



IBM Software Group



## DB2 Technology Preview

**DB2** Data Management Software

***Mats.Mohlin@se.ibm.com***

***Januari 2006***



© 2005 IBM Corporation

Information Management Software

## But first the latest news

- No charge DB2 Express announced

## DB2 Express: News



DB2 developer and user  
community development

**No charge DB2 Express-C now available**

### **New focus on partner and developer community development**

- High value, production, Linux or Windows data server
- No limits – on database size, number of users, instances per server
- A fast path to reliable relational and XML data management
- Free access to Web community forum and content
- Free to embed/redistribute with third party applications and products
- No application change when upgrading to other DB2 editions
- Easy up-sell to DB2 full product and the priced features
- Full product images for every FixPack

## DB2 Express: Understanding the differences

DB2 Express Edition	DB2 Express-C
Aligned to mid-market needs	Aligned to mid-market needs and developers, partners, academic communities considering or using open source databases
Per processor or Per server/users licenses orderable in Passport Advantage	Free download available late January 2006
12mo 24x7 support	Free Web community forum and content
HADR feature available	HADR feature purchase requires DB2 Express Edition purchase
Download footprint 390MB	Download footprint <300 MB
<p>Features <b>include</b>:</p> <p>Warehouse Manager tools &amp; servers, Extender support, DB2 Connect support, Informix Data Source Replication, Replication Data Capture, Web based tools, APPC and Netbios</p>	<p>Features <b>exclude</b>:</p> <p>Warehouse Manager tools &amp; servers, Extender support, DB2 Connect support, Informix Data Source Replication, Replication Data Capture, Web based tools, APPC and Netbios</p>

# Agenda

- Viper technology preview
  - ▶ Native XML data store
  - ▶ Range Partitioning support
  - ▶ Self Tuning Memory Manager
  - ▶ Miscellaneous Viper News



# 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

- What About?
  - ▶ More XML Support
  - ▶ Easier Maintenance
  - ▶ Partitioning
  - ▶ More Security
  - ▶ SQL Features
  - ▶ Database Design



| IBM Software Group

What about...

*More XML Support?*

**DB2** Data Management Software

A horizontal bar with a series of colored squares (green, yellow, red, purple, cyan) followed by a sequence of small, grayscale icons including a refresh symbol, a person's face, and a hand holding a device.

**ON** DEMAND BUSINESS™

© 2005 IBM Corporation

# XML Technology

- 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>
```

# Market Projections

- XML Storage is a high growth area

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

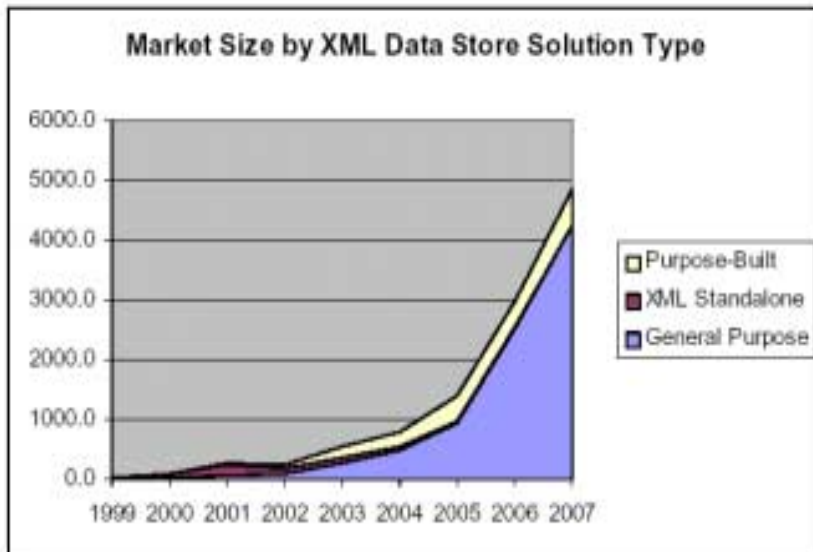
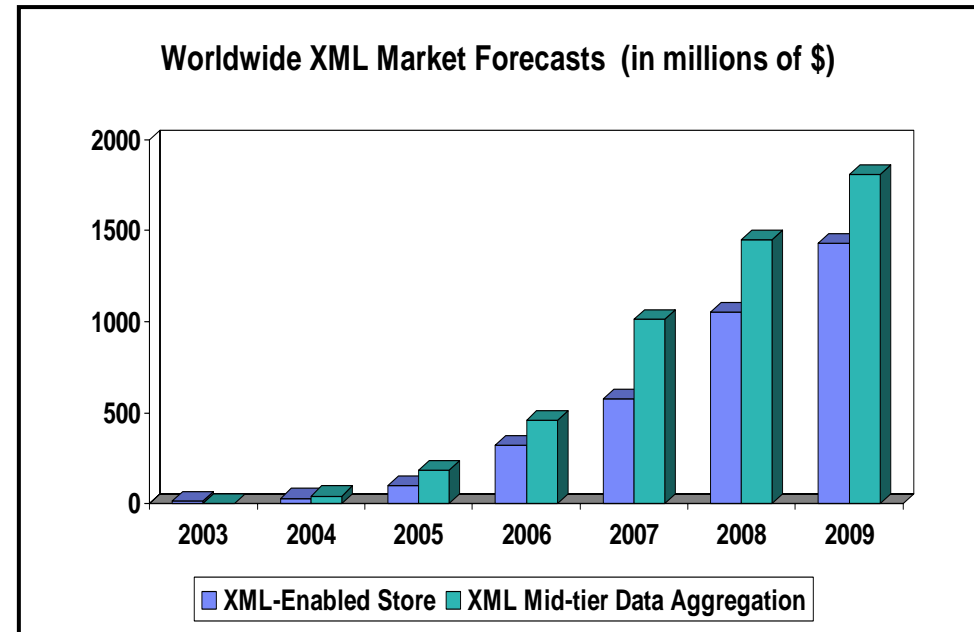


Chart Sources: XML Market Opportunities, Forecasts and Strategies, 2004-2009  
 WnterGreen Research Inc. ZapThink



- 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 – The Change is VERY 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

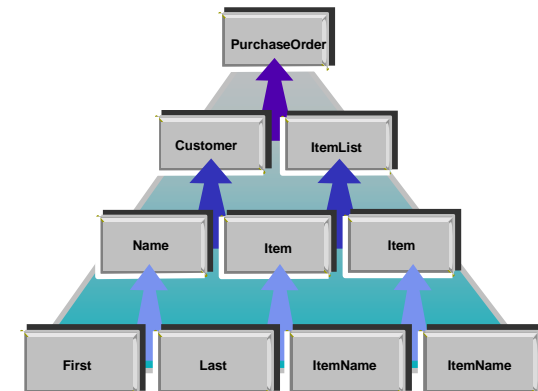
  

Id	LastName	FirstName	Street	City	State	Zip
1	Pirahesh	Hamid	1 Harry Rd	San Jose	CA	95141
3	Selinger	Pat	555 Bailey Ave	San Jose	CA	95141

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

