Db2 12
12 months agile production experience
with more focus on the Data and less on the base

Roy Boxwell
SOFTWARE ENGINEERING

Session code: < V11 >
Wednesday, May 2 - 10:30 – 11:30 a.m.

Db2 z/OS

Agenda

- Db2 migration and maintenance – meet the agile and continuous delivery face of Db2
- Db2 Maintenance, Catalog, Function and Application Levels – differences, dependencies and how to successfully manage them
- Running Db2 12 and exploiting new features – what's hot and how to exploit it best?
- Emphasizing the Data – what has changed since July and what are the hybrid aspects managing Data?
- Top 10 list of tips and gotchas from a year of Db2 12 production Data bases around the world

Db2 12 migration

- Db2 11 prereqs for migration to Db2 12
  - Ensure catalog consistency (REPAIR DBD TEST/DIAGNOSE + CHECK DATA/LOB/INDEX + DSNTESQ + ….)
  - Run pre-migration check queries DSNTIJPM (V12) or DSNTIJPC (PI58254)
  - Apply fallback SPE PTF to all DS members (PI33871/II14794)
  - Make sure Db2 11 PTF level is current and all migration related maintenance is applied – Use SMP/E Fix categories
    - IBM.Migrate-Fallback.DB2.V12 and
    - IBM.Coexistence.DB2.SYSPLEXDataSharing
- Minimum OS/hardware level: z/OS V2R1, z196/z114
- Watch out for deprecated objects (simple TS)

Db2 12 migration

- RACF changes
- Db2 now utilizes TCP/IP DROP API through the EZBNMIFR callable service (NMI)
  - requires RACF security profiles to be defined:
    - RACF PERMIT ACCESS(CONTROL) on MVS.VARY.TCPIP.DROP(OPERCMDS) for <userid> for <xxxx>DIST
- HVSHARE should be 510 TB (default)
  - Db2 12 requires 1 TB of 64-bit shared private storage in z/OS (virtual)
    - Use IFCIDs 217 and 225 to check

Db2 12 migration

- Plan for memory increase (real, not virtual)
  - larger sized memory delivers performance improvements
  - typically +~ 10%
  - up to +30% when exploiting new in-memory function
    - e.g. Fast Traverse Block area +20% on allocated VPSIZE
  - Consider increase of zIIP utilization
    - Db2 12 offloads more processing to zIIP
      - SQL query parallelism
      - REORG/LOAD RELOAD phase

Db2 12 migration

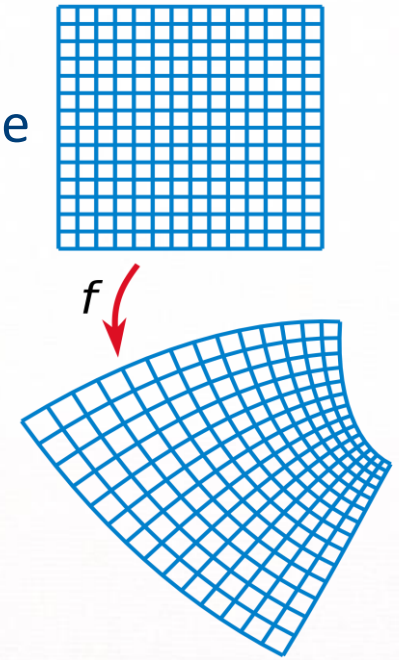
- Db2 Connect – any driver level works as is with V12R1M100 - V12R1M5++
- For V12R1M501, data server clients and drivers must be at the following levels:
  - JDBC and SQLJ: V3.72 and V4.22, or above
  - Other IBM data server clients and drivers: Db2 V11.1 fix pack 1, or later
    - clientApplCompat controls the capability of the client), e.g.
      - when new behavior is introduced, not handled by application
      - to ensure compatibility with new behavior
- clientApplCompat V12R1M500 is needed to exploit capabilities shipped after GA at function levels beyond Db2 V12R1M500

