



OPEN CURSOR

Het is een traditioneel verschijnsel bij elke nieuwe release van software: een rits uitdagende features worden door de meeste bedrijven slechts bij mondjesmaat geïmplementeerd. Het zijn features, die de traditionele applicatieontwikkeling grondig wijzigen, of die juist gericht zijn op de nieuwe ontwikkelparadigma's.

Wat zijn hiervan de oorzaken? Onbekendheid met de nieuwe technologie is er één; gewoontevorming een andere.

Aan de hand van 'Exploring DB2' doet ABIS een poging hier verandering in te brengen - door de relevante features voor DBA's en applicatieontwikkelaars op duidelijke manier in de schijnwerpers te plaatsen. In dit nummer bespreken we een onderwerp dat volledig in het teken staat van SQL performance - niet te missen!

Het ABIS DB2 team.

IN DIT NUMMER:

- Een uitgebreide studie van een nieuwe techniek om de *performance van SQL statements* te evalueren: de DSN_STATEMNT_TABLE techniek.
- *Dossier 7*: zie de catalog groeien - bewaar een historiek van uw optimizer statistieken.
- *Een oproep!*
- *DB2 Text Extender*: u vindt wat u zoekt ..., of hoe u DB2 kunt gebruiken als een tekst - zoekmachine!
- *Cursusplanning dec 2002 - feb 2003.*

CLOSE CURSOR

In het vierde nummer van Exploring DB2 staan we stil bij één van de meest uitdagende vernieuwingen waarmee DB2 vandaag wordt geconfronteerd: XML! En hebt u al gehoord van 'Federated Databases'?

Tot dan!

Hulpmiddel bij optimalisatie: DSN_STATEMNT_TABLE

Eric Venmans (ABIS)

Inleiding

Wanneer een SQL-statement geschreven wordt, is er een eerste, al-berbelangrijkste vraag: is het statement correct geformuleerd? Met andere woorden, doet DB2 precies wat er verwacht wordt, levert een SQL-vraag de juiste gegevens? Het is niet meer of niet minder dan een evaluatie van de functionele 'correctheid' van een SQL statement.

Een tweede vraag die daar onmiddellijk bijhoort, is: kan het systeem snel genoeg de gevraagde actie uitvoeren? Dit is een vraag naar de performance, van een SQL-statement.

De eerste vraag kan relatief eenvoudig beantwoord worden in een test-omgeving. Als we beschikken over (kleine) tabellen met een representatieve set aan gegevens, lukt dit wel.

De tweede vraag is moeilijker te beantwoorden. Het uitvoeren in een test-omgeving geeft allicht een totaal vertekend beeld: te kleine structuren en geen concurrency van andere gebruikers. Nochtans is een antwoord op de tweede vraag soms even belangrijk als een antwoord op de eerste. Wat kan een gebruiker doen met een juist antwoord, als het hopeloos te laat komt? Misschien haken potentiële klanten af, nog voor ze goed en wel gecontacteerd zijn.

Hoewel de tweede vraag dus moeilijker te beantwoorden is, kunnen we via kostenschattingen toch een benaderend antwoord krijgen. In deze context beschikken we sinds versie 6 van DB2 UDB for OS/390, over een interessant hulpmiddel: de DSN_STATEMNT_TABLE.

Algemene voorwaarden

- Een eerste voorwaarde om een relevante kostenschatting te bekomen, is het verkrijgen van een betrouwbaar toegangspad (access path) van DB2. Hiervoor hebben we een omgeving nodig met realistische statistieken. We kunnen in een test-omgeving een productie-omgeving simuleren. Ofwel kopiëren we statistieken uit een bestaande productie-omgeving, ofwel brengen we eigenhandig statistieken aan, gebaseerd op het resultaat van analyse-activiteiten. Voor een gegeven SQL-statement krijgen we dan via de EXPLAIN-functie een beschrijving van het toegangspad, zoals het door de DB2-optimizer wordt gekozen op basis van de beschikbare, gesimuleerde statistieken.

- Een tweede voorwaarde om een kostenschatting te krijgen van DB2, is het creëren van een tabel waarin het systeem zijn resultaten

kan publiceren: de DSN_STATEMNT_TABLE. De betrokken gegevens worden ingevuld tijdens het uitvoeren van de EXPLAIN-functie.

