

Graph databases

- why and how

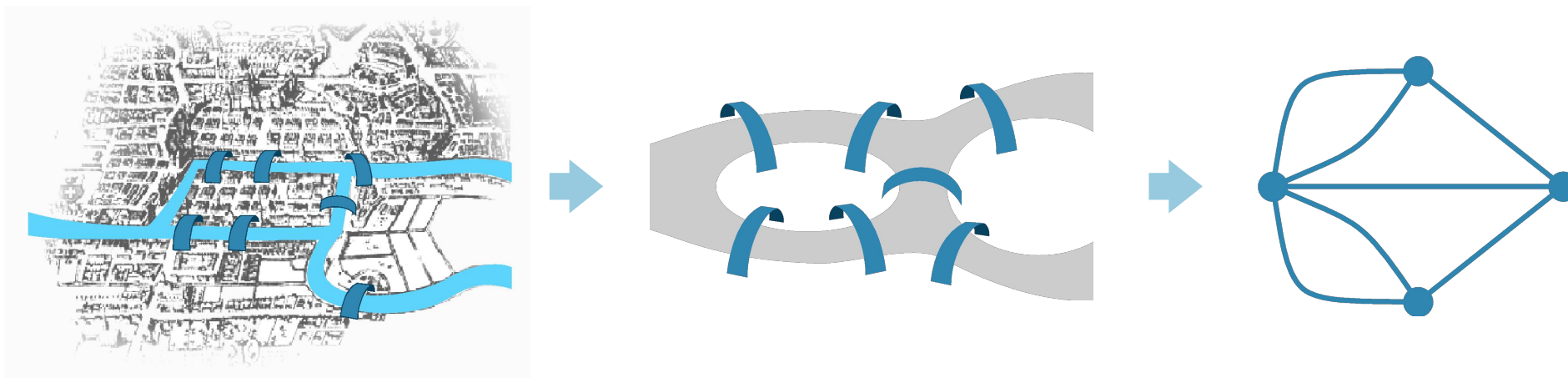
O mnie

- Chief Architekt w **ITMAGINATION**
- Architekt i TeamLeader w jednym z największych projektów w .NET w Polsce
- Zajmuję się systemami rozproszonymi i tym co można wywnioskować z danych
- Po godzinach **cook.it**
- No i można mnie spotkać na rekrutacji 😊

Agenda

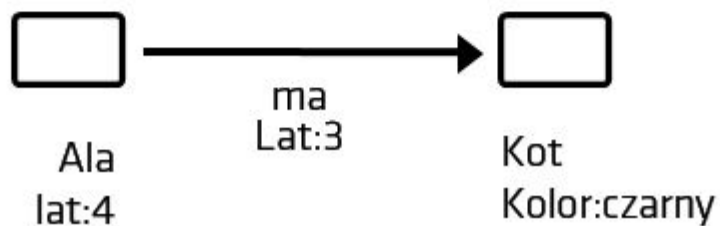
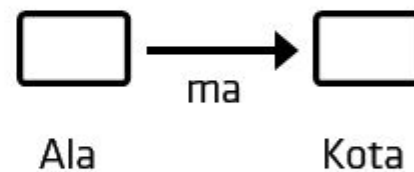
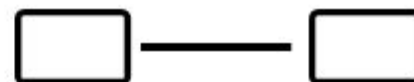
- Grafy
- Bazy grafowe
- Wydajność
- Zastosowania
- Neo4j i Cypher – demo
- Kiedy stosować

Graf

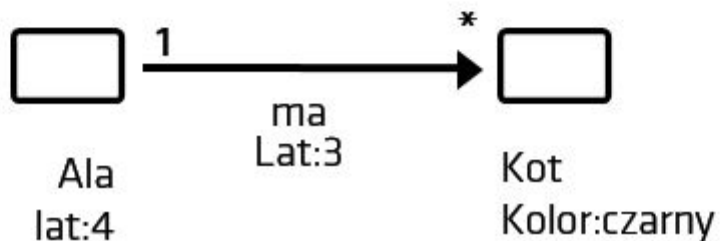
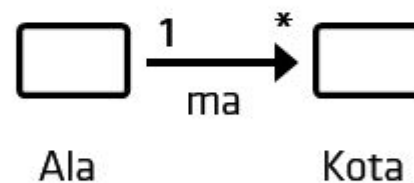
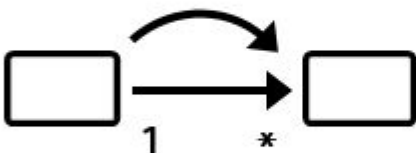
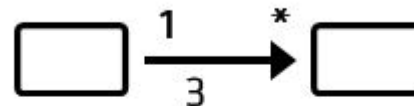
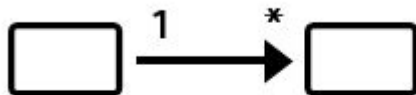
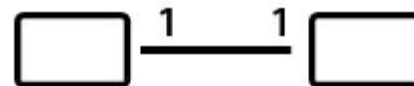


