An Hour of DB2 Tips

Craig S. Mullins

Mullins Consulting, Inc.

15 Coventry Court

Sugar Land, TX 77479

http://www.craigsmullins.com

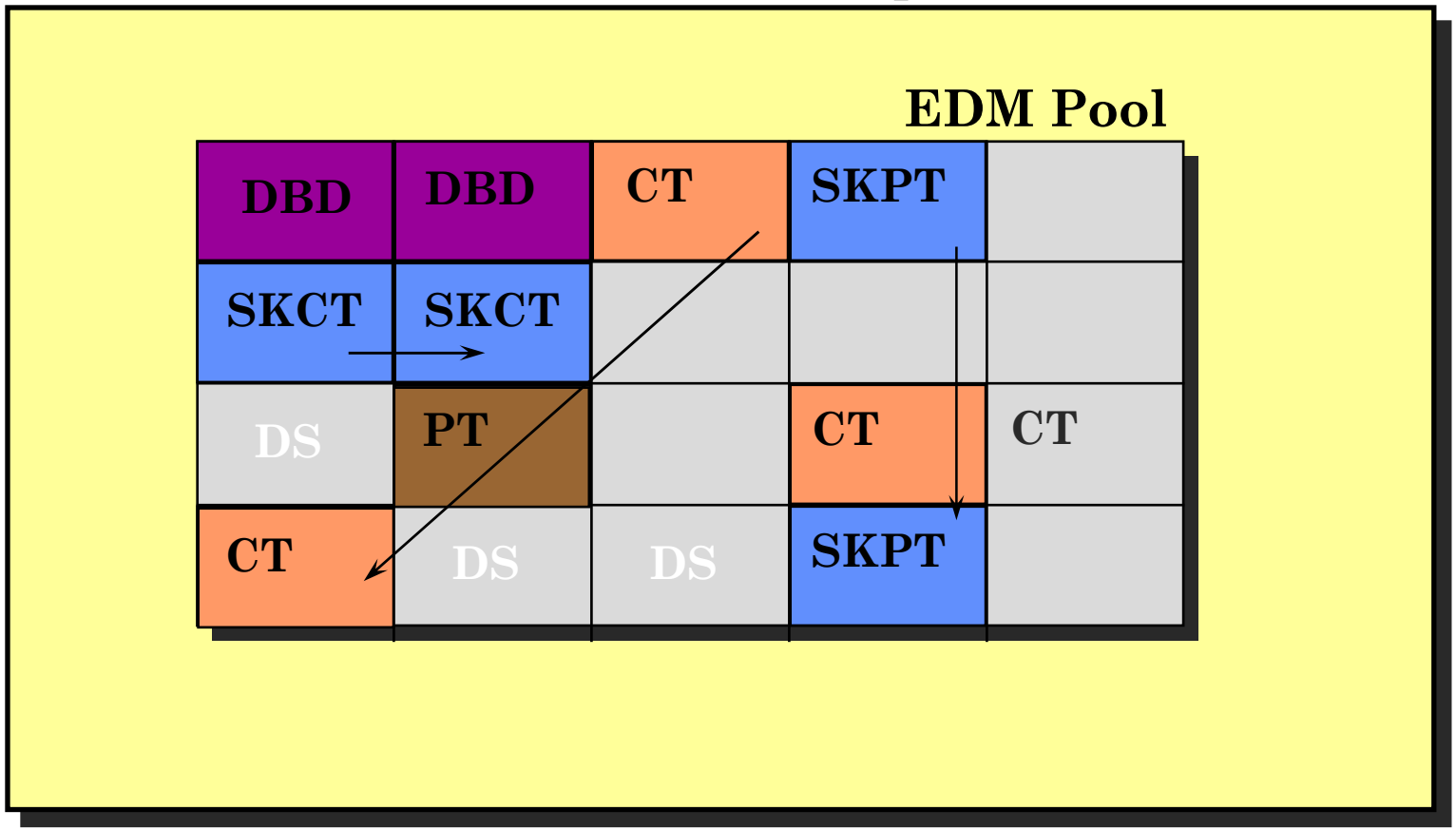
http://www.CraigSMullins.com/Hour.pdf

Agenda

A Bunch of DB2 Tips, Techniques, Thoughts, and Ideas

Managing the EDM Pool

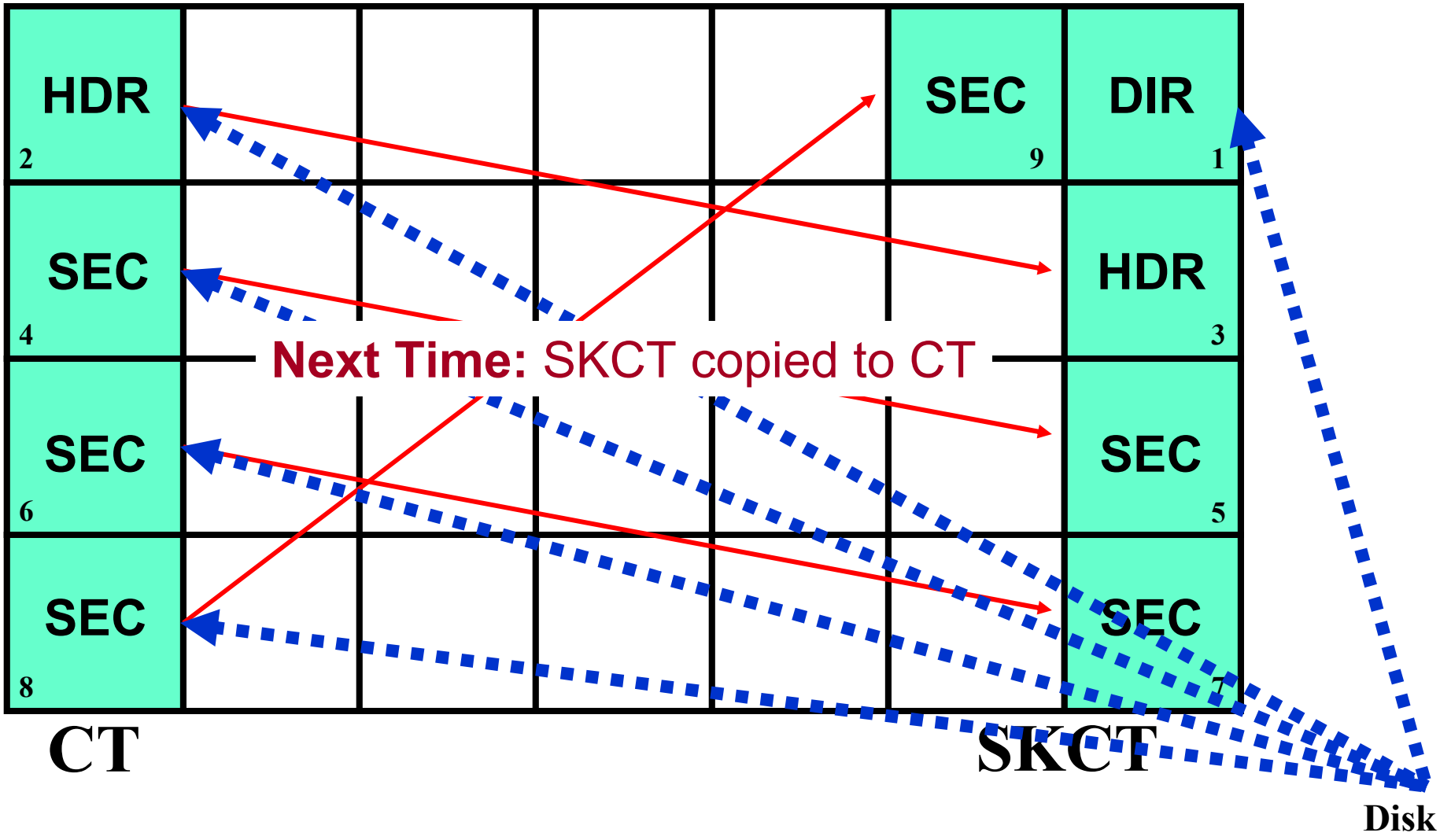
DB2 Database Services Address Space



What's in EDM Pool

- DBDs
- SKCTs
- CTs
- SKPTs
- PTs
- Auth Cache
- Dyn SQL Prep
- Free pages

First Time Plan Execution



The EDM Pool and Version 8

EDM Pool split into three specific pools:

- EDMPOOL: EDM Pool below 2GB Bar stores only CTs, PTs, SKCTs, SKPTs
  - **Should be able to reduce the size of this EDM pool**
  - **Provide some VSCR for below the 2GB Bar storage**
- EDM Pool above the 2GB Bar
  - **EDMDBDC**: DBDs
  - **EDMSTMTC**: Cached Dynamic Statements

General EDM Pool Tuning ROTs

If EDM Pool is too small:

- Fewer threads can run concurrently
- Increased response time due to loading of SKCT, SKPT, and DBD
- Increased I/O to SCT02, SPT01 and DBD01
- Repreparation on Caching Dynamic SQL

Watch out for DBD size:

- EDM Pool Size > 5x maximum DBD size

