ABIS NIEUWSBRIEF VOOR DB2 PROFESSIONALS

JAARGANG 3 - NUMMER 2, NOVEMBER 2004

OPEN CURSOR

De tijd van de nationale en internationale conferences is weer aangebroken - IDUG, DDUG, BDUG, GSE, Forums allerhande, werkgroepen en SIGs. Ze hebben eigenlijk allemaal hetzelfde bedoeling: de DB2 gemeenschap met kennis bijstaan bij het realiseren van specifieke doelstellingen.
Zo ook Exploring DB2!
Aan de hand van korte, praktische artikels pogen we stil te staan bij features die de DB2 professional zeker boeien. En we hopen er ook nu in dit nummer in geslaagd te zijn.
Want het is belangrijk om in de complexe realiteit van de hedendaagse DB2 applicatieontwikkeling de ontwikkelingen van DB2 op de voet te blijven volgen. Via conferences en seminars. Aan de hand van Exploring DB2. En natuurlijk ook via opleidingen - ABIS opleidingen.
Veel leesgenot!

Het ABIS DB2-team.

IN DIT NUMMER:

- Over het belang van utilities in de context van applicatieontwikkeling - in *DB2 utilities en applicaties -1*.
- Nieuwe tendenzen in security en DB2 versie 8, in *Multilevel Security in DB2 for z/OS version 8*.
- En *Dossier 8* vat een aantal korte vernieuwingen op het gebied van SQL syntax samen.
- *Cursusplanning januari 2005 - juni 2005*.
- Web-only: *Praktisch met web services aan de slag - een praktijkvoorbeeld!* - een ABIS presentatie op de GSE NL. (zie *http://www.abis.be/resources/presentations/gsenl20041104webservices.pdf*)

CLOSE CURSOR

In een volgend nummer, natuurlijk aandacht voor het 2e deel van onze reeks rond utilities. En staan we even stil bij 'Online statistics'.

Tot dan!

DB2 utilities en applicaties - 1

Eric Venmans (ABIS)

Inleiding

Een applicatieontwikkelaar die zich gedraagt als een echte, traditionele programmeur, probeert alle opdrachten binnen DB2 uit te werken met behulp van de SQL-taal. Niet te verwonderen want de SQL-interface is de hoofdtoegang tot DB2-gegevens en tot activiteiten met die gegevens. Het is een erg veilige manier om met gegevens om te gaan. Immers:

- Image copies en logging zorgen voor de fysieke integriteit van de gegevens.
- Locking en optioneel referential integrity, check constraints en triggers zorgen voor de logische integriteit van de gegevens.

De controle van fysieke en logische integriteit is uiteraard niet gratis. Het is immers zo dat:

- elke controleactie extra processingtijd (CPU) vraagt.
- logging en de controle van de referential integrity bovendien extra fysieke lees- en schrijfoperaties in het externe geheugen (I/O) vraagt.

Gelukkig geeft dit voor de meeste opdrachten die we DB2 laten uitvoeren, geen enkel probleem.

- Alternatieven voor de DB2-ondersteuning, als die er al zijn, vragen immers ook CPU en I/O.
- Bovendien maakt het systeem meestal efficiënt gebruik van interne hulpmiddelen (lees indexen). Dit vraagt geen extra programmeerwerk want de DB2 optimizer neemt zelf de beslissingen in dit verband. Alternatieven kunnen geen gebruik maken van deze hulpmiddelen, tenzij met (te) veel en (te) complexe extra programmering.

Toch zijn performance en SQL niet automatisch bondgenoten.

Batch-georiënteerde verwerking

Er zijn een aantal opdrachten die toch sneller en efficiënter gebeuren als men ze anders dan via de SQL-interface doorgeeft.

DB2 is namelijk vooral goed in online verwerking van opdrachten: korte, compacte acties in gegevensstrukturen. Echter massaal verwerken van gegevens in batch-georiënteerde applicaties ligt DB2 minder. Niet dat dit niet kan, maar de 'overhead' kan nu wel zwaar, te zwaar gaan doorwegen. DB2 voert namelijk alle acties uit 'in flight'.

