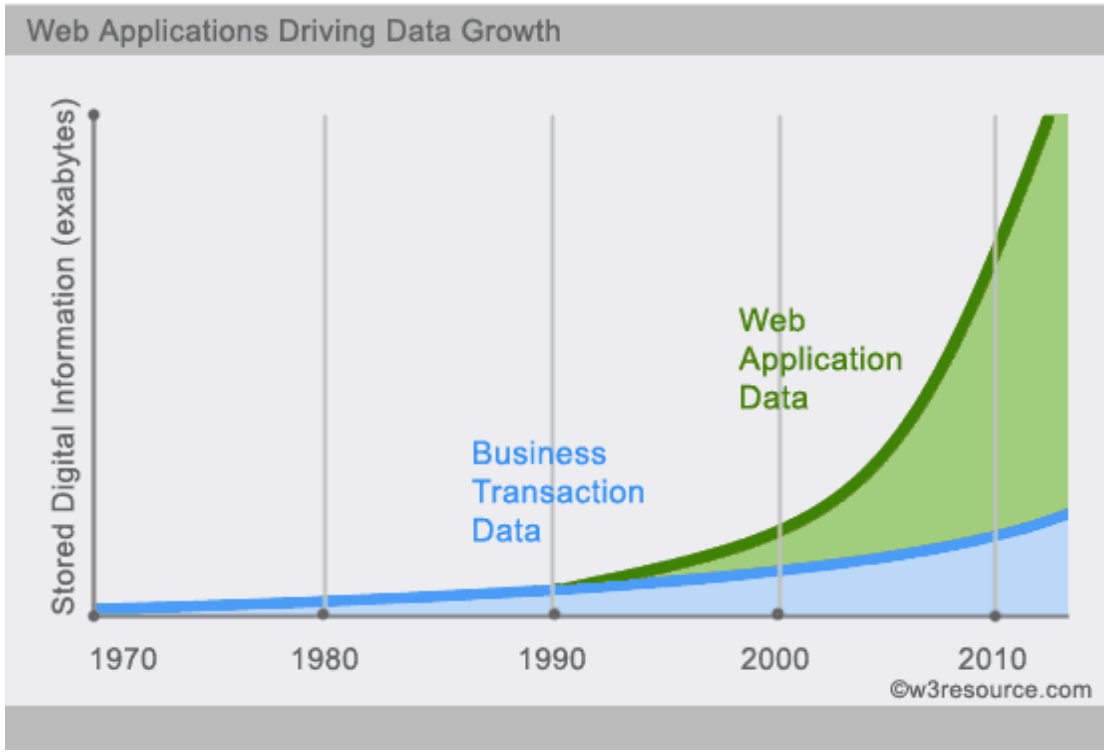
# *Big Data: basisprincipes*

# Lesmateriaal Big Data

- Theorie basisprincipes Big Data en NoSQL
  - Cursus op BB
  - Video's op PluralSight (enkel de opnames waarnaar wordt verwezen vanuit de cursus)

- Praktijk MongoDB
  - Installatiehandleiding en startbestanden op BB
  - Powerpoint op BB
  - Opdrachten op BB

# Inleiding




Relationele databanken: Gegevens

opslaan door bedrijven

Massa andere gegevens niet opgeslagen

in relationele databank:

tweets, facebook, weblogs, feeds, RFID-scans, sensordata, clickstreamdata,…

Nood aan:
- infrastructuur
- nieuwe programmeeromgeving
- nieuwe dataomgeving

# Terminologie

- Database: archief voor dataopslag
  - Opgeslagen gegevens als zodanig
  - Wijze waarop gegevens zijn opgeslagen
  - Software waarmee databases worden aangemaakt en benaderd

- Datawarehouse
  - Gegevensverzameling voor snelle ad-hoc vragen zonder belasting bron
  - Nooit rechtstreeks gegevens toegevoegd/gewijzigd/verwijderd
  - Gegevens worden gebruikt voor BI-doeleinden
  - Voorbeeld controle CV-ketels

# Terminologie

- Datamining
  - Gericht zoeken naar (statistische) verbanden tussen gegevensverzamelingen → patronen (Business Intelligence – BI)
  - Betekenis en inhoud (context) informatie cruciaal
  - Snelheid waarmee bruikbare resultaten worden  bekomen is in realtime-toepassingen zeer belangrijk bv. monitoren en bijsturen van bedrijfsprocessen
  - Doel? Wetenschappelijk, journalistiek, commercieel gebruik
    - Vb verband tussen leeftijd klant en type shampoo

# Big Data - vroeger

‘Big Data’: al in de jaren ’50

- **Aanvang:** Analyses via wiskunde en/of statistiek(manueel)
- **Later:** gebruik van applicaties o.a. spreadsheets en database-toepassingen(o.a.Access)

- **Doel:** beslissingen nemen voor de toekomst =>BI (Business Intelligence)

# Big Data – nu hype

- Voedingsbodem:
  - Hardware mogelijkheden, server
  - Goedkopere en ruimere opslag
  - Mogelijkheden van opensource software
  - Beschikbaarheid massa's gegeven
- Toepassingen:
  - marketing
  - politieonderzoek –en opsporing (fraude, cybercrime)
  - analyses datalekken (bv WikiLeaks, Luxleaks, Panama Papers)
  - onderzoek gezondheidssector (ziektes, erfelijkheid)
  - industrie (bv technologie veiligheid auto's)

# Term Big Data

- Honderden terabytes
- ‘Klassieke’ databank kan gegevens niet aan, alternatief nodig voor niet-relationele gegevens
- 3 V’s: - Volume: niet te verwerken op tradionele manier
  - Velocity: hoge snelheid van toevoer
  - Variety
- 4 V’s, of 5: - Veracity: betrouwbaarheid
  - Value

