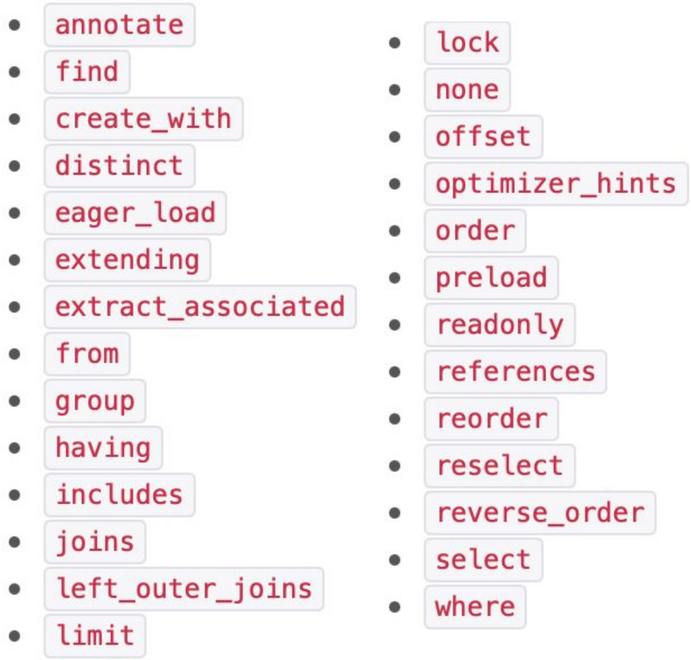
Написание запросов с Active Record

получение записи, получение коллекций,
выбор по условию, сортировка,
группировка, соединение таблиц

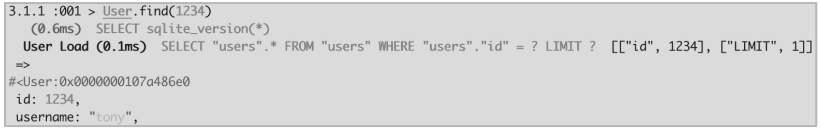
Интерфейс запросов Active Record

Помимо того, что Active Record позволяет настроить подключение к базе данных, он обладает интерфейсом для удобного написания запросов к базе данных. Это позволяет избежать записи SQL запросов вручную и учитывать детали каждой БД.

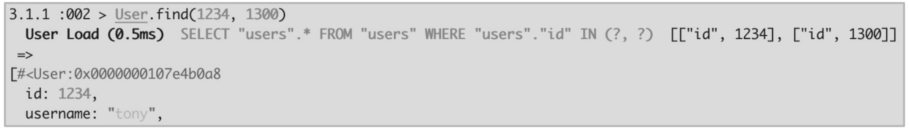


- `annotate`
- `find`
- `create_with`
- `distinct`
- `eager_load`
- `extending`
- `extract_associated`
- `from`
- `group`
- `having`
- `includes`
- `joins`
- `left_outer_joins`
- `limit`
- `lock`
- `none`
- `offset`
- `optimizer_hints`
- `order`
- `preload`
- `readonly`
- `references`
- `reorder`
- `reselect`
- `reverse_order`
- `select`
- `where`

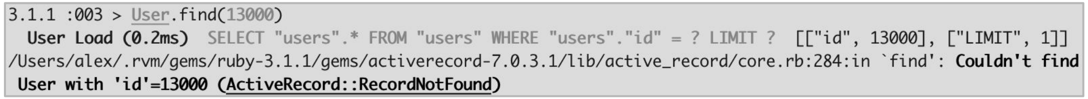
Получение записей из таблицы



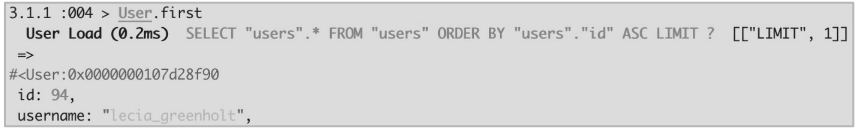
```
3.1.1 :001 > User.find(1234)
   (0.6ms)  SELECT sqlite_version(*)
  User Load (0.1ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ?  [["id", 1234], ["LIMIT", 1]]
 =>
#<User:0x0000000107a486e0
 id: 1234,
 username: "tony",
```



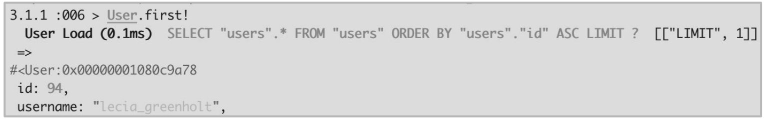
```
3.1.1 :002 > User.find(1234, 1300)
  User Load (0.5ms)  SELECT "users".* FROM "users" WHERE "users"."id" IN (?, ?)  [["id", 1234], ["id", 1300]]
 =>
[#<User:0x0000000107e4b0a8
  id: 1234,
  username: "tony",
```



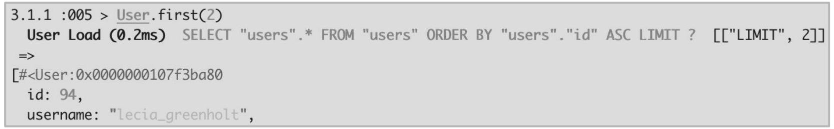
```
3.1.1 :003 > User.find(13000)
  User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ?  [["id", 13000], ["LIMIT", 1]]
/Users/alex/.rvm/gems/ruby-3.1.1/gems/activerecord-7.0.3.1/lib/active_record/core.rb:284:in `find': Couldn't find
 User with 'id'=13000 (ActiveRecord::RecordNotFound)
```



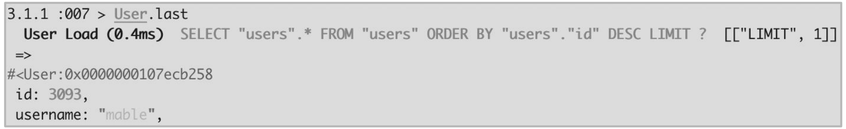
```
3.1.1 :004 > User.first
  User Load (0.2ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ?  [["LIMIT", 1]]
 =>
#<User:0x0000000107d28f90
 id: 94,
 username: "lecia_greenholt",
```



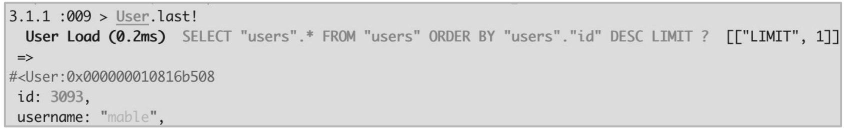
```
3.1.1 :006 > User.first!
  User Load (0.1ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ?  [["LIMIT", 1]]
 =>
#<User:0x00000001080c9a78
 id: 94,
 username: "lecia_greenholt",
```



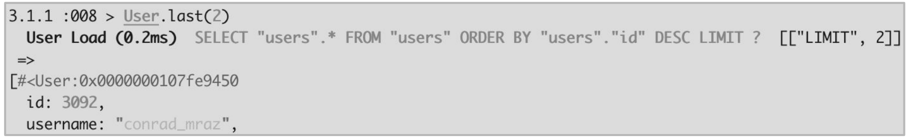
```
3.1.1 :005 > User.first(2)
  User Load (0.2ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ?  [["LIMIT", 2]]
 =>
[#<User:0x0000000107f3ba80
  id: 94,
  username: "lecia_greenholt",
```



```
3.1.1 :007 > User.last
  User Load (0.4ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" DESC LIMIT ?  [["LIMIT", 1]]
 =>
#<User:0x0000000107ecb258
 id: 3093,
 username: "mable",
```

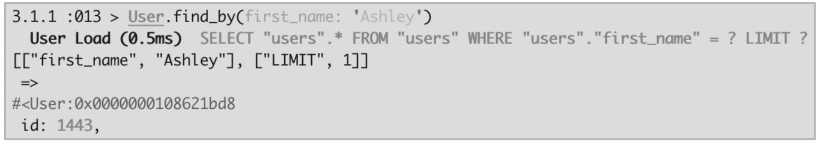


```
3.1.1 :009 > User.last!
  User Load (0.2ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" DESC LIMIT ?  [["LIMIT", 1]]
 =>
#<User:0x000000010816b508
 id: 3093,
 username: "mable",
```

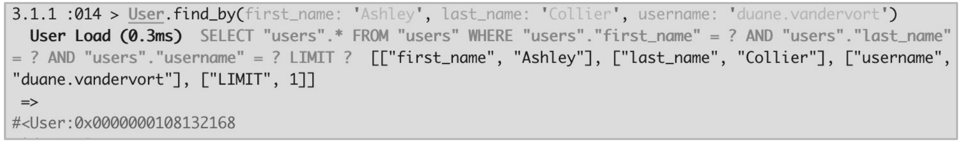


```
3.1.1 :008 > User.last(2)
  User Load (0.2ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" DESC LIMIT ?  [["LIMIT", 2]]
 =>
[#<User:0x0000000107fe9450
  id: 3092,
  username: "conrad_mraz",
```

