

# Εργαστήριο Σημασιολογικού Ιστού

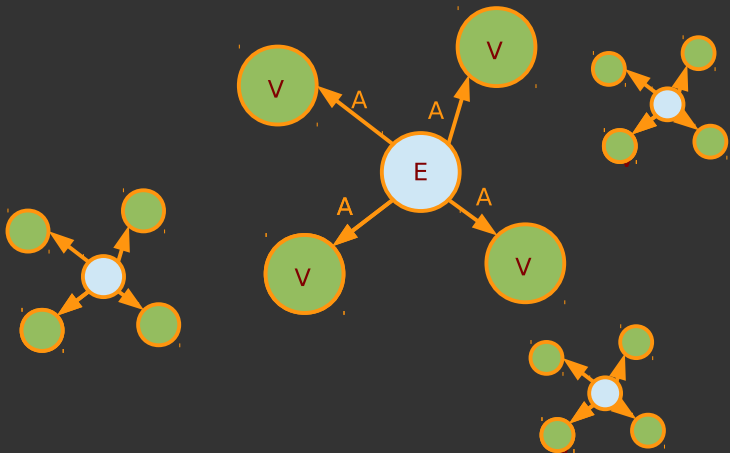
## Ενότητα 3: Από το μοντέλο EAV στους γράφους

Μ.Στεφανιδάκης

27-2-2020

# Το μοντέλο EAV σχηματικά

Τα δεδομένα ως τώρα έχουν τη μορφή μεμονωμένων “αστέρων”



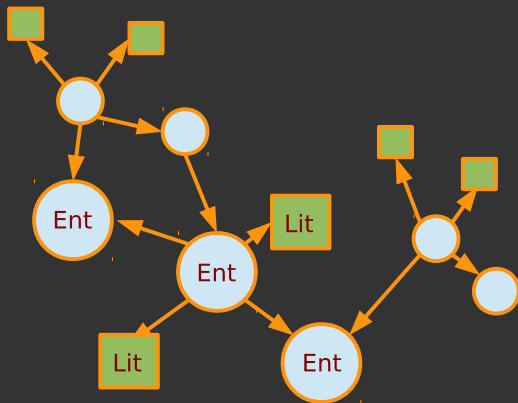
Είναι όμως πράγματι έτσι;

# Οι τιμές (values) στο μοντέλο EAV

- ▶ Σημασιολογικά, όλες οι τιμές (V) δεν είναι ίδιες
  - ▶ Στο παράδειγμα του ωρολογίου προγράμματος
- ▶ Υπάρχουν τιμές που θα μπορούσαν να είναι επίσης “οντότητες” (entities)
  - ▶ Με τις δικές τους ιδιότητες και τιμές
    - ▶ Τα Μαθήματα, οι Αίθουσες, οι Διδάσκοντες...
- ▶ Σε αντίθεση με
  - ▶ Τις Ημέρες, τις Ώρες...
  - ▶ Τα τελευταία είναι απλές σταθερές τιμές (literals)

## Ο μετασχηματισμός σε γράφο

- ▶ Θεωρώντας έναν μοναδικό κόμβο ανά οντότητα
- ▶ Ο γράφος περιγράφει τις σχέσεις μεταξύ οντοτήτων



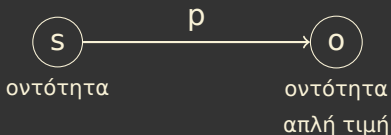
Η ιδέα είναι παλιά: βλ. “semantic networks” της Τεχνητής Νοημοσύνης (δεκαετίες 50-60)

# Πώς αναγνωρίζουμε τους κόμβους;

- ▶ Έμμεσα παραδεχόμαστε ότι το ίδιο “όνομα” (αλφαριθμητικό αναγνωριστικό) αναφέρεται στην ίδια οντότητα
  - ▶ Γι’ αυτό σας ζητήθηκε να τηρήσετε αυστηρά τα ίδια ονόματα στο παράδειγμα
- ▶ Η μέθοδος εφαρμόζεται όσο θεωρούμε ότι τα δεδομένα μας είναι μοναδικά στον κόσμο
  - ▶ Μη ρεαλιστική υπόθεση, θα ασχοληθούμε αργότερα με το θέμα αυτό...
- ▶ Προσοχή: για τις απλές τιμές (*literals*), το ίδιο αλφαριθμητικό δεν σημαίνει ταυτότητα
  - ▶ Π.χ. δύο εμφανίσεις του literal “Πέμπτη” δεν συγχωνεύονται σε μοναδικό κόμβο στον γράφο!

