



IBM Software Group – Information Management

# DB2 9 for z/OS

## Technical Overview & Migration

*Kevin Harrison,  
Senior IT Architect/Senior Software Engineer  
DB2 SWAT Team  
IBM Silicon Valley Lab  
kharrison@us.ibm.com*



**DB2 9 for z/OS**  
**Secure Information**  
**Services You Can Trust**

*Australia and ASEAN DB for z/OS Workshop Series*  
*August 2010*

## Topics

- Introduction
- 64-bit Evolution and Buffer Pool Enhancements
- pureXML
- SQL Enhancements
- Optimization Evolution
- Security Enhancements
- Application Design
- Indexing Enhancements
- Utilities Enhancements
- Miscellaneous Enhancements

## DB2 9 for z/OS Technology Themes

- ✓ **Enable high-volume transaction processing for next wave of Web applications**
- ✓ **Extend the lead in transaction processing availability, scalability and performance**
- ✓ **Reduce cost of ownership and System z-specific skill needs**
- ✓ **Improve data warehousing and OLTP reporting**

**DB2 9 for z/OS delivers on more than 225 requirements submitted by customers, business partners, and worldwide user group communities**

## DB2 9 for z/OS at a Glance – Very Significant Release

- SHRLEVEL(REFERENCE) for REORG of LOB tablespaces
- Online RENAME COLUMN
- Online RENAME INDEX
- Online CHECK DATA and CHECK LOB
- Faster REORG by intra-REORG parallelism
- More online REORG by eliminating BUILD2 phase
- LOB Locks reduction
- Skipping locked rows option
- Online REBUILD INDEX
- Renaming SCHEMA
- Renaming VCAT
- Tape support for BACKUP and RESTORE SYSTEM utilities
- Recovery of individual tablespaces and indexes from volume-level backups
- Enhanced STOGROUP definition
- Utility TEMPLATE switching
- Conditional restart enhancement: automatic search for the appropriate checkpoint
- Removing more reasons for 'soft' outages
- Thin DB2 Connect client
- Global query optimization
- Generalizing sparse index and in-memory data caching method
- Optimization Service Center
- Autonomic reoptimization
- Logging enhancements
- LOBs Network Flow Optimization
- Faster operations for variable-length rows
- NOT LOGGED tablespaces
- Index on expressions
- Universal Tablespaces
- Partition-by-growth tablespaces
- APPEND option at insert
- Autonomic index page split
- Different index page sizes
- Support for optimistic locking
- Faster and more automatic DB2 restart
- MODIFY RECOVERY enhancements
- RLF improvements for remote application servers such as SAP
- Preserving consistency when recovering individual objects to a prior point in time
- CLONE Table: fast replacement of one table with another
- Index compression
- DECIMAL FLOAT
- BIGINT
- VARBINARY, BINARY
- TRUNCATE TABLE statement
- MERGE statement
- FETCH CONTINUE
- ORDER BY and FETCH FIRST n ROWS in sub-select and full-select
- ORDER OF extension to ORDER BY
- Various scalar functions
- XML support in DB2 engine
- Enhancements to SQL Stored Procedures
- SELECT FROM UPDATE/DELETE/MERGE
- Enhanced CURRENT SCHEMA
- IP V6 support
- Unified Debugger
- Trusted Context
- Database ROLES
- Automatic creation of database objects
- Modify early code without requiring an IPL
- Utilities CPU reduction
- Temporary space consolidation
- . . .

**TCO Reduction**

**Continuous Operations**

**Performance**

**Scalability**

**SQL**

**Portability**



IBM Software Group – Information Management

# ***64-bit Evolution and Buffer Pool Enhancements***

***Virtual Storage Constraint Relief***

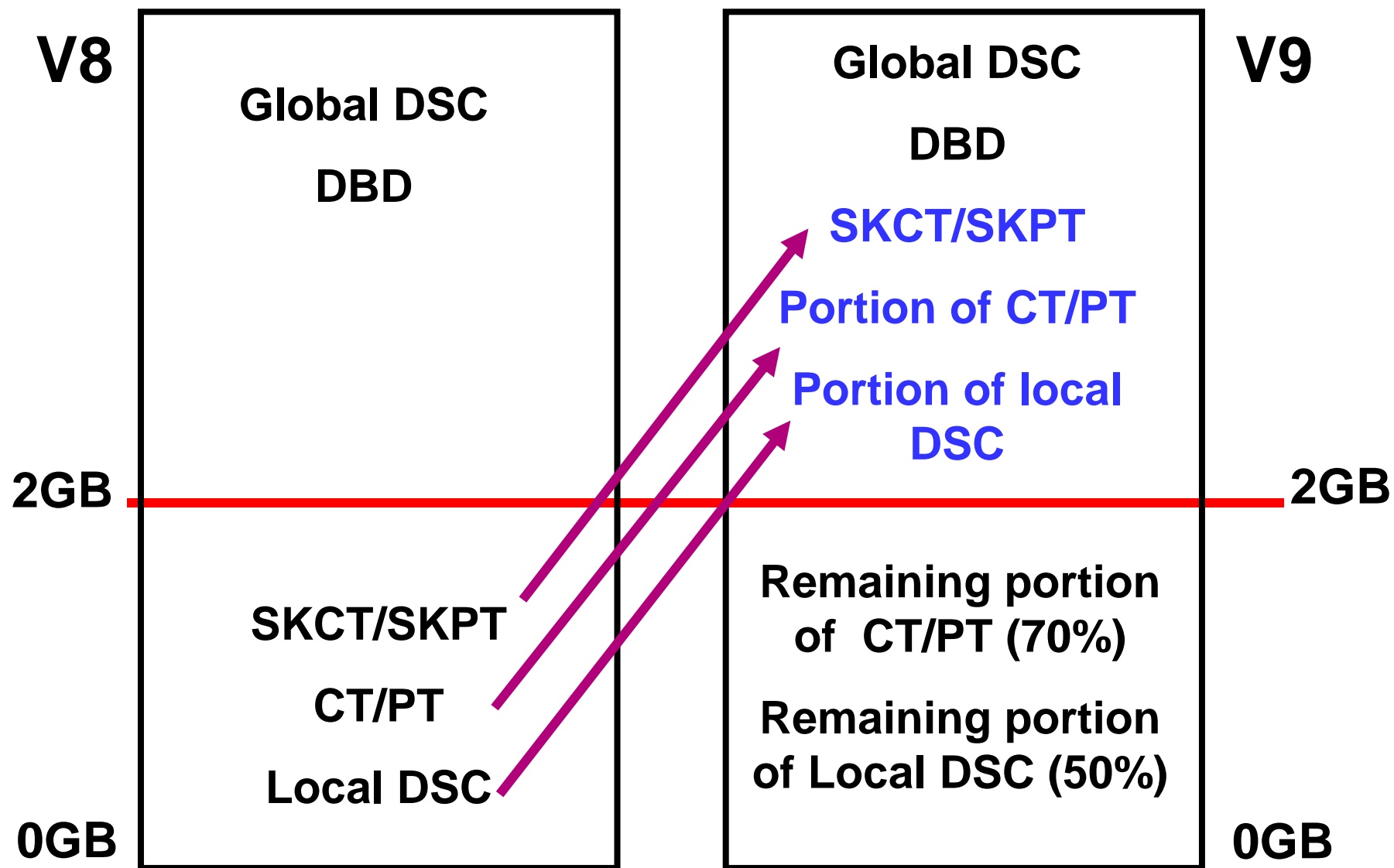
## DBM1 Virtual Storage below 2GB

- Three major storage areas still below 2GB in V8
  - ▶ EDM pool containing SKCT/SKPT and CT/PT
  - ▶ Local dynamic statement cache
  - ▶ Thread and stack storage
- EDM pool in DB2 9 – need REBIND
  - ▶ SKCT/SKPT moved above 2GB
  - ▶ A portion (close to 30%) of CT/PT moved above 2GB
  - ▶ Average estimated reduction of 60% but wide fluctuation from 20 to 90%
  - ▶ Plan on keeping to  $2 * 0.7 * CT/PT$  for EDM pool below 2GB

## DBM1 Virtual Storage below 2GB ...

- Local dynamic statement cache (KEEPDYNAMIC=YES)
  - ▶ Rough estimation of V9 = 50% of V8
- User thread storage, System thread storage, Stack storage
  - ▶ Current expectation of less than 10% difference overall
- V8 APAR PK20800 8/07 Display Thread(\*) Service(Storage) for agent-local virtual storage
- Potential reduction can range from 0 to 300MB depending on how much thread/stack storage usage

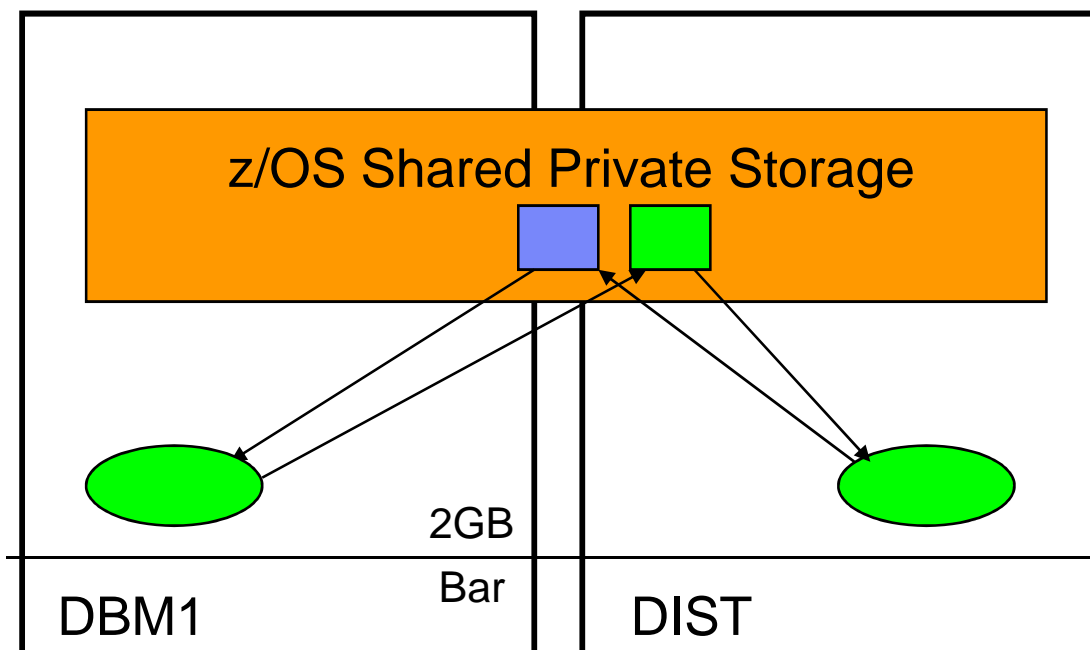
## DBM1 Virtual Storage below 2GB ...





## 64-bit DDF – Shared Private with DBM1

- DDF address space runs in 64-bit addressing mode
  - ▶ Shared 64-bit memory object avoids xmem moves between DBM1 and DDF and improves performance
  - ▶ VSCR

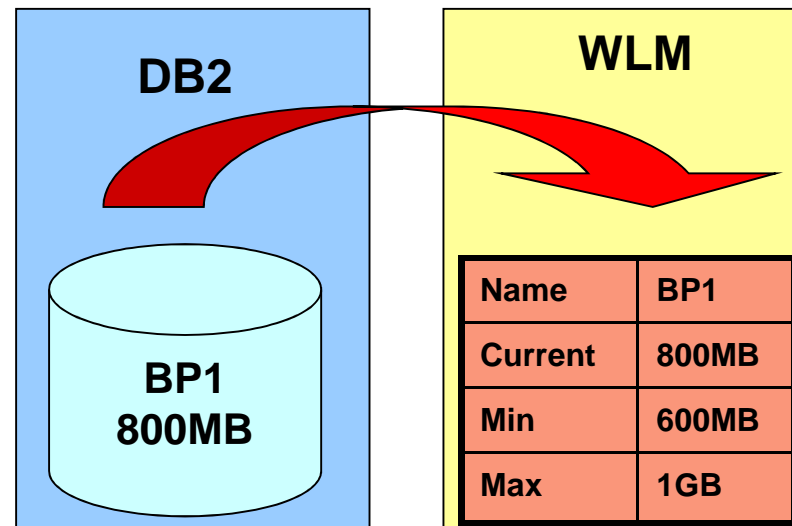


- Shared memory: new virtual storage type allowing multiple address spaces to share storage.
- Similar to ECSA – always addressable, avoids AR and XM moves.
- Different to ECSA – only available to those address spaces registering with z/OS to share this storage.
- Reduces data formatting and data movement
- Reduces virtual storage
  - ▶ It exists once, instead of in each address space

## WLM Buffer Pool Management

- WLM-assisted buffer pool management

- ▶ -ALTER BUFFERPOOL () AUTOSIZE(YES)
- ▶ z/OS 1.8
- ▶ DB2 registers BP to WLM and reports synch read I/O delays to WLM
- ▶ DB2 periodically reports BP hit stats to WLM
- ▶ WLM projects effect of adjusting BP size on workload performance goals
  - Takes into account overall system storage usage
- ▶ WLM drives DB2 exit to adjust size if appropriate
  - V9 restricts to +/- 25%



DSNB555I WLM RECOMMENDATION TO  
ADJUST SIZE FOR BUFFER POOL  
bpname HAS COMPLETED  
OLD SIZE = csize BUFFERS  
NEW SIZE = nsize BUFFERS

## Buffer Pool Management

- Efficient large BP support (>5GB)
  - ▶ z/OS1.8 RSM UIC update for LRU
  - ▶ More evenly balanced buffer hash chain
- Bigger prefetch and deferred write quantity for bigger buffer pool
  - ▶ Depends on
    - Page size
    - VPSEQT (in the case of prefetch)
    - SQL or utility, and type of prefetches
  - ▶ For example:
    - SELECT with tablespace scan invoking sequential prefetch using 4K BP, prefetch quantity doubled from 32 to 64 when VPSEQT\*buffers exceeds 40000
  - ▶ +36% MB/sec in non striped prefetch
  - ▶ +47% in 2-striped prefetch -> more effective striping
- Replace all sequential prefetch, except in tablespace scan, with dynamic prefetch in SQL calls
  - ▶ Up to 50% faster and more robust

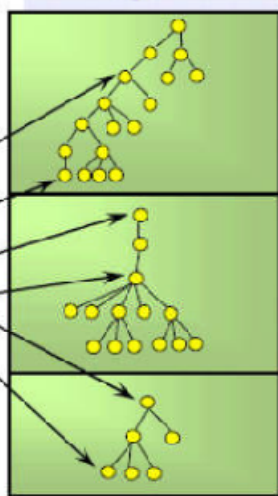
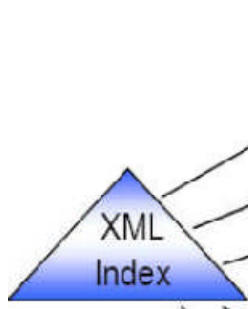


## pureXML Support

- SQL XML data type and native storage
- Designed specifically for XML
  - ▶ Supports XML hierarchical structure storage
  - ▶ Native operations and languages: XPath, SQL/XML, (XQuery in the future)
- Not transforming into relational
- Not using objects or nested tables
- Not using LOBs
- Integrated with relational engine, with all the utilities and tools support

# XML Storage – New

**DB2 9 pureXML:  
XML as XML**



*Maximum flexibility  
and performance*

XML Column

```

<PurchaseOrder PoNum="11223344"
  OrderDate="2007-07-18" Status="Unshipped">
  <client>
    <companyname> MyComp Ltd </companyname>
    <address>
      <line> 10, East Main Street </line>
      <line> 1234 Southern City </line>
      <line> Mr. Smith </line>
    </address>
  </client>
  <items>
    <item>
      <partid> 100-100-01 </partid>
      <name> Snow Shovel, Basic 22 inch </name>
      <quantity> 4 </quantity>
      <price> 9.99 </price>
    </item>
    <item>
      <partid> 100-103-01 </partid>
      <name> Snow Shovel, Super Deluxe 26 inch </name>
      <quantity> 2 </quantity>
      <price> 49.99 </price>
    </item>
  </items>
  <comment>
    Purchase order number, partid and
    quantity must show on each carton
  </comment>
</PurchaseOrder>
  
```

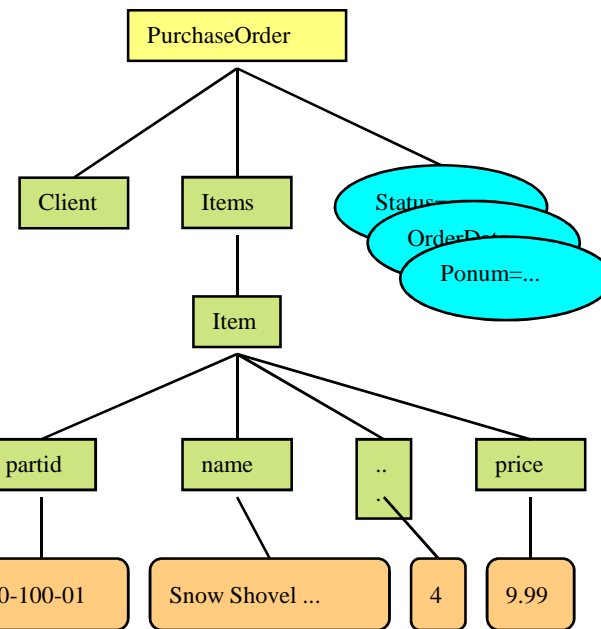
Document Node

Element Node

Text Node

Attribute Node

**Parsing**

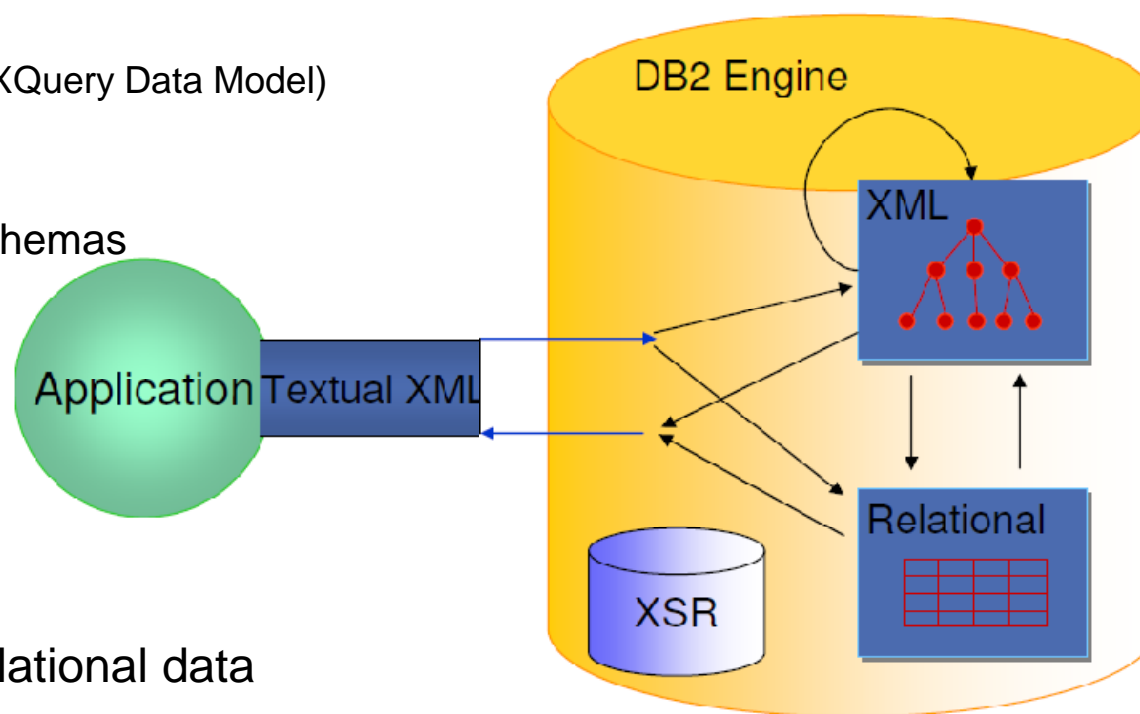


**Serialization**

## What You Can Do with DB2 for z/OS pureXML™

### *Managing XML data the same way as relational data*

- Create tables with XML columns or alter table add XML columns
  - ▶ Native XML storage
    - Hierarchical data model: XDM (XQuery Data Model)
- Insert XML data
  - ▶ Optionally validated against schemas
- Create indexes on XML data
- Efficiently search XML data
  - ▶ SQL/XML and XPath
- Extract XML data
- Decompose XML data into relational data
- Construct XML documents from relational and XML data
- All the utilities and tools support for XML



