DB2 and Database Trends circa 2015

An overview of an industry in transition

Sponsored by:

DATAKINETICS
DATA PERFORMANCE & OPTIMIZATION

http://www.dkl.com

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

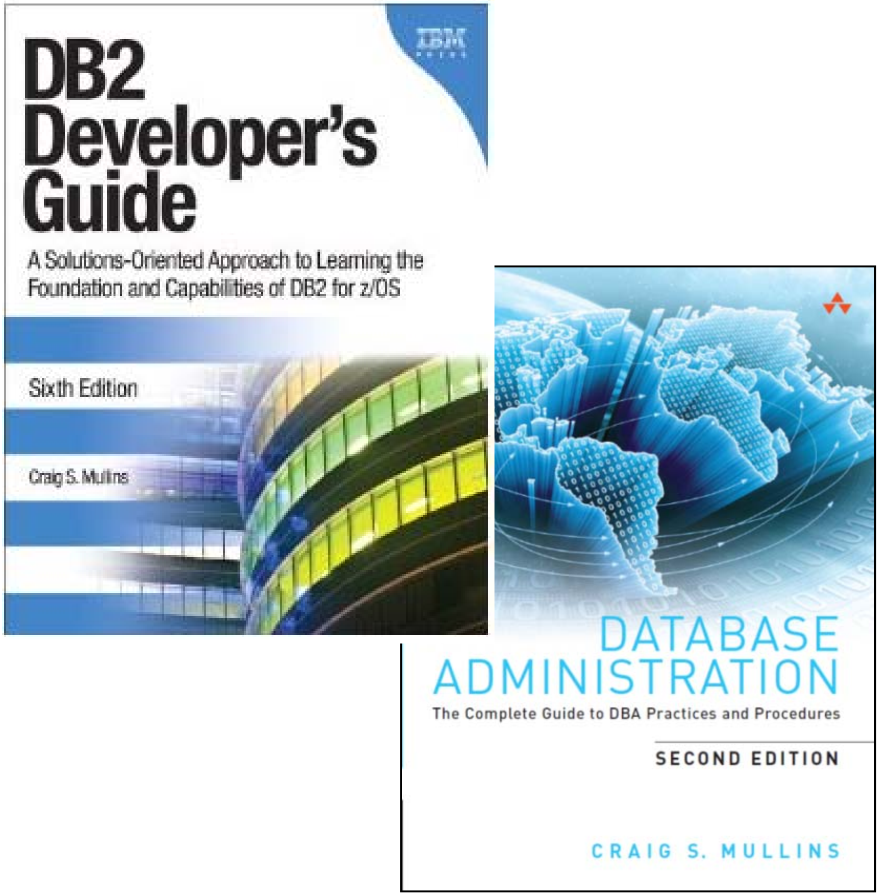
Author

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
Fax: 281.491.0637
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.


IBM
DB2 Developer's Guide
A Solutions-Oriented Approach to Learning the Foundation and Capabilities of DB2 for z/OS
Sixth Edition
Craig S. Mullins
DATABASE ADMINISTRATION
The Complete Guide to DBA Practices and Procedures
SECOND EDITION
CRAIG S. MULLINS

Agenda

Industry Trends

- Data growth – structured and unstructured
  - Big Data – what it is; why it matters – analytics
  - IoT, Mobile, Distributed
- DBA Trends:
  - Less downtime, fewer DBAs, more work
  - Autonomics
  - Types of DB2 users

DB2 Functionality Expansion

- New table spaces, unstructured data/LOBs
- Development trends: dynamic SQL, IDEs, web

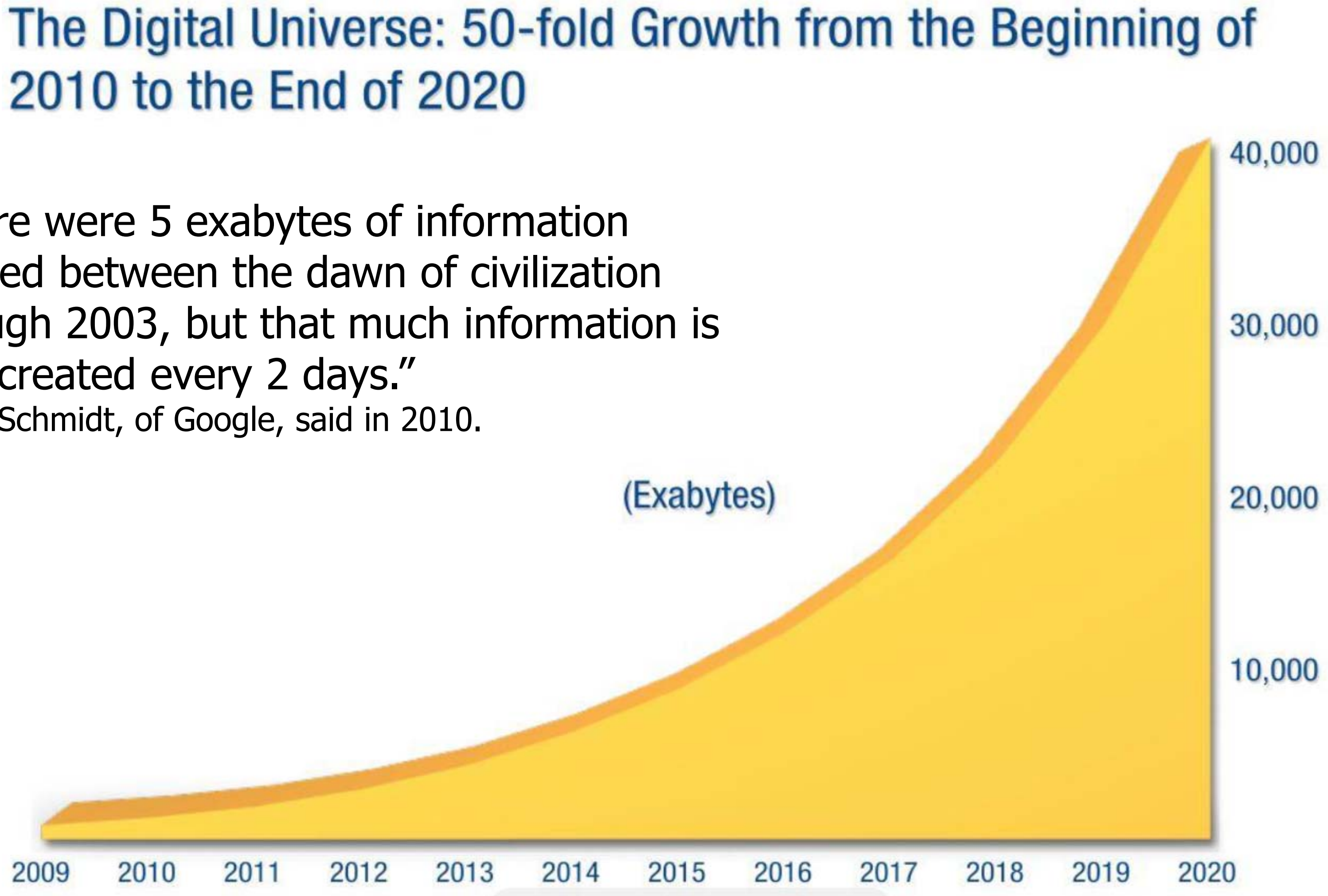
Guidance

- Treat DBA as a Management Discipline
- Automate what you can
- Embrace modern DB2 tools and utilities

Industry Trends

Phenomenal Data Growth is the Norm

“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.


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

Source: IDC's Digital Universe Study, sponsored by EMC, December 2012

Data Storage and Size Terminology

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

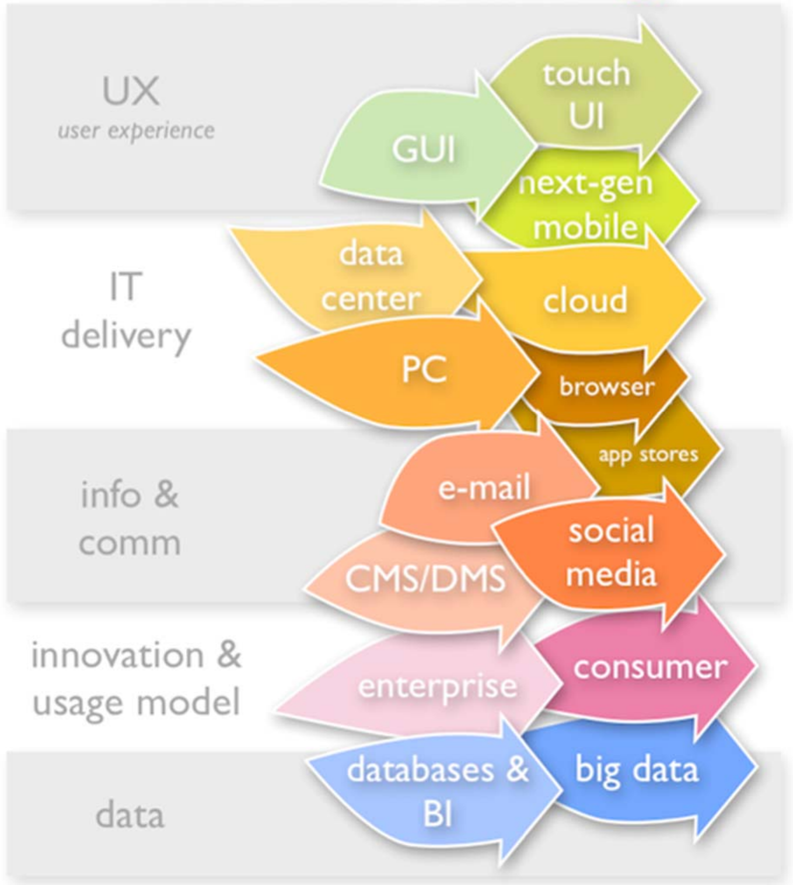
~1 book

128 TB (UTS)

3000x the Library of Congress

Big Data Represents a Major IT Shift

The Major Shifts in 21st Century Information Technology



From http://zdnet.com/blog/hinchcliffe on ZDNet

- 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
- Shift from local storage to storage in the cloud

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.

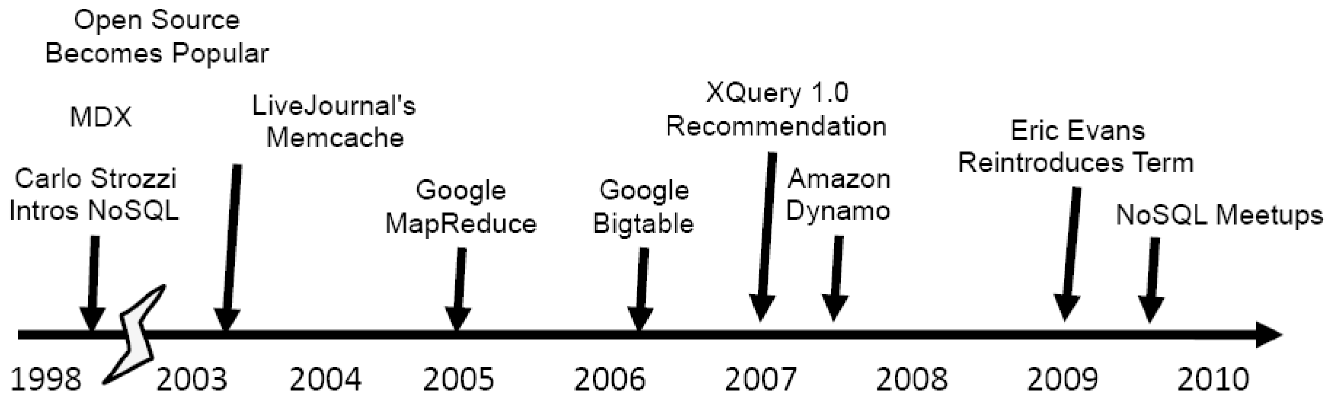
- **Business Intelligence – structured queries**
- **Cloud Computing – access to large pools of computing power available as needed**
- **Distributed data - data is usually physically distributed across a network using inexpensive commodity hardware**
- **NoSQL and Hadoop – new data persistence methods geared for storing and processing large amounts of data**
- **Analytical tools – for data from multiple sources and of variable types**
- **Networked devices – The number of networked devices overtook the global population of humans in 2011**
- **Sensors – more sensors producing more data more frequently**
- **The Internet of Things – machine-generated data read and used by other machines**

The Rise of NoSQL Database Systems

The NoSQL movement, which started out as "No SQL" has become "Not Only SQL"

- Non-relational and hybrid database systems
- NoSQL is based on the concept that relational databases are not the right database solution for ***all*** problems.

