DB2 Viper

Take the test drive at:
www.ibm.com/db2/viper

Paul C Zikopoulos, BA, MBA paulz@ca.ibm.com
DB2 Advanced Technical Expert (DRDA and Cluster)
DB2 Customer Solutions Expert (BI and DBA)
Presenter’s Biography

Paul C. Zikopoulos, BA, MBA, is an award-winning writer and speaker with the IBM Database Competitive Technology team. He has more than ten years of experience with DB2 UDB and has written over sixty magazine articles and several books about it. Paul has co-authored the books: DB2 9 New Features, DB2 Version 8: The Official Guide, DB2: The Complete Reference, DB2 Fundamentals Certification for Dummies, DB for Dummies, and A DBA’s Guide to Databases on Linux. Paul is a DB2 Certified Advanced Technical Expert (DRDA and Cluster/EEE) and a DB Certified Solutions Expert (Business Intelligence and Database Administration). In his spare time, he enjoys all sorts of sporting activities, running with his dog Chachi, and trying to figure out the world according to Chloë – his new daughter. You can reach him at: paulz_ibm@msn.com.
What’s on the Minds of 450 of the World’s Leading CEOs?

**CEO needs**
- **Revenue growth** with cost containment
- **Key competency:** responsiveness
- **Critical success factor:** enable effectiveness of people and processes

**CIO challenges**
- Aligning IT and business goals to grow revenue and contain costs
- Building responsiveness and agility into the organization through IT
- How can IT help enable people and teams to be more effective?

Source: CEO Study of 456 WW CEOs, IBM Corporation - Feb 2004

Source: Operating Environment Market Drivers Study, IBM Corp. 2004
Companies Face Increasing Information Related Challenges

60%+ of CEOs: Need to do a better job capturing and understanding information rapidly in order to make swift business decisions

79% of companies: have 2+ repositories... 25%: have 15+

85% of information is unstructured

17% of IT Budgets Spent on Storage HW, SW, People

37% CGR Disk Storage Growth '96-'07

122 Terabytes Disk Storage in 2005

Only 1/3 of CFOs believe that the information is easy to use, tailored, cost effective or integrated

48 disparate financial sys. and 2.7 ERP sys. in the average $1B company

30-50% of design time is copy management

30% of people’s time is spent searching for relevant information

40% of IT budgets may be spent on integration

Sources: IBM & Industry Studies, Customer Interviews
DB2 Viper test drive highlights

- Information as a Service
  - Pure XML / Relational Hybrid

- Secure and Resilient
  - Security
  - Limitless
  - Compression
  - Recovery
  - Deployment

- Manage your Business not your Database
  - Autonomic Computing
  - Partitioning

- Agile Development
  - .NET and Visual Studio
  - Eclipse
  - Partners

www.ibm.com/db2/viper
What is XML?

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

- **Benefits**
  - Extensible
    - No fixed format or syntax
    - Structures can be easily changed
  - Platform Independent
    - Not tied to any platform, operating system, language or software vendor
    - XML can be easily exchanged
  - Fully Unicode compliant

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

In files...
- Storage not managed and not secure

In LOBS...
- Content and business value locked up

Shred to tables...
- Complex and fragile mapping

In XML DB...
- Scalability & integration concerns
XML Usage in Business Today

- **Business to Business Integration**
  - Platform independent transport mechanism (SOA)
  - Example: Receipt of a purchase order triggering a transaction flowing over a service oriented architecture

- **Document Management**
  - Government and legal industry require digital signature
  - Example: Tax forms, require signature and often change year to year

- **Business Insight**
  - Universal representation of data from multiple data sources
  - Example: Claims adjustor can review damage estimates from multiple garages without consideration of original format
XML Market Projections

“XML Storage is a high growth area”

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

“XML database revenue to grow at twice the rate of the total database market” - IDC

Chart Sources: XML Market Opportunities, Forecasts and Strategies, 2004-2009
Wintergreen Research Inc. ZapThink
Use Case: Financial Data (FIXML)

- Buying 1000 Shares of IBM Stock..

Old FIX Protocol

- Extensible
- Lower appl development & maintenance cost

New FIXML Protocol

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

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

<table>
<thead>
<tr>
<th>Id</th>
<th>LastName</th>
<th>FirstName</th>
<th>Street</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pirahesh</td>
<td>Hamid</td>
<td>1 Harry Rd</td>
<td>San Jose</td>
<td>CA</td>
<td>95141</td>
</tr>
<tr>
<td>2</td>
<td>Selinger</td>
<td>Pat</td>
<td>555 Bailey Ave</td>
<td>San Jose</td>
<td>CA</td>
<td>95141</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ItemID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>#6 wire nut</td>
</tr>
<tr>
<td>3</td>
<td>Small Walrus</td>
</tr>
<tr>
<td>4</td>
<td>Apollo moon rocket</td>
</tr>
</tbody>
</table>

- **XML is a data model:**
  - Nodes (elements, attributes, comments, etc.)
  - Relationships between nodes
  - Sequence based w/ some sets
  - Flexible schema
DB2 Viper XML Technology

- Today’s Challenge: XML must be force fit into relational data model
  - 2 Choices:
    1. Shredding or decomposing
       - Loses hierarchical dependencies
       - Loses digital signature
       - Creation of several relational tables as pointers
       - Difficult to change original XML document
    2. Large Object (BLOB, CLOB, Varchar)
       - SLOW performance
         > Search performance is slow (must parse at search time)
         > Retrieval of sub-documents is expensive
         > Update inside the document is slow
         - Indexing is inefficient (based on relative position)
         - Difficult to join with relational
         - Costs get worse as document size increases
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
  - Influenced creation of SQL, delivery of Oracle by Relational software 1979, and delivery of DB2 in 1981 based on System R prototype

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

Continuous IBM innovation
The Database Inflection Point: XML Changes the Game

“IBM Moves the Database Goalposts…”

- Pure XML integration simplifies & unifies the infrastructure
  - Unifies Data and Content
  - Extends Asset Utilization
  - Leverages Existing Skills

- Pure XML document storage in DB2
  - Ensures fidelity of XML document
  - Optimizes Performance
  - Provides Flexibility
  - Leverages Mature DB2 Services
Native XML Storage

- Stores XML in parsed hierarchical format (similar to the DOM representation of the XML infoset)
  
  ```
  create table dept (deptID char(8),..., deptdoc xml);
  ```

- Relational columns are stored in relational format (tables)
- XML columns are stored hierarchically
- XML stored in UTF8
Pure XML in DB2 Viper

"Our tests of the new XML Decomposition in DB2 Viper showed an 8 times performance improvement over the DB2 XML Extender." Dr. Andreas Birkendorf

XML integrated in all facets of DB2!