Geleverde informatie

De DSN_STATEMNT_TABLE heeft een door IBM voorgeschreven structuur.

Naast de identificatie van de query, waarvoor kosteninformatie wordt gegeven zijn de volgende kolommen relevant:

- de geschatte kosten (PROCMS, PROCSU),
- een aanduiding van de kostencategorie (COST_CATEGORY),
- eventueel een uitleg die hoort bij de kostencategorie (REASON).

Tabelstructuur? zie [Bijlage](#) op p. 13

De geschatte kosten worden uitgedrukt in 'service units' en in 'milliseconden'. De kostencategorie is 'A' indien alle nodige informatie beschikbaar is voor het systeem om een relatief nauwkeurige kostenberekening te doen.

Kostencategorie 'B' duidt op een 'twijfelachtig' resultaat. De redenen worden aangegeven en hebben voor selects o.a. te maken met:

- gebruik van variabelen (zeer courant voor static SQL),
- onvolledige of ontbrekende statistieken.

Voor het wijzigen van gegevens ligt de oorzaak bij o.a. volgende, niet-berekende kosten:

- controle van referential integrity,
- uitvoeren van user-defined functions,
- activeren en uitvoeren van getriggerde acties.

Mogelijkheden

▪ De kostenberekeningen van DB2 kunnen gebruikt worden om varianten van SQL statements te vergelijken met behulp van het systeem. Varianten krijgen we bijvoorbeeld wanneer informatie uit meerdere tabellen moet gecombineerd worden. Doorgaans kunnen we kiezen uit verschillende formuleringen: een 'join', een 'gecorrleerde subquery', een 'niet-gecorrleerde subquery', ... Deze varianten hebben meestal een verschillend toegangspad en bijgevolg verschillende kosten. Door het vergelijken van objectiveerbare kostindicaties, als door DB2 gegenereerd, komt men sneller tot de keuze van het 'beter' geformuleerde SQL statement (zie voorbeeld 1).

▪ De kosteninformatie kan ook verschillen in toegangspaden blootleggen die niet meteen zichtbaar zijn in de PLAN_TABLE. In voorbeeld 2 is het verschil tussen beide varianten van de opgenomen SQL te vinden in de vorm van de voorwaarde. In queryno 201 hebben we te maken met stage-1 voorwaarden. Zowel de 'ORDER BY' (queryno 202) als de voorwaarde kunnen op basis van indexgegevens worden uitgewerkt. In queryno 202 is de voorwaarde van het

stage-2 type. De indexinformatie wordt enkel gebruikt om rijen in de gewenste volgorde te krijgen. Het controleren van de voorwaarde gebeurt pas na het binnenhalen van alle rijen in de buffers.

Voorbeeld 1

```

Explain plan set queryno = 101 for
  SELECT S.Id, S.Name, S.Address
     FROM Suppliers S
     WHERE S.Name like ?
        and exists (SELECT 'exists'
                   FROM   Products P
                   WHERE  P.SupplierId = S.Id)

```

```

Explain plan set queryno = 102 for
  SELECT distinct S.Id, S.Name, S.Address
     FROM Suppliers S, Products P
     WHERE P.SupplierId = S.Id
        and S.Name like ?

```

<u>QNo</u>	<u>SqlType</u>	<u>Category</u>	<u>MSec</u>	<u>SUnits</u>	<u>Reason</u>
101	SELECT	B	123	254	HOST VARIABLES
102	SELECT	B	788	1625	HOST VARIABLES

Voorbeeld 2

```

Explain plan set queryno = 201 for
  SELECT P.Class, P.Part, P.Name
     FROM Products P
     WHERE P.Class = ? and P.Part > ?
        or P.Class > ?
     ORDER BY P.Class, P.Part
     OPTIMIZE for 1 row

```

```

Explain plan set queryno = 202 for
  SELECT P.Class, P.Part, P.Name
     FROM Products P
     WHERE P.Class||P.Part > ?
     ORDER BY P.Class, P.Part
     OPTIMIZE for 1 row

