New England DB2 User's group Meeting September 15, 2011

Managing LOB, XML & Temporal Data

Hussaina Husain

IBM Senior I/T Specialist

hussaina@us.ibm.com

Disclaimer

- ***© Copyright IBM Corporation 2009. All rights reserved.***
- ***U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.***

- ***THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.***

- **IBM, the IBM logo, ibm.com, DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml**

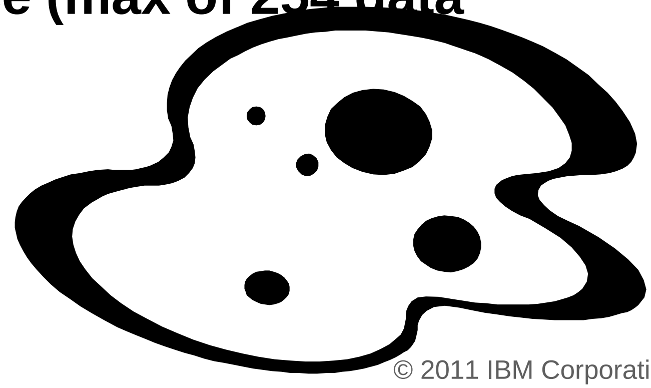
- **Other company, product, or service names may be trademarks or service marks of others.**

AGENDA

- **Overview of LOB concepts and architecture**
- **Overview of XML concepts and architecture**
- **LOB vs XML**
  - – How are they different
  - – When to use
- **Overview of Temporal Tables**
- **Advantages of Temporal tables**
- **Management LOB, XML objects & Temporal Data**
  - – CREATE / ALTER / DROP
  - – Migrate & Compare
  - – Utility Support
  - – Tools to administer and manage
- **References**
- Backup Slides

DB2 for z/OS LARGE OBJECTS (LOB)

- **Pictures, images, text documents, and movies**
- **3 types**
  - – BLOB – Binary Large Object – audio, image data
  - – CLOB – Character Large Object – SBCS, mixed character text
  - – DBCLOB – Double Byte Character Large Object
- **Each data set of a LOB table space → 64GB; 254 data sets / table space → max of 16TB for a non-partitioned LOB table space**
- **Max of 4096 partitions; 1 LOB table space (max of 254 data sets) / partition → 65,536 TB**
- **Usually accessed via GUI interfaces**



LOB ARCHITECTURE

BASE TABLE SPACE

1

BASE TABLE

3

4

KEY	ROWID	COL 2	LOB IND
A	Lob 1 value	User data A	LOB indicator 1
B	Lob 2 value	User data B	LOB indicator 2

GENERATED ALWAYS
GENERATED BY DEFAULT

Lob Indicator – 4 byte VARCHAR
2 byte flag (existence or validity)
2 byte version field

- •Rows represent LOBs
- •LOB's stored outside Base table
- •Base table space may be partitioned
- •If partitioned – 1 LOB TS / Partition

5

Auxiliary Index
based on ROWID
Used to navigate to
LOB data

One rowid
for multiple
LOB cols

AUXID
AUXVER

❖If you do not define a ROWID, DB2 will generate a hidden ROWID column at the end of the table.

❖No FIELDPROCs, EDITPROCs, VALIDPROCs, and check constraints allowed on LOB columns.

LOB TABLE SPACE-no compression allowed, DB2 V9 eliminated LOB locks, instead LRSN and page latching is used, DB2 10 Supports logging > 2 GB and DEFINE(NO)

2

AUXILIARY TABLE

AUXID	AUXVER	AUXVALUE (LOB data)
Lob 1 value		LOB data for data row A
Lob 2 value		LOB data for data row B

❖One LOB TS for each partition of a base table

XML – Extensible Markup Language

- **Standard for exchanging data – can be transformed to other formats (HTML …) – Platform and vendor independent**
- **Was developed to overcome the limitations of HTML**
  - Can store almost any type of data
  - Originally stored in flat files
    - Fixed length
    - Delimited
  - Moved to be stored in data bases
- **Define complex documents and structures**
  - Structured or semi-structured
  - Schema or schema-less
- **Flexible**
  - Self describing
  - Easy to extend – add tag

```
<DEPT DEPTNO="D01" NAME="DEVELOPMENT CENTER">
   <PROJ PROJNO="AD3100" NAME="ADMIN SERVICES">
      <EMP EMPNO="000010" EMPNAME="BRIAN BARTAK">
      </EMP>
   </PROJ>
   <PROJ PROJNO="MA2100" NAME="WELD LINE
AUTOMATION">
      <EMP EMPNO="000010" EMPNAME="BRIAN BARTAK">
      </EMP>
      <EMP EMPNO="000110" EMPNAME="VINCENZO
LUCCHESI">
      </EMP>
   </PROJ>
</DEPT>
```

WELL FORMED

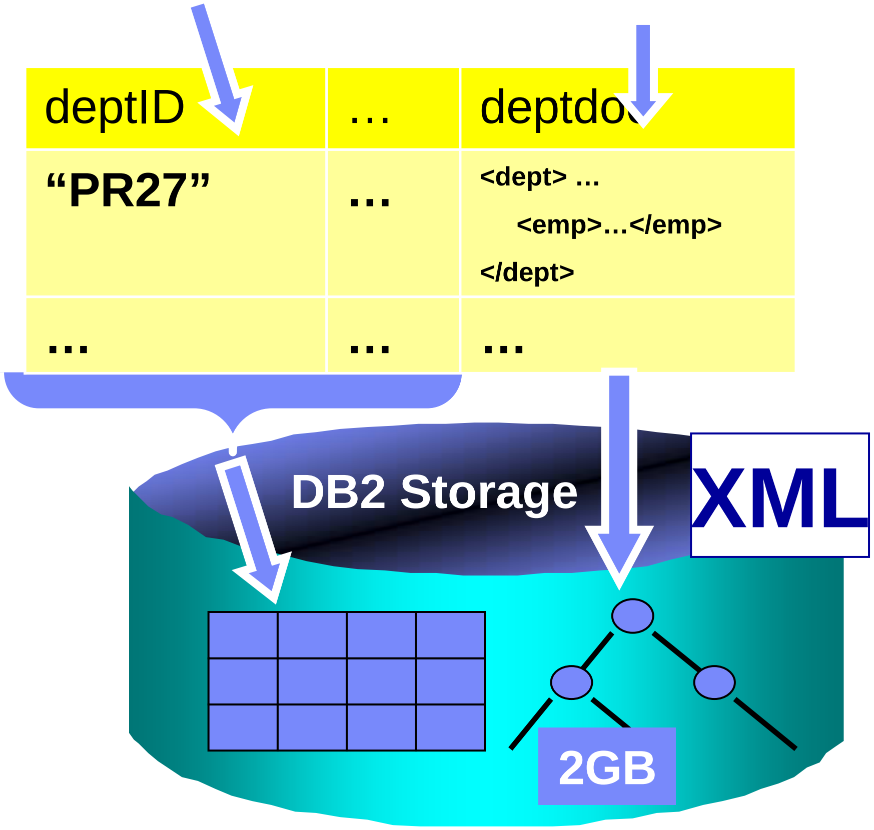
ROOT
NODES
ELEMENT
ATTRIBUTES
VALUES

DB2 9 pureXML Databases

- **XML data is stored in XML-typed columns in tables (UTF-8)**

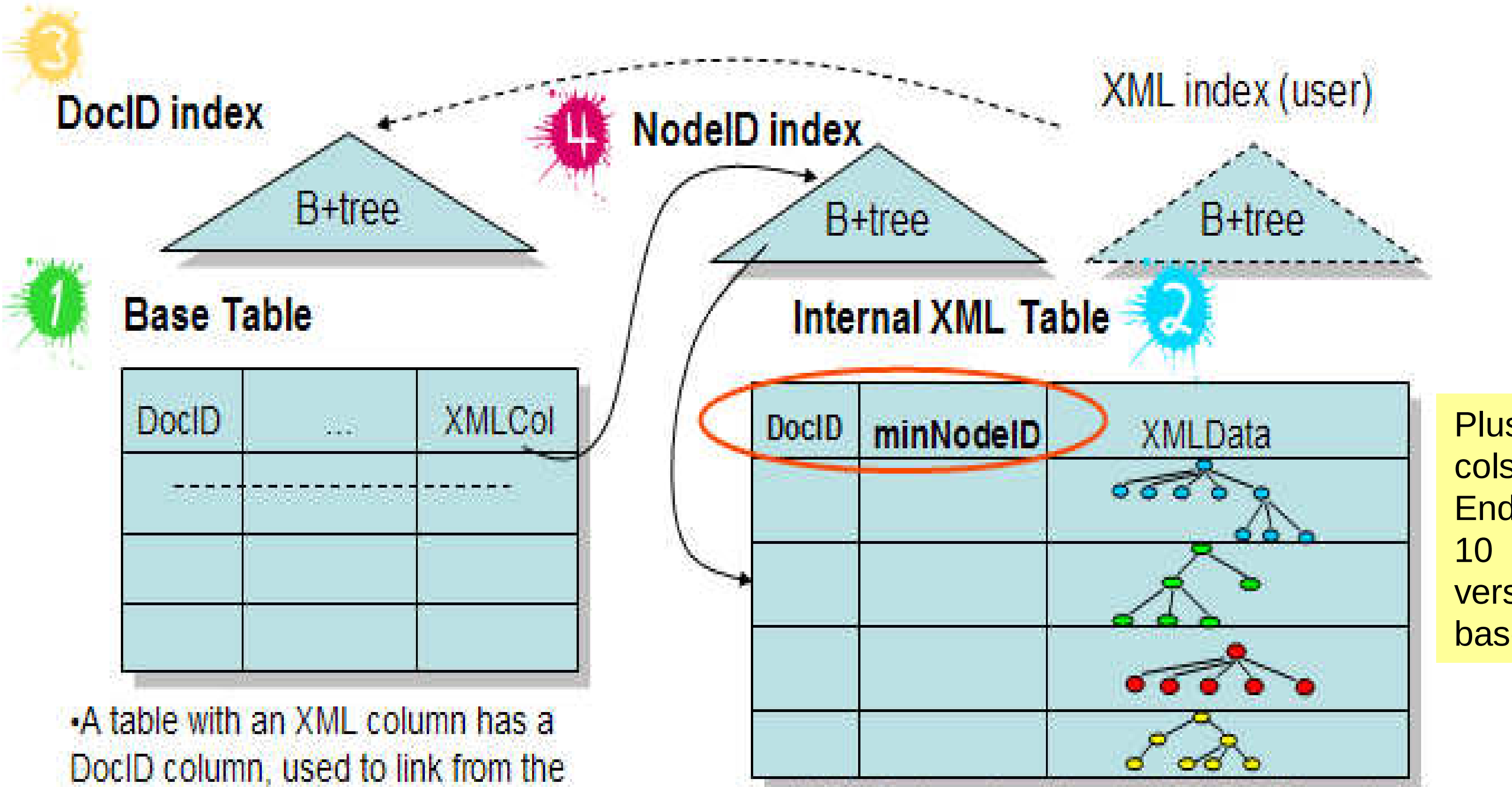
Create table dept (deptID char(8),…, deptdoc xml);

- **XML is stored in a** parsed hierarchical **format**
- **Relational columns are stored in relational format**
- **The XML table space inherits the base tablespace attributes**
  - Several items like Compression. ..
  - Range-partitioned base table spaces: XML partitioning follows base table partitioning.
  - For non range-partitioned base table spaces, PBG table space is used for XML
- **XML table space uses 16KB bufferpool**
- **No length associated with XML col**
- **XML type for col, host variable, Parameters for STP and UDF**



deptID	…	deptdoc
“PR27”	…	<dept> … <emp>…</emp> </dept>
…	…	…

XML Storage



Plus two more cols, start_TS & End-TS in DB2 10 to support versioning if the base TS is PBG

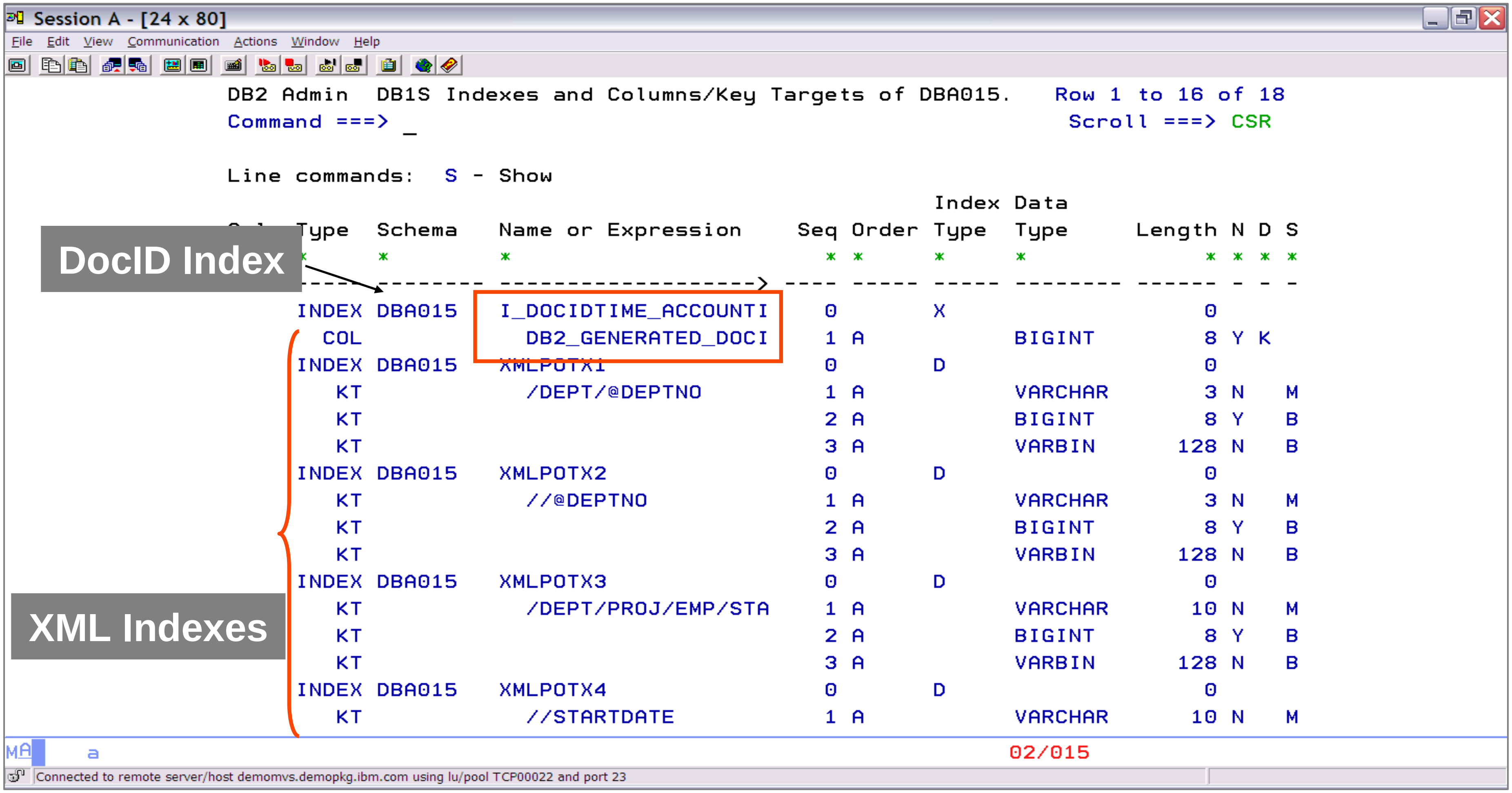
•A table with an XML column has a DocID column, used to link from the base table to the XML table.

•A DocID index is used for getting to base table rows from XML indexes.

Each XMLData column is a VARBINARY, containing a subtree or a sequence of subtrees, with context path. Rows in XML table are freely movable, linked with a NodeID index.

- NodeID index maintains document order, maps logical node IDs to physical record IDs.

XML Storage - Indexes



```
DB2 Admin  DB1S Indexes and Columns/Key Targets of DBA015.   Row 1 to 16 of 18
Command ===> _                                                Scroll ===> CSR

Line commands:  S - Show
                                                    Index Data
Sel  Type  Schema   Name or Expression    Seq Order Type  Type     Length N D S
     *     *        *                       * *     *     *             * * * *
     ----- -------- -------------------> ---- ----- ----- -------- ------ - - -
     INDEX DBA015   I_DOCIDTIME_ACCOUNTI    0       X                   0
       COL            DB2_GENERATED_DOCI    1 A           BIGINT        8 Y K
     INDEX DBA015   XMLPOTX1                0       D                   0
        KT            /DEPT/@DEPTNO         1 A           VARCHAR       3 N   M
        KT                                  2 A           BIGINT        8 Y   B
        KT                                  3 A           VARBIN      128 N   B
     INDEX DBA015   XMLPOTX2                0       D                   0
        KT            //@DEPTNO             1 A           VARCHAR       3 N   M
        KT                                  2 A           BIGINT        8 Y   B
        KT                                  3 A           VARBIN      128 N   B
     INDEX DBA015   XMLPOTX3                0       D                   0
        KT            /DEPT/PROJ/EMP/STA    1 A           VARCHAR      10 N   M
        KT                                  2 A           BIGINT        8 Y   B
        KT                                  3 A           VARBIN      128 N   B
     INDEX DBA015   XMLPOTX4                0       D                   0
        KT            //STARTDATE           1 A           VARCHAR      10 N   M
MA     a                                                   02/015
Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00022 and port 23
```

DocID Index

XML Indexes

XML indexes can be defined on XML columns using XMLPATTERN. These value indexes are to accelerate the query processing for the XMLEXISTS() predicate and XMLTABLE() function.

How is XML different from LOB?

XML

DB2 9 NFM – pureXML hierarchical

- Self describing **– metadata; structured**
- Data **- multiple types of data in a document; nesting and repeating**
- Inherent ordering **– data items are in order of the data in the document**
- Indexes **– create indexes for search purposes**
- Compression **supported**
- Size **– no architectural limit**
  - – For exchanging data, the limit is to 2GB.

LOB

Flat **–rigid row and column structure**

- Data defined by the column definition **– all data in a column must be the same type**
- Not ordered **– unless using the ORDER BY clause on one or more columns**
- **Cannot create** indexes **to be used for searching**
- Compression **is not supported**
- Maximum size
  - – Non-partitioned 16 TB
  - – Partitioned – 65,536 TB

When should I use XML vs LOB to store data?

FUNCTION	XML	LOB
Flexibility **making frequent design changes**	Y	
Maximum performance	**faster for returning smaller amounts of data**	Y **(large documents)**
Sparse data **- attributes do not apply to all occurrences of the entity**	Y	
Searches	Y	
INSERT / RETRIEVAL **of the entire document**		Y
Processing time – subsequent processing **depends on data being stored in a relational DB**	**Y**	
Frequent partial updates	**Y**	

CREATE / ALTER DB2 Objects

LOB / XML Columns

LOB

1 - **Manual**
  - Most flexibility, but most time consuming
  - Implement your own naming convention

2 **Automatic**
  - CURRENT RULES special register = ‘STD’ (V8 and higher) or DB2 V9 & 10 Automatic Creation of Objects – **do NOT specify the IN clause**
    - DB2 will create all the necessary LOB objects

XML

3 - **DB2 implicitly creates**
  - XML Table Space
  - XML Table
  - Adds the DocID when the 1st XML column is defined
  - DOCID / NODEID Indexes

MANUAL CREATION of LOB Objects

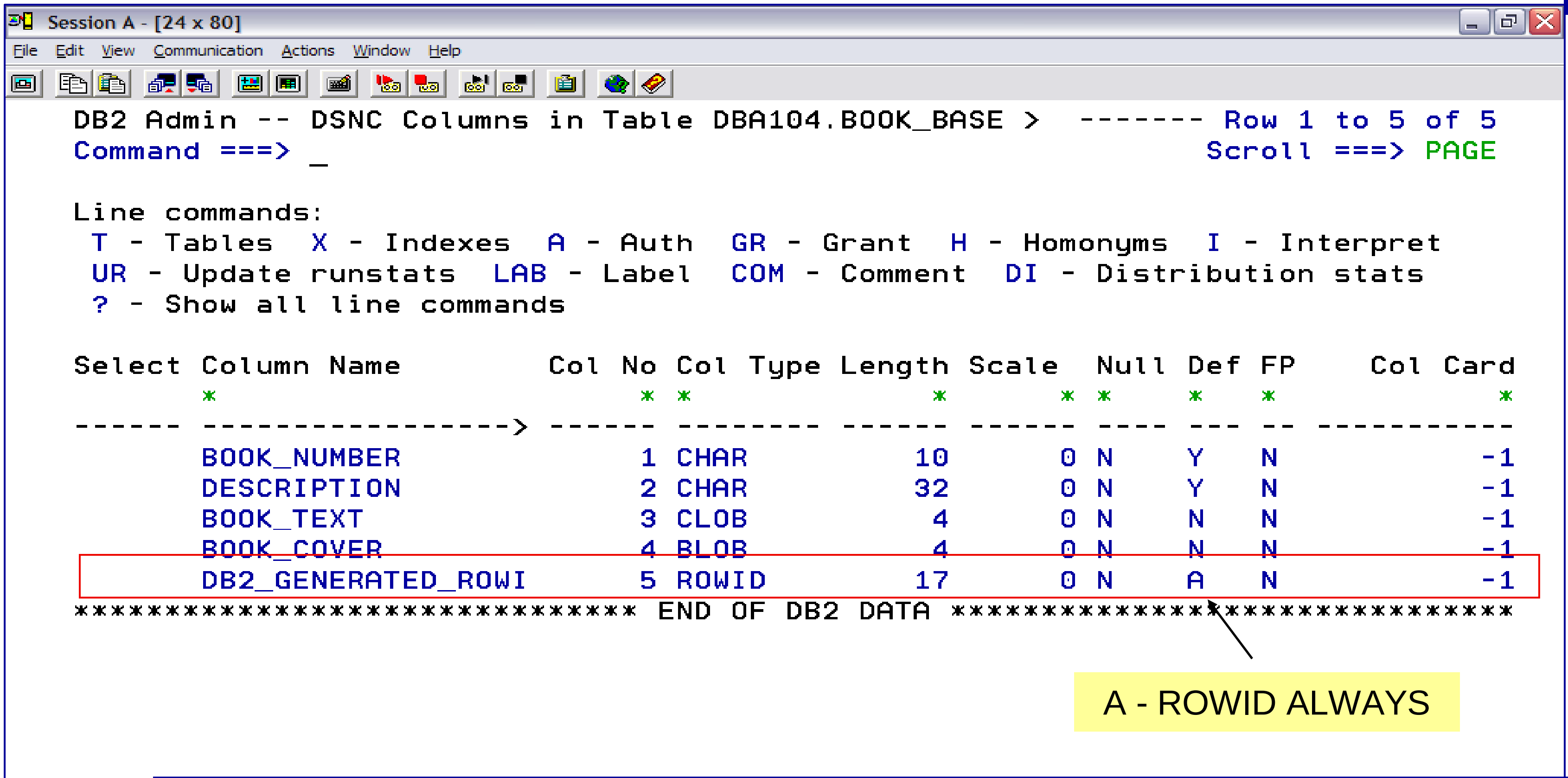
LOB

- **Create the** BASE **table**
  - Table space must be in the same database where the LOB table space(s) are stored
  - Base table contains a **ROWID**
    - Unique value related to auxiliary tables
    - Only need 1 / base table
- **Create the** LOB table space
  - Need 1 LOB table space for each LOB column in the base table
- **Create the** Auxiliary table **– 1 per LOB table space**
- **Create the** Auxiliary index **(only can have 1 / aux table)**

```
--create base table
CREATE TABLE TB01
  ( FKEY INTEGER, …
    FROWID ROWID,
    FCLOB CLOB(10M),….) …

--create LOB table space
CREATE LOBTABLESPACE LTS01 … NOT
   LOGGED;

--create auxiliary table for LOB column FCLOB
CREATE AUXTABLE AUXTB01 IN DB
   STORES TB01
   COLUMN FCLOB;

--create index for auxiliary table
CREATE UNIQUE INDEX AUXIX01
   ON AUXTB01…
```

GENERATED ROWID COLUMN for LOB in a base table

LOB

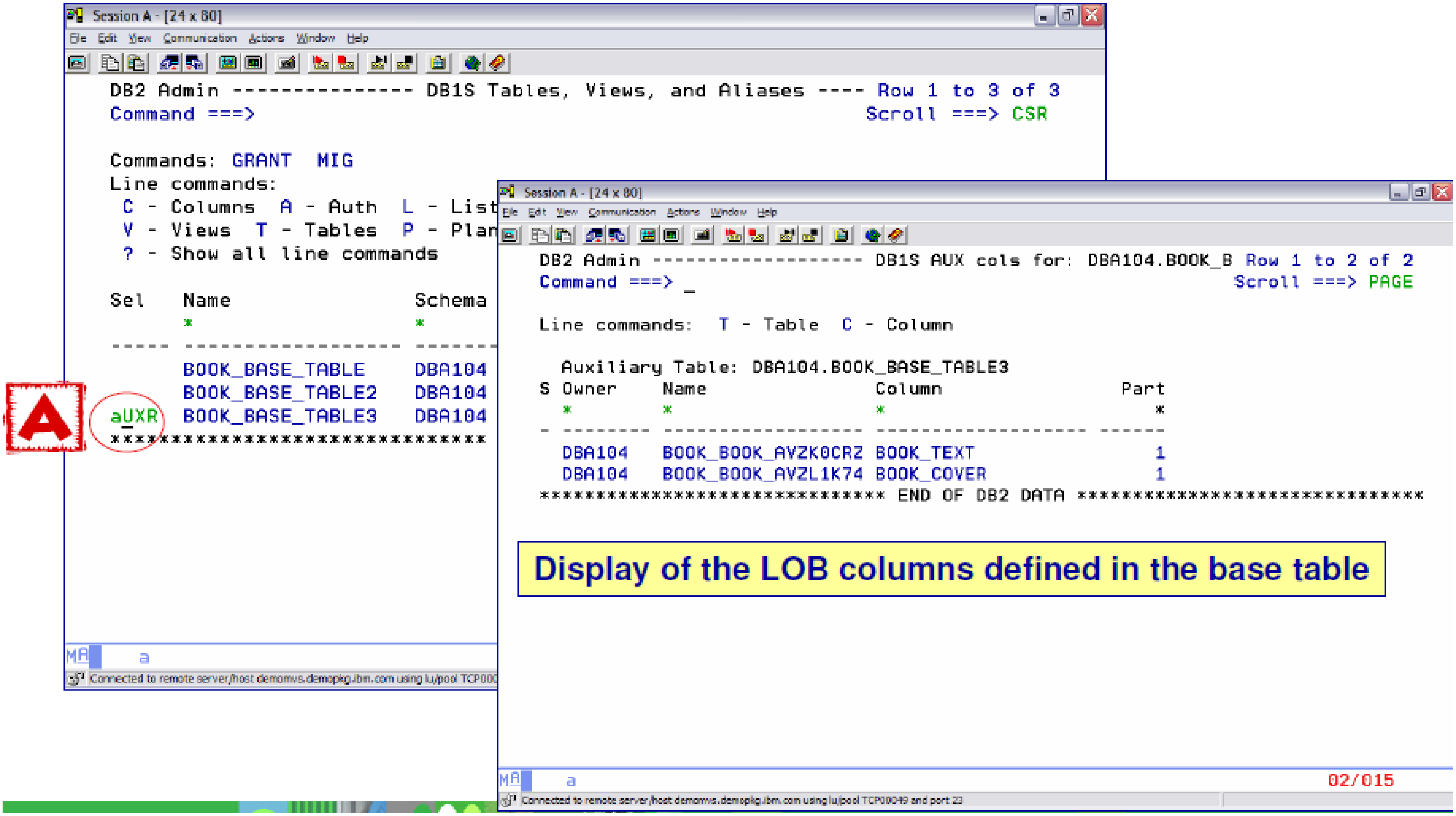


```
DB2 Admin -- DSNC Columns in Table DBA104.BOOK_BASE >  ------- Row 1 to 5 of 5
Command ===> _                                               Scroll ===> PAGE

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 ? - Show all line commands

Select Column Name         Col No Col Type Length Scale  Null Def FP     Col Card
       *                        * *             *      * *    *   *             *
------ ------------------> ------ -------- ------ ------ ---- --- -- -----------
       BOOK_NUMBER              1 CHAR         10      0 N    Y   N            -1
       DESCRIPTION              2 CHAR         32      0 N    Y   N            -1
       BOOK_TEXT                3 CLOB          4      0 N    N   N            -1
       BOOK_COVER               4 BLOB          4      0 N    N   N            -1
       DB2_GENERATED_ROWI       5 ROWID        17      0 N    A   N            -1
******************************* END OF DB2 DATA *******************************
```

