volgt weergegeven:

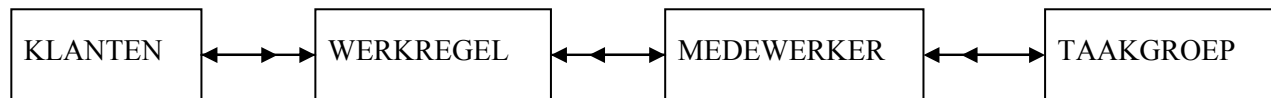
KLANTEN (KLANTNUMMER, KLANTNAAM, BUDGETBEDRAG)

WERKREGEL (KLANTNUMMER, MEDEWERKERSNUMMER, UREN)

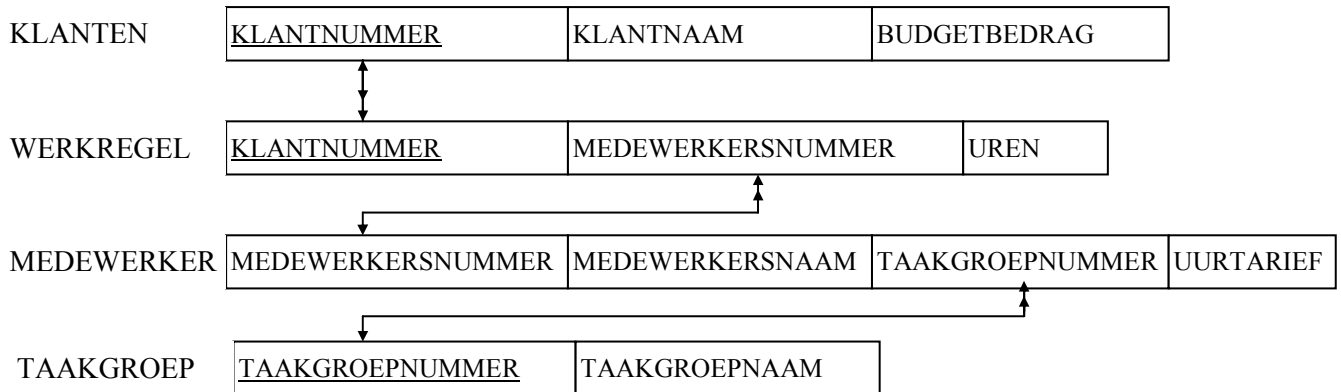
MEDEWERKER (MEDEWERKERSNUMMER, MEDEWERKERSNAAM, TAAKGROEPNUMMER, UURTARIEF)

TAAKGROEP (TAAKGROEPNUMMER, TAAKGROEPNAAM)

Bachman-diagram



Strokendiagram.

*Controle:*

Tussen de bestanden dient altijd een één-op-n-relatie te bestaan. Bovendien geldt in de derde normaalvorm, dat elk gegeven in het record betrekking heeft op de sleutel, de hele sleutel en niets anders dan de sleutel. Indien dit niet het geval is, is er tijdens het normaliseren iets fout gegaan.

VOORBEELD 2

De directie van een productiebedrijf wenst van *elke* afdeling per voorman het volgende overzicht:

Afdeling: 101 spuitrij			Afdeling: 102 zagerij		
nummer voorman	personeels- nummer	uren	nummer voorman	personeels- nummer	uren
1010	100	20	1020	110	32
	101	15		114	16
	104	30	1024	114	18
1012	101	25		116	36
	103	40			
	104	10			

Opmerking: Op één afdeling kunnen één of meer voormannen werkzaam zijn. Een personeelslid is niet gebonden aan een vaste voorman. Een voorman is gebonden aan een vaste afdeling.

Actie 0.1:

De volgende gegevens zijn nodig om het gewenste overzicht te kunnen vervaardigen:

afdelingsnummer
afdelingsnaam
nummer_voorman
personeelsnummer
uren

Actie 0.2:

In de gegevensverzameling komen geen procesgegevens voor.

Actie 0.3:

Een mogelijke naam voor het bestand is AFDELING. De mogelijke sleutel is het afdelingsnummer.

Na stap 0 moeten de volgende gegevens worden genormaliseerd

AFDELING = naam bestand

afdelingsnummer = sleutel

afdelingsnaam
nummer_voorman
personeelsnummer
uren

Actie 1.1:

De repeating group (bij sleutel afdelingsnummer) bestaat uit de volgende gegevens:

nummer_voorman personeelsnummer uren

Actie 1.2:

Het afdelingsnummer wordt uit het oorspronkelijke bestand overgenomen. Een mogelijke sleutel voor bestand AFDBLOK is het nummer_voorman. In dit geval is geen samengestelde sleutel nodig, omdat hetzelfde nummer voorman slechts eenmaal in het bestand kan voorkomen.

Na actie 1.2 zijn de volgende bestanden gevormd:

AFDELING	AFDBLOK	} naam bestand
<u>afdelingsnummer</u>	<u>nummer voorman</u>	

afdelingsnaam	afdelingsnummer	record-indeling
	personeelsnummer	
	uren	

Actie 1.3:

Bestand AFDBLOK bevat nog de volgende repeating group bij sleutel nummer_voorman:
personeelsnummer en uren

Het nummer_voorman wordt uit het oorspronkelijke bestand overgenomen. Een mogelijke sleutel voor het nieuwe bestand UREN, waarin we deze gegevens plaatsen, is het personeelsnummer. Omdat een personeelsnummer meermalen in het bestand kan voorkomen, wordt het nummer_voorman uit het oorspronkelijke bestand in de *sleutel* opgenomen. De samengestelde sleutel nummer_voorman en personeelsnummer zorgt voor een unieke sleutel.

Na actie 1.3 is de eerste stap van het normalisatieproces voltooid en zijn de volgende bestanden gevormd:

AFDELING	AFDBLOK	UREN
<u>afdelingsnummer</u>	<u>nummer voorman</u>	<u>nummer voorman</u>
afdelingsnaam	afdelingsnummer	<u>personeelsnummer</u>
		uren

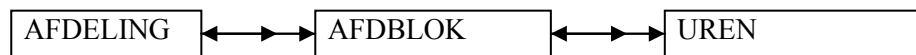
De eerste normaalvorm is tevens de laatste, omdat:

- in bestand UREN uren afhankelijk is van de gehele sleutel (stap 2);
- de bestanden niet uit twee of meer gegevens bestaan (stap 3).

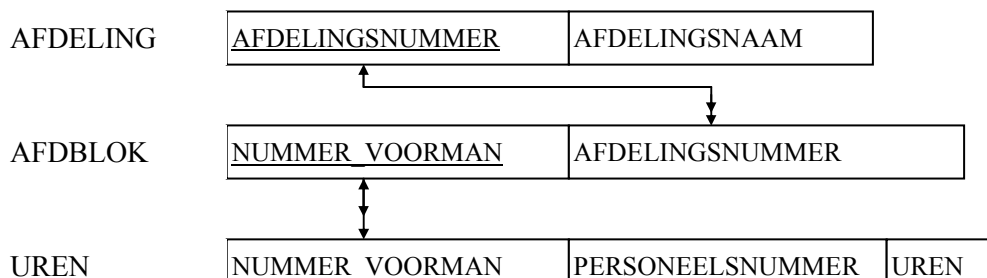
De genormaliseerde logische gegevensstructuur van de bestanden wordt als volgt weergegeven:

AFDELING	(<u>AFDELINGSNUMMER</u> , AFDELINGSNAAM)
AFDBLOK	(<u>NUMMER_VOORMAN</u> , AFDELINGSNUMMER)
UREN	(<u>NUMMER_VOORMAN</u> , <u>PERSONEELSNUMMER</u> , UREN)

Bachman-diagram



Strokendiagram.



Controle: Tussen de bestanden bestaat een een-op-n relatie en elk gegeven in de records heeft betrekking op de volledige sleutel.

VOORBEELD 3

De afdeling Magazijn wenst dagelijks het volgende overzicht, waarop de af te geven artikelen staan ten behoeve van de verschillende afdelingen:

<i>afdelings-nummer</i>	<i>afdelings-naam</i>	<i>artikel-nummer</i>	<i>artikel-naam</i>	<i>aantal</i>
10	ZAG	126	ROI	18
		145	ZO8	4
		210	B42	42
11	MAC	210	B42	88
		214	ZKR	17
		318	V3P	2

Actie 0.1:

De volgende gegevens zijn nodig om het gewenste overzicht te kunnen vervaardigen:

afdelingsnummer
afdelingsnaam
artikelnummer
artikelnaam
aantal

Actie 0.2:

In de gegevensvenweling komen geen procesgegevens voor.

Actie 0.3:

Een mogelijk naam voor het bestand is AFDELING.

De mogelijke sleutel is het afdelingsnummer,

Na stap 0 moeten de volgende gegevens worden genormaliseerd:

AFDELING = naam bestand

afdelingsnummer = sleutel

afdelingsnaam
artikelnummer
artikelnaam
aantal

Actie 1.1:

De volgende gegevens komen meermalen voor bij sleutel afdelingsnummer:

artikelnummer artikelnaam aantal

Actie 1.2:

Het afdelingsnummer wordt uit het oorspronkelijke bestand overgenomen. Een mogelijke sleutel voor bestand ARTIKELAFGIFTE is het artikelnummer. Omdat elk artikelnummer meermalen in het bestand kan voorkomen, wordt het afdelingsnummer in de sleutel opgenomen. De samengestelde sleutel bestaat uit afdelingsnummer en artikelnummer.