Over allocation is better than under allocation!

- If the pool is too large you are wasting memory, though

General Rule of Thumb:

Shoot for 80% or better DBD/CT/PT read efficiency

General SQL Tips

- Always provide ***only the exact columns*** that you need to retrieve in the SELECT-list of each SQL SELECT statement
- ***Use the WHERE clause to filter data*** in the SQL instead of bringing it all into your program to filter.
- In general, ***let SQL do the work***, not the program.
- ***Avoid writing SQL to access DB2 tables like flat files.*** Use joins instead of "master-file processing" techniques.
- ***Do not ask for what you already know*.**
  - See next slide for example.

A Simple Idea

- **Change this…**

```
SELECT  LAST_NAME, FIRST_NAME, JOB_CODE, DEPT
FROM    EMP
WHERE   JOB_CODE = 'A'
AND     DEPT = 'MIS';
```

- **To this…**

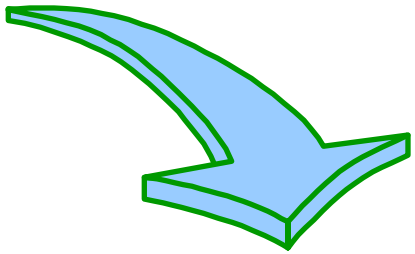
```
SELECT  LAST_NAME, FIRST_NAME, 'A', 'MIS'
FROM    EMP
WHERE   JOB_CODE = 'A'
AND     DEPT = 'MIS';
```

Wrangling Unruly Predicates

Try to change Stage 2 predicates to Stage 1 and non-indexable predicates to indexable predicates.

