

Arelを使う

Arelとは

- Rubyで関係代数を扱うためのヤツ
- ActiveRecordを実装するのに使われている
- SQL文の抽象構文木を弄る感じで使う
- 「RailsのプライベートAPIなので使うな」

<https://github.com/rails/rails/issues/16978>

Arelを使うメリット

- ActiveRecordで厳しいクエリを発行できる
- 生SQLを書かずにすむので精神衛生に良い
- ActiveRecordと共存できて、

SQL Injection知らず

```
User.find_by_sql('SELECT SUM(id) FROM users').class
```

```
# => Array
```

```
User.select('SUM(id)').class
```

```
User.select(Arel::Nodes::NamedFunction.new(  
    "SUM",[User.arel_table[:id]]  
)).class
```

```
# => ActiveRecord::Relation::ActiveRecord_Relation_User
```

```
->(x){User.find_by_sql("SELECT SUM(#{x}) FROM users").class}
```

```
->(x){User.select("SUM(#{x})")}
```

```
# Injectionされそう.....
```

```
->(x){User.select(  
    Arel::Nodes::NamedFunction.new(  
        "SUM",[User.arel_table[x]]  
    )  
)}
```

```
# Injection知らず！
```

How to use?

1. 構築したいSQL文を思念する
2. <http://www.rubydoc.info/github/rails/arel/> を開く

3. Arelする

4. ``Model.select(Arelしたやつ)``
``Model.where(Arelしたやつ)`` などする

=> SQL文を直書きする時と同じ感じ

Arelする

```
arel = Receipt.arel_table
```

```
Receipt.select(
```

```
  Arel::Nodes::NamedFunction.new(
```

```
    'AVG',
```

```
    [arel[:amount] + arel[:tax]]
```

```
  )
```

```
)
```

Model.arel_table

Model の Arel::Table を返す

``Arel::Table``

DBのTableを表すオブジェクト

`Arel::Table.new("hoge")[:fuga]`で、
hogeテーブルのfugaカラムを表す

Arel::Nodes

Nodeを作るクラスがいっぱいある

Arel::Nodes

- Arel::Nodes::NamedFunction

``HOGE(FUGA)`` を作る

syntax) Arel::Nodes::NamedFunction.new(func_name, [args])

使用例)

```
x = Arel::Nodes::NamedFunction.new("DATE", [User.arel_table[:id]])
```

```
x.to_sql # => "DATE(`users`.`id`)"
```

Arel::Nodes

- Arel::Nodes::Extract
`EXTRACT(a FROM b)` を作る
- Arel::Nodes::InfixOperation
`a op b` を作る

等々.....

他に使いそうなArelのヤツら

`Arel.star` : `Arel::Nodes::SqlLiteral.new('*')`と同義

`(Arel::Table.new(:table)[:column]).eq`等 'x' : 予想通り

`(Arel::Table.new(:table)[:column]).matches` 'x' : LIKE検索

