DB2 9.5 for Linux, Unix and Windows Fundamentals

Michele Benedetti   Certified IT Specialist
IBM Software Group Technical Sales
Introduction to DB2 9.5 for Linux, Unix and Windows

- Basic Terminology
- General Architecture: Instances & Databases
- Accessing Remote Databases: clients, drivers and catalogs
- Managing Storage
- Logging, Backup & Recoveries
- Managing Performances
- Managing Security, users & groups
- Autonomics Wrap-up
- High Availability
- pureXML®: native XML storage manager
- Monitoring
- Miscellaneous
- DB2 9.5 Editions
Informazione come Servizio

Processi, Applicazioni, Persone, accedono con difficoltà e in modo scoordinato alle informazioni che servono:
- inefficienza, duplicazione di codice…
- non univocità delle informazioni

Information Services
- Conformi alla SOA
- Riutilizzabili
- Semanticamente consistenti
- Esempi:
  - verifiche di correttezza
  - trasformazioni di formato
  - arricchimento, aggregazione
  - sincronizzazione
  - query federate
  - …

L’informazione corretta, univoca, fornita come servizio “su richiesta”

EIM

Info 2.0
IBM Leads in Data Server Innovation

Innovation Milestones

1968
- First Hierarchical Data Server
- IBM designs IMS starting in 1966 for the Apollo space program

1980
- First IBM Relational Data Server
- IBM releases RDBMS for System/38 implementing the Relational model first published by Dr. Edgar Codd

2006
- First Multi-Structured Data Server
- DB2 9 first to support both relational and XML structures managed by single data server

Continuous IBM innovation
IBM Intellectual property – Patented Technology Leadership

pureXML® is one out of the 70 patented technologies part of DB2 9 “Viper”
Basic Terminology
Basic Terminology: Relational Model

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>EXTENSION</th>
<th>MANAGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John S</td>
<td>54213</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Susan P</td>
<td>59867</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>Jennifer L</td>
<td>59415</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>Andrew J</td>
<td>55935</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>Michael B</td>
<td>52137</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>Jeremy W</td>
<td>50603</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>Leah E</td>
<td>58963</td>
<td>N</td>
</tr>
</tbody>
</table>
Basic Terminology: SQL

- DB2 9.5 supports SQL for querying relational data
- Also supports XQUERY for querying XML data (see more later)
- Supports “mixed” SQL/XML and XQUERY/SQL queries (see more later)
- 4 types of SQL statements:
  1. Data Definition Language (DDL) (create, drop, alter objects)
  2. Data Manipulation Language (DML) (insert, update, delete….)
  3. Data Control Language (DCL) (grant, revoke)
  4. Transaction Control Language (TCL) (commit, rollback…)
General Architecture: Instances & Databases
DB2 Instances

- **Instances**
  - Stand-alone DB2 environment
  - All instances share the same executable binary files
  - Each instance has its own configuration

- **DB2 Administration Server**
  - Special instance that responds to requests from the DB2 Administration Tools and the Configuration Assistant
DB2 Architecture

- **Instance = DB2 Engine (db2sysc)**
  - Threaded model in which one main engine process exists (db2sysc) in memory
  - Engine dispatchable units (EDU) exist as threads and perform work
  - Benefits of the threaded model:
    - Increased performance
    - Decreased memory usage

- **Connections = DB2 Agents (db2agent)**
  - All database requests are performed by db2agent EDU’s on behalf of an application
  - The DB2 engine keeps a pool of agents available to service requests
  - Two major types of agents: Coordinator Agents, Subagents
DB2 9.5 Processing Model

New UNIX/Linux Processing Model
The blue rounded rectangle represents a single OS process for the instance ('db2sysc'); The circles within it represent OS threads.
### DB2 Instances

- Some sample commands for working with instances
- Note that most commands here can be performed in Control Center

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>db2start</td>
<td>Start the default instance</td>
<td>db2start</td>
</tr>
<tr>
<td>db2stop</td>
<td>Stop the current instance</td>
<td>db2stop -f</td>
</tr>
<tr>
<td>db2icrt</td>
<td>Create an instance</td>
<td>db2icrt –u db2fenc1 db2inst1</td>
</tr>
<tr>
<td>db2idrop</td>
<td>Drop an instance</td>
<td>db2idrop –f db2inst1</td>
</tr>
<tr>
<td>db2ilist</td>
<td>List all instances</td>
<td>db2ilist</td>
</tr>
<tr>
<td>db2imigr</td>
<td>Migrate an instance after upgrading DB2</td>
<td>db2imigr –u db2fenc1 db2inst1</td>
</tr>
<tr>
<td>db2iupdt</td>
<td>Update an instance after installation of a fix pack</td>
<td>db2iupdt –u db2fenc1 db2inst1</td>
</tr>
</tbody>
</table>
## DB2 Instances: Instance and Database Configuration

### Viewing and Changing Instance Configuration:

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Database Manager Settings</td>
<td>db2 get dbm cfg show detail</td>
</tr>
<tr>
<td>Change a Database Manager Setting</td>
<td>db2 update dbm cfg using health_mon off</td>
</tr>
</tbody>
</table>

### Viewing and Changing Database Configuration:

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Database Settings</td>
<td>db2 get db cfg for testdb</td>
</tr>
<tr>
<td></td>
<td>db2 connect to testdb</td>
</tr>
<tr>
<td></td>
<td>db2 get db cfg show detail</td>
</tr>
<tr>
<td>Change a Database Setting</td>
<td>db2 update db cfg using logprimary 10</td>
</tr>
</tbody>
</table>
Memory Architecture (with background processes)