Actie 1.3:

Bestand ARTIKELAFGIFTE bevat geen gegevens die meermalen bij de sleutel voorkomen.

In de eerste normaalvorm zijn de volgende bestanden gevormd:

AFDELING

ARTIKELAFGIFTE

afdelingsnummer
afdelingsnaam

afdelingsnummer
artikelnummer
artikelnaam
aantal

Actie 2.1:

afdelingsnummer
artikelnummer
artikelnaam
aantal

- afhankelijk van artikelnummer
- afhankelijk van artikelnr. + afdelingsnr,

Actie 2.2:

Een mogelijke sleutel voor bestand ARTIKEL is het artikelnummmmer.

Na de tweede normaalvorm zijn de volgende bestanden gevormd:

AFDELING	ARTIKELAFGIFTE	ARTIKEL
afdelingsnummer	afdelingsnummer	artikelnummer
afdelingsnaam	artikelnummer	artikelnaam
	aantal	

De tweede normaalvorm is tevens de laatste stap van het normaliseren.

De genormaliseerde logische gegevensstructuur van de bestanden wordt als volgt weergegeven:

AFDELING (AFDELINGSNUMMER, AFDELINGSNAAM)

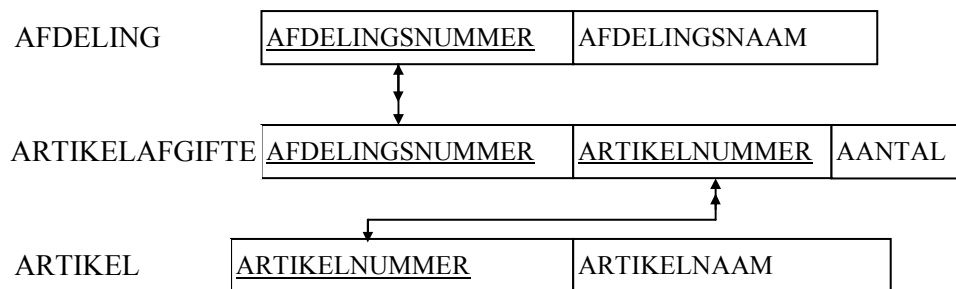
ARTIKELAFGIFTE (AFDELINGSNUMMER, ARTIKELNUMMER, AANTAL)

ARTIKEL (ARTIKELNUMMER, ARTIKELNAAM)

Bachman-diagram



Strokendiagram



Controle:

Tussen de bestanden bestaat een één-op-n-relatie. Elk gegeven in de records heeft betrekking op de volledige sleutel.

Ontwerp bij de volgende opgaven de benodigde bestanden volgens de drie stappen van Codd met de bijbehorende Bachmandiagrammen. Geef tevens aan welke bestanden in elke normaalvorm gevormd zijn.

1. Op het einde van elke maand wenst de afdeling INKOOP het volgende overzicht:

artikel-nummer	artikel-naam	voorraad in stuks	leveranciers-nummer	leveran-naam
1010	ACI	50	6071	PH
1012	CDP	10	W52	SIE
1015	KVE	60	6071	PH
1020	VTW	22	6113	MIE
1021	STR	86	6052	SIE
...	...	...	...	...

Een artikel wordt altijd bij dezelfde leverancier gekocht. Van een leverancier worden een of meer artikelen gekocht.

2. Elke week wenst de loonadministratie het volgende overzicht:

afdelings-nummer	afdeling-naam	personeels-nummer	personeels-naam
20	VOORBEW	1020	Pietersen
		1062	Jansen
21	PRODUC	1015	Adriane
		1047	Karels
		1075	Viersen
22	...		...

Een personeelslid is slechts op één afdeling werkzaam.

3. De afdeling calculatie wenst periodiek het volgende overzicht:

produkt-num	produkt-naam	artikel-nummer	aantal	prijs per stuk	waarde
521	BOVK	1060	2	4,00	8,00
		1081	4	8,00	32,00
526	ONDK	1020	1	52,00	52,00
		1060	4	4,00	16,00
		1090	3	7,00	21,00
532	LADE	1060	6	4,00	24,00
		1081	1	8,00	8,00
		1090	2	7,00	14,00
538	DEUR	...			

Een produkt is opgebouwd uit een of meer artikelen. Een artikel kan in meerdere produkten voorkomen.

4. Op het einde van elk kwartaal wordt van de klanten van de firma JOKIMEX het volgende overzicht gemaakt. De klanten zijn ingedeeld per categorie en alle facturen van een bepaalde klant staan bij elkaar gegroepeerd.

Categorie: 01 Groothandel

Klantnummer	Naam klant	Factuurnr	factuur bedrag
1001	Pieters	9109	f 100,46
		9121	f 135,55
		9133	f 235,67
		totaal	f 471,68
1009			
			
		totaal	
.....			
			
			
		totaal	

Categorie: 02 Kleinhandel

Klantnummer	Naam klant	Factuurnr	factuur bedrag
2001	Peeter	6109	f 100,46
		6121	f 135,55
		6133	f 235,67
		totaal	f 471,68
2009			
			
		totaal	

OPDRACHTEN

5. Een aannemer wenst een wekelijks kostenoverzicht van zijn projecten volgens onderstaand model:

Project- nummer	Project- naam	Code kosten	omschrijving kosten	bedrag
....				
....				
....				
....				

6. De afdeling Verkoop van een handels-onderneming ontvangt via haar vertegenwoordigers de afgesloten orders volgens onderstaand model:

Ordernummer :1571 Klantnummer :23575
 Datum :25mei 1991 Klantnaam :Andriessen
 Adres :Meerstraat 5
 Vertegenwoordiger :L. v Leeuw Woonplaats :4257 Nisp

Artikel- num	Artikel- naam	aantal	prijs p stuk	totaal bedrag	korting percenten	lever- datum
6571	TV 500	3	1950,-	5550,-	25	010791
3277	Cassette	4	320,-	1280,-	20	150691
6572	TV 400	1	2550,-	2550,-	18	020891
4781	video	6	950,-	5700,-	31	010791

Elk artikel komt op de order slechts eenmaal voor. De prijs per stuk voor een bepaald artikel is voor elke order gelijk. Het kortingspercentage en de leverdatum worden per orderregel door de vertegenwoordiger overeengekomen en kunnen per artikel en per order verschillen.

7. Een boekhandel gebruikt de volgende bestelbon om bij de uitgeverijen boeken te bestellen.

BOEKHANDEL LEES MAAR RAAK

Bestelnummer :91841 datum: 24 mei 2002
 Uitgever :011
 Naam :Educaboek
 Adres :Postbus 220
 Woonplaats :4123AB Calemberg

ISBN	TITEL	JAAR	SCHRIJVER	AANTAL
9011006135	Werk boek WORD	1989	Covree	12
9011006132	Excel	1989	Covree	15
9011006130	Word	1989	Covree	22

Het ISBN-nummer is een uniek nummer, waarmee elk boek eenduidig wordt geïdentificeerd. Er kunnen meer bestelorders lopen bij dezelfde uitgever. Ook kan het dat een boek meermalen besteld wordt.

8. Een CD-theek hanteert voor haar inkoop de volgende bestelformulieren:

Een boekhandel gebruikt de volgende bestelbon om bij de uitgeverijen boeken te bestellen.

CD-THEEK 'VAN ALLES WAT'

Ordernummer :24 datum: 24 mei 2002
 Leveranciersnummer :6
 Leverancier :Inter-Continental
 :Breeschot 12
 :4123AB Calemberg

CD-nr	Titel	Soort	Genre	Muzikant	Land	aantal
10	Free	Dubbel	Jazz	Jackson	USA	20
12	Angle	Normal	Rock	Anka	USA	30
..						

9. Een confectiebedrijf hanteert voor haar inkoop van stoffen de volgende bestellijst:

CONFECTIEBEDRIJF 'GOEDKOOP'

Bestelnummer :64 datum: 24 mei 2002
 Levnummer :6
 Naam :Inter-Continental
 Adres :Breeschot 12
 Plaats :4123AB Calemberg
 Telef. :07190-123456
 Inkoper :ZA
 Naam :K. Zakenman

Stofnr	Omschrijving	Breedte	Kleur	meter	prijs pm
20	Katoen413	1.64	rood	20	15.50
24	Viscosek3m	1.24	wit	10	20.00
12	Acryl50/30	1.40	paars	50	8.50
..					

ACCESS

Microsoft Access is een Database Management Systeem (DBMS), waarmee je gegevens kunt verzamelen, organiseren, opzoeken, weergeven en afdrucken.

Database

Een database is niets anders dan een verzameling bij elkaar horende objecten voor het beheren van gegevens. Het telefoonboek is een voorbeeld van een database waarin gegevens van bedrijven en particulieren staan. Hetzelfde geldt voor de ladenkasten of de kaartenbakken op een kantoor. De databases van Access bevatten objecten als tabellen, queries, formulieren en dergelijke.

