DATA-BASES: EEN INTRODUCTIE

door Drs. R. A. M. Pruijm R. A.

„I could have done it in a much more complicated way said the Red Queen immensely proud"
uit: Through the looking glass.

1 Inleiding

De ontwikkeling en toepassing van databases zal voor de rest van dit decennium ongetwijfeld een van de belangrijkste automatiseringsaktiviteiten zijn.

In toenemende mate wordt informatie beschouwd als een vitaal ondernemingsaktief dat op een zodanige wijze moet worden beheerd dat er een optimaal gebruik van kan worden gemaakt.

Dit is de verklaring voor de opkomst van databases.

In dit artikel zal worden geprobeerd de niet-technische lezer een beeld te schetsen van het fenomeen database.

In een volgend artikel „Databases en de accountant" zal worden ingegaan op de implicaties welke het gebruik van databases heeft voor de niet-gespecialiseerde accountant, belast met algemene controle-opdrachten.

Voor een groot deel is het materiaal voor beide artikelen opgedaan gedurende een studiereis van 3 maanden door de Verenigde Staten. Deze reis werd gesubsidieerd door het Instituut voor Zuiver-Wetenschappelijk Onderzoek (ZWO) en had tot doel de opleiding aldaar te bestuderen van accountants in het controleren van geautomatiseerde gegevensverwerkende processen (Auditing & EDP).

2 Wat is een database?

Velen Uwer zullen ongetwijfeld wel een vaag begrip hebben van een „data base" of gegevensbank.

Het beeld dat dan oprijst is dat van een grote centrale gegevensopslag waaruit velerlei gebruikers hun gegevens putten.

Dit kan geschieden door gebruikers van zeer diverse aard, variërend van programmeurs - die applicatieprogramma's schrijven die de database benutten - tot administratieve medewerkers op een afdeling die d.m.v. een eenvoudige taal gegevens opvragen uit de database, of deze muteren.

De verwerkingsprocessen zijn eveneens zeer verschillend, variërend van groepsgewijze verwerkende processen tot en met postgewijze invoer.

Kenmerkend is dat de data base geïntegreerd is, dat betekent dat de database gegevens bevat voor vele gebruikers en niet slechts voor één.

Dit impliceert dat elke gebruiker slechts te maken heeft met een klein deel van de aanwezige gegevens maar ook dat vele gebruikers behoefte kunnen hebben aan dezelfde gegevens ofwel dat bepaalde gegevens door meerdere gebruikers gedeeld worden.

m a b blz. 438

Een database is een groot reservoir gegevens waaruit velerlei gebruikers naar behoefte zonder problemen hun gegevens uit kunnen putten.

Een dergelijke database zou zeer complex zijn en we kunnen in zijn algemeenheid wel stellen dat deze nog geen werkelijkheid is geworden, hoogstens is het een doelstelling op lange termijn.

3 Ontstaansmotieven voor een database

Men kan stellen dat databases vooral zijn ontstaan als reactie op de beperkingen van de conventionele bestandsvormen, die zich vooral bij de grotere en meer geavanceerde computergebruikers voordeden.

De kenmerken van de conventionele bestandsvormen zijn:

1 de beperkte toegankelijkheid
2 redundantie van gegevensopslag
3 programma afhankelijkheid.

ad 1

Elk bestand is aangemaakt voor een bepaald doel, hetgeen impliceert dat niet alleen de hierin opgeslagen gegevens maar ook de volgorde ervan zodanig opgezet is dat de beoogde verwerking zo efficiënt mogelijk uitgevoerd kan worden.

Een bestand dat bijvoorbeeld de verkoopgegevens bevat zal voor een inzicht in de omzetten per artikelgroep, in volgorde van artikelgroep, aan het hiervoor bestemde programma A worden aangeboden. Voor programma B dat per vertegenwoordiger een overzicht van de verkopen wil leveren zal het noodzakelijk zijn dat bestand te ordenen in volgorde van vertegenwoordiger.

Uit bovenstaande valt af te leiden dat de ordening van een bestand beperkingen kan inhouden voor de toegankelijkheid van meerdere programma's.

Dit laatste wordt vooral van belang in een omgeving met veel dialoog (ofwel real-time) toepassingen, waarbij meerdere programma's veelal tegelijkertijd ieder naar behoefte eenzelfde bestand willen benaderen.

Wil men aan deze wensen voldoen dan zullen er meerdere bestanden moeten worden aangelegd, met dezelfde inhoud maar met een verschillende ordening. Hiermede wordt dan eveneens gegevensredundantie ingebracht.

Hiernaast wordt dan veelal de inhoud van de bestanden zelf aangepast aan de specifieke behoeften van de verwerkingsprogramma's. Het reeds eerder gememoreerde programma B zal alleen aan gegevens nodig hebben: de naam van de vertegenwoordiger, het rayon en de behaalde omzet; voor programma A zullen meer gegevens nodig zijn, artikelnummers, artikelcategorie, brutowinstpercentage etc. Het gevolg hiervan is dat er ook verschillen gaan optreden in de indeling van de gegevens op de bestanden, de record lay-out. De beperkte toegankelijkheid van de conventionele bestanden heeft resumerende tot gevolg het aanhouden van meerdere bestanden, elk qua ordening en indeling aangepast aan de specifieke behoeften van het erbij behorende verwerkingsprogramma.

ad 2

Redundantie is in deze context te omschrijven als opslag van méér gegevens

dan feitelijk benodigd voor de verwerking. De oorzaken hiertoe zijn reeds in het punt ad 1. besproken en resulteren in duplicatie van bestanden, al of niet met veranderingen in de recordindeling.