Inderdaad, elke aanpassing (insert, update en delete) van een rij uit een tabel kan gepaard gaan met:

- locken van de betrokken gegevens
- controle van check constraints
- controle van de referential integrity
- aanpassingen in andere tabellen omwille van de referential integrity (cascade en set null)
- uitvoeren van triggeracties
- loggen van aanpassingen (basisactie + gegenereerde acties)
- bijwerken van de betrokken indexen
- synchroniseren van alle acties (commit of rollback van de basisactie).

Enkel locken en synchroniseren kan men manipuleren via programmacode.

- Als concurrency geen punt is, kunnen we voor elke tabel die een zware batch-verwerking krijgt, één lock vragen voor de gegevensstructuur in zijn geheel (tenzij dit reeds spontaan gebeurt via een beslissing van de optimizer).
- Op synchronisatie kunnen we besparen door aanpassingen te groeperen. We vragen enkel een commit na x aantal aanpassingen of na y aantal eenheden verwerkingstijd.

Voor een aantal, vooral eenvoudige processen, kunnen we de net beschreven 'overhead' aanzienlijk verminderen. We kunnen namelijk ook van de 'online in flight' overhead batch-georiënteerde processen maken. Hiervoor onttrekken we het proces geheel of gedeeltelijk aan de controle die hoort bij de 'normale' verwerking via SQL. We gebruiken DB2 utilties indien deze de verwerking kunnen overnemen. Ofwel doet de utility volledig wat er moet gebeuren; ofwel gaat het om een stuk voorbereidend werk dat we later vervolledigen door programma's uit te voeren die de gegevens verder verwerken volledig buiten DB2.

Batch-georiënteerde verwerking van de 'overhead' krijgen we nu:

- door logging af te zetten en te vervangen door image copies na verwerking
- door referential integrity, indien relevant, te laten controleren via de CHECK utility
- door index-aanpassingen uit te stellen tot na de verwerking: gebeurt bv. automatisch in de 'index rebuild' faze van een LOAD of RELOAD.

In een reeks vervolgartikels zullen we een aantal opdrachten beschrijven die lange uitvoertijden vragen indien ze met klassieke SQL worden gestuurd. We kunnen deze uitvoertijden soms zeer sterk reduceren door te werken via utilities. We zullen deze alternatieve oplossing(en) beschrijven en van commentaar voorzien.

Voorbeeld: sequentiële leesactie

Alternatieve verwerking via utilities is in de eerste plaats interessant wanneer men DB2-tabellen intensief aanpast. Toch kunnen ook sommige leesakties profiteren van deze alternatieve werkwijze. In dit artikel beschrijven we zulk een situatie als opwarming voor de vervolgartikels. Op basis van een probleemstelling wordt eerst een traditionele oplossing uitgewerkt. Daarna worden alternatieve oplossingen, gebruik makend van utilities, aangedragen.

Probleemstelling.

- Een grote tabel X wordt online bijgewerkt en dit zo goed als permanent (24 uur per dag).
- Men wil elke nacht om 12 uur een 'copy' maken van de rijen die de voorbije dag gewijzigd zijn (+/- 30%). De gekopieerde gegevens moeten weggeschreven worden in een sequentiële file die input wordt voor een ander gegevensverwerkend systeem.
- In elke rij van de betrokken tabel staat de 'datum van de laatste wijziging' genoteerd.

Randvoorwaarden:

- de bestaande transacties die de wijzigingen aanbrengen, mogen niet extra belast worden (dus geen extra rapportering, ook niet onrechtstreeks via een triggeractie).
- men wil een 'exacte' copy (elke gewijzigde rij één maal en slechts één maal).
- de online bijwerkingen door de bestaande transacties kunnen wel stilgelegd worden rond middernacht, maar de periode van onbeschikbaarheid moet tot een minimum beperkt worden.