A - ROWID ALWAYS

- ✓ROWID is defined as a varchar 17
- ✓Only 1 ROWID column – even if there are multiple LOB columns
- ✓ROWID is a unique and permanent identifier for ea. row in the base table

***AUXR** Display Associated AUX Data Columns*

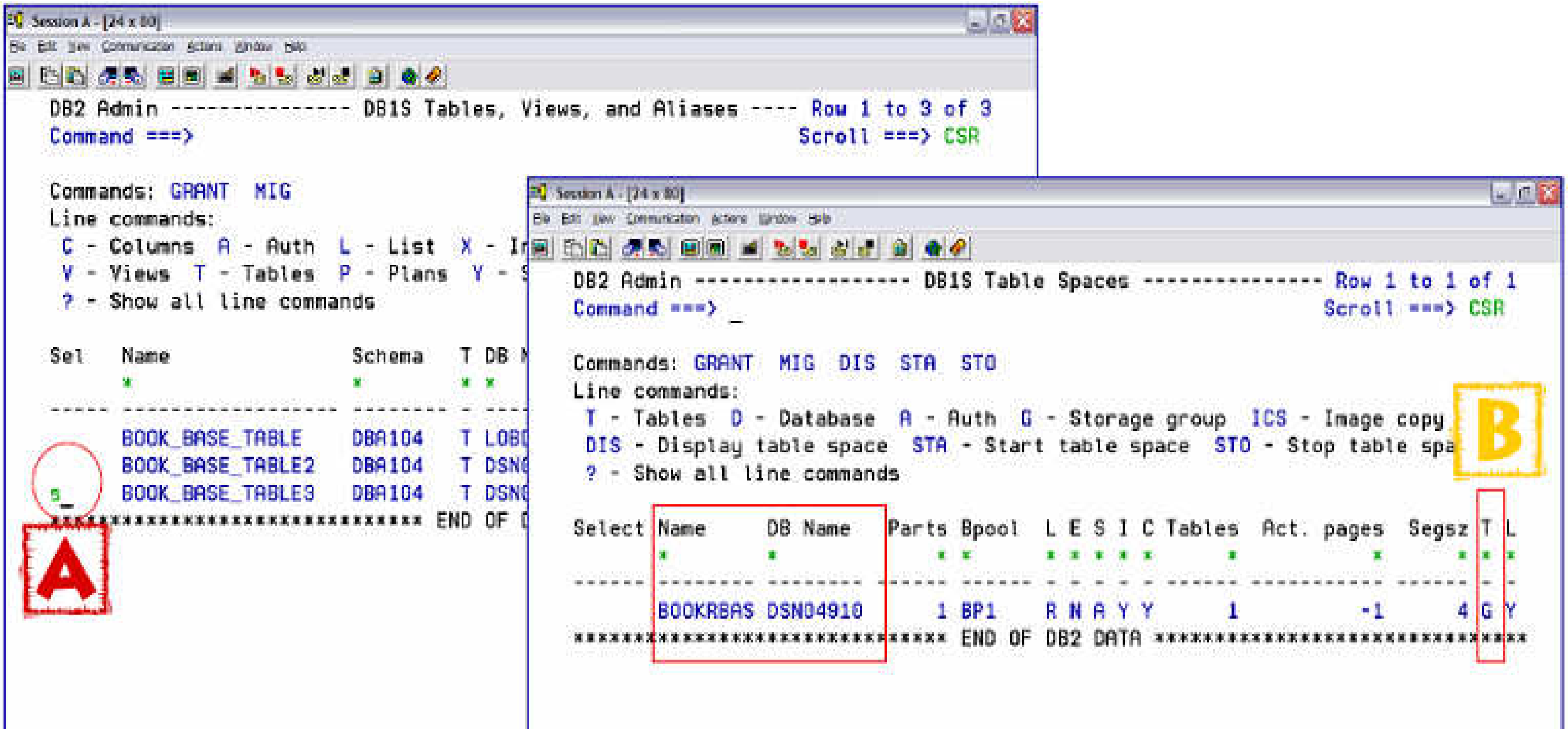


```
DB2 Admin --------------- DB1S Tables, Views, and Aliases ---- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List
 V - Views   T - Tables  P - Plan
 ? - Show all line commands

Sel   Name                 Schema
      *                    *
----- ------------------ -------
      BOOK_BASE_TABLE      DBA104
      BOOK_BASE_TABLE2     DBA104
aUXR  BOOK_BASE_TABLE3     DBA104
*******************************

MA        a
Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP000
```

```
DB2 Admin ------------------- DB1S AUX cols for: DBA104.BOOK_B Row 1 to 2 of 2
Command ===> _                                               Scroll ===> PAGE

Line commands:  T - Table  C - Column

 Auxiliary Table: DBA104.BOOK_BASE_TABLE3
S Owner    Name               Column                 Part
  *        *                  *                         *
- -------- ------------------ ------------------ ------
  DBA104   BOOK_BOOK_AVZK0CRZ BOOK_TEXT                 1
  DBA104   BOOK_BOOK_AVZL1K74 BOOK_COVER                1
******************************** END OF DB2 DATA ********************************

MA        a                                                           02/015
Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00049 and port 23
```

Display of the LOB columns defined in the base table

New Table – Put an **S** to list the table space DB2 created

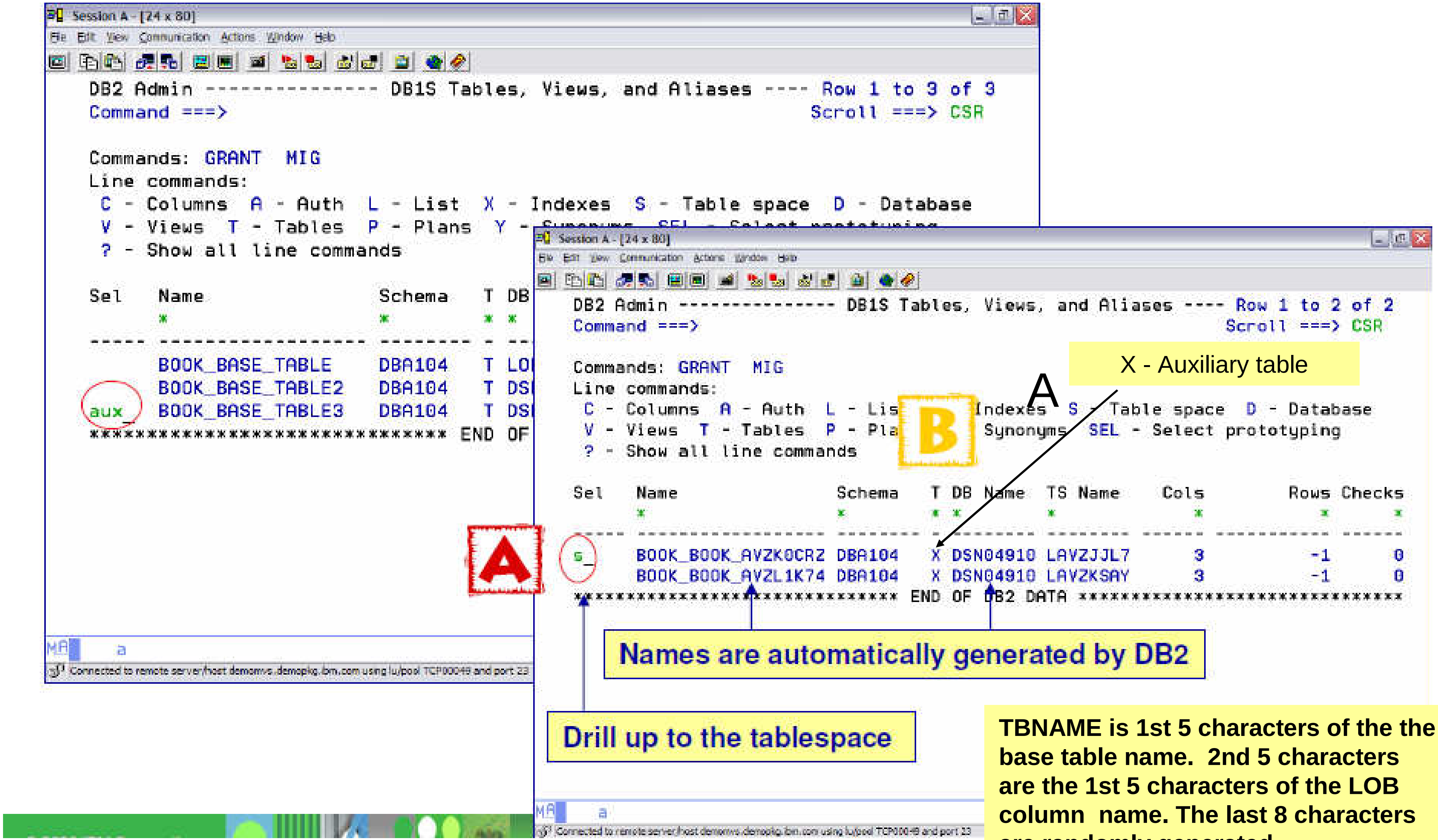


```
DB2 Admin --------------- DB1S Tables, Views, and Aliases ---- Row 1 to 3 of 3
Command ===>                                                   Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - I
 V - Views  T - Tables  P - Plans  Y - S
 ? - Show all line commands

Sel   Name               Schema   T DB N
      *                  *        * *
----- ------------------ -------- - ----
      BOOK_BASE_TABLE    DBA104   T LOBC
      BOOK_BASE_TABLE2   DBA104   T DSNC
S_    BOOK_BASE_TABLE3   DBA104   T DSNC
****************************** END OF D
```

```
DB2 Admin ------------------ DB1S Table Spaces --------------- Row 1 to 1 of 1
Command ===> _                                                 Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
 DIS - Display table space  STA - Start table space  STO - Stop table spa
 ? - Show all line commands

Select Name     DB Name  Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *            * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       BOOKRBAS DSN04910      1 BP1    R N A Y Y      1          -1      4 G Y
******************************* END OF DB2 DATA *******************************
```

Universal TS PBG for the base tablespace

AUX – *Display Auxiliary Objects*



```
DB2 Admin --------------- DB1S Tables, Views, and Aliases ---- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name               Schema   T DB
      *                  *        * *
----- ------------------ -------- - --
      BOOK_BASE_TABLE    DBA104   T LO
      BOOK_BASE_TABLE2   DBA104   T DS
aux   BOOK_BASE_TABLE3   DBA104   T DS
******************************* END OF
```

MA a

Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00049 and port 23

```
DB2 Admin --------------- DB1S Tables, Views, and Aliases ---- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - Lis     Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Pla     Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name               Schema   T DB Name  TS Name   Cols         Rows Checks
      *                  *        * *        *         *               *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
s     BOOK_BOOK_AVZK0CRZ DBA104   X DSN04910 LAVZJJL7      3          -1      0
      BOOK_BOOK_AVZL1K74 DBA104   X DSN04910 LAVZKSAY      3          -1      0
******************************* END OF DB2 DATA *******************************
```

MA a

Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00049 and port 23

X - Auxiliary table

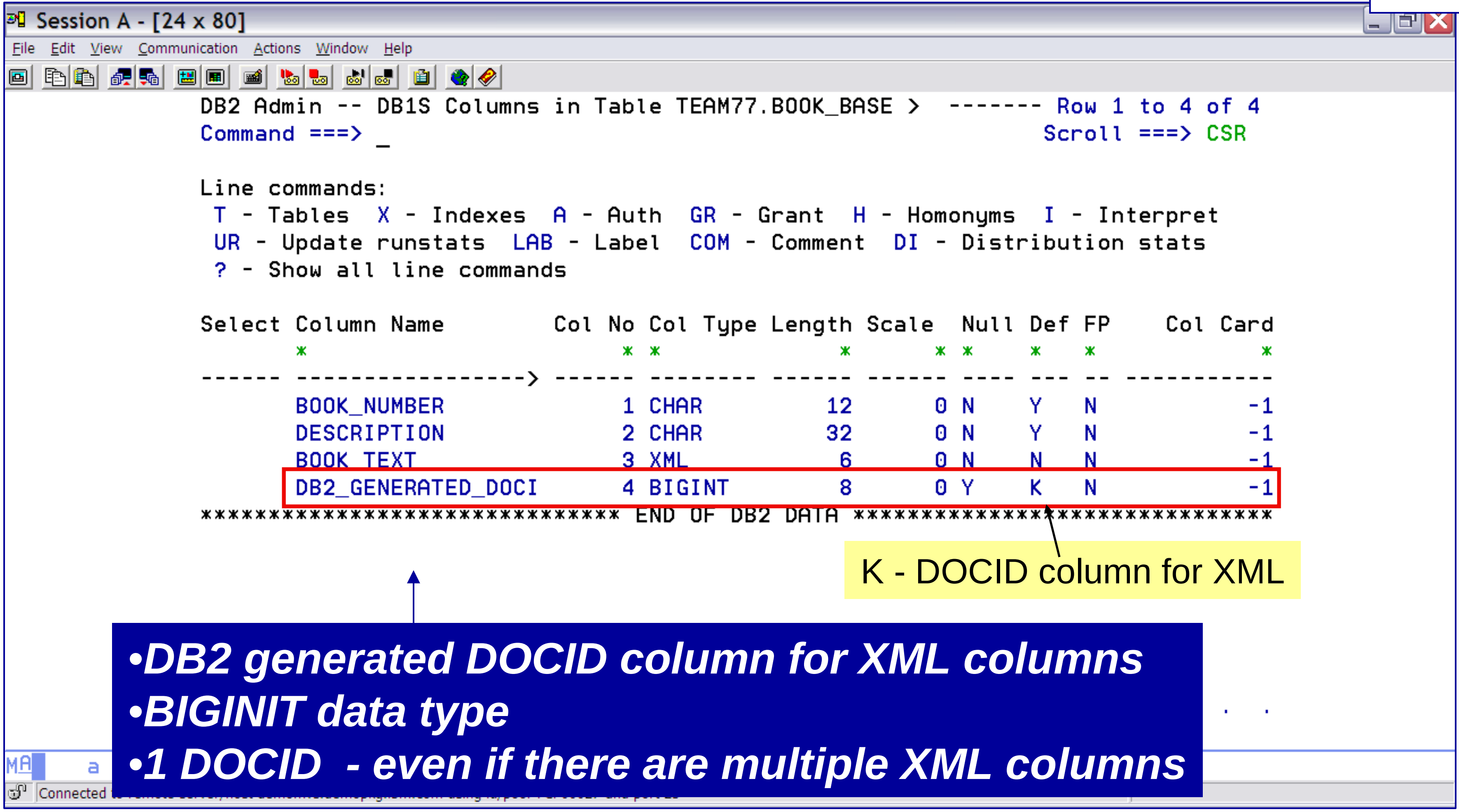
Names are automatically generated by DB2

Drill up to the tablespace

TBNAME is 1st 5 characters of the the base table name. 2nd 5 characters are the 1st 5 characters of the LOB column name. The last 8 characters are randomly generated

CREATE / ALTER TABLE w/ XML Column(s)

XML



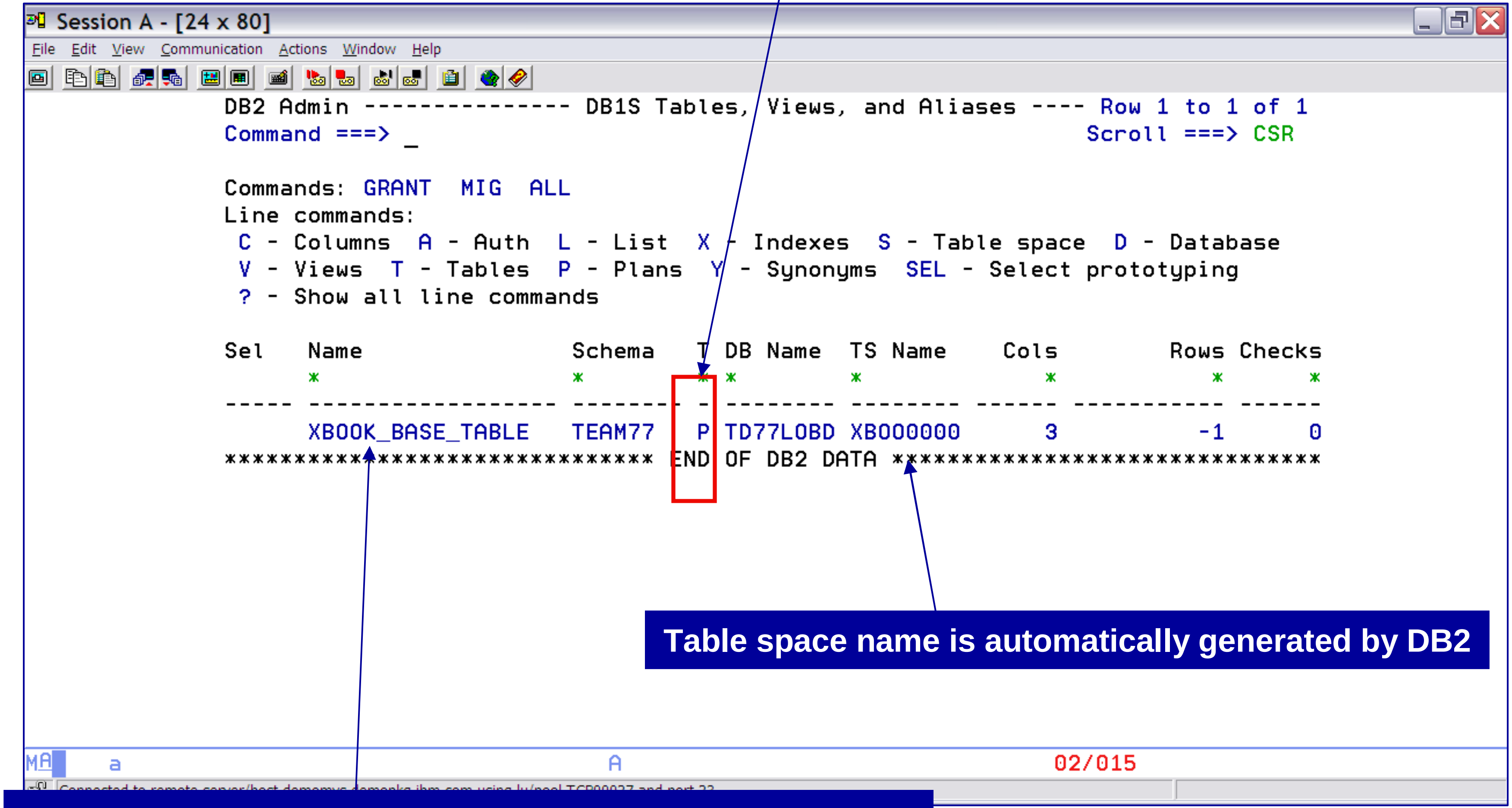
```
DB2 Admin -- DB1S Columns in Table TEAM77.BOOK_BASE >  ------- Row 1 to 4 of 4
Command ===> _                                                 Scroll ===> CSR

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 ? - Show all line commands

Select Column Name         Col No Col Type Length Scale  Null Def FP    Col Card
       *                        * *             *      * *    *   *            *
------ ------------------> ------ -------- ------ ------ ---- --- -- -----------
       BOOK_NUMBER              1 CHAR         12      0 N    Y   N           -1
       DESCRIPTION              2 CHAR         32      0 N    Y   N           -1
       BOOK_TEXT                3 XML           6      0 N    N   N           -1
       DB2_GENERATED_DOCI       4 BIGINT        8      0 Y    K   N           -1
******************************* END OF DB2 DATA *******************************
```

K - DOCID column for XML

- *DB2 generated DOCID column for XML columns*
- *BIGINIT data type*
- *1 DOCID - even if there are multiple XML columns*

XML Objects

Implicitly defined XML Tables are stored in SYSTABLES as a type P



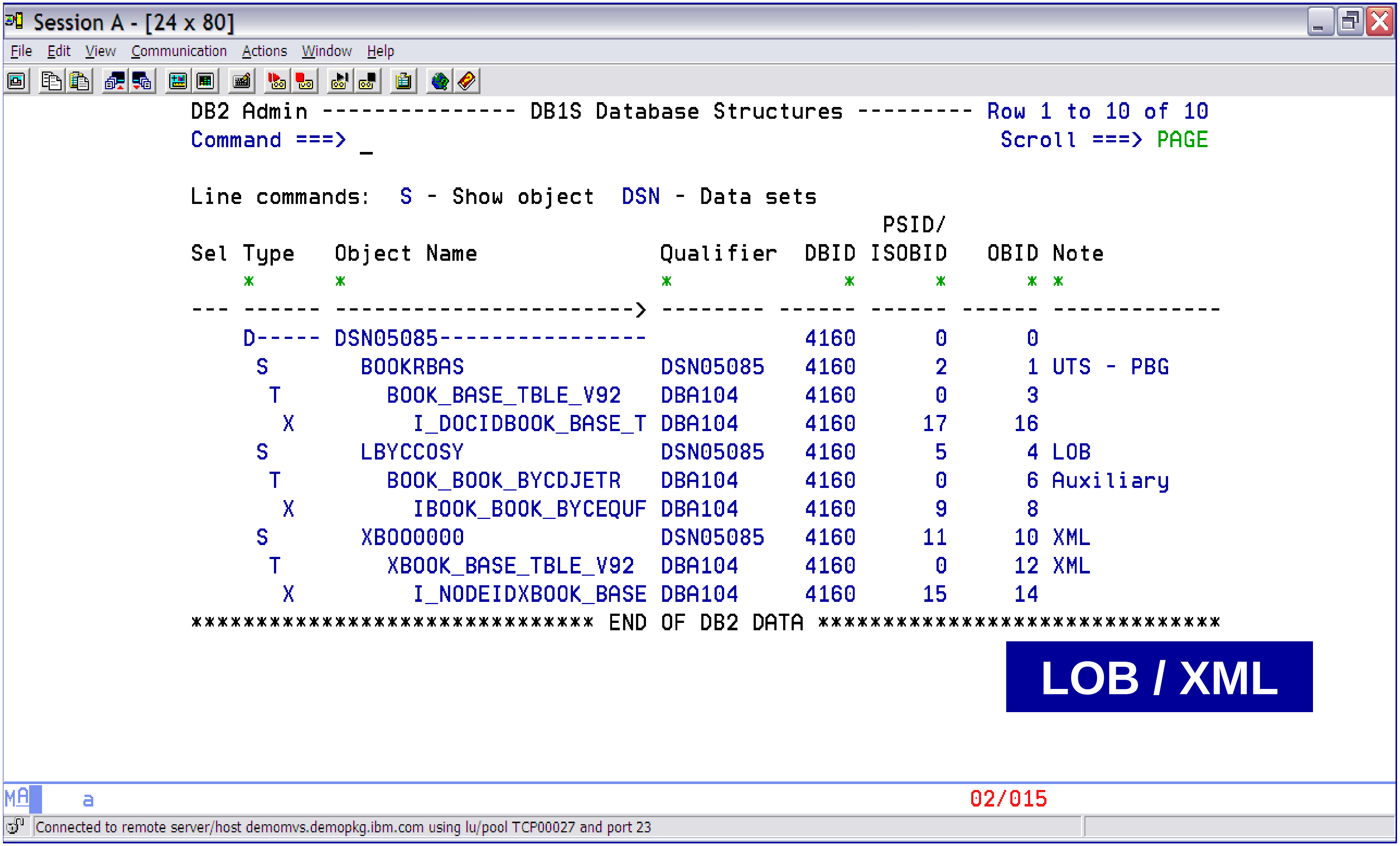
```
DB2 Admin --------------- DB1S Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ===> _                                                 Scroll ===> CSR

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Schema   T DB Name  TS Name    Cols         Rows Checks
      *                   *        * *        *             *            *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      XBOOK_BASE_TABLE    TEAM77   P TD77LOBD XB000000      3           -1      0
******************************* END OF DB2 DATA *******************************
```

```
MA    a                                 A                          02/015
```

Table space name is automatically generated by DB2

- **Table name is automatically generated by DB2**
- **Appended an X to the front of the base table name**

