

# VOLVO

UTGIVARE: Volvo Data AB  
Avd 2540 Database Systems

NYHETSBREVET

*DATABasen* 

## Nyhetsbrevet DATABasen - December - 1989

God Jul och Gott Nytt År!



Nu när vintermörkret sänkt sig både över tall och gran och månen vandrar sin tysta ban gäller det att ta vara på ljusglimtarna. Och med sedvanlig blygsamhet kommer här DATABasen för att kasta sin alltid lika avslöjande och penetrerande ljuskägla på några av datavärldens mest invecklade mysterier.

Nåja, skämt åsido, den här gången finns lite information från oss på 2540 om de saker vi har för oss. Vi håller bl a på att göra om två av de verktyg som vi använder, VIP och VINFO. De flesta känner nog till dessa.

Dessutom går vi igenom de nya releaserna av IMS och DB2. Beskrivningen av DB2 är till viss del ganska teknisk, så alla Ni som inte känner er helt familjära på denna nivå av DB2, försök häng med ändå, det finns mycket allmän information också.

Och, hallå där, Du som sysslar med något väsentligt inom databas- eller SU-världen, **skriv ihop något** om detta och bibringa Dina kollegor lite av Din kunskap. Delad vishet är dubbel vishet (gammalt djungelordspråk). Ditt bidrag kan vara allt ifrån

artiklar till kortare frågor som har allmänt intresse. Och kom ihåg, några litterära mästerverk krävs inte, huvudsaken är att det går att läsa.

För 1990 planerar vi följande utgivningstillfällen:  
**W013, W025, W039, W050**

Manusstopp för material att publicera är 14 dagar innan varje utgivning. Skicka ett memo till memoid **VD.DATABAS** och tala om var Ditt bidrag finns, så tar vi in det i nästa nummer.

DATABasen distribueras till dom för oss kända IMS- och DB2-kontakterna. Är det någon som vi missat, sänd ett memo till **VD.DATABAS** så ordnar vi det till nästa gång.

För formens skull påpekas till sist att självklart står varje artikelförfattare för respektive artikels innehåll, vi på 'redaktionen' utövar ingen censur eller kontroll, möjligen lite redigering av insänt material.

Med vänlig hälsning

Henrik Holst / 2540 DATABASE SYSTEMS

## VINFO-omskrivning

### Vad är VINFO?

VINFO är ett system för att hålla reda på relationerna mellan transaktioner och DL/1-databaser, dvs vilka transaktioner som använder sig av vilka DL/1-databaser. Den vanligaste tillämpningen är när man kör procedur VVDBCHNG eller som den också heter VDBCHNG då man inte bara stoppar/startar en DL/1-databas utan också alla transaktioner som hör till den. Denna förändring sparas även som data i VINFO.

### Varför omskrivning?

Bakgrunden till omskrivningen är att VINFO-systemet idag inte fungerar som det skall, fel finns i systemet, och att omvärlden har ändrat sig, så tex. har DB2 tillkommit. Så VINFO är nu i behov av att skrivas om. Detta är ett första utkast till vad som systemet kan tänkas utföra i en framtid. Förslaget har skickats ut för synpunkter, och jag skall sammanställa dessa till ett mer genomarbetat förslag som kan ligga till grund för ett beslut. Är det någon som nu läser detta för första gången och har synpunkter skicka dem gärna till undertecknad.

### Förslaget:

Här nedan följer förslaget i sina viktigaste delar.

### Huvud-funktionerna

- 1 Stoppa/starta transaktioner när en DL1-databas, DB2-tablespace eller annan databas stoppas/startas.
- 2 Om IMS tas/går ner minnas ovansående status vid upptagning för att då sätta rätt status på databaser och transaktioner. Antingen som när IMS togs/gick ner eller med de ändringar som gjorts under tiden IMS har varit nere.

### Databas-typer

Tre databas-typerna bör finnas där man bör kunna uppdatera de två första automatiskt och den sista blir helt manuell.

