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"
}
```

Better. Stronger. Faster.
RAVENDB

Hibernating Rhinos

ПЕРВОЕ ПОКОЛЕНИЕ БАЗ ДАННЫХ NOSQL
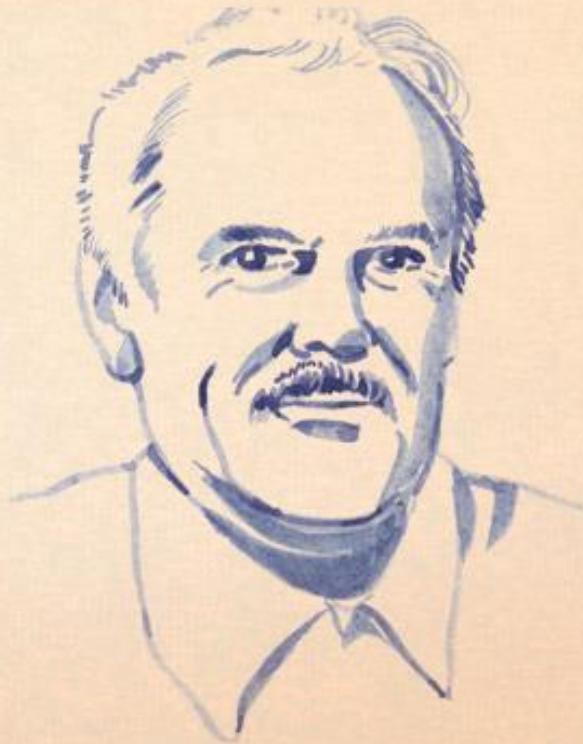
А ЧТО НАСЧЕТ SQL?

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...

```csharp
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!

СТРЕЛЬБА ПО НОГАМ - 1

```csharp
public IEnumerable<Order> GetOrdersWithPaging(int latestShippingYear,int pageNumber, int pageSize)
{
    using (var db = new Database("Northwind"))
    {
        var orders = db.Query<Order>("SELECT * FROM Orders WHERE YEAR(ShippedDate) > " + latestShippingYear);

        return orders.Skip(pageNumber * pageSize).Take(pageSize);
    }
}
```

СТРЕЛЬБА ПО НОГАМ - 2

```csharp
using (var db = new Database("Northwind"))
{
    var orders = db.Query<Order>("SELECT * FROM Orders");

    foreach (var order in orders)
    {
        var orderLines = db.Query<OrderLine>("SELECT * FROM OrderLines WHERE OrderId = " + order.Id);
        DoSomethingUsefulWithOrderLines(orderLines);
    }
}
```

```
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);
    }
}
```
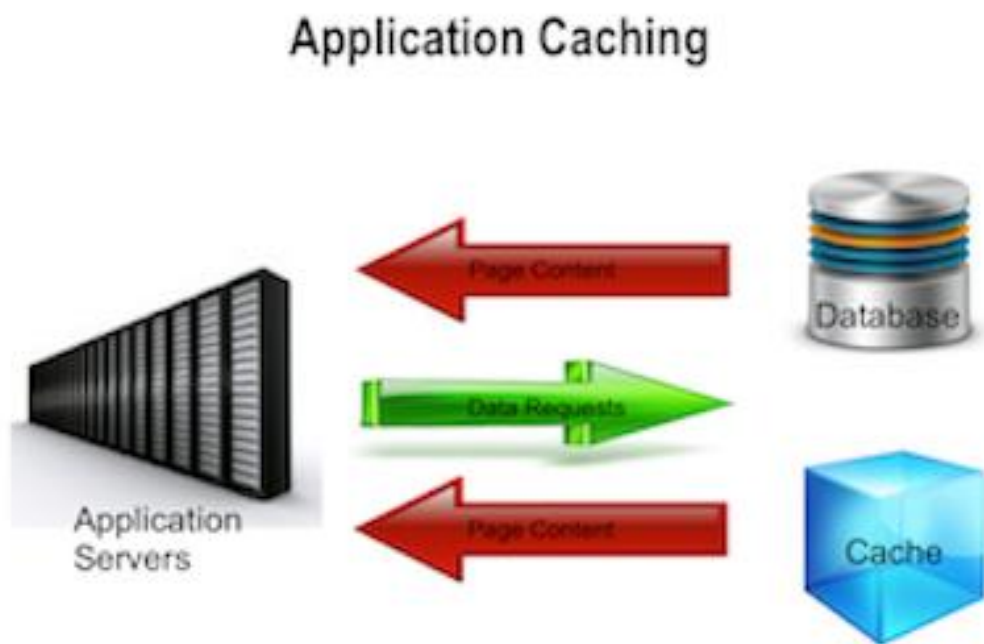
ГАРАНТИИ…

