Optimization Hints Grow Up

Chad Reiber

BMC Software

Agenda

Optimization Hints Grow Up

What is optimization hints we know and love

List of features Db2 z/OS provides us to control access path selection

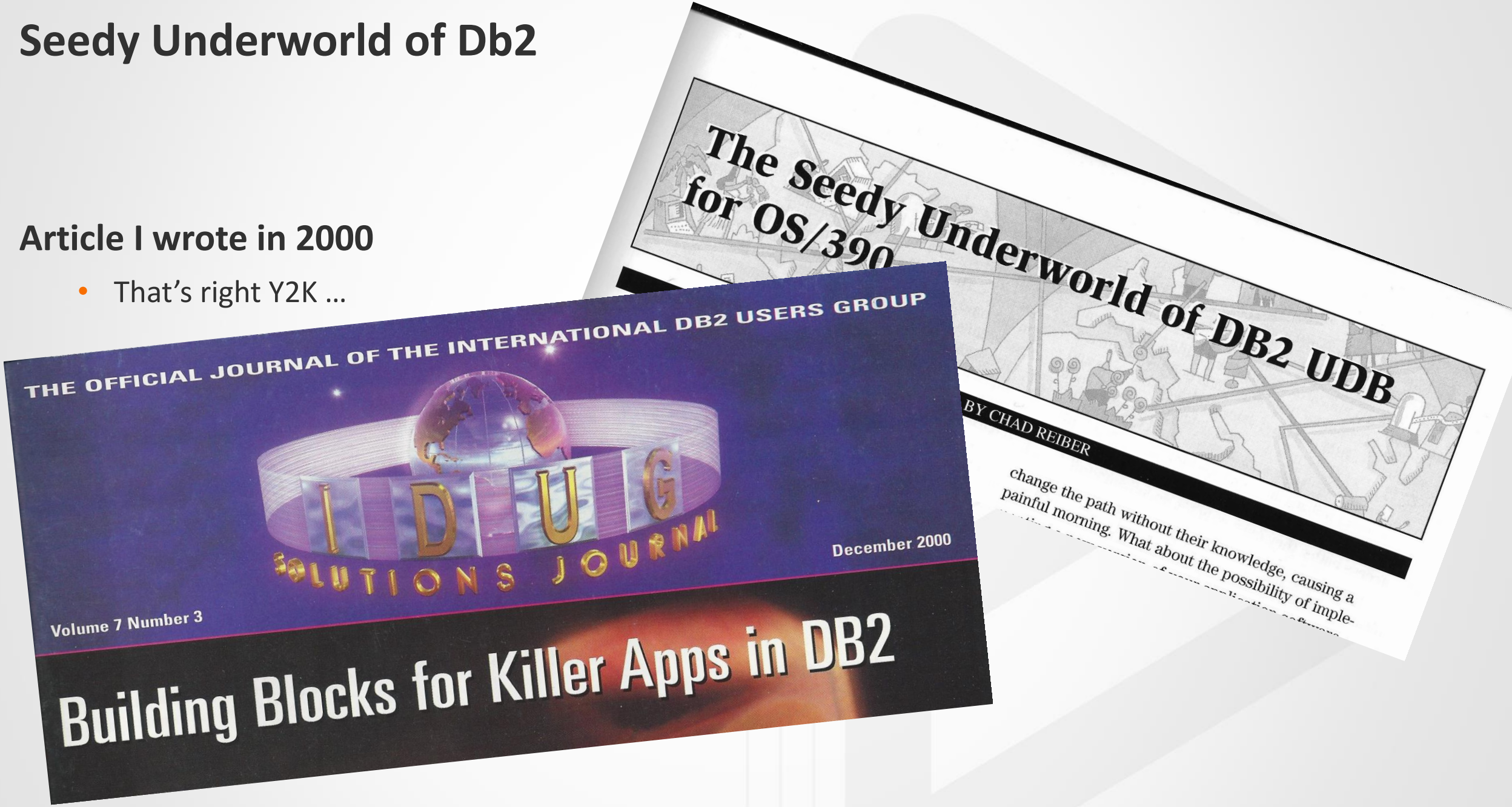
Influencers of Access Paths

Examples you can use to help your application performance

Seedy Underworld of Db2

Article I wrote in 2000

- That's right Y2K …



Where have we come in 20+ Years

Db2 Optimizer is still #1

- Better and Better access paths
- Tell us when statistics will help … please SYSSTATFEEDBACK and PROFILES

More options to control what happens

- Deployment of new code
- Release / Function Level upgrades

Manage Access Paths

- Avoid pain !!!

Manage Access Paths

Want great efficient access paths?

- Queries should use effective predicates
- Build Indexes to support the data access
- Collect Statistics on the application data that is correct (and you got it all)
- Perfect !!

Do you have a couple of hours to discuss this?

When that is out of your control or that just doesn't work …

Influence Access Paths

Prevent Access Path Changes

Influencing Statement Access Paths

Optimization Parameters at Statement Level

- ZPARM such as STARJOIN
- Runtime Reoptimization at run time

Override predicate selection at Statement Level

- Predicate Selectivity Overrides

Specify access path matching by Statement

- Statement-Level access paths

Specify access path in Plan Table

- Optimization Hints

OPTHINTS last on the list but first in our hearts …

Remember this is how Access Paths partied in V6 - 1999

If it is Db2 z/OS must be a ZPARM to control this

- OPTHINTS = YES

Controlled by a Plan Table

- The access path for a particular statement is inserted / update in the plan table
- Plan_Table Column: OPT_HINT is updated with a value

Example

Particular Statement in a Package does not get an acceptable Access Path
Add an Optimization Hint (IDUG1) on that statement to use previous optimization
Rebind Package (CRABCPK) attempting to use new Hint

Add OPTHINT (IDUG1) to the Package Qualifier's Plan Table

```
DJJ1 ----------------------------  Edit DB2 Table  -------------------------------
Command ===>                                                      Scroll ===> CSR
MKTCWR.PLAN_TABLE (1/7)
NAME                 TYPE      LENGTH NULL VALUE
WHEN_OPTIMIZE        CHAR           1
QBLOCK_TYPE          CHAR           6       NCOSUB
BIND_TIME            TIMESTMP      26       2014-06-18-13.54.49.019168
OPTHINT              VARCHAR      128       'IDUG1'


HINT_USED            VARCHAR      128       ''


PRIMARY_ACCESSTYPE CHAR             1
PARENT_QBLOCKNO      SMALLINT       2             1
TABLE_TYPE           CHAR           1 N     T
TABLE_ENCODE         CHAR           1       E
TABLE_SCCSID         SMALLINT       2            37
TABLE_MCCSID         SMALLINT       2            -2
TABLE_DCCSID         SMALLINT       2            -2
ROUTINE_ID           INTEGER        4                   0
CTEREF               SMALLINT       2             0
STMTTOKEN            VARCHAR      240 Y
```