- 1 DL/1-databaser, uppdatering automatiskt som nu från acblib och stage-info.
- 2 DB2-tablespace (= minsta fysiska enhet, pga detta ej tabell). Uppdatering automatiskt från DB2:s systemtabeller.
- 3 Övriga (tex nya databaser i MVS, om distribuerade databaser kommer och man vill ha register ifrån andra miljöer såsom VM, S/3x, AS/400, PC:s, DEC, etc.). Uppdatering manuellt mellan databas och transaktioner.

### Start/stopp databas

Online-change bör kunna göras antingen per per databas (DL/1, DB2 eller annan) eller per system-nivå. System-nivå är ett antal databaser som man är manuellet grupperat i VINFO.

Databas-status enbart för online-change, dvs ange endast status i online-IMS och **ej** för batch status. Detta var en kvarleva sedan DBENQ-tiden.

Föreslagna kommando.

- 1 DL/1 STA, DBD, DBR och RDO
- 2 DB2 STA och STO
- 3 Annat STA och STO

Gamla kommando (A,P,U) kommer att vara kompatibla såsom de är idag. Sist undrade jag om någon idag använde sig av NOPAR-parametern?

### VINFO-datan

Idag ligger den på en VSAM-fil med stora svårigheter att arbeta med. Var skall den ligga rent fysiskt i framtiden? Mitt förslag är att lägga den i DB2-tabeller för att få enkla sätt att söka i datat. Det finns emellertid en del problem med detta. Vad händer om DB2 är stoppat? Mitt förslag är att köa upp data för uppdatering tills

när DB2 är aktivt igen. Ett annat inte så attraktivt val är ju DL/1-databaser.

### Funktioner

Vilka funktioner skall finnas i fortsättningen? Såklart kommer en del överflödiga tas bort, en del ändras och en del till läggas till. Huvud-ändringen är att flytta info-systemet från IMS till TSO, detta blir möjligt pga DB2.

- 1 VDBCHNG- och VVDBCHNG-procedurer, för online-change.
- 2 Vid upptagning av IMS sätta rätt online-status på databaser och och transaktioner. Antingen som när IMS togs ner plus ändringar under tiden IMS har varit nere. Funktionen blir som idag.
- 3 TSO-funktioner:

- Updatera transaktioner
- Updatera databaser
- Koppla transaktioner till ett system
- Lista cross-referenser
- Lista senaste actions

### Övriga frågor var

- Kunna stoppa enbart en transaktion, skall det ligga här?
- Fel idag:
  - ACTION kan ibland fyllas med samma action
  - DL/1-databaser stoppas inte, pga uppdateringen av VINFO är felaktig
  - Borttag går ej att göra ur VINFO-filen

/Mats Ericson.

---

## VIP blir GENI

VIP är ett IMS-system skrivet i Assembler som hanterar information om transaktioner och program till IMS-generingen. Alla som har kört VIP har säkert lagt märke till att det finns en del brister och att det saknas funktioner som exempelvis crossreferencelistor. Detta skall vi försöka råda bot på med GENI, GENErerings Informa- tion.

GENI kommer att följa SAA och utvecklas i DB2 och REXX. Detta innebär att flexibiliteten ökas och att underhållsarbetet kommer att förenklas. GENI kommer att läggas i TSOVD och för att inte alla skall behöva skaffa sig ett id dit, kommer distribuerade databaser att användas. An-

vändarna kan köra GENI i sina hemmanoder och ändå komma åt databaserna som ligger i VD02. Genom att lägga GENI i TSOVD undviker vi den dubbellagring som finns nu. Däremot kan man inte längre använda ingången till VINFO som finns i VIP, istället kommer man åt VINFO under VSERV.

En del av de nya i GENI: Crossreferencelistor över applikationer, program, transar. Koppling till VIKOS, koppla transkod till ukod. Ta hänsyn till O-miljöer, ex. IMSV1 delas upp i IMSRS, IMSPENT, IMSVBIL, IMSRS...

/Yvonne Sundwall

## IMS versionsbyte

På Volvo Data pågår just nu ett projekt IMS22 för byte av IMS-version. Nuvarande version 1.3 skall migreras till 2.2.