$$G=(E,V)$$

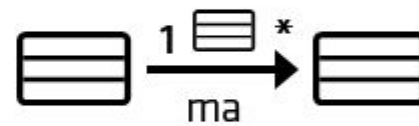
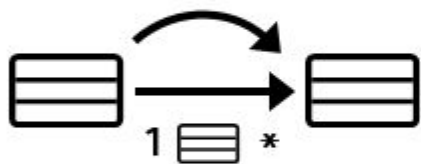
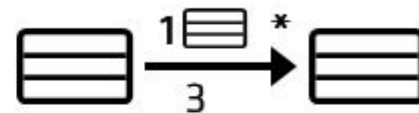
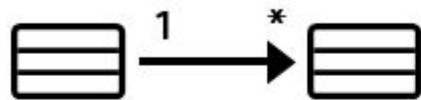
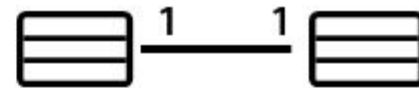
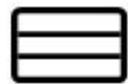
Graf



Graf



Graf



Ala

Kota



Ala
lat:4

ma
Lat:3

Kot
Kolor:czarny

Odrobina historii

Historia baz relacyjnych

- 1970 – Edgar F. Codd – model relacyjny
- Co innego działo się latach 70?
 - Znaczący spadek ceny pamięci RAM – tylko **734\$ za 1 MB**
 - IBM 3310 oferuje 100MB dyski
 - 1971 Intel 4004 – 740kHz, 8 bitów
 - 1974 Intel 8008 – 800kHz, 16 bitów, 1MB pamięci
 - 1977 powstaje Oracle, apple 1
 - 1978 Intel 8086 – 4MHz- 10MHz



Historia baz grafowych

- 1736 – Problem mostów
- 1852 – Problem pokolorowania mapy
- 1936 – Książka „Teoria Grafów”
- 1948 – ENIAC
- 1969 – Częściowe rozwiązanie problemu mapy
- 1994 – Rozwiązanie problemu mapy
- 2002 – Pentium 4 3,8GHz
- 2003 – Pierwsze komercyjne bazy grafowe

