

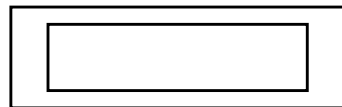
## Κεφάλαιο 2

# ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ Β' ΜΕΡΟΣ

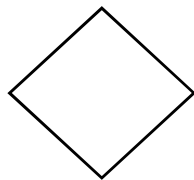
# Το E-R Μοντέλο: Γραφικός Συμβολισμός



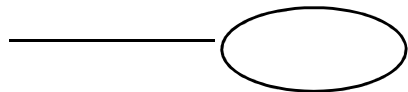
ENTITY SET



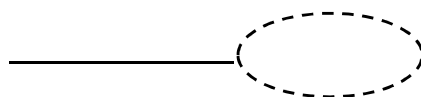
WEAK  
ENTITY SET



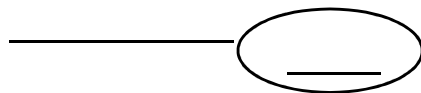
RELATIONSHIP  
SET



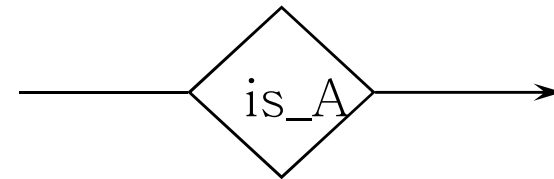
ATTRIBUTE



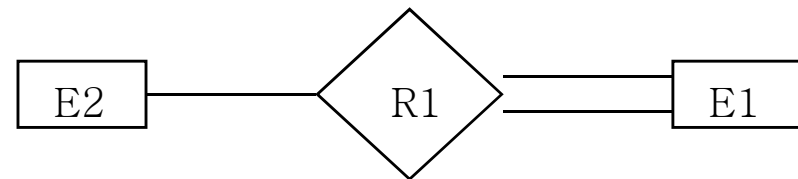
DERIVED  
Attribute



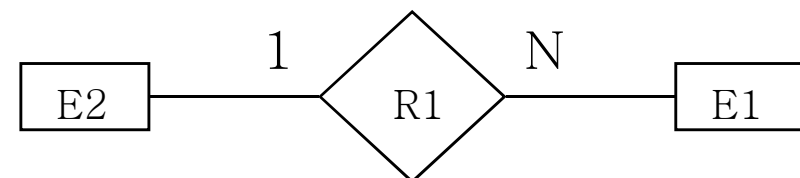
PRIMARY KEY



GENERALIZATION



Total PARTICIPATION of E1 in R1



Cardinality ratio 1:N for E2:E1 in R1

## Ενδεικτικό Παράδειγμα: ΒΔ για Εταιρεία (1)

- **Απαιτήσεις της Βάσης για κάποια Εταιρεία**
  - Μια εταιρεία οργανώνεται σε Τμήματα (DEPARTMENTS). Κάθε τμήμα έχει όνομα (*name*), αριθμό (*number*), και κάποιον υπάλληλο που είναι επικεφαλής (MANAGES) του τμήματος. Μας ενδιαφέρει η ημερομηνία έναρξης της θητείας του μάνατζερ (*start date*).
  - Ένα τμήμα κατανέμεται σε διάφορες τοποθεσίες (*locations*). Κάθε τμήμα ελέγχει έναν αριθμό από έργα (PROJECTS), όπου κάθε έργο έχει όνομα (*name*), αριθμό (*number*) και εκτελείται σε μια προκαθορισμένη τοποθεσία (*located*).

## Ενδεικτικό Παράδειγμα: ΒΔ για Εταιρεία (2)

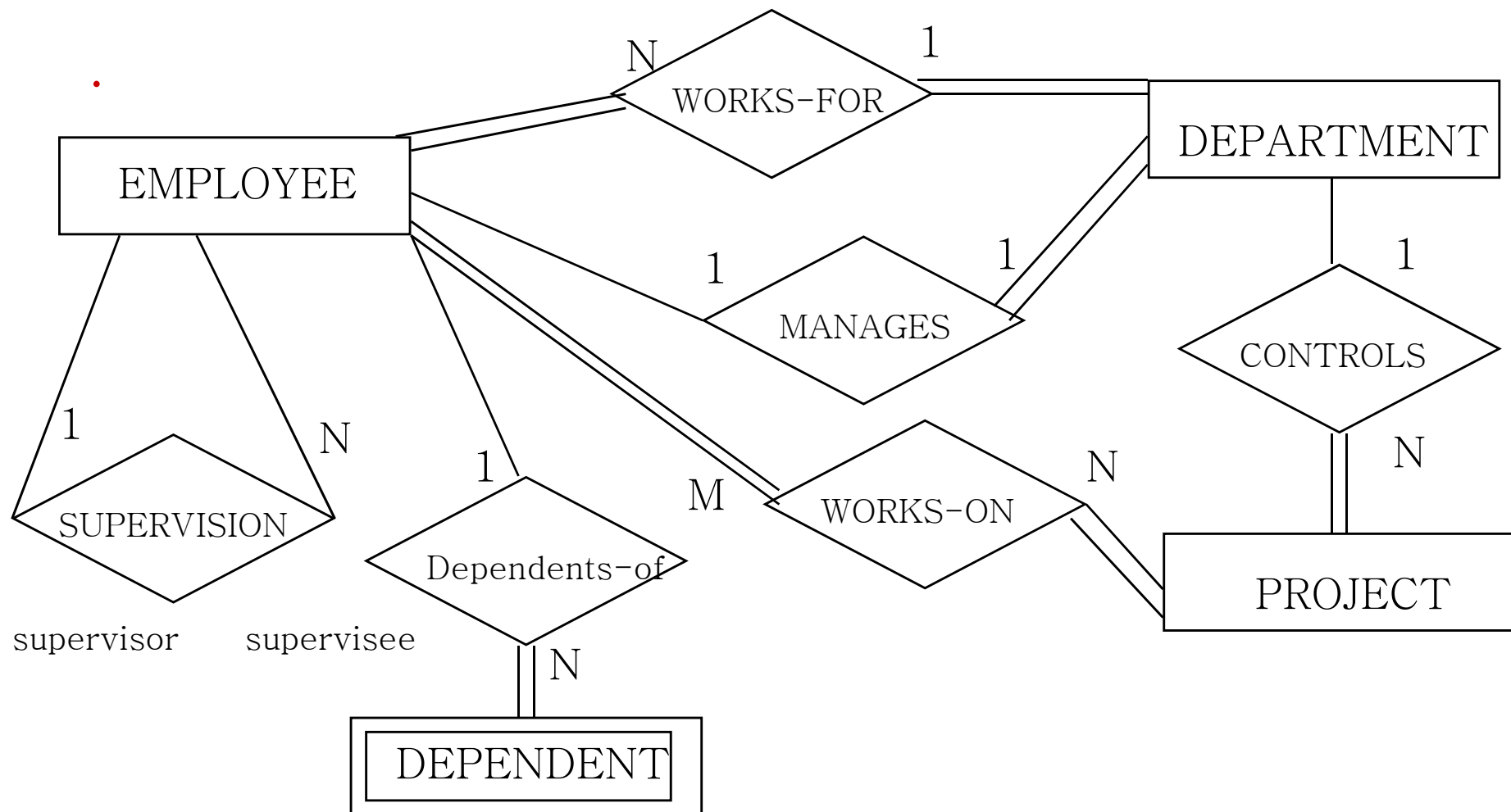
- Αναφορικά με Υπαλλήλους (EMPLOYEE), κρατάμε τον αριθμό κοινωνικής ασφάλισης (*social security number*), διεύθυνση (*address*), μισθό (*salary*), φύλλο (*sex*) και ημερομηνία γέννησης (*birth date*.)
- Κάθε Υπάλληλος εργάζεται (WORKS FOR) σε ένα τμήμα, αλλά μπορεί να εργάζεται (WORK ON) για πολλά έργα. Επίσης, κρατάμε τις ώρες / εβδομάδα (*number of hours*) που ένας υπάλληλος εργάζεται σε ένα έργο, όπως και τον/την άμεσο προϊστάμενο (*direct supervisor*.)
- Κάθε υπάλληλος μπορεί να έχει μια σειρά από εξαρτώμενα άτομα (DEPENDENTS) Για κάθε έναν, κρατάμε όνομα (*name*), ημερομηνία γέννησης (*birth date*), φύλλο (*sex*) και σχέση (*relationship*) με τον υπάλληλο.

## Περιγραφή της ΕΤΑΙΡΕΙΑΣ στο E-R (1)

### ■ ΓΝΩΡΙΣΜΑΤΑ σε Οντότητες και Συσχετίσεις

- EMPLOYEE -- SSN, Name, BirthDate, Sex, Address, Salary
- DEPARTMENT -- Number, Name, Locations, NoOfEmployees
- PROJECT -- Number, Name, Location
- DEPENDENT -- Name, Sex, BirthDate, Relationship
- WORKS-ON -- HoursPerWeek
- MANAGES -- StartDate

## Περιγραφή της ΕΤΑΙΡΕΙΑΣ στο E-R (1)



## Αδυναμίες του E-R Μοντέλου

- Έχουν προταθεί πολλά παρόμοια του **E-R** μοντέλα για να αντιμετωπίσουν τις εκφραστικές αδυναμίες αυτού.
  - Δεν υποστηρίζεται ο ορισμός συσχέτισης μεταξύ συσχετίσεων

*DATE is a relationship between MONTH, DAY and YEAR*

*EMPLOYMENT is a relationship between employee and employer*

*HIRING is a relationship between DATE and EMPLOYMENT*

## Επιπλέον ΔΟΜΙΚΟΙ ΠΕΡΙΟΡΙΣΜΟΙ

- **Συναρτησιακές Εξαρτήσεις (Functional dependencies):**
  - Π.χ., Ένα ΤΜΗΜΑ δεν μπορεί να παραγγείλει δύο διαφορετικά ΠΡΟΙΟΝΤΑ από τον ίδιο ΠΡΟΜΗΘΕΥΤΗ.
    - » Δεν μπορεί να εκφραστεί απευθείας στο ER μοντέλο.
- **Γενικοί Δομικοί Περιορισμοί:**
  - Π.χ., Ο προϋπολογισμός ενός Προϊσταμένου είναι λιγότερο κατά 10% του συνολικού Προϋπολογισμού όλων των τμημάτων που Προΐσταται



# Σήμερα

- **ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΜΟΝΤΕΛΟΥ ΟΝΤΟΤΗΤΩΝ – ΣΥΣΧΕΤΙΣΕΩΝ (Entity – Relationship)**
  - Επέκταση του Μοντέλου
  - Παραδείγματα
- **ΑΛΛΑ ΕΝΝΟΙΟΛΟΓΙΚΑ ΜΟΝΤΕΛΑ ΚΑΙ ΣΥΜΒΟΛΙΣΜΟΙ**
- **ΑΝΑΦΟΡΑ ΣΤΑ ΤΡΙΑ ΚΛΑΣΣΙΚΑ ΜΟΝΤΕΛΑ ΔΕΔΟΜΕΝΩΝ:**
  - Ιεραρχικό, Δικτύου (DBTG Network), και Σχεσιακό (Relational)

