



# Διάλεξη 03: Εννοιολογική Σχεδίαση Βάσης Δεδομένων I (Entity Relationship Modelling)

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:  
Εισαγωγή στις έννοιες:

- Μεθοδολογία Ανάπτυξης Βάσεων Δεδομένων
- Εννοιολογική Σχεδίαση Βάσης (ER Diagram-Entities)
- Παράδειγμα Ανάλυσης Απαιτήσεων Συστήματος και μετατροπή σε ER

**Διδάσκων: Παναγιώτης Ανδρέου**

# Διαδικασία Ανάπτυξης Βάσεων

Για την ανάπτυξη μιας Σχεσιακής Βάσης δεδομένων ακολουθούνται τα ακόλουθα βήματα

- A. Ανάλυση Απαιτήσεων (Requirement Analysis):** Ο Αναλυτής του συστήματος προετοιμάζει ένα έγγραφο απαιτήσεων με την βοήθεια του πελάτη.
- Τι **δεδομένα** να αποθηκεύονται και που;
  - Τι **εφαρμογές** θα χτιστούν πάνω από τα δεδομένα;
  - Ποιες **πράξεις/διαδικασίες** πρέπει να υποστηριχθούν και με τι **απαιτήσεις επίδοσης**;
- B. Εννοιολογική Σχεδίαση Βάσης (Conceptual Database Design):** Ο **DB Designer** ετοιμάζει ένα **ER διάγραμμα** το οποίο μπορεί να γίνει αντιληπτό στο πελάτη (για επικύρωση)
- Το διάγραμμα αυτό πρέπει να **ορθό, πλήρες και αποδοτικό** για να είναι εύκολη η μετατροπή στο επόμενο στάδιο.

# Διαδικασία Ανάπτυξης Βάσεων (συν.)

**C. Λογική Σχεδίαση Βάσης (Logical Database Design):** Μετατροπή του ER διαγράμματος σε ένα (Σχεσιακό) Σχήμα Βάσης.

- Μπορεί να γίνει και αυτόματα με κάποιο αλγόριθμο / εργαλείο

**D. Εκλέπτυνση Σχήματος (Schema Refinement):**

Ανάλυση σχήματος για εντοπισμό ενδεχόμενων προβλημάτων

- π.χ., Student(ssn, vehicleID, motorID).
  - Αποφυγή πολλών κενών τιμών π.χ., Ο Κώστας ΔΕΝ έχει αυτοκίνητο δημιουργεί την μη επιθυμητή εγγραφή (111, NULL, NULL);

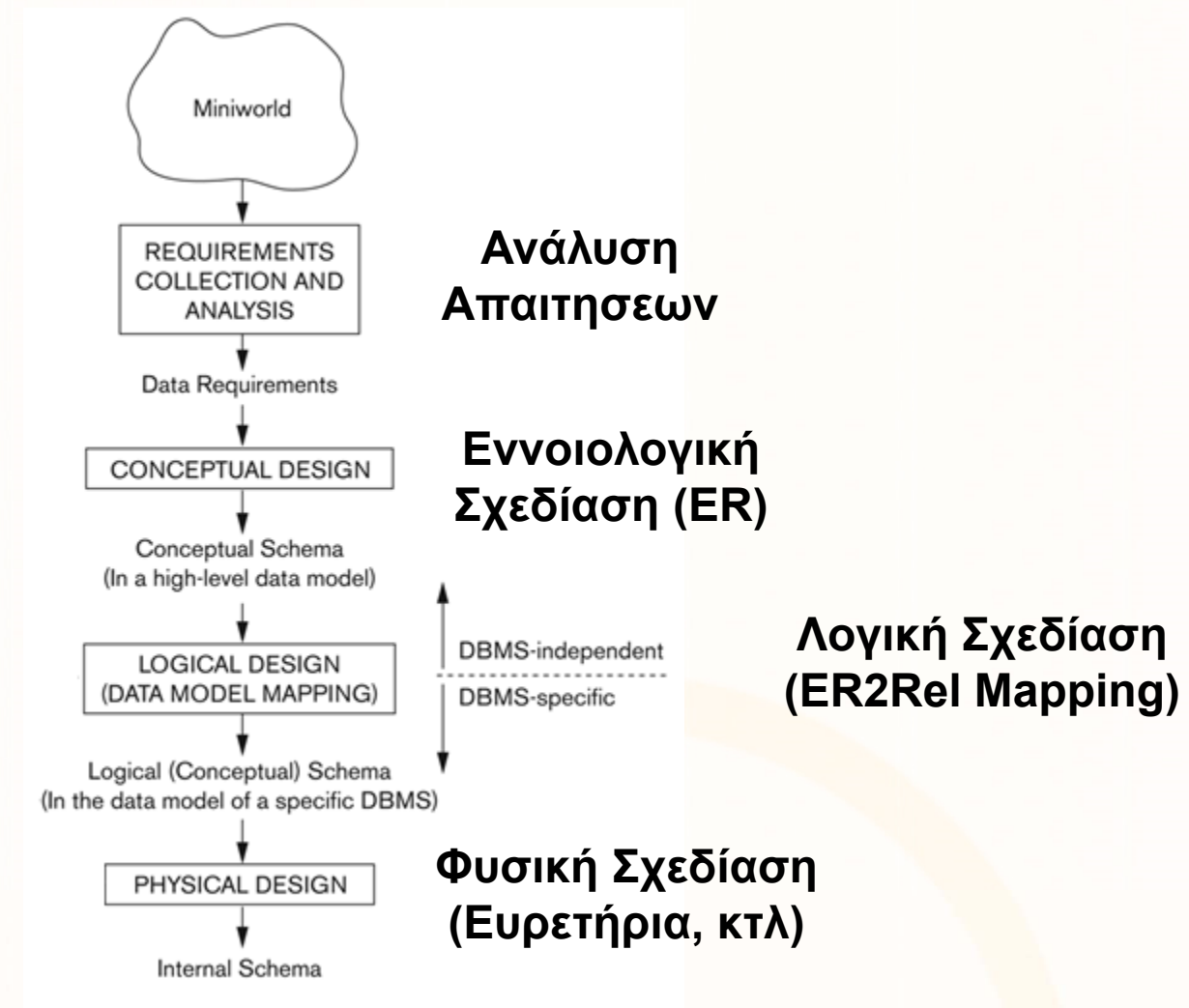
**E. Φυσικός Σχεδιασμός Βάσης (Physical Database Design)**

