Building a Performance Warehouse on DB2

Robert Andresen
Principal Consultant,
Database Management
Cell: +1 630 254-0168
Robert.Andresen@ca.com
— Identify performance metric sources

— Using Rexx to read DB2
   – Build list of columns by table (optional)
   – Build SELECT statement (optional)
   – Extract metrics as csv file

— Excel considerations
Metric sources

- SYSIBM tables
  - Catalog information
  - Real time statistics

- DB2 monitor(s)
  - Extract and load to database
  - Report functions, write csv directly

- Application tables
  - Co-relate performance vs. business functions
Nice to know

— Application
  — Business functions
  — Business cycles/peaks
  — SLAs

— Systems
  — What resources are in short supply
  — When are resources in short supply
— COLSTATS
— COLUMNS
— DATABASE
— SYSINDEXES
— TABLES
— TABLESPACE
— Etc.

— Join with monitor tables to fill in missing information
1. UPDATESTATSTIME
2. NACTIVE
3. NPAGES
4. EXTENTS
5. LOADRLASTTIME
6. REORGLASTTIME
7. REORGINSERTS
8. REORGDELETS
9. REORGUPDATES
10. REORGUNCLUSTINS
11. REORGDISORGLOB
12. REORGMASSDELETE
13. REORGNEARINDREF
14. REORGFARINDREF
15. STATSLASTTIME
16. STATSINSERTS
17. STATSDELETS
18. STATSUPDATES
19. STATSMASSDELETE
20. COPYLASTTIME
21. COPYUPDATEPAGES
22. COPYCHANGES
23. COPYUPDATELRSN
24. COPYUPDATETIME
25. IBMREQD
26. DBID
27. PSID
28. PARTITION
29. INSTANCE
30. SPACE
31. TOTALROWS
32. DATASIZE
33. UNCOMPRESSEDDATASIZE
34. DBNAME
35. NAME
<table>
<thead>
<tr>
<th></th>
<th>RTS: SYSINDEXSPACESTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>UPDATESTATSTIME</td>
</tr>
<tr>
<td>2.</td>
<td>NLEVELS</td>
</tr>
<tr>
<td>3.</td>
<td>NPAGES</td>
</tr>
<tr>
<td>4.</td>
<td>NLEAF</td>
</tr>
<tr>
<td>5.</td>
<td>NACTIVE</td>
</tr>
<tr>
<td>6.</td>
<td>SPACE</td>
</tr>
<tr>
<td>7.</td>
<td>EXTENTS</td>
</tr>
<tr>
<td>8.</td>
<td>LOADRLASTTIME</td>
</tr>
<tr>
<td>9.</td>
<td>REBUILDLASTTIME</td>
</tr>
<tr>
<td>10.</td>
<td>REORGLASTTIME</td>
</tr>
<tr>
<td>11.</td>
<td>REORGINSERTS</td>
</tr>
<tr>
<td>12.</td>
<td>REORGDELETES</td>
</tr>
<tr>
<td>13.</td>
<td>REORGAPPENDINSERT</td>
</tr>
<tr>
<td>14.</td>
<td>REORGPSEUDODELETES</td>
</tr>
<tr>
<td>15.</td>
<td>REORGMASSDELETE</td>
</tr>
<tr>
<td>16.</td>
<td>REORGLEAFNEAR</td>
</tr>
<tr>
<td>17.</td>
<td>REORGLEAFFAR</td>
</tr>
<tr>
<td>18.</td>
<td>REORGNUMLEVELS</td>
</tr>
<tr>
<td>19.</td>
<td>STATSLASTTIME</td>
</tr>
<tr>
<td>20.</td>
<td>STATSINSERTS</td>
</tr>
<tr>
<td>21.</td>
<td>STATSDELETES</td>
</tr>
<tr>
<td>22.</td>
<td>STATSMASSDELETE</td>
</tr>
<tr>
<td>23.</td>
<td>COPYLASTTIME</td>
</tr>
<tr>
<td>24.</td>
<td>COPYUPDATEDPAGES</td>
</tr>
<tr>
<td>25.</td>
<td>COPYCHANGES</td>
</tr>
<tr>
<td>26.</td>
<td>COPYUPDATERSN</td>
</tr>
<tr>
<td>27.</td>
<td>COPYUPDATETIME</td>
</tr>
<tr>
<td>28.</td>
<td>LASTUSED</td>
</tr>
<tr>
<td>29.</td>
<td>IBMREQD</td>
</tr>
<tr>
<td>30.</td>
<td>DBID</td>
</tr>
<tr>
<td>31.</td>
<td>ISOBID</td>
</tr>
<tr>
<td>32.</td>
<td>PSID</td>
</tr>
<tr>
<td>33.</td>
<td>PARTITION</td>
</tr>
<tr>
<td>34.</td>
<td>INSTANCE</td>
</tr>
<tr>
<td>35.</td>
<td>TOTALENTRIES</td>
</tr>
<tr>
<td>36.</td>
<td>DBNAME</td>
</tr>
<tr>
<td>37.</td>
<td>NAME</td>
</tr>
<tr>
<td>38.</td>
<td>CREATOR</td>
</tr>
<tr>
<td>39.</td>
<td>INDEXSPACE</td>
</tr>
</tbody>
</table>
DB2 Monitors

- Traditional Subsystem monitors
  - Subsystem metrics
  - Thread metrics

- SQL Monitors
  - SQL Metrics
  - Object metrics

- My samples show CA Technologies products, similar functions in other vendor products
DB2 monitor

— Insight requests can write data to output files
  — Write/customize request to write csv data
  — Other vendors have report writer capabilities that may allow the same function
— Insight provides the ability to load SMF data into DB2 tables
  — All or most other vendors do this too
Insight Accounting & Statistics Tables

- **Accounting:**
  - APPLICATION_DAILY
  - APPLICATIONDETAIL
  - APPL_BP_DAILY
  - APPL_BP_DETAIL
  - APPL_DDF_DAILY
  - APPL_DDF_DETAIL
  - APPL_GBP_DAILY
  - APPL_GBP_DETAIL
  - APPL_PGM_DAILY
  - APPL_PGM_DETAIL

- **Statistics:**
  - SUBSYSTEM_DAILY
  - SUBSYSTEMDETAIL
  - SUBSYS_BP_DAILY
  - SUBSYS_BP_DETAIL
  - SUBSYS_DDF_DAILY
  - SUBSYS_DDF_DETAIL
  - SUBSYS_GBP_DAILY
  - SUBSYS_GBP_DETAIL
Detector/Subsystem Analyzer tables

- PDT_STANDARD: One row for each Static or dynamic SQL statement
- PDT_STANTEXT: SQL Text
- PDT_OBJECT: one row for each table, TS, and index accessed
- PDT_DYNAMREQ: Contains 1 row for each Dynamic and Static SQL statement executed that was an exception.
- PDT_DYNAMTXT: Exception SQL Text
- PDT_HOSTVARS: One row for each Host Variable Value for exception SQL.
- PDT_SQLERROR: One row for each static or dynamic SQL error
- PDT_ERRORTXT: SQL error text
- PDT_ERRORVAR: One row for each Host Variable Value for SQL errors
Why Excel?

