



### OPEN CURSOR

*De zomervakantie is voor de meesten onder ons voorbij, de batterijen weer opgeladen, klaar voor nieuwe uitdagingen!*

*Ook ABIS medewerkers hebben genoten van de zomervakantie. Maar we hebben ook een reeks nieuwe Exploring DB2s voor u klaargestoomd; met artikels die u naar we hopen ook deze jaargang weer zullen boeien: maximaal gestoeld op concrete ervaringen opgedaan in praktijk situaties.*

*En we hebben ook voor de eerste maal gewerkt aan een 'Exploring DB2 - live' event: een 3-daagse technisch symposium waar we de nieuwe en uitdagende features van DB2 op een overzichtelijke manier voor u uiteenzetten. Zoals u gewoon bent van de papieren versie, maar nu 'live', met demo's en oefeningen.*

*U merkt het - wij zijn er weer klaar voor!*

*Het ABIS DB2-team.*

### IN DIT NUMMER:

- *DB2 Information Integrator*: over het integreren van informatie in DB2 UDB versie 8.
- *Dossier 8* handelt over tabel-gecontroleerde partiënering.
- En nog meer over integratie, nu met MQSeries, in *DB2 en MQSeries - Integratie*.
- *Cursusplanning september 2004 - december 2004*.



### CLOSE CURSOR

Utilities worden vaak gepositioneerd binnen het exclusieve werkgebied van de DBA. Ze stelden hem in staat onderhoud, beheer, beschikbaarheid van een DB2 voor z/OS systeem te realiseren. Echter, steeds vaker worden ze ook gebruikt als een aanvaardbaar alternatief voor vaak complexe SQL processing. Een nieuwe reeks start in volgend nummer. Tot dan!

# DB2 Information Integrator

*Diane Hendrix (ABIS)*

De laatste jaren is de behoefte aan informatieintegratie sterk toegenomen. Vele bedrijven worden immers meer en meer geconfronteerd met gegevens afkomstig van zeer diverse hardware en operating system platformen. Data opgeslagen in relationele en/of niet-relationele database systemen moeten gecombineerd worden met niet- of semigestructureerde data (XML, flat files, ...). Er is dan ook een snel uitbreidende categorie van nieuwe technologieën ontstaan, die allemaal één gemeenschappelijk streefdoel hebben: één universele toegang voorzien tot diverse en gedistribueerde informatiebronnen. Elke gebruiker moet in staat zijn via één API (Application Programming Interface), een grote diversiteit van data te benaderen alsof het zou gaan over 1 enkele bron van informatie. De gekozen informatie-integratietechniek moet er natuurlijk ook voor zorgen, dat de gebruiker afgeschermd wordt van de complexiteit geassocieerd met elke datalokatie op zich (semantiek, dataformaat, toegangsmethodes, ...).

Informatieintegratie is altijd gebaseerd op één (of een combinatie) van de volgende twee strategieën:

- datafederatie: de gegevens blijven gedistribueerd aanwezig, maar worden toegankelijk via 1 universeel toegangspunt. De vraag wordt als het ware naar de gegevens gebracht.
- dataconsolidatie: de gedistribueerde data worden gerepliceerd naar een lokatie die het meest efficiënt of consistent is voor de applicatie/gebruiker. In dit geval worden de gegevens naar de vraag gebracht.

Het data warehouse concept is een typisch voorbeeld van dataconsolidatie: de gegevens worden geëxtraheerd vanuit verschillende operationele systemen en opgeladen in een data warehouse. Dit proces vereist een zeer uitgebreide analyse en modellering van de dataomgeving en het gebruik van zogenaamde ETL-tools (Extract, Transform and Load). Het is echter niet altijd mogelijk om alle gegevens in een data warehouse te bewaren: gegevens die erg weinig benaderd worden, gegevens die niet-gestructureerd zijn, en natuurlijk gegevens waarvoor een real-time toegang noodzakelijk is.

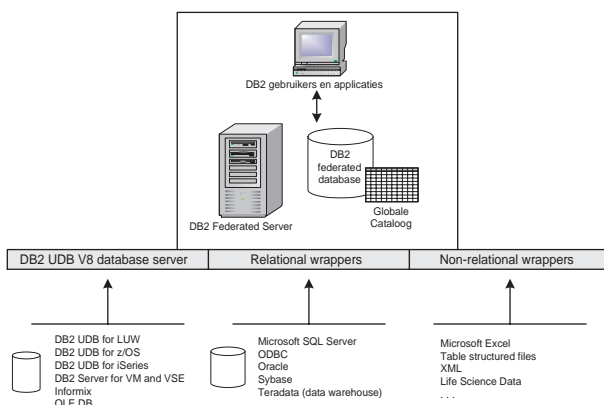
Datafederatie biedt een oplossing voor deze soorten data. Het IBM antwoord op datafederatie is de zogenaamde Federated Database technologie. Hierbij voorziet de 'federated server' één enkele API voor een veelheid aan informatiebronnen. Via SQL kunnen niet enkel alle bestaande relationele databases benaderd worden, maar ook een hele reeks niet-relationele.

Het zal echter duidelijk zijn dat voor sommige bedrijven zowel data-federatie als dataconsolidatie noodzakelijk is. Het is dan ook vanuit dat standpunt dat IBM zijn Information Integrator software heeft ontwikkeld. Niet om de bestaande data warehouses te vervangen, maar eerder als een supplement voor de bestaande technologie.

