

What is SQLAlchemy

[SQLAlchemy](#) is an open-source Python Database toolkit, which provides the following functionality:

1. It maps relational databases into objects
2. It manages an applications database connections
3. It can create/alter a database layout if it is allowed to

The most powerful feature of SQLAlchemy is the first point: given a table description, it maps the data in tables into classes of objects, where each instance of an object is a row in the table, and can be worked with like a class or structure in code.

Example:

Given a table called “Users”, with a FirstName and LastName column. Once the columns are described in the Python code, to add a row to the users table might look like this:

```
joebruin = User()
joebruin.FirstName = “Joe”
joebruin.LastName = “Bruin”
joebruin.save()
```

Given a table called “Addresses” with two columns called “Street” and “City”. It’s related to the “Users” table above using a column named User_Id, though SQLAlchemy will handle it for you. To add a row to the address table, and link it to the Users table, it might look like this:

```
uclaaddress = Address()
uclaaddress.Street = “405 Hilgard Ave.”
uclaaddress.City = “Los Angeles”
joebruin.addresses.append(uclaaddress)
uclaaddress.save()
joebruin.save()
```

Note: Even though the id column is in the Addresses table, SQLAlchemy exposes it a collection on the Users table because of the relationship.

Some of the interesting features are things like eager-/lazy-loading, poly-morphic table layouts (in other words, having a table where the rows have a “type” column, then breaking the data elements into subclasses based on that column), caching, and database-platform independence.

More Reading:

[A Critical Examination of SQLAlchemy Performance](#)

Revision #1

Created 2007-11-09 12:20:47 UTC by Garcia, Ricardo

Updated 2007-11-09 12:20:48 UTC by Garcia, Ricardo