

Βάσεις Δεδομένων - 3

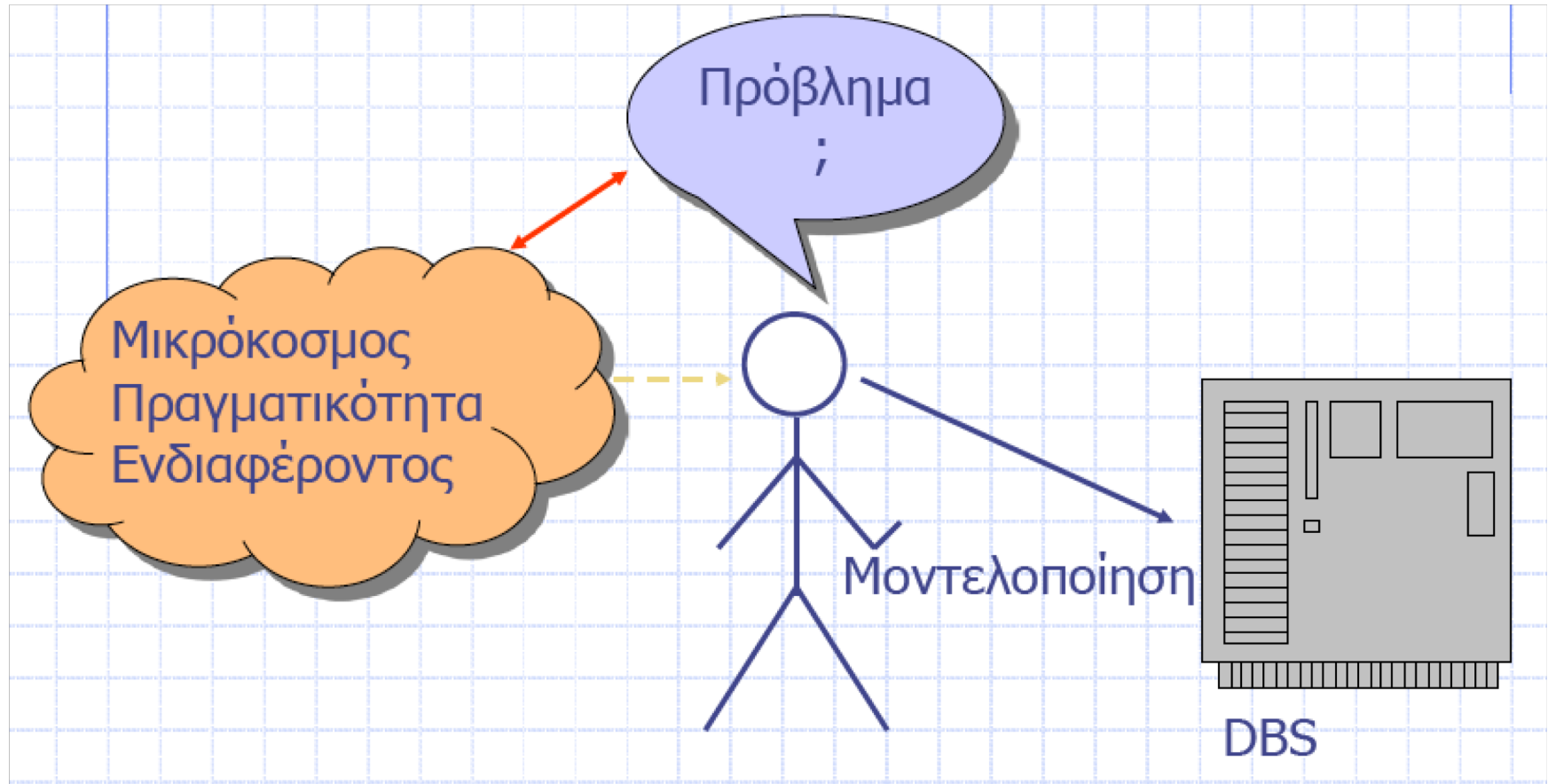
Μοντελοποίηση Δεδομένων χρησιμοποιώντας το
Μοντέλο Οντοτήτων-Συσχετίσεων
(Entity-Relationship ER)

Δρ. Χαράλαμπος Καρανίκας

Περιεχόμενα

- Διαδικασία σχεδίασης της βάσης δεδομένων
- Ένα παράδειγμα εφαρμογής βάσης δεδομένων (COMPANY)
- Βασικές έννοιες ER μοντέλου
 - Οντότητες (Entities) and Ιδιότητες (Attributes)
 - Τύποι οντοτήτων (Entity Types) και Κλειδιά (Key Attributes)
 - Συσχετίσεις (Relationships) και Τύποι Συσχετίσεων
 - Ασθενής Οντότητες
 - Ρόλοι και Ιδιότητες στους Τύπους Συσχετίσεων
- Διαγράμματα ER – Χρήση γραφικών συμβόλων
- ER διάγραμμα για το Σχήμα COMPANY
- Εναλλακτική χρήση γραφικών συμβόλων – UML class diagrams, κλπ

Σχεδιασμός Συστημάτων Βάσεων Δεδομένων - Μοντελοποίηση Δεδομένων



Διαδικασία Ανάπτυξης Βάσεων

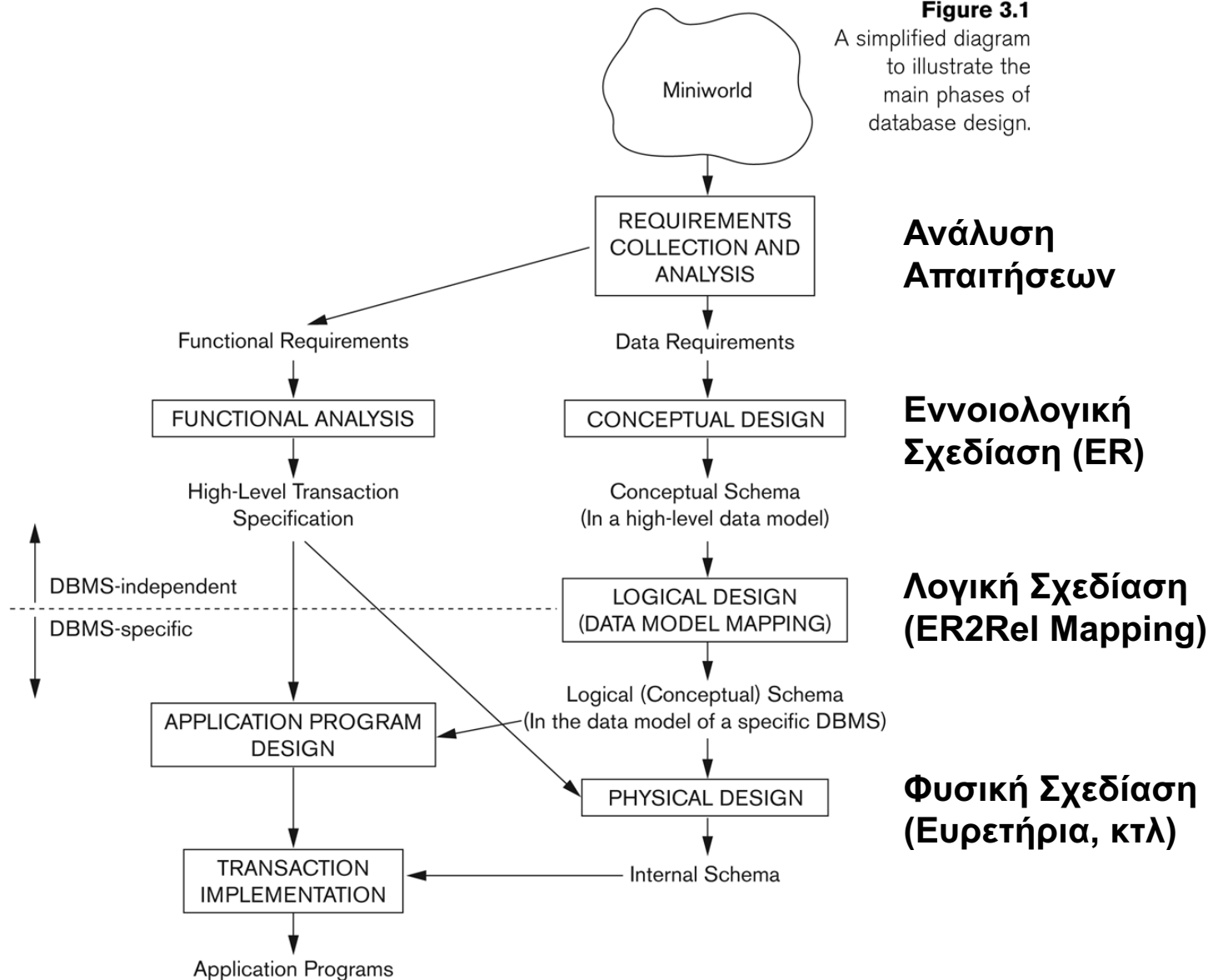
Για την ανάπτυξη μιας Σχεσιακής Βάσης δεδομένων ακολουθούνται τα ακόλουθα βήματα:

1. **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**;
2. **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 είναι **ορθό, πλήρες και αποδοτικό** για να είναι εύκολη η μετατροπή στο επόμενο στάδιο.

Διαδικασία Ανάπτυξης Βάσεων

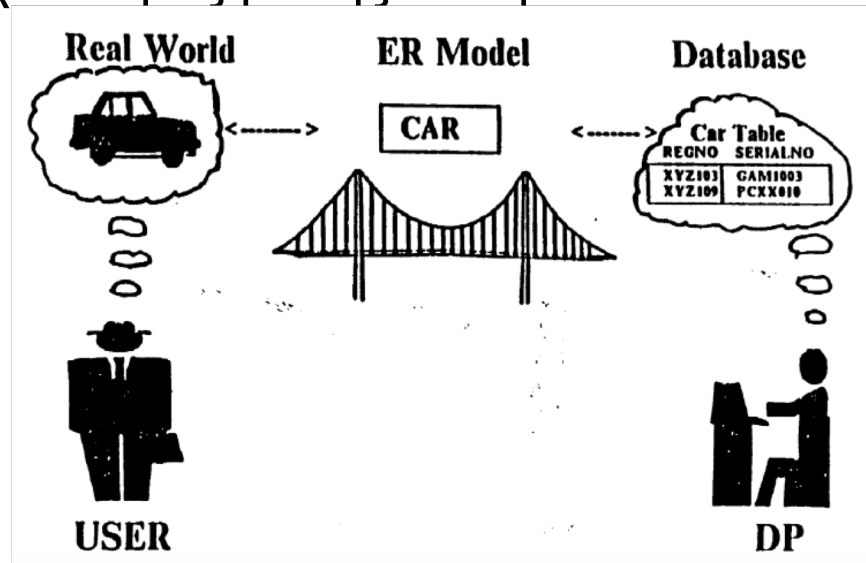
3. **Λογική Σχεδίαση Βάσης (Logical Database Design):** Μετατροπή του ER διαγράμματος σε ένα (Σχεσιακό) Σχήμα Βάσης. Μπορεί να γίνει και αυτόματα με κάποιο αλγόριθμο / εργαλείο
4. **Βελτίωση Σχήματος (Schema Refinement):** Ανάλυση σχήματος για εντοπισμό ενδεχόμενων προβλημάτων π.χ., Αποφυγή πολλών κενών τιμών κλπ
5. **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).
6. **Υλοποίηση Εφαρμογής και Μηχανισμών Ασφάλειας (Application & Security Design)**

Διαγραμματική Αναπαράσταση Διαδικασίας Ανάπτυξης Βάσεων



Σχεδιασμός Βάσης Δεδομένων

- Ο σχεδιασμός Βάσης Δεδομένων είναι η δραστηριότητα καθορισμού του σχήματος μιας βάσης δεδομένων σε ένα συγκεκριμένο μοντέλο δεδομένων
- Τρεις κατηγορίες:
 - **Εννοιολογικός σχεδιασμός βάσης δεδομένων**
 - Λογικός σχεδιασμός βάσης δεδομένων
 - Φυσικός σχεδιασμός βάσης δεδομένων



Entity-Relationship Model

- 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”
- Ένα εργαλείο το οποίο επιτρέπει σε ένα σχεδιαστή (designer) να εκφράσει τις λογικές ιδιότητες μιας βάσης δεδομένων σε ένα καθολικό σχήμα.
- Μια διαγραμματική τεχνική όπου τα αντικείμενα και οι κανόνες μιας εφαρμογής παρουσιάζονται σαν **Οντότητες, Σχέσεις, και Ιδιότητες**.

