NoSQL Βάσεις Δεδομένων

ΗΜΙΔΟΜΗΜΕΝΑ ΔΕΔΟΜΕΝΑ (SEMI-STRUCTURED DATA)

Ημιδομημένα Δεδομένα

- Δεδομένα που δεν ακολουθούν ένα σχεσιακό μοντέλο, αλλά έχουν κάποια δομή.
- Π.χ.
  - E-mails
  - XML & JSON
  - TCP/IP packets
  - Integration of data from different sources
  - Web pages

Ημιδομημένο μοντέλο δεδομένων (semi – structured data model)

- Ένα μοντέλο δεδομένο που:
  - βασίζεται σε γράφους με ετικέτες για την αναπαράσταση κανονικών και ακανόνιστων δεδομένων
  - οι ετικέτες αναπαριστούν τη δομική πληροφορία των δεδομένων
  - γλώσσες επερωτήσεων για την προσπέλαση και ενημέρωση των δεδομένων

Ημιδομημένα Δεδομένα - Βασικές Ιδέες

–Αυτό-περιγραφικά δεδομένα (self-describing data).

- Το περιεχόμενο έρχεται με τη δική του περιγραφή
- Σε αντίθεση με το σχεσιακό μοντέλο, οπού το σχήμα και το περιεχόμενα αναπαριστώνται ξεχωριστά

–Ευέλικτη τυποποίηση (flexible typing).

- Τα δεδομένα μπορεί να έχουν τύπο (π.χ., «οι κόμβοι είναι ακέραιες τιμές» ή «αυτό το μέρος του γράφου είναι συμβατό με αυτήν την περιγραφή»), ή μπορεί να μην έχουν καθόλου τύπο ή να έχουν κάποιο πολύ ευέλικτο τύπο

– Σειριοποιημένη μορφή (serialization).

- Η αναπαράσταση του γράφου συσχετίζεται με μια σειριοποιημένη μορφή, κατάλληλη για ανταλλαγές σε ετερογενή περιβάλλοντα

Αυτό-περιγραφικά δεδομένα

- Σημείο εκκίνησης: λίστες συσχέτισης, δλδ., εγγραφές από ζεύγη ετικέτας-τιμής.

{name: “Alan”, tel: 2157786, email: “agb@abc.com”}

- Φυσική επέκταση: οι τιμές μπορεί να είναι επίσης άλλες δομές:

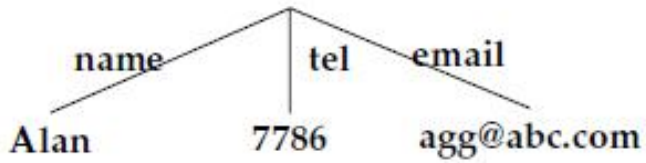
{name: {first: “Alan”, last: “Black”},
tel: 2157786,
email: “agb@abc.com”}

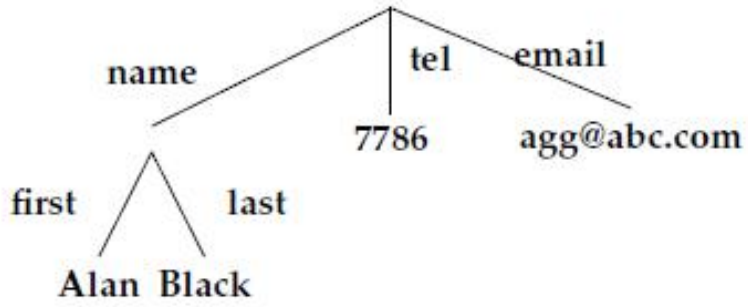
- Περαιτέρω επέκταση: επιτρέπονται διπλές ετικέτες

{name: “alan”, tel: 2157786, tel: 2498762}

Δενδρική Αναπαράσταση

- Τα δεδομένα μπορούν να αναπαρασταθούν γραφικά ως δέντρα:
  - Η δομή των ετικετών αναπαριστάται από τις ακμές του δέντρου
  - Οι τιμές βρίσκονται στα φύλλα



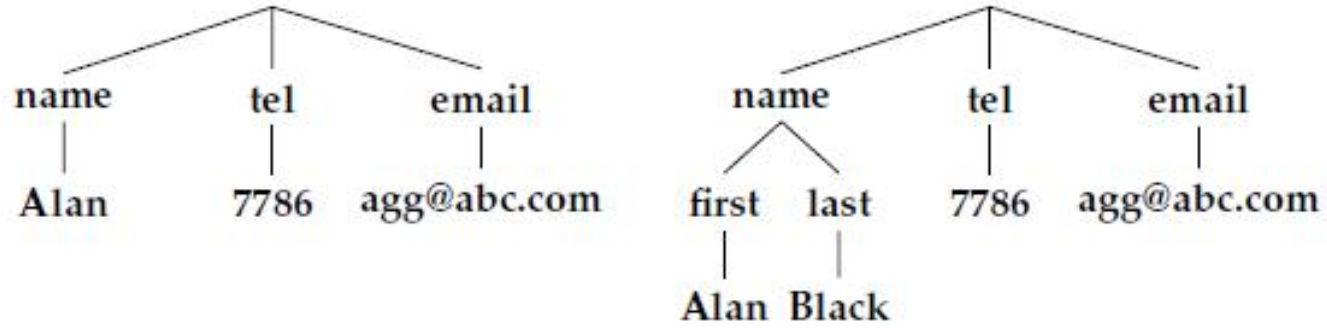



```
{name: "Alan", tel: 2157786,  email: "agb@abc.com"}
```

```
{name: {first: "Alan",  last: "Black"},
 tel: 2157786,
 email: "agb@abc.com"}
```

Δενδρική Αναπαράσταση

- Μια άλλη επιλογή είναι να αναπαριστώνται και οι ετικέτες και οι τιμές σαν κόμβοι του δέντρου



- Παρατήρηση:

Το μοντέλο δεδομένων της XML υιοθετεί αυτήν την αναπαράσταση.

Αναπαράσταση των κανονικών δεδομένων

- Η σύνταξη των ημιδομημένων δεδομένων διευκολύνει την περιγραφή συνόλων εγγραφών ως εξής:

```
{ person:  {name: "Alan", phone: 3127786, email: "alan@abc.com"},
  person:  {name: "Sara", phone: 2136877, email: "sara@xyz.edu"},
  person:  {name: "Fred", phone: 7786312, email: "fd@ac.uk"} }
```

- Παρατήρηση:
  - Μπορούμε να αναπαραστήσουμε σχεσιακά δεδομένα
  - Για κανονικά δεδομένα, η ημι-δομημένη αναπαράσταση δεδομένων εισάγει μεγάλο πλεονασμό

Αναπαράσταση ακανόνιστων δεδομένων

- Πολλές δυνατές διαφοροποιήσεις στη δομή:
  - Τιμές που λείπουν
  - Διπλότυπα πεδία
  - Αλλαγές στην αναπαράσταση, κ.α.

```
{person: {name: "Alan", phone: 3127786, email: "agg@abc.com"},
 person: &314
         {name: {first: "Sara", last: "Green"              },
          phone: 2136877,
          email: "sara@math.xyz.edu",
          spouse: &443                                     },
 person: &443
         {name: "Fred", Phone: 7786312, Height: 183,
          spouse: &314                                     }}
```

- Ταυτότητα κόμβου:
  - Οι κόμβοι μπορούν να αναγνωρίζονται και να αναφέρονται με την ταυτότητά τους.
  - Κύκλοι και μοντέλα αντικειμένων μπορούν να περιγράφοντα επίσης.

