MySQL Data Formats & XML Support

Sang Shin
www.javapassion.com
"Learning is fun!"

Topics

- Exporting records
- Importing records
- Getting result in XML format from SELECT
- Reading XML file
- XPath tutorial
- XML functions
  - ExtractValue(..)
  - UpdateValue(..)
- Exporting a table to an XML file
- XML support in MySQL 6.0

Exporting Records

Exporting Records into a File

- Use SELECT statement with INTO OUTFILE (non-binary data) or INTO DUMPFILE for (binary data - contents of BLOB fields)
- Field termination through FIELDS TERMINATED
- Line termination through LINES TERMINATED

```
SELECT age, first_name FROM person
WHERE age > 30
INTO OUTFILE '/tmp/person_condition.txt'
FIELDS TERMINATED BY ':'
LINES TERMINATED BY '\r\n';
```

Exporting Records Example

```
mysql> select * from person;
+------------+--------------+-------------+-----+
| person_id | first_name | last_name | age |
+------------+--------------+-------------+-----+
|             1 | sang          | shin          |  88 |
|             2 | kelly         | jones         |  22 |
|             3 | jack          | kennedy     |  56 |
|             4 | paul          | kennedy     |  34 |
|             5 | daniel        | song          |  24 |
+------------+--------------+-------------+-----+
5 rows in set (0.00 sec)

C:\tmp>type person_condition.txt
88:sang
56:jack
34:paul
```

Importing Records

Importing Records

- Used to read large number of records into the MySQL database in a single batch
- Used to read data that is in different format

Importing Records from a File

- Use LOAD DATA INFILE <file-name> INTO TABLE <table-name>

```
/* When the contains data for all fields */
LOAD DATA INFILE '/tmp/person.txt'
INTO TABLE newperson
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\r\n';

/* When the file contains data only for certain set of fields */
LOAD DATA INFILE '/tmp/person.txt'
INTO TABLE newperson
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\r\n'
(person_id, first_name, last_name, age);
```

Importing Records from a file

- NULL value is inserted when a value is missing
- Optional clauses
  - LOW_PRIORITY: Runs the importing task when no other threads are accessing the table
  - CONCURRENT: Applies only to MyISAM table, Run the importing task while other threads are accessing the table
  - IGNORE: If any new records being imported has a duplicate primary, just ignore it
  - IGNORE LINES: Skip the specified number of lines at the beginning of the file

Getting Result in XML format from SELECT

Getting Result in XML from SELECT

- If you start “mysql” client with “--xml” option, the result of SELECT will be in XML format (instead of default table format)
  - C:\Users\sang>mysql --xml -u root -p

Getting Result of SELECT in XML

/* Assuming mysql command line tool is started with --xml option */
SELECT * FROM person WHERE person_id=1;

```
<?xml version="1.0"?>

<resultset statement="SELECT * FROM person WHERE person_id=1"
xmlns:xsi="http://www.w3.org/2001/XMLS
chema-instance">
  <row>
      <field name="person_id">1</field>
      <field name="first_name">sang</field>
      <field name="last_name">shin</field>
      <field name="age">88</field>
  </row>
</resultset>
```

Reading XML File

Reading XML File

- Use LOAD_FILE(<full-path-to-the-XML-file>) built-in function
  - It returns the file contents as a string - you can then set session variable or table field with the returned string
  - To use this function, the file must be located on the server host
  - The full path name to the file must be specified

Reading XML file

```
mysql> SET @xml = LOAD_FILE('/tmp/people.xml');
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @xml\G
*************************** 1. row ***************************
@xml: <?xml version="1.0"?>
<people>

   <person born="1912" died="1954">
      <name>
         <first_name>Alan</first_name>
         <last_name>Turing</last_name>
      </name>
      <profession>computer scientist</profession>
      <profession>mathematician</profession>
      <profession>cryptographer</profession>
   </person>

   ...

</people>
1 row in set (0.00 sec)
```

XPath Tutorial

Let's say we have the following XML

```
<?xml version="1.0"?>
<people>
   <person born="1912" died="1954">
      <name>
         <first_name>Alan</first_name>
         <last_name>Turing</last_name>
      </name>
      <profession>computer scientist</profession>
      <profession>mathematician</profession>
      <profession>cryptographer</profession>
   </person>

   <person born="1918" died="1988">
      <name>
         <first_name>Richard</first_name>
         <middle_initial>M</middle_initial>
         <last_name>Feynman</last_name>
      </name>
      <profession>physicist</profession>
      <hobby>Playing the bongoes</hobby>
   </person>
</people>
```

XPath Expression Examples

- /people/person[1]/name/first_name
  - Get all “first_name” nodes under all “name” nodes of the 1st “person” node under all “people” node
  - Result: <first_name>Alan</first_name>
- /people/person[2]/name/first_name
  - Result: <first_name>Richard</first_name>
- //first_name
  - Get all “first_name” nodes under all parent nodes
  - Result: <first_name>Alan</first_name>, <first_name>Richard</first_name>
- //first_name/text()
  - Get text value of all “first_name” under all parent nodes
  - Result: Alan Richard