DB2 Linux/UNIX Example

---

**DB2 Application**

- **App. Global Memory**
  - `app_ctl_heap_sz` (DPF)

- **App. Private Memory**
  - `appheapsz` (non DPF)
  - `sortheap`
  - `stat_heap_sz`
  - `stmthep`
  - `agent_stack_sz`
  - `query_heap_sz`
  - `rqioblk`
  - `java_heap_sz`

- **App. Shared Memory**
  - `aslheapsz`
  - `dir_cache`

- **db2agent**

---

**DB2 Instance**

- **Monitor Heap**
  - `mon_heap_sz`

- **Audit Buffer**
  - `audit_buf_sz`

- **FCM Buffs (DPF)**
  - `fcm_num_buffers`

---

**DB2 Database**

- **Database Heap**
  - `logbufsz`
  - `catalog_cache_sz`

- **Utility Heap**
  - `util_heap_sz`

- **Buffer Pools**
  - 4k buffer pool
  - 8k buffer pool
  - 16k buffer pool
  - 32k buffer pool

- **Lock List**
  - `locklist`

- **Sorting**
  - `sheapthres_shr`
  - `sortheap`

- **Package Cache**
  - `pckcachesz`

- **Other memory areas**
  - `db2pfch`
  - `db2resyn`
  - `db2tcpcm`
  - `db2sysc`
  - `db2syslog`
  - `...others`

---

© 2008 IBM Corporation
**DB2 Directory Structure**

**Windows Example**

- **Instance Name**
  - `DB2`
  - `\node0000`
  - `\sql00001`

- **Database ID**
  - `/SQLOGDIR`
  - `/db2event`

- **Default LOG directory**
- **Monitor data for deadlocks**

- **TEMPSPACE1 table space (always created) containers**
- **USERSPACE1 table space (always created) containers**
- **SYSCATSPACE1 (catalog/object cross reference data (always created) containers**

- **SQLDBCON**
  - SQLDBCONF db configuration parameter file
  - Db2rhist.asc db recovery history file*.IN1 (contains index table data), etc.

- **/program files**
  - `\sqlib`
  - `\bin`

- **db2start.exe**
- **db2stop.exe**
- **db2cmd.exe**, etc.
Instance Directory (Windows)

- \Document and Settings\all users\application\data\IBM
  - \DB2\DB2COPY1: Copy of the installation Code (may be more than one of different levels)
  - \DB2: Instance Name
  - \SQLDBDIR: Local System Database Directory / Catalog
  - \SQLNODIR: Node Directory / Catalog
  - ......: Other system and temp dirs
    - DB2DIAG.LOG: Diagnostic file (text)
    - *.dmp: dump file (in case of fatal errors)
    - *.trc: trace files in case of fatal errors
    - etc.:
DB2 Directory Structure
Linux-Unix Example

Instance Name

Partition number (note: in ESE non DPF databases are a single node implementation)

Database ID

Default LOG directory

Monitor data for deadlocks

TEMPSPACE1 table space (always created) containers

USERSPACE1 table space (always created) containers

SYSCATSPACE1 (catalog/object cross reference data (always created) containers

SQLDBCON
SQLDBCONF db configuration parameter file
Db2rhist.asc db recovery history file*.IN1 (contains index table data), etc.

/opt

/ibm

/db2

/db2inst1

/node0000

/sql00001

/SQLOGDIR

/db2event

/SYSCATSPACE1

/V9.5

/bin

/instance

/java

/instance

/db2inst1

/node0000

/sql00001

/SQLOGDIR

/db2event

/SYSCATSPACE1

/V9.5

/bin

/instance

/java

/instance
Instance Directory (Linux-Unix)

```
/home/db2inst1

/SQLLIB

/SQLDBDIR

Local System Database Directory / Catalog

/SQLNODIR

Node Directory /Catalog

/cfg

Db2profile
profile file for user db2inst1 (contains environment settings)

/db2dump

Instance directory for diagnostic
(db2diag.log) trace and dump files
```

Instance Name = Instance Owner User
Registry / Environment Variables

- **DB2 Registry**: used to enable/disable specific functionality/behaviours
- **Work at Environment, Server, Instance level**
- **Via “db2set” command**
- **Examples:**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>db2adminserver</td>
<td>Specifies which instance runs the admin. server</td>
</tr>
<tr>
<td>db2comm</td>
<td>Started communications manager</td>
</tr>
<tr>
<td>db2include</td>
<td>Path to include in SQL searches</td>
</tr>
<tr>
<td>db2instance</td>
<td>Current instance</td>
</tr>
<tr>
<td>db2instdef</td>
<td>Default instance</td>
</tr>
<tr>
<td>db2owner</td>
<td>Instance owning machine</td>
</tr>
<tr>
<td>db2slogon</td>
<td>Enables secure logon</td>
</tr>
<tr>
<td>db2path</td>
<td>Directory where product is installed</td>
</tr>
<tr>
<td>db2system</td>
<td>Server name id</td>
</tr>
</tbody>
</table>
Accessing Remote Databases: clients, drivers and catalogs
DB2 Clients

