

SQL Performance Analyzer

Version 3 Release 1

PTF level 3103

----- DB2 Primary Option Menu -----

OPTION ==> 3

I B M

- 1 DB2I - Database 2 Interactive Functions Ver. 8.1.0
- 2 QMF - Query Management Facility & SQL PA Ver. 8.1.0
- 3 SQL PA - SQL Performance Analyzer Ver. 3.1.0
- 4 EEE - DB2 Easy Explain Functions Ver. 8.2.0

- X EXIT

The TSO interface starts with the main selection panel (ANL@DSN above), normally initiated by running the ANLI clist from ISPF panel 6.

SQL PA ----- SQL Performance Analyzer ----- 14:06

Licensed Materials - Property of IBM Corporation - 5697-I04

(C) Copyright IBM Corporation 2000, 2005 All Rights Reserved
(C) Copyright Innovative Management Solutions, Inc 1993, 2005
All Rights Reserved.

US Government Users Restricted Rights -
Use, Duplication or Disclosure restricted by GSA ADP
Schedule Contract with IBM Corporation.

Welcome to the SQL Performance Analyzer for DB2

Version 3 Release 1

Please press the ENTER key to continue processing

The redesigned logo panel is for accessibility (screen readers)

SQL PA ----- Main Control Panel ----- 14:08

COMMAND ==>

DBRM member pattern.... ==> +OFF+

*Input data set name... ==> INPUT.SQL(WHOME)

Edit input data set... ==> YES

Save edited changes.... ==> YES

Define output files... ==> NO

Go to review reports... ==> NO

Show Current Parm.... ==> NO

Reset 'Thresholds' ==> NO

Reset 'User' parms.... ==> NO

Reset 'System' parms... ==> NO

Process Retro Explain ==> NO

View Help Tutorial..... ==> NO

Compare Old/New Plans ==> NO

Go to Easy Explain..... ==> NO

Modify Statistics..... ==> NO

Go to What If Analysis ==> NO

>>> Press PF3 or PF12 to exit SQL PA, or press Enter to continue <<<

The new main control panel has many new functions in V3, including Retro Explain, Compare Old and New plans, What If? Catalog statistics modification and wholesale statistics migration

SQL PA ----- Show Parameters Panel ----- 14:14
COMMAND ===>

<-----> Current User Parameters <----->				<-----> Thresholds <----->	
REPORTS	ALL	QUALIFY	TDT690	MESSAGE	YES
ADVISOR	ALL	USEPLAN	TDT690	NOSTATS	YES
PRECISE	ALL	CONNECT	CAF	NOTSCAN	NO
DEGREES	ANY	DELIMIT	+OFF+	NONONIX	NO
REFRESH	ANY	NLSCODE	+OFF+	TSCANPG	50000
SHOWALT	YES	RETCODE	NO	ISCANPG	1000
KEEPPLAN	NO	DBTRACE	YES	NONIXPG	5000
OBJECTS	YES				
<-----> Current System Parameters <----->				INLISTS	10
TARGET HOST(ANLSAMP)		NEWSTOR	+OFF+	JOINTAB	10
STORAGE	ESS800	NEWSEEK	0.0120	ALLPART	YES
VIADRDA	+OFF+	NEWROTA	0.0071	IXUPDAT	10
VERSION	V8R1	NEWXFER	4200.0		
BUFFHIT	0	DSGROUP	0	MATCOLS	0.50
HPOOLRD	0	DATASHR	0	PREDPCT	0.15
PROCESS	+OFF+ (ASIS)	PROCESS	+OFF+ (NOSEQ)		

>>> Press PF3 or PF12 to return to main panel for any changes <<<

The Show Parameters panel maps all of the current parameter names and values on a new panel in 3 categories (in prior versions it was on the bottom of the Main panel)

SQL PA ----- Output Reports Panel ----- 14:23
COMMAND ===>

Output cost summary file ===: TDT690.ANLCOST.LOG (Required name)

Output SQL, objects file ===: TDT690.ANLOUT.SQL (Required name)

*Output explain plan file ===> ANLREP.RPT

*Output detail trace file ===> QTRACE.RPT

*Output query limits file ===> QLIMIT.RPT

Output file dispositions ===> OVERLAY (OVERLAY or APPEND)

>>> Press PF3 or Enter to return to Main Panel, or PF12 to Exit <<<

Output reports are selected here, with the required cost log (ANLCOST.LOG) and SQL/Objects file (ANLOUT.SQL) using "standard" names; all files are subject to the same report disposition (overlay or append)

SQL PA ----- Change User Parameters ----- 14:37

COMMAND ==> EXPAND

Reports to be created	==> ALL	(REPORTS EXP ALL)
Set SQL advisor level	==> ALL	(ADVISOR NO YES ALL)
Use optimizer estimate	==> ALL	(PRECISE NO YES ALL)
Allow parallel degrees	==> ANY	(DEGREES ANY ONE 1)
Consider MQTs in plan	==> ANY	(REFRESH ANY YES NO)
Show alternate indexes	==> YES	(SHOWALT NO YES)
Show DB2 Objects used	==> YES	(OBJECTS NO YES)
*Default qualifier name	==> TDT690	(QUALIFY authid) >
Owner of explain tables	==> TDT690	(USEPLAN authid)
Attach overhead assumed	==> CAF	(CONNECT NONE CAF CICS ...)
Delimiter quote (Cobol)	==> +OFF+	(DELIMIT +OFF+ QUOTE)
National language used	==> +OFF+	(NLSCODE +OFF+ KOR JPN ...)
Return code = Msg level	==> NO	(RETCODE NO YES)
Preserve plan records	==> NO	(KEEPLAN NO YES)
Turn on the Diagnostics	==> YES	(DBTRACE NO YES ALL DMP)

>>> Press PF3 to Return to Main Menu, or PF12 to Exit SQL PA <<<

This is the change "USER" parameters panel. Type 'expand' with cursor on QUALIFIER field to expand the name field (the field scrolls)

QUAL+0

Line 1 of 2

Command ==>

Scroll ==> PAGE

TDT690_IS_AN_EXAMPLE_OF_A_VERY_LONG_QUALIFIER_NAME

User can enter a long QUALIFIER name (up to 128 chars) if desired in V8 NFM

SQL PA ----- Change User Parameters ----- 14:37

COMMAND ==> EXPAND

Reports to be created	==> ALL	(REPORTS EXP ALL)
Set SQL advisor level	==> ALL	(ADVISOR NO YES ALL)
Use optimizer estimate	==> ALL	(PRECISE NO YES ALL)
Allow parallel degrees	==> ANY	(DEGREES ANY ONE 1)
Consider MQTs in plan	==> ANY	(REFRESH ANY YES NO)
Show alternate indexes	==> YES	(SHOWALT NO YES)
Show DB2 Objects used	==> YES	(OBJECTS NO YES)
*Default qualifier name	==> VERY_LONG	(QUALIFY authid) <>
Owner of explain tables	==> TDT690	(USEPLAN authid)
Attach overhead assumed	==> CAF	(CONNECT NONE CAF CICS ...)
Delimiter quote (Cobol)	==> +OFF+	(DELIMIT +OFF+ QUOTE)
National language used	==> +OFF+	(NLSCODE +OFF+ KOR JPN ...)
Return code = Msg level	==> NO	(RETCODE NO YES)
Preserve plan records	==> NO	(KEEPLAN NO YES)
Turn on the Diagnostics	==> YES	(DBTRACE NO YES ALL DMP)

>>> Press PF3 to Return to Main Menu, or PF12 to Exit SQL PA <<<

The name of QUALIFIER can be scrolled using the F10 and F11 keys (note the <> scroll bar)

SQL PA ----- Change System Thresholds ----- 14:31

COMMAND ===>

Show Full Message Text	===> YES	(MESSAGE YES NO)
Notify if No Runstats	===> YES	(NOSTATS YES NO)
Flag all Tspace Scans	===> NO	(NOTSCAN NO YES)
Flag all NM Index Scans	===> NO	(NONONIX NO YES)
Flag all Parts scanned	===> YES	(ALLPART YES NO)
Match Index Scan NLEAF	===> 1000	(ISCANPG 1000 nnnnn)
Nonmatch Ix Scan NLEAF	===> 5000	(NONIXPG 5000 nnnnn)
Tablespace Scan NPAGES	===> 50000	(TSCANPG 50000 nnnnn)
Limit no. tables joined	===> 10	(JOINTAB 10 nnn)
Limit IN(list) elements	===> 10	(INLISTS 10 nnn)
Limit Indexes Updated	===> 10	(IXUPDAT 5 nnn)
Pct matching Ix columns	===> 0.50	(MATCOLS 0.50 n.nn)
Composite Filter Factor	===> 0.15	(PREDPCT 0.15 n.nn)

>>> Press PF3 to Return to Main Menu, or PF12 to Exit SQL PA <<<

The change parameters function is now processed on three separate panels: one for **system** oriented parms, one for **user** parms, and one for the new **thresholds**, which the user controls.