— Probably already have
  ◆ No additional cost
  ◆ Lotus, OpenOffice are other options
— Relatively easy to use
— Many mathematical functions
— Powerful graphing capabilities
  ◆ PC does this better than mainframe
Disclaimers:

- Rexx code samples provided with no warrantee implied
- Rexx code has limited error checking
- Samples were developed using the CA DB2 monitor, Insight, but the Rexx programs are designed to work with any performance data residing in DB2 tables.
- Your mileage may vary
REXX ACCESS TO DB2
— Optional install step for DB2

— DB2 9.1: Step 15 Create default storage group, define temporary work files, and bind DB2 REXX Language Support
  – DSNTIJTM

— DB2 10: Migration step 22: Prepare dynamic SQL Program: DSNTIJTM
  BIND PLAN(DSNREXX) –
  PKLIST(*.DSNREXX.DSNREXX, -
  *.DSNREXUR.DSNREXX, -
  *.DSNREXCS.DSNREXX, -
  *.DSNREXRS.DSNREXX, -
  *.DSNREXRR.DSNREXX) –
  ACT(REP) ISO(CS) SQLRULES(DB2) –
  ENCODING(EBCDIC)
Overview of process:

- Rexx: Build an inventory of the columns in those tables
- Build a SQL statement
  - Manually
  - Using second Rexx sample
- Rexx: Run the SQL, Create a CSV file
- Download CSV file to workstation
- Open CSV file with spreadsheet
- Subtotals, Reports, Graphs
Rexx to generate Table/Column Inventory

Edit Rexx for your names and products

Exec loops for each table/member pair

◊ Open file output to specified member
◊ Build SQL statement with specified table
◊ Write column info to file
◊ Close cursor
◊ Close & free file

Think of this as a Data Dictionary
SQL, Open File, Prepare statement

```sql
SQLSTMT = "SELECT TBCREATOR, TBNAME, NAME,",
    "COLNO, COLTYPE, LENGTH",
    "FROM SYSIBM.SYSCOLUMNS",
    "WHERE ",
    "TBCREATOR = '" || DB || "' AND",
    "TBNAME LIKE '" || TABLE.INDX || "'",
    "ORDER BY TBNAME, COLNO"

SAY SQLSTMT

"ALLOCATE DDNAME(INVTRY) DSNAME("OUTDSN") SHR REUS"

ADDRESS TSO "SUBCOM DSNREXX" /* HOST CMD ENV AVAILABLE ? */
IF RC <> 0 THEN S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')

ADDRESS DSNREXX "CONNECT" SUBSYS
IF SQLCODE <> 0 THEN CALL SQLCA

ADDRESS DSNREXX "EXECSQL DECLARE C1 CURSOR FOR S1"
IF SQLCODE <> 0 THEN CALL SQLCA

ADDRESS DSNREXX "EXECSQL PREPARE S1 FROM :SQLSTMT"
IF SQLCODE <> 0 THEN CALL SQLCA

ADDRESS DSNREXX "EXECSQL DESCRIBE S1 INTO :OUTSQLDA"
IF SQLCODE <> 0 THEN CALL SQLCA
```
Open cursor, fetch, write, close

ADDRESS DSNREXX "EXECSQL OPEN C1"
IF SQLCODE <> 0 THEN CALL SQLCA

DO UNTIL(SQLCODE <> 0)
  ADDRESS DSNREXX "EXECSQL FETCH C1 USING DESCRIPTOR :OUTSQLDA"
  IF SQLCODE = 0 THEN
    DO
      RECORD = ""
      DO I = 1 TO OUTSQLDA.SQLD
        RECORD = RECORD OUTSQLDA.I.SQLDATA
      END
      PUSH RECORD
      "EXECIO 1 DISKW INVTRY" /* WRITE RECORD */
    END
  END
ADDRESS DSNREXX "DISCONNECT" /* CLOSE CURSOR */
IF SQLCODE <> 0 THEN CALL SQLCA

"EXECIO 0 DISKW INVTRY (FINIS" /* CLOSE OUTPUT FILE */
"FREE DDNAME(INVTRY)"
/* FREE FILE */
END /* LOOP BACK FOR NEXT TABLE */
RETURN
SQL error display

SQLCA:
    SAY "SQLCODE = " SQLCODE
    SAY "SQLSTATE = " SQLSTATE
    SAY "SQLERRMC = " SQLERRMC
EXIT
Batch Rexx Job

- Rexx can be run from TSO if DB2 libraries are allocated to your STEPLIB.

```
//ANDRO16R JOB (10031), 'ANDRO16', CLASS=A, MSGCLASS=X,
// MSGLEVEL=(1,1), REGION=0M, NOTIFY=ANDRO16
//STEP1 EXEC PGM=IKJEFT01, PARM='TABINV'
//STEPLIB DD DISP=SHR, DSN=SYS2.DEMOED.LINKLIB
// DD DISP=SHR, DSN=DB2.DB2810.SDSNLOAD
// DD DISP=SHR, DSN=S81A.PRIVATE.SDSNEXIT
//SYSPRINT DD SYSOUT=* 
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=* 
//SYSEXEC DD DSN=ANDRO16.JCL.CNTL, DISP=SHR
```
Dataset output

EDIT ANDRO16.TABCOL.INVENTRY
Command ===> Row

Name Prompt Size Created Changed
_________ APBPDET *Edited
_________ APPGDET
_________ APPLDET
_________ SUBPDET
_________ SUBSDET

BROWSE ANDRO16.TABCOL.INVENTRY(APBPDET)
Command ===> Line

******************************************************************************* Top of Data **************
INSIGHT APPL_BP_DETAIL BEGIN_DATE_TIME 1 TIMESTMP 10
INSIGHT APPL_BP_DETAIL END_DATE_TIME 2 TIMESTMP 10
INSIGHT APPL_BP_DETAIL SYSTEM_ID 3 CHAR 4
INSIGHT APPL_BP_DETAIL SUBSYSTEM 4 CHAR 4
INSIGHT APPL_BP_DETAIL PLAN_NAME 5 CHAR 8
INSIGHT APPL_BP_DETAIL AUTH_ID 6 VARCHAR 128
INSIGHT APPL_BP_DETAIL PARALLEL_REC_TYPE 7 CHAR 6
INSIGHT APPL_BP_DETAIL CONNECTION_ID 8 CHAR 8
INSIGHT APPL_BP_DETAIL CORRELATION_ID 9 CHAR 12
INSIGHT APPL_BP_DETAIL RELEASE 10 CHAR 4
INSIGHT APPL_BP_DETAIL LOCAL_LOCATION 11 VARCHAR 128
INSIGHT APPL_BP_DETAIL YEAR 12 CHAR 4
INSIGHT APPL_BP_DETAIL MONTH 13 CHAR 2
INSIGHT APPL_BP_DETAIL DAY 14 CHAR 2
INSIGHT APPL_BP_DETAIL HOUR 15 CHAR 2
Rexx builds CSV file