## De federated database technologie

Vóór DB2 Information Integrator werden de eigenschappen van een federated systeem geïmplementeerd in de DB2 UDB Enterprise Server Edition (vanaf V7.1), eventueel gecombineerd met DB2 Relational Connect en DB2 Life Sciences Data Connect. Met de komst van DB2 information Integrator is het mogelijk om via één product een brede waaier aan informatiebronnen aan te spreken. Figuur 1 geeft hier een overzicht van.

*Figuur 1: de federated database technologie*



In vorige artikels van 'Exploring DB2' (Jaargang 1, Nummer 4, December 2002 en Jaargang 2, Nummer 8, Mei 2004) werd de federated database technologie reeds uitgebreid besproken: WRAPPERS, SERVERS, USER MAPPINGS en NICKNAMES.

De federated objecten worden aangemaakt a.d.h.v. CREATE statements. Deze zijn uit te voeren vanaf de DB2 command prompt of gemakkelijker via het DB2 Control Center aan de hand van wizards.

Enkele voorbeelden van eerder speciale CREATE statements zijn in onderstaand voorbeeld opgenomen. Voorbeeld 1 toont aan hoe aan de hand van DDL statements, integratie met een reeks disparate gegevensbronnen mogelijk is. Andere voorbeelden, alsook CREATE statements voor integratie met relationele databases, vindt u op het net. De hieronder opgenomen voorbeelden gebruiken respectievelijk een Excel-spreadsheet en een XML-document als databron. De

Zie [Relevante definities federated databases](#) op p. 13

Zie [Creatie van federated objecten](#) op p. 13

definitie van Excel-spreadsheet als databron is niet erg uitdagend; dit is echter wel het geval voor het XML-document. Inderdaad, aan de hand van een XPATH-expressie moet een relatie worden gelegd tussen kolommen in de nickname enerzijds, en de relevante attributen van het betreffende XML-document anderzijds.

### *Voorbeeld 1: relevante DDL*

---

```
CREATE WRAPPER excel_wrapper LIBRARY 'db2lsexls.dll'
CREATE SERVER excel WRAPPER excel_wrapper
CREATE NICKNAME feduser.person_excel
(lastname VARCHAR(40), firstname VARCHAR(20), telephone VARCHAR (15),
 birthday DATE, company INTEGER, no INTEGER)
FOR SERVER excel OPTIONS (FILE_PATH 'c:\DB2\personlist.xls')

CREATE WRAPPER xml_wrapper LIBRARY 'db2lsexml.dll'
CREATE SERVER xml WRAPPER xml_wrapper
CREATE NICKNAME feduser.companyxml
compno SMALLINT NOT NULL OPTIONS(XPATH './@Number'),
compname VARCHAR (20) OPTIONS(XPATH './compInfo/compName/text()),
comptel VARCHAR (15) OPTIONS(XPATH './compInfo/compTel/text()),
compbankno VARCHAR (14) OPTIONS(XPATH './compInfo/compBankno/text()),
compsector VARCHAR (40) OPTIONS(XPATH './compInfo/compSector/text())
FOR SERVER xml
OPTIONS(XPATH '//company' ,FILE_PATH 'C:\db2\company.xml')
```

---

## **Nieuw in DB2 Information Integrator**

In vergelijking met DB2 UDB v7 Federated Server bevat DB2 Information Integrator een aantal nieuwe features. Een kleine inventarisatie van deze features.

- meer databronnen: naast alle relationele databases, zijn nu ook een groot aantal niet-relationele databronnen toegankelijk geworden (spreadsheets, XML files, een aantal Life Science data producten, ...);
- groter aanbod van beschikbare platformen voor de installatie van de federated server: niet enkel AIX en Windows NT servers, maar ook Linux, HP-UX, Solaris en Windows 2000 operating systemen;
- READ/WRITE-mogelijkheden: niet enkel SELECT, maar ook INSERT, UPDATE en DELETE voor de meeste relationele systemen;
- transparent DDL: in vorige versies was de creatie van een tabel op een externe relationele databron enkel mogelijk in een pass-through sessie, nu kan het rechtstreeks vanuit het Control Center met de optie Create Remote Table;
- verbeterde mogelijkheden van het DB2 Control Center: vele taken die in vorige versies via de command prompt moesten uitgevoerd worden, kunnen nu in het Control Center. Elk federated object kan op een eenvoudige, gestuurde manier aangemaakt worden met behulp van wizards. De creatie van nicknames is sterk vereenvoudigd dankzij het 'discovery' mechanisme (XML-documenten kunnen bijvoorbeeld opgespoord worden op filenaam, schemanaam of directory path). Ook de opties van wrappers, servers, user map-

pings en nicknames kunnen nu via het Control Center ingesteld worden;

- ondersteuning van Large Objects (LOBs): voor Oracle tabellen zijn alle SQL DML statements mogelijk, voor andere RDBMs enkel SELECT;
- ondersteuning van MQTs (Materialized Query Tables) voor relationele databronobjecten;
- betere performance dankzij geïntegreerde statistische informatie: gebruikmakende van het GET\_STATS utility voor relationele en niet-relationele databronnen, kun je de statistische informatie van de nicknames aanpassen. In feite is het utility vergelijkbaar met het RUNSTATS utility, maar dan voor nicknames i.p.v. tabellen. Na de uitvoering van het utility zal de optimizer van de federated server terug in staat zijn om een optimaal toegangspad voor een federated query te bepalen.