Display of LOB/XML related objects using Admin Tool-DS command



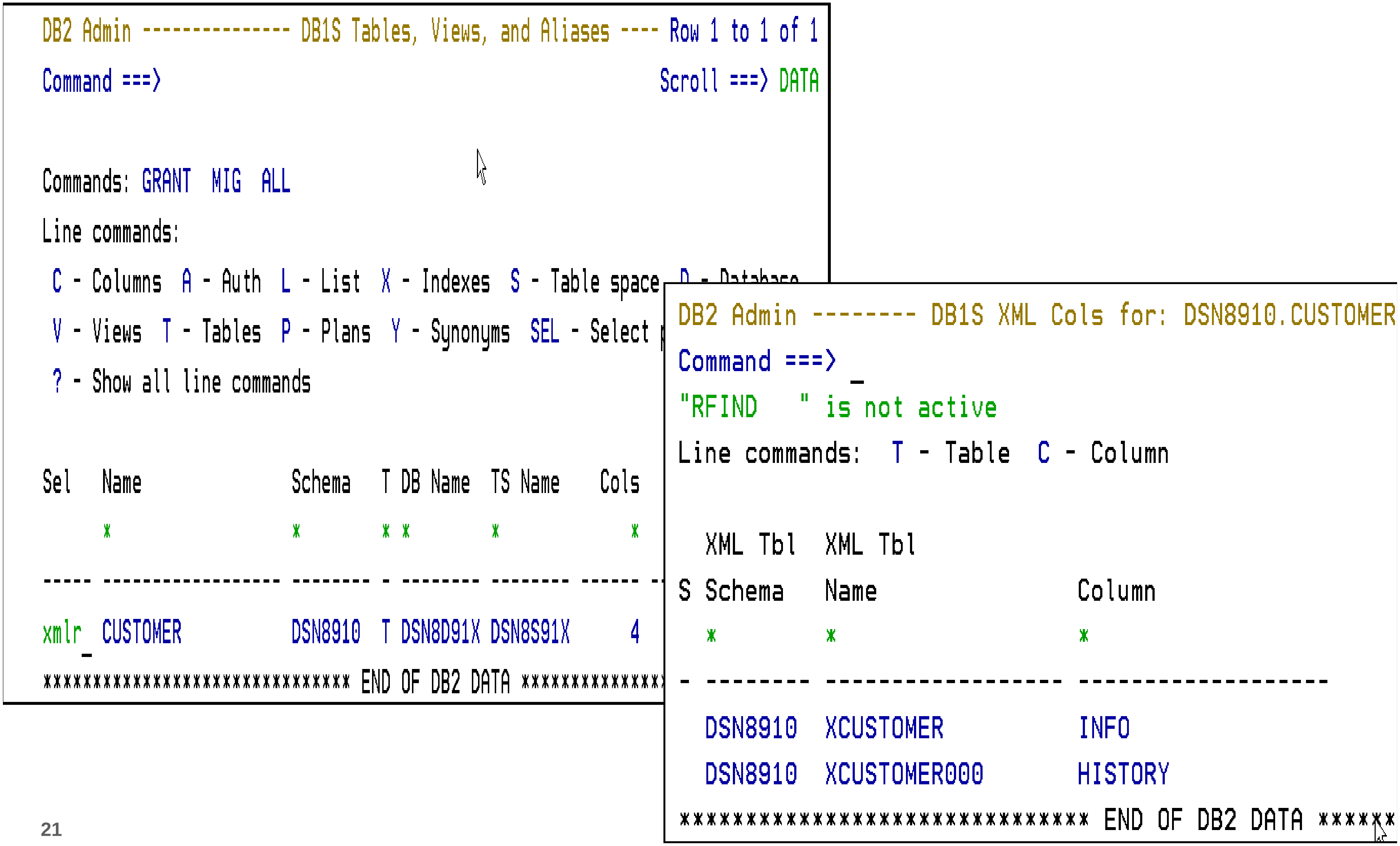
```
DB2 Admin --------------- DB1S Database Structures --------- Row 1 to 10 of 10
Command ===> _                                                 Scroll ===> PAGE

Line commands:  S - Show object  DSN - Data sets
                                                      PSID/
Sel Type   Object Name             Qualifier  DBID  ISOBID    OBID Note
    *      *                       *             *       *       * *
--- ------ ----------------------> -------- ------ ------ ------ -------------
    D----- DSN05085----------------             4160      0      0
     S       BOOKRBAS              DSN05085    4160      2      1 UTS - PBG
      T        BOOK_BASE_TBLE_V92  DBA104      4160      0      3
       X         I_DOCIDBOOK_BASE_T DBA104     4160     17     16
     S       LBYCCOSY              DSN05085    4160      5      4 LOB
      T        BOOK_BOOK_BYCDJETR  DBA104      4160      0      6 Auxiliary
       X         IBOOK_BOOK_BYCEQUF DBA104     4160      9      8
     S       XB000000              DSN05085    4160     11     10 XML
      T        XBOOK_BASE_TBLE_V92 DBA104      4160      0     12 XML
       X         I_NODEIDXBOOK_BASE DBA104     4160     15     14
******************************* END OF DB2 DATA *******************************
```

LOB / XML

```
MA      a                                                   02/015
```

Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00027 and port 23

Display associated XML data column



```
DB2 Admin --------------- DB1S Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> DATA


Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select
 ? - Show all line commands

Sel   Name                Schema   T DB Name  TS Name    Cols
      *                   *        * *        *             *
----- ------------------ -------- - -------- -------- ------
xmlr_ CUSTOMER            DSN8910  T DSN8D91X DSN8S91X      4
******************************** END OF DB2 DATA ***************
```

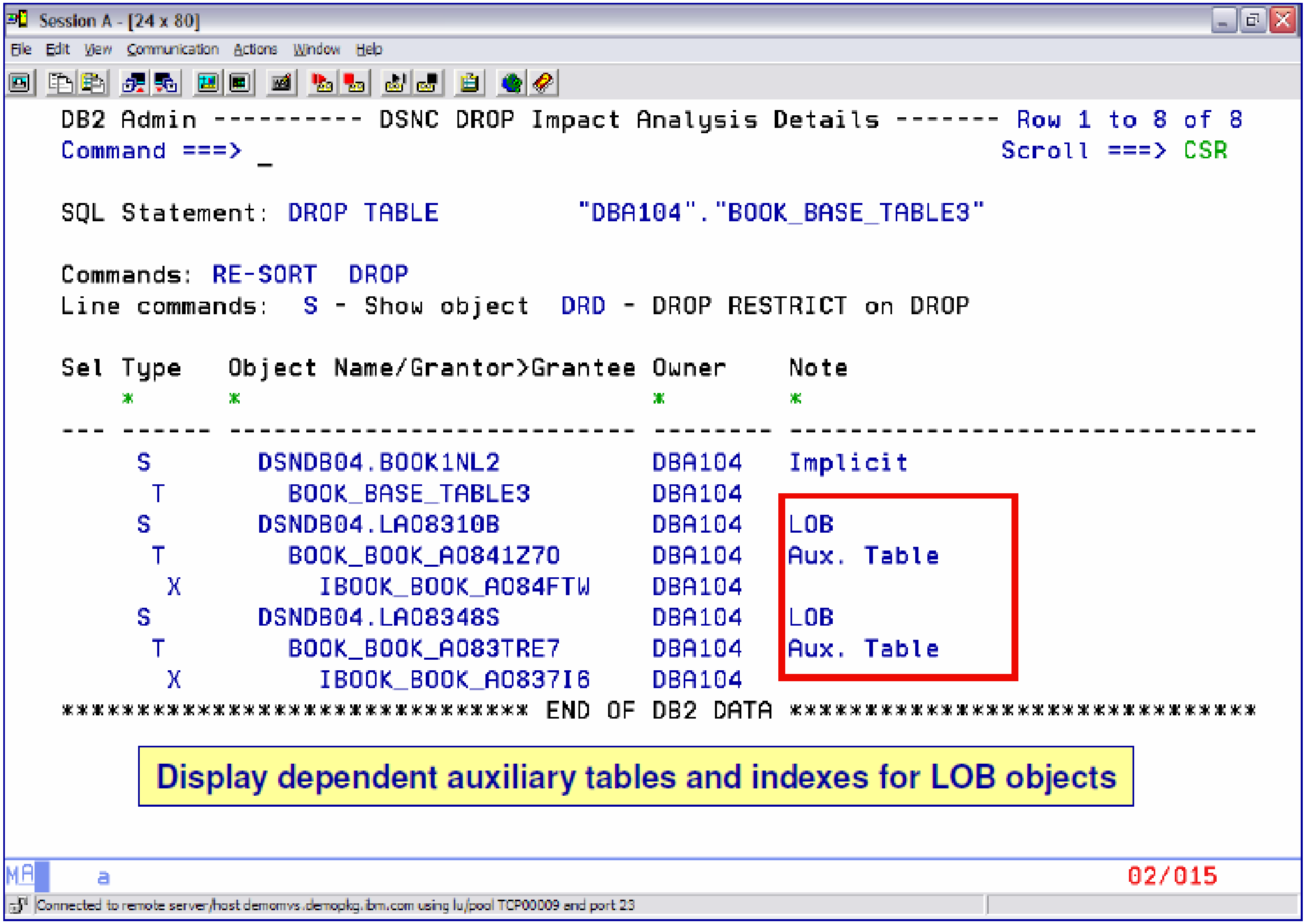
```
DB2 Admin -------- DB1S XML Cols for: DSN8910.CUSTOMER
Command ===> _
"RFIND   " is not active
Line commands:  T - Table  C - Column


  XML Tbl  XML Tbl
S Schema   Name                Column
  *        *                   *
- -------- ------------------ ------------------
  DSN8910  XCUSTOMER           INFO
  DSN8910  XCUSTOMER000        HISTORY
******************************** END OF DB2 DATA ******
```

DROP LOB / XML Objects

OB & xm

- LOB
  - – DROP objects created (**CURRENT RULE = ‘STD’** or without In clause)
    - Enforcing PK index
    - Enforcing unique key index
    - Index on ROWID column on base table if applicable
    - Auxiliary Table
    - Auxiliary Index
    - TS for base table and auxiliary table
  - – DROP base table or base table space **manually created**
    - Auxiliary table and auxiliary index are dropped
    - **LOB tablespace definition remains**
- XML
  - – DROP base table or base table space
    - Auxiliary XML table and auxiliary XML indexes are dropped

DROP IMPACT REPORT from Admin Tool



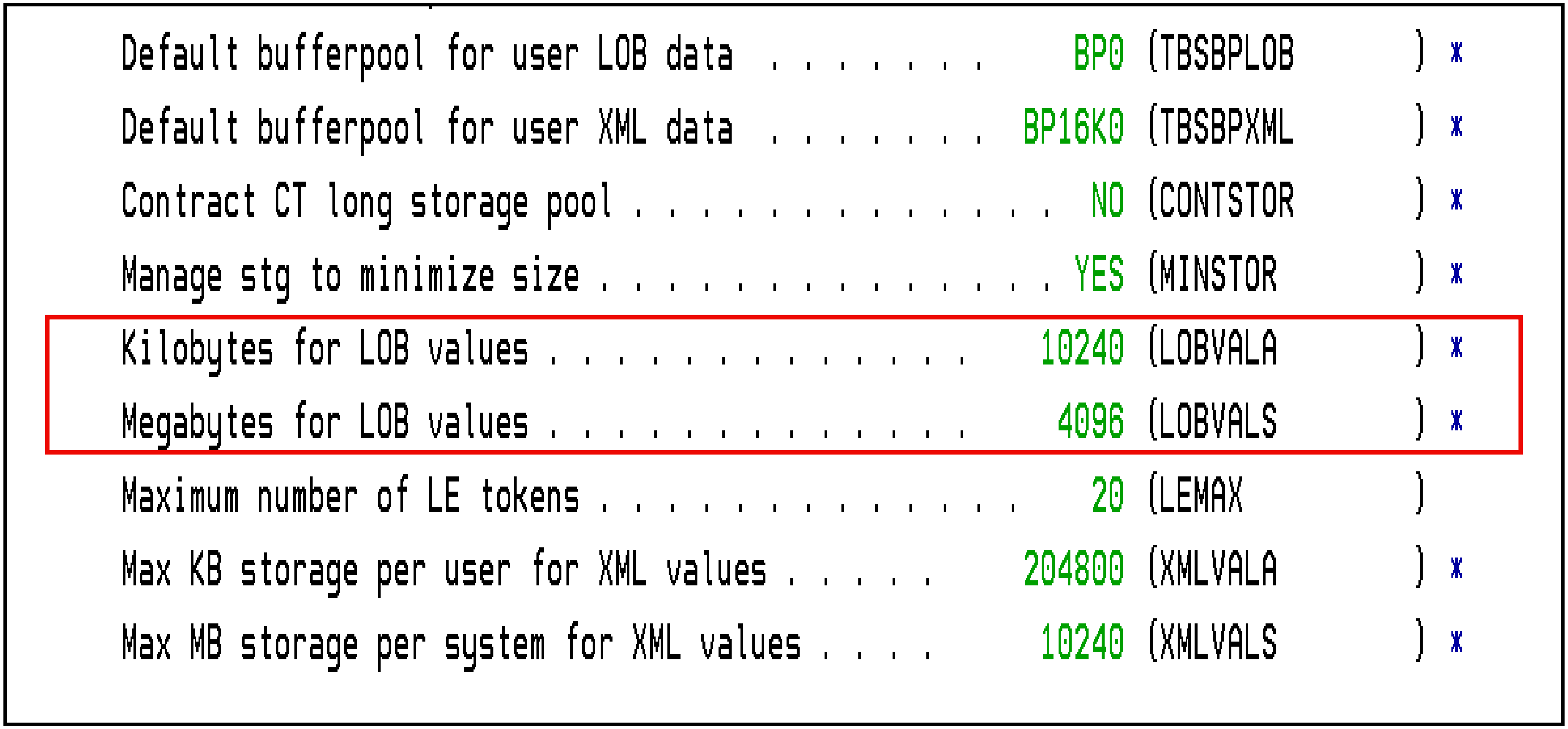
```
DB2 Admin ---------- DSNC DROP Impact Analysis Details ------- Row 1 to 8 of 8
Command ===> _                                                Scroll ===> CSR

SQL Statement: DROP TABLE         "DBA104"."BOOK_BASE_TABLE3"

Commands: RE-SORT  DROP
Line commands:  S - Show object  DRD - DROP RESTRICT on DROP

Sel Type   Object Name/Grantor>Grantee Owner    Note
    *      *                           *        *
--- ------ --------------------------- -------- -------------------------------
     S       DSNDB04.BOOK1NL2          DBA104   Implicit
      T        BOOK_BASE_TABLE3        DBA104
     S       DSNDB04.LA08310B          DBA104   LOB
      T        BOOK_BOOK_A0841Z70      DBA104   Aux. Table
       X         IBOOK_BOOK_A084FTW    DBA104
     S       DSNDB04.LA08348S          DBA104   LOB
      T        BOOK_BOOK_A083TRE7      DBA104   Aux. Table
       X         IBOOK_BOOK_A083716    DBA104
******************************* END OF DB2 DATA *******************************
```

Display dependent auxiliary tables and indexes for LOB objects

```
MA      a                                                              02/015
```

Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00009 and port 23

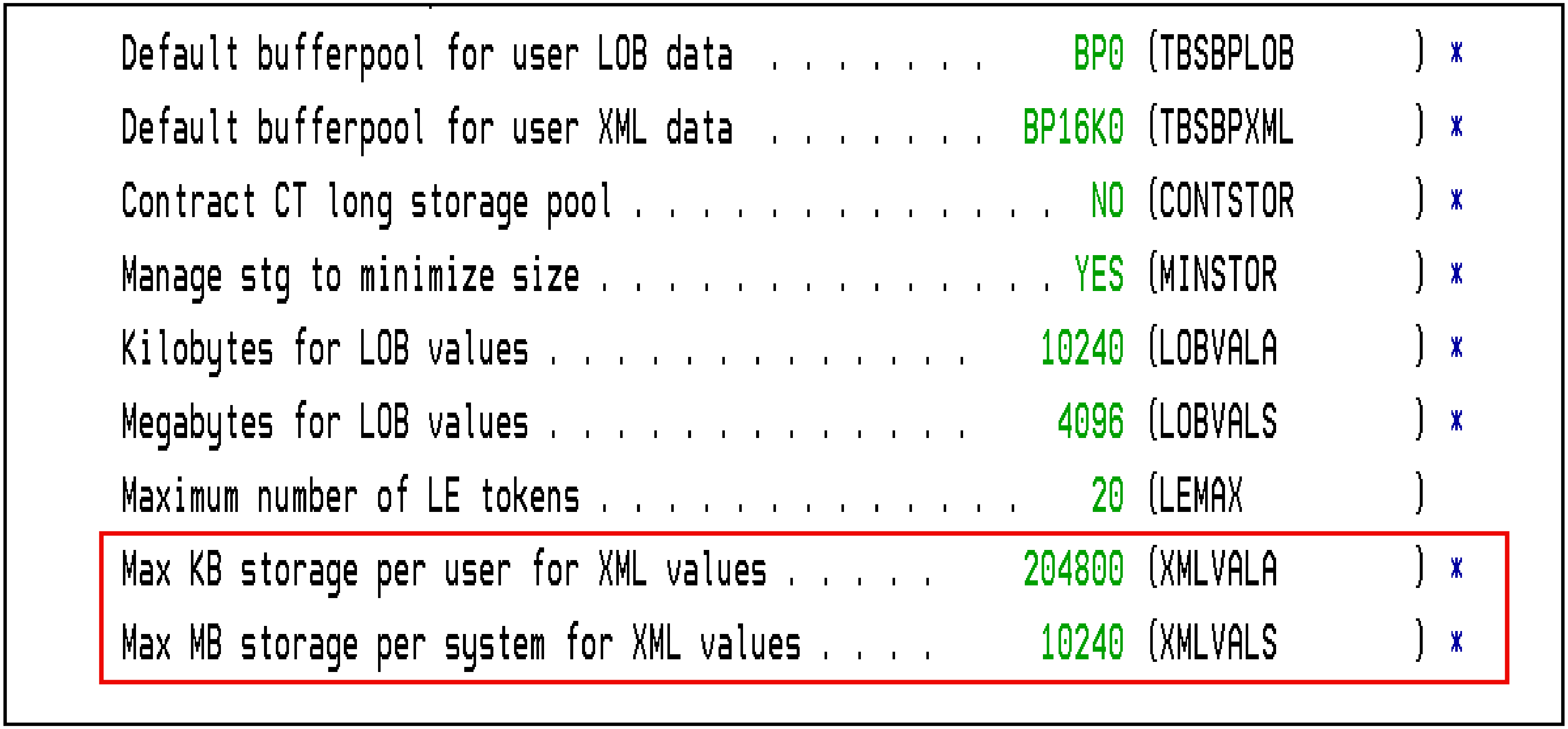
Restrict Virtual Storage use by LOB- zPARMS



```
Default bufferpool for user LOB data . . . . . . .    BP0 (TBSBPLOB    ) *
Default bufferpool for user XML data . . . . . . . BP16K0 (TBSBPXML    ) *
Contract CT long storage pool . . . . . . . . . . .    NO (CONTSTOR    ) *
Manage stg to minimize size . . . . . . . . . . . .   YES (MINSTOR     ) *
Kilobytes for LOB values . . . . . . . . . . . .    10240 (LOBVALA     ) *
Megabytes for LOB values . . . . . . . . . . . .     4096 (LOBVALS     ) *
Maximum number of LE tokens . . . . . . . . . . . .    20 (LEMAX       )
Max KB storage per user for XML values . . . . .   204800 (XMLVALA     ) *
Max MB storage per system for XML values . . . .    10240 (XMLVALS     ) *
```

LOBVALA – upper limit for amount of variable storage that each USER can have for storing LOB values
LOBVALS – upper limit for the amount of variable storage each system can have for storing LOB values in megabytes

Restrict Virtual Storage use by XML- zPARMS



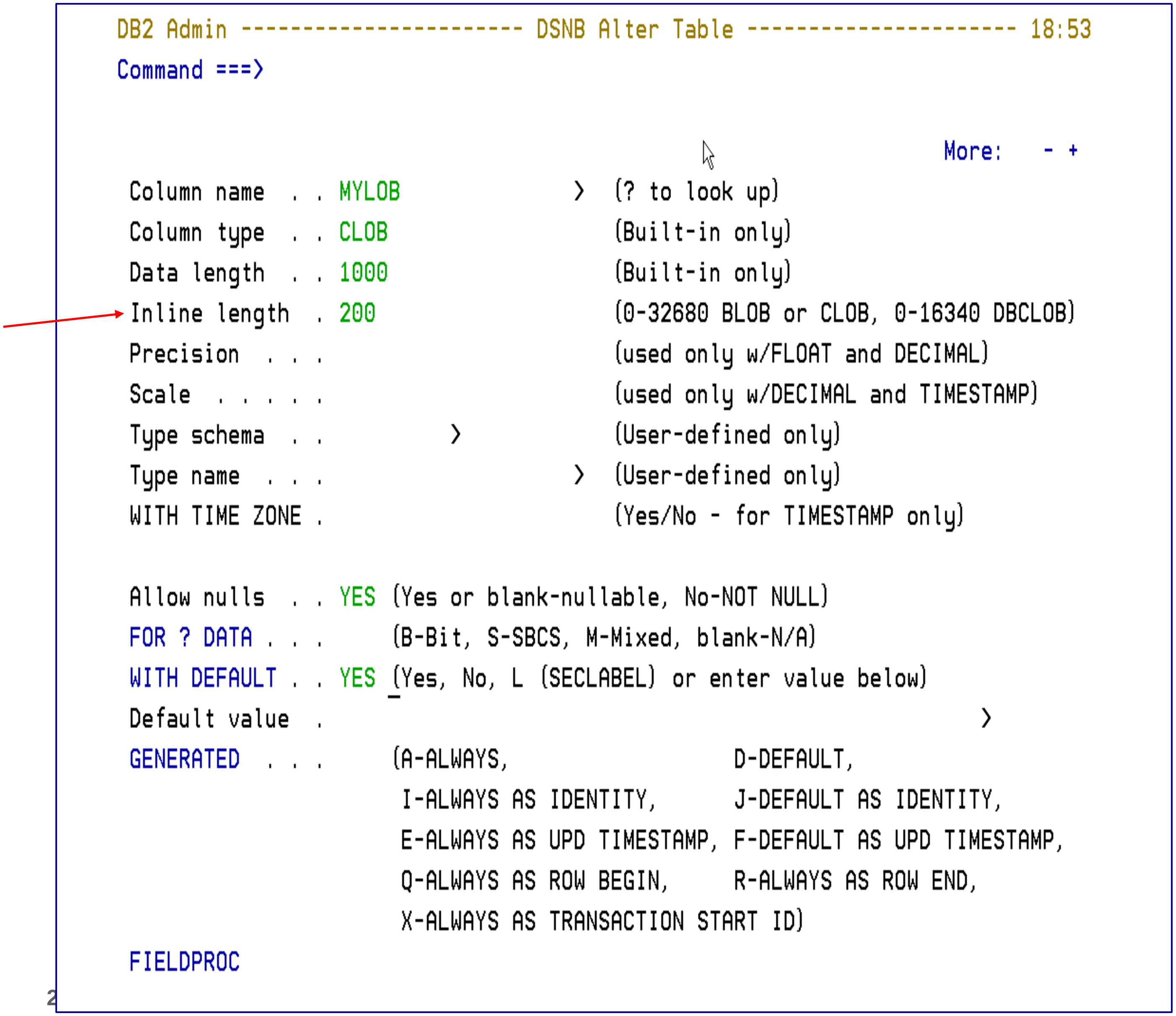
```
Default bufferpool for user LOB data . . . . . . .     BP0 (TBSBPLOB      ) *
Default bufferpool for user XML data . . . . . . .  BP16K0 (TBSBPXML      ) *
Contract CT long storage pool . . . . . . . . . . .     NO (CONTSTOR      ) *
Manage stg to minimize size . . . . . . . . . . . .    YES (MINSTOR       ) *
Kilobytes for LOB values . . . . . . . . . . . .     10240 (LOBVALA       ) *
Megabytes for LOB values . . . . . . . . . . . .      4096 (LOBVALS       ) *
Maximum number of LE tokens . . . . . . . . . . .       20 (LEMAX         )
Max KB storage per user for XML values . . . . .    204800 (XMLVALA       ) *
Max MB storage per system for XML values . . . .     10240 (XMLVALS       ) *
```

XMLVALA – upper limit for amount of memory that each thread can have for processing XML data (recommendation is to set at least 4x the max document size)
XMLVALS – upper limit for the amount of memory that each subsystem can have for processing XML data (set to max # threads * 2gb or XMLVALA)

Performance Enhancement of DB2 10- Inline LOB

- Prior: A LOB value is housed in a separate table and table space
- DB2 10: A portion of the LOB value can be stored with the base table data
- LOBs of “smallish” size may fit entirely
  - **Performance improvement**
- Inline portion can be used in an index on expression

Inline LOB example- using DB2 Admin Tool

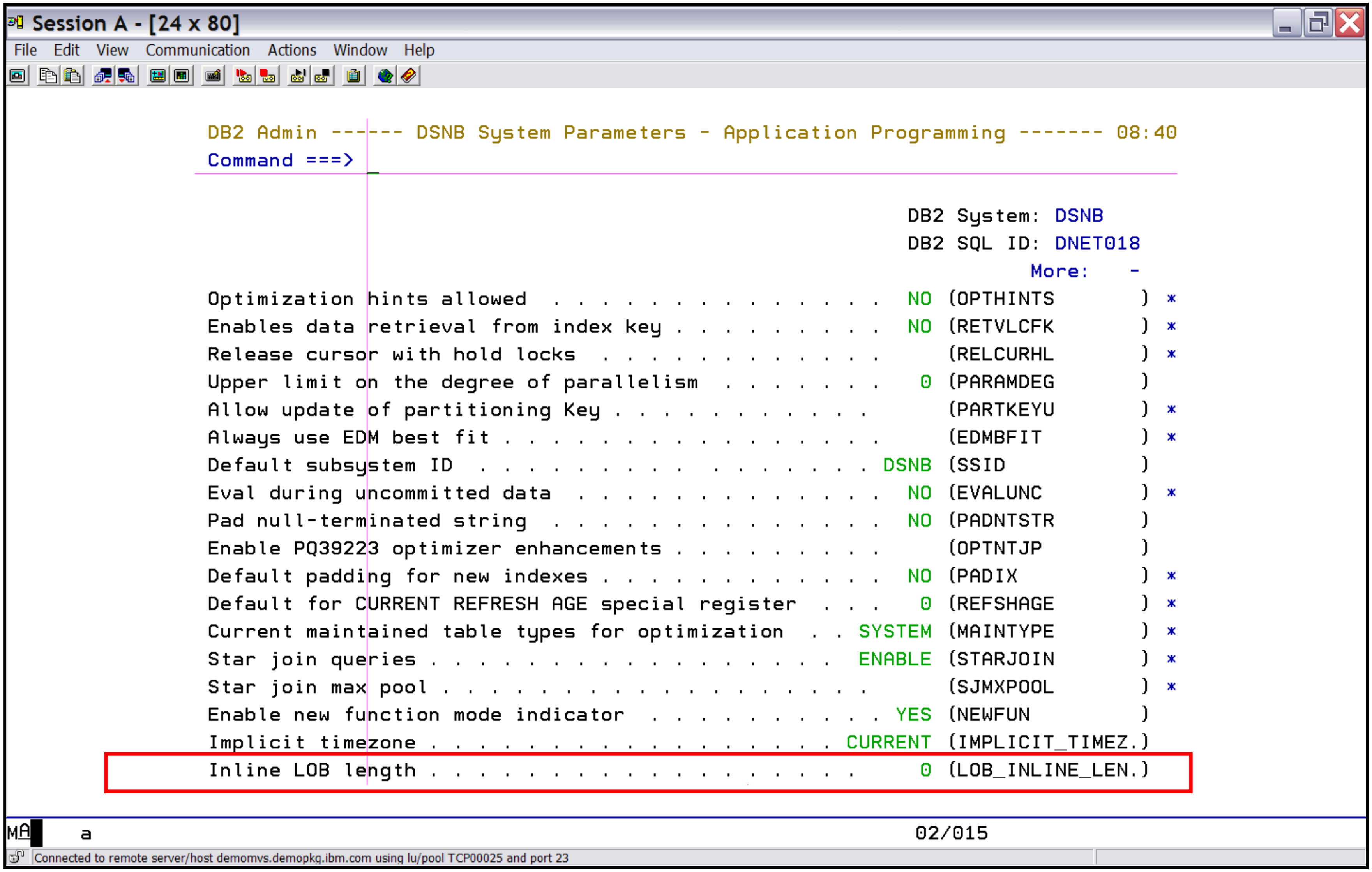


```
DB2 Admin ---------------------- DSNB Alter Table --------------------- 18:53
Command ===>

                                                          More:   - +
 Column name  . . MYLOB               > (? to look up)
 Column type  . . CLOB                  (Built-in only)
 Data length  . . 1000                  (Built-in only)
 Inline length  . 200                   (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
 Precision  . . .                       (used only w/FLOAT and DECIMAL)
 Scale  . . . . .                       (used only w/DECIMAL and TIMESTAMP)
 Type schema  . .           >           (User-defined only)
 Type name  . . .                     > (User-defined only)
 WITH TIME ZONE .                       (Yes/No - for TIMESTAMP only)

 Allow nulls  . . YES (Yes or blank-nullable, No-NOT NULL)
 FOR ? DATA . . .     (B-Bit, S-SBCS, M-Mixed, blank-N/A)
 WITH DEFAULT . . YES (Yes, No, L (SECLABEL) or enter value below)
 Default value  .                                                  >
 GENERATED  . . .     (A-ALWAYS,                D-DEFAULT,
                       I-ALWAYS AS IDENTITY,     J-DEFAULT AS IDENTITY,
                       E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP,
                       Q-ALWAYS AS ROW BEGIN,    R-ALWAYS AS ROW END,
                       X-ALWAYS AS TRANSACTION START ID)

 FIELDPROC
```

Alter table to convert to or add a LOB column

- •DB2 Object Comparison Tool
  - •Inline LOB column length can be changed using masking
  - •UDT with inline length can be compared
  - •Inline length can be ignored using ignore

DB2 Admin Tool V10.1 INLINE LOB



```
 DB2 Admin ------- DSNB System Parameters - Application Programming ------- 08:40
 Command ===> _

                                                        DB2 System: DSNB
                                                        DB2 SQL ID: DNET018
                                                               More:    -
 Optimization hints allowed  . . . . . . . . . . . . . . . NO (OPTHINTS        ) *
 Enables data retrieval from index key . . . . . . . . . . NO (RETVLCFK        ) *
 Release cursor with hold locks  . . . . . . . . . . . . .    (RELCURHL        ) *
 Upper limit on the degree of parallelism  . . . . . . . .  0 (PARAMDEG        )
 Allow update of partitioning Key . . . . . . . . . . . .     (PARTKEYU        ) *
 Always use EDM best fit . . . . . . . . . . . . . . . . .    (EDMBFIT         ) *
 Default subsystem ID  . . . . . . . . . . . . . . . . . DSNB (SSID            )
 Eval during uncommitted data  . . . . . . . . . . . . . . NO (EVALUNC         ) *
 Pad null-terminated string  . . . . . . . . . . . . . . . NO (PADNTSTR        )
 Enable PQ39223 optimizer enhancements . . . . . . . . . .    (OPTNTJP         )
 Default padding for new indexes . . . . . . . . . . . . . NO (PADIX           ) *
 Default for CURRENT REFRESH AGE special register  . . . .  0 (REFSHAGE        ) *
 Current maintained table types for optimization  . . SYSTEM (MAINTYPE        ) *
 Star join queries . . . . . . . . . . . . . . . . . . ENABLE (STARJOIN        ) *
 Star join max pool . . . . . . . . . . . . . . . . . . . .   (SJMXPOOL        ) *
 Enable new function mode indicator  . . . . . . . . . . . YES (NEWFUN         )
 Implicit timezone . . . . . . . . . . . . . . . . . . CURRENT (IMPLICIT_TIMEZ.)
 Inline LOB length . . . . . . . . . . . . . . . . . . . .   0 (LOB_INLINE_LEN.)

MA      a                                                        02/015
Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00025 and port 23
```

What’s Temporal Table

- Retain historical copies of a row –SYSTEM_TIME
  - **When updated, created and deleted**
- Maintain rows based upon business-determined periods - BUSINESS_TIME
- **Bi-temporal table – is both a System Period and Application Temporal Table**
- Can query data “as of” a point in time
- **One of the major improvements in DB2 10 is to reduce the**
  - complexity and amount of coding needed to implement “versioned” data, data that has different values at different points in time
- XML & LOB support temporal data.