Stan obecny

Natywne

Procesowanie

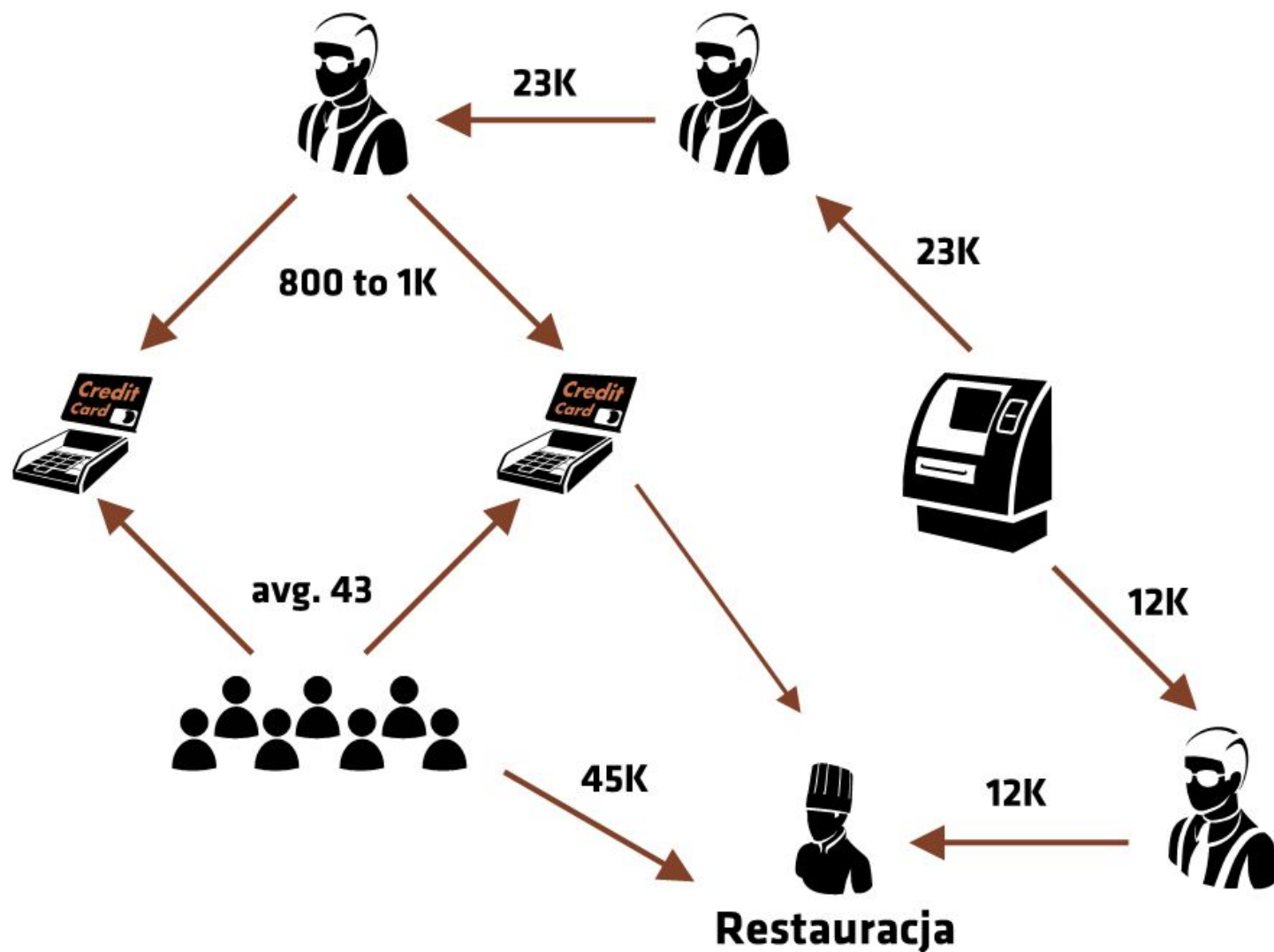


Przechowywanie

Natywne

Zastosowania

Pranie brudnych pieniędzy



Podpowiedzi

The screenshot shows the Amazon product page for the book "Writing High-Performance .NET Code" by Ben Watson. The page includes the book cover, a "Look inside" link, and a description. Below the description, there is a "Frequently Bought Together" section with three books: "Writing High-Performance .NET Code", "Adaptive Code via C#", and "Microsoft .NET - Architecting Applications for the Enterprise". The "Customers Who Bought This Item Also Bought" section lists seven related books, each with a "Look inside" link, a star rating, and a price.

Writing High-Performance .NET Code Paperback – July 23, 2014
by Ben Watson (Author)
★★★★★ 33 customer reviews

See all 2 formats and editions

Kindle	Paperback
\$9.96	\$22.49

Read with our free app 12 Used from \$18.96 21 New from \$21.57

Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works—it teaches you exactly what you need to do now to obtain the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality.