Ημι-δομημένα Δεδομένα - Ανακεφαλαίωση

- Ευέλικτη τυποποίηση: επιλέγουμε να μην ενδιαφερόμαστε για τους τύπους
- Αυτό-περιγραφικά: σημειώνουμε κάθε δεδομένο με την περιγραφή του (όπως name, phone, κ.τ.λ.)
- Σειριοποιήσιμα δεδομένα: η σειριοποίηση σημαίνει τη μετατροπή των δεδομένων σε ένα ρεύμα από bytes που μπορεί εύκολα να μεταδίδεται στο διαδίκτυο και να ανακατασκευασθεί στον παραλήπτη

- Τα αυτό-περιγραφικά δεδομένα καταλαμβάνουν πολύ χώρο
  - Οι περιγραφές επαναλαμβάνονται σε κάθε δεδομένο
- Κερδίζουμε όμως σε διαλειτουργικότητα

NOSQL


HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"
Leverage the NoSQL boom

Ορισμός

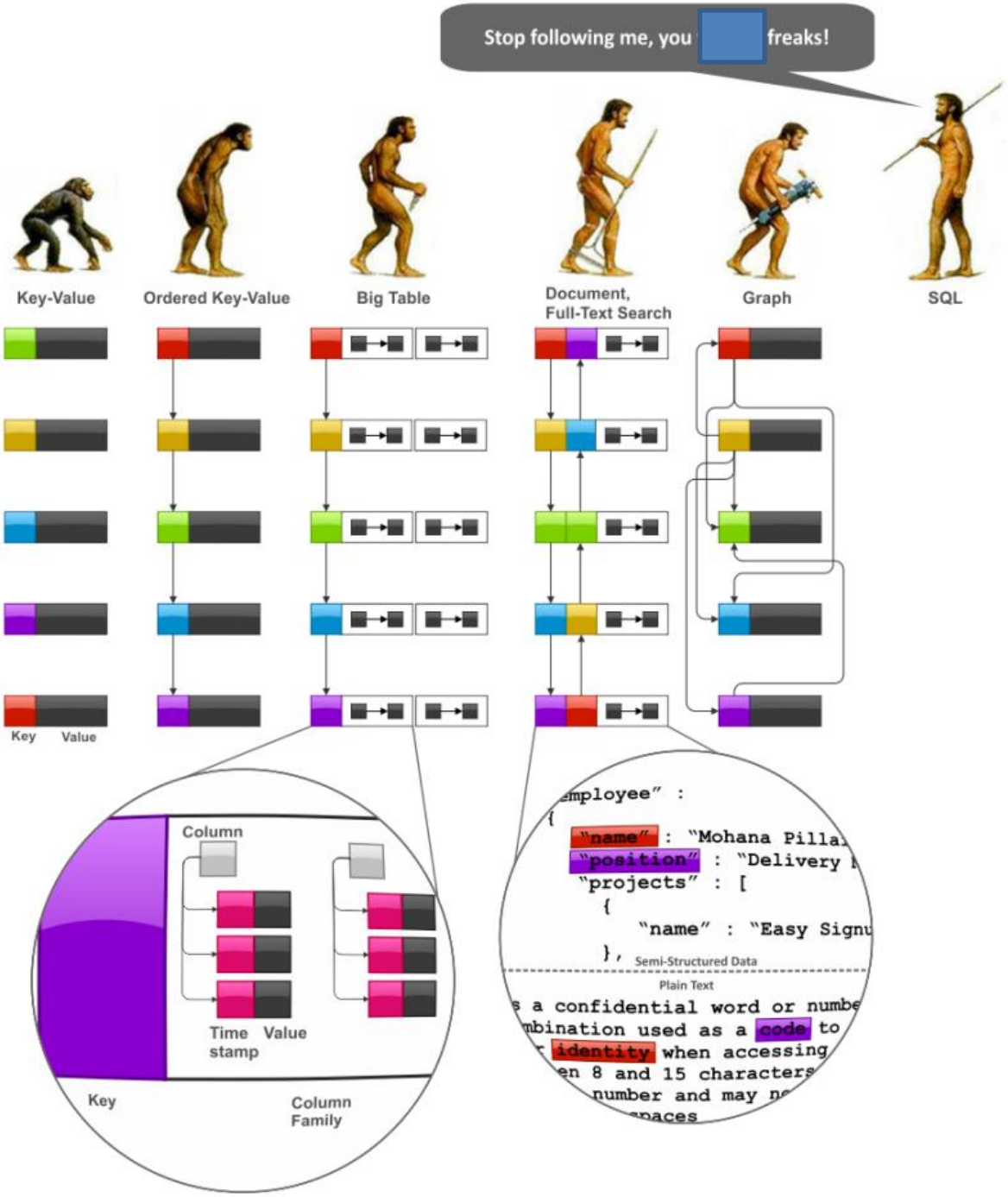
NOSQL DEFINITION:Next Generation Database Management Systems mostly addressing some of the points: being **non-relational, distributed, open-source** and **horizontally scalable**.

The original intention has been modern web-scale database management systems. The movement began early 2009 and is growing rapidly. Often more characteristics apply such as: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge amount of data and more. So the misleading term "nosql" (the community now translates it mostly with "not only sql") should be seen as an alias to something like the definition above.

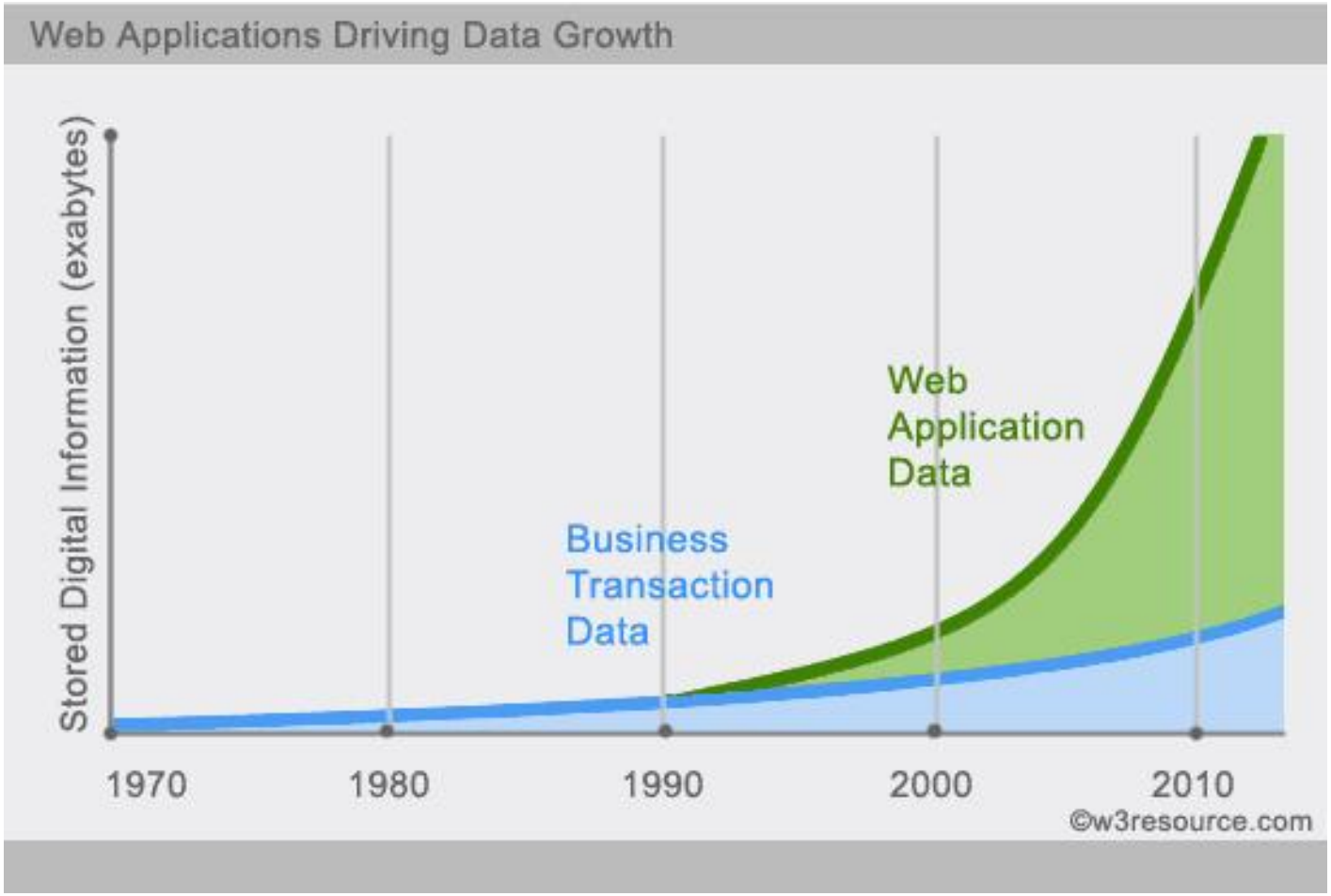
https://hostingdata.co.uk/nosql-database/

