

Κεφάλαιο 4

Λογικός Σχεδιασμός

Κανονικοποίηση

Στο κεφάλαιο αυτό παρουσιάζονται οι βασικοί κανόνες σχεδίασης της δομής των πινάκων στο σχεσιακό μοντέλο, και αναλύεται η τεχνική της κανονικοποίησης που είναι το βασικό εργαλείο αυτής της διαδικασίας σχεδίασης.

Στο προηγούμενο κεφάλαιο αναλύσαμε τη δομή του σχεσιακού μοντέλου βάσεων δεδομένων και τον τρόπο με τον οποίο τα δεδομένα μιας εφαρμογής αποθηκεύονται στους πίνακες της βάσης. Η βασική αρχή σχεδίασης αυτού του μοντέλου, είναι **η ομαδοποίηση των πεδίων των τύπων οντότητας, και η δημιουργία πινάκων για τους τύπους οντότητας και συσχέτισης** όπως αυτοί έχουν ορισθεί στο μοντέλο οντοτήτων συσχετίσεων.

Ωστόσο, ένα από τα βασικά ερωτήματα που προκύπτουν κατά την πραγματοποίηση αυτής της διαδικασίας, έχει να κάνει **με την επιλογή των πεδίων που θα ομαδοποιηθούν και θα σχηματίσουν ένα πίνακα**. Με ποιο κριτήριο θα γίνει αυτή η επιλογή? Πως είμαστε σίγουροι πως η ομαδοποίηση των πεδίων στους πίνακες έγινε με τον καλύτερο δυνατό τρόπο? Με άλλα λόγια, μήπως υπάρχει κάποιος καλύτερος συνδυασμός των πεδίων, ο οποίος θα μπορούσε να οδηγήσει σε μια πιο ευέλικτη και αξιόπιστη βάση δεδομένων?

Η διαδικασία της σωστής σχεδίασης της δομής των πινάκων της βάσης, έτσι ώστε να ικανοποιεί τις προδιαγραφές που έχουν τεθεί και ταυτόχρονα να είναι ευέλικτη και αποδοτική στη χρήση της ονομάζεται **λογικός ή εννοιολογικός σχεδιασμός (conceptual or logical design)**. Στις σελίδες που ακολουθούν θα παρουσιάσουμε τους βασικούς κανόνες που ακολουθούμε κατά τη λογική σχεδίαση της δομής μιας βάσης δεδομένων, και θα περιγράψουμε αναλυτικά την **κανονικοποίηση (normalization)** των πινάκων της βάσης που αποτελεί μια από τις πιο σημαντικές διαδικασίες αυτού του είδους.

ΟΙ ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΣΧΕΔΙΑΣΗΣ ΤΗΣ ΔΟΜΗΣ ΤΟΥ ΣΧΕΣΙΑΚΟΥ ΜΟΝΤΕΛΟΥ

Όπως έχει ήδη αναφερθεί, ένα από τα πρώτα βήματα που κάνουμε κατά τη φάση της σχεδίασης των πινάκων της βάσης είναι η **ταυτοποίηση των πεδίων που περιέχουν**, τα οποία στη συνέχεια **ομαδοποιούνται** για να σχηματίσουν αυτούς τους πίνακες. Αυτή η διαδικασία ομαδοποίησης, θα πρέπει να γίνεται έτσι ώστε τα πεδία ενός πίνακα, να συσχετίζονται μεταξύ τους και τελικά, ολόκληρη η δομή του πίνακα να αναδεικνύει κάποια λογική. **Όσο πιο ξεκάθαρη είναι αυτή η λογική, τόσο πιο εύκολη γίνεται τελικά και η σχεδίαση του σχήματος ολόκληρης της βάσης δεδομένων.**

Ας θεωρήσουμε ως παράδειγμα για μία ακόμη φορά, τη βάση δεδομένων της εταιρείας η δομή της οποίας παρουσιάζεται στο **σχήμα 10** (σελ 43). Θεωρώντας τον πίνακα **EMPLOYEE**, διαπιστώνουμε πως η **δομή του παρουσιάζει κάποια λογική**, διότι **τα πεδία που περιλαμβάνονται σε αυτόν, χαρακτηρίζονται από άμεση συσχέτιση μεταξύ τους**. Η κάθε μια από τις πλειάδες του πίνακα, αναπαριστά και κάποιο υπάλληλο της εταιρείας. Για κάθε υπάλληλο καταχωρούμε το όνομά του, τη διεύθυνσή του, την ημερομηνία γέννησής του, το φύλο του, και το μισθό του. Το πεδίο **DNO** είναι **ξένο κλειδί** και αναπαριστά τη συσχέτιση που υφίσταται ανάμεσα στους πίνακες **EMPLOYEE** και **DEPARTMENT**. Η συσχέτιση αυτή είναι ξεκάθαρη και πλήρως ορισμένη: **ένας υπάλληλος πρέπει υποχρεωτικά να ανήκει σε κάποιο τμήμα**. Με τον ίδιο τρόπο μπορούμε να διαπιστώσουμε και την πλήρη σαφήνεια που χαρακτηρίζει τη συσχέτιση των πινάκων **DEPARTMENT** και **PROJECT**: **ένα τμήμα μπορεί να αναλάβει πολλά έργα ταυτόχρονα**. Τα πεδία **MGRSSN** του πίνακα **DEPARTMENT** και **DNUM** του πίνακα **PROJECT** είναι **ξένα κλειδιά**, και συσχετίζουν **το κάθε τμήμα με τον MANAGER που το διευθύνει, και το κάθε PROJECT με το DEPARTMENT που έχει αναλάβει την υλοποίησή του**.

Εξετάζοντας τώρα τη λογική που αναδεικνύεται από τις σχέσεις **DEPT_LOCATIONS** και **WORKS_ON**, διαπιστώνουμε πως εδώ τα πράγματα είναι πιο πολύπλοκα. Κάθε πλειάδα στον πίνακα **DEPT_LOCATIONS** περιέχει ένα κωδικό τμήματος, και κάποια από τις τοποθεσίες στις οποίες στεγάζεται αυτό το τμήμα. Με τον ίδιο τρόπο, κάθε πλειάδα στον πίνακα **WORKS_ON**, περιέχει τον κωδικό **SSN** κάποιου **EMPLOYEE**, τον κωδικό κάποιου από τα **PROJECTS** στα οποία εργάζεται, και τον αριθμό των ωρών ανά εβδομάδα, που απασχολείται σε αυτό το έργο. Παρά όμως αυτό τον αυξημένο βαθμό πολυπλοκότητας, το νόημα που αναδεικνύεται από αυτούς τους δύο πίνακες, δεν είναι δύσκολο να αποκαλυφθεί: **ο πίνακας DEPT_LOCATIONS αναφέρεται σε ένα πεδίο πολλαπλής τιμής (multivalued attribute) ενώ ο πίνακας WORKS_ON υλοποιεί τη σχέση M:N που υφίσταται ανάμεσα στους πίνακες EMPLOYEE και PROJECT**. Σε γενικές γραμμές λοιπόν, μπορούμε να ισχυρισθούμε, πως το σχεσιακό σχήμα της βάσης δεδομένων της εταιρείας, χαρακτηρίζεται από καλή οργάνωση όσον αφορά την ευκολία ανάδειξης της λογικής που κρύβεται πίσω από αυτό.

Η τελευταία πρόταση της προηγούμενης παραγράφου αποτελεί ουσιαστικά και το βασικό κανόνα σχεδίασης της δομής των πινάκων της βάσης. Σύμφωνα με τον κανόνα αυτό, **θα πρέπει να οργανώνουμε τους πίνακες έτσι ώστε μια απλή ανάγνωση του σχεσιακού σχήματος να είναι αρκετή για να καταλάβουμε το είδος της**

πληροφορίας που θα καταχωρούμε σε κάθε περίπτωση. Αυτό τυπικά σημαίνει να δημιουργούμε ένα ξεχωριστό πίνακα για κάθε τύπο οντότητας και για κάθε τύπο συσχέτισης (αυτό ισχύει για συσχετίσεις με πολλαπλότητα M:N) και σε καμιά περίπτωση να μη συνδυάζουμε πεδία διαφορετικών τύπων οντότητας, στον ίδιο πίνακα. Τυπικά εάν ακολουθήσουμε ένα προς ένα, τα έξι βήματα απεικόνισης του μοντέλου οντοτήτων συσχετίσεων στο σχεσιακό μοντέλο, μπορούμε να διασφαλίσουμε ένα αξιόπιστο σχήμα βάσεως δεδομένων, όσον αφορά τη λογική του οργάνωσης.

Τι θα συμβεί όμως εάν δεν ακολουθήσουμε τον παραπάνω κανόνα και τοποθετήσουμε στον ίδιο πίνακα, πεδία που ανήκουν σε διαφορετικούς τύπους οντότητας? Το πιο σημαντικό από τα προβλήματα που θα παρουσιαστούν στην περίπτωση αυτή, είναι η εμφάνιση **περιττών δεδομένων (redundant data)** στους πίνακες της βάσης. Αυτό σημαίνει πως καταχωρούμε την ίδια πληροφορία περισσότερες από μια φορές στον ίδιο πίνακα, κάτι που έχει ως αποτέλεσμα **τόσο την άσκοπη σπατάλη αποθηκευτικού χώρου, όσο και τον κίνδυνο εμφάνισης ασυνεπών δεδομένων (inconsistent data)** στην περίπτωση κατά την οποία τροποποιήσουμε τα πεδία που καταχωρούνται περισσότερες από μια φορές.