Redundantie wordt in de accountantsliteratuur vaak besproken in de zin van nuttige overmaat; voorbeelden hiervan zijn de pariteitsbit en het checkdigit. In deze betekenis is het een algemeen middel om te bereiken dat juiste verwerking van informatie wordt gesignaleerd. In de vorm zoals hier besproken levert het echter vele nadelen op.

Allereerst veroorzaakt het extra kosten door het aanhouden van vele kopieën van bestanden.

Ten tweede zijn er veelvuldige bewerkingen nodig om al deze extra bestanden up to date te houden. Dit klemt te meer voor bestanden met een zeer hoge mutatiegraad.

Ten derde zullen de bestandskopieën in verschillende stadia van bijwerken zijn, zodat er inconsistente informatie kan worden verschaft. Dit wordt dan veroorzaakt door de onmogelijkheid de verschillende kopieën synchroon bij te werken.

Vooral uit hoofde van een goede interne controle zal dit laatste punt veelal niet aanvaardbaar zijn.

ad 3

Wil een programma een bestand verwerken, dan zal dit programma, of beter gezegd, de programmeur, de juiste inhoud van dat bestand moeten kennen. Hij moet weten welke velden er in het record voorkomen, de lengte ervan, de hoedanigheid (numeriek, alfanumeriek etc.) alsmede de juiste positie van dat veld in het record.

Eveneens zal hij op de hoogte moeten zijn van de volgorde waarin het bestand is gesorteerd, dit laatste indien het gaat om sequentiële bestanden. Hanteert men complexere bestandsstructuren zoals bijv. indexed-sequential, dan zal hij moeten weten welk veld als index wordt gebruikt.

Met behulp van al deze gegevens wordt het programma als het ware gebouwd op basis van de in het bestand neergelegde opslagstructuur (de recordindeling, veldtype etc.) en de benaderingswijze (sequentieel e.d.).

Dit geldt voor elk programma dat het betrokken bestand benadert, hetgeen impliceert dat, indien er iets in de opslagstructuur of benaderingswijze verandert, deze programma's alle aan de veranderde omstandigheden moeten worden aangepast.

Deze nauwe verwantschap tussen enerzijds de opslagstructuur en benaderwijze en anderzijds het programma noemt men programma afhankelijkheid ofwel de „data dependency" van een programma.

Aangezien het in onze dynamische maatschappij moeilijk is vast te stellen wat in de tijd de behoeften zullen zijn aan informatie, is het aanpassen van de bestanden en daarmede ook van de programma's hieraan een veelvuldig voorkomende en daarmede kostbare zaak.

Eerlijkheidshalve moet ik vermelden dat de conventionele bestandsvormen met het voorgaande in een veel te zwart daglicht zijn gesteld, dit is echter gebeurd om de factoren welke tot het ontstaan van databases hebben geleid,

goed zichtbaar te maken. Voor talloze applicatievormen zijn en blijven de conventionele bestandsorganisaties nog uitermate efficiënt en in vele gevallen doelmatiger dan de toepassing van een database.

4 De doelstellingen van een database

De doelstellingen van de ontwerpers van databases waren vooral gericht op het oplossen van de reeds eerder genoemde nadelen, gepaard gaande aan het gebruik van de conventionele bestandsvormen.

Een goede weerspiegeling hiervan is te vinden bij Prof. Gordon Everest die de volgende doelstellingen van databases noemt:

1 „shared data", dezelfde gegevens moeten door meerdere gebruikers kunnen worden benut.
2 „availability of data", de gegevens dienen voor allerlei categorieën gebruikers beschikbaar te zijn onder toepassing van de door hen gewenste verwerkingswijze en taal.
3 „evolvability", de verkrijgbaarheid van gegevens dient ook in de toekomst gehanhaafd te kunnen blijven zodat de database gemakkelijk moet kunnen worden aangepast aan de veranderende technologie en gebruikersbehoefte.
4 „integrity", het voortdurend handhaven van de juistheid en volledigheid van de opgeslagen gegevens alsmede de bescherming tegen ongewenste of ongeautoriseerde kennisname van de opgeslagen gegevens.

Alvorens over te gaan tot een meer gedetailleerde bespreking van bovengenoemde doelstellingen wil ik erop wijzen dat het hier gaat om doelstellingen welke in de praktijk nog maar ten dele zijn gerealiseerd.

ad 1 „Shared data"

Deze doelstelling houdt in dat meerdere gebruikers toegang moeten kunnen hebben tot hetzelfde gegeven opdat doublures in de gegevensopslag vermeden kunnen worden. Zoals in de vorige paragraaf vermeld, leidde het opslaan van dezelfde gegevens op meerdere bestanden tot extra verwerkings- en onderhoudskosten alsmede tot een verhoogd risico van inconsistente informatie.

Een database systeem dient erop gericht te zijn deze doublures te voorkomen en zoveel mogelijk te bereiken dat een gegeven maar eenmaal behoeft te worden opgeslagen.

Dit laatste met in achtname van handhaving van de toegangsmogelijkheden voor een grote verscheidenheid van gebruikers.

ad 2 „Availability of data"

De gegevens opgeslagen in de database, moeten zo doelmatig en tijdig als mogelijk beschikbaar zijn t.b.v. zeer uiteenlopende categorieën gebruikers met verschillende verwerkingswijzen, elk met de door hen gewenste taal waarin zij instructies aan de computer geven en tenslotte met verschillende behoeften.