Een voor de handliggende procedure voor het oplossen van het gestelde probleem is de volgende:

- stoppen (disablen) van de update transacties om middernacht;
- uitvoeren van een programma met in grote lijnen de SQL opgenomen in onderstaande beschrijving - situatie 1;
- backup nemen van de output file;
- starten (enablen) van de update transacties.

Situatie 1

```
DECLARE GewijzigdeRecords CURSOR FOR
        SELECT * FROM X
        WHERE DatumLaatsteWijziging = current date - 1 day
LOCK TABLE X in share mode
OPEN GewijzigdeRecords
PERFORM until ...
...
FETCH GewijzigdeRecords INTO ...
...
CLOSE GewijzigdeRecords
```

Opmerkingen bij deze procedure:

- Update transacties worden gestopt omdat ze tijdens het uitvoeren van het batch proces toch maar tegen time-outs zouden aanlopen.
- DatumLaatsteWijziging is niet geïndexeerd. Een index zou trouwens niet helpen omdat de filterfactor (percentage geselecteerde rijen) erg hoog is.
- Er wordt in de procedure geen image copy gemaakt van de tabel zelf. Dit gebeurt online ('shrlevel change') terwijl de update transacties actief zijn.

Resultaten van testen (met extrapolatie naar productie) geven echter aan dat de voorgestelde procedure te veel tijd in beslag zal nemen.

Via utilities kan men meerdere alternatieven uitproberen. In wat volgt worden een aantal varianten beschreven.

VARIANTE 1A: UNLOAD utility

Variante 1a maakt gebruik van het UNLOAD utility - zie voorbeeld.

Variante 1a - UNLOAD

```
UNLOAD DATA FROM TABLE X
WHEN (DatumLaatsteWijziging = current date - 1 day)
```

Uit metingen blijkt het werken via de UNLOAD utility - gebruik makende van het in voorbeeld opgenomen statement - bijna drie maal sneller te gaan dan de SQL-oplossing.

Voordeel (naast de uitvoertijd) is dat we niet moeten programmeren. Het maken van het utility statement is veel eenvoudiger.

Nadeel is wel de lay-out van de output records. We moeten allicht nog een conversie doen naar het formaat dat gewenst wordt door het target systeem (dit gebeurt echter volledig buiten DB2 terwijl de update transacties alweer actief zijn).

VARIANTE 1B: REORG utility

Het REORG utility statement is een logisch alternatief - variante 1b bevat een voorbeeld.

Variante 1b - REORG

```
REORG TABLESPACE dbx.tbx
UNLOAD EXTERNAL FROM TABLE X
WHEN (DatumLaatsteWijziging = current date - 1 day)
```

Dit is de voorloper van de UNLOAD utility. Deze laatste is pas beschikbaar sinds Versie 7 van DB2. Enig echt verschil met de UNLOAD is dat 'shrlevel' hier altijd NONE is. Voor onze procedure is dit zeker geen probleem.

De opmerking boven wat betreft de layout van de output records is ook hier relevant - naverwerking is noodzakelijk!

VARIANTE 2: COPY utility

Het COPY utility statement - variante 2 - is eveneens een duidelijk bruikbaar alternatief.

Variante 2 - COPY

```
COPY TABLESPACE dbx.tbx
COPYDDN ximcopy
SHRLEVEL REFERENCE
```

De tijd die DB2 nu nodig heeft om dit proces uit te voeren, is nogmaals minder dan de helft van hetgeen voor de UNLOAD utility nodig is. Het niveau van verwerking gaat immers van record per record naar page per page (of, als je het VSAM-niveau verkiest, naar control interval per control interval).

De naverwerking wordt echter iets zwaarder. We hebben namelijk nog geen filtering gedaan op onze gegevens (alles is gekopieerd). Bovendien is de informatie in een formaat dat enkel voor interne DB2 verwerking is bestemd. De naverwerking kan echter weer via een utility gebeuren.