Frequently Bought Together

Price for all three: **\$87.33**

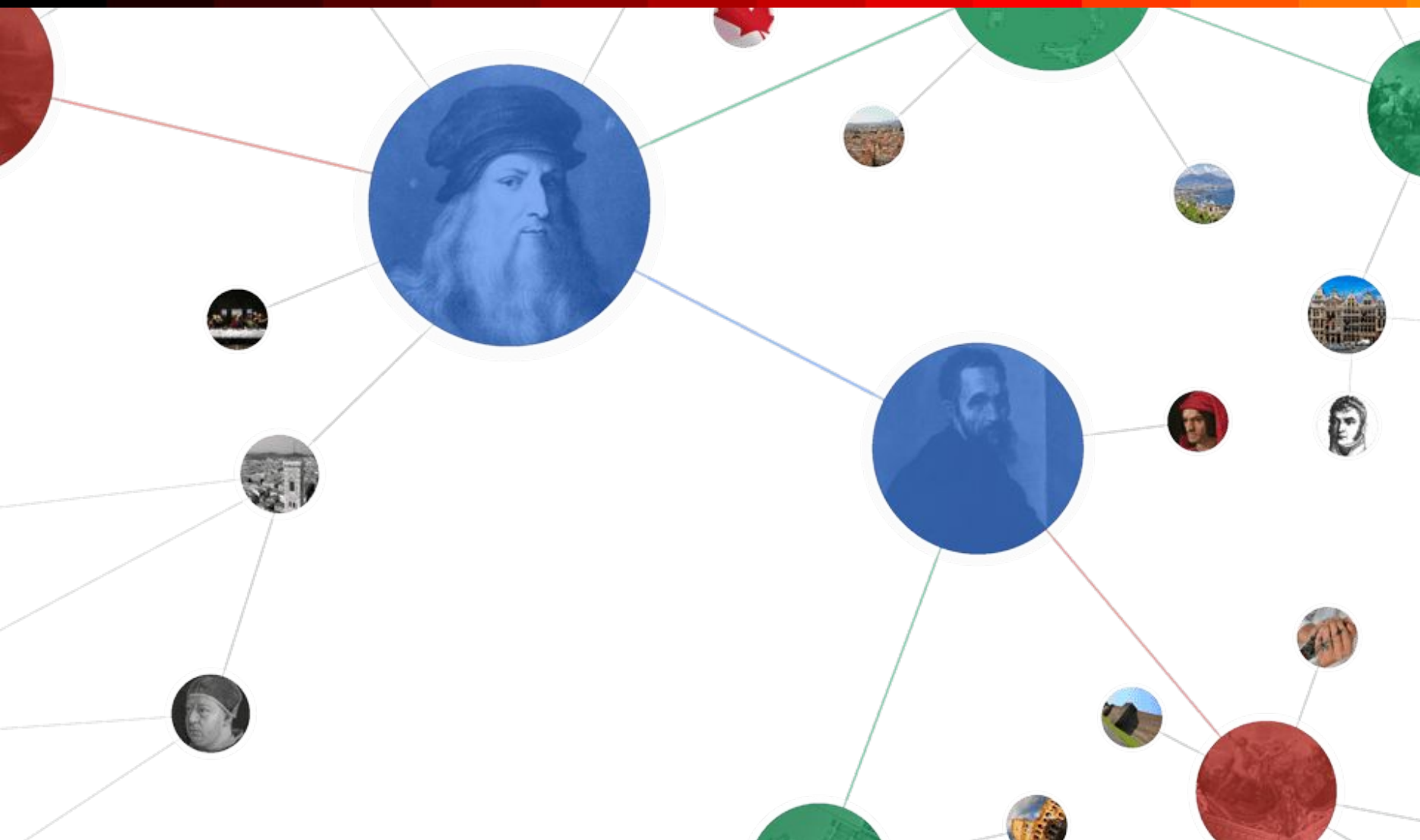
Add all three to Cart
Add all three to Wish List
Show availability and shipping details

- ☑ This item: **Writing High-Performance .NET Code** by Ben Watson: Paperback **\$22.49**
- ☑ **Adaptive Code via C#**: Agile coding with design patterns and SOLID principles (Developer Reference) by Gary McLean Hall: Paperback **\$34.46**
- ☑ **Microsoft .NET - Architecting Applications for the Enterprise (2nd Edition)** (Developer Reference) by Dino Esposito: Paperback **\$30.38**

