IBM Software Group

DB2 UDB
Viper Technology Preview

Worldwide Information Management Technical Presales

ON DEMAND BUSINESS

Disclaimer/Trademarks

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataJoiner, DataPropagator, DB2, DB2 Connect, DB2 Extenders, DB2 OLAP Server, DB2 Universal Database, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, Net.Data, OS/390, OS/400, PowerPC, pSeries, RS/6000, SQL/400, SQL/DS, Tivoli, VisualAge, VM/ESA, VSE/ESA, WebSphere, z/OS, zSeries

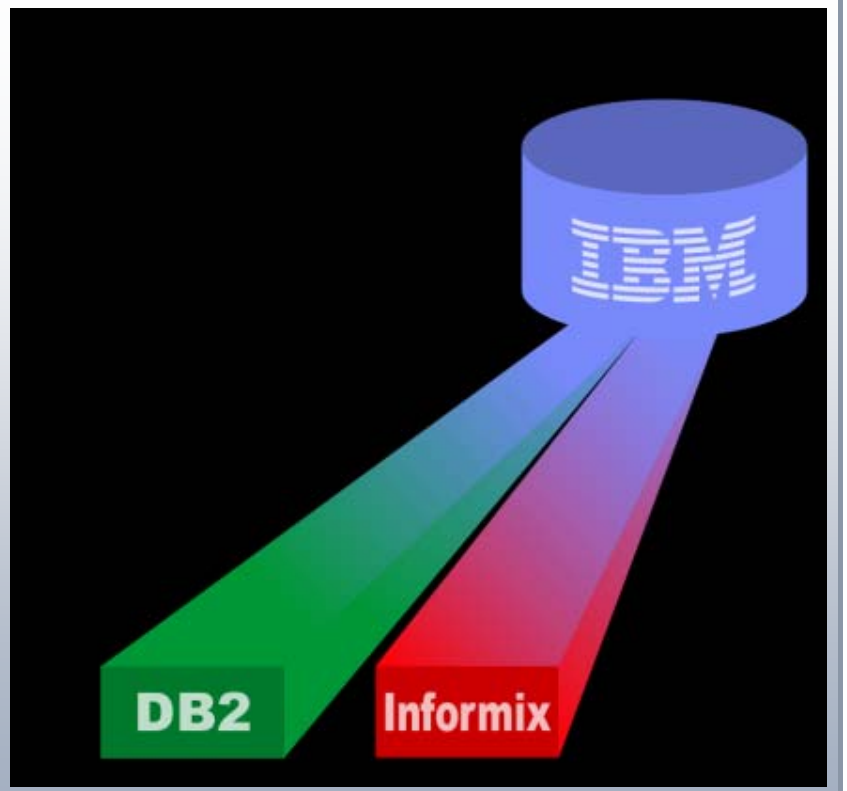
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

Agenda

- DB2 Viper Strategy and Key Investment Areas
- XML Support
- Reducing the Total Cost of Ownership
- Expanding Database Capacity and Removing Limits
- Granular Security
- Table Compression

IBM Database Technology Strategy

- Continued Focus on Performance, Scale, Availability
- Reduce TCO and Accelerate Time-to-value
- Support for New Data Types
- Deep Cross-middleware Integration



Commonality Across DB Servers

IBM Software Group

XML Technology

Integrating XML into the DB2 Engine

ON DEMAND BUSINESS

What is XML?

- XML Technology
  - XML = Extensible Markup Language
  - Self-describing data structures
  - XML Tags describe each element and their attributes
- Benefits
  - Extensible
    - No fixed format or syntax
    - Structures can be easily changed
  - Platform Independent
    - Not tied to any platform, operating system, language or software vendor
    - XML can be easily exchanged
  - Fully Unicode compliant

```
<? xml version="1.0" ?>
<purchaseOrder id='12345" secretKey='4x%$^'>
    <customer id="A6789">
        <name>John Smith Co</name>
        <address>
            <street>1234 W. Main St</street>
            <city>Toledo</city>
            <state>OH</state>
            <zip>95141</zip>
        </address>
    </customer>
    <itemList>
        <item>
            <partNo>A54</partNo>
            <quantity>12</quantity>
        </item>
        <item>
            <partNo>985</partno>
            <quantity>1</quantity>
        </item>
    </itemList>
</purchaseOrder>
```

XML Example: Financial Data (FIXML)

- Buying 1000 Shares of IBM Stock..

```
8=FIX.4.2^9=251^35=D^49=AFUNDMGR^56=ABROKER^34=2
^52=20030615-01:14:49^11=12345^1=111111^63=0^64=2003
0621^21=3^110=1000^111=50000^55=IBM^48=459200101^22=
1^54=1^60=2003061501:14:4938=5000^40=1^44=15.75^15=USD
^59=0^10=127
```

Old FIX Protocol

New FIXML Protocol

- extensible
- lower appl development & maintenance cost

```
<FIXML>
  <NewOrdSingle ClOrdID="123456"
    Side="2"
    TransactTm="2003-06-15T01:14:49-05:00"
    OrderType="2"
    Price="93.25"
    Acct="26522154">
    <Header Sent="2001-06-21T01:31:28-05:00"
      PosDup="N"
      PosRsnd="N"
      SeqNum="521">
      <Sender ID="AFUNDMGR"/>
      <Target ID="ABROKER"/>
    </Header>
    <Instrument Symbol="IBM"
        ID="459200101"
        IDSrc="1"/>
    <OrderQuantity Qty="1000" Cur="USD"/>
  </NewOrdSingle>
</FIXML>
```

XML Market Projections

- **XML Storage is a high growth area**

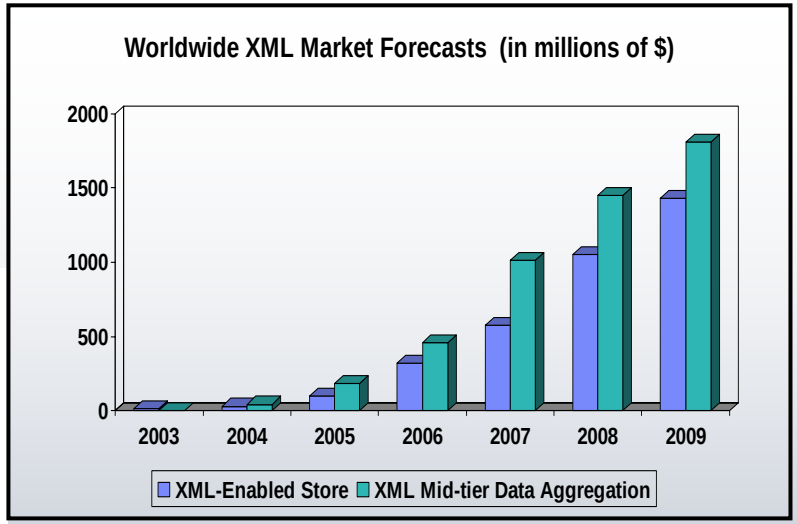
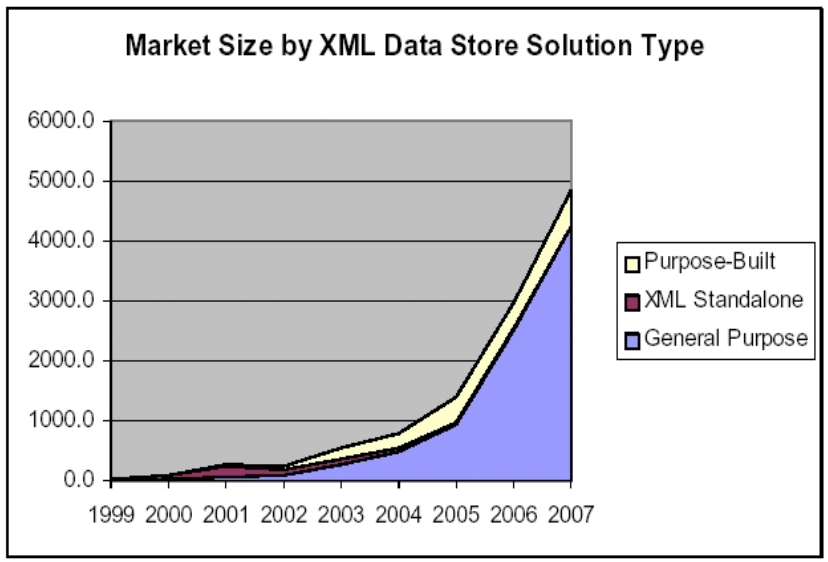