SQL PA ----- Change System Parameters ----- 10:19
COMMAND ==>

*Target DB2 host member ==> ANLSAMP (SANLDATA(member) has ANLCNTL parms)

Connect to DB2 via DRDA ==> +OFF+ (VIADRDA location)

DB2 current version is ==> V8R1 (VERSION V8R1 | V7R1 | V6R1 | V6R2)
Buffer hit ratio (avg.) ==> 0 (BUFFHIT range 0 - 100)
Hiper pool read percent ==> 0 (HPOOLRD range 0 - 100)
DASD storage subsystem ==> ESS800 (STORAGE 3390-3, NEWDSK, ESS, ...)

User-defined DASD name ==> +OFF+ (if NEWDSK, assign NEWSTOR name)
User-defined seek time ==> 0.0120 (if NEWDSK, Avg Seek Time in Sec)
User-defined rotation ==> 0.0071 (if NEWDSK, Rotation Delay in Sec)
User-defined xfer rate ==> 4200.0 (if NEWDSK, Transfer Rate in KB/Sec)

Percent of data sharing ==> 0 (DATSHR 0 - 100)
Data shr group members ==> 0 (DSGROUP 0 - 32)
Process DBRM with NOSEQ ==> +OFF+ (PROCESS +OFF+ | NOSEQ)
Treat Version parm ASIS ==> +OFF+ (PROCESS +OFF+ | ASIS)

>>> Press PF3 to Return to Main Menu, or PF12 to Exit SQL PA <<<

The change **System** parameters panel modifies the configuration parms which supplement the info in the Target Hosts library (SANLDATA). The PROCESS parms have moved to this panel and are now set to **NOSEQ | ASIS** or **+OFF+** instead of **YES | NO** as in V2.2

SQL PA ----- Main Control Panel ----- 14:08

COMMAND ==>

DBRM member pattern.... ==> +OFF+

*Input data set name... ==> INPUT.SQL(WHOME)

Edit input data set... ==> YES

Save edited changes.... ==> YES

Define output files... ==> NO

Go to review reports... ==> NO

Show Current Params.... ==> NO

Reset 'Thresholds' ==> NO

Reset 'User' parms.... ==> NO

Reset 'System' parms... ==> NO

Process Retro Explain ==> YES

View Help Tutorial..... ==> NO

Compare Old/New Plans ==> NO

Go to Easy Explain..... ==> NO

Modify Statistics..... ==> NO

Go to What If Analysis ==> NO

>>> Press PF3 or PF12 to exit SQL PA, or press Enter to continue <<<

Some of the new options in V3.1 include the Retro-Explain feature, illustrated next.

SQL PA ----- Retro Plan Explains ----- 10:29
COMMAND ==>

Existing Plan Table Owner..... ==> TDT690 Owner of the "Plan Table"

Enter the Range of query numbers to Retro Explain, or ALL of them:

Starting Plan Query Number..... ==> ALL Starting Queryno, or ALL

Terminating Plan Query No..... ==> 999999999 Ending Queryno, or 999999999

>>> To process, Press Enter or PF3 key. To cancel, Press PF12 key<<<

To obtain a re-explain of an existing plan, provide the high level qualifier of the PLAN_TABLE and the range of QUERYNO that you want re-explained. This could be ALL plans in the table or just a subset.

SQL PA ----- Compare Old Vs. New ----- 10:29
COMMAND ==>

EEEPATH (EEE) Table Owner..... ==> TDT690 Owner of the "EEEPATH" table

Corresponding SQL File ==> ANLEEE.SQL

Starting Plan Query Number..... ==> ALL Starting Queryno, or ALL

Terminating Plan Query No..... ==> 999999999 Ending Queryno, or 999999999

>>> To process, Press Enter or PF3 key. To cancel, Press PF12 key<<<

To compare the access paths and costs of a plan previously stored by the Easy Explain component, use the Compare Old-New Plans panel. The SQL is kept in corresponding files, along with the key plan and cost data in the user's **EEEPATH** table.


```

EDIT ---- TDT690.INPUT.SQL(LGCLUS) - 01.24 ----- Columns 001 072
Command ==> ANL                                     Scroll ==> PAGE
***** ***** Top of Data *****
=====
==MSG> WELCOME TO SQL PA PROCESSING, EXECUTED FROM WITHIN NORMAL ISPF EDITOR
==MSG> LINE NUMBER COMMANDS P, PP, AND P9 CAN HELP LIMIT SET OF SQL SELECTED
==MSG> TYPE 'ANL' IN COMMAND FIELD ALONG WITH PP LINE COMMANDS TO SELECT SQL
==MSG> ELSE TO SUBMIT THE ENTIRE FILE PRESS PF03 OR PF15 TO BEGIN PROCESSING
==MSG> TO REMOVE THIS REMINDER NOTE, JUST TYPE 'RESET' IN THE COMMAND FIELD.
==MSG> TO MAKE EDIT CHANGES PERMANENT, SET SAVE CHANGES ON MAIN PANEL TO YES
=====
000100 -----
000200 --          C L U S T E R      I N D E X      ( C I K E Y )      T E S T S      --
000300 -----
000400 -- ROWS RETURNED VALIDATED AND VERIFIED                                V8 BENCH      --
PP      -----
000600 -- EXPLAIN ALL SET QUERYNO = 1001 FOR
000700     SELECT * FROM TDT690.L1000
000800     WHERE CIKEY = 1;
000900 --                                     1 ROW RETURNED ("=" PRED)
001000 -----
001100 -- EXPLAIN ALL SET QUERYNO = 1002 FOR
001200     SELECT * FROM TDT690.L1000
PP      WHERE CIKEY = 50086;

```

Traditional SQL PA execution involves editing the input data set, using **PP-PP** in the line number fields, and **ANL** on the command line, to extract the lines of SQL to consider.

SQL PA ----- Main Control Panel ----- 14:08

COMMAND ==>

DBRM member pattern.... ==> +OFF+

*Input data set name... ==> INPUT.SQL(WHOME)

Edit input data set... ==> * Save edited changes.... ==> YES

Define output files... ==> NO Go to review reports... ==> NO

Show Current Params.... ==> NO Reset 'Thresholds' ==> NO

Reset 'User' parms.... ==> NO Reset 'System' parms... ==> NO

Process Retro Explain ==> NO View Help Tutorial..... ==> NO

Compare Old/New Plans ==> NO Go to Easy Explain..... ==> NO

Modify Statistics..... ==> NO Go to What If Analysis ==> NO

>>> Press PF3 or PF12 to exit SQL PA, or press Enter to continue <<<

Editing is now complete...if the user selects SAVE on the main panel, then the edited changes are saved automatically into the permanent data set; else, they are discarded.

```
* * * * *
*
*   THE SQL PERFORMANCE ANALYZER IS NOW CONTEMPLATING THE
*   COST OF YOUR SQL STATEMENTS... THIS SHOULD JUST TAKE
*   A FEW MOMENTS...
*   ...THE SQL ADVISOR IS ALSO CONSIDERING YOUR OPTIONS...
*
* * * * *
```

Execution takes just a few seconds ...

SQL PA ----- Output Reports Menu ----- 10:36

OPTION ==> 1

Choose a Report to Browse:

- 1 SQL PA COST Report - A cost summary for each SQL statement
- 2 SQL PA EXPLAIN Report - Enhanced explain for each SQL statement
- 3 SQL PA TRACE Report - Detailed analysis of each SQL statement
- 4 SQL PA LIMITS Report - Query limit report: one line summary
- 5 SQL PA EXTRACT Report - Extracted SQL and Objects used in this run

Other options:

T TUTORIAL

X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

The main report menu allows selection of several report levels and in V3 the users may also display the SQL EXTRACT report, a file that contains the original SQL and optional Objects Used report (all tables and indexes referenced by the SQL).

