DB2 for z/OS V11 Overview

Kevin Harrison

Sr. Systems Engineer/Sr. IT Architect
IBM Silicon Valley Lab
kharrison@us.ibm.com

Disclaimer and Trademarks

Information contained in this material has not been submitted to any formal IBM review and is distributed on "as is" basis without any warranty either expressed or implied. Measurements data have been obtained in laboratory environment. Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. The use of this information is a customer responsibility.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

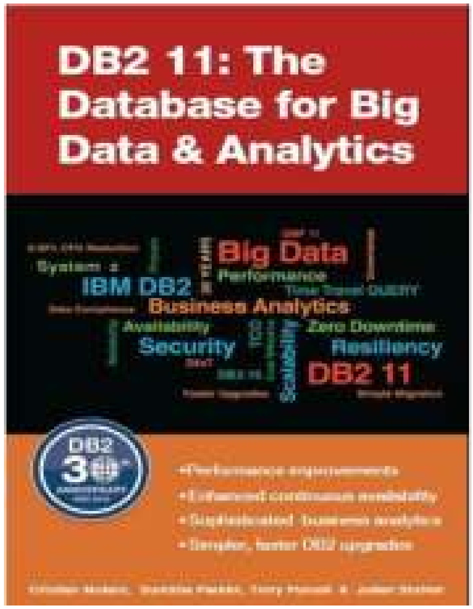
TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, DATABASE 2, DB2, Enterprise Storage Server, FICON, FlashCopy, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, System z, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, zSeries

The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95

For additional information see ibm.com/legal/copytrade.phtml

DB2 11 Major Themes

- **Out-of-the-box CPU Savings***
  - Improving efficiency, reducing costs, no application changes
  - Up to 10% for complex OLTP
  - Up to 10% for update intensive batch
  - Up to 40% for queries
  - Additional performance improvements through use of new DB2 11 features
- **Enhanced Resiliency and Continuous Availability**
  - Improved autonomics which reduces costs and improves availability
  - Making more online changes without affecting applications
  - Online REORG improvements, less disruption
  - DROP COLUMN, online change of partition limit keys
  - Extended log record addressing capacity - 1 yottabyte (or 1B petabytes)
  - BIND/REBIND, DDL break into persistent threads
- **Enhanced business analytics**
  - Expanded SQL, XML, and analytics capabilities
  - Temporal and SQLPL enhancements
  - Transparent archiving
- **Simpler, faster DB2 version upgrades**
  - No application changes required for DB2 upgrade
  - Access path stability improvements
  - Product quality/stability: support pre GA customer production



*REBIND may be required for best results

DB2 11 Performance Sweet Spots

- **Write Intensive Batch**
- **Queries**
  - With compressed tables
  - With access path improvement
  - With sort intensive workload
  - Accessing multiple DPSI partitions
  - IDAA with large result sets
- **Online transactions**
  - Write intensive transactions
  - With large # of partitions (>200 partitions) with REL(COMMIT)
  - With large buffer pools
  - With queries returning a large number of columns
  - Chatty DDF applications with z/OS Communications Server PM80004 / UK92097 (closed in Feb. 2013)
- **Cost saving from zIIP eligible address space SRB time**
  - DBM1 in data sharing
  - MSTR address space for update intensive workloads

Performance Improvements – no REBIND needed

- **DDF performance improvements**
  - Reduced SRB scheduling on tcp/ip receive using new CommServer capabilities
  - Improved autocommit OLTP performance
  - DRDA package based continuous block fetch
- **xProcs above the bar**
  - 31-bit Vstor relief enabled by RMODE 64 support in z/OS 1.13 and above
  - Enables other internal performance improvements
- **zIIP enablement for all SRB-mode DB2 system agents that are not response time critical**
- **Avoid cross-memory overhead for writing log records**
- **INSERT performance**
  - Latch contention reduction for classes 6, 14, 19
  - CPU reduction for Insert column processing and log record creation
  - Data sharing LRSN spin avoidance
  - Page fix/free avoidance in GBP write

Performance Improvements – no REBIND needed…

- **Automatic index pseudo delete cleanup**
  - DBA work would be required for fine tuning
- **ODBC/JDBC type2 performance improvements**
  - Stored procedure invocation
- **Java stored procedure multi-threading improvements**
- **Sort performance improvements**
- **DPSI performance improvements for merge**
- **Performance improvements with large number of partitions**
- **XML performance improvements**
- **Optimize RELEASE(DEALLOCATE) execution so that it is consistently better performing than RELEASE(COMMIT)**
  - Monitor # parent locks and cleanup internal structures when threshold is hit
- **IFI 306 filtering capabilities to improve Replication capture performance**
- **Utilities performance improvements**

Performance Improvements – no REBIND needed…

- **ACCESS DATABASE command performance**
- **DGTT performance improvements**
  - Avoid incremental binds for reduced cpu overhead
- **P-procs for LIKE predicates against Unicode tables**
- **Improved performance for ROLLBACK TO SAVEPOINT**
- **zEC12 integration for performance improvements**
  - Pageable 1M frames for DB2 CPU savings (requires Flash Express)
    - Buffer pool control structures (retrofit to V10)
    - DB2 executable code (requires z/OS 2.1)
  - 2G page frame size to position for extremely large main memory sizes
  - Optimizer CPU and I/O cost balancing improvements (can also benefit z196 and z10)
- **Latch contention reduction and other high n-way scalability improvements**
- **Data sharing performance improvements**
  - LRSN spin reduction with extended LRSNs
  - Castout performance
  - GBP write-around
  - Index split performance

Performance Improvements – REBIND required (with or without APREUSE)

- **Query transformation improvements – less expertise required to write performaning SQL**
  - Enhanced query rewrite to improve predicate indexability
    - new situations where non-indexable predicates can be rewritten by Optimizer to be indexable
    - Convert some common stage 2 predicates to indexable (YEAR(), DATE(), SUBSTR(col,1,x), value BETWEEN COL1 AND COL2)
    - Improved indexability for OR COL IS NULL predicates
    - Push complex predicates inside materialized views/table expressions
  - Enhanced pruning of "always true" and "always false" predicates
- **Enhanced duplicate removal**
  - Lots of queries require duplicate removal: e.g. DISTINCT, GROUP BY, etc.
  - Dup elimination via sorting can be expensive
  - New techniques: Index duplicate removal, early out
  - Will not show in Explain table, need to look at IXSCAN_SKIP_DUPS column in DSN_DETCOST_TABLE to determine if sort avoided
- **DDF and RDS runtime result set optimizations**
  - Reduced DB2 CPU for IDAA queries

Performance Improvements – REBIND required (with or without APREUSE)…

- **In-memory techniques**
  - In-memory, reusable workfile
  - Sparse index (limited hash join support)
  - Non-correlated subquery using MXDTCACH
  - Correlated subquery caching
- **Non correlated subquery with mismatched length**
- **Data decompression performance improvement**
- **Select list do-once**
  - Non column expressions in the select list can be executed once rather than per-row
- **Column processing improvements**
  - Xproc (generated machine code) for output column processing
  - Optimized machine instructions for input/output column processing

Performance Improvements – REBIND required (with or without APREUSE)…

- **RID overflow to workfile handled for Data Manager set functions**
  - – DB2 10 added RID overflow to workfile
  - – DB2 11 adds support for set functions (COUNT, MAX, MIN etc) which was excluded in DB2 10
- **Performance improvements for common operators**
  - – MOVE, CAST, output hostvar processing, CASE, SUBSTR, DATE, others
- **DECFLOAT data type performance improvements**
  - – Up to 23% CPU reduction for conversion to/from decfloat
  - – Approx. 50% cpu reduction in INSERT, FETCH for decfloat columns
  - – Helped further by zEC12 hw improvements for decimal floating point

Performance Improvements – REBIND required (without APREUSE)

- **DPSI and page range performance improvements**
  - Page range screening for join/correlation predicates
  - Parallelism optimization for DPSI access

- **Optimizer CPU and I/O cost balancing improvements**
  - Measured results: 3% to >30% performance improvement for query workloads

Performance Improvements – DBA or application effort required

- **Suppress-null indexes**
  - – Index entries not created when all values for indexed columns are NULL
  - – Reduced index size, improved insert/update/delete performance, compatibility with other DBMSes
  - – Improved utility CREATE INDEX performance
- **New PCTFREE FOR UPDATE attribute to reduce indirect references**
- **DGTT performance improvements**
  - – Non logged DGTTs
- **Global variables**
  - – Easier, more efficient sharing of data between SQL statements

Performance Improvements – DBA or application effort required

- **Optimizer externalization of missing/conflicting statistics**
  - Identify missing statistics during bind/prepare/explain
  - DBA or tooling to convert output to RUNSTATS input
- **Extended optimization - selectivity overrides (filter factor hints)**
  - Improve optimizer's ability to find the cheapest access path
  - Collect filter factors for predicates in a Selectivity Profile
  - Selectivity Profile is populated via BIND QUERY
- **Open dataset limit raised to 200K**

DB2 11 and zEC12 Synergy

- Faster CPU – 1.25x compared to z196
  - 5.5GHz processors, bigger/faster cache
  - 25% reduction measured with DB2 workloads
- 50% More System Capacity to help consolidation
  - Up to 3TB real memory per server
  - Excellent synergy with DB2 10 and 11 scalability
- New Features that DB2 11 Exploits
  - FLASH Express and pageable 1MB frames, used for:
    - Buffer pool control blocks
    - DB2 executable code
  - 2GB frame support for buffer pools
    - Performance improvement expected for extremely large memory sizes
- New zEC12 GA2 features that benefit DB2
  - zEDC Express for enhanced DB2 SMF data compression
  - RoCE Express for faster, cheaper z/OS to z/OS DRDA communication
    - Preliminary measurements show up to 2x DRDA transaction throughput increase

DB2 and IBM zIIP Add Value to Database Work

Portions of the following DB2 workloads in enclave SRB mode are eligible for zIIP*

DB2 9 in blue DB2 10 in green DB2 11 in orange

1. **DRDA over TCP/IP connections: up to 60% of the processing**
   - DB2 9 for zOS remote native SQL procedures
   - DB2 9 XML parsing, schema validation

2. **Requests that use parallel queries: up to 80% of the processing after reaching a CPU usage threshold**
   - DB2 9 and DB2 10 remove restrictions for query parallelism enabling more queries to run with parallelism and therefore to potentially increase zIIP eligibility

3. **DB2 utilities: up to 100% of the processing**
   - LOAD, REORG and REBUILD functions used to maintain index structures and sort
   - DB2 10 RUNSTATS – options other than column group, inline
   - DB2 11 RUNSTATS column group and inline

4. **Asynchronous processing that is charged to a DB2 address space (introduced in DB2 10, expanded in DB2 11): up to 100% of the processing**
   - DB2 10 buffer pool prefetch and deferred write
   - All other such asynchronous processing, except for P-lock negotiation

* NOTE: This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Feature Disclaimer

- The topics in this short overview were hand picked based on V11 ESP feedback and ranking
- There is a significant list of technical features both large and small in the release not included here
- Your local DB2 Advisor can offer an expanded DB2 MPW that will discuss many more items in the V11 release

Connectivity Enhancement Business Value

- **Performance Improvements**
  - All remote applications should see CPU and network latency reductions by DB2 exploiting a new sync receive model
  - Any remote application using Autocommit should see significant CPU and network latency reductions by default
  - Read-only z/OS application requester should see significant CPU and network latency reductions by exploiting new blast protocol
    - Z requester to Z server only
- **Serviceability Improvements**
  - Enhanced client information
  - Improvements to cancel thread and client interrupt processing
  - Improved application profiling capability

