

## Η Θεωρία των Βάσεων Δεδομένων

### Εισαγωγή στις Βάσεις Δεδομένων

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Τα συστήματα βάσεων δεδομένων τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά.

### Τα Δεδομένα και οι Πληροφορίες

Με τον όρο *πληροφορία* αναφερόμαστε συνήθως σε ειδήσεις, γεγονότα και έννοιες που αποκτάμε από την καθημερινή μας επικοινωνία και τα θεωρούμε ως αποκτηθείσα γνώση, ενώ τα δεδομένα μπορούν να είναι μη κατάλληλα επεξεργασμένα και μη ταξινομημένα σύνολα πληροφοριών. Ένας αυστηρός ορισμός για το τι είναι δεδομένα και τι είναι πληροφορία, σύμφωνα με την επιτροπή ANSI των ΗΠΑ, είναι ο εξής :

- *Δεδομένα (data)* είναι μια παράσταση, όπως γράμματα, αριθμοί, σύμβολα κ.ά. στα οποία μπορούμε να δώσουμε κάποια σημασία (έννοια).
- *Πληροφορία (information)* είναι η σημασία που δίνουμε σ' ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε έτσι κάποια χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και βοηθιόμαστε έτσι στο να λάβουμε σωστές αποφάσεις.

Τα δεδομένα μπορούν να θεωρηθούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που μπορούν να υποστούν διαχείριση και επεξεργασία. Η συλλογή και αποθήκευση ενός τεράστιου όγκου δεδομένων όπως απαιτούν οι κοινωνικές συνθήκες σήμερα, δεν λύνει τελείως το πρόβλημα της σωστής οργάνωσης και ταξινόμησης των δεδομένων. Τα δεδομένα θα πρέπει να οργανωθούν με τέτοιο τρόπο έτσι ώστε να μπορούμε να τα εντοπίζουμε και να τα αξιοποιούμε εύκολα και γρήγορα και τη στιγμή που τα χρειαζόμαστε.

Ένα κλασικό παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν για παράδειγμα ο τηλεφωνικός κατάλογος της πόλης της Θεσσαλονίκης, όπου οι συνδρομητές δεν θα ήταν καταχωρημένοι αλφαβητικά σύμφωνα με το επώνυμο και το όνομά τους, αλλά εντελώς τυχαία. Ένας τέτοιος τηλεφωνικός κατάλογος θα περιείχε μια τεράστια ποσότητα δεδομένων αλλά θα ήταν ουσιαστικά άχρηστος.

Ένα άλλο παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν μια πολύ μεγάλη βιβλιοθήκη με χιλιάδες τόμους βιβλίων και χωρίς να διαθέτει κάποιο υποτυπώδες

σύστημα οργάνωσης και ταξινόμησης των βιβλίων. Ούτε οι υπάλληλοι της βιβλιοθήκης θα μπορούσαν να κάνουν τη δουλειά τους αλλά ούτε και οι επισκέπτες θα μπορούσαν να αξιοποιήσουν την πληθώρα των πληροφοριών που περιέχονται στα βιβλία. Εκτός λοιπόν από τη μόνιμη αποθήκευση των δεδομένων, χρειαζόμαστε και κάποιους τρόπους ευέλικτης και αποδοτικής οργάνωσής τους.

Χαρακτηριστικά παραδείγματα δεδομένων που απαιτούν σωστή και αποδοτική οργάνωση είναι τα εξής :

- Τα στοιχεία υπαλλήλων, πελατών, προμηθευτών και παραγγελιών μιας εμπορικής επιχείρησης.
- Τα στοιχεία υλικών μιας αποθήκης.
- Τα στοιχεία ταινιών, πελατών και δανεισμών μιας βιντεολέσχης.
- Τα στοιχεία υπαλλήλων, γιατρών, ασθενών αλλά και υλικών ενός νοσοκομείου.
- Τα στοιχεία βιβλίων, χρηστών (δανειστών) και δανεισμών μιας βιβλιοθήκης.

### **Η Οργάνωση Αρχείων**

Ο πιο γνωστός τρόπος οργάνωσης δεδομένων με τη χρήση ηλεκτρονικών υπολογιστών είναι σε αρχεία εγγραφών. Για να κατανοήσουμε καλύτερα ορισμένες έννοιες, θα εξετάσουμε την περίπτωση ενός αρχείου πελατών και παραγγελιών μιας εμπορικής επιχείρησης. Για να οργανώσουμε σωστά το αρχείο μας, θα πρέπει να δημιουργήσουμε καρτέλες για τους πελάτες, αλλά και για τις παραγγελίες τους αργότερα, που θα πρέπει να περιέχουν τα εξής στοιχεία ανά πελάτη :

- Κωδικός
- Επώνυμο
- Όνομα
- Διεύθυνση
- ΤΚ
- Πόλη
- Τηλέφωνο
- ΑΦΜ
- ΔΟΥ

Η αντιστοίχιση του παλιού τρόπου οργάνωσης με τις καρτέλες σε σχέση με τον σύγχρονο ηλεκτρονικό τρόπο οργάνωσης, έχει ως εξής :

- Συρτάρι – Αρχείο Δεδομένων
- Καρτέλα πελάτη – Εγγραφή του αρχείου δεδομένων
- Στοιχείο της καρτέλας – Πεδίο της εγγραφής

Ένα *αρχείο (file)* θα μπορούμε να το χαρακτηρίσουμε σαν ένα σύνολο που αποτελείται από οργανωμένα ομοειδή στοιχεία. Τα στοιχεία ενός αρχείου μπορούμε να τα οργανώσουμε σε λογικές ενότητες και το σύνολο των στοιχείων που περιέχει μια λογική ενότητα καλείται *εγγραφή (record)*. Το κάθε στοιχείο της εγγραφής καλείται *πεδίο (field)*. Το πεδίο αποτελεί και τη μικρότερη δυνατή υποδιαίρεση των στοιχείων ενός αρχείου. Ένα πεδίο χαρακτηρίζεται από τον μέγιστο αριθμό των

χαρακτήρων (bytes) που απαιτούνται για την καταχώρησή του στη μνήμη του υπολογιστή και που αποκαλείται *μήκος του πεδίου (field length)*.

Σε μια οργάνωση αρχείου όπως είναι οι πελάτες μιας εμπορικής επιχείρησης που είδαμε νωρίτερα, τα αντίστοιχα πεδία όλων των εγγραφών καταλαμβάνουν τον ίδιο αριθμό σε bytes που είναι αυτός που έχουμε ορίσει κατά τη δημιουργία του αρχείου. Για παράδειγμα, αν ορίσαμε ότι το πεδίο Επώνυμο θα έχει μήκος 15 χαρακτήρες, τότε το πεδίο της εγγραφής του πελάτη με επώνυμο Παπαδόπουλος, αλλά και το πεδίο της εγγραφής του πελάτη με επώνυμο Βες θα καταλαμβάνουν από 15 bytes στη μνήμη του υπολογιστή, ενώ αν ένας πελάτης ονομάζεται Παπαχριστοδουλόπουλος, τότε θα γίνει αποκοπή του επωνύμου του και θα καταχωρηθούν στη μνήμη του υπολογιστή μόνο τα 15 πρώτα γράμματα, δηλ. τα Παπαχριστοδουλό.

Ένα πεδίο χαρακτηρίζεται ακόμη και από το είδος των δεδομένων που μπορεί να περιέχει, όπως :

- *Αλφαριθμητικό (alphanumeric)*, μπορεί να περιέχει γράμματα, ψηφία ή και ειδικούς χαρακτήρες.
- *Αριθμητικό (numeric)*, μπορεί να περιέχει μόνο αριθμούς.
- *Αλφαβητικό (alphabetic)*, μπορεί να περιέχει μόνο γράμματα (αλφαβητικούς χαρακτήρες).
- *Ημερομηνίας (date)*, μπορεί να περιέχει μόνο ημερομηνίες.
- *Διαδικό (binary)*, μπορεί να περιέχει ειδικού τύπου δεδομένα, όπως εικόνες, ήχους κ.ά.
- *Λογικό (logical)*, μπορεί να περιέχει μόνο μία από δύο τιμές, οι οποίες αντιστοιχούν σε δύο διακριτές καταστάσεις και μπορούν να χαρακτηρισθούν σαν 0 και 1 ή σαν αληθές (true) και ψευδές (false).
- *Σημειώσεων (memo)*, μπορεί να περιέχει κείμενο με μεταβλητό μήκος, το οποίο μπορεί να είναι και αρκετά μεγάλο και συνήθως αποθηκεύεται σαν ξεχωριστό αρχείο από το κύριο αρχείο.

