



Διάλεξη 05: Εννοιολογική Σχεδίαση Βάσης Δεδομένων III (Entity Relationship Modelling)

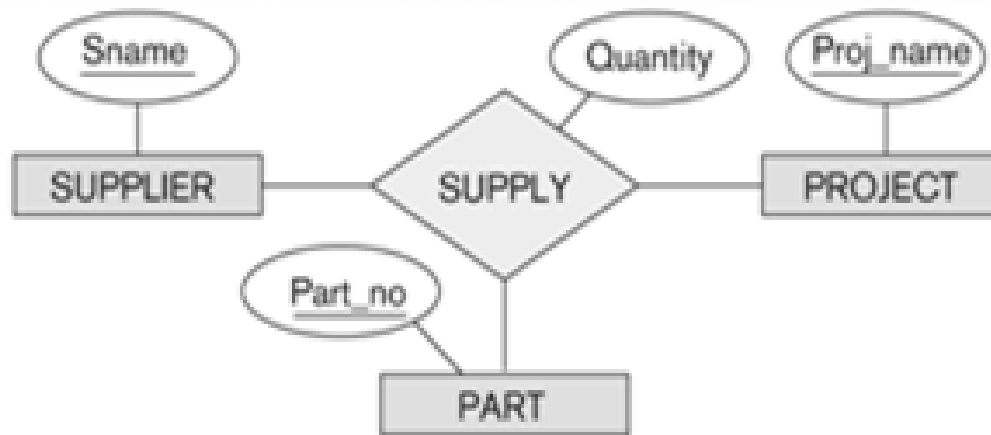
Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:
Εισαγωγή στις έννοιες:

- Συσχετίσεις Υψηλότερου Βαθμού
- UML Σημειογραφία για ER
- Επεκταμένο Μοντέλο ER (EER)

Διδάσκων: Παναγιώτης Ανδρέου

Συσχετίσεις Υψηλότερου Βαθμού

- Τύποι συσχετίσεων βαθμού N
 - 2-οντότητες → δυαδική (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)



Π.χ.,
(Supplier, Part, Project, Qty)
(Lenovo, Thinkpad T60,
SensorNet,1)
(Lenovo, Thinkpad T60,
Mobileware,1)
(IBM, Thinkpad T60,
SensorNet,2)

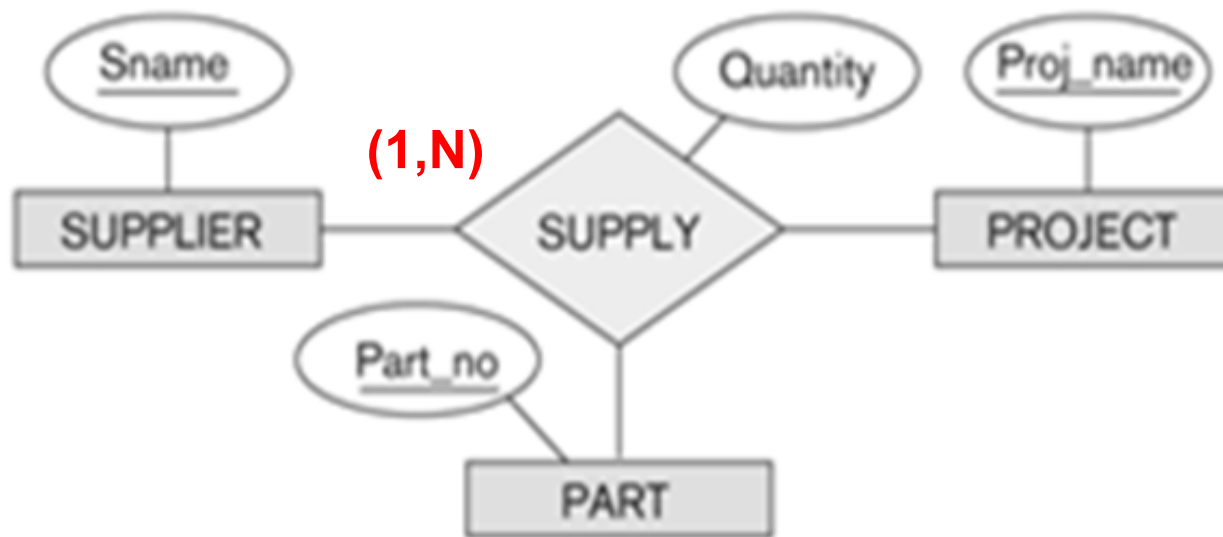
Συσχετίσεις Υψηλότερου Βαθμού (συν.)

Άλλα Παραδείγματα

- **(3-αδική) Μάθημα: Student, Subject, Room**
 - (1111, ΕΠΛ342, ΧΩΔ109)
 - (1112, ΕΠΛ342, ΧΩΔ109)
 -
- **(4-αδική) Παιχνίδι: Team1, Team2, Referee, Stadium**
 - (ΑΠΟΕΛ, ΟΜΟΝΟΙΑ, Ιωάννου, ΓΣΠ)
 - (ΑΝΟΡΘΩΣΗ, ΑΠΟΛΛΩΝΑΣ, Χρίστου, ΓΣΖ)
 -
- **(5-αδική) Παιχνίδι: Team1, Team2, Referee, Stadium, Αγωνιστική**
 - (ΑΠΟΕΛ, ΟΜΟΝΟΙΑ, Ιωάννου, ΓΣΠ, 7^η)
 - (ΑΝΟΡΘΩΣΗ, ΑΠΟΛΛΩΝΑΣ, Χρίστου, ΓΣΠ, 8^η)
 -

Συσχετίσεις Υψηλότερου Βαθμού (συν.)

- Για την διατύπωση **δομικών περιορισμών** μιας **n-αδικής** συσχέτισης προτρέπεται η χρήση του **(min,max)**.
- **Παράδειγμα**
 - **(1,N)**: «Κάθε **SUPPLIER** συμμετέχει στη σχέση **SUPPLY** από **1** μέχρι **N** φορές»,
π.χ.,
 - (Lenovo, Thinkpad T60, SensorNet,1)
 - (Lenovo, Thinkpad T60, Mobileware,1)



Συσχετίσεις Υψηλότερου Βαθμού (συν.)