- Read SQL input file
- Build SQL statement
- Build column name array
- Write column names to file, comma separated
- Prepare & issue SQL statement
- Write data to file, comma separated
- Close files
TABCSV Setup

/* REXX */
/* */
ARG MEMBER /* pass member name in via parm */
/* Edit these values if needed */

SUBSYS = "S81A"
INDSN = "DB2.SQL(" || MEMBER || ")"
OUTDSN = "PERF.CSV(" || MEMBER || ")"

say " In:" INDSN
say "Out:" OUTDSN

"ALLOCATE DDNAME(SQLIN) DSNAME("INDSN") SHR REUS"
"ALLOCATE DDNAME(CSVOUT) DSNAME("OUTDSN") SHR REUS"

EOF = "NO"
SQLSTMT = ""
Heading = ""
Rec = 0

Batch Rexx parm:
PARM=(TABCSV,'SUBSDETC')
Call Readin /* Read 1st line of select */

doi = 0 while EOF = "NO"
   PULL RECORD
   PARSE VAR RECORD Word1 Word2

   if Word2 = ""
      then SQLSTMT = SQLSTMT Word1 /* build SQL stmt string */
      else SQLSTMT = SQLSTMT Word1 Word2

   IF Word1 <> "SELECT" &, /* Ignore SQL verbs */
      Word1 <> "AND" &,
      Word1 <> "FROM" &,
      Word1 <> "ORDER" &,
      Word1 <> "WHERE" &,
      Word1 <> "GROUP" &,
      Word1 <> "FETCH" &,
      Word1 <> "OPTIMIZE"
   THEN
      DO
         Heading = heading Word1 /* if we get here assuming */
      END /* it’s a column name */
   END

   Call Readin

end

PUSH Heading
Call Writeout
DO UNTIL(SQLCODE <> 0)
  ADDRESS DSNREXX "EXECSQL FETCH C1 USING DESCRIPTOR :OUTSQLDA"
  IF SQLCODE = 0 THEN
    DO
      RECORD = ""
      DO I = 1 TO OUTSQLDA.SQLD
          RECORD = RECORD OUTSQLDA.I.SQLDATA ','
      END
      PUSH RECORD
      Call Writeout /* WRITE RECORD */
      Rec = Rec + 1
    END
  END
  say Rec "Records written"
  ADDRESS DSNREXX "DISCONNECT" /* CLOSE CURSOR */
  IF SQLCODE <> 0 THEN CALL SQLCA
  "EXECIO 0 DISKW SQLIN (Finis"
  "EXECIO 0 DISKW CSVOUT (Finis"
  "FREE DDNAME(CSVOUT SQLIN)"

  exit 0

READIN:
"EXECIO 1 DISKR SQLIN"
if RC <> 0 THEN EOF = "YES"
return ""
| SYSTEM_ID | SUBSYSTEM | YEAR | MONTH | DAY | HOUR | DAY_OF_WEEK | MSTR_TCB | MSTR_SRB | XE21 | 881A | 2008 | 03 | 16 | 20 | SUN | 0.158171 | 0.097101 | 0.015924 | 0.05 |
|-----------|-----------|------|-------|-----|------|-------------|---------|----------|------|------|------|----|----|----|-----|----------------|-------------|------------|------|------|
| XE21      | 881A      | 2008 | 03    | 16  | 20   | SUN         | 0.166300 | 0.098381 | 0.016218 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 16  | 21   | SUN         | 0.177105 | 0.104755 | 0.017323 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 16  | 21   | SUN         | 0.104590 | 0.105103 | 0.017823 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 16  | 22   | SUN         | 0.183629 | 0.105945 | 0.017965 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 16  | 22   | SUN         | 0.210550 | 0.118448 | 0.019958 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 16  | 23   | SUN         | 0.196856 | 0.111722 | 0.019147 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 16  | 23   | SUN         | 0.205867 | 0.115315 | 0.019653 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 00   | MON         | 0.189299 | 0.110466 | 0.018966 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 00   | MON         | 0.160569 | 0.092935 | 0.016312 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 01   | MON         | 0.162725 | 0.097352 | 0.016892 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 01   | MON         | 0.165585 | 0.099770 | 0.016584 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 02   | MON         | 0.168299 | 0.099283 | 0.017021 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 02   | MON         | 0.170434 | 0.100147 | 0.017250 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 03   | MON         | 0.176614 | 0.101526 | 0.017784 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 03   | MON         | 0.179535 | 0.100839 | 0.018295 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 04   | MON         | 0.164854 | 0.099576 | 0.038500 | 0.06 |
| XE21      | 881A      | 2008 | 03    | 17  | 04   | MON         | 0.155383 | 0.168361 | 0.151435 | 0.09 |
| XE21      | 881A      | 2008 | 03    | 17  | 05   | MON         | 0.157785 | 0.095933 | 0.017047 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 05   | MON         | 0.163102 | 0.099584 | 0.016435 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 06   | MON         | 0.168440 | 0.100439 | 0.017119 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 06   | MON         | 0.160108 | 0.094723 | 0.025512 | 0.06 |
| XE21      | 881A      | 2008 | 03    | 17  | 07   | MON         | 0.170166 | 0.102048 | 0.017787 | 0.05 |
| XE21      | 881A      | 2008 | 03    | 17  | 07   | MON         | 0.179063 | 0.104885 | 0.018052 | 0.06 |
| XE21      | 881A      | 2008 | 03    | 17  | 08   | MON         | 0.158935 | 0.097636 | 0.016678 | 0.05 |
## Sample file allocation

<table>
<thead>
<tr>
<th>Data Set Name . . . .</th>
<th>ANDRO16.PERF.CSV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Data</strong></td>
<td></td>
</tr>
<tr>
<td>Management class . .</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>Storage class . . .</td>
<td>DEMOED</td>
</tr>
<tr>
<td>Volume serial . . .</td>
<td>DEMS09</td>
</tr>
<tr>
<td>Device type . . .</td>
<td>3390</td>
</tr>
<tr>
<td>Data class . . .</td>
<td>DCWRKD</td>
</tr>
<tr>
<td>Organization . . .</td>
<td>PO</td>
</tr>
<tr>
<td>Record format . . .</td>
<td>FB</td>
</tr>
<tr>
<td>Record length . . .</td>
<td>1024</td>
</tr>
<tr>
<td>Block size . . .</td>
<td>24576</td>
</tr>
<tr>
<td>1st extent cylinders:</td>
<td>5</td>
</tr>
<tr>
<td>Secondary cylinders :</td>
<td>2</td>
</tr>
<tr>
<td>Data set name type :</td>
<td>PDS</td>
</tr>
<tr>
<td><strong>Current Allocation</strong></td>
<td></td>
</tr>
<tr>
<td>Allocated cylinders :</td>
<td>29</td>
</tr>
<tr>
<td>Allocated extents :</td>
<td>13</td>
</tr>
<tr>
<td>Maximum dir. blocks :</td>
<td>50</td>
</tr>
<tr>
<td><strong>Current Utilization</strong></td>
<td></td>
</tr>
<tr>
<td>Used cylinders :</td>
<td>28</td>
</tr>
<tr>
<td>Used extents :</td>
<td>13</td>
</tr>
<tr>
<td>Used dir. blocks :</td>
<td>1</td>
</tr>
<tr>
<td>Number of members :</td>
<td>1</td>
</tr>
</tbody>
</table>
Download file to workstation