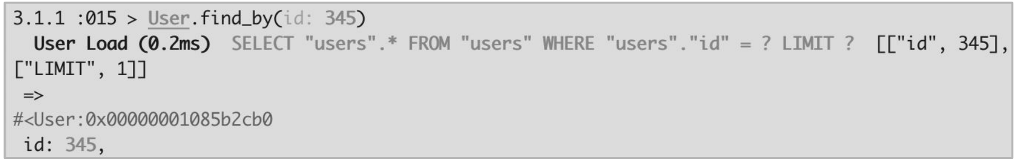
Поиск записи по условию



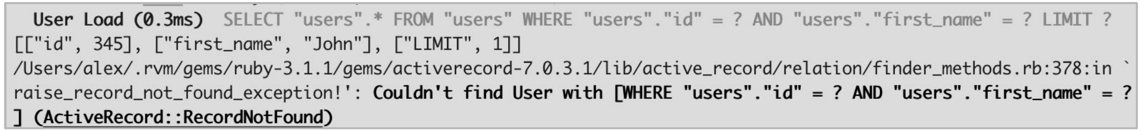
```
3.1.1 :013 > User.find_by(first_name: 'Ashley')
  User Load (0.5ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ? LIMIT ?
[["first_name", "Ashley"], ["LIMIT", 1]]
 =>
#<User:0x0000000108621bd8
 id: 1443,
```



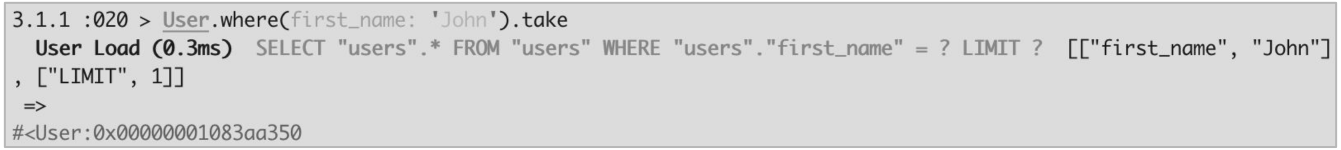
```
3.1.1 :014 > User.find_by(first_name: 'Ashley', last_name: 'Collier', username: 'duane.vandervort')
  User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ? AND "users"."last_name"
= ? AND "users"."username" = ? LIMIT ?  [["first_name", "Ashley"], ["last_name", "Collier"], ["username",
"duane.vandervort"], ["LIMIT", 1]]
 =>
#<User:0x0000000108132168
```



```
3.1.1 :015 > User.find_by(id: 345)
  User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ?  [["id", 345],
["LIMIT", 1]]
 =>
#<User:0x00000001085b2cb0
 id: 345,
```

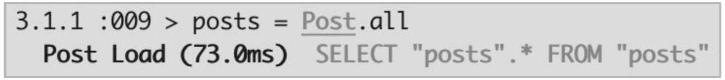


```
  User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? AND "users"."first_name" = ? LIMIT ?
[["id", 345], ["first_name", "John"], ["LIMIT", 1]]
/Users/alex/.rvm/gems/ruby-3.1.1/gems/activerecord-7.0.3.1/lib/active_record/relation/finder_methods.rb:378:in `
raise_record_not_found_exception!': Couldn't find User with [WHERE "users"."id" = ? AND "users"."first_name" = ?
] (ActiveRecord::RecordNotFound)
```

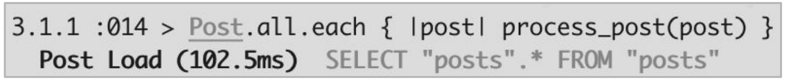


```
3.1.1 :020 > User.where(first_name: 'John').take
  User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ? LIMIT ?  [["first_name", "John"]
, ["LIMIT", 1]]
 =>
#<User:0x00000001083aa350
```

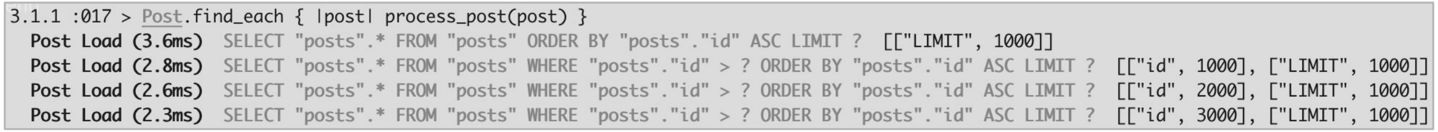
Получение коллекции записей



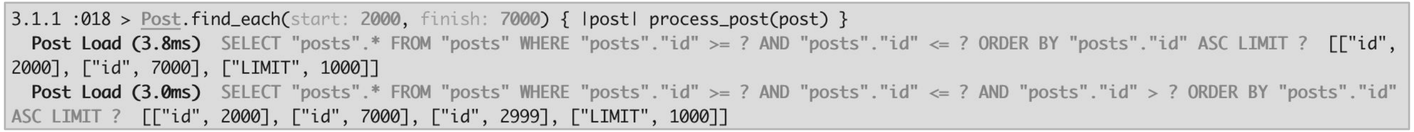
```
3.1.1 :009 > posts = Post.all
  Post Load (73.0ms)  SELECT "posts".* FROM "posts"
```



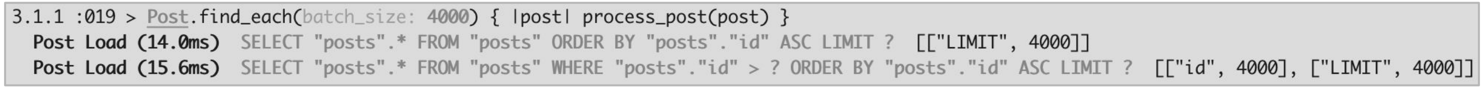
```
3.1.1 :014 > Post.all.each { |post| process_post(post) }
  Post Load (102.5ms)  SELECT "posts".* FROM "posts"
```



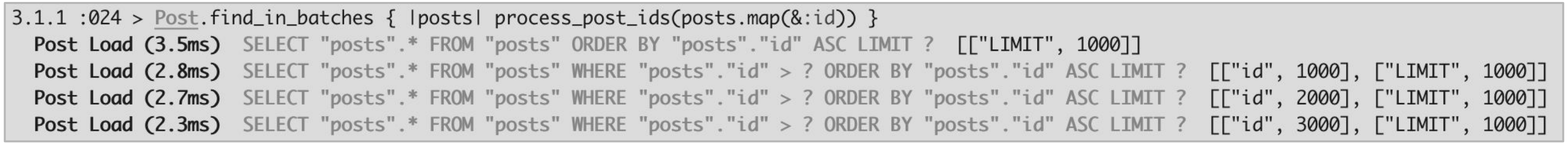
```
3.1.1 :017 > Post.find_each { |post| process_post(post) }
  Post Load (3.6ms)  SELECT "posts".* FROM "posts" ORDER BY "posts"."id" ASC LIMIT ?  [["LIMIT", 1000]]
  Post Load (2.8ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 1000], ["LIMIT", 1000]]
  Post Load (2.6ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 2000], ["LIMIT", 1000]]
  Post Load (2.3ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 3000], ["LIMIT", 1000]]
```



```
3.1.1 :018 > Post.find_each(start: 2000, finish: 7000) { |post| process_post(post) }
  Post Load (3.8ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" >= ? AND "posts"."id" <= ? ORDER BY "posts"."id" ASC LIMIT ?  [["id",
2000], ["id", 7000], ["LIMIT", 1000]]
  Post Load (3.0ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" >= ? AND "posts"."id" <= ? AND "posts"."id" > ? ORDER BY "posts"."id"
ASC LIMIT ?  [["id", 2000], ["id", 7000], ["id", 2999], ["LIMIT", 1000]]
```

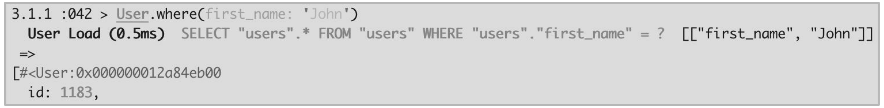


```
3.1.1 :019 > Post.find_each(batch_size: 4000) { |post| process_post(post) }
  Post Load (14.0ms)  SELECT "posts".* FROM "posts" ORDER BY "posts"."id" ASC LIMIT ?  [["LIMIT", 4000]]
  Post Load (15.6ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 4000], ["LIMIT", 4000]]
```

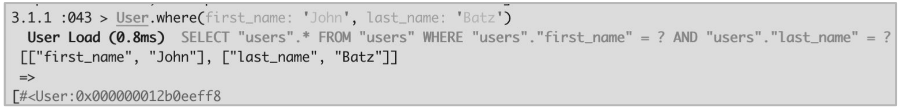


```
3.1.1 :024 > Post.find_in_batches { |posts| process_post_ids(posts.map(&:id)) }
  Post Load (3.5ms)  SELECT "posts".* FROM "posts" ORDER BY "posts"."id" ASC LIMIT ?  [["LIMIT", 1000]]
  Post Load (2.8ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 1000], ["LIMIT", 1000]]
  Post Load (2.7ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 2000], ["LIMIT", 1000]]
  Post Load (2.3ms)  SELECT "posts".* FROM "posts" WHERE "posts"."id" > ? ORDER BY "posts"."id" ASC LIMIT ?  [["id", 3000], ["LIMIT", 1000]]
```

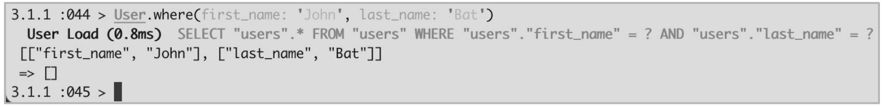
Выбор объектов по условию



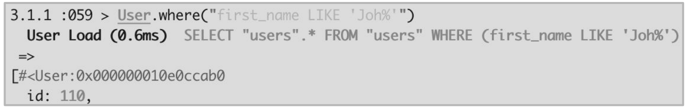
```
3.1.1 :042 > User.where(first_name: 'John')
  User Load (0.5ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ?  [["first_name", "John"]]
 =>
[#<User:0x000000012a84eb00
  id: 1183,
```



```
3.1.1 :043 > User.where(first_name: 'John', last_name: 'Batz')
  User Load (0.8ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ? AND "users"."last_name" = ?
 [["first_name", "John"], ["last_name", "Batz"]]
 =>
[#<User:0x000000012b0eeff8
```