The World Wide NoSQL market is expected to reach $3.4 billion by 2018, with a compound annual growth rate of 21%.[1]



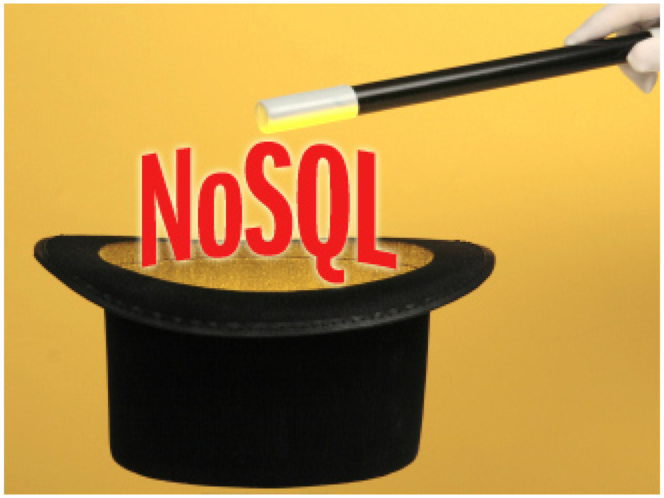
[1] Source: NoSQL Market Forecast 2013-2018, Tabular Analysis, Market Research Media, Ltd., July 2013

NoSQL Drivers

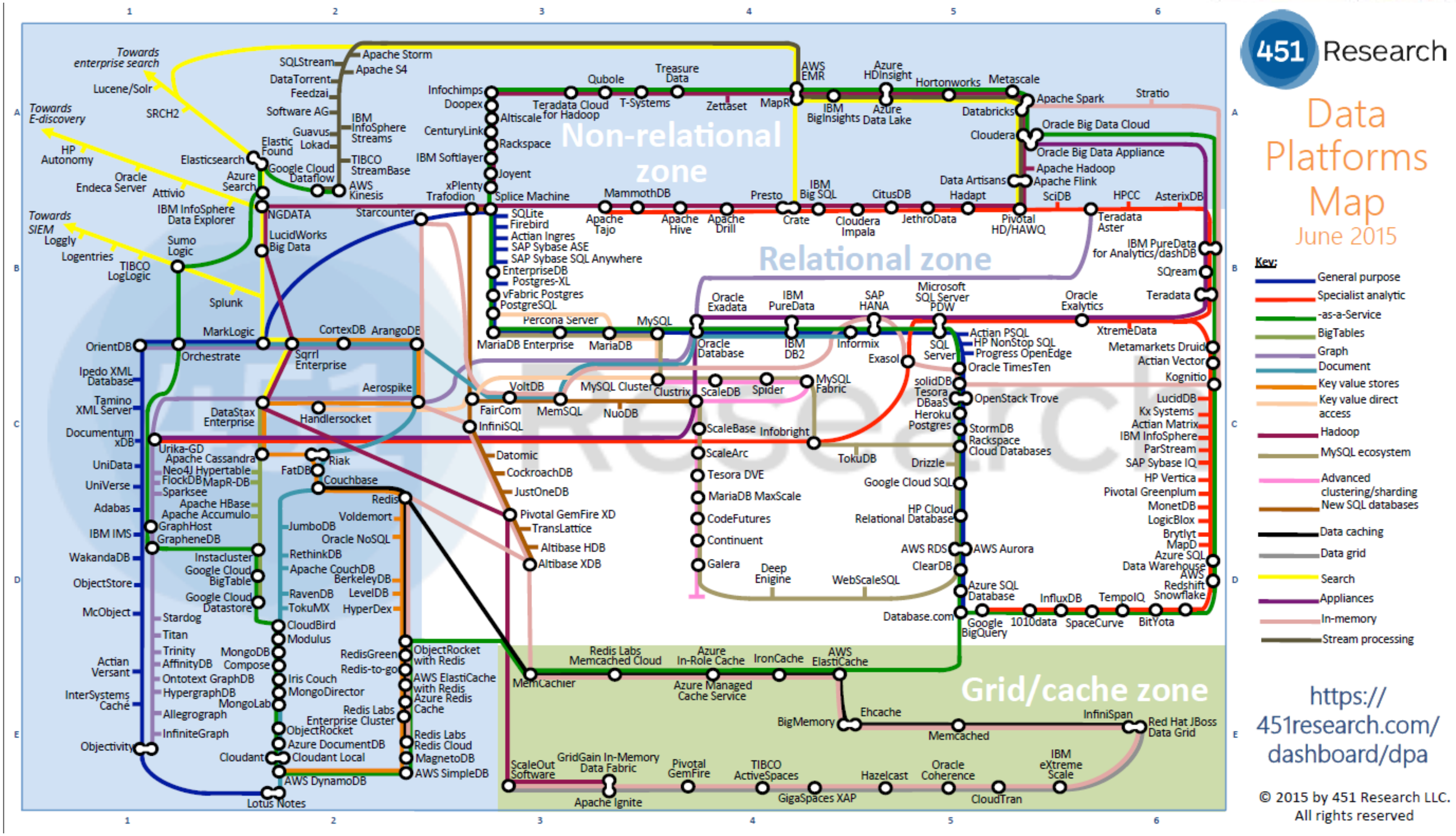
- **More users** – 1000 users used to be a lot and 10000 was extreme; the web renders these numbers quaint
- **More data** – difficult to scale to terabytes of data for traditional relational database applications
- **Different data** – unstructured data is not easily handled by relational databases
  - Documents, social media, etc.
- **More analytics** – different use cases can require different technology
- **Simplicity** – at least in terms of the features supported by the applications/systems
- **Rapid development** – schema-free databases deliver flexibility for quicker development

Types of NoSQL Database Systems

1. **Wide Column Store**
   HBase, Cassandra/Datastax, Accumulo…
2. **Document Store**
   Couchbase, MongoDB, RavenDB…
3. **Key/Value**
   Aerospike, Redis, Riak…
4. **Graph**
   Neo4j, InfiniteGraph, GraphBase…
5. **Multi-model**
   MarkLogic, Amazon DynamoDB…
6. **NewSQL**
   TransLattice, VoltDB, nuodb…
   - Relational/SQL DBMS
   - Scalable like NoSQL but ACID like SQL



The Database Landscape Map - June 2015



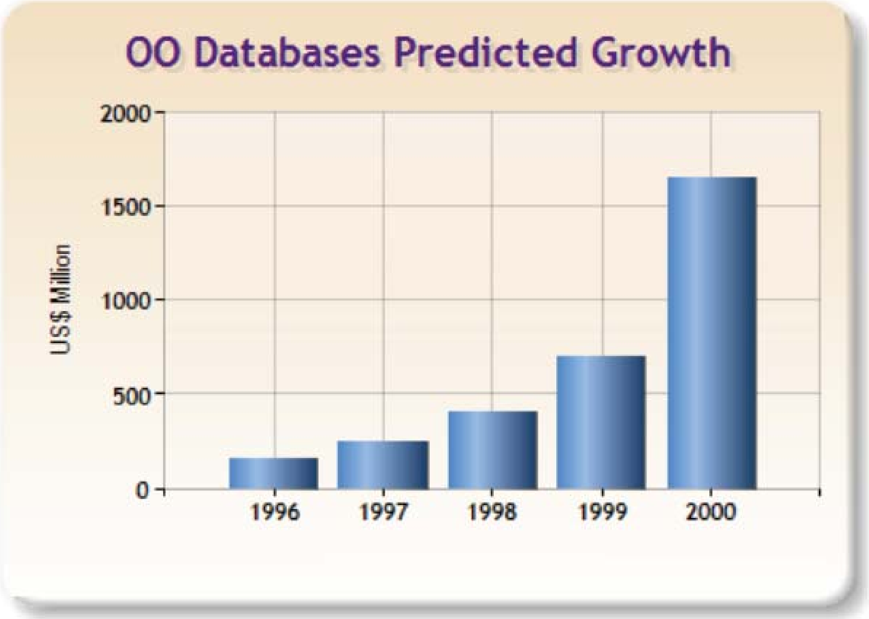
© 2015 Mullins Consulting, Inc.

http://db-engines.com/en/ranking

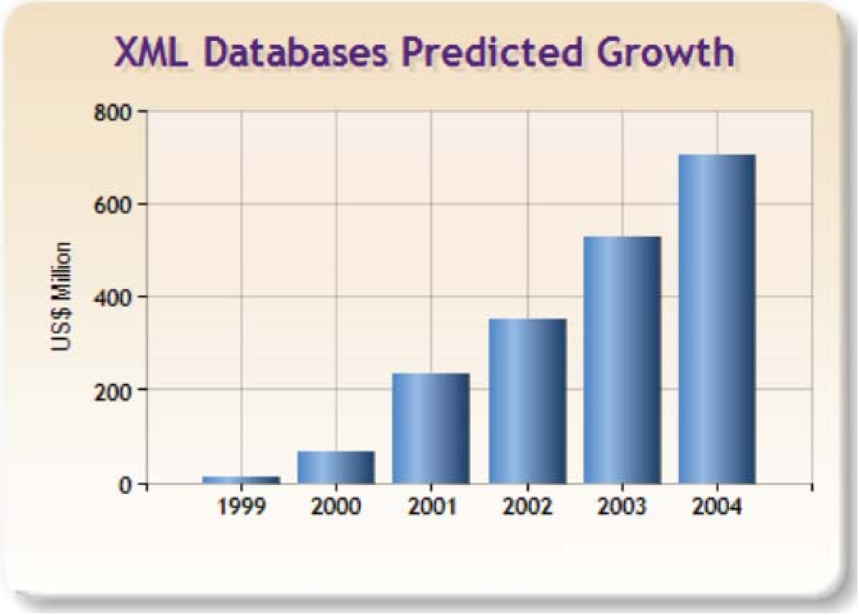
Remember Your Database History!

According to analysis by Wikibon’s David Floyer (highlighted in the Wall Street Journal), the NoSQL database market is expected to grow at a compound annual growth rate of nearly 60% between 2011 and 2017.”

- But let’s not forget (recent) history!

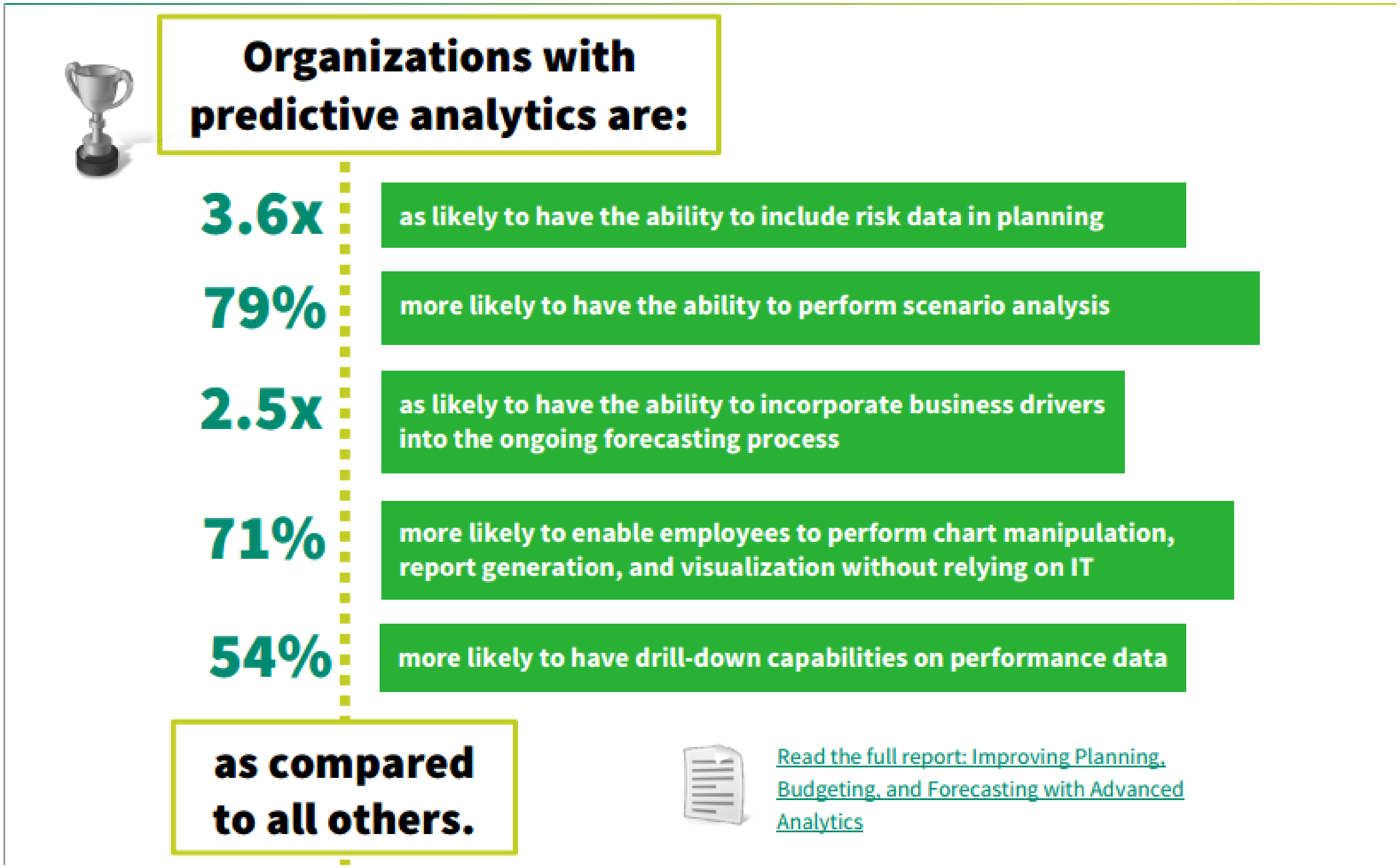


1990s ‣ Object Databases



2000s ‣ XML Databases

Analytics – Driving the Move to Big Data



Organizations with predictive analytics are:

3.6x as likely to have the ability to include risk data in planning

79% more likely to have the ability to perform scenario analysis

2.5x as likely to have the ability to incorporate business drivers into the ongoing forecasting process

71% more likely to enable employees to perform chart manipulation, report generation, and visualization without relying on IT

54% more likely to have drill-down capabilities on performance data

as compared to all others.

