

Information Management



# IBM® DB2® Workshop for Oracle® Professionals

## Hands-On Labs Workbook



IBM Canada Ltd.

Information Management Ecosystem  
Partnerships

V-1.3  
12/16/2010



# **IBM® DB2® Workshop for Oracle® Professionals**

## **Hands-On Labs Workbook**

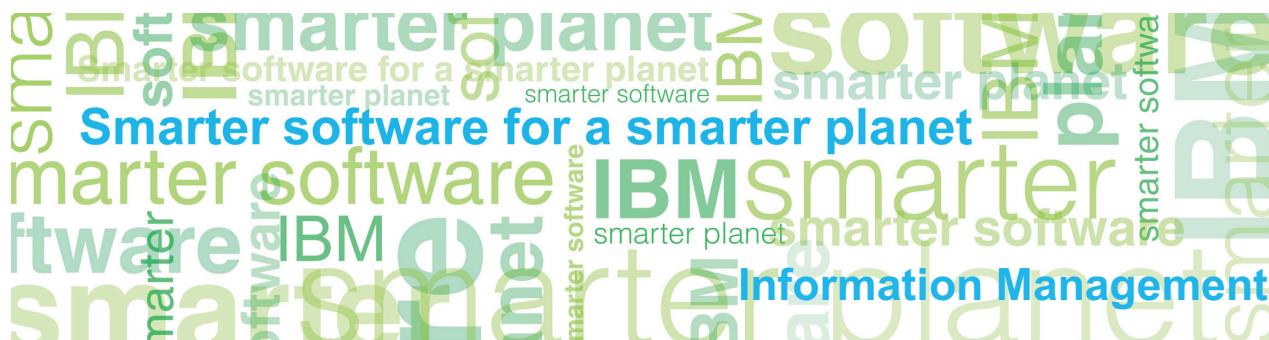
**Information Management Ecosystem Partnerships**  
**[techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)**

# Table of Contents

<b>WELCOME.....</b>	<b>4</b>
<b>DB2 FUNDAMENTALS FOR ORACLE PROFESSIONALS.....</b>	<b>11</b>
<b>DB2 PRACTICAL AUTONOMICS.....</b>	<b>36</b>
<b>DB2 PRACTICAL AUTONOMICS HANDS-ON LABS.....</b>	<b>54</b>
<b>DB2 ADVANCED FEATURES.....</b>	<b>91</b>
<b>BREAKING FREE FROM ORACLE WITH DB2 9.7.....</b>	<b>118</b>
<b>DATA CONCURRENCY.....</b>	<b>129</b>
<b>DB2 PUREXML TECHNOLOGY OVERVIEW.....</b>	<b>142</b>
<b>XQUERY HANDS-ON LABS.....</b>	<b>168</b>

# DB2® Workshop for Oracle® Professionals

## Welcome



© 2011 IBM Corporation

Information Management



## Disclaimer

© Copyright IBM Corporation 2011. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Other company, product, or service names may be trademarks or service marks of others.

DB2 Workshop for Oracle Professionals page 4 of 207.

## Agenda – Day 1

- 1.0 Welcome! 15min
- 1.1 DB2 Fundamentals for Oracle Professionals 1,5h
  - < Break > 15min
- 1.2 Practical Autonomics 45min
- Lunch
- 1.3 DB2 Practical Autonomics Lab (**Hands On**) 1h
  - < Break >
- 1.4 DB2 Advanced Features 1h
- 1.5 M45 Exam Review 30min
- 1.6 M45 DB2 Technical Professional Certification Exam 1,5h

3

© 2011 IBM Corporation

## Agenda – Day 2

- 2.1 Breaking Free from Oracle with DB2 9.7 45min
- 2.2 Migrating to DB2 (**Demo**) 45 min
  - < Break > 15 min
- 2.3 Data Concurrency 20min
- 2.4 DB2 pureXML 30min

### Lunch

- 2.5 XQuery Lab (**Hands On**) 45min
- 2.6 Exam Review 1h
  - < Break >
- 2.7 DB2 Database Associate Certification Exam 2h

## IBM Guided Hands-on Technical Learning -Technical Education Bootcamps

- **IM Technical Bootcamps**

- WW Bootcamps Available for:

- DB2 9.7 LUW & Migration Clinic
- DB2 for SAP
- DB2 pureXML
- Informix 11.5
- IBM InfoSphere Warehouse v9.7
- InfoSphere DataStage and QualityStage
- InfoSphere Optim Solutions
- InfoSphere Guardium
- InfoSphere Change Data Capture
- InfoSphere Information Server
- InfoSphere Master Data Management
- SolidDB



- **2011 schedule, bootcamp agenda's and registration available here:**  
[www.ibm.com/developerworks/data/bootcamps/](http://www.ibm.com/developerworks/data/bootcamps/)

5

© 2011 IBM Corporation

## DB2 Technical Mastery Exam

- **Test M45: IBM Information Management DB2 Technical Mastery**

- This technical mastery test meets one of the requirements for the IBM Software Value Incentive Program and is eligible for reimbursement under the PartnerWorld We Pay Program.
- Objectives

- IBM Information Management DB2 (56%)
  - Discuss what is new in DB2 9.7
  - Getting started with DB2 installation
  - DB2 installation
  - DB2 basic administration
  - DB2 administration and development tools
- IBM Information Management DB2 Express (44%)
  - IBM DB2 Express and DB2 Express-C
  - Benefits of DB2 Express
  - DB2 fundamental concepts and tools
  - DB2 security
  - Data movement utilities
  - Basic Performance Tuning



[http://www-03.ibm.com/certify/mastery\\_tests/ovrM45.shtml](http://www-03.ibm.com/certify/mastery_tests/ovrM45.shtml)

## DB2 Family Fundamentals – DB2 Database Associate

- **Test 730: IBM Information Management DB2 Database Associate**

- This is an entry level DBA or user of any of the DB2 family products. The certified individual is knowledgeable about the fundamental concepts of DB2 9.
- Exam breakdown:
  - Planning (14%)
  - Security (11%)
  - Working with Databases and Database Objects (17%)
  - Working with DB2 Data using SQL (23.5%)
  - Working with DB2 Tables, Views and Indexes (23.5%)
  - Data Concurrency (11%)



<http://www-03.ibm.com/certify/tests/obj730.shtml>

## Individual Reading



- **Reading Materials – printed**

- [www.ibm.com/software/data/education/bookstore](http://www.ibm.com/software/data/education/bookstore)
- Study Guides
  - DB2 9 Fundamentals 978-1-58-347072-5
  - DB2 9 for Linux, UNIX, and Windows Database Administration 158347-077-8
  - DB2 9 for z/OS Database Administration 978-158347-074-9
  - DB2 9 for Linux, UNIX, and Windows Database Administration Upgrade 158347-078-6
- Books
  - DB2 9 for Linux, UNIX, and Windows – Sixth Ed. 0-13-185514-X
  - Understanding DB2: Learning Visually
- Manuals: <http://www-01.ibm.com/support/docview.wss?rs=71&uid=swg27015148>
- DB2 Information Center: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>
- IBM Redbooks: <http://www.redbooks.ibm.com/> with Examples 0-13-158018-3
  - Understanding DB2 9 Security 0-13-1345907
  - DB2 9 for Developers 978-158347-071-9



- **Tutorials/Self-Study**

- [www.ibm.com/software/data/education/selfstudy.html](http://www.ibm.com/software/data/education/selfstudy.html)

- **Data Management Magazine – (Former IBM Database Magazine)**

- <http://www.ibmdmmagazinedigital.com/>

- **Performance Perspectives – Insights and ideas on Information on Demand**

- <http://www-01.ibm.com/software/data/performance-perspectives/>

## DB2 Certification & Free Tutorials

- **Information Management Certification website:**

<http://www.ibm.com/software/data/education/certification.html>



- Step 1

- **IBM Certified Database Associate for DB2 9 Fundamentals, Exam 730**

- Exam info: <http://www-03.ibm.com/certify/tests/obj730.shtml>
- Free tutorial: <http://www.ibm.com/developerworks/offers/lp/db2cert/db2-cert730.html>

**DBA**

- Step 2 (LUW)
  - **IBM Certified Database Administrator for DB2 9.7 DBA for LUW, Exam 541**
    - Exam info: <http://www-03.ibm.com/certify/tests/obj541.shtml>
  - **OR DB2 9.7 Database Administrator for LUW Upgrade, Exam 546**
    - Exam info: <http://www-03.ibm.com/certify/tests/ovr546.shtml>
- Step 3(LUW)
  - **IBM Certified Advanced Database Admin. for DB2 9.7 DBA for LUW, Exam 544**
    - Exam info: <http://www-03.ibm.com/certify/tests/ovr544.shtml>
- Step 2 (for z/OS)
  - **IBM Certified DB2 10 DBA for z/OS, Exam 612**
    - Exam info: <http://www-03.ibm.com/certify/tests/ovr612.shtml>

**App. Dev**

- Step 2 (LUW)
  - **IBM Certified Application Developer for DB2 9.7, Exam 543**
    - Exam info: <http://www-03.ibm.com/certify/tests/obj543.shtml>

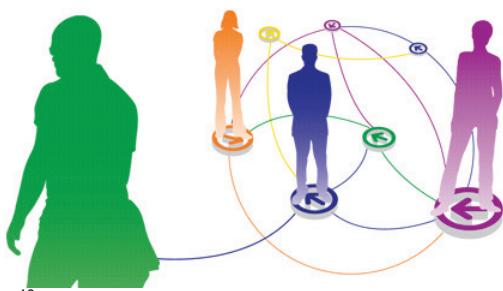
9

© 2011 IBM Corporation

## Bringing it all Together -Information on Demand Conference



- IOD Global, 23 - 27 October 2011 – Las Vegas, United States
  - Education/Certification
  - Business Prospecting
  - Partner Networking
  - IBM Relationships



10

## Access to Software

- DB2 Express-C
  - <http://www.ibm.com/db2/express/>
- IBM Data Server Client
- Application Development Downloads
  - IBM Data Studio: <http://www.ibm.com/software/data/optim/data-studio/>
  - Zend Core for IBM
  - More Info: [www.ibm.com/software/data/db2/ad/](http://www.ibm.com/software/data/db2/ad/)
- Download Via: [www.ibm.com/db2/express/download.html](http://www.ibm.com/db2/express/download.html)
- Software Value Package:



**International DB2 User Group**



## IDUG – DB2 Community Support

- FREE membership
- User-driven education and training at in-person events around the world:
  - **Annual Conferences** in North America, Europe, and Australia
  - **Regional Events** bring DB2 to the local community
- Wealth of FREE online content
  - **DB2-L** is a discussion forum with more than 3,000 registered participants
  - **Code Place** stores real-life examples, best practices, and downloads
  - **Whitepapers** provide a broad source of industry-specific information
  - **Local User Group** information
  - **Conference Proceedings** for several year's of IDUG in-person events
  - **Blogs** supporting a diverse array of DB2 discussion topics
  - **DB2 Community Calendar** to detail DB2 events and webcasts
- Opportunities for Volunteer Leadership
- Find out more by registering for FREE at [www.idug.org](http://www.idug.org)

## DB2 Express-C vs. DB2 Express Fixex Term License (FTL)

Feature	DB2 Express-C Free (UNWARRANTED)	DB2 Express FTL* Paid Subscription
Core DB2 capabilities	YES	YES
Free admin tools	YES	YES
Free development tools	YES	YES
Autonomic capabilities	YES	YES
pureXML feature	YES	YES
No-charge community based assistance***	YES	YES
Official IBM 24x7 support	NO	YES
Fixpacks	NO	YES
High Availability (HADR)	NO	YES
Data Replication	NO	YES
Processor Usage Limit****	2 cores	4 cores (max 2 sockets)
Memory Usage Limit****	2 GB	4 GB
Update Availability	Full release at major milestones	Fixpacks generally every 3 months
Free access to old/current Product images	Yes, until major release only	Product Lifetime
Price per Server per Year**	\$ 0	\$ 2,995

- \* Features entitled with Subscription are available only while Subscriptions are valid
- \*\* Subscription Price indicated is for United States and subject to change. Pricing in other countries may vary.
- \*\*\* No-charge community-based assistance is available via the online forum.
- \*\*\*\* CPU and memory limitations for DB2 Express-C are not limitations of the machine size, rather they specify DB2 usage limits on those machines.

Information Management Technology Ecosystem  
IBM Toronto Lab



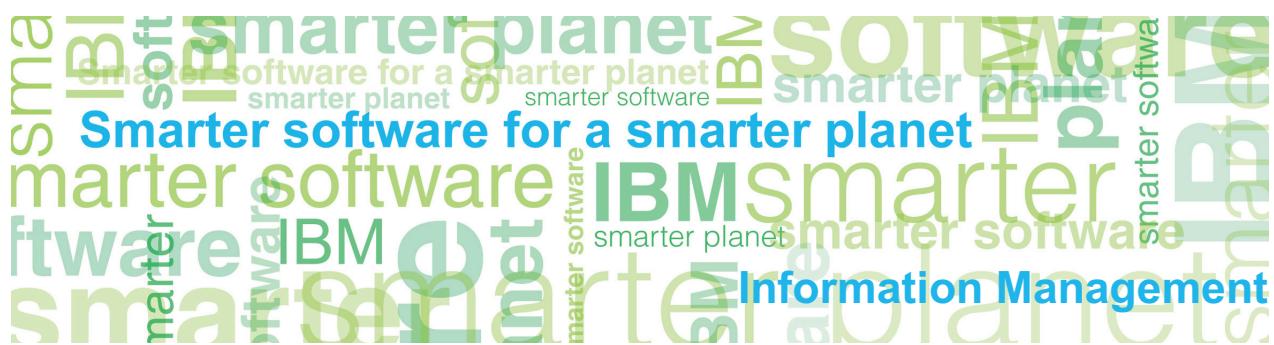
Questions?

Winter/Spring 2011

E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)  
 Subject: “DB2 Workshop for Oracle Professionals”



# DB2® Fundamentals for Oracle® Professionals



© 2011 IBM Corporation

Information Management



## Agenda

- **Product Overview**
  - Editions & Features
  - Licensing
- **Fundamentals**
  - Architecture
  - Users
  - Instances
  - DB2 Client
  - Storage
    - Table spaces
    - Buffer pools
  - XML Storage
  - Security
- **Data Studio**
- **DSHM**

Break free with DB2



## DB2 Product Overview

- **Officially released June 2009**
  - FP1 released Dec 2009
  - FP2 released May 2010
  - FP3 released Sep 2010
- **Full Multi-Platform Support**
  - Linux, UNIX (AIX, HP-UX)
  - Windows 2000, 2003, 2008, XP, Vista, 7
  - Solaris
- **Common code base “DB2 is DB2 is DB2”**
  - No need to port between platforms
  - New versions available on all platforms at the same time



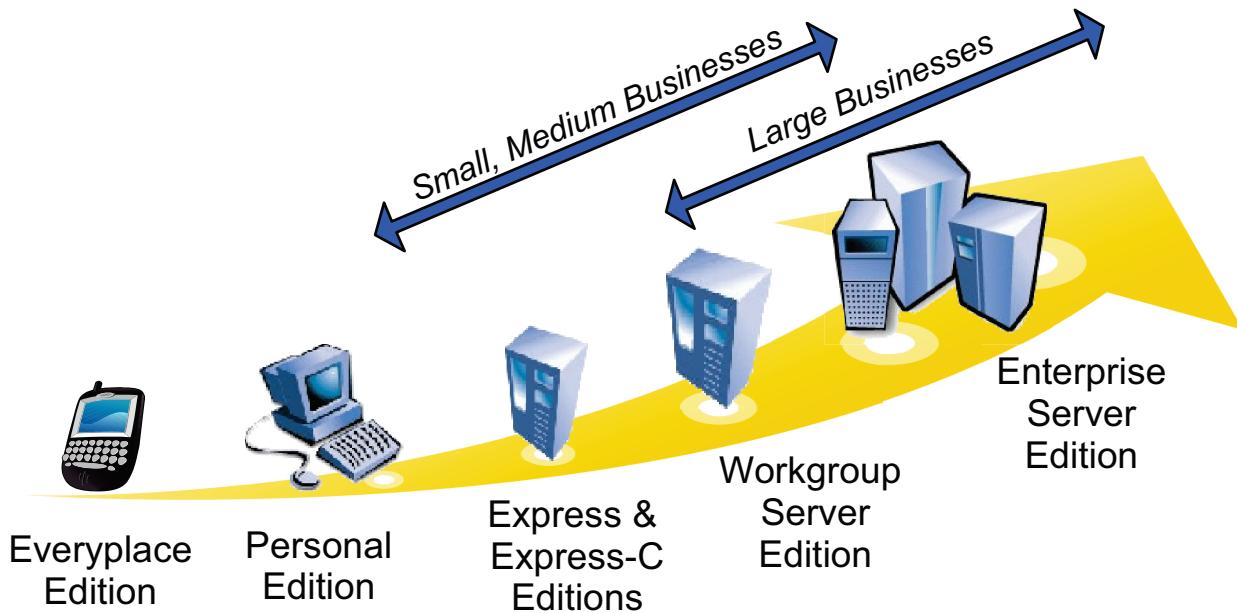
DB2 LUW main site:

<http://www-01.ibm.com/software/data/db2/linux-unix-windows/>

3

© 2011 IBM Corporation

## DB2 Packaging and Editions



### Database Enterprise Developer Edition

Allows developers to develop and perform quality assurance. The edition is a product bundle that includes many DB2 features.

## DB2 vs. Oracle Editions

- **DB2 Express-C (FREE)**
  - Windows, UNIX, Linux, Solaris
  - 2 GB memory, 2 CPU
  - No Database size limit
  - Includes pureXML
- **DB2 Express Edition**
  - Windows, Linux, UNIX
  - 4 GB memory, 4 CPU
  - Includes HADR (FTL)
- **DB2 Workgroup Server Edition**
  - Windows, Linux, Solaris
  - 16 GB memory, 4 sockets
  - Includes HADR / pureXML
- **DB2 Enterprise Server Edition**
  - Windows, Linux, Solaris, IBM AIX®, HP-UX, Linux on system z
  - No limit on memory and CPU
- **Oracle Express Edition (FREE)**
  - Windows, Linux, UNIX
  - 1GB memory, 1 CPU
  - Up to 4GB user data
- **Oracle Standard One Edition**
  - Windows, Linux, UNIX
  - 2 sockets
  - Does not include RAC
- **Oracle Standard Edition**
  - Windows, Linux, UNIX
  - 4 sockets
- **Oracle Enterprise Edition**
  - Windows, Linux, UNIX
  - No limit on memory and CPU

**DB2 Advanced Enterprise Server Edition**  
**LBAC, Compression, WLM and Optim Performance Manager**  
**Federation: DB2 ↔ Oracle**  
**Active/Active HA: Q-Replication between DB2 servers**



**huge value for  
low cost!**

5

© 2011 IBM Corporation

## DB2 Features and Functionality by Edition

Features	Express FTL	Express	Workgroup Server Edition	Enterprise	Advanced Enterprise
Label Based Access Control	No	No	No	Advanced Access Control Feature	Yes
Geodetic Feature	No	No	No	Geodetic Data Management Feature	Geodetic Data Management Feature
Compression: data / index / XML / temp tables	No	No	No	Storage Optimization Feature	Yes
Workload Management	No	No	No	Performance Optimization Feature	Yes
Query Patroller	No	No	No		No
Performance Expert	No	No	No		No
Homogeneous Q Replication	No	No	No	Homogeneous Replication Feature for ESE	Replicates single DB2 for LUW source data server to single DB2 for LUW target data server.
Connection Concentrator	No	No	No	Yes	Yes
DB2 Governor	No	No	No	Yes	Yes
Materialized Query Tables (MQT)	No	No	No	Yes	Yes
Multi Dimensional Clustering (MDC) tables	No	No	No	Yes	Yes
Query Parallelism	No	No	No	Yes	Yes
Table Partitioning	No	No	No	Yes	Yes
Advanced Copy Services	Yes	High Availability Feature	Yes	Yes	Yes
High Availability Disaster Recovery	Yes		Yes	Yes	Yes
Online Reorganization	Yes		Yes	Yes	Yes
Tivoli System Automation	Yes		Yes	Yes	Yes
Compression: backup	Yes	Yes	Yes	Yes	Yes
Homogeneous Federation	Yes	Yes	Yes	Yes	Between Oracle and DB2 only
Homogeneous SQL Replication	Yes	Yes	Yes	Yes	Yes
Net Search Extender	Yes	Yes	Yes	Yes	Yes
pureXML storage	Yes	Yes	Yes	Yes	Yes
Spatial Extender	Yes	Yes	Yes	Yes	Yes
Optim Development Studio	Yes	Yes	Yes	Yes	Yes
Optim Database Administrator	Yes	Yes	Yes	Yes	Yes

DB2 Workshop for Oracle Professionals page 13 of 207.

# Licensing

## DB2 Database editions can be purchased by:

- **Authorized user**
    - License type: "Authorized User Option"
  - **Processor Value Unit (PVU)**
    - License type: "CPU Option"
  - **Per Socket**
    - Workgroup edition
  - **Per Server (virtual / physical)**
    - Express edition

**Check the type of license being used:**

- 1) Use command **db2licm -l**
  - 2) Licensing center

# Licensing:: Metrics and Summary

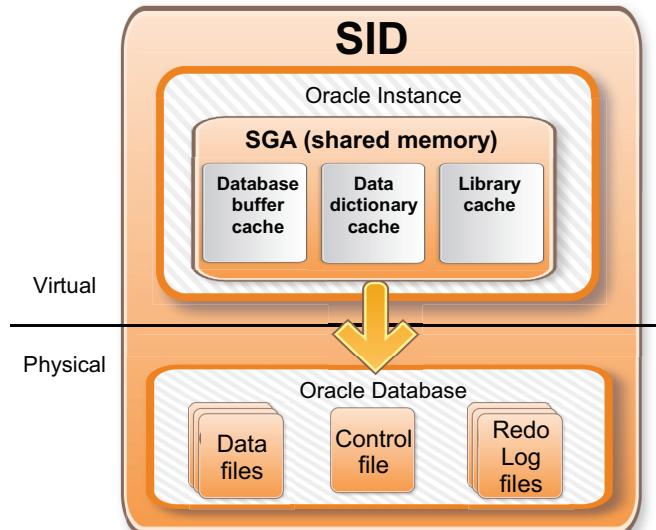
	Personal	Express-C	Express	Workgroup	Enterprise
Pricing metric	Per install (Assumes one user)	Free Download (Unsupported)	Authorized Users (minimum of 5 per server) or <b>Per Server</b> or <b>PVUs (limited to 200 PVUs)</b> Eligible for Sub-capacity pricing	Authorized Users (minimum of 5 per socket) or <b>Per Socket</b> or <b>PVUs (limited to 480 PVUs)</b> Eligible for Sub-capacity pricing	Authorized Users (minimum of 25 per 100 PVUs) or <b>PVUs</b> Eligible for Sub-capacity pricing
Processor limit	N/A	DB2 throttles itself to use maximum of 2 cores	DB2 throttles itself to use maximum of 4 cores	DB2 throttles itself to use maximum of 16 cores and 4 sockets	No Limit
Memory limit	N/A	DB2 throttles itself to use maximum of 2 GB	DB2 throttles itself to use a maximum of 4GB	DB2 throttles itself to use a maximum of 16GB	No Limit
Platforms supported	Windows & Linux	Windows, Linux, Solaris (x64)	Windows & Linux	Windows, Linux, AIX, Solaris, HP-UX	Windows, Linux, AIX, Solaris, HP-UX

## Tools & Products Comparison

Oracle	IBM	Description
• Oracle Enterprise Manager • Enterprise Login Assist • Database Configuration Assistant • Administration Assistant	• DB2 Control Center • IBM Data Studio • DB2 Recovery Expert • Optim Performance Manager • IBM Optim Database Administrator	Database Management
• SQL Developer	• Command Editor • IBM Data Studio	SQL and Procedure developer
• Oracle SQL Plus	• CLPPlus • Command Line Processor (CLP)	Command line administration tool
• Oracle NET Configuration Assistant • Oracle NET Manager	• DB2 Configuration Assistant	Node cataloging / connecting, networking
• Oracle SQL Analyzer	• DB2 Visual Explain • Db2xfmt • db2expln	Query performance analysis Query visual explanation
• Oracle Automatic Database Diagnostic Monitor	• Data Studio Health Monitor • DB2 Event Monitor & Analyzer	Database event information and statistic collection
• Oracle Database Resource Manager	• DB2 Workload Manager • DB2 Governor • DB2 Query Patroller	Workload management
• Oracle Data Guard	• High Availability and Disaster Recovery Feature	High Availability and Disaster Recovery

## Server Architecture Comparisons

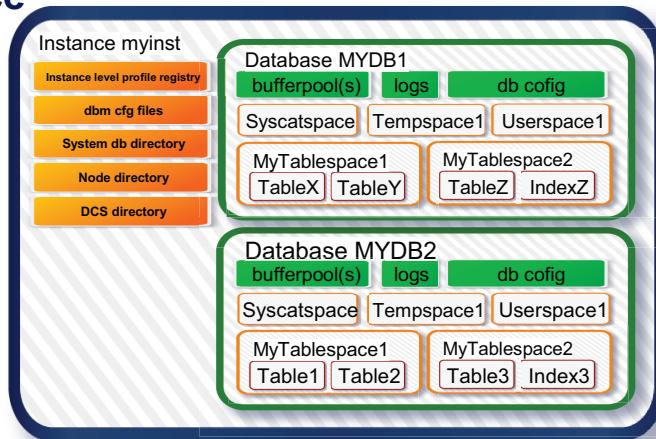
- Oracle System ID (Non RAC)



- An Oracle SID can access **only one database at a time**
- Common terms:
  - “**Instance**” Memory structure and the background processes used to access data from a database
  - “**Database**” physical files used to store information

## Server Architecture Comparisons

### ▪ DB2 Instance



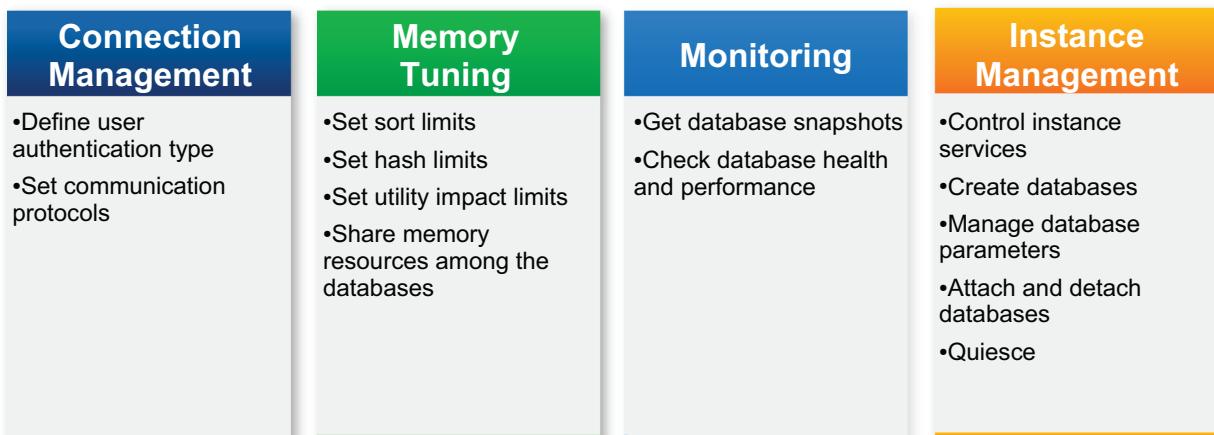
- DB2 server is able to host **multiple instances** and each instance can have and access **more than one database at a time**
- It is the access point to the databases logical & physical structures
- Each DB2 instance has:
  - Multiple Engine Dispatchable Units (EDUs) shared among the databases in that instance
  - Its own configuration file (dbm cfg)

11

© 2011 IBM Corporation

## DB2 Instance (a.k.a. DBM or Database Manager)

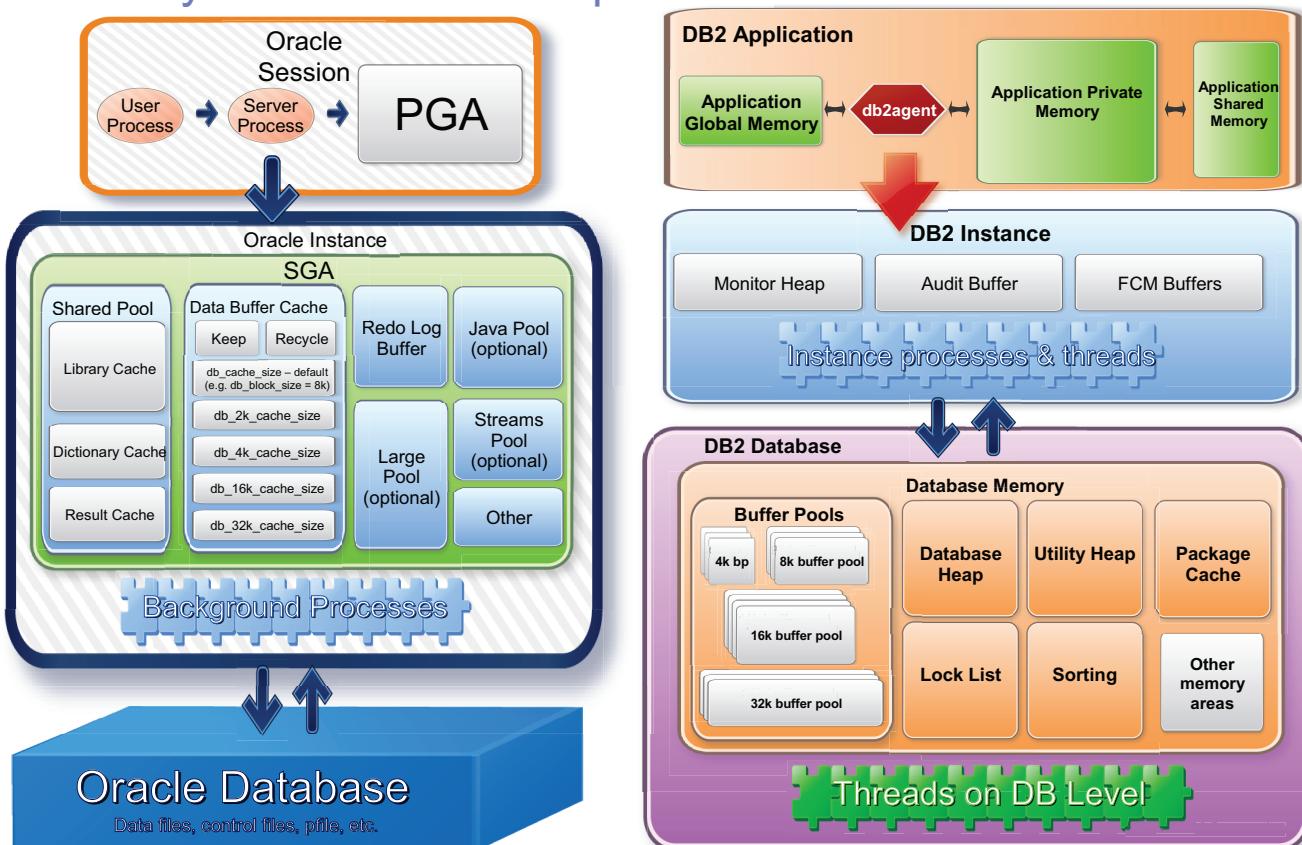
### ▪ What you can do:



**AND MUCH MORE!**



## Memory Architecture Comparisons



## Some DB2 and Oracle Term Mapping

Oracle	DB2
Dynamic views (V_\$ views)	Administration views (SYSIBMADM)
Data cache	Buffer pool
Large pool	Utility Heap
Data dictionary cache	Catalog cache
Autoextend	Autosize
Data dictionary	System catalog
Data file	Container
Alert Log	db2diag
Pfile / spfile (init.ora)	DB / DBM configuration
Transparent appl. failover	Automatic client reroute

# DB2 Installation and Requirements

## Installation Methods

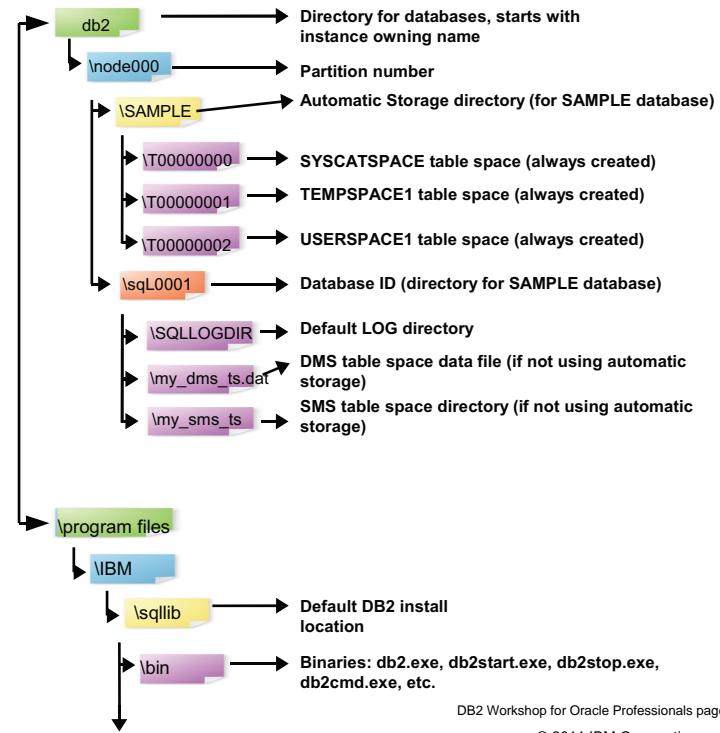
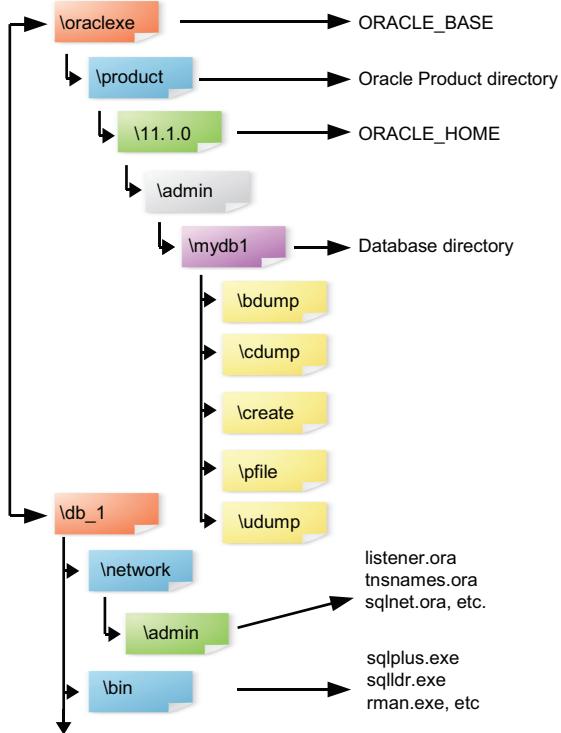
Installation	Windows	UNIX
db2setup Wizard	✓	✓
db2_install command	✗	✓
Response file	✓	✓

## Minimum Requirements

Resource	Oracle	DB2
CPU	1Ghz processor	1Ghz processor
RAM	1GB	512MB
Temp space	200MB	60MB

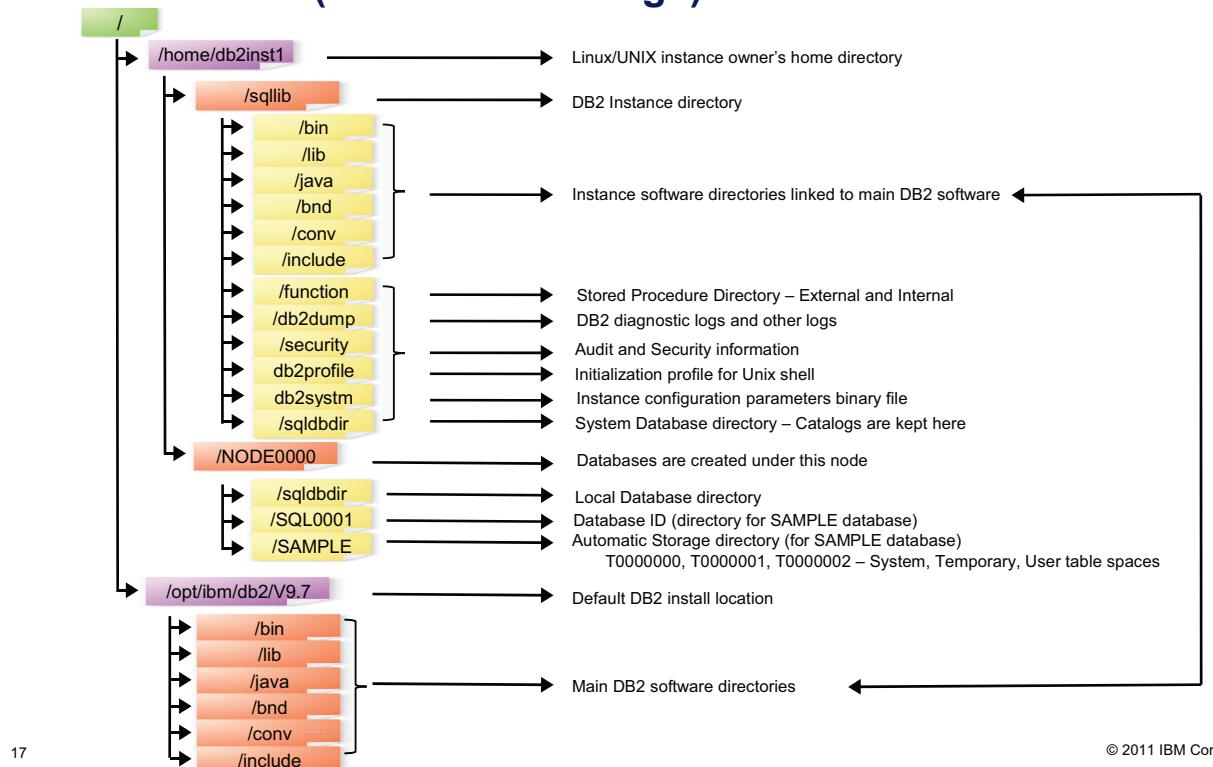
# Directory Structure Comparison

## Windows



## Directory Structure - DB2

### ▪ Linux / UNIX (Automatic Storage)



17

© 2011 IBM Corporation

## Parameter Configuration Comparison

Oracle	DB2
Oracle 11g contains nearly 300 parameters	DB2 9.7 has nearly 210 parameters
PFILEs are necessary for database creation	DB2 <b>automatically generates</b> all parameters during instance / database creation
Oracle needs either PFILES or templates to create a database	DB2 <b>AUTOCONFIGURE</b> feature suggests database parameter values at database creation
PFILE to SPFILE conversion is necessary otherwise the parameter changes won't persist when you use the ALTER command.	No conversion is necessary. <b>The DB and DBM have that functionality by default</b>

# Registry / Environment Variables Comparison

## ▪ DB2 registry variables

– Managed by using **db2set** or the **Configuration Assistant**

Variable	Function
db2adminserver	Administration server instance
db2comm	Communications manager
db2include	SQL searches path
db2instance	Current instance
db2instdef	Default instance
db2path	DB2 directory

## ▪ Oracle environment variables

Variable	Function
ora_cwd	Current working directory
ora_sid_pfile	Path to initialization parameter file (init.ora)
oracle_home	Home directory in which Oracle is installed
oracle_sid	Oracle DB instance on the host computer
oracle_base	Top-level Oracle directory

### db2set command

db2set -h	Help
db2set db2comm=tcpip	Assign a value to a variable
db2set db2comm=	Resets a variable to its default
db2set -all	Displays all variables

```
[e] DB2PATH=C:\Program Files\IBM\SQLLIB
[i] DB2ACCOUNTNAME=IBM-UPDT123H3BR\db2admin
[i] DB2INSTOWNER=IBM-UPDT123H3BR
[i] DB2PORTRANGE=60000:60003
[i] DB2INSTPROF=C:\PROGRA~1\IBMSQLLIB
[i] DB2COMM=TCP/IP
[g] DB2_USERSGROUP=DB2USERS
[g] DB2_ADMINGROUP=DB2ADMNS
[g] DB2_EXTSECURITY=YES
[g] DB2SYSTEM=IBM-UPDT123H3BR
[g] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2INSTDEF=DB2
[g] DB2ADMINSERVER=DB2DAS00
```

[e] Environment Variables – OS controlled  
 [g] Global Registry Variables – effects entire server  
 [i] Instance Registry Variables – effects current instance

# Database Creation

## ▪ Oracle

- 1) Set environment variables
  - ORACLE\_SID, ORACLE\_BASE, ORACLE\_HOME
- 2) Create password file
- 3) Create pfile
- 4) STARTUP NOMOUNT
- 5) CREATE DATABASE...
- 6) Create data dictionary views
- 7) Create PL/SQL packages
- 8) Add the SID to tnames.ora

DB2 uses external authentication, no password file necessary

## ▪ DB2

- 1) Set registry variables
  - DB2INST, DB2INSTPROF
- 2) DB2START
- 3) CREATE DATABASE...
- 4) CATALOG DATABASE (Optional)

DB2 uses “autoconfigure” feature to set the variables. No files (like Oracle pfile's) are needed

## DB2

DB2 automatically enable Self Tuning Manager enabled by default

During the database creation, when the compatibility vector is enabled, DB2 automatically builds the data dictionary views and PL/SQL packages

## Cataloguing a Database:: Setting Up Communications

- To use a remote database:

1. Catalog the remote system (node)
  2. Catalog the database within the remote node

- 3 tools for the job:

- Configuration Assistant (graphical)
  - Data Studio (graphical)
  - Command Line Processor (CLP)

```
db2 list [database|node]  
directory
```

can be used to find the locally catalogued DB or node

The diagram illustrates two catalog commands with their corresponding parameters:

- catalog tcpip node** *db2\_node* **remote** *mysystem* **server** *db2tcp42*

  - db2\_node* is labeled as **node\_name**.
  - db2tcp42* is labeled as **service\_name or port\_number**.
  - mysystem* is labeled as **host\_name or ip\_addr**.

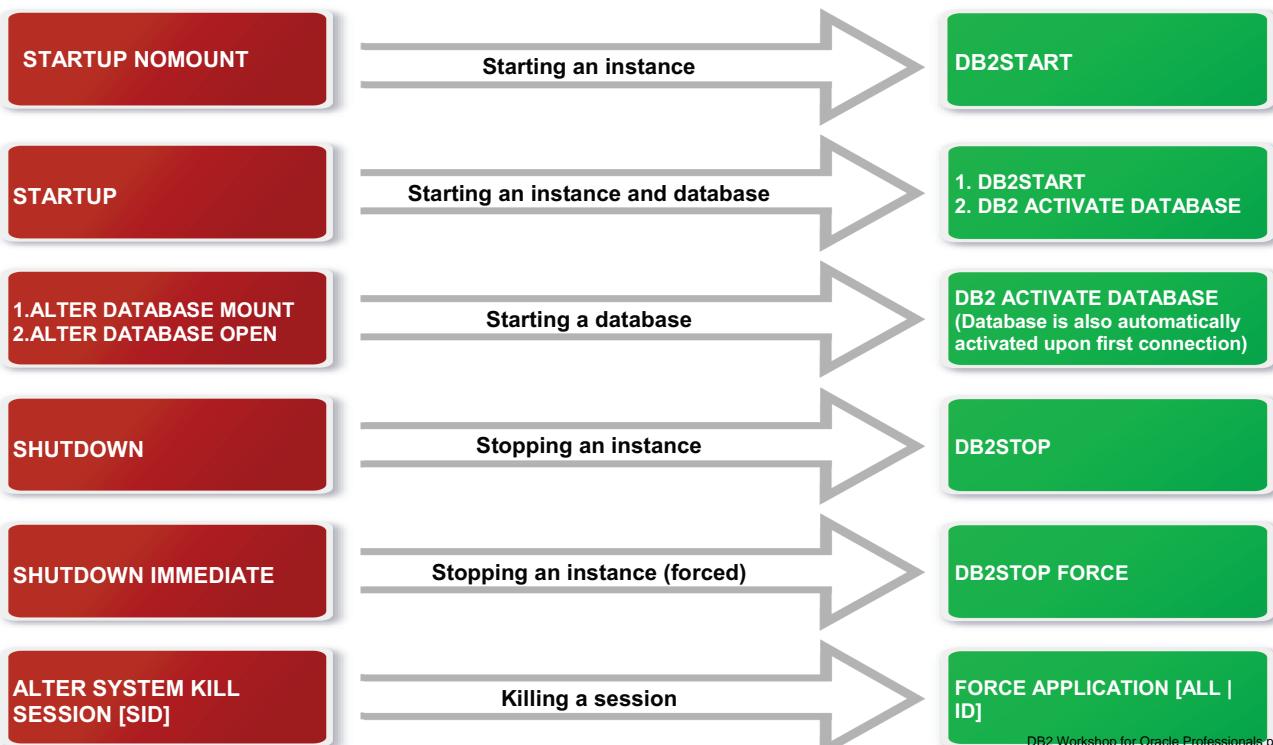
- catalog database** *sample* **as** *mysample* **at** **node** *db2node*  
  **authentication** *server*

  - sample* is labeled as **database\_name**.
  - mysample* is labeled as **database\_alias**.
  - db2node* is labeled as **node\_name**.
  - server* is labeled as **authentication\_type**.

21

© 2011 IBM Corporation

# Instance & Database Administration



## Instance & Database Administration

- **Other commands**

Command	Description	Example
db2start	Start the default instance	db2start
db2stop	Stop the current instance	db2stop -f
db2icrt	Create an instance	db2icrt -u db2fenc1 db2inst1
db2idrop	Drop an instance	db2idrop -f db2inst1
db2ilist	List all instances	db2ilist
db2imigr	Migrate an instance after upgrading DB2	db2imigr -u db2fenc1 db2inst1
db2iupd	Update an instance after installation of a fix pack	db2iupd -u db2fenc1 db2inst1

Description	Example
View Database Manager Settings	db2 get dbm cfg show detail
Change a Database Manager Setting	db2 update dbm cfg using health_mon off

Description	Example
View Database Settings	db2 get db cfg for testdb db2 connect to testdb db2 get db cfg show detail
Change a DB Setting	db2 update db cfg using logprimary 10

## DB2 Sample Database

- To create the sample database populated with both relational data and XML data
- Verify the database creation by simply connecting and querying the data

```
db2sampl -sql -xml
db2 connect to sample
```

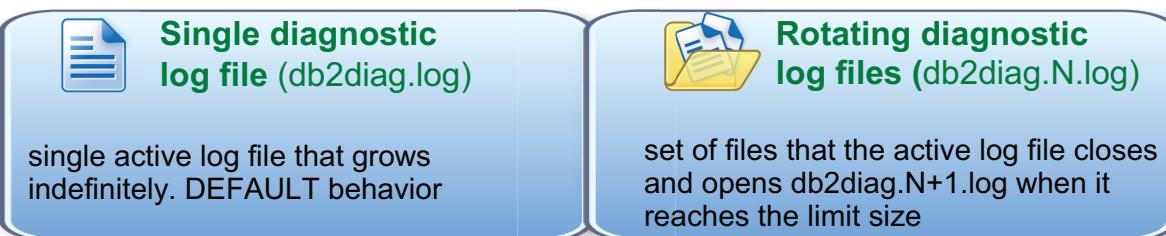
- For remote databases:

```
db2 catalog database sample as db_sample
at node mynode1
db2 connect to db_sample
```

## Logging:: db2diag.log

- Trouble shooting and diagnostic purposes
- Located in **\$DB2INSTANCE\_HOME/sql1lib/db2dump/** by default
- General log which contains all DB2 errors and warnings
- Use “**db2diag**” tool to filter and forma db2diag log files

2 forms:



### Configuration parameters:

- **Diagsize:** size of the log files for rotating log files form; 0 for single log file form
- **Diagpath:** Location of the log file(s)
- **Diaglevel:** Types of errors to be written to log

## Oracle Dictionary vs. DB2 System Catalog

**Oracle dictionary views will be created by setting the DB2\_COMPATIBILITY\_VECTOR registry variable**

### Oracle Data Dictionary

SYS.xx\$ - Base tables

SYS.GV\_\$ - Admin views

SYS.V\_\$ - Admin views

USER\_\$ - Users views

ALL\_\$ - Expanded users views

DBA\_\$ - Database Admin views

`SELECT <column> FROM user_tables;`

`SELECT <column> FROM user_views;`

### DB2 System Catalog

SYSIBM - Base tables

SYSIBMADM – Admin views

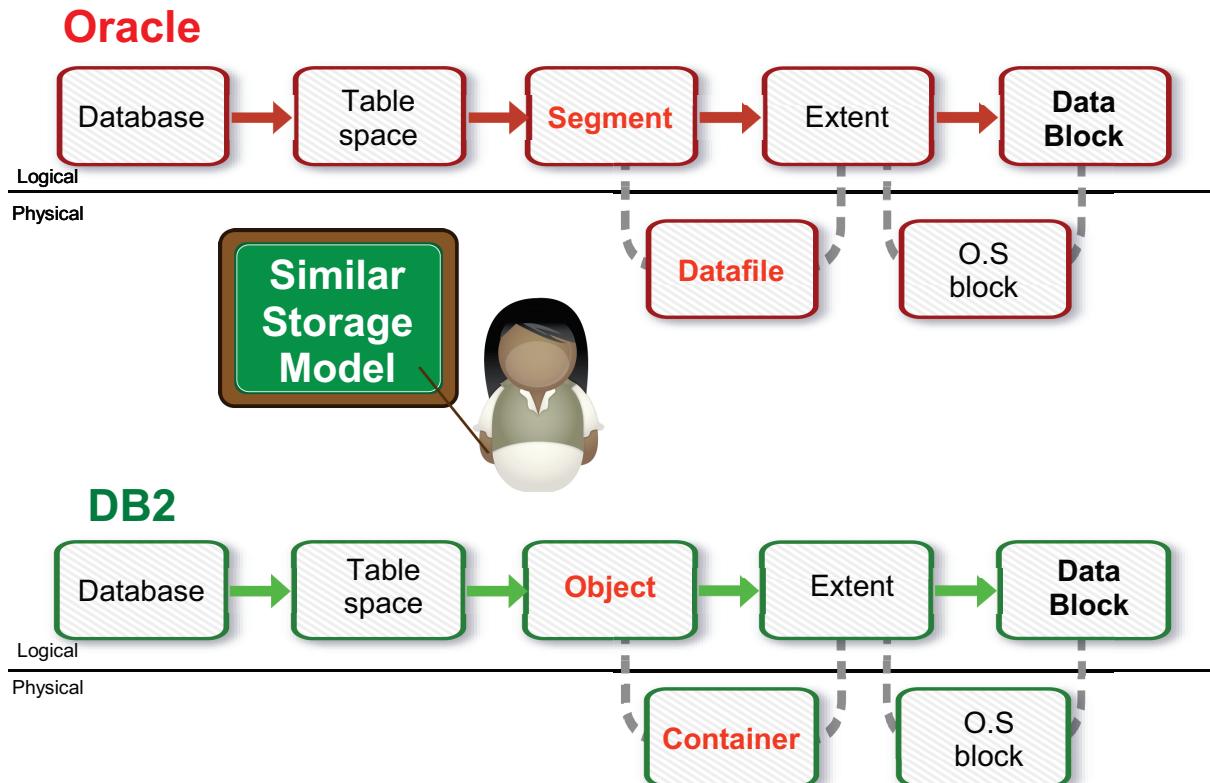
SYSCAT – Catalog base tables

SYSSSTAT – Updateable views

```
SELECT <column> FROM syscat.tables
WHERE OWNER=user AND
TYPE='T';
```

```
SELECT <column> FROM syscat.tables
WHERE OWNER=user AND
TYPE='V';
```

## Storage Model Comparison

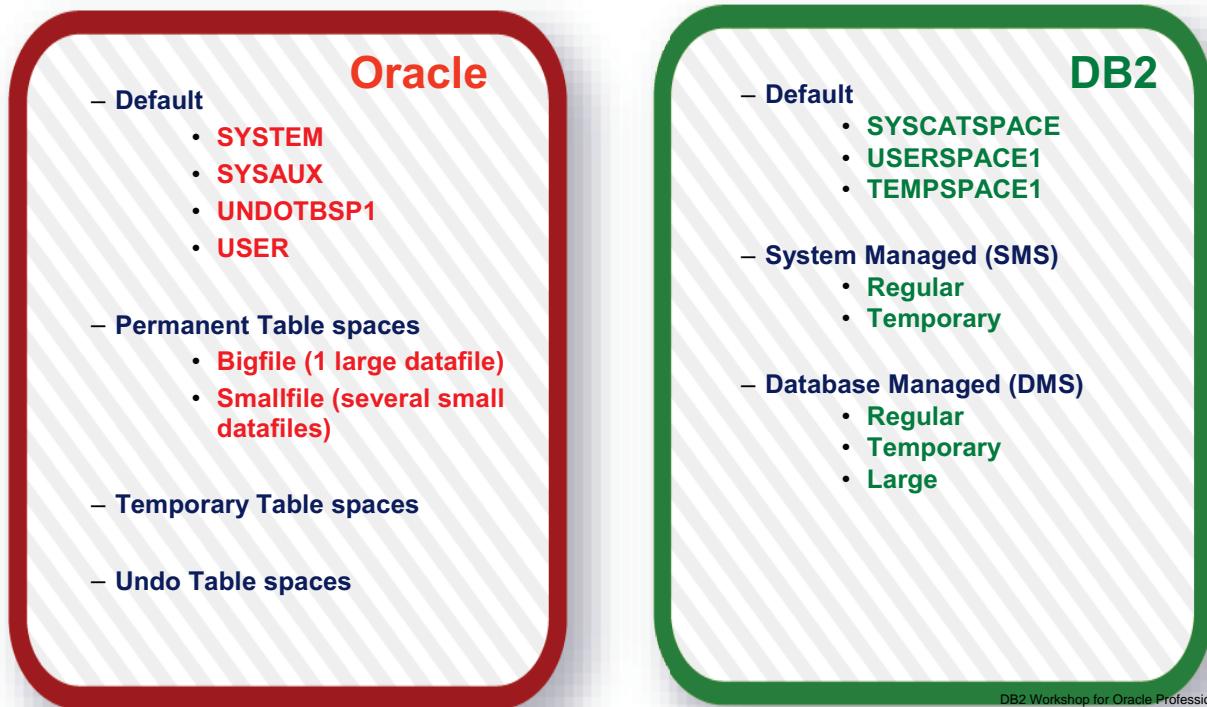


27

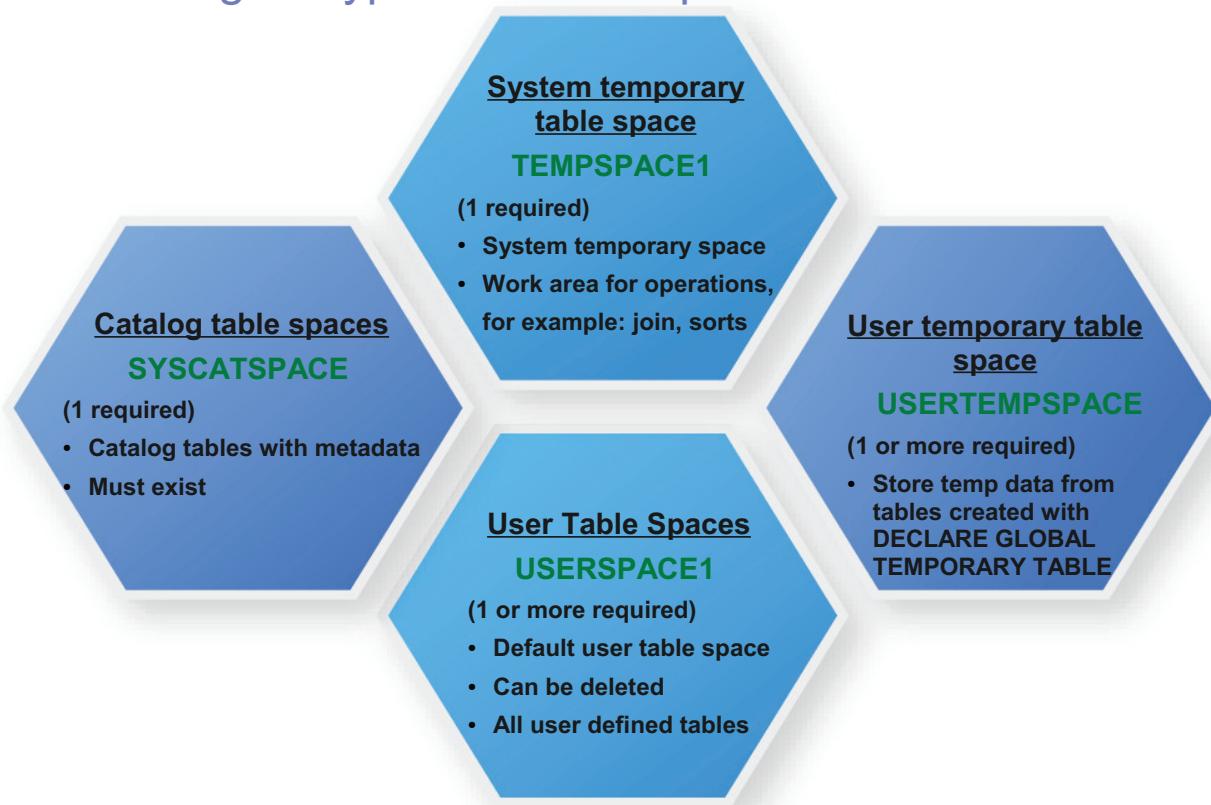
© 2011 IBM Corporation

## Storage Model Comparison – Table Spaces

### ▪ Table Spaces



## DB2 Storage:: Types of Table Spaces



29

© 2011 IBM Corporation

## DB2 Storage:: Table Space Management

### ▪ System Managed (SMS)

- Oracle has NO equivalent!
- Data stored in files representing data objects
- Space is allocated on demand
- Access to data controlled using standard I/O functions of the OS
- ✓ Ideal for small, personal databases and databases that grow/shrink rapidly
- ✗ Lower performance, as the OS handles the I/O

```
CREATE TABLESPACE byoracle MANAGED BY SYSTEM
  USING ('d:\acc_tbsp', 'e:\acc_tbsp', 'f:\acc_tbsp')
```

### ▪ Database Managed (DMS)

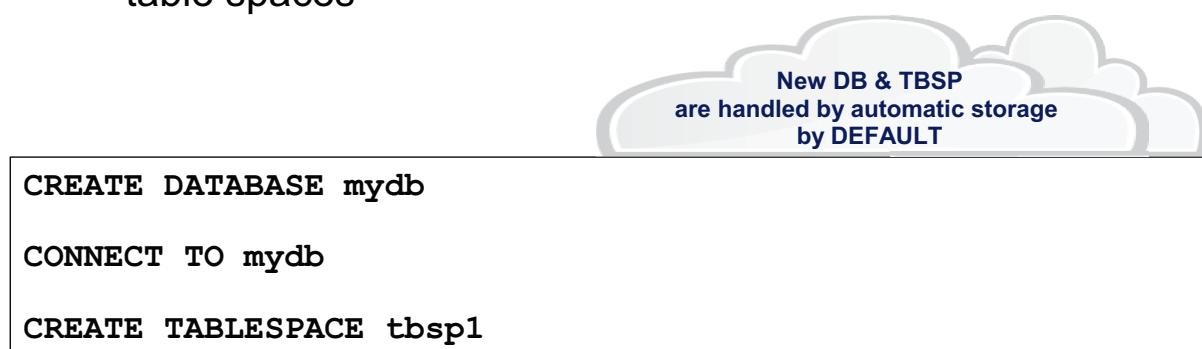
- All Oracle table spaces works like DB2's DMS
- Data stored in files or on raw devices
- Storage space pre-allocated in file system, typically contiguous physically
- ✓ Ideal for performance-sensitive applications
- ✗ Increased maintenance and monitoring

```
CREATE TABLESPACE byoracle MANAGED BY DATABASE
  USING (FILE '/db2/file1' 1 M, FILE '/db2/file2' 1 M)
  AUTORESIZE YES
  INCREASESIZE 2 M
  MAXSIZE 100 M
```

## DB2 Storage:: Table Space Management

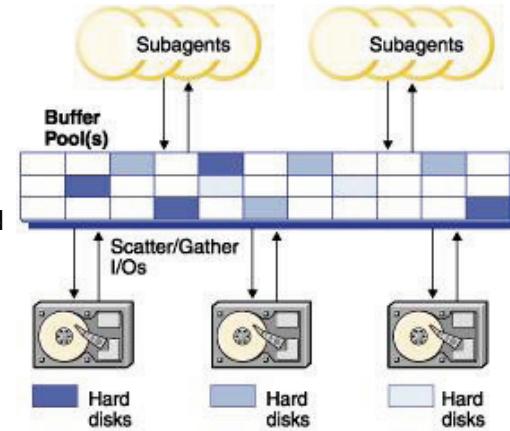
- **Automatic Storage Table Space**

- DBM creates and extends containers as needed up the limits imposed by the storage paths associated with the database
- Automatically handles resizing table spaces
- Creates a DMS table space for regular/large table spaces
- Creates a SMS table space for user or system temporary table spaces



## DB2 Storage:: Buffer Pools

- **Area of main memory used to cache table and index data**
- **Each database must have at least one buffer pool**
  - By default IBMDEFAULTTBP is used
  - Buffer pools can be created, dropped or altered
  - SYSCAT.BUFFERPOOLS catalog view accesses the information for the buffer pools defined in the database
- **Every table space associates a specific buffer pool of the same page size**
  - Match buffer pool size with purpose of table to increase hit ratio
- **Self-Tuning Memory Manager (STMM) available**



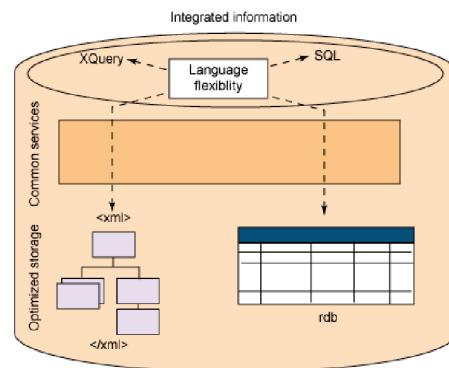
```

CREATE BUFFERPOOL bp4k PAGESIZE 4K
CREATE TABLESPACE tbsp1 PAGESIZE 4K BUFFERPOOL bp4k

```

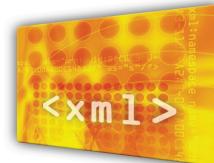
## pureXML & XQuery

- DB2 is fully hybrid DBMS with pureXML technology
  - Native storage of XML data type
- XQuery can be used for querying and modifying XML data
  - Search for objects that are at unknown levels of the hierarchy.
  - Perform structural transformations on the data
  - Return results that have mixed types.
  - Update existing XML data



```
xquery db2-fn:xmlcolumn ("XMLCUSTOMER.INFO");
```

```
xquery
for $p in db2-fn:xmlcolumn("XMLPRODUCT.DESCRIPTION") /product
let $limit := 0.05
where $p/description/price > $limit
order by $p/data(@pid) descending
return ( $p/description/name )
```



Returns xml data in the column

Retrieve all XML documents from an XML column,  
then process them with an XQuery expression

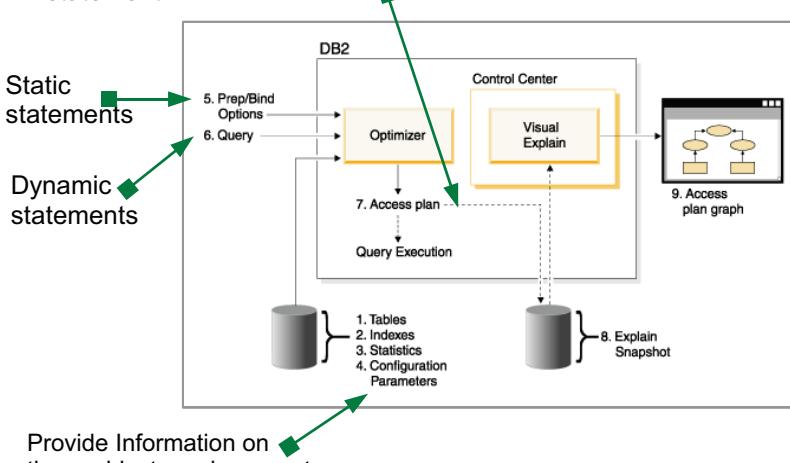
© 2011 IBM Corporation

33

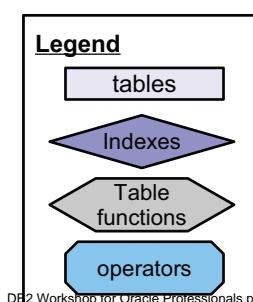
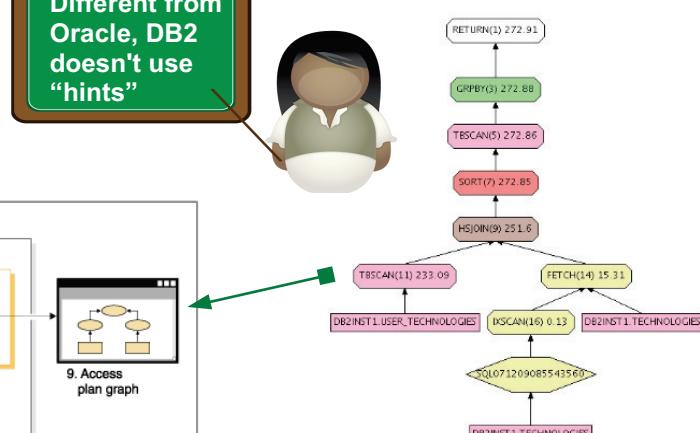
## DB2 Access Plan

An access plan specifies the order of operations for accessing data necessary to resolve a SQL or XQuery statement

Different from Oracle, DB2 doesn't use "hints"



DB2 automatically detects outdated statistics and update them on-the-fly!



