ABIS NIEUWSBRIEF VOOR DB2 PROFESSIONALS

JAARGANG 4 - NUMMER 2, JANUARI 2006

OPEN CURSOR

Integratie. Een modewoord. Men wil, moet, zal, ... alles integreren.
Dat hierbij gegevens betrokken zijn zal ons geenszins verbazen, maar evident is het daarom niet.
Gegevens worden beheerd door een reeks systemen die niet zo maar met elkaar te combineren zijn. Zelfs als ze op dezelfde machine, ondersteund door hetzelfde operating systeem, actief zijn.
Niet alleen de systemen, maar ook de aard van de gegevens spelen een rol. Denk maar aan het combineren van informatie over een productaankoop met de telefoongesprekken die daarrond plaats hebben gehad.
Moeilijk wil echter niet zeggen onmogelijk. Producten zoals 'DB2 Content Manager' helpen ons bij het integreren van gegevens. Het hoe daarvan proberen we in dit en volgende nummers toe te lichten.

Het ABIS DB2-team.

IN DIT NUMMER:

- In het vorige nummer hadden we het over DB2 datummanipulaties - daar is nog een staart aangekomen: *Over datum, tijd en tijdsduur in DB2 - 2.*
- Ook de reeks over de verschillen tussen DB2, Oracle en SQL Server gaat verder. Dit keer gaat de aandacht naar external storage: *SQL Server Architecture: DB-Filegroup-Extent.*
- We beginnen ook een nieuwe reeks. Content management wordt steeds belangrijker. Ook DB2 heeft hierin een rol te spelen. Meer hierover in: *DB2 en Content Management - 1.*
- *Cursusplanning januari 2006 - juni 2006.*

CLOSE CURSOR

In een volgend nummer gaan we verder met het uitdiepen van de DB2 Content Manager. Daarnaast bekijken we in een ander artikel hoe we vanuit WSAD een DB2 systeem kunnen benaderen en gebruiken

Tot dan!

Over datum, tijd en tijdsduur in DB2 - 2

Peter Vanroose (ABIS)

In het vorige nummer vroegen we naar efficiënte manieren om uit een tabel enkel die entries te halen die betrekking hebben op de vorige kalendermaand.

```
WHERE my_date BETWEEN
LAST_DAY(CURRENT DATE - 2
MONTHS) + 1 DAY
AND LAST_DAY(CURRENT_DATE-
1 MONTH)
```

Op z/OS kan hiervoor de functie LAST_DAY zoals hiernaast aangegeven, gebruikt worden.

```
WHERE my_date BETWEEN
DATE(SUBSTR(CHAR(CURRENT
DATE - 1 MONTH, ISO) , 1 ,
8) || '01')
AND
DATE(SUBSTR(CHAR(CURRENT
DATE, ISO) , 1 , 8) || '01')
- 1 DAY
```

Meer algemeen (dus zonder LAST_DAY te gebruiken) kan dit ook met de tweede formulering.

Let op: vergeet het tweede argument ISO van CHAR() niet; er is anders geen garantie dat een conversie uitgevoerd wordt naar een ISO-datum-representatie (yyyy-mm-dd)!

Bemerk dat beide oplossingen performant zijn, vermits de tekst- en datummanipulaties slechts eenmalig en dus niet op elke rij van de tabel moeten uitgevoerd worden.

```
WHERE my_date BETWEEN
CURRENT DATE - (DAY(CURRENT
DATE)-1) DAYS - 1 MONTH
AND
CURRENT DATE - DAY(CURRENT
DATE) DAYS
```

Het is nog mooier als we geen tekstmanipulaties nodig hebben zoals de derde variant aangeeft.

Voor het "vorige kwartaal" wordt het uiteraard nog iets omslachtiger...

Een totaal andere aanpak, die niet afhangt van de waarde van CURRENT DATE op het ogenblik van de uitvoering van de query, bestaat uit het opnemen van het maandnummer als een extra kolom in de basistabel. Uiteraard is dit niet altijd mogelijk, maar dan valt dit te simuleren via een JOIN met een hulptabel, met daarin een datum-kolom en een maandnummer-kolom. Deze redenering doortrekkend, volstaat het eigenlijk, een hulptabel van 1 datumkolom en hoogstens 31 rijen te maken, met daarin alle datums van vorige kalendermaand.

```
SELECT * FROM my_table
INNER JOIN  hulptabel ON
my_table.datumveld =
hulptabel.kolom
```

De gevraagde query wordt dan de variante hiernaast aangegeven.

```
SELECT * FROM my_table  MT
WHERE EXISTS
(SELECT 1 FROM hulptabel
WHERE MT.datumveld = kolom)
```

Alternatief voor een inner join is uiteraard de correlated subquery.