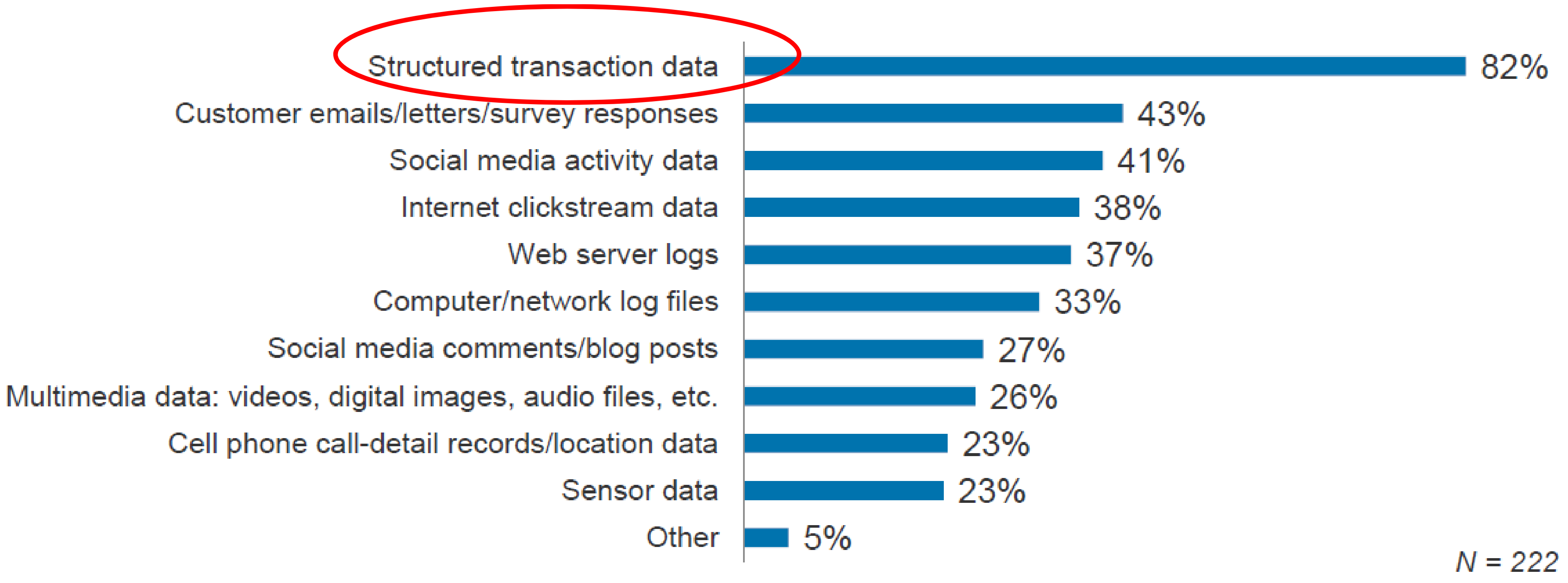
Read the full report: Improving Planning, Budgeting, and Forecasting with Advanced Analytics

Source: Aberdeen Group, 2015
http://resources.idgenterprise.com/original/AST-0146289_Infographic__Predictive_analytics_provides.PDFs

But things are not a grim as they may seem for us relational dinosaurs!

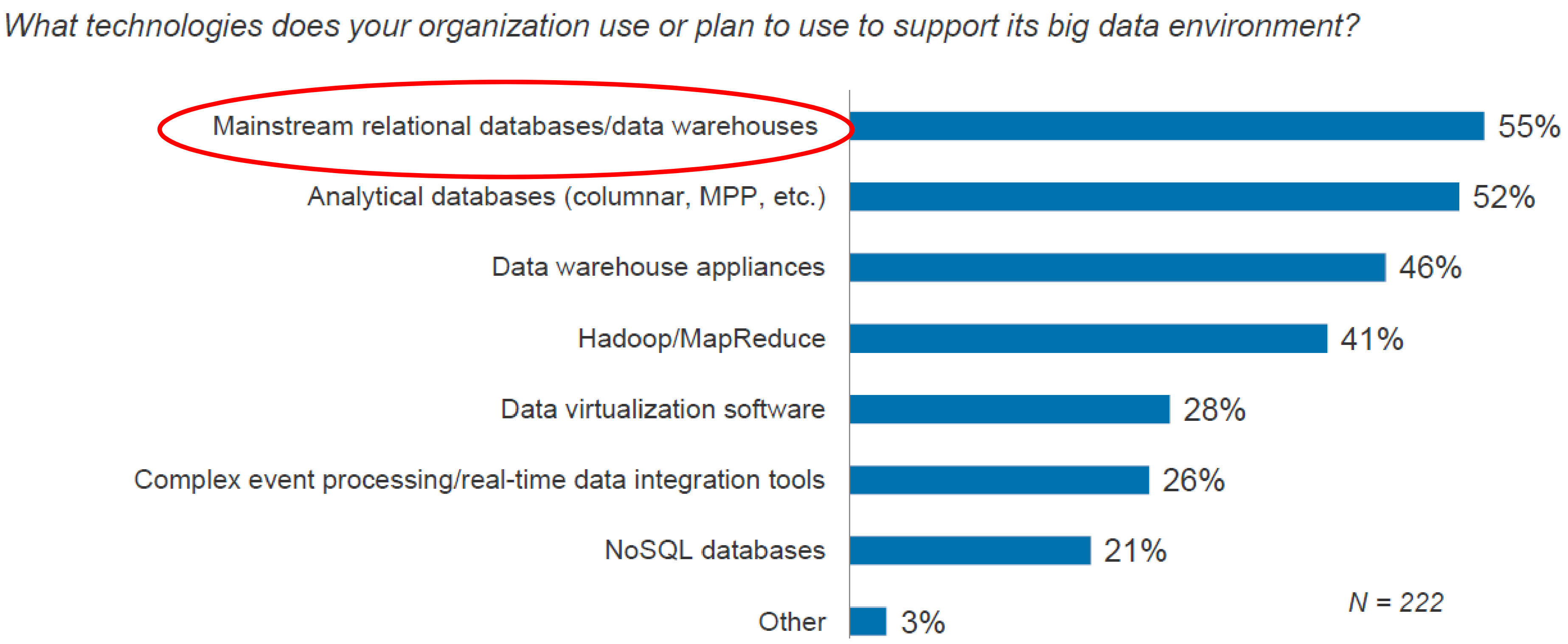
Types of Data in Big Data Projects

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



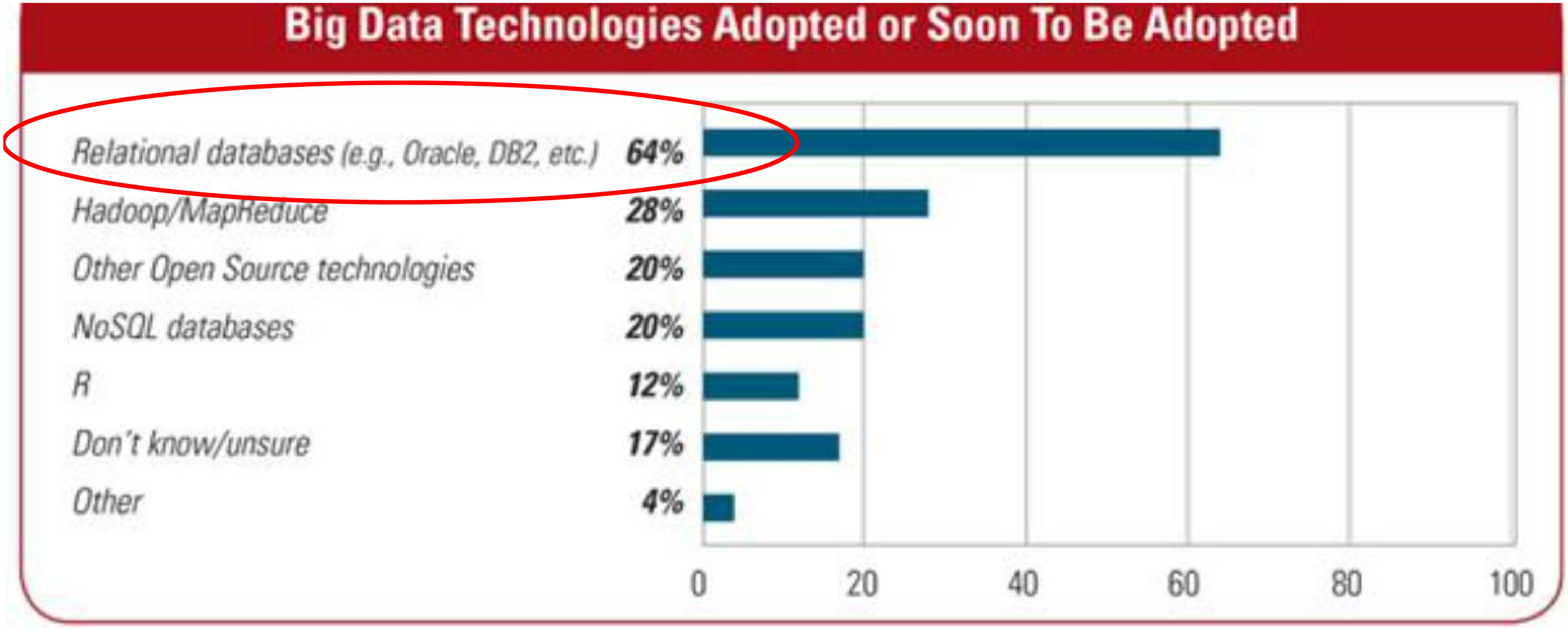
Source: 2013 BI and Data Warehousing Survey
http://searchbusinessanalytics.techtarget.com/report/2013-BI-Data-Warehousing-Survey-Results

Architecture Used for Big Data Projects



Source: 2013 BI and Data Warehousing Survey
http://searchbusinessanalytics.techtarget.com/report/2013-BI-Data-Warehousing-Survey-Results

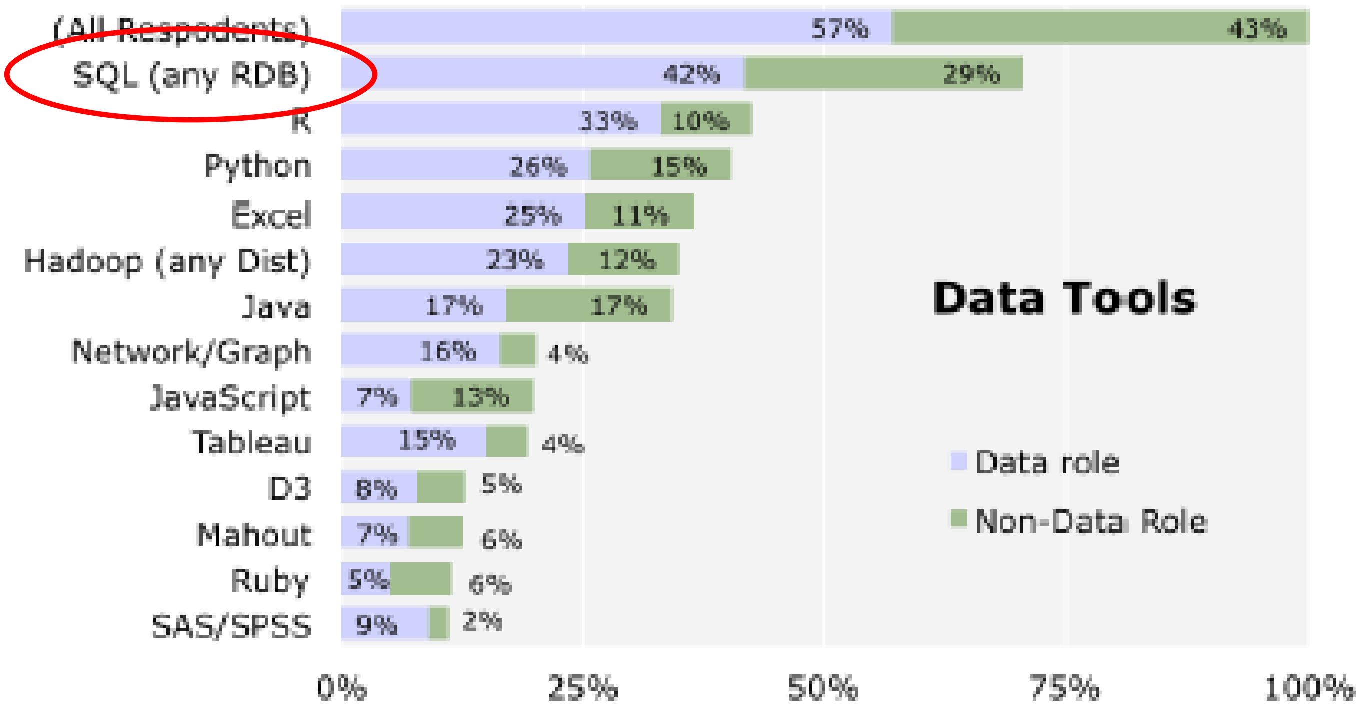
Technologies Adopted for Big Data Projects