Customers Who Bought This Item Also Bought

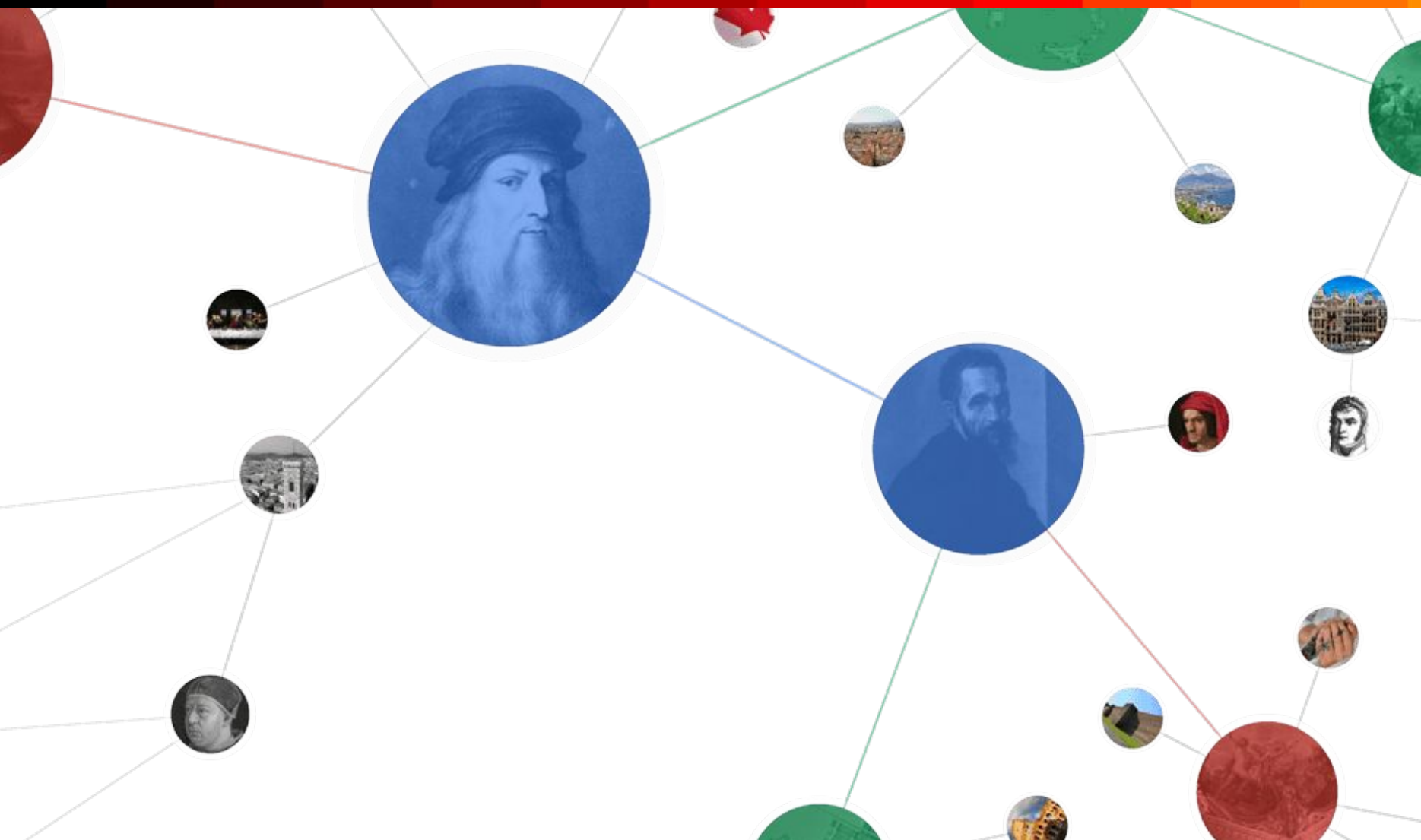
Book Title	Author	Rating	Price
Microsoft .NET - Architecting Applications for the Enterprise (2nd Edition)	Dino Esposito	★★★★★ 22	Paperback \$30.38 Prime
Adaptive Code via C#	Gary McLean Hall	★★★★★ 42	Paperback \$34.46 Prime
Concurrency in C# Cookbook	Stephen Cleary	★★★★★ 12	Paperback \$36.97 Prime
C# in Depth, 3rd Edition	Jon Skeet	★★★★★ 40	Paperback \$33.38 Prime
Pro C# 5.0 and the .NET 4.5 Framework	Andrew Troelsen	★★★★★ 97	Paperback \$41.86 Prime
C# Design Pattern Essentials	Tony Bevis	★★★★★ 35	Paperback \$28.36 Prime
ASP.NET Web API 2: Building a REST API	Jamie Kurtz	★★★★★ 14	Paperback \$29.99 Prime

New York Times - apple



- Jabłko
 - Gatunki
 - Właściwości odżywcze
 - Dane handlowe
 - Firmy handlujące
- Apple inc.
 - Pracownicy
 - Firmy powiązane
 - Produkty
 - Tim Cook
- Film (y)
- Odcinek Star Trek
- Apple Records
- Rosyjska partia
- ...

New York Times



Spock

Fictional Character

Spock, commonly Mr. Spock, is a fictional character in the Star Trek media franchise. Spock was first portrayed by Leonard Nimoy in the original Star Trek series, and also appears in the animated Star ...
[Wikipedia](#)

Played by: [Leonard Nimoy](#)

Species: [Vulcan](#)

Postings: [USS Enterprise \(NCC-1701\)](#), [USS Enterprise \(NCC-1701-A\)](#)

Rank: [Captain](#)

Occupations: [Federation ambassador](#)

Affiliations: [Starfleet](#), [United Federation of Planets](#)

People also search for

[View 15+ more](#)



[James T. Kirk](#)



[Uhura](#)



[Leonard McCoy](#)



[Scotty](#)



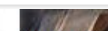
[Hikaru Sulu](#)



[Carl Friedrich](#)



[Joseph-Louis Lagrange](#)



[Pierre de Fermat](#)

