It’s Not Your Daddy’s Db2!

Challenges of Db2 in the Digital Age

Craig S. Mullins
Mullins Consulting, Inc.

Session Code: B13
Wed, May 25, 2016 (03:30 PM - 04:30 PM) : Trinity B | Platform: Db2 for z/OS
This presentation was prepared by:

Craig S. Mullins
President & Principal Consultant
IBM Champion for Information Management
IBM Gold Consultant

Mullins Consulting, Inc.
15 Coventry Ct
Sugar Land, TX 77479
Tel: 281-494-6153

Skype: cs.mullins
E-mail: craig@craigsmullins.com
http://www.mullinsconsulting.com

Craig was named one of the Top 200 Thought Leaders in BigData & Analytics by AnalyticsWeek.

This document is protected under the copyright laws of the United States and other countries as an unpublished work. This document contains information that is proprietary and confidential to Mullins Consulting, Inc., which shall not be disclosed outside or duplicated, used, or disclosed in whole or in part for any purpose other than as approved by Mullins Consulting, Inc. Any use or disclosure in whole or in part of this information without the express written permission of Mullins Consulting, Inc. is prohibited.

© 2016 Craig S. Mullins and Mullins Consulting, Inc. (Unpublished). All rights reserved.
Agenda

• Industry & DBA Trends
  • Data Growth - Big Data & Analytics
  • IoT, Mobile, Distributed
  • Less downtime, fewer DBAs, more work
  • Autonomics

• Db2 Modernization
  • Database Design
    • UTS, unstructured data/LOBs, hashing, HTAP
  • Security
    • SECADM, Roles/TRUSTED CONTEXT, Improved audit, MLS/SECLABEL, Row Permissions, Column masks

• Development trends
  • Dynamic SQL, IDEs, web
  • New SQL: time travel, OLAP, recursion

• Some Guidance
Part 1

INDUSTRY & DBA TRENDS
Database Market Continues to Grow

Worldwide Total Data Revenue by Segment ($M) 2014-2019

Source: 451 Research Market Monitor

https://451research.com/market-monitor-forecast-overview
451 Research predicts Total Data market to reach $138.5bn by 2021

MAY 23 2017
BY MATT ASLETT, GREG ZWAKMAN

The Total Data market, consisting of data platforms and analytics, will grow at a compound annual growth rate of 11.5% from the end of 2016 to 2021, according to the latest market-sizing and -forecasting update from 451 Research.
The Nexus of Forces: 
*Driving New Transactions*

- Social
- Mobile
- Cloud
- Information

**Interactions**
- Unstructured, ACID not required
- Emails, "likes," user content, tweets, weblogs
- **Typical persistence:** document store

**Observations**
- Unstructured and structured, ACID not required
- Sensors, meters, geolocation
- **Typical persistence:** key-value store

**Transactions**
- Require structure, ACID
- Orders, payroll, purchases, trades
- **Typical persistence:** RDBMS

ACID = atomicity, consistency, isolation and durability
RDBMS = relational database management system

Source: Gartner (May 2013)
Significant Market Trends Impacting DBMS

- Not just disk-based, but also in-memory
- Not just relational, but also NoSQL
- Not just DBMS, but also Hadoop and Spark
- Not just commercial, but also open source
- Not just on premises, but also in the cloud
- From the “Big 3” to a hundreds of contenders...
The Database Landscape Map

Non-relational zone
Relational zone
Grid/cache zone

Data Platforms Map
January 2016

Visit 451research.com/state-of-the-database-landscape for more information.
DATA GROWTH
Phenomenal Data Growth is the Norm

The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020

“There were 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days.”
– Eric Schmidt, of Google, said in 2010.

More than one-third of companies in a recent survey on database growth report data stores growing at a rate greater than 20% a year
– “Managing the Rapid Rise in Database Growth” 2011

## Data Storage and Size Terminology

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Term</th>
<th>Size</th>
<th>Power of 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Byte</td>
<td>8 bits</td>
<td></td>
</tr>
<tr>
<td>KB</td>
<td>Kilobyte</td>
<td>1,024 bytes</td>
<td>$2^{10}$ bytes</td>
</tr>
<tr>
<td>MB</td>
<td>Megabyte</td>
<td>1,024 KB</td>
<td>$2^{20}$ bytes</td>
</tr>
<tr>
<td>GB</td>
<td>Gigabyte</td>
<td>1,024 MB</td>
<td>$2^{30}$ bytes</td>
</tr>
<tr>
<td>TB</td>
<td>Terabyte</td>
<td>1,024 GB</td>
<td>$2^{40}$ bytes</td>
</tr>
<tr>
<td>PB</td>
<td>Petabyte</td>
<td>1,024 TB</td>
<td>$2^{50}$ bytes</td>
</tr>
<tr>
<td>EB</td>
<td>Exabyte</td>
<td>1,024 PB</td>
<td>$2^{60}$ bytes</td>
</tr>
<tr>
<td>ZB</td>
<td>Zettabyte</td>
<td>1,024 EB</td>
<td>$2^{70}$ bytes</td>
</tr>
<tr>
<td>YB</td>
<td>Yottabyte</td>
<td>1,024 ZB</td>
<td>$2^{80}$ bytes</td>
</tr>
<tr>
<td>BB</td>
<td>Brontobyte</td>
<td>1,024 YB</td>
<td>$2^{90}$ bytes</td>
</tr>
</tbody>
</table>