## ΕΠΕΚΤΕΙΝΟΝΤΑΣ τo E-R ΜΟΝΤΕΛΛΟ

- Εξειδίκευση/Γενίκευση (is\_A)
  - Περιορισμοί:
    - » Πως ορίζεται (από τον χρήστη / υπό συνθήκη)
    - » disjoint/ overlapping
    - » Ολική/μερική
    - » Κλπ.
- Κληρονομικότητα Γνωρισμάτων
- Συνάθροιση (Aggregation)

# Εξειδίκευση

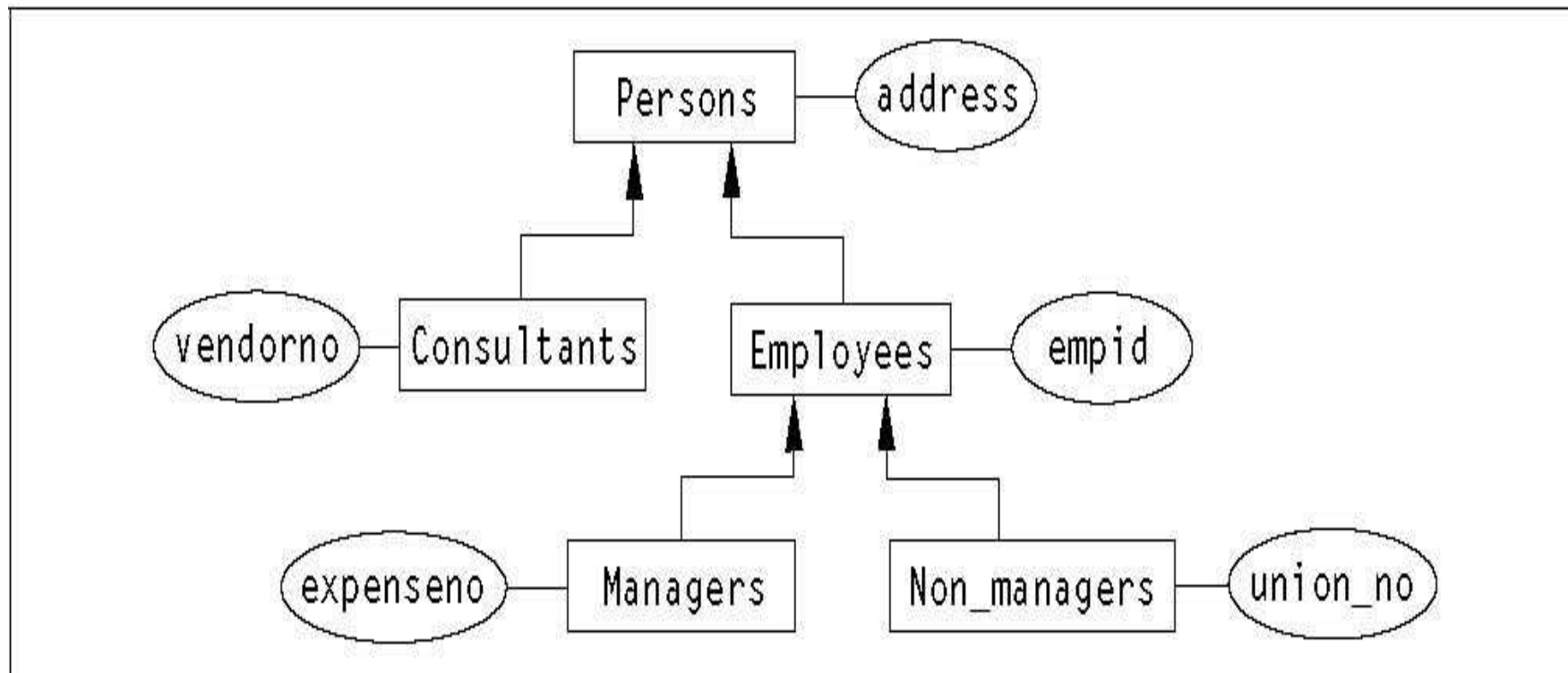
## Η Συσχέτιση ΕΞΕΙΔΙΚΕΥΣΗ (is-A)

- Ακολουθείται top-down λογική
- Ορίζουμε υποκλάσεις μιας οντότητας που έχουν ορισμένα εξειδικευμένα χαρακτηριστικά.
- Οι υποκλάσεις αυτές γίνονται ξεχωριστές οντότητες με γνωρίσματα που συμμετέχουν σε συσχετίσεις που δε συμμετέχει η υπερκλάση τους

Π.χ., PAINTING **is-A** ART-OBJECT, DRAWING **is-A** ART-OBJECT

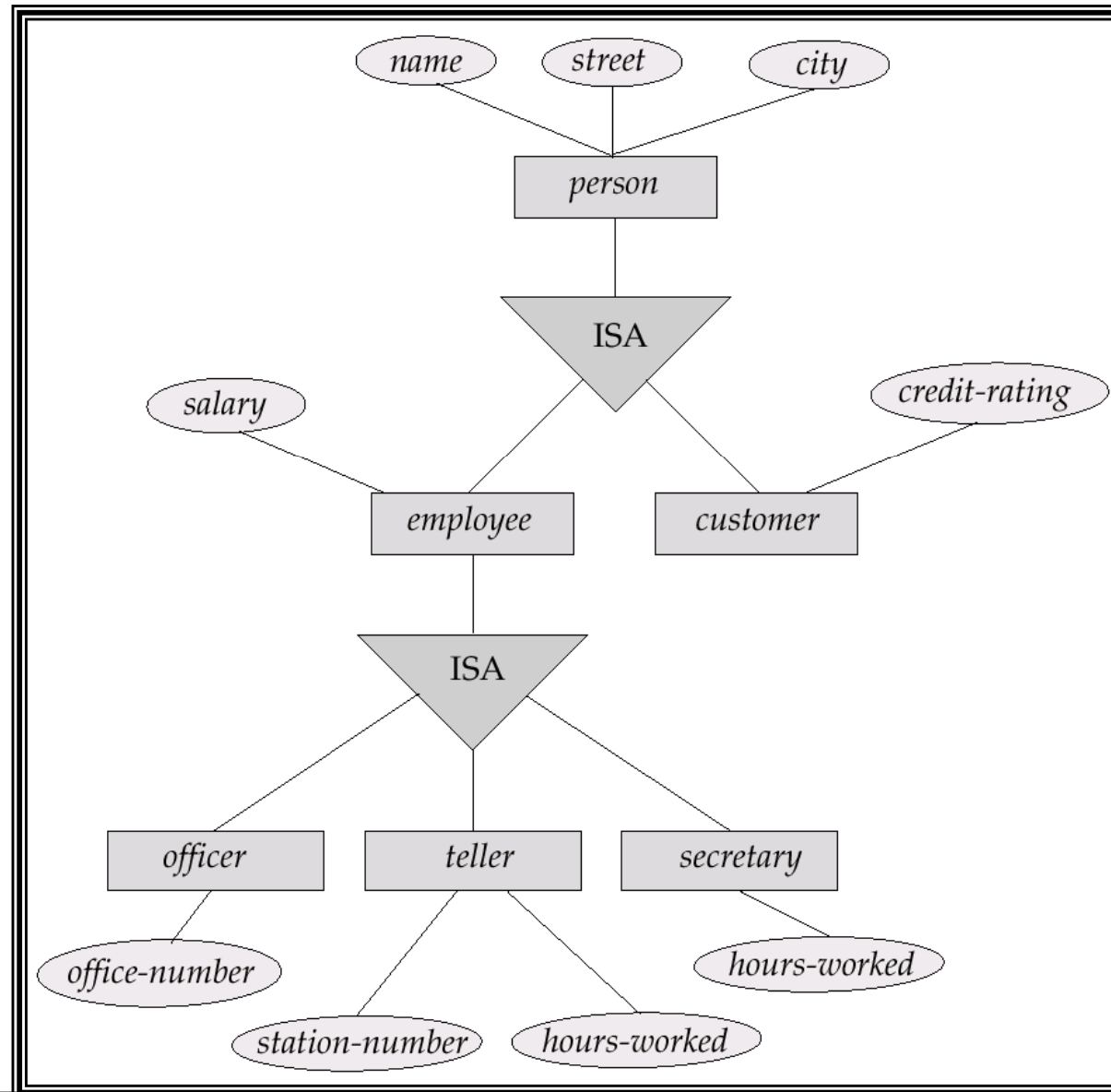
- Το βασικό πλεονέκτημα αυτού είναι η ιδιότητα της ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑΣ (*Inheritance*) των Γνωρισμάτων
- Ενδεικτικά, κρατάμε Γνωρίσματα που είναι κοινά για όλα τα ART-OBJECT και δεν χρειάζεται να τα επαναλάβουμε για τα PAINTINGS και DRAWINGS – κληρονομούνται.

## Παράδειγμα της is-A Συσχέτισης



A MANAGER **is\_A** EMPLOYEE, An EMPLOYEE **is\_A** PERSON

# Παράδειγμα της is-A Συσχέτισης – Κληρονομικότητα



## Γενίκευση

- Α bottom-up λογική– συνδυασμός από οντότητες που μοιράζονται κοινά χαρακτηριστικά φτιάχνοντας μια οντότητα υψηλότερου επιπέδου
- Είναι η αντίστροφη διαδικασία της εξειδίκευσης και αναπαριστώνται με τον ίδιο τρόπο στο E-R διάγραμμα
- Οι όροι γενίκευση και εξειδίκευση στην πράξη χρησιμοποιούνται με την ίδια έννοια

## Εξειδίκευση και Γενίκευση

- Μπορούμε να έχουμε πολλές εξειδικεύσεις μιας οντότητας με βάσει διαφορετικά χαρακτηριστικά
- Π.χ., permanent-employee vs. temporary-employee, αλλά και officer vs. secretary vs. teller
- Κάθε συμμετέχων εργαζόμενος θα ήταν
  - Μέλος ενός από τα permanent-employee ή temporary-employee,
  - Κι επίσης μέλος ενός από τα officer, secretary, ή teller
- Η συσχέτιση ISA λέγεται και superclass - subclass relationship

## Εξειδίκευση και Γενίκευση: Περιορισμοί

- Περιορισμοί για το ποιες οντότητες μπορούν να είναι μέλη μιας εξειδίκευσης
  - condition-defined  
Π.χ. όλοι οι πελάτες άνω των 65 γίνονται μέλη της οντότητας senior-citizen  
senior-citizen ISA person.
  - user-defined
- Περιορισμός μη επικάλυψης: ορίζει αν μια οντότητα ανήκει σε μία ή περισσότερες υποκλάσεις της ίδιας εξειδίκευσης
  - Μη επικαλυπτόμενες υποκλάσεις (disjoint)  
μια οντότητα μπορεί να ανήκει μόνο σε μια υποκλάση της εξειδίκευσης  
σημειώνεται στο E-R diagram γράφοντας disjoint δίπλα στο ISA τρίγωνο
  - Επικαλυπτόμενες υποκλάσεις (overlapping)  
μια οντότητα μπορεί να ανήκει σε περισσότερες από μία υποκλάσεις