Figure VI.2: Market Size by XML Data Store Solution Type



- **XML database revenue to grow at twice the rate of the total database market**

- IDC

Worldwide Enterprise Database Management Systems Software Forecast Update, 2003-2007

Chart Sources: XML Market Opportunities, Forecasts and Strategies, 2004-2009 Wintergreen Research Inc. ZapThink

XML – A Very Different Data Model

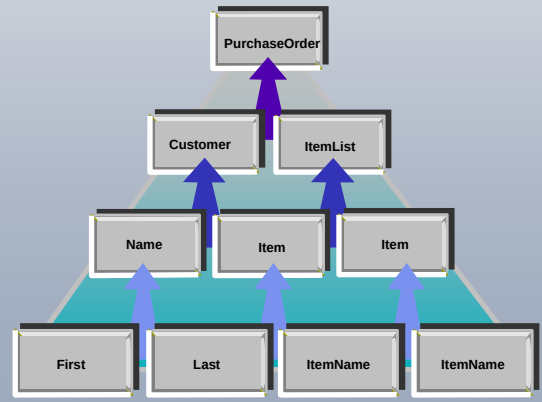
- Relational is a data model:
  - Relations (tables)
  - Attributes (columns)
  - Set based w/ some sequences
  - Strict schema

POID	CustomerID	ItemID
12	1	2
162	3	4
162	3	5

Id	LastName	FirstName	Street	City	State	Zip
1	Pirahesh	Hamid	1 Harry Rd	San Jose	CA	95141
3	Selinger	Pat	555 Bailey Ave	San Jose	CA	95141

ItemID	Name
2	#6 wire nut
5	Small Walrus
4	Apollo moon rocket

- XML is a data model:
  - Nodes (elements, attributes, comments, etc.)
  - Relationships between nodes
  - Sequence based w/ some sets
  - Flexible schema

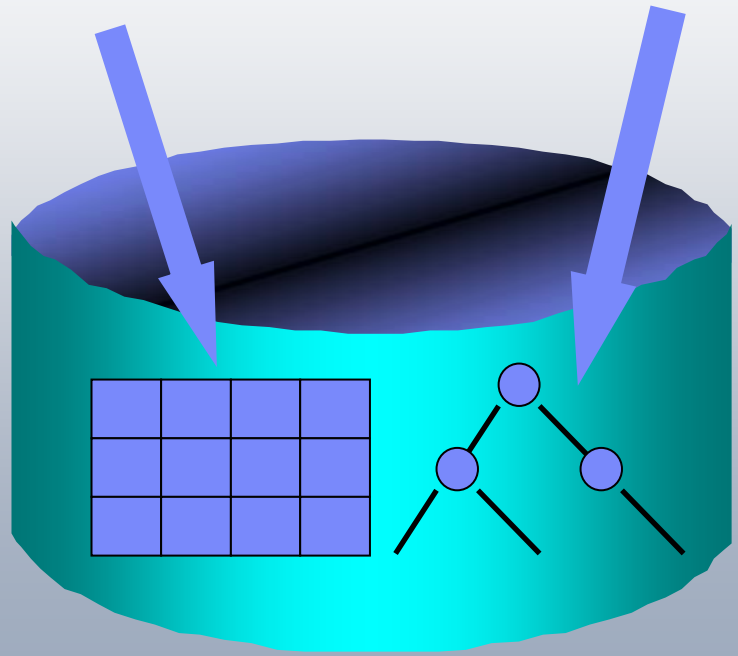


Native XML Storage

- Must store XML in parsed hierarchical format
  (similar to the DOM representation of the XML infoset)

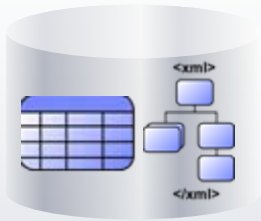
```
create table dept (deptID char(8),…, deptdoc xml);
```

- Relational columns are stored in relational format (tables)
- XML columns are stored natively
- XML stored in UTF8



XML in DB2 Viper

SQL Person… "I see a world class RDBMS that also supports XML"



DB2 with XML Support

XML Person… "I see a world class XML repository that also supports SQL"

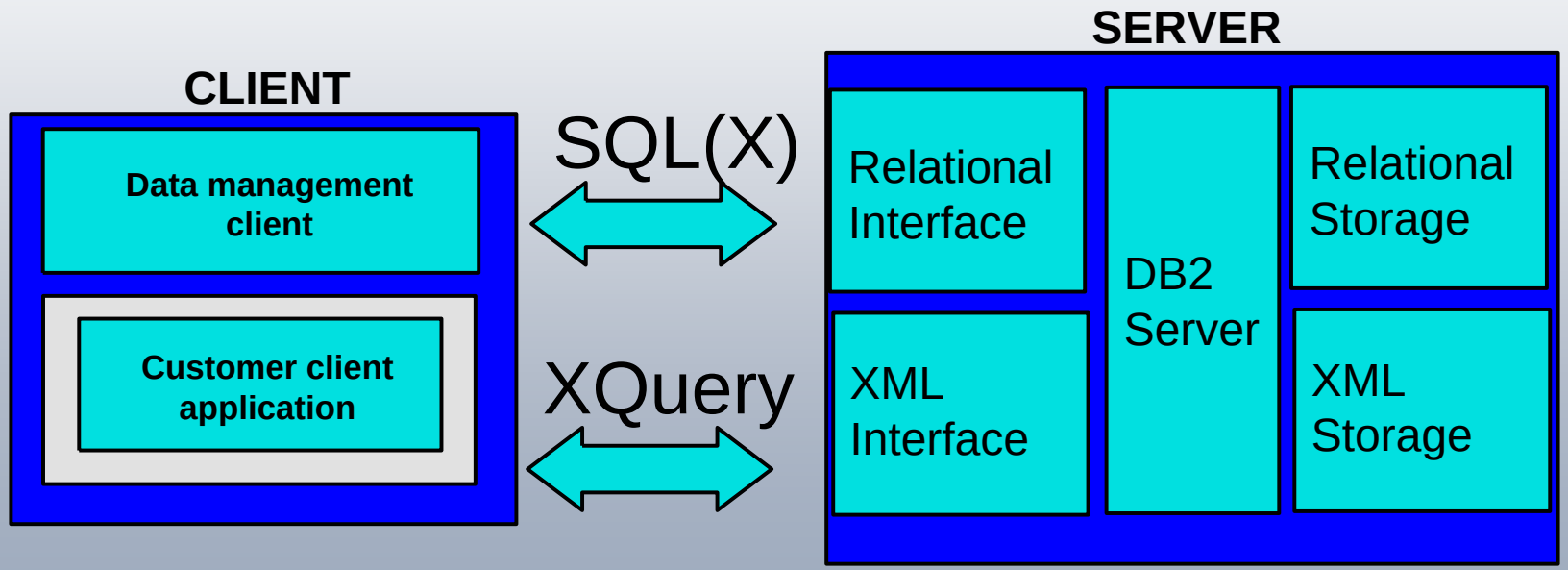
XML integrated in all facets of DB2!

New XML applications benefit from:

- Ability to seamlessly leverage relational investment
- Proven Infrastructure that provides enterprise-class capabilities

XML in DB2

- "Feels" relational and/or XML
- Both SQL flavor and fully XML flavor
- XML *is* DB2 internals - XML Extender becomes one with the data engine



IBM Software Group

Improved Database Maintenance

Reducing the Total Cost of Ownership

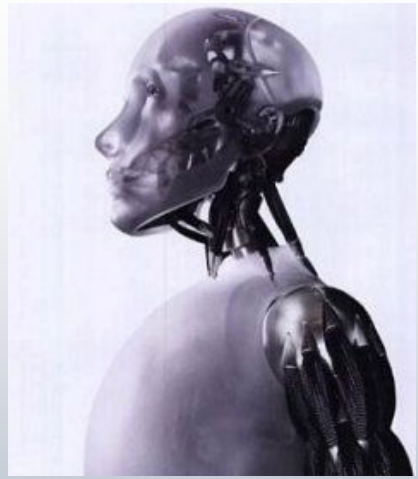
ON DEMAND BUSINESS

Installation Improvements

- Reduce installation complexity
  - Multiple DB2 versions and fixpacks on the same Windows system
- Multiple instances for maintenance
  - On Windows, Linux, and UNIX
- Uninstall
  - Allow full uninstall on Windows

Automation Automatically!

