DB2 Information Management Software

DB2 9 for Linux, UNIX, Windows

Giulia Caliari
giulia_caliari@it.ibm.com

Bari, 20.09.2006

Take Control.

Lancio italiano di DB2 9 (Viper)

"L'annuncio più importante degli ultimi 20 anni"

24 Ottobre 2006
Roma

8 Novembre 2006
Milano

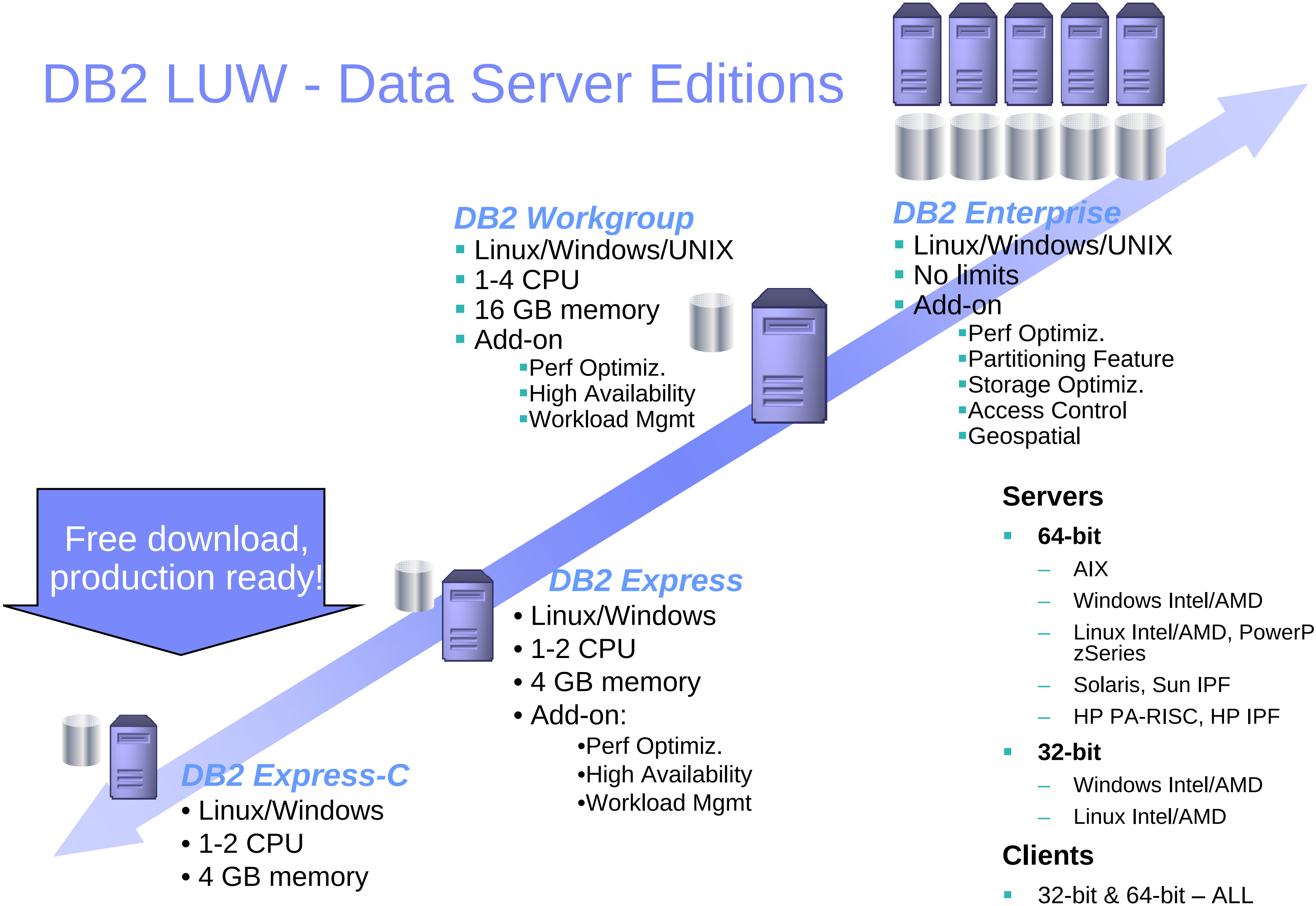
Perchè partecipare?

- E' il primo database a struttura multipla: DB2 9 è oggi l'unico sul mercato che contenga struttura relazionale e quella per dati XML in modalità nativa
- E' un cambio di generazione dei database, non è il semplice lancio di una nuova versione di prodotto

Agenda

- **Introduction**
- **New XML Support**
  - Why Native XML in DB2?
  - Native XML support in DB2 9
- **Other Features**
  - Table Partitioning
  - Granular Security
  - Data Compression
  - New Autonomic Features:
    - Self Managing Memory Tuning
    - Automatic Storage

DB2 LUW - Data Server Editions



Free download, production ready!

DB2 Express-C

- Linux/Windows
- 1-2 CPU
- 4 GB memory

DB2 Express

- Linux/Windows
- 1-2 CPU
- 4 GB memory
- Add-on:
  - Perf Optimiz.
  - High Availability
  - Workload Mgmt

DB2 Workgroup

- Linux/Windows/UNIX
- 1-4 CPU
- 16 GB memory
- Add-on
  - Perf Optimiz.
  - High Availability
  - Workload Mgmt

DB2 Enterprise

- Linux/Windows/UNIX
- No limits
- Add-on
  - Perf Optimiz.
  - Partitioning Feature
  - Storage Optimiz.
  - Access Control
  - Geospatial

Servers

- **64-bit**
  - AIX
  - Windows Intel/AMD
  - Linux Intel/AMD, PowerPC, zSeries
  - Solaris, Sun IPF
  - HP PA-RISC, HP IPF
- **32-bit**
  - Windows Intel/AMD
  - Linux Intel/AMD

Clients

- 32-bit & 64-bit – ALL

DB2 Strategy
For application developers

- A “bit bucket at the end of the driver”
- Quality of the database is measured by how much I like the xxx driver (where xxx is JDBC, ADO.NET, ADO, ODBC, OLE DB, CLI)
- Why can’t database programming be just like yyy (where yyy is Java, C#, VB, Perl, Python, Cobol etc.)

- DB2 as a middleware that is more then a dumb bit bucket
- Provide best implementations of the drivers to support popular APIs (JDBC, ADO.NET, ADO, ODBC, OLE DB, CLI)
- Make database a natural fit in to the programmers’ world

Database Application Development Technologies

- Key Database Technologies
  - SQL / SQL Procedures
  - XML
  - SOA / Web Services
- Developer communities
  - C/C++
  - Java (JDBC / SQLJ)
  - .NET (C#, VB .NET)
  - Open Source
    - PHP
    - Perl
    - Python

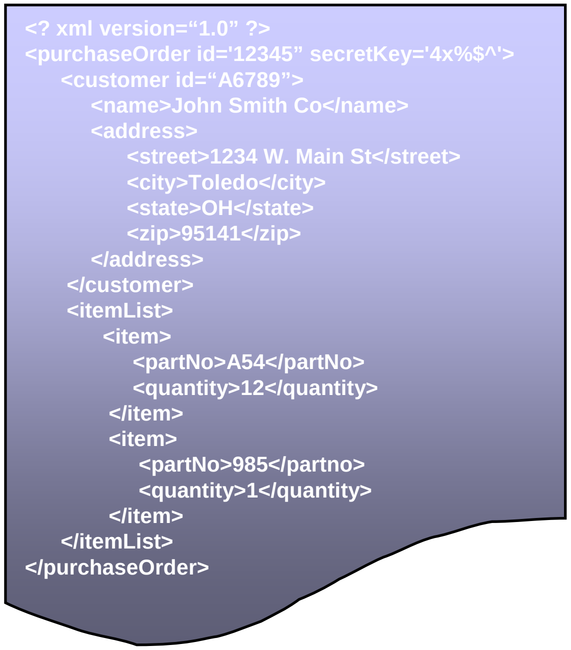
DB2 Information Management Software

Native XML support in DB2

Bari, 20.09.2006

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
    - For any data: structured, semi-structured, schema-less
  - Platform Independent
    - Not tied to any platform, operating system, language or software vendor
    - Can be easily exchanged between applications
    - Can be easily validated, transformed into other formats, etc.
  - 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 – The change is fundamental

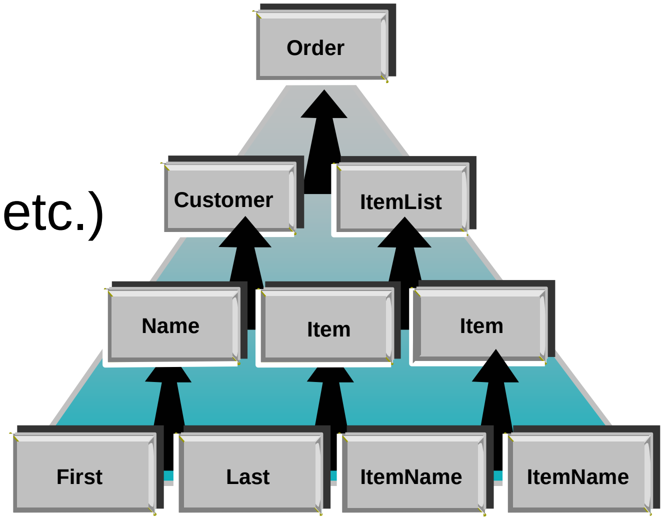
- Relational is a data model:
  - •Relations (tables)
  - •Attributes (columns)
  - •Set based w/ some sequences
  - •Strict schema

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

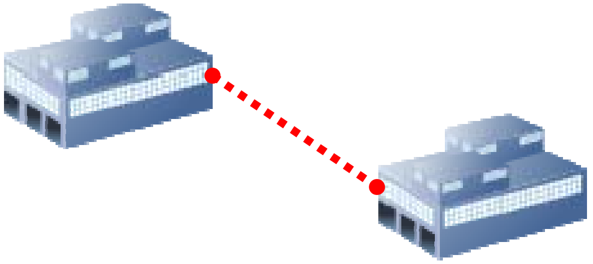
- XML is a data model:
  - •Nodes (elements, attributes, comments, etc.)
  - •Relationships between nodes
  - •Sequence based w/ some sets
  - •Flexible schema



XML Solves Business Problems Today

- Business to Business Integration
  - Platform independent transport mechanism

    Purchase order triggers transactions flowing over a service oriented architecture

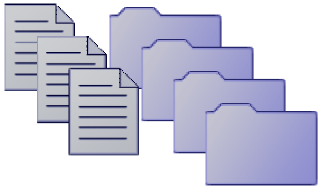


- Document Management
  - Government and legal industry require digital signature

    Tax forms require signature & change year to year

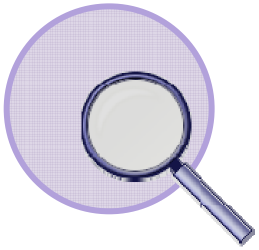
  - Documents often contain sub-documents

    Books, chapters, and sub-chapters



- Business Intelligence
  - Universal representation from multiple sources

    Claims adjustor reviews damage estimates from multiple garages with consideration of original format



Who uses XML? Everybody!

Financial		
ACORD	XML for Insurances	http://www.acord.org/standards/lifexml.aspx
FIXML	Financial Information eXchange protocol	http://www.fixprotocol.org/cgi-bin/Spec.cgi?menu=4
FPML	Financial Product ML	http://www.fpml.org/spec/index.asp
FUNDSML	Funds Markup Language	http://www.funds-xml.org/html/download.htm
XBRL	eXtensible Business Markup Language	http://www.xbrl.org/r
Life Sciences		
AGAVE	Architecture for Genomic Annotation, Visualization and Exchange	http://www.lifecde.com/products/agave/
BSML	Bioinformatic Sequence Markup Language	http://www.bsml.org/resources/default.asp
CML	Chemical Markup Language	http://www.xml-cml.org/
Publication etc.		
SportML	Sport Markup Language	http://www.sportsml.com/specifications.php
NewsML	News Markup Language	http://www.newsml.org/pages/spec_main.php
XBITS	XML Book Industry Transaction Standards	http://www.xmlbits.org/docs.asp
XPRL	eXtensible Public Relations Language	http://www.xprl.org/
Other		
LandML	Land Development Markup Language	http://www.landxml.org/spec.htm
MODA-ML	Middleware tOols and Documents to Enhance the textile/clothing supply chain through xML	http://www.moda-ml.net/moda-ml/repository/schema/V2003-1/default.asp?lingua=en
MatML	Materials Property Data Markup Language	http://www.matml.org/schema.htm
JXDM	Global Justice XML Data Model	http://it.ojp.gov/jxdm/3.0/index.html
ebXML	Electronic Business using eXtensible Markup Language	http://www.ebxml.org/specs/
...	...	...