Big Data Technologies Adopted or Soon To Be Adopted

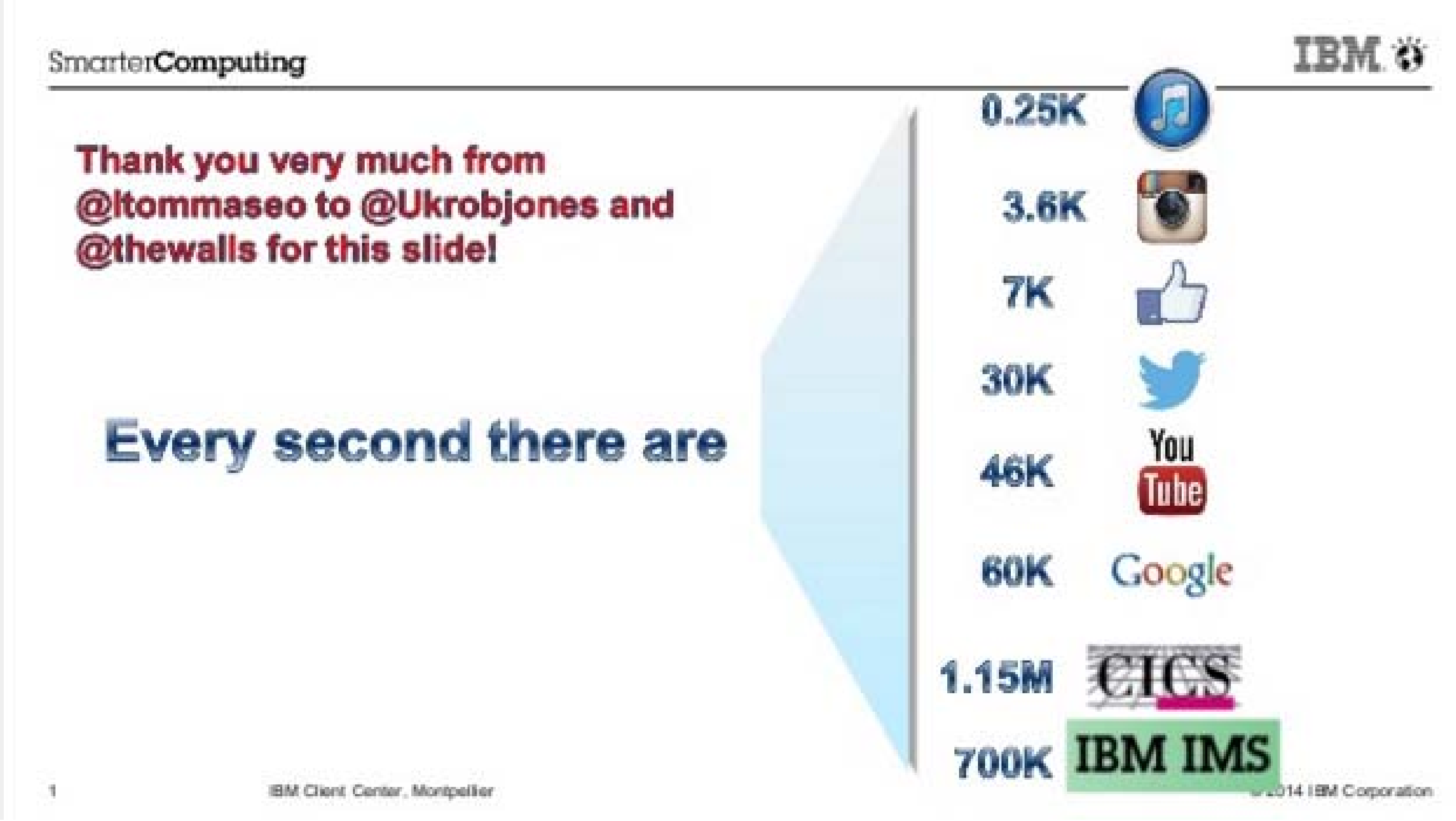
Source: Survey of 304 data managers and administrators who are subscribers to *Database Trends & Applications*, 2013 BIG DATA OPPORTUNITIES SURVEY, Unisphere Research, May 2013.

SQL Still Top Tool of Data Scientists



Source: 2013 Data Science Salary Survey, by O'Reilly conducted at the Strata Conference

And the mainframe continues to define what is BIG



SmarterComputing

IBM

Thank you very much from @ltommaseo to @Ukrobjones and @thewalls for this slide!

Every second there are

0.25K

3.6K

7K

30K

46K You Tube

60K Google

1.15M CICS

700K IBM IMS

1 IBM Client Center, Montpellier © 2014 IBM Corporation

So What is BIG for DB2???

- **Do we count number of rows, number of pages, or disk space consumed?**
- **Do we count just the base data or add up the space used by indexes as well?**
  - What about compressed data?
- **Does type of data matter?**
  - Traditional vs. multimedia
- **All we really should care about is how does the large amount of data impact our job.**
  - Think in terms of how storing larger amounts of data **complicates database administration** and data availability
  - In many cases, today we are talking **billions** of rows…
  - *Compare & contrast* using the number of *pages*, instead of rows.
    - Easier to compare the size of one table space to another

© 2015 Mullins Consulting, Inc.

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?

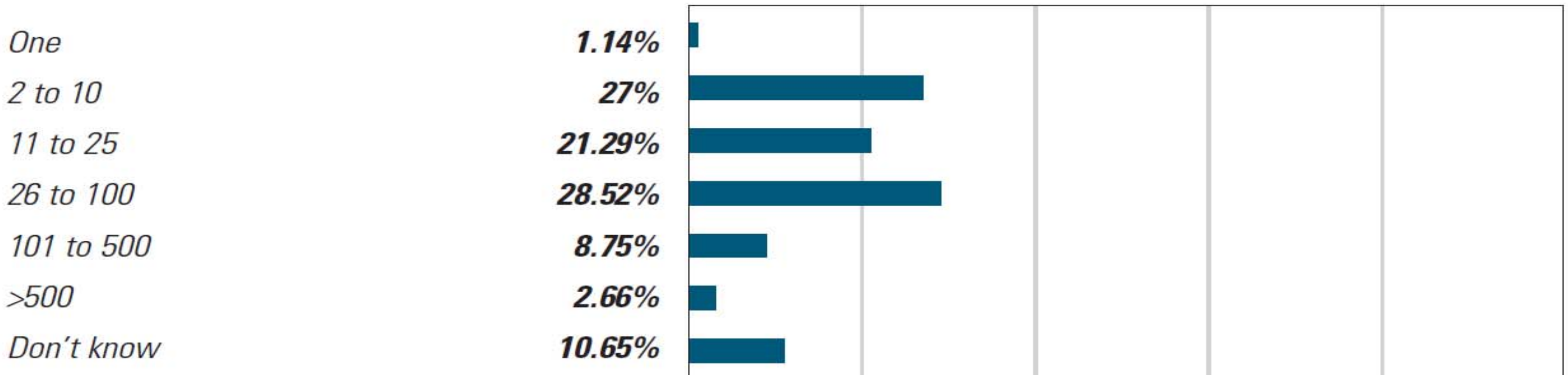
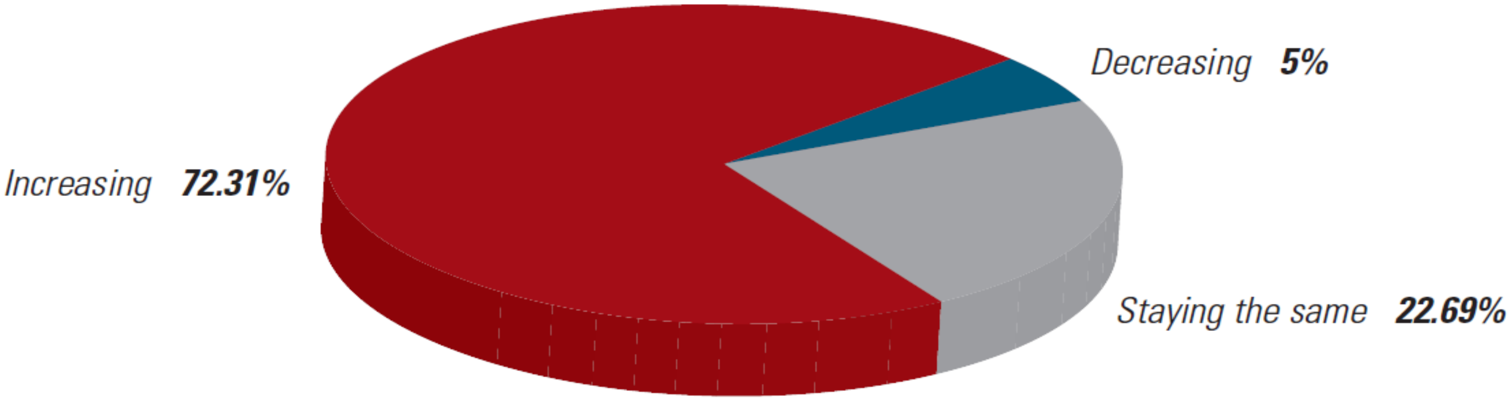


Figure 7: Is the number of databases for which each DBA is responsible increasing, decreasing, or staying the same?



And More 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?

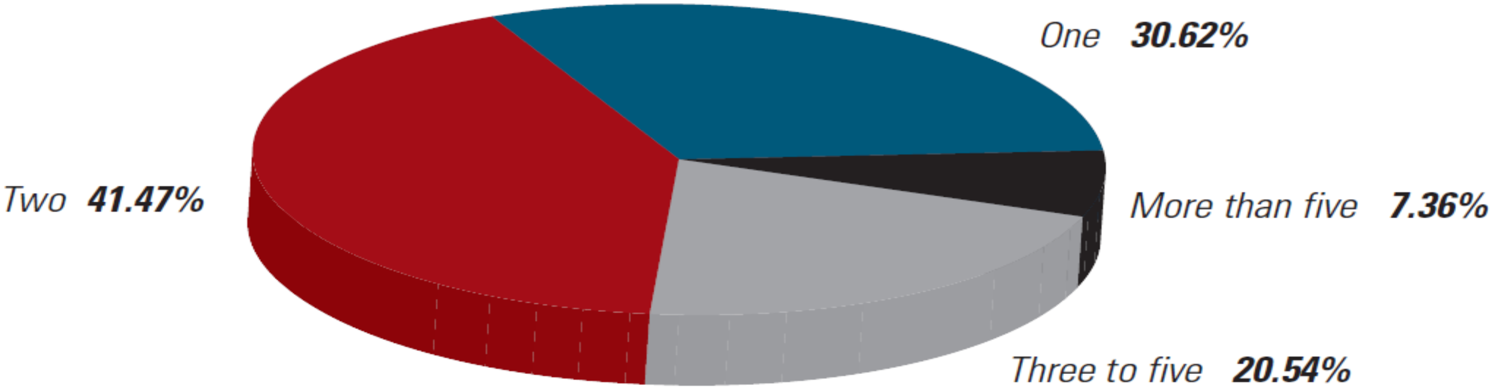
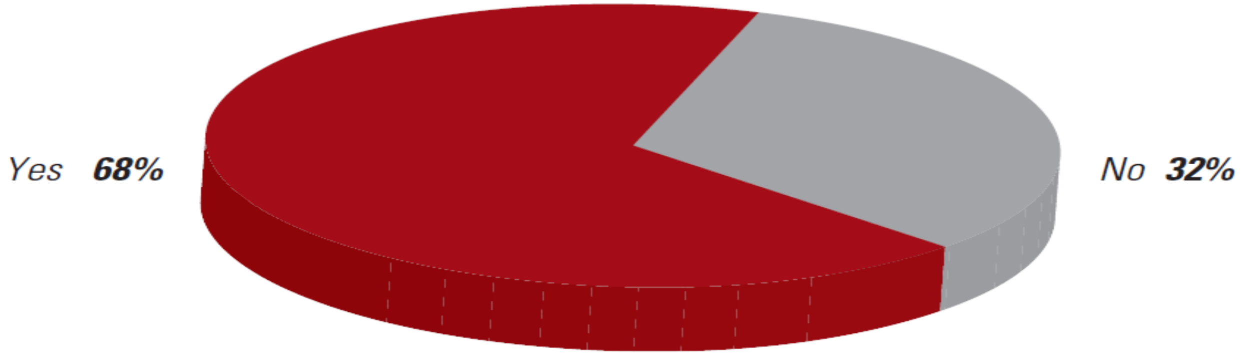


Figure 9: Are the DBAs that are responsible for managing relational database management systems also responsible for managing non-relational systems (such as NoSQL and Hadoop)?

(Only respondents who had deployed Hadoop or NoSQL technology)



Key DBA Challenges

- A key challenge for DBAs is learning **new technologies**.
- **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.

    CAGR for Data: 125%

    DBA growth: 3-5% per year
  - More data types are being captured, stored, and made available for analysis. More external data sources, too.
- While **maintenance** and **performance** are the top responsibilities for most DBAs, **security** is an increasingly important item on their agendas.
  - Data protection is no longer an afterthought

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

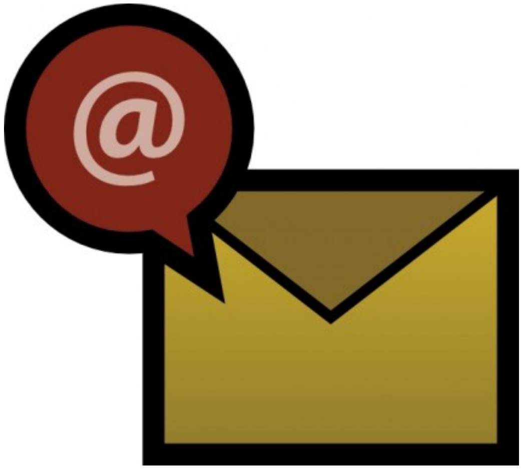
DBAs Are Tackling More Unstructured Data

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

- http://www.information-management.com/news/unstructured-data-offers-big-benefits-10025915-1.html

The most important drivers for the growth of unstructured data:

- Internally generated **documents**
- …followed by **email**.