# DB2 Data Server Clients

## IBM Data Server Runtime Client

- JDBC, ADO.NET, OLE DB, ODBC, CLI, PHP, and Ruby
- Has CLP but GUI tools are not included
- Support LDAP exploitation, TCP/IP and Named Pipe, cataloging

## IBM Data Server Client

- All the functionality of IBM Data Server Runtime Client
- Plus functionality for database administration, application development, and client/server configuration.
- Capabilities include GUI tools such as configuration assistant, control center, visual studio tools

## IBM Data Server Drivers

- Light weight deployment solution for ISVs
- Must be installed manually
- Can be embedded in applications and freely redistributed

### IBM Data Server Driver for JDBC and SQLJ

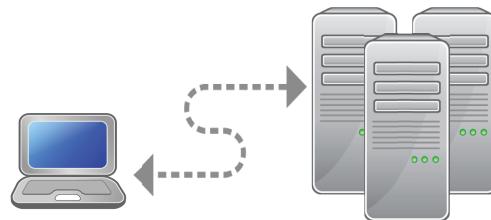
- Java stored procedures and user-defined functions
- JDBC, SQLJ

### IBM Data Server Driver for ODBC and CLI

- ODBC API, or CLI API

### IBM Data Server Driver Package

- ODBC, CLI, .NET, OLE DB, PHP, Ruby, JDBC, or SQLJ



© 2011 IBM Corporation

# Export, Import and Load Utility

## Oracle tools

- Exporting data: Oracle exp, Data Pump
- Importing data: Oracle imp, Data Pump

## DB2 Export Utility

- Move data from table or view to files

## DB2 Import Utility

- Performs SQL INSERTs

## DB2 Load Utility

- Moving large quantities of data
- Writes formatted pages directly into the database
- Does not fire triggers, and does not perform referential or table constraints
- Handles most data types: XML, LOBs, UDTs
- 4 distinctive phases: **LOAD, BUILD, DELETE, INDEX COPY**

## DB2MOVE Utility

- **COPY:** Copy/move schemas (with or without data) from a source database to a local target database
- **IMPORT/EXPORT:** Move large numbers of tables between DB2 databases

## DB2LOOK Utility

- Extracts DDL statements object by object



## Data Studio 2.2 Overview



- **No-charge Integrated Development Environment (IDE)**

- Geared towards application developers and DBAs
- Supports DB2 for LUW, i5/OS and z/OS, Apache Derby, Informix IDS, and others

- **Benefits**

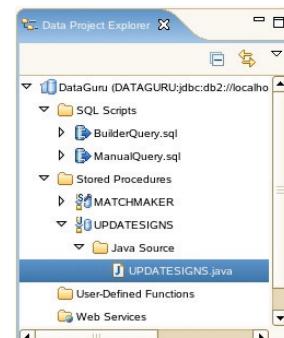
- Integrates features previously available in separate tools to minimize context switching
- Built on the Eclipse platform, offers low learning curve
- Simplifies development and administration functionality to increase productivity for all roles throughout the data life cycle

Download now at <http://www.ibm.com/software/data/studio>

## Key Features

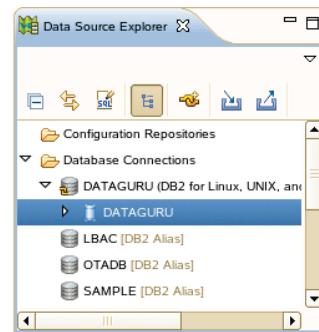
### Data application developer features

- **Routine wizards, editors and debugger** to create, test, debug, and deploy routines, eg. stored procedures and UDFs
- **SQL Query builder** and the SQL and XQuery editor to create, edit, and run SQL queries.
- **Visual Explain** to tune routines and SQL queries
- Create **Web services** that expose database operations to client applications
- **XML Wizards and editors** to develop XML applications
- Develop **SQLJ** applications in a Java project



### Data and database object management features

- Establish **connection to data sources**
- Work with data objects: browse, modify privileges, drop
- **Data object editors and wizards** to create and alter data objects
- **Change impact analysis**
- Work with data: basic support for extracting and loading data
- Use **data diagrams** to visualize the relationships between data objects



# Integrated Data Management (IDM) Portfolio

- **IBM Optim Integrated Data Management solutions**

- Manage data from requirements to retirement
- Boost performance
- Empower collaboration
- Improve governance across applications, databases and platforms.



- **Integrated Data Management Information Center**

<http://publib.boulder.ibm.com/infocenter/idm/v2r2/index.jsp>

- **In addition to Data Studio for development, there are paid editions with additional functionality**

★ **Optim Development Studio 2.2**

- Create and test database and pureQuery applications
- Support for Oracle databases

★ **Optim Database Administrator 2.2**

- Automates and simplifies complex database structural changes

## Data Studio Health Monitor (DSHM)

- Web based tool for database health monitoring
- Provides a single portal for viewing the status of all your databases
- Available for Linux, UNIX and Windows
- Available as a free download from:

[http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.db.web.health.doc/topics/helpindex\\_dshm.html](http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.db.web.health.doc/topics/helpindex_dshm.html)

Add connection by specifying the database name, host, port, user and password

Connection Name	Database Name	Data Server Type	Host Name and Port	Credentials	Data Server Monitoring	Last Update	Comment
SAMPLE	SAMPLE	DB2 for LUW	localhost:50011	No	Disabled	2008-12-27 16:42:29	

## DSHM Capabilities

**Health Summary** tab quickly summarizes the status of all databases monitored by DSHM

The screenshot shows the DSHM interface with three tabs: Health Summary, Alert List, and Dashboard. The Alert List tab is currently selected, showing a message 'No alerts to show.' The Dashboard tab displays various performance metrics with their current values and 90-minute trends.

**Alert List** tab displays the warnings associated for each database

**Dashboard** tab shows various performance metrics and their values

## DB2 Security

### ▪ Authentication vs. Authorization

#### • Authentication

- Determines who the user is by identifying with a password

#### • Authorization

- Grants privileges to that user once authenticated

#### • Explicit vs. Implicit vs. Indirect Access

DB2 use external auth. while Oracle uses internal



Explicit
•User
•Group
•Role

Implicit
•When a database or database object is created

Indirect
•Inherited through execution of packaged code

## Authentication Types

- **With DB2 9.7, the following authentication types are available:**

- SERVER
- SERVER\_ENCRYPT
- CLIENT
- KERBEROS
- KRB\_SERVER\_ENCRYPT
- DATA\_ENCRYPT
- DATA\_ENCRYPT\_CMP
- GSSPLUGIN
- GSS\_SERVER\_ENCRYPT

## Authorities

- **Administrative Authorities**

- SYSADM, SYSCTRL, SYSMAINT, SYSMON
- Eg: SYSADM - control over all resources created and maintained by the Database Manager (instance)

- **Database-level Authorities**

- DBADM, SECADM, SQLADM (New), WLMADM (New), EXPLAIN (New), ACCESSCTRL (New), DATAACCESS (New), etc.
- Eg: WLMADM – allows creation of WLM objects; grant/revoke of WLM privileges; and execution of WLM routines
- Eg: DATAACCESS – allows access to data within a specific database

## Changes to Authorities in DB2 9.7

- **Security administrator (SECADM) abilities have been extended**
  - Now grants and revokes all authorities and privileges, including DBADM and SECADM
  - SECADM authority can be granted to roles and groups
  - SECADM authority is no longer required to run the audit stored procedures and table functions
    - AUDIT\_ARCHIVE AUDIT\_LIST\_LOGS AUDIT\_DELIM\_EXTRACT
  
- **Database administrator (DBADM) authority scope has been reduced**
  - DBADM can only be granted or revoked by the SECADM
  - Holds implicitly the following authorizations: BINDADD, CONNECT, CREATETAB, CREATE\_EXTERNAL\_ROUTINE, CREATE\_NOT\_FENCED\_ROUTINE, IMPLICIT\_SCHEMA, QUIESCE\_CONNECT, SQLADM, EXPLAIN, WLMADM and LOAD
  - When granting DBADM the following options are available:
    - Allow data access: WITH / WITHOUT DATAACCESS option
      - Eg: GRANT DBADM ON DATABASE TO USER U1 WITH DATAACCESS
    - Allow grant and revoke of privileges and authorities WITH / WITHOUT ACCESSCTRL option
      - Eg: GRANT DBADM ON DATABASE TO USER U1 WITHOUT ACCESSCTRL
    - Default is WITH DATAACCESS and ACCESSCTRL

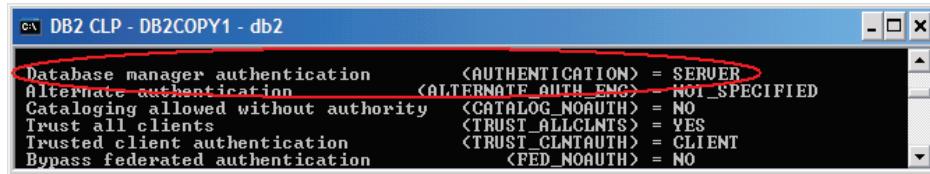
## Changes to Authorities in DB2 9.7

- **System Monitor (SYSMON) authority scope has been extended**
  - Now able to use the db2 memory tracker (db2mtrk) for a report of memory status
  - Now able to use the LIST commands to better monitor the system. Affected list commands:
    - LIST DATABASE PARTITION GROUPS, LIST DRDA INDOUBT TRANSACTIONS, LIST PACKAGES, LIST TABLES, LIST TABLESPACE CONTAINERS, LIST TABLESPACES, LIST UTILITIES
  
- **System Administrator (SYSADM) authority scope has been reduced**
  - No longer has implicit DBADM authority.
  - When a user with SYSADM authority creates a database, DATAACCESS, ACCESSCTRL, SECADM and DBADM authorities are automatically granted, which gives the user the same abilities as in Version 9.5
  - No longer able to grant any authorities or privileges except table space privileges

## Configuration of Authentication on DB2 Server

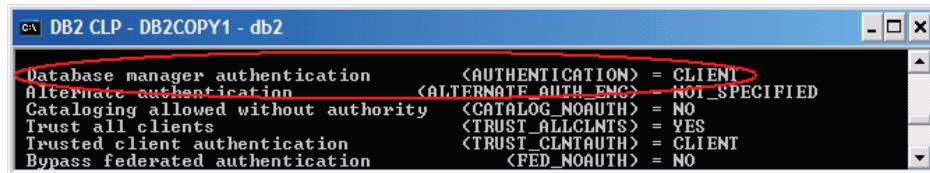
- Authentication type is defined in the Database Manager configuration file (DBM CFG)**

```
db2 "GET DBM CFG"
```



- To configure how and where DB2 authenticates users, set the authentication parameter at the DB2 server**

```
db2 "UPDATE DBM CFG USING AUTHENTICATION CLIENT"
```



## Trusted Context

- Provide a means whereby the end-user identity in a three-tier environment can be easily and efficiently propagated to the database server**
- Introduce concept of a trusted context between a database server and a specific application tier**
- Why not just keep one common user ID?**
  - Loss of user identity for auditing purposes
  - Hard to distinguish actions needed by app vs needed by user
  - Middle tier is “over granted” privileges
  - If ID is compromised, high risk of security exposure

## Trusted Context

- **Implementation Considerations**

- Users need to be identified individually but do not want expensive new connections
- How do we identify a trusted source?

- **Solution: Create a “Trusted Context”**

- A trusted relationship between the DB and the application
  - Switch current user ID
  - Acquire additional privileges via role inheritance

### Relationship identified by connection attributes

- IP Address, Domain Name, Authorization ID, Data Encryption used

```
CREATE TRUSTED CONTEXT ctxt
BASED UPON CONNECTION USING SYSTEM AUTHID smith
ATTRIBUTES (ADDRESS '192.168.2.27')
DEFAULT ROLE managerRole ENABLE
```

Questions?

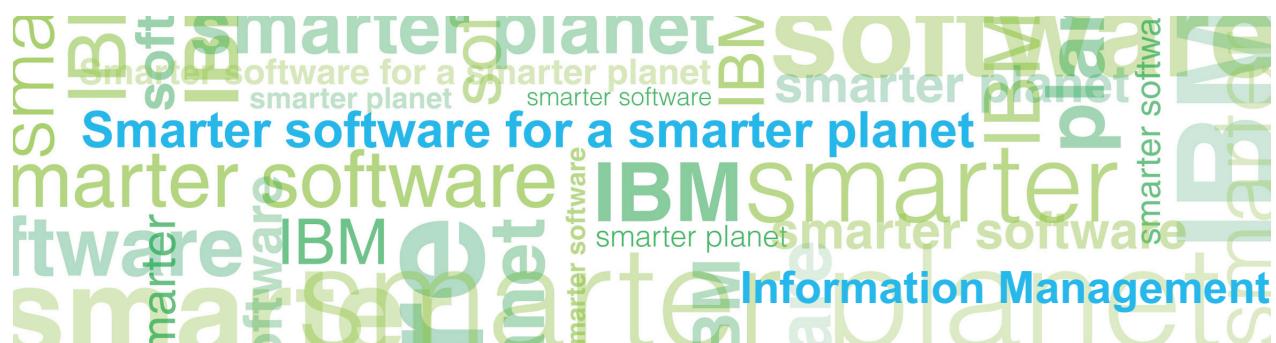
Winter/Spring 2011

E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)

Subject: “DB2 Workshop for Oracle Professionals”



# DB2® Practical Autonomics



© 2011 IBM Corporation

Information Management

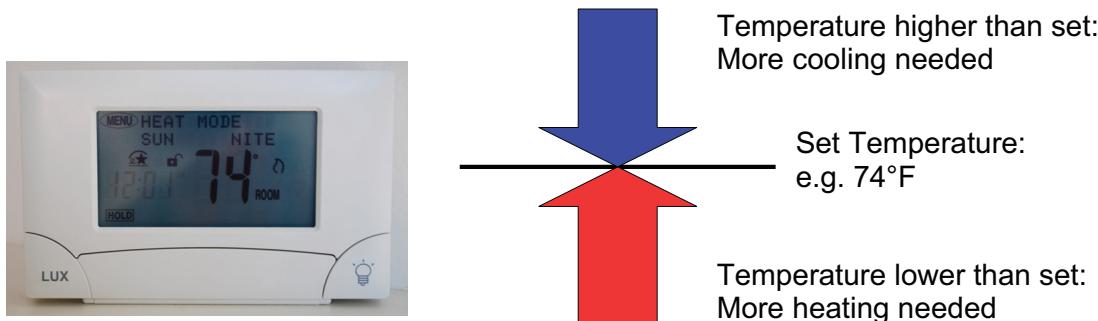


## Practical Autonomics – Agenda

- **Why Autonomics and what is it?**
- **Autonomics Tools and Examples**
  - Configuration and Design Advisors
  - Automatic Object Maintenance
  - Automatic Storage
- **Self-Tuning Memory Manager (STMM)**

## What is autonomic computing?

- Technology features in IT systems that help manage the increasing complexity of technology
- In many ways autonomic computing works with simple control theories
  - Just like a thermostat of a room that senses temperature and adjusts more heating or cooling



– Autonomic computing features adjust the computer system based upon set parameters for normal operation

## Why do we care about autonomics?

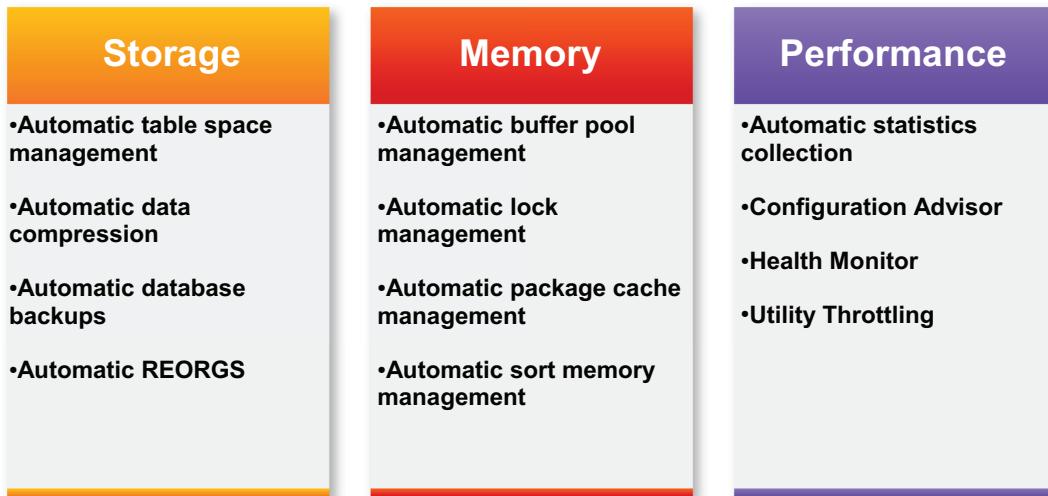
- Focus: Manage your business, not your database!
  - Autonomic features alleviate the need for constant attention by DBAs and other IT personnel
  - Staff can then focus on other important matters
    - Things that can save you money! e.g. more efficient processes, cost-saving projects
    - Increase skills (such as coming to this bootcamp!)
    - ..... (see below) ☺

SIGNS YOUR CODERS DON'T HAVE ENOUGH WORK TO DO:



# Why do we care about autonomies?

- DB2 Autonomics will handle



5

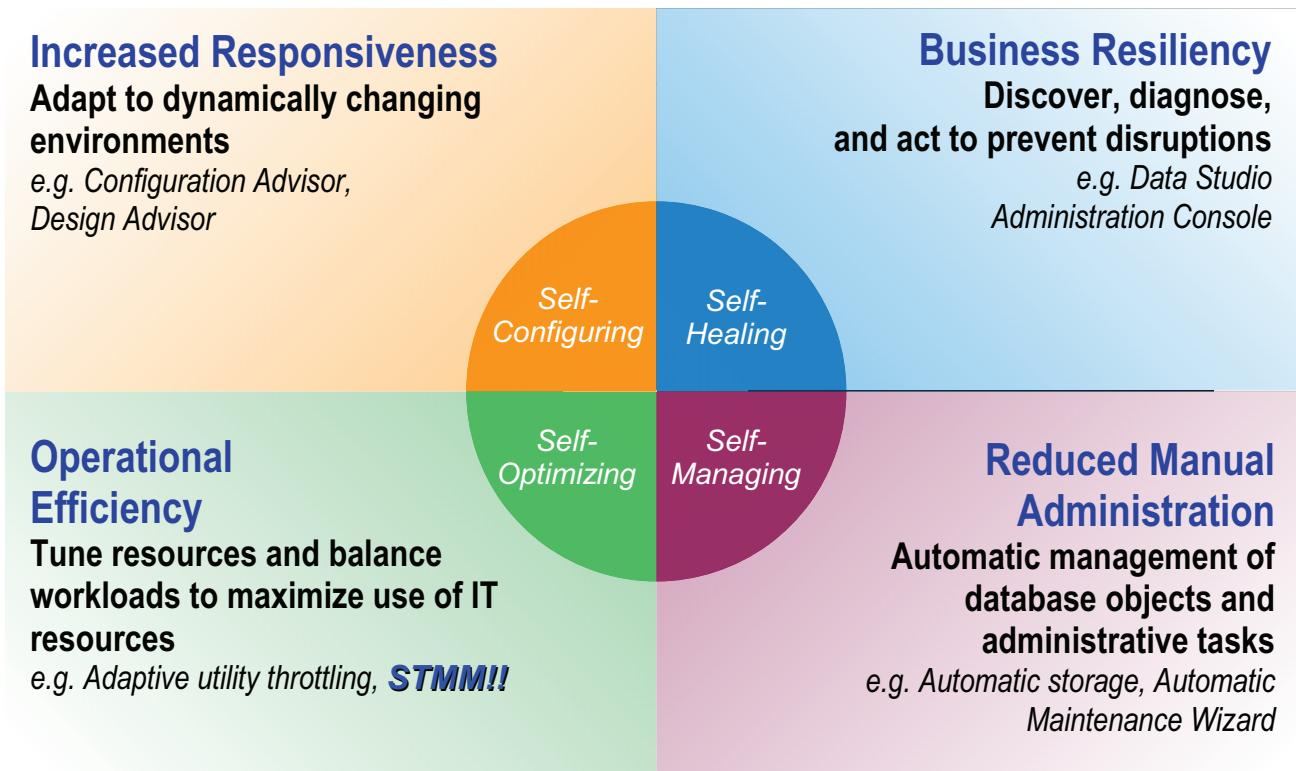
© 2011 IBM Corporation

# Why do we care about autonomies?

- **Time and cost savings**
    - Over 70% of IT budgets are consumed by labour costs
    - Ability to grow IT projects while limiting project support size
      - As data and other IT systems grows, this must all be contained with existing support personnel
      - Greater efficiency, productivity required
  - **Therefore:**
    - Very beneficial for everyone to leverage these features and tools in DB2



# Autonomic Computing Attributes and DB2 Features

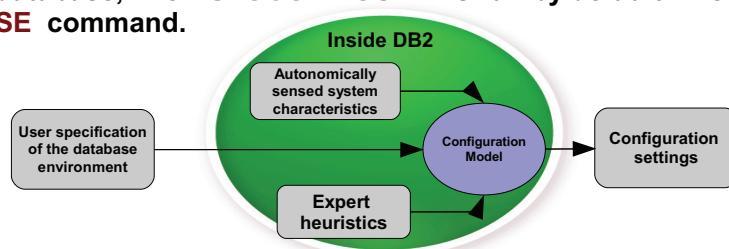


7

© 2011 IBM Corporation

## Self-Configuring Example – Configuration Advisor

- DB2 Configuration Advisor uses expert heuristics to tune performance and to balance memory requirements
  - Automatic environment characteristic detection
  - User description of basic operational questions
  - Combined with mathematical model of each configuration parameter, based on expert heuristics
- To use the Configuration Advisor, specify the **AUTOCONFIGURE** command for an existing database. The **AUTOCONFIGURE** is run by default when you issue the **CREATE DATABASE** command.



### Oracle Database Configuration Assistant

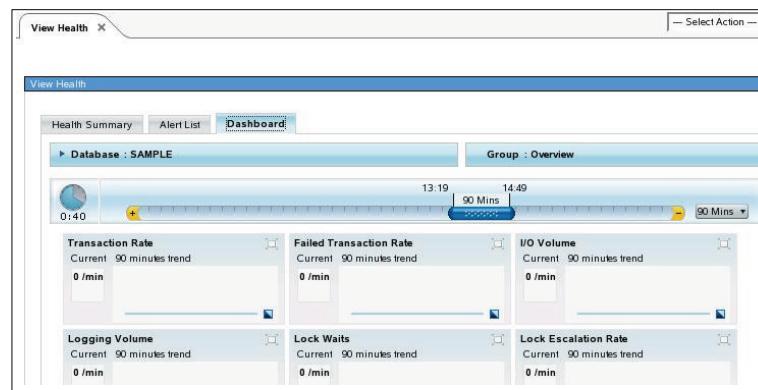
- has only a few database templates and won't help you to set the performance parameters and cache settings all at once, in one tool
- DBCA Steps:
  - 1 – create a parameter and password files.
  - 2- Use the parameter file to build an instance.
  - 3- Issue the CREATE DATABASE command.
  - 4- Run SQL scripts to generate the data dictionary.
  - 5- Run SQL scripts to generate the Enterprise Manager Database Control that database will require

## Self-Configuring Example – Design Advisor

- **Tool that suggests modifications to the database's physical design to improve performance**
- **Allows to display, edit, or save the new database object creation recommendation set**
- **Invoked with the db2advis command**
- **Implementation of the recommendations by creating:**
  - Indexes on the base tables
  - Materialized Query Tables (MQT)
  - Indexes on the Materialized Query Tables
  - Converting regular tables into Multi-Dimensional Clustering (MDC) tables
  - Re-partitioning (DPF) existing tables with a new set of partitioning key and/or in a new table space.

## Self-Healing Example – DSHM

- **Data Studio Health Monitor is a web application able to monitor your DB2 LUW or DB2 z/OS databases**
  - Free download
- <http://www.ibm.com/developerworks/downloads/im/data/>
  - Designed to be the successor of the deprecated Health Monitor
- **Provides a single portal for viewing the status of all your databases**
  - **Health Summary** of all databases monitored
  - **Alert List** showing all notifications for monitored databases
  - **Dashboard** showing different attributes of the system and databases



## Self-Managing Example – Automatic Storage

- **Simplifies storage management for databases and table spaces**
  - User specifies storage paths where DB2 will store the data
  - DB2 automatically manages container and space allocation as data is inserted
- **Benefits:**
  - User does not need to provide container definitions for table spaces
  - Growth of existing containers and addition of new ones completely managed by DB2
- **Enabled by default for all new databases since DB2 9**
  - Databases (and table spaces inside) upgraded from older versions can be converted to use Automatic Storage
    - Alter the database and then each table space after upgrading

[http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.db\\_obj.doc/doc/t0054953.html](http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.db_obj.doc/doc/t0054953.html) © 2011 IBM Corporation

## DB2 Automatic Storage

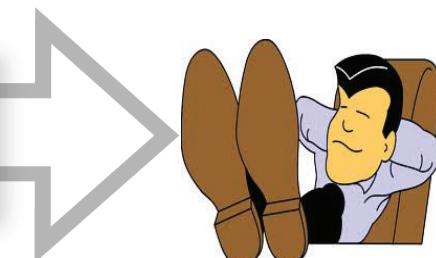
- Create a database with automatic storage ON

```
CREATE DATABASE <dbname> AUTOMATIC STORAGE  
YES ON /data/storagePath1, /data/storagePath2
```

DB2 will create the containers inside these paths

- Create a tablespace with automatic storage ON

```
CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE  
INITIALSIZE 500 K INCREASESIZE 100 K MAXSIZE 100 M
```



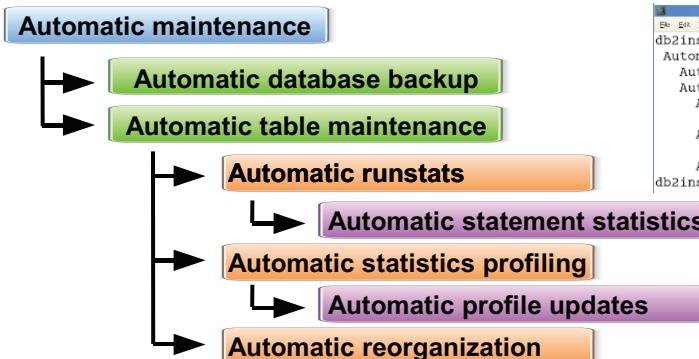
Autonomics

## Oracle's Automatic Storage (ASM)

- **Works as a “separate” instance**
  - One ASM instance to manage all databases in the same organization
- **Limited connections**
  - All connections must be via **SYS** and **SYSTEM** users
  - Operating system authentication only (CONNECT / AS SYSADM / SYSASM / SYSOPER)
- **Centralized point of control**
  - Loosing the ASM instance, you loose the automatic storage management for all databases in the organization

## Self-Managing Example – Automatic Object Maintenance

- **Database maintenance is essential for ensuring optimized performance and recoverability**
  - Therefore, setting up automatic maintenance removes burden from DBA to have to manually do these tasks
- **AUTO\_MAINT parameter is the master on/off switch**
  - Individual Child parameters can be set to ON/OFF and the settings are persisted in the database configuration file.
- **Enabled/disabled through database configuration parameter hierarchy**



```

db2inst1@ncis:~> db2 get db cfg for sample |grep AUTO |grep Automatic
Automatic maintenance
  Automatic database backup
  Automatic table maintenance
  Automatic runstats
    Automatic statement statistics
    Automatic statistics profiling
    Automatic profile updates
    Automatic reorganization
  (AUTO_MAINT) = ON
  (AUTO_DB_BACKUP) = OFF
  (AUTO_TBL_MAINT) = ON
  (AUTO_RUNSTATS) = ON
  (AUTO_STMT_STATS) = ON
  (AUTO_STATS_PROF) = OFF
  (AUTO_PROF_UPD) = OFF
  (AUTO_REORG) = OFF
  
```

## Self-Managing Example – Automatic Object Maintenance

- Each of these object maintenance types needs a policy that is set through:

–IBM Data Studio

- Right click on database and select “Configure Automatic Maintenance”

## **-Stored Procedures**

- SYSPROC.AUTOMAINT\_SET\_POLICY or SYSPROC.AUTOMAINT\_SET\_POLICYFILE
  - The policy for each maintenance type is in XML format
  - Samples available in your instance  
\$INSTANCEHOME/sqllib/samples/automaintcfg
  - More information at:  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051354.html>

# Oracle's Automatic Maintenance

- Not available in Oracle Express Edition

–Needs Oracle Enterprise Manager Diagnostic Pack

- #### ▪ JOB based

- You must manually schedule jobs

- Three automated database maintenance tasks:

### -Gathering Optimizer Statistics

- Oracle uses the “10% of row changes” algorithm while DB2’s automatic maintenance considers the table size, how often the table is accessed and when the last statistics have been collected

-Segment Advisor and SQL Tuning Advisor

- Tools must be engaged manually so that it can act in the database



## Self-Optimizing Example – STMM

- **Self Tuning Memory Manager (STMM) simplifies the task of memory configuration**
- **Optimizes the performance of your database by automatically adjusting the values of:**
  - Memory configuration parameters in instance configuration
    - *Total instance memory*
  - Memory configuration parameters in database configuration
    - *e.g. Sort heap, lock list, package cache, and total DB memory*
  - Size of *buffer pools*
- **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

## STMM Configuration

- **STMM is ON by default for all new databases since DB2 9**
  - Databases upgraded from older versions can turn STMM on
- **The database configuration parameter **SELF\_TUNING\_MEM** is the master switch for STMM**
  - Configuration parameters and buffer pool sizes should be set to **AUTOMATIC** to enable tuning

```
jasoncha@ncis:~> db2 get db cfg for magnum |grep AUTOMATIC
Size of database shared memory (4KB) (DATABASE_MEMORY) = AUTOMATIC(87040)
Max storage for lock list (4KB) (LOCKLIST) = AUTOMATIC(6200)
Percent. of lock lists per application (MAXLOCKS) = AUTOMATIC(60)
Package cache size (4KB) (PCKCACHESZ) = AUTOMATIC(1533)
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = AUTOMATIC(8892)
Sort list heap (4KB) (SORTHEAP) = AUTOMATIC(1778)
Database heap (4KB) (DBHEAP) = AUTOMATIC(2339)
SQL statement heap (4KB) (STMTHEAP) = AUTOMATIC(4096)
Default application heap (4KB) (APPLHEAPSZ) = AUTOMATIC(256)
Application Memory Size (4KB) (APPL_MEMORY) = AUTOMATIC(40000)
Statistics heap size (4KB) (STAT_HEAP_SZ) = AUTOMATIC(4384)
```

## STMM Configuration

- **If upgrading from older version of DB2 or previously turned off STMM:**
  - 1) Turn on the master switch:
    - `update db cfg for <db_name> using SELF_TUNING_MEM ON`
  - 2) Set each parameter that you wish to tune to AUTOMATIC
    - `update db cfg for <db_name> using <param> AUTOMATIC`
- **For buffer pools to be tuned by STMM:**
  - All newly created buffer pools default to AUTOMATIC
  - For migrated database's bufferpools
    - `alter buffer pool <bp_name> size AUTOMATIC`
- **To query whether or not STMM is tuning your DB:**
  - 1) Connect to the database
  - 2) Get Database Configuration with details
    - `get db cfg for <db_name> show detail`
  - 3) Check the value for SELF\_TUNING\_MEM
    - If set to **ON (ACTIVE)** then system is being tuned
    - If set to **OFF**, or **ON (INACTIVE)** then system is not being tuned

## STMM – More Information

- **STMM works with the Database Partitioning Feature (DPF)**
  - As this is a more complex environment, need to understand situation to determine whether to use STMM or not
  - General rule of thumb for DPF w/ STMM:
    - When all database partitions have similar memory requirements and are running on similar hardware, self-tuning memory can be enabled without any modifications
    - More information:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0023815.html>

- **All STMM affected memory parameters are fully dynamic**
  - Changes do not require an instance restart

## STMM Operating Modes

- STMM works in two modes depending on the configuration of the DATABASE\_MEMORY parameter
- DATABASE\_MEMORY = AUTOMATIC

(DATABASE\_MEMORY) = AUTOMATIC(90800)

- Total amount of memory used by a particular DB2 database can grow or shrink over time
- Takes from, and returns memory to, the OS as necessary

- DATABASE\_MEMORY = COMPUTED or <number>

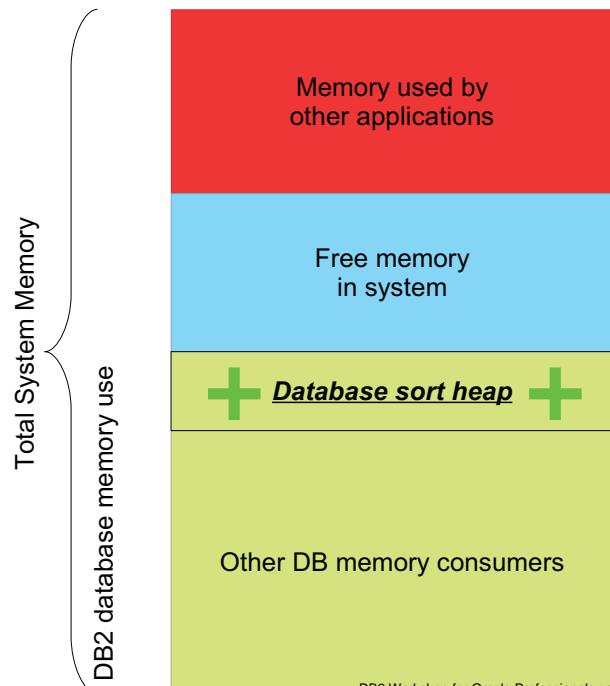
(DATABASE\_MEMORY) = COMPUTED(87040)  
 (DATABASE\_MEMORY) = 90000

- Memory tuning still occurs but total memory used by database is constant
- For one heap to grow another heap must shrink

## STMM Operating Modes

- DATABASE\_MEMORY = AUTOMATIC

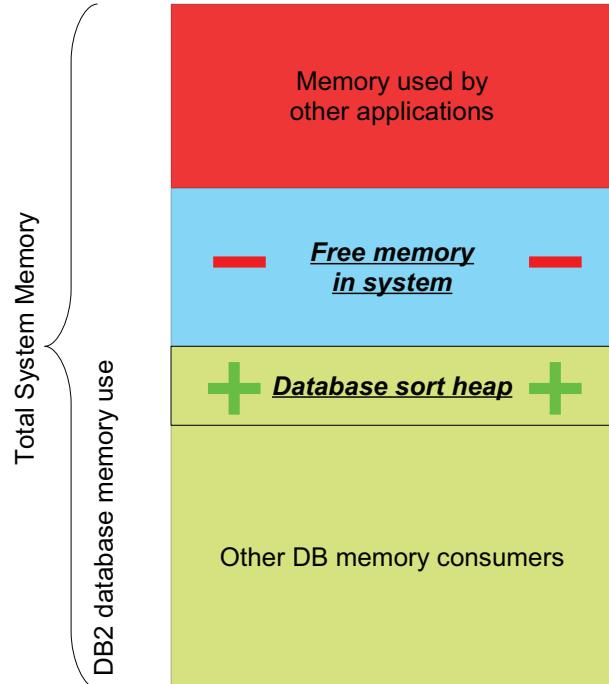
- 1) Scenario involves a change in workload that now requires more memory for sorts



## STMM Operating Modes

- **DATABASE\_MEMORY = AUTOMATIC**

- 1) Scenario involves a change in workload that now requires more memory for sorts
- 2) DB2 requests and gets more memory from the OS, therefore shrinking the free memory in the system



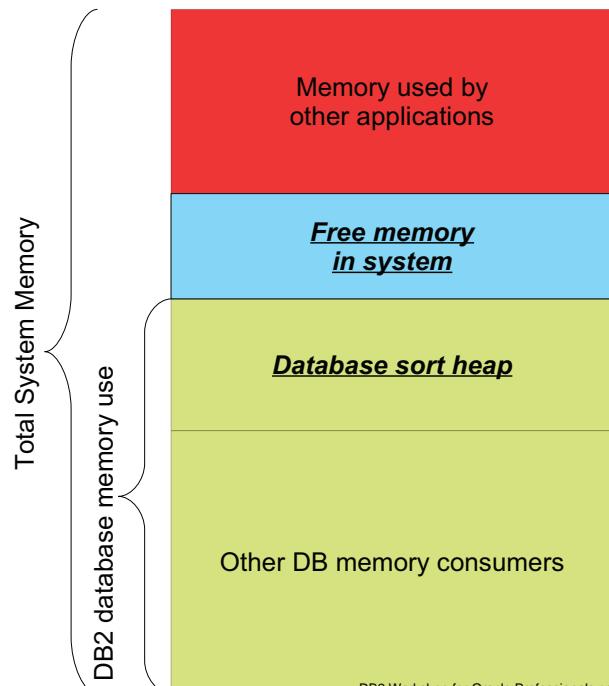
23

© 2011 IBM Corporation

## STMM Operating Modes

- **DATABASE\_MEMORY = AUTOMATIC**

- 1) Scenario involves a change in workload that now requires more memory for sorts
- 2) DB2 requests and gets more memory from the OS, therefore shrinking the free memory in the system
- 3) DB2 uses newly acquired memory in sort heap



24

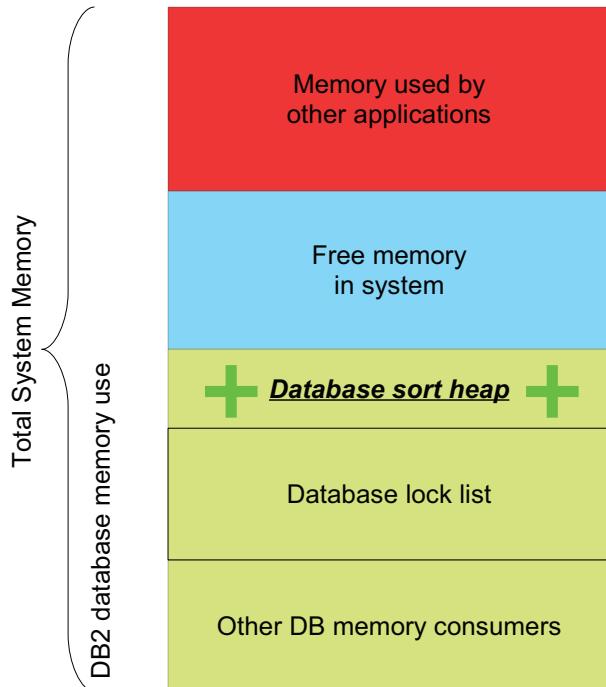
DB2 Workshop for Oracle Professionals page 47 of 207.

© 2011 IBM Corporation

## STMM Operating Modes

- **DATABASE\_MEMORY = COMPUTED or <number>**

1) Scenario involves a change in workload that now requires more memory for sorts



26

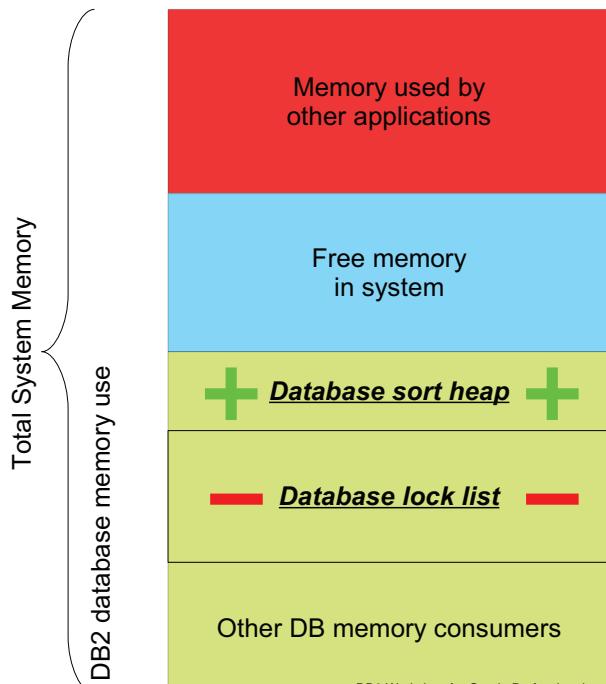
© 2011 IBM Corporation

## STMM Operating Modes

- **DATABASE\_MEMORY = COMPUTED or <number>**

1) Scenario involves a change in workload that now requires more memory for sorts

2) DB2 database is set at fixed memory usage, and thus cannot take memory from OS. Therefore identifies another memory consumer in the database that does not need its memory anymore (e.g. lock list)



26

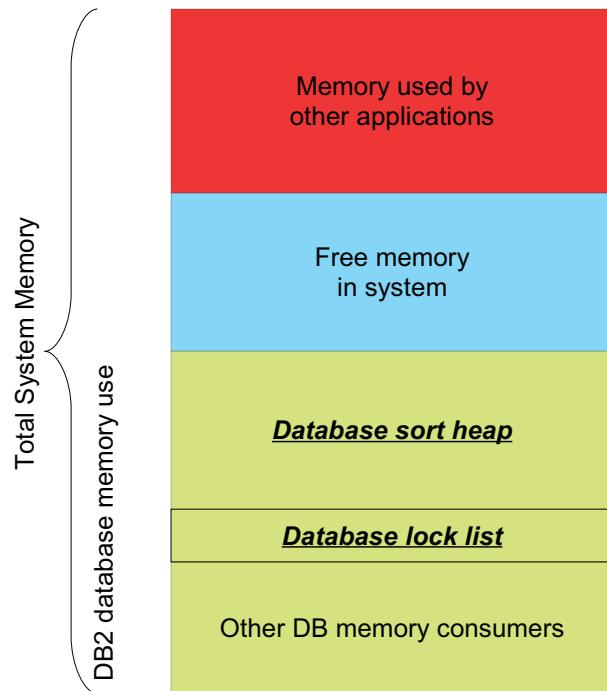
DB2 Workshop for Oracle Professionals page 48 of 207.

© 2011 IBM Corporation

## STMM Operating Modes

- **DATABASE\_MEMORY = COMPUTED or <number>**

- 1) Scenario involves a change in workload that now requires more memory for sorts
- 2) DB2 database is set at fixed memory usage, and thus cannot take memory from OS. Therefore identifies another memory consumer in the database that does not need its memory anymore (e.g. lock list)
- 3) The memory is transferred between the memory consumers. The overall memory usage for this DB2 database stays the same.

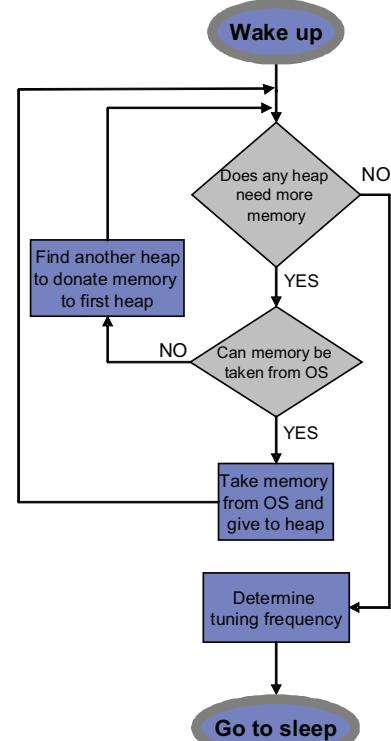


© 2011 IBM Corporation

27

## Memory Monitoring by STMM – Iterative Config. Changes

- Constantly monitors system to make use of or return any free memory to OS (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
- In each interval each heap can only grow by 50% or decrease by 20%



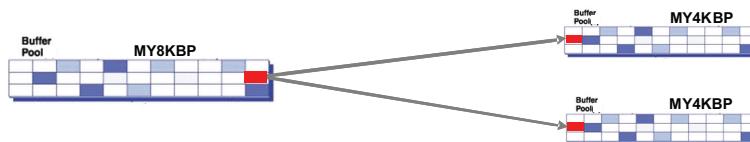
DB2 Workshop for Oracle Professionals page 49 of 207.

© 2011 IBM Corporation

28

## STMM and Buffer pools

- **Buffer pools are unlike other memory consumers in the database**
  - A buffer pool is a database object
  - There can be multiple buffer pools in a database
- **Trades memory between buffer pools based on relative need**
  - New metrics determine where memory is most needed such that total performance is improved
- **Zero, one or more buffer pools can be set to AUTOMATIC**
  - In newly created DB2 9 or later databases, all buffer pools size is defaulted to AUTOMATIC
- **Works with buffer pools of any page size**
  - Eg: To trade memory between 8K page size BP (MY8KBP) and 4K page size BP (MY4KBP), 1 page in MY8KBP will turn into 2 pages in MY4KBP



29

© 2011 IBM Corporation

## Scenarios where STMM shines

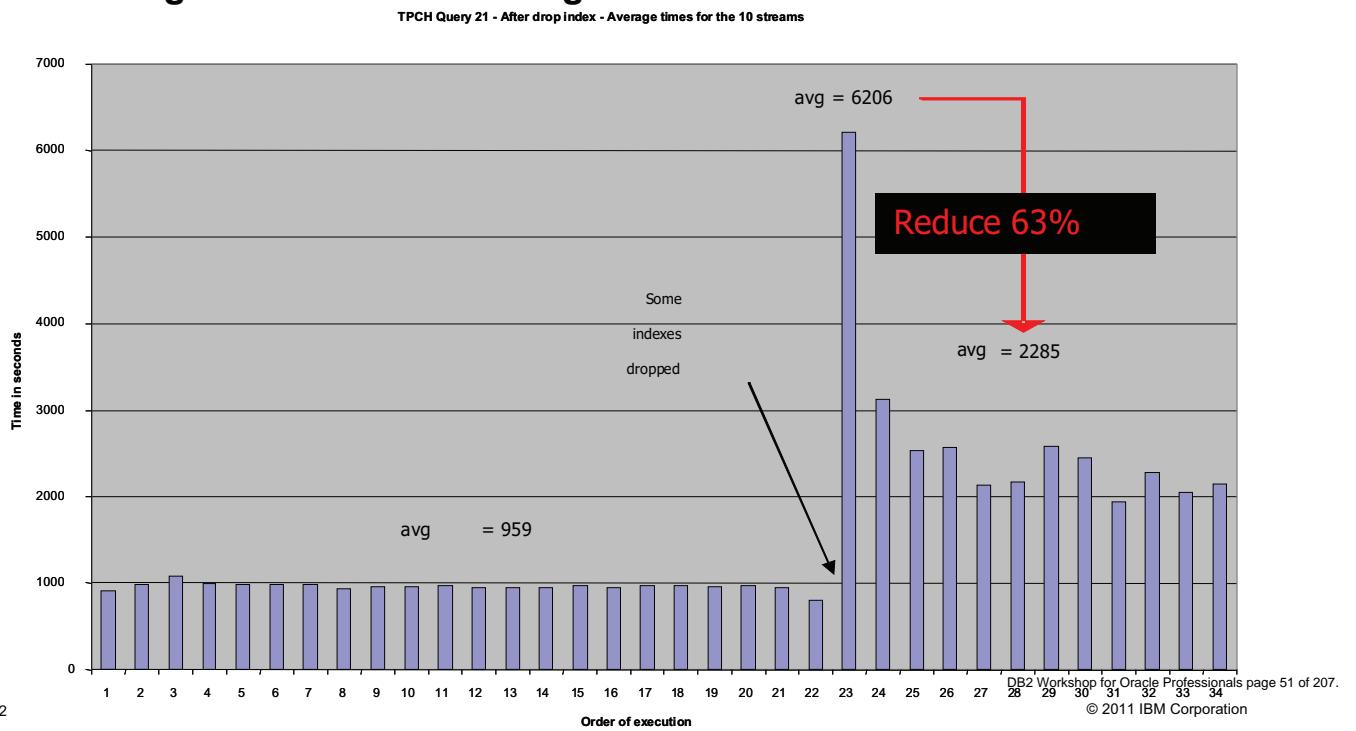
- **Bufferpool Tuning**
  - Difficult to tune memory when there are multiple buffer pools
    - More bufferpools causes high number of configurations for trading memory
  - STMM works with multiple buffer pools regardless of page size
- **Memory Varied Workload**
  - Some workloads have dramatically varied memory demands
  - **STMM constantly re-evaluates the memory requirements**
    - Can update the memory once every 3 minutes
  - Will optimize the memory usage based on the currently running workload
- **Unknown memory requirements**
  - New workload with unknown memory requirements
  - STMM works deep down in DB2 and is able to sense workload memory requirements
    - Requires absolutely no DBA interaction once turned on
  - **Will stop tuning automatically when it reaches optimal configuration**

## STMM in action – Dropping an important index

- 10 agents executing queries with “order by” clause
- At first, queries use indexes to avoid sorting
- After several iterations some of the indexes are dropped
  - Simulates DBA error in dropping vital indexes
- Lack of indexes forces sorts to be performed
- Dramatically increases the demand on the sort memory
- With only manual tuning, workload will likely get much slower
- Lets see what STMM does...

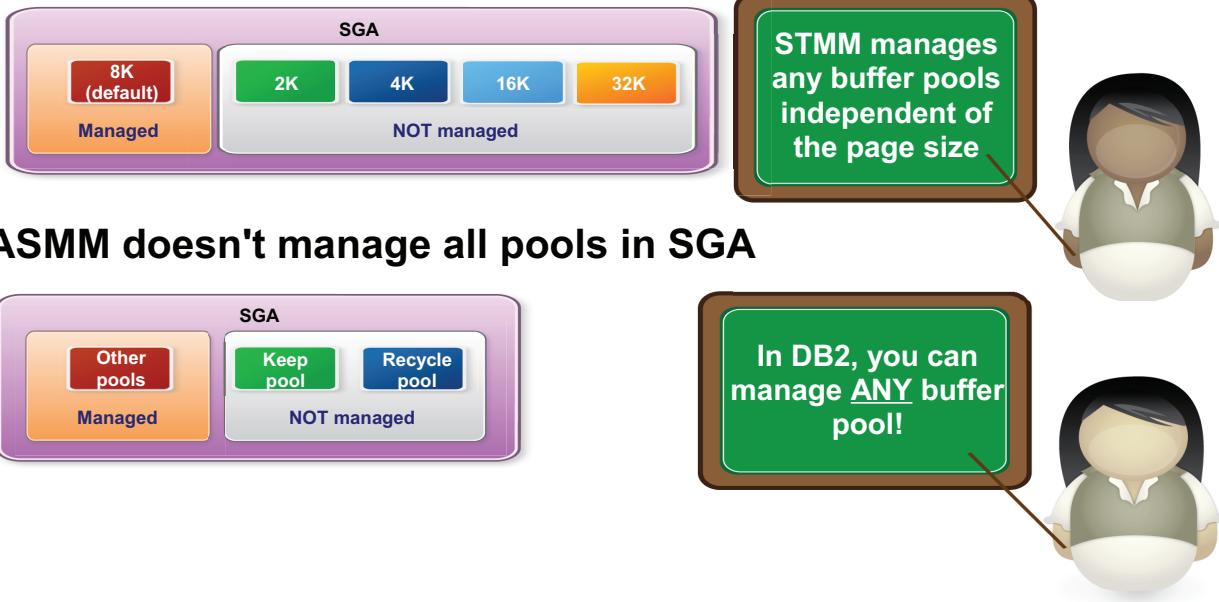
## STMM in action – Dropping an important index

- Query time is reduced by 63% in a few cycles through STMM reacting to environment changes



## Oracle's Automatic Shared Memory Management (ASMM)

- Activated by the **MEMORY\_TARGET** parameter
- SGA doesn't manage non-standard block size pools



## Summary – Practical Autonomics in DB2

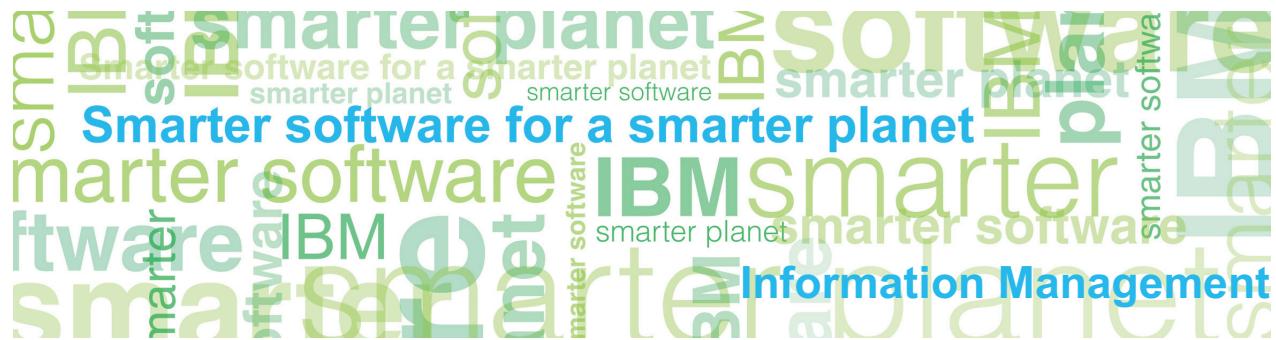
- DB2 autonomic features integrated in many forms to help DBAs
  - Just a small subset of examples:
    - Automatic Storage
    - Automatic Object Maintenance: Auto - Backup, Reorg, Runstats etc
    - STMM
    - ...
- Focus: **Manage your business, not your database**
  - All of these autonomic features work together to alleviate the need for constant monitoring and tuning by DBAs
  - DBAs are free to do more important work

## Questions?

Winter/Spring 2011

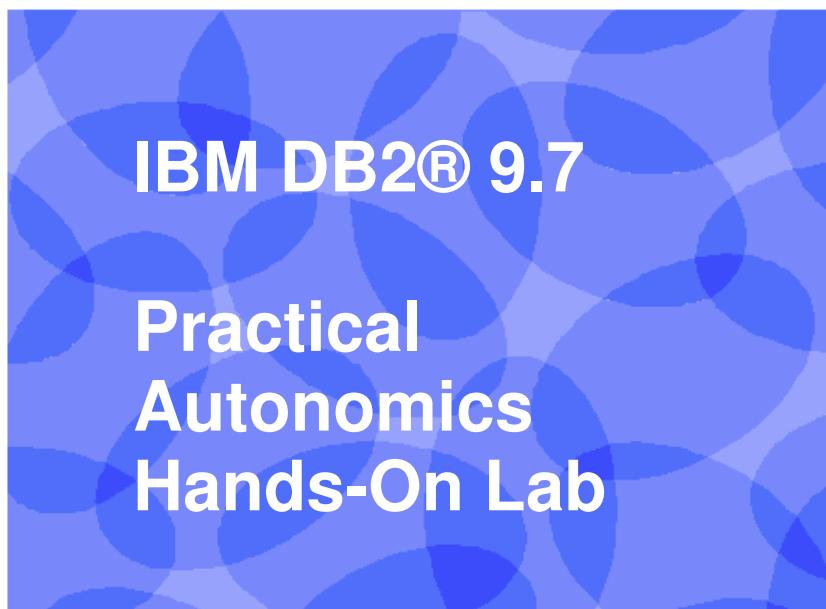
E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)

Subject: “DB2 Workshop for Oracle Professionals”



© 2011 IBM Corporation

**IBM** Information Management software



**Information Management Technology Ecosystem**

**IBM Canada Lab**

---

## Contents

<b>1. INTRODUCTION.....</b>	<b>3</b>
<b>2. BASIC SET UP AND START OF LAB.....</b>	<b>3</b>
2.1 ENVIRONMENT SETUP REQUIREMENTS .....	3
2.2 PREPARATION STEP .....	4
2.3 LAUNCHING DATA STUDIO.....	4
2.4 PREPARING THE DATABASE.....	6
<b>3. CONFIGURATION ADVISOR .....</b>	<b>7</b>
3.1 OPTIMIZE DATABASE USING CONFIGURATION ADVISOR.....	7
<b>4. AUTOMATIC STORAGE.....</b>	<b>9</b>
4.1 CHECK AUTOMATIC STORAGE SETTINGS.....	9
4.2 LOAD DATA INTO THE DATABASE .....	10
4.3 VIEW TABLESPACE AUTOMATIC RESIZE.....	11
4.4 ALTERING VPSLARGE TABLESPACE FOR AUTOMATIC STORAGE .....	12
<b>5. SET UP AUTOMATIC MAINTENANCE FOR THE LABDB DATABASE...14</b>	
5.1 CONNECT TO THE LABDB DATABASE .....	15
5.2 SELECTING AUTOMATIC MAINTENANCE OPTIONS.....	16
5.3 SELECTING THE ONLINE MAINTENANCE WINDOW.....	18
5.4 SELECTING THE OFFLINE MAINTENANCE WINDOW .....	19
5.5 PERFORM MAINTENANCE SELECTION AGAINST LABDB.....	20
5.6 ENABLING UTILITY THROTTLING.....	21
<b>6. DESIGN ADVISOR .....</b>	<b>21</b>
6.1 CREATE THE EXPLAIN TABLES .....	22
6.2 DESIGN ADVISOR RECOMMENDATIONS .....	22
6.3 EXECUTING THE QUERY AND SEEING BUFFERPOOL MEMORY CHANGES.....	26
<b>7. SELF-TUNING MEMORY MANAGER (STMM) .....</b>	<b>28</b>
7.1 CONNECT TO THE LABDB DATABASE AND EXAMINE THE DATABASE CONFIGURATION PARAMETERS .....	28
<b>8. STMM IN ACTION – VPS DEMO .....</b>	<b>30</b>
8.1 PREPARE TO EXECUTE THE VPS DEMO .....	30
8.2 VIEW THE VPS ENVIRONMENT IN A WEB BROWSER .....	31
8.3 START THE OLTP TRANSACTION WORKLOAD.....	32
8.4 START THE STMM SCENARIO .....	34
8.5 STOP THE STMM SCENARIO.....	35
8.6 CLEAN UP.....	36

---

## 1. Introduction

Today's business challenges require that databases be optimized for the best possible performance. The database engine must be highly tunable and easily configured for the best performance regardless of the skill level of the DBA. To face these challenges, DB2 comes with a rich set of autonomic features for tuning the configuration parameters adaptively and accurately.