## SQL/XML – Work in both worlds

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

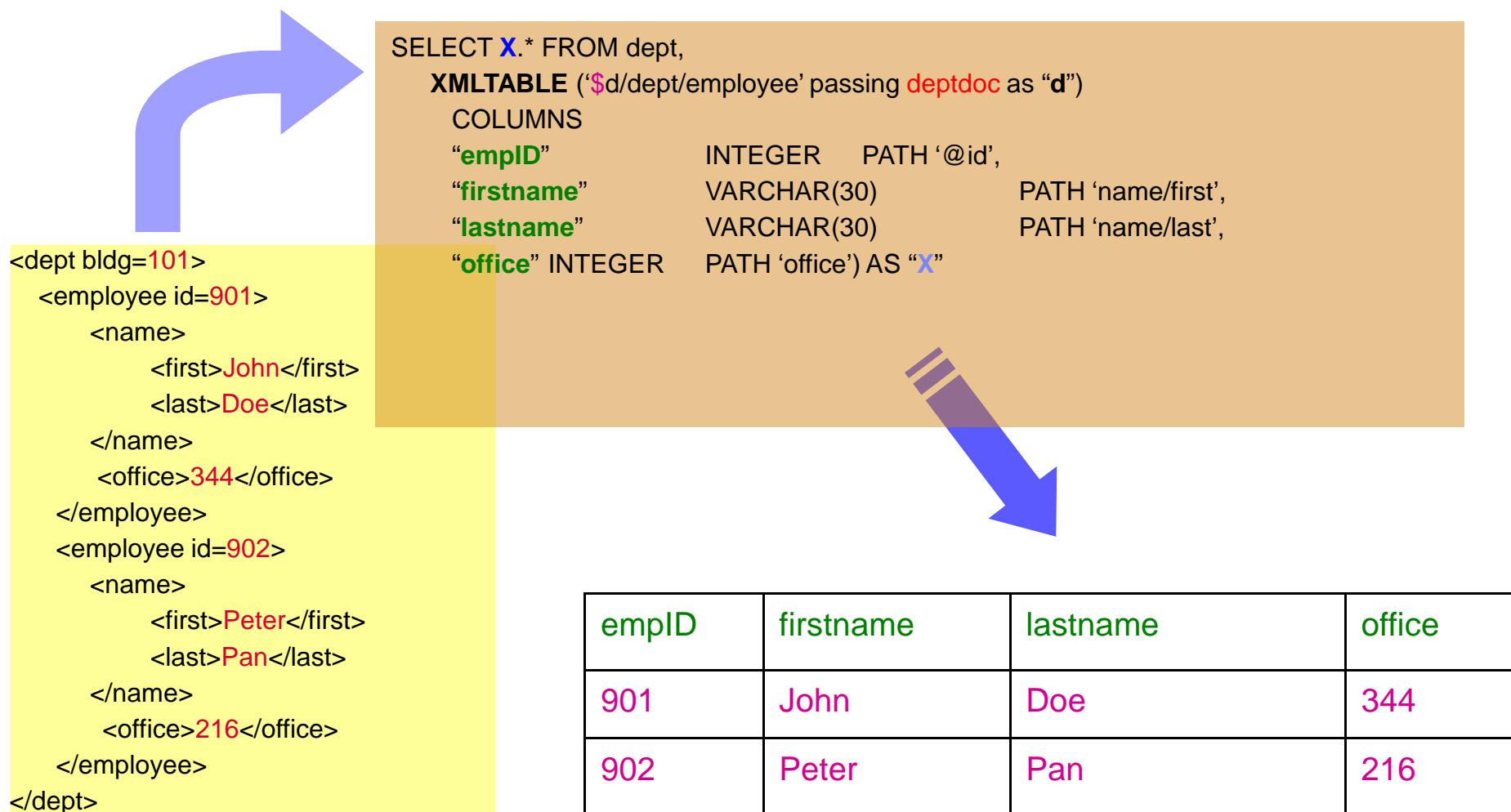


```
select d.deptID , u.headcount, xmlquery('$deptdoc/dept/name' passing d.deptdoc as "deptdoc")
  from dept d, unit u
 where d.deptID = u.unitID and u.headcount > 200
    and xmlexists('$deptdoc/dept[@bldg = $b]' passing d.deptdoc as "deptdoc", u.bldg as "b")
    and xmlexists('$deptdoc/dept/employee/name'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



## More XML support in DB2 9 in APARs

### XMLTABLE: return XML in tabular format



## Utilities

*Enhanced to handle new XML type, XML tablespaces, and XML indexes*

- CHECK DATA
- CHECK INDEX
- COPY INDEX
- COPY TABLESPACE
- COPYTOCOPY
- LISTDEF
- LOAD
- MERGECOPY
- QUIESCE TABLESPACESET
- REAL TIME STATISTICS
- REBUILD INDEX
- RECOVER INDEX
- RECOVER TABLESPACE
- REORG INDEX
- REORG TABLESPACE
- REPORT TABLESPACESET
- UNLOAD
- Basic RUNSTATS

## DB2 pureXML Advantages

- Directly store XML, no decomposition/composition, no normalize/de-normalize
- Eliminates database schema evolution bottleneck
- Declarative language, reduce complexity, dramatically improve application development productivity
- Native processing, high performance
- Manage XML data with ACID properties, auditing and regulatory compliance, together with relational data.
- Use XML as a powerful data model, with powerful declarative query language
- Managing large volumes of XML data as for relational data
- Integrate new XML data with existing relational data
- Publish (relational) data as XML
- Exploit database support for web applications, SOA, web services (SOAP)
- Benefit from unparalleled reliability, availability, scalability

## pureXML Usage Scenarios

- Directly processing XML
  - ▶ UNIFI, ACORD, FIXML, FpML, MIMSO, XBRL,
  - ▶ DJXDM, HR-XML, HL7, ARTS, HIPAA, NewsML, XForms
  - ▶ Insurance quotation/policy, contract, purchase order, emails etc
- Versatile schemas that are diverse and evolve and end-user customizable applications
- Sparse attribute values (null vs. absence)
- Object persistence (single column vs. many tables)
- Migration from legacy data model (network, hierarchical, relational)
- Generating web pages: XHTML
- Provide/consume Web Services (SOAP), support SOA



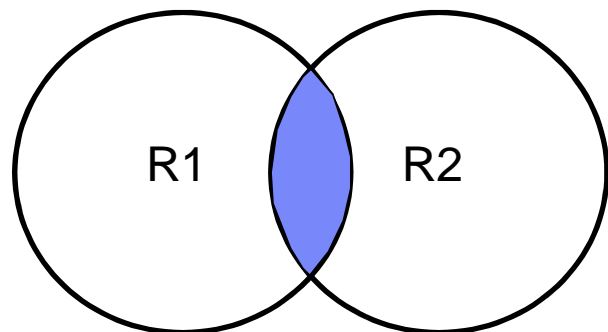
IBM Software Group – Information Management

# ***SQL Enhancements***

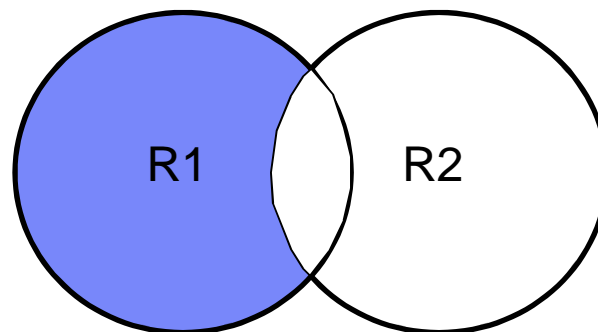
## SQL Enhancements

- INTERSECT, EXCEPT
- INSTEAD OF triggers
- MERGE
- SELECT FROM MERGE / UPDATE / DELETE
- TRUNCATE
- New data types
- New SQL functions
- Native SQL Procedures

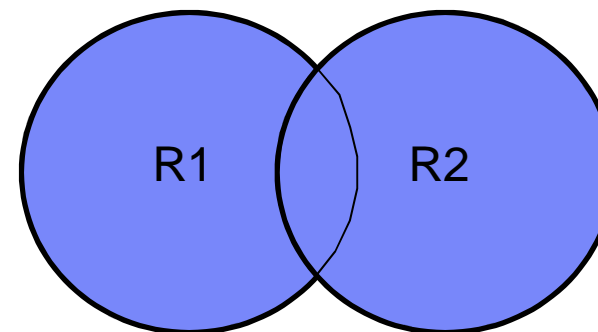
# INTERSECT/EXCEPT



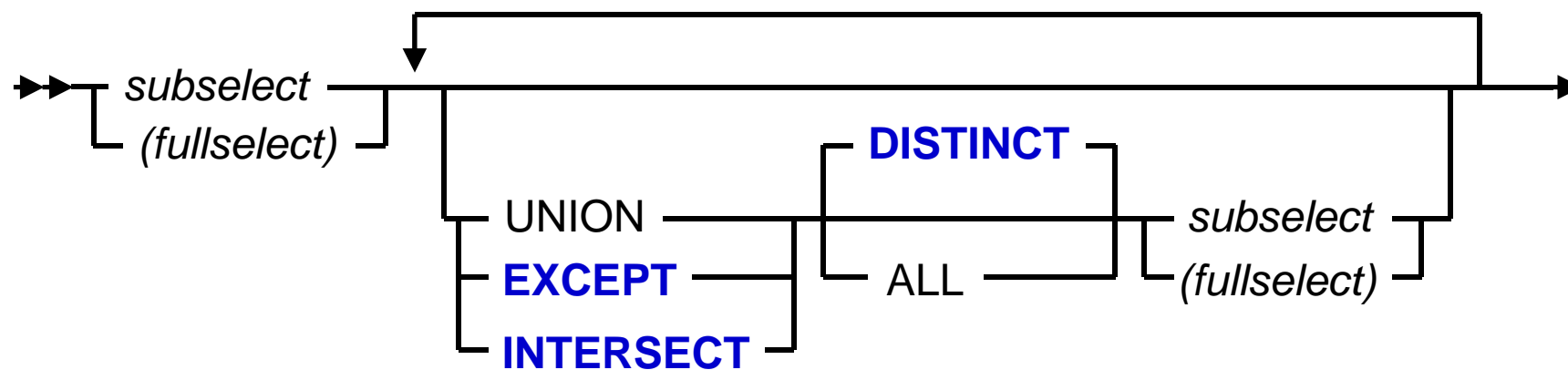
INTERSECT



EXCEPT



UNION

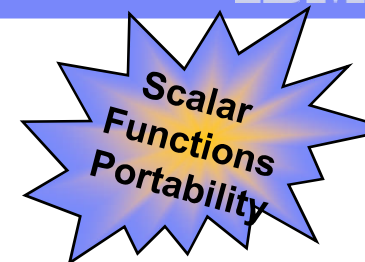


## Built-in Functions

- RANK
  - ▶ Generates a row number defining the rank as 1 plus the number of rows that precede the row.
  - ▶ If two or more rows are not distinct with respect to the ordering, there will be one or more gaps in the sequential rank numbering. Also referred to as Olympic Ranking.
- DENSE\_RANK
  - ▶ Similar to RANK but ensures there are no gaps.
- ROW\_NUMBER
  - ▶ Creates a sequential row number that is computed for each row starting with 1.
  - ▶ No number skipping.
- Even though these functions are ordered, cursors remain updatable.



## SOUNDEX & DIFFERENCE Scalar Functions



- Powerful scalar functions for name identification
- SOUNDEX
  - ▶ The SOUNDEX function returns a 4 character code that represents the sound of the words in the argument.
  - ▶ The result can be used to compare with the sound of other strings.
  - ▶ The SOUNDEX function is useful for finding strings for which the sound is known but the precise spelling is not.
- DIFFERENCE
  - ▶ A function that returns a value from 0 to 4
  - ▶ Represents the difference between the sounds of two strings based on applying the SOUNDEX function to the strings.
    - A value of 4 is the best possible sound match.

## INSTEAD OF Triggers

- For consistent use of views with SELECT, INSERT, UPDATE, and DELETE
  - ▶ Many views are not updatable, so customers have to access base tables for data changes. Triggers can be used to help control updates.
- Processed instead of the UPDATE, DELETE or INSERT statement that activated the trigger
- Can only be defined on views at the current server
  - ▶ provides an extension to the updatability of views
  - ▶ requested update operation against the view gets replaced by the trigger logic
  - ▶ application still believes all operations are performed against the view
  - ▶ applicable even for updatable views

## MERGE

- “Upsert”
  - ▶ A combination of insert and update, MERGE is a single SQL operation
  - ▶ Single row or multi-row
- If a row matches the ON condition it is updated,
- Otherwise it is inserted

```
MERGE INTO account AS T
USING VALUES (:hv_id, :hv_amt) FOR 5 ROWS AS S(id,amt)
ON T.id = S.id
WHEN MATCHED THEN
    UPDATE SET balance = T.balance + S.amt
WHEN NOT MATCHED THEN
    INSERT (id, balance) VALUES (S.id, S.amt)
NOT ATOMIC CONTINUE ON SQL EXCEPTION;
```

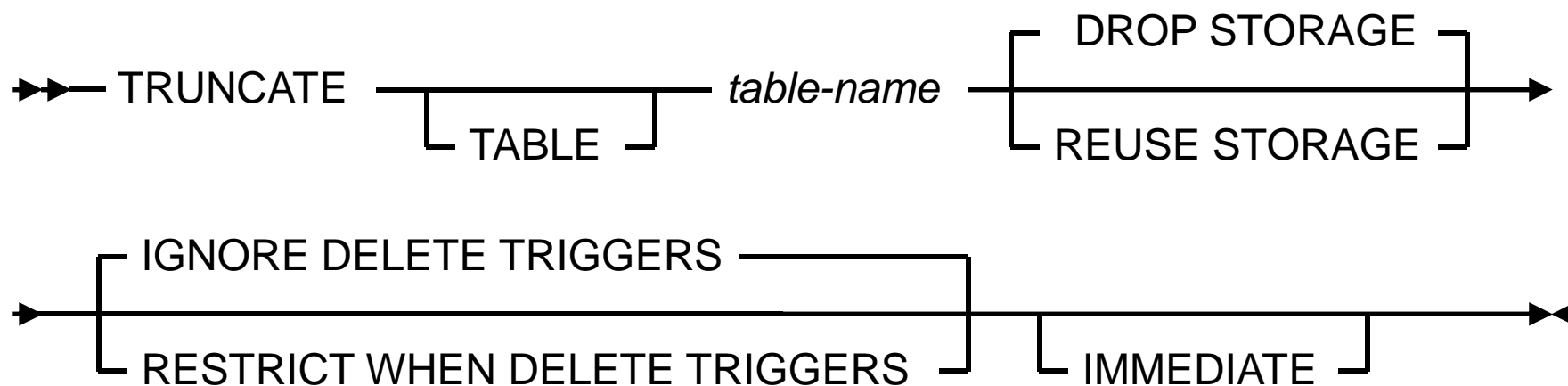
## SELECT FROM UPDATE/DELETE/MERGE

