

Sage Advice Part 3: Predictive Index Impact Analysis -- Know Before you CREATE

Scott Richard Hayes, *DBI Software, @dbisoftware*

Session Code: D11

Wednesday, 16 November, 11:00-12:00

Platform: DB2 for Linux, UNIX, and Windows



Submitted to IDUG...

Abstract & Key Bullet Points

- Whenever a DBA has an index they want to create to solve a performance issue, there is often someone raising an objection “But will this new index cause the database/application any harm?” For those that desire to successfully create indexes with confidence, this session will present a new method for predictively measuring the impacts of any new indexes so that informed decisions can be fearlessly made. Example commands and SQL will be provided.
- This session continues the Sage Advice series from Parts 1 (Weight Analysis) and 2 (Advanced Index Benefit Analysis)
- Bullet Objectives in slide notes...

Our Agenda today...

- Quick Review of Sage Advice, Part 1, Weight Analysis
- Quick Review of Sage Advice, Part 2, Advanced Index Benefit Analysis
- Sage Advice, Part 3, Predicting Index Impact Analysis



Part 1: Quick Review

SAGE ADVICE PART 1: WORKLOAD WEIGHT ANALYSIS

How much does it weigh?

TOTAL weight and RELATIVE weight...



We have a **WEIGHT** “Opportunity for Improvement”



Table Performance Analysis

Table Rows Read per Transaction (TBRRTX) & WEIGHT

- Not every TX accesses every table, so we expect Rows Read/#TX to be a small average, normally < 10, and often 3 or less
 - TBRRTX tells you where you have Data Page scans occurring
 - > 10, likely opportunity for improvement
 - > 100, definitely opportunity for improvement
 - > 1,000, crisis! **DO NOT UPGRADE HARDWARE**
- In addition to the cost per TX, find the % of DB Rows Read (**Relative Weight**) by expressing Table Rows Read x 100 / Sum of all Rows Read.

Examples

Table Relative Weights and Read I/O Costs

Brother-Panther@ :50016/DBIRL

File Edit View Tools Reports Window Help

Table Performance for D 01/WCSP

Table Workload from 10/31/13 9:25 AM to 10/31/13 12:25 P

Schema	Table	Size (MB)	% Space	Rows Read	% Rows Read	Rows Read/Tx	Rows Read/Sec	Rows Written	% Rows Written	Rows Written/Tx
WCSADM	XPX_PROMOTION	59.414	0.000%	314,269,599	9.250%	3.88	28,315.13	0	0.000%	0.00
WCSADM	PX_PROMOTION	13,581.410	0.190%	252,701,117	7.430%	3.12	22,767.92	0	0.000%	0.00
WCSADM	TABSCHEMA									0.00
WCSADM	TABNAME									0.00
WCSADM	ROWSREAD									0.02
WCSADM	PCT_DB_TB_ROWSREAD									0.00
WCSADM	TBRRTX									0.00
WCSADM	DBIPOC	WEBSITE_DATA_TB		1325709286		99.23	322086.804			0.00
WCSADM	DBIPOCDMS	WEBSITE_DATA_TB		10141825		0.75	2464.000			0.00
WCSADM	IDUG	WEBSITE_DATA_TB		15695		0.00	3.813			0.00
WCSADM	SYSIBM	SYSPLAN		559		0.00	0.135			0.00
WCSADM	SYSIBM	SYSROUTINES		146		0.00	0.035			0.03
WCSADM	SYSIBM	SYSSEVENTMONITORS		133		0.00	0.032			0.00
WCSADM	SYSIBM	SYSHISTOGRAMTEMPLATEBINS		40		0.00	0.009			0.01
WCSADM	SYSIBM	SYSTABLES		35		0.00	0.008			0.00
WCSADM	SYSIBM	SYSHISTOGRAMTEMPLATEUSE		28		0.00	0.006			0.00
WCSADM	SYSIBM	SYSBUFFERPOOLS		18		0.00	0.004			0.00
WCSADM	SYSIBM	SYSDBAUTH		10		0.00	0.002			0.04
WCSADM	SYSIBM	SYSTABLESPACES		8		0.00	0.001			0.01
WCSADM	SYSIBM	SYSROLEAUTH		8		0.00	0.001			0.00
WCSADM	SYSIBM	SYSSEVENTTABLES		8		0.00	0.001			0.00
WCSADM	SYSIBM	SYSROLES		7		0.00	0.001			0.00
WCSADM	SYSIBM	SYSVERSIONS		6		0.00	0.001			0.05
WCSADM	SYSIBM	SYSSEVICECLASSES		6		0.00	0.001			0.00
WCSADM	SYSIBM	SYSWORKLOADS		4		0.00	0.000			0.00
WCSADM	SYSIBM	SYSNODEGROUPS		2		0.00	0.000			0.00
WCSADM	SYSIBM	SYSCONTEXTS		2		0.00	0.000			0.00
WCSADM	20 record(s) selected.									
WCSADM	XCEFFM	46,277.750	0.660%	43,625,040	1.280%	0.54	3,930.54	41,319	0.110%	0.00

SQL WEIGHTS

Aggregated, Concentrated Costs, & their WEIGHTS

- Now that you know the TABLES with the heaviest WEIGHTS, what is the heavy SQL driving I/O to the heavy tables?
 - STMT_TEXT like %TABLE_NAME% has some limitations
 - grep -i "TABLE_NAME" has similar limitations
- What are the HEAVIEST SQL – By table? Across the DB?
 - CPU %
 - Rows Read %
 - Logical Reads %
 - Physical Reads %
 - Rows Written %
 - Execution Time %
 - Sort Time %

SQL HEAVY WEIGHTS

by CPU Time (microseconds)

```
SELECT
  CAST( ( ( (A.TOTAL_USR_CPU_TIME * 1000000) + A.TOTAL_USR_CPU_TIME_MS
            + (A.TOTAL_SYS_CPU_TIME * 1000000) + A.TOTAL_SYS_CPU_TIME_MS )
        / A.NUM_EXECUTIONS )
        AS DECIMAL (15,0)) AS AVG_CPU_TIME_MS,
  CAST (A.NUM_EXECUTIONS AS INTEGER) AS NUM_EXECS,
  CAST(((
    ((A.TOTAL_USR_CPU_TIME * 1000000) + A.TOTAL_USR_CPU_TIME_MS
    + (A.TOTAL_SYS_CPU_TIME * 1000000) + A.TOTAL_SYS_CPU_TIME_MS)
    * 100.0)
    / (select (SUM(B.TOTAL_USR_CPU_TIME) * 1000000)
        + (SUM(B.TOTAL_SYS_CPU_TIME) * 1000000)
        + SUM(B.TOTAL_USR_CPU_TIME_MS)
        + SUM(B.TOTAL_SYS_CPU_TIME_MS)
        + 1.0
      FROM SYSIBMADM.SNAPDYN_SQL B
      WHERE A.DBPARTITIONNUM = B.DBPARTITIONNUM
    )) AS DECIMAL(5,2)) AS PCT_CPU_TIME,
  SUBSTR(A.STMT_TEXT,1,100) AS CPU_SUCKING_SQL
FROM SYSIBMADM.SNAPDYN_SQL A
WHERE A.NUM_EXECUTIONS > 0
ORDER BY A.DBPARTITIONNUM ASC, 3 DESC, 1 DESC FETCH FIRST 25 ROWS ONLY;
```