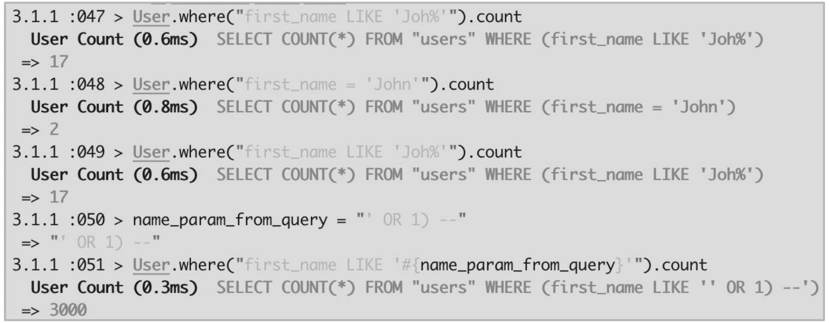


```
3.1.1 :044 > User.where(first_name: 'John', last_name: 'Bat')
  User Load (0.8ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ? AND "users"."last_name" = ?
 [["first_name", "John"], ["last_name", "Bat"]]
 => []
3.1.1 :045 >
```

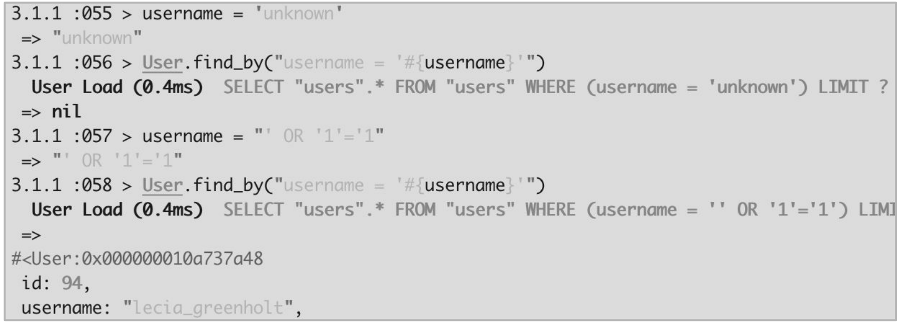


```
3.1.1 :059 > User.where("first_name LIKE 'Joh%'")
  User Load (0.6ms)  SELECT "users".* FROM "users" WHERE (first_name LIKE 'Joh%')
 =>
[#<User:0x000000010e0ccab0
  id: 110,
```

Добавление в метод выбора части запроса, созданного с помощью интерполяции, опасно, так как есть вероятность выполнения SQL инъекции.

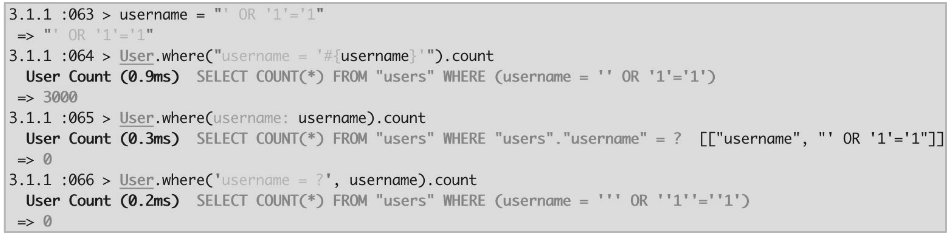


```
3.1.1 :047 > User.where("first_name LIKE 'Joh%'").count
  User Count (0.6ms)  SELECT COUNT(*) FROM "users" WHERE (first_name LIKE 'Joh%')
 => 17
3.1.1 :048 > User.where("first_name = 'John'").count
  User Count (0.8ms)  SELECT COUNT(*) FROM "users" WHERE (first_name = 'John')
 => 2
3.1.1 :049 > User.where("first_name LIKE 'Joh%'").count
  User Count (0.6ms)  SELECT COUNT(*) FROM "users" WHERE (first_name LIKE 'Joh%')
 => 17
3.1.1 :050 > name_param_from_query = "' OR 1) --"
 => "' OR 1) --"
3.1.1 :051 > User.where("first_name LIKE '#{name_param_from_query}'").count
  User Count (0.3ms)  SELECT COUNT(*) FROM "users" WHERE (first_name LIKE '' OR 1) --')
 => 3000
```

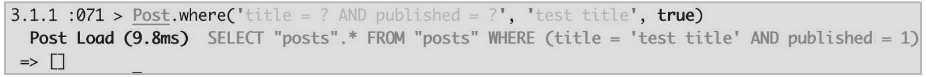


```
3.1.1 :055 > username = 'unknown'
 => "unknown"
3.1.1 :056 > User.find_by("username = '#{username}'")
  User Load (0.4ms)  SELECT "users".* FROM "users" WHERE (username = 'unknown') LIMIT ?
 => nil
3.1.1 :057 > username = "' OR '1'='1"
 => "' OR '1'='1"
3.1.1 :058 > User.find_by("username = '#{username}'")
  User Load (0.4ms)  SELECT "users".* FROM "users" WHERE (username = '' OR '1'='1') LIMI
 =>
#<User:0x000000010a737a48
 id: 94,
 username: "lecia_greenholt",
```

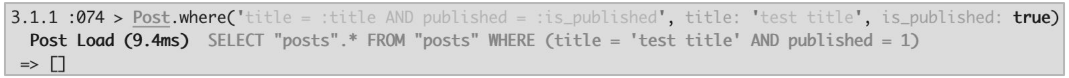
Экранирование параметров



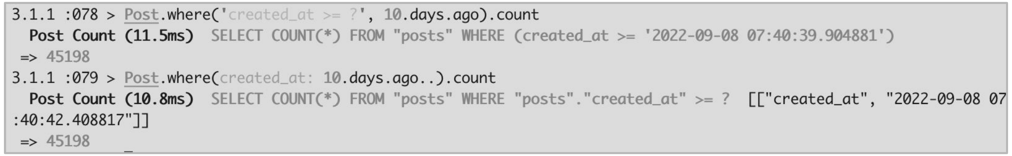
```
3.1.1 :063 > username = "' OR '1'='1"
 => "' OR '1'='1"
3.1.1 :064 > User.where("username = '#{username}'").count
  User Count (0.9ms)  SELECT COUNT(*) FROM "users" WHERE (username = '' OR '1'='1')
 => 3000
3.1.1 :065 > User.where(username: username).count
  User Count (0.3ms)  SELECT COUNT(*) FROM "users" WHERE "users"."username" = ?  [["username", "' OR '1'='1"]]
 => 0
3.1.1 :066 > User.where('username = ?', username).count
  User Count (0.2ms)  SELECT COUNT(*) FROM "users" WHERE (username = ''' OR ''1''=''1')
 => 0
```



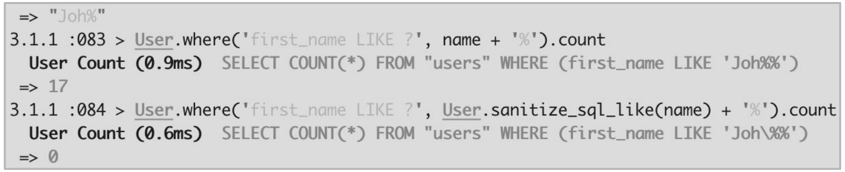
```
3.1.1 :071 > Post.where('title = ? AND published = ?', 'test title', true)
  Post Load (9.8ms)  SELECT "posts".* FROM "posts" WHERE (title = 'test title' AND published = 1)
 => []        _
```



```
3.1.1 :074 > Post.where('title = :title AND published = :is_published', title: 'test title', is_published: true)
  Post Load (9.4ms)  SELECT "posts".* FROM "posts" WHERE (title = 'test title' AND published = 1)
 => []
```

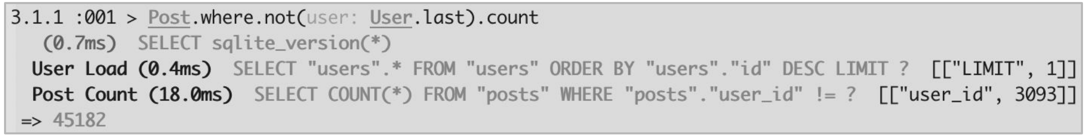


```
3.1.1 :078 > Post.where('created_at >= ?', 10.days.ago).count
  Post Count (11.5ms)  SELECT COUNT(*) FROM "posts" WHERE (created_at >= '2022-09-08 07:40:39.904881')
 => 45198
3.1.1 :079 > Post.where(created_at: 10.days.ago..).count
  Post Count (10.8ms)  SELECT COUNT(*) FROM "posts" WHERE "posts"."created_at" >= ?  [["created_at", "2022-09-08 07
:40:42.408817"]]
 => 45198    _
```