Rebind the Package with OPTHINT Parameter

```
---------------------------------  DSN Commands  ------------------------------
DSN SYSTEM(DJJ1)
REBIND PACKAGE(MKTCWR_COLLID.CRABCPK.(CMP-2601))  ENABLE(*)+
     OWNER(MKTCWR)          QUALIFIER(MKTCWR)     VALIDATE(BIND)+
     CURRENTDATA(NO)        ISOLATION(CS)         EXPLAIN(YES)+
     DEGREE(1)              KEEPDYNAMIC(YES)      REOPT(NONE)+
     DBPROTOCOL(DRDA)       IMMEDWRITE(INHERITFROMPLAN)+
     OPTHINT('IDUG1')       ENCODING(37)          ROUNDING(HALFEVEN)+
     PLANMGMT(EXTENDED)     APRETAINDUP(YES)      APREUSE(NO)+
     APPLCOMPAT(V10R1)      DESCSTAT(YES)         CONCENTRATESTMT(NO)+
     ARCHIVESENSITIVE(NO)   BUSTIMESENSITIVE(NO)+
     SYSTIMESENSITIVE(NO)
END
EXIT CODE(&LASTCC)
END
```

```
DSNT222I  *DJJ1 DSNTBBP2 REBIND WARNING
           FOR PACKAGE = DJJ.MKTCWR_COLLID.CRABCPK.
           USE OF OPTHINT RESULTS IN
           1 STATEMENTS WHERE OPTHINT FULLY APPLIED
           0 STATEMENTS WHERE OPTHINT NOT APPLIED
             OR PARTIALLY APPLIED
           13 STATEMENTS WHERE OPTHINT IS NOT FOUND
```

```
 FILE  COMMANDS  OPTIONS  HELP
--------------------------------------------------------------------------------------------------------------------------------
DJJ1                                   Explain Results for PACKAGE MKTCWR_COLLID.CRABCPK.CMP-2601
Command ===>                                                                                          Scroll ===> CSR
                                                                                                        More:     + >

   Actions: S H K R RS RW RI XD XS XP W P T C D U IM SA
   LBL    STMTNO      COST*RATE SQL-STATEMENT
   XD01*     472     663.575928 DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , B . ORDER_NBR , B . ORDER_A
   XS01*     472     405.719482 DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , B . ORDER_NBR , B . ORDER_A
   XP01      472       0.000000 DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , B . ORDER_NBR , B . ORDER_A
         COST*RATE QB PL MIX QTYPE   METH ACC
   XD01* 663.57592  1  1   0 UNIONA    3
   XD01* 24.567215  3  1   0 NCOSUB    0 R
   XD01*  0.022927  4  1   0 CORSUB    0 I
   XD01* 323.23291  5  1   0 NCOSUB    0 R
   XD01*  0.079407  6  1   0 CORSUB    0 I
   XS01* 405.71948  1  1   0 UNIONA    3
   XS01* 24.544296  3  1   0 NCOSUB    0 R
   XS01*  0.022927  4  1   0 CORSUB    0 I
   XS01* 65.398941  5  1   0 NCOSUB    0 I
   XS01*  0.079407  6  1   0 CORSUB    0 I
   XP01             1  1   0 UNIONA    3
   XP01             1  1   0 UNIONA    3
   XP01             2  1   0 NCOSUB    0 R
   XP01             4  1   0 CORSUB    0 I
   XP01             5  1   0 NCOSUB    0 I
   XP01             6  1   0 CORSUB    0 I
```

```
Explain Results for PACKAGE MKTCWR_COLLID.CRABCPK.CMP-2601
                                                                                                                    Sc

XD XS XP W P T C D U IM SA
S.U.*RATE CAT REASON          PROCMS
     3636 A                       99
     2630 B   OPTIMIZA            72

JPG SCPG SNPG PAR MERGE PRANGE JTYPE GME      OPTHINT  USED PTYPE CORR WHEN BIND_TIME
                                     DJJ1                                   2021-03-29-
                                     DJJ1                         A         2021-03-29-
                                     DJJ1                         B         2021-03-29-
                                     DJJ1                         A         2021-03-29-
                                     DJJ1                         B         2021-03-29-
                                     DJJ1                         B         2021-03-29-
                                     DJJ1                         A         2021-03-29-
                                     DJJ1              IDUG                 2021-03-29-
                                     DJJ1              IDUG       A         2021-03-29-
                                     DJJ1              IDUG       B         2021-03-29-
                                     DJJ1              IDUG       A         2021-03-29-
                                     DJJ1              IDUG       B         2021-03-29-
                                     DJJ1              IDUG       B         2021-03-29-
                                     DJJ1              IDUG       A         2021-03-29-
                                     DJJ1              EXPL                 2021-03-29-
```

Proof Points

How do I know if this is taking affect

SYSPACKAGE

- OPTHINT column contains the IDUG1

SYSPACKSTMT

- ACCESSPATH column contains ‘H’

PLAN_TABLE

- We updated OPT_HINT with IDUG1
- HINT_USED column will also have IDUG1 if the hint was valid

Some Good News and some Bad News

Optimizer Hints works for Static and Dynamic Statements

- Dynamic SQL uses Special Register
  - SET CURRENT OPTIMIZATION HINT IDUG1
  - Uses CURRENT SQLID to find the Plan Table
  - +394 Statement Hint found and used, +395 invalid Hint, not used

How is Db2 match the statement to the plan table (static or dynamic)

- Matching is based on following key fields
- QUERYNO, APPLNAME,PROGNAME,VERSION,COLLID,OPTHINT
- Ugh … so if keys don't match?
  - You can code QUERYNO in your statements

```
SELECT IND_NAME FROM CR_INDIVIDUAL
    WHERE IND_ID = 12345 AND STATE = 'NJ'
        QUERYNO 99;
```

Let's do better - Specify access path matching by Statement

PreReq

- ZPARM OPTHINTS = YES
- New user table required DSN_USERQUERY_TABLE
  - Lets call this the BIG BOSS TABLE
- Statement to be "matched" must be created by BIND PACKAGE
  - NO Create Function, Create Trigger, or Create Procedure
  - Still supports Dynamic and Static

Scope of the Optimization (new OPT Hint!)