Tabellen

In Access worden de gegevens opgeslagen in tabellen.

- De *kolom* van een tabel bevat steeds gegevens van dezelfde soort, zoals de namen van personen. Iedere kolom is een veld.
- Een rij van een tabel bevat gegevens die bij elkaar horen over de kolommen heen. Iedere rij is een record.

Tabellen die bij elkaar horen vormen dan een database.

	naam		
record 1			
record 2			

Relationele databases

Access is een relationele database, hetgeen betekent dat het mogelijk is om gegevens die op verschillende plaatsen staan opgeslagen (normaal gesproken in diverse tabellen), te koppelen. Als je bijvoorbeeld de naam van een auteur zou invoeren, kan Access de titel en het onderwerp van het boek uit de andere tabellen ophalen waarin deze gegevens staan, als deze tabellen aan elkaar zijn gekoppeld.

Primaire sleutelvelden - primary keys

Bij het invoeren van gegevens in tabellen maken we gebruik van *primaire sleutelvelden*. Deze voegen een uniek gegeven toe aan de tabel, waardoor een record uniek identificeerbaar is. Je kunt zelf sleutelvelden instellen of dit automatisch laten doen (Auto-numbering).

Formulieren - forms

Een formulieren is een soort 'kaart' op je scherm waarmee je eenvoudig gegevens kunt invoeren, lezen wijzigen en ook opzoeken. De gegevens worden namelijk per record overzichtelijk getoond. Deze formulieren kunt je ook afdrucken.

Queries

Een lijst die u opvraagt van alle boeken, die gisteren weer terug hadden moeten zijn en niet werden teruggebracht, wordt gemaakt met een query. Een query is een verzoek aan Access om informatie. Een query kun je maken met *SQL* (zie dictaat SQL) of met een *QBE* (query by example). Bij QBE kies je de tabellen en daaruit de gegevens die je wilt hebben.

Rapporten - reports

Rapporten zijn afdrucken op papier van tabellen of queries. Rapporten kun je desgewenst ook Afdrucken naar bestand.

We starten Access net zoals de andere Windows-programma's op via *Start* dan *Programs* en dan Microsoft Access. We krijgen dan het openingsscherm waar we kunnen kiezen om:

- een blank database te starten
- de database wizard te starten of
- een bestaande database te openen.

Het openingsscherm

Als we Access starten dan moeten we eerst kiezen of we een nieuwe database gaan maken , een 'wizard' willen starten of een bestaande database willen openen. We gaan een nieuwe database aanmaken door te kiezen voor Blank Access database.

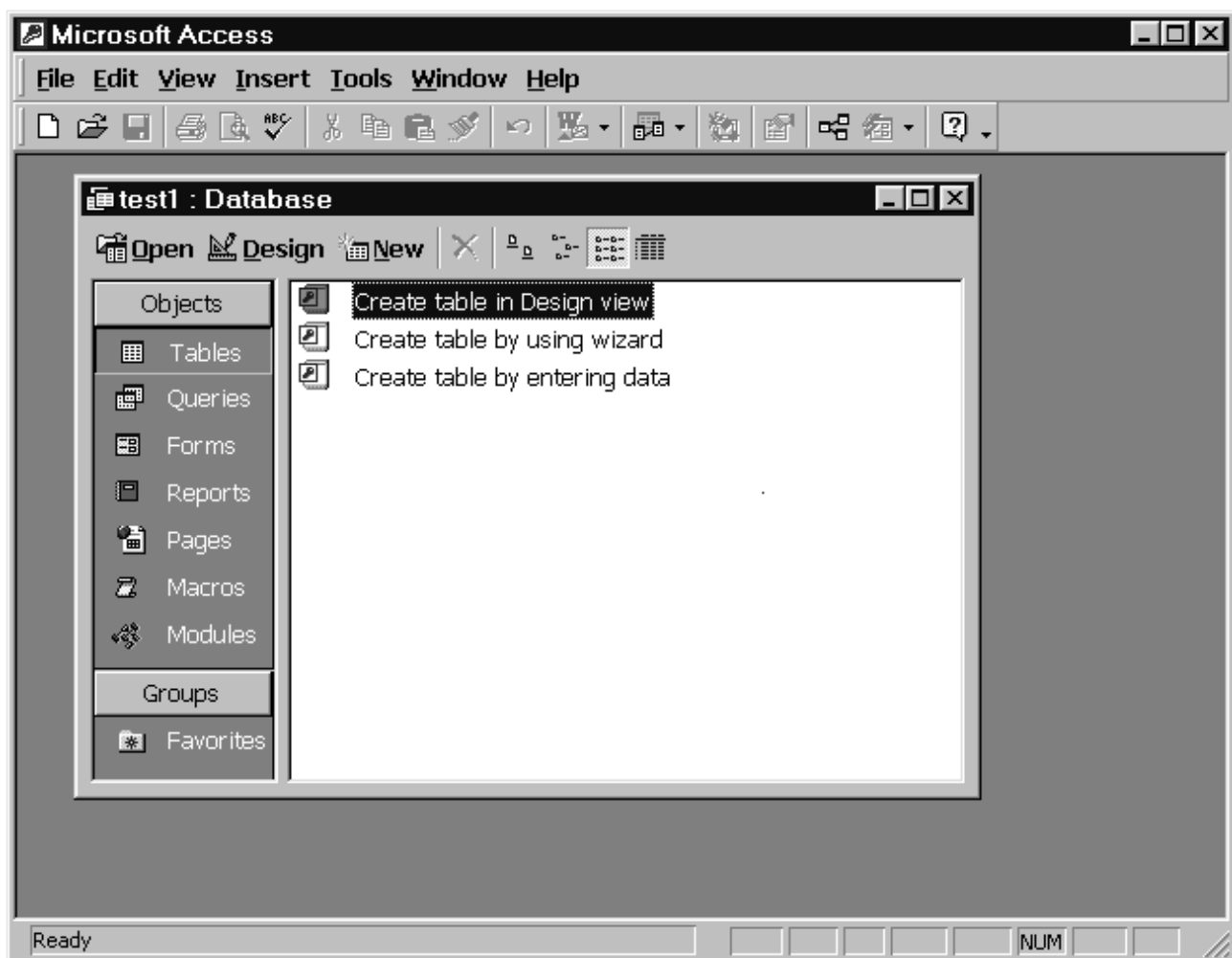
We krijgen dan een window en moeten als eerste de database een naam geven.

Daarna krijgen we het onderstaande scherm te zien, met de vertrouwde *Title bar*, *Menu bar*, *Toolbar(s)* en *Status bar*.

Op de toolbar zien we enkele nieuwe knoppen, zoals

-  Analyze om de tabellen met behulp van Access te onderzoeken op redundantie en inconsistentie
-  Relationships om de tabellen te koppelen
-  New object om een nieuwe tabel, rapport, formulier etc te maken.

In het database-venster zien we links de verschillende database-objecten, zoals Tables, Queries, Forms en Reports. Deze klikken we aan als we respectievelijk een tabel, query, formulier of rapport willen maken. Boven staan de knoppen *Open*, *Design* en *New* om een nieuwe tabel, query, etc te maken, te openen of te ontwerpen.



De verschillende onderdelen van Access worden verder uitgewerkt aan de hand van opdracht 6 uit het hoofdstuk Normaliseren

Een tabel maken.

Om een tabel te maken ga je als volgt te werk:

- Klik op het tabblad Tables en rechts op New.
- Kies dan voor design View.
- Je krijgt dan een window waarin je de veldnamen, de gegevenssoort en opmerkingen kunt intypen. Er zijn verschillende gegevenssoorten mogelijk, namelijk text, memo, number, date/time, currency (valuta), autonumber, yes/no, etc.
- Dan kies je de primaire sleutel(s) kiezen door op zijn knop te klikken. Bij een samengestelde sleutel moeten we eerst de rijen selecteren die de sleutels moeten vormen terwijl je de CTRL-knop vasthoudt en dan de knop klikken.

