

RAVENDB 3.0



Михаил Ярийчук

Hibernating Rhinos

michael.yarichuk@hibernatingrhinos.com



А ЧТО ЭТО ТАКОЕ, RAVENDB? NOSQL DOCUMENT DATABASE

companies/1

Data

Metadata

```
{
  "ExternalId": "ALFKI",
  "Name": "Alfreds Futterkiste",
  "Contact": {
    "Name": "Maria Anders",
    "Title": "Sales Representative"
  },
  "Address": {
    "Line1": "Obere Str. 57",
    "Line2": null,
    "City": "Berlin",
    "Region": null,
    "PostalCode": "12209",
    "Country": "Germany"
  },
  "Phone": "030-0074321",
  "Fax": "030-0076545"
}
```



ПЕРВОЕ ПОКОЛЕНИЕ БАЗ ДАННЫХ NOSQL

Fault-tolerance

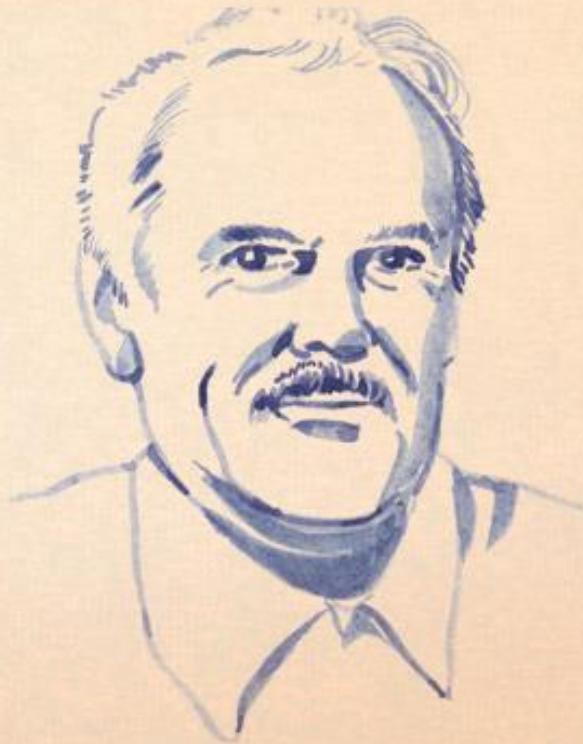
by @jrecursive



А ЧТО НАСЧЕТ SQL?

Father of the Relational Database: Edgar F. Codd

A British computer scientist, Codd made important contributions to the theory of relational databases. While working for IBM, he created the relational model for database management.



СИЛЬНЫЕ СТОРОНЫ RAVENDB...



```

public class User
{
    -references
    public string Id { get; set; }
    -references
    public string Name { get; set; }
    -references
    public string Email { get; set; }
}

-references
public void SimpleExample()
{
    using (var store = new EmbeddableDocumentStore())
    {
        store.Initialize();
        using (var session = store.OpenSession())
        {
            session.Store(new User { Name = "Bob", Email = "bob@foo.com" });
            session.SaveChanges();
        }

        using (var session = store.OpenSession())
        {
            var fetchedDataObject = session.Query<User>()
                .FirstOrDefault(o => o.Name.Equals("Bob") &&
                    o.Email.StartsWith("bob"));

            if (fetchedDataObject != null)
                Console.WriteLine(fetchedDataObject.Id + "," + fetchedDataObject.Name);
        }
    }
}

```

ИНТУИТИВНЫЙ API?

1. Загрузить в БД
новый объект
2. Получить объект
из БД используя
запрос



ОСТОРОЖНО, НОГА!



<http://freedomfeens.com/blog/2013/02/02/prohibited-thou-shalt-not-photoshop/obamas-shotgun-foot/>



```
using (var session = store.OpenSession())
{
    var recentOrders = session.Query<Order>()
        .Include(o => o.Company)
        .Where(o => o.OrderedAt.Year >= 1998)
        .Take(5)
        .ToList();

    foreach (var order in recentOrders)
    {
        //in the query there was Include() --> inserts the relevant companies to cache --> Load() loads record from cache
        var company = session.Load<dynamic>(order.Company);
        Console.WriteLine("Order Id: {0}, ordered by company: {1}", order.Id, company.Name);
    }
}
```


ГАРАНТИИ...