Entity-Relationship Model

- 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) των διαφανειών.
- Παραλλαγές τέτοιων διαγραμμάτων χρησιμοποιούνται σε διάφορα εμπορικά εργαλεία για την ανάπτυξη βάσεων δεδομένων.

Παράδειγμα COMPANY Database

3.2 An Example

In this section we describe an example database, called COMPANY, that serves to illustrate the ER model concepts and their use in schema design. We list the data requirements for the database here, and then we create its conceptual schema step-by-step as we introduce the modeling concepts of the ER model. The COMPANY database keeps track of a company's employees, departments, and projects. Suppose that, after the requirements collection and analysis phase, the database designers stated the following description of the "miniworld"—the part of the company to be represented in the database:

1. The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
2. A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
3. We store each employee's name, social security number, address, salary, sex, and birthdate. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
4. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's name, sex, birthdate, and relationship to the employee.

Figure 3.2 shows how the schema for this database application can be displayed by means of the graphical notation known as ER diagrams. We describe the process of deriving this schema from the stated requirements—and explain the ER diagrammatic notation—as we introduce the ER model concepts in the following section.

Παράδειγμα COMPANY Database - Απαιτήσεις

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).

Παράδειγμα COMPANY Database - Απαιτήσεις

- ❑ 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 - Αποτέλεσμα: Διάγραμμα ER

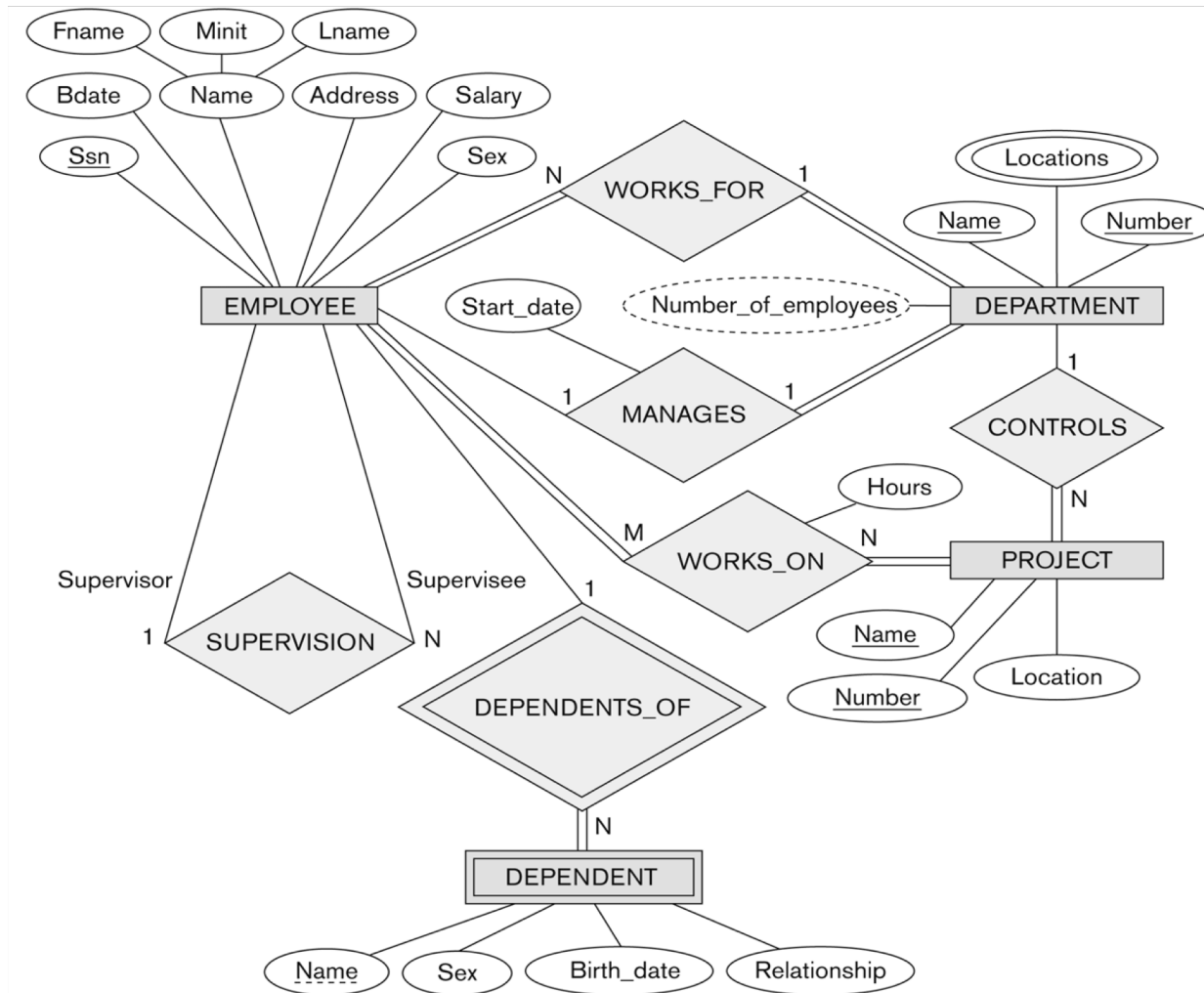


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Έννοιες ER Model

- **Οντότητες (Entities) και Ιδιότητες ή χαρακτηριστικά (Attributes)**
 - Οντότητες (Entities) είναι συγκεκριμένα αντικείμενα ή πράγματα στο mini-world που αναπαρίστανται στη βάση δεδομένων.
 - Για παράδειγμα the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT
 - Ιδιότητες (Attributes) χρησιμοποιούνται για να περιγράψουν μια οντότητα.
 - Για παράδειγμα an EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate
 - Μια συγκεκριμένη οντότητα θα έχει μια τιμή για κάθε ένα από τα χαρακτηριστικά του (ιδιότητες).
 - Για παράδειγμα a specific employee entity may have Name='John Smith', SSN='123456789', Address ='731, Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'
 - Κάθε χαρακτηριστικό έχει μια καθορισμένη τιμή (*value set* ή τύπο δεδομένων data type) που συνδέονται με αυτό π.χ., integer, string, subrange, enumerated type, ...

Τύποι Χαρακτηριστικών (Types of Attributes)

- Απλά ή Μονότιμα (Simple)
 - Κάθε entity έχει μια απλή τιμή για την συγκεκριμένη attribute. Για παράδειγμα SSN="123456" or Sex="M".
- Σύνθετα (Composite)
 - Το χαρακτηριστικό μπορεί να αποτελείται από διάφορα στοιχεία. Για παράδειγμα:
 - Address(Apt#, House#, Street, City, State, ZipCode, Country),
ή
 - Name(FirstName, MiddleName, LastName).
 - Η σύνθεση μπορεί να δημιουργήσει μια ιεραρχία όπου κάποια χαρακτηριστικά είναι και αυτά σύνθετα (αυτό θα πρέπει να αποφεύγεται)
- Με πολλαπλή τιμή (Multi-valued)
 - Μια οντότητα μπορεί να έχει πολλές τιμές για το χαρακτηριστικό. Για παράδειγμα, Color of a CAR {Red, white} or Degrees of a Person {BSc, MSc, Phd}

Παράδειγμα ενός σύνθετου χαρακτηριστικού (composite attribute)

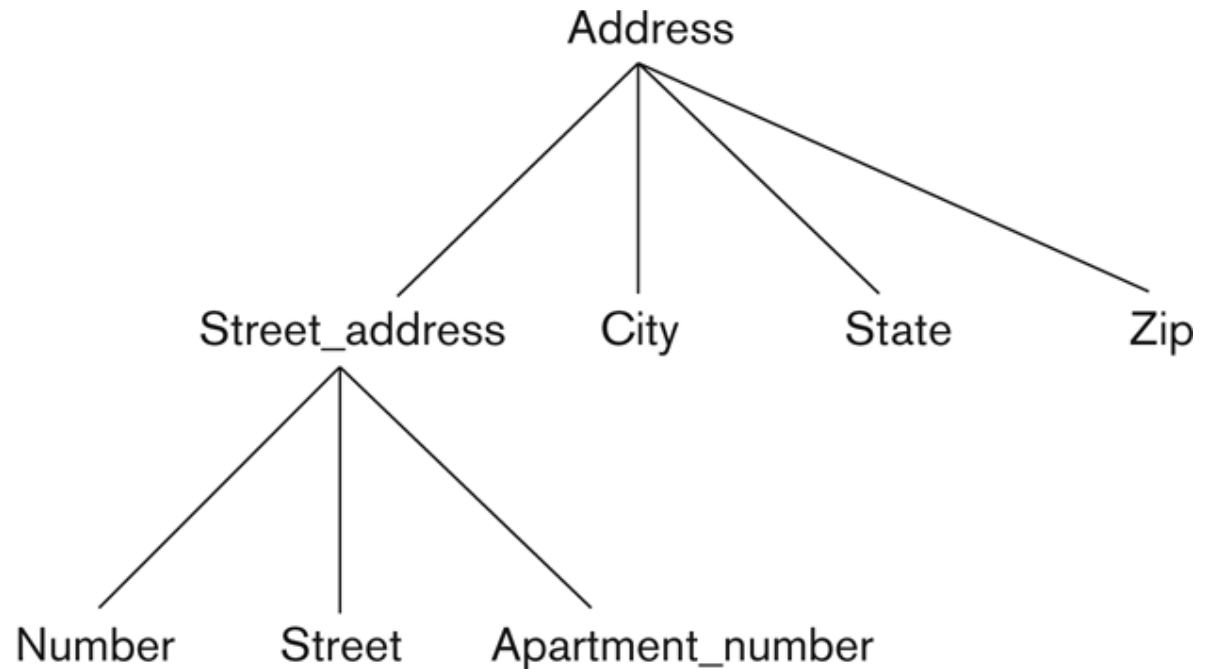


Figure 3.4

A hierarchy of composite attributes.

Τύποι Οντοτήτων (Entity Types) και Χαρακτηριστικά Κλειδί (Key Attributes) (1)

- Οι οντότητες (entities) με τα ίδια βασικά χαρακτηριστικά ομαδοποιούνται σε ένα τύπο οντότητας (entity type) δηλ. σχήμα ή δομή οντότητας.
 - Για παράδειγμα, the entity type EMPLOYEE and PROJECT.