Structured and Unstructured Data

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

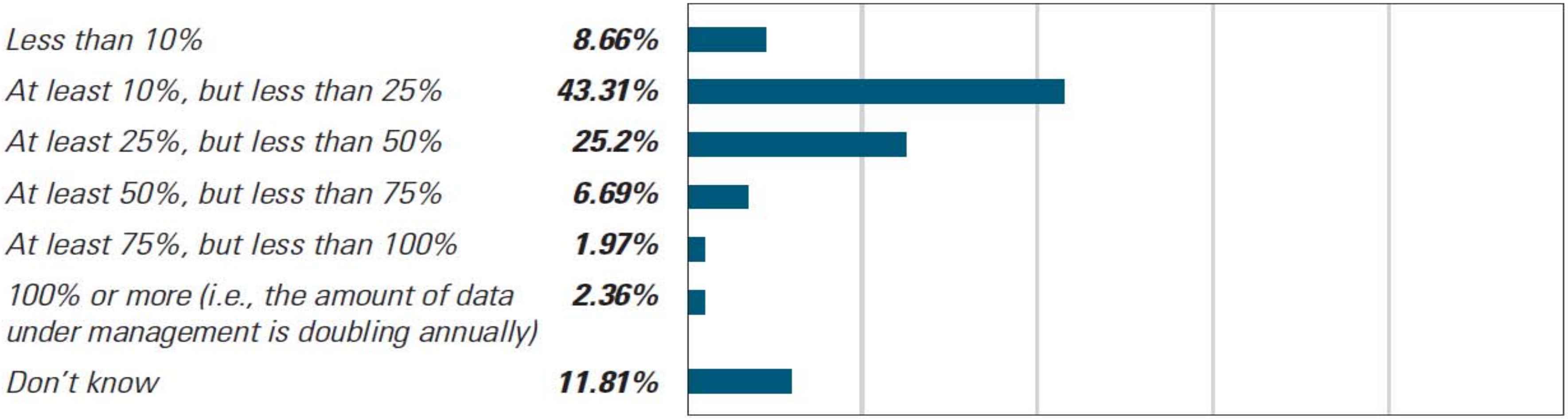
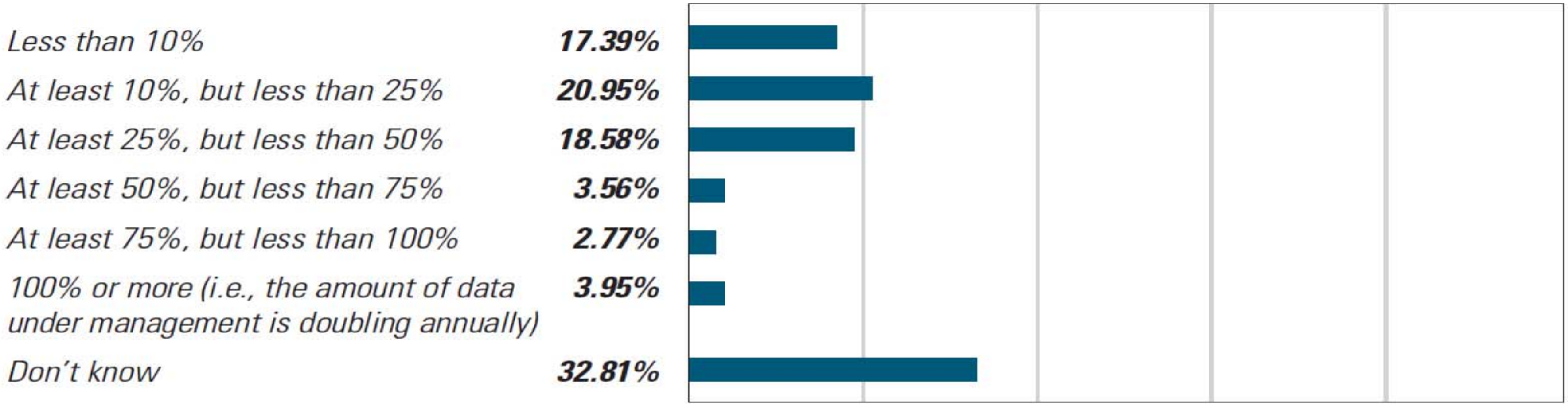
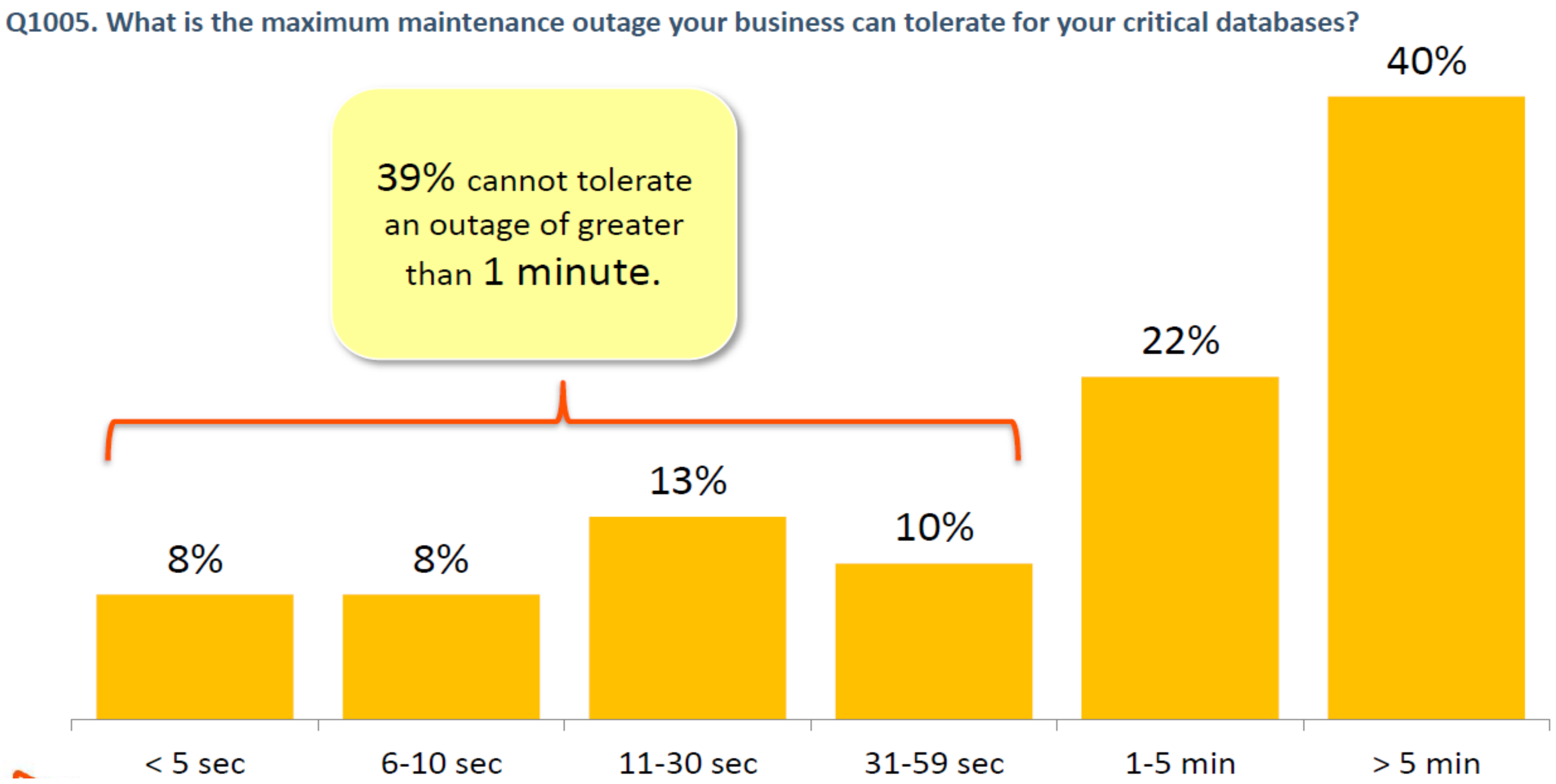


Figure 13: How fast is the amount of unstructured data growing annually?



Always Available

- And downtime is non-existent

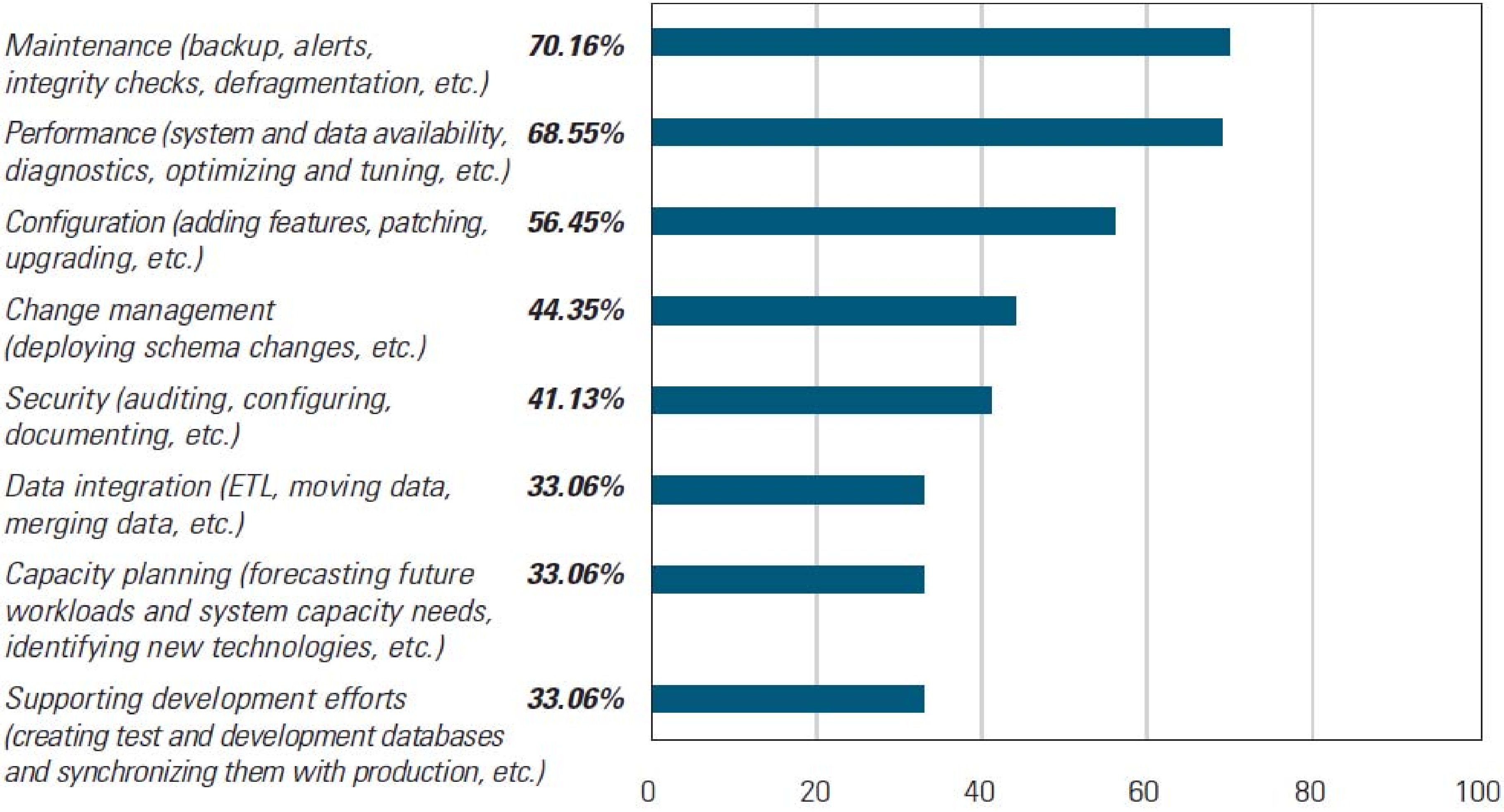


Source: 2014 Annual Mainframe Research, BMC Software, October 2014

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

Most Important DBA Duties

Figure 16: What do you consider the most important responsibilities for DBAs?



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.

Source: Wikipedia, Sept 2014, http://en.wikipedia.org/wiki/Autonomic_computing

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 adapts 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

- Many aspects to "self" managing… [next slide]

The Primary "Self" Aspects of Autonomics

Self-configuration: Automatic configuration of the system and its components;

Self-healing: Automatic discovery, and correction of faults;

Self-optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements;

Self-protection: Proactive identification and protection from arbitrary attacks.

Self-inspection: Understands itself and its interactions with other systems in order to make intelligent decisions;

Self-organization: Proactive modification of data structures and organization to optimize access.

The Promise of Autonomics

We have been promised a light-out self-managing environment for years

- That goal is still somewhere in the distant future, if at all
  - Consider backup/recovery
- But advances are being made… to performance management and utility management

Ignore autonomic systems and database administration at your peril

- Can help to alleviate the burden put on DBA staffs
  - …if your management cares about such things

DB2 for z/OS 2015

It Ain’t Your Daddy’s DB2!

The History of DB2 for z/OS

DB2 was
announced
on
June 7, 1983

Version	GA	EoM	EoS
1.1	1985-04-02		
1.2	1986-03-07		
1.3	1987-06-26		
2.1	1988-09-23		
2.2	1989-09-22		
2.3	1991-10-25		
3	1993-12-17	1999-11-30	
4	1995-10-30	2000-12-01	
5	1997-06-27	2001-12-31	2002-12-31
6	1999-06-15	2002-06-30	2005-06-30
7	2001-03-30	2007-03-05	2008-03-30
8	2004-03-26	2009-09-08	2012-04-30
9	2007-03-06	2012-12-10	2014-06-27
10	2010-10-22	2015-07-06	2017-09-30
11	2013-10-25		

And then there is DB2 12 for z/OS

Early Support Program (ESP) announced Oct 2015 for Mar 2016

Highlights of DB2 12:

- Expanded in-memory processing for greater performance improvement and emerging use case support, such as synergy with latest z Systems™ processors, which offer significantly expanded memory capacity.
  - Up to 25% CPU improvement for query workloads and up to 10% online transaction processing (OLTP) CPU savings with larger memory and activation of memory exploitation features.
  - CPU reductions and performance improvements for certain OLTP as well as select query workloads.
  - Improved zIIP offload in support of in-transaction analytics. 100% of parallel child task processing is now zIIP eligible.
- Enhanced support for cloud and mobile workloads with scalability improvements for tables (max size increased from 16 TB to 4 PB); up to 256 trillion rows in a single DB2 table.
- Support for SQL as a Service (SQLaaS) through RESTful connectivity to your DB2 for z/OS data.

But before you go 12, be sure you are up to speed on 9, 10, and 11

If you are using DB2 the same way you did in 1995…

- It is time to modernize!

Modern DB2 Today
It Ain't Your Daddy's DB2!