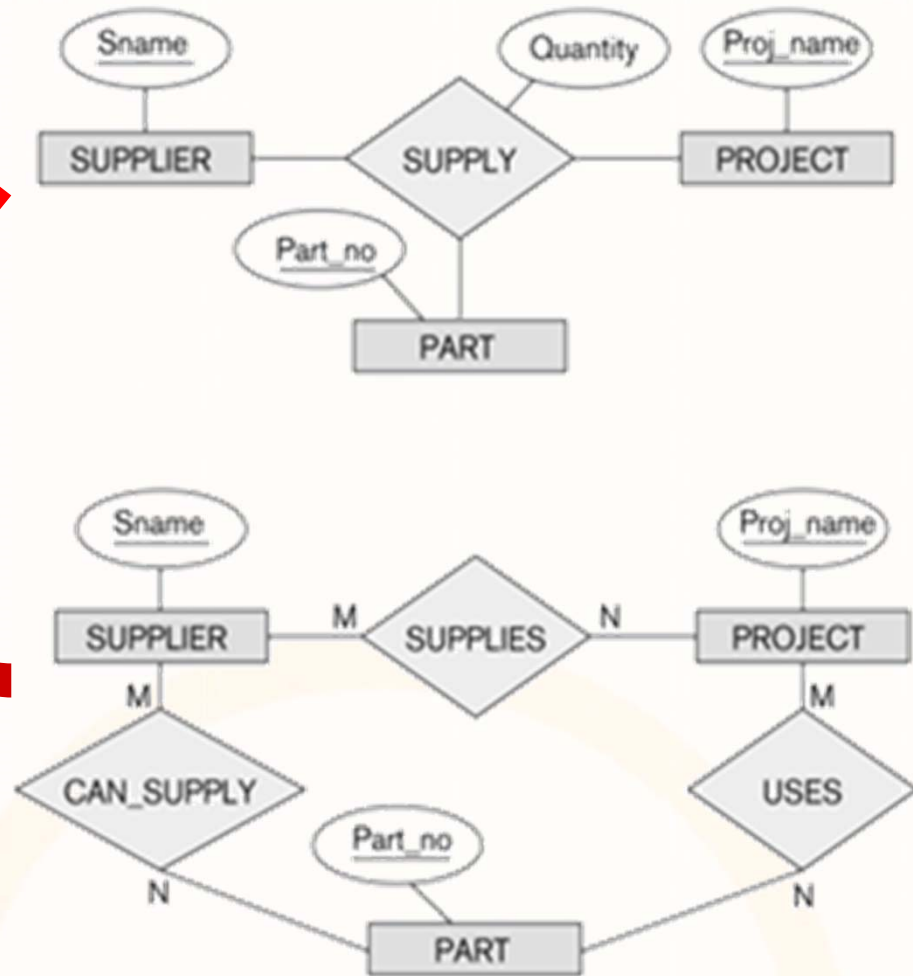
Μετατροπή σε Πολλαπλές Δυαδικές Συσχετίσεις

- **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

Μετατροπή n-αδικών σε 2-αδικές σχέσεις

Προσπάθεια 1

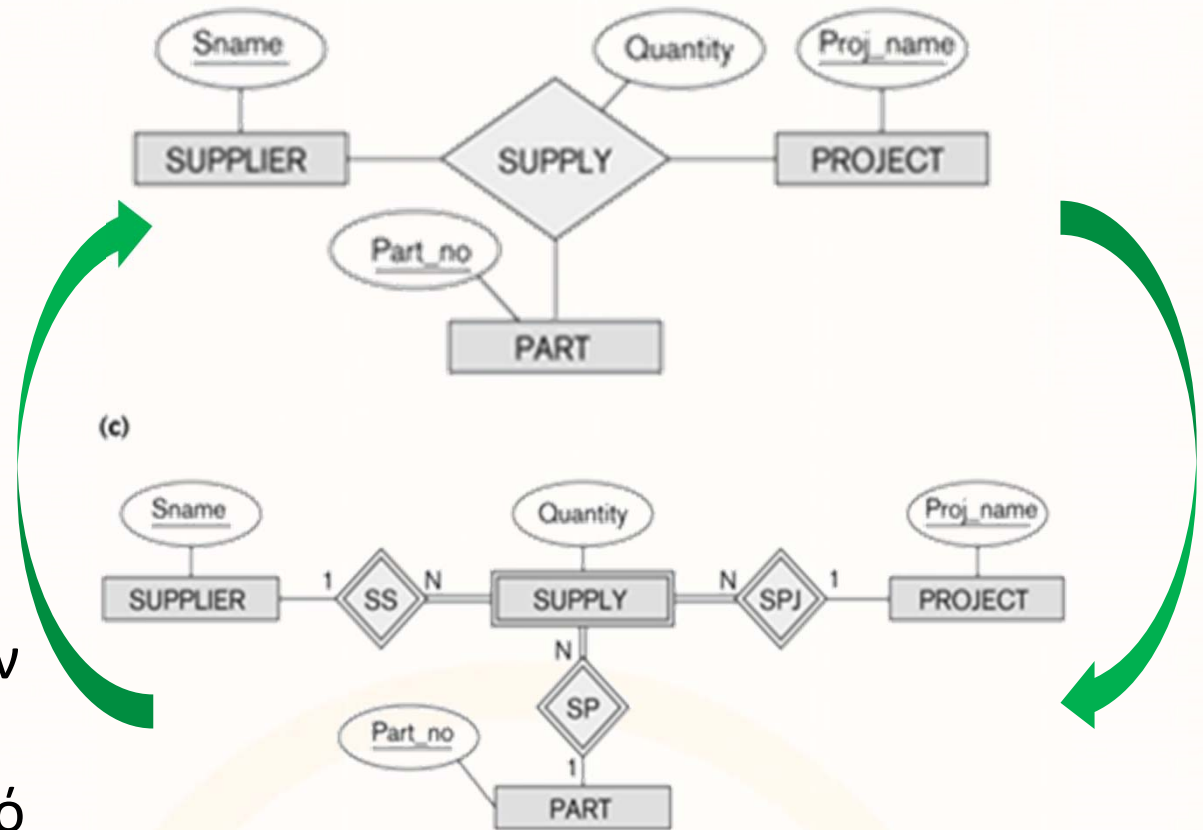
- Διάσπαση των n-αδικών σχέσεων σε ατομικές 2-αδικές σχέσεις
- **Ερώτηση:** Είναι ισοδύναμες αναπαραστάσεις;
- **Απάντηση: ΟΧΙ.**
Παράδειγμα: Ύπαρξη **(supplier, part)** στο διαμορφωμένο μοντέλο δεν σημαίνει ότι υπάρχει **(supplier, part, project)** ενώ στο αρχικό μοντέλο είναι ρητό.