Έλεγχος ότι η βάση θα ικανοποιεί τις απαιτήσεις επίδοσης/φόρτου κτλ. τα οποία επιβάλλουν οι απαιτήσεις του εγγράφου ανάλυσης.

- Εάν δεν ικανοποιούνται τότε σε αυτό το στάδιο λαμβάνονται διορθωτικά μέτρα (π.χ., προσθήκη ευρετηρίων, κτλ).

**F. Υλοποίηση Εφαρμογής και Μηχανισμών Ασφάλειας (Application & Security Design)**

# Διαδικασία Ανάπτυξης Βάσεων (συν.)



# Ανάλυση Απαιτήσεων

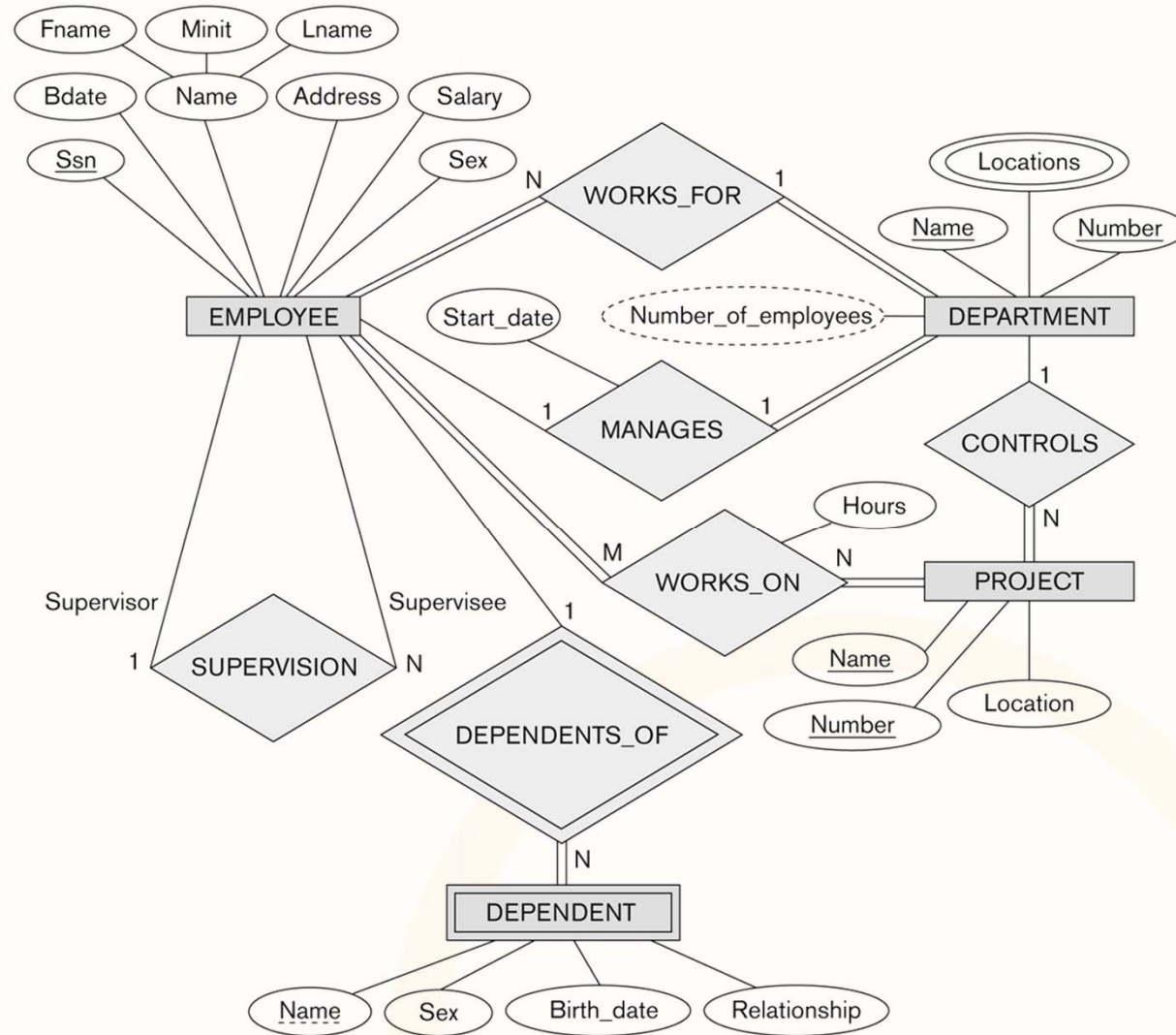
- Θεωρήστε τις ακόλουθες **Απαιτήσεις (Requirements)** μιας υποθετικής βάσης **COMPANY** για την οποία πρέπει να κάνουμε την **Εννοιολογική Σχεδίαση**:
  - Η **COMPANY** οργανώνεται σε **DEPARTMENTS**. Κάθε department έχει τα ακόλουθα πεδία:
    - **name, number** και ένα **manager** που διαχειρίζεται το **department, start date** του department manager. Κάθε **department** μπορεί να έχει πολλαπλά **locations**.
  - Κάθε **department** *ελέγχει ένα αριθμό από* **PROJECTS**. Κάθε project έχει ένα **μοναδικό name, μοναδικό number** και βρίσκεται σε μια μόνο τοποθεσία (location).