- Manually with IND$FILE or ftp
- Push the file from the mainframe?
  - Windows doesn’t have an ftp server
  - Shared drive on a server with ftp server?
FTP commands

C:\Documents and Settings\andro16>cd \TradeShow\CMG08\DB2Excel

C:\TradeShow\CMG08\DB2Excel>ftp usi121me
Connected to usi121me.ca.com.
220-FTPD211 IBM FTP CS V1R9 at USI121ME, 18:10:45 on 2008-05-20.
220 Connection will close if idle for more than 8 minutes.
User (usi121me.ca.com:(none)): andro16
331 Send password please.
Password:
230 ANDRO16 is logged on. Working directory is "ANDRO16."
ftp> cd perf.csv
250 The working directory "ANDRO16.PERF.CSV" is a partitioned data set
ftp> ascii
200 Representation type is Ascii NonPrint
ftp> get subsdetc subsdetc.csv
200 Port request OK.
125 Sending data set ANDRO16.PERF.CSV(SUBSDETC) FIXrecfm 1024
250 Transfer completed successfully.
ftp: 10183530 bytes received in 69.53Seconds 37.78Kbytes/sec.
Stupid FTP tricks

- ftp supports PDS files

- Commands:
  - `mget`: Download all members into a directory
  - `mput`: Upload all files in a directory
  - `prompt`: Toggle prompt for each file, on/off

- Can be masked

- Script:
  - `ftp -s:script.txt ftp.server`
  - Need to: `ren * *.csv`
Double Click CSV File….

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>SYSTEM</td>
<td>SUBSYSTEM</td>
<td>YEAR</td>
<td>MONTH</td>
<td>DAY</td>
<td>HOUR</td>
<td>DAY_OF_WEEK</td>
<td>MSTR_TCB</td>
<td>MSTR_SR2</td>
<td>DBM1_TCB</td>
<td>DBM1_SR2</td>
<td>IRLM_TCB</td>
<td>IRLM_SR2</td>
<td>DIST_SR2</td>
</tr>
<tr>
<td>2</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>20</td>
<td>SUN</td>
<td>0.158171</td>
<td>0.097101</td>
<td>0.015924</td>
<td>0.054143</td>
<td>0.000099</td>
<td>0.830697</td>
<td>0.008421</td>
</tr>
<tr>
<td>3</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>20</td>
<td>SUN</td>
<td>0.1663</td>
<td>0.088381</td>
<td>0.016218</td>
<td>0.053443</td>
<td>0.000109</td>
<td>0.837067</td>
<td>0.008133</td>
</tr>
<tr>
<td>4</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>21</td>
<td>SUN</td>
<td>0.177105</td>
<td>0.104755</td>
<td>0.017232</td>
<td>0.054036</td>
<td>0.000109</td>
<td>0.84237</td>
<td>0.008844</td>
</tr>
<tr>
<td>5</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>21</td>
<td>SUN</td>
<td>0.18459</td>
<td>0.105103</td>
<td>0.017823</td>
<td>0.053685</td>
<td>0.000117</td>
<td>0.854789</td>
<td>0.008105</td>
</tr>
<tr>
<td>6</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>22</td>
<td>SUN</td>
<td>0.183629</td>
<td>0.105045</td>
<td>0.017965</td>
<td>0.057512</td>
<td>0.000111</td>
<td>0.857040</td>
<td>0.009596</td>
</tr>
<tr>
<td>7</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>22</td>
<td>SUN</td>
<td>0.21055</td>
<td>0.119442</td>
<td>0.019908</td>
<td>0.058311</td>
<td>0.000112</td>
<td>0.88283</td>
<td>0.012781</td>
</tr>
<tr>
<td>8</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>23</td>
<td>SUN</td>
<td>0.196556</td>
<td>0.111722</td>
<td>0.019147</td>
<td>0.057269</td>
<td>0.000108</td>
<td>0.808952</td>
<td>0.00102</td>
</tr>
<tr>
<td>9</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>16</td>
<td>23</td>
<td>SUN</td>
<td>0.205687</td>
<td>0.115315</td>
<td>0.019653</td>
<td>0.058003</td>
<td>0.000117</td>
<td>0.876082</td>
<td>0.001132</td>
</tr>
<tr>
<td>10</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>0</td>
<td>MON</td>
<td>0.185299</td>
<td>0.110460</td>
<td>0.018966</td>
<td>0.059498</td>
<td>0.000113</td>
<td>0.849106</td>
<td>0.005777</td>
</tr>
<tr>
<td>11</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>0</td>
<td>MON</td>
<td>0.160569</td>
<td>0.092933</td>
<td>0.016312</td>
<td>0.057083</td>
<td>0.000094</td>
<td>0.83143</td>
<td>0.009244</td>
</tr>
<tr>
<td>12</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>1</td>
<td>MON</td>
<td>0.162725</td>
<td>0.097352</td>
<td>0.016892</td>
<td>0.056364</td>
<td>0.000101</td>
<td>0.81898</td>
<td>0.008821</td>
</tr>
<tr>
<td>13</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>1</td>
<td>MON</td>
<td>0.165558</td>
<td>0.09977</td>
<td>0.016584</td>
<td>0.05672</td>
<td>0.000102</td>
<td>0.832807</td>
<td>0.001064</td>
</tr>
<tr>
<td>14</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>2</td>
<td>MON</td>
<td>0.168299</td>
<td>0.099283</td>
<td>0.017021</td>
<td>0.058654</td>
<td>0.000092</td>
<td>0.825367</td>
<td>0.001197</td>
</tr>
<tr>
<td>15</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>2</td>
<td>MON</td>
<td>0.170437</td>
<td>0.100147</td>
<td>0.01725</td>
<td>0.057047</td>
<td>0.000097</td>
<td>0.838987</td>
<td>0.008838</td>
</tr>
<tr>
<td>16</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>3</td>
<td>MON</td>
<td>0.176614</td>
<td>0.101526</td>
<td>0.017784</td>
<td>0.057563</td>
<td>0.000097</td>
<td>0.84068</td>
<td>0.008884</td>
</tr>
<tr>
<td>17</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>3</td>
<td>MON</td>
<td>0.175853</td>
<td>0.100838</td>
<td>0.018285</td>
<td>0.057763</td>
<td>0.000102</td>
<td>0.840877</td>
<td>0.008884</td>
</tr>
<tr>
<td>18</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>4</td>
<td>MON</td>
<td>0.164854</td>
<td>0.098576</td>
<td>0.01862</td>
<td>0.056322</td>
<td>0.000105</td>
<td>0.835248</td>
<td>0.000884</td>
</tr>
<tr>
<td>19</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>4</td>
<td>MON</td>
<td>0.15583</td>
<td>0.168361</td>
<td>0.151435</td>
<td>0.09316</td>
<td>0.000102</td>
<td>0.835248</td>
<td>0.000884</td>
</tr>
<tr>
<td>20</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>5</td>
<td>MON</td>
<td>0.157875</td>
<td>0.099333</td>
<td>0.017047</td>
<td>0.059574</td>
<td>0.000102</td>
<td>0.84068</td>
<td>0.008884</td>
</tr>
<tr>
<td>21</td>
<td>XE21</td>
<td>S81A</td>
<td>2008</td>
<td>3</td>
<td>17</td>
<td>5</td>
<td>MON</td>
<td>0.163102</td>
<td>0.099584</td>
<td>0.016435</td>
<td>0.058483</td>
<td>0.000102</td>
<td>0.835248</td>
<td>0.000884</td>
</tr>
</tbody>
</table>