- 1 MB ~ 1 book
- ~300 MP3 songs
- Hubble generates 10 TB/year
- 128 TB (UTS)
- 4 PB (PBR RPN UTS) Db2 12
- 3000x the Library of Congress
- A gram of DNA can hold 490 EB
- Human brain can store 2.5 PB of data
Big Data Represents a Major IT Shift

- Shift from mostly internal data to information from multiple sources
- Shift from transactional to add analytical data
- Shift from structured to add unstructured data
- Shift from persistent data to data constantly on the move

Actually, less of a shift and more of an addition
So What is Big Data?

The essence of the Big Data movement is being able to derive meaning quickly from vast quantities of data – both structured and unstructured – in order to improve business decision making.

It is not just about counting up your V’s: Volume, Velocity, Variety, etc.

- **Business Intelligence** – structured queries
- **Cloud Computing** – access to large pools of computing power available as needed
- **Distributed data** - data is usually physically distributed across a network using inexpensive commodity hardware
- **NoSQL and Hadoop** – new data persistence methods geared for storing and processing large amounts of data
- **Sensors** – more sensors producing more data more frequently
- **Analytical tools** – for data from multiple sources and of variable types
- **Mobile and Networked devices** – The number of networked devices overtook the global population of humans in 2011
- **The Internet of Things** – machine-generated data read and used by other machines
- **Traditional RDBMS** – your existing systems and knowledge are NOT obsolete
Types of Data in Big Data Projects

What types of data does your organization collect or plan to collect as part of its big data program?

- Structured transaction data: 82%
- Customer emails/letters/survey responses: 43%
- Social media activity data: 41%
- Internet clickstream data: 38%
- Web server logs: 37%
- Computer/network log files: 33%
- Social media comments/blog posts: 27%
- Multimedia data: videos, digital images, audio files, etc.: 26%
- Cell phone call-detail records/location data: 23%
- Sensor data: 23%
- Other: 5%

Source: 2013 BI and Data Warehousing Survey
http://searchbusinessanalytics.techtarget.com/report/2013-BI-Data-Warehousing-Survey-Results
Architecture Used for Big Data Projects

What technologies does your organization use or plan to use to support its big data environment?

- **Mainstream relational databases/data warehouses**: 55%
- **Analytical databases (columnar, MPP, etc.)**: 52%
- **Data warehouse appliances**: 46%
- **Hadoop/MapReduce**: 41%
- **Data virtualization software**: 28%
- **Complex event processing/real-time data integration tools**: 26%
- **NoSQL databases**: 21%
- **Other**: 3%

Source: 2013 BI and Data Warehousing Survey
http://searchbusinessanalytics.techtarget.com/report/2013-BI-Data-Warehousing-Survey-Results
Technologies Adopted for Big Data Projects

- Relational databases (e.g., Oracle, DB2, etc.) - 64%
- Hadoop/MapReduce - 28%
- Other Open Source technologies - 20%
- NoSQL databases - 20%
- R - 12%
- Don’t know/unsure - 17%
- Other - 4%

Source: Survey of 304 data managers and administrators who are subscribers to Database Trends & Applications, 2013 BIG DATA OPPORTUNITIES SURVEY, Unisphere Research, May 2013.
SQL Still Top Tool of Data Scientists

Source: 2015 Data Science Salary Survey, by O'Reilly
WHAT ABOUT DBA?
DBA Trends

Add the following DBA trends to the mix and things can start to look somewhat dire:

- Fewer DBAs are being asked to manage more data
  - Although more and more data is being stored and accessed – as evidenced by the Big Data trend – that is **not** translating into additional DBAs being hired.

- Many DBAs are tasked with managing multiple DBMSes
  - Most DBAs are **responsible for multiple databases** from multiple vendors.
  - Most companies run multiple databases and are **open to adding new database platforms** if there is a need to do so.

“The Real World of the Database Administrator” by Dr. Elliot King, Unisphere Research, March 2015
More Databases Being Managed...

**Figure 6:** Approximately how many database instances does each DBA manage?

<table>
<thead>
<tr>
<th>Number of Instances</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>1.14%</td>
</tr>
<tr>
<td>2 to 10</td>
<td>27%</td>
</tr>
<tr>
<td>11 to 25</td>
<td>21.29%</td>
</tr>
<tr>
<td>26 to 100</td>
<td>28.52%</td>
</tr>
<tr>
<td>101 to 500</td>
<td>8.75%</td>
</tr>
<tr>
<td>&gt;500</td>
<td>2.66%</td>
</tr>
<tr>
<td>Don’t know</td>
<td>10.65%</td>
</tr>
</tbody>
</table>

**Figure 7:** Is the number of databases for which each DBA is responsible increasing, decreasing, or staying the same?

- Increasing: 72.31%
- Staying the same: 22.69%
- Decreasing: 5%
Different Types of DBMSes Being Managed

**Figure 8:** How many database platforms (i.e., platforms from different vendors) is each DBA responsible for managing in your organization?