BROWSE -- TDT690.ANLCOST.LOG ----- Line 00000022 Col 001 080
Command ==> Scroll ==> PAGE

```
ANLPGM3N ** Statistics ** Total statements examined = 2
          ** for this ** Good statements prepared = 2
          ** Parser run ** Bad statements w/errors = 0
```

ANL4004I The CAF Close return code was 0 for ANLPGM3N on: DSNB

```
ANL2000I Total number of plan records processed was 6
          Total number of parsed records scanned was 1
          Total number of errors encountered in run 0
```

```
Catalog Reads - Stored: Tables = 2 Tabspaces = 1
                        Indexes = 2 Routines = 0
                        Idxkeys = 2 Columns = 6
```

```
Cat Reads - Not Stored: Relations = 0
                        Coldist = 5 Alt Index = 13
```

ANLPGM3N Program ends at: 10:50:15.944 on 06-05-2006 Version 3.1.1 3N-3101

Highest Message Code generated by ANLPGM3N was 12

```
* SQL Performance Analyzer * 5697-I04 * 3N-3101 *
* (C) Copyright IBM Corporation 2000, 2006 *
* (C) Copyright Innovative Management Solutions *
```

SQL PA ----- Output Reports Menu ----- 09:26

OPTION ==> 2

Choose a Report to Browse:

- 1 SQL PA COST Report - A cost summary for each SQL statement
- 2 SQL PA EXPLAIN Report - Enhanced explain for each SQL statement
- 3 SQL PA TRACE Report - Detailed analysis of each SQL statement
- 4 SQL PA LIMITS Report - Query limit report: one line summary
- 5 SQL PA EXTRACT Report - Extracted SQL and Objects used in this run

Other options:

- T TUTORIAL
- X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

The enhanced Explain report is option 2, containing the Explain plans and catalog stats plus Advisor messages and other essential cost information.

```
BROWSE -- TDT690.ANLREP.RPT ----- Line 00000000 Col 001 080
Command ==>                               Scroll ==> PAGE
***** Top of Data *****
10:50:11.165          SQL Performance Analyzer          Version  3.1.1
06-05-2006           Enhanced Explain Report           Level   3N-3101
```

SQL PA Parameters

(ANLPARM)

```
REPORTS ALL
OBJECTS YES
VERSION V8R1
STORAGE ESS800
BUFFHIT 0.00
HPOOLRD 0.00
DEGREES ANY
CONNECT CAF
PRECISE ALL
QUALIFY TDT690
USEPLAN TDT690
VIADRDA +OFF+
ADVISOR ALL
SHOWALT YES
DBRMKEY +OFF+

RUNFROM TSO
```

The Enhanced Explain report shows the parameters set by the user on the “change User parameters” panel (ANLPARM)...

BROWSE -- TDT690.ANLREP.RPT ----- Line 00000022 Col 001 080
Command ==> Scroll ==> PAGE

NOSTATS YES
JOINTAB 10
INLISTS 10
IXUPDAT 10
NOTSCAN NO
NONONIX NO
ALLPART YES
TSCANPG 50000
ISCANPG 1000
NONIXPG 5000
MATCOLS 0.50
PREDPCT 0.15

(ANLCNTL)

SUBSYST	DSNB	NAME OF DB2 SYSTEM: DSN ** DEFAULT CONFIGURATION **
SUBVERS	V8R1	DB2 VERSION: V8COM V8NFM V7R1 V6R1 V6R2
SETPLAN	YES	CAN USERS SPECIFY THEIR OWN PLAN_TABLE HLQ? NO YES
DBTRACE	OFF	FOR DIAGNOSTICS ONLY: OFF ON ALL DMP
MIPRATE	0450.000	IBM 2084-301 IS DEFAULT MACHINE
ENGINES	00001	ONE ENGINE
SRMCONS	21857.92	AS MEASURED FROM TSO TEST
LPARENG	01.000	PORTION OF THE BOX DEDICATED TO DB2

... as well as the Thresholds and those set in the Target Host library configuration member (ANLCNTL).

```
BROWSE -- TDT690.ANLREP.RPT ----- Line 00000044 Col 001 080
Command ==>                               Scroll ==> PAGE
```

```
SQL PA Analysis for Queryno 190000000
```

```
SELECT *
FROM TDT690.L1000
WHERE CIKEY BETWEEN 10
AND 5000
```

```
Queryno: 190000000   Qblockno: 1   Planno: 1   MixOpSeq: 0
Process ->
```

```
+-----+
|ANL7003I *** GUIDELINE:
|Close Yes was specified for the Tablespace and/or the index...
|if these are little used this is OK. If high volume access then
|consider Close No. Extremely relevant pages can be "page fixed"
|in memory by highly referencing, using Hiperpools (through V7),
|putting into a dedicated buffer pool large enough to hold all
|pages, deploying data sharing with Group Buffer Pool Cache All
|option, etc. each with associated costs. Close No also increases
|chances that DBD will remain in EDM Pool for next execution.
|
+-----+
```

The SQL Advisor provides some insights into this specific SQL statement, under this DB2 release level.

+-----+
|ANL7006I *** GUIDELINE:|

|This statement contains a Select of all the columns in the table. |
|That is generally "bad form" for SQL writing, which will result in |
|increased processing time to fetch and process each column of each |
|row. Only use this form when you wish to select all columns in a |
|View definition. Else, name columns that you want individually. |

+-----+
|ANL7037I *** GUIDELINE:|

|Dynamic SQL can cache the prepared statement to reduce reexecution |
|overhead. Use REOPT(ONCE) to load "best fit" access plan. Also, |
|use RUNSTATS with REPORTS NO UPDATE NONE to invalidate (remove) |
|prepared SQL statements in cache. |

+-----+
Access is via the clustering (Insert & Load order) index for this table

+-----+
|ANL5018W *** ALERT:|

|This clustering (insert) index has a low cluster ratio (below 80) |
|and/or is not well clustered (Clustered flag = N), meaning that |
|inserts are following an unclustered pattern: this is not helpful. |
|You should Reorganize your Table and Index, and re-run Runstats to |
|update these statistics. That should result in better access paths |
|and performance in general. |

BROWSE -- TDT690.ANLREP.RPT ----- Line 00000095 Col 001 080
Command ==> Scroll ==> PAGE

CLUSTER MATCH IX SCAN

IX Creator: TDT690
Index Name: L1000CIN

Vers: 0 Key Len: 6 Padded: - C-ed: N C-ing: Y CluRatio: 93.2500
Fullkey card: 100000 Firstkey card: 99800
Type: 2 Nleaf pages: 459 Nlevels: 3 Unique: D DUPLICATE OK
1 of 1 columns are matched Close: Y Lock mode: IS Bpool: BP10

Key	Column Name	Order	Type	Dist	Len	Null	Colcard	Dist#
1	CIKEY	A	DECIMAL	N	9	Y	99340	10

Alternative Index
++++
Creator: TDT690
Ix Name: L1000R1N

Vers: 0 Key Len: 6 Padded: - C-ed: N C-ing: N CluRatio: 11.6627
Fullkey Card: 5954 Firstkey Card: 5954
Type: 2 Nleaf Pages: 155 Nlevels: 2 Unique: D Key Cols: 1

Key	Order	Colcard	Column Name
-----	-------	---------	-------------

1	A	5954	RIKEY1
---	---	------	--------

There is a new look to the 'chosen' index information, with more data content.
The same is true for the 'alternate' indexes, not selected by DB2 for this query.

SQL PA ----- Output Reports Menu ----- 09:26

OPTION ===> 3

Choose a Report to Browse:

- 1 SQL PA COST Report - A cost summary for each SQL statement
- 2 SQL PA EXPLAIN Report - Enhanced explain for each SQL statement
- 3 SQL PA TRACE Report - Detailed analysis of each SQL statement
- 4 SQL PA LIMITS Report - Query limit report: one line summary
- 5 SQL PA EXTRACT Report - Extracted SQL and Objects used in this run

Other options:

T TUTORIAL

X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

The Detail Trace report is option 3, and provides the most information

BROWSE -- TDT690.QTRACE.RPT ----- Line 00000055 Col 001 080
Command ==> Scroll ==> PAGE

ANL3098I *** Configuration Note:

The overall rating for this system is 450.000 Million Instructions/Second
Each individual engine will operate at 450.000 Million Instructions/Second

ANL3036I *** The assumed Overhead for Connection Type of CAF has been
estimated in Millions of Instructions for these Batch SQL Stmts:

Thread Mgmt = 0.096400 Attach = 0.300000 Appl = 0.075000

ANL6042I *** NOTE:

Dynamic statement caching is assumed for this set of estimates.