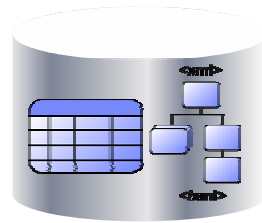
- XML is a data model:
  - Nodes (elements, attributes, comments, etc.)
  - Relationships between nodes
  - Sequence based w/ some sets
  - Flexible schema



## XML in DB2 UDB 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:

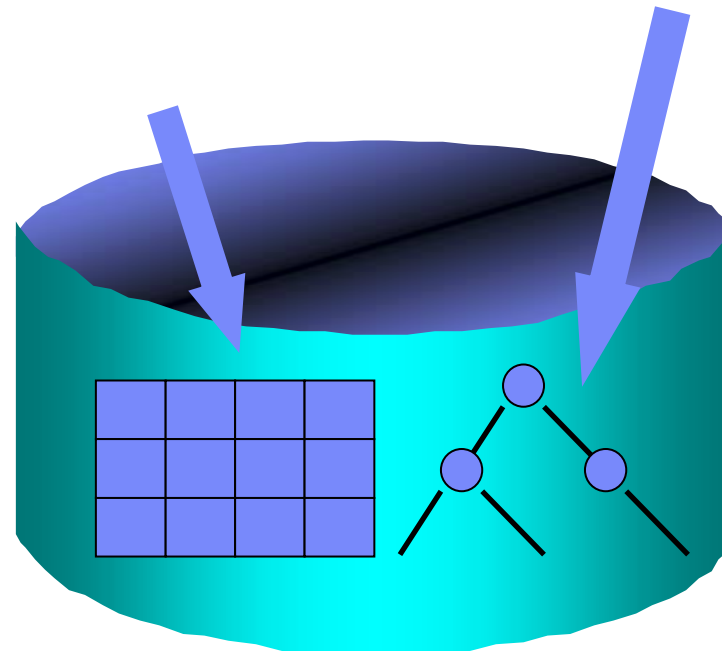
- Ability to seamlessly leverage relational investment
- Proven Infrastructure that provides enterprise-class capabilities

## Native XML Storage

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

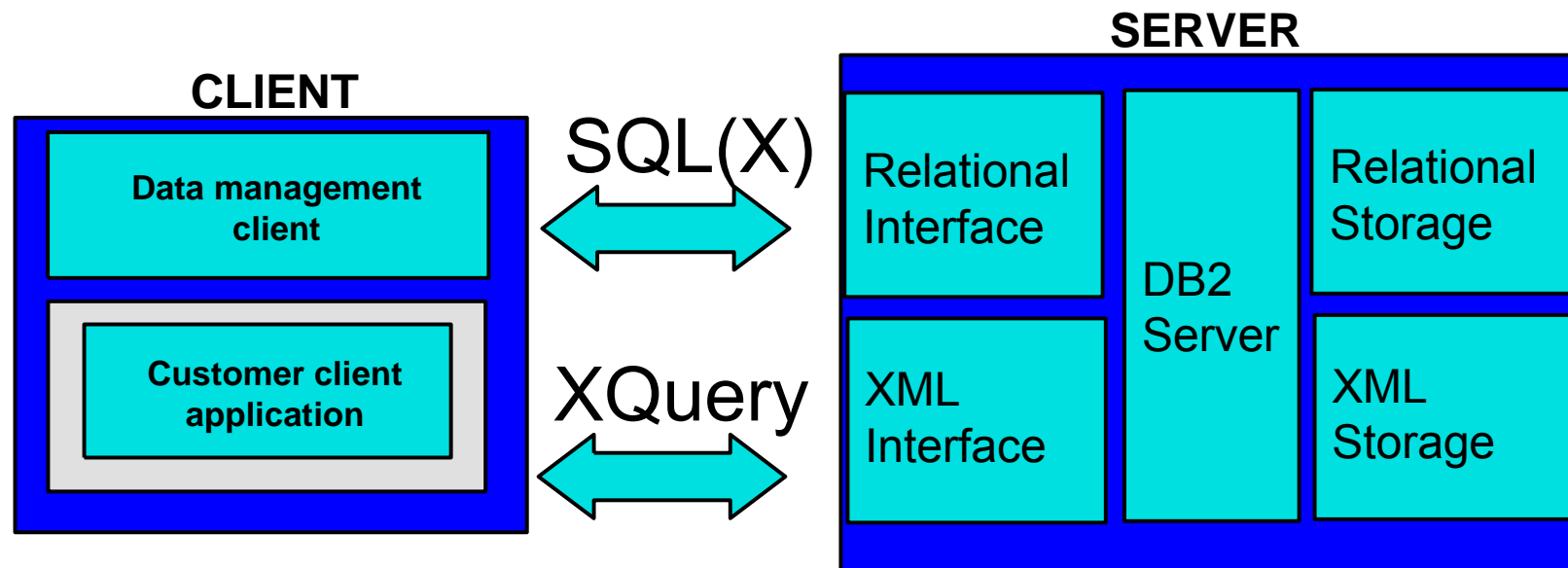
```
create table dept (deptID char(8), ..., deptdoc xml);
```

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



## XML in DB2

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



## Users will see:

- New XML data type for columns  
`create s1.t1 (c1 int, c2 xml)`
- New language bindings for that type  
cobol, c, java, etc..
- New XML indexes  
`create ix1 on s1.t1(c2) generate keys using pattern '/dept/emp/@empno'`
- An XML schema/DTD repository
- Support for XQuery as a primary language as well as:
  - Support for SQL within XQuery
  - Support for XQuery with SQL
  - Support for new SQL/XML functions
- Performance, scale, and everything else they expect from a DBMS





| IBM Software Group

What about...

*Easier maintenance?*

**DB2** Data Management Software

The IBM On Demand Business logo, featuring the word 'ON' in a red square followed by 'DEMAND BUSINESS' in white text on a blue background.

**ON** DEMAND BUSINESS™

© 2005 IBM Corporation

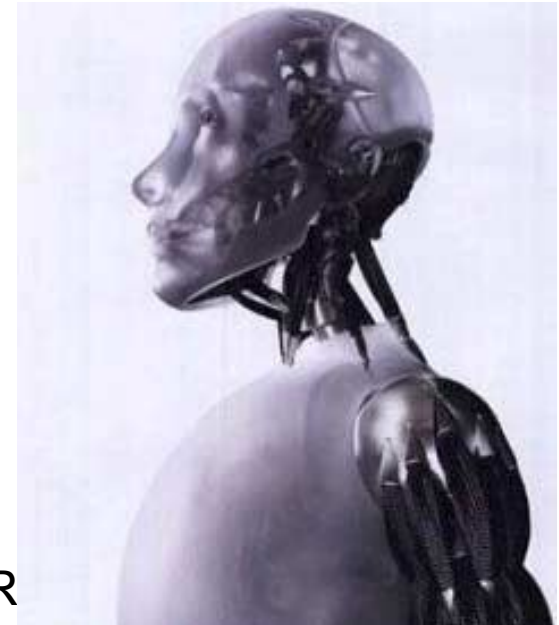
## Installation Thoughts

- Reduce installation complexity
  - ▶ Allow non-admin user to install using Elevated Privileges
  - ▶ 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\_IOCLEANER 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



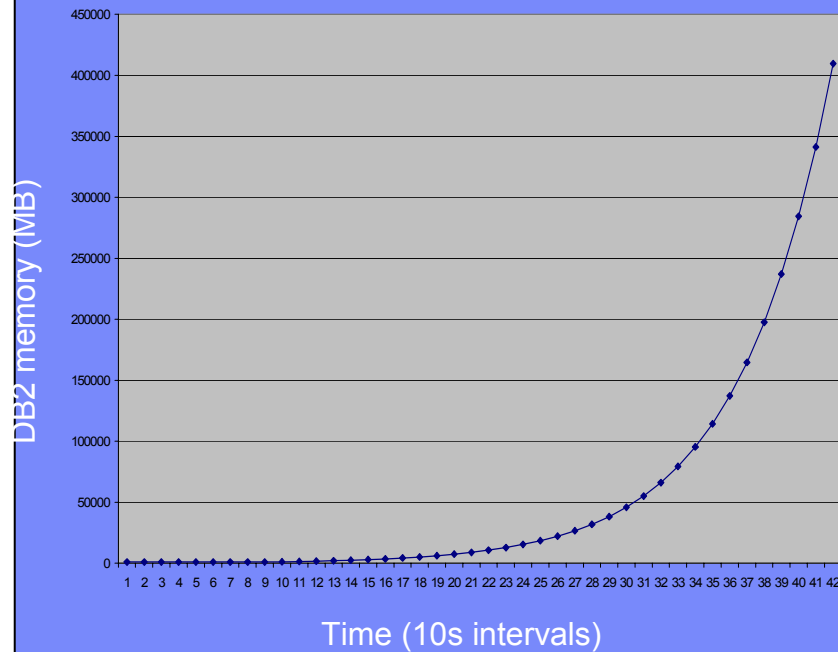
## Adaptive Self-Tuning Memory

- Automatically manage resources to achieve optimal performance
- Manages sort heaps, bufferpools, package cache, lock list size
- Advanced control logic
- Fast convergence and reaction time

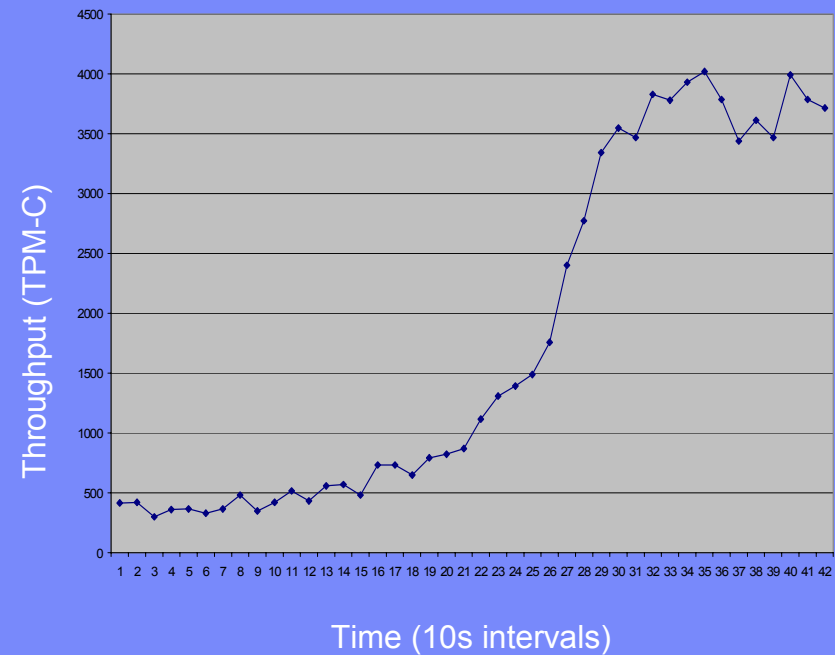


# Adaptive self tuning memory: business value.

Automatically gives more memory to DB2



System performance immediately improves by 10x.



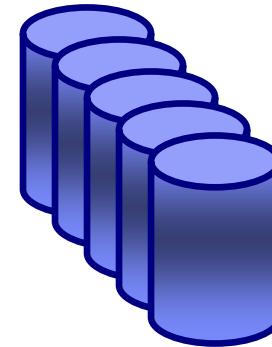
## DB2 Simplified Storage Administration

- 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



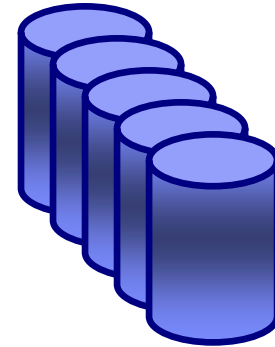
## Automatic Storage Provisioning - Syntax

- ▶ 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).





| IBM Software Group

What about...

***More options for partitioning?***

**DB2** Data Management Software

A horizontal bar with a series of colored squares (green, yellow, red, purple, cyan) followed by a grayscale image strip containing icons of a person, a refresh symbol, and a hand.

**ON DEMAND BUSINESS™**

© 2005 IBM Corporation

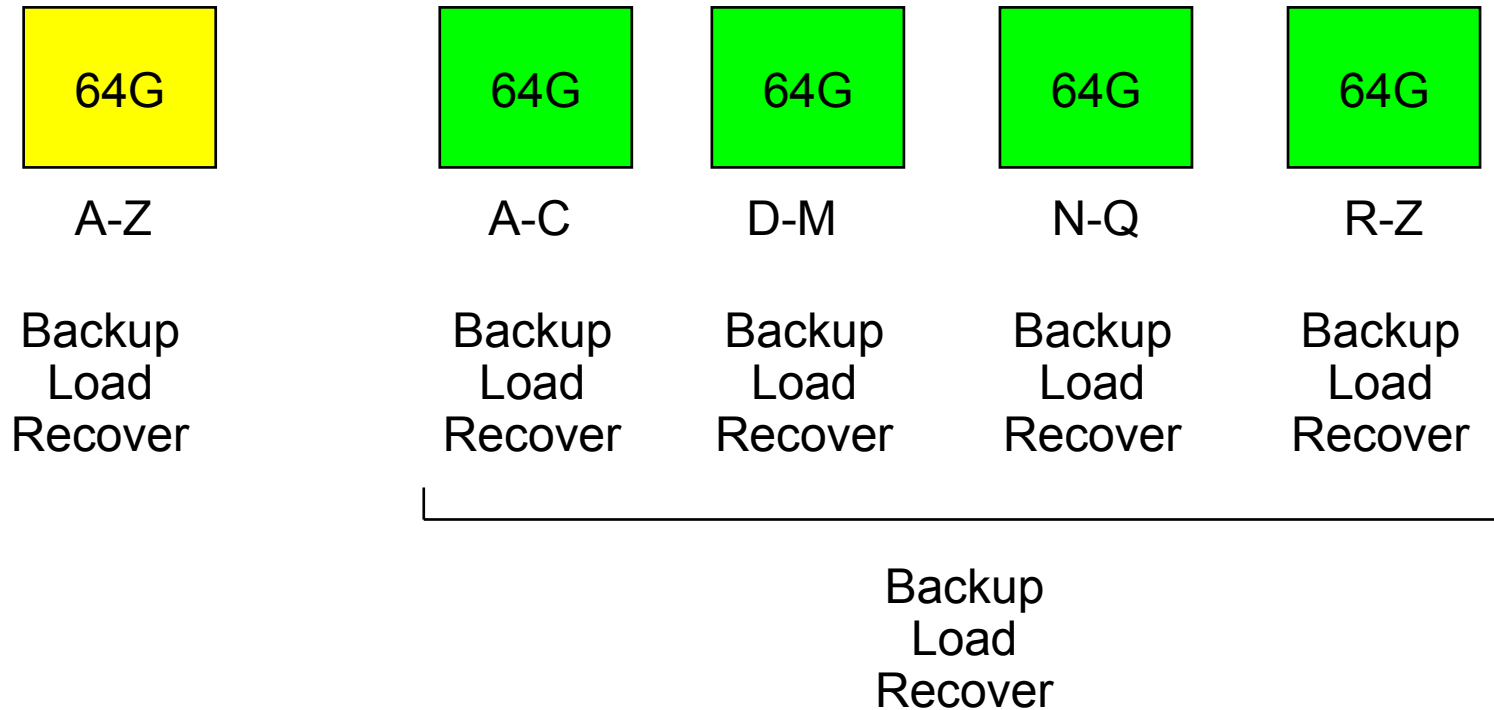
# Range Partitioning Thoughts

- What is Range Partitioning ?
  - ▶ Storing a table in more than one physical object, across one or more tablespaces
  - ▶ Each tablespace 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

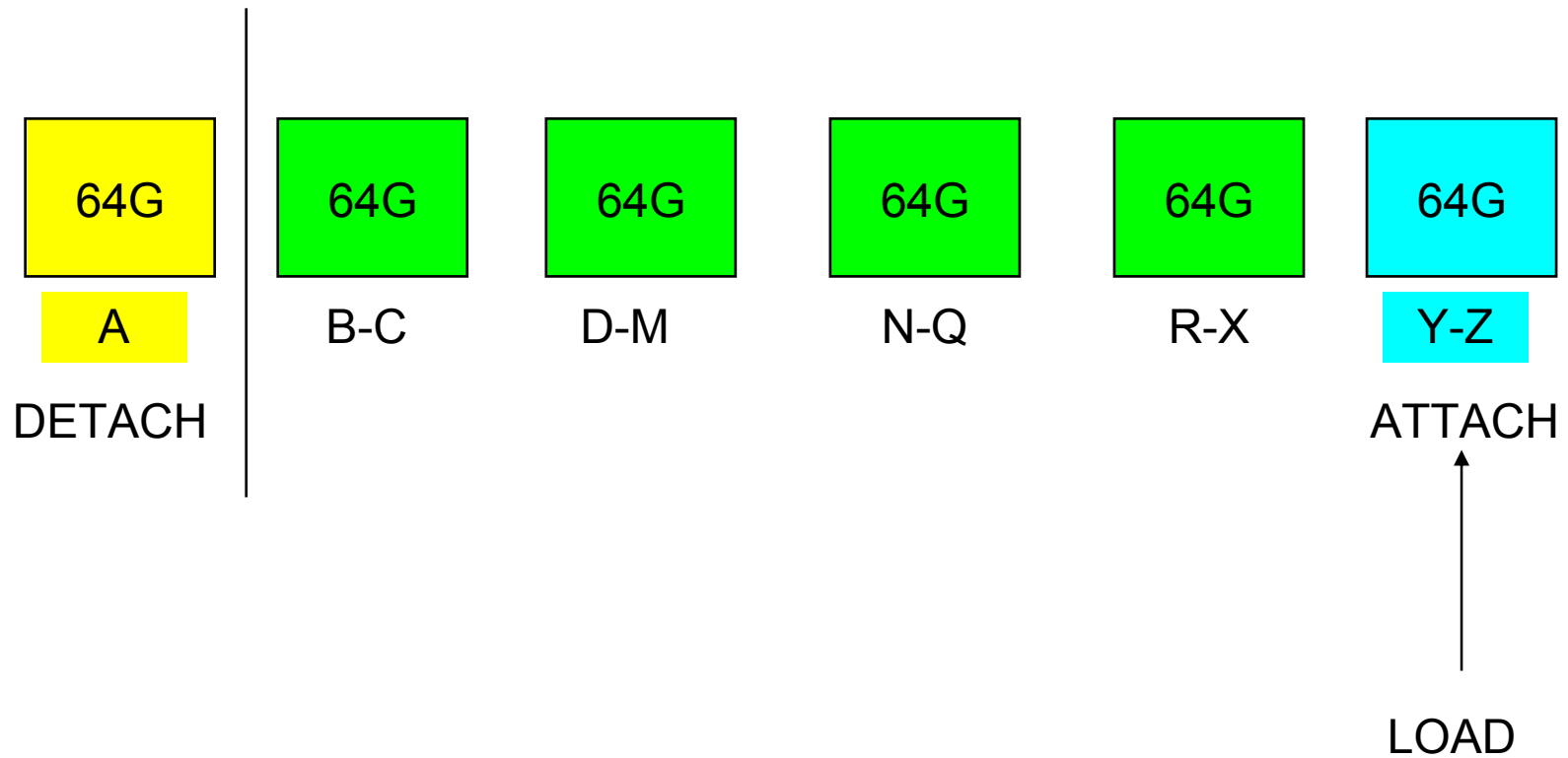


# Range Partitioning

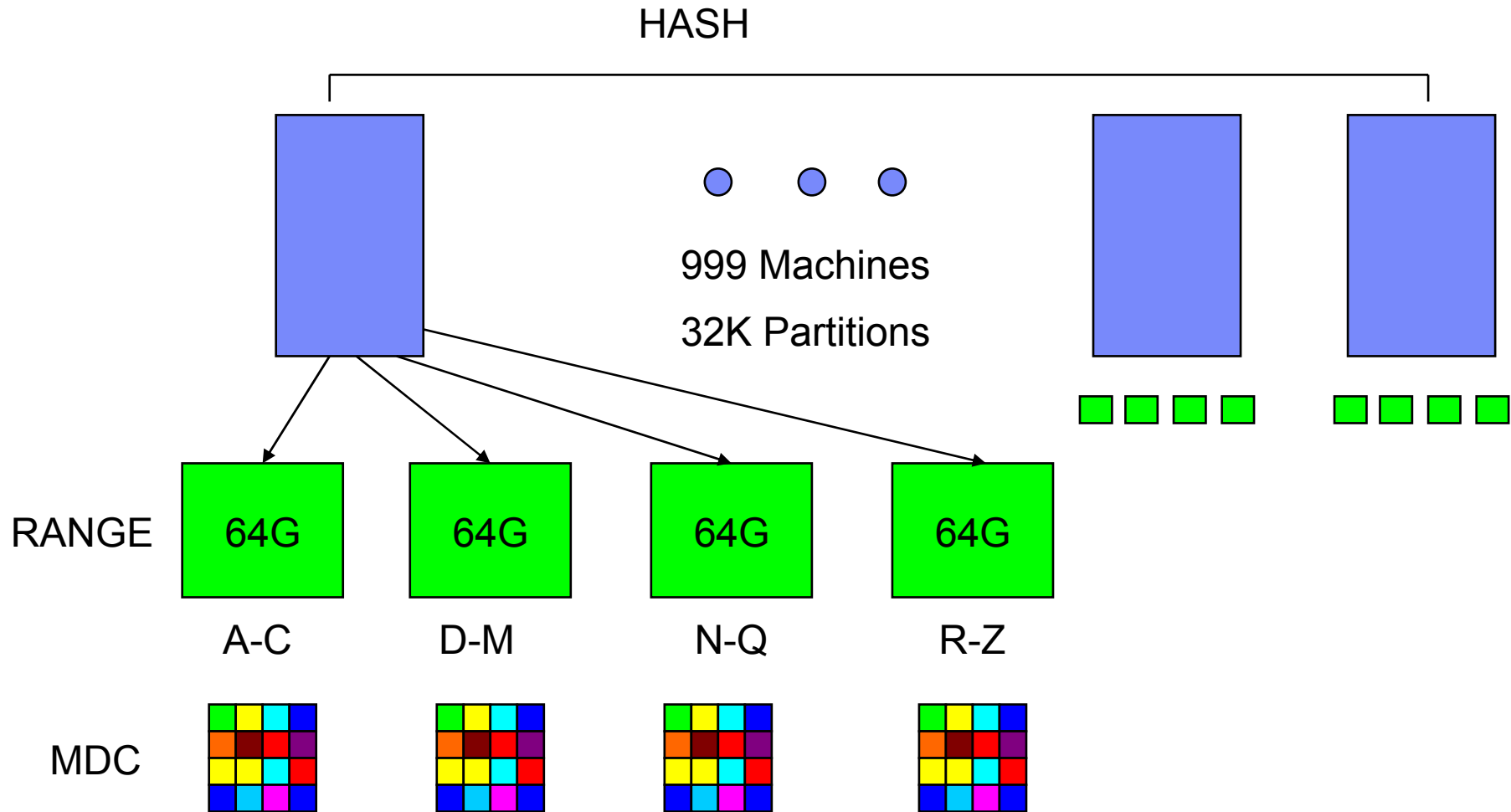
32K Partitions



# Range Partitioning



# Hybrid Partitioning



## Table Partitioning