- One: 30.62%
- Two: 41.47%
- Three to five: 20.54%
- More than five: 7.36%

**Figure 9:** Are the DBAs that are responsible for managing relational database management systems also responsible for managing non-relational systems (such as NoSQL and Hadoop)?

(Only respondents who had deployed Hadoop or NoSQL technology)

- Yes: 68%
- No: 32%
DBA Activities Are Varied

How DBAs Spend Their Time on Top Database Activities

What stands out when looking at how DBAs spend their time is that the pie chart is almost “equally” split between these top six activities. The top two activities, of Security and Database changes, take 23% and 22% respectively. Adding the next two activities (Optimization – 19%, Backup & Recovery – 13%) and we already account for 77% of the DBAs time.

Top 4 comprise 77%

Source: 2018 Database DevOps Survey, Dbmaestro
Complexity is Increasing

- To what degree has the complexity of your database environment changed over the past 5 years?

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>DBAs</th>
<th>Execs/Mgrs</th>
<th>Programmer/arch./cons.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extremely more complex:</td>
<td>6%</td>
<td>5%</td>
<td>6%</td>
<td>6%</td>
</tr>
<tr>
<td>Significantly more complex:</td>
<td>40%</td>
<td>36%</td>
<td>40%</td>
<td>42%</td>
</tr>
<tr>
<td>Somewhat more complex:</td>
<td>43%</td>
<td>48%</td>
<td>43%</td>
<td>45%</td>
</tr>
<tr>
<td>No noticeable change:</td>
<td>10%</td>
<td>9%</td>
<td>10%</td>
<td>7%</td>
</tr>
<tr>
<td>Less complex:</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>Don’t know/unsure:</td>
<td>1%</td>
<td>2%</td>
<td>1%</td>
<td>0%</td>
</tr>
</tbody>
</table>

*Programmers, architects, consultants, project managers

Source: 2016 Database Lifecycle Management survey, conducted by Unisphere Media
And Let’s Not Forget DevOps

- The 2017 State of Database DevOps report from Redgate revealed that in 75% of the 1,000 organizations responding developers were responsible for both application and database development.
  - Developers also built the database deployment scripts in 60%
  - More frequent changes are being migrated to production
  - Agile is mainstream and DevOps is increasing in acceptance
    - But slower for legacy applications
  - “Dev” tends to overshadow the “Ops” portion
    - Control of process is driven by development
    - Sometimes without the traditional control and oversight of DBA
Always Available

• And downtime is non-existent

39% cannot tolerate an outage of greater than 1 minute.

Data management staff as a percentage of IT staff has risen a meager 0.5%.

Computer Economics, 2015
DBA Growth?

Trend in Database Administrators as Percentage of IT Staff

<table>
<thead>
<tr>
<th>Year</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>3.3%</td>
</tr>
<tr>
<td>2014</td>
<td>3.3%</td>
</tr>
<tr>
<td>2015</td>
<td>2.9%</td>
</tr>
<tr>
<td>2016</td>
<td>3.0%</td>
</tr>
<tr>
<td>2017</td>
<td>2.8%</td>
</tr>
</tbody>
</table>

Source: Computer Economics, 2017
And Things Aren’t Getting Any Better

By the year 2020

The world will generate 50X as much data.

But the IT staff who manages it will only grow 1.5X.

QuinStreet Enterprise Research, 2014
What About Autonomics?

- Autonomics is more than mere automation...
  - Automation is good, but autonomics adds intelligence

- **Autonomic computing** refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users.
  - An autonomics initiative ultimately aims to:
    - Develop computer systems capable of self-management;
    - Overcome the rapidly growing complexity of computing systems management
    - Reduce the barrier that complexity poses to further growth.

Characteristics of Autonomic Systems

- The system is *automatic*, meaning it makes decisions on its own, using:
  - Accumulated performance and usage metrics
  - High-level policies developed by administrators
- The system is *adaptive*, meaning it can automatically adapt to changing conditions.
- The system is *aware*, meaning it can monitor (or sense) its operational context as well as its current state to determine if it is reaching its specific purpose.
- Goal is to allow for the system to be *self-managing*, without human interaction being required for the system to optimize and administer itself. There are many aspects to “self” managing...
  - Self-configuration; Self-healing; Self-optimization; Self-protection; Self-inspection; Self-organization.
Part 2

DB2 MODERNIZATION
Bottom Line on Today’s Db2 Usage

Db2 is being asked to do more…

…with larger amounts and more types of data…

…being accessed more rapidly and from more sources…

…without any prolonged downtime permitted…

…and with fewer DBAs devoted specifically to Db2 than ever before
Db2 at the Extremes

