

# Database Optimization



**Scaling  
Ruby-on-Rails  
By Example**

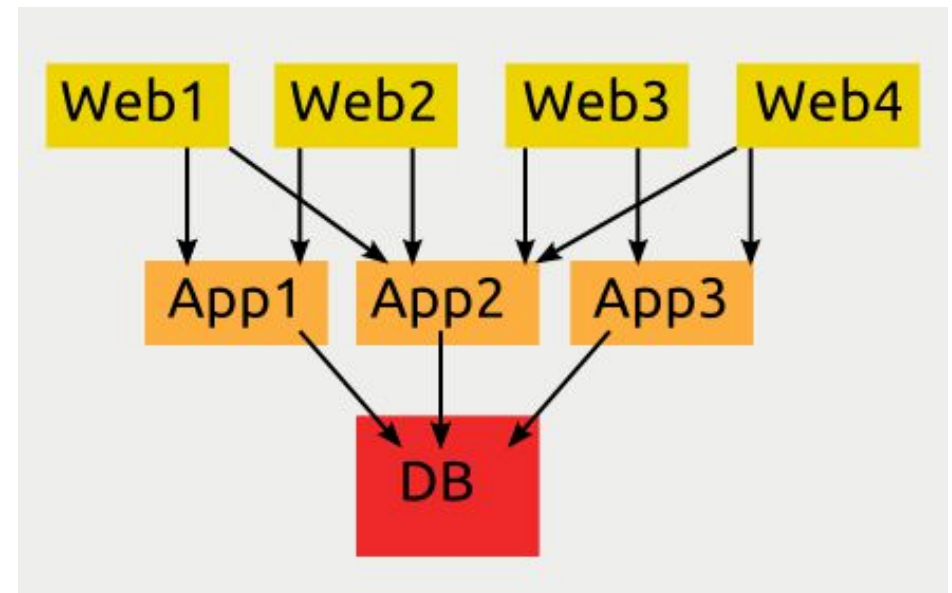
Karsten Meier  
meier-online.com

# 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



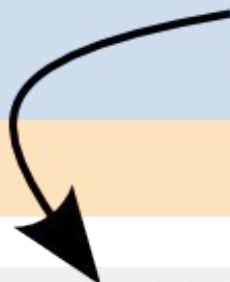
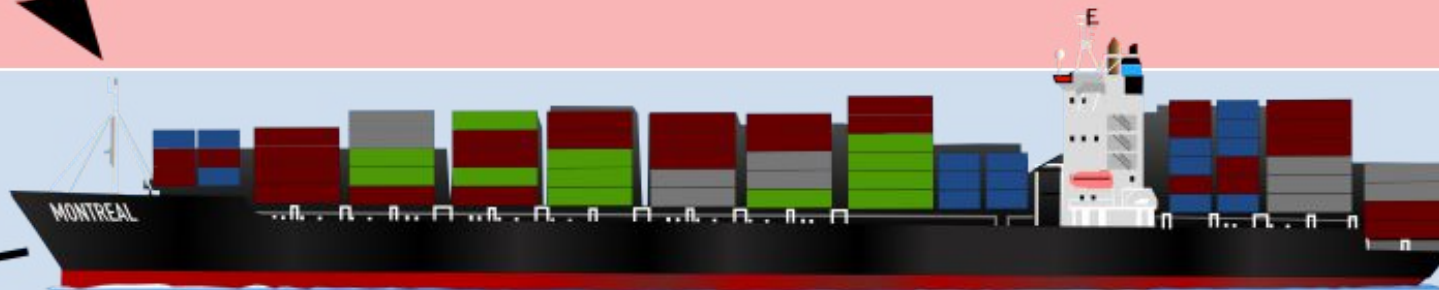
# Effects Of Less Database Queries