In this lab, you will explore some of the autonomic computing capabilities of DB2 that helps reduce skill requirements and still ensure reliability and availability of your mission critical environment. After completion of this lab you will be able to use the following tools:

- Configuration Advisor
- Automatic storage
- Utility throttling
- Automatic Maintenance
- Design Advisor
- Self-Tuning Memory Manager (STMM)

The exercises in the lab were designed to give you a good idea of the features available with DB2 autonomics. Note this lab will not necessarily cover the approach you would take to initially configure a data server; however, it will give you a good idea of some necessary steps required to use each of these useful autonomic features.

---

## 2. Basic Set up and Start of Lab

The lab consists of a series of tasks, each task highlighting one or more concepts of the DB2 Practical Autonomics described in the introduction.

Some of the commands and queries required to perform certain parts of certain tasks are quite long. In order to avoid having to type these longer commands, we have provided you with scripts for the tasks. These are located in the `/home/db2inst1/Documents/LabScripts/Autonomics` directory.

### 2.1 Environment Setup Requirements

To complete this lab you will need the following:

- DB2 9.7 Bootcamp VMware® image
- VMware Player 2.x or VMware Workstation 5.x or later

For help on how to obtain these components please follow the instructions specified in **VMware Basics and Introduction** from module 1.

## 2.2 Preparation Step

- Start the VMware image by clicking the  Power On button in VMware.

- At the login prompt, login with the following credentials:

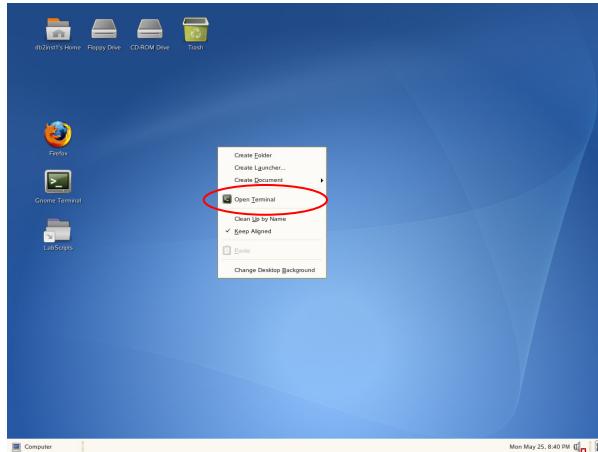
- ▶ Username: **db2inst1**
- ▶ Password: **password**

- Read and accept the license agreement.

- Start the graphical interface by issuing the following command at the prompt:

```
db2inst1@db2rules:~> startx
```

- Open a terminal window as follows by right-clicking on the **Desktop** and choosing the **Open Terminal** item.

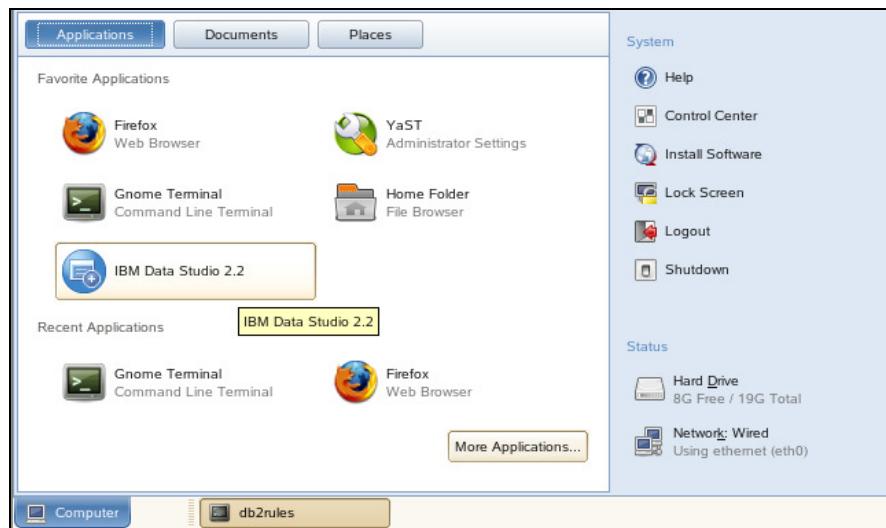


Start the Database Manager by entering the following command:

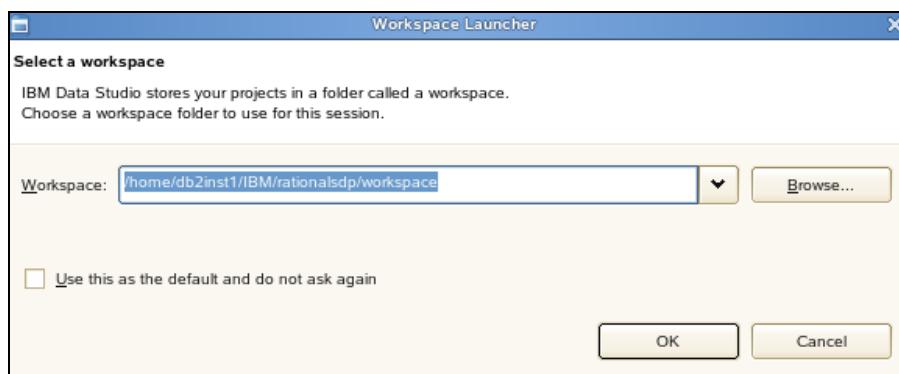
```
db2inst1@db2rules:~> db2start
```

## 2.3 Launching Data Studio

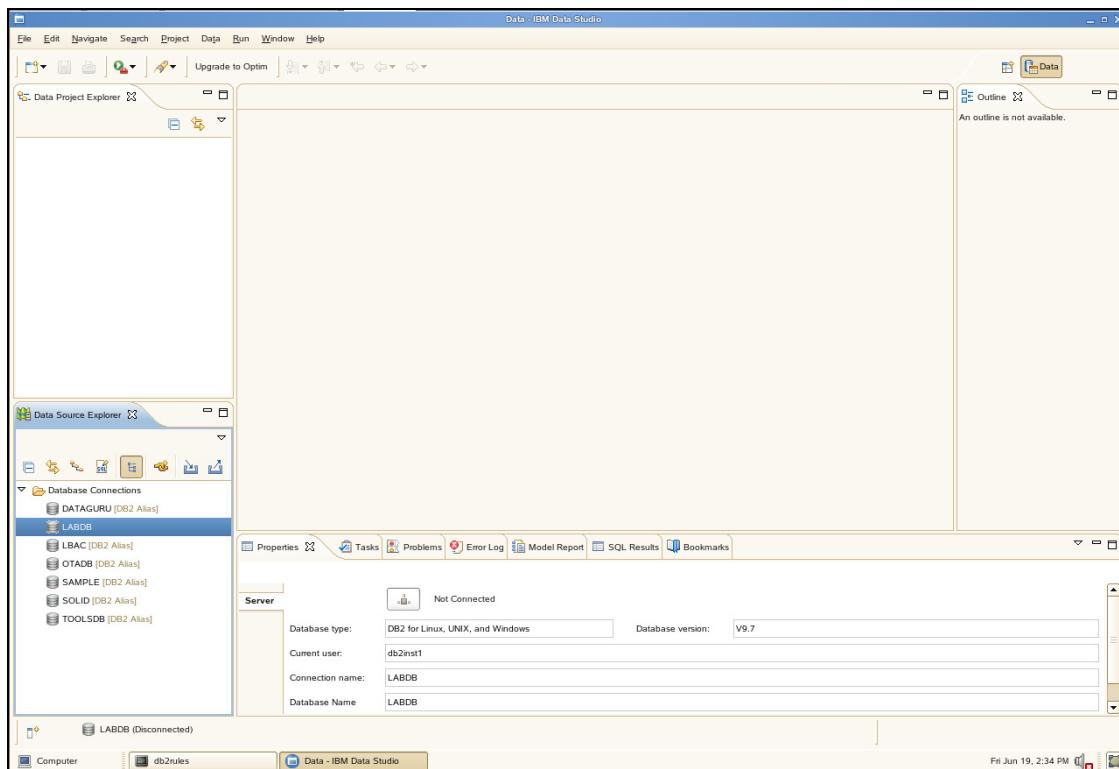
- Click on the **Computer** button in the bottom left corner of the screen, and select **IBM Data Studio 2.2**.



2. In the **Select a workspace** dialog, accept the default path and check the **Use this as the default and do not ask again** checkbox. Click **OK**.



3. Data Studio will now start in the **Data** perspective as shown below.



## 2.4 Preparing the Database

Throughout this lab we will be working with a database named LABDB that contains several tablespaces and is managed by automatic storage. We will use this database to describe all of the autonomic components described above by executing various queries on this database throughout the lab.

1. Open up a new terminal window or use the minimized terminal window (from the beginning of the lab).
2. Change your current directory to the directory containing **all** the scripts for this lab:

```
db2inst1@db2rules:~> cd /home/db2inst1/Documents/LabScripts/Autonomics
```

3. Use gedit to view the contents of the create\_db\_labdb.sql script by executing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/AUTONOMICS> gedit
create_db_labdb.sql
```

The following should display:

Enabling the database for  
**Automatic Storage.**  
This is enabled by default unless it is explicitly turned off.

```
DROP DB labdb;
CREATE DB labdb ON /db2fs/db2inst1/sp1, /db2fs/db2inst1/sp2 USING
CODESET UTF-8 TERRITORY US;
CONNECT TO labdb;
CREATE SCHEMA vps;
CREATE SCHEMA dupvps;
CREATE BUFFERPOOL VPSBUFF IMMEDIATE AUTOMATIC PAGESIZE 16k;
CREATE TABLESPACE VPSSLARGE PAGESIZE 16k MANAGED BY DATABASE USING (FILE
`/db2fs/db2inst1/tempspace' 300) BUFFERPOOL VPSBUFF AUTORESIZE YES;
CREATE TABLESPACE vpsspc1 PAGESIZE 16k MANAGED BY AUTOMATIC STORAGE
AUTORESIZE YES BUFFERPOOL VPSBUFF;
CREATE TABLESPACE vpsspc2 PAGESIZE 16k MANAGED BY AUTOMATIC STORAGE
BUFFERPOOL VPSBUFF;
CREATE TABLESPACE vpsspc3 PAGESIZE 16k MANAGED BY AUTOMATIC STORAGE
BUFFERPOOL VPSBUFF;
UPDATE DBM CFG USING SHEAPTHRES 0;
```

Notice that the database manager SHEAPTHRES parameter is set to zero. This is to enable sort memory tuning (SORTHEAP) for optimal configuration parameter settings when using the configuration advisor.

4. Execute the script by issuing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/AUTONOMICS> db2 -tvf
create_db_labdb.sql
```

All commands should execute successfully, creating your database with various tablespaces.

## 3. Configuration Advisor

DB2 configuration parameters play an important role in performance as they affect the operating characteristics of a database or database manager. The Configuration Advisor makes recommendations on the initial settings for configuration parameters that can be easily adopted by inexperienced administrators or fine-tuned by more experienced administrators. We will see in this section how the Configuration Advisor (invoked via the **AUTOCONFIGURE** command) can generate values for various performance critical parameters.

### 3.1 Optimize Database Using Configuration Advisor

A good place to start with this database is to run the **AUTOCONFIGURE** command on a database which will recommend enablement of the Self Tuning Memory Manager (STMM). We will be covering STMM later in this lab.

Although when a database is created the Configuration Advisor runs automatically, explicitly invoke the Configuration Advisor using the command below.

1. Execute the following two commands in a terminal:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 connect to labdb
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 "AUTOCONFIGURE APPLY
DB AND DBM"
```

More configuration parameters are available for the AUTOCONFIGURE command; however, by not specifying them, the defaults are accepted. The following should output:

Former and Applied Values for Database Manager Configuration			
Description	Parameter	Former Value	Applied Value
Application support layer heap size (4KB)	(ASLHEAPSZ)	= 15	15
No. of int. communication buffers(4KB) (FCM_NUM_BUFFERS)	= AUTOMATIC	AUTOMATIC	AUTOMATIC
Enable intra-partition parallelism	(INTRA_PARALLEL)	= NO	NO
<b>Maximum query degree of parallelism</b> ( <b>MAX_QUERYDEGREE</b> )	= ANY		1
Agent pool size	(NUM_POOLAGENTS)	= AUTOMATIC(100)	AUTOMATIC(100)
Initial number of agents in pool	(NUM_INITAGENTS)	= 0	0
Max requester I/O block size (bytes)	(RQRIOBLK)	= 32767	32767
Sort heap threshold (4KB)	(SHEAPTHRES)	= 0	0
Former and Applied Values for Database Configuration			
Description	Parameter	Former Value	Applied Value
Default application heap (4KB)	(APPLHEAPSZ)	= 256	256
Catalog cache size (4KB)	(CATALOGCACHE_SZ)	= (MAXAPPLS*4)	260
<b>Changed pages threshold</b>	(CHNGPGS_THRESH)	= 60	80
<b>Database heap (4KB)</b>	(DBHEAP)	= 1200	2288
Degree of parallelism	(DFT_DEGREE)	= 1	1
...	...	...	...

**Note:** These results may vary since the system defaults are being used by the AUTOCONFIGURE statement. Also, the above command can be executed anytime if you feel your system performance is not optimized to its maximum capabilities.

2. Restart the instance for the changes to take affect by issuing the following commands in the terminal:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 FORCE APPLICATION ALL
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2stop
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2start
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 connect to labdb
```

We issued the first command to force/release all applications (users) off the database for proper restart of the instance.

---

## 4. Automatic Storage

Automatic storage simplifies storage management for multiple table spaces. When you create a database, you specify the storage paths where the database manager will place your table space data. Then, the database manager will manage the container and space allocation for the table spaces as you create and populate them. Hence, DB2 completely manages the growth of containers.

### 4.1 Check Automatic Storage Settings

Databases that are enabled for automatic storage have a set of one or more storage paths associated with them at database creation time (`/home/db2inst1/sp1` and `/home/db2inst1/sp2` in our case). A table space can be defined as "managed by automatic storage" and its containers assigned and allocated by DB2 based on those storage paths.

1. Use the terminal window in the `/home/db2inst1/Documents/LabScripts/Autonomics` directory to get snapshots of all of the tablespaces for LABDB database before loading any data. This will also help confirm automatic storage on the tablespaces. Issue the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 get snapshot  
for tablespaces on labdb > noautotbs_before.snap
```

2. Use gedit to view the contents of the `noautotbs_after.snap` output by executing the following command

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> gedit  
noautotbs_before.snap
```

After opening the file we can analyze the file to see that automatic storage is not yet turned on for VPSLARGE tablespace. This is because we specified that the VPSLARGE tablespace to be managed by DMS (database managed storage) during creation time.

Also pay close attention to the Initial/Current tablespace size and time of last successful resize parameters, since we will analyze how these will change as data is loaded into the database.

Tablespace name	= VPSLARGE
Tablespace ID	= 3
Tablespace Type	= Database managed space
Tablespace Content Type	= All permanent data. Large table space.
Tablespace Page size (bytes)	= 16384
Tablespace Extent size (pages)	= 32
Automatic Prefetch size enabled	= Yes
Buffer pool ID currently in use	= 2
Buffer pool ID next startup	= 2
<b>Using automatic storage</b>	= No
<b>Auto-resize enabled</b>	= Yes
File system caching	= No
Tablespace State	= 0x'00000000'
Detailed explanation:	
Normal	
Tablespace Prefetch size (pages)	= 32
<b>Total number of pages</b>	= 300
<b>Number of usable pages</b>	= 256
<b>Number of used pages</b>	= 96
<b>Number of pending free pages</b>	= 0
<b>Number of free pages</b>	= 160
High water mark (pages)	= 96
Current tablespace size (bytes)	= 4915200
Maximum tablespace size (bytes)	= NONE
Increase size (bytes)	= AUTOMATIC
<b>Time of last successful resize</b>	=
Last resize attempt failed	= No
Rebalancer Mode	= No Rebalancing
Minimum Recovery Time	=
Number of quiescers	= 0
Number of containers	= 1

Container Name = /home/db2inst1/Documents/LabScripts/Autonomics/tempspace

In this snapshot output we can also identify the storage path of the container used for the VPSLARGE DMS tablespace. Also note the current number of pages available and the total number of pages that are free.

Notice that the VPSLARGE tablespace can automatically resize to accommodate the data loaded (specified by the AUTORESIZE YES clause during creation), however, it is not yet set for automatic storage.

## 4.2 Load Data into the Database

Now let's see how DB2 will enlarge a DMS tablespace with the AUTORESIZE YES clause in order to accommodate data loaded into the database. We will now load a defined amount of data into our LABDB database and check if tablespace VPSLARGE automatically resizes if the data is larger than what the tablespace can currently hold.

1. Use the terminal window in the /home/db2inst1/Documents/LabScripts/Autonomics directory to initiate the PopulateDB.sh script by issuing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> ./PopulateDB.sh
```

2. You will be prompted for some information regarding the size of the relational data that you would like to load. For this particular section in the lab we will input 100MB as the amount of data that we would like to insert into our database.

Size of Relational Data (MBs) :  
100

After some time, the following should display, indicating that everything has been processed successfully:

```
- Summary -
OLTP Size: 100
XML Size: 0
User: db2inst1
Pass: password
Port: 50001

Creating data tables...[DONE]
Starting OLTP Data Creation...[DONE]
Loading data into tables...[DONE]
Performing integrity check...[DONE]
Creating indexes...[DONE]
Creating stored procedures...[DONE]
LABDB Database creation completed.
Removing flatfiles...[DONE]
LABDB Database population completed.
All DB2 commands have been logged to VPS_Setup.log
```

All of the above steps should complete successfully, with the [DONE] tag beside them.

### 4.3 View Tablespace Automatic Resize

Initially, the tablespace could only hold approximately 4MB (256 pages as seen in first snapshot, each 16kb in size) of storage. Now that we loaded 100MB of data we can reissue the GET SNAPSHOT command to view the automatic resize.

1. Use the terminal window in the /home/db2inst1/Documents/LabScripts/Autonomics directory to get snapshots of all of the tablespaces for the LABDB database:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 get snapshot
for tablespaces on labdb > noautotbs_after.snap
```

2. Use gedit to view the contents of the noautotbs\_after.snap output by executing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> gedit noautotbs_after.snap
```

Within this file you need to confirm that the VPSLARGE tablespace has been automatically resized. This means that you need to ensure that the current tablespace size is larger than the initial size. As well you need to check the time of last successful resize.

The contents should look something like the following:

Tablespace name	= VPSLARGE
Tablespace ID	= 3
Tablespace Type	= Database managed space
Tablespace Content Type	= All permanent data. Large table space.
Tablespace Page size (bytes)	= 16384
Tablespace Extent size (pages)	= 32
Automatic Prefetch size enabled	= Yes
Buffer pool ID currently in use	= 2
Buffer pool ID next startup	= 2
Using automatic storage	= No
Auto-resize enabled	= Yes
File system caching	= No
Tablespace State	= 0x'00000000'
Detailed explanation:	
Normal	
Tablespace Prefetch size (pages)	= 32
<b>Total number of pages</b>	<b>= 31008</b>
<b>Number of usable pages</b>	<b>= 30976</b>
<b>Number of used pages</b>	<b>= 29792</b>
<b>Number of pending free pages</b>	<b>= 0</b>
<b>Number of free pages</b>	<b>= 1184</b>
<b>High water mark (pages)</b>	<b>= 29792</b>
<b>Current tablespace size (bytes)</b>	<b>= 508035072</b>
<b>Maximum tablespace size (bytes)</b>	<b>= NONE</b>
<b>Increase size (bytes)</b>	<b>= AUTOMATIC</b>
<b>Time of last successful resize</b>	<b>= 02/01/2009 16:31:07.428080</b>
Last resize attempt failed	= No
Rebalancer Mode	= No Rebalancing
Minimum Recovery Time	=
Number of quiescers	= 0
Number of containers	= 1

The initial number of pages defined for the tablespace was 300 pages. After loading the data, the total number of pages is 31008! This shows how DB2 expanded the tablespace to accommodate this data. Please continue to further examine and compare the output before loading the data and after.

## 4.4 Altering VPSLARGE Tablespace for Automatic Storage

An automatic storage database contains containers and space management characteristics that are completely determined by DB2. At the most basic level, databases that are enabled for automatic storage have a set of one or more storage paths associated with them (in our case /home/db2inst1/sp1 and /home/db2inst1/sp2 – as defined at creation time). A table space can be defined as "managed by automatic storage" and its containers assigned and allocated by DB2 based on those storage paths.

Let's change the VPSLARGE DMS tablespace to one that is managed by automatic storage. Use gedit to view the chg\_to\_auto.sql script in the /home/db2inst1/Documents/LabScripts/Autonomics directory which contains the necessary commands to perform this action.

```
ALTER TABLESPACE VPSLARGE MANAGED BY AUTOMATIC STORAGE;
COMMIT;
ALTER TABLESPACE VPSLARGE REBALANCE;
```

The REBALANCE clause will shift all of the data onto the two defined automatic storage paths for the database and then drop the DMS container used for the tablespace.

**Note:** If we do not execute the REBALANCE statement, the database manager takes over the DMS container and uses the two paths for any new additional data. Therefore, the DMS container becomes unusable to the user and cannot be altered or dropped.

1. Execute this script by issuing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 -tvf chg_to_auto.sql
```

2. Take a snapshot of all the tablespaces to see how the VPSLARGE tablespace has been changed to take advantage of Automatic Storage:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 get snapshot
for tablespaces on labdb > autotbs.snap
```

3. Use gedit to view the contents of the autotbs.snap.

Tablespace name	= VPSLARGE
Tablespace ID	= 3
...	...
Buffer pool ID currently in use	= 2
Buffer pool ID next startup	= 2
<b>Using automatic storage</b>	<b>= Yes</b>
<b>Auto-resize enabled</b>	<b>= Yes</b>
File system caching	= No
...	...
Total number of pages	= 30976
Number of usable pages	= 30912
Number of used pages	= 30880
Number of pending free pages	= 0
Number of free pages	= 32
High water mark (pages)	= 30880
Initial tablespace size (bytes)	= 0
Current tablespace size (bytes)	= 507510784
Maximum tablespace size (bytes)	= NONE
Increase size (bytes)	= AUTOMATIC
Time of last successful resize	= 02/01/2009 16:31:07.428080
Last resize attempt failed	= No
Rebalancer Mode	= No Rebalancing
Storage paths have been dropped	= No
Minimum Recovery Time	=
Number of quiescers	= 0
Number of containers	= 2

Notice that the VPSLARGE tablespace can still automatically resize and is now managed by automatic storage

The output above displays that the VPSLARGE tablespace is now enabled for automatic storage. Also note that the two containers used for the tablespace are on the storage path defined for the automatic storage database.

**Note:** If the DMS container is still present, take another snapshot in several minutes after the REBALANCE of the containers has finished. You will notice that the following section of the snapshot will disappear once the REBALANCE has been completed:

Rebalancer Mode	= Reverse
Start Time	= 02/01/2009 16:36:34.000000
Restart Time	= 02/01/2009 16:36:34.000000
Number of extents processed	= 489
Number of extents remaining	= 476
Last extent moved	= 475
Current priority	= 0

The REBALANCE runs asynchronously in the background and does not affect the availability of data. Therefore, access is not restricted to the table during rebalancing.

## 5. Set Up Automatic Maintenance for the LABDB Database

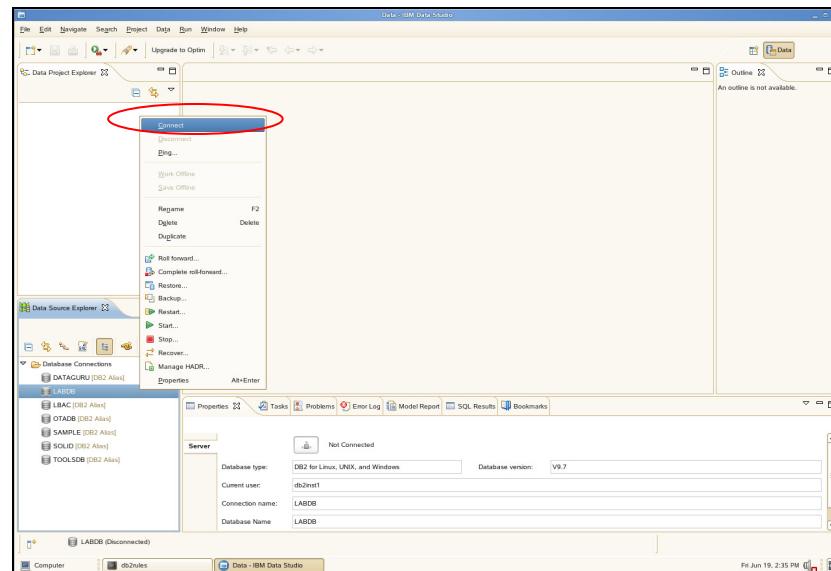
DB2 provides automatic maintenance capabilities for performing database backups, keeping statistics current and reorganizing tables and indexes as necessary. Performing maintenance activities on your databases is essential to ensure that it is optimized for performance and recoverability. Enablement of the automatic maintenance features is controlled using the automatic maintenance database configuration parameters. These are a hierarchical set of switches to allow for simplicity and flexibility in managing the enablement of these features.

Automatic maintenance is enabled by default however, only the automatic RUNSTATS feature is ON by default. The parameters AUTO\_STATS\_PROF and AUTO\_PROF\_UPD are not turned on automatically but could be set if automatic statistics profiling is desired. Automatic statistics profiling goes one step further by determining when and how to collect statistics. A statistics profile is generated automatically for use by the automatic statistics collection process to schedule RUNSTATS. An internal algorithm is used to compare newly collected statistics with a saved set of statistics, and RUNSTATS with sampling is issued based on certain conditions being triggered.

Configure Automatic Maintenance for other maintenance activities like BACKUP, REORG, STATS\_PROF and PROF\_UPD by following the sections below.

## 5.1 Connect to the LABDB Database

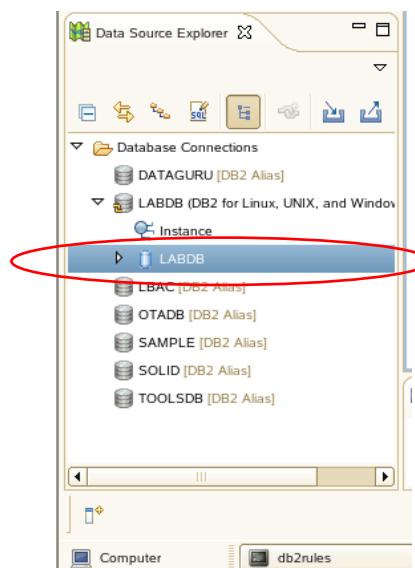
1. Connect to the LABDB database by navigating to the **Data Source Explorer** on the bottom left hand side of Data Studio. Navigate under **Database Connections** to find the LABDB database. Right-click on **LABDB** and click **Connect**.



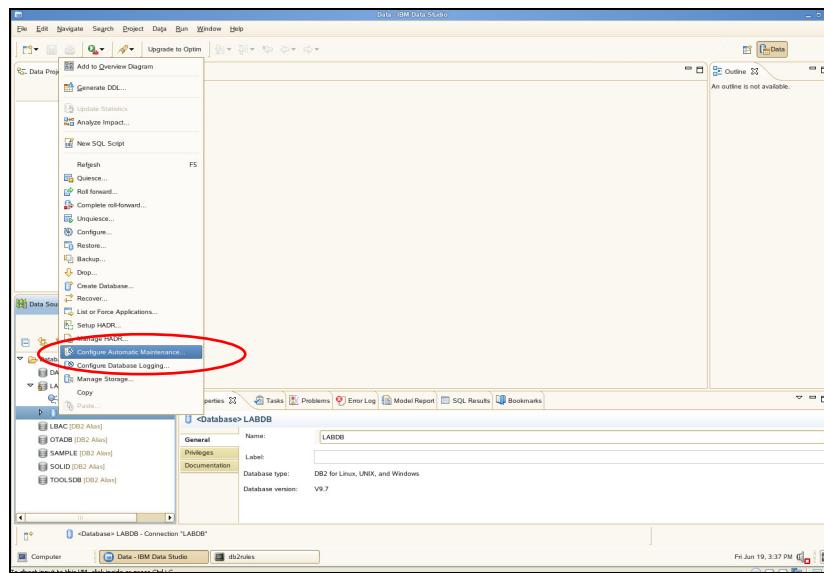
In the Properties for LABDB window, specify following credentials:

- ▶ Username: **db2inst1**
- ▶ Password: **password**

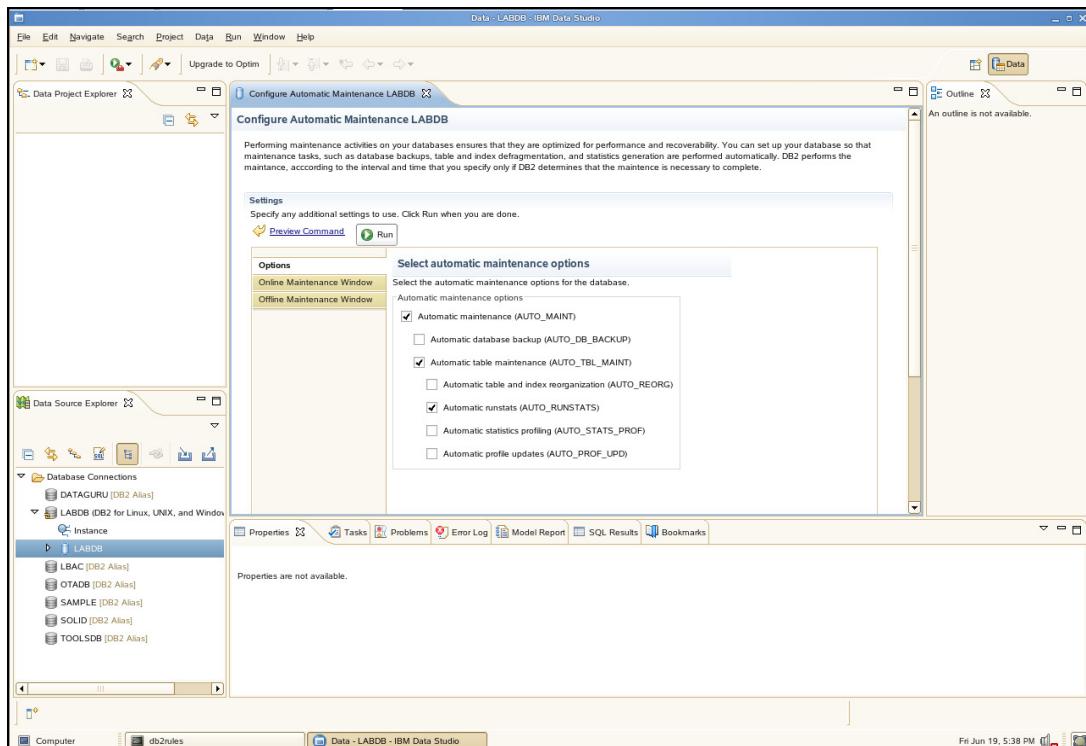
2. After the connection has successfully completed, expand the **LABDB** artifact to find the database artifact with the same name.



3. Launch DB2 Automatic Maintenance by right-clicking on this artifact and click on **Configure Automatic Maintenance...**



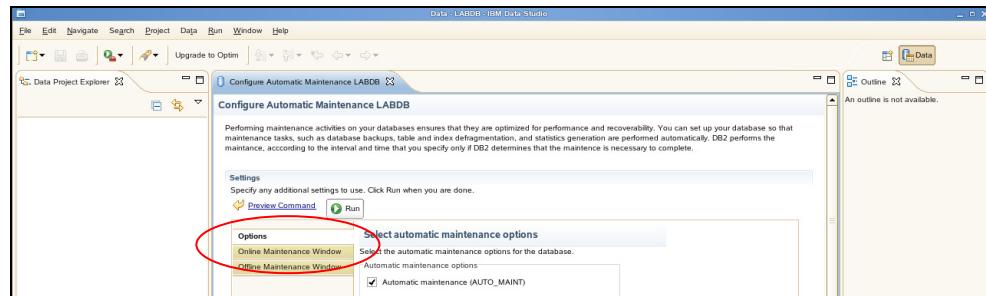
4. The Configure Automatic Maintenance LABDB tab will appear.



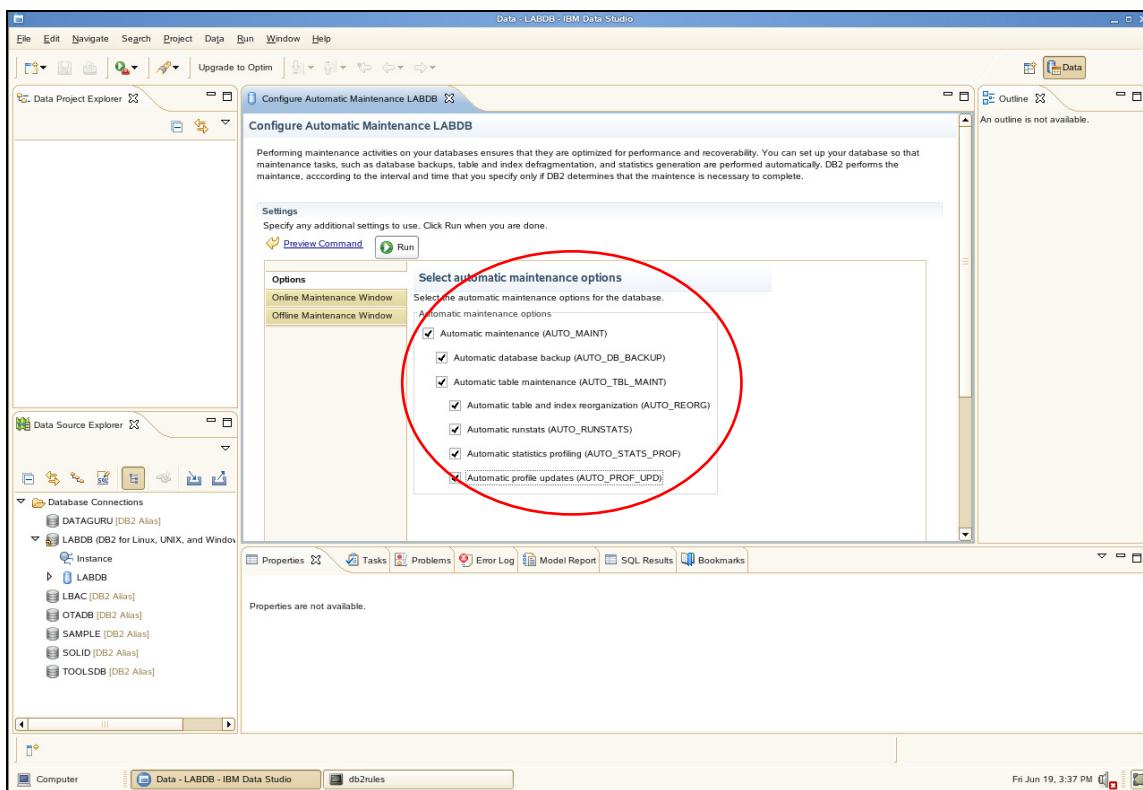
## 5.2 Selecting Automatic Maintenance Options

Configure Automatic Maintenance by following the below steps:

1. Read the Introduction section at the top of the tab to learn about DB2 automatic maintenance. If not already selected, click on the **Options** menu item to receive a list of all maintenance options that could be activated.



2. Select the maintenance activities that you wish to automate. In this case, select all activities by clicking the radio box beside the name (AUTO\_MAINT, AUTO\_DB\_BACKUP, AUTO\_TBL\_MAINT, AUTO\_REORG, AUTO\_RUNSTATS, AUTO\_STATS\_PROF, AUTO\_PROF\_UPD). As mentioned in the introduction, some of these are already selected because they are activated by default when creating a database.

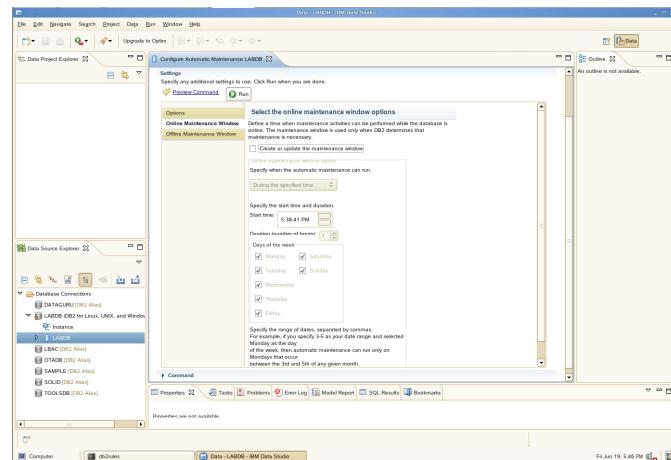


**Note:** If AUTO\_MAINT is turned off then all other options will become disabled. This is because this option is necessary for the others to be active. A similar situation occurs for the AUTO\_TBL\_MAINT option and ATUO\_REORG, AUTO\_RUNSTATS, AUTO\_STATS\_PROF, and AUTO\_PROF\_UPD.

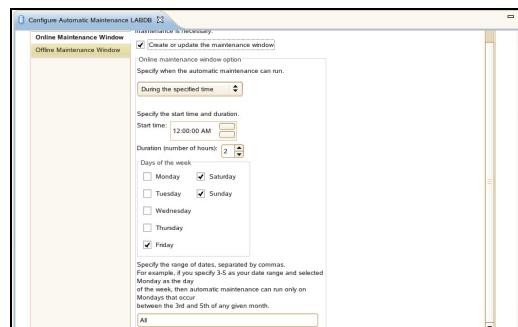
## 5.3 Selecting the Online Maintenance Window

By defining the online maintenance window we will be able to select a time when the maintenance activities can be performed while the database is online. This is only used when DB2 determines that maintenance is necessary.

1. Select the **Online Maintenance Window** menu option to receive the following screen.



2. Select the **Create or update the maintenance window** button to select a time slot.
3. Specify any Start Time and a Duration for the online maintenance activity. You can also specify the **Days of the week** or **Days of the Month** for these activities to run. Select any days/months you would like.



For the purpose of this document we selected:

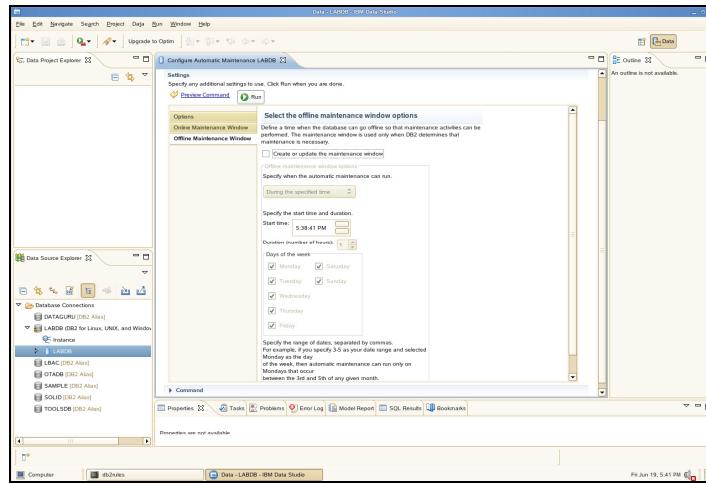
- **Start time:** 12:00:00 AM
- **Duration (number of hours):** 2
- **Days of the week:** Friday, Saturday and Sunday
- **Days of the month:** All

Review the maintenance windows you have selected.

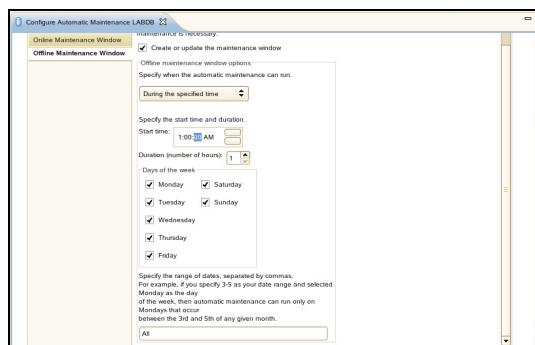
## 5.4 Selecting the Offline Maintenance Window

It is also possible to define an offline maintenance window to perform maintenance on the database when it goes offline.

- Select the **Offline Maintenance Window** menu option to receive the following screen.



- Select the **Create or update the maintenance window** button to select a time slot.
- Specify any Start Time and a Duration for the online maintenance activity. You can also specify the **Days of the week** or **Days of the Month** for these activities to run. Select any days/months you would like.



For the purpose of this document we selected:

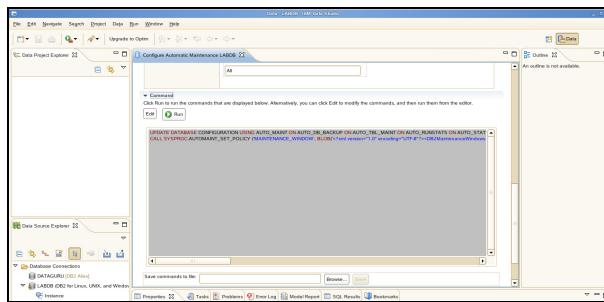
- Start time:** 4:00:00 AM
- Duration (number of hours):** 1
- Days of the week:** Sunday
- Days of the month:** All

Review the maintenance windows you have selected.

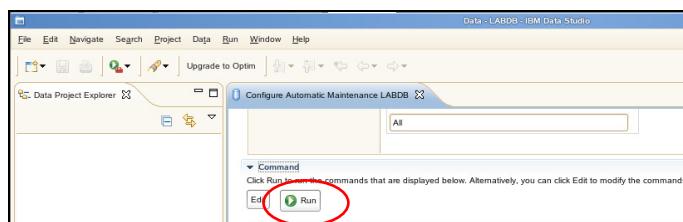
## 5.5 Perform Maintenance Selection against LABDB

All of the selections defined in the previous sections have not yet been committed. Make sure that you are satisfied with all of your selections and perform the following steps:

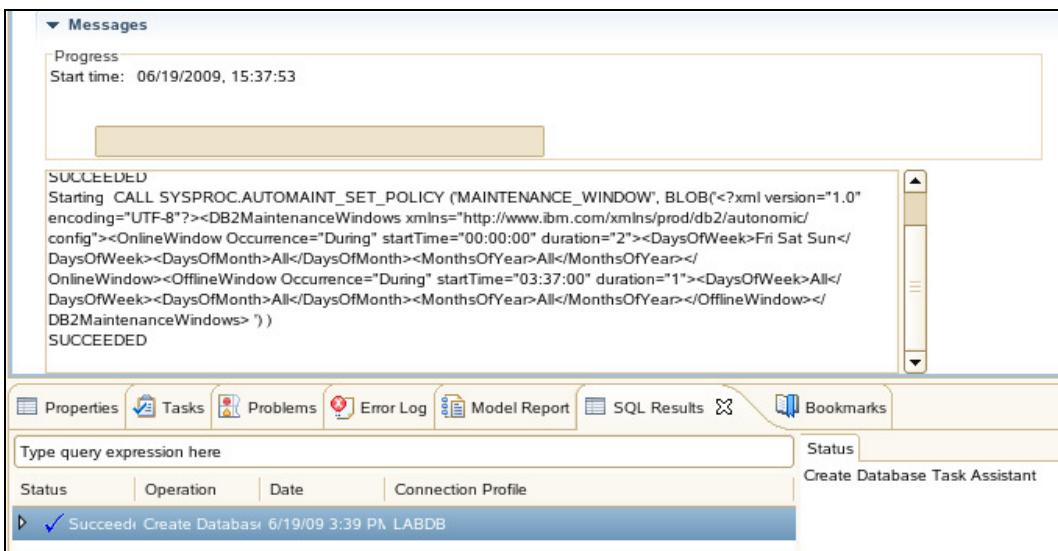
1. Click on the **Preview Command** link located underneath the *Settings* header. This will allow you to see the code that will be executed against the database.



2. Click **Run** to perform these actions against the database.



All actions should complete successfully.



We have now succeeded in setting up regular maintenance activities such as backups, table reorganizations, and statistics collection for the database. These maintenance activities will run only if needed. As an example, a backup of a database will not initialize unless there has been changes in data or configuration on the database.

## 5.6 Enabling Utility Throttling

Utility throttling regulates the performance impact of maintenance utilities so that they can run concurrently during production periods. The throttling system ensures that the throttled utilities are run as frequently as possible without violating the impact policy. You can throttle statistics collection, backup operations, rebalancing operations, and asynchronous index cleanups. Let's execute a command to limit the impact of these utilities, and view the current configuration for the LABDB database.

1. Execute the following two commands in a terminal window:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 update dbm cfg using
util_impact_lim 10
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 get db cfg for labdb
```

Specifying a *util\_impact\_lim* (impact policy) value of 10 can expect that a throttled backup invocation will not impact the workload by more than 10 percent. The following output should display (note that this is not the entire output):

Database Configuration for Database labdb	
Self tuning memory	(SELF_TUNING_MEM) = ON
Size of database shared memory (4KB)	(DATABASE_MEMORY) = AUTOMATIC
Database memory threshold	(DB_MEM_THRESH) = 10
Max storage for lock list (4KB)	(LOCKLIST) = AUTOMATIC
Percent. of lock lists per application	(MAXLOCKS) = AUTOMATIC
...	...
Automatic maintenance	(AUTO_MAINT) = ON
Automatic database backup	(AUTO_DB_BACKUP) = ON
Automatic table maintenance	(AUTO_TBL_MAINT) = ON
Automatic runstats	(AUTO_RUNSTATS) = ON
Automatic statement statistics	(AUTO_STMT_STATS) = ON
<b>Automatic statistics profiling</b>	<b>(AUTO_STATS_PROF) = ON</b>
<b>Automatic profile updates</b>	<b>(AUTO_PROF_UPD) = ON</b>
Automatic reorganization	(AUTO_REORG) = ON
...	...

From the output of the above, confirm that all automatic maintenance utilities are enabled with the **ON** clause beside each. Also notice that **SELF\_TUNING\_MEM** is turned on, this was set in the beginning of the lab after running the configuration advisor on the database.

---

## 6. Design Advisor

The task of selecting which indexes, MQTs, clustering dimensions, or database partitions to create for a complex workload can be quite daunting. The Design Advisor identifies all of the objects needed to improve the performance of the workload.

## 6.1 Create the Explain Tables

Create Explain tables to capture access plans when the Explain facility is activated. The Explain tables must be created before Explain can be invoked.

1. Issue the following two commands:

```
db2inst1@db2rules:~> cd /home/db2inst1/sqllib/misc
db2inst1@db2rules:~/sqllib/misc> db2 -tvf EXPLAIN.DDL
```

This will create Explain tables to capture access plans when the Explain facility is activated. The Explain tables must be created before Explain can be invoked. All commands should execute successfully.

## 6.2 Design Advisor Recommendations

The Design advisor can be used to get recommendations to tune individual SQL statements or a workload of statements. The Design Advisor analyzes a specified workload and considers factors such as the type of workload statements, the frequency with which a particular statement occurs, and characteristics of your database to generate recommendations that minimize the total cost to run the workload.

1. Change back to the directory containing all of the lab scripts:

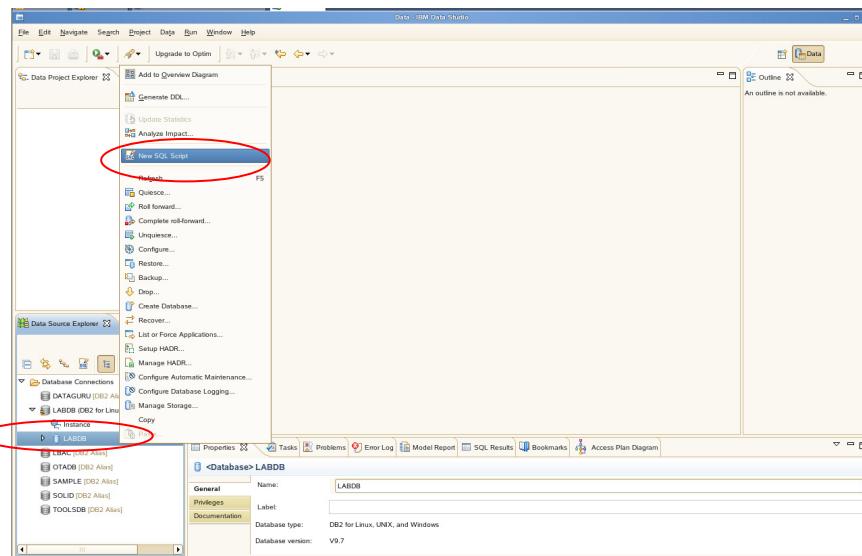
```
db2inst1@db2rules:~/sqllib/misc> cd /home/db2inst1/Documents/LabScripts/Autonomics
```

2. View and execute the initDB2advis script to create duplicate tables of existing ones that will be used for this section.

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 -tvf initDB2advis.sql
```

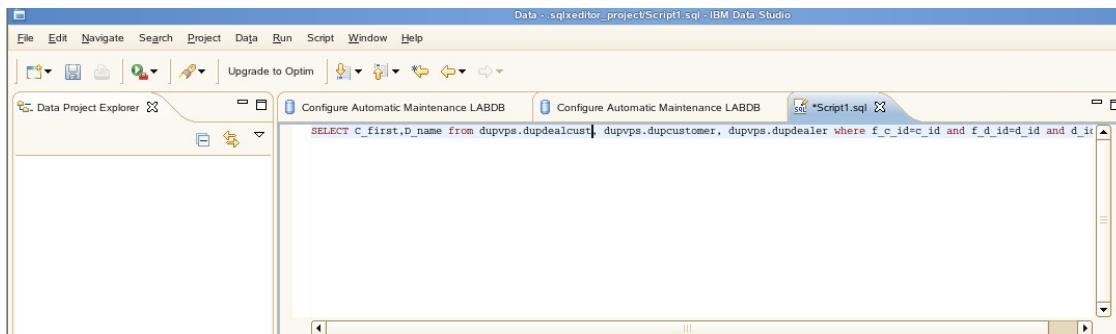
Creation of all tables and indexes should execute successfully. We will now use Visual Explain to see the Query plan and table scans required for a particular query. We will then use Design Advisor to see if we can improve the amount of time it takes to execute this query by creating indexes.

3. From Data Studio right-click on the database artifact **LABDB**. A menu will appear, select **New SQL Script** from within this menu.

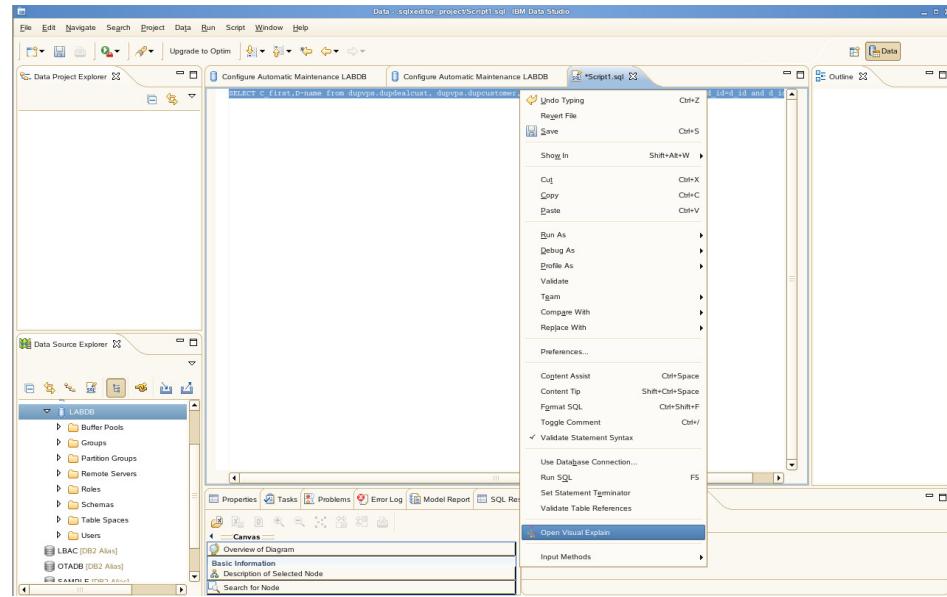


4. Copy the following query into the “*Explain Query Statement - LABDB*” box:

```
SELECT c_first,d_name FROM dupvps.dupdealcust,
dupvps.dupcustomer, dupvps.dupdealer WHERE f_c_id=c_id AND
f_d_id=d_id AND d_id=548
```

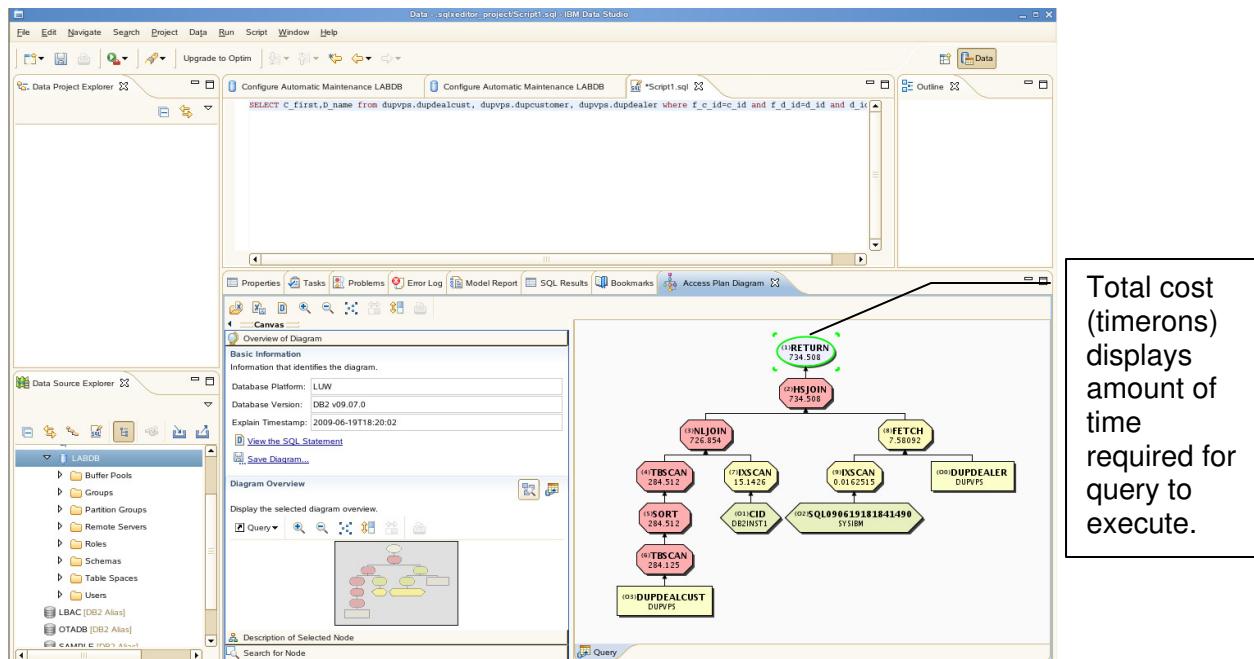


5. After inputting the above statement right-click within the screen and select **Open Visual Explain** from within this menu.



The Visual Explain screen lets you view the access plan for explained SQL or XQuery statements as a graph. You can use the information available from the graph to tune your queries or create objects within your database for better performance.

Accept the defaults in the pop-up by clicking **Finish**. The following will display:



We can see from the Explain Query that a table scan on DUPDEALCUST is done. We can now invoke the Design Advisor to see if there are any suggestions that can help us improve this query plan.

6. Invoke the Design advisor by executing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2advis -d labdb -s "SELECT c_first, d_name FROM dupvps.dupdealcust, dupvps.dupcustomer, dupvps.dupdealer WHERE f_c_id=c_id AND f_d_id=d_id AND d_id=548"
```

As you can see, we are calling the Design advisor with the same command that we used in the Explain Query so that it can suggest the necessary things to help us improve performance. The following output should display:

```
... ...
[34.01%] improvement ...
-- LIST OF RECOMMENDED INDEXES
-- =====
-- index[1], 0.884MB
CREATE INDEX "DB2INST1"."IDX907021751190000" ON "DUPVPS  "."DUPDEALCUST"
("F_D_ID" ASC, "F_C_ID" ASC) ALLOW REVERSE SCANS COLLECT SAMPLED DETAILED
STATISTICS;
COMMIT WORK ;
-- index[2], 0.013MB
CREATE INDEX "DB2INST1"."IDX907021751120000" ON "DUPVPS  "."DUPDEALER"
("D_ID" ASC, "D_NAME" DESC) ALLOW REVERSE SCANS COLLECT SAMPLED DETAILED
STATISTICS;
COMMIT WORK ;
...
...
```

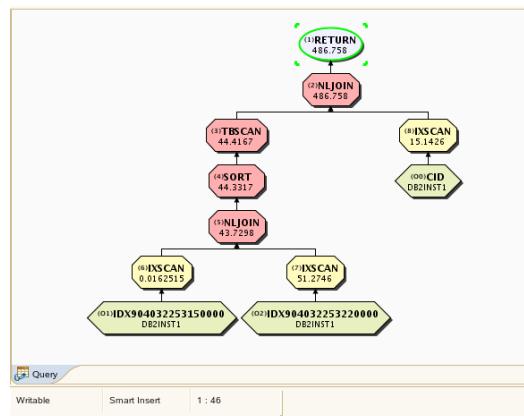
**Size of index provided for convenience.**

We will now create the recommended indexes and confirm the improvement in the Query plan by executing the crt\_index\_recmd.sql script.