## Εξειδίκευση και Γενίκευση: Περιορισμοί

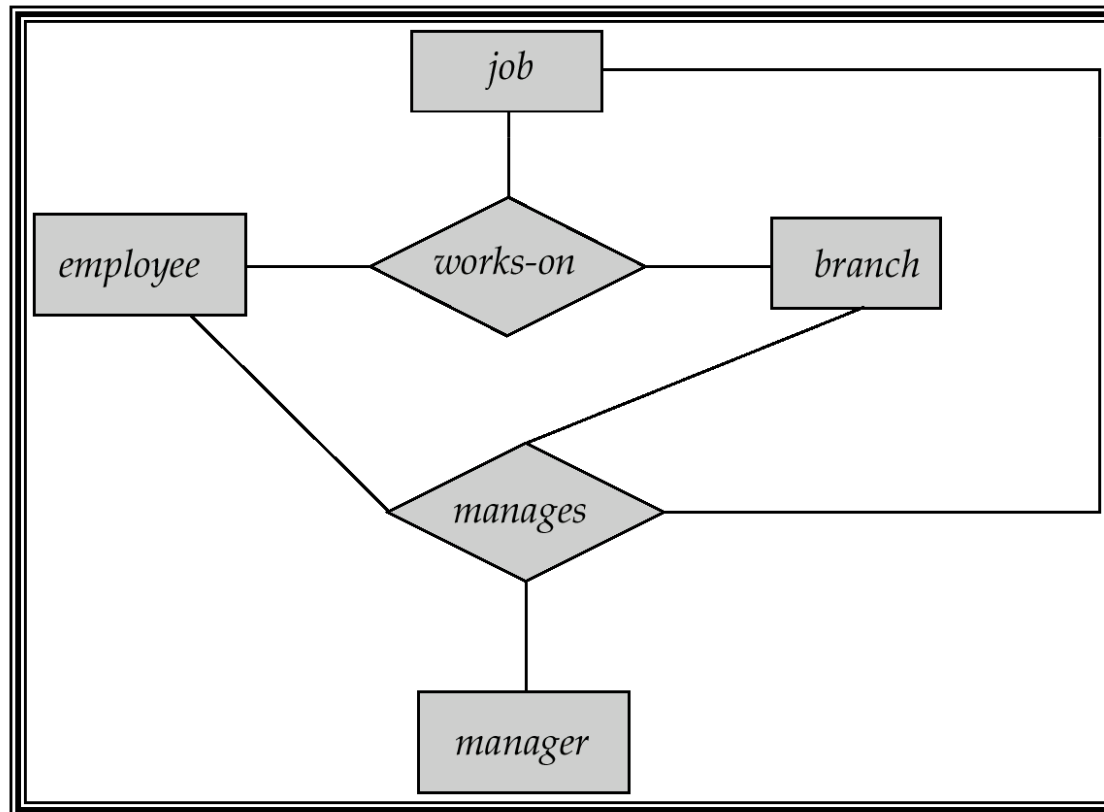
Περιορισμός πληρότητας: Ορίζει αν μια οντότητα της υπερκλάσης πρέπει να ανήκει τουλάχιστον σε μια υποκλάση της εξειδίκευσης

**Ολική εξειδίκευση:** μια οντότητα πρέπει να ανήκει σε μια από τις υποκλάσεις της εξειδίκευσης

**Μερική εξειδίκευση:** μια οντότητα δε χρειάζεται να ανήκει σε κάποια από τις υποκλάσεις της εξειδίκευσης

## Συνάθροιση (aggregation)

- Θεωρούμε τριπλή (ternary) συσχέτιση works-on
- Θέλουμε να καταγράψουμε τους managers για δουλειές που έγιναν από έναν employee σε ένα branch

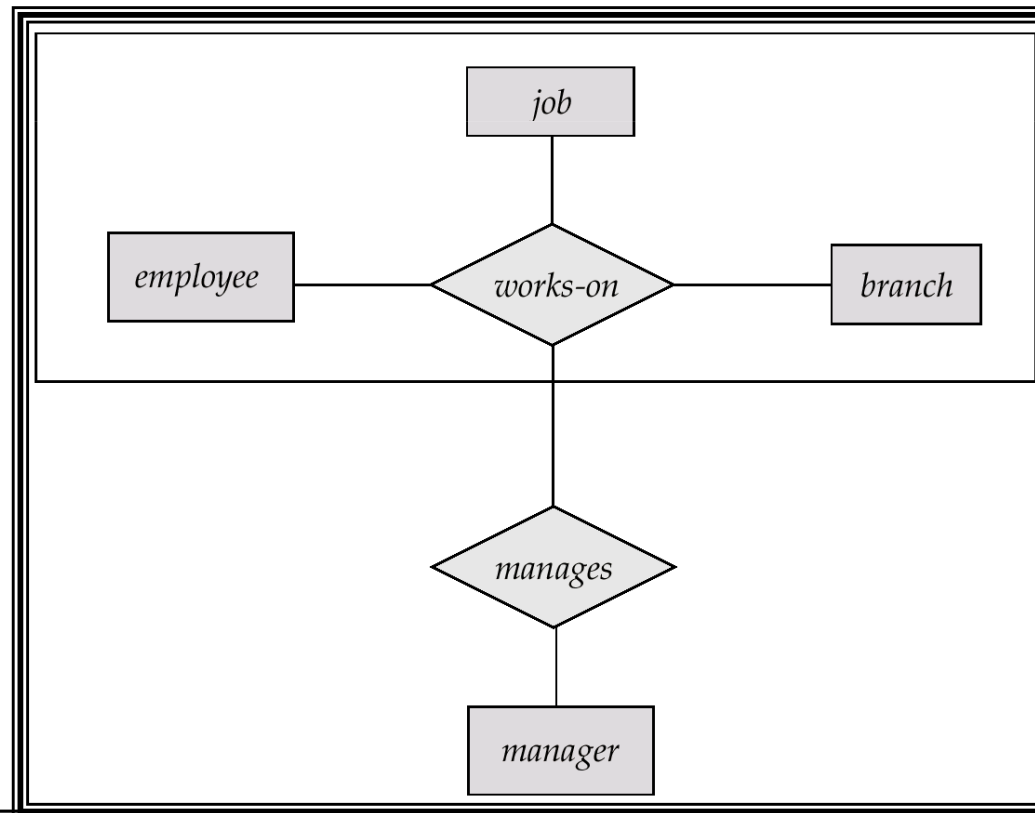


## Συνάθροιση

- Οι συσχετίσεις *works-on* και *manages* αναπαριστούν επικαλυπτόμενη πληροφορία
  - Κάθε συσχέτιση *manages* αντιστοιχεί σε μια *works-on*
  - Ωστόσο, κάποιες συσχετίσεις *works-on* μπορεί να μην αντιστοιχούν σε συσχετίσεις *manages*, οπότε δεν μπορούμε να αφαιρέσουμε τη συσχέτιση *works-on*
- Αυτός ο πλεονασμός αφαιρείται με *aggregation*
  - Μεταχειριζόμαστε τη συσχέτιση ως αφαιρετική οντότητα
  - Έτσι μπορούμε να έχουμε συσχέτιση συσχετίσεων

## Παράδειγμα E-R

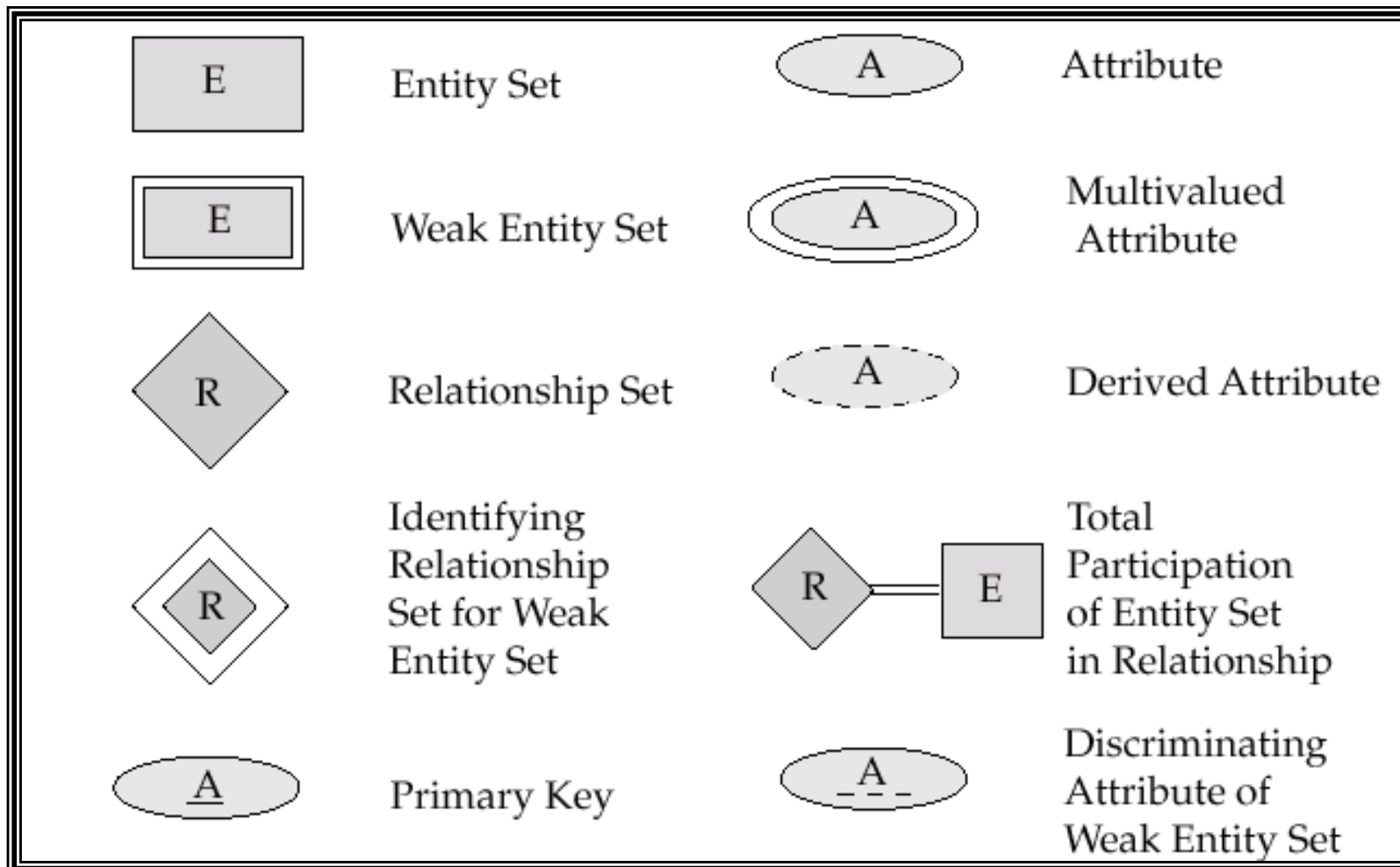
- An employee works on a particular job at a particular branch
- An employee, branch, job combination may have an associated manager