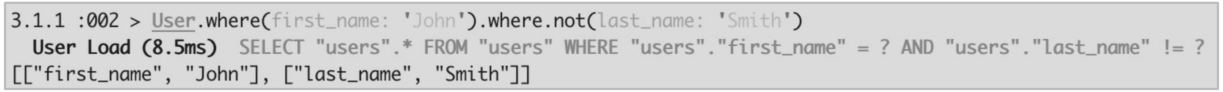


```
 => "Joh%"
3.1.1 :083 > User.where('first_name LIKE ?', name + '%').count
  User Count (0.9ms)  SELECT COUNT(*) FROM "users" WHERE (first_name LIKE 'Joh%%')
 => 17
3.1.1 :084 > User.where('first_name LIKE ?', User.sanitize_sql_like(name) + '%').count
  User Count (0.6ms)  SELECT COUNT(*) FROM "users" WHERE (first_name LIKE 'Joh\%%')
 => 0
```

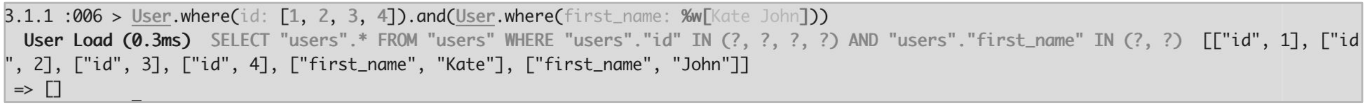
Условия NOT, AND, OR



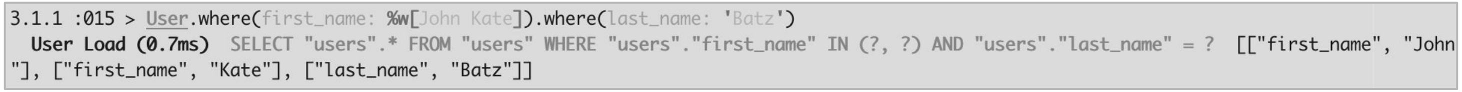
```
3.1.1 :001 > Post.where.not(user: User.last).count
   (0.7ms)  SELECT sqlite_version(*)
  User Load (0.4ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" DESC LIMIT ?  [["LIMIT", 1]]
  Post Count (18.0ms)  SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" != ?  [["user_id", 3093]]
 => 45182
```



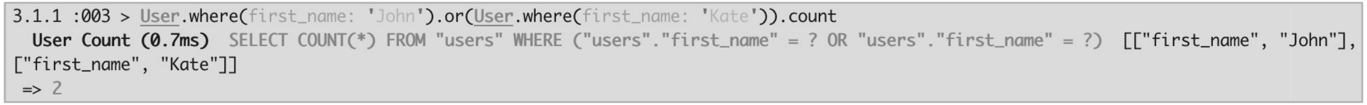
```
3.1.1 :002 > User.where(first_name: 'John').where.not(last_name: 'Smith')
  User Load (8.5ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" = ? AND "users"."last_name" != ?
[["first_name", "John"], ["last_name", "Smith"]]
```



```
3.1.1 :006 > User.where(id: [1, 2, 3, 4]).and(User.where(first_name: %w[Kate John]))
  User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."id" IN (?, ?, ?, ?) AND "users"."first_name" IN (?, ?)  [["id", 1], ["id
", 2], ["id", 3], ["id", 4], ["first_name", "Kate"], ["first_name", "John"]]
 => []
```

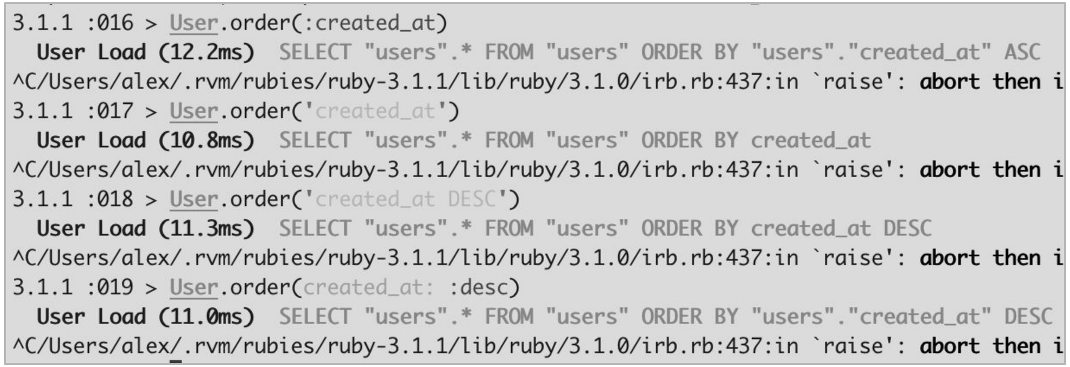


```
3.1.1 :015 > User.where(first_name: %w[John Kate]).where(last_name: 'Batz')
  User Load (0.7ms)  SELECT "users".* FROM "users" WHERE "users"."first_name" IN (?, ?) AND "users"."last_name" = ?  [["first_name", "John
"], ["first_name", "Kate"], ["last_name", "Batz"]]
```

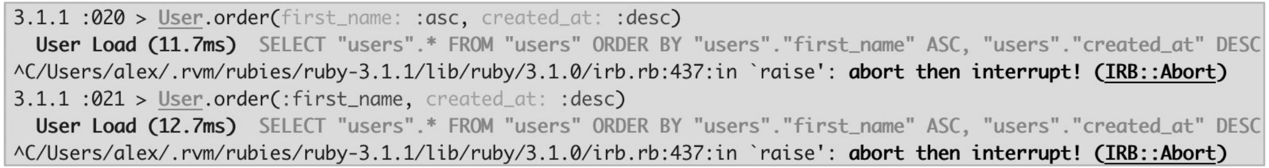


```
3.1.1 :003 > User.where(first_name: 'John').or(User.where(first_name: 'Kate')).count
  User Count (0.7ms)  SELECT COUNT(*) FROM "users" WHERE ("users"."first_name" = ? OR "users"."first_name" = ?)  [["first_name", "John"],
["first_name", "Kate"]]
 => 2
```

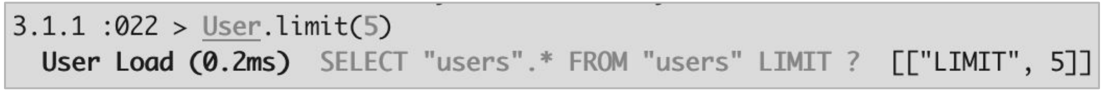
Сортировка и ограничение



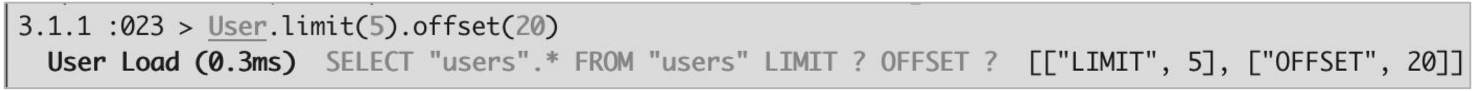
```
3.1.1 :016 > User.order(:created_at)
  User Load (12.2ms)  SELECT "users".* FROM "users" ORDER BY "users"."created_at" ASC
^C/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:437:in `raise': abort then i
3.1.1 :017 > User.order('created_at')
  User Load (10.8ms)  SELECT "users".* FROM "users" ORDER BY created_at
^C/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:437:in `raise': abort then i
3.1.1 :018 > User.order('created_at DESC')
  User Load (11.3ms)  SELECT "users".* FROM "users" ORDER BY created_at DESC
^C/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:437:in `raise': abort then i
3.1.1 :019 > User.order(created_at: :desc)
  User Load (11.0ms)  SELECT "users".* FROM "users" ORDER BY "users"."created_at" DESC
^C/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:437:in `raise': abort then i
```



```
3.1.1 :020 > User.order(first_name: :asc, created_at: :desc)
  User Load (11.7ms)  SELECT "users".* FROM "users" ORDER BY "users"."first_name" ASC, "users"."created_at" DESC
^C/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:437:in `raise': abort then interrupt! (IRB::Abort)
3.1.1 :021 > User.order(:first_name, created_at: :desc)
  User Load (12.7ms)  SELECT "users".* FROM "users" ORDER BY "users"."first_name" ASC, "users"."created_at" DESC
^C/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:437:in `raise': abort then interrupt! (IRB::Abort)
```

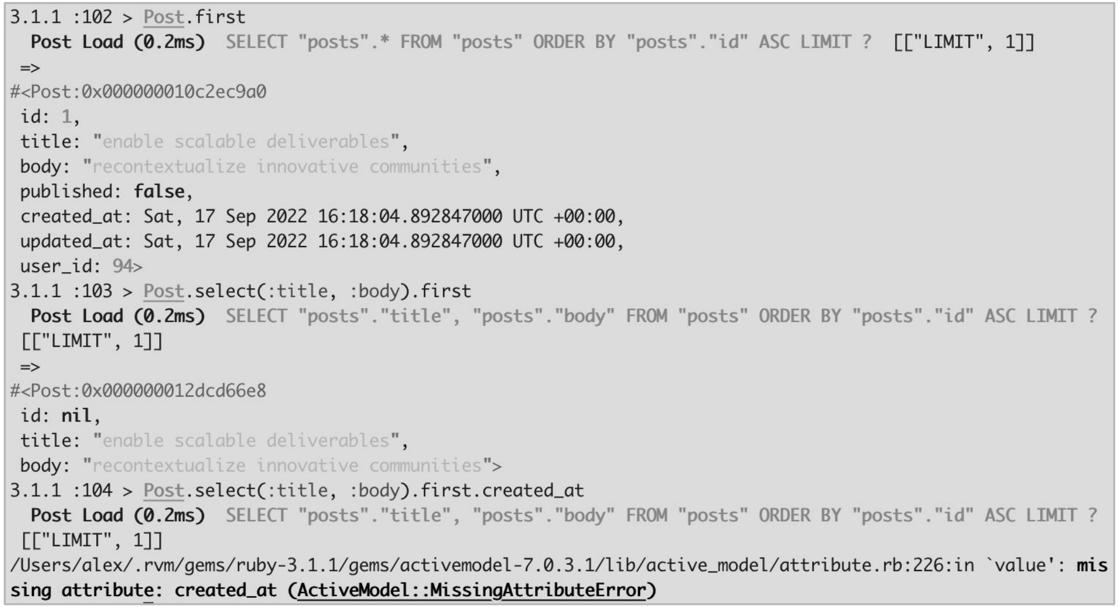


```
3.1.1 :022 > User.limit(5)
  User Load (0.2ms)  SELECT "users".* FROM "users" LIMIT ?  [["LIMIT", 5]]
```