Db2 12 migration

- Utility suite installation requires registration in SYS1.PARMLIB(IFAPRDxx)
  - CBPDO introduced and SystemPac is the direction
  - Change seperate products using only F or J FMIDs to use an E or H base FMID
- From the Db2 Utilities Suite program directory:
  PRODUCT OWNER('IBM CORP') NAME('DB2 UTIL SUITE') ID('577-AF4')
  VERSION(12) RELEASE(1) MOD() FEATURENAME('V12R1') STATE(ENABLED)
- An unregistered Utilities Suite results in utility error (DSNU3333I)

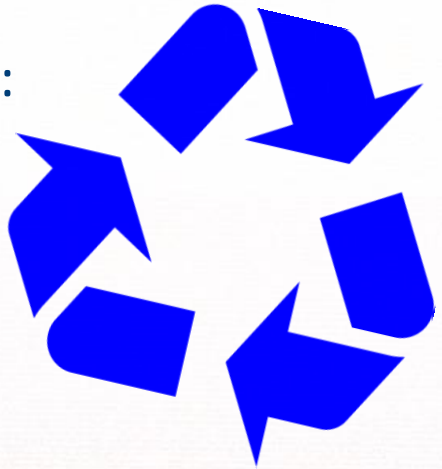
Db2 12 migration

- REORG MAPPING TABLE NRID length extended (7 byte RIDs)
  - No fallback toleration logic in V11
    - V11 REORG running with the V12 mapping table will fail
  - V12R1M100 REORG tolerates V11 mapping table
  - V12R1M100 and V12R1M5xx REORG supports V12 mapping table
- Basic Row Format (BRF) deprecated
  - Pagesets in BRF will continue to be supported
  - zparm SPRMRRF will be removed in V12
  - REORG/LOAD REPLACE converts BRF to RRF
  - New objects created with RRF only
  - ROWFORMAT option in REORG removed from the documentation, but still supported



Db2 12 migration

- Invalidation of prepared SQL statements in DSC
  - RUNSTATS always flushed statements from DSC before V12
  - V12 RUNSTATS won't flush statements from DSC,except
    - INVALIDATECACHE YES was specified
    - RUNSTATS after CREATE/DROP INDEX and statistics profile updated
    - RUNSTATS UPDATE(NONE) REPORT(NO)
  - Other utilities flush the DSC, if an exception state exists, e.g.:
    - REBUILD PENDING
    - REORG PENDING



Db2 12 migration:

- Avoid automatic Binds - identify plans/packages in advance
- Keep your plans granular - automatic Binds of plans can be particularly disruptive for a few, or single large plan.
- Free inactive package copies and Bind with PLANMGMT(EXTENDED) – assures your current copy in the original slot, ready for REBIND SWITCH.
- Upgrade EXPLAIN tables
- Set ABIND=COEXIST in DS prevents the "autobind ping-pong"*
- Apply PTFs for APARs PI69589 (V11) & PI69584 (V12) reduces catalog contention during migration
- Identify packages that were last bound earlier than V10

*fixed by APAR PI87675

Db2 12 maintenance best practices:

- When fallback occurs in non-data sharing
  1. rebind any package that was bound on the up-level
  2. rebind any packages or plans that were automatically bound after fallback
  - without explicit rebinds, automatic bind occurs for those packages and plans when going to the up-level release again.
- Mass Bind/Rebind on the new level only after completion of all members - no concern of down-level members automatic rebinds in coexistence

Db2 levels

- Starting with Db2 12 GA (V12R1M500), IBM is going forward to the CD model
- Only single phase migrations, no more CM, ENFM, NFM
- Enhancements may or may not require CATMAINT
- A new level is identified by
  - Version ID
  - Release number
  - Modification identifier
- New Enhancements and capabilities delivered as part of the maintenance stream as Function levels
- Function levels enabled via –ACTIVATE command
- Db2 Function Levels may be skipped, but always include all new/changed features/behavior of prior Function levels