Als gebruikers kunnen we aantreffen: afdelingsmedewerkers, boekhouders, chefs, programmeurs, etc.

De verwerkingswijze kan variëren van groepsgewijze (batch) verwerking tot dialoog (real-time) toepassingen.

Elk van de gebruikers zal met de database communiceren in een taal die voor hem het meest geschikt is.

De beeldstation bediende zal bijv. beschikken over enkele eenvoudige functies waarmee hij bepaalde opvragingen of mutaties kan uitvoeren. De programmeur zal zich voor administratieve toepassingen bedienen van de programmeertaal Cobol en voor meer wetenschappelijke programma's van bijv. Fortran of Algol.

Hun behoeften kunnen tenslotte uiteenlopen in eenmalige of herhaalde uit te voeren bewerkingen of verzoeken tot informatie.

Deze grote verscheidenheid aan wensen zal door een database vervuld moeten kunnen worden wil men inderdaad kunnen spreken van een gegevensopslag die voldoet aan de wensen die de gebruikers hieraan stellen.

ad 3 „Evolvability"

Deze doelstelling houdt in dat ook in de toekomst de database moet kunnen blijven voldoen aan een zo groot mogelijke verkrijgbaarheid van gegevens.

Dat betekent dat de database met de organisatie mee moet kunnen groeien; zowel in omvang als met veranderingen in de behoeften van de gebruikersomgeving.

Wil dit mogelijk zijn dan moet de database zodanig geconstrueerd worden dat aanpassingen t.b.v. het groeiproces uitgevoerd kunnen worden op een wijze dat de gebruikers hiervan geen nadeel ondervinden en zonder dat dit met al te hoge kosten gepaard gaat. Deze doelstelling hangt nauw samen met het nadeel van programma afhankelijkheid dat in de vorige paragraaf over de conventionele bestandsstructuur werd vermeld.

Hier immers hadden veranderingen in de opslagstructuur en/of verwerkingswijze tot gevolg dat de met deze bestanden werkende programma's aangepast moesten worden.

In een dynamische omgeving als die van een database, waar veranderingen eerder regel dan uitzondering zijn, is het begrijpelijk dat dit niet gepaard kan gaan met evenzovele veranderingen in de, de database benaderende, programma's.

Ten einde de doelstelling van „evolvability" dan ook te bereiken zijn de databases zo geconstrueerd dat de wijze waarop de gegevens zijn opgeslagen in de database, de zgn. „fysieke structuur", is losgemaakt van de structuur die een gebruiker in de gegevensverzameling „ziet", de zgn. „logische structuur".

Deze ontkoppeling noemt men „data independence" ofwel programma onafhankelijkheid, in paragraaf 5 wordt dit begrip verder behandeld.

De verbinding tussen de logische en fysieke structuur wordt gelegd door een stuk programmatuur, het „Data Base Management System, DBMS") dat in samenwerking met het bedieningssysteem (Operating System) er voor zorgdraagt dat fysiek wordt opgehaald of weggeschreven wat logisch nodig is.

ad 4 „Integrity"

Het begrip „integrity" slaat op het handhaven bij voortduring van de juistheid en volledigheid van de in de database opgeslagen gegevens, alsmede het beschermen tegen ongewenste of ongeautoriseerde kennisname ervan (het privacy aspekt).

Deze doelstelling volgt expliciet uit de reeds eerder genoemde doelstellingen.

Het voorkomen van doublures bij de gegevensopslag heeft ten gevolge dat de risico's van verminking, verlies etc. steeds meer worden geconcentreerd op gegevens die niet meer elders (op andere bestanden) beschikbaar zijn.

Anderzijds hebben een groter aantal gebruikers dan vroeger toegang tot dezelfde gegevens, hetgeen de kans op fouten verhoogt.

Bovendien ontstaat door het samenbrengen van gegevens in een gemeenschappelijke database het gevaar dat gebruikers zich toegang verschaffen tot gegevens waartoe zij niet geautoriseerd zijn. Tenslotte wordt door het samenvoegen van gegevens ook de kwetsbaarheid van de onderneming verhoogd m.b.t. de continuïteit in de informatievoorziening.

Een technische storing in de computer, het uitvallen van de database, dit alles heeft tot gevolg dat *alle* gebruikers geen toegang meer tot de gegevens hebben.

Ik hoop dat uit het voorgaande duidelijk geworden is dat integriteitshandhaving van de database een uitermate belangrijke doelstelling is. Alle vier genoemde doelstellingen zijn zoals reeds eerder vermeld in de praktijk nog maar ten dele gerealiseerd.

De nu op de markt zijnde database management systemen vervullen alle maar één of enkele van de genoemde doelstellingen.

Hieruit mag blijken dat een zorgvuldige evaluatie van database management systemen gewenst is alvorens tot aanschaf van één ervan wordt besloten.

