

DB2 Darwin achterna: Schema-evolutie

Peter Vanroose

ABIS Training & Consulting



Nationale GSE-conferentie “Keep IT Going”

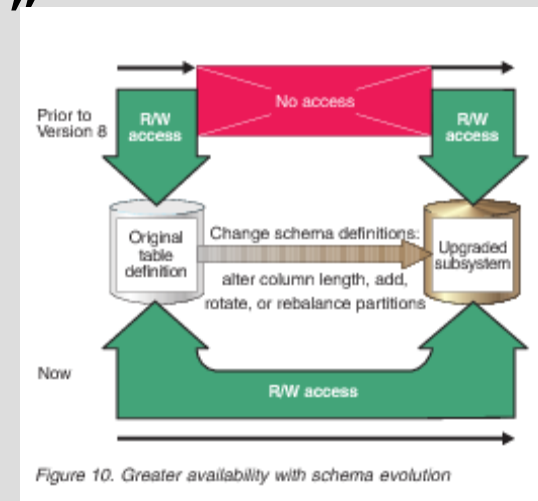
Garderen, 29 Oktober 2009

Aanleiding: evolutie en DB2 ?

- aanleiding 1: Darwin-jaar 2009
- aanleiding 2: “schema-evolutie” in DB2:
 - schema? --> XML?
 - “evolutie”-theorie: survival of the fittest?
 - evolueert DB2? (1983 – 2009)
 - evolueert (wijzigt) de data in DB2?
 - evolueert (wijzigt) de meta-data (object-definities)?
- Antwoord op deze en andere vragen...

Aanleiding: evolutie en DB2 ?

- Citaat uit document van DB2 v8 z/OS “What's New?” (GC18-7428) :
“Greater availability and flexibility with online schema evolution”



- Niet hernomen in “What's New” van DB2 9...
- Vraag: “Is er nog evolutie in DB2 9?”

DB2 v8 “What's New?”

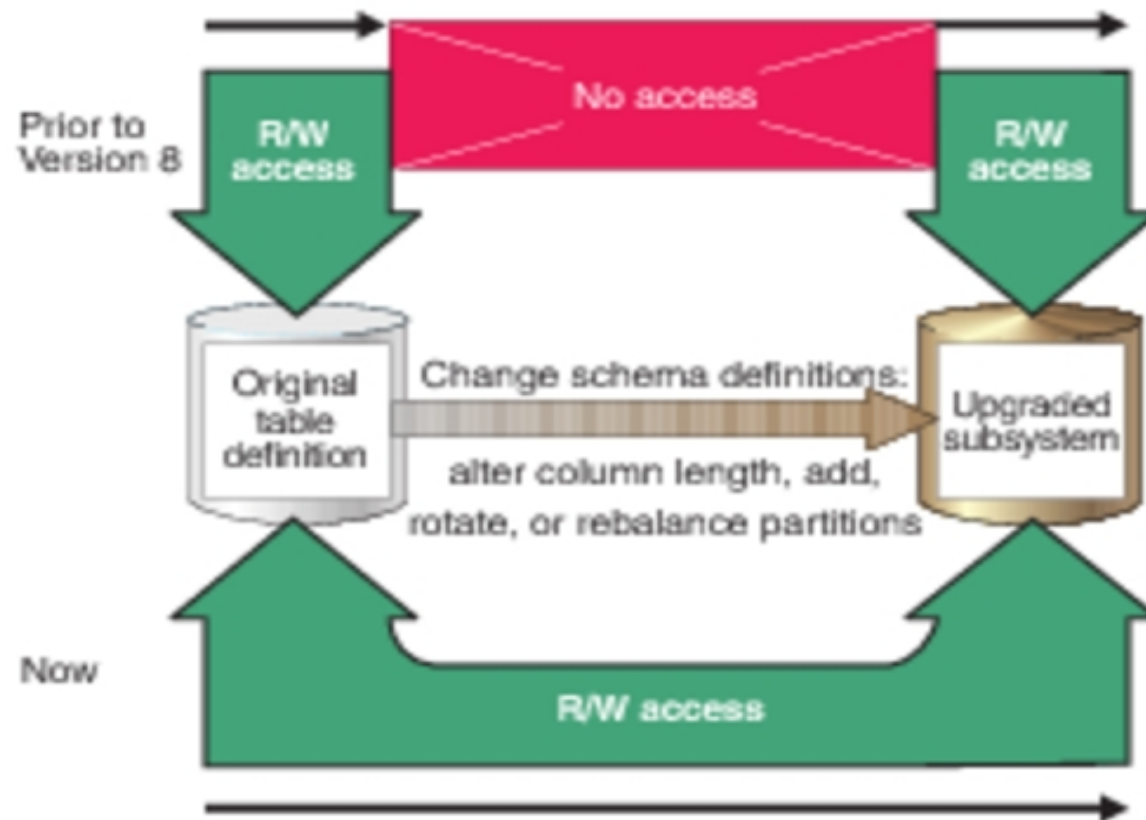


Figure 10. Greater availability with schema evolution

Agenda

- Afbakening van het onderwerp
- tabellen
- indexen
- tablespaces
- andere objecten
- partitionering
- “clone” tables

