



Usage of XML data in DB2 V9 z/OS

XML ELEMENTS	1
<i>Glossary</i>	1
<i>The XML docid</i>	2
SQL RESTRICTIONS	2
XML AND UTILITIES	2
<i>UNLOAD'ing XML data</i>	3
<i>LOAD'ing XML data</i>	3
<i>Cross LOAD'ing data</i>	3
INTERNAL STRUCTURE	4
<i>DDL</i>	4
<i>Tablespaces</i>	5

XML elements

Glossary

- **AUXILIARY INDEX**
An index on an auxiliary table in which each index entry refers to a LOB or XML document.
- **AUXILIARY TABLE**
A table that contains columns outside the actual table in which they are defined.
Auxiliary tables
- **DOCUMENT ID**
A value that uniquely identifies a row that contains an XML column. This value is stored with the row and never changes.
- **FOREST**
An ordered set of subtrees of XML nodes.
- **FULLY ESCAPED MAPPING**
A mapping from an SQL identifier to an XML name when the SQL identifier is a column name.
- **XML ATTRIBUTE**
A name-value pair within a tagged XML element that modifies certain features of the element.
- **XML COLUMN**
A column of a table that stores XML values and is defined using the data type XML. The XML values that are stored in XML columns are internal representations of well-formed XML documents.
- **XML DATA TYPE**
A data type for XML values.
- **XML ELEMENT**
A logical structure in an XML document that is delimited by a start and an end tag. Anything between the start tag and the end tag is the content of the element.
- **XML INDEX**
An index on an XML column that provides efficient access to nodes within an XML document by providing index keys that are based on XML patterns.
- **XML LOCK**
A column-level lock for XML data. The operation of XML locks is similar to the operation of LOB locks.
- **XML NODE**
The smallest unit of valid, complete structure in a document. For example, a node can represent an element, an attribute, or a text string.
- **XML NODE ID INDEX**
An implicitly created index, on an XML table that provides efficient access to XML documents and navigation among multiple XML data rows in the same document.
- **XML PATTERN**
A slash-separated list of element names, an optional attribute name (at the end), or kind tests, that describe a path within an XML document in an XML column. The pattern is a restrictive form of path expressions, and it selects nodes that match the specifications. XML patterns are specified to create indexes on XML columns in a database.
- **XML PUBLISHING FUNCTION**
A function that returns an XML value from SQL values. An XML publishing function is also known as an XML constructor.
- **XML SCHEMA**
In XML, a mechanism for describing and constraining the content of XML files by indicating which elements are allowed and in which combinations. XML schemas are an alternative to document type definitions (DTDs) and can be used to extend functionality in the areas of data typing, inheritance, and presentation.
- **XML SCHEMA REPOSITORY (XSR)**
A repository that allows the DB2 database system to store XML schemas. When registered with the XSR, these objects have a unique identifier and can be used to validate XML instance documents.
 - XML serialization function



A function that returns a serialized XML string from an XML value.

- XML TABLE

An auxiliary table that is implicitly created when an XML column is added to a base table. This table stores the XML data, and the column in the base table points to it.

- XML TABLE SPACE

A table space that is implicitly created when an XML column is added to a base table. The table space stores the XML table. If the base table is partitioned, one partitioned table space exists for each XML column of data.

The XML docid

DB2 automatically generates the DocID column for each row that is loaded into a table with at least one XML column.

The DocID column is partially hidden. It is not included in the result set of a SELECT * statement. However, you can query this column by name and view information about this column and its index in the catalog. Several utilities report information on this column in their output.

SQL restrictions

Restrictions when using XML values: With a few exceptions, you can use XML values in the same contexts in which you can use other data type. XML values cannot be used in the following contexts:

- SELECT lists that are preceded by the DISTINCT clause
- GROUP BY clauses
- ORDER BY clauses
- A subselect of a fullselect with a set operator that is not UNION ALL
- Basic predicates, quantified predicates, BETWEEN predicates, DISTINCT predicates, IN predicates, or LIKE predicates
- Aggregate functions with the DISTINCT keyword
- Primary, unique, or foreign keys

