Database Optimization

Scaling Ruby-on-Rails By Example

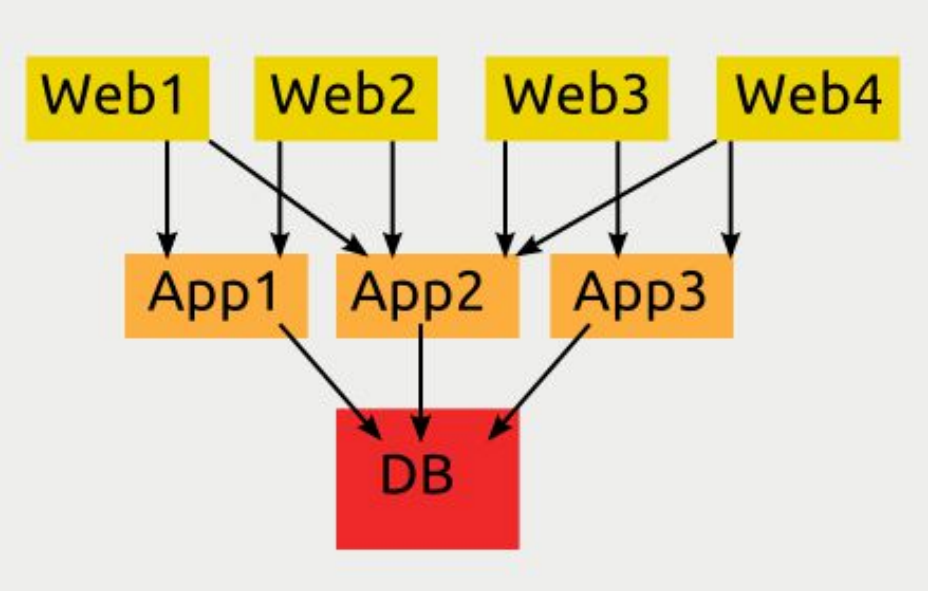
Karsten Meier
meier-online.com

My Technical Background

- 1986: SQL at university
- 1996: QuarkXpress -> HTML Converter
- 1998-2001: WebObjects, MVC, ORM
- 2004: First contact with Ruby (Pleac)
- Since 2005: Handylearn Projects
- Since 2009: Use of Rails in projects

Use Case: Cycosmos

- Community
- Webobjects
- ORM
  Enterprise Objects
- 3 Appserver,
  1 DB Server

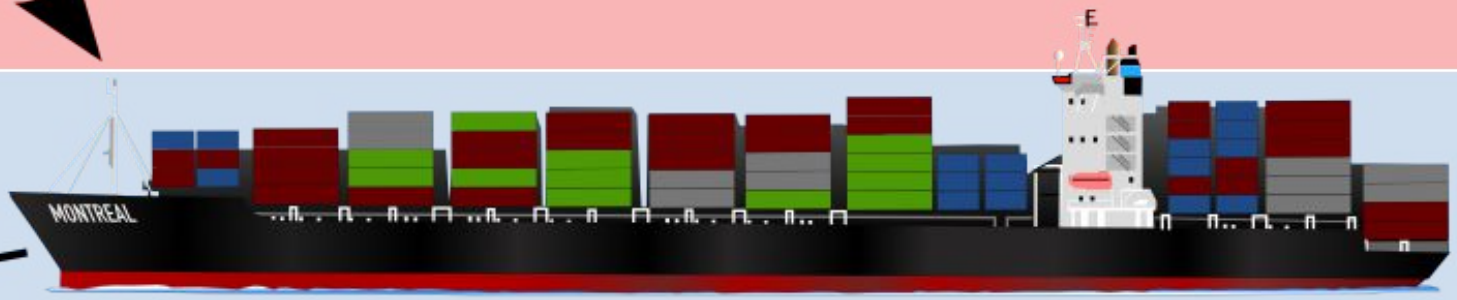


Effects Of Less Database Queries

- Better response times
- Less database load
- 300% higher throughput
- Higher stability