Synchronous Receive - All remote applications…

- **Pre DB2 11 DDF uses asynchronous receive, which requires expensive SRB switch when data becomes available**
  - Async receive necessary to detect client connection termination so DDF can cancel thread immediately for long processing SQL statements
- **DB2 11 DDF will exploit synchronous receive**
  - z/OS Communications Server provides new Termination Notification
    - z/OS 1.13 via APAR (PM80004) or z/OS 2.1
  - DDF will be notified of any socket failure, even while in sync receive mode
  - DDF can abort threads running synchronously
  - Available in CM without application changes or REBINDs required
  - Preliminary measurements using the distributed IRWW suite
    - 6% to 8% ITR improvement
    - 4% to 6% CPU improvement
- **Only affects non-stored procedure workloads**

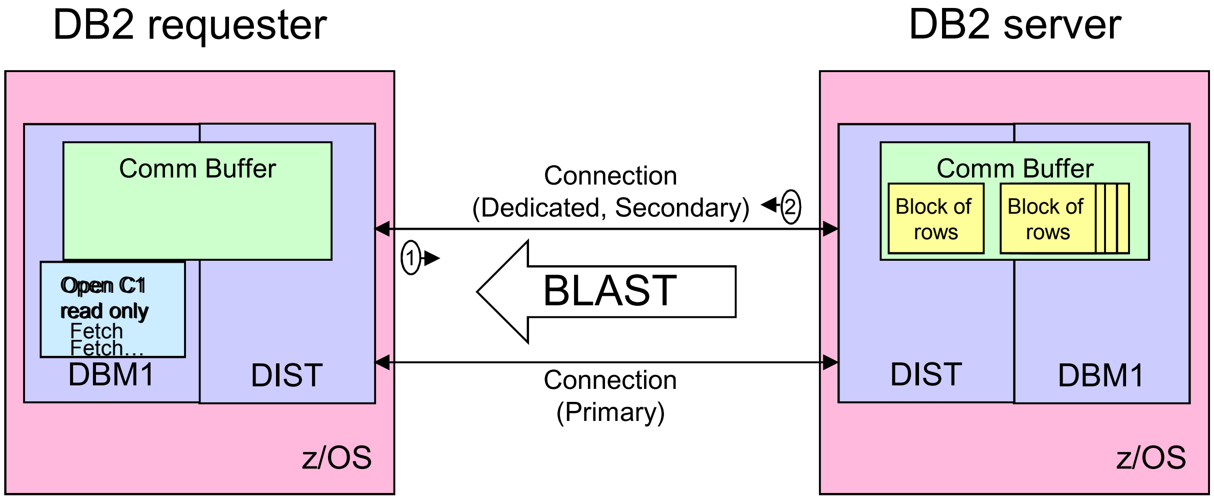
Remote Applications using Autocommit

- **Pre V11 Distributed applications using Autocommit:**
  - – For most SQL, requester can chain commit after request without additional network flow
  - – For stored procedure requests, requester cannot chain a commit
    - Commit request sent in separate flow after result set queries closed, causing delay
- **DB2 11 supporting client driver provides a new Autocommit indicator**
  - – Allows DB2 to initiate commit after all result sets or cursors closed on initial reply
    - Achieve efficient Autocommit feature when calling stored procedures
  - – **Requires updated DB2 V10.5 IBM Data Server Driver**
    - No application changes or property setting required when using Autocommit
- **Benefits**
  - – Preliminary measurements using the distributed IRWW suite
    - 11% reduction in elapsed time
    - 4**%** reduction in CPU time
  - – Available in CM, no REBIND

Package Based Continuous Block Fetch …

- **New capability for DB2 read-only z/OS applications called blasting**
  - Requester package is required to be bound with DBPROTOCOL(DRDACBF)
  - Application is required to set APPLCOMPAT to V11R1
  - Any statement that performs an update on the server will fail

    -817 THE SQL STATEMENT CANNOT BE EXECUTED BECAUSE THE STATEMENT WILL RESULT IN A PROHIBITED UPDATE APPLCOMPAT(V11R1)
  - Query blocks flow on secondary connection until cursor exhausted
    - Each read-only cursor opens a secondary connection
    - Impacts DBAT settings
  - When result set or cursor is exhausted, server terminates connection and thread is immediately pooled



Package Based Continuous Block Fetch

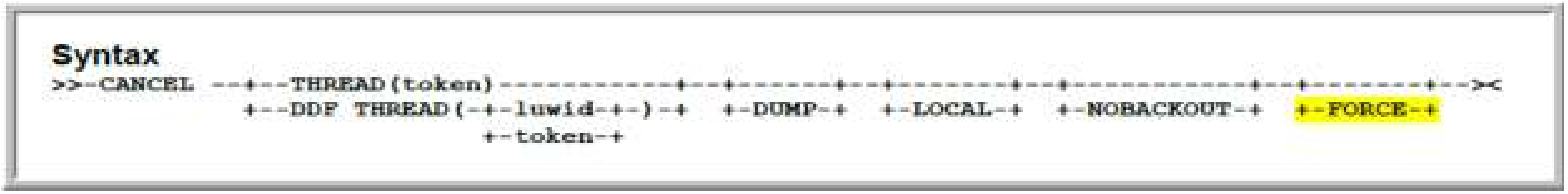
- **Benefits**
  - DB2 more effective in supporting applications that query very large result sets
  - Requests for next block no longer required
  - Internal moves between application and communications I/O buffers eliminated
- **Network latency significantly improved**
  - Additional threads to process queries in parallel on server
  - Concurrency improved for queries running under separate connections,
  - Better thread utilization because server thread is pooled when query is complete.
  - Commit not required to pool DBAT
  - Available in NFM and requires APPLCOMPAT = V11R1 to be set

Exploit Enhanced Client Information Fields …

- **Longer client information fields exploited for distributed applications**
  - Pre V11 - value set by the API is truncated to 16 bytes
  - V11 - value set by the API supports extended lengths (NFM)
  - Tolerated in CM, still truncated to16 bytes (same as pre V11)
  - Longer client information fields are exploited in NFM + APPLCOMPAT V11R1
- **Longer field support for special registers:**
  - CLIENT_USERID supports up to 128 bytes
  - CLIENT_WRKSTNNAME supports up to 255 bytes
  - CLIENT_APPLNAME supports up to 255 bytes
  - CLIENT_ACCTNG supports up to 255 bytes
- **Longer fields supported in DDF traces, messages and displays in CM**
  - Resource Limit Facility (RLF) can govern SQL statement activity based on longer client information fields in NFM
  - System Profile Monitoring supports longer client information fields in NFM
  - WLM classification rules can utilize longer client information fields when running z/OS V2R1

Cancel thread and Interrupt statement …

- **Pre DB2 11 DDF CANCEL THREAD may not work in some cases**
  – Historically "soft", or reactive in nature
  – Cancel detection point may not be encountered in a timely manner
- **DB2 11 New FORCE keyword on CANCEL THREAD command**
  – Purges thread of a remote connection in the DB2
  – FORCE option accepted only after CANCEL THREAD issued without the FORCE option
  – Available in CM
- **Purge running DDF SRB using new z/OS purge SRB function**
  – Requires new z/OS CALLRTM TYPE=SRBTERM service
    - Available in z/OS 1.13 or later via APAR OA39392



Syntax

```
>>-CANCEL --+--THREAD(token)------------+--+------+--+-------+--+-----------+--+-------+-->< 
            +--DDF THREAD(-+-luwid-+-)-+   +-DUMP-+   +-LOCAL-+   +-NOBACKOUT-+   +-FORCE-+
                           +-token-+
```

BIND/DDL Break-in

- **DB2 10 DDF threads**
  - To enable
    - Rebind packages with RELEASE(DEALLOCATE)
    - Issue command –MODIFY DDF PKGREL(BINDOPT)
  - To disable
    - Wait 200 commits (existing DBATs)
    - –MODIFY DDF PKGREL(COMMIT), which only effects DBATs created after the command
- **There is no solution for non-DDF thread in DB2 10**
- **DB2 11 DDF threads**
  - Enable same as was done in DB2 10
  - To disable
    - Automatically done on next COMMIT if waiter on a package lock
- **DB2 11 non-DDF threads**
  - Automatically done on next COMMIT if waiter on a package lock
  - Idle threads break in is coming soon (PM95929, PM96001, PM96004)
  - zPARM PKGREL_COMMIT=YES must be set (parameter is online changeable)
    - Default is YES
  - Idle threads are also handled
  - Not designed to handle long running threads that don‘t commit or transactions with held cursors
- **Packages resume normal RELEASE(DEALLOCATE) behavior after the break-in operation completes**

Group Buffer Pool Write Around Protocol ...

- **Circumvent situations where significant, sustained page write workload causes**
  - Application slow downs
  - Pages to be placed in the LPL in extreme cases
- **DB2 detects large write activity and**
  - Write to the Group Bufferpool if the page is already cached
  - Otherwise write are directed to DASD
    - 50% of GBP or 20% of Class
- **These are actions beyond the use of the Auto Alter option**

Group Buffer Pool Write Around Protocol

- **Normal GBP use is restored when the storage shortage is relieved**
  - 40% of GBP or 10% of Class
- **Available in CM without a REBIND, with**
  - 1.13 with OA37550 PTFs or
  - z/OS 2.1 and
  - CFLEVEL=17
- **DISPLAY GROUPBUFFERPOOL ... MDETAIL**
  - WRITE-AROUND PAGES instrumentation in DSNB777I
  - Also in IFCIDs 0002 & 0003

64 Bit Enhancements

- **More “Above the Bar” exploitation**
  - xPROCS / zPROCS, moving above the bar
  - RMODE64 now available
  - Select, Insert, Update Predicates and Keys affected
  - REBIND of static package required to move all xPROCS above the bar
- **Virtual Storage Improvements and UTS PBG Table Spaces**
  - DB2 10 thread allocation based on maximum number of PBG partitions
  - DB2 11 thread allocation is done for 20 partitions initially, then for another 20 as more partitions accessed
- **More CSA/ECSA moving above the bar**
  - HVSHARE growing from 128GB to 1T
  - Addressable Range, not allocated unless used
- **DB2 11 uses 64 bit Common for Log Buffers**
  - Avoids cross memory access for logging

Hardware Related Storage Exploitation

- **1 MB page frame utilization**
  - Log Buffers backed by 1 MB frames
    - Allocation precedence over buffer pools
    - May want to increase LFAREA 1 MB page fix to accommodate output log buffers
  - Exploit pageable 1 MB frames to take advantage of CPU reduction for Translation Lookaside Buffers
    - Buffer Pool control blocks (not data pages)
    - EC12 required
    - Flash Express Memory required
    - RSM Enablement Offering (FMID JDB778H)

Buffer Pool enhancements …

- **Buffer pool FRAMESIZE attribute**
  - DB2 11 allows FRAMESIZE(2G)
    - PGFIX(YES) required
    - Requires z/OS 2.1 & EC12 (possible retro to 1.13) with OA40967
    - If PGFIX(NO), then buffer pool allocated using 4K pages
  - DB2 11 allows FRAMESIZE(1M) with PGFIX(NO)
    - Control blocks pageable (PMB – Page Manipulation Blocks)
      - LFAREA to include control blocks
      - ((VPSIZE*pagesize/1024/1024) + 6399)/6400 = # of MBs of storage used for the PMBs
    - Data pages will remain 4K
    - Requires EC12 and Flash Express – otherwise demoted to 4K
    - CPU savings by reducing Translation Lookaside Buffer access
      - When you cannot perform PGFIX(YES)

Buffer Pool enhancements…

Frame size	Page fix	Supported DB2	H/W Requirement	Benefit
4K	NO	All	N/A	Most flexible configuration
4K	YES	All	N/A	CPU reduction during I/O
1M	NO	DB2 10 with APAR, or DB2 11	zEC12 and Flash Express Backed by real or LFAREA	CPU reduction from TLB hit Control Blocks only, also applies to 2G Framesize
1M	YES	DB2 10 above	z10 above LFAREA 1M=xx	CPU reduction during I/O, CPU reduction from TLB hit
2G	YES	DB2 11	zEC12 LFAREA 2G=xx	CPU reduction during I/O, CPU reduction from TLB hit

* If any HW/SW requirements unmet, 4K frames used

Buffer Pool enhancements…

- **New attributes VPSIZEMIN, VPSIZEMAX for WLM managed**
  - Requires AUTOSIZE(YES)
  - VPSIZEMIN(integer | *)
    - Integer is minimum number of buffers to allocate
    - * is default, indicates DB2 will set the minimum value to 75% of current size
  - VPSIZEMAX(integer | *)
    - Integer is maximum number of buffers to allocate
    - * is default, indicates DB2 will set the maximum value to 125% of current size
- **BP Reclassification**
  - DB2 9/DB2 10
    - Pages remain classified as prefetch even after random I/O against page
  - DB2 11
    - Prefetched pages can be reclassified as random if random getpage against page
    - Potentially less sync I/Os for mixed access (prefetch + random)

Work File Space Instrumentation ...