Ας υποθέσουμε για παράδειγμα πως οι πληροφορίες που αφορούν τους υπαλλήλους της εταιρείας, τα έργα στα οποία απασχολούνται, και τα τμήματα στα οποία ανήκουν, δεν αποθηκεύονται στους πίνακες **EMPLOYEE**, **PROJECT** και **DEPARTMENT** σύμφωνα με το μοντέλο που έχουμε αναπτύξει, αλλά στους πίνακες **EMP_DEP** (ο οποίος συσχετίζει τον κάθε υπάλληλο με το τμήμα στο οποίο ανήκει) και **EMP_PROJ** (ο οποίος συσχετίζει τον κάθε υπάλληλο με το έργο στο οποίο απασχολείται). Οι δύο αυτοί πίνακες θα έχουν την ακόλουθη δομή :

EMP_DEP

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
-------	-----	-------	---------	---------	-------	---------

EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
-----	---------	-------	-------	-------	-----------

Εάν παρατηρήσουμε προσεκτικά τη δομή αυτών των δύο πινάκων, θα διαπιστώσουμε πως δεν παραβιάζει τον βασικό μας κανόνα σχεδίασης, διότι **τα πεδία που περιλαμβάνονται σε αυτούς τους πίνακες, συσχετίζονται μεταξύ τους**, και περιγράφουν με αρκετή σαφήνεια τη φύση της πληροφορίας που καταχωρούμε σε αυτούς. Παρά το γεγονός όμως αυτό οι δύο αυτοί πίνακες, **περιέχουν πεδία που ανήκουν σε διαφορετικούς τύπους οντότητας**. Ο πίνακας **EMP_DEP** περιέχει τα πεδία **ENAME**, **SSN**, **BDATE** και **ADDRESS** του τύπου οντότητας **EMPLOYEE** και τα πεδία **DNUMBER**, **DNAME** και **DMGRSSN** του τύπου οντότητας **DEPARTMENT**. Με τον ίδιο τρόπο, ο πίνακας **EMP_PROJ**, περιέχει τα πεδία **SSN** και **ENAME** του τύπου οντότητας **EMPLOYEE**, τα πεδία **PNUMBER**, **PNAME** και **PLOCATION** του τύπου οντότητας **PROJECT** και το πεδίο **HOURS** του τύπου συσχέτισης **WORKS_ON**.

Αυτή η λανθασμένη σχεδίαση των πινάκων, προκαλεί την καταχώρηση σε αυτούς περιττών δεδομένων, δηλαδή τιμών που καταχωρούνται περισσότερες από μια φορές. Για να γίνει το γεγονός αυτό πιο εύκολα αντιληπτό, σε επόμενη σελίδα παρουσιάζουμε ένα στιγμιότυπο αυτών των πινάκων με δεδομένα από τους πίνακες **EMPLOYEE**, **DEPARTMENT** και **PROJECT**. Σε κάθε πίνακα έχει προστεθεί και μια επιπλέον στήλη (**A/A**) με αύξουσα αρίθμηση. Αυτή η στήλη προφανώς δεν ανήκει στο μοντέλο της βάσης αλλά έχει προστεθεί προκειμένου να αναφερόμαστε στις εγγραφές των πινάκων με μεγαλύτερη ευκολία.

Από τα στιγμιότυπα των δύο πινάκων που παρουσιάζονται στη συνέχεια, γίνεται εύκολα αντιληπτός ο λόγος για τον οποίο η χρήση αυτής της εσφαλμένης δομής οδηγεί σε απώλεια αποθηκευτικού χώρου. Πράγματι, στον πίνακα **EMP_DEP** οι τιμές των πεδίων **DNUMBER**, **DNAME** και **DMGRSSN** επαναλαμβάνονται σε κάθε εγγραφή, σε αντίθεση με το σωστό σχεσιακό μοντέλο (**Σχήμα 10**) στο οποίο για κάθε τμήμα, υπάρχει μια και μοναδική εγγραφή στον πίνακα **DEPARTMENT** ενώ η μοναδική πληροφορία που επαναλαμβάνεται σε κάθε εγγραφή του πίνακα **EMPLOYEE**, είναι ο κωδικός του τμήματος στον οποίο ανήκει ο κάθε υπάλληλος της εταιρείας. Η κατάσταση είναι ακόμη χειρότερη στον πίνακα **EMP_PROJ** όπου επαναλαμβάνονται όχι μόνο οι πληροφορίες που αφορούν το **PROJECT** αλλά και οι πληροφορίες που αφορούν τον **EMPLOYEE**. Αντίθετα, στο σωστό σχεσιακό σχήμα, τα στοιχεία για κάθε **EMPLOYEE** και για κάθε **PROJECT** καταχωρούνται μόνο μια φορά στους ομώνυμους πίνακες, και η μοναδική πληροφορία που επαναλαμβάνεται στον πίνακα **WORKS_ON** είναι ο συνδυασμός των πρωτευόντων κλειδιών των δύο πινάκων.

Το δεύτερο και σοβαρότερο πρόβλημα που ανακύπτει από τη λανθασμένη σχεδίαση του σχεσιακού σχήματος μιας βάσης δεδομένων, είναι οι **ανωμαλίες** που προκύπτουν κατά τις διάφορες διαδικασίες **τροποποίησης** του περιεχομένου της βάσης (**update anomalies**). Όπως έχουμε αναφέρει στο προηγούμενο κεφάλαιο, αυτές οι διαδικασίες μπορεί να ανήκουν σε τρεις διαφορετικές κατηγορίες: **εισαγωγή νέων εγγραφών στους πίνακες της βάσης, διαγραφή υπαρχόντων εγγραφών, και τροποποίηση του περιεχομένου των εγγραφών των πινάκων της βάσης**. Αυτό σημαίνει πως αυτές οι ανωμαλίες που συσχετίζονται με την τροποποίηση του περιεχομένου της βάσης, μπορούν να εμφανιστούν και στις τρεις παραπάνω περιπτώσεις, όπως φαίνεται από τα παραδείγματα που ακολουθούν:

Ανωμαλίες που εμφανίζονται κατά τη διαδικασία της εισαγωγής νέων εγγραφών (insertion anomalies): έστω ότι θέλουμε να προσθέσουμε μια νέα εγγραφή στον πίνακα **EMP_DEP**. Αυτή η εγγραφή εκτός από τα χαρακτηριστικά του **EMPLOYEE**, θα πρέπει να περιλαμβάνει και τα χαρακτηριστικά του **DEPARTMENT** στο οποίο εργάζεται ο **EMPLOYEE**. Εάν ο **EMPLOYEE** δεν έχει ακόμη τοποθετηθεί σε κανένα τμήμα, τα πεδία της νέας εγγραφής που αφορούν το **DEPARTMENT**, θα λάβουν την τιμή **NULL**. Στην αντίθετη περίπτωση θα λάβουν τις τιμές εκείνες που αντιστοιχούν στο τμήμα στο οποίο τοποθετείται ο **EMPLOYEE**. Το πρόβλημά μας σε αυτή τη διαδικασία εισαγωγής της νέας εγγραφής, είναι πως θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί όσον αφορά τις τιμές του **DEPARTMENT** που θα χρησιμοποιήσουμε, διότι αυτές θα πρέπει να είναι κάθε φορά οι σωστές τιμές. Αντίθετα, στο σωστό σχεσιακό μοντέλο, δεν υπάρχει αυτό το πρόβλημα, διότι η διαδικασία καταχώρησης κάποιου υπαλλήλου, δεν έχει καμία

σχέση με τη διαδικασία καταχώρησης κάποιου τμήματος. Η κάθε εγγραφή καταχωρείται στο δικό της πίνακα, και κάθε φορά που καταχωρούμε ένα νέο **EMPLOYEE**, το μόνο που έχουμε να κάνουμε είναι να καταχωρήσουμε – εκτός από τα πεδία του – τον κωδικό του τμήματος στο οποίο πρόκειται να τοποθετηθεί.

Επιπλέον, χρησιμοποιώντας το σωστό σχεσιακό μοντέλο, δεν έχουμε πρόβλημα στο να καταχωρήσουμε στη βάση ένα **DEPARTMENT** που δεν έχει καθόλου **EMPLOYEES**. Αντίθετα η διαδικασία αυτή είναι δύσκολο να πραγματοποιηθεί στο εσφαλμένο σχεσιακό μοντέλο που χρησιμοποιεί τον πίνακα **EMP_DEP**. Στο μοντέλο αυτό για να είναι δυνατή μια τέτοια καταχώρηση, θα πρέπει η νέα εγγραφή να εμφανίζει τιμές **NULL** στα πεδία που αφορούν τον **EMPLOYEE** – αφού θέλουμε να καταχωρήσουμε ένα άδικο τμήμα. Ένα όμως από αυτά τα πεδία είναι και το πεδίο **SSN** που αποτελεί και το πρωτεύον κλειδί του πίνακα **EMP_DEP**, και το οποίο σύμφωνα με τον δεύτερο κανόνα ακεραιότητας δεν μπορεί να πάρει την τιμή **NULL**. Αλλά ακόμη και εάν ήταν δυνατή η προσθήκη αυτής της εγγραφής, το πρόβλημα παραμένει. Πράγματι, εάν αργότερα καταχωρήσουμε και τον **EMPLOYEE** που λείπει τοποθετώντας τον ταυτόχρονα σε κάποιο τμήμα, η πρώτη εγγραφή δεν θα είναι πλέον αναγκαία, και θα πρέπει να τη διαγράψουμε. Όλα αυτά τα προβλήματα δεν εμφανίζονται όταν χρησιμοποιούμε το σωστό σχεσιακό σχήμα, διότι εκεί, η καταχώρηση ενός τμήματος δεν έχει καμία απολύτως σχέση με την καταχώρηση ενός υπαλλήλου – **τα πεδία του κάθε τύπου οντότητας καταχωρούνται στο δικό τους ξεχωριστό πίνακα το καθένα.**

Ανωμαλίες που εμφανίζονται κατά τη διαδικασία της διαγραφής υπαρχόντων εγγραφών (deletion anomalies): το πρόβλημα που προκύπτει κατά τη διαγραφή των εγγραφών του πίνακα έχει να κάνει με το γεγονός πως **η κάθε εγγραφή περιέχει δεδομένα από δύο τύπους οντότητας ταυτόχρονα.** Στο παράδειγμά μας, τα στοιχεία για τα τμήματα της εταιρείας τοποθετούνται στον ίδιο πίνακα με τα στοιχεία των υπαλλήλων της εταιρείας. Αυτό σημαίνει πως αν διαγράψουμε όλους τους υπαλλήλους από τον πίνακα **EMP_DEP**, θα χάσουμε και όλα τα στοιχεία που αφορούν τα τμήματα της εταιρείας – κάτι που βέβαια δεν είναι επιθυμητό. Αντίθετα, στο σωστό σχεσιακό σχήμα, τα στοιχεία των διαφόρων τύπων οντότητας της εφαρμογής βρίσκονται καταχωρημένα σε διαφορετικούς πίνακες, και η διαγραφή των εγγραφών του ενός πίνακα, αφήνει ανεπηρέαστα τα περιεχόμενα των υπολοίπων πινάκων.