```
CREATE TABLE orders(... delivery DATE)
  IN tbsp1, tbsp2, tbsp3, tbsp4
  PARTITION BY RANGE(delivery)
  STARTING FROM ('Jan-01-2005')
  EVERY (3 MONTHS) ENDING ('Dec-31-2005')
```



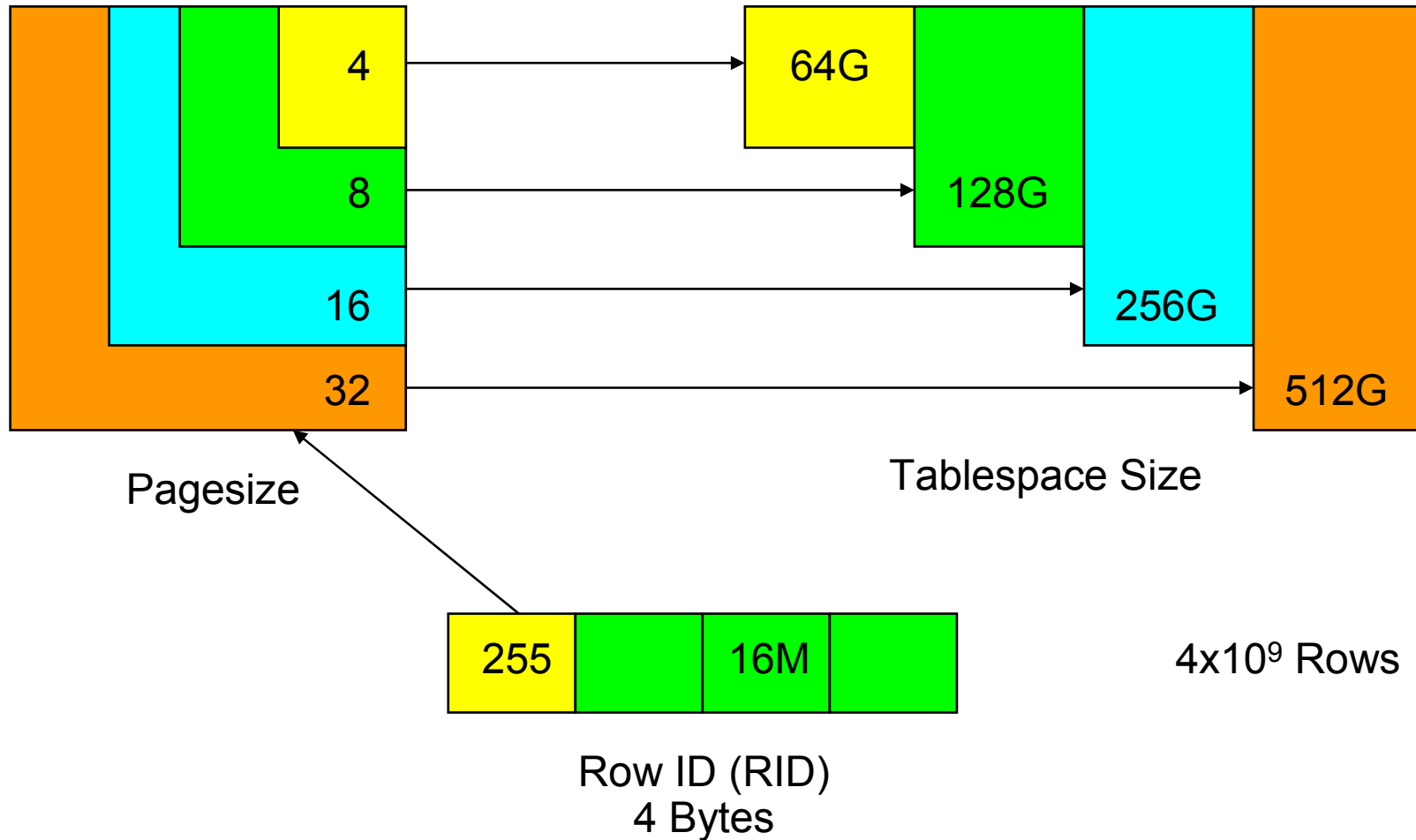
The syntax also supports clauses where specific values can be specified.

## Large Row Identifiers

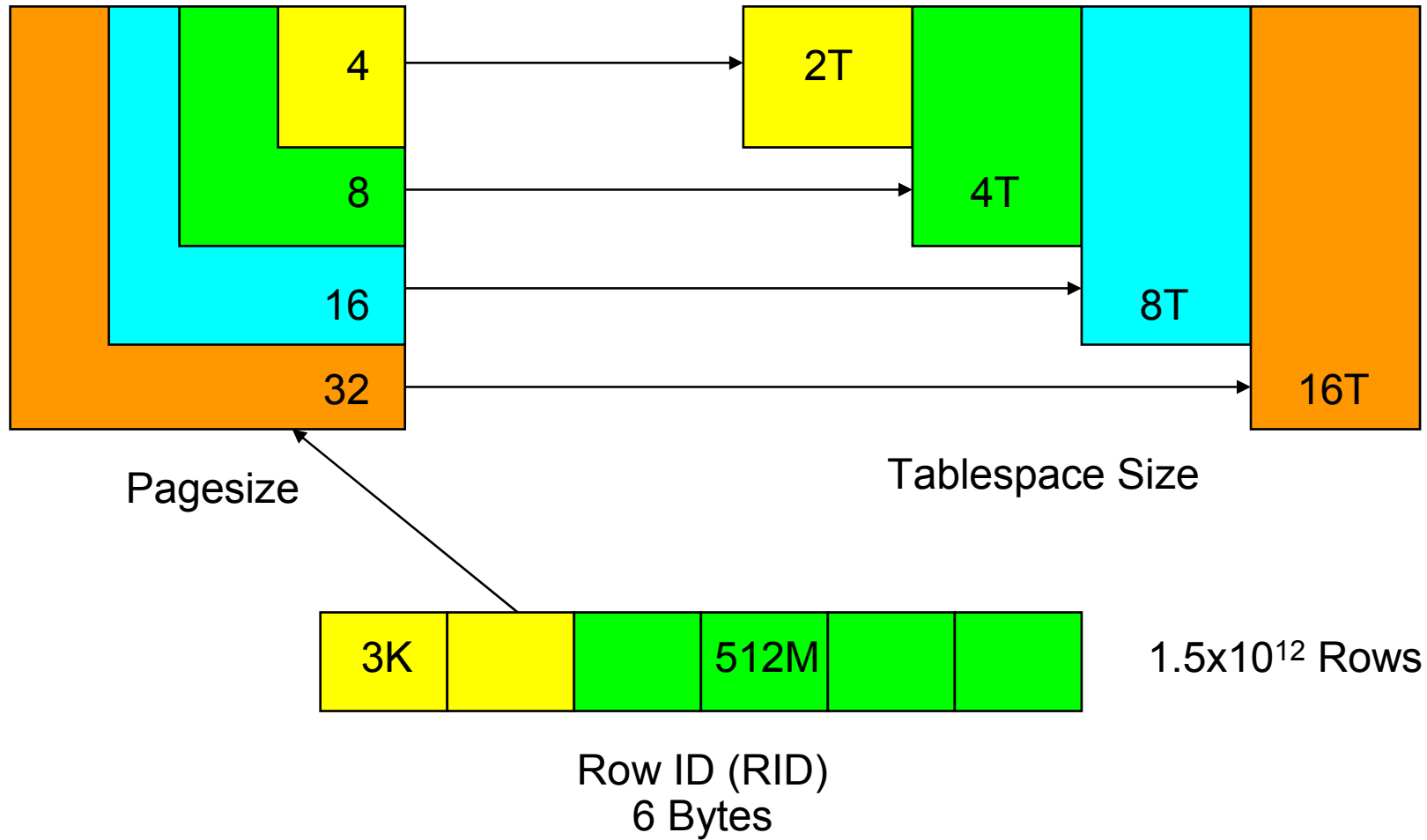
- Increase In table size limits and rows per page
  - ▶ Tablespace level definition
  - ▶ DMS Tablespace only
- ALTER TABLESPACE <name> CONVERT TO LARGE
  - ▶ Tablespace is locked, definition is modified and catalogues are updated
  - ▶ Every index for every table will be marked bad
  - ▶ Indices will have will be rebuilt on first table access



# Current Tablespace Design



# New Tablespace Design





| IBM Software Group

What about...

*Even more security!?*

**DB2** Data Management Software

The logo for IBM On Demand Business, featuring the word 'ON' in a red square followed by 'DEMAND BUSINESS' in white text on a blue background.

**ON** DEMAND BUSINESS™

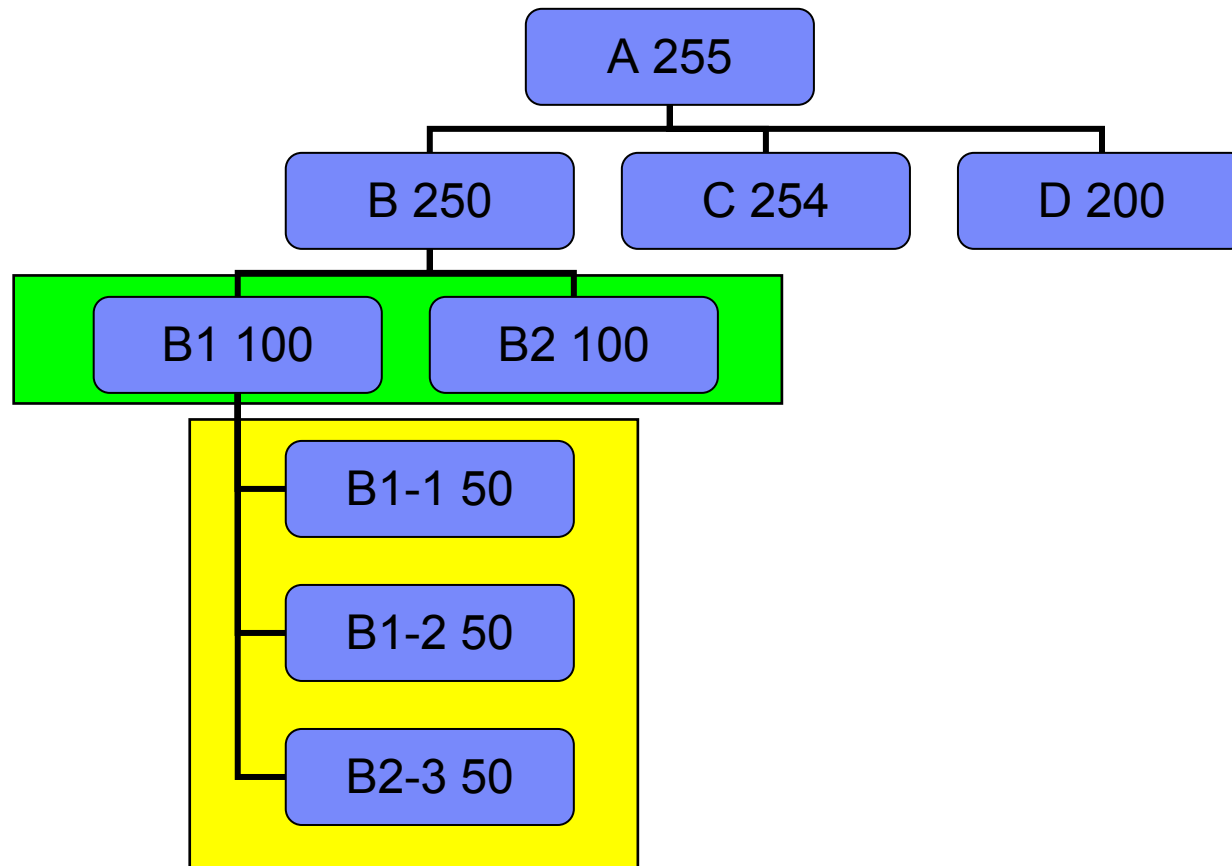
© 2005 IBM Corporation

## Security - Label Based Access Control

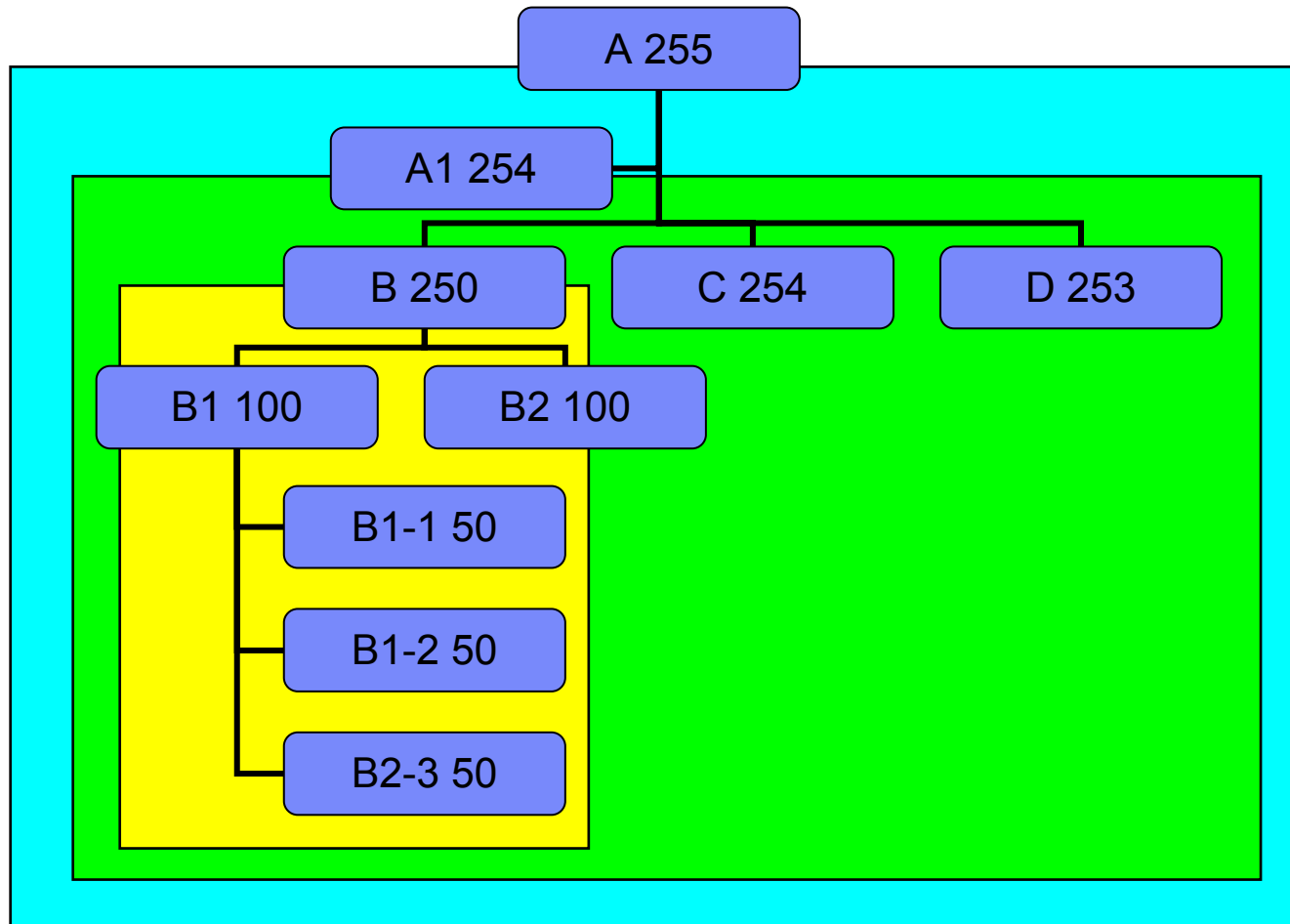
- Label Based Access Control (LBAC)
  - ▶ A “label” is associated with both user sessions and data rows
  - ▶ Rules for comparing users and data labels provide allow access controls to be applied at the row level
  
- Labels may consist of multiple components
  - ▶ Hierarchical, group or tree types
  - ▶ 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 – Update



# LBAC Hierarchy – Read



# LBAC Query

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

User Level = 100

No LBAC	LBAC	ID	SALARY
Red		255	60000
Red	Green	100	50000
Red	Green	50	70000
		50	45000
		60	30000
Red		250	56000
Red		102	82000
Red	Green	100	54000
		75	33000
		253	46000
Red	Green	90	83000
Red		200	78000
		105	45000



| IBM Software Group

What about...

***SQL features?***

**DB2** Data Management Software

A horizontal bar with a series of colored squares (green, yellow, red, purple, cyan) followed by a grayscale image strip containing icons of a hand holding a pen, a person's face, and a hand holding a device.

**ON DEMAND BUSINESS™**

© 2005 IBM Corporation

## How about Less Limits?

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



## 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  $(A \text{ IS NULL AND } B \text{ IS NULL})$  predicates



| IBM Software Group

What about...

*Database Design?*

**DB2** Data Management Software

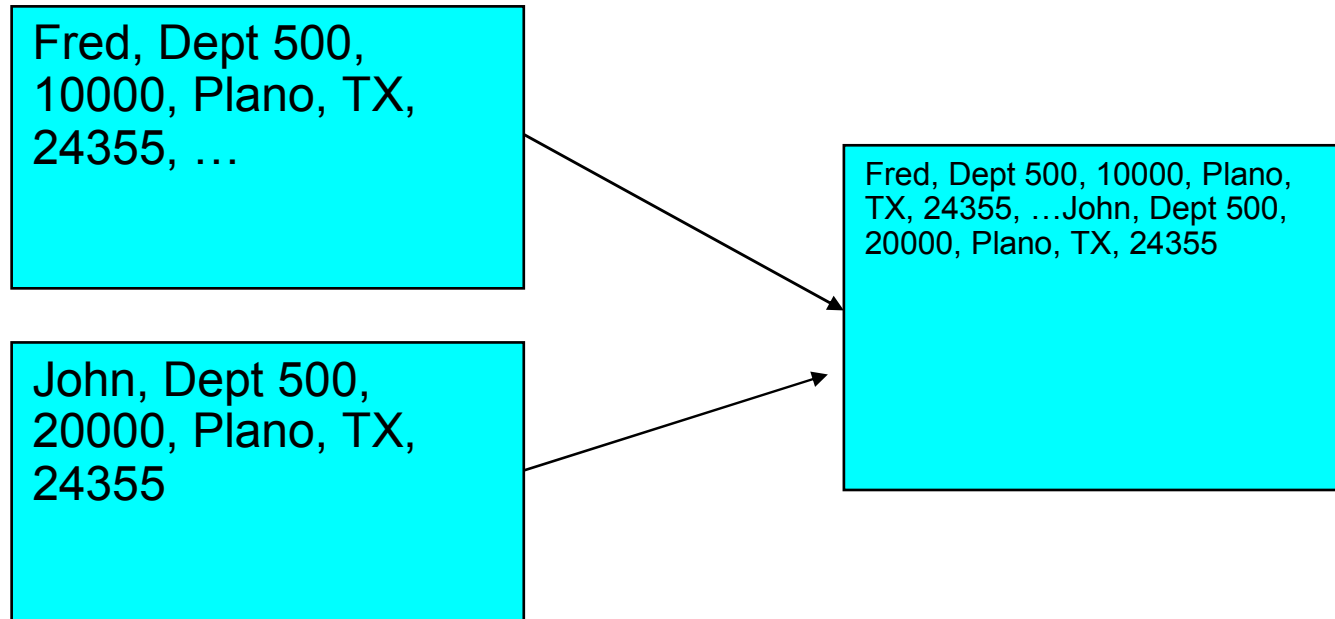
A horizontal decorative bar spanning the width of the slide, featuring a series of small, semi-transparent icons including a green square, a yellow square, a red square, a purple square, a cyan square, a person's face, a circular arrow, and a hand holding a pen.

**ON** DEMAND BUSINESS™

© 2005 IBM Corporation

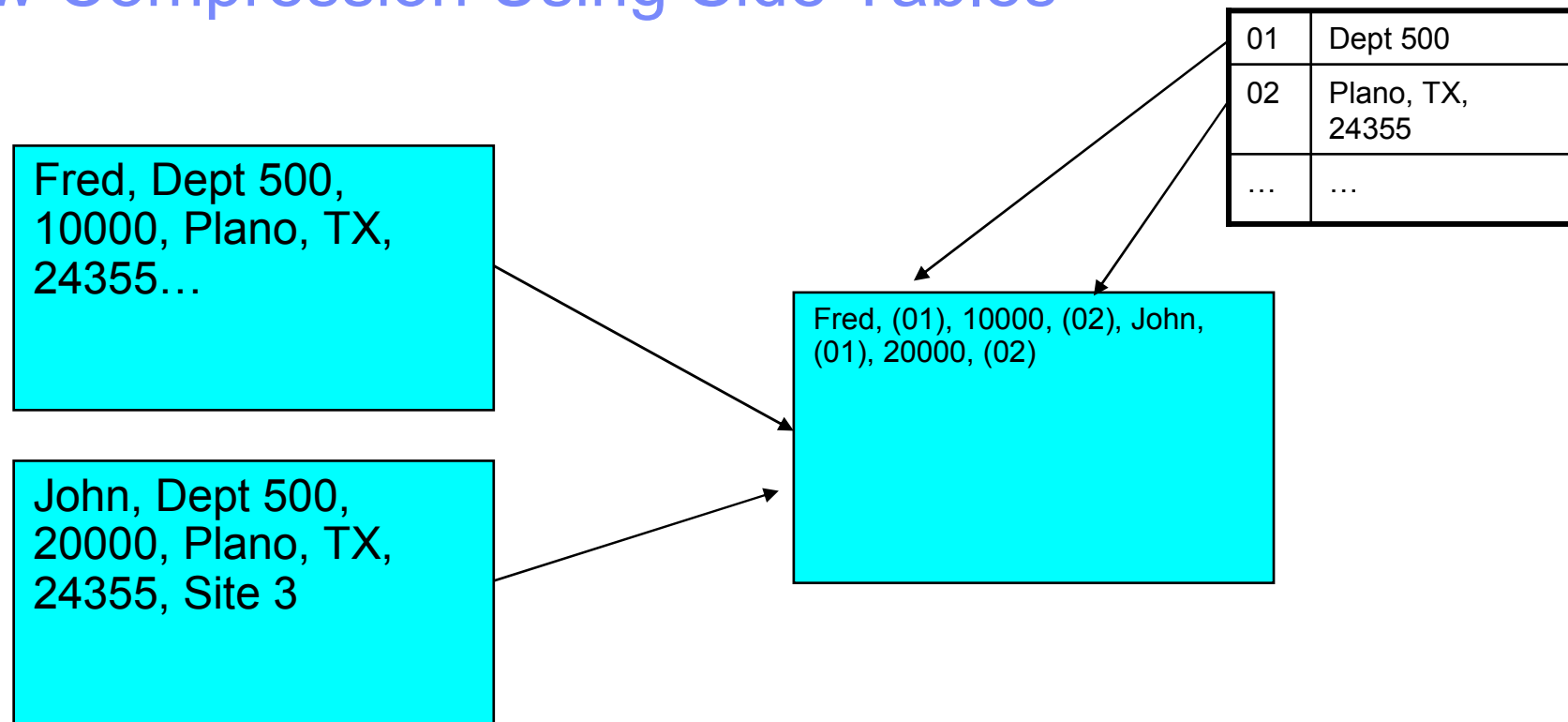


## Row Compression using LZV Algorithms



Database algorithms are based on Row IDs which are based on page boundaries. Compressing multiple rows into one page is not efficient from a database perspective.

# Row Compression Using Side Tables



Side tables contain repeated information from the rows.

## Some more details....?

- XML
- Range Partitioning
- STMM.



# DB2 Viper XML Why Do I care?



## What is XML?

- XML: eXtensible Markup Language
  - ▶ It is a Structured Document Definition
- It Represents a Hierarchy of Elements
- Two Main Types of XML Documents
  - ▶ Document and Data
- Two Ways to Process XML Documents
  - ▶ Store documents whole
  - ▶ Explode the XML Document into Columns
- Electronic Document Exchange (Data Interchange)

# XML – a practical view

- Relational is a data model:

