官网：<http://www.mycli.net/docs>

# Docs

## [AutoCompletion](http://www.mycli.net/completion) [local](#completion)

Auto-completion kicks in as soon as you start typing. The suggestions are context sensitive（建议是环境敏感的） so the REPL（交互式解释环境） will only suggest relevant items.

[Read More...](http://www.mycli.net/completion)

## [Config](http://www.mycli.net/config) local

The config file for mycli is located in the home folder (~/.myclirc).

[Read More...](http://www.mycli.net/config)

## [Colors](http://www.mycli.net/syntax)

Syntax highlighting has plenty of themes that can be changed via the [config](http://www.mycli.net/config) file.

Showcase of various color themes:

[Gallery](http://www.mycli.net/syntax)

## [Keybindings](http://www.mycli.net/keybindings)

There are two types of keybindings available. Emacs mode and Vi mode. The keybindings can be changed via the config file (~/.myclirc) or toggled via F4 key.

[Read more...](http://www.mycli.net/keybindings)

## [Favorite Queries](http://www.mycli.net/favorites)

Frequently used queries can be saved as favorite queries with a name.

[Read More...](http://www.mycli.net/favorites)

## [Logging](http://www.mycli.net/logging)

Default log file is located at ~/.mycli.log. There is additional user friendly audit logging.

[Read More..](http://www.mycli.net/logging)

## [Open Editor](http://www.mycli.net/editor)

Use the \e command in the prompt, to launch the default editor.

[Read More...](http://www.mycli.net/editor)

## [Multi-line mode](http://www.mycli.net/multi-line)

Mycli doesn't expect a semi-colon at the end of a line to execute the query. But this can be changed by setting the multi-line mode.

[Read more...](http://www.mycli.net/multi-line)

## [History & Search](http://www.mycli.net/history)

MyCli keeps track of the queries entered in the repl. Up/Down arrow can be used to navigate the history.

Pressing <C-r> will enable incremental history search. So press <C-r> and then start typing your search term to see the queries narrowed down. You can cycle through the matches by pressing <C-r> again.

[Read more...](http://www.mycli.net/history)

## [Working with multiple servers](http://www.mycli.net/loginpath)

When working with multiple servers, it is possible to store their connection parameters using mysql\_config\_editor, and connect providing --login-path <name> <database> with a server alias.

[Read more...](http://www.mycli.net/loginpath)

## [Pager Output](http://www.mycli.net/pager)

Mycli uses pager programs to make it easier to view large result sets. This can be configured or disabled.

Auto-Completion

AutoCompletion

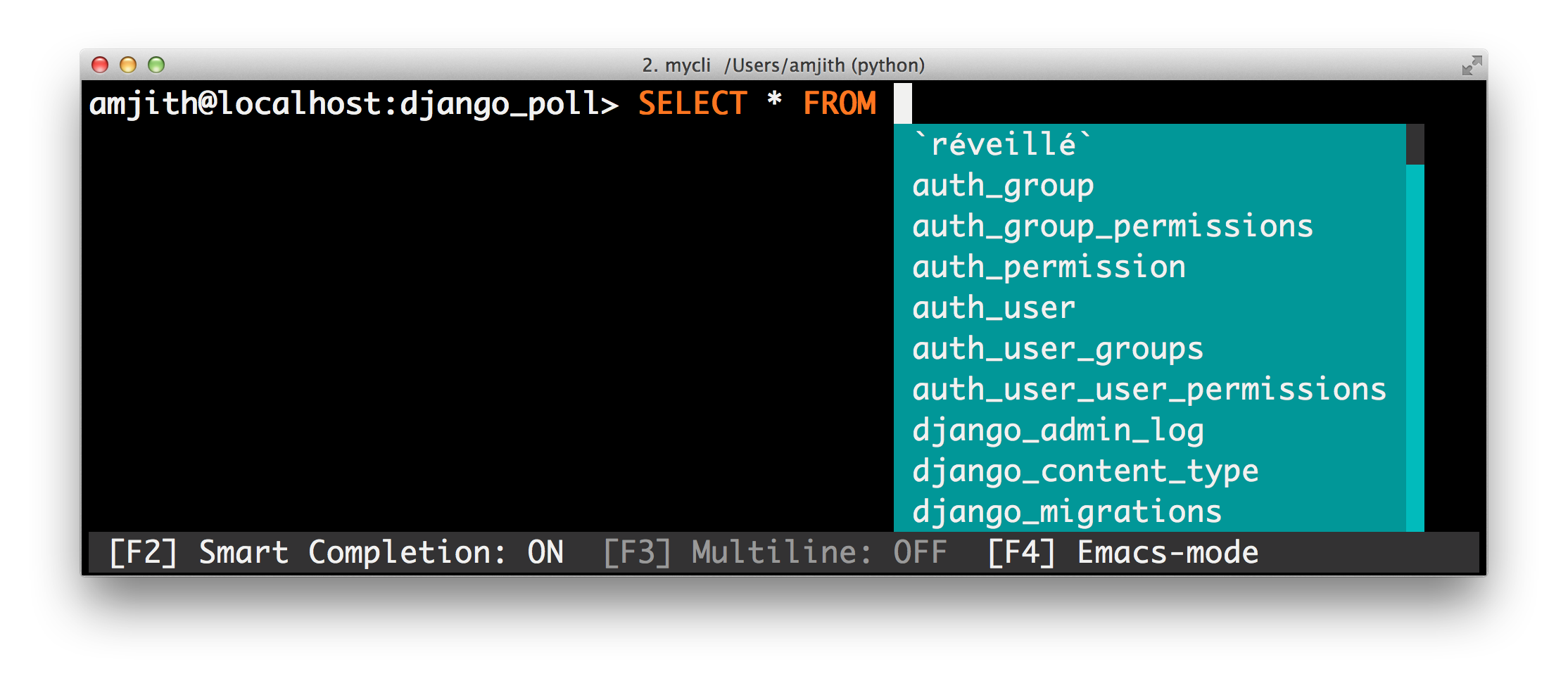
Auto-completion is on by default（默认启动）. The REPL will pop up a suggestion menu as soon as you start typing. The suggestions are context sensitive based on the position of the cursor（命令建议是根据光标的位置敏感的，即在特定的位置产生不同的建议）. eg: Only tables are suggested after the FROM keyword, only column names are suggested after the WHERE clause.

Smart Completion

I call this context sensitive suggestion as smart completion. Here are a few examples of smart completion.

Table

Only table names from the current database are suggested after the FROM keyword.



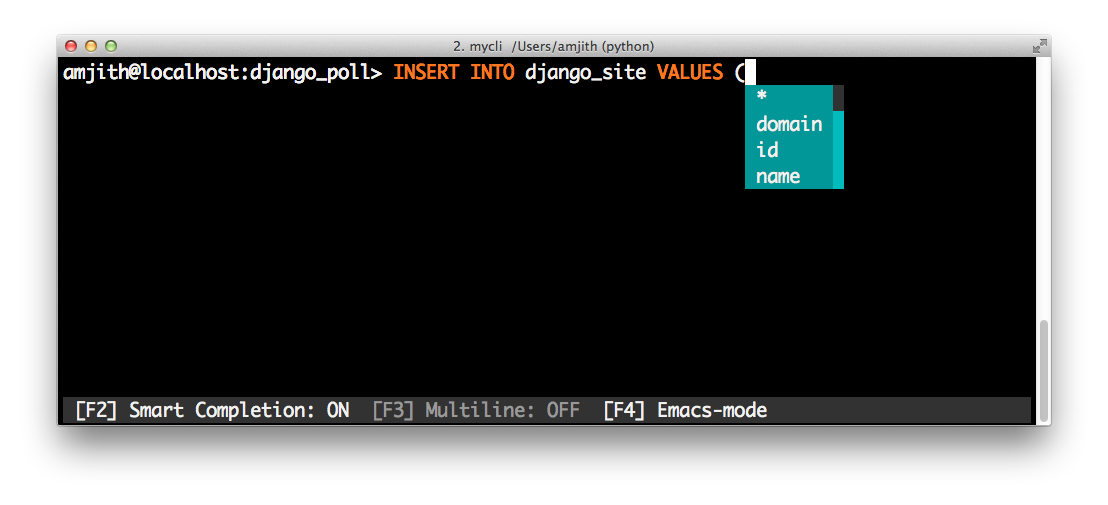
Column

Column names from the current table are suggested after the WHERE clause.