Ανωμαλίες που εμφανίζονται κατά τη διαδικασία της τροποποίησης των περιεχομένων των εγγραφών του πίνακα (modification anomalies): επειδή η εσφαλμένη δομή του πίνακα **EMP_DEP** προκαλεί την εμφάνιση επαναλαμβανόμενων πεδίων, είναι προφανές, πως εάν η τιμή κάποιου από αυτά τα πεδία τροποποιηθεί, αυτή η αλλαγή θα πρέπει να μεταφερθεί και σε όλα τα στιγμιότυπα αυτού του πεδίου. Για παράδειγμα, εάν αλλάξουμε τον κωδικό **SSN** του **MANAGER** του τμήματος με κωδικό αριθμό **5**, θα πρέπει να αλλάξουμε αυτή την τιμή σε όλους τους υπαλλήλους της εταιρείας που εργάζονται σε αυτό το τμήμα – στην αντίθετη περίπτωση η βάση μας θα χαρακτηρίζεται από την παρουσία ασυνεπών δεδομένων. Αντίθετα στο σωστό σχεσιακό σχήμα, αυτή η αλλαγή μπορεί να πραγματοποιηθεί χωρίς κανένα πρόβλημα.

EMP_DEP

A/A	ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
1	Smith, John B.	123456789	09-Jan-55	731 Fonders, Houston, TX	5	Research	333445555
2	Wong, Franklin T.	333445555	08-Dec-45	638 Voss, Houston, TX	5	Research	333445555
3	Zelaya, Alicia J.	999887777	19-Jul-58	3321 Castle, Spring, TX	4	Administration	987654321
4	Wallace, Jennifer S.	987654321	19-Jun-31	291 Berry, Bellaire, TX	4	Administration	987654321
5	Narayan, Ramesh K.	666884444	15-Sep-52	975 FireOak, Humble, TX	5	Research	333445555
6	English, Joyce A.	453453453	31-Jul-62	5631 Rice, Houston, TX	5	Research	333445555
7	Jabbar, Ahmad V.	987987987	29-Mar-59	980 Dallas, Houston, TX	4	Administration	987654321
8	Borg, James E.	888665555	10-Nov-27	450 Stone, Houston, TX	1	Headquarters	888665555

EMP_PROJ

A/A	SSN	PNUMBER	HOURS	ENAME	PNAME	PLLOCATION
1	123456789	1	32.5	Smith, John B.	ProductX	Bellaire
2	123456789	2	7.5	Smith, John B.	ProductY	Sugarland
3	666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
4	453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
5	453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
6	333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
7	333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
8	333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
9	333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
10	999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
11	999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
12	987987987	10	35.0	Jabbar, Ahmad V	Computerization	Stafford
13	987987987	30	5.0	Jabbar, Ahmad V	Newbenefits	Stafford
14	987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
15	987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
16	888665555	20	NULL	Borg, James E.	Reorganization	Houston

Επομένως, ένας δεύτερος κανόνας που θα πρέπει να ακολουθήσουμε κατά το στάδιο της σχεδίασης του σχεσιακού σχήματος μιας βάσης δεδομένων, είναι **η οργάνωση της δομής των πινάκων με τέτοιο τρόπο ώστε να μην εμφανίζονται στις εγγραφές των πινάκων της βάσης, επαναλαμβανόμενα πεδία.**

Σε μια πιο αυστηρή θεωρητική περιγραφή, μπορούμε να ορίσουμε και άλλους κανόνες σχεδίασης του σχεσιακού σχήματος μιας βάσης δεδομένων. Μιλώντας γενικά αυτοί οι κανόνες **απαγορεύουν την προσθήκη στους βασικούς πίνακες της βάσης, πεδίων, που να λαμβάνουν NULL τιμές**, ή εάν κάτι τέτοιο δεν μπορεί να αποφευχθεί, να ισχύει μόνο για ένα μικρό ποσοστό των εγγραφών του πίνακα, και όχι για την πλειοψηφία των εγγραφών. Περισσότερες λεπτομέρειες πάνω στο θέμα αυτό μπορούν να βρεθούν στη βιβλιογραφία.

ΣΥΝΑΡΤΗΣΙΑΚΕΣ ΕΞΑΡΤΗΣΕΙΣ

Η έννοια της **συναρτησιακής εξάρτησης (functional dependency)** κατέχει κεντρική θέση στη θεωρία των σχεσιακών βάσεων δεδομένων, και ορίζεται ως **μια συσχέτιση ανάμεσα στα πεδία ενός πίνακα**. Πιο συγκεκριμένα, **ένα πεδίο B κάποιου πίνακα θεωρείται συναρτησιακώς εξαρτώμενο από ένα πεδίο A του ίδιου πίνακα, όταν σε κάθε χρονική στιγμή, η τιμή του A καθορίζει μονοσήμαντα την τιμή του B**. Σε μια ισοδύναμη διατύπωση, μπορούμε να περιγράψουμε την παραπάνω συσχέτιση ανάμεσα στα πεδία A και B λέγοντας πως **το πεδίο A προσδιορίζει συναρτησιακά το πεδίο B**. Για αυτή τη συναρτησιακή εξάρτηση, χρησιμοποιούμε το συμβολισμό $A \rightarrow B$, ενώ ισοδύναμα μπορούμε να γράψουμε και $B = B(A)$. Είναι προφανές πως τα πεδία A και B μπορεί να είναι τόσο **απλά (atomic)** όσο και **σύνθετα (composite)** πεδία. Για παράδειγμα, στον πίνακα **WORKS_ON**, ο συνδυασμός των πεδίων {**ESSN, PNO**} – που αποτελεί και το πρωτεύον κλειδί του πίνακα – καθορίζει μονοσήμαντα την τιμή του πεδίου **HOURS**. Στην περίπτωση αυτή μπορούμε να γράψουμε $\{ESSN, PNO\} \rightarrow HOURS$.

Σε ένα οποιοδήποτε πίνακα μιας σχεσιακής βάσης δεδομένων, **όλα τα πεδία του πίνακα που δεν ανήκουν στο πρωτεύον κλειδί του, εξαρτώνται συναρτησιακά από τα πεδία του πρωτεύοντος κλειδιού**. Ένα παράδειγμα συναρτησιακής εξάρτησης είναι η $SSN \rightarrow LNAME$ ανάμεσα στα πεδία **SSN** και **LNAME** του πίνακα **EMPLOYEE**. Πράγματι, γνωρίζοντας τον κωδικό αριθμό ενός υπαλλήλου, προσδιορίζουμε μονοσήμαντα το επώνυμό του. Αντίθετα, η συσχέτιση $LNAME \rightarrow SSN$ δεν είναι έγκυρη, διότι είναι πιθανόν να υπάρχουν υπάλληλοι με το ίδιο επώνυμο, και οι οποίοι ασφαλώς, θα χαρακτηρίζονται από διαφορετικό κωδικό **SSN**. Μιλώντας γενικά, **η ισχύς της συσχέτισης $A \rightarrow B$ δεν προϋποθέτει και την ισχύ της συσχέτισης $B \rightarrow A$** .

Η συναρτησιακή εξάρτηση μιας ομάδας πεδίων από κάποια άλλη, μπορεί να είναι τόσο **ολική (full dependency)** όσο και **μερική (partial dependency)**. Δύο ομάδες πεδίων X και Y λέμε ότι **χαρακτηρίζονται από ολική εξάρτηση, όταν η αφαίρεση έστω και ενός από τα πεδία του X, καταργεί την εξάρτηση $X \rightarrow Y$** . Στην αντίθετη περίπτωση, όταν δηλαδή η εν λόγω εξάρτηση διατηρείται ακόμη και μετά την αφαίρεση του πεδίου από το σύνολο πεδίων X, τότε οι δύο ομάδες πεδίων χαρα-

κτηρίζονται από **μερική εξάρτηση**. Ας σημειωθεί πως σε όλες σχεδόν τις περιπτώσεις η ομάδα πεδίων **Y**, περιλαμβάνει συνήθως μόνο ένα πεδίο.