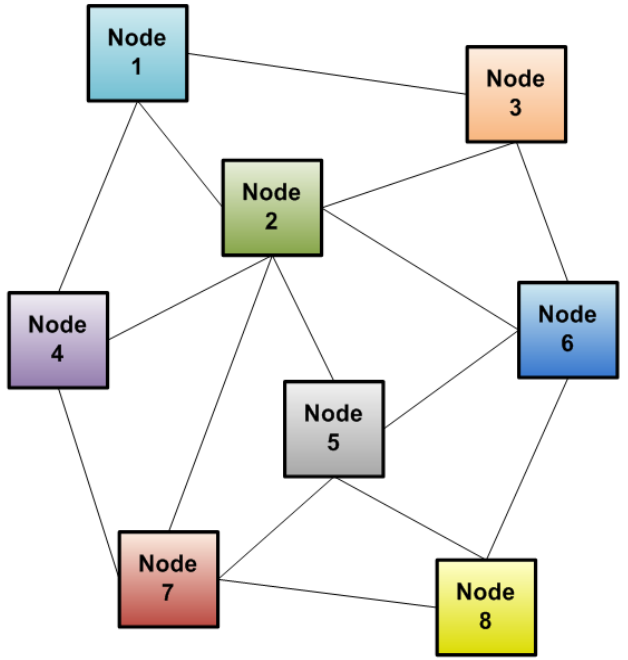
# Hoe werkt Big Data?

- Architectuur
- Distributed System met nodes
- CAP-theorema

# Architectuur

- Architectuur: https://app.pluralsight.com/player?author=ben-sullins&name=data-analytics-hands-on-m9&mode=live&clip=3&course=data-analytics-hands-on

# Distributed system




- Big data → grote hoeveelheden
  → geen structuur

  Gevolg: verwerkingstijd schaalt met hvh informatie
- Hoe verwerking versnellen?
  → snellere server
  → meer servers
  → optimalisering programma's
- Distributed system

  mainframes, workstations, PC's communiceren via netwerk

https://app.pluralsight.com/player?author=ben-sullins&name=data-analytics-hands-on-m9&mode=live&clip=4&course=data-analytics-hands-on

# Distributed datastores

Datastores in een gedistribueerd systeem

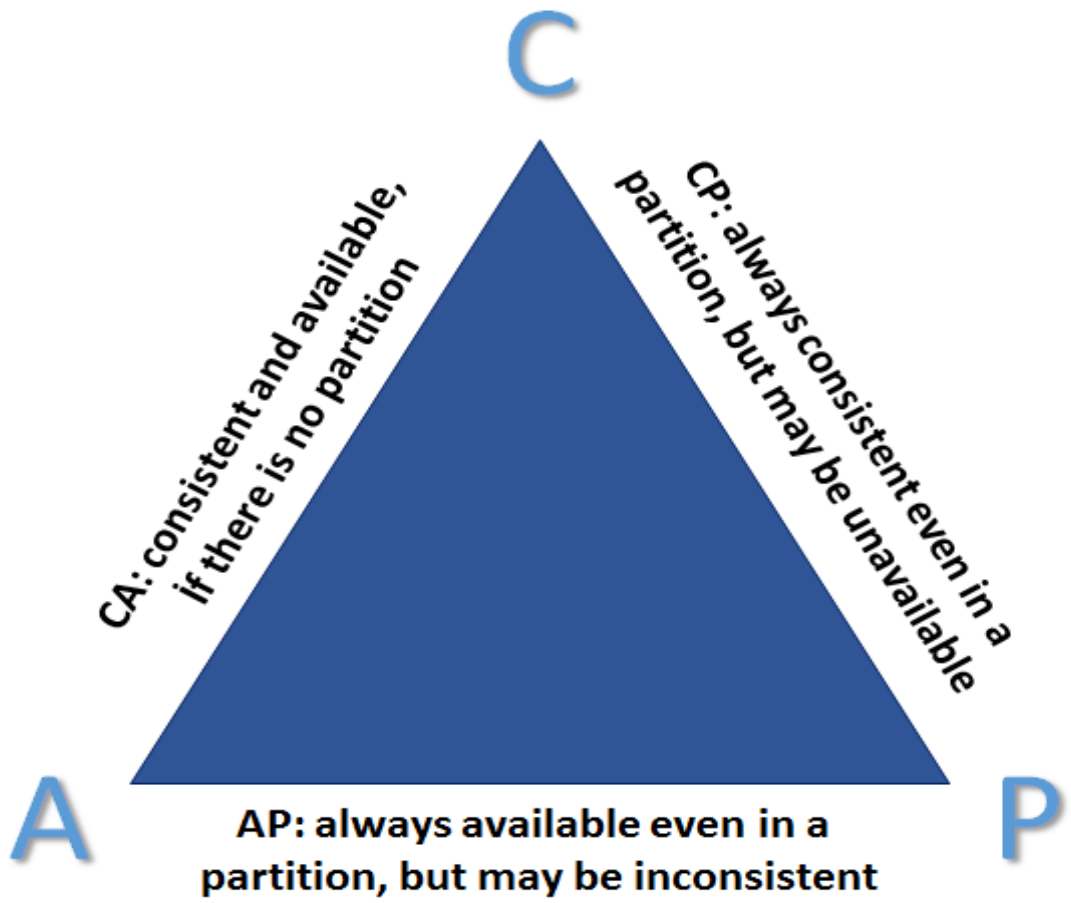
➡ RDBMS komen hiervoor niet in aanmerking, dus NoSQL

Voordelen dergelijk systeem:

- Reliability
- Scalability
- Sharing resources
- Flexibility
- Speed
- Open system
- Performance

