



Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές τους στο Διαδίκτυο

σημειώσεις μαθητή

Γ' ΕΠΑ.Λ.

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΘΡΗΣΚΕΥΜΑΤΩΝ ΚΑΙ ΑΘΛΗΤΙΣΜΟΥ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

**Γιάτας Δ., Γώγουλος Γ., Κοτίνη Ι.,
Κυριακάκη Γ., Μωράκης Δ., Τζελέπη Σ., Φραγκονικολάκης Μ.**

Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές τους στο Διαδίκτυο

Γ' Τάξη ΕΠΑ.Λ.

Ειδικότητα: Τεχνικός Εφαρμογών Πληροφορικής

Τομέας Πληροφορικής

ΣΗΜΕΙΩΣΕΙΣ ΜΑΘΗΤΗ

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Πρόεδρος: *Σωτήριος Γκλαβάς*

ΓΡΑΦΕΙΟ ΕΡΕΥΝΑΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΕΦΑΡΜΟΓΩΝ Β΄

Προϊστάμενος: *Πάυλος Μάραντος*

ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υπεύθυνος: *Θεοδόσιος Τσαπέλας*, Σύμβουλος Β΄ Πληροφορικής ΙΕΠ

ΣΥΓΓΡΑΦΕΙΣ:

Γιάτας Δημήτρης, Καθηγητής Πληροφορικής

Γώγουλος Γιώργος, Σχολικός Σύμβουλος Πληροφορικής Δυτ. Κρήτης

Κοτίνη Ισαβέλλα, Σχολική Σύμβουλος Πληροφορικής Κ. Μακεδονίας

Κυριακάκη Γεωργία, Καθηγήτρια Πληροφορικής

Μωράκης Διονύσης, Καθηγητής Πληροφορικής

Τζελέπη Σοφία, Σχολική Σύμβουλος Πληροφορικής

Φραγκονικολάκης Μανόλης, Καθηγητής Πληροφορικής

ΣΥΝΤΟΝΙΣΤΗΣ

Γώγουλος Γιώργος, Σχολικός Σύμβουλος Πληροφορικής Δυτ. Κρήτης

ΕΠΙΤΡΟΠΗ ΚΡΙΣΗΣ:

Κωτσάκης Σταύρος, Σχ. Σύμβουλος Πληροφορικής-ΠΕ19

Μπαμπαλώνα Ελένη, Σχ. Σύμβουλος Πληροφορικής-ΠΕ19

Ταταράκη Αλεξάνδρα, Καθηγήτρια Πληροφορικής-ΠΕ19

Περιεχόμενα

1.	Από τα Δεδομένα στις Βάσεις Δεδομένων (ΒΔ)	5
1.1	Δεδομένα και Πληροφορίες	6
1.2	Σημασία της Διαχείρισης Δεδομένων	8
1.3	Εξέλιξη της Διαχείρισης Δεδομένων.....	10
1.4	Εφαρμογές Βάσεων Δεδομένων στην Καθημερινή Ζωή, σε Επιχειρήσεις και Οργανισμούς	14
2.	Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ)	17
2.1	Λειτουργίες ΣΔΒΔ	18
2.2	Χαρακτηριστικά ενός ΣΔΒΔ	19
2.3	Επισκόπηση της Εξέλιξης των ΣΔΒΔ.....	20
2.4	Αρχιτεκτονική και Συστατικά ΣΔΒΔ	22
2.5	Κατηγορίες ΣΔΒΔ	25
2.6	Διαδεδομένα ΣΔΒΔ	26
3.	Σχεδιασμός Βάσεων Δεδομένων.....	27
3.1	Εισαγωγή.....	28
3.1.1	Ανάλυση Απαιτήσεων	29
3.1.2	Εννοιολογικός, Λογικός και Φυσικός Σχεδιασμός.....	30
3.2	Μοντέλα Δεδομένων.....	31
3.2.1	Εννοιολογικό, Λογικό και Φυσικό Σχήμα (Μοντέλο)	32
3.2.2	Μοντέλο Οντοτήτων Συσχετίσεων	34
3.2.3	Σχεσιακό Μοντέλο	36
3.3	Διαγραμματική Αναπαράσταση Συσχετίσεων Οντοτήτων	39
3.3.1	Οντότητες, Στιγμιότυπα, Γνωρίσματα Ιδιότητες και Κλειδιά.....	40
3.3.2	Συσχετίσεις μεταξύ Οντοτήτων Τύποι Συσχετίσεων.....	43
3.3.3	Μοντελοποίηση Συσχετίσεων Οντοτήτων και ER Διάγραμμα	45
3.3.4	Επίλυση Συσχετίσεων Πολλά προς Πολλά	49
3.4	Μοναδικά Κλειδιά και Κανονικοποίηση (Normalization)	51
3.4.1	Πρωτεύον, Σύνθετο και Δευτερεύον κλειδί	52
3.4.2	Κανονικοποίηση και Πρώτη Κανονική Μορφή.....	54
3.5	Μετασχηματισμός από το Εννοιολογικό Μοντέλο στο Σχεσιακό Μοντέλο.....	57
3.5.1	Εισαγωγή στις έννοιες των Σχεσιακών Βάσεων Δεδομένων	58
3.5.2	Διαδικασία Μετασχηματισμού από το Εννοιολογικό Μοντέλο στο Σχεσιακό Μοντέλο.....	61

4.	Η Γλώσσα Βάσεων Δεδομένων SQL	64
	Εισαγωγή	65
4.1	SQL Σχεσιακή Γλώσσα Ορισμού Δεδομένων (DDL).....	67
4.1.1	Τύποι Δεδομένων της SQL	68
4.1.2	Δημιουργία Πίνακα	72
4.2	SQL Σχεσιακή Γλώσσα Χειρισμού Δεδομένων (DML).....	78
4.2.1	Εισαγωγή Δεδομένων σε Πίνακα Σχεσιακής ΒΔ με SQL	79
4.2.2	Ενημέρωση Εγγραφών Πίνακα Σχεσιακής ΒΔ	81
4.2.3	Διαγραφή Εγγραφών Πίνακα Σχεσιακής ΒΔ.....	83
4.2.4	Βασικές Ερωτήσεις Ανάκτησης.....	85
4.2.5	Ταξινόμηση Αποτελεσμάτων	93
4.2.6	Συναρτήσεις Συνάθροισης και Ομαδοποίησης	95
	ΠΑΡΑΡΤΗΜΑ 4.1: Οι Τύποι Δεδομένων της Γλώσσας SQL	99
	ΠΑΡΑΡΤΗΜΑ 4.2: Δημιουργία Πινάκων στο Σχεσιακό ΣΔΒΔ MySQL.....	100
5.	Διαχείριση Βάσεων Δεδομένων	105
5.1	Οι Ρόλοι του Διαχειριστή Βάσης Δεδομένων.....	106
5.1.1	Εγκατάσταση και Παραμετροποίηση του ΣΔΒΔ	107
5.1.2	Ορισμός του Σχήματος της ΒΔ.....	111
5.1.3	Δημιουργία Χρηστών και Απόδοση Ρόλων	113
5.1.4	Καθορισμός Μεθόδων Προσπέλασης.....	119
5.1.5	Εργασίες Συντήρησης και Αναβάθμισης.....	122
5.2	Εργαλεία Διαχείρισης ΒΔ	127
6.	Βάσεις Δεδομένων στο Διαδίκτυο	130
6.1	Νέες Ανάγκες και NoSQL Βάσεις Δεδομένων	131
6.1.1	Χαρακτηριστικά NoSQL Βάσεων Δεδομένων	132
6.1.2	Τύποι (Κατηγορίες) NoSQL Βάσεων Δεδομένων	133
6.2	Αρχιτεκτονική Εφαρμογών ΒΔ στο Διαδίκτυο.....	138
7.	Εφαρμογές Βάσεων Δεδομένων	139
7.1	Εισαγωγή.....	140
7.2	Διαδικτυακή Βάση Δεδομένων Διαθέσιμων Ραντεβού Ιατρών ΕΟΠΥΥ.....	141
7.3	Μελέτη Κοινωνικών Δικτύων ως προς τις Χρησιμοποιούμενες Τεχνολογίες σε ΒΔ.....	143
7.4	Ψηφιοποιημένο Αρχείο Εθνικού Θεάτρου	145
	Βιβλιογραφία, Αναφορές	147