SQL HEAVY WEIGHTS

by CPU Time (microseconds) - Examples

Brother-Panther® - db2admin@win7srv1:50000/DBI2REP

Statement Performance for WIN7SRV1:50000/PRODDB91											
Statement Workload from 3/12/14 12:00 AM to 3/12/14 3:00 AM											
Follow Up	Stmt ID	Verb	Type	Exec Time (sec)	Avg Exec Time (sec)	% Exec Time	# Execs	CPU Time (sec)	Avg CPU Time (sec)	CPU Cost (\$)	% CPU Time
	644F60DF8...	SELECT	DYNAMIC	2,097.979211	0.925035	19.160%	2,268	886.959273	0.391076	\$1,773.9185	25.292%
	64D382EB4...	SELECT	DYNAMIC	1,761.873208	0.789724	16.090%	2,231	747.728393	0.335154	\$1,495.4568	21.322%
	E58E9C040...	SELECT	DYNAMIC	2,368.219343	0.986758	21.628%	2,400	571.852871	0.238272	\$1,143.7057	16.307%
AUG_CPU_TIME_MS	NUM_EXECS	PCT_CPU_TIME	CPU_SUCKING_SQL								
134767.	36	14.90	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = '204.62.53.126' fetch first 10								
128845.	27	10.68	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'v35.nat.svl.searchme.com' fet								
1575610.	1	4.83	select dayofweek(hittimestamp), count(*) from dbipoc.website_data_tb group by dayofweek(hittimestamp) order by								
133467.	9	3.68	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'crawl1.nat.svl.searchme.com'								
120343.	7	2.58	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = '89.122.29.77' fetch first 10								
137801.	6	2.53	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'ops8-r106.searchme.com' fetch								
670804.	1	2.06	select domainname, count(*) from DBIPOC.WEBSITE_DATA_TB group by domainname order by 2 desc fetch first 10 row								
111800.	6	2.06	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = '65.55.213.36' fetch first 10								
159901.	4	1.96	select protocol, targetfile from DBIPOC.FAILED_HITS_UW where domainname = '61.135.134.215' fetch first 10 rows								
128701.	4	1.58	select protocol, targetfile from DBIPOC.FAILED_HITS_UW where domainname = 'ip-234-26.csmnap.net' fetch first 1								
128701.	4	1.58	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = '204.62.55.12' fetch first 10								
124801.	4	1.53	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'customer-GDL-41-201.megared.n								
150801.	3	1.38	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'ec2-174-129-111-17.compute-1.								
130000.	3	1.19	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'ec2-67-202-48-105.compute-1.a								
179401.	2	1.10	select protocol, authid, dayofweek(hittimestamp), targetfile from DBIPOC.FAILED_HITS_UW where domainname = 'so								
148201.	2	0.91	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = '204.62.53.128' fetch first 10								
148201.	2	0.91	select protocol, targetfile from DBIPOC.FAILED_HITS_UW where domainname = '61.135.134.78' fetch first 10 rows								
265201.	1	0.81	select month(hittimestamp), count(*) from dbipoc.website_data_tb group by month(hittimestamp) order by 1 fetch								
132600.	2	0.81	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = '208.111.154.249' fetch first								
249601.	1	0.76	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'msnbot-65-55-165-16.search.ms								
124801.	2	0.76	select actionverb, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where domainname = 'ec2-67-202-41-52.compute-1.am								
234002.	1	0.71	select ipaddr, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where hittimestamp = '2009-03-11-11.45.43.000000' ord								
218402.	1	0.67	select ipaddr, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where hittimestamp = '2009-08-05-05.04.53.000000' ord								
218402.	1	0.67	select ipaddr, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where hittimestamp = '2009-10-04-21.57.08.000000' ord								
218402.	1	0.67	select ipaddr, targetfile from DBIPOC.SUCCESSFUL_HITS_UW where hittimestamp = '2009-03-13-20.13.36.000000' ord								

25 record(s) selected.

Take a picture of your luggage before you fly-
- easy description when lost
- documents bag condition

- #WISDOM



Part 2: Quick Review

SAGE ADVICE PART 2: ADVANCED INDEX BENEFIT ANALYSIS

So, you found a heavy weight SQL statement, and you passed it to the Design Advisor (db2advis), and the Design Advisor suggests that you create 3, 5, 11, or 13 indexes for a solution!

HOW MANY?

REALLY?

OPTIMIZING INDEX SOLUTIONS

Optimizing Index Solutions

Solving a “Heavy” Query

Execute SQL: db2admin@WIN7SRV1:50000/PRODDB91



Current Schema: DB2ADMIN

Editor

```
SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd, v.verb_desc
FROM DBIPOC.SUCCESSFUL_HITS_VW A,
     DBIPOC.VERB_DESCRIPTIONS V
where a.domainname = 'webnj1.bbhh.com'
and a.targetfile = '/blog/rss/Scott_Hayes_rss2.xml'
and a.bytesxferd < (select avg(b.bytesxferd) from DBIPOC.SUCCESSFUL_HITS_VW B)
and a.hittimestamp < '2011-12-31-21.35.43.304000'
and a.actionverb = v.actionverb
fetch first 100 rows only;
```

[illegible]

Optimizing Index Solutions

Relative Benefit Value Analysis

- **Two Methods to Consider**
 - Index Addition – Add indexes one at a time to assess individual value
 - Index Subtraction – Subtract Indexes one at a time from the solution set to assess the value lost
- **Design Advisor can be overly aggressive on Index Only Access**
 - Sometimes additional columns are added to existing indexes to achieve IX Only access – we anticipate these will have less value
 - Give consideration to predicates involved when making final decisions on which indexes to implement