Ας θεωρήσουμε για παράδειγμα τον πίνακα **EMP_PROJ** του εσφαλμένου σχεσιακού σχήματος και τη συναρτησιακή εξάρτηση **{SSN, PNUMBER}→HOURS**. Αυτή η εξάρτηση είναι έγκυρη, διότι η γνώση του κωδικού **SSN** του υπαλλήλου και του κωδικού του έργου στο οποίο απασχολείται, ταυτοποιεί μονοσήμαντα το πλήθος των ωρών που εργάζεται ο υπάλληλος σε αυτό το έργο. Εάν τώρα αφαιρέσουμε το πεδίο **SSN**, η εξάρτηση **PNUMBER→HOURS**, δεν υφίσταται πλέον, διότι η γνώση του κωδικού του υπαλλήλου δεν αρκεί για να ταυτοποιήσουμε το πλήθος των ωρών που απασχολείται σε κάποιο έργο – απαιτείται και η γνώση του κωδικού αυτού του έργου. Με τον ίδιο τρόπο δεν υφίσταται ούτε η εξάρτηση **SSN→HOURS** που απομένει εάν αφαιρέσουμε το πεδίο **PNUMBER**. Εφ' όσον λοιπόν η εξάρτηση **{SSN, PNUMBER}→HOURS** καταργείται εάν αφαιρέσουμε έστω και ένα πεδίο από αυτή, τα πεδία **HOURS** και **{SSN, PNUMBER}**, είναι **ολικώς εξαρτημένα (fully dependent)**.

Αντίθετα, αν θεωρήσουμε την εξάρτηση **{SSN, PNUMBER}→ENAME** αυτή δεν καταργείται εάν αφαιρέσουμε το πεδίο **PNUMBER**. Πράγματι, η συναρτησιακή εξάρτηση **SSN→ENAME** που απομένει μετά την αφαίρεση του εν λόγω πεδίου, είναι έγκυρη, διότι η τιμή του κωδικού **SSN** για ένα υπάλληλο, καθορίζει μονοσήμαντα το όνομά του. Στην περίπτωση αυτή λέμε πως τα πεδία **{SSN, PNUMBER}** και **ENAME** είναι **μερικώς εξαρτημένα (partially dependent)**.

Δύο πεδία A και B χαρακτηρίζονται από έμμεση συναρτησιακή εξάρτηση όταν υπάρχει ένα τρίτο πεδίο C τέτοιο ώστε $A→C$ και $C→B$ ή ισοδύναμα όταν ισχύουν οι συσχετίσεις $C = C(A)$ και $B = B(C)$. Εάν επιπλέον το πεδίο **C** δεν ανήκει στα πεδία κλειδιά του πίνακα, τότε η εν λόγω εξάρτηση λέγεται **μεταβατική εξάρτηση (transitive dependency)**. Μια συναρτησιακή εξάρτηση μπορεί να είναι **μόνιμη ή προσωρινή**, ανάλογα με το χρονικό διάστημα για το οποίο ισχύει. Για παράδειγμα η συσχέτιση **SSN→LNAME** του πίνακα **EMPLOYEE** είναι μόνιμη διότι ο κωδικός **SSN** ενός υπαλλήλου δεν πρόκειται να αλλάξει ποτέ. Αντίθετα, η συσχέτιση **DNUMBER→MGRSSN** του πίνακα **DEPARTMENT** είναι προσωρινή, διότι είναι πιθανόν σε κάποια χρονική στιγμή να λάβει χώρα αντικατάσταση του **MANAGER** κάποιου τμήματος με κάποιο άλλο υπάλληλο της εταιρείας.

Ένα τελευταίο είδος εξάρτησης που θα μας απασχολήσει στο σημείο αυτό είναι η **συναρτησιακή εξάρτηση δύο πεδίων A και B εκ των οποίων το ένα πεδίο είναι πεδίο πολλαπλής τιμής (multivalued dependency)**. Στην περίπτωση αυτή η γνώση του **A** δεν καθορίζει μονοσήμαντα την τιμή του **B**, αλλά ένα σύνολο τιμών για το **B** που είναι γνωστό και καθορισμένο εκ των προτέρων. Μια τέτοια εξάρτηση θα είχαμε για παράδειγμα εάν στον πίνακα **EMP_DEP** προσθέταμε ως πεδίο και την τοποθεσία του τμήματος **DLOCATION**. Επειδή ένα τμήμα μπορεί να έχει γραφεία σε περισσότερες από μια τοποθεσίες, η γνώση του κωδικού **DNUMBER** για ένα τμήμα δεν προσδιορίζει μονοσήμαντα την τοποθεσία του τμήματος, αλλά συσχετίζεται με ένα μια ολόκληρη ομάδα τιμών.

Με βάση την έννοια της **συναρτησιακής εξάρτησης**, μπορούμε τώρα να περάσουμε στην περιγραφή της διαδικασίας κανονικοποίησης του σχεσιακού σχήματος μιας βάσης δεδομένων που επιτρέπει τη σωστή σχεδίαση του σχήματος της βάσης σύμφωνα με τα όσα έχουν αναφερθεί στις προηγούμενες σελίδες.

ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ

Η διαδικασία της **κανονικοποίησης (normalization process)** προτάθηκε από τον **Codd το 1970**, και η βασική της λειτουργία, είναι να **δέχεται ως είσοδο το σχεσιακό σχήμα μιας βάσης δεδομένων και να εφαρμόζει πάνω του μια σειρά από ελέγχους προκειμένου να διαπιστώσει εάν ανήκει ή όχι σε κάποια κανονική μορφή**. Ο **Codd** εισήγαγε τρεις κανονικές μορφές, **την πρώτη, δεύτερη και τρίτη κανονική μορφή** οι οποίες συσχετίζονται με τις συναρτησιακές εξαρτήσεις που υφίστανται ανάμεσα στα πεδία των πινάκων της βάσης, ενώ αργότερα προτάθηκαν άλλες δύο κανονικές μορφές – **η τέταρτη και η πέμπτη** – οι οποίες στηρίζονται σε άλλου είδους εξαρτήσεις (**multivalued dependencies και join dependencies**).

Σε μία πρώτη περιγραφή, η **κανονικοποίηση εφαρμόζεται πάνω σε ένα σχεσιακό σχήμα, και προσπαθεί να το μετασχηματίσει σε μια νέα μορφή, η οποία να είναι απαλλαγμένη από τις διάφορες ανωμαλίες εισαγωγής, διαγραφής και τροποποίησης εγγραφών που παρουσιάσαμε σε προηγούμενες σελίδες**. Πιο συγκεκριμένα, αυτή η μεθοδολογία **αναλύει τη δομή των πινάκων του σχεσιακού σχήματος με βάση τα πρωτεύοντα κλειδιά τους και τις συναρτησιακές εξαρτήσεις που υφίστανται ανάμεσα στα πεδία τους**, και, στη συνέχεια **ελέγχει αυτές τις δομές προκειμένου να διαπιστώσει εάν ικανοποιούν κάποιες συνθήκες**. Εάν οι εν λόγω συνθήκες δεν ικανοποιούνται, λαμβάνει χώρα **διάσπαση των πινάκων της βάσης σε μικρότερους πίνακες**, κάθε ένας εκ των οποίων, ικανοποιεί πλέον αυτές τις συνθήκες.

Μια κανονική μορφή ορίζεται ως η «κατάσταση» στην οποία βρίσκεται μια σχέση όσον αφορά τους τύπους των εξαρτήσεων που υφίστανται ανάμεσα στα πεδία της. Επειδή γενικά αυτές οι εξαρτήσεις οδηγούν σε προβληματική συμπεριφορά του σχήματος της βάσης – όπως είναι για παράδειγμα η εμφάνιση επαναλαμβανόμενων πεδίων – θα πρέπει να απομακρυνθούν από τους πίνακες. Για το λόγο αυτό η **κανονικοποίηση πραγματοποιείται σε μια ακολουθία βημάτων, με το κάθε βήμα να περιλαμβάνει την απομάκρυνση από τη δομή των πινάκων κάποιου τύπου συναρτησιακής εξάρτησης, διαδικασία, η οποία έχει ως αποτέλεσμα, τη μετάβαση του σχεσιακού σχήματος, από μια κανονική μορφή σε μια άλλη**. Πιο συγκεκριμένα, οι διαδικασίες που θα πρέπει να λάβουν χώρα για τη μετάβαση των σχεσιακών σχημάτων σε κάθε μια από τις πέντε κανονικές μορφές που έχουν οριστεί, είναι οι ακόλουθες :

Πρώτη Κανονική Μορφή (1st Normal Form, 1NF): Για να φέρουμε ένα πίνακα σε πρώτη κανονική μορφή, **θα πρέπει να απομακρύνουμε τις επαναλαμβανόμενες ομάδες πεδίων**, έτσι ώστε η τομή μιας γραμμής και μιας στήλης του πίνακα, να αντιστοιχεί πάντα σε μια απλή τιμή.

