

Κεφάλαιο 15

ΕΙΔΙΚΑ ΘΕΜΑΤΑ

15.1 Προστασία δεδομένων

15.2 Συμπίεση δεδομένων

15.3 Συντονισμός ταυτόχρονων προσπελάσεων

15.4 Επίλογος

Κεφάλαιο 15

ΕΙΔΙΚΑ ΘΕΜΑΤΑ

15.1 Εισαγωγή

Κατά τη φάση του φυσικού σχεδιασμού μία βάσης δεδομένων, ο διαχειριστής πρέπει να μεριμνήσει και για μία σειρά θεμάτων που σχετίζονται με την εύρυθμη και αποτελεσματική λειτουργία του συστήματος. Επίσης, πέραν του διαχειριστή το ίδιο το σύστημα απαρτίζεται από συντελεστές που έχουν τους ίδιους στόχους αναφορικά με τη λειτουργία του. Αντικείμενο, λοιπόν, αυτού του κεφαλαίου είναι η παρουσίαση ειδικών προβλημάτων που περισσότερο σχετίζονται με τη γενικότερη λειτουργία του συστήματος και λιγότερο με κάποια ειδική δομή. Τα προβλήματα αυτά αφορούν την προστασία και συμπύεση των δεδομένων και τον έλεγχο των ταυτόχρονων προσπελάσεων χρηστών.

15.2 Προστασία δεδομένων

Χωρίς ιδιαίτερα μέτρα προστασίας (security) από πιθανές βλάβες ή καταστροφές μπορεί να συμβούν μεγάλες ζημιές σε ένα υπολογιστικό σύστημα. Στο κεφάλαιο αυτό δεν τίγονται θέματα προστασίας των εγκαταστάσεων και του προσωπικού αλλά μέθοδοι προστασίας των δεδομένων από καταστροφές ή απώλειες, εκτός αν αυτό προκύψει σύμφωνα με την προδιαγεγραμμένη διαδικασία. Ειδικότερα θα μελετηθεί η ακρίβεια (accuracy) και η ακεραιότητα (integrity) των αρχείων. Τα δεδομένα ενός αρχείου θεωρούνται ακριβή αν

κατά την εισαγωγή τους δεν περιέχουν λάθη. Ο όρος ακεραιότητα των δεδομένων ισχύει για δεδομένα που έχουν ήδη αποθηκευθεί στη δευτερεύουσα μνήμη.

Προστασία των δεδομένων επιτυγχάνεται με διάφορους ελέγχους κατά την είσοδο και την επεξεργασία των δεδομένων χρησιμοποιώντας μία πληθώρα τεχνικών. Είναι γνωστή η δυνατότητα των λειτουργικών συστημάτων να πιστοποιούν την αυθεντικότητα (authentication) της ταυτότητας των χρηστών με τη μέθοδο των κλειδαρίθμων (passwords). Μάλιστα, το λειτουργικό σύστημα UNIX έχει τη δυνατότητα να δίνει ή να αρνείται στον κάθε χρήστη οποιαδήποτε δυνατότητα επέμβασης σε ένα συγκεκριμένο αρχείο με σκοπό την ανάγνωση, την ενημέρωση ή την εκτέλεση. Ανάλογη δυνατότητα υπάρχει και στα συστήματα διαχείρισης βάσεων δεδομένων.

Πρωταρχικός έλεγχος είναι η εξουσιοδότηση των χρηστών με συγκεκριμένα δικαιώματα χρήσης των αρχείων ενός συστήματος και επιτυγχάνεται με τη βοήθεια του πίνακα ελέγχου προσπέλασης (access control matrix). Μία γραμμή του πίνακα αυτού αντιστοιχεί σε κάθε χρήστη, οι στήλες του πίνακα αντιστοιχούν στα αρχεία του συστήματος, ενώ σε κάθε θέση του πίνακα εισάγονται τα δικαιώματα του χρήστη (δηλαδή, ανάγνωση, αποθήκευση, διαγραφή, αντιγραφή). Για παράδειγμα, η γλώσσα ερωταπαντήσεων SQL, που είναι μία πρότυπη γλώσσα για όλα τα σχεσιακά συστήματα, δίνει στο διαχειριστή του συστήματος τη δυνατότητα παραχώρησης και ανάκλησης δικαιωμάτων στο χρήστη με τη βοήθεια των εντολών ορισμού δεδομένων GRANT και REVOKE <privilege list>. Έτσι, λοιπόν, οι απαιτήσεις του χρήστη μέσα από το σύστημα διαχείρισης της βάσης δεδομένων μεταφράζονται σε απαιτήσεις προς το σύστημα αρχείων του λειτουργικού συστήματος. Στη συνέχεια οι ρουτίνες προσπέλασης συμβουλεύονται τον πίνακα ελέγχου προσπέλασης και επιστρέφουν στο σύστημα διαχείρισης της βάσης τα σχετικά δεδομένα.

Μία άλλη μέθοδος προστασίας των δεδομένων είναι ο μηχανισμός της όψης (view), που ορίζεται ως ένα αρχείο (πίνακας του σχεσιακού μοντέλου) αποτελούμενο από μερικά πεδία ενός ή περισσότερων αρχείων. Ας σημειωθεί ότι η όψη είναι ιδεατό αρχείο, δηλαδή δεν είναι στην πραγματικότητα αποθηκευμένη στη δευτερεύουσα μνήμη. Στη γλώσσα SQL η όψη υλοποιείται με την εντολή VIEW, ενώ και πάλι με τις εντολές GRANT και REVOKE δίνεται στο χρήστη η δυνατότητα επεξεργασίας μίας όψης. Η προστασία των δεδομένων προέρχεται από γεγονός ότι η όψη αποκρύπτει από το χρήστη μερικά πεδία, που μπορεί να είναι κρίσιμα σύμφωνα με τη

γνώμη του διαχειριστή του συστήματος.

Η ακρίβεια επιτυγχάνεται κατά την είσοδο των στοιχείων με έλεγχο των χαρακτήρων, έλεγχο των πεδίων και έλεγχο των εγγραφών που εισάγονται ομαδικά στο αρχείο. Οι χαρακτήρες μίας εγγραφής ελέγχονται αν είναι κενά, γράμματα, αριθμοί, πρόσημο ή τέλος αν είναι κάποιοι ειδικοί χαρακτήρες. Για παράδειγμα, η παλαιότερη DBase IV αναγνωρίζει τον αλφαριθμητικό τύπο, τον αριθμητικό, τους αριθμούς κινητής υποδιαστολής, τις ημερομηνίες, το λογικό τύπο και τον τύπο των σημειώσεων. Κάτι αντίστοιχο ισχύει σε όλα τα σύγχρονα εμπορικά συστήματα διαχείρισης βάσεων δεδομένων.

Ο έλεγχος των πεδίων είναι πιο ενδιαφέρων για τον προγραμματιστή και μπορεί να επισημάνει και λάθη λογικής εκτός από λάθη πληκτρολόγησης. Ο έλεγχος των πεδίων μπορεί να γίνει κατά πολλούς τρόπους.

Αρίθμηση συναλλαγών (transaction numbering). Μία τεχνική είναι να εισάγεται ένα σύνολο από εγγραφές ταξινομημένες κατά αύξουσα (ή φθίνουσα) τάξη του πρωτεύοντος κλειδιού της. Αν, λοιπόν, το κλειδί μίας εγγραφής δεν υπακούει στην αύξουσα ακολουθία, τότε απορρίπτεται. Επίσης απορρίπτονται οι εγγραφές με ίδιο κλειδί. Ακόμη πολλές φορές οι εγγραφές αριθμούνται και το σύνολό τους πρέπει να είναι σταθερό. Επομένως, αν το πλήθος των εισαγόμενων εγγραφών δεν ισούται με τον προκαθορισμένο αριθμό, τότε η συναλλαγή δεν συνεχίζεται.

Έλεγχος περιορισμών ορθότητας (integrity constrained checking). Μία άλλη τεχνική είναι να ελέγχεται το εύρος των τιμών των πεδίων. Για παράδειγμα, δεν είναι δυνατόν ένας υπάλληλος να έχει εργασθεί περισσότερο από 365 μέρες το χρόνο ή ο μισθός του να υπερβαίνει το 1.000.000 δραχμές το μήνα. Για το σκοπό αυτό, μάλιστα οι γλώσσες ερωταπαντήσεων των συστημάτων διαχείρισης βάσεων δεδομένων έχουν ειδικές εντολές που θέτουν περιορισμούς σε πεδία εγγραφής ενός μόνο αρχείου ή και σε πεδία εγγραφών περισσότερων αρχείων. Ένα παράδειγμα είναι η εντολή DEFINE INTEGRITY της INGRES: DEFINE INTEGRITY ON Student IS Student.mark < 11.

Έλεγχος πληρότητας . Σε εγγραφές με μεταβλητό αριθμό πεδίων πρέπει να γίνουν ιδιαίτεροι έλεγχοι για τη διαπίστωση ύπαρξης των εγγραφών αυτών.

Έλεγχος ημερομηνιών (date checking). Η ημερομηνία δίνεται υπό τον τύπο MMDDYY ή DDMMYY. Αυτό είναι ένας πρώτος έλεγχος. Ένας δεύτερος έλεγχος εξετάζει τις τιμές που λαμβάνουν οι χαρακτήρες που αντιστοιχούν στις ημέρες, τους μήνες και τα έτη. Για παράδειγμα, έτσι μπορούν να απορριφθούν χρονολογίες μεγαλύτερες από το 1990.

Έλεγχος συνέπειας (consistency checking). Πολλές φορές οι τιμές δύο πεδίων είναι σε άμεση συνάρτηση. Για παράδειγμα, έστωσαν τα πεδία 'Κέρδη' και 'Ζημίες'. Αν στο ένα πεδίο υπάρχει συγκεκριμένη τιμή, τότε στο άλλο πεδίο πρέπει να υπάρχει κενό.