Optimizing Index Solutions

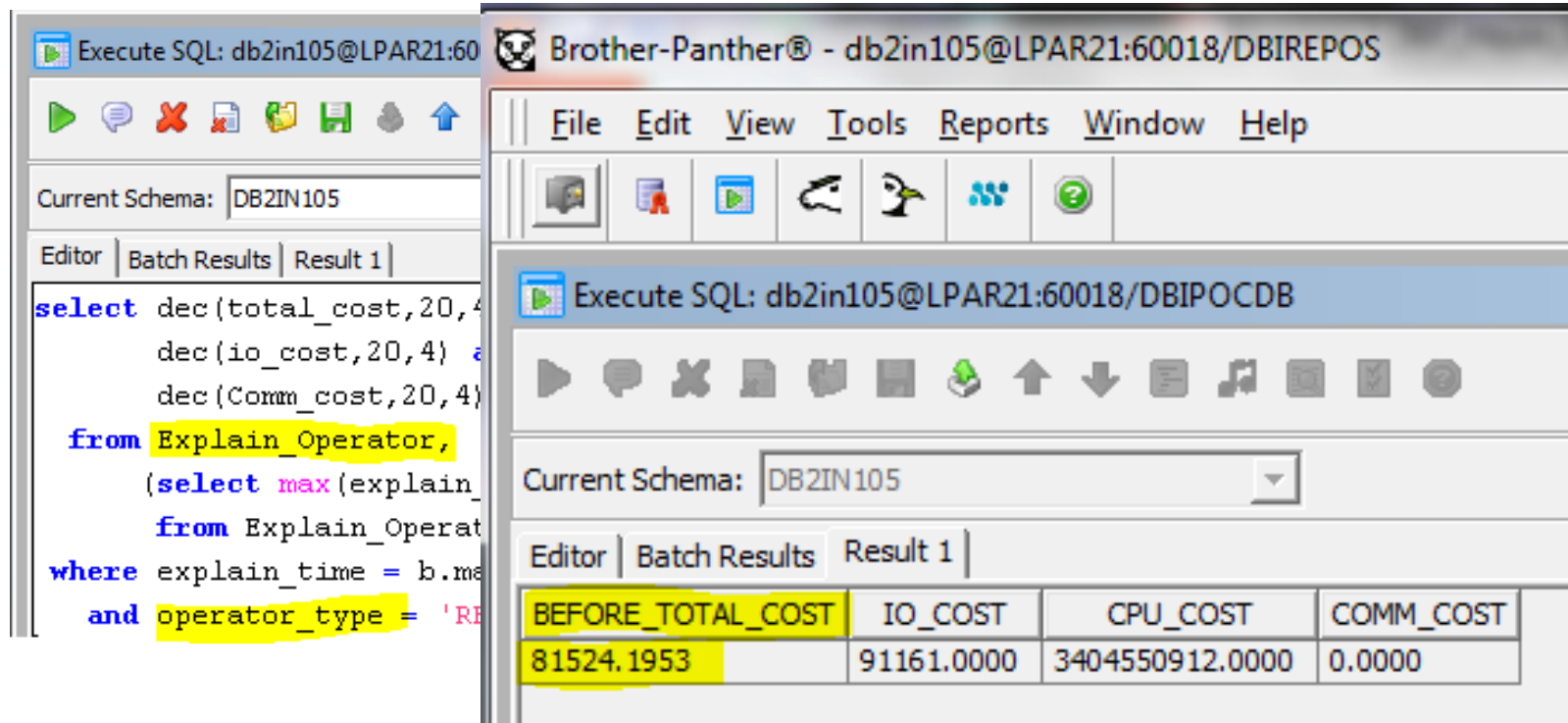
Index Addition 1

- Start with a clean Explain & Advise Environment
 - Delete from Explain_Instance
 - Delete from Advise_Index
- Explain the statement
 - `db2batch -d dbipocdb -f 3Table_Heavy_Query.sql -o e explain`

Optimizing Index Solutions

Index Addition 2

- Find the original/"Before" Explain Cost



The screenshot shows two overlapping DB2 GUI windows. The left window, titled 'Execute SQL: db2in105@LPAR21:60...', displays a SQL query in the Editor tab. The query is: `select dec(total_cost,20,4) as total_cost, dec(io_cost,20,4) as io_cost, dec(comm_cost,20,4) as comm_cost from Explain_Operator, (select max(explain_time) as max_explain_time from Explain_Operator) as max_explain_time where explain_time = max_explain_time and operator_type = 'RE'.` The right window, titled 'Execute SQL: db2in105@LPAR21:60018/DBIPOCDB', shows the results of the query in the Result 1 tab. The results are displayed in a table with four columns: BEFORE_TOTAL_COST, IO_COST, CPU_COST, and COMM_COST. The values are 81524.1953, 91161.0000, 3404550912.0000, and 0.0000 respectively.

Left Window: Execute SQL: db2in105@LPAR21:60...

Current Schema: DB2IN105

Editor | Batch Results | Result 1

```
select dec(total_cost,20,4) as total_cost,
       dec(io_cost,20,4) as io_cost,
       dec(comm_cost,20,4) as comm_cost
from Explain_Operator,
     (select max(explain_time) as max_explain_time
      from Explain_Operator) as max_explain_time
where explain_time = max_explain_time
and operator_type = 'RE'
```

Right Window: Execute SQL: db2in105@LPAR21:60018/DBIPOCDB

Current Schema: DB2IN105

Editor | Batch Results | Result 1

BEFORE_TOTAL_COST	IO_COST	CPU_COST	COMM_COST
81524.1953	91161.0000	3404550912.0000	0.0000

Optimizing Index Solutions

Index Addition 3B – alternate method

- Populate the ADVISE_INDEX table - CLP
 - db2 "select current explain mode from sysibm.sysdummy1"
 - "NO"
 - db2 "set current explain mode **recommend indexes**"
 - db2 -stvf 3Table_Heavy_Query.sql
 - Does not execute the query!
 - Populates the ADVISE_INDEX table
 - db2 "set current explain mode NO"
 - So you can run queries again!

Optimizing Index Solutions

ADVISE_INDEX Table 1

- USE_INDEX Column – the “magic”
 - ‘Y’ Index Recommended or Evaluated
 - ‘N’ Index not to be Recommended or Evaluated
 - ‘R’ An existing clustering RID index was recommended by Design Advisor to be unclustered – this is the case when a new clustering RID index is recommended for the table
 - ‘I’ Ignore an existing non-unique index for Evaluation. The EXISTS column should be ‘Y’ in this case or the index will not be ignored
- Several other interesting and helpful columns too
 - See sample query and results, next slide

Optimizing Index Solutions

ADVISE_INDEX Table 2

Execute SQL: db2in105@LPAR21:60018/DBIPOCDB

Brother-Panther® - db2in105@LPAR21:60018/DBIREPOS

File Edit View Tools Reports Window Help

Execute SQL: db2in105@LPAR21:60018/DBIPOCDB

Connections: LPAR21:60018/DBIPOCDB

Current Schema: DB2IN105 Recent SQL: --delete from advise_index;(1)

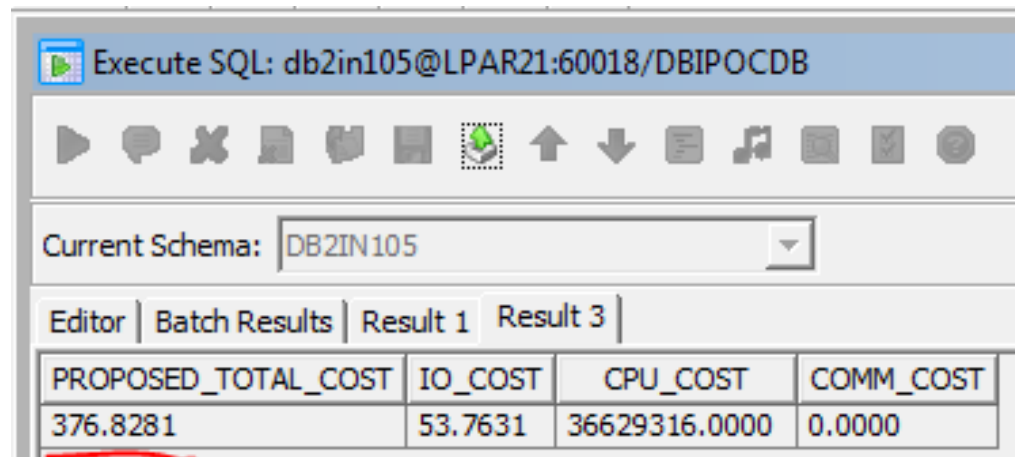
Editor | Batch Results | Result 1 | Result 23 | Result 26 | Result 31

PROPOSED_INDEX	ON_TABLE	EXISTS	USE_INDEX	INDEX_COLS	NLEVELS	NLEAF	UNIQUERULE	FIRSTKEYCARD	FULLKEYCARD
IDX1503092345460	DBIPOC HTML_STATUS_CODES	N	Y	+STATUS_DESC+STATUS_CODE	2	3	D	38	38
IDX1503092345460	DBIPOC HTML_STATUS_CODES	N	Y	+STATUS_DESC+STATUS_CODE	2	3	D	38	38
IDX1503092345530	DBIPOC WEBSITE_DATA_TB	N	Y	+WEBSTATUS-BYTESXFERD	3	1891	D	10	189450
IDX1503092346050	DBIPOC VERB_DESCRIPTIONS	N	Y	+ACTIONVERB-VERB_DESC	2	3	D	12	12
IDX1503092346070	DBIPOC WEBSITE_DATA_TB	N	Y	+DOMAINNAME+TARGETFILE+BYTESXFERD+HITTIMESTAMP+PROTOCOL+ACTIONVERB+WEBSTATUS	3	896	D	134	134

Optimizing Index Solutions

So, what are those proposed indexes worth?

