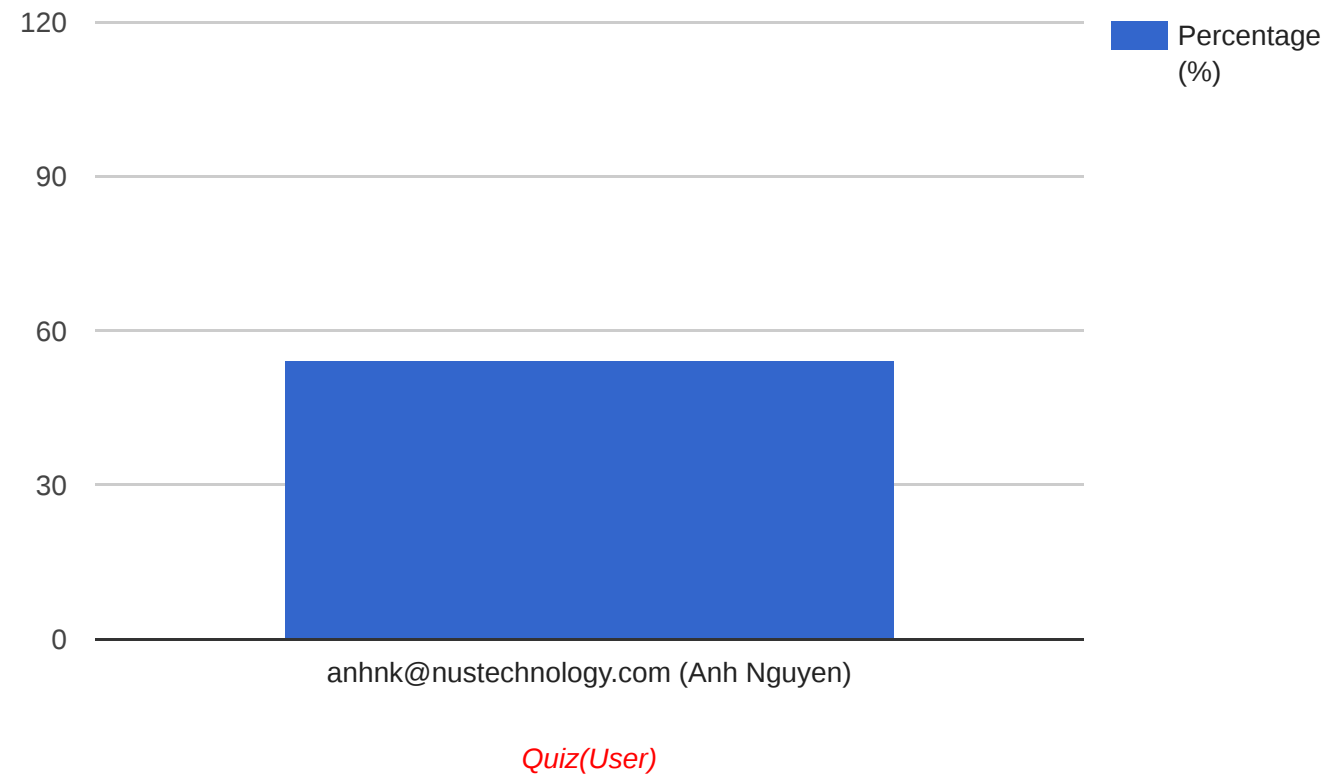


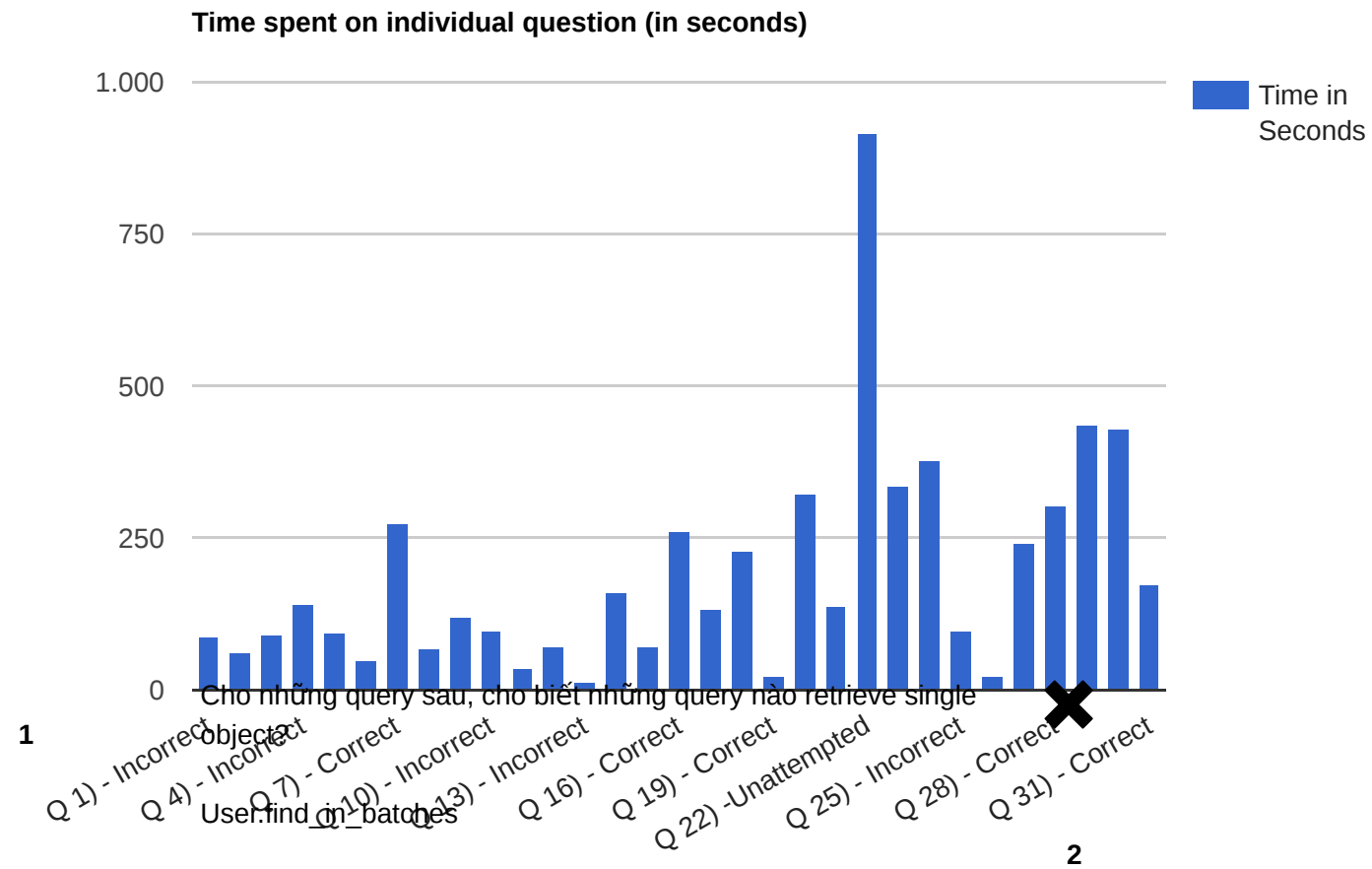
First Name	Anh
Last Name	Nguyen
Email	anhnk@nustechology.com
Quiz Name	[Training Test] Active Record Query Interface
Attempt Time	2018-04-24 06:45:11
Time Spent (Approx.)	97:46
Percentage	54.8387%
Percentile	100%
Score	17
Status	Fail