- Relations (tables)
- Attributes (columns)
- Set based w/ some sequences
- Strict schema

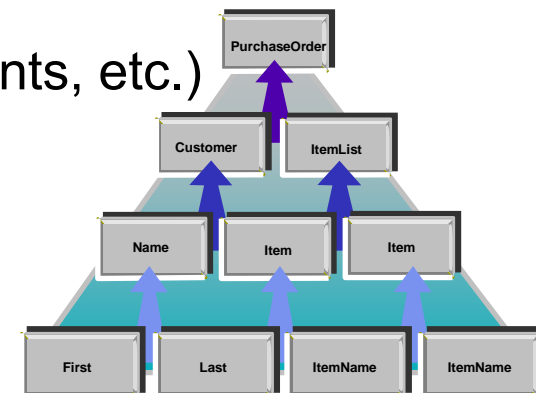
POID	CustomerID	ItemID
12	1	2
162	3	4
162	3	5

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



# Why do we need XML ?

## Hot Dog with Onions



### Ingredients:

1 Hot Dog	1 Hot Dog Bun
1 Tablespoon Sliced Onion	2 Teaspoons Mustard
2 Teaspoons Relish	

### Preparation Steps

Preheat Grill to 350' F.  
 Grill Hot dog for 5 Minutes, Turning frequently to avoid burning  
 Put Hot Dog on bun, add mustard, onion, and relish

### Comments

Some like to use spicy mustard or replace mustard with catsup.

### Nutrition Information

300 calories, 18.5 grams fat,  
 12 grams carbohydrates 9.5 grams protein

**Time To Prepare:** 5 Minutes

**Difficulty:** Easy

Serves: 1

- Relational is more structured, but very fast.
- XML is very flexible.

## Create Table Recipe (

Recipeld integer Primary Key,  
 Title varchar(64),  
 Calories integer,  
 FatGrams integer,  
 CarboGrams integer,  
 ProteinGrams integer,  
 TimeToPrepare integer,  
 Difficulty varchar(12) );

Create Table Ingredients...

Create Table PreparationSteps...

Create Table Comments....

Create Index TitleIX on Recipe(Title)....

Create Index Calorielx on Recipe(Calories)

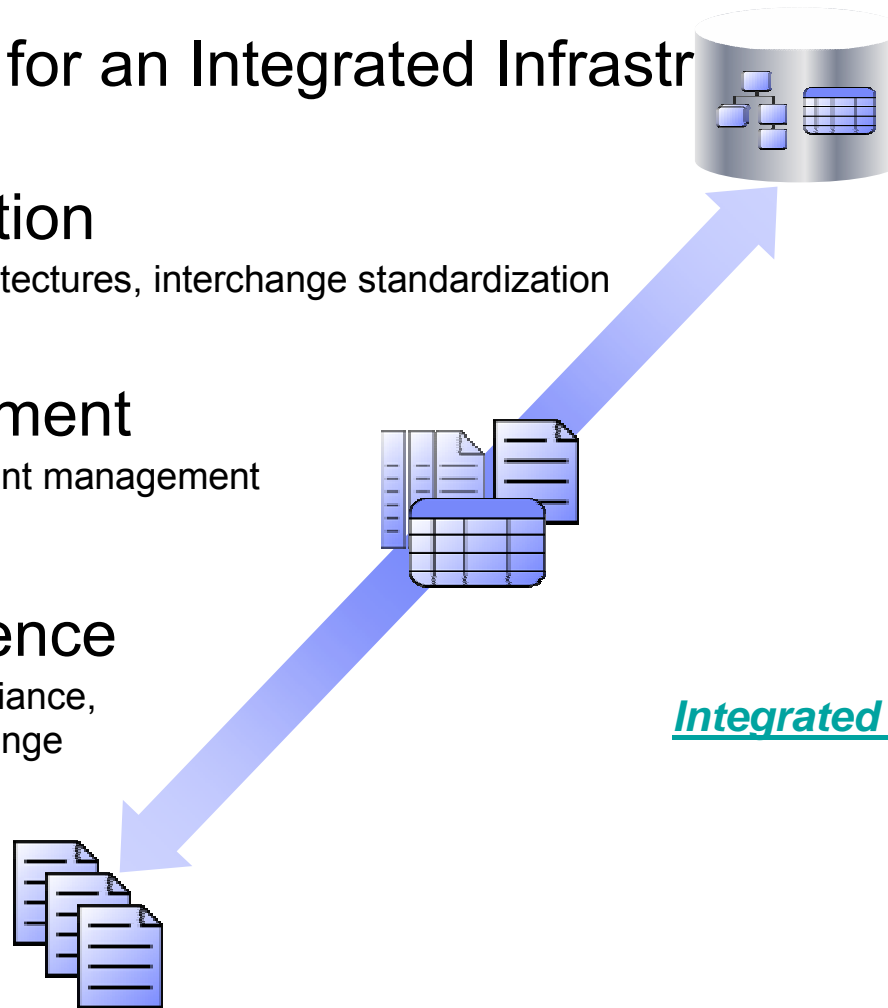
```
<?xml version="1.0" encoding="UTF-8" ?>
<Recipe TimeToPrepare="5" CookMethod="Grill" Difficulty="Easy" Serves="1" Category="Entrees">
  <Title>Hot Dog with Onions</Title>
  <Ingredients>
    <Ingredient Name="Hot Dog" Amount="1" />
    <Ingredient Name="Hot Dog Bun" Amount="1" />
    <Ingredient Name="Sliced Onion" Amount="1" Unit="tablespoon" />
    <Ingredient Name="Mustard" Amount="2" Unit="teaspoon" />
    <Ingredient Name="Relish" Amount="2" Unit="teaspoon" />
  </Ingredients>
  <Preparation>
    <Step>Preheat Grill to 350&#176; F. </Step>
    <!-- &#176; is the degree symbol -->
    <Step>Grill Hot dog for 5 Minutes, Turning frequently to avoid burning</Step>
    <Step>Put Hot Dog on bun, add mustard, onion, and relish</Step>
  </Preparation>
  <Comment>Some like to use spicy mustard,or replace mustard with catsup</Comment>
  <Nutrition>
    <Calories>300</Calories>
    <FatGrams>18.5</FatGrams>
    <CarboGrams>12</CarboGrams>
    <ProteinGrams>9.5</ProteinGrams>
  </Nutrition>
</Recipe>
```

## XML is "on-demand":

- No need to predefine schema means fast setup and use.
- No rigid structure means fast and easy change.

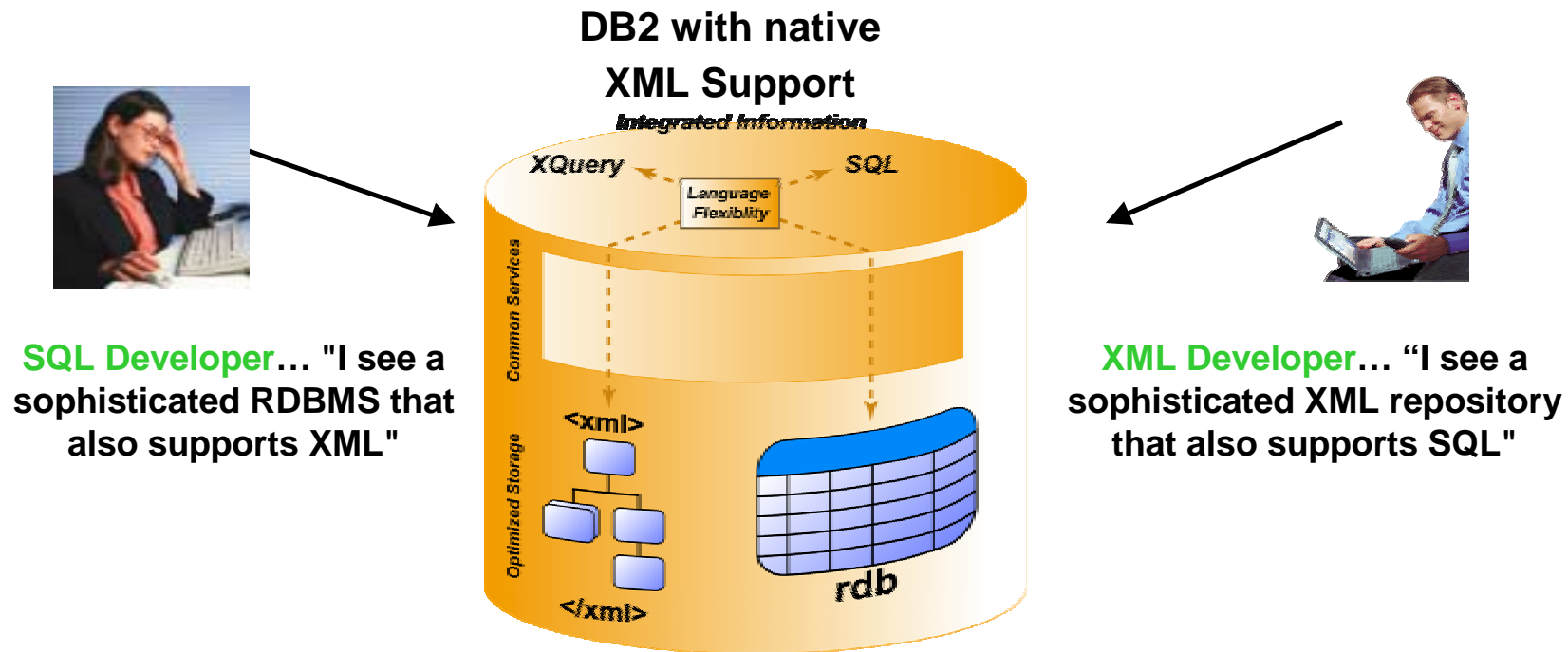
# XML and Information Management Evolution

- XML is Essential for an Integrated Infrastructure
- Business Integration
  - ▶ Service-Oriented Architectures, interchange standardization
- Content Management
  - E-forms, document management and access +++
- Business Intelligence
  - Regulatory compliance, information exchange



# XML in DB2

The most significant feature of the DB2 Viper release  
**Taking DB2 into new markets**

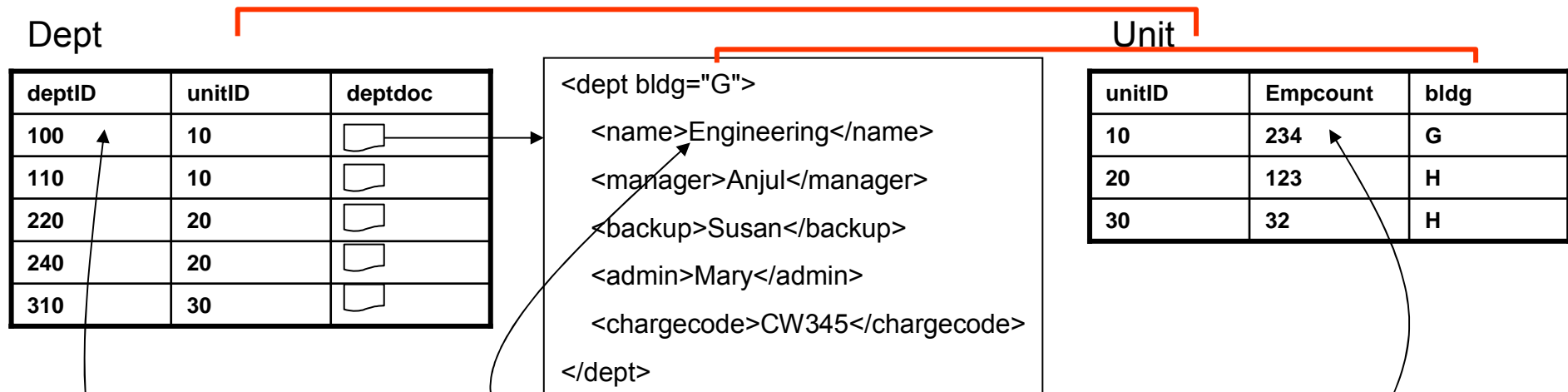


**XML integrated in all facets of DB2!**

## Why DB2 Native XML?

- **More and more information is being managed using XML**
  - Web Services & Information Integration
  - Workflow and Business Processes
  - XML content
  
- **A first class storage option hasn't been available, until now**
  - LOB - Fast and efficient document storage, but the content is locked up
  - Shred - opens up the content but storage management is complex and fragile
  - Pure XML database – Creates XML silos and doesn't scale
  
- **DB2 Native XML**
  - Fast, efficient robust document storage
  - Sophisticated access to the content
  - Integrated with existing infrastructure
  - Builds on the tried and trusted DB2 platform offering enterprise scalability and reliability

# SQL/XML