Όσον αφορά τις εγγραφές, χρήσιμοι ορισμοί είναι οι εξής :

- *Μήκος εγγραφής (record length)* καλείται το άθροισμα που προκύπτει από τα μήκη των πεδίων που την αποτελούν.
- *Δομή εγγραφής (record layout)* ή *γραμμογράφηση* καλείται ο τρόπος με τον οποίο οργανώνουμε τα πεδία μιας εγγραφής.
- *Διάβασμα (read)* από αρχείο σημαίνει τη μεταφορά των δεδομένων του αρχείου, που γίνεται συνήθως ανά μία εγγραφή, από το μέσο αποθήκευσης (σκληρό δίσκο ή δισκέτα) στην κεντρική μνήμη του υπολογιστή για επεξεργασία.
- *Γράψιμο (write)* σε αρχείο σημαίνει μεταφορά των δεδομένων του αρχείου, που γίνεται συνήθως ανά μία εγγραφή, από την κεντρική μνήμη του υπολογιστή στο μέσο αποθήκευσης (σκληρό δίσκο ή δισκέτα).

## Προβλήματα της Οργάνωσης Αρχείων

Στα αρχικά στάδια της οργάνωσης αρχείων, ήταν πολύ συνηθισμένη πρακτική η δημιουργία ξεχωριστών εφαρμογών (προγραμμάτων) και ξεχωριστών αρχείων, όπως

για παράδειγμα η δημιουργία ενός αρχείου πελατών και ενός άλλου ανεξάρτητου αρχείου για τις παραγγελίες των πελατών. Τα προβλήματα που προέκυψαν από την πρακτική αυτή είναι τα εξής :

- *Πλεονασμός των δεδομένων (data redundancy)*. Υπάρχει η περίπτωση να έχουμε επανάληψη των ίδιων δεδομένων σε αρχεία διαφορετικών εφαρμογών. Για παράδειγμα, αν έχουμε ένα αρχείο πελατών και ένα αρχείο παραγγελιών αυτών των πελατών, είναι σχεδόν σίγουρο ότι θα υπάρχουν κάποια στοιχεία των πελατών που θα υπάρχουν και στα δύο αρχεία.
- *Ασυνέπεια των δεδομένων (data inconsistency)*. Αυτό μπορεί να συμβεί όταν υπάρχουν τα ίδια στοιχεία των πελατών (πλεονασμός) και στο αρχείο πελατών και στο αρχείο παραγγελιών και χρειασθεί να γίνει κάποια αλλαγή στη διεύθυνση ή στα τηλέφωνα κάποιου πελάτη, οπότε είναι πολύ πιθανό να γίνει η διόρθωση μόνο στο ένα αρχείο και όχι και στο άλλο.
- *Αδυναμία μερισμού δεδομένων (data sharing)*. Μερισμός δεδομένων σημαίνει δυνατότητα για κοινή χρήση των στοιχείων κάποιων αρχείων. Για παράδειγμα, ο μερισμός δεδομένων θα ήταν χρήσιμος αν με την παραγγελία ενός πελάτη μπορούμε να έχουμε πρόσβαση την ίδια στιγμή στο αρχείο πελατών για να δούμε το υπόλοιπο του πελάτη και μετά στο αρχείο της αποθήκης για να δούμε αν είναι διαθέσιμα τα προϊόντα που παρήγγειλε ο συγκεκριμένος πελάτης. Η αδυναμία μερισμού δεδομένων δημιουργεί καθυστέρηση στη λήψη αποφάσεων και στην εξυπηρέτηση των χρηστών.
- *Αδυναμία προτυποποίησης*. Έχει να κάνει με την ανομοιομορφία και με την διαφορετική αναπαράσταση και οργάνωση των δεδομένων στα αρχεία των εφαρμογών. Η αδυναμία αυτή δημιουργεί προβλήματα προσαρμογής των χρηστών καθώς και προβλήματα στην ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.

## **Οι Βάσεις Δεδομένων και τα ΣΔΒΔ (DBMS)**

Για να δοθεί μια λύση σ' όλα τα παραπάνω προβλήματα, και με βάση το γεγονός ότι η χρήση των ηλεκτρονικών υπολογιστών και συνεπώς η ηλεκτρονική καταχώρηση και επεξεργασία δεδομένων αυξήθηκε κατακόρυφα ήδη από τη δεκαετία του '70 στις μεγάλες επιχειρήσεις και άρα είχαμε πάρα πολλές εφαρμογές να επεξεργάζονται δεδομένα σε πάρα πολλά αρχεία ταυτόχρονα, προτάθηκε η συνένωση όλων των αρχείων μιας εφαρμογής. Εκτός, όμως, από τη συνένωση των αρχείων, απαιτείτο και μια σωστή οργάνωσή τους. Δημιουργήθηκαν έτσι οι Τράπεζες Πληροφοριών ή Βάσεις Δεδομένων (Data Bases).

Μια *Βάση Δεδομένων (ΒΔ)* είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες ταυτόχρονα. Υπάρχει ένα ειδικό λογισμικό το οποίο μεσολαβεί ανάμεσα στις αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται *Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ)* ή *DBMS (Data Base Management System)*. Το ΣΔΒΔ είναι στην ουσία ένα σύνολο από προγράμματα και υπορουτίνες που έχουν να

κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφαλείας κ.ά.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

Ένα *Σύστημα Βάσης Δεδομένων (ΣΒΔ)* ή *DBS (Data Base System)* αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά :

- Το *υλικό (hardware)* αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το *λογισμικό (software)* είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.
- Η *βάση δεδομένων (data base)* αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι *ενοποιημένα (data integration)*, δηλ. δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και *μερισμένα (data sharing)*, δηλ. υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.
- Οι *χρήστες (users)* μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :
  - *Τελικοί χρήστες (end users)*. Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τούς επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
  - *Προγραμματιστές εφαρμογών (application programmers)*. Αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
  - *Διαχειριστής δεδομένων (data administrator – DA)*. Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.

- *Διαχειριστής βάσης δεδομένων (database administrator – DBA)*. Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ.

## Η Αρχιτεκτονική των ΣΔΒΔ

Όπως είδαμε νωρίτερα, ένα ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) έχει σαν αποστολή τη διαχείριση των δεδομένων των αρχείων της βάσης, δηλ. την προσθήκη, διαγραφή, τροποποίηση εγγραφών, την αναζήτηση μέσα στις εγγραφές κ.ά.). Το ΣΔΒΔ δέχεται αιτήσεις από τους χρήστες των εφαρμογών και επικοινωνεί με τα αρχεία της βάσης δεδομένων για να τις διεκπεραιώσει.

Αυτή η κοινή διεπαφή (interface) των εφαρμογών με τα αρχεία αποκαλείται *λογική διεπαφή*. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν και αυτή η ιδιότητα είναι γνωστή ως *ανεξαρτησία δεδομένων*.

Αυτό σημαίνει πρακτικά ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές· ένα πρόβλημα που ταλαιπωρούσε πολύ τους προγραμματιστές παλαιότερων εποχών. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων. Στα ΣΔΒΔ έχει επικρατήσει η λεγόμενη αρχιτεκτονική των τριών επιπέδων (βαθμίδων), όπου τα τρία επίπεδα είναι τα εξής :

- *Εσωτερικό επίπεδο (internal level)*, έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάστασή τους.
- *Εξωτερικό επίπεδο (external level)*, έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και οι διαχειριστές της βάσης δεδομένων.
- *Εννοιολογικό επίπεδο (conceptual level)*, είναι ένα ενδιάμεσο επίπεδο που διασυνδέει τα δύο άλλα επίπεδα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων.

## Οι Οντότητες (Entities)

Με τον όρο *οντότητα (entity)* εννοούμε ένα αντικείμενο, ένα πρόσωπο, μια κατάσταση και γενικά οτιδήποτε μπορεί να προσδιορισθεί σαν ανεξάρτητη ύπαρξη (αυτόνομη μονάδα του φυσικού κόσμου). Για παράδειγμα, σε μια βάση δεδομένων μιας εμπορικής εταιρείας, οντότητες μπορεί να είναι οι εργαζόμενοι, οι πελάτες, οι προμηθευτές, οι παραγγελίες, τα είδη της αποθήκης (προϊόντα) κ.ά.

Το *Μοντέλο Οντοτήτων Συσχετίσεων (Entity Relationship Model, ER Model)* είναι μια διαγραμματική αναπαράσταση της δομής μιας βάσης δεδομένων και χρησιμοποιείται κατά τη φάση του λογικού σχεδιασμού της βάσης. Δηλαδή, δεν ασχολείται με τον τρόπο που αποθηκεύονται τα δεδομένα της βάσης, αλλά με την ταυτοποίηση των δεδομένων και με τον τρόπο με τον οποίο αυτά συσχετίζονται μεταξύ τους.



Θα δούμε ένα παράδειγμα μιας εταιρείας, η οποία περιέχει δεδομένα που αφορούν τους υπαλλήλους της (employees), τα τμήματά της (departments) και τα έργα (projects) που έχουν αναλάβει αυτά τα τμήματα. Ένα τμήμα της εταιρείας μπορεί να εποπτεύει ένα ή περισσότερα έργα (projects) και ένας υπάλληλος ανήκει σ' ένα μόνο τμήμα της εταιρείας αλλά μπορεί να απασχολείται ταυτόχρονα σε πολλά έργα, τα οποία δεν είναι υποχρεωτικό να παρακολουθούνται από το ίδιο τμήμα.

