Table of Contents

We have been using the mysql command line primarily to either start the mysql client or to run the script-to-spool process. And in unit 01 we discussed several options for the mysql command. There are many other options and one you may see in discussions on MySQL and MXL is the `--xml` option.

1.1. using the --xml option to start a mysql session

Start mysql including the --xml option. My command line would be:

```
mysql -u a_rose -p --xml
```

Then switch to the a_testbed and display the first three rows from the zoo_animals table - or another table you have.

```
use a_testbed;

select * from zoo_animals limit 3;
```

This is the output

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

<resultset statement="select * from zoo_animals limit 3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
        <field name="an_id">23</field>
        <field name="an_name">Sam</field>
        <field name="an_type">Giraffe</field>
        <field name="an_cost">5000.00</field>
        <field name="an_dob">2002-05-15 10:45:00</field>
        <field name="an_aquired">2002-05-15</field>
  </row>

  <row>
        <field name="an_id">25</field>
        <field name="an_name">Abigail</field>
        <field name="an_type">Armadillo</field>
        <field name="an_cost">490.00</field>
        <field name="an_dob">2010-01-15 09:00:00</field>
        <field name="an_aquired">2010-04-15</field>
  </row>

  <row>
        <field name="an_id">56</field>
        <field name="an_name">Leon</field>
        <field name="an_type">Lion</field>
        <field name="an_cost">5000.00</field>
        <field name="an_dob">2009-03-25 00:00:00</field>
        <field name="an_aquired">2010-03-25</field>
  </row>
</resultset>
3 rows in set (0.03 sec)
```

Looking again at the output of this select. The root of the document is the <resultset> element and it contains three <row> elements- one for each of the three rows allowed by the limit clause.

Each of the <row> elements contains <field> elements with a name attribute; that attribute has a value that is the column/attribute name from the table. The <field> element has a value which is the value from the table.

That is kind of nice but suppose we run the following command- `show tables Like 'zoo%'`. In a regular mysql command line client we would get

```
+----------------------------+
| Tables_in_a_testbed (zoo%) |
+----------------------------+
| zoo_animals                |
| zoo_ex                     |
+----------------------------+
2 rows in set (0.00 sec)
```

But in the client started with the --xml option that command's output is

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

<resultset statement="show tables like 'zoo%'"
xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance">
  <row>
        <field name="Tables_in_a_testbed (zoo%)">zoo_animals</field>
  </row>

  <row>
        <field name="Tables_in_a_testbed (zoo%)">zoo_ex</field>
  </row>
</resultset>
2 rows in set (0.00 sec)
```

Running the client with the --xml option means that the output is cast to XML.

The XML document makes sense and is a useful result. We just don't really want to run the client in this mode when we are doing interactive work.

1.2. Using the --xml option for creating an xml file

A more useful approach would be to put the query in a script file and send the output to a file.

This is the command line model you have been using for assignments

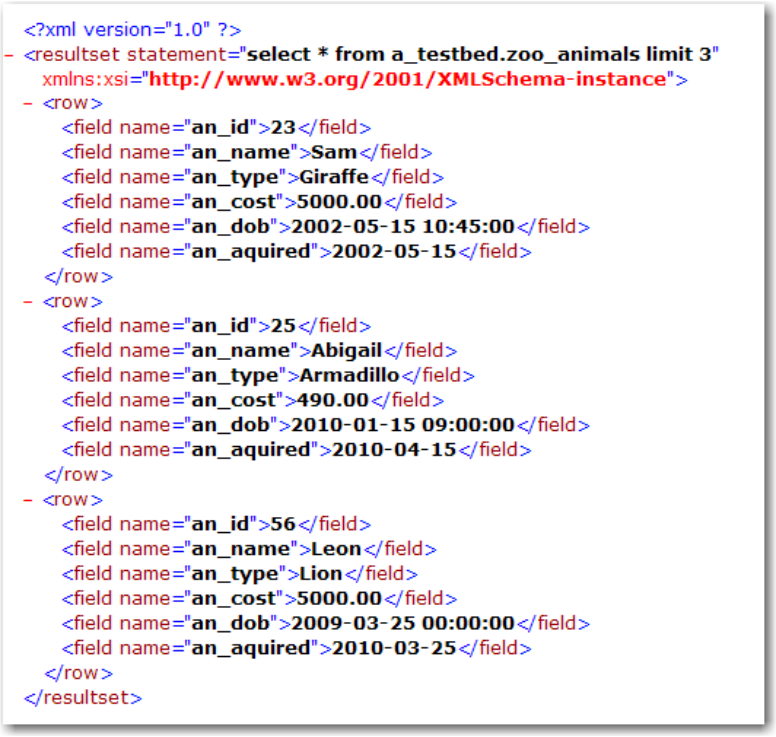
```
mysql -u a_rose -p -vvv --force --comments <C:/myscript.sql >C:/myresult.lst
```