**Omfattning** Målet är att åstadkomma en för användarna så transparent övergång som möjligt, samtidigt som vi åstadkommer en del förbättringar. Nyheter börjar vi utnyttja när systemet visar sig stabilt.

Arbetet med migreringen är omfattande och inkluderar vissa insatser från kundbolagen, för anpassning och framför allt test av ny version.

**Tidplan** De aktiviteter som främst påverkar kundbolagen är planerade enligt:

|                   |                                    |
|-------------------|------------------------------------|
| <b>v50, fre:</b>  | kundtest i IMSV22<br>(SU-systemet) |
| <b>v02, fre:</b>  | kundtest i IMSV12<br>(SU-systemet) |
| <b>v02, helg:</b> | övergång IMSV12                    |
| <b>v03, helg:</b> | övergång IMSV22                    |
| <b>v05, helg:</b> | kundtest i IMSV1                   |
| <b>v06, helg:</b> | kundtest i IMSV2                   |
| <b>v08:</b>       | övergång IMSV1                     |
| <b>v09:</b>       | övergång IMSV2                     |
| <b>v10:</b>       | test + övergång IMSV3,<br>IMSPP    |

**Orsaker till versionsbyte** Fördelarna med ett versionsbyte är främst en förväntad högre tillgänglighet, snabbare svarstider och bättre funktioner.

Man är också på en datacentral tvingad att regelbundet byta versioner av denna typ av programvara, därför att support från leverantör upphör för gamla versioner, underhåll på Volvo Data blir svårare och problem i samspelet med andra produkter uppstår.

Volvo Data ligger relativt sent vad gäller byte av version 1.3. Det är alltså hög tid att förverkliga detta.

**Förbättringar och nyheter** Här följer nu en kortfattad genomgång av de förändringar vi ser som intressantast i IMS 2.2. Ett seminarie för mer detaljerad genomgång är planerad till vecka 06.

Vi delar här in förändringarna i en del främst inriktad på användning av IMS för systemerare, och en del med mer driftsnära aspekter.

### Användarorienterat

**DBRC-förbättringar:** Data Base Recovery Control är nu en mer integrerad del av IMS databasdel, med förbättrad funktionalitet. Förändringarna är väl dock inte enbart till fördel utan kräver både anpassning och utbildning (vi planerar också att ta fram en efterfrågad DBRC-kurs till våren).

Databaser kan nu grupperas och anropas som en grupp vid t.ex. kopiering och rekonstruktion.

DBRC kan användas för att skapa och köra olika jobb eller annat (vad som helst) som man som användare specificerar.

Så kallad skelett-JCL för jobb som DBRC skapar har nytt utseende och all gammal sådan måste anpassas.

Den specifika DBRC-manualen har utgått och DBRC-information är nu insprängd i andra manualer, fr.a. OAG och UTR (se manualer nedan).

### Förbättrad I/O mot databaser

Sequential Buffering är en ny metod att avsevärt snabba upp sekvensiella genomgångar av stora databaser. Kan göra stor nytta för många långa jobb och minska 'the batch window'.

Förbättringen åstadkoms genom att flera block läses i en IO och att detta dessutom görs 'i förväg' medan programmet arbetar. Gäller OSAM-databaser, ej VSAM (ersätter ej helt HSSR, High Speed Sequential Retrieval)

Vidare kan nu bearbetning fortsätta mot databaser som drabbats av I/O-fel. Detta kan fördröja behovet av rekonstruktion och rentav göra detta onödigt om 'retry' när databasen stängs lyckas.

## **Program som anropar stoppade databaser**

Online-program kan nu kontrollera om en databas är tillgänglig för anrop. Program stoppas alltså nu inte om en av dess baser är stoppad.

Kontroll sker med ett antal nya DL/1-anrop, INIT, SETS och ROLS. Programmet får statuskod 'BA' vid anrop av stoppad databas och kan därför självt hantera denna situation med t.ex backout och felutskrift eller svar att viss information ej kan nås.

Om INIT-anrop ej gjorts får programmet ABEND U3033 och transaktionen läggs på en ny inkö, SUSPEND QUEUE. Transkoden kan också bli USTOPPED. Det hela är inte helt enkelt men vi skall utreda och informera mer om detta.