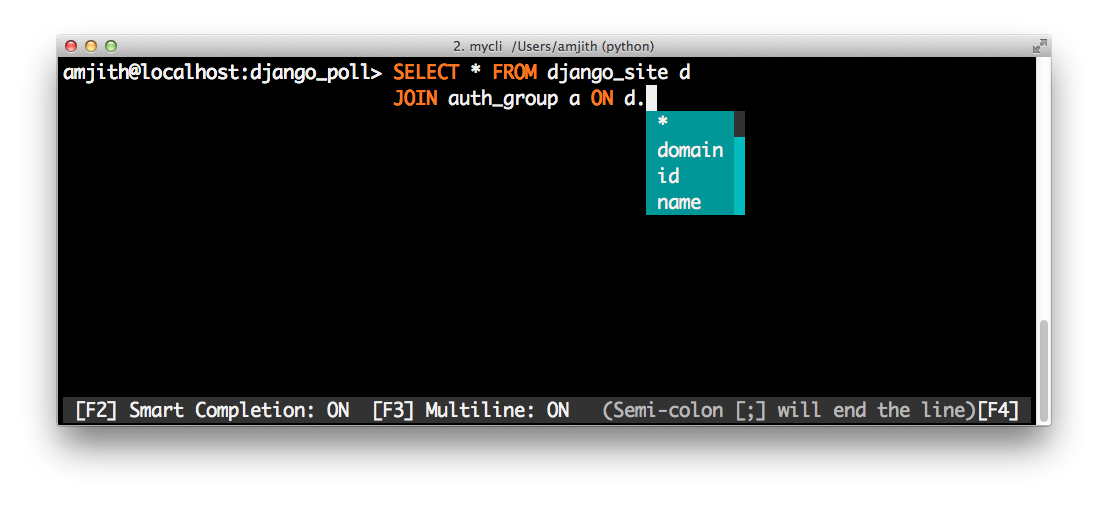
Insert

Insert statement will suggest the column names.



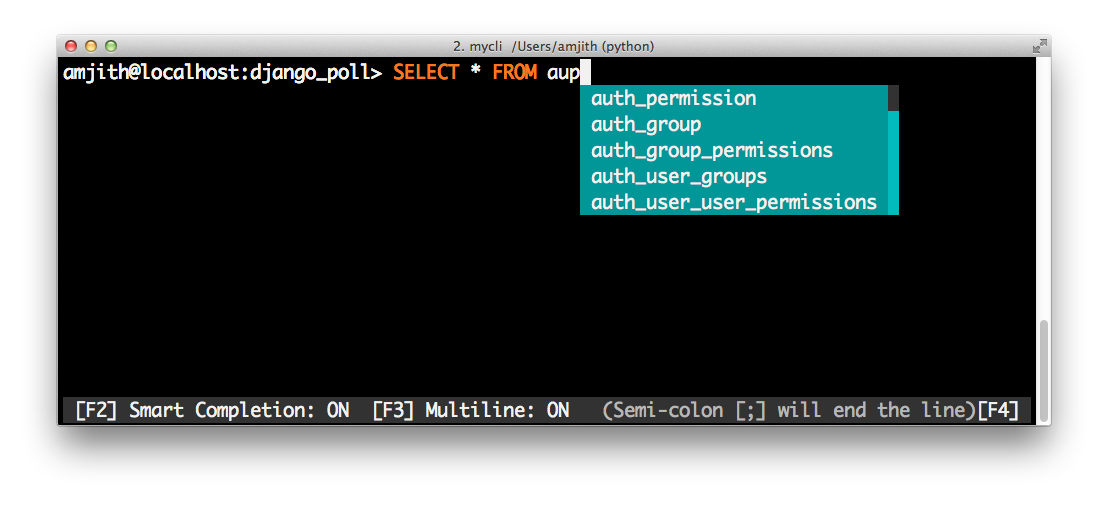
Alias

Aliases in the query are resolved and the columns from the table aliases are suggested.



Fuzzy Match

The completions are matched using a [fuzzy algorithm](http://blog.amjith.com/fuzzyfinder-in-10-lines-of-python)（模糊算法）. For example typing 'djmi' will match the table 'django\_migrations' because 'djmi' has parts of matching substrings（以包含字符的形式匹配子字符串）. Here's an example:



Configuration

Introduction

Most of mycli's user settings are configured via the file located at ~/.myclirc, which is a hidden file in your home folder（配置文件在家目录） in Linux and OS X. On Windows it is located at C:\Users\<username>\.myclirc.

The config file is created when mycli is launched for the very first time. Updates to that file are not overwritten by subsequent launches of mycli or updating the version of mycli（更新版本并不会更新这个配置文件）.

Mycli also reads the [client] section of MySQL's option file, ~/.my.cnf (on Windows: C:\Users\<username>\.my.cnf). See the example file at the bottom of this page for more information.

**NOTE:** Mycli does not read the [mysql] section of MySQL's option files. It only reads the [client] section.

MySQL recommends that users store their passwords in an encrypted login path file. Mycli will read this file（建议将加密后的密码放在登录文件中便于mycli启动时读取这个配置文件）. For more information about how to store your authentication credentials in this file, see the [MySQL Configuration Utility documentation](https://dev.mysql.com/doc/refman/5.7/en/mysql-config-editor.html).

Sample Mycli Config File

**[main]**

*# Enables context sensitive auto-completion. If this is disabled then all*

*# possible completions will be listed.*

smart\_completion = True

*# Multi-line mode allows breaking up the sql statements into multiple lines. If*

*# this is set to True, then the end of the statements must have a semi-colon.*

*# If this is set to False then sql statements can't be split into multiple*

*# lines. End of line (return) is considered as the end of the statement.*

multi\_line = False

*# Destructive warning mode will alert you before executing a sql statement*

*# that may cause harm to the database such as "drop table", "drop database"*

*# or "shutdown".*

destructive\_warning = True

*# log\_file location.*

log\_file = ~/.mycli.log

*# Default log level. Possible values: "CRITICAL", "ERROR", "WARNING", "INFO"*

*# and "DEBUG".*

log\_level = INFO

*# Log every query and its results to a file. Enable this by uncommenting the*

*# line below.*

*# audit\_log = ~/.mycli-audit.log*

*# Timing of sql statments and table rendering.*

timing = True

*# Table format. Possible values: ascii, double, github,*

*# psql, plain, simple, grid, fancy\_grid, pipe, orgtbl, rst, mediawiki, html,*

*# latex, latex\_booktabs, textile, moinmoin, jira, vertical, tsv, csv.*

*# Recommended: ascii*

table\_format = ascii

*# Syntax Style. Possible values: manni, igor, xcode, vim, autumn, vs, rrt,*

*# native, perldoc, borland, tango, emacs, friendly, monokai, paraiso-dark,*

*# colorful, murphy, bw, pastie, paraiso-light, trac, default, fruity*

syntax\_style = default

*# Keybindings: Possible values: emacs, vi.*

*# Emacs mode: Ctrl-A is home, Ctrl-E is end. All emacs keybindings are available in the REPL.*

