Ten Breakthroughs That Changed DB2 Forever!

Sponsored by:

http://www.compuware.com

Craig S. Mullins
Mullins Consulting, Inc.
http://www.craigsmullins.com
http://www.mullinsconsultinginc.com

© 2013 Mullins Consulting, Inc.
This presentation was prepared by:

Craig S. Mullins  
President & Principal Consultant

Mullins Consulting, Inc.  
15 Coventry Ct  
Sugar Land, TX  77479  
Tel: 281-494-6153  
Fax: 281.491.0637  
E-mail: craig@craigsmullins.com

This document is protected under the copyright laws of the United States and other countries as an unpublished work. Any use or disclosure in whole or in part of this information without the express written permission of Mullins Consulting, Inc. is prohibited.  
© 2013 Craig S. Mullins, Mullins Consulting, Inc. All rights reserved.
Objectives

Gain an historical perspective of the features and functionality of DB2.

We will count down and explain the ten most important technological breakthroughs made to DB2 for z/OS over the years since the introduction of V1 in the early 1980s.

› Each breakthrough will be covered in chronological order and introduced by the version of DB2 where it was delivered.
› Each feature, and its significance will be explained along with examples and what was done prior to its introduction.
› Coverage of several breakthroughs that did not make the top ten will be offered, along with brief explanations of why not, and how things could change based on adoption and usage.

A discussion of what may come next in terms of future DB2 breakthroughs will be offered at the end.
# The History of DB2 for z/OS

<table>
<thead>
<tr>
<th>Version</th>
<th>GA</th>
<th>EoM</th>
<th>EoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>1985-04-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>1986-03-07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>1987-06-26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>1988-09-23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>1989-09-22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>1991-10-25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1993-12-17</td>
<td>1999-11-30</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1995-10-30</td>
<td>2000-12-01</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1997-06-27</td>
<td>2001-12-31</td>
<td>2002-12-31</td>
</tr>
<tr>
<td>7</td>
<td>2001-03-30</td>
<td>2007-03-05</td>
<td>2008-03-30</td>
</tr>
<tr>
<td>8</td>
<td>2004-03-26</td>
<td>2009-09-08</td>
<td>2012-04-30</td>
</tr>
<tr>
<td>9</td>
<td>2007-03-06</td>
<td>2012-12-10</td>
<td>2014-06-27</td>
</tr>
<tr>
<td>10</td>
<td>2010-10-22</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Version vs. Release

- IBM has not delivered a “release” since 1991 with DB2 Version 2 Release 3.

- What is the difference between a version and a release?
  - A new version of software is a major concern, with many changes and new features.

  - A release is typically minor, with fewer changes and not as many new features.
In the early days of DB2...

- Throughout the Version 1 releases of DB2 it was considered an “information center” product only.
So, What Features Contributed to DB2’s Status Today as...

- A high-performance, feature-laden transaction processing DBMS that also...
- Offers robust data warehousing and analytical capabilities while...
- Providing high availability with...
- On demand administrative and management capabilities?

Ten Breakthroughs That Changed DB2 Forever
DATE and TIME Data Types
DB2 V1.3

Prior to DB2 V1.3, there was no DATE and TIME data type support.

Most designs used the CHAR data type for date/time data.

Drawbacks:

- No validation of the data
- No ability to perform date/time arithmetic
- In general, bad data quality
DB2 can add and subtract DATE, TIME, and TIMESTAMP values and columns

› and DATE, TIME, and TIMESTAMP

Guidelines:

› Let DB2 do the hard work for you.
› Use the proper DB2 data types.
› Understand durations (*see next slide*).
› Know your DB2 functions.
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: 00201104 (20 years, 11 months, and 4 days)

Time Durations - hhmmssss DECIMAL(6,0)
› Example: 081144 (8 hours, 11 minutes, and 44 seconds)

TI MESTAMP Durations - yyyyxxddhhmmsszzzzzzzz
› DECIMAL(20,6)
› Example: 00201104081144.002351 (20 years, 11 months, 4 days, 8 hours, 11 minutes, 44 seconds, & 2351 microseconds)
# DATE / TIME vs. TIMESTAMP

## DATE / TIME
- Requires 2 columns.
- Saves storage: only 7 total bytes required.
- Less precise: seconds.
- DB2 provides formatting options for DATE and TIME (not TS).