- **New System Parameters for Work File alerts**
  - See Migration presentation
  - WFSTGUSE_AGENT_THRESHOLD
    - Agent level warning
    - Percentage of work file space in use by an agent
    - Message DSNI052I issued when exceeded
      - Once per unit of work
  - WFSTGUSE_SYSTEM_THRESHOLD
    - System level warning
    - Percentage of work file space in use by the system
    - Message DSNI053I issues when exceeded
      - With PM94722, this message is not repeated for 5 minutes
  - If WFDBSEP=YES
    - The percentages are of the segregated temporary spaces
    - Messages indicate if due to DGTTs, Work Files, or Both

Work File Space Instrumentation

- **The space map pages are updated under an S Latch**
  - Reducing space map contention
- **IFCIDs**
  - 2
    - Statistics Class 1
    - MXDTCACH storage use / shortages resulting in work file use
  - 27
    - Types of Sparse Indexes used and storage consumed
  - 311
    - Performance Class 8
    - DGTT Information
    - See DSNDQW04 member of SDSNMACS for more information

How DB2 11 uses the new RBA/LRSN format

- **DB2 11 is fundamentally a 10-byte format system**
  - Catalog columns are created / expanded during migration
- **All values are kept in 10-byte format internally**
  - Avoids the complexity of a dual code path for dealing with 6-byte and 10-byte formats
  - Conversion is done for DB2 objects still in BASIC format
    - 6 byte values are mapped into a 10 byte bucket
- **DB2 messages use 10-byte format in all DB2 11 modes**
  - Allows a consistent message format in DB2 11
  - e.g. for QUIESCE or REPORT RECOVERY output
    - LRSN values contain either padded x'00' or precision value
- **You need to be aware of this when going to DB2 11 CM!**
  - Maintenance procedures and 3rd party tools must support DB2 11
    - E.g. IFCID 306 (always new log format)

zIIP Expansion

- **zIIP for all asynchronous processing executed as SRB mode system agents**
- **Not expected to assist as much as V10, but can provide increase use**
- **Up to 100% of processing for asynchronous enclave service request blocks that execute in the DB2 ssnmMSTR, ssnmDIST, ssnmDBM1 and ssnmDIST address spaces.**
  - Prefetch (V10)
  - Deferred write (V10)
  - Log write and read
  - Index pseudo delete / XML clean up
- **Expanded RUNSTATS and LOAD redirect**
  - Enclave SRB tasks for LOAD, REORG, and REBUILD INDEX Inline Stats
  - Enclave SRB tasks for RUNSTATS Column Group Distribution Stats
  - Enclave SRB tasks for LOAD REPLACE index management
- **IIPHONORPRIORITY=NO**
  - Disables system agent (including prefetch) redirect

Selected New / Changed IFCIDs

- **New IFCIDs in DB2 11**
  - 374: Routine / Package cache maintenance
  - 376: Statement incompatibility reporting
  - 377: Pseudo deleted index entry clean
  - 382 / 383: Parallel task synchronization suspend / resume
  - 384 / 385: Trace statistics feedback recommendations
  - 386: ENF Signaling
- **Changed IFCIDs in DB2 11**
  - IFCIDs 217 & 225 enhanced with IRLM private storage information
  - IFCID 204 modified for RBA / LRSN expansion
  - IFCID 306 returns a subset of log records if the log reader provides an array of DBID / PSID pairs (IFI Log Filter)
    - Retrofitted to DB2 10 via PM90568

SQL Enhancements

- **Application Compatibility**
- **Global Variables**
- **Native SQL PL Enhancements**
  - Arrays
  - Autonomous Transactions
- **GROUP BY Expansion**
  - Grouping Sets
  - Super Groups

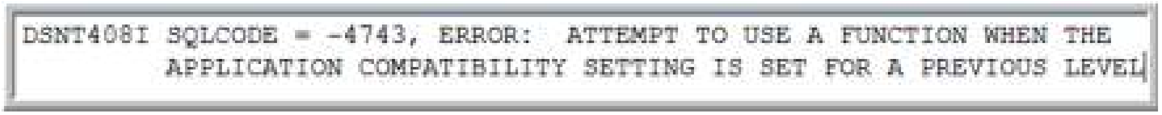
Application Compatibility ...

- **Occasionally SQL functions change behavior**
  - Usually in support of family compatibility and SQL standards
  - This introduces an application incompatibility which can
    - Delay version migrations
    - Potentially create a single version charge issue
  - These are documented in the Release Incompatibilities

- **Example DB2 10 incompatibilities & resolutions**
  - CHAR function results (also for VARCHAR and CAST of these data types)
    - Leading zeroes no longer returned when there is a decimal point
    - ZParm BIF_COMPATIBILITY was introduced to reverse the V10 behavior
    - IFCID 366 introduced to report on use of V9 code path while in V10 ZParm enabled
  - Strong data typing for .NET stored procedures
    - ZParm DDF_COMPATIBILITY reversed this behavior
  - Acceptance of unsupported Timestamp formats was also reversed by BIF_COMPATIBILITY
  - Others required application changes starting in Conversion Mode

Application Compatibility ...

- **DB2 11 fences DML behavior change beginning in CM with APPLCOMPAT**
  - Does not fence DDL or DCL
  - Separates the application migration from the system migration
    - Application migration can begin after the system migration is complete
    - Or the application migration can be delayed for up to 2 future DB2 versions
- **APPLCOMPAT ZParm *and* Bind Parameter**
  - V10R1



```
DSNT408I SQLCODE = -4743, ERROR:  ATTEMPT TO USE A FUNCTION WHEN THE
         APPLICATION COMPATIBILITY SETTING IS SET FOR A PREVIOUS LEVEL
```

    - DML behaves as it did for DB2 10
    - Must use V10R1 until NFM
    - Attempting to use new features under V10R1 results in SQLCODE -4743
  - V11R1
    - Requires the subsystem / group to be in NFM
    - New DML behavior is introduced
    - Also required for new features of V11

Application Compatibility ...

- **ZParm APPLCOMPAT is the default for the BIND / REBIND parameter**
  - Has no effect on existing packages or statements
  - Defaults to V11R1 for installations and V10R1 for migrations
- **BIND / REBIND parameter**
  - Defaults to ZParm APPLCOMPAT
    - After the previous setting for REBIND
  - V11R1 will cause an error before NFM
    - Once NFM, can BIND / REBIND with V10R1 or V11R1 irrespective of the ZParm
  - REBIND existing application when ready for *new behavior*
  - REBIND existing application when wanting to use *new features*
    - WARNING: Must also be ready for potential behavioral changes
- **CURRENT APPLICATION COMPATIBILITY special register for dynamic SQL**
  - Defaults to Package APPLCOMPAT
  - Cannot be set to V11R1 until NFM
    - Once NFM, can bet set to V10R1 or V11R1 irrespective of the Package or ZParm settings
  - Can SET via a System Profile with PM93658 (still open)

Application Compatibility

- **Instrumentation**
  - When using the V10R1 compatibility level, 3 IFCIDs are produced to identify application at risk at the V11R1 level
    - IFCID 239 enhanced to report on Packages that use the V10 code path
    - IFCID 366 (also in V10) identifies Statements in Packages that use V10 code path
    - IFCID 376 reports the same as 366, but attempts to eliminate duplicates
- **Considerations**
  - Behavioral change is possible in the down level compatibility level when needing to conform to SQL Standards
  - This does not fence DDL or DCL, in NFM (system) / V10R1 (package)
    - A Global Variable can be created (DDL) and have authority granted (DCL)
    - But applications cannot SET or reference the Global Variable until V11R1
  - Build your NFM plan to adopt V11R1 for applications

Global Variables ...

- **Benefits**
  - Allows users to share information across SQL statements
  - Allows capability to set once, use everywhere
  - Provides additional security via DB2 GRANT and REVOKE
- **Characteristics**
  - Similar to special registers
  - Their definitions are global and shared across different connections
  - Their contents are only shared within the same connection
  - Each connection maintains its own instantiation
  - Their contents are NOT affected by COMMIT or ROLLBACK

Global Variables ...

- **Enable the sharing of data between SQL statements (including stored procedures) without the need for application logic**
- **Memory variables maintained by DB2 and are available throughout a session**
- **Similar to special registers**
  - Settings last for the connection
  - In thread storage
- **Have access controlled by GRANT and REVOKE statements**
- **There are also Built In Global Variables for DB2 11:**
  - Archive Tables
  - Client IP Address information

Global Variables

- **Easy way to share information between SQL statements within the same application context and among applications**
- **Can be created, instantiated, accessed, and modified by applications through SQL statements**
- **CREATE VARIABLE creates a new variable definition in the DB2 catalog**
  - Definition is shared by all applications
- **Variables are instantiated and maintained at the application thread level**
  - The variable content is only shared among SQL statements within the same connection
- **Access controlled by GRANT and REVOKE statements.**

Example:

```
CREATE VARIABLE PAY INTEGER DEFAULT 0;

SET PAY = (SELECT AVG(SALARY) FROM...) + (SELECT AVG(BONUS) FROM...);

SELECT employee_name FROM...WHERE COMPENSATION > PAY;
```

Array types

An array type is a user-defined data type consisting of an ordered set of elements of a single data type.

- **DB2 11 introduces support for 2 array types**
  - **Ordinary: a defined upper bound on the number of elements and uses the ordinal position as the array index**
  - **Associative: no specific upper bound on the number of elements and each element has an associated index value (useful when you don't know size ahead of time)**
- **Only available in SQLPL**
- **Use of Arrays gives significant performance benefit for TPC-E benchmark**

Example 1:
Create an array type named PHONENUMBERS with a maximum of 5 elements that are of the DECIMAL(10, 0) data type.
CREATE TYPE PHONENUMBERS AS DECIMAL(10,0) ARRAY[5]

Example 2:
Create an associative array type for product descriptions of up to 40 characters long, where the indexes are the product numbers, which are up to 12 chars long:
CREATE TYPE PRODUCTS AS VARCHAR(40) ARRAY[VARCHAR(12)]

ARRAY Built-in Function Support

- **Scalar functions**
  - ARRAY_DELETE / TRIM_ARRAY
  - ARRAY_FIRST / ARRAY_LAST
  - ARRAY_NEXT / ARRAY_PRIOR
  - CARDINALITY / MAX_CARDINALITY
- **Aggregate function**
  - ARRAY_AGG
- **Table function**
  - UNNEST
- **Predicate**
  - ARRAY_EXISTS

Autonomous Procedures

- An autonomous procedure is a Native SQL Procedure that can commit work outside the commit scope of the calling program
- Syntax: new AUTNOMOUS clause on CREATE PROCEDURE
- Executes independently from the calling application
- Always commits its updates before returning to the caller
- Can invoke other nested routines (Procedures, Triggers, UDFs), but not another autonomous procedure
- Can be invoked from an application, Stored Procedure, UDF, or Trigger
- DISPLAY THREAD gives indication of autonomous transactions
- Useful for event or audit logs

__Example__

```
INSERT INTO T1                    NOT COMMITTED
CALL SP1 (autonomous)
        UPDATE T2                 COMMITTED
ROLLBACK
```

Grouping Sets & Super Groups ...

