

Plugin SQL

Use this [wiki plugin](#) to display the results of database queries. The plugin displays the results of the query in a formatted table. Or in other words: have you ever wished you could retrieve data that is actually in Tiki's (or some other) database, but not organized or displayed on-screen in the way you'd like? The SQL plugin allows you to do that. If you want to create a database, please see [Trackers](#).

Parameters

Query a MySQL database and display the results

Introduced in Tiki 1. Required parameters are in **bold**.

[Go to the source code](#)

Preferences required: `wikiplugin_sql`

| Parameters | Accepted Values | Description | Default | Since |
|------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------|---------|-------|
| (body of plugin) | | The SQL query goes in the body. Example: <code>SELECT column1, column2 FROM table</code> | | |
| db | | DSN name of the database being queried. The DSN name needs to first be defined at <code>tiki-admin_dsn.php</code> | | 1 |
| <code>delim</code> | | The delimiter to be used between data elements (sets <code>raw=1</code>) | | 11.0 |
| <code>audit</code> | | Create a log entry containing information about the SQL call. | 0 | 21.2 |
| <code>audit_csv</code> | text | If set, a CSV file will be created or appended with information about the SQL call performed. | | 21.2 |
| <code>raw</code> | (blank) 0 1 | Return with table formatting (default) or raw data with no table formatting | 0 | 11.0 |
| <code>wikiparse</code> | (blank) 0 1 | Turn wiki parsing of select results on and off (default is on) | 1 | 11.0 |

Troubleshooting

DSN

- Check your DSN exists (see [Admin DSN](#))
- Check your DSN has the right syntax, naming the right user, password, host and database.
- Check the user in your DSN can actually login into your database (for mysql you could test a login with phpmyadmin for example)

Permissions

- Ensure the Tiki user you are logged in as is a member of a group that has the permission to use the DSN

A Word About Security

Use of the SQL plugin can pose a risk to the security of your Tiki site. If you use the SQL plugin on a Wiki page and give users permission to access the page, *be sure to lock the page avoid users to see the source of that page*. If you don't, users with appropriate permission could see the SQL code — and might decide to do a little "experimentation" to see what else they can pull out of the database!

The SQL plugin can be used to pull in data from other databases, but this capability should be used only with caution. *If you want to use the SQL plugin to access confidential or sensitive data, do so ONLY in the context of a well-secured intranet that is not accessible to external users.*

To enforce security any user viewing a page that uses the PPluginSQL will need the permission `tiki_p_dsn_dsname` for the PluginSQL to execute where `dsname` is the name of the dsn. For instance a dsn `localhost` , will need a perm `tiki_p_dsn_localhost` . Further information can be found at [Admin DSN](#) and in the Troubleshooting section below.

How to use the plugin

To use the SQL plugin, you must do the following:

1. Use **Admin DSN** to define the database to be used by the SQL plugin. For instructions, please see [Admin DSN](#).
2. Place a SQL plugin statement in a Wiki page.
Start by trying a simple query — and experiment!

The following sections detail the plugin syntax, discuss some simple queries, and illustrate some more complex (and useful) queries you can try.

Remember, though, that this page tells you how to insert the SQL statement in a Wiki page, but it won't work until you've defined a DSN.

SQL

This table represents some helpful information about the structure of an SQL "SELECT" statement. Detailed information regarding MySQL specific statements can be found on the [MySQL documentation](#) site. Much of the MySQL syntax is transferrable to other databases, but database specific documentation should be referenced to take full advantage of database nuances.