## **Uitvoering van een query door de federated server**

Alle federated objecten moeten gedefinieerd worden in de federated database. De catalogoog van de federated instance bevat alle informatie over de lokale federated objecten en hun opties, alsook over de externe objecten (bv. externe kolomnamen, datatypes en indexinformatie). De federated databasecatalogoog wordt dan ook een globale catalogoog genoemd. Wanneer een client en/of eindgebruiker connecteert met de federated database, heeft hij onmiddellijke toegang tot alle gedefinieerde externe databronnen, via de logische definities van de federated objecten.

Zoals hierboven al werd vermeld, wordt een externe databron benaderd via een nickname. In de definitie van de nickname (meer bepaald in de kolomopties) moet onder andere vermeld worden hoe de lokale nickname gemapped moet worden met het externe data object. Enkele voorbeelden vind je in Figuur 2.

Alle request moeten geformuleerd worden in SQL, ook als de externe databron een ander SQL-dialect, of zelfs helemaal geen SQL spreekt. Wanneer je externe data wil lezen of wijzigen, doe je dit via SELECT, INSERT, UPDATE of DELETE van de nicknames. Quasi alle DML-operaties die door DB2 SQL ondersteund zijn, kan je ook gebruiken in een federated omgeving: JOIN, UNION(ALL), nested table expression, subselects, kolomfuncties, scalar functions, ... . Het is dus perfect mogelijk om met één SQL JOIN statement relationele data te combineren met data in een niet-relationeel formaat.

Alle queries komen terecht bij de DB2 SQL Compiler (en meer bepaald de optimizer) van de federated server. Deze analyseert de query, splitst hem eventueel op in query fragmenten, en maakt gebruik van de informatie in de globale catalogoog om te beslissen of de query ter plaatste kan worden uitgevoerd of (gedeeltelijk) moet doorgestuurd worden naar de gedistribueerde databronnen (=push-down). Een voorbeeld van een optimizer analyse vind je in Figuur 3.

