



HASHROCKET

EXPERTLY CRAFTED WEB

squealer

MongoDB to SQL

PRESENTED BY

Josh Graham
@delitescere



Codd is Godd



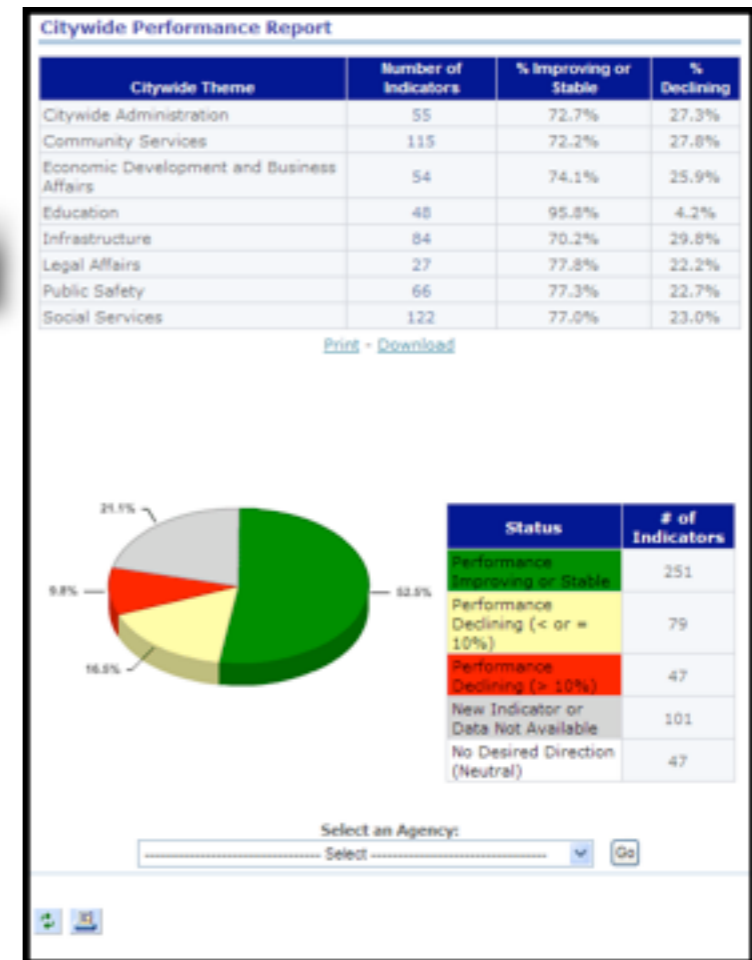
http://upload.wikimedia.org/wikipedia/en/5/58/Edgar_F_Codd.jpg



ORM is in the wrong place!



Don't let this

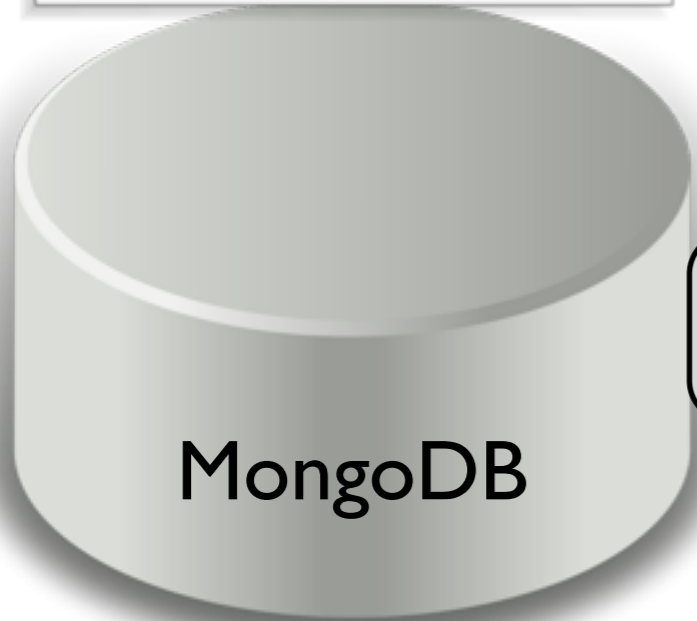
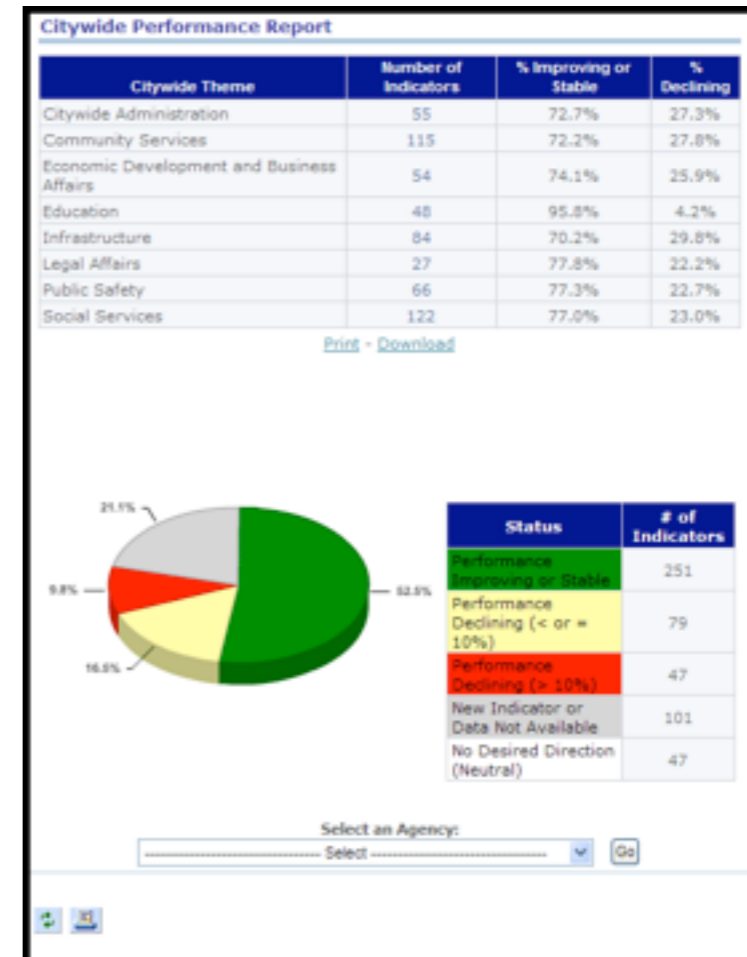