Μετατροπή n-αδικών σε 2-αδικές σχέσεις

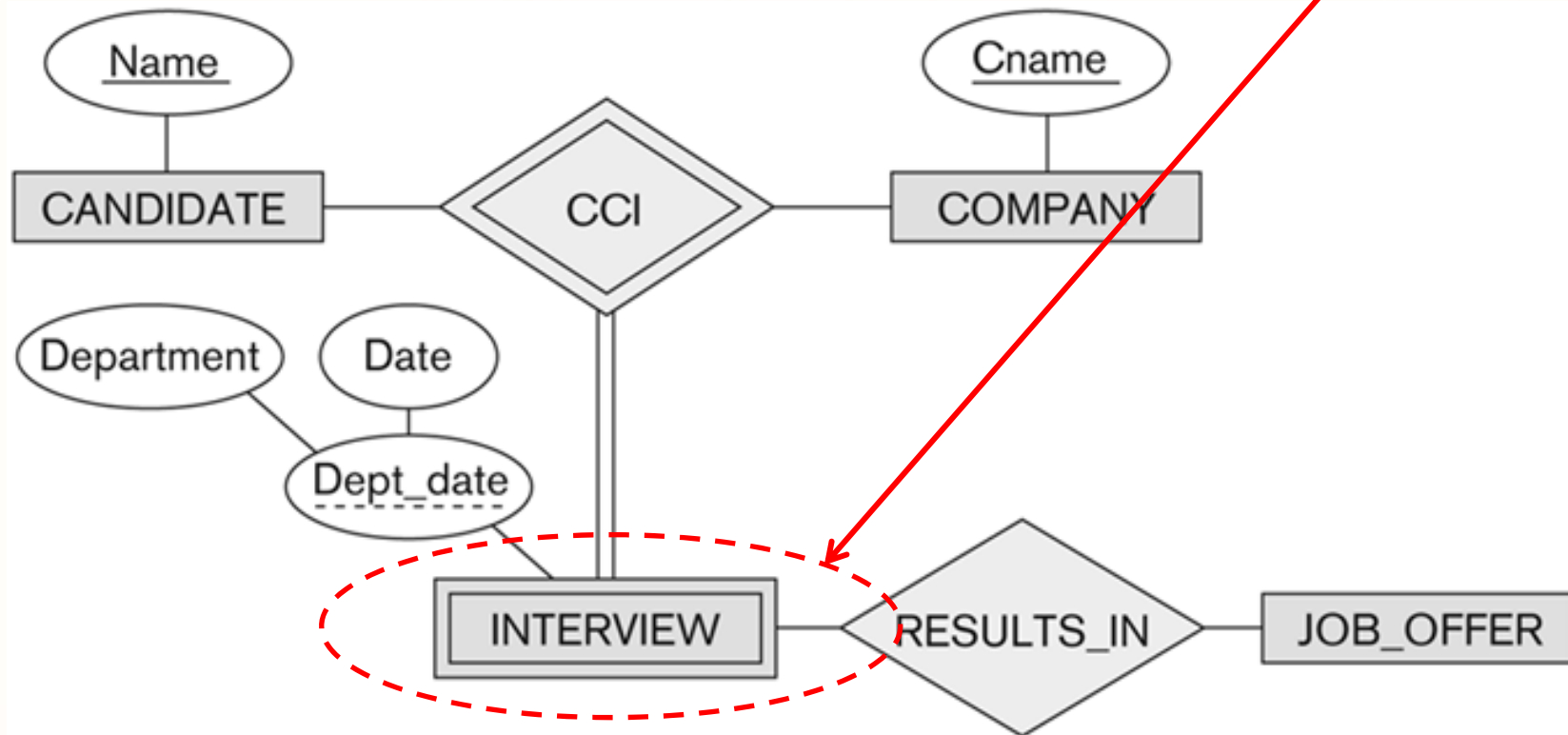
Προσπάθεια 2

- Μετατροπή n-αδικών σχέσεων σε ασθενείς οντότητες
- **Ερώτηση:** Είναι ισοδύναμες αναπαραστάσεις;
- **Απάντηση:** **NAI** Γιατί;
- Λόγω των προσδιορίζων σχέσεων, το κλειδί της SUPPLY αποτελείται από τα κλειδιά όλων των οντοτήτων που συμμετέχουν στην σχέση



Μετατροπή n-αδικών σε 2-αδικές σχέσεις (συν.)

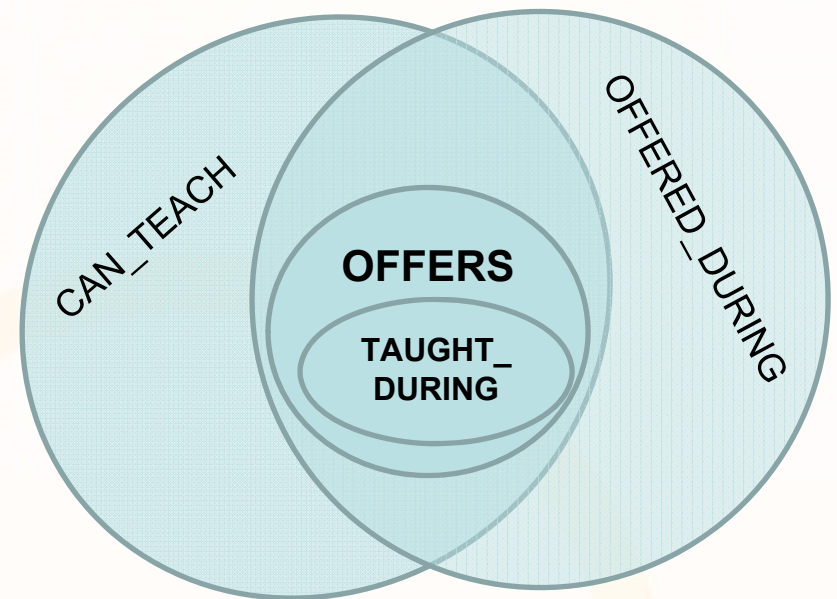
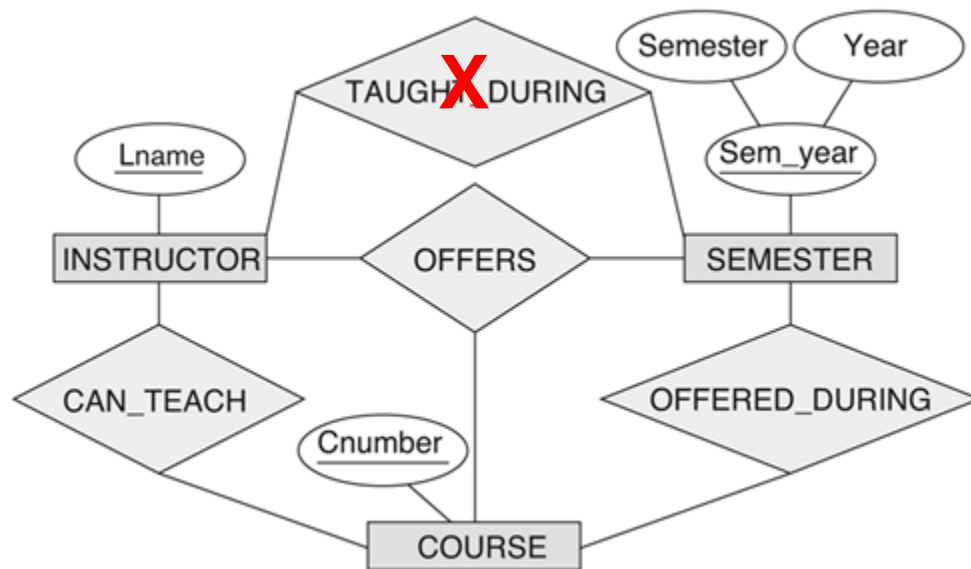
- Σημειώστε ότι είναι δυνατό να υπάρχουν και **ασθενείς τύποι οντοτήτων (π.χ., Interview)** οι οποίες να συμμετέχουν σε συσχετίσεις βαθμού > 2



- * Το Interview χαρακτηρίζεται από το **Candidate.Name**, **Company.Cname** και **Interview.Dept_date**

Μετατροπή n-αδικών σε 2-αδικές σχέσεις (συν.)

- Σημειώστε ότι εάν μια δυαδική συσχέτιση απορρέει από μια υψηλότερου βαθμού σχέση για όλες τις περιπτώσεις τότε η δυαδική αυτή σχέση είναι περιττή και μπορεί να αφαιρεθεί
 - π.χ., η δυαδική σχέση **TAUGHT_DURING** απορρέει απευθείας από την τριαδική συσχέτιση **OFFERS**



Διάγραμμα Venn:
Συνολοκεντρική Απεικόνιση

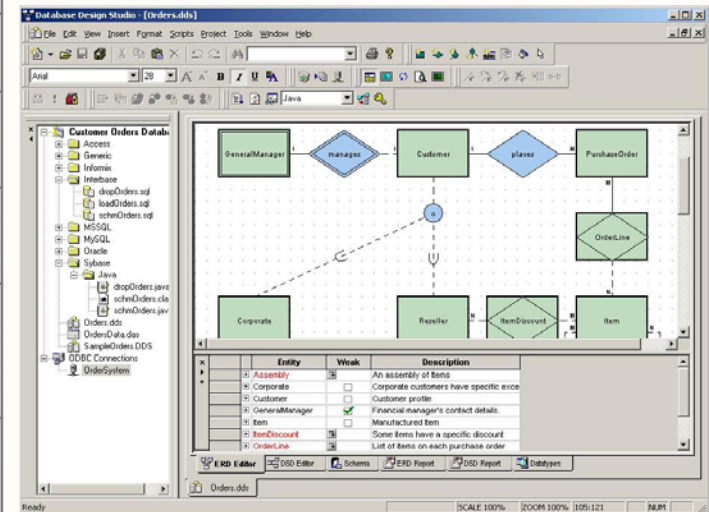
Εργαλεία Μοντελοποίησης Δεδομένων

- 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**.