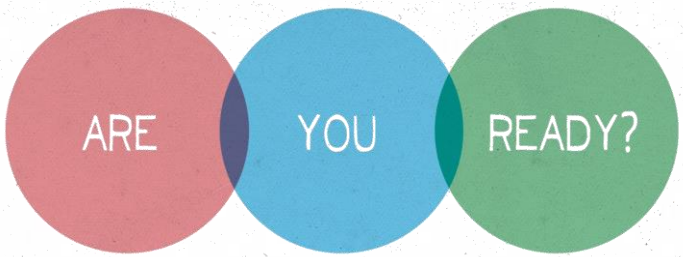
Db2 levels

- Db2 customers will see a significantly high volume of CD items in a single maintenance stream
- Entirely new releases, or versions will be very rare and only for major changes, like
  - Extending control structures
  - Enabling an architecture level set
  - Adopting a new compiler

- Recommended Service Upgrades (RSUs) are still available on a monthly/quarterly interval and may, or may not include new features
- Although a code drop may include fixes as well as new/changed features and functions, Db2 users keep control over what they activate

Db2 maintenance recommendations

- Regularly apply maintenance as needed
  - preventative service
  - corrective fixes
- Stabilize the Maintenance level for a month
  - Monitor and keep the system stable before moving on
- Keep an eye on HOLDDATA and check specifically for HIPERs and PEs
  - Apply where needed
- Cultivate the approach and avoid applying and activating too many changes at once
- Prepare for fallback – exploit Function/Application levels
- Prepare for adding critical APAR fixes



Db2 maintenance recommendations

- If your shop can‘t accepted outages for hours/days…
  - Make sure you consider
    - Time to detect anomalies
    - Time to analyze effect and origin
    - Time to evaluate a forward and a backward strategy
    - Time to fix/recover

…. you gotta TEST, TEST, TEST

- Carefully
- Thoroughly
- Rigorously

Db2 maintenance recommendations

- Regression testing is **the** critical piece to keep up production
  - Include critical as well as custom processes
  - Scale up testing to cover workloads peaks
  - Build a (performance) baseline for comparison and trending
  - Only move on after tests completed to your satisfaction
    - Postpone maintenance as opposed to forcing in
  - Practice fallback scenarios for individual Applications as well as the system
  - Make fallback strategies part of the (pre-production) testing and practice them
- Always assure yourself a back door or Plan B
- Minimize change and use of new function

Db2 maintenance recommendations

To minimize change and impact, applying maintenance now consists of up to four different, hierarchical levels:

1. Db2 Maintenance (aka. Code) level – applies new/fixed code
   - Db2 libraries via SMP/E. “Most new features are disabled”
   - Contains fixes and enhancements
2. *Db2 Catalog level – updates the Db2 Catalog via CATMAINT
   - Updates the Db2 Catalog if needed by a new FL
3. *Db2 Function level – controls functions generally available
   - via Db2 command (-ACTIVATE FUNCTION LEVEL)
4. *Application level – controls functions per application/query
   - via Db2 BIND/REBIND

*modifications are accumulated when skipping levels

Db2 maintenance recommendations

- Available levels can be set in multiple ways
- Exploiting new levels granularly minimizes impact and testing needs:
  - Bind/Rebind (default via ZPARM: SQLLEVEL, APPLCOMPAT, or BIND with APPLCOMPAT VvvRrMmmm)
  - SQLLEVEL – DSNHDECP for precompile or coprocessor (no more NEWFUN)
  - SET CURRENT APPLCOMPAT may be used for prior levels
  - Client property – clientApplCompat may be used for prior levels