# Ανάλυση Απαιτήσεων (συν.)

- Για κάθε **EMPLOYEE** αποθηκεύουμε το social security number (**SSN** – ταυτότητα), **address**, **salary**, **sex**, και **birthdate**.
  - Κάθε **employee** δουλεύει για ένα **department** αλλά μπορεί να **εργάζεται πάνω σε πολλά projects**.
  - Για κάθε **project** καταγράφουμε τον **χρόνο** (number of hours per week) τα οποία αφιερώνει ένας employee πάνω σε κάθε project.
  - Επίσης καταγράφουμε τον **προϊστάμενο** (*direct supervisor*) **κάθε employee**.
- Κάθε employee μπορεί να έχει ένα αριθμό από **Εξαρτώμενα (DEPENDENTS)**, δηλ., παιδιά.
  - Για κάθε depend, κρατούμε το **name**, **sex**, **birthdate**, και τον τύπο της σχέσης (**relationship**) με τον employee.

# Εννοιολογική Σχεδίαση

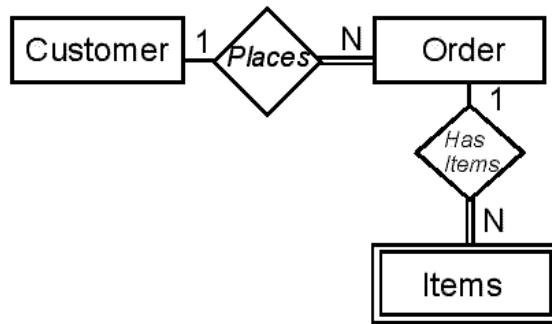
- Παράδειγμα: Entity Relationship Diagram (ERD)



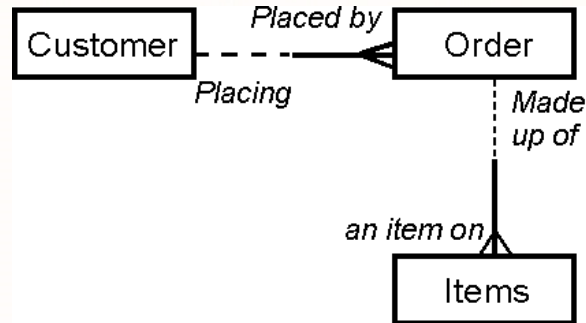
# Εννοιολογική Σχεδίαση

- Άλλα Παραδείγματα Συμβολισμών

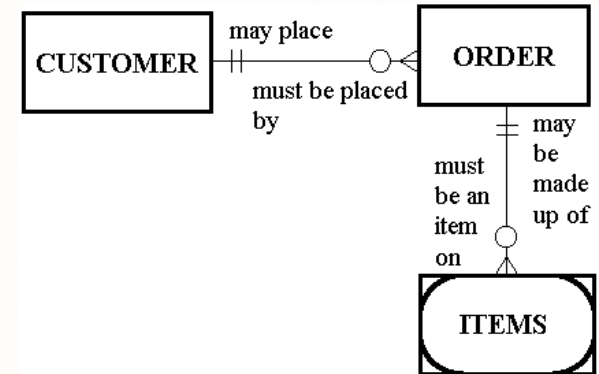
Chen



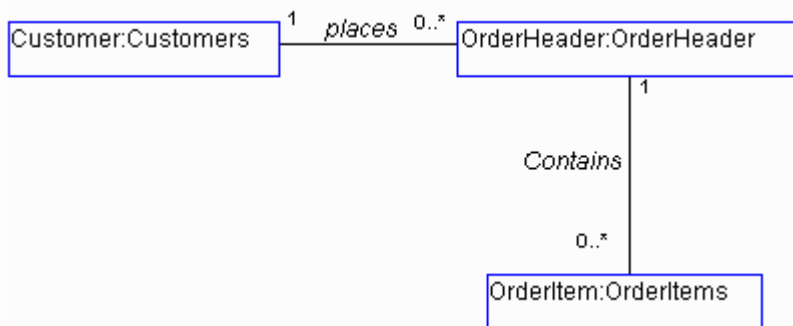
CASE



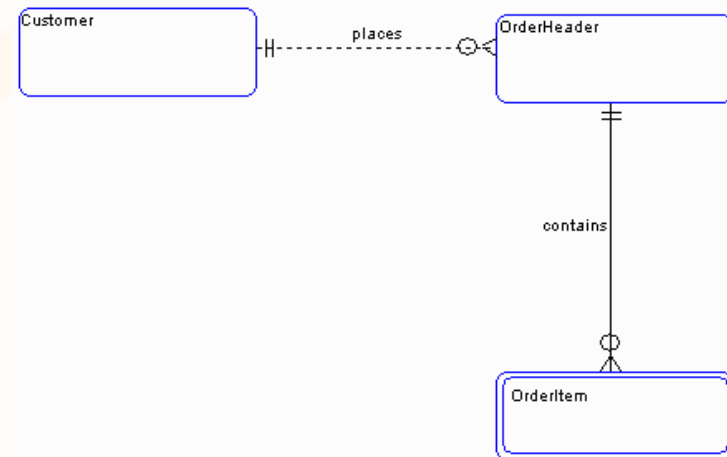
Visible Analyst



UML



Crows Feet





# Λογική Σχεδίαση

- Παράδειγμα: Relational Diagram (RD)

## STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

## COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

## PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

## SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

## GRADE\_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

# Διαδικασία Σχεδίασης

- Τώρα θα μελετήσουμε αναλυτικότερα τη φάση της σχεδίασης για την οποία υπάρχουν δυο βασικές διαδικασίες:
  - **Σχεδιασμός Βάσης (Database design)**
  - **Σχεδιασμό Εφαρμογής (Applications design)**
- Ο στόχος μας σε αυτή την ενότητα επικεντρώνεται στο πρώτο, δηλ., επικέντρωση στο σχεδιασμό του **εννοιολογικού σχήματος** μιας βάσης.
  - Ένα τέτοιο σχήμα θα αντικατοπτρίζει τις **προδιαγραφές** του **χρήστη (πελάτη)** όπως αυτές έχουν προκύψει από την **ανάλυση απαιτήσεων**.
  - Ο σχεδιασμός της **εφαρμογής** και των **συστημάτων διαπροσωπείας** γενικά μελετάται στο πλαίσιο της **Τεχνολογίας Λογισμικού**.

# ER Διαγράμματα

- Διαγράμματα τα οποία χρησιμοποιούνται για την **εννοιολογική αναπαράσταση** δεδομένων.
- Ορίζεται το 1976 από τον Prof. Peter Chen
  - *“The Entity-Relationship Model--Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, Pages 9 – 36”*
- Θα αναφερθούμε σε αυτά τα διαγράμματα ως
  - Διάγραμμα Οντοτήτων Συσχετίσεων (ΔΟΣ) ή ER Diagram (ERD)
- Υπάρχουν διάφοροι (όμοιοι) συμβολισμοί οι οποίοι χρησιμοποιούνται.
  - Στο μάθημα (εργασίες, εξετάσεις, κτλ) θα χρησιμοποιήσουμε την σημειογραφία (notation) του βιβλίου.
- Παραλλαγές τέτοιων διαγραμμάτων χρησιμοποιούνται σε διάφορα εμπορικά εργαλεία για την ανάπτυξη βάσεων δεδομένων.

# ER: Οντότητες, Συσχετίσεις και Γνωρίσματα

## 1) Οντότητες και Γνωρίσματα

- **Οντότητες (Entities):** είναι συγκεκριμένα αντικείμενα του mini-world που αναπαριστώνται από την βάση.

COURSE

- Π.χ., Ο EMPLOYEE Γιάννος Ιωάννου, Το DEPARTMENT Πληροφορικής, το COURSE ΕΠΛ342

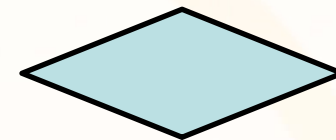
- **Attributes (Γνωρίσματα, Πεδία ή Χαρακτηριστικά):** είναι ιδιότητες που περιγράφουν μια οντότητα.

Name

- Π.χ., Μια **οντότητα EMPLOYEE** έχει τα **γνωρίσματα** Name, SSN (Social Security Number – Ταυτότητα), Address, Sex, BirthDate
- Κάθε γνώρισμα έχει κάποιο **τύπο δεδομένων** (data type), integer, string, binary, bit, κτλ.

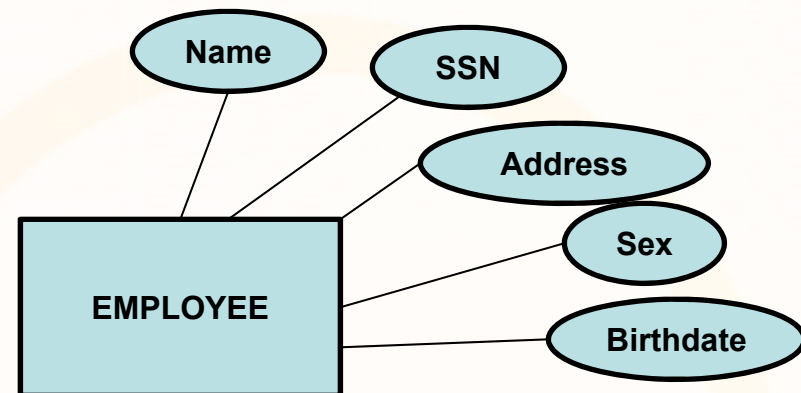
## 2) Συσχετίσεις (Relationships)

- Θα μελετηθούν την επόμενη διάλεξη



# ER: Οντότητες και Γνωρίσματα

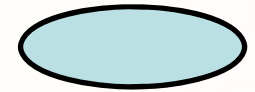
- Μια **οντότητα** έχει τιμές για τα **πεδία** της:
  - Π.χ., Οντότητα EMPLOYEE A:
    - Name='John Smith',
    - SSN='123456789',
    - Address = '16, Ιφιγενείας, 1890 Λευκωσία',
    - Sex='M',
    - BirthDate='09-JAN-55'
- **Διαγραμματικά**, τα γνωρίσματα συνδέονται με την οντότητα μέσω **ακμών**



# ER: Γνωρίσματα

- **Απλά (ή Μονότιμα Simple)**

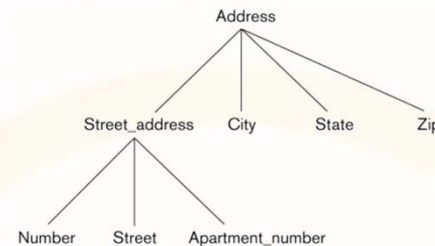
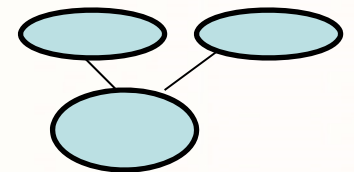
- Το γνώρισμα έχει απλή τιμή π.χ., SSN='123456' or Sex='M'