Κάθε οντότητα **ΠΡΕΠΕΙ** να έχει ένα χαρακτηριστικό το οποίο θα μας επιτρέπει να το αναγνωρίζουμε **ΜΟΝΑΔΙΚΑ**.

- Ένα χαρακτηριστικό σε ένα τύπο οντότητας για το οποίο κάθε οντότητα πρέπει να έχει μια μοναδική τιμή ονομάζεται χαρακτηριστικό κλειδί αυτού του τύπου οντότητας.
 - Για παράδειγμα, SSN of EMPLOYEE.

Τύποι Οντοτήτων (Entity Types) και Χαρακτηριστικά Κλειδί (Key Attributes) (1)

- Ένα κλειδί χαρακτηριστικό μπορεί να είναι σύνθετο
 - VehicleTagNumber is a key of the CAR entity type with components (Number, State).
- Ένας τύπος οντότητας μπορεί να έχει περισσότερα από ένα κλειδιά.
 - The CAR entity type may have two keys:
 - VehicleIdentificationNumber (popularly called VIN) Αριθμός Πλαισίου αυτοκινήτου.
 - VehicleTagNumber (Number, State), Αριθμός Πινάκιδας του αυτοκινήτου.
- Κάθε κλειδί είναι υπογραμμισμένο.

Κλειδιά (Keys)

- **Υπερκλειδί (Super Key):** Οποιοσδήποτε συνδυασμός χαρακτηριστικών που μοναδικά καθορίζει μια οντότητα π.χ., Name and SSN, Name and StudentID,
- **Υποψήφιο Κλειδί (Candidate Key)** είναι ένα ελάχιστο υπερκλειδί π.χ., SSN and StudentID
- **Πρωτεύων Κλειδί (Primary Key)** είναι ένα από τα υποψήφια κλειδιά (SSN)
- **Εναλλακτικά κλειδιά** είναι τα εναπομείναντα υποψήφια κλειδιά. Το πρωτεύων κλειδί είναι υπογραμμισμένο, τα εναλλακτικά κλειδιά είναι over-lined

Εμφάνιση ενός τύπου οντότητας στο διάγραμμα

- Στα ER διαγράμματα, ένας τύπος οντότητας (entity type) εμφανίζεται με *ορθογώνιο σχήμα*.
- Τα χαρακτηριστικά ή ιδιότητες (attributes) εμφανίζονται σε *οβάλ σχήμα*
 - Κάθε χαρακτηριστικό συνδέεται με τον τύπο οντότητας του.
 - Τα συστατικά (χαρακτηριστικά) του σύνθετου χαρακτηριστικού συνδέονται με το οβάλ που αντιπροσωπεύει το σύνθετο χαρακτηριστικό
 - Κάθε χαρακτηριστικό κλειδί είναι υπογραμμισμένο
 - Χαρακτηριστικά πολλαπλών τιμών (Multivalued attributes) εμφανίζονται με διπλό οβάλ

Τύπος οντότητας *CAR* με δύο κλειδιά και ένα αντίστοιχο σύνολο οντοτήτων (*entity set*)

(a)

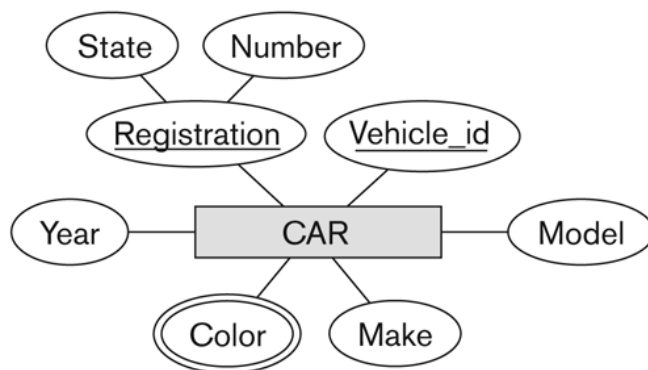


Figure 3.7

The *CAR* entity type with two key attributes, *Registration* and *Vehicle_id*. (a) ER diagram notation. (b) Entity set with three entities.

(b)

CAR
Registration (Number, State), *Vehicle_id*, Make, Model, Year, {Color}

*CAR*₁
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

*CAR*₂
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

*CAR*₃
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Σύνολο Οντοτήτων (Entity Set)

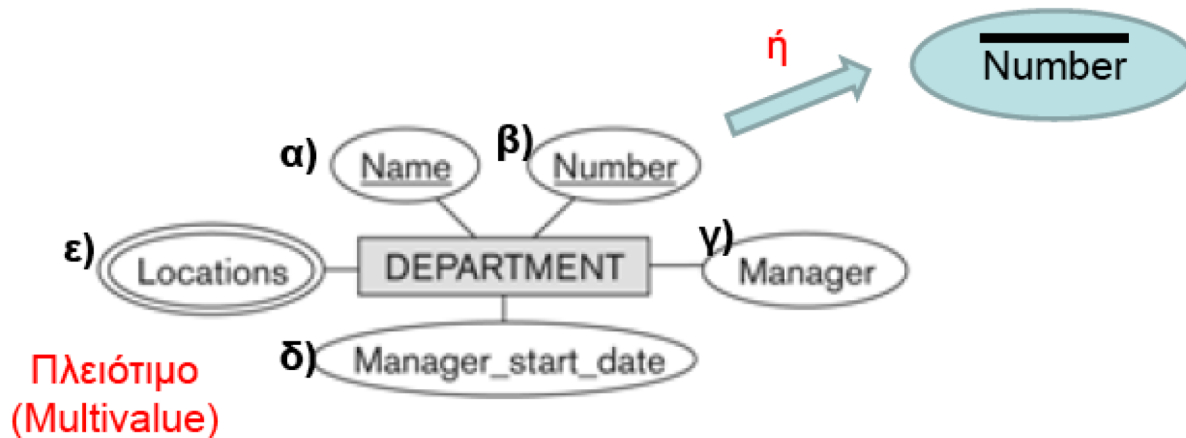
- Κάθε τύπος οντότητας (entity type) θα έχει μια συλλογή από οντότητες που είναι αποθηκευμένα στη βάση δεδομένων και θα ονομάζεται Σύνολο οντοτήτων (entity set)
- Το Σύνολο Οντοτήτων είναι η τρέχουσα κατάσταση (current state) των οντοτήτων αυτού του τύπου που είναι αποθηκευμένα στη βάση δεδομένων.

Αρχικός σχεδιασμός Τύπων Οντοτήτων (Entity Types) για την COMPANY Database Schema

- Με βάση τις απαιτήσεις, μπορούμε να προσδιορίσουμε τέσσερις αρχικούς τύπους οντοτήτων στη βάση δεδομένων COMPANY:
 - ✓ DEPARTMENT
 - ✓ PROJECT
 - ✓ EMPLOYEE
 - ✓ DEPENDENT

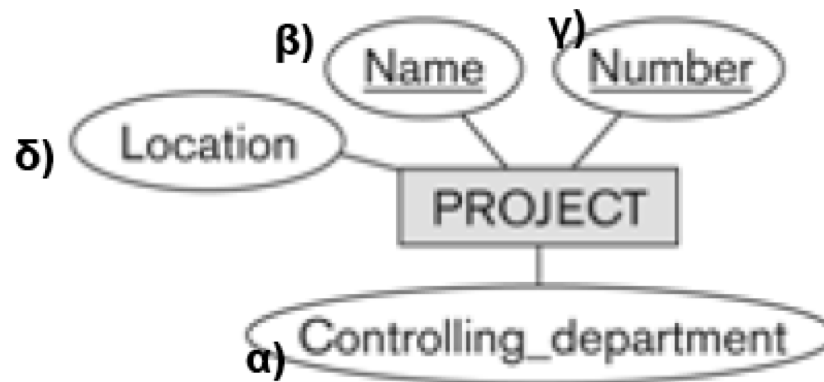
Τύπος Οντότητας: DEPARTMENT

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**.



Τύπος Οντότητας: PROJECT

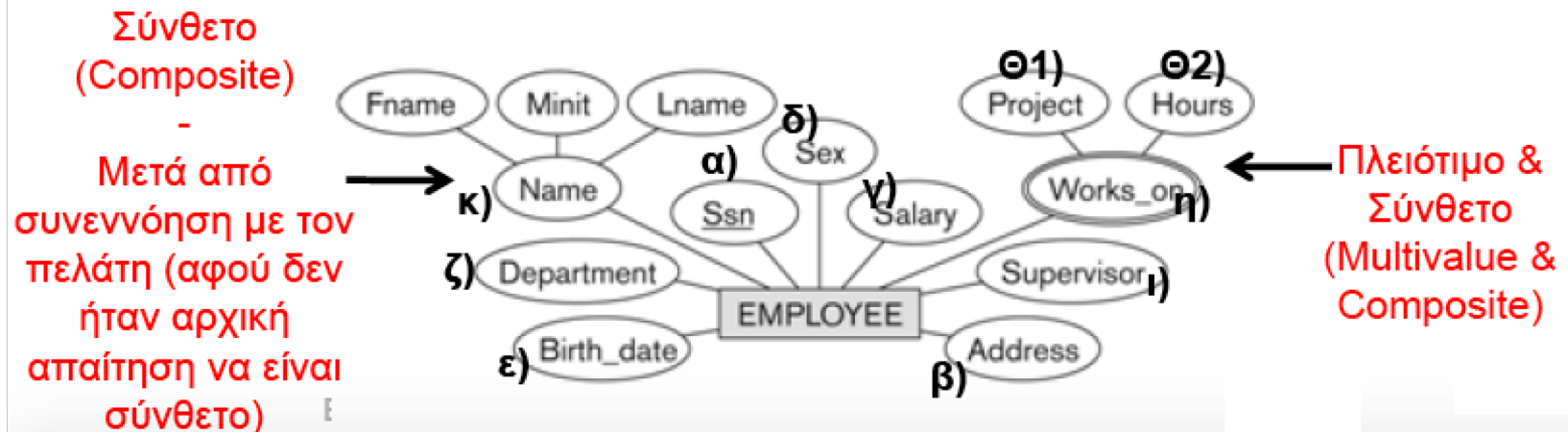
Προδιαγραφή II: **α)** Κάθε department ελέγχει ένα αριθμό από PROJECTs. **β)** Κάθε project έχει ένα μοναδικό **name**, **γ)** μοναδικό **number** και βρίσκεται σε μια **δ)** μόνο τοποθεσία (location).



Τύπος Οντότητας: EMPLOYEE

Προδιαγραφή III: Για κάθε EMPLOYEE αποθηκεύουμε το **α)** social security number (SSN – ταυτότητα), **β)** address, **γ)** salary, **δ)** sex, και **ε)** birthdate.