```
SELECT athlete, score FROM FINAL TABLE
  (UPDATE events SET SCORE=SCORE+10
   WHERE EVENT='BOXING');
SELECT MEMBER_ID, UPSERT_INDICATOR FROM FINAL TABLE
  (MERGE INTO MEMBER_PROFILE AS A
   INCLUDE (UPSERT_INDICATOR CHAR(1))
   USING (VALUES (20, 'PAUL', 22) ) AS B
   (MEMBER_ID, MEMBER_NAME, MEAL_PREFERENCE)
   ON (A.MEMBER_ID = B.MEMBER_ID)
   WHEN MATCHED THEN
     UPDATE SET A.MEMBER_NAME = B.MEMBER_NAME,
               A.MEAL_PREFERENCE = B.MEAL_PREFERENCE,
               UPSERT_INDICATOR = 'U'
   WHEN NOT MATCHED THEN
     INSERT (MEMBER_ID, MEMBER_NAME, MEAL_PREFERENCE, UPSERT_INDICATOR)
     VALUES (B.MEMBER_ID, B.MEMBER_NAME, B.MEAL_PREFERENCE, 'I')
   NOT ATOMIC CONTINUE ON SQLEXCEPTION);
```

- First example shows SELECT returning data from the nested UPDATE
- Second example shows SELECT returning data from the nested MERGE with an INCLUDE column
  - ▶ This include column returns the compare column from the ON clause and an indicator of whether the row was Updated or Inserted.

# TRUNCATE

- Allows fast delete of all rows in base tables or declared global temporary tables
  - ▶ Simple, segmented, partitioned, universal table spaces
  - ▶ If table contains LOB or XML columns also truncates auxiliary table spaces
  - ▶ IMMEDIATE option – **operation cannot be rolled back**
    - Allows immediate reuse of allocated space for subsequent insert operations in the same UOW without committing
- Deletes rows without firing DELETE triggers
- Option to REUSE or DROP STORAGE



## SKIP LOCKED DATA



- Rows with incompatible locks by other transactions are skipped
- Clause available
  - ▶ On SELECT INTO, PREPARE, searched UPDATE, searched DELETE, UNLOAD
  - ▶ Effective when CS or RS is in use
    - Otherwise it is ignored
  - ▶ Data is locked at the row or page
- QW0018SK – ROWS SKIPPED DUE TO INCOMPATIBLE LOCK HELD
  - ▶ Reported in IFCID 018
- Logic / Scenario
  - ▶ When a transaction needs to find work to do, regardless of order.
  - ▶ Messaging applications without strict ordering requirements expect to be able to skip over records that are locked by other transactions

## New Data Types: DECFLOAT and BIGINT

- DECFLOAT

- ▶ Well suited to typical customer financial calculations
- ▶ Similar to “calculator” mathematics
  - Eliminates rounding errors by using base 10 maths
  - Has up to 34 digits of precision
- ▶ DECIMAL(16)
  - $10^{+384}$  to  $10^{-383}$  Positive & Negative
- ▶ DECIMAL(32)
  - $10^{+6144}$  to  $10^{-6143}$  Positive & Negative
- ▶ Floating point convenience with fixed point precision!



- BIGINT

- ▶ An exact numeric capable of representing 63-bit integers
- ▶ -9223372036854775808 to 9223372036854775807

## New Data Types: BINARY & VARBINARY

- BINARY fixed-length binary string
  - ▶ 1 to 255 bytes
- VARBINARY variable-length binary string
  - ▶ 1 to 32704 bytes; maximum length determined by the maximum record size associated with the table
- Compatible with BLOBs
- Not compatible with character string data types
  - ▶ Similar to FOR BIT DATA character strings
  - ▶ Can use CAST to change FOR BIT DATA character string into binary string
- Prior to V9, FOR BIT DATA column use character based padding





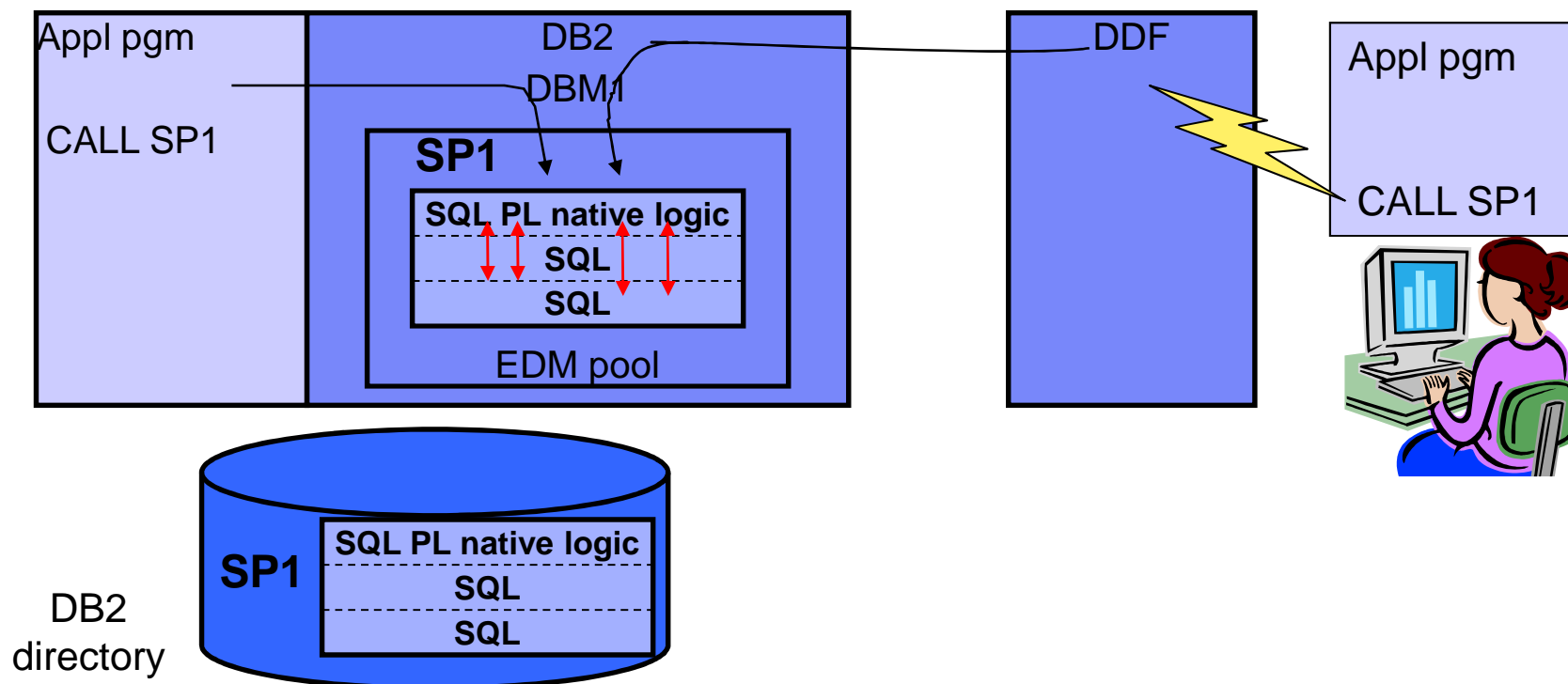
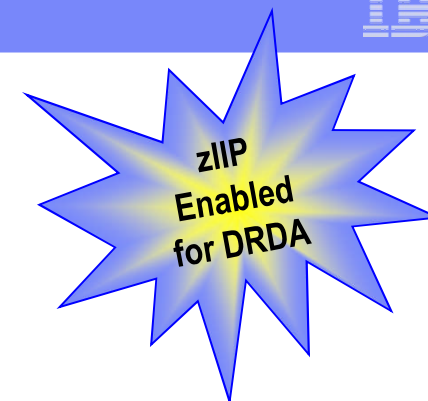
## Built-in Functions



- OLAP
  - ▶ RANK
  - ▶ DENSE\_RANK
  - ▶ ROW\_NUMBER
- Date and Time
  - ▶ TIMESTAMPADD
  - ▶ TIMESTAMP\_ISO
  - ▶ ...
- Sound representation
  - ▶ SOUNDEX
  - ▶ DIFFERENCE
- String handling
  - ▶ ASCII\_STR
  - ▶ EBCDIC\_STR
  - ▶ UNICODE\_STR
  - ▶ ASCII\_CHR
  - ▶ EBCDIC\_CHR
  - ▶ ...
- And More ...

## Native SQL Procedural Language

- Eliminates generated C code and compilation
- Fully integrated into the DB2 engine



## Native SQL Procedural Language ...

- Extensive support for versioning
  - ▶ VERSION keyword on CREATE PROCEDURE
  - ▶ CURRENT ROUTINE VERSION special register
  - ▶ ALTER ADD VERSION
  - ▶ ALTER REPLACE VERSION
  - ▶ ALTER ACTIVATE VERSION
- BIND PACKAGE with new DEPLOY keyword
- Enhanced SQL PL support and portability
  - ▶ New constructs only available for native SQL procedures
    - Nested compound statements
    - Multiple general conditions on a handler
    - FOR loop
    - Extended GOTO
  - ▶ More data types
    - BIGINT, BINARY, VARBINARY, DECFLOAT

*An SQL procedure created without FENCED or EXTERNAL is a native stored procedure*

## SQL Stored Procedures

### ▶ SQL Stored Procedure terminology change

- "SQL procedures" in Version 8 are called "external SQL procedures" in Version 9.1.
- Version 9.1 also introduces a new type of SQL procedure called native SQL procedures. You cannot create and use native SQL procedures until Version 9.1 new-function mode.

### ▶ External SQL procedures:

- Generated C program which runs in a WLM environment
- Not zIIP eligible

### ▶ Native SQL procedures:

- The SQL procedure logic runs in the DBM1 address space
- zIIP eligible ~ 55% redirect

## DB2 for z/OS V9 zIIP Redirect

- Other V9 zIIP exploitation
  - ▶ Global Query Optimization with parallel query
  - ▶ Pair Wise Join
- Native SQL Procedure Benchmark
  - ▶ <http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD104524>



## Plan Stability

- What is the problem?
  - ▶ REBINDs can cause access path changes
  - ▶ Most of the time, this improves query performance ...
  - ▶ ... But when it doesn't
    - No easy way to undo the REBIND
    - Can lead to a lot of grief to our customers and to IBM
- Existing “solutions” inadequate
  - ▶ Preventing REBINDs altogether
  - ▶ REBINDing into alternate collections
  - ▶ Using hints

## Plan Stability ...

- What is the solution?
  - ▶ At REBIND, DB2 will save old copies of packages
  - ▶ In the event of a performance regression, users will have a way to fallback to an older copy
- New function of DB2 9 (PK52523)
  - ▶ Protects customers against access path regression
  - ▶ Allows for a “safe” way to REBIND (fall back)
  - ▶ Available even in DB2 9 (CM) as it can benefit migration and fallback
  - ▶ Strongly recommended
  - ▶ Make sure that the pre-conditioning APAR for Plan Stability (PK52522) is applied on all V8 (NFM) systems



## Dynamic Index AND-ing for Star Join (Pair-Wise Join with Join Back)

- Multi-index access steps are considered independent
- Apply filtering to dimension tables before the fact table
  - ▶ Exploit single and/or multi-column fact table indexes
- Runtime assessment of filtering
  - ▶ Pre-fact dimensions with poor filter factors can be discarded at runtime and accessed post-fact
- Independent join of each dimension table to fact table via index
- The result of each pair-wise join is a set of fact table rids
- Perform Rid Sort and Rid Merge (ANDing) to generate final fact table rid list
- Final Rid list then used to retrieve data from Fact table
- Join back to dimension table(s) as necessary.

## Histogram Statistics - RUNSTATS

- V8 – DB2 has data skew awareness for single values
- Histogram statistics addresses skews across ranges of data values
- Summarizes data distribution on an interval scale
- DB2 uses equal-depth histograms
  - ▶ Each quantile has about the same number of rows
  - ▶ Example - 1, 3, 3, 4, 4, 6, 7, 8, 9, 10, 12, 15 (sequenced), cut into 3 quantiles

Seq No	Low Value	High Value	Cardinality	Frequency
1	1	4	3	5/12
2	6	9	4	4/12
3	10	15	3	3/12

## Histogram Statistics - RUNSTATS Notes

- RUNSTATS

- ▶ Maximum 100 quantiles for a column
- ▶ Same value columns WILL be in the same quantile
- ▶ Quantiles will be similar size but:
  - Will try and avoid big gaps between quantiles
  - A column high value and low value may have separate quantiles
  - Null WILL have a separate quantile
- ▶ If less than 100 column values, reverts to Distribution Stats
- ▶ Not supported with LOAD and REORG
- ▶ For indexes with a mixed order
  - Can only collect Histograms on prefix column of the same order

- Supports column groups as well as single columns

## Global Query Optimization

```
SELECT * FROM T1  
WHERE EXISTS  
(SELECT 1 FROM T2,T3 WHERE T2.C2 = T3.C2 AND T2.C1 = T1.C1);
```

- For subqueries that cannot be transformed into joins
- Cross Query Block Optimization
  - ▶ DB2 will break this query into 2 parts
    - correlated subquery and the outer query
    - optimized independently in V8 and prior
  - ▶ Global Query Optimization will allow DB2 to optimize a query as a whole rather than as independent parts
    - considers the effect of one query block on another
    - considers reordering query blocks

## Global Query Optimization

- Virtual Tables
  - ▶ A new abstraction to internally represent subqueries
  - ▶ Gives the optimizer the capability to assess the subquery as a table and assess it in different sequences
  - ▶ Will show in the PLAN\_TABLE if materialized
    - Table Type of “W” (Workfile)
    - Name of DSNVT(xx) where “xx” is the Subquery Query Block No
- Other considerations
  - ▶ INSERT, UPDATE, and DELETE have the same support for subquery to join transformation
    - V8 is limited to SELECT
  - ▶ Support for Optimization Hints

## Cluster Ratio Formula Enhancement

- New Cluster Ratio formula using DATAREPEATFACTOR
  - ▶ The number of data pages repeatedly scanned following index key order
  - ▶ Defaulted to enabled via ZParm STATCLUS=ENHANCED
  - ▶ STATCLUS=STANDARD results in V8 clustering statistics
- Where is it?
  - ▶ SYSINDEXES / SYSINDEXES\_HIST
  - ▶ SYSINDEXSTATS / SYSINDEXSTATS\_HIST
- Refines Cluster Ratio formula to determine the density of sequential reads
  - ▶ More accurately predicts the likelihood of triggering dynamic prefetch
  - ▶ Sequential prefetch is not used in DB2 9 for data page access via an index
- Recommend collection of RUNSTATS before REBIND while in Conversion Mode
  - ▶ Otherwise, 2 access path migrations may be experienced
    - 1: New access paths during CM (or first) REBIND
    - 2: First REBIND after RUNSTATS
- Make sure you have PK51450 (2/2/2008)

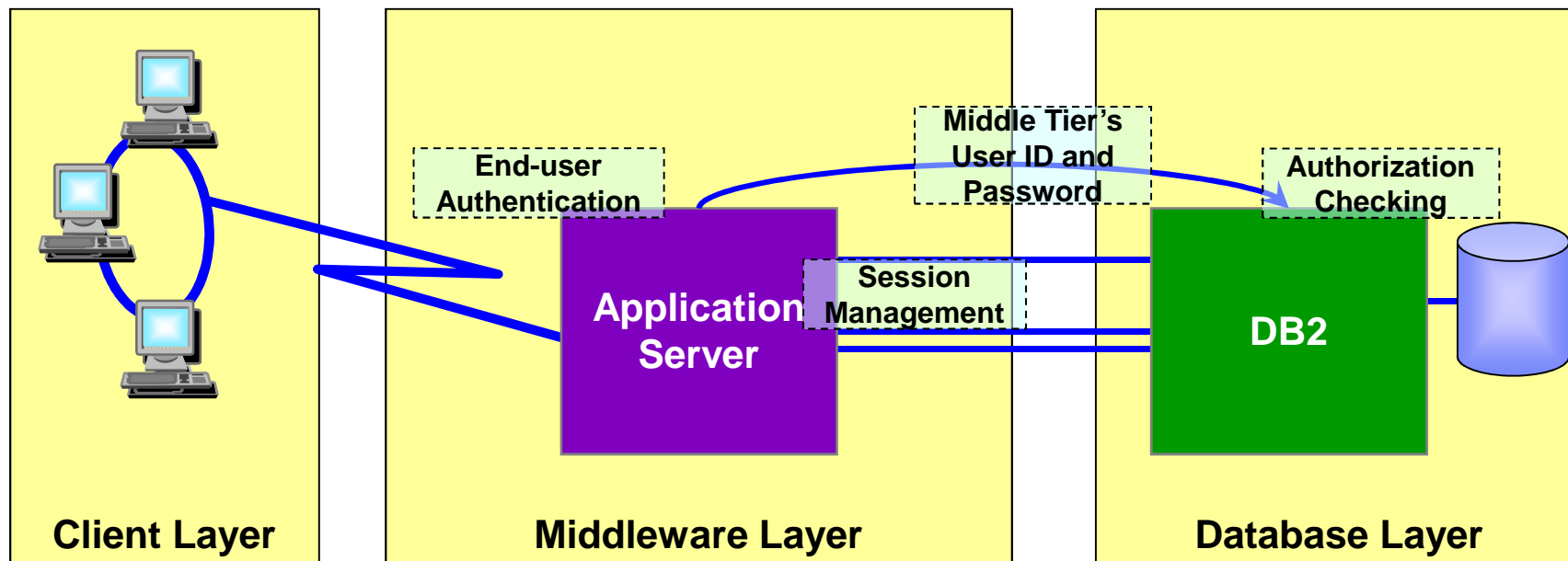


IBM Software Group – Information Management

# ***Security Enhancements***

## ***Trusted Contexts and Roles***

## Current Authentication in a Multi-tier Architecture



- A three-tiered application model with DB2 as the database server:
  - ▶ The middle layer authenticates users running client applications
  - ▶ It also manages interactions with the database server
  - ▶ The middle layer's user ID and password are used for database authentication
  - ▶ The privileges of the associated authorization id are checked when accessing the database, including all access on behalf of all end-users



## Trusted Contexts – An Introduction

- A **TRUSTED CONTEXT** establishes a trusted relationship between DB2 and an external entity such as a middleware server. For example:
  - ▶ WebSphere Application Server
  - ▶ Lotus Domino
  - ▶ SAP NetWeaver
  - ▶ PeopleSoft
- ▶ A set of *trust attributes* is evaluated to determine if a specific context is to be trusted
- A trusted context allows the external entity to use a database connection under a different user ID without the database server authenticating that ID
- It also allows an AUTHID to acquire database privileges associated with that trusted context, and not available outside it, via a **ROLE**

WebSphere softwareLotus softwareSAPPEOPLE  
Soft

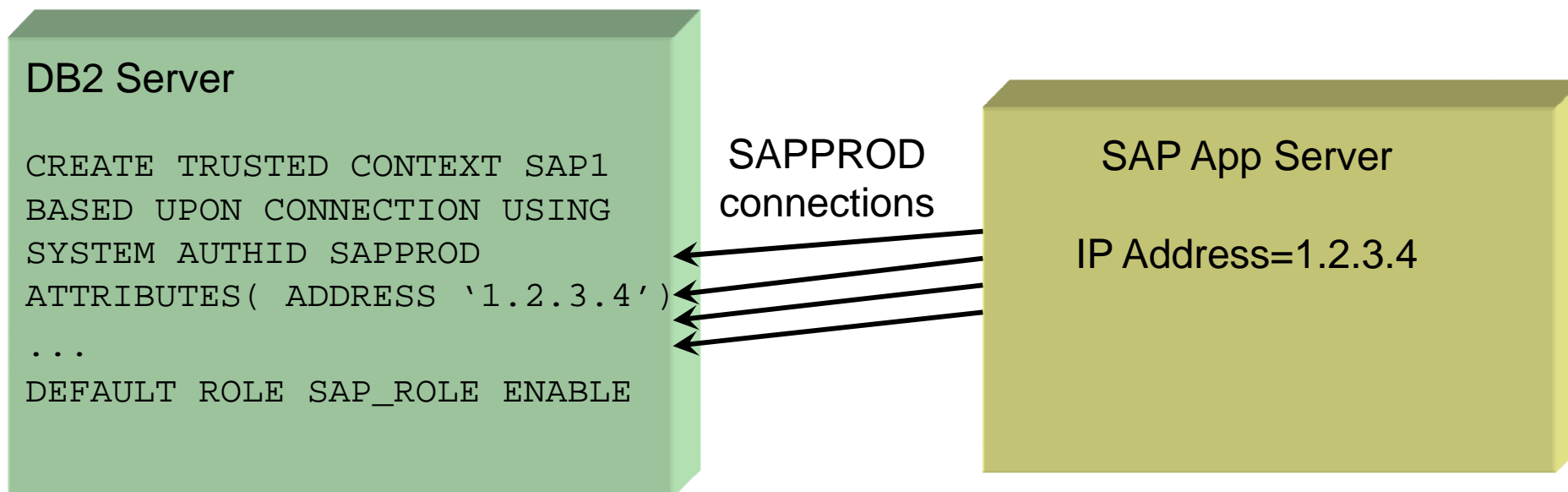
## Roles and Context-specific Privileges

- **Roles** provide the flexibility to grant privileges to an AUTHID only when the user is connected via a trusted context
- They greatly simplify management of authorization
- An individual **role** can be defined for any AUTHID using the trusted connection, in which case the user inherits the privileges granted to the individual **role**
- Where there is no individual **role**, any AUTHID using a trusted context inherits the privileges of the trusted context's default **role**, if defined