XML Example: Financial Data (FIXML)

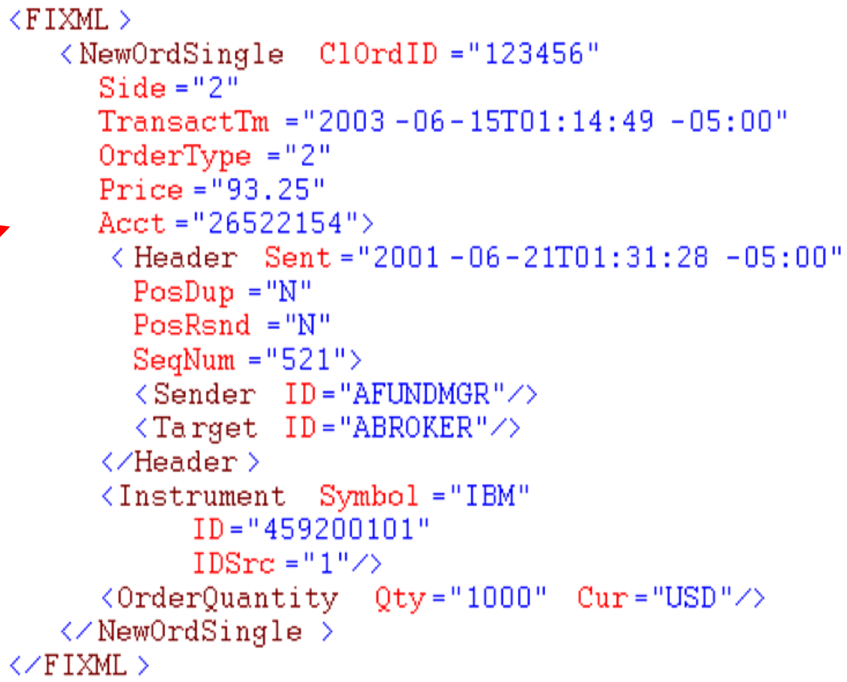
- Buying 1000 Shares of IBM Stock..

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 >
```

When to use XML?
Can be a better choice than relational for...

- **Data that’s inherently hierarchical or nested in nature**
  - Example: Medical data, Bill-of-materials, etc., OO & Multi-value
- **Data sets with sparsely populated attributes**
  - Example: FIXML, FpML, Customer profiles
- **Schema evolution**
  - Example: Frequently changing services/products/processes
- **Variable schemas, many schemas**
  - Example: Data integration, consolidation of diverse data sources
- **Combining structured & unstructured data**
  - Example: CM, Life Sciences, News & Media

XML Data Needs Relational Maturity

- XML Data Needs Protection
  - *Backup and recovery features to ensure continuity*
  - *Data is protected using database security*
  - *Transactions and concurrency*
- Simplified XML Data Access
  - *Centrally store and access difficult to retrieve data*
  - *SQL or XQuery can be used to retrieve data*
  - *Join XML data with it’s related relational data*
- Search Speed
  - *Search documents quickly and efficiently using proven search optimization engine of mature database*
- Optimize Existing Investments
  - *Use existing technology infrastructure and skills to store and manage both relational and XML*

XML Market Projections

- XML Storage is a high growth area

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

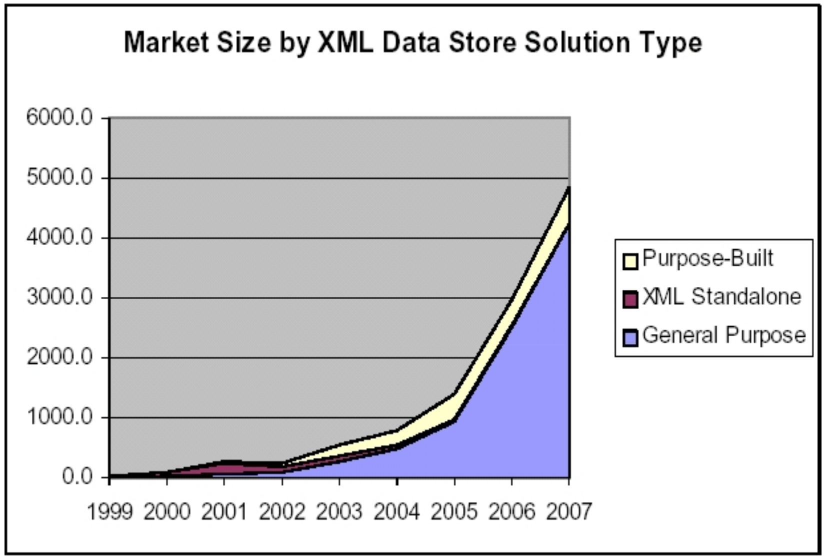
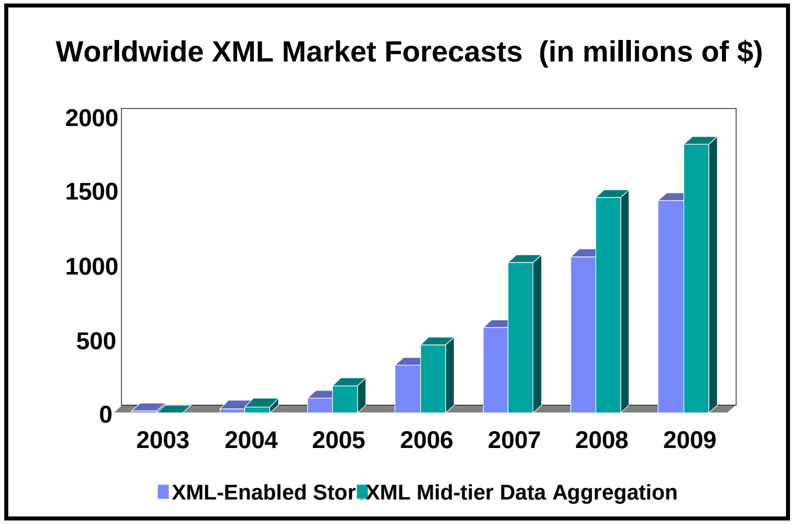


Chart Sources: XML Market Opportunities, Forecasts and Strategies, 2004-2009
Wintergreen Research Inc. ZapThink



- 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

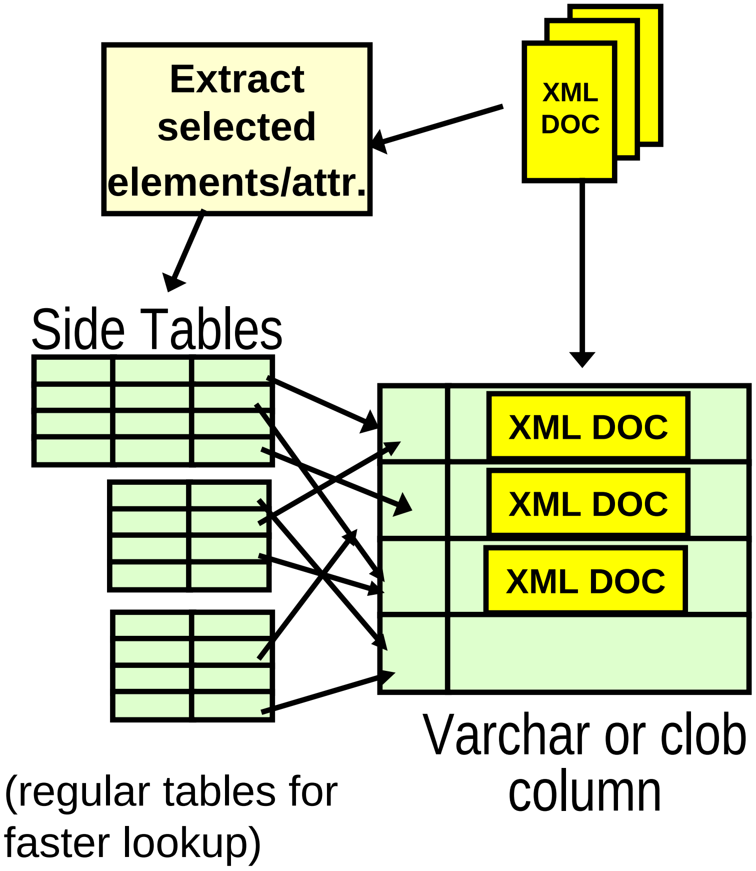
XML Databases (before DB2 9 ..)

- **XML-enabled Databases**
  - The core data model is not XML (but e.g. relational)
  - Mapping between XML data model and DB’s data model is required, or XML is stored as text
  - E.g.: DB2 XML Extender (V7, V8)

- **Native XML Databases**
  - Use the hierarchical XML data model to store and process XML internally
  - No mapping, no storage as text
  - Drawbacks:
    - Still lack of scalability, robustness, performance, utilities, etc..
    - No integration with relational data

XML Enabled Databases: Two Main Options

CLOB/Varchar



Extract selected elements/attr.

XML DOC

Side Tables

XML DOC

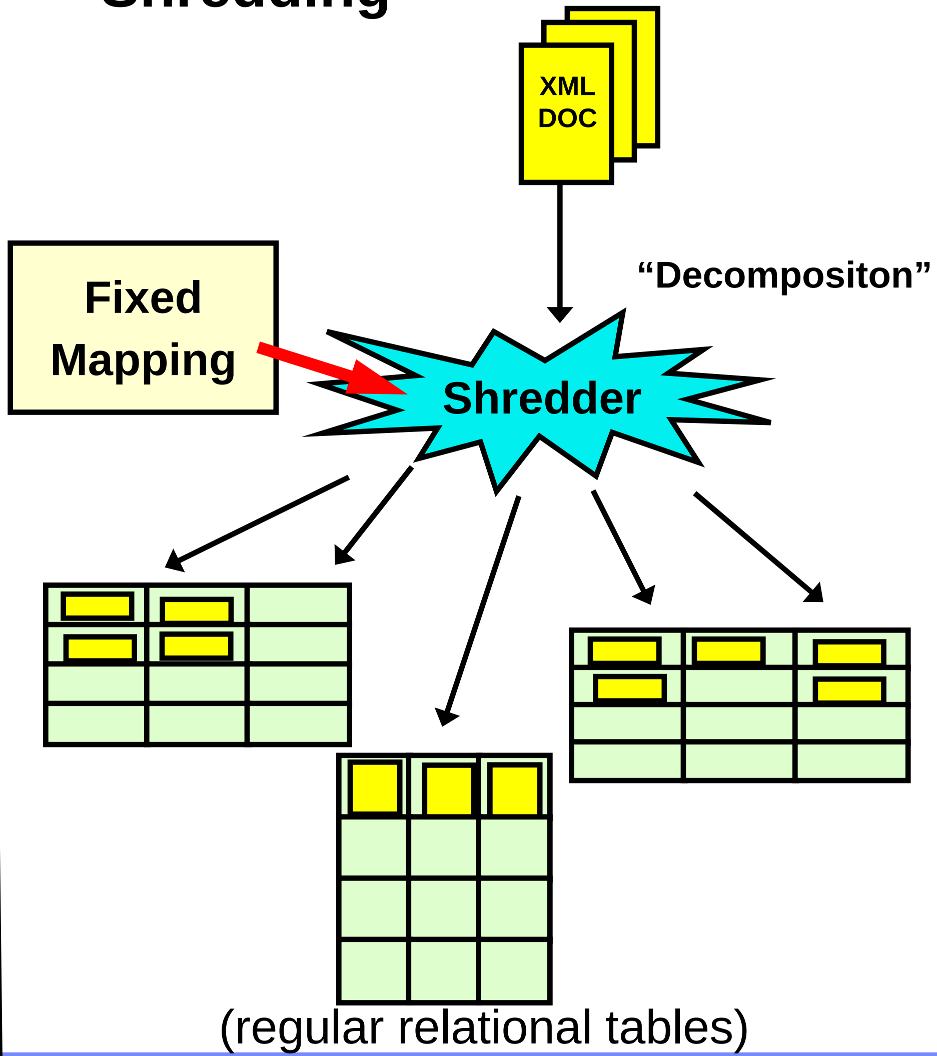
XML DOC

XML DOC

(regular tables for faster lookup)

Varchar or clob column

Shredding



XML DOC

Fixed Mapping

“Decompositon”

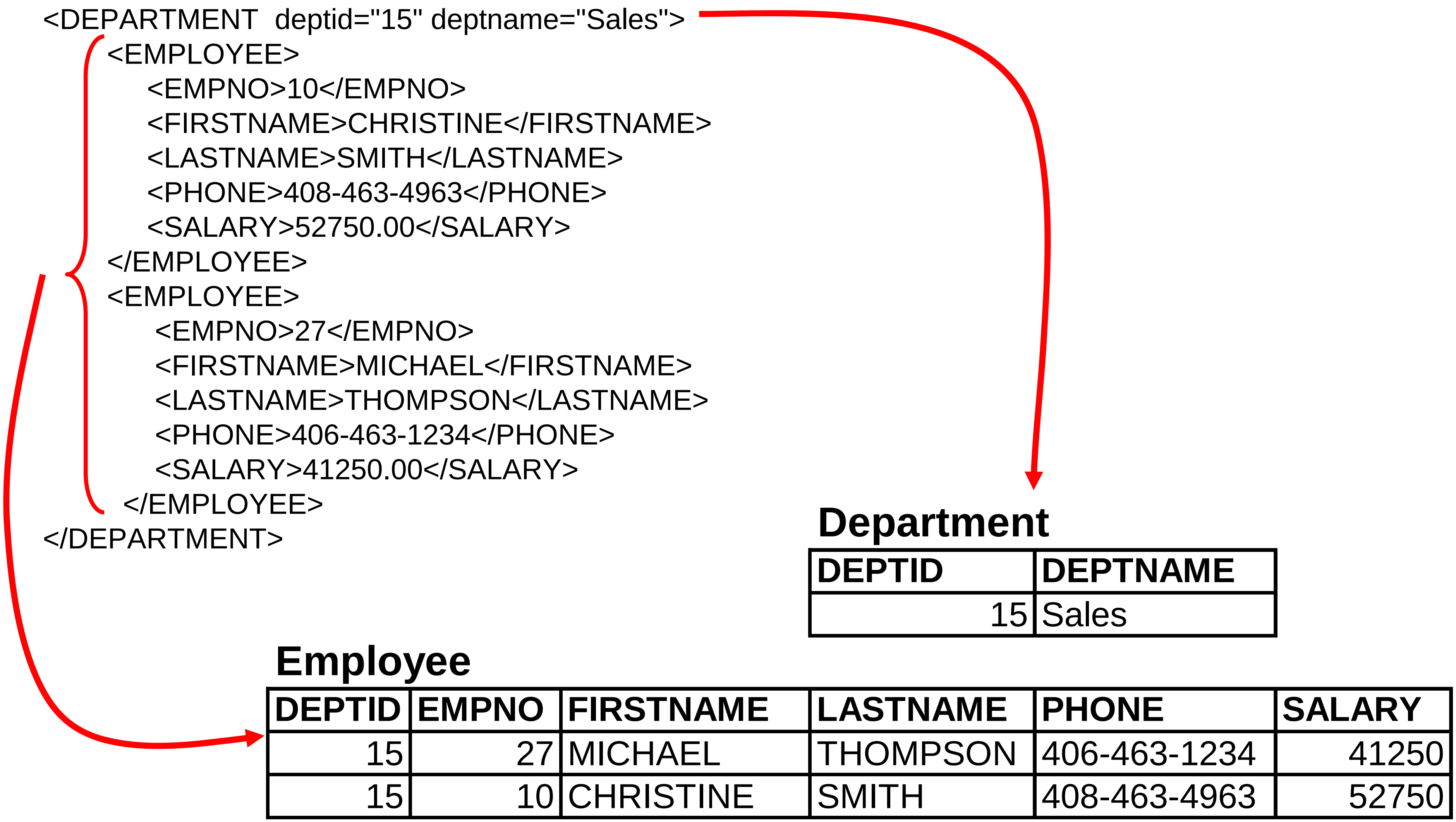
Shredder

(regular relational tables)

Problems of XML-enabled Databases

- **CLOB storage:**
  - Query evaluation & sub-document level access requires costly XML Parsing – too slow
- **Shredding:**
  - Mapping from XML to relational often too complex
  - Often requires dozens or hundreds of tables
  - Complex multi-way joins to reconstruct documents
  - XML schema changes break the mapping
    - no schema flexibility !
    - For example: Change element from single- to multi-occurrence requires normalization of relational schema & data
  - Every transformation (e.g. shredding) is expensive and potentially lossy (e.g., digital signature)

Shredding: A simple case