- Enable many of the DB2 autonomic computing features by default.
- Examples:
  - Configuration Advisor (2 second tuning)
  - Adaptive Self Tuning Memory
  - Automatic data statistics collection.
- Better defaults for I/O Cleaners and I/O servers
  - Default for NUM_IOSERVERs (3) and NUM_IOCLEANERS (1) set to AUTOMATIC
  - Values calculated at database startup time
    IOCLEANERS calculated based on number of CPUS and partitions
  - IOSERVERS calculated based on parallelism settings of all the tablespaces



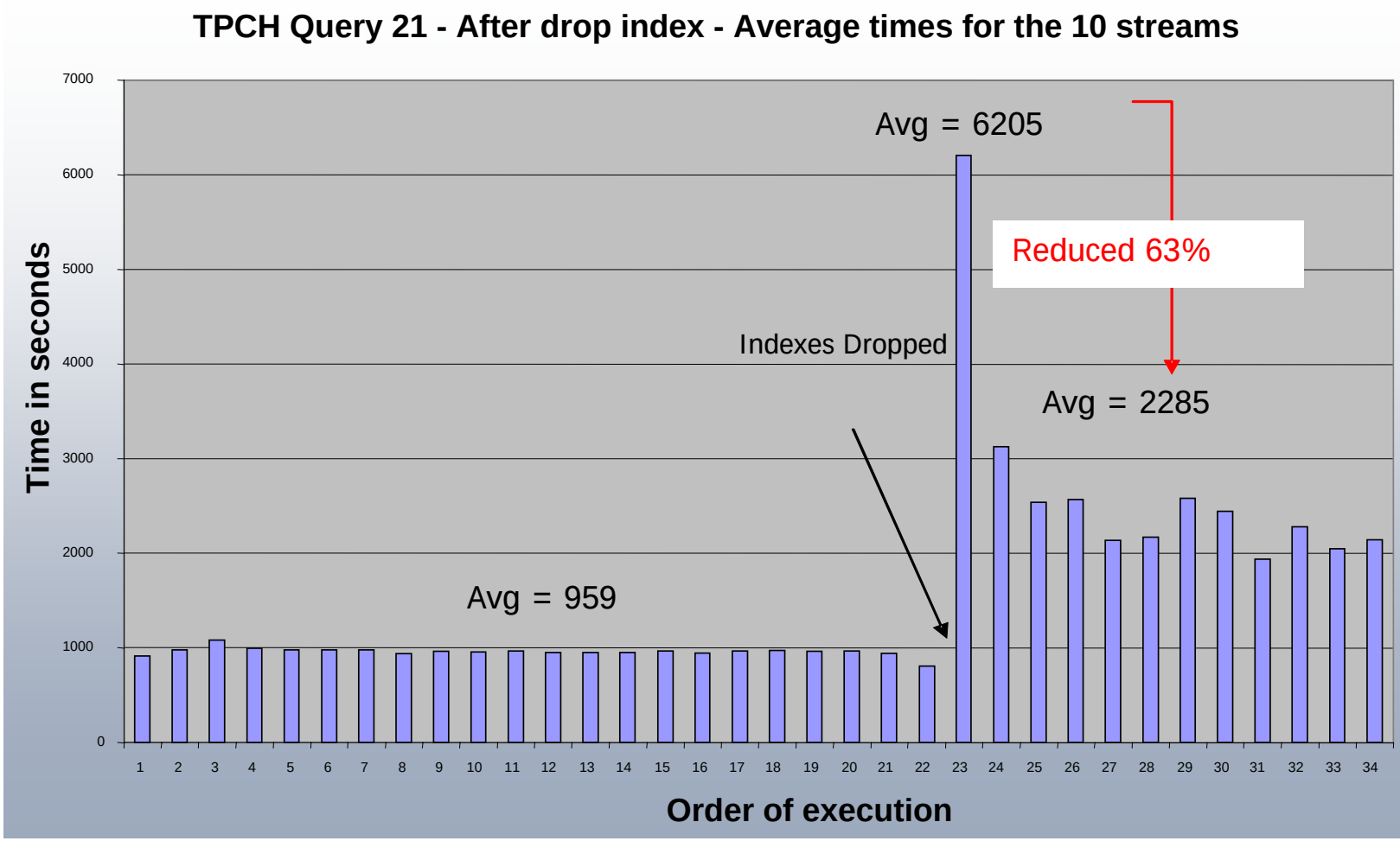
Backup and Restore

- Table function to list files in a database
  - Used to automate support of split mirror backup/recovery
- Restartable Recovery
  - Re-issuing RECOVER command will pick up where it left off
  - Ability to change Point In Time in either direction
- Rebuild partial database
  - Eliminate the need for FULL db backup
  - Ability to rebuild entire DB, including DPF, from set of table space backup images
  - Scans the history file to build the list of images to restore
- Redirected Restore Script builder
  - Build a redirected restore script from a backup image