5 Het begrip data-independence

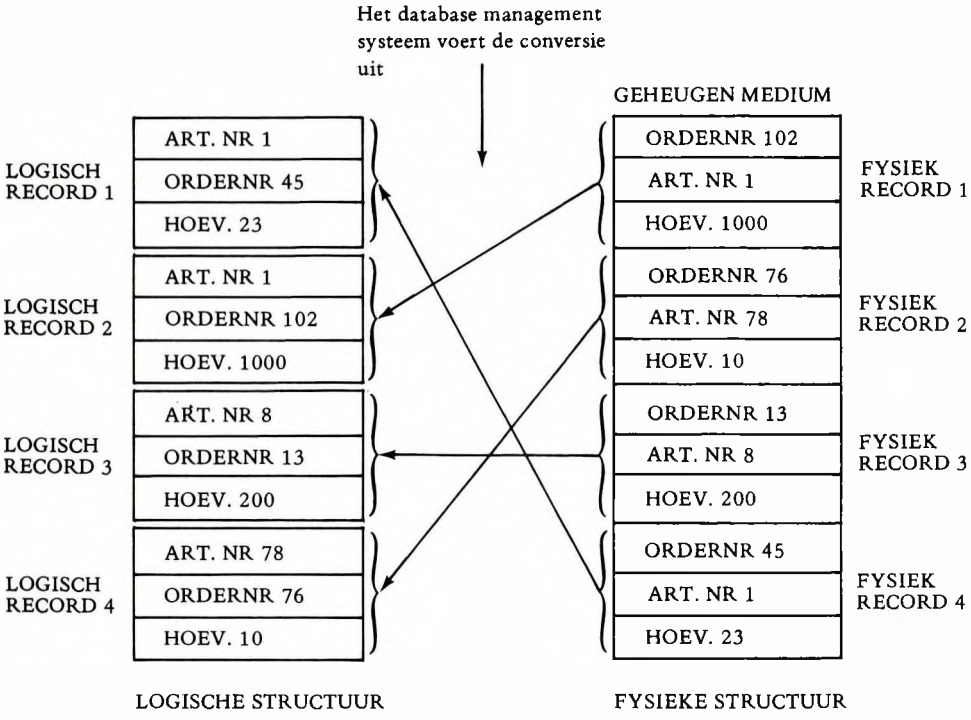
Zoals reeds eerder vermeld is data-independence noodzakelijk om de doelstelling „evolvability" te kunnen bereiken.

Data-independence is hierbij omschreven als de ontkoppeling van de wijze waarop de gegevens zijn opgeslagen (de fysieke structuur) van de structuur die een gebruiker in de gegevensverzameling „ziet" (logische structuur).

Fysiek verwijst hier naar de manier waarop de gegevens fysiek zijn vastgelegd op een informatiedrager, logisch daarentegen verwijst naar de manier waarop de gegevens worden gepresenteerd aan de programmeur of de daadwerkelijke gebruiker ervan.

Een eenvoudig voorbeeld ter illustratie:

In een bedrijf zijn op een bestand de ontvangen orders in volgorde van binnenkomst vastgelegd. De vastgelegde gegevens bevatten het artikelnummer, het ordernummer en de gewenste hoeveelheid. De gebruiker wenst een overzicht van de orders per artikel gesorteerd op volgnummer van artikelnummer en ordernummer.



Figuur 1

We zien uit het voorbeeld dat de gebruiker zich geen zorgen hoeft te maken over de fysieke structuur. De wijze waarop de gegevens zijn opgeslagen op het extern geheugen medium is voor hem niet relevant. Zijn behoefte aan een bepaalde logische structuur wordt door het database management systeem vervuld; deze programmatuur verricht de vereiste conversie.

In dit geval is het een conversie van velden en volgorde. Deze conversie wordt in de literatuur „mapping" genoemd.

Bijzonder belangrijk is nu de bereikte scheiding tussen logische en fysieke structuren, deze heeft als resultaat dat veranderingen die in een van de structuren worden aangebracht geen invloed hebben op de andere structuur.

Wordt de wijze van opslag, de benaderingswijze etc. veranderd, dan heeft dit geen consequenties voor de gebruiker en verandert er omgekeerd iets aan de logische structuur, bijv. door programma-aanpassingen, dan heeft dit geen consequenties voor de fysieke structuur.

Het grote voordeel van data-independence is dat het onderhoud van bestaande programma's zeer efficiënt kan zijn, aanpassingen aan veranderingen in de bestandenorganisatie, recordindelingen e.d. worden vermeden. Hier ook hebben veranderingen in bepaalde programma's geen invloed op andere programma's.

Anderzijds kan de bestandsorganisatie voortdurend worden aangepast aan de wisselende eisen zonder dat dit kostbare veranderingen in de programma's nodig maakt.

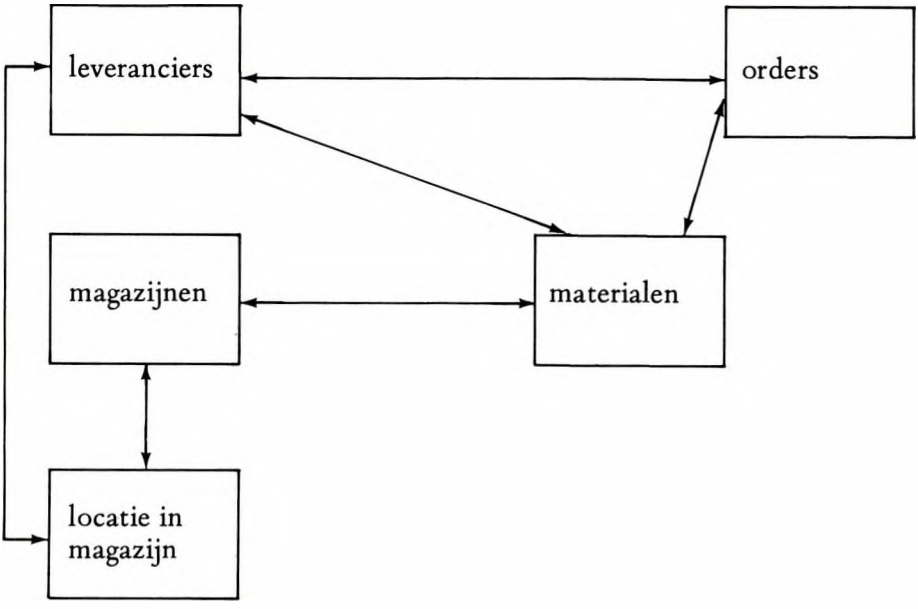
m a b blz. 444

In de volgende paragraaf zal worden nagegaan hoe data-independence in een database wordt gerealiseerd.