```
<DEPARTMENT  deptid="15" deptname="Sales">
    <EMPLOYEE>
        <EMPNO>10</EMPNO>
        <FIRSTNAME>CHRISTINE</FIRSTNAME>
        <LASTNAME>SMITH</LASTNAME>
        <PHONE>408-463-4963</PHONE>
        <SALARY>52750.00</SALARY>
    </EMPLOYEE>
    <EMPLOYEE>
         <EMPNO>27</EMPNO>
         <FIRSTNAME>MICHAEL</FIRSTNAME>
         <LASTNAME>THOMPSON</LASTNAME>
         <PHONE>406-463-1234</PHONE>
         <SALARY>41250.00</SALARY>
     </EMPLOYEE>
</DEPARTMENT>
```

Department

DEPTID	DEPTNAME
15	Sales

Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

Shredding: A schema change…

"Employees are now allowed to have multiple phone numbers…"

```
<DEPARTMENT  deptid="15" deptname="Sales">
      <EMPLOYEE>
          <EMPNO>10</EMPNO>
          <FIRSTNAME>CHRISTINE</FIRSTNAME>
          <LASTNAME>SMITH</LASTNAME>
          <PHONE>408-463-4963</PHONE>
          <PHONE>415-010-1234</PHONE>
          <SALARY>52750.00</SALARY>
      </EMPLOYEE>
      <EMPLOYEE>
           <EMPNO>27</EMPNO>
           <FIRSTNAME>MICHAEL</FIRSTNAME>
           <LASTNAME>THOMPSON</LASTNAME>
           <PHONE>406-463-1234</PHONE>
           <SALARY>41250.00</SALARY>
        </EMPLOYEE>
</DEPARTMENT>
```

Requires:

- **Normalization of existing data !**
- **Modification of the mapping**
- **Change of applications**

Phone

EMPNO	PHONE
27	406-463-1234
10	415-010-1234
10	408-463-4963

Department

DEPTID	DEPTNAME
15	Sales

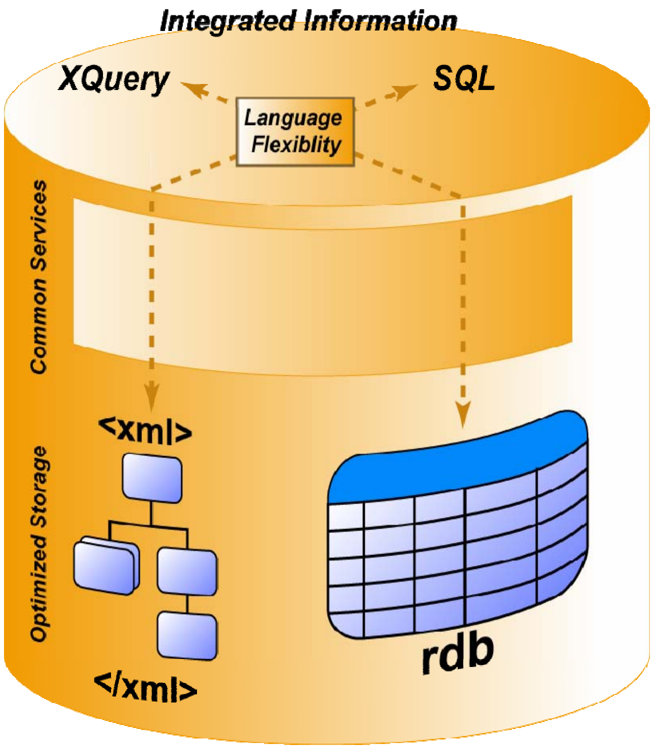
Costly!

Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	~~PHONE~~	SALARY
15	27	MICHAEL	THOMPSON	~~406-463-1234~~	41250
15	10	CHRISTINE	SMITH	~~408-463-4963~~	52750

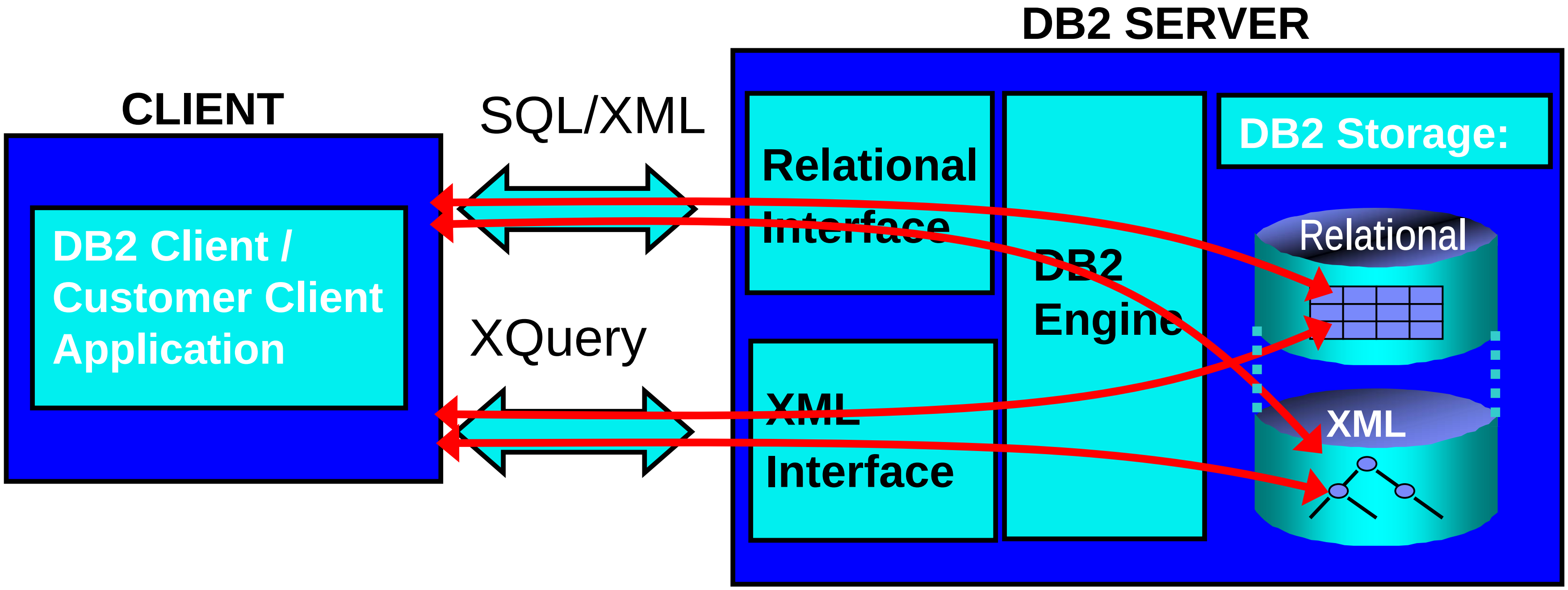
DB2 9: The Hybrid Database

- **Security and protection of XML data in a mature database**
- **Relational operators for Insert, Update and Delete**
- **Store some document parts in object-relational, and others in XML**
- **XQuery gives applications more flexible search paradigms**
- **Easily access XML data using SQL**
- **Easily correlate XML and relational information**



Integration of XML & Relational Capabilities

- Applications combine XML & relational data
- Native XML data type (server & client side)
- XML Capabilities in all DB2 components

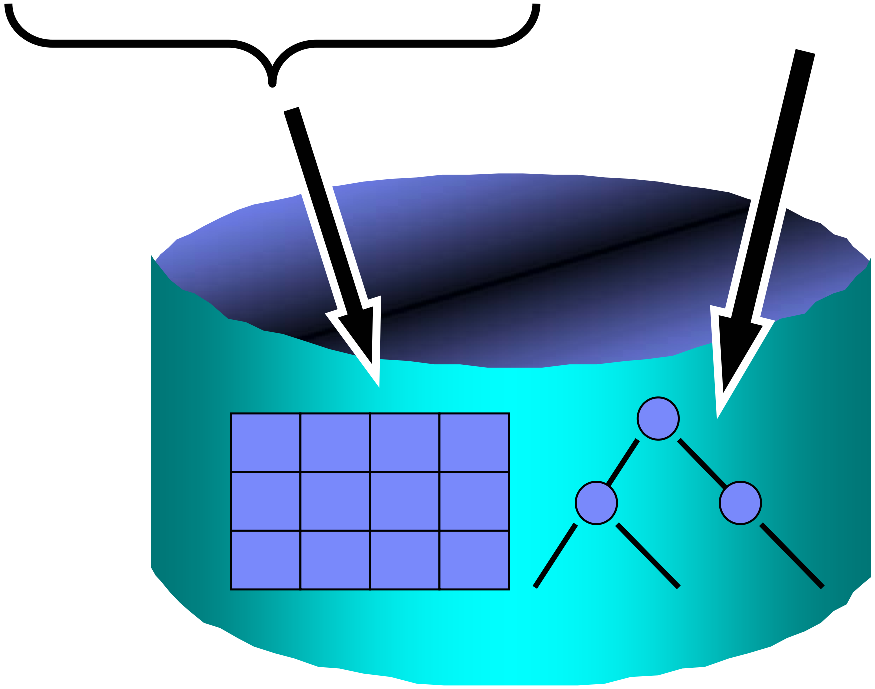


Native XML Support in DB2 9

- ➡ XML as a native data type
- ➡ Pure Native XML Storage
- ➡ Querying XML and relational via SQL, XQuery, or mix
- XML Indexes and Text Search
- XML Schema Support and Validation
- Decomposition
- Application Support (Java, C/C++, .NET, PHP, etc.)
- Visual Tooling, Control Center
- DB2 Utilities

Native XML Storage

create table dept (deptID char(8), doc xml)

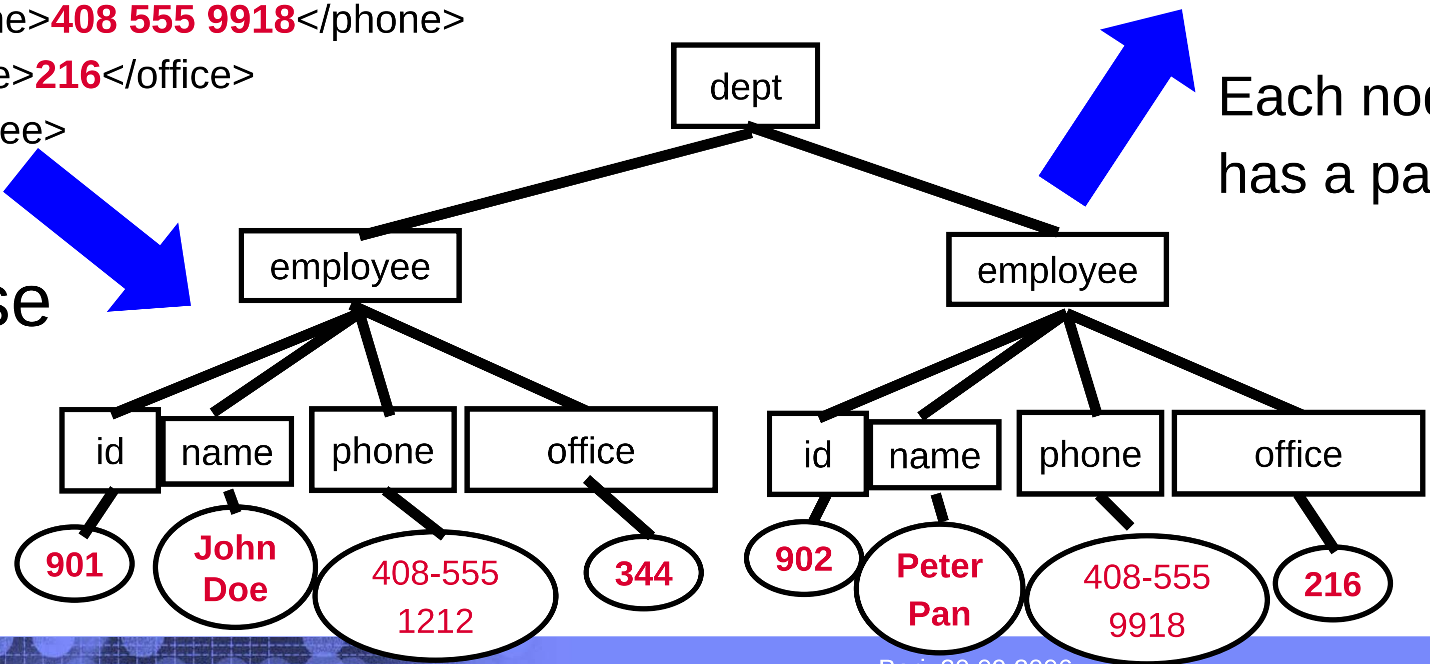


- **Store XML in parsed hierarchical format**
- **Relational columns are stored in relational format**
- **XML columns are stored** natively
- **All XML data is stored in XML-typed columns**

The Document Tree