Figuur 2: externe data

ORACLE

CNOO	CONAME	COPIESET	COINSTG	COINSTN	COINSTONS	COINSTNR	COINR
1	LABOUR	1	LABOR	2000	B		0104570
4	LOC COMPUTER	001	LOC	2000	B		0104570
5	LOC COMPUTER	002	LOC	2000	B		0104570
6	LOC COMPUTER	003	LOC	2000	B		0104570
7	LOC COMPUTER	004	LOC	2000	B		0104570
8	LOC COMPUTER	005	LOC	2000	B		0104570
9	LOC COMPUTER	006	LOC	2000	B		0104570
10	LOC COMPUTER	007	LOC	2000	B		0104570
11	LOC COMPUTER	008	LOC	2000	B		0104570
12	LOC COMPUTER	009	LOC	2000	B		0104570
13	LOC COMPUTER	010	LOC	2000	B		0104570
14	LOC COMPUTER	011	LOC	2000	B		0104570
15	LOC COMPUTER	012	LOC	2000	B		0104570
16	LOC COMPUTER	013	LOC	2000	B		0104570
17	LOC COMPUTER	014	LOC	2000	B		0104570
18	LOC COMPUTER	015	LOC	2000	B		0104570
19	LOC COMPUTER	016	LOC	2000	B		0104570
20	LOC COMPUTER	017	LOC	2000	B		0104570
21	LOC COMPUTER	018	LOC	2000	B		0104570
22	LOC COMPUTER	019	LOC	2000	B		0104570
23	LOC COMPUTER	020	LOC	2000	B		0104570
24	LOC COMPUTER	021	LOC	2000	B		0104570
25	LOC COMPUTER	022	LOC	2000	B		0104570
26	LOC COMPUTER	023	LOC	2000	B		0104570
27	LOC COMPUTER	024	LOC	2000	B		0104570
28	LOC COMPUTER	025	LOC	2000	B		0104570
29	LOC COMPUTER	026	LOC	2000	B		0104570
30	LOC COMPUTER	027	LOC	2000	B		0104570
31	LOC COMPUTER	028	LOC	2000	B		0104570
32	LOC COMPUTER	029	LOC	2000	B		0104570
33	LOC COMPUTER	030	LOC	2000	B		0104570
34	LOC COMPUTER	031	LOC	2000	B		0104570
35	LOC COMPUTER	032	LOC	2000	B		0104570
36	LOC COMPUTER	033	LOC	2000	B		0104570
37	LOC COMPUTER	034	LOC	2000	B		0104570
38	LOC COMPUTER	035	LOC	2000	B		0104570
39	LOC COMPUTER	036	LOC	2000	B		0104570
40	LOC COMPUTER	037	LOC	2000	B		0104570
41	LOC COMPUTER	038	LOC	2000	B		0104570
42	LOC COMPUTER	039	LOC	2000	B		0104570
43	LOC COMPUTER	040	LOC	2000	B		0104570
44	LOC COMPUTER	041	LOC	2000	B		0104570
45	LOC COMPUTER	042	LOC	2000	B		0104570
46	LOC COMPUTER	043	LOC	2000	B		0104570
47	LOC COMPUTER	044	LOC	2000	B		0104570
48	LOC COMPUTER	045	LOC	2000	B		0104570
49	LOC COMPUTER	046	LOC	2000	B		0104570
50	LOC COMPUTER	047	LOC	2000	B		0104570
51	LOC COMPUTER	048	LOC	2000	B		0104570
52	LOC COMPUTER	049	LOC	2000	B		0104570
53	LOC COMPUTER	050	LOC	2000	B		0104570
54	LOC COMPUTER	051	LOC	2000	B		0104570
55	LOC COMPUTER	052	LOC	2000	B		0104570
56	LOC COMPUTER	053	LOC	2000	B		0104570
57	LOC COMPUTER	054	LOC	2000	B		0104570
58	LOC COMPUTER	055	LOC	2000	B		0104570
59	LOC COMPUTER	056	LOC	2000	B		0104570
60	LOC COMPUTER	057	LOC	2000	B		0104570
61	LOC COMPUTER	058	LOC	2000	B		0104570
62	LOC COMPUTER	059	LOC	2000	B		0104570
63	LOC COMPUTER	060	LOC	2000	B		0104570
64	LOC COMPUTER	061	LOC	2000	B		0104570
65	LOC COMPUTER	062	LOC	2000	B		0104570
66	LOC COMPUTER	063	LOC	2000	B		0104570
67	LOC COMPUTER	064	LOC	2000	B		0104570
68	LOC COMPUTER	065	LOC	2000	B		0104570
69	LOC COMPUTER	066	LOC	2000	B		0104570
70	LOC COMPUTER	067	LOC	2000	B		0104570
71	LOC COMPUTER	068	LOC	2000	B		0104570
72	LOC COMPUTER	069	LOC	2000	B		0104570
73	LOC COMPUTER	070	LOC	2000	B		0104570
74	LOC COMPUTER	071	LOC	2000	B		0104570
75	LOC COMPUTER	072	LOC	2000	B		0104570
76	LOC COMPUTER	073	LOC	2000	B		0104570
77	LOC COMPUTER	074	LOC	2000	B		0104570
78	LOC COMPUTER	075	LOC	2000	B		0104570
79	LOC COMPUTER	076	LOC	2000	B		0104570
80	LOC COMPUTER	077	LOC	2000	B		0104570
81	LOC COMPUTER	078	LOC	2000	B		0104570
82	LOC COMPUTER	079	LOC	2000	B		0104570
83	LOC COMPUTER	080	LOC	2000	B		0104570
84	LOC COMPUTER	081	LOC	2000	B		0104570
85	LOC COMPUTER	082	LOC	2000	B		0104570
86	LOC COMPUTER	083	LOC	2000	B		0104570
87	LOC COMPUTER	084	LOC	2000	B		0104570
88	LOC COMPUTER	085	LOC	2000	B		0104570
89	LOC COMPUTER	086	LOC	2000	B		0104570
90	LOC COMPUTER	087	LOC	2000	B		0104570
91	LOC COMPUTER	088	LOC	2000	B		0104570
92	LOC COMPUTER	089	LOC	2000	B		0104570
93	LOC COMPUTER	090	LOC	2000	B		0104570
94	LOC COMPUTER	091	LOC	2000	B		0104570
95	LOC COMPUTER	092	LOC	2000	B		0104570
96	LOC COMPUTER	093	LOC	2000	B		0104570
97	LOC COMPUTER	094	LOC	2000	B		0104570
98	LOC COMPUTER	095	LOC	2000	B		0104570
99	LOC COMPUTER	096	LOC	2000	B		0104570
100	LOC COMPUTER	097	LOC	2000	B		0104570

CREATE NICKNAME feduser.oraclecompanies  
FOR ora.oracle.companies

FILE

1, LOC Computer Corp.,02/6784523  
5,003-780034-54,COVERDAT,08879-3844  
3,739-0000673-02,ABIS,016/245610

CREATE NICKNAME feduser.companyfile  
(companyno SMALLINT,  
companybankno CHARACTER (14),  
companyname VARCHAR (40),  
companytelephone VARCHAR (15))  
FOR SERVER flat\_file  
OPTIONS(COLUMN\_DELIMITER ',' ,  
FILE\_PATH 'c:\db2\companyfile.txt')