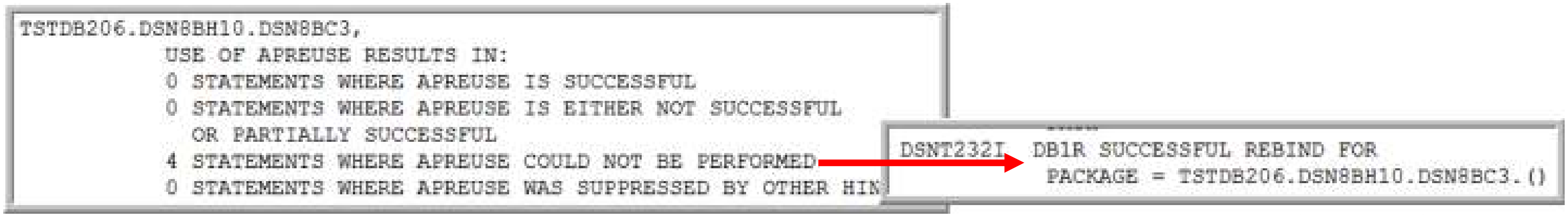
- **More in SQL BI / OLAP function with 3 GROUP BY Super Group options:**
  - GROUPING SETS
  - ROLLUP
  - CUBE
- **GROUPING SETS**
  - Provides for multiple GROUP BY results in a single query
- **SQL Grouping Sets, including Rollup, Cube**
  - Rollup is helpful in providing subtotaling along a hierarchical dimension such as time or geography
  - CUBE is helpful in queries that aggregate based on columns from multiple dimensions

Optimization Evolution

- APREUSE
- Extended Optimization
- Statistics Feedback
- Real Time Stat changes
- Query Transformations
- RDS / Runtime Optimizations
- DPSI improvements
- Parallelism enhancements
- BIND / REBIND changes

APREUSE

- **DB2 10 APCOMPARE / APREUSE stability across BIND & REBIND**
  - APREUSE options were NONE/NO or ERROR
- **DB2 11 provides APREUSE(WARN)**
  - Attempts to hint the same access path for matched statements
  - When the access path cannot be maintained, a new one is calculated for those specific matching statements
  - Effectively operates at the statement level
  - The same considerations from DB2 10 carry forward
    - Uses the Explain Data Block, as does APREUSE(ERROR) and APCOMPARE(WARN / ERROR)
    - Some PLAN_TABLE columns are not hint-able (MATCHCOLS)
      - APREUSE(WARN) will allow the BIND / REBIND to continue
- **DB2 10 APREUSE(ERROR) ... EXPLAIN(ONLY) may represent invalid plan**
  - In DB2 11, this will rolled backed



```
TSTDB206.DSN8BH10.DSN8BC3,
            USE OF APREUSE RESULTS IN:
            0 STATEMENTS WHERE APREUSE IS SUCCESSFUL
            0 STATEMENTS WHERE APREUSE IS EITHER NOT SUCCESSFUL
              OR PARTIALLY SUCCESSFUL
            4 STATEMENTS WHERE APREUSE COULD NOT BE PERFORMED
            0 STATEMENTS WHERE APREUSE WAS SUPPRESSED BY OTHER HIN
```

```
DSNT232I  DB1R SUCCESSFUL REBIND FOR
           PACKAGE = TSTDB206.DSN8BH10.DSN8BC3.()
```

Statistics Feedback ...

- **The Optimizer can provide feedback on missing statistics during**
  - – Access path selection
  - – Explain processing
- **Access path selection (BIND / REBIND / PREPARE)**
  - – Externalized to new Catalog table SYSSTATFEEDBACK
    - Table added in CM
    - Feedback is externalized asynchronously starting in NFM
      - – On STATSINT interval, or
      - – With ACCESS DB command
      - – At the Table level
  - – Controlled by ZParm STATFDBK_SCOPE (ALL)
    - ALL, DYNAMIC, NONE, STATIC
  - – Additional control at the table level
    - SYSTABLES STATS_FEEDBACK which defaults to Y
  - – Redundant recommendations are not created
  - – RUNSTATS for an object removes these associated recommendations

Statistics Feedback ...

	BIND	REBIND	PREPARE	SQL EXPLAIN
SYSSTATFEEDBACK Table/Index level In NFM, governed by ZParm and table control	**Yes** **Including SQLERROR(CHECK) EXPLAIN(ONLY)**	**Yes**	**Yes**	**Yes**
DSN_STAT_FEEDBACK Statement level In CM, governed by existence of the table	**w/ EXPLAIN (YES or ONLY)**	**w/ EXPLAIN (YES or ONLY)**	**w/ CURRENT EXPLAIN MODE YES or EXPLAIN**	**Yes**

- **COLLID for dynamic SQL**
  - Identifies source of dynamic SQL (Cache, CURRENT EXPLAIN MODE YES/EXPLAIN)
- **Use common sense when considering feedback**
- **Consider the use of Statistics Profiles (SYSTABLES_PROFILE)**
  - Not used via Inline Statistics

Query Transformation ...

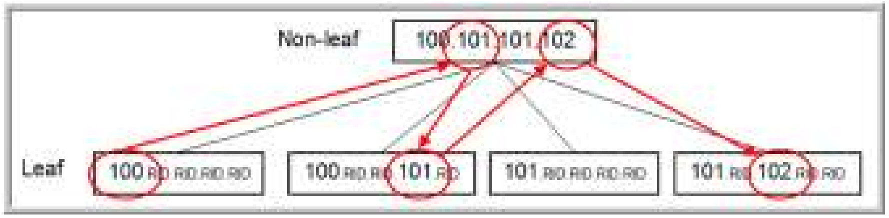
- **Common stage 2 predicates transformed to stage 1...**
  - – Requires REBIND
  - – Index on Expression cases where no IOE exists
    - • If IOE exist, this transformation will not be attempted
  - – YEAR(date column) *op* **:hv** or **?** or **literal**
  - – DATE(timestamp column)
  - – SUBSTR(left most portion)
  - – x BETWEEN col1 AND col2
  - – OR and IN on different columns
    - • Rewritten to ORs (if IN list < INLISTP) for multiple index access
    - • WHERE C1 = ? OR C2 IN (1,2)
      - – becomes WHERE C1 = ? OR C2 = 1 OR C2 = 2

Query Transformation ...

- **Pruning always TRUE / FALSE predicates**
  - Tools / ISV Apps use them (Ex. turn predicates “on” and “off”)
  - In DB2 10 they are evaluated for each row
  - DB2 11 prunes them to avoid this redundant checking
    - DB2 10 already prunes some always false predicates
      - “=“, or “IN” with ORs
      - IS NULL checking of NOT NULL columns with ANDs
  - WHERE, HAVING, and ON clauses
  - Must be literals (no Host Variables or Parameter Markers)
    - REOPT not support
  - **OR 0=1** is not pruned

RDS / Runtime Optimizations ...

- **Index Skipping / Early Out**
  - Duplicate removal
  - Benefits GROUP BY and DISTINCT functions
  - Upward probe to non-leaf page to find the next unique value
  - DSN_DETCOST_TABLE
    - IXSCAN_SKIP_DUPS
    - EARLY_OUT

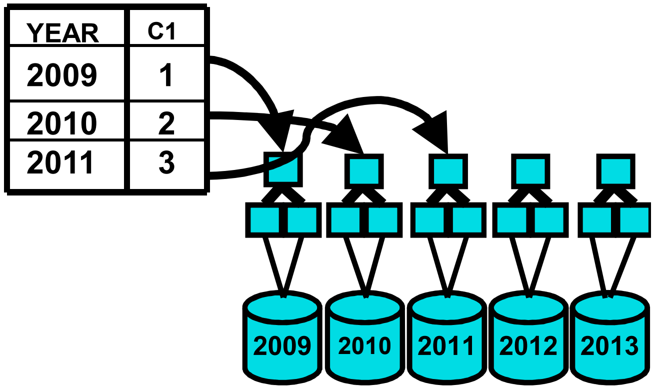


- **RID Pool**
  - Overflow to work file expanded to COUNT, MAX, MIN
  - Restrict Hybrid Join to 80% of total RID Pool
    - Before V11, this could take 100% of the RID Pool
    - Other queries (needing > 1 block) could then run into RID limits

RDS / Runtime Optimizations

- **Sort Improvements**
  - MAXSORT_IN_MEMORY ZParm
    - Defaults to 1MB / query
    - Fences query sort resources
  - Avoid work files for final merge
    - Application fetches from the tournament tree
- **NOT LOGGED DGTTs**
- **Less Overhead for Decompression**
  - Column Level

Data Partitioned Secondary Indexes (DPSI) ...



YEAR	C1
2009	1
2010	2
2011	3

- **Join probing**
  - Before DB2 11
    - Each composite row access all parts
  - DB2 11 can eliminate parts when
    - Qualified on the DPSI columns
    - The partitioning scheme (PI) is qualified in the join predicates
      - Must be Equal (“=“) with same data type / length
    - EXPLAIN
      - PLAN_TABLE.PAGERANGE=Y
      - DSN_FILTER_TABLE.STAGE=PAGERANGE
  - If there is no qualification to eliminate parts (PI columns)
    - The composite table can be replicated for local joins to each part for each Child task

Data Partitioned Secondary Indexes (DPSI)

- **DPSI Return In Order**
  - Keys are ordered within each partition of a DPSI
    - Ordering across has typically involved a sort to return keys like an NPSI
  - DB2 11 can merge the keys without a sort, managed by:
    - The Parent task when parallelism is used (1 Child task per partition)
    - Data Manager when there is no parallelism
    - Supported with Index on Expression
    - May use Index Lookaside
      - May require more leaf pages in the Buffer Pool
    - This is an access path choice
      - Larger results are often more efficient to sort
      - Smaller results benefit by avoiding the sort
    - EXPLAIN
      - METHOD=3 row will not be present

Parallelism Improvements ...

- **Degrees of parallelism is determined at BIND / PREPARE**
- **May be adjusted during Host Variable / Parameter Marker substitution**
- **May be reduced / degraded at run time**
  - Buffer Pool negotiation
- **DB2 11 introduced System Negotiation**
  - LPAR storage usage may reduce / degrade parallelism
  - New DSNDQXST instrumentation fields

BIND / REBIND Changes ...