```

<u>QNo</u>	<u>Method</u>	<u>TableName</u>	<u>Access</u>	<u>MatchCols</u>	<u>IndexName</u>	<u>IndexOnly...</u>
201	0	PRODUCTS	I	0	IXPRID	N
202	0	PRODUCTS	I	0	IXPRID	N

...	<u>Sort N</u>	<u>Sort C</u>	<u>Prefetch</u>
	NNNN	NNNN	
	NNNN	NNNN	

<u>QNo</u>	<u>SqlType</u>	<u>Category</u>	<u>MSec</u>	<u>SUnits</u>	<u>Reason</u>
201	SELECT	B	6094	12565	HOST VARIABLES
202	SELECT	A	36739	75751	

- Ook de absolute waarde van de kostencalculatie kan een aanwijding zijn voor de te verwachten performance of gebrek daaraan. Het kan ons aanzetten om reeds in een vroeg stadium te zoeken naar alternatieve oplossingen.

Als deze oplossingen in een test-omgeving kunnen gesimuleerd worden (andere clustering, kleinere subtype tabel, ...), geeft een kostencalculatie reeds vooraf een indicatie over het effect van de ingreep. Eventueel kan de ingreep 'vergeten' worden, als het verhoopte effect minimaal is.

Beperkingen

Uiteraard is de DB2 kostencalculatie geen 'wondermiddel'. De belangrijkste beperkingen zijn:

- er is enkel informatie over toegangspaden die door DB2 worden gekozen,
- de gegeven tijd is enkel processor-tijd (dus geen aanduiding voor de totale, te verwachten doorlooptijd van een query),
- dikwijls is de kostencategorie 'B'.

Dit laatste is vooral voor 'embedded static SQL' een probleem. Variabelen worden ingevuld nadat het toegangspad reeds gekozen is tijdens het BIND-proces. De keuze is gebeurd op basis van 'default' filter-factors. Een filter-factor is een maat voor de selectiviteit van een voorwaarde in een SQL-statement. De 'default' waarden zijn nauw verwant met wat 'statistisch' te verwachten valt. Ze kunnen echter sterk afwijken van de reële waarden. Vooral bij 'range'-predicaten komt dit vaak voor: de gebruiker van een applicatie zal uit ervaring dikwijls waarden kiezen waarbij de hoeveelheid informatie beperkt en overzichtelijk blijft. De reële filter-factor is in dit geval selectiever dan de 'default'. Dit heeft als gevolg dat de DB2 kostenberekeningen negatiever uitvallen dan de reële kosten.

Waarschuwing

De filter-factoren zullen niet steeds dezelfde rol spelen in varianten van een gegeven SQL-vraag. Dit kan een 'foute' veronderstelling van het systeem (afwijkende default-waarden t.o.v. reële waarden) tot gevolg hebben; men krijgt een vertekend beeld van wat 'echt' de goedkoopste variant is. Een kritische analyse van de EXPLAIN informatie (toegangspaden en kostencalculaties) is nodig om een correcte conclusie te trekken.

In een later artikel zullen enkele sprekende voorbeelden gegeven worden van zulke situaties. Bovendien zal er aangegeven worden hoe we de kostencalculatie kunnen 'bijsturen'.

Besluit

Werken aan performance gebeurt op verschillende niveaus. Als het gaat om het vergelijken van toegangspaden die we krijgen voor verschillende varianten van eenzelfde SQL-vraag, kan de kostencalculatie die DB2 publiceert in de DSN_STATENT_TABLE, een bruikbaar hulpmiddel zijn. We moeten echter voorzichtig zijn met het interpreteren van de gegevens. Een grondige kennis van filter-factoren (default en reële) is hierbij een onmisbaar element.

OPROEP!

- uw artikel hier -

'Exploring DB2' wordt uitgegeven door ABIS voor DB2 specialisten. Wij doen er alles aan om de kwaliteit van wat in deze publicatie verschijnt op het hoogst mogelijke niveau te houden.

Deze artikels worden in eerste plaats geschreven door DB2 specialisten - niet door professionele auteurs. Graag hadden wij dan ook van u - een collega DB2 specialist - input ontvangen:

Zijn er topics die u graag behandeld had willen zien in 'Exploring DB2'?

Zijn er specifieke vragen en problemen die wij voor u kunnen behandelen in een volgend artikel?