Έλεγχος κωδικών (code checking). Οι κωδικοί αριθμοί συγκρίνονται με πίνακες κωδικών για τη διαπίστωση των ανύπαρκτων. Αν ο πίνακας των κωδικών είναι μεγάλος και η διαδικασία χρονοβόρα, τότε εξετάζονται μόνο τμήματα του κωδικού.

Έλεγχος ψηφίων (digit checking). Τα κυριότερα λάθη που γίνονται κατά την πληκτρολόγηση αριθμών είναι η παραδρομή, πχ. 1237 αντί 1234, η αντιμετάθεση, πχ. 1324 αντί 1234, η διπλή αντιμετάθεση, πχ. 4231 αντί 1234, και τέλος τα τυχαία λάθη, δηλαδή ένας συνδυασμός των προηγούμενων λαθών, πχ. 0214 ή 2124. Τα λάθη που ανήκουν στις τρεις πρώτες κατηγορίες λαθών μπορούν να ανιχνευθούν αν χρησιμοποιηθεί ένα επιπλέον ψηφίο ελέγχου με μία μέθοδο που αναλύεται στη συνέχεια.

Κάθε ψηφίο ενός αριθμού από τα δεξιά προς τα αριστερά πολλαπλασιάζεται με 2,3,...,11,2,3... αντίστοιχα και τα γινόμενα αθροίζονται. Υπ' όψη ότι πρέπει ο μεγαλύτερος αριθμός που χρησιμοποιείται να είναι ένας πρώτος αριθμός, όπως το 11. Έτσι, για τον αριθμό 1234 προκύπτει $1*5+2*4+3*3+4*2 = 5+8+9+8 = 30$. Το 30 διαιρούμενο δια 11 δίνει ακέραιο υπόλοιπο 8. Το 8 αφαιρούμενο από το 11 δίνει υπόλοιπο 3. Το ψηφίο αυτό θεωρείται ψηφίο ελέγχου και τελικά ο αριθμός αποθηκεύεται ως 12343. Αν και πάλι εκτελεσθούν οι προηγούμενες πράξεις και το ψηφίο ελέγχου πολλαπλασιασθεί με τη μονάδα και προστεθεί στο προηγούμενο άθροισμα, δηλαδή το 30, προκύπτει 33 που διαιρούμενο δια 11 δίνει υπόλοιπο 0. Ας σημειωθεί ότι εναλλακτικά, αντί της αλληλουχίας 2,3,..., μπορούν να χρησιμοποιηθούν ως βάρη είτε μία γεωμετρική ακολουθία 1,2,4,8 κλπ. είτε μία ακολουθία πρώτων αριθμών 1,3,5,7,11 κλπ.

Κατά την επεξεργασία γίνεται ο επανέλεγχος και μπορούν να ανιχνευθούν λάθη ενός ή δύο ψηφίων αν ο διαιρέτης είναι πρώτος αριθμός και

μάλιστα μεγαλύτερος από το πλήθος των ψηφίων του αριθμού. Έτσι, ο αριθμός που προκύπτει από το 12373 διαιρούμενος με το 11 δίνει υπόλοιπο 6, αυτός που προκύπτει από το 13243 δίνει υπόλοιπο 1, ενώ αυτός που προκύπτει από το 42313 δίνει υπόλοιπο 9.

Τα τυχαία λάθη δεν ανιχνεύονται πάντοτε. Η πιθανότητα μη ανίχνευσης τυχαίου λάθους είναι ίση προς τον αντίστροφο του διαιρέτη. Αυτό μπορεί να φανεί από τα επόμενα παραδείγματα. Από τον αριθμό 02143 προκύπτει $0*5+2*4+1*3+4*2+3*1 = 22$, που διαιρείται ακριβώς με το 11, αλλά από τον αριθμό 21243 προκύπτει $2*5+1*4+2*3+4*2+3*1 = 31$ που δεν διαιρείται ακριβώς με το 11.

As σημειωθεί ότι για την ανίχνευση και τη διόρθωση λαθών υπάρχει μία πληθώρα μεθόδων που ονομάζονται κώδικες διόρθωσης λαθών (error correcting codes) και αποτελούν αντικείμενο της Θεωρίας Πληροφοριών. Η πιο γνωστή μέθοδος είναι ο κώδικας Hamming, που αναφέρθηκε στο Κεφάλαιο 2 και χρησιμοποιείται στις μαγνητικές ταινίες και στα συστήματα RAID. Η λεπτομερής εξέταση των κωδίκων διόρθωσης λαθών είναι εκτός των ορίων του βιβλίου αυτού.

15.3 Συμπύεση δεδομένων

Όπως αναφέρθηκε στο Κεφάλαιο 1 εφαρμόζοντας μεθόδους κωδικοποίησης (coding) και συμπίεσης (compression) των δεδομένων επιτυγχάνεται οικονομία χώρου αποθήκευσης, χρόνου προσπέλασης και ταξινόμησης, καθώς και χρόνου ανάπτυξης των προγραμμάτων εφαρμογών. Στη συνέχεια δίνονται μερικές από τις περισσότερο χρησιμοποιούμενες τεχνικές στην πράξη.

Ένα πολύ χαρακτηριστικό παράδειγμα είναι η συμπίεση των ημερομηνιών. Έτσι, η αποθήκευση της ημερομηνίας '28 Οκτωβρίου 1940' απαιτεί 17 χαρακτήρες ως συμβολοσειρά, ενώ η αποθήκευσή της υπό τη μορφή '281040' απαιτεί 4 χαρακτήρες ως πραγματικός. Αν όμως θεωρηθεί ως αρχή η 1η Ιανουαρίου 1940, τότε η ημερομηνία αυτή μπορεί να αποθηκευθεί ως '301', δηλαδή 2 χαρακτήρες ως ακέραιος. Βέβαια, όπως είναι γνωστό, οι τεχνικές αυτές οδήγησαν στην εμφάνιση του προβλήματος του 2000 (Y2K problem). Το πρόβλημα αυτό πρόκυψε όχι από λάθος αλλά από ανάγκη. Αρκεί κάποιος να σκεφθεί ότι οι χωρητικότητες των μνημών και των δίσκων πριν, για παράδειγμα, από 20 χρόνια δεν είχαν τα τωρινά μεγέθη, και η εξοικονόμηση έστω και μερικών χαρακτήρων ήταν σημαντική.

Αρχικά δεδομένα:	0,14,0,0,0,15,0,0,0,3,3,0,0,0,0,0,0,423,0,*
Αναπαράσταση με συντεταγμένες:	2,14,6,15,10,3,11,3,18,423,20,*
Δυαδική αναπαράσταση:	01000100011000000101 και 14,15,3,3,423,*
Δεκαδική αναπαράσταση:	280068 και 14,15,3,3,423,*
Τρεις διακριτές περιπτώσεις:	0110001100011110000011010 και 14,15,3,3,423
Τέσσερις διακριτές περιπτώσεις:	0011000000110000001101000000000000110010 και 14,15,3,3,423

Πίνακας 15.1: Συμπύεση αραιών δεδομένων.

Στη συνέχεια εξετάζονται μερικές περιπτώσεις συμπύεσης αριθμητικών δεδομένων. Λέγεται ότι τα δεδομένα είναι **αραιά** (sparse) όταν σε συντριπτικό ποσοστό έχουν μηδενικές τιμές. Η περίπτωση των αραιών πινάκων έχει αναπτυχθεί και στο βιβλίο των Δομών Δεδομένων, αλλά το θέμα λόγω της σπουδαιότητας του θίγεται και εδώ. Στην πρώτη γραμμή του Πίνακα 15.1 δίνεται ένα παράδειγμα αραιών δεδομένων, όπου το σύμβολο '*' δηλώνει μία μη καθορισμένη (don't care) τιμή. Στη δεύτερη γραμμή του πίνακα δίνεται η αντίστοιχη παράσταση χρησιμοποιώντας τη γνωστή μέθοδο, όπου μόνο οι μη μηδενικές τιμές μαζί με τις συντεταγμένες τους στο αρχείο. Στην τρίτη γραμμή του πίνακα δίνεται η επίσης γνωστή αναπαράσταση αραιών δεδομένων με τη βοήθεια δυαδικών συμβολοσειρών (bitmap). Υπ' όψη ότι αυτή η δυαδική συμβολοσειρά μπορεί να γραφεί σε δεκαδική μορφή ως ο ακέραιος 280068, όπως φαίνεται στην τέταρτη σειρά του πίνακα. Αν υποθεθεί ότι οι μη καθορισμένες τιμές είναι σχετικά πολλές και ότι ενδιαφέρει η επακριβής θέση τους στον πίνακα με δυαδική αναπαράσταση, τότε πρέπει να γίνει διάκριση σε τρεις διακριτές περιπτώσεις. Έτσι, στην πέμπτη σειρά του πίνακα, οι μη καθορισμένες τιμές παριστώνται με 10, οι μη μηδενικές τιμές παριστώνται με 11, ενώ φυσικά με 0 παριστώνται οι μηδενικές τιμές. Τέλος, αν υπάρχουν πολλές επαναλαμβανόμενες τιμές τότε μπορούν να χρησιμοποιηθούν τέσσερις διακριτές τιμές: το 00 για τις μηδενικές τιμές, το 01 για τις επαναλαμβανόμενες τιμές, το 10 για τις μη καθορισμένες και το 11 για τις καθορισμένες μη μηδενικές τιμές, όπως παρουσιάζεται στην έκτη σειρά του πίνακα. Έχει αναφερθεί ότι αν η τελευταία μέθοδος εφαρμοσθεί σε μεγάλα αρχεία με στατιστικά δεδομένα, τότε ο απαιτούμενος χώρος μειώνεται κατά 45% ως 80%.