Screw with that





That's better!

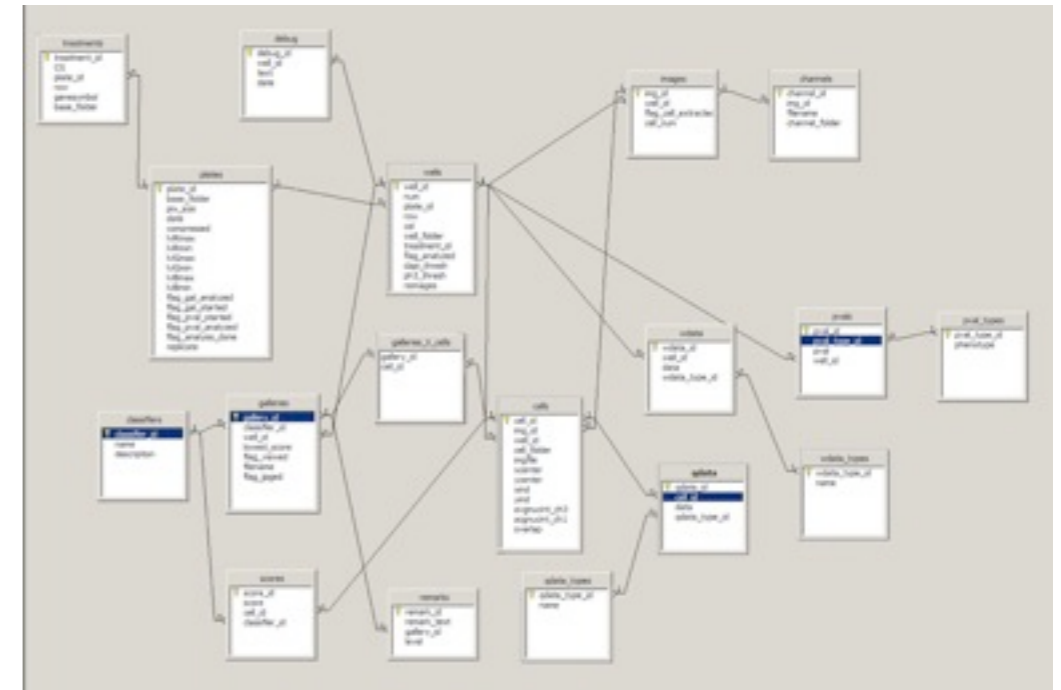
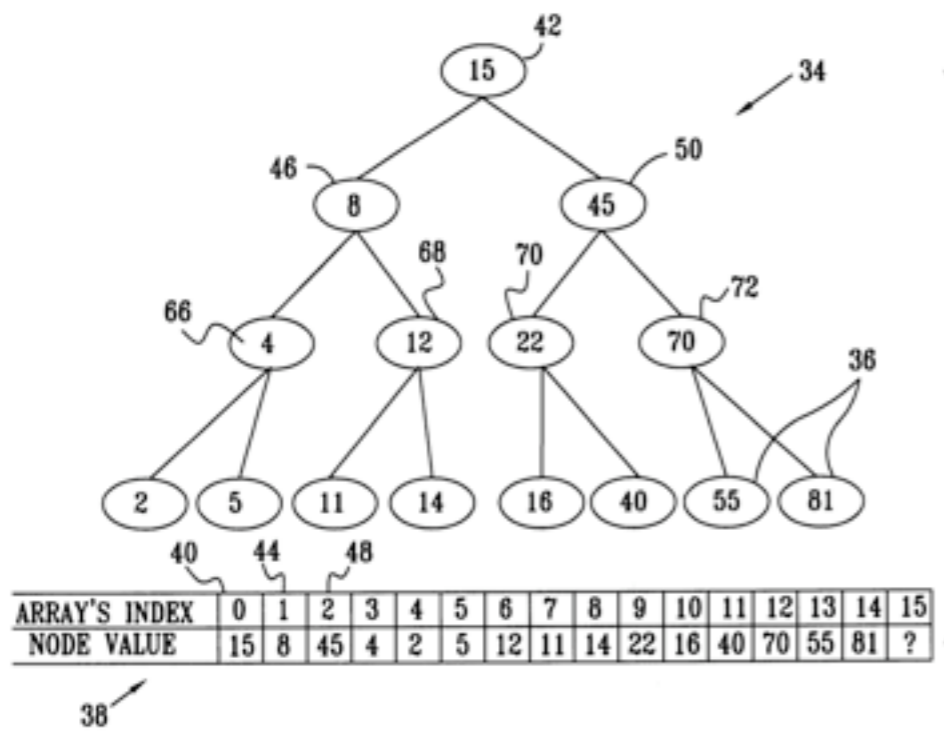