Temporal tables - Example

```
CREATE TABLE EMPDB
(Empname VARCHAR (40),
Salary INTEGER,
SysTmSta TS(12) NN GENERATED ALWAYS AS ROW BEGIN,
SysTmEnd TS(12) NN GENERATED ALWAYS AS ROW END, ..
PERIOD SYSTEM_TIME (SysTmSta, SysTmEnd));
```

```
CREATE TABLE EMPHIST
(Empname VARCHAR (40),
Salary INTEGER,
SysTmSta TS(12)
SysTmEnd TS(12) .. )
;
```

Step	Date	Activity
1	6/15/2007	New Employee Hired
2	6/15/2008	Employee Gets Salary Raise
3	9/15/2008	Employee quits

```
ALTER TABLE ADD VERSIONING USE HISTOY TABLE  EMPHIST
```

Empname	Salary	SysTmSta	SysTmEnd

```
INSERT INTO EMPDB

VALUES ('John Smith', '75000')
```

Empname	Salary	SysTmSta	SysTmEnd	
John Smith	75000	6/15/2007	12/31/9999	Current Row

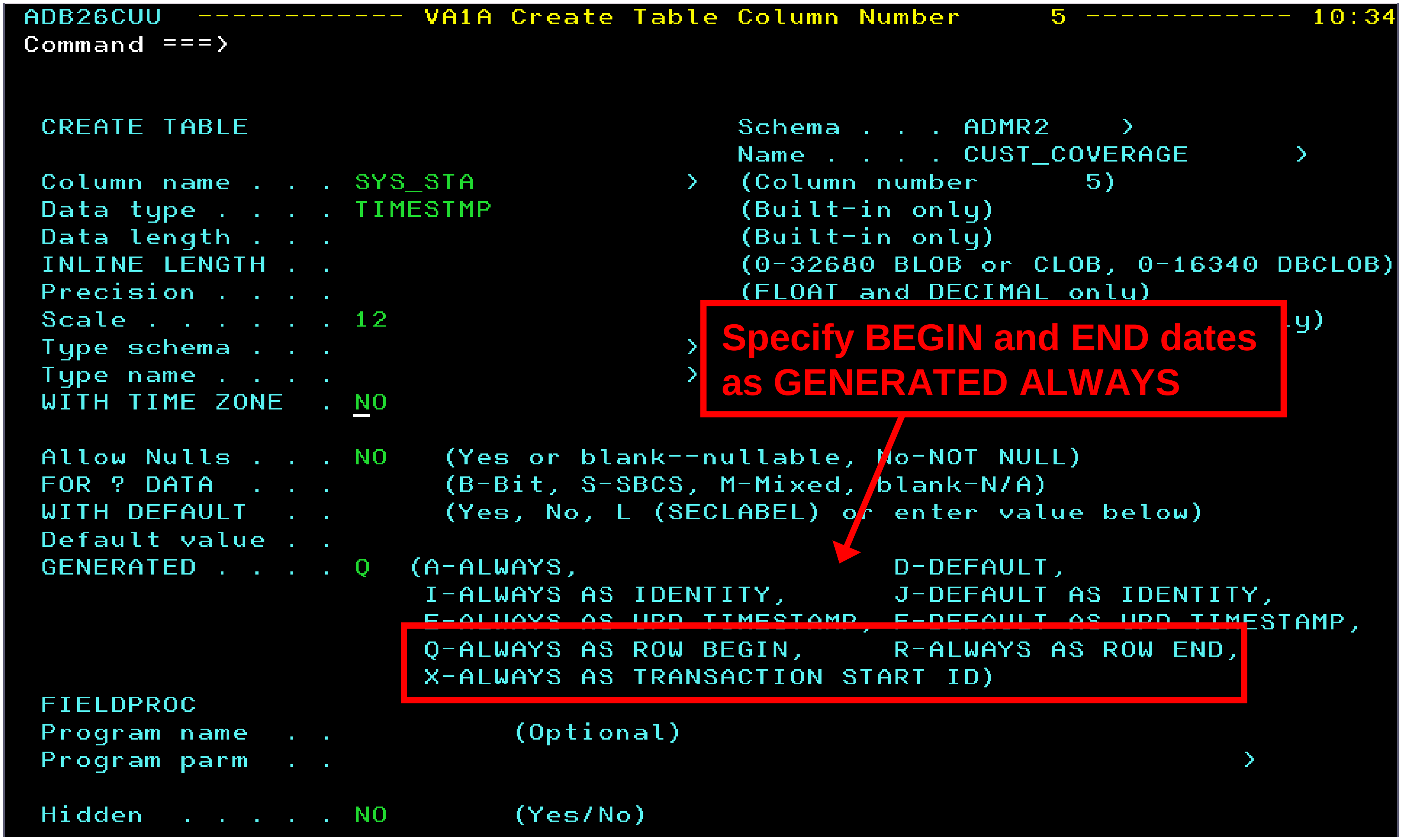
```
UPDATE EMPDB

SET Salary=Salary+5000
WHERE Empname='John Smith'
```

Empname	Salary	SysTmSta	SysTmEnd	
John Smith	75000	6/15/2007	6/15/2008	History Row
John Smith	80000	6/15/2008	12/31/9999	Current Row

```
DELETE FROM EMPDB

WHERE Empname = 'John Smith'
```

Empname	Salary	SysTmSta	SysTmEnd	
John Smith	75000	6/15/2007	6/15/2008	History Row
John Smith	80000	6/15/2008	9/15/2008	History Row

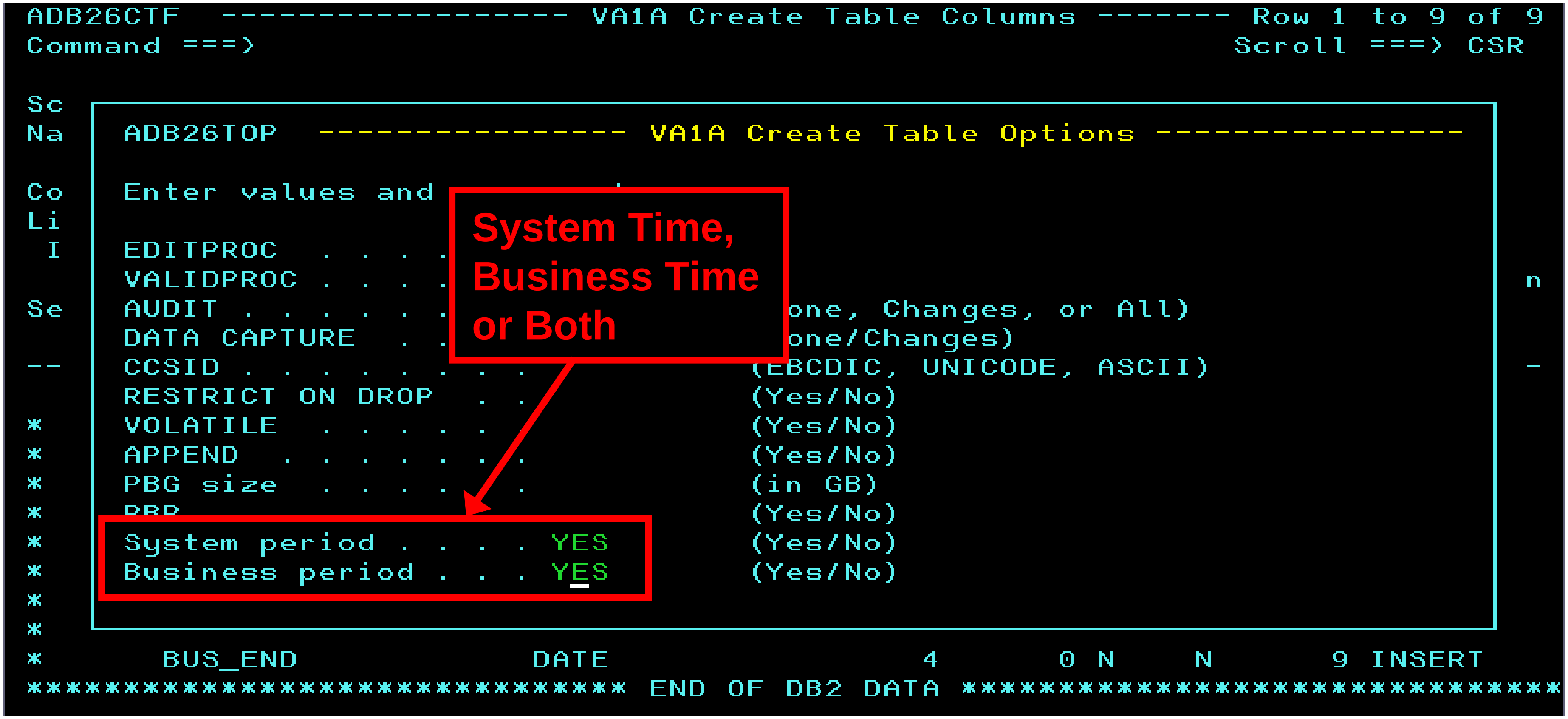
Create Temporal table -using Admin Tool- specify begin & end dates



```
ADB26CUU ------------ VA1A Create Table Column Number     5 ------------ 10:34
Command ===>


 CREATE TABLE                                  Schema . . . ADMR2     >
                                               Name . . . . CUST_COVERAGE      >
 Column name . . . SYS_STA                >    (Column number       5)
 Data type . . . . TIMESTMP                    (Built-in only)
 Data length . . .                             (Built-in only)
 INLINE LENGTH . .                             (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
 Precision . . . .                             (FLOAT and DECIMAL only)
 Scale . . . . . . 12                                                        y)
 Type schema . . .                        >
 Type name . . . .                        >
 WITH TIME ZONE  . NO

 Allow Nulls . . . NO    (Yes or blank--nullable, No-NOT NULL)
 FOR ? DATA  . . .       (B-Bit, S-SBCS, M-Mixed, blank-N/A)
 WITH DEFAULT  . .       (Yes, No, L (SECLABEL) or enter value below)
 Default value . .
 GENERATED . . . . Q  (A-ALWAYS,                    D-DEFAULT,
                       I-ALWAYS AS IDENTITY,        J-DEFAULT AS IDENTITY,
                       E-ALWAYS AS UPD TIMESTAMP,   F-DEFAULT AS UPD TIMESTAMP,
                       Q-ALWAYS AS ROW BEGIN,       R-ALWAYS AS ROW END,
                       X-ALWAYS AS TRANSACTION START ID)
 FIELDPROC
 Program name  . .           (Optional)
 Program parm  . .                                                      >

 Hidden  . . . . . NO        (Yes/No)
```

Specify BEGIN and END dates as GENERATED ALWAYS

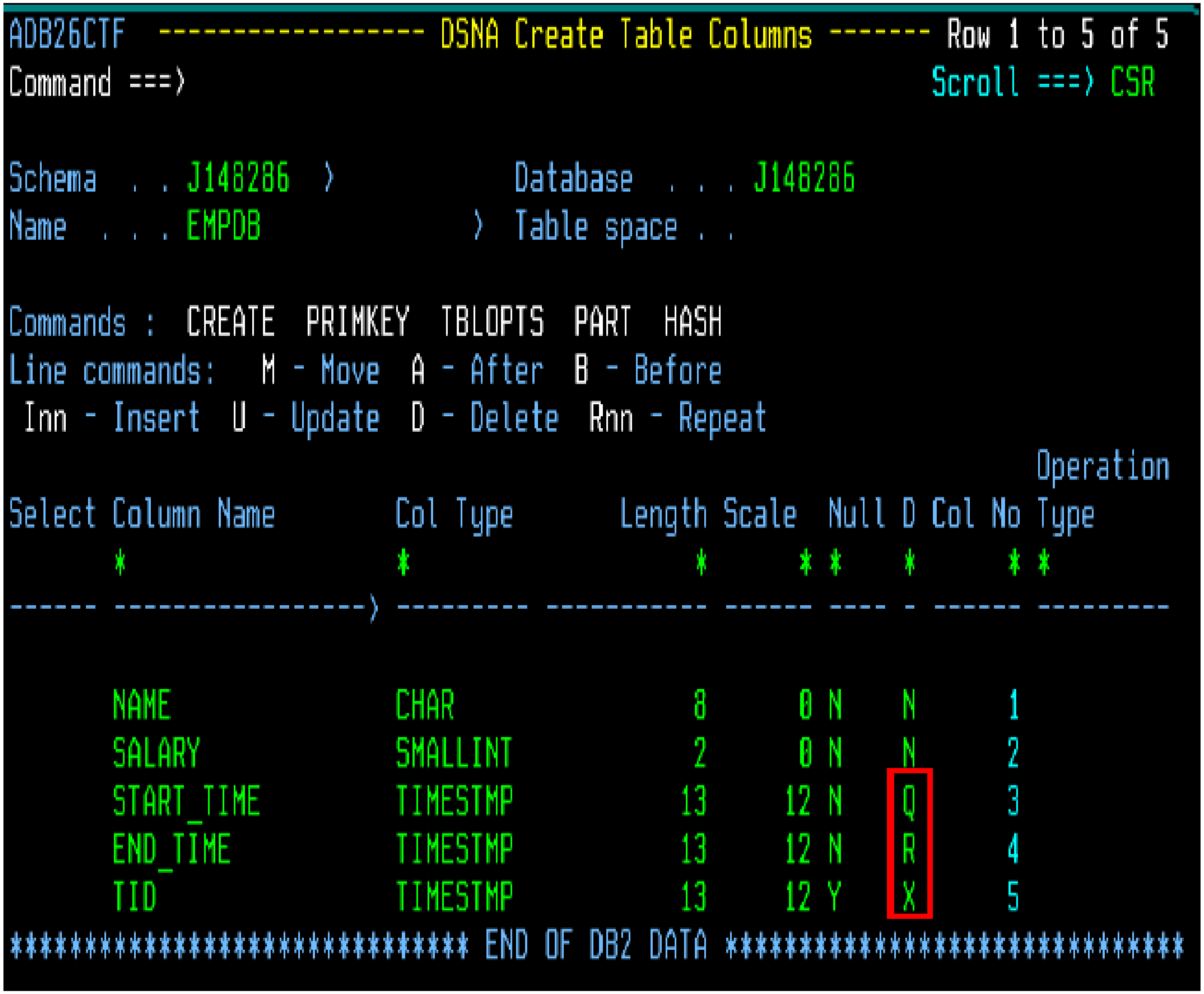
Create Temporal table -using Admin Tool- specify system or business period



```
ADB26CTF ------------------ VA1A Create Table Columns -------- Row 1 to 9 of 9
Command ===>                                                Scroll ===> CSR

Sc
Na   ADB26TOP ---------------- VA1A Create Table Options -----------------

Co   Enter values and           .
Li
 I   EDITPROC . . . .
     VALIDPROC . . . .                                                      n
Se   AUDIT . . . . . .                 one, Changes, or All)
     DATA CAPTURE  . .                 one/Changes)
--   CCSID . . . . . . . .           (EBCDIC, UNICODE, ASCII)               -
     RESTRICT ON DROP  . .           (Yes/No)
*    VOLATILE . . . . . .            (Yes/No)
*    APPEND  . . . . . . .           (Yes/No)
*    PBG size  . . . . . .           (in GB)
*    PBR                             (Yes/No)
*    System period . . . . YES       (Yes/No)
*    Business period . . . YES       (Yes/No)
*
*
*      BUS_END             DATE                4      0 N    N      9 INSERT
******************************** END OF DB2 DATA ********************************
```

System Time, Business Time or Both

Create Temporal tables –Example- Admin Tool



```
ADB26CTF  ------------------ DSNA Create Table Columns ------- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR

Schema . . . J148286  >            Database . . . J148286
Name . . . . EMPDB               > Table space . .

Commands :  CREATE  PRIMKEY  TBLOPTS  PART  HASH
Line commands:   M - Move  A - After  B - Before
 Inn - Insert  U - Update  D - Delete  Rnn - Repeat
                                                                   Operation
Select Column Name        Col Type       Length Scale  Null D Col No Type
       *                  *                   *      * *   *      * *
------ -----------------> --------- ----------- ------ ---- - ------ ---------

       NAME               CHAR                8      0 N   N      1
       SALARY             SMALLINT            2      0 N   N      2
       START_TIME         TIMESTMP           13     12 N   Q      3
       END_TIME           TIMESTMP           13     12 N   R      4
       TID                TIMESTMP           13     12 Y   X      5
****************************** END OF DB2 DATA ******************************
```

The base table has columns with temporal attributes

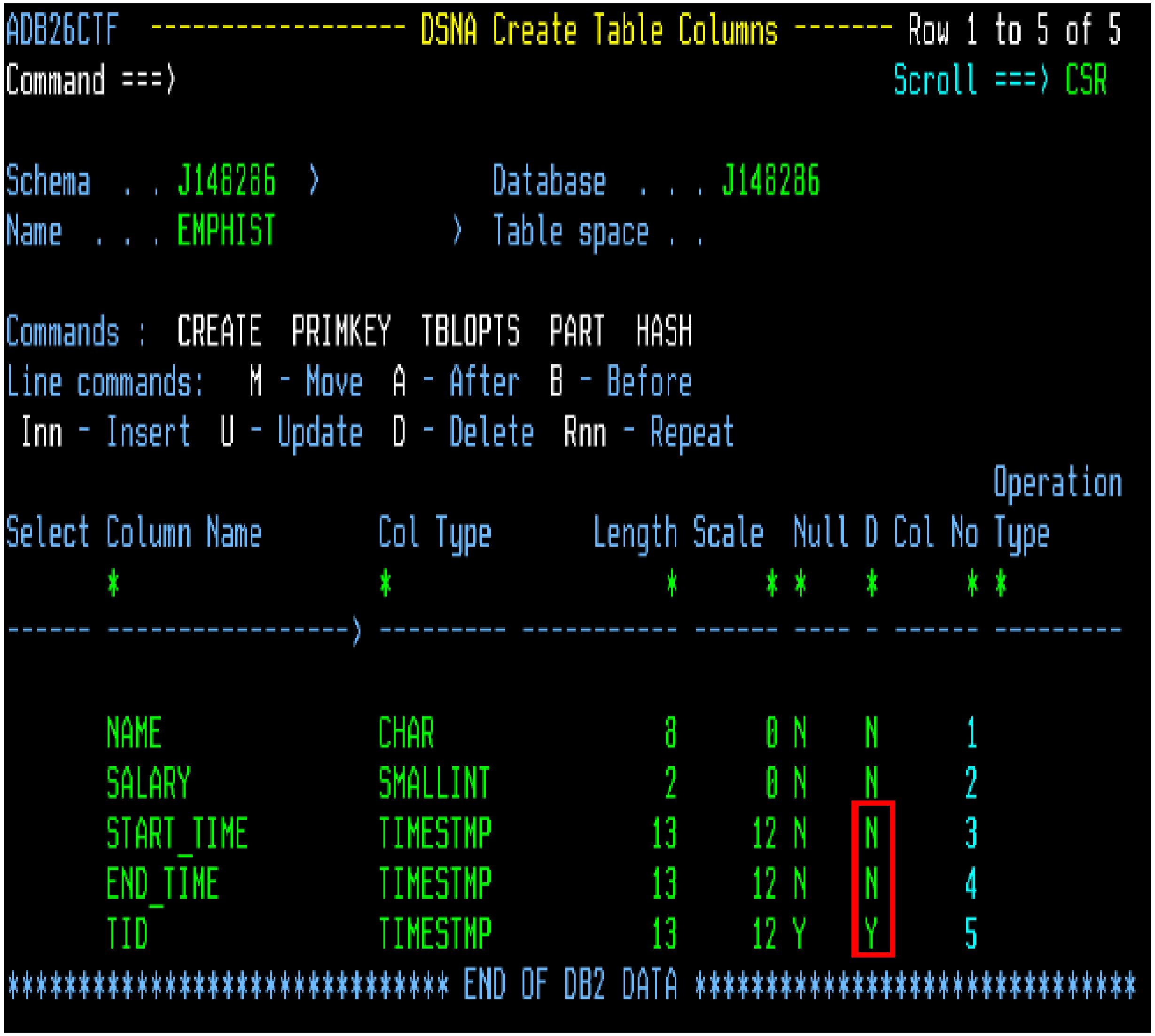
Q –AS ROW BEGIN

R –AS ROW END

X –AS TRANSACTION START ID

Any System Time columns can be defined as Implicitly Hidden

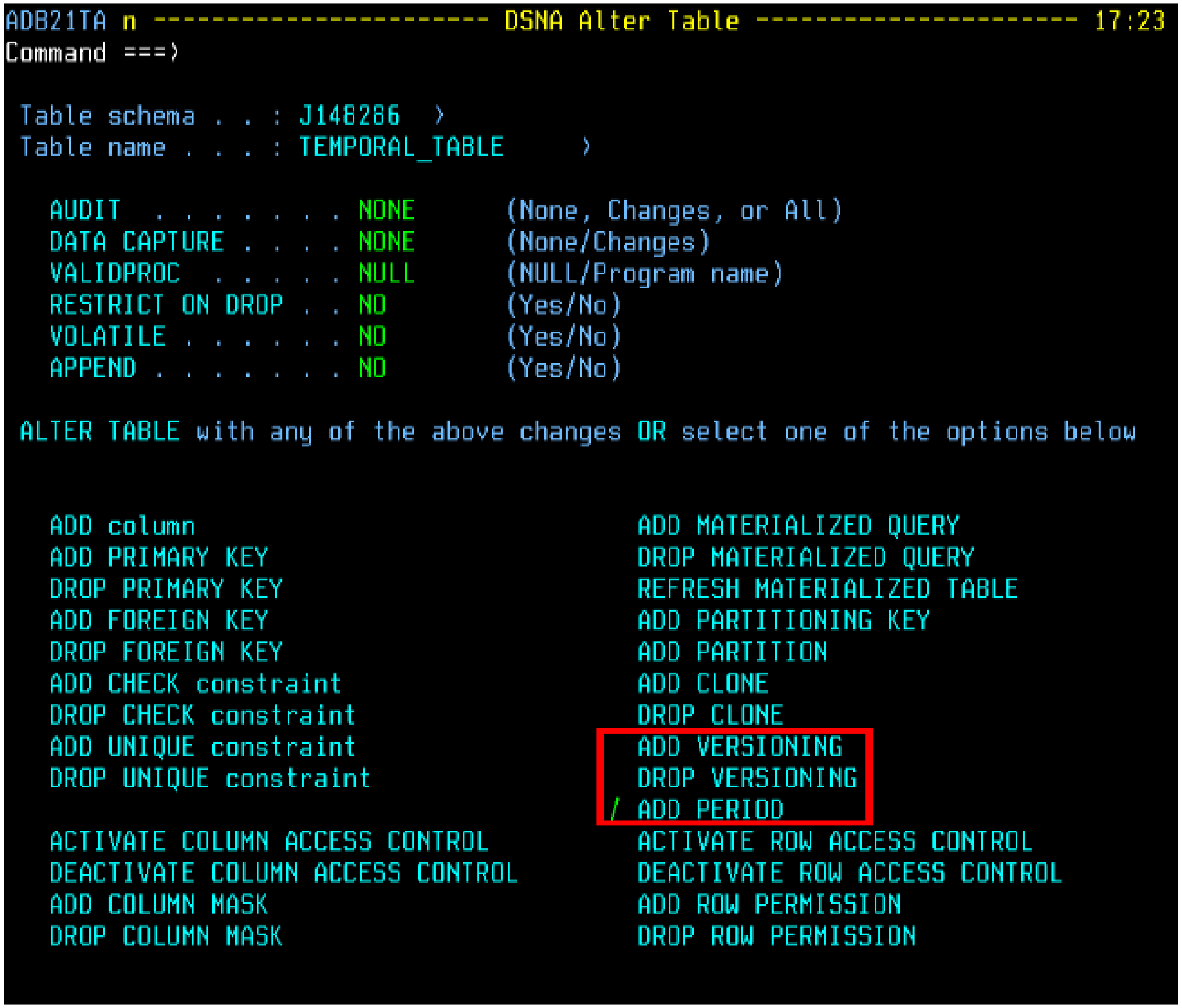
Create Example- Temporal History Table using Admin Tool



```
ADB26CTF  ------------------ DSNA Create Table Columns ------- Row 1 to 5 of 5
Command ===>                                                 Scroll ===> CSR

Schema . . J148286  >            Database . . . J148286
Name . . . EMPHIST          >  Table space . .

Commands : CREATE  PRIMKEY  TBLOPTS  PART  HASH
Line commands:  M - Move  A - After  B - Before
 Inn - Insert  U - Update  D - Delete  Rnn - Repeat
                                                                 Operation
Select Column Name        Col Type       Length Scale  Null D Col No Type
       *                  *                    *      * *   *      * *
------ ----------------> --------- ---------- ------ ---- - ------ ---------

       NAME               CHAR                 8      0 N   N      1
       SALARY             SMALLINT             2      0 N   N      2
       START_TIME         TIMESTMP            13     12 N   N      3
       END_TIME           TIMESTMP            13     12 N   N      4
       TID                TIMESTMP            13     12 Y   Y      5
******************************* END OF DB2 DATA ******************************
```