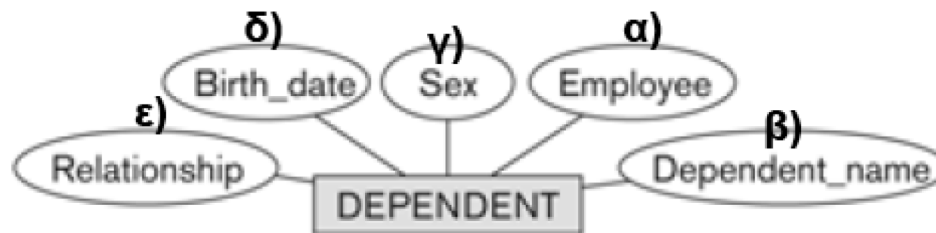
- Κάθε employee **ζ)** δουλεύει για ένα department αλλά μπορεί να **η)** εργάζεται πάνω σε πολλά projects.
- **θ)** Για κάθε project καταγράφουμε τον χρόνο (num of hours per week) τα οποία αφιερώνει ένας employee πάνω σε κάθε project.
- **ι)** Επίσης καταγράφουμε τον προϊστάμενο (*direct supervisor*) κάθε employee αλλά και το όνομα του Employee



Τύπος Οντότητας: DEPENDENT

Προδιαγραφή IV: α) Κάθε employee μπορεί να έχει ένα αριθμό από **Εξαρτώμενα (DEPENDENTS)**, δηλ., παιδιά.

Για κάθε depend, κρατούμε το β) name, γ) sex, δ) birthdate, και ε) τον τύπο της σχέσης (relationship) με τον employee.



Αρχικός σχεδιασμός Τύπων Οντοτήτων: EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

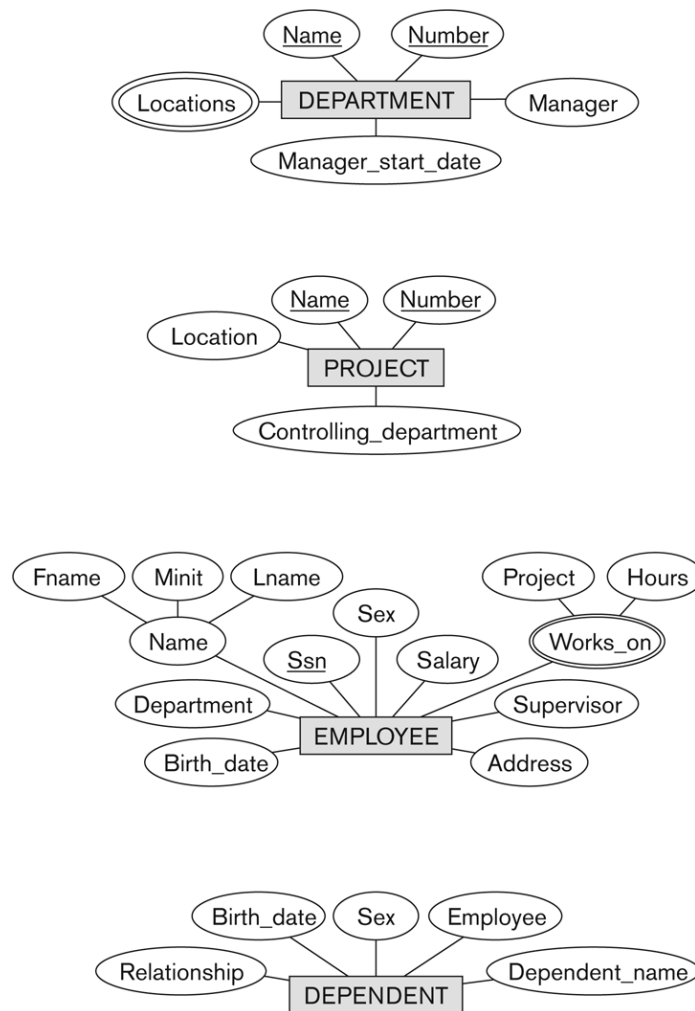
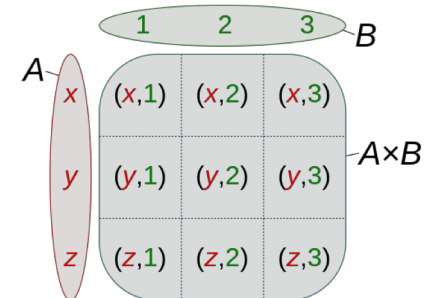


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

Τελειοποίηση του αρχικού σχεδιασμού, εισάγοντας τις συσχετίσεις (relationships)

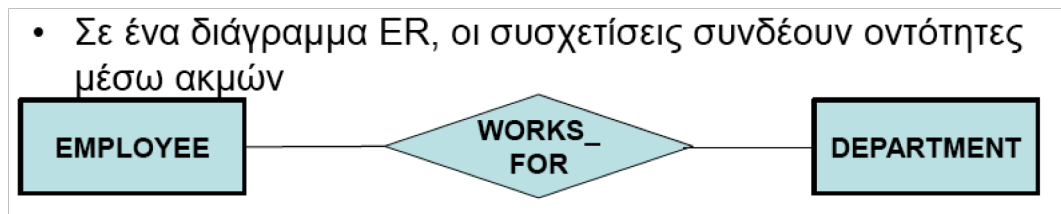
- Ο αρχικός σχεδιασμός τυπικά δεν είναι πλήρης
- Μέχρι τώρα είδαμε πως μπορούμε να αναπαραστήσουμε τις **Οντότητες** και τα **Χαρακτηριστικά τους** σε ένα διάγραμμα ER.
- Τώρα θα δούμε πως μπορούμε να συσχετίσουμε αυτές τις οντότητες μέσω **Συσχετίσεων (Relationships)**,
 - π.χ., Οντότητες: EMPLOYEE, DEPARTMENT
 - Συσχετίσεις: WORKS_FOR
- Μια **Συσχέτιση** συνδέει δυο ή περισσότερες οντότητες με συγκεκριμένο νόημα.
 - Π.χ., EMPLOYEE Γιάννης Μιχαήλ works_for CS DEPARTMENT.
- Τυπικά, μια συσχέτιση μεταξύ δυο συνόλων A , B είναι ένα υποσύνολο του καρτεσιανού γινομένου $A \times B$
 - Καρτεσιανό Γινόμενο: $A \times B = \{ (a,b) \mid a \in A \text{ and } b \in B \}$
 - $R \subseteq A \times B$



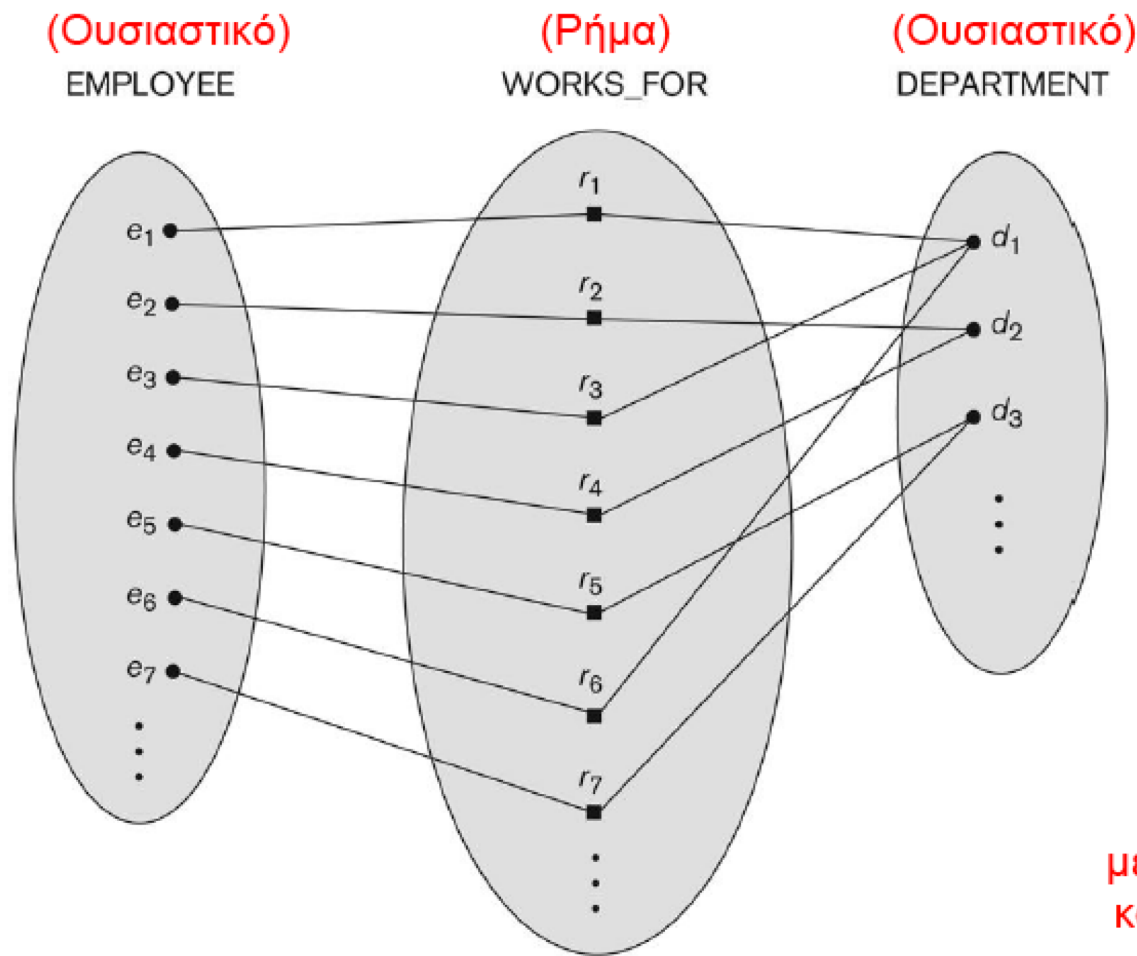
Συσχετίσεις και Τύπος Συσχέτισης

Relationships and Relationship Types

- Συσχετίσεις του ίδιου τύπου ομαδοποιούνται σε **Τύπο Συσχέτισης**
 - Το σχήμα (δομή) μιας συσχέτισης.
 - Για παράδειγμα, the WORKS_ON relationship type in which EMPLOYEES and PROJECTs participate, or the MANAGES relationship type in which EMPLOYEES and DEPARTMENTs participate.
- **Σύνολο Συσχετίσεων (Relat. Set ή Rela. State)** Ένα σύνολο συσχέτισεων με τα ίδια χαρακτηριστικά
 - π.χ., {works_for1, works_for2, ...} κτλ.
- **Ο βαθμός ενός τύπου συσχέτισης (Degree of Relationship Type)** είναι το πλήθος των οντοτήτων που συμμετέχουν.
 - Both MANAGES and WORKS_ON are *binary* relationships.