**Toolbar**

We krijgen ook een andere Toolbar. daarop zien we:

- de knop view die de mogelijkheid geeft tot
 - Design view om de tabel, query of formulier, te ontwerpen.
 - Datasheet view om de tabel te vullen nadat we hem ontworpen hebben.
- de knop om de primaire sleutel te kiezen,
- indexen
- Rij invoegen
- rij verwijderen en
- eigenschappen (properties)

**opdracht:**

- Maak de vier tabellen zoals hieronder: Let ierbij op de Data types en de sleutels!

bestelregel : Table			
Field Name	Data Type	Description	
ordernr	Number		
artnr	Number		
aantal	Number		
korting	Number		
leverdatum	Date/Time		

klant : Table			
Field Name	Data Type	Description	
Klantnr	Number		
Knaam	Text		
Kladres	Text		
Klwoonplaats	Text		

bestel : Table			
Field Name	Data Type	Description	
ordernr	Number		
datum	Date/Time		
vertegenwoordiger	Text		
Klantnr	Number		

artikel : Table			
Field Name	Data Type	Description	
artnr	Number		
artomschr	Text	omschrijving	
prijs	Currency		

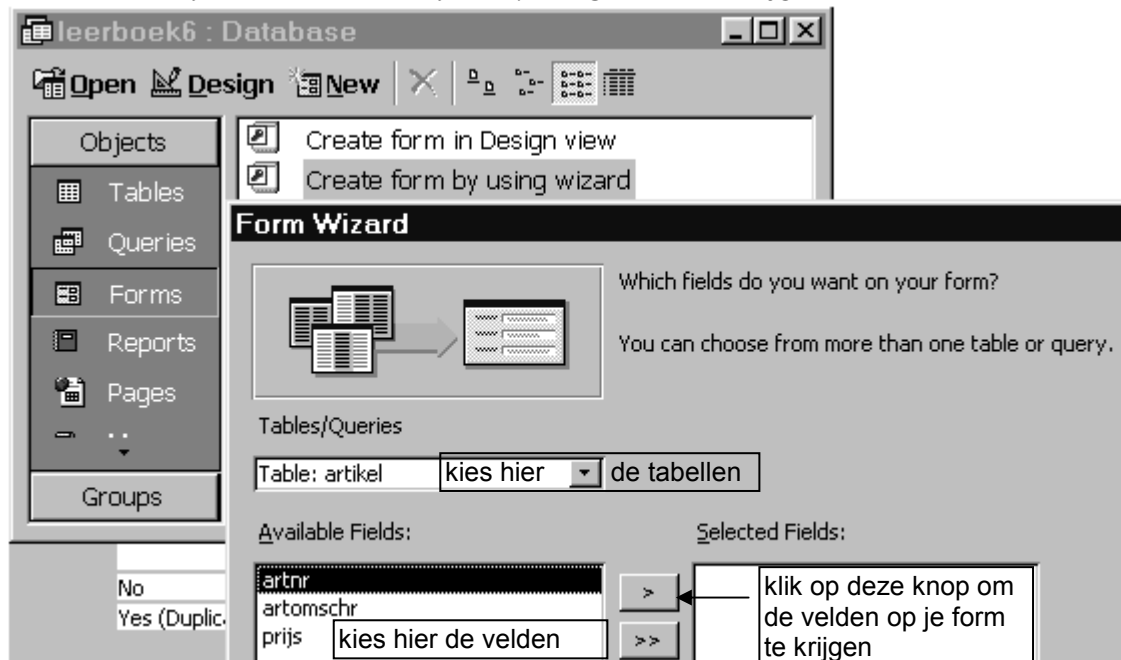
FORMS/FORMULIEREN

Formulieren maken

Als de opzet van de tabellen gemaakt is dan kun je de gegevens invoeren door de tabellen in datasheet view te bekijken, of door een *formulier* te maken.

Een formulier maak je door:

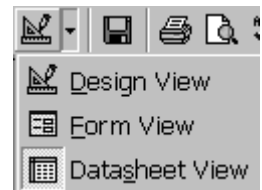
- bij objects te kiezen voor *forms* en dan *Create form by using wizard*. Je krijgt dan onderstaand venster.



- Dan kies je de tabel(len) en velden die op je formulier moeten komen.
- Dan kies je een van de mogelijkheden om de formulier te ontwerpen of zelf te laten makendoor Access.

Een formulier kan op verschillende manieren bekeken worden:

- In de *form view* krijgen we het formulier om in te vullen en
- In de *datasheet view* krijg je een tabelvorm met rijen en kolommen te zien. In deze vorm is het gemakkelijk om hele records te verwijderen.
- In de *design view* kan je het ontwerp van de formulier maken of veranderen door de *labels*, dat zijn de 'titels' en de *tekstvakken*, dat zijn de hokjes die ingevuld moeten worden te verplaatsen, vergroten, verkleinen, enz.



Opdracht:

- Maak formulieren voor de vier tabellen.

Klantnr	Klnaam	Kladres	Klwoonplaats	ordernr	artnr	aanal	korting	leverdatum	artnr	artomschr	prijs
1	Alhala Z	Kaya Buriku 0	Sta Maria	1571	3277	4	20	6/15/91	2301	TV-300	\$3,000.00
2	Mozaro B	Kaya Bonanza 2	Sta Rosa	1571	4781	6	31	7/1/91	3001	CD-1	\$350.00
3	Papaya K	Kaya Bibi 7	Souax	1571	6571	3	25	7/1/91	3002	CD-5	\$450.00
4	Poporopo P	Kaya Popo 3	Popo	1571	6575	1	18	8/2/91	3277	Cassette	\$320.00
5	Kiri P	Kaya A 22	Popo	1581	2301	5	0		4781	Video	\$950.00
6	Kuara X	Kaya B 5	Sta Rosa	1581	3001	6	25	7/7/91	6571	TV-500	\$1,850.00
7	Munula B	Kaya X 99	Sta Rosa	1581	3002	8	0		6572	TV-400	\$2,550.00
23575	Andriessen	Meerstraat 58	Abrahams	1581	4781	12	0		6666	DVD	\$1,000.00
				1591	3277	6	0				
				1591	4781	10	0				
				2200	2301	4	10	6/30/91			
				2200	6570	4	10	6/30/91			
				2200	6572	4	10	6/30/91			
				5000	3001	5	0				
				5000	4781	2	0				
				5000	6571	7	0				

ordernr	datum	vertegenwoordiger	klantnr
1571	5/25/91	Leeuwen L v	23575
1591	5/30/91	Klaas V	23575
2200	6/24/91		4
5000	1/1/00	Zwarte Piet	2
6000			0

- Voer de onderstaande gegevens van de tabellen in via de formulieren.

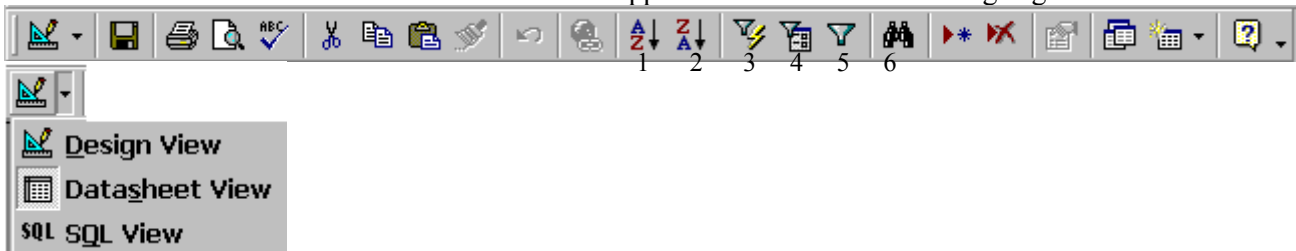
Form View

We kunnen de gegevens bekijken in de form view. We krijgen dan bijvoorbeeld onderstaand scherm:

Met de navigatieknoppen onder kunnen we de verschillende records bekijken door naar de eerste record te lopen, of de vorige , of de volgende , of de laatste . Met het laatste knopje kunnen we een nieuw record invoeren.

Toolbar

De toolbar van de form view bevat verschillende knoppen die hieronder worden uitgelegd.



Door op het driehoekje naast de viewknop te klikken kun je kiezen uit de verschillende views.

De genummerde knoppen hebben de volgende betekenis als je op ze klikt:

1. zet de records op volgorde van klein naar groot
2. zet de records op volgorde van groot naar klein

Filteren

We kunnen met de volgende knoppen gemakkelijk bepaalde records opvragen door een criterium op te geven. We krijgen dan allen de records die aan dat criterium of criteria voldoen.

3. filter by selection. Deze knop toont alleen de records die het gegeven van het veld waarin de cursor staat ook bevat!
4. filter by form. Met deze knop kun je kiezen uit het veld waar de cursor instaat aan welk gegeven de records moeten voldoen die je wilt opzoeken. Door op de tab Or te klikken kun je meerdere criteria opgeven. Daarna klik je op
5. apply filter om de filter uit te voeren
Deze knop verandert dan in remove filter om de filtering weer weg te halen
6. find. Met deze knop krijg je een window waarmee je kunt gaan zoeken naar een bepaalde record.

Zet je de cursor in het veld artomschr en druk op de find knop. Vul dan in TV*. en klik op find next.

De records waarvan de omschrijving met TV begint worden een voor een getoond als je find next blijft klikken.

Opdracht:

- Selecteer de records van alle klanten met woonplaats Sta Rosa met de *filter by form* knop. Merk op dat onderin het volgende komt te staan: Record: 3 of 3 (Filtered)
- Klik op de knop *Remove Filter*.
- Selecteer de records uit tabel bestelregel met een leverdatum 7/1/91

Query's maken

Inleiding

In dit hoofdstuk gaan we in op het werken met gegevens uit de tabellen die we gemaakt hebben. We maken query's op basis van de tabellen van opgave 6 uit hoofdstuk NORMALISEREN.