7. Execute the crt\_index\_recmd.sql script in DB2 by issuing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 -tvf
crt_index_recmd.sql
```

8. View the new execution plan for the query by going through **steps 3-5**.



You can see the query plan is better since DB2 has avoided a table scan and the *Total cost (timerons)* has significantly decreased.

## 6.3 Executing the Query and Seeing Bufferpool Memory Changes

Now that we have optimized the database for that particular query, let's execute this query and use memory tracker to see what happens with the bufferpools and all other memory components as this query is executed.

1. Let's start by decreasing the current size of the bufferpool to something very small (1.3 MB) to ensure change after execution of the query and immediately take snapshot of the memory:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 alter bufferpool
vpsbuff immediate size 50 automatic; db2mtrk -d > bfr_query.snap;
```

Using gedit to view the bfr\_query.snap file (please note that this is system dependent and therefore, your values may be slightly different):

Tracking Memory on: 2009/02/01 at 16:35:31						
Memory for database: LABDB						
bph (2) 1.3M	utilh 64.0K	pckcacheh 1.1M	other 128.0K	catcacheh 576.0K	bph (1) 4.4M	
bph (S32K) 832.0K	bph (S16K) 576.0K	bph (S8K) 448.0K	bph (S4K) 384.0K	shsorth 128.0K	lockh 16.7M	
dbh 22.9M	apph (61747) 64.0K	apph (61716) 64.0K	apph (61715) 64.0K	apph (61714) 64.0K	apph (61713) 64.0K	
apph (61712) 64.0K	apph (61711) 64.0K	appshrh 576.0K				

BPH (2) should be the VPSBUFF buffer pool associated with the LABDB database. We can verify this by executing the following query: `db2 "SELECT * FROM SYSCAT.BUFFERPOOLS"` which provides the ID of each bufferpool.

During creation we used the `SIZE AUTOMATIC` clause for the bufferpool meaning that it is allowed to expand if needed. Let's see how this bufferpool will expand when we run the select query from the previous section.

2. Execute the query by issuing the following command:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2 "SELECT
c_first,d_name FROM dupvps.dupdealcust, dupvps.dupcustomer,
dupvps.dupdealer WHERE f_c_id=c_id AND f_d_id=d_id AND d_id=548"
```

This query should return all of the records with these specifications.

3. We can now check how the BP memory allocation has changed during this query:

```
db2inst1@db2rules:~/Documents/LabScripts/Autonomics> db2mtrk -d >
after_query.snap
```

Something similar to the following output should display:

Tracking Memory on: 2009/02/01 at 02:58:45						
Memory for database: LABDB						
utilh	pckcacheh	other	catcacheh	bph (2) 16.1M	bph (1)	
64.0K	1.3M	128.0K	1.3M		9.6M	
bph (S32K)	bph (S16K)	bph (S8K)	bph (S4K)	shsorth	lockh	
832.0K	576.0K	448.0K	384.0K	704.0K	16.7M	
dbh	apph (5939)	apph (5936)	apph (4764)	apph (4766)	apph (4765)	
24.1M	64.0K	64.0K	64.0K	64.0K	64.0K	
apph (4763)	apph (4762)	apph (4761)	appshrh			
64.0K	64.0K	128.0K	832.0K			

Therefore, it is evident that this bufferpool has increased in size to accommodate the query that was being performed.

## 7. Self-Tuning Memory Manager (STMM)

A revolutionary memory tuning system called the **Self-Tuning Memory Manager** was introduced in DB2 9. It works on main database memory parameters including buffer pools, sort heaps, locklist, package cache, and total database memory. It allows hands-off online memory tuning without DBA intervention by sensing the underlying workload and tunes the memory based on needs. When workload shifts, and memory redistribution is required to achieve optimal performance, STMM can adapt quickly to adjust memory configuration.

Self tuning memory simplifies the task of memory configuration by automatically setting values for memory configuration parameters and sizing buffer pools. When enabled, the memory tuner dynamically distributes available memory resources between several memory consumers including sort, package cache and lock list areas and buffer pools.

STMM can be enabled for all of the major memory consumers within DB2 or it can be individually enabled for each of the following memory consumers:

- Buffer pools (controlled by the `ALTER BUFFERPOOL` and `CREATE BUFFERPOOL` statements)
- Package cache (controlled by the `pckcachesz` configuration parameter)
- Locking memory (controlled by the `locklist` and `maxlocks` configuration parameters)
- Sort memory (controlled by the `sheapthres_shr` and the `sortheap` configuration parameters)
- Database shared memory (controlled by the `database_memory` configuration parameter)

In this section of the lab, we will enable STMM for the LABDB database and explore a few of the related database configuration parameters.

### 7.1 Connect to the LABDB Database and Examine the Database Configuration Parameters

1. Launch a terminal window or use one that is already open.
2. Connect to the LABDB database by issuing the following commands:

```
db2inst1@db2rules:~> db2 FORCE APPLICATION ALL
db2inst1@db2rules:~> db2 connect to labdb
```

The connection to the database should now be established.

3. Look at the database configuration parameters for the LABDB database. Execute the following command:

```
db2inst1@db2rules:~> db2 get db cfg for labdb show detail | more
```

Description	Parameter	Current Value	Delayed Value
Database configuration release level		= 0x0d00	
Database release level		= 0x0d00	
Database territory		= US	
Database code page		= 1208	
Database code set		= UTF-8	
Database country/region code		= 1	
Database collating sequence	(ALT_COLLATE)	= IDENTITY	IDENTITY
Alternate collating sequence		=	
Number compatibility		= OFF	
Varchar2 compatibility		= OFF	
Date compatibility		= OFF	
Database page size		= 4096	4096
Dynamic SQL Query management	(DYN_QUERY_MGMT)	= DISABLE	DISABLE
Statement concentrator	(STMT_CONC)	= OFF	OFF
Discovery support for this database	(DISCOVER_DB)	= ENABLE	ENABLE
Restrict access		= NO	
Default query optimization class	(DFT_QUERYOPT)	= 5	5
Degree of parallelism	(DFT_DEGREE)	= 1	1
Continue upon arithmetic exceptions	(DFT_SQLMATHWARN)	= NO	NO
Default refresh age	(DFT_REFRESH_AGE)	= 0	0
Default maintained table types for opt	(DFT_MTTB_TYPES)	= SYSTEM	SYSTEM
Number of frequent values retained	(NUM_FREQVALUES)	= 10	10
Number of quantiles retained	(NUM_QUANTILES)	= 20	20
Decimal floating point rounding mode	(DECFLT_ROUNDING)	= ROUND_HALF_EVEN	ROUND_HALF_EVEN
Backup pending		= NO	
All committed transactions have been written to disk		= NO	
Rollforward pending		= NO	
Restore pending		= NO	
Multi-page file allocation enabled		= YES	
Log retain for recovery status		= NO	
User exit for logging status		= NO	
Self tuning memory	(SELF_TUNING_MEM)	= ON (Active)	ON
Size of database shared memory (4KB)	(DATABASE_MEMORY)	= AUTOMATIC(34840)	AUTOMATIC(34840)
Database memory threshold	(DB_MEM_THRESH)	= 10	10
Max storage for lock list (4KB)	(LOCKLIST)	= AUTOMATIC(4120)	AUTOMATIC(4120)
Percent. of lock lists per application	(MAXLOCKS)	= AUTOMATIC(97)	AUTOMATIC(97)
Package cache size (4KB)	(PCKCACHESZ)	= AUTOMATIC(523)	AUTOMATIC(523)
--More--			

4. Press the **space bar** to scroll down until you see the section regarding STMM similar to the below screen.

**Note:** STMM is already enabled for the LABDB database. This is because the following are enabled by default during database creation:

- The value of the **SELF\_TUNING\_MEM** database parameter is **ON (Active)** which means STMM is enabled.
  - Notice that the value of the **DATABASE\_MEMORY** database parameter is **AUTOMATIC**. STMM can dynamically ask for and give back memory to the operating system.
  - Notice that the memory consumers that we have enabled for STMM all shows database parameter of value **AUTOMATIC**.
5. Enter “q” to exit this view or hit the space bar until you have scrolled through the output on screen.
  6. Let’s update the values described above to provide tuning changes when running the OLTP Scenario in the next section.

```
db2inst1@db2rules:~> db2 update db cfg for labdb using LOCKLIST
10 automatic MAXLOCKS 4 automatic PCKCACHESZ 100 automatic
SHEAPTHRES_SHR 5000 automatic SORTHEAP 256 automatic

db2inst1@db2rules:~> db2 FORCE APPLICATION ALL
```

To enable self tuning memory for buffer pools, you have to set the buffer pool size to AUTOMATIC. You can do this using the ALTER BUFFER POOL statement for existing buffer pools or the CREATE BUFFER POOL statement for new buffer pools. In this lab, we’ve already seen the automatic buffer pool increase in size when we used the `db2mttrk` command to check the memory status. This was seen in the previous section.

## 8. STMM in Action – VPS Demo

To showcase the features of STMM, we use the Virtual Production System (VPS) for DB2 as a backbone to drive a load on the database. With some of the out-of-the-box tools provided with DB2, we can watch as STMM takes over and adjusts many database parameters to make the currently running system more efficient.

### 8.1 Prepare to Execute the VPS Demo

1. Please close all windows from previous activities on the GUI desktop.
2. Open a terminal window and execute the `StartWAS.sh` script under the root user to start the Websphere® Application Server.

```
db2inst1@db2rules:~> su - root
```

**Password:** password (when prompted)

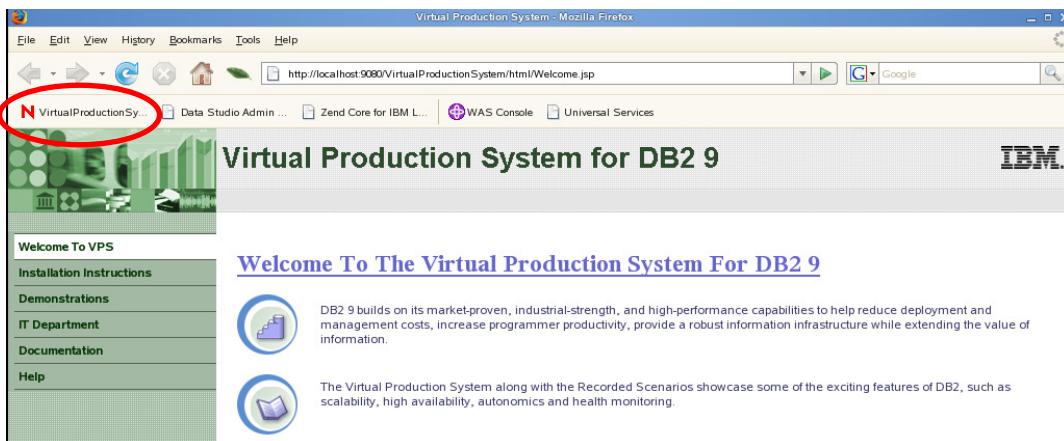
```
db2rules:~> /root/Desktop/StartWAS.sh
```

When this is completed, you should see the status of the WAS startup as active. At the end of this lab make sure to issue the StopWAS.sh script in order to free any resources from this script (these steps are described in the last section of the lab).

## 8.2 View the VPS Environment in a Web Browser

1. Launch the Firefox® browser to see the VPS application.

- ▶ Double-click on the **Firefox browser icon** on the desktop.
- ▶ Point your browser to the following website:  
[\*\*http://localhost:9080/VirtualProductionSystem\*\*](http://localhost:9080/VirtualProductionSystem) to receive the VPS welcome page.



2. Click on the “**Demonstrations**” link on the left-navigation bar.

- ▶ Enter or select the following information in the “**Application Configuration**” section:
  - Server: **localhost**
  - Port: **50001**
  - DBName: **labdb**
  - User: **db2inst1**
  - Pass: **password**
  - Driver: **JDBC Type 4**
- ▶ Press “**Connect**” to attempt a connection to the database from the application via WAS. Notice the status should show up as connected if the connection is successful.

**Virtual Production System - Mozilla Firefox**

File Edit View History Bookmarks Tools Help

http://localhost:9080/VirtualProductionSystem/html/Demo.jsp

N VirtualProductionSy... Data Studio Admin ... Zend Core for IBM L... WAS Console Universal Services

## Virtual Production System for DB2 9

Welcome To VPS

**Installation Instructions**

**Demonstrations** (Red Circle)

XML Scenario

OLTP Scenario

IT Department

Documentation

Help

**Application Configuration**

Server: localhost Status: Connected

Port: 50001 Connect

DBName: labdb

User: db2inst1

Pass: \*\*\*\*\*

Driver: JDBC Type 4

XML Scenario OI

## 8.3 Start the OLTP Transaction Workload

- Start the OLTP transaction workload as follows.
  - Click on the “**OLTP Scenario**” link under the “**Demonstrations**” link in the left-navigation bar in the VPS console.
  - Change the number of users under “**Demo Controls**” to **6** and click “**Update**” to see the “**Updated!**” text stating that this value has been changed.

Total Transactions:

### Demo Controls

Updated! Workload: Mixed

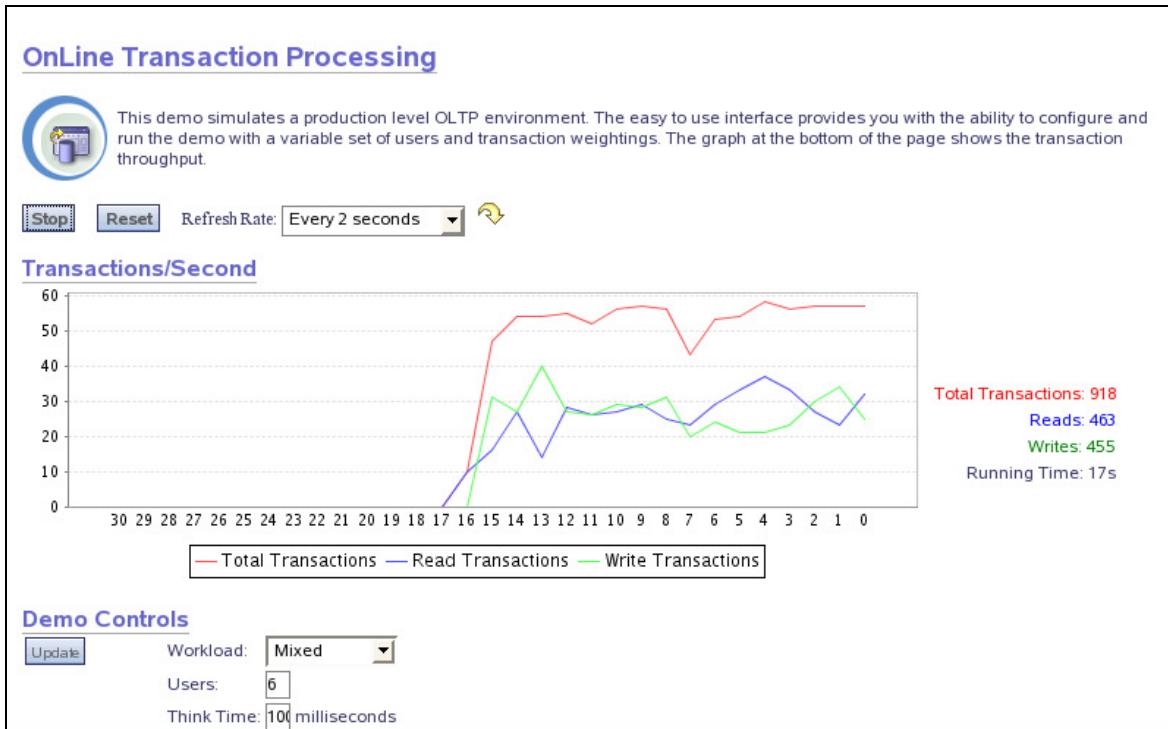
Users: 6 (Red Box)

Think Time: 100 milliseconds

- ▶ Press “Start” to start the OLTP load.

The screenshot shows the 'Virtual Production System for DB2 9' interface. On the left, there's a sidebar with links like 'Welcome To VPS', 'Installation Instructions', 'Demonstrations', 'XML Scenario', 'OLTP Scenario' (which is circled in red), 'Average Transaction Time', 'Locking Scenario', 'STMM Scenario', 'IT Department', and 'Documentation'. The main content area has a title 'OnLine Transaction Processing'. It includes a small icon of a database, a 'Start' button (also circled in red), a 'Reset' button, and a 'Refresh Rate' dropdown set to 'Never Refresh'. Below these are sections for 'Transactions/Second' and 'Demo Controls' (with fields for Update, Workload, Users, and Think Time).

The workload will now start, with the graph showing three different coloured lines representing **Total, Read, or Write Transactions**. The number of read, write and total transactions is displayed on the right of the graph.



## 8.4 Start the STMM Scenario

- Within the Firefox window, open a new tab containing the STMM scenario webpage.
  - Using your mouse, right-click on the “**STMM Scenario**” link under “**OLTP Scenario**”, then choose “**Open Link in New Tab**”.
  - A new tab will open displaying the STMM scenario webpage.

The screenshot shows the 'Virtual Production System for DB2 9' web interface. On the left, there's a sidebar with links like 'Welcome To VPS', 'Installation Instructions', 'Demonstrations', 'XML Scenario', 'OLTP Scenario', 'STMM Scenario' (which is selected), and 'IT Department', 'Documentation', 'Help'. The main content area has a title 'Virtual Production System for DB2 9' and a sub-section 'Self-Tuning Memory Management' with a note about the STMM Scenario guide. Below that is a section titled 'STMM Parameters' with a table. The table has columns for 'Parameter', 'Baseline Value', 'Current Value', 'Automatic Update', 'New Value', and 'Action'. The 'Baseline Value' column is highlighted with a red box. At the bottom of the table are buttons for 'Reset Baseline', 'All Auto', 'All Manual', and 'Update Values'.

Parameter	Baseline Value	Current Value	Automatic Update	New Value	Action
application heap size	256	256	<input checked="" type="checkbox"/>	256	
database memory	33360	33360	<input checked="" type="checkbox"/>	33360	
database heap	2286	2286	<input checked="" type="checkbox"/>	2286	
lock list	10	10	<input checked="" type="checkbox"/>	10	<button>Obtain Locks</button>
max locks	4 %	4 %	<input checked="" type="checkbox"/>	4	<button>Tune Cache</button>
package cache size	100	100	<input checked="" type="checkbox"/>	100	<button>Execute Sorts</button>
sort heap threshold	5000	5000	<input checked="" type="checkbox"/>	500	
sort heap	256	256	<input checked="" type="checkbox"/>	256	
stat heap size	4384	4384	<input checked="" type="checkbox"/>	438	
statement heap	2048	2048	<input checked="" type="checkbox"/>	2048	
buffer pool size	1000	1000	<input checked="" type="checkbox"/>		

- Take note of the **Baseline Value** column. This is the starting values that the VPS database will start off for various parameters in automatic state. It will take approximately **5 – 10 minutes** after starting the OLTP Scenario for STMM to tune the database parameters. This is because a lot of data is necessary to ensure that the parameters are tuned to the best possible values. So take a break while VPS does all of the work for you!
- By pressing on the “**Refresh**” button in Firefox, you will eventually see that the **Current Value** column will deviate from the **Baseline Value** column.

The screenshot shows a Mozilla Firefox browser window displaying the "Virtual Production System for DB2 9" webpage. The left sidebar contains navigation links for "Welcome To VPS", "Installation Instructions", "Demonstrations", "XML Scenario", "OLTP Scenario", "Average Transaction Time", "Locking Scenario", and "STMM Scenario". The main content area has a title "Virtual Production System for DB2 9" and a sub-section "Self-Tuning Memory Management" with a note about the STMM Scenario guide. Below this is a table titled "STMM Parameters" with the following data:

Parameter	Baseline Value	Current Value	Automatic Update	New Value	Action
application heap size	256	256	<input checked="" type="checkbox"/>	256	
database memory	33488	40900	<input checked="" type="checkbox"/>	40900	
database heap	2286	2286	<input checked="" type="checkbox"/>	2286	
lock list	10	4096	<input checked="" type="checkbox"/>	4096	
max locks	4 %	97 %	<input checked="" type="checkbox"/>	97	<button>Obtain Locks</button>
package cache size	100	2412	<input checked="" type="checkbox"/>	2412	<button>Tune Cache</button>
sort heap threshold	5000	4008	<input checked="" type="checkbox"/>	4008	
sort heap	256	206	<input checked="" type="checkbox"/>	206	<button>Execute Sorts</button>
stat heap size	4384	4384	<input checked="" type="checkbox"/>	4384	
statement heap	2048	2048	<input checked="" type="checkbox"/>	2048	
buffer pool size	1000	1000	<input checked="" type="checkbox"/>		

At the bottom of the table are buttons for "Reset Baseline", "All Auto", "All Manual", and "Update Values".

- Any values for parameters that have changed will appear in red.

## 8.5 Stop the STMM Scenario

- At this point, please click back to the first tab showing the “OLTP Scenario” webpage, and click on the **Stop** button to halt the transactions.



2. Please close the Firefox browser window by clicking on the “X” in the top right-hand corner of the window, and accept any warnings that may appear regarding closing multiple tabs.
3. To free up virtual machine resources, you should now stop the WebSphere Application Server. Open a terminal window and execute the **StopWAS.sh** script under the root user.

```
db2inst1@db2rules:~> su - root
Password: password (when prompted)
```

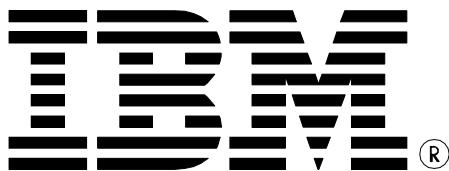
```
db2rules:~> /root/Desktop/StopWAS.sh
```

## 8.6 Clean Up

To clean your environment after completing the exercise, if you would like to redo this lab in the future, please execute the following in a terminal window:

```
su - root
cd /home/db2inst1/Documents/LabScripts/Autonomics
make clean
make install
exit
```

Enter the password “**password**” when prompted.



© Copyright IBM Corporation 2011  
All Rights Reserved.

IBM Canada  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

Printed in Canada  
01/01/2011

IBM, IBM (logo), DB2, WebSphere, and WebSphere Application Server are trademarks of International Business Machines Corporation in the United States, other countries, or both.

VMware is a trademark or VMware Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION  
PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY  
OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING,  
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-  
INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A  
PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

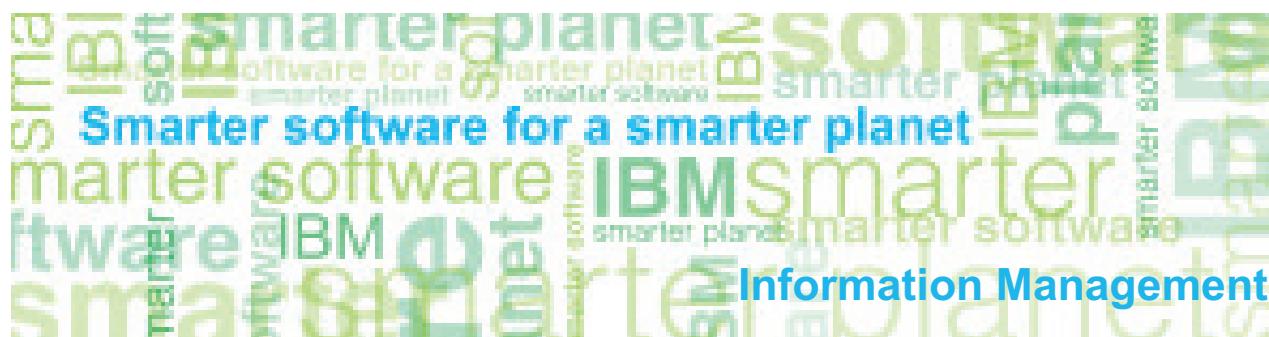
This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

The information in this publication is provided AS IS without warranty. Such information was obtained from publicly available sources, is current as of July 2009, and is subject to change. Any performance data included in the paper was obtained in the specific operating environment and is provided as an illustration. Performance in other operating environments may vary. More specific information about the capabilities of products described should be obtained from the suppliers of those products.

# DB2® 9.7 Advanced Features



© 2011 IBM Corporation



## Disclaimer

© Copyright IBM Corporation 2011. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Other company, product, or service names may be trademarks or service marks of others.

DB2 Workshop for Oracle Professionals page 91 of 207.

© 2011 IBM Corporation

## Agenda

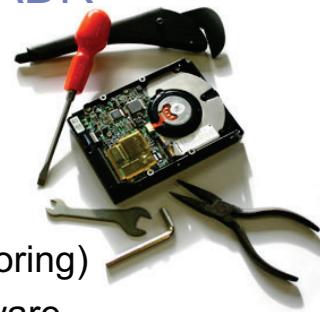
- **High Availability and Disaster Recovery – HADR**
- **DB2 pureScale**
- **DB2 Partitioning**
- **DB2 Storage Optimization**
- **DB2 LBAC**
- **DB2 pureXML**
- **DB2 Workload Management**

## Agenda

- **High Availability and Disaster Recovery – HADR**
- **DB2 pureScale**
- **DB2 Partitioning**
- **DB2 Storage Optimization**
- **DB2 LBAC**
- **DB2 pureXML**
- **DB2 Workload Management**

## High Availability and Disaster Recovery - HADR

- **High Availability:** keeps the data available whenever required by eliminating or at least minimizing downtime.
  - Commons solutions for HA:
    - Advanced storage scenario (like split mirroring)
    - Database partitioning with clustering software
    - Data replication (SQL or Q replication)
- **Disaster Recovery:** protect against data loss caused by unrecoverable failure.
  - Common solutions for DR:
    - Backup and Restore
    - Log shipping

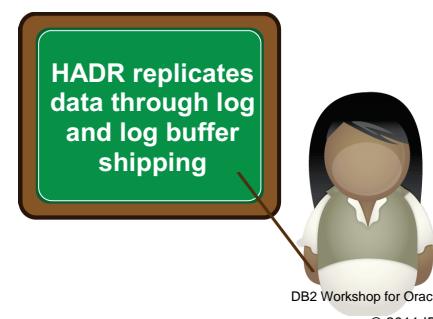
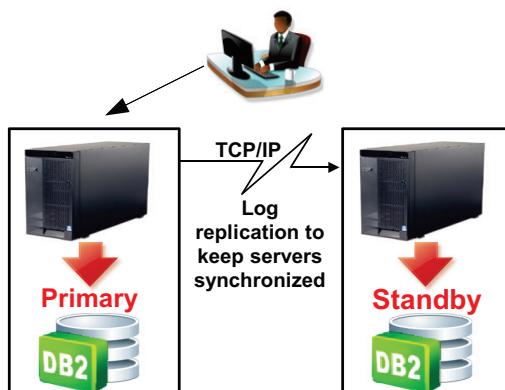


5

© 2011 IBM Corporation

## High Availability and Disaster Recovery - HADR

- **HADR solution is based on a synchronized pair (primary and standby) of databases. In case of failure the standby database can take over the workload.**
- **Primary**
  - Handles all client connections and processes transactions
  - Continuously ships transaction logs to the standby over the network
- **Standby**
  - Kept in sync with primary by applying received transaction logs



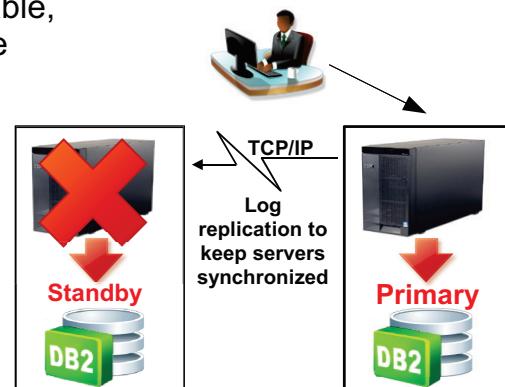
# High Availability and Disaster Recovery

- Primary Failure (planned or unplanned)

- The standby can take over the transactional workload and it becomes the new primary
  - When the failed machine becomes available, it will be synchronized and it becomes the new standby

- **Standy Failure (planned or unplanned)**

- Primary continues to process transactions
  - When Standby becomes available again, it gets re-integrated and after catching up on missed transactions the HADR pair will reach peer state



- **Automatic Failover with Cluster Manager**

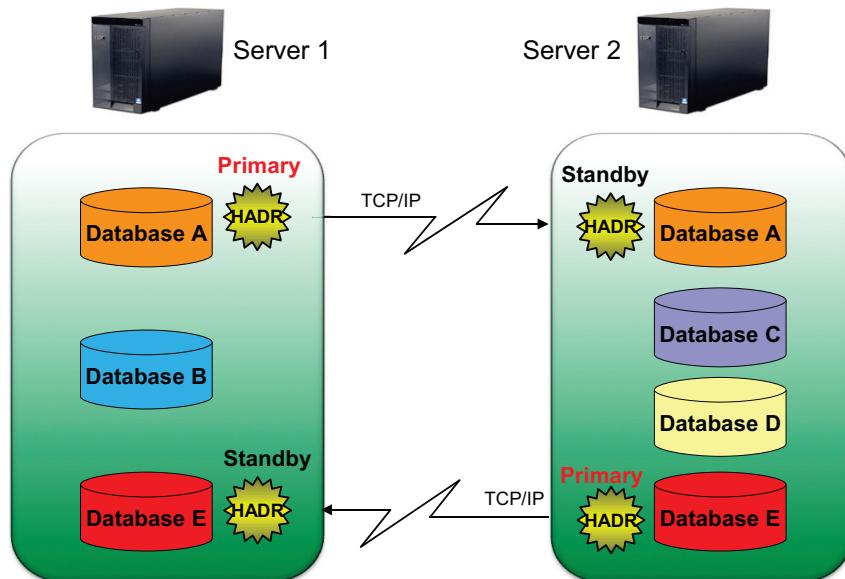
- Integrated with Tivoli System Automation for Multiplatforms for node failure detection and automatic failover
  - Easy setup using the “db2haicu” wizard

7

© 2011 IBM Corporation

## HADR Scope of Action

- HADR replication takes place at the database level



## HADR Synchronous Mode

- **Synchronous**

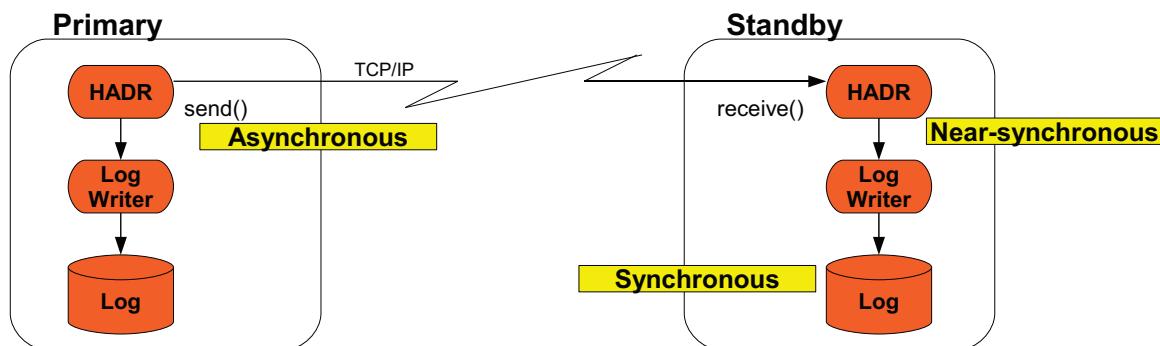
- Greatest protection against transaction loss
- A higher cost of transaction response time

- **Near Synchronous (default)**

- Better performance than Synchronous mode
  - no waiting for the log buffer to be written to disk on standby

- **Asynchronous**

- If latency of the network between the two servers is too great



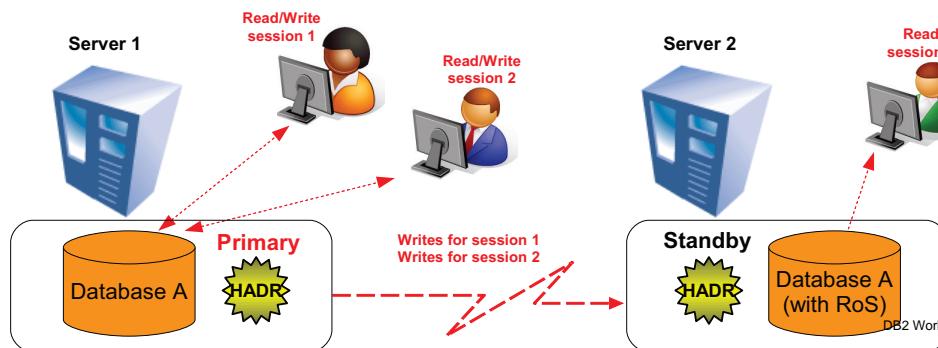
9

© 2011 IBM Corporation

## HADR Reads on Standby (RoS)

New in DB2 9.7 FP1

- Perform read-only operations on HADR Standby database
- Run concurrent read-only workloads with minimal impact to Standby system's high availability and disaster recovery role
- Offload reporting, DSS/BI workloads to Standby
- During failover, DB2 seamlessly turns the read-on-standby to a primary read/write server
- To enable RoS, set **DB2\_HADR\_ROS** registry variable to ON



10

DB2 Workshop for Oracle Professionals page 95 of 207.

© 2011 IBM Corporation

## Oracle Data Guard

- Oracle has the same modes as DB2's HADR.

Oracle Mode	DB2 Mode
Maximum Performance	DB2 asynchronous
Maximum Availability	DB2 near synchronous
Maximum Protection	DB2 synchronous

- Feature Availability Comparison

Oracle Express	Not available	DB2 Express-C	Not available
Oracle Standard One	Not available	DB2 Express FTL DB2 Express	Free Purchase
Oracle Standard	Not available	DB2 Workgroup	Purchase
Oracle Enterprise	Purchase	DB2 Enterprise	Purchase

## Agenda

- High Availability and Disaster Recovery – HADR
- DB2 pureScale
- DB2 Partitioning
- DB2 Storage Optimization
- DB2 LBAC
- DB2 pureXML
- DB2 Workload Management

## DB2 pureScale – Designed for OLTP

- **Unlimited Capacity**

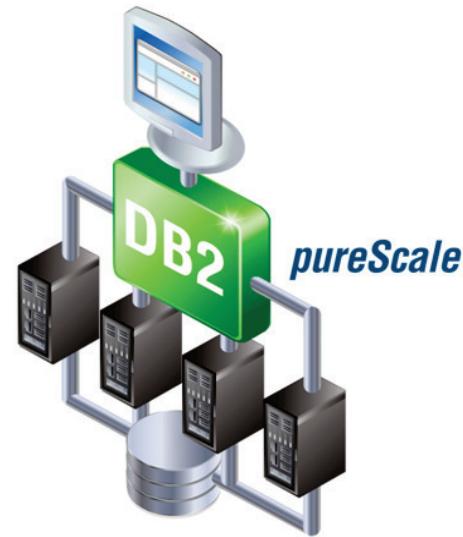
- Buy only what you need, add capacity as your needs grow
- Handle key capacity spikes with **pay by the day pricing**

- **Application Transparency**

- Easy to implement, easy to grow

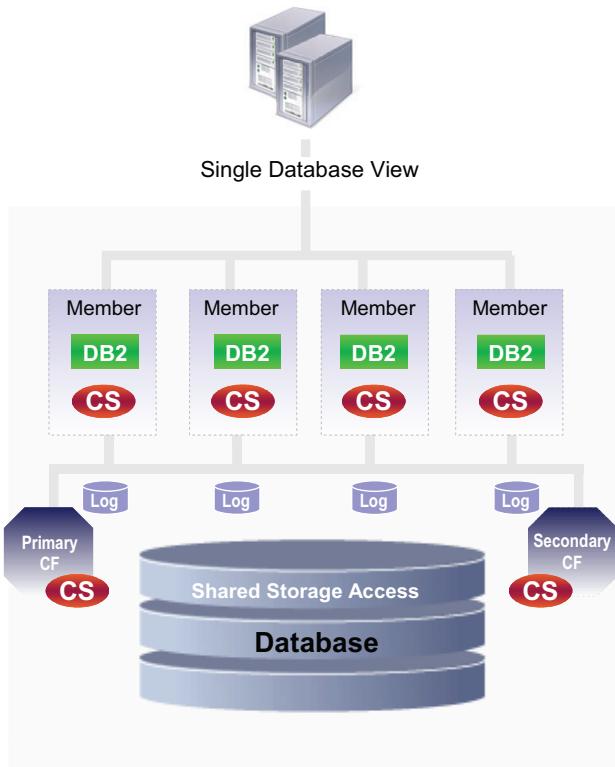
- **Continuous Availability**

- 24x7 availability so key systems never go down, even if multiple servers fail
- DB2 recovers itself from crashes



*Learning from the undisputed Gold Standard - System z*

## DB2 pureScale Architecture Overview



Clients connect anywhere,...  
... see single database

- Clients connect into any member
- Automatic load balancing and client reroute may change underlying physical member to which client is connected

DB2 engine runs on several host computers

- Co-operate with each other to provide coherent access to the database from any member

Integrated cluster services

- Failure detection, recovery automation, cluster file system
- In partnership with STG and Tivoli

Low latency, high speed interconnect

- Special optimizations provide significant advantages on RDMA-capable interconnects (eg. InfiniBand)

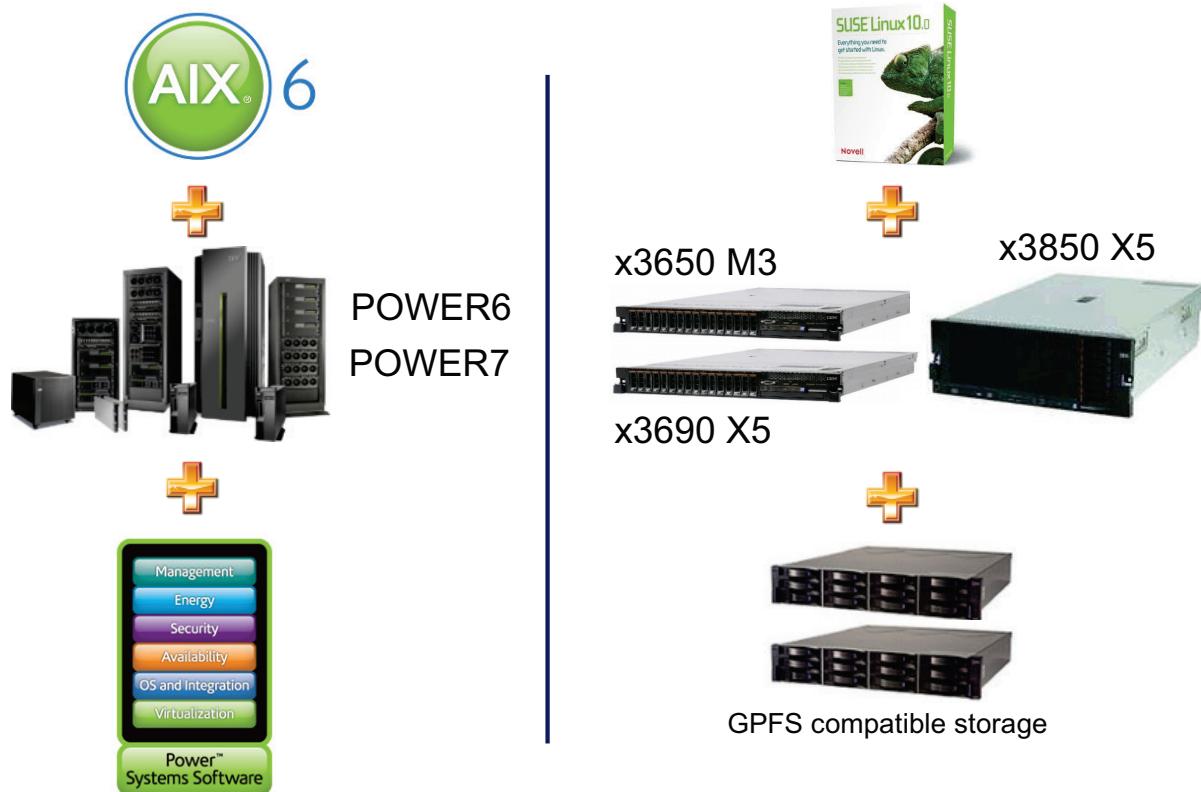
Cluster caching facility (CF)

- Efficient global locking and buffer management
- Synchronous duplexing to secondary ensures availability

Data sharing architecture

- Shared access to database
- Members write to their own logs on shared disk
- Logs accessible from another host (used during recovery)

## DB2 pureScale Architecture



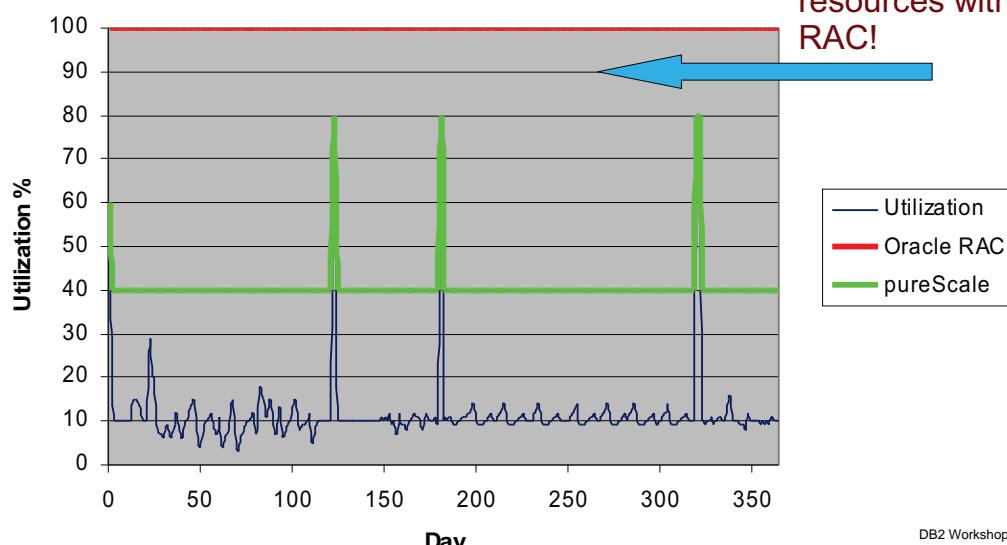
15

© 2011 IBM Corporation

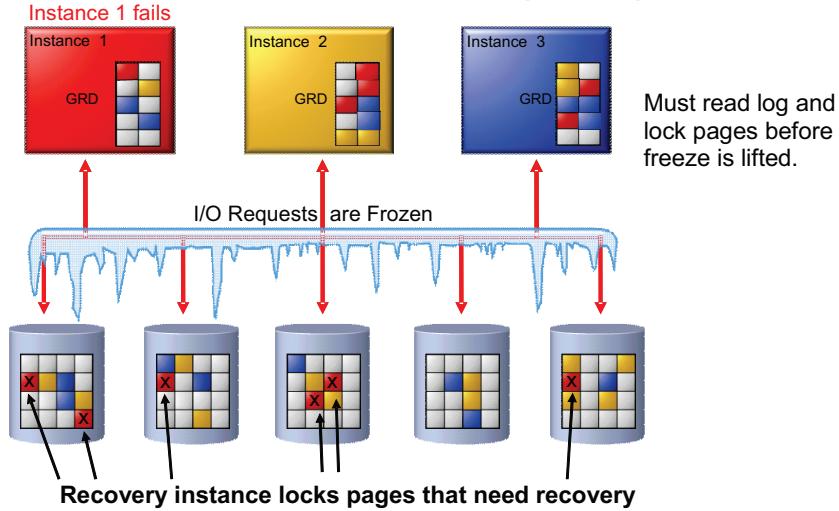
## Purchasing Choices

- **System Utilization:**
  - **Traditional**
    - Buy more licenses than you need and sit idle most of the year
  - **Oracle**
    - Sells the customer Oracle RAC and sits idle ALL year with a big bill
  - **DB2 pureScale**
    - Just add capacity when needed

Paying for unused resources with RAC!



## Oracle Real Application Clusters (RAC)



- Distributed locking in RAC results in higher overhead and lower scalability
  - Oracle RAC best practices recommends
    - Fewer rows per page (to avoid hot pages)
    - Partition database to avoid hot pages
    - Partition application to get some level of scalability
    - All of these result in higher management and development costs

17

© 2011 IBM Corporation

# Agenda

- High Availability and Disaster Recovery – HADR
  - DB2 pureScale
  - **DB2 Partitioning**
  - DB2 Storage Optimization
  - DB2 LBAC
  - DB2 pureXML
  - DB2 Workload Management

## DB2 Database Partition Feature (DPF)

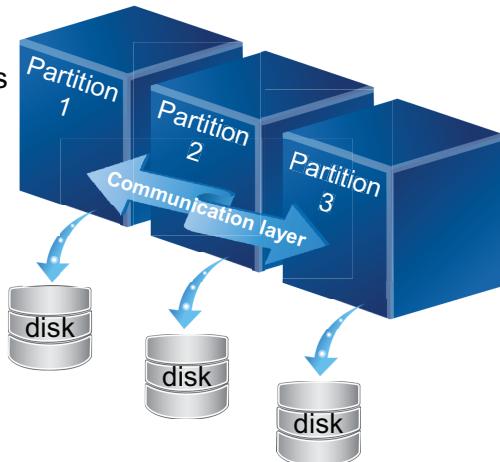
## **Split database data across multiple servers**

- Benefits:

- Support to various hardware configurations
  - Improved performance
  - Scalability
  - Parallelism
  - Split workload
  - Shared nothing architecture
  - Query optimization

- Main target:

- Data mining
  - Data warehousing
  - Online analytical processing



**DPF is a feature available for FREE in any InfoSphere Warehouse package**

19

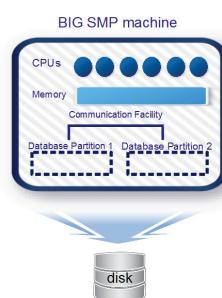
© 2011 IBM Corporation

## DB2 DPF – Supported Configurations

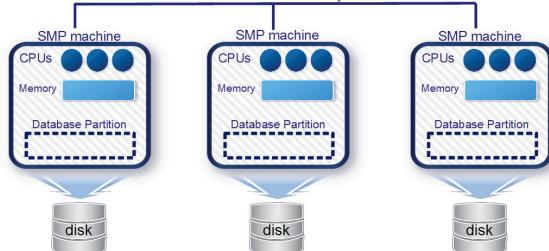
## Single node



## Multiple logical nodes in a single box

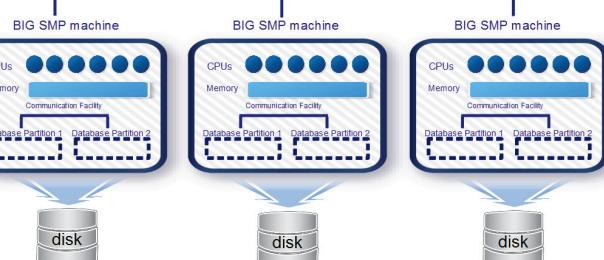


## Communication Facility



## Multiple physical nodes

Communication Facility

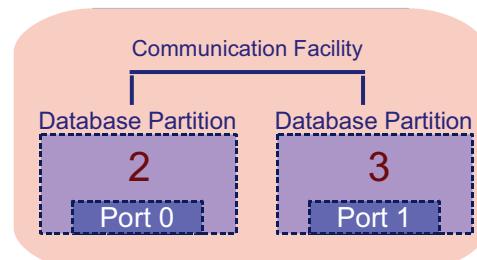


Multiple physical and logical nodes DSC Workshop for Change Professionals

## DPF – Node Configuration (dbnodes.cfg)

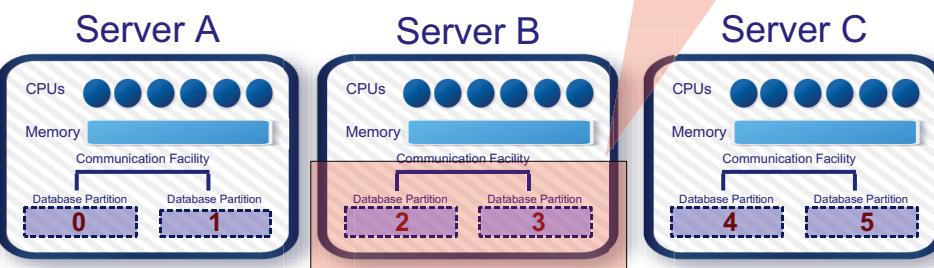
- DB partition number:** Unique database partition ID
- Server hostname:** Machine's name or IP address
- Server logical-port:** Logical partition ID within a machine

Partition	Server Name	Logical Port
0	Server A	0
1	Server A	1
2	Server B	0
3	Server B	1
4	Server C	0
5	Server C	1



21

© 2011 IBM Corporation



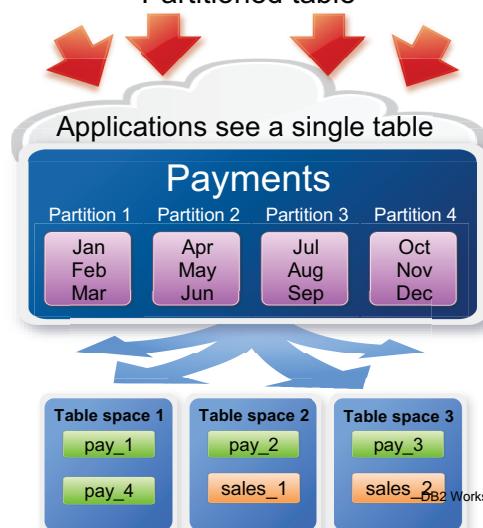
## Table Partitioning

- Allows a single logical table to be broken up into multiple separate physical storage objects (a.k.a. Partitions)
  - Up to 32K range partitions
  - Each partition defines a range of values
  - A partition will only contain rows that match its range of values

Non-partitioned table



Partitioned table



## Table Partitioning - Features

### ▪ Roll-in / Roll-out: ALTER TABLE ... ATTACH / DETACH

- Incorporates / exclude an existing table as a new range
- Data becomes visible / invisible all at once after **COMMIT**
- Minimal interruption to other queries accessing table
- **No data movement → Fast Operation**



### ▪ Table Partition Elimination

- Ability to determine that only a subset of the data partitions in a table are necessary to answer a query

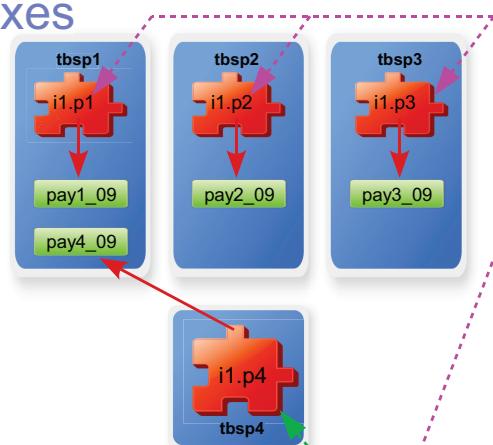


© 2011 IBM Corporation

23

## Table Partitioning – Partitioned Indexes

- Improved data roll-in and roll-out performance
- Less contention on index pages, because the index is partitioned
- Each table partition has its own index partition
- Each index partition is stored in the same table space as the corresponding data partition (override using INDEX IN clause).

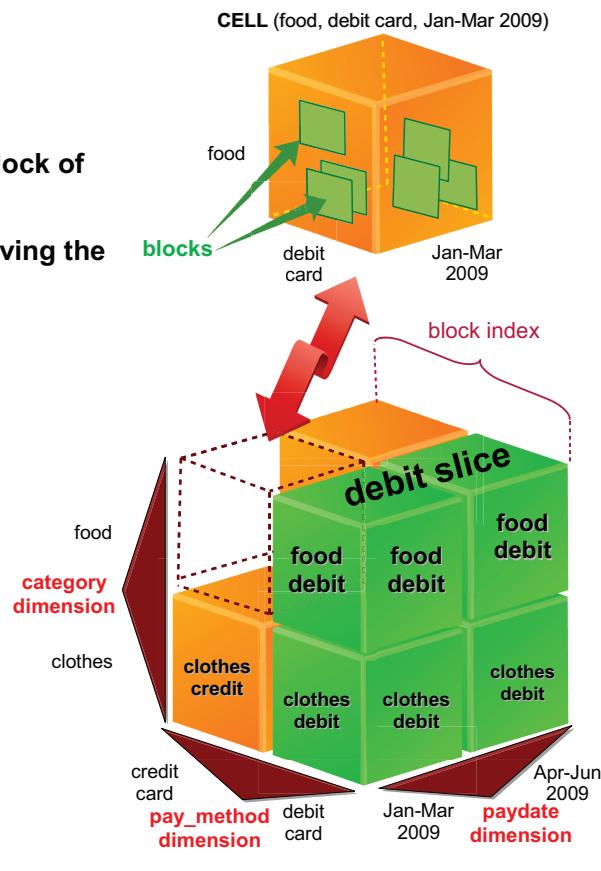


### PARTITIONED INDEX EXAMPLE