```
<dept>
   <employee id=901>
        <name>John Doe</name>
        <phone>408 555 1212</phone>
        <office>344</office>
    </employee>
    <employee id=902>
        <name>Peter Pan</name>
        <phone>408 555 9918</phone>
        <office>216</office>
    </employee>
</dept>
```

/
/dept
/dept/employee
/dept/employee/@id
/dept/employee/name
/dept/employee/phone
/dept/employee/office
(...)



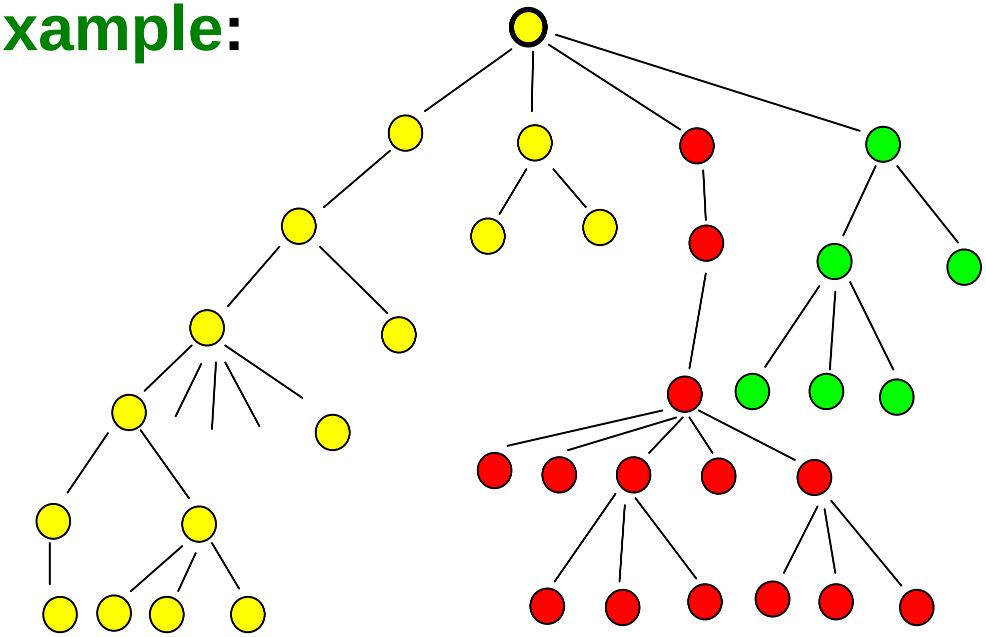
Parse

Each node has a path

XML Node Storage Layout

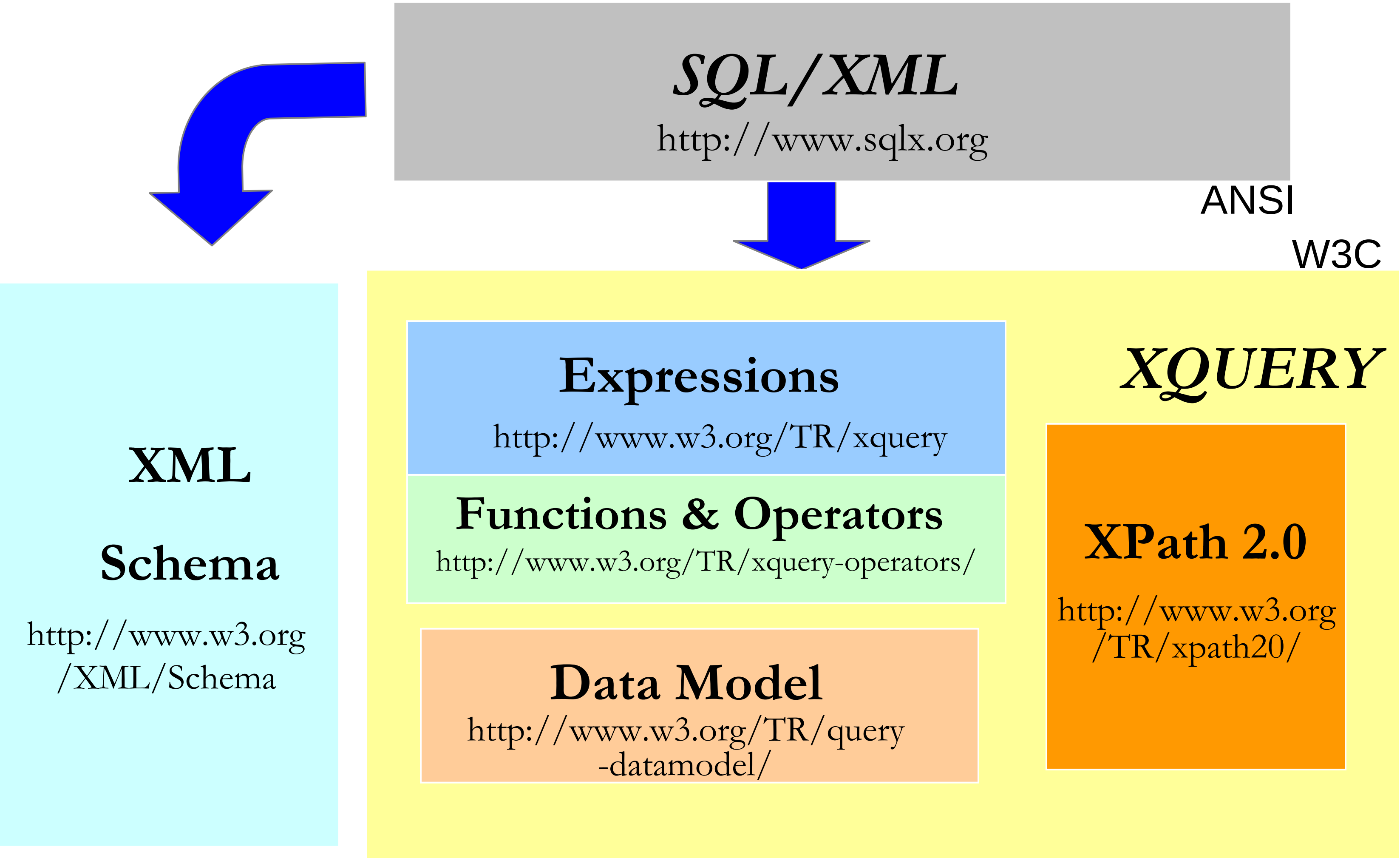
- **Node hierarchy of an XML doc stored on DB2 pages**
- **Documents that don’t fit on 1 page: split into pages/regions**

Example:



Document split into 3 regions, stored on 3 pages

XML Standards: querying XML data



SQL/XML

http://www.sqlx.org

ANSI

W3C

XML

Schema

http://www.w3.org /XML/Schema

Expressions

http://www.w3.org/TR/xquery

Functions & Operators

http://www.w3.org/TR/xquery-operators/

Data Model

http://www.w3.org/TR/query -datamodel/

XQUERY

XPath 2.0

http://www.w3.org /TR/xpath20/

XPath Expressions & predicates

- **Root Node (/), Step (/)**
  - /customers/customerinfo/name
- **Descendants (//)**
  - /customers//city
- **Any element node (*)**
  - /customers/customerinfo/*/city
- **Attribute node (@)**
  - //@country
- **Text node Text()**
  - //phone/text()
- **Predicates are always embedded in square brackets**
  - /customers/customerinfo/addr[@country=‘Canada’]

```
<customers>
 <customerinfo>
        <name>Kathy Smith</name>
        <addr country="Canada">
                <street>25 EastCreek</street>
                <city>Toronto</city>
                 <prov-state>Ontario</prov-state>
                 <pcode-zip>M8X-3T6</pcode-zip>
             </addr>
        <phone type="work">416-555-1358</phone>
 </customerinfo>
 <customerinfo>
       <name>Amir Malik</name>
         <addr country="United States">
                  <street>555 Bailey Ave</street>
                  <city>San Jose</city>
                  <prov-state>California</prov-state>
                  <pcode-zip>95141</pcode-zip>
             </addr>
        <phone type="work">408-555-1358</phone>
        <phone type=“home">408-555-1375</phone>
 </customerinfo>
</customers>
```

Querying relational & XML data

Pure SQL

```
select info from xmlcustomer where
CID=1004
```

XQuery

```
xquery for $i in db2-fn:xmlcolumn(
'XMLCUSTOMER.INFO')return $i/customerinfo
/addr
```

XQuery: The FLWOR Expression

- **FOR: iterates through a sequence, bind variable to items**
- **LET: binds a variable to a sequence**
- **WHERE: eliminates items of the iteration**
- **ORDER: reorders items of the iteration**
- **RETURN: constructs query results**

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

 xquery
 for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
 let $emp := $d//employee/name
 where $d/@bldg > 95
 order by $d/@bldg
 return
    <EmpList>
    {$d/@bldg, $emp}
    </EmpList>
```

Input

```
<dept bldg=101>
   <employee id=901>
         <name>John Doe</name>
          <phone>408 555 1212</phone>
          <office>344</office>
     </employee>
     <employee id=902>
          <name>Peter Pan</name>
          <phone>408 555 9918</phone>
          <office>216</office>
     </employee>
</dept>
```

