



Sandvik Systems Development

Tine Björk

Eva-Maria Mård

Nytt sedan RUG 2005

Sandviks presentation kommer att handla om två saker

- **Fånga förändrat data**
- **Performance projekt**

Här kommer först status kring vissa produkter vi använder, samt annan information

- **CA Platinum status**

- Vi kör version R11
- Tips efterlyses! Vi har problem vid migrering på så sätt att vissa saker ej följer med t.ex. generated always, konstanter, restrict on drop m.m.

- **Application Development Client (Development Center) status**

- Vi kör version 8.1 med fixpack 10
- Med fixpack 10 så funkade det ej att göra drop av SP. Vi har fått en hotfix som löser detta (APAR JR07050). Den kommer att ingå i fixpack 12.

- **Grant av Stored Procedure**

- Tips! Vi har gjort egna paneler i TSO för att granta SP och paket till RACF-grupper (utvecklare utan DBA behörighet har annars problem med behörighet att granta den SP som de har skapat)

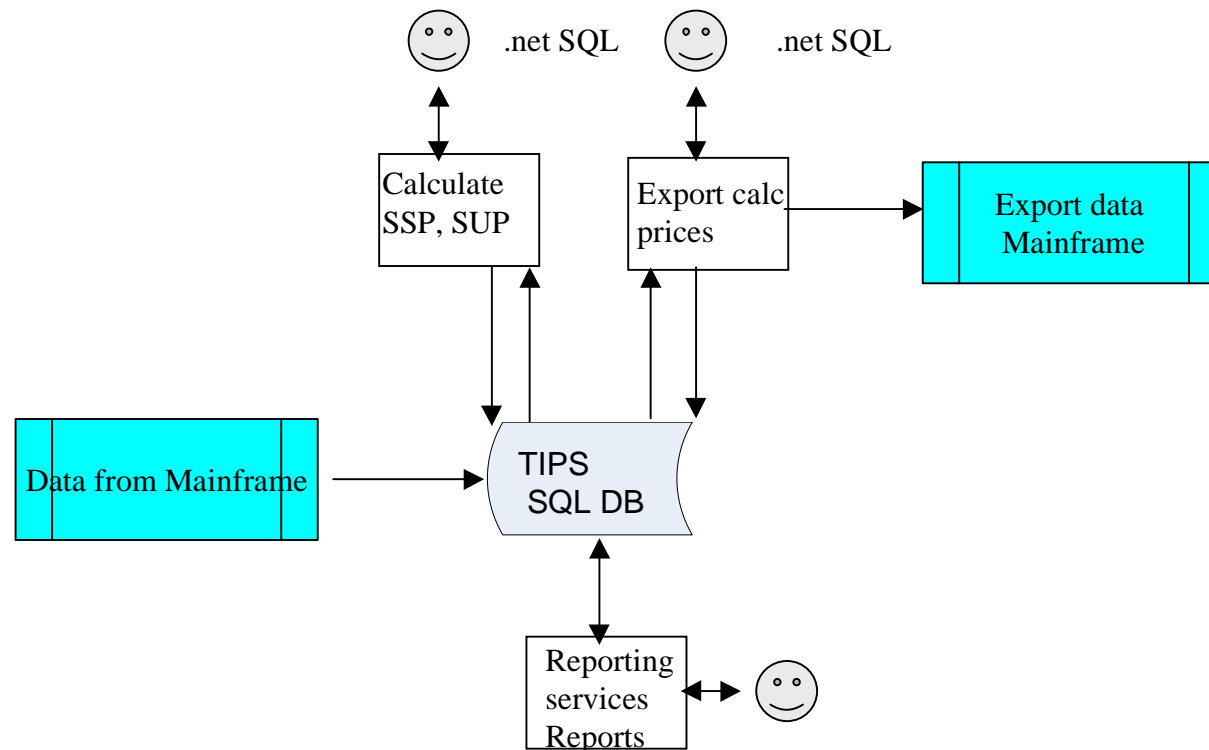
- **Visual Explain v8 status**

- Där har vi haft vissa problem pga att PLAN_TABLE läses med felaktig timestamp i kolumnen BIND_TIME. Det finns rättningar sen december 2005 som löser detta. Vi kör nu version 8.1.0.8.