NoSQL: Not Only SQL

- NoSQL μη σχεσιακό σύστημα διαχείρισης δεδομένων
- Σχεδιασμένο για κατανεμημένα δεδομένα με απαιτήσεις αποθήκευσης για πολύ μεγάλης κλίμακας δεδομένα (π.χ. Google, Facebook, Twitter)
- Δεν είναι απαραίτητη η ύπαρξη σταθερού σχήματος
- Αποφεύγονται λειτουργίες σύζευξεις (joins)
- Συνήθως οριζόντια κλιμάκωση


Stop following me, you freaks!
Key-Value
Ordered Key-Value
Big Table
Document, Full-Text Search
Graph
SQL
Key
Value
Column
Time stamp
Value
Key
Column Family
"employee" :
"name" : "Mohana Pilla
"position" : "Delivery
"projects" : [
"name" : "Easy Signu
},
Semi-Structured Data
Plain Text
s a confidential word or numbe
mbination used as a code to
identity when accessing
en 8 and 15 characters
number and may ne
spaces

Γιατί NoSQL?


Web Applications Driving Data Growth
Stored Digital Information (exabytes)
Web Application Data
Business Transaction Data
1970
1980
1990
2000
2010
©w3resource.com

Προκλήσεις

- Big Data
  - δεδομένα που συνήθως αποθηκευόνται σε πολλαπλές μονάδες αποθήκευσης
- Δύσκολη η αποδοτική αποθήκευση και επεξεργασία μεγάλων όγκων δεδομένων, ειδικά σε σχέση και με ανοχή σε σφάλματα και λήψη εφεδρικών αντιγράφων
- Ο χειρισμός μεγάλου όγκου δεδομένων απαιτεί την εκτέλεση μεγάλων παράλληλων διεργασιών
- Η διαχείριση συνεχώς εξελισσόμενων σχημάτων και μεταδεδομένων για ημι-δομημένα και αδόμητα δεδομένα είναι επίσης δύσκολη

Πώς αναπτύχθηκαν τα NoSQL συστήματα;

- Έκρηξη ιστότοπων κοινωνικών μέσων με μεγάλες απαιτήσεις στη διαχείριση δεδομένων
- Ανάπτυξη λύσεων στο υπολογιστικό νέφος
- Προτίμηση προς δεδομένα με δυναμική-τυποποίηση και συχνές αλλαγές στο σχήμα τους
- Κοινότητα ανοιχτού λογισμικού
- BigTable (Google)
- Dynamo (Amazon)

RDBMS vs NoSQL

RDBMS

- Σχεσιακά και οργανωμένα δεδομένα
- Δομημένη γλώσσα επερωτήσεων (SQL)
- Τα δεδομένα και οι συσχετίσεις τους αποθηκεύονται σε ξεχωριστούς πίνακες
- Γλώσσα διαχείρισης δεδομένων, γλώσσα ορισμού δεδομένων
- Αυστηρή συνέπεια
- Αφαίρεση από το φυσικό επίπεδο
- Οι εφαρμογές καθορίζουν το “τι”, όχι το “πώς”

NoSQL

- Σημαίνει Not Only SQL
- Δεν υπάρχει δηλωτική γλώσσα επερωτήσεων
- Δεν υπάρχει προκαθορισμένο σχήμα
- Key-Value pair storage, Column Store, Document Store, Graph databases
- Τελική συνέπεια (eventual consistency) και όχι ACID ιδιότητες
- Αδόμητα και απρόβλεπτα δεδομένα
- Περισσότερες ερωτήσεις, λίγες ενημερώσεις
- CAP Theorem
- Προτεραιότητα η υψηλή απόδοση, διαθεσιμότητα και κλιμακωσιμότητα
- Συναλλαγές BASE
- Ανάπτυξη ανοιχτού λογισμικού

Δυνατότητες κλιμάκωσης

- Τα σχεσιακά ΣΔΒΔ δεν σχεδιάστηκαν για να είναι κατανεμημένα
- Τα παραδοσιακά ΣΔΒΔ τρέχουν καλύτερα σε μία μηχανή
  - Για την κλιμάκωση (χειρισμό περισσότερων δεδομένων/πράξεων) απαιτείται αναβάθμιση διακομιστή με ταχύτερη CPU, περισσότερη μνήμη -- κάθετη κλιμάκωση
- Τα NoSQL συστήματα έχουν σχεδιαστεί για να εκτελούνται σε clusters (ή γενικότερα σε πολλά μηχανήματα)
  - Για την κλιμάκωση απαιτείται η προσθήκη περισσότερων μηχανών στο cluster -- οριζόντια κλιμάκωση
  - Διάφορες τεχνικές
    - Master-slave αναπαραγωγή
    - Διαμέριση (sharding)

Sharding

- Η τεχνική που χωρίζει μια λογική βάση δεδομένων σε πολλαπλά μέρη ονομάζεται διαμέριση (partitioning).
  - horizontal partioning ή sharding
  - vertical partioning
- Sharding: διαμοιράζει έναν μεγάλο πίνακα σε υποσύνολα γραμμών που μοιράζονται σε πολλαπλούς φυσικούς διακομιστές.
  - το σχήμα επαναλαμβάνεται μαζί με κάθε θραύσμα (shard) του διαμερισμένου πίνακα
  - τα δεδομένα μοιράζονται σε διαφορετικούς διακομιστές με βάση την τιμή ενός κλειδιού

Sharding

- key-based/hash-based
  - static key
  - load balance
- range-based
  - directory
- directory-based
  - ρητή αντιστοίχιση κλειδιών σε shard