New XML applications benefit from:

- Ability to seamlessly leverage relational investment
- Proven Infrastructure that provides enterprise-class capabilities
## DB2 Viper: Early adopter feedback

### Proto-type results using DB2 Viper based SOA solution
- **Fast, easy access to richer product & client information**
- **Quickly create customized products that customers want**
- **Expected to process five times more business**

### Development and app. performance re: XML data

<table>
<thead>
<tr>
<th>Action</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI/IO: 8 hrs</td>
<td>30 min.</td>
</tr>
<tr>
<td>Shred: 2 hrs</td>
<td>30 min.</td>
</tr>
<tr>
<td>Add field to schema</td>
<td>1 week</td>
</tr>
<tr>
<td>Development of search &amp; retrieval business processes</td>
<td>1 week</td>
</tr>
<tr>
<td>Relative lines of I/O code (65% reduction)</td>
<td>100</td>
</tr>
<tr>
<td>XML element non-shredded</td>
<td>24 - 36 hrs</td>
</tr>
<tr>
<td>XML element shredded</td>
<td>5 min.</td>
</tr>
<tr>
<td>Query non-shredded</td>
<td>20 sec - 10 min</td>
</tr>
<tr>
<td>Query</td>
<td>5 min.</td>
</tr>
</tbody>
</table>
DB2 Viper Pure XML Storage vs. the Competition

- **Information Fidelity**: ✔️, ✗, ✔️, ✔️
- **Integration**: ✔️, ✗, ✗, ✔️
- **Schema Flexibility**: ✗, ✗, ✔️, ✔️
- **Performance/Scale**: ✗, ✗, ✗, ✔️
- **Programming Models**: ✗, ✗, ✗, ✔️
- **Manageability**: ✔️, ✗, ✗, ✔️
Net Search Extender Now Free

SQL-Query using NSE functionality:

```sql
SELECT XMLSERIALIZE(XMLQUERY('// PASSING BY REF T.Description AS "t") AS CLOB(32K))
FROM xmlproduct AS T
WHERE CONTAINS(Description, 'RESULT LIMIT 10 sections ("name") "silver"') - 1
```
IBM’s Open XML Strategy

Deliver XML-based solutions that will help our customers and business partners build, deploy, and manage e-business applications.

- Ensuring strong, open standards
  - Driving W3C activities to evolve standards
    - e.g. XML, DOM, XSL, Schema, Linking, Info Set, RDF, SVG, HTML ...
  - Using business relationships to help create and establish industry-specific standards
    - e.g., fpML, HL7, HR-XML...
  - XML.org for access and interoperability

- Enabling our entire product line for XML with a world class XML infrastructure
  - Parser, Storage, Search, Transform,
  - Digital signature, Web Services,
  - Hardware assist, Compression
What industry experts are saying about DB2 Viper XML Technology?

- “…this leaves Oracle and Sybase …well behind the curve, Microsoft and the others more or less out of sight.”
  - Philip Howard, Bloor Research, The Register

- “…enable users to work with both types of data via SQL or Xquery requests. It could also open up the world of database applications to developers and ISVs with experience in other data types and applications.”
  - Barbara Darrow, CRN

- "You want to be able to take those data-centric things in XML and put them into a database without a loss of fidelity, and this is one area where IBM is going further than Oracle and Microsoft."
  - Peter O'Kelley Burton Group, Internet News
Words from DB2 Viper Beta Customers and Partners

“We are delighted to be partnering with IBM to integrate our platform with DB2 Viper. The combination of industrial strength database management for native XML by DB2 Viper and Skytide’s ability to provide direct multidimensional analysis of XML data, removes two key barriers to widespread adoption of XML and the transformation of this data into actionable business information.”

- Joseph Rozenfeld, Vice President of Products for Skytide

“We wanted to be able to support queries that just were based on information in the e-records that had not been indexed. The way we have to do that at the moment is not terribly efficient … [Native XML support in DB2 Viper] is going to enable us to store things more compactly and access them easier … and make it easy for us to be able to ingest and then export data in XML when we’re able to migrate to that version of DB2.”

- Dave Richards, Chief Technology Officer for The Research Libraries Group Inc.

“The upcoming release of DB2, offers leading-edge technology for storing, managing and searching XML data in a secure, highly scalable environment. The new multi-structure hybrid architecture offered by DB2 combines the best of relational database management technology with the best of XML data management.”

- Tim Harvey, CEO of XAware.

“Our development time using Viper's native XML store is a radical improvement over existing XML 'shred' technology. We are now able to make schema changes in minutes rather than days and will dramatically improve our customer response time.”

- Thore Thomassen, Senior Enterprise Architect for Storebrand
Partners Enabling XML Based Solutions for DB2 V9
Autonomics and Manageability

Paul C Zikopoulos, BA, MBA paulz@ca.ibm.com
DB2 Advanced Technical Expert (DRDA and Cluster)
DB2 Customer Solutions Expert (BI and DBA)
Automation by default

- Many autonomic computing features turned on by default for new Viper databases

- Included:
  - Configuration Advisor (2 second tuning)
  - Adaptive Self Tuning Memory
  - Automatic data statistics collection.

- Better defaults for I/O Cleaners and I/O servers
  - Default for `NUM_IOSERVERS` and `NUM_IOCLEANERS` set to `AUTOMATIC`
    - Values calculated at database startup time
      - `NUM_IOCLEANERS` calculated based on number of CPUs and partitions
      - `NUM_IOSERVERS` calculated based on parallelism settings of all the tablespaces
Self-Tuning Memory Manager

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

- **Memory tuning before DB2 8**
  - Many of the memory configuration parameters and buffer pools could not be changed online
  - If a DBA wished to modify the amount of memory allocated to a given memory consumer while the database was running, you had to recycle the database, and in some cases, the entire instance
Memory Tuning In Version 8

- Most important configuration parameters are now dynamically updatable
  - Example: Can ALTER the size of buffer pools online, as well as package cache, sort heap, utilities heap, etc.
  - No more system downtime for many configuration changes
    - Solves the availability issue, but not tuning expertise

- In V8 the best memory tuning technique is to use the DB2 Configuration Advisor or some derivative thereof (i.e. AUTOCONFIGURE command)
  - Sets all important memory parameters
  - Tunes most workloads very well
    - For OLTP workloads it can tune as well as performance experts
  - Considerations with the DB2 Configuration Advisor results:
    - Configuration is static
      - If workload changes over time configuration may become out of date
    - Configuration is dependant on DBA description of workload
      - If workload isn’t as described, configuration may be less than optimal
Memory Tuning In Version 8.2

- Heap overflows
  - Package and Catalog caches allowed overflows and consume more than their configured memory if needed
  - DB2 would allocate 20% more than the configured amount of memory for the database server to ensure this amount was available
Database Configuration Settings

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
Package cache size (4KB) (PCKCACHESZ) = AUTOMATIC
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = AUTOMATIC
Sort list heap (4KB) (SORTHEAP) = AUTOMATIC

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) = 50
Percent. of lock lists per application (MAXLOCKS) = 22
Package cache size (4KB) (PCKCACHESZ) = (MAXAPPS*8)
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 5000
Sort list heap (4KB) (SORTHEAP) = 256