Viola! Instant spreadsheet. Just add formulas!
Repeatable Reporting

- Use the Import External Data instead of Open dialog
- Update the imported file
- On the Data menu, pick Refresh Data option
- Reports and charts updated with new data
- Files must be same size (rows & columns)
Extract Frequency/Automation

- Run Frequency based on underlying data
  - Daily
  - Weekly
  - Monthly

- Automation
  - Easy to automate on mainframe
  - FTP may be a challenge
    - FTP server - not on Windows workstations
    - Windows server may allow push to a network drive
    - Schedule a pull from Workstation if push from mainframe not possible
REPORTING SCENARIOS
Don’t load entire record

- Focus on salient metrics
- Simply what is on spreadsheet
  ◊ Spreadsheet limit
  ◊ Limit of human comprehension
- Copy output member
- Delete all non-related metrics
- Change order of columns selected (optional)

“Taking a sip from a fire hose”
What to Focus on?

- Which resources are in contention at your shop?
- What are the times of peak contention/use?
- Which resources trigger outages or problems?
- What are your critical business applications?

- The usual suspects:
  - Bufferpool hit ratio
  - CPU Use
  - Response times

- Learning/iterative process
Poor performance causes:

- No Buffer pool Strategy, or strategy not enforced
- Bad SQL
- Inadequate database maintenance

- Source: Data Gathering for DB2 Performance Tuning and Health Checks
  - Jeff M. Sullivan
  - IBM Software Group
  - Share 2010
A buffer pool strategy should...

- At least separate tablespaces and indexes
- Separate highly accessed tablespaces and indexes from less active tablespaces and indexes
- Separate by random versus sequential access
- Separate by size
- Put work tablespaces into their own pool
- Put temporary tablespaces into their own pool
IBM DB2 Bufferpool Suggestion:

- BP0 – DB2 Catalog
- BP1 – Small-sized Reference Tablespaces
- BP2 – Small-sized Reference Indexspaces
- BP3 to BP6, BP8 to BP9 – expansion/isolation for performance bottlenecks in BP1 and BP2
- BP7 – Sort DSNDB07
- BP10 to BP19 – Tablespace buffer pools
- BP10 – Medium-sized Sequential Access
- BP11 – Medium-sized Random Access
- BP12 – Large-sized Sequential Access
- BP13 – Large-sized Random Access
- BP14 to BP19 – expansion/isolation for performance bottlenecks
- BP20 to BP29 – Indexspace buffer pools
- BP20 – Medium-sized Sequential Access
- BP21 – Medium-sized Random Access
- BP22 – Large-sized Sequential Access
- BP23 – Large-sized Random Access
- BP24 to BP29 – expansion/isolation for performance bottlenecks
### I/O type by TS, BP

<table>
<thead>
<tr>
<th>TSNAME</th>
<th>TYPE</th>
<th>BPID</th>
<th>PCTGP</th>
<th>PCTRIO</th>
<th>PCTWIO</th>
<th>PCTFIO</th>
<th>BUFHIT</th>
<th>GETPAGE</th>
<th>SYNC_RIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSDBASE</td>
<td>NSEG</td>
<td>BP8K0</td>
<td>99.2%</td>
<td>99.8%</td>
<td>.0%</td>
<td>.0%</td>
<td>19.9%</td>
<td>8294</td>
<td>135</td>
</tr>
<tr>
<td>SYSRTSTS</td>
<td>SEGM</td>
<td>BP0</td>
<td>.3%</td>
<td>.2%</td>
<td>.0%</td>
<td>.0%</td>
<td>92.3%</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>SYSUSER</td>
<td>NSEG</td>
<td>BP0</td>
<td>.2%</td>
<td>.0%</td>
<td>.0%</td>
<td>.0%</td>
<td>100.0%</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>SYSSEQ</td>
<td>SEGM</td>
<td>BP0</td>
<td>.1%</td>
<td>.0%</td>
<td>.0%</td>
<td>.0%</td>
<td>100.0%</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>SYSSTATS</td>
<td>SEGM</td>
<td>BP16K0</td>
<td>.1%</td>
<td>.0%</td>
<td>.0%</td>
<td>.0%</td>
<td>100.0%</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>SYSDBAUT</td>
<td>NSEG</td>
<td>BP0</td>
<td>.1%</td>
<td>.0%</td>
<td>.0%</td>
<td>.0%</td>
<td>100.0%</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Save Object access metrics to verify Buffer pool strategy not being violated
This data is out of Detector/Subsystem Analyzer, check your vendor’s product documentation for similar metrics
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1 | SSID  | 20 | TB_GETP  |
| 2 | INTERVAL_START  | 21 | TB_GETP_SEQU  |
| 3 | INTERVAL_END  | 22 | TB_GETP_INDX  |
| 4 | RECTYPE  | 23 | TB_GETP_LINK  |
| 5 | PLANNAME  | 24 | TB_INDX_GETP  |
| 6 | PROGRAM  | 25 | TB_INDX_CNT  |
| 7 | PGMTYPE  | 26 | IS_GETP  |
| 8 | SECT#  | 27 | IS_TABL_GETP  |
| 9 | STMT#  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
Find BP Strategy Violators

