XML/XPath Support In MySQL-5.x

Alexander Barkov
Full time developer

April, 2005

MySQL AB

Copyright 2005 MySQL AB

Plan for this session

- Creating XML database
- Querying XML data using XPath
- Updating XML data
- Optimizing XPath queries
- Current status
- Future development directions

Copyright 2005 MySQL AB

Creating XML columns

An XML value can be stored in any textual data type:

- TEXT
- VARCHAR
- CHAR

No special XML type at this moment.

```
mysql> CREATE TABLE t1 (xml TEXT);
Query OK, 0 rows affected (0.04 sec)
```

Copyright 2005 MySQL AB

Inserting into an XML column

Using INSERT

```
mysql> INSERT INTO t1 VALUES
('<a>a1<b>b1</b>b2</a>');
Query OK, 1 rows affected (0.00 sec)
```

Using INSERT with LOAD_FILE:

```
mysql> insert into t1 select load_file
  ('/home/bar/example.xml');
Query OK, 1 row affected (0.00 sec)
```

Copyright 2005 MySQL AB

Fetching entire XML Values

```
mysql> select xml from t1;
+-----------------------+
| xml                   |
+-----------------------+
| <a>a1<b>b1</b>a2</a> |
+-----------------------+
1 row in set (0.00 sec)
```

Copyright 2005 MySQL AB

Fetching XML parts using XPath

New function:

`ExtractValue(xml, xpath)`

Returns a part of XML value *xml* addressed by an XPath query *xpath*.

Usage example:

```
mysql> select ExtractValue('<a>a1<b>b1</b>a2</a>',
  '/a/b');
+-----------------------------------------+
| ExtractValue('<a>a1<b>b1</b>a2</a>', '/a/b') |
+-----------------------------------------+
| b1                                      |
+-----------------------------------------+
```

Copyright 2005 MySQL AB

XPath implementation in MySQL

- According to W3C Recommendations
- Available from http://www.w3.org/TR/xpath
- Subset of XML Path Language Version 1.0
- About 90% currently implemented

Copyright 2005 MySQL AB

A more complex XML Example

```
<section>
  <title>Choosing an Install Package</title>
  <para>
    There are three different packages available that you can use to install the MySQL server:
  </para>
  <itemizedlist id=”noinstall”>
    <listitem>
      <para><literal>mysql-version-win-noinstall.zip</literal>: This is a Zip archive. It must be manually
extracted.
      </para>
    </listitem>
    <listitem id=”full”>
      <para><literal>mysql-version-win.zip</literal>: This package contains the InstallWizard with all
components of MySQL and is intended for a full installation with all optional packages.
      </para>
    </listitem>
    <listitem id=”essential”>
      <para><literal>mysql-version-essential-win.msi</literal>: This package contains the InstallWizard
 with the minimum components required to install a working MySQL server. Omitted packages can be
 added later.
      </para>
    </listitem>
  </itemizedlist>
  <para>
    For most users, the <literal>mysql-version-essential-win.msi</literal> package is recommended
 because of its smaller size and decreased download time.
  </para>
</section>
```

Copyright 2005 MySQL AB

Example#1: Simple query

```
<section>
 <title>Choosing an Install Package</title>
 <para>...three different packages...</para>
 <itemizedlist>
  <listitem id=”noinstall”>
   <para><literal>....noinstall.zip</literal>: ...Zip archive....</para>
  </listitem>
  <listitem id=”full”>
   <para><literal>...win.zip</literal>: ...full installation with all optional packages...</para>
  </listitem>
  <listitem id=”essential”>
   <para><literal>...win.msi</literal>: ...minimum components...</para>
  </listitem>
 </itemizedlist>
 <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml,'/section/title') from t1;
+-------------------------------------+
| ExtractValue(xml, '/section/title') |
+-------------------------------------+
| Choosing an Install Package         |
+-------------------------------------+
```

Copyright 2005 MySQL AB

Example#2: by index

```
<section>
  <title>Choosing an Install Package</title>
  <para>...three different packages...</para>
  <itemizedlist>
    <listitem id="noinstall">
      <para><literal>....noinstall.zip</literal>: ...Zip archive....</para>
    </listitem>
    <listitem id="full">
      <para><literal>...win.zip</literal>: ...full installation with all optional packages...</para>
    </listitem>
    <listitem id="essential">
      <para><literal>...win.msi</literal>: ...minimum components...</para>
    </listitem>
  </itemizedlist>
  <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml, '/section/para[1]') from t1;
+--------------------------------------+
| ExtractValue(xml, '/section/para[1]') |
+--------------------------------------+
| ...three different packages...        |
+--------------------------------------+
```

