



Sequelのすすめ

～私がSQLを嫌いな理由～

とみたまさひろ

RubyHiroba 2013

2013-06-02

自己紹介



- とみた まさひろ
- 長野県北部在住
- プログラマー (**Ruby** & C)
- <http://tmtms.hatenablog.com>
- <http://twitter.com/tmtms>
- 好きなもの
 - **Ruby**, MySQL, Linux Mint, Emacs, Git

OSS貢献者賞





諸君、私はSQLが嫌いだ



```
select name from users  
where loginid = 'tmtms'
```



英語っぽい



どれが予約語やねん



```
select name from users  
where loginid = 'tmtms'
```




予約語が多い



カラム名にgroupが 使えなくて死ぬ

まあクォートすればいいんだけども…



余計な単語



insert into tblname〜

select〜group by col



insert **into** tblname〜

select〜group **by** col



構文解析上なくてもいい



英語っぽくするため(?)

RSpecのようなキモさ



insertとupdateの 構文の非対称



**insert into tbl (col1,col2)
values (val1,val2)**

**update tbl set
col1=val1,col2=val2**



全然違う



MySQLはこれも可能

```
insert tbl set  
col1=val1,col2=val2
```

さすが変態



集約関数



select length(id) from tbl

select count(id) from tbl



select **length**(id) from tbl

→ 結果件数は**レコード数**

select **count**(id) from tbl

→ 結果件数は**1**



同じ構文なのに 関数によって結果が異なる



そんなこんなでSQL嫌い



SQLを書かずにRDBを使うには



Rubyで書けばいいじゃない



そこでSequel

テーブルの操作



SELECT col2 **FROM** **tbl** WHERE col1=1

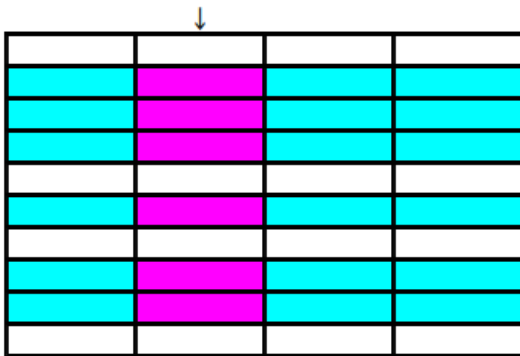
条件で行の絞り込み

SELECT col2 FROM tbl **WHERE col1=1**

				←
				←
				←
				←
				←
				←

その中の列を選択

SELECT col2 FROM tbl WHERE col1=1





Sequelで

Sequel 初期化



```
require 'sequel'
DB = Sequel.connect(
  'mysql://user:passwd@hostname:3306/dbname'
)
```


コマンドラインから



```
% sequel mysql://user:passwd@hostname:3306/dbname  
Your database is stored in DB...  
irb(main):001:0>
```

テーブルの操作



DB[:tbl]

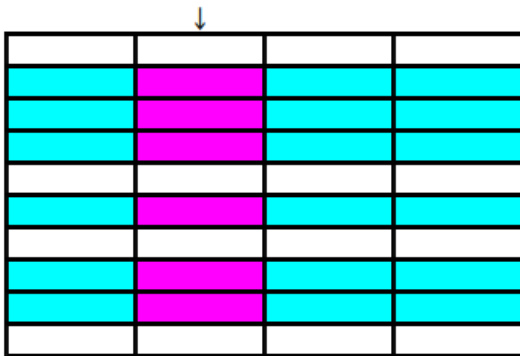
条件で行の絞り込み

DB[:tbl].where(col1: 1)

				←
				←
				←
				←
				←
				←

その中の列を選択

DB[:tbl].where(col1: 1).select(:col2)





とても自然！

Select/Update/Delete/Insert



```
DB[:tbl].where(col1: 1).select(:col2)
```

```
DB[:tbl].where(col1: 1).update(col2: 3)
```

```
DB[:tbl].where(col1: 1).delete
```

```
DB[:tbl].insert(col1: 1, col2: 2)
```

Select は実行遅延



```
ds = DB[:tbl].where(col1: 1).select(:col2)
# まだ実行されない
ds.first
# ここで実行される
#=> {:col2=>'value'}
```

関数

```
DB[:tbl].select(Sequel.function(:length, :col))  
#=> SELECT length(col) FROM tbl
```

```
DB[:tbl].select{length(col)}  
#=> SELECT length(col) FROM tbl
```


集約関数



DB[:tbl].count

#=> SELECT COUNT() FROM tbl*



条件いろいろ

比較

```
where(id: 123)  
#=> WHERE id=123
```

配列



```
where(id: [123, 456])  
#=> WHERE id IN (123, 456)
```

Range

```
where(id: 123..456)  
#=> WHERE id >= 123 AND id <= 456
```

nil, true, false



```
where(id: nil)  
#=> WHERE id is NULL
```

データセット



```
where(id: DB[:tbl2].select(:id))  
#=> WHERE id IN (SELECT id FROM tbl2)
```

正規表現



```
where(id: /abc/)  
#=> WHERE id REGEXP BINARY 'abc'
```

```
where(id: /abc/i)  
#=> WHERE id REGEXP 'abc'
```


複数

```
where(id: 123, name: 'hoge')  
#=> WHERE id=123 AND name='hoge'
```

不等号



```
where(Sequel.expr(:id) > 123)  
#=> WHERE id > 123
```

```
where{id > 123}  
#=> WHERE id > 123
```



JOIN

JOIN



```
DB[:posts].left_join(:users, :id=>:users_id)
#=> SELECT * FROM posts LEFT JOIN users
#    ON users.id=posts.users_id
```

自己結合



```
DB[:hoge].left_join(:hoge.as(:fuga), :x=>:y).  
  select(:hoge__id)  
#=> SELECT hoge.id FROM hoge  
#    LEFT JOIN hoge AS fuga  
#    ON fuga.x=hoge.y
```



FAQ



Q. ActiveRecord じゃだめなん？



A1. Active Record は RDB にデータ置いて読み書きしてるだけで、RDB として使ってるわけじゃないんでちょっと…



A2. ActiveRecord は Ruby/MySQL で動かないし…

そういえば activerecord-ruby_mysql-adapter ってのを
作ったような気もする



**A3. でも Active Record 便利ですよ
ね。Sequel でも使えるよ！**

Sequel::Model



```
class User < Sequel::Model
  plugin :validation_helpers

  def validate
    super
    validates_unique :loginid
    validates_presence :name
  end
end
```

まとめ



- SQL の構文はイケてない(個人の感想です)
- Sequel で Ruby っぽくクエリを書ける
- Sequel は Active Record パターンも使える
- SQL を「シークエル」と発音する人は混乱するがいいさ