- The history table has the same columns but without temporal attributes

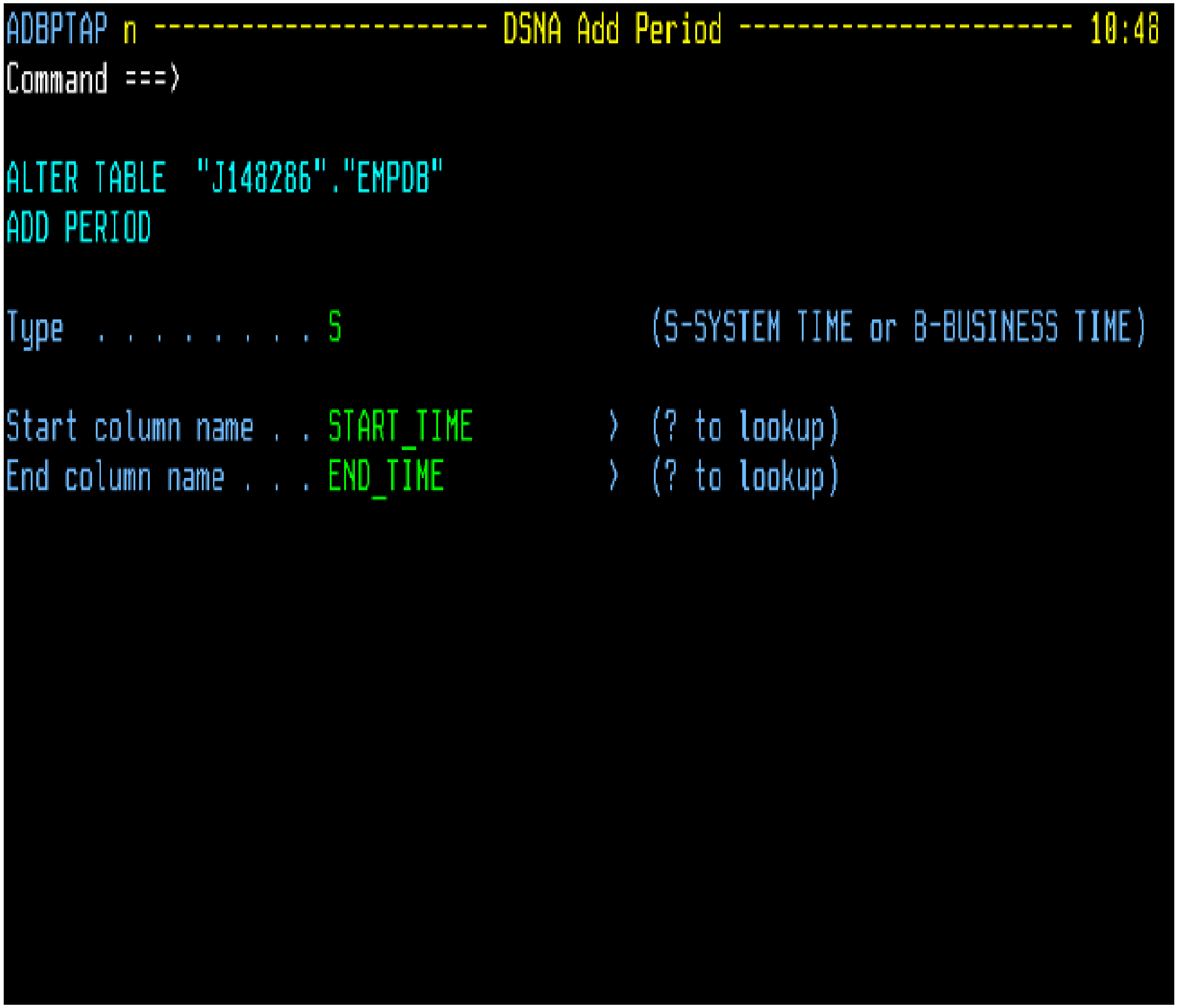
Alter table for Temporal Table –Admin Tool



```
ADB21TA n ----------------------- DSNA Alter Table ---------------------- 17:23
Command ===>

 Table schema . . : J148286   >
 Table name . . . : TEMPORAL_TABLE     >

   AUDIT  . . . . . . . . NONE       (None, Changes, or All)
   DATA CAPTURE . . . . . NONE       (None/Changes)
   VALIDPROC  . . . . . . NULL       (NULL/Program name)
   RESTRICT ON DROP . . . NO         (Yes/No)
   VOLATILE . . . . . . . NO         (Yes/No)
   APPEND . . . . . . . . NO         (Yes/No)

 ALTER TABLE with any of the above changes OR select one of the options below


   ADD column                               ADD MATERIALIZED QUERY
   ADD PRIMARY KEY                          DROP MATERIALIZED QUERY
   DROP PRIMARY KEY                         REFRESH MATERIALIZED TABLE
   ADD FOREIGN KEY                          ADD PARTITIONING KEY
   DROP FOREIGN KEY                         ADD PARTITION
   ADD CHECK constraint                     ADD CLONE
   DROP CHECK constraint                    DROP CLONE
   ADD UNIQUE constraint                    ADD VERSIONING
   DROP UNIQUE constraint                   DROP VERSIONING
                                          / ADD PERIOD
   ACTIVATE COLUMN ACCESS CONTROL           ACTIVATE ROW ACCESS CONTROL
   DEACTIVATE COLUMN ACCESS CONTROL         DEACTIVATE ROW ACCESS CONTROL
   ADD COLUMN MASK                          ADD ROW PERMISSION
   DROP COLUMN MASK                         DROP ROW PERMISSION
```

- Alter the table
- ADD PERIOD
  - Add the SYSTEM_TIME period attribute
- ADD VERSIONING
  - Begins versioning the data on DML
- DROP VERSIONING
  - Removes versioning of data

Alter Table to add Period (system time) for Temporal Data from Admin Tool

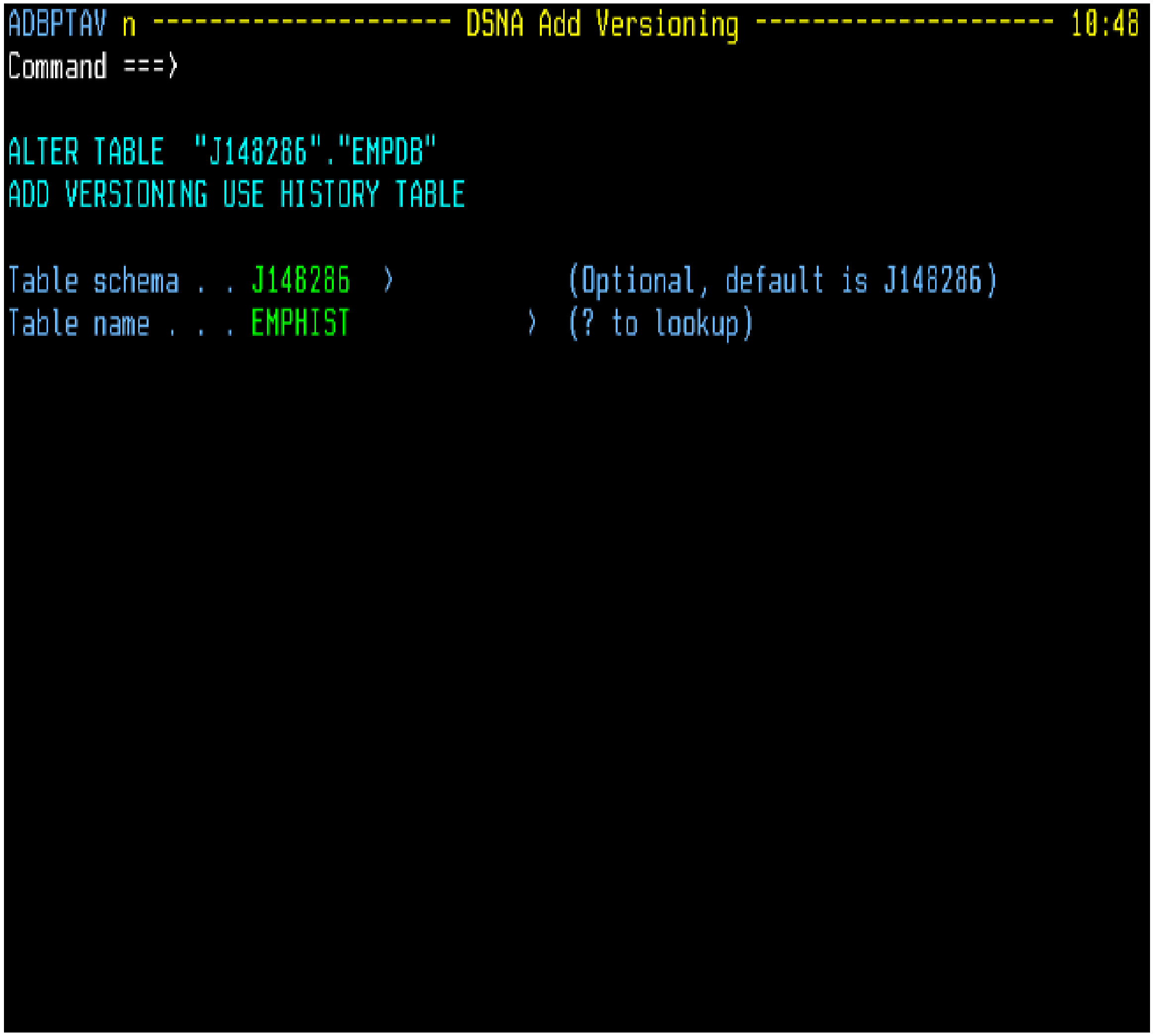


```
ADBPTAP n ----------------------- DSNA Add Period ----------------------- 10:48
Command ===>

ALTER TABLE  "J148266"."EMPDB"
ADD PERIOD

Type  . . . . . . . . . S                    (S-SYSTEM TIME or B-BUSINESS TIME)

Start column name . . START_TIME         >  (? to lookup)
End column name . . . END_TIME           >  (? to lookup)
```

- Alter the table

- ADD PERIOD to add the SYSTEM_TIME or BUSINESS_TIME period

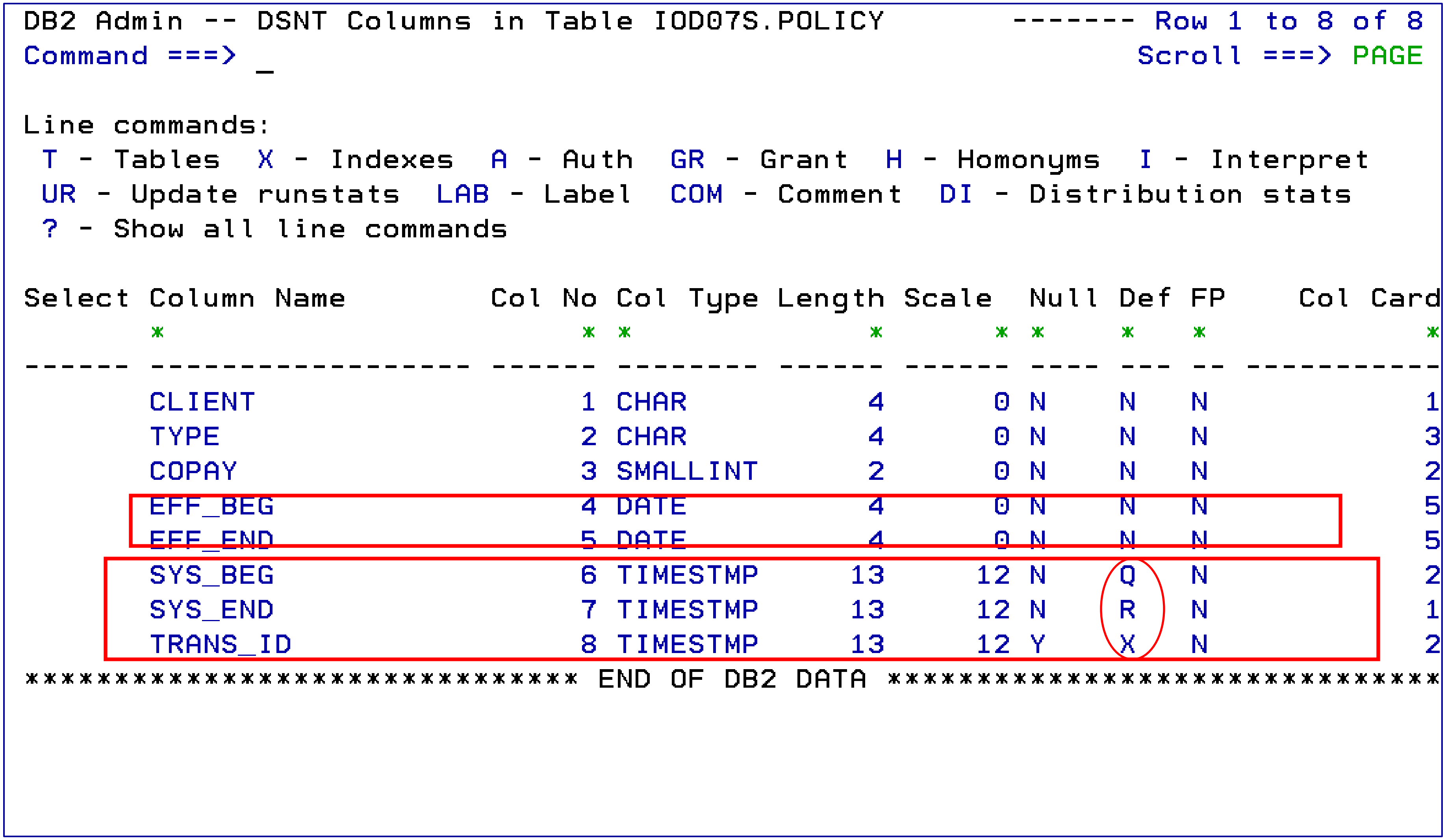
Alter Table to add Versioning to associate Temporal Data with History table -Admin Tool



```
ADBPTAV n -------------------- DSNA Add Versioning -------------------- 10:48
Command ===>

ALTER TABLE  "J148286"."EMPDB"
ADD VERSIONING USE HISTORY TABLE

Table schema . . J148286  >           (Optional, default is J148286)
Table name . . . EMPHIST            > (? to lookup)
```

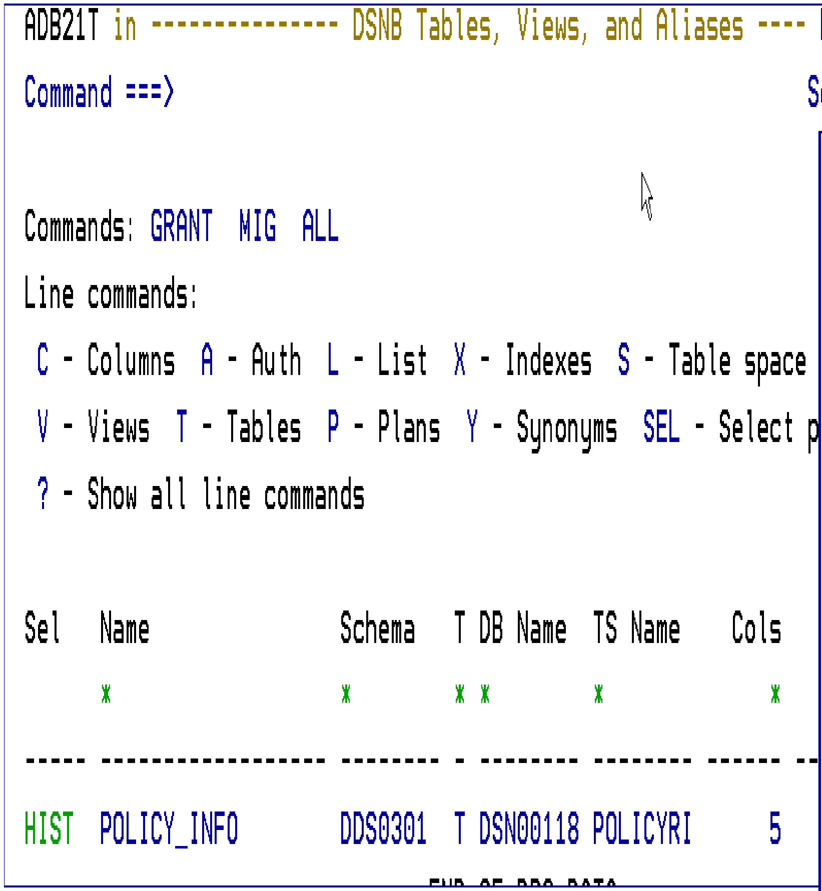
- Alter the table
- ADD VERSIONING
- Choose the table to become the history table

Display of columns for the Bi-Temporal Table-Admin Tool

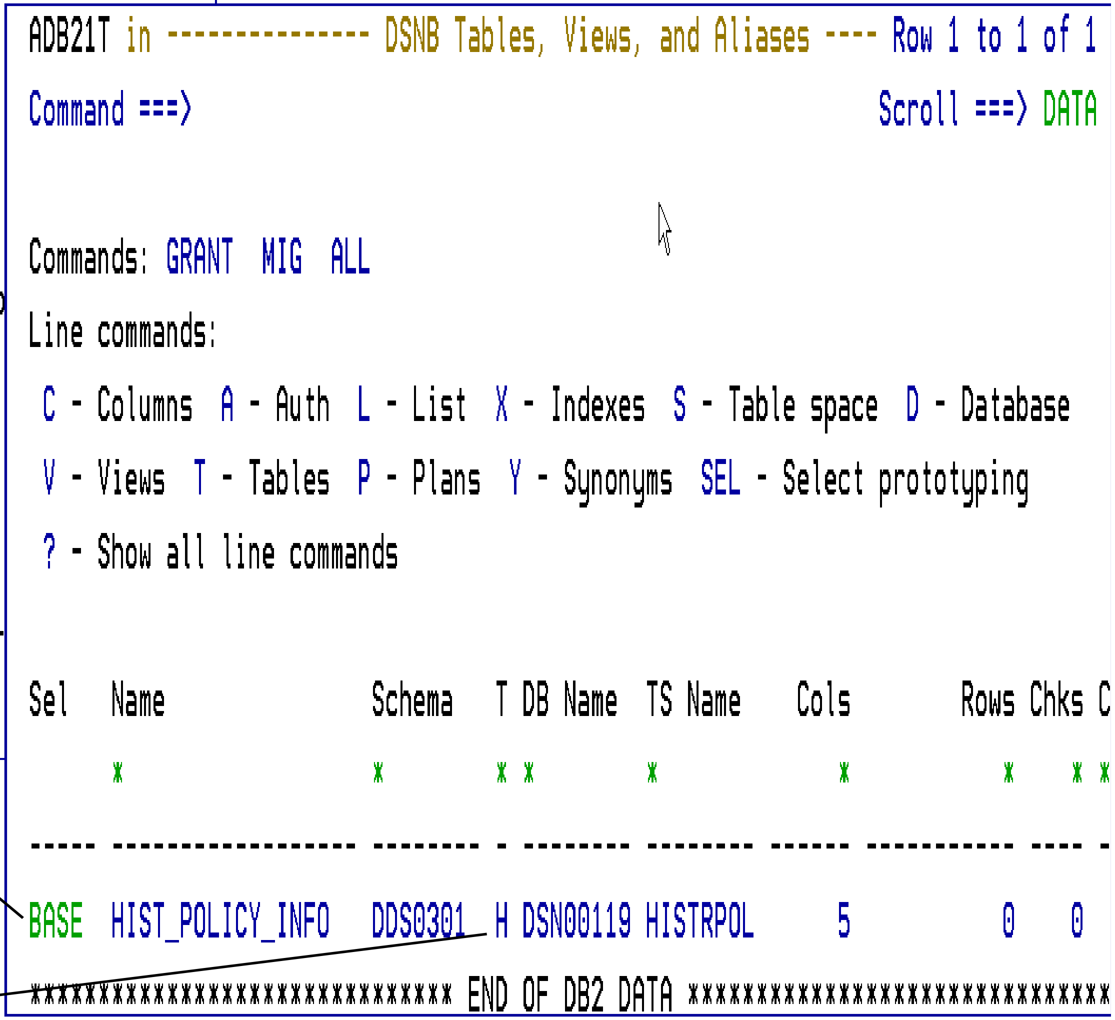


```
DB2 Admin -- DSNT Columns in Table IOD07S.POLICY       ------- Row 1 to 8 of 8
Command ===> _                                                Scroll ===> PAGE

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 ? - Show all line commands

Select Column Name        Col No Col Type Length Scale  Null Def FP    Col Card
       *                       * *             *      * *    *   *            *
------ ------------------ ------ -------- ------ ------ ---- --- -- -----------
       CLIENT                  1 CHAR          4      0 N    N   N            1
       TYPE                    2 CHAR          4      0 N    N   N            3
       COPAY                   3 SMALLINT      2      0 N    N   N            2
       EFF_BEG                 4 DATE          4      0 N    N   N            5
       EFF_END                 5 DATE          4      0 N    N   N            5
       SYS_BEG                 6 TIMESTMP     13     12 N    Q   N            2
       SYS_END                 7 TIMESTMP     13     12 N    R   N            1
       TRANS_ID                8 TIMESTMP     13     12 Y    X   N            2
******************************* END OF DB2 DATA *******************************
```

Display associated Temporal history table from Admin Tool



```
ADB21T in -------------- DSNB Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> DATA

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select p
 ? - Show all line commands

Sel   Name               Schema   T DB Name  TS Name    Cols
      *                  *        * *        *             *
----- ------------------ -------- - -------- -------- ------ --
HIST  POLICY_INFO        DDS0301  T DSN00118 POLICYRI      5
                                  END OF DB2 DATA
```



```
ADB21T in -------------- DSNB Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> DATA

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C
      *                  *        * *        *             *           *    * *
----- ------------------ -------- - -------- -------- ------ ----------- ---- -
BASE  HIST_POLICY_INFO   DDS0301  H DSN00119 HISTRPOL      5           0    0
******************************** END OF DB2 DATA ********************************
```

BASE- Display associated base table

H - History table

Benefits of using temporal tables …

- **Move the logic from application layer to database layer**
- Consistent handling of temporal data
- **Reduce Application development time by up to 10x**
- **Run current applications with no code change**
- **Preserve execution time for current queries going after**
  - current data (System Time)
- Business Problems you can solve with temporal tables
  - Ensure that a customer only has one financial position at a given time
    - Was an insured covered for a procedure on a specific date?
    - Was that information correct at the time the claim was processed?
  - Answer a customer complaint about an old bill
  - … and many, many more

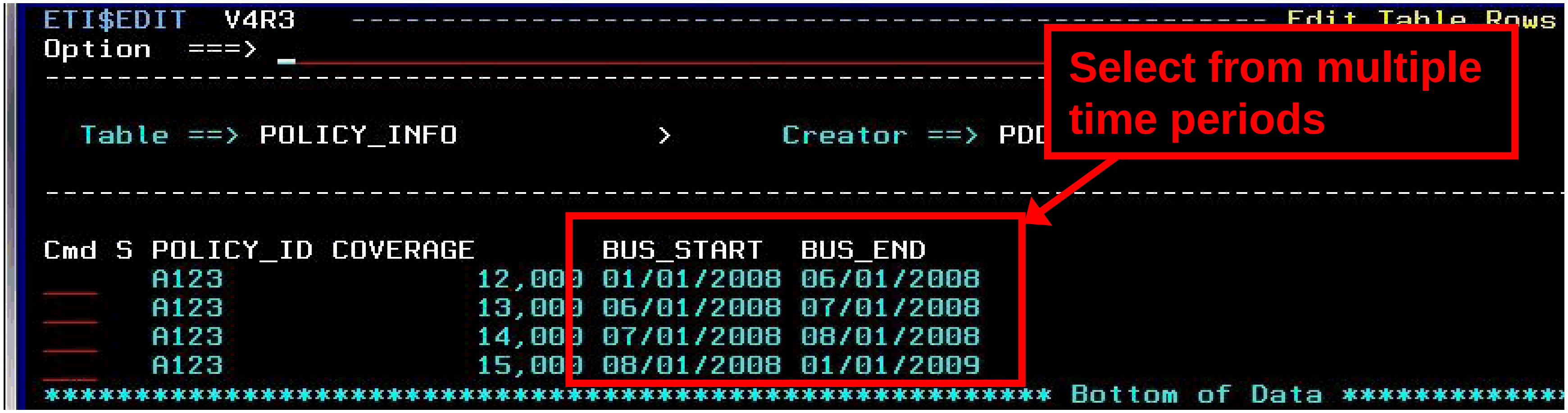
DB2 Admin Tool V10.1 Temporal Table

What Customers’ Appreciate about DB2 System Time and Versioning

- Simplicity of enabling History generation in DB2 10
    - DDL change to add system time begin and end columns, TranID begin time
    - Define schema matching history table
    - Hook up current and history tables
- Non-disruptive Enablement
    - Existing applications continues to work on current table
- Flexibility for History tables
    - may be managed differently
    - may be indexed differently