```

select d.deptID, xmlquery('$deptdoc/dept/name' passing d.deptdoc as "deptdoc"), u.empcount
from dept d, unit u
where d.unitID = u.unitID and u.empcount > 200
and xmlcast(xmlquery('$deptdoc/dept/@bldg' passing d.deptdoc as "deptdoc") as int) = u.bldg
and xmlexists('$deptdoc/dept/name' passing d.deptdoc as "deptdoc")

```

## DB2 Viper at Storebrand



- **Leading player in pensions, life and health insurance, banking and asset management in Norway.**
- **Early adopter of SOA, Web Services and XML**
- **Senior Enterprise Architect Thore Thomassen Worked closely with IBM through the Viper Alpha**
- **Distinct advantages for native XML support in comparative tests**
  - **Reduced the time it took to generate an internal report from more than 1 day to less than 10 minutes.**
  - **Cut the I/O portions of select Web services an average of 65% and maintenance time by 20%.**
  - **Implemented schema changes in a few minutes instead of a full day to prototype and test and a full week to deploy.**
  - **Implemented six new database search and retrieval requirements using native XML features in 30 minutes, compared to 2 hours with decomposition and 8 hours with CLOBs.**

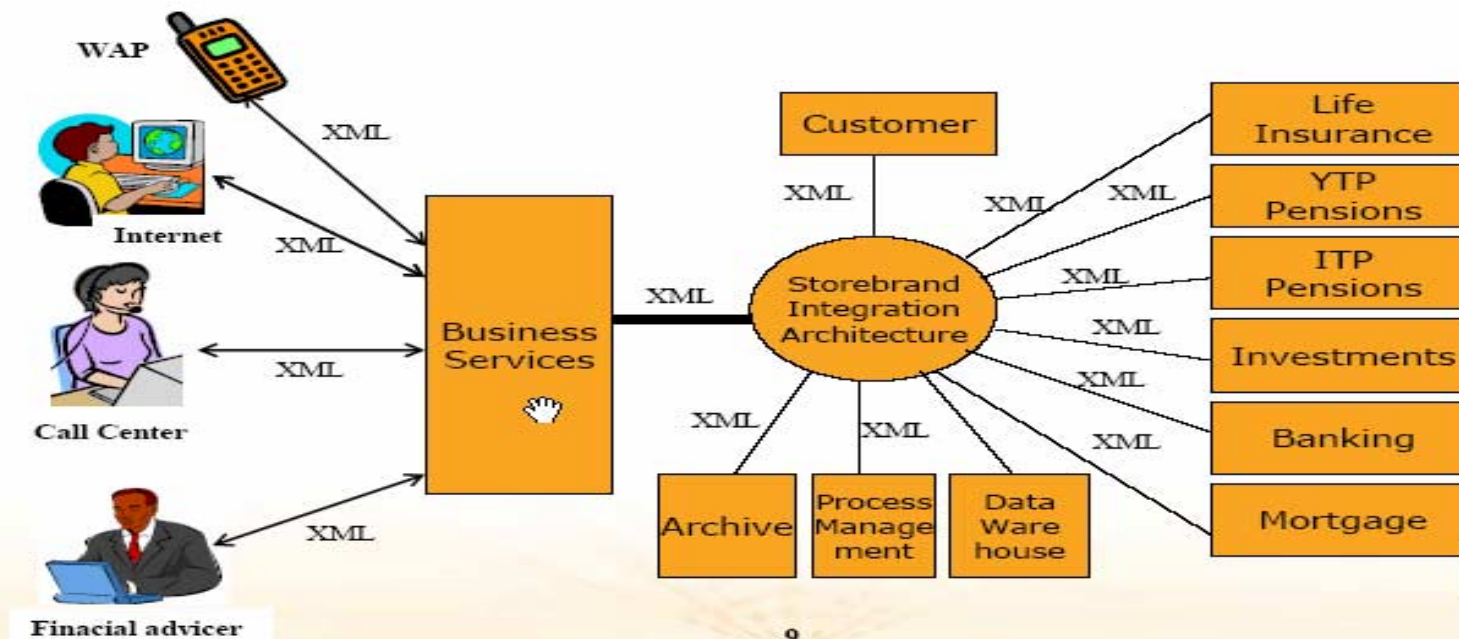
# DB2 Viper at Storebrand



IDUG® 2005 – Europe

Where Business & Data Converge

## Integration of systems



## Generalized Usage Requirements

- **Data has a highly variable schema**
- **Data has a schema that changes frequently**
- **Data that represents complex business object**
- **The document or message is the business artifact and needs to be preserved**
- **Data has sparse attributes or is otherwise unsuited for relational storage**

## A financial brokerage

- **General usage model**

- SOA with XML as business object

- **Climate**

- New architecture based on SOA/Web-services allows member brokers to send transactions in XML format using web services. Those XML documents ARE the business artifact (purchase order, order acknowledgement, etc..).

- **Requirements**

- Persistence of XML message (digitally signed business artifact necessary for audit)
- Search of XML (regulatory compliance)
- Shred of XML into relational (integration with existing business processes)

## A Travel Agency

### ■ General usage model

- XML with Relational

### ■ Climate

- Most of their data is highly structured
- Customer profiles
  - Have lots of attributes (>200)
  - Attributes are sparsely populated.
  - Attributes come-and-go comparatively quickly
  - Profiles are highly variable

### ■ Requirements

- Normal tables don't work well – too many joins to rebuild any given customers profile, they are too hard to process, and the profile changes too quickly (on average 1 change to the schema of the profile every week)
- Today this is done with a 'blob' – but any time they change the blob they need to change most apps that work with that blob. They also cannot 'search into' it quickly, or index any parts of it.

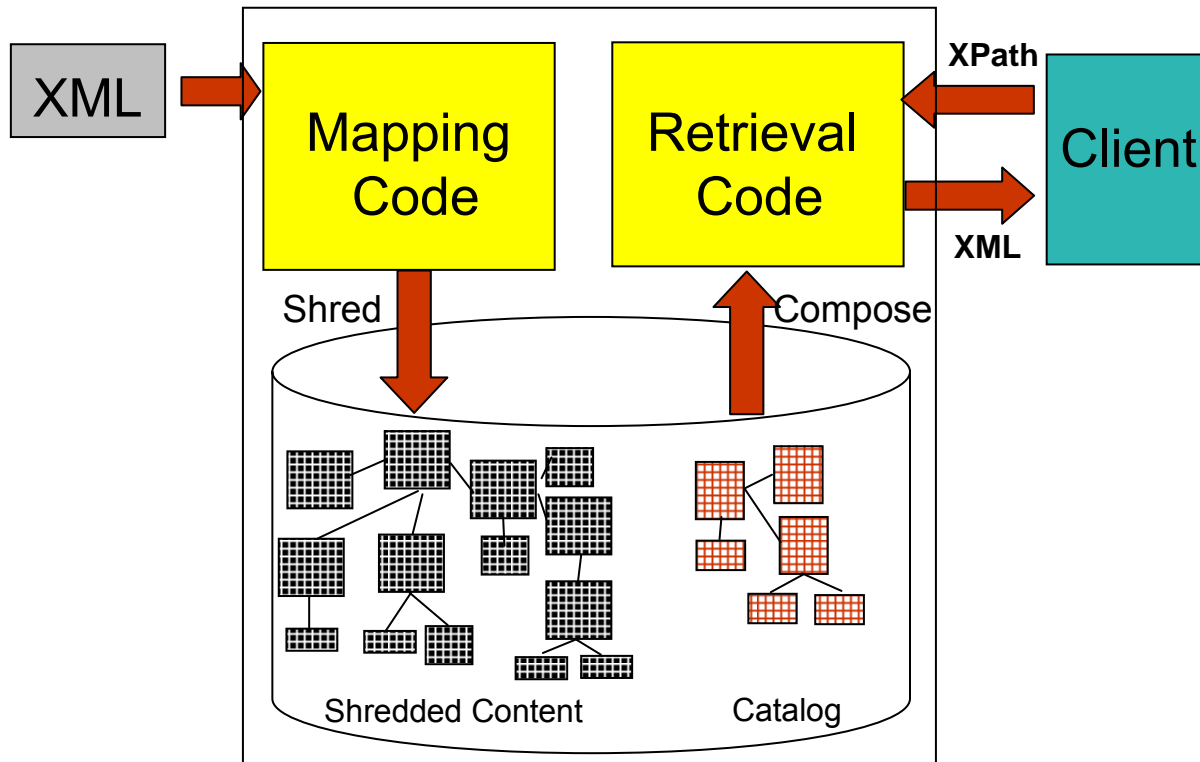
# A Manufacturing Company

- **General usage Model**
  - XML for document management
  
- **Climate**
  - Equipment is build from smaller parts, each with its own manual
  - Manuals are composite document
  - XML makes it easy to edit, compose, customize, translate
  
- **Requirement**
  - Persistent storage of XML fragments
  - Query and Search
    - What is affected when I change this part?
    - What existing documents cover this part?
    - Where is this document used?

## A Government Agency

- **General usage model**
  - SOA with XML
  
- **Climate**
  - Accident records are stored as XML documents
  - Sparsely populated attributes for accident details
  
- **Requirements**
  - This information needs to be sent to other agencies
  - Needs to be searched and co-related with Relational data

## An Example – Without Native Storage



- Incoming messages are shredded to relational storage
- The shredding is based on mappings controlled by 'schemas' stored in a proprietary catalog
- The storage schema is too complex to expose to clients
- XPath is used to express queries and the retrieval code parses the XPath and references the catalog to determine the SQL to generate in order to construct XML result sets

# Mapping is Complex and Fragile

- The simplest rule for shredding XML is to create a new table for every element that can occur multiply within its parent element.
- Elements that occur exactly once within a parent element are placed in the same table as the parent.
- Optional elements are problematic.

```

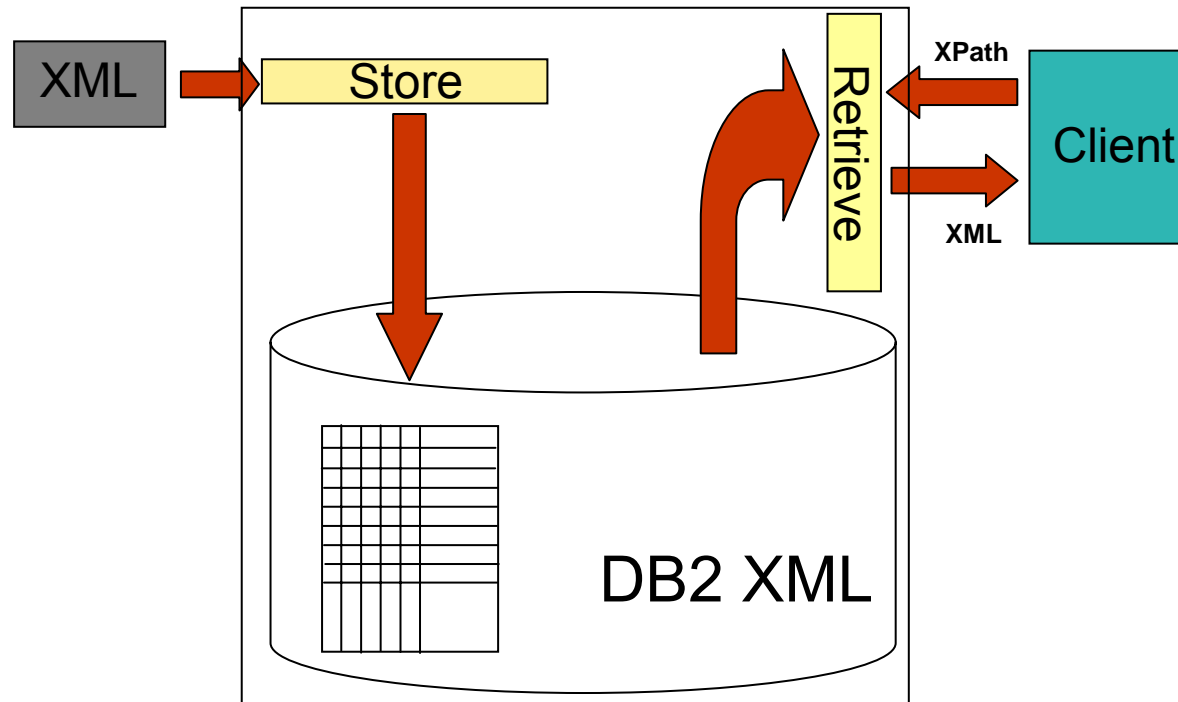
<? xml version="1.0" ?>
<purchaseOrder id='12345' secretKey='4x%$^'>
  <customer id="A6789">
    <name>John Smith Co</name>
    <address>
      <street>1234 W. Main St</street>
      <city>Toledo</city>
      <state>OH</state>
      <zip>95141</zip>
    </address>
  </customer>
  <itemList>
    <item>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

POid	CustId
987564331	A6789
....	...

POid	PartNo	Quantity
987564331	A54	12
987564331	985	1
....	....	....

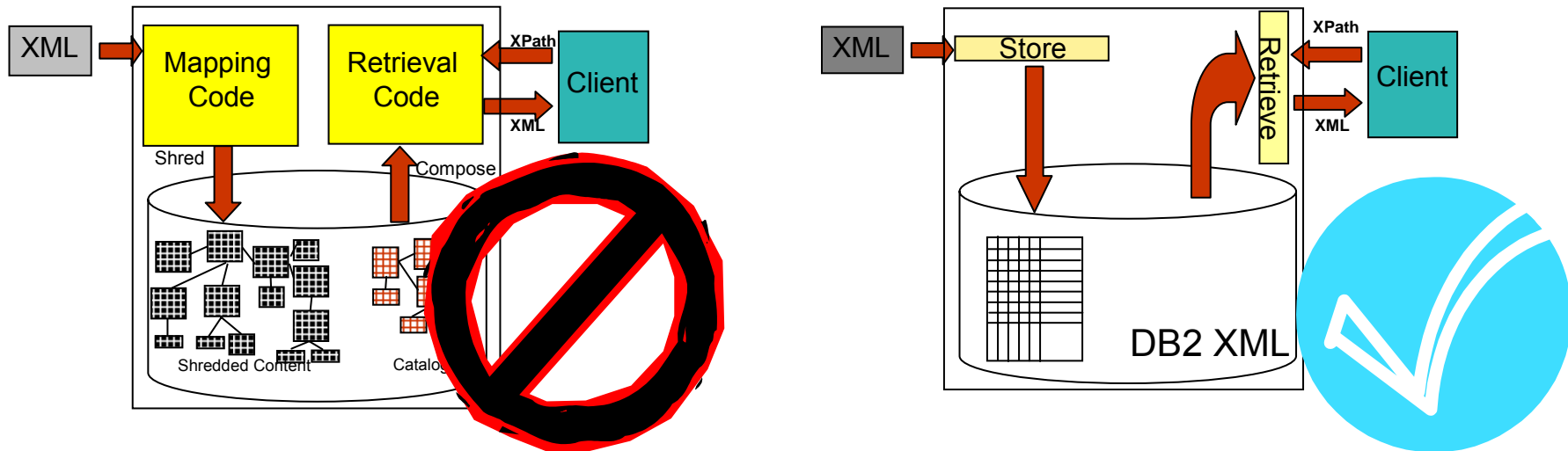
Custid	Name	Street	City	State	Zip
A6789	John Smith Co	1234 W. Main St	Toledo	OH	95141
...	...	...	...	...	...

## With Native Storage



- Incoming messages are stored intact in one table with key attributes, no shredding
- Messages can optionally be validated using DB2's XML Schema Repository, no proprietary catalog
- The schema is simple and can be exposed to clients
- Existing clients can continue to use the query interface, DB2 handles the XPath parsing, data retrieval and XML result set composition

## DB2 XML Benefits



- **Simplified and streamlined solution**

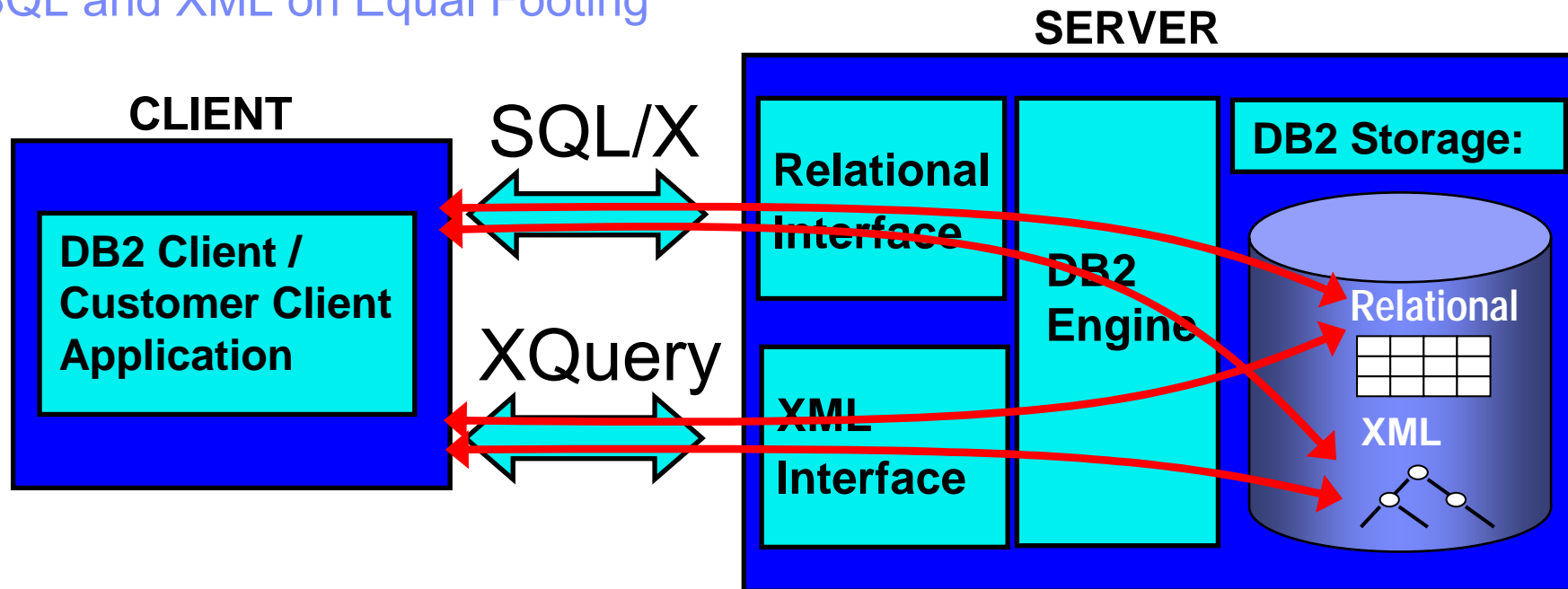
- No mapping code to write and maintain
- No complex schema to manage and maintain
- No proprietary catalog
- No XPath parsing and result set composition

- **Improved performance and flexibility**

- **Lower development and maintenance costs and faster to market**

## XML In DB2

### SQL and XML on Equal Footing



#### XML capabilities built into DB2

XML Extender integrated into the data engine  
 Deep engine implementation means high performance!