```
CREATE TABLE payments(id INT, paydate DATE,
type VARCHAR,...)
IN tbsp1, tbsp2, tbsp3
PARTITION BY RANGE (paydate)
( STARTING FROM '1/1/2009' ENDING 3/31/2009,
STARTING FROM '4/1/2009' ENDING 6/30/2009,
STARTING FROM '7/1/2009' ENDING 9/30/2009,
STARTING FROM '10/1/2009' ENDING 12/31/2009 INDEX IN tbsp4
) default
```

## Multi-Dimensional Clustering

- Blocks:** consecutive set of pages on the disk.
- Block Indexes:** indexes that point to an entire block of pages
- Cells:** portion of the table that contains rows having the same unique set of dimension values
- Dimension:** axis along which data is organized in an MDC table
  - Good candidates:
    - columns referenced by **ORDER BY** and **GROUP BY** clauses
    - Columns used for range, equality, and **IN** predicates
    - Roll-in or roll-out** of data
- Slice:** portion of the table that contains all the rows that have a specific value for one of the dimensions.

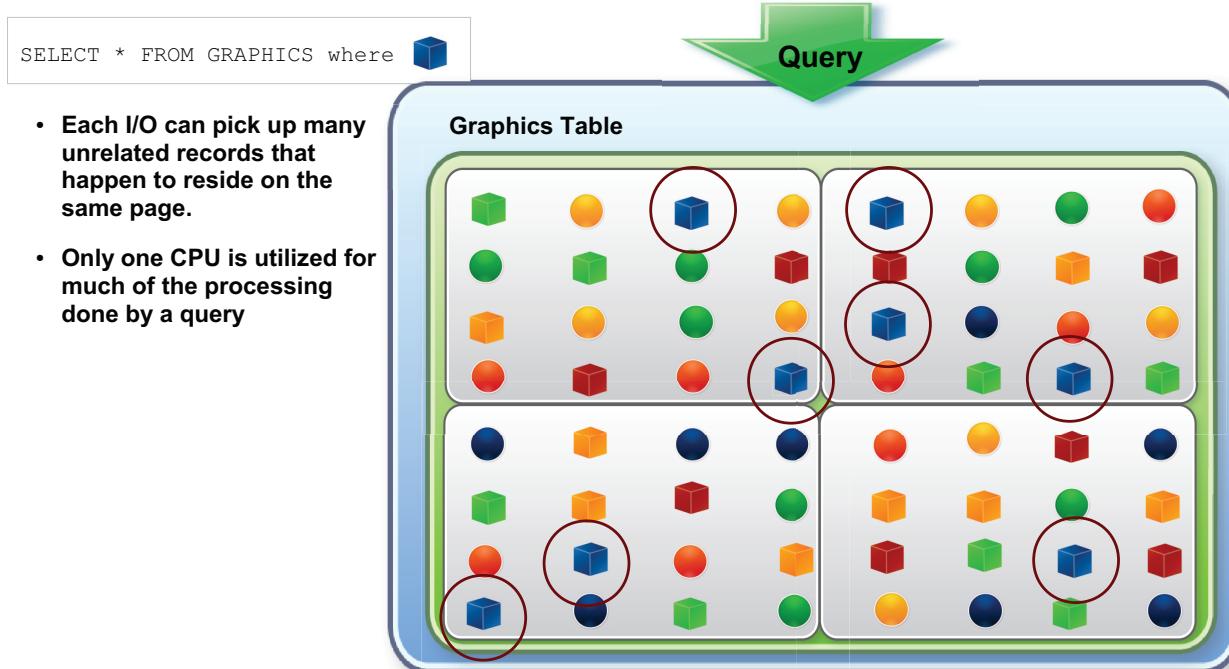


© 2011 IBM Corporation

26

## Combining All Features

Single large table without partitioning



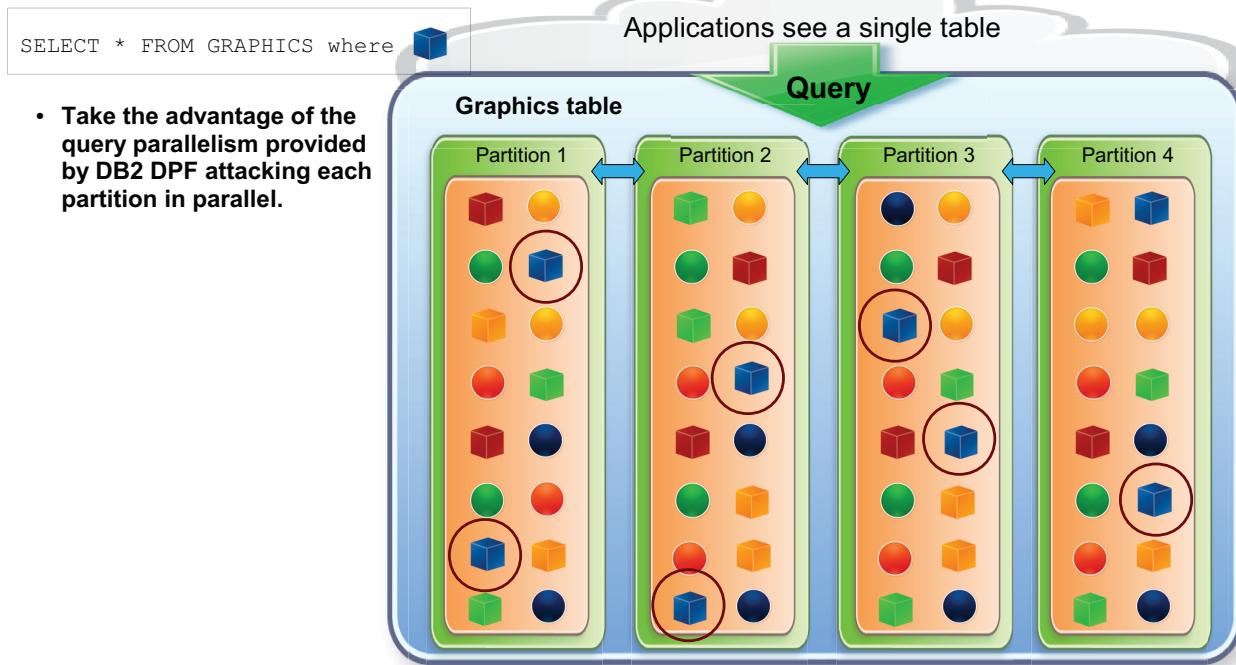
DB2 Workshop for Oracle Professionals page 103 of 207.

© 2011 IBM Corporation

26

## Combining All Features

### Using database partitioning

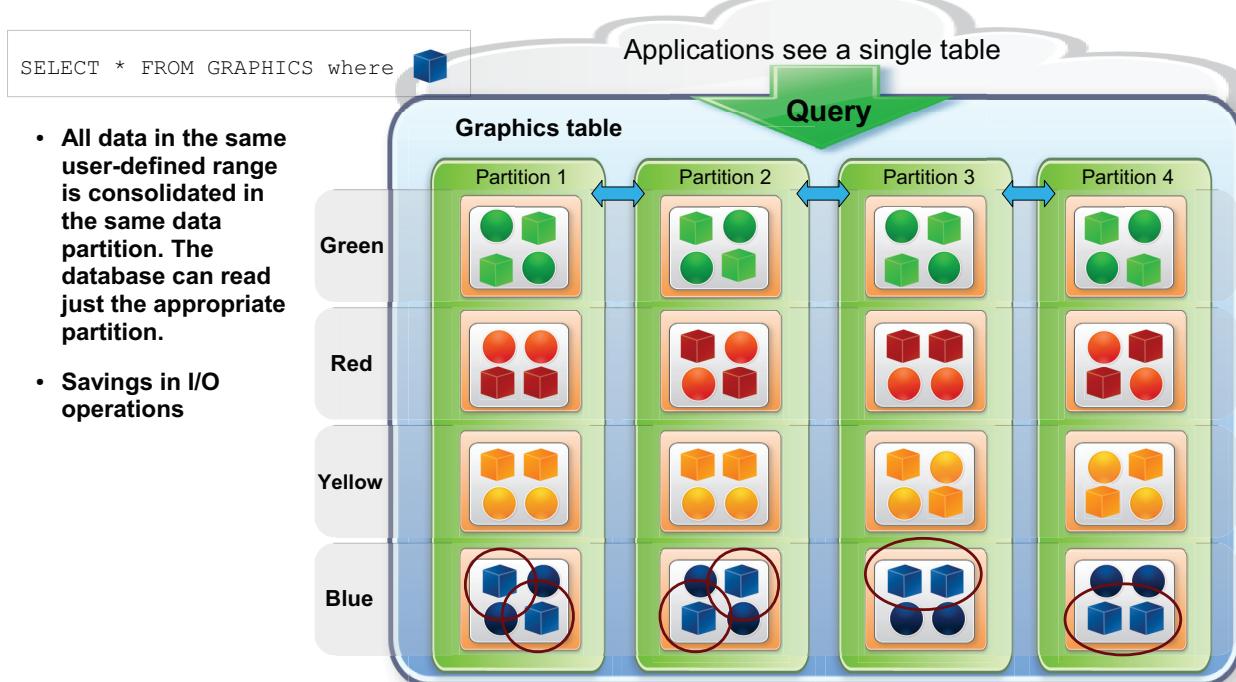


27

© 2011 IBM Corporation

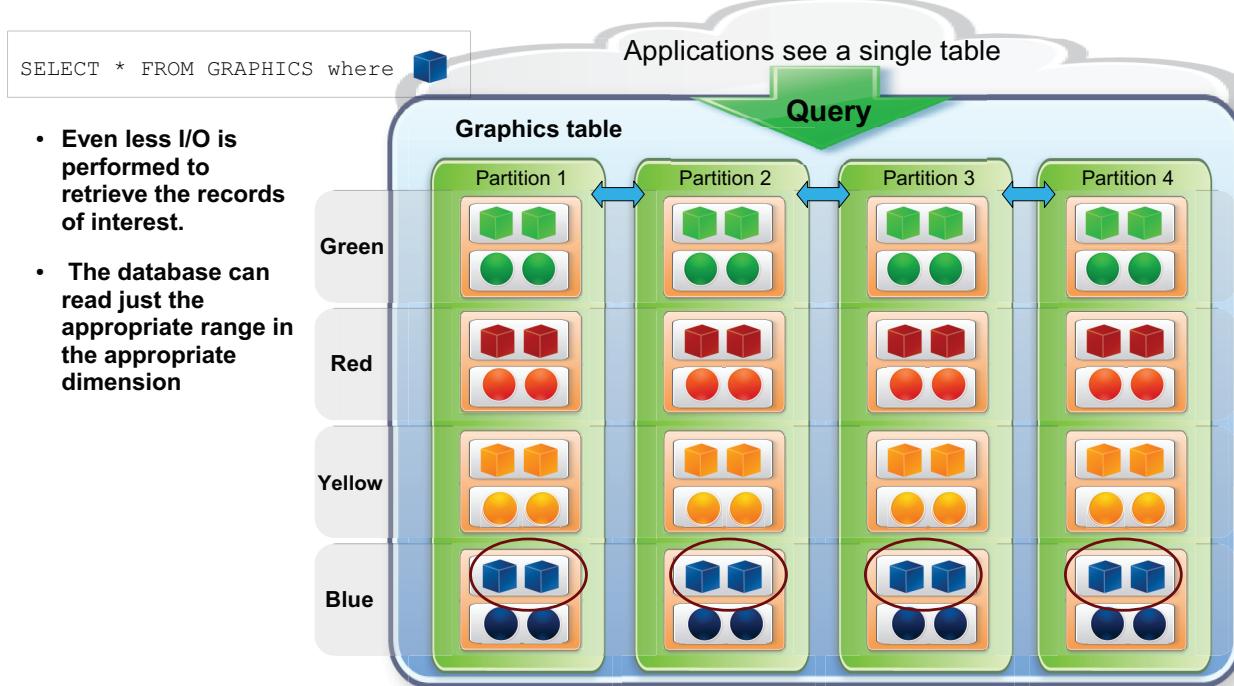
## Combining All Features

### Using database partitioning and table partitioning



## Combining All Features

Using database partitioning, table partitioning, and MDC



29

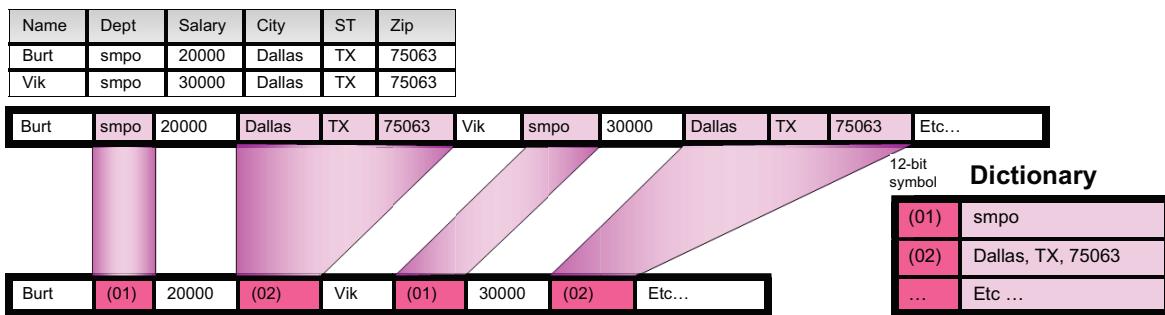
© 2011 IBM Corporation

## Agenda

- High Availability and Disaster Recovery – HADR
- DB2 pureScale
- DB2 Partitioning
- **DB2 Storage Optimization**
- DB2 LBAC
- DB2 pureXML
- DB2 Workload Management

## Storage Optimization – Table Compression

- **Dictionary based - symbol table for compressing/decompressing data records**
  - Dictionary per table stored within the permanent table object
- **Data resides compressed everywhere**
  - **Table space** - significant I/O bandwidth savings
  - **Buffer pools** - improvement in performance and memory savings
  - **Transaction logs** – significant disk space savings for backups



31

© 2011 IBM Corporation

## Storage Optimization – Oracle Comparison

- Oracle does allow tables or partitions of tables to be compressed  
**However,**
- ☒ While DB2 compresses out common values at the **TABLE LEVEL**, Oracle compresses at **PAGE LEVEL**
  - Repeating values in a single page are replaced by a symbol in Oracle
    - The symbols are stored in the page header
- ☒ Therefore, in Oracle:
  - Consistent repeating values throughout the entire table will be stored multiple times in each page header
  - There may be repeating patterns in the table but not on each page
    - Oracle will miss out on these compression opportunities
  - Oracle only supports compression for bulk loads
    - DB2 supports compression for load, insert, and import



*"Row-level compression is a revolutionary development that will leave Oracle and Microsoft green with envy".*

DB2 Workshop for Oracle Professionals page 106 of 207.

© 2011 IBM Corporation

## Temporary Table Compression

- **User Temporary Tables (DGTT/CGTT) and System Temporary Tables are automatically compressed as of DB2 9.7**
- **Prerequisites for Temporary Table Compression:**
  - ✓ Storage Optimization Feature license has been applied
  - ✓ There is sufficient memory available at the time to build the compression dictionary
  - ✓ DB2 Optimizer makes the final decision
- **Compression of temporary tables aims to:**
  - ✓ Reduce the amount of temporary disk space required
  - ✓ Have **no performance penalty** as a result of the extra processing required for row compression
  - ✓ May result in query performance increment
- **Realization of benefit**
  - Report on temp table space usage



33

© 2011 IBM Corporation

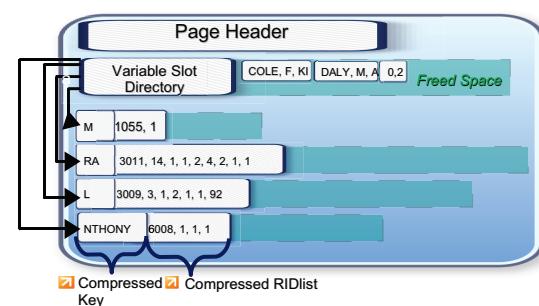
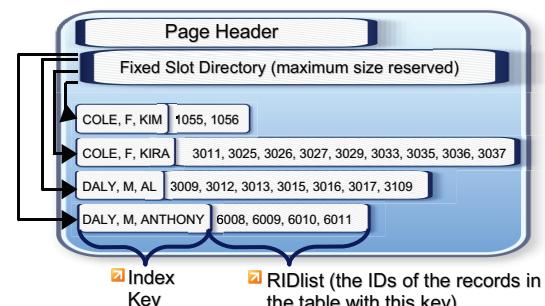
## Index Compression

### RIDlist compression

- **Different algorithms implemented by the DB2 engine for:**
  - RID list compression, prefix compression, variable slot directory, key compression
- **Activated by:**
  - Default when row compression is activated on a table

```
CREATE INDEX <index_name> COMPRESS YES
```

```
ALTER INDEX <index_name> COMPRESS YES
REORG INDEXES ON TABLE t1
```



## XML and LOB Inlining

- Default behavior**

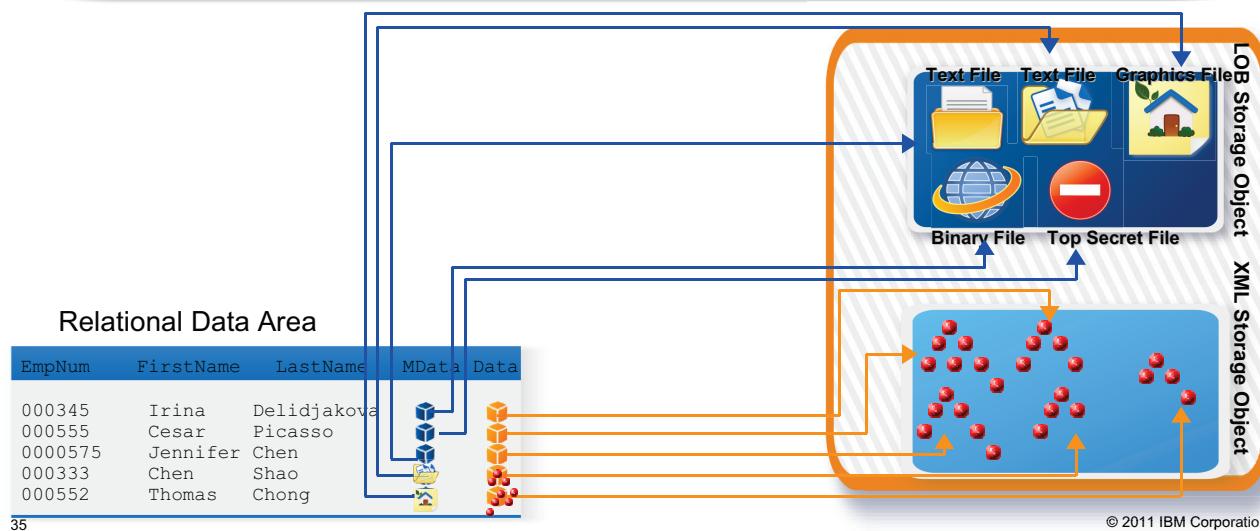
- All XML documents stored in XDA object; All LOB objects stored in LOB Storage object
- Descriptors in rows identify location of each object

- Inlining**

- Allows small (<32KB) XML documents and LOB to be stored in the same space as regular table data
- Better performance and reduced storage needs

```
CREATE TABLE PROJECTS (PID INTEGER, LEAD EMP INLINE LENGTH 300, STARTDATE DATE, ...)
```

```
ALTER TABLE PROJECTS ALTER COLUMN LEAD SET INLINE LENGTH 1004
```



35

© 2011 IBM Corporation

## XML (XDA) Compression

- XML docs in the XML Data Area (XDA) are compressed
- Same dictionary approach used for Table Compression
- Classic/'Offline' REORG table based
- Enablement is via the table COMPRESS attribute

```
CREATE/ALTER TABLE mytab1 COMPRESS YES
```

Relational Data Area

EmpNum	FirstName	LastName	Data
000345	Irina	Delidjakova	
000555	Cesar	Picasso	
0000575	Jennifer	Chen	
000333	Chen	Shao	
000552	Thomas	Chong	

```
REORG TABLE mytab1 LONGLOBDATA
```

Compressed Relational data incl. in-lined XML documents

Compressed XML documents, directly in XDA

DB2 Workshop for Oracle Professionals page 108 of 207.

© 2011 IBM Corporation

36

## Competitive Comparison

*Comparison to Oracle*

✓ DB2 delivers superior compression through additional capabilities not available in 11gR2

	DB2 9.7	Oracle 11gR2
Table Compression		
Large Domain Pattern Matching		
Column value compression		
Multi column compression		
Substring column compression		
Index compression		
Storage of one key per rid list		
Prefix compression for index		
Dynamic prefix compression		
Rid list compression		
Compression for sort overflows		
Compression for join temps		
Compression for aggregate temps		

## Agenda

- **High Availability and Disaster Recovery – HADR**
- **DB2 pureScale**
- **DB2 Partitioning**
- **DB2 Storage Optimization**
- **DB2 LBAC**
- **DB2 pureXML**
- **DB2 Workload Management**

## DB2 LBAC – Label Based Access Control

- Available with purchase of “DB2 Advanced Access Control Feature”
  - Both user sessions and data rows has an associated label for comparison
  - LBAC use rules for comparing users & data labels allow access controls to be applied at the row level



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

= row returned

© 2011 IBM Corporation

39

## DB2 LBAC – Label Based Access Control

- Why granular privileges?
  - The need to restrict access to specific portion of data in a table
- How to implement?
  - Views
    - 1)Simulate a new table
    - 2)Create a view (subset of the data from the base table)
    - 3)Authorize the user to access the view
    - 4)Revoke access from the user to the base table
- LBAC (Label Based Access Control)
  - Can restrict read/write access to row(s) and/or column(s) of a table

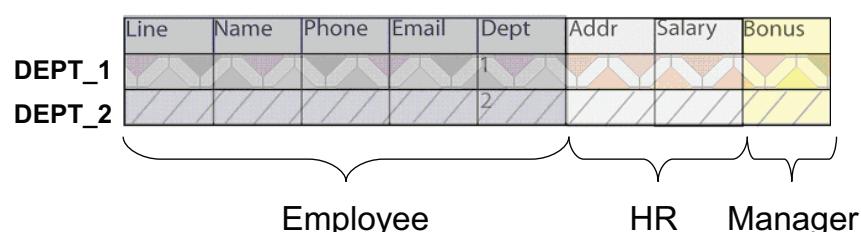
## DB2 LBAC – Label Based Access Control

### TWO levels of access control on protected tables

- Access Control at the table level via traditional privileges
  - Does the user hold the required privilege to perform the requested operation on the table?
- Label Based Access Control
  - Sets security labels at the row level, column level or both
- How does LBAC work?
  - Users and Objects (rows/columns) are assigned labels that are later compared to authorize access

## DB2 LBAC – Label Based Access Control

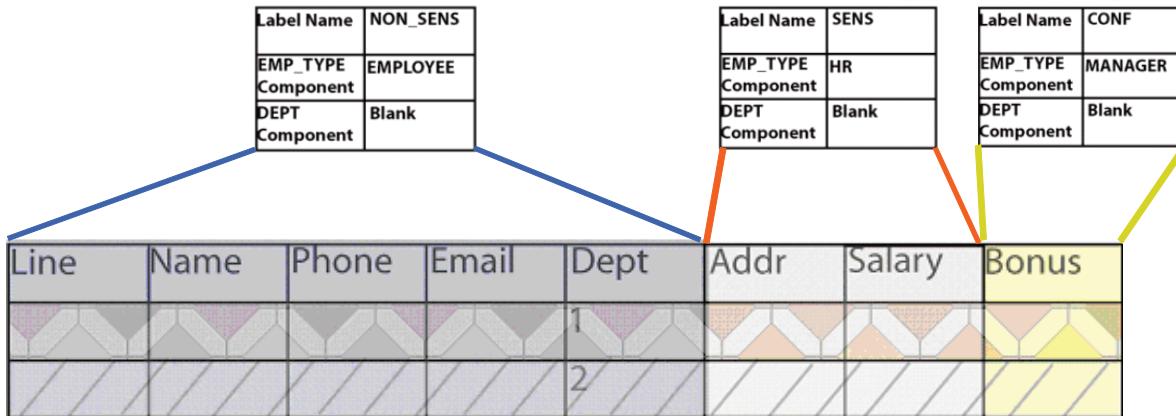
- Access Control at the table level via traditional privileges
  - Does the user hold the required privilege to perform the requested operation on the table?
- Label Based Access Control
  - Sets security labels at the row level, column level or both
- How does LBAC work?
  - Users and Objects (rows/columns) are assigned labels that are later compared to authorize access



## Use Case: Defining Column Labels

- **3 column labels:**

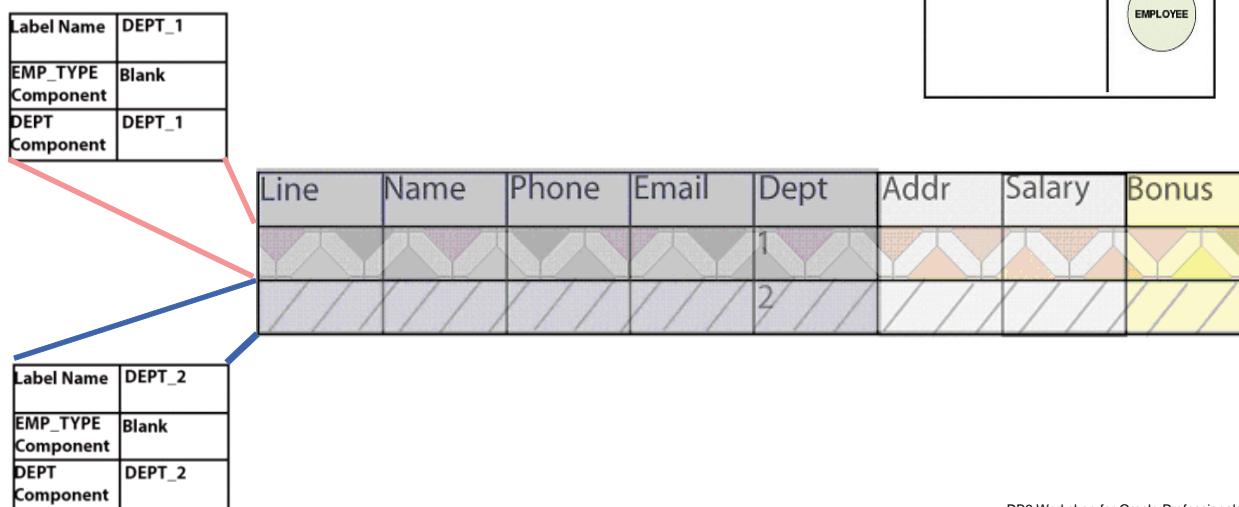
- **NON\_SENS** for non-sensitive data
- **SENS** for sensitive data
- **CONF** for confidential data



## Use Case: Defining Row Labels

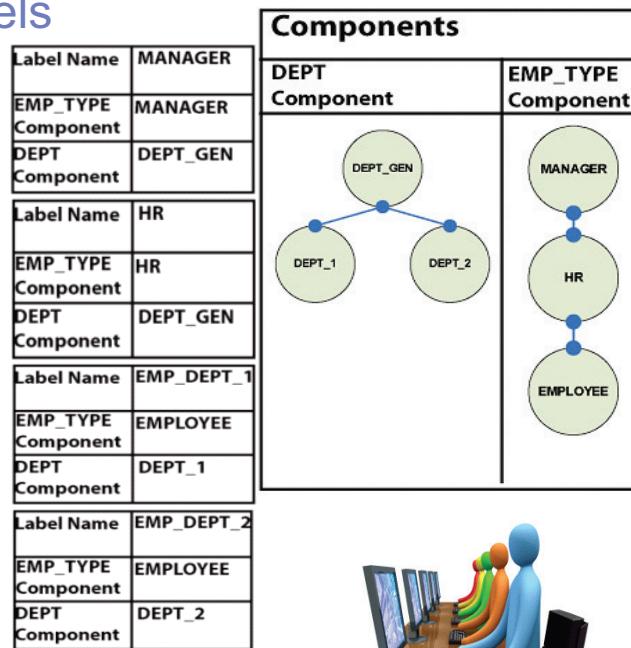
- **2 row labels:**

- **DEPT\_1** for Department 1
- **DEPT\_2** for Department 2



## Use Case: Defining User Labels

- **4 User labels:**
  - **MANAGER** for Managers in Department 1 or 2
  - **HR** for Human Resources in Department 1 or 2
  - **EMP\_DEPT\_1** for regular employees in Department 1
  - **EMP\_DEPT\_2** for regular employees in Department 2



```

CREATE SECURITY LABEL EMP_POLICY.EMP_DEPT_2
COMPONENT EMP_TYPE 'EMPLOYEE',
COMPONENT DEPT 'DEPT_2'

```



© 2011 IBM Corporation

45

## Protected Table

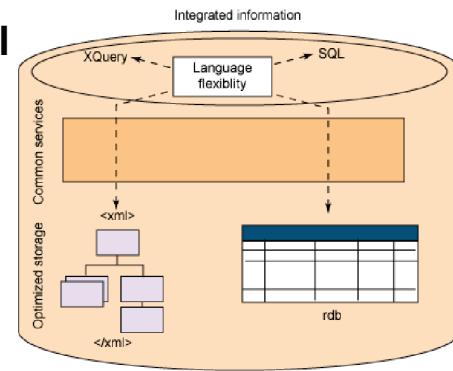
- **A protected table is a table on which LBAC is enforced**
- **Types of protected tables:**
  - Protected table with row level granularity
  - Protected table with column level granularity
- **Under LBAC, DBADM and SYSADM have no inherent ability to access data within a protected table!**
  - No security label or exemption, no data access!
- **The content of a protected table with row level granularity appears different depending on the user accessing such table**
  - **YOU ONLY SEE THE ROWS WHICH YOUR LBAC CREDENTIALS ALLOW YOU TO SEE**

## Agenda

- **High Availability and Disaster Recovery – HADR**
- **DB2 pureScale**
- **DB2 Partitioning**
- **DB2 Storage Optimization**
- **DB2 LBAC**
- **DB2 pureXML**
- **DB2 Workload Management**

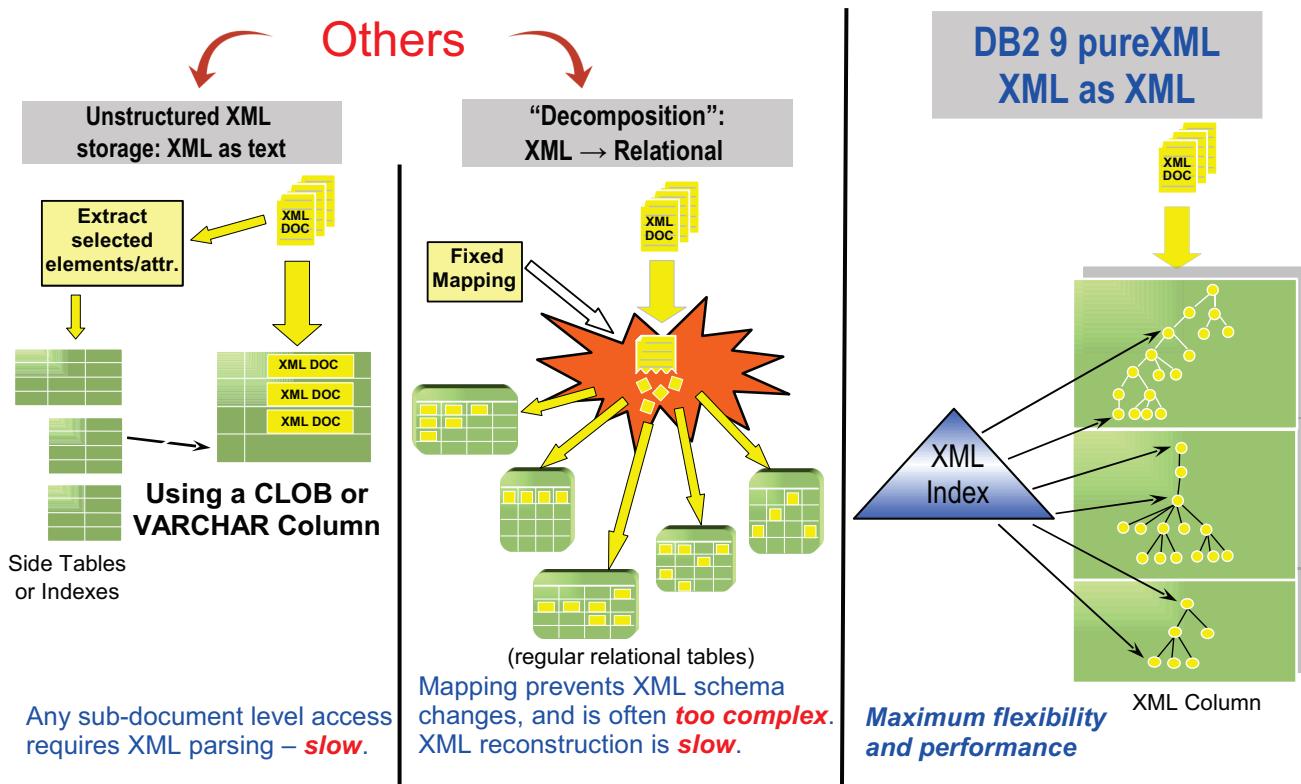
## Storing XML: Native XML Storage

- **DB2 stores XML in parsed hierarchical format (~DOM)**
- **Relational columns are stored in relational format (tables)**
- **XML columns are stored natively**
- **No XML parsing for query evaluation!**



- **Rows hold relational data**
- **XML values in pureXML (XDM) storage**
- **Descriptor pointing from row to XML storage**

## XML Storage: DB2 vs. Others



49

© 2011 IBM Corporation

## Quick Overview of pureXML Capabilities

- Support to XQuery and SQL / XML
- XML schema validation
- Transformation with XSLT stylesheet
- XML indexing
- Compression of XML data
- Application development support for C, C++, COBOL, JAVA, C#, VB, PHP, PERL and SQL routines.



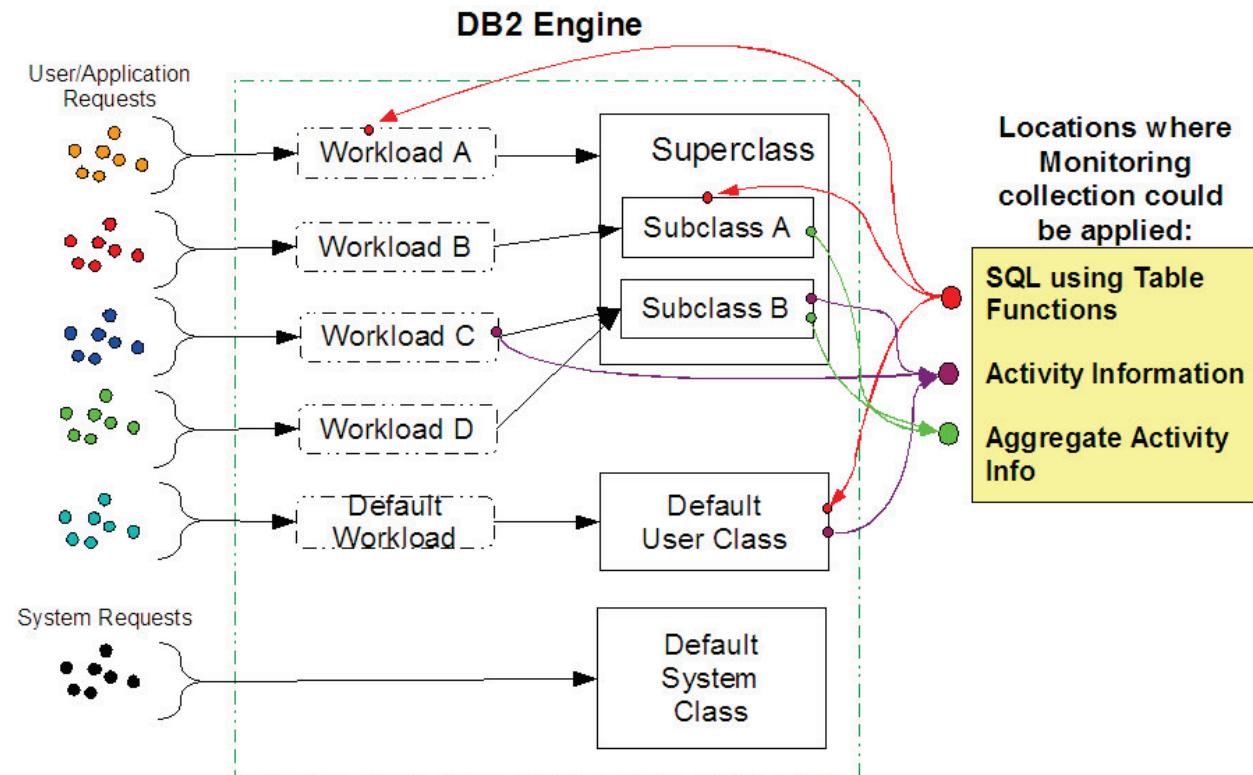
## Agenda

- **High Availability and Disaster Recovery – HADR**
- **DB2 pureScale**
- **DB2 Partitioning**
- **DB2 Storage Optimization**
- **DB2 LBAC**
- **DB2 pureXML**
- **DB2 Workload Management**

## Highlights of Workload Management in DB2

- **A stable, predictable execution environment**
- **A light-weight, granular way to monitor active work**
- **Better resource management – Enhanced in DB2 9.7**
  - Able to explicitly allocate resources amongst work
  - Able to limit excessive, unexpected resource consumption
- **Better request management – Enhanced in DB2 9.7**
  - Able to manage work based on its business priority
  - Able to track performance of work
- **Optim Performance Manager - control all aspects of Workload Management through a single tool integrated into Data Studio and leverage advanced analysis in Performance Expert**

## DB2 WLM Overview



53

© 2011 IBM Corporation

Information Management Technology Ecosystem  
IBM Toronto Lab



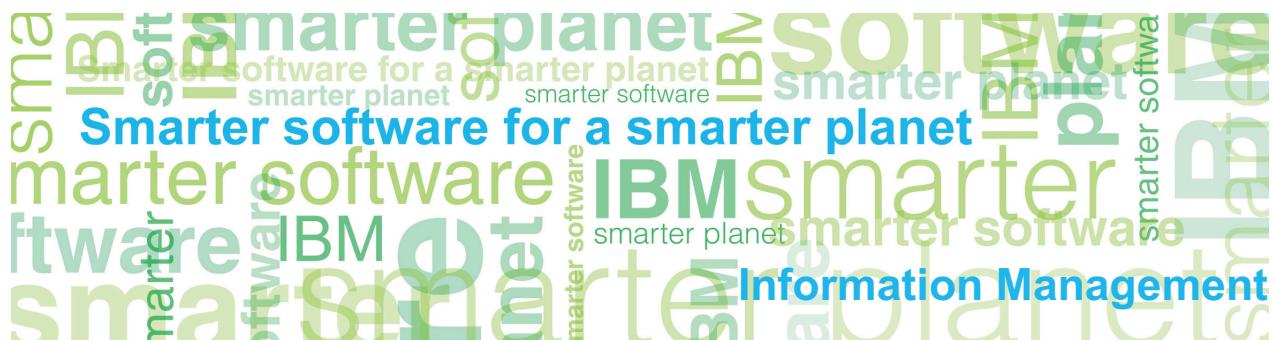
Questions?

Winter/Spring 2011

E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)  
 Subject: “DB2 Workshop for Oracle Professionals”



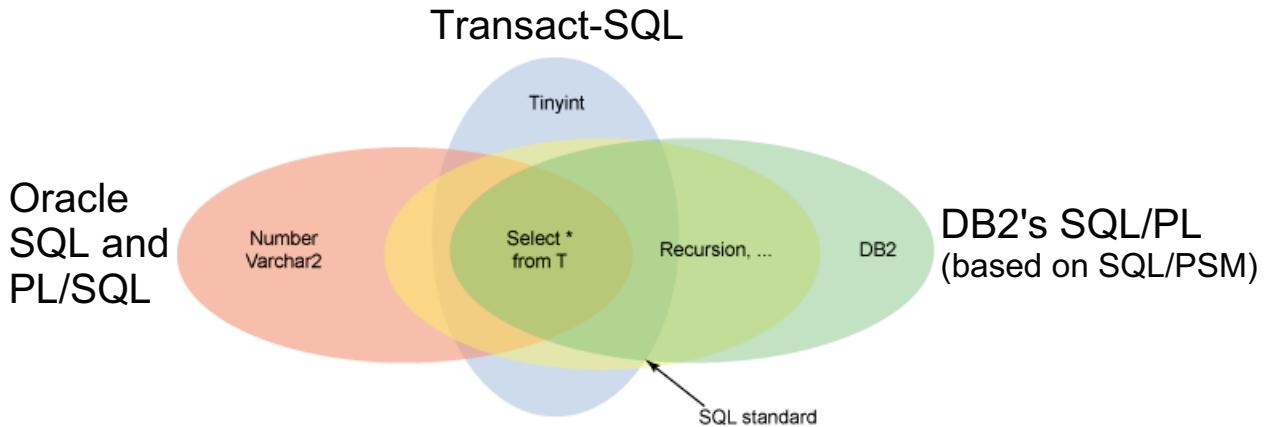
# Breaking Free From Oracle® with DB2® 9.7



© 2011 IBM Corporation

Information Management

## Migration Challenges

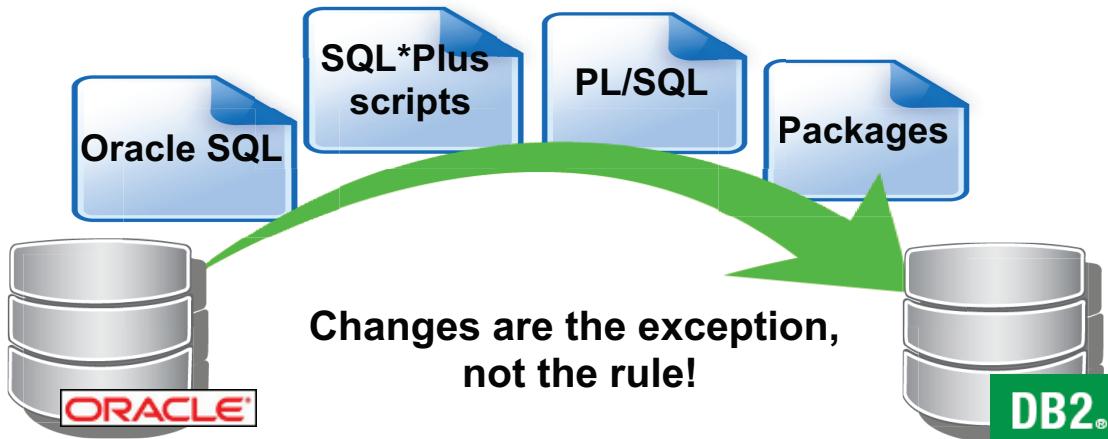


- **Applications are locked to the chosen RDBMS**
  - Moving to a different RDBMS requires extensive rewrite of code
  - Your team's skill set cannot be leveraged
- **DB2 9.7 for LUW introduces a revolutionary approach to the way you can port your application**

## Easy Migration with DB2 9.7

### Oracle DB Compatibility Features in DB2 9.7

- Out-of-the-box support for Oracle's SQL and PL/SQL dialects
- Improved concurrency control
- Extended data type support
- ... and more



3

© 2011 IBM Corporation

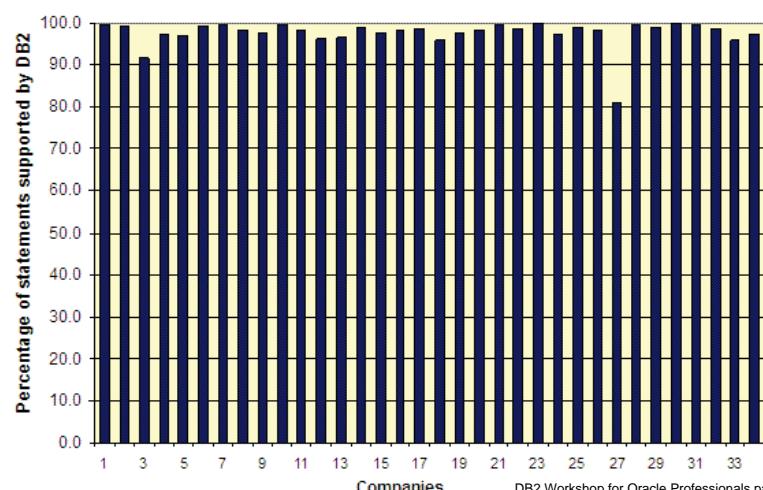
## Breadth of Oracle DB Compatibility Features in DB2 9.7

In average, out-of-the-box compatibility ratio of  
**90% - 99%**

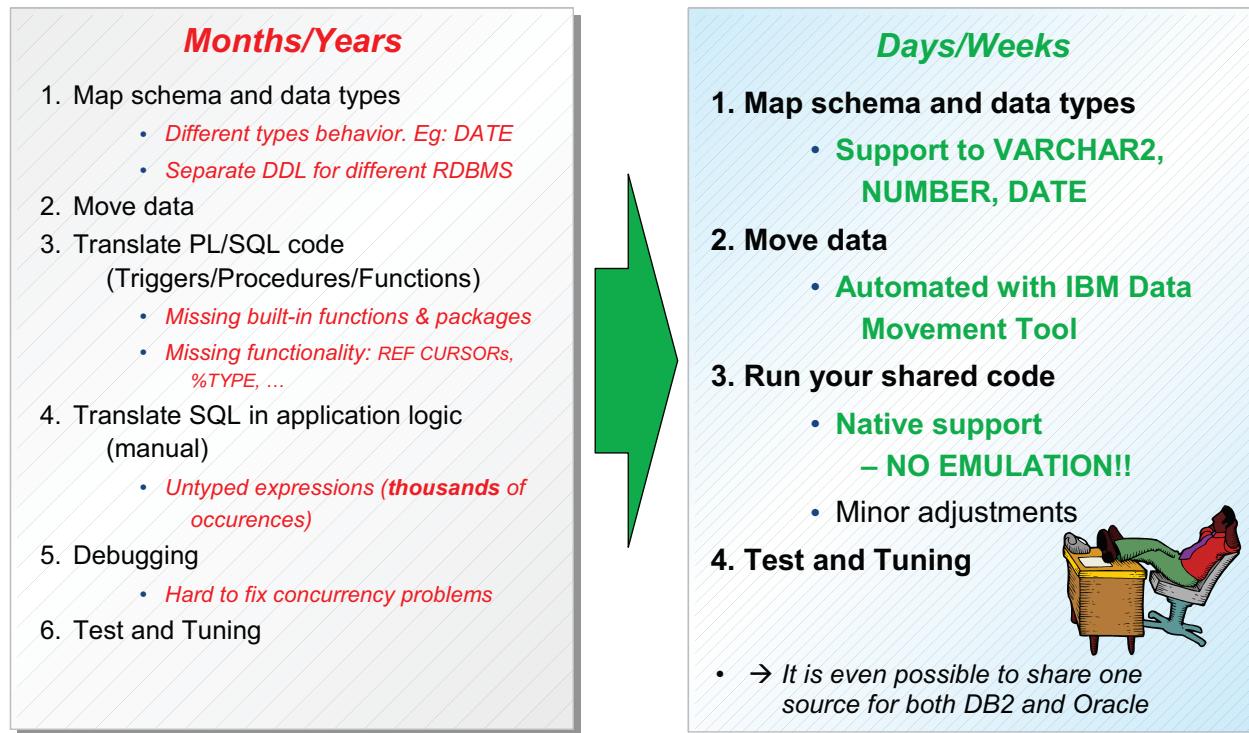
- Not based on a specific Oracle Database version
  - Driving factor was to add support to the most used features in Oracle applications

- Result of a work with a wide range of apps. from partners & clients

- Big (185k statements)
- Small (2k statements)
- Different industries
- Different solutions
- Different countries



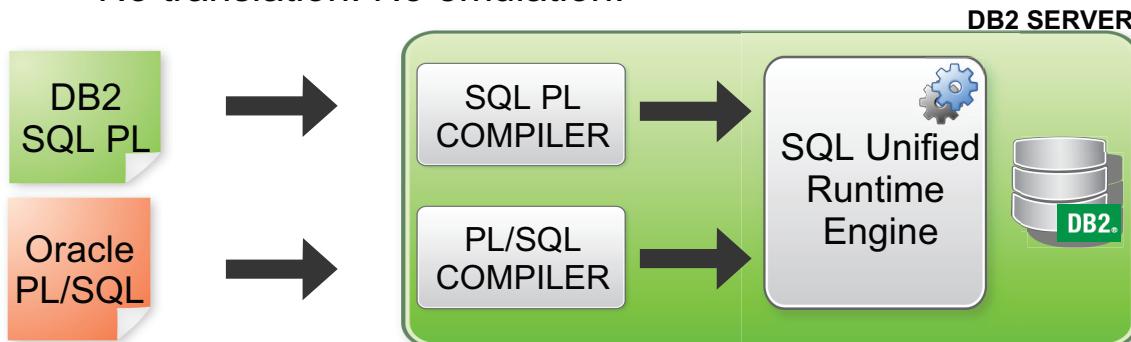
## Migration Roadmap - Before and After DB2 9.7



## Support to Oracle SQL Dialect and PL/SQL

- Native PL/SQL support**

- The DB2 engine now includes a PL/SQL compiler
- No translation. No emulation.

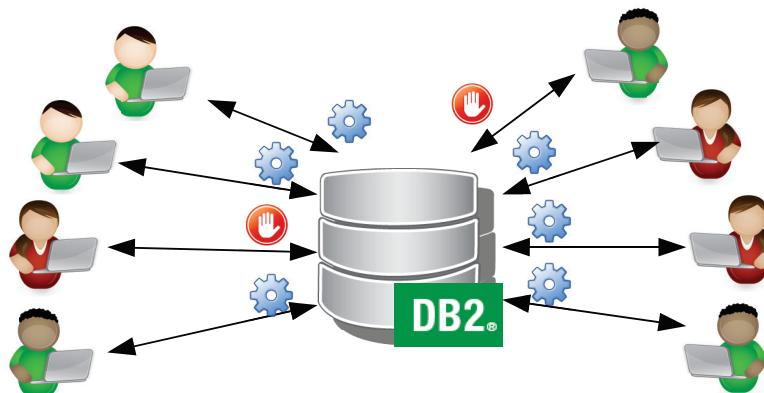


- Several benefits**

- Developers can continue working in the language they are familiar with
- One source code against both Oracle and DB2
- Both PL/SQL and SQL PL perform at the same speed

## Enhanced Concurrency Control

- Some applications used to need tweaking because of differences in the concurrency control between Oracle and DB2
  - “Readers don't block writers and writers don't block readers”
- New isolation level named **Currently Committed**
  - Eliminate conflicts between Read workloads and Write workloads
  - Optimize throughput for mixed workloads
  - Log-based control → **No management overhead**

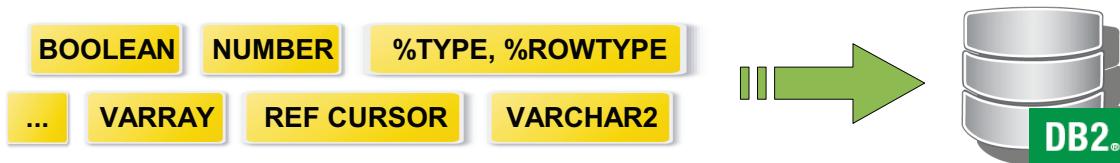


© 2011 IBM Corporation

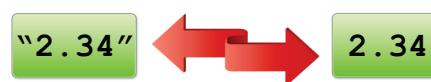
7

## Extended and More Flexible Data Type System

- New data types add support to non-standard Oracle types
  - Basic and complex PL/SQL types



- **Implicit Casting and Type Resolution**
  - Strings and numbers can be compared, assigned, and operated on in a very flexible fashion
  - Untyped NULLs and parameter markers can be used in many more places
    - DB2 defers the type resolution until a value is assigned



DB2 Workshop for Oracle Professionals page 121 of 207.

© 2011 IBM Corporation

8

## Soft Invalidation

- Allows you to replace or drop an object even if other running transactions are using it
  - Simplifies DBA maintenance work
  - CREATE OR REPLACE** and **DROP**: alias, function, view, trigger

### hard validation

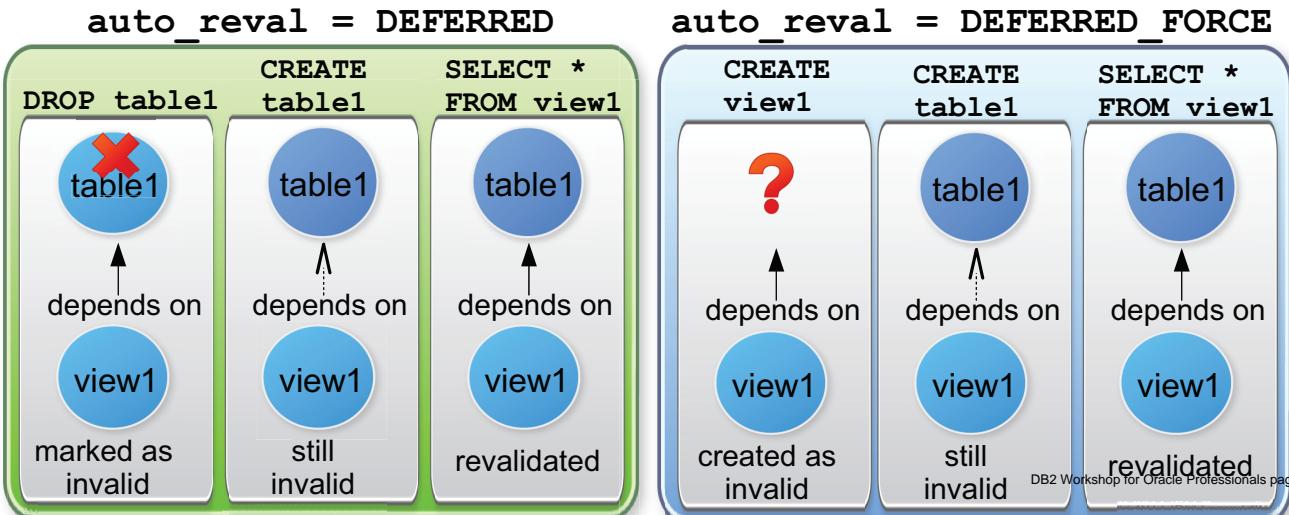


### soft validation



## Automatic Revalidation

- It is a mechanism whereby database objects that have been invalidated undergo revalidation automatically.
  - By default, validation is DEFERRED until the next time an invalid object is used
  - Configured by DB configuration parameter AUTO\_REVAL
    - DEFERRED\_FORCE**: same as DEFERRED, but additional CREATE with error feature is enabled



## Other Features

- **Oracle-specific JDBC extensions**
  - DB2 9.7 JDBC driver provides, among other things, support for calling procedures with reference cursor and VARRAY parameters
- **Support for OCI applications**
  - DB2CI supports 150+ OCI functions
- **SQL\*Plus scripts support using CLPPlus**
  - CLPPlus: a new command line interface for execution of SQL statements and database commands.
- **Moving away from Oracle Forms**
  - Partnership between IBM & RealEase
  - Automated conversion of Oracle Forms to Java



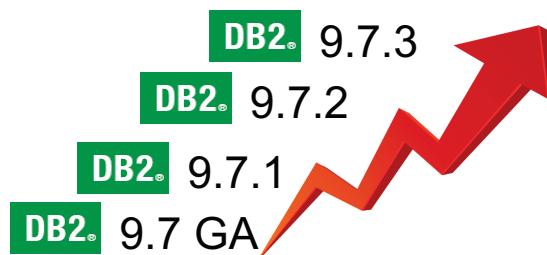
<http://realeasellc.com>

11

© 2011 IBM Corporation

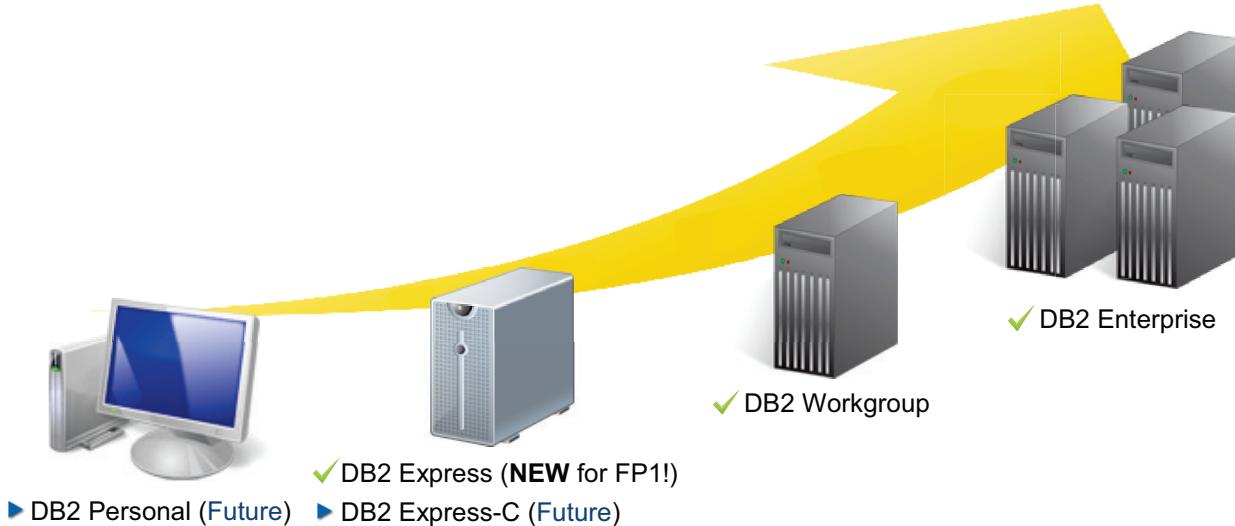
## Continuous Effort

- **DB2 Fix Packs include new feature requests and refinements that continue to improve the level of compatibility.**
- **Some examples from Fix Pack 1:**
  - The OCI application compatible DB2CI client
  - **FORALL** and **BULK COLLECT** support
  - The **SUBSTRB** function and refinements to **SUBSTR**
- **Some examples from Fix Pack 2:**
  - The IBM Data Server Provider for .NET has been enhanced
  - Added support to Oracle's NCHAR, NVARCHAR and NCLOB data types
  - Ability to use default parameters and named parameters in UDFs
  - Obfuscation with **DBMS\_DDL.WRAP** and **DBMS\_DDL.CREATE\_WWRAPPED**
- **Fix Pack 3 is expected to provide an even higher level of compatibility.**



DB2 Workshop for Oracle Professionals page 123 of 207.

## Which DB2 9.7 editions natively support DB2 PL/SQL compilation?



## Configuring Compatibility Mode

- **DB2 9.7 Oracle DB Compatibility Features**
  - Registry variable: **DB2\_COMPATIBILITY\_VECTOR**
    - Accepts hexadecimal values (NULL or 00 to FFF)
    - Recommended value for Oracle migration is **ORA** (same as FFF)

Bit Position	Compatibility Feature	Comment
1 (0x01)	ROWNUM	Synonym for ROW_NUMBER() OVER()
2 (0x02)	DUAL	Enables the <b>DUAL</b> dummy table
3 (0x04)	(+) Outer join operator	
4 (0x08)	Hierarchical queries	Using <b>CONNECT BY</b> clause
5 (0x10)	NUMBER *	
6 (0x20)	DATE *	Enables DATE as TIMESTAMP(0)
7 (0x40)	VARCHAR2 *	
8 (0x80)	TRUNCATE TABLE	
9 (0x100)	Character literals	Character and Graphic string constants whose byte length is less than 255 are assigned the data types CHAR and GRAPHIC (instead of VARCHAR and VARGRAPHIC) respectively.
10 (0x200)	Collection methods	Enables the use of methods to perform operations on arrays, such as <b>first</b> , <b>last</b> , <b>next</b> , and <b>previous</b> . Enables the use of parentheses in place of square brackets in references to specific elements in an array; Eg:array1(i) refers to element i of array1.
11 (0x400)	Data dictionary-compatible views *	Data dictionary-compatible views are created when the database is created
12 (0x800)	PL/SQL Compilation	Enables compilation and execution of PL/SQL statements

## Creating a DB2 Database with Oracle Compatibility

- Create target DB2 database enabling all compatibility features

```
db2set DB2_COMPATIBILITY_VECTOR=ORA
db2stop force
db2start
db2 "CREATE DATABASE dbName AUTOMATIC STORAGE YES
      ON <storagePath1> PAGESIZE 32 K"
-- Recommended but not required
db2 UPDATE DB CFG FOR dbName
      USING AUTO_REVAL deferred_force
      DECFLT_ROUNDING round_half_up
```

**Enables all Oracle DB compatibility features**

**Recommended to accommodate large row length**

**Adjust rounding behavior to match that of Oracle**

**If you want to deploy objects out of dependency order**

## Checking Enabled Compatibility Features

- Database Configuration Parameters

```
db2 get database cfg
```

```
C:\>db2 connect to byeora
      Database Connection Information
      Database server      = DB2/NT 9.7.0
      SQL authorization ID = AMARANHA
      Local database alias = BYEORA

C:\>db2 get database cfg
      Database Configuration for Database
      Database configuration release level          = 0x0d00
      Database release level                      = 0x0d00
      Database territory                         = US
      Database code page                        = 1208
      Database code set                          = UTF-8
      Database country/region code             = 1
      Database collating sequence            = SYSTEM_1252
      Alternate collating sequence           = <ALT_COLLATE>
      Number compatibility                   = ON
      Varchar2 compatibility                 = ON
      Date compatibility                     = ON
      Database page size                    = 4096
      Dynamic SQL Query management        = DISABLE
      Statement concentrator              = OFF
      Discovery support for this database = ENABLE
      Restrict access                      = NO
```

## Summary of PL/SQL Support in DB2 9.7

- You can compile:

- |                                  |                                    |
|----------------------------------|------------------------------------|
| • PL/SQL <b>procedures</b>       | - CREATE [OR REPLACE] PROCEDURE    |
| • PL/SQL <b>functions</b>        | - CREATE [OR REPLACE] FUNCTION     |
| • PL/SQL <b>triggers</b>         | - CREATE [OR REPLACE] TRIGGER      |
| • PL/SQL <b>anonymous blocks</b> | - BEGIN ... END                    |
| • PL/SQL <b>packages</b>         | - CREATE [OR REPLACE] PACKAGE      |
| • PL/SQL <b>package bodies</b>   | - CREATE [OR REPLACE] PACKAGE BODY |

- Features:

- Data types: NUMBER, VARCHAR2, NCHAR, NCLOB, NVARCHAR, DATE, %TYPE, %ROWTYPE, ...
- Basics: NULL, assignments, IF, CASE, EXIT, LOOP, ...
- System defined functions: TO\_CHAR, TO\_NUMBER, TO\_NCLOB, TO\_NCHAR, LPAD, RPAD, ...
- Packages
- REF CURSORS
- SQL dialects: TRUNCATE TABLE, ROWNUM, DUAL, CONNECT BY, ...
- Implicit casting (aka Weak Typing)
- Data Dictionary-compatible views
- And more!

## PL/SQL Code in DB2

```

CREATE PACKAGE sample IS
    TYPE emps_array IS VARRAY(30) OF VARCHAR2(6);

    PROCEDURE get_dept_emp (
        dno          IN      emp.deptno%TYPE DEFAULT 1,
        emps_dno     OUT     emps_array );
END sample;
/
CREATE PACKAGE BODY sample IS
    PROCEDURE get_dept_emp (
        dno          IN      emp.deptno%TYPE,
        emps_dno     OUT     emps_array ) IS
    BEGIN
        IF dno IS NOT NULL THEN
            SELECT empno BULK COLLECT INTO emps_dno FROM emp WHERE deptno=dno;
            FOR x in emps_dno.FIRST .. emps_dno.LAST LOOP
                DBMS_OUTPUT.PUT_LINE(emps_dno(x));
            END LOOP;
        END IF;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('SQLCODE: ' || SQLCODE);
    END get_dept_emp;
END sample;
/

CREATE TYPE emp_dept
    IS TABLE OF emp%ROWTYPE;

```

## PL/SQL Code in DB2

```

CREATE PROCEDURE add_dept (dept_no IN emp_dept) IS
BEGIN
    FORALL deptno _____
        IN dept_no.FIRST .. dept_no.LAST
        INSERT INTO emp_dept
            VALUES dept_no(deptno);
    ...
END;
...
CREATE PROCEDURE emp_info
IS
    CURSOR empcur IS SELECT ename, deptno FROM emp;
    myvar empcur%ROWTYPE;
BEGIN
    OPEN empcur;
    LOOP
        EXIT WHEN empcur%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE( myvar.ename || ' in department ' || myvar.deptno );
    END LOOP;
    CLOSE empcur;
END;
...
CREATE PROCEDURE emp_info WRAPPED SQL09072
aBcDefg12AbcasHGJG6JKHhgkjFGHHkkkljk878979HJHui99

```

**FORALL now supported  
Added in FP1**

**%ROWTYPE**

**%FOUND and %NOTFOUND**

**Built-in Packages**

**Obfuscation with DBMS\_DDL.WRAP  
New in FP2**

## Running PL/SQL Code in DB2

- You can use any of the DB2 interfaces to run PL/SQL code
  - CLPPlus and programming interfaces (JDBC, .NET, etc)
  - DB2 automatically detects the syntax in use

The screenshot shows the CLP Plus application window. It displays the following session:

```

C:\Temp>clppplus db2admin@localhost:50000/byeora
Enter password:
Database Connection Information

Hostname = localhost
Database server = DB2/NT SQL09070
SQL authorization ID = db2admin
Local database alias = BYEORA
Port = 50000

CLPPlus: Version 1.0
Copyright (c) 2009, IBM CORPORATION. All rights reserved.

SQL> @ c:\temp\sample_pkg.sql
DB25000I: The command completed successfully.

DB25000I: The command completed successfully.

SQL> set serveroutput on;
SQL> call sample.get_dept_emp(1, ?);
Value of output parameters
EMPS_DNO = com.ibm.db2.jcc.am.q@b800b80
123
456
DB25000I: The command completed successfully.