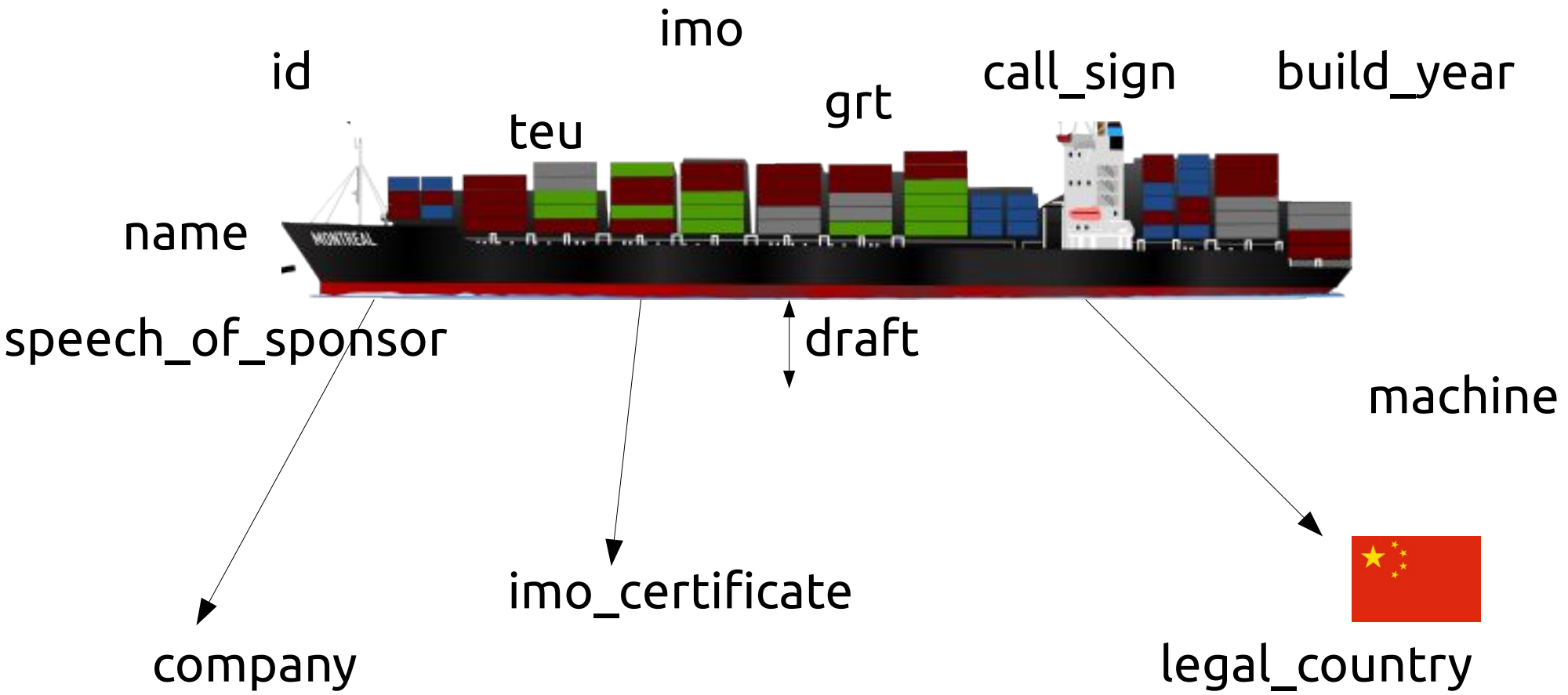
Layers Of A Web Application

id	company_id	name	imo	teu	build_year	legal_country_id
3	2	Bahia	9360752	3630	2007	7
4	2	Cap Jackson	9484560	4600	2010	7
5	2	Annabelle Schulte	9240873	2602	2002	8
6	2	Santa Isabel	9444728	7154	2010	4
7	2	Natalie Schulte	9305881	2474	2005	8
8	2	Monte Olivia	9283198	5560	2004	4
9	3	Houston Express	9294991	8411	2005	4
10	3	Ottawa Express	9165360	2808	1998	9
11	3	Oakland Express	9200811	4890	2000	10
12	1	Ever Unity	9169158	5652	1999	2
13	1	Ever Deluxe	9134256	4211	1997	11
14	1	Ever Unicorn	9196967	5652	2000	11
15	1	Cosco Asia	9345403	10062	2007	2
16	4	HYUNDAI TOGETHER	9473731	13100	2012	7
17	5	Montreal	9253739	4402	2003	5
18	6	CMA CGM TURQUOISE	9386471	4360	2008	7




```
<tr><td>Montreal</td><td>9253739</td>
```

Fat Objects


imo
id
call_sign
build_year
grt
teu
name
speech_of_sponsor
draft
machine
imo_certificate
company
legal_country

Schadow Objects

```
ContainerVessel.
   select('id, name')
   order('name')
```

- Read-Only
- Only given attributes
- Exception if unknown
- ID does not throw exception

`ActiveRecord::ReadOnlyRecord`

`ActiveRecord::MissingAttributeError`

Cherrypicking

- Only one column
- Object not needed
- pluck(column)
- since Rails 3.2

```
ContainerVessel.pluck(:name)
['Australia', 'Brisbane', 'Busan',...]
```

Is vessel ready to cast off?