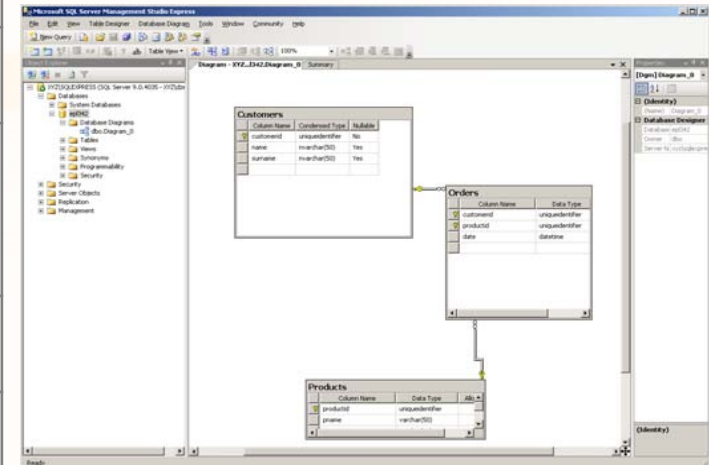
Εργαλεία για Εννοιολογική Σχεδίαση

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
Microsoft	SQL Server Management Studio	Database Modeling, Development, Maintenance
Persistence Inc.	Pwertier	Mapping from O-O to relational model
Rational (IBM)	Rational Rose	UML Modeling & application generation in C++/JAVA
Chillsource	Database Design Studio (DDS)	Modeling, forward engineering and Backward Engineering of relational db Supports: Chen Entity Relationship Diagram
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio	Visio Enterprise	Data modeling, design/reengineering Visual Basic/C++

Database Design Studio (DDS)



SQL Server Management Studio



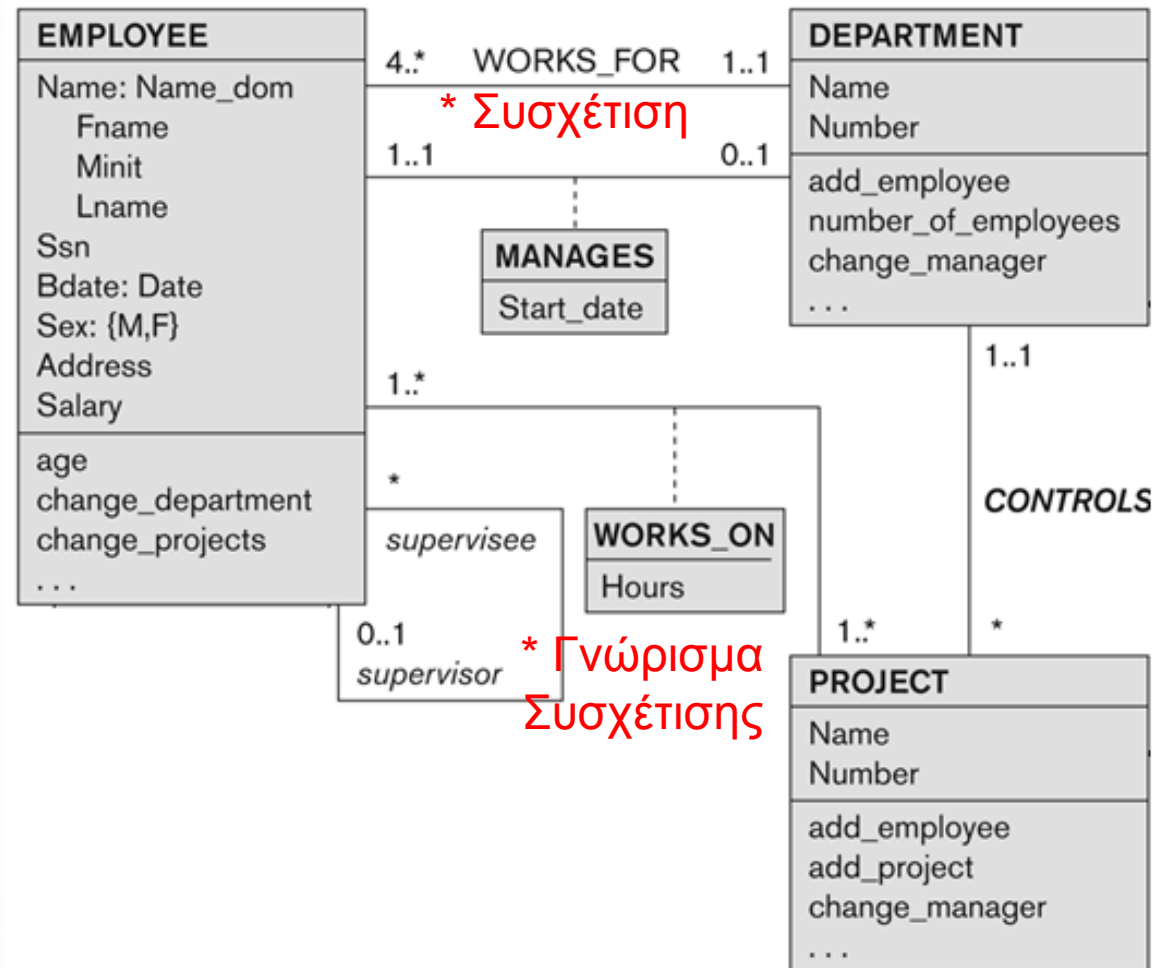
Διάγραμμα Κλάσης UML

- Διαγράμματα Κλάσης **UML** χρησιμοποιούνται σε Αντικειμενοστρεφή Σχεδίαση

* Γνωρίσματα

* Λειτουργίες

- **Σημείωση:** Οι περιορισμοί min..max διαβάζονται αντίστροφα απ' ότι στο διάγραμμα ER