### Οι Ιδιότητες (Attributes)

Με τον όρο *ιδιότητα* ή *χαρακτηριστικό* ή και *πεδίο (attribute)* μιας οντότητας, αναφερόμαστε σ' ένα από τα συστατικά της στοιχεία που την περιγράφουν και την κάνουν να ξεχωρίζει από τα άλλα στοιχεία της ίδιας οντότητας. Για παράδειγμα, η οντότητα ΠΕΛΑΤΗΣ μπορεί να έχει ως ιδιότητες (χαρακτηριστικά) τον κωδικό, το επώνυμο, το όνομα, τη διεύθυνση, το τηλέφωνο, το ΑΦΜ κ.ά., με τη βοήθεια των οποίων μπορούμε να ξεχωρίσουμε τους πελάτες μεταξύ τους.

Επίσης, η οντότητα ΠΑΡΑΓΓΕΛΙΑ μπορεί να έχει ως ιδιότητες (χαρακτηριστικά) τον κωδικό, τον αριθμό παραστατικού, την ημερομηνία, τον κωδικό πελάτη, το προϊόν κ.ά., με τη βοήθεια των οποίων μπορούμε να ξεχωρίσουμε τις παραγγελίες μεταξύ τους. Στο παράδειγμα της εταιρείας, μπορούμε να ορίσουμε έναν τύπο οντότητας για τους υπαλλήλους της εταιρείας (EMPLOYEE), έναν τύπο οντότητας για τα τμήματα που έχει η εταιρεία (DEPARTMENT) και έναν τύπο οντότητας για τα έργα που έχει αναλάβει η εταιρεία (PROJECT). Καθένας από τους παραπάνω τύπους οντοτήτων περιγράφεται από ένα όνομα και από το σύνολο των πεδίων που περιέχει. Οι πληροφορίες αυτές αποτελούν το *σχήμα (schema)* της οντότητας.

### Τα Στιγμιότυπα (Snapshots)

Το κάθε διαφορετικό (αυτόνομο) στοιχείο μιας οντότητας αποκαλείται *στιγμιότυπο (snapshot)* ή και *εμφάνιση* της οντότητας. Για παράδειγμα, στην οντότητα ΠΕΛΑΤΗΣ, άλλο στιγμιότυπο είναι ο πελάτης με επώνυμο Παπαδόπουλος και άλλο στιγμιότυπο είναι ο πελάτης με επώνυμο Σουμπάσης.

### Το Πρωτεύον Κλειδί (Primary Key)

*Πρωτεύον κλειδί* ή *πεδίο κλειδί (primary key)* μιας οντότητας καλείται εκείνη η ιδιότητα (ή ο συνδυασμός ιδιοτήτων) που έχει μοναδική τιμή για όλα τα στιγμιότυπα (εμφανίσεις) της οντότητας. Για παράδειγμα, στην οντότητα ΠΕΛΑΤΗΣ πρωτεύον κλειδί είναι ο κωδικός πελάτη, στην οντότητα ΠΑΡΑΓΓΕΛΙΑ πρωτεύον κλειδί μπορεί να είναι ο κωδικός παραγγελίας ή ο αριθμός παραστατικού κοκ.

Υπάρχουν περιπτώσεις όπου το πεδίο κλειδί ενός τύπου οντότητας μπορεί να μην είναι απλό αλλά σύνθετο, να αποτελείται δηλαδή από πολλά απλά πεδία και τότε η συνθήκη της μοναδικότητας για την τιμή του κλειδιού δεν εφαρμόζεται σε κάθε πεδίο του σύνθετου κλειδιού αλλά στο σύνολο του συνδυασμού αυτών των πεδίων.

### Οι Συσχετίσεις (Relationships)

Με τον όρο *συσχέτιση (relationship)* αναφερόμαστε στον τρόπο σύνδεσης (επικοινωνίας) δύο ξεχωριστών οντοτήτων, ώστε να μπορούμε να αντλούμε στοιχεία (πληροφορίες) από τον συνδυασμό τους.

Για παράδειγμα, η οντότητα ΓΙΑΤΡΟΣ συσχετίζεται με την οντότητα ΑΣΘΕΝΗΣ αλλά και με την οντότητα ΚΛΙΝΙΚΗ στη βάση δεδομένων ενός νοσοκομείου. Μπορούμε να δεχθούμε ότι ένας γιατρός παρακολουθεί (συσχετίζεται με) πολλούς ασθενείς, αλλά ένας ασθενής παρακολουθείται από (συσχετίζεται με) έναν μόνο γιατρό και επίσης ένας γιατρός συσχετίζεται με (ανήκει σε) μία μόνο κλινική, αλλά μια κλινική συσχετίζεται με (απασχολεί) πολλούς γιατρούς.

Στο παράδειγμα της εταιρείας, η οντότητα EMPLOYEE συσχετίζεται με την οντότητα DEPARTMENT και η οντότητα DEPARTMENT συσχετίζεται με την οντότητα PROJECTS. Ένας υπάλληλος ανήκει σ' ένα μόνο τμήμα και ένα τμήμα μπορεί να έχει πολλούς υπαλλήλους. Επίσης, ένα τμήμα εποπτεύει πολλά έργα αλλά ένα έργο εποπτεύεται από ένα μόνο τμήμα.

### **Το Ιεραρχικό Μοντέλο Βάσεων Δεδομένων**

Υπάρχουν τρία βασικά μοντέλα που έχουν επικρατήσει στις βάσεις δεδομένων, το ιεραρχικό, το δικτυωτό και το σχεσιακό, και τα οποία αναπτύχθηκαν με βάση αντίστοιχες δομές. Το ιεραρχικό μοντέλο (hierarchical) έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες.

Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο εκκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μία και μόνο μία οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίστηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

### **Το Δικτυωτό Μοντέλο Βάσεων Δεδομένων**

Και στο δικτυωτό (network) μοντέλο, τα στοιχεία τοποθετούνται σ' ένα επίπεδο ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σ' ένα κατώτερο ή σ' ένα ανώτερο επίπεδο.

### **Το Σχεσιακό Μοντέλο Βάσεων Δεδομένων**

Το σχεσιακό (relational) μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται σχεσιακές (relational databases). Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε με πολύ μεγάλες βάσεις δεδομένων.



Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα.

Ένας *πίνακας (table)* αποτελείται από γραμμές (rows) και στήλες (columns), όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται *χαρακτηριστικό (attribute)* ή *πεδίο (field)*, ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες (στήλες) που αφορούν ένα στοιχείο της οντότητας και αποκαλείται *πλειάδα (tuple)* ή *εγγραφή (record)*.

Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται *πεδίο ορισμού (domain)*.

Για να μπορέσουμε να κατανοήσουμε τις σχεσιακές βάσεις δεδομένων, ένα πολύ χαρακτηριστικό παράδειγμα αποτελεί ένας πίνακας πελατών και ένας πίνακας παραγγελιών μιας εμπορικής εταιρείας.

Τα πεδία που μπορούμε να ορίσουμε στους πίνακες αυτούς είναι τα εξής :

#### ΠΙΝΑΚΑΣ (ΟΝΤΟΤΗΤΑ) ΠΕΛΑΤΕΣ

(ΚωδικόςΠελάτη, Επώνυμο, Όνομα, Διεύθυνση, ΤΚ, Πόλη, ΑΦΜ, Υπόλοιπο)

#### ΠΙΝΑΚΑΣ (ΟΝΤΟΤΗΤΑ) ΠΑΡΑΓΓΕΛΙΕΣ

(ΚωδικόςΠελάτη, ΚωδικόςΠαραγγελίας, Ημερομηνία, Είδος, Ποσότητα, ΤιμήΜονάδας)

Βλέπουμε ότι οι δύο πίνακες έχουν ένα κοινό πεδίο (στήλη), τον ΚωδικόςΠελάτη, και αυτό είναι απαραίτητο στις σχεσιακές βάσεις δεδομένων για να μπορέσουμε να κάνουμε τη δουλειά μας και να συνδυάσουμε πληροφορίες και από τους δύο πίνακες.

Όπως είναι εύκολα κατανοητό, η βασικότερη εργασία που έχουμε να κάνουμε κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι να ορίσουμε τους πίνακες που θα χρησιμοποιήσουμε καθώς και τα πεδία που θα περιέχει ο καθένας απ' αυτούς. Η διαδικασία αυτή αποκαλείται κατασκευή του *σχήματος (schema)* μιας βάσης δεδομένων.

Οι κανόνες που πρέπει να ακολουθούμε πιστά κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι οι εξής :

- Η κάθε οντότητα πρέπει να παριστάνεται ως ένας ξεχωριστός πίνακας.
- Η κάθε στήλη του πίνακα αντιστοιχεί σε μια ιδιότητα της οντότητας.
- Η κάθε γραμμή του πίνακα αντιστοιχεί σε μια εμφάνιση της οντότητας.