Δεύτερη Κανονική Μορφή (2nd Normal Form, 2NF): Για να φέρουμε ένα πίνακα σε δεύτερη κανονική μορφή, θα πρέπει πρώτα να τον φέρουμε σε πρώτη κα-

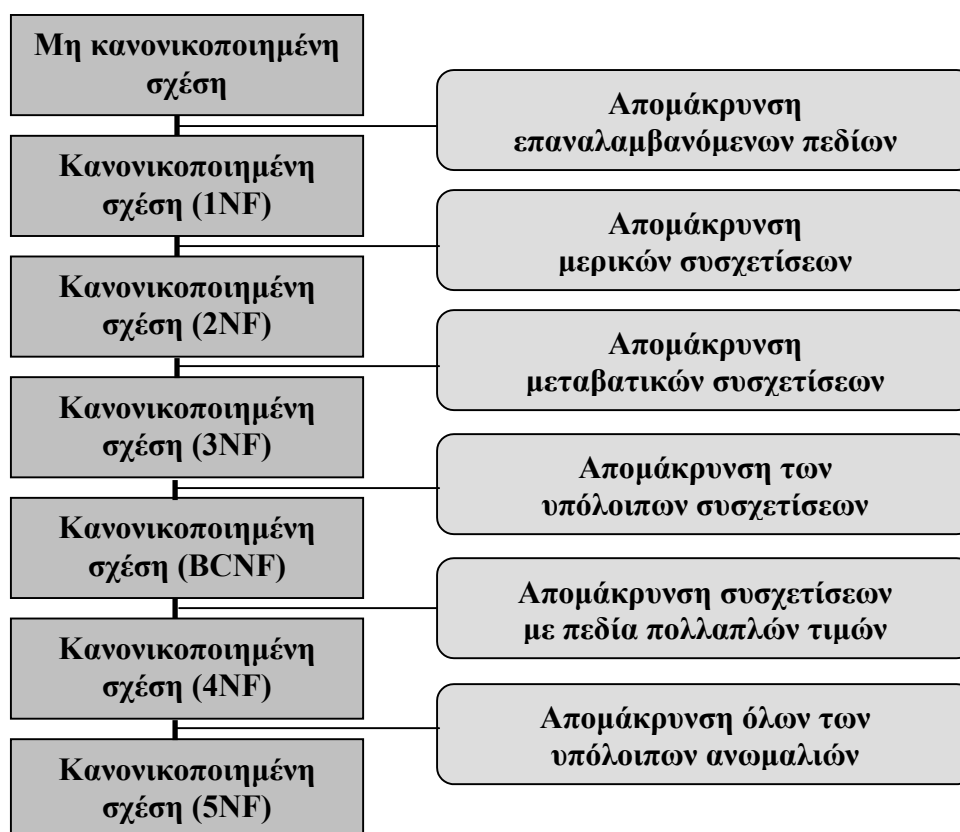
νονική μορφή, και στη συνέχεια να απομακρύνουμε όλες τις μερικές συναρτησιακές εξαρτήσεις (partial dependencies) που υφίστανται ανάμεσα στα πεδία του.

Τρίτη Κανονική Μορφή (3rd Normal Form, 3NF): Για να φέρουμε ένα πίνακα σε τρίτη κανονική μορφή, θα πρέπει πρώτα να τον φέρουμε σε δεύτερη κανονική μορφή, και στη συνέχεια να απομακρύνουμε όλες τις μεταβατικές συναρτησιακές εξαρτήσεις (transitive dependencies) που υφίστανται ανάμεσα στα πεδία του.

Τέταρτη Κανονική Μορφή (4th Normal Form, 4NF): Για να φέρουμε ένα πίνακα σε τέταρτη κανονική μορφή, θα πρέπει πρώτα να τον φέρουμε σε τρίτη κανονική μορφή και στη συνέχεια να απομακρύνουμε όλες τις συναρτησιακές εξαρτήσεις που περιλαμβάνουν πεδία πολλαπλών τιμών (multivalued dependencies).

Πέμπτη Κανονική Μορφή (5th Normal Form, 5NF) : Για να φέρουμε ένα πίνακα σε πέμπτη κανονική μορφή θα πρέπει πρώτα να τον φέρουμε σε τέταρτη κανονική μορφή, και στη συνέχεια να απομακρύνουμε όλες τις υπόλοιπες εξαρτήσεις που ενδεχομένως έχουν παραμείνει στη δομή του, μετά την εφαρμογή των παραπάνω διαδικασιών.

Εκτός από τις πέντε κανονικές μορφές που περιγράψαμε παραπάνω, μια επιπλέον κανονική μορφή που χρησιμοποιείται σε αρκετές περιπτώσεις, είναι η κανονική μορφή των **Boyce-Codd (Boyce Codd Normal Form, BCNF)**. Η BCNF, μπορεί να θεωρηθεί ως μια πιο αυστηρά διατυπωμένη 3NF. Αυτό σημαίνει πως ένας πίνακας που βρίσκεται σε BCNF βρίσκεται αυτόματα και σε 3NF – το αντίστροφο όμως δεν ισχύει. Η BCNF θα μελετηθεί αναλυτικά στις επόμενες σελίδες. Στο σχήμα που ακολουθεί αναπαρίστανται με γραφικό τρόπο, τα βήματα που θα πρέπει να ακολουθήσουμε προκειμένου να μετασχηματίσουμε μια μη κανονικοποιημένη σχέση από την πρώτη έως και την πέμπτη κανονική μορφή.



Σχήμα 11 : Τα έξι βήματα της κανονικοποίησης μιας δεδομένων

Μετά την παραπάνω συνοπτική περιγραφή των βασικών αρχών της κανονικοποίησης, ας περάσουμε τώρα στην αναλυτική παρουσίαση των πέντε κανονικών μορφών στις οποίες είναι δυνατό να βρεθούν τα σχήματα των πινάκων μιας σχεσιακής βάσης δεδομένων.

Πρώτη κανονική μορφή (1st Normal Form, 1NF)

Η πρώτη κανονική μορφή δημιουργήθηκε για να αποτρέψει την εμφάνιση στα πεδία ενός πίνακα σύνθετων τιμών, πολλαπλών τιμών καθώς και συνδυασμούς αυτών των δύο. Ένας πίνακας λέμε ότι βρίσκεται σε 1NF, όταν η τιμή του κάθε πεδίου σε κάθε πλειάδα, είναι ατομική δηλαδή δεν μπορεί να διασπαστεί σε μικρότερες μονάδες πληροφορίας. Εάν ένας πίνακας περιέχει πεδία πολλαπλών τιμών, τότε θα πρέπει να μετασχηματιστεί με τέτοιο τρόπο, ώστε σε κάθε κελί του να βρίσκεται μια απλή τιμή.

Για να κατανοήσουμε τη διαδικασία μετατροπής ενός πίνακα σε πρώτη κανονική μορφή, ας υποθέσουμε πως ο πίνακας **DEPARTMENT** δεν έχει τη δομή που έχουμε ορίσει στο Σχήμα 10, αλλά περιλαμβάνει τα ακόλουθα πεδία :

DEPARTMENT

DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS
-------	----------------	---------	------------

Σύμφωνα με το μοντέλο της εταιρείας, που παρουσιάσαμε στα προηγούμενα κεφάλαια, ένα τμήμα της μπορεί να έχει γραφεία σε περισσότερες από μια τοποθεσίες. Αυτό σημαίνει πως η στήλη **DLOCATIONS** θα περιέχει για κάθε τμήμα περισσότερες από μια τιμές όπως φαίνεται στο ακόλουθο παράδειγμα.

DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS
Research	5	333445555	Bellaire Sugarland Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Ο πίνακας **DEPARTMENT** στην παραπάνω μορφή, δεν ακολουθεί τον ορισμό της σχέσης που έχουμε δώσει για το σχεσιακό μοντέλο και θα πρέπει να μετασχηματιστεί. Υπάρχουν δύο είδη μετασχηματισμών που μπορούμε να εφαρμόσουμε. Ο ένας μετασχηματισμός είναι **να ορίσουμε για κάθε τιμή του πεδίου DLOCATIONS και μια ξεχωριστή πλειάδα** – στην περίπτωση αυτή το πρωτεύον κλειδί του πίνακα που θα προκύψει, θα είναι ο **συνδυασμός των πεδίων DNUMBER και DLOCATION**:

DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Μετά την εφαρμογή αυτού του μετασχηματισμού, ο πίνακας βρίσκεται σε **1NF**, διότι σε κάθε κελί του περιέχει μια απλή τιμή. Ωστόσο, χαρακτηρίζεται από την παρουσία **επαναλαμβανόμενων πεδίων**, κάτι που σημαίνει πως κατά τη διαχείριση του περιεχομένου του, θα εμφανιστούν οι ανωμαλίες εισαγωγής, διαγραφής και τροποποίησης εγγραφών που παρουσιάσαμε σε προηγούμενες σελίδες. Για το λόγο αυτό, μια δεύτερη εναλλακτική λύση που μπορούμε να εφαρμόσουμε, είναι **να απομακρύνουμε το πεδίο DLOCATIONS που δημιουργεί το πρόβλημα με τις πολλαπλές τιμές που περιέχει, και να το τοποθετήσουμε σε ένα άλλο πίνακα – στον πίνακα DEPT_LOCATIONS – μαζί με το πρωτεύον κλειδί του πίνακα DEPARTMENT**. Τα δύο αυτά πεδία θα αποτελούν και το πρωτεύον κλειδί του πίνακα **DEPT_LOCATIONS**.

DEPARTMENT

DNAME	<u>DNUMBER</u>	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

<u>DNUMBER</u>	DLOCATION
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston

Μετά την εφαρμογή αυτού του μετασχηματισμού, οι πίνακες **DEPARTMENT** και **DEPT_LOCATIONS** βρίσκονται σε **1NF**, και επιπλέον δεν περιέχουν τιμές που να επαναλαμβάνονται. Επομένως ανάμεσα στις δύο προσεγγίσεις που μπορούμε να ακολουθήσουμε για να φέρουμε τον πίνακα **DEPARTMENT** σε **1NF**, θα επιλέξουμε την τελευταία.