*# When Vi mode is enabled you can use modal editing features offered by Vi in the REPL.*

key\_bindings = emacs

*# Enabling this option will show the suggestions in a wider menu. Thus more items are suggested.*

wider\_completion\_menu = False

*# MySQL prompt*

*# \t - Product type (Percona, MySQL, Mariadb)*

*# \u - Username*

*# \h - Hostname of the server*

*# \d - Database name*

*# \n - Newline*

prompt = '\t \u@\h:\d> '

*# Skip intro info on startup and outro info on exit*

less\_chatty = False

*# Use alias from --login-path instead of host name in prompt*

login\_path\_as\_host = False

*# Cause result sets to be displayed vertically if they are too wide for the current window,*

*# and using normal tabular format otherwise. (This applies to statements terminated by ; or \G.)*

auto\_vertical\_output = False

*# Custom colors for the completion menu, toolbar, etc.*

**[colors]**

*# Completion menus.*

Token.Menu.Completions.Completion.Current = 'bg:#00aaaa #000000'

Token.Menu.Completions.Completion = 'bg:#008888 #ffffff'

Token.Menu.Completions.MultiColumnMeta = 'bg:#aaffff #000000'

Token.Menu.Completions.ProgressButton = 'bg:#003333'

Token.Menu.Completions.ProgressBar = 'bg:#00aaaa'

*# Selected text.*

Token.SelectedText = '#ffffff bg:#6666aa'

*# Search matches. (reverse-i-search)*

Token.SearchMatch = '#ffffff bg:#4444aa'

Token.SearchMatch.Current = '#ffffff bg:#44aa44'

*# The bottom toolbar.*

Token.Toolbar = 'bg:#222222 #aaaaaa'

Token.Toolbar.Off = 'bg:#222222 #888888'

Token.Toolbar.On = 'bg:#222222 #ffffff'

*# Search/arg/system toolbars.*

Token.Toolbar.Search = 'noinherit bold'

Token.Toolbar.Search.Text = 'nobold'

Token.Toolbar.System = 'noinherit bold'

Token.Toolbar.Arg = 'noinherit bold'

Token.Toolbar.Arg.Text = 'nobold'

*# Favorite queries.*

**[favorite\_queries]**

Sample MySQL Option File

**[client]**

*# The client section is read by mycli and all MySQL applications.*

*# Default connection information*

user = thor

password = hammer

host = localhost

database = asgard

port = 3306

*# Connect via a socket.*

*# socket = '/tmp/mysql.sock'*

*# Use the UTF-8 character set*

default-character-set = utf-8

*# SSL options - see the MySQL documentation for more information.*

*# https://dev.mysql.com/doc/refman/5.7/en/secure-connection-options.html#option\_general\_ssl-ca*

*# ssl-ca*

*# ssl-cert*

*# ssl-key*

*# ssl-cipher*

*# ssl-verify-server-cert*

*# Turn on the LOAD DATA INFILE statement*

local-infile = on

*# Another local infile alias.*

*# Use it if the previous one clashes with other MySQL tools.*

loose-local-infile = on

*# Configure the pager*

pager = 'vim -'

*# Disable the pager*

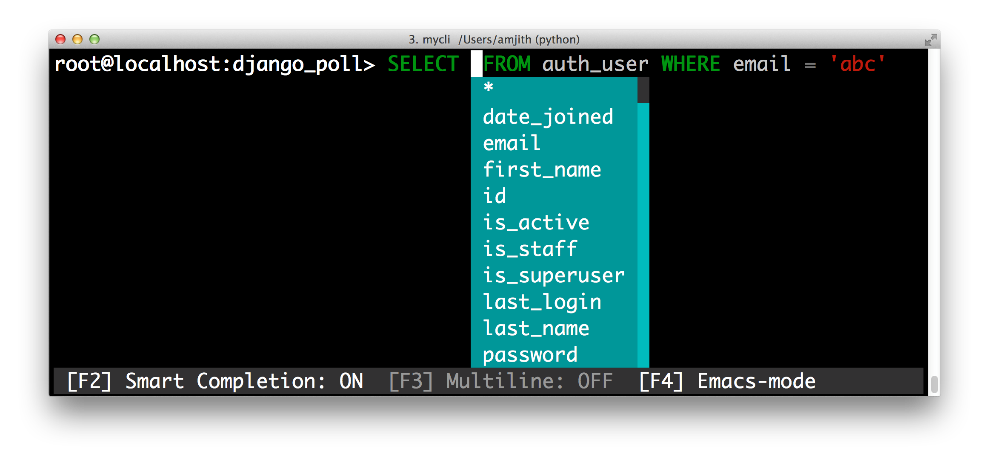
*# skip-pager = on*

Syntax Colors

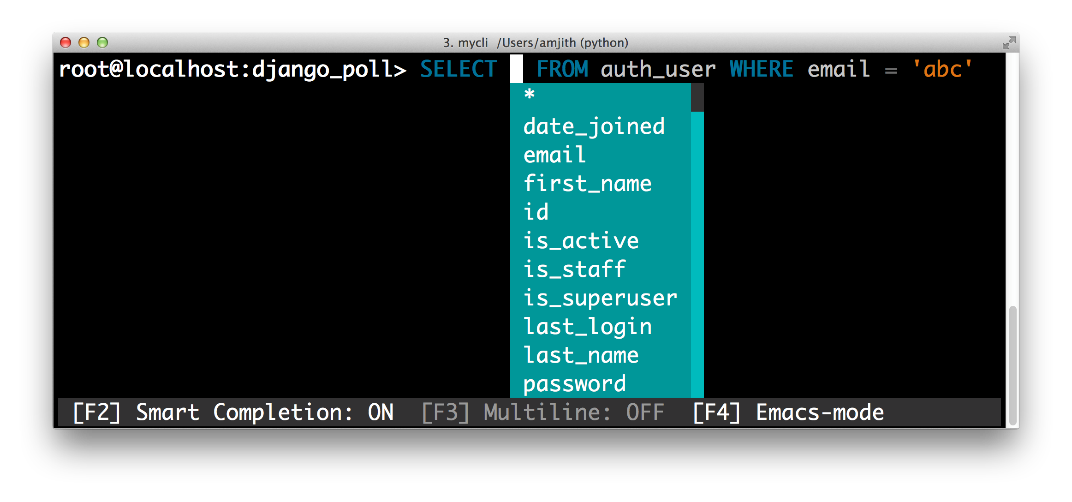
Here is a gallery of syntax color themes. These themes can be modified via the [config](http://www.mycli.net/config) file (~/.myclirc).