#### Optimized data store

New storage and indexing techniques for XML

#### Multiple Query Interfaces

SQL(X) and XQuery  
 ⇒ Both languages have full access to stored data  
 ⇒ Pick the view of the data that best suits the application

## Interface Overview

- **Data Definition**

```
create table dept(deptID int, deptdoc xml);
```

- **Insert**

```
insert into dept(deptID, deptdoc) values (?,?)
```

- **Index**

```
create index xmlindex1 on dept(deptdoc)  
generate key using xmlpattern '/dept/name' as varchar(30);
```

- **Retrieve**

```
select deptdoc from dept where deptID = ?
```

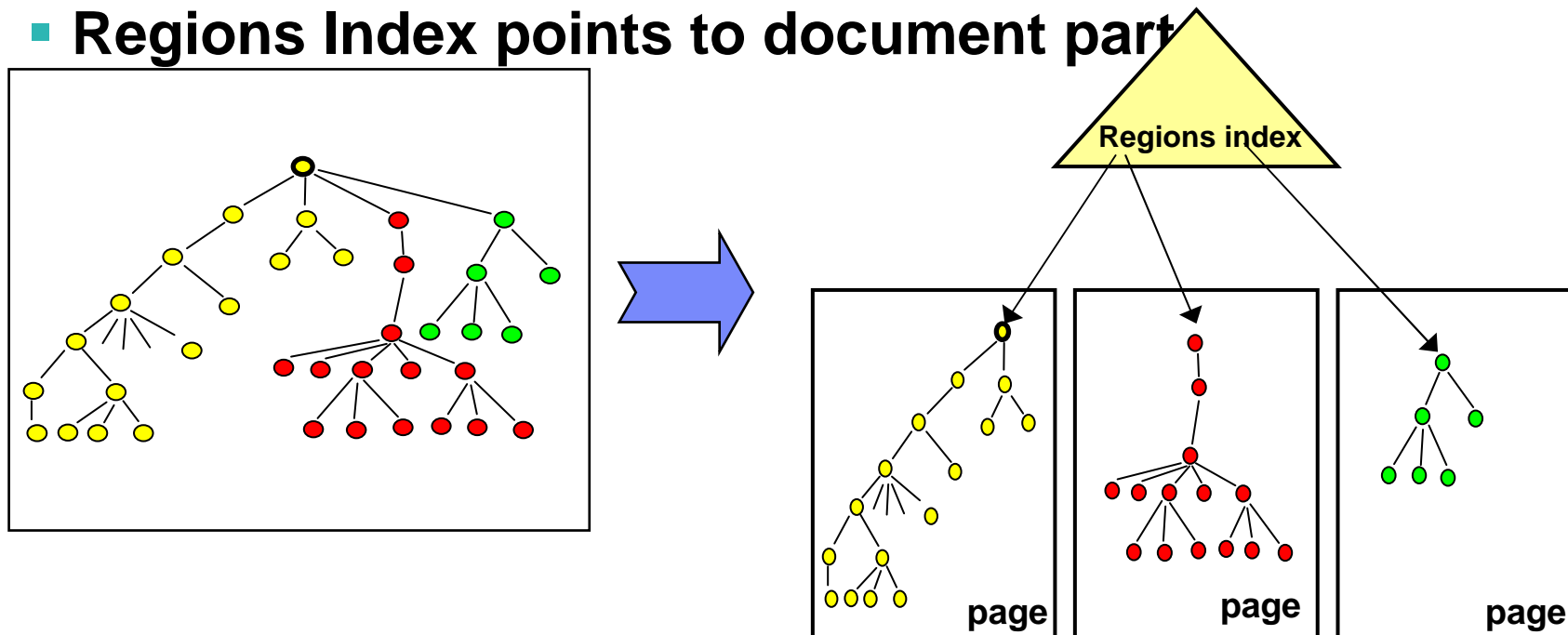
- **Query**

```
select deptID, xmlquery('$d/dept/name' passing deptdoc as "d")  
from dept where deptID <> "PR27";
```



## Document Storage

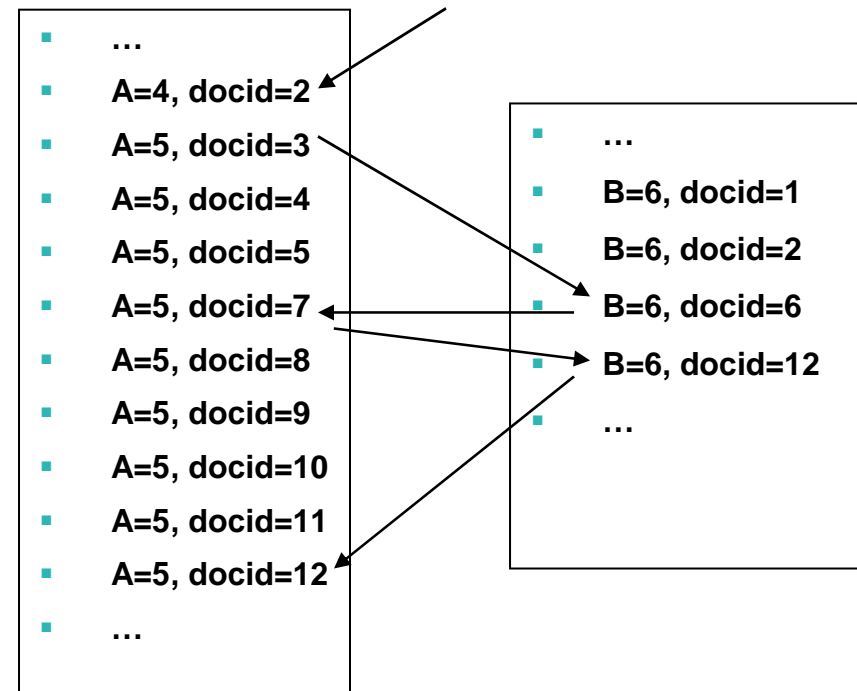
- **Node hierarchy of the XML Documents stored on DB2 pages**
- **Documents that don't fit on 1 page are split into pages/regions**
- **Regions Index points to document part**



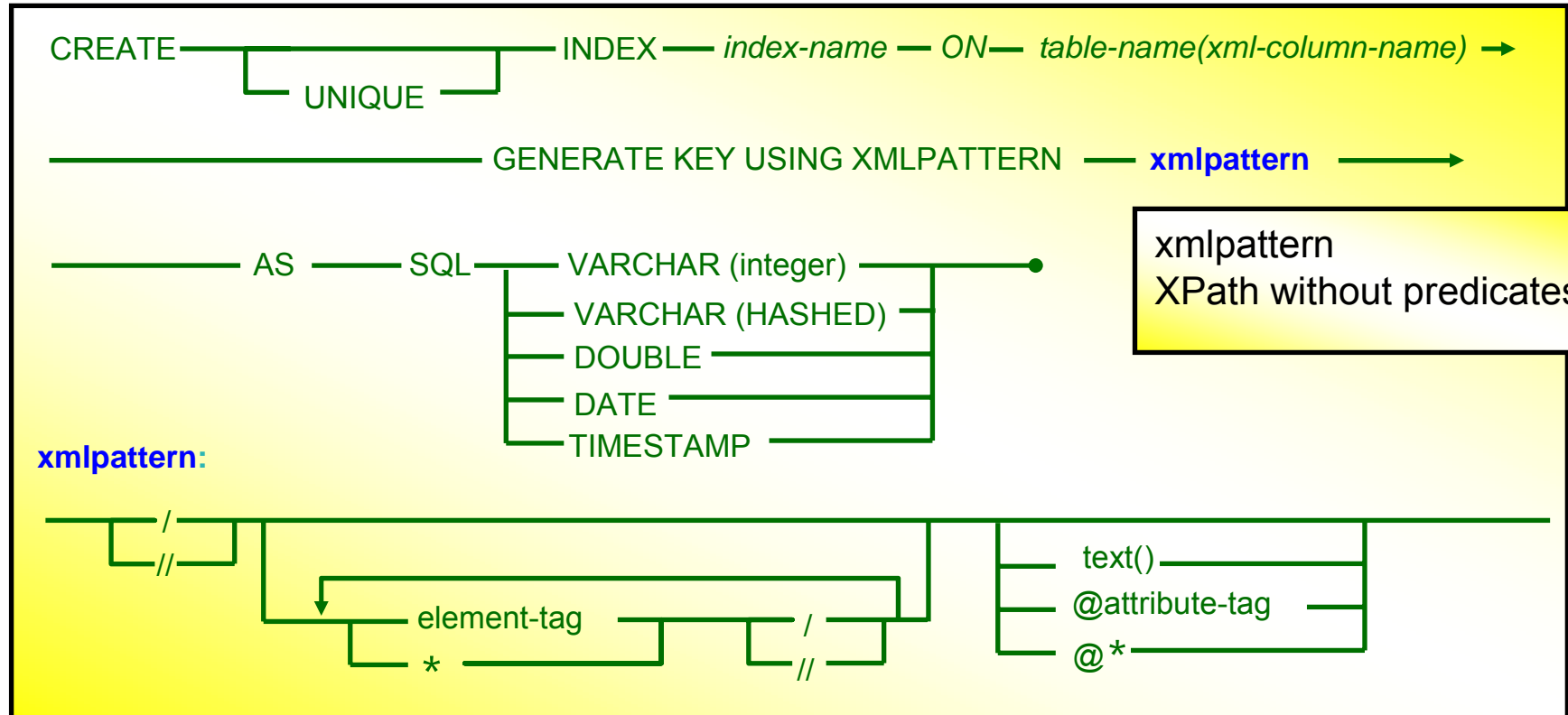
# Indexing

- **B-tree indexing extended and enhanced for XML Document management**
  
- **Additional fields**
  - PATHID: the path to the value
  - VALUE: the value that is indexed
  - DOCID: the document the value came from
  - NODEID: the location of the value in the doc
  
- **Additional evaluation techniques**
  - Pivot Join
  - Concurrent evaluation of “and” and “or” predicates
  - Multiple cursors into the index, each evaluating a predicate

Find docids where **x/y[A=5 and B=6]**



# Indexing



## Document Retrieval using SQL

- Retrieve XML documents

Select i, **deptdoc** from dept



- Retrieve some XML documents based on SQL Predicate

Select deptnum, **deptdoc** from dept

Where deptnum like 'PR%'

## Document Query using XQuery

- **Full power of XQuery and XPath 2.0**
  - Includes FLWOR nested FLWOR etc
- **XQuery over all documents in a column**
  - FOR \$d in **db2-fn:xmlcolumn**('dept.deptdoc')...
- **XQuery over documents selected using SQL predicates**
  - FOR \$d in **db2-fn:sqlquery**("select deptdoc from dept where deptID LIKE 'PR%' ")....

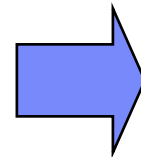


## The FLWOR Expression

- **F**OR: iterates through a sequence, bind variable to items
- **L**ET: binds a variable to a sequence
- **W**HERE: eliminates items of the iteration
- **O**RDER: reorders items of the iteration
- **R**ETURN: constructs query results



```
FOR $movie in db2-fn:xmlcolumn('movies')
LET $actors := $movie//actor
WHERE $movie/duration > 90
ORDER by $movie/@year
RETURN <movie>
      {$movie/title, $actors}
      </movie>
```



```
<movie>
  <title>Chicago</title>
  <actor>Renee Zellweger</actor>
  <actor>Richard Gere</actor>
  <actor>Catherine Zeta-Jones</actor>
</movie>
```

## SQL/XML – The best of both worlds

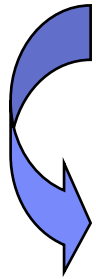
- Full power of SQL to address structured fields
- Full power of XQuery and XPath 2.0 to address and transform XML documents
- Join XML documents and tables
- SQL and XML Predicates
- Create XML from structured fields
- Materialize tables from XML documents



```
select d.deptID , u.headcount, xmlquery('$deptdoc/dept/name' passing d.deptdoc as "deptdoc")
  from dept d, unit u
  where d.deptID = u.unitID and u.headcount > 200
     and xmlcast(xmlquery('$deptdoc/dept/@bldg' passing d.deptdoc as "deptdoc") as int) = u.bldg
     and xmlexists('$deptdoc/dept/employee/name' passing d.deptdoc as "deptdoc")
```

# SQL/XML – XML Document Publishing

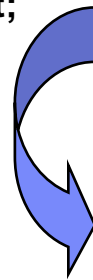
FIRSTNAME	LASTNAME	DEPARTMENT
SEAN	LEE	A00
MICHAEL	JOHNSON	B01
VINCENZO	BARELLI	A00



**SELECT**

**XMLELEMENT** (NAME "Department",  
**XMLATTRIBUTES** (e.department AS "name" ),  
**XMLAGG** ( **XMLELEMENT**(NAME "emp", e.firstname))) AS "department\_list"

**FROM** employee e **GROUP BY** e.department;



**department\_list**

```
<Department name="A00">  
  <emp>VINCENZO </emp>  
  <emp>SEAN</emp>  
</Department>  
<Department name="B01">  
  <emp>MICHAEL</emp>  
</Department>
```

# SQL/XML Publishing Functions

- **Scalar functions**
  - **XMLELEMENT** - generates an XML element
  - **XMLATTRIBUTES** - used within XMLELEMENT, specifies attributes
  - **XMLFOREST** - produces a forest of XML elements from SQL values
  - **XMLCONCAT** - concatenates a variable number of XML values
  - **XMLNAMESPACES** – produces a namespace declarations
- **Aggregate function**
  - **XMLAGG** - to group or aggregate XML data
- **Cast functions**
  - **XMLCAST** - converts between XML data type and standard relational types
  - **XMLSERIALIZE** – converts XML data type to serialized XML as a char/varchar/clob/blob
- **Table**
  - **XMLTABLE** – materializes a table from XML documents

# Language Bindings



## Java Example

```
PreparedStatement stmt1 = con.prepareStatement("Select deptdoc from dept where id = '001' ");
```

```
ResultSet rs = stmt1.executeQuery();
```

```
rs.next();
```

```
// Get the first returned document as a string
```

```
String xmlString = rs.getString(1);
```

```
// As a binary stream
```

```
InputStream is = rs.getBinaryStream(1);
```

```
// As an XML object
```

```
// The XML object class supports access methods for String, Byte and Streams
```

```
com.ibm.db2.jcc.DB2Xml xml = (com.ibm.db2.jcc.DB2Xml) rs.getObject (1);
```

```
rs.close();
```

```
stmt1.close();
```

# XML Schema Management

- **XML Schema Repository (XSR)**
  - Stores registered schemas
  - Managed as part of DB2 Catalogs
  - Tables and Views created automatically
    - SYSCAT.XSROBJECTS, SYSCAT.XSROBJECTCOMPONENTS
    - SYSCAT.XSROBJECTAUTH, SYSCAT.XSROBJECTHIERARCHIES
  - Command Line and Stored Procedure Interfaces

	<b>Stored Procedure</b>
<b>Register</b>	<b><i>XSR_REGISTER(rschema, name, schemalocation, content, docproperty)</i></b>
<b>Add</b>	<b><i>XSR_ADDSCHEMADOC(rschema, name, schemalocation, content, docproperty)</i></b>
<b>Complete</b>	<b><i>XSR_COMPLETE(rschema, name, schemaproperties, isusedforshred)</i></b>

## Validation using XML Schemas

- **Validation is optional and at the document level**

- **No validation**

- insert into dept(deptdoc) values (?)

- **With validation**

- insert into dept(deptdoc) values (xmlvalidate(?))

- **Schema override by referencing a schema from DB2's schema repository**

- insert into dept(deptdoc) values (

xmlvalidate(? according to xmlschema id "ibm.invoice")

- insert into dept(deptdoc) values (

xmlvalidate(? according to xmlschema uri 'http://my.world.com')



## Text Search

- **XML aware text index over XML documents**

Create index myIndex for text on dept(deptdoc) format xml

- **Not transactional**

Update index myIndex for text connect to mydb

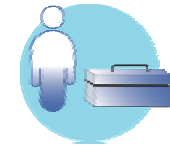
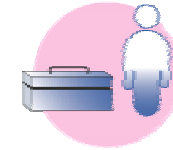


- **Simple and complex usage**

select i, doc from T where contains (doc, 'sections("/a/b") "zero") = 1

- **Includes Thesaurus support, Wildcard searches and proximity searches in same sentence or paragraph**

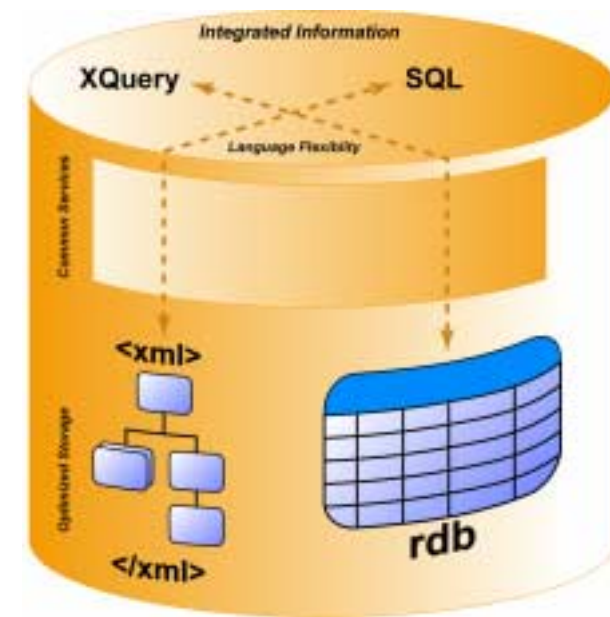
# Native XML Document Storage & RDB Integration



*"I see a rich XML repository that also supports SQL!"*

*"I see a rich RDBMS that also supports XML!"*

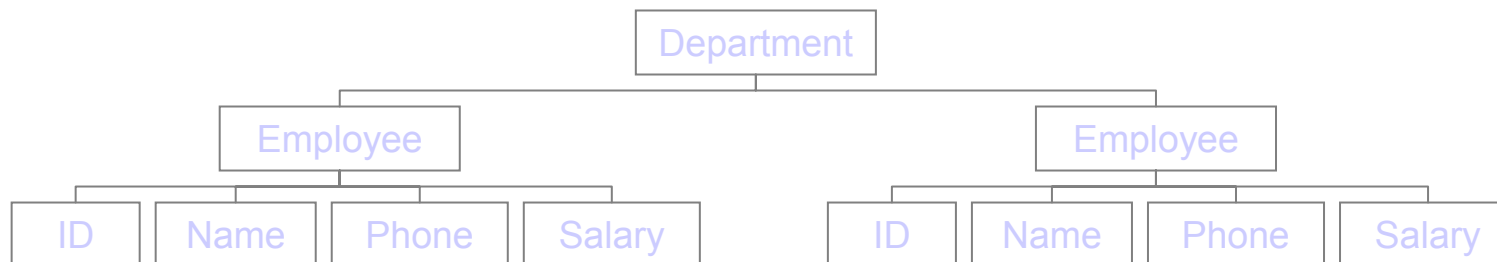
- Ensures fidelity of XML documents
- Integrates access using XQuery, SQL ...or both
- Optimizes performance & search-ability for any XML
- Native storage leverages mature DBMS services
  - ▶ Backup / Recovery, Security
  - ▶ Performance, Scale, Platform coverage
  - ▶ Transaction Support (ACID), Views
  - ▶ Administrative Skills, Privileges/Security +++



*Documents stored in a parsed hierarchy*  
*Any size, Any schema. No mapping, shredding, or complex application logic*  
*Uniform performance*