Afbakening onderwerp

- Object-definities (i.h.b. tabellen)
aanpassen
zonder data onbeschikbaar te maken
(voor te lange tijd)
- Bij uitbreiding ook voor andere objecten:
 - packages/plans
 - indexen
 - tablespaces
 - databases (of eigenlijk: DBDs)

Waarom belangrijk(er)?

- Data moet 24/7 beschikbaar zijn
- “Support window” korter
- Nauwere koppeling tussen applicatie-ontwikkeling en data-design
- Steeds snellere vraag naar aanpassingen (extreme programming; Rational Unified Process (RUP) ==> iteratief ontwikkelen; SAP e.a.; ...)

Waarom doet DB2 zo moeilijk?

- tabel: “logisch” concept;
kolom-definities = structuur van de rijen
- fysische realiteit: tablespace
structuur van de records
- structuur wijzigen
==> ogenblikkelijke fysische wijziging?
NEE: DDL is *nooit* DML (“lazy” change)
enige uitzondering: CREATE INDEX
- andere afhankelijkheden, i.h.b. packages:
“static” SQL is referentiemodel voor DB2

Technische scope

- DDL: ALTER
(table, tablespace, index, view, ...)
- DDL: DROP en dan CREATE
- BIND en REBIND (package, plan)
- GRANT, REVOKE
- REORG: onverwachte neveneffecten
 - RRF
 - partitionering
 - indexen tijdelijk onbeschikbaar

Concreet voorbeeld

- Tabel: klantgegevens

```
CREATE TABLE clients
( id          INTEGER      NOT NULL
, name       CHAR(45)     NOT NULL
, address    VARCHAR(240)
, telephone  DECIMAL(18,0)
, PRIMARY KEY (id)
)
IN db.ts
```

Agenda

- Afbakening van het onderwerp
- **tabellen**
- indexen
- tablespaces
- andere objecten
- partitionering
- “clone” tables

Tabeldefinitie wijzigen

- Online schema change:
 - via ALTER TABLE
 - indien DML geen time-out “ziet”
data blijft 24/7 beschikbaar
 - mogelijk is er tijdelijk verlaagde efficiëntie!
vereist REORG van de tablespace
evtl. REGENERATE van view
evtl. REBUILD van index(en)

Tabeldefinitie wijzigen

- Kolom toevoegen
 - lichte ingreep
 - mogelijk met ALTER TABLE (sinds lang)
 - wijzigt de catalog: extra lijn SYSCOLUMNS
 - bestaande records in tablespace: geen effect
 - bestaande applicaties: geen effect
 - bestaande views: geen effect (! “select *”)

```
ALTER TABLE clients  
ADD COLUMN email VARCHAR(100)
```

Tabeldefinitie wijzigen

- Kolomkarakteristieken wijzigen:
 - **datatype wijzigen**: kan sinds DB2 v8 **met beperkingen** (zie volgende slide)
 - bestaande records in tablesp.: (even) geen effect
==> versioning (denk ook aan image copies)!
 - bestaande index & views: automatisch aangepast

```
ALTER TABLE clients  
ALTER name SET DATA TYPE  
VARCHAR(1000)
```

Tabeldefinitie wijzigen

- **datatype wijzigen:** beperkingen:

 - numeriek --> numeriek, tekst --> tekst

 - bestaande data mag niet “onmogelijk” worden:

geen (var)char(100) naar (var)char(99)

geen int naar smallint

de kolom mag niet voor RI gebruikt zijn

de kolom mag geen identity column zijn

er mag geen index on expression naar verwijzen

er mag geen MQT naar de tabel verwijzen

(indien geen edit proc, field proc, validation proc)

(niet mogelijk voor LOB, XML, DATE, TIME)

Tabeldefinitie wijzigen

- Kolomkarakteristieken wijzigen:
 - default-waarde wijzigen (sinds DB2 9)
is “pure” DDL: geen effect op bestaande data
 - kolomnaam wijzigen: vanaf DB2 X ?
 - “NOT NULL”-eigenschap wijzigen:
onmogelijk! ==> DROP & CREATE (LUW wel)
 - kolom verwijderen:
onmogelijk! ==> DROP & CREATE
alternatief: datatype wijzigen naar VARCHAR(n)
en kolom “implicitly hidden” maken