Self tuning memory (SELF_TUNING_MEM) = OFF
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
Package cache size (4KB) (PCKCACHESZ) = 2400
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = AUTOMATIC
Sort list heap (4KB) (SORTHEAP) = AUTOMATIC
STMM In Action – Tuning An OLTP Workload
STMM In Action

Index Dropped
Average Response Time = 6,206 Seconds

63% Reduction
Average Response Time = 2,285 Seconds

Average Response Time = 959 Seconds
STMM In Action – Two Databases On The Same Box

![Graph showing memory usage over time for two databases on the same box.](image_url)
STMM In Action – Against Default and Benchmark Configurations

STMM beats default configuration by nearly 4x

STMM also edges out benchmark tuned system
Object Ownership Transfer

- DB2 9 gives you the ability to transfer the ownership of objects from one user to another
  - The creator of any object has CONTROL authority over that object for the life of that object
  - In DB2 9, the CONTROL privilege (or ownership) can be transferred to someone else
    - Either SECADM user or the owner of the object can transfer ownership

- You can transfer the ownership of virtually any object in DB2 (table, index, alias, constraint, function, procedure, trigger, tablespace, schema, and more)
  - Good for personnel turnover – simply revoke user account and don’t worry about schema removal, just transfer to someone else
Object Ownership Transfer

![Diagram of DB2 interface showing object ownership transfer process.]

- **Ownership transfer in DB2**: A command-line interface showing the process of transferring object ownership from one user to another.
- **Database connection information**: Details of the database server, authorization ID, and database alias.
- **Ownership transfer command**: Syntax for transferring ownership of a table from one user to another, preserving privileges.

---

36
Copy Schema

- DB2 9 now gives you the ability to copy all of the objects in one schema to another using a new routine called `ADMIN_COPY_SCHEMA`
  - Use it to duplicate all of the objects (tables, indexes, views, MQTs, and so on) from one schema to another in the same or different table spaces
    - Optionally include the data (in which case it is `LOAD` from `CURSOR`) or just `DDL`
      
      ```sql
      ADMIN_COPY_SCHEMA (source_schema, target_schema, copymode, new_owner, source_tablespace, target_tablespace, error_table_schema, error_table_name)
      ```

- Notes:
  - Can use `DDL` or `COPY` (copies data) for `copymode` option
  - `NULL` value for `new_owner` defaults ownership to connection ID
  - This procedure does not support copying the following objects: Index Extensions, Nicknames, Packages, Typed tables, UDSTs, Types views, .JARS
    - If any of these objects exist, the rest of the objects in the schema are copied, and an entry is added to the exception table to notify that DBA this the object was not copied
Copy Schema

```
G:\Documents and Settings\viper>db2 list tables for schema paulz
Table/View
------------------------------------------ Schema Type Creation time
0 record(s) selected.

G:\Documents and Settings\viper>db2 call admin_copy_schema ('VIPER', 'PAULZ', 'DDL', 'PAULZ', 'USERSPACE1', 'USERSPACE1', 'PAULZ', 'SCHEMACOPYERRORS')
SQL0443N Routine "ADMIN_COPY_SCHEMA" (specific name "x_COPY_SCHEMA") has
returned an error SQLSTATE with diagnostic text 'SYSTABLESPACE tablespace does
not exist'. SQLSTATE=38000

G:\Documents and Settings\viper>db2 call admin_copy_schema ('VIPER', 'PAULZ', 'DDL', 'PAULZ', 'USERSPACE1', 'USERSPACE1', 'PAULZ', 'SCHEMACOPYERRORS')
Value of output parameters
Parameter Name : ERRORTABASCHHEMA
Parameter Value : PAULZ
Parameter Name : ERBORTABASNAME
Parameter Value : SCHEMACOPYERRORS
Return Status = 0

G:\Documents and Settings\viper>db2 list tables for schema paulz
Table/View
------------------------------------------ Schema Type Creation time
AGT PAULZ T 2006-08-04 17:45:27.625000
AGT_USR PAULZ T 2006-08-04 17:45:27.625000
APPLICATIONS PAULZ T 2006-08-04 17:45:27.844000
CL_SCHED PAULZ T 2006-08-04 17:45:26.107000
DEPARTMENT PAULZ T 2006-08-04 17:45:21.231000
DEPT PAULZ A 2006-08-04 17:45:28.515000
EMP PAULZ A 2006-08-04 17:45:24.259000
EMP_DESC PAULZ A 2006-08-04 17:45:24.329000
EMP_PHOTO PAULZ A 2006-08-04 17:45:29.375004
EMP_RECRUIT PAULZ T 2006-08-04 17:45:29.457000
EMP_RESUME PAULZ A 2006-08-04 17:45:29.532000
EMPLOYEE PAULZ T 2006-08-04 17:45:30.859000
EMPLOC PAULZ T 2006-08-04 17:45:23.140000
EMPLOYEEPROJECT PAULZ T 2006-08-04 17:45:45.831000
EXPLAIN_ARGUMENT PAULZ T 2006-08-04 17:45:32.859000
EXPLAIN_DIAGNOSTIC PAULZ T 2006-08-04 17:45:33.109000
EXPLAIN_DIAGNOSTIC_DATA PAULZ T 2006-08-04 17:45:34.210000
EXPLAIN_INSTANCE PAULZ T 2006-08-04 17:45:33.399000
EXPLAIN_OBJECT PAULZ T 2006-08-04 17:45:33.515000
EXPLAIN_OPERATOR PAULZ T 2006-08-04 17:45:35.812000
EXPLAIN_PREDICATE PAULZ T 2006-08-04 17:45:33.937000
```
Drop Schema

- DB2 9 now gives you the ability to drop all of the objects in a schema using a new routine called ADMIN_DROP_SCHEMA
  - Use it to drop all of the objects (tables, indexes, views, MQTs, and so on) in a schema
    \[
    \text{ADMIN_DROP_SCHEMA (schema\_name, drop\_mode, error\_table\_schema, error\_table\_name)}
    \]

- Notes:
  - DROP_MODE in DB2 9 must be set to \text{NULL}
    - Future versions of DB2 may add new functionality to this function with this parameter
Drop Schema

![DB2 output showing the drop schema command and its effects on the database schema.](image)