ДАЧА

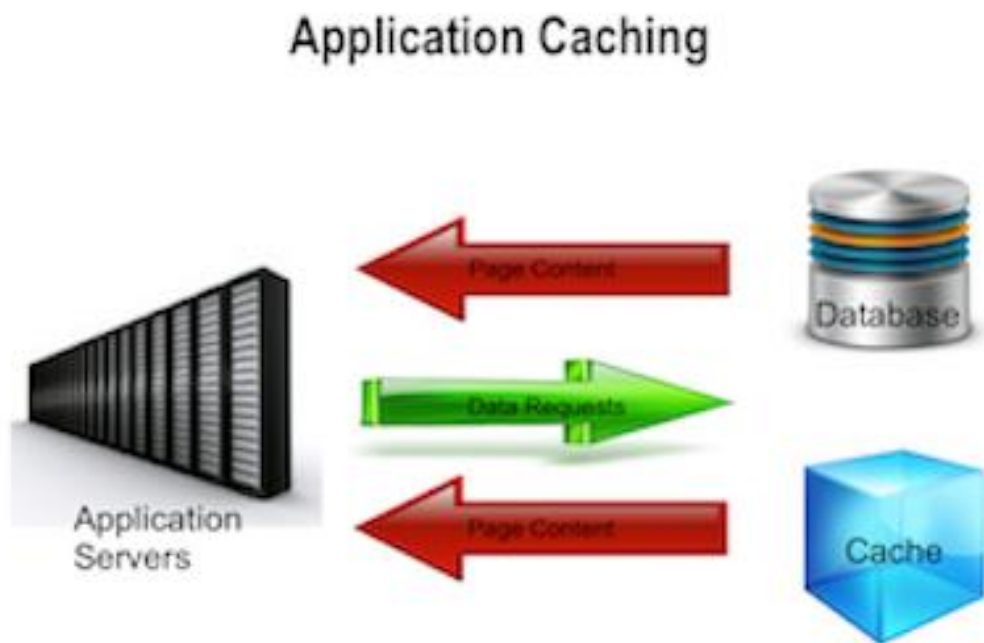
ГАРАНТИЯ ХОРОШЕГО НАСТРОЕНИЯ

CHANGES... API

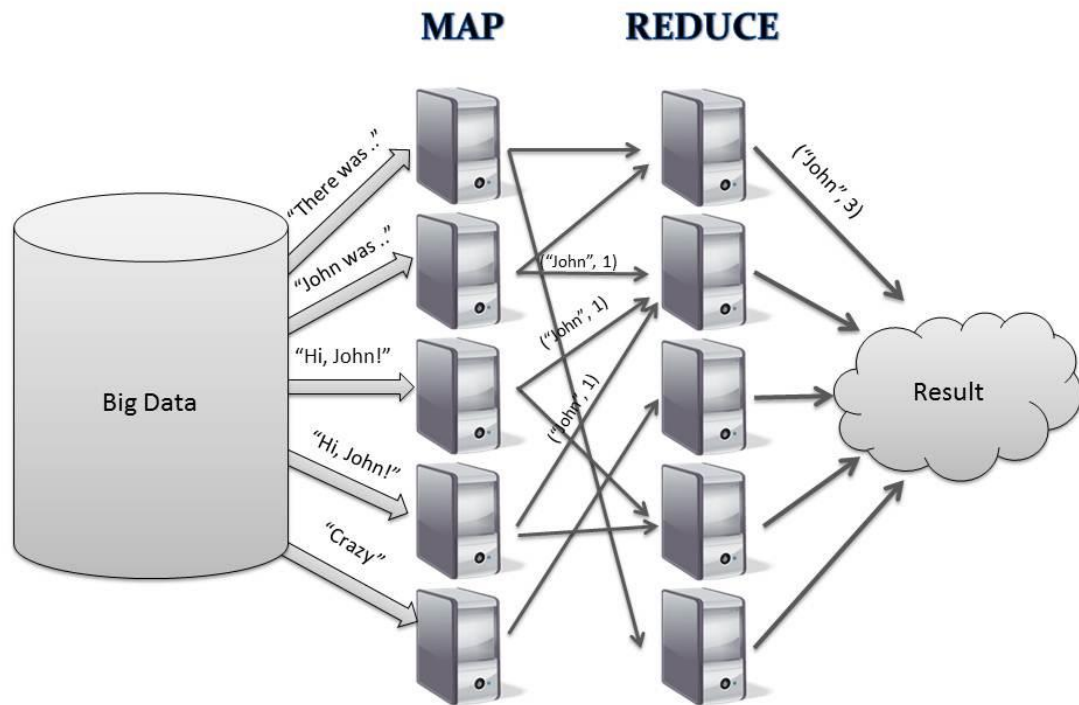
```
var store = new DocumentStore
{
    Url = "http://localhost:8079"
}.Initialize();

store.Changes()
    .ForDocumentsOfType(typeof (Order))
    .Where(notification => notification.Type == DocumentChangeTypes.Put)
    .Subscribe(notification => Console.WriteLine("received order " + notification.Id));
```

