Here is the tale of my rambling experience with this problem. Would love to see it edited or generalised if you have better experience of the issue... apply a bit of that SO magic.

**Note: Comments in next paragraph applied to Snow Leopard, but not to Lion, which appears to require 64-bit MySQL**

First off, the author (still?) of MySQLdb says here that one of the most pernicious problems is that OS X comes installed with a 32 bit version of Python, but most average joes (myself included) probably jump to install the 64 bit version of MySQL. Bad move... remove the 64 bit version if you have installed it (instructions on this fiddly task are available on SO here), then download and install the 32 bit version (package here)

There are numerous step-by-steps on how to build and install the MySQLdb libraries. They often have subtle differences. This seemed the most popular to me, and provided the working solution. I've reproduced it with a couple of edits below

**Step 0:** Before I start, I assume that you have MySQL, Python, and GCC installed on the mac.

**Step 1:** Download the latest MySQL for Python adapter from SourceForge.

**Step 2:** Extract your downloaded package:

```
tar xzvf MySQL-python-1.2.2.tar.gz
```

**Step 3:** Inside the folder, clean the package:

```
sudo python setup.py clean
```

COUPLE OF EXTRA STEPS, (from this comment)

**Step 3b:** Remove everything under your MySQL-python-1.2.2/build/* directory -- don't trust the "python setup.py clean" to do it for you

**Step 3c:** Remove the egg under Users/$USER/.python-eggs

**Step 4:** Originally required editing _mysql.c, but is now NO LONGER NECESSARY. MySQLdb community seem to have fixed this bug now.

**Step 5:** Create a symbolic link under lib to point to a sub-directory called mysql. This is where it looks for during compilation.

```
sudo ln -s /usr/local/mysql/lib /usr/local/mysql/lib/mysql
```

**Step 6:** Edit the setup_posix.py and change the following

mysql_config.path = "mysql_config"

to

mysql_config.path = "/usr/local/mysql/bin/mysql_config"

**Step 7:** In the same directory, rebuild your package (ignore the warnings that comes with it)

```
sudo python setup.py build
```

**Step 8:** Install the package and you are done.

```
sudo python setup.py install
```

**Step 9:** Test if it's working. It works if you can import MySQLdb.

```
python
```

```
>>> import MySQLdb
```

**Step 10:** If upon trying to import you receive an error complaining that `Library not loaded: libmysqlclient.18.dylib` ending with: `Reason: image not found` you need to create one additional symlink which is:

```
sudo ln -s /usr/local/mysql/lib/libmysqlclient.18.dylib /usr/lib/libmysqlclien
```

You should then be able to `import MySQLdb` without any errors.

One final hiccup though is that if you start Python from the build directory you will get this error:

/Library/Python/2.5/site-packages/MySQL_python-1.2.3c1-py2.5-macosx-10.5-i386.egg/_mysql.py:3: UserWarning: Module _mysql was already imported from /Library/Python/2.5/site-packages/MySQL_python-1.2.3c1-py2.5-macosx-10.5-i386.egg/_mysql.pyc, but XXXX/MySQL-python-1.2.3c1 is being added to sys.path

This is pretty easy to Google, but to save you the trouble you will end up here (or maybe not... not a particularly future-proof URL) and figure out that you need to `cd ..` out of build directory and the error should disappear.

As I wrote at the top, I'd love to see this answer generalised, as there are numerous other specific experiences of this horrible problem out there. Edit away, or provide your own, better answer.