SQL>

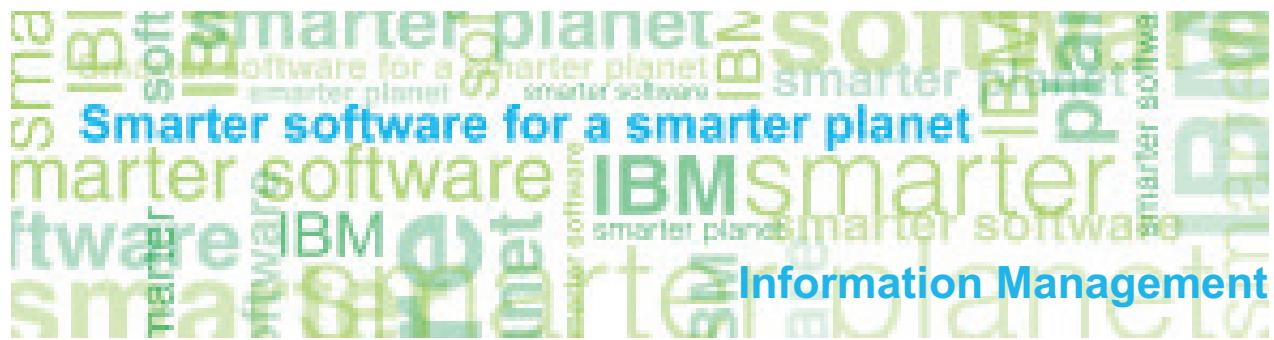
```

## Questions?

Winter/Spring 2011

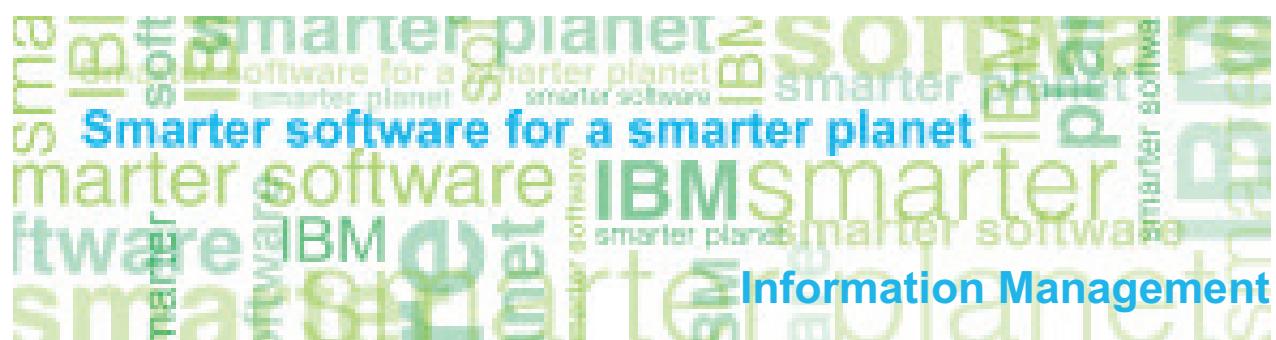
E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)

Subject: "DB2 Workshop for Oracle Professionals"



© 2011 IBM Corporation

# Data Concurrency



© 2011 IBM Corporation



## Agenda

- **Database Transactions**
- **Concurrency**
- **Concurrency Issues**
- **Concurrency Control**
  - Isolation Levels
  - Locking
  - Specifying Isolation Levels

## Database Transactions

- **Transaction**

- sequence of one or more SQL operations, grouped together as a single unit
- also known as a **unit of work**

- **Committed Data**

- using the **COMMIT** statement commits any changes made during the **transaction to the database**

- **Uncommitted Data**

- changes during the transaction before the **COMMIT** statement is executed

## Concurrency

- **Concurrency**

- Sharing of resources by multiple interactive users or application programs at the **same time**

- **Having multiple interactive users can lead to:**

- Lost Update
- Uncommitted Read
- Non-repeatable Read
- Phantom Read

- **Need to be able to control the degree of concurrency:**

- With proper amount of **data stability**
- Without loss of **performance**

## Concurrency Issues

### ▪ Lost Update

–Occurs when two transactions read and then attempt to update the **same data**, the **second update will overwrite the first update before it is committed**

- 1) Two applications, A and B, both read the same row and calculate new values for one of the columns based on the data that these applications read
- 2) A updates the row
- 3) Then B also updates the row
- 4) A's update lost

## Concurrency Issues

### ▪ Uncommitted Read

–Occurs when **uncommitted data is read** during a transaction  
–Also known as a Dirty Read

- 1) Application A updates a value
- 2) Application B reads that value before it is committed
- 3) A backs out of that update
- 4) Calculations performed by B are based on the uncommitted data

## Concurrency Issues

- **Non-repeatable Read**

–Occurs when a transaction **reads the same row of data twice** and returns **different data values** with each read

- 1) Application A reads a row before processing other requests
- 2) Application B modifies or deletes the row and commits the change
- 3) A attempts to read the original row again
- 4) A sees the modified row or discovers that the original row has been deleted

7

© 2011 IBM Corporation

## Concurrency Issues

- **Phantom Read**

–Occurs when a **search based on some criterion** returns **additional rows** after **consecutive searches** during a transaction

- 1) Application A executes a query that reads a set of rows based on some search criterion
- 2) Application B inserts new data that would satisfy application A's query
- 3) Application A executes its query again, within the same unit of work, and some additional phantom values are returned

## Concurrency Control

- **Isolation Levels**

- determine how data is locked or isolated from other concurrently executing processes while the data is being accessed
- are in effect while the transaction is in progress

- **There are four levels of isolation in DB2:**

- Repeatable read
- Read stability
- Cursor stability
  - Currently Committed
- Uncommitted read

## Locking

- **Isolation levels are enforced by locks**

- locks limit or prevent data access by concurrent users or application processes

- **Locking Attributes**

- resource being locked is called object
- objects which can be explicitly locked are databases, tables and table spaces
- objects which can be implicitly locked are rows, index keys, and tables
- implicit locks are acquired by DB2 according to isolation level and processing situations
- object being locked represents granularity of lock
- length of time a lock is held is called duration and is affected by isolation level

## Types of Locks

- **Share (S)**
  - concurrent transactions are limited to **read-only** operations
- **Update (U)**
  - concurrent transactions are limited to **read-only** operations
  - if the transactions have **not declared** that they might **update** a row, the database manager **assumes** that transaction currently looking at a row **might update** it
- **Exclusive (X)**
  - concurrent transactions are **prevented** from accessing the data in any way
  - does **not apply** to transactions with an isolation level of **UR**
- **Database manager places exclusive locks on every row that is inserted, updated, or deleted**

## Deadlock

- **Deadlock Detector**
  - discovers **deadlock cycles**
    - **randomly** selects one of the transactions involved to roll back and **terminate**
    - transaction chosen is then sent an SQL error code, and every **lock** it had acquired is released



## Repeatable Read

- **Highest level of isolation**
  - No dirty reads, non-repeatable reads or phantom reads
- **Locks the entire table or view being scanned for a query**
  - Provides **minimum concurrency**
- **When to use Repeatable Read:**
  - Changes** to the result set are **unacceptable**
  - Data stability is more important than performance

## Read Stability

- **Similar to Repeatable Read but not as strict**
  - No dirty reads or non-repeatable reads
  - Phantom reads can occur
- **Locks only the retrieved or modified rows in a table or view**
- **When to use Read Stability:**
  - Application needs to operate in a **concurrent environment**
  - Qualifying rows must remain stable for the duration of the unit of work
  - Only issue unique queries** during a unit of work
    - If the same query is issued more than once during a unit of work, the same result set should not be required

## Cursor Stability

- **Default isolation level**
  - No dirty reads
  - Non-repeatable reads and phantom reads can occur
- **Locks only the row currently referenced by the cursor**
- **When to use Cursor Stability:**
  - Want **maximum concurrency** while seeing only **committed** data

## Currently Committed

- **Currently Committed is a variation on Cursor Stability**
  - Avoids timeouts and deadlocks**
  - Log based:
    - No management overhead

## Cursor Stability

Situation	Result
Reader blocks Reader	No
Reader blocks Writer	Maybe
Writer blocks Reader	Yes
Writer blocks Writer	Yes



## Currently Committed

Situation	Result
Reader blocks Reader	No
Reader blocks Writer	No
Writer blocks Reader	No
Writer blocks Writer	Yes

## Currently Committed

- Up to DB2 9.5
  - Cursor Stability is the default isolation level
- Now in DB2 9.7
  - Currently Committed is the default for NEW databases
  - Currently Committed is disabled for upgraded databases, i.e., Cursor Stability semantics are used
- Applications that depend on the old behavior (writers blocking readers) will need to update their logic or disable the Currently Committed semantics

## Example – Cursor Stability Semantics

Transaction A	Transaction B
update T1 set col1 = ? where col2 = 2	
	update T2 set col1 = ? where col2 = ?
select * from T2 where col2 >= ?	
	select * from T1 where col5 = ? and col2 = ?

→ Waiting because is reading uncommitted data      → Waiting because is reading uncommitted data

**DEADLOCK!!**



## Example – Currently Committed Semantics

Transaction A	Transaction B
update <b>T1</b> set col1 = ? where col2 = 2	
	update <b>T2</b> set col1 = ? where col2 = ?
select * from <b>T2</b> where col2 >= ?	
	select * from <b>T1</b> where col5 = ? and col2 = ?
commit	
	commit

→ No locking  
→ Reads last committed version of the data

← No locking  
← Reads last committed version of the data

No deadlocks, no timeouts in this scenario!

19

© 2011 IBM Corporation

# Currently Committed – How to use it?

- `cur_commit` – DB config parameter

- **ON**: default for new DBs created in DB2 9.7 – CC semantics in place
  - **DISABLED**: default value for existing DBs – old CS semantics in place

#### •PRECOMPILE/BIND

- **CONCURRENTACCESSRESOLUTION**: Specifies the concurrent access resolution to use for statements in the package.
    - USE CURRENTLY COMMITTED
    - WAIT FOR OUTCOME

## Uncommitted Read

- **Lowest level of isolation**
  - Dirty reads, non-repeatable reads and phantom reads can occur
- **Locks only rows being modified in a transaction involving DROP or ALTER TABLE**
  - Provides maximum concurrency
- **When to use Uncommitted Read:**
  - Querying **read-only** tables
  - Using only **SELECT** statements
  - Retrieving **uncommitted data** is acceptable
- **Uncommitted Read with Read-Write tables**
  - UR behaves like CS with updateable cursors

## Isolation Levels

- **Summary**

Isolation Level	Dirty Read	Non-repeatable Read	Phantom Read	
Repeatable Read (RR)	-	-	-	
Read Stability (RS)	-	-	Possible	
Cursor Stability (CS)	-	Possible	Possible	← <b>DEFAULT</b>
Uncommitted read (UR)	Possible	Possible	Possible	

Application Type	High data stability required	High data stability not required
Read-write transactions	Read Stability (RS)	Cursor Stability (CS)
Read-only transactions	Repeatable Read (RR) or Read Stability (RS)	Uncommitted Read (UR)

## Specifying Isolation Levels

- **Precompile / Bind**

- ISOLATION option of PREP or BIND command
- Can determine isolation level of a package by executing the following query

```
SELECT ISOLATION FROM syscat.packages
WHERE pkgname = 'pkgname'
AND pkgschema = 'pkgschema'
```

- **Statement Level**

- Use the WITH {RR, RS, CS, UR} clause
- The WITH UR option applies only to read-only operations
  - ensure that a result table is read-only by specifying FOR READ ONLY in the SQL statement
- Overrides the isolation level specified for the package

```
SELECT * FROM tbl WITH RR
```

## Specifying Isolation Levels

- **Dynamic SQL within the current session**

- SET CURRENT ISOLATION
- For all subsequent dynamic SQL statements within the same session

- **JDBC or SQLJ at run time**

- SQLJ profile customizer (db2sqljcustomize command)

- **CLI or ODBC at run time**

- CHANGE ISOLATION LEVEL command specified during the program preparation process

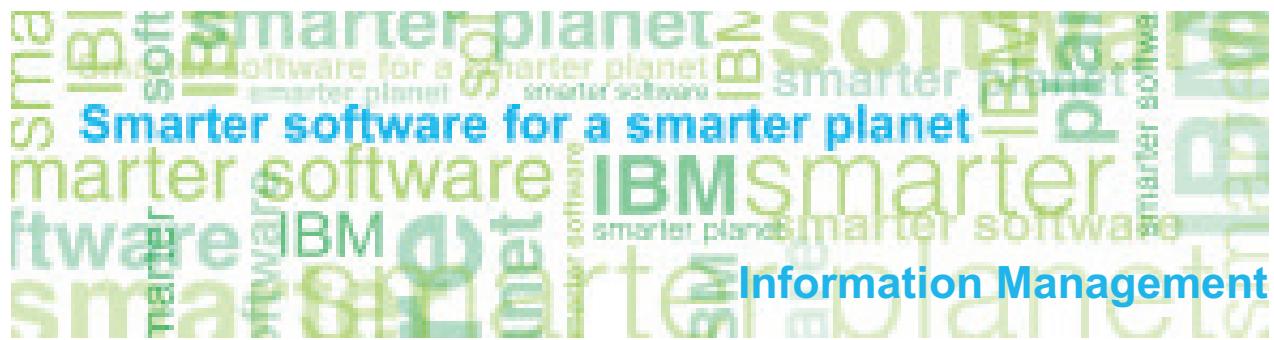
```
CHANGE ISOLATION LEVEL TO RR
```

## Questions?

Winter/Spring 2011

E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)

Subject: "DB2 Workshop for Oracle Professionals"



© 2011 IBM Corporation

# PureXML® Technology Overview



© 2011 IBM Corporation

Information Management



## Agenda

- **Overview**
  - What is XML?
- **pureXML in DB2**
- **XML Data Movement in DB2**
- **SQL/XML and XQuery**
- **XML Schemas in DB2**
- **SQL/XML Publishing Functions**
- **XML Indexes in DB2**
- **Application Development**

## Who Uses XML?

### **Banking**

IFX, OFX, SWIFT, SPARCS,  
MISMO +++

### **Healthcare**

HL7, DICOM, SNOMED,  
LOINC, SCRIPT +++

### **Insurance**

ACORD  
XML for P&C, Life +++

### **Financial Markets**

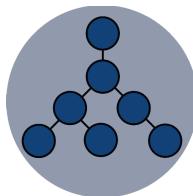
FIX Protocol, FIXML, MDDL,  
RIXML, FpML +++

### **Cross Industry**

PDES/STEPml  
SMPI Standards  
RFID, DOD XML+++

### **Life Sciences**

MIAME, MAGE,  
LSID, HL7, DICOM,  
CDIS, LAB, ADaM +++



### **Retail**

IXRetail, UCCNET, EAN-UCC  
ePC Network +++

### **Electronics**

PIPs, RNIF, Business Directory,  
Open Access Standards +++

### **Telecommunications**

eTOM, NGOSS, etc.  
Parlay Specification +++

### **Energy & Utilities**

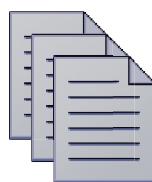
IEC Working Group 14  
Multiple Standards  
CIM, Multispeak

### **Automotive**

ebXML,  
other B2B Stds.

### **Chemical & Petroleum**

Chemical eStandards  
CyberSecurity  
PDX Standard+++

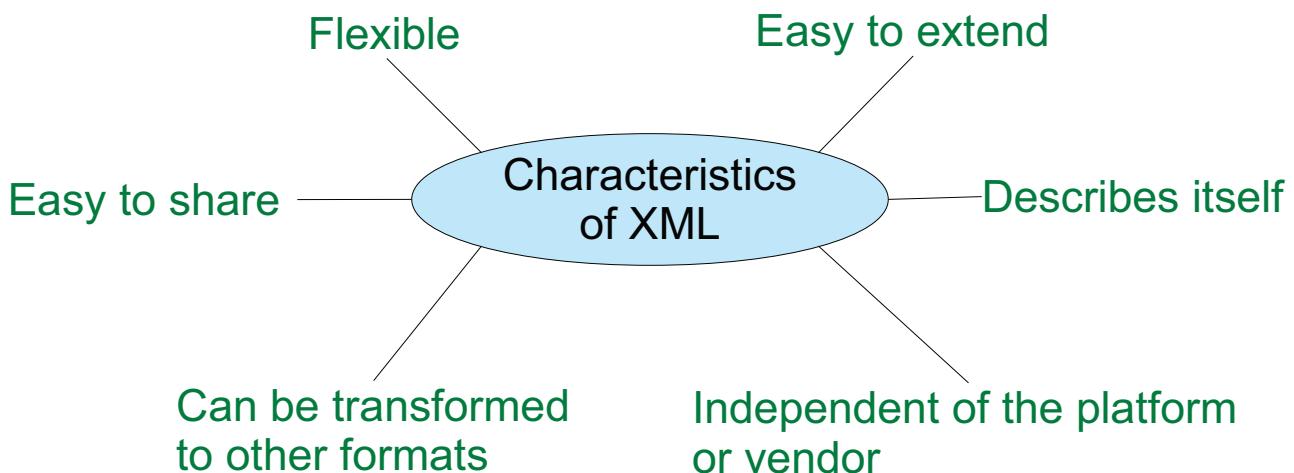


3

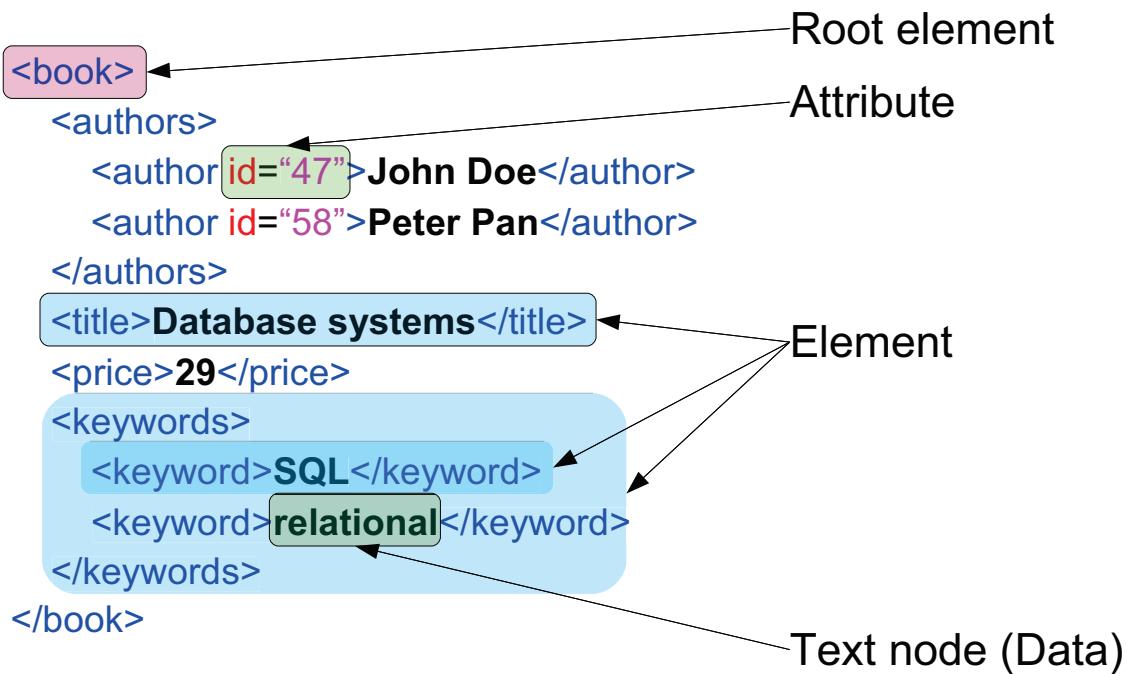
© 2011 IBM Corporation

## What is XML?

- **eXtensible Markup Language**
  - XML is a language designed to describe data
- **A hierarchical data model**



## XML Document Components

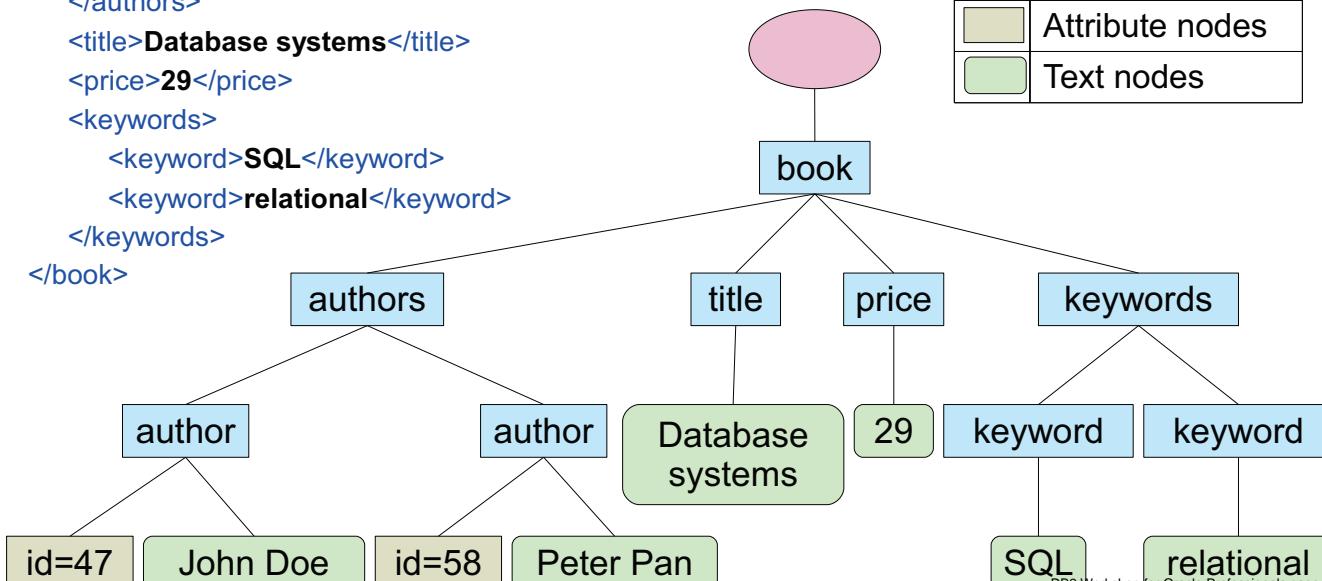


## The XML Data Model: Node Types

```

<book>
  <authors>
    <author id="47">John Doe</author>
    <author id="58">Peter Pan</author>
  </authors>
  <title>Database systems</title>
  <price>29</price>
  <keywords>
    <keyword>SQL</keyword>
    <keyword>relational</keyword>
  </keywords>
</book>
  
```

Node Types	
	Document node
	Element nodes
	Attribute nodes
	Text nodes



## Well-Formed Versus Valid XML Documents

- **A well-formed XML document is a document that follows basic rules:**
  - 1) It must have one and only one root element
  - 2) Each element begins with a start tag and ends with an end tag
  - 3) An element can contain other elements, attributes, or text nodes
  - 4) Attribute values must be enclosed in double quotes. Text nodes, on the other hand, should not.

(i.e. it can be parsed by an XML parser without error)
  
- **A valid XML document is BOTH:**
  - 1) A well-formed XML document
  - 2) A document compliant with the rules defined in an XML schema document or a Document Type Definition (DTD) document.

XML Parsers can optionally perform “validation”

## XML Schema

- **Defines the structure for an XML document**
- **Can define data types for elements/attributes**
  - Basic types: `integer`, `date`, `decimal`, `string`, etc.
  - User Defined Types, Complex Element Types, etc.
  - Allowed length and patterns for string values.
  - Supports type inheritance and derived data types.
- **An XML Schema document is in XML syntax**
  - You can use the same editors and parsers to manipulate them
- **Detailed occurrence and value range definitions**
- **Can define a certain element to be the root element**
- **A Schema can be composed of multiple schema documents**
  - Import/Include of other schemas

## XML Schema Example

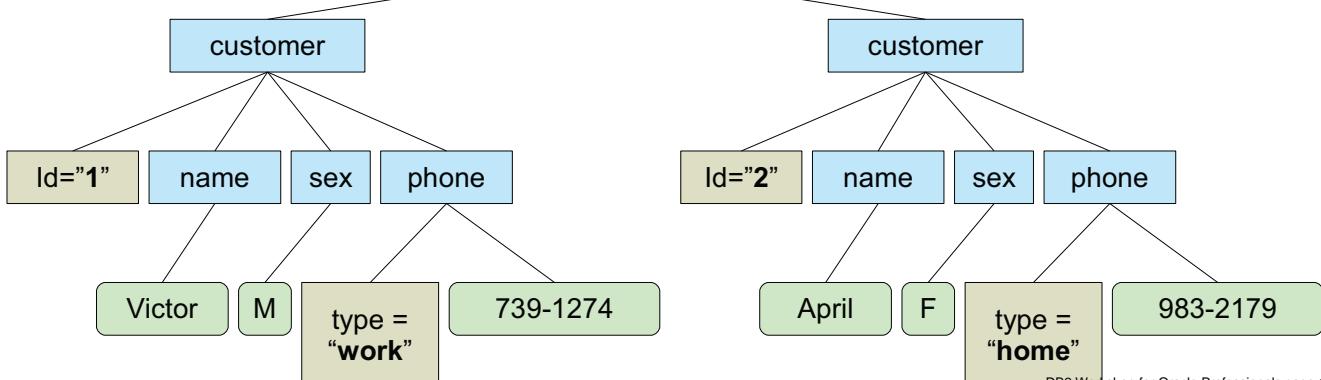
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:simpleType name="PriceType">
    <xsd:restriction base="xsd:decimal">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="100000"/>
      <xsd:totalDigits value="9"/>
      <xsd:fractionDigits value="3"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="StockPriceType">
    <xsd:sequence>
      <xsd:element name="Ask" type="PriceType"/>
      <xsd:element name="Bid" type="PriceType"/>
      <xsd:element name="P50DayAvg" type="PriceType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="StockPrice" type="StockPriceType"/>
</xsd:schema>
```

## XPath

```
<customerInfo>
  <customer id ="1">
    <name>Victor</name>
    <sex>M</sex>
    <phone type="work">739-1274</phone>
  </customer>
  <customer id ="2">
    <name>April</name>
    <sex>F</sex>
    <phone type="home">983-2179</phone>
  </customer>
</customerInfo>
```

Parse  
↓

Path Table
/
/customerInfo
/customerInfo/customer/@id
/customerInfo/customer/name
/customerInfo/customer/sex
/customerInfo/customer/phone
/customerInfo/customer/phone/@type



## Some Common XPath Expressions

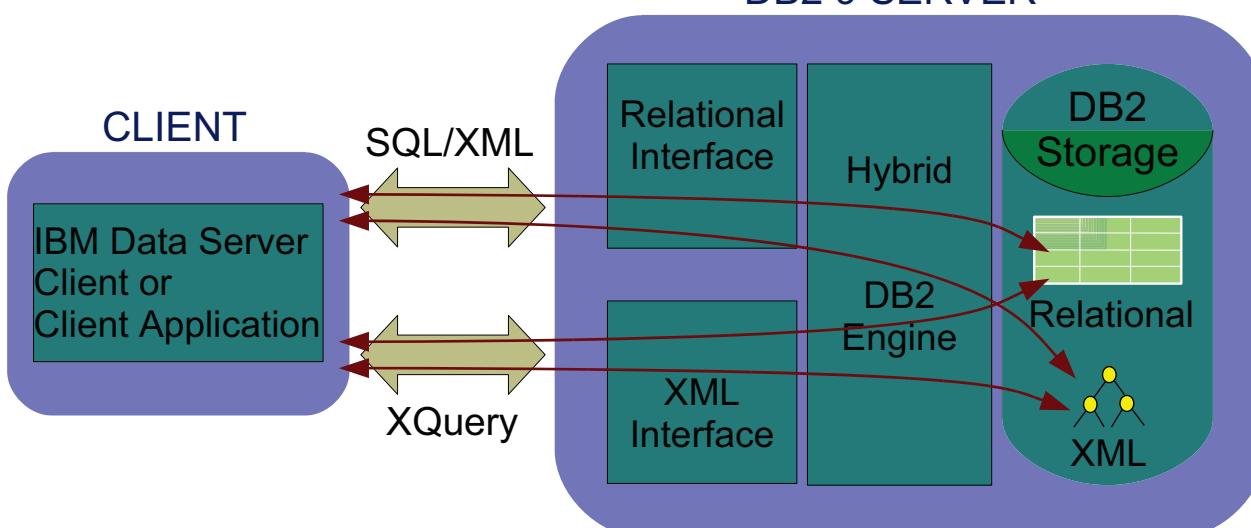
```
<customerInfo>
  <customer id ="1">
    <name>Victor</name>
    <sex>M</sex>
    <phone type="work">739-1274</phone>
  </customer>
  <customer id ="2">
    <name>April</name>
    <sex>F</sex>
    <phone type="home">983-2179</phone>
  </customer>
</customerInfo>
```

/	Selects from the root node.
//	Selects nodes in the document from the current node that match the select.
text()	Specifies the text node under an element.
@	Specifies an attribute.
*	Matches any element node.
@*	Matches any attribute node.
[ ... ]	Predicates

XPath Expression	Result Description	Result
/customerInfo/*/phone/text ()	Selects the text node under the phone element of customerInfo	739-1274 983-2179
/customerInfo//phone/@type	Selects the type attribute under the phone element of customerInfo	work home
/customerInfo/customer[1]/phone/text ()	Selects the phone element text node under the first customer of customerInfo	739-1274
/customerInfo//phone[@type='home ']	Selects all phone elements under cusomterInfo which has an attribute named type with a value of 'home'	<phone type="home"> 983-2179 </phone>

## The pureXML Engine

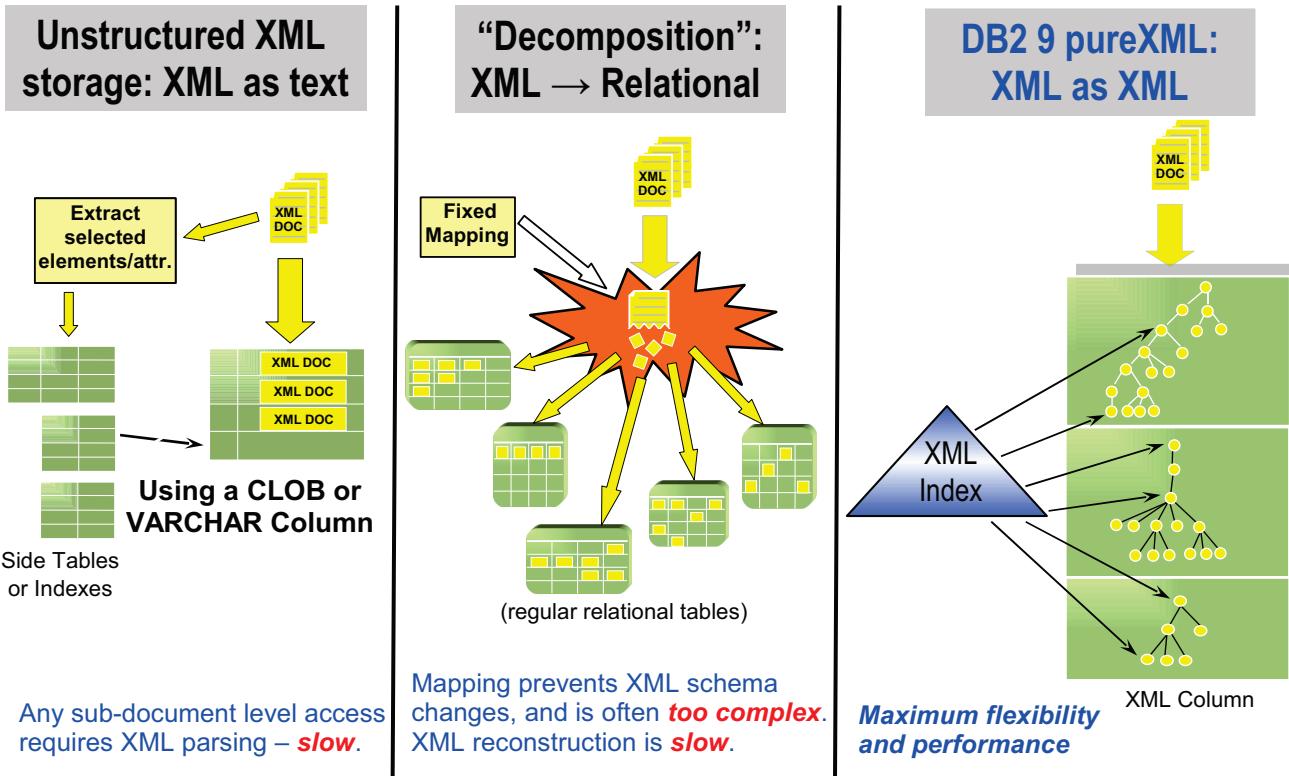
- Relational and XML data are stored differently, but closely linked
- XML Capabilities in all DB2 components
- Combine XML and relational data      DB2 9 SERVER



## Relational Versus Hierarchical (XML) Model

Relational	Hierarchical (XML)
Relational data is <b>flat</b>	XML data is <b>nested</b> .
Relational model is <b>set oriented</b> . Sets are unordered.	XML retrieves <b>sequences</b> (the order <b>matters</b> )
Relational data is <b>structured</b> .	XML data is <b>semi-structured</b> .
Relational data has a <b>strong schema</b> , unlikely to change often.	XML data has a <b>flexible schema</b> , appropriate for constant changes.
Use <b>NULL</b> for an unknown state.	<b>NULLS don't exist</b> . Don't add any XML element.
Based on the <b>ANSI/ISO</b> industry standards.	Based on the <b>W3C</b> industry standards.

## XML Storage: Old (V8) and New (V9)



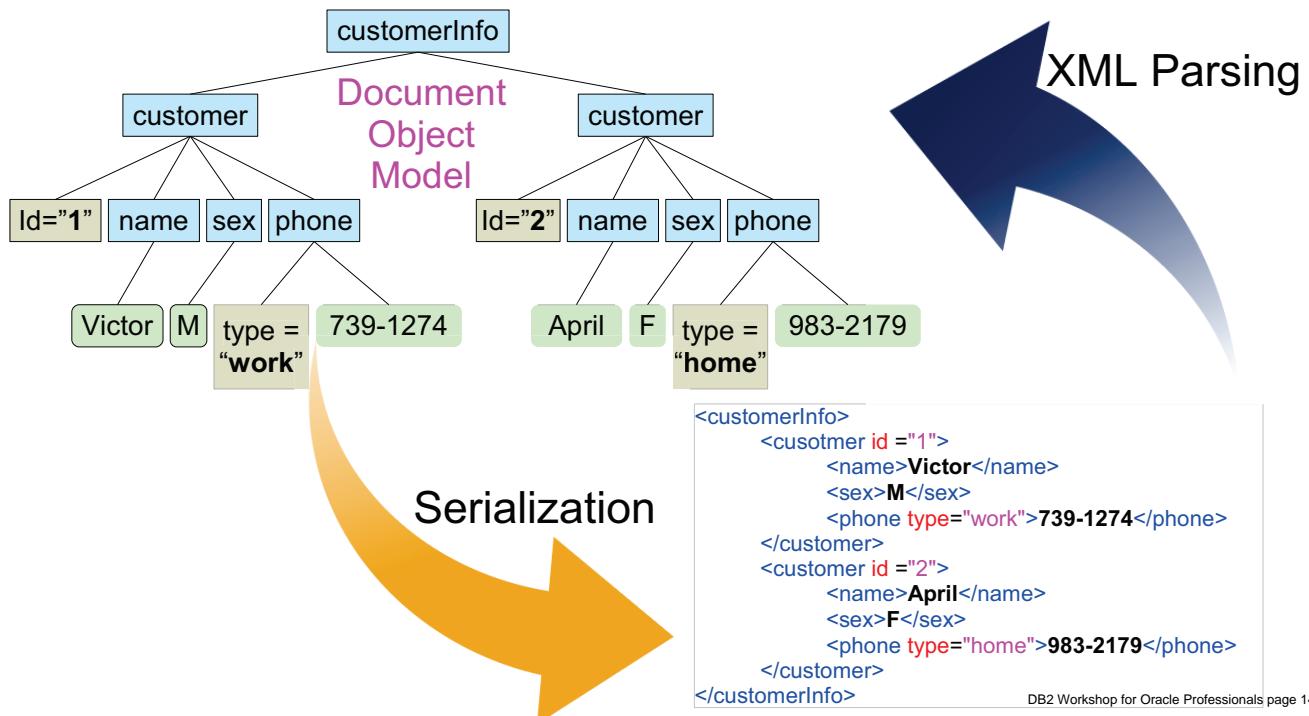
## Storing XML: Native XML Storage

- DB2 stores XML in parsed hierarchical format (~DOM)
- Relational columns are stored in relational format (tables)
- XML columns are stored **natively**
- **No XML parsing for query evaluation!**

- Rows hold relational data
- XML values in pureXML (XDM) storage
- Descriptor pointing from row to XML storage

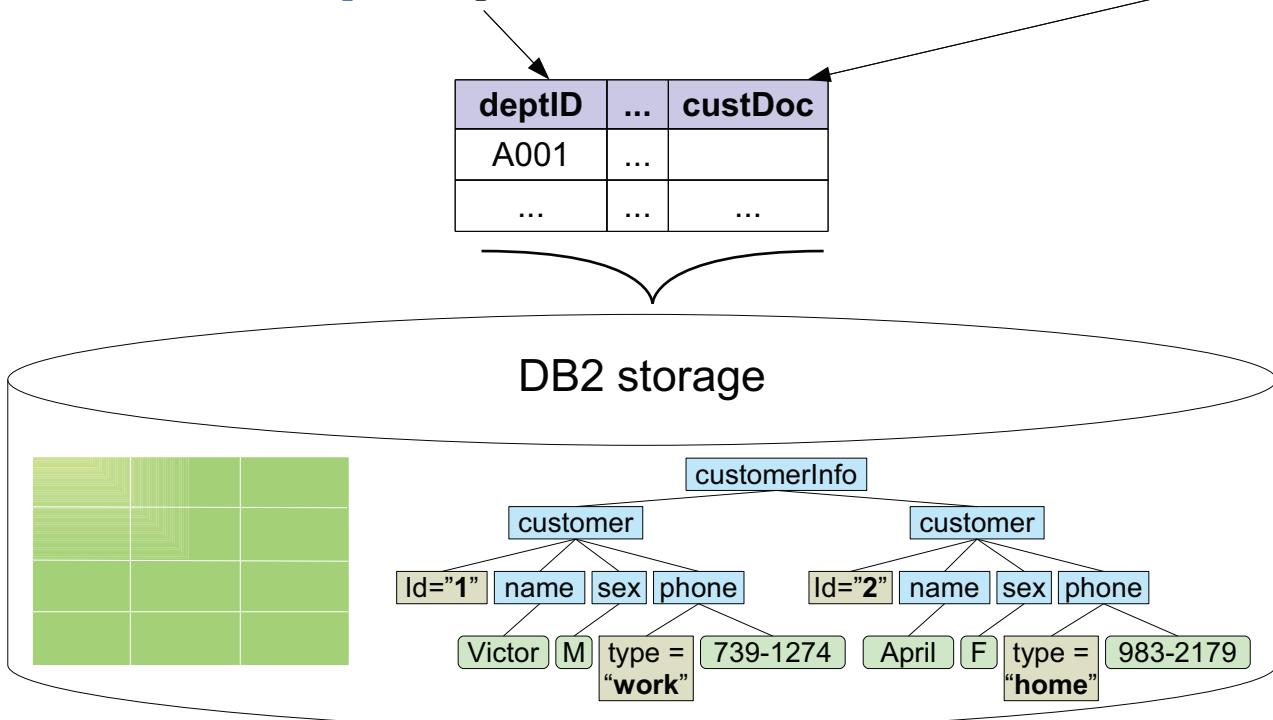
## Storage as XML Document Tree

- Documents are stored in parsed representation

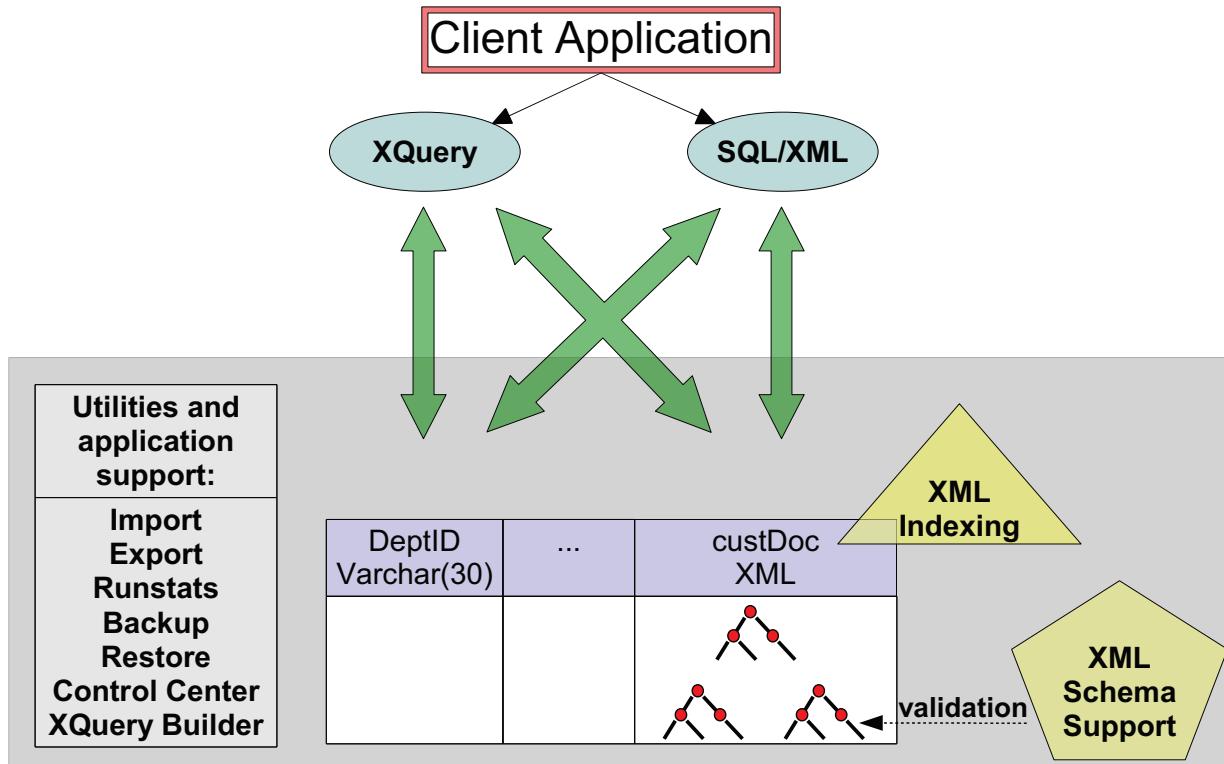


## PureXML Storage in DB2: XML Data Type

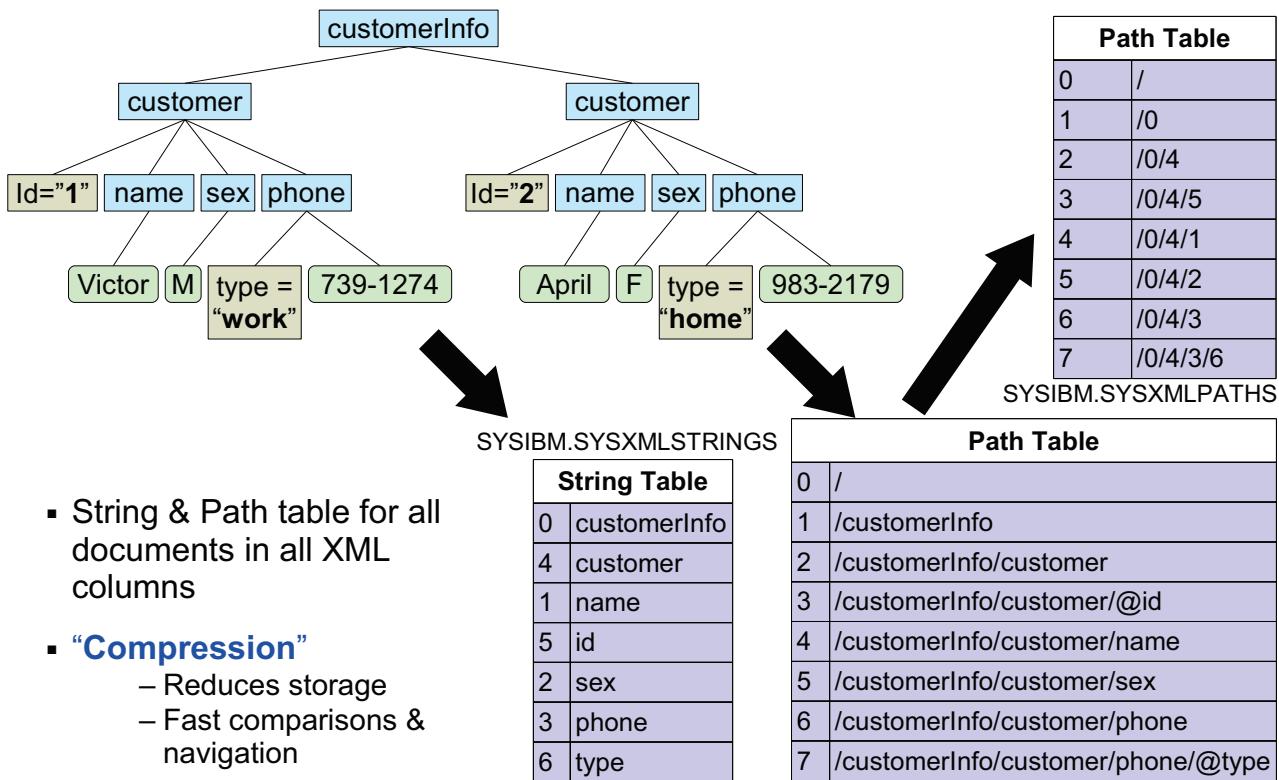
```
CREATE TABLE dept (deptID VARCHAR(30), ..., custDoc XML)
```



## PureXML in DB2: The big picture



## String Table and Paths Table



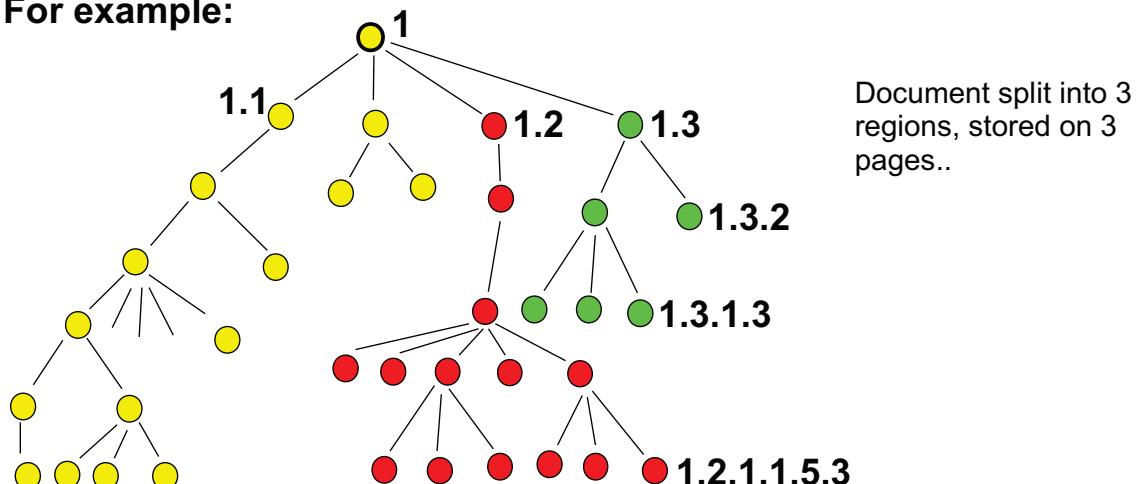
19

© 2011 IBM Corporation

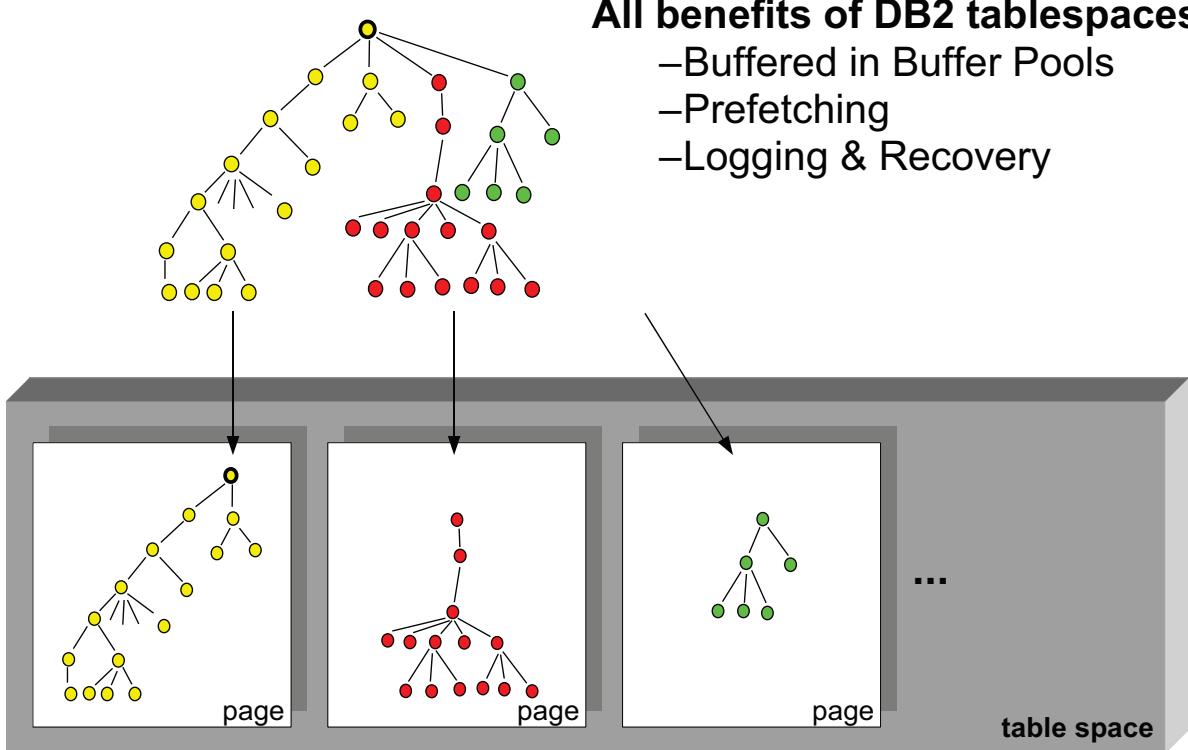
## XML Node Storage Layout

- Node hierarchy of an XML document is stored on DB2 pages**
- Document that don't fit on 1 page are split into **pages/regions****
- No architectural limit for size of XML documents**
- NodeIDs are used to identify individual nodes**

For example:



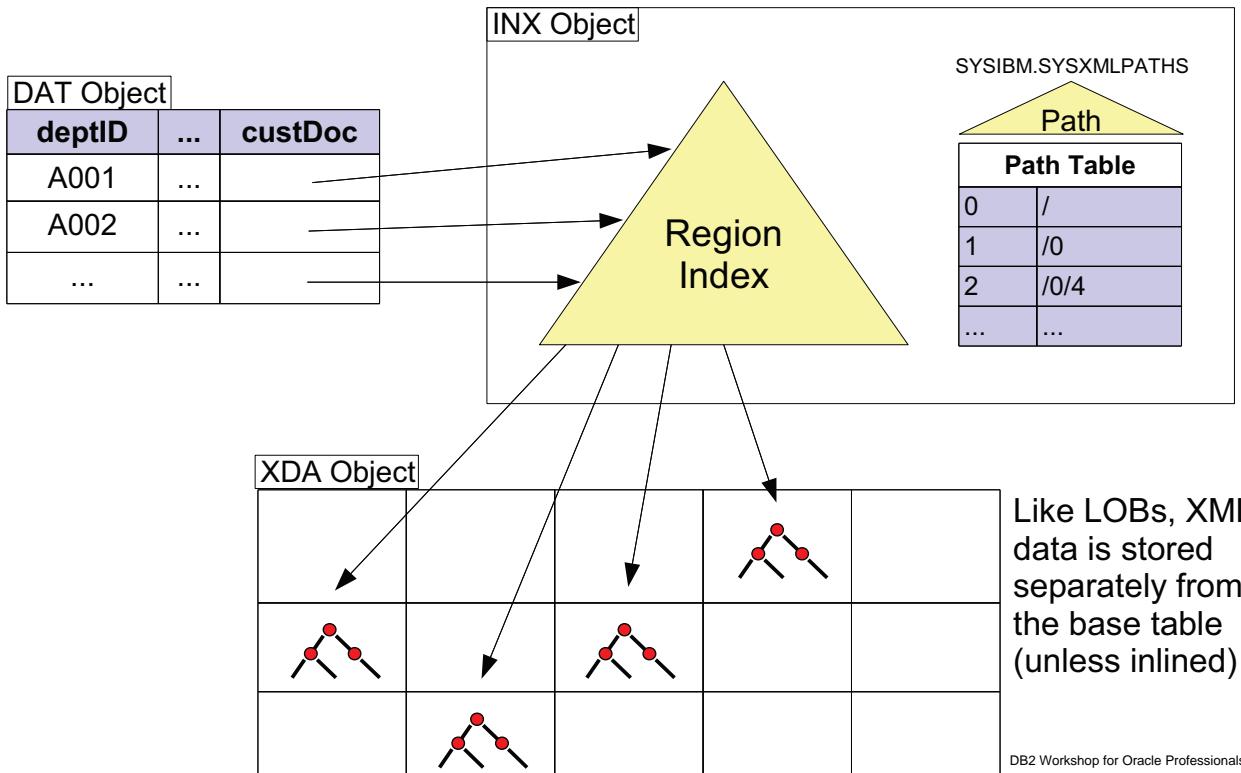
## XML Data: As Trees on DB2 Pages



21

© 2011 IBM Corporation

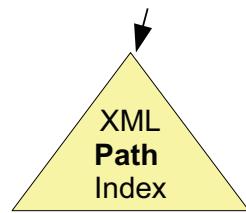
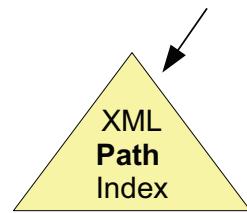
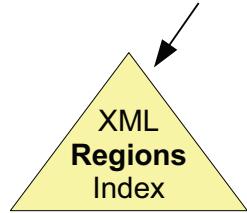
## XML Storage: Internal Objects and Their Relationship



22

## New System Indexes

```
CREATE TABLE myTable (col1 INT, col2 XML, col3 XML)
```



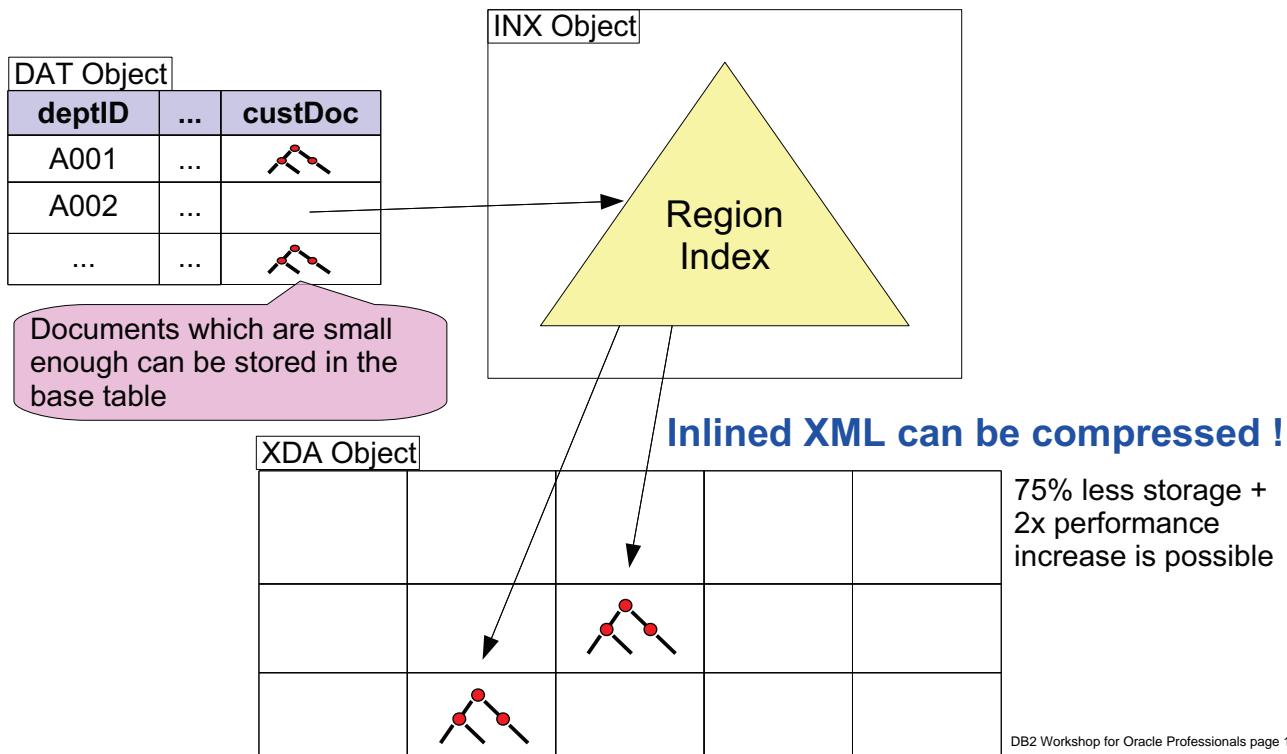
### ▪ XML Region Index

- Created once for table with XML column(s)
- Maps logical pointers to XML data pages

### ▪ XML Path Index

- Created for each XML column
- Holds local subset of global path/pathID mapping information / path table
- Can be used for wildcard resolution

## XML Inlining



## XML Inlining

```
CREATE DATABASE departments PAGESIZE 16 K;

CREATE TABLE dept(deptID VARCHAR(30), ..., custDoc XML inline length 10240);

ALTER TABLE dept compress yes;
```

- Maximum inline length depends on size of page
- Minimum inline length is 292
- Inlining stores documents in the base table object, not in the XDA, if the tree representation fits within the specified inline line length.

Note: inlining increases the base table row length ▶ less rows per page  
 ▶ queries that only read relational columns may suffer.

## Functions Regarding XML Inlining

### SYSCAT.SYSCOLUMNS catalog

- Provides information on the columns of tables
- **INLINE\_LENGTH** column shows
  - Inline length specified for the column of the table
  - 0 if no inline length is specified

TABNAME	COLNAME	INLINE_LENGTH
SAMPLETAB	XMLDOC	3000
T1	XMLDOC	0

### ADMIN\_IS\_INLINED(xmlcol) & ADMIN\_EST\_INLINE\_LENGTH(xmlcol)

```
SELECT id, ADMIN_IS_INLINED(xmlcol) AS inlined,
       ADMIN_EST_INLINE_LENGTH(xmlcol) AS inline_length
FROM customer;
```

ID	INLINED	INLINE_LENGTH
1000	1	770
1001	0	2345
1002	1	796
1003	0	1489
1004	0	1910
1005	0	-1

6 record(s) selected.

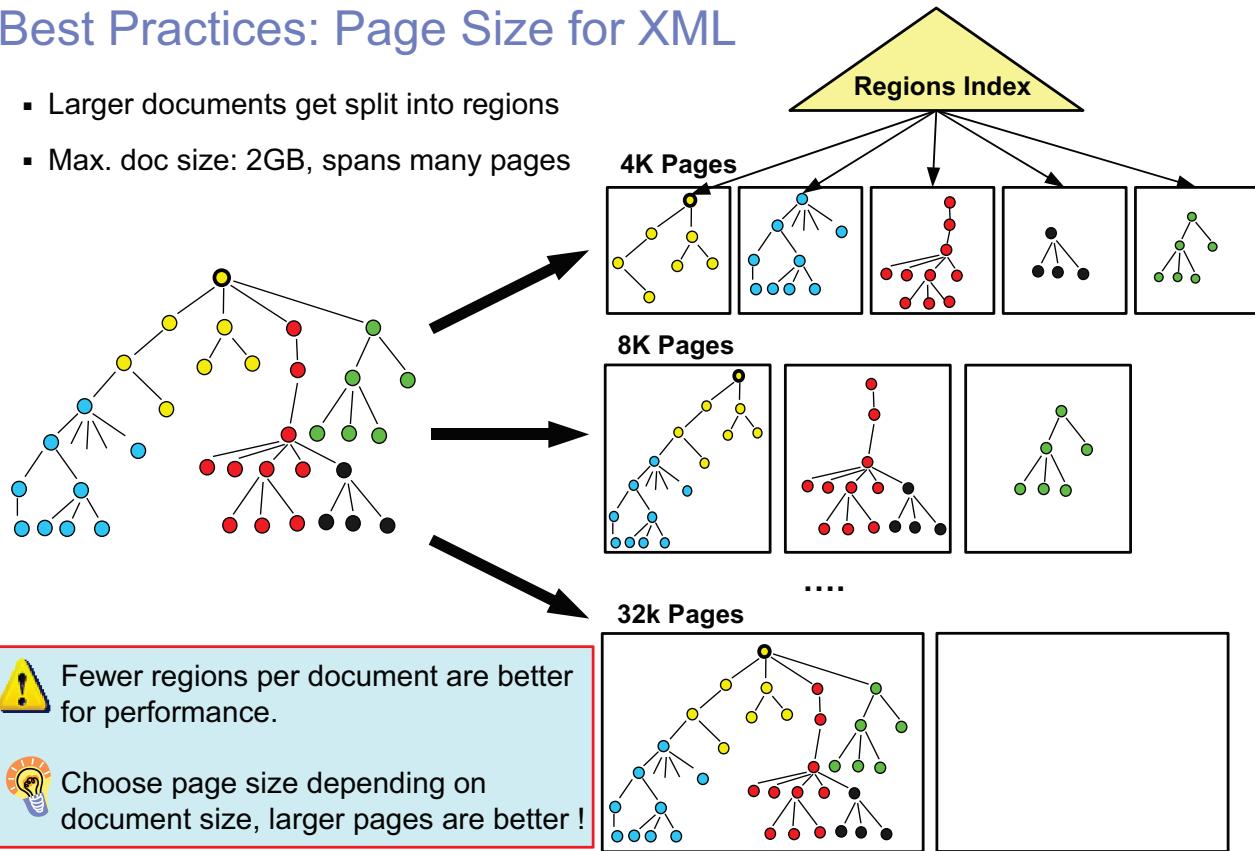
Is inlined, uses 770 bytes.