## Categorywise Analysis

Category Name	Score	Time Spent (Approx.)	Correct	Incorrect	
[Training Test] Active Record Query Interface	17	97:46 Min.	17	13	1
<b>Total</b>	<b>17</b>	<b>97:46 Min. (Approx.)</b>	<b>17</b>	<b>13</b>	<b>1</b>

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





User.first

User.where email: "sam@nustechology.com"

User.last

User.find 2

User.take

User.find\_each

User.find\_by\_email "sam@nustechonology.com"

User.find\_by email: "sam@nustechonology.com"

User.all

Answer

Word counts: 14

User.first / User.last / User.find 2 / User.take / User.where  
email: "sam@nustechonology.com"

-----

User.first

User.last

User.find 2

User.take

User.find\_by\_email "sam@nustechonology.com"

User.find\_by email: "sam@nustechonology.com"

So sánh



3

User.find 1

User.find [1, 2]

Answer

Word counts: 20

User.find 1 => Lấy ra user có id = 1

User.find [1,2] => Lấy ra các user có id = 1 và id = 2

So sánh



4

User.find [1, 2]

và User.where id: [1, 2]

Answer

Word counts: 19

Khác biệt khi không tìm thấy object

User.find[1,2] => exception

User.where id: [1,2] => nil

-----

"- Tìm User id=1 or 2, output là array các instance của class User (model User)

- Tìm User id=1 or 2, output là array các instance của class User (model User) được wrap trong class

User::ActiveRecord\_Relation. Vì vậy, có thể thực hiện query tiếp tục nếu muốn: User.where(id: [1, 2]).active

"

So sánh cách thức hoạt động của



5

User.all

User.find\_each

User.find\_in\_batches

Answer

Word counts: 52

User.all => Lấy tất cả các object user

User.find\_each => Duyệt qua tất cả các object và tìm object phù hợp

User.find\_in\_batches => Duyệt qua tất cả các object trong một khoảng giới hạn

-----

"- Load tất cả các user lên 1 lúc

- Load lần lượt các batches user theo 1 batch\_size nào đó,  
truyền lần lượt từng user trong batch vào block  
- Load lần lượt các batches user theo 1 batch\_size nào đó,  
truyền nguyên user batch vào block  
"

So sánh

User.find 1



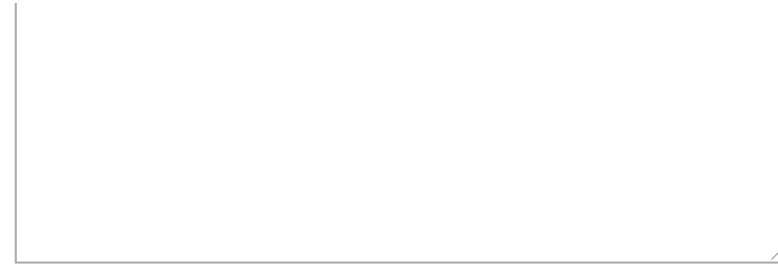


User.find\_by\_id 1

Answer

Word counts: 25

User.find 1 => Khi không tìm thấy sẽ raise exception  
User.find\_by\_id 1 => Không tìm thấy trả về nil



So sánh



7

User.find\_by\_name "samnus"

User.find\_by\_name! "samnus"

Answer

Word counts: 32

User.find\_by\_name "samnus" => Trả về nil nếu không tìm thấy user

User.find\_by\_name! "samnus" => Exception nếu không tìm thấy user

Bảng Product có column "active" với 2 giá trị true, false. Hãy viết 1 scope để khi gọi scope này sẽ lấy được các Product có giá trị active = true