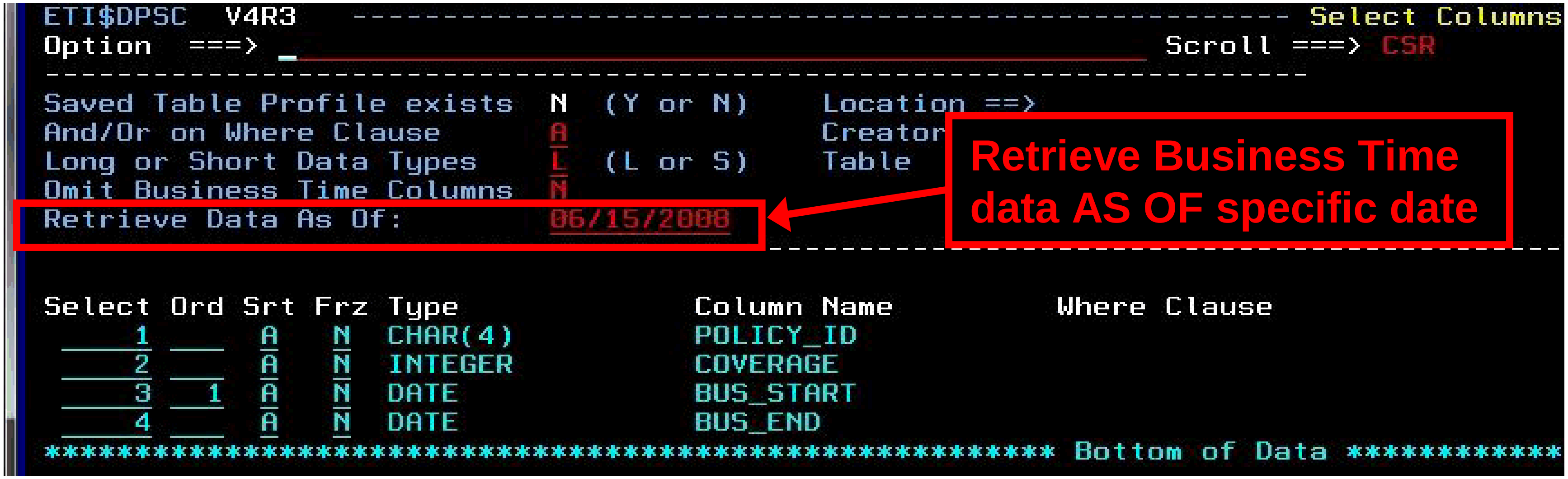
- Customers can **satisfy new compliance laws** faster and cheaper
- **For clients with existing history and history generation processes**
    - Simplification of the process of history generation – both enablement and maintenance
    - Preservation of investment –alter history tables into DB2 history tables

Browsing/Editing Business Time data “AS OF” with DB2 Table Editor



```
ETI$EDIT   V4R3   ---------------------------------------------------- Edit Table Rows
Option  ===> _
---------------------------------------------------------------------------------

  Table ==> POLICY_INFO                >      Creator ==> PDD

---------------------------------------------------------------------------------

Cmd S POLICY_ID COVERAGE          BUS_START  BUS_END
      A123                 12,000 01/01/2008 06/01/2008
      A123                 13,000 06/01/2008 07/01/2008
      A123                 14,000 07/01/2008 08/01/2008
      A123                 15,000 08/01/2008 01/01/2009
********************************************************* Bottom of Data *************
```

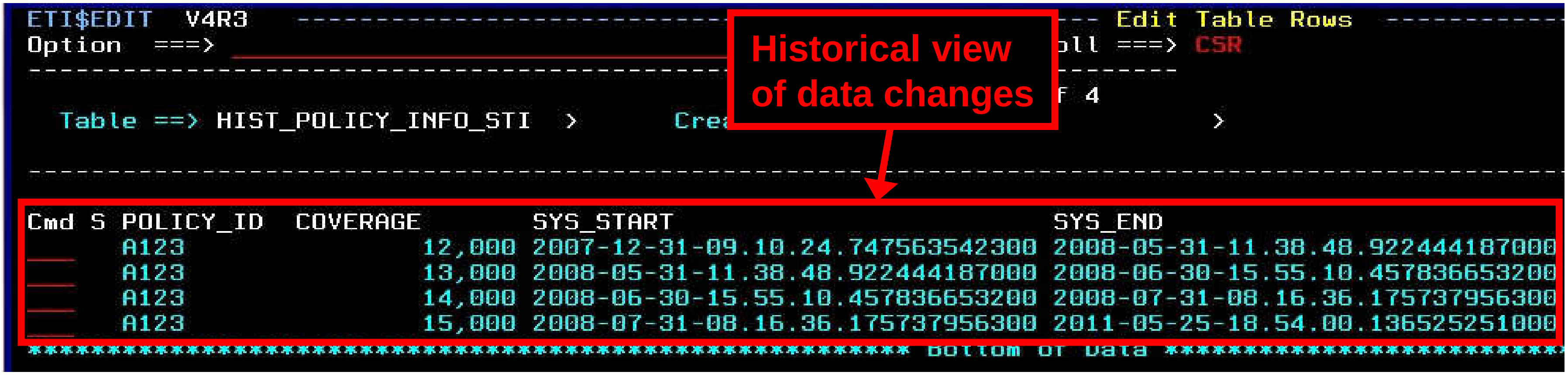
Select from multiple time periods



```
ETI$DPSC   V4R3   ---------------------------------------------------- Select Columns
Option  ===> _                                                   Scroll ===> CSR
----------------------------------------------------------------------
Saved Table Profile exists    N  (Y or N)    Location ==>
And/Or on Where Clause        A              Creator
Long or Short Data Types      L  (L or S)    Table
Omit Business Time Columns    N
Retrieve Data As Of:          06/15/2008
                                         --------------------------------------------

Select  Ord  Srt  Frz  Type            Column Name          Where Clause
     1        A    N   CHAR(4)         POLICY_ID
     2        A    N   INTEGER         COVERAGE
     3     1  A    N   DATE            BUS_START
     4        A    N   DATE            BUS_END
********************************************************* Bottom of Data *************
```

Retrieve Business Time data AS OF specific date

DB2 Table Editor - View and Edit System Time Data



```
ETI$EDIT  V4R3  ---------------------------                  --- Edit Table Rows ------------
Option  ===> ________________________________                oll ===> CSR
-------------------------------------------------            --------
                                                             f 4
  Table ==> HIST_POLICY_INFO_STI  >       Cre                                >

---------------------------------------------------------------------------------------------
Cmd S POLICY_ID  COVERAGE      SYS_START                        SYS_END
      A123             12,000 2007-12-31-09.10.24.747563542300 2008-05-31-11.38.48.922444187000
      A123             13,000 2008-05-31-11.38.48.922444187000 2008-06-30-15.55.10.457836653200
      A123             14,000 2008-06-30-15.55.10.457836653200 2008-07-31-08.16.36.175737956300
      A123             15,000 2008-07-31-08.16.36.175737956300 2011-05-25-18.54.00.136525251000
******************************************************** Bottom of Data **************************
```

Historical view of data changes

DB2 Table Editor can be used to edit/browse LOB/XML data as well.

DB2 Admin Tool will only show first 256 bytes of data from XML/LOB.

Temporal Table-System Period Versioning Considerations

- Base and History tables must be RECOVERed as a set
  - VERIFYSET NO can override the need to RECOVER together
- No utility operations that deletes data from base table
  - LOAD REPLACE
  - REORG DISCARD
  - CHECK DATA DELETE YES
- No CHECK utilities that invalidate AUX/LOB/XML
- Cannot ALTER the schema while versioning
- No temporal SELECT, UPDATE, or DELETE against the History
- Cannot be an MQT
- Cannot have a Clone Table, Column Mask, Row Permission

DB2 Utilities Suite 10

- COPY(Full, Incremental, Concurrent, FlashCopy), RECOVER (with VERIFYSET, ENFORCE, BACKOUT), REORG with AUX YES/NO, RUNSTATS & autonomic statistics
- CHECK DATA /INDEX /LOB with SHRLEVEL CHANGE, REPAIR, REPORT Table
- LOAD and UNLOAD, with file reference variable & spanned format
- Load with PERIODOVERRIDE and TRANSIDOVERRIDE to reload into temporal table columns that are defined as GENERATED ALWAYS.
- LISTDEF ALL to support all related objects including XML/LOB & HISTORY option to support History for Temporal data
- **Restriction:**
  - **DSN1COPY can't be** used to copy XML table spaces from one subsystem to another, since DB2 XML data is condensed by substituting strings by unique IDs which are stored in SYSIBM.SYSXMLSTRINGS and are not available in the XML table space. (Use Migrate function of DB2 Admin Tool to copy DDL, Data or Runstats)

Unloading a LOB/XML

- Application
  - – Host Variable
    - Enough application storage to retrieve the entire document
  - – LOB Locators for LOB (extracting in pieces)
  - – SQL FETCH WITH CONTINUE & FETCH CURRENT CONTINUE
  - – File Reference Variable
- DB2 UNLOAD Utility or HPU (High performance Unload)
  - – Normal Output record along with the data from non-LOB/XML, can't be > 32K
  - – SPANNED format in DB2 10, can be of any size
  - – File Reference Variable with Template
  - – Always unload from the base table

DB2 9 File Reference Variables

LOB / XML

- **Used to <u>import / export</u> data between a LOB/XML column and an external file outside of DB2**
- **Can be used to retrieve an entire LOB/XML.**
- **Use less CPU and avoid using application storage**
- **Bypasses any restrictions of the program language**
- **3 types**
  - BLOB_FILE
  - CLOB_FILE
  - DBCLOB_FILE
- **Supported by LOAD / UNLOAD**
- **External files can be PDS/PDSE or HSF files**
- **Application**

UNLOAD SYNTAX

LOB & XML

```
TEMPLATE LOBFRV1 DSN 'UN.&DB..&TS..RESUME'
   DSNTYPE(PDS) UNIT(SYSDA)
TEMPLATE LOBFRV2 DSN 'UN.&DB..&TS..PHOTO'
   DSNTYPE(PDS) UNIT(SYSDA)
UNLOAD DATA FROM TABLE DSN8910.EMP_PHOTO_RESUME
 (EMPNO CHAR(6),
 RESUME VARCHAR(255) CLOBF LOBFRV1,
  PHOTO VARCHAR(255) BLOBF LOBFRV2)
 SHRLEVEL CHANGE
```

OUTPUT

```
"000001","UN.DB1.TS1.RESUME(AI3WX3JT)","UN.DB1.TS1.PHOTO(AI3WX3JT)"
"000002","UN.DB1.TS1.RESUME(AI3WX5BS)","UN.DB1.TS1.PHOTO(AI3WX5BS)"
"000003","UN.DB1.TS1.RESUME(AI3WX5CC)","UN.DB1.TS1.PHOTO(AI3WX5CC)"
"000004","UN.DB1.TS1.RESUME(AI3WX5CK)","UN.DB1.TS1.PHOTO(AI3WX5CK)“
…
```

❖**Generate UNLOD utility using Automation Tool or Admin Tool or use HPU**

Populating a LOB/XML

- LOAD utility

1. **Basic LOAD utility: Objects** (includes non-LOB/XML + LOB/XML data)
   - Loads data as normal fields from the LOAD input file
   - Always LOAD into the base table
   - DB2 loads LOB to auxiliary table and XML to XML table
   - Use SPANNED option if the input file is created in a SPANNED format

2. Using **file reference variables** when each LOB/XML value is a member in a separate input file (DB2 9) (PDS/PDSE or HSF or Spanned)
   - Normal input file
     - Contains the data for the non-LOB/XML columns and the names of the LOB/XML input files (BLOBF, CLOBF, DBCLOBF)
     - LOAD syntax contains the names of the input files for LOB/XML
     - Base data + file names for LOB/XML data cannot > 32k
   - LOB/XML input file
     - Can be PDS, HFS directory
     - Contains the entire LOB/XML value

LOAD Syntax for using File Reference Variables

Names of the files containing the LOB / XML column values

```
//SYSREC DD *
"000001","UN.DB1.TS1.RESUME(AI3WX3JT)","UN.DB1.TS1.PHOTO(AI3WX3JT)"
"000002","UN.DB1.TS1.RESUME(AI3WX5BS)","UN.DB1.TS1.PHOTO(AI3WX5BS)"
"000003","UN.DB1.TS1.RESUME(AI3WX5CC)","UN.DB1.TS1.PHOTO(AI3WX5CC)"
"000004","UN.DB1.TS1.RESUME(AI3WX5CK)","UN.DB1.TS1.PHOTO(AI3WX5CK)“

LOAD DATA FORMAT DELIMITED
   INTO TABLE MY_EMP_PHOTO_RESUME
   (EMPNO CHAR,
    RESUME VARCHAR CLOBF,
    PHOTO VARCHAR BLOBF)
```

DB2 Administration Tool generates LOAD JCL using Template for file reference

Populating (LOADING) a LOB/XML

3. **For LOB only, Cross Loader (load data directly from another table)**
   - LOB value can be > 32k
   - DB2 uses a separate buffer for the LOB data – above the 16mb line

- Application – INSERT
  - Use a host variable large enough to hold the entire LOB/XML value
  - Use File Reference variables (DB2 9)
  - For XML, INSERT/UPDATE with or without schema validation

DB2 CHECK UTILITY to check consistency for LOB/XML

AUXERROR / LOBERROR / XMLERROR **REPORT/INVALIDATE**

FUNCTION	DESCRIPTION	LOB	XML
CHECK DATA SHRLEVEL reference/change	Check consistency **between a base table space and any associated LOB or XML table spaces. Report on missing LOBs & XML.**	**Y**	**Y** **With INCLUDE XML TABLESPACES**
CHECK INDEX SHRLEVEL reference/change	**Check consistency of indexes with data to which the index points**	**Y**	**Y**
CHECK LOB (SHRLEVEL reference/change)	**Checks consistency of a LOB table space (structural defects). Checks validity.** **Doesn't report on missing LOBs**	**Y**	**N/A**

- **Generate Check utility using DB2 Admin Tool**

Why do you need to REORG a LOB / XML table space or auxiliary index?

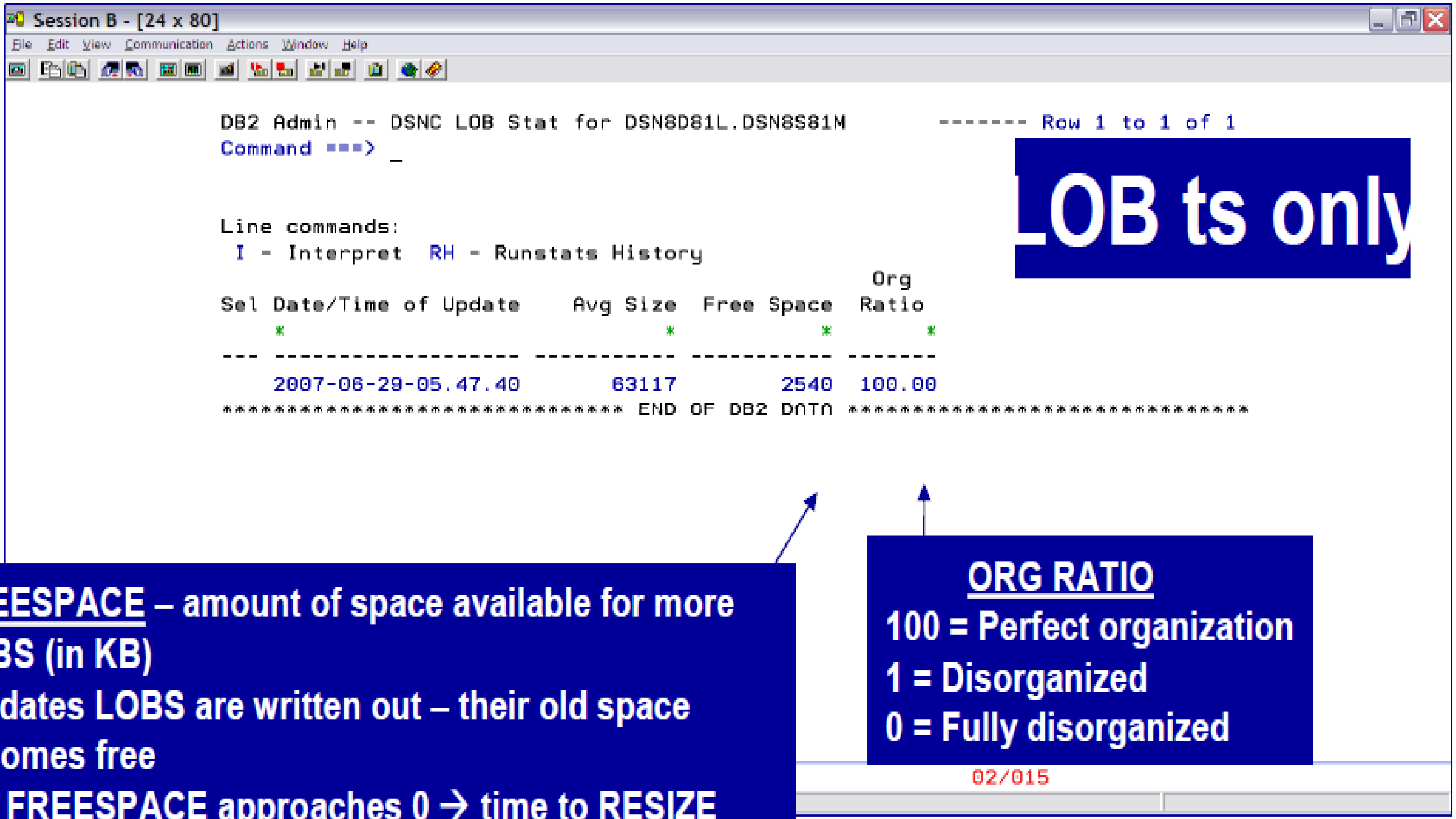
- Performance
- Reclaim physical space
- Improve clustering

SHRLEVEL NONE – REORG is done in place
SHRLVEL REFERENCE/CHNAGE – uses a shadow data set
RECOMMENDED METHOD for REORG LOB/XML TS

How do you know when to REORG a LOB or XML?

- ORGRATIO and FREESPACE in SYSLOBSTATS for LOB
- DSNUM and EXTENTS in SYSTABLEPART
- DSNUM, EXTENTS, LEAFNEAR, LEAFFAR, and PSEUDO_DEL_ENTRIES in SYSINDEXPART
- REORGINSERTS, REORGDLETES, REORGUPDATES, REORGDISORGLOB, REORGMASSDELETE, and EXTENTS in SYSTABLESPACES
- DISORGED_LOBS, DISORGED_LOBS_PCT from Real Time Stats

Display LOB statistics to determine if Reorg is required-Admin Tool



```
DB2 Admin -- DSNC LOB Stat for DSN8D81L.DSN8S81M        ------- Row 1 to 1 of 1
Command ===> _


Line commands:
 I - Interpret  RH - Runstats History
                                                          Org
Sel Date/Time of Update     Avg Size  Free Space  Ratio
    *                              *           *       *
--- ------------------- ----------- ----------- -------
    2007-06-29-05.47.40       63117        2540  100.00
******************************* END OF DB2 DATA *******************************
```

LOB ts only

FREESPACE – amount of space available for more LOBS (in KB)

- Updates LOBS are written out – their old space becomes free
- As FREESPACE approaches 0 → time to RESIZE using REORG SHRLEVEL REF

ORG RATIO

100 = Perfect organization

1 = Disorganized

0 = Fully disorganized

Reorg considerations for LOB/XML Table Space

- **REORG of a base table space does not result in a reorganization of the auxiliary LOB/XML objects**
- **FOR XML, must specify the name of the auxiliary XML object or LISTDEF with XML/ALL or for LOB specify AUX YES**
- **For XML TS, cannot specify**
  - DISCARD
  - REBALANCE
  - UNLOAD EXTERNAL
- **Must also specify the WORKDDN keyword**
- **When LOADing XML data, compression only occurs at the time of the REORG**

❖DB2 Automation Tool generates appropriate REORG JCL based on exceptions

Managing LOB, XML & Bi-Temporal Data using IBM DB2 Tools

IBM DB2 Tools:

- ➢ **Drive immediate DB2 10 out-of-the-box Performance Savings**

- ▪ **Fast Data Unload**
  - ➢ DB2 High Performance Unload (supports LOB, XML, Bi-Temporal table)
  - ➢ Can run outside DB2 and can create output in multiple formats
- ▪ **Enhance DB2 Utility performance with High speed DB2 utility sort**
  - ➢ DB2 Sort
    - ➢ Improves DB2 utility sort performance
    - ➢ DB2 Sort leverages the strengths of the System z platform, DB2 for z/OS and the DB2 Utilities Suite to drive:
      - ➢ Significant savings in elapsed time and CPU during utility sort processing, especially LOAD, REORG and RUNSTATS
      - ➢ Relief from application constraints of large volumes of data in highly-transactional workloads performing numerous insert, update and delete operations against DB2 for z/OS databases

Enhance Management- DB2 Utility Enhancement Tool V2.1

- **Offers a proactive way to cancel threads holding locks so the utility or a batch job can complete without -904 failure**
- **Extends utility functions:**
  - **LOAD:**
    - ➢ **CONSTANT** or **conditionally replace the value**
    - ➢ **PRESORT** to reduce elapsed/CPU times
  - **REORG:**
    - ➢ Automatically size and create the mapping table/index and drop upon completion
  - **CHECK DATA**
    - ➢ **Discards to** a flat file in a load format and creates Load control cards
    - ➢ Automatically creates/sizes discard table and drops it upon completion

➢ Utility Syntax Monitor

- ➢ Changes utility syntax at run-time based on Policy rules to Enforce company IT policies
  - ➢ **ADD/ REMOVE /SUBSTITUTE parameters**
  - ➢ **FAIL the utility based on object name, or user ID**
- ➢ Each action is logged
  - ➢ **Audit who is doing what**
  - ➢ **See what syntax was originally specified**
  - ➢ **See what the original syntax was changed to**

Extend Administration Capabilities

- **DB2 Administration Tool & Object Compare**
    - Support of LOB, XML, Bi-Temporal tables & DB2 10 features
    - Catalog Navigation
  - Change Management (Enhanced CM) & DB2 10 Pending Changes
    - Migration of DDL, DATA and /or Catalog Statistics
    - Utility generation, LISTDEF/TEMPALTE support
    - Space Management and performance queries
    - Compare object definitions and apply the changes to synchronize the environments

Automate Routine Maintenance jobs

- DB2 Automation Tool:
  - Automate routine maintenance tasks as and when needed basis
  - Prevent unnecessary maintenance from being run
    - Saves on CPU and IO costs
  - Easy to setup:
    - Define object, exception, utility profiles, and combine them into a job profile to generate the utility JCL
  - Provides statistical history reports for trend analysis and forecasting
  - Data Page Display allows to edit data pages directly to correct invalid data
  - DB2 Command Processor allows to issue DB2 commands without having to leave the interface
  - Dataset Manager enables to view, evaluate, and relocate DB2 data sets
  - DB2 10 support:
    - Supports Autonomic Statistics
    - Supports FlashCopy Image Copy
    - Supports Avoiding Unnecessary Reorgs
      - Set REORG thresholds based on DB2 10 Best Practices
      - > Detect when Indexes are insensitive to Clustering
      - > Avoid REORGs for poorly structured Indexes

Manage Backup/Recovery using advanced technology

DB2 Recovery Expert V2.2

- **Simplifies and automates the recovery processes**
- **Recommends the fastest and least costly recovery options**
- **Builds and validates the recovery plans supporting all types of recoveries including the recovery of dropped objects**
- **Provides an instantaneous backup and recovery solutions using fast replication storage hardware**
- **Automates backup/restore of an entire DB2 subsystem or partial backups**
- **Copies and restores individual DB2 objects or groups of DB2 objects from the system level backups**
- **Exploits FlashCopy Image Copy to take Consistent Online Image Copies in seconds**
- **Exploit RECOVER BACKOUT for faster recoveries**

- **Includes a subset of DB2 Log Analysis services**
  - **Undo and Redo of LOB, XML & Bi-Temporal Data for application recovery plan**
  - Quiet point analysis
- **Automates disaster recovery process**

Identify, and Restore unwanted changes &
Automate Subsystem and Object Cloning

- DB2 Log Analysis Tool 3.2
  - Report on log activity & Quiet time detection
  - Generate Undo and Redo records for application repair, testing and debugging including LOB/XML/Temporal tables
  - Support FlashCopy Image Copy

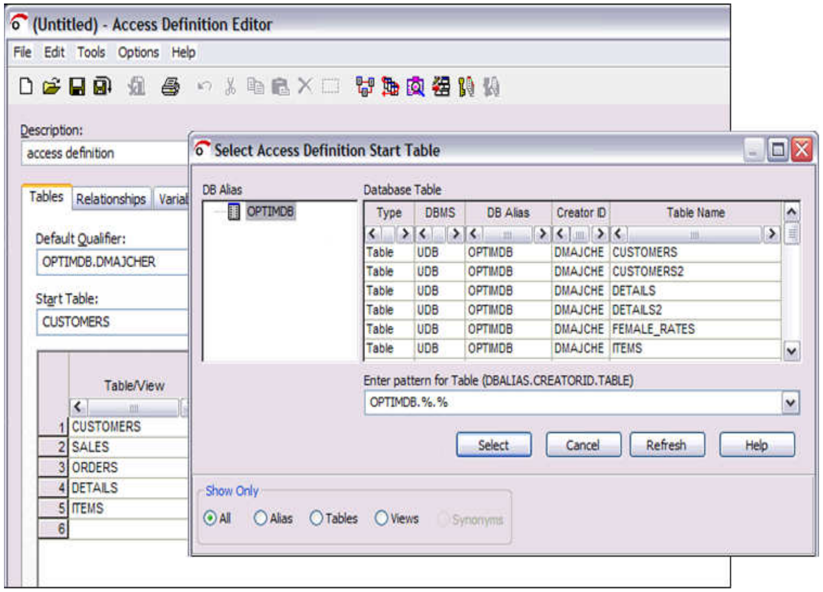
- DB2 Cloning Tool V2.2
  - Exploit Storage-based copies to drastically reduce CPU and outages
  - Create Subsystem and Object Clones with minimal effort
    - Automatically reduce number of Data Sharing Members
    - Convert Data Sharing to non-Data Sharing
    - Create Subsystem Clone from System Level Backup created by Recovery Expert
    - Mask sensitive production data
  - Supports native IBM, EMC and Hitachi Storage-based copies

Create “right-size” production-like environments for application testing

SUPPORT BUSINESS GROWTH
PROTECT INFORMATION
MANAGE COSTS
BUSINESS VALUE
INTEGRATED DATA MANAGEMENT
Accelerate Solution Delivery
Simplify Test Data Management
Maximize Data Availability
Assure Compliance
Ensure Data Privacy and Security
Optimize Performance
Manage Data Growth
Streamline Upgrades/Migrations & Retirement/ Consolidation
Manage Data for Greater Efficiencies
Streamline Compliance Process

Test Data Management

IBM InfoSphere Optim Test Data management Solution


(Untitled) - Access Definition Editor
File Edit Tools Options Help
Tables Relationships Varial
Select Access Definition Start Table
Select
Cancel
Refresh
Help

Description: access definition

Default Qualifier: OPTIMDB.DMAJCHER

Start Table: CUSTOMERS

	Table/View
1	CUSTOMERS
2	SALES
3	ORDERS
4	DETAILS
5	ITEMS
6	

DB Alias: OPTIMDB

Database Table

Type	DBMS	DB Alias	Creator ID	Table Name
Table	UDB	OPTIMDB	DMAJCHE	CUSTOMERS
Table	UDB	OPTIMDB	DMAJCHE	CUSTOMERS2
Table	UDB	OPTIMDB	DMAJCHE	DETAILS
Table	UDB	OPTIMDB	DMAJCHE	DETAILS2
Table	UDB	OPTIMDB	DMAJCHE	FEMALE_RATES
Table	UDB	OPTIMDB	DMAJCHE	ITEMS