- **For example:**

```
SELECT COLA, COLB, COL6
FROM   T1
WHERE  COL1 NOT BETWEEN 'A' AND 'G';
```



```
SELECT COLA, COLB, COL6
FROM   T1
WHERE  COL1 >= 'H';    -or-   COL1 BETWEEN 'H' AND 'Z';
```

Note: NOT is non-indexable

Take Time to Understand NULLs

Every null column requires a one byte null indicator

Nulls do NOT save space – *ever*!

- **Nulls are not variable columns but…**
- ***Can be* used with variable columns to (perhaps) save space**

Do NOT bury your head in the sand & ignore nulls

- **You can code a query against a database w/o any nulls and receive null as the answer**

Nulls

INVEN_LOC_TAB

WAREHSE_NO SMALLINT	BIN_NO SMALLINT	PROD_NO CHAR(5)	PROD_QTY INT
1	1	A100	???
1	2	A150	2000
1	3	B167	30
2	1	A100	775
2	2	D400	1585

NULL:
Has no value
Is not = anything
Is not < anything
Is not > anything
Is not = NULL
Is an UNKNOWN value

ATTRIBUTE QUALIFIER	DESCRIPTION
DEFAULT NULL	When no value is provided, DB2 automatically assigns nulls to the Column
NOT NULL	The column must always contain a value, whether a default or explicity provided

01/31/94 DSGN0509

```
SELECT  SUM(SALARY)
FROM    EMP
WHERE   DEPTNO > 999;
```

RI: System or User-Managed?

System-Managed

- Standard declarative implementation.
- Less coding required.
- Easier to modify later. (DDL and CHECK)
- More efficient.
- Ad hoc and planned updates.

User-Managed

- Requires program code to be written.
- Hard to modify later.
- Sometimes there is the possibility for better insert performance.
- Works only for planned updates.

DB2 V8:
Informational Referential Constraints

Use DB2 Declarative Integrity Constraints

CUST_NO SMALLINT	CUST_NAME CHAR(30)	CUST_ADDR CHAR(25)	CUST_CITY CHAR(20)	CUST_ STATE CHAR(2)	CUST_ ZIP INT	CUST_ AREA_ CODE SMALLINT	CUST_ PHONE INT	CREDIT_ CAT CHAR(3)	SALES_ REP INT	COLLECT_ AGY_NO SMALLINT
1	ACME INC	222 ELM ST	LISLE	IL	60532	708	5551212	AAA	123456	
2	PXI CORP	5 PXI PLAZA	DALLAS	TX	76543	407	8326745	AAA	273818	5
3	SYNC CORP	2254 HAMILTON DR	FT.WASHINGTON	PA	19003	215	8983736	C	583490	
4	FR STANLEY INC	987 BEAVER DR	ANNAPOLIS	MD	30589	301	2734147	AA	123456	
5	BUYLO INC	235 IRON ST	BLOOMSBURG	PA	17815	717	9895280	BB	903757	
7	WHOS AIRCRAFT	8837 DESERT RD	TUCSON	AZ	80345	602	6743333	A	583490	
8	BUSINESS INFO CO	555 WATERSEDGE	LOMBARD	IL	60406	708	4836285	AAA	273818	2
9	CPU INC	1123 PEACH ST	COLUMBUS	OH	50387	216	4823778	C	999475	
10	XYZ CORP	4590 WAYNE RD	LIVONIA	MI	46827	313	3435555	B	999475	

FOREIGN KEY

ASSOCIATION INTEGRITY

Each city/state pair has a valid zip code

SALES_ORDR_TAB

SALES_ ORDR_ NO INT	ORDR_DATE DATE	CUST_NO SMALLINT	SALES_HIST_ CUST_NO SMALLINT	ORDR_AMT DEC(9,2)
1	1991-09-15		2	1923.45
3	1991-09-23		1	2407.53
4	1991-09-23		10	57613.89
5	1991-09-29	3	5	67000.00
6	1991-10-01	2		42345.88
7	1991-10-02	2		122345.61
8	1991-10-02	7		23007.34
9	1991-10-05	1		9823.55
10	1991-10-07	5		223019.27
11	1991-10-11			78780.99

DOMAIN INTEGRITY

Each value of CREDIT_CAT is valid

DEPENDENT TABLE

REFERENTIAL INTEGRITY

Each value of CUST_NO exists as a value of CUST_NO in CUST_TAB

Data Type and Length

It might seem like a simplistic tip, but choosing the correct data type and length for your data will greatly improve data integrity.