DB2 9 API:
- embedded SQL
- CLI, ODBC, OLE-DB, ADO.Net
- JDBC, SQLJ

protocollo DRDA usato da ogni client per accedere a:
- DB2 UDB for LUW: non richiede DB2 Connect
- DB2 UDB for i-Series: richiede DB2 Connect
- DB2 UDB for OS/390 & z/OS: richiede DB2 Connect
- DB2 for VSE & VM: richiede DB2 Connect

✔ funzionalità DRDA AR inclusa in ogni DB2 Client
  - Solo Drivers (diversi pkg a seconda della tipologia)
  - IBM Client (contiene TUTTI I drivers + CLI)
  - Run Time client (per amministrazione)

✔ funzionalità DRDA AS inclusa in ogni DB2 Server (compresi mainframes e i-Series)
DB2 Catalogs (directories)

- **On a Server:**
  - Local System Database Directory
    - Lists local & Remote databases accessible by this server
  - Local Volume Database Directory
    - On each volume in which a DB exist, describe/list available local DBs

- **On a Client (CLI) and on a Server accessing remote DBs:**
  - Local System Database Directory
    - Lists only remote DBs accessible by the client

- **On both Server and Client (CLI):**
  - Node Directory
    - Lists and describe Remote Servers/Instances hosting a DB
      - Specifies Protocol (TCPIP) ADDRESS/HOSTNAME, Port to be accessed (Server and/or Service) or TCPIP Service

  - Possible in Control Center

- **Examples:**
  - `db2 catalog tcpip node db2node remote SERVER1 server 50001`
  - CATALOGUED IN THE LOCAL SYSTEM DB DIRECOTRY
  - THIS DEFINITION MUST EVENTUALY RELY ON A NODE/SERVER DEFINITION
  - `db2 catalog tcpip node db2das remote SERVER1`

- **Possible commands:**
  - `db2 catalog admin tcpip node db2das remote SERVER1`
Sample Catalog Commands

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>db2 catalog/uncatalog database/node</td>
</tr>
<tr>
<td>db2 list database directory</td>
</tr>
<tr>
<td>db2 list node directory</td>
</tr>
<tr>
<td>db2 list admin directory</td>
</tr>
<tr>
<td>......</td>
</tr>
</tbody>
</table>
Tools
DB2 Control Center

- Centers
- Commands and scripts execution
- Create Object Wizards
- Backup Execution and Scheduling
- DBs and Instance cfg
- Activity Monitor (performance inspection)
- SQLStmts fed by Activity Monitor
- Visual Explain (also fed by Activity Monitor or Command Editor)
- Graphical Monitor and Suggestions

And many more such as:
- Recovery Wizard
- Log Configuration Wizard
- Task Scheduling
- License management
- Lock Inspection
- Data Partitioning Wizard
- Data Movement Wizards
- Data Maintenance Wizard
- Autobackup Wizard
IBM Data Studio

Eclipse based: other tools available as plug-ins

Perspectives
Object Properties
ER Diagramming
Snapshot data Skews
Stored Procedure Builder and Debugging
Data Web Services in 5 simple steps

1. Create and Test Queries or Stored Procedures
2. Create Service
3. Drag ‘n Drop Resources
4. Deploy Service
5. Test and Deliver
IBM Data Studio Administrative Console: Quick & Easy Problem Determination

Heatchart – Overall Health Status

Where are the most important hotspots that need my attention?

Dashboard – Adhoc Investigation

Something doesn't seem quite right. I wonder what's happening?

Alert List – Historical Investigation

What happened when I was out for lunch? ... Away for weekend?

Recommendations – Root Cause Analysis

Guide me to the root cause and help me fix it properly; I need to know all the relevant info to make the best decision.
Managing Storage
Tablespaces

- **Tablespaces**
  - Logical layer to store data as link between
    - logical objects (tables)
    - Physical devices (disks): containers

- **Three kinds:**
  - SMS: System Managed Space.
    - Containers = directories
    - Average Performances
    - No maintenance (grow/shrink on need automatically)
  - DMS: Database Managed Space
    - Containers = files or raw devices
    - High to Best Performances
    - Requires manual intervention when out of disk space
  - Automatic Storage
    - Containers = files
    - High Performances
    - Autoresizing (grow/shrink) when needed based on predefined policies
    - Manual Intervention only to add NEW devices/disks when automatic storage pool is exhausted
Tablespace Usage

- **Possibilities:**
  - Regular
  - Large

- **Type of usage:**
  - System
  - System Temporary
  - User
  - User Temporary
For all System/User (temporary or not) both Regular and Large:

- Must specify a size for the data page at TBSPC level
  - 4,8,16 or 32kb
- Row must fit into page. If not, must use BLOB
- Associate ONE Buffer Pool (cache memory for I/O)
  - Same page size as Tablespace associated
  - Can make use of Self Tuning Memory Manager
DB2 Storage: Buffer Pools

- **Buffer Pool:**
  - Area of main memory used to cache table and index data
  - Can decrease time to access data
  - Each database must have at least one buffer pool
  - Tuning buffer pools effortless with Self-Tuning Memory Manager (STMM)
Tablespace Capacity

Old Tablespace Design

New Large Tablespace Design

Page size
- 4KB
- 8KB
- 16KB
- 32KB
- 64GB
- 128GB
- 256GB
- 512GB

Table space size
- 4x10^9 Rows
- 10M
- 255

Row ID (RID) 4 Bytes

For tables in LARGE table spaces (DMS only)
Also all SYSTEM and USER temporary table spaces
### DB2 Storage: Table Spaces

- DB2 performs I/O grouping bits in sets of Data Pages. Data page have a fixed sized of 4K, 8K, 16K, or 32K.
- Table spaces are created specifying one page size out of 4K, 8K, 16K, or 32K.
- Page size limits the data that can be stored in the table space. One row MUST fit page size, except for Binary Data.
- Default page size is 4KB.
- Inside the page, row is identified by ROWID.