6 De logische structuur van een database

De logische structuur is omschreven als de wijze waarop de gebruiker de gegevens „ziet".

Ter illustratie hiervan zullen we het voorbeeld van een onderneming met stukproduktie eens nader bekijken (figuur 2).



Figuur 2

Deze onderneming zal o.a. informatie nodig hebben over de in bewerking zijnde orders, de voor de orders verbruikte materialen, de leveranciers van deze materialen, de magazijnen waar ze zijn opgeslagen en de lokatie ervan in het betrokken magazijn.

Dit zullen in grote lijnen de objecten zijn waarover informatie zal worden vastgelegd.

Eveneens valt op te merken dat er tussen de verschillende grootheden relaties bestaan, deze worden in figuur 2 met pijlen aangegeven.

Er is een relatie tussen leveranciers en materialen: elke leverancier levert bepaalde materialen en bepaalde materialen worden door maar enkele leveranciers geleverd.

Materialen worden in orders gebruikt en omgekeerd gebruiken orders bepaalde materialen. Materialen worden opgeslagen in magazijnen en magazijnen worden gebruikt om materialen op te slaan.

U ziet dat er, behalve dat globale beeld van informatie over de grootheden

leveranciers, orders, materialen, magazijnen en locaties nu bepaalde relaties tussen deze grootheden zijn aangebracht.

Deze relaties zijn niet voor iedere gebruiker even interessant: de produktiechef is geïnteresseerd in de relatie orders - materialen als antwoord op de vraag is er voldoende materiaal om de orders te kunnen uitvoeren.

Voor de magazijnbeheerder is de relatie magazijn - materialen relevant, en dan alleen voor het magazijn waarvan hij beheerder is, de voorraadbeheerder daarentegen is geïnteresseerd in de totale voorraadpositie.

Resumerend kunnen we zeggen dat we begonnen zijn met het schetsen van een globaal totaalbeeld van de door de onderneming benodigde informatie en dat we vervolgens binnen dit kader voor enkele gebruikers de voor hen specifieke informatie hebben aangegeven. In database terminologie zouden we zeggen dat we praten over de logische structuur van de informatie.

Voor het totaalbeeld van de logische structuur gebruikt men dan de engelse termen „data structure" of „schema".

Voor het specifieke beeld dat elke gebruiker heeft van zijn informatiebehoefte hanteert men de term „subschema".

Het is goed om hierbij nogmaals te benadrukken dat zowel het „schema" als „subschema" volkomen los staan van de wijze waarop deze gegevens fysiek worden opgeslagen (de fysieke structuur), het is echter wel zo dat door het database management systeem de relaties tussen de verschillende grootheden bij de fysieke opslag van gegevens moeten worden meegegeven, dit geschiedt d.m.v. het meegeven van wijzers of „pointers" aan de opgeslagen gegevens.

Het is een van de taken van de hierna nog te beschrijven database beheerder of database administrator om de totale logische structuur (schema) en de subschema's in een dusdanige vorm te beschrijven dat het Data Base Management Systeem hiermede de „mapping" kan uitvoeren. Deze beschrijving gebeurt m.b.v. een hiertoe speciaal gemaakte taal, de „Data Description Language" (DDL).

7 De werking van een database

In deze paragraaf zal geprobeerd worden een totaalbeeld te geven van de werking van een database management systeem (DBMS) en de onderdelen ervan (zie figuur 3).