DB2 Data Types

- CHAR / VARCHAR
- CLOB
- DBCLOB
- GRAPHIC / VARGRAPHIC
- BLOB
- DATE
- TIME
- TIMESTAMP
- INTEGER
- SMALLINT
- DECIMAL
  - NUMERIC
- FLOAT
  - REAL
  - DOUBLE

Numeric vs. Character

Need: numeric data with leading zeroes

Character	Numeric (INT or DEC)
If input properly, leading zeroes always show.	Automatic edit checking for numeric data.
Requires rigorous edit checking for data entry.	Potential for more efficient access because filter factors are more accurate.
Not the “best” choice for the value domain.	Best choice for domain.

Handling Large Character Data

DATA TYPE	DESCRIPTION	LENGTH
VARCHAR(X)	Variable Length String	Max 4046
LONG VARCHAR (4K Page)	Variable Length String	Max 4046 Calculated by DB2
LONG VARCHAR (32K Page)	Variable Length String	Max 32704 Calculated by DB2

- **Actual data lengths should vary widely before VARCHAR should be considered.**
- **LONG VARCHAR prevents column additions.**
- **Consider CLOB for very large character columns.**
- **Breaking up a VARCHAR into two tables can help performance under the proper conditions.**
- **Consider using compression instead of VARCHAR!** →

Consider DB2 Compression

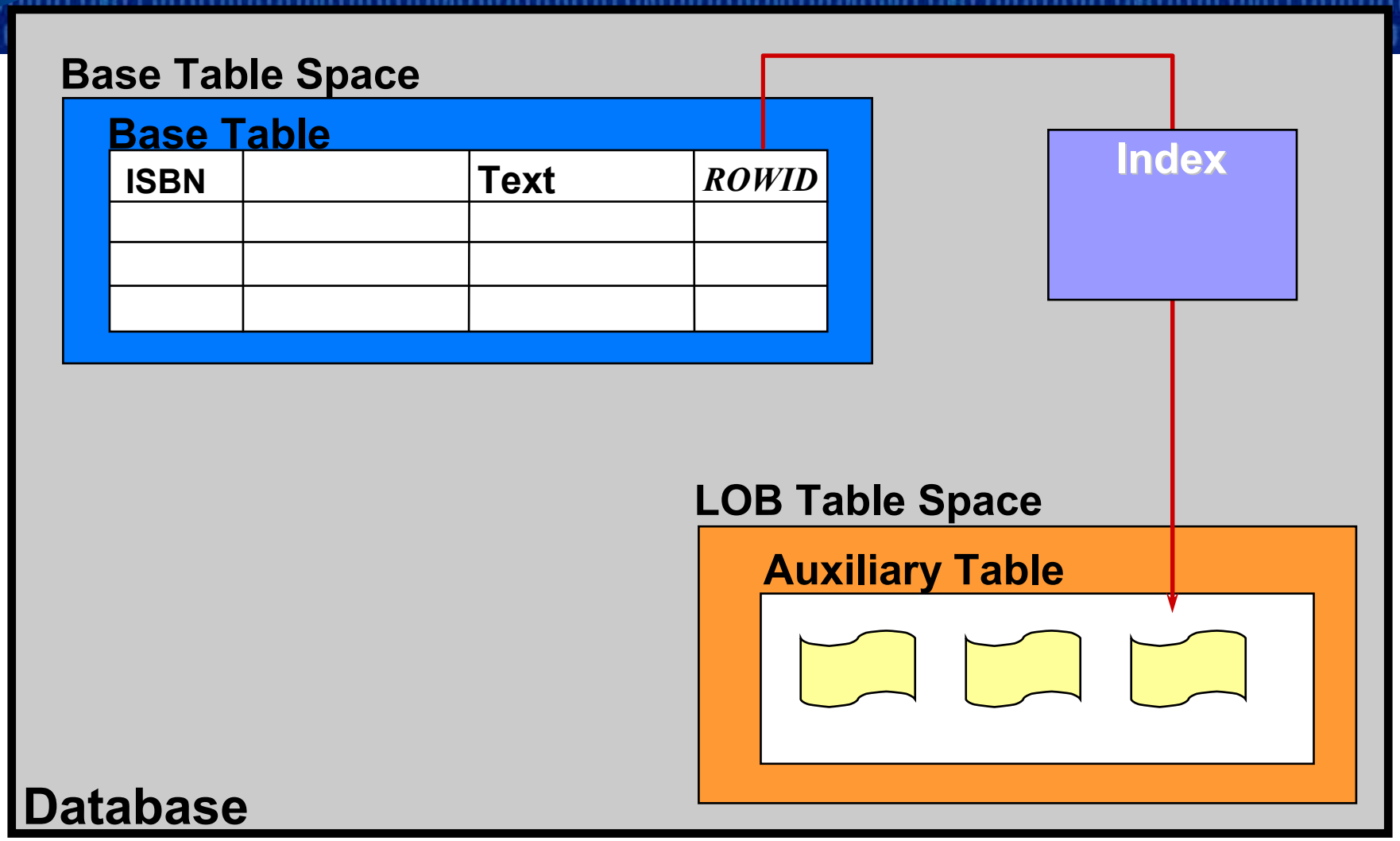
Consider compression instead of VARCHAR

- **Compression = less overhead (no 2 byte prefix)**
- **Compression requires no programmatic handling**
  - But it does add a compression dictionary to the table so a compressed small table may be larger than a non-compressed small table.
  - Also, be sure to weigh increased CPU to compress/de-compress against the decreased I/O due to smaller row sizes

Same basic impact, to minimize row length…

- **Compression will compress the entire row**
- **VARCHAR only shrinks the size of the column(s)**

Storing Large Objects



DATE/TIME vs. TIMESTAMP

Need: date and time for each row of a table

DATE / TIME

Requires 2 columns.