## **Diverse annat**

Dynamisk allokering av databaser är nu möjlig även i batch-jobb. Denna möjlighet skall dock inte användas i onödan eftersom det kan finnas risk för att fel databas används.

De poster IMS skapar på logg är som vanligt i ny version förändrade och program som läser dessa kan därför behöva justeras.

ISAM-datamängder stöds fortfarande men detta upphör efter denna IMS-version. All ISAM bör därför nu konverteras till VSAM.

**Driftorienterat:** Den nya versionen har bra rykte vad gäller performance och tillgänglighet. RAS (Reliability Availability Serviceability) är bättre än i tidigare versioner och detta bådär ju gott.

## **Maskinresurser**

Som vanligt har vissa performance-förbättringar införts. Dessa bör påverka svarstiderna något till det bättre. Däremot är den resursförbrukning (CPU-tid) som påverkar den direkta kostnaden per transaktion inte med säkerhet lägre än tidigare.

Vidare har en hel del kontrollblock och pooler lagts i annan del av minnet så att tidigare minnesbrist avseende 'private area' har eliminerats. Dock på bekostnad av mer reellt minne.

## **Offline dump formatting**

Den tidigare så tunga formateringen av minnesutskriften (dumpen) kan nu ske i efterhand och systemet kan därför återstartas snabbare. Något som förkortar nedgångstiden vid systemfel och därför ökar systemets tillgänglighet. Förutsatt att antal nedgångar ej ökar naturligtvis, men det hoppas och tror vi ej. Även om ett versionsbyte ofta åstadkommer en viss 'svallvåg' av problem efter sig första tiden.

## **Nya IMS-kommandon**

Vissa kommandon har förbättrats, nu ger t.ex. display av transaktionskoder uppgift om aktuellt program och vice versa.

Man kan också begränsa användning av tunga DISPLAY ALL-kommandon.

**Manualer:** Manualuppsättningen för IMS har genomgått en kraftig 'omstrukturering'. Följande IMS 2.2-manualer är/kan vara av intresse för kundbolagen:

**AP** Application Programming, SC26-4178

**CSAP** Application Programming for CICS, SC26-4177

**DBAG** Data Base Administration Guide, SC26-4179

**GIM** General Information, GC26-4180

**MCRM** Messages and Codes Reference Manual, SC26-4174

**MFSUG** Message Format Service Users Guide, SC26-4181

**OAG** Operations Administration Guide, SC26-4323

**OPRM** Operators Reference Manual, SC26-4175

**SAG** System Administration Guide, SC26-4176

**UTR** Utilities Reference, SC26-4173

För att ha DBRC-information samlad rekommenderas också:

**DBRC** IMS/VS Version 2, DBRC Guide and Reference, SC26-4209

Övriga tillgängliga manualer framgår av förteckning på baksidan av alla manualer.

/Perry Lundqvist

---

## **DB2 Version 2 Release 2 nyheter**

### **Distribuerade databaser**

Den mest kända nyheten i DB2 2.2 är väl möjligheten att komma åt data i andra DB2-system än det egna. Detta öppnar helt nya möjligheter i framtiden genom att man kan ha sitt data spritt i flera miljöer om detta är önskvärt.

I framtiden kommer man även att kunna komma åt data som finns på andra databashanterare såsom SQL/DS OS2 Database Manager etc. I denna första DB2-release är det emellertid endast andra DB2-system man kan komma åt.

De möjligheter som finns i DB2 2.2 är :

Läsning från flera DB2-system under ett commit scope, dock endast ett per SQL-sats. Man kan alltså inte jo-ina två tabeller som finns i olika DB2-system.

Uppdatera ett DB2-system inom ett COMMIT-scope. Detta system kan vara det lokala systemet eller ett remote system för applikationer som går under TSO (DSN) eller Call Attach Facility (CAF).

Applikationer som går under IMS eller CICS kan bara uppdatera det lokala systemet, men givetvis läsa från vilket system som helst.