Met dank aan de inzenders van deze alternatieven: *Christiaan Haverbeke* (Dexia) en *Davy Goethals* (Sidmar).

SQL Server Architecture: DB-Filegroup-Extent

Eric Everaert (ABIS)

In ons vorige nummer hebben we de architectuur van het instance en van de SQL Server database besproken. Nu beschrijven we hoe SQL Server zijn database-inhoud opslaat op schijf.

De database

Logisch beschouwd

Zoals in DB2 en in Oracle, maakt SQL Server gebruik van logische containers om er al zijn gegevens in te bewaren. Maar in tegenstelling tot de concurrentie waar men dit een tablespace noemt, is de terminologie hier 'filegroup'. Verder zijn er weinig verschillen tussen een filegroup en een tablespace. Het betreft hier eveneens een verzameling blocks die gegroepeerd worden per extent.

Fysiek beschouwd

SQL Server brengt de filegroups fysiek onder in bestanden op het bestandssysteem. Er zijn twee types bestanden: het primaire bestand en de secundaire bestanden.

- *Primair bestand:* dit bestand, met als standaard extensie ".mdf", is geassocieerd met de primary filegroup en wordt dus aangemaakt op het moment dat de database wordt aangemaakt. Het is mogelijk dat de gegevens van een SQL Server database enkel in dit bestand worden ondergebracht (de logs gaan uiteraard wel sowieso in een apart bestand). Het bevat alle nuttige informatie voor het starten van de database. Er is echter niets dat u belet om er ook gegevens in op te slaan. Elke database bestaat dus op z'n minst verplicht uit een primair gegevensbestand.

- *Secundaire bestanden:* deze bestanden bevatten alle gegevens die niet in het primaire gegevensbestand passen. Deze bestanden worden pas aangemaakt wanneer het primaire gegevensbestand niet groot genoeg meer is om alle gegevens van de database te bevatten. Het is vaak interessanter om al vanaf het begin met meerdere bestanden te werken, om dezelfde redenen die DBA's van andere systemen ertoe aanzetten om de gegevens over bestanden te verdelen.

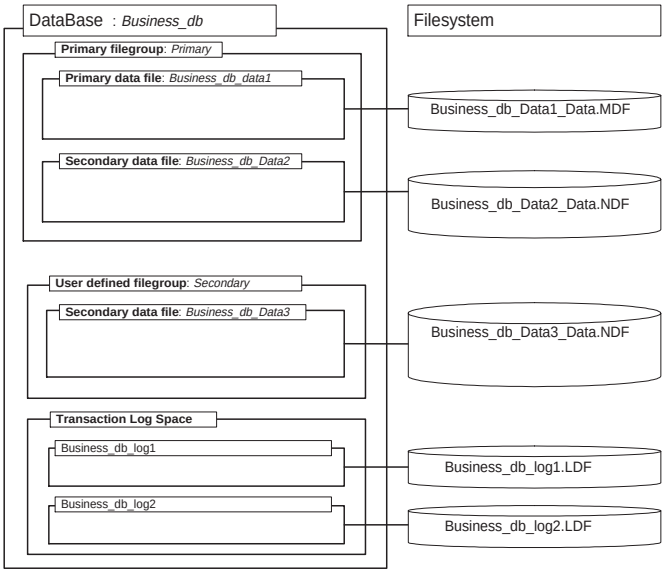
Een filegroup

Er zijn twee soorten filegroups – vergelijkbaar met een tablespace zoals hierboven uitgelegd: de primaire filegroup en de gebruikersgedefinieerde filegroup. De primaire filegroup bevat het primaire bestand. Hij bevat dus alle pagina's van de systeemtabellen. Eén van de filegroups van een database wordt ook beschouwd als de standaardfilegroup. Het is eveneens mogelijk om een filegroup te gebruiken in alleen-lezen-modus (dit geldt niet voor de primaire filegroup).

