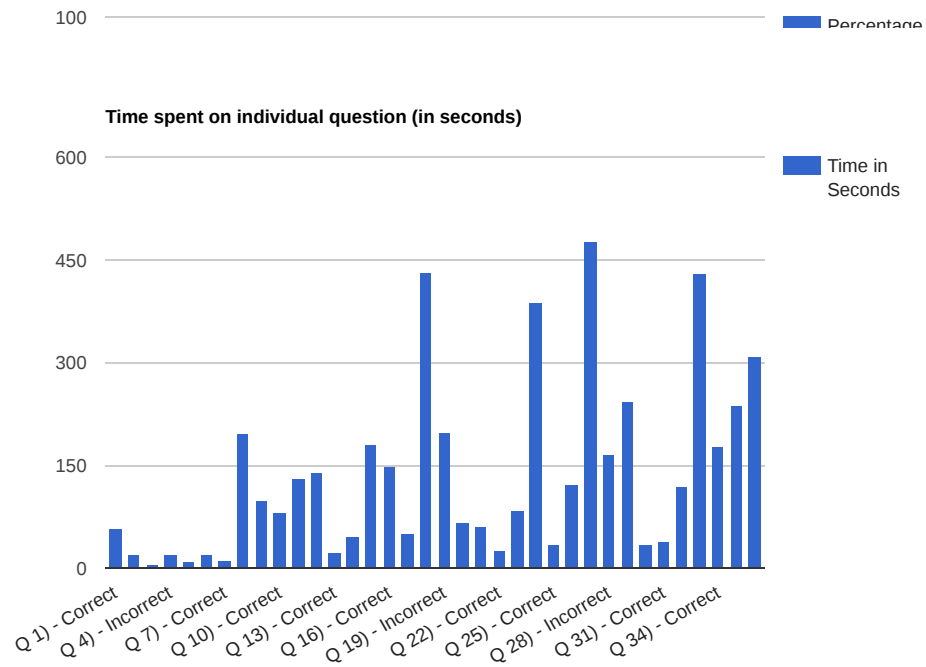


First Name	Anh
Last Name	Nguyen
Email	anhnk@nustechology.com
Quiz Name	[Training Test] Active Record Associations
Attempt Time	2018-04-23 07:33:33
Time Spent (Approx.)	81:28
Percentage	77.7778%
Percentile	100%
Score	28
Status	Fail

Categorywise Analysis

Category Name	Score	Time Spent (Approx.)	Correct	Incorrect	
[Training Test] Active Record Associations	28	81:28 Min.	28	4	4
Total	28	81:28 Min. (Approx.)	28	4	4

Top 10 results for Quiz: [Training Test] Active Record Associations



Answer Sheet

1

Có bao nhiêu kiểu association trong rails model, kể tên?



Answer

Word counts: 23

Có 6 kiểu association trong rails model

2

belongs_to

has_one

has_one through:

has_many

has_many through:

Một user có nhiều comments. Đây là mối quan hệ gì trong rails association ?



3

Answer

Word counts: 2

has_many

Một comment thuộc về một user nào đó. Đây là mối quan hệ gì trong rails association ?

Answer

Word counts: 2

belongs_to

Một bài post có thể thuộc nhiều categories khác nhau và một category có thể có nhiều bài posts. Đây có thể là những mối quan hệ gì trong rails association ?

Answer

Word counts: 4

has_and_belongs_to

has_many :through

has_and_belongs_to_many

Một user có một wallet. Đây là mối quan hệ gì trong rails association ?

Answer

Word counts: 2

has_one

Một user có nhiều bài posts, một bài post có nhiều comments



7

Vậy một user có nhiều comments. Đây là mối quan hệ gì trong rails association ?

Answer

Word counts: 3

has_many through:

Một user có một wallet, một wallet có một bank account



8

Vậy một user có một bank account. Đây là mối quan hệ gì trong rails association ?