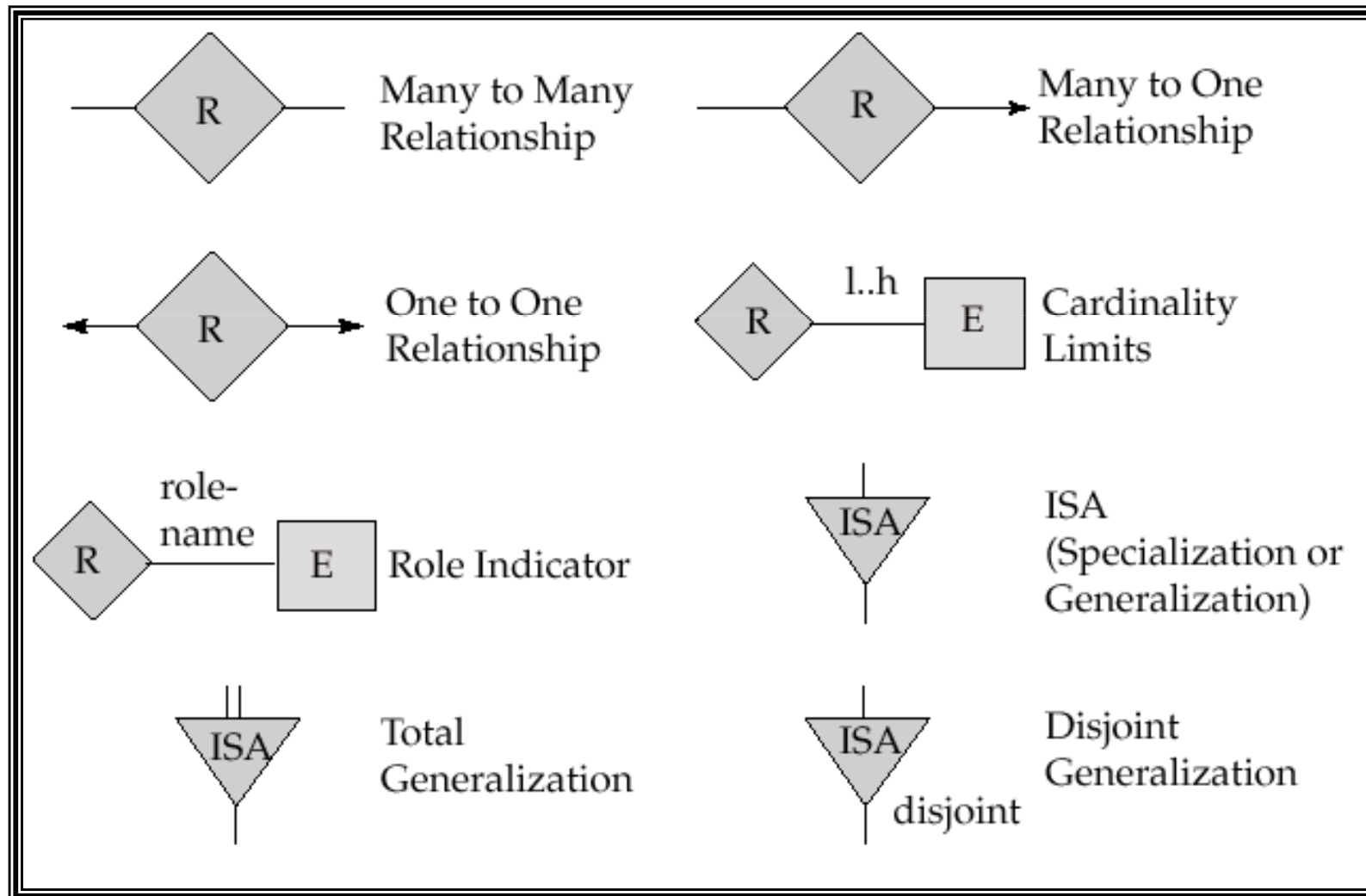
## Σχεδιαστικές αποφάσεις

- Γνώρισμα vs οντότητα για αναπαράσταση αντικειμένου
- Οντότητα vs συσχέτιση για αναπαράσταση έννοιας
- Ternary vs binary relationships
- Χρήση ασθενούς τύπου οντοτήτων
- Χρήση γενικεύσεων/εξειδικεύσεων
- Χρήση συνάθροισης

# ΣΥΜΒΟΛΙΣΜΟΙ στο Βιβλίο Silberschatz et al για το Εκτεταμένο Μοντέλο E-R



# ΣΥΜΒΟΛΙΣΜΟΙ στο Βιβλίο Silberschatz et al για το Εκτεταμένο Μοντέλο E-R(2)



# Εναλλακτικοί Συμβολισμοί

Entity set E with attributes A1, A2, A3 and primary key A1

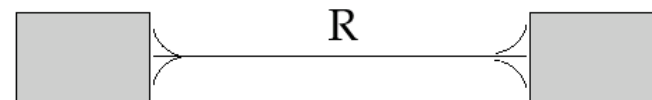
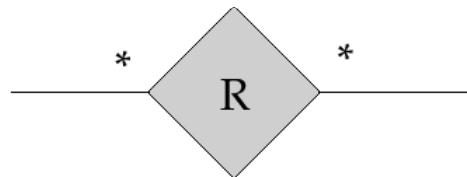
E

A1

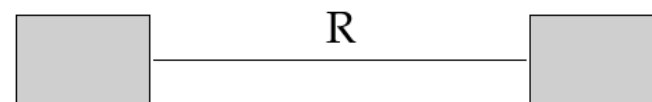
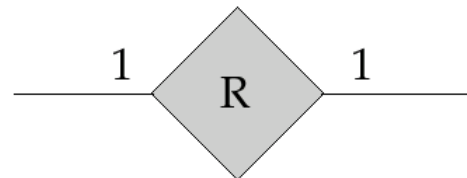
A2

A3

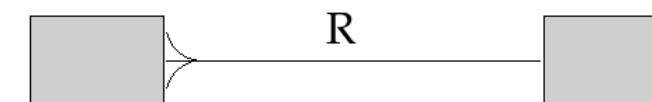
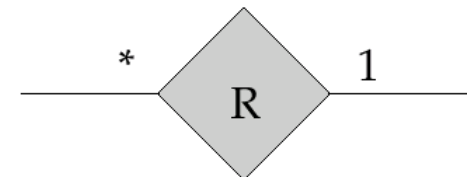
Many to Many Relationship