κεφάλαιο

1

Από τα δεδομένα στις Βάσεις Δεδομένων (ΒΔ)

1.1 Δεδομένα και Πληροφορίες



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να διαχωρίζει την έννοια των δεδομένων από την έννοια της πληροφορίας
- να περιγράφει τη διαδικασία παραγωγής πληροφοριών από την επεξεργασία δεδομένων
- να προσδιορίζει τις διαδικασίες διαχείρισης των δεδομένων

Η σημερινή **εποχή της πληροφορίας** χαρακτηρίζεται από την δυνατότητα των ανθρώπων να ανταλλάσσουν και να μεταφέρουν πληροφορίες με μεγάλο βαθμό ελευθερίας έχοντας άμεση πρόσβαση σε γνώσεις που θα ήταν δύσκολο ή αδύνατο να αποκτηθούν στο παρελθόν. Κάθε άνθρωπος στην καθημερινή του ζωή διαχειρίζεται ένα πλήθος από πληροφορίες, οι οποίες τον καθοδηγούν να κατανοεί οτιδήποτε συμβαίνει στο ευρύτερο περιβάλλον του και να αντιμετωπίζει τα προβλήματα που τον απασχολούν παίρνοντας αποφάσεις και δίνοντας λύσεις. Στην ανθρώπινη καθημερινότητα σαν έκφραση η έννοια πληροφορία σημαίνει κυρίως ειδήσεις, γεγονότα, ιδέες, κάτι που μπορεί να αποκτηθεί και να θεωρηθεί γνώση. Στην αρχή του 20ου αιώνα όταν εμφανίστηκε και τεκμηριώθηκε η θεωρία των πληροφοριών, καθορίστηκε σαφώς και το πλαίσιο της με αυστηρούς μαθηματικούς νόμους. Οι εφαρμογές στη μετάδοση των εικόνων, στη σχεδίαση συστημάτων ανίχνευσης πλοίων αεροπλάνων κλπ, στη συλλογή στοιχείων από δορυφόρους και διστημόπλοια, στην επικοινωνία υπολογιστών και κινητών τηλεφώνων, αποτελούν μερικές μόνο από τις επιτυχημένες πρακτικές της θεωρίας των πληροφοριών. Καταλαβαίνουμε με αυτό τον τρόπο και τη διαφορετική προσέγγιση του θεμελιώδους όρου πληροφορία όσον αφορά την καθημερινή μας ζωή και την παιδεία στις διάφορες επιστήμες. Από τους ορισμούς που υπάρχουν στη διεθνή βιβλιογραφία θα υιοθετήσουμε τον ορισμό της Αμερικανικής Επιτροπής Εθνικών Προτύπων (ANSI American National Standards Institute) που αναφέρει ότι πληροφορία είναι η σημασία που έχουν για τον άνθρωπο τα στοιχεία, από τα οποία αυτή αποτελείται. Ειδικότερα:

Στοιχεία πληροφορίας ή Δεδομένα (DATA) είναι οποιαδήποτε παράσταση, όπως χαρακτήρες ή αριθμητικές ποσότητες, σύμβολα κτλ., στην οποία δίνεται ή είναι δυνατόν να δοθεί μια σημασία (έννοια).

Πληροφορία είναι η σημασία που δίνει ο άνθρωπος σε ένα σύνολο δεδομένων, τα οποία επεξεργάζεται με τη βοήθεια προκαθορισμένων συμφωνιών που έχουν θεσπιστεί από τον ίδιο.

Από τους παραπάνω ορισμούς σαν εύκολο συμπέρασμα προκύπτει ότι τα δεδομένα μπορεί να είναι αποσπασματικά και ακατέργαστα. Όμως, η **συλλογή και ο συσχετισμός** των δεδομένων δίνει ως αποτέλεσμα την πληροφορία. Ο συσχετισμός δεδομένων πρέπει να ακολουθεί ένα αυστηρό και συγκεκριμένο πλαίσιο από κανόνες, π.χ. γραμματικούς για τη γλώσσα, διαχωρισμό των bytes για τα δυαδικά ψηφία κ.λ.π. Αυτό το πλαίσιο από κανόνες είναι που στον ορισμό αναφέρονται ως προκαθορισμένες, από τους ανθρώπους, συμφωνίες.

Οι περιορισμένες δυνατότητες της ανθρώπινης μνήμης προσανατόλισε τον άνθρωπο από αρχαιοτάτων χρόνων να δημιουργήσει μεθόδους μόνιμης αποθήκευσης των δεδομένων. Η συλλογή δεδομένων είχε καθορισμένο στόχο ή αφορούσε συγκεκριμένο αντικείμενο για θέματα που η εκάστοτε κοινότητα χρειαζόταν ή υπήρχε περίπτωση να χρειαστεί.

Το ερώτημα βέβαια είναι αν η παραπάνω διαδικασία αρκεί για να ολοκληρωθεί η υποβοήθηση των δραστηριοτήτων που επιβάλλουν την χρήση της. Φανταστείτε, π.χ., ένα τηλεφωνικό κατάλογο μιας πόλης 100.000 κατοίκων, στον οποίο τα ονόματα των συνδρομητών δεν θα ήταν ταξινομημένα αλφαβητικά, αλλά θα γράφονταν με τυχαίο τρόπο. Κανείς δεν θα ήταν διατεθειμένος να διαβάσει ολόκληρο τον κατάλογο, προκειμένου να βρει ένα τηλέφωνο επομένως η ύπαρξη του καταλόγου θα ήταν άχρηστη.

Η επόμενη κίνηση επομένως, μετά την συλλογή και καταγραφή των δεδομένων ήταν να εφευρεθούν τρόποι αποδοτικής οργάνωσης αυτών. Οι πρώτες προσπάθειες που αναφέρονται ιστορικά έχουν να κάνουν με χειρογραφικά συστήματα αποθήκευσης και οργάνωσης δεδομένων. Οι **βασικές λειτουργίες**, που τα συστήματα αυτά έδιναν στους χρήστες τους ήταν:

- **Εισαγωγή νέων δεδομένων**
- **Διαγραφή δεδομένων**
- **Επαναφορά δεδομένων**
- **Ενημέρωση δεδομένων**



Ερωτήσεις / Δραστηριότητες

1. Πώς αντιλαμβάνεστε τις έννοιες δεδομένα και πληροφορία; Δώστε ένα δικό σας παράδειγμα.
2. Περιγράψτε πώς οι συλλογές δεδομένων μπορούν να γίνουν πρακτικά χρήσιμες.