SELECT B.BPOOL, A.DBNAME, A.TSNAME,
       SUM(A.TB_GETP) AS GETPAGES,
       SUM(A.TB_GETP_SEQU) AS GETPSEQU,
       SUM(A.TB_GETP_INDX) AS GETPINDX,
       SUM(A.TB_GETP_LINK) AS GETPLINK,
       SUM(A.TB_INDX_GETP) AS INDXGETP,
       SUM(A.TB_INDX_CNT) AS INDXCNT,
       SUM(A.IS_GETP) AS ISGETP,
       SUM(A.IS_TABL_GETP) AS TABLGETP
FROM PTI.PDT_OBJECT_115 A, SYSIBM.SYSTABLESPACE B
WHERE A.DBNAME = B.DBNAME
   AND A.TSNAME = B.NAME
GROUP BY B.BPOOL, A.DBNAME, A.TSNAME
ORDER BY B.BPOOL, A.DBNAME, A.TSNAME
## Result

<table>
<thead>
<tr>
<th>BPOOL</th>
<th>DBNAME</th>
<th>TSNAME</th>
<th>GETPAGES</th>
<th>GETPSEQU</th>
<th>GETPINDX</th>
<th>GETPLINK</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSCOPY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSDBAUT</td>
<td>422</td>
<td>0</td>
<td>422</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSEBCDC</td>
<td>36</td>
<td>36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSGPAUT</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSGROUP</td>
<td>55</td>
<td>7</td>
<td>48</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSPKAGE</td>
<td>2116</td>
<td>2088</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSPLAN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BP0</td>
<td>DSNDB06</td>
<td>SYSRTSTS</td>
<td>18</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDXGETP</th>
<th>INDXCNT</th>
<th>ISGETP</th>
<th>TABLGETP</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>178</td>
<td>34</td>
<td>178</td>
<td>422</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>58</td>
<td>7</td>
<td>58</td>
<td>16</td>
</tr>
<tr>
<td>46</td>
<td>11</td>
<td>46</td>
<td>48</td>
</tr>
<tr>
<td>1386</td>
<td>38</td>
<td>1386</td>
<td>28</td>
</tr>
<tr>
<td>52</td>
<td>11</td>
<td>52</td>
<td>0</td>
</tr>
<tr>
<td>88</td>
<td>10</td>
<td>88</td>
<td>18</td>
</tr>
</tbody>
</table>
Bad SQL Examples

- CPU by day of week
  - Identify peak usage times
  - Capacity Planning history
- Buffer pool hit ratios by hour
  - Identify times of contention
  - Look for root cause
- Elapsed time by plan
  - Identify poor performers
Track CPU Use by Day of Week

ANDRO16.TABCOL.INVENTORY(SUBSDETC) - 01.00

================================ Top of Data ********

<table>
<thead>
<tr>
<th>INSIGHT SUBSYSTEM_DETAIL</th>
<th>SYSTEM_ID</th>
<th>3</th>
<th>CHAR 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>SUBSYSTEM</td>
<td>4</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>YEAR</td>
<td>7</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>MONTH</td>
<td>8</td>
<td>CHAR 2</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>DAY</td>
<td>9</td>
<td>CHAR 2</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>HOUR</td>
<td>10</td>
<td>CHAR 2</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>DAY_OF_WEEK</td>
<td>11</td>
<td>CHAR 3</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>MSTR_TCB</td>
<td>12</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>MSTR_SR</td>
<td>13</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>DBM1_TCB</td>
<td>15</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>DBM1_SR</td>
<td>16</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>IRLM_TCB</td>
<td>18</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>IRLM_SR</td>
<td>19</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>DIST_SR</td>
<td>21</td>
<td>DECIMAL 15</td>
</tr>
<tr>
<td>INSIGHT SUBSYSTEM_DETAIL</td>
<td>DIST_TCB</td>
<td>22</td>
<td>DECIMAL 15</td>
</tr>
</tbody>
</table>

SELECT
SYSTEM_ID,
SUBSYSTEM,
YEAR,
MONTH,
DAY,
HOUR,
DAY_OF_WEEK,
MSTR_TCB,
MSTR_SR,
DBM1_TCB,
DBM1_SR,
IRLM_TCB,
IRLM_SR,
DIST_SR,
DIST_TCB
FROM
INSIGHT.SUBSYSTEM_DETAIL

Customized Inventory member

Generated SELECT
### Subtotal on Day of Week

#### Excel Spreadsheets

**Example Data**

<table>
<thead>
<tr>
<th>DAY_OF_WEEK</th>
<th>MSTR_TCB</th>
<th>MSTR_SR</th>
<th>DBM1_TCB</th>
<th>DBM1_SR</th>
<th>IRLM_TCB</th>
<th>IRLM_SR</th>
<th>DIST_SR</th>
<th>DIST_TCB</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRI</td>
<td>70.10288</td>
<td>39.99093</td>
<td>10.706266</td>
<td>30.808349</td>
<td>0.04391</td>
<td>284.947</td>
<td>0.602841</td>
<td>1.580859</td>
</tr>
<tr>
<td>MON</td>
<td>73.91204</td>
<td>43.87631</td>
<td>18.654894</td>
<td>32.830876</td>
<td>0.045608</td>
<td>314.5082</td>
<td>0.573947</td>
<td>1.785976</td>
</tr>
<tr>
<td>SAT</td>
<td>59.28325</td>
<td>32.34347</td>
<td>7.377633</td>
<td>23.803918</td>
<td>0.037879</td>
<td>253.73</td>
<td>0.409213</td>
<td>1.290095</td>
</tr>
<tr>
<td>SUN</td>
<td>45.603764</td>
<td>27.693833</td>
<td>5.005152</td>
<td>16.996587</td>
<td>0.029909</td>
<td>249.7907</td>
<td>0.298373</td>
<td>1.24006</td>
</tr>
<tr>
<td>THU</td>
<td>75.28737</td>
<td>42.045519</td>
<td>11.278816</td>
<td>30.564438</td>
<td>0.047138</td>
<td>305.8034</td>
<td>0.605448</td>
<td>1.69764</td>
</tr>
</tbody>
</table>

**Subtotal Dialog Box**

- **At each change in:** DAY_OF_WEEK
- **Use Function:** Sum
- **Add subtotal to:**
  - MSTR_TCB
  - MSTR_SR
  - DBM1_TCB
  - DBM1_SR
  - IRLM_TCB
  - IRLM_SR

**Additional Options:**
- Replace current subtotals
- Page break between groups
- Summary below data

---

*Ready: subdetc*

Average: 97.44806041
Graph of CPU Use by Day of Week
Let's look at Buffer Pools by hour

EDIT ANDRO16.TABCOL.INVENTORY(APBPDETH) - 01.00
Command ===>

****** ******************************************* Top of Data **************
000001  INSIGHT APPL_BP_DETAIL YEAR 12 CHAR     4
000002  INSIGHT APPL_BP_DETAIL MONTH 13 CHAR     2
000003  INSIGHT APPL_BP_DETAIL DAY 14 CHAR      2
000004  INSIGN APPL_BP_DETAIL HOUR 15 CHAR      2
000005  INSIGHT APPL_BP_DETAIL BUF_POOL_NAME 18 CHAR     8
000006  INSIGHT APPL_BP_DETAIL BP_GETPAGE 19 DECIMAL  15
000007  INSIGHT APPL_BP_DETAIL BP_SYNC_READ_IO 20 DECIMAL  15