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



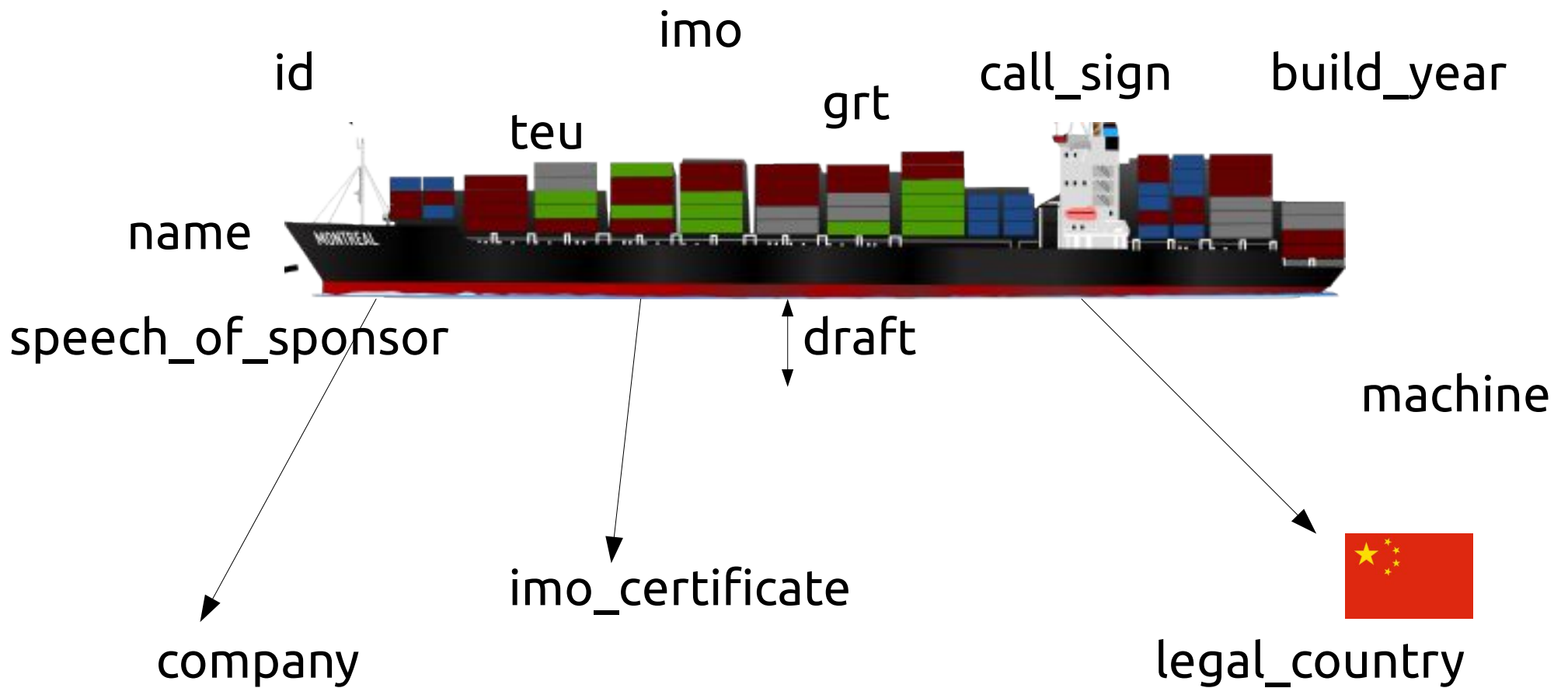
# Layers Of A Web Application

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	9283298	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	4221	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