## TIMESTAMP
- Everything in 1 column.
- Requires 10 bytes of storage.
- More precise: microseconds.
- DATE arithmetic easier using 1 column.
DB2 9 and 10: Adding Timestamp Functionality

**DB2 9: New Timestamp Functions**

- **TIMESTAMPADD** - adds an interval to a timestamp.
- **TIMESTAMPDIFF** - subtracts two timestamps & returns an interval.
- **TIMESTAMP_FORMAT** - changes the display format for a timestamp value. Valid formats that can be specified are:
  - ‘YYYY-MM-DD’
  - ‘YYYY-MM-DD-HH24-MI-SS’
  - ‘YYYY-MM-DD-HH24-MI-SS-NNNNNN’

**DB2 10: More Flexible Timestamps**

- Can now specify a precision to indicate the amount of fractional seconds (default is 6, compatible with old TIMESTAMPS).
- Can optionally specify **TIMESTAMP WITH TIME ZONE** to add time zone information to your data.
The addition of RI in DB2 V2.1 was a boon to data integrity

<table>
<thead>
<tr>
<th>CUST_NO</th>
<th>CUST_NAME</th>
<th>CUST_ADDR</th>
<th>CUST_CITY</th>
<th>CUST_STATE</th>
<th>CUST_ZIP</th>
<th>CUST_AREA_CODE</th>
<th>CUST_PHONE</th>
<th>CREDIT_CAT</th>
<th>SALES_REP</th>
<th>COLLECT_AGENCY_NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACME INC</td>
<td>222 ELM ST</td>
<td>Lisle</td>
<td>IL</td>
<td>60532</td>
<td>708</td>
<td>5551212</td>
<td>AAA</td>
<td>123456</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PXI CORP</td>
<td>5 PXI PLAZA</td>
<td>Dallas</td>
<td>TX</td>
<td>78643</td>
<td>407</td>
<td>8926745</td>
<td>AAA</td>
<td>234567</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SYNG CORP</td>
<td>2254 HAMILTON DR</td>
<td>PA</td>
<td>PA</td>
<td>19003</td>
<td>214</td>
<td>898637286</td>
<td>C</td>
<td>567890</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>FR STANLEY INC</td>
<td>387 BEAVER DR</td>
<td>ANNA/OLIS</td>
<td>MO</td>
<td>30589</td>
<td>301</td>
<td>234567890</td>
<td>AA</td>
<td>123456</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>BUYLO INC</td>
<td>236 IRON ST</td>
<td>Bloomberg</td>
<td>PA</td>
<td>17815</td>
<td>717</td>
<td>365528007</td>
<td>BB</td>
<td>900012</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>WHOA AIRCRAFT</td>
<td>8837 DESERT RD</td>
<td>TUCSON</td>
<td>AZ</td>
<td>80346</td>
<td>602</td>
<td>67433333</td>
<td>A</td>
<td>567890</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BUSINESS INFO CO</td>
<td>555 WATERIDGE</td>
<td>Lombard</td>
<td>IL</td>
<td>60406</td>
<td>708</td>
<td>493628564</td>
<td>AAA</td>
<td>234567</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CPU INC</td>
<td>1123 PEAK ST</td>
<td>COLUMBUS</td>
<td>OH</td>
<td>45367</td>
<td>216</td>
<td>46423778</td>
<td>C</td>
<td>999475</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>XYZ CORP</td>
<td>4590 WAYNE RD</td>
<td>Livonia</td>
<td>MI</td>
<td>48637</td>
<td>313</td>
<td>3435555555</td>
<td>B</td>
<td>999475</td>
<td></td>
</tr>
</tbody>
</table>

**FOREIGN KEY**
Each city/state pair has a valid zip code

**ASSOCIATION INTEGRITY**

**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
RI: System or User-Managed?

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

- 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

© 2013 Mullins Consulting, Inc.
Some newer DB2 programmers may not realize that packages were not always part of DB2

- Packages simplify application development.

