* Login:
  + cd fullstack/vagrant
  + vagrant up
  + vagrant ssh
* Access psql:
  + cd /vagrant/forum
  + psql forum
  + Run queries
  + Exit to psql console, type: q
  + Exit database: \q
* Turn off machine:
  + vagrant halt

psql:

* Update:
  + Update posts set content = ‘cheese’ where content like ‘%spam%’;
* Delete:
  + Delete from posts where content = ‘cheese’;

The **psql** command-line tool is really powerful. There's a complete reference to it [in the PostgreSQL documentation](http://www.postgresql.org/docs/9.4/static/app-psql.html).

To connect **psql** to a database running on the same machine (such as your VM), all you need to give it is the database name. For instance, the command **psql forum** will connect to the **forum** database.

From within **psql**, you can run any SQL statement using the tables in the connected database. Make sure to end SQL statements with a semicolon, which is not always required from Python.

You can also use a number of special **psql** commands to get information about the database and make configuration changes. The **\d posts**command shown in the video is one example — this displays the columns of the **posts** table.

Some other things you can do:

**\dt** — list all the tables in the database.

**\dt+** — list tables plus additional information (notably, how big each table is on disk).

**\H** — switch between printing tables in plain text vs. HTML.

Here's a fun one to run in **psql** while your forum web app is running:

**select \* from posts \watch**

(Note that **\watch** replaces the semicolon.) This will display the contents of the **posts** table and refresh it every two seconds, so you can see changes to the table as you use the app.

In order to do this, you'll need two terminal sessions into your VM — one running the forum app, and the other running **psql**. You can connect to the VM from any number of terminal windows at once — just open up another terminal, change to the **vagrant** directory, and type **vagrant ssh** again.