Όταν τιμές των δεδομένων είναι μικρές τότε μπορούν να παρασταθούν με λίγα bits αντί για ολόκληρες λέξεις. Ο αριθμός των απαραίτητων bits βρίσκειται εύκολα με μία σάρωση των δεδομένων. Για παράδειγμα, ο αριθμός

423 του Πίνακα 15.1 απαιτεί 9 bits. Συνεπώς, ενώ με την τελευταία μέθοδο του πίνακα απαιτούνται εκτός του δυαδικού ανύσματος άλλες τέσσερις λέξεις, τώρα απαιτούνται μία λέξη για το δείκτη μεγέθους και 36 bits για τα τέσσερα σημαντικά στοιχεία.

Αρχικά δεδομένα:	854, 855, 857, 856, 859, 860, 863
Συμπιεσμένη μορφή:	854, 4, 1, 2, -1, 3, 1, 3

Πίνακας 15.2: Κωδικοποίηση Δέλτα.

Αν οι τιμές των δεδομένων μεταβάλλονται αργά και βαθμιαία, τότε μπορεί να εφαρμοσθεί η **κωδικοποίηση Δέλτα**. Σύμφωνα με τη μέθοδο αυτή κατ' αρχήν αποθηκεύεται η πρώτη τιμή, κατόπιν ο αριθμός των απαραίτητων bits για την αποθήκευση των διαφορών και στη συνέχεια οι διαφορές. Ένα τέτοιο παράδειγμα φαίνεται στον Πίνακα 15.2. Σύμφωνα με μία παραλλαγή της μεθόδου αυτής παρεμβάλλεται περιοδικά ένας ολόκληρος αριθμός μετά από ένα συγκεκριμένο αριθμό διαφορών. Ειδική περίπτωση είναι η λεγόμενη **κωδικοποίηση των Αζτέκων**, όπου χρειάζεται μόνο ένα bit για την αύξηση ή τη μείωση των τιμών των δεδομένων. Η τελευταία μέθοδος είναι αποτελεσματική μόνο αν η περίοδος δειγματοληψίας είναι πολύ μικρή.

Αρχικοί ακέραιοι:	v_1, v_2, v_3, v_4
Τελικός αέρας:	$v_1 + m \times v_2 + m^2 \times v_3 + m^3 \times v_4 \quad (m > v_i)$

Πίνακας 15.3: Συνδυασμός πολλών μικρών ακεραίων σε ένα αέρας.

Πολλές φορές είναι δυνατόν ένας αριθμός μικρών ακεραίων να συνδυασθεί σε ένα μόνο αέρας με τη μέθοδο που φαίνεται στον Πίνακα 15.3. Αν μάλιστα η παράμετρος m του παραδείγματος είναι δύναμη του δύο, τότε η κωδικοποίηση και η αποκωδικοποίηση διευκολύνονται με την πράξη της διολίσθησης (shift).

Η συμπίεση χειμένου είναι επίσης ένα μεγάλο κεφάλαιο με ευρεία πρακτική εφαρμογή. Ήδη στο Κεφάλαιο 12.2 έγινε εκτενής αναφορά στη μέθοδο της κωδικοποίησης με υπέρθεση και δεν πρόκειται να εξετασθεί πάλι στο παρόν κεφάλαιο. Ένα πολύ συχνό παράδειγμα είναι η **σύντμηση ή συντομία** (abbreviation) ταξινομημένων κλειδιών καταλόγων και λέγεται **εσωτερική**

Edgar R.B. Hagstrom
 155 Proteus Park 60282HGS5155POT31
 Chicago, Ill 60282

Πίνακας 15.4: Ταχυδρομική διεύθυνση και αντίστοιχη κωδικοποιημένη.

(internal) ή εξωτερική (external), αν γίνεται πριν ή μετά την εισαγωγή στο σύστημα αντίστοιχα. Παράδειγμα εξωτερικής σύντμησης είναι μία μέθοδος κωδικοποίησης που εφαρμόζεται κυρίως σε στοιχεία συνδρομητών. Στον Πίνακα 15.4 δίνεται μία πλήρης ταχυδρομική διεύθυνση και η αντίστοιχη κωδικοποιημένη. Η κωδικοποιημένη διεύθυνση προκύπτει με πρόταξη του ταχυδρομικού κώδικα και παράθεση όλων των υπογραμμισμένων γραμμάτων, ενώ οι αριθμοί αντιστοιχούν στο πλήθος των πλαγίων γραμμάτων. Ακολουθεί ένας μετρητής, το 1, που αυξάνει στην περίπτωση που προκύψουν δύο ή περισσότερες όμοιες συμπιεσμένες συμβολοσειρές. Ας σημειωθεί ότι η πρόταξη του ταχυδρομικού κώδικα εξυπηρετεί στην ταξινόμηση.

1.	Αγνοούνται όλοι οι μη αλφαβητικοί χαρακτήρες	
2.	Οι πεζοί χαρακτήρες μετατρέπονται σε κεφαλαία	
3.	Λαμβάνεται ο πρώτος χαρακτήρας του ονόματος στο αποτέλεσμα	
4.	Αγνοούνται τα φωνήεντα A,E,I,O,U,Y και τα σύμφωνα H,W	
5.	Γίνονται οι εξής μετατροπές:	
	τα χειλέοφωνα	B,F,P,V σε 1
	τα λαρυγγικά και συριστικά	C,G,J,K,Q,S,X,Z σε 2
	τα οδοντικά	D,T σε 3
	τα μακρά υγρά	L σε 4
	τα ένρινα	M,N σε 5
	τα βραχεία υγρά	R σε 6
6.	Δύο ή περισσότερα ίδια όμοια ψηφία αντικαθιστώνται με ένα	
7.	Τα πρώτα τρία ψηφία λαμβάνονται στο αποτέλεσμα	

Πίνακας 15.5: Διαδικασία συμπίεσης με μέθοδο SOUNDEX.

Η SOUNDEX είναι μία σπουδαία μέθοδος εξωτερικής σύντμησης, η οποία χρησιμεύει για τη συμπίεση κυρίων αγγλικών ονομάτων και μπορεί να θεωρηθεί σαν ένας φωνητικός κατακερματισμός. Ο λόγος αυτής της θεωρήσης είναι το γεγονός ότι ονόματα που μοιάζουν φωνητικά μεταξύ τους συγκεντρώνονται σε παραπλήσιες θέσεις του αρχείου, όπως συμβαίνει με

τις συγχρούμενες εγγραφές ενός αρχείου κατακερματισμού. Το πλεονέκτημα της μεθόδου είναι ότι με τη μέθοδο αυτή μειώνονται οι συνέπειες που μπορεί να συμβούν από ένα λάθος κατά την επεξεργασία ενός ονόματος. Οι κανόνες συμπίεσης ενός ονόματος με τη μέθοδο SOUNDEX παρουσιάζονται στον Πίνακα 15.5. Όπως εύκολα φαίνεται, το αποτέλεσμα της διαδικασίας είναι μία συμβολοσειρά από ένα μέχρι τέσσερις χαρακτήρες. Το πλήθος των διαφορετικών συμβολοσειρών SOUNDEX είναι 26×7^3 , δηλαδή 8918 δυνατότητες. Άρα είναι πολύ πιθανό δύο ή περισσότερα ονόματα να δίνουν τον ίδιο κώδικα SOUNDEX, όπως εξάλλου συμβαίνει και στον απλό κατακερματισμό. Για παράδειγμα, τα ονόματα MacCloud και McLeod ή τα ονόματα Smith και Schmidt δίνουν αποτέλεσμα M243 και Σ53 αντίστοιχα. Επομένως είναι πιθανό κατά την αναζήτηση ενός ονόματος να επιστραφεί στο χρήστη ένα σύνολο ονομάτων, οπότε η επιλογή θα γίνει με ευθύνη του σε συνδυασμό με τα υπόλοιπα πεδία.

Οι μέθοδοι εσωτερικής σύντμησης είναι βασικά δύο, αλλά υπάρχει και ο υβριδικός συνδυασμός τους. Σύμφωνα με τη **σύντμηση κλειδιού χαμηλής τάξης** (low order key abbreviation) ή **οπίσθια περικοπή** (rear truncation) λαμβάνονται από τα διαδοχικά κλειδιά τόσοι αρχικοί χαρακτήρες, ώστε να υπάρξει διαφοροποίηση μεταξύ τους και αγνοούνται οι τελικοί χαρακτήρες τους. Η τεχνική αυτή είναι γενικά γνωστή από τους τόμους των πολύτομων εγκυκλοπαιδειών και λεξικών, αλλά σε ότι αφορά στα αρχεία χρησιμοποιείται στα Προθεματικά B⁺-δένδρα επιτυχάνοντας μεγαλύτερο λόγο διακλάδωσης και επομένως μεγαλύτερο κέρδος σε χώρο και χρόνο. Πιο συγκεκριμένα, στις σελίδες των κατωτέρων επιπέδων του δένδρου δεν αποθηκεύονται ολόκληρα τα κλειδιά αν όλα έχουν κοινό πρόθεμα. Το πρόθεμα αυτό μπορεί να ανιχνευθεί στα ανώτερα επίπεδα του δένδρου. Ακόμη στα Προθεματικά B⁺-δένδρα τα συμπιεσμένα κλειδιά μίας σελίδας δεν αποθηκεύονται ολόκληρα αλλά μόνο οι διαφορές τους ως προς το προηγούμενο κλειδί. Η τεχνική αυτή χρησιμοποιείται και από την οργάνωση VSAM στην ομάδα καταλόγων ως εξής: κάθε συμβολοσειρά αποθηκεύεται με ένα ψηφίο που δείχνει ποσά κοινά γράμματα έχει το κλειδί αυτό με το προηγούμενό του και έπονται τα υπόλοιπα γράμματα της συμβολοσειράς. Αν οι διαδοχικές συμβολοσειρές διαφέρουν κατά πολύ, τότε το κέρδος σε χώρο είναι σημαντικό.

