

# pyPgSQL FAQ

**Q:** I'm getting a

```
ImportError: No module named DateTime
```

**A:** You need to install the BASE package from the eGenix mx Extensions:  
<http://www.egenix.com/files/python/>

**Q:** Inserting data for the SQL IN-operator doesn't seem to work.

**A:** In pyPgSQL 2.3, you can use tuples:

```
cursor.execute("select * from test where id in %s", ((3,4,5),))
```

If you want to use ARRAYS, you'll now need to wrap your (nested) lists with `PgSQL.PgArray`.

**Q:** I get an error like:

```
>>> cursor.execute("select foo from bar where baz=%f", (3.14,))
Traceback (most recent call last):
  File "somefile.py", line 1, in ?
  File "pyPgSQL/PgSQL.py", line 2579, in execute
    self.res = self.conn.conn.query(_qstr % parms)
TypeError: bad argument type for built-in operation
```

**A:** Using pyPgSQL, you use %s for all parameters, no matter which type they have.

**Q:** I've heard of Unicode support for pyPgSQL. What's the current status?

**A:** It's integrated in pyPgSQL 2.3.

**Q:** Large Object support doesn't work with Python 2.2. What gives?

**A:** It's a Python bug. Please upgrade to Python 2.2.2.

**Q:** Why isn't `cursor.rowcount` working as expected?

**A:** By default, PgSQL uses PostgreSQL Portals (i.e. cursors). As a result of this, PgSQL doesn't know how many rows resulted from the query until they are fetched. (Note: `rowcount` will be set to the number of rows returned by the `fetchXXX()` method. If you `fetchone()`, `rowcount` will be 1.) There are a couple of ways to get the number of rows returned by a query:

1. Do a `fetchall()`. `rowcount` will then be set to the number of rows retrieved.
2. Set `PgSQL.noPostgresCursor` to 1. This will prevent PgSQL from using PostgreSQL Portals (which will result in the entire result set being read into memory). `Rowcount` will be set to the number of rows retrieved.
3. Execute a **`select count(*) ...`** to get the number of rows.
4. Re-evaluate why you need to know the number of rows before processing the results.

**Q:** When building pyPgSQL, I get an error message like the following: `gcc: unrecognized option '-R/usr/lib/'` What's up?

**A:** This happens with pre-2.2 versions of Python's distutils, when using the gcc compiler. This warning can be safely ignored if your PostgreSQL libraries can normally be found by the dynamic loader. Otherwise, you need to work around this problem by putting the PostgreSQL library directory into the search path of your dynamic loader. See **man 1 ld** for details.

**Q:** I've little or no experience using Python for database programming, where can I find additional documentation?

**A:** pyPgSQL is an implementation of the Python DB-API 2.0 specification (<http://www.python.org/topics/database/DatabaseAPI-2.0.html>). There's also an article from Andrew Kuchling about using the DB-API (<http://www.amk.ca/python/writing/DB-API.html>). Unfortunately, there doesn't seem to be a lot of beginner-level documentation out there currently.

**Q:** How can I speed up the fetching of data with PgSQL?

**A:** You can use

```
PgSQL.fetchReturnsList = 1
```

and

```
PgSQL.noPostgresCursor = 1
```

This will increase the performance of fetches by ca. 25 %. There's potential for increasing the performance of fetches in PgSQL by a lot more. This is something that will be looked onto for future releases of pyPgSQL.

**Q:** Is pyPgSQL available in packaged form by third parties?

**A:** There's a pyPgSQL port available for FreeBSD. It can be installed with **cd /usr/ports/databases/py-pyPgSQL && make install**

pyPgSQL is also available as a Debian package. You can install it with **apt-get install python-pgsql**