Query by Example

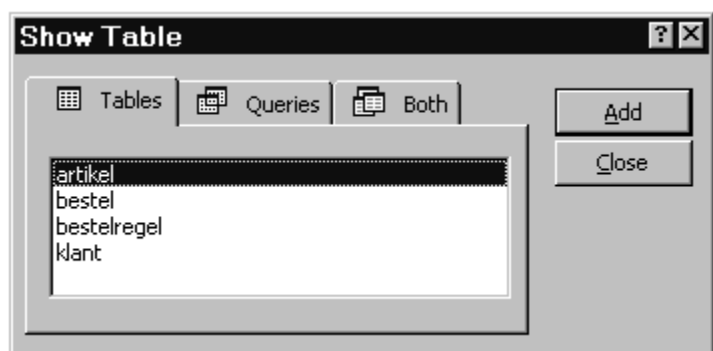
Met een QBE (Query by Example) kunnen bepaalde gegevens of een verzameling gegevens worden gevonden en op het scherm worden getoond, zonder dat je een computertaal hoeft te kennen. We kiezen namelijk gewoon uit de tabellen welke gegevens of berekeningen we willen zien en aan welke voorwaarden (criteria) ze moeten voldoen. We kunnen bepalen hoe gegevens getoond worden door de tabellen te kiezen waaruit de query samengesteld wordt. Bij elkaar kan een query bestaan uit 255 kolommen afkomstig uit 16 verschillende tabellen. Aan de hand van de volgende opdracht zullen we aangeven hoe een QBE gemaakt wordt.

Opdracht:

- Maak een query die de artikelomschrijvingen toont, waarvan er minder dan 10 besteld zijn. Het overzicht moet op alfabetische volgorde staan.

Om een QBE te maken

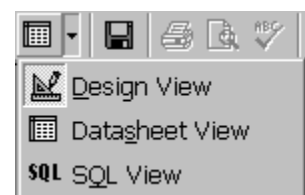
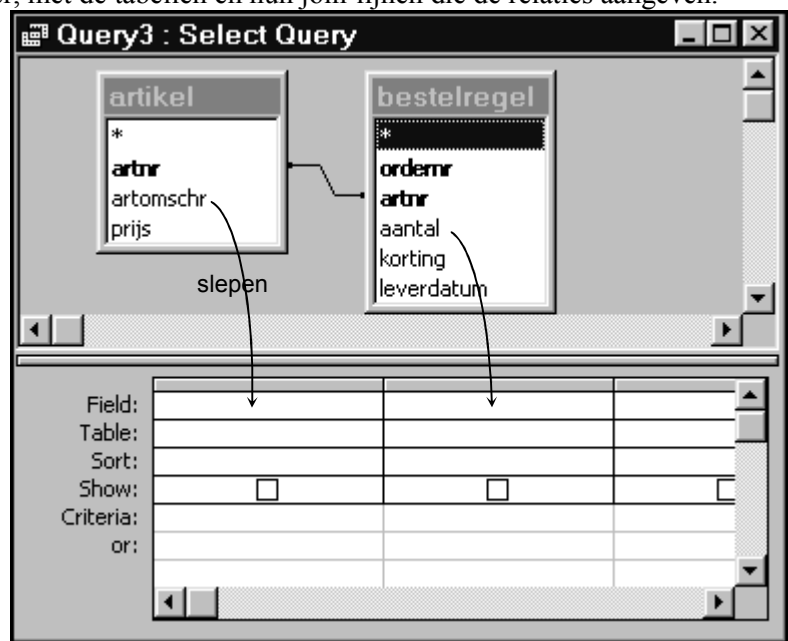
- klikken we op het tabblad queries en dan new en dan Design View. We krijgen dan het venster zoals hiernaast.
- We kiezen de tabel(len) die we nodig hebben voor onze query door ze aan te klikken en dan op de knop Add te klikken. In onze opdracht gaat het om de tabellen artikel en bestelregel.
- Daarna sluiten we dit venster door op de knop Close te klikken.



We krijgen dan een venster zoals hieronder, met de tabellen en hun join-lijnen die de relaties aangeven.

Je gaat dan verder door

- de velden uit de tabellen te slepen naar de rij Field in de kolommen op de *queryraster*
- Automatisch wordt in de rij Table ingevuld uit welke tabel dat veld komt.
- Je klikt in de rij Sort, dan op het pijltje ▼ en kiest dan voor Ascending
- Zorg dat je in de rij Show het vinkje ☒ in kolom aantal weer wegklikt, want je wilt het aantal niet zien, maar de artikelomschrijvingen.
- In de rij criteria tik je dan in kolom aantal; < 10
- Door op de *Datasheet view* te klikken zien we het resultaat van de query. (We kunnen ook kiezen om de achterliggende SQL-opdrachten te bekijken door op *SQL View* te klikken.)



Toolbar



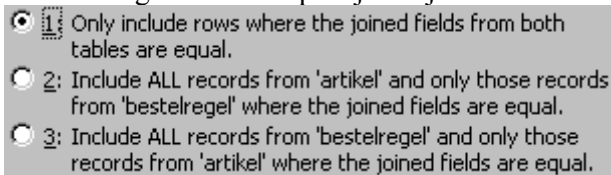
Belangrijke knoppen in de toolbar van queries zijn van links naar rechts:

- De view knop om de queries op verschillende manieren te bekijken: design, datasheet en sql view
- Met de volgende knop kun je verschillende soorten queries kiezen.
- daarnaast is de run-knop, waarmee een query uitgevoerd kan worden
- dan de show table knop, waarmee je de tabellen krijgt die je eventueel kunt toevoegen aan je query
- de group by knop
- en de knop om je database window te zien. (Je openingsscherm)

Joints

Zijn de join-lijnen er niet, dan moeten we de tabellen koppelen, door dezelfde velden in de verschillende tabellen met elkaar te verbinden. Gebeurt dit niet dan worden alle records uit de tabellen aan elkaar gekoppeld zonder koppelingsvoorwaarde!!!

Door de 'right-clicken' op de join-lijn kunnen we kiezen uit 3 typen joins.



- je kiest in de kolom uit de keuzevakjes die je krijgt door op de kolom te klikken of
- je tikt een veldnaam of een berekening in, bv prijs*aantal.

Criteria

In de rij *Criteria* kun je aangeven of je die gegevens wilt selecteren door bepaalde voorwaarden (criteria) op te geven voor de betreffende velden. Dit kunnen enkelvoudige voorwaarden zijn, verschillende voorwaarden gecombineerd door "OR" of AND" en "parameterqueries".

Inhet volgende overzicht worden veelgebruikte criteria genoemd die we kunnen opgeven.

- De waarde in een veld is gelijk aan de waarde die we opgeven. Voer gewoon de waarde in de criteriacycel van het veld in. Bijvoorbeeld: om alle cursisten die in Utrecht wonen te vinden, typen we Utrecht in de criteriacycel van het veld Plaats. Bevat de tekst waarde een spatie of een komma, zet de waarde dan tussen aanhalingstekens.
- De waarde van een veld valt in een reeks van waarden. We voeren een zin of een uitdrukking in die de reeks die we op het oog hebben, definieert. Bijvoorbeeld, om alle cursisten vanaf de letter 'C' te tonen, voeren we >C in de criteriacycel van het veld Achternaam in. Willen we de cursistenummers tussen 1 en 40 zien, dan typen we in de criteriacycel van het veld Cursistnr **BETWEEN 1 AND 40** of **>1 AND <40**
- De waarde in een veld begint met de waarde die we opgeven. Gebruik de asterisk (*) als joker. Bijvoorbeeld: om alle achternamen in de tabel Cursisten te zien die met een 'V' beginnen, voeren we V* in de criteriacycel van het veld Achternaam in. Tekstvelden in Access zijn niet'case sensitive', dat wil zeggen dat het niet uitmaakt of de waarde begint met een hoofdletter of een kleine letter. OF
- Als je iemand zoekt, waarvan je de naam niet precies kent, maar je weet wel dat het met een K begint, dan kun je ook de instructie **LIKE** gebruiken. Je vult dan in de criteriacycel van Klnaam in LIKE K* LIKE *PA* toont iedereen die een PA in zijn naam heeft.
- De waarde in een veld is er één uit een lijst van waarden. Gebruik de functie In(). Om bijvoorbeeld de records van alle klanten uit Sta Rosa en Popo te laten tonen, typen we in de criteriacycel van het veld Klwoonplaats **IN(Sta Rosa, Popo)** of we typen Sta Rosa en daaronder in de rij OR Popo.

Parameterqueries

Een parameterquery is een query waarbij gevraagd wordt naar een bepaalde waarde van een veld. De vraag typ je tussen rechte haken in in het criteriaveld.

Bij uitvoer wordt dan een window getoond die je moet invullen, waarna de records

getoond worden die voldoen aan de waarde die je hebt ingevuld, bv Sta Rosa.