## DB2 Technology Preview

- New native XML storage manager in DB2 UDB
  - ▶ New XML column type
  - ▶ Documents are stored in parsed hierarchical format
    - No size, no schema, no mapping, no shredding, no complex application logic
    - Uniform performance regardless of document size or structure
  - ▶ Documents are indexed by any node or attribute (XPath specification)
    - Indexes are document-structure aware, can filter by containment, and 'point' directly to nodes
    - Cool new pivot (zipper) join makes queries self-tuning



***A “native” XML data store is one in which the representation of the data from client to disk and back is the XML data model, i.e. both the logical and the physical data model are XML.***

# ***XML Exploitation***



## Solving Business Problems & Overcoming Complexity

- *XML solves business problems today and is a pervasive technology & skill set*
- *Delivering information as a service helps overcome complexity*
- *XML is critical to deliver information as a service*

## XML benefits to Information Management

### ■ XML Technology

- ▶ XML = Extensible Markup Language
- ▶ Self-describing data structures
- ▶ Tags describe each element and their attributes

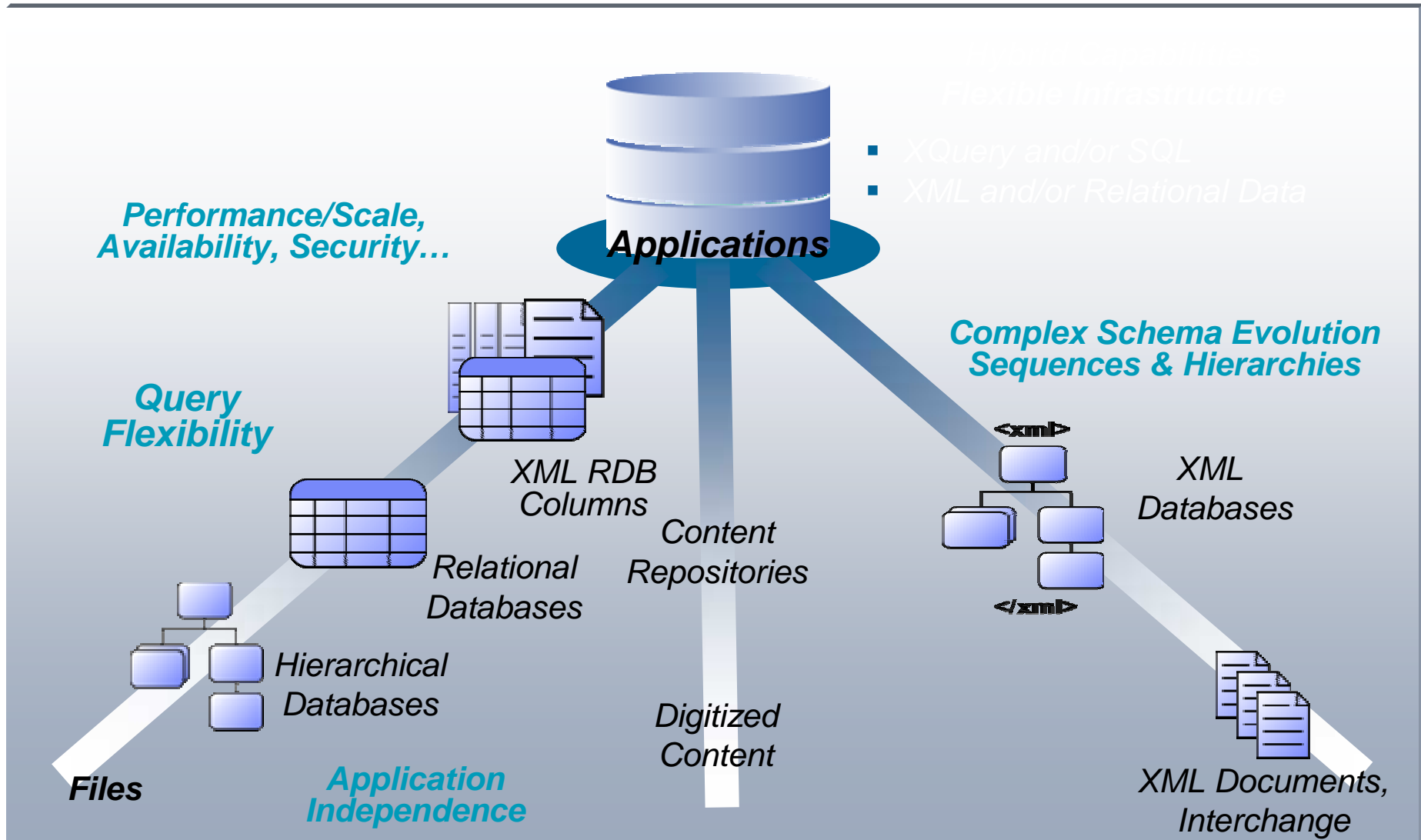
### ■ Benefits

- ▶ Extensible
  - No fixed format or syntax
  - Structures can be easily changed
- ▶ Platform Independent
  - Cross operating system, language or software vendor
  - 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>
```

# The Database Industry Inflection Point

## XML is Changing the Game

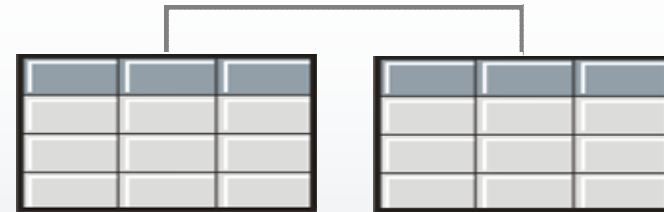


# Contrasting the Models

## *XML and Relational*

### ■ **Relational**

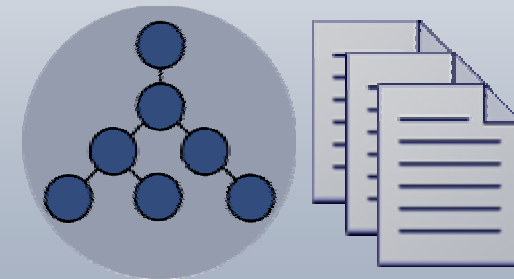
- ▶ Strength: Static data
  - Strict schema ensures data integrity
  - High performance indexing on fixed data
- ▶ Strength: 'Set-based' data
  - Multiple results returned
  - Retrieving rows



*Over \$20B Annual Customer Technology Investment in RDB Alone...*

### ■ **XML**

- ▶ **Strength:** Semi-structured, frequently changing data
  - Self-describing, flexible schema
  - Easily modified format
- ▶ **Strength:** Retrieving sequences
  - Documents, subdocuments, related documents



*XML database investments growing twice as fast as total database investment...*

# XML Data Needs Relational Maturity

## *Complementing XML Processing*

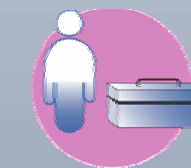
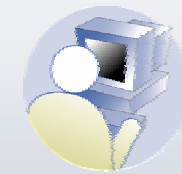
### ■ ***XML Data Needs Protection***

- ▶ Backup and recovery features to ensure continuity
- ▶ Data is protected using database security



### ■ ***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

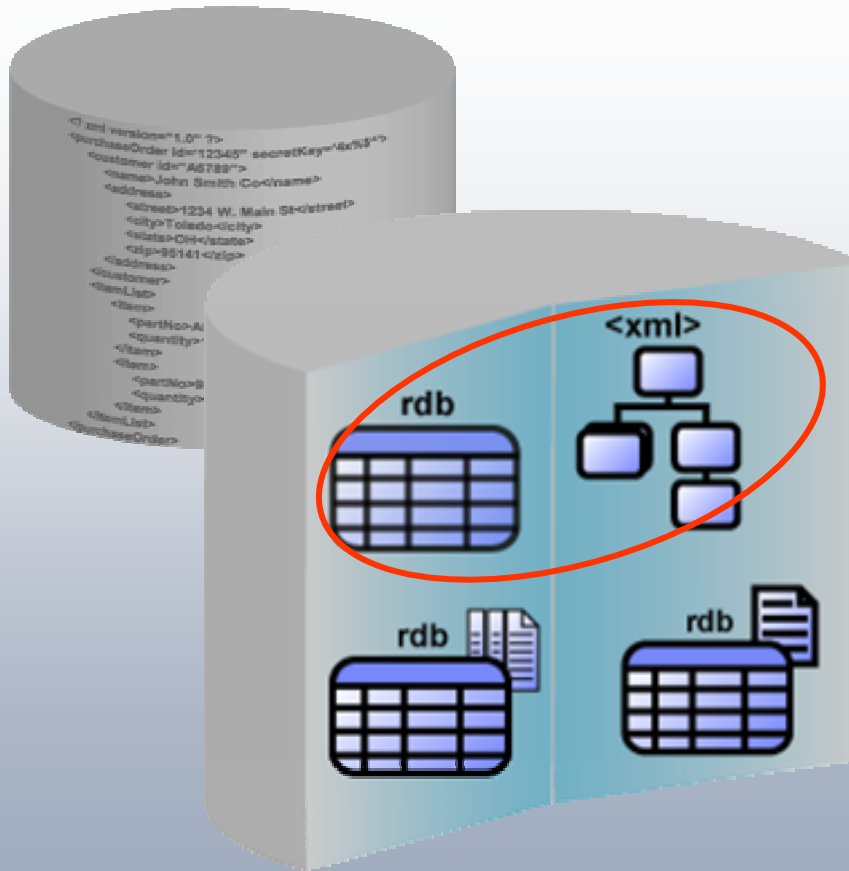
## Why is this a good XML Storage Model?

- Hierarchical storage for hierarchical data model  
 ⇒ doc traversal/sub-doc level access performance, flexibility

	CLOB	Shred	Native
Schema Flexibility	Best	Bad	Best
Search Performance	Bad	Good	Best
Full document return performance	Best	Bad	Good
Full document return performance (new API)	Good	Bad	Best
Partial document return performance	Bad	Good	Best
Partial document return performance (new API)	Bad	Good	Best
Insert performance	Best	Bad	Good
Update performance	Bad	Good	Best
Delete performance	Best	Bad	Good
Subdocument level privilege	Bad	Good	Best
Subdocument level concurrency control	Bad	Good	Best
Development cost (for IBM dev)	Best	Good	Bad

# Design Considerations

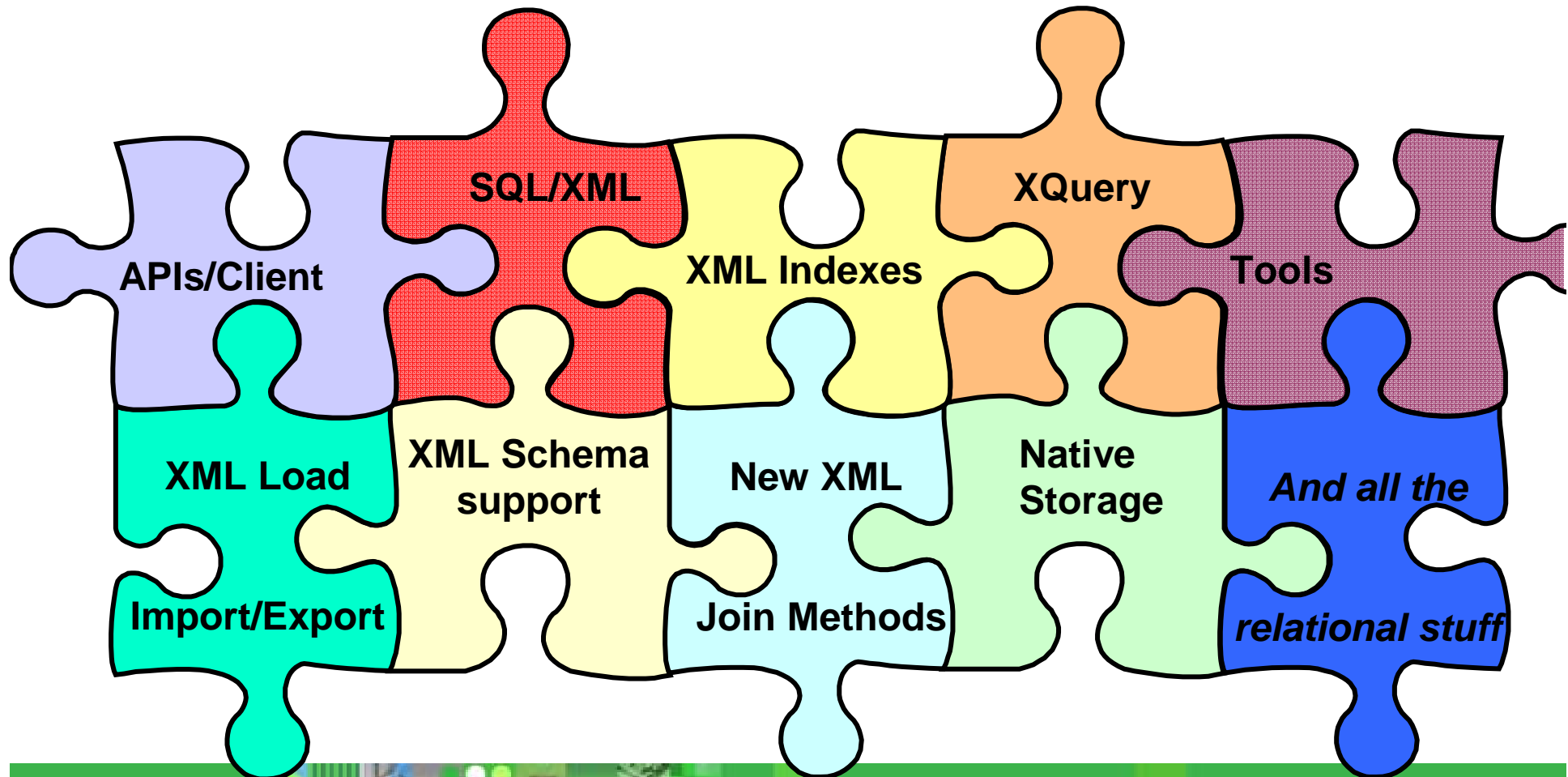
*Why a hybrid approach for DB2?*



	S	C	A	H
<b>Information Fidelity</b>	X	✓	✓	✓
<b>Integration</b>	✓	X	X	✓
<b>Schema Flexibility</b>	X	=	✓	✓
<b>Performance/Scale</b>	=	X	X	✓
<b>Programming Models</b>	X	X	=	✓
<b>Manageability</b>	✓	X	X	✓

## XML Summary

- Native XML support in DB2 with more to come
- Seamless integration with the relational world



## Native XML Store - summary

- **What does this mean ?**

- ▶ XML storage
- ▶ XML query