Variante 2 - COPY - naverwerking

```
UNLOAD FROMCOPYDDN ximcopy
FROM TABLE X
WHEN (DatumLaatsteWijziging = current date - 1 day)
```

Ook nu moeten we allicht nog een conversie doen naar het formaat dat gewenst wordt door het targetsysteem.

Als we de onbeschikbaarheid van de update transacties tot een absoluut minimum willen brengen, nemen we deze naverwerking erbij.

Besluit

Hoewel het werken via utilities vooral tijdwinst oplevert voor zware update processen (zie vervolg artikels), kan ook het lezen van gegevens via deze weg aanzienlijk versneld worden. We moeten dan wel te maken hebben met grote tabellen waaruit we gemiddeld meerdere rijen per page nodig hebben.

DOSSIER 8

SQL Bits-and-Bytes

In wat volgt worden een aantal nieuwe eigenschappen van SQL in DB2 for z/OS versie 8 in hoofdlijnen op een rijtje gezet. Een aantal van deze wijzigingen lijken eerder onbelangrijk; ze zijn voornamelijk doorgevoerd met het oog op de SQL-compatibiliteit binnen de DB2 UDB familie.

Identity kolommen bestaan reeds lang in DB2 for z/OS versie 6 en 7 (zie bijvoorbeeld Exploring DB2, Jaargang 1, Nr. 1 en 5). Het SEQUENCE object - de getallengenerator - was reeds gekend in DB2 UDB, en dus nu ook beschikbaar in DB2 for z/OS. Het idee is heel eenvoudig: een specifiek object wordt aangemaakt ('create sequence') dat, tabel- en kolomonafhankelijk, op basis van een aantal definities, getallen voortbrengt. De aldus beschikbare getallen kunnen worden gebruikt in verschillende tabellen; en de sequence generator kan simultaan gerefereerd worden door verschillende applicaties. Sequences kunnen worden aangemaakt als constanten, kunnen 'gaten' vertonen in gegenereerde waarden, en kunnen dubbels bevatten. Ze worden gebruikt in een NEXT|PREVIOUS VALUE FOR sequence expressie. Het sequence object zal zeker gebruikt worden in situaties waar primary key/foreign key data gesynchroniseerd moet worden toegevoegd; of waar denormalisatie heeft geleid tot het aanmaken van tabellen op basis van supertype-subtype constructies, met behoud van een overkoepelende, unieke sleutel.

DB2 versie 8 biedt nu ook de mogelijkheid om verschillende DISTINCT instructies naast mekaar in een SELECT of HAVING constructie aan te wenden.

Het gebruik van triggers, sequences, identity kolommen leidt tot het toevoegen van waarden aan DB2-tabellen; waarden die op dat moment door de applicatieontwikkelaar niet gekend zijn. Indien deze waarden voor verdere verwerking vereist zijn, is een tweede (nu 'lees'-) opdracht nodig. DB2 versie 8 heeft het 'SELECT FROM INSERT' statement beschikbaar om dit te vermijden: schrijven (INSERT) en lezen (SELECT) wordt in één enkele actie gecombineerd. Een aantal beperkingen zijn van toepassing, onder andere in functie van de omgeving waarin de SQL wordt gebruikt, het type SQL statement, en de karakteristieken van de bestaande dependent objecten (triggers, etc).

Ook de scalaire fullselect wordt nu ondersteund. Concreet: een fullselect kan worden toegevoegd tussen haakjes in een standaard expressie, voor zover de fullselect null dan wel één waarde als resultaat oplevert (nested table expressions zijn hiervoor een alternatief). Zowel in de SELECT als WHERE instructies wordt dit gebruik ondersteund.

Het is in DB2 for z/OS versie 8 toegestaan kolomnamen in INSERT en UPDATE statements te kwalificeren met de tabelnaam. Dit geldt ook voor de set instructie in het UPDATE statement. In dit laatste geval is bij aanwezigheid van een specifieke correlatie naam voor de tabel, het gebruik van deze correlatienaam als qualifier verplicht.