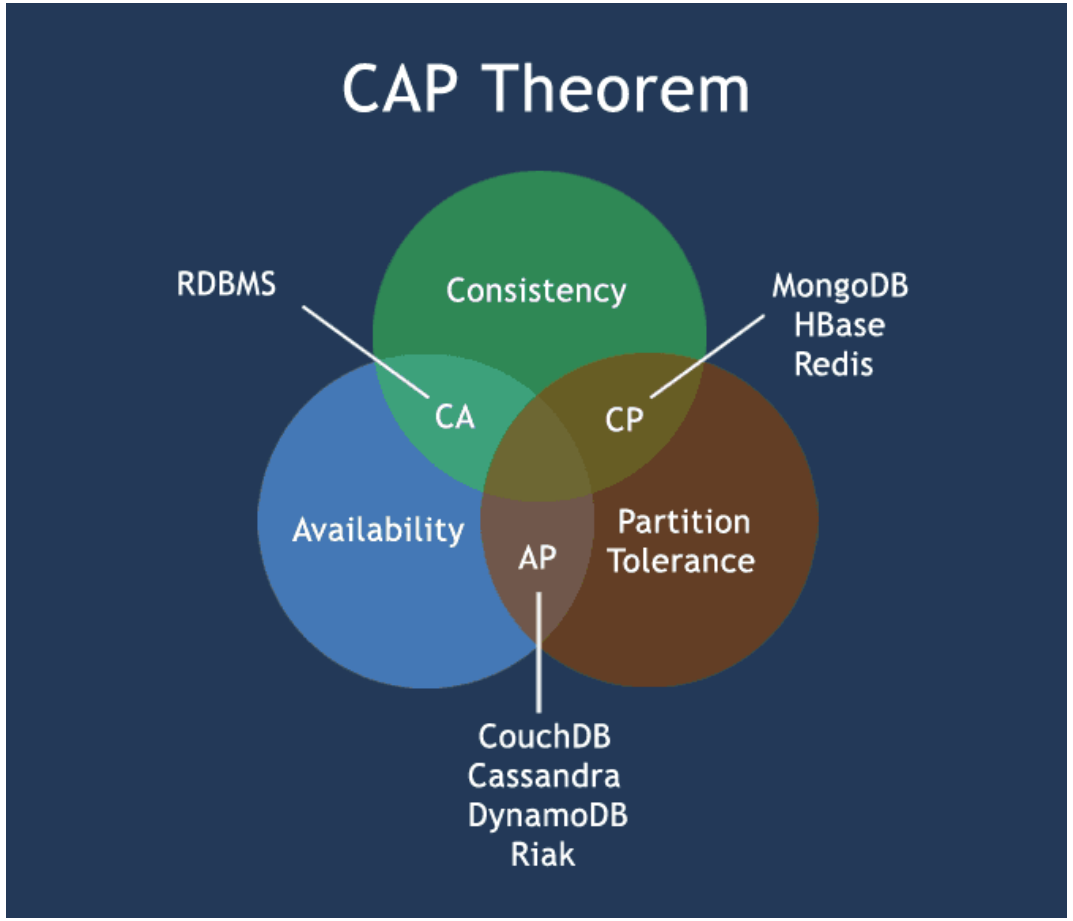
# CAP-theorema

Belang van CAP-stelling: https://app.pluralsight.com/player?author=ben-sullins&name=data-analytics-hands-on-m9&mode=live&clip=5&course=data-analytics-hands-on




- Consistency
- Availability
- Partition tolerance

# CAP-stelling – soorten DB




CA: consistentie en beschikbaar
CP: consistentie, alle data niet direct bereikbaar
AP: beschikbaar, niet altijd volledig

# Database principes

- ACID
- BASE

# Database principe - ACID

- Atomic:
  Elke transactie slaagt volledig, inclusief deelacties, of niet
- Consistent:
  Gegevens mogen niet tegenstrijdig worden.
  Referentiële integriteit.
- Isolated:
  Elke transactie wordt los van andere transactie uitgevoerd.
- Durable:
  Transactie is permanent/onomkeerbaar.

# Database principe - BASE

- Basic Availability
  Beschikbaarheid van data, zelfs met tijdelijke fouten (spreiding gegevens over meerdere opslagsystemen)
- Soft State
  Consistent zijn ligt bij ontwikkelaar, niet bij databank.
- Eventual Consistency
  Uiteindelijk komen tot consistentie, niet meteen - staat haaks op ACID.

# NoSQL DBMS

- niet-relationeel databasemanagement systeem
- distributed data stores met big data
- geen vaste structuren
- vermijdt join-operaties