# Shadow 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

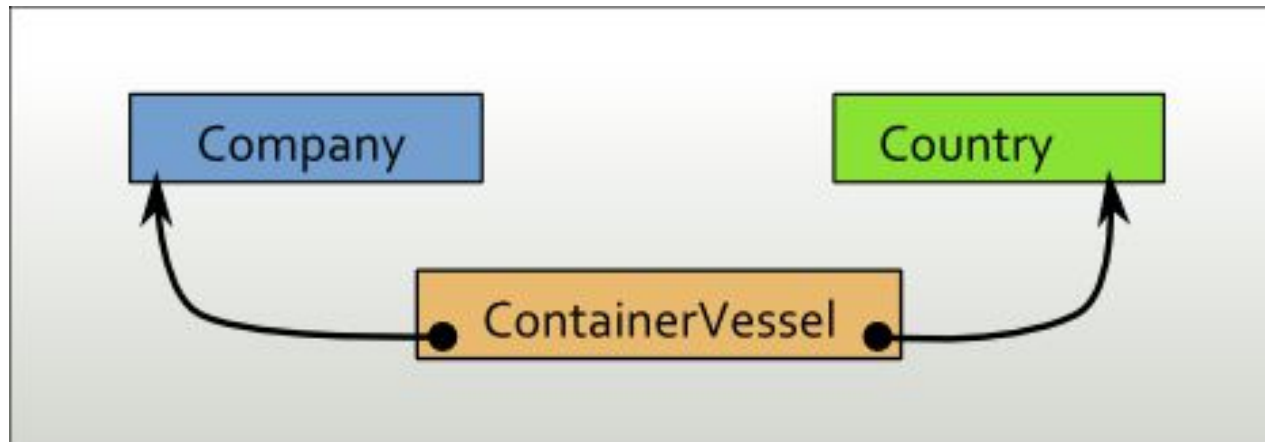
- Weight of all container on the vessel?
- DB can do the calculation
- Rails does not see the individual containers

~~@vessel.containers.inject{...}~~

@vessel.containers.sum('weight')

# Linked Objects

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





# 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?

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	9283398	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	9169198	5652	1999	2
13	1	Ever Deluxe	9134256	4211	1997	11
14	1	Ever Unicorn	9196967	5652	2000	11
15	3	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

id	created_at	name	tld
1	1340042500	Nigeria	ng
2	1340042554	Panama	pa
3	1340101811	Malta	mt
4	1340101846	Germany	de
5	1340101923	China	cn
6	1340109243	Taiwan	tw
7	1340115789	Liberia	lr
8	1340117368	Cyprus	cy
9	1340118541	Bermuda	bm
10	1340118673	Greece	gr
11	1340120321	Singapore	sg
12	1340148819	South Korea	kr
13	1342158580	France	fr

id	name	...	legal_country_id	country.name	...
3	Bahia	...	7	Liberia	...
4	Cap Jackson	...	7	Liberia	...
5	Annabelle Schulte	...	8	Cyprus	...
6	Santa Isabel	...	4	Germany	...
7	Natalie Schulte	...	8	Cyprus	...
8	Monte Olivia	...	4	Germany	...
9	Houston Express	...	4	Germany	...
10	Ottawa Express	...	9	Bermuda	...
11	Oakland Express	...	10	Greece	...
12	Ever Unity	...	2	Panama	...
13	Ever Deluxe	...	11	Singapore	...
14	Ever Unicorn	...	11	Singapore	...
15	Cosco Asia	...	2	Panama	...
16	HYUNDAI TOGETHER	...	7	Liberia	...
17	Montreal	...	5	China	...
18	CMA CGM TURQUOISE	...	7	Liberia	...