En als u tenslotte echt complexe SQL wil coderen: de maximale lengte van de SQL tekststring is ondertussen opgetrokken tot 2 MB (single-byte character sets).

Tom Avermaete (ABIS)

Multilevel Security in DB2 for z/OS versie 8

Kris Van Thillo (ABIS)

'Security': een verzamelwoord dat een hele reeks van elementen en componenten bevat. Vaak bedoelen we zowel toegangscontrole op niveau van het DB2-systeem, als op het niveau van DB2-objecten. En natuurlijk is er ook - zij het onrechtstreeks - de link met 'integriteit'. Koppel hieraan de problematiek van 'auditing', en de cirkel is rond.

Op het DB2 for z/OS platform heeft RACF steeds een belangrijke rol gespeeld wat betreft security. In het verleden vaak op een wat beperkte manier - primaire en secundaire authorisatiegegevens waren (en zijn) strikt genomen van RACF afkomstig. Vandaag de dag wordt echter reeds vaak de security op het niveau van het volledige interne DB2 'objectmodel' door RACF beheerd.

Mits het gebruik maken van DB2 for z/OS versie 8, biedt RACF nu een bijkomede mogelijkheid: de implementatie van 'multilevel security' in DB2. Een inleidende bespreking.

Definities en context

Gegeven de enorme toename aan data opgeslagen in DB2-systemen, alsook de verscheidenheid aan omgevingen (al dan niet controleerbaar, denk maar aan het web) van waaruit DB2-data wordt benaderd, is vaak de vraag ontstaan naar ROW LEVEL security. Met name, de wens om op het niveau van individuele rijen 'rechten' te kunnen toewijzen. Omdat toewijzen van rechten op rijniveau aan users en groepen beheerstechnisch een aanzienlijk probleem is, en omdat vaak toegang tot data hiërarchisch op groepsniveau kan worden georganiseerd, wordt ROW LEVEL security dan ook meestal als een hiërarchisch security systeem geïmplementeerd.

Waar gaat het dus over?

- HIERARCHISCHE SECURITY, d.w.z. toekennen van 'rechten' op basis van hiërarchische classificatie van 'users' in de brede zin - ook jobs, procedures, etc. komen in aanmerking. Een reeks security LEVELs of klassen worden aldus aangemaakt, geïdentificeerd aan de hand van LABELS; hun onderlinge hiërarchie bepaalt de toegangs'rechten';
- CATEGORIES, dwz. de inhoud, het rechtenprofiel van een boven genoemd LEVEL;
- LABELS, dwz. de level naam. De verhouding tussen het security label van het object en/of de rij enerzijds, en anderzijds, het security label van de 'user' bepaalt toegangs- en manipulatiemogelijkheden.

Voor implementatie op rijniveau moet een DB2-tabel hiertoe van een extra, speciale kolom worden voorzien. Merk op dat multilevel security ook op het niveau van DB2-objecten kan worden geëmplementeerd.

ROW LEVEL security is geen nieuw concept. Database vendors gebruiken een veelheid aan termen voor implementaties die globaal gesproken dezelfde lading dekken: 'Virtual Private Databases', 'Label Security'. Oracle heeft 'Label Security' geïmplementeerd in 1979 - maar dat heeft waarschijnlijk te maken met hun eerste klant: de CIA.

RACF in z/OS V1R5 - SecureWay Security Server

Het opzetten van multilevel security vergt een coördinatie tussen DB2 for z/OS versie 8 en RACF. Zowel de definitie van de security labels per 'user' enerzijds, als de opbouw van de security hiërarchie anderzijds, vindt plaatst binnen RACF.

Ook het verder besproken 'write-down' recht wordt op het niveau van RACF (profielen) toegekend.

Bij aanloggen (lokale connecties) wordt het SECURITY LABEL van het primaire DB2 authorisatie-id gebruikt.

Multilevel security op rij niveau is beschikbaar bij gebruik van zowel DB2 security checking als externe access control mechanismen (bijvoorbeeld door RACF). DB2 object multilevel security is enkel beschikbaar bij gebruik van externe access controle mechanismen.