Fånga förändrat data m.h.a Log Analyzer

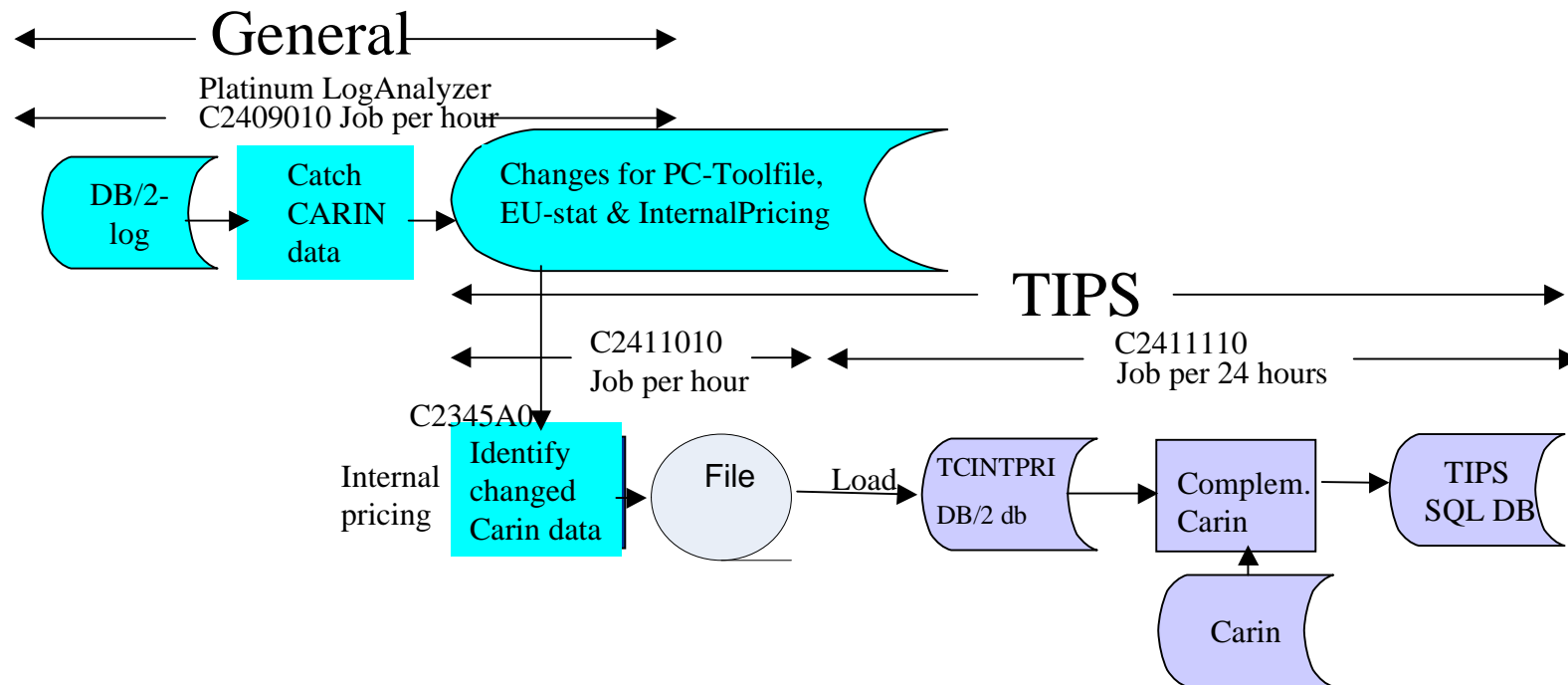
Internal Pricing System, overview

Calculate Supplier Sales Price & Supplier Unit Price



TIPS overview

LogAnalyzer-Refresh data Job set up



Log Analyzer, Strategi

Första steget är att sätta ihop en Strategi i Platinum. Gör man en strategi kan den lätt underhållas. All kriterier sparas i Platimums egna DB2 tabeller.

```
----- r11 ----- Unicenter DB2 Products Main Menu ----- 06-01-22 12:09
OPTION ==>> SCROLL ==>> CSR
```

```
DB2 SSID ==> DB2P LOCATION ==> LOCAL DB2 VERSION : V7R1M
ACM ==> OFF ACMID ==> DS1384 SQLID ==> C20
```

```
<-> Backup and Recovery
    s LA Log Analyzer
    _ MM Merge/Modify
    _ Z Recovery Analyzer
```

```
<-> Database Administration
    _ PX Partition Expert
    _ 1 RC/Query
    _ 2 RC/Migrator
    _ 3 RC/Update
    _ 4 RC/Secure
    _ 5 RC/Extract
```

```
<+> General Facilities
```

```
<-> Report Facility
    _ R Report Facility Menu
```

```
<-> Utilities
    _ U DB2 Object Manager
```

```
<-> Value Pack
    _ B Batch Processor
    _ C DB2 Command Processor
    _ I Interactive SQL
    _ M Alt. Catalog Mapping
    _ TT Thread Term/Dynam DSNZPARM
    _ Y Utility Manager
    _ 6 RI Manager
```