- System Wide
- Any Version of a Collection and package
- Specific Version of a Collection and package
- Controlled by values in DSN_USERQUERY_TABLE

SYSIBM.SYSQUERY
SYSIBM.SYSQUERYPLAN
SYSIBM.SYSQUERYOPTS
SYSIBM.SYSQUERYPREDICATE
SYSIBM.SYSQUERYSEL

What can we change by using this new Big Boss Table ?

DSN_USERQUERY_TABLE

All at the individual SQL Statement Level

1. Specifying Optimization Parameters
2. Specifying Access Paths
3. Overriding Predicate Selectivities

```
QUALIFIER: TABLE=MKTCWR.DSN_USER..RY_TABLE
Cmd   Column Name        ColNo Datatyp Length
----v----1----v----2----v----3----v----4----v
      QUERYNO                1 INTEGER      4
      SCHEMA                 2 VARCHAR    128
      HINT_SCOPE             3 SMALLINT     2
      QUERY_TEXT             4 CLOB         4
      QUERY_ROWID            5 ROWID       17
      QUERYID                6 BIGINT       8
      USERFILTER             7 CHAR         8
      OTHER_OPTIONS          8 CHAR       128
      COLLECTION             9 VARCHAR    128
      PACKAGE               10 VARCHAR    128
      VERSION               11 VARCHAR    128
      REOPT                 12 CHAR         1
      STARJOIN              13 CHAR         1
      MAX_PAR_DEGREE        14 INTEGER      4
      DEF_CURR_DEGREE       15 CHAR         3
```

Insert your “statement” in the DSN_USERQUERY_TABLE

- QUERYNO – Used as primary key
- SCHEMA – if SQL contains unqualified stuff – what is the Schema
- QUERY_TEXT – you guessed it, this is the statement you want to influence
- HINT_SCOPE – two choices, its binary
  - 0 – system wide
    - Only uses the statement and the schema
  - 1 – Package-level
    - Uses the Collection, Package, Version + the statement
- COLLECTION
- PACKAGE
- VERSION
- SELECTVTY_OVERRIDE (Y or N)
- ACCESSPATH_HINT (Y or N)
- OPTION_OVERRIDE (Y or N)

Insert your “statement” in the DSN_USERQUERY_TABLE

Some information about that SQL statement you are adding …

- Db2 modifies the statement by removing non-important information
  - Application Defaults are important CCSID, DECIMAL POINT, STRING DELIMITER
- While not required
  - PACKAGE, COLLECTION, VERSION will allow Db2 to use SYSIBM.SYSPACKSTMT
  - If Multiple Versions and * is specified Db2 will use the smallest VERSION value
- Where do I get the QUERY_TEXT to use?
  - Static
    - DBRM / SYSIBM.SYSPACKSTMT
  - Dynamic
    - Dynamic Statement Cache
  - Object Names and SQL Keywords need to be in UpperCase

1. Specifying Optimization Parameters

REOPT Bind Option

Subsystem Parameters

- STARJOIN, PARAMDEG, CDSSRDEF, SJTABLES

OPTION_OVERRIDE = Y

Example

Want Db2 to use REOPT(ALWAYS) for a particular statement

- Update DSN_USERQUERY_TABLE with statement, system wide
- BIND QUERY

bmc AMI Command Center for Db2 | Workspace | Apps

Signed in as MKTCWR

Reiber-DBA*

DB2 Navigator | File Locator | Job Browser | **Performance** | Schema Management | Recovery Management | Scratchpad | Product Tools

- SQL Tuning
  - Tuning Session 1
  - Tuning Session 2
  - Tuning Session 3
  - Tuning Session 4
- DB2 Statement Cache
  - DJJ
    - Program Filter: CR*

DB2 Statement Cache for Program Filter: CR*

STMT_ID	PROGRAM_NAME	PRIMA
9402	CRBMDPK	MKTC
9403	CRBMDPK	MKTC

2 row(s) returned

DJJ
-04-06 09:11:26

NBR, B.ORD...
RY_KEY_A, ...

Properties

Attribute Name	Attribute Value
STMT_ID	9402
PROGRAM_NAME	CRBMDPK
EXPLAIN_TS	2021-04-06-09.11.26.760000
CACHE_TS	2021-04-06-09.09.24.524312
BIND_QUALIFIER	MKTCWR
CURSQLID	MKTCWR

```
SELECT   ORDER_NBR, B.ORDER_AMT
FROM     MKTCWR.CR_INDIVIDUAL A, MKTCWR.CR_ORDERS B
WHERE    A.CUST_PHONE_NBR LIKE ?
  AND    A.PRIMARY_KEY_A = B.PRIMARY_KEY_A
```

Close

bmc AMI Command Center for Db2 | Workspace | Apps

Signed in as MKTCWR

Reiber-DBA*

DB2 Navigator | File Locator | Job Browser | Performance | Schema Management | Recovery Management | Scratchpad | Product Tools

Current SQLID: USER

- Files
- DJJ
  - Tables
    - *INSERT for Table:*

INSERT for Table: MKTCWR.DSN_USERQUERY_TABLE*

Command Text | Output

```
  OTHER_PARMS,
  SELECTVTY_OVERRIDE,
  ACCESSPATH_HINT,
  OPTION_OVERRIDE
) VALUES (
 100, --INTEGER         "QUERYNO"
'MKTCWR', --VARCHAR(128)    "SCHEMA"
 0, --SMALLINT        HINT_SCOPE
'SELECT   ORDER_NBR, B.ORDER_AMT
 FROM     MKTCWR.CR_INDIVIDUAL A, MKTCWR.CR_ORDERS B
 WHERE    A.CUST_PHONE_NBR LIKE ?
   AND    A.PRIMARY_KEY_A = B.PRIMARY_KEY_A    ', --CLOB(2097152)   QUERY_TEXT
' ', --CHAR(128)       OTHER_OPTIONS
' ', --VARCHAR(128)    "COLLECTION"
' ', --VARCHAR(128)    "PACKAGE"
' ', --VARCHAR(128)    VERSION
'Y', --CHAR(1)         REOPT
' ', --CHAR(1)         STARJOIN
-1 , --INTEGER         MAX_PAR_DEGREE
' ', --CHAR(3)         DEF_CURR_DEGREE
-1 , --INTEGER         SJTABLES
' ', --VARCHAR(128)    OTHER_PARMS
'N', --CHAR(1)         SELECTVTY_OVERRIDE
'N', --CHAR(1)         ACCESSPATH_HINT
'Y'  --CHAR(1)         OPTION_OVERRIDE
)
```