Enter pattern for Table (DBALIAS.CREATORID.TABLE): OPTIMDB.%.%

Show Only: ● All ○ Alias ○ Tables ○ Views ○ Synonyms

Data Masking

Benefits

- Protect sensitive information from misuse & fraud
- Prevent data breaches and associated fines
- Achieve better data governance

Requirements

- Create referentially intact, “right-sized” test databases
- Automate test result comparisons to identify hidden errors
- Shorten iterative testing cycles and accelerate time to market

Benefits

- Deploy new functionality more quickly and with improved quality
- Easily refresh & maintain test environments
- Reduce storage and operational costs

Manage data growth and improve performance by intelligently archiving historical data

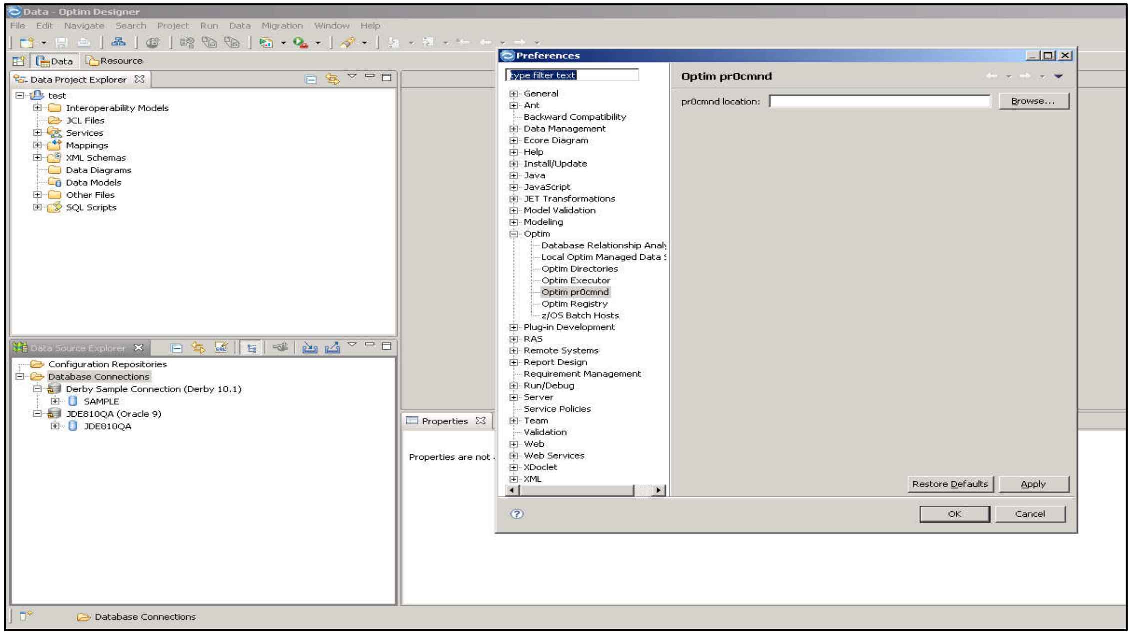
SUPPORT BUSINESS GROWTH
PROTECT INFORMATION
MANAGE COSTS
BUSINESS VALUE
INTEGRATED DATA MANAGEMENT
Accelerate Solution Delivery
Simplify Test Data Management
Maximize Data Availability
Assure Compliance
Ensure Data Privacy and Security
Optimize Performance
Manage Data Growth
Streamline Upgrades/Migrations & Retirement/Consolidation
Manage Data for Greater Efficiencies
Streamline Compliance Process

Data Growth

IBM InfoSphere

Optim Data Growth

Solution



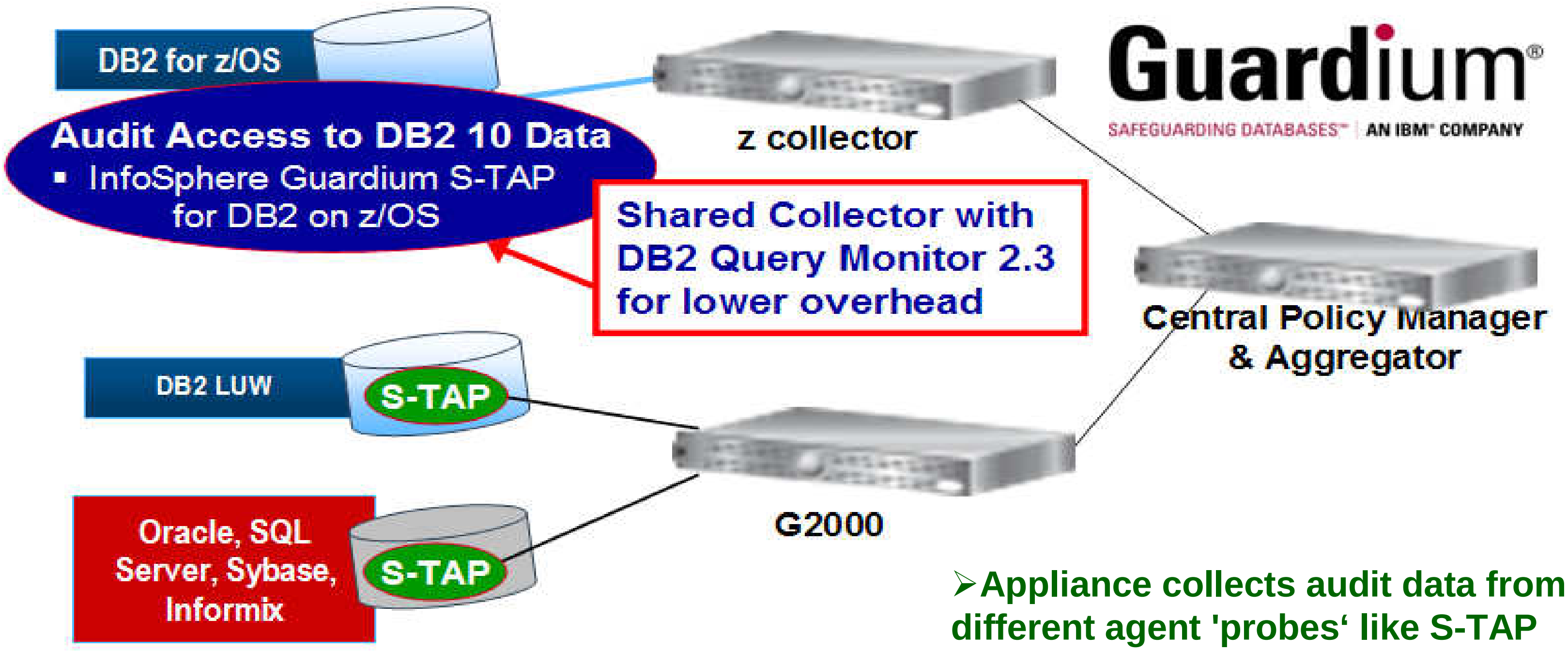
Requirements

- Archive, manage and retain application data according to business policies
- Minimize downtime during application upgrades
- Consolidate application portfolio and retire legacy applications

Benefits

- Reduce hardware, storage and maintenance costs
- Streamline application upgrades and improve application performance
- Safely retire legacy & redundant applications while retaining the data

Safeguard Data- InfoSphere Guardium S-TAP for DB2 on z/OS & Data Encryption Tool



DB2 for z/OS

Audit Access to DB2 10 Data
- InfoSphere Guardium S-TAP for DB2 on z/OS

z collector

Guardium®
SAFEGUARDING DATABASES℠ | AN IBM® COMPANY

Shared Collector with DB2 Query Monitor 2.3 for lower overhead

Central Policy Manager & Aggregator

DB2 LUW

S-TAP

G2000

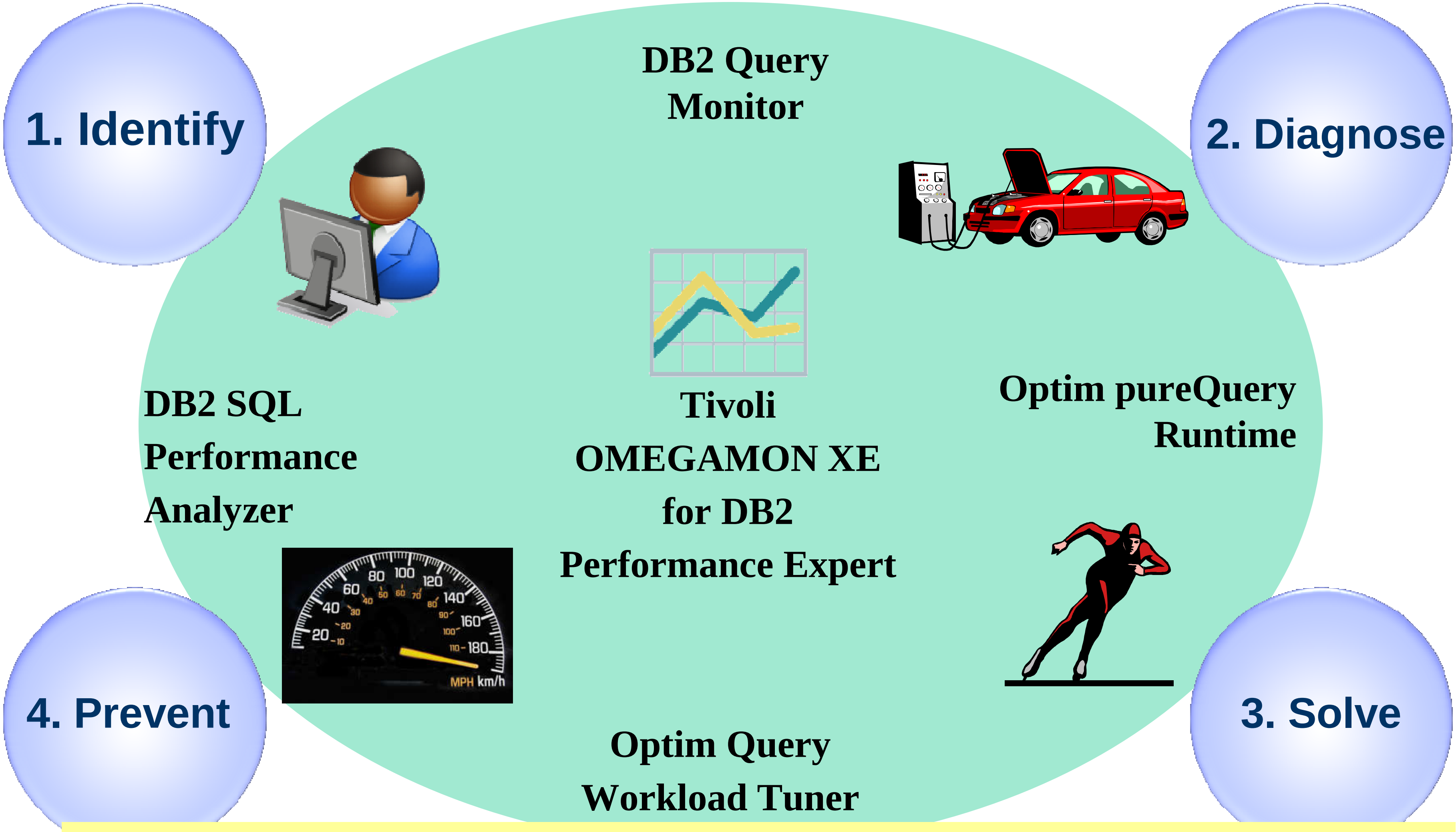
Oracle, SQL Server, Sybase, Informix

S-TAP

- ➢Real time security monitoring & blocking attackers
- ➢Non-invasive architecture
  - ➢ Outside database
  - ➢Minimal performance impact
  - ➢No DBMS or application changes
- ➢Cross-DBMS solution
- ➢100% visibility including local DBA access

- ➢Appliance collects audit data from different agent 'probes' like S-TAP
- ➢Comprehensive Audit trail, Pre-packaged compliance reporting
- ➢Audit repository and policy engine self contained in hardened hardware appliance No root privileges made available to the customer
- ➢Complete separation of roles protection

Performance Management & Tuning



1. Identify

2. Diagnose

3. Solve

4. Prevent

DB2 Query Monitor

DB2 SQL Performance Analyzer

Tivoli OMEGAMON XE for DB2 Performance Expert

Optim pureQuery Runtime

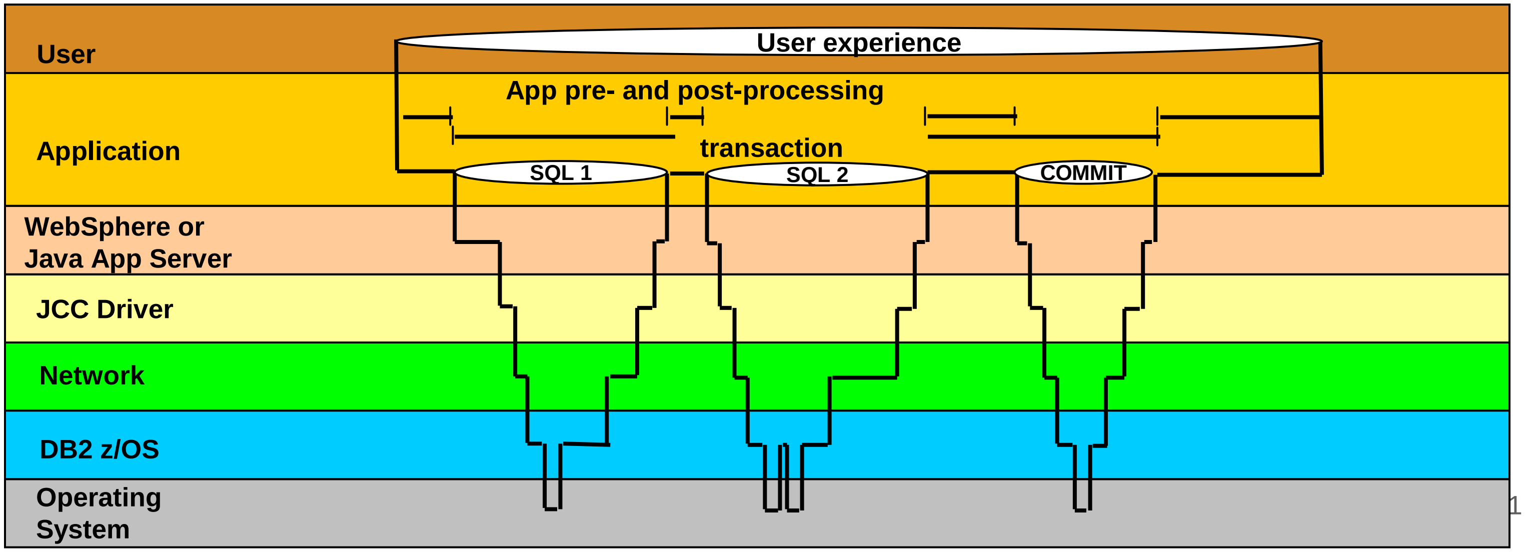
Optim Query Workload Tuner

Identify, diagnose, solve and prevent performance problems

Optimize Dynamic Infrastructure Performance

OMEGAMON XE for DB2 Performance Expert 5.1 Exploitation

- **Extended Insight**
  - Surface DB2 for z/OS end-to-end response time metrics
    - ➢ Visibility to **<u>all</u>** the components that make up end-user response time
    - ➢ Facilitates platform-agnostic identification of response time bottlenecks
    - ➢ Enables near-instantaneous response to and prevention of application slowdowns
  - Leverages Tivoli Enterprise Portal GUI



Exploit DB2 10 for z/OS with IBM DB2 Tools

Accelerate your ability to leverage compelling DB2 10 features with comprehensive Tools support

- Data Encryption Tool for IMS and DB2 Databases
- DB2 Administration Tool / DB2 Object Compare for z/OS
- DB2 Audit Management Expert for z/OS
- DB2 Automation Tool for z/OS
- DB2 Bind Manager for z/OS
- DB2 Change Accumulation Tool for z/OS
- DB2 Cloning Tool for z/OS
- DB2 High Performance Unload for z/OS
- DB2 Log Analysis Tool for z/OS
- DB2 Object Restore for z/OS
- DB2 Path Checker for z/OS
- DB2 Query Management Facility for z/OS
- DB2 Query Monitor for z/OS
- DB2 Recovery Expert for z/OS
- DB2 SQL Performance Analyzer for z/OS
- DB2 Table Editor for z/OS
- DB2 Utilities Enhancement Tool for z/OS
- DB2 Utilities Suite for z/OS
- InfoSphere Change Data Capture
- InfoSphere Data Event Publisher
- InfoSphere Replication Server
- InfoSphere Optim Data Growth Solution for z/OS
- Optim Development Studio
- Optim pureQuery Runtime
- Optim Query Workload Tuner
- InfoSphere Optim Test Data Management Solution for z/OS
- Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS

Exploitation PTFs: http://www-01.ibm.com/support/docview.wss?uid=swg21409518

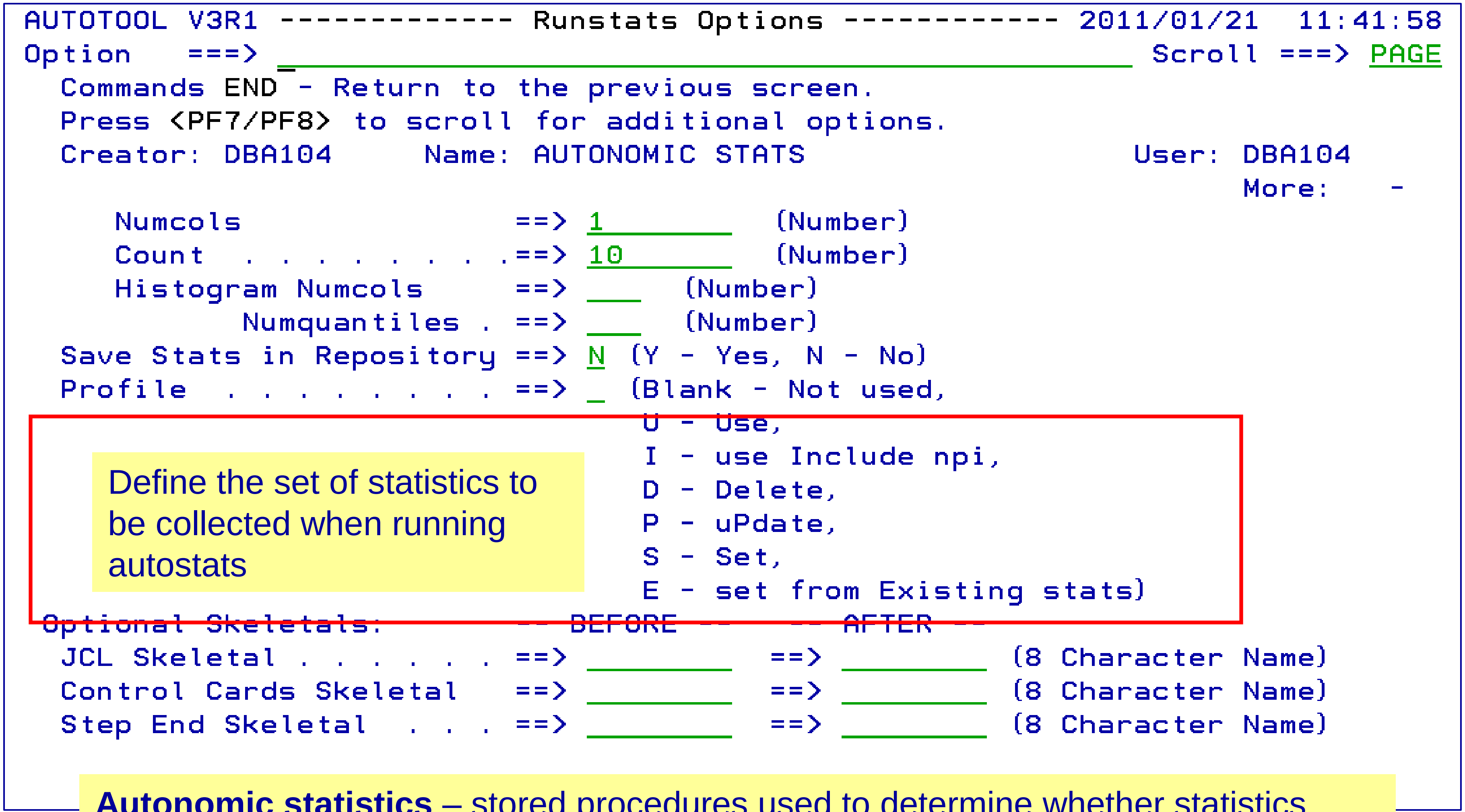
REFERENCES

- **DB2 UDB for z/OS home page**

  http://www.ibm.com/software/data/db2/zos/index.html
- **Redbook -LOBs with DB2 for z/OS: Stronger and Faster, SG24-7270**
- **Redbook- Extremely pureXML in DB2 10 for z/OS-**SG24-7915-00
- **Redbook-DB2 V9 for z/OS Technical Overview, SG24-7330**
- **Redbook-DB2 10 for z/OS Technical Overview,** SG24-7892-00
- **DB2 10 for z/OS *Managing Performance*, SC19-2978-01**
- **DB2 V9 XML Guide, SC18-9858-04**
- **DB2 10 XML Guide SC19-2981-02**
- **DB2 Tools for z/OS home page**

  http://www.ibm.com/software/data/db2imstools

RUNSTAT profile using DB2 10 Autonomic Statistics – DB2 Automation Tool V3.1

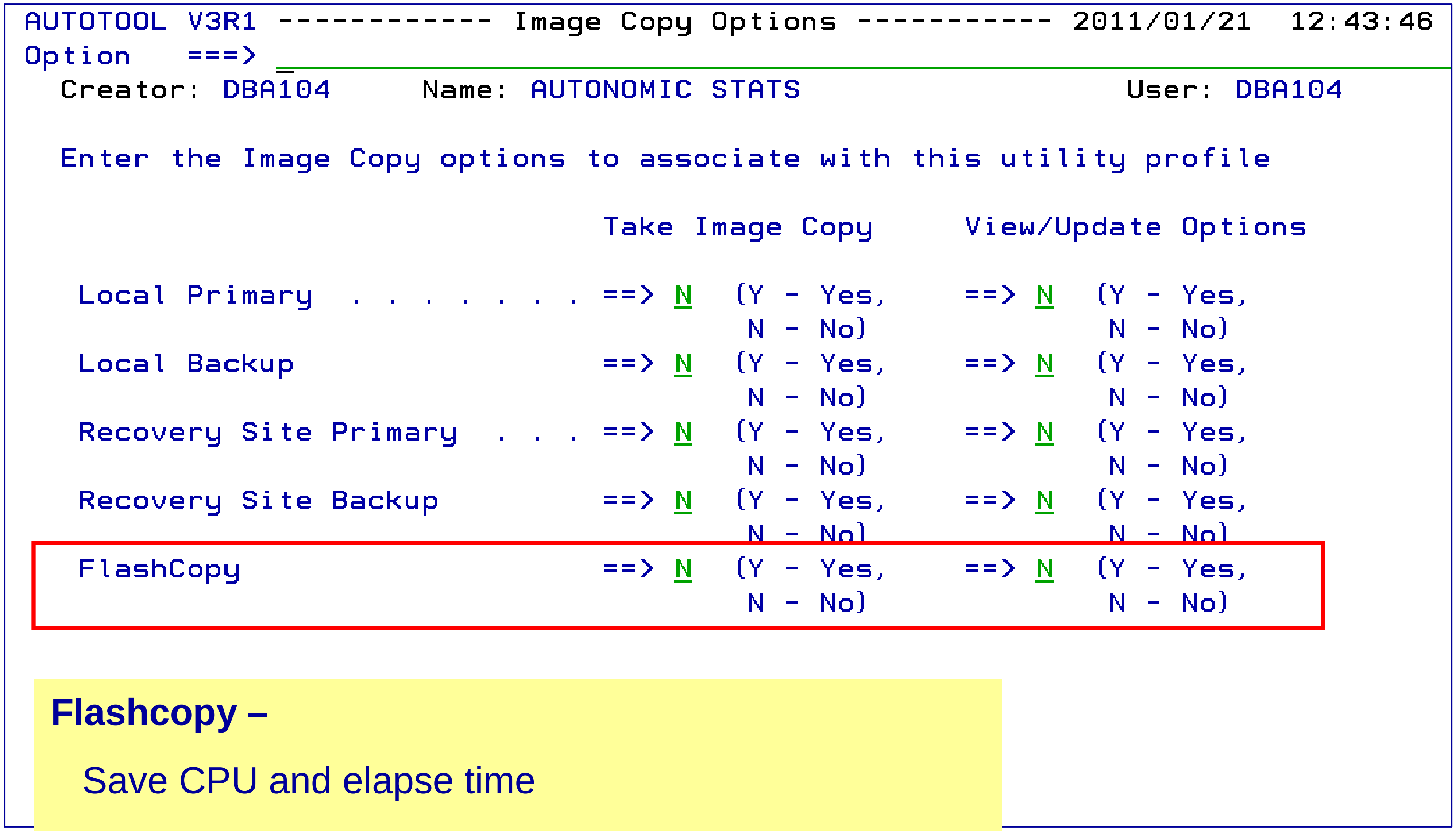


```
AUTOTOOL V3R1 ------------- Runstats Options ------------ 2011/01/21  11:41:58
Option   ===> _______________________________________________ Scroll ===> PAGE
  Commands END - Return to the previous screen.
  Press <PF7/PF8> to scroll for additional options.
  Creator: DBA104     Name: AUTONOMIC STATS                  User: DBA104
                                                                   More:    -
     Numcols                  ==> 1________   (Number)
     Count  . . . . . . . . .==> 10_______   (Number)
     Histogram Numcols        ==> ___    (Number)
            Numquantiles .    ==> ___    (Number)
  Save Stats in Repository ==> N (Y - Yes, N - No)
  Profile  . . . . . . . .   ==> _ (Blank - Not used,
                                     U - Use,
                                     I - use Include npi,
                                     D - Delete,
                                     P - uPdate,
                                     S - Set,
                                     E - set from Existing stats)
 Optional Skeletals:           -- BEFORE --   -- AFTER --
  JCL Skeletal . . . . . . . ==> ________   ==> ________  (8 Character Name)
  Control Cards Skeletal     ==> ________   ==> ________  (8 Character Name)
  Step End Skeletal    . . . ==> ________   ==> ________  (8 Character Name)
```

Define the set of statistics to be collected when running autostats

Autonomic statistics – stored procedures used to determine whether statistics should be collected or recollected (ADMIN_UTL_MONITOR; ADMIN_UTL_EXECURE; ADMIN_UTL_MODIFY)

DB2 Automation Tool V3.1 FLASHCOPY



```
AUTOTOOL V3R1 ------------ Image Copy Options ----------- 2011/01/21  12:43:46
Option   ===> _
  Creator: DBA104     Name: AUTONOMIC STATS                      User: DBA104

  Enter the Image Copy options to associate with this utility profile

                              Take Image Copy     View/Update Options

   Local Primary . . . . . . . ==> N  (Y - Yes,     ==> N  (Y - Yes,
                                       N - No)              N - No)
   Local Backup                ==> N  (Y - Yes,     ==> N  (Y - Yes,
                                       N - No)              N - No)
   Recovery Site Primary . . . ==> N  (Y - Yes,     ==> N  (Y - Yes,
                                       N - No)              N - No)
   Recovery Site Backup        ==> N  (Y - Yes,     ==> N  (Y - Yes,
                                       N - No)              N - No)
   FlashCopy                   ==> N  (Y - Yes,     ==> N  (Y - Yes,
                                       N - No)              N - No)
```