CHANGES... API

```csharp
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));
```
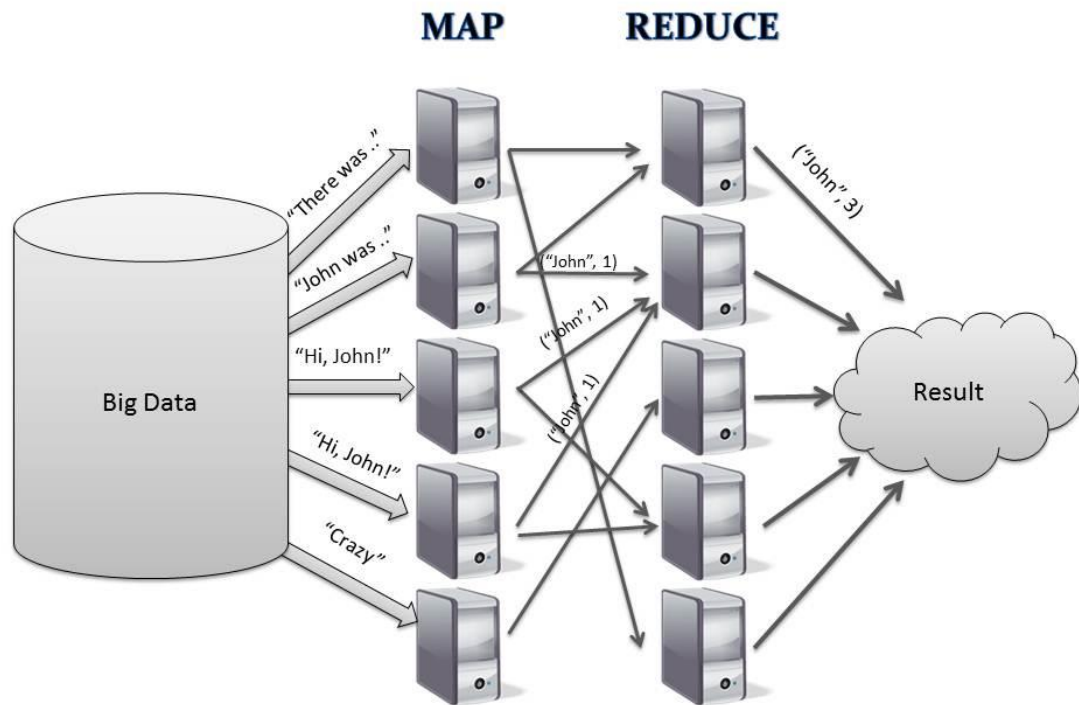
КЭШ - ДЕЛО ТОНКОЕ!

MAP/REDUCE ЭТО СЛОЖНО?

MAP/REDUCE — SQL

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


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

MAP/REDUCE - ИНДЕКС

```csharp
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

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

HIGH AVAILABILITY/DR

HIGH AVAILABILITY/DR

VORON И ОСТАЛЬНЫЕ НОВШЕСТВА

Вопросы?