Join two tables  
meier-online.com

# Inner/Left/Outer/Right

Inner



Outer / Left



Right

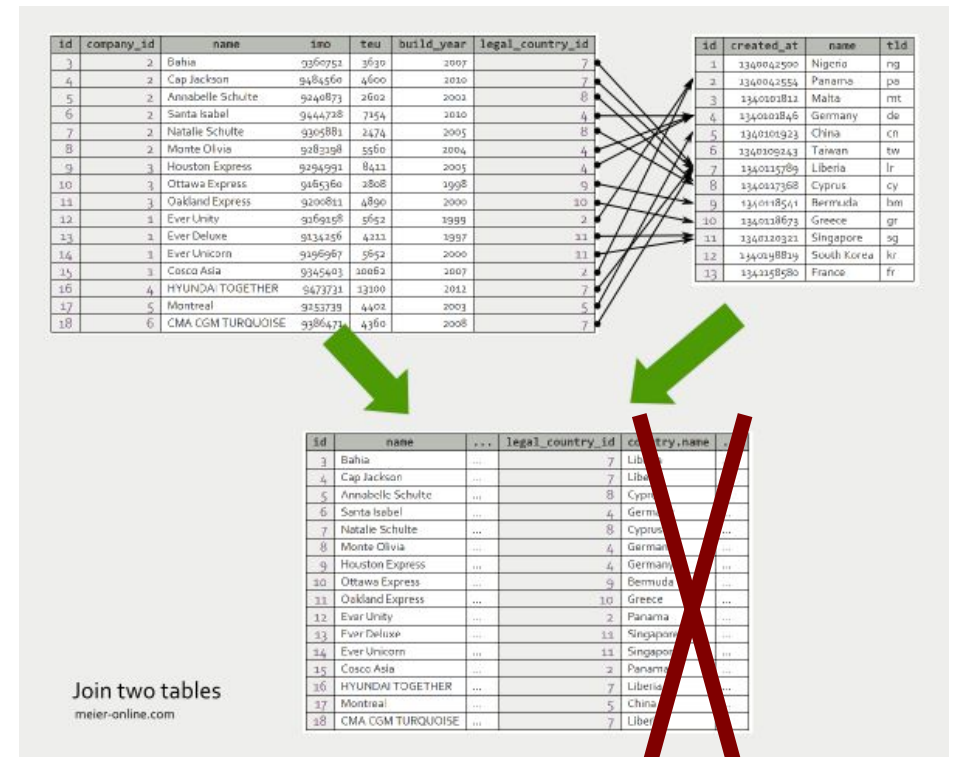




# Rails joins

```
@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

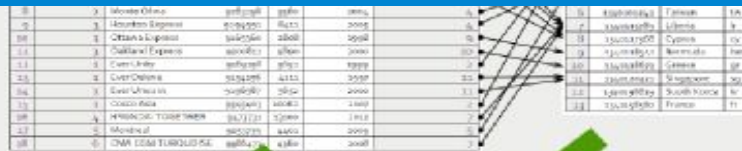
```
class Country < ActiveRecord::Base
  has_many :registering_companies,
           :through => :registered_vessels,
           :source => 'company',
           :class_name => 'Company',
           :uniq => true

  ...

  @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?



id	name	iso	taxi	build_year	legal_country_id
1	Shiba	JP	470	2007	7
2	Capulakina	---	---	---	7
3	Aerobike-Schulke	---	---	---	8
4	Santa Isabel	---	---	---	4
5	Fabula Schulke	---	---	---	8
6	Monte Carlo	---	---	---	8
7	Hausen Express	---	---	---	4
8	Ottawa Express	---	---	---	9
9	Ottawa Express	---	---	---	10
10	Ever Unity	---	---	---	3
11	Ever Unity	---	---	---	3
12	Low-E-Bus	---	---	---	11
13	Ever Unity	---	---	---	11
14	Coast Asia	---	---	---	11
15	Coast Asia	---	---	---	11
16	HYUNDAI TOBI SEW	---	---	---	7
17	Montreal	---	---	---	5
18	OMA COM TURQUOISE	---	---	---	7

id	name	iso	legal_country_id	build_year	taxi
1	Shiba	---	7	---	---
2	Capulakina	---	7	---	---
3	Aerobike-Schulke	---	8	---	---
4	Santa Isabel	---	4	---	---
5	Fabula Schulke	---	8	---	---
6	Monte Carlo	---	8	---	---
7	Hausen Express	---	4	---	---
8	Ottawa Express	---	9	---	---
9	Ottawa Express	---	10	---	---
10	Ever Unity	---	3	---	---
11	Ever Unity	---	3	---	---
12	Low-E-Bus	---	11	---	---
13	Ever Unity	---	11	---	---
14	Coast Asia	---	11	---	---
15	Coast Asia	---	11	---	---
16	HYUNDAI TOBI SEW	---	7	---	---
17	Montreal	---	5	---	---
18	OMA COM TURQUOISE	---	7	---	---