Hebt u bepaalde problemen ondervonden, of beter nog, oplossingen gevonden voor bepaalde problemen?

Hebt u interesse om een artikel bij te dragen - u zal zien dat een artikel samenstellen veel minder inspannend is dan u in eerste instantie denkt. Een kleine honderd woorden zijn meestal voldoende om een probleem en oplossing op papier te zetten.

Neem contact met ons op via training@abis.be en wij doen de rest.

Het ABIS DB2 team.

DOSSIER 7

De catalog groeit! - Deel 2 (over de geschiedenis van uw database)

In een vorige editie bespraken we reeds een tweetal nieuwe catalog tabellen. Deze keer kozen we voor de history tabellen.

Een aparte categorie van tabellen in de catalog zijn deze die statistieken bevatten. Apart omdat ze toch op een volledig andere manier gemanipuleerd worden, denk alsmaar aan hoe de gegevens erin komen. Ze zijn ook van cruciaal belang, de kwaliteit van de statistische gegevens bepaalt in belangrijke mate de efficiëntie van de dataophaling. De statistische gegevens moeten ook up to date zijn, op regelmatige basis worden ze met het RUNSTATS utility aangepast. Maar wie zou durven beweren dat de overschreven versie geen interessante informatie bevat - informatie die we elke keer weer kwijt spelen.

Alle tabellen die statistieken bevatten of tabellen met kolommen die statistieken bevatten hebben in versie 7 van DB2 UDB voor OS/390 een tweelingbroertje gekregen, weliswaar niet identiek want in deze tabellen komt de oude RUNSTATS data terecht. Er zijn 9 history tabellen:

SYSIBM.SYSCOLDIST_HIST,
SYSIBM.SYSCOLUMNS_HIST,
SYSIBM.SYSINDEXES_HIST,
SYSIBM.SYSINDEXPART_HIST,
SYSIBM.SYSINDEXSTATS_HIST,
SYSIBM.SYSLOBSTATS_HIST,
SYSIBM.SYSTABLEPART_HIST,
SYSIBM.SYSTABLES_HIST,
SYSIBM.SYSTABSTATS_HIST.

Alleen de kolommen die statistieken bevatten komen voor in de history tabellen.

Deze tabellen worden opgevuld met data uit de overeenkomstige catalog tabel wanneer het RUNSTATS utility wordt uitgevoerd met history options - een bijkomende optie die werd toegevoegd aan de RUNSTATS control statement set. Het is ook mogelijk history records toe te voegen wanneer statistieken aangemaakt worden tijdens het uitvoeren van het LOAD, REORG en REBUILD INDEX utility. Maar wanneer de statistieken in de catalog tabellen manueel geupdate worden, worden de history tabellen niet aangepast. Met het MODIFY STATISTICS utility tenslotte kunnen oude statistieken die niet meet nuttig zijn verwijderd worden uit de history tabellen.

De mogelijkheden die DB2 hiermee biedt zijn interessant:

- om de groei van uw database te kunnen opvolgen en inschatten;
- om een performance val na het draaien van een runstats tegen te kunnen gaan;
- om een 'fallback' mogelijkheid te hebben na het verkeerdelijk manueel editeren van statistieken.

Bron: www.ibm.com/db2

Katrien Platteborze (ABIS)

DB2 Text Extender: u vindt wat u zoekt...

Kris Van Thillo (ABIS)

Inleiding

Traditiegetrouw wordt DB2 gebruikt voor het opslaan van gestructureerde data: getallen, strings, datum waarden. Wat de meeste DBA's en ontwikkelaars echter geneigd zijn te vergeten, is dat DB2 reeds zeer lang het zogenaamde LONG VARCHAR datatype ondersteunt. Dit datatype biedt de mogelijkheid om ongestructureerde tekstdata op te slaan - weliswaar beperkt in omvang. Ook werken met dit datatype is aan een aantal beperkingen onderworpen. Ik ben er van overtuigd dat u, net zoals ik, door illustere voorgangers werd aangemaand dit datatype 'nooit' te gebruiken.

En dan nieuwe kreten: 'Text Management', 'Content Management'. En dus moet DB2 de mogelijkheid krijgen - en bieden - om zowel grote hoeveelheden gestructureerde als ongestructureerde data te verwerken. We willen deze data rubriceren, indexeren en we willen ze op een efficiënte manier kunnen doorzoeken.