Zoals reeds vermeld, kan een filegroup uit meerdere bestanden worden opgebouwd. Als er in dit geval een object in deze filegroup geplaatst wordt, worden beide bestanden tegelijkertijd gebruikt. Daarbij schrijft SQL Server alternerend een deel van de gegevens in het ene bestand en een ander deel in het andere bestand. Dit is dus een verschil met andere systemen, met name Oracle, waar deze bestanden één na één worden opgevuld.

Wat de bestanden betreft, kunnen deze een vooraf vastgelegde grootte hebben, of ze kunnen groeien al naargelang de behoefte. In dit laatste geval kan de grootte van de toevoegingen ook worden ingesteld, evenals een maximumgrootte voor het volledige bestand.

Figuur 1: Filegroups in SQL Server



Op het schema (zie figuur 1) kunnen we dus zien dat onze database "Business_db" twee filegroups bevat:

- de eerste filegroup werd "Primary" genoemd en deze werd in twee bestanden ondergebracht, waarbij we er willen op wijzen dat deze door de DB gekende logische namen hebben gekregen en fysieke namen die in het bestandssysteem worden gebruikt.

- de tweede filegroup met als naam "Secondary", wordt in één enkel bestand ondergebracht (uiteraard kan u de namen zelf vrij kiezen).

Indien u de grootte van de database wenst uit te breiden, kan u:

- ofwel de grootte van de bestanden verhogen door het instellen van een nieuwe vaste grootte, of ze laten groeien, naarmate de behoefte;

- ofwel een bestand toevoegen aan een bestaande filegroup;

- ofwel een nieuwe filegroup toevoegen die rechtstreeks gekoppeld is met een bestand.

De pagina's en de extents

Wanneer een object, hetzij een tabel hetzij een index, wordt aangemaakt in een filegroup, worden hieraan extents toegewezen om er de rijen in op te slaan. Zoals dat het geval is bij de concurrentie, bestaat een extent uit een groep blocks (of pagina's) die logisch opeenvolgend zijn. Het beheer van de extents is sterk vereenvoudigd, aangezien SQL Server enkel extents van gelijke grootte gebruikt, met name 64k, opgebouwd uit 8 blocks van 8k. De grootte van de blocks ligt eveneens vast. Net zoals in DB2 en in tegenstelling tot hoe het er in Oracle aan toe gaat, kan een rij niet in meerdere blocks ondergebracht worden. Een rij kan dus nooit groter zijn dan 8k behalve als men gebruik maakt van specifieke datatypes, vergelijkbaar met de BLOB's (grote binaire objecten) van de andere database-producten.

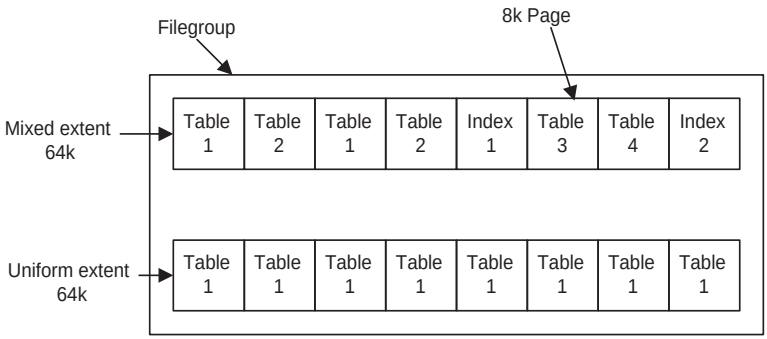
Men moet zich dus niet bekommeren over de grootte van de extents. We voegen hier nog toe dat het interne beheer van de extents en van de beschikbare ruimte lokaal wordt aangepakt op het niveau van de bestanden. Er zijn dus bepaalde blocks die een bitmap bevatten waarmee de toewijzing van de extents beheerd wordt.

SQL Server gebruikt twee soorten extents om er de tabellen en indexen in op te slaan: uniforme en mixed extents (zie ook figuur 2):

- Mixed extent: de 8 pagina's worden door verschillende objecten gebruikt
- Uniform Extent: de 8 pagina's worden alle toegewezen aan hetzelfde object.

Wanneer een object wordt aangemaakt, worden de eerste rijen ervan in een mixed extent ondergebracht. Vervolgens worden er, naarmate de grootte van de tabellen of indexen toeneemt, uniforme extents aan toegewezen.