Activation of levels

1. SMP/E apply of PTFs, or RSUs may increase the Db2 Maintenance level – on the member level of a DS group
   → Verify the system to be stable
2. CATMAINT (if needed) increases Db2 Catalog level
   → Maintenance level fallback is not possible after CATMAINT
3. ACTIVATE FUNCTION LEVEL (if needed/available) increases the Db2 Function level – on the group level of a DS group
   → New functions/behavior that aren't related to SQL DML, DDL and DCL available
   → REBINDs pick up Db2 Optimizer enhancements/changes
   → Dynamic SQL (non-stabilized) affected accordingly
4. BIND/REBIND APPLCOMPAT and/or ZPARM APPLCOMPAT (if needed/available) increases the Db2 Function level of an individual package
   → New functions/behavior related to SQL DML, DDL and DCL available
   → SET CURRENT APPLCOMPAT may be used for prior levels
   → clientApplCompat may be used for prior levels

Deactivation of levels

- Db2 Maintenance level fallback is only possible when no CATMAINT has been executed and the Function level is still supported
- Db2 Catalog level fallback is impossible
- FUNCTION LEVEL fallback is possible via ACTIVATE FUNCTION LEVEL, but
  - the Db2 Maintenance level has to remain on the highest level

- BIND/REBIND APPLCOMPAT and/or ZPARM APPLCOMPAT can be used for Db2 Function level fallback of an individual package
  → old functions/behavior related to SQL DML, DDL and DCL
- SET CURRENT APPLCOMPAT may be used for prior levels
- clientApplCompat may be used for prior levels
- Unless you REBIND, packages bound with higher Function levels remain functional…

Db2 12 maintenance best practices:

Db2 comes with the right capabilities to protect your applications

- Use PLAN MANAGEMENT to fallback to prior runtime structures
- Stabilized dynamic queries to freeze dynamic runtime structures
- Use Db2 collections to REBIND (distributed) packages into a separate collection to compare and switch
- Monitor and be aware of applications being affected by changed behavior
- Always have IFCID 376 look for potential incompatibilities (Incompatibility Change Indicator – ICI)

Db2 12 maintenance best practices:

You might need (your existing) tools to be able to

- Test any level, or combination of levels flexibly and reliably
- Speed up and fully automate testing to keep up with code drops being *continuously* released
- Compare to a selection of individual Workload sets to easily detect anomalies

→ A cloning tool is a good choice to speed up and fully automate testing for an entire environment

→ A capturing, or monitoring tool, with Workload Replay capabilities is a good choice for including and automating application tests

More focus on the Data and less on the base

The **data**base focus is about making the most of the data by

- making it easy to deploy it wherever it is needed
  - Db2 12 can handle up to 360 million transactions per hour through a RESTful web API
- fluidly adapting to changing needs
  - Db2 12 flexibly allows adding new functions via CD
- allowing for integration of multiple platforms, languages and workloads
  - Db2 12 is intended to be hybrid, supporting various types of workload

→ “data without limits”

Hybrid aspects of managing data – need more?

To help clients easily identify how IBM enables these principles, “Db2” will no longer represent a single product. Instead it will represent an entire hybrid data management family designed to span transaction databases and facilitate data warehousing that helps clients achieve “data without limits”.

Details about the new Db2 family naming – including the “Db2 Warehouse” offerings formerly known as “dashDB” – can be found in IBMs latest announcements.

Hybrid aspects of managing data

- While Db2 proved itself for ages, being the premiere transactional database server, analytics processing didn't always perform to our full satisfaction. Additionally the complexity of transactional systems is increasing.

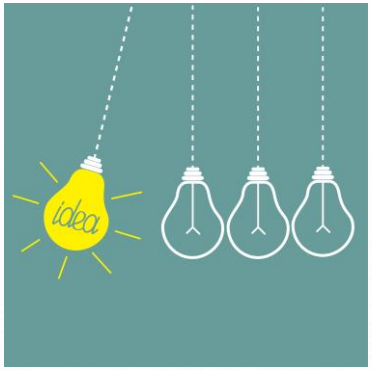
  Key for a premiere Hybrid Transactional/Analytical Processing (HTAP) database is performance.