Dankzij de 'DB2 Text Extender' en zijn onderliggende database technologieën (UDFs, UDTs, triggers, stored procedures) is DB2 op weg een volwaardig tekst management systeem te worden. Heeft IBM trouwens niet aangekondigd dat de volgende releases van Lotus Notes gebruik gaan maken van UDB als database systeem?

Architectuur

DB2 Text Extender heeft 2 belangrijke onderdelen.

Een server, de eigenlijke tekstmachine, is verantwoordelijk voor de tekst index, tekst beheer, tekst zoeken etc. Deze tekst server is volledig geïntegreerd met en geïnstalleerd naast de traditionele DB2 database. Het is in deze database dat de gebruikersdefinities, triggers etc. gecreëerd worden. Het is ook vanuit deze database dat de tekst features van de extender zullen worden gebruikt.

Het tweede onderdeel - de text extender client - voorziet in een interface: een commandolijn en een aantal tekst 'utilities' voor beheer en onderhoud. Tevens is een volledig functionele voorbeeldapplicatie voorhanden.

Vooraf

Voorafgaand aan de ontwikkeling van een applicatie met uitgebreide zoekfunctionaliteit in DB2, moeten we bij het volgende stilstaan.

- De structuur van de te indexeren en te doorzoeken tekst. Is het gestructureerde tekst, zoals HTML of XML, of juist niet, zoals rtf, ms-

word of platte tekst. Als het gaat om gestructureerde tekst, wil men dan het zoeken beperken op basis van die structuur?

- Hoe wil men de tekst doorzoeken? Zoeken we woorden, strings of karakters? Zoeken we tevens afgeleide woorden, of synoniemen? Hebben we interesse voor logische relaties tussen woorden: hebben we bijvoorbeeld enkel interesse als een woord voorkomt samen met een ander woord in dezelfde zin of paragraaf?

- De dynamiek van de tekstapplicatie: hoe frequent worden teksten toegevoegd? Moeten deze teksten onmiddellijk via de zoekmachine terug te vinden zijn? Worden reeds geïndexeerde teksten gewijzigd?

Text index

Een DB2 Text Extender 'text index' is een directory structuur op de tekst server. Deze bevat een lijst van alle significante termen aanwezig in het geïndexeerde document, inclusief een verwijzing naar de documenten die deze termen bevatten. Onbelangrijke termen worden weggefilterd door het gebruik van stopwoordlijsten.

Misschien een merkwaardige observatie: tekst indexen worden niet aangemaakt ('create index') maar geactiveerd ('enabled'). Dit laatste creëert niet alleen de index, maar tevens een aantal verborgen, ondersteunende structuren (tabellen, triggers, etc).

DB2 Text Extender ondersteunt momenteel *drie verschillende tekst indexen* - 'linguistic', 'precise', en 'Ngram'. In de toegevoegde tekstvakjes worden hun belangrijkste karakteristieken uiteengezet.

Index types

Definitie	Mogelijkheden
Linguistic Biedt de mogelijkheid semantisch te zoeken. Te indexeren woorden worden omgevormd naar een basisvorm en opgeslagen. Zoektermen worden eveneens vertaald. Ook met synoniemen en samenstellingen wordt rekening gehouden. Een dictionary is vereist.	<ul style="list-style-type: none">◦ woord/zin herkenning◦ stopwoord filters◦ reductie naar basisvorm◦ woordontleding◦ geen 'case' herkenning
Precise De zoekterm wordt in de index opgenomen zoals deze in het document voorkomt. Woorden worden niet gereduceerd tot hun basisvorm. Een dictionary is vereist.	<ul style="list-style-type: none">◦ woord/zin herkenning◦ stopwoord filters◦ 'case' herkenning
Ngram Karakter gebaseerde. Spaties, karaktervariaties, tot spelfouten toe kunnen worden opgezocht.	<ul style="list-style-type: none">◦ karakter 'as is' zoeken

Het indexeerproces zelf beslaat twee stappen. In eerste instantie wordt naar alle documenten die moeten worden geïndexeerd (gewijzigd en/of toegevoegd) verwezen in een logfile (log tabel). Deze logtabel wordt onderhouden aan de hand van triggers aangemaakt op de tabel waarin de documenten worden opgeslagen. In een tweede stap - de echte indexeerstap - worden alle documenten aangeduid in de logtabel verwerkt. Hierdoor wordt synchronisatie tussen de documenten en de tekst index bewerkstelligd.

