



Drupal 7 Quick Database Reference

Functions

db_delete(\$table, array \$options = array())

Returns a new DeleteQuery object for the active database.

```
db_delete('node')->condition('nid', $nid)->execute();
```

db_query(\$query, \$args = array(), \$options = array())

Execute an arbitrary query string against the active database.

```
$result = db_query('SELECT nid FROM {my_table}
WHERE nid = :nid', array('nid' => $edit['nid']));
```

pager_query(\$query, \$limit = 10, \$element = 0, \$count_query = NULL)

Use this function when doing select queries you wish to be able to page.

```
$sql = 'SELECT * FROM {url_alias} WHERE dst LIKE :keys';
$header = array(
  array('data' => t('Alias'), 'field' => 'dst', 'sort' => 'asc'),
  array('data' => t('System'), 'field' => 'src'),
  array('data' => t('Operations'), 'colspan' => '2')
);
$sql .= tablesort_sql($header);
$result = pager_query($sql, 50, 0, NULL, array(':keys' => '%'.
$keys .'%'));
```

db_select(\$table, \$alias = NULL, array \$options = array())

Returns a new SelectQuery object, add conditions and execute.

```
$result = db_select('menu_links')
->fields('menu_links', array('menu_name'))
->condition('link_path', $link_path)
->condition('customized', 0)
->condition('module', $module)
->execute();
```

db_update(\$table, array \$options = array())

Returns a new UpdateQuery object and change the uid column in all rows.

```
db_update('poll_vote')
->fields(array('uid' => 0))
->execute();
```

Relevant Hooks

hook_update(\$node)

It is called to allow the module to take action when an edited node is being updated in the database.

```
db_query("UPDATE {mytable} SET extra = '%s' WHERE nid = %d",
  $node->extra, $node->nid);
```

hook_query_alter(QueryAlterableInterface \$query)

Perform alterations to a structured query before it is executed.

```
$query->condition("#{access_alias}.grant_$op", 1, '>=');
```

hook_insert(\$node)

It is called to allow the module to take action when a new node is being inserted in the database.

```
db_query("INSERT INTO {mytable} (nid, extra)
VALUES (%d, '%s')", $node->nid, $node->extra);
```

hook_delete(&\$node)

It is called to allow the module to take action when a node is being deleted from the database.

```
db_query('DELETE FROM {mytable} WHERE nid = %d', $node->nid);
```

hook_load(\$nodes)

It is called to allow the module a chance to load extra information that it stores about a node. The hook should not be used to replace information from the core {node} table since this may interfere with the way nodes are fetched from cache.

```
$result = db_query('SELECT nid, foo FROM {mytable}
WHERE nid IN (:nids)', array(':nids' => array_keys($nodes)));
foreach ($result as $record) {
  $nodes[$record->nid]->foo = $record->foo;
}
```