- IDAA and Spark turn out to be the right components to boost these rapidly evolving systems, besides major Db2 12 enhancements in these categories:
  - Application enablement
  - Database administrator (DBA) function
  - OLTP performance
  - Query performance

Hybrid aspects of managing data

- Application enablement
  - expand the use of the existing features of Db2
  - mobile, hybrid cloud, and DevOps enablement
  - provide enhance IDAA functionality
  - provide incremental improvements in SQL and SQL/PL

→ make Db2 ready for the next wave of applications



Application enablement

- Enhanced SQL MERGE
- SQL pagination syntax LIMIT/OFFSET
- Lifting partition size limit to 1TB (UTS PBR)
  - Requires Relative Page Numbering (RPN) – no way back!
    - results in AREO
    - REORG needs DRAIN on the entire TS
      - Needs extended mapping table (7 byte RID)
- LOB compression – (actually done by hardware)
- DRDA fast LOAD

Hybrid aspects of managing data

- Database administrator (DBA) function
  - relieve object (scalability) limits beyond 128TB
  - simplify large table management
  - remove the biggest inhibitors to 24 × 7 continuous availability
  - provide incremental security and compliance improvements

→ free Db2 from sizing and availability constrains

Database administrator (DBA) function

- In-Memory continuous BP
- Granular global commit LSN and read LSN
- Online ALTER to increase DSSIZE
- Insert partition (UTS PBR)
  - space/free space taken from last partition
  - results in AREO
  - REORG needs DRAIN on the entire TS
  - package/DSC invalidated
- LOB compression – (actually done by hardware)
- REORG/LOAD use of STATS profiles

Hybrid aspects of managing data

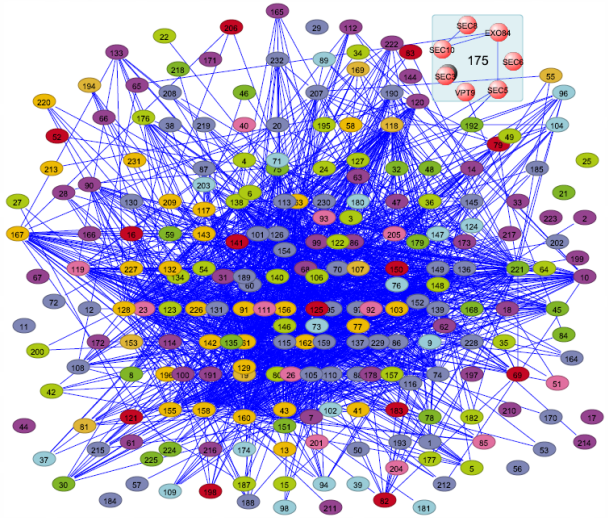
- OLTP performance
  - deliver performance improvements built on Db2 10 and 11
  - reduce CPU consumption in the range of 5%–10% by exploiting in-memory features
  - double the throughput when inserting into a non-clustered table
  - remove system scaling bottlenecks associated with high n-way systems
  - provide incremental improvements related to serviceability and availability

→ substantially improve Db2s OLTP strengths

OLTP performance

- Fast Index Traversal (FTB)
- Fast unclustered insert
- Granular global commit LSN and read LSN
- Asynch CF lock duplexing

Hybrid aspects of managing data



- Query performance
  - 20%–30% CPU reduction for complex query workloads compared to Db2 11
  - improve efficiency
  - reduce other resource consumption
    - e.g. 80% UNION ALL performance improvement
  - simplify access path management, especially for dynamic SQL

→ expand query performance of complex workload (BI, OLAP)

Query performance

- Dynamic plan stability
- Asynch CF lock duplexing

Db2 12 – what's hot?

- More use of list prefetch – LPO?
  - Db2 12 uses more list prefetch (and potentially hybrid join), but not necessarily changes in the access plan where Db2 would previously choose a sort avoidance plan
  - The Optimizer cost model is enhanced in 12 to more closely reflect the true cost (and benefit) of list prefetch
  - Trying to be careful not to select list prefetch (with sort) as an access path when there was an alternative access path that could use an index to avoid a sort i.e., for pagination type SQL

LPO: List Prefetch Optimizer is a feature of DS8000; R6.2

Db2 12 – what‘s hot?