XML

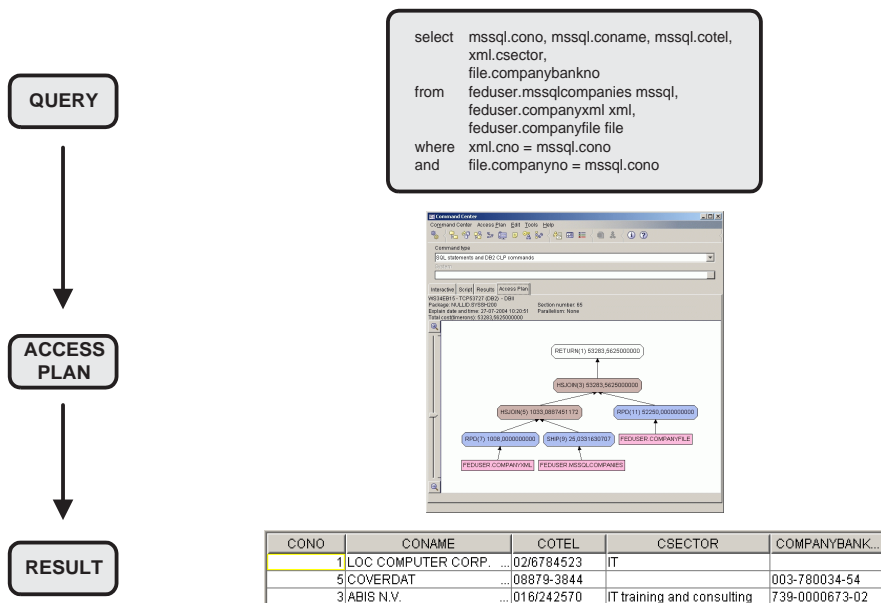
```
<?xml version="1.0" encoding="UTF-8"?>
<companylist>
  <company Number=" 1">
    <companyInformation>
      <companyName> LOC Computer Corp</companyName>
      <companyTelephone> 02/6784523</companyTelephone>
      <companySector> IT</companySector>
    </companyInformation>
  </company>
  <company Number=" 5">
    <companyInformation>
      <companyName> COVERDAT</companyName>
      <companyTelephone> 08879-3844</companyTelephone>
      <companyBankno> 003-780034-54</companyBankno>
    </companyInformation>
  </company>
  <company Number=" 3">
    <companyInformation>
      <companyName> ABIS</companyName>
      <companyTelephone> 016/245610</companyTelephone>
      <companyBankno> 739-0000673-02</companyBankno>
      <companySector> IT training and consulting</companySector>
    </companyInformation>
  </company>
</companylist>
```

CREATE NICKNAME feduser.companyxml  
(companyno SMALLINT NOT NULL  
OPTION(XPATH '//\*[@Number]'),  
companyname VARCHAR (20)  
OPTION(XPATH '://companyInformation/companyName/text()'),  
companytelephone VARCHAR (15)  
OPTION(XPATH '://companyInformation/companyTelephone/text()'),  
companybankno VARCHAR (14)  
OPTION(XPATH '://companyInformation/companyBankno/text()'),  
companysector VARCHAR (40)  
OPTION(XPATH '://companyInformation/companySector/text()'))  
FOR SERVER xml  
OPTION(XPATH '://company' ,FILE\_PATH 'C:\db2\company.xml')

Over het algemeen is een pushdown meer efficiënt, maar de keuze van de cost-based optimizer zal beïnvloed worden door tal van factoren (hoeveelheid data te verwerken, snelheid van de externe databron, snelheid van het communicatiekanaal). Natuurlijk moet ook nagekeken worden of de externe databron de vraag wel kan beantwoorden. Databronnen die geen SQL ondersteunen, kennen bijvoor-

beeld geen COUNT of GROUP BY. In dit geval heeft de optimizer geen andere keuze: alle gegevens worden binnengehaald en de operatie wordt lokaal uitgevoerd. Voor externe databronnen die wel SQL ondersteunen, kan er compensatie optreden: DB2 'vertaalt' de vraag in één of meerdere externe operaties in het externe SQL dialect.

Figuur 3: optimizer analyse



Om externe data te wijzigen gebruik je een SQL INSERT, UPDATE of DELETE. Het is evident dat de nodige autorisaties moeten toegekend zijn op elk van de externe systemen. Wel even opletten: deze acties zijn ondersteund door bijna alle relationele databronnen. Het is echter niet mogelijk gegevens te wijzigen in Excel-bestanden, XML-documenten en table-structured files.

Een pass-through sessie laat toe om bepaalde taken rechtstreeks door de externe databron te laten uitvoeren. Op het moment dat je zo'n pass-through sessie opzet, gebruik je enkel nog de plaatselijke taal/SQL-dialect en definities van de externe databron. De nicknames zijn dus niet meer geldig.

### Besluit

Met DB2 Information Integrator biedt IBM een oplossing voor het probleem van informatieintegratie. Datafederatie kan inderdaad een 'eenvoudige' manier zijn om gedistribueerde data universeel toegankelijk te maken of om bestaande data warehouses uit te breiden.

## DOSSIER 8

### **Over tabel - gecontroleerde partitionering!**

Iedereen kent - en gebruikt - natuurlijk partitioned tablespaces. Reeds lang behoren ze tot de standaard toolset van DB2. En bij elke nieuwe DB2 release werden de mogelijkheden van partitioned tablespaces uitgebreid. Maar toch bleef het één van de meest eigenaardige objecten in DB2 - eigenaardig, omdat logische tabelkarakteristieken enkel konden worden afgedwongen op het niveau van indexcreatie. Men spreekt tot voor DB2 versie 8 dan ook van 'index-gecontroleerde partitionering'.

Dit is niet langer het geval in DB2 versie 8 - 'tabel-gecontroleerde partitionering' is een feit!

Strikt genomen komt het hierop neer: bij het definiëren van de tabel wordt aangegeven dat deze moet worden gepartitioneerd. Hiertoe werd de CREATE syntax uitbereid met de 'PARTITION BY - ENDING AT' instructie. Deze instructie stelt ons in staat op te geven, op basis van welke kolommen (en kolomwaarden) de partitionering moet gebeuren. Op deze kolommen kan voorts geen traditionele 'partitioning index' meer worden aangemaakt.