Log Analyzer, Strategi

Strategierna skapas mot varje DB2-subsystem för att lättare kunna underhålla urvalet på tabeller och kolumner.

```
LASTR1 R11      ----- Log Analyzer - Strategy Services ----- 06-01-18 15:05
COMMAND ===>                                         SCROLL ===> CSR
```

```
DB2 SSID ===> DB2P
```

```
Strategy ===> *           Creator ===> DS1384   Log SSID ===> *
```

```
----- DS1384
```

```

O Strategy Description          Creator  S Log  +---- Last Update  +-----+
O SSID User                    Date    Time
_ LOGTIPS LOGGING OF TIPS DATA DS1384  U DB2P DS1384   06-01-18 15:05:33
```

Log Analyzer, tabeller

Här specificeras de tabeller som är med i utsökningen.

SSID: DB2P ----- DS1384

Enter selection criteria to display a table selection list.

Table Name ==> Creator ==> Where ==> N

Enter the desired table names below. The names can include mask characters. Cmd Table Creator Where Apply

___	TCART_____	C20_____	S TIART___	BOTH
___	TCARTAL_____	C20_____	S TIARTAL_	BOTH
___	TCARTSTO_____	C20_____	S TIARTSTO	BOTH
___	TCARTSUB_____	C20_____	S TIARTSUB	BOTH
___	TCGRADE_____	C20_____	S TIGRADE_	BOTH
___	TCLOCART_____	C20_____	S TILOCART	BOTH
___	TCMRKT_____	C20_____	S TIMRKT__	BOTH
___	TCPKPRI_____	C20_____	S TIPKPRI_	BOTH
___	TCPROD_____	C20_____	S TIPROD__	BOTH

Valid Cmds: I - Insert, D - Delete, R - Repeat

Sandvik Systems Development



Log Analyzer, kolumner

Nästa steg är att selektera in vilka kolumner från resp. tabell som loggning skall ske på.

```
Query Name ==> TIARTAL                For Table ==> C20.TCARTAL
Description ==> TCART      TIPS      STAT.                Share ==> N  Default ==> N
Confirm Replace ==> Y                Auto Cast ==> N  Userid - DS1384
TEST COUNT ==> N      VIEW SQL ==> N  SHRINK ==> N    ROW LIMIT ==>
```

```
SEL C ORD COLUMN NAME          TYPE          ----- WHERE CONDITION ----- AND
S    ___  1  PRODID              C(10)
S    ___  2  GRADEID             C(4)
___  ___  3  ALTARTCDPACK         C(35)
S    ___  4  ALTARTCDPREP        C(35)
___  ___  5  CHANGEDATE           DATE
___  ___  6  BRANDCD            C(3)
```

Log Analyzer, Procedur C2409010

En Log Analyzer procedur skapas m.h.a Platinum´s Strategi. Proceduren körs en gång / tim. och ett Easytrieve program sätter om till aktuellt tidsintervall.

```
      BROWSE      PE111.JOBBLIB(C2409010) - 01.42                Line 00000000 Col 001 080
      Command ==>                                           Scroll ==> CSR
*****
***** Top of Data *****
//C2409010 JOB P-324Z08-C202,MSGCLASS=W,MSGLEVEL=(2,1),
//
//          CLASS=S TYPRUN=HOLD
//*****
//*ACCOUNT:      P-324Z08-C202                PLANLIB = UC4 TIMEJOB      *
//*ROUTINE:
//*FREQUENCY:    VARJE TIMME                  *
//*FUNCTION:      UTSÖKNING AV DB2-LOGGEN MED HJÄLP AV LOGANALYZER *
//*LIABLE:       LDB /                      *
//*****
//*
//C2409010 EXEC BATCH,                SKAPA CA-TID RECORD
//          MBR=E192362,FTAG=0180
//*
//DIN1      DD   DSN=PC202.REDO.I2300S(0),DISP=SHR
//DUT1      DD   DSN=&&E1101,
```

Log Analyzer, Procedur forts.

```
      BROWSE      PE111.JOBBLIB(C2409010) - 01.42                Line 00000062 Col 001 080
Command ===>                                           Scroll ===> CSR
.CALL LOG
.DATA
SSID      = (DB2P)
STRATEGY = (DB2P,DS1384,LOGTIPS)
//          DD DSN=&&E1101,DISP=(OLD,PASS)
//          DD *
LOGSRC    = (BSDS)
OBSRC    = (CATALOG)
COPYSRC   = (CATALOG)
DYN SORT  = (DSNUM(12),SPACE(100,100),MAINSIZE(1000),MSG(N))
GENUNIT   = (SYSDA)
RPTLINES  = (60)
S99WAIT   = (YES)
RESOLVUR  = (YES)
WORKLOAD  = (SMALL)
DMLREPT   = (LEVEL      (DETAIL)
              ,ROLLBACK (INCLUDE)
              ,CATALOG  (EXCLUDE)
              ,ORDERBY  (URID)
              ,INCLUDE  (AND
```

