ASSOCIAÇÕES E VALIDAÇÕES

LEMBRAM DO PRIMEIRO DIA?

Aprendemos sobre e desenhamos nossas próprias associações entre as tabelas. Fomos apresentadas às formas: 1:N e N:N

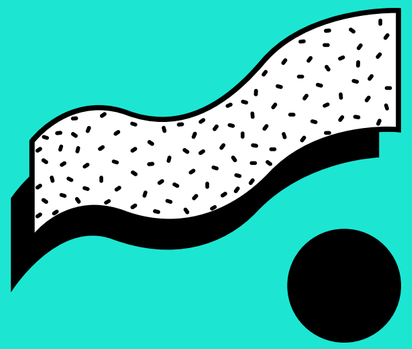
Criando a coluna de associação com outra tabela

```ruby
class CreateProducts < ActiveRecord::Migration[6.0]
  def change
    create_table :products do |t|
      t.string      :description
      t.string      :title
      t.float       :value
      t.references  :store, foreign_key: true
      t.timestamps
    end
  end
end
```

E no model?

```ruby
class Product < ActiveRecord::Base
  belongs_to :store
end
```

Como fazemos para pegar todos os products que pertecem a uma store?

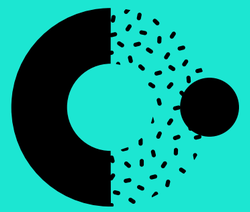


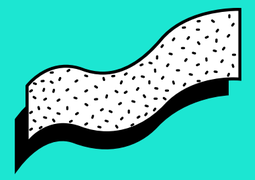
```ruby
store = Store.first
store.products
```

Mas se uma store pertence a um user, podemos ver todos os products de todas as stores de um determinado user?

```ruby
class User < ActiveRecord::Base
  has_many :stores
  has_many :products, through: :stores
end
```

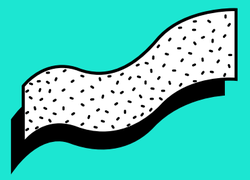
Conheça mais na documentação

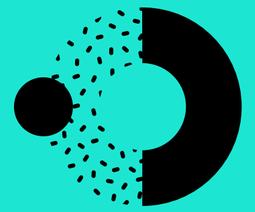




VALIDAÇÕES

Os dados que recebemos, muitas vezes, de diversas fontes podem estar: incorreto, inválido, incompleto ou violando alguma regra de negócio. Por isso precisamos garantir a consistência dos dados que guardamos no banco. Fazemos isso através do uso de validações que configuramos em cada caso para permitir apenas dados corretos.





```ruby
class User < ActiveRecord::Base
  has_many :stores
  has many :products, through: :stores

  validates :email, presence: true
end
```

```ruby
class Product < ActiveRecord::Base
  belongs_to :store

  validates :title, uniqueness: {scope: :store}
end
```

Como vimos existem muitas possibilidades de erros nos dados que recebemos, algumas formas comuns de validação são: ***presence, uniqueness, length, format***

Conheça mais na Doc!

BANCOS NÃO-RELACIONAIS

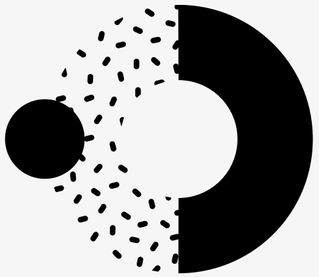
O que são? Onde vivem? O que comem?

NoSQL
Not only SQL

Definição wikipedia

NoSQL (originalmente se referindo a "no SQL": "não SQL" ou "não relacional", posteriormente estendido para Not Only SQL - Não Somente SQL) é um termo genérico que representa os bancos de dados não relacionais.[1] Uma classe definida de banco de dados que fornecem um mecanismo para armazenamento e recuperação de dados que são modelados de formas diferentes das relações tabulares usadas nos bancos de dados relacionais.

USOS NO RDSTATION



Redis

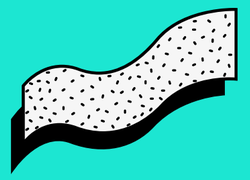
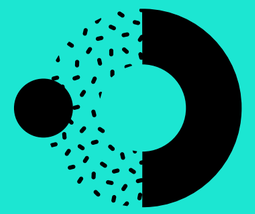
Amplamente usado em envio de jobs para o Sidekiq, além de usos com cache

MongoDB

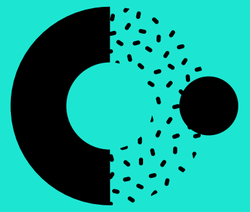
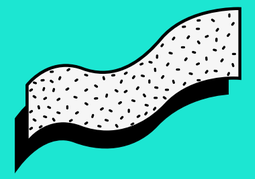
Antigamente era utilizado para fornecer os dados da timeline de leads para o RDSM, está ainda em uso apenas por uma pendência com a funcionalidade Marketing BI

Conheça mais sobre o Redis

```ruby
module DashboardRepairQueue
  class << self

    def enqueue(platform_account_id, force)
      redis.sadd('accounts', platform_account_id)
      redis.set("force:#{platform_account_id}", force)
    end

    def dequeue(platform_account_id)
      redis.srem('accounts', platform_account_id)
      redis.del("force:#{platform_account_id}")
    end

    def account_enqueued?(platform_account_id)
      redis.sismember('accounts', platform_account_id)
    end

    def platform_account_ids
      redis.smembers('accounts').map(&:to_i)
    end

    def force_repair?(platform_account_id)
      redis.get("force:#{platform_account_id}") == 'true'
    end

    private

    def redis
      Redis::Namespace.new('dashboard-repair', redis: RedisFactory.default_client)
    end
  end
end
```

PARSING E SCRAPING

Nokogiri

Possui uma gema maravilhosa para auxiliar com scraping e parsing

CHEATSHEET de Nokogiri

CSV

A Biblioteca padrão do Ruby de CSV é muito poderosa e útil quando temos dados em csv que queremos inserir em um banco de dados

Yaml e JSON

Também são bibliotecas padrão do Ruby e possuem diveras funções para nos ajudar a fazer parse de dados nesses formatos

- Yaml
- JSON

Você chegou!