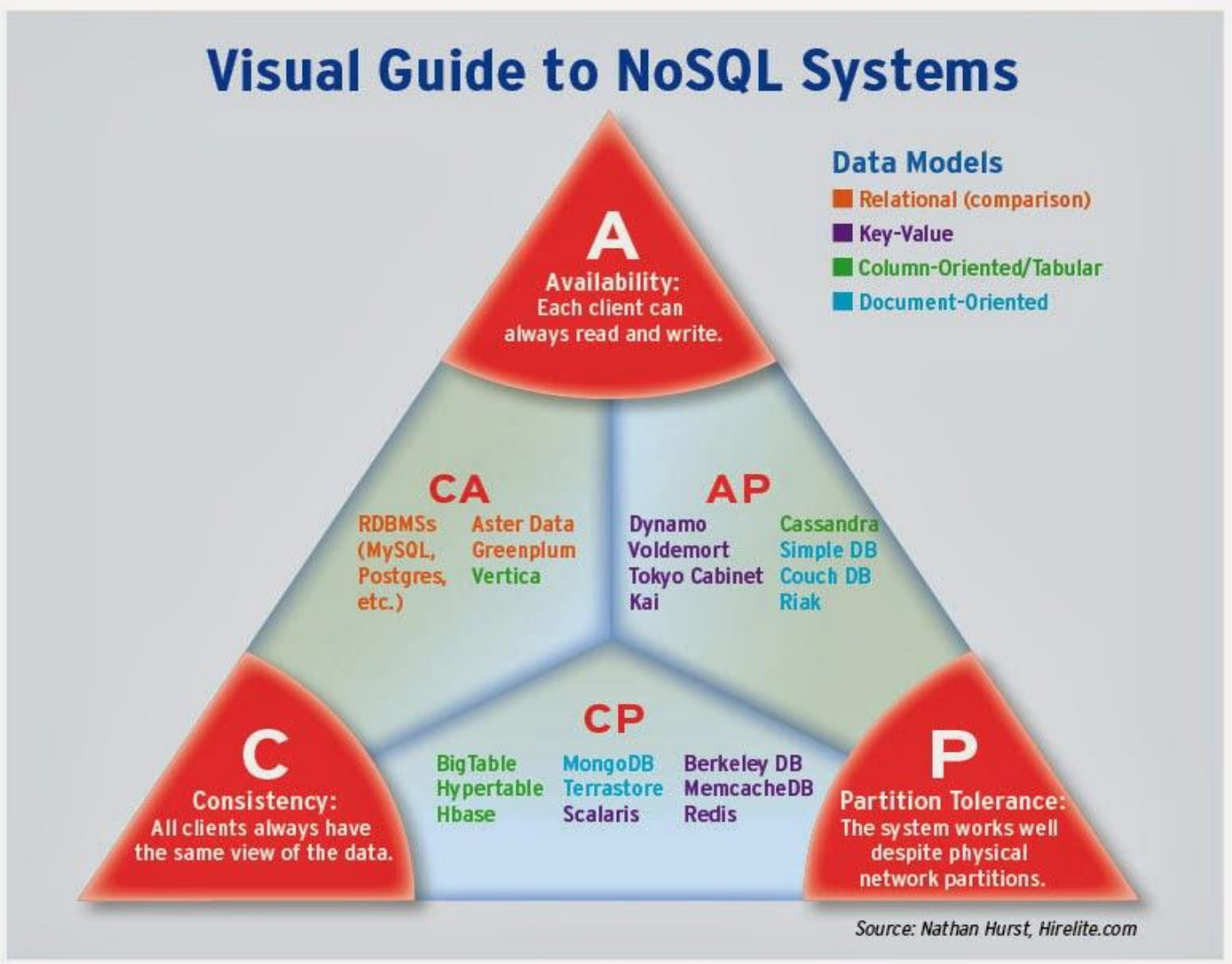
# RDBMS ⇔ NoSQL

| **RDBMS** | **NoSQL** |
|---|---|
| Gestructureerde data | Not Only SQL – ook ongestructureerde data |
| SQL – structured query language | Geen standard query language |
| Data en relaties worden in aparte tabellen opgeslagen | Geen vooraf gedefinieerde structuur |
| DML – data manipulation language<br>DDL – data definition language | Soms onvoorspelbare data |
| Altijd consistent | Eventual consistentie maar wel hoge performantie |
| ACID-transacties | BASE-transacties |

# Voordelen/Nadelen NoSQL

| **Voordelen NoSQL** | **Nadelen NoSQL** |
|---|---|
| Hoge scalability | Geen standaard |
| Distributed computing | Beperkte query mogelijkheden |
| Lagere kost | Eventual consistency is moeilijk programmeerbaar |
| Flexibiliteit in structuur van data | |
| Geen gecompliceerde relaties/joins | |

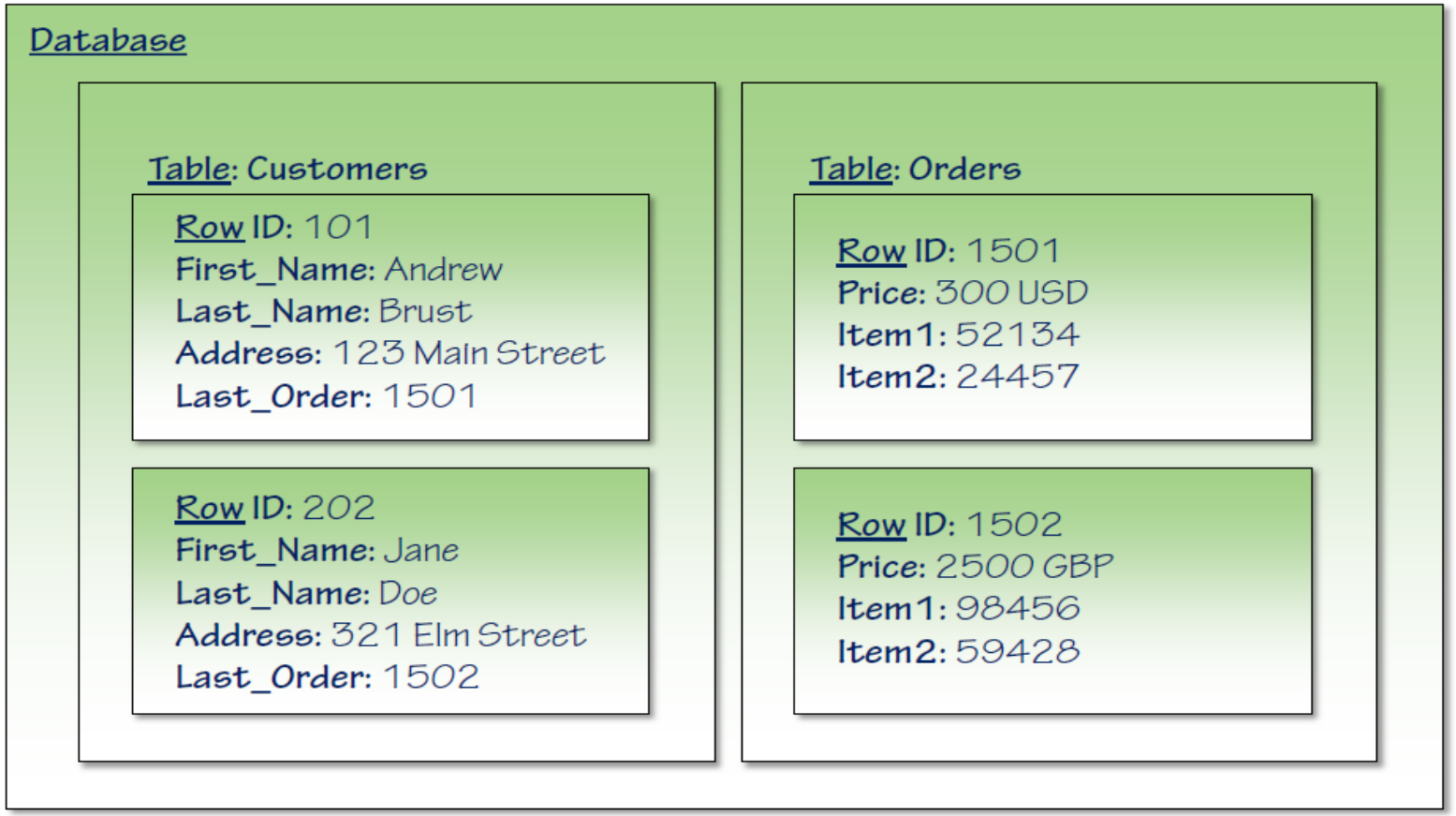
# NoSQL database types



# Database type: Key-value stores

- Meest gebruikte datatype
- Kan vele TB aan gegevens aan
- Laten ongestructueerde gegevens toe
- Makkelijk uitbreidbaar
- Gegevens opgeslagen als hashtable
  elke key uniek, value kan string, JSON-object, BLOB-object,.. zijn
- Key-value pair kan bestaan uit naam gecombineerd met waarde
- Beperking: je kan enkel zoeken via key!