Tabeldefinitie wijzigen

- Tabelnaam wijzigen
 - niet mogelijk indien \exists view, synonim of trigger
 - “cascaded update” in de catalog: (behalve SYSSTMT)
 - bestaande indexen: blijven werken
 - bestaande autorizaties: blijven behouden
 - bestaande RI: automatisch overgenomen
 - packages/plans **invalidated** ==> REBIND nodig

```
RENAME TABLE clients TO entities
```

```
CREATE VIEW clients AS  
  SELECT * FROM entities  
  WHERE kind='client'
```

Tabeldefinitie wijzigen

- tabel-**partitionering** aanpassen:
 - zie later
- **unique index** creëren: wordt dadelijk gecontroleerd
 - is on-line, maar S-lock blokkeert DML-wijzigingen
- **primary key** toevoegen / weghalen
 - lichte ingreep (!) zonder side effect (tenzij reeds RI)
- foreign key / check **constraint** weghalen / **toevoegen**
 - zeer snel, maar **check pending**: dus niet on-line!
 - overweeg “not enforced” FK (sinds DB2 v8)
- clustering sequence wijzigen
 - via ALTER van de index(en)

Tabeldata wijzigen

Ook twee DML-situaties van “online evolution”:

- “**massive delete**”: tabel (of partitie) leeg maken

TRUNCATE TABLE entities

IGNORE DELETE TRIGGERS

indien enkel eerste partitie: gebruik REORG

- “**massive replace**”:

- Voorzie een “kloon” (clone table)

==> zie verderop

Agenda

- Afbakening van het onderwerp
- tabellen
- **indexen**
- tablespaces
- andere objecten
- partitionering
- “clone” tables

Indexdefinitie wijzigen

- DROP en CREATE van index
 - heeft “weinig” negatieve neveneffecten:
 - index niet rechtstreeks zichtbaar met DML
 - index-inhoud is altijd up-to-date
 - ==> in-flight “online” runtime overhead
 - maar: packages / plans invalidated
- ALTER INDEX:
 - kan invalidation vermijden
 - rebuild-pending state (of AREO) mogelijk

Indexdefinitie wijzigen

- kolom toevoegen aan index (sinds DB2 v8)
 - advisory reorg-pending state
 - bestaande packages blijven werken
 - mag niet PK-index or FK-index zijn
- PADDED <---> NOT PADDED
 - indien VARCHAR-kolommen in de index
 - voordelen van NOT PADDED:
 - index-only access wordt mogelijk
 - compact
 - rebuild-pending state

Indexdefinitie wijzigen

- CLUSTER <---> NOT CLUSTER
 - puur declaratief
 - pas effectief na volgende REORG van tabel
- Indexnaam wijzigen (sinds DB2 9)
 - “cascaded update” in de catalog:
 - bestaande autorizaties: blijven behouden
 - plan_table entries worden **niet** aangepast!
 - packages/plans blijven geldig
==> want OBID niet gewijzigd

Agenda

- Afbakening van het onderwerp
- tabellen
- indexen
- **tablespaces**
- **andere objecten:** views, databases, ...
- partitionering
- “clone” tables

Tablespace-definitie wijzigen

- Pre DB2 v8: ALTER van
 - LOCKSIZE, FREESPACE, PCTFREE, BUFFERPOOL, USING STOGROUP, PRIQTY, SECQTY, COMPRESSED, CCSID, ...
- ALTER van
 - LOGGED, MAXPARTITIONS
- Ook voor indexen en voor afzonderlijke partities
- Wordt effectief na volgende **REORG**
 - ==> is een zeer “on-line” schema change

View-definities wijzigen

- DROP & CREATE: verlies van autorisaties
implicit drop van dep. views
- REGENERATE:
ALTER VIEW clients REGENERATE
d.i. geen impliciete REBIND (b.v. SELECT *)

Database-definities wijzigen

- ALTER van
 - BUFFERPOOL, STOGROUP, CCSID
- Wijzigt enkel de defaults voor nieuwe tablespaces
 - ==> is een zeer on-line schema change
 - ==> heeft nooit effect op bestaande objecten

Maar ... DDL heeft grote impact op DBD !
(X-lock conflicteert met S-lock van b.v. BIND)

Andere database-objecten

- Buffer pools, storage groups
- zParms: meer on-line changes mogelijk
- Autorisaties:
 - geen afzonderlijke objecten: zeer on-line
 - maar: let op voor invalidated packages/plans
 - let ook op voor cascaded REVOKE !
 - ook belangrijk voor availability: SQLCODE -551