1.2 Σημασία της Διαχείρισης Δεδομένων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει τα βασικά μέρη ενός συστήματος Βάσης Δεδομένων
- να προσδιορίζει τις διαδικασίες διαχείρισης των δεδομένων
- να αναγνωρίζει τους διαφορετικούς τύπους δεδομένων

Τα δεδομένα στις μέρες μας καταγράφονται και χρησιμοποιούνται σε κάθε δραστηριότητα επιστημονική, εμπορική, παραγωγική, κυβερνητική, στρατιωτική κλπ. Οι μεγάλες ποσότητες δεδομένων επιβάλλουν την εύρεση αποτελεσματικών μεθόδων αποθήκευσης. Τα δεδομένα πρέπει να είναι οργανωμένα με τέτοιο τρόπο ώστε να διευκολύνεται η αναζήτηση και η ενημέρωσή τους και αυτό ακριβώς είναι το αντικείμενο της μελέτης μας, που απαιτεί την καλύτερη λύση.

Ένας από τους βασικούς λόγους για τους οποίους αναθέτουμε σε υπολογιστές να επιλύουν προβλήματα είναι η δυνατότητα τους να αποθηκεύουν μεγάλο όγκο δεδομένων με ταχύτητα ακρίβεια και ασφάλεια. Αυτό είχε ως αποτέλεσμα να αναπτυχθούν συστήματα όπου τα δεδομένα αποθηκεύονταν με τη **μορφή αρχείων** σε ψηφιακά μέσα, όπως μαγνητικές ταινίες και μαγνητικούς δίσκους, με τη χρήση κατάλληλων **εφαρμογών** λογισμικού. Οι εφαρμογές αυτές, εκτός από δυνατότητες εισαγωγής και αποθήκευσης δεδομένων στο σύστημα, έδιναν επιπλέον στον χρήστη τη ευκαιρία να ανακαλεί, ενημερώνει, διαγράφει και γενικά να επεξεργάζεται και να αξιοποιεί με πολλούς τρόπους τα δεδομένα των αρχείων. Η διαχείριση των δεδομένων ανήκε στην ευθύνη του προγραμματιστή της εφαρμογής.

Ας αναλύσουμε εδώ ένα παράδειγμα πελατολογίου και παραγγελιών μιας εμπορικής επιχείρησης για να μελετήσουμε την δημιουργία και χρήση των αρχείων. Για την σωστή οργάνωση του αρχείου μας, θα πρέπει να δημιουργήσουμε καρτέλες αρχικά για τους πελάτες και ακολούθως για τις παραγγελίες τους. Περιεχόμενο (ενδεικτικό) των καρτελών θα πρέπει να είναι τα εξής στοιχεία ανά πελάτη:

- | | | |
|-----------|-------------|------------|
| • Κωδικός | • Διεύθυνση | • Τηλέφωνο |
| • Επώνυμο | • TK | • ΑΦΜ |
| • Όνομα | • Πόλη | • ΔΟΥ |

Μπορούμε εδώ να ισχυριστούμε ότι τελικά ένα αρχείο είναι ένα σύνολο οργανωμένων ομοειδών στοιχείων. Μέσα σε ένα αρχείο τα στοιχεία οργανώνονται σε **λογικές ενότητες**. Το σύνολο των στοιχείων που περιέχονται σε μια λογική ενότητα ονομάζεται **εγγραφή** (record). Το κάθε στοιχείο της εγγραφής που καταχωρίζεται στο αρχείο ονομάζεται **πεδίο** (field). Στο παραπάνω παράδειγμα το αρχείο της εταιρείας περιέχει στοιχεία που αφορούν όλους τους πελάτες της εταιρείας και για κάθε πελάτη δημιουργείται μια εγγραφή που περιέχει συγκεκριμένα στοιχεία γι' αυτόν. Δηλαδή, το αρχείο περιέχει τόσες εγγραφές όσοι είναι και οι πελάτες της εταιρείας. Ορίζοντας καλύτερα τώρα τις έννοιες μπορούμε να δώσουμε τους παρακάτω ορισμούς.

- **Αρχείο** (file) είναι ένα σύνολο εγγραφών λογικά συνδεδεμένων μεταξύ τους, που είναι καταχωρισμένες σε ένα (ή και περισσότερα, σύμφωνα με τις δυνατότητες της σημερινής τεχνολογίας) μαγνητικό μέσο αποθήκευσης.
- **Εγγραφή** (record) του αρχείου είναι το σύνολο των πεδίων που ανήκουν στην ίδια λογική ενότητα.
- **Πεδίο** (field) είναι ένα από τα επιμέρους στοιχείαπληροφορίες που συνθέτουν την εγγραφή, όπως αυτή θα καταχωριστεί σε ένα αρχείο.

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

- Συρτάρι – Αρχείο Δεδομένων
- Καρτέλα πελάτη – Εγγραφή του αρχείου δεδομένων Στοιχείο της καρτέλας – Πεδίο της εγγραφής
- Το πεδίο, καθώς γίνεται η επεξεργασία του από τον Η/Υ, δεσμεύει κάποιο χώρο στην κεντρική ή στη δευτερεύουσα μνήμη του υπολογιστή, χώρο ο οποίος μετρείται σε bytes. Ο αριθμός των απαιτούμενων bytes είναι το **μήκος του πεδίου** (field length). Με βάση τις τιμές που μπορούν να δεχτούν, μπορούμε να διακρίνουμε τα πεδία σε διάφορες κατηγορίες. Οι κυριότερες από αυτές αναφέρονται παρακάτω:

- **Αλφαριθμητικά** (alphanumeric), όταν περιέχουν αλφαριθμητικούς χαρακτήρες, ειδικούς χαρακτήρες, αριθμούς ή συνδυασμούς αυτών.
- **Αριθμητικά** (numeric), όταν περιέχουν μόνο αριθμούς.
- **Αλφαβητικά** (alphabetic), όταν περιέχουν μόνο αλφαριθμητικούς χαρακτήρες. Ημερομηνίας (date), όταν έχουν τη δυνατότητα αποθήκευσης ημερομηνιών με διάφορες μορφές (π.χ., 121298 ή 12031998 κ.ο.κ., ανάλογα με το προγραμματιστικό περιβάλλον που χρησιμοποιείται).
- **Διαδικά** (binary), όταν είναι κατάλληλα να αποθηκεύουν ειδικού τύπου δεδομένα, όπως εικόνα, ήχο, video κτλ.).
- **Λογικά** (logical), όπου επιτρέπονται μόνο δύο τιμές οι οποίες αντιστοιχούν σε δύο διακριτές καταστάσεις: Αλήθεια (True) ή Ψέμα (False).
- **Σημειώσεις** (memo), όπου είναι επιτρεπτή η εισαγωγή κειμένου με μεταβλητό μήκος, το οποίο, συνήθως, χρησιμοποιείται για την αποθήκευση σημειώσεων (κειμένου), που μπορεί, ανάλογα και με το περιβάλλον ανάπτυξης, να είναι και αρκετά μεγάλο.
- **Μήκος εγγραφής** (record length) είναι το άθροισμα του μήκους των πεδίων που την αποτελούν.
- **Δομή εγγραφής** (ή γραμμογράφηση) (record layout) είναι ο τρόπος με τον οποίο οργανώνονται τα πεδία της. Το βασικό στοιχείο που καθορίζει τη δομή εγγραφής είναι η σειρά με την οποία εμφανίζονται τα πεδία μέσα στην εγγραφή, δηλαδή ποιο είναι πρώτο, ποιο δεύτερο κ.ο.κ.