# Key-Value Stores




Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m1-tech-breakdown&clip=0&mode=live&start=313.083998&noteid=b4515986-a34e-4593-a9e1-ce2874bf44b9

# Vergelijking RDBMS - column-oriented store

RDBMS → tabel met rijen en kolommen

Opslag geserialiseerd op rij:

```
001:10,Smith,Joe,40000;
002:12,Jones,Mary,50000;
003:11,Johnson,Cathy,44000;
004:22,Jones,Bob,55000;
```

| RowId | EmpId | Lastname | Firstname | Salary |
|---|---|---|---|---|
| 001 | 10 | Smith | Joe | 40000 |
| 002 | 12 | Jones | Mary | 50000 |
| 003 | 11 | Johnson | Cathy | 44000 |
| 004 | 22 | Jones | Bob | 55000 |

# Vergelijking RDBMS - column-oriented store

Column-oriented geserialiseerd op kolomwaarden:

```
10:001,12:002,11:003,22:004;
Smith:001,Jones:002,Johnson:003,Jones:004;
Joe:001,Mary:002,Cathy:003,Bob:004;
40000:001,50000:002,44000:003,55000:004;
```

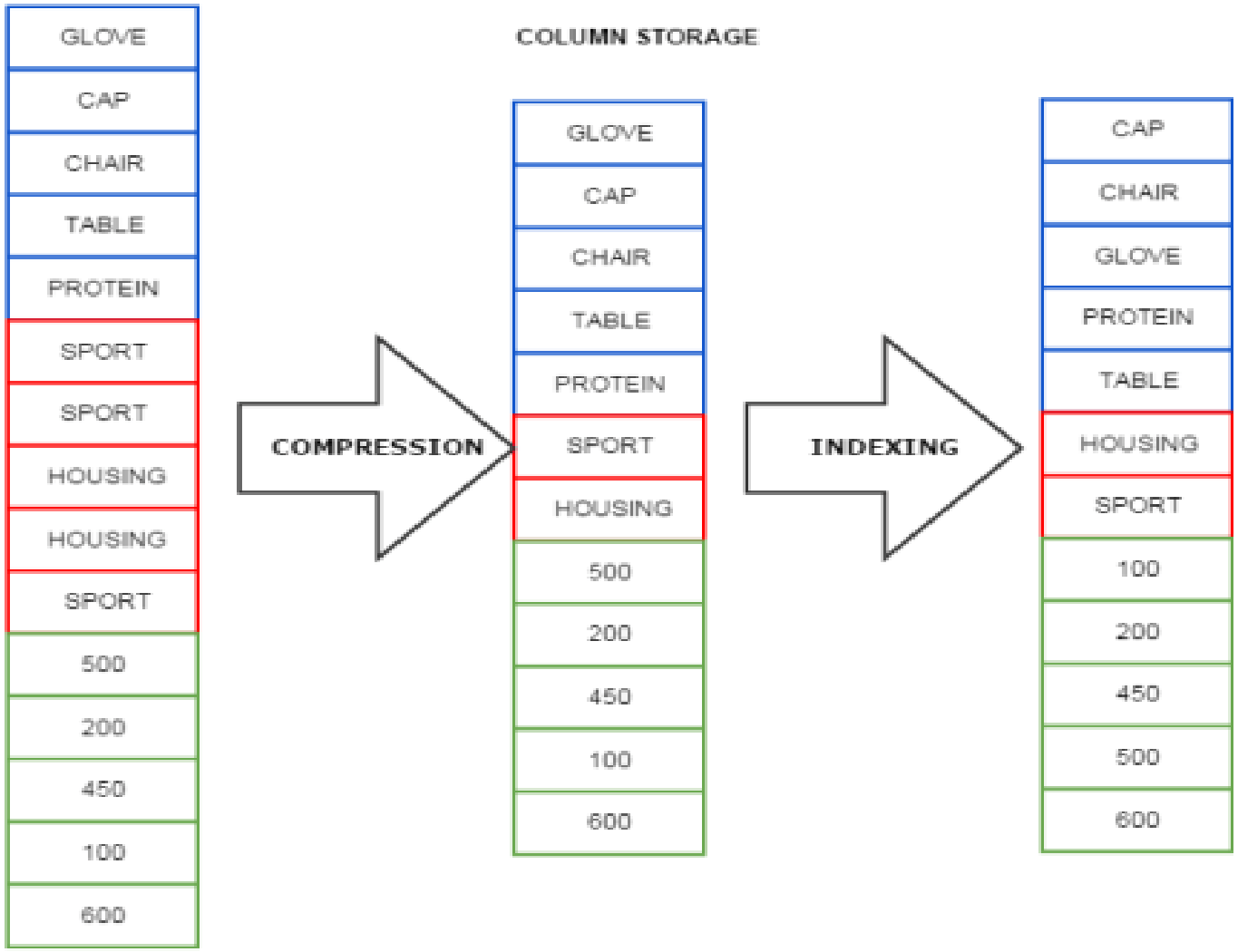
| RowId | EmpId | Lastname | Firstname | Salary |
|---|---|---|---|---|
| 001 | 10 | Smith | Joe | 40000 |
| 002 | 12 | Jones | Mary | 50000 |
| 003 | 11 | Johnson | Cathy | 44000 |
| 004 | 22 | Jones | Bob | 55000 |