- **Σύνθετα (Composite)**

- Το γνώρισμα περιέχει πολλαπλά πεδία:

- Address(Apt#, House#, Street, City, State, ZipCode, Country), ή
- Name(FirstName, MiddleName, LastName).
- Η **σύνθεση** μπορεί να δημιουργήσει μια **ιεραρχία** όπου κάποια γνωρίσματα είναι και αυτά **σύνθετα** (εάν και όπως θα δούμε αργότερα θα αποφεύγεται αυτή η λογική)



- **Πλειότιμα (Multi-valued)**

- Το γνώρισμα μπορεί να πάρει πολλαπλές απλές τιμές
- Π.χ., Χρώμα Αυτοκινήτου: {red, white}.
- Προηγούμενα Πτυχία Ατόμου: { }, {bsc,msc}, {bsc,phd},



# ER: Τύποι Οντοτήτων και Κλειδιά

- **Τύποι Οντοτήτων (Entity Types):**

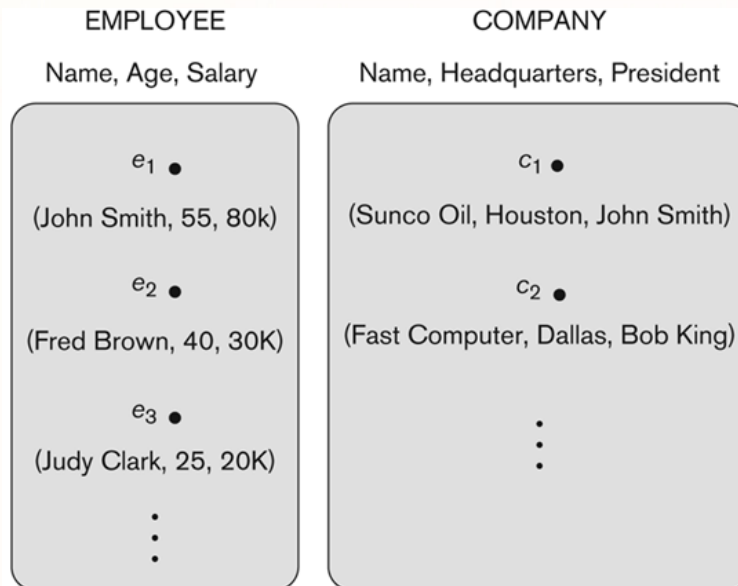
- Το σχήμα (δομή) μιας οντότητας.
- Π.χ., STUDENT(SSN, name, age)

ENTITY

- **Σύνολο Οντοτήτων (Entity Set ή Entity State)**

- Ένα σύνολο οντοτήτων με τα ίδια χαρακτηριστικά
- π.χ., {Student1, Student2, ...} κτλ.

Τύποι  
Οντοτήτων



Σύνολο  
Οντοτήτων

Συνήθως θα αναφερόμαστε και στα: **Οντότητα, Τύπος Οντότητας, Σύνολο Οντότητας** ως **Οντότητα** (εκτός και εάν γίνεται ρητή διαφοροποίηση)



# ER: Κλειδιά

- Κάθε Οντότητα ΠΡΕΠΕΙ να έχει ένα χαρακτηριστικό το οποίο θα μας επιτρέπει να το αναγνωρίζουμε **ΜΟΝΑΔΙΚΑ**
  - Π.χ., **SSN για Student.**
  - Το πεδίο που αναγνωρίζει μοναδικά μια Οντότητα ονομάζεται **Κλειδί της Οντότητας (Key)**
- Ένα κλειδί (key) μπορεί να είναι **σύνθετο**
  - Π.χ., VehicleTagNumber = (Number, State)
  - Ένα κλειδί υπογραμμίζεται στο διάγραμμα
- Άλλες Λεπτομέρειες:
  - Μια οντότητα μπορεί να έχει **πάνω από 1 κλειδιά**, π.χ.,:
    - Ο Αριθμός **Πλαισίου του Αυτοκινήτου** (VehicleIdentificationNumber) ή Ο Αριθμός **Πινακίδας του Αυτοκινήτου** (VehicleTagNumber) (Number, State).