```
<myshoes>
  for $a in collection(/foo)
    where $a//hatsize/text()='12'
      return $a/shoesize
</myshoes>
```

- ▶ XML indexing
- ▶ XML privileges
- ▶ XML transformation
- ▶ XML schema support
- ▶ XML interfaces (DOM, SAX, etc..)
- ▶ Fully integrated text search
- ▶ Related tooling and infrastructure
  - IMPORT, EXPORT, LOAD, RUNSTATS, Replication, control center, monitor

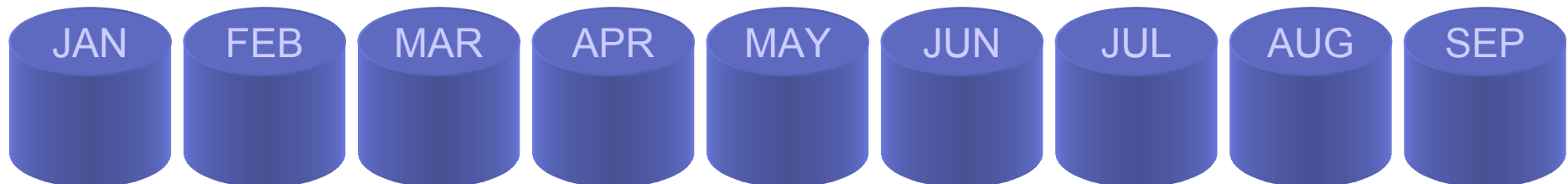
# Agenda

- Viper technology preview
  - ▶ Native XML data store
  - ▶ Range Partitioning support
  - ▶ Self Tuning Memory Manager
  - ▶ Miscellaneous Viper News



## Table Range Partitioning in DB2 LUW

- Partition a table by range
- Each range can be in a different tablespace
- Ranges can be scanned independently
- Use new ALTER ATTACH/DETACH statements for roll-in/roll-out



# Terms

- DATABASE PARTITIONING ( DPF was DB2 EEE)
  - ▶ Splitting data across logical nodes of the database
- TABLE PARTITIONING
  - ▶ Splitting data across multiple tablespaces within a logical node
- DATABASE PARTITION
- ▶ An individual “node” of a database that is using database partitioning
- RANGE or DATA PARTITION
  - ▶ An individual “range” of a table partitioned using table partitioning
  - ▶ Maps to an object on disk
- MULTI DIMENSIONAL CLUSTERING (MDC)
  - ▶ Physically organize data into blocks along multiple dimensions
- MATERIALIZED QUERY TABLE (MQT)
  - ▶ A table containing precomputed results of a query, maintained by the database

## Benefits of Range Partitioning

- Support for vastly larger tables
- Easier roll-in/roll-out
- Improved performance for certain business intelligence style queries
- Indexes can now be placed in different tablespaces

## Target Audience for Table Partitioning

- Data warehouse customers with large tables
- Data warehouse customers wanting easier roll-in/roll-out
- Customers considering migration to DB2
- Customers wishing to make better use of Hierarchical Storage Management (HSM) solutions

## Typical Use of Table Partitioning

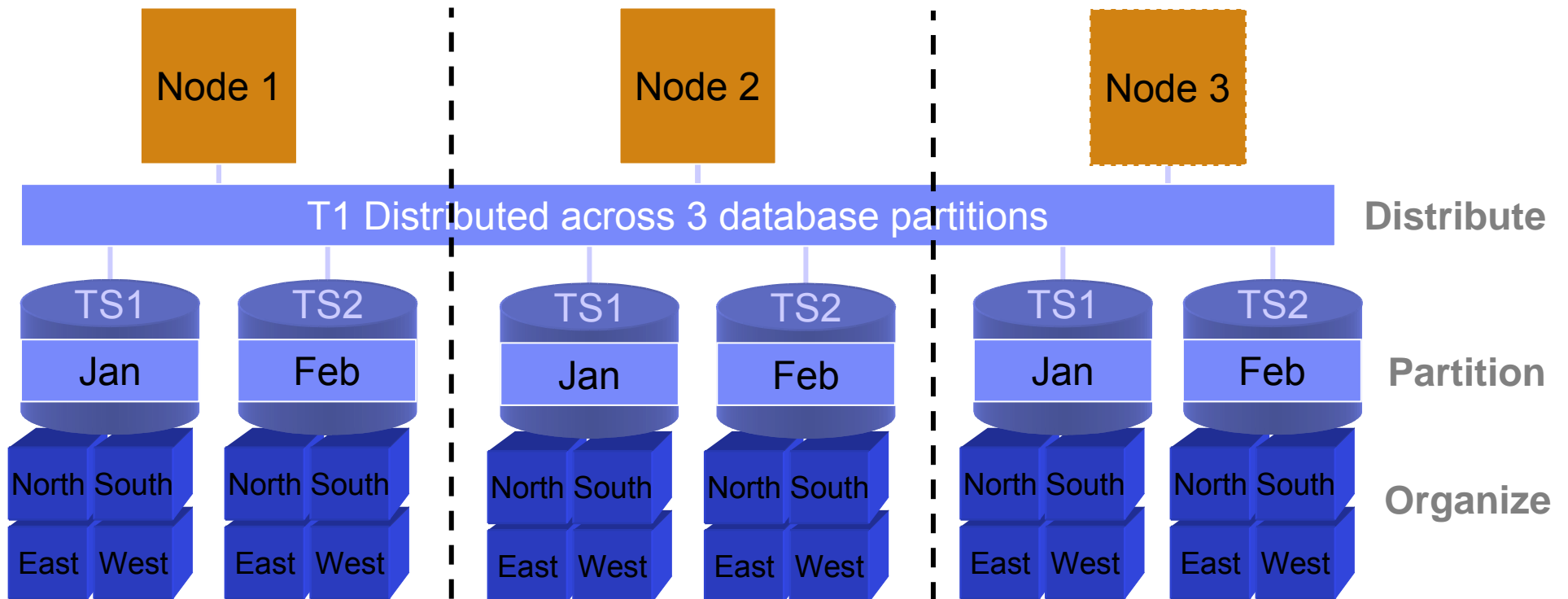
- Large fact table in a data warehouse
- Table is partitioned by range on a date column
- Ranges match granularity of roll-in/roll-out
- Data is rolled in/out using new ALTER ATTACH/DETACH statements

## Compatibility with Other DB2 Features

- Table partitioned tables can also be database partitioned and/or organized by dimension (MDC)
- MQTs work well on table partitioned tables
- There are some new things to know for indexes, LOAD, REORG
- Details later in this presentation

## Data Placement - Grand Unification

- Three ways to spread data
  - ▶ DISTRIBUTE BY HASH - currently in EEE/DPF
  - ▶ PARTITION BY RANGE – aka table partitioning
  - ▶ ORGANIZE BY DIMENSIONS – aka MDC



## New Operations for Roll-In and Roll-Out

- **ALTER TABLE ... ATTACH**
  - ▶ Incorporates an existing table as a new range
  - ▶ Then SET INTEGRITY to validate data and maintain indexes
  - ▶ Data becomes visible all at once after COMMIT
  - ▶ Minimal interruption to other queries accessing table
- **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
- Key points
  - ▶ No data movement
  - ▶ Nearly instantaneous
  - ▶ SET INTEGRITY is now online

## Using Table Partitioning with ESE w/DPF

Need to Address	Best	Explanation
Table capacity	Partitioning	Table Partitioning is simpler to setup and maintain.
Parallel query execution (query performance)	ESE w/DPF	ESE w/ DPF provides query parallelism.
Partition elimination (query performance)	Partitioning	Table partitioning provides partition elimination.
Maximum query performance	Both	Query parallelism and partition elimination are complementary. Use both for maximal query performance.
Divide and conquer for administration tasks	ESE w/DPF	Tasks such as reorg and backup can be done one node at a time.

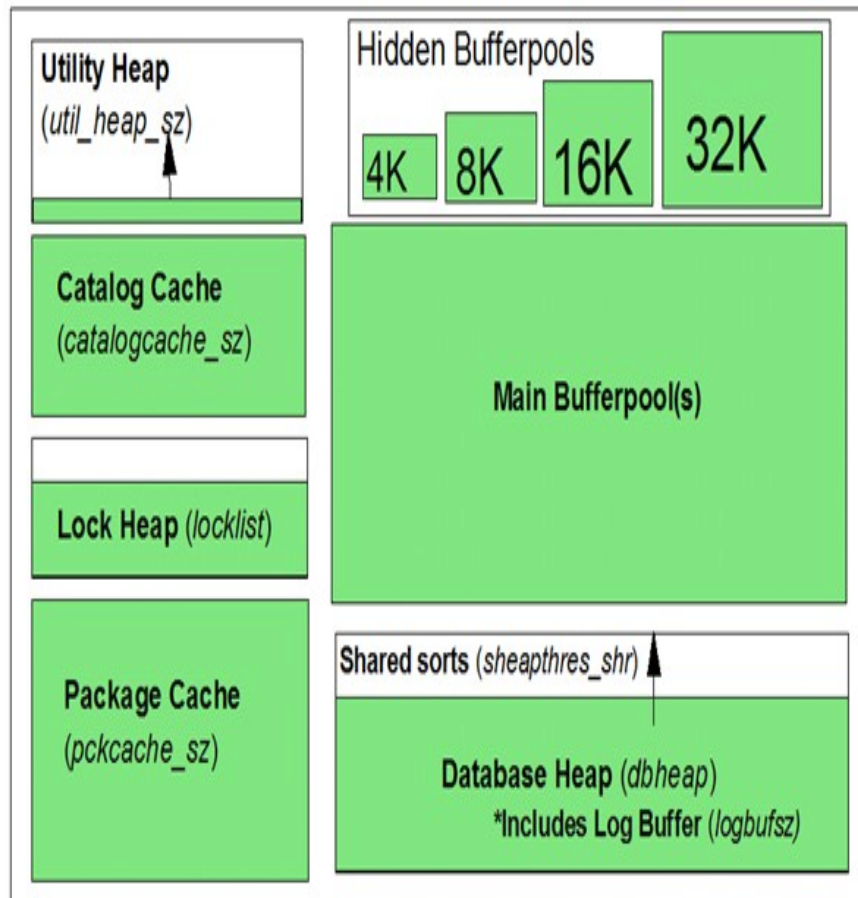
# Agenda ?

## Viper technology preview

- ▶ Native XML data store
- ▶ Range Partitioning support
- ▶ **Self Tuning Memory Manager**
- ▶ Miscellaneous Viper News



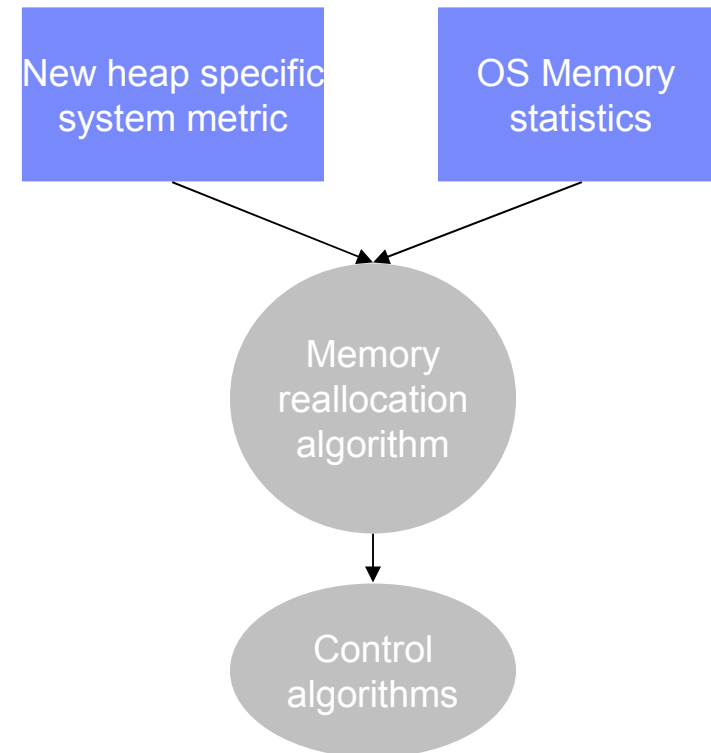
## DB2's database memory model – shared memory



- All memory heaps are contained within the database shared memory set
  - ▶ On **non-Windows** platforms the set memory is all allocated at db startup and can not grow beyond its allocated size
  - ▶ On **Windows** the memory is allocated at startup but can grow or shrink as needed
- On 32 bit platforms set size is limited
- On 64 bit platforms set size virtually unlimited

## STMM Overview

- Uses totally new system metrics
- Constantly monitors system to make use of any free OS memory (only if DATABASE\_MEMORY is set to AUTOMATIC)
- Works iteratively to determine an optimal memory configuration for all heaps
  - ▶ Iterative approach prevents instability
- Control algorithms help determine interval length and prevent oscillations



## STMM Modes of operation

- Works in two different modes
  - ▶ Tuning DATABASE\_MEMORY parameter
    - Takes from, and returns memory to, the OS as necessary
    - Total amount of memory used by DB2 can grow over time
    - Requires only one heap for tuning
  - ▶ No tuning of DATABASE\_MEMORY parameter
    - Memory tuning still occurs but total memory used by database is constant
    - For one heap to grow another heap must shrink
      - Requires two heaps to be able to tune
- Is able to tune multiple databases and instances on the same box at the same time
- Works in non-partitioned and in partitioned (DPF) environments

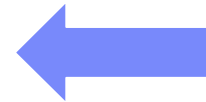
## Summary

- STMM - advanced automated memory tuning for DB2 Viper
- Tunes online and without any DBA intervention
- Tunes total database memory usage
- Tunes the four largest consumers of database memory
  - ▶ Is activated through database configuration (and alter bufferpool command)
- Alleviates the burden of manual tuning
  - ▶ Can find a near optimal solution in only an hour
- Works with DPF when nodes are similar

## Agenda ?

### Viper technology preview

- ▶ Native XML data store
- ▶ Range Partitioning support
- ▶ Self Tuning Memory Manager
- ▶ **Miscellaneous Viper News**



## Miscellaneous Viper news

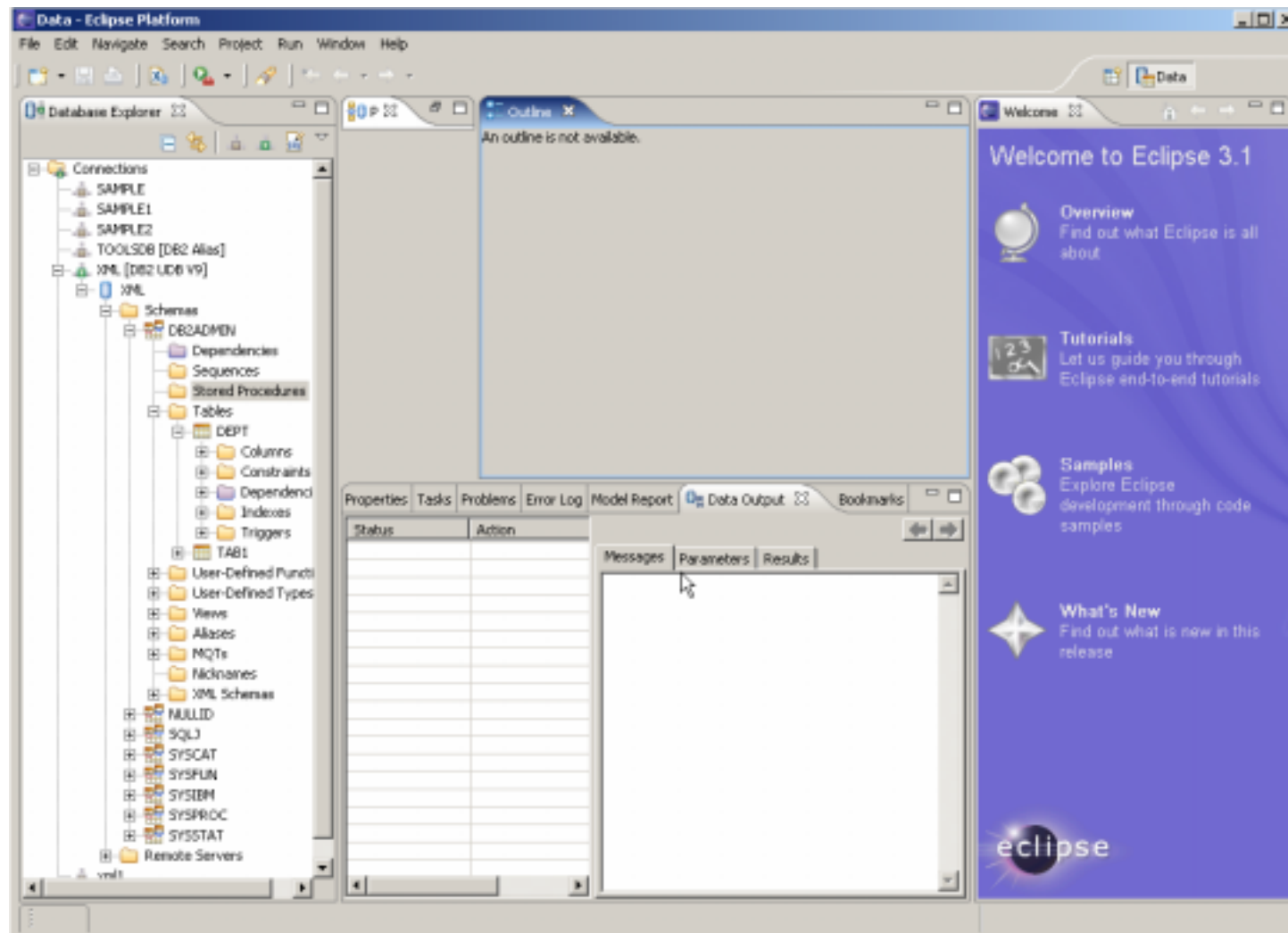
- **Development**
  - ▶ New Developer Workbench
- **Security**
  - ▶ Role based access control
    - defining and managing roles inside the database.
  - ▶ New Security Administrator authority (SECADM)
  - ▶ Trusted connections
    - ability for DB2 to trust the application that connects to it to perform authentication, avoiding overhead of DB2 auth checking
  - ▶ LDAP Authentication
  - ▶ Enhanced support for Windows security
    - long names
  - ▶ Row level access control
    - A mechanism for restricting access to rows in a table
    - Also known as "Label Security" and "Fine Grained Access Control"
- **Autonomic Technology**
  - ▶ Automatic Storage Provisioning
    - Greatly simplify the task of apportioning storage for DB2 logs and data
  - ▶ Progressive Re-optimization
    - "The learning optimizer" (LEO)
    - Possible approaches might enable re-optimization in mid-stream

## Miscellaneous Viper news - Development

### ■ **Developer Workbench**

- ▶ Eclipse-based tool
- ▶ The next generation of data application development tooling for DB2.
- ▶ Develop, debug, deploy, and test stored procedures and UDF
- ▶ Create and run SQL statements and XML queries.
- ▶ Integrate data objects into Java applications using the Developer Workbench's Eclipse Java project support.

# DB2 Developer Workbench



## Miscellaneous Viper news - Security

### ■ Label-Based Access Control

- ▶ Allows control access to tables at the row-, column level or both
- ▶ using security labels.
- ▶ Implementation of Mandatory Access Control (MAC)
- ▶ Tailor the security label definition to your own specific needs.
- ▶ Build secure applications that manage classified information or simply apply a fine-grained access control policy to your database table rows and/or columns
- ▶ DB2 LBAC is fully integrated and is very easy to use.
- ▶ Extensions to DB2 SQL allows
  - to set up security policies,
  - attach a security policy to a database table to control access to rows or columns within that table,
  - manage database users' security label and exemption credentials.

## Miscellaneous Viper news - Security

- **Enhancements to the current DB2 security model**
  - ▶ **Introduction of the Security Administrator authority (SECADM),**
  - ▶ **New SETSESSIONUSER database privilege**
    - **gives more flexibility in controlling who can change the session authorization ID and to which value,**
  - ▶ **New RESTRICT option on the CREATE DB**
    - **allows to create a db without PUBLIC access to the system catalogs**

## Other SQL news (found by Saeid)

- ALTER TABLE XXX DROP COLUMN C1;
- ALTER TABLE XXX ALTER COLUMN C1 SET NOT NULL
- ALTER TABLE XXX ALTER COLUMN C1 DROP NOT NULL

## DB2 Viper Summary

- Viper technology preview
  - ▶ Native XML data store
  - ▶ Range Partitioning support
  - ▶ Self Tuning Memory Manager
  - ▶ Miscellaneous Viper News