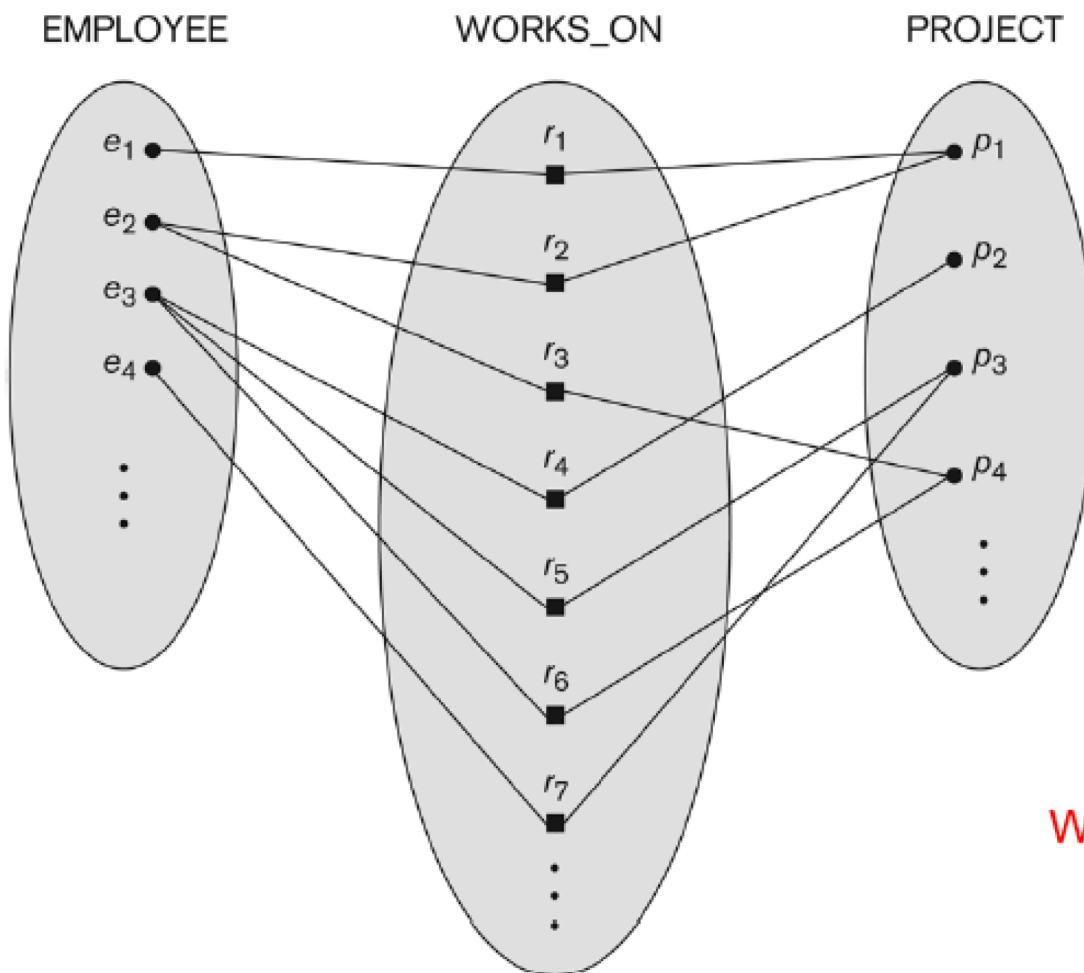
Συσχετίσεις N:1 (Διαγραμματική Απεικόνιση με Σύνολα)



Όπως θα δούμε
αργότερα πρόκειται
για μια **N:1 σχέση**
(δηλ., σε ένα Dep.
δουλεύουν πολλοί
Emp. αλλά κάθε
Emp. σε ένα Dep.

Στιγμιότυπα του
τύπου σχέσης
WORKS_FOR
μεταξύ EMPLOYEE
και DEPARTMENT

Συσχετίσεις M:N (Διαγραμματική Απεικόνιση με Σύνολα)



Όπως θα δούμε αργότερα πρόκειται για μια **M:N σχέση** (δηλ., ένας Emp. δουλεύει σε πολλά Proj. και σε ένα Proj. πολλοί Emp.)

Στιγμιότυπα του τύπου σχέσης WORKS_ON μεταξύ EMPLOYEE και PROJECT

Σχήμα COMPANY με Συσχετίσεις

Μελετώντας τις **απαιτήσεις (requirements)** της εφαρμογής αυτής, εντοπίζουμε τις ακόλουθες **έξι δυαδικές συσχετίσεις**:

- **WORKS_FOR** (μεταξύ **EMPLOYEE, DEPARTMENT**)
 - ♦ «... Κάθε employee δουλεύει για ένα department ...»
- **MANAGES** (και αυτή μεταξύ **EMPLOYEE, DEPARTMENT**)
 - ♦ «... Κάθε department έχειένα manager που διαχειρίζεται το department ...»
- **CONTROLS** (μεταξύ **DEPARTMENT, PROJECT**)
 - ♦ «... Κάθε department ελέγχει ένα αριθμό από PROJECTs...»
- **WORKS_ON** (μεταξύ **EMPLOYEE, PROJECT**)
 - ♦ «Κάθε employee ... εργάζεται πάνω σε πολλά projects.»
- **SUPERVISION** (μεταξύ **EMPLOYEE** (ως υφιστάμενος) και **EMPLOYEE** (ως προϊστάμενος))
 - ♦ «...καταγράφουμε τον προϊστάμενο (direct supervisor) κάθε employee...»
- **DEPENDENTS_OF** (μεταξύ **EMPLOYEE, DEPENDENT**)
 - ♦ «...Κάθε employee μπορεί να έχει ένα αριθμό από (DEPENDENTS)...»

Μια Μεθοδολογία Εύρεσης Σχέσης

- Παρατήρηση
 - ουσιαστικά → τύποι/σύνολα οντοτήτων
 - ρήματα → τύποι συσχετίσεων

1. Κατασκεύασε ένα πίνακα Οντότητας/Οντότητας (Τριγωνικό)
2. Έλεγξε την τομή κάθε στήλης και γραμμής για πιθανή σχέση
3. Γράψε το ρήμα που περιγράφει τη σχέση

	ΦΟΙΤΗΤΗΣ	ΜΑΘΗΜΑ	ΚΑΘΗΓΗΤΗΣ
ΦΟΙΤΗΤΗΣ	τίποτα	Παρακολουθεί	τίποτα
ΜΑΘΗΜΑ		τίποτα	Διδάσκεται από
ΚΑΘΗΓΗΤΗΣ			τίποτα

ER DIAGRAM - Relationship Types are:

WORKS_FOR, MANAGES, WORKS_ON, CONTROLS, SUPERVISION,
DEPENDENTS_OF

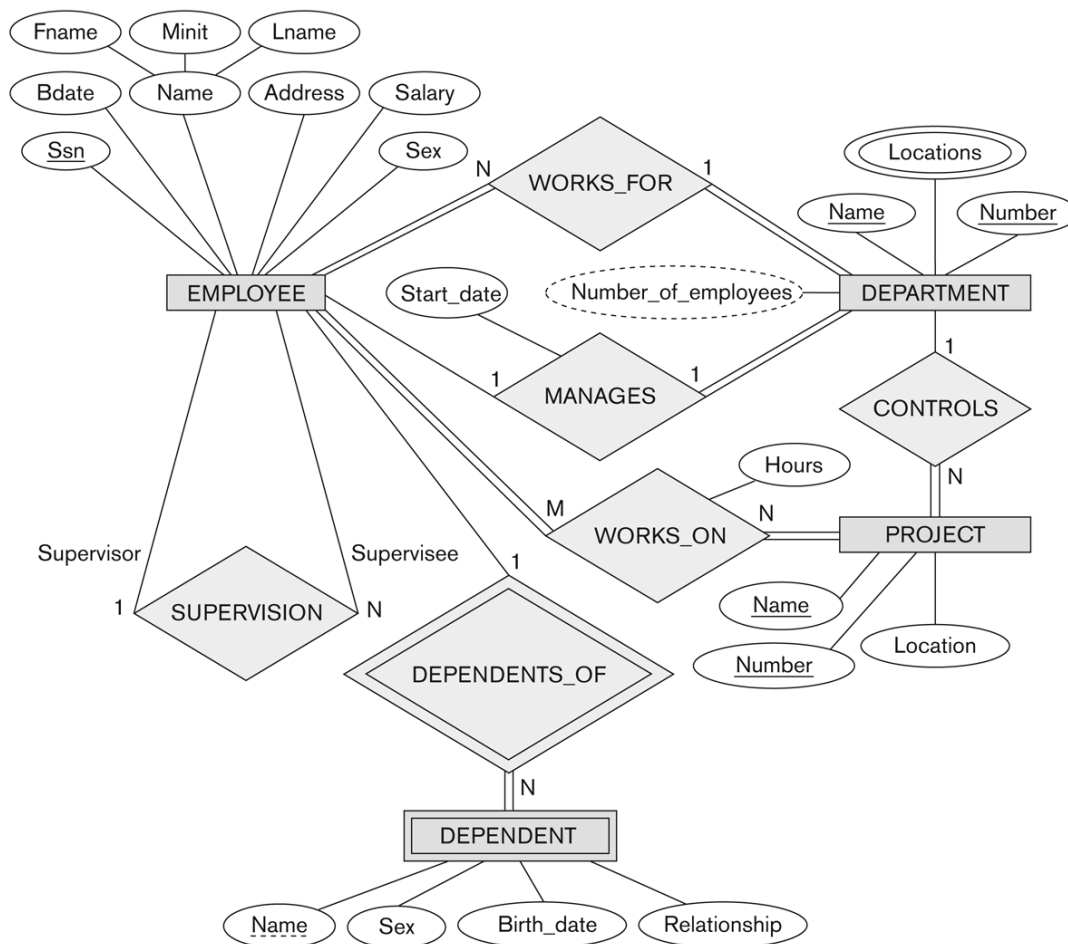


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Θέματα που θα δούμε αργότερα:

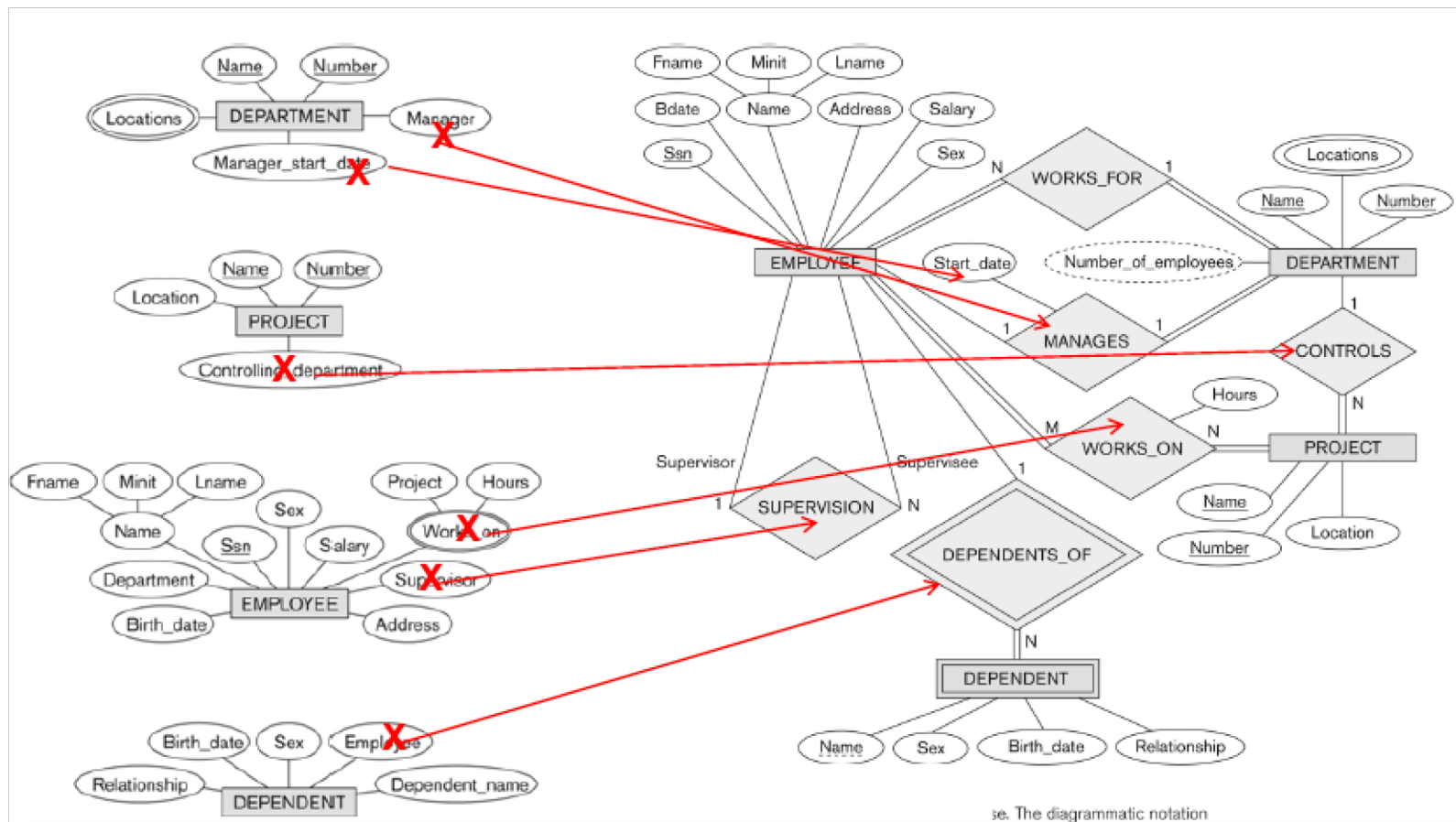
1.Λόγοι Πληθικότητας
(1:1, 1:N, N:1, M:N);