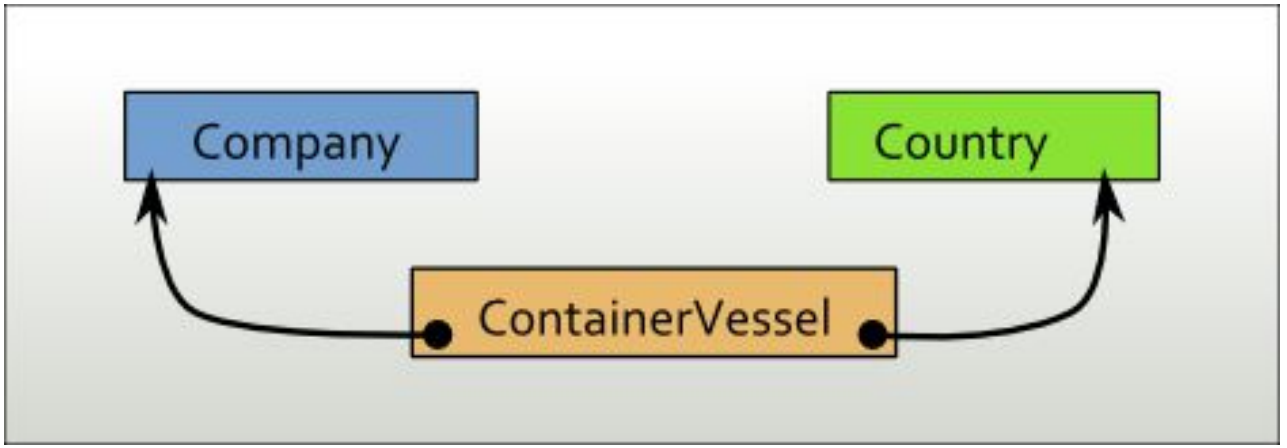
Outsourcing

- Weight of all container on the vessel?
- DB can do the calculation
- Rails does not see the individual containers

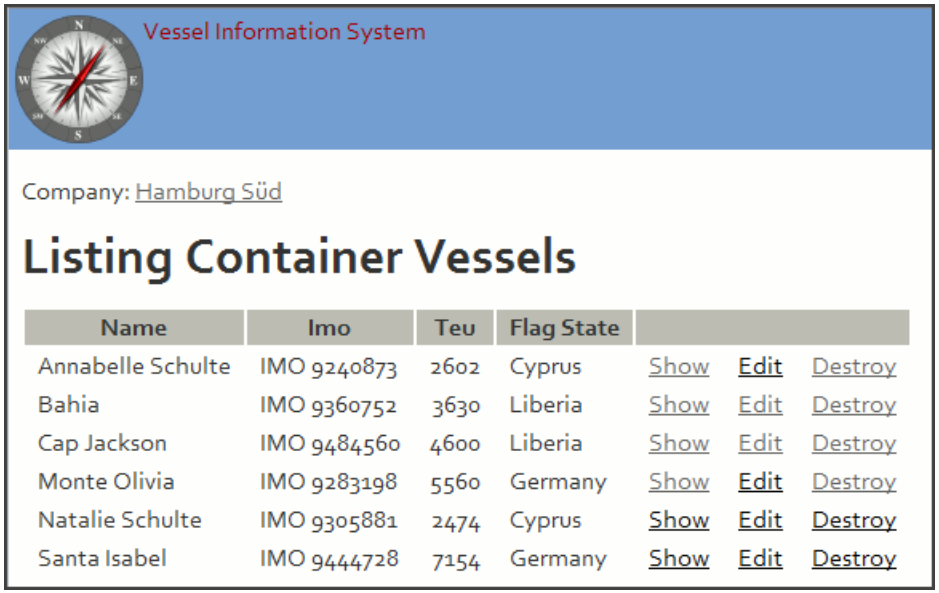
```
@vessel.containers.inject{...}

@vessel.containers.sum('weight')
```