Bij de definitie van een tekst index, moeten een aantal indexeigenschappen worden gespecificeerd.

- De eigenschappen van de te indexeren tekst, inclusief tekst formaat (e.g. XML), taal (e.g. Duits), codepagina, en karakterdefinities.
- De karakteristieken van de index, inclusief de index 'locatie directory', het index type (bijv. Ngram) en de index synchronisatie frequentie (bijv. tijdstip, aantal documenten in de log).
- De eigenschappen van index processen, inclusief de initiële opbouw van de index, en de 'commit count'. Deze mogelijkheden staan de ontwikkelaar/DBA toe om te specificeren wanneer de structuur van de index aanvankelijk moet worden gevuld en de veelvuldigheid waarmee 'commits' moeten worden aangevraagd.

Een concrete uitwerking? Zie [Bijlage](#) op p. 13.

SQL syntax

Eens de index is ontworpen, kan deze gebruikt worden voor het uitvoeren van complexe zoekopdrachten. Hiertoe werd een uitgebreide SQL syntax uitgewerkt. Uitgebreid, in de zin dat de syntax nog niet formeel is gestandaardiseerd: het is nog geen ANSI SQL.

Ongeveer 13 verschillende UDFs staan ter beschikking: de drie meest voorkomende functies worden kort toegelicht. Merk op dat de relevantie van bepaalde functies bepaald wordt door het gekozen index type!

- De 'contains' functie, die de mogelijkheid biedt een tekstkolom te doorzoeken naar een specifieke tekststring. Een identificatie van de tekst waarin de string werd gevonden wordt teruggegeven.
- De 'no_of_matches' functie, die aangeeft hoeveel maal een tekststring voorkomt in de geïndexeerde documenten.

Voorbeeld 1: contains

```
"select id from st_content
where db2tx.contains(hcontent, '\"tekst\"')=1"
```

```
ID
-----
  1
  2
```

- De 'rank' functie, die middels een absolute waarde aangeeft hoe goed het document overeenkomt met de zoekcriteria in relatie tot andere gevonden documenten. De waarde geeft het aantal over-

eenkomsten aan die werden gevonden in het document in relatie tot de grootte van het document.

Bovendien ondersteunen de meeste functies bijkomende zoekfuncties:

- string concatenatie;
- booleaanse operatoren 'and' en 'or';
- gebruik van 'wildcards';
- zoeken op speciale karakter tekens;
- zoeken naar tekst in dezelfde paragraaf of dezelfde zin.

Voorbeeld 2: contains

```
"select id from st_content
where db2tx.contains(hcontent, '\"tekst\"' IN SAME SENTENCE AS
\"lange\"')=1"
```

```
ID
-----
                2
```

Aanvullende kenmerken

DB2 UDB Text Extender biedt verder een aantal krachtige aanvullende opties.

- Werken met gestructureerde documenten.

Een gestructureerd document is een document dat secties bevat (HTML document, XML document), bijvoorbeeld document titel, auteur en onderwerp. Men kan het zoekbereik limiteren tot een bepaalde sectie van de documenten.

- Werken met externe files.

Text Extender kan niet alleen zoeken in tekst opgeslagen in DB2 UDB tabellen, maar tevens in elders opgeslagen tekstfiles (bv. in operating system files). En dus kan ook de inhoud van de externe files worden geïndexeerd en doorzocht.

Besluit

DB2 Text Extender, alhoewel relatief nieuw, biedt een aantal uitdagende mogelijkheden die het zoeken door teksten mogelijk maakt. De integratie met traditionele, gestructureerde data, biedt een veelde aan bijkomende opportuniteiten voor de traditionele applicatieontwikkelaar.

Een integratie met de DB2 XML extender biedt daarnaast ook de mogelijkheid om op een gestructureerde manier XML files te doorzoeken. Meer hierover in een volgend nummer.