2.Περιορισμοί Συμμετοχής (διπλές και μονές γραμμές)

3.Ασθενείς Οντότητες
(οντότητες με διπλές γραμμές)

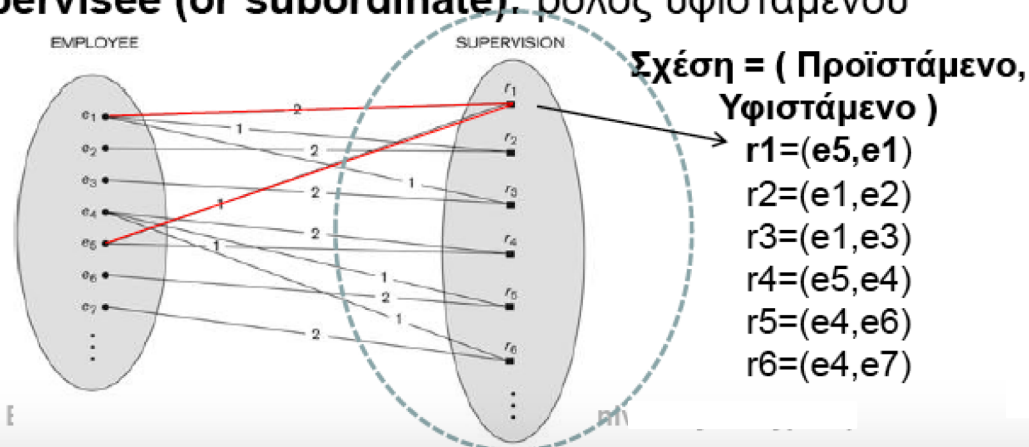
Προσέξτε ότι μερικά **χαρακτηριστικά** των αρχικών **οντοτήτων** έχουν **μετατραπεί** σε **σχέσεις**. Ποια είναι αυτά;

Διάγραμμα ER με Συσχετίσεις (Γνωρίσματα που Αναπαριστώνται ως Σχέσεις)



Αναδρομικός Τύπος Συσχέτισης (Recursive Relationship Type)

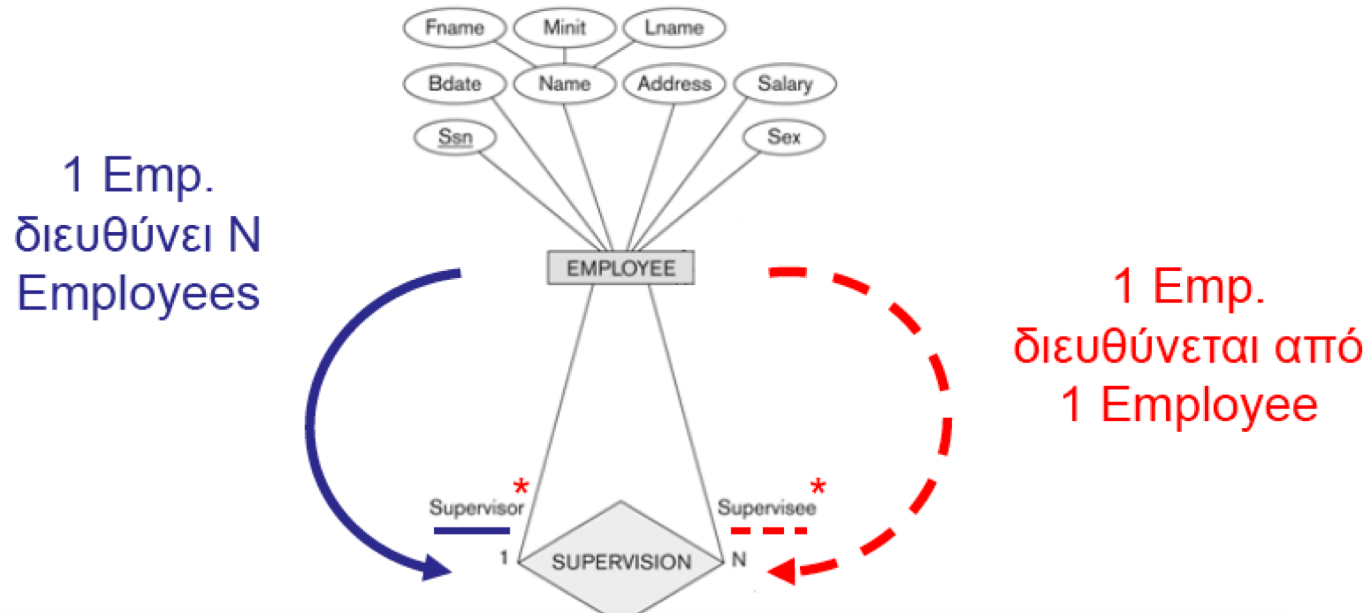
- **Αναδρομικός Τύπος Συσχέτισης:** Ένας τύπος συσχέτισης ο οποίος έχει τον **ίδιο τύπο οντότητας** σε **επί μέρους ρόλους**
- Π.χ., η συσχέτιση **SUPERVISION**, όπου **EMPLOYEE** συμμετέχει διπλά σε επί μέρους ρόλους:
 - 1) **supervisor (or boss)**: ρόλος προϊσταμένου
 - 2) **supervisee (or subordinate)**: ρόλος υφιστάμενου



Αναδρομικός Τύπος Συσχέτισης (Recursive Relationship Type)

- **Διαγραμματική Απεικόνιση Αναδρομικής Σχέσης :**

- **supervisor (or boss):** ρόλος προϊσταμένου
- **supervisee (or subordinate):** ρόλος υφιστάμενου



* Η ρητή αναφορά στους επί μέρους ρόλους είναι απαραίτητη για να είναι ξεκάθαρη η σχέση

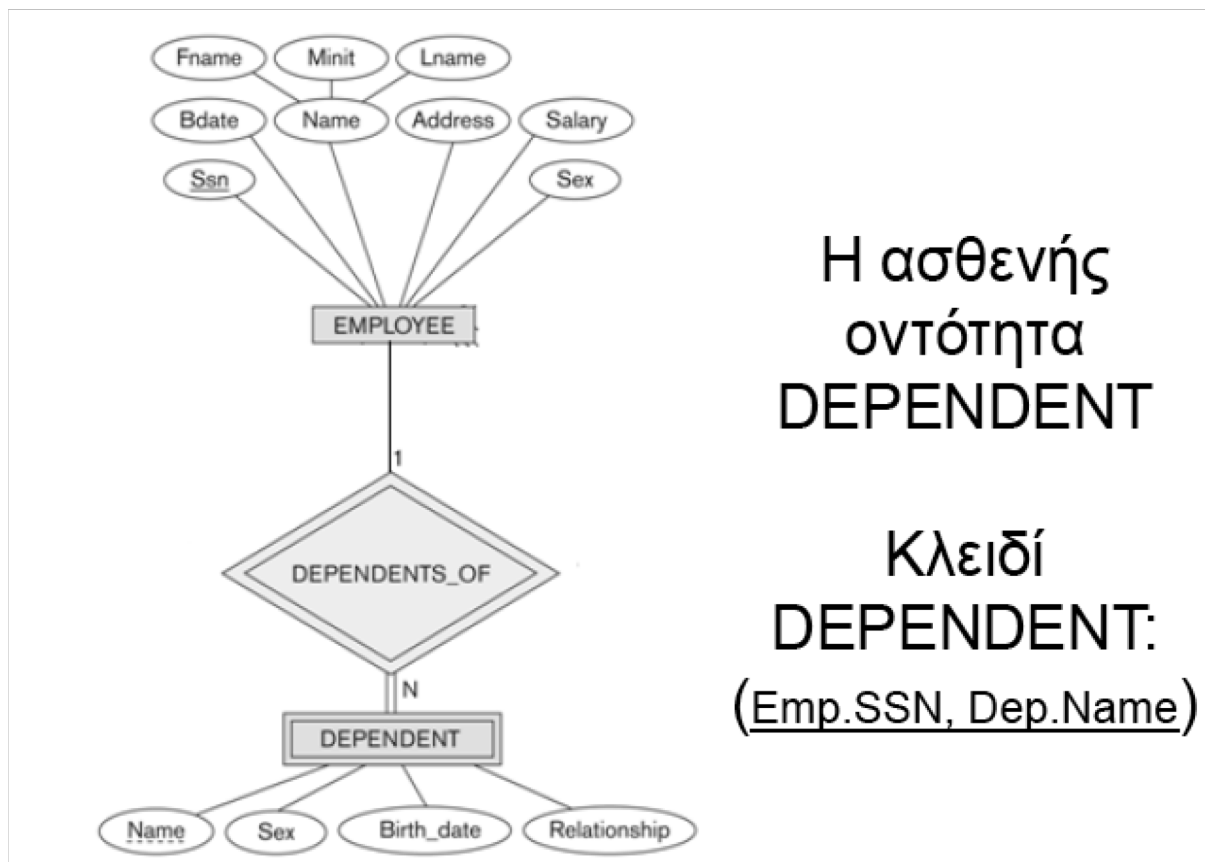
Ασθενείς Τύποι Οντοτήτων (Weak Entity Types)

- **Ασθενής Οντότητα (Weak Entity):** Μια οντότητα η οποία δεν έχει κλειδί.