Line: 6 Column: 3

INSERT for Table: MKTCWR.DSN_USERQUERY_TABLE*

Command Text | Output

INSERT INTO MKTCWR.DSN_USERQUERY_TABLE ("QUERYNO", "SCHEMA", HINT_SCOPE, QUERY_TEXT, OTHER_OPTIONS,

```
INSERT MKTCWR.DSN_USERQUERY_TABLE executed, SQL rc = 0
Rows affected = 1
```

2. Specifying Access Paths at Statement Level

Need DSN_USERQUERY_TABLE and PLAN_TABLE

ACCESSPATH_HINT = Y

- Can have OPTION_OVERRIDE as well

PLAN_TABLE Update

- DO NOT specify OPT_HINT
- BIND/REBIND Does not require OPTHINT keyword
- Use QUERYNO of Big Boss Table as the value QUERYNO in PLAN_TABLE

Example

Particular Statement in a Package does not get an acceptable Access Path

- Add an entry into Big Boss Table
- Add appropriate access path in PLAN_TABLE
- BIND QUERY
- Rebind Package (CRABCPK) use appropriate access path

bmc AMI Command Center for Db2 | Workspace | Apps

Reiber-DBA*

DB2 Navigator | File Locator | Job Browser | Performance | Schema Management | Recovery Management | Scratchpad

- Files
- DJJ
  - Tables
    - *INSERT for Table:*
    - INSERT for Table: MI

INSERT for Table: MKTCWR.DSN_USERQUERY_TABLE*

Command Text | Output

```
  "SCHEMA",
  HINT_SCOPE,
  QUERY_TEXT,
  OTHER_OPTIONS,
  "COLLECTION",
  "PACKAGE",
  VERSION,
  SELECTVTY_OVERRIDE,
  ACCESSPATH_HINT,
  OPTION_OVERRIDE
) VALUES (
 9999, --INTEGER         "QUERYNO"
'MKTCWR', --VARCHAR(128)    "SCHEMA"
 1, --SMALLINT        HINT_SCOPE
(select statement from sysibm.syspackstmt
  where collid = 'MKTCWR_COLLID'
    and name = 'CRABCPK' and and version = 'CMP-2601'
    and stmtno = 472'),  --CLOB(2M) QUERY_TEXT
'MKTCWR_COLLID', --VARCHAR(128)    "COLLECTION"
'CRABCPK', --VARCHAR(128)    "PACKAGE"
'*', --VARCHAR(128)    VERSION
' ', --VARCHAR(128)    OTHER_PARMS
'N', --CHAR(1)         SELECTVTY_OVERRIDE
'Y', --CHAR(1)         ACCESSPATH_HINT
'N'  --CHAR(1)         OPTION_OVERRIDE
)
```

bmc AMI Command Center for Db2 | Workspace | Apps

Signed in as MKTCWR

Reiber-DBA*

DB2 Navigator | File Locator | Job Browser | Performance | Schema Management | Recovery Management | Scratchpad | Product Tools

Current SQLID: USER

- Files
- DJJ
  - Tables
    - *INSERT*
    - INSERT
    - *SELE*

SELECT for Table: MKTCWR.DSN_USERQUERY_TABLE*

Last executed on DJJ

Command Text | Output | Result Set 1

QUERYNO	SCHEMA	HINT_SCOPE	SELECTVTY_OVERR...	ACCESSPATH_HINT	OPTION_OVERRIDE	QUERY_TEXT	QUER
9999	MKTCWR	1	N	Y	N	DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRI...	X'001
100	MKTCWR	0	N	N	Y	SELECT ORDER_NBR, B.ORDER_AMT. FROM MKTCWR.CR_INDIVIDUA...	X'001

SELECT for Table: MKTCWR.PLAN_TABLE*

Last executed on DJJ

Command Text | Output | Result Set 1

QUERYNO	QBLOCKNO	PROGNAME	OPTHINT	PLANNO	METHOD	TNAME	TABNO	ACCESSTYPE	MATCHCOLS	ACCESSCREATOR
9999	5	CRABCPK		1	0	PD_ORDERS	5	I	0	MKTCWR
9999	3	CRABCPK		1	0	PD_INDIVIDUAL	3	R	0	
9999	2	CRABCPK		2	1	PD_ORDERS	2	I	2	MKTCWR
9999	2	CRABCPK		1	0	PD_INDIVIDUAL	1	R	0	
9999	1	CRABCPK		1	3		0		0	
9999	4	CRABCPK		1	0	PD_ORDERS	4	I	2	MKTCWR
9999	6	CRABCPK		1	0	PD_INDIVIDUAL	6	I	2	MKTCWR

BIND QUERY … always the next step

DSN_USERQUERY_TABLE
PLAN_TABLE
DSN_PREDICAT_TABLE
DSN_PREDICATE_SELECTIVITY

BIND QUERY

SYSIBM.SYSQUERY
SYSIBM.SYSQUERYPLAN
SYSIBM.SYSQUERYOPTS
SYSIBM.SYSQUERYPREDICATE
SYSIBM.SYSQUERYSEL

Takes the changes you have implemented in the Big Boss Table

Inserts them where they will do the most good.