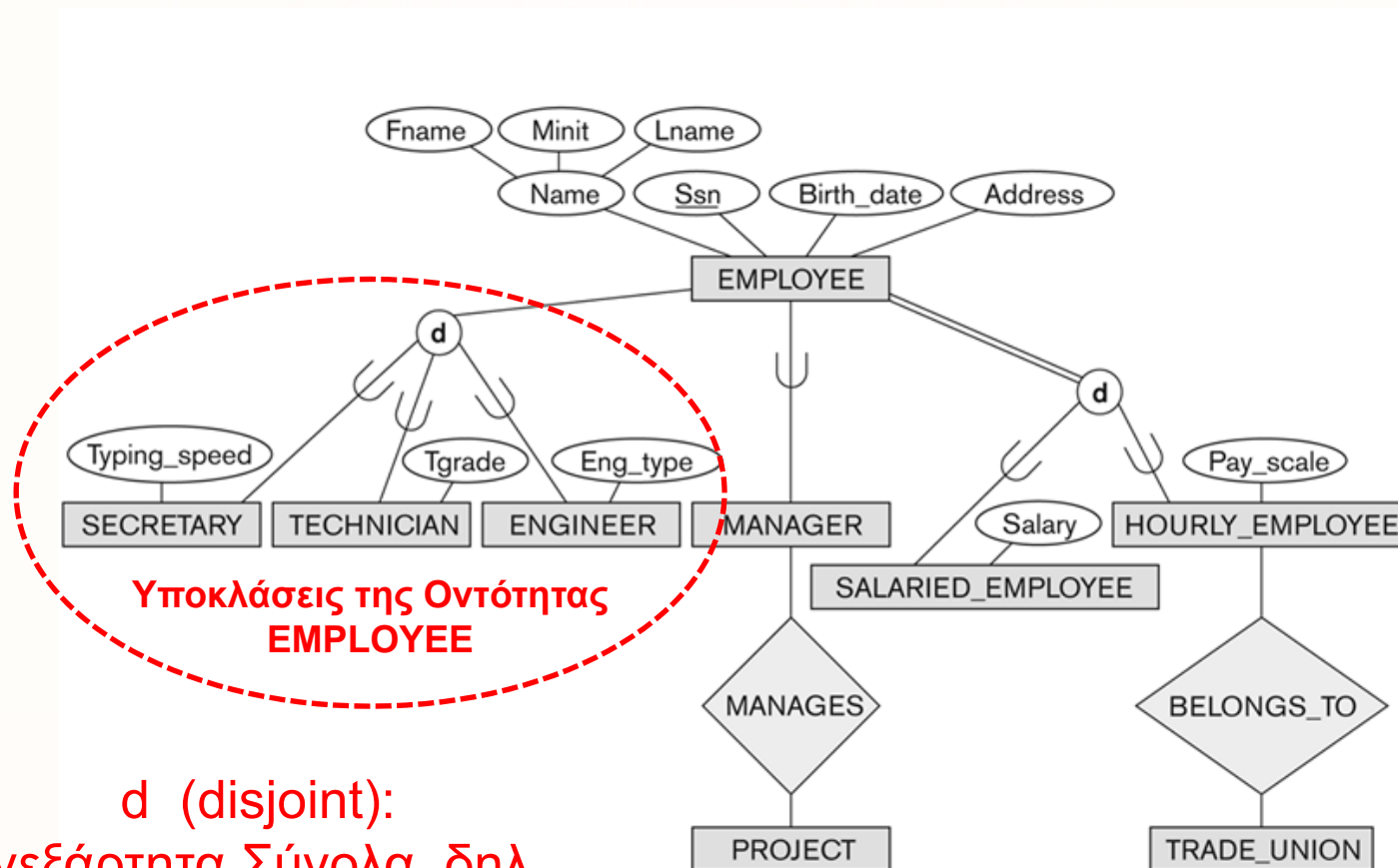


Το Επεκταμένο (Enhanced) ER - (EER)

- Το Διάγραμμα ER στην αρχική του μορφή δεν υποστήριζε την έννοια της **εξειδίκευσης (specialization)** και την έννοια της **γενίκευσης (generalization)**
 - π.χ., Εξειδικεύσεις **CS-Student**, **Math-Student**, της **Student**
- Το **Επεκταμένο ER (EER - Enhanced ER ή Extended ER)** παρέχει επεκτάσεις για πληρέστερη μοντελοποίηση των απαιτήσεων
 - Το EER συμπεριλαμβάνει **αντικειμενοστρεφείς έννοιες** όπως αυτή της **κληρονομικότητας**
- Ένας **τύπος οντότητας** μπορεί να έχει διάφορες υποκατηγοριοποιήσεις, π.χ., ο **EMPLOYEE** μπορεί να χωριστεί σε:
 - Βασισμένο στον **τύπο εργασίας** ενός EMPLOYEE
 - π.χ., **SECRETARY**, **ENGINEER**, **TECHNICIAN**, ...
 - Βασισμένο στον **τρόπο πληρωμής** ενός EMPLOYEE
 - π.χ., **SALARIED_EMPLOYEE**, **HOURLY_EMPLOYEE**
 - Βασισμένο σε **άλλα χαρακτηριστικά**
 - Π.χ., **MANAGER**

Το Επεκταμένο (Enhanced) ER - (EER) (συν.)

- Τα διαγράμματα παρέχουν **επεκτάσεις** για να αναπαραστήσουν αυτές τις **υπό-ομαδοποιήσεις** οι οποίες ονομάζονται **υποκλάσεις** (*subclasses*) ή **υποτύποι** (*subtypes*)



d (disjoint):

Ανεξάρτητα Σύνολα, δηλ.,
($Secretary \cap Technician = \emptyset$)

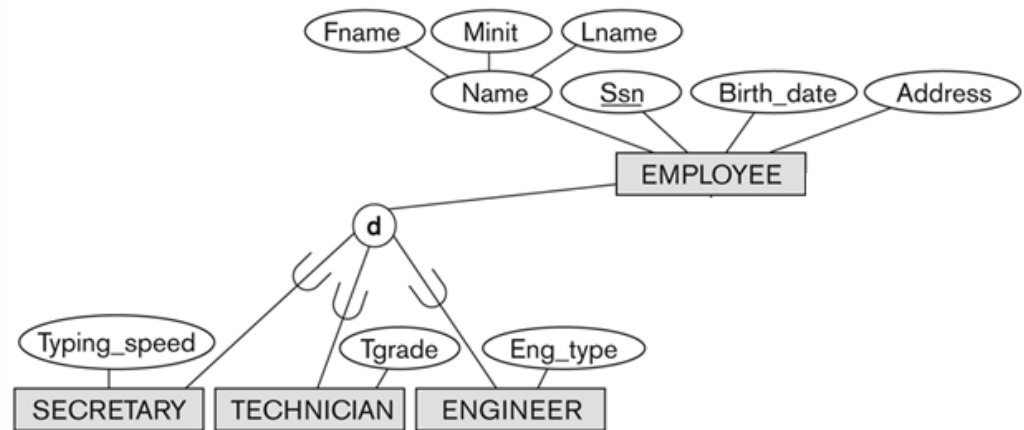
Το Επεκταμένο (Enhanced) ER - (EER) (συν.)

- Συσχετίσεις Υπερκλάσης / Υποκλάσης

- EMPLOYEE/SECRETARY

- EMPLOYEE/TECHNICIAN

- EMPLOYEE/ENGINEER



- Οι πιο πάνω συσχετίσεις μεταξύ υπερκλάσης / υποκλάσης ονομάζονται συχνά και συσχετίσεις **IS-A (relationships)**