Είναι σημαντικό να αναφερθεί στο σημείο αυτό, πως τα πεδία πολλαπλών τιμών σε ένα πίνακα που επιβάλλουν το μετασχηματισμό του σε **1NF**, μπορεί να είναι τόσο **απλά**, όσο και **σύνθετα** όπως συμβαίνει για παράδειγμα στο ακόλουθο σχεσιακό σχήμα :

EMP_PROJ	SSN	ENAME	PROJECTS	
			PNUMBER	HOURS

Δεν είναι δύσκολο να διαπιστώσει κανείς, πως η **ύπαρξη σύνθετων πεδίων που ταυτόχρονα είναι και πεδία πολλαπλών τιμών ισοδυναμεί με την ύπαρξη ενός πίνακα μέσα σε ένα άλλο**. Αυτού του είδους οι σχέσεις λέγονται **ένθετες (nested relations)** και διατηρούν όλα τα χαρακτηριστικά μιας σχέσης. Στο παραπάνω παράδειγμα, για κάθε **EMPLOYEE** ορίζεται και ένας ξεχωριστός πίνακας που περιέχει τον κωδικό του έργου στο οποίο απασχολείται καθώς και τον αριθμό των ωρών που εργάζεται εβδομαδιαίως σε αυτό το έργο. Στον ένθετο αυτό πίνακα, που ορίζεται για κάθε υπάλληλο της εταιρείας, **δεν πρέπει να υπάρχουν εγγραφές που να φέρουν την ίδια τιμή για το πεδίο PNUMBER** – επομένως το πεδίο **PNUMBER**, θα αποτελεί **πρωτεύον κλειδί του ένθετου πίνακα**. Σύμφωνα με την ορολογία του σχεσιακού μοντέλου, τα κλειδιά αυτού του είδους, ονομάζονται **μερικά πρωτεύοντα κλειδιά (partial primary keys)**. Ένα στιγμιότυπο αυτού του σχεσιακού σχήματος, παρουσιάζεται στη συνέχεια.

SSN	ENAME	PROJECTS	
		PNUMBER	HOURS
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, Alicia J.	30	30.0
		10	10.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		10	15.0
888665555	Borg, James E	20	NULL

Για να μετασχηματίσουμε τον πίνακα αυτό σε 1NF, θα πρέπει να δημιουργήσουμε ένα νέο πίνακα, και να μεταφέρουμε σε αυτόν τόσο τα πεδία του ένθετου πίνακα (που είναι τα PNUMBER και HOURS) όσο και το πρωτεύον κλειδί του πίνακα EMP_PROJ (που είναι το SSN). Έτσι οι δύο νέοι πίνακες που θα προκύψουν θα είναι οι

EMP_PROJ1 SSN ENAME και

EMP_PROJ2 SSN PNUMBER HOURS με δεδομένα

SSN	ENAME
123456789	Smith, John B.
666884444	Narayan, Ramesh K.
453453453	English, Joyce A.
333445555	Wong, Franklin
999887777	Zelaya, Alicia J.
987987987	Jabbar, Ahmad V.
987654321	Wallace, Jennifer S.
888665555	Borg, James E

SSN	PNUMBER	HOURS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

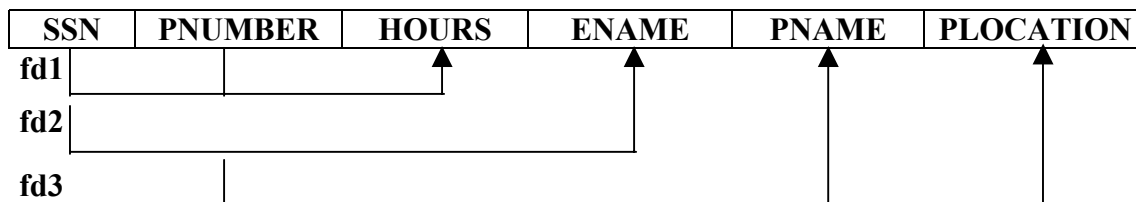
Όπως μπορεί εύκολα να διαπιστώσει κανείς ο πίνακας EMP_PROJ2, είναι στην πραγματικότητα ο πίνακας WORKS_ON

Δεύτερη κανονική μορφή (2nd Normal Form, 2NF)

Η δεύτερη κανονική μορφή προκύπτει από την πρώτη κανονική μορφή, εάν μετασχηματίσουμε τη δομή του πίνακα, με τέτοιο τρόπο ώστε να απομακρυνθούν όλες οι μερικές συναρτησιακές εξαρτήσεις (partial dependencies) που υφίστανται ανάμεσα στα πεδία του. Με άλλα λόγια ένας πίνακας βρίσκεται σε δεύτερη κανονική μορφή, όταν όλα τα πεδία που δεν ανήκουν στο πρωτεύον κλειδί του πίνακα, εξαρτώνται συναρτησιακά μόνο από τα πεδία του πρωτεύοντος κλειδιού, και μάλιστα, μέσω πλήρους συναρτησιακής εξάρτησης (full dependency). Επειδή ο ορισμός της πλήρους και της μερικής συναρτησιακής εξάρτησης μεταξύ δύο πεδίων X και Y εφαρμόζεται μόνο όταν το X περιλαμβάνει περισσότερα από ένα πεδία, είναι προφανές πως η αναγωγή ενός πίνακα στη δεύτερη κανονική μορφή, έχει νόημα μόνο όταν το πρωτεύον κλειδί του είναι σύνθετο. Αντίθετα, όταν το κλειδί του πίνακα είναι ένα απλό πεδίο, η αναγωγή του πίνακα σε 1NF, τον μετασχηματίζει αυτόματα και σε 2NF.

Ένα παράδειγμα ενός πίνακα που δεν είναι σε δεύτερη κανονική μορφή, είναι ο πίνακας EMP_PROJ που ορίσαμε στις προηγούμενες σελίδες. Μελετώντας τις τιμές που έχουμε καταχωρήσει σε αυτόν τον πίνακα, διαπιστώνουμε πως ο πίνακας

είναι σε 1NF, διότι, το κάθε κελί του περιέχει μια απλή τιμή. Ωστόσο δεν είναι σε 2NF διότι περιέχει μερικές συναρτησιακές εξαρτήσεις (partial dependencies). Για να κατανοήσουμε καλύτερα αυτό το παράδειγμα, στο ακόλουθο σχήμα παρουσιάζουμε τη δομή του πίνακα **EMP_PROJ** και τις συναρτησιακές εξαρτήσεις που υφίστανται ανάμεσα στα πεδία του.



Από το παραπάνω σχήμα διαπιστώνουμε πως η δομή του πίνακα **EMP_PROJ** περιέχει τρεις πλήρεις συναρτησιακές εξαρτήσεις, την **fd1** που ορίζεται ως **{SSN, PNUMBER} → HOURS**, την **fd2** που ορίζεται ως **SSN → ENAME**, και την **fd3** που ορίζεται ως **PNUMBER → {PNAME, PLOCATION}** – ο συμβολισμός που χρησιμοποιήθηκε στην **fd3** υποδηλώνει πως η γνώση του **PNUMBER** ταυτοποιεί μονοσήμαντα τα πεδία **PNAME** και **PLOCATION**. Επίσης, το πρωτεύον κλειδί του πίνακα, είναι σύνθετο, και αποτελείται από το συνδυασμό των πεδίων **SSN** και **PNUMBER**.

Ας θεωρήσουμε τώρα τη συναρτησιακή εξάρτηση **{SSN, PNUMBER} → ENAME**. Αυτή είναι μια μερική συναρτησιακή εξάρτηση, διότι εάν αφαιρέσουμε το πεδίο **PNUMBER** από αυτή, η εξάρτηση που απομένει και που είναι η **SSN → ENAME**, εξακολουθεί να ισχύει (στην πραγματικότητα είναι η **fd2**). Στην περίπτωση αυτή λέμε πως το πεδίο **ENAME** παραβιάζει τη συνθήκη της δεύτερης κανονικής μορφής. Το ίδιο ισχύει και για τις εξαρτήσεις **{SSN, PNUMBER} → PNAME** και **{SSN, PNUMBER} → PLOCATION** οι οποίες δεν καταργούνται εάν αφαιρέσουμε το πεδίο **SSN**, αφού εκφυλίζονται στη συναρτησιακή εξάρτηση **fd3**. Επομένως ο πίνακας **EMP_PROJ** στην τρέχουσα μορφή του, δε βρίσκεται σε 2NF.

Για να μετασχηματίσουμε τον πίνακα σε 2NF, θα πρέπει να τον διασπάσουμε σε μικρότερους πίνακες, με τέτοιο τρόπο, ώστε τα πεδία που δεν ανήκουν στο πρωτεύον κλειδί του, να τοποθετηθούν μαζί με τα πεδία του πρωτεύοντος κλειδιού, με τα οποία συσχετίζονται μέσω πλήρους συναρτησιακής εξάρτησης. Αυτό πρακτικά σημαίνει να δημιουργήσουμε τόσους πίνακες όσες είναι και οι πλήρεις συναρτησιακές εξαρτήσεις των πεδίων του πίνακα, και να τοποθετήσουμε σε αυτούς, τα πεδία που συμμετέχουν σε αυτές τις εξαρτήσεις. Έτσι, στο παράδειγμά μας, οι τρεις συναρτησιακές εξαρτήσεις **fd1**, **fd2** και **fd3**, θα οδηγήσουν στη δημιουργία των πινάκων **EP1 {SSN, PNUMBER, HOURS}**, **EP2 {SSN, ENAME}** και **EP3 {PNUMBER, PNAME, PLOCATION}**, κάθε ένας εκ των οποίων βρίσκεται σε 2NF.

Τρίτη κανονική μορφή (3rd Normal Form, 3NF)

Η τρίτη κανονική μορφή προκύπτει από τη δεύτερη κανονική μορφή **εάν μετασχηματίσουμε τον πίνακα με τέτοιο τρόπο ώστε να απομακρυνθούν όλες οι μεταβατικές συναρτησιακές εξαρτήσεις (transitive dependencies) που υφίστανται ανάμεσα στα πεδία του**. Επαναλαμβάνουμε από τη βασική θεωρία πως **μια συναρτησιακή εξάρτηση $A \rightarrow B$ ονομάζεται μεταβατική, όταν υπάρχει ένα πεδίο C που δεν ανήκει στο πρωτεύον κλειδί του πίνακα, τέτοιο ώστε $C = C(A)$ και $B = B(C)$** .

