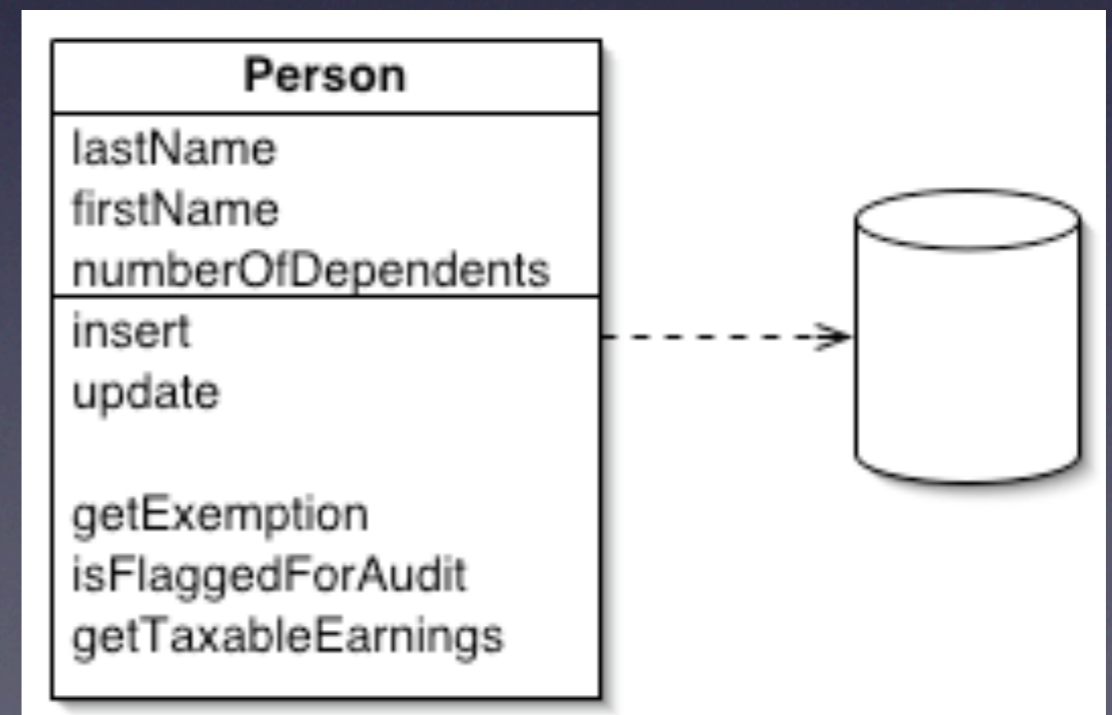


ActiveRecord

Design Pattern

An object that wraps a row in a database table or view, encapsulates the database access, and adds domain logic on that data.

/Martin Fowler/




RoR Active Record

```
class Entry < ActiveRecord::Base
  # nothing here!
end

entry = Entry.new
entry.name = "John"
entry.comments = "What a great web site!"
entry.save

entry = Entry.find(1)
puts entry.name # output: "John"
```



```
CREATE TABLE entries (
  id int(11) NOT NULL auto_increment,
  name varchar(255),
  comments varchar(255),
  PRIMARY KEY (id)
);
```

Tabulas un klases

Klase	Tabula	Klase	Tabula
Order	orders	LineItem	line_items
TaxAgency	tax_agencies	Person	people
Batch	batches	Datum	data
Diagnosis	diagnoses	Quantity	quantities

```
class Klienti < ActiveRecord::Base
  set_table_name "klienti"
end
```

Kolonnas un atribūti

```
def self.up
  create_table :posts do |t|
    t.column :title, :string
    t.column :body, :text
    t.column :created_at, :datetime
  end
end
```

Post.column_names

Post.columns_hash["title"]

Primārās atslēgas

```
post.id
```

```
Post.find(1)
```

```
class Post < ActiveRecord::Base
  set_primary_key "post_id"
end
```

Ieraksta izveidošana

```
post = Post.new  
post.title = "Jauns ieraksts"  
post.body = "Ļoti interesants"  
post.save
```

```
Post.new do |post|  
  post.title = "Jauns ieraksts"  
  post.body = "Ļoti interesants"  
  post.save  
end
```

```
post = Post.new(:title => "Jauns ieraksts",  
               :body => "Ļoti interesants")  
post.save!
```

Ieraksta izveidošana

```
post = Post.create(:title => "Jauns ieraksts",  
                  :body => "Ļoti interesants")
```

```
posts = Post.create([  
  {:title => "Pirmais ieraksts", :body => "Ļoti interesants"},  
  {:title => "Otrais ieraksts", :body => "Vēl interesantāks"},  
])
```

```
post = Post.new(params[:post])
```

Ierakstu meklēšana

```
Post.find(1)
```

```
Post.find(:all)
```

```
Post.find(:all, :conditions => "title = 'Pirmais ieraksts'")
```

```
# NEDARIET ŠĀDI!!!
```

```
title = params[:title]
```

```
Post.find(:all, :conditions => "title = '#{title}'")
```

```
# BET GAN ŠĀDI
```

```
Post.find(:all, :conditions => ["title = ?", title])
```

Ierakstu meklēšana

```
Post.find(:all,  
  :conditions => ["title LIKE ?", title+"%"],  
  :order => "created_at DESC",  
  :limit => 5,  
  :offset => 10)
```

```
Comment.find(:first, :conditions => ["author = ?", author])
```

```
Comment.find_by_author(author)
```

```
Comment.find_all_by_author_and_post_id(author, post_id)
```

```
Post.find_by_sql("SELECT posts.*, user.login FROM posts,  
users" +  
  "WHERE posts.user_id = user.id")
```

Labošana, dzēšana

```
post = Post.find(params[:id])  
post.title = params[:title]  
post.save
```

```
post = Post.find(1)  
post.update_attributes(:title => "Cits virsraksts",  
                      :body => "Cits saturs")  
post.save
```

```
Post.update(params[:id], :title => params[:title])
```

```
Post.delete(123)
```

```
Post.delete([1, 2, 3])
```

```
Post.delete_all("created_at < ?", 1.month.ago)
```

Relācijas

```
has_many :comments  
belongs_to :post  
has_one :post_details  
has_and_belongs_to_many :tags
```

```
comment.post = post  
comment.post.title
```

```
post.comments  
post.build_comment(:author => "Raimonds", :body => "bla bla")  
post.create_comment(params[:comment])
```

```
post.comments.find_all_by_author("Raimonds")
```

Validācijas

```
validates_presence_of :title, :description
```

```
validates_acceptance_of :terms, :message => "Lūdzu  
akceptējiet noteikumus pirms turpiniet"
```

```
validates_confirmation_of :password
```

```
validates_length_of :login, :within => 3..20
```

```
validates_format_of :person_code, :with => /\d{6}-\d{5}/
```

```
validates_uniqueness_of :title
```

Speciālie atribūti

created_at	#{table_name}_count
created_on	position
updated_at	parent_id
updated_on	lft
lock_version	rgt
type	quote
id	template