- Η κάθε γραμμή πρέπει να είναι μοναδική, δηλ. αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν τα ίδια ακριβώς στοιχεία.
- Η σειρά εμφάνισης των γραμμών δεν έχει καμία σημασία.
- Η κάθε στήλη έχει μια δική της μοναδική ονομασία.
- Οι τιμές που ανήκουν στην ίδια στήλη πρέπει να είναι του ίδιου τύπου, δηλ. ή όλες αριθμοί ή όλες αλφαριθμητικές κοκ.
- Η στήλη που αποτελεί το πρωτεύον κλειδί (primary key) μιας οντότητας, δεν πρέπει να είναι ποτέ κενή (null).
- Αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν την ίδια τιμή στο πρωτεύον κλειδί.
- Το πρωτεύον κλειδί μιας οντότητας αποκαλείται ξένο κλειδί (foreign key) σε μια άλλη οντότητα, με την οποία υπάρχει συσχετισμός.
- Μπορεί να υπάρχουν πολλές γραμμές που να έχουν την ίδια τιμή στο ξένο κλειδί.

### Τα Σχεσιακά ΣΔΒΔ (RDBMS)

Τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ) ή RBMS (Relational DataBase Management Systems) αναπτύχθηκαν με βάση το σχεσιακό μοντέλο και έχουν επικρατήσει πλήρως στον χώρο. Κατά τον σχεδιασμό και τη δημιουργία μιας σχεσιακής βάσης δεδομένων, οι πίνακες αποτελούν το μοναδικό δομικό και απαραίτητο στοιχείο για μπορέσουν να αναπαρασταθούν οι πληροφορίες που περιέχονται στη βάση δεδομένων.

Για να μπορέσουμε να προσθέσουμε, διαγράψουμε ή τροποποιήσουμε τα στοιχεία που περιέχονται σε μια βάση δεδομένων, χρησιμοποιούμε ειδικές γλώσσες προγραμματισμού που αποκαλούνται *γλώσσες ερωταπαντήσεων (query languages)*. Η γλώσσα που αποτελεί σήμερα ένα διεθνές πρότυπο για την επικοινωνία των χρηστών με τα Σχεσιακά ΣΔΒΔ είναι η *SQL (Structured Query Language)* ή *Δομημένη Γλώσσα Ερωτημάτων*. Μπορεί να λειτουργήσει αυτόνομα αλλά και σε συνεργασία μ' άλλες γλώσσες προγραμματισμού.

Μια άλλη, φιλική προς τον χρήστη γλώσσα προγραμματισμού για να μπορούμε να υποβάλουμε ερωτήματα σε σχεσιακές βάσεις δεδομένων και να λαμβάνουμε απαντήσεις είναι η *QBE (Query By Example)*, η οποία χρησιμοποιεί φόρμες για τη γραφική απεικόνιση των ερωτημάτων μας.

Σήμερα, υπάρχουν εξελιγμένα εργαλεία διαχείρισης σε γραφικό και φιλικό προς τον χρήστη περιβάλλον για να κάνουμε τα εξής :

- Δημιουργία πινάκων
- Δημιουργία φορμών
- Δημιουργία ερωτημάτων
- Δημιουργία εκθέσεων (αναφορών)

Τα Σχεσιακά ΣΔΒΔ τα διακρίνουμε στα *μεγάλα*, τα οποία αφορούν κυρίως μεγάλους οργανισμούς και επιχειρήσεις, έχουν τεράστιο όγκο δεδομένων και πολλούς χρήστες ταυτόχρονα, και τέτοια συστήματα είναι τα Oracle, Ingres, Informix, SQL Server κ.ά.

και τα *μικρά*, τα οποία αφορούν κυρίως απλούς χρήστες, όπως είναι η Microsoft Access, η Paradox, η FoxPro κ.ά.

## **Το Μοντέλο Οντοτήτων–Συσχετίσεων**

Το μοντέλο που έχει επικρατήσει σήμερα για να παραστήσει τις έννοιες ή τη δομή μιας βάσης δεδομένων είναι το *Μοντέλο Οντοτήτων–Συσχετίσεων (ΟΣ)*. Οι βασικές (θεμελιώδεις) έννοιες του μοντέλου αυτού είναι οι εξής :

- Οντότητες
- Ιδιότητες ή Χαρακτηριστικά
- Συσχετίσεις

Για να αναπαραστήσουμε ένα Μοντέλο Οντοτήτων – Συσχετίσεων χρησιμοποιούμε ειδικά διαγράμματα, όπου τα ορθογώνια συμβολίζουν τις οντότητες, οι ρόμβοι τις συσχετίσεις και οι ελλείψεις τις ιδιότητες. Με ευθείες γραμμές συνδέουμε τις οντότητες που συσχετίζονται με κάποιο τρόπο μεταξύ τους. Όλα τα παραπάνω αποτελούν τη λογική δομή μιας βάσης δεδομένων, μια εργασία που είναι απαραίτητο να γίνει πριν από την καταχώριση και την επεξεργασία των στοιχείων (πληροφοριών) της βάσης δεδομένων.

Το μοντέλο οντοτήτων–συσχετίσεων αποτελεί μια γενική περιγραφή των γενικών στοιχείων που απαρτίζουν μια βάση δεδομένων και απεικονίζει την αντίληψη που έχουμε για τα δεδομένα (εννοιολογικό), χωρίς να υπεισέρχεται σε λεπτομέρειες υλοποίησης.

## **Οι Οντότητες**

Με τον όρο *οντότητα (entity)* αναφερόμαστε σε κάθε αντικείμενο, έννοια, πρόσωπο ή κατάσταση που έχει μια ανεξάρτητη ύπαρξη. Είναι κάτι που ξεχωρίζει και μπορούμε να συγκεντρώσουμε πληροφορίες (στοιχεία) γι' αυτό. Η οντότητα είναι αντίστοιχη με την έννοια της εγγραφής που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια του αντικειμένου στις σύγχρονες αντικειμενοστραφείς γλώσσες προγραμματισμού.

Παραδείγματα οντοτήτων είναι τα εξής : ΠΕΛΑΤΗΣ, ΠΑΡΑΓΓΕΛΙΑ, ΠΡΟΜΗΘΕΥΤΗΣ, ΑΠΟΘΗΚΗ, ΜΑΘΗΤΗΣ, ΚΑΘΗΓΗΤΗΣ, ΑΘΛΗΤΗΣ, ΑΓΩΝΙΣΜΑ, ΧΩΡΑ, ΠΟΛΕΙΣ κ.ά.

Μια βάση δεδομένων μπορεί να περιέχει πολλές διαφορετικές οντότητες, οι οποίες απεικονίζονται με ορθογώνια παραλληλόγραμμα και συσχετίζονται μεταξύ τους ανά δύο.

## **Οι Ιδιότητες (Χαρακτηριστικά) των Οντοτήτων**

Με τον όρο *ιδιότητες (properties)* ή *χαρακτηριστικά (attributes)* αναφερόμαστε στα συστατικά (δομικά) στοιχεία που προσδιορίζουν (αποτελούν) μια οντότητα. Η ιδιότητα είναι αντίστοιχη με την έννοια του πεδίου που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια της μεταβλητής στις γλώσσες προγραμματισμού.

Για παράδειγμα, η οντότητα ΓΙΑΤΡΟΣ μπορεί να αποτελείται από τις ιδιότητες (χαρακτηριστικά) ΑριθμόςΜητρώου, Επώνυμο, Όνομα, Πατρώνυμο, Ειδικότητα, Βαθμός, ΈτοςΓέννησης, Διεύθυνση, ΑΦΜ, Τηλέφωνο, Κινητό κ.ά., ενώ η οντότητα ΑΘΛΗΤΗΣ μπορεί να αποτελείται από τις ιδιότητες (χαρακτηριστικά) ΚωδικόςΑθλητή, Επώνυμο, Όνομα, Πατρώνυμο, Αγώνισμα, Επίδοση, Σύλλογος, ΈτοςΓέννησης, Διεύθυνση, Τηλέφωνο, Κινητό κ.ά.

Απ' όλες τις ιδιότητες μιας οντότητας, υπάρχει μία μόνο ιδιότητα, και σπανιότερα ένας συνδυασμός δύο ή και περισσότερων ιδιοτήτων, η τιμή της οποίας είναι μοναδική και προσδιορίζει την κάθε εμφάνιση (στιγμιότυπο) της οντότητας και αποκαλείται **πρωτεύον κλειδί (primary key)**. Το πρωτεύον κλειδί εμφανίζεται στα διαγράμματα με υπογράμμιση ή με έντονη γραφή ή έχει ως πρόθεμα τον χαρακτήρα #.

Στο διάγραμμα οντοτήτων–συσχετίσεων οι ιδιότητες απεικονίζονται με σχήματα ελλειπτικής μορφής, τα οποία ενώνονται με ευθείες γραμμές με την οντότητα στην οποία ανήκουν.

## Τα Κλειδιά

Όπως είδαμε και νωρίτερα, με τον όρο **κλειδί (key)** ή πιο σωστά **πρωτεύον κλειδί (primary key)** αναφερόμαστε σε μια ιδιότητα (πεδίο), ή σπανιότερα σ' ένα σύνολο ιδιοτήτων (πεδίων), η τιμή της οποίας είναι μοναδική σ' ολόκληρη την οντότητα (πίνακας). Στην πράξη, το πρωτεύον κλειδί έχει διαφορετική τιμή για κάθε εμφάνιση της οντότητας ή για κάθε γραμμή (εγγραφή) του πίνακα και ποτέ δεν μπορεί να έχει μηδενική (κενή) τιμή (null). Προσοχή, άλλο πράγμα είναι ο αριθμός 0 και άλλο πράγμα είναι η κενή τιμή (null), δηλ. η μη ύπαρξη τιμής.