XPath Expression Examples

- /people/person/profession[2]
  - Get the 2nd profession node under all person nodes under all people nodes
  - Result: <profession>mathematician</profession>
- /people/person/profession[1]
  - Get the 1st profession nodes under all person nodes under all people nodes
  - Result: <profession>computer scientist</profession>,<profession>physicist</profession>
- /people/person[2]/profession[1]
  - Get the first profession node of the 2nd person node under all people nodes
  - Result: <profession>physicist</profession>

XPath Expression Examples

- //profession
  - Get all profession nodes
  - Result: 4 of <profession>...</profession>
- //profession[1]
  - Get 1st profession nodes among all possible parent nodes
  - Result: <profession>computer scientist</profession>, <profession>physicist</profession>
- /people/person[@born=1918]/profession
  - Get all profession nodes of all person nodes whose born attribute is set to 1918 under all people nodes
  - Result: <profession>physicist</profession>

XML Functions

XML Functions

- MySQL 5.1 introduced two new built-in functions for XML processing
  - *ExtractValue(...)* for retrieving XML elements
  - *UpdateXML(...)* for updating XML elements
- These functions use XPath expressions for referencing XML elements

XML Functions:
ExtractValue(...)

ExtractValue(...)

- ExtractValue(xml_frag, xpath_expr)
  - xml_frag: a fragment of XML markup
  - xpath_expr: XPath expression
- Returns the text of the first text node which is a child of the element(s) matched by the XPath expression
- It is the equivalent of performing a match using the xpath_expr after appending /text().
  - ExtractValue('<a>Sakila', '/a/b') and ExtractValue('<a>Sakila', '/a/b/text()') produce the same result.

ExtractValue(...) Examples

```
mysql> SELECT ExtractValue('<a><b>xx</b></a>', '/a/b/text()');
+-------------------------------------------------------------+
| ExtractValue('<a><b>xx</b></a>', '/a/b/text()') |
+-------------------------------------------------------------+
| xx                                                          |
+-------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT ExtractValue('<a><b>xx</b></a>', '/a/b');
+-------------------------------------------------------------+
| ExtractValue('<a><b>xx</b></a>', '/a/b')          |
+-------------------------------------------------------------+
| xx                                                          |
+-------------------------------------------------------------+
1 row in set (0.00 sec)
```

ExtractValue(...) Examples

```
mysql> SELECT ExtractValue('<a><b>xx</b></a>', '/a/b')
    -> AS value;
+-------+
| value |
+-------+
| xx    |
+-------+
1 row in set (0.00 sec)

mysql> SELECT ExtractValue('<a><b>xx</b></a>', '//b');
+-----------------------------------------------------------+
| ExtractValue('<a><b>xx</b></a>', '//b')                   |
+-----------------------------------------------------------+
| xx                                                        |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

ExtractValue(...) Examples

```
mysql>  SELECT ExtractValue(@xml,
'/people/person[1]/name/first_name');
+-----------------------------------------------------------------------+
| ExtractValue(@xml, '/people/person[1]/name/first_name') |
+-----------------------------------------------------------------------+
| Alan                                                                  |
+-----------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>  SELECT ExtractValue(@xml,
'/people/person[2]/name/first_name');
+-----------------------------------------------------------+
| ExtractValue(@xml, '/people/person[2]/name/first_name') |
+-----------------------------------------------------------+
| Richard                                                   |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

XML Functions:
UpdateValue(...)

UpdateValue(...)

- UpdateXML(xml_target, xpath_expr, new_xml)
  - xml_target: a given fragment of XML markup to be replaced with new_xml
  - xpath_expr: XPath expression
  - new_xml: new xml fragment that replaces the xml_target

ExtractValue(...) Examples

```
mysql>  SELECT ExtractValue(@xml,
'/people/person[1]/name/first_name');
+-----------------------------------------------------------------------------+
| ExtractValue(@xml, '/people/person[1]/name/first_name') |
+-----------------------------------------------------------------------------+
| Alan                                                                        |
+-----------------------------------------------------------------------------+

mysql> SET @xml = UpdateXML (@xml,
    -> '/people/person[1]/name/first_name',
    -> '<first_name>Sang</first_name>');
Query OK, 0 rows affected (0.00 sec)

mysql>  SELECT ExtractValue(@xml,
'/people/person[1]/name/first_name');
+-----------------------------------------------------------------------------+
| ExtractValue(@xml, '/people/person[1]/name/first_name') |
+-----------------------------------------------------------------------------+
| Sang                                                                        |
+-----------------------------------------------------------------------------+
```

Exporting a table to an XML file

Exporting a Table to XML file

- In MySQL 5.1, there is no equivalent of SELECT INTO OUTFILE for XML.
  - The only way that work is starting MySQL client with "--xml" option or using mysqldump with "--xml" option.
- mysqldump --xml -u root -p mydb person > /tmp/myperson.xml
- mysql --xml -u root -p --execute="SELECT * FROM person" mydb > /tmp/myperson2.xml

XML Support in MySQL 6.0

New XML Features in MySQL 6.0

- Storing data from XML in a MySQL database using the LOAD XML statement
- Some new XPath functions will be added

Thank you!

Sang Shin
http://www.javapassion.com
“Learning is fun!”