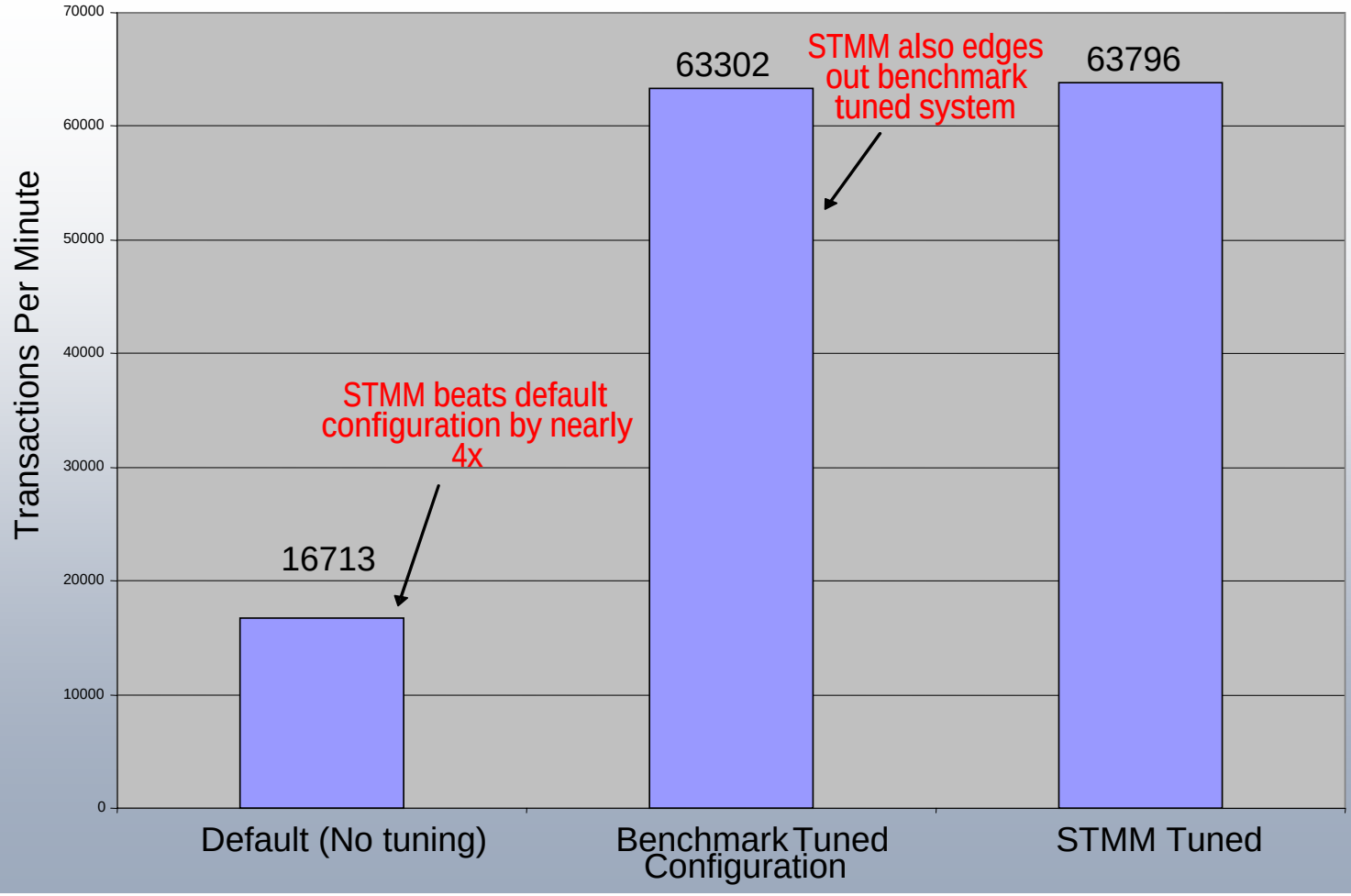
Adaptive Self-Tuning Memory

- Viper will introduce a revolutionary memory tuning system called the Self Tuning Memory Manager (STMM)
  - Works on main database memory parameters
    - Sort, locklist, package cache, buffer pools, and total database memory
  - Hands-off online memory tuning
    - Requires no DBA intervention
  - Senses the underlying workload and tunes the memory based on need
  - Can adapt quickly to workload shifts that require memory redistribution
  - Adapts tuning frequency based on workload

STMM in Action – Dropping an Important Index



STMM in Action – Comparing Different Configurations

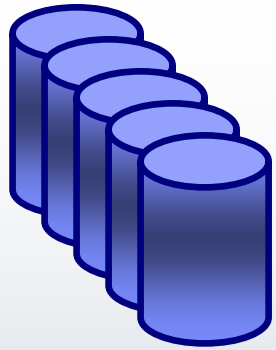


DB2 Simplified Storage Administration

- User specifies a group of storage devices for DB2, DB2 allocates and grows table consumption of storage on demand.
  - New to the “Saturn” release of DB2
  - Intended as a “single point of storage management” for table spaces
  - Create a database and associate a set of storage paths with it
- AUTOMATIC STORAGE table spaces
  - No explicit container definitions are provided
  - Containers automatically created across the storage paths
  - Growth of existing containers and addition of new ones completely managed by DB2
- Built around DMS storage model
- Add storage paths to the database afterwards
- Redefine those storage paths during a database RESTORE

Automatic Storage Provisioning - Syntax

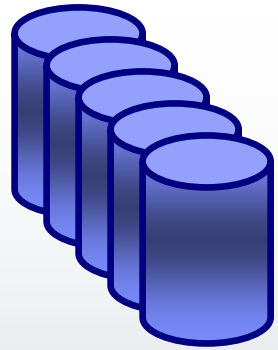
- CREATE DATABASE DB1
  AUTOMATIC STORAGE YES
- CREATE DATABASE DB3
  ON /data/path1, /data/path2
- CREATE TABLESPACE TS2
  MANAGED BY AUTOMATIC STORAGE
- CREATE TEMPORARY TABLESPACE TEMPTS
- CREATE USER TEMPORARY TABLESPACE USRTMP
  MANAGED BY AUTOMATIC STORAGE
- CREATE TABLESPACE TS1
  INITIALSIZE 500 K
  INCREASESIZE 100 K
  MAXSIZE 100 M



Automatic Storage Provisioning - Restore

- RESTORE DATABASE TEST1

- RESTORE DATABASE TEST3
  ON /path1, /path2, /path3



- If the ON clause is specified, all of the paths listed are considered storage paths, and these paths are used instead of the ones stored within the backup image.

- If the ON clause is not specified, no change is made to the storage paths (the storage paths stored within the backup image are maintained).

Availability Enhancements

- Error Toleration
  - Retry read operation
- Error Isolation
  - Log the specific problems to record the failure and provide appropriate diagnostics.
- Copy, Drop or Rename Schema
  - Within a database
  - between databases

Materialized Query Table Improvements

- Explain Un-used MQTs
  - Lists MQTs which were not used and the reason for their exclusion
- Mismatched Elements and Expression Support
  - Allow for expressions with different order of operation to be considered for MQT selection
  - C = A + B is equivalent to C = B + A
- Maintenance of NULLable MQT columns
- Efficient handling of A=B OR (A IS NULL AND B IS NULL) predicates

IBM Software Group

Expanding Database Capacity

More room for growth and less limits in the database

ON DEMAND BUSINESS™

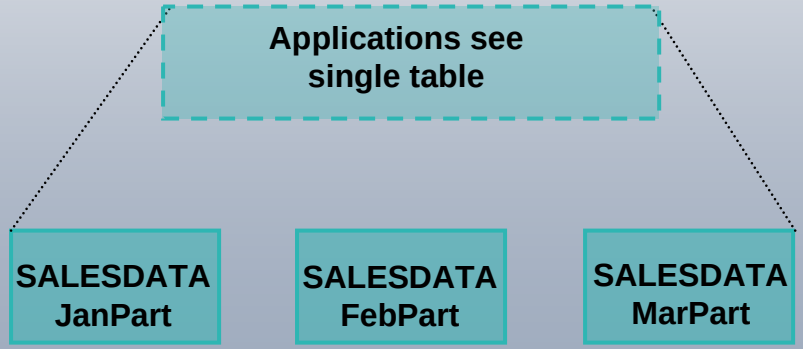
Table Partitioning

- What is Table (Range) Partitioning ?
  - Storing a table in more than one physical object, across one or more table spaces
  - Each table space contains a range of the data that can be found very efficiently
- Why?
  - Increase table capacity limit
  - Increase large table manageability
  - Improve SQL performance through partition elimination
  - Provide fast & online data roll-in and roll-out
  - Converge towards Informix functionality
  - Family compatibility with DB2 on zOS and IDS