Ο συνδυασμός δύο ή και περισσότερων ιδιοτήτων (πεδίων) για τη δημιουργία ενός πρωτεύοντος κλειδιού αποκαλείται **σύνθετο κλειδί**. Ένα παράδειγμα σύνθετου κλειδιού θα μπορούσε να είναι ο συνδυασμός των ιδιοτήτων Επώνυμο, Όνομα και Πατρώνυμο, εφόσον φυσικά είμαστε απολύτως βέβαιοι ότι δεν υπάρχουν δύο ή και περισσότερα άτομα με κοινές τιμές στις παραπάνω ιδιότητες.

**Ξένο κλειδί** αποκαλείται μια ιδιότητα (πεδίο) που είναι πρωτεύον κλειδί σε μια οντότητα (πίνακας) αλλά που υπάρχει και σε μια άλλη οντότητα (πίνακας) σαν απλή ιδιότητα. Τα ξένα κλειδιά είναι απαραίτητα για να μπορέσουμε να κάνουμε τις συσχετίσεις (συνδέσεις, επικοινωνίες) ανάμεσα στις οντότητες (πίνακες).

Για παράδειγμα, στην οντότητα ΣΥΛΛΟΓΟΣ, το πεδίο ΚωδικόςΣυλλόγου είναι πρωτεύον κλειδί, ενώ στην οντότητα ΑΘΛΗΤΗΣ, το πεδίο ΚωδικόςΣυλλόγου είναι ξένο κλειδί και είναι απαραίτητο για να μπορέσουμε να υλοποιήσουμε τη συσχέτιση ΑΝΗΚΕΙ, δηλ. να αντλήσουμε την πληροφορία ποιοι αθλητές ανήκουν σε ποιους συλλόγους. Προφανώς, στην οντότητα ΣΥΛΛΟΓΟΣ, το πεδίο Κωδικός Συλλόγου θα έχει μοναδικές τιμές, ενώ στην οντότητα ΑΘΛΗΤΗΣ, το πεδίο Κωδικός Συλλόγου θα έχει επαναλαμβανόμενες τιμές και αυτό γιατί πολλοί αθλητές θα ανήκουν στον ίδιο σύλλογο, αλλά ένας αθλητής ανήκει υποχρεωτικά σ' έναν και μόνο έναν σύλλογο.

Σ' ένα άλλο παράδειγμα, στην οντότητα ΓΙΑΤΡΟΣ, το πεδίο ΚωδικόςΓιατρού είναι πρωτεύον κλειδί, ενώ στην οντότητα ΑΣΘΕΝΗΣ, το πεδίο ΚωδικόςΓιατρού είναι

ξένο κλειδί και είναι απαραίτητο για να μπορέσουμε να υλοποιήσουμε τη συσχέτιση ΠΑΡΑΚΟΛΟΥΘΕΙΤΑΙ, δηλ. να αντλήσουμε την πληροφορία ποιοι ασθενείς παρακολουθούνται από ποιους γιατρούς. Προφανώς, στην οντότητα ΓΙΑΤΡΟΣ, το πεδίο ΚωδικόςΓιατρού θα έχει μοναδικές τιμές, ενώ στην οντότητα ΑΣΘΕΝΗΣ, το πεδίο ΚωδικόςΓιατρού θα έχει επαναλαμβανόμενες τιμές και αυτό γιατί πολλοί ασθενείς θα παρακολουθούνται από τον ίδιο γιατρό, αλλά ένας ασθενής παρακολουθείται μόνο από έναν γιατρό.

Αυτό αποτελεί βέβαια μια παραδοχή που κάνουμε για να μπορέσουμε να υλοποιήσουμε μια συσχέτιση σαν την παραπάνω σε μια βάση δεδομένων ενός Νοσοκομείου, αλλά μπορεί να θεωρήσει κάποιος ότι ένας ασθενής μπορεί να παρακολουθείται από πολλούς γιατρούς ταυτόχρονα, οπότε θα πρέπει να μεταβάλλουμε και τον τρόπο συσχέτισης των παραπάνω οντοτήτων.

### **Οι Συσχετίσεις Μεταξύ Οντοτήτων**

Ο σωστός σχεδιασμός και προσδιορισμός των οντοτήτων και των ιδιοτήτων τους αποτελούν τα θεμελιώδη βήματα για τη σωστή σχεδίαση και υλοποίηση μιας βάσης δεδομένων. Μια συσχέτιση συνδέει δύο ή και περισσότερες οντότητες μεταξύ τους και παριστάνεται στο διάγραμμα οντοτήτων–συσχετίσεων μ' έναν ρόμβο.

Οι συσχέτισεις είναι απαραίτητες για να μπορέσουμε να αντλήσουμε πληροφορίες που αφορούν δύο ή και περισσότερες οντότητες, όπως για παράδειγμα ποιοι πελάτες έκαναν παραγγελίες κάποια συγκεκριμένη χρονική περίοδο (συσχέτιση ΠΑΡΑΓΓΕΛΝΕΙ) ή ποιοι αθλητές ανήκουν σε ποιους συλλόγους (συσχέτιση ΑΝΗΚΕΙ) ή ποιοι αθλητές έλαβαν μέρος σε αγωνίσματα μια συγκεκριμένη χρονιά (συσχέτιση ΣΥΜΜΕΤΕΧΕΙ) κ.κ.

Όταν οι οντότητες που συμμετέχουν σε μια συσχέτιση είναι δύο, η συσχέτιση αποκαλείται *διμελής* ή *δυναδική*. Ο βαθμός μιας συσχέτισης είναι ίσος με το πλήθος των οντοτήτων που συμμετέχουν σ' αυτήν. Μια συσχέτιση μπορεί και η ίδια να έχει ιδιότητες που να περιγράφουν ορισμένα χαρακτηριστικά της, όπως για παράδειγμα η συσχέτιση ΠΑΡΑΓΓΕΛΙΑ ανάμεσα στις οντότητες ΠΕΛΑΤΗΣ και ΠΡΟΪΟΝ μπορεί να περιέχει τις ιδιότητες (πεδία) ΚωδικόςΠελάτη, ΚωδικόςΠροϊόντος, ΚωδικόςΠαραγγελίας, ΗμερομηνίαΠαραγγελίας, Ποσότητα κ.ά.

Στην περίπτωση αυτή το σωστό είναι να δημιουργήσουμε μια ακόμα οντότητα, την οντότητα ΠΑΡΑΓΓΕΛΙΑ, η οποία και θα περιέχει όλες τις παραπάνω ιδιότητες, και να μετονομάσουμε την προηγούμενη συσχέτιση από ΠΑΡΑΓΓΕΛΙΑ σε ΣΥΝΑΛΛΑΓΗ, που δεν θα περιέχει τώρα ιδιότητες. Έτσι, η παραπάνω συσχέτιση θα μετατραπεί από διμελή σε τριμελή.

Όταν σχεδιάζουμε μια βάση δεδομένων, θα πρέπει να εκχωρούμε ιδιότητες μόνο στις οντότητες και να έχουμε τις συσχέτισεις απλά και μόνο για να κατανοούμε τις λογικές συνδέσεις ανάμεσα στις οντότητες.

### **Οι Διμελείς Συσχετίσεις**

Οι διμελείς συσχέτισεις μεταξύ οντοτήτων είναι αυτές που θα μας απασχολήσουν ιδιαίτερα και υπάρχουν τρία βασικά είδη συνδέσεων σ' αυτές, τα εξής :