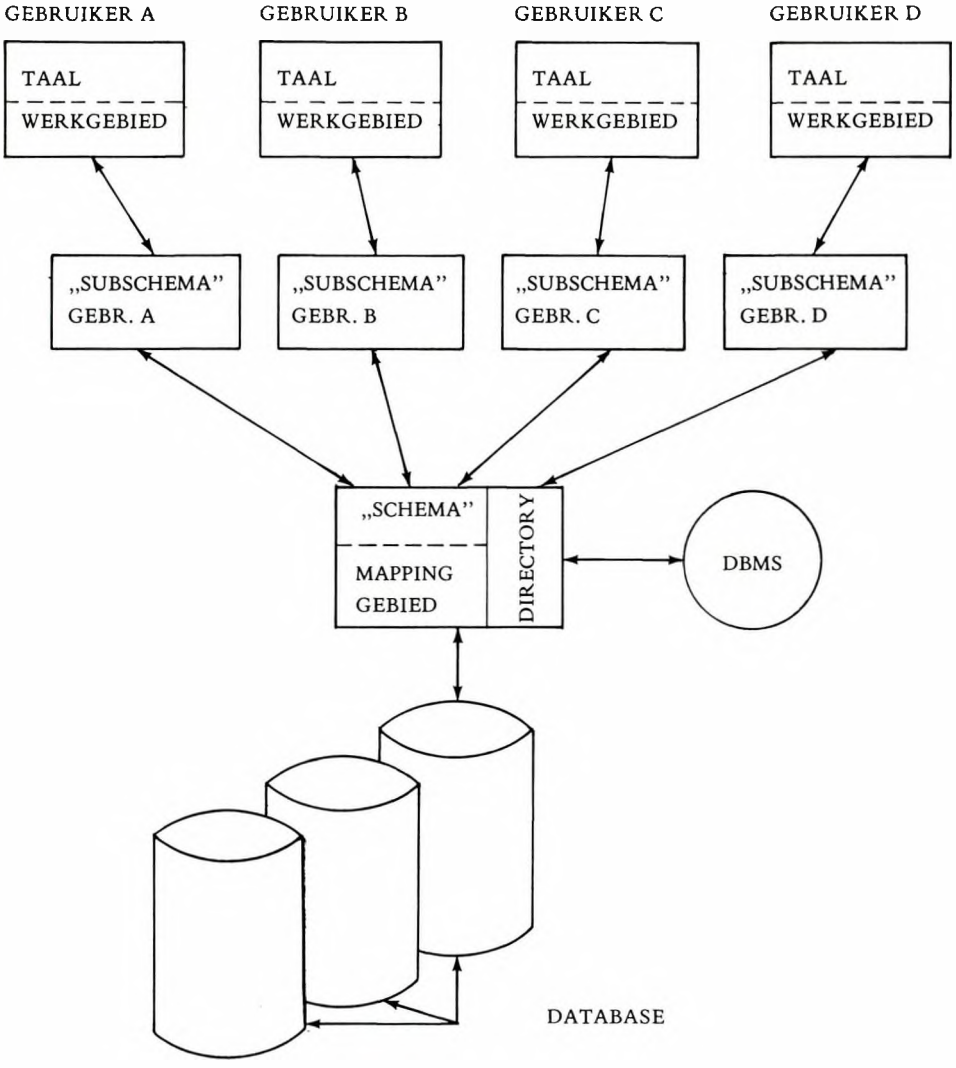
De GEBRUIKERS zijn applicatieprogrammeurs of terminalgebruikers met een uiteenlopende mate van kennis.

Elke gebruiker heeft een TAAL tot zijn beschikking.

De programmeur zal werken met Cobol of Assembler, de terminal gebruiker zal waarschijnlijk een speciale taal tot zijn beschikking hebben.

Elk van hen heeft in zijn taal een zogenaamde DATA SUBLANGUAGE (DSL), hetgeen een onderdeel van de taal is dat het lezen en wegschrijven van gegevens regelt.

Elke gebruiker heeft een WERKGEBIED dat optreedt als het ontvangsten zendgebied voor alle gegevens die tussen de gebruiker en de database worden overgestuurd.



Figuur 3

De DATABASE zijn de gegevens zoals deze fysiek op direct toegankelijke geheugenmedia, zoals bijv. magneetschijf, zijn opgeslagen. Tussen de database en de gebruikers bevindt zich het data-model of „SCHEMA".

Het data-model is in wezen het logische totaalbeeld van de informatie-inhoud van de gehele database, zoals deze door elk van de verschillende gebruikers wordt gezien.

De Data Base Administrator beschrijft de logische gegevensstructuur van de database in zijn totaliteit m.b.v. de „Data Description Language" (DDL).

Elke gebruiker verwacht in zijn werkgebied een record zoals hij dat behoeft, het zogenaamde „logisch record".

Het DBMS heeft de taak ervoor te zorgen dat de gebruiker dit logisch record ontvangt.

De Data Base Administrator beschrijft het logisch record zoals de gebruiker dat ziet met behulp van de „Data Description Language" (DDL), de instructies hiervan vormen het „SUB-SCHEMA" of logisch record voor een gebruiker.

Dit „SUB-SCHEMA" is volkomen gegevensonafhankelijk, het geeft alleen de namen van de benodigde velden, de gewenste representatie, de relaties tussen de velden, de gewenste bewerkingen (bijwerken, lezen etc.) en de gewenste volgorde.

Dit „SUB-SCHEMA" moet nu door het DBMS uit de database gehaald worden, dit noemt men „MAPPING".

Het data-model moet vertaald worden naar de fysieke opslagvorm in de database.

Dit houdt in dat er „MAPPING DEFINITIES" opgesteld moeten worden die o.a. aangeven:

- hoe het „SUB-SCHEMA" vertaald moet worden naar de opgeslagen records;
- welke indices er bestaan;
- de vorm van opslag van de records;
- de volgorde van de opgeslagen records etc.

Indien de opslagstructuur en de benaderingswijze van de opgeslagen records veranderen heeft dit tot gevolg dat alleen de Mapping Definities veranderen, niet de Data Models.

De gebruiker merkt niets van de veranderingen in de Database, het onderhoud van de Mapping Definities is een taak van de Data Base Administrator.

Om onze beschrijving van de structuur van een DBMS te vervolgen: het DBMS is de programmatuur die de toegang tot de Database regelt. Elk verzoek tot toegang tot de Database - geschreven in een Data Sublanguage - wordt geïnterpreteerd door het DBMS dat daarbij de SUB-SCHEMA'S en de Mapping Definities gebruikt.

Uit de Database worden de velden gelezen die voldoen aan de Mapping Definities en in het MAPPING GEBIED geplaatst. Indien de gewenste representatie verschilt van die in de Database worden de noodzakelijke conversies uitgevoerd, waarna het record in het werkgebied van de gebruiker wordt geplaatst.