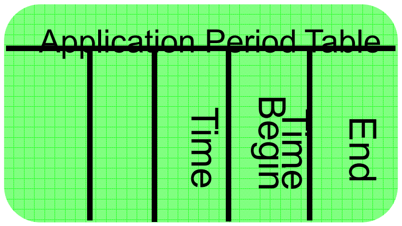
- **Packages**
  - APPLCOMPAT
  - APREUSE(WARN)
  - ARCHIVESENSITIVE
  - BUSTIMESENSITIVE
  - DBPROTOCOL(DRDACBF)
  - DESCSTAT
  - SYSTIMESENSITIVE
  - Support for mixed case characters when delimited
    - Collection-ID, DBRM-Name, DBRM-Library, Package-ID
- **Plan**
  - PROGAUTH

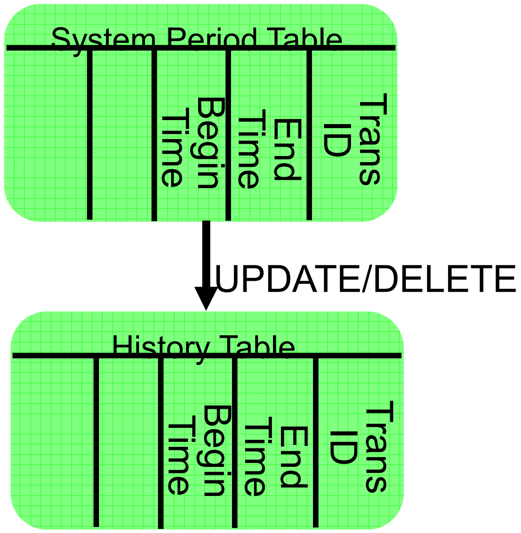
Temporal Update and Archive Transparency

- Temporal and Views
- Temporal Special Registers
- Archive Transparency

Temporal High Level Review

- **Business Time**
  - Begin & End business time columns
  - Set by the application
  - Modifications to SQL
  - Update/Delete modify periods
    - Could split rows to preserve Business Time
- **System Time**
  - Begin, End, and Trans time columns
  - Maintained by DB2
  - History Table for previous row versions
  - UPDATE/DELETE
    - DB2 populates the History Table
  - DML changes for SELECT only
    - Implicit UNION ALL to query History

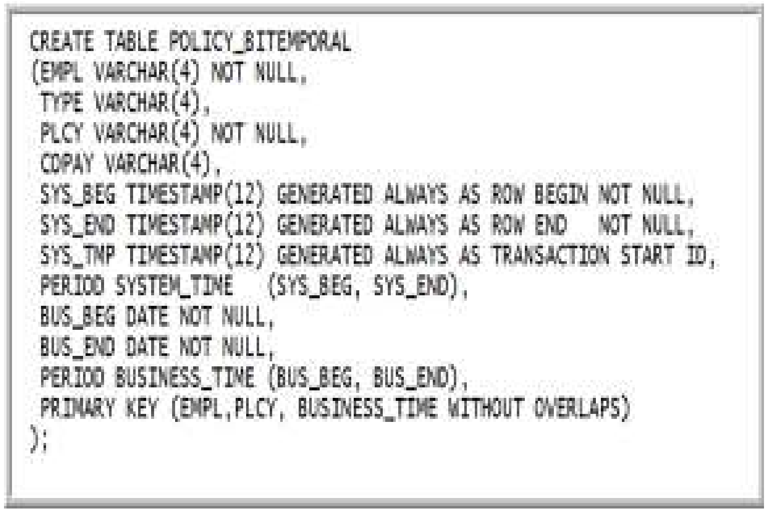




Versioning & Views …

- **DB2 11 - You can use temporal predicates when referring to a view**
- **DB2 10 & DB2 11 - You can not use temporal predicates in a view**

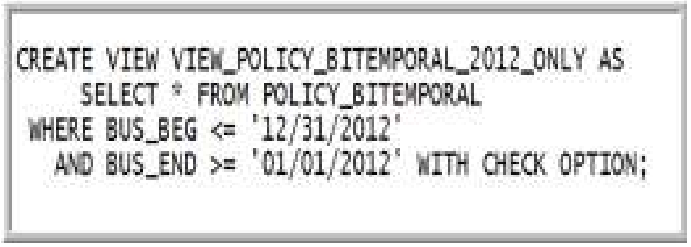
Base Table



```
CREATE TABLE POLICY_BITEMPORAL
(EMPL VARCHAR(4) NOT NULL,
 TYPE VARCHAR(4),
 PLCY VARCHAR(4) NOT NULL,
 COPAY VARCHAR(4),
 SYS_BEG TIMESTAMP(12) GENERATED ALWAYS AS ROW BEGIN NOT NULL,
 SYS_END TIMESTAMP(12) GENERATED ALWAYS AS ROW END   NOT NULL,
 SYS_TMP TIMESTAMP(12) GENERATED ALWAYS AS TRANSACTION START ID,
 PERIOD SYSTEM_TIME   (SYS_BEG, SYS_END),
 BUS_BEG DATE NOT NULL,
 BUS_END DATE NOT NULL,
 PERIOD BUSINESS_TIME (BUS_BEG, BUS_END),
 PRIMARY KEY (EMPL,PLCY, BUSINESS_TIME WITHOUT OVERLAPS)
);
```

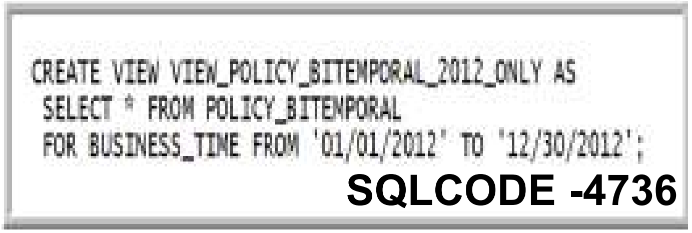
VIEW

✓



```
CREATE VIEW VIEW_POLICY_BITEMPORAL_2012_ONLY AS
     SELECT * FROM POLICY_BITEMPORAL
 WHERE BUS_BEG <= '12/31/2012'
   AND BUS_END >= '01/01/2012' WITH CHECK OPTION;
```

🚫

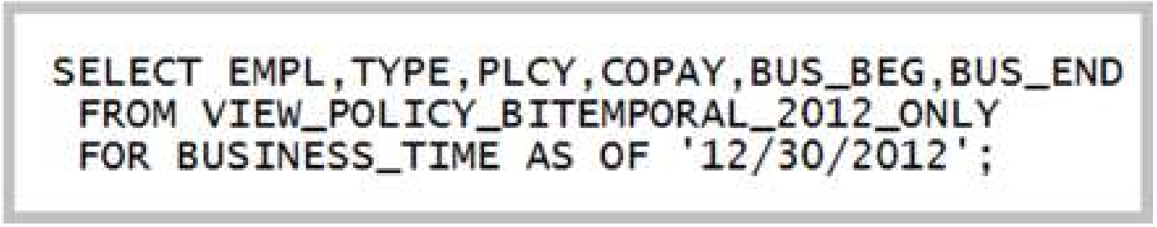


```
 CREATE VIEW VIEW_POLICY_BITEMPORAL_2012_ONLY AS
  SELECT * FROM POLICY_BITEMPORAL
  FOR BUSINESS_TIME FROM '01/01/2012' TO '12/30/2012';
```

SQLCODE -4736

- **Temporal predicates can now be used in DML on statements referencing views**

✓

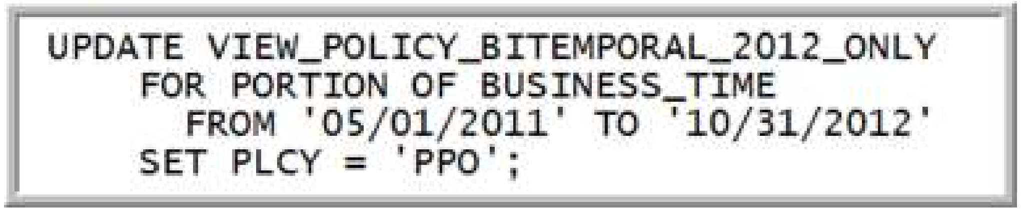


```
SELECT EMPL,TYPE,PLCY,COPAY,BUS_BEG,BUS_END
 FROM VIEW_POLICY_BITEMPORAL_2012_ONLY
 FOR BUSINESS_TIME AS OF '12/30/2012';
```

Versioning & Views Temporal Modifications ...

- **UPDATE or DELETE with the FOR PORTION OF clause can be applied to Views**

- **Temporal modifications can cause rows to be split**
  - – Rows that are created by splitting a row through a VIEW update may not be visible in the view after the update



```
UPDATE VIEW_POLICY_BITEMPORAL_2012_ONLY
    FOR PORTION OF BUSINESS_TIME
      FROM '05/01/2011' TO '10/31/2012'
    SET PLCY = 'PPO';
```

- **Symmetric Views are Views WITH CHECK OPTION**
  - – Temporal modifications are not constrained by the check option
  - – Split rows that disappear from the View definition are still allowed for a complete temporal modification

Temporal Special Registers …

- **Enable customers to be able to code applications using temporal data and to be able to test the system, possibly "without changing code"**
- **DB2 will be able to run the same query for different times by changing the special registers**
- **Have the ability to run AS OF any date by changing the special register**
- **Provides "Time Machine" capability**
  - Setting the temporal special registers to a specific point in time
  - Works for all subsequent SQL statements
  - Including those in invoked functions, stored procedures, and triggers
  - This allows the application to see data from a different point in time without modifying the SQL statements

Temporal Special Registers

- **CURRENT TEMPORAL SYSTEM_TIME**
  - TIMESTAMP(12), nullable
- **CURRENT TEMPORAL BUSINESS_TIME**
  - TIMESTAMP(12), nullable
- **SET Temporal Registers**
  - For DRDA the value of the special register is sent to remote side for implicit connect
    - When using a 3-part name
  - If you use the special registers, they continue to be used for that session until you turn them off by setting them to NULL
- **Setting the Special Register(s) generates implicit temporal qualifications**
- **Bind parameters determine if the Special Register will be honored when set**

Archive Transparency

- **What is the purpose of archiving?**
  - When you want to delete rows from the table, but need to keep the deleted rows for legal or business purposes
  - To move data to a cheaper storage medium
  - When you do not need to access the old data often, but need to be able to retrieve the data quickly
  - When you do not care about the lineage of a row
    - This means that you do not care about the changes to a row over time
- **Do we add extra columns for archiving like we do for system time tables?**
  - You do not need extra columns to enable Archive Transparency
- **Temporal and Archive Tables are mutually exclusive**
  - Can not build an Archive Table on a table that has either Business Time or System Time
- **Archive a large amount of data using REORG DISCARD to facilitate migration**
  - User would be responsible for loading data from the DISCARD file into the archive table

Archive Transparency Compared to System Time

	System Time	Archive Transparency
Schema -- two table approach	current table & history table same column #, column name, column attributes (data type, etc)	archive-enabled table & archive table same column #, column name, column attributes (data type, etc)
ROW BEGIN/ROW END/TRANS ID columns	mandatory	none
Period concept	yes – SYSTEM_TIME period	none, not compatible with system time
Compatible with Business Time tables	yes	no
Bind option	SYSTIMESENSITIVE	ARCHIVESENSITIVE
Implicit union all query transformation	controlled by CURRENT TEMPORAL SYSTEM_TIME special register	controlled by built-in global variable SYSIBMADM.GET_ARCHIVE
Data propagation to history/archive table	UPDATE and DELETE	DELETE SYSIBMADM.MOVE_TO_ARCHIVE
Implicit Static DMLs	Implicit two section design	Implicit two section design

Archive Transparency Management ...

- **LOAD RESUME can be used to archive data**
  - You can add your own data into the archive table
  - Use this when doing a REORG with DISCARD to load the DISCARD file into the ARCHIVE table

Archive Transparency Comparison

- **Archive Transparency**
  - Works one a single table
  - Deletes the entire row from the base table
  - Inserts the deleted row into a DB2 archive table
  - May not satisfy legal archival requirements
- **IBM InfoSphere Optim Data Growth Solution**
  - Works on business objects
  - Can delete selected rows (keep customer, delete orders) from the base table
  - Writes row to a non updateable extract file
  - Satisfies legal archival requirements

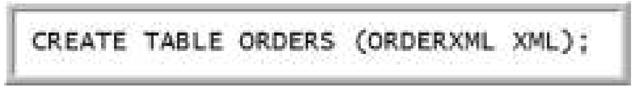
XML Enhancements

- **New Features**
  - Basic xQuery (retrofit to v10, PM47617, PM47618)
  - COBOL samples for XML (published on Developerworks website)
- **Features Enhancements**
  - Implicitly add doc node during insert/update
  - Crossloader support
  - Fix error reporting position predicate
  - Support xquery constructor as the source expression of insert and replace
- **Performance Enhancements**
  - Binary XML validation *(retrofit to DB2 V10)*
  - Partial validation after update
  - Date/Time Predicate Pushdown
  - XQuery(FLWOR) and XMLQUERY enhancement (Select From Where)
  - Optimize Index Search Keys
  - XML Operator Improvements, use less storage and CPU
  - XQuery deferred construction
  - XMLTABLE pushdown cast
  - Avoid validation of validated binary XML data during LOAD

XQUERY ...