<table>
<thead>
<tr>
<th>Description</th>
<th>4K page size limit</th>
<th>8K page size limit</th>
<th>16K page size limit</th>
<th>32K page size limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of columns in a table</td>
<td>500</td>
<td>1012</td>
<td>1012</td>
<td>1012</td>
</tr>
<tr>
<td>Maximum length of a row including all overhead (bytes)</td>
<td>4005</td>
<td>8101</td>
<td>16293</td>
<td>32677</td>
</tr>
<tr>
<td>Maximum size of a table per DB partition in a regular table space (GB)</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>Maximum size of a large DMS table space, using the default 6 byte RID size (TB)</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>
DB2 Storage: Table Spaces Example
Logging, Backup & Recoveries
- Logging
- Circular vs Archive
- Backup & Logging
  - Backup
    - Cold
    - Warm
    - Partial
- Recovery
  - Restore vs Forward Recovery
DB2 Logging – Overview

- **synchronous write** on COMMIT/ROLLBACK
- **MINCOMMIT**
- **SOFTMAX**

- **asynchronous write** when triggered (chngpgs_thres hold)

- Log Buffer
- Buffer Pool
- db2loggr
- db2pclnr
- Online Active Log
- Database Files

© 2008 IBM Corporation
ARIES – Write-ahead Logging

- **ARIES (Algorithm for Recovery and Isolation Exploiting Semantics)** was invented in the IBM Almaden Research Center by Dr. C. Mohan

- **Write-ahead logging:**
  - Must force the log record for an update before the corresponding data page gets to disk
    - Guarantees Atomicity (all actions in a transaction happen, or none happen)
  - Must write all log records for a transaction before commit
    - Guarantees Durability (if a transactions commits, its effects persist)

- **How is it done:**
  - Each log record has a unique Log Sequence Number (LSN)
  - Each data page contains a pageLSN
  - System keeps track of flushed LSN
  - Before a page is written, ensure pageLSN <= flushed LSN

- **DB2 uses ARIES (Algorithm for Recovery and Isolation Exploiting Semantics) as the transaction recovery method supporting fine-granularity locking and partial rollbacks**
DB2 Logging – Circular Logging

1

"n"

PRIMARY

2

1

"n"

SECONDARY

3
DB2 Logging – Archival Logging

DB2 archive log file (db2uext2)

Configured via DB2 Logging Wizard (DB2 CC)

OFFLINE ARCHIVE - Archive moved from ACTIVE log subdirectory. (May also be on other media)

ACTIVE - Contains information for non-committed

ONLINE ARCHIVE - Contains information for committed. Stored in the ACTIVE log subdirectory.

12 13 14 15 16
Backup

- Used to protect data in case of physical/logical failure affecting them
- Basic protection different from HA & DR concepts
- Can be:
  - Full (entire DB is backed-up)
    - Cold (Full backup “offline”)
    - Warm (full backup online)
  - Partial (tablespace(s))
  - Incremental (only changes from latest activity)
    - Delta
    - Increment
- Produce an asset of type “file”
- Can be compressed
- Can be put to disk or tape
Warm backup

- **Need LOG ARCHIVE method**
  - Data is “copied” to disk WHILE applications are changing them
  - Changes are written to LOG files

- **Log files written WHILE backing up DB are PART of the Backup image file produced.**
Partial Backup

- **Partial (Incremental) backup types**
  - **Cumulative**
    backs-up everything since most recent full backup
  - **Delta**
    backs-up only data that changed since last backup (either full or incremental)
Automatic backup

- **Automatic backup was:**
  - Scheduled backup (fixed time)

- **Automatic backup for DB2 is:**
  - The possibility to tell “when” a backup is “needed”
    - How many times (a day, a week, a month….)
    - How many changes/log spaces
  - The possibility to tell “when” a backup can be taken
    - Provide a time-frame
      - Within which the backup must be executed
      - Outside of which the backup can be executed
  - Number of images to be kept on line
  - The same for log sequences
Recovery

- **Main assets:** backup Image(s) + Log Files (when required)
- **Main steps:**
  - Restore (of the backup image)
    - Rebuild the file system(s) onto which tables were stored
  - Forward Recovery (Roll Forward)
    - Mimic the changes accumulated in the Log Files “after” (and maybe “during”) the backup timeframe.
Forward Recovery

– Roll Forward (or Forward Recovery) when possible:
  • Optional: when Log Archive enabled and when restoring a Full Backup Offline
  • Mandatory: when Log Archive enabled and when restoring a Full Backup Online.
    – Log files contain eventually modified data during the timeframe of online backup execution.
    – Minimum: rollforward reading AT LEAST the log files written during online backup timeframe.
Forward Recovery: how far?

- **Two options:**
  - **TO END OF LOGS**
    - Mimic all the changes in ALL the log files DB2 can read
    - Beware of log sequences associated with the backup Image
  - **TO ISO TIME**
    - Mimic all the changes contained in ALL the log files DB2 can read AND up to a specified timestamp.
Managing Performance
Performance: an esoteric approach?