Ας θεωρήσουμε για παράδειγμα τον πίνακα **EMP_DEP** το σχήμα του οποίου απεικονίζεται στη συνέχεια, μαζί με τις συναρτησιακές εξαρτήσεις που ορίζονται ανάμεσα στα πεδία του.

ENAME	<u>SSN</u>	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
-------	------------	-------	---------	---------	-------	---------



Αυτός ο πίνακας, βρίσκεται σε **δεύτερη κανονική μορφή** καθώς η δομή του δεν χαρακτηρίζεται από μερικές συναρτησιακές εξαρτήσεις, όχι όμως και σε τρίτη κανονική μορφή, διότι ανάμεσα στα πεδία του υπάρχουν μεταβατικές εξαρτήσεις. Μια τέτοια εξάρτηση ορίζεται ανάμεσα στα πεδία **SSN** και **DMGRSSN**, καθώς ισχύουν οι σχέσεις **SSN $\rightarrow$ DNUMBER** και **DNUMBER $\rightarrow$ DMGRSSN** με το πεδίο **DNUMBER** να μην αποτελεί τμήμα του πρωτεύοντος κλειδιού του πίνακα. Μια δεύτερη μεταβατική εξάρτηση ορίζεται με τον ίδιο τρόπο ανάμεσα στα πεδία **SSN** και **DNAME**, καθώς μπορούμε να γράψουμε **SSN $\rightarrow$ DNUMBER** και **DNUMBER $\rightarrow$ DNAME**. Αυτού του είδους οι συναρτησιακές εξαρτήσεις θα πρέπει να απομακρυνθούν.

Η απομάκρυνση αυτών των εξαρτήσεων θα γίνει όπως και προηγουμένως **διασπώντας τον πίνακα EMP_DEP στους πίνακες ED1 {ENAME, SSN, BDATE, ADDRESS, DNUMBER} και ED2 {DNUMBER, DNAME, DMGRSSN}** κάθε ένας εκ των οποίων βρίσκεται τώρα σε **3NF**.

Κανονική μορφή των Boyce – Codd (Boyce Codd Normal Form, BCNF)

Η κανονική μορφή των **Boyce** και **Codd**, αποτελεί μια **πιο αυστηρή διατύπωση της τρίτης κανονικής μορφής**, και χρησιμοποιείται για να απομακρύνει ανωμαλίες που ενδέχεται να προκύψουν σε περιπτώσεις κατά τις οποίες **ένας πίνακας έχει περισσότερα από ένα υποψήφια κλειδιά, τα οποία μάλιστα είναι σύνθετα, αποτελούνται δηλαδή από πολλά πεδία**.

Για να κατανοήσουμε καλύτερα αυτή την κανονική μορφή ας πάρουμε ως παράδειγμα τη βάση δεδομένων ενός εκπαιδευτικού ιδρύματος. Ας υποθέσουμε πως για κάθε **μάθημα** που διδάσκεται σε αυτό το ίδρυμα, οι μαθητές εκτός από τις ώρες διδασκαλίας μπορούν να παρακολουθήσουν και κάποια **μαθήματα** πρόσθετης διδακτικής στήριξης τα οποία πραγματοποιούνται από κάποιο άλλο **καθηγητή (advisor)**. Όσοι μαθητές επιθυμούν να παρακολουθήσουν αυτό το πρόγραμμα, θα πρέπει να δηλώ-

σουν τα μαθήματα στη Γραμματεία, η οποία θα καταχωρεί τα στοιχεία των μαθητών, των καθηγητών, και των μαθημάτων, σύμφωνα με το ακόλουθο μοντέλο :

- Κάθε μαθητής μπορεί να δηλώσει περισσότερα από ένα μαθήματα
- Για κάθε μάθημα, ο κάθε μαθητής έχει ένα και μοναδικό καθηγητή
- Για κάθε μάθημα υπάρχουν πολλοί καθηγητές που το διδάσκουν
- Κάθε καθηγητής μπορεί να διδάξει ένα μόνο μάθημα
- Κάθε καθηγητής διδάσκει σε πολλούς μαθητές ταυτόχρονα.

Ας υποθέσουμε επίσης πως τα στοιχεία για αυτή την εφαρμογή καταχωρούνται στον πίνακα **CLASS** {**StudentId**, **Lesson**, **Advisor**} ένα στιγμιότυπο του οποίου παρουσιάζεται στη συνέχεια :

StudentId	Lesson	Advisor
123	Physics	Brown
123	Music	Jones
456	Biology	Stewart
789	Physics	Lenon
999	Physics	Brown

Μελετώντας προσεκτικά τη δομή του πίνακα και τις βασικές υποθέσεις του μοντέλου, μπορούμε να ταυτοποιήσουμε τις ακόλουθες συναρτησιακές εξαρτήσεις ανάμεσα στα πεδία του.

- **StudentId**→**Lesson** (multivalued dependency – ο κάθε μαθητής μπορεί να δηλώσει πολλά μαθήματα τα οποία μπορούμε να ανακτήσουμε γνωρίζοντας τον κωδικό του μαθητή)
- {**StudentId**, **Lesson**} → **Advisor**: η γνώση του κωδικού του μαθητή και κάποιου από τα μαθήματα που έχει δηλώσει επιτρέπει την ταυτοποίηση του καθηγητή που το διδάσκει
- **Lesson**→**Advisor** (multivalued dependency – το κάθε μάθημα διδάσκεται από πολλούς καθηγητές τους οποίους μπορούμε να ανακτήσουμε γνωρίζοντας το όνομα του μαθήματος)
- **Advisor**→**Lesson**: η γνώση του ονόματος του καθηγητή επιτρέπει την ταυτοποίηση του μαθήματος που διδάσκει.
- **Advisor**→**StudentId** (multivalued dependency – ένας καθηγητής μπορεί να διδάξει σε πολλούς μαθητές ταυτόχρονα τους οποίους μπορούμε να ανακτήσουμε εάν γνωρίζουμε το όνομα του καθηγητή).

Επιπλέον, δεν είναι δύσκολο να διαπιστώσουμε, πως ο εν λόγω πίνακας διαθέτει **δύο υποψήφια κλειδιά**, τα οποία μάλιστα είναι **σύνθετα πεδία**. Τα κλειδιά αυτά είναι ο συνδυασμός των πεδίων {**StudentId**, **Lesson**} που ταυτοποιεί μονοσήμαντα την τιμή του **Advisor**, και ο συνδυασμός των πεδίων {**StudentId**, **Advisor**} που ταυτοποιεί μονοσήμαντα την τιμή του **Lesson**. Στο παράδειγμά μας επιλέγουμε ως πρωτεύον κλειδί το συνδυασμό {**StudentId**, **Lesson**}.

Σε ποια κανονική μορφή βρίσκεται αυτός ο πίνακας? Παρατηρώντας προσεκτικά τη δομή του μπορούμε εύκολα να βγάλουμε το συμπέρασμα ότι **βρίσκεται σε**

τρίτη κανονική μορφή, αφού δεν περιέχει ούτε μερικές αλλά ούτε και μεταβατικές συναρτησιακές εξαρτήσεις. Το συμπέρασμα αυτό είναι σωστό, αλλά παρόλα αυτά, δε μας διασφαλίζει την απουσία των ανωμαλιών που παρουσιάζονται κατά την εισαγωγή, διαγραφή και τροποποίηση εγγραφών. Ας υποθέσουμε για παράδειγμα πως ο μαθητής με κωδικό **456** επιθυμεί να αλλάξει το μάθημα που δήλωσε, από **Βιολογία** σε **Μαθηματικά**. Εάν τροποποιήσουμε το περιεχόμενο της αντίστοιχης γραμμής του πίνακα, έτσι ώστε να καταγράψουμε την παραπάνω μεταβολή, **θα χάσουμε την πληροφορία πως ο καθηγητής Stewart διδάσκει Βιολογία**. Τα πράγματα είναι ακόμη χειρότερα, εάν ο εν λόγω μαθητής ακυρώσει εντελώς τη συμμετοχή του στο πρόγραμμα διδακτικής στήριξης. Στην περίπτωση αυτή, θα πρέπει να διαγράψουμε την αντίστοιχη γραμμή από τον πίνακα **CLASS**, αλλά αυτή η διαγραφή **θα οδηγήσει και σε πλήρη απώλεια όλων των δεδομένων που αφορούν τον καθηγητή Stewart όσον αφορά τα μαθήματα που διδάσκει στο πρόγραμμα πρόσθετης διδακτικής στήριξης**.

Οι παραπάνω ανωμαλίες δεν περιορίζονται μόνο στις διαδικασίες διαγραφής και τροποποίησης των εγγραφών του πίνακα, αλλά παρουσιάζονται και κατά το στάδιο της εισαγωγής νέων εγγραφών. Έστω πως σε κάποια χρονική στιγμή εκδηλώνεται ενδιαφέρον για την παρακολούθηση του μαθήματος **Computer Networks** υπεύθυνος του οποίου ορίζεται ο καθηγητής **Tanenbaum**. Η πληροφορία όμως αυτή θα καταχωρηθεί στον πίνακα **CLASS**, μόνο όταν κάποιος μαθητής προσέλθει στη Γραμματεία και δηλώσει αυτό το μάθημα – και αυτό, επειδή οι εγγραφές του πίνακα προϋποθέτουν και την καταχώρηση του κωδικού γι αυτόν το μαθητή. Το γεγονός δε, πως αυτός ο κωδικός αποτελεί μέρος του πρωτεύοντος κλειδιού, κάνει τα πράγματα ακόμη χειρότερα. Πράγματι, εάν ο εν λόγω κωδικός δεν ανήκε στο πρωτεύον κλειδί, η παραπάνω πληροφορία θα μπορούσε να καταχωρηθεί, προσθέτοντας την πλειάδα {**NULL, Computer Networks, Tanenbaum**}. Από τη στιγμή όμως που το πεδίο **StudentId** αποτελεί μέρος του πρωτεύοντος κλειδιού του πίνακα, η εν λόγω καταχώρηση δεν μπορεί να πραγματοποιηθεί, διότι θα λάβει χώρα **παραβίαση του δεύτερου κανόνα ακεραιότητας (entity integrity constraint)**, σύμφωνα με τον οποίο **κανένα από τα πεδία του πρωτεύοντος κλειδιού του πίνακα, δεν μπορεί να λάβει την τιμή NULL**.