Copyright 2005 MySQL AB

Example#3: attribute

```
<section>
  <title>Choosing an Install Package</title>
  <para>...three different packages...</para>
  <itemizedlist>
    <listitem id="noinstall">
      <para><literal>....noinstall.zip</literal>: ...Zip archive....</para>
    </listitem>
    <listitem id="full">
      <para><literal>...win.zip</literal>: ...full installation with all optional packages...</para>
    </listitem>
    <listitem id="essential">
      <para><literal>...win.msi</literal>: ...minimum components...</para>
    </listitem>
  </itemizedlist>
  <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml,
'/section/itemizedlist/listitem[1]/@id')from t1;
+-----------------------------------------------------------+
| ExtractValue(xml, '/section/itemizedlist/listitem[1]/@id') |
+-----------------------------------------------------------+
| noinstall                                                 |
+-----------------------------------------------------------+
```

Example#4: by value

```
<section>
  <title>Choosing an Install Package</title>
  <para>...three different packages...</para>
  <itemizedlist>
    <listitem id="noinstall">
      <para><literal>....noinstall.zip</literal>: ...Zip archive....</para>
    </listitem>
    <listitem id="full">
      <para><literal>...win.zip</literal>: ...full installation with all optional packages...</para>
    </listitem>
    <listitem id="essential">
      <para><literal>...win.msi</literal>: ...minimum components...</para>
    </listitem>
  </itemizedlist>
  <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml,
'/section/itemizedlist/listitem/@id[contains(../para/literal,"noinstall.zip")]')
as value from t1;
+-----------+
| value     |
+-----------+
| noinstall |
+-----------+
```

Copyright 2005 MySQL AB

Example#5: boolean AND

```
<section>
  <title>Choosing an Install Package</title>
  <para>...three different packages...</para>
  <itemizedlist>
    <listitem id="noinstall">
      <para><literal>....noinstall.zip</literal>: ...Zip archive....</para>
    </listitem>
    <listitem id="full">
      <para><literal>...win.zip</literal>: ...full installation with all optional packages...</para>
    </listitem>
    <listitem id="essential">
      <para><literal>...win.msi</literal>: ...minimum components...</para>
    </listitem>
  </itemizedlist>
  <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml,
'/section/itemizedlist/listitem/@id[contains(../para/literal,"zip") and
   contains(../para,"Zip")]') as value from t1;
+-----------+
| value     |
+-----------+
| noinstall |
+-----------+
```

Example#6: boolean OR

```
<section>
  <title>Choosing an Install Package</title>
  <para>...three different packages...</para>
  <itemizedlist>
    <listitem id="noinstall">
      <para><literal>mysql-version-win-noinstall.zip</literal>: ...Zip archive....</para>
    </listitem>
    <listitem id="full">
      <para><literal>mysql-version-win.zip</literal>: ...full installation...</para>
    </listitem>
    <listitem id="essential">
      <para><literal>mysql-version-essential-win.msi</literal>: ...minimum components...</para>
    </listitem>
  </itemizedlist>
  <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml,
'/section/itemizedlist/listitem/para[contains(./literal,"zip") or contains(./literal,"win")]/literal')
as value from t1;
+-----------------------------------------------------------------------------------------+
| value                                                                                   |
+-----------------------------------------------------------------------------------------+
| mysql-version-win-noinstall.zip mysql-version-win.zip mysql-version-essential-win.msi |
+-----------------------------------------------------------------------------------------+
```

Copyright 2005 MySQL AB

Example#7: position()

```
<section>
  <title>Choosing an Install Package</title>
  <para>...three different packages...</para>
  <itemizedlist>
    <listitem id="noinstall">
      <para><literal>....noinstall.zip</literal>: ...Zip archive....</para>
    </listitem>
    <listitem id="full">
      <para><literal>mysql-version-win.zip</literal>: ...full installation...</para>
    </listitem>
    <listitem id="essential">
      <para><literal>...win.msi</literal>: ...minimum components...</para>
    </listitem>
  </itemizedlist>
  <para>...win.msi...package is recommended...because of its smaller size.</para>
</section>
```

```
mysql> select ExtractValue(xml,
'/section/itemizedlist/listitem[position()=2]/para/literal')
 as value from t1;
+-----------------------+
| value                 |
+-----------------------+
| mysql-version-win.zip |
+-----------------------+
```