```
CREATE TRUSTED CONTEXT CTX1  
BASED UPON CONNECTION USING SYSTEM AUTHID WASADM1  
DEFAULT ROLE CTXROLE  
ATTRIBUTES (ADDRESS '9.67.40.219')  
ENABLE  
WITH USE FOR JOE ROLE JROLE;
```

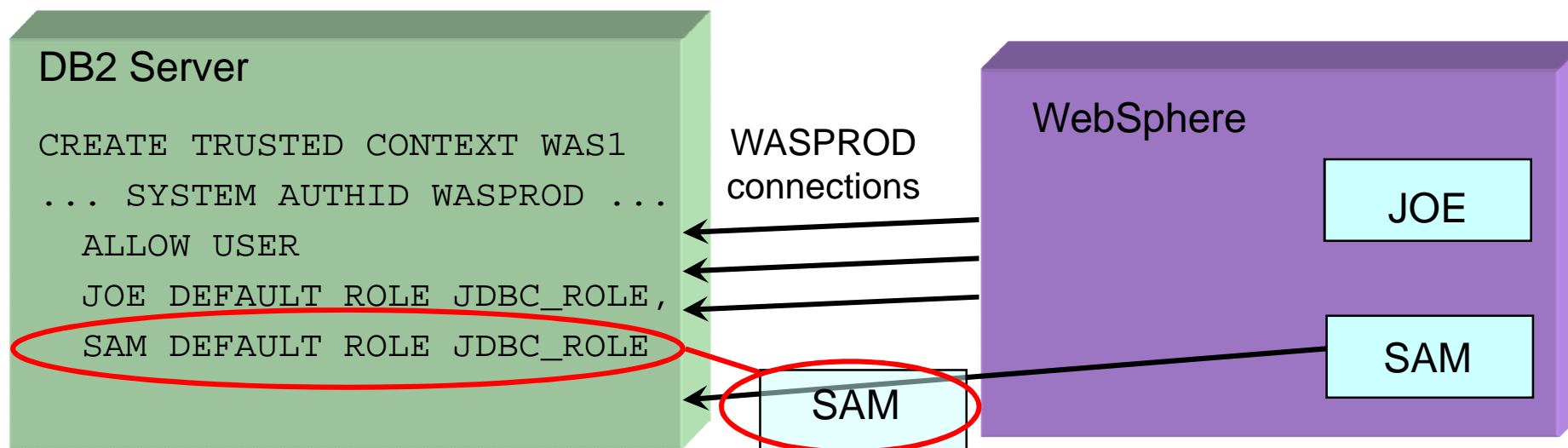
## Example 1: Securing An Application Server

- Most existing application servers connect to DB2 using userid/password pairs:
  - ▶ Significant exposure if someone steals the userid/password!!!
- Trusted Context and ROLES can be used to limit exposure:
  - ▶ GRANTs to SAP\_ROLE can be restricted so that they are only valid when used by a valid SAP app server IP address.
- No change required to the code in the application server.



## Example 2: Dynamic SQL Auditing

- Better auditing controls:
  - ▶ GRANT dynamic SQL privileges to a ROLE
  - ▶ End user identity can be delegated directly to DB2 without granting dynamic SQL privileges directly to the end user
  - ▶ End user passwords can be optional.
  - ▶ No added complexity for administration of GRANTS, while retaining the ability to audit the end user's identity!





IBM Software Group – Information Management

# *Application Design*



## Universal Table Spaces

- Combination of segmented with partitioning options
  - ▶ Better space management
  - ▶ Support of mass deletes / TRUNCATE
- If partitioned
  - ▶ Still must be one table per table space
  - ▶ Can choose [Range Based partitioning](#) (as before: PBR)
  - ▶ Can choose [Partitioned By Growth](#) (PBG)
- DROP / CREATE to migrate existing page sets
- Simple table spaces cannot be created
  - ▶ Default table space is now Segmented

## Syntax

- CREATE TABLESPACE OldStylePartitioned
  - ▶ NUMPARTS x <MEMBER CLUSTER>
- CREATE TABLESPACE PartByRangeUTS
  - ▶ NUMPARTS x SEGSIZE y (key limits on CREATE TABLE)
- CREATE TABLESPACE PartByGrowthUTS
  - ▶ MAXPARTITIONS x <SEGSIZE y> (default segsize 4)
- CREATE TABLESPACE OldStyleSegmented
  - ▶ <SEGSIZE y> (default segsize 4)
- CREATE TABLE without naming table space (DB2 creates db,tsp)
  - ▶ Segmented table space (segsize 4) (CM)
  - ▶ PBG UTS (segsize 4, maxpartitions 256, dssize 4G) (NFM)

## Partition By Range (PBR) UTS

- Data is partitioned by key range; data organization and space management within partition is segmented
- The partitioning brings:
  - ▶ Partition level utilities / partition independence
  - ▶ Partition pruning (since value-based)
  - ▶ Many flavors of indexes (PI, NPI, DPSI) (since value-based)
  - ▶ Table space capacity up to 128TB
- The segmented organization brings:
  - ▶ Better space management for varying-length rows
  - ▶ Efficient mass delete / TRUNCATE (and mass delete at part level for ROTATE)

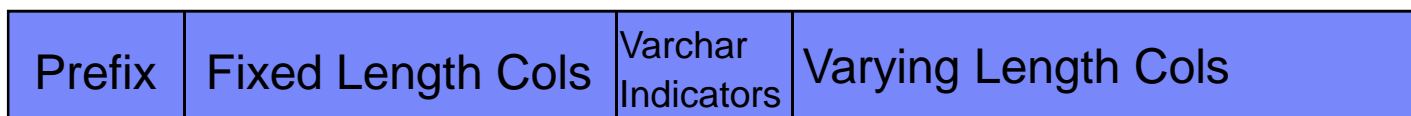


## Partition By Growth (PBG) UTS

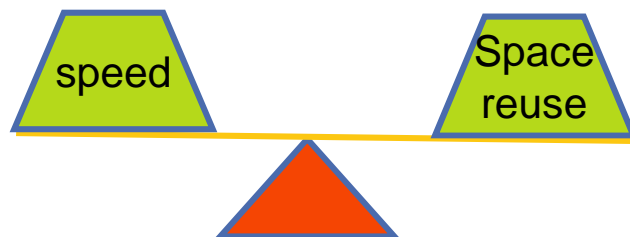
- Table space begins life as single partition; partitions are added to accommodate data growth (up to MAXNUMPARTS (alterable)). Data organization and space management with partition is segmented.
- The partitioning brings:
  - ▶ Partition level utilities (NO LOAD PART) / partition independence
  - ▶ Table space capacity up to 128TB
- The segmented organization brings:
  - ▶ Better space management for varying-length rows
  - ▶ Efficient mass delete / TRUNCATE
- Growth partitions inherit compression dictionary, if any

## Other Performance Improvements

- Reordered Row Format applicable to all table space types



- TABLE APPEND option
  - ▶ Maximizes performance for “INSERT at end”
  - ▶ Avoids overhead of attempting to preserve clustering sequence
  - ▶ CREATE or ALTER table

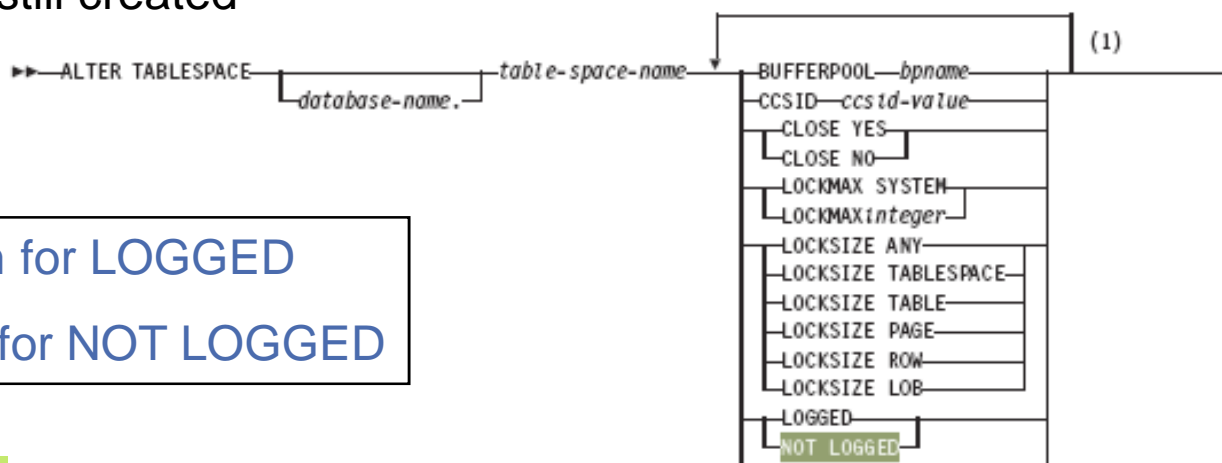


## CLONE Tables

- Allows fast replacing production data without renames and rebinds
  - ▶ An alternative to online load replace
- ALTER TABLE to create a Clone Table
  - ▶ All indexes, LOB and XML objects are also cloned
  - ▶ Structures ONLY – not data
  - ▶ Base and Clone tables share the same table space and index names
  - ▶ Underlying data sets are differentiated by a data set instance number
- On single-table table spaces (partitioned or non-partitioned)
- Use insert or load to populate clone tables
- Utilities (except RUNSTATS) can operate on clone tables with new CLONE keyword
- EXCHANGE DATA switches logical names with underlying data

## NOT LOGGED Tables

- Is actually NOT LOGGED tables, indexes, LOB, XML
- ALTER / CREATE a TABLESPACE as NOT LOGGED
  - ▶ ALTER not allowed if in same UOW with an update to the table space
- Indexes, LOB, and XML inherit the logging attribute of the base
  - ▶ This are considered “Linked” objects
- Effects the UNDO / REDO records
  - ▶ Control information is still logged
- LOB continue to log system pages & auxiliary indexes
- A Unit of Recovery (UR) is still created

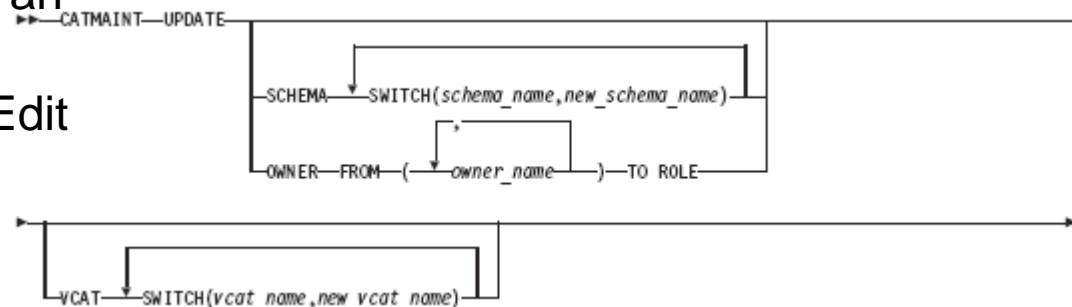


LOG YES is a synonym for LOGGED

LOG NO is a synonym for NOT LOGGED

## Schema Evolution

- ALTER TABLE ... RENAME COLUMN
  - ▶ Restricted if the Column
    - Is referenced in a View
    - Has an Index on Expression defined
    - Has a Check Constraint or Field Procedure defined
  - ▶ Restricted if the Table
    - Has a trigger
    - Is an MQT or reference by an MQT
    - Has a Valid Procedure or Edit Procedure
    - Is a Catalog Table
- RENAME INDEX
  - ▶ Existing PLAN\_TABLE contents remain unchanged
- New CATMAINT options
  - ▶ Switch schema name
  - ▶ Change from owner to role (NFM)
  - ▶ Change VCAT

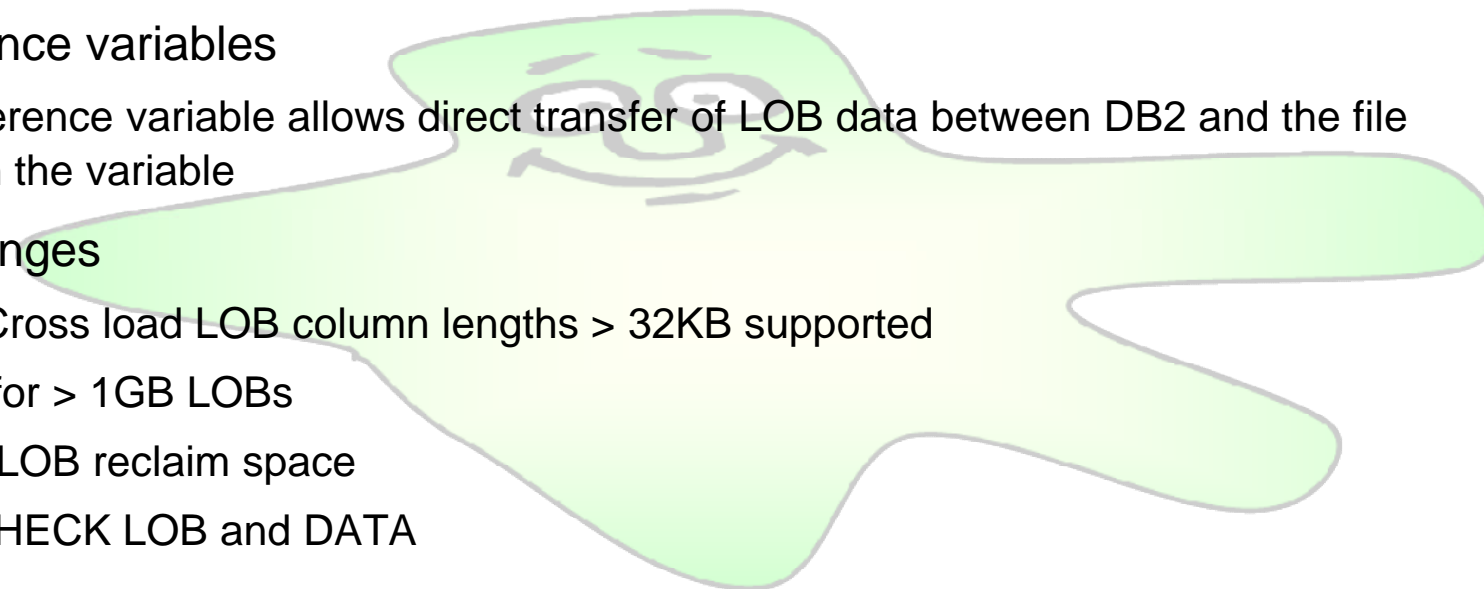


## Other Design Changes

- Optimistic Locking Support
  - ▶ Built-in timestamp for each row or page
    - Automatically updated by DB2 as a GENERATED ALWAYS column
    - Allows timestamp predicate to validate that row has not changed since last access
- Implicit Database Support
  - ▶ Automatic DB & TS
  - ▶ MAXPARTITIONS defaults to 256
  - ▶ Support for Primary Key and Unique Keys, LOB and XML objects
- STOGROUP support for SMS classes

## LOB Improvements

- Progressive Streaming for LOB Locator Values
  - ▶ DB2 uses LOB size to determine whether to send LOB data to Java or DB2 CLI clients in one go (<32KB), in chunks (<1MB) or as LOB locator (>=1MB)
    - Transparent to application using LOB locators
- FETCH CONTINUE
  - ▶ Allows applications to retrieve LOB/XML data in pieces without the use of locators
- File reference variables
  - ▶ A file reference variable allows direct transfer of LOB data between DB2 and the file named in the variable
- Utility Changes
  - ▶ LOAD / Cross load LOB column lengths > 32KB supported
  - ▶ Logging for > 1GB LOBs
  - ▶ REORG LOB reclaim space
  - ▶ Online CHECK LOB and DATA
- Elimination of LOB locks for improved availability and performance





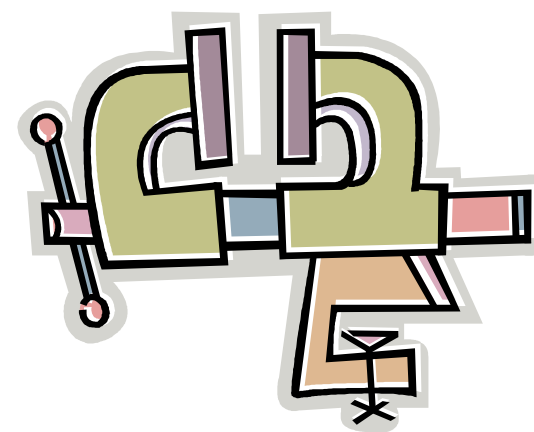
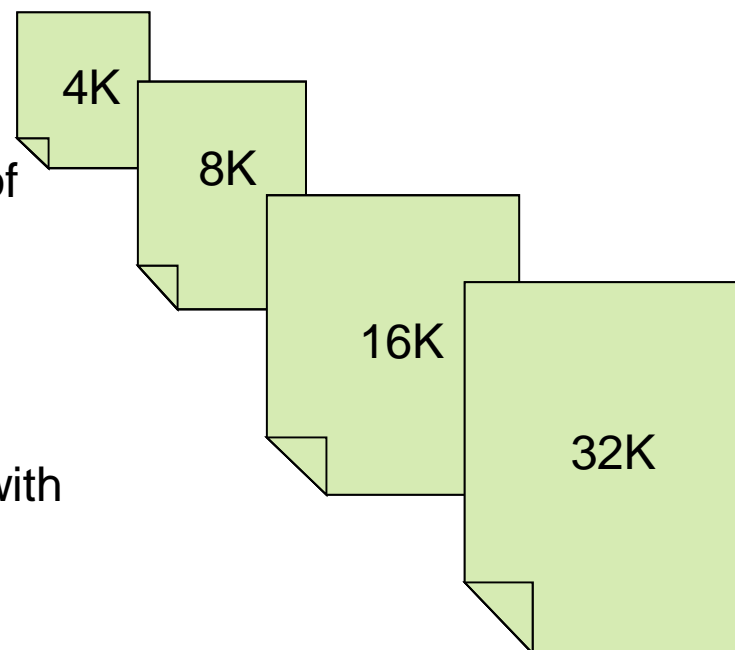
IBM Software Group – Information Management

## *Indexing Enhancements*



## Indexing Enhancements

- Larger index pages allow for more efficient use of storage
  - ▶ Fewer page splits for long keys
  - ▶ More key values per page
- Define RANDOM index keys to avoid hot spots with multiple processes inserting sequential keys
- Rebuild Index SHRLEVEL CHANGE
- Index compression provides page-level compression
  - ▶ Data is compressed to 4K pages on disk
  - ▶ 32K/16K/8K pages results in up to 8x/4x/2x disk savings
  - ▶ No compression dictionaries
    - Compression on the fly
    - No LOAD or REORG required

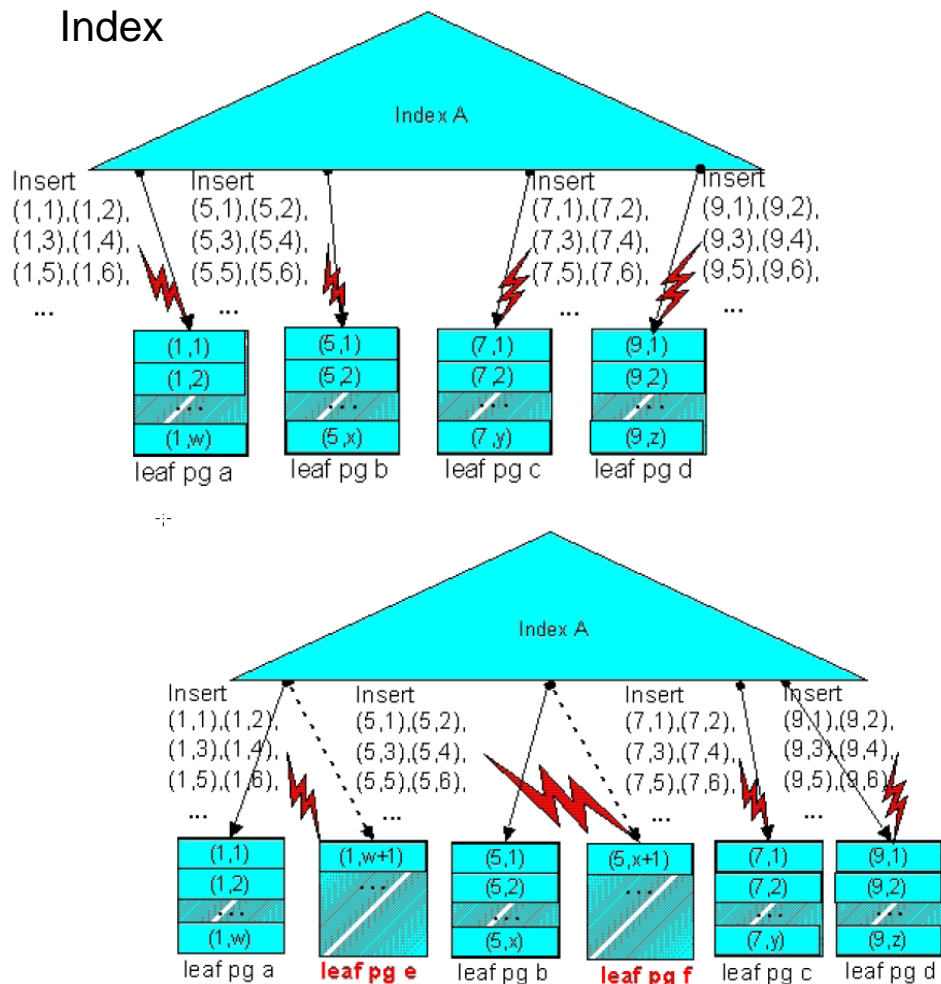


## Asymmetric Index Page Splits

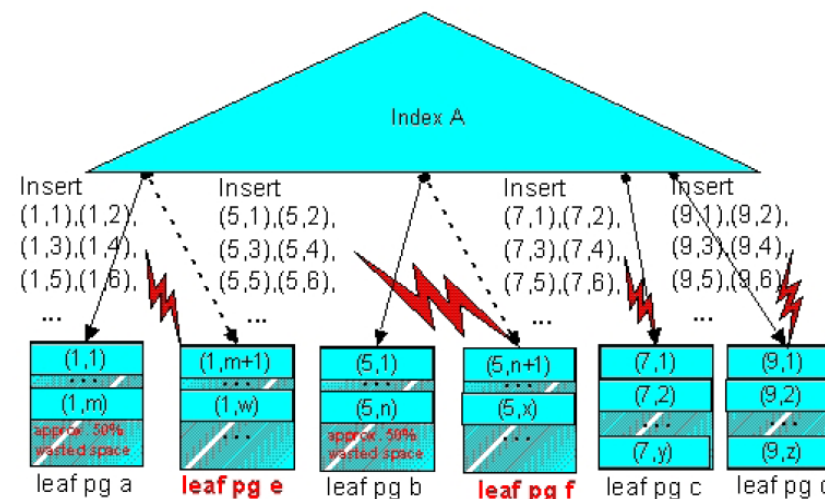
- Effectiveness of Asymmetric leaf page split function
  - ▶ Design point is to provide performance relief for classic sequential index key problem
  - ▶ Asymmetric split information is tracked in the actual pages that are inserted into, so it is effective across multiple threads across DB2 members
  - ▶ Prior to APAR PK62214, DB2 only remembered the last insert position and a counter
  - ▶ APAR PK62214 introduces changes to the tracking and detection logic, and it should work much better for data sharing
  - ▶ The new approach remembers an insert 'range' and tolerates entries being slightly out of order
  - ▶ It may still not be effective for large key sizes (hundreds of bytes), or if entries come in very bad order (i.e., they do not look sequential)
  - ▶ But for simple cases like 3, 2, 1, 6, 5, 4, 9, 8, 7, 12, 11, 10 ... DB2 will be able to determine that the inserted entries are ascending

# Asymmetric Index Page Splits

Multiple Sequential Insert Patterns on an Index



Sequential inserts into the middle of an index resulted in some pages with 50% free space prior to V9



New algorithm dynamically accommodates a varying pattern of inserts

## Index on Expression

- Simple indexes can contain concatenated columns