De „SUB-SCHEMA'S" en Mapping Definities vormen een onderdeel van de Data Definition Directory of DIRECTORY, een verzamelnaam van alle informatie die het DBMS nodig heeft om zijn taken te vervullen. De Directory wordt onderhouden door de Data Base Administrator, behalve de Sub-Schema's en Mapping Definities bevinden zich o.a. hierin:

- de USER DIRECTORY: de identificatie van alle gebruikers die geautoriseerd zijn tot toegang tot de Database.
- de DATA SET DIRECTORY: de identificatie van alle bestanden in de Database met de namen, de adresseringssleutel, de organisatie methode en de velden.

- de FIELD DIRECTORY: de identificatie van alle velden in de bestanden met de namen ervan, de bestandsnaam, de representatie, de geautoriseerde gebruikers en de door hen uit te voeren bewerkingen.

Door gebruik te maken van deze Directory is het mogelijk de toegang van elke gebruiker tot de Database te regelen tot op het niveau van een veld en tevens de voor hem toegestane bewerkingen.

Op deze wijze wordt een groot deel van de in paragraaf 4 genoemde doelstelling van het handhaven van de integriteit der gegevens gerealiseerd.

8 De Data Base Administrator

De functie van Data Base Administrator (DBA) of Data Base Beheerder is in het voorgaande meermalen ter sprake gekomen. De DBA is de persoon die centraal het beheer voert over de gegevens en het Data Base Management Systeem. De functie van DBA vloeit logisch voort uit het idee van „shared data"; indien de gegevens door meerdere gebruikers worden benut ontstaat de noodzaak tot een coördinator. Ook de hoge technische complexiteit van een DBMS maakt een dergelijke speciale functie gewenst.

Voor de controlerende accountant is het belangrijk de verschillende aspecten van deze functie eens nader te bekijken.

Gordon Everest, die op dit gebied veel onderzoek heeft gedaan, noemt 5 specifieke taken van de DBA:

1 de definiëring, creatie en vernietiging van gegevens;
2 het voorzien in toegangsmiddelen tot de Database;
3 het onderhouden van contacten met de gebruikers;
4 het in stand houden van de integriteit van de Database;
5 het waarnemen en bewaken van het gebruik van de Database.

ad 1 de definiëring, creatie en vernietiging van gegevens

De DBA is de functionaris die de behoefte van de gebruikers aan gegevens - het logisch record - interpreteert en ervoor zorgdraagt dat het DBMS de gewenste gegevens oplevert.

Dit houdt in dat hij het logisch record beschrijft in de Data Definition Language van het betrokken DBMS, de Mapping Definities opstelt en de verschillende Directories bijwerkt.

Betreft het nieuwe gegevens dan moet ervoor gezorgd worden dat deze in de Database komen in overeenstemming met de logische en fysieke structuur, veelal gaat dit gepaard met uitgebreide validatiecontroles. Ook aanpassing van de logische definiëring aan de wensen van de gebruikers valt onder dit punt.

Indien de behoefte aan bepaalde gegevens afneemt is het ten slotte gewenst deze uit de database te verwijderen en over te brengen naar andere bestanden; de DBA verricht deze taak in nauw overleg met de leiding.

ad 2 het voorzien in toegangsmiddelen tot de Database
Om de doelstelling van verkrijgbaarheid der gegevens te bereiken moet de DBA voorzien in de benodigde faciliteiten en hulpmiddelen om elke gebruiker in staat te stellen de Database te benaderen op de meest geschikte manier. Dit noodzaakt de DBA tot aankoop of ontwikkeling van deze hulpmiddelen.

ad 3 het onderhouden van contacten met de gebruikers
De DBA is de dienaar van de verscheidenheid van personen waaruit de gebruikersomgeving bestaat.

Zijn taak is het inlichten en helpen van afdelingsemployés, topleiding, programmeurs en systeemanalisten.

De contacten met de gebruikersomgeving hebben betrekking op:
- voorlichting van de gebruikers over de mogelijkheden van de Database;
- opleiding van de gebruikers;
- het verlenen van hulp en advies aan de gebruikers.

Dit impliceert dat er documentatie moet zijn over de structuur en inhoud van de Database, wat erin is opgeslagen, wat de betekenis van de opgeslagen gegevens is, hoe de gegevens benaderd kunnen worden, de gebruikte veldnamen etc.

De database documentatie moet voor de gebruikers beschikbaar zijn en zij moeten op de hoogte worden gesteld van voor hen relevante veranderingen.

Een groot deel der contacten zal verder betrekking hebben op de onder punt 1 en 2 genoemde taken van de DBA.

ad 4 het in stand houden van de integriteit van de Database
Hiermede wordt bedoeld dat de gegevens in de Data intact moeten blijven.

Over deze taak van de DBA zijn in de literatuur, voor zover mij bekend, nog geen uitspraken gedaan over de concrete verantwoordelijkheid van de DBA.

Naar mijn mening kan de DBA alleen de verantwoordelijkheid op zich nemen voor 2 aspecten van de Database integriteit nl.
a. de verantwoordelijkheid dat alleen personen die daarvoor autorisatie hebben de gegevens in de Database kunnen benaderen en daarmede alleen de bewerkingen kunnen uitvoeren waarvoor zij een specifieke opdracht hebben;
b. de verantwoordelijkheid dat de gegevens welke de geautoriseerde gebruikers in de database opslaan hierin ongewijzigd blijven.

ad a.
Voor regulering van de toegang dienen de reeds eerder genoemde Directories. Het is belangrijk dat er voor de autorisatie een goede administratieve procedure - zoals bijv. vergelijkbaar met het geven van procuratie - wordt ontworpen, met scherp toezicht van de leiding.

ad b.
De DBA kan alleen een formele verantwoordelijkheid op zich nemen voor

de juistheid van de opgeslagen gegevens, dat wil zeggen hij kan de regels stellen voor de opslag van de gegevens opdat deze door het DBMS verwerkt kunnen worden.