What is package?

- A single, bound program with optimized access paths.
- Prior to packages, programs were bound at the plan level.
- A plan could consist of multiple DBRMs.
- But what if you only needed to rebind one program?
- The only way to do it without packages was to rebind the entire plan (including every program).
  - If a plan consists of hundreds (or even thousands) of programs, the rebind can take a long time. And that translates into downtime for the applications using the plan.
Program Preparation

The DB2 precompiler produces a DBRM and a modified source program with commented SQL statements.

The DB2 BIND command turns a DBRM into an executable format.

The DB2 Catalog stores information about the plan and package.

DB2 Directory stores the actual plan and package.

Source Program

DCLGEN

Precompiler

DBRM

Bind Package

Bind Plan

Plan

DB2 Catalog

DB2 Directory

Modified Source

Compiler

Compiled Source

Linkage Editor

Load Module
Package Benefits

Simplified program management

› BIND and REBIND can be done at the program level instead of the plan level

Granularity of BIND parameters

› For example, multiple packages, each with different ISOLATION or RELEASE specifications, can be bound to single plan

Versioning

› Multiple versions of a package can exist
Data Sharing is perhaps the single biggest improvement in DB2’s three decade life span

- Runs in a z/OS Parallel Sysplex
- Allows concurrent read/write data access from multiple DB2s
- Data must reside on shared devices
  - Single DB2 Catalog/Directory
- Application can run on any member DB2
The Importance of Data Sharing

The Critical Mass

➢ Prior to Data Sharing, everything that required access to DB2 had to reside on the same CPU
  ▪ DB2, TSO, IMS/TM, CICS…

➢ When errors occurred you could not move subsystems independently to another LPAR without losing access to DB2

➢ A large shop could quickly exhaust machine resources this way
Benefits of Data Sharing

No special programming required
Improved data availability
Extended processing capacity
Flexible system configuration
Improved transaction rates
Version 4 was a very significant release because it also brought Type 2 indexes.

The old indexes (Type 1) required locking, whereas Type 2 indexes do not.

- With Type 2 indexes, a lock on the data page or row acts as a lock on the index key.

Index locking was one of the most significant barriers to DB2 performance.
Type 1 vs. Type 2 Index

Type 1 indexes supported subpages

› A way to break up a 4K page into smaller units
› Because index rows are smaller than table rows locking a full 4K index page could cause significant bottlenecks
  - 1, 2, 4, 8, or 16 subpages

Type 2 indexes do not require subpages

› No locking
DB2 was late to the game with stored procedure support

- But starting with V4, and with significant improvements over subsequent releases DB2 offers stellar stored procedure support today

A stored procedure is procedural, business logic that is managed in the DBMS and is executed by name.
Stored Procedure Benefits

Reduce network traffic for distributed applications

Consistent behavior

＞ When any application calls the stored procedure, it processes data in a consistent way

Ease of maintenance

＞ If you need to change the rules, you only need to make the change once in the stored procedure, not in every application that calls the stored procedure.

Flexible development choices

＞ Can be coded in C, C++, COBOL, Assembler, PL/I, REXX, SQL Procedures language and Java.

Can access non-DB2 resources

＞ VSAM files, flat files, IMS or CICS transactions, DL/I databases, MVS/APPC conversations
Stored Procedures - Through The Years

Originally coded using traditional programming language only and executed via SPAS

› WLM replaced SPAS in V5

Could not return result sets until V5

Originally had to be manually registered to DB2 Catalog

› CREATE PROCEDURE added in V6

SQL stored procedures - supported as of V6

› But SQL procedures were converted to C programs

› Native SQL stored procedures
  
  ▪ Supported as of DB2 9 for z/OS
  
  ▪ No external load module is created
  
  ▪ Run within the DB2 engine (DBM1 address space)

    No WLM/SPAS required

  ▪ zIIP eligible when called via distributed connection
Stored Procedure Usage Guidelines