One to One Relationship

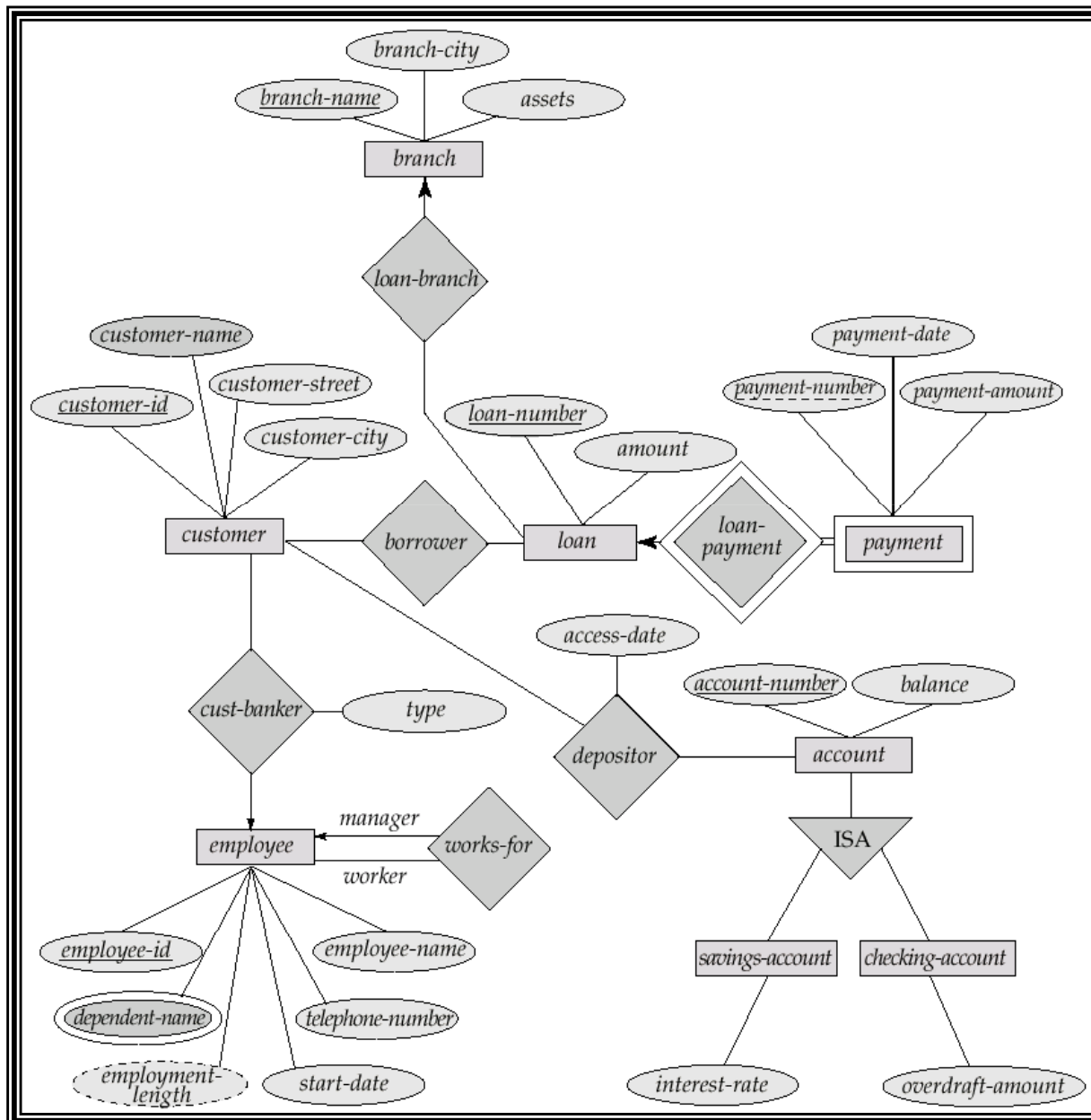


Many to One Relationship





# Ένα ενδεικτικό παράδειγμα στο E-R

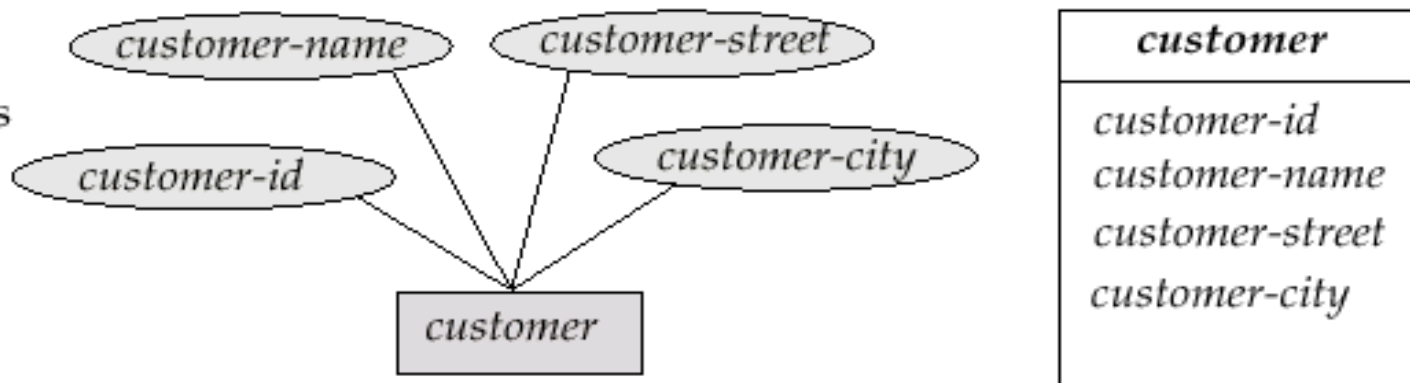


# UML

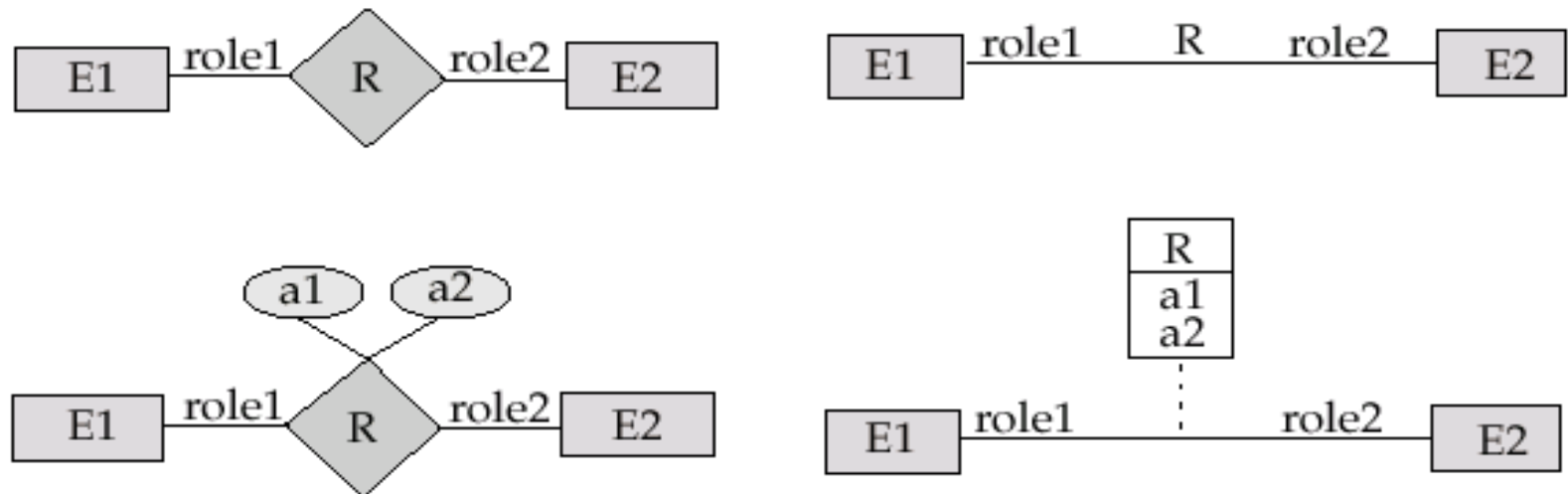
- UML: Unified Modeling Language
- Η UML έχει πολλές συνιστώσες για να είναι δυνατή η γραφική αναπαράσταση των διαφορετικών στοιχείων ενός γενικής μορφής συστήματος λογισμικού.
- Η UML αναπτύχθηκε για Τεχνολογίες Λογισμικού και κατόπιν εξελίχθηκε και για εργαλείο μοντελοποίησης Βάσεων Δεδομένων και επιχειρηματικών ροών (workflows)
- Τα UML Διαγράμματα Κλάσης (Class Diagrams) αντιστοιχούν στα E-R Διαγράμματα αλλά με ΠΟΛΛΕΣ διαφορές.

# Σύνοψη των UML Class Diagram Συμβολισμών

1. Entity sets and attributes



2. Relationships

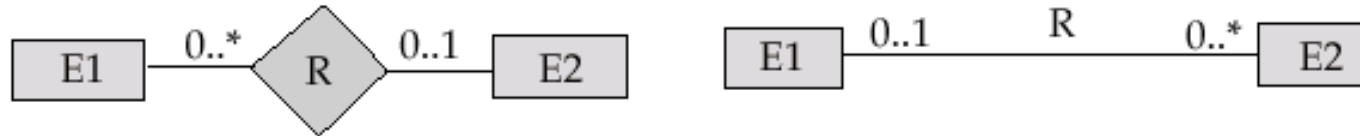


## UML Class Diagrams (συνέχεια)

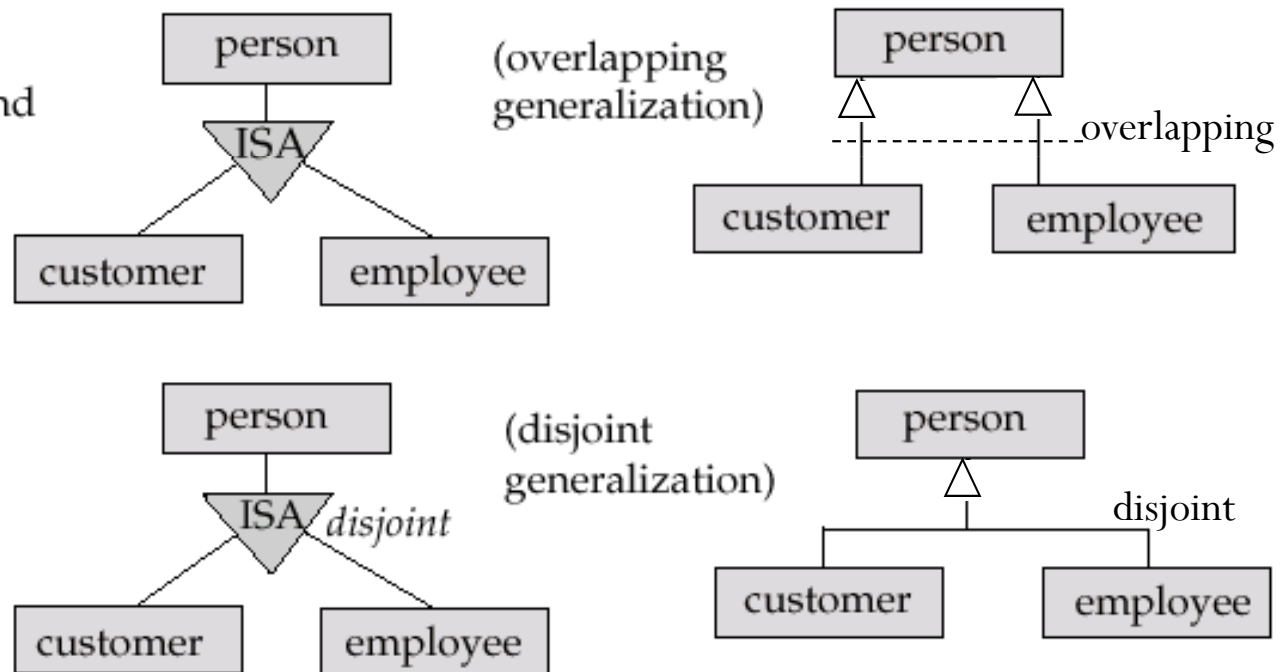
- Τα γνωρίσματα είναι ΜΕΣΑ στα κουτιά που παριστούν τις Οντότητες (και όχι μέσα σε ξεχωριστές ελλείψεις)
- Οι συσχετίσεις παρίστανται στην UML με μια απλή γραμμή που ενώνει τις οντότητες. Το όνομα της συσχέτισης παρατίθεται δίπλα στη γραμμή
- Ο ρόλος μιας οντότητας σε μια σχέση ΕΠΙΣΗΣ γράφεται δίπλα στη γραμμή.
- Οι μη-δυναδικές σχέσεις παρίστανται με «διαμάντια» όπως στο ER

# UML Class Diagram (συνέχεια)

3. Cardinality constraints



4. Generalization and Specialization



## UML Class Diagrams (συνέχεια)

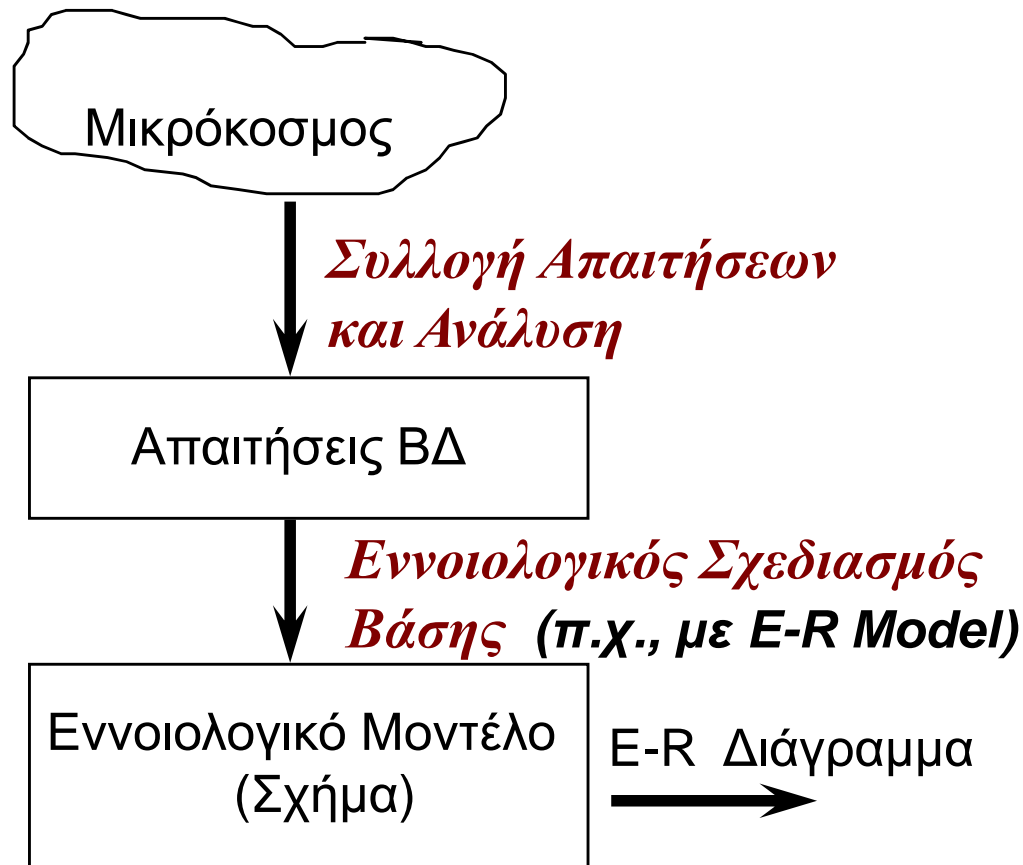
- Οι περιορισμοί συμμετοχής έχουν τη μορφή:  $l..h$ , όπου το  $l$  παριστά τον ελάχιστο και  $h$  το μέγιστο αριθμό συσχετίσεων όπου μια οντότητα δύναται να συμμετέχει
- Οι κατευθύνσεις στους περιορισμούς συμμετοχής (cardinality) ΕΙΝΑΙ ΑΚΡΙΒΩΣ ΑΝΤΙΘΕΤΟΙ από το E-R
- Ο περιορισμός  $0..*$  στην πλευρά του  $E2$  και ο  $0..1$  στην πλευρά του  $E1$  σημαίνει ότι η κάθε  $E2$  οντότητα μπορεί να συμμετέχει το πολύ σε μια συσχέτιση, ενώ κάθε  $E1$  οντότητα μπορεί να συμμετέχει σε πολλές (N:1 από το  $E2$  στο  $E1$ .)
- Ατομικές τιμές όπως το 1 ή το αστεράκι \* μπορεί να γραφεί στις ακμές. Το 1 είναι ισοδύναμο με  $1..1$  και το \* είναι ισοδύναμο με  $0..*$ .