DDF Transaction rates:

- 1000 or more per second (multi member DSG);
- 750 or more per second (single Db2 subsystem)

DDF share of workload is increasing

- Up to 95% at the extreme

Buffer Pool Size – extreme is 100 GB+

- But this is becoming more common with higher z/OS memory limits

Members in a Db2 data sharing group

- Extreme is more than 20
# 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>2015-07-06</td>
<td>2017-09-30</td>
</tr>
<tr>
<td>11</td>
<td>2013-10-25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2016-10-21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Db2 was announced on June 7, 1983

No skip level release
New and Deprecated Structures/Processes

• If you look at the applications and databases comprising your Db2 environment today, there should be many parts of it that look VERY different from 10 or 15 years ago...
  • Table Spaces – universal (PBG, range)
  • Unstructured Data
    • LOBs and PureXML
  • New data types and SQL statements
  • Incompatibilities (e.g. BIFs)
  • More dynamic SQL
  • Deprecated features removed including:
    • Simple TS
    • DBRMs in plans
    • Private protocol distributed data
    • Synonyms
    • Etc.
Types of Table Spaces

- **Simple** – max size 64 GB
  - Contains one or more tables
  - No new *simple table space* as of Db2 9
  - Replace with universal table space partition-by-growth
- **Segmented** – max size 64 GB
  - Contains one or more tables
  - Segments between 4 and 64 (multiples of 4)
- **Partitioned** - up to 4,096 partitions of up to 64 GB each
  - Contains one table
  - Usually used for larger tables
  - Index-controlled (old) versus table-controlled
- **Universal** - up to 128 TB
  - Combine partitioned and segmented characteristics
  - Can contain both a base and a clone table
  - Two types: Partition-By-Growth and Partition-By-Range
- **LOB** – up to 16 TB
- **XML**
Universal Table Spaces

- Many new Db2 features require Universal Table Spaces (UTS) in order to function
  - Introduced in Db2 9 for z/OS - combine the benefits of segmented space management with partitioned table space organization
  - Two types: PBG and PBR – can be up to 128 TB
  - You can alter existing table spaces to universal table spaces by using the ALTER TABLESPACE statement
  - Shops will be converting to UTS... Why?
    - UTS are the future and earlier TS eventually will go away
    - Larger size
    - Newer features that only work with UTS include: clone tables, hash-organized tables, currently committed locking, pending DDL, inline LOBs, XML multi-versioning, ALTER TABLE with DROP COLUMN
Changing Your Table Spaces

- **Simple Table Space**: Single table only
- **Segmented Table Space**: Single table only
- **Classic Partitioned Table Space**: Table-controlled
- **PBGrowth Table Space**: One Way
- **PBRange Table Space**: Two Ways
- **Hash Table Space**: Two Ways

Source: Db2 10 for z/OS Technical Overview (Redbook)
Structured versus Unstructured Data

**Structured data remains the bedrock** of the information infrastructure in most organizations, **but unstructured data is growing in importance.**

Raw data growth is only part of the story.

More data types are being captured, stored, and made available for analysis. More external data sources, too.

Unstructured data accounts for 90% of all digital information (according to International Data Corp.


The most important driver for the growth of unstructured data is internally generated documents and email.

Source: "The Real World of the Database Administrator" by Dr. Elliot King, Unisphere Research, March 2015
Structured and Unstructured Data

**Figure 12:** How fast is the amount of structured data growing annually?

- Less than 10%: 8.66%
- At least 10%, but less than 25%: 43.31%
- At least 25%, but less than 50%: 25.2%
- At least 50%, but less than 75%: 6.69%
- At least 75%, but less than 100%: 1.97%
- 100% or more (i.e., the amount of data under management is doubling annually): 2.36%
- Don’t know: 11.81%

**Figure 13:** How fast is the amount of unstructured data growing annually?

- Less than 10%: 17.39%
- At least 10%, but less than 25%: 20.95%
- At least 25%, but less than 50%: 18.58%
- At least 50%, but less than 75%: 3.56%
- At least 75%, but less than 100%: 2.77%
- 100% or more (i.e., the amount of data under management is doubling annually): 3.95%
- Don’t know: 32.81%
Unstructured Data Usage is Growing

• Requiring LOBs
  • Better supported than in the past

• More use cases requiring non-traditional data
  • Not just a check image but
    • Large text documents
    • Audio
    • Video
    • More complex images and photos

• And you’re getting more LOBs whether you want them or not...
## LOBs in the Db2 Catalog

