

2011

AALL

★ July 23-26

Annual Meeting & Conference

Philadelphia★

Coding Potpourri: MySQL

Nicole C. Engard

Director of Open Source Education, ByWater Solutions

Author of *Practical Open Source Software for Libraries*

My What?

- MySQL: My Structured Query Language
- Relational Database Management System
- Licensed with the GNU GPL
 - That means it's Open Source
- Used to organize data in a structured way

Who's Using It?

- MySQL is the database behind many popular web based applications
 - Wordpress
 - Drupal
 - Wikipedia
 - Facebook
 - YouTube
 - Flickr
 - Ebay
 - Google (not searches)

Reading Table Structure

```
CREATE TABLE `branches` (  
  `branchcode` varchar(10) NOT NULL default "",  
  `branchname` mediumtext NOT NULL,  
  `branchaddress1` mediumtext,  
  `branchaddress2` mediumtext,  
  `branchaddress3` mediumtext,  
  `branchphone` mediumtext,  
  `branchfax` mediumtext,  
  `branchemail` mediumtext,  
  UNIQUE KEY `branchcode` (`branchcode`)  
)
```

Reading Table Structure

- This table's name is 'branches' and it stores the information about libraries or branches in Koha.
- Each field is easy to identify because of its name (ex. branchname is the library name).
- A field with a number in parens after it is a field that is limited in size.
 - For example varchar(10) means the field can have no more than 10 characters in it
- Lastly, we see that 'branchcode' is the unique key or unique identifier in the table.

Inserting Data

- Using the *branches* table enter a branch's info

```
INSERT INTO branches (branchcode,  
branchname, branchaddress1,  
branchaddress2, branchphone, branchemail)  
VALUES ('LIB', 'My Library', '123 Library  
Road', 'Philadelphia, PA', '215.555.1234',  
'info@library.com');
```

Querying MySQL

- To query a single table you will structure your query like this:
- `SELECT column_names FROM table_name [WHERE ...conditions] [ORDER BY ...conditions];`
- Statements in brackets are optional
- You can also select everything in a table by using an `*` in place of `column_names`

Querying MySQL

- Using the *branches* table let's query the data
- This query will pull out only the Branch Names and Emails and put them in ascending order by name

```
SELECT branchname, branchemail FROM  
branches ORDER BY branchname ASC;
```

```
+-----+-----+  
| branchname      | branchemail      |  
+-----+-----+  
| My Library      | info@library.com |  
| Nicole's Library | nengard@bywatersolutions.com |  
+-----+-----+
```

Querying MySQL

- Using *branches* let's get the address info from one specific branch

```
SELECT branchname, branchaddress1,  
branchaddress2, branchaddress3, branchemail  
FROM branches WHERE branchcode='LIB';
```

```
+-----+-----+-----+-----+-----+  
| branchname | branchaddress1 | branchaddress2 | branchaddress3 | branchemail |  
+-----+-----+-----+-----+-----+  
| My Library | 123 Library Road | Philadelphia, PA | | info@library.com |  
+-----+-----+-----+-----+-----+
```

Querying MySQL

- Using *branches* let's pull out the branch's phone and fax in one column

```
SELECT CONCAT('ph. ', branchphone, ' fax ',  
branchfax) as 'contact info' FROM branches  
WHERE branchcode= 'LIB';
```

```
+-----+  
| contact info |  
+-----+  
| ph. 215.555.1234 fax 215.555.2233 |  
+-----+
```

Another Table

```
CREATE TABLE `issues` (  
  `borrowernumber` int(11) default NULL,  
  `itemnumber` int(11) default NULL,  
  `date_due` date default NULL,  
  `branchcode` varchar(10) default NULL,  
  `returndate` date default NULL,  
  `return` varchar(4) default NULL,  
  `renewals` tinyint(4) default NULL,  
  `issuedate` date default NULL,  
  KEY `issuesborridx` (`borrowernumber`),  
  KEY `issuesitemidx` (`itemnumber`),  
)
```

Joining Tables

- Using *branches* and *issues* let's find out how many items circulated at each branch in a specific time period

```
SELECT b.branchname, count(i.branchcode) as  
count FROM issues i LEFT JOIN branches b ON  
(i.branchcode=b.branchcode) WHERE i.issuedate  
BETWEEN '2011-06-01' AND '2011-07-01' GROUP  
BY b.branchcode ORDER BY count DESC;
```

```
+-----+-----+  
| branchname      | count |  
+-----+-----+  
| Nicole's Library |      7 |  
| My Library      |      1 |  
+-----+-----+
```

Date & Time Functions

- The most common use for reports is for end of the month or end of the year statistics
- The MySQL manual on Date & Time functions is essential for these queries
- <http://dev.mysql.com/doc/refman/5.6/en/date-and-time-functions.html>

String Functions

- Often you want to join two or more strings together, find a part of a string or even change a part of a string
- String functions are defined in the MySQL manual
- <http://dev.mysql.com/doc/refman/5.6/en/string-functions.html>

Math Functions

- Using MySQL you can get fancy with statistics by using the number related functions
- The Numeric Functions section of the manual can help you here
- <http://dev.mysql.com/doc/refman/5.6/en/numeric-functions.html>

Common Functions

- `COUNT(FIELD)` or `SUM(FIELD)`
 - Counts the number of or adds up the total value of results in a column
- `CURDATE()`
 - Is the current date (not time, just date)
- `MONTH(FIELD)` and `YEAR(FIELD)`
 - Return the month and year from a field
- `DATE_SUB(DATE, INTERVAL)`
 - Subtract a period of time from a date

2011

AALL

★ July 23-26

Annual Meeting & Conference

Philadelphia★

Thank You

Nicole C. Engard

nengard@gmail.com

Slides: web2learning.net > Publications &
Presentations