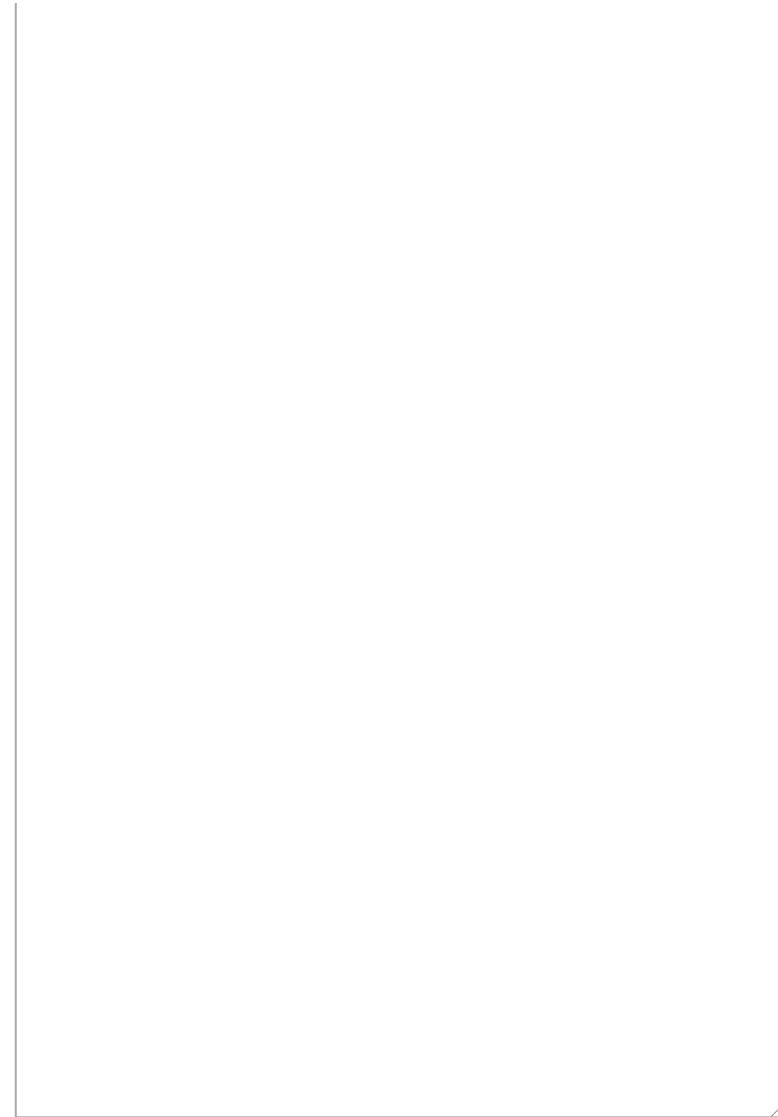
Answer

Word counts: 9

```
Class Product < ApplicationRecord
  scope :active { where(active: true) }
end
```



8



Bảng Product có column "deleted" với 2 giá trị true, false. Hãy viết 1 scope mặc định cho Product chỉ lấy các giá trị deleted = false

Answer

Word counts: 9



9

```
Class Product < ApplicationRecord
  default_scope { where(deleted: false) }
end
```

```
class User < ApplicationRecord
  default_scope { where state: 'pending' }
  scope :active, -> { where state: 'active' }
end
```



10

User.active sẽ tương ứng với câu Query này sau đây?

**Your Answer:** SELECT "users".\* FROM "users" WHERE  
"users"."state" = 'active'

**Correct Options:** SELECT "users".\* FROM "users" WHERE  
"users"."state" = 'pending' AND "users"."state" = 'active'



11

```
class User < ApplicationRecord
  default_scope { where deleted: false }
  scope:active, -> { where state:'active' }
end
```

Lệnh nào sau đây sẽ lấy tất cả User có state = 'active' bất kể  
column delete là true hay false:

**Your Answer:** cả 3 câu trên đều sai,

**Correct Options:** User.unscoped.active



12

Câu nào dưới đây là string condition?

**Your Answer:** User.where("id = #{id}"),

**Correct Options:** User.where('id = 1') , User.where("id = #{id}")



13

Câu nào dưới đây là array condition?

**Your Answer:** User.where('id=?', 1),

**Correct Options:** User.where(['id=?',1]), User.where("id = #{id}"), User.where('id=?', 1)

Câu query nào dưới đây sai?