ANL3037I *** Estimated ESS800 DASD Service Times are:

Sync Read 4K Page = 0.0042 Prefetch block of 32 4K Pages = 0.0115

Sync Read 8K Page = 0.0045 Prefetch block of 16 8K Pages = 0.0115

Sync Read 16K Page = 0.0049 Prefetch block of 8 16K Pages = 0.0115

Sync Read 32K Page = 0.0059 Prefetch block of 4 32K Pages = 0.0115

Async Wrt 4K Pages = 0.0115 Async Write of 16K Page Block = 0.0115

Async Wrt 8K Pages = 0.0115 Async Write of 32K Page Block = 0.0115

=====

The Detail Trace report provides the internal CPU and DASD processing characteristics used by SQL PA, based on the Target Host parameters.

BROWSE -- TDT690.QTRACE.RPT ----- Line 00000195 Col 001 080
Command ==> Scroll ==> PAGE

Rows processed =	4957	Percent table processed =	0.049566172
Cols processed =	50	Boolean Filter from OPT =	0.049566168
Data pages read=	1240	Indexed Leaf pages read =	23

Sync Read I/Os =	5	Table =	2	Index =	3
Prefetch I/Os =	40	Table =	39	Index =	1
Async Wrt I/Os =	0	Table =	0	Index =	0
Get Page Calls =	1292	SysIO =	7	LogIO =	0

Sync Read =	0.037500	Prefetch =	1.144000	Async Wrt =	0.000000
Get Pages =	3.100800	Systems =	0.258450	Log Write =	0.000000
Decompress=	0.000000	Compress =	0.000000	Hiperpool =	0.000000
Fetch Row =	13.631750	Lock/etc =	1.520400	Processes =	16.043610

Predicate = 1 Class 1 = 34.592510 Other O/H = 1.615400

The DB2 Optimizer has provided a processing estimate of 122 Msec
equating to 2706 system Service Units. The "Cost Category" is: A

The cost estimates are broken down into their individual components, estimated in million of instructions, so a SYNC READ cost of 0.037500 = 37,500 instructions.

The 'Class 1' and 'Other O/H' times are rollups of the various cost components, and should equate roughly to the Class 1 and Class 2 CPU time estimates.

BROWSE -- TDT690.QTRACE.RPT ----- Line 00000195 Col 001 080
Command ==> Scroll ==> PAGE

Queryno: 190000000 Qblockno: 1 Planno: 1 MixOpSeq: 0
Summary ->

The Total Cumulative Path Length for this query is 36.207910M Ins.
Resulting in a Total CPU Time of 0.08046 Seconds consumed overall.
DB2 will put Class 1 CPU Time of 0.07687 Seconds in SMF 101 record.
DB2 shows additional CPU Time of 0.00359 Seconds in SMF 100 record.
Estimated Total Logical I/O calls = 45 (excluding system) and
Estimated Total Physical I/O calls = 45 with Hit Ratio = 1.000.
Wait Time for Sync Read I/O = 0.02118 Prefetch I/O = 0.46118
Wait Time on Async Write I/O = 0.00000 Total IWAIT = 0.48235
Wait Time for VSAM Open/Close macros, Binding and Locking = 0.11261

*_**

*_**

* Query 190000000 will require 0.67542 Seconds of Elapsed Time *
* During which 0.08046 Seconds of CPU Time will be consumed and *
* a Total of 45 Physical I/O requests will be issued to disk *
* Qunits 18 Estimated processing cost \$ 0.4631 DOLLARS *

*_**

=====

Elapsed Times are also estimated by SQL PA for I/O Waits and other related operations, like VSAM Open/Close, Prepare, Bind, Locking, etc.

SQL PA ----- Output Reports Menu ----- 09:26

OPTION ===> 4

Choose a Report to Browse:

- 1 SQL PA COST Report - A cost summary for each SQL statement
- 2 SQL PA EXPLAIN Report - Enhanced explain for each SQL statement
- 3 SQL PA TRACE Report - Detailed analysis of each SQL statement
- 4 SQL PA LIMITS Report - Query limit report: one line summary
- 5 SQL PA EXTRACT Report - Extracted SQL and Objects used in this run

Other options:

T TUTORIAL

X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

The Query LIMITS report is a one line summary and eyecatcher per Query evaluated by SQL PA -- so you can quickly spot the long running queries.

BROWSE -- TDT690.QLIMIT.RPT ----- Line 00000000 Col 001 080

Command ==>

Scroll ==> PAGE

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

10:50:11.165
06-05-2006

SQL Performance Analyzer
Query Limits Report

Version 3.1.1
Level 3N-3101

CEIQ\$	Error	Queryno	CPU Time	Elapsed	Phys I/O	Qunits	Monetary	Cost
-----	0	190000000	0.08046	0.675	45	18		0.4631
-----	0	90000000	0.08046	0.675	45	18		0.4631
-----	0	190000001	0.03573	0.179	31	8		0.3155
-----	0	90000001	0.03618	0.155	30	8		0.3055

***** Bottom of Data *****

.....

If the input comes from a DBRM or Path Checker, then the Member name and Reason are also populated in the QLIMIT report: (same Report, shifted RIGHT here)

BROWSE -- TDT690.QLIMIT.RPT ----- Line 00000000 Col 025 105

Command ==>

Scroll ==> PAGE

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

SQL Performance Analyzer
Query Limits Report

Version 3.1.1
Level 3N-3101

CPU Time	Elapsed	Phys I/O	Qunits	Monetary	Cost	Member	Reason
0.08046	0.675	45	18	0.4631		MEMB001	TSCAN
0.08046	0.675	45	18	0.4631		MEMBER2	ACCPATH CHG
0.03573	0.179	31	8	0.3155		MEMBER2	NONMATCH IX
0.03618	0.155	30	8	0.3055		MEMB003	TSCAN

***** Bottom of Data *****

SQL PA ----- Output Reports Menu ----- 09:26

OPTION ==> 5

Choose a Report to Browse:

- 1 SQL PA COST Report - A cost summary for each SQL statement
- 2 SQL PA EXPLAIN Report - Enhanced explain for each SQL statement
- 3 SQL PA TRACE Report - Detailed analysis of each SQL statement
- 4 SQL PA LIMITS Report - Query limit report: one line summary
- 5 SQL PA EXTRACT Report - Extracted SQL and Objects used in this run

Other options:

T TUTORIAL

X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

New report option with PTF3101 is EXTRACT report, containing SQL statements and a list of Objects used by them.

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

```
SET CURRENT SQLID = TDT690 ; /* 10:50 06-05-2006 */
```

```
EXPLAIN ALL SET QUERYNO = 190000000 FOR
```

```
SELECT * FROM L1000
```

```
WHERE CIKEY BETWEEN 10 AND 5000
```

```
;
```

```
EXPLAIN ALL SET QUERYNO = 190000001 FOR
```

```
SELECT * FROM L1000 A, S100 B
```

```
WHERE A.CIKEY BETWEEN 10 AND 5000
```

```
AND B.CIKEY BETWEEN 10 AND 5000
```

```
AND A.CIKEY = B.CIKEY
```

```
;
```

DB2 OBJECTS USED: TABLES

CREATOR: TDT690

TABNAME: L1000

CREATOR: TDT690

TABNAME: S100

DB2 OBJECTS USED: INDEXES

CREATOR: TDT690

IDXNAME: L1000CIN

CREATOR: TDT690

IDXNAME: S100CIN

The SQL EXTRACTS report shows the SQL statements, plus optional Objects used when the OBJECTS YES parm is set.

SQL PA ----- Tutorial Main Menu ----- 11:49

OPTION ===>

Choose a topical area below, or select 1 to see the full index of topics.