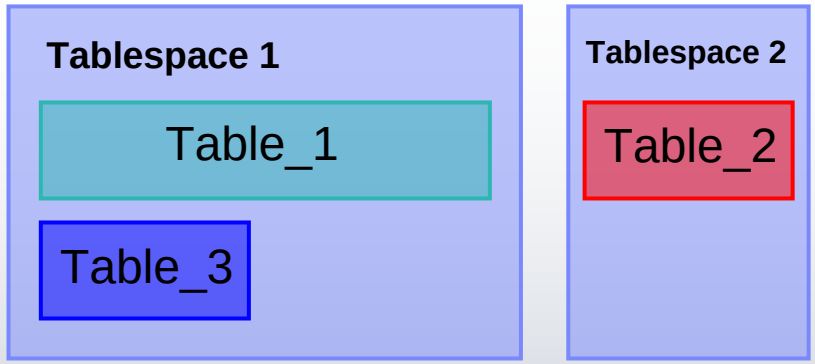
Table Partitioning : Benefits

Without Partitioning

With Partitioning



Without Partitioning



With Partitioning

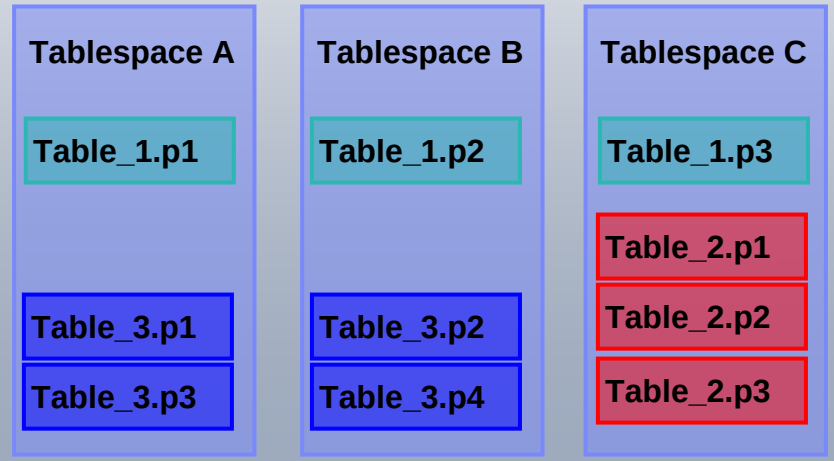


Table Partitioning

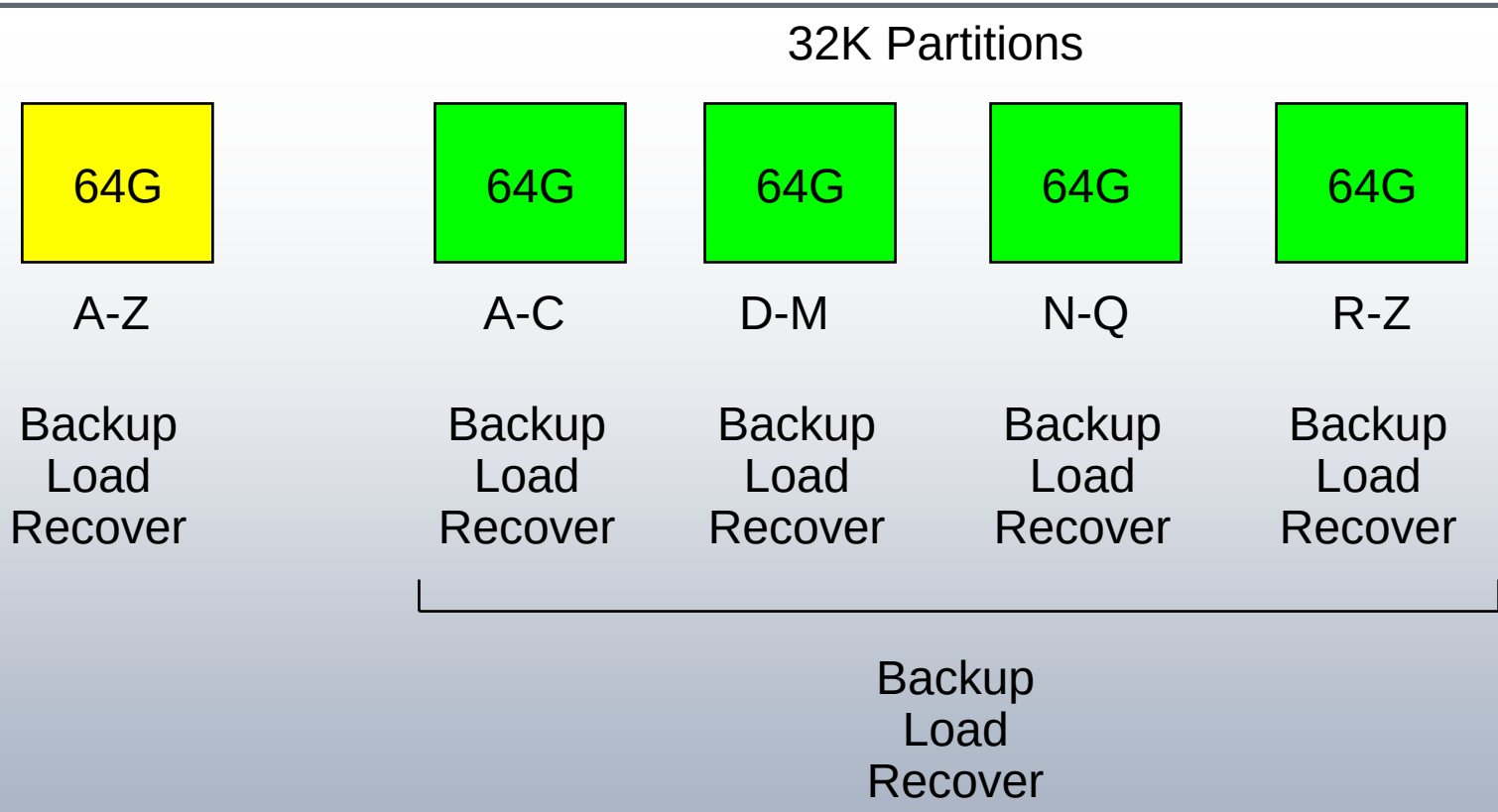
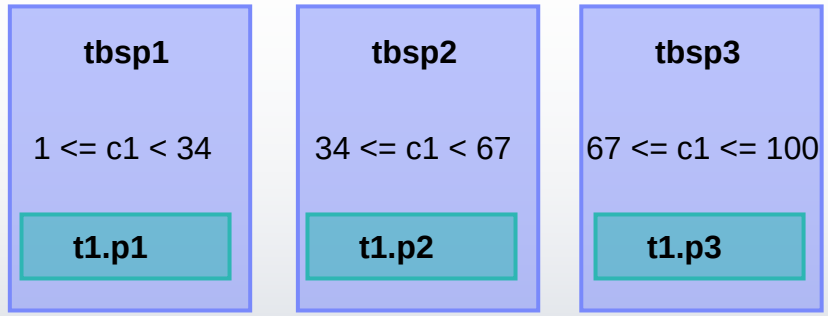


Table Partitioning

Creating a Range Partitioned Table

- Short and Long Forms
- Partitioning column(s)
  - Must be base types (eg. No LOBS, LONG VARCHARS)
  - Can specify multiple columns
  - Can specify generated columns
- Notes
  - Special values, MINVALUE, MAXVALUE can be used to specify open ended ranges, eg:

    CREATE TABLE t1 …
    (STARTING(MINVALUE)
    ENDING(MAXVALUE) …



Short Form

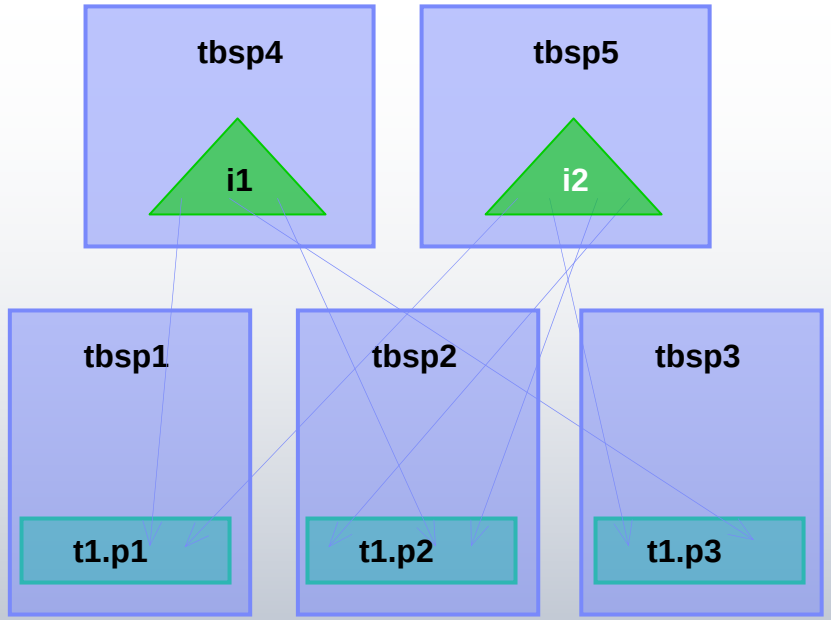
```
CREATE TABLE t1(c1 INT)
  IN tbsp1, tbsp2, tbsp3
  PARTITION BY RANGE(c1)
  (STARTING FROM (1) ENDING( 100) EVERY (33))
```

Long Form

```
CREATE TABLE t1(c1 INT)
  PARTITION BY RANGE(a)
  (STARTING FROM (1) ENDING(34)IN tbsp1,
  ENDING(67)  IN tbsp2,
  ENDING(100) IN tbsp3)
```

Storage Mapping: Indexes are Global in Viper

- Indexes are ***global*** (in Viper)
- Each index is in a separate storage object
  - By default, in the same tablespace as the first data partition
  - Can be created in different tablespaces, via
    - INDEX IN clause on CREATE TABLE (default is tablespace of first partition)
    - New IN clause on CREATE INDEX
- Recommendation
  - Place indexes in LARGE tablespaces