Figuur 2: types of extents



Wetenswaardigheden i.v.m. SQL Server

Volgende kenmerken zullen DBA's van DB2 wellicht verbazen.

- Een filegroup, die nochtans vergelijkbaar is met een tablespace, kan worden aangemaakt zonder reeds vanaf het begin gebonden te moeten zijn aan een bestand. De filegroup bestaat dan wel, maar bevat geen gegevens. Een tweede eigenaardigheid is dat er geen ruimte wordt toegewezen in de filegroup bij de aanmaak van een tabel. Extents worden pas toegewezen op het moment dat we rijen aan de tabel toevoegen. Het is dus mogelijk om een lege tabel aan te maken in een filegroup die niet gebonden is aan een bestand. Deze tabel is dan uiteraard niet meteen bruikbaar. Een voordeel hiervan is de mogelijkheid om de stappen bij het opzetten van de DB volledig te scheiden. De eerste fase wordt uitgevoerd door de DBA. De tweede fase wordt daarna uitgevoerd door de systeembeheerders die dan vrij bestanden kunnen toewijzen aan gecreëerde filegroups.

Voorbeeld 1: create table en filegroups

```
ALTER DATABASE business_db ADD FILEGROUP fg3
CREATE TABLE tab_test (c char(20)) on fg3
=> het is nu reeds mogelijk om een tabel aan te maken! Er wordt nog
niets toegewezen tijdens de aanmaak van de tabel.
INSERT INTO tab_test VALUES ('abcde');
=> foutmelding, er werd nog geen bestand gebonden aan de filegroup fg3
ALTER DATABASE business_db
ADD FILE (NAME=bus_db_dg3_1,
filename='h:\data\sqlserver\bus_db_dg3_1.ndf')
TO FILEGROUP fg3
INSERT INTO tab_test VALUES ('abcde');
=> OK, dit werkt.
```

- Filegroup SITS? PATS? SEGS?

Onder DB2 moet de container, m.a.w. de tablespace, aangemaakt worden in functie van zijn toekomstig gebruik. Er zijn dan drie keuzes beschikbaar (sequentieel, gesegmenteerd of gepartitioneerd). Onder SQL Server bestaat er slechts één enkel type filegroup, de verdeling van de tabellen overheen de pagina's gebeurt min of meer zoals in een gesegmenteerde tablespace. Twee tabellen delen dus nooit dezelfde pagina's, hoogstens hetzelfde extent (in het geval van een mixed extent). SQL Server 2000 beschikt niet over een partitionering zoals bij DB2. Men kan stellen dat de partitionering hier grotendeels volgens de "view partitioning"-methode gebeurt.

- Filegroup voor indexen of indexspace?

Ook hier, niets van dat alles. De indexen kunnen worden aangemaakt in dezelfde filegroups als de tabellen, al weet elke DBA dat dit niet aan te raden is. Het aantal filegroups is dus veel lager dan het aantal table- en indexspaces die men op een DB2-systeem aantreft.

- Beheer van de filegroups

Men kan voor een filegroup bestanden toevoegen, verwijderen, de grootte ervan aanpassen, ... en wat de tabellen betreft, een deel ervan kan men naar een andere filegroup verplaatsen.

DB2 en content management - 1

Eric Venmans (ABIS)

Inleiding

DB2 was, zoals de meeste DataBase Management Systemen (DBMS), een hulpmiddel om gestructureerde gegevens te bewaren. De kwaliteit van zulke systemen hangt onder meer samen met:

- *de veiligheid:* niet om het even wie mag de gegevens bekijken, er is toegangscontrole nodig (security); gegevens mogen niet zomaar ingevoerd, gewijzigd of verwijderd worden; er is procescontrole nodig (integriteit); na verlies van gegevens (omwille van technische problemen of verkeerde manipulaties) moet men deze kunnen recupereren (backup & recovery);

- *de performance:* voor heel wat toepassingen is de snelheid belangrijk waarmee men de gegevens kan verwerken (invoeren, wijzigen, verwijderen en vooral terugvinden);

- *de functionaliteit:* het gebruik van de gegevens moet ondersteund worden door een gebruiksvriendelijke interface (SQL-taal, query tools, ...) die een gamma aan functies moet beschikbaar stellen (bv. voor manipulatie van datums, voor genereren van statistieken, ...); via triggers en stored procedures kan de functionaliteit verder uitgebreid worden.