XQuery Join over 2 XML Columns

- **Nested FLWOR:**

```
XQUERY
  for $book in db2-fn:xmlcolumn('BOOKS.DOC')/book
    for $entry in db2-
  fn:xmlcolumn('REVIEWS.DOC')/entry
    where $book/title = $entry/title
  return
  <review>
    {$entry/review/text()}
  </review>;
```

Querying relational & XML data

Pure SQL

- `select info from xmlcustomer where CID=1004`

XQuery SQL

-
  ```
  xquery for $i in db2-fn:sqlquery ('select
  INFO from XMLCUSTOMER where
  CID=1003')/customerinfo return $i
  ```

XQuery

-
  ```
  xquery for $i in db2-fn:xmlcolumn(
  'XMLCUSTOMER.INFO')return $i/customerinfo
  /addr
  ```

Embed SQL in XQuery

for $d in db2-fn:sqlquery(“select doc from dept”)…

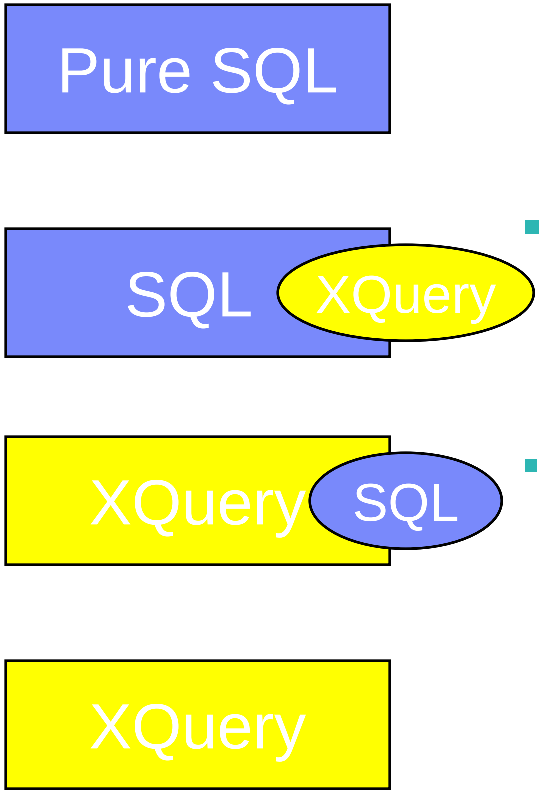
for $d in db2-fn:sqlquery(“select doc from dept where deptID = ‘PR27’ ”)…

for $d in db2-fn:sqlquery(“select doc from dept where deptID LIKE ‘PR%’ ”)…

for $d in db2-fn:sqlquery(“select dept. doc from dept, unit where

dept.deptID=unit.ID and unit.headcount > 200”)…

Querying relational & XML data



-
  ```
  select info from xmlcustomer where
  CID=1004
  ```

-
  ```
  select cid, xmlquery
  ('$doc/customerinfo/phone' passing
  info  as "doc") from xmlcustomer
  ```

-
  ```
  xquery for $i in db2-fn:sqlquery ('select
  INFO from XMLCUSTOMER where
  CID=1003')/customerinfo return $i
  ```

-
  ```
  xquery for $i in db2-fn:xmlcolumn(
  'XMLCUSTOMER.INFO')return $i/customerinfo
  /addr
  ```

Embed XQUERY in SQL

```
create table dept(deptID char(8) primary key,  deptdoc xml);
create table  unit(unitID char(8), headcount integer, bldg integer);
```

```
select   deptID,
         xmlquery('for $d in $doc/dept where $d/@bldg = 101
                   return $d/name' passing deptdoc as “doc")
from dept where deptID <> “PR27”;
```

```
select d.deptID , u.headcount,
       xmlquery(‘$doc/dept/name’ passing d.deptdoc as “doc“)
from dept d, unit u
where d.deptID=u.unitID  and  u.headcount > 200
and xmlquery(‘$doc/dept/@bldg’ passing d.deptdoc as “doc“) = u.bldg
and xmlexists(‘$doc/dept/employee/name’ passing d.deptdoc as “doc“)
```

XMLTable: make table from XML

➔Each item in the sequence returned by the XQuery will result in a row

➔The row data is created with that item as context

```
SELECT X.* from

XMLTABLE (‘db2-fn:xmlcolumn(“PORDERS.PO”)//customer’
      COLUMNS
      “CID”     INTEGER      PATH ‘@id’,
      “Name”    VARCHAR(30) PATH ‘name’,
      “ZipType” CHAR(2)      PATH ‘zip/@type’,
      “Zip”     XML          PATH ‘zip’

   ) AS “X”
```

CID	Name	ZipType	Zip
1325	**Bobby**	**US**	**<zip>33129<zip>**
4711	**null**	**US**	**<zip>95023<zip>**

SQL/XML Publishing Example

```
SELECT
   XMLELEMENT(NAME "Department",
                   XMLATTRIBUTES (e.department AS "name" ),
                   XMLAGG( XMLELEMENT(NAME "emp", e.firstname)  )
                             )  AS "department_list"
FROM employee e
WHERE …..
GROUP BY  e.department;
```

firstname	lastname	department
SEAN	LEE	A00
MICHAEL	JOHNSON	B01
VINCENZO	BARELLI	A00
CHRISTINE	SMITH	A00

```
department_list
<Department name="A00">
    <emp>CHRISTINE</emp>
    <emp>VINCENZO </emp>
    <emp>SEAN</emp>
</Department>

<Department name="B01">
    <emp>MICHAEL</emp>
</Department>
```

Business Benefits of XML with DB2 9

- Lower Development Costs
  - *Reduced code and development complexity*
  - *Improved developer productivity*

Quicken solution development and gain cost savings

- Greater Business Agility
  - *Easily accommodate changes to data and schemas*
  - *Update applications rapidily and reduce maintenance costs*

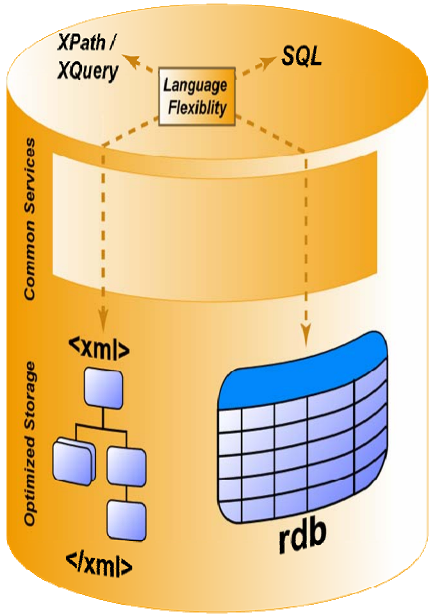
Respond quickly to dynamic conditions and faster time to value

- Improved Business Insight
  - *Access to "hidden gems" (data) in unexploited documents*
  - *Unprecedented application performance*

Gain competitive advantage through better and quicker information

storebrand

DB2 9: Early adopter feedback



Proto-type results using DB2 Viper based SOA solution

- Fast, easy access to richer product & client information
- Quickly create customized products that customers want
- Expected to process five times more business

Development and app. performance re: XML data	with relational data server	with DB2 Viper hybrid data server
Development of search & retrieval business processes	CLOB: 8 hrs Shred: 2 hrs	30 min.
Add field to schema	1 week	5 min.
Relative lines of I/O code (65% reduction)	100	35
Queries	24 - 36 hrs	20 sec - 10 min
Query non-shredded XML element	1 week	½ day

DB2 Information Management Software

Table Partitioning

More room for growth and less limits in the database

Bari, 20.09.2006

Table Partitioning

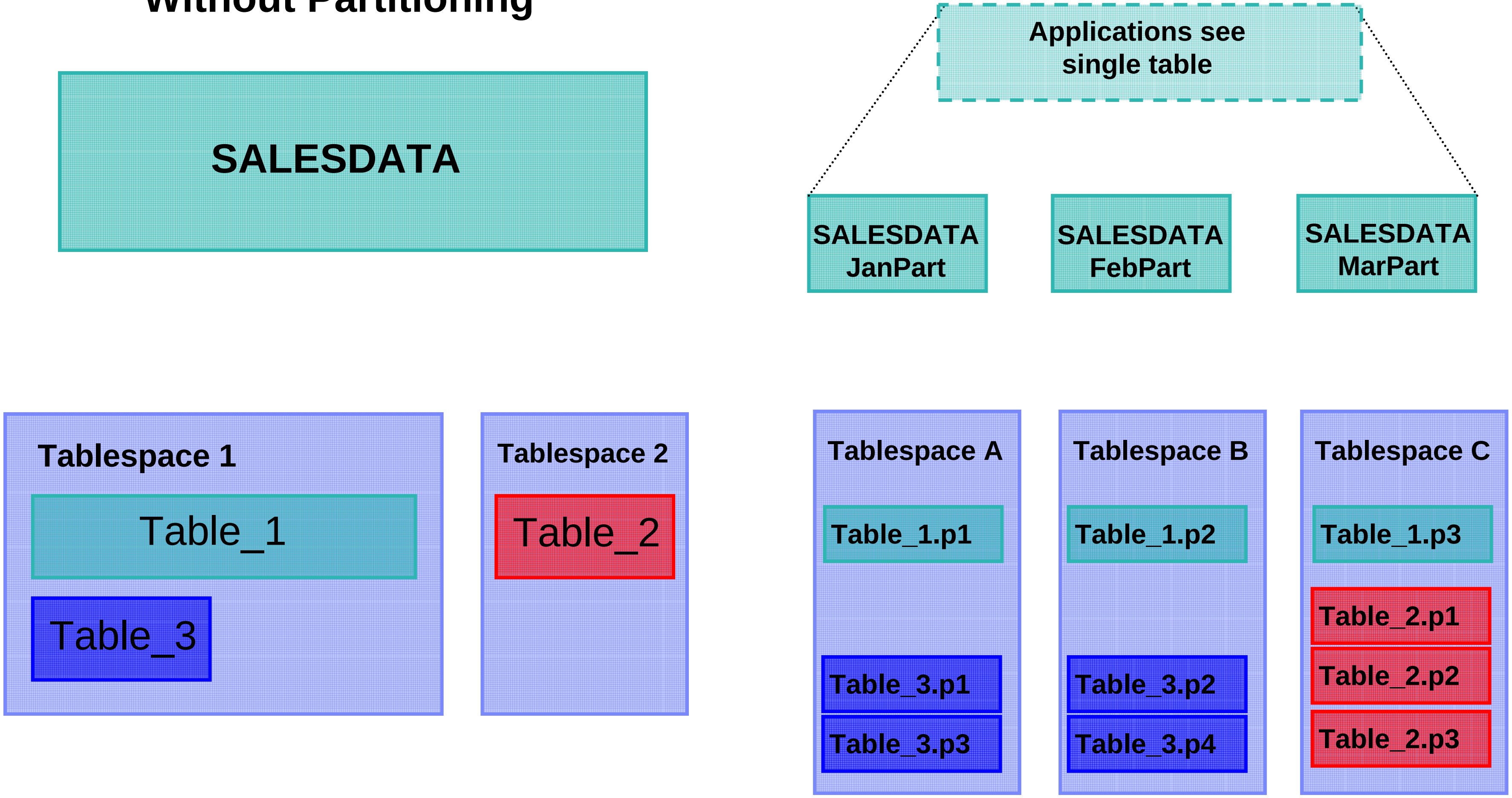
- **What is Table (Range) Partitioning ?**
  - Storing a table in more than one physical object, across one or more table spaces
  - Each table space contains a range of the data that can be found very efficiently