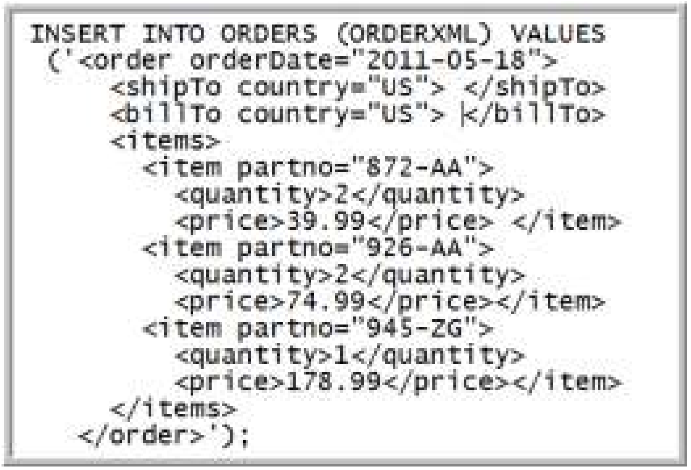
- **FLOWR expressions**
  - Retrofit into V10
  - Give developers more power to write XML applications in a familiar language

1) Create table



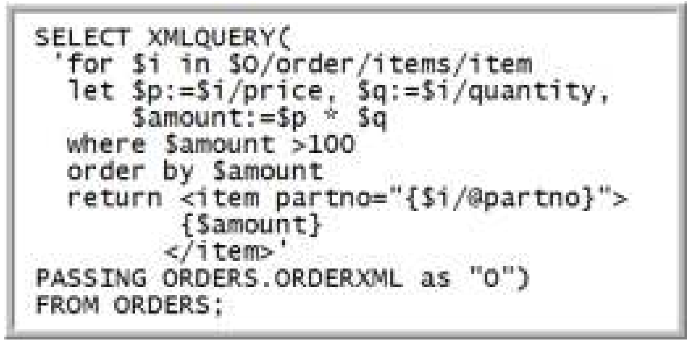
```
CREATE TABLE ORDERS (ORDERXML XML);
```

2) Insert XML document into table



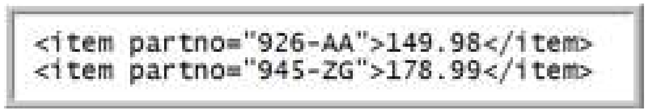
```
INSERT INTO ORDERS (ORDERXML) VALUES
 ('<order orderDate="2011-05-18">
     <shipTo country="US"> </shipTo>
     <billTo country="US"> |</billTo>
     <items>
       <item partno="872-AA">
         <quantity>2</quantity>
         <price>39.99</price> </item>
       <item partno="926-AA">
         <quantity>2</quantity>
         <price>74.99</price></item>
       <item partno="945-ZG">
         <quantity>1</quantity>
         <price>178.99</price></item>
     </items>
   </order>');
```

3) Select items, part numbers and amount for orders whose items are ordered in an amount greater than $100



```
SELECT XMLQUERY(
 'for $i in $O/order/items/item
  let $p:=$i/price, $q:=$i/quantity,
      $amount:=$p * $q
  where $amount >100
  order by $amount
  return <item partno="{$i/@partno}">
           {$amount}
         </item>'
PASSING ORDERS.ORDERXML as "O")
FROM ORDERS;
```

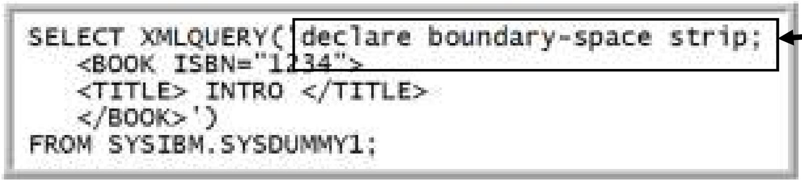
4) Output from FLOWR expression



```
<item partno="926-AA">149.98</item>
<item partno="945-ZG">178.99</item>
```

XQUERY Prolog ...

- **The *prolog* contains a series of optional declarations that define the processing environment for the expression**
  - Boundary space
    - Controls if whitespace is preserved or stripped in XQuery
  - Copy-namespaces declaration
    - Controls how namespace bindings are assigned when copying elements or nodes
  - Namespace declaration
    - Declares a namespace prefix and associates the prefix with a namespace URI
  - Default namespace declaration
    - Declares namespace to use for unprefixed qualified names



```
SELECT XMLQUERY('declare boundary-space strip;
   <BOOK ISBN="1234">
   <TITLE> INTRO </TITLE>
   </BOOK>')
FROM SYSIBM.SYSDUMMY1;
```

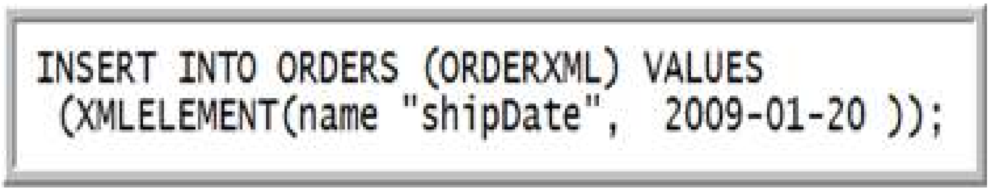
Prolog

Additional XQuery constructors ...

- **Document node constructor**
  - Construct the root node of an XML document .
  - Equivalent to the XMLDOCUMENT function, but can be used in XQuery expression
- **Comment node constructor**
  - Construct an XML comment node.
  - Equivalent to the XMLCOMMENT function, but can be used in XQuery expression
- **Processing Instruction constructor**
  - Construct an XML processing instruction (PI) node.
    - PIs are instructions for applications,
    - PI begins with a target used to identify the application to which the instruction is directed.
  - Equivalent to the XMLPI function, but can be used in XQuery expression
- **See the pureXML Guide section on XQuery constructors**

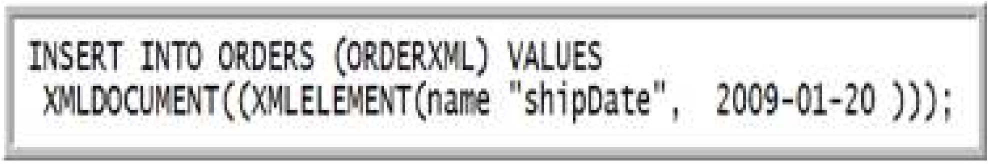
Implicit Document Node

- **Pre DB2 11 SQL requires an explicit XMLDOCUMENT function during insert/update.**
  - XML requires a “root node”. Pre DB2 11, you could not insert a document using XMLELEMENT, because it would not create the “root node”.
  - In V11, you can use XMLELEMENT to insert an XML document



```
INSERT INTO ORDERS (ORDERXML) VALUES
 (XMLELEMENT(name "shipDate",  2009-01-20 ));
```

SQLCODE -20345 in V10
SQLCODE of 0 in V11



```
INSERT INTO ORDERS (ORDERXML) VALUES
 XMLDOCUMENT((XMLELEMENT(name "shipDate",  2009-01-20 )));
```

Required V10 syntax

Crossloader support

- Tables with XML columns can now be the target of the Crossloader utility

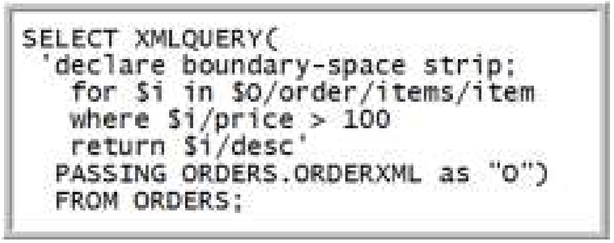
Validation Improvements Binary XML

- **PRE DB2 11, to validate binary XML, we needed to serialize the XML before we could validate**
- **IN DB2 11, we can validate binary XML in both LOAD and INSERT processing without serializing**
  - XML is always stored in parsed format
  - Eliminate the need to serialize before storing
  - Dependency on z/OS 1.13 PTF OA36712 or 1.12 PTF OA38633
- **Performance improvement**
  - INSERT into XML column with type modifier using binary XML
    - 30~40% CPU reduction
    - 15~18% CPU reduction vs.. string XML.
  - LOAD into XML column with type modifier using binary XML
    - 41% CPU reduction vs. DB2 10,
    - 18% improvement vs.. string XML

XML Query Rewrite

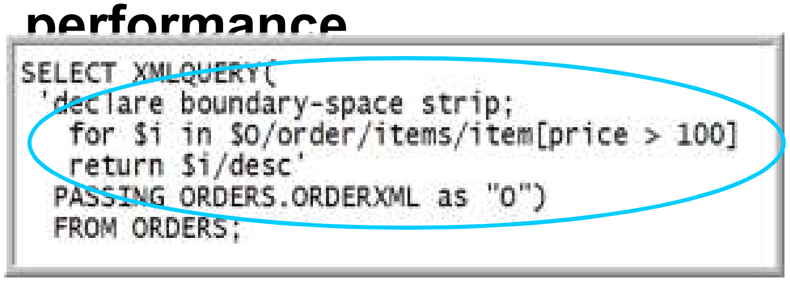
- Transforming the XML query into a better performing query

Original Query



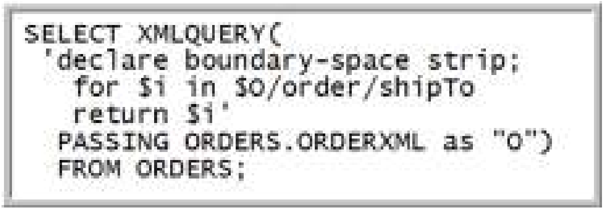
```
SELECT XMLQUERY(
 'declare boundary-space strip;
   for $i in $O/order/items/item
   where $i/price > 100
   return $i/desc'
  PASSING ORDERS.ORDERXML as "O")
  FROM ORDERS;
```

Price > 100 predicate is pushed down into the “for” clause for better performance



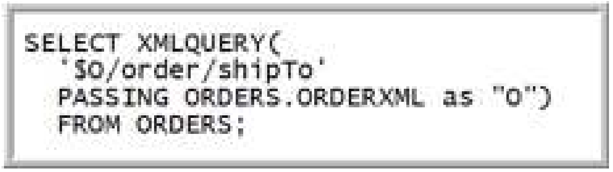
```
SELECT XMLQUERY(
 'declare boundary-space strip;
   for $i in $O/order/items/item[price > 100]
   return $i/desc'
  PASSING ORDERS.ORDERXML as "O")
  FROM ORDERS;
```

Original Query



```
SELECT XMLQUERY(
 'declare boundary-space strip;
   for $i in $O/order/shipTo
   return $i'
  PASSING ORDERS.ORDERXML as "O")
  FROM ORDERS;
```

Simple query is converted to XPath expression for better performance



```
SELECT XMLQUERY(
  '$O/order/shipTo'
  PASSING ORDERS.ORDERXML as "O")
  FROM ORDERS;
```

XMLTABLE Performance

- Remove unreferenced column definitions.
- Merge common column path expressions
- Date and Timestamp predicates pushdown
- Optimize index key range for Varchar predicates
- Process large number of small documents much more efficiently
- Push cast into column XPath