<table>
<thead>
<tr>
<th>Version</th>
<th>Table Spaces</th>
<th>Tables</th>
<th>Indexes</th>
<th>Columns</th>
<th>LOB columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>11</td>
<td>25</td>
<td>27</td>
<td>269</td>
<td>0</td>
</tr>
<tr>
<td>V3</td>
<td>11</td>
<td>43</td>
<td>44</td>
<td>584</td>
<td>0</td>
</tr>
<tr>
<td>V5</td>
<td>12</td>
<td>54</td>
<td>62</td>
<td>731</td>
<td>0</td>
</tr>
<tr>
<td>V7</td>
<td>20</td>
<td>84</td>
<td>118</td>
<td>1212</td>
<td>2</td>
</tr>
<tr>
<td>V8</td>
<td>22</td>
<td>85</td>
<td>132</td>
<td>1265</td>
<td>2</td>
</tr>
<tr>
<td>V9</td>
<td>28</td>
<td>104</td>
<td>165</td>
<td>1652</td>
<td>6</td>
</tr>
<tr>
<td>V10</td>
<td>95</td>
<td>134</td>
<td>233</td>
<td>2036</td>
<td>36</td>
</tr>
<tr>
<td>V11</td>
<td>108</td>
<td>143</td>
<td>250</td>
<td>2202</td>
<td>42</td>
</tr>
</tbody>
</table>

*13 new LOB columns in Db2 12 for z/OS*
Using LOBs Requires Additional Care & Feeding

**Issues with LOBs**

- Size and amount of data
- LOB Logging
- SQL Restrictions
- Managing LOB Inconsistencies
## Typical Average Size for Large Objects

<table>
<thead>
<tr>
<th>Object</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank checks</td>
<td>30 KB</td>
<td>40 KB</td>
</tr>
<tr>
<td>Small image</td>
<td>30 KB</td>
<td>50 KB</td>
</tr>
<tr>
<td>Large image</td>
<td>200 KB</td>
<td>3 MB</td>
</tr>
<tr>
<td>Color image</td>
<td>20 MB</td>
<td>40 MB</td>
</tr>
<tr>
<td>Radiology image</td>
<td>40 MB</td>
<td>60 MB</td>
</tr>
<tr>
<td>Video</td>
<td>.5 GB/hour</td>
<td>-</td>
</tr>
<tr>
<td>Feature length movie</td>
<td>1 GB/hour</td>
<td>-</td>
</tr>
<tr>
<td>High resolution video</td>
<td>3 GB/hour</td>
<td>-</td>
</tr>
<tr>
<td>High resolution movie</td>
<td>5 GB/hour</td>
<td>6 GB</td>
</tr>
<tr>
<td>High definition TV</td>
<td>720 GB/hour</td>
<td>-</td>
</tr>
</tbody>
</table>

**Source:** LOBs with Db2 for z/OS: Stronger and Faster IBM RedBook (SG24-7270)
LOB Logging Considerations

- Do you want to log changes to LOB columns?
  - Probably not, because they are so large
- Logging can be turned off using the NOT LOGGED parameter for the LOB table space

```
CREATE LOB TABLESPACE PHOTOLTS
  IN DSN8D11A
  USING STOGROUP DSN8G110
    PRIQTY 3200
    SECQTY 1600
  LOCKSIZE LOB
  BUFFERPOOL BP16K0
  GBPCACHE SYSTEM
  NOT LOGGED
  CLOSE NO;
```
## SQL Restrictions on LOB Columns

<table>
<thead>
<tr>
<th>Context of usage</th>
<th>LOB (CLOB, DBCLOB, or BLOB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A GROUP BY clause</td>
<td>Not allowed</td>
</tr>
<tr>
<td>An ORDER BY clause</td>
<td>Not allowed</td>
</tr>
<tr>
<td>A CREATE INDEX statement that creates an index using an expression</td>
<td>Not allowed except when the index is created using an expression, in which case an inline LOB column can be referenced as the source data type for the SUBSTR ad SUBSTRING built-in functions.</td>
</tr>
<tr>
<td>A SELECT DISTINCT statement</td>
<td>Not allowed</td>
</tr>
<tr>
<td>A MERGE statement</td>
<td>Cannot be used in the context of an INCLUDE column-name clause</td>
</tr>
<tr>
<td>A subselect of a set operation except UNION ALL</td>
<td>Not allowed</td>
</tr>
<tr>
<td>Predicates</td>
<td>Cannot be used in any predicate except EXISTS, LIKE, and NULL. This restriction includes a simple-when-clause in a CASE expression. <em>expression WHEN expression</em> in a simple-when-clause is equivalent to a predicate with <em>expression=expression</em>.</td>
</tr>
<tr>
<td>The definition of primary, unique, and foreign keys</td>
<td>Not allowed</td>
</tr>
<tr>
<td>Check constraints</td>
<td>Not allowed</td>
</tr>
</tbody>
</table>
What Type of Things Can Go Wrong?

• Errors with LOBS occur when there are inconsistencies between the three main component objects.
• Whereas normal Db2 indexes can be inconsistent with their associated table, the issues are multiplied for LOB indexes:
  1. The ROWID-Version number in the Base Table row may not be found in the LOB index.
  2. There may be entries in the LOB index that are not referenced by any row in the base table.
  3. The LOB data itself may not be where the LOB index points to.
  4. There may be LOBs in the LOB table space that are not referenced by the LOB index.
The Consequences of LOB Pointer Issues

• If the LOB index is inconsistent with the *Base Table* data, the **LOB data is lost**, it cannot be accessed. There is no direct access to the LOB TS except through the LOB index.