下面是颜色的主题

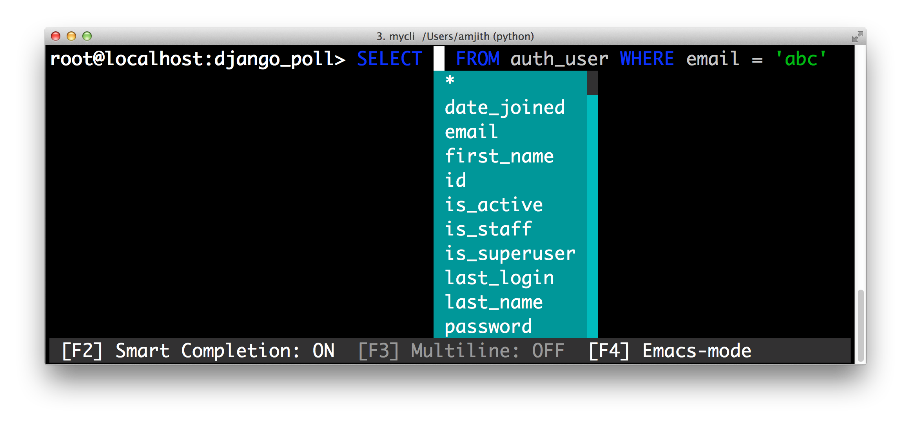
default



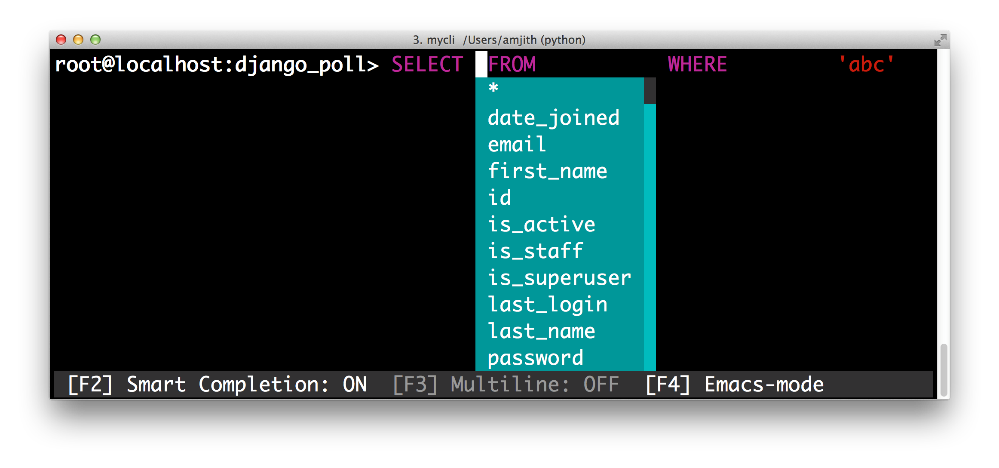
manni

Click the image to toggle dark/light background. 

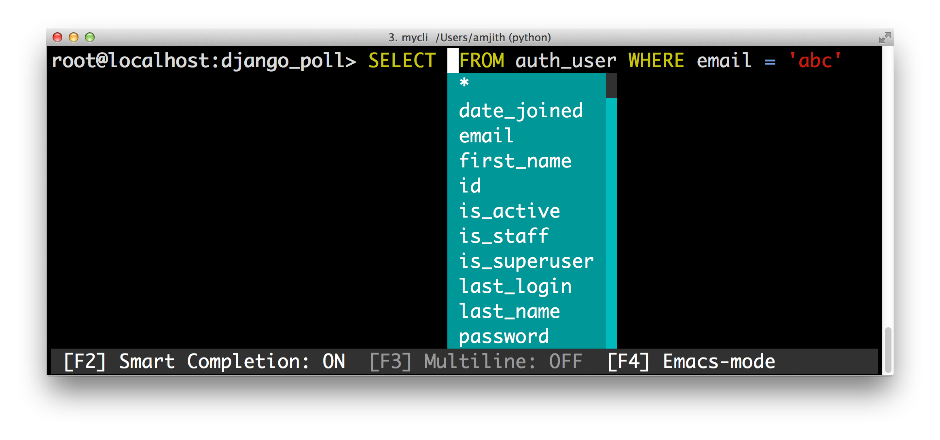
igor

Click the image to toggle dark/light background. 

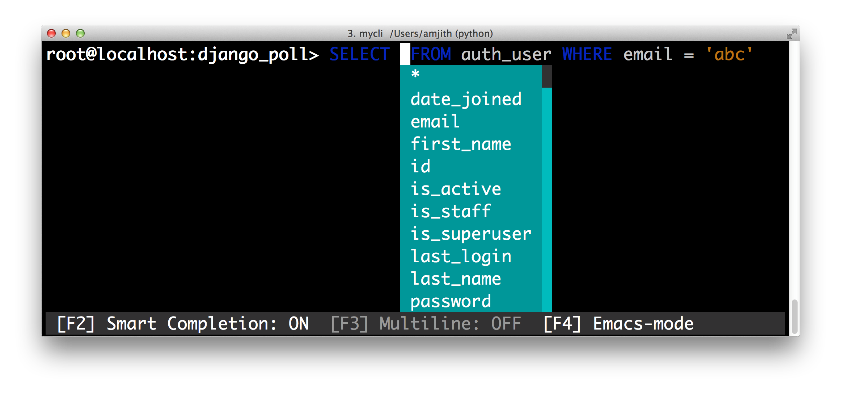
xcode

Click the image to toggle dark/light background. 

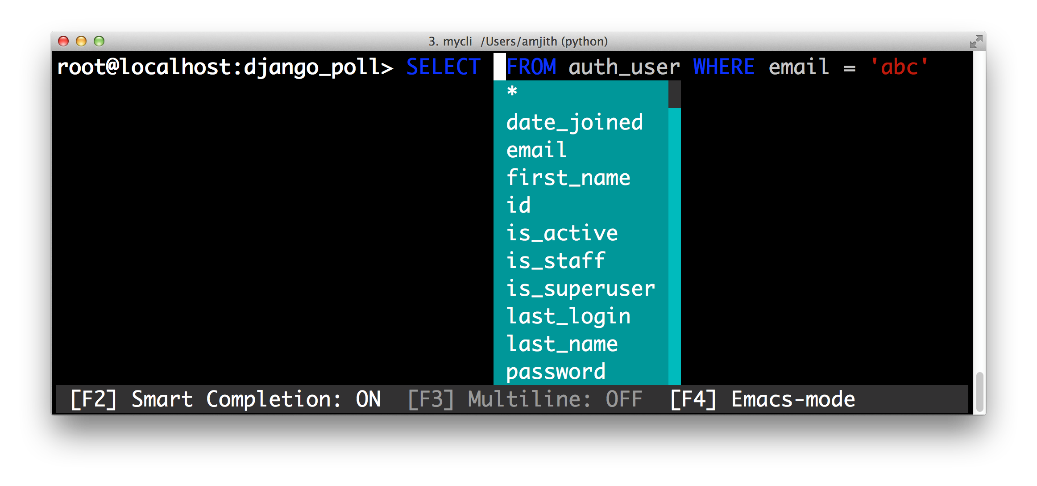
vim

Click the image to toggle dark/light background. 

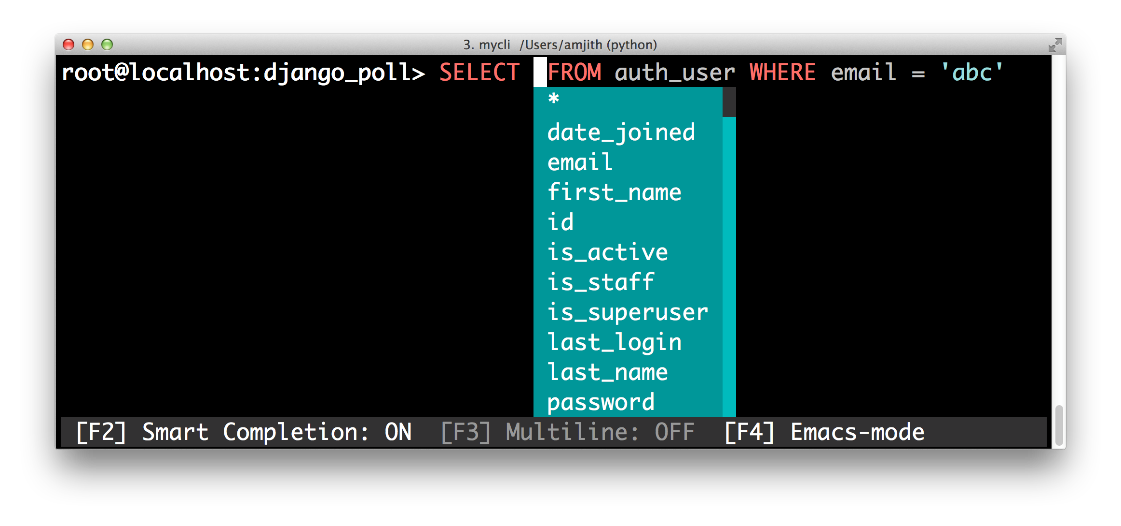
autumn

Click the image to toggle dark/light background. 