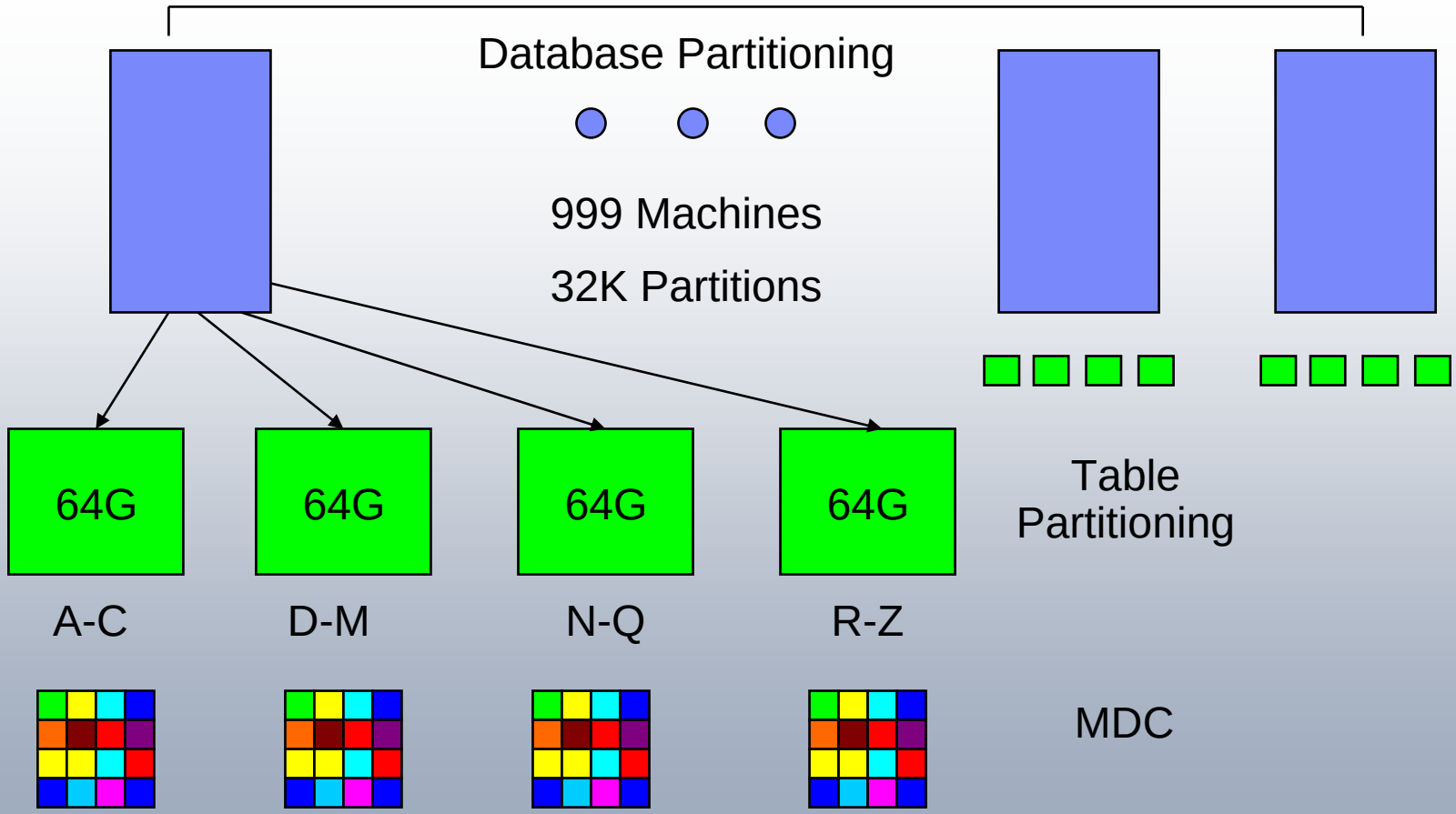



```
CREATE TABLE t1(c1 INT, c2 INT, …)
 IN tbsp1, tbsp2, tbsp3
 INDEX IN tbsp4
 PARTITION BY RANGE(a)
 (STARTING FROM (1) ENDING (100)
   EVERY (33))
CREATE INDEX i1(c1)
CREATE INDEX i2 (c2) IN tbsp5
```

New Operations for Roll-Out and Roll-In

- ALTER TABLE … DETACH
  - An existing range is split off as a stand alone table
  - Data instantly becomes invisible
  - Minimal interruption to other queries accessing table
- ALTER TABLE … ATTACH
  - Incorporates an existing table as a new range
  - Follow with SET INTEGRITY to validate data and maintain indexes
  - Data becomes visible all at once after COMMIT
  - Minimal interruption to other queries accessing table
- Key points
  - No data movement
  - Nearly instantaneous
  - SET INTEGRITY is now online