Evaluatie van toegang

Toegang tot data en objecten is afhankelijk van de onderlinge relatie tussen security labels. Twee belangrijke termen moeten worden geëntroduceerd.

Een eerste belangrijke term is 'dominantie': een eerste label domineert een tweede als en slechts als:

- het eerste label hiërarchisch hoger staat of op hetzelfde niveau als het tweede;
- de security categorieën van het eerste label volledig deze van het tweede omvatten.

Een tweede belangrijke term is 'equivalentie': een eerste label is equivalent met een tweede als en slechts als:

- beide labels identiek zijn;
- beide labels niet identiek zijn, maar wel hetzelfde level in de hiërarchie hebben EN dezelfde security categorieën bevatten.

Uitgaande van deze definitie, is het volgende van toepassing.

- indien het security label van de user het label van de rij/het object domineert, is read-only access toegestaan;
- indien het security label van de user label equivalent is aan het label van de rij/het object, is read-write toegang mogelijk;
- indien het security label van de user het label van de rij/het object niet domineert, is toegang niet mogelijk.

WRITE-DOWN rechten

Zoals boven reeds aangegeven, vergt read-write toegang tot gegevens security label equivalentie. Dit is een relatief 'strenge' eis. Het WRITE-DOWN recht staat een 'user' read-write toegang toe zonder label equivalentie; de huidige hiërarchische relatie met de rij/het object is niet langer relevant.

Het WRITE-DOWN controle- en implementatiemechanisme moet uitdrukkelijk geactiveerd worden.

- Is dit mechanisme geactiveerd, dan moet WRITE-DOWN uitdrukkelijk worden toegekend, voordat het kan worden gebruikt;
- Is dit mechanisme niet geactiveerd, wordt WRITE-DOWN recht standaard toegepast.

Rechtentabel

Onderstaande figuur geeft een overzicht (basis instructies) van de door DB2 toegepaste regels in de context van multilevel security - ROW LEVEL security.

Multilevel security - ROW LEVEL - toegestane acties

```
SELECT opdrachten   ° user label dominant

DELETE opdrachten   ° labels equivalent:
                      RIJ verwijderd

                    ° user label dominant + geen write down control:
                      RIJ verwijderd

                    ° user label dominant + write down control:
                      -> WRITE DOWN toegekend: RIJ verwijderd
                      -> geen WRITE DOWN: error

UPDATE opdrachten   ° labels equivalent:
                      RIJ updated
                      -> geen write down control: elk LABEL OK
                      -> write down control + WRITE DOWN toegekend:
                         elk nieuw label OK
                    -> write down control + geen WRITE DOWN toegekend:
                         enkel user label OK

                    ° user label dominant:
                      -> geen write down control:
                         RIJ updated + elk nieuw label OK
                      -> write down control + WRITE DOWN toegekend:
                         RIJ updated + elk nieuw label OK
                    -> write down control + geen WRITE DOWN toegekend:
                         ERROR

INSERT opdrachten   ° geen write down control:
                    INSERT OK; elk nieuw label OK; default: user LABEL

                    ° wel write down control:
                      -> WRITE DOWN toegekend:
                     INSERT OK; elk nieuw label OK; default: user LABEL
                      -> geen WRITE DOWN toegekend:
                         INSERT OK; label wordt user LABEL
```

Tenslotte nog even volgende opmerkingen:

- Ook voor de utilities LOAD, UNLOAD en REORG moet met security labels rekening worden gehouden - afhankelijk van het utility en de gespecificeerde opties komt dit overeen met de boven geschetste regels;
- MQTs en GLOBAL TEMPORARY TABLES hebben een specifieke implementatie regels.

zie *Voorbeeld Multilevel Security* op p. 13

DB2 tabel setup

Om in aanmerking te komen voor ROW LEVEL security op basis van multilevel security, zijn volgende voorwaarden vereist wat betreft de tabelstructuur.