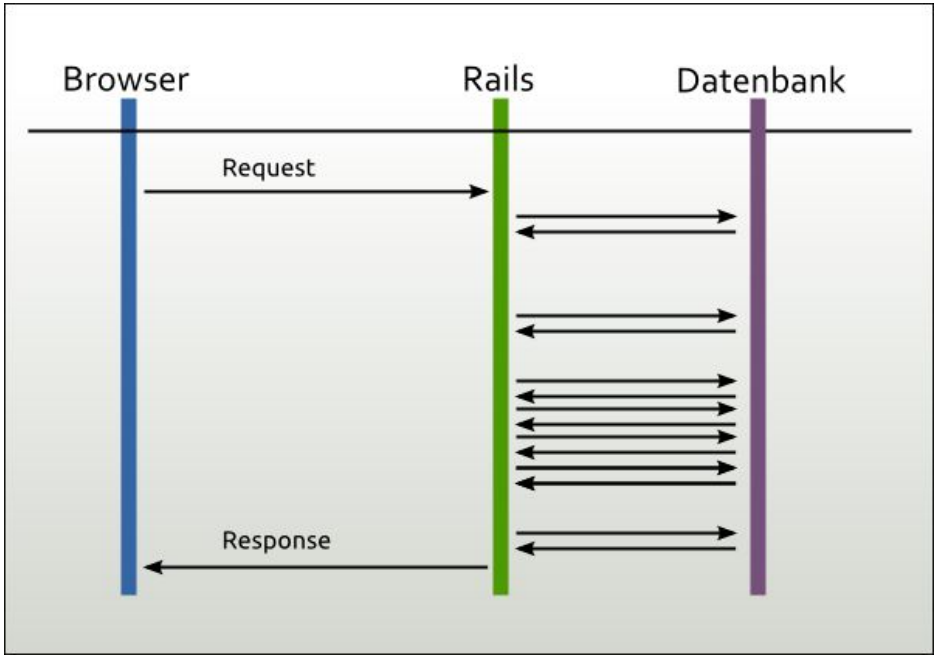
Linked Objekts

- A company with a list of vessels, each with a flag country



Sequence Diagram





includes()

```
@container_vessels =
  @company.container_vessels.
     order(:name).
     includes(:legal_country)
```

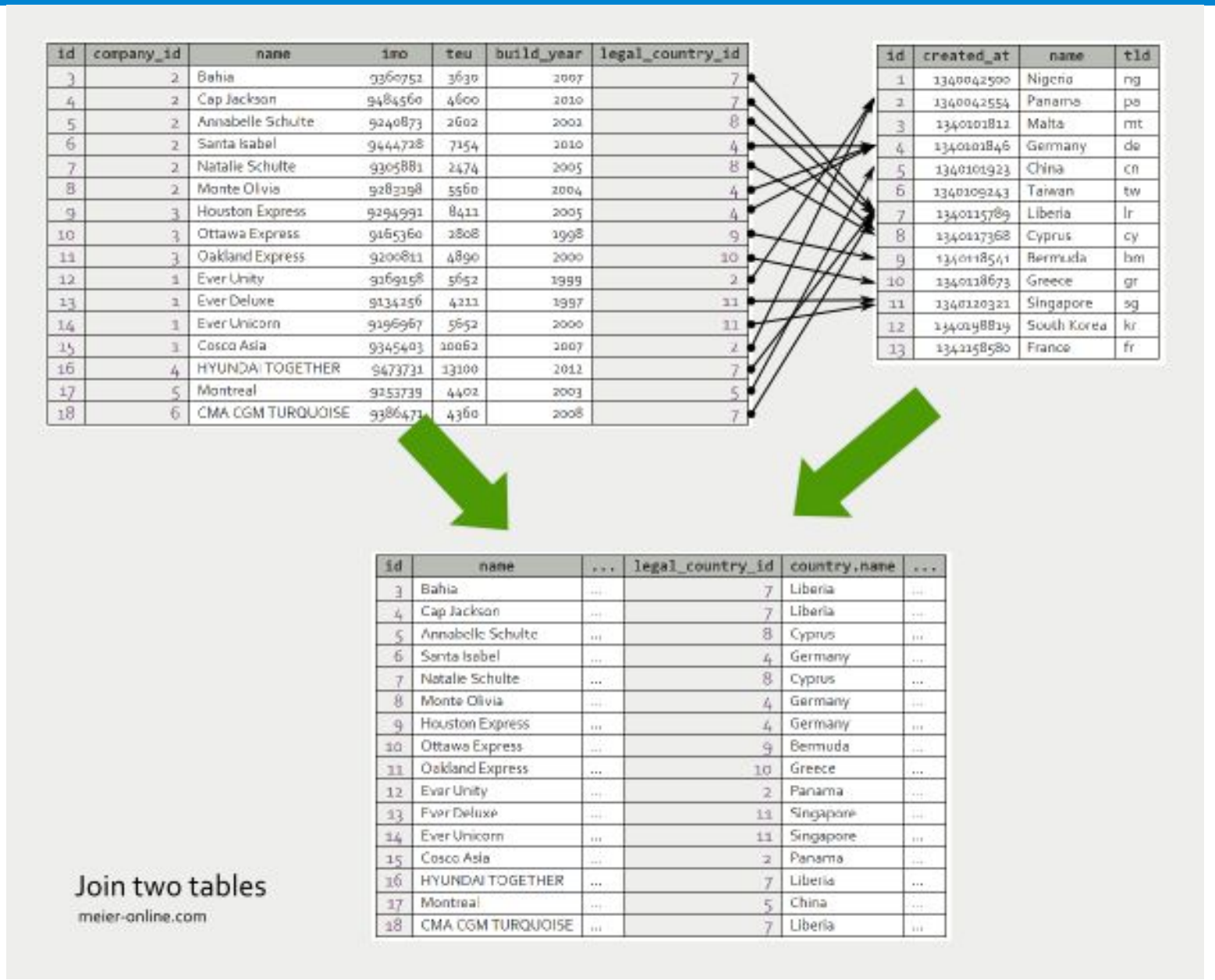
```
SELECT "container_vessels".*
   FROM "container_vessels"
   WHERE "container_vessels"."company_id" = 2
   ORDER BY name

SELECT "countries".*
   FROM "countries"
   WHERE "countries"."id" IN (8, 7, 4)
```

includes()