**Example:**

```sql
DROP Schema Example

C:\Documents and Settings\viper>db2 connect to sample

Database connection information:
- Database server: DB2/NT 9.0.0
- SQL authorization ID: VPER
- Local database alias: SAMPLE

C:\Documents and Settings\viper>db2 list tables for schema chloe

Table/View | Schema | Type | Creation time
---|---|---|---
CUSTOMER | CHLOE | T | 2006-09-07 19:47:28.343001
DEPT | CHLOE | A | 2006-09-07 19:47:28.687000
EMP | CHLOE | A | 2006-09-07 19:47:28.687000
EMP_Act | CHLOE | A | 2006-09-07 19:47:28.687000
EMP_ACT | CHLOE | A | 2006-09-07 19:47:29.312002
EMPLOYEE | CHLOE | T | 2006-09-07 19:47:29.328000
INVENTORY | CHLOE | T | 2006-09-07 19:47:30.231001
ORG | CHLOE | T | 2006-09-07 19:47:30.673001
PRODUCT | CHLOE | T | 2006-09-07 19:47:30.187003
PROD | CHLOE | A | 2006-09-07 19:47:30.625002
SALES | CHLOE | T | 2006-09-07 19:47:31.000004
STAFF | CHLOE | T | 2006-09-07 19:47:31.093000
SUPPLIERS | CHLOE | T | 2006-09-07 19:47:31.171001

C:\Documents and Settings\viper>db2 call admin_drop_schema <‘CHLOE’,NULL,’PAUL2’,’DROPSCHEMERRORS’>

Value of output parameters:
- Parameter Name: ERRORDSBSHEMA
  Parameter Value: 
- Parameter Name: ERRORTAB
  Parameter Value: 
  Return Status = 0

C:\Documents and Settings\viper>db2 list tables for schema chloe

Table/View | Schema | Type | Creation time
---|---|---|---

0 record(s) selected.
```
Simplified Storage Administration

- User specifies a group of storage devices and DB2 allocates and grows table consumption of storage on demand
  - First introduced in Version 8.2.2
  - Extended to DPF in Viper
  - Intended as a “single point of storage management” for table spaces
  - Create a database and associate a set of storage paths with it

- AUTOMATIC STORAGE table spaces
  - No explicit container definitions are provided
  - Containers automatically created across the storage paths
  - Growth and addition of containers managed by the data server

- Built around the DMS storage model

- Easy to add storage paths to the database afterwards
- Add storage paths to the database afterwards
- Redefine those storage paths during a database RESTORE
Automatic Storage Provisioning - Syntax

CREATE DATABASE DB1
   AUTOMATIC STORAGE YES

CREATE DATABASE DB3
   ON /data/path1, /data/path2

CREATE TABLESPACE TS2
   MANAGED BY AUTOMATIC STORAGE

CREATE TEMPORARY TABLESPACE TEMPTS

CREATE USER TEMPORARY TABLESPACE USRTMP MANAGED BY
   AUTOMATIC STORAGE

CREATE TABLESPACE TS1
   INITIALSIZE 500 K
   INCREASESIZE 100 K
   MAXSIZE 100 M
Automatic Storage Provisioning - Restore

RESTORE DATABASE TEST1

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

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

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

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

- **Why?**
  - Large FACT table in data warehouse
    - Data rolled-in weekly in batches using ATTACH
    - Data rolled-out periodically using DETATCH
    - Ranges match granularity of roll-in/roll-out
    - Small or non existent maintenance windows
  - BI style queries
    - Queries to roll up data by DATE, REGION, PRODUCT, CATEGORY, etc.
    - Queries are complex and/or long running
  - Non-trivial database schemas
    - Indexes
    - Dimension tables
    - Referential integrity constraints
    - MQTs
Table Partitioning

<table>
<thead>
<tr>
<th>Single table</th>
<th>Partitioned Table 32K Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-Z</td>
<td>A-C</td>
</tr>
<tr>
<td>Backup Load</td>
<td>Backup Load</td>
</tr>
<tr>
<td>Recover</td>
<td>Backup Load</td>
</tr>
<tr>
<td></td>
<td>Recover</td>
</tr>
<tr>
<td>64G</td>
<td>64G</td>
</tr>
<tr>
<td></td>
<td>64G</td>
</tr>
<tr>
<td></td>
<td>64G</td>
</tr>
<tr>
<td></td>
<td>64G</td>
</tr>
</tbody>
</table>
Table Partitioning

- Jan: 64G (DETACH)
- Feb: 64G
- Mar: 64G
- Apr: 64G
- May: 64G
- June: 64G (ATTACH)

LOAD
Partitioning - Grand Unification

- Three ways to spread data
  - DISTRIBUTE BY HASH - currently in DB2 Enterprise with DPF
  - PARTITION BY RANGE – table partitioning
  - ORGANIZE BY DIMENSIONS – MDC

- Use similar syntax and concepts for all partitioning
  - Similar algorithms (HASH, RANGE, etc.)
  - Conciseness

Syntax Unification (xxxxx BY yyyyy)
- DISTRIBUTE BY HASH
- PARTITION BY RANGE
- ORGANIZE BY DIMENSIONS

```sql
create table orders
(order_id, ship_date, region, category)
in tbsp1, tbsp2, tbsp3, tbsp4
  distribute by (order_id)
  partition by (ship_date)
  (starting from ('01-01-2005')
    ending ('12-31-2006') every (1 month)
  organize by (region, category);
```
Table Partitioning – Syntax Example

- **Example MDC:**

  ```sql
  CREATE TABLE ORDERS (ORDER_ID, SHIP_DATE, REGION, CATEGORY) 
  IN TBSP1, TBSP2, TBSP3, TBSP4 PARTITION BY (SHIP_DATE) 
  (STARTING FROM('01-01-2005') 
  ENDING ('12-31-2006') EVERY (1 MONTH)) 
  ORGANIZE BY REGION,CATEGORY);
  ```

- **Example:**

  ```sql
  CREATE TABLE ORDERS(...) DELIVERY_DATE) 
  IN TBSP1, TBSP2, TBSP3, TBSP4 PARTITION BY DELIVERY_DATE) 
  (STARTING FROM('01-01-2005') ENDING ('12-31-2005') 
  EVERY (3 MONTHS));
  ```

- **Syntax also supports clauses where specific values can be specified**

<table>
<thead>
<tr>
<th>TBSP1</th>
<th>TBSP2</th>
<th>TBSP3</th>
<th>TBSP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAN</td>
<td>APR</td>
<td>JUL</td>
<td>OCT</td>
</tr>
<tr>
<td>FEB</td>
<td>MAY</td>
<td>AUG</td>
<td>NOV</td>
</tr>
<tr>
<td>MAR</td>
<td>JUN</td>
<td>SEP</td>
<td>DEC</td>
</tr>
</tbody>
</table>
Table Partitioning - Benefits

- **Performance through partition and index elimination**

- **Fast roll-in and roll-out of data**
  - Easily add or remove data from the table without having to take the database offline
  - Useful in a data warehouse environment where you often need to load or delete data to run decision-support queries

- **Better optimization of storage costs**
  - Lets you integrate better with hierarchical storage models by only using your fastest and most expensive storage hardware for only the most active table partitions,
  - If most of your queries only run against the last three months of data, you have the option to assign slower and less expensive storage hardware to older data.

- **Larger table capacity**
  - Removes limits on the maximum amount of data a storage object can hold
  - By dividing the contents of the table into multiple storage objects or partitions, each capable of supporting as much data as in a non-partitioned table, you can effectively create databases that are virtually unlimited in size.