# Οι τριάδες ξανά – με άλλο όνομα

- ▶ Ισοδύναμο σχήμα με το μοντέλο EAV για την περιγραφή της οργάνωσης των δεδομένων
  - ▶ Χρήση **τριάδων** για την περιγραφή **γράφων δεδομένων**
  - ▶ Κάθε τριάδα αποτελεί μια “δήλωση” (statement) πληροφορίας
    - ▶ **Subject - Predicate - Object** ή απλά **(s,p,o)**
    - ▶ πολύ κοντά στην απλή φυσική μορφή “υποκείμενο – ρήμα – αντικείμενο”



## Σχετικά με τη φορά των ακμών $p$

- ▶ Ο γράφος που παράγουν οι τριάδες  $(s,p,o)$  είναι **κατευθυνόμενος**
  - ▶ Πώς διαλέγουμε τη φορά;
- ▶ Εξαρτάται από τις ανάγκες της εφαρμογής μας
  - ▶ Λειτουργικά, η τριάδα **(Διδάσκων $X$ , διδάσκει, Μάθημα $Y$ )** είναι ισοδύναμη με την **(Μάθημα $Y$ , διδάσκεται\_από, Διδάσκων $X$ )**
  - ▶ Μπορούμε να διαλέξουμε οποιαδήποτε από τις δύο
- ▶ Προσοχή: για τις απλές τιμές (**literals**), δεν έχουμε επιλογές
  - ▶ Εμφανίζονται πάντα στη θέση **o** (object) (ως στόχος της ακμής  $p$ )

## Ανώνυμοι κόμβοι (blank nodes)

- ▶ Κάθε οντότητα χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό όνομα
  - ▶ Χρησιμοποιείται π.χ. στα ερωτήματα
    - ▶ “Τί ξέρω για την οντότητα **ΌνομαΧ;**”
- ▶ Υπάρχουν όμως οντότητες “εσωτερικές” σε κάθε μοντέλο οργάνωσης δεδομένων
  - ▶ Βρίσκονται εκεί απλά και μόνο για να διασυνδέουν άλλες οντότητες
  - ▶ Δεν υπάρχει περίπτωση να είναι ο (κύριος) στόχος μιας ερώτησης
  - ▶ Το αναγνωριστικό τους δεν θα μεταδοθεί ποτέ “προς τα έξω”
    - ▶ το αναγνωριστικό αυτό έχει τοπική (local) μόνο σημασία
- ▶ Οι κόμβοι των εσωτερικών αυτών οντοτήτων ονομάζονται **ανώνυμοι** (blank nodes)



# Παράδειγμα ανώνυμων κόμβων

- ▶ Η κομβική οντότητα **Διάλεξη** στο παράδειγμα του ωρολογίου προγράμματος
  - ▶ Υπάρχει για να διασυνδέει Μαθήματα, Αίθουσες, Διδάσκοντες, Ημέρες και Ώρες
  - ▶ Δεν θα υπάρξει ερώτηση ειδικά για μία συγκεκριμένη διάλεξη
    - ▶ Παρά μόνο στα πλαίσια μιας ερώτησης σχετικής με τις άλλες οντότητες
  - ▶ Το (τεχνητό) αναγνωριστικό των διαλέξεων (1, 2, 3...) δεν έχει ιδιαίτερο νόημα εκτός της εφαρμογής μας
- ▶ Συνεπώς, οι κόμβοι των διαλέξεων είναι ένα τυπικό παράδειγμα **ανώνυμων κόμβων**

# Δοκιμάστε και εσείς

- ▶ Φτιάξτε νέο πρόγραμμα Python
  - ▶ Διαβάστε το τελευταίο csv αρχείο σας που περιέχει τις τριάδες
  - ▶ Για κάθε μία γραμμή, κατασκευάστε το μέρος του γράφου που συμβολίζει
  - ▶ Χρησιμοποιώντας το module **pydot**
    - ▶ Δείτε το παράδειγμα στις επόμενες διαφάνειες
    - ▶ (ή βρείτε το [on-line](#))
  - ▶ Θυμηθείτε ότι ταυτόσημα αναγνωριστικά οντοτήτων ή ανώνυμων κόμβων δημιουργούν **έναν και μοναδικό κόμβο** στον γράφο
    - ▶ Ενώ οι απλές τιμές (literals) δημιουργούν πάντα **έναν νέο κόμβο η κάθε μία**
  - ▶ Ως ετικέτες χρησιμοποιήστε τις τιμές των s, p, o
    - ▶ Αν υπάρχουν χαρακτήρες εκτός από αλφαριθμητικά και \_ τοποθετήστε " " γύρω από την ετικέτα

# pydot: παράδειγμα κώδικα

```
import pydot
```

```
# create the pydot directed graph
```

```
g = pydot.Dot(graph_type='digraph', splines='true',  
              overlap='false', size='80.0,80.0')
```

```
# add a graph node
```

```
node = pydot.Node("n1", shape='circle', style='filled',  
                  fillcolor='#FFFFFF', fontsize='8', margin='0')
```

```
node.set_label("α")
```

```
g.add_node(node)
```

```
# add a second node
```

```
node = pydot.Node("n2", shape='circle', style='filled',  
                  fillcolor='#FFFFFF', fontsize='8', margin='0')
```

```
node.set_label("β")
```

```
g.add_node(node)
```

```
# add an edge to graph
```

```
e = pydot.Edge("n1", "n2", color="#f89f12", fontsize='7')
```

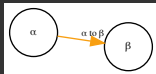
```
e.set_label("α to β")
```

```
g.add_edge(e)
```

```
# output graph (svg format)
```

```
g.write('test.svg', prog='neato', format='svg')
```

# Αποτέλεσμα προηγούμενου παραδείγματος



(test.svg)