BIND QUERY LOOKUP (YES|NO) EXPLAININPUTSCHEMA ('schema name')

- Only works if ZPARM OPTHINTS is YES
- BIND QUERY processes the entire Big Boss Table
- BIND QUERY LOOKUP(NO) does the validation and inserts into "Query" Tables
- BIND QUERY LOOKUP (YES)
  - Reads the Big Boss Table – DSN_USERQUERY_TABLE
  - For every match it finds in the "Query" Tables
  - Updates the QUERYID column in the DSN_USERQUERY_TABLE
  - No rows are inserted

```
DJJ1-R ---------------------------  Confirm DSN  --------------------- Row 1 of 6
Command ===>                                                Scroll ===> CSR
                                                                          02
Edit DSN Commands  . . . . . . . .  *  (Y/E/N E-remove TSO WRITEs and DSN output)
Save DSN in SQL table  . . . . .    Y  (A/R/Y/N A-append, R-replace, Y-append)
Execute DSN Commands now . . . .    Y  (Y/N)        TSO id used : MKTCWR
Create batch job . . . . . . . .    N  (Y/N)
Name of saved DSN  . . . . . . .    BINDQUERY
-----------------------------  DSN Commands  ------------------------------
CONTROL ASIS
DSN SYSTEM(DJJ1)
BIND QUERY LOOKUP(NO) EXPLAININPUTSCHEMA('MKTCWR')
END     -------------------------------  DSN Commands  ------------------------------
EXIT CO DSNT280I  *DJJ1 BIND QUERY FOR QUERYNO = 100 SUCCESSFUL
END     DSNT280I  *DJJ1 BIND QUERY FOR QUERYNO = 9999 SUCCESSFUL
******* DSNT290I  *DJJ1 BIND QUERY COMMAND COMPLETED                          *
        CONTROL ASIS
        CONTROL ASIS
        DSN SYSTEM(DJJ1)
        BIND QUERY LOOKUP(NO) EXPLAININPUTSCHEMA('MKTCWR')
        END
        EXIT CODE(&LASTCC)
        END
```

DSNT281I BIND QUERY FOR QUERYNO=xxxx NOT SUCCESSFUL REASON CODE ….

Some notes on BIND QUERY

Reads every row in DSN_USERQUERY_TABLE

- Might not want to leave stuff around
- If using ACCESSPATH – tries to match to PLAN_TABLE
  - Too much data, might want to separate
  - Reason for EXPLAININPUTSCHEMA

Does require a high authority

- SYSADM, SYSOPR,SYSCTRL, System Level DBADM, SQLADM

FREE QUERY

- Multiple ways to select what you want to free
- USERFILTER (group optimization)
- Package, QUERYID, ALL
- REBIND …

That was just the set up … when do I get new access paths?

Static SQL Statements

- Rows in “query” tables are validated and applied when you REBIND the package containing the statements

Dynamic SQL Statements

- Validated and Enforced when the statements are prepared
- Check for SQLCODE +395 – something wrong
  - ZPARM SUPPRESS_HINT_SQLCODE_DYN

```
23 'SELECT   ORDER_NBR, B.ORDER_AMT
24  FROM     MKTCWR.CR_INDIVIDUAL A, MKTCWR.CR_ORDERS B
25  WHERE    A.CUST_PHONE_NBR LIKE ?
26    AND    A.PRIMARY_KEY_A = B.PRIMARY_KEY_A    ', --CLOB(2097152)   QUERY_TEXT
```

```
BMCSftwr.SQMCACTY     --        SQL STATEMENT TEXT        --        03/30 1
   ACTIONS FOR +: T-DETAIL   E-ERRORS   H-HEADER   O-OBJECTS
           FOR *: X-EXPLAIN SQL TEXT
   SUBSYS: DJJ1   CORRID:              PLAN:          CLNTAP:
   CONNID:          USER:            SECTNO:        8 CLNTID:
   APPGRP:                                            CLNTWS:
                 STMT   +-------- SQL --------+ +- TOTAL IN-SQL TIME -+
   PROGRAM        NO.   CALLS  OPEN  FETCH ERRS ELAPSED     CPU
   --------     ------- ------ ----- ----- ---- ----------- -----------
o  CRBMDPK          625      3      1      0    0 00:01.03500 00:00.12284
   CALL TYPE: CURSOR
   STMT TYPE: DYNAMIC

   DYNAMIC SQL STATEMENT TEX
*    SELECT    ORDER_NBR, B.O
     FROM      MKTCWR.CR_INDI
     WHERE     A.CUST_PHONE_N
       AND     A.PRIMARY_KEY_
```

```
BMCSftwr.SQMCOBJS     --    SQL STMT AND OBJECT DETAIL    --        03/30 14
   ACTIONS:  H-HEADER
   DB2: DJJ1 DSGRP: DSNDJJ    PLAN:                PROG: CRBMDPK
     CORRID:                CONNID:                STMT:       625
   SQL CALLS:          3    STATEMENT TYPE: DYNAMIC       CALL TYPE: CURS
   SQL ELAPSED TIME: 00:01.035
                                                BPOOL  +---------- GETPAGE --
   CREATOR   NAME                 TY BPOOL  HRATIO NUMBER     TIME      %
   --------- ------------------ -- ------ ------ ------ --------------
+  SYSIBM    DSNDSX01             I  BP0     100 %      4  00.0000    0.0
+  SYSIBM    DSNDRX01             I  BP0     100 %      2  00.0000    0.0
+  SYSIBM    DSNQPX01             I  BP0     100 %      4  00.0000    0.0
+  SYSIBM    SYSTABLES            T  BP0     100 %      2  00.0000    0.0
+  SYSIBM    SYSKEYS              T  BP0     100 %      2  00.0000    0.0
+  SYSIBM    SYSQUERY             T  BP8K0   100 %      4  00.0000    0.0
+  SYSIBM    SYSINDEXES           T  BP0     100 %      2  00.0000    0.0
+  SYSIBM    SYSQUERYOPTS         T  BP0     100 %      2  00.0000    0.0
```