squeal





List of Hashmaps → Relations



<http://www.freepatentsonline.com/6567815-0-large.jpg>

<http://www.mcb.ucdavis.edu/faculty-labs/scholey/roy/tableStructure.jpg>



squealer:

Ruby

command line

DSL

MongoDB

DataObjects

MySQL

PostgreSQL

```
> skewer  
# _squeal.rb  
# _ddl.sql
```



```
1 require 'squealer'
2
3 # connect to the source mongodb database
4 import 'localhost', 27017, 'development'
5
6 # connect to the target mysql database
7 export 'mysql', 'localhost', 'root', '', 'reporting_export'
8
9 import.source('users').each do |user|
10   target(:user) do
11     assign(:name) { "#{user.last_name.upcase}, #{user.first_name}" }
12     assign(:dob) { user.date_of_birth }
13     assign(:gender) #or# assign(:gender) { user.gender }
14   end
15 end
```



```
15 # You can normalize the export...
16 # home_address and work_address are a formatted string like: "661 W Lake St, Suite 3NE, Chicago IL, 60611, USA"
17 addresses = []
18 addresses << atomize_address(user.home_address) # atomize_address is some custom method of yours
19 addresses << atomize_address(user.work_address)
20 addresses.each do |address|
21   target(:address) do
22     assign(:street)
23     assign(:city)
24     assign(:state)
25     assign(:zip)
26   end
27 end

28
29 # You can denormalize the export...
30 # user.home_address = { street: '661 W Lake St', city: 'Chicago', state: 'IL' }
31 assign(:home_address) { flatten_address(user.home_address) } # flatten_address is some custom method of yours
32 assign(:work_address) { flatten_address(user.work_address) }
33
```



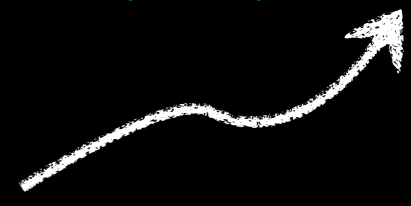
```
33
34 user.activities.each do |activity|
35   target(:activity) do
36     assign(:user_id) #or# assign(:user_id) { user._id }
37     assign(:name)
38     assign(:due_date)
39   end
40
41   activity.tasks.each do |task|
42     target(:task) do
43       assign(:user_id)
44       assign(:activity_id)
45       assign(:due_date)
46     end
47   end #activity.tasks
48 end #user.activities
49 end
50 end #collection("users")
51
52 # Here we use a procedural "join" on related collections to update a target...
53 import.source('organizations', {'disabled_date' => {'exists' => true}}).each do |organization|
54   import.source('users', {'organization_id' => organization.id}) do |user|
55     target(:user) do
56       assign(:disabled) { true }
57     end
58   end
59 end
```



[=====]

] 2/3 (66%)

Coming soon: documents per second



[=====]

] 3/3 (100%)

Start: Fri Jun 25 08:47:37 -0400 2010
End: Fri Jun 25 08:47:48 -0400 2010
Duration: 00:00:11.13617



Restrictions

- Import
 - All (inc. embedded) documents must have `_id`
 - Mongoid for skewer (currently)
- Export
 - Primary key `"id" CHAR(24)`
 - DateTime / Timestamp type for temporal data (MongoDB)
 - No referential integrity
 - No foreign keys
 - No indexes (while loading)



HASHROCKET

EXPERTLY CRAFTED WEB

Thank you

Josh Graham

josh@hashrocket.com

@delitescere

<http://rubygems.org/gems/squealer>