…;Smith:001;**Jones:002,004**;Johnson:003;…

LOGICAL TABLE STRUCTURE

| MATERIAL | CATEGORY | REVENUE (EUR) |
|---|---|---|
| GLOVE | SPORT | 500 |
| CAP | SPORT | 200 |
| CHAIR | HOUSING | 450 |
| TABLE | HOUSING | 100 |
| PROTEIN | SPORT | 600 |

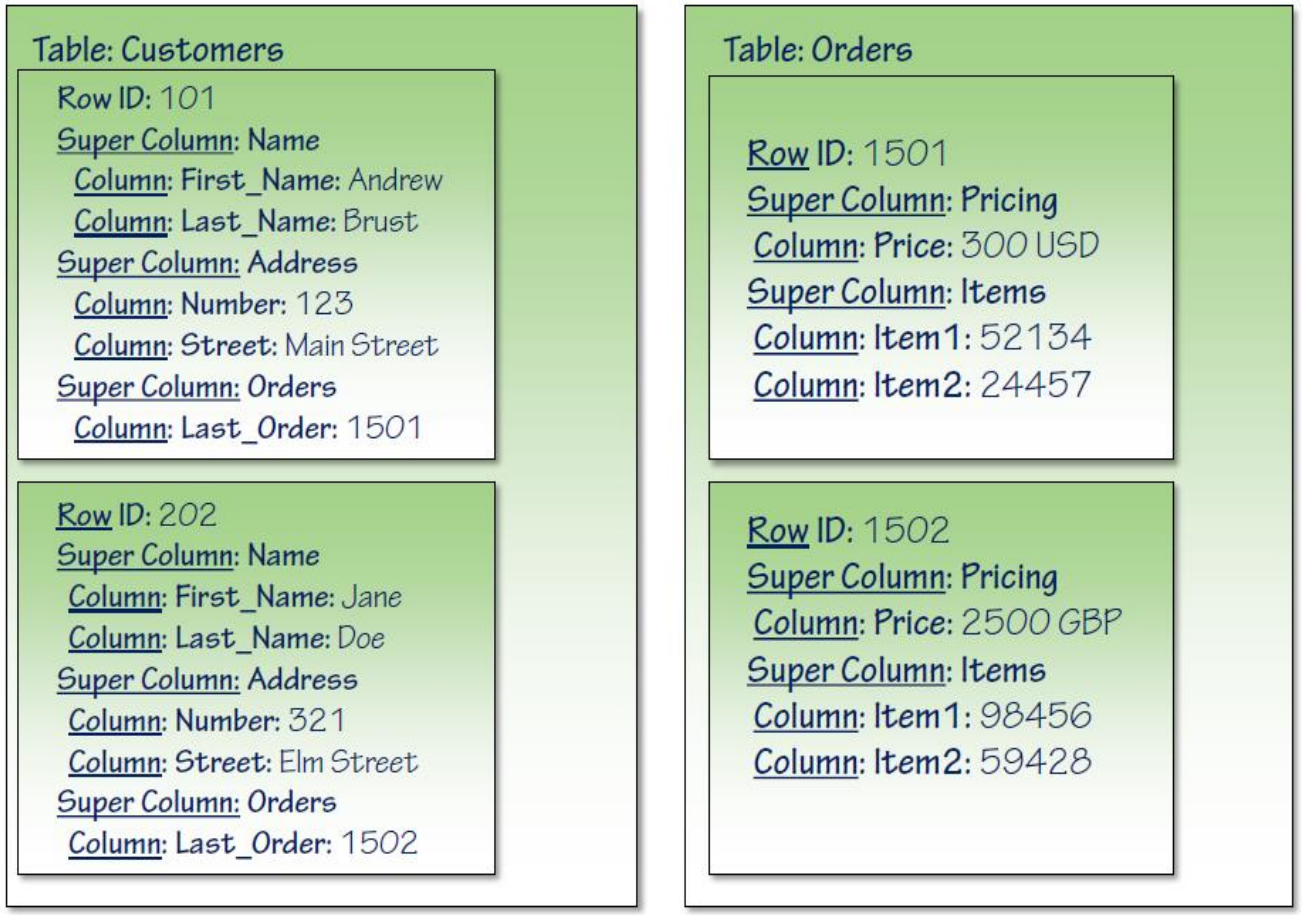
ROW STORAGE


GLOVE
CAP
CHAIR
TABLE
PROTEIN
SPORT
SPORT
HOUSING
HOUSING
SPORT
500
200
450
100
600
COMPRESSION
COLUMN STORAGE
GLOVE
CAP
CHAIR
TABLE
PROTEIN
SPORT
HOUSING
500
200
450
100
600
INDEXING
CAP
CHAIR
GLOVE
PROTEIN
TABLE
HOUSING
SPORT
100
200
450
500
600

# Column-oriented store

- Werken met kolommen
- Slaan values kolom aaneengesloten op
- Kolomgegevens in specifieke files
- Keys verwijzen naar verschillende kolommen
- Queries mogelijk
- Data in kolomfile → zelfde type → gemakkelijke compressie
- Hoge performantie bij gewone queries en groepsqueries → zeer geschikt voor BI en CRM
- Vb: Hbase Cassandra, SimpleDB, SAP HANA