More Data ≈

New and Deprecated Structures

- The move to Universal Table Spaces √

Unstructured Data in LOBs √

Dynamic SQL vs. Static SQL √

HTAP and IDAA

New SQL Capabilities and Functionality

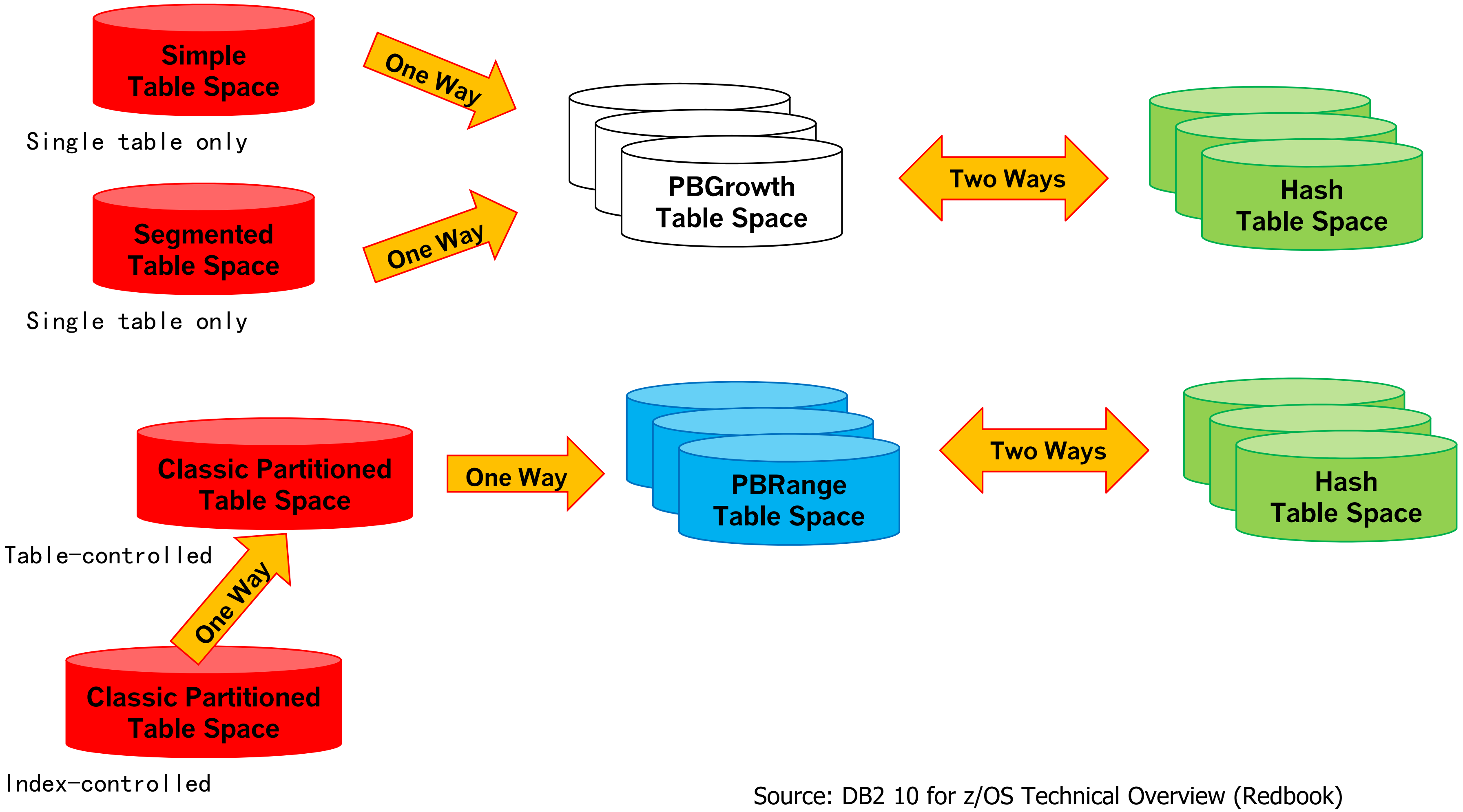
- OLAP functions
- Temporal data
- New data types including XML
- Multi-row FETCH, INSERT and UPDATE

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



Source: DB2 10 for z/OS Technical Overview (Redbook)

Three Type of Schema Changes

Simple

- **Can be implemented immediately upon request**

Medium

- **More complex; some type of versioning is required to implement**
  - Requires a REORG to fully implement the change
  - We will look at the different types of changes that fall into this category

Online

Complex

- **Requires DROP and re-CREATE process**
  - These are the un-supported changes
  - Require UNLOAD, DROP, CREATE, RELOAD

Offline

REORGs to Clear up Deferred ALTERs

- **Deferred ALTER –** syntax checked and approved for implementation, but actual changes are delayed until later
  - Introduced in DB2 10 for z/OS
  - Many more “online” changes became possible
  - But the change is pending in the DB2 Catalog
  - Requires understanding and management
- **Changes take effect with the next REORG**
- **REORG can be at Table Space or Index level**
  - REORG TABLESPACE also implements any Index Changes
- **Recovery can not be performed to a point before the changes were materialized.**
  - So take care…

Deferred ALTER Considerations

- **Remember that Universal table spaces are required for deferred ALTER**
  - Except for converting to UTS
- **DB2 does not permit combining deferred and immediate ALTERs in single SQL statement**
- **Additionally, most immediate ALTERS are not possible while changes are pending**
  - Both of the above will throw a -20385 SQLCODE
- **To avoid confusion materialize pending changes before making new changes whenever possible**
- **And clear up AREO* ASAP**

Another “Opportunity”?

When the change from non-universal to universal is materialized, (via REORG), DB2 will invalidate packages that depend on the table space.

That means you either have to REBIND these packages explicitly or they will be automatically rebound when the programs are next executed…

Large Objects (LOBs): BLOBs, CLOBs, and DBCLOBs

Driving Factors for LOB Usage in DB2

- Multimedia Data
  - Mostly images, but also audio and video
- Large Text Documents
  - Contracts, agreements, medical records, etc.
  - International: Double-byte character support
- Big Data
  - Unstructured data (text and multimedia)
    - e.g. JSON support

書

ASSIGNMENT OF REAL ESTATE CONTRACT

eceived, which is acknowledged, ______ (assignor name) (the "Assignor")

all interest and benefit in an agreement of purchase and sale of ______ (the "Vendor")

ndor on ______ (date) ______ (the "Assignee")

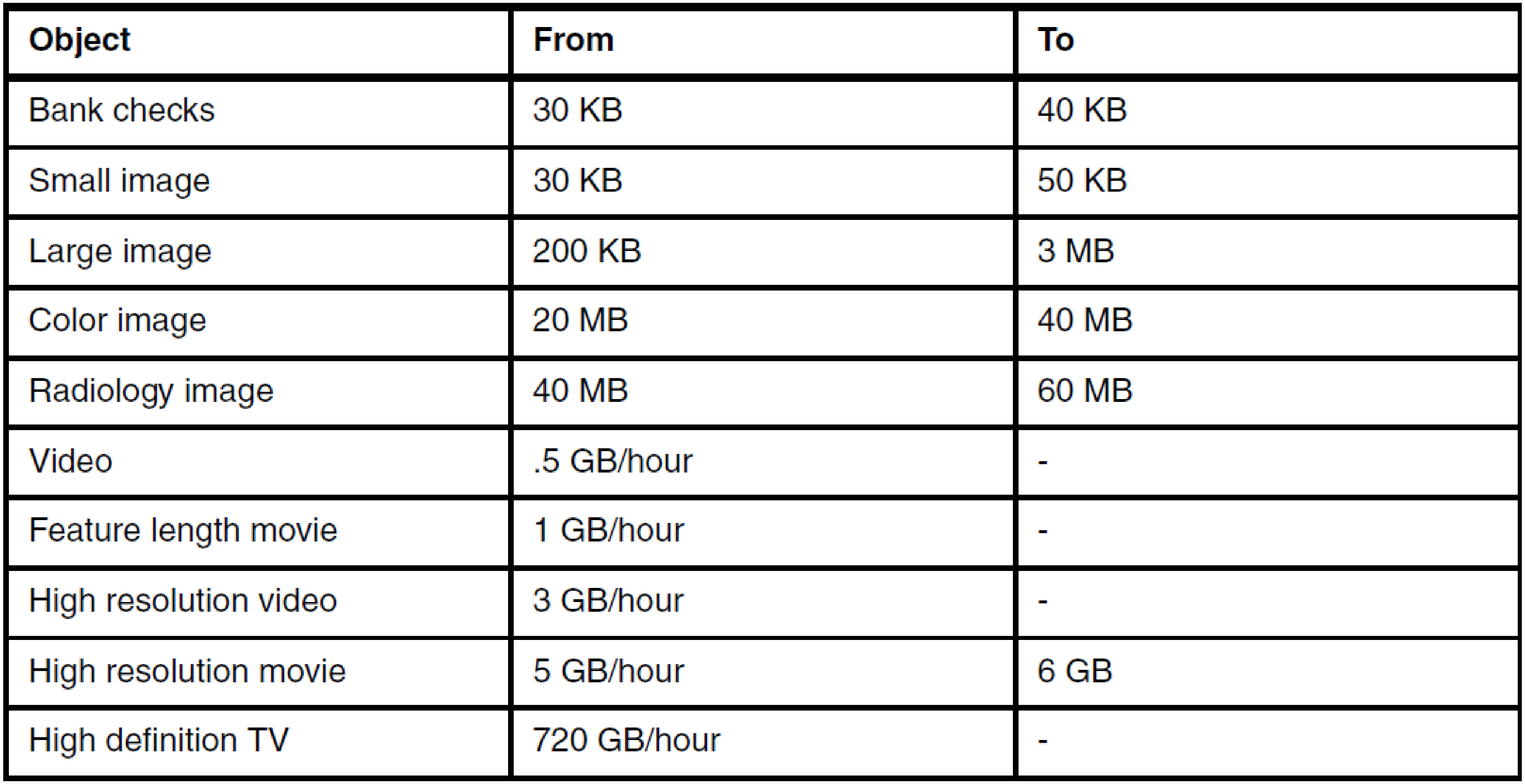
A Little History…

LOBs were introduced in Version 6 of DB2 for z/OS

Various levels of functionality and usefulness between V6 and V10

- DB2 Extenders to support multimedia and XML (Versions 7 and 8)
- SQL stored procedures and LOB variables (V8)
- Automatic creation of LOB objects (V9)
- File reference variables in LOAD and UNLOAD (V9)
- LOB lock avoidance (V9)
- FETCH continue (V9)
- Inline LOBs (V10)

Typical Average Size for Large Objects



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

Source: LOBs with DB2 for z/OS: Stronger and Faster
IBM RedBook (SG24-7270)

LOB Usage is Growing

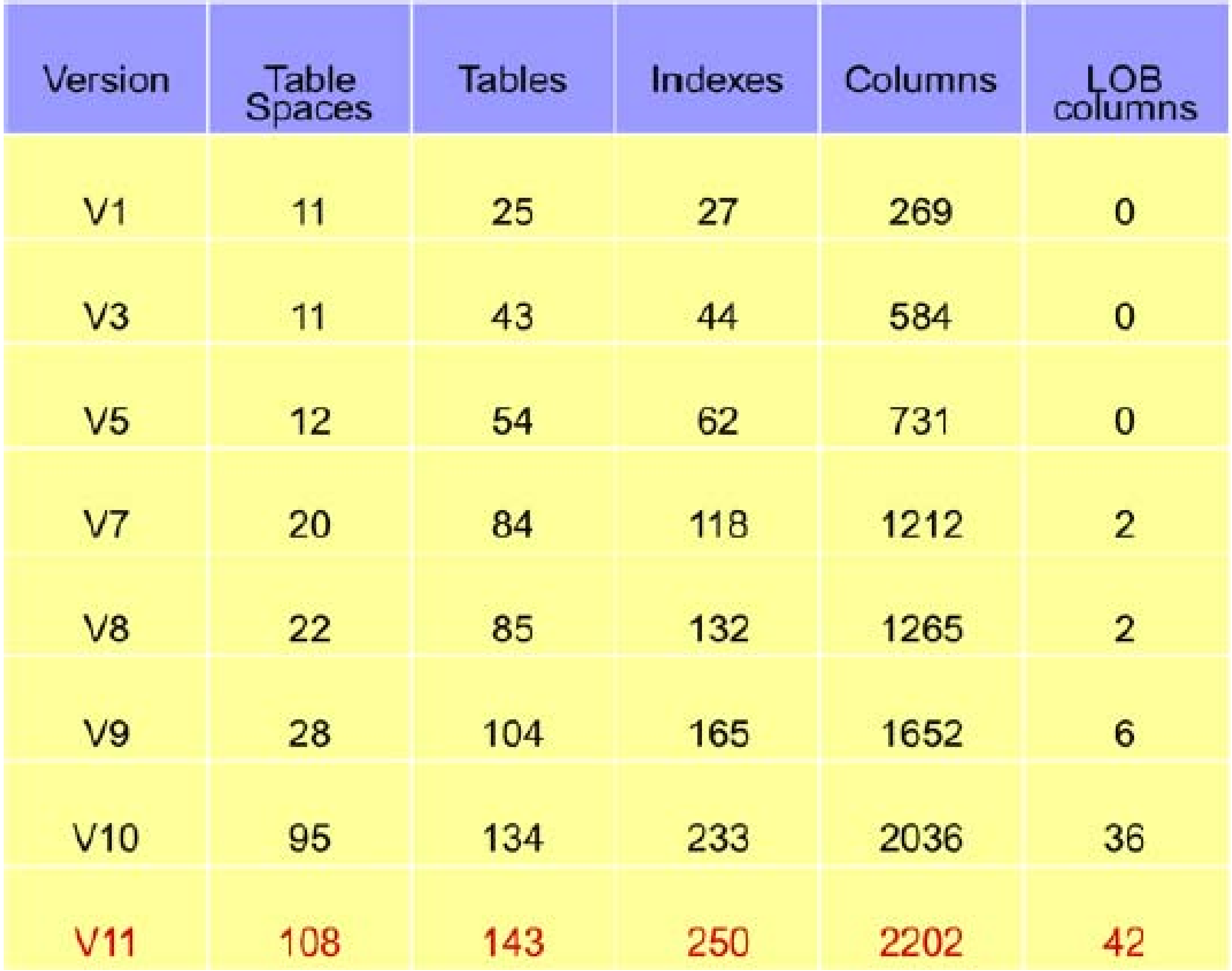
Better supported than in the past

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



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

Using LOBs Requires Additional Care & Feeding

Issues with LOBs

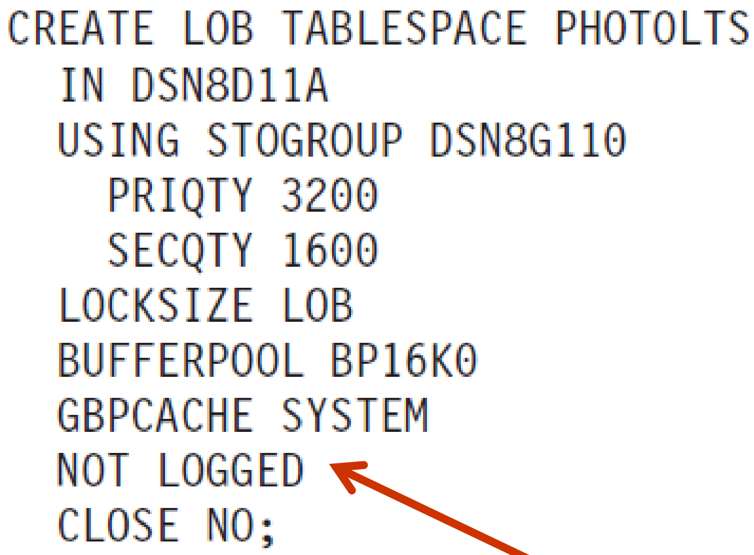
- **LOB Logging**
- **SQL Restrictions**
- **LOB Peculiarities**
- **Managing LOB Inconsistencies**

LOB Logging Considerations

Do you want to log changes to LOB columns?