Σύμφωνα με τη **σύντμηση κλειδιού υψηλής τάξης** (high order key abbreviation) ή **εμπρόσθια περικοπή** (front truncation), αν οι αρχικοί χαρακτήρες διαδοχικών συμβολοσειρών είναι ίδιοι τότε μπορούν να αγνοηθούν ή να συμπιεσθούν. Παρ' ότι η τεχνική αυτή θεωρητικά έχει πλεονεκτήματα

καθώς επιτυγχάνει οικονομία χώρου, εντούτοις έχει σοβαρά μειονεκτήματα καθώς:

- απαιτεί αυξημένο χρόνο επεξεργασίας, και
- καθιστά επαχθέστερο το συντονισμό των ταυτόχρονων προσπελάσεων (δες επόμενο υποκεφάλαιο).

Για παράδειγμα, στην έκδοση Oracle 5 χρησιμοποιήθηκε η τεχνική αυτή, που όμως καταργήθηκε στην επόμενη έκδοση, καθώς διαπιστώθηκε ότι η επίδραση της στην επίδοση ήταν αρνητική. Πιο σύνθετη είναι βέβαια η περίπτωση που χρησιμοποιείται συνδυασμός των συντμήσεων υψηλής και χαμηλής τάξης. Προφανώς τα δεδομένα που είναι αποθηκευμένα κατά αυτόν τον τρόπο μπορούν να αναγνωσθούν μόνο σειριακά. Στον Πίνακα 15.6 φαίνονται τα κλειδιά που περιέχονται σε μία σελίδα αρχικά χωρίς σύντμηση, με σύντμηση υψηλής τάξης, με σύντμηση χαμηλής τάξης και κατόπιν με συνδυασμό των δύο συντμήσεων.

Κλειδί	Υψηλής τάξης	Χαμηλής τάξης	Συνδυασμός
JEFCOATE IF	JEFCOATE IF	JEFCO	JEFCO
JEFCUT R	4UT R	JEFCU	3CU
JEFF CJ	3F CJ	JEFF C	3F C
JEFF D	5D	JEFF D	5D
JEFF KD	5KD	JEFF K	5K
JEFFAIR DC	4AIR DC	JEFFAI	4AI
JEFFARES AS	5RES AS	JEFFARES A	5REA A
JEFFARES K	9K	JEFFARES K	9K
JEFFARES KM	9KM	JEFFARES KM	9KM
JEFFCOATE DV	4COATE DV	JEFFCOATE DV	4COATE DV

Πίνακας 15.6: Μέθοδοι εσωτερικής σύντμησης.

Εκτός από τη σύντμηση υπάρχουν και πολλές άλλες μέθοδοι κωδικοποίησης συμβολοσειρών. Έτσι, αν κάθε χαρακτήρας παρασταθεί με 7 ή με 6 bits γίνεται οικονομία κατά 12.5% ή 25% αντίστοιχα. Με 6 bits μπορούν να παρασταθούν 64 διαφορετικοί χαρακτήρες που συνήθως είναι τα κεφαλαία και τα πεζά αγγλικά γράμματα, τα δέκα ψηφία και δύο σύμβολα (το κενό και η παύλα). Αν δεν υπάρχει ανάγκη χρήσης πεζών γραμμάτων, τότε αρκούν 5 bits.

Συχνά στις συμβολοσειρές συναντώνται ακολουθίες κενών, όπως στον Πίνακα 15.7. Μία τέτοια ακολουθία μπορεί να παρασταθεί με ένα μόνο

Giovannini	123.89	Giovannini	1123.89
Jean	87.50	Jean	887.50
Johann	103.76	Johann	5103.76
John	154.66	John	7154.66

Πίνακας 15.7: Διαγραφή κενών.

κενό ακολουθούμενο από ένα αριθμό που δηλώνει το πλήθος των κενών. Επίσης αντί ενός πλήθους συνεχόμενων ομοίων χαρακτήρων μπορεί να τεθεί ένα ειδικό σύμβολο, ένας αριθμός που δηλώνει το πλήθος των χαρακτήρων και στη συνέχεια ο χαρακτήρας. Για παράδειγμα, αντί `111111` μπορεί να γραφεί `*61`. Προφανώς, όταν το πλήθος των ομοίων χαρακτήρων είναι μεταξύ του τέσσερα και του εννέα, τότε η μέθοδος είναι αποδοτική. Αν το πλήθος των ομοίων συνεχόμενων χαρακτήρων είναι περισσότερο του δέκα, τότε μπορεί να γίνει χρήση διψήφιου μετρητή. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για τη συμπίεση χαρτών που παριστώνται με πίνακες τύπου raster. Συνηθισμένη τεχνική επίσης είναι η αντικατάσταση των συχνών λέξεων με απλούς χαρακτήρες, όπως για παράδειγμα οι αγγλικές λέξεις the, of, to, and, in, that, is, it, for, one.

Γράμμα	Συχνότητα	Κωδικός	Γράμμα	Συχνότητα	Κωδικός
E	133	111	U	28	01010
T	93	000	M	27	00110
O	85	1111	P	22	00100
A	81	1110	Y	15	011111
N	75	1100	W	15	011110
I	71	1011	G	14	001111
R	70	1010	B	13	001110
S	65	0110	V	10	001010
H	61	0100	K	5	0010110
D	43	11011	X	3	00101110
L	38	11010	J	2	001011110
C	31	01110	Q	2	0010111111
F	29	01011	Z	1	00101111110

Πίνακας 15.8: Συχνότητα εμφάνισης και κωδικοί Huffman.

Είναι γνωστό ότι τα γράμματα του αλφαβήτου δεν εμφανίζονται ισόπιθانا σε ένα κείμενο. Στον Πίνακα 15.8 δίνεται η σχετική συχνότητα

εμφάνισης των γραμμάτων του αγγλικού αλφαβήτου και παρατηρείται ότι η απόκλιση στη συχνότητα εμφάνισης μεταξύ των γραμμάτων E και Z είναι τεράστια. Μπορεί να υπάρξει οικονομία στο χώρο αποθήκευσης αν κάθε χαρακτήρας δεν αντιστοιχεί στον ίδιο αριθμό bits. Στην παρατήρηση αυτή στηρίζεται η μέθοδος κωδικοποίησης με σύμβολα μεταβλητού μήκους (variable length symbol encoding), γνωστή και ως μέθοδος Huffman, που αναπτύχθηκε αλγοριθμικά στο βιβλίο των Δομών Δεδομένων. Σύμφωνα με τη μέθοδο Huffman αντιστοιχίζεται μικρότερος ή μεγαλύτερος κωδικός στα συχνότερα και στα σπανιότερα γράμματα, αντίστοιχα. Στον Πίνακα 15.8 φαίνεται επίσης για κάθε γράμμα ο αντίστοιχος κωδικός. Λαμβάνοντας υπ' όψη τις συχνότητες προκύπτει ότι το μέσο μήκος των κωδικών είναι 4,1754 bits, ενώ αν κάθε γράμμα παριστάνονταν με ισομήχεις κωδικούς τότε θα χρειαζόταν $\log_2 26 = 4,70$ bits. Όσο περισσότερο η κατανομή πιθανότητας εμφάνισης των γραμμάτων ενός αλφαβήτου αποκλίνει από την ομοιόμορφη κατανομή, τόσο η μέθοδος αυτή δίνει μεγαλύτερα κέρδη σε χώρο. Επίσης, σημειώνεται ότι αν αλλάξει η κατανομή πιθανοτήτων, τότε το κέρδος μπορεί να μειωθεί αλλά μπορεί και να αυξηθεί. Η προσπέλαση αρχείου κειμένου που έχει συμπιεσθεί με τη μέθοδο αυτή μπορεί να γίνει μόνο με σειριακό τρόπο.

Αν στα δεδομένα εφαρμόζεται κάποια μέθοδος συμπίεσης, τότε προκύπτουν εγγραφές μεταβλητού μήκους που απαιτούν ιδιαίτερη προσοχή στο χειρισμό τους. Σημειώνεται ότι δεν προσφέρουν όλα τα συστήματα αρχείων τη δυνατότητα χρήσης τέτοιων εγγραφών και απομένει στον προγραμματιστή η υλοποίηση των σχετικών διαδικασιών.

15.4 Συντονισμός ταυτόχρονων προσπελάσεων

Όταν ένα αρχείο προσπελάζεται από πολλούς χρήστες ταυτόχρονα, τότε μπορεί να παρουσιασθούν τα λεγόμενα προβλήματα ταυτοχρονισμού (concurrency) και μπορεί να συμβούν μερικές απροσδόκητες παρενέργειες. Οι παρενέργειες αυτές φαίνονται ανάγλυφα στο επόμενο παράδειγμα.