| Param | Values | Defaults | Req ? | Explanation | Since |
|----------|---------------------------|----------|-------|--------------------------------------------|---------|
| "Select" | Just the word "select" | None | Y | SQL command to retrieve data from database | SQL 1.0 |

| Param | Values | Defaults | Req ? | Explanation | Since |
|-----------------------------------------------------|----------------------|----------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>column1</i> , <i>column2</i> , <i>etc.</i> | names of columns | None | Y | The names of the columns from which you want to retrieve data. These must be typed exactly as they are stored in your database program. Tip: To make sure you spell the column names correctly, take a look at your database's structure with a utility such as PHPMYAdmin. | SQL 1.0 |
| from | Just the word "from" | None | Y | Descriptor used to indicate the next parameter is the name of of an data entity, usually a table, but also maybe a view | SQL 1.0 |
| <i>table</i> | name of table | None | Y | The name of the table from which you want to retrieve the data. Again, the name must be typed exactly as it is stored in the database | |

For more information on SQL when used with MySQL, see

- [Select Syntax](#)
- <http://www.mysql.com/doc/en/SELECT.html>

Examples

Basic Example

This code:

```
{SQL(db="tiki")} SELECT authorName,title FROM tiki_articles {SQL}
```

Would produce:

| authorName | title |
|---------------------|-------------------------------------------------------------|
| admin | Database protection legislation is back... |
| admin | The "IP as Private Property" Papers: Volume I |
| Rocky Zandieh | Man charged pirating Oscar movies |
| miruna_tecuci | Recording Artists Coalition |
| aaron_still | SCO Letter |
| tyrus_berry | TEACH Act |
| christopher_clarke | A History of Hip Hop and Sampling |
| rocky_zandieh | Recent email worm attacks SCO |
| admin | More on hip-hop, copyright, and creativity... |
| admin | Songwriters and music publishers are thriving |
| andrew_leonard | Pepsi and the RIAA |
| charles_harrison | Negative Copyright Examples |
| McCullerses, Lanita | Music in the public domain |
| manesh_mathew | Netscape, Playboy settle search trademark case |
| brianne_herman | Copyright extended for 20 years |
| derek_yeung | MikeRoweSoft No More |
| derek_yeung | More payout on patent |
| derek_yeung | Italy: Our western pirates? |
| andrew_leonard | Google vs Booble? |
| stephen_guy | Safety vs. Private Property |
| brianne_herman | Piracy forces Blockbuster to shut down |
| admin | USPTO Gives California Attorney Patent on WWW Naming Scheme |
| matthew_spear | Google suit over paid ads |
| manesh_mathew | Netscape, Playboy settle search trademark case |
| Pirzadeh, Nima | NFL threatens Vegas Casinos |

Raw Parameter Example

The `raw` parameter requests that returned data be presented with no headers or table formatting at all. This is useful for creating in-line queries. As illustrated in the below example:

Wizard

SQL

Query a MySQL database and display the results

DSN Name
DSN name of the database being queried. The DSN name needs to first be defined at tiki-admin_dsn.php

Raw return
Return raw data with no headers

Delim
The delimiter to be used between data elements when "raw" is used

Wiki Parse Flag
Turn wiki parsing of select results on and off (0=off, 1=on)

The SQL query goes in the body. Example: `SELECT column1, column2 FROM table`
`select count(*) from tiki_pages`

Click to expand

This code:

```
{BOX(bg="beige" width="384px")}There are {SQL(db="tiki" raw="1")}select count(*) from tiki_pages{SQL} total pages on this site.{BOX}
```

Would produce:

There are 1465 total pages on this site.