Answer

Word counts: 3

has_one through:

Cho biết điểm khác nhau giữa has_many :through và

has_and_belongs_to_many

Khi nào sử dụng has_many :through, khi nào sử dụng

has_and_belongs_to_many



9

Answer

Word counts: 86

has_many through: tạo ra một bảng trung gian khi liên kết giữa hai model
has_and_belongs_to liên kết trực tiếp hai model

Sử dụng has_many through: khi cần nơi để xử lý data trên dữ liệu liên kết giữa hai model/validation

Sử dụng has_and_belongs_to_many khi cần gọn nhẹ,

Ta có các mối quan hệ sau:



10

```
class A < ApplicationRecord
  has_many :bs
end
```

```
class B < ApplicationRecord
  has_many :cs
end
```

Mối quan hệ giữa A và C là gì và hãy set up nó trong class A

Answer

Word counts: 12

```
class A < ApplicationRecord
  has_many :bs
  has_many :cs through: :bs
end
```

Ta có các mối quan hệ sau:



11

```
class A < ApplicationRecord
  has_many :bs
end
```

```
class B < ApplicationRecord
  belongs_to :a
  belongs_to :c
end
```

```
class C < ApplicationRecord
  has_many :bs
end
```

Mối quan hệ giữa A và C là gì và hãy set up nó trong class A

Answer

Word counts: 12

```
class A < ApplicationRecord
  has_many :bs
  has_many :cs through: :bs
end
```

Bảng Employee được dùng để lưu thông tin của nhân viên:
Nhân viên A có thể quản lý nhiều nhân viên khác (có nhiều staff). Nhân viên A này cũng chịu sự quản lý từ cấp cao hơn (chịu quản lý mới manager).



12

Hãy viết association để có thể tìm được nhân viên cấp dưới (staff), người quản lý của nhân viên (manager) (2 associations khác nhau)

Answer

Word counts: 20

```
class Employee < ApplicationRecord
  has_many :staffs, class_name: "Employee", foreign_key:
    "manager_id"
  belongs_to :manager, class_name: "Employee"
end
```

1 User có nhiều bài Post, 1 bài post thuộc về 1 user. Bảng Post đã có sẵn cột people_id type :integer.



13

Hãy viết association thể hiện mối quan hệ giữa bảng user và bảng post dựa theo cột people_id (không cần tạo thêm cột user_id cho bảng post)

Answer

Word counts: 21

```
class User < ApplicationRecord
  has_many :posts
end

class Post < ApplicationRecord
  belongs_to :user, class_name: "User", foreign_key:
    "people_id"
end
```

Cho biết lợi ích khi dùng Association?



14

Answer

Word counts: 22

Dễ dàng hơn trong việc liên kết các model, ít code, dễ hiểu hơn



Cho 2 model users và posts như sau:

15

```
# app/models/user.rb
class User < ActiveRecord::Base
end
```

```
# app/models/post.rb
class Post < ActiveRecord::Base
```

end

Giả sử 1 user luôn có nhiều bài post và một bài post PHẢI thuộc một user.

Table users có các column: username, full_name

Table posts có các column: published_at, user_id, content.

Cho trước: @user = User.first

Trình bày cách dùng association để tạo mối quan hệ 1 user nhiều bài post và khi xóa user thì tất cả bài post sẽ được xóa.

Answer

Word counts: 26

```
# app/models/user.rb
class User < ActiveRecord::Base
  has_many :posts
end

# app/models/post.rb
class Post < ActiveRecord::Base
  belongs_to :user, dependent: :destroy
end

-----

# app/models/user.rb
class User < ActiveRecord::Base
  has_many :posts, dependent: :destroy
end

# app/models/post.rb
class Post < ActiveRecord::Base
  belongs_to :user
end
```

Cũng trong ngữ cảnh của câu hỏi trên.
Hãy trình bày cách tạo một bài post với association và không



dùng với association?

Answer

Word counts: 19

```
Với association: @post = @user.posts.create()  
Không dùng association: @post = Post.create(user_id:  
@user.id)
```

Cũng trong ngữ cảnh của câu hỏi trên.



17

Giả sử khi xóa 1 user, tất cả bài post sẽ được xóa. Hãy trình bày cách làm với association và không dùng với Association?