- *Ένα-προς-ένα (1:1)*, όπου μια εμφάνιση της μιας οντότητας συνδέεται με μία και μόνο μία εμφάνιση της άλλης οντότητας. Για παράδειγμα, η οντότητα ΣΥΛΛΟΓΟΣ έχει έναν μόνο προπονητή, ενώ η οντότητα ΠΡΟΠΟΝΗΤΗΣ συνδέεται μ' έναν και μόνο έναν σύλλογο. Σ' ένα άλλο παράδειγμα, η οντότητα ΝΟΜΟΣ έχει μία μόνο πόλη σαν πρωτεύουσα, ενώ η οντότητα ΠΡΩΤΕΥΟΥΣΑ αντιστοιχεί σ' έναν και μόνο έναν νομό. Στην περίπτωση των διμελών συσχετίσεων του τύπου ένα-προς-ένα, μπορούμε να ενώσουμε τα στοιχεία και των δύο ιδιοτήτων και να δημιουργήσουμε μια μοναδική οντότητα (πίνακα).
- *Ένα-προς-πολλά (1:M)*, όπου μια εμφάνιση της μιας οντότητας συνδέεται με πολλές εμφανίσεις της άλλης οντότητας αλλά κάθε εμφάνιση της δεύτερης οντότητας συνδέεται με μία και μόνο μία εμφάνιση της πρώτης οντότητας. Για παράδειγμα, ένας ΠΕΛΑΤΗΣ κάνει πολλές παραγγελίες, αλλά μια ΠΑΡΑΓΓΕΛΙΑ αντιστοιχεί σ' έναν και μόνο έναν πελάτη. Σ' ένα άλλο παράδειγμα, ένας ΣΥΛΛΟΓΟΣ έχει πολλούς αθλητές, αλλά ένας ΑΘΛΗΤΗΣ ανήκει σ' έναν και μόνο έναν σύλλογο. Οι διμελείς συσχετίσεις του τύπου ένα-προς-ένα είναι οι πιο συχνά συναντώμενες και οι πιο βολικές στη διαχείριση.
- *Πολλά-προς-πολλά (M:N)*, όπου σε μια εμφάνιση της μιας οντότητας αντιστοιχούν πολλές εμφανίσεις της άλλης οντότητας και σε κάθε εμφάνιση της δεύτερης οντότητας αντιστοιχούν πολλές εμφανίσεις της πρώτης οντότητας. Για παράδειγμα, ένας ΑΘΛΗΤΗΣ συμμετέχει σε πολλούς αγώνες αλλά και σ' έναν ΑΓΩΝΑ λαμβάνουν μέρος πολλοί αθλητές. Σ' ένα άλλο παράδειγμα, ένας ΚΑΘΗΓΗΤΗΣ διδάσκει σε πολλούς μαθητές αλλά και ένας ΜΑΘΗΤΗΣ διδάσκεται από πολλούς καθηγητές. Για να μπορέσουμε να διαχειριστούμε μια διμελή σχέση του τύπου πολλά-προς-πολλά, θα πρέπει να δημιουργήσουμε έναν τρίτο πίνακα που θα περιέχει δύο μόνο ιδιότητες (πεδία), δηλ. τα πεδία κλειδιά των δύο οντοτήτων, οπότε ο συνδυασμός τους θα είναι και το πεδίο κλειδί (σύνθετο κλειδί) του νέου πίνακα.

### **Το Διάγραμμα Οντοτήτων Συσχετίσεων**

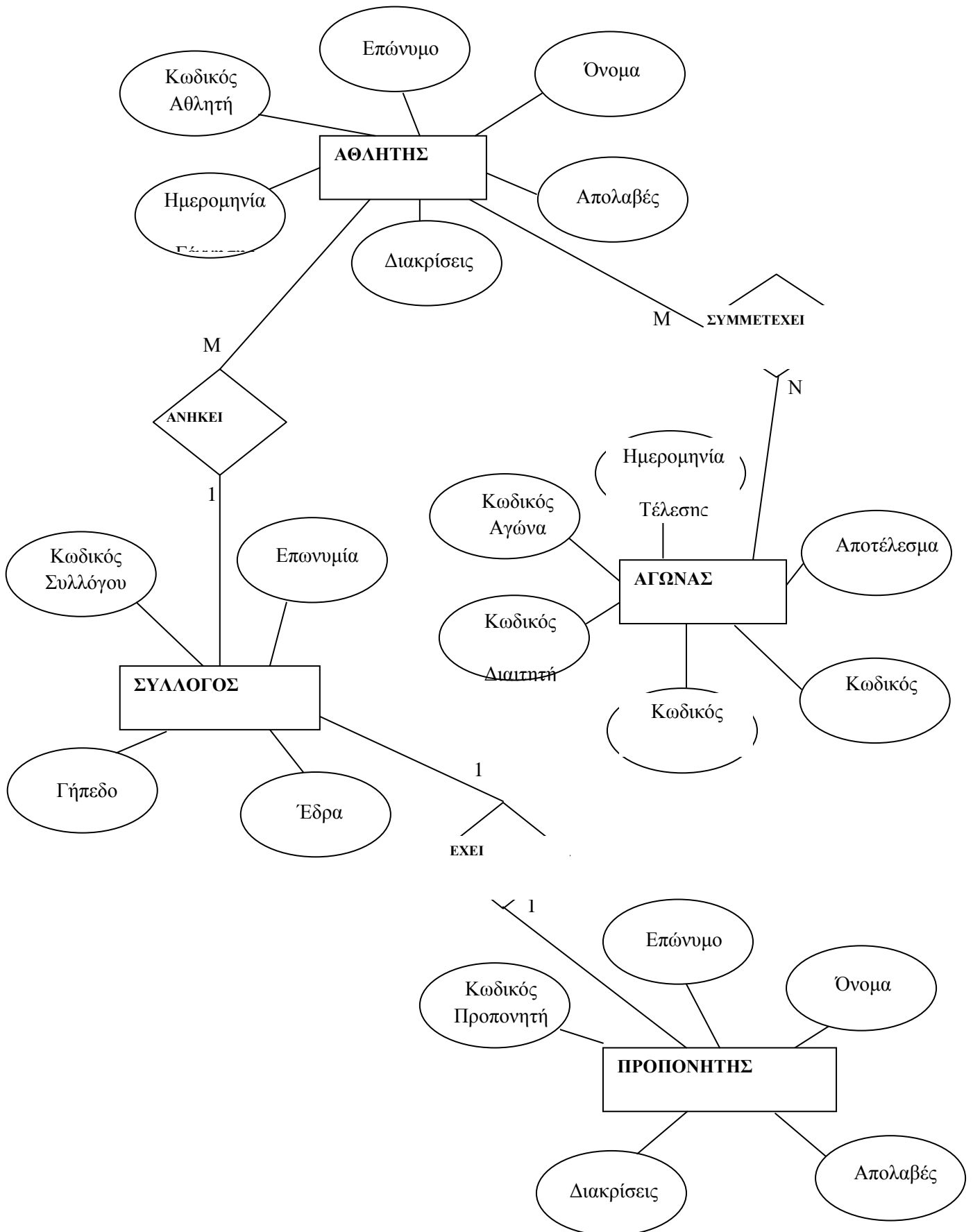
Για να μπορέσουμε να διαμορφώσουμε το διάγραμμα οντοτήτων συσχετίσεων, θα πρέπει να ακολουθήσουμε τα εξής βήματα :

- Να ορίσουμε τις οντότητες (πίνακες) που θα ανήκουν στη βάση δεδομένων που θέλουμε να δημιουργήσουμε.
- Να ορίσουμε τις ιδιότητες (πεδία) και τα πρωτεύοντα κλειδιά της κάθε οντότητας (πίνακα).
- Να ορίσουμε τις συσχετίσεις ανάμεσα στις οντότητες.
- Δημιουργούμε το διάγραμμα οντοτήτων συσχετίσεων, όπου θα απεικονίσουμε τις οντότητες, τις ιδιότητές τους και τις συσχετίσεις τους.

Θα δούμε το διάγραμμα οντοτήτων συσχετίσεων για μια βάση δεδομένων με ομάδες (συλλόγους) ποδοσφαίρου, όπου θα έχουμε τις οντότητες ΑΘΛΗΤΗΣ, ΣΥΛΛΟΓΟΣ, ΠΡΟΠΟΝΗΤΗΣ και ΑΓΩΝΑΣ. Οι συσχετίσεις ανάμεσα στις οντότητες αυτές θα είναι οι εξής :



- ΑΘΛΗΤΗΣ – ΣΥΛΛΟΓΟΣ : ένα-προς-πολλά (1:M)
- ΣΥΛΛΟΓΟΣ – ΠΡΟΠΟΝΗΤΗΣ : ένα-προς-ένα (1:1)
- ΑΘΛΗΤΗΣ – ΑΓΩΝΑΣ : πολλά-προς-πολλά (M:N)



## Λογικός Σχεδιασμός μιας Βάσης Δεδομένων

Αφού έχουμε δημιουργήσει το διάγραμμα οντοτήτων συσχετίσεων και έχουμε επιλέξει το σχεσιακό μοντέλο δεδομένων για την υλοποίηση της βάσης δεδομένων, ακολουθούμε τη διαδικασία της κανονικοποίησης και είμαστε έτοιμοι για την καταχώριση των στοιχείων της βάσης δεδομένων. Ανάλογα τώρα με το είδος της διμελούς συσχέτισης, διακρίνουμε τις εξής περιπτώσεις ως προς τον λογικό σχεδιασμό που θα πρέπει να ακολουθήσουμε :

Αν η συσχέτιση των δύο πινάκων είναι ένα-προς-ένα, τότε μπορούμε είτε να συνενώσουμε τους δύο πίνακες, με τις αντίστοιχες εγγραφές φυσικά, ή να προσθέσουμε το ένα από τα δύο πεδία κλειδιά σαν ξένο κλειδί στον άλλον πίνακα ή τέλος να δημιουργήσουμε έναν καινούργιο πίνακα με μόνα πεδία τα πεδία κλειδιά των δύο πινάκων (σύνθετο κλειδί). Η προτιμότερη διαδικασία είναι η πρώτη, δηλ. η συνένωση των δύο πινάκων σ' έναν ενιαίο πίνακα.

