**SQL Lesson #2**

Previously, on SQL: We logged into the MySQL console for Zephir and played around with INSERT and SELECT statements. With these two commands, we can place data into the database, and then pull it out. Now we'll look at how to edit and delete data, using two new keywords: UPDATE AND DELETE.

In the following examples, we're going to use this hypothetical table of users:

| **id** | **username** | **email** | **date\_created** | **premium\_member** |
| --- | --- | --- | --- | --- |
| 1 | alice | [alice@example.com](mailto:alice@example.com) | 2007-01-04 | false |
| 2 | bob | [bob@example.com](mailto:bob@example.com) | 2012-05-06 | false |
| 3 | charlie | [charlie@example.com](mailto:charlie@example.com) | 2005-07-01 | false |
| 4 | dawn | [dawn@example.com](mailto:dawn@example.com) | 2010-12-12 | true |

The schema for this table is as follows, if you're curious:

CREATE TABLE users (

id SERIAL,

username TEXT,

email TEXT,

date\_created DATE,

premium\_member BOOLEAN

);

**UPDATE**

We often want to change information in the database. For example, if someone edits a post on a blog, or changes their settings on an admin site, we need to edit the existing stored information to match. To do so, we use the UPDATE statement, which behaves like a combination of INSERT and SELECT. The simple form of UPDATE is as follows:

UPDATE table SET column = value WHERE condition;

The WHERE clause and its condition is extremely important: without it, UPDATE will cheerfully change *every row in the table* to match the new values.

Let's write some example statements to update this table. For example, we might decide that we want to make all early members of our site "premium members" as a reward for their support. The following command will set that up for us:

UPDATE users SET premium\_member = true WHERE date\_created < "2008-01-01";

Alice and Charlie are now premium members, because they joined before January 1, 2008.

Perhaps Dawn has decided to change her e-mail address. We can also change a single row using UPDATE with the same syntax, just by using the equality comparison instead of a greater-than or less-than comparison. This is also made easier by the use of the SERIAL type on our id row, which automatically creates a new, unique integer for us whenever we insert a row. That means each user has their own ID, but we don't have to manually keep track of them. To change Dawn's e-mail, we'll run the following statement:

UPDATE users SET email = "dawn.new.address@example.com" WHERE id = 4;

Again, note that unlike in PHP, we don't need to use == for comparison and = for assignment, but can use = for both. Because SQL is a declarative language, it can figure out from context what we mean when we use = in a statement--in comparisons it's a test, but in SET clauses, it performs assignment.

**DELETE/DROP**

Sometimes we just want to get rid of rows. The DELETE keyword lets us drop any rows that match its condition. Like UPDATE, if we don't specify a conditional WHERE clause, it will delete *every* row in a table--handy for resetting a database, but otherwise hazardous. Be sure to specify a condition!

In our sample database, Alice has decided to delete her account, so we're going to remove her from the database table. The following statement will do that for us:

DELETE FROM users WHERE id = 1;

**Best practice note:** Most of the time, you don't want to delete anything from a database, ever. It's almost always better to save information for later use, such as troubleshooting or if someone decides to restore their account. In this case, rather than deleting Alice's information, we should probably have a column in the database named "enabled" or "active" that we can set to false when they "delete" their account. That's not quite as helpful for example code, however!

In addition to being able to delete rows, we can also delete tables. This uses a slightly different keyword: DROP. Unlike with DELETE, there's no need to specify a condition, because you can't test a whole table. To get rid of our users table completely, we'll write this code:

DROP TABLE users;

It's rare to actually drop a database table, but it does happen from time to time.