XML and Utilities

Utility	supports XML	comments
CHECK DATA	yes	with different SCOPE's (e.g. SCOPE AUXONLY)
CHECK INDEX	yes	
COPY	yes	
COPYTOCOPY	yes	
EXEC SQL	no	see section 'Cross LOAD'ing XML data'
LISTDEF	yes	additional keyword ALL, BASE, LOB, XML
LOAD	yes	see section 'LOAD'ing XML data'
QUIESCE	yes	
REBUILD INDEX	yes	
RECOER	yes	
REORG	yes	some restrictions (no DISCARD, REBALANCE, UNLOAD EXTERNAL)
RUNSTATS TABLESPACE	yes	some restrictions (ignores COLGROUP, FREQVAL MOST LEAST BOTH, HISTOGRAM)
RUNSTATS INDEX	yes	some restrictions (ignores KEYCARD, FREQVAL MOST LEAST BOTH, HISTOGRAM)
UNLOAD	yes	some restrictions (no HISTOGRAM, KEYCARD, FREQVAL NUMCOLS COUNT)



UNLOAD'ing XML data

You can use the UNLOAD utility to unload XML data. In the UNLOAD statement, specify the base table space. (You do not have to specify the XML table space.) Also specify the XML keyword in the field specifications for the XML columns.

LOAD'ing XML data

You can use the LOAD utility to load XML data. The steps for loading XML data are similar to the steps for loading other types of data, except that you need to also perform the following actions:

- In the input data set:
 - If the data set is in delimited format, ensure that the XML input fields follow the standard LOAD utility delimited format.
 - If the data set is not in delimited format, specify the XML input fields similar to the way that you specify VARCHAR input. Specify the length of the field in a 2-byte binary field that precedes the data.
- In the LOAD statement:
 - Specify the keyword XML for all input fields of type XML.
 - If you want the whitespace to be preserved in the XML data, also specify the keywords PRESERVE WHITESPACE. By default, LOAD strips the whitespace.

Loading XML data with the LOAD utility has the following restrictions:

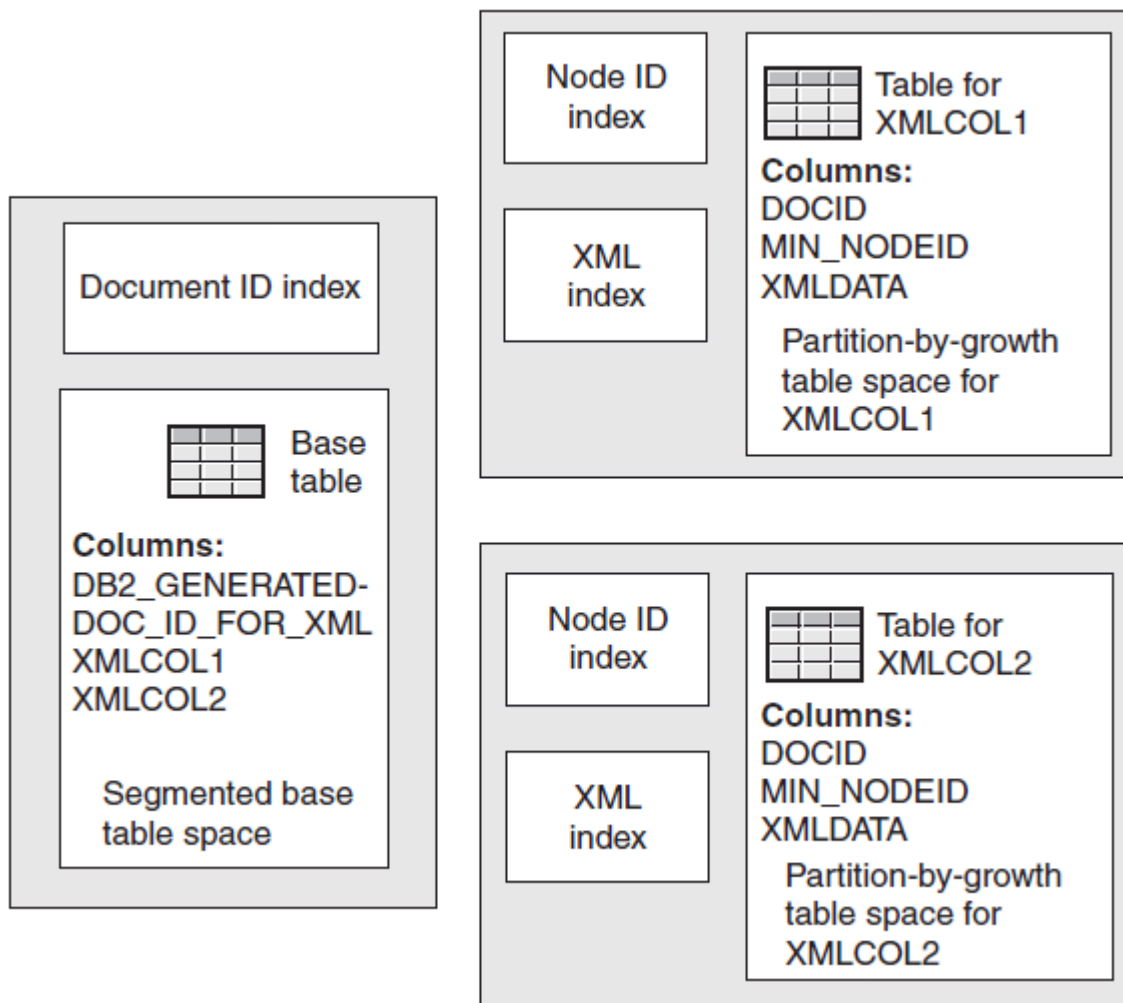
- You cannot specify that XML input fields be loaded into non-XML columns, such as CHAR or VARCHAR columns.
- DB2 does not perform any specified compression until the next time that you run the REORG utility on this data.
- DB2 ignores any specified FREEPAGE and PCTFREE values until the next time that you run the REORG utility on this data.
- If you specify PREFORMAT, DB2 preformats the base table space, but not the XML table space.
- You cannot directly load the DocID column of the base table space.
- You cannot specify a default value for an XML column.
- You cannot load XML values that are greater than 32 KB. To load such values, use file reference variables in the LOAD utility, or use applications with SQL XML AS file reference variables.