## Άλλοι Συμβολισμοί

- Πολλά από τα εργαλεία για Μοντελοποίηση, όπως το VISIO ή το E-R Designer, χρησιμοποιούν άλλους τυποποιημένους συμβολισμούς, ενδεικτικά:
  - Chen' s ER
  - Barker ER Information Engineering (IE)
  - IDEF1X,
  - Object role modeling (ORM)
    - » Μεγάλη λεπτομέρεια και ακρίβεια – Τεράστια Διαγράμματα
  - Unified Modeling Language (UML)
    - » Ιδιαίτερο χαρακτηριστικό η ικανότητα παράστασης της ανάπτυξης ΒΔ σε ΚΑΘΕ αφαιρετικό επίπεδο

# Διαδικασία Ανάπτυξης ΒΔ: Εννοιολογικό Μοντέλο

## Ανεξάρτητα του DBMS



-Μετά την χρήση ενός Εννοιολογικού Μοντέλου Δεδομένων, όπως το E-R, ερχόμαστε σε ένα σημείο όπου αρχίζει η εξάρτηση από το επιλεγέν DBMS και πρέπει να γίνει η ΜΕΤΑΤΡΟΠΗ τις «διαισθητικής» περιγραφής σε μια «τυπική», που να μπορεί να εκτελεστεί από το DBMS.



# ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

## Εξάρτηση από το *DBMS*

Σε αυτό το σημείο **ΕΠΙΛΕΓΟΥΜΕ**  
το Μοντέλο Δεδομένων που  
Υποστηρίζεται από το *DBMS*.

Ο βασικός υποψήφιος είναι το  
**ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ** (άλλοι  
Υποψήφιοι ήταν το Ιεραρχικό, ή το  
Μοντέλο Δικτύου)

Εννοιολογικό Σχήμα

E-R διάγραμμα

Σχεσιακό  
Μοντέλο

Λογικό (Εννοιολογικό)  
Σχήμα και Εξωτερικές  
Όψεις

Λογικός Σχεδιασμός  
Βάσης Δεδομένων

# Λογικός Σχεδιασμός Βάσης

- Όλα τα Λογικά Μοντέλα για Σχεδιασμό παριστούν τους **Τύπους Οντοτήτων** (στο E-R) ως **ΑΡΧΕΙΑ** ( Relations)

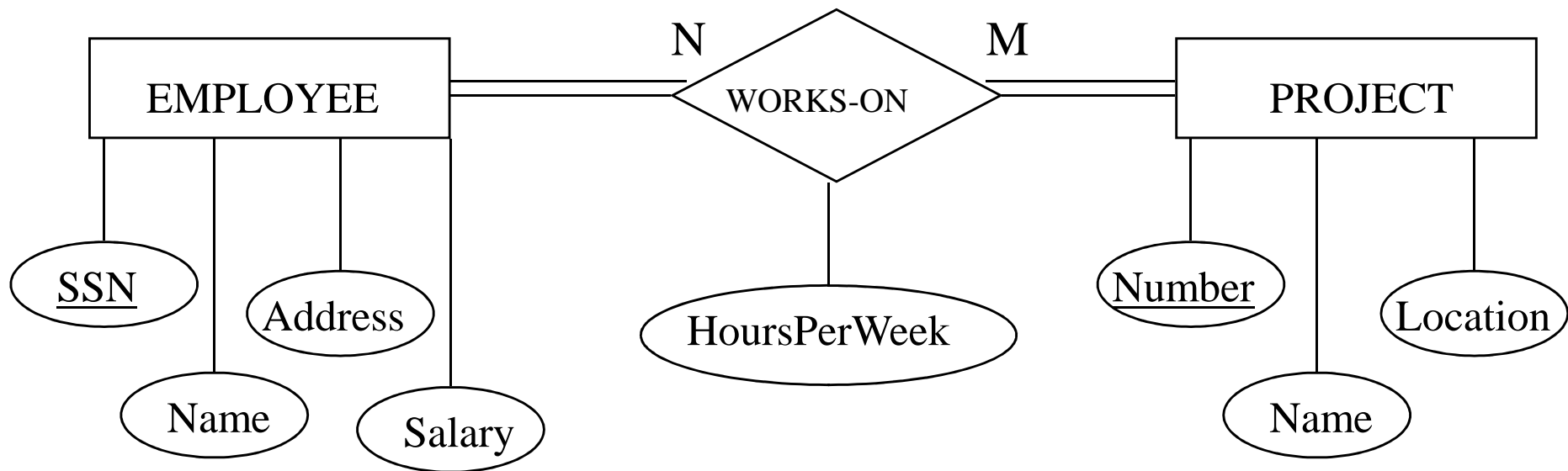
Entity Set	↔	File (relation)
Entity	↔	Record (Tuple)
Attribute	↔	Field (attribute)

- Οι διαφοροποιήσεις μεταξύ λογικών μοντέλων είναι στο **τρόπο παράστασης των Τύπων Συσχετίσεων**
  - Μερικά Μοντέλα (π.χ. Ιεραρχικό) επιτρέπουν μόνο συναρτησιακές συσχετίσεις (κάτι που δυσκολεύει την παράσταση N:M συσχετίσεων – μόνο έμμεσα είναι δυνατή)

- **ΒΑΣΙΚΕΣ ΤΕΧΝΙΚΕΣ:**

- Παράσταση συσχετίσεων με **ΣΧΕΣΕΙΣ** (π.χ., Σχισιακό Μοντέλο)
- Παράστασή των με **ΣΥΝΔΕΣΜΟΥΣ** μεταξύ αρχείων (π.χ., Μοντέλα Δικτύου και Ιεραρχικό)

## Παράδειγμα: Μικρό υποσύνολο της Βάσης Δεδομένων μιας Εταιρείας



Στο E-R Μοντέλο: Μια **N:M ολική συσχέτιση (WORKS-ON)** μεταξύ των τύπων οντοτήτων EMPLOYEE και PROJECT

## Παράδειγμα: (2)

- Θεωρήστε τις παρακάτω 5 πράξεις

**Q1:** Ποιοι Υπάλληλοι εργάζονται στο έργο P1?

**Q2:** Σε ποια έργα εργάζεται ο Υπάλληλος E1?

**INS:** Εισαγωγή πληροφοριών για ένα νέο έργο

**DEL:** Διαγραφή του γεγονότος ότι ο Υπάλληλος E3 εργάζεται στο έργο P4

**MOD:** Το έργο P2 μεταφέρθηκε από την Αθήνα στην Πάτρα

## ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (1)

- ΔΟΜΕΣ: ΔΕΝΤΡΑ (TREES) – Κόμβοι-Σύνδεσμοι
- ΠΡΑΞΕΙΣ: ΣΑΡΩΣΗ ΔΕΝΤΡΩΝ (προδιατεταγμένη)
- ΔΟΜΙΚΟΙ ΠΕΡΙΟΡΙΣΜΟΙ: *ελάχιστοι*, οι περισσότεροι έμφυτοι από την δομική μορφή
- Αντιστοιχίες με το E-R Μοντέλο

Τύπος Οντοτήτων <-> Τύπος Εγγραφής (Record Type)

Οντότητα <-> Εγγραφή (Record -Segment)

Γνώρισμα <-> Πεδίο (Field)

Συσχέτιση <-> Γονέας – Παιδί (Parent-to-child)

Οι Συσχετίσεις Γονέα-Παιδιού είναι **ΜΟΝΟ 1:N**

## ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (2)

### ■ ΙΕΡΑΡΧΙΚΟ ΣΧΗΜΑ

#### EMPLOYEE

SSN	Name	Address	Salary
-----	------	---------	--------

#### PROJECT

Number	Name	Location	HoursPW
--------	------	----------	---------

-- Ένα Ιεραρχικό Σχήμα είναι ένα Δέντρο όπου κάθε κόμβος είναι ένας Τύπος Εγγραφής

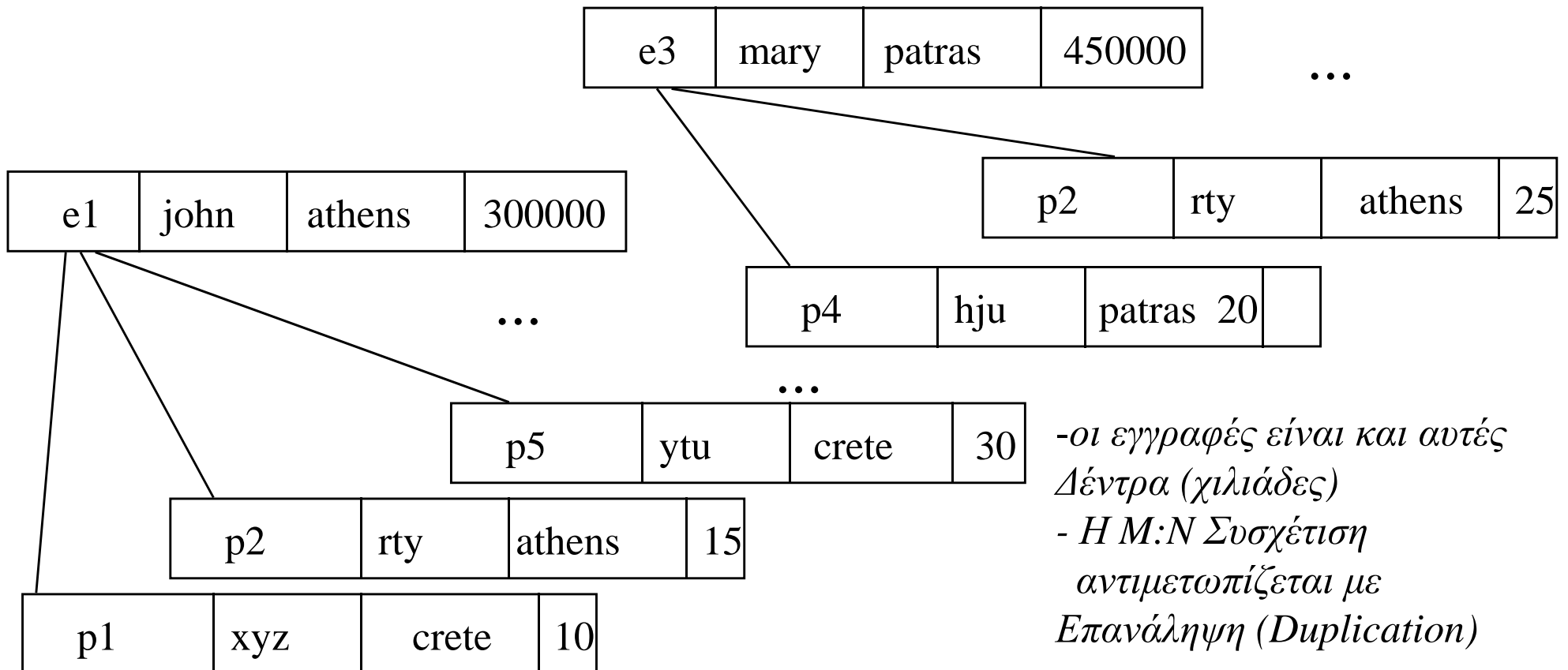
-Οι Σύνδεσμοι στο Δέντρο Παριστούν Συσχετίσεις (μόνο 1:N)