Τα προγράμματα εφαρμογών διαχείρισης των αρχείων διαβάζουν ή γράφουν τα δεδομένα στα αρχεία με τη μορφή των εγγραφών. **Διάβασμα** (read) από αρχείο σημαίνει, στη γλώσσα των υπολογιστών, ότι το πρόγραμμα πηγαίνει στο μέσο αποθήκευσης του αρχείου, παίρνει τα δεδομένα (συνήθως ανά μία εγγραφή) και τα φέρνει στην κεντρική μνήμη του υπολογιστή, για να τα επεξεργαστεί σύμφωνα με το σχεδιάσμά του. **Γράψιμο** (write) στο αρχείο σημαίνει, ότι το πρόγραμμα επεξεργάζεται δεδομένα που βρίσκονται στην κεντρική μνήμη του υπολογιστή και στη συνέχεια δίνει εντολή για αποθήκευση, μεταβολή ή και διαγραφή εγγραφών στο αρχείο. Με τις διαδικασίες αυτές υλοποιούνται οι βασικές εργασίες που παρουσιάστηκαν στην παράγραφο 1.1.



Ερωτήσεις / Δραστηριότητες

1. Ποιες είναι οι βασικές δυνατότητες που παρέχουν στους χρήστες τους τα συστήματα οργάνωσης δεδομένων;
2. Πώς αντιλαμβάνεστε τις έννοιες αρχείο, εγγραφή και πεδίο στα συστήματα “ηλεκτρονικής” τήρησης δεδομένων σε αντιδιαστολή με τα χειρογραφικά συστήματα της προηγούμενης παραγράφου;
3. Αναφέρετε τις κύριες κατηγορίες των πεδίων στα αρχεία εγγραφών.

1.3 Εξέλιξη της Διαχείρισης Δεδομένων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να αναγνωρίζει τα πλεονεκτήματα της οργάνωσης δεδομένων σε βάσεις δεδομένων αντί αρχείων
- να αντιλαμβάνεται την έννοια του μοντέλου και των δομικών του στοιχείων
- να κατανοεί πώς ο κόσμος αναπαριστάται από σύνολα οντοτήτων και συσχετίσεων

Τα πρώτα συστήματα εφαρμογών ηλεκτρονικής τήρησης δεδομένων, που χρησιμοποιήθηκαν (και πιθανόν χρησιμοποιούνται ακόμα) για την αποθήκευση και εκμετάλλευση πληροφοριακών δεδομένων, ήταν οργανωμένα σε ανεξάρτητα και μεμονωμένα αρχεία. Τα συστήματα αυτά αναπαράγουν, ως ένα βαθμό, τη λογική οργάνωσης των πληροφοριών των αντίστοιχων χειρογραφικών συστημάτων και ταυτόχρονα την επεκτείνουν, αξιοποιώντας τις τεράστιες δυνατότητες που παρέχει η τεχνολογία των Υπολογιστών. Ο τρόπος αυτός διαχείρισης, αν και δίνει στον προγραμματιστή την ευελιξία να υλοποιήσει τις μεθόδους επεξεργασίας που αυτός επιθυμεί, ωστόσο δημιουργεί προβλήματα καθώς αυξάνεται ο όγκος των δεδομένων και η πολυπλοκότητα της πληροφορίας.

Χαρακτηριστικό των εφαρμογών αυτών ήταν ότι για να μπορέσουν να επικοινωνήσουν με τα αρχεία, έπρεπε να γνωρίζουν τον ακριβή τρόπο οργάνωσης των δεδομένων μέσα σ' αυτά. Για να μπορέσει επομένως μια εφαρμογή να διαβάσει τα δεδομένα ενός αρχείου, να εγγράψει νέα δεδομένα, να μεταβάλει ή διαγράψει κάποια από αυτά, πρέπει να περιλαμβάνει στον κώδικά της ένα τμήμα που να περιγράφει με ποια σειρά αποθηκεύονται τα δεδομένα στο αρχείο, πώς αναπαριστούνται, πόσο χώρο καταλαμβάνει το καθένα στο φυσικό μέσο κτλ. Συμπερασματικά θα μπορούσαμε να ισχυριστούμε ότι υπήρχε ανάγκη συμβατότητας της λογικής συγγραφής του κώδικα της εφαρμογής με τον τρόπο οργάνωσης των δεδομένων των αρχείων με τα οποία επικοινωνεί.

Αυτό το είδος της απευθείας επικοινωνίας της εφαρμογής με τα αρχεία δεδομένων, την οποία εξασφαλίζει ο κώδικας της κάθε εφαρμογής ξεχωριστά, ονομάζεται φυσική διεπαφή (interface) της εφαρμογής με το αρχείο. Έτσι, κάθε εφαρμογή μπορεί να επικοινωνήσει μόνο με τα δικά της αρχεία. Από τα παραπάνω γίνεται φανερό ότι στα συστήματα αυτά οι εφαρμογές είναι σε μεγάλο βαθμό εξαρτημένες από τα δεδομένα (datadependent).

Ας εξετάσουμε συνολικά μια περίπτωση που ξεφεύγει από μια απλή διαχείριση αριθμητικών δεδομένων, όπου υπάρχει η ανάγκη για αποθήκευση και διαχείριση μερικών εκατοντάδων αριθμών, πρόβλημα που μπορεί να υλοποιηθεί εύκολα αποθηκεύοντας τους αριθμούς σε ένα αρχείο του λειτουργικού συστήματος, και υλοποιώντας τις απαραίτητες λειτουργίες υπολογισμών (π.χ., μέσος όρος, ελάχιστη τιμή, μέγιστη τιμή) στη λογική της εφαρμογής.

Θεωρούμε μία περισσότερο πολύπλοκη εφαρμογή που έχει υλοποιηθεί για να υποστηρίξει την οργάνωση μίας ογκώδους ηλεκτρονικής βιβλιοθήκης. Στην βιβλιοθήκη αυτή απαιτείται η αναζήτηση άρθρων ή βιβλίων με βάση το συγγραφέα, τον τίτλο, την ημερομηνία έκδοσης, ενώ ακόμη απαιτείται η αναζήτηση άρθρων με βάση λέξεις-κλειδιά (keywords) από όλο το κείμενο. Ο προγραμματιστής για την ολοκλήρωση της εφαρμογής χρειάζεται να λάβει υπ' όψη ότι:

- Το πλήθος των άρθρων ή των βιβλίων θα είναι μεγάλο, με αποτέλεσμα να επιβάλλεται η χρήση αποδοτικών μεθόδων αναζήτησης.
- Η δομή της πληροφορίας είναι περίπλοκη, αφού ένας συγγραφέας μπορεί να έχει συμβάλει στη συγγραφή πολλών άρθρων, ενώ στη συγγραφή ενός άρθρου μπορεί να έχουν συμβάλει πολλοί συγγραφείς.