• If the LOB index is inconsistent with the *LOB TS*, Db2 will get **errors trying to access the LOB data** for that row.

• LOB data in the LOB TS, by virtue of its size, can be distributed over many different pages of the LOB TS. Db2 uses a structure of “MAP” pages to point to these data pages, which do not need to be contiguous. If all the data pages are not referenced by these “MAP” pages or if the “MAP” pages themselves are not properly referenced by a higher level “MAP” page, **LOB data will be lost**.
Modern Storage = RAID

• Mainframe disk, or DASD, is usually equated to a 3380 or 3390. In other words, physical hardware devices with a one-to-one relationship between a disk drive and a volume. The logical view is broken down as:
  • Track size, or the number of bytes per track.
    • 47476 for 3380
    • 56664 for 3390
  • Capacity, or the size of the device, in terms of number of tracks or gigabytes.
  • Device address, sometimes called device number, which is a thread onto which I/O operations are serialized by the operating system

• Today these physical devices are replaced by disk arrays.
  • An array is the combination of two or more physical disk storage devices in a single logical device or multiple logical devices.
What is RAID?

- RAID = Redundant Array of Independent Disks

The basic idea of RAID is this:

- Multiple disks configured together into an array.
- The array is perceived by the system to be a single disk device.
- Hot-swappable drives improve availability
  - A drive can be replaced while the array is up & running.

There are many levels of RAID technology

- Each delivers different levels of fault-tolerance and performance.
- Examples on next slide...
DS8000: RAID10

The IBM DS8000 disk array is a RAID10 implementation.

- RAID10 is implemented as a striped array whose segments are RAID 1 arrays.
- RAID 10 has the same fault tolerance as RAID level 1.
- RAID 10 has the same overhead for fault-tolerance as mirroring alone.

- High I/O rates are achieved by striping RAID 1 segments.
- Under certain circumstances, RAID 10 array can sustain multiple simultaneous drive failures.
- Excellent solution for sites that would have otherwise gone with RAID 1 but need some additional performance boost.
Storage Has Changed
...so DBAs Have Changed

Historical Worries

- Extreme data set placement
  - VCAT-defined data sets
- Managing Db2 stogroups
  - Carefully associating volumes to STOGROUPs
- Separation of data sets
  - careful placement to avoid contention
  - always putting indexes on separate devices from data
- Hyper-vigilant extent mgmt

“Recent” Storage Advances

- DFSMS for the Db2 Catalog (Db2 10)
- Sliding scale extent allocation (V8)
- Extent consolidation (z/OS v1.5)
  - If the new extent is adjacent to the old, they will be merged together
- Variable CI size (V8)
  - DSVCI DSNZPARM
- STOGROUPs work w/DFSMS (Db2 9)
  - Can specify DATACLAS, MGMTCLAS, and STORCLAS and omit the VOLUMES clause from the Db2 STOGROUP
CLONE Tables

• Quick introduction to cloning...
  • Basically, cloning creates a table with the exact same attributes as a table that already exists, except that it has no data.
  • It is created using the ALTER TABLE SQL statement with the ADD CLONE parameter.
  • The clone table is created in the same table space as the existing table.
    • But in a different VSAM data set.
  • After creating the clone table you can do whatever you want to do with it. LOAD it, INSERT rows to it, etc.
  • When the clone is ready to become the base table it is exchanged with the base table. This is done using the EXCHANGE SQL statement.
  • After running an EXCHANGE the clone becomes the “real” table and the previously “real” table becomes the clone - - and you can repeat the process.

• Bottom Line
  • You’ll need to manage the process, ensure adequate storage, and understand how it works...
IDAA – IBM Db2 Analytics Accelerator

• IBM Db2 Analytics Accelerator for z/OS is a high-performance appliance that integrates IBM Netezza and zEnterprise technologies.
  • The solution delivers extremely fast results for complex and data-intensive Db2 queries on data warehousing, business intelligence and analytic workloads.
  • Which is Great! But...
    • It complicates SQL performance tuning, database administration, and data movement/latency.
Db2 Security Modernization

• More granular control of System authority
  • Db2 10: SECADM, System DBADM, SQLADM
  • ACCESSCTRL | DATAACCESS

• Improved audit functionality
  • No expensive data collectors
  • Audit Policies are managed in the catalog
    • Audit policy does not require AUDIT clause to be specified

• Row Permission
  • Improved access by row contents

• Column Masks
  • Improved compliance

• TRUSTED CONTEXT and Roles
  • Authorized remote connections for applications
SQL is Getting More and More Complex

• New versions add more SQL features and functionality
  • Dynamic vs. Static
  • Multiple ways to code SQL and get the same result
    • Multiple ways to code a join (table1, table2) vs JOIN...ON
  • New functions, including OLAP functionality, temporal support...
  • New SQL “stuff” in Db2 Versions 9, 10, and 11:
    • TRUNCATE, DECIMAL FLOAT, VARBINARY, optimistic locking, FETCH CONTINUE, ROLE, MERGE, SELECT from MERGE, XML, FETCH FIRST & ORDER BY in subselect and fullselect, INTERSECT, EXCEPT, Indicator Variables, TIMESTAMP precision and time zones, Moving sums and averages, Inline and Non-inline SQL scalar functions, SQL table functions, extended implicit casting, RANK(), ROW_NUMBER(), XQuery, transparent archive query, IDAA/analytics, grouping sets, ROLLUP, Hadoop access...