Deze vorm van partitionering wordt vaak 'range partitioning' genoemd. Andere, door concurrerende database leveranciers vaak ondersteunde vormen van partitionering, zoals hash-partitioning, list-partitioning, of een combinatie van deze vormen, wordt op dit moment nog niet ondersteund.

Merk echter op dat nog steeds, voorafgaand aan de creatie van de tabel, een gepartitioneerde tablespace moet worden aangemaakt. Het aantal partities opgegeven bij het 'create tablespace' statement en het 'create table' statement moet identiek zijn.

Een detailbeschrijving van de voor- en nadelen van 'tabel-gecontroleerde partitionering' maakt het onderwerp uit van een uitgebreid vervolgartikel. Sta ons hier even toe te benadrukken dat door het loskoppelen van 'partitionering' en 'indexering', de bruikbaarheid, onderhoudbaarheid, en effectiviteit van partitionering in een groot aantal applicatieomgevingen aanzienlijk toeneemt. Hoe vaak komt het bijvoorbeeld voor dat een partitiesleutel ('index-gecontroleerde partitionering') ook de sleutel is waarvan applicaties gebruik maken om data op te zoeken? Wat is de zin van het steeds onvoorwaardelijk clusteren van data in partities op basis van diezelfde index? In datawarehouse situaties is tabel-gecontroleerde partitionering zeker geen luxe!

Merk op dat bij gebruik van deze nieuwe vorm van partitionering, de index toolset waarover de DBA beschikt aanzienlijk is toegenomen - meer hierover in datzelfde uitdiepingsartikel. IBM raadt gebruikers van 'index-gecontroleerde partitionering' aan over te stappen naar deze nieuwe structuur. Migratie statements om dit online te doen zijn beschikbaar - de oude partitioning index kan naderhand worden gedropt.

*Eric Venmans (ABIS)*



# DB2 en MQSeries - Integratie

*Kris Van Thillo (ABIS)*

In het eerste artikel van dit nummer hadden we aandacht voor informatieintegratie: de mogelijkheid om transparant, aan de hand van SQL, een veelheid aan databronnen te benaderen. Los van deze architectuur heeft IBM echter ook veel oog voor een rechtstreekse integratie tussen DB2 en MQSeries. In dit artikel wordt hieraan aandacht besteed.

## **Over UDFs**

User Defined Functions (UDFs) laten het de DBA/applicationontwikkelaar toe de mogelijkheden van de DB2 database engine uit te breiden met nieuwe functionaliteiten. UDFs kunnen worden gecreëerd door een applicatieontwikkelaar; maar ze kunnen ook door IBM standaard worden aangeleverd met het oog op deze uitbreiding. DB2 extenders zijn hier een mooi voorbeeld van (Text Extender, XML Extender, etc). Door bijkomende functionaliteit via UDFs (en dus de DB2 database engine) beschikbaar te stellen, kan dit op een centrale, applicatieonafhankelijke wijze gebeuren. Immers, enkel SQL is nodig om de functionaliteit van de UDF - centraal - aan te roepen.

DB2 databases kunnen zodanig worden geconfigureerd, dat ook MQSeries functies, via UDFs, centraal vanuit de DB2 database engine kunnen worden aangeroepen. Men spreekt van de DB2 MQSeries UDF functies.

## **Over MQSeries**

Het MQSeries product zal alsdusdanig voor de lezer van Exploring DB2 geen onbekende zijn. Kort samengevat: MQSeries maakt asynchrone communicatie mogelijk overheen verschillende communicatieprotocollen, tussen applicaties, processen en componenten, geïnstalleerd in vaak disparate omgevingen. Kenmerkend voor deze vorm van communicatie-integratie is het gebruik van een standaard Application Messaging Interface (AMI). De DB2 UDFs waarvan boven sprake maken gebruik van deze AMI-interface. De karakteristieken en de aard van de interactie tussen MQSeries en DB2 worden geconfigureerd aan de hand van 2 belangrijke concepten: SERVICES enerzijds en POLICIES anderzijds.

- Aan de hand van SERVICES definieert de systeemadministrator de MQSeries-server en queue die moet worden gebruikt - een server bijvoorbeeld typisch geconfigureerd op de DB2 database engine. Belangrijk hierbij is de naam van de MQSeries Queue Manager (MQM), en de naam van de te gebruiken queues. Ook de karakteristieken van de queues worden opgenomen. Een service kan dan ook worden beschouwd als de MQSeries DB2 UDF 'aanspreekpunten'.

Zie [Het AMI configuratie tool](#) op p. 15

- Aan de hand van POLICIES wordt voornamelijk de quality of service gedefinieerd: prioriteit van bezorging, en persistentie.

Deze configuratie/definitie wordt opgeslagen in een aantal externe configuratie files ami.xml en amihost.xml, gebruikt door de DB2 MQseries UDFs. De hier bedoelde definities worden typisch aangeemaakt aan de hand van het AMI Administration Tool - een Java GUI.