- set current explain mode **EVALUATE INDEXES**
 - USE_INDEX = 'Y' for all Proposed Indexes
- \$ db2 -tvf 3Table_Heavy_Query.sql
- set current explain mode NO
- \$ db2 -tvf Query_In_Slide_Notes.sql
- 376 timerons
- Down from 81,524
 - 99.54% Reduced



Execute SQL: db2in105@LPAR21:60018/DBIPOCDB

Current Schema: DB2IN105

Editor | Batch Results | Result 1 | Result 3

PROPOSED_TOTAL_COST	IO_COST	CPU_COST	COMM_COST
376.8281	53.7631	36629316.0000	0.0000

Optimizing Index Solutions

Index Addition

- What is the value of each index individually, in isolation?
- Set USE_INDEX to 'N' for all Indexes
 - update advise_index set use_index='N';
- For each proposed index:
 - Set USE_INDEX to 'Y'
 - Update ADVISE_INDEX set USE_INDEX = 'Y' where NAME = 'IXNAME(N)'
 - set current explain mode EVALUATE INDEXES
 - db2 -tvf 3Table_Heavy_Query.sql
 - Retrieve the TOTAL_COST from EXPLAIN_OPERATOR table
 - db2 -tvf Query_In_Slide_Notes.sql
 - Compute Savings Percentage
 - Repeat!

Optimizing Index Solutions

Index Addition – 1st Index

```
$ db2 "set current explain mode NO"
DB20000I  The SQL command completed successfully.
$ db2 "update advise_index set use_index='N'"
DB20000I  The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092345460'"
DB20000I  The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I  The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W  The statement was not executed as only Explain information requests
are being processed.  SQLSTATE=01604
```

ORIGINAL_COST	ADD_IX1_TOTAL_COST	TIMERON_SAVINGS	VALUE_PCT
81524.1953	81524.1406	0.0547	0.0000670

Optimizing Index Solutions

Index Addition – 2nd Index

```
$ db2 "set current explain mode NO"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='N' where name = 'IDX1503092345460'"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092345530'"
DB20000I The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W The statement was not executed as only Explain information requests
are being processed.  SQLSTATE=01604
```

ORIGINAL_COST	ADD_IX2_TOTAL_COST	TIMERON_SAVINGS	VALUE_PCT
81524.1953	1844.3532	79679.8421	97.7376615

Optimizing Index Solutions

Index Addition – 3rd Index

```
$ db2 "set current explain mode NO"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='N' where name = 'IDX1503092345530'"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092346050'"
DB20000I The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W The statement was not executed as only Explain information requests
are being processed.  SQLSTATE=01604
```

ORIGINAL_COST	ADD_IX3_TOTAL_COST	TIMERON_SAVINGS	VALUE_PCT
81524.1953	81524.1953	0.0000	0E-7

Optimizing Index Solutions

Index Addition – 4th Index

```
$ db2 "set current explain mode NO"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='N' where name = 'IDX1503092346050'"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092346070'"
DB20000I The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W The statement was not executed as only Explain information requests
are being processed.  SQLSTATE=01604
```

ORIGINAL_COST	ADD_IX4_TOTAL_COST	TIMERON_SAVINGS	VALUE_PCT
81524.1953	6536.0610	74988.1343	91.9826733

Optimizing Index Solutions

Index Addition - Summary

Index Name	Timeron Savings	Value %
IDX1503092345460	0.0547	0.0000670
IDX1503092345530	79679.8421	97.7376615
IDX1503092346050	0.0000	0.0000000
IDX1503092346070	74988.1343	91.9826733
		189,204018 %


**And the award for
LEAST valuable
index goes to...**

**And the award for
MOST valuable
index goes to...**

2nd Place MVI

Optimizing Index Solutions

Compare Explain Plans


Compare Explain Plans - Execute SQL

☒ Synchronize Scrolling

Explain Plan 1:

Baseline

SELECT a.hittimestamp, a.actionverb, a.protocol,

← RETURN [12] (Total Cost=81,524.195)

HSJOIN [11] (Total Cost=81,524.195)

NLJOIN [10] (Total Cost=81,517.406)

GRPBY [9] (Total Cost=40,686.531)

HSJOIN [8] (Total Cost=40,684.035)

TBSCAN [7] (Total Cost=40,634.012)

Table: WEBSITE_DATA_TB (Row Count=633860)

TBSCAN [6] (Total Cost=6.813)

Table: HTML_STATUS_CODES (Row Count=38)

FILTER [5] (Total Cost=40,830.875)

NLJOIN [4] (Total Cost=40,830.875)

TBSCAN [3] (Total Cost=6.813)

Table: HTML_STATUS_CODES (Row Count=38)

TBSCAN [2] (Total Cost=40,824.062)

Table: WEBSITE_DATA_TB (Row Count=633860)

TBSCAN [1] (Total Cost=6.791)

Table: VERB_DESCRIPTIONS (Row Count=12)

Explain Plan 2:

Alternative-2(Virtual Index)

SELECT a.hittimestamp, a.actionverb, a.protocol,

← RETURN [11] (Total Cost=376.828)

NLJOIN [10] (Total Cost=376.828)

NLJOIN [9] (Total Cost=370.042)

NLJOIN [8] (Total Cost=363.251)

GRPBY [7] (Total Cost=356.466)

NLJOIN [6] (Total Cost=353.968)

IXSCAN [5] (Total Cost=6.786)

Index: IDX1503110552010

IXSCAN [4] (Total Cost=353.949)

Index: IDX1503110552080

IXSCAN [3] (Total Cost=6.786)

Index: IDX1503110552010

IXSCAN [2] (Total Cost=13.557)

Index: IDX1503110552370

IXSCAN [1] (Total Cost=6.785)

Index: IDX1503110552200

31

Optimizing Index Solutions

Does a High Value Index have IX Access Only “Baggage”?

PROPOSED_INDEX	ON_TABLE	EXISTS	USE_INDEX	INDEX_COLS	NLEVELS	NLEAF	UNIQUERULE	FIRSTKEYCARD	FULLKEYCARD
IDX1503092345460	DBIPOC HTML_STATUS_CODES	N	Y	+STATUS_DESC+STATUS_CODE	2	3	D	38	38
IDX1503092345530	DBIPOC WEBSITE_DATA_TB	N	Y	+WEBSTATUS-BYTESXFERD	3	1891	D	10	189450
IDX1503092346050	DBIPOC VERB_DESCRIPTIONS	N	Y	+ACTIONVERB-VERB_DESC	2	3	D	12	12
IDX1503092346070	DBIPOC WEBSITE_DATA_TB	N	Y	+DOMAINNAME+TARGETFILE+BYTESXFERD+HITTIMESTAMP+PROTOCOL+ACTIONVERB+WEBSTATUS	3	896	D	134	134

RELOP_TYPE	HOW_APPLIED	PREDICATES
EQ	JOIN	(Q5.ACTIONVERB = Q7.ACTIONVERB)
LT	JOIN	(Q5.BYTESXFERD < (Q4.\$C0 / Q4.\$C1))
EQ	JOIN	(Q2.WEBSTATUS = Q1.STATUS_CODE)
EQ	START	(Q1.STATUS_DESC = 'OK. Request Fulfilled.')
EQ	STOP	(Q1.STATUS_DESC = 'OK. Request Fulfilled.')
EQ	START	(Q2.WEBSTATUS = Q1.STATUS_CODE)
EQ	STOP	(Q2.WEBSTATUS = Q1.STATUS_CODE)
LT	RESID	(Q5.BYTESXFERD < (Q4.\$C0 / Q4.\$C1))
EQ	JOIN	(Q5.WEBSTATUS = Q6.STATUS_CODE)
EQ	START	(Q6.STATUS_DESC = 'OK. Request Fulfilled.')
EQ	STOP	(Q6.STATUS_DESC = 'OK. Request Fulfilled.')
LT	SARG	(Q5.HITTIMESTAMP < '2011-12-31-21.35.43.304000000000')
EQ	SARG	(Q5.TARGETFILE = '/blog/rss/Scott_Hayes_rss2.xml')
EQ	SARG	(Q5.DOMAINNAME = 'webnj1.bbhh.com')
EQ	START	(Q5.WEBSTATUS = Q6.STATUS_CODE)
EQ	STOP	(Q5.WEBSTATUS = Q6.STATUS_CODE)
EQ	START	(Q5.ACTIONVERB = Q7.ACTIONVERB)
EQ	STOP	(Q5.ACTIONVERB = Q7.ACTIONVERB)

