



ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Διάλεξη 9^η: Μη σχεσιακές Χωρικές Βάσεις Δεδομένων

Διδάσκοντες:

- Αναστάσιος Ζαφειρόπουλος, Δρ. Μηχ. ΣΗΜΜΥ ΕΜΠ
- Νικόλαος Μήτρου, Ομότιμος Καθ. ΣΗΜΜΥ ΕΜΠ



Μη σχεσιακές βάσεις δεδομένων

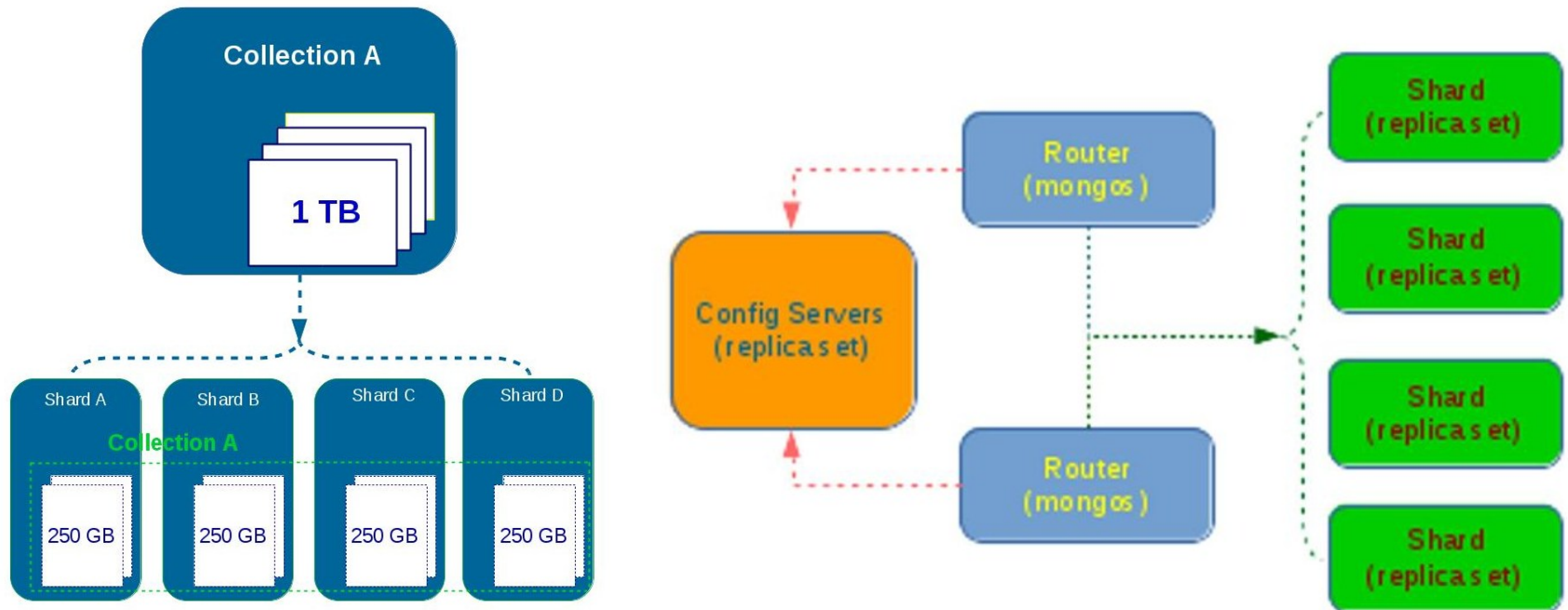
- Στόχος η βελτίωση του χρόνου απόκρισης σε ερωτήματα που διαχειρίζονται μεγάλο όγκο δεδομένων.
- Οριζόντια κλιμακωσιμότητα (horizontal scalability).
- Δεν ακολουθείται η αυστηρή δομή αναπαράστασης δεδομένων με τη μορφή οντοτήτων και συσχετίσεων μεταξύ τους





Μη σχεσιακές βάσεις δεδομένων (MongoDB)

