Kolten Robison
Team 1

Unit 10 Topic Report: Active Record

When working in a web framework, raw SQL queries can not only be a daunting task, but they can conflict with your program layout. Because SQL queries lack a class structure, and because relational databases can require many difficult SQL queries, people invented wonderful sub-frameworks to deal with all the nasty database management in a clean and object-oriented manner that also plays well with MVC. Active Record is the gem that was built for the Rails framework to resolve these issues.

Examples of Active Record usage in Rails:

- Querying by primary key

```
#Finding Car with id=12
#Active Record
    car = Car.find(12)
#Raw SQL
    SELECT * FROM cars WHERE (cars.id = 10) LIMIT 1
```

- Testing values existing in an array

```
#Cars with model == Forester or Legacy
#Active Record
    Car.where(model: ["Forester","Legacy"])
#Raw SQL
    SELECT * FROM cars WHERE (cars.model IN ("Forester","Legacy"))
```

- Queries in relation to current time (also using ranges)

```
#Active Record
    User.where(created_at: (Time.now.midnight - 1.day)..Time.now.midnight)
#Raw SQL
    #Generate format like follows, but replace with time (cannot directly call
    ruby classes like Time)
    SELECT * FROM users WHERE (users.created_at BETWEEN '2014-03-25
    00:00:00' AND '2014-03-25 00:00:00')
```

- Relational (Associations) queries

```
#Testing if a car has wheels
#Active Record
    cars.wheels.empty?
#Raw SQL
    SELECT * FROM cars JOIN wheels USING (car_id)
```

Using Active Record can make your code much cleaner, and your life as a programmer easier. Links:

http://guides.rubyonrails.org/active_record_querying.html

http://guides.rubyonrails.org/association_basics.html

http://api.rubyonrails.org/classes/ActiveRecord/Base.html