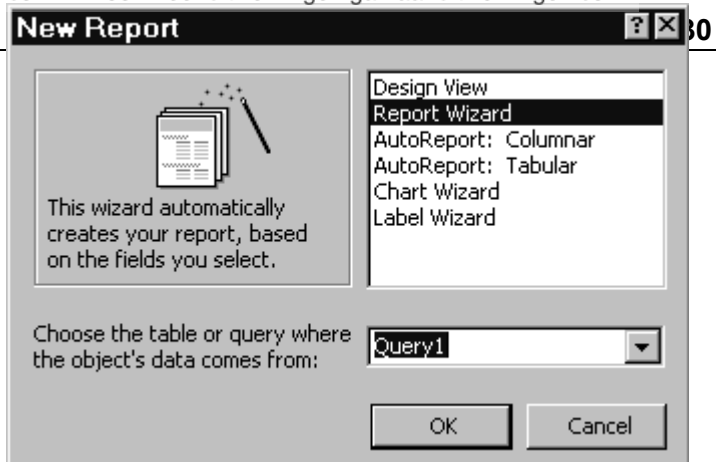
Opdrachten:

1. Maak een query die de artikelomschrijvingen toont van die artikelen waarvan er minder dan 5 of meer dan 10 zijn besteld.
2. Maak een query die de artikelomschrijvingen toont van die artikelen waarvan er minder dan 10 en meer dan 5 zijn besteld. (<10 AND >5)
3. Maak een query die artomschrijving, aantal, korting en de waarde van de artikelen toont.
typ in de laatste kolom waarde : prijs * aantal*(100-korting)/100
4. Maak een query die naam en adres afdruckt van alle klanten die een video hebben besteld.
5. Maak een query die naam en adres afdruckt van alle klanten die niet in Sta Rosa wonen.
6. Maak een query die naam en adres afdruckt van alle klanten die een video of een CD-1 hebben besteld.
7. Maak een query die naam en adres afdruckt van alle klanten die een tv hebben besteld. (Je moet LIKE gebruiken, omdat er verschillende soorten tv's zijn))
8. Maak een parameterquery die naam en adres afdruckt van alle klanten als je de woonplaats invoert.
Laat de namen op alfabetische volgorde getoond worden.
Voer de query uit en vul in Sta Rosa.

Access en SQL**Rapporten maken**

Om een rapport te maken

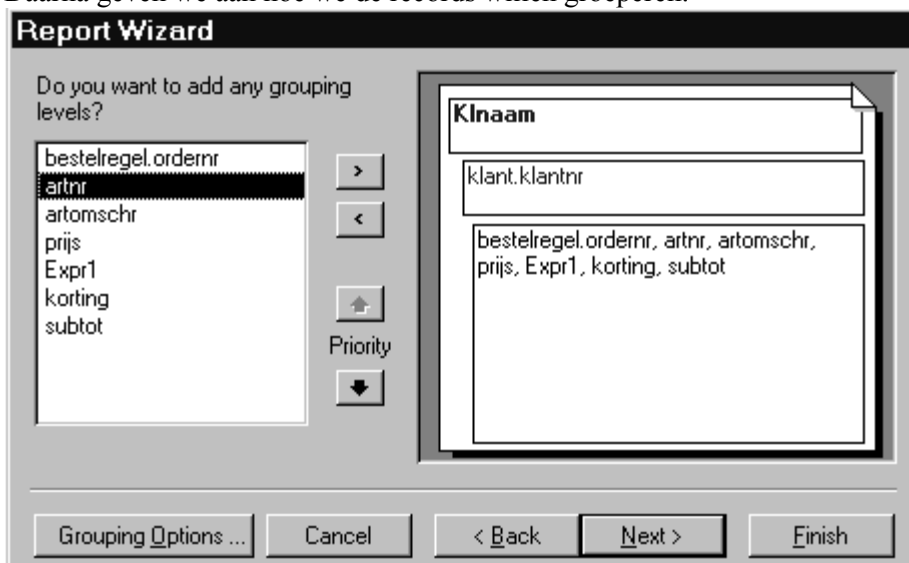
- klikken we op het tabblad *reports* en dan *new* en dan *Report Wizard*. We kiezen de tabel of query, waarvan we een rapport willen maken en klikken op *OK*.



- We krijgen dan een venster zoals hieronder, met de velden en kiezen de velden die we in ons rapport willen hebben, door op ze te dubbelklikken of op ze te klikken en met de pijlen naar het rechtervak te sturen of terug te zetten.



- Daarna geven we aan hoe we de records willen groeperen.



- En dan stellen we eventueel de sorteervolgorde vast. Als er gegroepeerd wordt en er zijn numerieke velden, dan krijgen we onder de *sort options* ook een mogelijkheid om totalen, gemiddelden ed. te laten berekenen met *Summary Options*...

What summary values would you like calculated?

Field	Sum	Avg	Min	Max
prijs	☐	☐	☐	☐
Expr1	☐	☐	☐	☐
korting	☐	☐	☐	☐
subtot	☒	☐	☐	☐

OK
Cancel

Show
☒ Detail and Summary
☐ Summary Only

☐ Calculate percent of total for sums

- We kunnen het rapport voorzien van een opmaakstijl en klikken dan op finish.
- We krijgen dan een preview van het rapport te zien, die we kunnen afdrukken of bewaren, of
- als we niet tevreden zijn met de layout, dan gaan we terug naar de *design view* en veranderen de layout, net zoals we bij de formulieren hebben gedaan.

opdracht:

Maak het rapport zoals hieronder aangegeven.

Klnaam		Andriessen			
klant.klantnr		23575			
artnr	artomschr	prijs	aantal	korting	subtot
6572	TV-400	\$2,550.00	1	18	\$2,091.00
6571	TV-500	\$1,850.00	3	25	\$4,162.50
4781	Video	\$950.00	6	31	\$3,933.00
3277	Cassette	\$320.00	4	20	\$1,024.00
Summary for 'klant.klantnr' = 23575 (4 detail records)					
Sum					\$11,210.50
Summary for 'Klnaam' = Andriessen (4 detail records)					
Sum					\$11,210.50
Grand					\$11,210.50

STRUCTURED QUERY LANGUAGE (SQL)

SQL is een vierde generatie programmeertaal, een vraagtaal waarmee je in staat bent om een *database* en zijn *tabellen* te creëren, *gegevens* in de tabellen op te slaan en de tabellen te wijzigen, en *informatie* te halen uit de tabellen. We zullen in dit stencil behandelen hoe je informatie kunt halen uit tabellen, die al *genormaliseerd* zijn.

SELECT FROM

De eerste twee opdrachten waar iedere zoekopdracht in SQL mee begint zijn SELECT en FROM. Hiermee wordt aangegeven:

- wat je zoekt of selecteert uit een of meer tabellen en
- uit welke tabel(len) je selecteert.

Voorbeelden:

```
SELECT      NAAM, ADRES
FROM WERKNEMER;
```

```
SELECT      SALARIS +500
FROM WERKNEMER;
```

```
SELECT      SUM(SALARIS)
FROM WERKNEMER;
```

```
SELECT      *
FROM        WERKNEMER;
```

Na SELECT kan komen

- een of meer *kolomnamen*,
- een *berekening* of
- een *statistische functie*.

NB: Het is meestal verplicht een puntkomma (;) te plaatsen aan het eind van een SQL-opdracht!

VOORWAARDEN AAN IEDERE REGEL met WHERE

Zoek je naar gegevens die aan een of meer voorwaarden moeten voldoen, dan geef je de voorwaarden in de derde regel aan achter de instructie WHERE.

Voorbeelden:

```
SELECT      NAAM
FROM        WERKNEMER
WHERE       WOONPLAATS = "WILLEMSTAD";
```

```
SELECT      NAAM, SALARIS
FROM WERKNEMER;
WHERE       SALARIS > 4500;
```

AND en OR

Je kunt ook een zoekopdracht hebben waarbij aan meerdere voorwaarden voldaan moet worden. Hiervoor kun je de instructies AND of en OR gebruiken.

Voorbeelden:

```
SELECT      NAAM
FROM        WERKNEMER
WHERE       WOONPLAATS = "WILLEMSTAD" OR WOONPLAATS = "STA MARIA";
```

```
SELECT      NAAM
FROM WERKNEMER
WHERE      WOONPLAATS = "WILLEMSTAD" AND SALARIS >4000 AND SALARIS < 5000;
```

ORDER BY

Om de gevens die je zoekt op volgorde afgedrukt te krijgen (te sorteren) of alfabetisch in groepen bij elkaar, kun je in een SQL-opdracht de instructie ORDER BY opnemen.

Voorbeelden:

```
SELECT      NAAM
FROM        WERKNEMER
ORDER BY    NAAM;
```

```
SELECT      NAAM, WOONPLAATS
FROM WERKNEMER
ORDER BY    WOONPLAATS, NAAM;
```

Deze opdracht drukt een lijst af van namen en woonplaatsen van de werknemers, gesorteerd op alfabetische volgorde van WOONPLAATS, en binnen elke WOONPLAATS de namen op alfabetische volgorde.

```
SELECT      NAAM
FROM WERKNEMER
ORDER BY    SALARIS DESC;
```

Deze opdracht sorteert de namen op SALARIS maar van groot naar klein (descending)

STATISTISCHE FUNCTIES

Tot nu toe heb je 'queries' gemaakt waarin afzonderlijke rijen geselecteerd werden en waarin soms bepaalde waarden in afzonderlijke rijen bewerkt werden. Hier werd een tabel dus rij voor rij doorzocht. Soms wil je echter bepaalde groepen van waarden in een aantal rijen samen nemen en daar een bewerking op toepassen. Je zoekt bijvoorbeeld niet naar de som van alle salarissen, maar naar de som van alle salarissen uit elke woonplaats.

Met de volgende zogeheten statische functies kun je bewerkingen uitvoeren op groepen van waarden.