- **Greater index placement flexibility**
  - Can store indexes for partitioned tables in their own storage objects (table spaces), as opposed to being in the same storage object as the non-partitioned table
  - Useful for performing faster index operations (such as drop index, online index create, and index reorganization), managing table growth, and reduced I/O contention, providing more efficient concurrent access to the index data for the table
What I Like About DB2 Partitioning

- DB2 syntax is significantly simpler
- Detaching a partition creates a real table & doesn’t destroy the data
- Attaching a partition takes a table & creates a partition anywhere
  - No restriction on putting it at the high end of the table
- DB2 allows for virtual partitioning via MDC which gives ranges within ranges or list within Range
- Combined with DPF we also allow Hash partitioning across servers and Range partitioning within Servers.
- All three (Table, DPF and MDC) deliver more composite partitioning strategies than Oracle provides
- DB2 provides MDC and DPF advisors leaving only the granularity of table partitioning up to the DBA
  - The granularity in this case depends on business considerations for rollin/rollout periods and not on query workload due to the better parallelism built into DB2
Availability

Paul C Zikopoulos, BA, MBA paulz@ca.ibm.com
DB2 Advanced Technical Expert (DRDA and Cluster)
DB2 Customer Solutions Expert (BI and DBA)
Backup and Restore

- Table function to list files in a database
  - Used to automate support of split mirror backup/recovery
- Perform redirected restore operations with scripts automatically generated from existing backup images.
Restore Database from Only Table Space Backups

- In DB2 8, it was not possible to recover an entire database (from scratch) from table space level backups
  - You have to start with at least a full backup image

- In DB2 9 a table space level backup contains all the necessary metadata to know the full structure of the database (including the structure of table spaces not included in the given table space backup image)
  - Never take a full database backup again
    - When you restore just a table space in DB2 9, it knows what other table spaces existed at the time the target image was backed up
    - Can access the database even if all table spaces have not been restored
      - You can restore other table spaces as your discretion so long as the catalog table spaces is restored

- How it works
  - Start a restore into a database with a table space backup image
    - This puts the database in a REBUILDING phase
      - During this phase you can restore as many other table space images as you want
      - Once finished, just roll forward to the P.I.T. you want to open the database at
    - Any table spaces not restored are placed in a RESTORE PENDING state for which you can:
      1. Drop the table space if it is not loner needed
      2. Restored at a future time and roll forward to a point of consistency
      3. Leave them in RESTORE PENDING state and deal with the later

- Use cases
  - Create a test system with only those table spaces you need for your tests
  - For VLDBs, just perform table spaces backups without exposure concerns of not having a full backup image
    - No need to do full database restores with this method if the size of your database is prohibitive
Change Point in Time for Recovery

- DB2 8.2 delivered `RECOVER` command that allows you to combine that `RESTORE` and `ROLLFOWARD` operations in a single command
  - If a failure occurred, `RECOVER` had to start again
  - Could be problematic, especially if you put the wrong recover time for large database – must stop and restart from beginning

- DB2 9 lets you change the PIT that you want to recover and allow a recovery process to continue where it left off
  - By default, DB2 first looks first looks to see if a previous `RECOVER` command failed, and if so, pick up where it left off
    - If `RESTORE` isn’t finished before change of PIT, DB2 will have to start this again
    - If `RESTORE` is finished and you haven’t hit new PIT, DB2 will just change it
    - If `RESTORE` is finished and your past the new PIT, you can’t undo the roll forward, so DB2 will start again
Problem Resolution – Engine Trap Data Collector

- Part of minimizing outages is to quickly diagnose the root cause of a failure scenario
  - If DB2 aborts (due to a trap or segment violation for example) you want to capture as much information as possible to avoid having to create the problem
  - DB2 8 gives you a fixed set of trap information

- DB2 9 lets you customize the information dumped at server failure time
  - New call out to the `db2cos` script (`SQLLIB/BIN`) that is automatically executed in the event of a DB2 engine trap
    - You can edit this script to dump additional information
    - By default this script files contains the following `db2pd` command:
      ```bash
      db2pd -d %DATABASE% -inst >>
      %DIAGPATH%/db2cos%PID%%TID%.%DPART%
      ```
    - This information by default contains platform, memory, process information, as well as configuration parameters and other database information
Server Hang Data Collection

- Just like engine traps, server hangs require specific diagnostic information to be captured to get at the root cause of the problem
  - Though very rare, there are occasional defects that results in one process holding an internal DB2 latch with it abends
    * A latch is like a row lock, but it protects an area of memory
    * If DB2 wants to update the catalog cache in memory with new authorization memory it first must obtain a latch for that area of memory so it doesn’t overwrite someone else’s memory

- DB2 9 has new diagnostic information on the db2pd command to get latch information using the -latch keyword
  - Likely no output from this command because latches are held unless there is an issue
  - If you ran this command multiple times and had the same information, you could suspect a problem
    * Use `db2pd -stack <pid>` to dump the stack to identify the line of code where the process is hung
Compression

Paul C Zikopoulos, BA, MBA paulz@ca.ibm.com
DB2 Advanced Technical Expert (DRDA and Cluster)
DB2 Customer Solutions Expert (BI and DBA)
Compression - Today

- **Null Default Compression (V8.1)**
  Joint benefit with Row Level compression

- **Multi-Dimensional Clustering (V8.1)**
  Like “Index compression”

- **Backup Compression (V8.2)**
  - Additional gain with uncompressed objects

- **pureXML Compression (V9)**
  - XML tag substitution

---

"With the new compression technology in DB2 Viper, we realized an 80 percent improvement in space savings for our most critical tables in our Data Warehouse. We were even more pleased with this technology when we found that Viper’s compression capability helped us process queries to the database an average of 40 Percent faster than before. We’re looking forward to seeing the same results with our Operational Data Store and OLTP systems.” — Donny Ledbetter, Sr. DBA, AutoZone
Why Compression?

- Although the cost of disk drives measured in $/GB continues to decline, the overall IT expenses attributed to the storage infrastructure continues to climb.
  - Results of both explosive growth in storage requirements of both structured and unstructured data that needs to be managed.
  - What about the associated costs of the backup media, times, labor, restore storage and more need to be considered?
  - Assuming 50% compression:
    - Backup images on disk are 50% smaller, 50% faster, and require 50% less storage space.
    - Add in compliment costs to this like HVAC, power, floor space.
    - These costs can be reduced with less storage.

- Compression isn’t just about storage savings:
  - More rows on a page.
  - More densely packed buffer pools.
  - Alleviation of I/O bound systems.
  - Faster utilities (REORG, RUNSTATS, BACKUP, RECOVERY, and so on).
Storage growth outpacing Server growth

“The networked and attached data storage market is growing 5 to 9 percent faster than the server and personal computer markets, according to a Merrill Lynch industry survey of 75 U.S. and 25 European CIOs.”