- Πρέπει να βρεθεί ένας αποτελεσματικός και αποδοτικός τρόπος αναζήτησης των λέξεων μέσα στο κείμενο. Μεγάλο ρόλο για την επιλογή θα έχει στην περίπτωση αυτή το κόστος της εφαρμογής.
- Υπάρχει η ανάγκη για εισαγωγή νέων στοιχείων, όπως για παράδειγμα η ενημέρωση των δεδομένων όταν υπάρχει ένα νέο άρθρο που πρέπει να καταχωρισθεί.
- Αναμένεται ότι πολλοί χρήστες θα χρησιμοποιούν την εφαρμογή ταυτόχρονα, με αποτέλεσμα να απαιτούνται ειδικοί μηχανισμοί προσπέλασης και συγχρονισμού των προσπελάσεων από διαφορετικούς χρήστες.

Ας εξετάσουμε ποια είναι μερικά από τα βασικά μειονεκτήματα της αποθήκευσης και διαχείρισης των δεδομένων απ' ευθείας σε αρχεία του λειτουργικού συστήματος.

- Ο ομάδα προγραμματιστών της εφαρμογής είναι υπεύθυνη για τη σωστή ενημέρωση των αρχείων των δεδομένων, αναλόγως με τις εισαγωγές και διαγραφές που πραγματοποιούνται. Για παράδειγμα, έστω ότι στο αρχείο A1 αποθηκεύονται τα στοιχεία ενός άρθρου (π.χ. τίτλος, αριθμός σελίδων), ενώ στο αρχείο A2 αποθηκεύονται τα ονόματα των συγγραφέων κάθε άρθρου. Κατά την εισαγωγή ενός νέου άρθρου θα πρέπει να ενημερωθεί το αρχείο A1 με τα απαραίτητα στοιχεία, ενώ το αρχείο A2 θα πρέπει να ενημερωθεί με τα στοιχεία των συγγραφέων του άρθρου. Επίσης, κατά την αναζήτηση στοιχείων θα πρέπει σε πρώτο στάδιο να προσδιορισθούν τα απαιτούμενα αρχεία, ενώ σε δεύτερο στάδιο πρέπει να γίνει προσεκτική ανάγνωση των δεδομένων, συνδυάζοντας κατάλληλα τις πληροφορίες που περιέχουν τα διαφορετικά αρχεία. Η κατάσταση δυσκολεύει όσο αυξάνεται ο αριθμός των αρχείων δεδομένων και όσο αυξάνεται η πολυπλοκότητα των ερωτημάτων.
- Η δομή του κάθε αρχείου καθορίζεται από την ίδια την εφαρμογή. Αυτό έχει ως αποτέλεσμα να είναι δύσκολη η αλλαγή της δομής ενός αρχείου. Πολλές φορές παρουσιάζεται η ανάγκη να προστεθεί ένα νέο χαρακτηριστικό στα δεδομένα. Για παράδειγμα, μπορεί να απαιτηθεί η προσθήκη στο αρχείο A1 ενός ακόμη πεδίου που να περιέχει το πλήθος των βιβλιογραφικών αναφορών (citations) που περιέχονται στο άρθρο. Σε μία τέτοια περίπτωση πρέπει απ' ενός να μεταβληθεί η δομή του αρχείου A1, έτσι ώστε να δημιουργηθεί χώρος για το νέο πεδίο, και απ' ετέρου να μεταβληθεί η λογική της εφαρμογής ώστε να ληφθεί το νέο πεδίο υπ' όψη κατά την αναζήτηση, εισαγωγή και διαγραφή δεδομένων. Οι αλλαγές αυτές είναι χρονοβόρες και επικίνδυνες για την ακεραιότητα των αποθηκευμένων δεδομένων.
- Εφ' όσον ο τρόπος δημιουργίας των αρχείων καθορίζεται από την εφαρμογή, η μορφοποίηση των αρχείων εξαρτάται από την εκάστοτε γλώσσα προγραμματισμού που χρησιμοποιεί ο προγραμματιστής (όπως C, C++, Java, Visual Basic). Το γεγονός αυτό δημιουργεί δυσκολίες όταν απαιτείται η χρήση των αρχείων από διαφορετικές εφαρμογές ή όταν απαιτείται η κατασκευή των αρχείων από άλλους προγραμματιστές που χρησιμοποιούν διαφορετικές γλώσσες προγραμματισμού σε διαφορετικά λειτουργικά συστήματα (όπως Linux, Windows, MacOS).
- Σε πολλές περιπτώσεις η εφαρμογή εξυπηρετεί πολλούς χρήστες ταυτόχρονα. Η διασφάλιση της προστασίας και της ακεραιότητας των δεδομένων κατά την ταυτόχρονη αναζήτηση, εισαγωγή και διαγραφή στοιχείων είναι αρκετά περίπλοκη. Επίσης, σε συγκεκριμένες λειτουργίες πρέπει να διασφαλίζεται η ατομικότητα της εκτέλεσης. Με τον όρο "ατομικότητα" εννοούμε ότι ένα σύνολο λειτουργιών είτε θα εκτελεσθεί συνολικά, είτε θα απορριφθεί συνολικά. Χαρακτηριστικό παράδειγμα αποτελεί η ανάληψη χρημάτων από μία τραπεζική αυτόματη ταμειακή μηχανή (ATM). Εδώ διακρίνουμε δύο ξεχωριστές λειτουργίες:
 - α) ανάληψη ενός ποσού και
 - β) ενημέρωση του λογαριασμού με το νέο υπόλοιπο. Αν αποτύχει μία από τις δύο λειτουργίες, τότε θα αποτύχει και το σύνολό τους. Σε διαφορετική περίπτωση θα δημιουργηθεί σοβαρό πρόβλημα ακεραιότητας δεδομένων καθώς τα δεδομένα του συστήματος θα είναι λανθασμένα (π.χ., ενώ έγινε η ανάληψη δεν αφαιρέθηκε το ποσό από το υπόλοιπο του λογαριασμού).

- Στη γενική περίπτωση, κάθε χρήστης της εφαρμογής έχει διαφορετικά δικαιώματα ως προς την προσπέλαση των δεδομένων. Για παράδειγμα, μία ομάδα χρηστών μπορεί να έχει μόνο δικαιώματα ανάγνωσης των δεδομένων, ενώ μία άλλη ομάδα χρηστών μπορεί να έχει δικαιώματα ανάγνωσης και εισαγωγής νέων δεδομένων αλλά όχι διαγραφής. Η εφαρμογή των κανόνων προστασίας των δεδομένων ώστε να αποφεύγονται δυσάρεστες καταστάσεις απώλειας δεδομένων (είτε κατά λάθος, είτε εσκεμμένα), στηριζόμενοι στα εργαλεία του λειτουργικού συστήματος, επιτυγχάνεται δύσκολα. Επιπλέον, είναι πολύ δύσκολο να ορίσουμε διαφορετικά δικαιώματα μέσα στο ίδιο αρχείο.