SELECT
YEAR ,MONTH ,DAY ,HOUR ,BUF_POOL_NAME ,BP_GETPAGE ,BP_SYNC_READ_IO
FROM   INSIGHT.APPL_BP_DETAIL
WHERE   BUF_POOL_NAME = 'BP0'
   AND DATE(END_DATE_TIME) = '2008-04-29'

In: DB2.SQL(APBPDETH)
Out: PERF.CSV(APBPDETH)

SELECT YEAR ,MONTH ,DAY ,HOUR ,BUF_POOL_NAME ,BP_GETPAGE ,BP_SYNC_READ_IO FROM
   WHERE BUF_POOL_NAME = 'BP0'
2237 Records written
READY
END
### Open CSV File:

![Microsoft Excel spreadsheet](image)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>MONTH</td>
<td>DAY</td>
<td>HOUR</td>
<td>BUF_POOL</td>
<td>BP_GETPAGE</td>
<td>BP_SYNC_READ</td>
<td>IO</td>
</tr>
<tr>
<td>2</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>8</td>
<td>BP0</td>
<td>77</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>8</td>
<td>BP0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>45</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>52</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>70</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>2008</td>
<td>4</td>
<td>29</td>
<td>9</td>
<td>BP0</td>
<td>32</td>
<td>0</td>
</tr>
</tbody>
</table>
Hit Ratio Calculation

Buffer Pool Hit Ratio = \( \frac{(\text{Getpages} - \text{ReadIO})}{\text{Getpages}} \)
Let's look at elapsed time by plan

- Picked columns
- Generated SQL
- Edit generated SQL to add an ORDER BY
  
  ```sql
  SELECT PLAN_NAME, APPL_ELAPSED, APPL_TCB, DB2_ELAPSED, DB2_TCB, DB2_IO_ELAPSED
  FROM INSIGHT.APPLICATION_DETAILZ
  ORDER BY PLAN_NAME
  ```
### Avg Elapsed, Total TCB

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>DB2C1150 Total</td>
<td>0.017255</td>
<td>0.015577</td>
<td>0.012247</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DB2C1150 Average</td>
<td>0.5012906</td>
<td>0.405267</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1038</td>
<td>DGL11150 Total</td>
<td>6.62154</td>
<td>6.224319</td>
<td>2.124699</td>
<td></td>
</tr>
<tr>
<td>1039</td>
<td>DGL11150 Average</td>
<td>0.07961461</td>
<td>0.032024488</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1045</td>
<td>DISTSERV Total</td>
<td>0.407215</td>
<td>0.334014</td>
<td>1.079495</td>
<td></td>
</tr>
<tr>
<td>1046</td>
<td>DISTSERV Average</td>
<td>132.3854786</td>
<td>7.9586384</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1063</td>
<td>DSNUTIL Total</td>
<td>2.929425</td>
<td>0.40226</td>
<td>0.55417</td>
<td></td>
</tr>
<tr>
<td>1064</td>
<td>DSNUTIL Average</td>
<td>3.502734813</td>
<td>2.890242125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1066</td>
<td>IDBS1150 Total</td>
<td>0.046814</td>
<td>0.042253</td>
<td>0.37171</td>
<td></td>
</tr>
<tr>
<td>1067</td>
<td>IDBS1150 Average</td>
<td>6.248416</td>
<td>6.242729</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1081</td>
<td>IQPI1150 Total</td>
<td>0.050513</td>
<td>0.045984</td>
<td>0.021916</td>
<td></td>
</tr>
<tr>
<td>1082</td>
<td>IQPI1150 Average</td>
<td>33.78017423</td>
<td>0.337893154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1087</td>
<td>PDTA1150 Total</td>
<td>0.010985</td>
<td>0.010784</td>
<td>0.136695</td>
<td></td>
</tr>
<tr>
<td>1088</td>
<td>PDTA1150 Average</td>
<td>0.2725935</td>
<td>0.2725405</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1092</td>
<td>PDB11150 Total</td>
<td>0.000252</td>
<td>0.00013</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1093</td>
<td>PDB11150 Average</td>
<td>0.000142</td>
<td>9.76667E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1098</td>
<td>PPAD1150 Total</td>
<td>0.028702</td>
<td>0.025992</td>
<td>0.159079</td>
<td></td>
</tr>
<tr>
<td>1099</td>
<td>PPAD1150 Average</td>
<td>0.72602875</td>
<td>0.68890775</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1113</td>
<td>PPAM11150 Total</td>
<td>0.651416</td>
<td>0.628242</td>
<td>2.291807</td>
<td></td>
</tr>
</tbody>
</table>
## Calculate % of Total

![Excel spreadsheet](image)