Updating entire XML values

```
mysql> update t2 set
xml='<a>a1<b>b1<c>c1</c>b2</b></a>';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Copyright 2005 MySQL AB

Updating XML parts using XPath

New function:

`UpdateXML(xml, xpath, content)`

Returns a changed copy of XML value *xml* by replacing a node addressed by an XPath query *xpath* with a new content *content*.

Usage example:

```
mysql> select updateXML(
'<a>a1<b>b</b></a>','/a/b','<b>b1<c>c1</c>b2</b>'
) as xml;
+---------------------------------+
| xml                             |
+---------------------------------+
| <a>a1<b>b1<c>c1</c>b2</b></a> |
+---------------------------------+
```

Optimizing XPath queries: table structure

```
mysql> show create table t1;
+-------+----------------------------------------+
| Table | Create Table                           |
+-------+----------------------------------------+
| t1    | CREATE TABLE `t1` (                    |
|       |  `id` int(11) NOT NULL auto_increment, |
|       |  `xml` text NOT NULL,                  |
|       |  PRIMARY KEY  (`id`),                  |
|       |  FULLTEXT KEY `xml` (`xml`)            |
|       |) ENGINE=MyISAM                         |
+-------+----------------------------------------+
mysql> select count(*) from t1;
+----------+
| count(*) |
+----------+
|    65536 |
+----------+
```

Copyright 2005 MySQL AB

Optimizing XPath queries using FULLTEXT

Slow query:

```
mysql> select
ExtractValue(xml,'/section/itemizedlist/listitem[1]/para/literal')
from t1 where ExtractValue(xml,'/section/title')='NewTitle';
+-----------------------------------------------------------------------+
| ExtractValue(xml,'/section/itemizedlist/listitem[1]/para/literal') |
+-----------------------------------------------------------------------+
| mysql-version-win-noinstall.zip                                       |
+-----------------------------------------------------------------------+
1 row in set (4.84 sec)
```

Fast query:

```
mysql> select ExtractValue(xml,'/section/itemizedlist/listitem[1]/
para/literal') from t1 where ExtractValue(xml,'/section/title')
='NewTitle' and match (xml) against ('NewTitle');
+-----------------------------------------------------------------------+
| ExtractValue(xml,'/section/itemizedlist/listitem[1]/para/literal') |
+-----------------------------------------------------------------------+
| mysql-version-win-noinstall.zip                                       |
+-----------------------------------------------------------------------+
1 row in set (0.02 sec)
```

Copyright 2005 MySQL AB

Summary

Data types:

- Any character string

Handlers supporting XPath optimization:

- MyISAM

Supported SQL functions:

- ExtractValue()
- UpdateXML()

Supported XPath functionality:

- Absolute/Relative location path with full and abbreviated step syntax
- Axes: ancestor, self, parent, descendant, attribute, child
- Boolean (i.e. predicates) and numeric (by index) filters
- Booleans: AND, OR, NOT, =, !=, <=, >=, >, <, false(), true(), last()
- Numeric operators: +, -, *
- String functions: contains(), substring(), concat()
- Numeric functions: mod(), div(), ceiling(), floor(), round(), sum(), count(), position()
- Type conversion functions: boolean(), number()

Copyright 2005 MySQL AB

Current status

- Available as a separate patch for MySQL-5.0
- From: http://d.udm.net/~bar/myxml/mysql-xml.tar.gz
- No documentation yet
- Questions are welcome at <bar@mysql.com>
- This presentation:
  http://d.udm.net/~bar/myxml/XMLXpathSupportInMySQL.sxi

Copyright 2005 MySQL AB

Near Future TODO

- Push into the official source tree
- Implement 100% XPath Version 1.0
- Special XML data type, for validity checking of inserted values, and perhaps for more optimized storage (compression?)
- ExistsNode() - an optimized boolean function to check whether a node exists, without fetching its value
- A preprocessor for FULLTEXT to build indexes optimized for XML purposes
- Automatic invocation of FULLTEXT searches from inside ExtractValue() and ExistsNode() without having to use explicit MATCH operator

Copyright 2005 MySQL AB

Good Bye!