14

**Your Answer:** User.where('id=#{1}'),

**Correct Options:** User.where('id=#{1}'), User.not.where(id:1)

Với placeholder condition, parameter truyền vào câu lệnh where là gì?



15

Answer

Word counts: 1

key



Làm sao để truy vấn các user có id là 1,2,3?

**Your Answer:** User.where(id: [1,2,3]), User.where(id: [1..3]),  
User.where(id: 1).where(id:2).where(id: 3),

**Correct Options:** User.where(id: [1,2,3]), User.where(id: [1..3])



16

Câu truy vấn sau có vấn đề gì không?

Client.where("orders\_count = #{params[:orders]}")



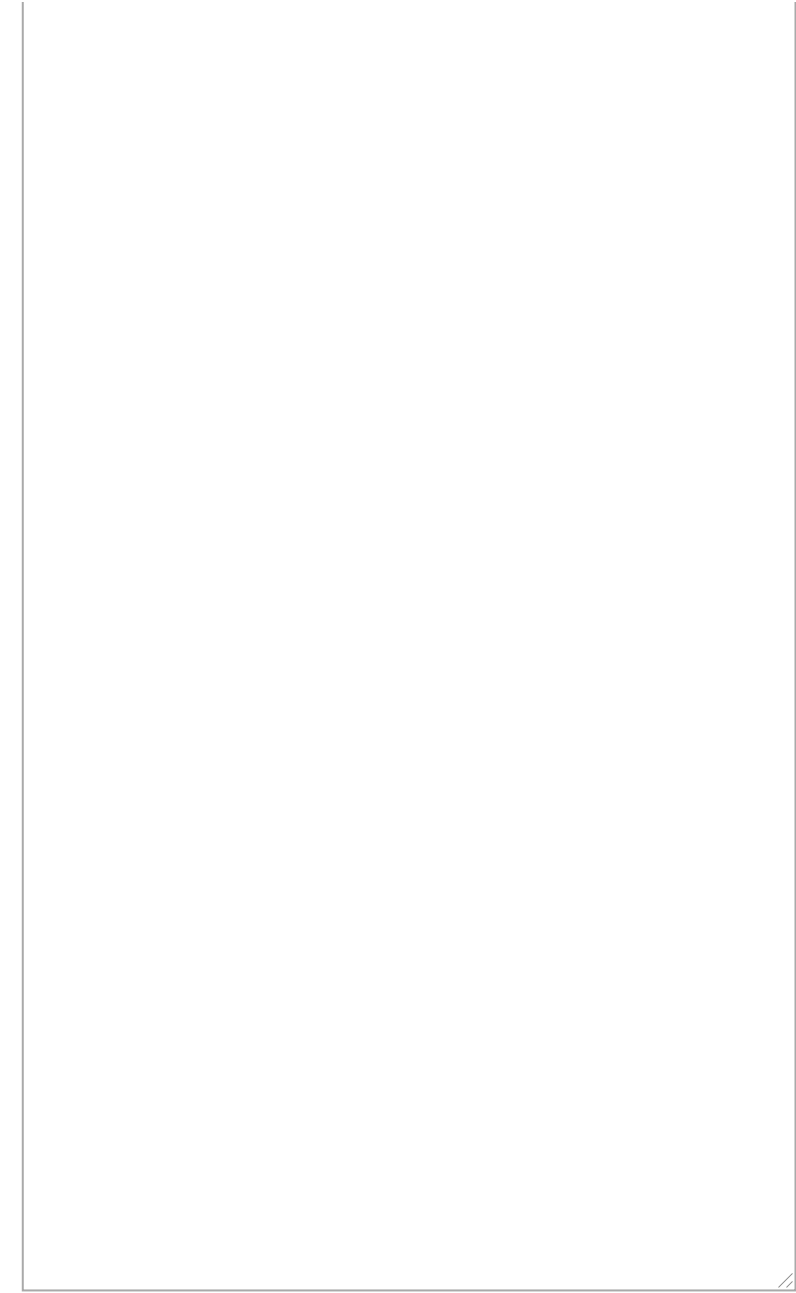
17

Answer

Word counts: 28

Gây nguy hiểm cho database vì người dùng có thể khai thác lỗi hỏng database bằng kỹ thuật SQL Injection





Nếu bạn là một hacker, và đối phương đang sử dụng câu truy



vấn trên, bạn sẽ truyền gì vào params :order để gây ra SQL injection?