<table>
<thead>
<tr>
<th>PLAN_NAME</th>
<th>APPL_ELAPSED</th>
<th>APPL_TCB</th>
<th>DB2_ELAPSED</th>
<th>DB2_TCB</th>
<th>% of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2C1150 Total</td>
<td>0.017255</td>
<td></td>
<td>0.015577</td>
<td>0.105%</td>
<td></td>
</tr>
<tr>
<td>DDL11150 Total</td>
<td>6.62154</td>
<td></td>
<td>6.224319</td>
<td>41.59%</td>
<td></td>
</tr>
<tr>
<td>DISTRERV Total</td>
<td>0.407215</td>
<td></td>
<td>0.334014</td>
<td>2.23%</td>
<td></td>
</tr>
<tr>
<td>DSNUTIL Total</td>
<td>2.929425</td>
<td></td>
<td>0.40226</td>
<td>2.69%</td>
<td></td>
</tr>
<tr>
<td>IDBS1150 Total</td>
<td>0.046814</td>
<td></td>
<td>0.042253</td>
<td>0.28%</td>
<td></td>
</tr>
<tr>
<td>IQPI1150 Total</td>
<td>0.050513</td>
<td></td>
<td>0.045984</td>
<td>0.31%</td>
<td></td>
</tr>
<tr>
<td>PDTA1150 Total</td>
<td>0.010985</td>
<td></td>
<td>0.010784</td>
<td>0.07%</td>
<td></td>
</tr>
<tr>
<td>PDTB1150 Total</td>
<td>0.000252</td>
<td></td>
<td>0.00013</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>PPAD1150 Total</td>
<td>0.028702</td>
<td></td>
<td>0.026992</td>
<td>0.18%</td>
<td></td>
</tr>
<tr>
<td>PPAO1150 Total</td>
<td>0.65146</td>
<td></td>
<td>0.628242</td>
<td>4.20%</td>
<td></td>
</tr>
<tr>
<td>PSAA1150 Total</td>
<td>0.262489</td>
<td></td>
<td>0.118112</td>
<td>0.79%</td>
<td></td>
</tr>
<tr>
<td>PTT81150 Total</td>
<td>0.000352</td>
<td></td>
<td>0.000193</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>RAPA1150 Total</td>
<td>1.813037</td>
<td></td>
<td>1.71181</td>
<td>11.44%</td>
<td></td>
</tr>
<tr>
<td>RAPB1150 Total</td>
<td>0.008831</td>
<td></td>
<td>0.007951</td>
<td>0.05%</td>
<td></td>
</tr>
<tr>
<td>RAPS1150 Total</td>
<td>0.010951</td>
<td></td>
<td>0.036457</td>
<td>3.72%</td>
<td></td>
</tr>
<tr>
<td>RBPA1150 Total</td>
<td>0.46679</td>
<td></td>
<td>0.456288</td>
<td>3.05%</td>
<td></td>
</tr>
<tr>
<td>RCMA1150 Total</td>
<td>1.005759</td>
<td></td>
<td>0.70231</td>
<td>4.69%</td>
<td></td>
</tr>
<tr>
<td>RCMO1150 Total</td>
<td>0.123496</td>
<td></td>
<td>0.117296</td>
<td>0.78%</td>
<td></td>
</tr>
<tr>
<td>RCMX1150 Total</td>
<td>0.009673</td>
<td></td>
<td>0.008728</td>
<td>0.06%</td>
<td></td>
</tr>
<tr>
<td>RCUS1150 Total</td>
<td>0.021998</td>
<td></td>
<td>0.020017</td>
<td>0.13%</td>
<td></td>
</tr>
<tr>
<td>RCUU1150 Total</td>
<td>1.845208</td>
<td></td>
<td>1.801916</td>
<td>12.04%</td>
<td></td>
</tr>
<tr>
<td>RDPC1150 Total</td>
<td>0.049676</td>
<td></td>
<td>0.048458</td>
<td>0.32%</td>
<td></td>
</tr>
</tbody>
</table>
SOME DETECTOR SAMPLES
SELECT DISTINCT  A.INTERVAL_START,  A.RECTYPE,  A.PLANNNAME,  
A.PROGRAM,  A.PGMTYPE,  A.SECT#,  A.STMT#,  A.SQL_CALL,  
A.DYN_USE_COUNT,  A.DYN_TEXT_TOKEN,  A.DSGROUP,  
A.CONN_TYPE,A.CONN_NAME,  A.CORRID,  A.LOCATION,  A.USERID,  
A.END_USER_ID, A.TRANSACTION_ID,  A.WORKSTATION_ID,  B.SEQNO,  
B.SQL_TEXT  
FROM PTI.PDT_STANDARD_115  A, PTI.PDT_STANTEXT_115  B  
WHERE  
( B.PLANNNAME = A.PLANNNAME AND  B.PROGRAM = A.PROGRAM AND  
B.COLLID = A.COLLID AND  
B.SECT# = A.SECT# AND  
B.STMT# = A.STMT# AND  
A.DYN_TEXT_TOKEN = B.DYN_TEXT_TOKEN)  
AND ( B.SQL_TEXT <> '  
AND  A.DYN_USE_COUNT >&CNT)  
AND USERID = ???????  
ORDER BY  A.DYN_TEXT_TOKEN,  A.SECT#,  A.STMT#
Index utilization -

SELECT MAX( B.NAME), MAX( B.CREATOR), MAX( B.CLUSTERED), SUM(A.IS_GETP), SUM( A.IS_TABL_GETP) FROM PTI.PDT_OBJECT_115 A, SYSIBM.SYSINDEXES B WHERE ( A.ISNAME = B.INDEXSPACE) AND ( A.DBNAME = '&DBNAME' AND A.SQL_CALL NOT IN ('INSERT', 'DELETE') ) GROUP BY A.ISNAME HAVING SUM( A.IS_GETP) >=0 ORDER BY 4 DESC
Indexes not used

```
SELECT (SUBSTR(CREATOR,1,8)),
   (SUBSTR(NAME,1,28)),
   'NO SELECT INDEX IO'
FROM SYSIBM.SYSINDEXES
WHERE DBNAME='PTDB'
AND   NAME NOT IN
(SELECT  MAX ( B.NAME )
 FROM  PTI.PDT_OBJECT_115 A,
      SYSIBM.SYSINDEXES B
WHERE ( A.ISNAME = B.INDEXSPACE )
   AND A.DBNAME='PTDB'
GROUP BY A.ISNAME
HAVING SUM ( A.IS_GETP ) > 0
   AND MAX ( A.SQL_CALL ) IN ( 'PREPARE' , 'SELECT' , 'OPEN' ))
;
```

Note Join with SYSIBM tables
Most CPU resource intensive statements in a specific interval for specific COLLECTION-ID’s:

```
SELECT COLLID,
    PROGRAM,
    STMT,
    SQL_CALL,
    DECIMAL(SUM(INDB2_TIME), 6, 1) AS ELAPSE_SECONDS,
    DECIMAL(SUM(INDB2_CPU), 6, 1) AS CPU_SECONDS,
    STRIP(DIGITS(SUM(SQL_CALLS)), L, '0') AS SQL_CALLS,
    STRIP(DIGITS(SUM(GETPAGE)), L, '0') AS GETPAGE
FROM   PDT_STANDARD
WHERE  SQL_CALL NOT LIKE 'CALL%'
AND    RECTYPE = 'STMT'
AND    STMT# > 0
AND    COLLID like 'M5P1%'
AND    INTERVAL_START between '2005-12-12-06.00.00.000000' and '2005-12-19-10.00.00.000000'
GROUP BY COLLID, PROGRAM, STMT, SQL_CALL
ORDER BY 6 DESC
FETCH FIRST 1000 ROWS ONLY
```
SELECT PROGRAM, STMT#, SQL_CALL,
  DECIMAL(SUM(INDB2_TIME)/60, 5, 1) AS ELAPSE_MINUTES,
  DECIMAL(SUM(INDB2_CPU)/60, 5, 1) AS CPU_MINUTES,
  DECIMAL(SUM(TOTAL_WAITTIM)/60, 5, 1) AS WAIT_MINUTES,
  (DECIMAL(SUM(INDB2_TIME)/60, 5, 1)-
   DECIMAL(SUM(INDB2_CPU)/60, 5, 1) -
   DECIMAL(SUM(TOTAL_WAITTIM)/60, 5, 1)) AS NOT_ACC_MINUTES,
  STRIP(DIGITS(SUM(SQL_CALLS)), L, '0') AS SQL_CALLS,
  STRIP(DIGITS(SUM(GETPAGE)), L, '0') AS GETPAGE
FROM PDT_STANDARD
WHERE SQL_CALL NOT LIKE 'CALL%'
AND STMT# > 0
GROUP BY STMT#, PROGRAM, SQL_CALL
ORDER BY 4 DESC
FETCH FIRST 100 ROWS ONLY;
Questions?
Resources Available at ca.com/db

- CA Catalog Poster for DB2 9
- CA Reference Guide for DB2 9
- CA Performance Handbook for DB2
- Data Access Paths for DB2 Flash