- Keep it simple - make each procedure do one thing only
- Use stored procedures to create reusable “components” of business logic
- Putting multiple types of business logic into a single stored procedure makes it more difficult to tune, modify, and understand
- Document the purpose of each procedure as well as any required input, outputs, and each change made to the stored procedure
Execution is automatic and implicit
Triggers implement “active” database systems
Trigger is executed, or “fired,” based upon a pre-defined “firing” activity:
- Database Modification (INSERT, UPDATE, DELETE)

Trigger uses include:
- Perform subsequent modification
- Implement business rules
- Maintain redundant data
- Maintain derived data
- Validate data

Like with stored procedures, DB2 was late to the game with trigger support, too
Trigger Impact on DBAs

- Performance Implications
  - Monitoring and Tuning
  - Statement vs. “Row-at-a-time” Behavior
    - Statement triggers fire even if no rows are impacted.
    - Not so with ROW triggers; only if rows are impacted.

- Cascading triggers
  - Can change a lot of data with one SQL modification

- Trigger execution order (when multiple triggers exist)
  - Of the same type on the same table...
  - Oldest to newest - be careful

- Trigger Access Paths
  - Must REBIND TRIGGER PACKAGE ... EXPLAIN YES
Additional Trigger Impacts and Issues

Utilities do not activate triggers

▶ **LOAD** can be troublesome

Trigger Dependencies:

▶ If you **DROP** any object used by the trigger the trigger package will become invalid, and the trigger will not run

When data already exists?

▶ If you add a trigger to a table that already contains data, the trigger will not be fired for the existing data (data integrity problems may exist unless they are addressed by other means)

SQL termination character

▶ **DSNTEP2**: SET TERMINATOR

▶ **SPUF1** Defaults
Procedural DBA Requirements
Role of the Procedural DBA

DBCO Implementation
(COMMIT in proc, write or guide)

DBCO Administration
(trigger firing order, proc set)

On Call for DBCO Abends

Ensuring Reuse

Design Reviews

Coding Complex Queries

EXPLAIN Analysis

Tuning SQL

Debugging SQL

Schema Resolution

DBCO = Database Code Object
shorthand for trigger, UDF, stored procedure

© 2013 Mullins Consulting, Inc.
DB2 V9 added a new type of trigger: INSTEAD OF triggers

- INSTEAD OF triggers can only be defined on VIEWS.
- INSTEAD OF triggers enable views that would not otherwise be modifiable to support INSERTs, UPDATEs, and DELETEs.
  - Typically, a view that consists of multiple base tables cannot be updated.
- With an INSTEAD OF trigger you can code logic to direct inserts, updates and deletes to the appropriate underlying tables that comprise the view.
  - Each requested modification made against the view is replaced by the trigger logic. The trigger performs the insert, update, or delete on behalf of the view.

In the earliest days of DB2, the “best practice” advice was to lump *everything* into BP0 and “let DB2 manage it.” This might have been reasonable in the days when only 5 buffer pools were available:

- BP0, BP1, BP2, BP3, and BP32K
Use Multiple Buffer Pools

IBM provides 80 buffer pools for a reason
- 4K: BP0 thru BP49
- 8K: BP8K0 thru BP8K9
- 16K: BP16K0 thru BP16K9
- 32K: BP32K thru BP32K9

Using them effectively optimizes performance

Ideas:
› isolate the catalog in BP0
› separate indexes from table spaces
› isolate heavily hit data
› isolate sort work area
› optimize BP strategy for your data & app processing mix: sequential vs. random
› there is no “silver bullet” approach
DB2 10 Buffer Pool Enhancements

Taking advantage of System z and z/OS improvements

Improved memory allocation

> Memory is allocated as data is brought into the buffer pool instead of at DB2 startup

New System z10 1 Megabyte page size

> 1MB page size enables DB2 to manage larger buffer pools better
The number 9 DB2 breakthrough is a bit more nebulous than the previous 8 we’ve walked through, but it is just as important, if not moreso.