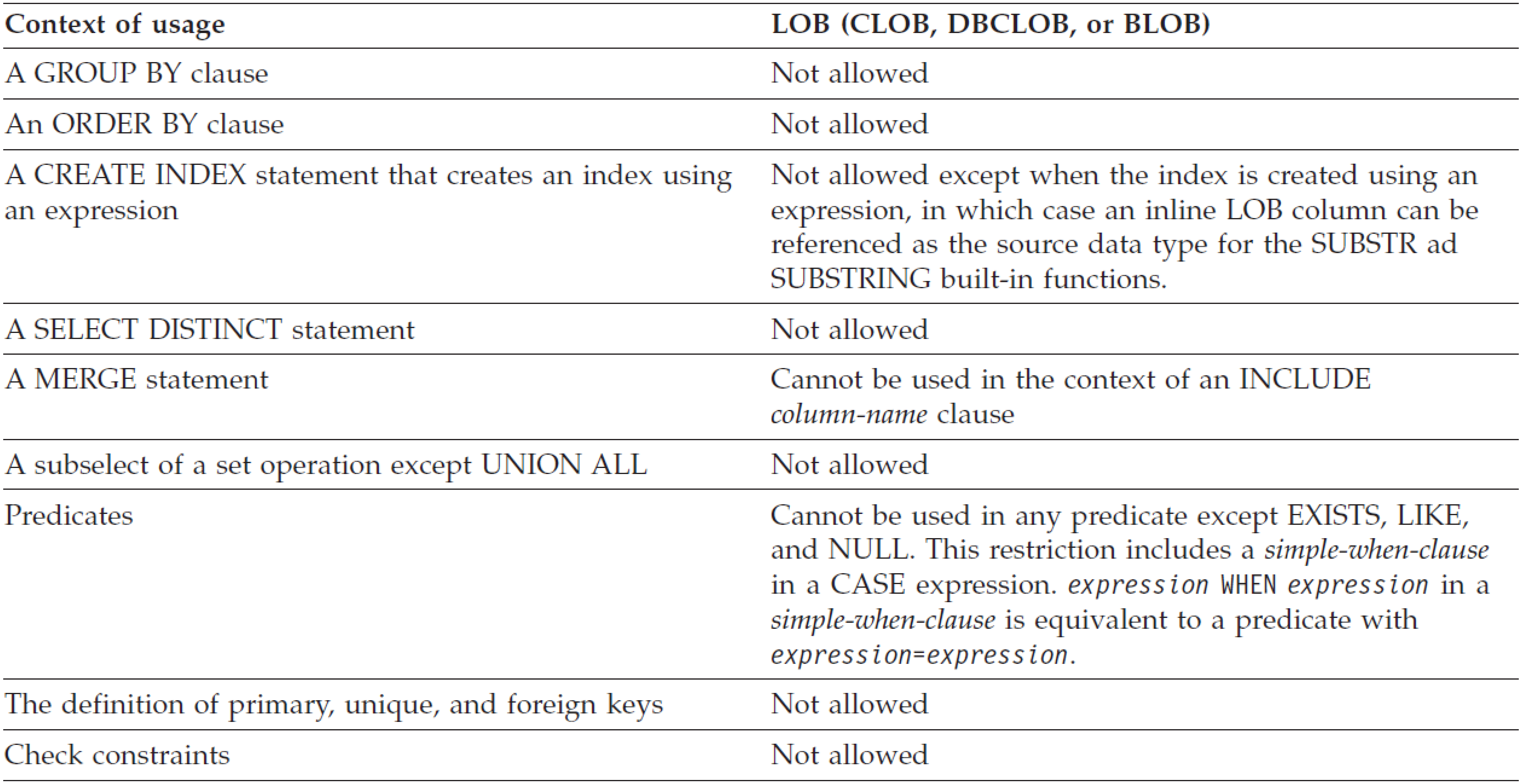
- Probably not, because they are so large

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



Context of usage	LOB (CLOB, DBCLOB, or BLOB)
A GROUP BY clause	Not allowed
An ORDER BY clause	Not allowed
A CREATE INDEX statement that creates an index using an expression	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.
A SELECT DISTINCT statement	Not allowed
A MERGE statement	Cannot be used in the context of an INCLUDE *column-name* clause
A subselect of a set operation except UNION ALL	Not allowed
Predicates	Cannot be used in any predicate except EXISTS, LIKE, and NULL. This restriction includes a *simple-when-clause* in a CASE expression. *expression* `WHEN` *expression* in a *simple-when-clause* is equivalent to a predicate with *expression*`=`*expression*.
The definition of primary, unique, and foreign keys	Not allowed
Check constraints	Not allowed

Peculiarities to Keep in Mind

You do not specify column names for an Auxiliary Table

- You identify the LOB column from your base table and DB2 automatically generates the columns needed

You do not specify any column names for an Auxiliary Index

- DB2 implicitly generates the needed index keys

Dropping a LOB column from a base table does not automatically clean up LOB table spaces.

- You can either drop them yourself explicitly, or reuse them for another LOB.

LOB columns are not really updated

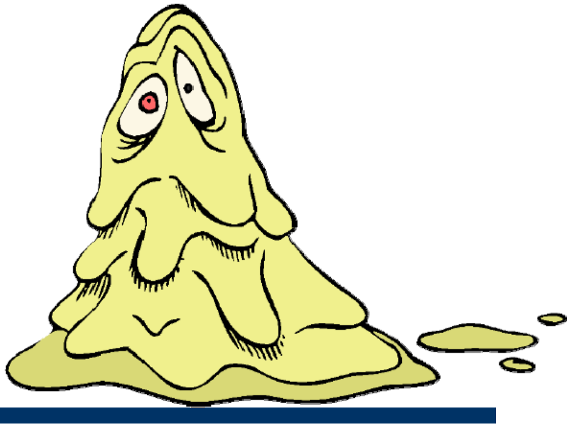
- Old version of the LOB is de-allocated, new is allocated

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.

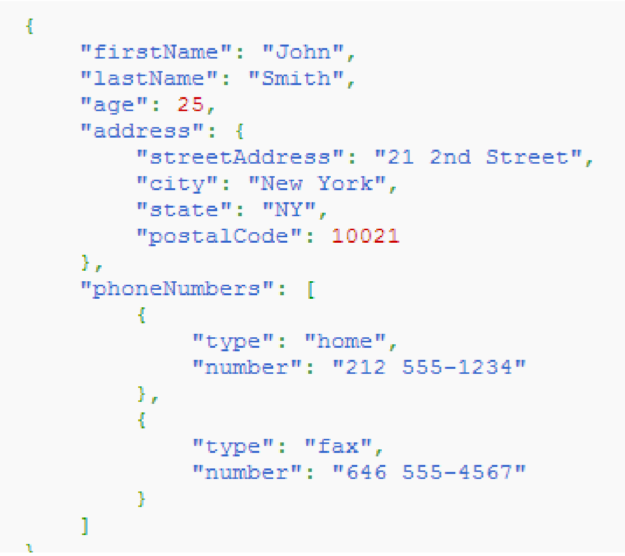
LOB data in the LOB TS, by virtue of its size, can be distributed over many different pages of the LOB TS. DB2 uses a structure of "MAP" pages to point to these data pages, which do not need to be contiguous. If all the data pages are not referenced by these "MAP" pages or if the "MAP" pages themselves are not properly referenced by a higher level "MAP" page, **LOB data will be lost**.

LOBs and Big Data

JSON data can be stored in a DB2 BLOB

What is JSON?

- An open standard format for data interchange that uses human-readable text to transmit data objects consisting of key–value pairs.
- Language-independent but uses conventions that are familiar to programmers of the C-family of languages
  - JSON has no tags – not self-descriptive
- JSON is built on two structures:
  - A collection of name/value pairs
  - An ordered list of values
- It is used primarily to transmit data between a server and web application.

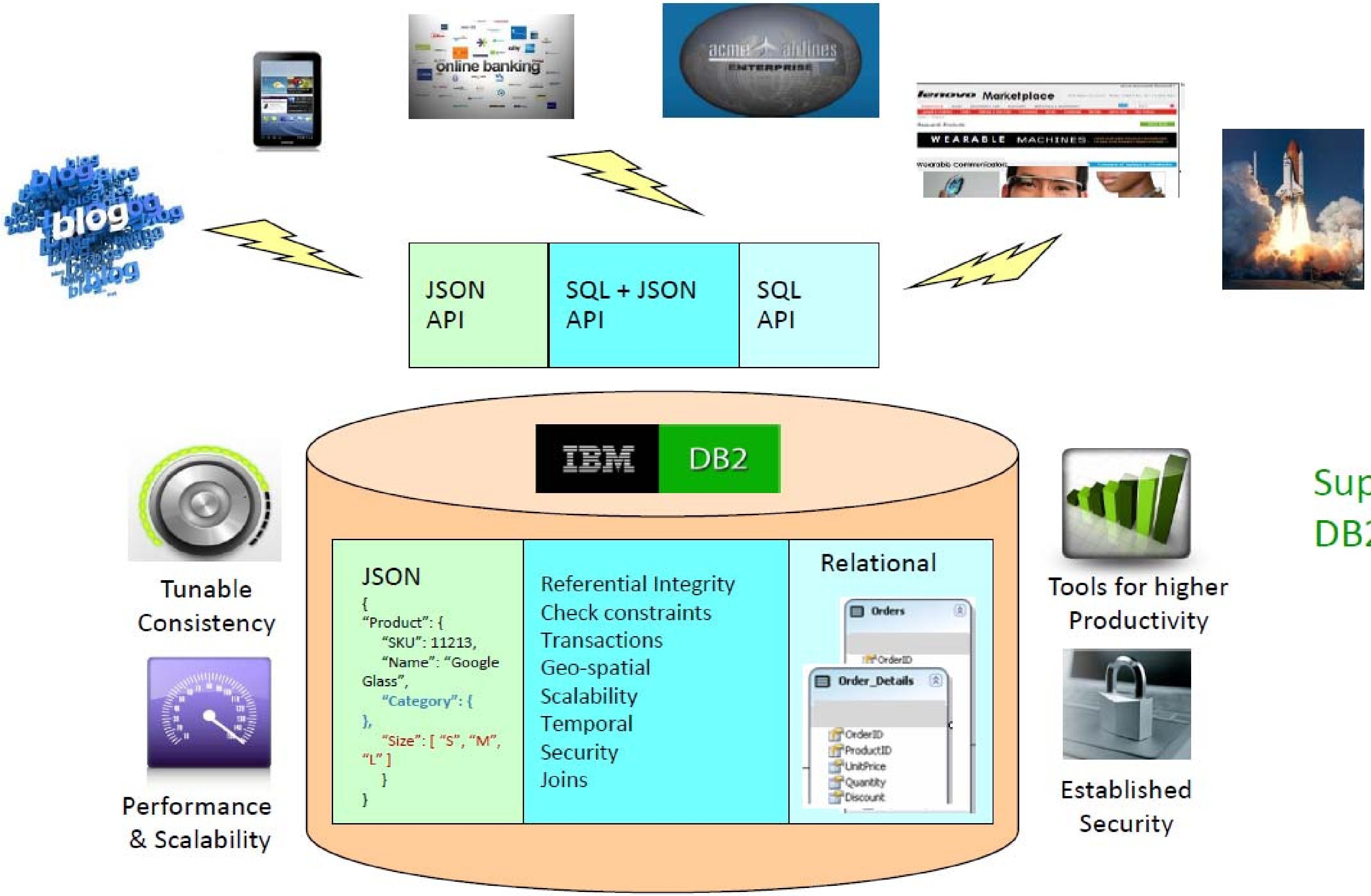


```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "fax",
            "number": "646 555-4567"
        }
    ]
}
```