Let's play Predicate BINGO!
\$ db2 -tvf Query_In_Notes.sql

VERB_DESC & PROTOCOL
are supporting IX Access Only

Part 3: The New Stuff

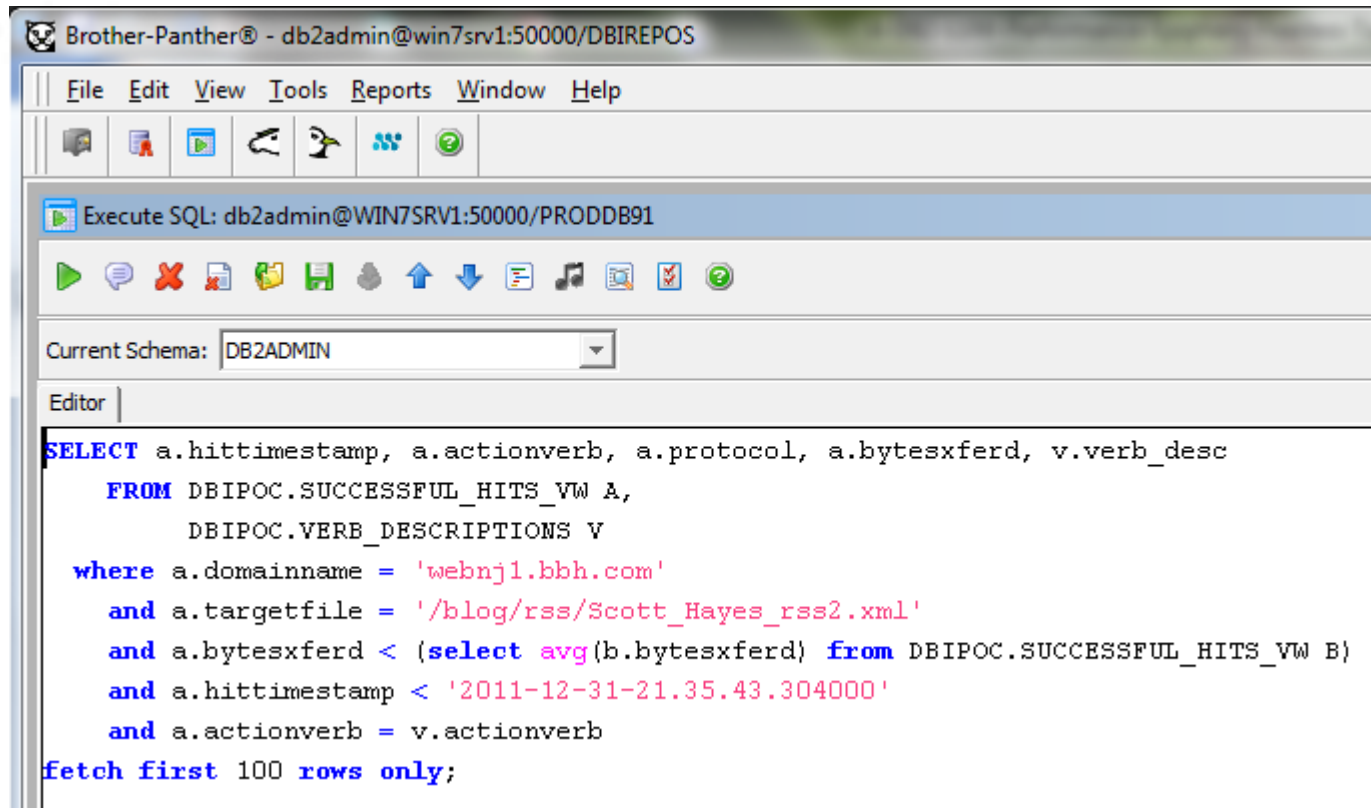
SAGE ADVICE PART 3: PREDICTING INDEX IMPACT ANALYSIS

DBA Performance Analysis Challenges

1. What needs to be fixed or improved?
 - ***Make sure you are fighting the right fires*** - via Weight Analysis
2. What are the optimal design solutions?
 - ***Make sure you are fighting the right fires with the right type of fire extinguishers and equipment*** – via Advanced Index Benefit Analysis
3. Will proposed design solutions cause any inadvertent harm? Will benefits exceed expectations? Can “multiple birds be killed with just one stone?”
 - ***Make sure you are fighting the right fires without causing inadvertent damaging explosions*** – via Predictive Index Impact Analysis

Review

The “Heavy_Query” – 90% of CPU & I/O



Brother-Panther® - db2admin@win7srv1:50000/DBIREPOS

File Edit View Tools Reports Window Help

Execute SQL: db2admin@WIN7SRV1:50000/PRODDB91

Current Schema: DB2ADMIN

Editor

```
SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd, v.verb_desc
FROM DBIPOC.SUCCESSFUL_HITS_VW A,
     DBIPOC.VERB_DESCRIPTIONS V
where a.domainname = 'webnj1.bbhh.com'
and a.targetfile = '/blog/rss/Scott_Hayes_rss2.xml'
and a.bytesxferd < (select avg(b.bytesxferd) from DBIPOC.SUCCESSFUL_HITS_VW B)
and a.hittimestamp < '2011-12-31-21.35.43.304000'
and a.actionverb = v.actionverb
fetch first 100 rows only;
```

Review

Explain Heavy SQL & Get Costs: 187,411 Timerons

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\
delete from explain_instance
DB20000I The SQL command completed successfully.
```

```
delete from advise_index
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\
set current explain mode explain
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\
SQL0217W The statement was not executed as only Expl
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
select dec(total_cost,20,4) as Query_Timeron_Cost
      dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as CPU_cost,
      dec(Comm_cost,20,4) as comm_cost,
      explain_time
from Explain_Operator
where operator_type = 'RETURN'
```

QUERY_TIMERON_COST	IO_COST	CPU_COST
187411.2500	105847.0000	387620.0000

Brother-Panther® - db2admin@win7srv1:50000/DBIREPOS

File Edit View Tools Reports Window Help

Explain: Execute SQL - WIN7SRV1:50000/PRODDB91

Tune SQL Design Analysis Show Graphical View Help

SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd,
v.verb desc

RETURN [14] (Total Cost=187,411.25, 100.0%)

- NLJOIN [13] (Total Cost=187,411.25, 100.0%)**
 - GRPBY [12] (Total Cost=93,657.477, 50.0%)**
 - HSJOIN [11] (Total Cost=93,656.617, 50.0%)**
 - TBSCAN [10] (Total Cost=93,618.906, 50.0%)**
 - Table: WEBSITE_DATA_TB (Row Count=753349)
 - TBSCAN [9] (Total Cost=12.873, 0.0%)**
 - Table: HTML_STATUS_CODES (Row Count=38)
 - FILTER [8] (Total Cost=93,753.773, 50.0%)**
 - HSJOIN [7] (Total Cost=93,753.773, 50.0%)**
 - NLJOIN [6] (Total Cost=93,740.914, 50.0%)**
 - TBSCAN [5] (Total Cost=12.873, 0.0%)**
 - SORT [4] (Total Cost=12.873, 0.0%)**
 - TBSCAN [3] (Total Cost=12.873, 0.0%)**
 - Table: HTML_STATUS_CODES (Row Count=38)
 - TBSCAN [2] (Total Cost=93,728.039, 50.0%)**
 - Table: WEBSITE_DATA_TB (Row Count=753349)
 - TBSCAN [1] (Total Cost=12.862, 0.0%)**
 - Table: VERB_DESCRIPTIONS (Row Count=12)

Review

Get Recommended Indexes - 1

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm_cost
from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'I'
with UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
187411.2500	105847.0000	3876200960.0000	0.0000

1 record(s) selected.

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode recommend indexes"
set current explain mode recommend indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tf heavy_query.sql
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm_cost
from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'I'
with UR
```