Andere database-objecten

- Packages en plans:
 - schema changes: REBIND
wijzigt: QUALIFIER, OWNER,
ISOLATION, REOPT,
PLANMGMT,
EXPLAIN, ...
 - nodig voor invalidated packages/plans
 - mogelijk auto-rebind
 - DB2 9: herstel vorige versie
--> goed voor availability

Agenda

- Afbakening van het onderwerp
- tabellen
- indexen
- tablespaces
- andere objecten
- **partitionering**
- “clone” tables

Partitionering

- Pre-v8:
 - Logische partitionering: tabel-niveau
 - Fysische partitionering: tablespace-niveau
 - Partitionering gedefinieerd door: **partitioning** index
- v8:
 - Eerst fysisch partitioneren: tablespace, NUMPARTS
 - Dan tabel creëren en dadelijk partitionering definiëren:

```
CREATE TABLE entities ( ..... )
PARTITION BY (kind)
(PARTITION 1 ENDING AT ('client'),
PARTITION 2 ENDING AT (MAXVALUE) )
```

Partitionering

- v9:
 - Eerst fysisch partitioneren mogelijk maken:

```
CREATE TABLESPACE db.ts MAXPARTITIONS 1
```
 - Dan tabel creëren:

```
CREATE TABLE entities ( ... ) IN db.ts  
PARTITION BY (kind)  
(PARTITION 1 ENDING AT ('client'))
```
 - Tweede partitie toevoegen:

```
ALTER TABLESPACE db.ts MAXPARTITIONS 2  
ALTER TABLE entities  
ADD PARTITION ENDING AT (MAXVALUE)
```


Partitionering: voordelen

- Online schema change voordelen:
 - ALTER TABLESPACE db.ts ALTER PART 1 ..
 - REORG TABLESPACE db.ts PART 1
 - ==> partitie 2 blijft 100% on-line
tenzij er non-partitioned indexes (NPIs) zijn
die door een package/plan gebruikt worden
- Maak dus van alle indexen DPSIs:
 - Data-partitioned secondary index
 - Bestaat zelf uit 2 fysische partities
 - Voor queries met “WHERE kind='client' AND ...”

Partitionering en DB2 9

- Universal tablespaces
 - nieuw sinds versie 9
 - zijn t.z.t. segmented en partitioned
 - de nieuwe default (b.v. voor impliciet gecreëerde TS)
 - Partitioned by range
 - zie vorig voorbeeld
 - getriggerd door clause MAXPARTITIONS in CREATE
 - Partitioned by growth
 - is de default
 - mist alle voordelen van partitioning (behalve datasize)
- ==> geen DPSIs, geen partitie-selectieve queries

Partitionering en DB2 9

- Universal tablespaces en on-line changes
 - onmogelijk om van tablespace type te veranderen
- ==> segmented, simple, partitioned:
 - kan geen universal worden
 - tenzij met DROP / CREATE
 - Dit reduceert het nut van de on-line voordelen:
 - * partition dependency
 - * partities toevoegen
 - * DPSIs
 - Wordt beloofd voor DB2 X

Agenda

- Afbakening van het onderwerp
- tabellen
- indexen
- tablespaces
- andere objecten
- partitionering
- **“clone” tables**

Clone tables

- Probleemstelling:
 - Zeer grote tabel
 - Massieve DML: insert, update, delete, combinatie
 - Reden: replicatie (b.v. 1x per dag)
 - Tabel is verder read-only
- Tijdelijke onbeschikbaarheid van de data:
 - SQL DML: langlopende X-lock
 - kan lang duren vooraleer begint
 - LOAD REPLACE: langlopende X-lock
 - CLAIM ==> kan sneller beginnen

Clone tables (DB2 9)

- Voeg een “kloon” toe aan de basistabel:
ALTER TABLE entities ADD CLONE ent2
 - Bevolk deze kloon met de nieuwe data:
INSERT INTO ent2 VALUES (...)
of LOAD DATA INTO TABLE ent2
 - Wissel de twee bestanden van plaats:
EXCHANGE DATA
BETWEEN TABLE entities AND ent2
 - Beperkingen:
 - geen RI; geen triggers; enkel voor universal TS
- Implementatie: zoals **on-line REORG**

Besluit

- Online schema evolutie: een noodzaak
- Leer de nieuwe mogelijkheden (v8, 9) kennen
- Let op voor verborgen gevaren
- I.h.b.: overweeg (zinnig) partitioneren met UTS
maak tekstkolommen VARCHAR
maak indexen NOT PADDED
gebruik TRUNCATE

Q & A

• ...

DB2 Darwin achterna: Schema-evolutie

Peter Vanroose

pvanroose@abis.be

ABIS Training & Consulting

Leuven / Woerden

<http://www.abis.be/>



Nationale GSE-conferentie "Keep IT Going"

Garderen, 29 Oktober 2009