### Procedure

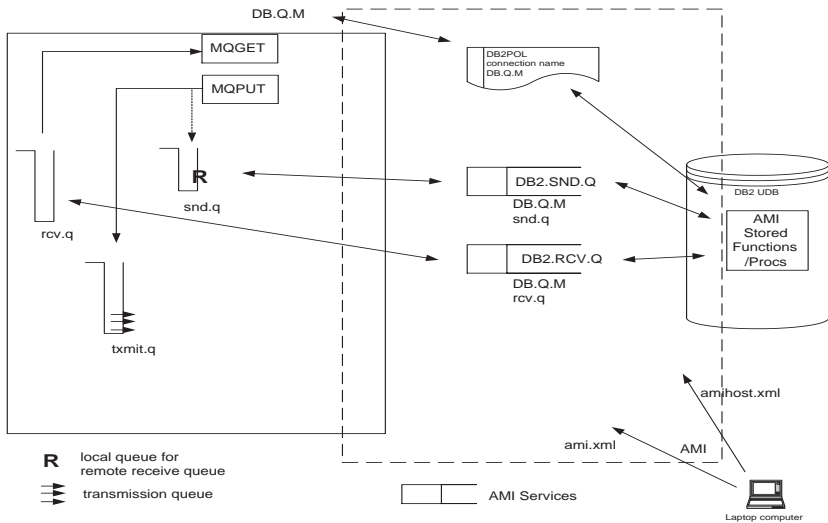
Na het aanmaken van een database, moeten in deze database de MQSeries UDFs worden gecreëerd. Hiertoe is een speciale script voorzien: 'enable\_MQFunctions'. De UDF-omgeving wordt opgezet; en een aantal policies en services worden gedefinieerd uitgaande van een demo, standaard MQSeries omgeving. Merk op dat, afhankelijk van de DB2-versie, vooraf de AMI-interface al dan niet afzonderlijk moet worden geïnstalleerd.

Eens geïnstalleerd, kan de MQSeries-omgeving worden gedefinieerd naar DB2 UDB toe - via het manueel editeren van de XML files hierboven opgesomd, dan wel aan de hand van het GUI tool.

### Een praktisch voorbeeld

Onderstaande figuur geeft een overzicht van een praktische configuratie.

Voorbeeld 1: In de praktijk



Merk het volgende op:

- de geïnstalleerde AMI interface - in de figuur weergegeven als een stippellijn;

- de DB2 UDB omgeving bevat de vereiste UDFs - weergegeven als een vierkant in de DB2 database omgeving. Relevante UDFs zijn bijvoorbeeld MQSEND, MQRECEIVE, MQREADALL, MQRECEIVEALL;
- aan de hand van de amihost.xml en ami.xml files, wordt de MQSeries-omgeving geïdentificeerd naar DB2 UDB toe - zowel de QueueManager 'DB.Q.M' als de queues 'snd.q' en 'rcv.q' zijn in de SERVICE-configuratie opgenomen. Merk op dat 2 services werden aangemaakt - hun naam is identiek aan de MQSeries queues waarmee ze overeenkomen;
- de gedefinieerde policy.

'DB2POL' is de naam van de gedefinieerde POLICY.

Een aantal instructies zijn in onderstaand voorbeeld 2 opgenomen. Merk op dat zowel voor SERVICES als POLICIES defaults bestaan.

#### *Voorbeeld 2: SQL statements*

---

```
select MQSEND('DB2.SND.Q', 'DB2POL', ctitle)
from coures where cid= 1234
```

```
select a.* from table (MQREADALL('DB2.RCV.Q', 'DB2POL')) a
select a.msg from table (MQREADALL('DB2.RCV.Q')) a
```

---

## **Verschillen tussen DB2 UDB en DB2 voor z/OS**

Conceptueel zijn er tussen beide systemen quasi geen verschillen naar gebruik toe (API). Echter, naar opzet en configuratie zijn wel een aantal verschillen op te merken, eigen aan de verschillende architectuur en implementaties op beide platformen:

- het gebruik van UDFs vergt het activeren van een WLM stored procedure address space;
- het gebruik van RSS voor transactie- en commit-coördinatie overheen diverse subsystemen. Merk op dat deze mogelijkheid op LUW-platformen nog niet wordt ondersteund.

## **Besluit**

De DB2 UDB - MQseries integratie kan op een relatief eenvoudige manier worden geïmplementeerd. De mogelijkheden die via de integratie met SQL via UDFs worden geboden, zijn aanzienlijk