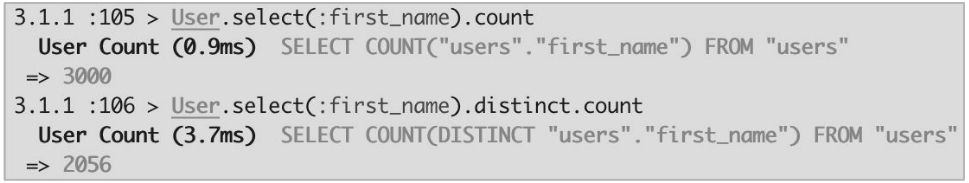


```
3.1.1 :023 > User.limit(5).offset(20)
  User Load (0.3ms)  SELECT "users".* FROM "users" LIMIT ? OFFSET ?  [["LIMIT", 5], ["OFFSET", 20]]
```

Проекция выбранных полей

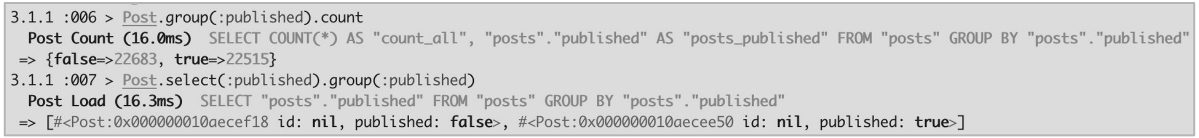


```
3.1.1 :102 > Post.first
  Post Load (0.2ms)  SELECT "posts".* FROM "posts" ORDER BY "posts"."id" ASC LIMIT ?  [["LIMIT", 1]]
 =>
#<Post:0x000000010c2ec9a0
 id: 1,
 title: "enable scalable deliverables",
 body: "recontextualize innovative communities",
 published: false,
 created_at: Sat, 17 Sep 2022 16:18:04.892847000 UTC +00:00,
 updated_at: Sat, 17 Sep 2022 16:18:04.892847000 UTC +00:00,
 user_id: 94>
3.1.1 :103 > Post.select(:title, :body).first
  Post Load (0.2ms)  SELECT "posts"."title", "posts"."body" FROM "posts" ORDER BY "posts"."id" ASC LIMIT ?
 [["LIMIT", 1]]
 =>
#<Post:0x000000012dcd66e8
 id: nil,
 title: "enable scalable deliverables",
 body: "recontextualize innovative communities">
3.1.1 :104 > Post.select(:title, :body).first.created_at
  Post Load (0.2ms)  SELECT "posts"."title", "posts"."body" FROM "posts" ORDER BY "posts"."id" ASC LIMIT ?
 [["LIMIT", 1]]
/Users/alex/.rvm/gems/ruby-3.1.1/gems/activemodel-7.0.3.1/lib/active_model/attribute.rb:226:in `value': mis
sing attribute: created_at (ActiveModel::MissingAttributeError)
```

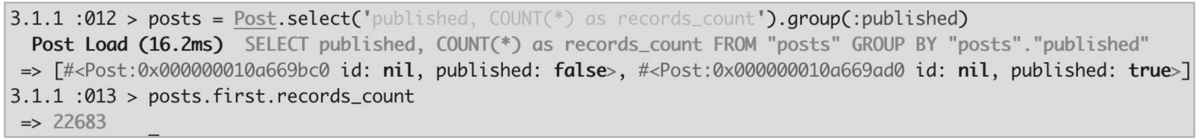


```
3.1.1 :105 > User.select(:first_name).count
  User Count (0.9ms)  SELECT COUNT("users"."first_name") FROM "users"
 => 3000
3.1.1 :106 > User.select(:first_name).distinct.count
  User Count (3.7ms)  SELECT COUNT(DISTINCT "users"."first_name") FROM "users"
 => 2056
```

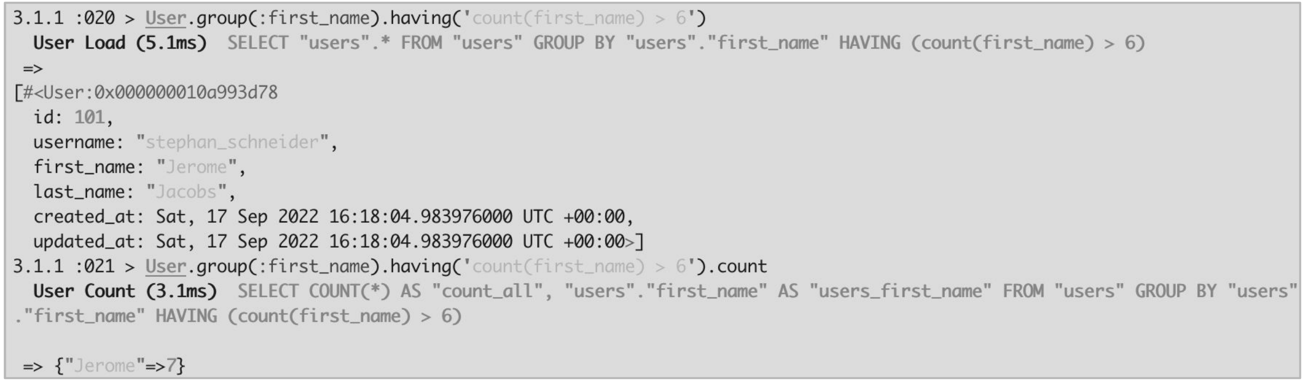
Группировка записей из таблицы



```
3.1.1 :006 > Post.group(:published).count
  Post Count (16.0ms)  SELECT COUNT(*) AS "count_all", "posts"."published" AS "posts_published" FROM "posts" GROUP BY "posts"."published"
 => {false=>22683, true=>22515}
3.1.1 :007 > Post.select(:published).group(:published)
  Post Load (16.3ms)  SELECT "posts"."published" FROM "posts" GROUP BY "posts"."published"
 => [#<Post:0x000000010aecef18 id: nil, published: false>, #<Post:0x000000010aecee50 id: nil, published: true>]
```



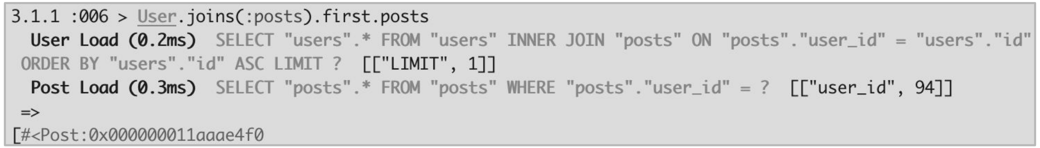
```
3.1.1 :012 > posts = Post.select('published, COUNT(*) as records_count').group(:published)
  Post Load (16.2ms)  SELECT published, COUNT(*) as records_count FROM "posts" GROUP BY "posts"."published"
 => [#<Post:0x000000010a669bc0 id: nil, published: false>, #<Post:0x000000010a669ad0 id: nil, published: true>]
3.1.1 :013 > posts.first.records_count
 => 22683
```



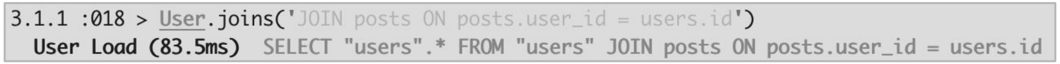
```
3.1.1 :020 > User.group(:first_name).having('count(first_name) > 6')
  User Load (5.1ms)  SELECT "users".* FROM "users" GROUP BY "users"."first_name" HAVING (count(first_name) > 6)
 =>
[#<User:0x000000010a993d78
  id: 101,
  username: "stephan_schneider",
  first_name: "Jerome",
  last_name: "Jacobs",
  created_at: Sat, 17 Sep 2022 16:18:04.983976000 UTC +00:00,
  updated_at: Sat, 17 Sep 2022 16:18:04.983976000 UTC +00:00>]
3.1.1 :021 > User.group(:first_name).having('count(first_name) > 6').count
  User Count (3.1ms)  SELECT COUNT(*) AS "count_all", "users"."first_name" AS "users_first_name" FROM "users" GROUP BY "users"
."first_name" HAVING (count(first_name) > 6)

 => {"Jerome"=>7}
```