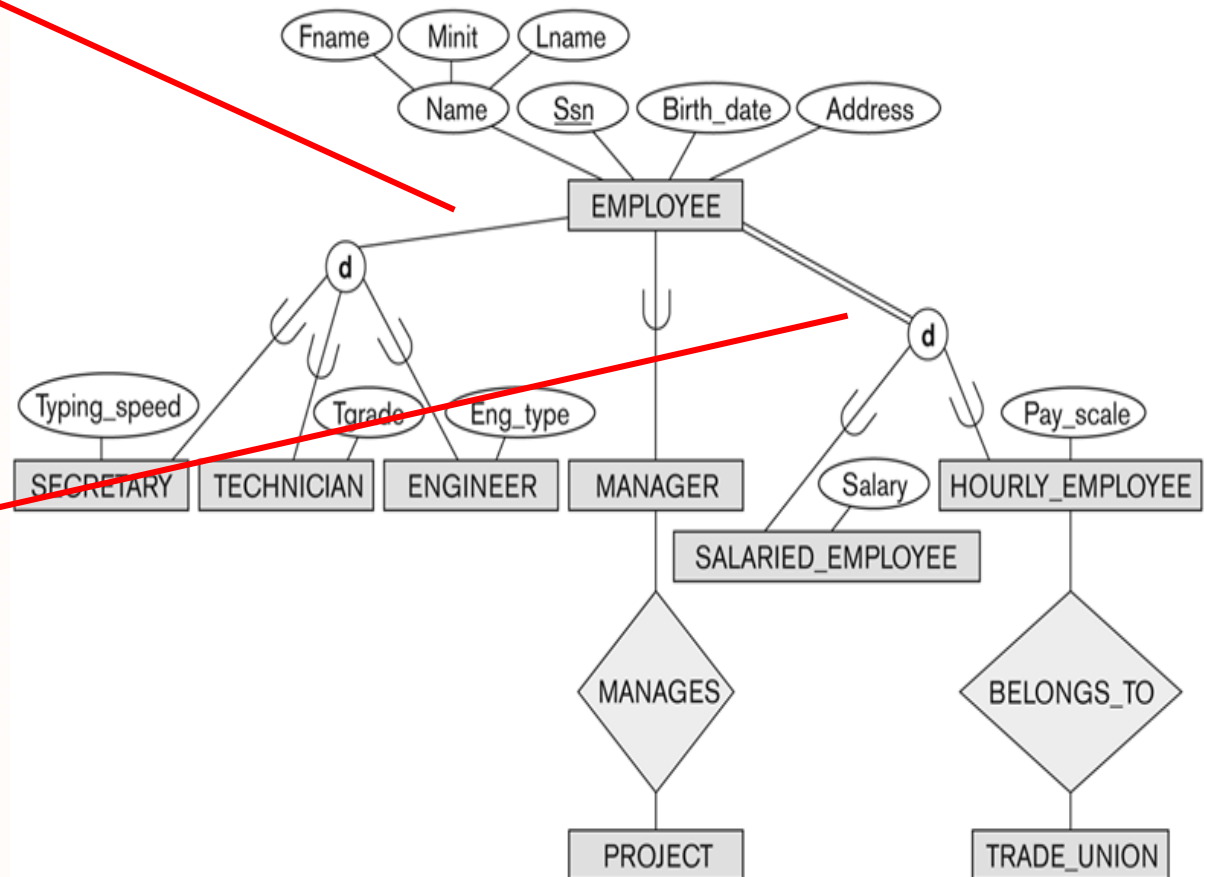
- Π.χ., SECRETARY IS-A EMPLOYEE, TECHNICIAN IS-A EMPLOYEE,

- Σημειώστε ότι μια **οντότητα ΔΕΝ** μπορεί να **υπάρχει** στη βάση δεδομένων **απλά** ως **μέλος της υποκλάσης**.

- Μια οντότητα **ΠΡΕΠΕΙ** να είναι και μέλος της **υπερκλάσης** (δηλαδή η υποκλάση κληρονομεί όλα τα γνωρίσματα της υπερκλάσης).

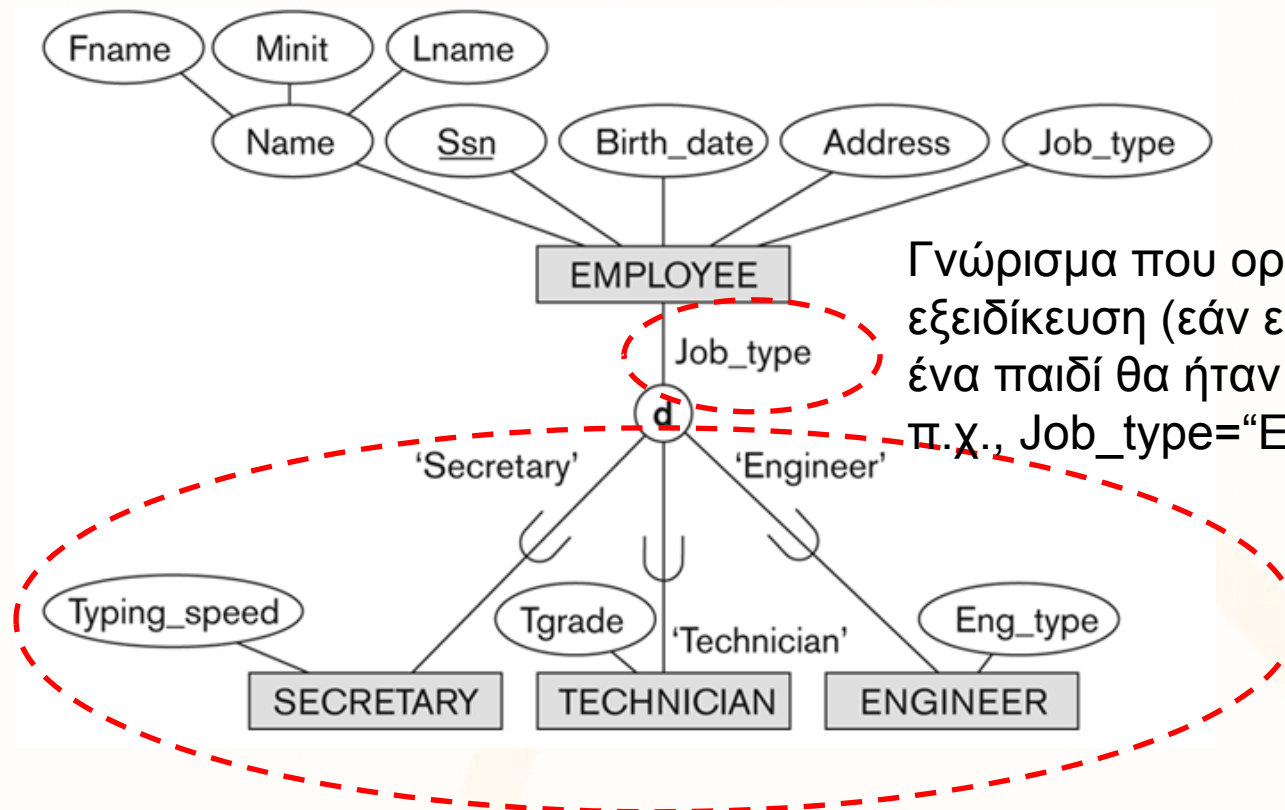
Περιορισμοί Συμμετοχής σε EER

- Μια οντότητα δεν είναι υποχρεωτικό να ανήκει σε ΟΛΕΣ τις υποκλάσεις
- Τι σημαίνει η ΜΕΡΙΚΗ συμμετοχή;
- Ένας EMPLOYEE **μπορεί** να είναι SECRETARY είτε TECHNICIAN είτε ENGINEER ή/και MANAGER
- Τι σημαίνει η ΟΛΙΚΗ συμμετοχή;
- **Κάθε** EMPLOYEE πληρώνεται με την ώρα ή με τον μήνα