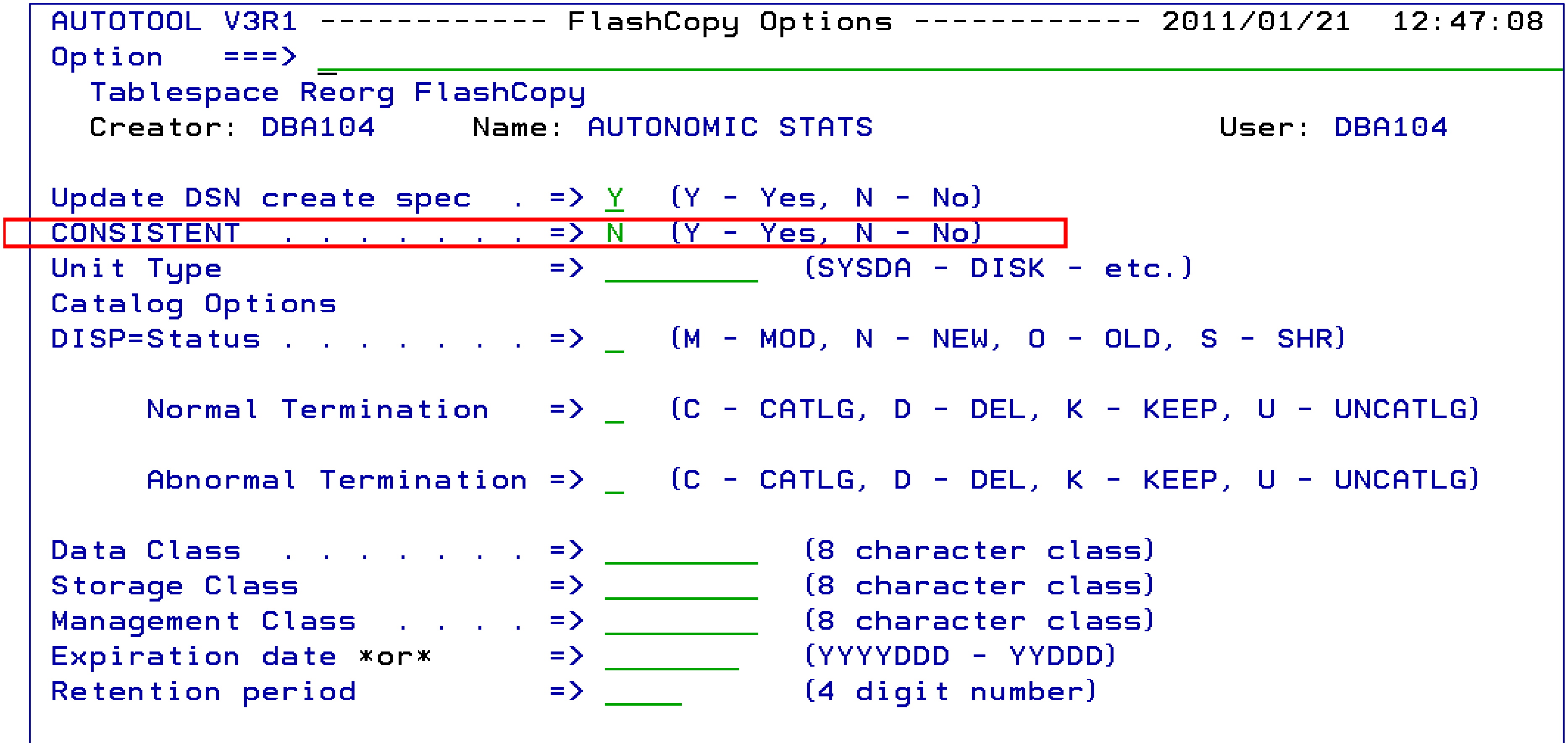
Flashcopy –

Save CPU and elapse time

Create with COPY, REORG, LOAD, REBUILD INDEX

Can be used by RECOVER

FLASHCOPY CONSISTENT Option- DB2 Automation Tool V3.1

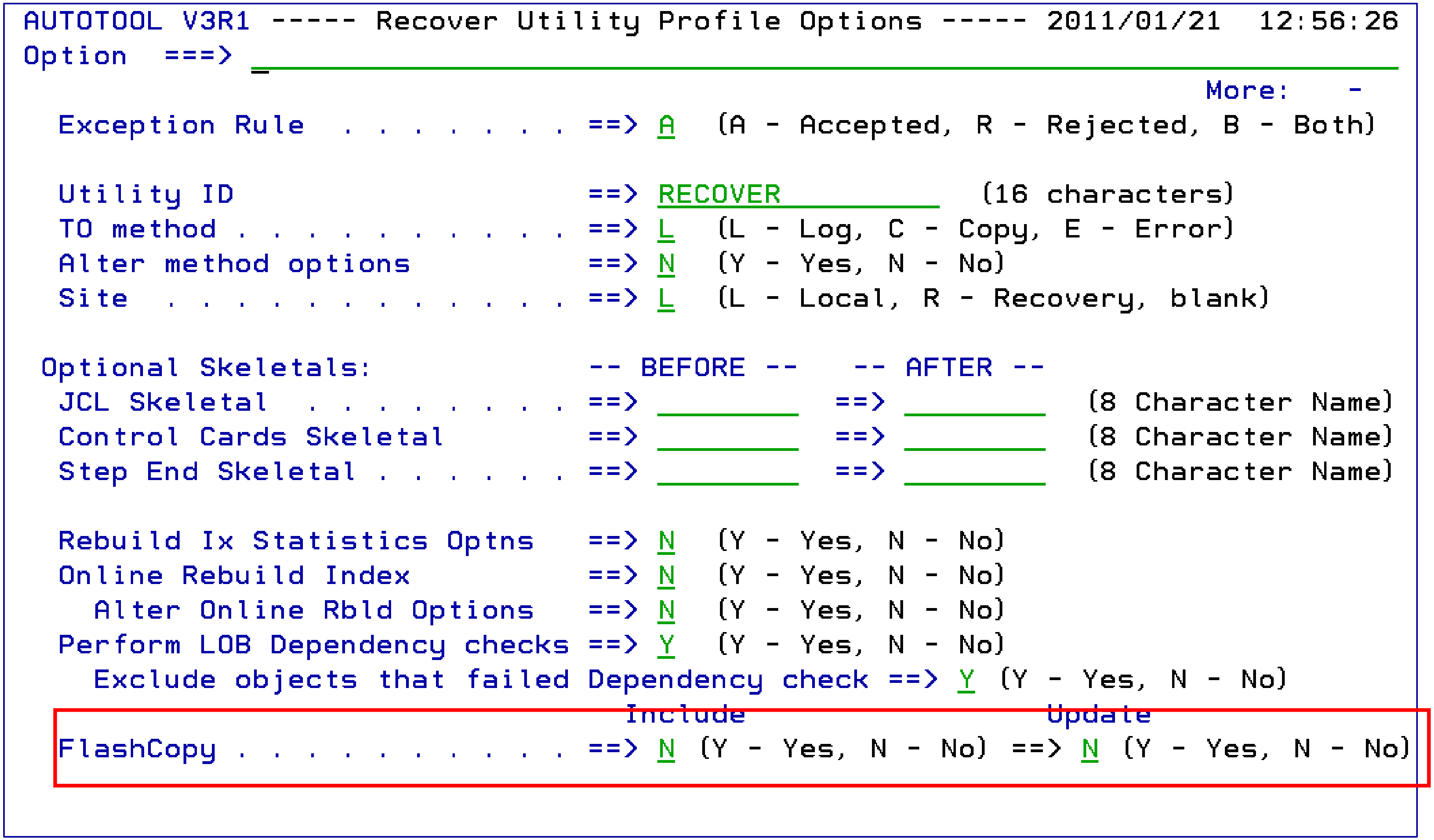


```
AUTOTOOL V3R1 ------------ FlashCopy Options ------------ 2011/01/21  12:47:08
Option   ===> _
  Tablespace Reorg FlashCopy
  Creator: DBA104     Name: AUTONOMIC STATS                 User: DBA104

Update DSN create spec  . => Y  (Y - Yes, N - No)
CONSISTENT  . . . . . . . => N  (Y - Yes, N - No)
Unit Type                 => ________  (SYSDA - DISK - etc.)
Catalog Options
DISP=Status . . . . . . . => _  (M - MOD, N - NEW, O - OLD, S - SHR)

     Normal Termination   => _  (C - CATLG, D - DEL, K - KEEP, U - UNCATLG)

     Abnormal Termination => _  (C - CATLG, D - DEL, K - KEEP, U - UNCATLG)

Data Class  . . . . . . . => ________  (8 character class)
Storage Class             => ________  (8 character class)
Management Class  . . . . => ________  (8 character class)
Expiration date *or*      => _______   (YYYYDDD - YYDDD)
Retention period          => ____      (4 digit number)
```

COPY & LOAD w/ SHRLEVEL CHANGE

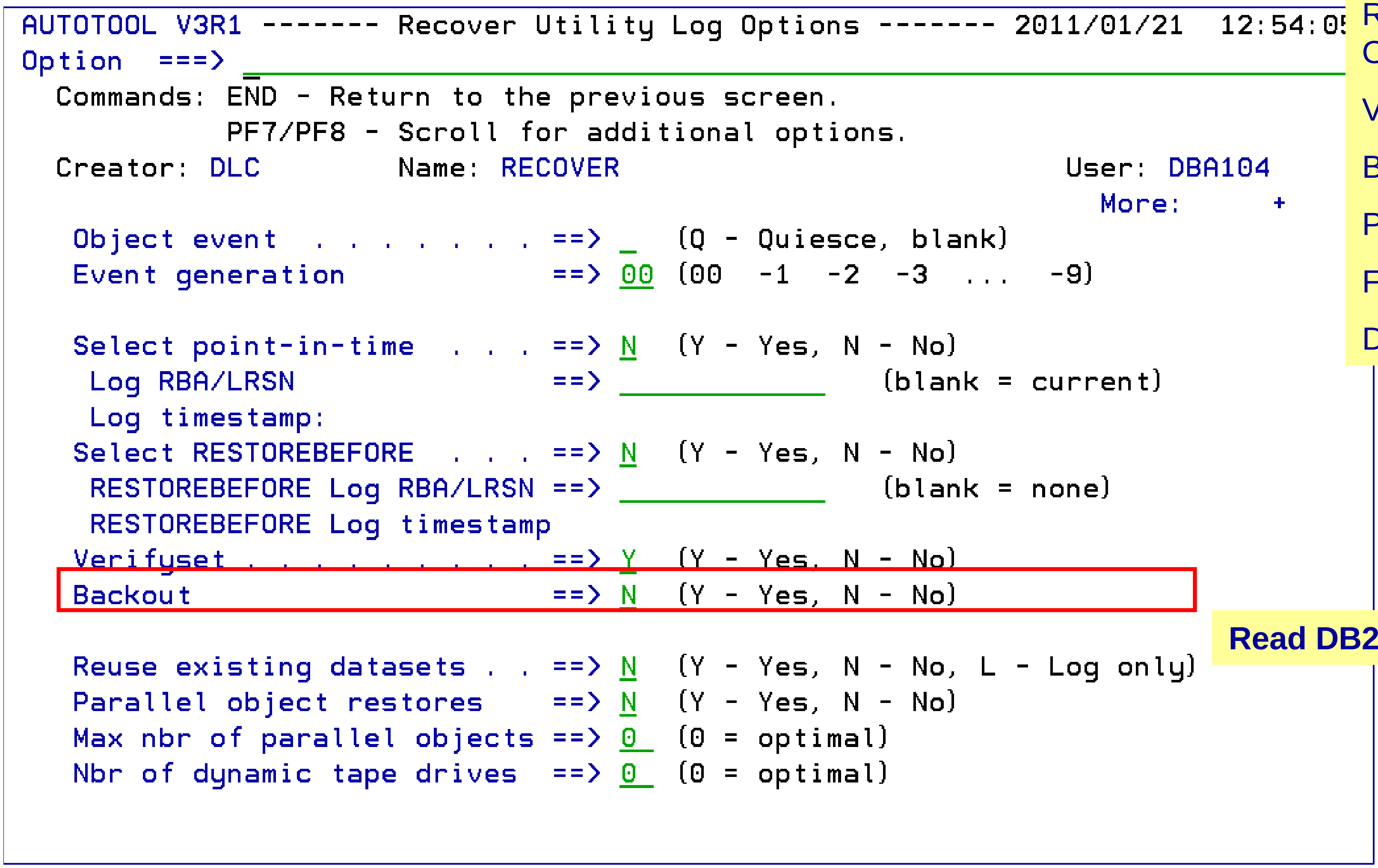
CONSISTENT copy – copy the object and back out uncommitted changes

FLASHCOPY w/ RECOVER using DB2 Automation Tool V3.1



```
AUTOTOOL V3R1 ----- Recover Utility Profile Options ----- 2011/01/21  12:56:26
Option  ===> _
                                                             More:    -
  Exception Rule  . . . . . . . ==> A  (A - Accepted, R - Rejected, B - Both)

  Utility ID                    ==> RECOVER           (16 characters)
  TO method . . . . . . . . . . ==> L  (L - Log, C - Copy, E - Error)
  Alter method options          ==> N  (Y - Yes, N - No)
  Site  . . . . . . . . . . . . ==> L  (L - Local, R - Recovery, blank)

 Optional Skeletals:            -- BEFORE --   -- AFTER --
  JCL Skeletal  . . . . . . . . ==>          ==>          (8 Character Name)
  Control Cards Skeletal        ==>          ==>          (8 Character Name)
  Step End Skeletal . . . . . . ==>          ==>          (8 Character Name)

  Rebuild Ix Statistics Optns   ==> N  (Y - Yes, N - No)
  Online Rebuild Index          ==> N  (Y - Yes, N - No)
    Alter Online Rbld Options   ==> N  (Y - Yes, N - No)
  Perform LOB Dependency checks ==> Y  (Y - Yes, N - No)
    Exclude objects that failed Dependency check ==> Y (Y - Yes, N - No)
                                  Include                Update
  FlashCopy . . . . . . . . . . ==> N (Y - Yes, N - No) ==> N (Y - Yes, N - No)
```

Generate RECOVER BACKOUT using DB2 Automation Tool V3.1



```
AUTOTOOL V3R1 ------- Recover Utility Log Options ------- 2011/01/21  12:54:05
Option  ===> ________________________________________________________________
  Commands: END - Return to the previous screen.
            PF7/PF8 - Scroll for additional options.
  Creator: DLC        Name: RECOVER                         User: DBA104
                                                                 More:     +
     Object event  . . . . . . . ==> _  (Q - Quiesce, blank)
     Event generation            ==> 00 (00  -1  -2  -3  . . .  -9)

     Select point-in-time  . . . ==> N  (Y - Yes, N - No)
      Log RBA/LRSN               ==> ____________   (blank = current)
      Log timestamp:
     Select RESTOREBEFORE  . . . ==> N  (Y - Yes, N - No)
      RESTOREBEFORE Log RBA/LRSN ==> ____________   (blank = none)
      RESTOREBEFORE Log timestamp
     Verifyset . . . . . . . . . ==> Y  (Y - Yes, N - No)
     Backout                     ==> N  (Y - Yes, N - No)

     Reuse existing datasets . . ==> N  (Y - Yes, N - No, L - Log only)
     Parallel object restores    ==> N  (Y - Yes, N - No)
     Max nbr of parallel objects ==> 0  (0 = optimal)
     Nbr of dynamic tape drives  ==> 0  (0 = optimal)
```

TOLOGRBA

TOLOGPOINT

RESTOREBEFORE

VERIFYSET

BACKOUT

PARALLEL

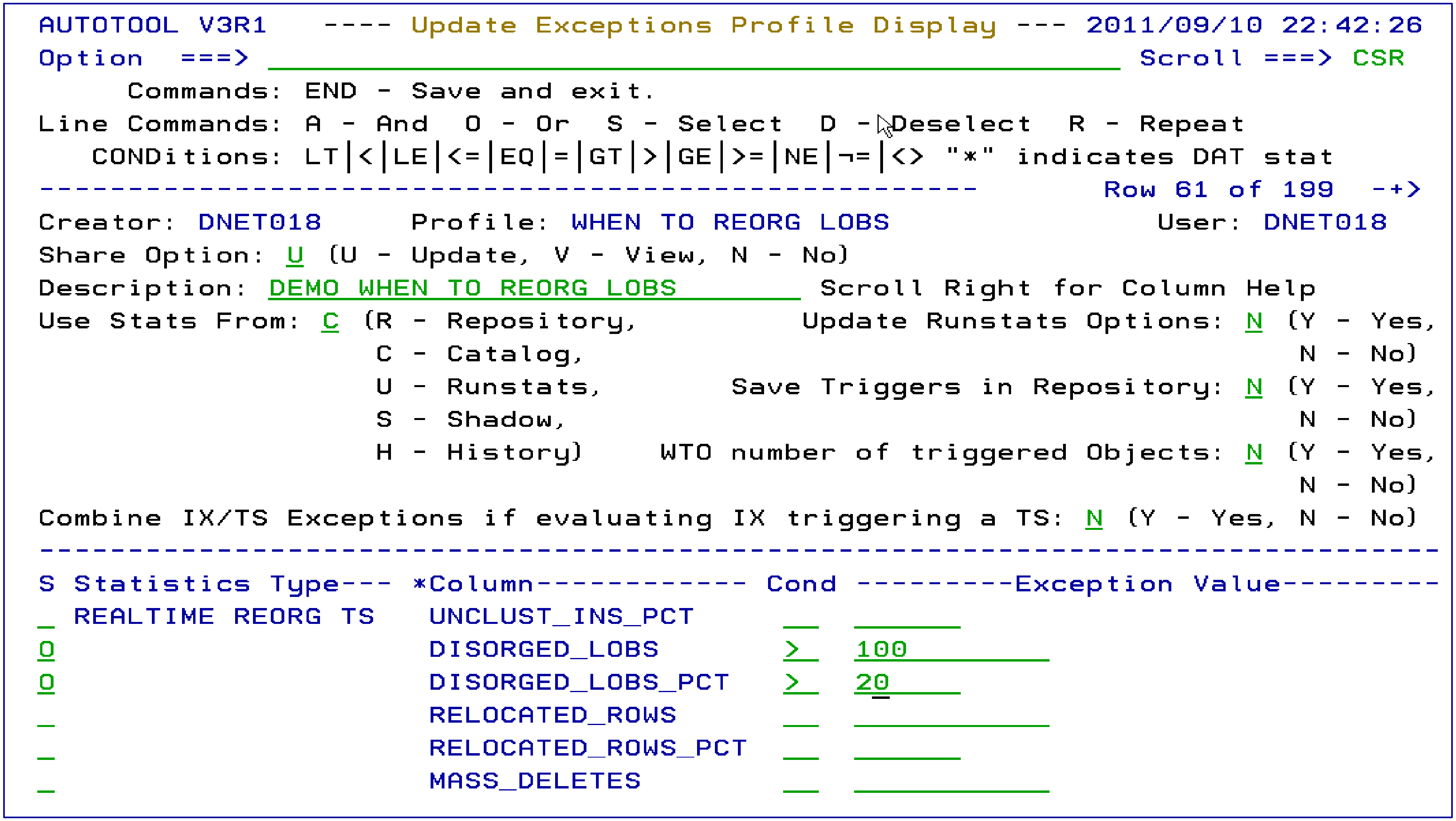
FROMDUMP

DUMPCLASS

Read DB2 log backwards

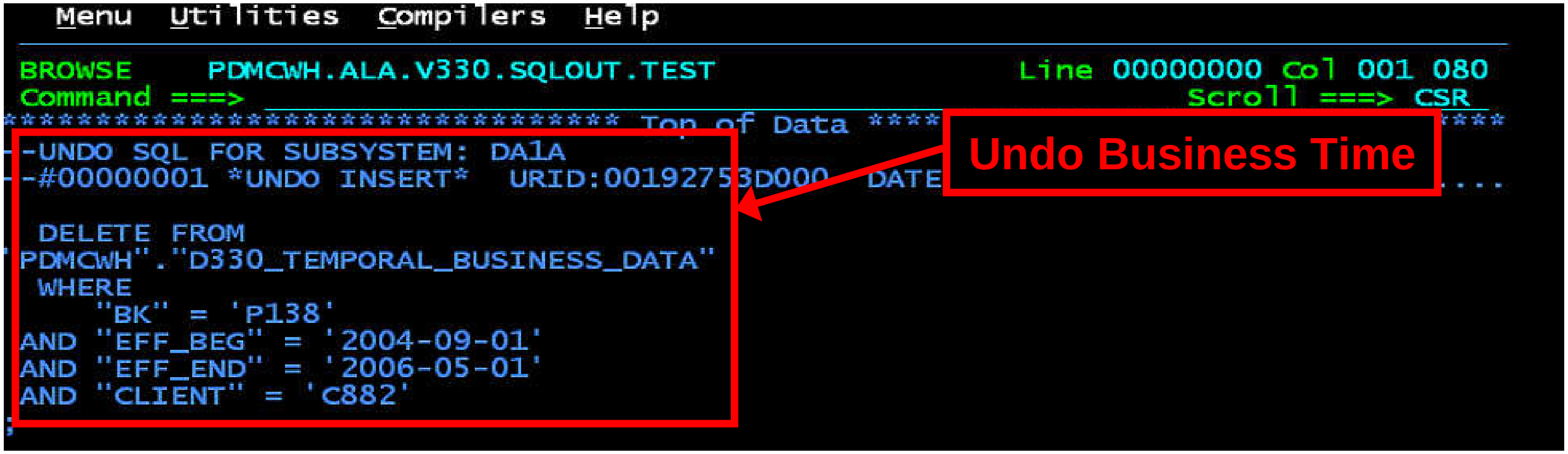
DB2 Automation Tool V3.1 Exceptions

on Real Time Stats on **DISORGED_LOBS DISORGED_LOBS_PCT**



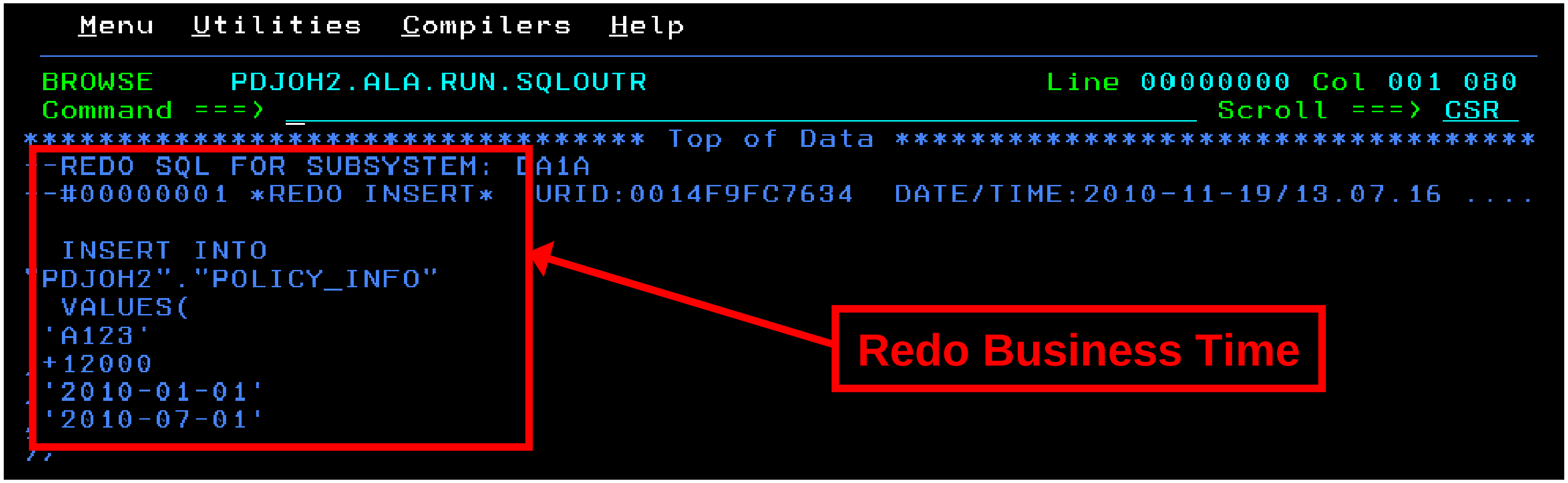
```
AUTOTOOL V3R1   ---- Update Exceptions Profile Display --- 2011/09/10 22:42:26
Option  ===> ________________________________________________ Scroll ===> CSR
     Commands: END - Save and exit.
Line Commands: A - And  O - Or  S - Select  D - Deselect  R - Repeat
   CONDitions: LT|<|LE|<=|EQ|=|GT|>|GE|>=|NE|¬=|<> "*" indicates DAT stat
------------------------------------------------------      Row 61 of 199   -+>
Creator: DNET018     Profile: WHEN TO REORG LOBS               User: DNET018
Share Option: U (U - Update, V - View, N - No)
Description: DEMO WHEN TO REORG LOBS_______ Scroll Right for Column Help
Use Stats From: C (R - Repository,         Update Runstats Options: N (Y - Yes,
                   C - Catalog,                                        N - No)
                   U - Runstats,       Save Triggers in Repository: N (Y - Yes,
                   S - Shadow,                                         N - No)
                   H - History)    WTO number of triggered Objects: N (Y - Yes,
                                                                       N - No)
Combine IX/TS Exceptions if evaluating IX triggering a TS: N (Y - Yes, N - No)
--------------------------------------------------------------------------------
S Statistics Type--- *Column------------ Cond ---------Exception Value----------
_ REALTIME REORG TS   UNCLUST_INS_PCT     __  ______
O                     DISORGED_LOBS       >_  100________
O                     DISORGED_LOBS_PCT   >_  20____
_                     RELOCATED_ROWS      __  ___________
_                     RELOCATED_ROWS_PCT  __  ______
_                     MASS_DELETES        __  ___________
```

Undo Redo of Bi-temporal Data- Supports undo/redo of LOB/XML



```
BROWSE    PDMCWH.ALA.V330.SQLOUT.TEST                   Line 00000000 Col 001 080
Command ===>                                                     Scroll ===> CSR
********************************* Top of Data ****                          ****
-UNDO SQL FOR SUBSYSTEM: DA1A
-#00000001 *UNDO INSERT*  URID:00192753D000  DATE                           ....

 DELETE FROM
"PDMCWH"."D330_TEMPORAL_BUSINESS_DATA"
 WHERE
    "BK" = 'P138'
AND "EFF_BEG" = '2004-09-01'
AND "EFF_END" = '2006-05-01'
AND "CLIENT" = 'C882'
;
```

Undo Business Time



```
BROWSE    PDJOH2.ALA.RUN.SQLOUTR                        Line 00000000 Col 001 080
Command ===> _                                                   Scroll ===> CSR
******************************** Top of Data ***********************************
-REDO SQL FOR SUBSYSTEM: DA1A
-#00000001 *REDO INSERT*  URID:0014F9FC7634  DATE/TIME:2010-11-19/13.07.16 ....

 INSERT INTO
"PDJOH2"."POLICY_INFO"
 VALUES(
'A123'
,+12000
,'2010-01-01'
,'2010-07-01'
);
```

Redo Business Time