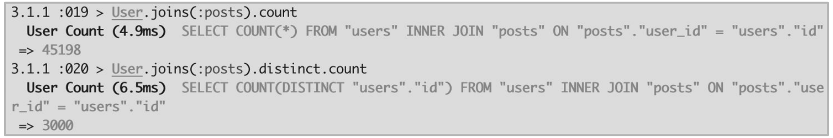
Соединение таблиц



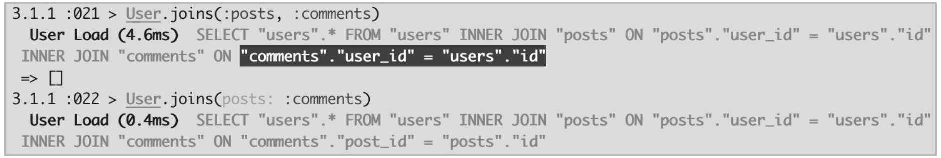
```
3.1.1 :006 > User.joins(:posts).first.posts
  User Load (0.2ms)  SELECT "users".* FROM "users" INNER JOIN "posts" ON "posts"."user_id" = "users"."id"
 ORDER BY "users"."id" ASC LIMIT ?  [["LIMIT", 1]]
  Post Load (0.3ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ?  [["user_id", 94]]
 =>
[#<Post:0x000000011aaae4f0
```



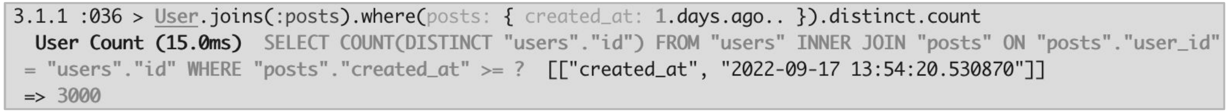
```
3.1.1 :018 > User.joins('JOIN posts ON posts.user_id = users.id')
  User Load (83.5ms)  SELECT "users".* FROM "users" JOIN posts ON posts.user_id = users.id
```



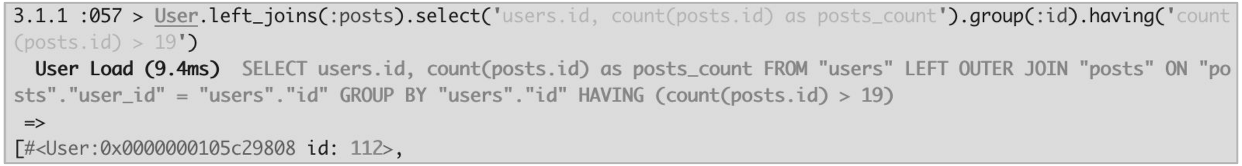
```
3.1.1 :019 > User.joins(:posts).count
  User Count (4.9ms)  SELECT COUNT(*) FROM "users" INNER JOIN "posts" ON "posts"."user_id" = "users"."id"
 => 45198
3.1.1 :020 > User.joins(:posts).distinct.count
  User Count (6.5ms)  SELECT COUNT(DISTINCT "users"."id") FROM "users" INNER JOIN "posts" ON "posts"."use
r_id" = "users"."id"
 => 3000
```



```
3.1.1 :021 > User.joins(:posts, :comments)
  User Load (4.6ms)  SELECT "users".* FROM "users" INNER JOIN "posts" ON "posts"."user_id" = "users"."id"
 INNER JOIN "comments" ON "comments"."user_id" = "users"."id"
 => []
3.1.1 :022 > User.joins(posts: :comments)
  User Load (0.4ms)  SELECT "users".* FROM "users" INNER JOIN "posts" ON "posts"."user_id" = "users"."id"
 INNER JOIN "comments" ON "comments"."post_id" = "posts"."id"
```

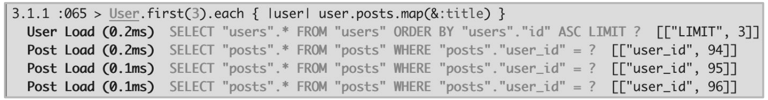


```
3.1.1 :036 > User.joins(:posts).where(posts: { created_at: 1.days.ago.. }).distinct.count
  User Count (15.0ms)  SELECT COUNT(DISTINCT "users"."id") FROM "users" INNER JOIN "posts" ON "posts"."user_id"
 = "users"."id" WHERE "posts"."created_at" >= ?  [["created_at", "2022-09-17 13:54:20.530870"]]
 => 3000
```

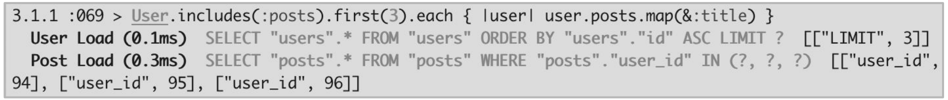


```
3.1.1 :057 > User.left_joins(:posts).select('users.id, count(posts.id) as posts_count').group(:id).having('count
(posts.id) > 19')
  User Load (9.4ms)  SELECT users.id, count(posts.id) as posts_count FROM "users" LEFT OUTER JOIN "posts" ON "po
sts"."user_id" = "users"."id" GROUP BY "users"."id" HAVING (count(posts.id) > 19)
 =>
[#<User:0x0000000105c29808 id: 112>,
```

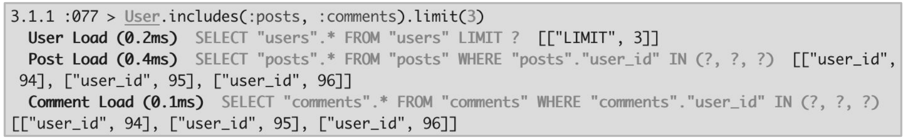
Подгрузка записей. Проблема N+1



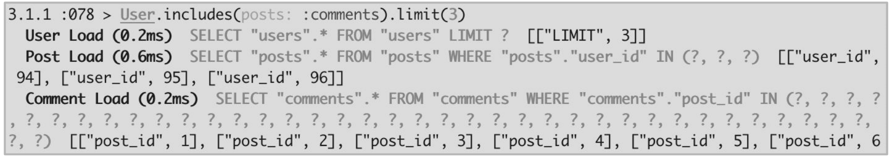
```
3.1.1 :065 > User.first(3).each { |user| user.posts.map(&:title) }
  User Load (0.2ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ?  [["LIMIT", 3]]
  Post Load (0.2ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ?  [["user_id", 94]]
  Post Load (0.1ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ?  [["user_id", 95]]
  Post Load (0.1ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ?  [["user_id", 96]]
```



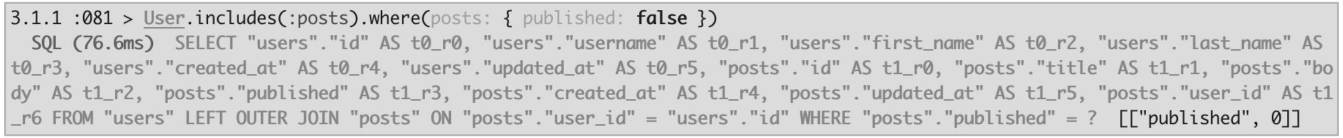
```
3.1.1 :069 > User.includes(:posts).first(3).each { |user| user.posts.map(&:title) }
  User Load (0.1ms)  SELECT "users".* FROM "users" ORDER BY "users"."id" ASC LIMIT ?  [["LIMIT", 3]]
  Post Load (0.3ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" IN (?, ?, ?)  [["user_id",
94], ["user_id", 95], ["user_id", 96]]
```



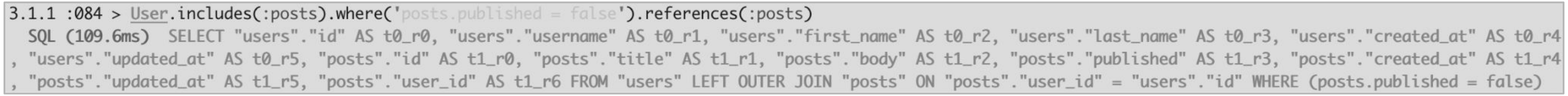
```
3.1.1 :077 > User.includes(:posts, :comments).limit(3)
  User Load (0.2ms)  SELECT "users".* FROM "users" LIMIT ?  [["LIMIT", 3]]
  Post Load (0.4ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" IN (?, ?, ?)  [["user_id",
 94], ["user_id", 95], ["user_id", 96]]
  Comment Load (0.1ms)  SELECT "comments".* FROM "comments" WHERE "comments"."user_id" IN (?, ?, ?)
[["user_id", 94], ["user_id", 95], ["user_id", 96]]
```



```
3.1.1 :078 > User.includes(posts: :comments).limit(3)
  User Load (0.2ms)  SELECT "users".* FROM "users" LIMIT ?  [["LIMIT", 3]]
  Post Load (0.6ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" IN (?, ?, ?)  [["user_id",
 94], ["user_id", 95], ["user_id", 96]]
  Comment Load (0.2ms)  SELECT "comments".* FROM "comments" WHERE "comments"."post_id" IN (?, ?, ?, ?
, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?)  [["post_id", 1], ["post_id", 2], ["post_id", 3], ["post_id", 4], ["post_id", 5], ["post_id", 6
```

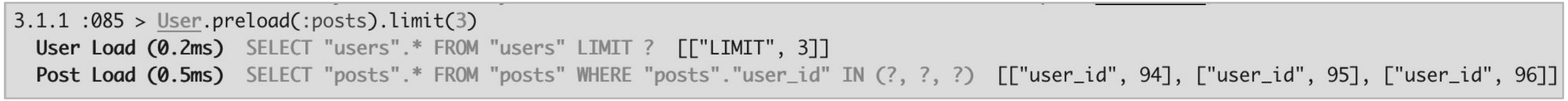


```
3.1.1 :081 > User.includes(:posts).where(posts: { published: false })
  SQL (76.6ms)  SELECT "users"."id" AS t0_r0, "users"."username" AS t0_r1, "users"."first_name" AS t0_r2, "users"."last_name" AS
t0_r3, "users"."created_at" AS t0_r4, "users"."updated_at" AS t0_r5, "posts"."id" AS t1_r0, "posts"."title" AS t1_r1, "posts"."bo
dy" AS t1_r2, "posts"."published" AS t1_r3, "posts"."created_at" AS t1_r4, "posts"."updated_at" AS t1_r5, "posts"."user_id" AS t1
_r6 FROM "users" LEFT OUTER JOIN "posts" ON "posts"."user_id" = "users"."id" WHERE "posts"."published" = ?  [["published", 0]]
```