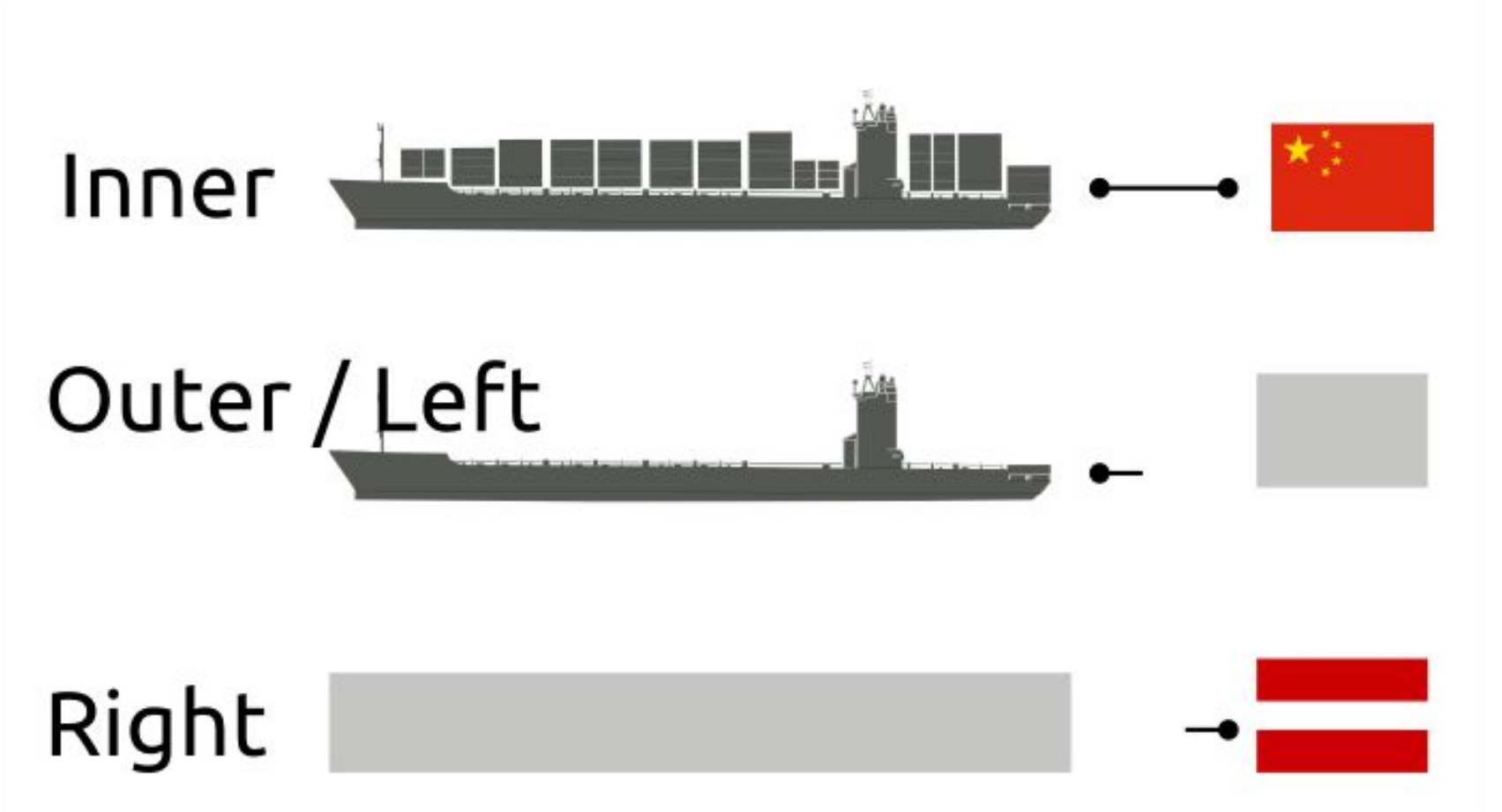
- Each query returns objects of one type
- Rails always in control
- Nesting possible
- Fine tuning difficult

```
.includes(:legal_country => :tax_rates)

.select('country.image????')
```

How does a join works again?