DEPENDENT

- Αυτές οι οντότητες ορίζονται από κάποιο άλλο τύπο οντοτήτων (τον **προσδιορίζοντα – owner entity type**).
- Συγκεκριμένα, ορίζονται από τον ακόλουθο συνδυασμό:
 - Ένα **μερικό κλειδί (partial key)** της ασθενής οντότητας
 - Το (πρωτεύων) **κλειδί** του **προσδιορίζοντα**
- **Παράδειγμα:**
 - Η οντότητα DEPENDENT ορίζεται από το dependent's **first name**, και το **κλειδί της σχέσης EMPLOYEE** (με τον οποίο/α συσχετίζεται η/ο DEPENDENT)
 - (EMPLOYEE.SSN, DEPENDENT.Name) γίνεται το κλειδί της σχέσης DEPENDENT

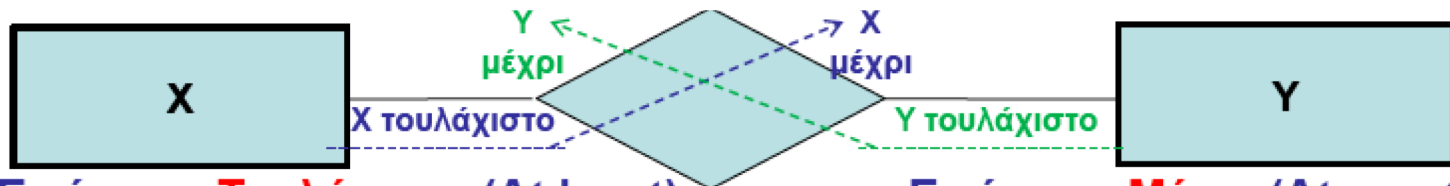
Ασθενείς Τύποι Οντοτήτων (Weak Entity Types)



Περιορισμούς στις συσχετίσεις (Constraints on Relationships)

- **Λόγος Πλήθους (Cardinality Ratio)** (specifies *maximum* participation)
Καθορίζει τον αριθμό στιγμιότυπων της συσχέτισης στα οποία μια οντότητα μπορεί να συμμετέχει.
 - One-to-one (1:1) π.χ., 1 Τμήμα έχει 1 Πρόεδρο
 - One-to-many (1:N) π.χ., 1 Μητέρα έχει N Παιδιά
 - Many-to-one (N:1) π.χ., N Παιδιά έχουν 1 Μητέρα
 - Many-to-many (M:N) π.χ., M Υπάλληλοι δουλεύουν σε N projects
- **Περιορισμοί Συμμετοχής (participation constraint)**(specifies *minimum* participation) ορίζει την **ελάχιστη** συμμετοχή μιας οντότητας σε μια σχέση.
 - zero (optional participation, not existence-dependent)
 - • **Μερική Συμμετοχή (Partial):** ΚΑΠΟΙΑ ή ΚΑΝΕΝΑ στοιχεία μιας οντότητας A συσχετίζονται με την οντότητα B
 - one or more (mandatory participation, existence-dependent)
 - ή — • **Ολική Συμμετοχή (Total – Existence Dependency):** ΚΑΘΕ στοιχείο μιας οντότητας A συσχετίζεται με την οντότητα B

Ανάγνωση ER με Δομικούς Περιορισμούς (Η Ερώτηση Τουλάχιστο / Μέχρι)



Ερώτηση Τουλάχιστο (At least)

- Μονή Γραμμή => Τουλάχιστο 0 (δεν αναφέρεται)
- Διπλή Γραμμή => Τουλάχιστο 1

Ερώτηση Μέχρι (At-most)

- x:1 => Μέχρι 1
- x:N => Μέχρι N.
- * το x υποδηλεί το οτιδήποτε

Σημείωση: **ΠΑΝΤΑ** διαβάζουμε τη σχέση και προς τις δυο κατευθύνσεις



* Δοκιμάστε να διαβάσετε το διάγραμμα

Χαρακτηριστικά σε Τύπους Συσχετίσεων

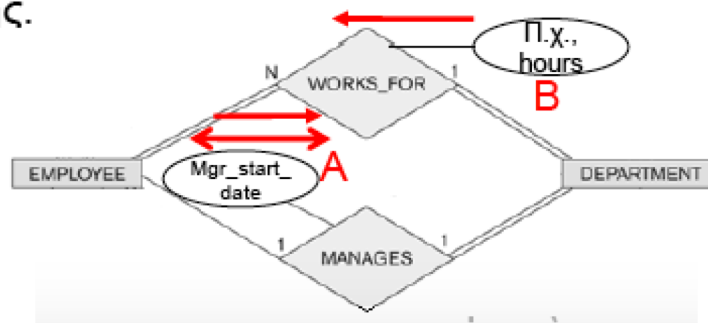
Μια συσχέτιση μπορεί να φέρει διάφορα γνώρισμα:

- Π.χ., Hours στη συσχέτιση WORKS_ON το οποίο περιγράφει πόσες ώρες δουλεύει ένας EMPLOYEE σε ένα PROJECT.



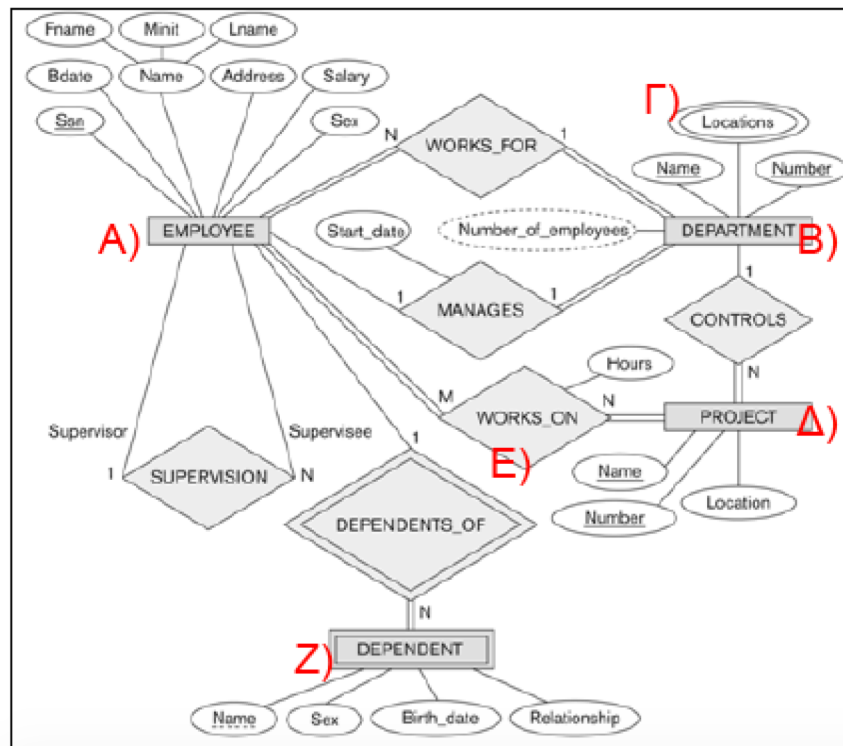
Στις πλείστες περιπτώσεις, τα γνώρισμα εμφανίζονται στις M:N συσχετίσεις. **Γιατί;**

- Διότι στις υπόλοιπες περιπτώσεις μπορεί να **μεταφερθεί** το γνώρισμα σε κάποια από τις **οντότητες** με την **ακόλουθη λογική**
 - **A) 1:1.** Το γνώρισμα μπορεί να μεταφερθεί προς **οποιαδήποτε** από τις δυο **οντότητες** που συμμετέχουν (προτιμότερο προς ολική συμμετοχή)
 - **B) 1:N, N:1.** Το γνώρισμα μπορεί να μεταφερθεί **προς το N** μέρος της συσχέτισης.



Σχεσιακό Σχήμα (όπως θα οριστεί σε ερχόμενες διαλέξεις)

Σημείωση: Δεν χρειάζεται να καταλάβετε ακόμη πως ακριβώς προέκυψε το πιο κατω Σχεσιακό Σχήμα. Απλά παρατηρήστε την σχέση των γνωρισμάτων μεταξύ του ER διαγράμματος (αριστερά) και του Σχεσιακού Σχήματος (δεξιά).



A)	EMPLOYEE	Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-----------	-----------------	-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

B)	DEPARTMENT	Dname	Dnumber	Mgr_ssn	Mgr_start_date
-----------	-------------------	-------	---------	---------	----------------

Γ)	DEPT_LOCATIONS	Dnumber	Dlocation
-----------	-----------------------	---------	-----------

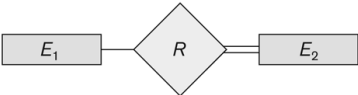
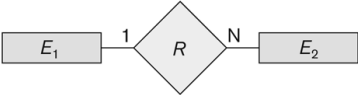
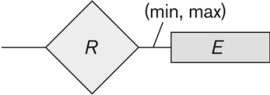
Δ)	PROJECT	Pname	Pnumber	Plocation	Dnum
-----------	----------------	-------	---------	-----------	------

Ε)	WORKS_ON	Essn	Pno	Hours
-----------	-----------------	------	-----	-------

Ζ)	DEPENDENT	Essn	Dependent_name	Sex	Bdate	Relationship
-----------	------------------	------	----------------	-----	-------	--------------

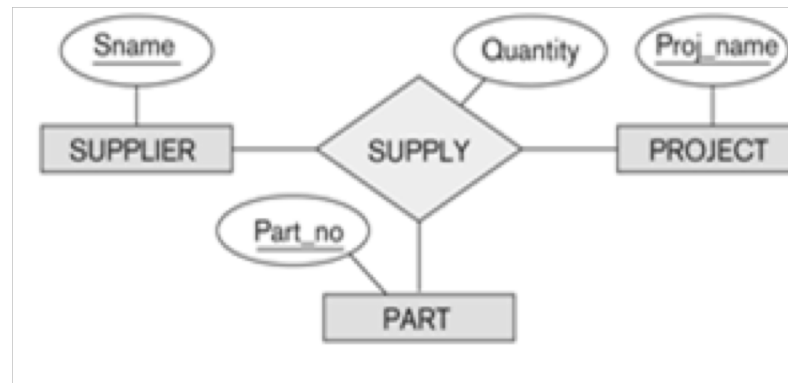
Περίληψη της σημειογραφίας (notation) για το ER διάγραμμα

Figure 3.14
Summary of the
notation for ER
diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for $E_1:E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

Συσχετίσεις ψηλότερου βαθμού

- Τύποι συσχετίσεων βαθμού N
 - 2-οντότητες → **δυναδική (binary)** συσχέτιση
 - 3-οντότητες → **τριαδική (ternary)** συσχέτιση
 - ...
 - N-οντότητες → **N-αδική (n-ary)** συσχέτιση
- Η συσχέτιση **SUPPLY** ορίζει τι **PARTs** προμηθεύει ένας **SUPPLIER** σε ένα **PROJECT** (και σε τι ποσότητες)



Συσχετίσεις ψηλότερου βαθμού

- Σε γενικές γραμμές, μια N -ary σχέση δεν είναι ισοδύναμη με n -δυναδικές σχέσεις.
- Περιορισμοί είναι πιο δύσκολο να καθοριστούν για τις σχέσεις υψηλότερου βαθμού ($n > 2$) από ότι για τις δυναδικές σχέσεις.
- Σε γενικές γραμμές, 3 δυναδικές συσχετίσεις μπορεί να παριστάνουν διαφορετικές πληροφορίες από ότι μια τριμερή συσχέτιση.