Cross LOAD'ing data

You cannot declare a cursor that includes XML data. Thus, you cannot use the DB2 UDB family cross-loader function to transfer data in XML columns. However, you can declare a cursor on a table with XML columns if the cursor does not include any XML columns.



Internal structure



DDL

```

CREATE
  TABLESPACE TSXML001
    IN DBXML
    USING STOGROUP SYSDEFLT
    PRIQTY -1
    SECQTY -1
    ERASE NO
    FREEPAGE 0
    PCTFREE 5
    GBPCACHE CHANGED
    COMPRESS YES
    TRACKMOD YES
    LOGGED
    DSSIZE 4G
    SEGSIZE 4
    MAXPARTITIONS 256
    BUFFERPOOL BP32K
    LOCKSIZE ROW
    LOCKMAX SYSTEM
    CLOSE YES
    CCSID EBCDIC
    MAXROWS 255
    
```

;



```

CREATE TABLE MYUSER.XMLT1 (
  ID BIGINT NOT NULL
  , SHORT XML
  , LONG XML
  , CONSTRAINT ID
  PRIMARY KEY (ID)
)
IN DBXML.TSXML001
CCSID EBCDIC
NOT VOLATILE
APPEND NO
;
CREATE UNIQUE INDEX MYUSER.XMLX1
ON MYUSER.XMLT1
(
  ID ASC
)
USING STOGROUP SGBWSYST
PRIQTY -1
SECQTY -1
ERASE NO
FREEPAGE 0
PCTFREE 10
GBPCACHE CHANGED
NOT CLUSTER
BUFFERPOOL BP3
CLOSE YES
COPY NO
PIECESIZE 4G
COMPRESS NO
;
    
```

Table Name	Database	Tblspace	Cols	PK	Type
MYUSER.XMLT1	DBXML	TSXML001	4	1	T
MYUSER.XXMLT1	DBXML	XXML0000	3	0	P
MYUSER.XXMLT1000	DBXML	XXML0001	3	0	P

Tablespaces

XML TABLESPACE SET REPORT:

```

TABLESPACE                : DBXML.TSXML001

BASE TABLE               : MYUSER.XMLT1
COLUMN                   : SHORT
  XML TABLESPACE        : DBXML.XXML0000
  XML TABLE              : MYUSER.XXMLT1
  XML NODEID INDEXSPACE  : DBXML.IRNODEID
  XML NODEID INDEX       : MYUSER.I_NODEIDXXMLT1
COLUMN                   : LONG
  XML TABLESPACE        : DBXML.XXML0001
  XML TABLE              : MYUSER.XXMLT1000
  XML NODEID INDEXSPACE  : DBXML.IRNO19HR
  XML NODEID INDEX       : MYUSER.I_NODEIDXXMLT1000
    
```