- | | | | |
|-----|---------------------|---|---|
| 1 | Index To Topics | - | Review complete SQL PA index of topics |
| 2 | Main Control Panel | - | Processing input from the Main panel |
| 3 | Output Datasets | - | Defining the output report data sets |
| 4 | Reporting Options | - | Learn about SQL PA reporting options |
| 5 | User Parameters | - | Review the SQL PA "user" parameters |
| 6 | System Parameters | - | Review the SQL PA "system" parameters |
| 7 | System Thresholds | - | Review the current "threshold" settings |
| 8 | Product Information | - | See Product Release and Copyright Info |
| 9 | Changing Statistics | - | Change catalog stats, migrate and apply |
| PF3 | EXIT Tutorial | - | Return to previous SQL PA processing |

The SQL PA Tutorial is available from the Main Menu as well as the Reports Menu

OPTION ==>

1 ==> Main Panel Options

- Input data set name
- Process DBRM members
- Edit/Save input data
- Define output files
- Reset User parms
- Reset System parms
- Reset Thresholds
- Retro Explains

2 ==> Reporting Options

- Reporting levels
- Cost summary
- Enhanced explain
- Detailed trace
- Query limits

6 ==> Catalog Statistics

PF3 ==> Exit

3 ==> User Parameters

- REPORTS - ADVISOR - QUALIFY
- PRECISE - DEGREES - USEPLAN
- REFRESH - SHOWALT - KEEPLAN
- CONNECT - DELIMIT - DBTRACE
- NLSCODE - RETCODE

4 ==> System Parameters

- Target DB2 Host Member
- VIADRDA - VERSION - BUFFHIT
- HPOOLRD - STORAGE - NEWSTOR
- DATASHR - DSGROUP - NEWSEEK
- PROCESS - NEWXFER - NEWROTA

5 ==> System Thresholds

- MESSAGE - NOSTATS - NOTSCAN
- INLISTS - JOINTAB - NONONIX
- TSCANPG - ISCANPG - NONIXPG
- ALLPART - MATCOLS - IXUPDAT
- PREDPCT - TURNOFF

The full complement of HELP panels is available as an online Tutorial. Also, in V3 all of the fields have context-sensitive help -- just Press F1 while the cursor is on any input field for a complete explanation of the field and its options.

```
SQL PA ----- Modify Statistics ----- 11:52
OPTION ==> 1
```

Choose an option:

- 1 COLLECT Statistics - Build parameters and execute ANLCAT31 program
- 2 APPLY Statistics - Update catalog with collected statistics
- 3 SET UP JCL defaults - Select Jobcard parameters, libraries, etc.

- X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

In PTF3101, a new set of features allows the definition of wholesale statistics collection and application to the catalog.

SQL PA builds JCL to execute the ANLCAT31 program to collect them, or applies them using ANLUPDAT JCL. Users can tailor the defaults to suit their needs and SUBMIT these jobs.

SQL PA ----- Collect Catalog Stats ----- 11:54

COMMAND ==>

*DB2 Subsystem ID.... ==> DSNB (DSN or subsystem ID)

*DB2 Version "FROM".. ==> V8NFM (V8NFM | V8COM | V7R1 | V6R1)

*"TO" New DB2 Version ==> V8NFM (V8NFM | V8COM | V7R1 | V6R1)

Using TDT690.ANL310.SANLSTAT data set:

*SAVE input parms for later use in member ==> MYSTATS (member name)

*WRITE output Catalog updates into member ==> NEWSTAT (member name)

(optional)

ROUTE this batch job to a different LPAR ==> N/A (/*ROUTE card)

Specify global output target high level names:

*Target database name ==> MYDATBAS (target database)

*Target creator name ==> MYCREATR (target creator)

>>> Press PF3 or Enter to define collection parms, or PF12 to Exit <<<

Select the members of ANL310.SANLSTAT that will hold the parms as well as the JCL for the Apply run. You can also set global database and authid names, choose the Subsystem ID and select the "From", "To" DB2 Versions.

```
SQL PA ----- Collect Catalog Stats ----- 11:55  
COMMAND ===>
```

Specify options for this collection run:

```
Collect COLUMN data    ===> YES          ( Column statistics  Yes | No )
```

```
Collect COLDIST data   ===> YES          ( Distribution stats Yes | No )
```

```
Collect PARTITIONED    ===> YES          ( Partitioned stats  Yes | No )
```

>> Place cursor on scrollable fields below using PF10 (left) and PF11 (right)<<

```
Long Names COLLATH    ===> .            ( Long name authid ) >
```

```
COLLTAB               ===> .            ( Long table name ) >
```

```
Auth.Tname COLLECT    ===> TDT690      ===> L1000          ( 1 table )
```

```
Wildcards FINDALL DB  ===> ANL*        ( Database part )
```

```
FINDALL TS           ===> ANLT*        ( Tablespace part )
```

```
FINDALL TB           ===> *           ( Table name part )
```

>>> Press PF3 or Enter to accept collection parms, or PF12 to Exit <<<

The second select panel allows the user to choose the COLLECT parms, by long name, by short name or wild carding (FINDALL). Other options are also set here, like Column, Distribution and Partition level stats collection.

```

EDIT ---- TDT690.ANL310.SANLSTAT(MYSTATS) - 01.00 ----- Columns 001 072
Command ==>                                     Scroll ==> PAGE
***** ***** Top of Data *****
=====
==MSG> THESE PARAMETERS WERE GENERATED BY THE SQLPA COLLECT STATISTICS PANEL
==MSG> AND MAY BE FURTHER EDITED OR EMBELLISHED BEFORE YOU EDIT THE JCL JOB
==MSG> THAT WILL SUBMIT THEM FOR PROCESSING. MAKE SURE YOU SAVE ANY CHANGES.
==MSG> TO REMOVE THIS REMINDER NOTE, JUST TYPE 'RESET' IN THE COMMAND FIELD.
=====
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 COMMENT *CREATED BY ANLCOLL CLIST *TDT690*
000002 SUBSYST DSNB
000003 SUBVERS V8NFM
000004 VERSION V8NFM
000005 CREATOR MYCREATR
000006 DATABAS MYDATBAS
000007 COLUMNS YES
000008 COLDIST YES
000009 GETPART YES
000010 COLLECT TDT690 .L1000
000011 FINDALL DB ANL*
000012 FINDALL SP ANLT*

000013 FINDALL TB *

```

The user can Edit the resulting parameter list and add more Collection names before submission of the ANLCAT31 job -- this is the **CONTROL** parm list for that JCL.

```

EDIT ---- TDT690.ANL310.SANLSTAT(JOBCARD) - 01.00 ----- Columns 001 072
Command ==>                                     Scroll ==> PAGE
***** ***** Top of Data *****
=====
==MSG> THIS JCL WAS CREATED TO COLLECT CATALOG STATISTICS THAT YOU REQUESTED
==MSG> EDIT THIS DATASET FOR LAST MINUTE CHANGES OR UPDATES, AND THEN SUBMIT
==MSG> FOR PROCESSING. CHECK YOUR OUTPUT STATUS TO ENSURE THAT JOB COMPLETED
==MSG> SUCCESFULLY. DDL TO APPLY THESE STATISTICS WILL BE SAVED IN SANLSTAT.
==MSG> TO REMOVE THIS REMINDER NOTE, JUST TYPE 'RESET' IN THE COMMAND FIELD.
=====
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
==MSG> -CAUTION- Profile is set to STATS ON. Statistics did not exist for
==MSG>           this member, but will be generated if data is saved.
000001 //TDT690ST JOB (T0313DL),'SQL PA',CLASS=E,
000002 //           MSGCLASS=X,NOTIFY=TDT690,REGION=0M
000003 //*ROUTE XEQ LOCAL
000004 //*****
000005 //* READ PARS IN CONTROL AND RETRIEVE CATALOG STATS FOR TABLES *
000006 //* REQUESTED, WRITING MIGRATION DDL TO OUTSTAT FILE FOR REVIEW. *
000007 //*****
000008 //*

```

The JCL deck is then built in member **JOBCARD** of the user's **ANL310.SANLSTAT** library, and the JCL can also be tailored or modified to suit the execution of **ANLCAT31**.

```
EDIT ---- TDT690.ANL310.SANLSTAT(JOBCARD) - 01.00 ----- Columns 001 072
Command ==> Scroll ==> PAGE
```

```
000009 /* JCL IS DYNAMICALLY RECREATED EACH TIME BY 'ANLCOLL' CLIST.
000010 /* SELECT OPTION FOR JCL SETUP PANEL TO CHOOSE DEFAULT LIBRARIES.
000011 /* EXEC FROM DB2 RUN LIBRARY USING THE DSN COMMAND PROCESSOR
000012 /* AND DSNTEP2 PROGRAM.
000013 /*
000014 /*
000015 /* THE INPUT AND OUTPUT DATA SETS ARE MEMBERS OF THE USER'S
000016 /* "TSOID.ANL310.SANLSTAT" PDS, WHICH IS MAINTAINED BY SQL PA.
000017 /* MEMBERS MAY BE REUSED, RENAMED OR COPIED AS DESIRED.
000018 /*
000019 //STEP1      EXEC PGM=ANLCAT31
000020 //STEPLIB   DD DSN=TDT690.ANL310.SANLLOAD,DISP=SHR
000021 //          DD DSN=SYS2.DSNB.DSNLINK,DISP=SHR
000022 //          DD DSN=SYS1.CEE.SCEERUN,DISP=SHR
000023 //SYSPRINT DD SYSOUT=*
000024 //CONTROL  DD DSN=TDT690.ANL310.SANLSTAT(MYSTATS),DISP=SHR
000025 //OUTSTAT  DD DSN=TDT690.ANL310.SANLSTAT(NEWSTAT),DISP=SHR
```

```
SUBMIT
```

After verifying that the JCL is correct, the job is submitted for batch processing by the user.

```
SQL PA ----- Modify Statistics ----- 12:02
OPTION ==> 2
```

Choose an option:

- 1 COLLECT Statistics - Build parameters and execute ANLCAT31 program
- 2 APPLY Statistics - Update catalog with collected statistics
- 3 SET UP JCL defaults - Select Jobcard parameters, libraries, etc.