Flera COMMIT-scope kan finnas i ett program, därmed kan flera DB2-system uppdateras i en och samma applikation, bara COMMIT görs emellan uppdateringarna.

All DDL och DCL kan bara göras i det lokala systemet. LOCK TABLE kan bara göras i det lokala systemet.

Vilket DB2-system som avses anges genom att treställiga namn införs: location.authid.name. Treställiga namn kan användas överallt utom i LOAD utility och CREATE SYNONYM. Dessutom måste location i DDL och DCL vara det lokala systemet. Default för location är det lokala systemet. Location name måste vara

unikt för varje DB2-system inom ett nätverk.

Emellertid kan denna hantering göras helt transparent genom att ett nytt begrepp, ALIAS, införs. Ett alias är ett lokalt namn för en

- lokal tabell
- lokal view
- remote tabell
- remote view

Ett alias kan också peka ut en synonym men i så fall löses synonymen ut så att alias pekar direkt på den underliggande tabellen.

Ett alias kan till skillnad från andra objekt skapas utan den tabell som pekas ut existerar. Ingen behörighet behövs för att accessa ett alias utan behörighet måste finnas att komma åt den underliggande tabellen. Eftersom ett ALIAS närsomhelst kan ändras att peka ut en annan LOCATION, om en tabell t.ex. flyttas behöver inte applikationer vara transparenta.

Genom att ALIAS fungerar som "globala synonymer" kan de bli mycket användbara även för icke-distribuerade funktioner, t.ex. som populärnamn på löpnummertabeller mm.

För att skapa ett alias behöver man en särskild behörighet CREATALIASAUTH som är systembunden.

### **Lokal autonomi**

Genom all nödvändig information fortfarande finns lokalt blir de enskilda DB2-systemen helt autonoma. Givetvis blir applikationer som är beroende av åtkomst till distribuerat data lidande vid t.ex. ett linjebortfall, men alla lokala applikationer är oberörda.

Det lokala systemet är helt suveränt när det gäller säkerheten. Det är det lokala systemet som ger behörighet till remoteanvändare. Varje lokalt system bestämmer hur validering ska ske. DB2 kan

skicka kryperade lösenord om säkerhets-systemet stöder detta.

Ett tillägg till GRANT och REVOKE har gjorts för tabell privilegier: PUBLIC AT ALL LOCATIONS som innebär att ALLA har behörighet. PUBLIC innebär att alla i det LOKALA systemet har behörighet.

Kommunikation mellan de olika DB2-systemen sker helt separat från den lokala verksamheten och det lokala systemet kan genom olika prioritering av kommunikationssystemet styra vilken prioritet externa anrop ska ha.

DISPLAY DATABASE kommandot har utvidgats att visa vilken access från ett annat DB2-system som förekommer.  
h2.Program preparering

DCLGEN kan göras på en remote tabell.

En PLAN måste bindas i det DB2-system där den ska exekveras, däremot kan pre-processing ske var som helst. Ev. beroenden till remote object kommer att registreras i SYSIBM.SYSPLANDEP.

Rent tekniskt kommer även statisk SQL som går mot ett remote objekt att exekveras dynamiskt i det systemet och alla ALIAS kommer att lösas upp vid exekveringen inte vid BIND. Däremot kommer accessprivilegier att fungera som vid statiskt SQL, dvs den som binder en PLAN måste vara behörig att göra det som planen ska utföra.

Detta är viktigt vid utvärdering av accessvägar. EXPLAIN kan inte köras remote utan måste göras på det system där tabellen/er finns.

Kommunikationen mellan DB2-system sköts av en särskild DB2-funktion via VTAM. För att kunna optimera överföringen av data via nätet har FETCH satsten utökats med FOR FETCH ONLY. Den kan anges på Read only cursors för att DB2 ska kunna skicka hela block med data över linjen istället för rad för rad. Den tekniken används ändå om DB2 kan avgöra att cursorn är read only.

Slutligen har BIND kommandot utökats med DEFER(PREPARE) som sparar en linjeöverföring vid dynamisk SQL.