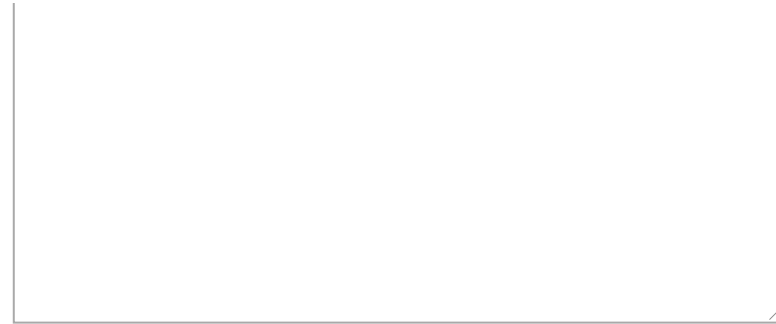
Answer

Word counts: 2

' OR 1 --

-----

params[:order] = "1 or 1 = 1"



19

Cho câu query sau: `Client.where("orders_count = #{params[:orders]}")`

Bạn cần viết câu truy vấn thế nào, để có cùng mục đích mà vẫn an toàn? Giải thích

Answer

Word counts: 6

`Client.where("orders_count = ?", params[:orders])`



Cách viết câu truy vấn dùng placeholder sau có an toàn (tránh được SQL Injection) không, nếu không thì cần viết như thế nào để có tác dụng tương đương?

```
Client.where("created_at >= :start_date AND created_at <= :end_date",  
{start_date: params[:start_date], end_date:  
params[:end_date]})
```

Answer

Word counts: 1

Có

20

Kể tên những hàm (của ActiveRecord) giúp ta lấy được



multiple objects?

Answer

Word counts: 10

find\_each

find\_in\_batches

all

find([ ])

take()

first()

last()

-----

find\_each

find\_in\_batches

all



Giải thích ý nghĩa của lệnh sau:

```
User.find_each(start: 2000, finish: 10000, batch_size: 5000) do  
  |user|  
  # blah blah blah  
end
```

Answer

Word counts: 35

Duyệt tìm qua tất cả các object User bắt đầu từ user có id 2000 tới 10000 nhưng batch\_size chỉ có 5000 nên sẽ dừng ở user có id 7000



22

Tìm điểm khác nhau của 2 đoạn chương trình sau:

1. users = User.all; age = 18;