Οριζόντια Κλιμακωσιμότητα



Source: <https://dzone.com/articles/divide-and-conquer-high-scalability-with-mongodb-t>



Μη σχεσιακές βάσεις δεδομένων

- Αποθήκευση δεδομένων με τη μορφή εγγράφων (document-oriented databases)
- Αποθήκευση δεδομένων με τη μορφή γράφων (graph stores)
- Αποθήκευση δεδομένων με βάση τη συσχέτιση ανάμεσα σε λέξεις-κλειδιά και στις τιμές τους (key-value stores)
- Αποθήκευση δεδομένων με τη μορφή πινάκων που δεν έχουν στατική μορφή αναπαράστασης (wide-column stores)



Κατηγοριοποίηση ΧΒΔ

Κατηγορία	Χαρακτηριστικά	Χωρικά δεδομένα
Σχεσιακές βάσεις	<ul style="list-style-type: none">- Καθορισμός οντοτήτων και συσχετίσεων- Σχεσιακό μοντέλο με αυστηρό ορισμό πινάκων και πεδίων- Συσχέτιση ανάμεσα στα πεδία των πινάκων με χρήση δευτερευόντων κλειδιών	<ul style="list-style-type: none">- Επεκτάσεις για χωρικά δεδομένα (π.χ., PostGIS)- Καθορισμός γεωμετριών ή γεωγραφιών
Μη Σχεσιακές βάσεις - αποθήκευση δεδομένων με τη μορφή εγγράφων	<ul style="list-style-type: none">- Μη αυστηρή δομή αναπαράστασης ενός εγγράφου- Σύνθετη δομή δεδομένων- Αποφυγή καθορισμού πεδίων που δεν χρησιμοποιούνται σε όλα τα έγγραφα- Κλιμακωσιμότητα (η βάση δεδομένων μπορεί να υποστηρίζεται καταναμημένα από πολλούς εξυπηρετητές)	<ul style="list-style-type: none">- Υποστήριξη χωρικών δεδομένων (π.χ., GeoJSON format)
Μη Σχεσιακές βάσεις - αποθήκευση δεδομένων με τη μορφή γράφου	<ul style="list-style-type: none">- Καλύτερη απόδοση σε σχήματα με πολύπλοκες συσχετίσεις- Εύληπτη γραφική αναπαράσταση εννοιών και συσχετίσεων	<ul style="list-style-type: none">- Επεκτάσεις για χωρικά δεδομένα



Αναπαράσταση δεδομένων με δυναμικό σχήμα με τη μορφή εγγράφων

- Η αναπαράσταση των δεδομένων γίνεται με βάση την περιγραφή ενός εγγράφου (αντικειμένου) το οποίο ενσωματώνει μια εσωτερική δομή αναπαράστασης της πληροφορίας.
- Κάθε έγγραφο στη βάση δεδομένων είναι ισοδύναμο με τον όρο του αντικειμένου, όπως αυτός χρησιμοποιείται στις αντικειμενοστραφείς γλώσσες προγραμματισμού.
- Κάθε έγγραφο δεν είναι αναγκαίο να ακολουθεί προκαθορισμένη δομή, αντίστοιχα με τις έννοιες των πινάκων, των συσχετίσεων και των πεδίων στις σχεσιακές βάσεις δεδομένων.
- Κάθε έγγραφο μπορεί να αποθηκεύεται με βάση διαφορετικό σύστημα αναπαράστασης (π.χ., JSON, XML).

```
{  
  "person": {  
    "name": "Maria Papadopoulou",  
    "address": {  
      "street": "Karaiskaki Street",  
      "city": "Athens",  
      "country": "Greece",  
      "postalCode": "10431",  
      "location": {  
        "latitude": 37.983810,  
        "longitude": 23.727750  
      }  
    }  
  }  
}
```