**DB2 V8 was re-architected to support 64 bit**

- Required new hardware, new OS, and Unicode (in DB2 Catalog)
- With 64-bit addressing DB2 can theoretically support up to 16 exabytes
  - $1 \text{ EB} = 1,000,000,000,000,000,000 \text{B} = 10^{18} \text{ bytes} = 1,000,000,000 \text{ gigabytes} = 1,000,000 \text{ terabytes}$

**Other architectural improvements in DB2 V8 included:**

- Long names (up to 128 bytes)
  - Columns 30 bytes, table spaces 8 bytes, packages 8 bytes (except trigger packages 128 bytes).
- DB2 Catalog table spaces with larger page sizes (>4K)
The EDM Pool and V8, V9

V8: EDM Pool split into three specific pools:

- **Below the 2GB Bar**
  - **EDMPOOL**: EDM Pool stores only CTs, PTs, SKCTs, SKPTs
    - Should be able to reduce the size of this EDM pool
    - Provides some VSCR for below the 2GB Bar storage

- **Above the 2GB Bar**
  - **EDMDBDC**: DBDs
  - **EDMSTMTC**: Cached Dynamic Statements

V9: Introduces additional changes

- **Above the 2GB Bar**: **EDM_SKELETON_POOL**
  - All SKCTs and SKPTs
- A portion of the CT and PT is moved above the bar, too
Of Mainframe Storage, Bars & Lines

- 64 bits - 16Eb
- 49 bits - 512Tb
- 32 bits - 4Gb
- 31 bits - 2Gb
- 24 bits - 16Mb

User Area
Above the Bar
Reserved For System
Wasted
Above the Line
Below the Line

The Bar
MVS/XA (1981)

The Line

z/OS 1.2 64 bit addressing (2001)

Not to scale
Modifying database structures after they have been created had always been somewhat troubling before online schema evolution in DB2 V8

- Many types of common changes were not supported by ALTER
- Required dropping and re-creating the object
  - As well as all related objects, security, etc. that cascaded DROP
- Resulted in significant outages to make database changes

Renamed Database Definition on Demand in DB2 V9

- Each new version of DB2 adds more DDOD capabilities
Why This is a Breakthrough: An Example

Steps to change CHAR(10) column to CHAR(15)

1. Unload data and extract DDL and authorizations
2. Drop the table
3. Modify the DDL to change the column to CHAR(15)
4. Re-create the table
5. Re-create all dependent objects
6. Rebuild authorization for the table
7. Rebuild authorizations for all dependent objects
8. Reload the data (using the new 15 byte column)
9. Run utilities (RUNSTATS and COPY)
10. REBIND all impacted programs
11. Test it all to make sure it worked
Online Schema Evolution
DB2 V8

**ALTER** can extend the length of a CHAR column to a greater size

Can change data type within family
- CHAR → CHAR, numeric → numeric, date/time → date/time

Can add a column to an index and change padding

Can change the clustering index

Partitioning changes
- Add partitions
- Rotate partitions
- Change partition boundaries
- Rebalance partitions
Database Definition on Demand
Post V8

IBM continues to advance the ability to make changes to DB2 databases without an outage

› DB2 V9
- Elimination of REORG BUILD2 phase
- Cloning tables
- Rename a column within a table and rename indexes

› DB2 V10
- ALTER a table space’s page size, data set size, segment size, and/or MEMBER CLUSTER structure
- ALTER a simple or segmented table space with only one table to a partition-by-growth Universal table space
- ALTER a classic partitioned table space to a partition-by-growth or range-partitioned Universal table space
- ALTER an index’s page size
Other Important Breakthroughs?

Temporal - DB2 10
pureXML - DB2 9
Real Time Statistics - DB2 V7
LOB support - DB2 V6

Parallelism
  › I/O parallelism V3
  › CP parallelism V4
  › Sysplex parallelism V5

DB2 LUW (UDB)
  › OS/2 EE DBM
What Lies Ahead?

More changes, more availability, more usability
But what will DB2 V30 look like in the year 2041?*

* There is no guarantee that Version 30 will be available before the end of 2041 and mention of it in this presentation should not indicate an actual future version or release of DB2
Both of my books were updated in 2012...

6th edition available now... covering up thru DB2 V10!

Craig S. Mullins
Mullins Consulting, Inc.
15 Coventry Court
Sugar Land, TX 77479

http://www.craigsmullins.com

2nd edition available now...