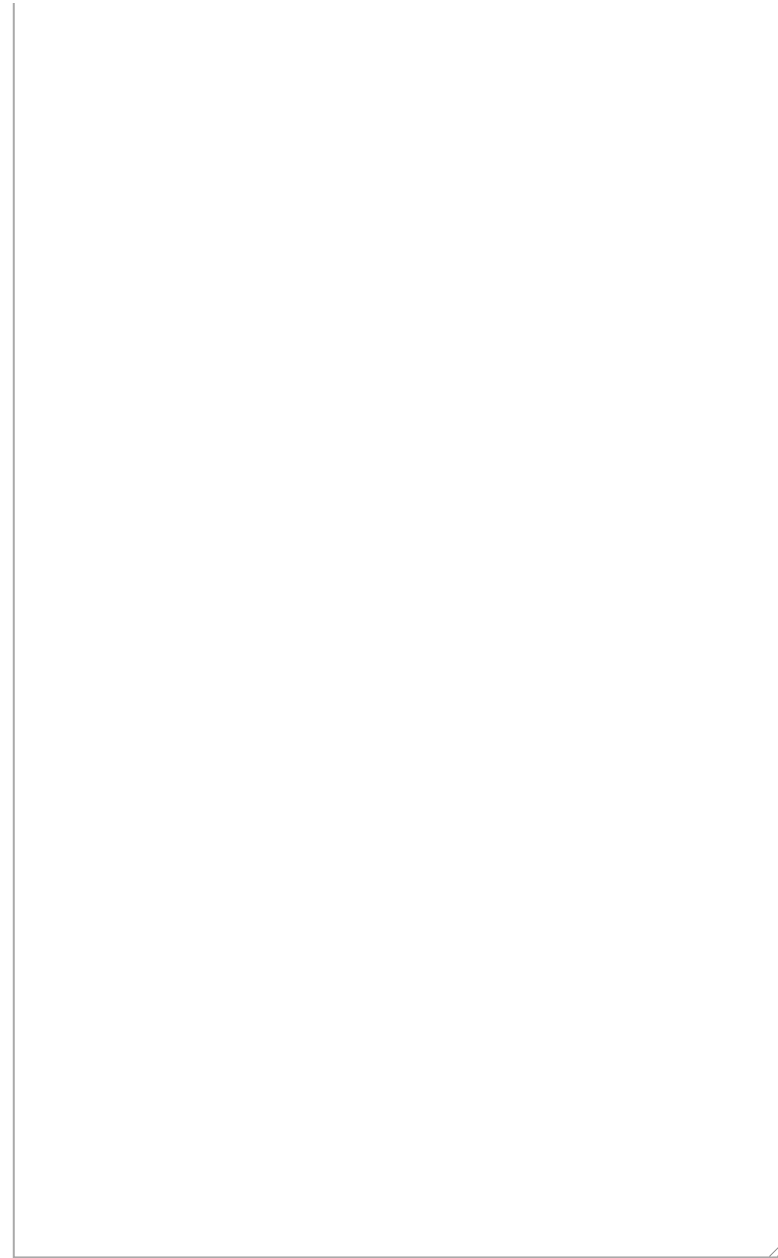
2. users = User.all

Answer

Word counts: 0

- "1. không query vì users không có nhu cầu sử dụng
- 2. query all user trong users table vì có nhu cầu sử dụng"





Các truy vấn database trong rails là lazy load (khi nào cần mới load). Làm sao để force records load



Answer

Word counts: 4

Sử dụng includes

-----

"dùng method load

ví dụ: users = User.all.load"

size khác với count, và length chỗ nào?



25

Answer

Word counts: 77

length sẽ load tất cả các object lên để đếm  
count sẽ thực hiện việc đếm ở database  
size sẽ kiểm tra xem object đã được load lên chưa, nếu true  
=> trả về length mà không phải gọi tới database. Nếu false  
size sẽ kiểm tra counter\_cache (nếu có). Nếu tất cả đều  
false size sẽ gọi đến count



