

Database and ActiveRecord

MIS 21

A. Navarro, R. Alampay



INFORMATION SYSTEMS &
COMPUTER SCIENCE

Learning Objectives

- Define persistence options for web applications
- To define databases and understand basic database concepts
- To understand the ActiveRecord pattern
- To learn the different ActiveRecord methods
- Learn how to utilize ActiveRecord objects in applications



Persistence

- Most modern web applications need a mechanism to store data
- Examples:
 - Amazon needs to keep data about different products for sale
 - Facebook needs to keep data about profiles and posts
 - Instagram needs to keep photos



Databases

- Central repository of shared data
- Data is managed by a controlling agents
- Multiple clients can access data and integrity will still be maintained
- Database Management System (DBMS) – A software system used to create, maintain, and provide controlled access to user databases



DBMS

- A lot of DBMS software is available and free for use
 - SQLite
 - MySQL
 - PostgreSQL
 - MongoDB
 - Redis
- Ruby provides various mechanisms to provide to almost any DBMS



Databases (Cont'd)

- Databases organize data into tables
- Tables are uniquely named
- Consists of rows and columns
- Each row is unique
- Databases allow almost limitless storage of rows of data (for most applications)



An Example Table (User)

id	name	email	password	salary
1	John	john@company.com	U09nu3q4c 4m	10000.00
2	Paul	paul@gmail.com	F09ru2n51v52	25000.00
3	George	george@hotmail.com	90543hfhsfdsa	17000.00
4	Ringo	ringo@yahoo.com	702nc98nc3803	30000.00

- Databases allow you to select specific rows or specific columns of a table of data
- Databases also allow you to combine data from diff tables (More on this later on MIS 122)
- Each row is distinguished using a unique column called a PRIMARY KEY



ActiveRecord

- ActiveRecord is a design pattern where an ActiveRecord instance represents one row of a database table
- The class name of an ActiveRecord object usually matches the table storing rows
- Creating and saving instances of an ActiveRecord object creates a new row on a database
- This is also called “Object Relational Mapping”



Active Record in Ruby

- A ruby gem called ActiveRecord allows this functionality in Ruby
- ActiveRecord is part of the Rails framework
- Classes that represent tables inherit from ActiveRecord::Base which provides database functionality
- These subclasses gain accessors that equate to column names in a database table
- Any change to instances of a this will later update the corresponding row in a database table



Models

- ActiveRecord subclasses are also often called *Models*
- An ActiveRecord object will have a property called `id` (by default) which maps to the PRIMARY KEY of a table



Active Record in Ruby

```
1 # definition of an Active Record class
2 class User < ActiveRecord::Base
3 end
4
5 # You may pass a hash to the initializer to populate
6 # column data
7 user = User.new(name: 'Katy', email: 'katy@gmail.com', password: 'password')
8
9 # The save method inserts the instance into a row in the table
10 user.save!
11
12 # ! or BANG methods are "dangerous" methods
13 # usually they modify the caller
14
15 # The create method is similar to new method
16 # except that it automatically inserts into the database
17 User.create!(name: 'Taylor', email: 'tswizzle@gmail.com', password: 'blah')
```



Active Record in Ruby

- Active Record needs to be initialized for the functionality to be available in a Sinatra or Rails application
- You must execute this code before instantiating any Active Record subclasses

```
7 # database.yml
8 adapter: 'sqlite3'
9 database: 'alingnena.db'
10 pool: 5
11 timeout: 5000
12
13 # another rb file
14 connection_details = YAML::load(File.open('config/database.yml'))
15 ActiveRecord::Base.establish_connection(connection_details)
16
```



Active Record in Ruby

- Active Record also provides class methods that allow you to retrieve data from the database
- `all` – retrieves all rows of a table and returns an array of objects
- `find(id)` – retrieves a row of a database, given an id value and returns instance
- `find_by_attribute` – retrieves a row of a database given an attribute
 - E.g. `find_by_name("Karla")`



Migrations

- Migrations define and create a table in a database

```
1 class CreateItems < ActiveRecord::Migration
2   def change
3     create_table :items do |t|
4       t.string :name
5       t.decimal :price
6       t.integer :quantity
7       t.integer :sold
8     end
9   end
10 end
```



More info

- <http://api.rubyonrails.org/classes/ActiveRecord/Base.html>