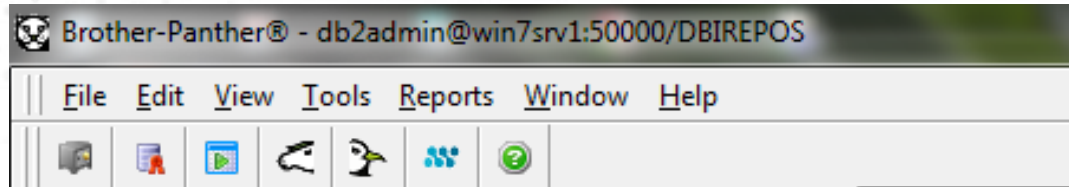
QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
1760.0567	136.4210	40684996.0000	0.0000

1 record(s) selected.

Review

Get Recommended Indexes - 2

Advanced Index Benefit Analysis (AIBA) identifies that some indexes are more beneficial than others



Let's assume we want to create the last 4 indexes after AIBA

We thus know the tables that we'll be creating indexes on (impacted tables)

Three Distinct Table Names are Impacted

PROPOSED INDEX	ON_TABLE	EXISTS	USE_IN	LEVELS	NLEAF
IDX1602060629490	DBIPOC HTML_STATUS_CODES	N	Y		3
IDX1602060629500	DBIPOC WEBSITE_DATA_TB	N	Y		5032
IDX1602060630060	DBIPOC WEBSITE_DATA_TB	N	Y		1063
IDX1602060630030	DBIPOC VERB_DESCRIPTIONS	N	Y	2	3
IDX1602060629480	DBIPOC HTML_STATUS_CODES	N	Y	2	3

Predictive Index Impact Analysis (PIIA) – Step 1

Determine SQL that Impacts the Impacted Tables

- For each impacted table, determine the SQL queries that have contributed I/O
 - In Sage Advice Part 1, we looked at SQL queries that would find “heavy” queries contributing I/O to a table or the database overall
 - Recall that:
 - STMT_TEXT like %TABLE_NAME% has some limitations
 - grep -i “TABLE_NAME” has similar limitations
 - Query the package cache with MON_GET or SYSIBMADM views
 - **Be mindful to include relevant and significant workload timeframes when finding SQL**
- Consider capturing and concatenating workloads from different time periods
 - Sample query in notes

Predictive Index Impact Analysis (PIIA) – Step 2

Determine the **Distinct** Impacting SQL

- For efficiency, determine the DISTINCT SQL statements (workload) of SQL across the UNION ALL of impacted tables.
 - For Example:
 - `SELECT A.C1, B.C1 FROM TB1 A, TB2 B WHERE A.ID1 = B.ID2`
 - This SQL would contribute I/O to BOTH tables TB1 and TB2, but for PIIA it only needs to be analyzed once.
- This step is optional but can save time and processing
- By this point, you have determined dozens, hundreds, or maybe thousands of (distinct) SQL that contribute I/O to the impacted tables. Henceforth, we'll simply call this the **“IMPACTING WORKLOAD”**

Predictive Index Impact Analysis (PIIA) – Step 3

EXPLAIN the IMPACTING WORKLOAD

- For each (distinct) SQL within the Impacting Workload:
 - Set USE_INDEX = 'N' for ALL Contemplated Indexes
 - EXPLAIN the SQL statement to learn its current/original Timeron Cost (Explain Mode EVALUATE INDEXES).
 - Set USE_INDEX = 'Y' for the Indexes that you intend to create per your AIBA (4 out of 5 in our earlier example)
 - EXPLAIN the SQL statement to learn its forecasted/new Timeron Cost
 - Compute **Original** Timeron Cost – **New** Timeron Cost = Timeron **Savings** (or degradation if negative), and determine the Savings Percent. Savings% could be multiplied against workload execution totals to predict new relative weights (heaviness)
 - Tabulate the sums of all Original Timeron Costs and New Timeron Costs to understand overall workload impact

Predictive Index Impact Analysis (PIIA) ILLUSTRATED

- From earlier slides, there were 5 proposed indexes against 3 different tables.
 - Based on AIBA, we're assuming that 4 of the 5 indexes will be created: IDX1602060629500, IDX1602060630060, IDX1602060630030, and IDX1602060629480
 - For our Impacting Workload, for sake of example, we'll assume there are 10 distinct statements driving I/O to our 3 different tables. Each of these will be stored individually in a file **Snn.SQL** where "nn" is the distinct statement number. For convenience, our original heavy query will be contained within file S00.SQL.

Predictive Index Impact Analysis for S00.sql

Original: 187,411 New: 1,760 Savings: 185,651 99.06%

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S00.sql
SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd, v.verb_desc FROM DBIPOC.SUCCESSFUL_HITS_UW A, DBIPOC.VERB_DESCRIPTIONS U
inname = 'webnj1.bbh.com' and a.targetfile = '/blog/rss/Scott_Hayes_rss2.xml' and a.bytesxferd < (select avg(h.bytesxferd) from DBIPOC.S
TS UW B) and a.hittimestamp < '2011-12-31-21.35.43.304000' and a.actionverb = v.actionverb fetch first 100 rows only
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sq
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as
om Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type =
ith UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
1760.0701	136.4210	40762252.0000	0.0000

Predictive Index Impact Analysis for S01.sql

Original: 93,690 New: 26 Savings: 93,664 99.97%

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S01.sql
SELECT ACTIONVERB, TARGETFILE FROM DBIPOC.SUCCESSFUL_HITS_UW WHERE DOMAINNAME = :ls FETCH FIRST 10 ROWS ONLY
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm_cost,
from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'R'
with UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
25.7420	2.0000	242973.3750	0.0000

Predictive Index Impact Analysis for S02.sql

Original: 93,756 New: 39 Savings: 93,717 99.96%

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S02.sql
SELECT ACTIONVERB, TARGETFILE FROM DBIPOC.FAILED_HITS_UW WHERE DOMAINNAME = :ls FETCH FIRST 10 ROWS ONLY
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as com
on Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'R
ith UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
38.5943	3.0000	256244.2968	0.0000

Predictive Index Impact Analysis for S03.sql

Original: 93,677 New: 14,053 Savings: 85,624 91.40%

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S03.sql
SELECT IPADDR, TARGETFILE FROM DBIPOC.WEBSITE_DATA_TB WHERE HITTIMESTAMP = :ls ORDER BY TARGETFILE FETCH FIRST 10 ROWS ONLY
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as com
om Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'R
ith UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
14053.0556	1071.4849	1642515328.0000	0.0000

Predictive Index Impact Analysis for S04.sql

Original: 93,690 New: 4,192 Savings: 89,498 95.53%

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S04.sql
SELECT IPADDR, TARGETFILE FROM DBIPOC.SUCCESSFUL_HITS_UW WHERE HITTIMESTAMP = :ls ORDER BY TARGETFILE FETCH FIRST 10 ROWS ONLY
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm_cost
from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as h where explain_time = h.maxtime and operator_type = 'R'
with UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
4192.2255	1519.2368	69092672.0000	0.0000