Not inlined,  
requires inline  
length > 1489

Too large to be in-lined  
for the given page size

## Best Practices: Page Size for XML

- Larger documents get split into regions
- Max. doc size: 2GB, spans many pages



27

© 2011 IBM Corporation

## Best Practice: Deciding Where to Store XML

- Need smaller pages for the relational part of the table?**
- Need smaller pages for the indexes?**
- Separate bufferpool for XML?**

```
create tablespace relData pagesize 4K bufferpool bp4k;
create tablespace xmlData pagesize 32K bufferpool bp32k;

CREATE TABLE mytable(c1 integer, c2 char(8),...,c9 double, c10 XML)
  IN relData
  INDEX IN mytspace2
  LONG IN xmlData
```

XDA (XML Data Area) will go to LONG, if specified

**Place XML data and/or indexes in separate table spaces to use different page sizes and separate tuning (prefetch size, etc.)...**

**...but only if really needed ! Otherwise keep it simple !**

## XDA Compression

- **XDA = object where XML documents are stored that are not inlined in the base table**
- **DB2 9.7:**
  - XML docs in XDA object can also be compressed
  - With small documents, using XDA compression (no inlining) overall compression ratio could be 70%
- **Enable/Disable Compression**

```
CREATE TABLE table1 ( docID int, docName varchar(20), doc xml) COMPRESS YES;
ALTER TABLE table1 COMPRESS NO;
```

- **Automatic Dictionary Creation (ADC)**
  - Triggers when data in the XDA crosses 2MB threshold
  - Can be triggered during:
    - Insert/Import
    - Load Insert

## How to get Data In?

- **Implicit XML parsing:**
  - Inserting data of XML data type info a column

```
INSERT INTO dept VALUES
('PR27', ..., '<dept>...<emp>...</emp>...</dept>')
```

- **Explicit XMLPARSE**
  - Transform XML value from serialized (text) form into internal representation.
  - Tell system how to treat whitespaces (strip/preserve)
    - Default is 'Strip WHITESPACE'

```
INSERT INTO dept VALUES ('PR27', xmlparse(document '<a>...</a>'));
INSERT INTO dept VALUES ('PR27',
xmlparse(document '<a>...</a>' preserve whitespace));
```

# Deleting XML Data

#### ▪ **DELETE**

- Will delete every XML document for a row

```
DELETE FROM dept WHERE deptID='A001'
```

- You can also delete based on the XML content

```
DELETE FROM dept WHERE  
    XML EXISTS ('$d//phone[type="Home"]'  
                passing INFO as "d")
```

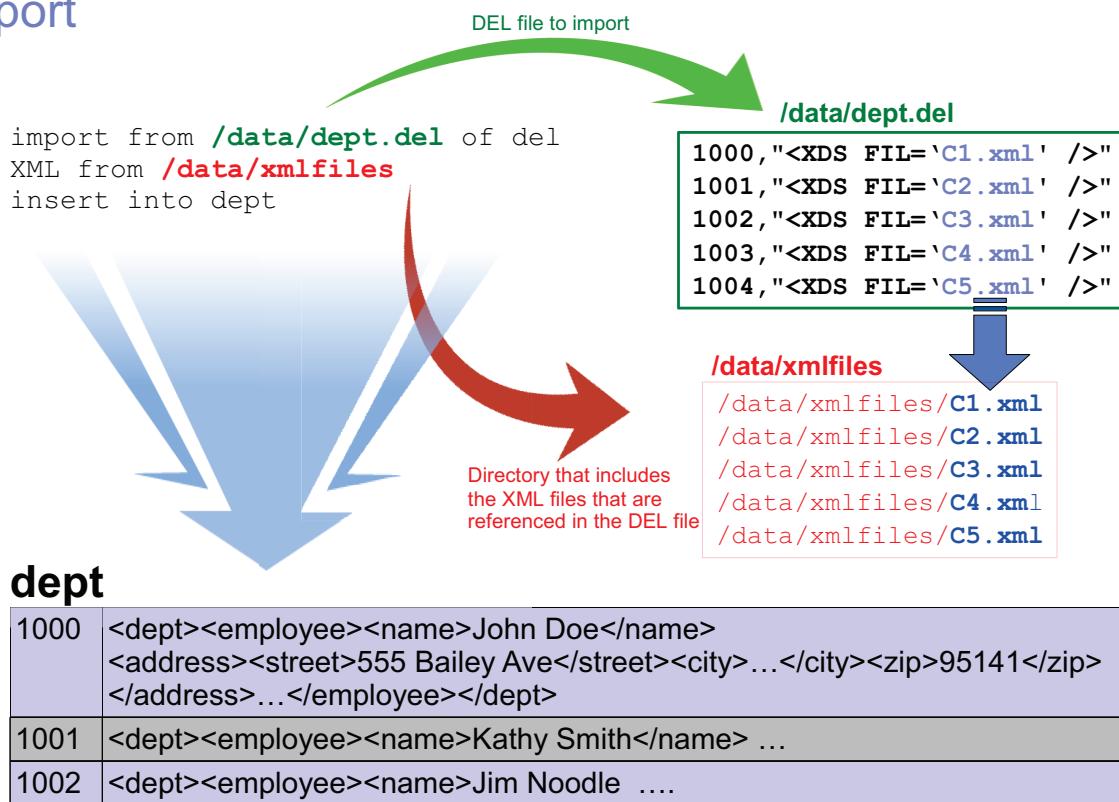
- Note: Setting an XML column to NULL deletes the XML document

```
UPDATE dept SET custDoc = NULL WHERE deptID='A001'
```

31

© 2011 IBM Corporation

Information Management



## Export

### dept

1000	<dept><employee><name>John Doe</name><address><street>555 Bailey Ave</street><city>...</city><zip>95141</zip></address>...</employee></dept>
1001	<dept><employee><name>Kathy Smith</name> ...
1002	<dept><employee><name>Jim Noodle ....

EXPORT TO **/data/dept.del** of DEL → DEL file to output  
 XML TO **/data/xmlfiles** → Directory to place XML files  
 XMLFILE deptdoc → Base name for exported XML files  
 MODIFIED BY XMLINSEPFIES → Store each XML document in separate file  
 (Optionally: Concatenate all XML documents in one large file.)  
 SELECT \* FROM dept → What to export

### /data/dept.del

```
1000,"<XDS FIL='C1.xml' />"  
1001,"<XDS FIL='C2.xml' />"  
1002,"<XDS FIL='C3.xml' />"  
1003,"<XDS FIL='C4.xml' />"  
1004,"<XDS FIL='C5.xml' />"
```

### /data/xmlfiles

```
/data/xmlfiles/C1.xml  
/data/xmlfiles/C2.xml  
/data/xmlfiles/C3.xml  
/data/xmlfiles/C4.xml  
/data/xmlfiles/C5.xml
```

## SQL/XML and XQuery

- **DB2 Supports two query languages:**

- **XQuery**
- **SQL/XML**

- **XPath**

- Cornerstone for both XQuery and SQL/XML standard
- Provides ability to navigate within XML documents

- **XQuery**

- Two important functions to access the database:
  - db2-fn:sqlquery
  - db2-fn:xmlcolumn
- Results returned as a sequence of items

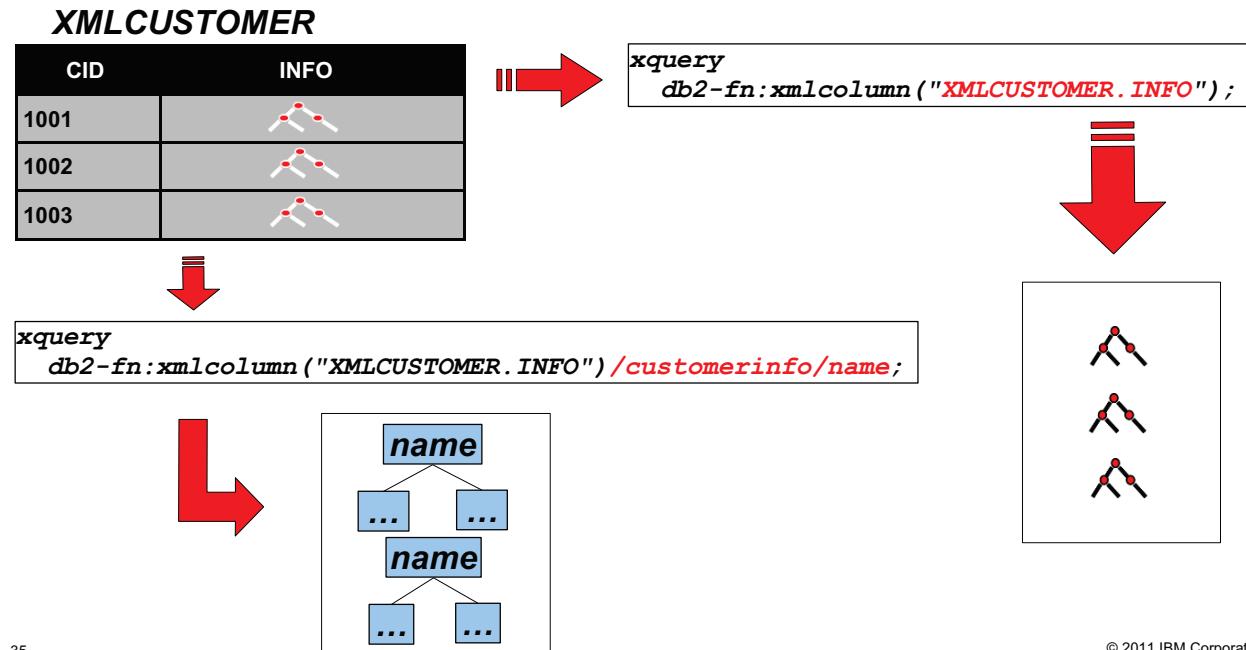
- **SQL/XML**

- Provides functions to work with both XML and relation data at the same time.

## XQuery Accessing XML Data

- **db2-fn:xmlcolumn**

- Retrieve all XML documents from an XML column, then process them with an XQuery expression.

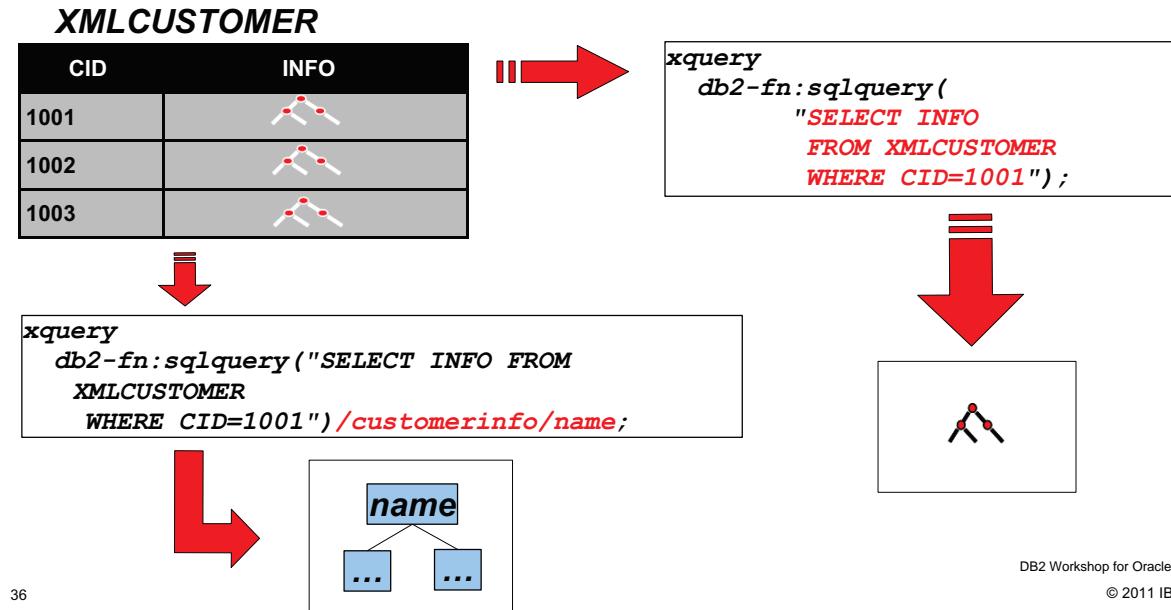


35

## XQuery/SQL accessing XML and Relational Data

- **db2-fn:sqlquery**

- Retrieve an XML document using SQL, then process it with an XQuery expression
- Allows filtering based on relational data



36

## SQL/XML Functions Supported in DB2

Function Name	Description
XMLVALIDATE	Validates XML value against an XML schema and type-annotates the XML value.
XMLEXISTS	Determines if an XQuery returns a result (i.e. a sequence of one or more items).
XMLQUERY	Executes an XQuery and returns the result sequence.
XMLTABLE	Executes an XQuery and returns the result sequence as a relational table (if possible).
XMLCAST	Casts to or from an XML type.
XMLEMENT	Creates an XML element.
XMLATTRIBUTES	Used within XMLEMENT to create attributes.
XMLFOREST	Produces a forest of XML elements from a list of SQL values.
XMLCONCAT	Concatenates a list of XML values.
XMLNAMESPACE	Provides XML namespace declarations in an XML element.
XMLAGG	Groups a set of XML rows.
XMLCOMMENT	Generates an XQuery comment node.
XMLPI	Generates an XQuery processing instruction node.
XMLTEXT	Generates an XQuery text node.
XMLDOCUMENT	Generates an XQuery document node.
XMLSERIALIZE	Converts an XML value into character/BLOB data.
XMLPARSE	Parses character/BLOB data, produces XML value.

37

© 2011 IBM Corporation

## SQL/XQuery XML Data for SQL Developers

### ■ XMLTABLE

–Creates a temporary SQL table using XML data

#### XMLCUSTOMER

CID	INFO
1001	
1002	
1003	

```
SELECT T.*  
FROM XMLTABLE (  
    'db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo'  
    COLUMNS "NAME" VARCHAR (20) PATH 'name',  
            "STREET" VARCHAR (20) PATH 'addr/street',  
            "CITY" VARCHAR (20) PATH 'addr/city'  
) AS T
```

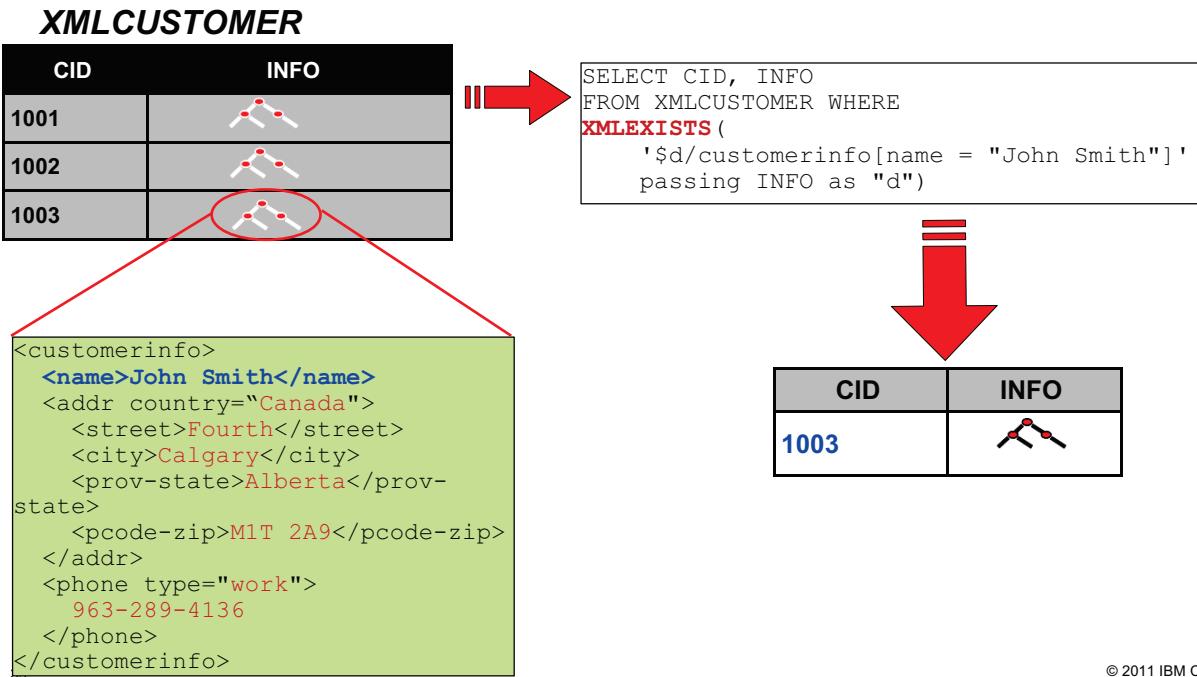
```
<customerinfo>  
  <name>John Smith</name>  
  <addr country="Canada">  
    <street>Fourth</street>  
    <city>Calgary</city>  
    <prov-state>Alberta</prov-  
state>  
    <pcode-zip>M1T 2A9</pcode-zip>  
  </addr>  
  <phone type="work">  
    963-289-4136  
  </phone>  
</customerinfo>
```

NAME	STREET	CITY
Amir Malik	Young	Toronto
John Smith	Fourth	Calgary
...	...	...

## SQL/XQuery XML Data for SQL Developers

### ▪ XMLEXISTS

- Predicate that tests if an XQuery expression returns a sequence



© 2011 IBM Corporation

## XML Schemas: Schema Overview

### ▪ XML Schema

- Define structure and data type constraints of XML elements
- If a XML document satisfies the schema definition, the **XML document is valid to the XML Schema**

### ▪ DB2 XML Schema Repository (XSR)

- Support for management of XML schemas
- Stable & high performance access to schemas
  - Avoids overhead to locate external documents
- All XML schemas **need to be registered** in the DB2 XSR before they can be used for validation

### ▪ What do you need to do?

- Register the XML Schema
- Use the **XMLVALIDATE** built-in function to validate your XML documents

## XML Schemas: Compatible XML Schema Evolution

- **Most XML Schema evolution is “compatible”, e.g.:**
  - Addition optional elements and attributes
  - Increasing the number of repeating elements
- **Non-compatible evolution is invasive to applications and less common**
- **Since v9.5, DB2 offers an “update XML schema” function which...:**
  - Assumes that the old and new schemas have been registered the normal way
  - Checks if the 2 schemas are 'compatible'
  - Replaces the old schema with the new one
    - Existing XML documents look like they were validated against the new XML Schema
    - Existing documents are not touched
    - New schema has the old schema's SQL name and objectID

## Schema Evolution: Previously...

***Employees need 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:**

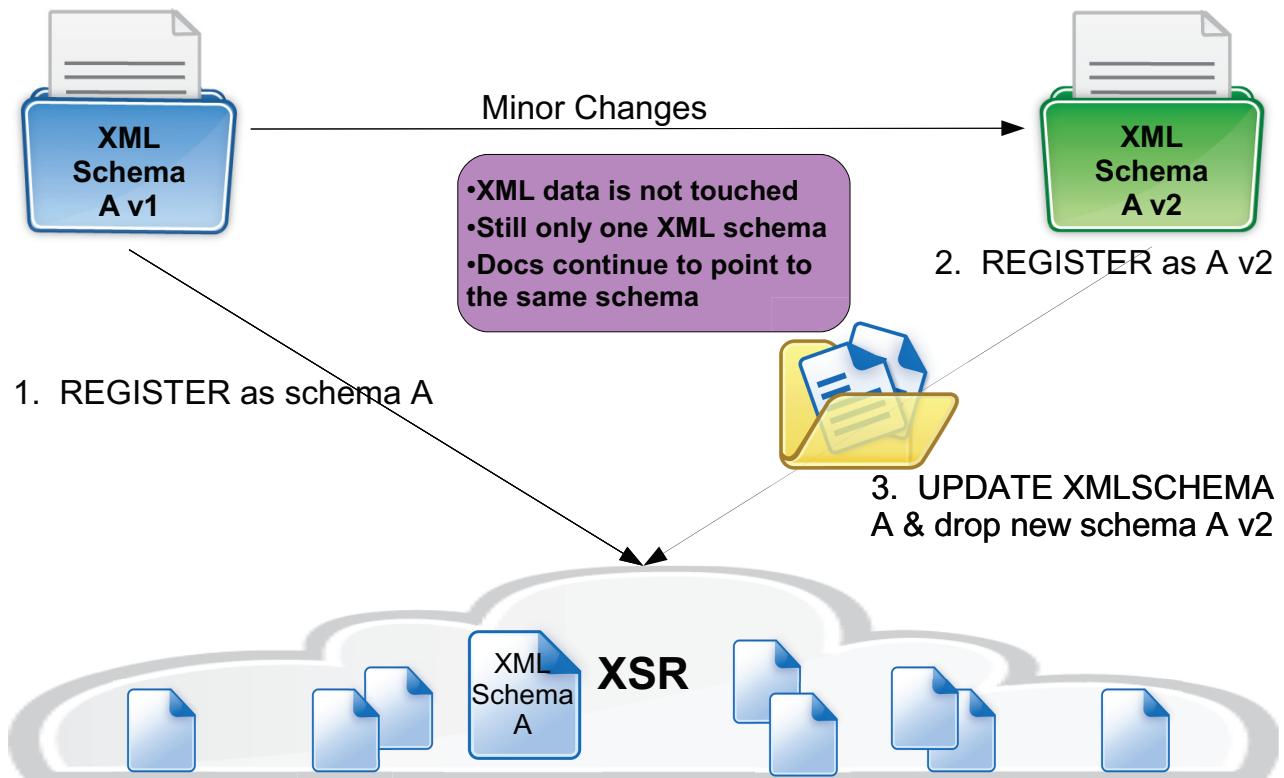
- Normalization of existing data !
- Modification of the mapping
- Change of applications

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

Department	
DEPTID	DEPTNAME
15	Sales

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

## Schema Evolution: Currently...



## SQL/XML Publishing Functions

- Publishing functions allow you to publish relational data from tables as an XML document.**
  - Construct XML documents with the following functions

Function Name	Type
XMLELEMENT	scalar
XMLATTRIBUTES	scalar
XMLFOREST	scalar
XMLCONCAT	scalar
XMLNAMESPACE	declaration
XMLAGG	aggregate
XMLCOMMENT	scalar
XMLPI	scalar
XMLTEXT	scalar
XMLDOCUMENT	scalar
XMLGROUP	aggregate
XMLROW	scalar
XMLTRANSFORM	scalar

**Example:** Construct an XML document with values from a single table

```
SELECT XMLELEMENT (
    NAME "Products"
    XMLROW(NAME as "item"))
FROM PRODUCT
```

**Result:**

```
<products>
  <row>
    <item>Snow Shovel</item>
  </row>
</products>
```

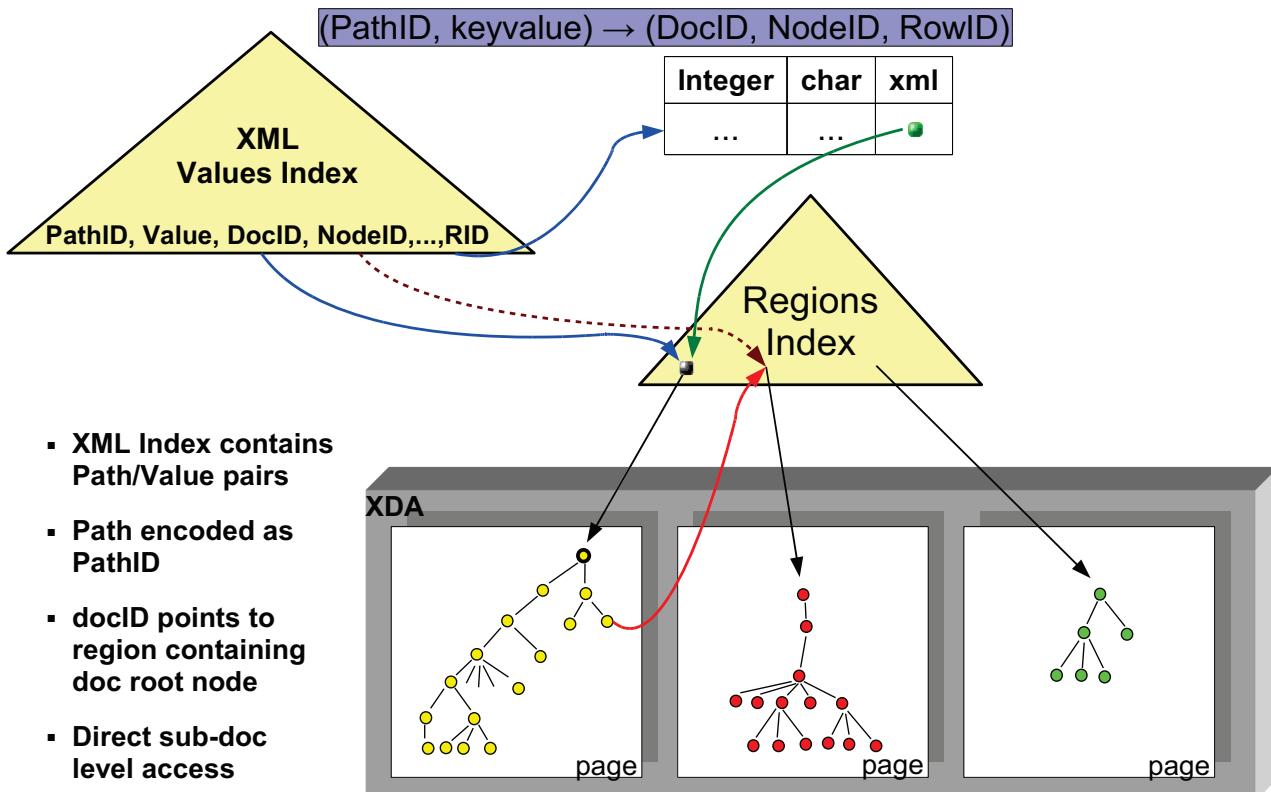
## XML Indexes in DB2: Index on an XML Column

- An index over XML data can be used to improve the efficiency of queries on XML documents
- It is based on XML pattern expressions
- Index entries will provide access to nodes within the document by creating index keys based on XML pattern expressions
- Like relational data they may have some cost
  - Performance for INSERT, UPDATE and DELETE
  - Space needed to store the indexes

Regular Indexes	Indexes for XML
Based on columns	Based on XML pattern expressions
1 or more columns	Only 1 XML column
1 row → 1 index key	All nodes that satisfy the XML pattern: 1 document → 0, 1 or more index keys
B-Tree	B-Tree

```
CREATE INDEX <IDX_NAME> ON <TABLE>(<XML_COLUMN>)
  GENERATE KEY USING XMLPATTERN <XML_PATTERN>
  AS SQL <DATA_TYPE>
```

## XML Indexes: Under the Covers



## Development Support for XML Data

- **C or C++ (embedded SQL or DB2 CLI)**
- **COBOL**
- **Java™**
  - JDBC 3.0 & 4.0
  - SQLJ
- **C# and Visual Basic (IBM® Data Server Provider for .NET)**
- **PHP**
  - Drivers for PHP to DB2 connectivity – ibm\_db2 & PDO\_IBM
  - Zend Core for IBM v2.0.1 ([http://www zend com/products/zend\\_core/zend\\_core\\_for\\_ibm](http://www zend com/products/zend_core/zend_core_for_ibm))
- **Perl**
- **Ruby**
- **SQL Procedures**
- **External routines (C, C++, COBOL, Java™, .NET languages)**

## JDBC and XML in DB2 (JDBC v4)(JSR 221)

- **JDBC 4.0 support has been added**
  - [IBM® Data Server Driver for JDBC and SQLJ Version 4.0](#) includes a number of JDBC 4.0 capabilities. To use these capabilities, you need an SDK for Java, Version 6.
- **SQLXML object added to JDBC specification**
  - It's the Java data type that maps the database SQL XML type.
  - `getSQLXML( . . . ) / updateSQLXML( . . . )`
  - It's a standard → Application portability
  - Support to the Source and Result interfaces from `javax.xml.transform` package.
    - Great integration with XML APIs: DOM, SAX
    - Benefits from DB's optimized representation of the XML data

## JDBC 4.0 – Additional Capabilities

- **DB2Connection.registerDB2XMLSchema**
  - Registers an XML schema in DB2, using one or more XML schema documents. There are two forms of this method: one form for XML schema documents that are input from an InputStream objects, and one form for XML schema documents that are in a String.
- **DB2Connection.deregisterDB2XMLObject**
  - Removes an XML schema definition from DB2
- **DB2Connection.updateDB2Xm1Schema**
  - Replaces the XML schema documents in a registered XML schema with the XML schema documents from another registered XML schema. Optionally drops the XML schema whose contents are copied.

## XML – Conclusion

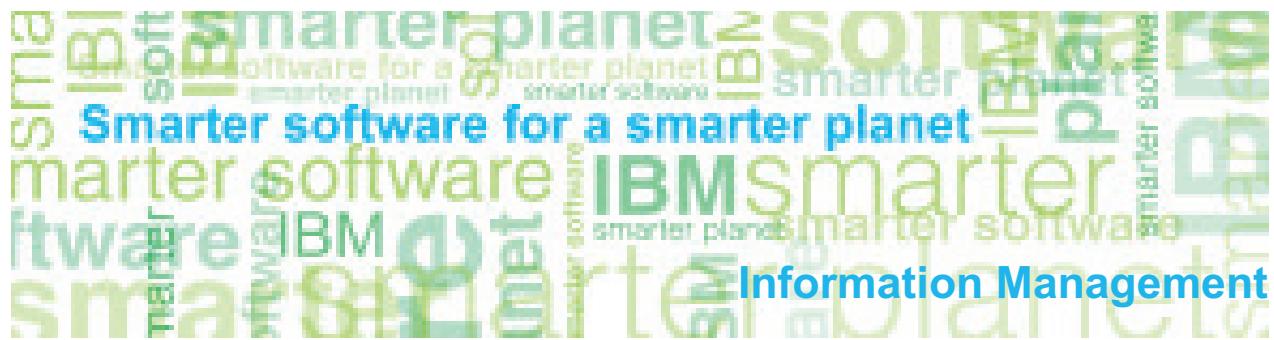
- **Native XML hierarchical storage**
  - No shredding, no CLOBs, no BLOBs required
  - Optimized for XPATH and XQuery (**L UW Only**) processing
- **High performance**
  - Superior indexing technology
  - No parsing of XML data at query runtime
- **Fully integrated XML and relational processing**
  - Seamlessly query various types of data at once
  - No internal translation of XQuery into SQL
- **Schema flexibility**
  - Changes don't force unload / reload of data
  - Multiple schemas allowed per XML column

## Questions?

Winter/Spring 2011

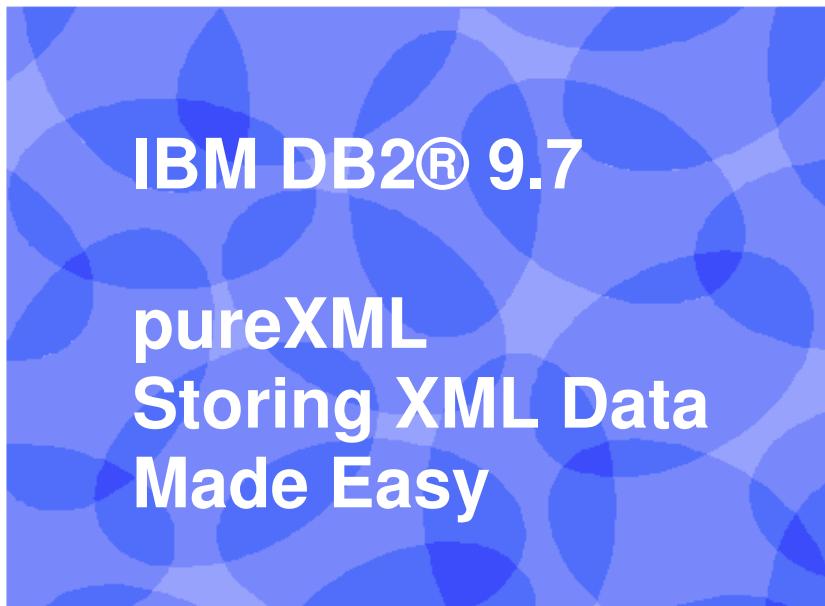
E-mail: [techcamp@ca.ibm.com](mailto:techcamp@ca.ibm.com)

Subject: "DB2 Workshop for Oracle Professionals"



© 2011 IBM Corporation

**IBM** Information Management software



**Information Management Technology Ecosystem**

**IBM Canada Lab**

---

## Contents

<b>1. XML BASICS AND INTRODUCTION.....</b>	<b>4</b>
<b>2. SETUP AND CREATE XML TABLES.....</b>	<b>4</b>
2.1    LOGIN TO THE VIRTUAL MACHINE.....	4
2.2    START DB2 SERVER, AND RUN CREATEDIEMO.SH .....	5
2.3    LAUNCHING DATA STUDIO.....	5
2.4    DATABASE CONNECTING.....	6
2.5    FILTERING BY SCHEMA.....	7
<b>3. VERIFY XML DATA.....</b>	<b>8</b>
3.1    VERIFY "XMLCUSTOMER" TABLE DATA.....	8
3.2    VERIFY "XMLPRODUCT" TABLE DATA.....	9
<b>4. EXECUTING XML QUERIES .....</b>	<b>9</b>
4.1    CREATING A DATA DEVELOPMENT PROJECT.....	9
4.2    CREATING AND EXECUTING XML QUERY .....	10
<b>5. WORKING WITH XML STANDARDS .....</b>	<b>11</b>
5.1    THE OTADB DATABASE.....	12
5.2    BACKGROUND .....	13
5.3    DATA INTEGRITY .....	13
5.3.1    Registering XML Schemas.....	14
5.3.2    Listing XML Schemas .....	17
5.3.3    Validating XML documents .....	17
5.3.4    Import & Validate .....	18
5.3.5    Enforcing XML Validation.....	19
TRANSFORMING YOUR XML DATA .....	21
5.4    XML INDEXES .....	23
5.5    PUBLISHING IN XML FORMAT.....	25
<b>6. QUERY QUESTIONS .....</b>	<b>28</b>
6.1    XQUERY AND SQL/XML.....	28
6.2    XQUERY WITH UPDATE FACILITY .....	29
<b>7. CHEAT SHEET .....</b>	<b>30</b>
7.1    SQL/XML .....	30
7.2    XQUERY .....	30
7.3    XQUERY WITH EMBEDDED SQL.....	30
7.4    XQUERY: THE FLWOR EXPRESSION .....	30
7.5    EXAMPLES ON HOW TO RETURN DIFFERENT OUTPUTS .....	31
7.6    NESTED FLWOR EXPRESSION .....	31
7.7    XPATH EXAMPLE.....	32

7.8	XQUERY – XML UPDATE FACILITY .....	32
7.9	XQUERY: UPDATING DOCUMENT IN THE DATABASE.....	32
7.10	XQUERY: UPDATING DOCUMENT ON THE FLY .....	32
7.11	XQUERY: DELETING A NODE.....	32
7.12	XQUERY: INSERT A NODE AT THE END OF A DOCUMENT. ....	33
7.13	XQUERY: INSERT A NODE BEFORE ANOTHER. ....	33
7.14	XQUERY: RENAMING AN ELEMENT .....	33
<b>8.</b>	<b>STRUCTURE OF XMLCUSTOMER TABLE XML DOCUMENTS .....</b>	<b>33</b>
8.1	TREE STRUCTURE – EXAMPLE 1.....	33
8.2	XML STRUCTURE – EXAMPLE 1: .....	34
<b>9.</b>	<b>STRUCTURE OF XMLPRODUCT TABLE XML DOCUMENTS.....</b>	<b>35</b>
9.1	TREE STRUCTURE – EXAMPLE 2:.....	35
9.2	XML STRUCTURE – EXAMPLE 2: .....	35
<b>10.</b>	<b>ANSWERS TO QUERIES .....</b>	<b>36</b>
10.1	ANSWER TO XQUERY AND SQL/XML WITH NO UPDATES .....	36
10.2	ANSWERS TO QUERIES WITH XML UPDATES .....	38
<b>APPENDIX A</b>	<b>.....</b>	<b>38</b>

---

## 1. XML Basics and Introduction

**XML** (Extensible Markup Language) is a flexible self-describing data structure. XML stores information in a meaningful structure which can be easily interpreted by humans or computers. XML provides a way to store information independent of the platform and language. This structure is not in a fixed format or syntax and it can be easily changed. The structure consists of XML tags which are used to describe each element and their attributes.

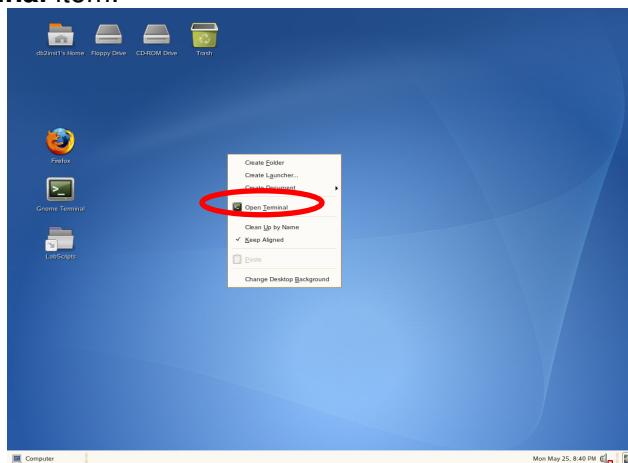
DB2 9 has the capabilities to store XML data natively within the database – this feature is known as **pureXML™**. Data is stored in a hierarchy of nodes (attributes, elements, comments, etc). In this lab, you will learn how to navigate through the XML data using XPATH. You will also learn how to create queries on the XML data using XQuery and SQL with embedded XQuery (SQL/XML).

---

## 2. Setup and Create XML Tables

### 2.1 Login to the Virtual Machine

1. Login to the VMware virtual machine using the following information:  
User: **db2inst1**  
Password: **password**
2. Type in the command **startx** to bring up the graphical environment.
3. Open a terminal window by right-clicking on the **Desktop** area and choose the **Open Terminal** item.

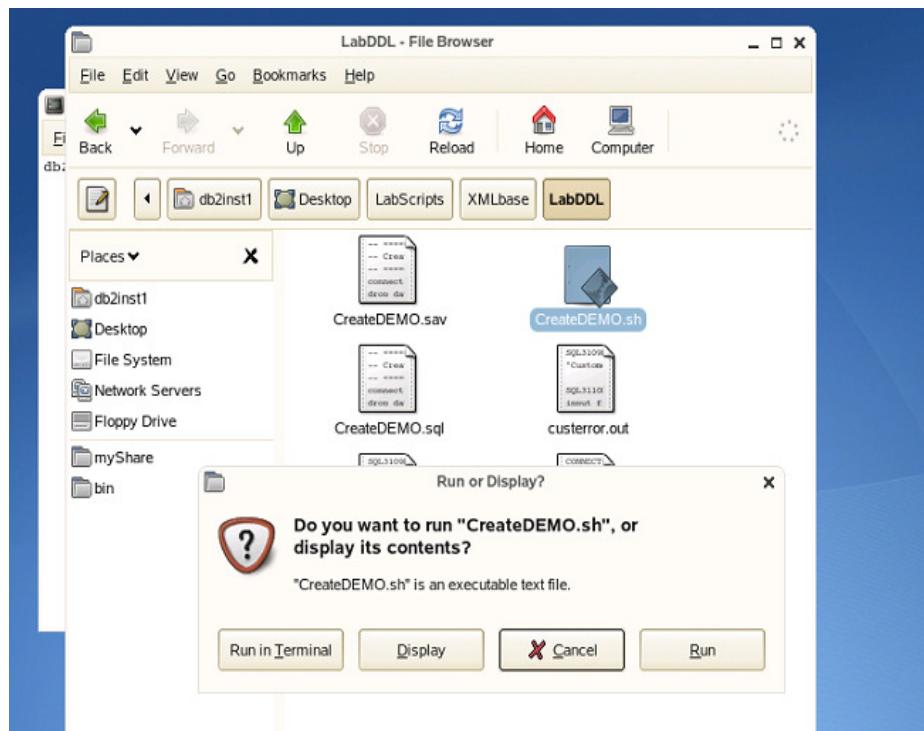


## 2.2 Start DB2 Server, and Run CreateDEMO.sh

1. Start up DB2 Server by typing the following commands in the terminal window:

**db2start**

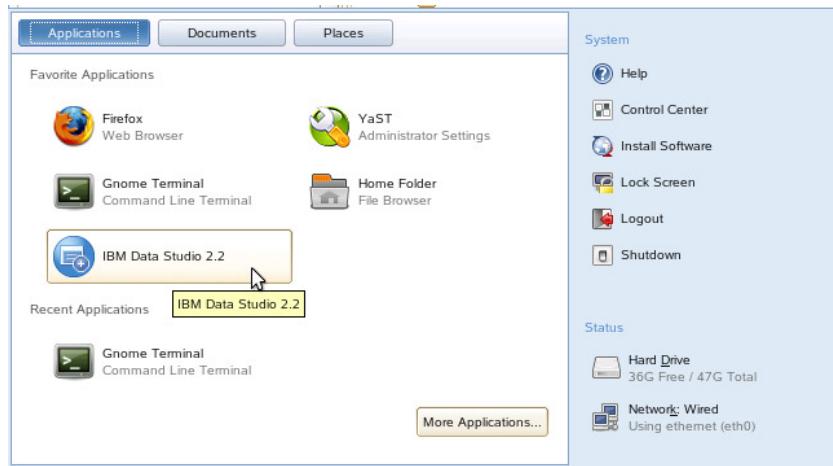
2. On the desktop, double click on the **LabScripts** folder.
3. In the **LabScripts** window that appears, double click on the **XMLbase** folder, once opened; double click on the **LabDDL** folder.
4. Double click on the **CreateDEMO.sh** file and click the **Run in terminal** button in the **Run or Display?** window.



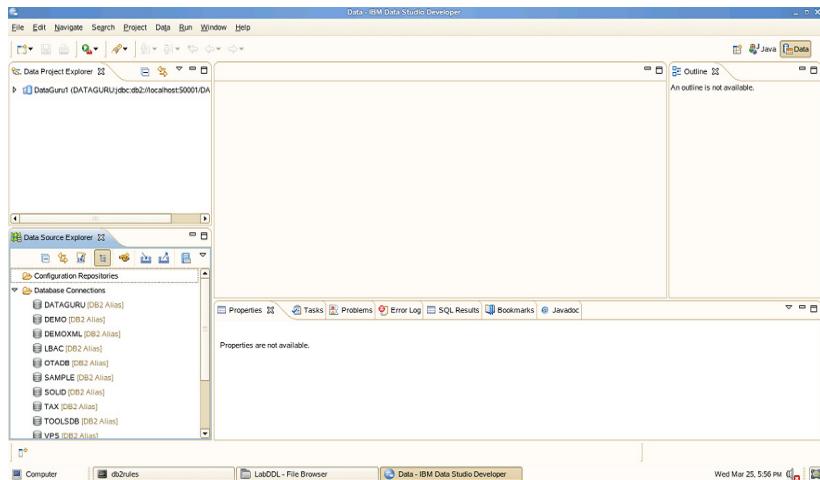
The **CreateDEMO.sh** script will create the **DEMOXML** database, along with several tables, views, indexes. This script also populates two tables that will be used throughout the lab sessions.

## 2.3 Launching Data Studio

1. Click on the **Computer** button in the bottom left corner of the screen, and select **IBM Data Studio**.



2. In the **Select a workspace** dialog, accept the default path and check the **Use this as the default and do not ask again** checkbox. Click **OK**.



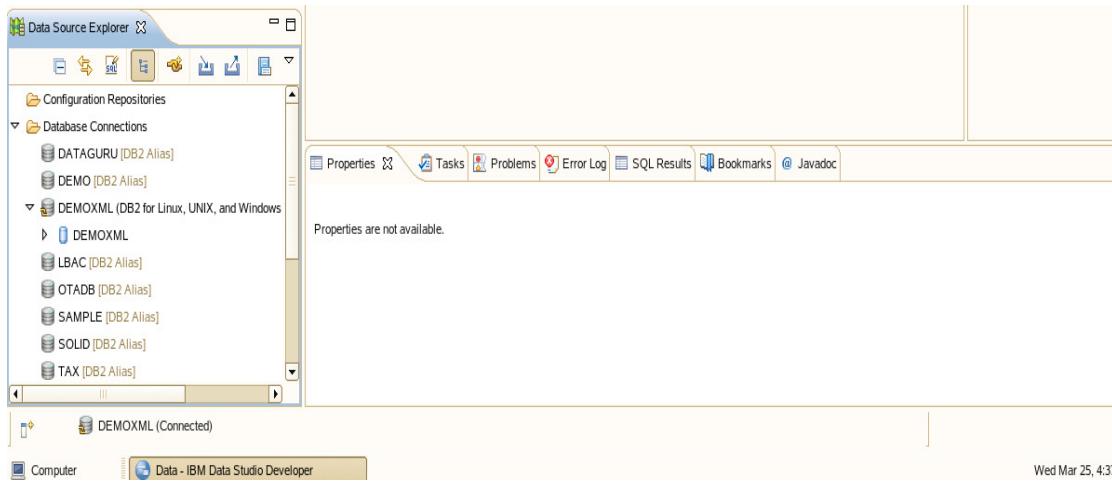
Data Studio will now start in the Data perspective as shown above.

## 2.4 Database Connecting

1. In the **Data Source Explorer** view, right-click on the **DEMOXML [DB2 Alias]** node, and Click **Connect**.
2. Enter the same credentials as before when the **Drive Properties** dialog appears:

**User Name:** db2inst1  
**Password:** password

Click **OK**. Notice that the **DEMOXML** connection icon now has a chain as well, signifying that the connection has been established.



## 2.5 Filtering by Schema

1. In the **Data Source Explorer**, expand the following nodes by clicking the icons beside them: **DEMOXML [DB2 for Linux...]** > **DEMOXML** > **Schemas**.

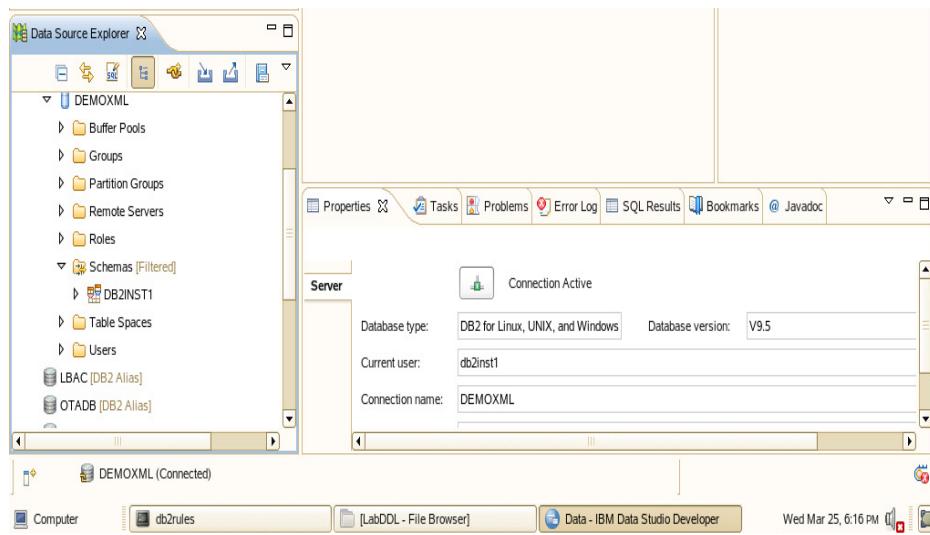
Notice that they are all sorts by schemas listed: DB2INST1, NULLID, SQLJ, etc. Because we will only be working with the **DB2INST1** schema, let's filter our list to show only that one schema (**DB2INST1**).

2. Right-click on the **Schemas** node and select **Filter...** The **Filter** dialog will appear, allowing you to filter either by typing in the name of a schema (or a portion thereof), or by selecting from a list of schemas. We'll filter by selection:

1. Uncheck the **Disable filter** checkbox.
2. Select the **Selection** radio button.
3. Select **Include selected items** in the drop down list.



4. Check the **DB2INST1** checkbox.
5. Click OK
3. Repeat step 1 and notice that the previous **Schemas** node now reads **Schemas [Filtered]**. Note as well that only **DB2INST1** now appears below it.

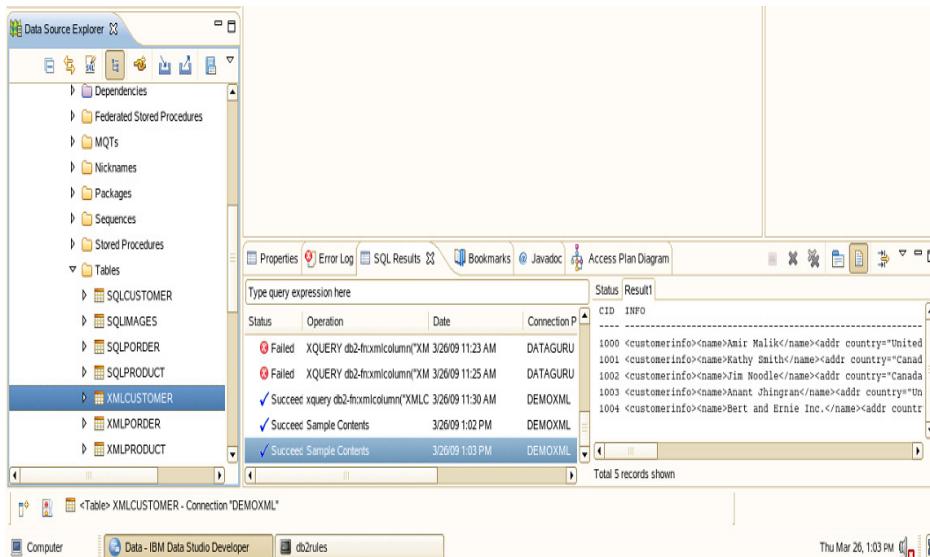


## 3. Verify XML Data

### 3.1 Verify “XMLCUSTOMER” Table Data

- In the Data Source Explorer, expand the following nodes by clicking the icons beside them: **DEMOXML [DB2 for Linux...]** > **DEMOXML** > **Schemas** > **DB2INST1** > **Tables** > **XMLCUSTOMER**.

Verify the sample data, right-click on the **XMLCUSTOMER** node and select **Data > Sample Contents**.



## 3.2 Verify “XMLPRODUCT” Table Data

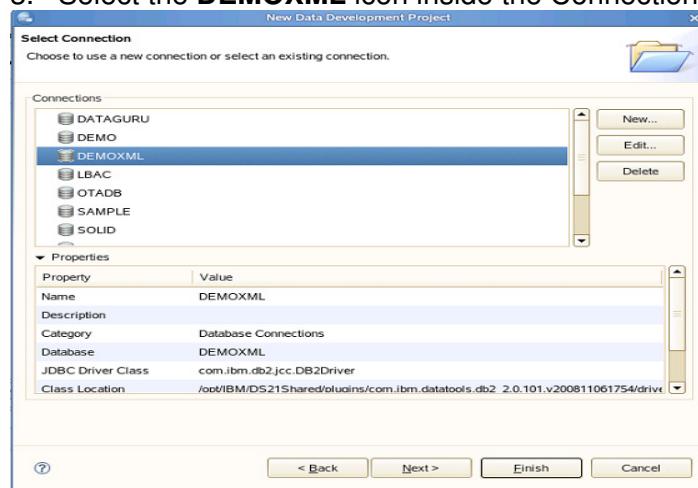
1. In the Data Source Explorer, expand the following nodes by clicking the icons beside them: **DEMOXML [DB2 for Linux...]** > **DEMOXML** > **Schemas** > **DB2INST1** > **Tables** > **XMLPRODUCT**.
2. Verify the sample data, right-click on the **XMLPRODUCT** node and select **Data > Sample Contents**.

The screenshot shows the IBM Data Studio Developer interface. On the left, the Data Source Explorer pane is open, showing a tree structure with nodes like Dependencies, Federated Stored Procedures, MQTs, Nicknames, Packages, Sequences, Stored Procedures, and Tables. Under Tables, several XML-related tables are listed: XMLCUSTOMER, XMLIMAGES, XMLPORDER, XMLPRODUCT, XMLCUSTOMER, XMLPORDER, and XMLPRODUCT. The XMLPRODUCT node under XMLPRODUCT is selected. On the right, the SQL Results pane displays the sample contents of the XMLPRODUCT table. The results table has columns: Status, Operation, Date, and Connection P. It shows various log entries, including successes for querying XML columns and sample contents, and one failure for XQUERY. The results pane also indicates a total of 44 records shown.

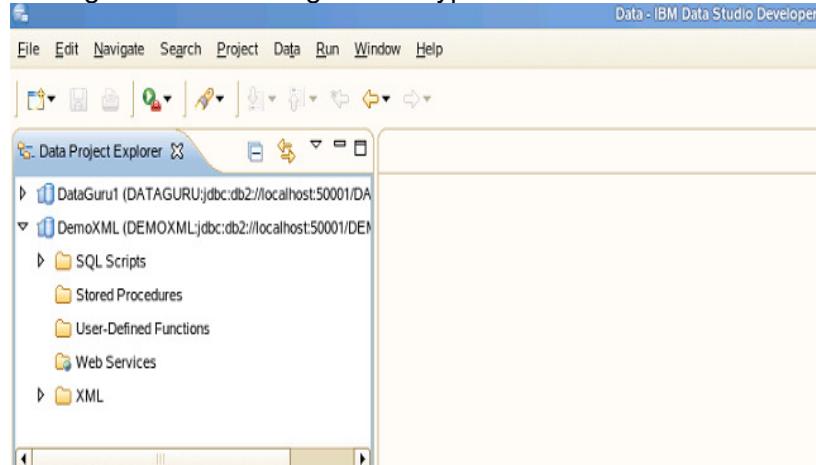
## 4. Executing XML Queries

### 4.1 Creating a Data Development Project

1. From the main menu, select **File > New > Data Development Project**.
2. In the **Project name** field, enter the value **DEMOXML**. Click **Next**.
3. Select the **DEMOXML** icon inside the Connections section and click **Finish**.



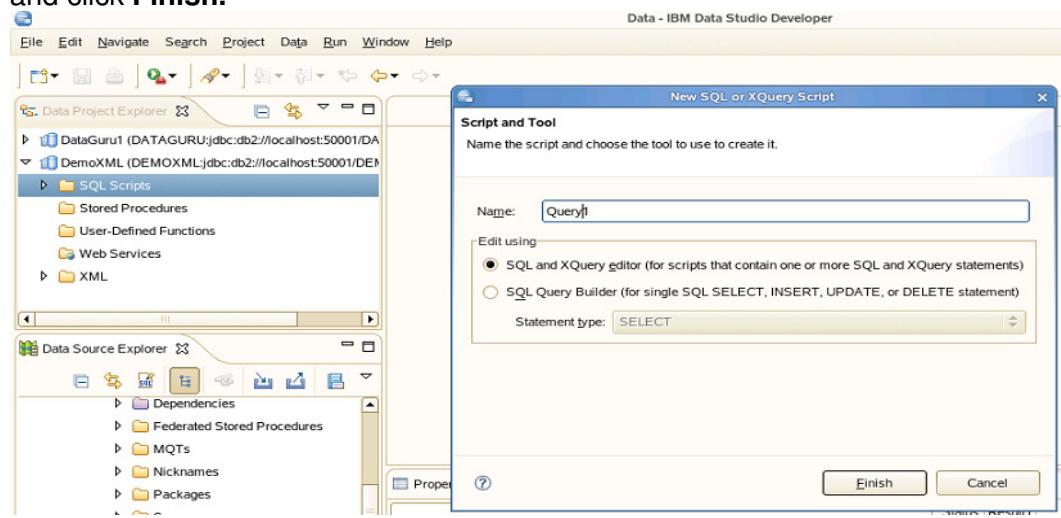
4. In the **Data Project Explorer** view, expand the **DEMOXML** project by clicking the icon. Notice the empty folders – these are where your project files will be organized according to their types.



## 4.2 Creating and Executing XML Query

Now that we have a project created, it's time to work with it. In working on a data project, you'll often have the need to write a simple SQL script outside of the context of a stored procedure or user defined function.

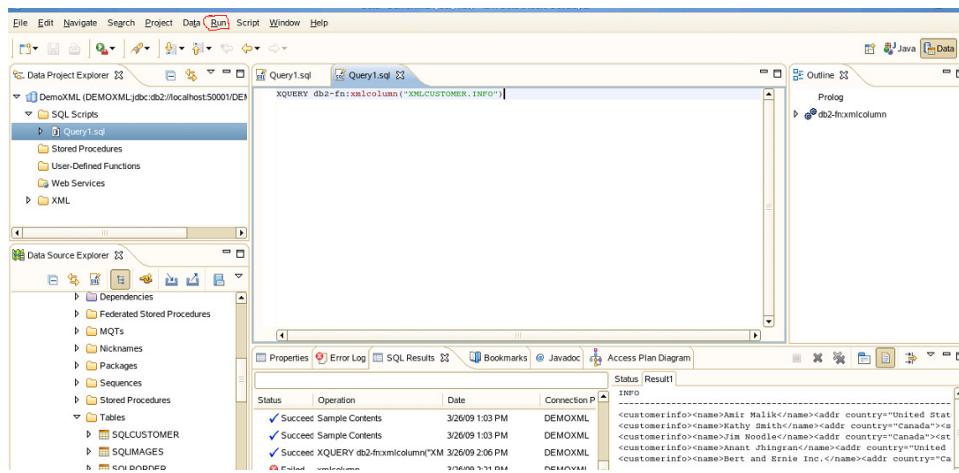
1. In the **Data Project Explorer**, right-click on the **SQL Scripts** folder and select **New > SQL or XQuery Script**.
2. In the Name field, enter the value **Query1**.
3. Select the **SQL and XQuery editor** radio button under the **Edit using** heading and click **Finish**.



4. Let's try to executing the first XQuery in the lab (Sample Question #1), by typing the following XQuery:

```
XQUERY db2-fn:xmlcolumn("XMLCUSTOMER.INFO")
```

5. From the main menu, select **Run > Run SQL**. The results of the query will appear in the **Results** tab of the **SQL Results** view, lower right corner.



6. When satisfied with the results of the above query, close the **Query1** tab by clicking on its icon. Click **Yes** when asked to save changes.

## 5. Working With XML Standards

Now that you have learned how to load XML data and how to retrieve it using SQL, SQL/XML and XQuery, we will begin using a real world XML industry standard to explore the pureXML technology.

The OTA (Open Travel Alliance - <http://www.opentravel.org>) specifications are used for business information exchanged in the area of travel, tourism and hospitality, used by airlines, car rental firms, hotels, cruise lines, railways, leisure suppliers, service providers, tour operators, and travel agencies.

You will play the role of a software developer of a fictional company called BonVoyageSoft. Your job is to explore DB2 pureXML capabilities for a project that needs to start using the OTA XML standard since this will be the standard for communicating data between travel systems.



This lab was based on the “**Industry Formats and Services with pureXML**” material available at IBM alphaWorks website.