Εφ' όσον η λογική της αναζήτησης στοιχείων από τα αρχεία δεδομένων υλοποιείται στον κώδικα της εφαρμογής, οι διαφορετικοί τύποι ερωτημάτων που μπορεί να δεχθεί η εφαρμογή είναι περιορισμένοι και πρέπει να υλοποιηθούν εκ των προτέρων. Αυτό αποτελεί αρκετά δεσμευτικό παράγοντα λαμβάνοντας υπ' όψη ότι είναι σχεδόν αδύνατο να προβλέψουμε όλα τα δυνατά ερωτήματα που μπορεί να επιθυμεί να θέσει ένας χρήστης. Η υποστήριξη νέων ερωτημάτων απαιτεί αλλαγή στον κώδικα της εφαρμογής, κάτι που γενικά δεν είναι επιθυμητό διότι οδηγεί σε νέες εκδόσεις (versions) του λογισμικού.

Συνοψίζοντας μπορούμε να πούμε ότι εν γένει τα προβλήματα που ήταν αναπόφευκτο να δημιουργηθούν από την μεθοδολογία και πρακτική της χρήσης αρχείων για την ηλεκτρονική τήρηση δεδομένων είναι τα εξής:

Πλεονασμός των δεδομένων (data redundancy). Υπάρχει η περίπτωση να έχουμε επανάληψη των ίδιων δεδομένων σε αρχεία διαφορετικών εφαρμογών. Για παράδειγμα στο αρχείο των άρθρων και στο αρχείο των συγγραφέων είναι σχεδόν σίγουρο ότι θα υπάρχουν κάποια κοινά στοιχεία.

Ασυνέπεια των δεδομένων (data inconsistency). Αυτό μπορεί να συμβεί όταν υπάρχουν τα ίδια στοιχεία των συγγραφέων (πλεονασμός) και στο αρχείο άρθρων και στο αρχείο συγγραφέων και χρειασθεί να γίνει κάποια αλλαγή στη διεύθυνση ή στα τηλέφωνα κάποιου συγγραφέα, οπότε είναι πολύ πιθανό να γίνει η διόρθωση μόνο στο ένα αρχείο και όχι και στο άλλο.

Αδυναμία μερισμού δεδομένων (data sharing). Μερισμός δεδομένων σημαίνει δυνατότητα για κοινή χρήση των στοιχείων κάποιων αρχείων. Για παράδειγμα, ο μερισμός δεδομένων θα ήταν χρήσιμος αν με την αναζήτηση ενός άρθρου μπορούμε να έχουμε πρόσβαση την ίδια στιγμή στο αρχείο άρθρων για να δούμε τους συγγραφείς και μετά στο αρχείο των συγγραφέων για να ελέγξουμε αν είναι διαθέσιμα και άλλα άρθρα των ίδιων συγγραφέων. Η αδυναμία μερισμού δεδομένων δημιουργεί καθυστέρηση στη λήψη αποφάσεων και στην εξυπηρέτηση των χρηστών.

Αδυναμία προτυποποίησης. Έχει να κάνει με την ανομοιομορφία και με την διαφορετική αναπαράσταση και οργάνωση των δεδομένων στα αρχεία των εφαρμογών. Η αδυναμία αυτή δημιουργεί προβλήματα προσαρμογής των χρηστών καθώς και προβλήματα στην ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.

Οι **βασικοί παράγοντες** οι οποίοι οδηγούν στους παραπάνω περιορισμούς αναφορικά με την προσέγγιση των συστημάτων που βασίζονται σε αρχεία είναι:

1. Ο ορισμός των δεδομένων είναι ενσωματωμένος στα προγράμματα των εφαρμογών, αντί να αποθηκεύεται κάπου ξεχωριστά και ανεξάρτητα.
2. Δεν υπάρχει κανένας έλεγχος για την προσπέλαση και τον χειρισμό των δεδομένων, πέρα από αυτόν που επιβάλλεται από τα προγράμματα των εφαρμογών.

Εδώ προκύπτει ένα νέο σημαντικό ερώτημα. **Τι περισσότερο χρειαζόμαστε από αυτά που μας προσφέρει ένα σύστημα αρχείων;** Τα αποτελέσματα των απαντήσεων εστιάζονται στους εξής άξονες:

- Απλές και αποδοτικές ερωτήσεις που μπορεί να γίνονται ανάλογα με το θέμα ή πρόβλημα που προκύπτει από την χρήση των δεδομένων.
- Έλεγχος συγχρονισμού.

- Γρήγορη ανάκαμψη από κάθε πρόβλημα.
- Οικειοποίηση από τα οφέλη της μοντελοποίησης.

Για μεγαλύτερη αποτελεσματικότητα επομένως σε αυτό το ταχύτατα αναπτυσσόμενο πεδίο υπήρχε η ανάγκη μια νέας προσέγγισης. Το νέο που προέκυψε σαν φυσική εξέλιξη της διαχείρισης δεδομένων με τα αρχεία, απαντώντας ταυτόχρονα με επιτυχία στους παραπάνω άξονες, ήταν οι **Βάσεις Δεδομένων** και τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) (Database Management Systems (DBMS)).

Μια **Βάση Δεδομένων ή ΒΔ** θα μπορούσε να χαρακτηριστεί ως ένα σύνολο αρχείων τα οποία διαθέτουν υψηλό βαθμό οργάνωσης και είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες.

Τα αρχεία δε δημιουργούνται πλέον ούτε ενημερώνονται από ανεξάρτητες εφαρμογές λογισμικού αλλά από ένα ξεχωριστό σύστημα προγραμμάτων (λογισμικό). Το σύστημα αυτό μεσολαβεί ανάμεσα στα αρχεία δεδομένων και στις εφαρμογές που χρησιμοποιούν οι χρήστες και λέγεται Σύστημα Διαχείρισης Βάσης Δεδομένων ή ΣΔΒΔ (Data Base Management System - DBMS).

Το **ΣΔΒΔ** είναι ένα σύνολο προγραμμάτων και ρουτινών, που σκοπό έχουν το χειρισμό της βάσης, όσον αφορά τη δημιουργία, συντήρηση, επεξεργασία στοιχείων, ελέγχους ασφαλείας κτλ., και την εξυπηρέτηση των χρηστών, όσον αφορά την παροχή στοιχείων και πληροφοριών, χωρίς αυτοί να πρέπει να ασχολούνται με το πώς και το πού τα δεδομένα είναι αποθηκευμένα στη βάση. Θα ασχοληθούμε με μεγαλύτερη λεπτομέρεια στο επόμενο κεφάλαιο με την έννοια του ΣΔΒΔ.



