

DBI tricks, tips and pitfalls

Read the documentation for the database you are using. Yes, SQL is standard, but that doesn't mean everyone follows the standard. You may also find some neat feature that only your database has.

You can use `DBI::errstr` for the most recent error (but don't wait too long to check it).

Be careful using placeholders that may contain NULL values. "SELECT * FROM table WHERE value = ?" won't do what you want when you execute(undef). Try "SELECT * FROM table WHERE (value = ? OR (? IS NULL AND value IS NULL))."

'use strict' – not DBI specific, but this tip should be in just about every list of perl tips.

Don't bother learning how your database quotes things... use `$dbh->quote()`.

Don't quote when using placeholders.

If using DBI in a CGI, `CGI::Carp(:fatalToBrowser)` can be useful for debugging but may be a security hole.

Working with BLOBs: check `LongReadLen` to make sure you won't get truncated data from a SELECT. Use placeholders because `quote()` might choke.

If you are going to perform an operation more than once, prepare it.

Use here-doc or `qq{ }` constructs to form your SQL code so it is easier to read.

If you need to get just the first few rows back from a large select, call `$sth->finish` when you are done to allow the database to free up memory.

It is probably a good idea to explicitly state the value of `AutoCommit`, since the database will act in a dramatically different manner depending on how this is set.

`$dbh->disconnect` to make perl happy (and for your own peace of mind).

Consider `RaiseError=>1`, especially if you use evals.

Column aliases can be useful to make better key names if you use `fetchrow_hashref`

Fun things you can do (or may have already done) with perl and DBI:

Keep track of articles you have read for research citation purposes (EndNote-style program).

Maintain a local protein or nucleic acid sequence database with annotations and handy links to remote sequence analysis programs. (OK, maybe this is something *I* would do...)

Write your own linux distribution, using MySQL for the package information database.

Keep track of your love life (some people should just use a simple flat-file database for this purpose).

Alternatively, if there isn't much love life to keep track of, start an internet matchmaking service. Rig the results to allow yourself to SELECT the right people. Now you will need a database to keep track of your love life.

While ripping your cds, connect to the CDDDB and get requisite information and place into database with a sequential numbering system. Organize your cds by the same number. Now when the RIAA's thugs come knocking on your door you can quickly locate the cd from which every mp3 on your hard drive was made.

Alternately, you could have a database of mp3s on your hard drive, complete with your own determination of what kind of music they are. Then write a script randomize what music you will hear based on what you tell the computer your mood is.

Gather information on world leaders. Sell this information to the highest bidder. A related idea is to break into CIA headquarters at Langley, write a quick DBD::NOClister driver while suspended from the ceiling (cleverly having given the operator something to cause gastrointestinal discomfort). With NOC list in hand, sell to highest bidder.

Other, perhaps more mundane things to do would include a web-based message board or helpdesk, address book, quiz, searchable database of O'Reilly book reviews...