- Financial Services customer:
  - „In some systems we have 90+ times 4GB logs filled per day“
    - Backup/Recovery strategy is struggeling
    - Risk of violating RTOs

Db2 12 solution → LARGER LOGs

  - Active LOGs defined with 8GB size
    - Use the same LOG size in a data sharing environment
    - Make sure your log analysis tool supports larger LOGs

Db2 12 – what‘s hot?

- Banking customer:
  - „In older systems 30% of objects have primary auth id as owner“
    - Thousands of ownerships are very confusing
    - This is considered as a security/compliance weakness

Db2 12 solution → TRANSFER OWNERSHIP

  - SECADM set via ZPARM
  - Approximately 7,000 ownerships transferred

Db2 12 – what's hot?

- Insurance customer:
  - „We've defined several RUNSTATS profiles not being honored by INLINE STATS"
    - REORG/LOAD with INLINE STATS compromises our statistics
    - Extra RUNSTATS takes additional time and burns resources

    Db2 12 solution → RUNSTATS PROFILES used INLINE

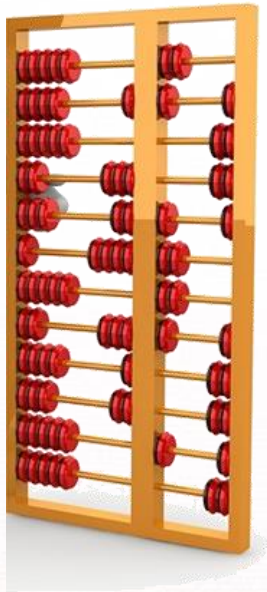
  - USE PROFILE option added to the REORG TS STATS specifications

Db2 12 – what's hot?

- Insurance customer:
  - „Defining extended stats collection is complex and sometimes difficult to determine“
    - Bad setup slows down RUNSTATS and collects too many/few information

    Db2 12 solution → RUNSTATS FREQVAL without COUNT

  - RUNSTATS automatically calculates the right value (Terry Purcell says it has a cap of 100)



Db2 12 – what's hot?

- Retail customer:
  - „Resource consumption and operational costs are always an issue"
    - Growing workload and shrinking margins are challenging our business.

Db2 12 solution → less resource consumption

&

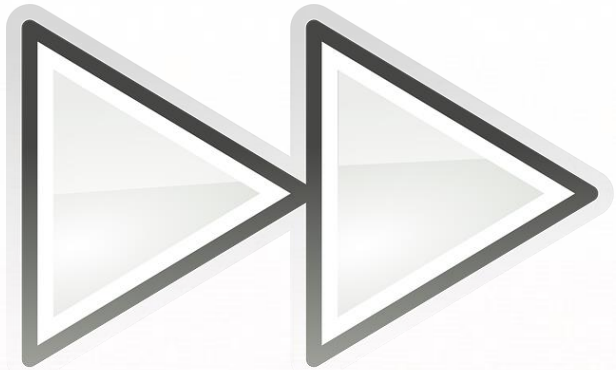
  - Reduced overall CPU consumption was observed
  - Accounting shows less CPU consumption in many cases

Db2 12 – what's hot?

- Insurance Customer:
  - „Some static scrollable cursors of our applications burn an incredible amount of resources compared to the rows actually being processed"
    - Cursors are used to scroll to a certain row wasting resources for materialization and recoverability

    Db2 12 solution → SQL Pagination

  - Queries need to be modified, but
    - No more materialization
    - No long running URs created by read-only applications