Ερωτήσεις / Δραστηριότητες

1. Ποια θεωρείτε σαν βασικά μειονεκτήματα της αποθήκευσης και διαχείρισης των δεδομένων απ' ευθείας σε αρχεία του λειτουργικού συστήματος;
2. Ποιοι είναι συνοπτικά οι βασικοί παράγοντες που οδηγούν στους περιορισμούς αναφορικά με την προσέγγιση των συστημάτων που βασίζονται σε αρχεία;
3. Τι περισσότερο χρειαζόμαστε από αυτά που μας προσφέρει ένα σύστημα αρχείων;

1.4 Εφαρμογές Βάσεων Δεδομένων στην Καθημερινή Ζωή, σε Επιχειρήσεις και Οργανισμούς



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να αναφέρει παραδείγματα εφαρμογών και υπηρεσιών για πολίτες και οργανισμούς/ επιχειρήσεις που βασίζονται σε βάσεις δεδομένων
- να ευαισθητοποιηθεί για τη σπουδαιότητα που έχει η διαχείριση των δεδομένων για τα άτομα και τους οργανισμούς/επιχειρήσεις
- να αντιλαμβάνεται την σημαντικότητα της διαχείρισης των δεδομένων που αφορούν προσωπικές και ευαίσθητες πληροφορίες

Κρατήσεις θέσεων σε αεροπορικές εταιρείες. Οι εφαρμογές αυτές υποστηρίζουν λειτουργίες, όπως κράτηση θέσης για μία συγκεκριμένη πτήση, αναζήτηση πληροφοριών για διαθέσιμες πτήσεις με βάση την αφετηρία και τον προορισμό, αναζήτηση πληροφοριών σχετικά με τις τιμές και τη διαθεσιμότητα των εισιτηρίων. Η λειτουργία της εφαρμογής επιτρέπει την απάντηση ερωτημάτων που αφορούν στην ώρα αναχώρησης και άφιξης συγκεκριμένων πτήσεων και την αποδοτική κράτηση θέσεων και έλεγχο της διαθεσιμότητας. Οι εφαρμογές αυτές είναι πολύτιμες για ταξιδιωτικούς πράκτορες και αεροπορικές εταιρείες λόγω των πολλών διευκολύνσεων που παρέχουν.

Τραπεζικές Συναλλαγές. Στις εφαρμογές αυτές η πληροφορία αποτελείται από ονόματα πελατών, διευθύνσεις, αριθμούς τραπεζικών λογαριασμών, υπόλοιπο λογαριασμών, δεδομένα που αφορούν σε δάνεια, πιστωτικές κάρτες και γενικά από οτιδήποτε σχετίζεται με τη λειτουργία μίας τράπεζας. Στόχος των εφαρμογών αυτών είναι η ταχύτερη εξυπηρέτηση των πελατών και η αποδοτικότερη λειτουργία της τράπεζας. Αναπτύσσονται ταχύτατα με νέα προϊόντα που βασίζονται στο διαδίκτυο και καταργούν την φυσική παρουσία σε κατάστημα. Προφανώς μια τέτοιου επιπέδου εφαρμογή πρέπει να υποστηρίζει ταυτόχρονες προσπελάσεις στα δεδομένα από πολλούς χρήστες.

Διαχείριση Εταιρικών Δεδομένων. Η καλή οργάνωση των δεδομένων μίας μεγάλης εταιρείας έχει ουσιαστική και δυναμική συμβολή στην αποδοτική λειτουργία της. Η μεθοδική οργάνωση των δεδομένων επιδρά σε όλα τα τμήματα της εταιρείας και αυτή η δομημένη οργάνωση της πληροφορίας επιτρέπει τη γρήγορη αναζήτηση στοιχείων και την άμεση ενημέρωση σε τυχόν αλλαγές στα δεδομένα.

Πολυμεσικές Εφαρμογές. Μιας και στις ημέρες μας ένα σημαντικό ποσοστό της πληροφορίας βρίσκεται σε μορφές όπως ήχος, εικόνα και video, η υποστήριξη των τύπων αυτών απαιτεί ταχύτητα στην επεξεργασία και περισσότερο χώρο αποθήκευσης λόγω του μεγάλου όγκου των δεδομένων. Οι Βάσεις Δεδομένων πολυμέσων (multimedia databases) έχουν τη δυνατότητα χειρισμού των σύνθετων αυτών τύπων δεδομένων χρησιμοποιώντας σύγχρονες και βελτιωμένες τεχνικές αναζήτησης. Έχουμε πλέον τη δυνατότητα να θέτουμε ερωτήσεις που αφορούν πολυμεσικά δομικά στοιχεία, κάτι που θα φαινόταν εξαιρετικά πολύπλοκο και δύσκολο στο παρελθόν.

Ανάκτηση πληροφορίας στο διαδίκτυο. Τα παραδοσιακά συστήματα επιτρέπουν την αναζήτηση κειμένων που περιέχουν λέξεις ορισμένες από τον χρήστη. Με την αλματώδη ανάπτυξη του Παγκόσμιου Ιστού και του Διαδικτύου, η πληροφορία είναι διάχυτη παντού κάνοντας την αναζήτηση το επόμενο στοιχείο. Ένα σημαντικό τμήμα της έρευνας εστιάζεται στη μελέτη αποτελεσματικών και αποδοτικών μεθόδων αναζήτησης πληροφορίας σύμφωνα με τις απαιτήσεις των χρηστών. Βασικό στοιχείο στη διαδικασία αυτή είναι η βαθμο-

λόγηση των απαντήσεων έτσι ώστε τα δεδομένα που σχετίζονται περισσότερο με το ερώτημα του χρήστη να πάρουν υψηλό βαθμό. Οι μηχανές αναζήτησης (search engines) όπως Google, Bing, Yahoo!, κ.λπ. χρησιμοποιούν εξελιγμένες τεχνικές και ικανοποιούν τα εκατομμύρια ερωτήματα που δέχονται καθημερινά με την βοήθεια ΣΔΒΔ με δυνατότητες αναζήτησης κειμένων σύμφωνα με τις λέξεις-κλειδιά που δίνει ο χρήστης.

Διαχείριση Γεωγραφικών Δεδομένων. Τα γεωγραφικά δεδομένα απαιτούν νέους τρόπους αποθήκευσης και επεξεργασίας λόγω της χωρικής (spatial) φύσης τους. Για παράδειγμα, για τα δεδομένα αυτά λόγω της διδιάστατης (ή γενικότερα πολυδιάστατης) φύσης τους δεν ισχύουν οι σχέσεις διάταξης (π.χ. στους ακεραίους και τους χαρακτήρες υπάρχει η έννοια του προηγούμενου επόμενου, στους πραγματικούς αριθμούς η έννοια του μικρότερου-μεγαλύτερου κ.λπ.). Ισχύουν όμως τοπολογικές σχέσεις (όπως αριστεράδεξιά, βόρειανότια, εντόςεκτός, τομή, απόσταση κ.λπ.). Το αντίστοιχο επομένως ΣΔΒΔ, που συνεργάζεται με ένα Σύστημα Γεωγραφικών Πληροφοριών (Geographical Information System, GIS), πρέπει να είναι εξοπλισμένο με εξειδικευμένες τεχνικές διαχείρισης.