Index Enhancements

- Suppress-null indexes
- Auto Cleanup of Pseudo-deleted Index Entries

Suppress-null Indexes

- **Support creation of indexes where the default NULL values are excluded from the index**
  - Reduced index size, improved insert/update/delete performance, 3rd party DBMS compatibility
- **New CREATE INDEX keywords**
  - INCLUDE NULL KEYS (default)
    - DB2 will create an index entry even if every key column contains the NULL value
  - EXCLUDE NULL KEYS
    - DB2 will *not* create an index entry when *every* key column contains the NULL value. If any key column is not null the index entry will be indexed
- **Conditions:**
  - EXCLUDE NULL KEYS must not be specified if any of the columns identified by column-name are defined as NOT NULL
  - Not supported if the index is defined as a partitioning index

Auto Cleanup of Pseudo-deleted Index Entries

- **DB2 11 automatically cleans up pseudo deleted entries**
  - zIIP eligible and runs in the background (as enclave)
  - Designed to have minimal or no disruption to applications
  - New ZParm INDEX_CLEANUP_THREADS (0-128) to control number of concurrent cleanup tasks
    - Default is 10
    - 0 disables index cleanup
    - Value can vary between members of a data sharing group
    - Online changeable
  - New SYSIBM.SYSINDEXCLEANUP Catalog table to control auto cleanup at index level
    - Day of week/month, start/end time.
    - By default cleanup is enabled for all indexes
  - IFCID 377 written when index is cleaned up
- **-DIS THREAD TYPE(SYSTEM) to see threads**
  - Correlation id in the output shows: 014.IDAEMKxx
  - xx indicates thread number (01, 02…)

DB2 11 Security Enhancements

- **DB2/RACF authorization control enhancements**
  - – Invalidate cached authorization info when RACF changes are made. Also invalidate static SQL packages
    - New AUTHEXT_CACHEREFRESH Zparm to activate the new behavior
  - – AUTOBIND, BIND, REBIND present PKG-owner ACEE to RACF
  - – Dynamic SQL authorisation checking:
    - When DYNAMICRULES not equal to RUN, DB2 presents AUTHID to RACF
    - DYNAMICRULES defines whether AUTHID is PKG owner, ID that defined the routine, or ID that invokes the routine
  - – New AUTHEXIT_CHECK zparm to activate the new behavior
    - PRIMARY: DB2 provides the ACEE of the primary auth. ID
    - DB2: ACEE of package owner, DB2 honors DYNAMICRULES
- **Bind plan option to ensure the program is authorized to use the plan**
  - – New PROGAUTH bind option

RACF Synchronization of Security Caches ...

- **When**
  - z/OS 1.13 with OA39486, OA39487, OA39506
  - In CM
  - DSNX@XAC is active
  - AUTHEXIT_CACHEREFRESH=ALL (NONE is the default)
    - Not online updateable
  - ENF 79 requires Resource Class SIGNAL=YES in RACF Class Descriptor Table
    - When not using IBM supplied resource classes for DB2 objects & Admin Authorities
    - YES is default for IBM supplied RACF resources
    - Single and Multi-system scope class names supported
- **RACF sends notifications to DB2 about these changes**

Fine Grained Access Control – Mask Improvements

- **DB2 10 has Column Mask limitations / inconsistencies**
- **When the query contained a GROUP BY on the Masked column**
  - Various limitations documented in SQLCODE -20478
  - DB2 11 has removed these limitations
  - Required NFM / V11R1
- **When DISTINCT, UNION ALL, or UNION DISTINCT is used**
  - With an Aggregate Function, applies Column Mask first
    - If PM61099 is applied, this causes -20478 reason code 26
  - Without an Aggregate Function, removed duplicates first
  - DB2 11 removes the duplicates first in both cases
  - This was also implemented in DB2 10 maintenance
    - Therefore there is no DB2 11 incompatibility

Advanced Design Options

- Online Schema Evolution
- PIT Recovery with Pending ALTER
- DGTT Improvements
- Reserved Space for Update
- Compression Improvements

Online Schema Evolution...

- **Drop Column restrictions:**
  - A column cannot be dropped if
    - The containing table space is NOT a Universal Table Space
    - The table is a created global temporary table, a system-period temporal table, a history table, MQT or table referenced by MQT plus others....
    - Tables with EDITPROC or VALIDPROC
    - There are row permissions or column masks dependent on the table
    - There are triggers defined on the table
    - The column is an XML column, is part of the table partitioning key, RI
    - Hash Key column
    - DOCID / ROWID column
    - More restrictions apply, see SQL Reference ALTER TABLE
  - No immediate alters (e.g. Can‘t change data type, add column, add/rotate parts) combined with pending ALTER. Option remains to DROP PENDING CHANGES, then do immediate alters.
  - PM95294 is open for DROP COLUMN performance issues.

Online Schema Evolution…

▪ Online ALTER Partition Limit Keys

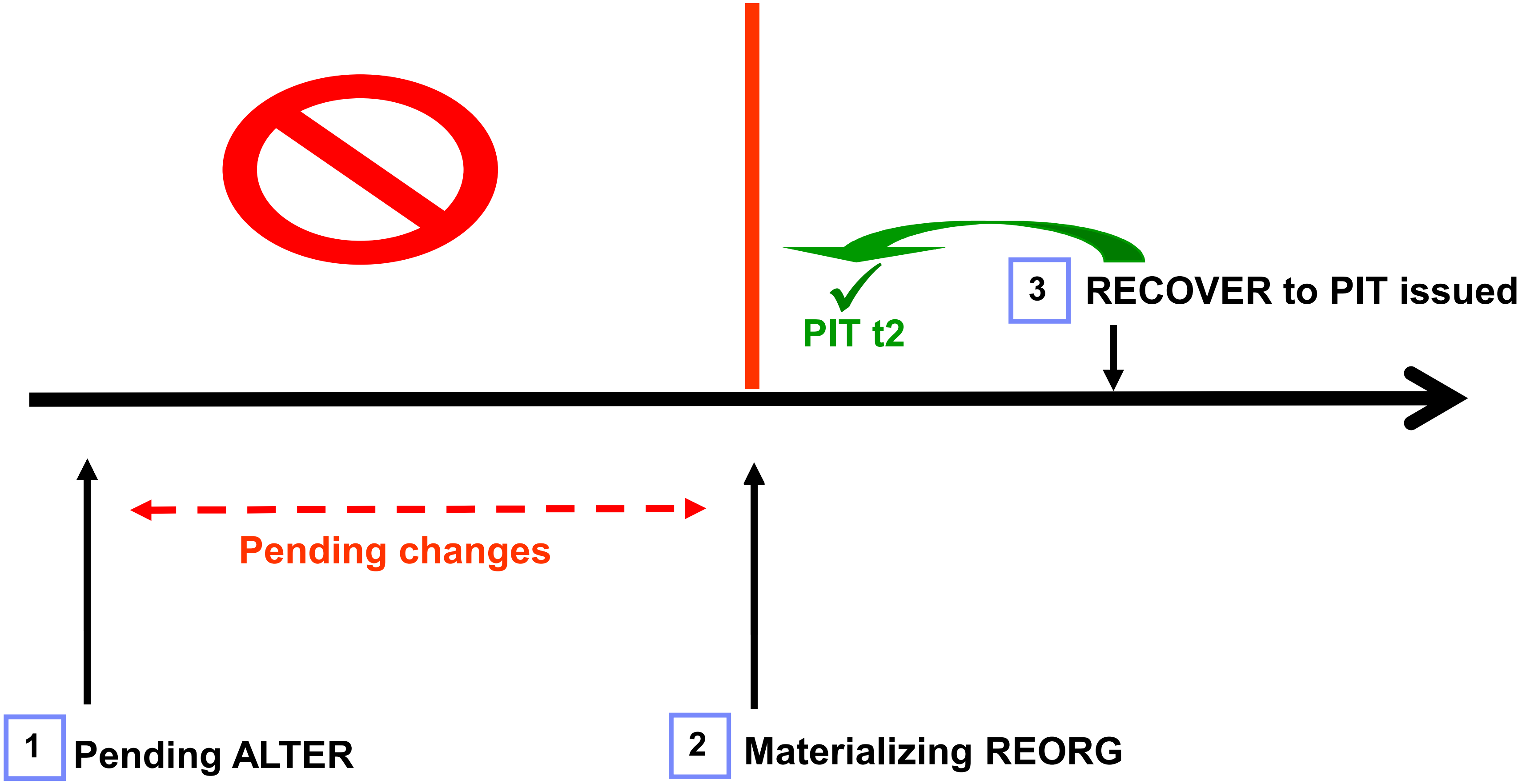
- DB2 11: Alter limit key is treated as a Pending Alter (NFM).
- The affected partitions are set to AREOR.
- Online REORG **must** be run to materialize the pending changes.
- Supported table spaces types are:
  - UTS – partitioned by range (PBR)
  - Classic partitioned table spaces (table controlled partitioning)
- The new limit keys are materialized in SYSTABLEPART in the SWITCH phase

Online Schema Evolution

- **ALTER TABLESPACE ... DROP PENDING CHANGES**
  - Pre DB2 11: AREOR setting is not removed.
  - DB2 11: AREOR setting is removed.
    - Exception: AREOR is still kept if transition to hash organization is in progress.