- De tabel bevat een kolom van het type single byte, alfanumeriek, lengte 8, not null with default. De kolom kan eventueel achteraf worden toegevoegd met het ALTER TABLE statement indien vereist.
- De betreffende kolom wordt gedefinieerd met optie AS SECURITY LABEL. Slechts één dergelijke kolom kan worden aangemaakt per tabel.
- De betreffende kolom kan niet worden aangemaakt met field procedures, constraints, etc. Unieke constraints zijn wel toegestaan.
- ROW LEVEL security kan niet worden gedesactiveerd, enkel een drop van de tabel is nog mogelijk.
- INDEX-ONLY toegang tot tabel data is enkel mogelijk als de betreffende LABEL kolom ook deel uitmaakt van de index zelf.

Merk tenslotte op dat deze speciale kolom door DB2 NIET automatisch aan het zicht wordt onttrokken; views zijn ook hier weer noodzakelijk!

Het multilevel security mechanisme wordt door DB2 onbeperkt en automatisch toegepast op een transparante wijze. Wijzigingen aan applicaties zijn dus niet noodzakelijk.

Alternatief?

Het is natuurlijk steeds mogelijk om als alternatief voor ROW LEVEL security met views en join-structuren te werken. Echter:

- views zijn fundamenteel niet transparant: ze moeten uitedrukkelijk aangemaakt worden, en de applicatieontwikkelaars moeten ze uitdrukkelijk gebruiken;
- update van data vergt in deze context meestal ingewikkelde set-up van ondersteunende DB2-structuren, met name, triggers en stored procedures.

Deze techniek wordt vaak aangewend in ‘read-only’ situaties.

CURSUSPLANNING JAN - JUN 2005

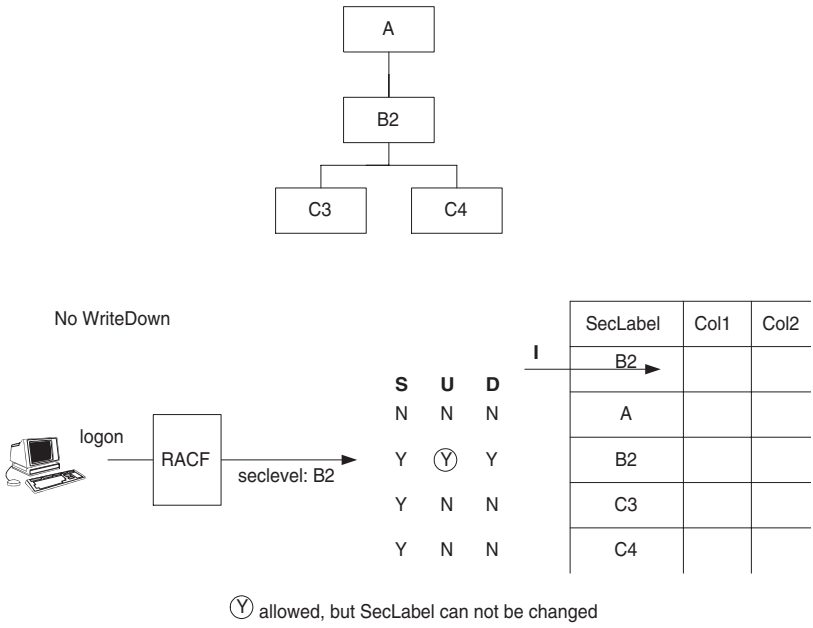
DB2 concepten	375 EUR	14/04 (L), 26/05 (W)
RDBMS concepten	325 EUR	10/01 (L), 24/01 (W), 14/03 (L), 11/04 (W), 09/05 (L), 13/06 (W), 20/06 (L)
Basiskennis SQL	325 EUR	11/01 (L), 25/01 (W), 15/03 (L), 12/04 (W), 10/05 (L), 14/06 (W), 21/06 (L)
DB2 for OS/390 basiscursus	975 EUR	12-14/01 (L), 26-28/01 (W), 16-18/03 (L), 13-15/04 (W), 11-13/05 (L), 15-17/06 (W)
DB2 UDB basiscursus	975 EUR	13-15/04 (W), 27-29/06 (L)
SQL workshop	700 EUR	03-04/03 (L), 25-26/04 (W), 23-24/05 (L), 27-28/06 (W)
DB2 for OS/390 programmering voor gevorderden	700 EUR	10-11/03 (L), 09-10/06 (W)
Gebruik van DB2 procedural extensions	350 EUR	08/03 (L), 21/06 (W)
DB2 for OS/390: SQL performance	1200 EUR	11-13/04 (L), 23-25/05 (W)
DB2 UDB applicatieperformance	400 EUR	09/05 (W)
Database applicatieprogrammering met JDBC	400 EUR	14/03 (L), 18/04 (W), 09/05 (L)
Tunen van Java applicaties voor DB2	400 EUR	24/01 (W), 21/04 (L)
Fysiek ontwerp v. relationele DB's	700 EUR	30-31/05 (L)
DB2 for OS/390 database administratie	1600 EUR	21-24/03 (W), 13-16/06 (L)
DB2 for OS/390 operations and recovery	1500 EUR	22-24/06 (W)
DB2 UDB DBA 1 - Kernvaardigheid	1600 EUR	25-28/04 (W)
DB2 UDB DBA 2 - Advanced tuning	1200 EUR	17-19/05 (W)
Extended SQL in DB2	400 EUR	07/03 (L), 29/06 (W)
XML in DB2	400 EUR	11/04 (L), 20/06 (W)