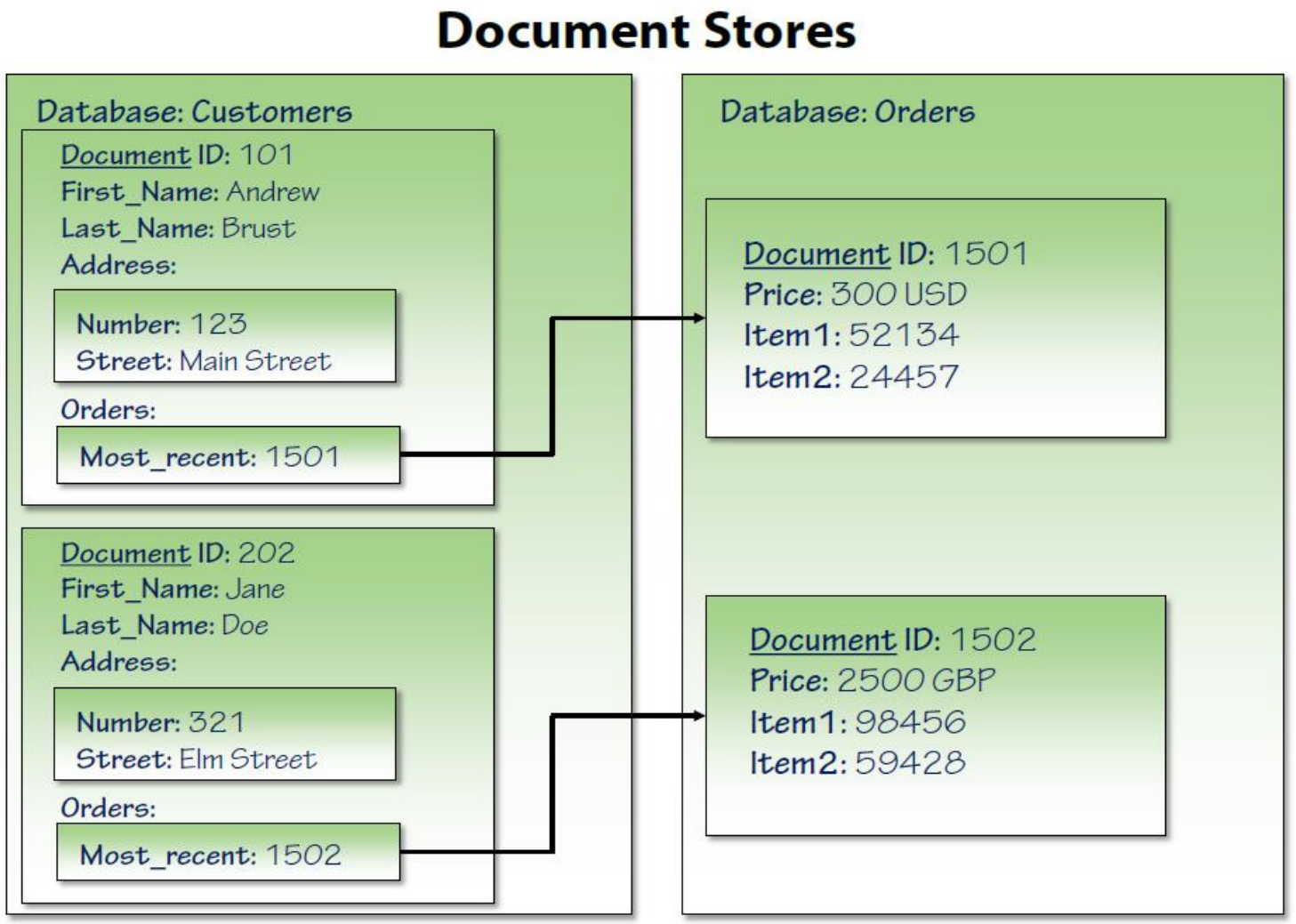
# Wide Column Stores




Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m1-tech-breakdown&clip=3&mode=live&start=191.175816&noteid=5757f4b9-450a-470e-87d2-8a6960d41904

# Documented-oriented store

- Verzameling van documenten
- Data in documenten, key geeft toegang
- Niet noodzakelijk vaste structuur
- Documents → collections: groepering data
  - → verschillende key-value pairs
  - → geneste documenten
- JSON objecten
- Vanuit applicaties verwijzing via URI’s
- Queries mogelijk