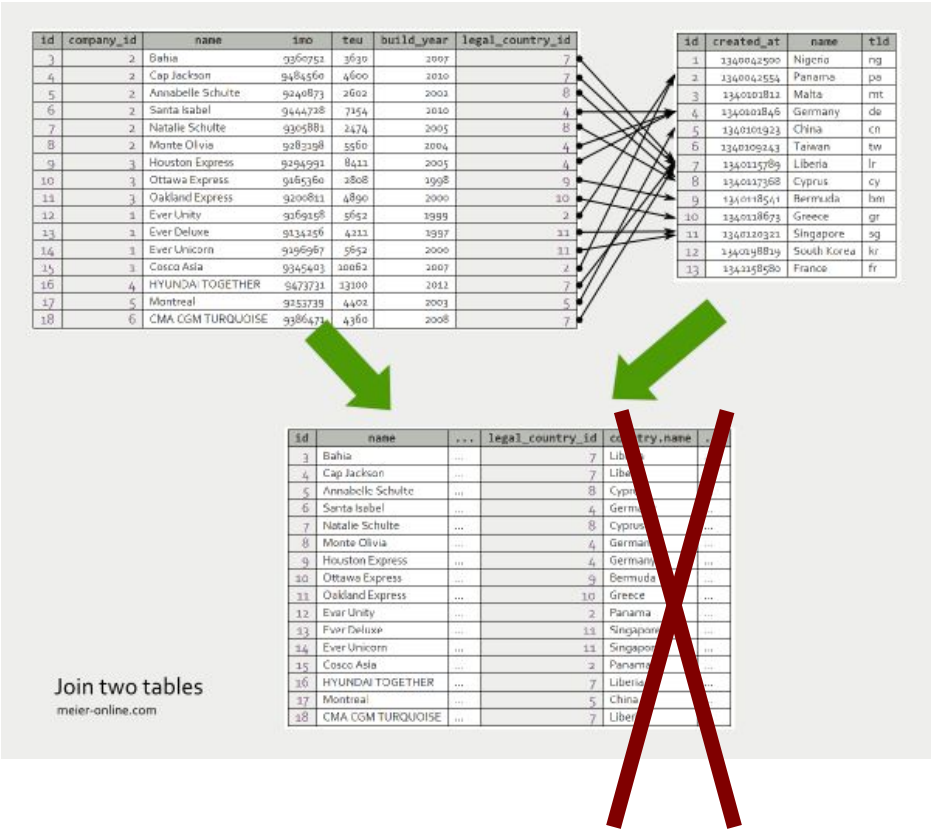
Inner/Left/Outer/Right



Rails joins

```
@container_vessels =
  @company.container_vessels.
   order(:name).
   joins(:legal_country)
```

- No vessels without a flag state
- No country data



Filter with joins()

- Filter with conditions in linked data
- Only target objects are returned
- Beware possible duplications!

```
@companies = Company.order(:name).
        joins(:container_vessels).
   where(["container_vessels.build_year > ?", 2009])
```

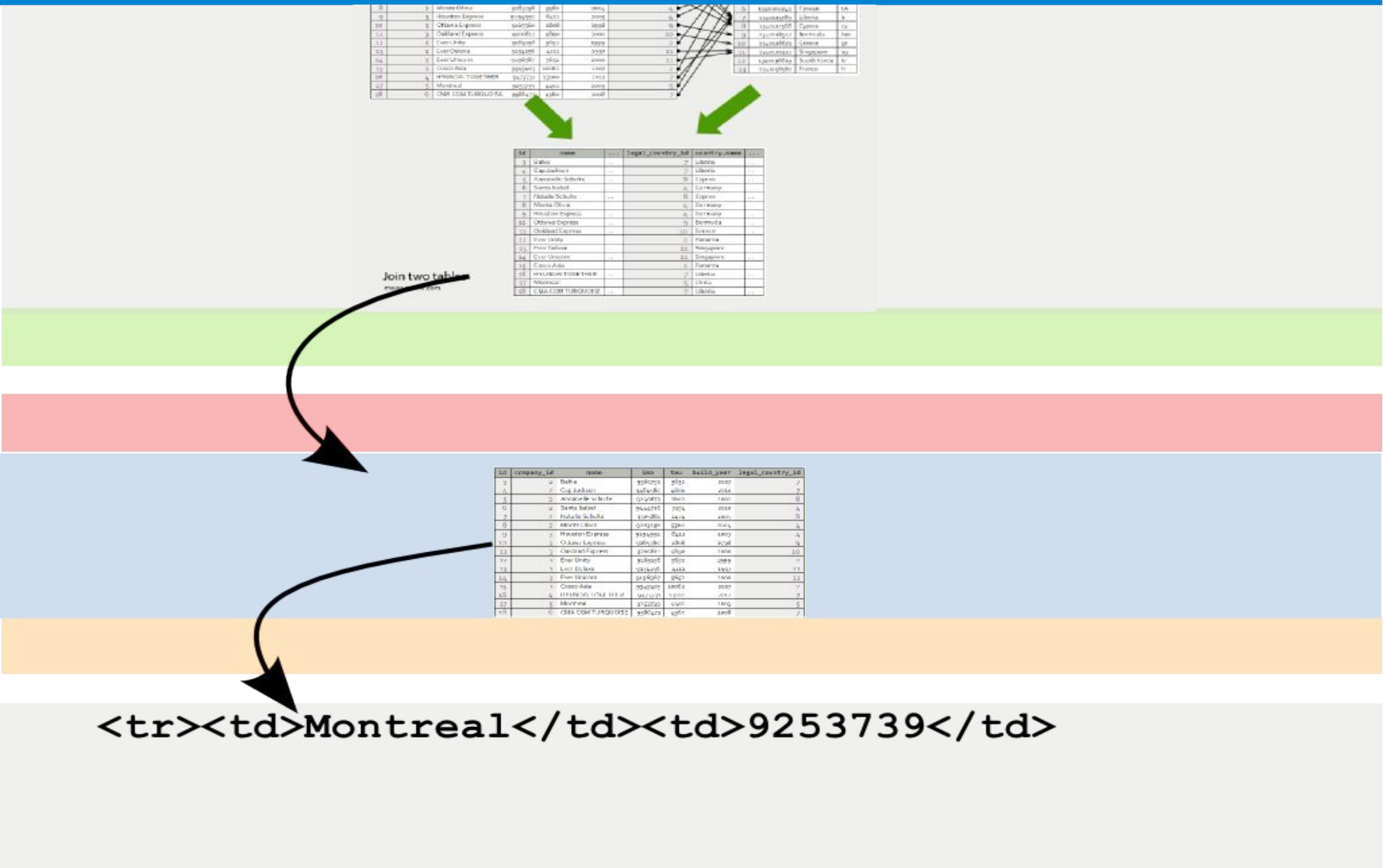
```
SELECT "companies".*
FROM "companies"
INNER JOIN "container_vessels"
  ON "container_vessels"."company_id" = "companies"."id"
WHERE (container_vessels.build_year > 2009)
ORDER BY name
```