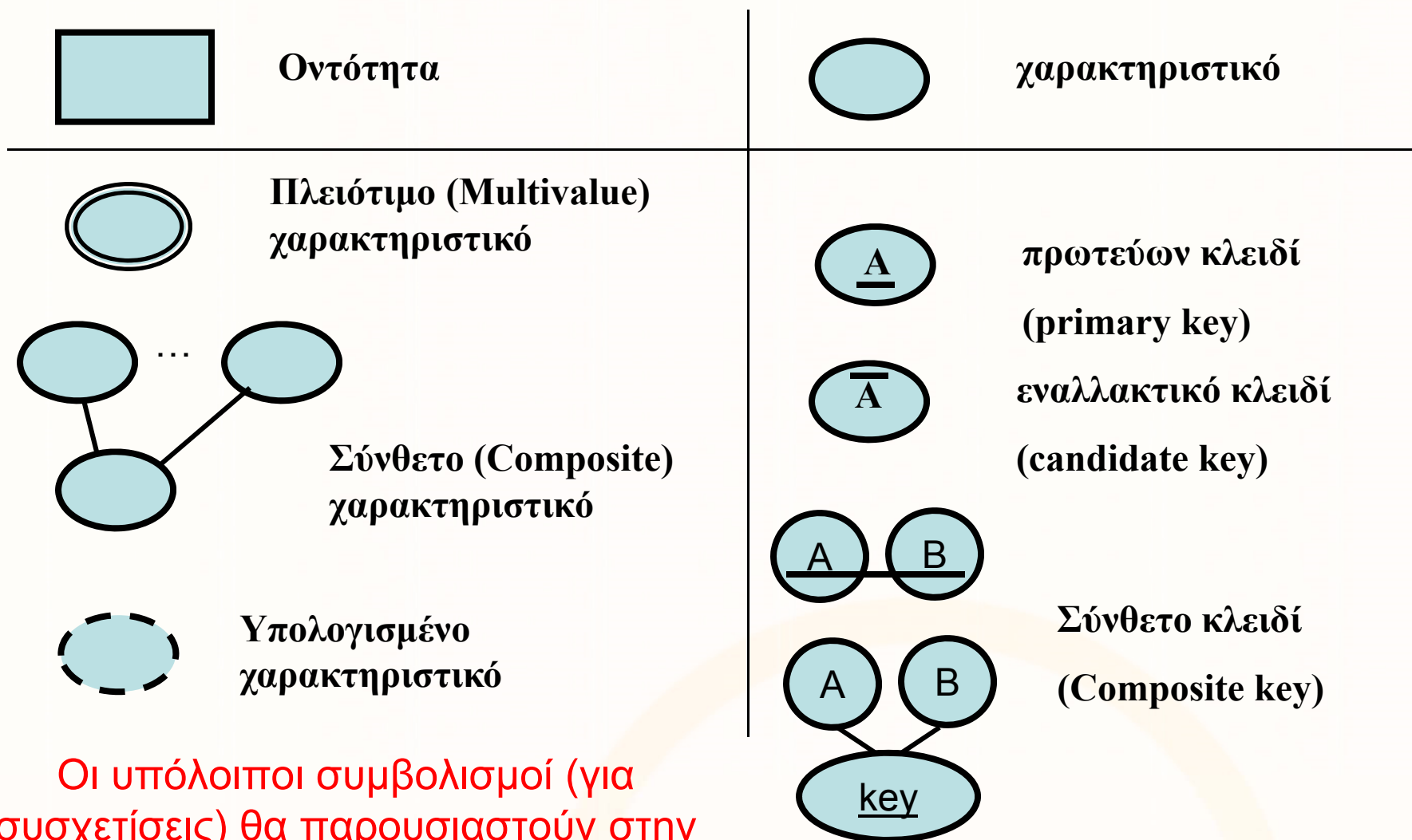




# ER: Κλειδιά

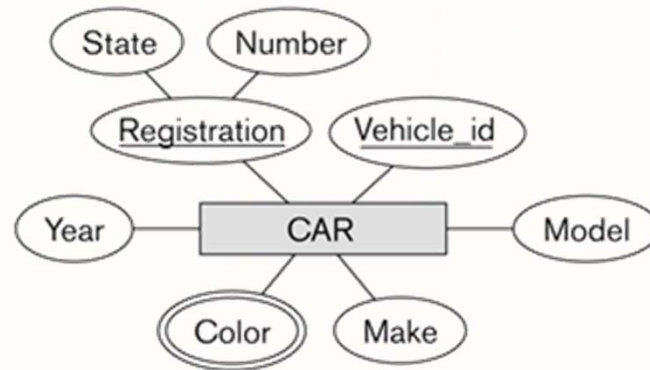
- **Πρωτεύων Κλειδί (Primary Key):** Ένα από τα κλειδιά τα οποία επιλέγεται από τον σχεδιαστή για να αντιπροσωπεύσει τη σχέση (SSN) 
  - **Unique Key (Μοναδικό Κλειδί):** Όμοια έννοια αλλά δεν εφαρμόζει τον περιορισμό NOT NULL ο οποίος ισχύει εξορισμού στο πρωτεύων κλειδί.
- **Υποψήφιο Κλειδί (Candidate Key) ή Εναλλακτικά κλειδιά:** τα εναπομείναντα κλειδιά 
  - Π.χ., *StudentID* αντί *SSN*
  - Το πρωτεύων κλειδί είναι υπογραμμισμένο, τα εναλλακτικά κλειδιά είναι *over-lined*
- **Υπερκλειδί (SuperKey):** Ένα σύνολο χαρακτηριστικών που περιλαμβάνουν τουλάχιστο 1 κλειδί της σχέσης.
  - Π.χ., (SSN, Name) ή (StudentID, Name, Age), κτλ.

# ER: Διαγράμματα



Οι υπόλοιποι συμβολισμοί (για συσχετίσεις) θα παρουσιαστούν στην επόμενη διάλεξη

# Παράδειγμα ER Διαγράμματος



CAR  
Registration (Number, State), Vehicle\_id, Make, Model, Year, {Color}

CAR<sub>1</sub>  
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR<sub>2</sub>  
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR<sub>3</sub>  
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Διάγραμμα  
ER