Log Analyzer, Output

Outputen från Log Analyzer är Redo sql på en fil. Vi fångar alltså samma sql som redan skett en gång i systemet genom att läsa DB2-loggen.

```
EDIT          XS1384.REDO.D2060113.T163444
Command ====>
002528
002529  UPDATE C20.TCART
002530      SET PRODIDREPL = '0000236560'
002531          , GRADEIDREPL = '0000'
002532          , PRODNONEAN = NULL
002533      WHERE PRODID = '0000205716'
002534          AND GRADEID = '0000'
002535 ;
002536
002537      DELETE FROM C20.TCPROVID
002538      WHERE PRODID = '0000205716'
002539          AND GRADEID = '0000'
002540          AND TOSTOCKID = '4  '
002541          AND SEQNO = '01'
```

Log Analyzer, Proc. C2411010

Ett Cobol program C2345A0 läser sedan filen med Redo sql och avgör om en delete, update eller insert har loggats. Programmet körs en gång i timmen och skapar en ny utfil med ett unikt nytt ID för varje tabell som loggning skett på.

```
003280*** UPDATE-PRODUCT
003281*           003285           IF TESTPOST(2:6)           = 'UPDATE'
003286*
003287*** UPDATE
003288*
003289           MOVE TESTPOST(9:18) TO TABLENAME-WS
003291*
003292           IF TESTPOST(9:15) = 'C20.TCART           '
003293               PERFORM BA-UPDATE-TCART
003294           END-IF
003295*
003296           IF TESTPOST(9:15) = 'C20.TCPROD           '
003297               PERFORM BB-UPDATE-TCPROD
003298           END-IF
003299*
003300           IF TESTPOST(9:15) = 'C20.TCGRADE           '
003301               PERFORM BC-UPDATE-TCGRADE
```

```

004148*  UPDATE TCART
004151      PERFORM XA-READ-INFIL.
004153      IF TESTPOST(4:3) = 'SET'
004154      PERFORM UNTIL TESTPOST(4:5) = 'WHERE'
004156          MOVE 0                      TO CTR
004157          MOVE TESTPOST(8:18) TO REDFALT
004158          INSPECT REDFALT TALLYING CTR FOR CHARACTERS
004159              BEFORE INITIAL ' '
004160          MOVE SPACE                      TO SOKNAMN
004161          MOVE REDFALT(1:CTR) TO SOKNAMN
004162          PERFORM XB-SOK-FALTNAMN
004163*
004164          PERFORM XA-READ-INFIL
004165      END-PERFORM
004166      IF SKAPA-TCINTPRI
004167          IF TESTPOST(10:6) = 'PRODID'
004168              MOVE TESTPOST(20:10) TO PRODID-IPR
004169          END-IF
004170*
004171          PERFORM XA-READ-INFIL
004172*
004173          IF TESTPOST(10:7) = 'GRADEID'
004174              MOVE TESTPOST(21:4) TO GRADEID-IPR
004175          END-IF
004176*
004177          MOVE 'U'                      TO STTMNTCD-IPR
004178          MOVE 'A'                      TO TABLEID-IPR

```

Log Analyzer, Proc. C2411010 forts.

Fältbeskrivning för posterna som skapas från C2345A0.

```
001006*****
001007 01  TCINTPRI-IPR.
001008      10  ROWNO-IPR          PIC S9(9) USAGE COMP.
001009      10  PRODID-IPR        PIC X(10).
001010      10  GRADEID-IPR       PIC X(4).
001011      10  MARKETNO-IPR     PIC X(5).
001012      10  STOCKID-IPR      PIC X(3).
001013      10  SUBCOMPCD-IPR    PIC X(4).
001014      10  PRODUNIT-IPR     PIC X(2).
001015      10  STTMNTCD-IPR    PIC X(1).
001016      10  TABLEID-IPR    PIC X(1).
001017      10  FROMDATE-IPR    PIC X(10).
001018      10  STATPERIOD-IPR   PIC X(6).
001019      10  CURRCDMARKET-IPR PIC X(3).
001020      10  FROMSTOCKID-IPR  PIC X(3).
001021      10  TOSTOCKID-IPR   PIC X(3).
001022      10  TIMESTAMP-IPR   PIC X(26).
001023      10  FILLER1-IPR    PIC X(5).
***** Bottom of Data *****
```

Log Analyzer, Proc. C2411010 forts.