Replication

- full replication
- partial replication
  - συχνότητα ανάκτησης
  - χρόνος απόκρισης
- κόστος ενημέρωσης αντιγράφων/συγχρονισμός

Συναλλαγές στα RDBMSs

- Ατομικότητα: απαιτεί κάθε συναλλαγή να είναι “ή όλα ή τίποτα”
- Συνέπεια: εξασφαλίζει ότι κάθε συναλλαγή θα φέρει τη βάση δεδομένων από μία συνεπή κατάσταση σε μια άλλη συνεπή κατάσταση
- Απομόνωση: εξασφαλίζει ότι η ταυτόχρονη εκτέλεση συναλλαγών έχει ως αποτέλεσμα μια κατάσταση συστήματος που θα είχαμε και αν οι συναλλαγές εκτελούταν σειριακά
- Μονιμότητα: σημαίνει ότι αν μια συναλλαγή έχει ολοκληρωθεί (commit), θα παραμείνει έτσι, ακόμα και στο ενδεχόμενο σφάλματος

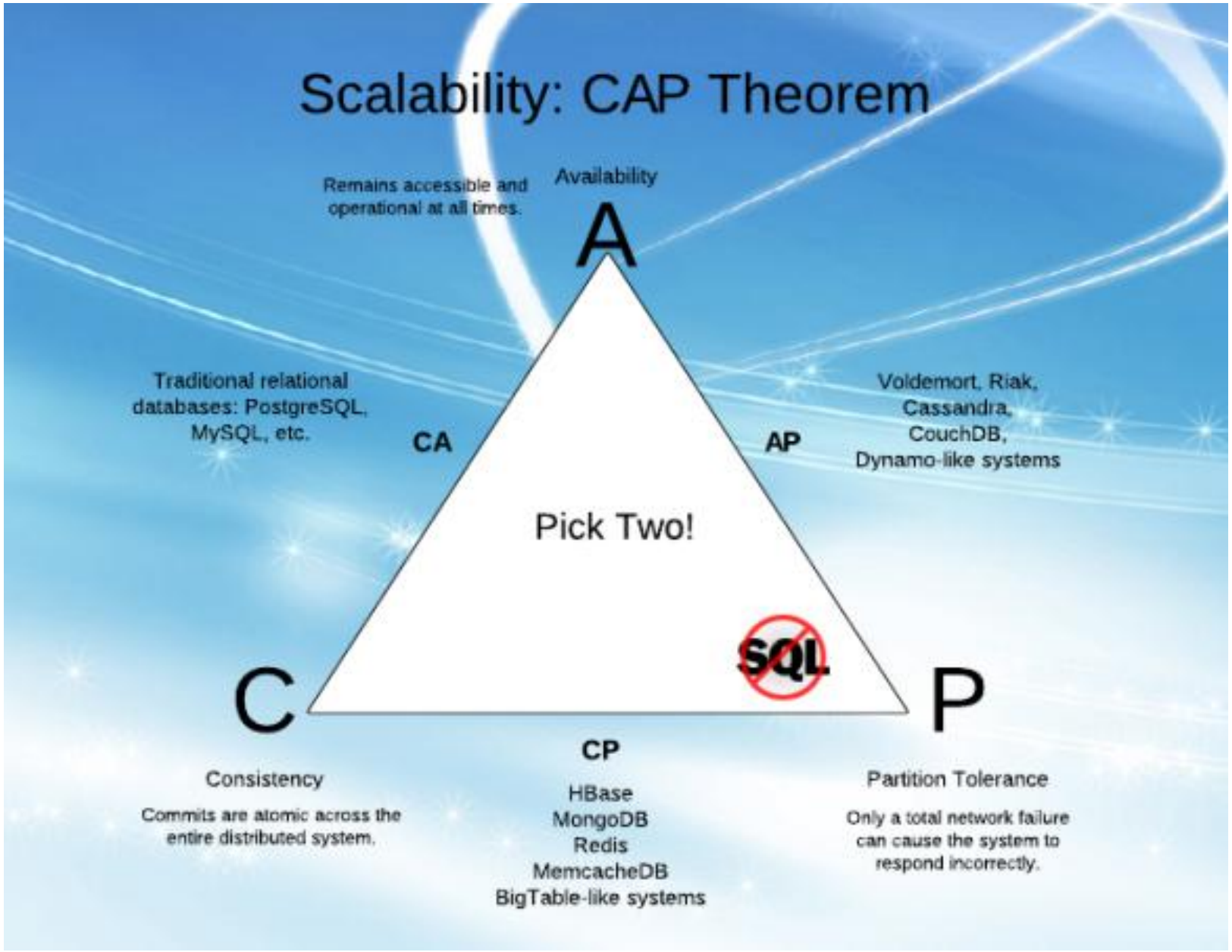
CAP Theorem (Brewer's Theorem)

- Συνέπεια (Consistency): Μια ανάγνωση βλέπει όλες τις προηγούμενες ολοκληρωμένες εγγραφέςs.
- Διαθεσιμότητα (Availability): Οι αναγνώσεις και οι εγγραφές πάντα πετυχαίνουν.
- Ανοχή στις διαμερίσεις (Partition tolerance): Οι εγγυημένες ιδιότητες διατηρούνται ακόμα κι αν σφάλματα στο δίκτυο αποτρέπουν κάποιες από τις μηχανές από το να επικοινωνούν με τις υπόλοιπες.

"Αν και είναι επιθυμητό να έχουμε Συνέπεια, Υψηλή Διαθεσιμότητα και Ανοχή στις Διαμερίσεις σε κάθε σύστημα, δυστυχώς κανένα σύστημα δεν μπορεί να τα πετύχει και τα τρία ταυτόχρονα."

Eventual Consistency

- **Eventual Consistency** or BASE (Basically Available, Soft State, Eventual Consistency)
- Μια μορφή πιο ασθενούς συνέπειας που επιτρέπει την βελτίωση της ταχύτητας και της διαθεσιμότητας
- Υπάρχει μια απεριόριστη καθυστέρηση στην διάδοση μιας αλλαγής που έγινε σε ένα μηχάνημα σε όλα τα άλλα αντίγραφα, το οποίο μπορεί να οδηγήσει σε παρωχημένα δεδομένα (stale data)
- Τελικά (Eventually) οι αλλαγές θα διαδοθούν σε όλες τις υπόλοιπες μηχανές
- Οι εφαρμογές πρέπει να μπορούν να χειρίζονται την πιθανότητα ανάγνωσης παρωχημένων (ασυνεπών) δεδομένων
- Best effort
- Οι προσεγγιστικές απαντήσεις είναι αποδεκτές
- Αισιόδοξη προσέγγιση



https://dzone.com/articles/better-explaining-cap-theorem

NoSQL pros/cons

- **Πλεονεκτήματα** :
  - Υψηλή κλιμακωσιμότητα
  - Κατανεμημένος υπολογισμός
  - Μικρότερο κόστος
  - Ευελιξία σχήματος, ημί-δομημένα δεδομένα
  - Χωρίς πολύπλοκες συσχετίσεις
- **Μειονεκτήματα**
  - Δεν υπάρχει standard
  - Περιορισμένες δυνατότητες επερωτήσεων (μέχρι τώρα)
  - Η τελική συνέπεια δεν είναι διαισθητικά εύκολη στους προγραμματιστές