COUNT ()	bepaalt het aantal rijen of waarden uit een bepaalde kolom
MIN ()	bepaalt de kleinste waarde uit een bepaalde kolom
MAX ()	bepaalt de grootste waarde uit een bepaalde kolom
SUM ()	bepaalt het totaal van de waarden uit een bepaalde kolom
AVG ()	bepaalt het rekenkundig gemiddelde van de waarden uit een bepaalde kolom

DISTINCT, GROUP BY

DISTINCT gebruik je als je wilt weten in *welke verschillende* plaatsen de werknemers wonen:

```
SELECT      DISTINCT PLAATS
FROM        WERKNEMER;
```

Bij sommige vragen levert COUNT alleen niet het gewenste resultaat. Als je bijvoorbeeld wilt weten in *hoeveel verschillende* plaatsen de werknemers wonen, dan kun je niet volstaan met:

```
SELECT      COUNT (PLAATS)
FROM        WERKNEMER;
```

De opdracht Count telt dan alle rijen, ongeacht of bepaalde waarden twee of meer keer voorkomen.

DISTINCT voorkomt dat er dubbele rijen in het resultaat voorkomen. De opdracht wordt dan:

```
SELECT      COUNT (DISTINCT PLAATS) FROM WERKNEMER; [werkt niet in Access]
SELECT      COUNT PLAATS FROM (SELECT DISTINCT PLAATS FROM WERKNEMER);
```

Of met group by werken, dan krijg je de verschillende plaatsen en hoe vaak ze voorkomen.

```
SELECT      COUNT (PLAATS) FROM WERKNEMER
GROUP BY    PLAATS;
```

Wil je in een zoekopdracht een statistische bewerking toepassen op *groepen* van gegevens, dan moet je in de opdracht aangeven dat er gegroepeerd moet worden en per groep de functie toepassen. Dit geschiedt met de instructie GROUP BY.

Voorbeelden:

```
SELECT      WOONPLAATS, COUNT (WOONPLAATS)
FROM        WERKNEMER
GROUP BY    WOONPLAATS;
```

Deze opdracht drukt iedere woonplaats één keer af en daarachter het aantal keren dat iedere woonplaats voorkomt (Turfen).

```
SELECT      WOONPLAATS, SUM (SALARIS)
FROM        WERKNEMER
GROUP BY    WOONPLAATS;
```

Deze opdracht drukt per woonplaats af wat de som van de salarissen der werknemers zijn.

```
SELECT      WOONPLAATS, LAND, SUM (SALARIS)
FROM        WERKNEMER
GROUP BY    WOONPLAATS, LAND; (NB. Let op dat je zowel per woonplaats en land moet groeperen!)
```

Als je in SELECT een kolom en een statistische functie gebruikt, dan moet je GROUP BY gebruiken.

Voorbeeld:

goed

```
SELECT      WOONPLAATS, SUM (SALARIS)
FROM        WERKNEMER
GROUP BY    WOONPLAATS;
```

fout

```
SELECT WOONPLAATS, SUM (SALARIS)
FROM WERKNEMER;
```

VOORWAARDEN AAN GROEPEN (VWO)

Wanneer je een voorwaarde stelt aan de groepen die je maakt, dan gebruik je HAVING.

Voorbeeld: Je wilt een overzicht van woonplaatsen met de som der salarissen van de werknemers, waarbij deze som meer moet zijn dan 10000 gulden.

```
SELECT      WOONPLAATS, SUM (SALARIS)
FROM        WERKNEMER
GROUP BY    WOONPLAATS
HAVING ;    SUM (SALARIS) > 10000
```

HAVING mag je alleen na GROUP BY gebruiken en na HAVING moet een voorwaarde met een statistische functie komen!!!

Deze opdracht werkt niet met WHERE, want WHERE bepaalt de voorwaarde voor de rij afzonderlijk en geen groepen rijen!!!

Subqueries (VWO)

Een subquery is een query in een query. Dit gebruik je als de voorwaarde die je stelt zelf het resultaat is van een query. Bijvoorbeeld je wilt de namen en salarissen hebben van die werknemers

- ♦ die een salaris hebben die groter is dan het gemiddelde salaris. OF
- ♦ die een salaris hebben die gelijk is aan het maximumsalaris OF
- ♦ die een salaris hebben die meer is dan 20% van de som van de salarissen

```
SELECT      naam, salaris
FROM        WERKNEMER
♦ WHERE     salaris > (SELECT AVG(salaris) FROM WERKNEMER);
♦ WHERE     salaris = (SELECT MAX(salaris) FROM WERKNEMER);
♦ WHERE     salaris > 0.2* (SELECT SUM(salaris) FROM WERKNEMER);
```

NB Na WHERE mag geen statistische functie komen. Dus FOUT is WHERE salaris = MAX(salaris)

MEERDERE TABELLEN KOPPELEN

Heel vaak bestaat een database uit meerdere tabellen en zoeken we naar gegevens uit verschillende tabellen of naar gegevens uit één tabel, maar die voldoen aan voorwaarden waarvoor een andere tabel geraadpleegd moet worden. We moeten dan altijd de *koppelingsvoorwaarde* aangeven, anders wordt elke rij uit de ene tabel gecombineerd met iedere rij uit de andere tabel.

Gegeven zijn de volgende twee tabellen WERKNEMER en AFDELING:

WERKNEMER			
NR	NAAM	AFDCODE	FUNKTIE
1	Mees	101	directeur
2	Karap	103	verkoper
3	Zali	101	adjunct

AFDELING	
AFDCODE	NAAM
101	Directie
103	Design

De SQL-opdracht

```
SELECT *
FROM WERKNEMER, AFDELING;
```

is een koppeling van de tabellen zonder voorwaarde en geeft de volgende tabel:

NR	NAAM	AFDCODE	FUNKTIE	AFDCODE	NAAM
1	Mees	101	directeur	101	Directie
1	Mees	101	directeur	103	Design
2	Karap	103	verkoper	101	Directie
2	Karap	103	verkoper	103	Design
3	Zali	101	adjunct	101	Directie
3	Zali	101	adjunct	103	Design

De juiste SQL-opdracht zou moeten zijn:

```
SELECT *
FROM WERKNEMER, AFDELING
WHERE WERKNEMER.AFDCODE = AFDELING.AFDCODE;
```

Hierbij staat achter WHERE de voorwaarde dat AFDCODE uit tabel WERKNEMER hetzelfde moet zijn als de AFDCODE uit tabel AFDELING.

En dit levert de volgende tabel op:

NR	NAAM	AFDCODE	FUNKTIE	AFDCODE	NAAM
1	Mees	101	directeur	101	Directie
0	Karap	103	verkoper	103	Design
3	Zali	101	adjunct	101	Directie

→ **NB.** Als een veld in meerdere tabellen staat dan moet je steeds voor het veld de tabelnaam vermelden waarin dat veld zit en niet alleen het veld noemen.

BV

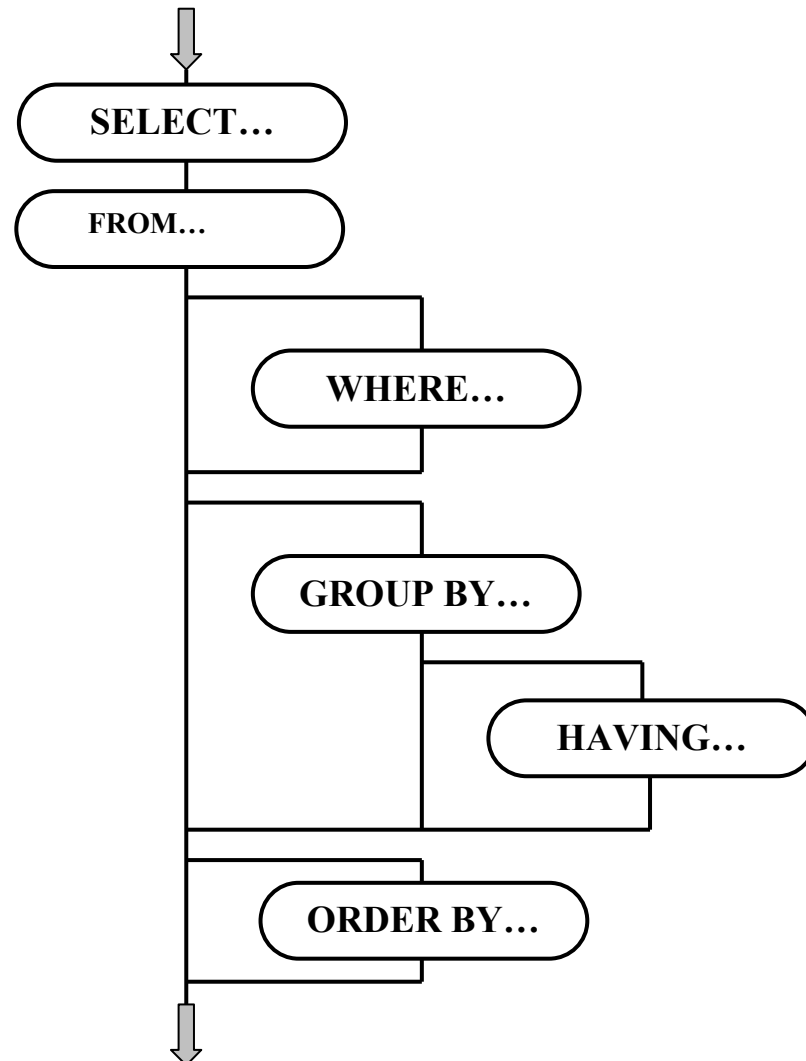
```
SELECT NR, WERKNEMER.NAAM, WERKNEMER.AFDCODE, AFDELING.NAAM
FROM WERKNEMER, AFDELING
WHERE WERKNEMER.AFDCODE = AFDELING.AFDCODE;
```

NR	NAAM	AFDCODE	NAAM
1	Mees	101	Directie
0	Karap	103	Design
3	Zali	101	Directie

SAMENVATTING

Om database-gegevens te **raadplegen** gebruiken we het **SELECT** commando. Dit commando werkt op de tabel(len) en levert een nieuwe tabel op.