- X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

Another option on the Modify Statistics panel is to **APPLY** statistics to the catalog, usually the previously collected stats, or possibly restoring the “**original**” catalog statistics.

```
SQL PA ----- Apply Catalog Stats ----- 12:03  
COMMAND ===>
```

```
Using TDT690.ANL310.SANLSTAT data set:
```

```
*READ the Catalog updates from member ===> MYSTATS ( member name )
```

```
(optional)
```

```
ROUTE this batch job to a different LPAR ===> N/A ( /*ROUTE card )
```

```
>>> Press PF3 or Enter to apply these statistics, or PF12 to Exit <<<
```

To “apply” statistics, simply select the member of SANLSTAT that contains the stats which you want to use to update the catalog. These are normally collected by the COLLECT function and saved in the PDS during a prior run.

NOTE: it is an excellent idea to always save the “original” catalog stats before applying any others to the catalog.

The JCL that follows is also built into JOBCARD member and must be submitted by the user for processing.

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

=====
==MSG> THIS JCL WAS CREATED TO APPLY THE CATALOG STATISTICS YOU'VE REQUESTED
==MSG> EDIT THIS DATASET FOR LAST MINUTE CHANGES OR UPDATES, AND THEN SUBMIT
==MSG> FOR PROCESSING. CHECK YOUR OUTPUT STATUS TO ENSURE THAT JOB COMPLETED
==MSG> SUCCESSFULLY. AFTER THAT, YOU CAN RESUME PROCESSING USING THESE STATS.
==MSG> TO REMOVE THIS REMINDER NOTE, JUST TYPE 'RESET' IN THE COMMAND FIELD.
=====

```
000001 //TDT690ST JOB (T0313DL),'SQL PA',CLASS=E,
000002 //          MSGCLASS=X,NOTIFY=TDT690,REGION=0M
000003 //*ROUTE XEQ LOCAL
000004 //*****
000005 //*
000006 //* PROCESSES DDL CREATED BY ANLCAT31 PROGRAM, UPDATING CATALOG
000007 //* WITH THE OPTIMIZER-SENSITIVE VARIABLES FOR TABLES, INDEXES,
000008 //* SPACES, ETC. JCL SETUP PANEL CHOOSES THE DEFAULT LIBRARIES.
000009 //*
000010 //* JCL IS DYNAMICALLY RECREATED EACH TIME BY 'ANLAPPLY' CLIST.
000011 //* EXEC FROM DB2 RUN LIBRARY USING THE DSN COMMAND PROCESSOR
000012 //* AND DSNTEP2 PROGRAM.
000013 //*
000014 //STEP1 EXEC PGM=IKJEFT01,REGION=4096K,TIME=600
000015 //DSNTRACE DD SYSOUT=*
000016 //SYSPRINT DD SYSOUT=*
000017 //SYSTSPRT DD SYSOUT=*
000018 //SYSUDUMP DD DUMMY
000019 //SYSTSIN DD *
000020     DSN S(DSNB)
000021     RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) LIB('SYS3.RUNLIB.LOAD.DSNB')
000022 /*
000023 //SYSIN DD DSN=TDT690.ANL310.SANLSTAT(MYSTATS),DISP=SHR
000024 /*
```


SQL PA ----- JCL default settings ----- 12:07
COMMAND ===>

*DB2 Subsystem ID.... ===> DSNB (DSN or subsystem ID)

Enter or Overlay standard JOB card information below:

*Jobcard 1 ===> //TDT690ST JOB (T0313DL),'SQL PA',CLASS=E,
*Jobcard 2 ===> // MSGCLASS=X,NOTIFY=TDT690,REGION=0M

(optional)

ROUTE this batch job to a different LPAR ===> N/A (/*ROUTE card)

Specify default system libraries for JCL use (do not use quotes):

*DB2 Load Library name ===> SYS2.DSNB.DSNLINK
*DB2 Run Library name ===> SYS3.RUNLIB.LOAD.DSNB
*CEE Run Library name ===> SYS1.CEE.SCEERUN

>>> Press PF3 or Enter to apply these JCL changes, or PF12 to Exit <<<

The default settings for the JCL in the reusable JOBCARD member can be set here, along with the default Subsystem ID, libraries and routing preferences.

SQL PA ----- What If? Analysis ----- 12:07
OPTION ===> 1

Choose an option:

- 1 Modify 'Table' Statistics - modify key Table catalog statistics
- 2 Modify 'Index' Statistics - modify key Index catalog statistics
- 3 Create or Delete an Index - build or remove a temporary index
- 4 Restore Table|Index Stats - restore 'saved' catalog statistics

- X EXIT

>>> To return to main SQL PA processing panel, Press PF3 key <<<

Another new option with V3 is the ability to play “What If?” games with the DB2 optimizer by modifying the key catalog statistics for Tables and Indexes, or by building temporary indexes and then re-evaluating the SQL to see what differences in the access path can be influenced.

```
SQL PA ----- What If Analysis: TABLE ----- 12:08  
COMMAND ===>
```

```
*DB2 Subsystem ID... ===> DSNB          ( DSN or subsystem ID)
```

```
*DB2 Current Version ===> V8NFM        ( V8NFM | V8COM | V7R1 | V6R1)
```

Select a TABLE to modify statistics:

>>place cursor on scrollable fields below, using PF10 (left) and PF11 (right)<<

```
*Table Creator  ===> TDT690          ( table creator ) >
```

```
*Table Name.... ===> L1000          ( table name ) >
```

```
SAVE OLD stats for later restore in member  ===> OLDS100      ( member or NOSAVE )  
of TDT690.ANL310.SANLSTAT data set
```

>>> Press PF3 or Enter to return to Main Panel, or PF12 to Exit <<<

To modify Table stats, first identify the table, and SQL PA will display the statistics on the next panel so that you can change them.

NOTE: be sure to specify the SAVE OLD stats member if you want to restore these later.

```
SQL PA ----- Modify TABLE Statistics ----- 12:10  
COMMAND ===>
```

```
>>place cursor on scrollable fields below, using PF10 (left) and PF11 (right)<<  
Table Creator    ===: TDT690          ( table creator )
```

```
Table Name..... ===: L1000          ( table name )
```

```
Type over any values that you wish to change and then Enter or PF3 :
```

```
CARD..... ===> 100000          ( number of rows in table )
```

```
NPAGES..... ===> 25000          ( number of pages in table )
```

```
NACTIVE..... ===> 25600          ( active pages in tabspace )
```

```
PCTROWCOMP..... ===> 0          ( percent rows compressed )
```

```
PCTPAGES..... ===> 99          ( percent pages with rows )
```

```
>>> Press PF3 or Enter to return to What Ifs, or PF12 to Exit <<<
```

The second panel display the current value of the key optimizer statistics -- the user can type over any of these fields and SQL PA will permanently update those statistics in the catalog. To reset them, choose the Restore option and the member to which you saved the stats on the previous panel.

```
* * * * *
*
*   STATISTICS FOR THIS DB2 TABLE HAVE BEEN UPDATED IN THE
*   CATALOG UNDER SUBSYSTEM DSNB FOR DB2 VERSION V8NFM
*   THE OLD STATISTICS HAVE BEEN SAVED IN MEMBER: OLDS100
*
* * * * *
```

```
***
```

SQL PA provides a quick message to let you know that the stats are updated.

```
SQL PA ----- What If Analysis: INDEX ----- 12:15  
COMMAND ==>
```

```
*DB2 Subsystem ID... ==> DSNB          ( DSN or subsystem ID)
```

```
*DB2 Current Version ==> V8NFM        ( V8NFM | V8COM | V7R1 | V6R1)
```

```
Select an INDEX to modify statistics:
```

```
>>place cursor on scrollable fields below, using PF10 (left) and PF11 (right)<<
```

```
*Index Creator ==> TDT690          ( index creator ) >
```

```
*Index Name.... ==> L1000CIN       ( index name ) >
```

```
SAVE OLD stats for later restore in member ==> OLDCIN ( member or NOSAVE )  
of TDT690.ANL310.SANLSTAT data set
```

```
>>> Press PF3 or Enter to return to Main Panel, or PF12 to Exit <<<
```

To modify Index stats, first identify the index name, and SQL PA will display the statistics on the next panel so that you can change them.