No OPTHINT in Bind Statement

```
DSN SYSTEM(DJJ1)
REBIND PACKAGE(MKTCWR_COLLID.CRABCPK.(CMP-2500))  ENABLE(*)+
     OWNER(MKTCWR)          QUALIFIER(MKTCWR)     VALIDATE(BIND)+
     CURRENTDATA(NO)        ISOLATION(CS)         EXPLAIN(YES)+
     DEGREE(1)              KEEPDYNAMIC(YES)      REOPT(NONE)+
     DBPROTOCOL(DRDA)       IMMEDWRITE(INHERITFROMPLAN)+
     ENCODING(37)           ROUNDING(HALFEVEN)    PLANMGMT(EXTENDED)+
     APRETAINDUP(YES)       APREUSE(NO)           APPLCOMPAT(V10R1)+
     DESCSTAT(YES)          CONCENTRATESTMT(NO)+
     ARCHIVESENSITIVE(NO)   BUSTIMESENSITIVE(NO)+
     SYSTIMESENSITIVE(NO)
END
EXIT CODE(&LASTCC)
END
```

```
  FILE  COMMANDS  OPTIONS  HELP
--------------------------------------------------------------------------------
DJJ1          Explain Results for PACKAGE MKTCWR_COLLID.CRABCPK.CMP-2500
Command ===>                                                  Scroll ===> CSR
                                                                 More:     + >
   Actions: S H K R RS RW RI XD XS XP W P T C D U IM SA
   LBL    STMTNO      COST*RATE SQL-STATEMENT
__ XD01*     474     663.575928 DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_
__ XS01      474     405.719482 DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_
          COST*RATE QB PL MIX QTYPE  METH ACC MTCH IX TBNAME              IXNAME
__ XD01* 663.57592  1  1   0 UNIONA    3         0 N
__ XD01* 24.544296  2  1   0 NCOSUB    0 R       0 N  PD_INDIVIDUAL
__ XD01* 11.815338  2  2   0 NCOSUB    1 I       2 N  PD_ORDERS          CRORDEX
__ XD01* 24.567215  3  1   0 NCOSUB    0 R       0 N  PD_INDIVIDUAL
__ XD01*  0.022927  4  1   0 CORSUB    0 I       2 Y  PD_ORDERS          CRORDEX
__ XD01* 323.23291  5  1   0 NCOSUB    0 R       0 N  PD_ORDERS
__ XD01*  0.079407  6  1   0 CORSUB    0 I       2 N  PD_INDIVIDUAL      CRINDEX
__ XS01  405.71948  1  1   0 UNIONA    3         0 N
__ XS01  24.544296  2  1   0 NCOSUB    0 R       0 N  PD_INDIVIDUAL
         BMC184738I-DB2 used your Hint to determine the access path for this
                   query. The identifier used was SYSQUERYPLAN 225.
__ XS01  11.815338  2  2   0 NCOSUB    1 I       2 N  PD_ORDERS          CRORDEX
__ XS01  24.544296  3  1   0 NCOSUB    0 R       0 N  PD_INDIVIDUAL
```

DSN_USER_QUERY
QUERYNO = 100
QUERYNO = 9999

PLAN_TABLE
QUERYNO = 9999

BIND QUERY LOOKUP(NO)

PREPARE
DYNAMIC SQL

REBIND PACKAGE

PLAN_TABLE
QUERYNO = 474
HINT_USED updated

SYSIBM.SYSQUERY
QUERYID = 224
QUERYID = 225

SYSIBM.SYSQUERYOPTS
QUERYID = 224

SYSIBM.SYSQUERYPLAN
QUERYID = 225

There is one more Access Path Influencer …
3. Overriding Predicate Selectivity

Overriding Predicate Selectivities … what?

Allow users to set Filter Factors for certain predicates

- Tells DB2 Optimizer % of rows when predicate is applied
- For example: FF .1 says 10% of rows qualify
- We like small – good index choice

Sometimes can't get a Filter Factor / Default

- Host Variables / Parameter Markers
- Expressions, Subqueries

Uses the Big Boss Table

DSN_USERQUERY_TABLE
DSN_PREDICAT_TABLE
DSN_PREDICATE_SELECTIVITY

BIND QUERY populates

SYSIBM.SYSQUERY,
SYSIBIM.SYSQUERYPREDICATE
SYSIBM. SYSQUERYSEL

To use this one …
need to understand what Db2 does for us

Run an explain on the object you are interested in changing FF

Suggest create a new

- PLAN_TABLE, DSN_PREDICAT_TABLE, DSN_PREDICATE_SELECTIVITY

```
XD01     474     66.357593 DECLARE INDORD CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , B . ORDER_NBR , B . ORDER_A
      QB LEFT-HAND          TYPE     RIGHT-HAND STG1 MISMATCH JOIN KEY       FILTER ORD# STAGE     PD NG AP OR TEXT
       2 CUST_PHONE_NBR     LIKE     VALUE      Y             N    N       0.100000    1 STAGE1       N     U  "A"."CUST_PHONE_N
       2 PRIMARY_KEY_A      EQUAL    PRIMARY_KE Y             Y    Y       0.058824    1 MATCHING     N     U  "A"."PRIMARY_KEY_
       2 PRIMARY_KEY_B      EQUAL    PRIMARY_KE Y             Y    Y       0.004650    2 MATCHING     N     U  "A"."PRIMARY_KEY_
       3                    NOTEXIST CORSUB     N             N    N       0.500000    2 STAGE2       N     U   NOT EXISTS(SELEC
       3 CUST_PHONE_NBR     LIKE     VALUE      Y             N    N       0.100000    1 STAGE1       N     U  "A"."CUST_PHONE_N
       4 PRIMARY_KEY_A      EQUAL    PRIMARY_KE Y             N    Y       0.058824    1 MATCHING     N     U  "B"."PRIMARY_KEY_
       4 PRIMARY_KEY_B      EQUAL    PRIMARY_KE Y             N    Y       0.004650    2 MATCHING     N     U  "B"."PRIMARY_KEY_
       5                    NOTEXIST CORSUB     N             N    N       0.500000    1 STAGE2       N     U   NOT EXISTS(SELEC
       6 PRIMARY_KEY_A      EQUAL    PRIMARY_KE Y             N    Y       0.058824    1 MATCHING     N     U  "A"."PRIMARY_KEY_
       6 PRIMARY_KEY_B      EQUAL    PRIMARY_KE Y             N    Y       0.004650    2 MATCHING     N     U  "A"."PRIMARY_KEY_
       6 CUST_PHONE_NBR     LIKE     VALUE      Y             N    N       0.100000    3 STAGE1       N     U  "A"."CUST_PHONE_N
```