Εξειδίκευση σε EER

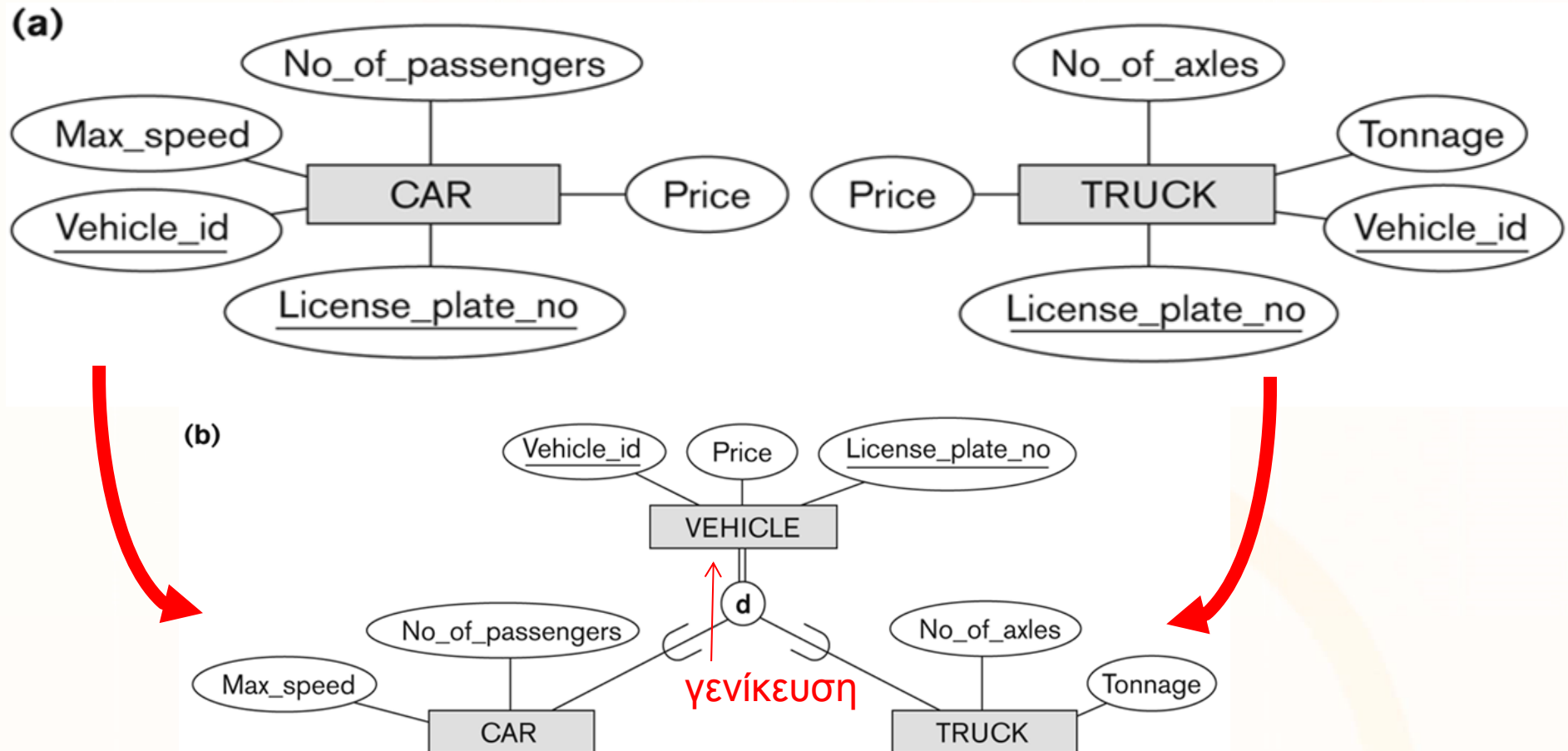
- Εξειδίκευση: η δήλωση του συνόλου υποκλάσεων μιας υπερκλάσης βάσει κάποιου αναγνωριστικού χαρακτηριστικού
- π.χ., Εξειδίκευση του EMPLOYEE βάσει του *job type*.



Γνώρισμα που ορίζει την εξειδίκευση (εάν είχε μόνο ένα παιδί θα ήταν συνθήκη, π.χ., `Job_type="Engineer"`)

Γενίκευση σε EER

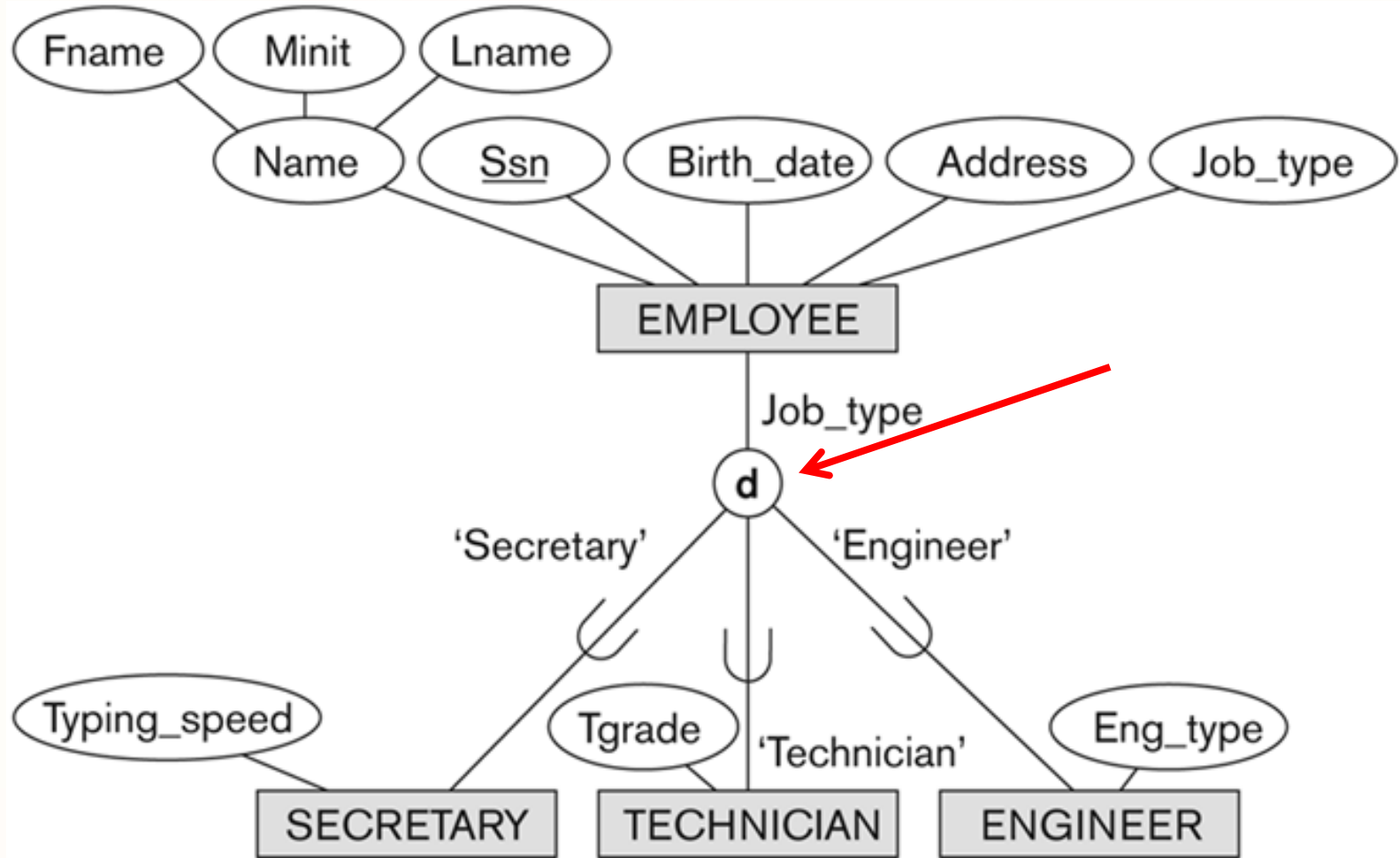
- **Γενίκευση (Generalization):** η αντίστροφη διεργασία της εξειδίκευσης (δηλ., δήλωση **υπερκλάσης** από **υποκλάσεις**)