```
create index totalComp  
on W2_TABLE(salary,bonus)
```

- Index on expression
  - ▶ Value of the index has been transformed
  - ▶ May not be the value of any of the columns that it is derived from
  - ▶ Optimizer can use this index

```
create index totalComp  
on W2_TABLE(salary+bonus)
```

W2\_TABLE

name	salary	bonus
Gary	20,000	500
Paul	40,000	5,000
Matt	400,000	10,000
Rex	40,000	2,000
Rob	200,000	210,000

Query / Order on Total Compensation



IBM Software Group – Information Management

# ***Data Sharing Enhancements***

## Data Sharing V9 Enhancements

- Log latch contention relief
- Restart performance enhancements
  - ▶ Reduced impact of retained locks – released as rollbacks are completed
  - ▶ Open data sets ahead of log apply
- Command to remove GBP-dependency at object level
  - ▶ ACCESS DB MODE(NGBPDEP)
  - ▶ Typical usage would be before batch run
  - ▶ Command to “prime” open data set
  - ▶ ACCESS DB MODE(OPEN) [PART]
- Auto-recover GRECP/LPL objects on group restart
  - ▶ Useful in Disaster Recovery or GDPS scenarios
- DB2 overall health taken into account for WLM routing
- Balance group attach connections across multiple members on same LPAR (V7, V8 usermod)



## Utilities Highlights

- More online utilities
  - ▶ Rebuild Index SHRLEVEL CHANGE
  - ▶ Reorg LOB now supports SHRLEVEL REFERENCE (space reclamation)
  - ▶ Check data, LOB and repair locate ... SHRLEVEL CHANGE
  - ▶ Check index SHRLEVEL REFERENCE supports parallel for > 1 index
- Online REORG BUILD2 phase elimination
- REORG parallelism for UNLOAD, RELOAD, LOG phases
- Utility TEMPLATE switching



## Utilities Highlights ...

- MODIFY Recovery enhancements
  - ▶ “Retain” keyword added to improve management of copies
    - LAST, LOGLIMIT, GDGLIMIT
- Volume-based COPY/RECOVER (BACKUP SYSTEM/RESTORE SYSTEM)
  - ▶ FlashCopy technology used to capture entire content of disk volumes
  - ▶ RECOVER modified to enable object-level recovery from volume FlashCopy
  - ▶ Full integration of tape into BACKUP/RESTORE SYSTEM utilities
  - ▶ Incremental FlashCopy (APAR PK41001)
- RECOVER to any point-in-time with consistency

## Utilities Performance Improvements

- CPU reductions in LOAD (with additional savings if data is PRESORTED), REORG, and REBUILD
  - ▶ Reductions mostly due to improved index processing (\* with exceptions)
    - 10 to 20% in Image Copy\* (even with forced CHECKPAGE YES)
    - 5 to 30% in Load, Reorg, Reorg Partition, Rebuild Index
      - Except REORG TABLESPACE SHR CHG PART with NPSIs
    - 20 to 40% in Load
    - 20 to 60% in Check Index
    - 35% in Load Partition
    - 30 to 40% in Runstats Index
    - 40 to 50% in Reorg Index
    - Up to 70% in Load Replace Partition with dummy input

## Miscellaneous

- Text Search support
- Spatial support enabling OGC (Open Geospatial Consortium) compliant GIS (geospatial) applications
- DB2 9 .Net Add-Ins for Microsoft Visual Studio 2005
- RLF enhancements
- Many serviceability enhancements
  - ▶ Messages
  - ▶ Display commands
  - ▶ Refresh ERLY code without IPL
  - ▶ Many others ...
- Converged TEMP Space
- SQL
  - ▶ FETCH FIRST/ ORDER BY in Subselect
- RTS moved to catalog – automatic update
- Global Query Optimization will allow DB2 to optimize a query as a whole rather than as independent parts
  - ▶ considers the effect of one query block on another
  - ▶ considers reordering query blocks
- Serviceability
  - ▶ DISPLAY enhancements
  - ▶ Health Monitoring – internal monitor
- V9 can use more index lookaside for INSERT & DELETE
- SSL
- Improved trace filtering



## Agenda

- Quick Hits
  - ▶ Preparing for the migration
- Migration Overview
- What to Expect?
  - ▶ DB2 9 CPU performance
  - ▶ DBM1 Virtual Storage relief below the 2GB bar
- Plan Stability
- Converged TEMP Space
- Reordered Row Format
- More Quick Hits

## Quick Hits

- No need to fear migration with proper planning and testing
  - ▶ Follow step by step approach adopted by other successful customers
  - ▶ Recommended minimum maintenance level RSU0912 plus HIPERs
  - ▶ Review DB2 9 migration info APARs for list of prereq APARs and PTFs
    - Especially II14401, II14464, II14441
    - Pre-conditioning APAR for Plan Stability (PK52522) must be applied on all V8 (NFM) systems
- Minimum levels required for DB2 Connect, but should follow CST/RSU levels
  - ▶ Full support with V9.1 FP1 but strongly recommend FP8 (RSU0912)
  - ▶ Full Support with V9.5 GA but strongly recommend FP5 (RSU0912)
  - ▶ Full Support with V9.7 GA
  - ▶ Need extended support contract in place for V8.1 FP13 or V8.2 FP6+
  - ▶ V9.5 FP3 introduces Sysplex Failover/Workload balancing and XA support for DB2 Connect Client

## Quick Hits ...

- PDSE required for SDSNLOAD and SDSNLOD2
  - ▶ Make PDSE address space restartable (see Backup Slides)
- Need to configure HVSHARE (64-bit shared private storage) in IEASYSxx parmlib at a minimum of 128GB per DB2 subsystem running on LPAR
  - ▶ Even if DDF is not used
  - ▶ Use DISPLAY VIRTSTOR,HVSHARE or D VS,HVSHARE to check
- Move MSTR into service class SYSSTC in WLM policy
  - ▶ DB2 Service Monitor is built into and operates within MSTR
  - ▶ DB2 Service Monitor must be running higher than the AS it monitors
  - ▶ DB2 Service Monitor will be impacted if MSTR is pre-empted

## Quick Hits ...

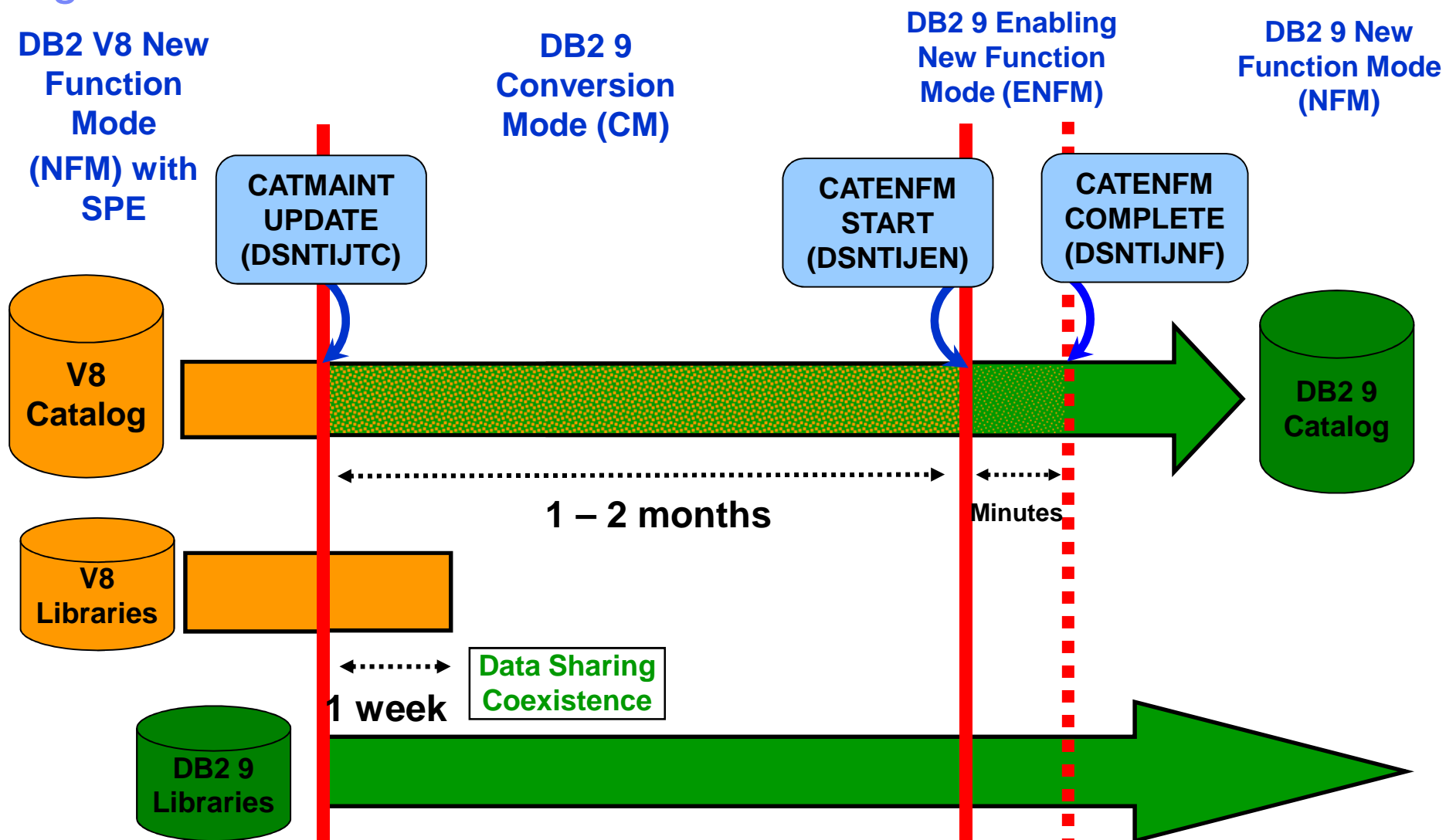
- APAR PK91610 allows “V7-like” Precompiler function
- Removed functions and incompatibilities
  - ▶ All stored procedures must be modified to become WLM enabled
  - ▶ The legacy JDBC driver will not work
    - In DB2 9, existing WLM environments configured to use the legacy driver will encounter a failure on address space initialization. This will happen in CM as well as NFM.
      - Data sharing groups with V8 in coexistence with DB2 9 (CM) will experience failure if a Java routine is invoked on any DB2 9 members where the WLM-SPAS JCL does not reference the Universal JDBC driver
    - Recommendation: Modify your WLM-SPAS JCL to use the Universal JDBC Driver while still on V8, prior to migration to DB2 9



## Quick Hits ...

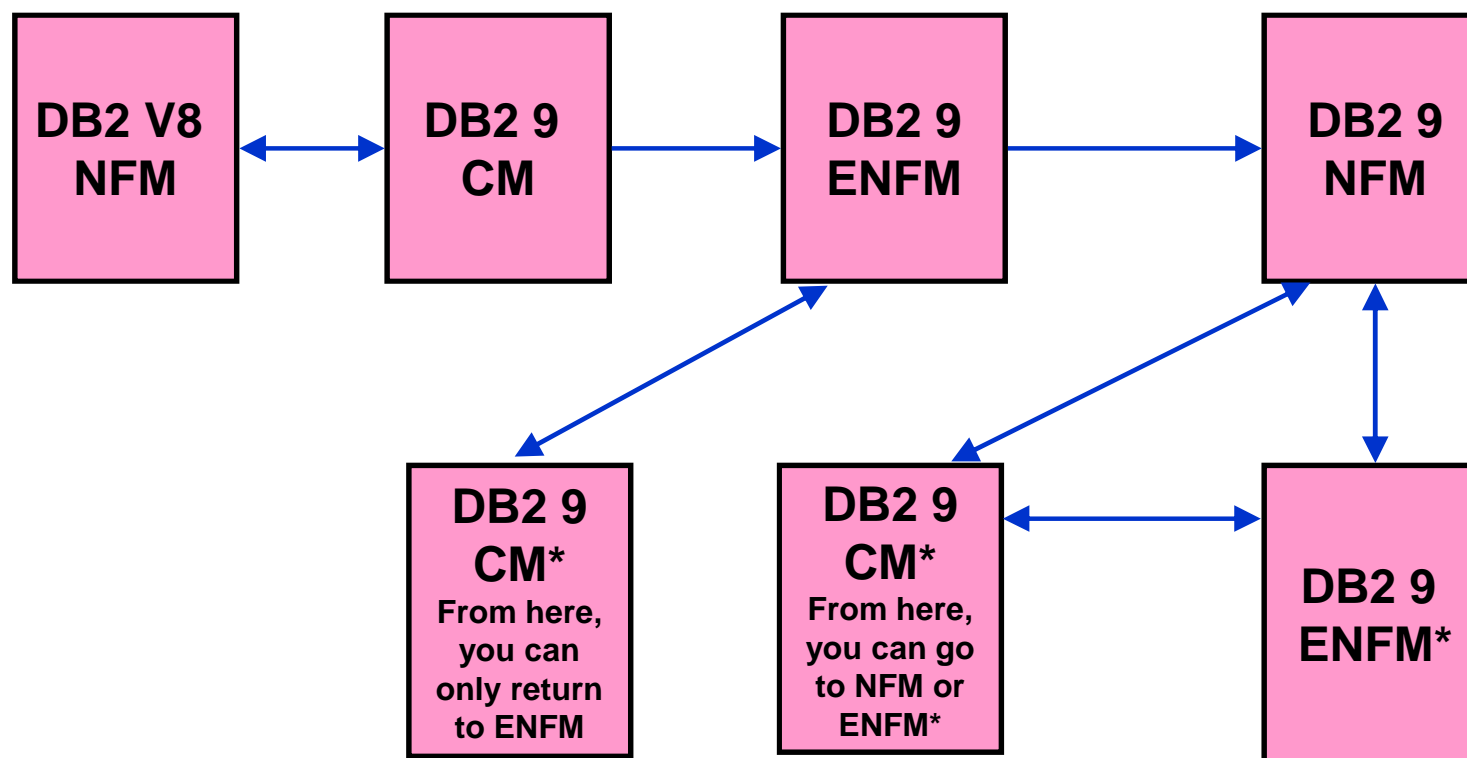
- Classic simple table space creation support removed in DB2 9
  - ▶ New defaults:
    - Implicitly created TS: Segmented (CM) or UTS - Partition by Growth (NFM)
    - Explicitly created TS: Segmented
  - ▶ Aim at converting existing classic simple table spaces to segmented, or UTS PBG, or single part partitioned with EA
- Changes to defaults
  - ▶ Default setting for MGEXTSZ (secondary extent allocation) is YES
    - Changed by the installation CLIST from NO to YES
  - ▶ BIND options
    - ISOLATION defaults to CS, CURRENTDATA defaults to NO
    - Not changed for REBIND i.e., use the existing value

# Migration Overview



## Migration and Fallback Paths

- With DB2 9, you can always drop back to the previous stage
- Cannot fallback to V8 after entry to DB2 9 (ENFM), but can fallback to DB2 9 (CM\*)



## DB2 9 CPU Performance

- The target for DB2 9 CPU performance is to be roughly equivalent or marginally better relative to V8
  - ▶ Assumes z890, z990, z9 or z10
  - ▶ Assumes no access path regression
  - ▶ Assumes REBIND on the new release
- Mileage in terms of reduced CPU reduction will vary
- Do not spend the benefits until you see them
- Customers running DB2 9 on old hardware (z800/z900) will definitely see CPU regression - may be 10-15% increase
- Data sharing customers may see significant savings from reduced LC19 contention and less spin to get unique LRSN
  - ▶ Spinning outside of the latch in DB2 9 (CM)
  - ▶ Allowing duplicate LRSNs for different pages in DB2 9 (NFM)

## DBM1 Virtual Storage below 2GB ...

- DBM1 Virtual Storage Constraint Relief for static SQL users
  - ▶ EDM pool in DB2 9 – need REBIND
    - SKCT/SKPT moved above 2GB
    - A portion (close to 30%) of CT/PT moved above 2GB
    - \*\* BUT \*\*
      - There are no stealable EDM components left below 2GB
      - Increase in package size after REBIND
      - Increased likelihood of hitting EDM Pool full condition
  - ▶ Recommendation for EDM Pool below 2GB
    - On V8, identify the max number of pages held by CT/PT at peak
    - Plan on keeping at least  $2 * 70% * \text{CT/PT (V8 peak)}$ 
      - Intent of 2x factor is to provide some headroom
- DBM1 Virtual Storage Constraint Relief for dynamic SQL users
  - ▶ Local dynamic statement cache (KEEPDYNAMIC=YES)
    - Rough estimation of V9 = 50% of V8

## DBM1 Virtual Storage below 2GB ...

- User thread storage, System thread storage, Stack storage
  - ▶ Current expectation of less than 10% difference overall
- Potential reduction typically ranges from 0 to 15% depending on thread/stack storage usage
  - ▶ Very few installations will see as much as a 200-300MB reduction!
    - Especially if you are an IMS/TM customer who has very large ECSA and a small Extended Private Region (i.e., 1GB)
    - No VSCR for dynamic SQL users unless they are using local dynamic statement cache as a result of using BIND option KEEP DYNAMIC(YES)
    - VSCR from local dynamic statement cache savings will be small if the V8 size is heavily constrained by low MAXKEEPD

## DBM1 Virtual Storage below 2GB ...

- Introduced with V8 APAR PK20800 8/07
  - ▶ -DISPLAY THREAD(\*) SERVICE(STORAGE)
  - ▶ Produces DSNV492I message that can be used by DB2 service for diagnostics