CURSUSPLANNING DEC 2002 - FEB 2003

DB2 for OS/390, een totaaloverzicht	1625 EUR	09-13/12 (L), 13-17/01 (W), 03-07/02 (L), 17-21/02 (W)
DB2 UDB, een totaaloverzicht	1625 EUR	17-21/02 (W), 24-27/02 (L)
RDBMS concepten	325 EUR	09/12 (L), 13/01 (W), 03/02 (L)
Basiskennis SQL	325 EUR	10/12 (L), 14/01 (W), 04/02 (L), 18/02 (W)
DB2 for OS/390 basiscursus	975 EUR	11-13/12 (L), 15-17/01 (W), 05-07/02 (L), 19-21/02 (W)
DB2 UDB basiscursus	975 EUR	27-29/11 (W), 19-21/02 (W)
DB2 UDB concepten	375 EUR	03/12 (L), 07/03 (W)
SQL workshop	700 EUR	16-17/12 (L), 27-28/01 (W), 24-25/02 (L)
DB2 applicatieprogrammering voor gevorderden	1050 EUR	04-06/12 (L), 02-04/04 (W)
DB2 for OS/390: SQL performance	1200 EUR	16-18/12 (L), 03-05/03 (W)
Fysiek ontwerp van relationele databases	700 EUR	11-12/02 (W), 28-29/04 (L)
DB2 database administratie	1600 EUR	18-21/03 (W), 12-15/05 (L)
DB2 UDB database administratie	1600 EUR	17-20/03 (W), 05-08/05 (L)
DB2 UDB systeembeheer en per- formance	400 EUR	25/11 (L), 21/03 (W), 09/05 (L)
DB2 UDB for OS/390 V7 upgrade voor ontwikkelaars	375 EUR	05/12 (W), 28/02 (L)
DB2 UDB en zijn extenders: XML en text search	200 EUR	16/12 (L), 28/03 (W)
DB2 UDB integratie met MQSeries	200 EUR	16/12 (L), 28/03 (W)

Integrating IBM technologies -
Web Services 800 EUR 12-13/12 (W)

We presenteren dit najaar een vervolg op 'Integrating IBM technologies'. Toegesplitst op Web Services, specifiek in een IBM omgeving met o.a. WebSphere en DB2 UDB.

Plaats: L = Leuven; W = Woerden

Details, andere data en bijkomende cursussen: www.abis.be

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



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

Bijlage

Tabelstructuur DSN_STATEMNT_TABLE

```
Create table DSN_STATEMNT_TABLE
(
QUERYNO           INTEGER           NOT NULL WITH DEFAULT,
APPLNAME          CHAR(8)           NOT NULL WITH DEFAULT,
PROGNAME          CHAR(8)           NOT NULL WITH DEFAULT,
COLLID            CHAR(18)          NOT NULL WITH DEFAULT,
GROUP_MEMBER      CHAR(8)           NOT NULL WITH DEFAULT,
EXPLAIN_TIME      TIMESTAMP         NOT NULL WITH DEFAULT,
STMT_TYPE         CHAR(6)           NOT NULL WITH DEFAULT,
COST_CATEGORY     CHAR(1)           NOT NULL WITH DEFAULT,
PROCMS            INTEGER           NOT NULL WITH DEFAULT,
PROCSU            INTEGER           NOT NULL WITH DEFAULT,
REASON            VARCHAR(254)      NOT NULL WITH DEFAULT)
in TBU00X.TSYSCOST
```

Voorbeeld Text Extender

```
rem Step1 -----
rem Creating the text extender instance
rem Creating the sample table
rem A default database is assumed (DB2DBDFT)

db2 create table st_content (content varchar(200), id integer);
txicrt eb4e

rem Step2 -----
rem Is the setup OK?
rem Check current configuration.

db2tx get environment

Current database manager instance          (DB2INSTANCE) = eb4e
Default database name                    (DB2DBDFT) = docmgt
Instance name                             (DB2TX_INSTOWNER) = DB2
Instance directory                        (DB2TX_INSTOWNERHOMEDIR) = d:\dmb

rem Step3 -----
rem Enable database for text extender access
rem Internal objects (UDF, UDT, etc) are created

db2tx enable database
db2tx get text configuration > stepca.txt
db2tx change text cfg using INDEXTYPE precise UPDATEFREQ min(4)
d(0,1,2,3,4,5,6) h(9,11,13,15,17) m(00)
db2tx get text configuration > stepcb.txt
```