“Storage hardware is steadily increasing as a percentage of IT budgets”

“Backup and recovery was the second most important driver of spending in 2006 and is the No. 1 storage issue keeping CIOs awake at night”

- **Large Wallstreet financial institution:**
  - IT Projects are charged $32/8GB/month for their storage usage
  - They have 250TB of storage allocated to databases in their shop
    - Includes storage for backups and online databases (majority is backup)
  - Chargeback for data storage = $1 Million per month ($12M/Yr)
  - If Storage Optimization Feature cuts storage usage by 40%, you could save them $4.8 Million dollars per year
Compression - Viper

- **Row Level Compression**
  - Modeled after DB2 z/OS
  - db2inspect for savings estimate
  - Transparent to application
  - Performance boost to I/O bound systems
  - Compressed when written to the logs
  - Static dictionary based compression algorithm by row
  - Only data on data page and log records are compressed

- **Dictionary based - symbol table for compressing/decompressing data records**

- **Dictionary per table stored within the permanent table object (~74KB in size)**

- **Data resides compressed on pages (both on-disk and in bufferpool)**
  - Significant I/O bandwidth savings
  - Significant memory savings
  - CPU costs: Rows must be decompressed before being processed for evaluation

- **Log data from compressed records in compressed format**

- **Does not compress rows where no storage saving is realized for that row**

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

- Dictionary contains repeated information from the rows in the table
  - Compression candidates can be across column boundaries or within columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Dept</th>
<th>Salary</th>
<th>City</th>
<th>Province</th>
<th>Postal_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zikopoulos</td>
<td>510</td>
<td>10000</td>
<td>Whitby</td>
<td>ONT</td>
<td>L4N5R4</td>
</tr>
<tr>
<td>Eatonopoulos</td>
<td>500</td>
<td>20000</td>
<td>Whitby</td>
<td>ONT</td>
<td>L4N5R4</td>
</tr>
</tbody>
</table>
Warehousing Queries on an I/O Bound System Test

- Chart shows the results of running the TPC-H workload on an I/O bound system
  - What BI systems aren’t I/O bound
  - Easy to fix a CPU bound system in a cluster
- Results are Win (storage savings) / Win (performance speedup)
  - Note: 100% means the workload runs the same as it did without compression
    - Therefore, Query 16 is 1% slower with compression
  - Average query speedup was 29% and storage savings was between 46%-70%
Trading CPU for I/O on an I/O Bound System Test

- Note the 50+% savings with row compression
- After compression, table scans run twice as fast
  - BUT the system is getting less MB/sec because of compression’s overhead
  - BUT you only have to get at ½ the pages, so the system runs faster

<table>
<thead>
<tr>
<th>Tablescan</th>
<th>No Compression</th>
<th>Row Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Rows</td>
<td>59,986,052</td>
<td>59,986,052</td>
</tr>
<tr>
<td># of Pages</td>
<td>1023162</td>
<td>426292</td>
</tr>
<tr>
<td>MB</td>
<td>7993</td>
<td>3330</td>
</tr>
<tr>
<td>Run time (secs)</td>
<td>90.993</td>
<td>43.268</td>
</tr>
<tr>
<td>MB/sec</td>
<td>87.847</td>
<td>76.972</td>
</tr>
<tr>
<td>Total user cpu time (sec.ms)</td>
<td>48.910</td>
<td>94.490</td>
</tr>
<tr>
<td>Time waited for prefectch</td>
<td>302878</td>
<td>67950</td>
</tr>
<tr>
<td>user cpu time/row</td>
<td>0.000000082</td>
<td>0.00000158</td>
</tr>
<tr>
<td>user cpu time/page</td>
<td>0.000048</td>
<td>0.000222</td>
</tr>
<tr>
<td>rows/page</td>
<td>59</td>
<td>141</td>
</tr>
</tbody>
</table>
Warehousing Queries on an CPU Bound System Test

- Chart shows the results of running the TPC-H workload on an CPU bound system
- Results aren't bad
  - Note: 100% means the workload runs the same as it did without compression
    - Therefore, Query 16 is 1% slower with compression
  - Query performance only slowed by 3% to achieve storage savings was between 46%-70%

<table>
<thead>
<tr>
<th>Query #</th>
<th>No Compression</th>
<th>Row Compression</th>
<th>% diff (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>226.8</td>
<td>286.7</td>
<td>126%</td>
</tr>
<tr>
<td>2</td>
<td>31.3</td>
<td>16.5</td>
<td>53%</td>
</tr>
<tr>
<td>3</td>
<td>130.0</td>
<td>159.1</td>
<td>122%</td>
</tr>
<tr>
<td>4</td>
<td>102.2</td>
<td>134.2</td>
<td>131%</td>
</tr>
<tr>
<td>5</td>
<td>122.2</td>
<td>161.9</td>
<td>132%</td>
</tr>
<tr>
<td>6</td>
<td>19.2</td>
<td>25.9</td>
<td>135%</td>
</tr>
<tr>
<td>7</td>
<td>108.3</td>
<td>125.9</td>
<td>116%</td>
</tr>
<tr>
<td>8</td>
<td>557.3</td>
<td>478.0</td>
<td>86%</td>
</tr>
<tr>
<td>9</td>
<td>222.4</td>
<td>283.0</td>
<td>127%</td>
</tr>
<tr>
<td>10</td>
<td>121.4</td>
<td>140.7</td>
<td>116%</td>
</tr>
<tr>
<td>11</td>
<td>97.0</td>
<td>38.8</td>
<td>40%</td>
</tr>
<tr>
<td>12</td>
<td>117.3</td>
<td>147.4</td>
<td>126%</td>
</tr>
<tr>
<td>13</td>
<td>185.7</td>
<td>186.0</td>
<td>100%</td>
</tr>
<tr>
<td>14</td>
<td>11.6</td>
<td>13.4</td>
<td>116%</td>
</tr>
<tr>
<td>15</td>
<td>18.2</td>
<td>21.0</td>
<td>115%</td>
</tr>
<tr>
<td>16</td>
<td>60.1</td>
<td>57.8</td>
<td>96%</td>
</tr>
<tr>
<td>17</td>
<td>90.3</td>
<td>77.2</td>
<td>85%</td>
</tr>
<tr>
<td>18</td>
<td>627.5</td>
<td>675.1</td>
<td>108%</td>
</tr>
<tr>
<td>19</td>
<td>864.1</td>
<td>727.1</td>
<td>84%</td>
</tr>
<tr>
<td>20</td>
<td>102.9</td>
<td>81.1</td>
<td>79%</td>
</tr>
<tr>
<td>21</td>
<td>354.6</td>
<td>467.3</td>
<td>132%</td>
</tr>
<tr>
<td>22</td>
<td>23.6</td>
<td>26.9</td>
<td>114%</td>
</tr>
<tr>
<td>Total Time</td>
<td>4194.0</td>
<td>4331.0</td>
<td>103%</td>
</tr>
</tbody>
</table>
### Statistics Before and After Compression

<table>
<thead>
<tr>
<th>TABNAME</th>
<th>CARD</th>
<th>NPAGES</th>
<th>AVG COMPRESSED ROWSIZE</th>
<th>AVGROW SIZE</th>
<th>AVG ROWSIZE</th>
<th>PCT ROWS COMPRESSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPRODUCT</td>
<td>608206</td>
<td>86936</td>
<td>0</td>
<td>+0.000000E+00</td>
<td>569</td>
<td>+0.000000E+00</td>
</tr>
<tr>
<td>TSALESFACT</td>
<td>1275908</td>
<td>70923</td>
<td>0</td>
<td>+0.000000E+00</td>
<td>218</td>
<td>+0.000000E+00</td>
</tr>
</tbody>
</table>