Τύπος  
Οντότητας

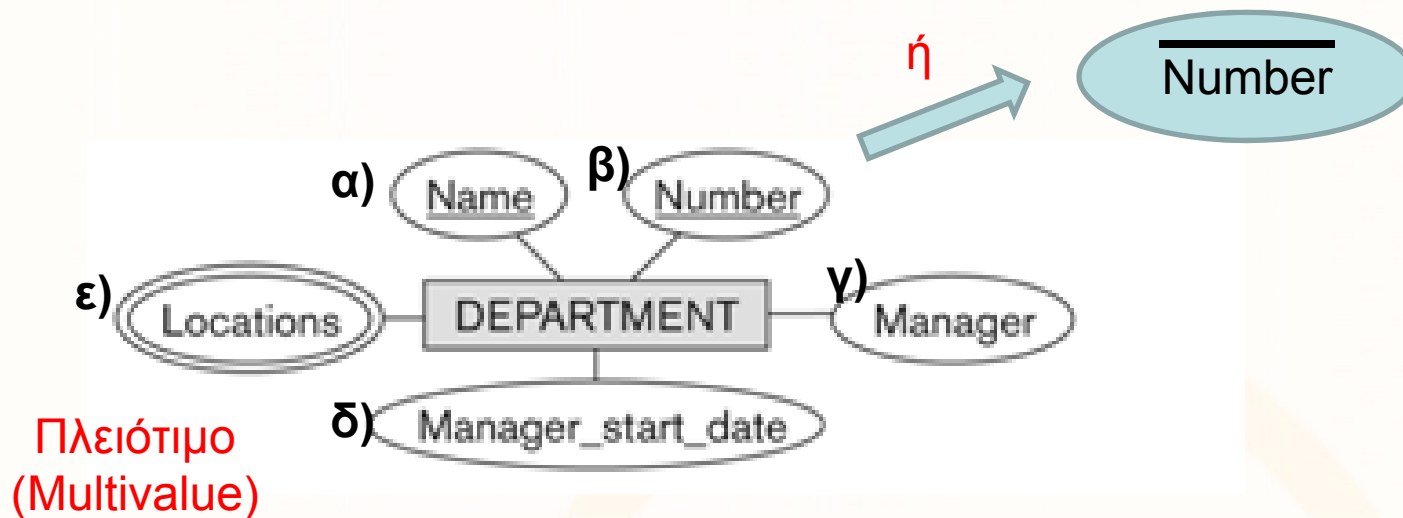
Σύνολο  
Οντότητας

# Σχεδίαση ER

- Βάση των **προδιαγραφών** διακρίνουμε τέσσερις τύπους οντοτήτων για την βάση **COMPANY**:
  - **DEPARTMENT**
  - **PROJECT**
  - **EMPLOYEE**
  - **DEPENDENT**
- Το πρώτο στάδιο της σχεδίασης περιλαμβάνει την **διαγραμματική απεικόνιση των οντοτήτων** μαζί με τα **επί μέρους γνωρίσματα** τους (όπως αυτά δίνονται από τις **προδιαγραφές**).
- Ακολουθεί **αναλυτική απεικόνιση ...**

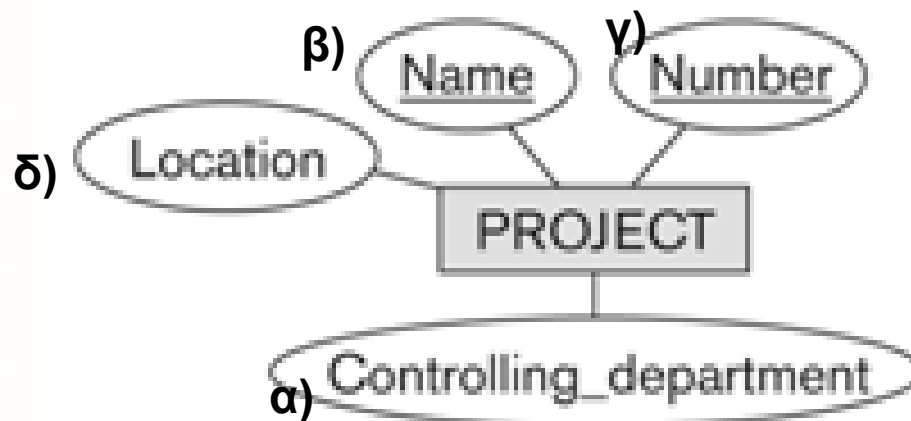
# Σχεδίαση ER

- Προδιαγραφή I: Κάθε Department έχει τα ακόλουθα πεδία: **α) Μοναδικό Name**, **β) Μοναδικό Number** και ένα **γ) Manager** που διαχειρίζεται το department, **δ) Start date** του department manager. Κάθε department μπορεί να έχει **ε) πολλαπλά Locations**.



# Σχεδίαση ER

- Προδιαγραφή II: **α)** Κάθε department ελέγχει ένα αριθμό από PROJECTs. **β)** Κάθε project έχει ένα μοναδικό name, **γ)** μοναδικό number και βρίσκεται σε μια **δ)** μόνο τοποθεσία (location).