Câu lệnh dưới cho ra kết quả gì? Giải thích  
`User.last(10).order(:first_name)`

26

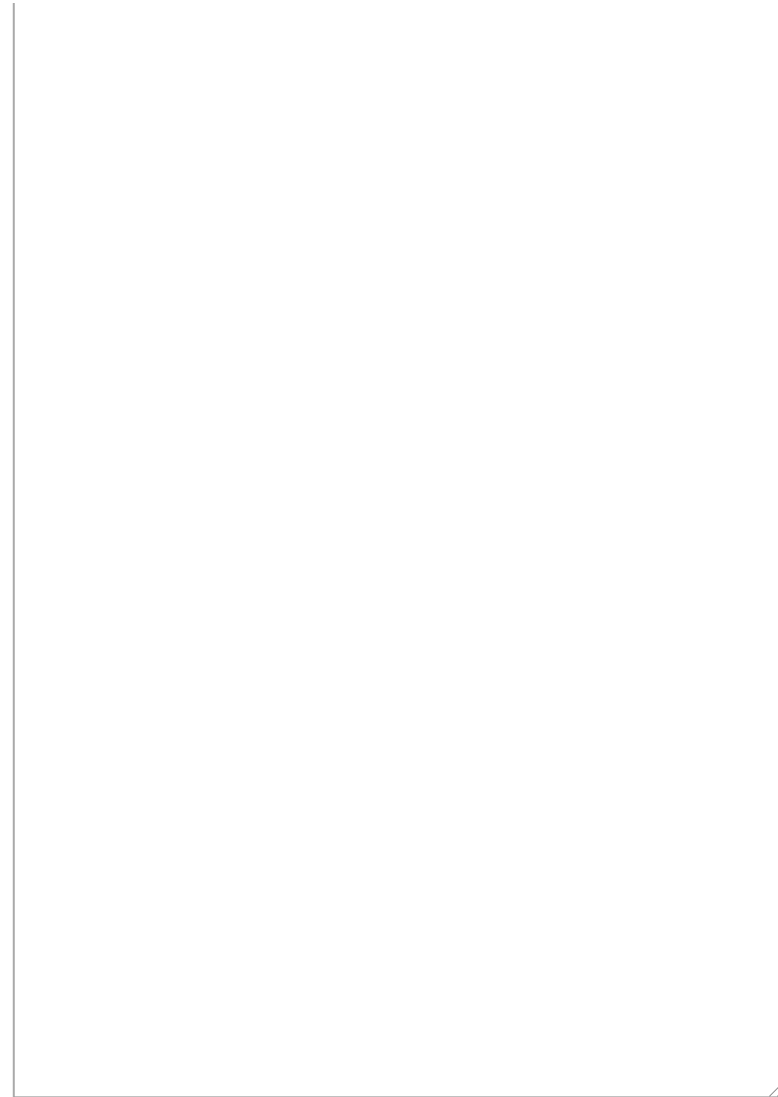
Answer

Word counts: 28

10 user cuối cùng và được sắp xếp theo thứ tự dựa trên  
`first_name`. Vì thứ tự của chuỗi query

-----

Error. Vì hàm `last` cho ra array, không phải relation



Dòng này tương đương với câu sql nào?  
Post.where(published: true).where(closed: false)

27

**Your Answer:** SELECT posts.\* FROM posts WHERE published  
= 1 AND closed = 0

**Correct Options:** SELECT posts.\* FROM posts WHERE  
published = 1 AND closed = 0

Phân biệt left join và inner join



28

Answer

Word counts: 58

Inner Join: Dữ liệu ở trường được join sẽ chứa những dòng  
có ở cả hai bảng

Left Join: Lấy tất cả dữ liệu của bảng bên trái và các giá trị ở  
trường được join tồn tại ở bảng bên phải

Hàm includes với hàm joins khác nhau chỗ nào?



29

Answer

Word counts: 10

includes dùng outer join  
joins dùng inner join

ý nghĩa của hàm .references và cho ví dụ



30

Answer

Word counts: 25

Chỉ dùng với includes giúp cho include tạo các query theo điều kiện của bảng reference



N + 1 query là gì? làm thế nào để tránh?



Answer

Word counts: 89

N + 1 query là khi cần load data của một model trong quan hệ parent-child hoặc một-nhiều. Truy vấn sẽ được tìm đến record cha sau đó mới thực hiện từng truy vấn với record con => Tạo ra các truy vấn không cần thiết gây giảm hiệu suất, tràn bộ nhớ...

Có thể sử dụng eager load để load cùng lúc các data

rewhere khác với unscoped ntn?



Answer

Word counts: 20

rewhere override lên một điều kiện where  
unscoped hủy luôn điều kiện



Powered by Savsoft Quiz (<https://savsoftquiz.com>)