Δύο άτομα έχουν κοινό λογαριασμό ταμειυτηρίου, ο οποίος προς στιγμή έχει υπόλοιπο 500.000 δραχμές. Ταυτόχρονα και οι δύο πηγαίνουν σε διαφορετικά υποκαταστήματα και οι ταμίες κάνουν προσπέλαση για την ανάγνωση της σχετικής εγγραφής. Έστω ότι ο πρώτος ζητά ανάληψη 100.000 δραχμών, ενώ ο δεύτερος 200.000 δραχμών. Συνεπώς από τον πρώτο ταμιά

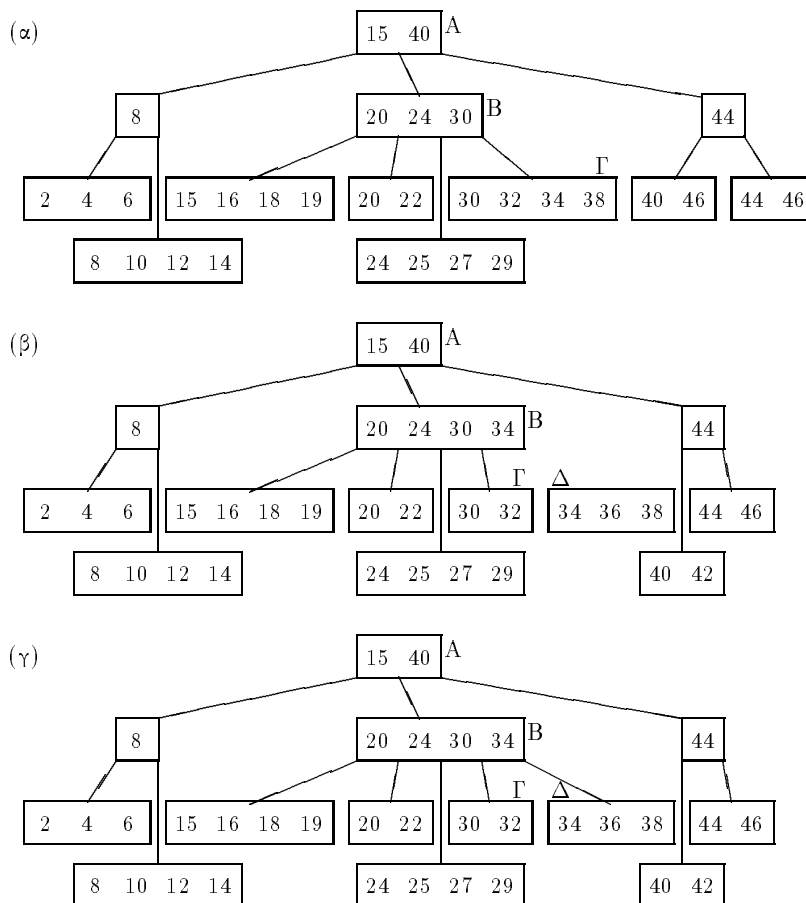
γίνεται αποθήκευση της τιμής 400.000 στο υπόλοιπο του λογαριασμού, ενώ από το δεύτερο γίνεται αποθήκευση της τιμής 300.000. Προφανώς, η τελική αυτή τιμή δεν είναι η ορθή.

Κάθε συναλλαγή ή δοσοληψία (transaction) αποτελείται από επιμέρους βήματα-διεργασίες σε εγγραφές ή σελίδες ενός αρχείου ή περισσοτέρων αρχείων. Η αλληλουχία των επιμέρους βημάτων λέγεται χρονικό (schedule) των συναλλαγών. Ένα χρονικό λέγεται σειριακό (serial), αν τα βήματα μίας συναλλαγής δεν παρεμβάλλονται στα βήματα μίας άλλης συναλλαγής, δηλαδή μία συναλλαγή αρχίζει να εκτελείται όταν τελειώσει η προηγούμενή της. Όμως ο κανόνας σε συστήματα πολλών χρηστών και καταμερισμού χρόνου είναι οι επιμέρους διεργασίες ταυτόχρονων συναλλαγών να παρεμβάλλονται μεταξύ τους. Σκοπός του ελέγχου του ταυτοχρονισμού (concurrency control) είναι τα βήματα ενός χρονικού να διαταχθούν, ώστε να προκύψει το ίδιο ορθό αποτέλεσμα σαν το χρονικό να ήταν σειριακό. Τότε το χρονικό λέγεται σειριοποιήσιμο (serializable). Μία απλή τεχνική που εγγυάται τη σειριοποιησιμότητα είναι το κλείδωμα (locking) των κρίσιμων δεδομένων, δηλαδή η στέρηση της δυνατότητας χρήσης τους από άλλους χρήστες.

Τα προβλήματα ταυτοχρονισμού δεν εμφανίζονται μόνο όταν δύο ή περισσότεροι χρήστες επιθυμούν να προσπελάσουν την ίδια εγγραφή του αρχείου, αλλά ακόμη και διαφορετικές εγγραφές. Για το λόγο αυτό, διακρίνεται το κλείδωμα εγγραφής (record locking) και το κλείδωμα σελίδας (page locking). Τα κλειδωμάτα σελίδων χρησιμοποιούνται ιδιαίτερα σε ισοζυγισμένες δομές, όπως σε AVL, B-δένδρα κλπ., αλλά και σε τυχαία δυναμικά αρχεία, που διακρίνονται από διασπάσεις και συγχωνεύσεις κόμβων. Στη συνέχεια θα δοθεί ένα ακόμη παράδειγμα ταυτόχρονων προσπελάσεων σε B^+ -δένδρο (με $d=2$ και $Bkfr=4$), που χρησιμοποιείται ως δευτερεύων κατάλογος.

Έστω ότι στο δένδρο του Σχήματος 15.1 πρέπει ταυτόχρονα να εισαχθεί το κλειδί 36 και να αναζητηθεί το κλειδί 38. Στον Πίνακα 15.9 φαίνεται ένα πιθανό χρονικό των επιμέρους βημάτων των δύο διεργασιών. Είναι φανερό ότι η αναζήτηση του κλειδιού 38 καταλήγει σε λανθασμένο συμπέρασμα. Λανθασμένο συμπέρασμα μπορεί να προκύψει σε ταυτόχρονη εισαγωγή και διαγραφή κλειδιών, όπως για παράδειγμα κατά την διαγραφή του κλειδιού 32 και την εισαγωγή του κλειδιού 31 στο δένδρο του Σχήματος 15.1γ.

Για τον έλεγχο του ταυτοχρονισμού σε δομές της οικογένειας των B-δένδρων έχει δοθεί πληθώρα λύσεων. Γενικά είναι παραδεκτή η επόμενη προ-



Σχήμα 15.1: Μορφή B⁺-δένδρου σύμφωνα με τον Πίνακα 15.9.

βληματική. Παρενέργειες δεν δημιουργούνται αν οι απαιτήσεις των χρηστών είναι μόνο αναγνώσεις. Στην αντίθετη περίπτωση πρέπει μερικοί κόμβοι να κλειδωθούν εκ μέρους μίας διεργασίας μέχρι την ολοκλήρωσή της, ώστε να μην είναι προσπελάσιμοι από τις άλλες διεργασίες. Οι αλγόριθμοι κλειδώματος-ξεκλειδώματος κόμβων χωρίζονται σε δύο κατηγορίες: από την κορυφή προς τη βάση (top-down) και από τη βάση προς την κορυφή (bottom-up). Οι αλγόριθμοι της δεύτερης κατηγορίας είναι πιο αποτελεσματικοί γιατί διατηρούν μικρότερο αριθμό κλειδωμένων κόμβων και επιτρέπουν περισσότερο ταυτοχρονισμό.

Βήμα	Εισαγωγή 36	Αναζήτηση 38
1	Βρες το παιδί του κόμβου A (είναι ο κόμβος B)	
2	Βρες το παιδί του κόμβου B (είναι ο κόμβος Γ)	
3		Βρες το παιδί του κόμβου A (είναι ο κόμβος B)
4		Βρες το παιδί του κόμβου B (είναι ο κόμβος Γ)
5	Ο κόμβος Γ είναι φύλλο, πρόσθεσε το 36 στον κόμβο Γ, ο κόμβος Γ είναι πλήρης	
6	Διάσπασε τον κόμβο Γ σε Γ και Δ	
7	Ανέβασε το κλειδί 34 στον κόμβο B	
8		Ο κόμβος Γ είναι φύλλο, αναζήτησε το 38 στον κόμβο Γ, ανεπιτυχής αναζήτηση !

Πίνακας 15.9: Χρονικό βημάτων δύο διεργασιών.

Ιδιαίτερη σημασία για την επίδοση των σχετικών αλγορίθμων έχει το επίπεδο, όπου βρίσκεται ο **ασφαλής κόμβος** (safe node), δηλαδή ο ανώτερος κόμβος του μονοπατιού που μπορεί να απορροφήσει οποιαδήποτε αλλαγή των κατωτέρων επιπέδων της δομής. Αν αυτός ο κόμβος περιέχει λιγότερο από το μέγιστο αριθμό επιτρεπόμενων κλειδιών λέγεται **ασφαλής κατά την εισαγωγή** (insertion safe), ενώ αν περιέχει περισσότερο από τον ελάχιστο αριθμό λέγεται **ασφαλής κατά τη διαγραφή** (deletion safe). Το παιδί του πλησιέστερου προς τα φύλλα ασφαλούς κόμβου ονομάζεται **υψηλότερος μη ασφαλής κόμβος** (highest unsafe node). Είναι φανερό ότι όσο πλησιέστερα προς τη ρίζα είναι ο υψηλότερος μη ασφαλής κόμβος, τόσο δυσκολότερη είναι η ταυτόχρονη χρήση του δένδρου από πολλούς χρήστες λόγω των κλειδωμάτων. Διακρίνονται τεσσάρων ειδών κλειδώματα: **ανάγνωσης, εισαγωγής, διαγραφής και αποκλειστικότητας**. Τα κλειδώματα εισαγωγής και διαγραφής είναι ταυτόχρονα και κλειδώματα ανάγνωσης.