- **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



Applications see single table

SALESDATA

SALESDATA JanPart

SALESDATA FebPart

SALESDATA MarPart

Tablespace 1

Table_1

Table_3

Tablespace 2

Table_2

Tablespace A

Table_1.p1

Table_3.p1

Table_3.p3

Tablespace B

Table_1.p2

Table_3.p2

Table_3.p4

Tablespace C

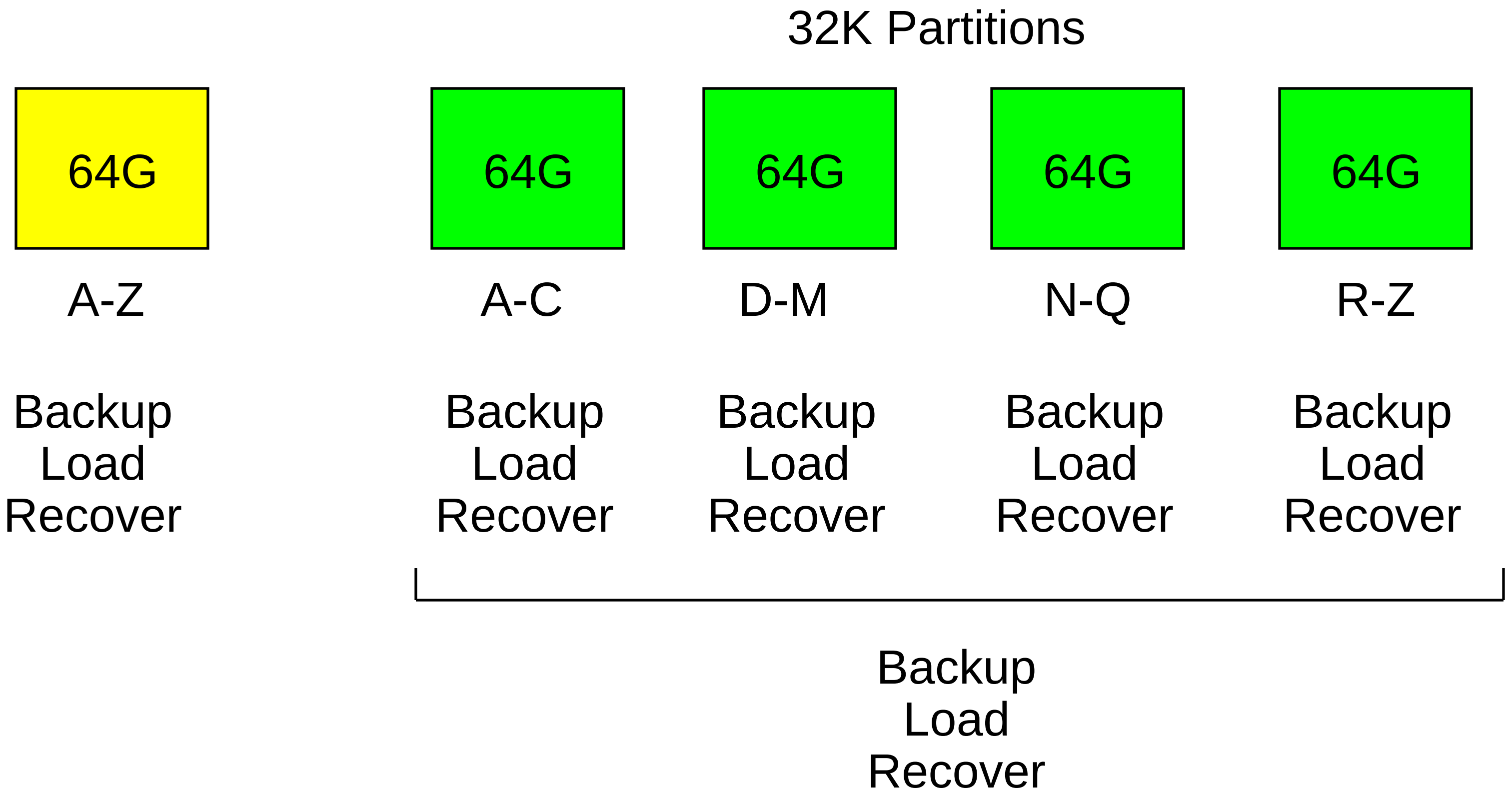
Table_1.p3

Table_2.p1

Table_2.p2

Table_2.p3

Table Partitioning



32K Partitions

64G

A-Z

Backup
Load
Recover

64G

A-C

Backup
Load
Recover

64G

D-M

Backup
Load
Recover

64G

N-Q

Backup
Load
Recover

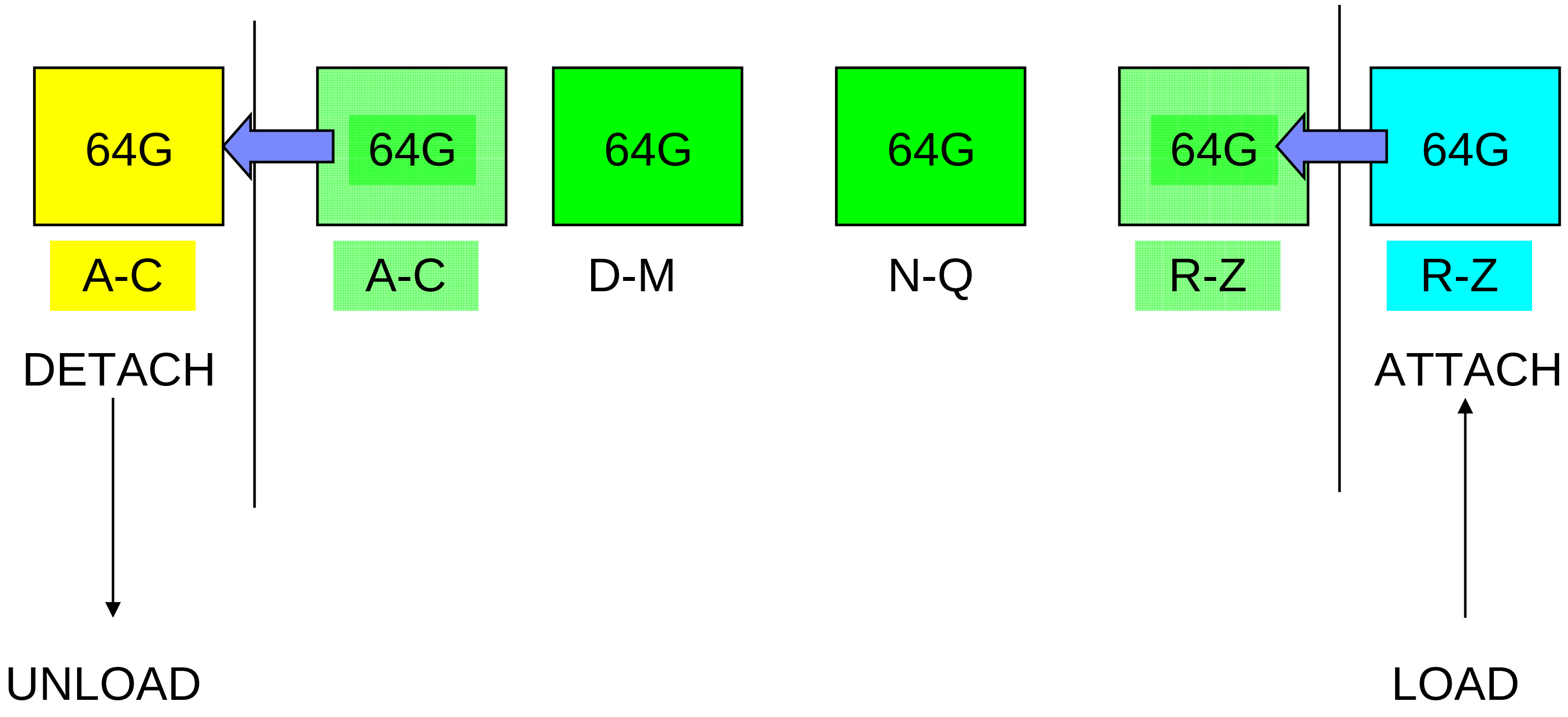
64G

R-Z

Backup
Load
Recover

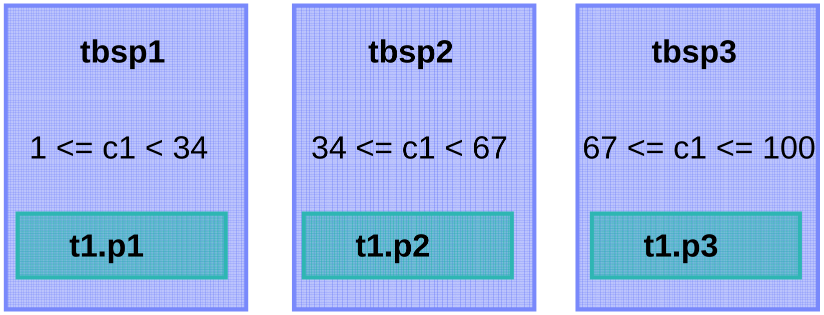
Backup
Load
Recover

Table Partitioning / Attach and Detach



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



Short Form

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

Long Form

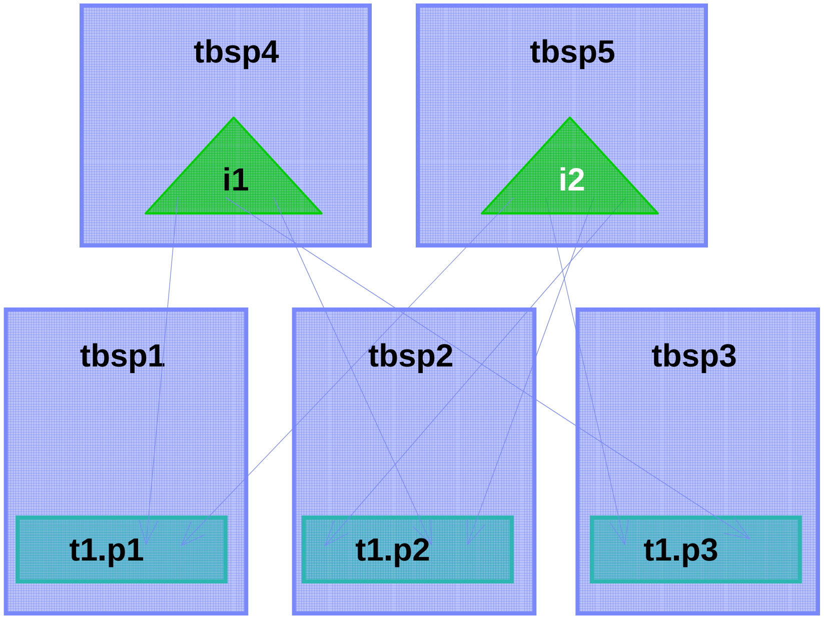
```
CREATE TABLE t1(c1 INT)
  PARTITION BY RANGE(a)
  (STARTING FROM (1) ENDING(34)IN tbsp1,
  ENDING(67)  IN tbsp2,
  ENDING(100) IN tbsp3)
```

- Notes
  - Special values, MINVALUE, MAXVALUE can be used to specify open ended ranges, eg:

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

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

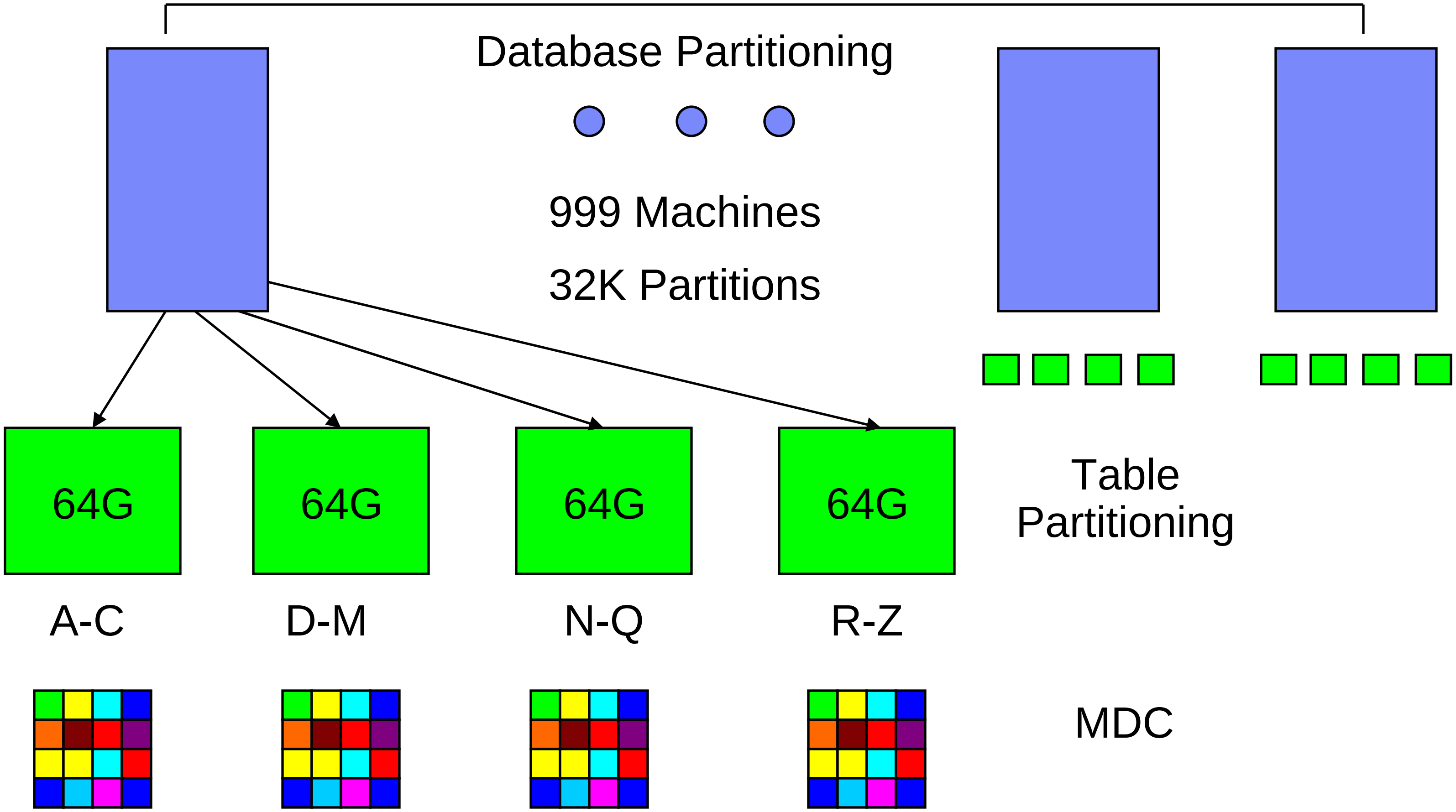


```
CREATE TABLE t1(c1 INT, c2 INT, …)
 IN tbsp1, tbsp2, tbsp3
 INDEX IN tbsp4
 PARTITION BY RANGE(a)
 (STARTING FROM (1) ENDING (100)
   EVERY (33))
CREATE INDEX i1(c1)
CREATE INDEX i2 (c2) IN tbsp5
```