Saves storage: only 7 total bytes required.

Less precise: seconds.

DB2 provides more formatting options for DATE and TIME.

TIMESTAMP

Everything in 1 column.

Requires 10 bytes of storage.

More precise: microseconds.

DATE arithmetic easier using 1 column

DATE/TIME Arithmetic

DB2 can add and subtract DATE, TIME, and TIMESTAMP values and columns

- and DATE, TIME, and TIMESTAMP durations

Guidelines:

- **Let DB2 do the hard work for you.**
- **Use the proper DB2 data types.**
- **Understand durations *(see next slide).***
- **Know your DB2 functions *(see slides after next).***

Understanding Durations

Labeled Durations – **YEAR(S), MONTH(S), DAY(S), HOUR(S), MINUTE(S), SECOND(S), MICROSECOND(S)**

- **Example(s): 10 DAYS 2 YEARS 33 MINUTES 1 SECOND**

Date Durations –yyyymmdd **DECIMAL(8,0)**

- **EXAMPLE: 00201304** (20 years, 13 months, and 4 days)

Time Durations – hhmmsss **DECIMAL(6,0)**

- **EXAMPLE: 081144** (8 hours, 11 minutes, and 44 seconds)

TIMESTAMP Durations – yyyyxxddhhmmsszzzzzz

- **DECIMAL(20,6)**
- **EXAMPLE: 00201304081144.004351**
  (20 years, 13 months, 4 days, 8 hours, 11 minutes, 44 seconds, & 4351 microsceonds)

Using Durations in a Program

Q: How can I subtract a variable from the current date. I want to subtract a number of days from the date, but I keep getting an error: undefined or unusable variable. Here is the code:

```
SELECT A.DOC, A.TRAN_DATE, A.CRTE_DATE..
FROM ABC.TODDOC A,.....
WHERE A.CMPY = B.CMPY
AND ....
AND (A.CRTE_DATE <= CURRENT DATE - :SUB2 DAYS)
AND (A.CRTE_DATE > CURRENT DATE - :SUB3 DAYS)
AND .....
```

A: You **cannot** use a labeled duration with a host variable like that. The solution is to use a date duration. Remove the “DAYS” literals and change :SUB2 and :SUB3 to represent a DECIMAL(8,0) number, formatted as shown on the previous slide.

Know Your DATE/TIME Functions!

Q: How can I get DB2 to express a duration resulting from DATE subtraction as a total number-of-days? For example:

```
SELECT DATE('03/01/2004') - DATE('12/01/2003')
```

A: Use the DAYS function to return the exact number of days between those two dates, as follows:

```
SELECT DAYS('03/01/2004') - DAYS('12/01/2003')
```

Sort by Day of Week Order?

First thought is to try this:

```
SELECT DAY_NAME, COL1, COL2 . . .
FROM TXN_TABLE
ORDER BY DAY_NAME;
```

But…the results from this query would be ordered alphabetically; in other words:

FRI
MON
SAT
SUN
THU
TUE
WED

Sort by Day of Week Order

Instead, try this:

```
SELECT DAY_NAME, COL1, COL2 . . .
FROM TXN_TABLE
ORDER BY LOCATE(DAY_NAME,'SUNMONTUEWEDTHUFRISAT');
```

LOCATE finds the position of the DAY_NAME value within the specified string, and returns the integer value of that position. So, if DAY_NAME is WED, the LOCATE function returns 10. Sunday would return 1, Monday 4, Tuesday 7, Wednesday 10, Thursday 13, Friday 16, and Saturday 19. This means that our results would be in the order we require.

(Note: Some other database systems have a function similar to LOCATE called INSTR.)

DATE/TIME Functions

CHAR
DATE
DAY
DAYOFMONTH
DAYOFWEEK
DAYOFYEAR
DAYS
HOUR
JULIAN_DAY
LAST_DAY
MICROSECOND
MIDNIGHT_SECONDS
MINUTE
MONTH
NEXT_DAY
QUARTER
SECOND
TIME
TIMESTAMP
WEEK
WEEK_ISO
YEAR

DATE/TIME Functions

(repeated for notes)

CHAR
DATE
DAY
DAYOFMONTH
DAYOFWEEK
DAYOFYEAR
DAYS
HOUR
JULIAN_DAY
LAST_DAY
MICROSECOND
MIDNIGHT_SECONDS
MINUTE
MONTH
NEXT_DAY
QUARTER
SECOND
TIME
TIMESTAMP
WEEK
WEEK_ISO
YEAR

Know Your DB2 Functions!

Not just for dates and times…

There are a LOT of DB2 functions these days – many times these functions can be used to cut down on a LOT of coding & additional work.