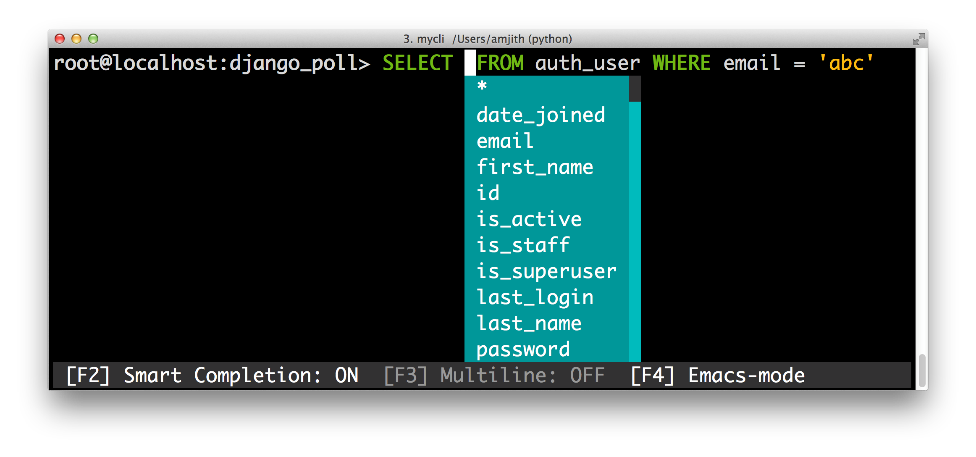
vs

Click the image to toggle dark/light background. 

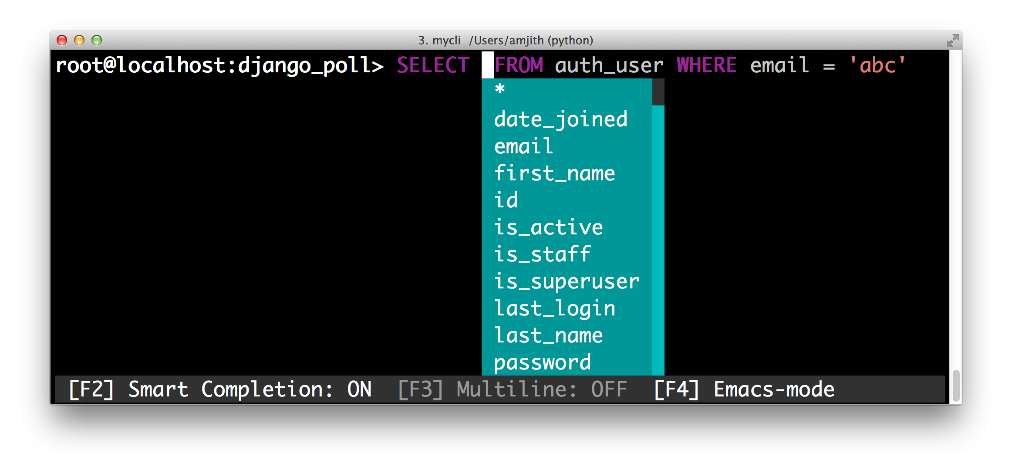
rrt

Click the image to toggle dark/light background. 

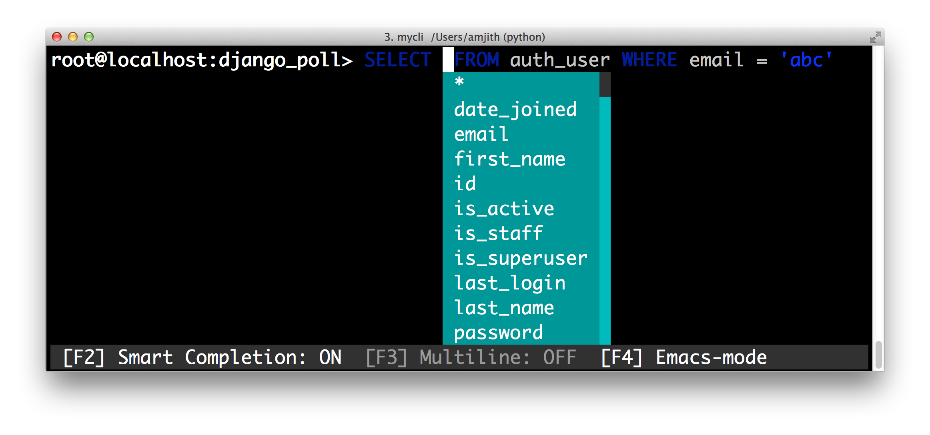
native

Click the image to toggle dark/light background. 

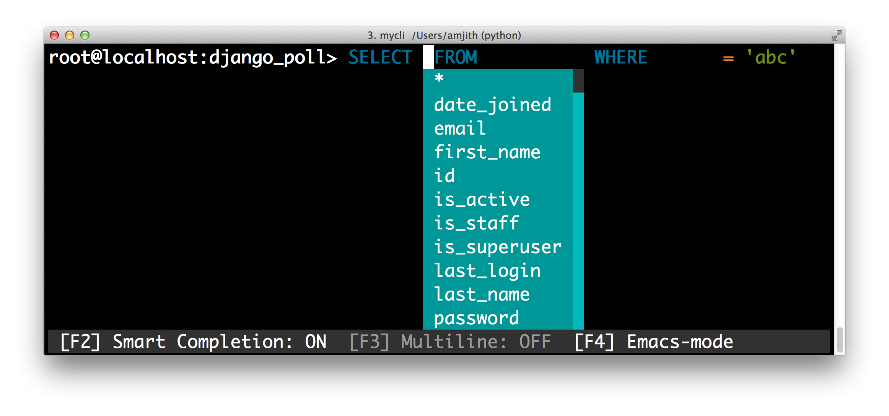
perldoc

Click the image to toggle dark/light background. 

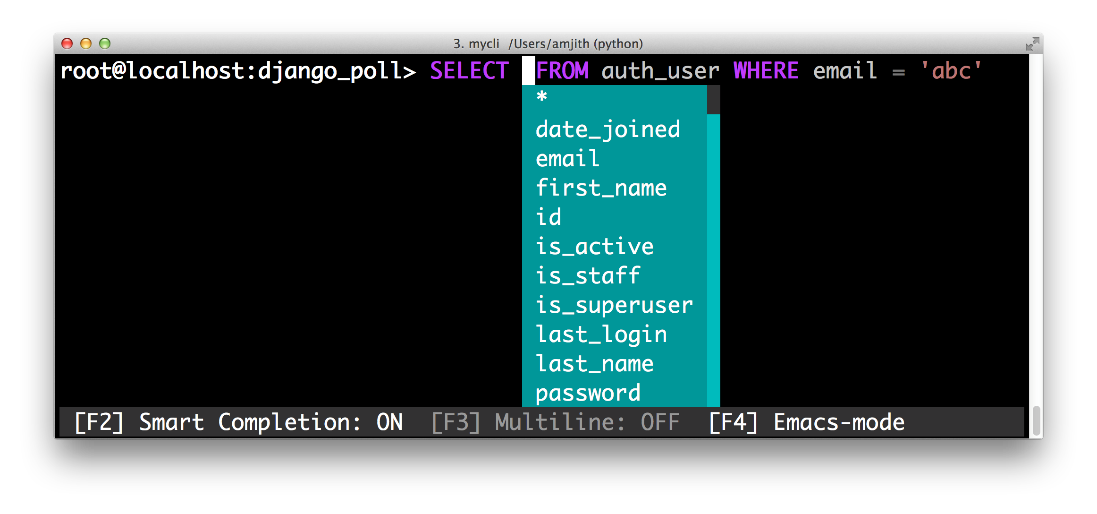
borland

Click the image to toggle dark/light background. 