• Great because you can do more things with SQL...

• But also problematic because:
  • It can be more confusing and difficult to learn
  • It is easier to create poor performing SQL
Common Table Expressions

- A common table expression, or CTE, allows a SQL statement to be defined using the WITH clause, and then referenced as a table within the rest of the SQL statement.
  - Contrast with a nested table expression, or NTE, which is defined in the FROM clause of an SQL statement.

```sql
WITH DTOTAL (deptno, totalpay) AS
  (SELECT deptno, sum(salary+bonus)
   FROM employee
   GROUP BY deptno)

SELECT deptno
FROM DTOTAL
WHERE totalpay =
  (SELECT max(totalpay)
   FROM DTOTAL);
```
Recursive SQL

- CTEs also enable us to write recursive SQL.
  - Fullselect of common table expression refers to itself in the FROM clause
  - Can be quite useful for bill of materials and organization hierarchy explosions
    - A single SQL statement can be used to traverse hierarchies
  - Be sure to add controls to stop
Recursive SQL Setup

CREATE TABLE ORG_CHART
(MGR_ID SMALLINT,
EMP_ID SMALLINT,
EMP_NAME CHAR(20));

<table>
<thead>
<tr>
<th>MGR_ID</th>
<th>EMP_ID</th>
<th>EMP_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>1</td>
<td>BIG BOSS</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>LACKEY</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>SUCKUP</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>BOOTLICKER</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>GRUNT</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>SUB-BOSS</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>LOW MAN</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>LOW WOMAN</td>
</tr>
</tbody>
</table>

CREATE TABLE ORG_CHART
(MGR_ID SMALLINT,
EMP_ID SMALLINT,
EMP_NAME CHAR(20));
Recursive SQL Example

WITH EXPL (MGR_ID, EMP_ID, EMP_NAME) AS
  (SELECT ROOT.MGR_ID, ROOT.EMP_ID, ROOT.EMP_NAME
   FROM ORG_CHART ROOT
   WHERE ROOT.MGR_ID = 3
  ) UNION ALL

  SELECT CHILD.MGR_ID, CHILD.EMP_ID, CHILD.EMP_NAME
  FROM EXPL PARENT, ORG_CHART CHILD
  WHERE PARENT.EMP_ID = CHILD.MGR_ID

  SELECT DISTINCT MGR_ID, EMP_ID, EMP_NAME
  FROM EXPL
  ORDER BY MGR_ID, EMP_ID;
The results of running this query would be:

<table>
<thead>
<tr>
<th>MGR_ID</th>
<th>EMP_ID</th>
<th>EMP_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>SUCKUP</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>SUB-BOSS</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>LOW MAN</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>LOW WOMAN</td>
</tr>
</tbody>
</table>
OLAP Functions

- RANK
- DENSE_RANK
- ROW_NUMBER

Example: This query will rank employees who have total compensation greater than $30,000, but order the results by last name. This allows you to rank data differently than the order in which it is presented.

```
SELECT EMPNO, LASTNAME, FIRSTNAME, 
    SALARY+BONUS+COMM AS TOTAL_COMP, 
    RANK() 
    OVER(ORDER BY SALARY+BONUS+COMM DESC) AS RANK_COMP 
FROM EMP 
WHERE SALARY+BONUS+COMM > 30000 
ORDER BY LASTNAME;
```
Moving Sums and Averages

- And Db2 10 gives us OLAP specifications for moving sums and moving averages.
  - Here is an example calculating the 30 day moving average for the stocks 'ABC' and 'XYX' during 2005:

```sql
CREATE VIEW V1 AS
SELECT SYMBOL, TRADINGDATE,
    AVG(CLOSINGPRICE) OVER (PARTITION BY SYMBOL
        ORDER BY TRADINGDATE
        ROWS BETWEEN 29 PRECEDING AND CURRENT ROW)
FROM DAILYSTOCKDATA
WHERE SYMBOL IN ('ABC', 'XYZ')
    AND TRADINGDATE BETWEEN DATE('2005-01-01') - 2 MONTHS AND '2005-12-31';

SELECT SYMBOL, TRADINGDATE, MOVINGAVG30DAY
FROM V1
WHERE TRADINGDATE BETWEEN '2005-01-01' AND '2005-12-31'
ORDER BY SYMBOL, TRADINGDATE;
```
Temporal Data: Business Time vs. System Time

• **Business Time** (aka application time or valid time)
  – Specifies when the facts stored in the database are true with respect to the real world.
  – These are the dates of interest to the business user interacting with the data.
  – Business time is useful for only certain types of data that change over time and the validity of the data is relevant to the application and users.