```
V91A  N *  0 003.RCRSC 02 SYSOPR      0067  0
V490-SUSPENDED 07213-09:59:18.02 DSNRCRSC +00000230 01.51
V492-LONG 252 K VLONG 40 K 64 1028 K
```
  - ▶ Includes Agent Local Non-System Storage usage
  - ▶ Does not include Getmained Stack Storage usage
  - ▶ The key values are the CT Long storage pool and the CT Very Long storage pool values. These reflects virtual storage consumption below the 2GB bar. They may be used to identify poorly behaved applications or DB2 code issues.

## Quick Hits ...

- Need to rebind all plans and packages that have not been rebound since DB2 V3
- Removed functions and incompatibilities
  - ▶ All stored procedures must be modified to become WLM enabled
  - ▶ The legacy JDBC driver will not work
    - In DB2 9, existing WLM environments configured to use the legacy driver will encounter a failure on address space initialization. This will happen in CM as well as NFM.
      - Data sharing groups with V8 in coexistence with DB2 9 (CM) will experience failure if a Java routine is invoked on any DB2 9 members where the WLM-SPAS JCL does not reference the Universal JDBC driver
    - Recommendation: Modify your WLM-SPAS JCL to use the Universal JDBC Driver while still on V8, prior to migration to DB2 9



## Why REBIND Plans and Packages?

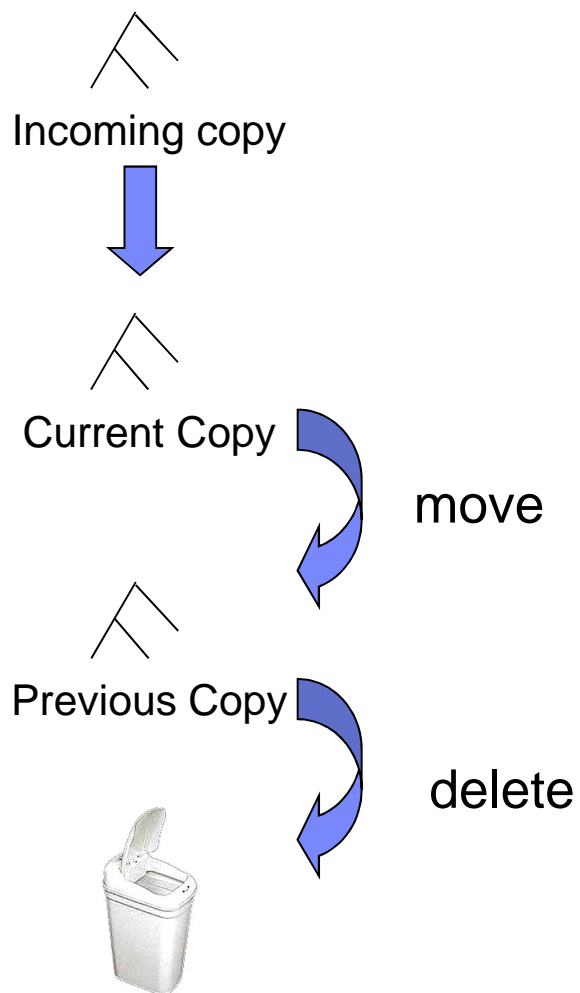
- As a general rule it is always a good recommendation to rebind on the new release to get the best performance
- Almost all Optimizer enhancements available in DB2 9 (CM)
- Avoid special DB2 handling to compensate for the different package structure in DB2 9 i.e., to avoid “puffing” of CTs/PTs/SLTEs
- Re-enable SPROCs
  - ▶ Many plans/packages have SPROCs for fast column processing
  - ▶ All plans/packages with SPROCs bound prior to DB2 9 will be disabled
  - ▶ As a result DB2 will build SPROCs dynamically at execution time
    - Typical CPU performance impact in 0 to 10% range
    - Non-zero value for BYPASS COL indicator of problem
    - Performance trace of IFCID 224 identifies plans and packages which need rebinding to re-enable SPROCs
- Exploit DBM1 31-bit virtual storage constraint relief for static SQL

## Plan Stability ...

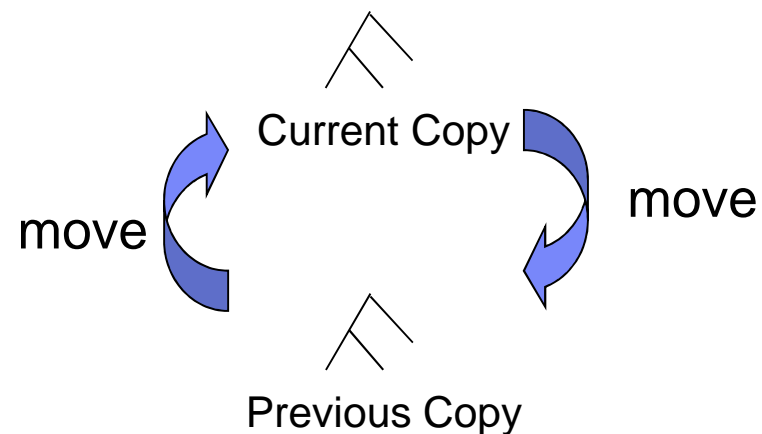
- At REBIND, save old copies of packages
  - ▶ Catalog tables
  - ▶ Directory
- Two flavours
  - ▶ BASIC and EXTENDED
  - ▶ Controlled by new ZPARM PLANMGMT
  - ▶ Default is OFF
  - ▶ Also supported as REBIND options
- REBIND PACKAGE ...
  - ▶ PLANMGMT(BASIC)
    - 2 copies: Current and Previous
  - ▶ PLANMGMT(EXTENDED)
    - 3 copies: Current, Previous, Original
- Most bind options can be changed at REBIND
  - ▶ But a few must be the same ...
  - ▶ Otherwise REBIND fails with DSNT216I
- REBIND PACKAGE ...
  - ▶ SWITCH(PREVIOUS)
    - Switch between current & previous
  - ▶ SWITCH(ORIGINAL)
    - Switch between current & original
- FREE PACKAGE ...
  - ▶ PLANMGMTSCOPE(ALL) – Free package completely
  - ▶ PLANMGMTSCOPE(INACTIVE) – Free all old copies
- Catalog support
  - ▶ SYSPACKAGE reflects active copy
  - ▶ SYSPACKDEP reflects dependencies of all copies
  - ▶ Other catalogs (SYSPKSYSTEM, ...) reflect metadata for all copies
- Invalidation and Auto Bind
  - ▶ Each copy invalidated separately
- Auto bind replaces only the current copy – previous and original are not affected

## Plan Stability ...

### REBIND ... PLANMGMT(BASIC)



### REBIND ... SWITCH(PREVIOUS)

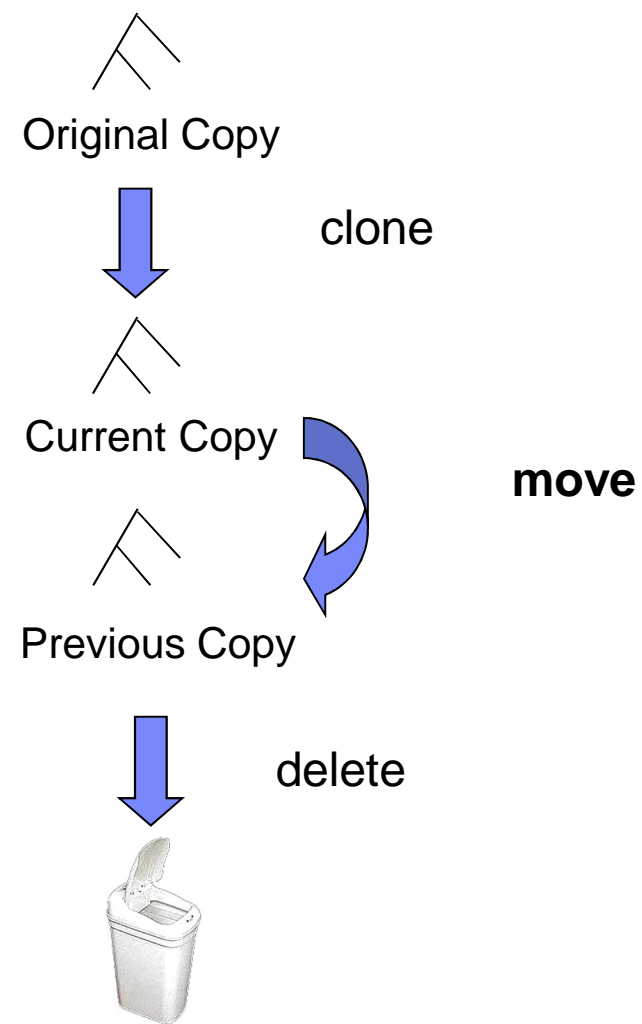
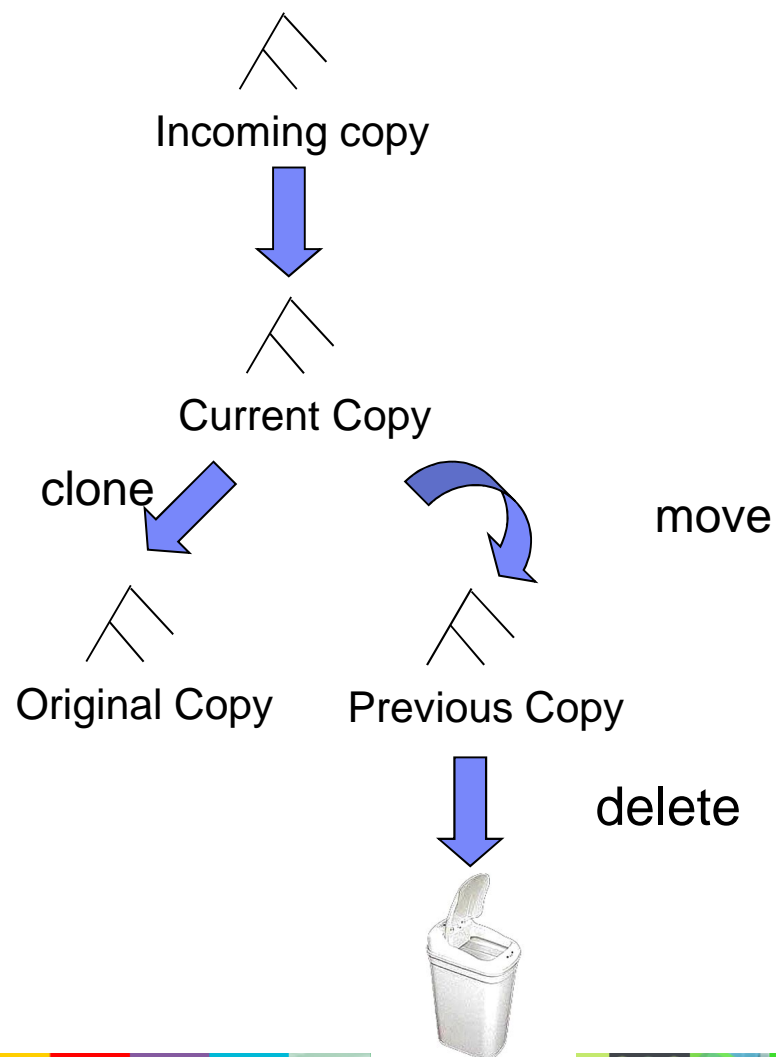


Similar processing for:

- EDM Sections
- SYSPACKDEP dependencies
- SYSPACKAGE record saved in EDM

# Plan Stability ...

## REBIND ... PLANMGMT(EXTENDED) REBIND ... SWITCH(ORIGINAL)



## Migration Strategy using Plan Stability

- Before migrating to DB2 9 (CM)
  - ▶ Ensure V8 plan table information is available
- On migration to DB2 9 (CM)
  - ▶ Set zparm PLANMGNT to EXTENDED
  - ▶ Objective:
    - Make sure that the V8 version of package is kept as the original in case a fallback to DB2 V8 is required
  - ▶ Restriction:
    - EXTENDED means that DB2 always keeps 3 versions of the package (even if they are the same)
    - Watch out for SPT01 space growth as limit is still 64GB with DB2 9
      - Apply APAR PK80375 to turn on ESA Compression for SPT01
      - REORG SPT01 to reduce number of pieces

## Migration Strategy using Plan Stability ...

- REBIND in DB2 9 (CM)
  - ▶ DB2 now stores 3 versions of the package
  - ▶ Initial REBIND
    - Current version = DB2 9 version
    - Previous version = Original version = V8 version
    - If needed because of V9 access path regression, use REBIND PACKAGE ... SWITCH(PREVIOUS) to fallback to the V8 version
  - ▶ Subsequent REBINDs
    - Current version = “New” DB2 9 version
    - Previous version = Latest DB2 9 version
    - Original version = V8 version
    - Use REBIND PACKAGE ... SWITCH(PREVIOUS) to fallback to the previous V9 version

## Migration Strategy using Plan Stability ...

- REBIND PACKAGE...SWITCH(PREVIOUS | ORIGINAL)
  - ▶ SWITCH(PREVIOUS): toggles previous and current copies
  - ▶ SWITCH(ORIGINAL): previous deleted; current->previous; original cloned to current
- FREE PACKAGE...PLANMGMTSCOPE(ALL | INACTIVE)
  - ▶ ALL is the default and frees all copies
  - ▶ INACTIVE frees all old copies
- SYSPACKAGE reflects the current copy
  - ▶ Other package related tables reflect dependencies of all packages
- DTYPE column of SYSPACKDEP overloaded to indicate 'P'previous or 'O'riginal
- To keep one V8 package
  - ▶ REBIND with PLANMGMT(BASIC) once
  - ▶ Subsequent REBINDs with PLANMGMT(OFF)
  - ▶ REBIND SWITCH(PREVIOUS) can be used to use the original V8 package

## Migration Strategy using Optimizer Enhancements

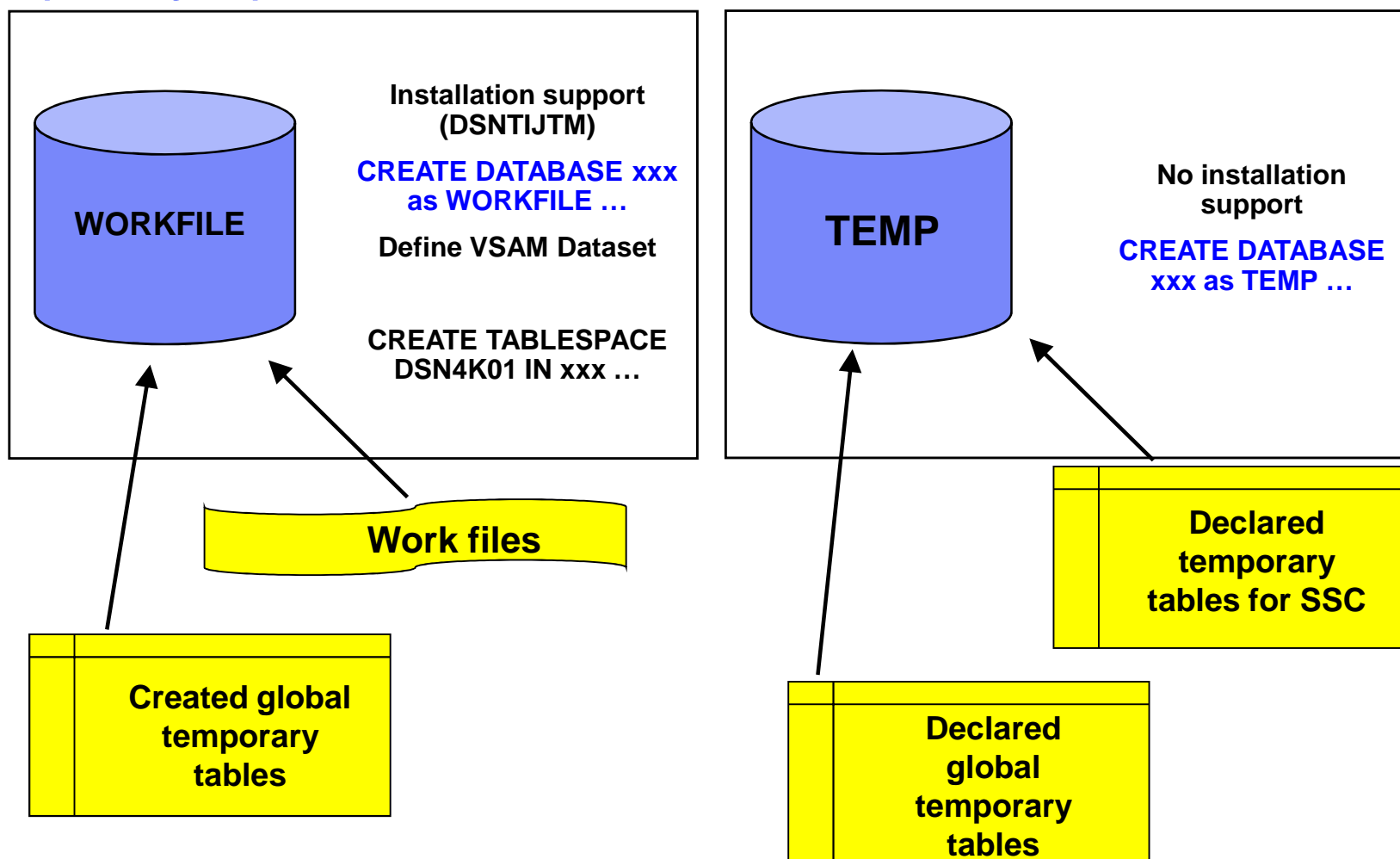
- Recommended Optimizer ZPARM
  - ▶ Set OPTIOWGT=ENABLE (APAR PK75643 reflects this recommendation)
  - ▶ Set OPTIXOPREF=ON (APAR PK77426 reflects this recommendation)
  - ▶ No need to disable global query optimisation now
    - Set OPTXQB=ON which is the default
- Delay REBIND on DB2 9 (CM) until running stable
  - ▶ Do not rebind on DB2 9 (CM) until RUNSTATS has been run
  - ▶ Execute RUNSTATS with TABLE(ALL) INDEX(ALL) KEYCARD
  - ▶ Make sure new ZPARM STATCLUS=ENHANCED (default)
    - Introduces major change to CLUSTERRATIO calculation and introduction of new statistic DATAPEATFACTOR
  - ▶ Use DB2 supplied RUNSTATS utility if ISV utility does not yet support the enhanced statistics collection



## Migration Strategy using Plan Stability ...

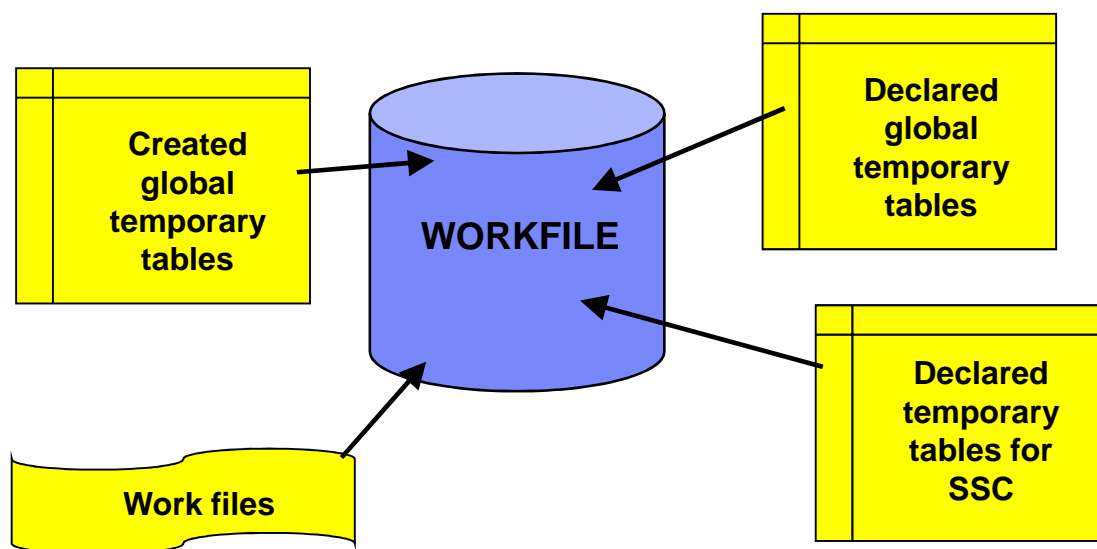
- In case of fallback to DB2 V8, before falling back to V8
  - ▶ Use REBIND PACKAGE ... SWITCH(ORIGINAL) to fall back to original version of package (V8 version)
- In the future, to establish a new original version e.g., to move forward to the version after DB2 9
  - ▶ Use FREE PACKAGE ... PLANMGMTSCOPE(INACTIVE) to free off the original (will also free off the previous version)

## Temporary Space - The DB2 V8 Picture



\* Only in a data sharing environment – in non-data sharing syntax is **CREATE DATABASE DSND07**

## Temporary Space - The DB2 9 Picture



Installation and migration support  
(REXX program called by DSNTIJTM)