Bijlage

Voorbeeld Multilevel Security

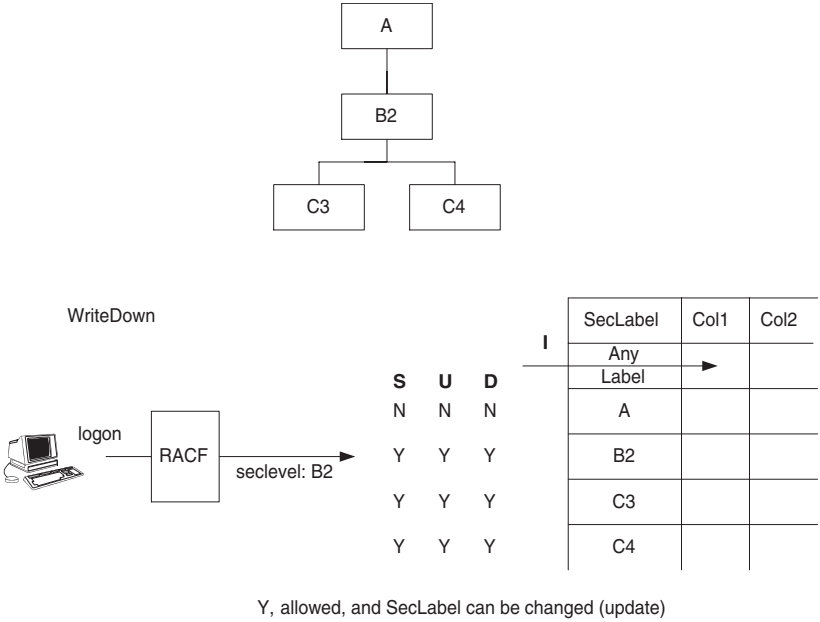
Een eerste voorbeeld wordt beschreven middels figuur 1. Bovenaan elke tekening is de security LABEL hiërarchie opgenomen. De uitgevoerde operaties worden aangegeven als S(select), U(pdate), I(nsert), D(elete). Respectievelijk 'Y' en 'N' geven aan of een bepaalde actie door een gebruiker met seclevel B2 toegestaan is of niet.

Figuur 1 - Multilevel Security - geen 'writedown'



Een tweede voorbeeld wordt beschreven middels figuur 2.

Figuur 2 - Multilevel Security - 'writedown'



Y, allowed, and SecLabel can be changed (update)