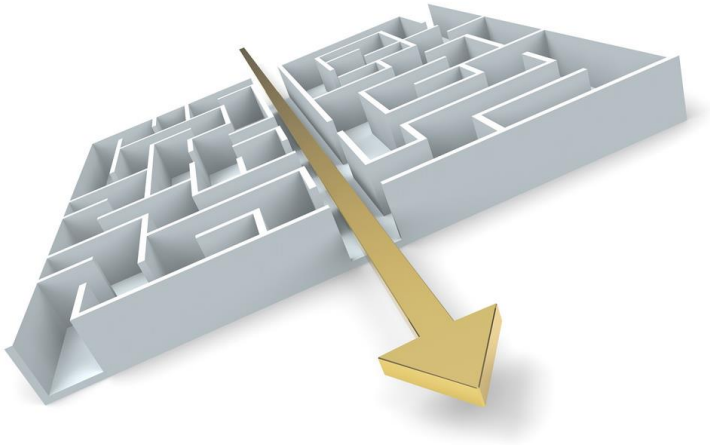


Db2 12 – what‘s hot?

- SQL developers:
  - „Horizontal recursive SQL is OK, but is complex to write and to understand”
    - Used to build up index key to table cross-references for developers

    Db2 12 solution → FL 501 LISTAGG

  - Greatly simplified SQLs



Db2 12 – what's hot?

```
,LISTAGG( CHAR(SUBSTR(DIGITS(D.COLSEQ) , 4, 2) CONCAT
                ' '                             CONCAT
                SUBSTR(D.COLNAME, 1, 18)        CONCAT
                CASE D.ORDERING
                WHEN ' ' THEN 'I'
                WHEN 'A' THEN '-'
                WHEN 'D' THEN 'D'
                WHEN 'R' THEN 'R'
                END
              )
        , ' ') WITHIN GROUP (ORDER BY D.COLSEQ ASC)
```

Db2 12 – what‘s hot?

```
DSNDB06   SYSDDF    LULIST               DSNFLX02  T      Y
DSNDB06   SYSDDF    LUMODES              DSNFMX01  T      Y
DSNDB06   SYSDDF    LUNAMES              DSNFNX01  T      Y
DSNDB06   SYSDDF    MODESELECT           DSNFDX01  T      Y
DSNDB06   SYSDDF    USERNAMES            DSNFEX01  T      Y
DSNDB06   SYSGPAUT  SYSRESAUTH           DSNAGH01  T      -
DSNDB06   SYSGPAUT  SYSRESAUTH           DSNAGX01  T      -
DSNDB06   SYSGRTNS  SYSROUTINES_OPTS     DSNROX01  T      Y
DSNDB06   SYSGRTNS  SYSROUTINES_SRC      DSNRSX01  T      -
DSNDB06   SYSGRTNS  SYSROUTINES_SRC      DSNRSX02  T      Y
```

```
01 LUNAME              -
01 LUNAME              - 02 MODENAME          -
01 LUNAME              -
01 LUNAME              - 02 AUTHID            D 03 PLANNAME
01 TYPE                - 02 AUTHID            D 03 LINKNAME
01 GRANTEE             - 02 QUALIFIER         - 03 NAME
01 GRANTOR             - 02 QUALIFIER         - 03 NAME
01 SCHEMA              - 02 ROUTINENAME       - 03 BUILDDATE
01 ROUTINENAME         -
01 SCHEMA              - 02 ROUTINENAME       - 03 BUILDDATE
```

Db2 12 – what's hot?

- SQL developers:
  - „Deleting data is nowadays critical, but having to create custom CURSORS just for delete to avoid locking problems and escalation is a real PITA!"

    Db2 12 solution → FETCH FIRST nnnn ROWS ONLY

  - Greatly simplified SQLs, a simple loop and you are done:

```
DELETE FROM BOXWELL.MY_HISTORY
WHERE HISTORY_DATE < CURRENT DATE - 90 DAYS
FETCH FIRST 5000 ROWS ONLY;
```

  - Do a COMMIT and then repeat until SQLCODE = +100

Roy Boxwell

SOFTWARE ENGINEERING

r.boxwell@seg.de

Session code: < V11>

Please fill out your session evaluation before leaving!