NoSQL Βάσεις Δεδομένων

- Key-value stores
- Column-oriented
- *Graph-based*
- Document oriented

Key-value stores

- Οι αποθήκες κλειδιού-τιμής (key-value) αποτελούν τον πιο απλό τύπο NoSQL βάσεων δεδομένων
- Σχεδιασμένες για να χειρίζονται τεράστιους όγκους δεδομένων
- Βασίζονται στο άρθρο για το Dynamo της Amazon
- Επιτρέπουν την αποθήκευση δεδομένων χωρίς σχήμα (schema-less data)
- Η βάση αποθηκεύσει τα δεδομένα ως έναν πίνακα κατακερματισμού όπου:
  - κάθε κλειδί είναι μοναδικό, και
  - η τιμή μπορεί να είναι αλφαρηθμιτικό, JSON, BLOB,κτλ
- Ένα κλειδί επίσης μπορεί να είναι αλφαρηθμιτικά, τιμές κατακερματισμού, λίστες, σύνολα, ταξινομημένα σύνολα και οι τιμές αποθηκεύονται έναντι αυτών των κλειδιών
- Για παράδειγμα ένα ζέυγος κλειδιού-τιμής μπορεί να αποτελείται από ένα κλειδί όπως "όνομα" που συσχετίζεται με μια τιμή όπώς "Μαρία".

Key-Value Stores

- SET Employee001:name Giorgos
- SET Employee001:city Thessaloniki
- SET Employee002:name Eleni
- SET Employee002:city Thessaloniki

- GET Employee001:city

- DEL Employee002:city

Key-value stores

- Τα Key-Value stores μπορούν να χρησιμοποιηθούν ως collections, dictionaries, associative arrays κτλ.
- Ακολοθούν τη 'Διαθεσιμότητα' και την 'Ανοχή στις διαμερίσεις' από το CAP θεώρημα.
- Consistent hashing για sharding
- Είναι κατάλληλες για τα περιεχόμενα καλαθιού αγορών, ατομικές τιμές όπως συνδυασμούς χρωμάτων, μια σελίδα ή ένας αριθμός λογαριασμού
- **Παραδείγματα Key-value stores :** Redis (Instagram, StackOverflow), Dynamo (Amazon, Elsevier, IMDb), Riak. etc.


Key - Value Store
Database
Table : supplier
Row
ID : 1
Name : Bob
City : New York
Country : USA
Order_no : ORD-0056
Row
ID : 2
Name : Jack
City : Paris
Country : France
Order_no : ORD-0057
Table : order
Row
Order_ID : ORD-0056
Cost : 250 USD
Item_Qty1 : 2450
Item_Qty2 : 2560
Row
Order_ID : ORD-0057
Cost : 400 USD
Item_Qty1 : 3000
Item_Qty2 : 3530

Column-based Databases (Column-Family Stores)

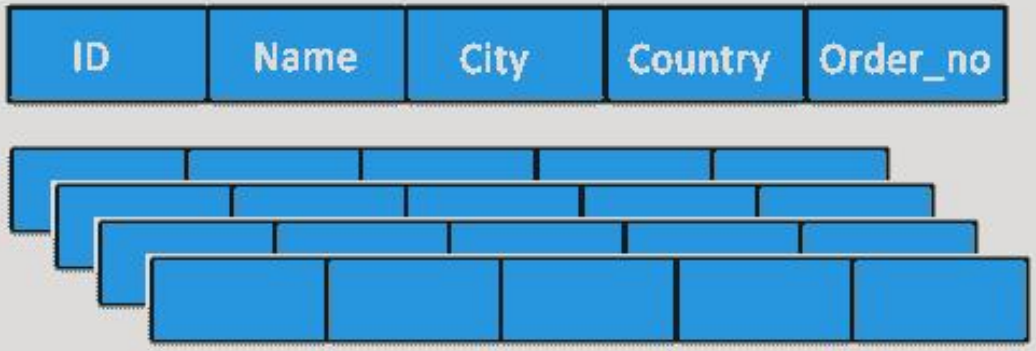
- Επιπλέον δομή σε σχέση με τα key-value stores -> ομαδοποίηση σε στήλες
- Οι βάσεις δεδομένων που προσανατολίζονται σε στήλες κυρίως δουλεύουν πάνω σε στήλες και κάθε στήλη την διαχειρίζονται ατομικά
- Οι τιμές μιας στήλης αποθηκεύονται συνεχόμενα, και αποθηκεύονται σε συγκεκριμένα αρχεία στηλών
- Και η επεξεργασία επερωτημάτων γίνεται πάνω σε στήλες
- Όλα τα δεδομενα εντός κάθε αρχείου στήλης δεδομένων έχουν το ίδιο τύπο, καθιστώντας τα ιδανικά για συμπίεση

Column-Family Stores

- Μπορούν να βελτιώσουν την απόδοση επερωτημάτων αφού μπορούν να προσπελάσουν δεδομένα συγκεκριμένης στήλης
- Υψηλή απόδοση σε ερωτήματα συνάθροισης (π.χ. COUNT, SUM, AVG, MIN, MAX)
- Batch inserts/updates

- Κατάλληλες για αποθήκες δεδομένων και επιχειρηματική ευφυια, CRM, κατάλογοι βιβλιοθηκών κ.α.
- **Παραδείγματα Column-oriented databases :** BigTable (Google...), Cassandra (Facebook, Reddit, Twitter...), SimpleDB etc.