Predictive Index Impact Analysis for S05.sql

Original: 27,483 New: 4,268 Savings: 23,215 84.47%

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX16020606294
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate index
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S05.sql
SELECT IPADDR, BYTESXFERD FROM DBIPOC.WEBSITE_DATA_TB WHERE TARGETFILE = :ls FETCH FIRST 10 ROWS ONLY
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.
```

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explains_Operator
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4)
om Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_t
ith UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
4268.2304	325.6221	484924704.0000	0.0000

Predictive Index Impact Analysis for S06.sql

Original: 93,677 New: 59 Savings: 93,618 99.94%

```
C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_N.sql
update advise_index set USE_INDEX = 'N'
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Update_Advise_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf S06.sql
SELECT IPADDR, ACTIONVERB, PROTOCOL FROM DBIPOC.WEBSITE_DATA_TB WHERE DOMAINNAME = :ls
SQL0217W The statement was not executed as only Explain information requests
are being processed. SQLSTATE=01604

C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shaves\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL>db2 -tvf Query_Timeron_Cost_from_Explain_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as com
on Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'F
ith UR
```

QUERY_TIMERON_COST	IO_COST	CPU_COST	COMM_COST
58.7012	4.5661	154283.2500	0.0000

Predictive Index Impact Analysis for S07.sql

Original: 40,330 New: 54 Savings: 40,276 99.87%

Predictive Index Impact Analysis for S08.sql

Original: 27,483 New: 4,268 Savings: 23,215 84.47%

Predictive Index Impact Analysis for S09.sql

Original: 93,756 New: 39 Savings: 93,717 99.96%

Predictive Index Impact Analysis The Grand Finale – Drum Roll Please!

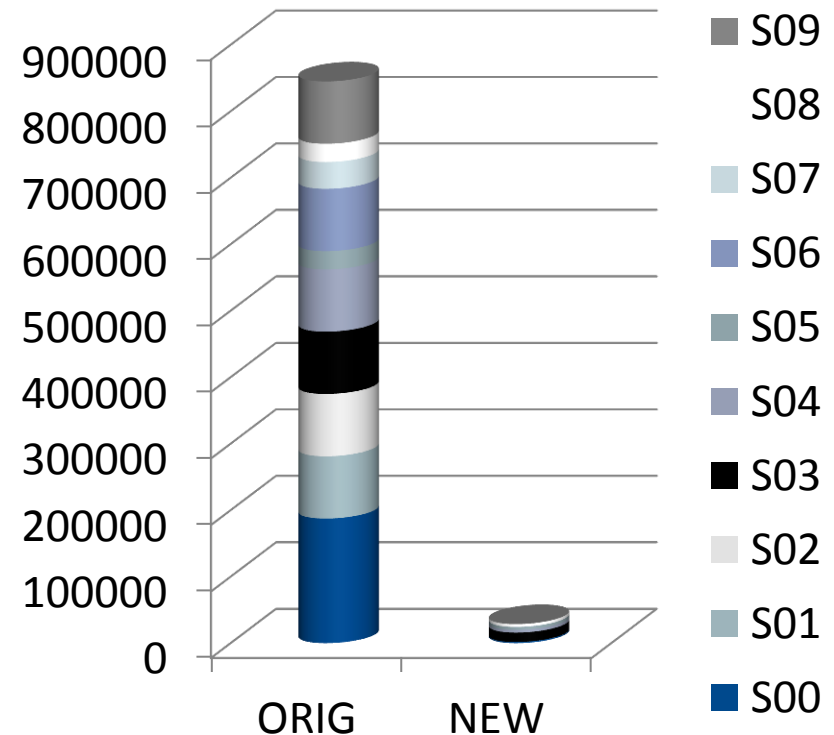


PIIA – Do you kill multiple birds with a few stones? Any adverse consequences? Safe to create indexes?

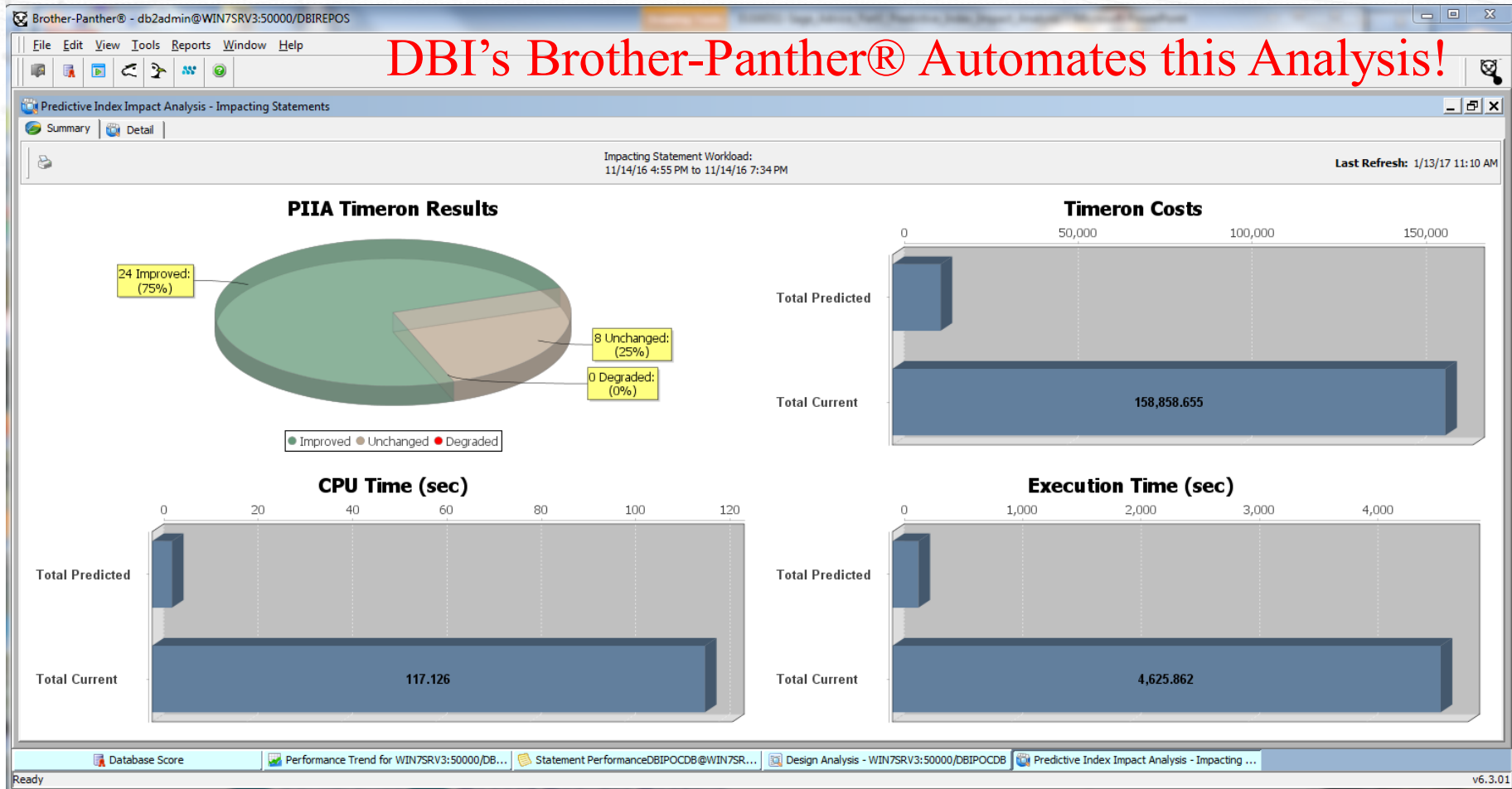
Impacting Workload

Query	OLD	NEW	DIFF
0	187411	1760	-185651
1	93690	26	-93664
2	93756	39	-93717
3	93677	14053	-85624
4	93690	4192	-89498
5	27483	4268	-23215
6	93677	59	-93618
7	40330	54	-40276
8	27483	4268	-23215
9	93756	39	-93717
Total	844953	28758	-816195

Off the chart savings!



COMMERCIAL BREAK- Folks! Don't Do this the Hard Way! It's Time Consuming and Error Prone!



COMMERCIAL BREAK- Folks! Don't Do this the Hard Way! It's Time Consuming and Error Prone!

Brother-Panther® - db2admin@WIN7SRV3:50000/DBIREPOS

DBI's Brother-Panther® Automates this Analysis!

Predictive Index Impact Analysis - Impacting Statements

Summary Detail

Impacting Statement Workload: 11/14/16 4:55 PM to 11/14/16 7:34 PM

From Table(s): DBIPOC.WEBSITE_DATA_TB

Last Refresh: 1/13/17 11:10 AM

Rows: 32

Follow Up	Stmt ID	Verb	Type	# Execs	Current Timeron Cost	Predicted Timeron Cost	Timeron Cost Improved	Delta Timeron Cost	Timeron Cost Savings %	Current CPU Time (sec)	Current % CPU Time	Predicted CPU Time (sec)	Predicted % CPU Time	Predicted CPU Time Saved (sec)	Current Exec Time (sec)	% Exec Time	Predicted Exec Time (sec)	Predicted % Exec Time	Predicted Exec Time Saved (sec)
	E58...	SELECT	DYNAMIC	600	6,533.8139	14.1243	Y	6,519.6896	99.784%	42.822275	36.561%	0.092570	2.210%	42.729705	1,546.372488	33.429%	3.342830	1.536%	1,543.0
	2F1...	SELECT	DYNAMIC	24	6,533.8144	21.1722	Y	6,512.6422	99.676%	2.059214	1.758%	0.006673	0.159%	2.052541	60.321051	1.304%	0.195465	0.090%	60.1
	39A...	SELECT	DYNAMIC	3	6,533.8144	21.1722	Y	6,512.6422	99.676%	0.249600	0.213%	0.000809	0.019%	0.248791	8.043290	0.174%	0.026064	0.012%	8.0