Click to expand

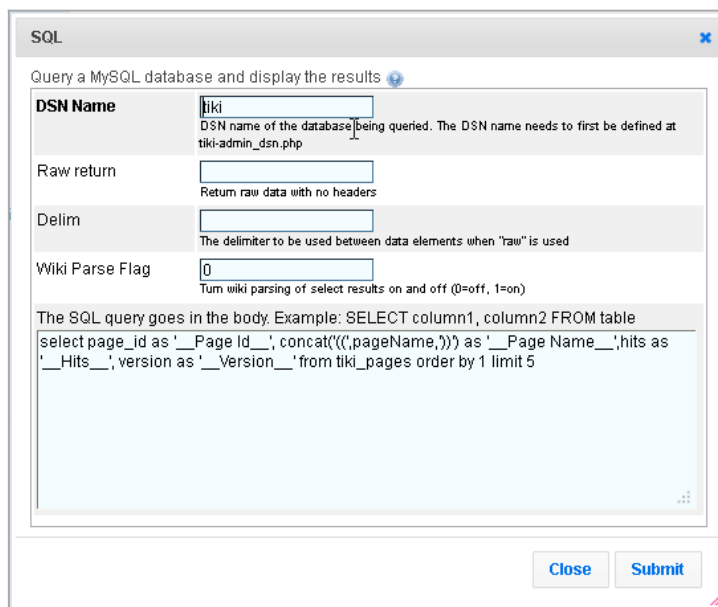
Wiki Parse Flag Example

The "Wiki Parse Flag" parameter enables the plugin implementor to determine if the returned data should be Wiki parsed. An example for how this can be handy is if someone had a list of servers they were maintaining and were to select server information out of the database with the hostname bracketed in parens to make the hostname automatically become a link. In this case a location available to anyone with access is automatically created for keeping special notes as related to each server.

In the below case Wiki pages are select from the Tiki site with the page reference being bracketed to become a live link. Column headers are also made bold.

Without parsing

Wizard



The screenshot shows a 'SQL Wizard' window with the following fields and values:

- DSN Name:** tiki (with a tooltip: "DSN name of the database being queried. The DSN name needs to first be defined at tiki-admin_dsn.php")
- Raw return:** (empty) (with a tooltip: "Return raw data with no headers")
- Delim:** (empty) (with a tooltip: "The delimiter to be used between data elements when 'raw' is used")
- Wiki Parse Flag:** 0 (with a tooltip: "Turn wiki parsing of select results on and off (0=off, 1=on)")

The SQL query entered in the body is: `select page_id as '__Page Id__', concat('('||pageName,')') as '__Page Name__', hits as '__Hits__', version as '__Version__' from tiki_pages order by 1 limit 5`

Buttons at the bottom: Close, Submit

Click to expand

This code:

```
{SQL(db="tiki" wikiparse="0")}select page_id as '__Page Id__', concat('('||pageName,')') as '__Page Name__', hits as '__Hits__', version as '__Version__' from tiki_pages order by 1 limit 5{SQL}
```

Would produce:

| __Page Id__ | __Page Name__ | __Hits__ | __Version__ |
|-------------|-----------------------------|----------|-------------|
| 1 | ((HomePage)) | 282 | 4 |
| 3 | ((ECP Main)) | 4180 | 36 |
| 4 | ((ToolsAndProcesses)) | 302 | 10 |
| 5 | ((Performance Engineering)) | 14 | 1 |
| 6 | ((Fabricator)) | 463 | 4 |

Click to expand

With Parsing

Wizard

SQL

Query a MySQL database and display the results

DSN Name
DSN name of the database being queried. The DSN name needs to first be defined at tiki-admin_dsn.php

Raw return
Return raw data with no headers

Delim
The delimiter to be used between data elements when "raw" is used

Wiki Parse Flag
Turn wiki parsing of select results on and off (0=off, 1=on)

The SQL query goes in the body. Example: SELECT column1, column2 FROM table

```
select page_id as '__Page Id__', concat('(' ,pageName,')') as '__Page Name__',hits as '__Hits__', version as '__Version__' from tiki_pages order by 1 limit 5
```

Click to expand

This code:

```
{SQL(db="tiki" wikiparse="1")}select page_id as '__Page Id__', concat('(' ,pageName,')') as '__Page Name__',hits as '__Hits__', version as '__Version__' from tiki_pages order by 1 limit 5{SQL}
```

Would produce:

| Page Id | Page Name | Hits | Version |
|---------|-----------------------------------------|------|---------|
| 1 | HomePage | 282 | 4 |
| 3 | ECP Main | 4180 | 36 |
| 4 | ToolsAndProcesses | 302 | 10 |
| 5 | Performance Engineering | 14 | 1 |
| 6 | Fabricator | 463 | 4 |