### **Användning av distribuerade databaser**

Vid releasebytet till 2.2 kommer inte distribuerade faciliteter att finnas tillgäng-

liga omedelbart utan kommer att införas senare. Emellertid kan man redan nu fundera vilka slags applikationer som kan vara tänkbara.

Några kriterier som bör uppfyllas är :

- Pilotapplikation bör vara liten
- Inte alltför mycket linjetrafik
- Större delen av datat lokalt
- Remote otillgänglighet inte kritisk
- Svarstiden inte kritisk
- Problemet går inte att lösa annars

### **Performanceförbättringar**

Något mindre känt än distribuerade faciliteter är de performanceförbättringar som release 2 har. Dessa är följande:

- Access även via ett index med låg clusterratio
- Flera index kan användas för en accessväg
- Utökad statistik om hur datat i en tabell är fördelat
- Subquery evaluering
- Evaluering av kolumnfunktioner
- Merge Scan Join
- EXPLAIN förbättringar
- Övriga förbättringar

#### **Access via index med låg clusterratio**

Tidigare kunde DB2 göra flera I/O för samma datasida genom att indexet lästes sekventiellt och rad för rad av datat begärdes. I V2R2 finns en ny funktion som heter List Prefetch. Den innebär att de sk. RID (rowid) extraheras ur indexet och sedan sorteras i sekvens innan datasidorna begärs in från Buffer Manager. Detta betyder högst en I/O per rad och i många fall många rader per I/O.

Tekniken kan även användas vid join.

#### **Flera index för en accessväg**

Förr kunde högst ett index per queryblock användas för access till en tabell även om flera lämpliga index fanns. I V2R2 kan flera index användas genom att DB2 läser RID:s med List Prefetch från varje varje index och sorterar dem. Sedan görs intersect för AND predikat och union för OR predikat.

Exempel på uppmätta förbättringar :

Fetch 9851 rader av 10 millioner - 2  
index Elapsed tid 10 ggr förbättring,  
CPU tid 5 gånger

Fetch 11000 rader av 10 millioner - 4  
index Elapsed tid 9 ggr förbättring,  
CPU tid 4 gånger

Resultatet är naturligtvis beroende på det aktuella datat. Detta kan också leda till att index som slagits samman för att få en bra accessväg nu kan hållas isär.

### **Förbättrad statistik över distributionen av värden**

Tidigare antog DB2 alltid att datat var jämnt distribuerat. I V2R2 samlas statistik för de 10 mest frekventa värdena. Detta görs i RUNSTATS för ickeunika enkelkolumnsindex och för första kolumnen av flerkolumnsindex (både unika och ickeunika). Värdena läggs in i SYSIBM.SYSFIELDS tabellen (som också används för FIELD PROCEDURES). I kolumnen EXITPARM lagras nyckelvärdet och i EXITPARML lagras antal rader för den nyckeln i procent.

Denna statistik används sedan vid BIND eller PREPARE. För sk hostvariabler antas dock även i v2r2 att distributionen är jämn, så att dynamisk SQL är den som kommer att tjäna mest på detta.

Exempel på förbättringar upp till 7 gånger i Elapsed tid och 2 gånger i CPU-tid.

### **Utvärdering av subqueries**

För korrelerade subqueries kan värdena från exekveringen av subqueryn sparas i minnet och en speciell sökteknik används för att slippa köra om subqueryn för varje kvalificerad rad i den yttre queryn.

För icke korrelerade subqueries med IN, NOT IN, = ANY,  $\neg$  = ALL och range ( $>$ ,  $>=$ ,  $<$ ,  $<=$ ) predikat har förbättringar gjorts så att de numera oftast kan utvärderas i stage1 istället för stage2 vilket innebär att färre rader skyfflas mellan Data Manager och Relational Data System.

Exempel på förbättringar upp till 4 ggr i Elapsed och CPU-tid.