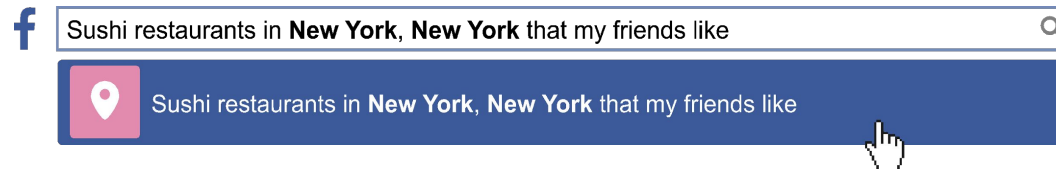


[Augustin-Louis Cauchy](#)

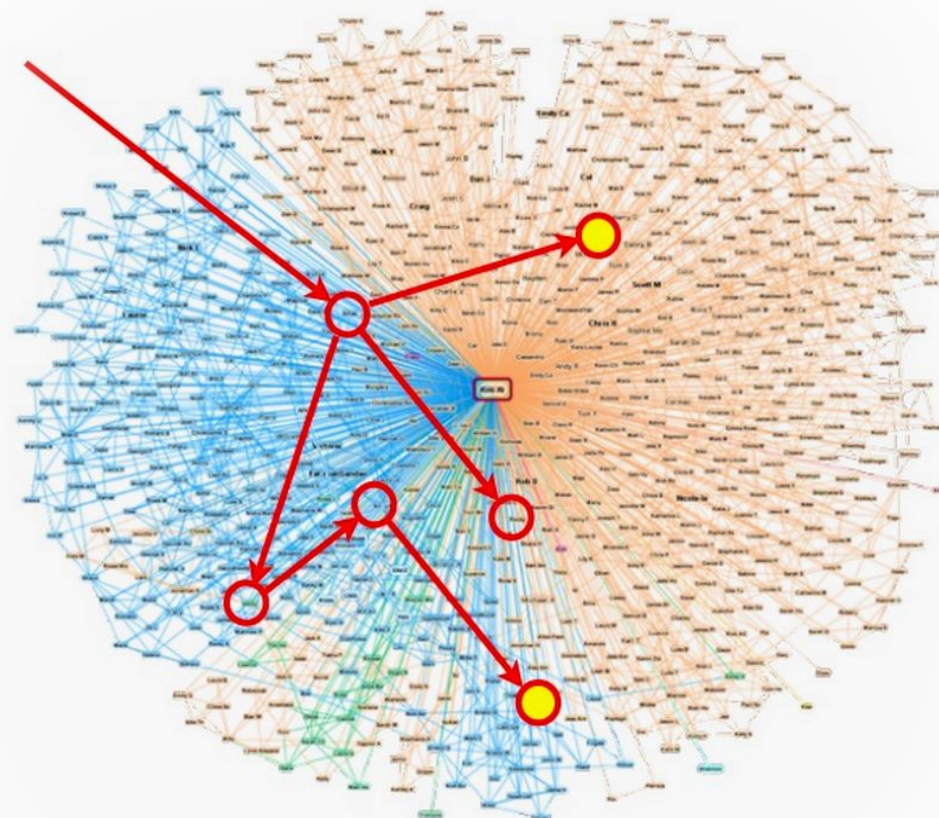
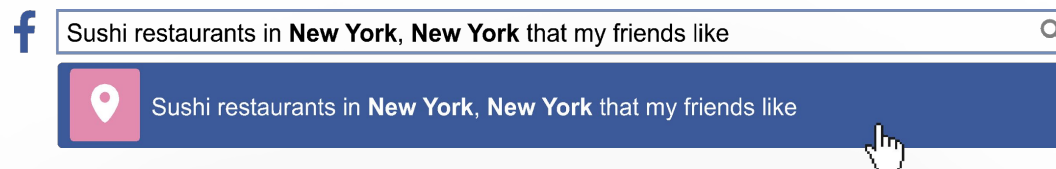


[Isaac Newton](#)