Συλλογή 1



Συλλογή 2



Αναπαράσταση δεδομένων με δυναμικό σχήμα με τη μορφή εγγράφων

Σχεσιακές Βάσεις Δεδομένων	Μη Σχεσιακές Βάσεις Δεδομένων
Βάση Δεδομένων	Βάση Δεδομένων
Προκαθορισμένο σχήμα Βάσης Δεδομένων	Δυναμικό σχήμα Βάσης Δεδομένων
Πίνακας (Table)	Συλλογή (Collection)
Στήλη (Column)	Πεδίο σε ένα έγγραφο (Field)
Εγγραφή (Row)	Έγγραφο (Document)
Πρωτεύον Κλειδί (Primary Key)	Πρωτεύον Κλειδί (π.χ., τυποποιημένο πεδίο <code>_id</code> στη MongoDB)
Δευτερεύον Κλειδί (Foreign Key)	Διασύνδεση πεδίων μέσω της ενσωμάτωσης των δεδομένων (Embedded) ή μέσω της δημιουργίας κατάλληλων αναφορών (Referenced)



Αναπαράσταση δεδομένων με δυναμικό σχήμα με τη μορφή εγγράφων

Student_Id	Student_Name	Age	Course
1001	Sarah	30	Networks
1002	John	29	Databases
1003	Alex	28	Databases

Σχισιακή βάση δεδομένων

```
{
  _id: ObjectId("60fbba5d4d3d9a4d34000001"),
  Student_Id: "1001",
  Student_Name: "Sarah",
  Age: 30,
  Course: "Networks"
},
{
  _id: ObjectId("60fbba5d4d3d9a4d34000002"),
  Student_Id: "1002",
  Student_Name: "John",
  Age: 29,
  Course: "Databases"
},
{
  _id: ObjectId("60fbba5d4d3d9a4d34000003"),
  Student_Id: "1003",
  Student_Name: "Alex",
  Age: 28,
  Course: "Databases"
}
```

Μη σχεσιακή βάση δεδομένων (MongoDB)

```
{
  {
    <key_1>: <value_1>,
    <key_2>: <value_2>,
    <key_3>: <value_3>,
    <key_4>: <value_4>
  },
  {
    <key_1>: <value_1>,
    <key_2>: <value_2>,
    <key_3>: <value_3>,
    <key_4>: <value_4>
  }
}
```

```
{
  {
    "_id": ObjectId("5ce45d7606444f199acfba1e"),
    "name": {given: "Nikos", family: "Nikolaou"},
    "email": "email@example.com"
    "age": 27
  },
  {
    "_id": ObjectId("5ce45d7606444f199acfba1e"),
    "name": {given: "Nikos", family: "Nikolaou"},
    "email": "email@example.com"
    "age": 27
  }
}
```



Αναπαράσταση δεδομένων με δυναμικό σχήμα με τη μορφή εγγράφων

- Ο σχεδιασμός του σχήματος μιας βάσης στη MongoDB πρέπει να γίνεται λαμβάνοντας υπόψη τις ανάγκες διαχείρισης δεδομένων και τις υπηρεσίες που πρέπει να παρέχονται προς τους τελικούς χρήστες.
- Είναι σημαντικό να συνδυάζονται πεδία σε ένα έγγραφο μιας συλλογής εφόσον η συνδυαστική πληροφορία που περιέχουν ζητείται συχνά, διαφορετικά, τα πεδία αυτά μπορούν να ανήκουν σε έγγραφα που αφορούν διαφορετικές συλλογές.
- Καλό είναι να αποφεύγεται η ρητή δήλωση συσχετίσεων (join).
- Σε περίπτωση που η πληροφορία ενός πεδίου απαιτείται να υπάρχει σε έγγραφα που αφορούν άλλες συλλογές, η πληροφορία αυτή μπορεί να σώζεται δύο ή και περισσότερες φορές.