Click to expand

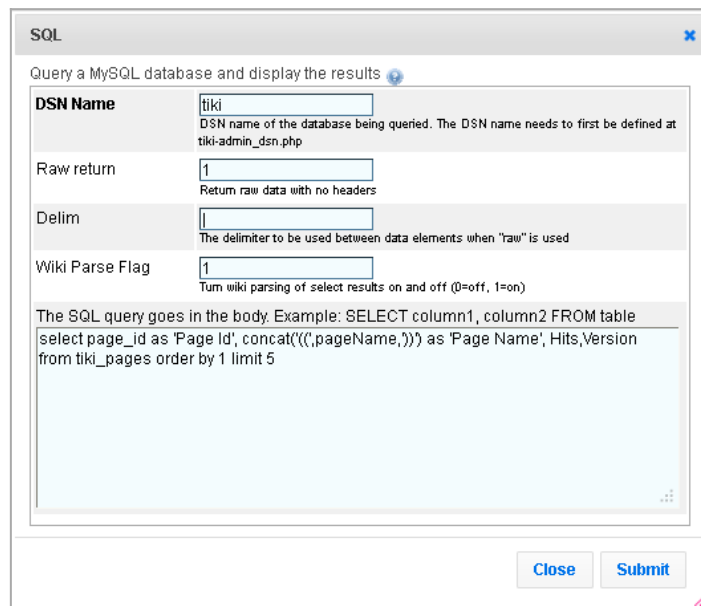
Delim Parameter Examples

The "Delim" parameter allows for the delimiter to be defined when "Raw" is used. Setting "Raw" is optional and assumed when a delimiter is defined.

Straight Wiki Table

The `delim` parameter can be used to create a straight wiki table instead of relying on the default table created by the SQL plugin itself. This is useful for maintaining a consistent look if regular wiki tables are also used in other areas of the same page. This example combines two queries to create one table.

Wizard:



SQL

Query a MySQL database and display the results

DSN Name: tiki
DSN name of the database being queried. The DSN name needs to first be defined at tiki-admin_dsn.php

Raw return: 1
Return raw data with no headers

Delim: |
The delimiter to be used between data elements when "raw" is used

Wiki Parse Flag: 1
Turn wiki parsing of select results on and off (0=off, 1=on)

The SQL query goes in the body. Example: SELECT column1, column2 FROM table
select page_id as 'Page Id', concat('(',pageName,')') as 'Page Name', Hits,Version from tiki_pages order by 1 limit 5

Close Submit

Click to expand

This code:

```
||__PID__|__Page Name__|__Hits__|__Version__ {SQL(db="tiki" raw="1" delim="|"
wikiparse="1")}}select page_id as 'Page Id', concat('(',pageName,')') as 'Page Name',
Hits,Version from tiki_pages order by 1 limit 5{SQL}||
```

Would produce:

| PID | Page Name | Hits | Version |
|-----|-----------------------------------------|------|---------|
| 1 | HomePage | 282 | 4 |
| 3 | ECP Main | 4180 | 36 |
| 4 | ToolsAndProcesses | 302 | 10 |
| 5 | Performance Engineering | 14 | 1 |
| 6 | Fabricator | 463 | 4 |

Click to expand

Deluxe Wiki Table

The below modification to the Wiki Code creates a Wiki Table with a summary statement at the

bottom.