- Managing performance can be pure “esoterism”
- In the past, a very high level of knowledge was necessary
  - Valuable physical database design
  - Valuable parameter settings
    - At system level
    - At RDBMS level
  - Valuable application logic design
Take it easy with DB2: Automatic Optimizer

- **DB2 Learning Optimizer (LEO)**
  - Analyze Query received
  - Automatically Calculate the best access plan based on
    - Best practices algorithm “cost based”
    - Optimization classes (eg: simple transactional accesses vs complex business intelligence ones)
    - Statistics automatically collected onto tables and indexes
    - Use of materialized views, clustering of data (eg: multidimensional) etc
  - Automatically re-writes queries
- **No need of high skilled DBA (no “hints” required by DBA to “induce” a good behaviour).**
Take it easy with DB2: Self Tuning Memory Manager
Take it easy with DB2: Automatic Data Maintenance

- **For best performances, Optimzer needs up to date Statistics**
  - Can be automatically calculated by DB2 when needed
  - Impact on system is minimized (on line execution, throttling and policies)

- **For best performances, data in tables and indexes need to be reorganized**
  - Can be automatically done when needed
  - Impact on system is minimized (on line execution, throttling and policies)

- **In specific cases, use of MDC avoid data reorganization**
DB2 aids to manage performances

- **Automatic settings of peculiar parameters affecting performances**
  - At instance and/or database level
  - Related to I/O, Concurrency of workload (e.g.: n. of concurrent SQL Statements executed, nr. Of concurrent connections etc)

- **Advisors to enhance physical db design in terms of:**
  - Indexes needed for a specific workload
  - Materialized Query Tables: materialized views for complex aggregations
  - Multidimensional Clustering: re-define critical tables in a multidimensional (ipercube) fashion.
  - Analyze SQL cache
  - Analyze single Query behaviour
DB2 advisors

- **DB2 Activity Monitor:**
  - To produce reports to analyze SQL cache and feed Design Advisor

- **DB2 Design Advisor:**
  - To analyze the workload (eg: coming from Activity Monitor) and suggest:
    - Indexes
    - MQTs
    - MDCs

- **DB2 Visual Explain**
  - To analyze single query (eg: coming from Activity Monitor) behaviour
    - Graphical
    - Specify “costs” for each part of the query
    - Determine when indexes are used and why they are not
Materialized Query Tables

Pre-aggregated data store

- Cost of plan calculation shared on several queries
  - no MQT: Optimizer calculate aggregation for each query
  - with MQT: Optimizer calculate ONCE the aggregation and reuse plan each time
- Automatic MQT use by Optimizer
- In case of partitioned DB, MQT can be replied on all nodes

create table m_vendite as
(SELECT v_cid,
  sum((V_QTA * V_PREZZO)*(100 - V_SCONTO)/100) as venduto
  FROM DB2ADM.IVENDITE group by v_cid)
data initially deferred refresh deferred
enable query optimization maintained by system;
Row Compression

- Automatically Compress a Table without DBA intervention (Lempel-Ziv)
- Automatically create a Compression Dictionary for entire Table
- Data stay compressed in Buffer Pools, in Logs, not only on disk!!
- Less disk, less I/O, better Performance

<table>
<thead>
<tr>
<th>Query #</th>
<th>No compression</th>
<th>Row Compression</th>
<th>% diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>97.3</td>
<td>73.8</td>
<td>76%</td>
</tr>
<tr>
<td>2</td>
<td>9.8</td>
<td>7.9</td>
<td>81%</td>
</tr>
<tr>
<td>3</td>
<td>77.0</td>
<td>55.6</td>
<td>72%</td>
</tr>
<tr>
<td>4</td>
<td>93.3</td>
<td>48.5</td>
<td>52%</td>
</tr>
<tr>
<td>5</td>
<td>103.0</td>
<td>62.8</td>
<td>61%</td>
</tr>
<tr>
<td>6</td>
<td>14.8</td>
<td>9.3</td>
<td>63%</td>
</tr>
<tr>
<td>7</td>
<td>66.7</td>
<td>44.3</td>
<td>66%</td>
</tr>
<tr>
<td>8</td>
<td>230.8</td>
<td>184.0</td>
<td>80%</td>
</tr>
<tr>
<td>9</td>
<td>146.3</td>
<td>92.1</td>
<td>63%</td>
</tr>
<tr>
<td>10</td>
<td>102.6</td>
<td>56.1</td>
<td>55%</td>
</tr>
<tr>
<td>11</td>
<td>93.9</td>
<td>34.0</td>
<td>66%</td>
</tr>
<tr>
<td>12</td>
<td>105.4</td>
<td>52.4</td>
<td>50%</td>
</tr>
<tr>
<td>13</td>
<td>78.2</td>
<td>76.7</td>
<td>98%</td>
</tr>
<tr>
<td>14</td>
<td>6.8</td>
<td>5.9</td>
<td>74%</td>
</tr>
<tr>
<td>15</td>
<td>9.8</td>
<td>9.7</td>
<td>99%</td>
</tr>
<tr>
<td>16</td>
<td>13.3</td>
<td>13.4</td>
<td>101%</td>
</tr>
<tr>
<td>17</td>
<td>37.2</td>
<td>28.9</td>
<td>78%</td>
</tr>
<tr>
<td>18</td>
<td>278.6</td>
<td>269.0</td>
<td>97%</td>
</tr>
<tr>
<td>19</td>
<td>345.2</td>
<td>266.1</td>
<td>79%</td>
</tr>
<tr>
<td>20</td>
<td>44.1</td>
<td>31.6</td>
<td>69%</td>
</tr>
<tr>
<td>21</td>
<td>300.2</td>
<td>166.8</td>
<td>56%</td>
</tr>
<tr>
<td>22</td>
<td>6.0</td>
<td>5.3</td>
<td>88%</td>
</tr>
<tr>
<td>TS</td>
<td>2258.2</td>
<td>1597.9</td>
<td>71%</td>
</tr>
</tbody>
</table>
DB2 Flexible Data Partitioning for High Performance