Αν η συσχέτιση των δύο πινάκων είναι ένα-προς-πολλά, τότε μπορούμε είτε να προσθέσουμε το ένα από τα δύο πεδία κλειδιά σαν ξένο κλειδί στον άλλον πίνακα ή να δημιουργήσουμε έναν καινούργιο πίνακα με μόνα πεδία τα πεδία κλειδιά των δύο πινάκων (σύνθετο κλειδί). Η προτιμότερη διαδικασία είναι η πρώτη, δηλ. η προσθήκη του ξένου κλειδιού στην πλευρά 'πολλά' της σχέσης.

Αν η συσχέτιση των δύο πινάκων είναι πολλά-προς-πολλά, τότε το μόνο που μπορούμε και πρέπει να κάνουμε είναι να δημιουργήσουμε έναν καινούργιο πίνακα με μόνα πεδία τα πεδία κλειδιά των δύο πινάκων (σύνθετο κλειδί), όπου το κάθε πεδίο κλειδί από μόνο του γίνεται ξένο κλειδί. Οι δύο αρχικοί πίνακες δεν μεταβάλλονται.

## Η Κανονικοποίηση

Τα προβλήματα που είναι πιθανό να παρουσιαστούν κατά τη διαδικασία της υλοποίησης του σχεδιασμού μιας βάσης δεδομένων είναι η περιττή (άσκοπη) επανάληψη πληροφοριών, που είναι γνωστή με τον όρο *redundancy*, καθώς και δυσκολίες στην ενημέρωση της βάσης δεδομένων. Τα παραπάνω προβλήματα είναι γνωστά ως *πλεονασμοί δεδομένων* και *ανωμαλίες ενημέρωσης* και για να αντιμετωπιστούν με επιτυχία, θα πρέπει να διασπάσουμε τις μεγάλες σχέσεις σε μικρότερες. Αυτό γίνεται με τη διαδικασία της κανονικοποίησης, έτσι ώστε η βάση δεδομένων να είναι έτοιμη για καταχώριση στοιχείων.

Η *κανονικοποίηση (normalization)* είναι μια τεχνική που ασχολείται με την ανάλυση των σχέσεων (συσχετίσεων) σε μια βάση δεδομένων, όπου κάνουμε μετατροπή των αρχικών μεγάλων σχέσεων σε μικρότερες.

## Πλεονασμός Δεδομένων και Ανωμαλίες Ενημέρωσης

Με τον όρο πλεονασμός δεδομένων (data redundancy) εννοούμε την άσκοπη επανάληψη στοιχείων (πληροφοριών). Τα προβλήματα που προκύπτουν από τον πλεονασμό δεδομένων είναι γνωστά με τον όρο ανωμαλίες ενημέρωσης (update anomalies). Για να μπορέσουμε να κατανοήσουμε τις παγίδες του πλεονασμού δεδομένων, θα δούμε ένα παράδειγμα με τους πίνακες ΠΕΛΑΤΗΣ και ΠΑΡΑΓΓΕΛΙΑ. Η σχέση μεταξύ τους είναι ένα-προς-πολλά, δηλ. ένας πελάτης μπορεί να κάνει πολλές παραγγελίες, αλλά μια παραγγελία γίνεται μόνο από έναν πελάτη.

Αν αποφασίσουμε να δημιουργήσουμε έναν μόνον πίνακα, όπου σε κάθε γραμμή (εγγραφή) του θα υπάρχουν όλα τα στοιχεία της παραγγελίας και δίπλα όλα τα στοιχεία του πελάτη που έχει κάνει την παραγγελία, τότε θα έχουμε πέσει στην παγίδα του πλεονασμού δεδομένων και αυτό γιατί τα στοιχεία του κάθε πελάτη θα επαναλαμβάνονται για κάθε παραγγελία που έχει κάνει.

**Ανωμαλία εισαγωγής** έχουμε στην περίπτωση που θελήσουμε να καταχωρήσουμε μια καινούργια παραγγελία, οπότε θα πρέπει να καταχωρήσουμε εκ νέου κι όλα τα στοιχεία του πελάτη που έκανε τη συγκεκριμένη παραγγελία, κάτι που είναι κουραστικό, χρονοβόρο και περιέχει τον κίνδυνο λαθών.

Ένα άλλο πρόβλημα με ανωμαλία εισαγωγής έχουμε στην περίπτωση που θελήσουμε να καταχωρήσουμε ένα καινούργιο πελάτη ο οποίος δεν έχει κάνει ακόμα καμία παραγγελία, οπότε η βάση δεδομένων δεν θα μας το επιτρέψει και αυτό γιατί δεν δέχεται κενή τιμή (null) για το πεδίο κλειδί Κωδικός Παραγγελίας.

**Ανωμαλία διαγραφής** έχουμε στην περίπτωση που θελήσουμε να διαγράψουμε μια παραγγελία που είναι η μοναδική ενός πελάτη, οπότε θα χάσουμε και όλες τις πληροφορίες του συγκεκριμένου πελάτη.

**Ανωμαλία τροποποίησης** έχουμε στην περίπτωση που θελήσουμε να αλλάξουμε κάποιο στοιχείο ενός πελάτη, όπως τη διεύθυνση ή το τηλέφωνό του, οπότε θα πρέπει να τροποποιήσουμε όλες τις εγγραφές του πίνακα όπου εμφανίζεται ο συγκεκριμένος πελάτης. Αν δεν κάνουμε την αλλαγή σ' όλες τις εγγραφές, τότε ο πελάτης θα εμφανίζεται να έχει δύο διευθύνσεις ή δύο τηλέφωνα κοκ. Μιλάμε τότε για **μη συνεπή (inconsistent)** βάση δεδομένων.

Ως γνωστόν, η λύση στο παραπάνω πρόβλημα είναι να δημιουργήσουμε έναν πίνακα με τα στοιχεία των πελατών και έναν ξεχωριστό πίνακα με τα στοιχεία των παραγγελιών, όπου θα υπάρχει και σαν πεδίο (ξένο κλειδί) ο Κωδικός Πελάτη.

## Η Διαδικασία της Κανονικοποίησης

Η μέθοδος της κανονικοποίησης βοηθάει στον λογικό σχεδιασμό μιας βάσης δεδομένων και είναι συμπληρωματική του μοντέλου οντοτήτων συσχετίσεων. Το κέρδος για μας είναι ότι δεν υπάρχουν προβλήματα συνέπειας, πλεονασμού και εγκυρότητας των πληροφοριών της βάσης δεδομένων. Ακολουθώντας τη διαδικασία της κανονικοποίησης κάνουμε συνεχείς διασπάσεις των πινάκων σε πιο απλές και συμπαγείς μορφές, με στόχο πάντα να αποφύγουμε τον πλεονασμό (επανάληψη) των δεδομένων. Αφαιρούμε πεδία από τις αρχικές μεγάλες σχέσεις και τα τοποθετούμε σε νέες σχέσεις έτσι ώστε να μπορούμε να έχουμε τις ίδιες πληροφορίες και με τις νέες σχέσεις.

Μπορούμε να πούμε ότι **κανονικοποίηση (normalization)** είναι η διαδικασία μετατροπής των δεδομένων κάποιων σχέσεων (πινάκων) σε πιο απλές και πιο σαφείς σχέσεις, χωρίς πλεονασμούς (επανάληψεις) των δεδομένων. Οι βασικές μορφές της κανονικοποίησης είναι τρεις, η πρώτη (1<sup>η</sup> NF), η δεύτερη (2<sup>η</sup> NF) και η τρίτη (3<sup>η</sup> NF). Θα ξεκινήσουμε τη μελέτη μας με μια βάση δεδομένων που θέλουμε να κατασκευάσουμε για τους αθλητές στο αγώνισμα των 100 μέτρων στίβου, οι οποίοι προέρχονται από διάφορους συλλόγους διαφόρων χωρών και όπου μας ενδιαφέρουν οι επιδόσεις τους σε διάφορους διεθνείς αγώνες.

Αρχικά, χωρίς καμία μελέτη, θα μπορούσε κάποιος να θεωρήσει ότι τα δεδομένα για τους αθλητές και τις επιδόσεις τους στους αγώνες θα ήταν κάπως έτσι :

Κωδικός_Αθλητή	Επώνυμο_Όνομα	Κωδικός_Συλλόγου	Ονομασία_Συλλόγου	Κωδικός_Αγώνα
100	Lewis Carl	200	America	01
				02
				03
101	Wells John	201	Rangers	01
				03
				04

Παρατηρούμε ότι τα πεδία **Κωδικός\_Αγώνα**, **Αγώνας** και **Επίδοση** έχουν περισσότερες από μία τιμές. Αυτό είναι αντίθετο με τις αρχές της σχεσιακής θεωρίας, γιατί κάθε σχέση του σχεσιακού μοντέλου θα πρέπει να έχει πεδία με μία και μοναδική τιμή σε κάθε σειρά (εγγραφή).

Δεν μπορούμε φυσικά να προσθέσουμε κι άλλα πεδία, δηλ. ένα πεδίο για κάθε αγώνα, γιατί στο σχεσιακό μοντέλο δεν μπορεί μια σχέση να έχει μεταβαλλόμενο αριθμό πεδίων και ούτε είμαστε σε θέση να γνωρίζουμε σε πόσους αγώνες πήρε μέρος ένας αθλητής.