Έστω ότι το μέγιστο επιτρεπόμενο περιεχόμενο ενός κόμβου είναι $2d$ κλειδιά, ενώ ένας συγκεκριμένος κόμβος περιέχει s κλειδιά. Ο κόμβος αυτός μπορεί να δεχθεί $(2d-s)$ κλειδώματα εισαγωγής και $(s-d)$ κλειδώματα διαγραφής. Τα κλειδώματα κάθε κόμβου τοποθετούνται σε μία αντίστοιχη

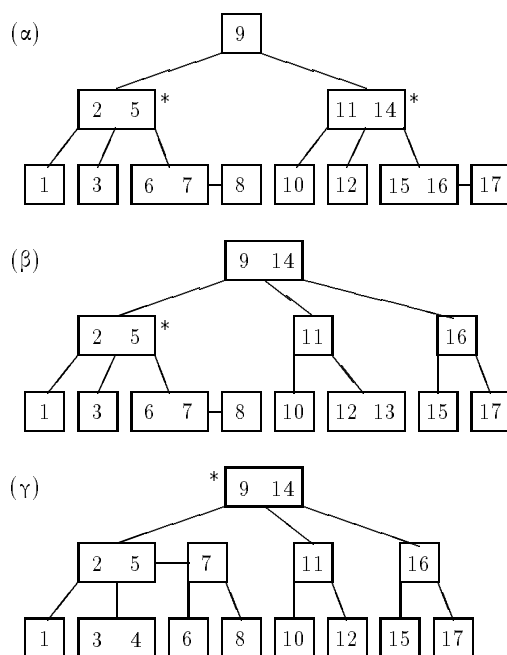
ουρά και παίζουν το ρόλο των κρατήσεων των ελεύθερων θέσεων του κόμβου. Αν μία συναλλαγή απαιτήσει να επιφέρει αλλαγή στο περιεχόμενο του κόμβου, τότε πρώτα το αντίστοιχο κλείδωμα μετατρέπεται σε αποχλειστικό κλείδωμα και διαγράφεται από τη λίστα, και ύστερα εκτελείται η αλλαγή που απαιτείται.

Είναι προφανές ότι ο έλεγχος ταυτοχρονισμού, εκτός του ότι είναι μία πολύπλοκη και χρονοβόρα διαδικασία, απαιτεί την ύπαρξη πινάκων στην κύρια μνήμη για τη διαχείριση των ουρών. Το πρόβλημα του συντονισμού των ταυτόχρονων προσπελάσεων έχει ερευνηθεί εκτεταμένα στο παρελθόν και συνεχίζει να αποτελεί επίκαιρο θέμα, τόσο από πλευράς αλγοριθμικής όσο και από αναλυτικής. Το αντικείμενο εξετάζεται κυρίως στο πλαίσιο του μαθήματος των Βάσεων Δεδομένων, και συνεπώς μεγαλύτερη ανάπτυξη των σχετικών τεχνικών συντονισμού των ταυτόχρονων προσπελάσεων υπερβαίνει τα πλαίσια του βιβλίου αυτού. Ωστόσο, έχει γίνει σημαντική έρευνα, και συνεχίζει να γίνεται, με σκοπό την εύρεση δομών που επιτρέπουν από τη φύση τους μεγαλύτερη ταυτόχρονη χρήση τους από πολλούς χρήστες. Στη συνέχεια θα παρουσιασθεί μία παραλλαγή του Β-δένδρου που χαρακτηρίζεται από καλή επίδοση κατά την ταυτόχρονη χρήση της από πολλούς χρήστες.

Είναι γνωστό ότι αν ένα φύλλο Β-δένδρου υπερχειλίζει λόγω εισαγωγής μίας εγγραφής, τότε γίνεται διάσπαση του κόμβου με ταυτόχρονη άνοδο του μεσαίου κλειδιού, που μπορεί να προκαλέσει νέα διάσπαση. Από τον Keller προτάθηκε η εξής ενδιαφέρουσα τεχνική (1988). Αν σε περίπτωση υπερχειλίσης ο πατρικός κόμβος δεν είναι πλήρης, τότε ακολουθείται η γνωστή διαδικασία. Όμως αν ο πατρικός κόμβος είναι πλήρης (οπότε με το ανερχόμενο κλειδί θα πρέπει να διασπασθεί) τότε: (α) δημιουργείται ένας νέος κόμβος-φύλλο και οι εγγραφές διαμοιράζονται στους δύο αδελφούς (sibling) κόμβους, (β) το μεσαίο κλειδί αποθηκεύεται στον πρώτο από τους δύο αδελφούς, (γ) οι δύο αδελφοί κόμβοι συνδέονται με έναν οριζόντιο δείκτη, και (δ) ο πατρικός κόμβος σημαδεύεται ότι ο συγκεκριμένος κόμβος δείχνει σε έναν αδελφό.

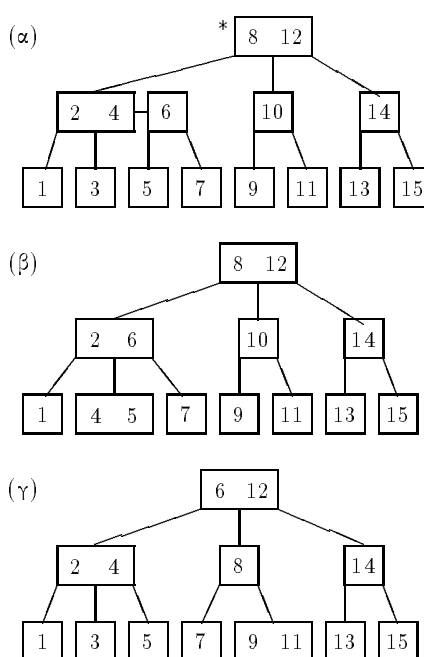
Για να υλοποιηθεί η παραλλαγή αυτή απαιτεί ένα νέο ορισμό του τύπου της εγγραφής της δομής. Πιο συγκεκριμένα, η εγγραφή περιλαμβάνει ένα επιπλέον πεδίο με τιμές από 0 ως m . Αν σε ένα κόμβο η τιμή του πεδίου αυτού είναι 0, τότε κανείς κόμβος που δεικτοδοτείται από το συγκεκριμένο κόμβο δεν δείχνει σε αδελφό του με οριζόντιο δείκτη. Αν η τιμή του πεδίου είναι από 1 ως m , τότε αυτό σημαίνει ότι ο αντίστοιχος δεικτοδοτούμενος κόμβος δείχνει με οριζόντιο δείκτη σε αδελφό κόμβο.

Η καταστρατήγηση του ορισμού του κλασικού B-δένδρου έχει το πλεονέκτημα ότι δεν επιτρέπει στο ανερχόμενο κλειδί να προκαλέσει τη διάσπαση ενός ή περισσότερων κόμβων του μονοπατιού προς τη ρίζα με επακόλουθο την αποφυγή των κλειδωμάτων στα υψηλότερα επίπεδα. Η καταστρατήγηση αυτή είναι προσωρινή, επειδή μπορεί να τακτοποιηθεί σε μία κατοπινή αναζήτηση, εισαγωγή ή διαγραφή. Για παράδειγμα, αν κατά την εκτέλεση μίας αναζήτησης ή μίας εισαγωγής προσπελασθεί ο σημαδεμένος κόμβος, τότε πριν την ολοκλήρωση της ακολουθείται πρώτα η εξής διαδικασία: (α) το φύλλο παύει να δείχνει στον αδελφό του με οριζόντιο δείκτη, (β) το μεσαίο κλειδί ανέρχεται κατά ένα επίπεδο, (γ) ο πατρικός κόμβος διασπάται και προκύπτουν δύο κόμβοι, (δ) το νέο μεσαίο κλειδί ανέρχεται κατά ένα επίπεδο μόνο υπάρχει διαθέσιμος χώρος, αλλιώς οι δύο κόμβοι συνδέονται με οριζόντιο δείκτη και σημαδεύεται ο νέος πατρικός κόμβος. Έτσι, το φαινόμενο αυτό μπορεί να επαναληφθεί σταδιακά στα ανώτερα επίπεδα μέχρι τη διάσπαση της ρίζας. Για το λόγο αυτό η παραλλαγή αυτή ονομάζεται B-δένδρο με τεμπέλικη διάσπαση ρίζας (lazy root splitting B-tree).



Σχήμα 15.2: Εισαγωγές με τεμπέλικη διάσπαση ρίζας.

Η περίπτωση της διαγραφής μίας εγγραφής που ανήκει σε κόμβο που δεικτοδοτείται από σημαδεμένο κόμβο είναι διαφορετική. Δηλαδή, δεν εκτελείται πρώτα η τακτοποίηση και κατόπιν η διαγραφή, αλλά γίνονται ταυτόχρονα. Για παράδειγμα, είναι πιθανό η διαγραφή να αφορά σε μία εγγραφή που είναι αποθηκευμένη σε έναν από τους δύο αδελφούς κόμβους που συνδέονται με οριζόντιο δείκτη. Είναι προφανές ότι μετά τη διαγραφή το περιεχόμενο των δύο κόμβων μπορεί να αποθηκευθεί σε ένα μόνο κόμβο, οπότε θα πρέπει ο σημαδεμένος κόμβος να ενημερωθεί ανάλογα.



Σχήμα 15.3: Διαγραφές με τεμπέλικο διάσπαση ρίζας.

Πρέπει να τονισθεί ότι οι διαδικασίες αναζήτησης, εισαγωγής και διαγραφής είναι αρκετά πιο πολύπλοκες σε αυτή τη δομή απ' ό,τι στο κλασικό B-δένδρο και οι απαραίτητες λεπτομέρειες βρίσκονται στην αναφορά. Όμως δίνονται μερικά παραδείγματα με σχήματα. Στο αρχικό B-δένδρο τάξης $m=3$ του Σχήματος 15.2α εισάγεται μία εγγραφή με κλειδί 13 και κατόπιν μία εγγραφή με κλειδί 4. Η διαδοχική μορφή της δομής παρουσιάζεται στα Σχήματα 15.2β και 15.2γ. Στο Σχήμα 15.3 παρουσιάζεται η μορφή του δένδρου κατά τη διαγραφή. Πιο συγκεκριμένα, το Σχήμα 15.3β προέρχεται από

το Σχήμα 15.3α με διαγραφή της εγγραφής με κλειδί 3, ενώ μετά τη διαγραφή της εγγραφής με κλειδί 10 προκύπτει το Σχήμα 15.3γ. Ο αστερίσκος δηλώνει ότι υπάρχει αδελφός με οριζόντιο δείκτη.