### **Utvärdering av kolumnfunktioner**

Tre olika förbättringar:

- MAX och MIN. Tidigare gjordes en index scan för följande fråga : SELECT MIN(SALARY) FROM EMP I V2R2 används en teknik som kallas 1-fetch om det finns ett index i rätt ordning på kolumnen, dvs Descending för MAX och ascending för MIN. Dessutom måste ev villkor på kolumner vara =.
- Kolumn funktion materialisation under stage 1 istället för stage 2 om vissa villkor är uppfyllda.
- Om detta inte är möjligt kan kolumnfunktionen materialiseras samtidigt som en sort för t.ex. GROUP BY istället för efter sorten.

Exempel på förbättringar Elapsedtid 3 ggr, CPU-tid 2ggr.

### **Förbättringar vid Merge Scan Join**

Tidigare sorterades alltid rader från in-  
nertabellen. I V2R2 kan sort unvikas om ett index istället kan användas för den inre tabellen. Detta ger då lägre Elapsed och CPU-tid.

### **Utökad EXPLAIN-information**

I V2R2 utökas EXPLAIN-tabellen med tre kolumner. Nya värden kan förekomma i existerande kolumner.

I den nya PREFETCH-kolumnen kan tre värden förekomma

S Sequential Prefetch

L List Prefetch

blank ingen prefetch

I den nya COLUMN\_FN\_EVALUATION kolumnen:

R Utvärdering i Data Manager (Stage 1)

S Utvärdering under sort (stage 2)

blank efter sort (stage2) eller icke applicerbar

I den existerande ACESSTYPE

I1 One fetch vid MAX eller MIN funktionen

N Index access: IN list predicate (samma som i V2R1)

M Multi index scans. Följs av MX, MI eller MU.

MX En av flera index scans

MI Intersection av flera index

MU Union flera index

I och i den nya MIXOPSEQ (Multiple Index Operation Sequence)

1,2..n Sekvenssteg för accesstyp  
MX, MI eller MU

0 för andra accesstyper

### **Övriga performanceförbättringar**

Dessa berör främst transaktionsinitiering och terminering. En gäller auktorisationskontroll där DB2 V2R2 numera håller en lista av upp till 124 användare per plan i EDM-poolen, vilket innebär att katalogaccessen för planer som inte är granade till PUBLIC minskar kraftigt.

Den andra förbättringen är att DB2 kan hoppa över andra delen av Two-phase commit protokollet i IMS och CICS om det bara har förekommit läsningar. Detta reducerar kostnad och elapsed tid för commit.

Förbättrad allokering av minne vilket reducerar tiden för CREATE THREAD och DEALLOCATE THREAD. Detta kommer att märkas mest för lätta transaktioner eftersom en stor del av deras tid går åt för THREAD processing.

Slutligen har tekniken för att validera dattasidor förbättrats.

Exempel på förbättringar i trans-miljö pga ovanstående: Upp till 11% i IMS och 7% i CICS.

### **Manualer**

De nya manualerna för release 2 har redan börjat distribueras:

- GC26-4363 General Information
- SC26-4374 Administration Guide (Obs 3-delad men ett best.nr)
- SC26-4377 Application Programming and SQL Guide
- SC26-4380 SQL Reference
- SC26-4378 Command and Utility Reference
- SC26-4379 Messages and Codes
- SX26-3771 Reference Summary

Dessutom har det kommit en röd manual som behandlar DB2 2.2: GG24-3383-00 DB2 Version 2 Release 2 Design Guidelines for High Performance.

### **Installation mm.**

Den nya releasen beräknas installeras under första kvartalet nästa år, första miljö omkring månadsskiftet januari-februari. Då kommer endast basfunktionerna att installeras. De distribuerade faciliteterna kommer att installeras senare under året och när behovet av dem finns.

Inga speciella åtgärder måste göras för att använda den nya releasen, men för att statiskt bundna planer ska utnyttja flera av de förbättrade accessvägarna krävs rebind.

/Gunnar Bengtsson



## Innehåll

|                         |   |                                   |   |
|-------------------------|---|-----------------------------------|---|
| VINFO-omskrivning ..... | 2 | IMS versionsbyte .....            | 4 |
| VIP blir GENI .....     | 3 | DB2 Version 2 Release 2 nyheter . | 6 |

NILSSON TOMAS

36520

HC2N