### Statistics After Compression

<table>
<thead>
<tr>
<th>TABNAME</th>
<th>CARD</th>
<th>NPAGES</th>
<th>AVG COMPRESSED ROWSIZE</th>
<th>AVGROW SIZE</th>
<th>AVG ROWSIZE</th>
<th>PCT ROWS COMPRESSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPRODUCT</td>
<td>608206</td>
<td>19301</td>
<td>125</td>
<td>+4.52530E+00</td>
<td>125</td>
<td>+1.000000E+02</td>
</tr>
<tr>
<td>TSALESFACT</td>
<td>1275908</td>
<td>13835</td>
<td>43</td>
<td>+5.02075E+00</td>
<td>43</td>
<td>+1.000000E+02</td>
</tr>
</tbody>
</table>

- **81% Less Pages**
- **78% Less Pages**
- **81% Smaller Rows**
- **78% Smaller Rows**
Space Savings from Compression

Overall Storage Savings: Uncompressed Data versus Compressed Data

SALES Fact Table: 81% Smaller
PRODUCT Dimension Table: 79% Smaller
Performance Speedup from Compression

Overall Workload Performance:
Uncompressed Data versus Compressed Data

Uncompressed Data | Compressed Data

42% Faster

IBM Information On Demand
No limits: Large Row Identifiers

- Increase in table size limits and rows per page
  - Tablespace level definition
  - DMS Tablespace only
  - Default for DMS Tablespaces

- ALTER TABLESPACE <name> CONVERT TO LARGE
  - Tablespace is locked, definition is modified and catalogues are updated
  - Indexes will need to be reorganized
    - Every index for every table in the converted tablespace needs to be reorganized or rebuilt to convert the RID entries from regular to large
Large Tablespaces

- **Before DB2 9**
  - RIDs are 4 bytes: 3 byte page number and 1 byte slot number
  - Default table space data type was REGULAR
    
    ```sql
    CREATE TABLESPACE <tbspace-name>
    MANAGED BY [DMS | AUTOMATIC STORAGE | SMS]
    ```
  - Tables (data part) could not be placed in LARGE table spaces

- **DB2 9**
  - RIDs can be 6 bytes: 4 byte page number and 2 byte slot number
  - Enabled by creating a table in a LARGE DMS tablespace (also enabled in TEMP table spaces)
  - LARGE is now the default type for new DMS and AUTO STORAGE tablespaces
    
    ```sql
    CREATE TABLESPACE <tbspace-name>
    MANAGED BY [DMS | AUTOMATIC STORAGE]
    ```
Tablespace Limit Changes

- Page size: 4KB, 8KB, 16KB, 32KB
- Tablespace size: 64GB, 128GB, 256GB, 512GB, 4TB, 8TB, 16TB

Row ID (RID) 4 Bytes: 16M, ~255, 4x10^9 Rows
Row ID (RID) 6 Bytes: 512M, ~2K, 1.1x10^12 Rows
Large Tablespaces

- **Table Capacity**
  - Single table partition (or single un-partitioned table) can grow to 16TB
  - Partitioned table can grow to 16TB * 32K * 1000 (with 32K ranges and 1000 db partitions) which is ½ ZB which is 1,000,000 PB…get real

- **Table space Capacity and Management**
  - Single DMS tablespace containing tables enlarged by 32x: tablespace consolidation

- **Better storage and memory utilization**
  - Removes ~255 row / page limit
  - Especially useful when row size is less than pagesize/255 (eg. temp tables DB2 can use for RID sorting)

- **Side Effect**
  - Indexes will generally be larger
    - Two more bytes per index key
    - Note: the size of RIDs in indexes is dependent on the size of RIDs in the table (it has nothing to do with the type of table space where the index resides)

- **Best Practices**
  - If existing tables require 4-byte page numbers, and don’t require >255 rows/page, use online index REOG
  - If existing tables require both, use classic table reorg
Converting from REGULAR to LARGE

1. ALTER TABLESPACE <name> CONVERT TO LARGE
   - Allows the **tablespace** to allocate 4-byte page numbers instead of just 3
   - Allows newly created tables to use 6-byte RIDs (both 4-byte page numbers and 2-byte slot numbers)
   - Existing tables are restricted to small RIDs until they and their indexes are REORGED)

2. Enable tablespace to exploit larger capacity (If necessary)
   - If table space is defined with AUTORESIZE YES
     • If MAXSIZE is NONE, then growth of the table space is automatic!
     • Else MAXSIZE is restricting table space growth and should be increased
   - Otherwise, storage has to be increased to benefit from a larger capacity
     • Enable AUTORESIZE, or add a new stripe set, or extend existing containers

3. Enable existing tables in the tablespace to exploit larger capacity and >255 rows/page (if necessary)
   - New tables created will fully support Large RIDs (both larger page numbers and slot numbers) so no additional action required to enable this
   - Existing tables restricted to 3 byte page #s until all indexes have been reorganized
   - Existing tables restricted to 255 rows/page until the table is reorganized with ‘classic’ REORG or reloaded
## Limit Relief

<table>
<thead>
<tr>
<th></th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tera = 10^{12} Peta = 10^{15} Exa = 10^{18} Zetta = 10^{21}</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keys per Index</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>Total Key Size (4K page)</td>
<td>1 KB</td>
<td>1 KB</td>
</tr>
<tr>
<td>Total Key Size (8K page)</td>
<td>1 KB</td>
<td>2 KB</td>
</tr>
<tr>
<td>Total Key Size (16K page)</td>
<td>1 KB</td>
<td>4 KB</td>
</tr>
<tr>
<td>Total Key Size (32K page)</td>
<td>1 KB</td>
<td>8 KB</td>
</tr>
<tr>
<td>Table Size (4K page)</td>
<td>64 TB</td>
<td>64 EB</td>
</tr>
<tr>
<td>Table Size (8K page)</td>
<td>128 TB</td>
<td>128 EB</td>
</tr>
<tr>
<td>Table Size (16K page)</td>
<td>256 TB</td>
<td>256 EB</td>
</tr>
<tr>
<td>Table Size (32K page)</td>
<td>0.5 PB</td>
<td>0.5 ZB</td>
</tr>
<tr>
<td>Rows per 1K Page (ceiling)</td>
<td>251</td>
<td>287</td>
</tr>
<tr>
<td>Rows per 4K Page (ceiling)</td>
<td>253</td>
<td>580</td>
</tr>
<tr>
<td>Rows per 8K Page (ceiling)</td>
<td>254</td>
<td>1165</td>
</tr>
<tr>
<td>Rows per 16K Page (ceiling)</td>
<td>251</td>
<td>2335</td>
</tr>
<tr>
<td>Some Object Names (eg Index)</td>
<td>18 B</td>
<td>128 B</td>
</tr>
</tbody>
</table>