After Explain DSN_PREDICATE_SELECTIVITY

```
DJJ1 ------------------------------------------------- Browse DB2 Table -------------------------
Command ===>
MKTCWR.DSN_PREDICATE_SELECTIVITY (1/21)
******      QUERYNO QBLOCKNO      PREDNO INSTANCE            SELECTIVITY        WEIGHT ASSUMPTION
000001          449        2           1        0                      1             1 'NORMAL'
000002          449        2           2        0    0.099999964237213135             1 'NORMAL'
000003          449        2           3        0    0.010101009160280228             1 'NORMAL'
000004          449        2           4        0  0.00025375885888934135             1 'NORMAL'
000005          449        2          16        0  2.5632216420490295E-06             1 'NORMAL'
000006          449        2          18        0  2.5632216420490295E-06             1 'NORMAL'
000007          449        3           5        0                      1             1 'NORMAL'
000008          449        3           6        0                    0.5             1 'NORMAL'
000009          449        3           7        0    0.099999964237213135             1 'NORMAL'
000010          449        3          17        0    0.050001136958599091             1 'NORMAL'
000011          449        4           8        0                      1             1 'NORMAL'
000012          449        4           9        0    0.010101009160280228             1 'NORMAL'
000013          449        4          10        0  0.00025375885888934135             1 'NORMAL'
000014          449        4          20        0  0.00025375885888934135             1 'NORMAL'
000015          449        5          11        0                    0.5             1 'NORMAL'
000016          449        6          12        0                      1             1 'NORMAL'
000017          449        6          13        0    0.010101009160280228             1 'NORMAL'
000018          449        6          14        0  0.00025375885888934135             1 'NORMAL'
000019          449        6          15        0    0.099999964237213135             1 'NORMAL'
000020          449        6          19        0  2.5632216420490295E-06             1 'NORMAL'
000021          449        6          21        0  2.5632216420490295E-06             1 'NORMAL'
****** ********************************************************** BOTTOM OF DATA ************************
```

```
DJJ1                                Explain Results for PACKAGE MKTCWR_COLLID.CRBMCPK.
Command ===>                                                                          Scroll ===> CSR
                                                                                         More:       >
  Actions: S H K R RS RW RI XD XS XP W P T C D U IM SA
  LBL   STMTNO      COST*RATE SQL-STATEMENT
  XD01     532  48905.417969 DECLARE INDINDH CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , A . CUST_PHONE_NBR , B . C
        DECLARE  INDINDH CURSOR FOR
        SELECT   A.PRIMARY_KEY_A
                , A.PRIMARY_KEY_B
                , A.CUST_PHONE_NBR
                , B.CUST_ENT_DATE
        FROM     CR_INDIVIDUAL A
        FULL
        JOIN     CR_HIST_IND B
          ON     A.PRIMARY_KEY_A = B.PRIMARY_KEY_A
        WHERE    B.FIRST_ORDER_DATE <(B.CUST_ENT_DATE + 60 DAYS)
        QB LEFT-HAND          TYPE     RIGHT-HAND STG1 MISMATCH JOIN KEY       FILTER ORD# STAGE     PD NG AP OR TEXT
         1 FIRST_ORDER_DATE   RANGE    COLEXP     N             N    N       0.333333    1 STAGE2       N     U  "B"."FIRST_ORDER_
         1 PRIMARY_KEY_A      EQUAL    PRIMARY_KE Y             Y    Y       0.010101    1 STAGE1       N     U  "A"."PRIMARY_KEY_
        COST*RATE QB PL MIX QTYPE  METH ACC MTCH IX TBNAME            IXNAME             NU J O G CU J O G LCK PRE CFE ADEG JDEG
  XD01  1726.8208  1  1   0 SELECT    0 R      0 N  CR_HIST_IND                         N  N N N N  N N N  IS S
  XD01  47178.597  1  2   0 SELECT    1 R      0 N  CR_INDIVIDUAL                       N  Y N N N  N N N  IS S
********************************************************** Bottom of Data **********************************************************
```

Example

Getting a high Filter Factor on a Range Predicate

Knowing the data this is inappropriate

Need to help the optimizer know how the data

```
DJJ1 ------------------------------------------------------- Browse DB2 Table -------------------------------
Command ===>
MKTCWR.DSN_PREDICATE_SELECTIVITY (1/3)
******      QUERYNO QBLOCKNO      PREDNO INSTANCE            SELECTIVITY        WEIGHT ASSUMPTION
000001          532        1           1        0                      1             1 'NORMAL'
000002          532        1           2        0     0.33333331346511841             1 'NORMAL'
000003          532        1           3        0    0.01010100916028022 8            1 'NORMAL'
****** ************************************************************** BOTTOM OF DATA ***************************************
```

Db2 Explain will populate (if exists):

- DSN_PREDICAT_TABLE
- DSN_PREDICATE_SELECTIVITY

Using that data make required changes

```
DJJ1 ------------------------------------------------  Edit DB2 Table  -------
Command ===>
MKTCWR.DSN_PREDICAT_TABLE (1/3)
******      QUERYNO       PREDNO TYPE     LEFT_HAND_SIDE
000001         2222            1 AND      ''
000002         2222            2 RANGE    'FIRST_ORDER_DATE'
000003         2222            3 EQUAL    'PRIMARY_KEY_A'
****** ************************************************************ BOTTOM OF DATA *****
```

```
DJJ1 ------------------------------------------------  Edit DB2 Table  ------------------------------------
Command ===>
MKTCWR.DSN_PREDICATE_SELECTIVITY (1/5)
******      QUERYNO QBLOCKNO      SECTNOI       PREDNO INSTANCE            SELECTIVITY         WEIGHT ASSUMPTION
000001         2222        1            5            1        0                      1              1 'NORMAL'
000002         2222        1            5            2        0     0.33333331346511841              1 'NORMAL'
*INS           2222        1            5            2        1                   0.05            .50 'OVERRIDE'
*INS           2222        1            5            2        2                   0.01            .25 'OVERRIDE'
000005         2222        1            5            3        0    0.01010100916028022 8              1 'NORMAL'
****** ************************************************************ BOTTOM OF DATA ************************************
```

- What am I updating?
- Insert new rows for this query
  - INSTANCE
  - SELECTIVITY
  - WEIGHT
  - ASSUMPTION

Key Numbers

.05/5% 50% of the time

.01/1% 25% of the time

Selectivity Instance

Selectivity Profile

```
DJJ1 ------------------------------------------------------  Edit DB2 Table  ------------------------------------
Command ===>
MKTCWR.DSN_PREDICATE_SELECTIVITY (1/5)
******      QUERYNO QBLOCKNO      SECTNOI       PREDNO INSTANCE            SELECTIVITY         WEIGHT ASSUMPTION
000001         2222        1            5            1        0                      1              1 'NORMAL'
000002         2222        1            5            2        0     0.33333331346511841              1 'NORMAL'
*INS           2222        1            5            2        1                   0.05            .50 'OVERRIDE'
*INS           2222        1            5            2        2                   0.01            .25 'OVERRIDE'
000005         2222        1            5            3        0    0.01010100916028022 8              1 'NORMAL'
****** ************************************************************ BOTTOM OF DATA ************************************
```