Περιορισμοί με Γενικεύσεις/Εξειδικεύσεις

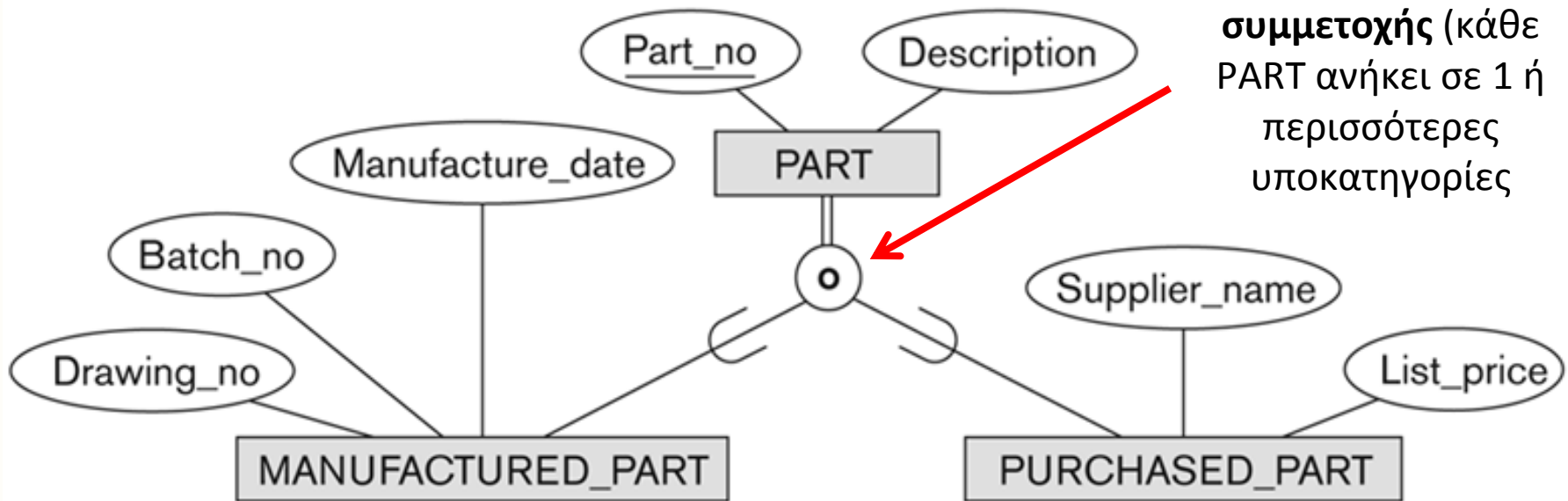
- **Οντότητες, Εξειδικεύσεις και Γενικεύσεις ονομάζονται Κλάσεις (Classes)**
- Υπάρχουν δυο είδη **περιορισμών** που εφαρμόζονται σε εξειδικεύσεις/γενικεύσεις:
 - **Περιορισμός Μη-Επικάλυψης (Disjointness Constraint):**
 - **d (disjoint):** Μη Επικάλυψης (\cap Υποκλάσεων = \emptyset) ή
 - **o (overlap):** Επικάλυψης ($(\cap$ Υποκλάσεων $\neq \emptyset)$)
 - **Περιορισμός Πληρότητας ή Συμμετοχής (Completeness Constraint):**
 - **Ολική Συμμετοχή (Total) Υπερκλάσης ή**
 - **Μερική Συμμετοχή (Partial) Υπερκλάσης**
 - Σημειώστε ότι το αντίστροφο ισχύει εξ' ορισμού: κάθε Υποκλάση έχει ολική συμμετοχή στην συσχέτιση με την υπερκλάση.

Περιορισμός Μη-Επικάλυψης (Disjointness)



Περιορισμός Επικάλυψης (Overlap)

Επιπλέον υπάρχει
περιορισμός ολικής
συμμετοχής (κάθε
PART ανήκει σε 1 ή
περισσότερες
υποκατηγορίες



Συνεπώς, υπάρχουν τέσσερις συνδυασμοί **εξειδίκευσης/γενίκευσης**:

- **Disjoint, total** (d + διπλή γραμμή)
- **Disjoint, partial** (d + μονή γραμμή)
- **Overlapping, total** (o + διπλή γραμμή)
- **Overlapping, partial** (o + μονή γραμμή)

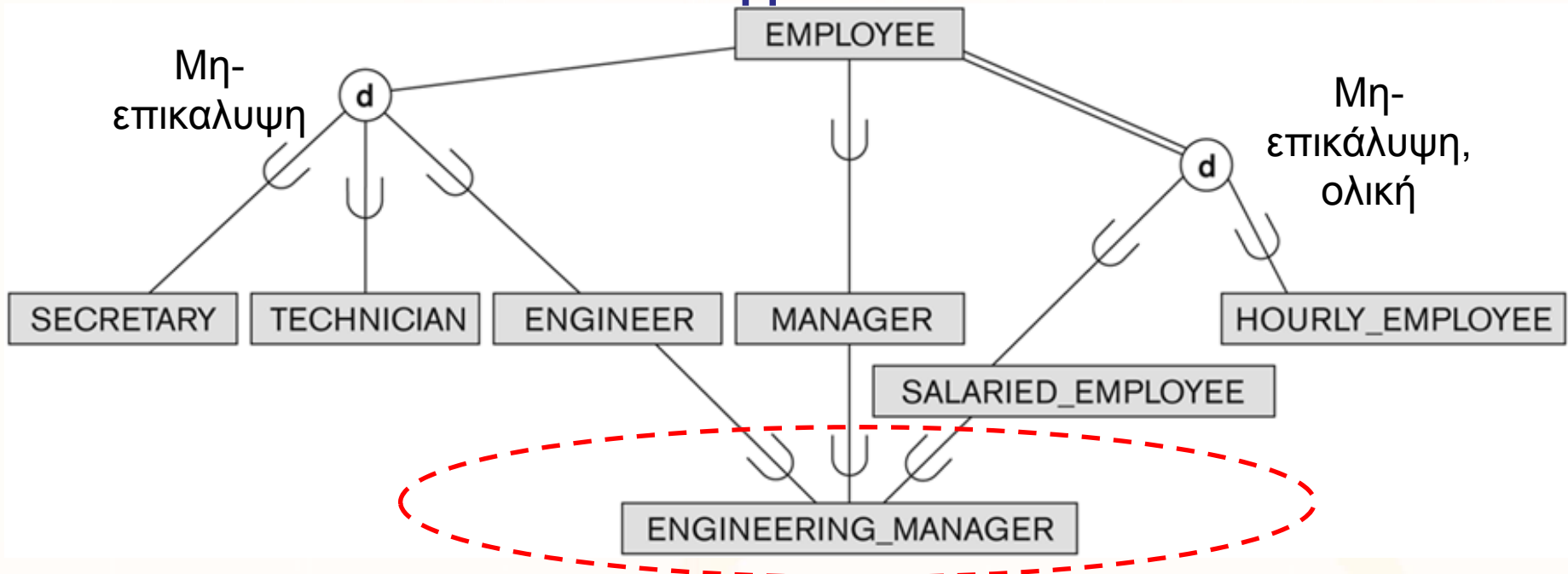
Ιεραρχίες και Πλέγματα

- Μια υποκλάση μπορεί να εξειδικεύεται περαιτέρω δημιουργώντας μια **Ιεραρχία (hierarchy)** ή **Πλέγμα (lattice)**
 - **Ιεραρχία (Hierarchy):** Κάθε κόμβος (κλάση) έχει ένα γονέα (δηλαδή μια **δενδρική** ιεραρχία)
 - *Μονή Κληρονομικότητα (single inheritance);*
 - **Πλέγμα (Lattice):** Κάθε κόμβος (κλάση) μπορεί να έχει **πολλαπλούς γονείς** (δηλαδή ένας **γράφος**)
 - *Πολλαπλή Κληρονομικότητα (multiple inheritance)*

Παράδειγμα Πλέγματος I

- Ένας Engineering Manager κληρονομεί γνωρίσματα από τον Engineer, Manager και Salaried_Employee.

Πλέγμα/Lattice



Διαμοιραζόμενη Υποκλάση (Shared Subclass)
(γνωρίσματα κληρονομούνται μόνο 1 φορά)

Παράδειγμα Πλέγματος II