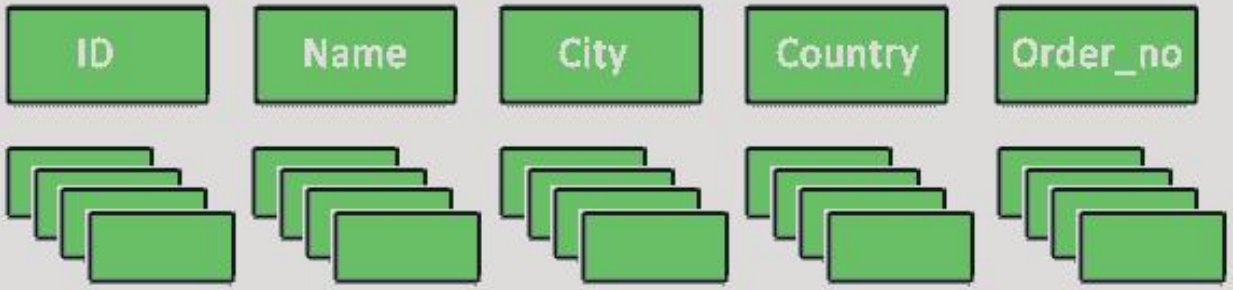
row-store



+ easy to add/modify a record

- might read in unnecessary data

column-store



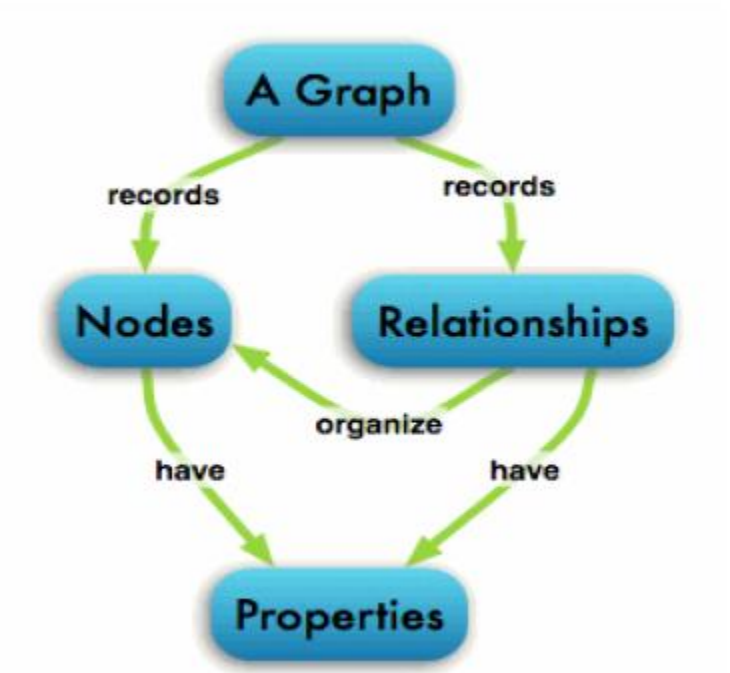
+ only need to read in relevant data

- tuple writes require multiple accesses

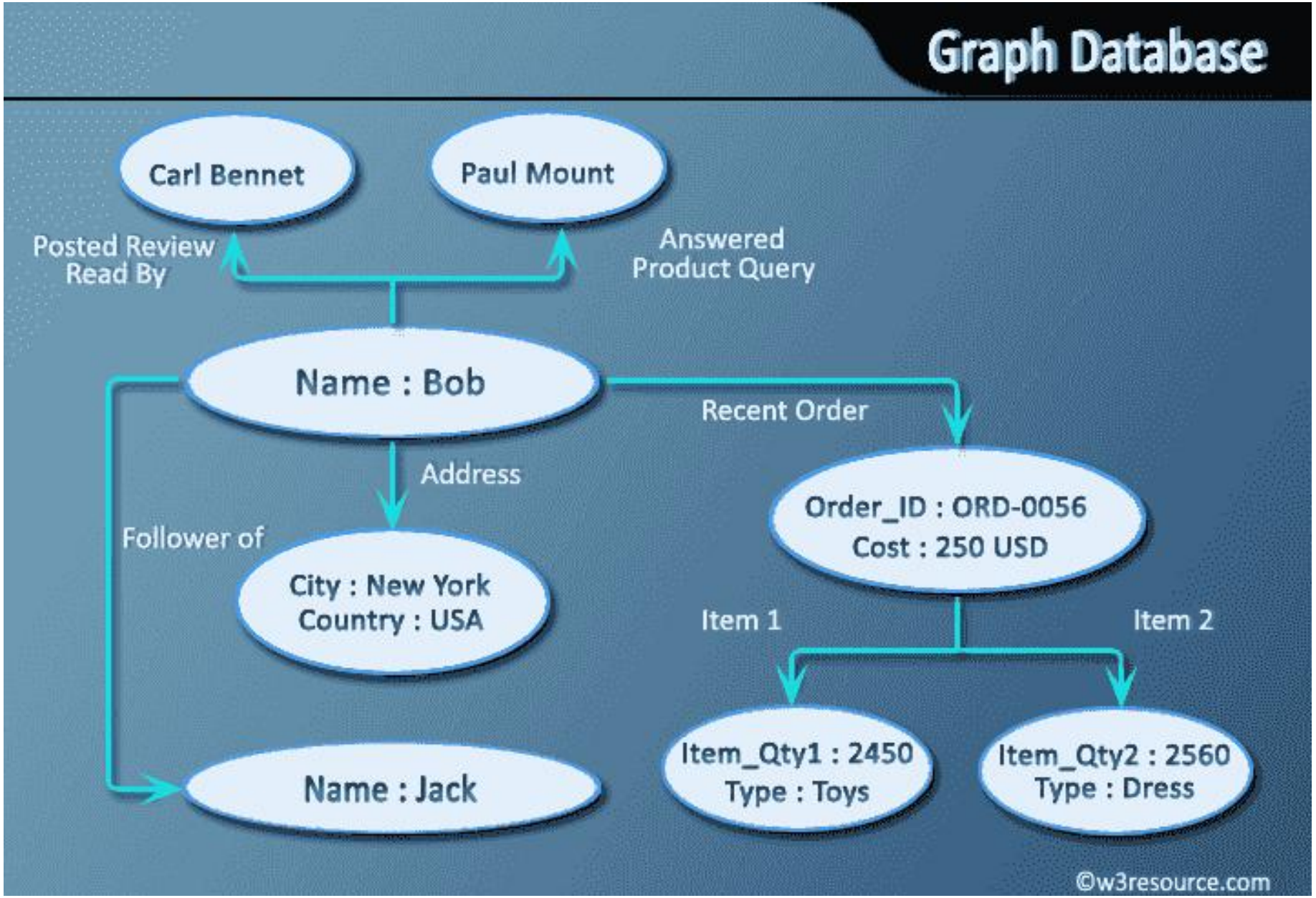
=>suitable for read-mostly, read-intensive, large data repositories

Graph Databases

- Κόμβοι: οντότητες
- Συσχετίσεις: ακμές (παρέχουν κατευθυνόμενες, ονομαστικές σημασιολογικά σχετικές συνδέσεις μεταξύ δύο κόμβων-οντοτήτων)
- Και οι κόμβοι, και οι συσχετίσεις έχουν ιδιότητες



http://neo4j.com/developer/graph-database/


Graph Database
Carl Bennet
Paul Mount
Posted Review
Read By
Answered
Product Query
Name : Bob
Recent Order
Address
Follower of
City : New York
Country : USA
Order_ID : ORD-0056
Cost : 250 USD
Item 1
Item 2
Item_Qty1 : 2450
Type : Toys
Item_Qty2 : 2560
Type : Dress
Name : Jack
©w3resource.com

Document Databases

- Μια συλλογή εγγράφων
- Τα δεδομένα σε αυτό το μοντέλο αποθηκεύνται μέσα σε έγγραφα
- Ένα έγγραφο είναι μια συλλογή από ζεύγη κλειδιού-τιμής όπου το κλειδί επιτρέπει την προσπέλαση στην τιμή του
- Τα έγγραφα τυπικά δεν είναι απαραίτητο να έχουν σχήμα και επομένως είναι ευέλικτα και επιτρέπουν εύκολα αλλαγές
- Τα έγγραφα αποθηκεύονται σε συλλογές έτσι ώστε να ομαδοποίησουμε διαφορετικά είδη δεδομένων
- Τα έγγραφα μπορούν να περιέχουν πολλά διαφορετικά ζεύγη κλειδιού-τιμής ή κλειδιού-πίνακα ή και εμφωλευμένα έγγράφα
- **Παραδείγματα Document databases :** MongoDB (FourSquarem Guthub, κτλ), CouchDB κτλ.

