

A DBI-related glossary (mostly TLA's)

'How to talk like one of them database people'

ACID – atomicity, consistency, isolation, durability – four qualities of reliable storage

CRUD – create, read, update, delete – the four types of statements in the SQL DML. They are INSERT INTO, SELECT – FROM, UPDATE – SET and DELETE FROM, respectively.

CSV – comma-separated values – data in a formatted text file. Columns may be (but are not necessarily) delimited by commas. Perl can treat these as a database with the help of SQL::Statement

DBD – database driver – a module that allows the DBI to access a specific database

\$dbh – database handle – the object which DBI associated with a database connection; returned from the DBI->connect.

DBI – database interface – a collection of Perl code that provides a common API for interacting with databases through the use of modular DBDs.

DDL – data definition language – that portion of SQL which is involved in describing the relations contained in a database. Compare to DML.

DML – data manipulation language – the portion of SQL involving interactions with existing databases. CRUD. Compare to DDL.

\$dsn – data source name – the first argument of the DBI->connect function. Tells DBI which DBD to use. Usually in the form 'DBI:\$driver:\$options'

OODBMS – object-oriented database management system – like an RDBMS but relations are called classes, observations (rows, entities) are called instances, column headings are called attributes and relations can inherit from other relations. I'm sure there's more to it than this and purists are probably rather unhappy with this definition.

RDBMS – relational database management system – a program or group of programs that stores data in groups of tables (or relations) that may be linked to each other.

schema – a set of tables that are logically linked.

\$sth – statement handle – the object that DBI gives you from a \$dbh->prepare(\$sql_statement). This object is meant to be executed.

SQL – structured query language – a means for interacting with databases that is somewhat standard. Some people pronounce this "sequel", but others pronounce it "ess-que-ell" and laugh at the people who pronounce it "sequel". "Squirrel" has even been proposed as an alternate pronunciation.

TLA – Three-letter acronym

Resources

Items were selected for inclusion here because I have read them and found them useful. If an item is not listed here, it does not mean it isn't useful – it could just mean that I haven't read it.

DBI:

perdoc DBI – your first (and usually totally sufficient) source of DBI information. Some introductory information and a good bit of reference, too.

perldoc DBD::{your favorite database} – some DBD's have extensive pod documentation. Others don't.

dbi-users mailing list archive – <http://outside.organic.com/mail-archives/dbi-users>

Programming the Perl DBI Alligator Descartes & Tim Bunce – (The cheetah book). You can read Megan's review on raleigh.pm.org. Here's my take: I see the book as split into three parts. The first talks about the basic ideas behind data storage and specifically when you might want to use a database. This also includes some historical information about SQL and RDBMS. The book then describes the workings of DBI. This middle half is well done but IMHO doesn't offer too much more information than can be found in the DBI perldoc. Still, the information is expanded and sometimes it is just nice to have something in book form. The authors then take a chapter to compare DBI with the Win32::ODBC modules and finish with the third half consisting mostly of reference material, including a thorough comparison of the DBDs and to some extent the various databases which can be interfaced with perl using DBI.

"A short guide to DBI" – <http://www.perl.com/pub/1999/10/DBI.html> – A good overview for when getting started.

SQL in general:

"Introduction to SQL" – <http://w3.one.net/~jhoffman/sqltut.html> – a good intro with examples and even a few self-quizzes.

I'm sure there are plenty of good books about SQL, but I haven't read them.

Some Databases:

MySQL – <http://www.mysql.org> – a fast-light database. Until recently technically non-free, until recently had no transaction support (transaction support probably still beta). Nonetheless a very popular database because it is quick and easy to use.

PostgreSQL – <http://www.postgresql.org> – another popular database, this one is an OODBMS. This one has been around for a long time. It originally had a "relational algebra"-type syntax but SQL was added in 1995. The documentation that comes with the distribution includes a good bit of this history as well as the theory behind databases.

For a discussion of the importance of ACID properties, I suggest "Why not MySQL?" at <http://openacs.org/philosophy/why-not-mysql.html>. Also just interesting reading if you want to get a change from the usual editor or distribution wars.

Other databases worth exploring include: Access, Informix, Interbase, Oracle, Sybase and many, many more.