```
CREATE DATABASE xxx as WORKFILE;
```

```
DSNTWFG DB41 DB2ADM xxx +
      3 10 16 BP0 SYSDEFLT +
      1 20 16 BP32K SYSDEFLT
```

- Declared Global Temporary Tables and Static Scrollable Cursors now use the WORKFILE database instead of the TEMP database
- Uses DB2-managed (instead of user-managed) storage in SYSDEFLT
- Segmented table space organisation (user-defined SEGSIZE or default of 16)
- 4KB and 32KB page sizes only – no 8KB or 16KB

## Planning For Converged TEMP Space

- Migration from DB2 V8
  - ▶ To reclaim TEMP database storage, \*YOU\* must drop the TEMP database and reallocate the storage
  - ▶ Recommendation: Do not drop the TEMP database until you are sure that you will not return to V8, to avoid having to recreate it after fallback
- New installation panel for work file database definitions (DSNTIP9)
  - ▶ In migration mode, if you specify non-zero values
    - Migration job DSNTIJTM will create additional DB2-managed WORKFILE table spaces in the SYSDEFLT storage group → new REXX program DSNTWFG
    - DB2 does not take into account the existing work file table spaces
- Recommendation: set the 'DSVCI' ZPARM to YES to allow DB2 to match VSAM CI size to table space page size

## Planning For Converged TEMP Space ...

- Heavier use of 32K workfiles to help performance (CM)
  - ▶ If the workfile record length is less than 100 bytes, then a 4KB page tablespace is used
  - ▶ In all other cases, DB2 tries to use a 32KB page tablespace
    - If no 32KB page TS is available, DB2 uses a 4KB page TS – lost performance benefit
  - ▶ Recommendations
    - Assign bigger 32KB page workfile buffer pool
    - Allocate bigger/more 32KB page workfile tablespaces

## Planning For Converged TEMP Space ...

- IFCID 2 updated with new counters
  - ▶ Monitoring of temp space usage at the DB2 subsystem level
    - Number of times MAXTEMPS has been exceeded
    - Number of times a 4KB page tablespace was used when a 32K page tablespace was preferable (but not available)

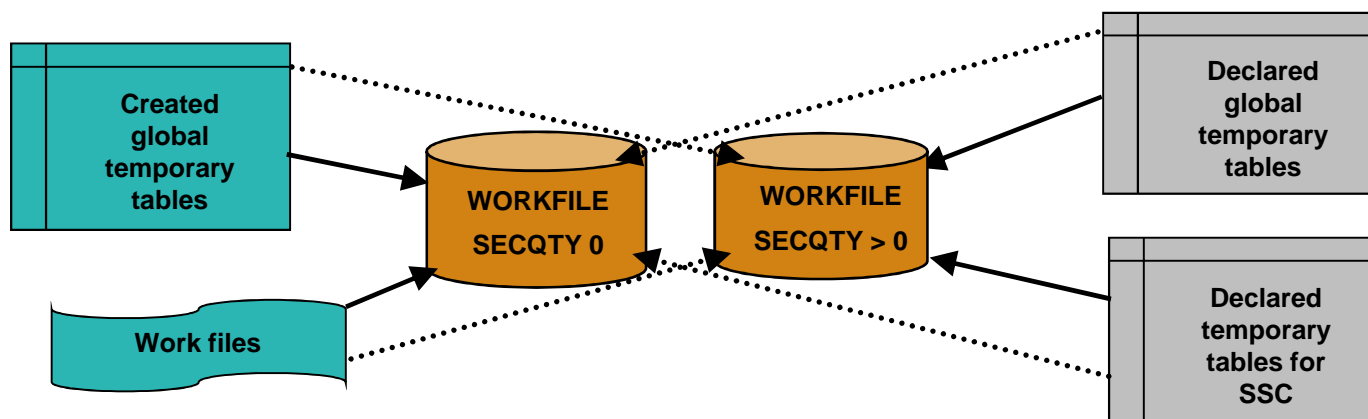
WORK FILE DATABASE	QUANTITY
MAX TOTAL STORAGE (KB)	5120
CURRENT TOTAL STORAGE (KB)	1600
STORAGE IN 4K TS (KB)	64
STORAGE IN 32K TS (KB)	1536
4K USED INSTEAD OF 32K TS	0
32K USED INSTEAD OF 4K TS	0
AGENT MAX STORAGE (KB)	0
NUMBER OF MAX EXCEEDED	0

Support in Omegamon XE for DB2 PM/PE V4.20

- New IFCID 343 (Performance Trace Class 3)
  - ▶ Written when the MAXTEMPS limit for an agent is exceeded

## Planning For Converged TEMP Space ...

- APAR PK70060 introduces a soft barrier between DGTT and non-DGTT space use in workfile database



- Recommendation: Define some tablespaces with zero secondary quantity and some with non-zero secondary quantity
  - ▶ For the DB2-managed workfile tablespaces with zero secondary quantity, make sure the primary quantity is a little bit less than 2GB to avoid DB2 allocating a new subsequent piece A002
- APAR PM02528 provides the choice to make it a hard barrier
  - ▶ New opaque zparm WFDBSEP=NO|YES

## Planning For Converged TEMP Space ...

- Declared Global Temporary Tables (DGTTs)
  - ▶ Up to 20x increase possible in both CPU resource consumption and elapsed time without fixing maintenance
    - APAR PK67301 to reduce excessive logging for sequences of insert and mass delete within the same commit scope
      - With more frequent commits, the performance degradation is less and PK67301 makes it even less
    - APAR PK62009 to reclaim the space emptied by mass delete after commit for ON COMMIT DELETE DGTT
      - However, if you have a loop of Inserts/Mass Delete without commit, then Insert performance will steadily deteriorate and that is where DELETE WHERE can help
    - APAR PK70347 to reduce DBM1 31-bit storage usage when Performance Trace Class 8 active



## Reordered Row Format (RRF)

- Potential for significant reduction in CPU resource consumption when accessing many rows with many varying length columns
  - ▶ Implements CPU tuning recommendation to place fixed length columns ahead of varying length columns
  - ▶ Provides for direct access to each varying length column
  - ▶ In some few cases may lead to increased logging volume
- Application transparent – even for SELECT \*
- On by default in DB2 9 (NFM) and applies to all tablespace types
- A pageset will be automatically converted from Basic Row Format (BRF) to RRF when pageset is re-allocated
  - ▶ REORG and LOAD REPLACE utilities override KEEPDICTIONARY during first time migration where data compression of variable-length rows

## Reordered Row Format (RRF)

- Automatic repositioning of Variable columns to end of row
  - ▶ Length attributes replaced with Indicators positioned after fixed length columns
- Any table space created in DB2 9 NFM
- To Convert:
  - ▶ REORG or LOAD REPLACE a table space or partition
  - ▶ ADD PARTITION
- EDITPROCs or VALIDPROCs **inhibit conversion**
  - EDITPROCs may need to be updated if implemented for specific columns
  - Byte RFMTTYPE passed to indicate fixed length, basic, or reordered format
  - Consider this impact on tables encrypted via an EDITPROC
- PIT RECOVER will set the table space to the row format of the PIT
- Catalog / Directory remains in Basic Row Format (BRF)



## Reordered Row Format (RRF) ...

- Be careful when using DSN1COPY to export/import data during the transition period when all DB2 systems and data sharing groups have not yet been migrated to DB2 9 (NFM)
  - ▶ Should consider disabling RRF on a temporary basis before migrating first system or data sharing group to DB2 9 (NFM), and re-enabling after all the systems and data sharing groups have been migrated to DB2 9 (NFM)

## Reordered Row Format (RRF) ...

- APAR PK87348 Enable BRF for UTS PBR/PBG
  - ▶ Hidden zparm SPRMRRF now converted to opaque zparm
    - SPRMRRF=ENABLE|DISABLE (online changeable)
  - ▶ SPRMRRF controls row format for new tablespaces
  - ▶ REORG and LOAD REPLACE will honour SPRMRRF setting
    - If SPRMRRF=ENABLE
      - Get BRF -> RRF conversion
      - COMPRESS YES|NO is no longer a factor
      - New PART via ALTER TABLE ADD PART or UTS PBG growth will be RRF
    - If SPRMRRF=DISABLE
      - No BRF -> RRF conversion
      - If already RRF then stay as RRF
      - New PART will be BRF

## Reordered Row Format (RRF) ...

- APAR PK85881 LOAD/REORG ROWFORMAT support
  - ▶ New keyword ROWFORMAT for LOAD REPLACE and REORG
  - ▶ Determines output row format
  - ▶ ROWFORMAT BRF | RRF
- Some restrictions...
  - ▶ SPRMRRF setting is ignored for
    - Existing tablespace with EDITPROC i.e., stays BRF
    - XML tablespace always RRF
      - New PART of existing XML tablespace will always be RRF
  - ▶ SPRMRRF and LOAD/REORG ROWFORMAT not honoured for CLONE tables relationship
    - UTS, LOB or XML
    - No conversion from BRF -> RRF or RRF -> BRF

## More Quick Hits ...

- Convert Private Protocol to DRDA
  - ▶ DB2 10 will no longer support ANY use of Private Protocol
    - Need to convert to DRDA prior to V10
    - Enhancements to V8 and V9 DRDA support to aid conversion
      - Tools to identify plans and packages requiring conversion (PK64045)
        - ◇ DSN1PPTP analyses IFCID157 trace records for input into DSNTP2DP
        - ◇ DSNTP2DP reports on applications using private protocol
        - ◇ DSNTPPCK scans SYSSTMT and SYSPACKSTMT for invalid syntax
      - Ability to enable or disable DRDA alias resolution processing (PK64045)
        - ◇ New zparm DRDA\_RESOLVE\_ALIAS
      - Ability to enable or disable Private protocol (PK92339)
        - ◇ New zparm PRIVATE\_PROTOCOL
        - ◇ Enables testing to ensure private is eliminated

## More Quick Hits

- Random Group Attach

- ▶ Different Group Attach behavior when running multiple DB2 members of the same data sharing group on the same LPAR

- V8: DB2 attaches to first available member based on IEFSSNxx parmlib
  - Usermod available on V8 to change the behaviour and spread connections across available DB2 members on the same LPAR
- V9: DB2 spreads connections across available members on the LPAR
- Co-requisite APARs PK79228 and PK79327 provide an option to re-instate the old V8 behaviour
  - New zparm RANDOMATT=YES|NO (online changeable)
  - If RANDOMATT=NO
    - ◇ DB2 member is considered "ineligible" for Random Group Attach
    - ◇ DB2 member will only be selected if no other members are available
  - To completely restore V8 behaviour, must specify RANDOMATT=NO for every member of the data sharing group

## More Quick Hits ...

- Convert Plans with DBRM to Packages
  - ▶ Plans bound directly with DBRMs are not supported in DB2 10
  - ▶ APARs PK79925 (V8) and PK62876 (V9)
    - New REBIND PLAN option COLLID('collection\_id'|\*)
    - Conversion from plans with DBRMs into plans with a package list
      - Read SYSIBM.SYSDBRM to get all DBRM records bound with the plan
      - Bind them into packages with the specified collection ID or use the default collection id (DSN\_DEFAULT\_COLLID\_<planname>)
      - Bind the package list into the plans
      - Delete records in SYSIBM.SYSDBRM and SYSIBM.SYSSTMT tables



## More Quick Hits ...

- Larger index page size (DB2 9 NFM)
  - ▶ Can be used with or without index compression
  - ▶ Index must be defined in 8 KB, 16 KB, or 32 KB bufferpool
  - ▶ Potential to reduce the number of index leaf page splits, which are painful especially for GBP-dependent index (data sharing)
    - Reduce index tree latching contention
    - Reduce index tree p-lock contention
    - Reduce active log write I/O
  - ▶ Potential to reduce the number of index levels
    - Reduce the number of getpages for index traversal
    - Reduce CPU resource consumption
  - ▶ Possibility that large index page size may aggravate index bufferpool hit ratio for random access

## More Quick Hits ...

- Stored Procedures - Performance of different languages
  - ▶ Environment Configuration
    - z/OS 1.9
    - DB2 9 for z/OS
    - Universal Driver 3.52.76
    - JDK 1.4.2 (SQLJ/JDBC stored procedures)
    - 3 CP's
    - 2 zIIP's
    - 2 zAAP's
    - IRWW OLTP workload

## More Quick Hits ...

- Stored Procedures - Performance of different languages ...

Language/API	Base CPU/tran Cost	Billable CPU/tran Cost after zIIP and/or zAAP redirect
COBOL stored proc	1X (BASE)	.88x
C stored proc	.95x	.83x
SQLJ stored proc	1.7x	1.15x (zIIP + zAAP)
JDBC stored proc	2.95x	1.76x (zIIP + zAAP)
External SQL stored proc	1.62x	1.49x
Native SQL stored proc	1.14x	.65x
Remote SQLJ	1.78x	1.06x

## More Quick Hits ...

- Identify and remove unused indexes
  - ▶ Difficult to determine in a dynamic SQL environment
  - ▶ SYSINDEXSPACESTATS.LASTUSED (RTS) records last used date
  - ▶ Improved insert and delete performance

## More Quick Hits ...

- Expand BSDS using V8 stand-alone utility DSNJCNVB
  - ▶ Or let DSNTIJUZ run the DSNJCNVB module during migration to DB2 9 (CM)
- Removed functions and incompatibilities
  - ▶ The DBPROTCL ZPARM is removed
    - The default distributed protocol for BIND will always be DRDA
  - ▶ The RELCURHL=NO ZPARM option is removed
    - Incompatibility for applications dependent on retaining page or row locks across commits for a WITH HOLD cursor

## More Quick Hits ...

- Dynamic prefetch replaces all sequential prefetch in SQL calls (CM), except in tablespace scan
  - ▶ No rebind is required to switch to dynamic prefetch as this change is transparent to Optimizer
  - ▶ In V8 when pages are read in via dynamic prefetch they are marked as sequential and when they are subsequently touched by random getpage they are reclassified as random
  - ▶ In DB2 9 the pages are not re-classified following subsequent touch by random getpage

## More Quick Hits ...

- Object level RECOVER from system level backups is available as soon as DB2 9 (CM), whereas object level RECOVER to PIT with consistency is not available until DB2 9 (NFM)
- Nice package of DB2 9 COPY enhancements
  - ▶ MRU to avoid throwing away useful random pages
  - ▶ Automatic implicit CHECKPAGE on table space pages and no setting of COPYPEND when damage found
  - ▶ All with less CPU

## More Quick Hits ...

- OPTIOWGT & SYSDUMMY1

- ▶ Execute RUNSTATS against DSNDB06.SYSEBCDC, so that Optimiser knows it is a 1-row table
- ▶ This will avoid CPU query parallelism being chosen on a SYSIBM.SYSDUMMY1 query after OPTIOWGT is enabled

- GROUP BY in different order

- ▶ Potential problem area is SQL query without an ORDER BY expecting order to be preserved from a GROUP BY
- ▶ Relational theory states that order is ONLY guaranteed with an ORDER BY
- ▶ DB2 9 can reorder the GROUP BY columns to match an index and thus allow a sort to be avoided
- ▶ Applications that have relied on the order from GROUP BY will be exposed



## More Quick Hits ...

- General sort avoidance