New Operations for Roll-Out and Roll-In

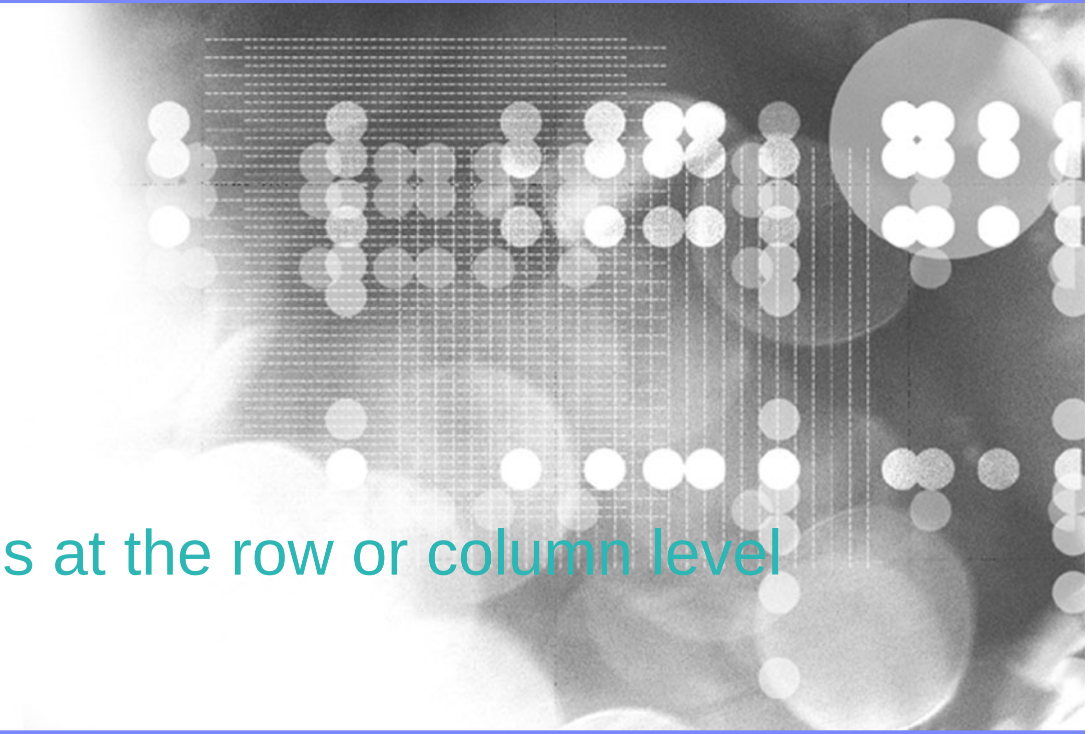
- **ALTER TABLE … DETACH**
  - An existing range is split off as a stand alone table
  - Data instantly becomes invisible
  - Minimal interruption to other queries accessing table
- **ALTER TABLE … ATTACH**
  - Incorporates an existing table as a new range
  - Follow with SET INTEGRITY to validate data 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



DB2 Information Management Software

Granular Security



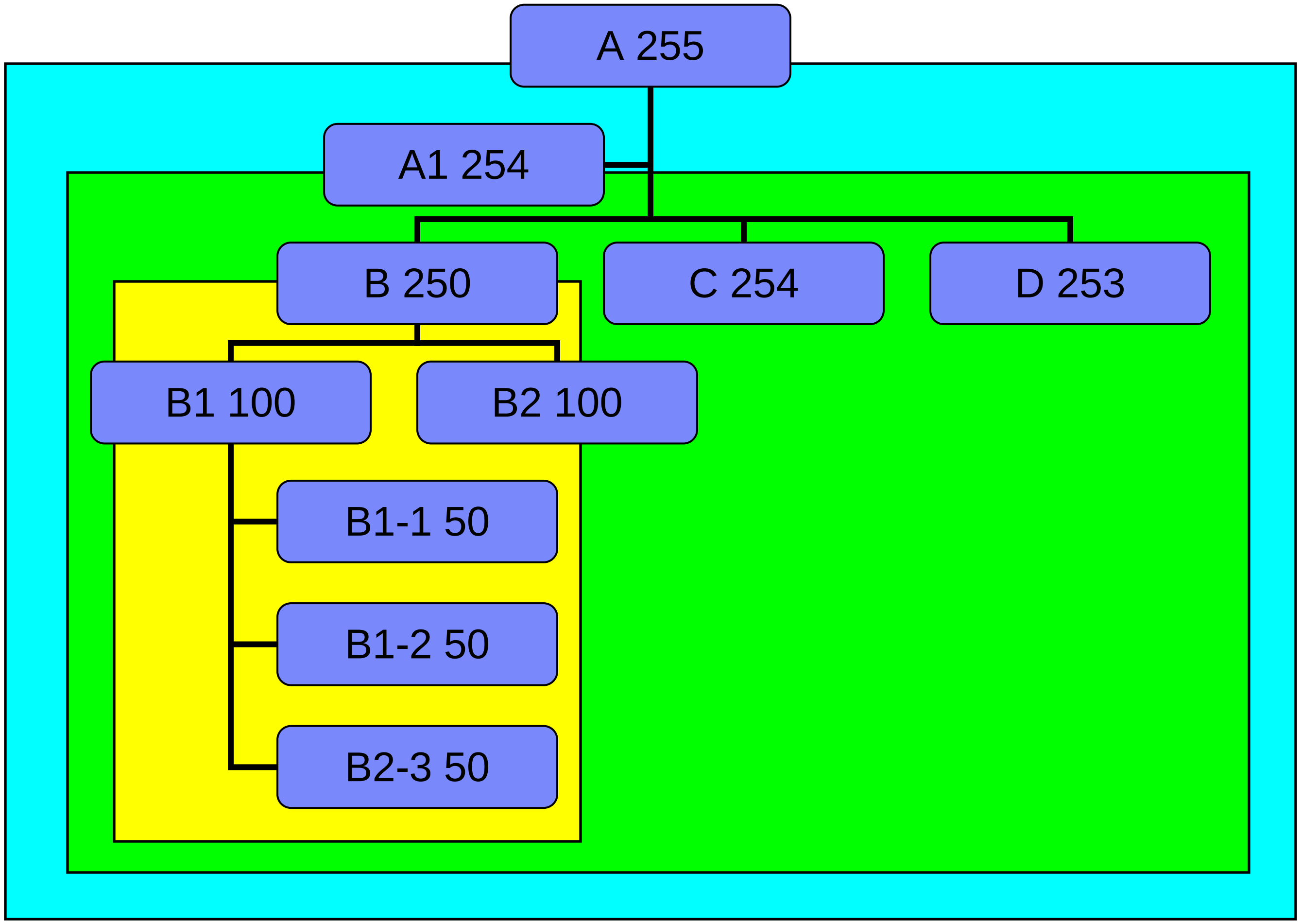
Securing tables at the row or column level

Bari, 20.09.2006

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

DB2 Information Management Software

Table Compression

Saving disk space for large database installations

Bari, 20.09.2006

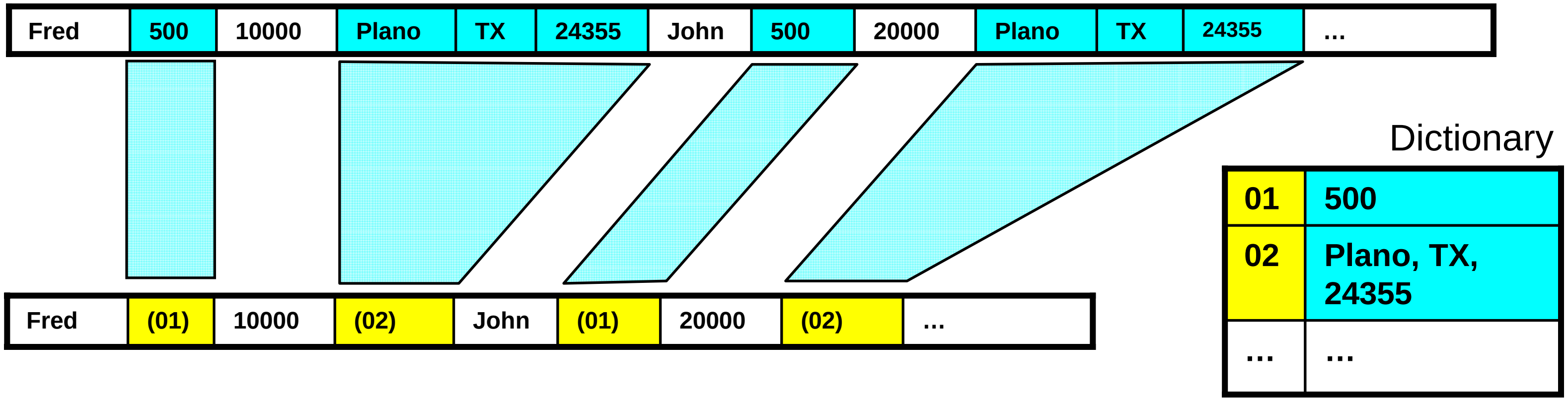
DB2 Compression

- **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

- **Repeating patterns within the data (and just within each row) is the key to good compression. Text data tends to compress well because of reoccurring strings as well as data with lots of repeating characters, leading or trailing blanks**

Name	Dept	Salary	City	State	ZipCode
Fred	500	10000	Plano	TX	24355
John	500	20000	Plano	TX	24355