Trigger Tips and Thoughts

SQL termination character

- **DSNTEP2: `SET TERMINATOR`**
- **Command Center: 'Tools' → 'Tools Settings' → 'Use statement termination character'**
- **SPUFI Defaults: Option #1 SQL Termination**

Consider trigger "testing" columns

- **TIMESTAMP**
- **Last Trigger Name**

REBIND TRIGGER PACKAGE

- **This is the only way to EXPLAIN the SQL in triggers**

Avoid Base Table Views

Do not implement one view per base table. There are no benefits to doing so, but more work if/when database changes are required.

Instead, just allow programs access to the base tables.

View Usage Guidelines

- There are five basic uses for which views excel…
  1. to provide row and column level security
  2. to ensure efficient access paths
  3. to mask complexity from the user
  4. to ensure proper data derivation
  5. to rename columns (and/or tables), and

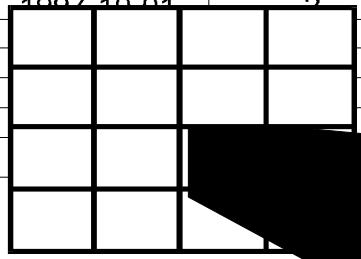
- Also, be sure to synchronize all views with base tables as they change.

Cluster on Appropriate Columns

NON-UNIQUE CLUSTERING INDEX ORDR_DATE

SALES_ORDER_TAB

SALES_ORDER_NO INT	ORDR_DATE DATE	CUST_NO SMALL INT	SALES_HIST_CUST_NO SMALL INT	ORDR_AMT DEC(9,2)
1	1997-09-15		2	1923.45
3	1997-09-23		1	2407.53
4	1997-09-23		10	57613.89
5	1997-09-29	3	5	67000.00
6	1997-10-01	3		42345.88
7				122345.61
8				23007.34
9				9823.55
10				223019.27
11				78780.99



ORDR_NO	DATE
000000125	19970923
000004946	19970923
000000013	19971002
000000615	19971002
000008885	19971002
000074155	19971004

CLUSTERING...

GIVES I/O

MORE

"BANG FOR THE BUCK!!"

Sequences or Identity Columns?

Identity Columns	Sequence Objects
Internal objects generated and maintained by DB2	Stand-alone objects created by the DBA
Associated with a single table	Not associated with any table
IDENTITY_VAL_LOCAL() to get last value assigned	PREVIOUS VALUE expr to get last value assigned
N/A – DB2 handles assigning next value	NEXT VALUE expression gets next value to be assigned
Add/change using ALTER TABLE *(V8+ only)*	Administer using ALTER SEQUENCE, DROP, GRANT, REVOKE, COMMENT.
Available as of V6 refresh	Not available until V8

Top Ten New DB2 V8 Features

1. 2M SQL Limit
2. Partitioning Changes 
3. Stage 1 for Unlike Data Types
4. Sequences
5. Distribution Statistics (DSTATS)
6. Multi-Row FETCH and INSERT
7. Materialized Query Tables
8. Dynamic Scrollable Cursors
9. Recursive SQL
10. Online Schema Evolution