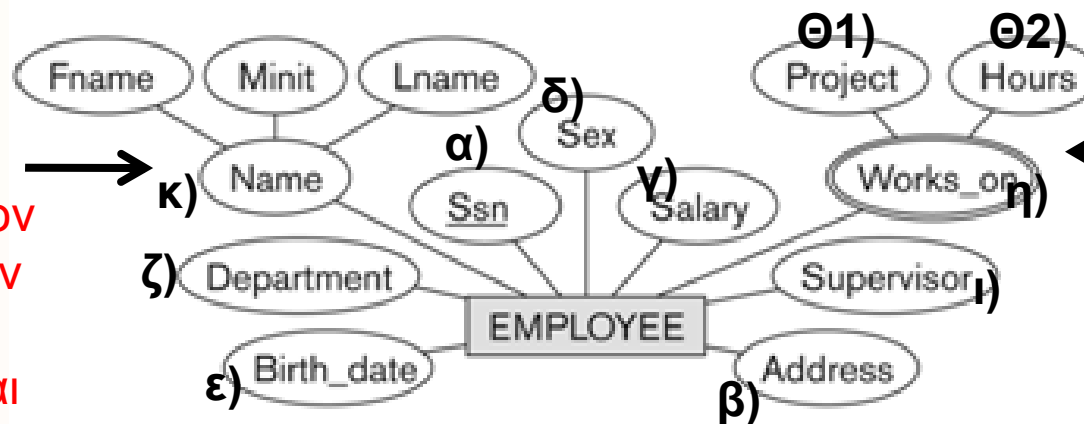


# Σχεδίαση ER

- **Προδιαγραφή III:** Για κάθε EMPLOYEE αποθηκεύουμε το **α)** social security number (SSN – ταυτότητα), **β)** address, **γ)** salary, **δ)** sex, και **ε)** birthdate.
  - Κάθε **employee ζ)** δουλεύει για ένα **department** αλλά μπορεί να **η)** εργάζεται πάνω σε πολλά **projects**.
  - **θ)** Για κάθε **project** καταγράφουμε τον χρόνο (num of hours per week) τα οποία αφιερώνει ένας employee πάνω σε κάθε project.
  - **ι)** Επίσης καταγράφουμε τον προϊστάμενο (*direct supervisor*) κάθε **employee** αλλά και το όνομα του Employee

Σύνθετο  
(Composite)

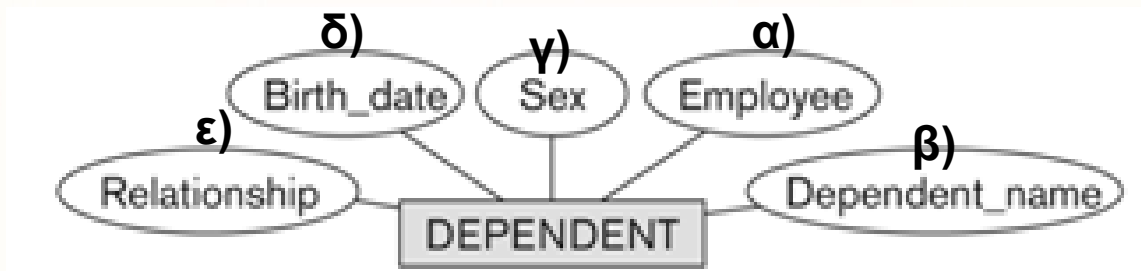
Μετά από  
συνεννόηση με τον  
πελάτη (αφού δεν  
ήταν αρχική  
απαίτηση να είναι  
σύνθετο)



Πλειότιμο &  
Σύνθετο  
(Multivalued &  
Composite)

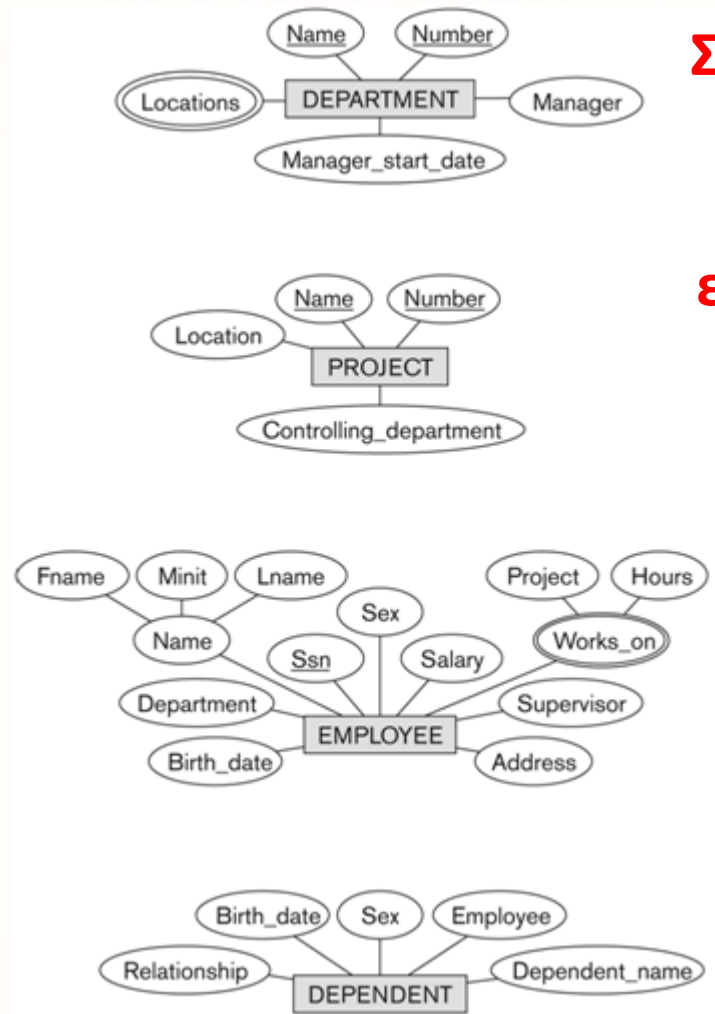
# Σχεδίαση ER

- Προδιαγραφή IV: α) Κάθε employeε μπορεί να έχει ένα αριθμό από Εξαρτώμενα (DEPENDENTS), δηλ., παιδιά.
  - Για κάθε dependent, κρατούμε το β) name, γ) sex, δ) birthdate, και ε) τον τύπο της σχέσης (relationship) με τον employeε.





# Σχεδίαση ER



Στην ερχόμενη διάλεξη  
θα εκλεπτύνουμε ο  
διάγραμμα μας  
εισάγοντας την έννοια  
συσχετίσεων μεταξύ  
των οντοτήτων