NOTE: be sure to specify the SAVE OLD stats member if you want to restore these later.

```
SQL PA ----- Modify INDEX Statistics ----- 12:16  
COMMAND ===>
```

```
>>place cursor on scrollable fields below, using PF10 (left) and PF11 (right)<<
```

```
Table Creator    ===: TDT690                ( table creator )
```

```
Table Name....  ===: L1000                  ( table name )
```

```
Index Creator    ===: TDT690                ( index creator )
```

```
Index Name....  ===: L1000CIN              ( index name )
```

```
Type over any values that you wish to change and then Enter or PF3 :
```

```
NLEAF.....    ===> 459                    ( number of leaf pages in index )
```

```
NLEVELS.....  ===> 3                     ( number of levels in index )
```

```
FIRSTKEYCARD.. ===> 99800                  ( number rows indexed by first col )
```

```
FULLKEYCARD... ===> 100000                 ( number rows indexed by all cols )
```

```
CLUSTERRATIO.. ===> 0.9324999             ( percent rows clustered together )
```

```
>>> Press PF3 or Enter to return to What Ifs, or PF12 to Exit << <
```

The second panel display the current value of the key optimizer statistics -- the user can type over any of these fields and SQL PA will permanently update those statistics in the catalog. To reset them, choose the Restore option and the member to which you saved the stats on the previous panel.

```
* * * * *
*
*   STATISTICS FOR THIS DB2 INDEX HAVE BEEN UPDATED IN THE
*   CATALOG UNDER SUBSYSTEM DSNB FOR DB2 VERSION V8NFM
*   THE OLD STATISTICS HAVE BEEN SAVED IN MEMBER: OLDCIN
*
* * * * *
```

```
***
```

SQL PA provides a quick message to let you know that the stats are updated.

SQL PA ----- Create Temporary Index ----- 12:17
COMMAND ===>

*DB2 Subsystem ID... ===> DSNB (DSN or subsystem ID)
*DB2 Current Version ===> V8NFM (V8NFM | V8COM | V7R1 | V6R1)

>>place cursor on scrollable fields below, using PF10 (left) and PF11 (right)<<

ON

*Table Creator ===> TDT690 (table creator) >
*Table Name.... ===> L1000 (table name) >

CREATE

*Index Creator ===> TDT690 (index creator) >
*Index Name.... ===> NEWCIN (index name) >

OR DELETE this index.... ===> NO (Delete THIS Index NO | YES)

Type over any values that you wish to change and then Enter or PF3:

UNIQUE ===> NO (Yes | No) WHERE NOT NULL ===> NO (Yes | No)
CLOSE ===> NO (Yes | No) CLUSTER INDEX ===> NO (Yes | No)

>>> Press PF3 or Enter to define the index keys, or PF12 to Exit <<<

Select the NEW index name and the table that you wish to build it upon, and also choose some of the parms, like unique (where not null) close yes | no, cluster index (keywords for Create Index).... The key columns are defined on the next page.

You can also DELETE an index from this panel (any index, not just the temp ones you built)

```
SQL PA ----- Create Temporary Index ----- 12:18
COMMAND ===>
```

```
>>place cursor on scrollable fields below, using PF10 (left) and PF11 (right)<<
 *Index Creator   ===: TDT690           ( index creator )
 *Index Name....  ===: NEWCIN           ( index name )
```

```
Enter up to 5 Column Names in Ascending|Descending Order, then Enter or PF3 :
```

```
A or D  ===> A           Column 1  ===> CIKEY
A or D  ===> A           Column 2  ===> RIKEY2
A or D  ===> A           Column 3  ===> RIKEY1
A or D  ===> A           Column 4  ===> .
A or D  ===> A           Column 5  ===> .
```

```
USING STOGROUP   ===> SYSDEFLT   PRQTY   ===> 360       SEQTY   ===> 180
BUFFERPOOL NAME  ===> BP1        DEFER   ===: YES       DEFINE  ===: NO
```

```
PADDED INDEX    ===> NO          ( Yes | No )      V8 option only
PARTITIONED     ===> NO          ( Yes | No )      V8 option only
```

```
>>> Press PF3 or Enter to define the index, or PF12 to Exit < < <
```

The second panel allows users to define up to a 5 column index, with most of the usual parms, like padded or partitioned, primary and secondary space quantity, buffer pool, etc.

The big difference is the index is ALWAYS built with DEFER YES, DEFINE NO, so there is no space and no build process -- just the structure is recorded in the catalog, after which you can then modify those catalog stats to suit your "What If?" scenarios.

```
* * * * *
*
*   A NEW INDEX HAS BEEN CREATED ON DB2 SUBSYSTEM DSNB
*   FOR DB2 VERSION V8NFM AS YOU HAVE REQUESTED.
*
* * * * *
```

```
***
```

SQL PA provides a quick message to let you know that the index is built

SQL PA ----- Restore Table or Index ----- 12:20

COMMAND ===>

*DB2 Subsystem ID... ===> DSNB (DSN or subsystem ID)

*DB2 Current Version ===> V8NFM (V8NFM | V8COM | V7R1 | V6R1)

From Statistics dataset select "Restore" Member: TDT690.ANL310.SANLSTAT

*Enter Member Name to Restore..... ===> OLDSTATS

>>> Press PF3 to process "Restore" statistics, or PF12 to Exit << <

You can restore any previously saved statistics for either Tables or Indexes by simply recalling the member name in SANLSTAT and issuing the Restore request.

SQL PA ----- Easy Explain Function ----- 12:21

COMMAND ==>

Enter the input data set name: (Can be sequential or partitioned)

INPUT DATA SET NAME ==> INPUT.SQL(DEMOSOME)

VOLUME SERIAL ==> (Enter if not cataloged)

DATA SET PASSWORD ==> (Enter if password protected)

Enter the output data set name: (Can be sequential or partitioned)

OUTPUT DATA SET NAME ==> OUTPUT.EEE

OUTPUT SQL PA STMTS ==> ANLEEE.SQL

Specify processing options:

EDIT INPUT DATA SET ==> YES (Y/N - Enter EEE statements?)

EXECUTE EASY EXPLAIN ==> YES (Y/N - Execute EEE statements?)

BROWSE OUTPUT REPORT ==> YES (Y/N - Browse output data set?)

PRINT OUTPUT REPORT ==> NO (Y/N - Print output data set?)

OWNER OF PLAN TABLE ==> TDT690 (Only meaningful for QNO option)

DB2 SUBSYSTEM NAME ==> DSNB (DB2 Subsystem to execute upon)

Press ENTER to process END to exit or HELP for more information

The Easy Explain function can also be invoked from either the primary ANL@DSN panel or from the Main SQL PA panel.