Exempel på utfil som skapats. Filerns körs sedan ihop till en dagfil och filen sorteras på Timestamp asc. En LOAD Replace Utility laddar in datat i en DB2 tabell.

```
EDIT          PC202.C4501S.G0014V00          Columns 00001 00072
Command ==>          Scroll ==> CSR
***** ***** Top of Data *****
000001  00000083100000      7      UI0001-01-01200601      2006-01-20-14
000002  00000082760000      7      UI0001-01-01200601      2006-01-20-14
000003  00000704180296      4      UI0001-01-01200601      2006-01-20-14
000004  00000083100000      7      UI0001-01-01200601      2006-01-20-14
000005  00000554420000      7      UI0001-01-01200601      2006-01-20-14
000006  00000633020187      4      UI0001-01-01200601      2006-01-20-14
000007  00000082900000      4      UI0001-01-01200601      2006-01-20-14
000008  00002276850000980      IF2005-09-02      AUD      2006-01-20-14
000009  00002276850000976      IF2005-09-02      TWD      2006-01-20-14
000010  00002276850000966      IF2006-01-01      SGD      2006-01-20-14
000011  00002276850000960      IF2006-01-01      MYR      2006-01-20-14
```

Load Procedur C2412110

```
BROWSE      PE111.JOBBLIB(C2412110) - 01.22                Line 00000029 Co
Command ==>                                           Scroll
//C2412130 EXEC DSNUPROC,
//              DB2=P,UID='C2412130',UTPROC=' '
//*
//SYSREC00 DD  DSN=PC202.C4601S.DAG(+1),DISP=OLD
//SYSIN      DD  DSN=P0110.KONSTANT(C2412130),DISP=SHR
//*-----
//C2412140 EXEC DSNUPROC,
//              DB2=P,UID='C2412140',UTPROC=' '
//*
//SYSPRINT DD  SYSOUT=*
//UTPRINT  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN      DD  DSN=P0110.KONSTANT(C2412140),DISP=SHR
```

DB2 tabell

```

DB2 Object ==> T
Table Name ==> TCINTPRI
Qualifier ==> *
Option ==> C
Creator ==> C20
Grantor ==> *
Where => N
Loc: LOCAL ----- SSID: DB2U LVL: 01 -XS1384 -- LINE 01 OF 16 >
CMD      TABLE NAME      CREATOR  COLUMN NAME      COLTYPE
-----
          TCINTPRI          C20
          ROWNO            INTEGER
          PRODID           CHAR
          GRADEID          CHAR
          MARKETNO        CHAR
          STOCKID          CHAR
          SUBCOMPCD        CHAR
          PRODUNIT         CHAR
          STTMNTCD         CHAR
          TABLEID         CHAR
          FROMDATE         DATE
          STATPERIOD       CHAR
          CURRCDMARKET     CHAR
          FROMSTOCKID      CHAR
          TOSTOCKID        CHAR
          TIMESTAMP        TIMESTMP
*****
***** BOTTOM OF DATA *****

```



DB2 tabellen´s index

```
CREATE TYPE 2 UNIQUE INDEX C20.PCINTPRI
ON C20.TCINTPRI
(ROWNO          ASC )
CLUSTER
```

Varje tabell har sedan ett skräddarsytt index för performance. Tabell C20.TCARTSTO har Tableid I.

```
CREATE TYPE 2 INDEX C20.XCINTPRI1
ON C20.TCINTPRI
(TABLEID          ASC
,ROWNO           ASC
,PRODID          ASC
,GRADEID        ASC
,STTMNTCD       ASC )
USING STOGROUP GROUPEC20
PRIQTY 480
```

DB2 tabellen's data

Tableid, RowNO, Prodid, Gradeid och Stmntcd.

```
For Table ==> C20.TCINTPRI                      Row number==> 1 OF 2055
Edit Mode ==> C                                  Max Char  ==> 070
SSID: DB2P ----- DS1384
OPT S ROWNO          PRODID      GRADEID STOCKID STTMNTCD TABLEID FROMDATE
-----
11,041,001 0000010121 0000      7      U      I      0001-01-01
11,041,002 0000002911 0000      7      U      I      0001-01-01
11,041,003 0000007270 0000      7      U      I      0001-01-01
11,041,004 0000010187 0006      7      U      I      0001-01-01
11,041,005 0000038846 0006      7      U      I      0001-01-01
11,041,006 0000003994 0070      7      U      I      0001-01-01
11,041,007 0000165479 0071      7      U      I      0001-01-01
11,041,008 0000165260 0000      7      U      I      0001-01-01
11,041,009 0000019490 0088      7      U      I      0001-01-01
11,041,010 0000022585 0094      7      U      I      0001-01-01
11,041,011 0000004510 0000      7      U      I      0001-01-01
11,041,012 0000029884 0128      7      U      I      0001-01-01
11,041,013 0000022936 0100      7      U      I      0001-01-01
```