Επακριβή ποσοτικά στοιχεία για τη σύγκριση των δύο δομών δεν υπάρχουν. Ωστόσο η μελέτη των Eisenbarth και άλλων, η οποία αναφέρθηκε στο αντίστοιχο κεφάλαιο δίνει μερικά ενδιαφέροντα στοιχεία για τα τυχαία B-δένδρα τάξης 3. Στη δομή, λοιπόν, αυτή η πιθανότητα να βρίσκεται ο ασφαλής κόμβος κατά την εισαγωγή στο πρώτο, δεύτερο, τρίτο ή ακόμη υψηλότερο επίπεδο από το επίπεδο των φύλλων είναι 0,57, 0,25, 0,1 και 0,08 αντίστοιχα. Βέβαια, οι τιμές αυτές δηλώνουν την ασυμπτωτική πιθανότητα να γίνει διάσπαση κατά την εισαγωγή της $(n+1)$ -οστής εγγραφής. Επομένως, με τις αντίστοιχες επίσης πιθανότητες θα γίνουν αποκλειστικά κλειδώματα στα αντίστοιχα επίπεδα. Αντίθετα, με την εφαρμογή της τεχνικής της τεμπέλικης διάσπασης της ρίζας κλειδώνεται μόνο ο πατρικός κόμβος του κόμβου που υπερχειλίζει. Έτσι είναι προφανές ότι κατά την επεξεργασία αυτής της δομής διατηρούνται λιγότεροι κόμβοι κλειδωμένοι απ' ότι στο απλό B-δένδρο. Εξ άλλου με την τακτοποίηση της δομής, η οποία γίνεται τοπικά μετά από μία διαγραφή αποφεύγεται το ενδεχόμενο να γίνουν διαδοχικές αλληπάλληλες συγχωνεύσεις λόγω διαγραφής μετά από διαδοχικές αλληπάλληλες διασπάσεις λόγω εισαγωγής. Βέβαια, ευνόητο είναι ότι όσο μεγαλύτερη είναι η τάξη του δένδρου, τόσο μειώνεται η ευεργετική επίδραση της τεχνικής αυτής στην επίδοση της ταυτόχρονης χρήσης του δένδρου.

15.5 Επίλογος

Ίσως στον αναγνώστη υπάρχει η αντίληψη ότι η σημασία της μελέτης των δομών των αρχείων είναι ανεπίκαιρη, καθώς φαίνεται ότι τα σύγχρονα εμπορικά συστήματα διαχείρισης βάσεων δεδομένων είναι πανίσχυρα. Ωστόσο, η άποψη αυτή είναι λανθασμένη, επειδή στο φυσικό επίπεδο οι βάσεις δεδομένων δεν είναι παρά αλληλοσυνδεδεμένα αρχεία και κατάλογοι και επομένως δεν μπορεί ένα σύστημα διαχείρισης βάσεων δεδομένων να επιτύχει αυξημένες επιδόσεις αν δεν στηρίζεται σε σωστά σχεδιασμένα αρχεία και καταλόγους. Η έρευνα των συστημάτων αρχείων δεν έχει σταματήσει λόγω των εξελίξεων στους παρακάτω, μεταξύ των άλλων, πέντε τομείς.

Πρώτον, η πρόταση νέων δομών αρχείων προκύπτει εν μέρει και από την εξέλιξη της τεχνολογίας υλικού. Όποτε εμφανίζονται νέες συσχευές αποθήκευσης, ταυτόχρονα αναπτύσσονται και νέες οργανώσεις. Ακόμη συμβαίνει

και το αντίστροφο, δηλαδή μία τεχνολογία εγκαταλείπεται η περιορίζεται η εφαρμογή της, αν δεν γίνει η απαραίτητη επένδυση στην ανάπτυξη λογισμικού. Έτσι η εξέλιξη της τεχνολογίας των laser και η εμπορική διάθεση οπτικών δίσκων (όπως VD-ROM, DVD κλπ.) έδωσε έδαφος για την πρόταση νέων δομών που αξιοποιούνται σε πλήθος νέων εφαρμογών, όπως για παράδειγμα στις διαχρονικές βάσεις δεδομένων, στις πολυμεσικές βάσεις δεδομένων (multimedia database systems) κλπ. Άλλη μία εξελισσόμενη τεχνολογία είναι, βέβαια, και η τεχνολογία των ημιαγωγών. Όλο και μεγαλύτερο τμήμα μίας βάσης μπορεί να αποθηκευθεί στην κύρια μνήμη και δεν χρειάζεται να προσπελάζονται τα ίδια δεδομένα συνεχώς στο δίσκο. Οι οργανώσεις αυτές ονομάζονται βάσεις δεδομένων κύριας μνήμης (Main Memory Databases) και έχουν ήδη αποκτήσει εμπορική υπόσταση με σκοπό την υποστήριξη εξειδικευμένων εφαρμογών, όπου ο χρήστης ή η διεργασία απαιτεί άμεση ικανοποίηση. Τέτοιες εφαρμογές εμφανίζονται στο ηλεκτρονικό εμπόριο, στις μηχανές αναζήτησης για τον παγκόσμιο ιστό, σε συστήματα κινητής τηλεφωνίας, καθώς και σε βιομηχανικά ευέλικτα συστήματα παραγωγής (flexible manufacturing systems, FMS). Τέτοια εμπορικά συστήματα είναι το TimesTen και το DataBlitz.

Δεύτερον, εμφανίζονται αυξημένες ανάγκες για γεωγραφικά κατανομημένη επεξεργασία. Αυτό έχει ως αποτέλεσμα να εφαρμόζονται οι τεχνικές της κατανομής (distribution) και της ολικής ή μερικής αντιγραφής (replication) των αρχείων σε περισσότερο από έναν κόμβους του δικτύου, οι οποίοι μπορεί να χρησιμοποιούν διαφορετικά υλικά και ετερογενή λειτουργικά συστήματα. Ο τελικός χρήστης δεν αντιλαμβάνεται που βρίσκονται τα δεδομένα και για το λόγο αυτό λέγεται ότι τα δεδομένα είναι διαφανή (transparent). Βέβαια, το μόνο που μπορεί να αντιληφθεί ο χρήστης είναι η χρονική επίδοση, που φθίνει αν η κατανομή και η αντιγραφή των δεδομένων δεν γίνει με τον καλύτερο τρόπο. Η κατανομή μίας οργάνωσης σε ένα σύνολο ανεξάρτητων δίσκων είναι ένα θέμα που έχει εξετασθεί σε βάθος για πλήθος μεθόδων προσπέλασης. Η κατανομή θα πρέπει να γίνει με τέτοιο έξυπνο τρόπο έτσι ώστε να εξυπηρετείται η αναζήτηση, δηλαδή κατά την αναζήτηση θα πρέπει να μεγιστοποιείται ο παραλληλισμός των προσπελάσεων. Βέβαια, η απαίτηση για μεγιστοποίηση του παραλληλισμού κατά την αναζήτηση ικανοποιείται αν υπάρχουν αντίγραφα των δεδομένων σε όλους τους κόμβους. Ωστόσο, μία τέτοια λύση δεν συνιστάται επειδή, εκτός του απαιτούμενου τεράστιου αποθηκευτικού χώρου, θα υπάρχει και επιπλέον κόστος ενημέρωσης των αντιγράφων σε κάθε περίπτωση εισαγωγής, διαγραφής ή ενημέρωσης. Τέλος, σε ένα κατανομημένο περιβάλλον πρέπει να υπάρχουν

τεχνικές ανάκτησης (recovery) των δεδομένων των αρχείων σε περίπτωση βλάβης ενός κόμβου. Επίσης, σημειώνεται ότι παρόμοια προβλήματα συναντώνται και στα συστήματα (RAID), που εξετάστηκαν στο Κεφάλαιο 2.6.

Τρίτον, η ανάπτυξη της μεθοδολογίας του αντικειμενοστραφούς προγραμματισμού (object oriented programming) άνοιξε ένα νέο πεδίο. Στις πρώτες γλώσσες προγραμματισμού ο μοναδικός τύπος ήταν οι πίνακες (μονοδιάστατοι και πολυδιάστατοι), ενώ τώρα όλες οι γλώσσες προσφέρουν και άλλους τύπους, όπως λίστες, εγγραφές, αρχεία, σύνολα κλπ. Οι αντικειμενοστραφείς γλώσσες δίνουν τη δυνατότητα ορισμού σύνθετων αντικειμένων και με βάση αυτά τη δυνατότητα ορισμού υπερ-αντικειμένων. Ο συγχερασμός της τεχνολογίας των βάσεων δεδομένων και της τεχνολογίας του αντικειμενοστραφούς προγραμματισμού έδωσε τη δυνατότητα να ανοίξει το πεδίο των αντικειμενοστραφών βάσεων δεδομένων. Στα συστήματα αυτά δημιουργούνται νέα προβλήματα σε σχέση με τη δημιουργία μεθόδων προσπέλασης, γιατί ο αριθμός των πεδίων του κάθε αντικειμένου είναι μεταβλητός και επομένως το μήκος της εγγραφής του κάθε αντικειμένου είναι επίσης μεταβλητό. Έτσι δεν είναι εύκολο να επιτευχθεί ο διαμερισμός ενός αρχείου αντικειμένων από δύο διαφορετικά προγράμματα. Επομένως, το 'πανταχού παρόν' B-δένδρο ή κάποιο άλλο δένδρο της ίδιας οικογένειας δεν είναι πρακτικά χρήσιμο αν εφαρμοσθεί με την κλασική του υλοποίηση. Ωστόσο, και στο σημείο αυτό έχουν προταθεί παραλλαγές του B-δένδρου για την ικανοποίηση των απαιτήσεων των συστημάτων αυτών.