Fred	500	10000	Plano	TX	24355	John	500	20000	Plano	TX	24355	…

Fred	(01)	10000	(02)	John	(01)	20000	(02)	…

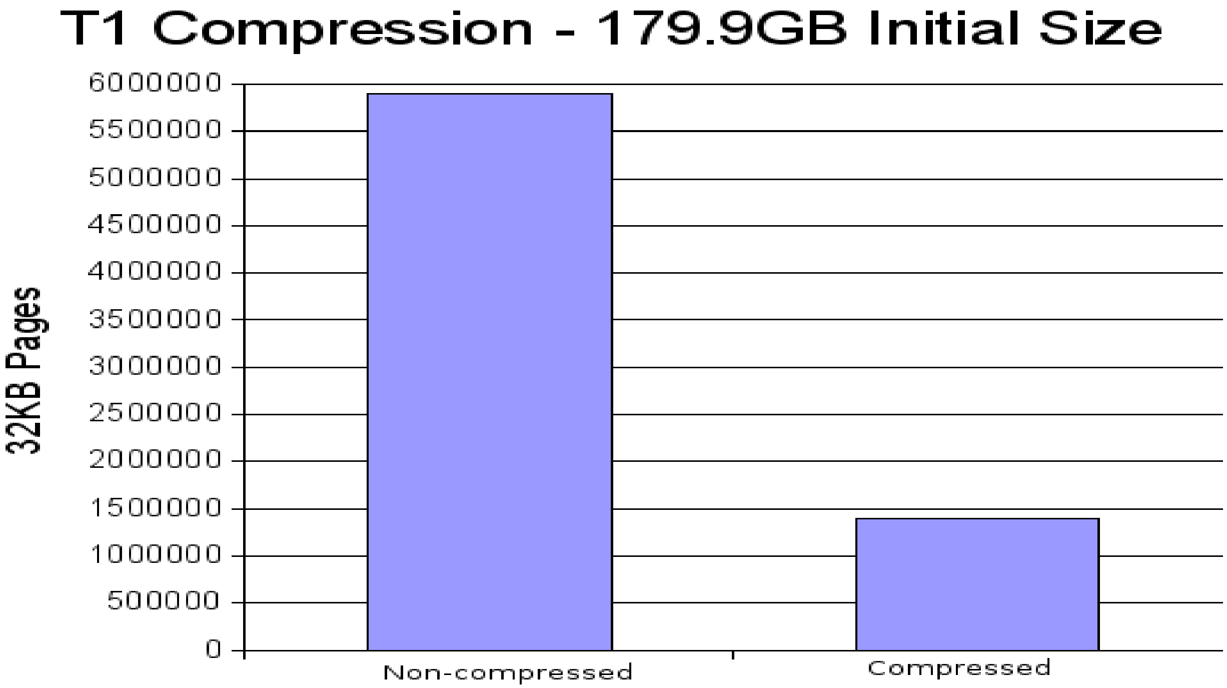
Dictionary

01	500
02	Plano, TX, 24355
…	…

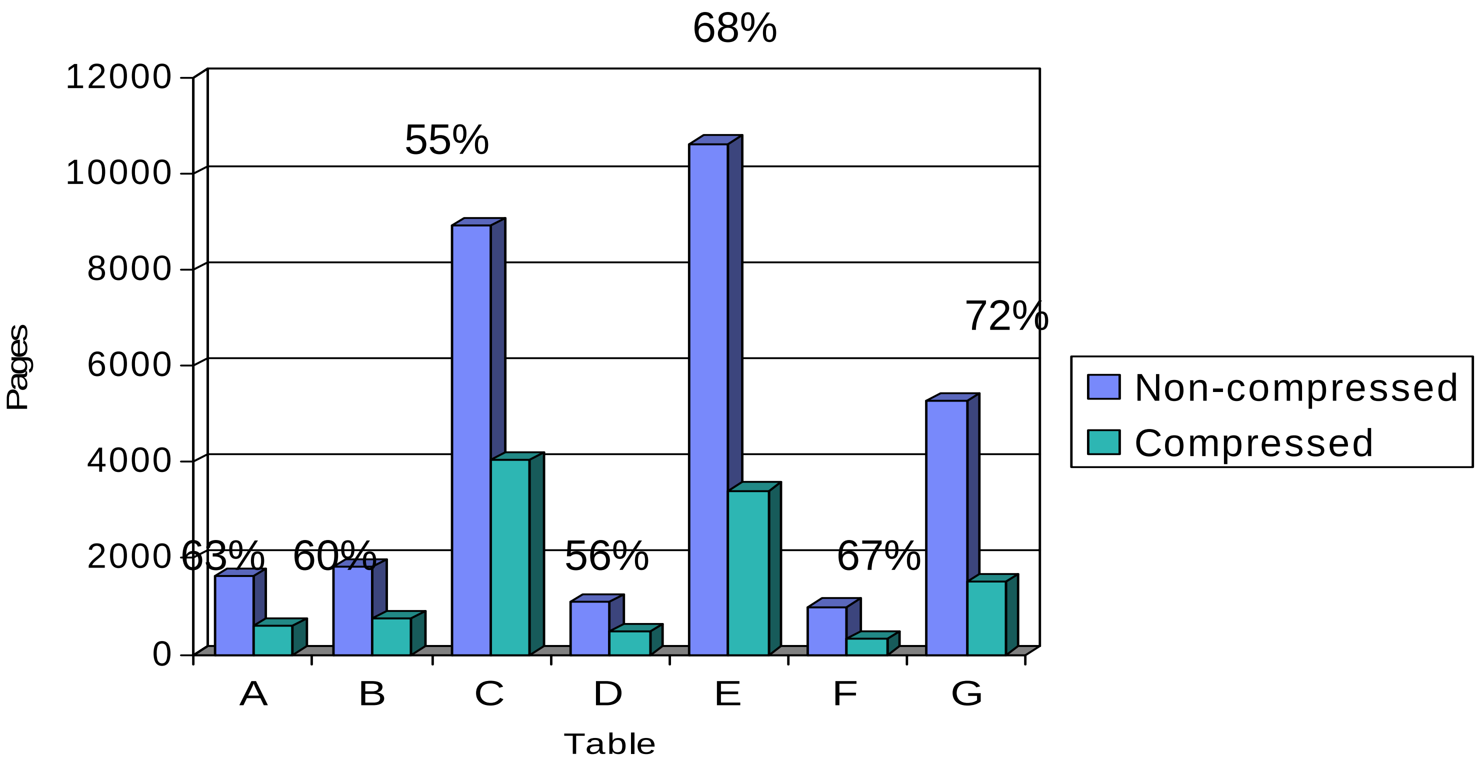
More Compression Ratios (Customer Data)

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



DB2 Information Management Software

Autonomic Features

Reducing the Total Cost of Ownership

Bari, 20.09.2006

Many Autonomic Functions in DB2 V8

- **Configuration Advisor (2 second tuning)**
  - Automatic hardware characteristic detection
  - User description of basic operational questions
  - Combined with mathematical model of each configuration parameter, based on expert heuristics.
- **Design Advisor**
  - Index selection
  - Materialized Query Tables selection (MQT Advisor)
  - Multidimensional Clustering selection (MDC Advisor)
  - Partitioning selection (Partitioning Advisor)
- **Health Monitoring**
- **Utilities:**
  - Online
  - Self-tuning
  - Throttling
  - Automatic with policy
- **Automatic data statistics collection**

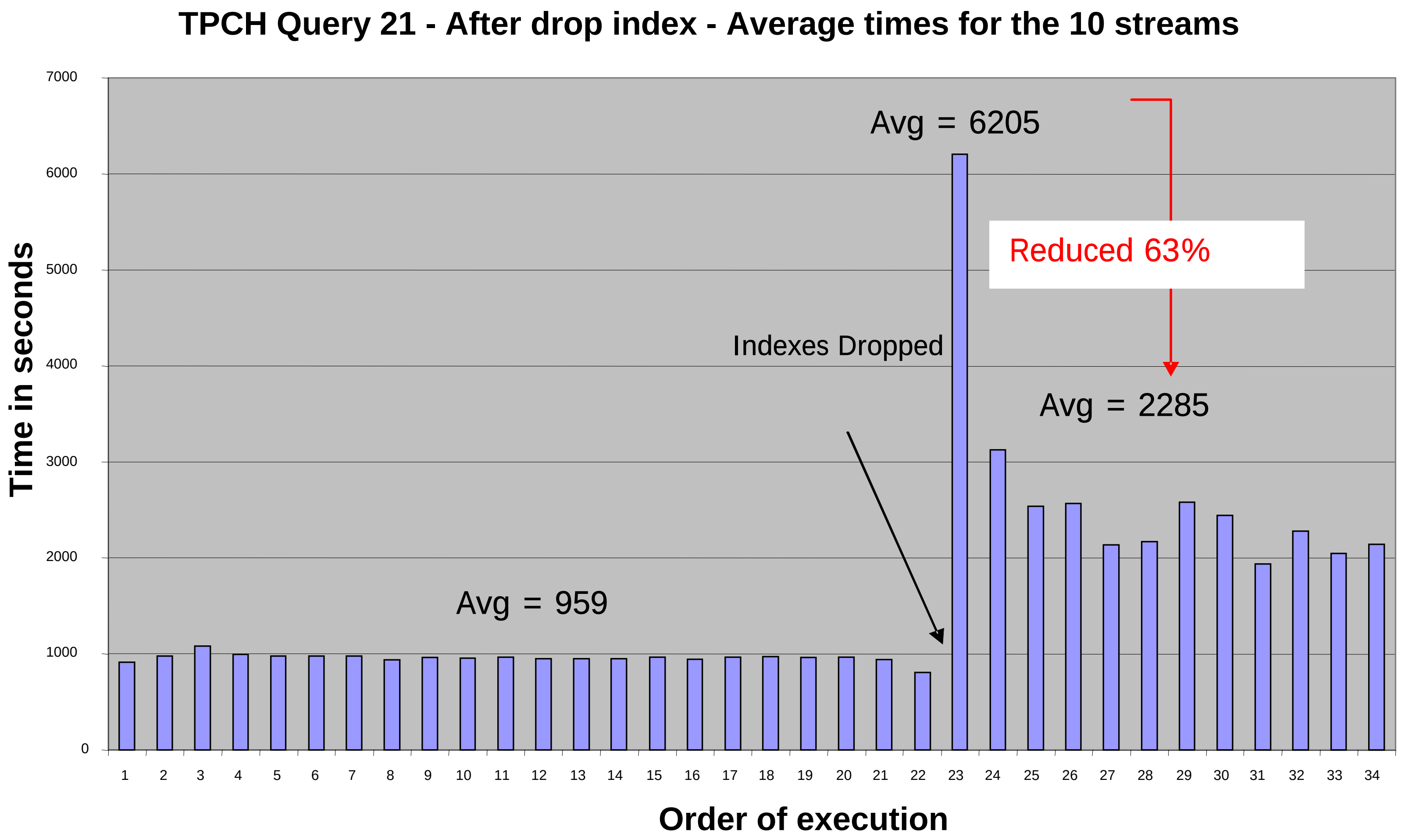
DB2 Simplified Storage Administration

- **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**

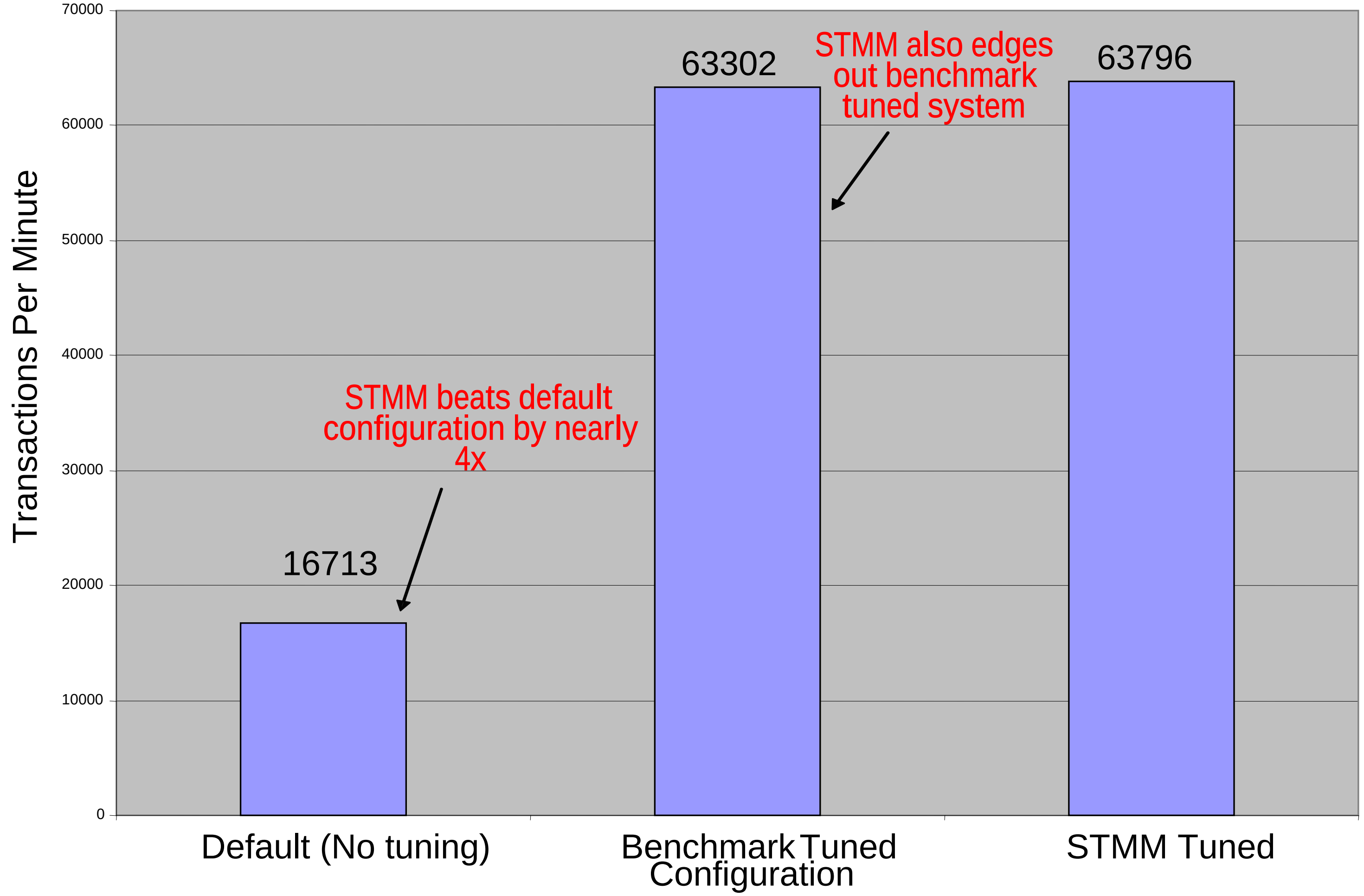
Adaptive Self-Tuning Memory

- **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



STMM beats default configuration by nearly 4x

STMM also edges out benchmark tuned system