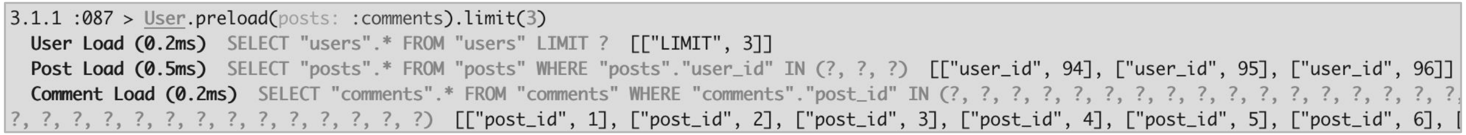


```
3.1.1 :084 > User.includes(:posts).where('posts.published = false').references(:posts)
  SQL (109.6ms)  SELECT "users"."id" AS t0_r0, "users"."username" AS t0_r1, "users"."first_name" AS t0_r2, "users"."last_name" AS t0_r3, "users"."created_at" AS t0_r4
, "users"."updated_at" AS t0_r5, "posts"."id" AS t1_r0, "posts"."title" AS t1_r1, "posts"."body" AS t1_r2, "posts"."published" AS t1_r3, "posts"."created_at" AS t1_r4
, "posts"."updated_at" AS t1_r5, "posts"."user_id" AS t1_r6 FROM "users" LEFT OUTER JOIN "posts" ON "posts"."user_id" = "users"."id" WHERE (posts.published = false)
```

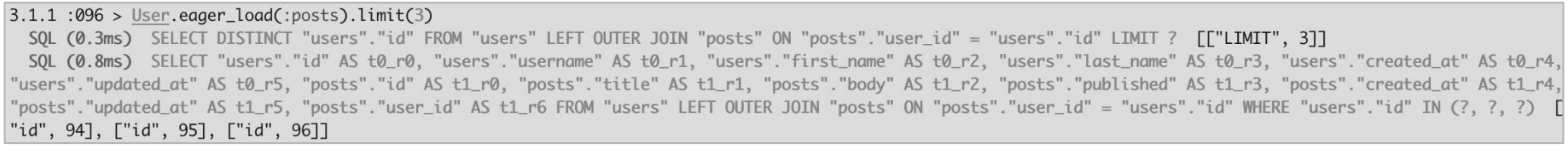
Загрузка с preload и eager_load



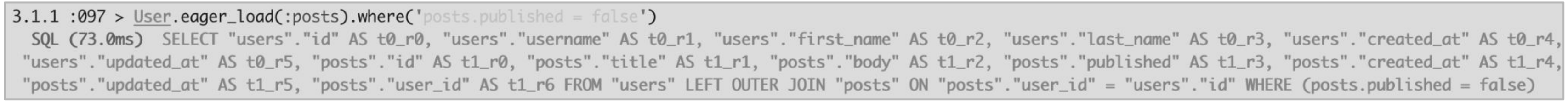
```
3.1.1 :085 > User.preload(:posts).limit(3)
  User Load (0.2ms)  SELECT "users".* FROM "users" LIMIT ?  [["LIMIT", 3]]
  Post Load (0.5ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" IN (?, ?, ?)  [["user_id", 94], ["user_id", 95], ["user_id", 96]]
```



```
3.1.1 :087 > User.preload(posts: :comments).limit(3)
  User Load (0.2ms)  SELECT "users".* FROM "users" LIMIT ?  [["LIMIT", 3]]
  Post Load (0.5ms)  SELECT "posts".* FROM "posts" WHERE "posts"."user_id" IN (?, ?, ?)  [["user_id", 94], ["user_id", 95], ["user_id", 96]]
  Comment Load (0.2ms)  SELECT "comments".* FROM "comments" WHERE "comments"."post_id" IN (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)  [["post_id", 1], ["post_id", 2], ["post_id", 3], ["post_id", 4], ["post_id", 5], ["post_id", 6], [
```

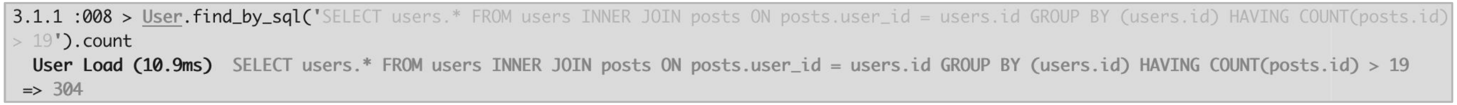


```
3.1.1 :096 > User.eager_load(:posts).limit(3)
  SQL (0.3ms)  SELECT DISTINCT "users"."id" FROM "users" LEFT OUTER JOIN "posts" ON "posts"."user_id" = "users"."id" LIMIT ?  [["LIMIT", 3]]
  SQL (0.8ms)  SELECT "users"."id" AS t0_r0, "users"."username" AS t0_r1, "users"."first_name" AS t0_r2, "users"."last_name" AS t0_r3, "users"."created_at" AS t0_r4,
"users"."updated_at" AS t0_r5, "posts"."id" AS t1_r0, "posts"."title" AS t1_r1, "posts"."body" AS t1_r2, "posts"."published" AS t1_r3, "posts"."created_at" AS t1_r4,
"posts"."updated_at" AS t1_r5, "posts"."user_id" AS t1_r6 FROM "users" LEFT OUTER JOIN "posts" ON "posts"."user_id" = "users"."id" WHERE "users"."id" IN (?, ?, ?)  [
"id", 94], ["id", 95], ["id", 96]]
```

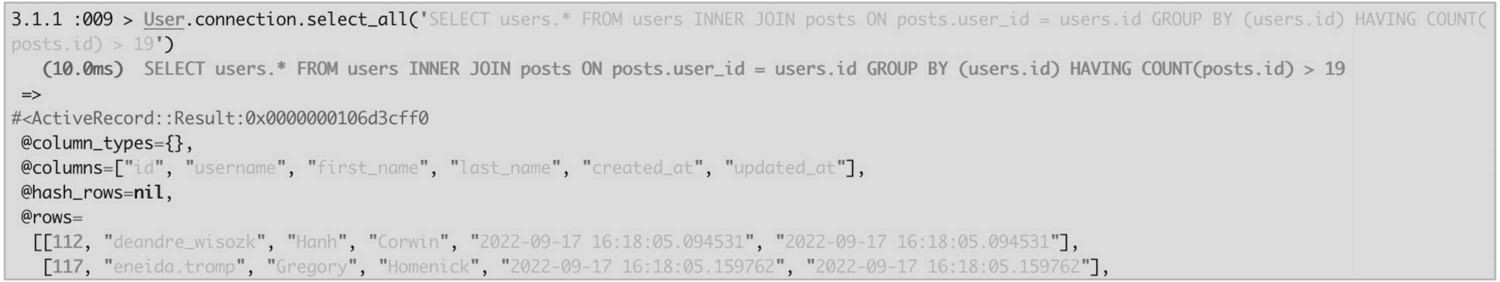


```
3.1.1 :097 > User.eager_load(:posts).where('posts.published = false')
  SQL (73.0ms)  SELECT "users"."id" AS t0_r0, "users"."username" AS t0_r1, "users"."first_name" AS t0_r2, "users"."last_name" AS t0_r3, "users"."created_at" AS t0_r4,
 "users"."updated_at" AS t0_r5, "posts"."id" AS t1_r0, "posts"."title" AS t1_r1, "posts"."body" AS t1_r2, "posts"."published" AS t1_r3, "posts"."created_at" AS t1_r4,
 "posts"."updated_at" AS t1_r5, "posts"."user_id" AS t1_r6 FROM "users" LEFT OUTER JOIN "posts" ON "posts"."user_id" = "users"."id" WHERE (posts.published = false)
```

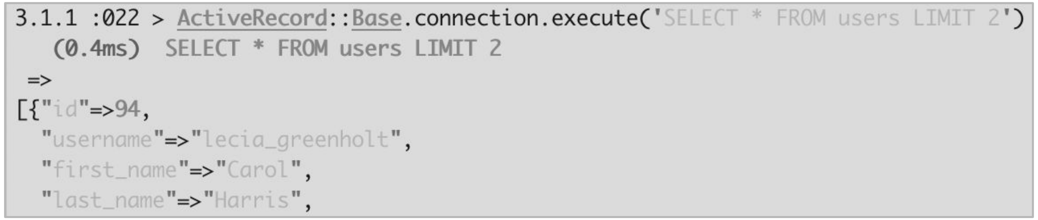
Выполнение запросов на SQL коде



```
3.1.1 :008 > User.find_by_sql('SELECT users.* FROM users INNER JOIN posts ON posts.user_id = users.id GROUP BY (users.id) HAVING COUNT(posts.id)
> 19').count
  User Load (10.9ms)  SELECT users.* FROM users INNER JOIN posts ON posts.user_id = users.id GROUP BY (users.id) HAVING COUNT(posts.id) > 19
 => 304
```

