

### Open database connection – once per script

```
$dbh = db_connect();
```

### Basic query

```
$user_info_q = "select * from $users_table where user_id=:user_id";
```

```
$parameters = array(
    array('P' => ':user_id', 'V'=> $user_id, 'T' => 'I')
);
```

```
$user_result = db_single_prepared_query($user_info_q, $parameters);
```

If a parameter is used more than once it must be declared with unique placeholders

```
$user_info_q = "select * from $users_table where user_id=:user_id or friend_id=user_id_f";
```

```
$parameters = array(
    array('P' => ':user_id', 'V'=> $user_id, 'T' => 'I'),
    array('P' => ':user_id_f', 'V'=> $user_id, 'T' => 'I')
);
```

Parameters array types (T)

I	Integer
S	String
N	Null

For dates which may be NULL, add type S or N to \$parameters as required, do not try to insert NULL as a string.

No need to use addslashes()

Returning results

SELECT, INSERT, DELETE

```
$field= $user_result['ROWS'][$row_index]['field'];
```

\$row\_index will typically be 0, or \$i in a loop

SELECT, INSERT, DELETE

```
$numrows = $user_result['ROWCOUNT']; // Rows in result, affected or deleted
```

INSERT

```
$last_insert_id = $user_result['LASTID'];
```

### Looped queries

Prepare query outside loop, with no parameters:

```
$ins_q = "insert into $ table values (:search_data_id,:access_group_id, :page_id,:live)";
```

```
$parameters=array();
```

Specify a unique name for the PDOStatement object used by db\_looped\_prepared\_query() so it can re-connect to the correct query during the loop

```
db_looped_prepared_query($ins_q, 'insaccess', $parameters, true);
```

Inside loop:

Set parameters within loop and call db\_looped\_prepared\_query() each time:

```
$parameters = array(  
    array('P' => ':page_id', 'V'=> $page_id, 'T' => 'I'),  
    array('P' => ':page_id2', 'V'=> $page_id, 'T' => 'I')  
);
```

```
$access_groups = db_looped_prepared_query($access_info_q, 'accessinfo', $parameters);
```

### Transactions

```
$dbh->beginTransaction();
```

```
$dbh->commit();
```

## Doing LIKE queries

```
$linked_query = "select distinct($pages_table.title)
  from $pages_table left join $page_fragment_contents_table using (page_id)
 where (int_link_id = :page_id OR image_1_int_link = :image_1_int_link
       OR image_2_int_link = :image_2_int_link
       OR title_1_int_link = :title_1_int_link
       OR text_1 like :text_1_page_id
       OR text_2 like :text_2_page_id
       OR text_3 like :text_3_page_id)";
```

```
// need to pass this as a string
$for_like_matching_page_id = '%/page/' . $page_id . '%';
```

```
$parameters = array(
    array('P' => ':page_id', 'V'=> $page_id, 'T' => 'I'),
    array('P' => ':image_1_int_link', 'V'=> $page_id, 'T' => 'I'),
    array('P' => ':image_2_int_link', 'V'=> $page_id, 'T' => 'I'),
    array('P' => ':title_1_int_link', 'V'=> $page_id, 'T' => 'I'),
    array('P' => ':text_1_page_id', 'V'=> $for_like_matching_page_id, 'T' => 'S'),
    array('P' => ':text_2_page_id', 'V'=> $for_like_matching_page_id, 'T' => 'S'),
    array('P' => ':text_3_page_id', 'V'=> $for_like_matching_page_id, 'T' => 'S')
);
```

Results in:

```
Prepareselect distinct(pages.title)
from  pages left join page_fragment_contents using (page_id)
where (int_link_id = ? OR image_1_int_link = ?
       OR image_2_int_link = ?
       OR title_1_int_link = ?
       OR text_1 like ?
       OR text_2 like ?
       OR text_3 like ?)
```

```
Execute      select distinct(pages.title)
from  pages left join page_fragment_contents using (page_id)
where (int_link_id = 339 OR image_1_int_link = 339
       OR image_2_int_link = 339
       OR title_1_int_link = 339
       OR text_1 like '%/page/339%'
       OR text_2 like '%/page/339%'
       OR text_3 like '%/page/339%')
```

## WHERE IN queries (Integers)

Must create SQL like : *where fragment\_id in (:fid\_0, :fid\_1, :fid\_2) order by sequence*

Pass bindings like:

```
Array
(
    [0] => Array
        (
            [P] => :fid_0
            [V] => 3526
            [T] => I
        )

    [1] => Array
        (
            [P] => :fid_1
            [V] => 3540
            [T] => I
        )

    [2] => Array
        (
            [P] => :fid_2
            [V] => 3527
            [T] => I
        )
)
```

Execute: *where fragment\_id in (3540,3527,3528) order by sequence*

With list of IDs in an array:

```
$parameters = array();
$placeholders_array = array();

for ($i=0;$i<count($unique_move_fragment_id_array);$i++)
{
    $fragment_id = $unique_move_fragment_id_array[$i];
    $place_holder = ":fid_" . $i;
    $placeholders_array[]=$place_holder;

    $parameters[] = array('P' => $place_holder, 'V'=> $fragment_id, 'T' => 'I');
}

$placeholders_str = implode (" , ", $placeholders_array);
```

If in a CSV string use *pdo\_in\_clause\_bindings\_ints()*