Automatic Join in Associations

```
class Country < ActiveRecord::Base
  has_many :registering_companies,
           :through => :registered_vessels,
           :source => 'company',
           :class_name => 'Company',
           :uniq => true
...
@companies = @country.registering_companies
```

```
SELECT DISTINCT "companies".*
FROM "companies"
INNER JOIN "container_vessels"
  ON "companies"."id" = "container_vessels"."company_id"
WHERE "container_vessels"."legal_country_id" = 10
```

Use Database-Join directly?



Real Database Joins In Rails

```
connection = Company.connection

columns = "container_vessels.id, container_vessels.name,\
   container_vessels.imo, container_vessels.teu, \
    countries.name as legal_country_name"

sql = 'SELECT ' + columns + ' FROM "container_vessels" \
  JOIN "countries" \
  ON "countries"."id" = "container_vessels"."legal_country_
  WHERE "container_vessels"."company_id" = ' + @company.id.
  ' ORDER BY "container_vessels".name'

@vessel_data = connection.select_all(
     sql, 'ContainerVessel Overview Load')
```

Returned Values

- select_all: array of hashes
- select_rows: array of arrays

```
<% @vessel_data.each do |data| %>
  <tr>
    <td><%= data['name'] %></td>
    <td><%= data['imo'] %></td>
    <td><%= data['teu'] %></td>
    <td><%= data['legal_country_name'] %></td>
  ...
<% end %>
```

Checking Parameters

- SQL-Injection
- Methods difficult to find
- Since Rails 3.2: ActiveRecord::Sanitization
- For IDs: to_i.to_str