- **DISTRIBUTE BY HASH**
- **PARTITION BY RANGE**
- **ORGANIZE BY DIMENSIONS**

Node 1
- **T1** Distributed across 3 database partitions
- TS1: Jan
  - North
    - East
    - West
  - South
- TS2: Feb
  - North
    - East
    - West
  - South

Node 2
- TS1: Jan
  - North
    - East
    - West
  - South
- TS2: Feb
  - North
    - East
    - West
  - South

Node 3
- TS1: Jan
  - North
    - East
    - West
  - South
- TS2: Feb
  - North
    - East
    - West
  - South

**World’s Richest Slice & Dice Capability**
No Partitioning

CREATE TABLE my_hybrid
    (A INT, B INT, C INT, D INT ...)
IN Tablespace A,
Distribute by Hash

Divide & Conquer Parallelism

CREATE TABLE my_hybrid (A INT, B INT, C INT, D INT ...) IN Tablespace A, DISTRIBUTED BY HASH (A)
CREATE TABLE my_hybrid
    (A INT, B INT, C INT, D INT ...
    )
IN Tablespace A, Tablespace B, Tablespace C
INDEX IN Tablespace B
DISTRIBUTE BY HASH (A)
PARTITION BY RANGE (B) (STARTING FROM (100) ENDING (300) EVERY (100))
Hash + Range + MDC
High density, High Value, Low IO Reads

CREATE TABLE my_hybrid
(A INT, B INT, C INT, D INT ...)
IN Tablespace A, Tablespace B, Tablespace C
INDEX IN Tablespace B
DISTRIBUTE BY HASH (A)
PARTITION BY RANGE (B) (STARTING FROM (100) ENDING (300) EVERY (100))
ORGANIZE BY DIMENSIONS (C,D)
Stable and Predictable Performances by predefining amount of System and DB2 resources to be used by workload
Locking and Concurrrency

- DB2 implements standard locking model
  - Row level not page level

- Implements standard ISO Isolation Levels
  - Proprietary “terminology” (from the lower to the higher level of control)
    - UR
    - CS
    - RS
    - RR

- CFR reference to understand how readers and writers must behave to guarantee the best compromise among:
  - Performances
  - Concurrency of access
  - Consistency of application logic
Managing Security, users & groups
Authentication

- Delegated by default to Operating System
- Can be done by Server (default) or Client (security issues here)
- Can be compliant with
  - LDAP/Active Directory
  - Kerberos
  - Data Encryption
- Optionally: via User exit (eg: Migrations)
There are 3 types of authorization in DB2:

- Authorities (Groups/Roles)
- Privileges (Grant/revoke on objects like Tables, Views, Packages, SP, Triggers..)
- LBAC credentials

Explicit authorities/privileges:
- Granted explicitly to the user, usually in the form of a GRANT statement

Implicit authorities/privileges:
- Granted to a group to which the user belongs, or to a role in which the user, the group, or another role is a member
DB2 Security: Authorities
### LBAC Query

```sql
SELECT * FROM EMP
WHERE SALARY >= 50000
```
Autonomics wrap-up
- LEO Optimizer and Automatic Query Rewrite
- Automatic Maintenance
- Self Tuning Memory Manager
- Automatic Configuration
- Self Healing and Automonitoring
- Automatic Storage
- Automatic Backup
- Automatic Log Sequence Maintenance
High Availability
The available Options

- **Via Scripts by use of**
  - On-line Split & Mirror functionality
  - Using Log File Shipping and Backup + File Transfer + Restore + Roll Forward

- **Relying on HA software built in in SO**
  - E.g: MS Cluster Services

- **Via use og HADR feature:** two DB copies kept in sync automatically

- In any of the above: 1 server is active 1 is stand-by (hot stand-by)

- Can have also Mutual Take Over (for two distinct workloads working on different DBs. One server at a time is active for each workload)
High Availability & Disaster Recovery (HADR)

- **Target Market**
  - Online commercial applications

- **Challenge**
  - 24 x 7 Availability
    - Failover in seconds
    - Disaster recovery

- **Solution: HADR**
  - Single solution handles
    - Ultra-fast failover
    - Local and remote site recovery

- **Value**
  - Business continuation
  - Tight integration; Very simple to use

---

Offsite Disaster Recovery

Onsite Warm Standby

More later
Synchronization modes

- This diagram shows when acknowledgements are received in Synchronous, Near Synchronous and Asynchronous modes.
pureXML®: native XML storage manager
pureXML®: native XML storage manager

- It’s IBM patented proprietary Technology
- It’s the only proprietary implementation as for XML
- “pureXML” is an IBM trademark
pureXML®: The XML Data Type