Rozpoznawanie języka naturalnego



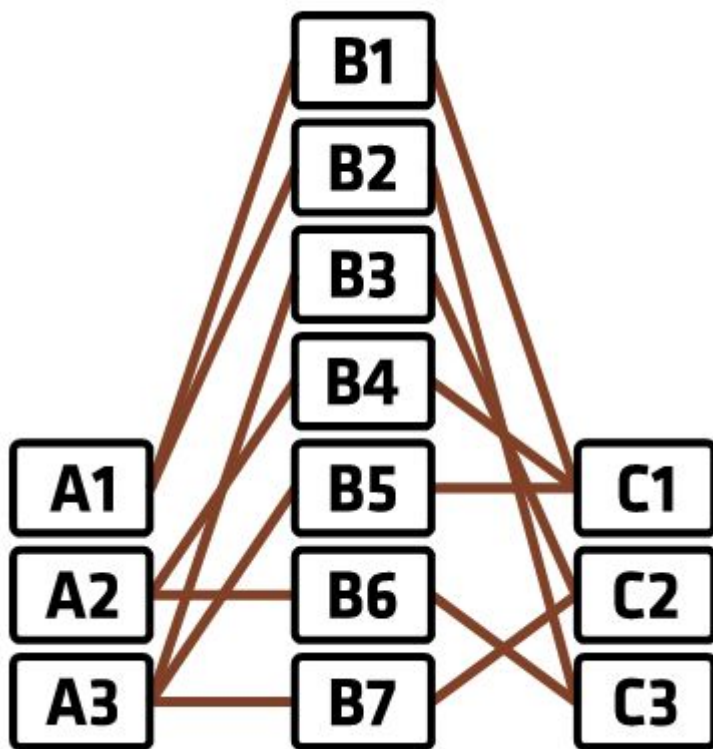
Rozpoznawanie języka naturalnego



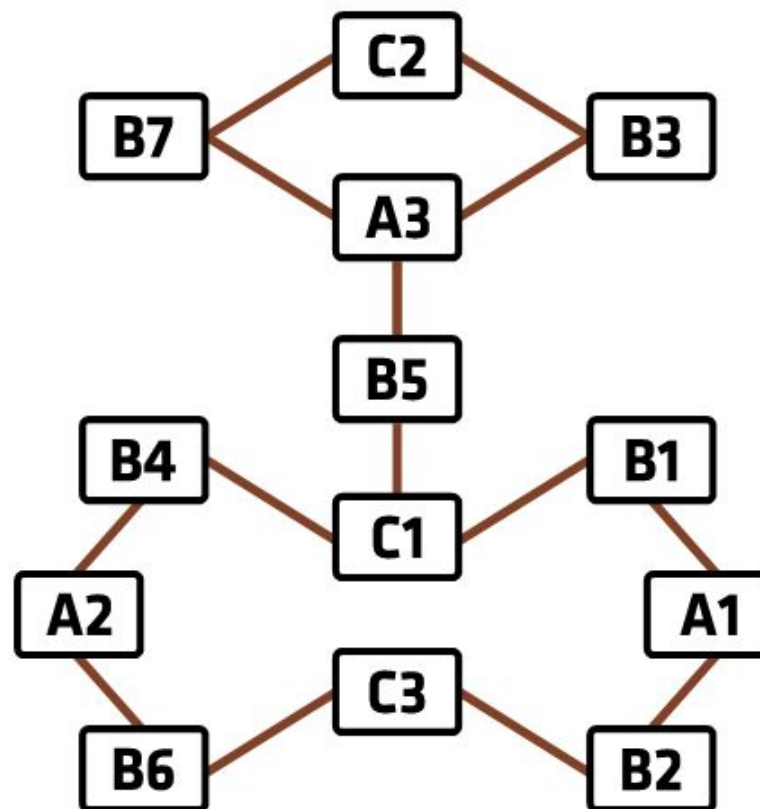
Wydajność

Wydajność – można inaczej

Relacyjne



Grafowe



Wydajność – testy

Twitter z 2009 roku

- 1,67 miliarda obserwowań
- 41,7 milionów userów

Operacje

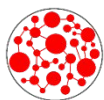
- Tworzenie kont – 1%
- Publikowanie tweetów – 15%
- Oczyt – 76%
- Rekomendacje – 7%

Wydajność – testy



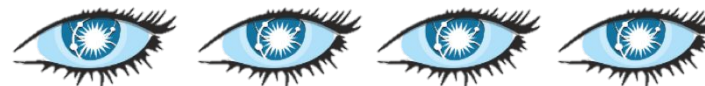
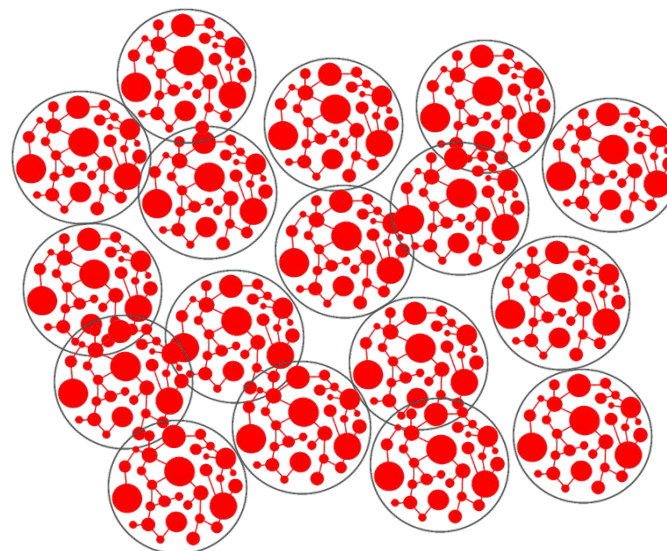
Przechowywanie danych