Join two tables  
using a join

id	company_id	name	iso	taxi	build_year	legal_country_id
7	4	Shiba	JP	470	2007	7
8	2	Capulakina	---	---	---	7
9	3	Aerobike-Schulke	---	---	---	8
10	4	Santa Isabel	---	---	---	4
11	5	Fabula Schulke	---	---	---	8
12	6	Monte Carlo	---	---	---	8
13	7	Hausen Express	---	---	---	4
14	8	Ottawa Express	---	---	---	9
15	9	Ottawa Express	---	---	---	10
16	10	Ever Unity	---	---	---	3
17	11	Ever Unity	---	---	---	3
18	12	Low-E-Bus	---	---	---	11
19	13	Ever Unity	---	---	---	11
20	14	Coast Asia	---	---	---	11
21	15	Coast Asia	---	---	---	11
22	16	HYUNDAI TOBI SEW	---	---	---	7
23	17	Montreal	---	---	---	5
24	18	OMA COM TURQUOISE	---	---	---	7

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

# 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_id" \  
  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()
```

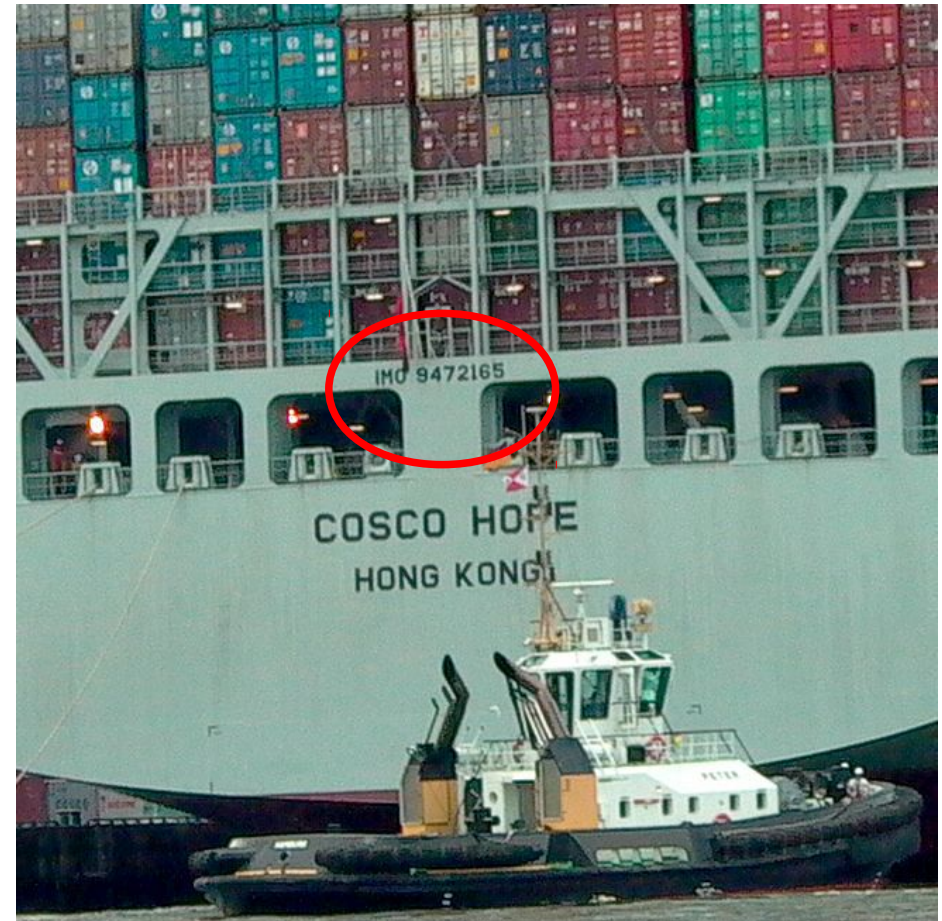
# Writing

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

Der Blog von Karsten Meier

[Home](#) [About](#) [Projekte](#) [Klassisch](#) [Kontakt](#)



#### Ruby On Rails Datenbankoptimierung Teil 2

EINEN KOMMENTAR HINTERLASSEN

2012 JULI 14

von meier

tags: Ruby, Sql

[\[Zu Teil 1\]](#)

#### Verknüpfungen mit dem SQL-join

Relationale Datenbanken erlauben es, Tabellen miteinander zu verknüpfen. In SQL gibt es für diese Verknüpfung das Schlüsselwort "JOIN". Eine solche Verknüpfung kann theoretisch sehr frei spezifiziert werden, in der Praxis wird man fast immer an der Gleichheit bestimmter ID-Spalten verknüpfen.

In unserem Beispiel eines Schiffsinformationssystems haben wir eine Tabelle "vessels" mit den Schiffsdaten und eine Tabelle "countries" mit den Daten zu einem Flaggenstaat. Typischerweise wollen wir die Schiffsdaten zusammen mit der Daten des dazugehörigen Flaggenstaates ausgeben. Dazu verknüpfen wir die Tabellen an Hand des Fremdschlüssels "legal\_country\_id".