-Η Σχετική τοποθέτηση των κόμβων στο Δέντρο υποδηλώνει και την κατεύθυνση /βαθμό της Συσχέτισης (Γονέα-Παιδιού)

- Προσέξτε ότι το Γνώρισμα HoursPW της Συσχέτισης WORKS-ON έχει μεταφερθεί στο PROJECT

## ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (3)

- Παράδειγμα ΒΔ: Ένα ΔΑΣΟΣ από δέντρα



-οι εγγραφές είναι και αυτές Δέντρα (χιλιάδες)  
- Η M:N Συσχέτιση αντιμετωπίζεται με Επανάληψη (Duplication)

## ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (4)

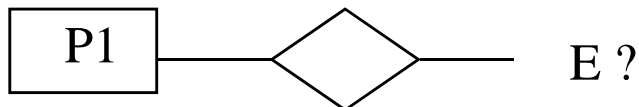
- **ΠΡΑΞΕΙΣ: Διάσχιση / Σάρωση Δέντρων**
  - *get next tree* (δεδομένου τύπου)
  - *get next child* (μιας δεδομένης εγγραφής)
  - προδιατεταγμενη διάσχιση (*pre-order tree-traversal*)
  - Πρόσβαση *RECORD-AT-A-TIME*
- Τα παραδείγματα Q1 και Q2 (ερωταποκρίσεις), που είναι πλήρως **ΣΥΜΜΕΤΡΙΚΑ** εκτελούνται στο Μοντέλο (DML) με έναν εντελώς **ΑΣΥΜΜΕΤΡΙΚΟ** τρόπο
- Αυτό έχει να κάνει με την επιλογή του **EMPLOYEE** ως Γονέα και του **Project** ως Παιδιού (θα μπορούσε να ήταν αντίστροφα)



## ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (5)

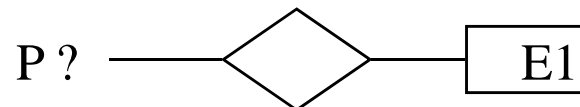
### • Q1: Employees of P1

```
do while there are still  
more employees  
get next EMPLOYEE  
get next PROJECT child  
where Number = "P1"  
if found then  
print EMPLOYEE.Name  
end
```



### Q2: Projects for E1

```
get next EMPLOYEE  
where SSN = "E1"  
do while there are still more  
project children of employee  
get next PROJECT child  
print PROJECT.Name  
end
```



## ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (6)

- **ΤΡΟΠΟΠΟΙΗΣΕΙΣ**

**INS:** **Εισαγωγή πληροφοριών για ένα νέο έργο**  
- δεν είναι δυνατόν να εισαχθούν πληροφορίες για ένα νέο έργο , αν δεν υπάρχει κάποιος Υπάλληλος ήδη εργαζόμενος σε αυτό (δεν υφίσταται Παιδί χωρίς Γονέα).

**DEL:** **Διαγραφή του γεγονότος ότι ο Υπάλληλος E3 εργάζεται στο έργο P4**  
- Βρες το Δέντρο με τον E3 στην κορυφή και διέγραψε το παιδί P4  
- ΠΡΟΒΛΗΜΑ: τι γίνεται αν είναι ο τελευταίος υπάλληλος στο P4 ?

**MOD:** **Το έργο P2 μεταφέρθηκε από την Αθήνα στην Πάτρα**  
- Βρες ΚΑΘΕ P2 εγγραφή και άλλαξε την τιμή της Πόλης

## Μοντέλο Δικτύου (DBTG-Network) (1)

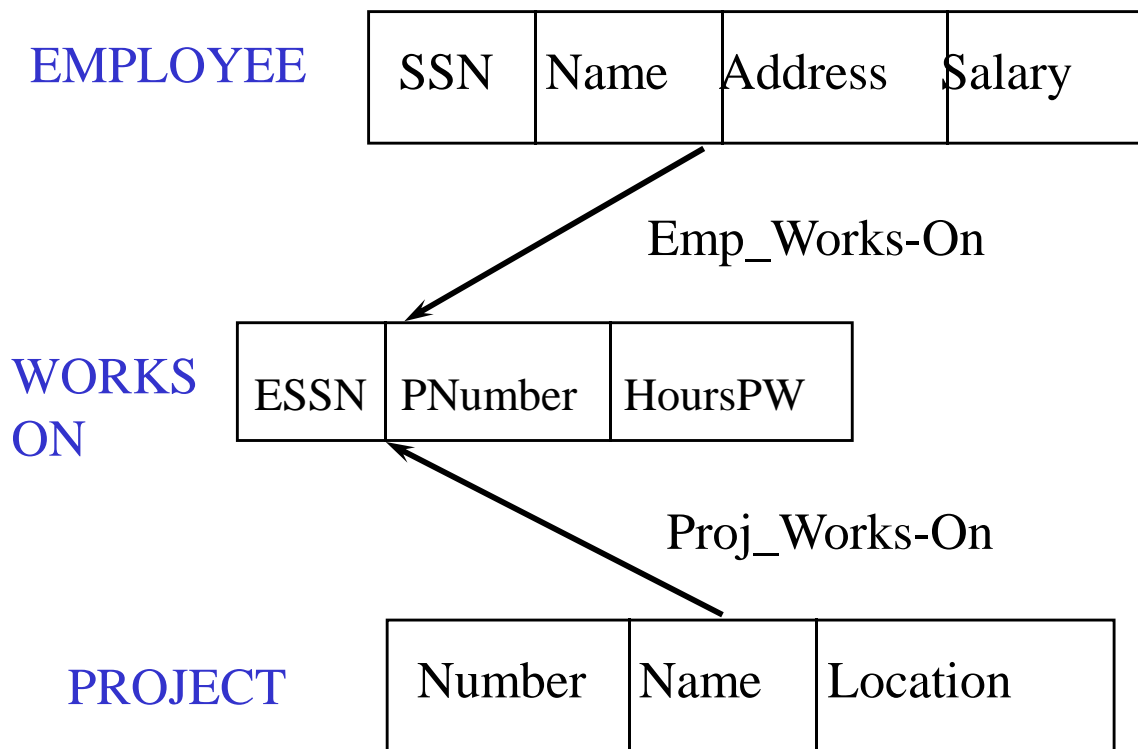
- **ΔΟΜΕΣ:** Γράφοι (**GRAPHS**) με Εγγραφές  
Και Συνδέσμους
- **ΠΡΑΞΕΙΣ:** Τυχαία Σάρωση Γράφων (**GRAPH-TRAVERSAL**)
- **ΔΟΜΙΚΟΙ ΠΕΡΙΟΡΙΣΜΟΙ:** *αρκετοί*, μερικοί *έμφυτοι* αλλά και *πολλοί ρητοί*.
- **Αντιστοιχίες με το E-R Μοντέλο**

Τύπος Οντοτήτων	<->	Τύπος Εγγραφής (Record Type)
Οντότητα	<->	Εγγραφή (Record)
Γνώρισμα	<->	Πεδίο (Data Item)
Συσχέτιση	<->	Τύπος Συνόλων (DBTG-set type) συσχετίσεις Ιδιοκτήτη-Μέλους (1:N)

- **Εγγραφές-Σύνδεσμοι** χρησιμοποιούνται για N:M συσχετίσεις

## Μοντέλο Δικτύου (DBTG-Network) (2)

### ■ DBTG-Network (CODASYL) Σχήμα



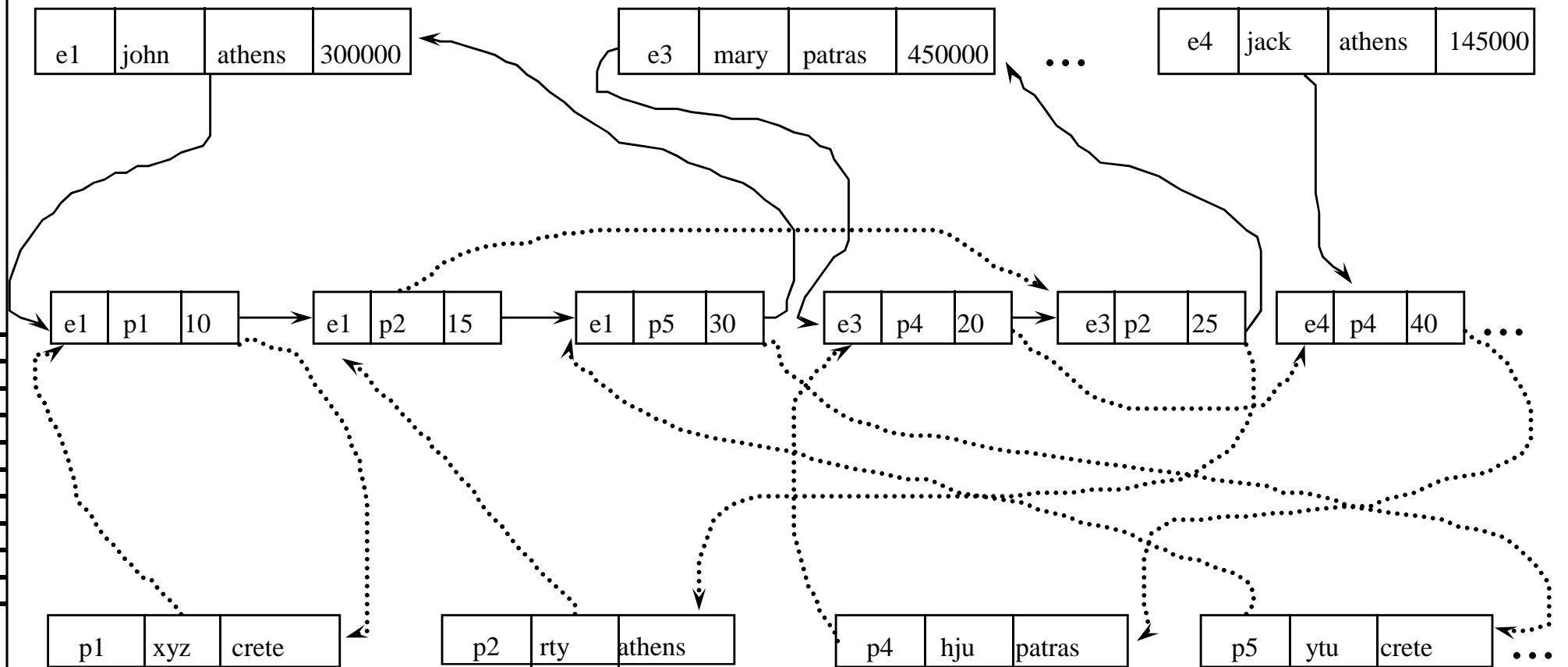
- Το Σχήμα στο Μοντέλο  
Είναι είναι Δίκτυο, όπου κάθε  
Κόμβος είναι Τύπος Εγγραφής