- SQL/XML introduces a new data type: XML

- Two XML type flavors:
  - Transient XML type
    - Can only be used internally
    - Needs to be converted to non-XML type before returning to client
    - XML type not exposed to user
    - Has already been supported by DB2 V8.x
    - XML is converted to string value using XML2CLOB or XMLSERIALIZE
  - Persistent XML type
    - Data type “XML” can be used for
      - Tables, Views, Functions, Procedures, …
    - Values of type XML can be stored in columns
    - Values of type XML are “visible” and, e.g., can be returned in queries
pureXML®: Native XML Storage

- DB2 stores XML in parsed hierarchical format (~DOM)

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

- Relational columns are stored in relational format (tables)

- XML columns are stored natively

- No XML parsing for query evaluation!
pureXML®: Native XML Storage VS XML Digital Signatures

- **pureXML preserves digital signatures**
  - Because XML is stored natively…
  - Digitally signed XML documents can be inserted in DB2, retrieved, and the signatures verified

- **Details:**
  - Creation of a signature requires canonicalization of the doc
  - W3C spec: [http://www.w3.org/Signature/](http://www.w3.org/Signature/)
  - Signatures can be created on all or part of a document
  - There are many canonicalization algorithms
  - Most canonicalization algorithms
    - need the whitespace preserved
    - Start with the "PRESERVE WHITESPACE" option
Storing a XML Document

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

XML text represented as document tree
Efficient Document Tree Storage

"Compression"
- Reduced storage
- Fast comparisons & navigation

SYSIBM.SYSEXMLSTRINGS

<table>
<thead>
<tr>
<th>String table</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Tag names encoded as unique integers
Efficient Document Tree Storage and Automatic Indexing

Each node has a path
pureXML®: XML Node Storage Layout & Region Index

- Node hierarchy of an XML doc stored on DB2 pages
- Documents that don’t fit on 1 page: split into pages/regions
- No architectural limit for size of XML documents
- NodeIDs used to identify individual nodes (1.2.4.3)

Example:

Document split into 3 regions, stored on 3 pages

Nodes are physically connected
- Regions are logically connected
- Key in Regions index is NodeID (NID)
- E.g. NID = 1.4.1.3
XML Storage

DAT Object

<table>
<thead>
<tr>
<th>ID</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR27</td>
<td>...</td>
</tr>
<tr>
<td>PR28</td>
<td>...</td>
</tr>
<tr>
<td>ACC</td>
<td>...</td>
</tr>
</tbody>
</table>

INX Object

Region

Path

/dept
/dept/employee
/dept/employee/@id
...

XDA Object
Index on an XML Column

- Created by users to improve query performance
- XML pattern identifies paths and values to index
The Big Indexing Picture

- **Relational Index**
- **Index on XML Column**
- **SQL Table with XML Column**

- **Relational Column 1**
- **Relational Column 2**
- **XML Column**

- **XML Regions Index**

- **Catalog Path Table**

- **XML Column Paths Index**

- **XML Storage**
  - .XDA file

Created by users to improve performance during queries on XML documents.

Logical mapping of regions in an XML document used to retrieve the document data.

Maps paths to path ids for each XML column. Subset of paths stored in global catalog path table.
Updating elements/attributes in DB2 9

DB2 Client Application

XML Parsing

XML Serialization

Change

Transmission

XML Serialization

XML Parsing

SQL Update

Parsed format

DB2

Parse format
Updating elements/attributes in DB2 9.5

Currently the first and unique implementation of the new W3C update standard
(released in April, 2007)
Old XML-Enabled Databases: Two Main Options

CLOB/Varchar or clob column

Extract selected elements/attr.

Side Tables

Regular tables for faster lookup

XML DOC

Fixed Mapping

Oracle

SQL Server

"Decomposition"

Shredder

Regular relational tables
DB2 9.5 XML features Recap

- Sub-document update
- Controlling index behavior
- Replication
  - Validation triggers
  - Validation check constraints
- Non-Unicode databases
- XML Load
- Compatible schema evolution
- sqlquery() parameters
- Base-table inlining/compression
- User-friendly publishing functions
- Simpler SQL/XML functions
- XSLT function
- Decomp enhancements
- Federation
- XQuery enhancements
- Omnifind
- Index Design Advisor**

** Available in future fixpacks
Monitoring
Tools

- DB2 collects monitoring data based on specific criteria
- Data exposed via non-graphical built-in tools such as:
  - Snapshot Monitor
  - Event Monitor
- Data exposed via Graphical Interface (DB2 CC):
  - Health Center
- Data exposed in reports with aggregation and trend analysis:
  - DB2 Performance Expert tool (java application)
- In future via Optim Data Studio Administration Console (web)
DB2 Editions
DB2 9.5 - Data Server Editions