De DBA is (naar mijn mening) niet verantwoordelijk voor de materiële inhoud van de gegevens, hij dient er alleen voor te zorgen dat de gegevens welke de diverse gebruikers in de database onderbrengen, daarin blijven in de vorm en hoedanigheid waarin de gebruiker deze heeft aangeboden.

De DBA bewaart de gegevens, de gebruikers voeren er het beheer over. De DBA heeft voor het handhaven van de integriteit der gegevens een aantal middelen tot zijn beschikking:

- hij kan zorgen voor de regelmatige aanmaak van BACK-UPS, dat zijn additionele bestanden waarop de database in zijn geheel gedupliceerd kan worden, of de situatie voor en na het verwerken van een transactie kan worden vastgelegd;
- hij kan een zogenaamde LOGTAPE laten aanmaken, een magneetband waarop alle verwerkte transacties worden vastgelegd met additionele controlegegevens zoals bijv. het transactienummer, de autorisatiecode van de opdrachtgever, het terminalnummer etc.;
- voor de toegangscontrole beschikt de DBA over de eerder genoemde Directories waarin tot op het laagste niveau van de database kan worden aangegegeven welke gebruikers, welke handelingen mogen verrichten met de betreffende gegevens;
- als laatste maatregel valt te noemen het voorschrijven door de DBA van uiterst stringente validatiecontroles alvorens gegevens de database binnenkomen. Deze kunnen worden uitgebreid tot de momenten waarop de gegevens worden opgeslagen of de Database verlaten.

Een regelmatige controle van de database op juistheid en volledigheid van de opgeslagen gegevens en relaties is onmisbaar, juist doordat de database de centrale opslagplaats is of wordt van *alle* voor de onderneming relevante gegevens.

De noodzaak hiertoe bestaat te meer omdat Data Base Management Systemen zeer complexe systemen zijn waarin storingen kunnen optreden. De relaties tussen de verschillende gegevens worden, zoals in paragraaf 6 beschreven weergegeven door „pointers", deze kunnen wegvallen of verminkt raken en dan zijn er in feite gegevens verloren gegaan.

ad 5 het waarnemen en bewaken van het gebruik van de Database

De DBA dient het gebruik te observeren en statistisch materiaal hierover te verzamelen opdat de database voortdurend aan de behoefte van de gebruikers kan voldoen.

Dit dient vooral voor aanpassing van de logische en fysieke structuur van de database.

De veelheid van taken wijst erop dat de DBA functie vrijwel niet door één persoon te vervullen is. De huidige ontwikkelingen gaan dan ook in de richting dat de DBA functie door meerdere personen wordt uitgevoerd.

9 Slot

In het voorgaande is geprobeerd het fenomeen data base te beschrijven aan de hand van de ontwikkelingen in de technieken van bestandsorganisatie.

De Data bases zijn vooral ontwikkeld als middelen ter voorkoming van inefficiency in de gegevensopslag en de verkrijging van een zo flexibel mogelijke aanpassing aan veranderende gebruikerswensen. Deze doelstellingen worden gerealiseerd door de ontkoppeling van de logische - en fysieke structuur van de gegevensverzamelingen.

Verder werd nader ingegaan op de functie van Data Base beheerder, de functionaris belast met het beheer over het Data Base Management Systeem, en de uit deze functie voortvloeiende taken.

Zoals in de inleiding vermeld, is in dit artikel alleen ingegaan op de technische aspecten van data bases. In een volgend artikel zullen de implicaties voor de niet gespecialiseerde accountant belast met algemene controle-opdrachten worden besproken. Met name zal dan nader worden ingegaan op de gevolgen voor de beoordeling en toetsing van de interne controle en het gebruik van standaardprogrammatuur t.b.v. de accountantscontrole.

Literatuurlijst

- Richard G. Canning: „The 'data administrator' function", EDP Analyzer (10:11), 1972 November.
- Richard G. Canning: „Problem areas in data management", EDP Analyzer, XII, 1973 March.
- Richard G. Canning: „That maintenance Iceberg", EDP Analyzer, X, 1972 October.
- E. F. Codd and C. J. Date: „Interactive support for non-programmers: the relational and network approaches", in Randall Rustin, e.d., ACM Sigmod workshop on data description, access and control, 1974, Vol. II (New York: Association for computing machinery, 1975).
- C. J. Date: „An introduction to database systems (Reading, Massachusetts: Addison - Wesley Publishing Company, 1975).
- CODASYL Programming Language Committee, Data Base Task Group Report (New York: Association for Computing Machinery, 1971).
- R. W. Engles: „A tutorial on data base organisation" Annual Review in Automatic Programming, Vol. 7, Part 1 Pergamon Press (July, 1972).
- Gordon C. Everest: „The objectives of database management" (Minneapolis, Minnesota: University of Minnesota, Management Information Systems Research Center Working Paper 71-04, May 1972).
- Gordon C. Everest: „Database Administrator: organizational role and functions" (Minneapolis, Minnesota: University of Minnesota, Management Information Systems Research Center Working Paper 73-05, September 1973).
- Gordon C. Everest: „Data base management: objectives, systems and administration". Komt uit bij McGraw-Hill in begin 1976.
- James Martin: „Computer Data-base Organisation", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.
- D. Overkleeft, R. van Dooren, E. J. Ruff: „Gegevensbanken" Kluwer/Philips Technische Bibliotheek, enz. 1973.