Στο σημείο αυτό, αξίζει να σημειωθεί η ύπαρξη των λεγόμενων επεκτατών (extensible) συστημάτων βάσεων δεδομένων, που μελετήθηκαν στα τέλη της δεκαετίας του '80. Ο σκοπός των συστημάτων αυτών ήταν να καλύψουν νέες εφαρμογές, όπως στατιστικές εφαρμογές, επεξεργασία εικόνας, πολυμεσικά συστήματα και εφαρμογές γραφείου. Οι εφαρμογές αυτές απαιτούν την υποστήριξη νέων τύπων (όπως κείμενο, ψηφιακή φωνή, εικόνα και βίντεο), νέες συναρτήσεις, σύνθετα αντικείμενα, κανόνες ενεργοποίησης, αποθηκευτικές τεχνικές και μεθόδους προσπέλασης με μεγαλύτερες δυνατότητες μοντελοποίησης και επίδοσης. Κατά δεύτερο λόγο, τα εμπορικά συστήματα είναι πάντα μονολιθικά και δεν μπορούν εύκολα να παρακολουθήσουν τις εξελίξεις. Επομένως, μία ανοιχτή αρχιτεκτονική σε επίπεδο συστήματος διαχείρισης βάσεων δεδομένων θα μπορούσε να καταστήσει το σύστημα περισσότερο εύελιχτο και προσαρμόσιμο στην τεχνολογία και τις ανάγκες των χρηστών. Έτσι, στο παρελθόν υπήρξαν πολλά πρωτότυπα επεκτατά συστήματα, όπως Adt-Ingres, Rad, Dasdbs, Postgres, Probe, Genesis, Exodus, Starburst, R²D², Sabrina κλπ. Εκτός των άλλων, στα

συστήματα αυτά υπήρχε η δυνατότητα ορισμού νέων μεθόδων προσπέλασης από τον (έμπειρο) προγραμματιστή/αναλυτή μέσω καταλλήλων διεπιφανειών. Έτσι, οι νέες αυτές μέθοδοι θα μπορούσαν να τρέχουν επάνω από τα υπάρχοντα συστήματα. Τέλος, η νεότερη αντίληψη στην αρχιτεκτονική των συστημάτων διαχείρισης βάσεων δεδομένων είναι τα **αντικειμενοστραφή-σχεσιακά** (object-relational) συστήματα που συνδυάζουν τα πλεονεκτήματα των αντικειμενοστραφών συστημάτων (δηλαδή, χρήση νέων τύπων) με τα πλεονεκτήματα των σχεσιακών συστημάτων (δηλαδή, ισχυρές μηχανές επεξεργασίας ερωτήσεων). Τα συστήματα αυτά, ακολουθώντας τη φιλοσοφία των επεκτατών συστημάτων, παρέχουν δυνατότητες εμπλουτισμού με νέα υποσυστήματα, σχεδιασμένα για ειδικές εφαρμογές. Μία κλασική περίπτωση είναι η ανοιχτή αρχιτεκτονική του Informix με τις επεκτάσεις των Datablades, τα οποία διατίθενται από τρίτους βιομηχανικούς κατασκευαστές ή και ακαδημαϊκά-ερευνητικά ιδρύματα (με ελεύθερη πρόσβαση από το διαδίκτυο). Από την ίδια φιλοσοφία της ανοιχτής αρχιτεκτονικής διέπεται η Oracle που δέχεται τα Cartridges ως επεκτάσεις, και η DB2 της IBM που δέχεται τους λεγόμενους Extenders. Στο πλαίσιο αυτό αναφέρεται και η μέθοδος του **γενικευμένου δένδρου αναζήτησης** (generalized search tree, GiST), που προτάθηκε από τον Hellerstein (1995). Η δομή GiST είναι σημαντική γιατί κάτω από ένα ενιαίο πλαίσιο έχουν υλοποιηθεί το B⁺-δένδρο, το R-δένδρο, ενώ ακόμη μπορεί να υλοποιηθεί κάθε άλλη δομή.

Τέταρτον, σπουδαίο παρόν αλλά και μέλλον στη θεωρία και την πρακτική των βάσεων δεδομένων έχουν οι λεγόμενες **αποθήκες δεδομένων** (data warehouses). Η έννοια αυτή δηλώνει συστήματα που αποτελούν συλλογές δεδομένων και μετα-δεδομένων (δηλαδή, δεδομένων σχετικά με δεδομένα). Πρακτικά, οι αποθήκες αυτές χρησιμοποιούνται ως/σε **Συστήματα Λήψης Αποφάσεων** (Decision Support Systems) με τη βοήθεια τεχνικών on-line **αναλυτικής επεξεργασίας** (on-line analytical processing, OLAP). Ο τελευταίος όρος δηλώνει κάτι διαφορετικό από τον όρο on-line **επεξεργασία συναλλαγών** (on-line transaction processing, OLTP), που αναφέρεται σε παραδοσιακές βάσεις δεδομένων, όπου η αποτελεσματικότητα του συστήματος αποτιμάται με βάση των αριθμό των συναλλαγών στη μονάδα του χρόνου. Αντίθετα, ο όρος OLAP δεν αποσκοπεί στο να καταδείξει την αποδοτικότητα ενός συστήματος, καθώς η λήψη μίας απόφασης απαιτεί ιδιαίτερη προσοχή και σκέψη. Ένα σύστημα με δυνατότητες OLAP μπορεί να είναι μία καλή επέκταση ενός σχεσιακού συστήματος ή μπορεί να είναι ένας ειδικά κατασκευασμένος OLAP server. Οι OLAP servers σε επίπεδο καταλόγων χρησιμοποιούν τη μέθοδο των **κύβων δεδομένων** (data cubes).

Οι κύβιοι αυτοί είναι πολυδιάστατες προβολές μίας σχέσης, όπου υπολογίζονται όλοι οι δυνατοί συνδυασμοί τελεστών groupby (κατά SQL) και τα αποτελέσματα τοποθετούνται σε μία πολυδιάστατη δομή, για παράδειγμα σε ένα R-δένδρο. Μία άλλη οικογένεια δομών που έχουν προταθεί για χρήση σε συστήματα OLAP είναι οι κατάλογοι με δυαδική αναπαράσταση (bitmap indices), που μοιάζουν με τα κλασικά B⁺-δένδρα, αλλά με τη διαφορά ότι αποθηκεύουν τα κλειδιά στα φύλλα με δυαδική αναπαράσταση. Οι κατάλογοι αυτοί, υπό προϋποθέσεις, μπορεί να καταλάβουν τεράστιο χώρο, και για το λόγο αυτό έχουν προταθεί αρκετές δομές στην προσπάθεια για την εύρεση της πιο κομψής και αποτελεσματικής λύσης. Σημειώνεται ότι στο εμπόριο διατίθενται ήδη μηχανές OLAP, όπως ο Express Server της Oracle.

Πέμπτον, νέα ώθηση στη θεωρία και την τεχνολογία των βάσεων δεδομένων έδωσε η επέκταση του παγκόσμιου ιστού (web). Ο παγκόσμιος ιστός μπορεί να θεωρηθεί ως μία τεράστια βάση δεδομένων, όπου ο καθένας μπορεί να κάνει αναζητήσεις (καθώς και άλλες πράξεις που συναντώνται στις βάσεις δεδομένων, όπως συνδέσεις κλπ). Ωστόσο, το πρόβλημα σε ένα τέτοιο περιβάλλον είναι η παντελής έλλειψη πειθαρχίας ως προς τους τύπους των δεδομένων. Για παράδειγμα, ένας αριθμός μπορεί να είναι τύπου ακεραίου αλλά και τύπου συμβολοσειράς. Για το λόγο αυτό, λέγεται ότι τα δεδομένα είναι ημι-δομημένα (semistructured). Η έρευνα για τη δημιουργία μεθόδων προσπέλασης για ημι-δομημένα δεδομένα είναι σε πρώιμα στάδια. Βασικά, έχουν προταθεί διάφορων ειδών κατάλογοι για τους γνωστούς τύπους δεδομένων αλλά και κατάλογοι για την οργάνωση των συνδέσεων (link), που συνδέουν αντικείμενα, σελίδες και τοποθεσίες (cites) μεταξύ τους. Ήδη διατίθενται μέσω του ιστού αρκετά πρωτότυπα συστήματα όπως τα Araneus, Lore, Strudel, Tsimmis, UnQL, W3QS, WebLog, WebSQL κλπ. Ο σκοπός σε τέτοια συστήματα είναι η αναζήτηση δεδομένων μέσα σε αρχεία HTML, XML, POSTSCRIPT, BIBTEX, ASCII κλπ. σε συνδυασμό με την ανάκτηση εικόνων τύπου BMP, GIF, JPG, MPEG κλπ. Προς το παρόν δεν υπάρχουν αντίστοιχα βιομηχανικά εργαλεία στο εμπόριο.

Πιστεύεται ότι ο τόμος αυτός θα αποτελέσει ένα βοήθημα στην περιοχή της φυσικής σχεδίασης βάσεων δεδομένων και θα συντελέσει στην περαιτέρω εξέταση πολλών επίκαιρων προβλημάτων. Η μελέτη των μεθόδων προσπέλασης θα συνεχίσει να αποτελεί μία ενδιαφέρουσα περιοχή για μελέτη, σχεδίαση και ανάπτυξη.