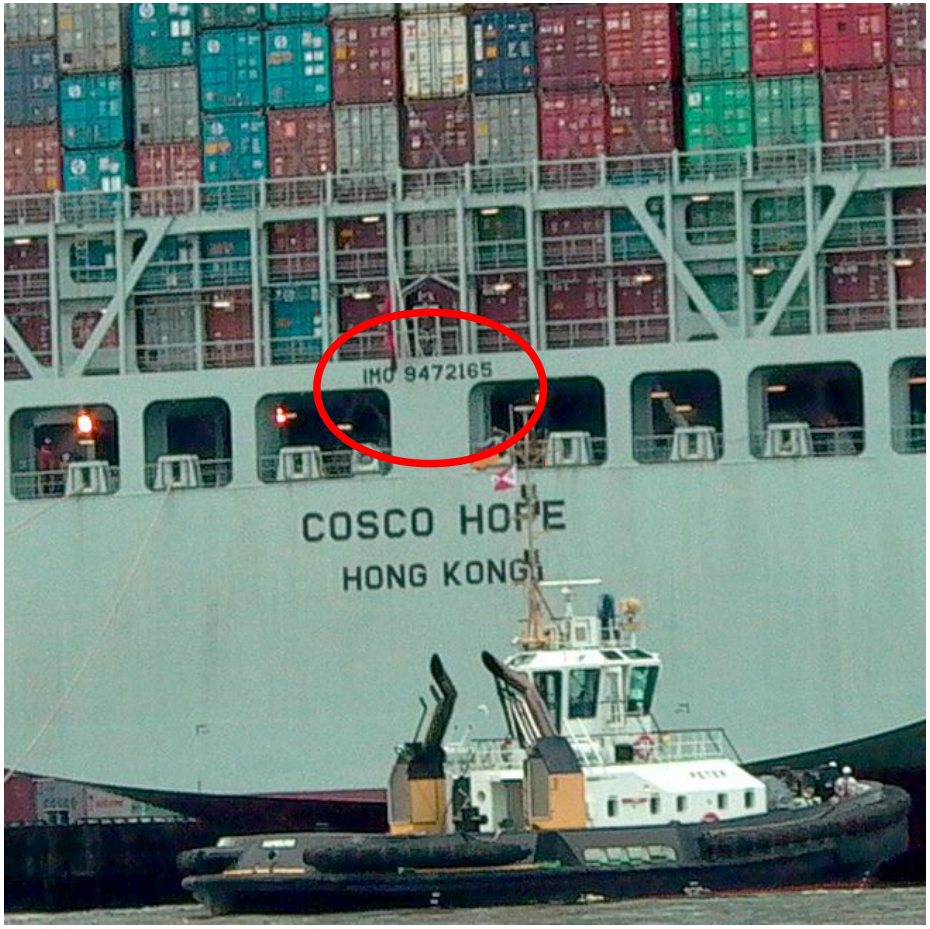
```
Company.where(
    'name like '%?', input)

record.sanitize_sql_array(..)

replace_bind_variables()
quote_bound_value()

connection.quote_string()
```

Writing

If you have performance problems during writing, the implications are often bad.

IDs

- ID-delivery can be a central bottle neck
- Sometimes already existing IDs can be used



Transactions

- Ensure consistency (ACID)
- Less locking, faster writes
- Use them if you have more than one write operation in an action

Mass Updates

- Company is sold
- All vessels get a new owner

```
UPDATE container_vessels
SET company_id = 7
WHERE company_id = 5
```

```
connection.update_sql(sql, "Updating vessel...")
```

Linked Updates

- Example usage: denormalisation
- Name of country should also be stored in vessel table

```
UPDATE container_vessels, country
SET container_vessels.country_name = country.name
WHERE container_vessels.legal_country_id = country.id
```

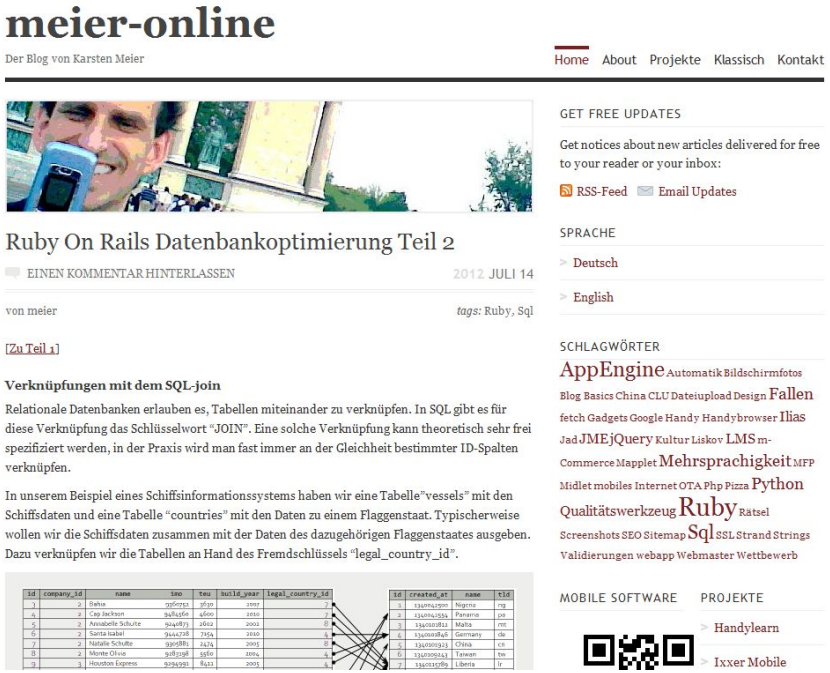
Don't be afraid of SQL

"Many people treat the relational database like a crazy aunt who's shut up in an attic and whom nobody wants to talk about"

Martin Fowler: OrmHate

... end

Website of Karsten Meier:



meier-online.com

Pictures:
Container ship by jogdragoon, openclipart.org
Hammer5 by Krystof Jetmar, openclipart.org
OOCL Montreal & Cosco Hope fotografiert by Karsten Meier in Hamburg Port 2012