content stepca.txt:

```
Database Connection Information
Database product           = DB2/NT 07.02.0001
SQL authorization ID      = TB00986
Local database alias      = DOCMGT

Coded character set ID    (CCSID) = 1252
Language                  (LANGUAGE) = US_ENGLISH
Format                    (FORMAT) = TDS
Index type                (INDEXTYPE) = NGRAM
Update frequency          (UPDATEFREQ) = NONE
Index directory           (DIRECTORY) = d:\INSTANCE\DB2\DB2TX\INDEXES
Update index option       (UPDATEINDEX) = UPDATE
Commit count              (COMMITCOUNT) = 0
Table space               (TABLESPACE) =
```

content stepcb.txt:

```
Database Connection Information
Database product           = DB2/NT 07.02.0001
SQL authorization ID      = TB00986
Local database alias      = DOCMGT

Coded character set ID    (CCSID) = 1252
Language                  (LANGUAGE) = US_ENGLISH
Format                    (FORMAT) = TDS
Index type                (INDEXTYPE) = PRECISE
Update frequency          (UPDATEFREQ) =
MIN(4)D(0,1,2,3,4,5,6)H(9,11,13,15,17)M(00)
Index directory           (DIRECTORY) = d:\INSTANCE\DB2\DB2TX\INDEXES
Update index option       (UPDATEINDEX) = UPDATE
Commit count              (COMMITCOUNT) = 0
Table space               (TABLESPACE) =
```

```
rem Step4 -----
rem Enable a text column
rem Internal, index related objects are created:
rem   an index column is created
rem   a log table is created
rem   a 'text index' column is added to the base table
rem   a trigger to populate the log table is added
```

db2 describe table st_content > stepd1.txt

db2tx enable text column st_content content handle hcontent

db2 describe table st_content > stepd2.txt

content stepd1.txt:

Column name	Type schema	Type name	Length	Scale	Nulls
CONTENT	SYSIBM	VARCHAR	200	0	Yes
ID	SYSIBM	INTEGER	4	0	Yes

2 record(s) selected.

content stepd2.txt:

Column name	Type schema	Type name	Length	Scale	Nulls
CONTENT	SYSIBM	VARCHAR	200	0	Yes
ID	SYSIBM	INTEGER	4	0	Yes
HCONTENT	DB2TX	DB2TEXTH	0	0	Yes

Rem Step 5 -----
Rem
db2tx GET INDEX STATUS st_content handle hcontent

Database Connection Information

Database product = DB2/NT 07.02.0001
SQL authorization ID = TB00986
Local database alias = DOCMGT

Node 0

Search status = Search available
Update status = Update available
Reorganization status = Reorganization available
Scheduled documents = 0
Indexed documents = 0
Primary index documents = 0
Secondary index documents = 0
Error events = No error events.

Rem 6 -----
rem
rem Insert van nieuwe text documenten
rem Step 6

db2 connect to event

db2 insert into st_content (content, id)
values('Dit is een lange tekst.', 1)
db2 insert into st_content (content, id)
values('Dit is een nog langere tekst', 2)

db2 commit

db2tx GET INDEX STATUS st_content handle hcontent

Database Connection Information

Database product = DB2/NT 07.02.0001
SQL authorization ID = TB00986
Local database alias = DOCMGT

Node 0

Search status = Search available
Update status = Update available
Reorganization status = Reorganization available
Scheduled documents = 0
Indexed documents = 0
Primary index documents = 0
Secondary index documents = 0
Error events = No error events.

db2tx UPDATE INDEX st_content handle hcontent

db2tx GET INDEX STATUS st_content handle hcontent

Database Connection Information

Database product = DB2/NT 07.02.0001
SQL authorization ID = TB00986
Local database alias = DOCMGT

Node 0

Search status = Search available
Update status = Update available
Reorganization status = Reorganization available
Scheduled documents = 0
Indexed documents = 2
Primary index documents = 2
Secondary index documents = 0
Error events = No error events.