Μια πρώτη λύση θα ήταν να μην έχουμε επαναλαμβανόμενες τιμές στην ίδια σειρά και να προσπαθήσουμε να δημιουργήσουμε έτσι έναν νέο πίνακα σε κάθε κελί του οποίου να περιέχεται μία μόνο τιμή, όπως φαίνεται στο παρακάτω σχήμα :

Κωδικός_Αθλητή	Επώνυμο_Όνομα	Κωδικός_Συλλόγου	Ονομασία_Συλλόγου	Κωδικός_Αγώνα	Αγώνας	Επίδοση	Κωδικός_Χώρας	Ονομασία_Χώρας
100	Lewis Carl	200	America	01	LA 2004	10.08	300	USA
100	Lewis Carl	200	America	02	MO 2003	10.04	300	USA
100	Lewis Carl	200	America	03	RO 2002	10.07	300	USA
101	Wells John	201	Rangers	01	LA 2004	10.04	301	Britain
101	Wells John	201	Rangers	03	RO 2002	10.05	301	Britain
101	Wells John	201	Rangers	04	GE 2001	10.03	301	Britain

Η σχέση ή ο πίνακας που επεξεργαζόμαστε βρίσκεται τώρα στην 1<sup>η</sup> κανονική μορφή (1<sup>η</sup> NF). Ο ορισμός λέει ότι μια σχέση (πίνακας) βρίσκεται στην **1<sup>η</sup> κανονική μορφή** όταν περιέχει σταθερό και όχι μεταβλητό αριθμό πεδίων (στηλών) και κάθε πεδίο της σχέσης δεν περιέχει επαναλαμβανόμενες τιμές. Επίσης, κάθε κελί της σχέσης (διασταύρωση γραμμής και στήλης) θα πρέπει να περιέχει μία μόνο τιμή. Τα μειονεκτήματα που βλέπουμε αμέσως ότι προκύπτουν από τη νέα μορφή που πήρε

ο πίνακας είναι ότι έχουμε τώρα περισσότερες γραμμές για να απεικονίσουμε τα ίδια ακριβώς δεδομένα και φυσικά έχουμε περιττή επανάληψη τιμών.

Αν είμαστε στην 1<sup>η</sup> κανονική μορφή, για να μπορέσουμε να προχωρήσουμε στην 2<sup>η</sup> και στην 3<sup>η</sup> κανονική μορφή, θα πρέπει να ορίσουμε πρώτα ένα πρωτεύον κλειδί, δηλ. ένα πεδίο ή έναν συνδυασμό από δύο ή περισσότερα πεδία (σύνθετο κλειδί) για να μπορούμε να προσδιορίζουμε μονοσήμαντα την κάθε γραμμή (εγγραφή ή και πλειάδα).

Στο παραπάνω παράδειγμα, παρατηρούμε ότι ο πιο κατάλληλος συνδυασμός πεδίων για να προσδιορίσει μονοσήμαντα την κάθε γραμμή είναι τα πεδία *Κωδικός\_Αθλητή* και *Κωδικός\_Αγώνα*. Τώρα, αν το κλειδί που έχουμε ορίσει είναι σύνθετο, δηλ. αποτελείται από δύο ή περισσότερα πεδία, θα πρέπει να συνεχίσουμε με την 2<sup>η</sup> κανονική μορφή (2<sup>η</sup> NF), αλλιώς θα πρέπει να συνεχίσουμε με την 3<sup>η</sup> κανονική μορφή (3<sup>η</sup> NF).

Συνεχίζοντας τώρα με την 2<sup>η</sup> κανονική μορφή, ψάχνουμε να βρούμε τα πεδία εκείνα που να συσχετίζονται με (αφορούν, εξαρτώνται από) ολόκληρο το σύνθετο κλειδί. Παίρνουμε τα πεδία που συγκροτούν το σύνθετο κλειδί και από τα πεδία αυτά δημιουργούμε έναν καινούργιο πίνακα.

Στον πίνακά μας, το μόνο πεδίο που έχει σχέση με τον συνδυασμό των πεδίων που συγκροτούν το σύνθετο κλειδί, δηλ. με τα πεδία *Κωδικός\_Αθλητή* και *Κωδικός\_Αγώνα*, είναι προφανώς το πεδίο *Επίδοση*. Δημιουργούμε τώρα τον παρακάτω πίνακα.

<i>Κωδικός_Αθλητή</i>	<i>Κωδικός_Αγώνα</i>	<i>Επίδοση</i>
100	01	10.08
100	02	10.04
100	03	10.07
101	01	10.04
101	03	10.05
101	04	10.03

Συνεχίζοντας με την 2<sup>η</sup> κανονική μορφή, προσπαθούμε τώρα να βρούμε ποια από τα υπόλοιπα πεδία του πίνακα εξαρτώνται από κάθε ξεχωριστό πεδίο του σύνθετου κλειδιού. Παίρνουμε αυτό το ξεχωριστό πεδίο ως πρωτεύον κλειδί και με τα πεδία



που έχουν σχέση μ' αυτό δημιουργούμε και από έναν καινούργιο πίνακα κάθε φορά.

Στο παράδειγμά μας, θα προκύψουν οι εξής δύο καινούργιοι πίνακες :

Κωδικός_Α αθλητή	Επώνυμο_ Όνομα	Κωδικός_Συ λλόγου	Ονομασία_Σ υλλόγου	Κωδικός_ Χώρας	Ονομασία_ Χώρας
100	Lewis Carl	200	America	300	USA
100	Lewis Carl	200	America	300	USA
100	Lewis Carl	200	America	300	USA
101	Wells John	201	Rangers	301	Britain
101	Wells John	201	Rangers	301	Britain
101	Wells John	201	Rangers	301	Britain

Κωδικός_Αγώνα	Αγώνας
01	LA 2004
02	MO 2003
03	RO 2002
01	LA 2004
03	RO 2002
04	GE 2001

Φυσικά, στον δεύτερο πίνακα που έχει σχέση με τους Αγώνες, θα μπορούσαμε να είχαμε συμπεριλάβει και πεδία που να αφορούν έναν συγκεκριμένο αγώνα, όπως Πόλη, Χώρα, Ημερομηνία, Διεθνής ή Φιλικός κ.ά. Βλέπουμε ότι έχουμε ήδη εξαλείψει ένα μεγάλο μέρος του πλεονασμού δεδομένων που είχαμε στον αρχικό πίνακα. Η σχέση που επεξεργαζόμαστε βρίσκεται τώρα στην 2<sup>η</sup> κανονική μορφή (2<sup>η</sup> NF). Ο ορισμός λέει ότι μια σχέση βρίσκεται στην **2<sup>η</sup> κανονική μορφή** όταν έχει προέλθει από σχέση της 1<sup>ης</sup> κανονικής μορφής και ακόμη τα πεδία που δεν ανήκουν στο κλειδί έχουν σχέση μόνο με το κλειδί.

Βλέπουμε, όμως, ότι υπάρχουν ακόμη πλεονασμοί δεδομένων, όπως συμβαίνει με τα πεδία *Όνομασία\_Συλλόγου* και *Όνομασία\_Χώρας* στον πίνακα που έχει ως πεδίο κλειδί τον *Κωδικό\_Αθλητή*. Θα πρέπει συνεπώς να προχωρήσουμε σε δύο ακόμη διασπάσεις του πίνακα αυτού για να αποφύγουμε αυτές τις επαναλήψεις τιμών.

Κωδικός_Αθλητή	Επώνυμο_Όνομα	Κωδικός_Συλλόγου
100	Lewis Carl	200
100	Lewis Carl	200
100	Lewis Carl	200
101	Wells John	201
101	Wells John	201
101	Wells John	201

Κωδικός_Συλλόγου	Όνομασία_Συλλόγου	Κωδικός_Χώρας
200	America	300
200	America	300
200	America	300
201	Rangers	301
201	Rangers	301
201	Rangers	301

Κωδικός_Χώρας	Όνομασία_Χώρας
300	USA
300	USA
300	USA
301	Britain
301	Britain
301	Britain

Έχουμε φθάσει σ' ένα σημείο που δεν χρειάζεται περαιτέρω διάσπαση των πινάκων καθώς στους πίνακες που έχουμε καταλήξει δεν υπάρχουν πεδία που να περιγράφουν κάτι που να έχει σχέση με κάποιο άλλο πεδίο εκτός από το πεδίο κλειδί. Η σχέση που επεξεργαζόμαστε βρίσκεται τώρα στην 3<sup>η</sup> κανονική μορφή (3<sup>η</sup>

NF). Ο ορισμός λέει ότι μια σχέση βρίσκεται στην *3<sup>η</sup> κανονική μορφή* όταν ικανοποιεί τις απαιτήσεις της 1<sup>ης</sup> και της 2<sup>ης</sup> κανονικής μορφής και ακόμη δεν υπάρχει κάποιο πεδίο στον πίνακα που να εξαρτάται από κάποιο άλλο πεδίο διαφορετικό του πρωτεύοντος κλειδιού.