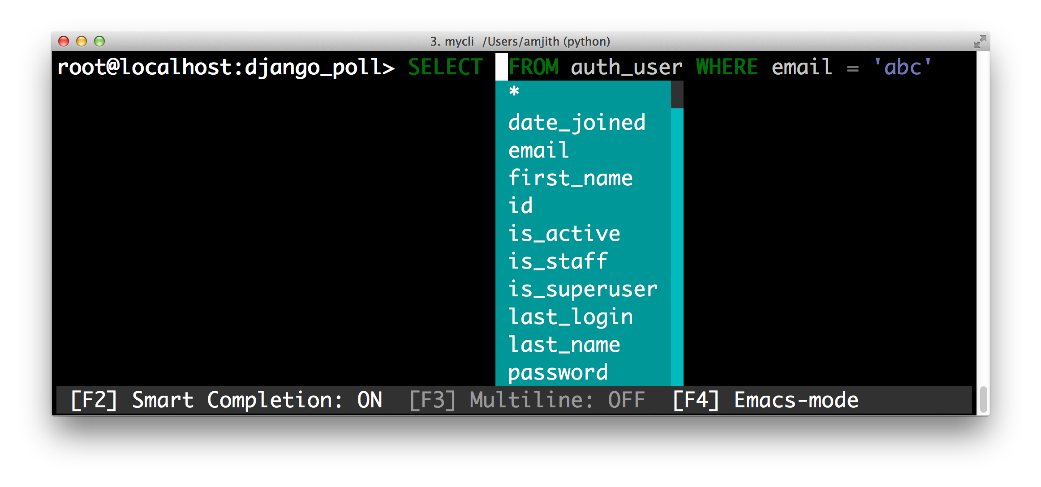
tango

Click the image to toggle dark/light background. 

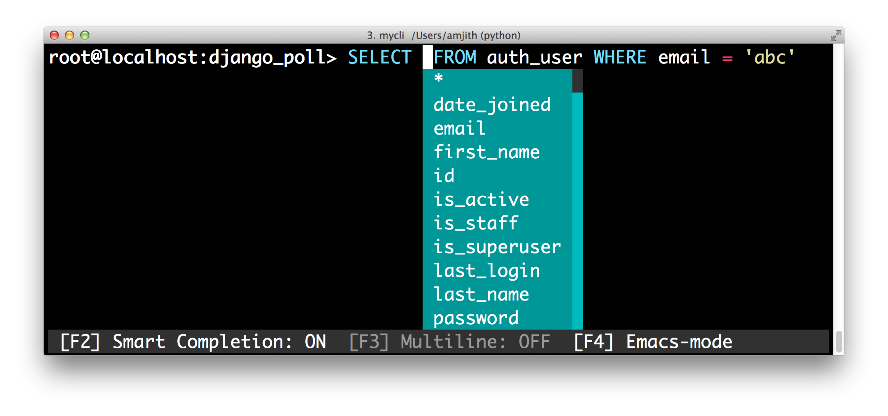
emacs

Click the image to toggle dark/light background. 

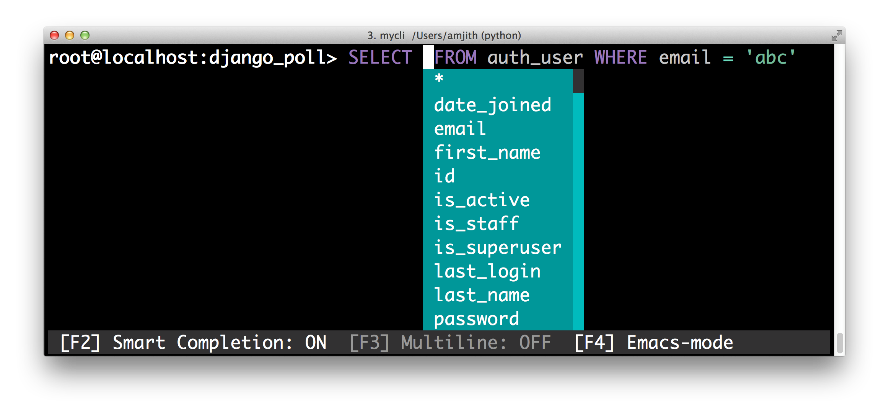
friendly

Click the image to toggle dark/light background. 

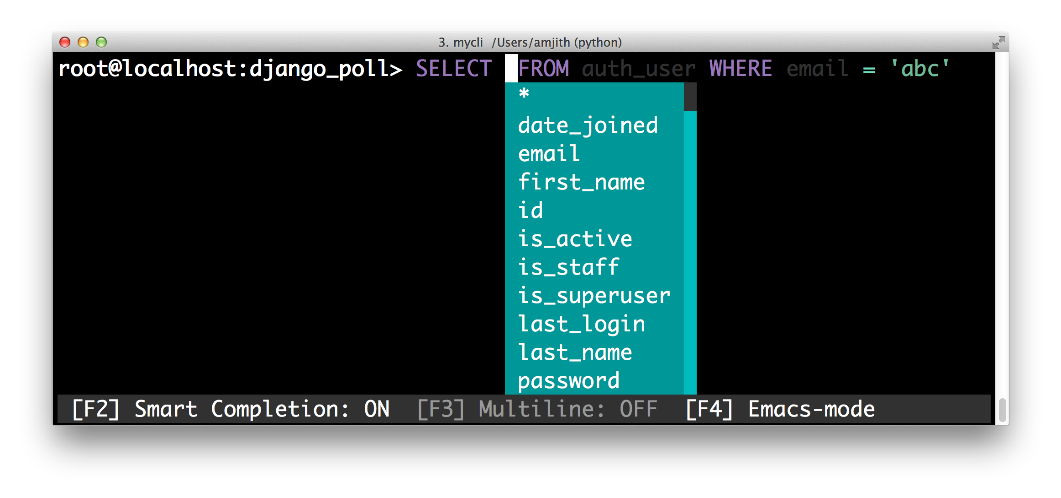
monokai

Click the image to toggle dark/light background. 

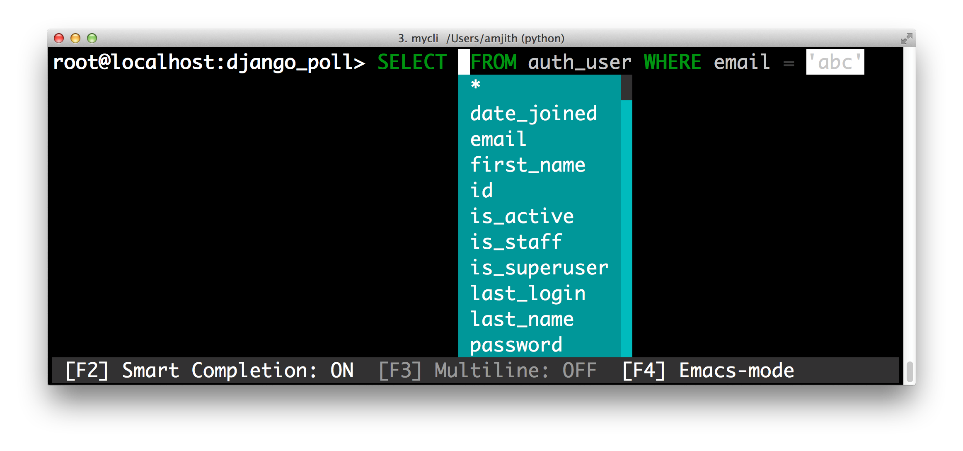
paraiso-dark

Click the image to toggle dark/light background. 