КЭШ - ДЕЛО ТОНКОЕ!



MAP/REDUCE ЭТО СЛОЖНО?



Complexity

Pull stick up, plane goes up. Push stick down, plane goes down.
The buttons? They're there to impress the chicks.
Chicks are COMPLEX

MAP/REDUCE — SQL

```
SELECT TOP 3 SUM(1.Quantity * (1.PricePerUnit - 1.Discount)), c.Id
FROM Orders o
    JOIN Companies c ON c.Id = o.CompanyId
    JOIN OrderLines l ON l.OrderId = o.Id
GROUP BY c.Id
ORDER BY SUM(1.Quantity * (1.PricePerUnit - 1.Discount)) DESC
```

```
SELECT c.Id, c.Name from Company c
WHERE c.Id in ('company/foo1', 'company/foo2', 'company/foo3')
```

MAP/REDUCE - ИНДЕКС

```
public class CompanyOrderTotalIndex : AbstractIndexCreationTask<Order, CompanyOrdersTotal>
{
    1 reference
    public CompanyOrderTotalIndex()
    {
        Map = orders => from order in orders
                        from orderLine in order.Lines
                        select new CompanyOrdersTotal
                        {
                            CompanyId = order.Company,
                            Total = orderLine.Quantity * (orderLine.PricePerUnit - orderLine.Discount)
                        };

        Reduce = results => from result in results
                            group result by result.CompanyId
                                into g
                            select new CompanyOrdersTotal
                            {
                                CompanyId = g.Key,
                                Total = g.Sum(x => x.Total)
                            };
    }
}
```

MAP/REDUCE - RAVENDB

```
var query = session.Query<CompanyOrdersTotal, CompanyOrderTotalIndex>()  
    .Include(x => x.CompanyId)  
    .OrderByDescending(x => x.Total)  
    .Take(3)  
    .ToList();
```

HIGH AVAILABILITY/DR



HIGH AVAILABILITY/DR

Documents ▾

Indexes ▾

Query ▾

Tasks

Settings

Status

+ New ▾

Go To Document

Databases ▸ TestDB ▸ Settings

Save Changes

Database Settings

Periodic Backup

Replication

+ Add Replication

Disable: ☐

Url:

Client Visible Url:

Database:

User Name:

Password:

Domain:

ApiKey:

Skip Failover: ☐

Transitive Replication:

Disable: ☐

Url:

Client Visible Url:

Database:

User Name:

Password:

Domain:

ApiKey:

Skip Failover: ☐

Transitive Replication:

VORON И ОСТАЛЬНЫЕ НОВШЕСТВА



http://fc03.deviantart.net/fs70/i/2011/296/5/d/huginn_and_muninn_by_victoryordeath-d4dq7zy.png

Вопросы?