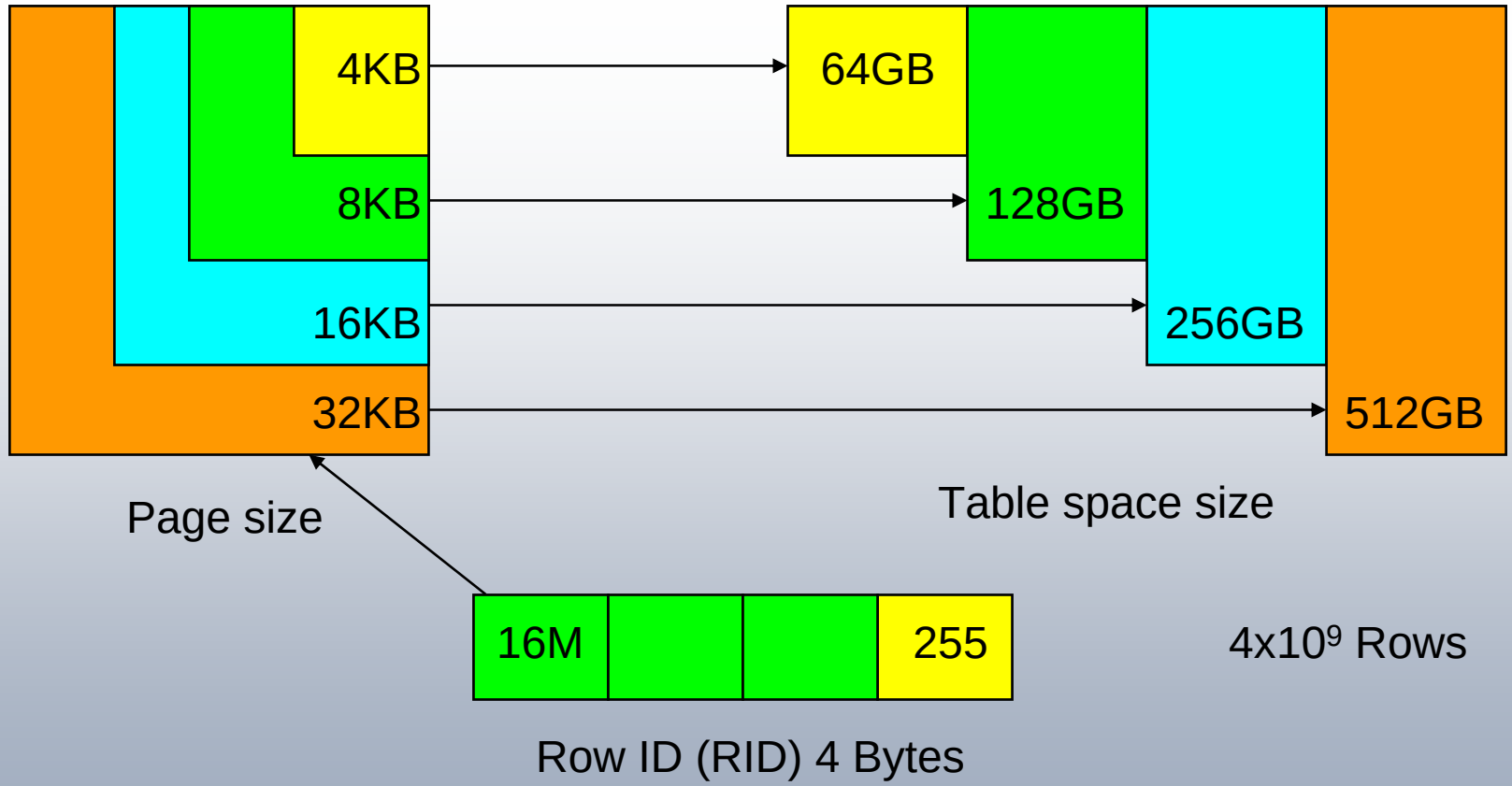
Hybrid Partitioning



Large Row Identifiers

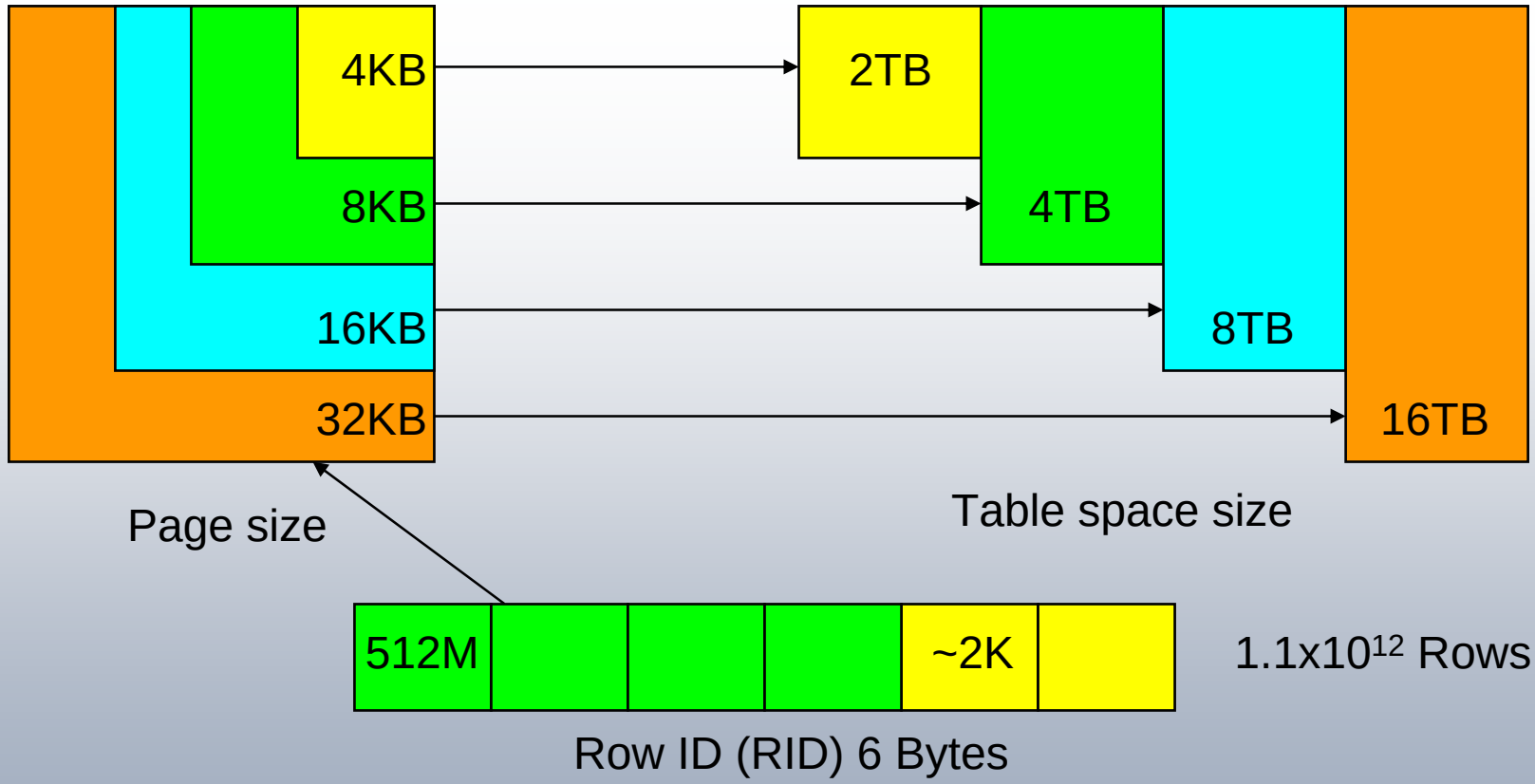
- Increase In table size limits and rows per page
  - Tablespace level definition
  - DMS Tablespace only
- ALTER TABLESPACE <name> CONVERT TO LARGE
  - Tablespace is locked, definition is modified and catalogues are updated
  - Indexes will need to be reorganized
    - Every index for every table in the converted tablespace needs to be reorganized or rebuilt to convert the RID entries from regular to large

Previous Table Space Design



For tables in all table spaces (regular, temporary, DMS, SMS)

New Large and Temporary Table Space Design



For tables in LARGE table spaces (DMS only)
Also all SYSTEM and USER temporary table spaces

Less Limits

- Support for larger index key parts and number of columns
- Support for >18 Char Function Name
- Increase identifier limits to 128 bytes

Version	Length of index key parts	# of columns in index key
Pre-Viper	1024	16
Post-Viper	1024 – 4K page	64
	2048 – 8K page	64
	4096 – 16 K page	64
	8192 – 32 k page	64

IBM Software Group

Granular Security

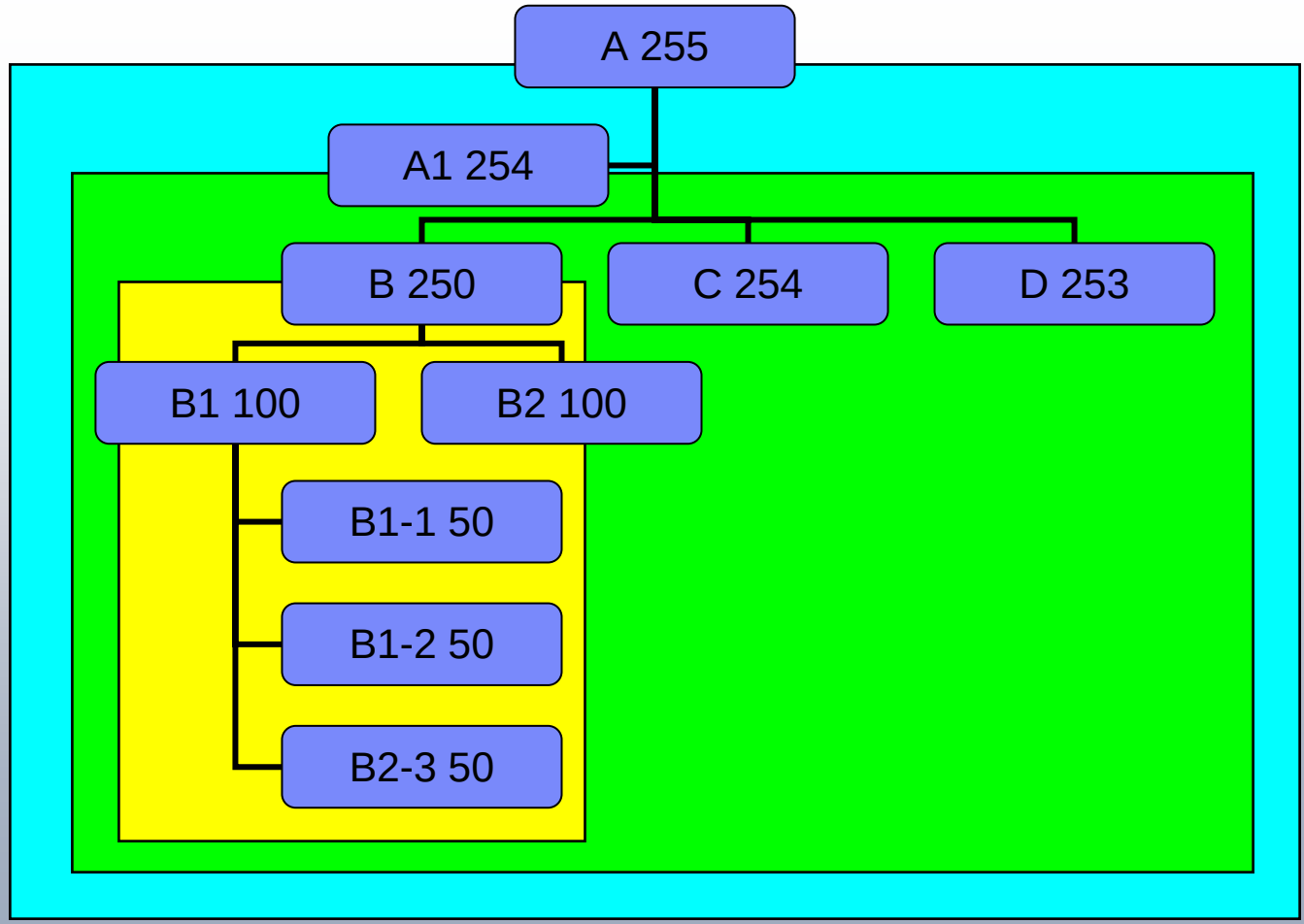
Securing tables at the row or column level