De bovenvermelde kwaliteiten zijn vooral belangrijk voor het dagelijks onderhoud van de operationele gegevens, de klassieke productieomgeving. Ook oudere DBMSen ondersteunen de belangrijkste taken in deze context. Hierbij denken we onder andere aan IMS, IDMS, ... Nadeel van deze oudere systemen is meestal de beperkte functionaliteit. Ze vragen meer programmeerwerk om de informatie in de gewenste vorm beschikbaar te krijgen.

DB2 is, zoals de meeste DBMSen, geëvolueerd. Er zijn verschillende extra taken toegeschoven naar deze systemen. Twee belangrijke zijn:

- *datawarehousing:* het gaat nog steeds over gestructureerde gegevens, maar men gaat ze hoofdzakelijk gebruiken voor analytische toepassingen; doorgaans zijn de bewaarde gegevens stabiel en worden ze vooral aangevuld; de structuren waarin men de gegevens bewaart wijken omwille van het specifieke gebruik meestal af van de structuren in de productieomgeving; kenmerkend is ook het bijhouden van de 'historiek' van heel wat informatie;

- het beheer van *'vreemde' en 'niet-gestructureerde' gegevens:* een DBMS gaat ook toegang verlenen tot gegevens die andere syste-

men beheren: 'federated databases' kunnen transparant informatie beschikbaar stellen die 'collega' systemen beheren; daarbij komen ook systemen in aanmerking die 'niet-gestructureerde' informatie beheren; een DBMS gaat ook zelf 'ongestructureerde' informatie kunnen opnemen in de vorm van Large OBjects (LOB); hierbij is vooral het bewaren van de gegevens in een veilige omgeving de hoofdtaak voor het systeem; de functionaliteit (het gebruik van de gegevens) wordt doorgaans ondersteund door hulpmiddelen die men buiten het DBMS moet zoeken.

Dit artikel (samen met een vervolg later) gaat in op dit beheer van deze 'niet-gestructureerde' informatie: we focussen ons hierbij op wat DB2 te bieden heeft via de DB2 Content Manager (DB2 ContMgr).

Content Management

- Een 'Content Management Systeem' (CMS) definieert men als een systeem dat dient om de inhoud van websites te beheren. Twee basiselementen vindt men terug in een CMS: de 'Content Management Application' (CMA) en de 'Content Delivery Application' (CDA). De CMA geeft de content manager of auteur de mogelijkheid om de inhoud van een website te creëren, te wijzigen of te verwijderen zonder daarbij HTML te moeten kennen. Het is de CDA die de informatie gebruikt en compileert om de website up-to-date te houden. De ingebouwde mogelijkheden van de CMS variëren van product tot product, maar de meeste ondersteunen web-based publicatie, formaatbeheer, indexering, zoek- en selectieactiviteiten.

In een DB2 ContMgr systeem is dit web content management slechts een onderdeel van de mogelijkheden. Een verzameling producten ondersteunt het DB2 content management Sommige zijn vereist voor de werking, andere zijn optioneel.

- *DB2 Document Manager.*

Dit product dient om de volledige levenscyclus van bedrijfsdocumenten te beheren. Via DB2 ContMgr kunnen ze beschikbaar komen op o.a. een externe of interne website.

- *DB2 Records Manager.*

Dit product verzamelt op een centrale plaats registraties van geselecteerde activiteiten. De registraties zelf kunnen door de applicaties of hun ondersteunende systemen lokaal geregistreerd worden. Na het overbrengen naar een centrale repository kan men ze via de DB2 ContMgr beschikbaar stellen.

- *DB2 Common Store for SAP/ DB2 Common Store for Lotus Domino.*

Deze producten kunnen gebruikt worden voor respectievelijk het archiveren van SAP gegevens en voor het archiveren van e-mails en attachments uit een Lotus Notes systeem. Men kan de informatie verder integreren via de DB2 ContMgr.

- *IBM Workplace Web Content Management.*

Dit product is het sluitstuk voor 'web content management'. Het wordt gebruikt om de informatie uit de DB2 ContMgr beschikbaar te maken voor toegang vanuit het internet, intranets, extranets of portal sites.

De architectuur van een DB2 ContMgr systeem

De ganse architectuur berust op een driehoekige structuur (zie figuur 1).