INSERT for Table: MKTCWR.DSN_USERQUERY_TABLE*

Command Text | Output

```
  "COLLECTION",
  "PACKAGE",
  VERSION,
  SELECTVTY_OVERRIDE,
  ACCESSPATH_HINT,
  OPTION_OVERRIDE
) VALUES (
2222    , --INTEGER         "QUERYNO"
'MKTCWR', --VARCHAR(128)    "SCHEMA"
1       , --SMALLINT        HINT_SCOPE
(SELECT STATEMENT FROM SYSIBM.SYSPACKSTMT
WHERE COLLID = 'MKTCWR_COLLID' AND NAME = 'CRBMCPK'
AND STMTNO = 532), --CLOB(2097152)   QUERY_TEXT
'MKTCWR_COLLID', --VARCHAR(128)    "COLLECTION"
'CRBMCPK', --VARCHAR(128)    "PACKAGE"
'*', --VARCHAR(128)    VERSION
'Y'                  , --CHAR(1)         SELECTVTY_OVERRIDE
'N'                  , --CHAR(1)         ACCESSPATH_HINT
'N'                    --CHAR(1)         OPTION_OVERRIDE
)
```

- Don’t forget the Big Boss Table

```
Name of saved DSN  . . . . . .  BINDQUERY
-------------------------------  DSN Commands  ----------------------
DSNT280I  *DJJ1 BIND QUERY FOR QUERYNO = 100 SUCCESSFUL
DSNT280I  *DJJ1 BIND QUERY FOR QUERYNO = 9999 SUCCESSFUL
DSNT280I  *DJJ1 BIND QUERY FOR QUERYNO = 2222 SUCCESSFUL
DSNT290I  *DJJ1 BIND QUERY COMMAND COMPLETED
CONTROL ASIS
CONTROL ASIS
DSN SYSTEM(DJJ1)
BIND QUERY LOOKUP(NO) EXPLAININPUTSCHEMA('MKTCWR')
```

REBIND PACKAGE(MKTCWR_COLLID.CRBMCPK)

```
-------------------------------  DSN Commands  ----------------------------
DSNT297I  *DJJ1 DSNTBBP2 REBIND WARNING FOR
           PACKAGE = DJJ.MKTCWR_COLLID.CRBMCPK,
           USE OF SELECTIVITY OVERRIDES RESULTS IN:
           1 STATEMENTS WHERE SELECTIVITY OVERRIDES ARE FULLY
             APPLIED,
           0 STATEMENTS WHERE SELECTIVITY OVERRIDES ARE INVALID,
           13 STATEMENTS WHERE SELECTIVITY OVERRIDES ARE NOT FOUND.
```

```
  LBL   STMTNO     COST*RATE SQL-STATEMENT
  XD01*    532   4890.541797 DECLARE INDINDH CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , A . CUST_PHONE_NBR , B . C
  XS11*    532   1602.486719 DECLARE INDINDH CURSOR FOR SELECT A . PRIMARY_KEY_A , A . PRIMARY_KEY_B , A . CUST_PHONE_NBR , B . C
       BMC184738I-DB2 used your Hint to determine the access path for this query. The identifier used was SYSQUERYSEL 232.
```

CRBMCPK STMT 532

PLAN_TABLE
QUERYNO = 532

DSN_USER_QUERY
QUERYNO = 2222

DSN_PREDICAT_TABLE
QUERYNO = 2222

DSN_PREDICATE_SELECTIVITY
QUERYNO = 2222

BIND QUERY LOOKUP(NO)

PREPARE
DYNAMIC SQL

SYSIBM.SYSQUERY
QUERYID = 232

REBIND PACKAGE

PLAN_TABLE
QUERYNO = 532
HINT_USED updated

SYSIBM.SYSQUERYPREDICATE
QUERYID = 232

SYSIBM.SYSQUERYSEL
QUERYID = 232

Brings us to protection … a little insurance we call in the biz

Prevent Access Path Changes

Extended Plan Management Policy – REBIND PLANMGMT(EXTENDED)

- REBIND SWITCH
- APREUSE(WARN|ERROR)
  - Create Hints try to reuse old access paths
  - APREUSE can be the HINT_USED
  - SYSPACKSTMT column ACCESSPATH = 'A'
- APCOMPARE(WARN|ERROR)

Dynamic SQL plan Stability

SYSIBM.SYSDYNQRY

- Identifying dynamic SQL statements to stabilize
- Stabilizing Access paths for dynamic SQL Statements
  - START DYNQUERYCAPTURE / STOP DYNQUERYCAPTURE
- Invalidation of stabilized SQL

Db2 applies only one method – order the are considered

1. PLAN_TABLE access path hints
2. Statement-level access paths or parameters for a specific version, collection, and package.
3. Statement-level access paths or parameters for a specific collection and package.
4. Statement-level access paths or parameters that have a system-wide scope.
5. Statement-level access paths that are created internally by Db2 for access path reuse
6. Statement-level predicate selectivity overrides

Couple of final thoughts

- Might want to know how to do this before there is a crisis
- These “hints” can be come stale
  - Want to monitor and limit the use
  - With OPTHINT not required on REBIND – need to mine the catalog
- Good practice to manage Big Boss Table – DSN_USERQUERY_TABLE
  - Have a best practice
    - Remove once bound / move to history table

Thank You

About BMC

BMC helps customers run and reinvent their businesses with open, scalable, and modular solutions to complex IT problems. Bringing both unmatched experience in optimization and limitless passion for innovation to technologies from mainframe to mobile to cloud and beyond, BMC helps more than 10,000 customers worldwide reinvent, grow, and build for the future success of their enterprises, including 92 of the Forbes Global 100.

BMC— www.bmc.com

bmc | Customer Experience

BMC—The Multi-Cloud Management Company.
BMC, BMC Software, the BMC logo, and the BMC Software logo are the exclusive properties of BMC Software Inc., are registered or pending registration with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners. ©Copyright 2019 BMC Software, Inc.