Sql SP läser tabellen

En Sql SP läser sedan ur datat från DB2-tabellen och kompletterar om det krävs ytterligare data från Carin db.

```
-- Cursors
DECLARE cursorA CURSOR WITH RETURN FOR
  SELECT
    INTPRIS.STTMNTCD, INTPRIS.PRODID, INTPRIS.GRADEID,
    ART.STDLISTCD, ART.STATMAINGRP, ART.STATSUBGRP, ART.SALESACCOUNT,
    ART.STDCATEG,
    ART.PKPERUNIT, ART.PRUNITCD, ART.SUBCOMPCD, ART.BRANDCD
  FROM C20.TCINTPRI AS INTPRIS
  LEFT OUTER JOIN C20.TCART AS ART
  ON      ART.PRODID = INTPRIS.PRODID
  AND     ART.GRADEID = INTPRIS.GRADEID
  WHERE   INTPRIS.TABLEID = 'A'
  AND     (INTPRIS.STTMNTCD = 'D'
  OR      ART.PRODID IS NOT NULL)
  ORDER BY INTPRIS.ROWNO
  WITH UR;
```

RUG 24-25/1 2006

Performance Projektet

- Ett uppdrag kom från ett av våra affärsområden som innebar att man ville minska driftskostnaderna för ett visst system. Man ville börja med att titta på DB2 performance för en viss databas, kallad CARIN, som lagrar artikeldata.
- Direktivet omfattade ett antal punkter. Bl.a. ville man att följande skulle åtgärdas:
 - Skapa nya index där det förbättrar prestandan
 - **Ta bort befintliga index som ej används**
 - Slå samman liknande index
 - Se över lås hantering
 - Se över DB2 parametrar för hela subsystemet
- En begränsning värd att nämna är att vi bara behöver ta hänsyn till att **statisk** SQL används i våra system och mot de aktuella databaserna.
- Vi började med att ta oss an punkt 2 ovan, dvs att ta bort index som ej används. Vi ville skapa en SQL som körs mot DB2 katalogen och som får fram index som ej används, som vi således skulle kunna ta bort.
- Det är den SQL vi skapade för ändamålet som följande sidor kommer att handla om. Det låter som ganska enkelt att få till, men det tog oss ett tag, och vi var benägna att ge upp flera gånger om. Alltså inse att vi måste göra ett program istället.

RUG 24-25/1 2006

Performance Projektet

- Här är våra krav på SQL:en. Den ska visa index, för en vald databas, som:
 - **inte** är definierade som primära index
 - **inte** används av några paket
 - **inte** används för att garantera unikheter dvs har en unique constraint definierad till sig
 - **inte** används internt av DB2 för att hantera RI (Referential Integrity)
- Det var punkt 4 som var en svår nöt att knäcka...det är nämligen så att det **inte syns** i katalogen när DB2 använder ett index **internt**.

Ett exempel när DB2 använder ett index internt, är när en rad i en föräldratabell tas bort, då måste DB2 kolla om det finns rader i barntabeller. Den interna sökningen i barntabellen bör ske via ett index, annars tar det tid.

För att DB2 ska använda index på barntabellen så **måste** indexet i barntabellen ha **exakt samma kolumner** och de måste vara i **exakt samma ordning** som RI kopplingen är definierad.

Indexet får ha fler kolumner än RI kopplingen, men de måste ligga **efteråt**.
- SQL:en fungerar även prestandamässigt. Vi har inte lagt på extra index i katalogen. När vi har kört mot vår katalog så har prestandan varit bra om man kör med antingen enbart CREATOR, eller med CREATOR och DATABAS. Man kan förstås även ha med TABELL som filterare.

Vi har haft CREATOR och DATABAS för att begränsa sökningen.

