

# Structured SQL Programming

By: *Brian Rosenthal and Zvi Boshernitzan*

Copyright 2006 Brian Rosenthal and Zvi Boshernitzan

## I. Overview

Many people program SQL as they might treat HTML – as a big string that they build up and then execute.

There are a number of problems with this approach:

- (1) the approach makes it more difficult to protect against SQL injection,
- (2) the approach makes it more difficult to have many different criteria build up vastly different SQL (mainly because you might add the same inner join more than once).

SQL, after all, stands for “structured query language”, and the following describes a way of treating SQL as a structured object, until execution that has worked elegantly for the authors.

## II. The Class: **dbquery**

The idea is that you instantiate a new “dbquery” object, alter it in different ways, and then export it to either a string or an associative array.

### ***Initialization***

The standard use case is initializing a new dbquery instance from an associative array:

```
$q = new dbquery({ 'select': ['firstname', 'lastname']  
                  , 'from': 'customers c'  
                  , 'ijoin': { 's': { 'table': 'sites', 'on': 's.id = c.site_id' } }  
                  , 'where': [ "c.name like '%brian%'", "c.id > 10" ]  
                  , 'limit': 10  
                  , 'offset': 5  
                  });
```

### ***Methods***

Methods that add or set parts of the SQL statement:

```
$q->add_select(...), $q->set_select(...)  
$q->set_from(...)  
$q->add_ijoin(...)  
$q->add_lojoin(...)
```

```

$q->add_where(...)
$q->set_limit(...)
$q->set_offset(...)
$q->add_order_by(...), $q->set_order_by(...)
$q->add_group_by(...), $q->set_group_by(...)

$q->load_dict({ 'select': 'c.address_id', 'where': ... })

```

Methods that allow the dbquery to be exported:

```

$q->to_sql()
$q->to_dict()

```

## ***How it looks***

Get a set of customers:

```

$q = new dbquery({ 'select': ['firstname', 'lastname']
                  , 'from': 'customers c'
                  , 'where': ["c.name like '%brian%'", "c.id > 10"]
                  , 'limit': 10
                  , 'offset': 5
                  });

// only include the site ijoin if the condition is specified.
if ($site_name) {
    $q->add_ijkoin({ 's': { 'table': 'sites', 'on': 's.id = c.site_id' } });
    $q->add_where({ 's.name': $site_name });
}

$customers = db::assocs($q->to_sql())

```

## **III. Summary**

This article has outlined a methodology for treating a SQL Query as a well-defined structure until it needs to be evaluated, such that SQL queries may be combined and built up based on a variety of criteria.