Answer

Word counts: 24

```
Với association: @user.destroy  
Không association:  
@posts = Post.where(user_id: @user.id)  
@posts.each do |post|  
  post.destroy  
end  
@user.destroy
```



18

Cho model dưới đây

```
class Author < ApplicationRecord
```

```
  has_many :books
```

```
end
```

author là 1 object của model Author.

Nếu chỉ muốn lấy những cuốn sách đã được in rồi (printed: true) mỗi lần gọi author.books thì chúng ta phải khai báo association ntn ?

Answer

Word counts: 16

```
Cho model dưới đây  
class Author < ApplicationRecord  
  has_many :books , -> { where printed: true }  
end
```

Cho model dưới đây

```
class Author < ApplicationRecord
```

```
  has_many :books
```

```
end
```

19

Book có rất nhiều attributes, và ta chỉ cần lấy title, published_date thì chúng ta phải khai báo association ntn ?

Answer

Word counts: 0

```
has_many :books, -> { select(:title, :published)}
```



20

Cho các model dưới đây

```
class Author < ApplicationRecord
  has_many :books
end
```

```
class Book < ApplicationRecord
  belongs_to :author
  has_many :line_items
end
```

Các object của Author thường truy xuất line_items thông qua books (author.books.first.line_items). Để tăng hiệu suất truy xuất thì khai báo association ntn?

Answer

Word counts: 14

```
class Book < ApplicationRecord
  belongs_to :author
  has_many :line_items, -> { includes: :authors }
end

-----
has_many :books, -> { includes :line_items }
```



21

Cho các model dưới đây

```
class Author < ApplicationRecord
  has_many :books
end
```

Muốn author.books có kết quả là danh sách các cuốn sách được sắp xếp theo thứ tự ngày xuất bản (published_at) giảm dần thì khai báo association ntn?

Answer

Word counts: 11

```
class Author < ApplicationRecord
  has_many :books, { order "published_at DESC" }
end
```

Làm sao để định nghĩa mối quan hệ một - một với Rails ?



22

Answer

Word counts: 8

Sử dụng `has_one` và `belongs_to`

Cho đoạn code sau, phát hiện điểm sai?

```
class Book < ApplicationRecord
  belongs_to :authors
end
```



23

Answer

Word counts: 4

authors số nhiều

Làm sao để định nghĩa mối quan hệ nhiều - nhiều với Rails ?



24

Answer

Word counts: 13

Sử dụng `has_many through:` hoặc
`has_and_belongs_to_many`

Ở tầng database, quan hệ `has_one` và `has_many` có gì khác



25

nhau ?

Answer

Word counts: 6

Không khác biệt

Làm sao để một model thuộc về hai model khác nhau, nhưng dùng chung một quan hệ



26

ví dụ:

User có nhiều avatar(image)

Post có nhiều image

image sẽ thuộc về user hoặc là post

Answer

Word counts: 5

Sử dụng polymorphic association

Cho mỗi quan hệ sau:

27

```
class Supplier < ApplicationRecord
  has_one :account
  has_one :account_history, through: :account
end
```

```
class Account < ApplicationRecord
  belongs_to :supplier
  has_one :account_history
end
```

```
class AccountHistory < ApplicationRecord
  belongs_to :account
```

end

Làm cách nào để chuyển mối quan hệ
has_one :account_history, through: :account
thành
has_one :acc_history, through: :account

Answer

```
class Supplier < ApplicationRecord
  has_one :account
  class Supplier < ApplicationRecord
    has_one :acc_history, through: :account, source: :account_history
    has_one :account
  end
  has_one :acc_history, through: :account, source:
    :account_history
end
```

Cho biết option source_type được sử dụng để làm gì và cho ví dụ



28

Answer

Word counts: 104

```
Chỉ định class sẽ lấy dữ liệu trong polymorphic
association

class Tag < ActiveRecord::Base
  has_many :taggings, :dependent => :destroy
  has_many :books, :through => :taggings, :source =>
    :taggable, :source_type => "Book"
```



29

Tiếp theo câu hỏi số, nếu ta chỉ khai báo cho User, không khai báo cho Post (chỉ còn 1 chiều), ta có sử dụng được các method được associations cung cấp cho User không? Ví dụ như user.posts, user.posts.new, user.posts << post, ...