  - ▶ As DB2 provides enhanced sort avoidance, applications written depending on cursor materialisation will be exposed
  - ▶ For example - Application FETCHes a row, and then issues a stand-alone UPDATE which moves the row forward, then it is possible that the row will be seen a 2nd time as the FETCH processes the rows
- Universal Table Space (UTS)
  - ▶ LOCKSIZE ROW (default) is not a recommendation
  - ▶ SEGSIZE 4 (default) is not a recommendation
  - ▶ No MEMBER CLUSTER support
  - ▶ Hard prerequisite for CLONE TABLE
  - ▶ Migration path: UNLOAD/DROP/CREATE/LOAD

## More Quick Hits ...

- SQL TRUNCATE
  - ▶ Efficient mass delete of data rows only with classic segmented or universal tablespace
  - ▶ No efficient mass delete of index entries
  - ▶ Performance is ugly with classic partitioned tablespace and/or many indexes
- REOPT(AUTO) only applies to dynamic SQL
- Optimistic Locking is not just for WebSphere!
- Real Time Statistics (RTS) always enabled
  - ▶ In DB2 9 the RTS tables are now in the DB2 Catalog
  - ▶ In-memory statistics are always externalised to the RTS tables
  - ▶ Whereas in V8 the process had to be explicitly started

## More Quick Hits ...

- Object level point-in-time RECOVER from System Level Backup (SLB) will fail if dataset has moved away to a different volume since the backup was taken
  - ▶ For example
    - PIT recovery to a point before REORG with inline copy
    - Storage administrator decides to move datasets around to defrag volumes
    - Migrate to new DASD
  - ▶ Consider use of Recovery Expert Tool, or ISV tool which provides similar function



IBM Software Group – Information Management

# DB2 9 for z/OS

## Performance Highlights



*Section 4*

## Agenda

- Performance Scalability Enhancements
- Query Performance Enhancement
- Utility CPU time reduction
- Other performance topics
  - ▶ zIIP and zAAP offload
  - ▶ Native SQL procedure
  - ▶ Index compression
  - ▶ Varchar
  - ▶ LOBs

## Insert/Update/Delete Performance Scalability

- More index lookaside (CM)
- Log-related enhancements (NFM)
- Table space Append option (NFM)
- Randomized index key (NFM)
- RTS index last used (NFM)
- Index page split reduction (NFM)
- ➔ Up to 3x reduction in CPU, Getpage, index page P-lock and up to 25x reduction in index page P-lock suspension/negotiation observed

## Insert/Update/Delete Performance Scalability ...

- Index lookaside for additional indexes (CM)
  - ▶ In V9, possible for more indexes in both Insert and Delete
    - More than 100 times reduction in Getpage possible for applicable IX
- LC19 Log latch contention relief in data sharing (NFM)
  - ▶ No Log table space option (NFM) where appropriate to reduce LC19 contention also
- Archive log to use BSAM, enabling (NFM)
  - ▶ I/O striping
  - ▶ Compression if Extended Format data set

## Insert/Update/Delete Performance Scalability ...

- Tablespace Append option (NFM)
  - ▶ CREATE/ALTER TABLE ... APPEND YES
  - ▶ To reduce longer chain of spacemap page search as tablespace keeps getting bigger
- Randomized index key to avoid hot spots (NFM)
  - ▶ Can be beneficial for data sharing because of IX page P-lock contention
  - ▶ Trade-off between contention relief and additional Getpage, read/write I/O, and lock request
    - Better for indexes resident in buffer pool



## Insert/Update/Delete Performance Scalability ...

- RTS index last used (NFM)
  - ▶ SYSINDEXSPACESTATS.LASTUSED indicates when index used in SELECT/FETCH, searched UPDATE/DELETE, and Referential Integrity check, but not INSERT, LOAD, etc.
- Index page split reduction
  - ▶ Bigger index page (NFM)
    - 4K, 8K, 16K, or 32K page
      - Up to 8 times less index split
    - Good for heavy inserts to reduce index splits

## Insert/Update/Delete Performance Scalability ...

- Index page split reduction...
  - ▶ Asymmetric index page split depending on an insert pattern when inserting in the middle of key range (NFM)
    - Instead of 50-50 split
    - Up to 50% reduction in index split
    - 20% class 2 cpu and 31% elapsed time improvement in one data sharing measurement
    - 10% cpu and 18% elapsed time improvement in one non data sharing measurement

## Database buffer scalability/performance

- Bigger prefetch and deferred write quantity for bigger buffer pool (CM)
  - ▶ Max of 128 V8 ->256KB V9 in SQL tablespace scan
  - ▶ Max of 256 V8 ->512KB V9 in utility
- Bigger preformatting quantity and trigger ahead (CM)
  - ▶ From 2 (V8) to 16 (V9) cylinders if >16cyl allocation
- Replace all sequential prefetch, except in tablespace scan, with dynamic prefetch in SQL calls (CM)
  - ▶ A big reduction in synchronous read I/Os possible in some cases

## Database buffer scalability/performance ...

- Heavier use of 32K workfile BP instead of 4K BP (CM)
  - ▶ V9 tries to use 32K BP for bigger record size to gain improved performance, especially I/O time
    - Less workfile space and faster I/O
      - Example: 15 2050byte records on one 32K page vs 8 records on 8 4K pages
  - ▶ Recommendation
    - Assign bigger 32K workfile BP
    - Allocate more 32K workfile datasets
    - If 4K workfile BP activity is significantly less, corresponding BP size and workfile datasets can be reduced.
  - ▶ Statistics on how often more optimal 4K workfiles ran out and 32K workfiles used instead, or vice versa

## Query/Access Path Performance Enhancement

- Optimizer cost model update (CM)
- Sort (CM)
- Sparse index or in-memory data extension (CM)
- More parallelism (CM)
- Cross-query block optimization (CM)
- Pair-wise join in star schema (CM)
- Histogram statistics (NFM)
- Automatic reoptimization (NFM)
- Varchar performance improvement (NFM)
- Index on expression (NFM)
- Concurrency improvement (NFM)

## Access Path Enhancement

- Optimizer cost model update (CM), eg
  - ▶ Cluster ratio
  - ▶ Parallel access path selection separate from sequential
  - ▶ List prefetch based on Cost Model, not Cluster ratio
  - ▶ Recommendations
    - Make sure new ZPARM STATCLUS is set to ENHANCED (Default)
    - Run with RUNSTATS TABLE(ALL) INDEX(ALL) KEYCARD as default

## Access Path Enhancement – continued

- Large record sort using 32K workfile page (CM)
- In-memory work file for small sorts (CM)
  - ▶ 10 to 30% cpu reduction for short-running SQL with small sorts
- Fetch First N Rows with Order By (CM)
  - ▶ By avoiding tournament tree sort for small N
  - ▶ Up to -50% cpu in one measurement test
  - ▶ Supported in Subselect also in V9 (NFM)
- Group By sort improvement when no column function and no available index, eg `SELECT a1 FROM A GROUP BY a1` (CM)
  - ▶ Up to 2 times improvement
- Distinct sort improvement when no index or only non unique index available, eg `SELECT DISTINCT a1 FROM A` (CM)

## Access Path Enhancement – continued

- More parallelism leading to more zIIP offload (CM)
  - ▶ Especially in star schema queries
  - ▶ Access path selection for parallelism separate from sequential
- Cross query block optimization in complex query (CM)
  - ▶ Optimization across, rather than within, query blocks
  - ▶ More predicate transitive closure across query blocks
- Pair-wise join in star schema queries (CM)
- Index on expression (NFM)



## Access Path Enhancement – continued

- Histogram statistics over a range of column values (NFM)
  - ▶ For more precise filter factor estimates and better access path selection when literals are used
  - ▶ Useful in range as well as equal predicates with high cardinality, eg NAME in contrast to STATE
  - ▶ Equal-depth (each interval with roughly same number of rows)
- REOPT AUTO (NFM)
  - ▶ REOPT None, Once, or Always in V8
  - ▶ Automatic REOPT based on parameter marker change
    - Only for dynamic statements that can be cached

## Concurrency Improvements

- Default bind option change (NFM)
  - ▶ Isolation RR to CS and Currentdata Yes to No
  - ▶ For reduced locking cpu and better concurrency
- Skip locked data option in SELECT (NFM)
  - ▶ This option skips locked row or page, while Uncommitted Read isolation mode does not
    - For CS or RS, not UR or RR
  - ▶ Conditional instead of unconditional lock request issued
    - Instant lock with immediate unlock for each row or page

## Utility CPU time reduction (CM) – primarily from index processing except Image Copy

- 10 to 20% in Image Copy, Recover index
- 5 to 30% in Load, Reorg, Rebuild Index
- 20 to 60% in Check Index
- 35% in Load Partition
- 30 to 40% in Runstats Index
- 40 to 50% in Reorg Index
- Up to 70% in Load Replace Partition with NPIs and dummy input

## V9 zIIP Additional Offload Opportunities

- DRDA over TCP/IP
  - ▶ Native SQL procedure under DRDA (NFM)
- Parallel query
  - ▶ Access path optimized for parallelism (CM) leading to more zIIP offload
- Load/Reorg/Rebuild utility
  - ▶ Less (0 to 40%) offload as big CPU reduction in build index in V9

## V9 zAAP Offload

- zAAP (System z Application Assist Processor) offload of XML System Services in TCB (zIIP offload if DRDA SRB)
  - ▶ Prerequisites: DB2 9 **NFM**, z9, z/OS 1.9 or 1.7/1.8 with PTF
  - ▶ XML Insert/Update/Load offload percentage depends on document size, complexity, number of indexes, etc.
  - ▶ Acctg support via PK50575
  - ▶ XML zAAP whitepaper on ....
    - <http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101088>

## Native SQL Procedure (NFM)

- Avoiding the stored proc invocation overhead and roundtrip between Work Load Manager and DBM1 address spaces for each SQL call
  - ▶ 0 to 40% ITR improvement compared to external SQL procedure
  - ▶ No difference if long-running SQL
- Some SELECT replaced by SET statements
- zIIP-eligible if DRDA as it runs in DBM1, not WLM, address space under DDF enclave SRB
- V9 PK45265 8/07 (also V8 PK28046 9/06) to reduce LC24, EDM pool full, and unavailable storage when same external stored procedure or native SQL procedure is called multiple times without an intervening commit or close result sets

## Index Compression (NFM)

- Difference between data and index compression

	<b>Data</b>	<b>Index</b>
<b>Level of compression</b>	<b>Row</b>	<b>Page (1)</b>
<b>CPU overhead</b>	<b>In Acctg</b>	<b>In Acctg and/or DBM1 SRB</b>
<b>Comp in DASD</b>	<b>Yes</b>	<b>Yes</b>
<b>Comp in BP and Log</b>	<b>Yes</b>	<b>No</b>
<b>Comp Dictionary</b>	<b>Yes</b>	<b>No (2)</b>
<b>'Typical' Comp Ratio CR</b>	<b>10 to 90%</b>	<b>25 to 75% (3)</b>

- Load or Reorg not required for IX compression
- DSN1COMP utility to simulate compression ratio without real index compression

## Index Compression – continued

- No compression or decompression in each Insert or Fetch; instead at I/O time
- CPU overhead critically sensitive to index BP hit ratio
  - ▶ Bigger index BP strongly recommended for index compression
  - ▶ No change in acctg CPU time if index pages are brought in by prefetch
- Work area for compressed IX I/O is long term page fixed
  - ▶ So do not page fix compressed index buffer pool
    - If a mix of compressed and non compressed indexes, consider using a separate buffer pool



## Varchar Performance ...

- 2 times or more improvement observed when many rows with many varchars are scanned and/or fetched using many predicates
- <5% improvement for a typical online transaction

## LOB

- Reorg LOB to reclaim space (CM)
- LOB read/write I/O performance improvement (CM)
- LOB lock avoidance (NFM)
  - ▶ Up to 100% reduction in Lock and Unlock requests in Fetch
  - ▶ One measurement with SAP optimized LOB streaming
    - -67% IRLM requests
    - -26% class2 accounting CPU time
    - -14% elapsed time
- LOB/XML flow optimization by size (NFM)
  - ▶ V9 DRDA LOB handling instead of SAP optimized LOB handling
    - Additional 11% elapsed time and 2% CPU time improvement

## Transaction and Utility CPU Time Summary

- 0 to 10% increase on average tran from V7 to V8
- No change on average tran from V8 to V9
  - ▶ 0 to 5% improvement especially for column-intensive transaction
    - Varchar performance improvement (NFM)
    - Long displacement instructions (CM)
    - DDF/DBM1 shared storage (CM)
    - Partition processing DSNB1DCM/CPS (CM)
    - Index access (CM)
    - DSNXECW (CM)
- 5 to 70% improvement for Utility depending on %CPU time in index processing, except Online Reorg of few out of many partitions with NPIs (CM)

## NOTES

- Long displacement instructions on z990 and higher
  - ▶ To avoid module split and promote more efficient register-to-register instructions
  - ▶ Bigger benefit with many input host variables and/or output columns
  - ▶ Up to 15% CPU reduction in many column Fetch/Insert
  - ▶ Simulated by microcode on z900 and can result in 10% or higher cpu time for column-intensive applications
- Table space, index, partition declaim cpu time reduction by only accessing partitions claimed
- DSNXECW cpu time reduction by cleaning up unused statement section at commit for JDBC/CLI