Συσχετίσεις στη MongoDB

```
{
  _id: <ObjectId>,
  username: "Alex",
  contact: {
    phone: "123-000-0000",
    email: "alex@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}
```

Ενσωματωμένη πληροφορία

Ενσωματωμένη πληροφορία

Ενσωματωμένη συσχέτιση

```
{
  _id: <ObjectId1>,
  username: "Alex"
}
```

```
{
  _id: <ObjectId2>,
  user_id: <ObjectId1>,
  phone: "123-000-0000",
  email: "alex@example.com"
}
```

```
{
  _id: <ObjectId3>,
  user_id: <ObjectId1>,
  level: 5,
  group: "dev"
}
```

Αναφορική συσχέτιση



Συσχετίσεις στη MongoDB

```
# User:
{
  "_id": ObjectId("52ffc33cd85242f436000001"),
  "name": "Nikos Nikolaou",
  "contact": "0123456789",
  "dob": "01-01-1991"
}

# Address:
{
  "_id": ObjectId("52ffc4a5d85242602e000000"),
  "building": "Building B.13",
  "pincode": 123456,
  "city": "Athens",
  "area": "City Center"
}

# Ενσωματωμένη Συσχέτιση (Embedded Relationship):
{
  {
    "_id": ObjectId("52ffc33cd85242f436000001"),
    "contact": "987654321",
    "dob": "01-01-1991",
    "name": "Nikos Nikolaou",
    "address": [
```

```
    {
      "building": "Building B.13",
      "pincode": 123456,
      "city": "Athens",
      "area": "City Center"
    },
    {
      "building": "Building B.14",
      "pincode": 345678,
      "city": "Athens",
      "area": "City Center"
    }
  ]
}

# Αναφορική Συσχέτιση (Referenced Relationship):
{
  "_id": ObjectId("52ffc33cd85242f436000001"),
  "contact": "987654321",
  "dob": "01-01-1991",
  "name": "Nikos Nikolaou",
  "address_ids": [
    ObjectId("52ffc4a5d85242602e000000"),
    ObjectId("52ffc4a5d85242602e000001")
  ]
}
```



Συσχετίσεις στη MongoDB

- Στην περίπτωση των αναφορικών συσχετίσεων μπορεί να γίνει χρήση των MongoDB DBRefs για τη διασφάλιση της σωστής αναφοράς σε πεδίο εγγράφου από άλλη συλλογή.
- Ένα DBRef περιέχει τα ακόλουθα πεδία:
 - \$ref: το πεδίο αυτό καθορίζει τη συλλογή του εγγράφου στο οποίο γίνεται αναφορά
 - \$id: το πεδίο αυτό καθορίζει το _id του εγγράφου στο οποίο γίνεται αναφορά
 - \$db: προαιρετικό πεδίο που περιέχει το όνομα της βάσης δεδομένων

```
{
  "_id": ObjectId("53402597d852426020000002"),
  "address": {
    "$ref": "address_home",
    "$id": ObjectId("534009e4d852427820000002"),
    "$db": "example"},
  "contact": "987654321",
  "dob": "01-01-1991",
  "name": "Nikos Nikolaou"
}
```



Χωρικά δεδομένα στη MongoDB

- Η MongoDB υποστηρίζει την αποθήκευση χωρικών δεδομένων καθώς και μια σειρά από συναρτήσεις για την υποβολή χωρικών ερωτημάτων.
- Τα χωρικά δεδομένα μπορούν να αποθηκευτούν με τη μορφή GeoJSON ή ως ζεύγος συντεταγμένων.
- Η MongoDB υποστηρίζει πολλαπλούς τύπους αναπαράστασης γεωμετριών με βάση την GeoJSON, όπως τα Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, και GeometryCollection

```
location: {
  type: "Point",
  coordinates: [-73.856077, 40.848447]
}

# Σημείο (Point)
{ type: "Point", coordinates: [ 40, 5 ] }

# Γραμμή (LineString)
{ type: "LineString", coordinates: [ [ 40, 5 ], [ 41, 6 ] ] }

# Πολύγωνο με απλό δακτύλιο (Polygon with a Single Ring)
{
  type: "Polygon",
  coordinates: [ [ [ 0 , 0 ] , [ 3 , 6 ] , [ 6 , 1 ] , [ 0 , 0 ] ] ]
}

# Πολύγωνο με πολλαπλούς δακτύλιους (Polygon with Multiple Rings)
{
  type : "Polygon",
  coordinates : [
    [ [ 0 , 0 ] , [ 3 , 6 ] , [ 6 , 1 ] , [ 0 , 0 ] ],
    [ [ 2 , 2 ] , [ 3 , 3 ] , [ 4 , 2 ] , [ 2 , 2 ] ]
  ]
}
```