We can modify this a bit for our purpose.

- skip the -vvv option for verbose ; we do not want to see the query statement
- skip the --comments option; we do not want to see comments in the result
- skip the --force option; we have only one query and it either works or it doesn't.
- You can name the output file with the xml file extension if you want.
- add the --xml option to get xml output

```
mysql -u a_rose -p  --xml  <C:/myscript.sql >C:/myresult.xml
```

This is the result file opened in a browser. So this a way to get an XML file from a table.



```
<?xml version="1.0" ?>
- <resultset statement="select * from a_testbed.zoo_animals limit 3"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <row>
      <field name="an_id">23</field>
      <field name="an_name">Sam</field>
      <field name="an_type">Giraffe</field>
      <field name="an_cost">5000.00</field>
      <field name="an_dob">2002-05-15 10:45:00</field>
      <field name="an_aquired">2002-05-15</field>
    </row>
  - <row>
      <field name="an_id">25</field>
      <field name="an_name">Abigail</field>
      <field name="an_type">Armadillo</field>
      <field name="an_cost">490.00</field>
      <field name="an_dob">2010-01-15 09:00:00</field>
      <field name="an_aquired">2010-04-15</field>
    </row>
  - <row>
      <field name="an_id">56</field>
      <field name="an_name">Leon</field>
      <field name="an_type">Lion</field>
      <field name="an_cost">5000.00</field>
      <field name="an_dob">2009-03-25 00:00:00</field>
      <field name="an_aquired">2010-03-25</field>
    </row>
</resultset>
```

Suppose our script query is

```
select title, author_name_last as author
from a_books.books
join a_books.book_authors  using (book_id)
join a_books.authors  using (author_id)
where a_books.book_authors.author_sequence = 1
order by author_name_last, title;
```

These are the first few lines from the result file. Note that the column alias was used as the value for the field name. and We get one sub-element per row.

```
<?xml version="1.0" ?>
- <resultset statement="select title, author_name_last as author from a_books.books join a_books.book_authors using (book_id) join
    a_books.authors using (author_id) where a_books.book_authors.author_sequence = 1 order by author_name_last, title"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <row>
      <field name="title">Intro to DB Systems-7th Ed</field>
      <field name="author" />
    </row>
  - <row>
      <field name="title">Temporal Data and the Relational Model</field>
      <field name="author" />
    </row>
  - <row>
      <field name="title">Beginning VB 2008 Databases</field>
      <field name="author">Agarwal</field>
    </row>
  - <row>
      <field name="title">Relational Database Theory</field>
      <field name="author">Atzeni</field>
    </row>
  - <row>
      <field name="title">Practical Standards for VB.NET</field>
      <field name="author">Balena</field>
    </row>
  - <row>
      <field name="title">Programming Visual Basic 2005: The Language</field>
      <field name="author">Balena</field>
    </row>
  - <row>
      <field name="title">Developer's Guide to SQL Server 2005</field>
      <field name="author">Beauchemin</field>
    </row>
  - <row>
      <field name="title">T_SQL Programming (Inside series)</field>
      <field name="author">Ben-Gan</field>
    </row>
  - <row>
      <field name="title">T_SQL Querying (Inside series)</field>
      <field name="author">Ben-Gan</field>
    </row>
```