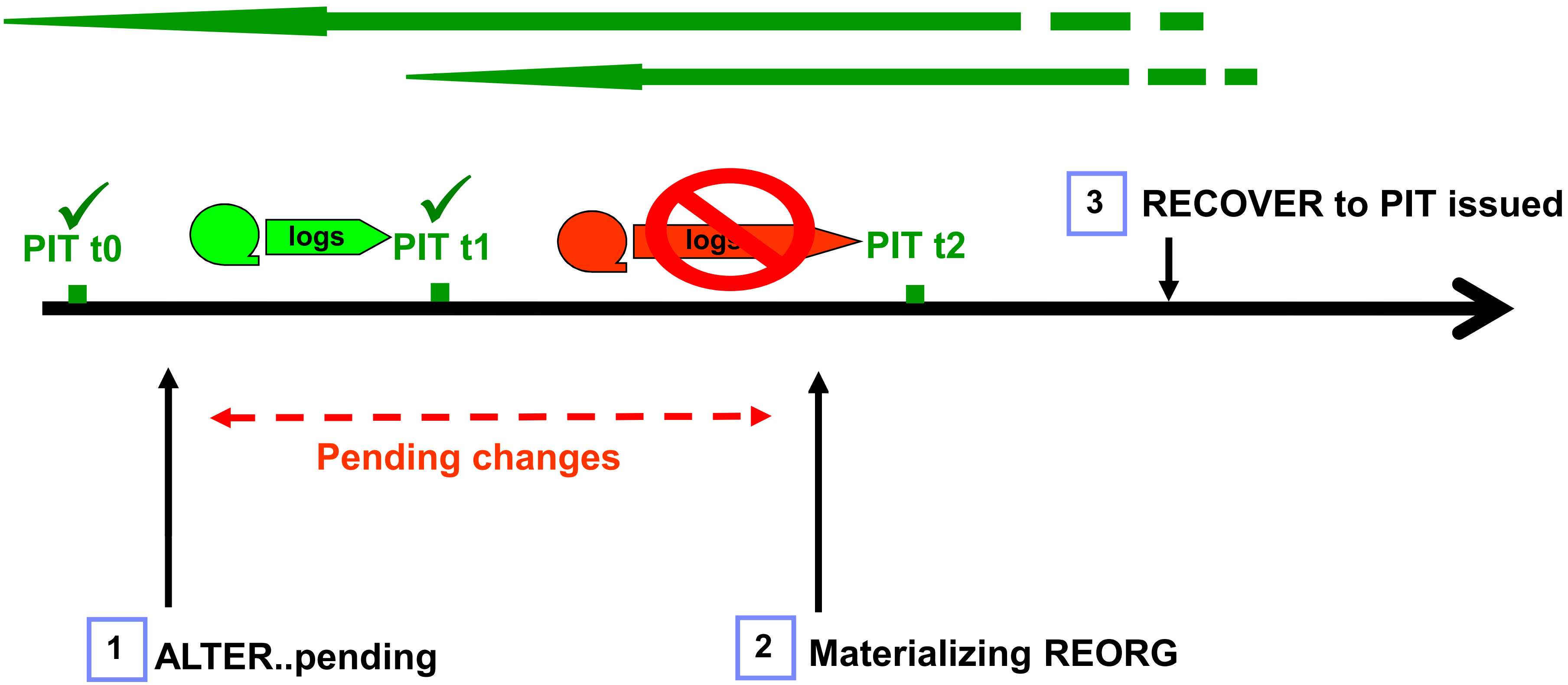
PIT Recovery with Pending ALTER …

- **Pre DB2 11: Restrict RECOVER across materializing REORGs**



PIT Recovery with Pending ALTER …

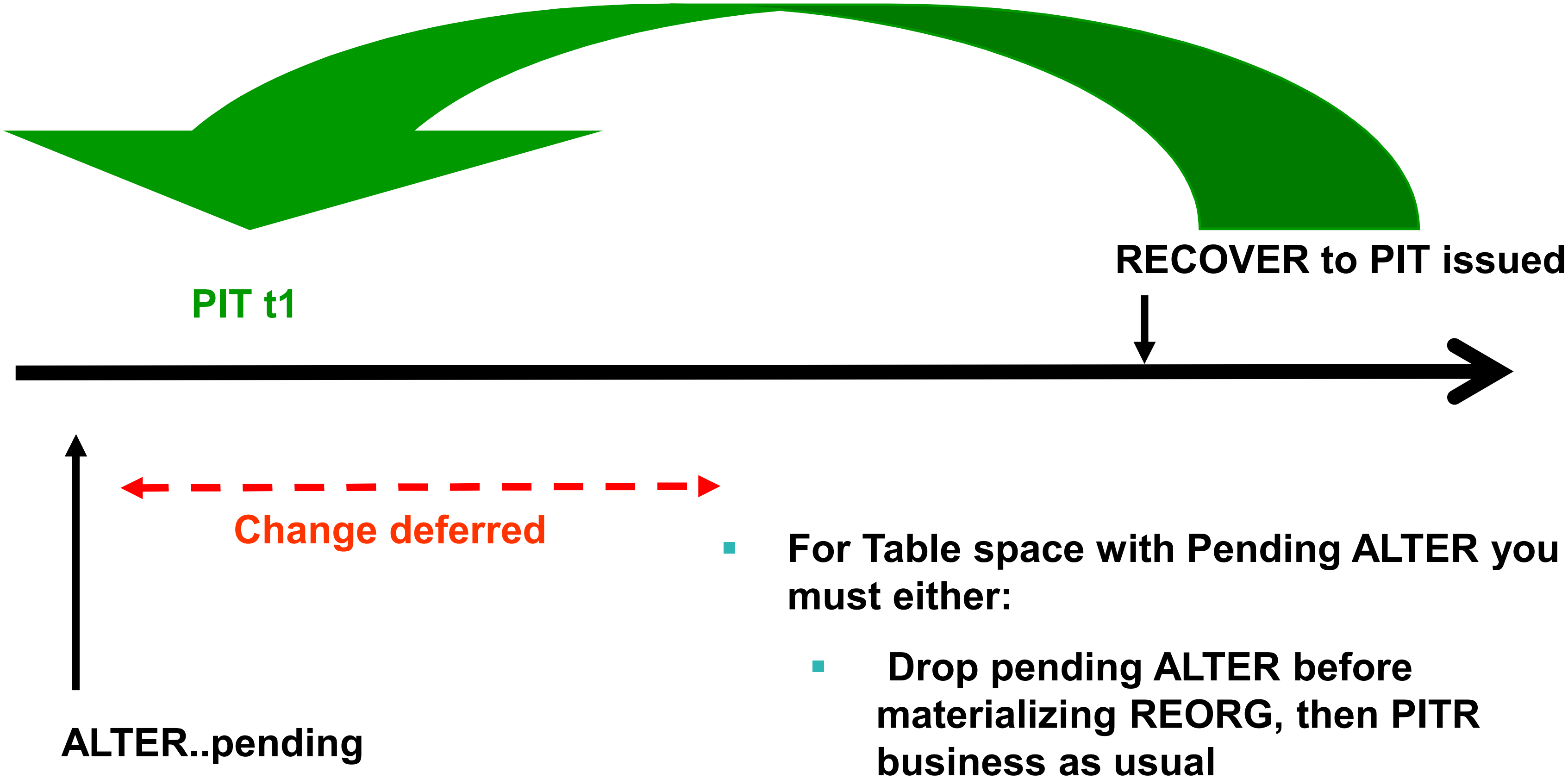
- DB2 11: Allow RECOVER across materializing REORGs



PIT Recovery with Pending ALTER…

- **The subsequent REORG is required**
  - Must on be on ENTIRE table space
  - SHRLEVEL NONE is not supported
  - SHRLEVEL CHANGE is *overridden* by SHRLEVEL REFERENCE
- **Before the subsequent REORG to materialize the schema**
  - No CREATE/ALTER/RENAME/DROP TABLE on the TS or AUX objects
  - Most other utility jobs fail: DSNU933I (REORG required)
- **REPAIR DBD and REPORT RECOVERY allowed**
- **Where there were pending changes on LOB table space:**
  - First REORG the LOB table space, then REORG the base table space

PIT Recovery with non-materialized Pending ALTER …



- For Table space with Pending ALTER you must either:
  - Drop pending ALTER before materializing REORG, then PITR business as usual

    or

  - Run materializing REORG and PITR restrictions apply

PIT Recovery with Pending ALTER

Supported ALTERs:

Attribute	LOB Table space	PBR UTS	XML Table space
SEGSIZE	**N/A**	✓	✓
DSSIZE	✓	✓	✓
Table space page size	✓	✓	**N/A**
MEMBER CLUSTER	**N/A**	✓	**N/A**
Partitioning Limit Key	**N/A**	✓	

*** Not all pending ALTERs supported**

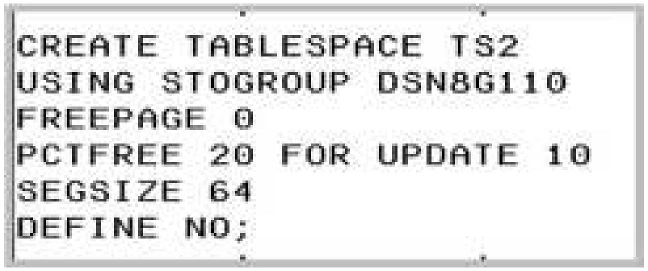
DGTT Improvements …

- **DB2 11: support for NOT LOGGED syntax for DGTTs**
  - Reduce logging overhead/volumes
  - Less CPU from log processing
  - Faster rollback/error processing after large insert into DGTTs
- **DB2 11: hold DGTT-based statements across COMMIT**
  - Reduce incremental binds and full prepares
  - Extended to non-cursor SQL statements
  - RELEASE (DEALLOCATE) required

DGTT Improvements…

- **Support for NOT LOGGED DGTTs at table level**
  - – LOGGED is default
  - – NOT LOGGED has two options for recovery
    - ON ROLLBACK DELETE ROWS is default
      - – If DGTT updated since last COMMIT, all rows in DGTT will be deleted.
    - ON ROLLBACK PRESERVE ROWS
      - – All rows preserved; DGTT and rows available, Cursors opened on DGTT have no position
  - – ROLLBACK TO SAVEPOINT behaves as ROLLBACK, except to the SAVEPOINT
  - – SQL error against a DGTT results in deleting rows from that DGTT regardless of ROLLBACK setting
    - Other tables (including other DGTTs) are not impacted)
  - – Application programmers must take ROLLBACK behavior into consideration

Reserved Space for Update



```
CREATE TABLESPACE TS2
USING STOGROUP DSN8G110
FREEPAGE 0
PCTFREE 20 FOR UPDATE 10
SEGSIZE 64
DEFINE NO;
```

- **DB2 11:**
  - – INSERT will reserve space for updaters
  - – New DDL keyword for CREATE TABLESPACE on PCTFREE clause:
    - FOR UPDATE *n*
      - – Where *n* is % of free space to leave in each data page by INSERT, LOAD or REORG
  - – New ZParm: PCTFREE_UPD
    - System default for FOR UPDATE value when it is not specified
    - If not specified, same behavior as DB2 10

Compression Improvements

- **Decompression can be the single most expensive instruction**
  - Executed during evaluating and retrieving records from compressed table space
  - Higher impact with simple queries scanning large number of row
- **DB2 11 Optimization in decompression process**
  - Software based decompression on top of today's H/W compression
  - New decompression routine provides:
  - Column level decompression
  - Special optimization to speed up the decompression
  - Efficient processor cache utilization when handling dictionary
  - No change on compression ratio
  - Only for data compression, not for index page
  - Compatible with existing compression, no user action necessary
- **When the compression dictionary is recomputed**
  - Old dictionary copied to the log for log reading tools

Utilities Enhancements

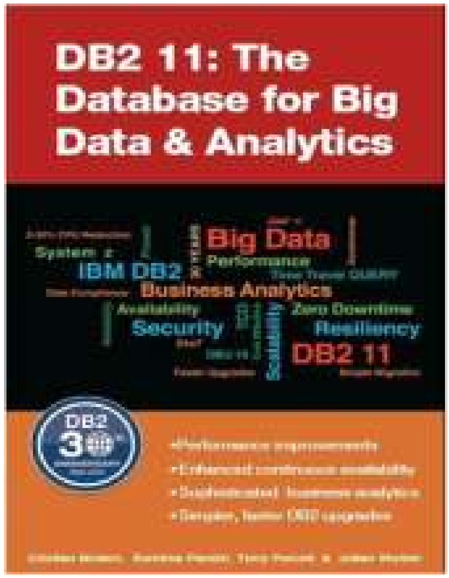
- **Compression Dictionary**
- **REORG**
  - NPSIs with Part Level REORG
  - Switch Phase
  - Delete Empty PBG Parts
  - Automated Mapping Tables
  - No Sort Options
  - Inline Copies
  - LISTDEF
  - REBALANCE
  - Serviceability
- **Statistics**
  - zIIP redirect
  - Inline enhancements
  - ACCESS DB Externalization
  - RESET ACCESSPATH
- **Backup & Recovery**
- **LOAD**
- **Parallelism**
- **General Enhancements**
- **Deprecated Items**

Summary of DB2 11 Business Value

- Application Compatibility
- Business Analytics
- Big Data & JSON
- Global Variables, Arrays, & Autonomous Transactions
- Optimization Feedback
- More access path stabilization
- Transparent Archive
- Suppress Null Indexes
- More/Improved Pending Alters
- DGTT Improvements
- Distributed performance
- Expanded RBA / LRSN
- High Availability Performance and Resiliency
- High levels of hardware exploitation
- More 64 bit storage & processing
- Compression Improvements
- Work file management
- Client information instrumentation
- Automated Index Clean Up
- RACF Integration

DB2 11 Resources

- **IBM Information Center / Knowledge Center**
- **DB2 11 Technical Overview Redbook (SG24-8180)**
- **DB2 11 links: https://www.ibm.com/software/data/db2/zos/family/db211/**
  - Links to DB2 11 Announcement Letter, webcasts and customer case studies
  - Whitepaper: “DB2 11 for z/OS: Unmatched Efficiency for Big Data and Analytics”
  - Whitepaper: “How DB2 11 for z/OS Can Help Reduce Total Cost of Ownership”
- **DB2 11 Migration Planning Workshop**
  - http://ibm.co/IIJxw8
- **Free eBook available for download**
  - http://ibm.co/160vQgM
- **SAP and DB2 11:**
  - “DB2 11 for SAP Mission Critical Solutions”: http://scn.sap.com/docs/DOC-50807
  - “DB2 11 with SAP Performance Report”: http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102394#!



Thank
You