Χωρικά δεδομένα στη MongoDB

- Η MongoDB υποστηρίζει την αποθήκευση χωρικών δεδομένων καθώς και μια σειρά από συναρτήσεις για την υποβολή χωρικών ερωτημάτων.
- Τα χωρικά δεδομένα μπορούν να αποθηκευτούν με τη μορφή GeoJSON ή ως ζεύγος συντεταγμένων.
- Η MongoDB υποστηρίζει πολλαπλούς τύπους αναπαράστασης γεωμετριών με βάση την GeoJSON, όπως τα Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, και GeometryCollection

```
# Συλλογή γεωμετριών (GeometryCollection)
{
  type: "GeometryCollection",
  geometries: [
    {
      type: "MultiPoint",
      coordinates: [
        [ -73.9580, 40.8003 ],
        [ -73.9498, 40.7968 ],
        [ -73.9737, 40.7648 ],
        [ -73.9814, 40.7681 ]
      ]
    },
    {
      type: "MultiLineString",
      coordinates: [
        [ [ -73.96943, 40.78519 ], [ -73.96082, 40.78095 ] ],
        [ [ -73.96415, 40.79229 ], [ -73.95544, 40.78854 ] ],
        [ [ -73.97162, 40.78205 ], [ -73.96374, 40.77715 ] ],
        [ [ -73.97880, 40.77247 ], [ -73.97036, 40.76811 ] ]
      ]
    }
  ]
}
```



Χωρικά δεδομένα στη MongoDB

- Για την υποβολή χωρικών ερωτημάτων στα χωρικά δεδομένα της βάσης, είναι αναγκαία η δημιουργία χωρικών ευρετηρίων (geospatial indexes).
- Υποστηρίζονται δύο τύποι χωρικών ευρετηρίων: 2dsphere και 2d.
- Ο τύπος 2dsphere αφορά ερωτήματα που υπολογίζουν γεωμετρίες στην επιφάνεια μιας σφαίρας (π.χ., επιφάνεια της Γης).
- Ο τύπος 2d αφορά ερωτήματα που υπολογίζουν γεωμετρίες σε ένα διδιάστατο επίπεδο.

```
db.collection.createIndex( { <location field> : "2dsphere" } )
```

```
db.collection.createIndex( { <location field> : "2d" } )
```



