

# SQLite

A light, fast and transactional DBMS  
embedded in PHP 5

# **PHP Conference Québec 2006**

Zak Greant (zak@ez.no)

Managing Director, North America  
eZ systems

# Overview

- Transactional (ACID)
- Mostly Typeless (v2) / Advisory Typing (v3)
- Small memory footprint (250 KiB or less)
- Databases == files (or blocks of memory)
- Database (or psuedo-table level) locking
- No configuration
- No access controls
- Much of SQL92

# (Non-)Licensing

**\*\* The author disclaims copyright to this material.**

**\*\* In place of a legal notice, here is a blessing:**

**\*\***

**\*\* May you do good and not evil.**

**\*\* May you find forgiveness for yourself and forgive others.**

**\*\* May you share freely, never taking more than you give.**

# When to Use SQLite? (for PHP 5)

- Data store for stand-alone apps
- Moderate read (low write) traffic (<20 queries/second avg. on commodity hardware)
- More reads than writes
- In short, 90+% of apps

# SQLite v3

## Enhancements

- A more compact format for database files
- Manifest typing
- BLOB support
- Support for both UTF-8 and UTF-16 text
- User-defined text collating sequences
- 64-bit ROWIDs
- Improved Concurrency

# Mostly Typeless

```
CREATE TABLE book (author, title);  
INSERT INTO book (author, title)  
VALUES ('MySQL', 'Paul DuBois');  
INSERT INTO book (author, title)  
VALUES (1, 2);  
  
# In 2.x, except for INTEGER PRIMARY KEY  
# columns  
  
# In 3.x, there is pseudo-typing
```

# Manifest Type

```
CREATE TABLE who (name CHAR, age INT);  
INSERT INTO who (name, age)  
VALUES ('Zak', 33);  
INSERT INTO who (name, age)  
VALUES ('Rasmus', 'Round up to  
Fourty');  
# Works, but 33 is stored as a native  
INT
```



# Transactional + Multi-Query PHP API

```
$db->query("
    BEGIN;
    CREATE TABLE book (author, title);
    INSERT INTO book (author, title)
        VALUES ('MySQL', 'Paul DuBois');
    INSERT INTO book (author, title)
        VALUES (1, 2);
    COMMIT;
");
```

# Open or Create I/II

- In PHP, w/ SQLite extension:

```
$path = '/path/to/db/file.sqlite';  
# $path = ':memory:';  
$db = new SQLiteDatabase($path);
```

- In PHP, w/ PDO:

```
$dbh = new PDO("sqlite:$path");
```

# Open or Create II/II

- Command Line:

```
sqlite file.sqlite
```

```
echo .quit | sqlite file.sqlite
```

- Note: You have to install SQLite separately from PHP, if you want the command line tool.

# Getting Results I/IV

```
$h= $db->query("SELECT * FROM book;");
if( 0 != $db->lastError() ){
    # error handling
} else {
    while($row = $h->fetch(SQLITE_ASSOC)) {
        echo join($row, "\t"), "\n";
    }
}

# MySQL      Paul DuBois
# 1          2
```

# Getting Results II/IV

```
$h= $db->unbufferedQuery("SELECT * FROM  
book;", SQLITE_ASSOC);
```

```
foreach($h as $row){  
    echo join($row, "\t"), "\n";  
}
```

```
# MySQL      Paul DuBois  
# 1          2
```

# Getting Results III/IV

```
$h= $db->query("SELECT * FROM books;");  
$r= $h->fetchAll(SQLITE_ASSOC);  
print_r($r);
```

```
# Array  
# (  
#     [0] => Array  
#         (  
#             [author] => MySQL  
#             [title] => Paul DuBois  
#         )  
#     ...
```

# Getting Results IV/IV

- Using PDO:

```
$query = 'SELECT * FROM books;'  
foreach($dbh->query($query) as $row) {  
    print_r($row);  
}
```

# Creating Indexes

```
CREATE INDEX name ON table (column,...);
```

```
CREATE TABLE n (i INTEGER PRIMARY KEY);
```

```
CREATE TABLE n (c UNIQUE);
```



# AUTO\_INCREMENT

- Every row has an auto-generated rowid. Just ask for ROWID to get it.

```
SELECT ROWID FROM book;
```

- Use INTEGER PRIMARY KEY to be able to manipulate the ROWID

# Command Line Use

```
cat <<EOI | sqlite file.sqlite  
    SELECT * FROM book;  
    SELECT * FROM sqlite_master;  
EOI
```

```
sqlite file.sqlite < commands.sql
```

# Performance Tips

- Indexes
- BEGIN/END TRANSACTION
- PRAGMA synchronous=OFF
- in-memory database

# Gotchas

- A few gotchas
  - Limited ALTER TABLE support (RENAME TABLE and ADD COLUMN only)
  - VACUUM to compact databases with deleted rows (or set PRAGMA auto\_vacuum)
  - SQLite v2 and v3 file formats are not directly interoperable

# More Information

- <http://php.net/sqlite>
- <http://sqlite.org>
- <http://www.zend.com/php5/articles/php5-sqlite.php>
- Slides at <http://zak.greant.com>

**Questions?**

**Thanks!**

- Abstraction layers
- Performance numbers
- Important SQL limitations
- Stored procs
- Prepared statements
- Compiled statements
- PDO support