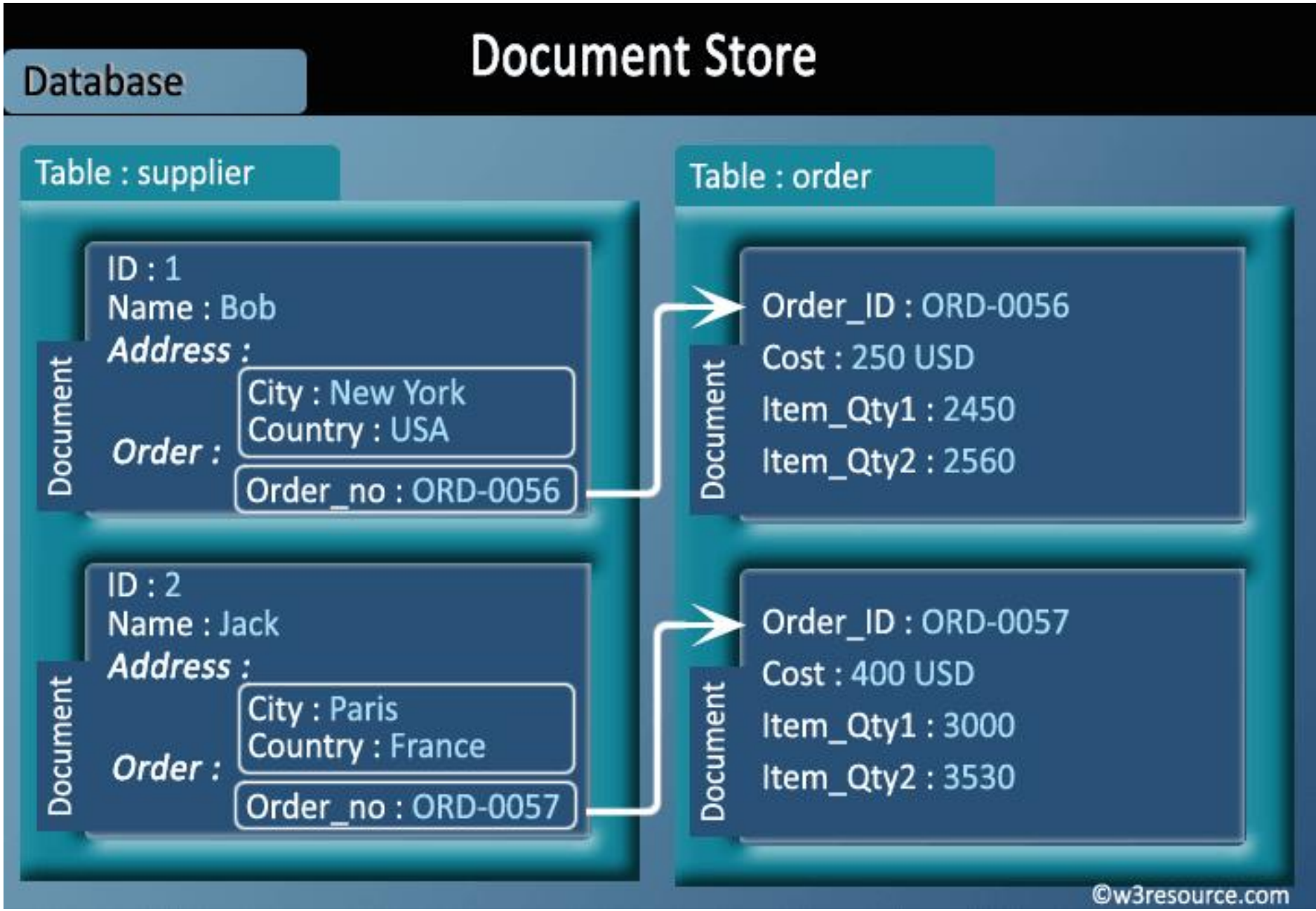
RDBMS vs Document DB

Σχεσιακό Μοντέλο

- Πίνακες
- Γραμμές
- Στήλες
- Συζεύξεις

Μοντέλο Εγγράφων

- Συλλογές
- Έγγραφα
- Ζεύγη κλειδιού-τιμής
- Μή διαθέσιμες


Database
Document Store
Table : supplier
Table : order
Document
ID : 1
Name : Bob
Address :
City : New York
Country : USA
Order :
Order_no : ORD-0056
Document
ID : 2
Name : Jack
Address :
City : Paris
Country : France
Order :
Order_no : ORD-0057
Document
Order_ID : ORD-0056
Cost : 250 USD
Item_Qty1 : 2450
Item_Qty2 : 2560
Document
Order_ID : ORD-0057
Cost : 400 USD
Item_Qty1 : 3000
Item_Qty2 : 3530
©w3resource.com

Μοντελοποίηση με NoSQL

- Η μοντελοποίηση για NoSQL δεδομένα συχνά ξεκινάει από τα ερωτήματα της εφαρμογής σε αντίθεση με τη σχεσιακή μοντελοποιήση:
  - Η σχεσιακή μοντελοποίηση συνήθως ωθείται από τη δομή των διαθέσιμων δεδομένων. Το βασικό θέμα της σχεδίασης είναι "**Τι είδους απαντήσεις έχω?**"
  - Η NoSQL μοντελοποίηση συνήθως οδηγείται από τα μοτίβα προσπέλασης της συγκεκριμένης εφαρμογής, δηλαδή, από τους τύπους ερωτημάτων που θα πρέπει να υποστηρίξουμε. Το βασικό θέμα σχεδιάσης είναι: **"Τι είδους ερωτήσεις έχω?"**
- Η αντιγραφή (duplication) και η αποκανονικοποίηση (denormalization) δεδομένων κυριαρχούν
- Οι σχεσιακές βάσεις δεδομένων δεν είναι πολύ βολικές για την μοντελοποίηση και την επεξεργασία ιεραρχικών ή γραφό-ειδών δεδομένων

Τεχνικές

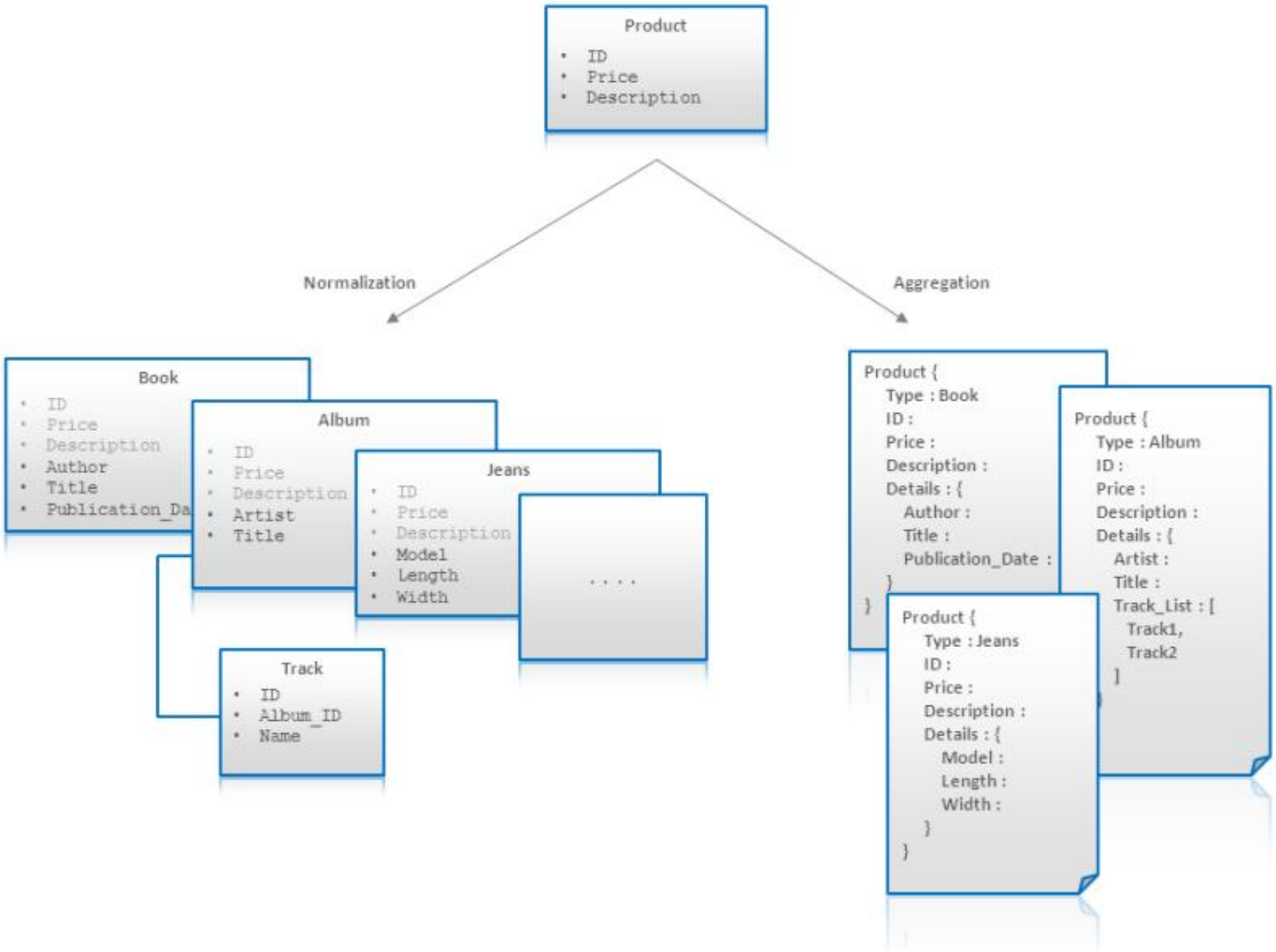
- Η αποκανονικοποίηση (denormalization) μπορεί να οριστεί ως η αντιγραφή των ίδιων δεδομένων σε πολλαπλά έγγραφα ή πίνακες έτσι ώστε να απλοποιηθεί/βελτιστοποιηθεί η επεξεργασία ερωτημάτων ή ώστε να ταιριάξουν τα δεδομένα του χρήστη σε ένα συγκεκριμένο μοντέλο
  - *Όγκος δεδομένων ερώτησης ή IO ανά ερώτηση* VS *συνολικός όγκος δεδομένων*
  - *Πολυπλοκότητα επεξεργασίας* VS *συνολικός όγκος δεδομένων*