Answer

Word counts: 2

Không

Có

Có 2 model User và Post, hãy khai báo mối quan hệ 1 user nhiều posts theo cả 2 chiều dùng convention

30

Answer

Word counts: 0

User:
has_many :posts
Post:
belongs_to :user

Tiếp theo câu hỏi trên, tables nào sẽ chứa khóa ngoại của mối quan hệ này, khóa ngoại đó phải có tên là gì



31

Answer
Word counts: 19

Table posts sẽ chứa khóa ngoại và khóa này tên là user_id

Có mấy cách biểu diễn mối quan hệ nhiều - nhiều? Kể tên từ khóa sử dụng cho mỗi cách



32

Answer
Word counts: 11

Có hai cách
has_many
has_and_belongs_to_many

Tiếp theo câu hỏi số trên, sự khác nhau giữa 2 cách là gì (khác nhau về database, khác nhau về model), khi nào ta nên sử dụng 'has_many with :through option'



33

Answer
Word counts: 70

has_many sẽ tạo thêm một bảng mới trong database khi liên kết hai model
has_and_belongs_to_many liên kết trực tiếp hai model với nhau

Nên sử dụng has_many với :through option khi định nghĩa quan hệ nhiều nhiều có tạo thêm bảng mới



34

Cho mỗi quan hệ sau:

```
class Book < ApplicationRecord
  belongs_to :author
end
class Author < ApplicationRecord
  has_many :books, dependent: :destroy
end
```

Giải thích ý nghĩa của dependent: :destroy
Nếu thay dependent: :destroy lần lượt bằng delete_all, nullify, restrict_with_exception, restrict_with_error thì lúc này ý nghĩa thay đổi ntn

Answer

Word counts: 96

Khi xóa Author sẽ xóa luôn các cuốn sách thuộc về Author đó
delete_all : xóa tất cả associated object ra khỏi database và không thể thực hiện callback
nullify : set foreign_key thành NULL, callback không thể thực hiện
restrict_with_exception : raise một exception khi tồn tại



35

Cho mỗi quan hệ sau:

```
class Assembly < ApplicationRecord
  has_and_belongs_to_many :parts
end

class Part < ApplicationRecord
  has_and_belongs_to_many :assemblies
end
```

Mặc định join table / table trung gian sẽ có tên gì ?
Làm cách nào để thay đổi tên mặc định này ?

Answer

Word counts: 35

assemblies_parts

lass Assembly < ApplicationRecord
has_and_belongs_to_many :parts, join_table:
this_is_the_name
end



36

Cho mỗi quan hệ sau:

```
class Author < ApplicationRecord
  has_many :books
end
```

Giả sử

```
a = Author.find 100 (Lấy author có id=100)
puts a.book_ids cho ra kết quả [1, 2, 4]
```

Chuyện gì sẽ xảy ra nếu ta thực hiện a.book_ids = [1, 3, 4]

Answer

Word counts: 17

Lúc này author có id 100 sẽ có 3 cuốn sách với id là 1,3,4

record book có id = 2: col author_id sẽ được set nil
record book có id = 3: col author_id sẽ được set 100
record book có id = 1 và 4: không có gì thay đổi

Cho mối quan hệ sau:

```
class Part < ApplicationRecord
  has_and_belongs_to_many :assemblies
end
```

Giả sử

```
part = Part.find 100 (Lấy part có id=100)
puts part.assembly_ids cho ra kết quả [1, 2, 4]
```

Chuyện gì sẽ xảy ra nếu ta thực hiện part.assembly_ids = [1, 3, 4]

Answer

Word counts: 0

record book có id = 2: col author_id sẽ được set nil
record book có id = 3: col author_id sẽ được set 100
record book có id = 1 và 4: không có gì thay đổi