id	company_id	name	imo	teu	builid_year	legal_country_id	id	created_at	name	iso
1	2	Bahia	9205011	2700	2007	2	1	2007-03-09	Nigeria	ng
4	2	Cap Jackson	9204006	4000	2000	2	1	2007-03-09	Paraguay	py
5	2	Annabelle Schulte	9204077	2000	2000	0	1	2007-03-09	Malta	mt
6	2	Santa Isabel	9204078	2000	2000	0	4	2007-03-09	Germany	de
7	2	Natalie Schulte	9204081	1800	2005	0	5	2007-03-09	China	cn
8	2	Moree Office	9204082	2000	2006	4	6	2007-03-09	Taiwan	tw
9	1	Houston Express	9204090	8000	2005	4	7	2007-03-09	Liberia	lr

#### GET FREE UPDATES

Get notices about new articles delivered for free to your reader or your inbox:

RSS-Feed  Email Updates

#### SPRACHE

> [Deutsch](#)

> [English](#)

#### SCHLAGWÖRTER

[AppEngine](#) [Automatik](#) [Bildschirmfotos](#) [Blog Basics](#) [China](#) [CLU](#) [Dateiupload](#) [Design](#) [Fallen](#) [fetch](#) [Gadgets](#) [Google](#) [Handy](#) [Handybrowser](#) [Ilias](#) [Jad](#) [jQuery](#) [Kultur](#) [Liskov](#) [LMS](#) [m-Commerce](#) [Mapplet](#) [Mehrsprachigkeit](#) [MFP](#) [Midlet](#) [mobiles Internet](#) [OTA](#) [Php](#) [Pizza](#) [Python](#) [Qualitätswerkzeug](#) [Ruby](#) [Rätsel](#) [Screenshots](#) [SEO](#) [Sitemap](#) [Sql](#) [SSL](#) [Strand](#) [Strings](#) [Validierungen](#) [webapp](#) [Webmaster](#) [Wettbewerb](#)

#### MOBILE SOFTWARE PROJEKTE



> [Handylearn](#)

> [Ixerx Mobile](#)

# meier-online.com

Pictures:

Container ship by jogdragoon, openclipart.org

Hammer5 by Krystof Jetmar, openclipart.org

OCL Montreal & Cosco Hope fotografiert by Karsten Meier in Hamburg Port 2012