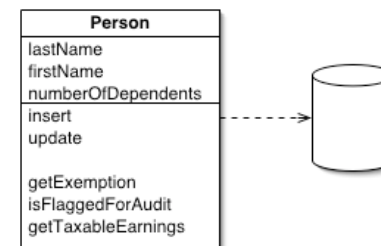


ActiveRecord Quick Reference



Mapping

Mapping to table (maps to pluralization of class name) <pre>class User < ActiveRecord::Base #maps to table users</pre> Override default by calling <code>table_name=</code> method	Mapping primary key (maps to column named 'id') Override default by calling <code>primary_key=</code> method
Relationships: how they are defined <pre>has_many # defines a one to many relationship</pre> <pre>has_one # defines a one to one relationship</pre> <pre>belongs_to # defines a many to one relationship</pre>	Accessors: making them clean Column names are mapped to accessors Can override by calling the <code>alias_attribute</code> method

Creating

Simple: <pre>user = User.new</pre> <pre>user.name = "Cheezy"</pre>	Hash: <pre>user = User.new(:name => "Cheezy",</pre> <pre> :occupation => "Dancer")</pre>
Block: <pre>user = User.new do u </pre> <pre> u.name = "Cheezy"</pre> <pre>end</pre>	

Finding

Find by single attribute User.find_by_email(“a@b.c”)	Find by several attributes User.find_by_email_and_password(“a@b.c” , “pass”)
Find or Initialize (If record exists, returns it. Else return new record without saving it first) User.find_or_initialize_by_email(“a@b.c”)	Find or Create (Creates a new record unless it exists) User.find_or_create_by_email(“a@b.c”)
Conditions: the WHERE part of the sql query :conditions => [“email = ? AND password = ?”, email, pass] :conditions => { :email => email, :password => pass }	Find all that match criteria User.find_all_by_last_name(“Smith”)
Order: specify SQL order by criteria Post.find(:all, :order => ‘column desc’)	Group: an attribute name by which the result should be grouped Post.find(:all, :group => ‘member_id’)
Limit: limits the number of rows to be returned Post.find(:all, :limit => 10)	Offset: an integer determining the offset from where the rows should be fetched Post.find(:offset => 10)

Operate on ActiveRecord objects

Getting the number of rows in a table User.count	Saving an object into the database user.save
Deleting an object from the database user.delete	Deleting all objects in a table User.delete_all