• **System Time** (aka transaction time)
  – Denotes the time when the fact became current in the database.
  – System time can be used to track the insertion and modification history of the data.
  – Unlike business time, transaction time may be associated with any database entity.
Temporal Data and Time Travel Queries

• Not going to get into all of the specifics and ins and outs of temporal data usage, but here are some SQL examples of querying temporal data:
  • What are the course terms on offer from April 30, 2009 until June 30, 2012?

```sql
SELECT COURSENO, TITLE, CREDITS, PRICE,
     BUS_START, BUS_END
FROM COURSE FOR BUSINESS TIME FROM '2009-04-30' TO '2012-06-30';
```

  • An UPDATE to change the price for the Ethics course for the time period from November 1, 2011 to February 2, 2012:

```sql
UPDATE COURSE
FOR PORTION OF BUSINESS_TIME FROM '2011-11-01' TO '2012-02-02'
SET PRICE = 325.00
WHERE COURSENO = 100;
```

  • Can impact multiple rows!
More Time Travel Queries (examples)

- Not going to get into all of the specifics and ins and outs of temporal data usage, but here are some SQL examples of querying temporal data:
  - DELETE course #220 for time period from July 3, 2011 to July 9, 2011

```
DELETE FROM COURSE
FOR PORTION OF BUSINESS_TIME FROM '2011-07-03' TO '2011-07-09'
WHERE COURSENO = 220;
```

- And a DELETE can cause data to be inserted!
- If the period being deleted is fully contained in a current single row, that row is deleted and two new rows are inserted with appropriate time spans

- Then there is SYSTEM time:

```
SELECT TITLE, CREDITS, PRICE
FROM COURSE FOR SYSTEM_TIME AS OF TIMESTAMP('2012-01-16')
WHERE COURSENO = 650;
```
Temporal Summary

- SQL gets more functionality and more clauses to learn
- Nevertheless, it is an improvement on implementing temporal data and queries without temporal support!
- Sometimes the impact of a statement seems to be contrary to our previous level of understanding
  - E.g.) A successful DELETE results in more rows in the table
- Can become very complex with both BUSINESS and SYSTEM TIME implementing in a bi-temporal table
- Will impact database design
  - RI and keys
Drivers of Dynamic SQL Growth

• Packaged applications use dynamic SQL
  • SAP R/3, PeopleSoft, Siebel, etc.
  • Easier to support multiple DBMSes that way

• Modern applications use dynamic SQL
  • Developed on distributed platforms and for the web
    • New developers are more familiar with GUI-based programming environments
    • Many of the current development tools provide better support for dynamic APIs (like JDBC), than they do for static SQL
    • Many developers never even sign on to mainframe/ISPF
      • Java and .NET developers
There is a Lot More, But We Cannot Cover it All

- RRF: Reordered Row Format
- No DBRM in PLANs
- Index on expressions
- PCTFREE for UPDATE
- Db2-managed data archiving
- Index compression
- DFSMS-controlled Db2 data sets
- INSTEAD OF triggers
- Histogram Statistics
- Materialized Query Tables
- Scrollable Cursors
- Improved optimization hints
- Improved database auditing

- SELECT from...
  - INSERT (V8)
  - UPDATE, DELETE and MERGE (9)
- MERGE aka UPSERT
- TRUNCATE
- Multi-row FETCH / INSERT
- INTERSECT and EXCEPT
- PLANMGMGT: Access Path Management
- SQL table functions
- SKIP LOCKED DATA
- USECURRENTLY-COMMITTED
- DSN_PROFILE_ATTRIBUTES
  - to model prod in test
Bottom Line on Today’s Db2 Usage

Db2 is being asked to do more...

...with larger amounts and more types of data...

...being accessed more rapidly and from more sources...

...without any prolonged downtime permitted...

...using and supporting new database structures and SQL constructs...

...and with fewer DBAs devoted specifically to Db2 than ever before.
Part 3 – The Finale

SOME GUIDANCE
Guidance

• You cannot manage a modern Db2 environment like you did back in the days of Db2 V7 or earlier
• Treat DBA as a Management Discipline
  • Proactive vs. Reactive
• Automate what you can
  • Turn tasks over to the computer to free up DBA time
  • Intelligent automation & autonometrics
• Embrace modern Db2 tools and utilities that understand the new digital landscape
  • Large amounts & types of data
  • Support new functionality/technologies
  • Always available / Easy to use
• Don’t ignore training!
Contact Information

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

E-mail: craig@craigsullins.com
Web: www.mullinsconsulting.com

http://mullinsconsulting.com/cm-book.htm

Craig S. Mullins
Mullins Consulting, Inc.
craig@craigsmullins.com

B13
It’s Not Your Daddy’s Db2!

Please fill out your session evaluation before leaving!