JSON and DB2

http://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson1/

http://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson2/



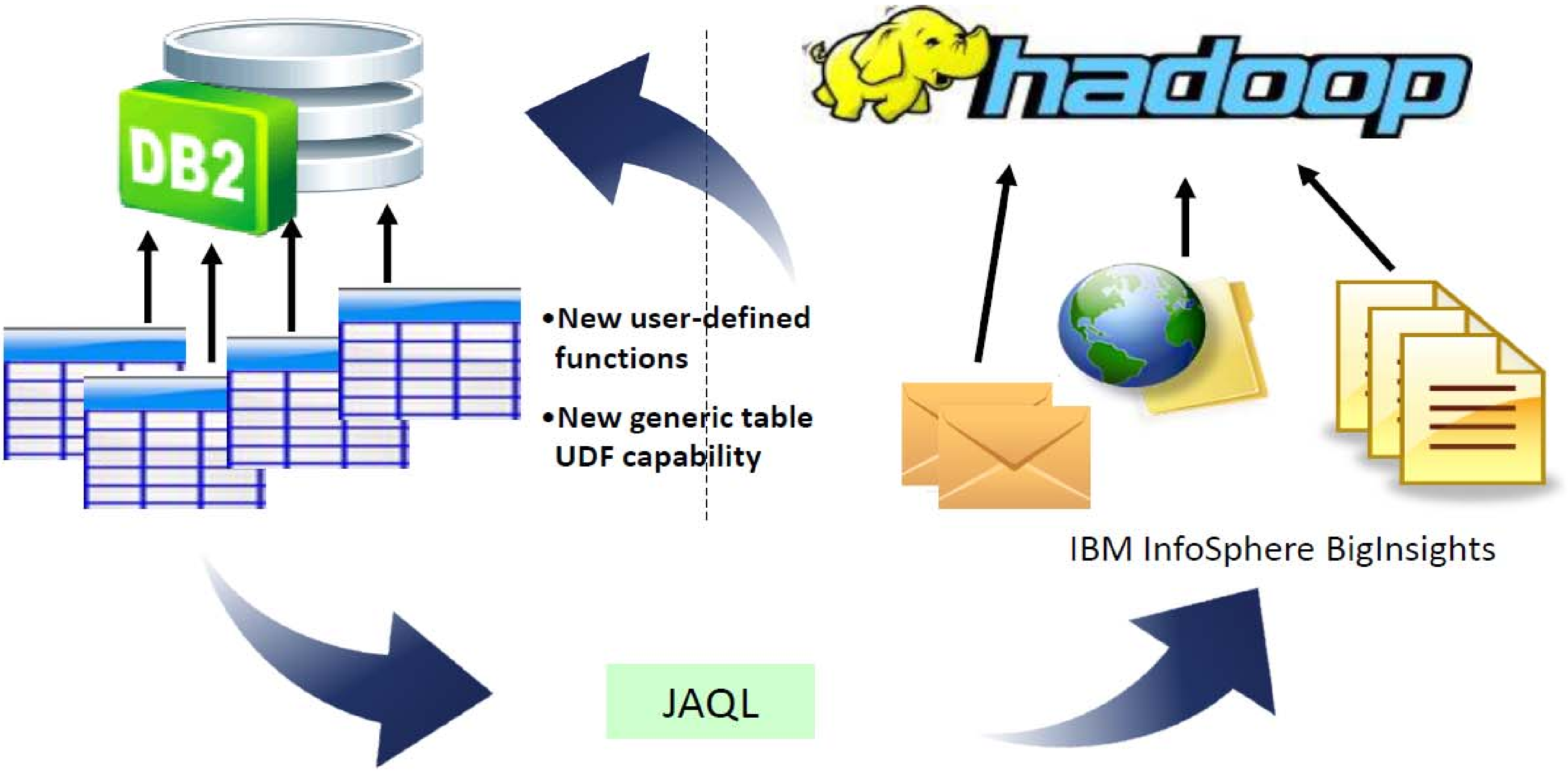
```
{
"Product": {
    "SKU": 11213,
    "Name": "Google
Glass",
    "Category": {
},
    "Size": [ "S", "M",
"L" ]
    }
}
```

Supported in DB2 10 and 11

Source: DB2 11 for z/OS: Technical overview - Part 2
IBM presentation by John Campbell, Feb 19, 2014

DB2 Connectors to Hadoop

DB2 is providing the connectors and the DB capability to allow DB2 applications to access data easily and efficiently in Hadoop



- New user-defined functions
- New generic table UDF capability

Source: DB2 11 for z/OS: Technical overview - Part 2
IBM presentation by John Campbell, Feb 19, 2014

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

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

Dynamic SQL Pains

1. More difficult to manage because the access paths are not readily accessible in plan tables (as with static SQL).
2. Dynamic SQL may be running that DBA groups are not aware of… a black hole!
3. Advice and Considerations:
   - Build steps into your development process & production turnover procedures to ensure efficient & effective SQL
   - Be prepared to support & maintain applications using dynamic SQL with tools and processes that can identify and optimize dynamic SQL statements
   - Take advantage of all that DB2 has to offer for managing, optimizing and controlling dynamic SQL applications… for example

     Use Dynamic Statement Caching

     -START TRACE(P) CLASS(30) IFCID(316,317,318)

     Protect: use TRUSTED CONTEXT as appropriate

Guidance

Things to Consider and Plan for

My Advice to DB2 DBAs

Treat DBA as a Management Discipline

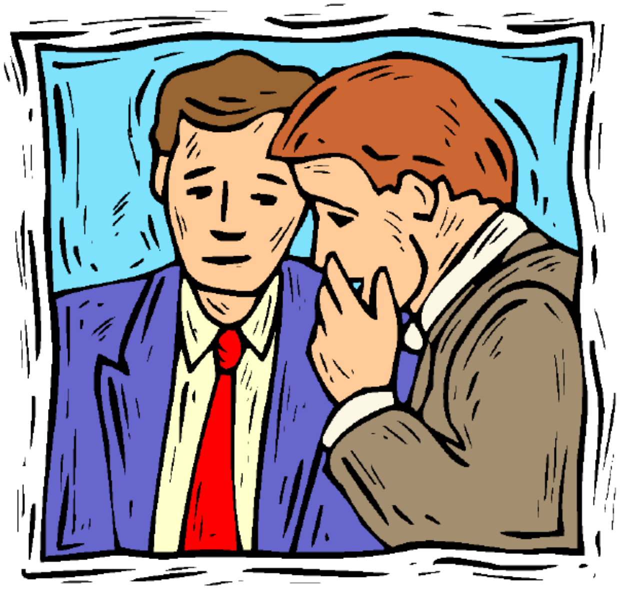
- Proactive vs. Reactive

Automate what you can

- Turn tasks over to the computer to free up DBA time
- Intelligent automation & autonomics

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



Contact Information

Craig S. Mullins

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

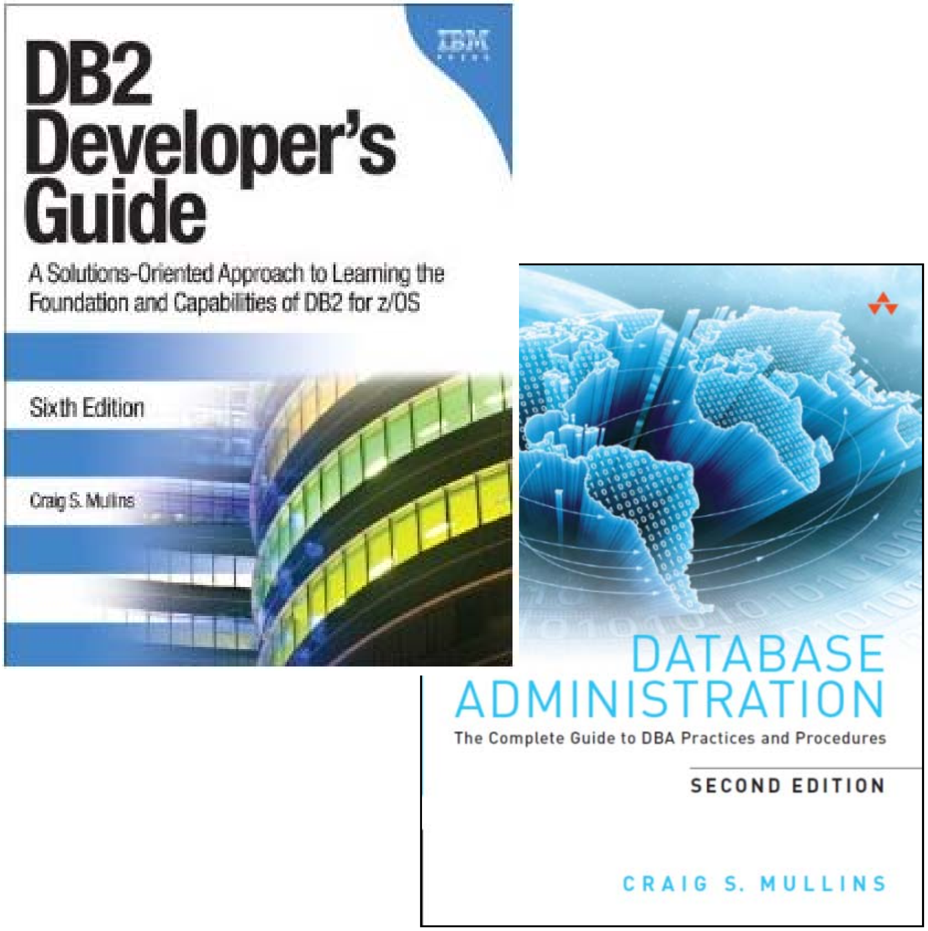
E-mail: craig@craigsmullins.com

Web: www.mullinsconsulting.com

IBM CHAMPION

IBM Gold CONSULTANT

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



IBM

DB2 Developer's Guide

A Solutions-Oriented Approach to Learning the Foundation and Capabilities of DB2 for z/OS

Sixth Edition

Craig S. Mullins

DATABASE ADMINISTRATION

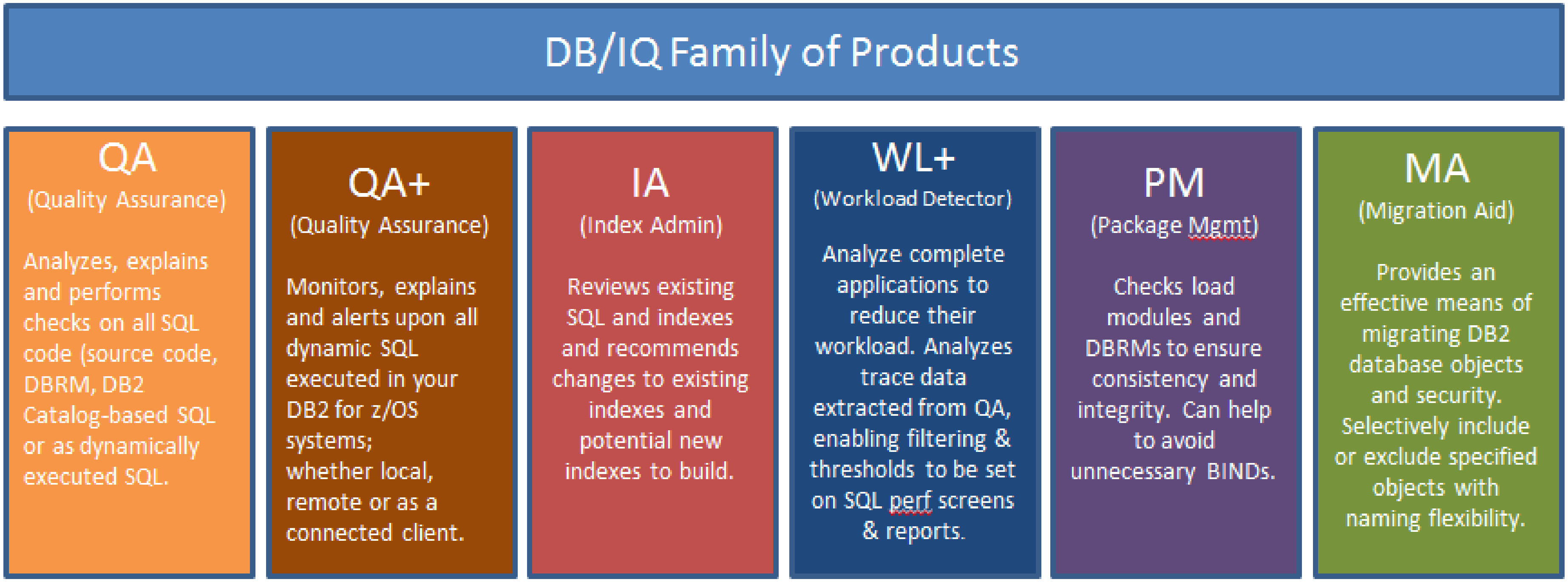
The Complete Guide to DBA Practices and Procedures

SECOND EDITION

CRAIG S. MULLINS

http://mullinsconsulting.com/dba_book.htm

DataKinetics: DB/IQ Suite of DB2 Products



DB/IQ Family of Products

QA
(Quality Assurance)

Analyzes, explains and performs checks on all SQL code (source code, DBRM, DB2 Catalog-based SQL or as dynamically executed SQL.

QA+
(Quality Assurance)

Monitors, explains and alerts upon all dynamic SQL executed in your DB2 for z/OS systems; whether local, remote or as a connected client.

IA
(Index Admin)

Reviews existing SQL and indexes and recommends changes to existing indexes and potential new indexes to build.

WL+
(Workload Detector)

Analyze complete applications to reduce their workload. Analyzes trace data extracted from QA, enabling filtering & thresholds to be set on SQL perf screens & reports.

PM
(Package Mgmt)

Checks load modules and DBRMs to ensure consistency and integrity. Can help to avoid unnecessary BINDs.

MA
(Migration Aid)

Provides an effective means of migrating DB2 database objects and security. Selectively include or exclude specified objects with naming flexibility.