Συσχετίσεις Υψηλότερου Βαθμού (Μετατροπή σε Δυαδικές Συσχετίσεις)

Μετατροπή σε Πολλαπλές Δυαδικές Συσχετίσεις

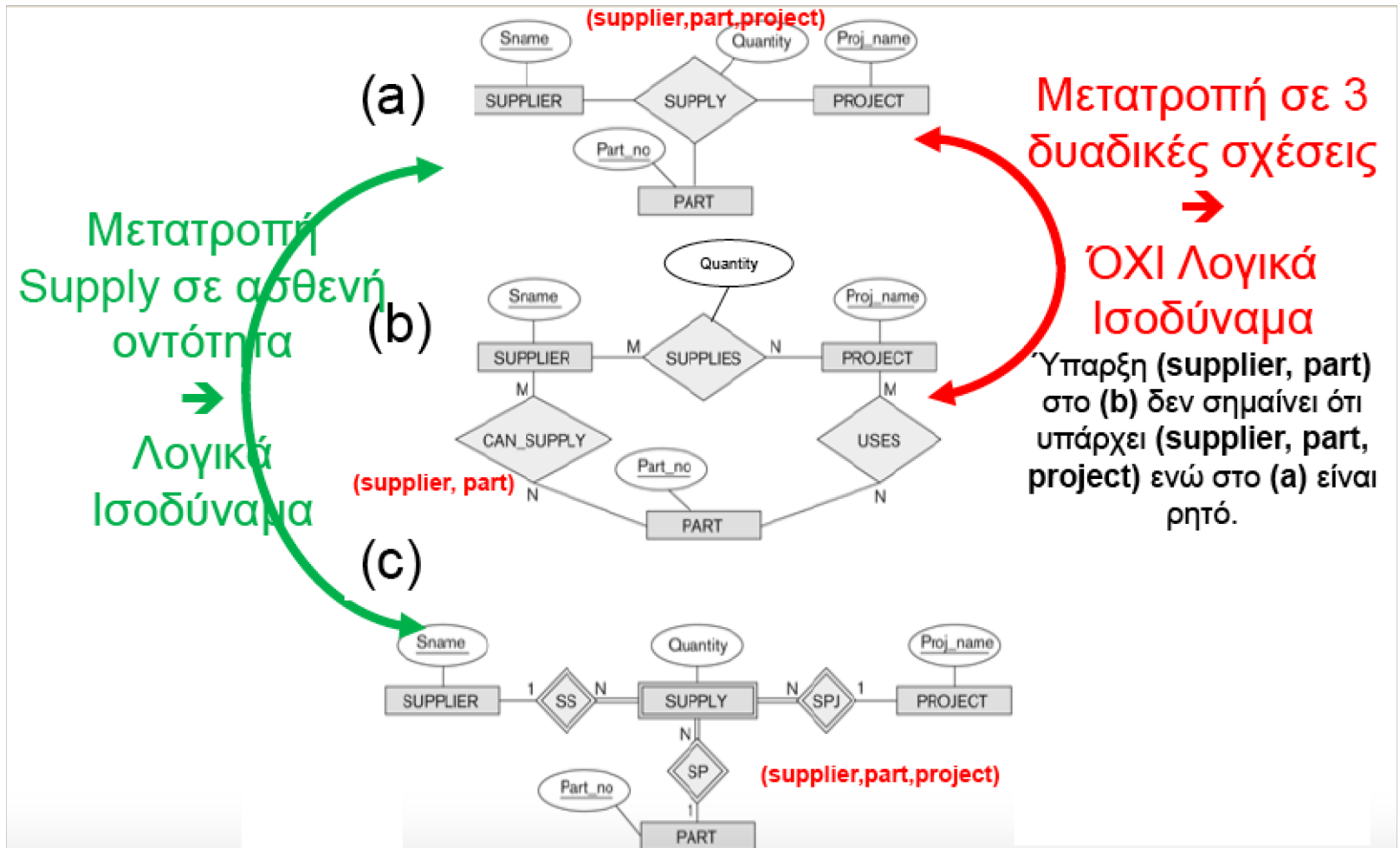
- Πολλά εργαλεία μοντελοποίησης επιτρέπουν **μόνο δυαδικές συσχετίσεις**

A. Για να είναι ευκολότερο να εκφραστούν οι **δομικοί περιορισμοί** (δηλ., λόγοι Πληθυκότητας, περιορισμοί συμμετοχής)

B. Διότι δεν υπάρχει κάποιος **προτυποποιημένος τρόπος** μετάφρασης μιας **n-αδικής** συσχέτισης σε **σχεσιακό σχήμα**

- Κάτι τέτοιο όμως θέλει προσοχή διότι το **αποτέλεσμα** μπορεί να **ΜΗΝ** είναι **λογικά ισοδύναμο**
 - π.χ., δες επόμενο παράδειγμα

Παράδειγμα Τριαδικής Συσχέτισης (n-αδική <> n Δυαδικές)



Εργαλεία Μοντελοποίησης Δεδομένων

Υπάρχει ένας μεγάλος αριθμός εργαλείων που χρησιμοποιούνται για την **εννοιολογική σχεδίαση** και για την **μετατροπή του σχεδίου στο σχεσιακό σχήμα**.

ΘΕΤΙΚΑ:

- **Επιταχύνουν** την διαδικασία ανάπτυξης μέσω γραφικών περιβαλλόντων ανάπτυξης.
- Επιτρέπουν την **αυτόματη ανάπτυξη** του τελικού σχεσιακού σχήματος (**forward engineering**)
- Επιτρέπουν την αυτόματη ανάπτυξη του **εννοιολογικού σχήματος** από το **σχεσιακό σχήμα** (**backward ή reverse engineering**)

ΑΡΝΗΤΙΚΑ:

- Δεν υπάρχει **κοινή σημειογραφία** ανάμεσα στα εργαλεία με αποτέλεσμα να απαιτείται χρόνος εξοικείωσης.
- Τα περισσότερα εργαλεία αξιοποιούν μια **σχεσιακή αναπαράσταση** της πληροφορίας αντί της πιο αφαιρετικής **ER προσέγγισης**.

Κάποια Εργαλεία

COMPANY	TOOL	FUNCTIONALITY
Embarcadero Technologies	ER Studio	Database Modeling in ER and IDEF1X
	DB Artisan	Database administration, space and security management
Oracle	Developer 2000/Designer 2000	Database modeling, application development
Popkin Software	System Architect 2001	Data modeling, object modeling, process modeling, structured analysis/design
Platinum (Computer Associates)	Enterprise Modeling Suite: Erwin, BPWin, Paradigm Plus	Data, process, and business component modeling
Persistence Inc.	Pwertier	Mapping from O-O to relational model
Rational (IBM)	Rational Rose	UML Modeling & application generation in C++/JAVA
Resolution Ltd.	Xcase	Conceptual modeling up to code maintenance
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio	Visio Enterprise	Data modeling, design/reengineering Visual Basic/C++

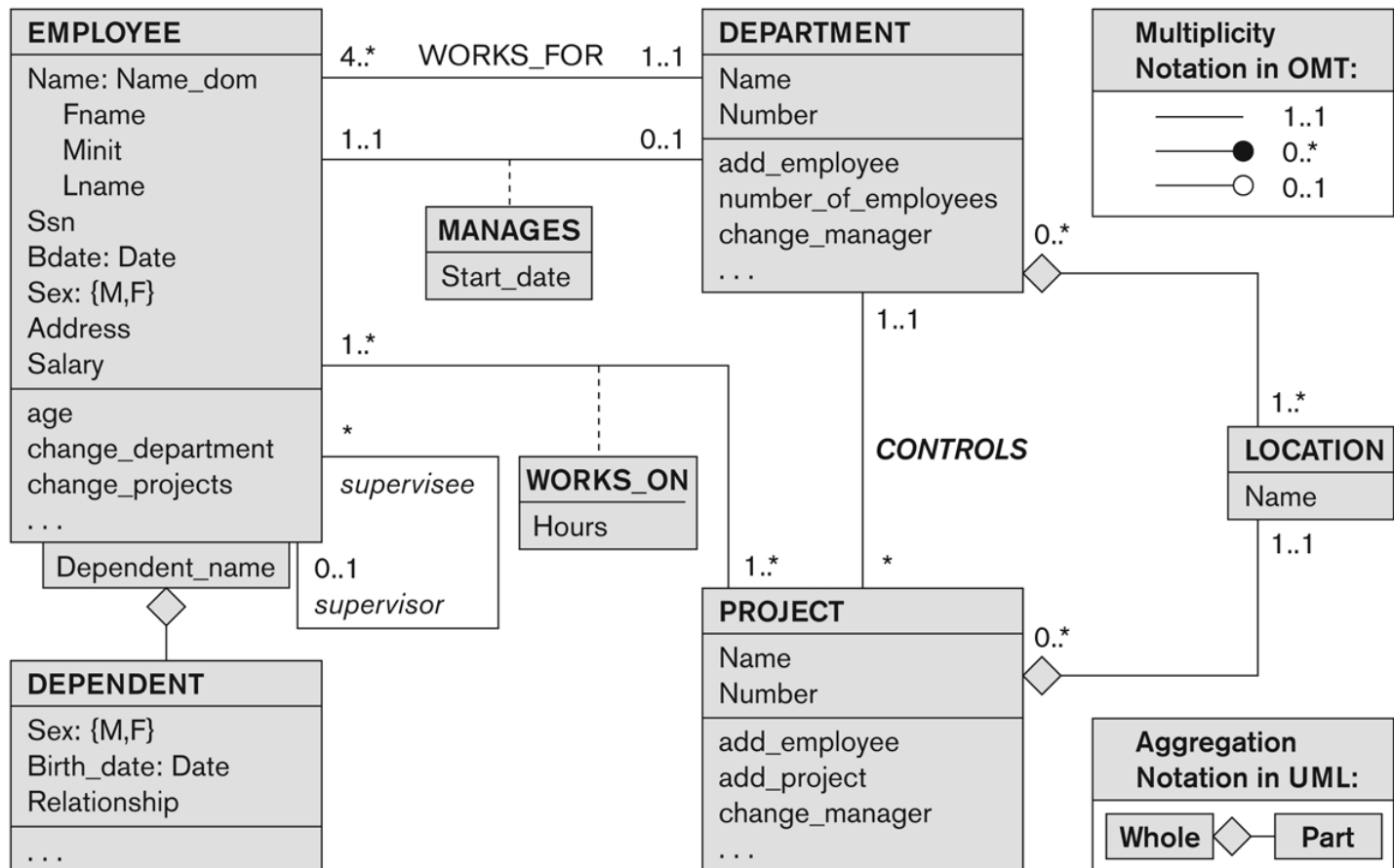
UML class diagrams

- Represent classes (similar to entity types) as large rounded boxes with three sections:
 - Top section includes entity type (class) name
 - Second section includes attributes
 - Third section includes class operations (operations are not in basic ER model)
- Relationships (called associations) represented as lines connecting the classes
 - Other UML terminology also differs from ER terminology
- Used in database design and object-oriented software design
- UML has many other types of diagrams for software design

UML class diagram for COMPANY database schema

Figure 3.16

The COMPANY conceptual schema in UML class diagram notation.



Περίληψη

- Βασικές έννοιες ER μοντέλου
 - Οντότητες (Entities) and Ιδιότητες (Attributes)
 - Τύποι οντοτήτων (Entity Types) και Κλειδιά (Key Attributes)
 - Συσχετίσεις (Relationships) και Τύποι Συσχετίσεων
- ER διάγραμμα για το Σχήμα COMPANY
- Εναλλακτική χρήση γραφικών συμβόλων – UML class diagrams, κλπ