- *Library Server (CMA).*

Dit is het hart van de DB2 ContMgr. Hier wordt alle informatie (metadata) over content bijgehouden. Ook alle controle op toegang tot content gebeurt hier.

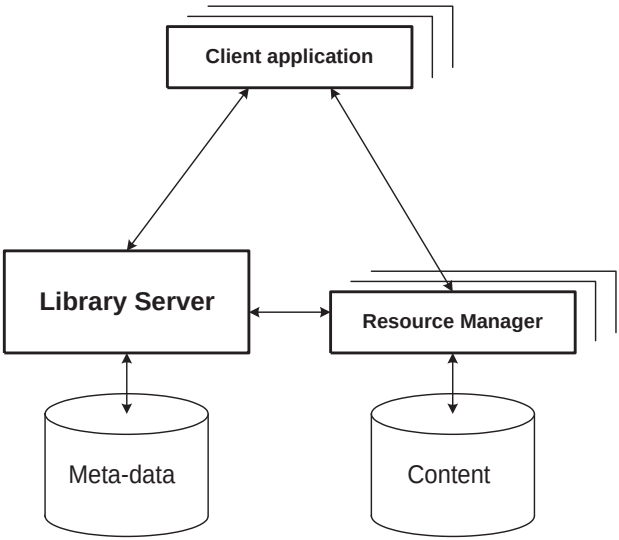
- *Resource Managers (CDA).*

De *Resource Managers* zijn verantwoordelijk voor het fysiek bewaren van content. Hiervoor komen een heel gamma aan systemen in aanmerking. Ze worden gesynchroniseerd via de *Library Server*. Die houdt de informatie over hetgeen *Resource Managers* beheren, up-to-date.

- *Clients (CDA)*

Is de gebruiker van content. Een request om content gaat van de *Client* naar de *Library Server*. De gevraagde informatie zelf wordt meestal geleverd door een betrokken *Resource Manager.*

Figuur 1: architectuur van de DB2 Content Manager



De driehoekige structuur werkt als volgt. De *Client* vraagt een actie i.v.m. content. De vraag gaat eerst naar de *Library Server*. Die gebruikt zijn repository om te controleren of de *Client* recht heeft om de actie uit te voeren. Zo ja, wordt de repository opnieuw gebruikt om eventueel de actie uit te voeren. Een deel van de content kan namelijk beheerd worden door de *Library Server* zelf. Beschikt de *Library Server* niet over alle content die bij de actie betrokken is, dan wordt een referentie (betrokken *Resource Manager* en identificatie van de content) naar de *Client* gestuurd. Deze zal dan rechtstreeks naar de aangeduide *Resource Manager* gaan om de actie daar te laten uitvoeren.

Door de *Client* los te koppelen van de *Resource Managers*, wordt de DB2 ContMgr meer dan een web content manager. De *Resource Managers* zijn slechts een onderdeel van de CDA (content delivery application). Zij leveren de inhoud, terwijl de *Client* verantwoordelijk is voor het gebruik. Dit kan het inbouwen zijn van de informatie in een website, maar voor intern gebruik liggen talloze andere mogelijkheden open.

De Library Server

In dit eerste artikel gaat het om een eerste kennismaking met de DB2 ContMgr. In een vervolgartikel zal meer informatie gegeven worden over onder meer het datamodel dat de *Library Server* gebruikt. Ook workflow-ondersteuning zal meer in detail aan bod komen.

Men beheert de *Library Server* en via deze component de ganse DB2 ContMgr vanuit een 'System Administration Client'. Dit is een Java-based windows toepassing. Deze toepassing ondersteunt onder meer:

- de configuratie van de *Library Server*,
- de configuratie van *Resource Managers*,
- het definiëren en onderhouden van het data model,
- het vastleggen van de beveiliging,
- het beheer van external storage (in DB2 of Oracle),
- het definiëren van workflows,
- het definiëren van zoekfaciliteiten.

Alle relevante informatie wordt door de *Library Server* bewaard in een repository. Dit is een database die de ondersteuning nodig heeft van een DB2 of van een Oracle DBMS. Hoe deze database verder wordt ingevuld met tabellen en indexen hangt nauw samen met het gedefinieerde data model (zie later).