PLI sample program output for SQL PA Stored Procedure call:

```
/* ----- */
/* -- THE SQL PA STORED PROCEDURE EXPECTS THE FOLLOWING FORMAT -- */
/* ----- */
/*
/* EXEC SQL CALL ANLPRC3C ( :SQL_STMT, :SQL_LEN, :ANL_PARM,
/*                          :ANL_CPU, :ANL_ELAP, :ANL_IOC,
/*                          :ANL_QUNT, :ANL_COST, :ANL_WARN,
/*                          :ANL_CODE, :SQL_CODE )
/* WHERE ...
/*
/* SQL_STMT CHAR(32000) VAR - CONTAINS THE SQL TO EVALUATE
/* SQL_LEN BIN FIXED (15) - CONTAINS THE LENGTH OF SQL_STMT
/* ANL_PARM CHAR(240) - CONTAINS SQL PA USER PARMS: 15 MAX
/* ANL_CPU BIN FLOAT (53) - SQL PA ESTIMATE OF CPU TIME
/* ANL_ELAP BIN FLOAT (53) - SQL PA ESTIMATE OF ELAPSED TIME
/* ANL_IOC BIN FLOAT (53) - SQL PA ESTIMATE OF I/O COUNT
/* ANL_QUNT BIN FLOAT (53) - SQL PA ESTIMATE OF QUNITS USED
/* ANL_COST BIN FLOAT (53) - SQL PA ESTIMATE OF MONETARY COST

/* ANL_WARN CHAR(5) VAR - WARNING FLAGS FOR LIMITS ABOVE
/* ANL_CODE BIN FIXED (31) - SQL PA'S RETURN CODE: 0 GOOD; +N BAD
/* SQL_CODE BIN FIXED (31) - SQLCODE FOR PREPARE/EXPLAIN STORPROC
/* OR SQL PA'S LAST PROCESS RETURN CODE
/* -----
/* WARNING FLAGS IN ANL_WARN ARE DEFINED AS 'Y' OR 'N', GIVEN IN
/* ORDER: CPUTIME, ELAPSED, IOCOUNT, QUNITS, AND MONEY.
/* ----- */
```

PLI sample program output for SQL PA Stored Procedure call:

```
/*                                                    */  
/* FIRST 3 PARMS ARE INPUT. ALL OTHERS ARE RETURNED BY ANLPROCC */  
/*                                                    */
```

```
EXEC SQL CALL ANLPRC3C (:SQL_STMT, :SQL_LEN, :ANL_PARM, :ANL_CPUTM,  
:ANL_ELAPS, :ANL_IOCNT, :ANL_QUNIT, :ANL_MONEY, :ANL_WARN,  
:ANL_CODE, :SQL_CODE);
```

```
.....  
* CAF OPEN RETCODE IS          0
```

```
* EXPLAIN PLAN FOR (LENGTH 240)  
* UPDATE SYSIBM.SYSTABLES  
* SET NPAGES = -1  
* WHERE NPAGES = -1
```

```
* ANLPRC3C SQLCODE IS          0  
* ANLPRC3C RETURNS ==> WARNING FLAGS: -----  
  ELAPSED:      38.70953  CPU TIME:      0.27816  
  I/O COUNT:    101      QUNITS:        16  
  MONETARY:     1.18  
  ANL CODE:     0        SQL CODE:      0
```

```
* CAF CLOSE RETCODE IS        0  
* PROGRAM TERMINATION
```

COBOL sample program output for SQL PA stored procedure call:

EXEC SQL

```
CALL ANLPRC3R ( :ANL-STMT-VAR, :ANL-LEN, :ANL-PARM,  
               :ANL-CPU, :ANL-ELAP, :ANL-IOC, :ANL-QUN,  
               :ANL-MONEY, :ANL-WARN, :ANL-CODE, :SQL-CODE )  
END-EXEC.
```

.....

```
CONNECT TO DB2 DSN RETCODE = 00000  
PARAMETERS ARE DBTRACE ALL REPORTS STP QUALIFY TDT690 CONNECT CAF  
PRECISE NO DEGREES ANY STORAGE 3390-2 VERSION V7R1 BUFFHIT 000  
RETCODE YES
```

```
STATEMENT LENGTH IS 0164  
STATEMENT TEXT IS UPDATE SYSIBM.SYSTABLES SET NPAGES = -1  
WHERE NPAGES = -1
```

```
SQLCODE RETURNED FROM DB2 000000000  
ANL RETURN CODE FROM PROC 00000  
SQL RETURN CODE FROM PROC 00000  
CPU TIME FOR STATEMENT IS 00000.278158  
ELAPSED TIME EXPECTED IS 00033.915607  
IO COUNT FOR STATEMENT IS 000000101  
QUNITS (SERVICE UNITS) IS 000000016  
MONETARY COST VALUED AT $ 00001.166023  
WARNING FLAGS (CEIQ$) ARE -----
```

```
-----  
CLOSED CAF DB2 DSN RETCODE = 00000
```



```
* EXPLAIN PLAN FOR (LENGTH 160)
* SELECT * FROM TDT690.L1000
* WHERE CIKEY < 10 OR CIKEY > 99999
```

The NEW Explain-capable
stored procedures in PTF3101
produce plans and stats as well.

```
* ANLPRE3C SQLCODE IS 0
* ANLPRE3C RETURNS ==> WARNING FLAGS: ---Y-
  ELAPSED: 50.20496 CPU TIME: 0.08100
  I/O COUNT: 800 QUNITS: 1795
  MONETARY: 0.02
  ANL CODE: 0 SQL CODE: 0
  RECORDS: 1
```

The DB2 Access Plan:

```
Qry: 100000001 Blk: 1 Pln: 1 Mth: 0 Typ: Mix: 0 Rid: Bif:
Acc: I Tno: 1 Table: TDT690 . L1000 Corr:
Index: TDT690 . L1000CIN Ixo: N Mtch: 0 SortC|N UJOG: NNNN NNNN
Jon: Mrg: -1 Adg: -1 Aid: -1 Jdg: -1 Jid: -1 Cid: -1 Nid: -1 Par:
Rng: Typ: T Enc: CCSID S|M|D: 0 0 0 Coll: SQLPA
Grp: Pref: S Lock: IS App: Pgm: ANLPRE3C Ver:
```

The DB2 Statistics used:

```
Tty: T Loc: Ncol: 50 Rowz: 1008 CARD: 100000
Npag: 25000 Pctp: 99 COMP: 0 Encd: E Tsty: Part: 0
Nidx: 3 Pgsz: 4 Ntab: 2 Nact: 32040 Sgsz: 32 Tclo: Y
Lkpt: N Maxr: 255 Lksz: A Crat: 99.999500 Cled: Y Cing: Y Ixsz: 4096
Iclo: Y Leaf: 412 lkey: 99340 Fkey: 99340
Nlvl: 3 Kcol: 1 Iunq: D Ityp: 2 Piec: 2097152 Log: Y
Tver: 0 Iver: 0 Nmqt: 0 Ailn: 0 Arln: 1008
Dsiz: 0 Pools: BP11 BP10 Volt: N Ipad: - Dpsi: 0 Irel: N
```

```
Proc MS: 81 Proc SU: 1795 Category: A Reason: NORMAL
```

Licensed Materials - Property of IBM
5645-DB2 5648-A70 (C) Copyright IBM Corp. 1982, 1998
All Rights Reserved.
IBM is a registered trademark of International Business Machines

QMF HOME PANEL
Version 8 Release 1

Query Management Facility

```

          *****          **          **          *****
Authorization ID          **          **          **          **
TDT690                   **          **          **          **
                          **          **          **          **
Connected to             **          **          **          **
DALD                     *****          **          **          **
                          **

```

Enter a command on the command line or press a function key.
For help, press the Help function key or enter the command HELP.

1=Help	2=List	3=End	4=Show	5=Chart	6=Query
7=Retrieve	8=Edit Table	9=Form	10=Proc	11=Profile	12=Report

OK, you may enter a command.
COMMAND ==>

SQL PA works with the QMF Governor to cost out queries before execution begins.

SQL QUERY

LINE 1

```
SELECT * FROM L1000
WHERE CIKEY < 25
```

*** END ***

1=Help	2=Run	3=End	4=Print	5=Chart	6=Draw
7=Backward	8=Forward	9=Form	10=Insert	11>Delete	12=Report

OK, QUERY is displayed.

COMMAND ==>

SCROLL ==> PAGE

After a query is entered, pressing the F2 key will “run” the query -- SQL PA intercepts these *before* execution to evaluate their costs.

SQL PA V3.1 Summary

- User-modifiable Thresholds to control SQL Advisor Warnings and Limits
- Full Easy Explain integration under TSO (use TOPA=ALL from any EEE source)
- TSO interface enhancements, including PDS for report files, field level help, diagnostic tracing, Retro-Explain, compare Old and New plans, etc.
- Stats migration on large scale with FINDALL

SQL PA V3.1 Summary

- Extensive “What If?” catalog statistics modification capabilities under TSO
- Explain-capable stored procedures for both CAF and WLM
- internal improvements, like Select with UR, DB2 version sensitive for better access paths
- use more lower case, some formatted SQL in reports
- message level controls... and much more