- 6 c1.4xl Nodów Casandry



Obliczenia

- 40 m1.small
- Obsługa transakcji i zapytań



Wydajność – testy

Typ transakcji	Liczba trx	Ważona czasu	Std czasu
Tworzenie konta	379,019	115,15ms	5,88ms
Tweetowanie	7,580,955	18,54ms	6,34ms
Odczyt	37.936,184	6,29ms	1,62ms
Rekomendacje	3.793,863	67,65ms	13,89ms
Total	49.690,061		
Czas trwania	2,3h		
Średnia	5.900 trx/sek		
Peek	10,300 trx/sek		

The image features the Neo4j logo, which consists of the text "Neo4j" in a white, sans-serif font. The logo is centered horizontally and vertically against a background of vertical stripes. The stripes transition from dark red on the left to bright yellow on the right, passing through orange and red in the middle. The text is positioned over the red and orange sections of the gradient.

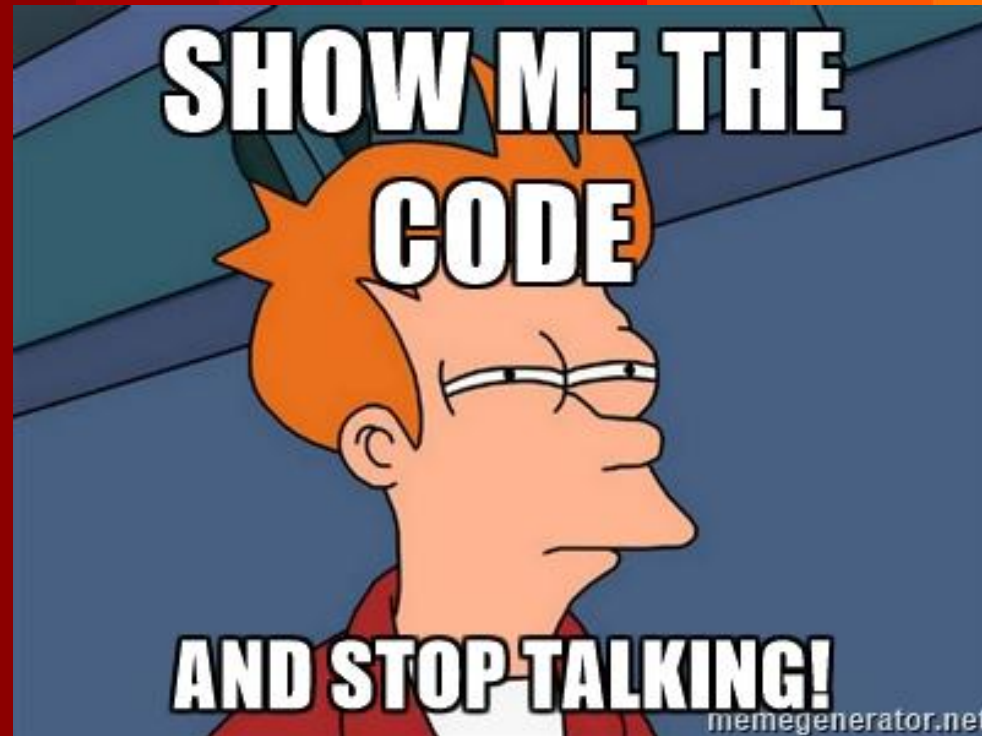
Neo4j

Neo4j



- db-engines.com – 20 pozycja, 1 w bazach grafowych
- Drivery dla:
 - Java
 - .NET
 - JavaScript
 - Python
 - Ruby
 - PHP
 - and R, Go, Clojure, Perl, Haskell
- Cypher jako standard
- Jest wersja darmowa

Demo



Wydajność – testy

Kiedy stosować:

- Dane hierarchiczne
- Ważniejsze są relacje, niż atrybuty obiektów
- Wyszukiwanie wzorów w danych
- Do lepszej wizualizacji danych
- Wyciągnąć sens z danych

Kiedy NIE stosować:

- Systemów gdzie ważniejsze są atrybuty obiektów niż relacje
- Uwaga z ACIDowe transakcje
- **Zamiast baz reakcyjnych**

Pytania?