Zoeken naar informatie gebeurt op twee manieren. Ofwel refereert men naar traditionele attributen, ofwel refereert men naar free text attributen. Voor deze tweede vorm kan een extra hulpmiddel worden ingeschakeld: de *DB2 Text Information Extender*.

De Resource Managers en hun Clients

Een *Resource Manager* is in de eerste plaats een DB2 ContMgr component. Het is een toepassing die werkt als een 'WebSphere Application'. Deze applicatie kan onder meer reageren op HTTP-requests die van een *Client* komen. Ook andere communicatieprotocols worden ondersteund.

De informatie die via een DB2 *Resource Manager* beschikbaar komt, kan men opsplitsen in twee soorten:

- de *Resource Manager* beschikt over eigen DB2 databases waarin gestructureerde gegevens, documenten, grafieken, tekeningen, foto's, ... worden bijgehouden,
- de *Resource Manager* kan eveneens beschikken over één of meerdere connectors naar externe *Resource Managers*, zoals bv. Lotus Domino, Microsoft Exchange Server, IMS databases, ...

Naast het beheren en beschikbaar stellen van informatie kan een R*esource Manager* ook meewerken aan een grotere beschikbaarheid van de informatie. Middelen die kunnen ingezet worden: caching van veelgebruikte informatie, replicatie van geografisch verspreide gegevens, ...

Voor het opvragen en eventueel manipuleren van de content worden *Clients* gebruikt:

- de 'eClient' is een browser-based applicatie die vooral het gewone gebruik ondersteunt: zoeken en selecteren van content, navigeren in deze content, importeren en exporteren van content,
- de 'Windows Client' ondersteunt als extra activiteit het werken via workflows.

Voorlopige afronding

Na een eerste oppervlakkige kennismaking met de DB2 Content Manager, zullen we in enkele vervolgartikels dieper ingaan op de belangrijkste componenten van dit systeem. Daarbij zullen we ook de rol van DB2 als ondersteunende DBMS belichten.

CURSUSPLANNING JAN - JUN 2006

DB2 for z/OS, een totaaloverzicht	1825 EUR	23-27/01(W), 06-10/02(L), 27-31/03 (L), 03-07/04 (W), 29/05-02/06 (W) 19-23/06 (L), 24-28/07 (W)
DB2 UDB, een totaaloverzicht	1750 EUR	23-27/01(W), 27-31/03 (L), 29/05-02/06 (W)
RDBMS concepten	350 EUR	23/01(W), 06/02(L), 27/03 (L), 03/04 (W), 29/05(W), 19/06 (L), 24/07 (W)
Basiskennis SQL	350 EUR	24/01(W), 07/02(L), 28/03 (L), 04/04 (W), 30/05(W), 20/06 (L), 25/07 (W)
DB2 for z/OS basiscursus	1125 EUR	25-27/01(W), 08-10/02(L), 29-31/03 (L), 05-07/04 (W), 31/05-02/06 (W), 21-23/06 (L),
DB2 UDB basiscursus	1050 EUR	25-27/01(W), 29-31/03 (L), 31/05-02/06 (W)
SQL workshop	750 EUR	13-14/02 (W), 27-28/02 (L), 18-19/04 (L), 02-03/05 (W)
Extended SQL in DB2	425 EUR	15/02 (W), 20/04 (L)
DB2 for z/OS programmering voor gevorderden	800 EUR	16-17/02 (W), 10-11/04 (L)
DB2 for OS/390: SQL performance	1275 EUR	08-10/03 (L), 22-24/05 (W)
XML in DB2	425 EUR	23/02 (W), 21/06 (L)
DB2 for z/OS database administratie	1700 EUR	20-23/03 (W), 12-15/06 (L)
DB2 for z/OS operations and recovery	1425 EUR	22-24/03 (W), 14-16/06 (L)
DB2 for z/OS DBA and operations	2175 EUR	20-24/03 (W), 12-15/06 (L)
DB2 for z/OS in een Java omgeving	425 EUR	04/04 (W), 27/06 (L)

Plaats: L = Leuven; W = Woerden; details en extra cursussen: www.abis.be

Postbus 220
Diestsevest 32
BE-3000 Leuven
Tel. 016/245610
Fax 016/245691
training@abis.be

abis
TRAINING & CONSULTING

Postbus 122
Pelmolenlaan 1-K
NL-3440 AC Woerden
Tel. 0348-435570
Fax 0348-432493
training@abis.be