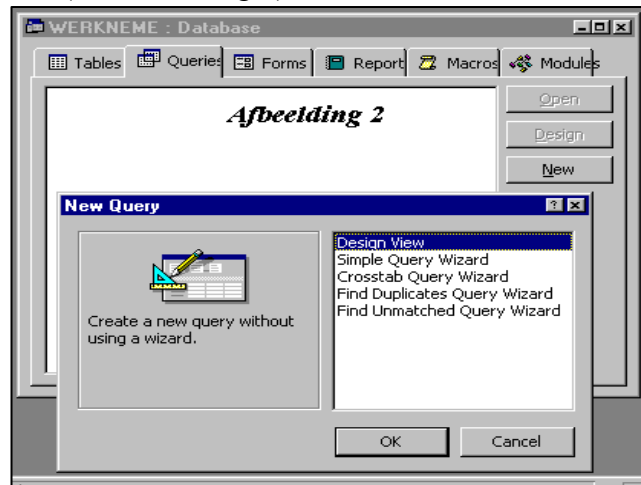
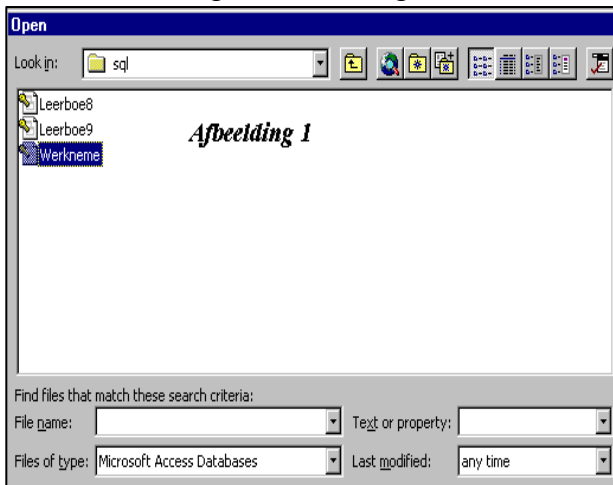
Het **SELECT** commando heeft zes onderdelen, waarmee in principe alle SQL bewerkingen om tabellen te raadplegen gedaan kunnen worden. Een schema van deze onderdelen ziet er als volgt uit:



- ♦ De volgorde van de onderdelen liggen hierbij vast
- ♦ SELECT en FROM zijn verplicht
- ♦ WHERE, GROUP BY en ORDER BY zijn optioneel
- ♦ HAVING komt niet voor zonder GROUP BY (alleen voor VWO, niet voor havo!!!)

SELECT	welke kolom(men) wil ik hebben in mijn resultaat tabel
FROM	welke tabel(len) moeten worden geraadpleegd
WHERE	waaraan moeten de rijen voldoen
GROUP BY	waarop moet worden gegroepeerd
HAVING	waaraan moeten de groepen rijen voldoen
ORDER BY	in welke volgorde moet het resultaat staan.

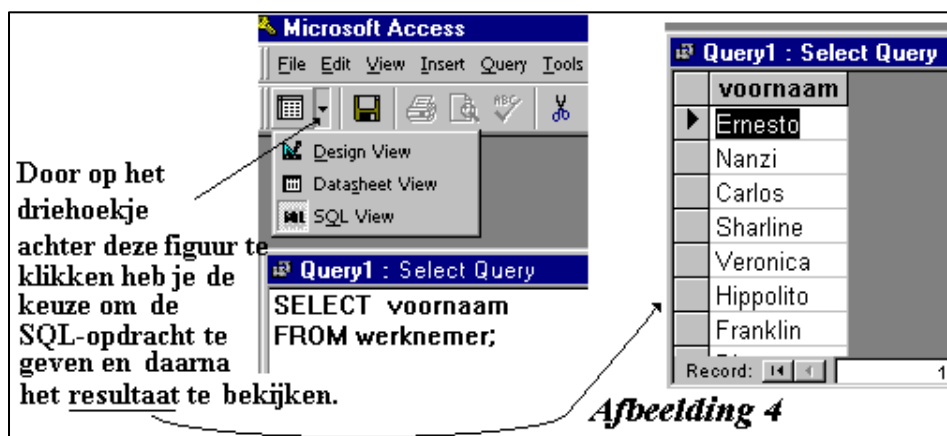
1. Start Microsoft Access
2. Via het menu File en dan Open kies je het gewenste Database-bestand. (Zie afbeelding 1). Als het bestand geopend is krijg je een window te zien zoals het hieronder is weergegeven in afbeelding 2.
3. In afbeelding 2 zie je een window met verschillende tabbladen. Kies voor het tabblad Queries. Kies vervolgens voor design view en klik op OK. (zie afbeelding 2).



4. Vervolgens krijg je een window te zien waarin in je de tabellen moet aangeven waarmee je wilt werken. (Show Table) Zie afbeelding 3. Klik op de gewenste tabel en klik vervolgens op Add. Hiermee wordt de tabel op het bewerkingsblad gezet. Doe dit voor alle tabellen waarop je bewerkingen wilt uitvoeren.



5. De volgende stap is om de window Show Table te sluiten als alle benodigde tabellen op het bewerkingsblad staan. (Close)
6. Vervolgens kun je linksboven d.m.v. de figuur getoond in afbeelding 4, kiezen voor de SQL View om de SQL-opdrachten in te toetsen. Daarna kan je met de keuze Datasheet View via dezelfde figuur het resultaat bekijken.



SUCCES met de SQL opdrachten.

opdrachten:

Bij opgave 8 uit hoofdstuk 10 van het leerboek:

1. Maak een overzicht met alle gegevens van de CDs
2. Maak een overzicht van alle muzikanten uit Puerto Rico
3. Van welke titels zijn er minder dan 5 besteld? (distinct)
4. Wat is het hoogste, het laagste en het gemiddelde aantal van ordernummer 24?
5. Geef een overzicht van leveranciersnamen, waar op 31 januari 1994 een bestelling is geplaatst.
6. Maak een overzicht van de ordernummers met hun totale aantal.
7. Maak een overzicht van alle leveranciersnamen, waar CD's van de muzikant Percy Sledge zijn besteld.

Bij opgave 9 uit hoofdstuk 10 van het leerboek:

8. Maak een overzicht van alle gegevens van de stoffen met de betalen prijs per stof uit bestelnr 64
9. Wat is de totaal te betalen prijs van bestelnr 64?
 - a. Maak een overzicht van de totaal te betalen prijzen per bestelnr.
 - b. Maak een overzicht van de totaal te betalen prijzen per bestelnr, waarbij dit totaal >1100 (HAVING)
 - c. Maak een lijst van leveranciersnamen, met bestelnr en datum, die in een bestelling meer dan 2 stoffen hebben besteld.
10. Hoeveel verschillende kleuren staan er op bestelnr 64?
11. Bereken de totaalprijs die leverancier Keek & Stoppenburg geleverd heeft.
12. Maak een overzicht van alle gegevens van de duurste stof.
 - a. Maak een overzicht van de gegevens van de stoffen die meer dan de helft van de duurste stof kosten.
13. Geef de bestelnr, stofnr en omschrijving van de stof, waarvoor het meest betaald is in alle bestellingen.
14. Geef stofnr en omschrijving van de stoffen die niet in een bestelling voorkomen.

levnr	naam	adres	plaats	tel
180	Goedkoop	Brabaweg 22	5678 AB Amsterdam	010-112233
184	Keek&Stoppenburg	Domburgseweg 33	3489 GH Hilversum	01790-223344

bestelnr	datum	levnr	inkoper	inkoper	inknaam
64	6/7/91	184	ZA	AZ	A. Zergoed
75	5/8/91	18	AZ	ZA	Z. Aleman
82	8/8/91	184	AZ	KL	K. Leren

bestelnr	stofnr	meter
64	12	50
64	20	20
64	24	10
75	12	40
75	20	50
82	24	30
82	12	25
82	20	40

stofnr	omschrijv	breedte	kleur	prijs
12	acryl 50/13	1.4	paars	\$8.50
20	katoen 413	1.64	rood	\$15.50
24	viscose k3m	1.24	wit	\$20.00
30	katoen 512	1.4	paars	\$12.00