RUG 24-25/1 2006

Performance Projektet

- SQL:en är gjord med hjälp av 3 selecter med UNION emellan.
- Den första selecten visar index, som stämmer med de tre första kraven, och där indexets första kolumn är lika som första kolumnen i någon RI, och därmed kan vara kandidat för att nyttjas internt, men vid vidare undersökning visar det sig att resten av indexets kolumner ej stämmer med kolumnerna i RI, och därmed kan index tas bort.
- Den andra selecten visar index, som stämmer med de tre första kraven, och där indexets första kolumn ej är lika som någon RI, och därmed kan index tas bort.
- Den tredje selecten visar index, som stämmer med de tre första kraven, och där indexets första kolumn är lika som första kolumnen i någon RI, och därmed kan vara kandidat för att nyttjas internt, och vid vidare undersökning visar det sig att index stämmer med kolumnerna i RI, men det finns andra index som också passar kolumnerna i RI, och därmed kan index tas bort.
- SQL:en körs mot följande katalog tabeller:
 - ✓ SYSINDEXES - innehåller alla index
 - ✓ SYSKEYS - innehåller alla kolummer i indexet
 - ✓ SYSRELS - innehåller alla RI
 - ✓ SYSPACKDEP - innehåller alla beroenden mellan index och paket
 - ✓ SYSTABCONST - innehåller alla unique constraint
 - ✓ (SYSTABLES - innehåller alla tabeller, är med på slutet för att visa totalt antal rader i tabellerna)

RUG 24-25/1 2006

Performance Projektet

- SQL'en finns med som bilaga till denna presentation. Den är komplex, men den har försetts med kommentarer. Här är resultatet av SQL'en, dessa index används inte och kan tas bort:

```
+-----+
! TBCREATOR !   TBNAME   ! IXCREATOR !   IXNAME   !   ANTAL_ROWS   !
+-----+
! C20       ! TCART     ! C20       ! XCART10   !   186549   !
! C20       ! TCARTBEL  ! C20       ! XCARTBE2  !   692703   !
! C20       ! TCARTBEL  ! C20       ! XCARTBE5  !   692703   !
! C20       ! TCARTLOC  ! C20       ! XCARTLOC9 !   25879    !
! C20       ! TCARTSUB  ! C20       ! XCARTSUB3 !  2306102   !
! C20       ! TCCYCLEC  ! C20       ! XCCYCLE1 !           0 !
! C20       ! TCCYCLEC  ! C20       ! XCCYCLE3 !           0 !
! C20       ! TCCYCLEC  ! C20       ! XCCYCLE4 !           0 !
! C20       ! TCGRADE   ! C20       ! XCGRADEC  !           679 !
! C20       ! TCGRADE   ! C20       ! XCGRADE2  !           679 !
! C20       ! TCPRODES  ! C20       ! XCPRODES  !  320355   !
! C20       ! TCPROVID  ! C20       ! XPROVID5  !  1131555  !
! C20       ! TCSTGRPC  ! C20       ! XCSTGRP1  !           4390 !
! C20       ! TCSTGRPC  ! C20       ! XCSTGRP4  !           4390 !
! C20       ! TCSTGRPC  ! C20       ! XCSTGRP5  !           4390 !
! C20       ! TCSTOCKB  ! C20       ! XCSTOCKB1 !  340543   !
! C20       ! TCSTOCKT  ! C20       ! XCSTOCT4  !  1066616  !
+-----+
```

RUG 24-25/1 2006

Performance Projektet

Här har prestandan mätts på SQL'en med hjälp av en Boole&Babbage trace:

```
BMC SOFTWARE ----- DB2 TRACE ENTRIES ----- RX AVAILABLE
SERV ==> LTRAC          INPUT  00:04:45  INTVL=> 3  LOG=> N  TGT==> DB2U
PARM ==> TINE1                               ROW 15 OF 16  SCROLL=> CSR
EXPAND:  MON(WKLD), TOTALS, I/O-DB/TS,        ENTRIES IN BUFFER 1 - 16
          AUTH, CONNECT, PLAN, TIME, LOC, CORR, LINESEL(STRAC)
23JAN06          DB2 APPLICATION TRACE
END TIME      PLAN      AUTHID  CONNECT  ELAPSED  CPU      Å STMTS  GETPAGE  REASON
-----
13:04:43      DSNTEP2  XS1821  BATCH    2,340 ms  750 ms      32    8,296  OK
***** END OF TRACE ENTRIES *****
```

- På nästa sida visas ett egenutvecklat gränssnitt mot DB2's katalog – DB2Info - som vi använder på Sandvik.

Just denna bild ger en samlad bild över en tabell och dess beroenden.

Här kan man t.ex. se en tabell - TCCYCLE - som föregående lista visade hade 3 index som ej används. De indexena är inte primära index, har inte några paket anslutna, har inte unique constraint samt har andra index som täcker upp för intern RI.