## CURSUSPLANNING SEP - DEC 2004

DB2 concepten	375 EUR	27/09 (L), 29/11(W)
DB2 for OS/390, een totaaloverzicht	1625 EUR	20-24/09 (L), 04-8/10 (W), 15-19/11 (W), 06-10/12 (L)
DB2 UDB, een totaaloverzicht	1625 EUR	04-08/10 (W)
RDBMS concepten	325 EUR	20/09 (L), 04/10 (W), 15/11 (W), 06/12 (L)
Basiskennis SQL	325 EUR	21/09 (L), 05/10 (W), 16/11 (W) 07/12 (L)
DB2 for OS/390 basiscursus	975 EUR	22-24/09 (L), 06-08/10 (W), 17-19/11 (W), 08-10/12 (L)
DB2 UDB basiscursus	975 EUR	06-08/10 (W)
SQL workshop	700 EUR	21-22/10 (L), 22-23/11 (W), 22-23/12 (L)
DB2 for OS/390 programmering voor gevorderden	700 EUR	28-29/9 (L), 25-26/11 (L)
DB2 procedural extensions	350 EUR	30/09 (L), 30/11 (W)
DB2 for OS/390: SQL performance	1200 EUR	11-13/10 (L), 01-03/12 (W)
DB2 UDB applicatieperformance	400 EUR	22/12 (W)
Database applicatieprogrammering met JDBC	400 EUR	18/10 (L), 26/11 (L), 14/12 (W)
Fysiek ontwerp v. relationele DB's	700 EUR	07-08/10 (L)
DB2 for OS/390 database administratie	1600 EUR	25-28/10 (W), 14-17/12 (L)
DB2 for OS/390 operations and recovery	1500 EUR	08-10/11 (W)
DB2 UDB DBA 1 - Kernvaardigheid	1600 EUR	26-29/10 (W)
DB2 UDB DBA 2 - Advanced tuning	1200 EUR	13-15/12 (W)
DB2 up-to-date: extended SQL	400 EUR	15/09 (W), 18/10 (L)
DB2 up-to-date: XML in DB2	400 EUR	28/09 (W), 18/10 (L)
Exploring DB2 - live!	1125 EUR	18-20/10/2004 (L)

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

## Relevante definities federated databases

FEDERATED SERVER: de DB2 database server in een federated systeem. Je kan een specifieke DB2 instance reserveren voor de federated activiteiten, of je kan gebruik maken van een bestaande instance. De server staat in voor het verwerken van de database requests die komen van clients en applicaties;

WRAPPER: een logische naam voor een externe databron. In de definitie wordt verwezen naar de 'wrapper library', een specifieke client dll nodig om de externe databron aan de hand van een client/server architectuur te benaderen. Elke databron die je wil benaderen, moet vertegenwoordigd zijn als wrapper;

SERVER, een concrete implementatie van een wrapper. In het server object wordt o.a. informatie bewaard over de versie en database-naam van de externe databron;

USER MAPPING: associatie tussen het ID van de federated server gebruiker en het ID voor de externe databron;

NICKNAME: een lokale, logische naam, die verwijst naar een concreet object van een externe databron. Voor relationele databronnen is dit een tabel of een view, voor niet-relationele databronnen is dit een XML-document, een table-structured file, een Excel-spreadsheet, ... .

## Creatie van federated objecten

creatie van de federated objects for DB2 for z/OS en OS/390

```
-----  
CREATE WRAPPER mf_wrapper  
LIBRARY 'db2drda.dll'  
  
CREATE SERVER mf  
TYPE DB2/MVS VERSION 7.1  
WRAPPER mf_wrapper  
AUTHORIZATION mfuser PASSWORD mfpasw  
OPTIONS(NODE 'mfnode',DBNAME 'mfdb1')  
  
CREATE USER MAPPING FOR feduser  
SERVER mf  
OPTIONS (REMOTE_AUTHID 'mfuser',REMOTE_PASSWORD 'mfpasw')  
  
CREATE NICKNAME feduser.mfpersons  
FOR mf.mfuser.persons  
CREATE NICKNAME feduser.mfcompanies  
FOR mf.mfuser.companies
```

creatie van de federated objects for MS SQL Server

```
-----  
CREATE WRAPPER mssql_wrapper  
LIBRARY 'db2mssql3.dll'  
  
CREATE SERVER mssql  
TYPE MSSQLSERVER VERSION 2000
```

```

WRAPPER mssql_wrapper
OPTIONS(NODE 'mssqlnode',DBNAME 'mssqldb1')

CREATE USER MAPPING FOR feduser
SERVER mssql
OPTIONS (REMOTE_AUTHID 'mssql',REMOTE_PASSWORD 'mssqlpassw')

CREATE NICKNAME feduser.mssqlpersons
FOR mssql.mssqluser.persons
CREATE NICKNAME feduser.mssqlcompanies
FOR mssql.mssqluser.companies

creatie van de federated objects for Oracle
-----
CREATE WRAPPER ora_wrapper
LIBRARY 'db2sqlnet.dll'

CREATE SERVER ora
TYPE ORACLE VERSION 8
WRAPPER ora_wrapper
OPTIONS(NODE 'oranode')

CREATE USER MAPPING FOR feduser
SERVER ora
OPTIONS (REMOTE_AUTHID 'orauser',REMOTE_PASSWORD 'orapassw')

CREATE NICKNAME feduser.orapersons
FOR ora.orauser.persons
CREATE NICKNAME feduser.oracompanies
FOR ora.orauser.companies

creatie van de federated objects voor een file
-----
CREATE WRAPPER file_wrapper
LIBRARY 'db2lsfile.dll'

CREATE SERVER flat_file
WRAPPER file_wrapper

CREATE NICKNAME feduser.companyfile
(companyyno SMALLINT,
companybankno CHARACTER (14),
companyname VARCHAR (40),
companytelephone VARCHAR (15))
FOR SERVER flat_file
OPTIONS(COLUMN_DELIMITER ',',FILE_PATH 'c:\db2\companyfile.txt')
CREATE NICKNAME feduser.personfile
(lastname VARCHAR(40),
firstname CHARACTER(20),
no SMALLINT NOT NULL,
salary SMALLINT)
FOR SERVER flat_file
OPTIONS(COLUMN_DELIMITER ',',FILE_PATH 'c:\db2\personfile.txt')

```

## Het AMI configuratie tool