**DB2 Workgroup**
- Linux
- Windows
- UNIX
- 4 CPUs, 16 GB RAM

**DB2 Express**
- Linux/Windows
- 2 CPUs, 4GB RAM

**DB2 Express-C**
- Linux/Windows
- 2GB RAM

**DB2 Enterprise**
- Linux/Windows/UNIX
- No limits

**DB2 for z/OS , DB2 for i5 OS**

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

**Clients**
- 32-bit & 64-bit – ALL

Free download,
# InfoSphere Warehouse 9.5 Editions

<table>
<thead>
<tr>
<th>Key Features</th>
<th>DW Starter</th>
<th>DW Intermediate</th>
<th>DW Advanced</th>
<th>DW Enterprise Base</th>
<th>DW Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licence limitations (maximums)</td>
<td>200 PVUs / 4GB database memory</td>
<td>400 PVUs / 32GB total memory</td>
<td>1000 PVUs / 2TB user data</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Platform Support</td>
<td>Linux Windows</td>
<td>Linux Windows</td>
<td>Linux only</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>DB2</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>SQL Warehousing Tool</td>
<td></td>
<td></td>
<td></td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>DWE Admin Console</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>DWE Design Studio</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Cubing Services &amp; MQT</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>DB2 Range Partitioning &amp; MDC</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Query Patroller / Workload Mgmt</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Performance Expert</td>
<td></td>
<td></td>
<td></td>
<td>Performance Optimization Feature</td>
<td>☑</td>
</tr>
<tr>
<td>Compression: Table and Backup</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>Storage Optimization Feature</td>
</tr>
<tr>
<td>Alphablox</td>
<td>Alphablox</td>
<td>Alphablox</td>
<td>DWE ABX Add-in</td>
<td>Alphablox</td>
<td>☑</td>
</tr>
<tr>
<td>Alphablox connectors</td>
<td></td>
<td></td>
<td>DWE ABX Add-in</td>
<td>non-DWE Connectors</td>
<td>☑</td>
</tr>
<tr>
<td>Intelligent Miner</td>
<td></td>
<td></td>
<td></td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Unstructured Text Analysis</td>
<td></td>
<td></td>
<td></td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Trade-ups available</td>
<td>DW Intermediate</td>
<td>DW Advanced</td>
<td>DW Enterprise</td>
<td>DW Enterprise Edition</td>
<td>N/A</td>
</tr>
</tbody>
</table>
DB2 9.5 Editions

- **DB2 Express-C**
  - Will use up to 2 processor cores, 2 GB memory
  - Available for:
    - Windows 32-bit (x86)
    - Windows 64-bit (x86_x64)
    - Linux 32-bit (x86)
    - Linux 64-bit (x86_64)
    - Linux on POWER (iSeries, pSeries)
  - Includes pureXML
  - Fixed Term License (FTL) can be purchased for support. See [http://www.ibm.com/software/data/db2/express/support.html](http://www.ibm.com/software/data/db2/express/support.html)
Misc
DB2 Data Types

- DB2 supports all standard SQL data types (e.g. CHAR, VARCHAR)

- DB2 9 introduced the native XML data type is available to store XML documents directly in a table
  - XML columns can be queried using XQUERY, allowing subsets of a document to be retrieved

- DB2 9.5 introduces two new data types:
  - DECFLOAT: Accuracy of DECIMAL; performance of FLOAT
  - ARRAY: Defines an array based on one of the built-in types

```sql
CREATE TYPE phonenumbers AS DECIMAL(10,0) ARRAY[5]
```
Database Application Development Technologies

- Key Database Technologies
  - SQL / SQL Procedures
  - XML
  - SOA / Web Services

- Developer communities
  - C/C++
  - Java (JDBC / SQLJ)
  - .NET (C#, VB .NET)
  - Open Source
    - PHP
    - Perl
    - Python

Powered By Ruby

© 2008 IBM Corporation
The SAMPLE database can be used for testing applications, trying features of DB2, etc.

To create the sample database populated with both standard relational data and XML data:

```
db2sample -sql -xml
```

The SAMPLE database can be dropped and recreated at any time:

```
db2 drop database SAMPLE
```

Most of the sample application programs that come with DB2 use sample database
Grazie
DB2 9.5 for Linux, Unix and Windows Fundamentals
Table spaces can be managed by the operating system, by the database manager, or by the DB2 automatic storage feature:

- **System Managed Space (SMS):**
  - Data stored in files in the file system
  - Access to data controlled using standard I/O functions of the OS
  - Space not allocated by the system until it is required
  - Ideal for small, personal databases; databases that grow/shrink rapidly
  - Low maintenance and monitoring

```sql
CREATE TABLESPACE tbsp1
MANAGED BY SYSTEM
USING ('d:\acc_tbsp', 'e:\acc_tbsp', 'f:\acc_tbsp')
```
DB2 Storage: Table Spaces

- **Database Managed Space (DMS):**
  - Data stored in files or on raw devices
  - Can bypass operating system I/O functions, increasing performance
  - Ideal for performance-sensitive applications
  - Increased maintenance and monitoring

```sql
CREATE TABLESPACE tbsp1
  PAGESIZE 8K
  MANAGED BY DATABASE
  USING (FILE 'd:\db2data\acc_tbsp' 5000, FILE 'e:\db2data\acc_tbsp' 5000)
```

- Can also have containers automatically resized when they fill up:

```sql
CREATE TABLESPACE tbsp2
  PAGESIZE 8K
  MANAGED BY DATABASE
  USING (FILE ' /storage/dms1' 10 M) AUTORESIZE YES
```
DB2 Storage: Table Spaces

- **Automatic Storage Table Space:**
  - Can be used if database is enabled for automatic storage
  - Database manager assigns containers automatically, one each storage path
  - 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

```
CREATE DATABASE mydb AUTOMATIC STORAGE YES on c:\storpath1, c:\storpath2, c:\storpath3
CONNECT TO mydb
CREATE TABLESPACE tbsp1 MANAGED BY AUTOMATIC STORAGE
```