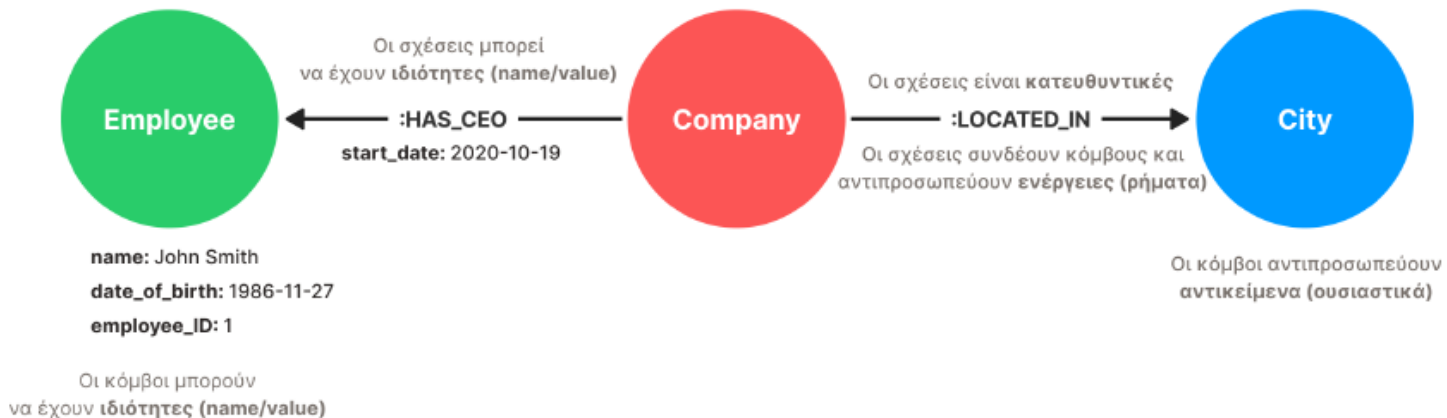
Τελεστές χωρικών ερωτημάτων στη MongoDB

Τελεστής	Περιγραφή
\$geoIntersects	Επιλέγει γεωμετρίες που τέμνονται με μια γεωμετρία (που αναπαρίσταται σε GeoJSON). Η συνάρτηση αυτή υποστηρίζεται από το ευρετήριο τύπου 2dsphere.
\$geoWithin	Επιλέγει γεωμετρίες εντός μιας οριοθετημένης γεωμετρίας GeoJSON. Το ερώτημα geoWithin μας δίνει τη δυνατότητα να αναζητήσουμε θέσεις που υπάρχουν πλήρως σε μια δεδομένη γεωμετρία, όπως ένας κύκλος, ένα πλαίσιο ή ένα πολύγωνο. Η συνάρτηση αυτή υποστηρίζεται από το ευρετήριο τύπου 2dsphere και 2d.
\$near	Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση near για να αναζητήσουμε μέρη σε μια δεδομένη απόσταση. Επιστρέφει γεωχωρικά αντικείμενα που βρίσκονται κοντά σε ένα σημείο. Απαιτεί την ύπαρξη χωρικού ευρετηρίου. Η συνάρτηση αυτή υποστηρίζεται από το ευρετήριο τύπου 2dsphere και 2d.
\$nearSphere	Επιστρέφει γεωχωρικά αντικείμενα που βρίσκονται κοντά σε ένα σημείο μιας σφαίρας. Απαιτεί την ύπαρξη χωρικού ευρετηρίου. Η συνάρτηση αυτή υποστηρίζεται από το ευρετήριο τύπου 2dsphere και 2d.
\$geoNear	Επιστρέφει μια ροή εγγράφων με βάση την εγγύτητα σε ένα χωρικό σημείο. Απαιτεί την ύπαρξη χωρικού ευρετηρίου.



Αναπαράσταση δεδομένων με τη μορφή γράφου

- Τα δεδομένα αναπαρίστανται με τη μορφή γράφου (graph data store).
- Η αναπαράσταση δεδομένων γίνεται με βάση τους κόμβους και τις ακμές του γράφου.
- Η μορφή αυτή είναι προτιμητέα κατά βάση σε περιπτώσεις όπου ορίζονται πολύπλοκες συσχετίσεις ανάμεσα στους κόμβους





Αναπαράσταση δεδομένων με τη μορφή γράφου

- Μια βάση δεδομένων με τη μορφή γράφου αποθηκεύει τις συνδέσεις μαζί με τα δεδομένα στο μοντέλο.
- Η απλή δομή αναπαράστασης πληροφορίας σε έναν γράφο έχει ως αποτέλεσμα πολύ απλούστερα και πιο εκφραστικά μοντέλα δεδομένων από αυτά που παράγονται χρησιμοποιώντας παραδοσιακές σχεσιακές ή άλλες μη σχεσιακές βάσεις δεδομένων.
- Η πρόσβαση σε κόμβους και σχέσεις σε μια βάση δεδομένων με τη μορφή γράφου είναι μια αποτελεσματική λειτουργία που επιτρέπει να διασχίζονται γρήγορα εκατομμύρια συνδέσεις ανά δευτερόλεπτο.
- Ανεξάρτητα από το συνολικό μέγεθος των δεδομένων, οι βάσεις δεδομένων με τη μορφή γράφου υπερέχουν στη διαχείριση δεδομένων, όταν υφίστανται πολλές συσχετίσεις (πυκνός γράφος) ή γίνεται συχνά υποβολή πολύπλοκων ερωτημάτων (π.χ., που διασχίζουν πολλές οντότητες μέσω των συσχετίσεων).