<http://www.alphaworks.ibm.com/tech/purexml>

## 5.1 The OTADB database

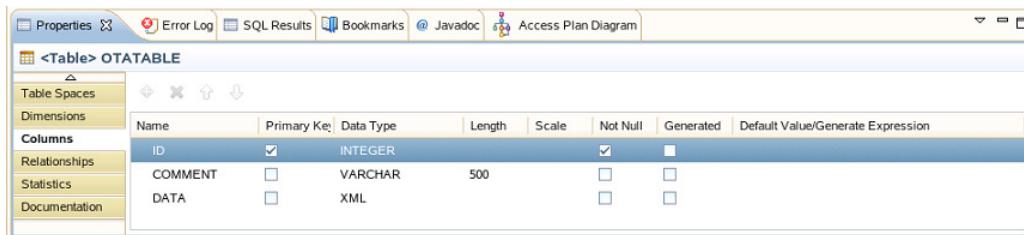
- For your pureXML experiments, you will be using the **OTADB** database. Before proceeding with the lab, take some time to explore it. Using **IBM Data Studio**, go to **Data Source Explorer** view, and double-click on **OTADB** node.
- Enter the same credentials as before when the **Drive Properties** dialog appears:

**User Name:** db2inst1  
**Password:** password

- Click **OK**

- In the **Data Source Explorer**, expand the following nodes by clicking the icons beside them: **OTADB > Schemas > XMLLAB > Tables**.
- Click on **OTATABLE** icon, in the view to the right click on **properties** tab. The **Properties** tab displays the columns and their properties. Repeat for **Transactions** and **Sales\_Types** tables.

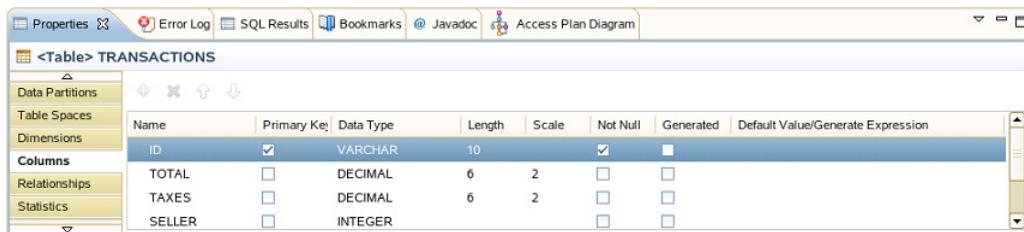
### OTATABLE



The screenshot shows the IBM Data Studio interface with the OTATABLE properties tab selected. The left sidebar shows the table structure with sections for Table Spaces, Dimensions, Columns, Relationships, Statistics, and Documentation. The main panel displays the column definitions:

Name	Primary Key	Data Type	Length	Scale	Not Null	Generated	Default Value/Generate Expression
ID	<input checked="" type="checkbox"/>	INTEGER			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
COMMENT	<input type="checkbox"/>	VARCHAR	500		<input type="checkbox"/>	<input type="checkbox"/>	
DATA	<input type="checkbox"/>	XML			<input type="checkbox"/>	<input type="checkbox"/>	

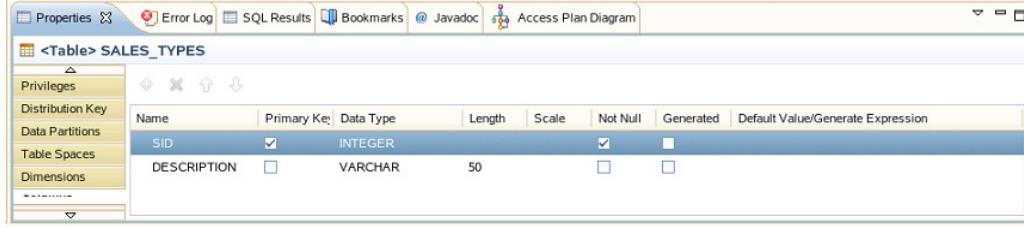
### TRANSACTIONS



The screenshot shows the IBM Data Studio interface with the TRANSACTIONS properties tab selected. The left sidebar shows the table structure with sections for Data Partitions, Table Spaces, Dimensions, Columns, Relationships, Statistics, and Documentation. The main panel displays the column definitions:

Name	Primary Key	Data Type	Length	Scale	Not Null	Generated	Default Value/Generate Expression
ID	<input checked="" type="checkbox"/>	VARCHAR	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TOTAL	<input type="checkbox"/>	DECIMAL	6	2	<input type="checkbox"/>	<input type="checkbox"/>	
TAXES	<input type="checkbox"/>	DECIMAL	6	2	<input type="checkbox"/>	<input type="checkbox"/>	
SELLER	<input type="checkbox"/>	INTEGER			<input type="checkbox"/>	<input type="checkbox"/>	

### SALES\_TYPES



The screenshot shows the IBM Data Studio interface with the SALES\_TYPES properties tab selected. The left sidebar shows the table structure with sections for Privileges, Distribution Key, Data Partitions, Table Spaces, Dimensions, and Documentation. The main panel displays the column definitions:

Name	Primary Key	Data Type	Length	Scale	Not Null	Generated	Default Value/Generate Expression
SID	<input checked="" type="checkbox"/>	INTEGER			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
DESCRIPTION	<input type="checkbox"/>	VARCHAR	50		<input type="checkbox"/>	<input type="checkbox"/>	

## 5.2 Background

BonVoyageSoft (the fictional company you work for) has a long history of developing applications for the travel/tourism industry. In order to keep up with your client demands and new technologies and industry standards you will need to enhance your applications to work with the OTA XML specifications.

One of the requirements is that all XML transactions/messages in the OTA specification need to be stored. For that you will need a database server capable of handling XML and relational data in a reliable and integrated way, while delivering high performance. That is where DB2 comes in.

In the next sections of this document you will be exploring each one of the following requirements in order to take BonVoyageSoft into the XML era:

- XML data integrity
  - Presenting your XML data in different formats
  - High performance querying for the XML data
  - Publishing relational data as XML
1. On the desktop, double click on the **LabScripts** folder.
  2. In the **LabScripts** window that appears, double click on the **pureXML** folder.
  3. Once opened; double click on the **LabData** folder.

The directory `/home/db2inst1/Documents/LabScripts/pureXML/labData` contains all SQL scripts, sample data and the OTA XML Schema documents used throughout this lab.

## 5.3 Data Integrity

You need to make sure that the XML data your application handles (1) remains unchanged when stored in the database and (2) that the XML documents content complies with the OTA specifications.

Being a database management system, DB2 automatically takes care of the first part, ensuring the integrity of your XML data for all transactions performed against the database.

The second part will need a little bit of set up from your part. In order to ensure the content of the XML documents are valid against the OTA specifications, you will need to validate the XML documents against the XML Schema definitions created by OTA. For that, we will take advantage of the **XML Validation** capabilities that DB2 provides.

The advantage of working with standards is that someone has already defined the XML documents structure for you, also called the XML schema. All you need to do is use the XML Schema documents (\*.xsd files) to validate your own XML documents. XML validation guarantees that a particular XML document is valid against the specification or domain that your application can handle, in our case, the OTA specification.

### 5.3.1 Registering XML Schemas

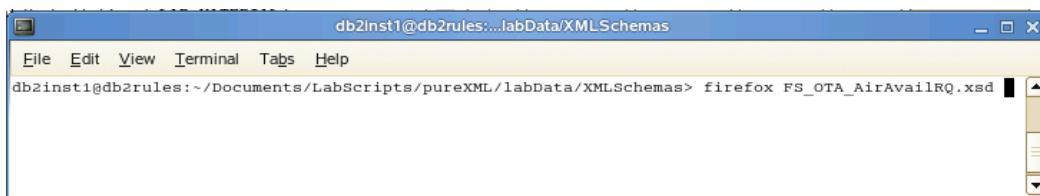
Before using an XML Schema to validate a document in DB2, you first need to register the XML Schema.

1. The OTA XML Schemas are located at:

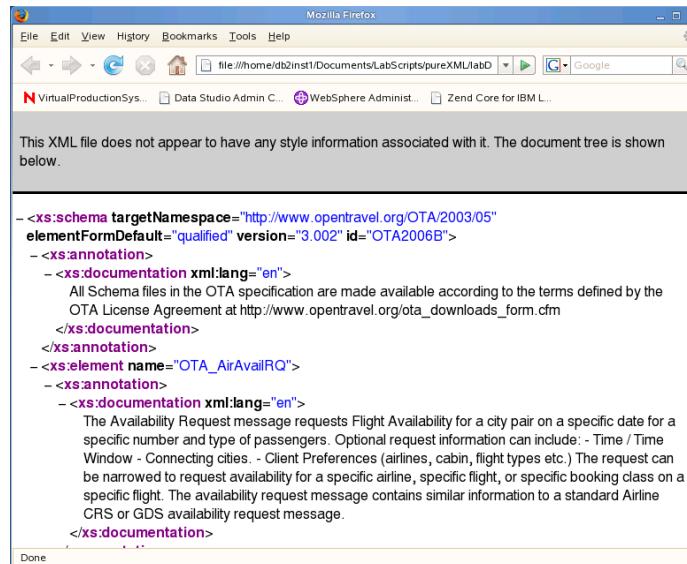
```
/home/db2inst1/Documents/LabScripts/pureXML/labData/XMLSchemas
```

2. You can visualize the contents of the \*.xsd files by:

- a. Open a terminal window on the Desktop;
- b. Change to the directory where the XML Schema files are located;
- c. Issue the command “**firefox <filename>**” to open the file in the browser. For example, to view the XML Schema for Air Availability XML messages, execute: “**firefox FS\_OTA\_AirAvailRQ.xsd**”.



- d. After investigating the file, close the browser window.

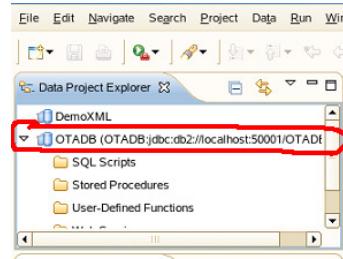


To register an XML schema with DB2 you need the \*.xsd files that contains the XML Schema definition. Perform the following steps to register the **FS\_OTA\_AirAvailRQ.xsd** XML Schema which defines the structure for Air Availability Request messages.

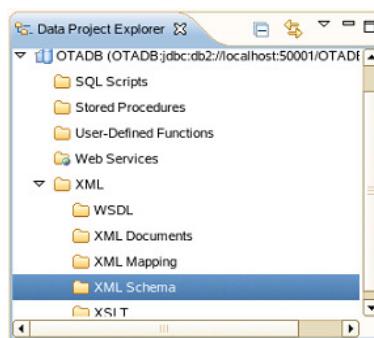
3. Using **IBM Data Studio**, from the menu bar click **File > New > Data Development Project** in the **Data Development Project** window enter:

Project name: **OTADB** > click **Next**;

4. Click **OTADB** > click **Finish**;



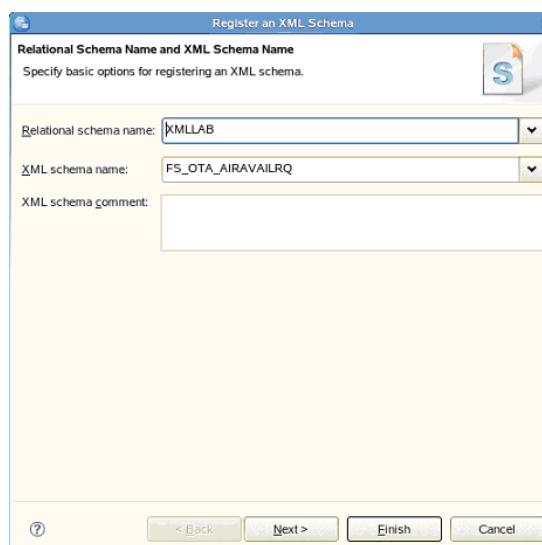
5. Using the **Data Project Explorer** view, expand the following nodes by clicking the icons beside them: **XML** > right-click **XMLSchema** > **Register an XML Schema...**



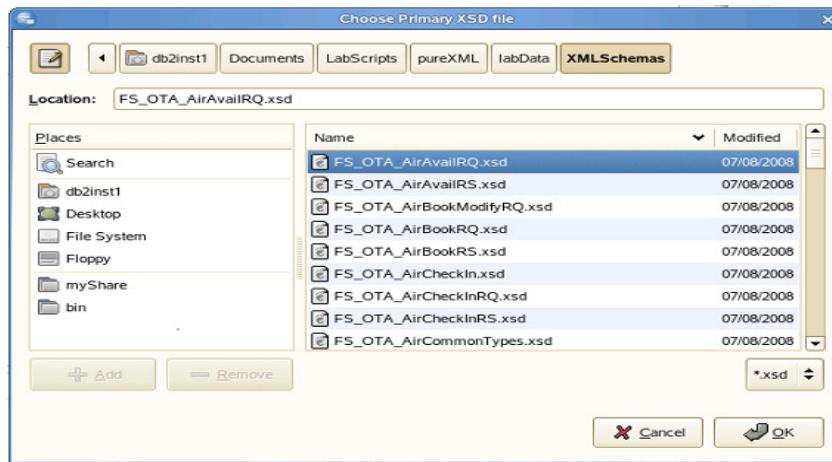
6. In the **Register an XML Schema...** window enter:

**Relational schema name:** XMLLAB

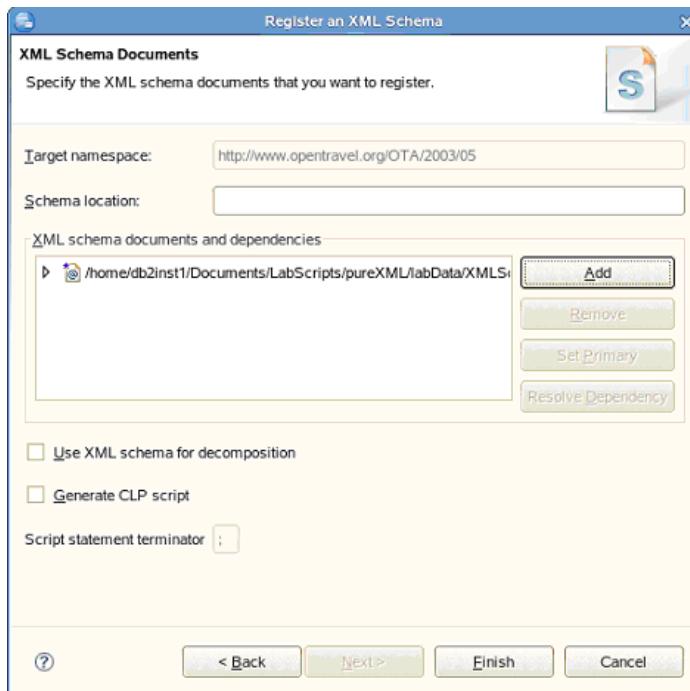
**XML schema Name:** FS\_OTA\_AirAvailRQ



7. Click **Next**;
8. In the **Register an XML Schema** window Click **Add**
9. In the Choose Primary XSD file window, Click **Documents > LabScripts > pureXML > labData > XMLSchemas**; Select **FS\_OTA\_AirAvailRQ.xsd**.



10. Click **OK** and then **Finish**;



11. Now, whenever you need, you can refer to this XML schema by its ID:  
`xmllab.FS OTA_AirAvailRQ.`

The OTA specification is actually composed of several XML Schema definitions. To save you time, we provide a script that will register all OTA XML Schema definitions for you.

12. If not already, open a terminal window by right-clicking on the **Desktop** area and choose the **“Open Terminal”** item;
13. Change directory to:  
`/home/db2inst1/Documents/LabScripts/pureXML/labData;`

14. Run the **register\_schemas.sql** script by making a connection to the **OTADB** database and execute **register\_schemas.sql** script. This task should take several minutes to complete as you are registering 233 XML Schemas.

```
db2 connect to OTADB
db2 -tvf ./register_schemas.sql
```

### 5.3.2 Listing XML Schemas

You have two options to get a list of all XML Schemas registered in your database. One is to query the **SYSCAT.XSROBJECTS** catalog table and the second is to use **IBM Data Studio**.

- From Command Line, execute the following query to list all XML Schemas:

```
SELECT objectschema, objectname, targetnamespace, schemalocation
FROM syscat.xsrobjects
WHERE objecttype='S' and status='C'
```

### 5.3.3 Validating XML documents

Now that you have registered your schemas, you can start using them to validate the XML documents you have in your database. Over the next several sections, to perform the following steps against the database, execute the following in the terminal:

**db2 "<query>"**

Where <query> are the set of statements outlined for each step.

- First, query the table to check the status of the XML documents. For that we will use the **IS VALIDATED** predicate, which returns **TRUE** if the XML document was validated against an XML Schema. Using Command Line, execute the following statement:

```
SELECT id
FROM xmllab.otatable
WHERE data IS VALIDATED
```

- As expected, no rows are returned since you have only inserted the data but did not validate them yet. To validate the XML documents in a table you will need to use the **XMLVALIDATE** function and update its status. Execute the following statement to validate the first XML document.

```
UPDATE xmllab.otatable
SET data = XMLVALIDATE(data ACCORDING TO XMLSCHEMA ID
xmllab.FS_OTA_AirAvailRQ)
WHERE id = 1
```

- Execute the query again to get all validated documents. This time the row with ID 1 is returned.

```
SELECT id
FROM xmllab.otatable
```

**WHERE data IS VALIDATED**

4. Now let's see what happens when we try to use the same XML Schema to validate the XML document in row 2.

```
UPDATE xmllab.otatable
SET data = XMLVALIDATE(data ACCORDING TO XMLSCHEMA ID
xmllab.FS_OTA_AirAvailRQ)
WHERE id = 2
```

5. The validation is not successful and DB2 informs you that element OTA\_AirAvailRS is not defined. This is expected since the XML document in row 2 is actually a Response message (OTA\_AirAvailRS) and you have specified the XML Schema for Request messages (OTA\_AirAvailRQ).

```
UPDATE xmllab.otatable SET data = XMLVALIDATE(data ACCORDING TO XMLSCHEMA ID xmllab.FS_OTA_AirAvailRQ) WHERE id = 2
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL16196N XML document contains an element "OTA_AirAvailRS" that is not
correctly specified. Reason code = "37"  SQLSTATE=2200M
```

37 The element is not defined.

6. Execute the statement again specifying the XML Schema for Response messages. This time the validation works.

```
UPDATE xmllab.otatable
SET data = XMLVALIDATE(data ACCORDING TO XMLSCHEMA ID
xmllab.FS_OTA_AirAvailRS)
WHERE id = 2
```

7. If you execute this query again, both rows are returned.

```
SELECT id
FROM xmllab.otatable
WHERE data IS VALIDATED
```

8. Sometimes you might want to search for documents that were validated against a specific XML Schema. You can do so by executing a query like the following:

```
SELECT id
FROM xmllab.otatable
WHERE data IS VALIDATED ACCORDING TO XMLSCHEMA ID xmllab.
FS_OTA_AirAvailRS
```

### 5.3.4 Import & Validate

Since you have all XML schemas already registered, you can bulk insert XML documents and validate them at the same time. Here we will use the IMPORT utility to fast insert the XML documents into the database and have them validated as well. All you need is to provide the XML Schema id in the XDS (XML Data Specifier) for each entry of your DEL file. Here is an example:

```
3, "", <XDS FIL='OTA_AirBookModifyRQ.xml' SCH='OTAADMIN.FS_OTA_AirBookModifyRQ' />
```

- From the Command Line, execute the following command to import the entries specified in the **otatable\_data.del** file. Note that you also specify where the location of XML files to be imported. The

```
IMPORT FROM
'/home/db2inst1/Documents/LabScripts/pureXML/labData/otatable_data.del' OF DEL XML FROM
'/home/db2inst1/Documents/LabScripts/pureXML/labData/XMLDATA'
XMLVALIDATE USING XDS COMMITCOUNT 10 INSERT INTO
XMLLAB.OTATABLE
```

- The **XMLVALIDATE USING XDS** clause states that the XML documents being imported are to be validated against the XML Schema specified in the XDS.
- Execute the following query to check that all imported documents are validated.

```
SELECT count(*)
FROM xmllab.otatable
WHERE data IS VALIDATED
```

### 5.3.5 Enforcing XML Validation

Now you have all the tools to make sure you will only have valid XML documents in your database, right? Let's check. Try to execute the following statement from the **Data Source Explorer** view, click **New SQL Script** button→**OTADB**→**Finish** and **Run SQL**.

```
INSERT INTO xmllab.otatable
VALUES (999, '', '<someElement>ABC</someElement>')
```

What happened? The row was inserted even though you provided a dummy XML document. This is not in accordance with your project requirements since it's possible that an application error or a user inserts an erroneous XML document into your database.

Fortunately, DB2 provides mechanisms to prevent this situation to happen. You can create CHECK constraints or triggers to ensure an XML document is valid before inserting it.

- First, let's delete the row you just inserted, execute the following statement from the Main View.

```
DELETE FROM xmllab.otatable WHERE id = 999
```

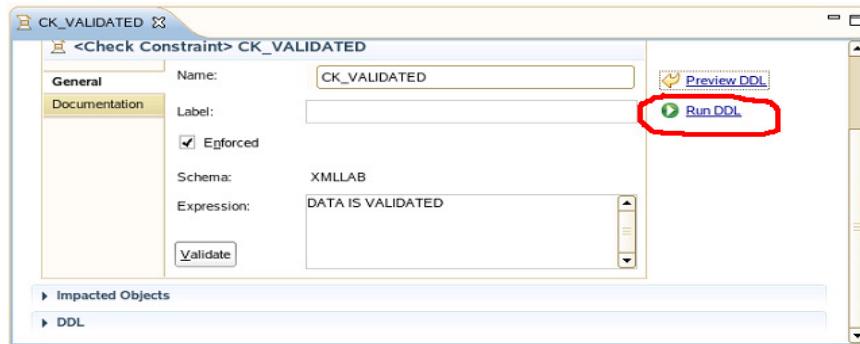
- Execute the following statement to create the CHECK constraint:

Expand the following nodes by clicking the icons beside them: **OTADB** > **Schemas** > **XMLLAB** > **Tables** > **OTATABLE**; right-click **Constraints** > **Create > Check Constraint**; Enter:

**Name:** CK\_VALIDATED

## Expression: DATA IS VALIDATED

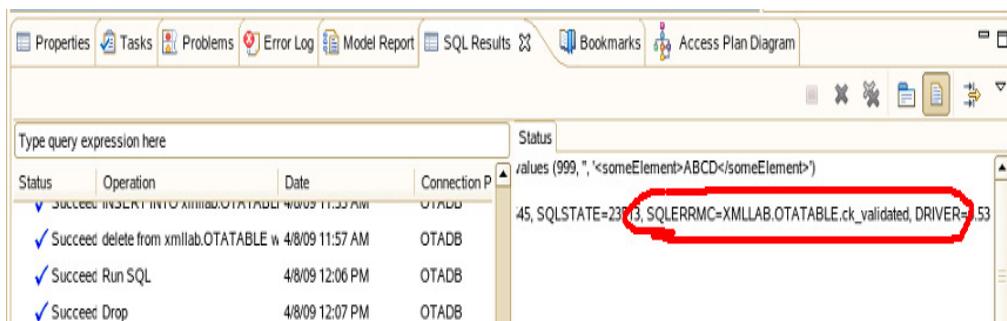
### 3. Run DDL



- Try to insert the dummy XML document again.

```
INSERT INTO xmllab.otatable
VALUES (999, '', '<someElement>ABC</someElement>')
```

- You will get an error stating the row does not satisfy the check constraint "XMLLAB.OTATABLE.CK\_VALIDATED".



- Now your table is protected against invalid XML document. Whenever you want to insert or update an XML document, you will have to use the XMLVALIDATE function to ensure the document is validated before it is stored. The INSERT statement below is an example of how you would do it:

```
INSERT INTO xmllab.otatable
VALUES (999, '',
XMLVALIDATE (XMLPARSE (DOCUMENT
'<OTA_VehLocDetailRQ
xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
Version="1.0">
<POS> <Source>
<RequestorID Type="5" ID="11112287"/>
</Source> </POS>
<Location LocationCode="DFW"/>
<Vendor Code="ZE"/>
</OTA_VehLocDetailRQ>' PRESERVE WHITESPACE )
```

ACCORDING TO XMLSCHEMA ID `xmllab.FS_OTA_VehLocDetailRQ` )

## Transforming Your XML Data

The next task is to find a way to transform your XML documents in order to present them in different formats such as HTML or even plain text. At the same time, you want to leverage the fact that your XML development team already developed some XSLT stylesheets that work with the OTA specification.

For example, the XSLT document below extracts information from an Air Booking Modify Request message and creates a HTML document from it. It creates a table by iterating over the `AirTraveler` elements. For each `AirTraveler` element it extracts the `PassengerTypeCode`, `Surname`, `GivenName`, `NameTitle` and `PhoneNumber`.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:ota="http://www.opentravel.org/OTA/2003/05">
<xsl:template match="/">
  <html>
    <body>
      <h2>Air Requests</h2>
      <table border="1">
        <tr bgcolor="#9acd32">
          <th align="left">Date</th>
          <th align="left">Origin</th>
          <th align="left">Destination</th>
        </tr>
        <xsl:for-each
select="ota:OTA_AirBookModifyRQ/ota:AirReservation/ota:TravelerInfo/ota
:AirTraveler">
          <tr>
            <td>
              <xsl:value-of select="@PassengerTypeCode"/>
            </td>
            <td>
              <xsl:value-of select="ota:PersonName/ota:Surname"/>,
              <xsl:value-of select="ota:PersonName/ota:GivenName"/> (<xsl:value-of
select="ota:PersonName/ota:NameTitle"/>)
            </td>
            <td>
              <xsl:value-of select="ota:Telephone/@PhoneNumber"/>
            </td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

With this scenario, you then decide to use DB2's built-in function **XSLTTRANSFORM** that can apply a XSLT transformation on a XML document and return the resulting document. Let's give it a try using the XSLT document above.

1. Using the Command Line change directory:

```
cd "/home/db2inst1/Documents/LabScripts/pureXML/labData"
```

2. To view script enter:

- a. less xslt\_script.sql
- b. (q)uit

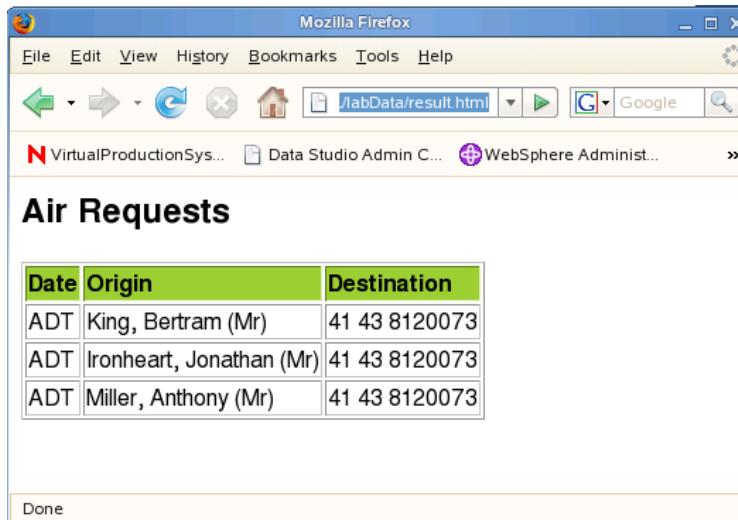
This script selects one Air Availability Request message from OTATABLE and use the **XSLTRANSFORM** function to create a HTML document.

```
SELECT XSLTRANSFORM(data using '<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:ota="http://www.opentravel.org/OTA/2003/05">
<xsl:template match="/">
  <html>
    <body>
      <h2>Air Requests</h2>
      <table border="1">
        <tr bgcolor="#9acd32">
          <th align="left">Date</th>
          <th align="left">Origin</th>
          <th align="left">Destination</th>
        </tr>
        <xsl:for-each
select="ota:OTA_AirBookModifyRQ/ota:AirReservation/ota:TravelerInfo/ota
:AirTraveler">
          <tr>
            <td>
              <xsl:value-of select="@PassengerTypeCode" />
            </td>
            <td>
              <xsl:value-of select="ota:PersonName/ota:Surname"/>,
              <xsl:value-of select="ota:PersonName/ota:GivenName"/> (<xsl:value-of
select="ota:PersonName/ota:NameTitle"/>)
            </td>
            <td>
              <xsl:value-of select="ota:Telephone/@PhoneNumber" />
            </td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>' AS VARCHAR(500))
FROM xmllab.otatable ota
WHERE ota.id=3'
```

3. Execute the SELECT statement. The text you get as result is in fact **HTML** text that can be visualized by the user on a web browser.

```
db2 -tvf ./xslt_script.sql > results.sav
```

4. If you want to see the HTML in a browser, you can copy and paste the results into a file and open it using your browser.
  5. To save time, we already created the file for you. Open file "/home/db2inst1/Documents/LabScripts/pureXML/labData/result.html" in your browser to visualize the results of the transformation.



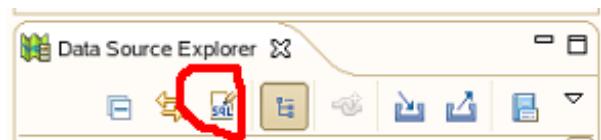
6. Similarly, if you need to present the data in any another format, for example CSV (Comma-Separated Values), you can just change the XSLT you provide and get the result in the format you want.

## 5.4 XML Indexes

In any database project, there is always the concern of delivering the best performance for queries execution. For this project in particular, you will be developing several queries against your XML documents, which means you will need a mechanism to improve the performance of these queries in case you need it.

DB2, just like with relational data, allows you to create indexes on your XML data to reduce the execution time of your queries. In the following steps you will try to optimize the performance of a query using indexes and use the Visual Explain tool to check the improvements.

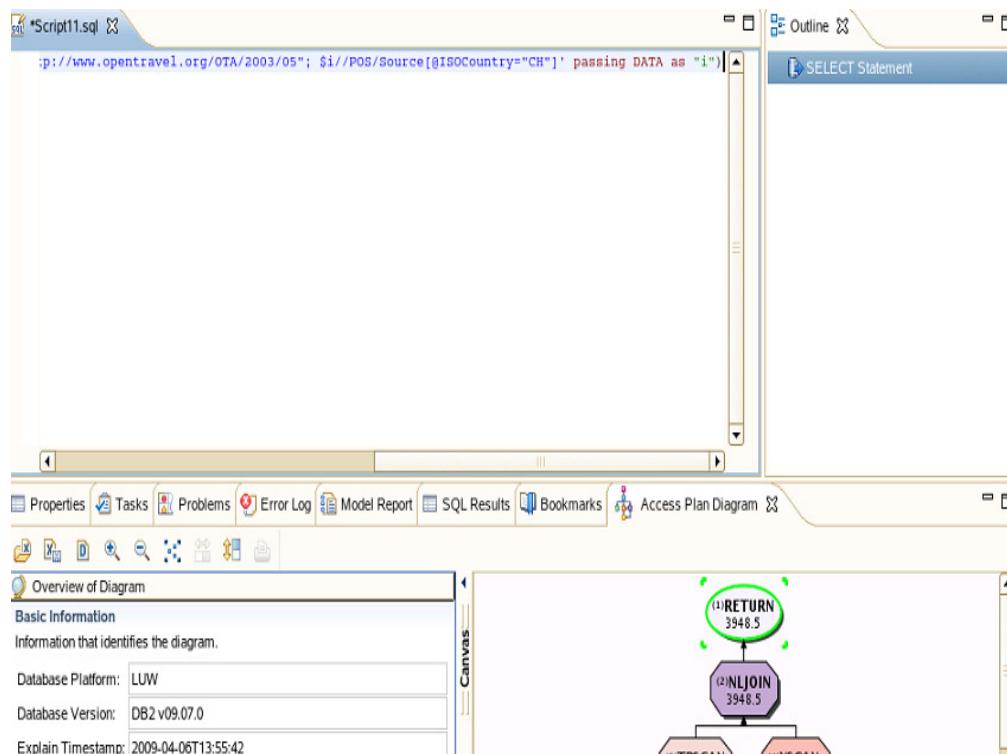
1. In the Database Explorer toolbar, click the SQL icon select OTADB→Finish. A new tab will appear in the main view, type the below query.



```
SELECT ID FROM xmllab.otatable WHERE XMLExists('declare
default element namespace
"http://www.opentravel.org/OTA/2003/05";
$i//POS/Source[@ISOCountry="CH"]'
passing DATA as "i")
```

**IMPORTANT:** Make sure that the statement is written on the same line, with no space between the “semicolon (;)” and “\$i”; otherwise this will not work.

2. From the main menu, select **Run > Run SQL**. If prompted to select a database connection to run the query, select **OTADB** and click **Finish**. The results of the query will appear in the **Results** tab of the **Data Output** view.
3. Your job is to optimize the query. It returns the ID of all XML documents where the value of **ISOCountry** attribute equals **CH**.
4. Before creating the indexes, check your query’s access plan. Select the entire block and right-click inside the main view (Script2.sql) and in the menu, select **Visual Explain** then click **Finish**. Keep this view open, we will want to rerun the query after creating the new index.



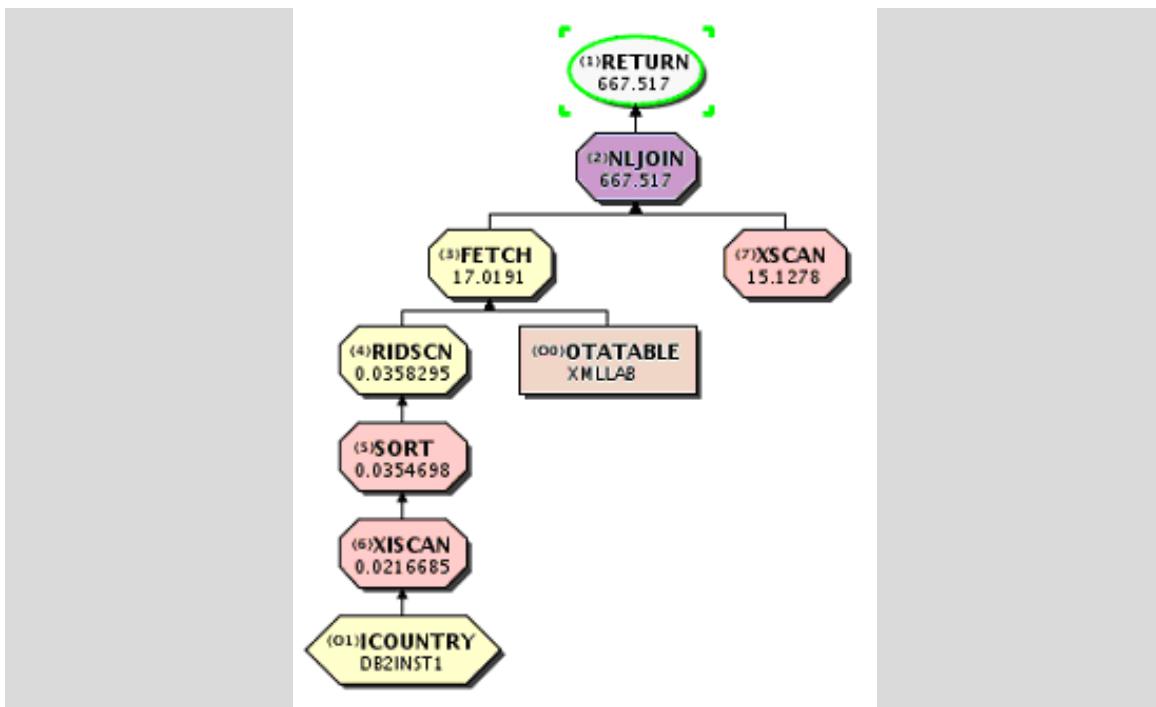
**Note:** The number in the RETURN operator might defer depending on your environment and DB2’s current internal state.

**Timerons** are an invented relative unit of measure. Timerons are determined by the optimizer based on internal values. It does not directly equate to any



actual elapsed time but gives a rough relative estimate of the resources (cost) required by the database manager to execute two plans for the same query base number to compare.

5. The next step is to create an XML index to reduce the execution time of your query. The index should address the predicate used in your query, in this case: `$i//POS/Source[@ISOCountry="CH"]`. To create the index, execute the following statements using the Command Line.
6. Using the Command Line, change directory:  
`"/home/db2inst1/Documents/LabScripts/pureXML/labData"`  
`db2 -tvf ./crt_iCountry.sql`
7. Rerun the query from the main menu, select **Run > Run SQL**. The results of the query will appear in the **Results** tab of the **Data Output** view.
8. Select the entire block and right-click inside the main view and in the menu, select **Visual Explain** then **Finish**. Notice the new plan makes use of the **iCountry** index you created in the previous step. Also compare the total estimated cost for this new plan. You will see there is a big improvement compared to the first plan.



## 5.5 Publishing in XML format

In some projects, you might have the necessity to communicate your data to other systems. You certainly know by now that one of the main uses of XML is data exchange between heterogeneous systems. Many protocols and data exchange formats use XML

as the structuring format, i.e., meaning they use XML document to carry the data between systems.

Your system is no different and also needs to exchange data with other systems. Some of these systems use XML-based messages to import and export data. The catch is that the data you need to import/export is stored in tables. That means you need a way to extract the data from your tables and transform it to a certain XML format. One alternative of course is to develop a program to do the job for you, which could imply big costs or changes to the underlying system.

What if you could delegate this task to the database server and get the XML documents generated straight from the data storage? That could save you development time and money, and also provide the same XML output to any application that needs it with no recoding.

For your project in particular, you need to export information about the completed sales to a certain XYZ financial system. XYZ can import information in the following format:

```
<XYZ_Trans>
  <Tran>
    <id>TR0001</id>
    <value>9999999.99</value>
    <taxes_value>9999999.99</taxes_value>
    <comment>CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC</comment>
  </Tran>
  <Tran>
    <id>TR0001</id>
    <value>9999999.99</value>
    <taxes_value>9999999.99</taxes_value>
    <comment>CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC</comment>
  </Tran>
  ...
</XYZ_Trans>
```

Your job is to create a query that extracts information from tables TRANSACTIONS and SALES\_TYPES to present them in a XML document similar to the one above. The answer for this exercise can be found at **Appendix A**.

1. Using **Data Source Explorer** execute the following statements, click **New SQL Script** button then select **OTADB** and then click **Finish**. Create a query that returns all the data (id, value, taxes and sales description) you need for creating the XML document.

```
SELECT <columns>
FROM XMLLAB.TRANSACTIONS tr, XMLLAB.SALES_TYPES st
WHERE <join_condition>
```

2. The next step is to use the **XMLGROUP** function to present the same data as an XML document.

```
XMLGROUP( <col1> as "alias", <col2> as "alias2", ...
           [OPTION ROW "row_element_name" ROOT "root_element_name"] )
```

*Note: elements between brackets [...] are optional.*

Using the XMLGROUP function you are able to easily create an XML document using all rows from a SELECT statement. What if you need to have 1 XML document for each row instead?

Consider now that you need to create an XML document containing the total sales and taxes for each type of sale. Each XML should look like the example below:

```
<Sales>
  <type>1</type>
  <value>007043.22</value>
  <taxes_value>0044242.98</taxes_value>
</Sales>
```

3. Using Command Line, create a query that returns all the data (sales type id, sales type description, sum of all sales, sum of all taxes) you need.

```
SELECT <columns>
FROM XMLLAB.TRANSACTIONS
GROUP BY <column>
```

4. The next step is to use the **XMLROW** function to create the XML documents from your data.

```
XMLROW( <col_exp1> as "alias", <col_exp2> as "alias2", ...
        [ OPTION ROW "row_element_name" [AS ATTRIBUTES] ] )
```

*Note: elements between brackets [...] are optional.*

5. Alternatively, you could use attributes in order to create smaller XML documents. The resulting XML documents should look like the one below:

```
<Sales type="1" value="007043.22" taxes_value="0044242.98">
</Sales>
```

*Hint: use the AS ATTRIBUTES option.*

*Please use the remainder of the time to work on a selection of the query questions included with this lab instruction.*

*Sample answers to these questions are also provided as a reference at the back.  
Please do attempt to write these queries before looking at the answers ☺ Enjoy*

---

## 6. Query Questions

### 6.1 XQuery and SQL/XML

#### Query 1

- Use XQUERY and db2-fn:xmlcolumn to retrieve all the XML documents
  - ▶ Table XMLCUSTOMER
  - ▶ See Cheat Sheet for example of how to output all XML documents from a table using XQuery.

#### Query 2

- Use XQUERY and db2-fn:sqlquery to retrieve all the XML documents
  - ▶ Table XMLCUSTOMER
  - ▶ Output should be identical to Query 1
    - ▶ See Cheat Sheet for example of how to output all XML documents in a table using SQL

#### Query 3

- Use XQUERY and db2-fn:xmlcolumn to retrieve all the names of customers where the city is “Toronto”.
  - ▶ See Cheat Sheet for example of applying a predicate on an element.

#### Query 4

- Count the number of customers with “Toronto” as the city. Use XQUERY and db2-fn:xmlcolumn and count function.
  - ▶ Table XMLCUSTOMER
  - ▶ Form of query becomes
   
xquery count(“whatever it is you want to count”)
  - ▶ Output with XML header information and without XML header information
  - ▶ HINT: look at command line parameters to see how to remove XML header. Use command “db2 list command options” to see what options are available

#### Query 5

- Find the customer name that has a “work” phone number of “905-555-7258”
  - ▶ Use XQuery

#### Query 6

- Use an XQUERY FLWOR (using just “for” and “return”) statement to list all the customer names.

#### Query 7

- Use an XQUERY FLWOR statement to list all the customer names in alphabetical order.

#### Query 8

- Use an XQUERY FLWOR statement to list all the customer names in alphabetical order but strips out all the XML tags.
  - ▶ Hint: xs:string(...) might be helpful here.
  - ▶ Try to use xs:string only once in the FLWOR statement.
    - Hint: assigning things to variables might help:  
let \$variable := something

#### Query 9

- Use an XQUERY FLWOR statement to list all the customer names in alphabetical order along with their address. Include the XML tags in the output.
- Hint: return (one thing, another thing) might be helpful here.

**Query 10**

- Use an XQUERY FLWOR statement to list all the customer names in alphabetical order along with their address. Only list the customers who live in the city of Markham. Include the XML tags in the output.
- Hint: You knew there was a reason for the “W” in FLWOR.

**Query 11**

- Use an XQUERY FLWOR statement to list all the product names ordered by price. Do not include the XML tags in the output.

**Query 12**

- Type in this XMLTable example (and get it to work).
  - ▶ Run this without the –x parameter to DB2 so you see the column names in the output in DB2 CLP
  - ▶ Hint: watch the quotes!

```
SELECT X.*  
FROM XMLCUSTOMER C,  
      XMLTable('$cu/customerinfo' PASSING C.INFO as "cu"  
              COLUMNS  
              "NAME"      CHAR(20)    PATH 'name',  
              "STREET"    CHAR(20)    PATH 'addr/street',  
              "CITY"      CHAR(20)    PATH 'addr/city'  
          ) AS X;
```

**Query 13**

- Add a column for country to the previous XMLTable example.

**Query 14**

- Modify the previous example to output only the name, city and country columns, but don't change the columns in the XMLTable definition.
- Select only the rows where the country is Canada.

## 6.2 XQuery with Update Facility

**Query 15**

- Modify the zip code of customer with CID = 1000 to “90210”.
- Save the output to the database.

**Query 16**

- Insert a <phone type="home">111-222-3333</phone> node after the “addr” of customer where CID = 10000
- Save the output to the database.

**Query 17**

- Insert a new element called <email> which contains the customer's email address amir@yahoo.com for customer where CID=1000
- Save output to database

## 7. Cheat Sheet

### 7.1 SQL/XML

#### Functions

- XMLQUERY – executes an XQuery and returns the result sequence
- XMLEXISTS – evaluates an XQuery predicates, returns TRUE/FALSE
- XMLTABLE – executes an XQuery, returns the result sequence as a relational table (if possible)
- XMLPARSE – parses character/BLOB data, produces XML value
- XMLVALIDATE – validates XML value against an XML schema

#### EXAMPLE 1:

```
SELECT XML2CLOB(
    XMLEMENT(NAME "Department",
        XMLATTRIBUTES( e.dept AS "name" ),
        XMLAGG( XMLEMENT( NAME "emp", e.firstname ) )
    )
) AS "dept_list"
FROM employee e
WHERE .....
GROUP BY e.dept
```

#### EXAMPLE 2:

```
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"
```

### 7.2 XQuery

#### EXAMPLE:

```
XQUERY db2-fn:xmlcolumn("dept.deptdoc")
```

### 7.3 XQuery with embedded SQL

#### EXAMPLE:

```
XQUERY db2-fn:sqlquery( 'select deptdoc
    from dept
    where deptID = "PR27" ' )...
```

### 7.4 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

**EXAMPLE:**

```

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>

```

## 7.5 Examples on How to Return Different Outputs

**EXAMPLE 1:**

```

xquery
for $d in xmlcolumn('deptdoc')/dept
where $d/@bldg = 101
return <namelist>
{$d/employee/name}
</namelist>

```

**RESULTS 1:**

```

<namelist>
<name>John Doe</name>
<name>Peter Pan</name>
</namelist>

```

**EXAMPLE 2:**

```

xquery
for $d in xmlcolumn('deptdoc')/dept
where $d/@bldg = 101
return $d/employee/name

```

**RESULTS 2:**

```

<name>John Doe</name>
<name>Peter Pan</name>

```

**EXAMPLE 3:**

```

xquery
for $d in xmlcolumn('deptdoc')/dept
where $d/@bldg = 101
return $d/employee/name/text()

```

**RESULTS 3:**

```

John Doe
Peter Pan

```

## 7.6 Nested FLWOR Expression

```

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>
```

## 7.7 XPath Example

### XML Document:

```
<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>
```

### XPath Examples:

XPath	Result
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>John Doe</name> <name>Peter Pan</name>
/dept/employee/name/text()	John Doe Peter Pan

## 7.8 XQuery – XML Update Facility

### Typical Syntax:

```
transform
  copy $new := ...
  modify ...$new...
return $new
```

## 7.9 XQuery: Updating Document in the database

```
update xmlcustomer
set info = xmlquery( 'transform
  copy $new := $i
  modify do replace value of $new/customerinfo/addr/pcode-zip with 90111
  return $new'
passing info as "i")
where cid = 1000
```

## 7.10 XQuery: Updating Document on the fly

```
select xmlquery( 'transform
  copy $new := $i
  modify do replace value of $new/customerinfo/addr/pcode-zip with 90111
  return $new'
passing info as "i")
from xmlcustomer
where cid = 1000
```

## 7.11 XQuery: Deleting a node

```
update xmlcustomer
set info = xmlquery( 'transform
  copy $new := $i
  modify do delete $new/customerinfo/phone
  return $new'
```

```
passing info as "i")
where cid = 1000
```

## 7.12 XQuery: Insert a node at the end of a document.

```
update xmldocument
set info = xmlquery( 'transform
  copy $new := $i
  modify do insert
    <phone type="cell">777-555-3333</phone>
  into $new/customerinfo
  return $new'
passing info as "i")
where cid = 1000
```

## 7.13 XQuery: Insert a node before another.

```
update xmldocument
set info = xmlquery( 'transform
  copy $new := $i
  modify do insert
    <phone type="cell">777-555-3333</phone>
  before $new/customerinfo/addr
  into $new/customerinfo
  return $new'
passing info as "i")
where cid = 1000
```

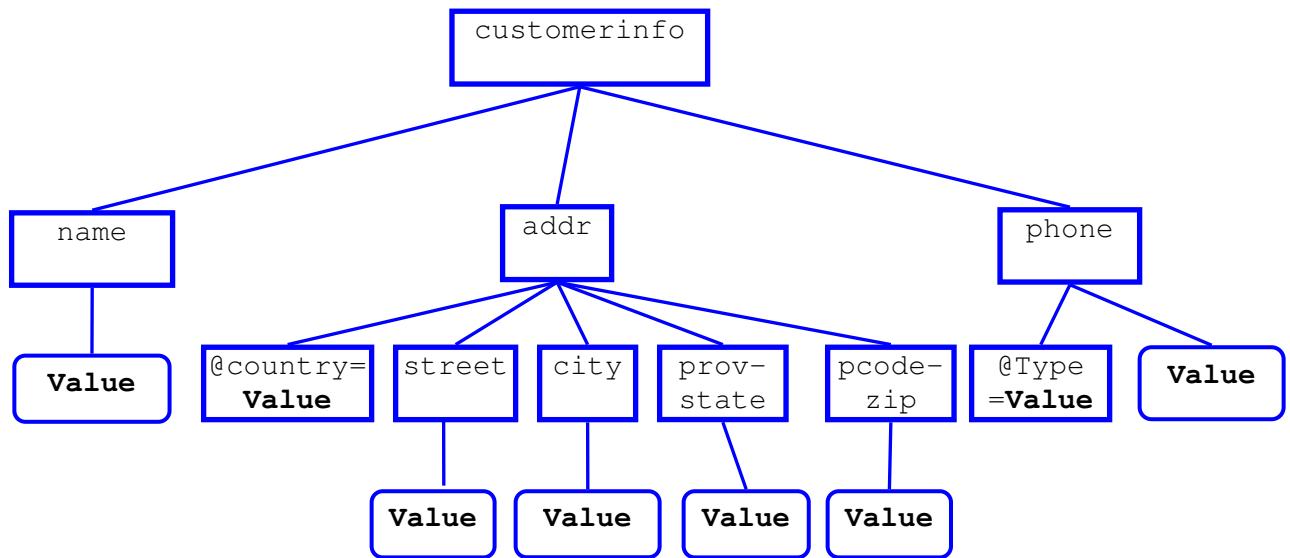
## 7.14 XQuery: Renaming an element

```
update xmldocument
set info = xmlquery( 'transform
  copy $new := $i
  modify do rename $new/customerinfo/addr as "address" before
  $new/customerinfo/addr
  into $new/customerinfo
  return $new'
passing info as "i")
where cid = 1000
```

# 8. Structure of XMLCUSTOMER Table XML Documents

## 8.1 Tree Structure – Example 1

Table: XMLCUSTOMER  
Column: INFO



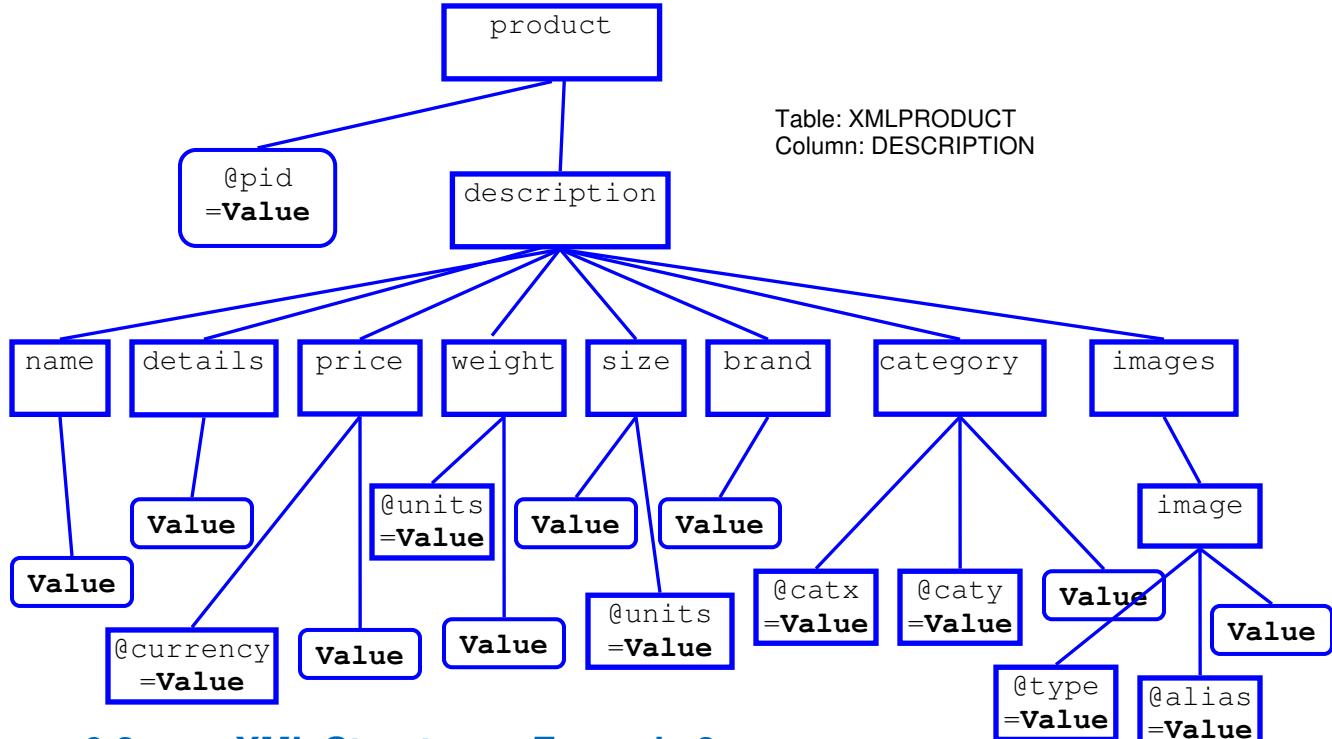
## 8.2 XML Structure – Example 1:

```

<?xml version="1.0" encoding="UTF-8" ?>
<customerinfo>
    <name>Value</name>
    <addr country="Value">
        <street>Value</street>
        <city>Value</city>
        <prov-state>Value</prov-state>
        <pcode-zip>Value</pcode-zip>
    </addr>
    <phone type="Value">Value</phone>
</customerinfo>
  
```

## 9. Structure of XMLPRODUCT Table XML Documents

### 9.1 Tree Structure – Example 2:



### 9.2 XML Structure – Example 2:

```
<?xml version="1.0" encoding="UTF-8" ?>
<product pid="Value">
  <description>
    <name>Value</name>
    <details>Value</details>
    <price currency="Value ">Value</price>
    <weight units="Value">Value</weight>
    <size units="Value">Value</size>
    <brand>Value</brand>
    <category catx="Value" caty="Value">Value</category>
    <images>
      <image type="Value" alias="">Value</image>
      <image type="Value" alias="">Value</image>
    </images>
  </description>
</product>
```

---

## 10. Answers to Queries

Keep in mind that these answers are for reference only. In a lot of cases, there are multiple ways to get the same answer.

### 10.1 Answer to XQuery and SQL/XML with no Updates

**Query 1**

```
XQUERY db2-fn:xmlcolumn ("XMLCUSTOMER.INFO")
```

**Query 2**

```
XQUERY db2-fn:sqlquery("select info from xmlcustomer")
```

**Query 3**

```
XQUERY db2-
fn:xmlcolumn('XMLCUSTOMER.INFO')/customerinfo[addr/city='Toronto']/name
```

*or*

```
XQUERY db2-
fn:xmlcolumn('XMLCUSTOMER.INFO')/customerinfo/addr[city='Toronto']/../name
```

*or*

```
XQUERY db2-
fn:xmlcolumn('XMLCUSTOMER.INFO')/customerinfo/addr/city[.='Toronto']/../../name
```

**Query 4**

```
XQUERY count(db2-fn:xmlcolumn('XMLCUSTOMER.INFO')/customerinfo[addr/city
='Toronto'])
```

**Query 5**

```
XQUERY db2-fn:xmlcolumn('XMLCUSTOMER.INFO')/
customerinfo/phone[@type='work' and text()='905-555-7258']/../name
```

**Query 6**

```
xquery
for $customer in db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo
return $customer/name
```

**Query 7**

```
xquery
for $customer in db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo
order by $customer/name
return $customer/name
```

**Query 8**

```
xquery
for $customer in db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo
let $cname := xs:string($customer/name)
order by $cname
return $cname
```

*or*

```
xquery
for $customer in db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo
order by $customer/name
return xs:string($customer/name)
```

**Query 9**

```
xquery
for $customer in db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo
order by $customer/name
return <RETURN>
{$customer/name, $customer/addr}
</RETURN>
```

**Query 10**

```
xquery
for $customer in db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo
where $customer/addr/city = 'Markham'
order by $customer/name
return ($customer/name, $customer/addr)
```

**Query 11**

```
xquery
for $prod in db2-fn:xmlcolumn("XMLPRODUCT.DESCRIPTION")/product/description
let $pname := xs:string($prod/name)
let $pprice := xs:string($prod/price)
order by $pprice
return ($pname, $pprice)
```

**Query 12**

```
SELECT X.* 
FROM XMLCUSTOMER C,
      XMLTable('$cu/customerinfo' PASSING C.INFO as "cu"
      COLUMNS
      "NAME"          CHAR(20)      PATH 'name',
      "STREET"        CHAR(20)      PATH 'addr/street',
      "CITY"          CHAR(20)      PATH 'addr/city'
      ) AS X
```

**Query 13**

```
SELECT X.* 
FROM XMLCUSTOMER C,
      XMLTable('$cu/customerinfo' PASSING C.INFO as "cu"
      COLUMNS
      "NAME"          CHAR(20)      PATH 'name',
      "STREET"        CHAR(20)      PATH 'addr/street',
      "CITY"          CHAR(20)      PATH 'addr/city',
      "COUNTRY"       CHAR(20)      PATH 'addr/@country'
      ) AS X
```

**Query 14**

```
SELECT X.NAME, X.CITY, X.COUNTRY
FROM XMLCUSTOMER C,
      XMLTable('$cu/customerinfo' PASSING C.INFO as "cu"
      COLUMNS
      "NAME"          CHAR(20)      PATH 'name',
      "STREET"        CHAR(20)      PATH 'addr/street',
      "CITY"          CHAR(20)      PATH 'addr/city',
      "COUNTRY"       CHAR(20)      PATH 'addr/@country'
      ) AS X
```

```
WHERE X.COUNTRY = 'Canada'
```

## 10.2 Answers to Queries with XML Updates

### Query 15

```
UPDATE XMLCUSTOMER
SET INFO = XMLQUERY(`transform
copy $new := $i
modify do replace value of
$new/customerinfo/addr/pcode-zip
with 90210
return $new'
PASSING INFO AS "i")
WHERE CID=1000
```

### Query 16

```
UPDATE XMLCUSTOMER
SET INFO = XMLQUERY(`transform
copy $new := $i
modify do insert
<phone type="home">111-222-3333</phone>
after $new/customerinfo/addr
return $new'
PASSING INFO AS "i")
WHERE CID=1000
```

### Query 17

```
UPDATE XMLCUSTOMER
SET INFO = XMLQUERY(`transform
copy $new := $i
modify do insert <email>amir@yahoo.com</email>
into $new/customerinfo
return $new'
PASSING INFO AS "i")
WHERE CID=1000
```

## Appendix A

### Query #1

```
SELECT tr.ID, tr.TOTAL, tr.TAXES, st.DESCRIPTION
FROM XMLLAB.TRANSACTIONS tr, XMLLAB.SALES_TYPES st
WHERE tr.SALE_TYPE = st.SID
```

### Query #2

```
SELECT XMLGROUP (
    tr.ID AS "id",
    tr.TOTAL AS "value",
    tr.TAXES AS "taxes_value",
    st.DESCRIPTION AS "comment"
    OPTION ROW "Tran" ROOT "XYZ_Trans")
FROM XMLLAB.TRANSACTIONS tr, XMLLAB.SALES_TYPES st
WHERE tr.SALE_TYPE = st.SID
```

#### Query #3

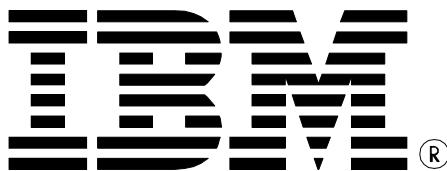
```
SELECT sale_type, sum(total), sum(taxes)
FROM xmllab.transactions
GROUP BY sale_type
```

#### Query #4

```
SELECT xmlrow(sale_type as "type",
    sum(total) as "value",
    sum(taxes) as "taxes_value"
    OPTION ROW "Sales")
FROM xmllab.transactions
GROUP BY sale_type
```

#### Query #5

```
SELECT xmlrow(sale_type as "type",
    sum(total) as "value",
    sum(taxes) as "taxes_value"
    OPTION ROW "Sales" AS ATTRIBUTES)
FROM xmllab.transactions
GROUP BY sale_type
```



© Copyright IBM Corporation 2011  
All Rights Reserved.

IBM Canada  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

Printed in Canada  
01/01/2011

IBM, IBM (logo), DB2, WebSphere, and WebSphere Application Server are trademarks of International Business Machines Corporation in the United States, other countries, or both.

VMware is a trademark or VMware Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION  
PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY  
OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING,  
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-  
INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A  
PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

The information in this publication is provided AS IS without warranty. Such information was obtained from publicly available sources, is current as of July 2009, and is subject to change. Any performance data included in the paper was obtained in the specific operating environment and is provided as an illustration. Performance in other operating environments may vary. More specific information about the capabilities of products described should be obtained from the suppliers of those products.