Η εμφάνιση όλων αυτών των προβλημάτων, οφείλεται στο γεγονός, πως **ο πίνακας CLASS διαθέτει δύο υποψήφια σύνθετα κλειδιά, τα οποία έχουν ένα κοινό πεδίο** – το **StudentId**. Αυτή είναι μια κατάσταση, η οποία δεν παρατηρείται αρκετά συχνά, αλλά εν τούτοις, θα πρέπει να λαμβάνεται υπ όψιν κατά το στάδιο της σχεδίασης μιας βάσης δεδομένων. Οι **Boyce** και **Codd**, προκειμένου να άρουν την παραπάνω αδυναμία της τρίτης κανονικής μορφής, προχώρησαν στον ορισμό της ομώνυμης κανονικής μορφής ο οποίος διατυπώνεται ως εξής : «**Μια σχέση βρίσκεται στην κανονική μορφή των Boyce – Codd (BCNF), αν και μόνο αν κάθε πεδίο του πίνακα που ταυτοποιεί μονοσήμαντα κάποιο άλλο πεδίο, είναι υποψήφιο κλειδί του πίνακα**». Εφαρμόζοντας τον παραπάνω ορισμό στον πίνακα **CLASS**, διαπιστώνουμε πως αυτός **δεν βρίσκεται στην BCNF** – πράγματι το πεδίο **Advisor** παρά το γεγονός πως ταυτοποιεί μονοσήμαντα το πεδίο **Lesson** (σύμφωνα με τη συναρτησιακή εξάρτηση **Advisor→Lesson**), δεν αποτελεί τμήμα του υποψήφιου κλειδιού του πίνακα.

Για να φέρουμε τον πίνακα **CLASS** σε **BCNF**, θα πρέπει κατά τα γνωστά να τον διασπάσουμε σε μικρότερους πίνακες, κάθε ένας εκ των οποίων θα βρίσκεται σε **BCNF**. Σε αυτή τη διαδικασία της διάσπασης, **το πεδίο Advisor που ταυτοποιεί μονοσήμαντα κάποιο άλλο πεδίο αλλά παρόλα αυτά δεν ανήκει στο υποψήφιο κλειδί του πίνακα, θα τοποθετηθεί σε μία άλλη σχέση, και ταυτόχρονα θα γίνει το πρωτεύον κλειδί της**. Στην προκειμένη περίπτωση, η διάσπαση μπορεί να γίνει με δύο τρόπους : (α) **CLASS1 {StudentId, Advisor}** και **CLASS2 {Advisor, Lesson}** ή (β) **CLASS1 {StudentId, Lesson}** και **CLASS2 {Advisor, Lesson}** – και στις δύο περιπτώσεις, το πεδίο **Advisor** θα αποτελέσει πρωτεύον κλειδί του πίνακα **CLASS2**.

Τέταρτη κανονική μορφή (4th Normal Form, 4NF)

Μετασχηματίζοντας ένα πίνακα σε **BCNF**, είμαστε πλέον σίγουροι πως η δομή του δεν περιέχει ανεπιθύμητες συναρτησιακές εξαρτήσεις που θα μπορούσαν να οδηγήσουν στην εμφάνιση ανωμαλιών κατά την εισαγωγή, διαγραφή και τροποποίηση δεδομένων. Παρά το γεγονός όμως αυτό, τέτοιου είδους ανωμαλίες ενδέχεται να υπάρξουν **εάν υφίσταται κάποια συσχέτιση μεταξύ δύο πεδίων, εκ των οποίων το ένα να είναι πεδίο πολλαπλών τιμών (multivalued dependencies)**. Ας θεωρήσουμε ως παράδειγμα το πληροφοριακό σύστημα του σχολείου, και ας περιοριστούμε στη σχέση που καθορίζει για το κάθε μάθημα, τα ονόματα των καθηγητών που το διδάσκουν, καθώς και τους τίτλους των διδακτικών βιβλίων που χρησιμοποιούνται. Στην περιγραφή που ακολουθεί, υποθέτουμε ότι ισχύουν οι ακόλουθες παραδοχές :

- Κάθε μάθημα μπορεί να διδαχθεί από πολλούς καθηγητές
- Σε κάθε μάθημα μπορούν να χρησιμοποιηθούν περισσότερα από ένα βιβλία
- Για κάθε μάθημα χρησιμοποιούνται πάντα τα ίδια βιβλία, ανεξάρτητα από τον καθηγητή που το διδάσκει.

Από την παραπάνω περιγραφή καθίσταται προφανές πως εάν χρησιμοποιήσουμε για την καταχώρηση όλων αυτών των πληροφοριών τον πίνακα **OFFERING {COURSE, INSTRUCTOR, TEXTBOOK}**, υπάρχουν δύο συναρτησιακές εξαρτήσεις στις οποίες συμμετέχουν πεδία πολλαπλών τιμών. Οι εξαρτήσεις αυτές είναι οι **Course→Instructor** και **Course→Textbook**, με τα πεδία **Instructor** και **Textbook**, να μπορούν να λάβουν περισσότερες από μια τιμές, σε κάθε εξάρτηση. Στο παρακάτω σχήμα, παρουσιάζεται ο πίνακας **OFFERING** με ένα σύνολο ενδεικτικών εγγραφών στην μη κανονικοποιημένη και στην κανονικοποιημένη του μορφή.

COURSE	INSTRUCTOR	TEXTBOOK
Management	White Green Black	Drucker Peters
Finance	Gray	Weston Gilford

Μη κανονικοποιημένη μορφή του πίνακα OFFERING

<u>COURSE</u>	<u>INSTRUCTOR</u>	<u>TEXTBOOK</u>
Management	White	Drucker
Management	Green	Drucker
Management	Black	Drucker
Management	White	Peters
Management	Green	Peters
Management	Black	Peters
Finance	Gray	Weston
Finance	Gray	Gilford

Κανονικοποιημένη μορφή του πίνακα OFFERING

Ο πίνακας **OFFERING** στην κανονικοποιημένη του μορφή διαθέτει ένα σύνθετο κλειδί – το συνδυασμό των πεδίων **COURSE**, **INSTRUCTOR** και **TEXTBOOK** – και βρίσκεται σε **BCNF**, αφού όλα τα πεδία του που ταυτοποιούν μονοσήμαντα κάποια άλλα πεδία ανήκουν στο πεδίο κλειδί του πίνακα. Ωστόσο, όπως μπορεί εύκολα να διαπιστωθεί, ο πίνακας περιέχει **περιττά δεδομένα**, τα οποία προκαλούν όλες εκείνες τις ανωμαλίες που παρουσιάσαμε στις προηγούμενες παραγράφους. **Εάν για παράδειγμα θελήσουμε να προσθέσουμε ένα καινούριο βιβλίο στον παραπάνω πίνακα, θα πρέπει να καταχωρήσουμε τρεις εγγραφές, μια για κάθε εκπαιδευτικό, διότι στην αντίθετη περίπτωση θα φαίνεται πως αυτό το βιβλίο χρησιμοποιείται από ένα μόνο καθηγητή.**

Οι παραπάνω ανωμαλίες προκύπτουν όταν σε ένα πίνακα υπάρχουν τουλάχιστον τρία πεδία – ας τα ονομάσουμε **A**, **B** και **C** – τέτοια ώστε για κάθε τιμή του **A**, να υπάρχει ένα σύνολο τιμών για το **B**, και ένα σύνολο τιμών για το **C** και ταυτόχρονα το σύνολο τιμών του **B** είναι ανεξάρτητο από το σύνολο τιμών του **C** και αντίστροφα. Οι εξαρτήσεις αυτού του είδους ονομάζονται **εξαρτήσεις πολλών τιμών (multivalued dependencies)**.

Για να απομακρύνουμε αυτές τις ανωμαλίες, θα πρέπει κατά τα γνωστά να διασπάσουμε τον πίνακα σε δύο μικρότερους πίνακες διαχωρίζοντας τα πεδία **B** και **C** – σύμφωνα με τον ορισμό που δώσαμε προηγουμένως. Έτσι ο πίνακας **OFFERING**, θα διαχωριστεί στους πίνακες **TEACHER {COURSE, INSTRUCTOR}** και **TEXT {COURSE, TEXTBOOK}** κάθε ένας εκ των οποίων βρίσκεται σε **4NF**.

Πέμπτη κανονική μορφή (5th Normal Form, 5NF)

Τέλος η πέμπτη κανονική μορφή, χρησιμοποιείται όταν οι πίνακες που προκύπτουν από την τέταρτη κανονική μορφή, χαρακτηρίζονται από **εξαρτήσεις σύζευξης (join dependencies)**. Αυτού του είδους οι εξαρτήσεις **απαγορεύουν τη διάσπαση ενός πίνακα σε μικρότερους πίνακες οι οποίοι εάν συνδυαστούν εκ νέου να μπορούν να αναδημιουργήσουν τους αρχικούς πίνακες**. Περισσότερες λεπτομέρειες σχετικά με την πέμπτη κανονική μορφή, μπορούν να βρεθούν στη βιβλιογραφία.