This code:

```
||__PID__|__Page Name__|__Hits__|__Version__ {SQL(db="tiki" raw="1" delim="|"
wikiparse="1")}select page_id as 'Page Id', concat('(',pageName,')') as 'Page Name',
Hits,Version from tiki_pages order by 1 limit 5{SQL}::_FIRST 5 of {SQL(db="tiki"
raw="1")}select count(*) from tiki_pages{SQL} SITE PAGES_::||
```

Would produce:

| PID | Page Name | Hits | Version |
|-----------------------------------|-----------------------------------------|------|---------|
| 1 | HomePage | 282 | 4 |
| 3 | ECP Main | 4180 | 36 |
| 4 | ToolsAndProcesses | 302 | 10 |
| 5 | Performance Engineering | 14 | 1 |
| 6 | Fabricator | 463 | 4 |
| FIRST 5 of 1465 SITE PAGES | | | |

Click to expand

CSV C&P

The `delim` parameter is useful for providing a CSV format for use in a poor man's export through cut and paste.

This code:

```
{SQL(db="tiki" wikiparse="1")}select page_id as 'Page Id', concat('(',pageName,')') as 'Page
Name', Hits, Version from tiki_pages order by 1 limit 5{SQL} __C&P for spreadsheet import
use__ ^{SQL(db="tiki" raw="0" delim="," wikiparse="0")}select page_id ,pageName, Hits,
Version from tiki_pages order by 1 limit 5{SQL}^
```

Would produce:

| Page Id | Page Name | Hits | Version |
|-------------------------------------------|-----------------------------------------|------|---------|
| 1 | HomePage | 282 | 4 |
| 3 | ECP Main | 4180 | 36 |
| 4 | ToolsAndProcesses | 302 | 10 |
| 5 | Performance Engineering | 14 | 1 |
| 6 | Fabricator | 463 | 4 |
| C&P for spreadsheet import use | | | |
| page_id,pageName,Hits,Version | | | |
| 1,HomePage,282,4 | | | |
| 3,ECP Main,4180,36 | | | |
| 4,ToolsAndProcesses,302,10 | | | |
| 5,Performance Engineering,14,1 | | | |
| 6,Fabricator,463,4 | | | |

Click to expand

Sorting and Selecting Data with the SQL Plugin

You can use `ORDER BY` to sort the output by a specified column name:

You do not have the permission that is needed to use this feature

Secondary sort:

```
{SQL(db="tiki")} SELECT authorName,title FROM tiki_articles ORDER BY authorName asc, title asc{SQL}
```

Use `WHERE` to limit the output to rows that meet specified criteria:

```
{SQL(db="tiki")} SELECT authorName,title FROM tiki_articles WHERE authorName='rocky'{SQL}
```

In the above example, note the use of single quotes — double quotes won't work.

Whoopee - a join:

```
{SQL(db="stshome")} SELECT t1.name, t2.value FROM tiki_tracker_fields AS t1, tiki_tracker_item_fields AS t2 WHERE t2.itemId = '5' AND t1.fieldId=t2.fieldId{SQL}
```

An example to retrieve in a single table all the distinct users assigned to every group. This is useful, for instance, in the education area where a teacher appreciates having a list of all the students successfully registered and assigned to his/her class/subject.

```
{SQL(db="db_uniwiki2")} SELECT DISTINCT `users_usergroups`.`groupName`,`users_users`.`login`,`users_users`.`email`,`users_users`.`default_group` FROM `users_usergroups`,`users_users` WHERE `users_users`.`userId` = `users_usergroups`.`userId` ORDER BY `users_usergroups`.`groupName` DESC,`users_users`.`login` ASC{SQL}
```

Note in the example above, that other columns are included in the table, such as their e-mail, default group, ... Note, also, that they are sorted first by group name in descending order (`DESC`), and then, by login (user name in Tiki) in ascendent order (`ASC`).

Global variables can also be used as the username to the query:

```
{SQL(db="local" 0="$user" 1="5")}SELECT * from users_users WHERE login=? OR userid=?{SQL}
```

where 0 is for the first `?` and 1 for the second

Related pages

- See [Admin DSN](#) to learn how to name and define an external DSN connection, which the SQL plugin requires.
- <http://tikiwiki.org/WikiPluginsDb>
- <http://tikiwiki.org/WikiPluginsDbTutorial>
- [PluginDBreport](#)