RUG 24-25/1 2006

DB2 info - Login - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://wde2devstudio6.is.sandvik.com/db2info/loginFrame.asp

Report: DB2 Tables
 DB2 System: DB2U
 Creator: c20
 Table: TCcycle
 Table Label/Comment:
 Tablespace:
 Database:
 Show max rows: 10

OK

Info about table C20.TCCYCLEC

Number	Creator	Table name	Label	Comment	No of rows	Runstats time	Tablespace name	Database name
1	C20	TCCYCLEC	12.11. Cycle Counting	12.11. Cycle Counting	0	1/2/2006	SCCYCLEC	DC20CARI

Index

Number	Name	Unique rule	Columns in index	Clustering	Clustered	No of package	Unique constraint
1	PCCYCLEC	YES, PRIMARY IX	(PRODID, GRADEID, STOCKID, CYCLECDATE)	YES	NO	3	PRODID
2	XCCYCLE1	NO	(PRODID, GRADEID)	NO	NO	0	
3	XCCYCLE3	NO	(STOCKID)	NO	NO	0	
4	XCCYCLE4	NO	(PRODID, GRADEID, STOCKID)	NO	NO	0	

Parent

Number	Parent table	Constraint name	Columns in constraint	Delete rule
1	C20.TCSTOCKB	TCSTOCKB	(PRODID, GRADEID, STOCKID)	CASCADE

Column

Number	Name	Type	Length	Decimal	Nulls	Label	Comment
1	PRODID	CHAR	10		N	Product Id	Product Id
2	GRADEID	CHAR	4		N	Grade Id	Grade Id
3	STOCKID	CHAR	3		N	Stock	Stock (L01)
4	CYCLECDATE	DATE	4		N	Cycle Counting Date	Cycle Counting Date (L86/07)
5	INVENTDIFF	DECIMAL	7		N	Inventory Difference	Inventory Difference (L85)
6	OLDQTY	DECIMAL	11		N	Old Quantity	Old Quantity
7	NEWQTY	DECIMAL	11		N	New Quantity	New Quantity
8	CYCLECUSER	CHAR	8		N	Cycle Counting User	Cycle Counting User
9	CHANGEDATE	DATE	4		N		

Done Trusted sites

RUG 24-25/1 2006

Performance Projektet

- Hur gick det då för vår databas, CARIN?
- CARIN
 - har 159 tabeller
 - hade 428 index
 - 55 av dessa index upptäckte vi att de ej användes, så de har tagits bort
 - det sparade totalt 5000 cylindrar, till en årlig kostnad av 30.000 kr
 - + den ej mätbara reducerade kostnaden av att DB2 inte behöver uppdatera dessa index
- Sandvik har reducerat driftskostnader med ca 4 milj kr förra året tack vara de totala insatserna som gjorts för att förbättra performance.

RUG 24-25/1 2006

ÖVRIGT

● DB2v8 status

- ✓ DB2v8 uppgradering skulle köras igång efter årsskiftet, men pga att vi har Coupling Facility på maskin med gammal teknik så klarar vi ej av uppgradering till v8.

● Sandviks DataWarehouse strategi

- ✓ Ett affärsområde använder SAP BW som plattform, med SQLServer som DB-motor och med QlikView som gränssnitt. De har också många "öar av info" där man kör QlikView direkt
- ✓ Ett affärsområde använder Microsoft, men har ett centralt DW i Mainframe med DB2, som ska fasas ut och ersättas. För dyrt och för lång utvecklingstid... Man har kuber i Analysis Services och använder Proclarity som analysverktyg
- ✓ Ett affärsområde använder Cognos som rapport och analysverktyg men Sandvik kommer att börja bygga en central lösning nu med Microsoft som plattform
- ✓ Business Objects används, men inga nya projekt/kunder i sikte - ska uppgraderas till senaste version. För komplext och dyrt för att uppskattas.
- ✓ Qlikview har bubblat upp till ytan och används flitigt och är omåttligt populärt. OK att användas som analysgränssnitt - inte som DW-ersättare
- ✓ Integration: Sandvik har tagit fram ett ramverk som vi installerar på sourcesystemen. Det är i Java och vi kan återanvända affärslogik oavsett plattform. Resultat kan produceras i XML, eller vilket format som helst, och sändas per MQ, Qliner eller vad som...till mottagande system. Övervakas av en egen utvecklad minibroker. Det gör att vi på ett relativt enkelt och snabbt sätt kan bygga logiken för att hämta det data vi vill ha till våra DW (eller andra konsumenter av datat) och spara + återanvända alla affärsregler som alltid måste tas fram.