New

Tera = 10^{12} Peta = 10^{15} Exa = 10^{18} Zetta = 10^{21}
Security

Paul C Zikopoulos, BA, MBA paulz@ca.ibm.com
DB2 Advanced Technical Expert (DRDA and Cluster)
DB2 Customer Solutions Expert (BI and DBA)
Security - Label Based Access Control

- **Label Based Access Control (LBAC)**
  - A “label” is associated with both user sessions and data rows
  - Rules for comparing users and data labels provide allow access controls to be applied at the row level

- **Labels may consist of multiple components**
  - Hierarchical, group or tree types
  - Row labels appear as a single additional column in a protected table
  - User labels are granted by a security administrator

- **Similar to the label security support in DB2 for z/OS v8**
LBAC Hierarchy – Tree
**LBAC Query**

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

Paul C Zikopoulos, BA, MBA paulz@ca.ibm.com
DB2 Advanced Technical Expert (DRDA and Cluster)
DB2 Customer Solutions Expert (BI and DBA)
Developers

Embrace and Support Established and Emerging Communities

- Visual Studio 2005 Support
  Shipped at Time of Release

- Viper Support
- DB2 for z/OS Support
- Over 27,000 Copies in '06...

- Over 10K DB2 Driver Downloads
- Over 3K IDS Driver Downloads

- Bringing PHP, SOA & Information Together
- DB2 Express-C Now Included

- New alphaWorks Toolkit 5/17/06
Application Development Philosophy

- All DB2 Editions run the same code base:
  - Same APIs
  - Same SQL
  - Same deployment

- Over 90% of our code is common across operating systems.

- The same application can access hundreds of gigabytes through Viper DB2 Express, or hundreds of terabytes through Viper DB2 Enterprise, without changing a line of application code.

- Edition upgrade is differentiated by scalability and performance potential.
Continued Deep Integration with VS .NET

- Deep integration in to VS.NET 2003 since 2003

- **Enhanced Support:**
  - DB2 Viper supports VS 2005
  - Supports access to all DB2 Servers including System i and System z
  - Delivered last November as developer’s release

- **Drivers:**
  - DB2 .NET Data Provider
  - DB2 Everyplace .NET Data Provider supports .NET Compact Framework
  - IDS .NET Data Provider

- **CLR Stored Procedures:**
  - DB2 has supported them for over a year
Deep integration into MS Visual Studio 2005
DB2 Developer Workbench

- The DB2 Developer Workbench is an Eclipse based tool replacing the DB2 Development Center
- XML Support
- Easy migration of existing projects
- Create a routine for DB2 for LUW then deploy to DB2 for z/OS.
- Stored procedure debugger
- Many other features to enhance deployment, packaging and Java support

Summary
On a whole, DB2 9 has really raised the bar for database development. Besides, who can go past the slick new Developer Workbench? With the workbench, IBM should be able to claw back some ground from developers who in the past may have looked past DB2 for a platform that was easier to develop on.
CREATE PROCEDURE count_job ( IN VJOB VARCHAR(30) )
  SPECIFIC count1
  DYNAMIC RESULT SETS 1

-- DB2 SQL Stored Procedure - Find available people

P1:  BEGIN
  -- Declare cursor
  DECLARE cursor1 CURSOR WITH RETURN FOR
  SELECT COUNT(ID) AS NUM_AVAIL
  FROM USER1.CONTACT_AVAIL
  WHERE TITLE = VJOB
  AND NOT
  XML EXISTS
  ('$c/schedule/activity/activitydate[start < current-date()][end > current-date()]' passing schedule as "c")

  -- Cursor left open for client application
  OPEN cursor1;
END P1

<table>
<thead>
<tr>
<th>Status</th>
<th>Action</th>
<th>Object Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>Run</td>
<td>COUNT_F</td>
</tr>
<tr>
<td>Success</td>
<td>Run</td>
<td>COUNT_F</td>
</tr>
<tr>
<td>Success</td>
<td>Run</td>
<td>COUNT_F</td>
</tr>
</tbody>
</table>
Visual XQuery Builder
Zend & IBM
Bringing PHP, SOA & Information Together

PHP is the most popular Web language in the world today:
70% of the UNIX-based, Apache Web platform market... 40% + of the overall market

- Simplified deployment package of PHP environment
- Enables and eases the development and deployment of dynamic, database-driven PHP applications

3,000+ Downloads
Quest & IBM
Viper Support and DB2 for z/OS Support Announced Today

Brings Familiar and Popular Development Tools to DB2

- Speeds DB2 developers using a proven application development solution
- Simplifies skills migration for developers needing to write code for more than one RDBMS
- Compliments Quest Central for DB2
- Provides an interactive community of peers

Development tools used by over 500,000 developers today…
BACKUP CHARTS
Deployment

- **Additional Features:**
  - Multiple versions and fix packs on the same Windows system
  - Full install images for Linux and Unix fix packs

- **Platforms**
  - Servers:
    - 64-bit - AIX, Windows Intel/AMD, Linux Intel/AMD, Linux PPC, Linux z-Series, Solaris, Sun IPF, HP PA-RISC, HP IPF
    - 32-bit – Windows Intel/AMD, Linux Intel/AMD
  - Clients
    - 32-bit & 64-bit – ALL
# DB2 Production Editions - Packaging

<table>
<thead>
<tr>
<th></th>
<th>Personal</th>
<th>Express</th>
<th>Workgroup</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Core” CD includes server + feature code</td>
<td><img src="image" alt="CD" /></td>
<td><img src="image" alt="CD" /></td>
<td><img src="image" alt="CD" /></td>
<td><img src="image" alt="CD" /></td>
</tr>
<tr>
<td>ECF determines “personality”</td>
<td>Personal ECF*</td>
<td>CPU ECF, User ECF</td>
<td>CPU ECF, User ECF</td>
<td>CPU ECF, User ECF</td>
</tr>
<tr>
<td>Features licensed separately</td>
<td>Performance Optimization</td>
<td>Performance Optimization</td>
<td>Storage Optimization, Performance Optimization</td>
<td>Homogenous Federation, DPF</td>
</tr>
<tr>
<td>ECF CDs</td>
<td>Workload Management</td>
<td>High Availability</td>
<td>Workload Management</td>
<td>High Availability</td>
</tr>
<tr>
<td></td>
<td>High Availability</td>
<td></td>
<td></td>
<td>Geodetic Data Mgmt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Advanced Access Control</td>
</tr>
</tbody>
</table>
# DB2 Connect Editions - Packaging

<table>
<thead>
<tr>
<th>“Core” CD includes server + feature code</th>
<th>Personal</th>
<th>Enterprise</th>
<th>Application</th>
<th>Unlimited zSeries</th>
<th>Unlimited for iSeries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DB2 Client</td>
<td>DB2 Connect Server</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ECF determines “personality”

- **Personal**: Authorized User and Concurrent User
- **Processor**: Host Server and MSUs
- **Managed Processor**: Managed Processor

Features included with Editions

- **Homogeneous Federation**
- **Homogeneous Federation**
- **Homogeneous Federation**
- **Homogeneous Federation**