-Σύνδεσμοι παριστούν Συσχετίσεις  
(μόνο 1:N) και έχουν Όνομα και  
Κατεύθυνση (OWNER - MEMBER)

-Ειδικό Τύποι Εγγραφών είναι  
-Απαραίτητοι για Σχέσεις N:M  
(Connectors)

# Μοντέλο Δικτύου (DBTG-Network) (3)

## ■ Παράδειγμα: ΒΔ Δικτύου (Σπαγγέτι)



## Μοντέλο Δικτύου (DBTG-Network) (4)

- Πράξεις: **GRAPH Traversal** (Navigation)
  - *find next record where ...*
  - *find next record within set type ...*
  - Τυπική ταξινομημένη Διάσχιση (*first, next, previous, last, κλπ.*)
  - Χρήση ΠΟΛΛΩΝ Δεικτών
  - Πρόσβαση *RECORD-AT-A-TIME*
- Τα παραδείγματα Q1 και Q2 (ερωταποκρίσεις), που είναι πλήρως **ΣΥΜΜΕΤΡΙΚΑ** εκτελούνται στο Μοντέλο (DML) με έναν εντελώς **ΣΥΜΜΕΤΡΙΚΟ** τρόπο

## Μοντέλο Δικτύου (DBTG-Network) (5)

• Q1: Employees of P1

```
find next PROJECT record  
  where Number = "P1"  
  
do while there are still more  
  connector records under project  
  find next WORKS-ON record  
    in Proj_Works-On set  
  find (owner) EMPLOYEE record  
    in Emp_Works-On set  
  print EMPLOYEE.Name  
end
```

Q2: Projects for E1

```
find next EMPLOYEE record  
  where SSN = "E1"  
  
do while there are still more  
  connector records under employee  
  find next WORKS-ON record  
    in Emp_Works-On set  
  find (owner) PROJECT record  
    in Proj_Works-On set  
  print PROJECT.Name  
end
```

## Μοντέλο Δικτύου (DBTG-Network) (6)

- **ΤΡΟΠΟΠΟΙΗΣΕΙΣ**

**INS:** **Εισαγωγή πληροφοριών για ένα νέο έργο**

- Δημιούργησε μια νέα εμφάνιση του PROJECT
- στην αρχή, δεν υπάρχουν σύνδεσμοι, (Οι Ιδιοκτήτες υπάρχουν χωρίς Μέλη,)

**DEL:** **Διαγραφή του γεγονότος ότι ο Υπάλληλος E3 εργάζεται στο έργο P4**

- διέγραψε την εγγραφή-σύνδεσμο (Works-On) για με τις κατάλληλες τροποποιήσεις Δεικτών

**MOD:** **Το έργο P2 μεταφέρθηκε από την Αθήνα στην Πάτρα**

- Βρες τη μοναδική εγγραφή P2 και άλλαξε την τιμή Πόλης



## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (1)

- **ΔΟΜΕΣ: RELATIONS (ΣΧΕΣΕΙΣ)** – μία μόνο Δομή
- **ΠΡΑΞΕΙΣ: Επεξεργασία των Σχέσεων (Πινάκων)**
- **ΔΟΜΙΚΟΙ ΠΕΡΙΟΡΙΣΜΟΙ:** *λίγοι έμφυτοι* (π.χ, σύνολα), *πολλοί* ρητοί
- **Αντιστοιχίες με το E-R Μοντέλο**

Τύπος Οντοτήτων	<->	Τύπος Σχέσης (Relation)
Οντότητα	<->	n-πλειάδα (Tuple)
Γνώρισμα	<->	Γνώρισμα (Attribute)
Συσχέτιση	<->	Τύπος Σχέσης (Relation)

**Ένα ΑΠΛΟ μοντέλο με ΙΣΧΥΡΕΣ (τυπικές) Ρίζες**

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (2)

### ■ Σχεσιακό Σχήμα

EMPLOYEE

SSN	NAME	ADDRESS	SALARY
-----	------	---------	--------

WORKS-ON

ESSN	PNUMB	HRSPW
------	-------	-------

PROJECT

NUMBER	NAME	LOCATION
--------	------	----------

-- Ένα Σχεσιακό Σχήμα είναι  
ένα Σύνολο Σχέσεων, ή  
Πινάκων (με όνομα)

- Τα ονόματα των στηλών  
είναι τα ονόματα των  
Γνωρισμάτων

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (3)

### ■ Παράδειγμα ΒΔ: Σύνολο Στιγμιότυπων

EMPLOYEE

SSN	NAME	ADDRESS	SALARY
e1	john	athens	300000
e3	mary	patras	450000
e4	jack	athens	145000

WORKS-ON

ESSN	PNUMB	HRSPW
e1	p1	10
e1	p2	15
e1	p5	30
e3	p4	20
e4	p4	40
e3	p2	25

PROJECT

NUMBER	NAME	LOCATION
p1	xyz	crete
p2	rty	athens
p4	hju	patras
p5	ytu	crete

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (4)

- **Operations: Set-Theoretic Simple Operations**
  - *select records from* RELATION(s) *where ...*
  - Το *where-clause* μπορεί να είναι Σύνθετο, με πολλούς τελεστές
  - Δεν υπάρχει σειρά (order) στην πρόσβαση σε δεδομένα
  - Πρόσβαση SET-AT-A-TIME στις εγγραφές
- Τα παραδείγματα Q1 και Q2 (ερωταποκρίσεις), που είναι πλήρως ΣΥΜΜΕΤΡΙΚΑ εκτελούνται στο Μοντέλο (DML) με έναν εντελώς ΣΥΜΜΕΤΡΙΚΟ τρόπο

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (5)

• Q1: Employees of P1

```
select from EMPLOYEE,  
WORKS-ON relations  
where (WORKS-ON.Pnumb = "P1")  
      and (WORKS-ON.ESSN =  
          EMPLOYEE.SSN)  
print EMPLOYEE.Name
```

Q2: Projects for E1

```
select from PROJECT,  
WORKS-ON relations  
where (WORKS-ON.ESSN= "E1")  
      and (WORKS-ON.Pnumb =  
          PROJECT.Number)  
print PROJECT.Name
```

Οι πράξεις στο Σχεσιακό Μοντέλο είναι ΚΛΕΙΣΤΕΣ (CLOSED)  
Δηλαδή, παίρνουν ΣΧΕΣΕΙΣ και παράγουν πάλι ΣΧΕΣΕΙΣ

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Εισαγωγή (6)

### ■ ΤΡΟΠΟΠΟΙΗΣΕΙΣ

- INS:**            **Εισαγωγή** πληροφοριών για ένα νέο έργο
- Εισαγωγή μιας *n*-πλειάδας στη Σχέση *PROJECT*
- DEL:**            **Διαγραφή** του γεγονότος ότι ο Υπάλληλος *E3*
- εργάζεται** στο έργο *P4*
- Διαγραφή της *n*-πλειάδας (*e2, p4, 20*) από την Σχέση *WORKS-ON*
- MOD:**            **Το έργο P2 μεταφέρθηκε** από την *Αθήνα* στην
- Πάτρα**
- Επέλεξε την *n*-πλειάδα (*p2, rty, athens*) από το *PROJECT*, και άλλαξε την τιμή “*athens*” σε “*patras*”

## Συγκρίσεις Μεταξύ Μοντέλων

- Η βασική διαφοροποίηση μεταξύ των τριών βασικών μοντέλων είναι το ***ΠΩΣ ΠΑΡΙΣΤΑΝΤΑΙ ΟΙ ΣΥΣΧΕΤΙΣΕΙΣ***
  - **ΣΧΕΣΙΑΚΟ**: Ρητά με τις ΣΧΕΣΕΙΣ (εφόσον η συσχέτιση είναι N:M) ή / και με έμμεσες συνδέσεις μεταξύ Σχέσεων που παρίστανται με τα Εξωτερικά Κλειδιά (*foreign keys.* )
  - **ΔΙΚΤΥΟΥ**: Ρητές συνδέσεις με ΣΥΝΔΕΣΜΟΥΣ (*links*) και εγγραφές σύνδεσης. ΣΧΟΛΙΟ: *Εκφραστικότητα αλλά και εγγενής πολυπλοκότητα*
  - **ΙΕΡΑΡΧΙΚΟ** (ειδική περίπτωση Δικτύου): Ρητές συνδέσεις με ΣΥΝΔΕΣΜΟΥΣ. Η περίπτωση των N:M συσχετίσεων γίνεται με διπλασιασμό εγγραφών. ΣΧΟΛΙΟ: *σημαντικές και σοβαρές αδυναμίες στην έκφραση μη-ιεραρχικών δομών.*

## Βάση Δεδομένων για Παραγγελίες (Α) στο Σχεσιακό Μοντέλο

- **Employees** - Employee ID, First Name, Last Name, Title, Work Phone
- **Customers** - Customer ID, Company Name, First Name, Last Name, Billing Address, City, State/Province, ZIP Code, Web site, Phone Number, Fax Number, Ship Address, Ship City, Ship State/Province, Ship ZIP Code, Phone Number
- **Products** - Product ID, Product Name, Unit Price
- **Orders** - Order ID, Customer ID, Employee ID, Order Date, PO Number, Ship Date, Shipping Method ID, Sales Tax Rate
- **Order Details** - Order Detail ID, Order ID, Product ID, Quantity, Unit Price, Price Discount



## Βάση Δεδομένων για Παραγγελίες (B) στο Σχεσιακό Μοντέλο

- **Payments** - **Payment ID, Order ID, Payment Amount, Payment Date, Status, Credit Card #, Cardholder Name, Card Exp. Date, Payment Method, Credit Card?**
- **Shipping Methods** - **Shipping Method ID, Shipping Method**
- **Our Company Info** - **SetupID, Company Name, Address, City, State/Province, Postal Code, Phone Number, Fax Number**