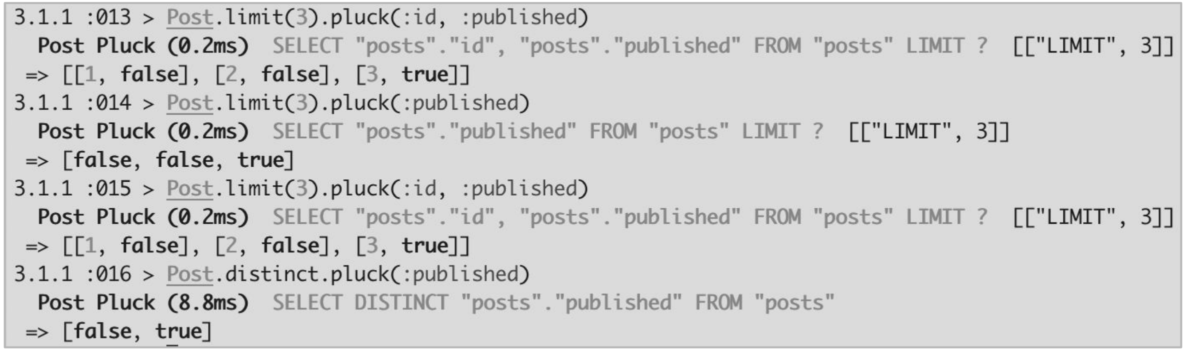


```
3.1.1 :009 > User.connection.select_all('SELECT users.* FROM users INNER JOIN posts ON posts.user_id = users.id GROUP BY (users.id) HAVING COUNT(
posts.id) > 19')
   (10.0ms)  SELECT users.* FROM users INNER JOIN posts ON posts.user_id = users.id GROUP BY (users.id) HAVING COUNT(posts.id) > 19
 =>
#<ActiveRecord::Result:0x0000000106d3cff0
 @column_types={},
 @columns=["id", "username", "first_name", "last_name", "created_at", "updated_at"],
 @hash_rows=nil,
 @rows=
  [[112, "deandre_wisozk", "Hanh", "Corwin", "2022-09-17 16:18:05.094531", "2022-09-17 16:18:05.094531"],
   [117, "eneida.tromp", "Gregory", "Homenick", "2022-09-17 16:18:05.159762", "2022-09-17 16:18:05.159762"],
```

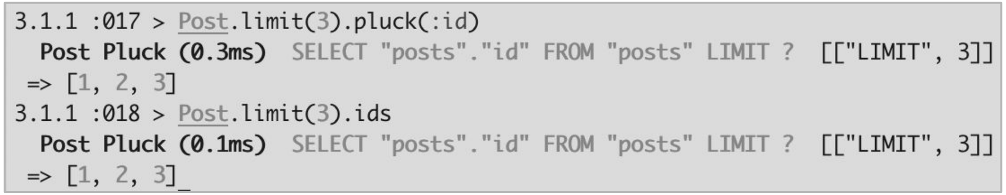


```
3.1.1 :022 > ActiveRecord::Base.connection.execute('SELECT * FROM users LIMIT 2')
   (0.4ms)  SELECT * FROM users LIMIT 2
 =>
[{"id"=>94,
  "username"=>"lecia_greenholt",
  "first_name"=>"Carol",
  "last_name"=>"Harris",
```

Запрос полей без модели

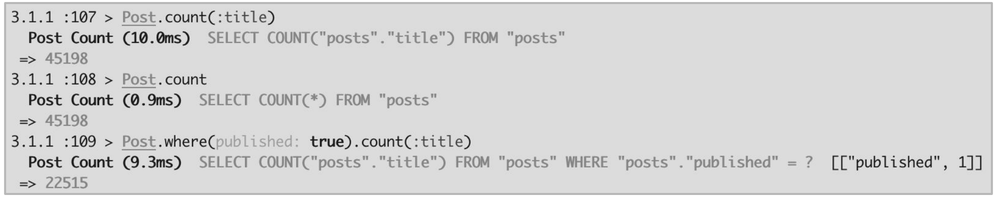


```
3.1.1 :013 > Post.limit(3).pluck(:id, :published)
  Post Pluck (0.2ms)  SELECT "posts"."id", "posts"."published" FROM "posts" LIMIT ?  [["LIMIT", 3]]
 => [[1, false], [2, false], [3, true]]
3.1.1 :014 > Post.limit(3).pluck(:published)
  Post Pluck (0.2ms)  SELECT "posts"."published" FROM "posts" LIMIT ?  [["LIMIT", 3]]
 => [false, false, true]
3.1.1 :015 > Post.limit(3).pluck(:id, :published)
  Post Pluck (0.2ms)  SELECT "posts"."id", "posts"."published" FROM "posts" LIMIT ?  [["LIMIT", 3]]
 => [[1, false], [2, false], [3, true]]
3.1.1 :016 > Post.distinct.pluck(:published)
  Post Pluck (8.8ms)  SELECT DISTINCT "posts"."published" FROM "posts"
 => [false, true]
```

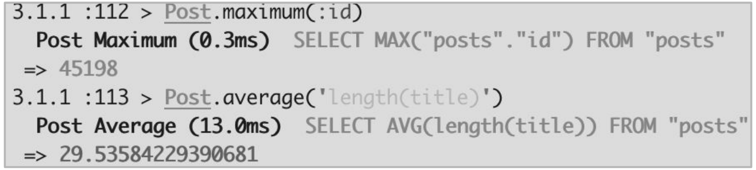


```
3.1.1 :017 > Post.limit(3).pluck(:id)
  Post Pluck (0.3ms)  SELECT "posts"."id" FROM "posts" LIMIT ?  [["LIMIT", 3]]
 => [1, 2, 3]
3.1.1 :018 > Post.limit(3).ids
  Post Pluck (0.1ms)  SELECT "posts"."id" FROM "posts" LIMIT ?  [["LIMIT", 3]]
 => [1, 2, 3]
```

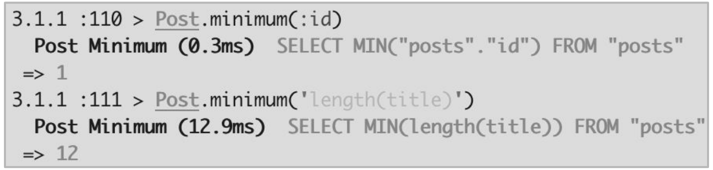
Вычисления и анализ запроса



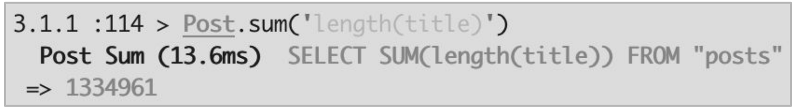
```
3.1.1 :107 > Post.count(:title)
  Post Count (10.0ms)  SELECT COUNT("posts"."title") FROM "posts"
 => 45198
3.1.1 :108 > Post.count
  Post Count (0.9ms)  SELECT COUNT(*) FROM "posts"
 => 45198
3.1.1 :109 > Post.where(published: true).count(:title)
  Post Count (9.3ms)  SELECT COUNT("posts"."title") FROM "posts" WHERE "posts"."published" = ?  [["published", 1]]
 => 22515
```



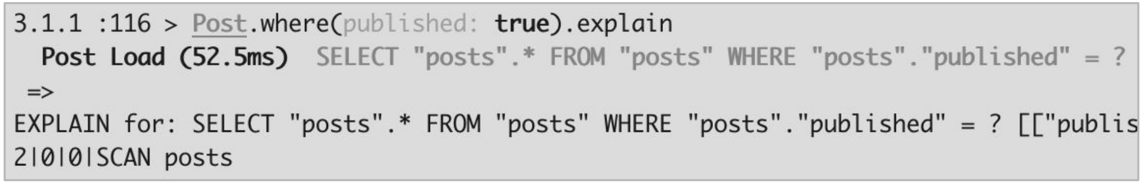
```
3.1.1 :112 > Post.maximum(:id)
  Post Maximum (0.3ms)  SELECT MAX("posts"."id") FROM "posts"
 => 45198
3.1.1 :113 > Post.average('length(title)')
  Post Average (13.0ms)  SELECT AVG(length(title)) FROM "posts"
 => 29.53584229390681
```



```
3.1.1 :110 > Post.minimum(:id)
  Post Minimum (0.3ms)  SELECT MIN("posts"."id") FROM "posts"
 => 1
3.1.1 :111 > Post.minimum('length(title)')
  Post Minimum (12.9ms)  SELECT MIN(length(title)) FROM "posts"
 => 12
```



```
3.1.1 :114 > Post.sum('length(title)')
  Post Sum (13.6ms)  SELECT SUM(length(title)) FROM "posts"
 => 1334961
```



```
3.1.1 :116 > Post.where(published: true).explain
  Post Load (52.5ms)  SELECT "posts".* FROM "posts" WHERE "posts"."published" = ?
 =>
EXPLAIN for: SELECT "posts".* FROM "posts" WHERE "posts"."published" = ? [["publis
2|0|0|SCAN posts
```

Полезные ссылки

http://rusrails.ru/active-record-querying – Обзор интерфейса запросов в Ruby on Rails;

https://dev.to/junko911/rails-n-1-queries-and-eager-loading-10eh – Краткое описание проблемы N+1;

https://dev.to/delbetu/join-vs-includes-vs-eager-load-vs-preload-36l3 – Сравнение методов, позволяющих решить проблему N+1.

Конец! Спасибо!