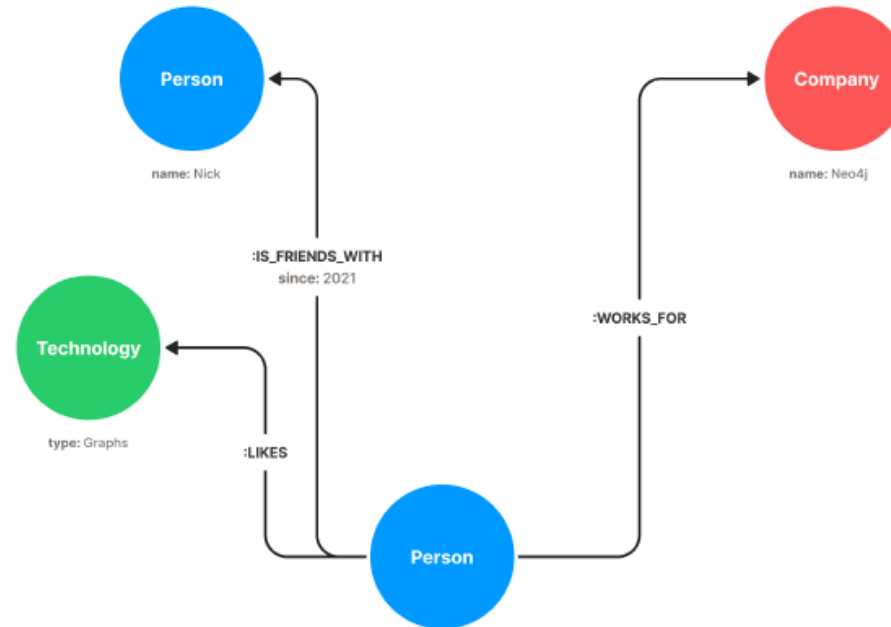


Αναπαράσταση δεδομένων με τη μορφή γράφου

- Τον Σεπτέμβριο του 2019, έγινε αποδεκτή από τα μέλη του ISO/IEC Joint Technical Committee (ISO/IEC JTC) η Graph Query Language (GQL) ως ένα νέο πρότυπο διατύπωσης ερωτημάτων σε γράφους (property graph language).
- Οι υφιστάμενες βάσεις δεδομένων με τη μορφή γράφων (π.χ., Neo4j, PGQL) υποστηρίζουν μέρος των προδιαγραφών που καθορίζονται από την GQL.



Αναπαράσταση δεδομένων με τη μορφή γράφου



```
// δεδομένα που αποθηκεύονται με συγκεκριμένη κατεύθυνση  
CREATE (p:Person)-[:LIKES]->(t: Technology)
```

```
// ερώτημα αναζήτησης δεδομένων με συγκεκριμένη κατεύθυνση  
// (πιθανότατα, δεν θα επιστρέψει κάποιο αποτέλεσμα)  
MATCH (p:Person)<[:LIKES]-(t: Technology)
```

```
// προτιμητέο το ερώτημα χωρίς δήλωση της κατεύθυνσης, εκτός αν  
// αυτή είναι ξεκάθαρη  
MATCH (p:Person)-[:LIKES]-(t: Technology)
```



Αναπαράσταση δεδομένων με τη μορφή γράφου

Λέξη	Περιγραφή
MATCH	Αναζητά έναν υπάρχοντα κόμβο, σχέση, ετικέτα, ιδιότητα ή μοτίβο στη βάση δεδομένων.
RETURN	Καθορίζει ποιες τιμές ή αποτελέσματα θα επιστραφούν από ένα ερώτημα σε Cypher. Επιτρέπεται η παροχή πληροφορίας για κόμβους, σχέσεις, ιδιότητες κόμβων και σχέσεων ή μοτίβα στα αποτελέσματα ενός ερωτήματος. Ο τελεστής RETURN δεν απαιτείται για την εγγραφή δεδομένων στη βάση, ωστόσο είναι αναγκαίος κατά την ανάγνωση δεδομένων.

```
MATCH (p:Person)
RETURN p
LIMIT 1

MATCH (nick:Person {name: 'Nick Nikolaou'})
RETURN nick

MATCH (:Person {name: 'Nick Nikolaou'})-[:DIRECTED]-&gt;(technology
:Technology)
RETURN technology

//poorly-named property
MATCH (nick:Person {name:'Nikos Nikolaou'})-[rel:DIRECTED]-(
technology:Technology)
RETURN nick.name, nick.email, technology.title, technology.
description
```