Version 8 – Major Changes

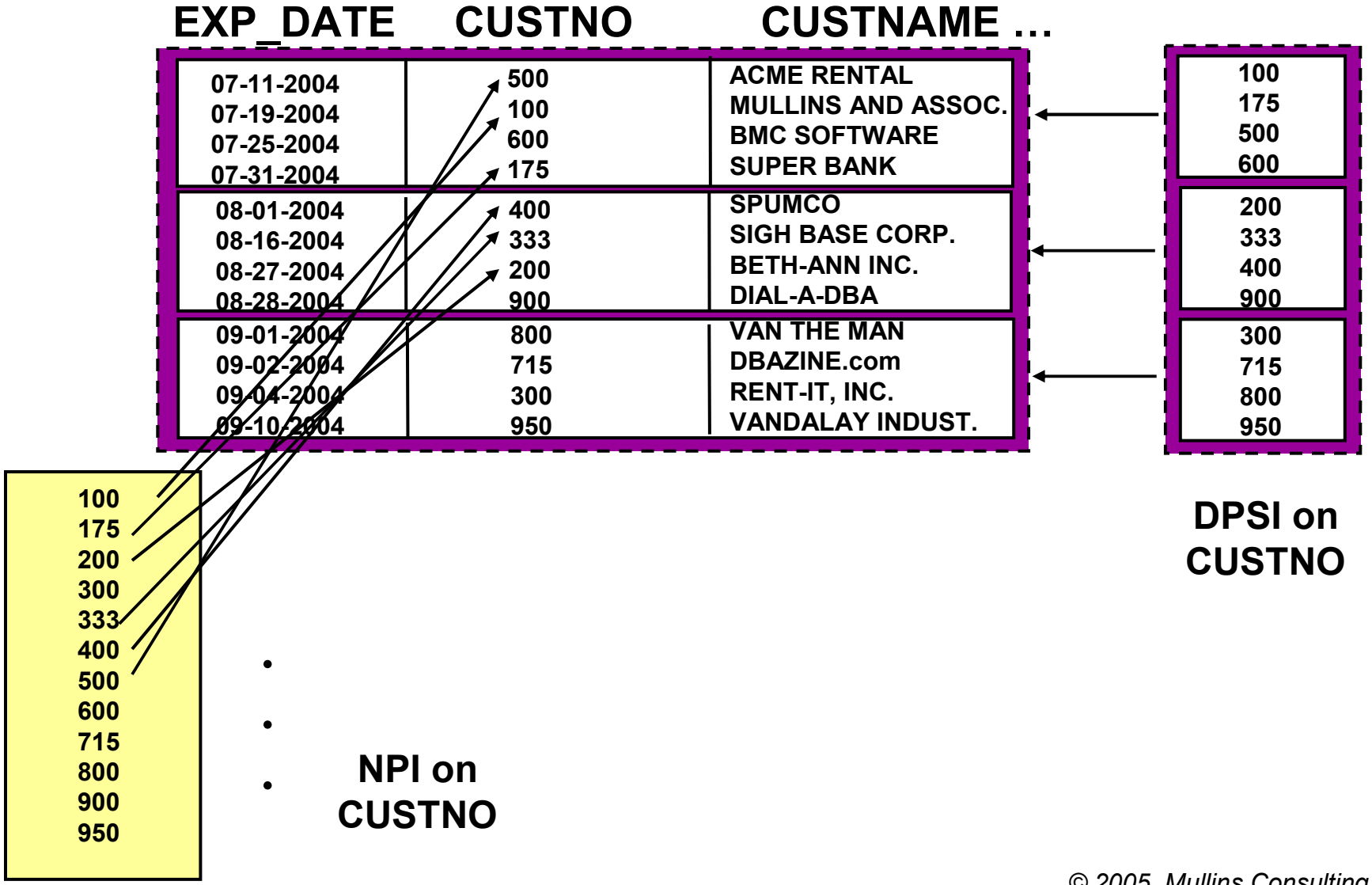
Partitioning and indexing are separated

- **Partitioned versus non-partitioned**
  - **Partitioned** = the index is physically partitioned into separate data sets;
  - **Non-partitioned** = the index is in one data set
  - Whether partitioned or not, the index may still be "partitioning"
- **Partitioning versus secondary**
  - **Partitioning** = the index aligns with the keys by which the data is partitioned
  - **Secondary** = index keys do not align with partitioning

Partitioning and clustering are separated

Can specify up to 4096 partitions

NPI versus DPSI



Real-Time Statistics (RTS)

With RTS, DB2 collects (some) statistics & periodically writes the stats to two user-defined tables.

Database is DSNRTSDB – two tables

- SYSIBM.TABLESPACESTATS
- SYSIBM.INDEXSPACESTATS
  - **Look in DSN710.SDSNSAMP(DSNTESS) for the information needed to create the RTS objects**

A supplied stored procedure, DSNACCOR, uses canned formulas to make recommendation on maintenance required…

Or you can "roll your own" queries

APARs PQ48447, PQ48448, PQ46859, and PQ56256

Index Reorg/Rebuild

Consider reorganizing your indexes more frequently instead of *over*-reorganizing your table spaces.

Added index levels? - NLEVELS > original NLEVELS

REORNUMLEVELS *(real time stats)*

LEAFDIST > 200

LEAFNEAR, LEAFFAR much > original values

More than 10% of leaf pages are far gaps:

((LEAFFAR * 100) / NLEAF) > 10

What Version of DB2 is Running?

To determine the version of DB2 for z/OS you can use the DB2 command: -DISPLAY GROUP

- This command can be issued from an MVS console, a DSN session under TSO, DB2I Commands, an IMS or CICS terminal, or any program using IFI (Instrumentation Facility Interface).

With data sharing, the output will include details about the data sharing group and each of its members. Each member will list DB2 level.

- Level is another name for DB2 Version. So, for Version 6, the DB2 level would be listed as 610.

For non-data sharing environments, of course, the output will not include any data sharing information.