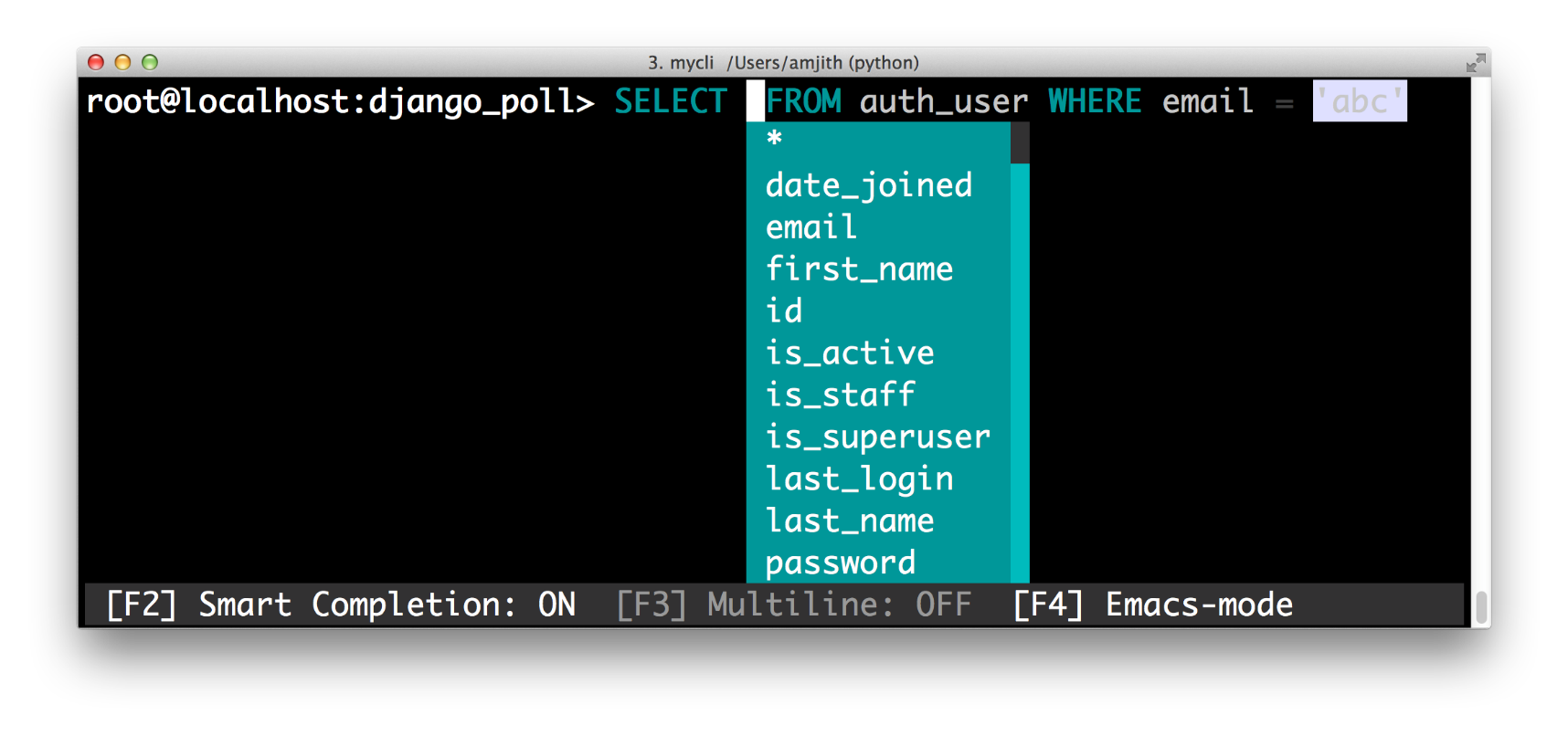
paraiso-light

Click the image to toggle dark/light background. 

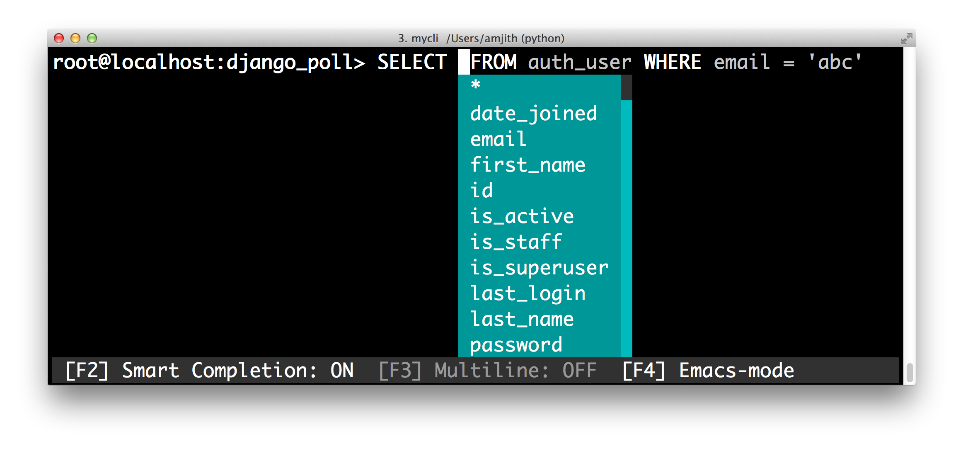
colorful

Click the image to toggle dark/light background. 

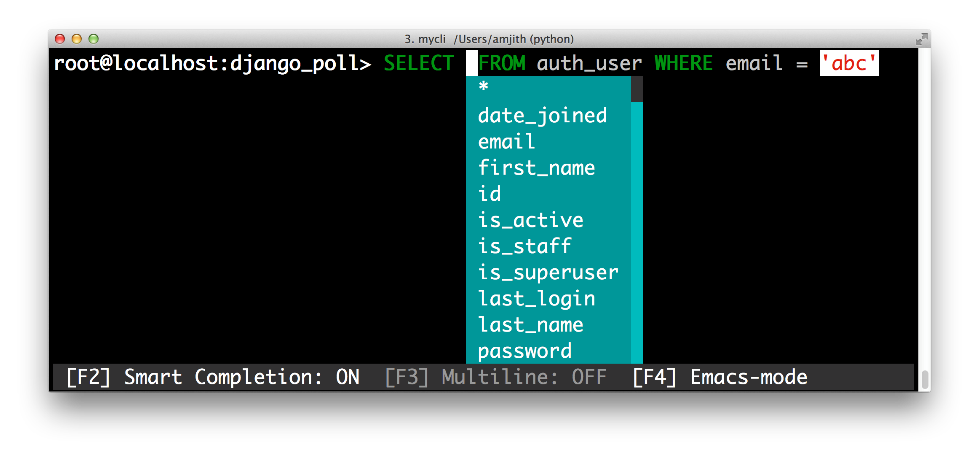
murphy

Click the image to toggle dark/light background. 