Χωρικά δεδομένα στη Neo4j

- Η Cypher έχει ενσωματωμένη υποστήριξη για τον χειρισμό χωρικών δεδομένων (σημείων).
- Υποστηρίζει την αποθήκευση τιμών χωρικών δεδομένων ως ιδιοτήτων σε κόμβους και σχέσεις.
- Το Neo4j υποστηρίζει μόνο έναν τύπο χωρικών δεδομένων για την αναπαράσταση σημείων (points)

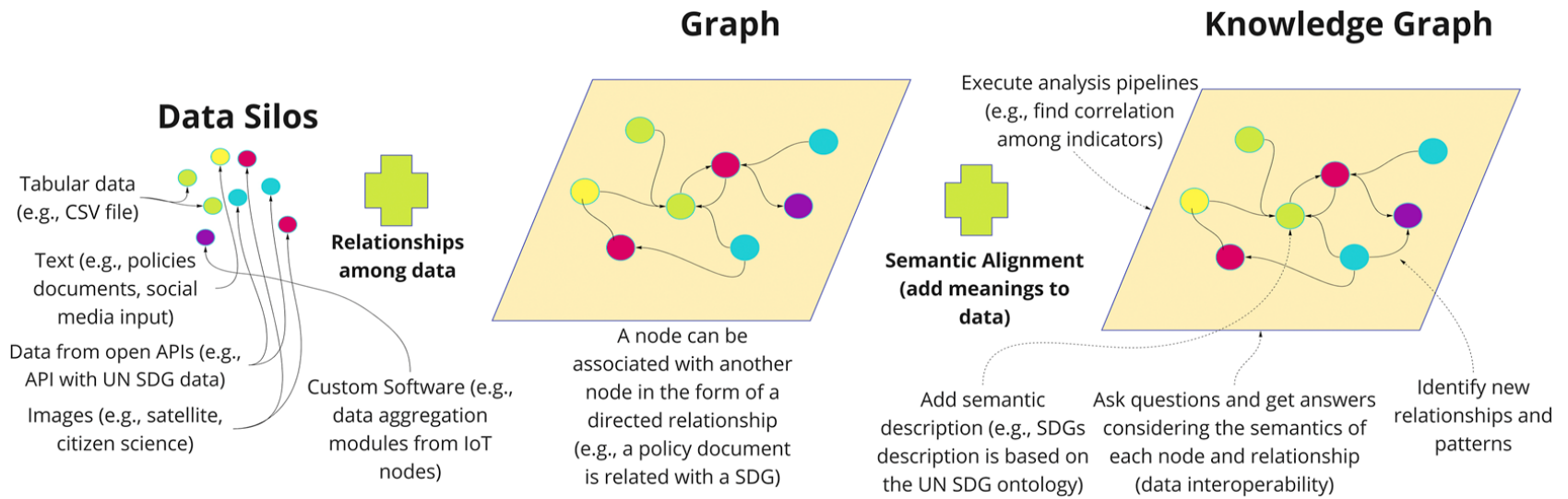
```
WITH
```

```
  point({latitude:toFloat('42.12'), longitude:toFloat('53.41')}) AS  
  point1,  
  point({latitude:toFloat('43.12'), longitude:toFloat('54.41')}) AS  
  point2
```

```
RETURN toInteger(point.distance(point1, point2)/1000) AS km
```



Γράφοι Γνώσης





Παράδειγμα: Γράφος Γνώσης SustainGraph