-DIS GROUP Example

V8 will have a **session variable** for version

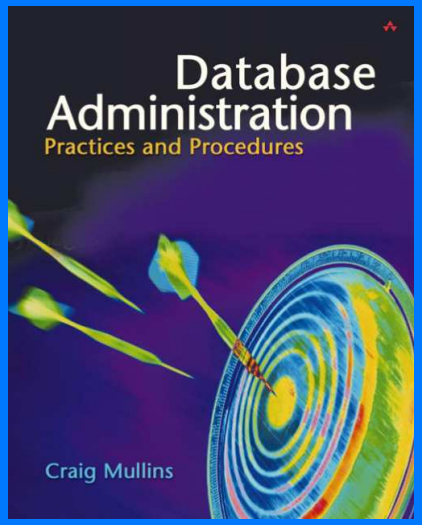
```
DSN7100I -DB1A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT ) GROUPLEVEL(610)
-------------------------------------------------------------------------
DB2                                          SYSTEM      IRLM
MEMBER   ID  SUBSYS CMDPREF   STATUS   NAME      LVL  SUBSYS IRLMPROC
-------- --- ---- -------- -------- -------- --- ---- --------
DB1A       1 DB1A -DB1A     ACTIVE   MVSA      610  IR21 ARLM21
DB1B       2 DB1B -DB1B     ACTIVE   MVSB      610  IR21 BRLM21
DB1C       3 DB1C -DB1C     ACTIVE   MVSC      510  IRLM CRLM21
DB2D       4 DB2D -DB2D     FAILED   MVSD      610  IR21 DRLM21
DB2E       5 DB2E -DB2E     QUIESCED MVSE      610  IR21 ERLM21
DB2F       6 DB2F -DB2F     ACTIVE   MVSF      610  IR21 FRLM21
DB2G       7 DB2G -DB2G     ACTIVE   MVSG      610  IR21 GRLM21
-------------------------------------------------------------------------
DB2      PARALLEL    PARALLEL
MEMBER   COORDINATOR ASSISTANT
-------- ---------- --------
DB2A            YES       NO
DB2B            YES      YES
DB2D            YES      YES
DB1C           ****     ****
DB2D           ****     ****
DB2E           ****     ****
DB2F              NO      YES
DB2G              NO       NO
-------------------------------------------------------------------------
SCA   STRUCTURE SIZE:     1024 KB, STATUS= AC,   SCA IN USE:    11 %
LOCK1 STRUCTURE SIZE:     1536 KB,             LOCK1 IN USE: <   1 %
NUMBER  LOCK ENTRIES:      262144, LOCK ENTRIES  IN USE:          33
NUMBER  LIST ENTRIES:        7353, LIST ENTRIES  IN USE:           0
*** END DISPLAY OF GROUP(DSNCAT )
DSN9022I -DB1A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

From DB2 Command Reference (SC26-9934)

Available Now

DB2 Developer's Guide, 5th *ed*

www.craigsmullins.com/cm-book.htm



DBA: Practices & Procedures

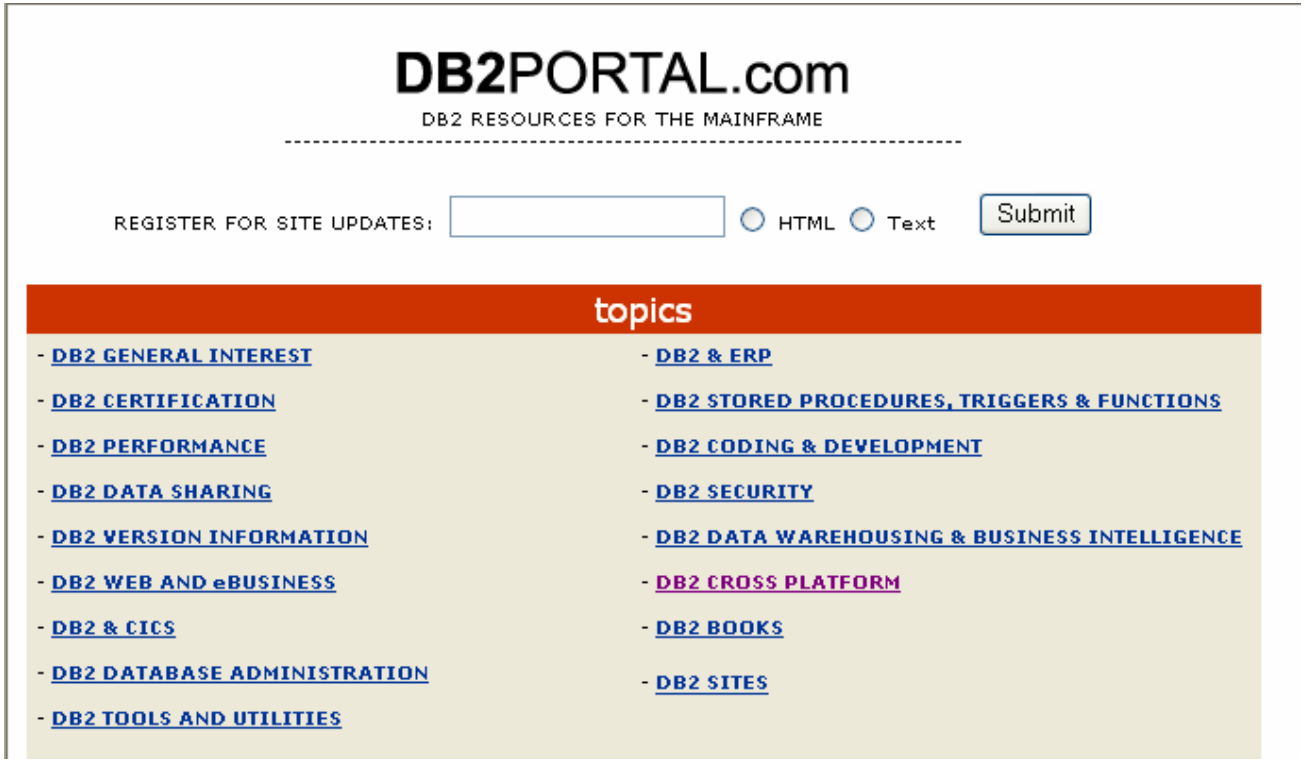
www.craigsmullins.com/dba_book.htm

Announcing DB2PORTAL.com

New, free web portal for DB2 DBA, programmers, & users

- Useful articles and links to DB2 information
- The only site devoted completely to mainframe DB2

http://www.DB2portal.com



An Hour of DB2 Tips

Craig S. Mullins

craig@craigsmullins.com

http://www.CraigSMullins.com