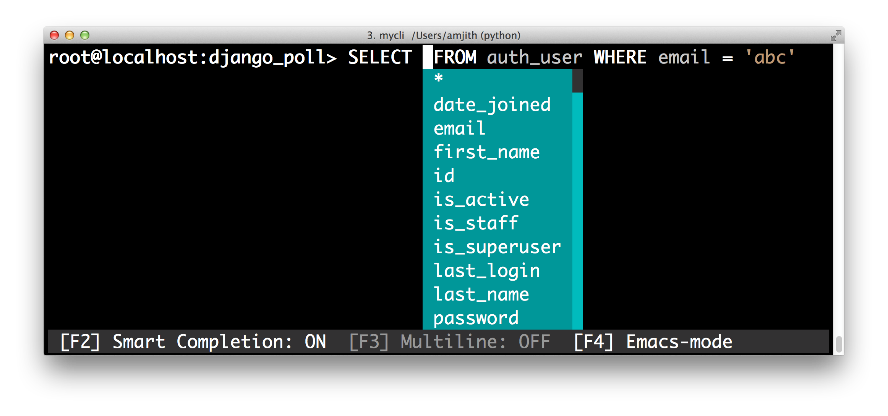
bw

Click the image to toggle dark/light background. 

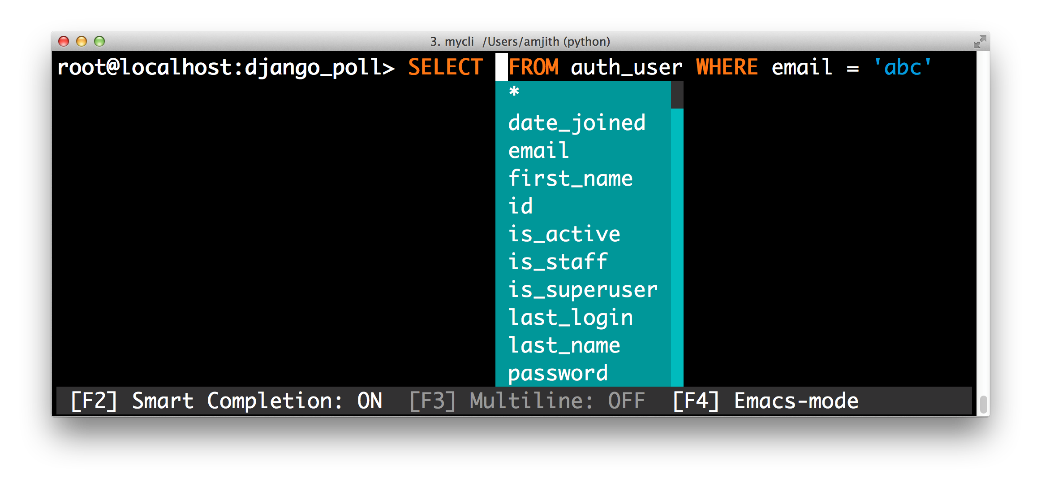
pastie

Click the image to toggle dark/light background. 

trac

Click the image to toggle dark/light background. 

fruity

Click the image to toggle dark/light background. 

# Keybindings

There are two types of keybindings available. Emacs mode and Vi mode. The keybindings can be changed via the config file (~/.myclirc) or toggled via F4 key（可以使用F4切换快捷键）.

Emacs mode is the default. In emacs mode you can type Ctrl-a to go to the beginning of a line and Ctrl-e to go to the end of a line and many more.

When Vi mode is enabled you can use modal editing features offered by Vi in the REPL command line.

# Favorite Queries

Favorite Queries are a way to save frequently used queries with a short name(试用很短的名字来为一个经常使用的query语句命名并保存).

\f - list all favorite queries.

\f <name> - Invoke a favorite query by its name（激活一个）.

\fs <name> <query> - Save a new favorite query called 'name'（使用名字保存一个query）.

\fd <name> - Delete an existing favorite query by its name（删除保存的query）.

Examples:

# Save a new favorite query.

> \fs simple select \* from abc where a is not Null;

# List all favorite queries.

> \f

+--------+---------------------------------------+

| Name | Query |

|--------+---------------------------------------|

| simple | SELECT \* FROM abc where a is not NULL |

+--------+---------------------------------------+

# Run a favorite query.

> \f simple

+--------+--------+

| a | b |

|--------+--------|

| 日本語 | 日本語 |

+--------+--------+

# Delete a favorite query.

> \fd simple

simple: Deleted

# Logging

There are two types of logging.

1. Operational logs are on by default and stored in the home folder as ~/.mycli.log. This is useful for debugging or developing mycli. But not very useful for end users.
2. Audit logs（编辑日志） can be optionally turned on（开启） using a command line option -l. This will log every command and it's output with a timestamp（带时间戳的形式记录）. This is useful for keeping track of all that happened in the CLI during a session.

# Editor

One of the most useful slash commands is \e. This will launch an editor to edit the current query (or previous query, if the current query is blank).

Here are some examples:

# This will launch the default editor (read from $EDITOR env variable).

# It will put the previous query in the editor's buffer.

> \e

# This will open the default editor with the current query in the buffer.

> SELECT \* FROM \e

# This will open an existing file in the editor and the contents of the file

will be populated into the prompt once the file is saved.

> \e <filename>

编辑完文件后就会自动将编辑的query显示在命令行中，而且可以多次修改。

# Multi-line Mode

The default behavior of mycli is to execute a query as soon as the enter key is pressed（默认按下回车键就会执行语句，不需要最后的；）. This might come as a surprise to seasoned MySQL users who always enter a semi-colon to denote the end of the query.

To emulate the original MySQL behavior you can enable the multi-line mode(多行模式的情况下需要;来控制执行) which will require a semi-colon to be present at the end of the line. This enables writing multi-line queries if they are really long. You can enable multi-line mode by pressing <F3> key or set it permanently via the [config](http://www.mycli.net/config) file.

# History and Search

MyCli keeps track of the queries entered in the repl. Up/Down arrow can be used to navigate the history.

Pressing <C-r> will enable incremental history search. So press <C-r> and then start typing your search term to see the queries narrowed down. You can cycle through the matches by pressing <C-r> again（在键入搜索词后，可以继续按<C-r>来循环查找，就像vi里面的N）.

# Storing server configuration

When working with multiple servers（多服务器）, it is advisable to use the [mysql\_config\_editor](https://dev.mysql.com/doc/refman/5.7/en/mysql-config-editor.html) tool（这个命令应该是属于mysql） that ships with MySQL to store the authentication credentials in an encrypted file called .mylogin.cnf. Then, mycli can read and use those credentials.

First, run the mysql\_config\_editor tool and enter the credentials:

$ mysql\_config\_editor set --login-path=my\_server\_alias --host=my.host.address --user=myusername --password

Enter password: \*\*\*\*\*

Then, provide the --login-path option when running mycli:

$ mycli --login-path my\_server\_alias my\_database\_name

Pager

Mycli outputs query results in a pager (e.g. less, more（windows下不支持less，可以使用more）)（分页输出）. This allows you to easily view large result sets one page at a time.

Configuring the Pager

When starting up, mycli checks the pager config option in MySQL's option files. If it's set, then mycli uses its value as the pager. If it's blank, then mycli will use the PAGER environment variable's value.

Once mycli is started, you can use the pager command to change which pager mycli uses. Or, nopager will disable the pager.

> pager less

PAGER set to less.

> nopager

Pager disabled.

Disable Paging

You can disable the pager by adding skip-pager = on to the [client] section of your MySQL option file. See [Configuration](http://www.mycli.net/config)for more information.

Pager Behavior

On macOS and Linux, the pager will default to less for most users. less sometimes has less-than-desirable behavior like clearing the screen, cutting lines off, etc. You can configure less through environment variables in your shell configuration files. Here are some common less options and configuration examples:

* -X leaves file contents on the screen when less exits.
* -F makes less quit if the entire output can be displayed on one screen.
* -R displays ANSI color escape sequences in "raw" form.
* -S disables line wrapping. Side-scroll to see long lines.

# This is a popular option among mycli users.

export LESS="-XFR"

# Some mycli users like to disable line wrapping.

export LESS="-SRXF"