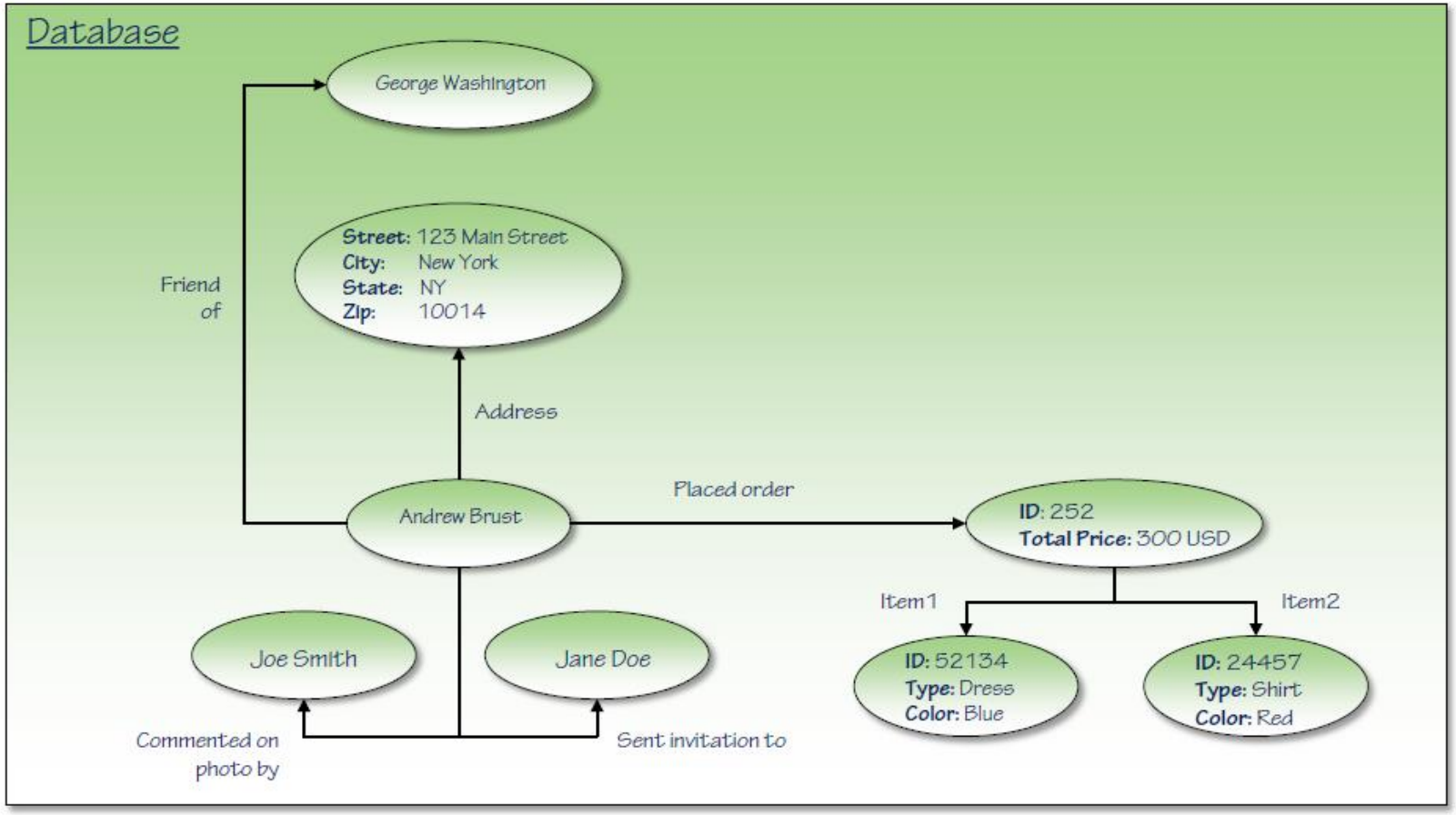



Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m1-tech-breakdown&clip=1&mode=live&start=168.972099&noteid=f255f879-b8c9-48c6-808b-ee9fc52d6191

# Graph store

- Slaan data op in grafiek
- Presentatie zeer toegankelijk
- Verzameling nodes en edges
- Indexen voor opzoeking
- Vb: OrientDB, Neo4J, Apache Giraph

# Graph Databases




Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m1-tech-breakdown&clip=4&mode=live&start=77.868176&noteid=fb45d5c9-4e66-4d99-8b73-352b6c0de7e6

# NoSQL, relational, or both?

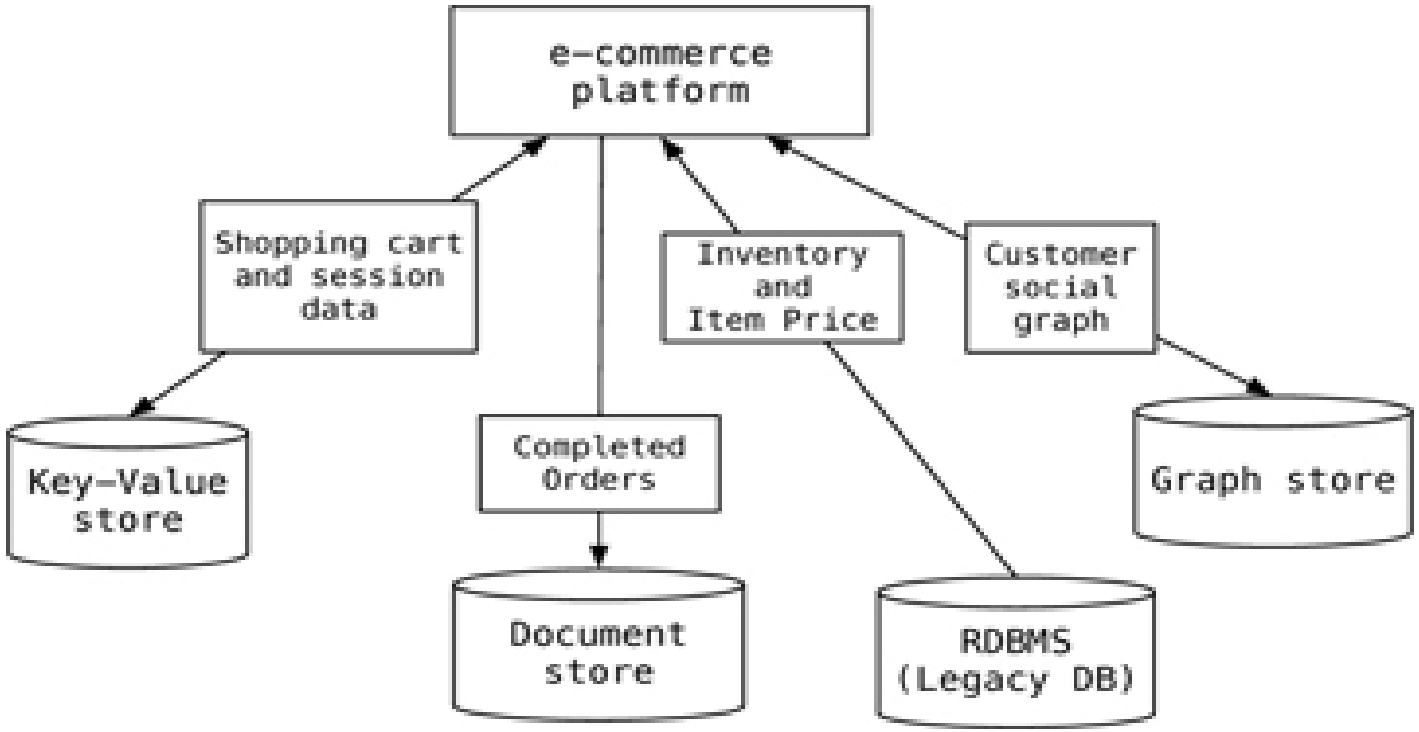



Figure 13.3. *Example implementation of polyglot persistence*

Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m5-both&clip=3&mode=live&start=1.257044&noteid=35a1c93e-d59d-4be8-b130-a9928583f170

# Recommendations

- **Large, public, content-centric properties: NoSQL**
- **Internal, LOB supporting business operations: relational**
- **Investment in RDBMS licenses, infrastructure, skills:**
  - Relational
  - Use both (application-dependent)
  - Use hybrid approaches
- **Productivity**
  - Do cost-benefit analysis
    - How much extra dev time/$$?
    - What is cost of less scalable system?
- **It will be tempting to use one for the other**
  - And it very well may work, but that doesn't make it right