	3E4...	SELECT	DYNAMIC	3	6,533.8144	21.1722	Y	6,512.6422	99.676%	0.249603	0.213%	0.000809	0.019%	0.248794	7.031631	0.152%	0.022785	0.010%	7.0
	464...	SELECT	DYNAMIC	3	6,533.8144	21.1722	Y	6,512.6422	99.676%	0.202801	0.173%	0.000657	0.016%	0.202144	7.827198	0.169%	0.025363	0.012%	7.8
	037...	SELECT	DYNAMIC	60	6,533.8139	21.1718	Y	6,512.6421	99.676%	3.962425	3.383%	0.012840	0.306%	3.949585	154.342801	3.337%	0.500124	0.230%	153.8
	4E2...	SELECT	DYNAMIC	3	6,533.8139	21.1718	Y	6,512.6421	99.676%	0.140401	0.120%	0.000455	0.011%	0.139946	7.229600	0.156%	0.023426	0.011%	7.2
	ACE...	SELECT	DYNAMIC	3	6,533.8139	21.1718	Y	6,512.6421	99.676%	0.218402	0.186%	0.000708	0.017%	0.217694	7.296079	0.158%	0.023642	0.011%	7.2
	F56...	SELECT	DYNAMIC	3	6,533.8139	21.1718	Y	6,512.6421	99.676%	0.156001	0.133%	0.000505	0.012%	0.155496	7.742056	0.167%	0.025087	0.012%	7.7
	FF6...	SELECT	DYNAMIC	3	6,533.8139	21.1718	Y	6,512.6421	99.676%	0.156000	0.133%	0.000505	0.012%	0.155495	7.676111	0.166%	0.024873	0.011%	7.6
	11C...	SELECT	DYNAMIC	3	6,495.8676	14.1169	Y	6,481.7507	99.783%	0.156001	0.133%	0.000339	0.008%	0.155662	7.106429	0.154%	0.015444	0.007%	7.0
	AEC...	SELECT	DYNAMIC	3	6,495.8676	14.1169	Y	6,481.7507	99.783%	0.327603	0.280%	0.000712	0.017%	0.326891	8.410919	0.182%	0.018279	0.008%	8.3
	BA3...	SELECT	DYNAMIC	600	6,495.8676	14.1169	Y	6,481.7507	99.783%	37.705444	32.192%	0.081942	1.956%	37.623502	1,548.430197	33.473%	3.365068	1.546%	1,545.0
	29D...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.265201	0.226%	0.000753	0.018%	0.264448	9.366337	0.202%	0.026601	0.012%	9.3
	37B...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.140400	0.120%	0.000399	0.010%	0.140001	8.481645	0.183%	0.024088	0.011%	8.4
	721...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.156000	0.133%	0.000443	0.011%	0.155557	7.999267	0.173%	0.022718	0.010%	7.9
	B07...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.280800	0.240%	0.000797	0.019%	0.280003	9.282945	0.201%	0.026364	0.012%	9.2
	B9E...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.124800	0.107%	0.000354	0.008%	0.124446	7.414252	0.160%	0.021057	0.010%	7.3
	DBC...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.109202	0.093%	0.000310	0.007%	0.108892	7.770123	0.168%	0.022068	0.010%	7.7
	F00...	SELECT	DYNAMIC	3	6,495.8676	18.4486	Y	6,477.4190	99.716%	0.140402	0.120%	0.000399	0.010%	0.140003	8.196855	0.177%	0.023279	0.011%	8.1
	D21...	SELECT	DYNAMIC	249	6,488.8051	14.1093	Y	6,474.6958	99.783%	16.551706	14.132%	0.035990	0.859%	16.515716	676.053826	14.615%	1.470016	0.675%	674.5
	33E...	SELECT	DYNAMIC	3	6,917.5073	937.4976	Y	5,980.0097	86.447%	0.702005	0.599%	0.095139	2.271%	0.606866	9.615855	0.208%	1.303192	0.599%	8.3
	3D8...	SELECT	DYNAMIC	282	2,675.0151	14.1045	Y	2,660.9106	99.473%	6.427241	5.487%	0.033889	0.809%	6.393352	298.251284	6.447%	1.572584	0.722%	296.6
	5B6...	SELECT	DYNAMIC	15	6,507.8466	6,500.8247	Y	7.0219	0.108%	0.421202	0.360%	0.020748	10.043%	0.000454	36.147067	0.781%	36.108065	16.588%	0.0
	0F1...	SELECT	DYNAMIC	3	1,899.5625	1,899.5625	U	0.0000	0.000%	0.343202	0.293%	0.343202	8.192%	0.000000	16.768675	0.362%	16.768675	7.703%	0.0
	1C0...	SELECT	DYNAMIC	288	14.1073	14.1073	U	0.0000	0.000%	0.062401	0.053%	0.062401	1.489%	0.000000	5.000356	0.108%	5.000356	2.297%	0.0
	560...	SELECT	DYNAMIC	150	14.1073	14.1073	U	0.0000	0.000%	0.015600	0.013%	0.015600	0.372%	0.000000	1.987764	0.043%	1.987764	0.913%	0.0
	644...	SELECT	DYNAMIC	567	15.5587	15.5587	U	0.0000	0.000%	0.156000	0.133%	0.156000	3.724%	0.000000	43.009418	0.930%	43.009418	19.758%	0.0
	64D...	SELECT	DYNAMIC	597	14.0994	14.0994	U	0.0000	0.000%	0.109201	0.093%	0.109201	2.606%	0.000000	39.435365	0.852%	39.435365	18.116%	0.0

Scott Richard Hayes

DBI Software, @dbisoftware

sales@dbisoftware.com

[@srhayes](#)

[D11] Sage Advice Part 3: Predictive
Index Impact Analysis -- Know Before
you CREATE



*Please fill out your session
evaluation before leaving!*