ON DEMAND BUSINESS

Security - Label Based Access Control

- Label Based Access Control (LBAC)
  - A “label” is associated with both user sessions and data rows or columns
  - Rules for comparing users and data labels provide allow access controls to be applied at the row level
- Labels may consist of multiple components
  - Hierarchical, group or tree types
  - Row labels appear as a single additional column in a protected table, regardless of the number of label components
  - User labels are granted by a security administrator
- Similar to the label security support in DB2 for z/OS v8

LBAC Hierarchy – Tree



LBAC Query

```
SELECT * FROM EMP
  WHERE
    SALARY >= 50000
```

No LBAC	SEC=254	SEC=100	SEC=50	ID	SALARY
				255	60000
				100	50000
				50	70000
				50	45000
				60	30000
				250	56000
				102	82000
				100	54000
				75	33000
				253	46000
				90	83000
				200	78000

IBM Software Group

Table Compression

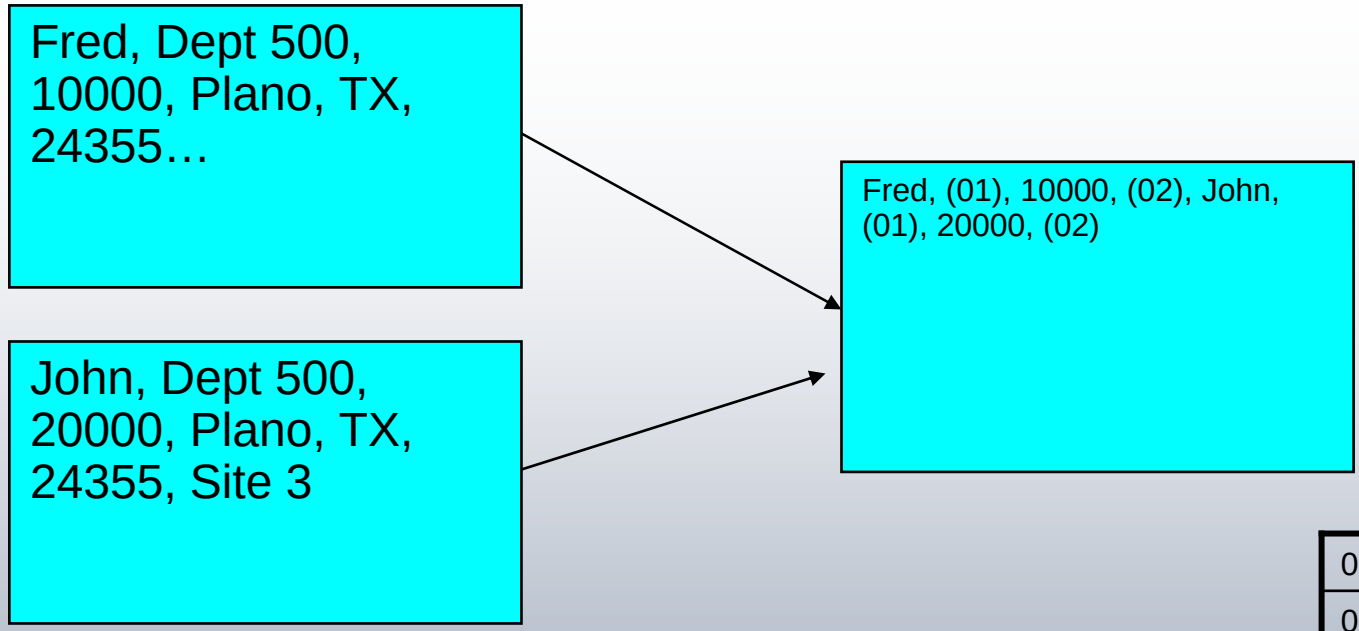
Saving disk space for large database installations

ON DEMAND BUSINESS

DB2 Compression

- NULL and Default Value Compression (V8 GA)
  - No disk storage consumed for NULL column values, zero length data in variable length columns and system default values
- Multidimensional Clustering (V8 GA)
  - Significant index compression can be achieved through block indexes
    - One key per thousands of records (vs one key per record with traditional indexes)
- Database Backup Compression (V8 FP4)
  - Smaller backup images; compress index and lf/lob tablespaces
- Data Row Compression (Viper)

Row Compression Using a Compression Dictionary



01	Dept 500
02	Plano, TX, 24355
…	…

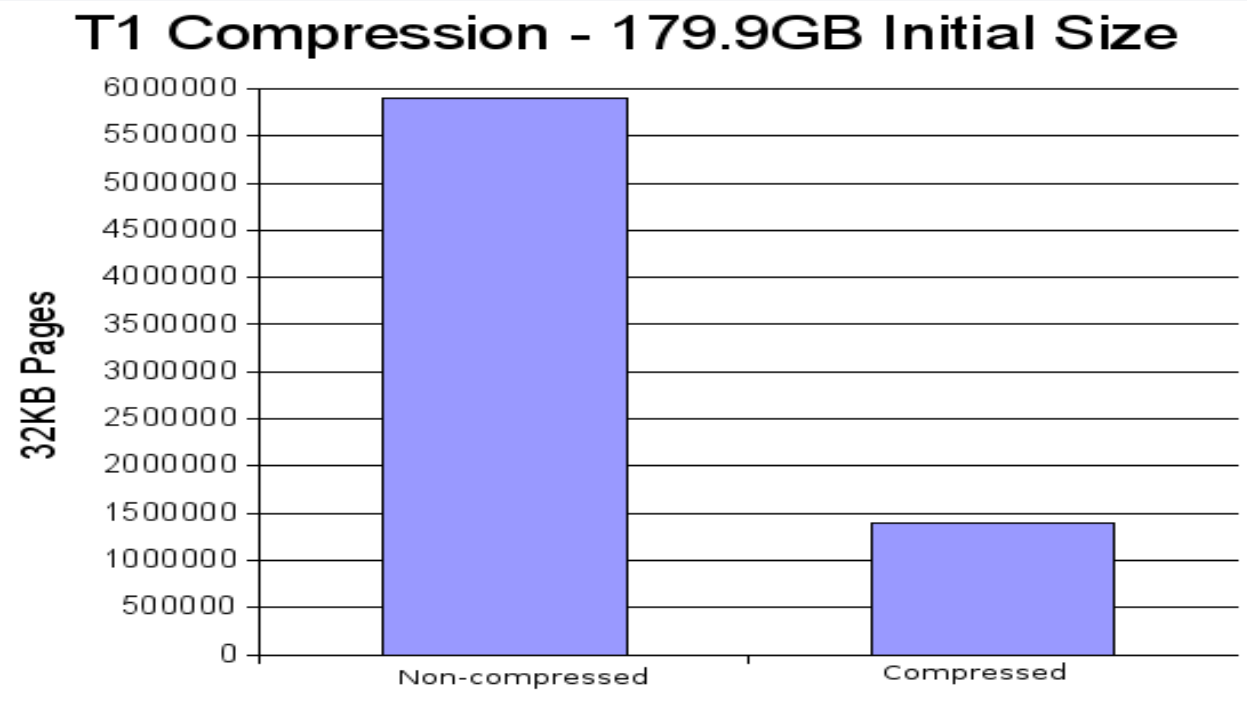
Side tables contain repeated information from the rows

- Can be across column boundaries or within columns

More Compression Ratios (Customer Data)

Compression Type	32KB Page Count	Space Required on Disk
No compression	5893888	179.9GB
Row compression	1392446	42.5GB

% Pages Saved: 76.4%



Compression Ratio – Customer Data