Τεχνικές

– Συνάθροιση (aggregation)

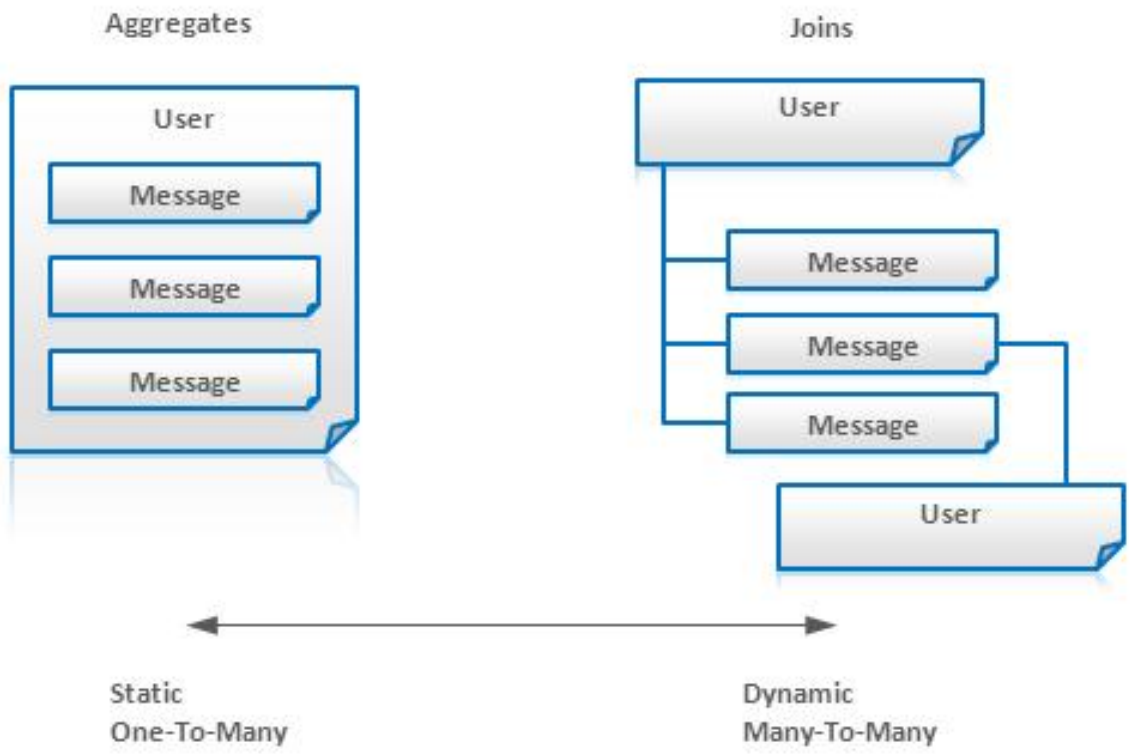
Το χαλαρό σχήμα επιτρέπει τη δημιουργία κλάσεων οντοτήτων με πολύπλοκες εσωτερικές δομές (εμφωλευμένες οντότητες) και τη διαφοροποίηση της δομής κάποιων οντοτήτων.

- Ελαχιστοποίηση των συσχετίσεων ένα-προς-πολλά με τη χρήση εμφωλευμένων οντοτήτων, και συνεπώς μείωση συζεύξεων.
- Κάλυψη “τεχνικών” διαφορών μεταξύ επιχειρησιακών οντοτήτων και τη μοντελοποίηση ετερογενών επιχειρησιακών οντοτήτων με τη χρήση μιας συλλογής εγγράφων ή ενός πίνακα.


Product
ID
Price
Description
Normalization
Aggregation
Book
ID
Price
Description
Author
Title
Publication_Da
Album
ID
Price
Description
Artist
Title
Jeans
ID
Price
Description
Model
Length
Width
. . . .
Track
ID
Album_ID
Name
Product {
Type : Book
ID :
Price :
Description :
Details : {
Author :
Title :
Publication_Date :
}
}
Product {
Type : Album
ID :
Price :
Description :
Details : {
Artist :
Title :
Track_List : [
Track1,
Track2
]
Product {
Type : Jeans
ID :
Price :
Description :
Details : {
Model :
Length :
Width :
}
}

Τεχνικές

- Application Side Joins
  - Τα συστήματα διαχείριζονται συχνά τις συζεύξεις κατά το χρόνο σχεδιασμού σε αντίθεση με τα σχεσιακά μοντέλα στα οποία οι συζεύξεις γίνονται στον χρόνο εκτέλεσης των ερωτημάτων



Συμπεράσματα

- Δεν υπάρχει η τέλεια NoSQL λύση
- Κάθε σύστημα έχει πλεονεκτήματα και μειονεκτήματα
- Κατάλληλα για:
  - χειρισμό μεγάλων δεδομένων
  - αραιών αδόμητων και ημί-δομημένων δεδομένων
  - υψηλή δυνατότητα οριζόντιας κλιμάκωσης
  - μαζικά παράλληλους υπολογισμούς
- Διαφορετικές εφαρμογές, στόχοι, προσεγγίσεις απαιτούν διαφορετικές NoSQL λύσεις

Αναφορές

https://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/
http://webdam.inria.fr/Jorge/files/sldatamodel.pdf