Αποθήκες Δεδομένων και Αναλυτική Επεξεργασία. Μία άλλη σύγχρονη εφαρμογή που προέκυψε ως ανάγκη λόγω της ύπαρξης τεράστιου όγκου δεδομένων σε μεγάλες επιχειρήσεις και οργανισμούς είναι τα Συστήματα Στήριξης Αποφάσεων (decision support systems). Για τη λήψη αποφάσεων σε θέματα ανάλυσης αγοράς, οικονομικού σχεδιασμού, marketing κ.λπ., ένας αναλυτής μίας επιχείρησης ενδιαφέρεται να υποβάλλει πολύπλοκα ερωτήματα περισσότερο σε συγκεντρωτικά δεδομένα παρά σε ατομικές συναλλαγές. Η υποβολή και η επεξεργασία τέτοιου τύπου ερωτημάτων είναι μία διαδικασία από δύσκολη μέχρι και αδύνατη, καθώς συχνά τα δεδομένα των μεγάλων φορέων είναι διασκορπισμένα σε πολλά ετερογενή συστήματα, που μπορεί να είναι και γεωγραφικά κατανεμημένα. Στις περιπτώσεις αυτές, τα πρωτογενή δεδομένα με ειδικές διαδικασίες ομογενοποιούνται και αποθηκεύονται σε ειδικές βάσεις με συνοπτικό τρόπο χρησιμοποιώντας διαφορετικές οπτικές γωνίες και διαφορετικές κλίμακες. Αυτές οι ειδικές βάσεις λέγονται Αποθήκες Δεδομένων (data warehouses), ενώ η επεξεργασία τους έχει την ειδική ονομασία Άμεση Αναλυτική Επεξεργασία (OnLine Analytical Processing, OLAP) σε αντίθεση με τα παραδοσιακά Συστήματα Βάσεων Δεδομένων, όπου μιλούμε για Άμεση Επεξεργασία Συναλλαγών.

Εξόρυξη Δεδομένων. Η Εξόρυξη Δεδομένων (data mining) είναι μία άλλη σύγχρονη περιοχή των Βάσεων Δεδομένων, που πρόκυψε ως ανάγκη λόγω του τεράστιου όγκου δεδομένων, που συλλέγονται είτε αυτοματοποιημένα από δορυφόρους, κάμερες παρακολούθησης, συσκευές μέτρησης είτε με τον κλασσικό παραδοσιακό τρόπο από ένα Σύστημα Βάσεων Δεδομένων. Ο τεράστιος όγκος δεδομένων αποτρέπει την χρήση συμβατικών μεθόδων επεξεργασίας και έτσι εφαρμόζουμε ειδικές αυτοματοποιημένες μεθόδους επεξεργασίας, οι οποίες αναλαμβάνουν να "σκάψουν" στα δεδομένα αναζητώντας κάποια χρήσιμη πληροφορία, προηγουμένως άγνωστη και κυρίως ενδιαφέρουσα, με την οποία να μπορούν ληφθούν αποφάσεις και να δρομολογηθούν διαδικασίες. Η αποδοτική διαχείριση τεράστιων όγκων δεδομένων με τη βοήθεια κλασσικών τεχνικών των ΒΔ είναι το χαρακτηριστικό της. Τυπικές μέθοδοι επεξεργασίας κατά την εξόρυξη δεδομένων είναι ο χαρακτηρισμός (characterization), η αντιδιαστολή (discrimination), η κατηγοριοποίηση (classification), η εξαγωγή κανόνων συσχέτισης (association rules), η ομαδοποίηση (clustering) κ.α.

Διαχείριση Κινούμενων Αντικειμένων είναι μια ακόμη εφαρμογή με ραγδαία ανάπτυξη. Εφαρμογές όπως η καταγραφή της κίνησης οχημάτων, η παρακολούθηση μίας πετρελαιοκλιδίας, η παρακολούθηση εξέλιξης παγετώνων, απαιτούν την αποθήκευση της θέσης των αντικειμένων σε σχέση με το χρόνο. Η μελέτη της κίνησης επιτρέπει να απαντούμε σε καίρια ερωτήματα που σχετίζονται κύρια με: α) την εξέλιξη της κίνησης στο παρελθόν και β) την πρόβλεψη για τη θέση των αντικειμένων στο κοντινό ή μακρινό μέλλον. Η επεξεργασία ερωτημάτων τέτοιου τύπου είναι εξαιρετικά δύσκολη και απαιτεί την κατάλληλη οργάνωση και διαχείριση των αντικειμένων με μοντέρνες ευρηματικές τεχνικές έτσι ώστε να υλοποιείται αποτελεσματικά και αποδοτικά.



Ερωτήσεις / Δραστηριότητες

1. Αναζήτηση πληροφοριών για τον εκτιμώμενο όγκο δεδομένων που παράγονται και διακινούνται με χρήση ΤΠΕ.
2. Αναζήτηση πληροφοριών για τη ζήτηση και τις προοπτικές θέσεων εργασίας σχετικών με ΒΔ.
3. Αναζήτηση πληροφοριών σχετικά με νομοθεσία που επιβάλλει τη διατήρηση δεδομένων από οργανισμούς/επιχειρήσεις.
4. Επίσκεψη ιστοσελίδων που βασίζονται σε ΒΔ για παροχή υπηρεσιών σε πολίτες και οργανισμούς/επιχειρήσεις (π.χ. Εθνικό Τυπογραφείο, βάση νομοθεσίας ΕΕ EURLex, πύλη δημόσιας διοίκησης ΕΡΜΗΣ, πύλη ΕΕ Europa, ηλεκτρονικές βιβλιοθήκες πανεπιστημίων).

κεφάλαιο

2

Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ)

2.1 Λειτουργίες ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει την έννοια ενός ΣΔΒΔ
- να αντιλαμβάνεται ότι τα ΣΔΒΔ έχουν πολλαπλούς τρόπους εγκατάστασης
- να απαριθμεί τις βασικές λειτουργίες ενός ΣΔΒΔ

Το **ΣΔΒΔ**, όπως είδαμε σε προηγούμενο κεφάλαιο έχει σαν βασικό στόχο τον γενικό χειρισμό της βάσης, όσον αφορά τη δημιουργία, συντήρηση, επεξεργασία στοιχείων, ελέγχους ασφαλείας κτλ, και την εξυπηρέτηση των χρηστών σε όλα τα επίπεδα. Ουσιαστικά, θα μπορούσαμε να ισχυριστούμε, το ΣΔΒΔ είναι ένας **μεσάζων** μεταξύ του χρήστη και της βάσης δεδομένων και μόνο μέσω αυτού ο χρήστης μπορεί να ζητήσει πληροφορίες από τη βάση. Μπορεί το ΣΔΒΔ να είναι εγκατεστημένο σε ένα Η/Υ και να χρησιμοποιείται από ένα χρήστη (single user system) ή να είναι εγκατεστημένο σε ένα σύνολο ηλεκτρονικών υπολογιστών, που επικοινωνούν μεταξύ τους μέσω κάποιου (τοπικού ή απομακρυσμένου) δικτύου, και να χρησιμοποιείται από πολλούς χρήστες (multi user system). Οι βασικές λειτουργίες τις οποίες αναλαμβάνει ένα ΣΔΒΔ είναι όπως είδαμε παραπάνω:

- Οργανώνει τη βάση δεδομένων στο μέσο αποθήκευσης (σκληροί δίσκοι, οπτικοί δίσκοι, νέφος κλπ.).
- Διαθέτει τους μηχανισμούς για τη διαχείριση των δεδομένων.
- Τροφοδοτεί τις εφαρμογές με δεδομένα στη μορφή που αυτές τα ζητούν.



Ερωτήσεις / Δραστηριότητες

1. Απαριθμήστε τις βασικές λειτουργίες που αναλαμβάνει ένα ΣΔΒΔ.
2. Ποιες είναι οι δυνατές περιπτώσεις εγκατάστασης ενός ΣΔΒΔ;

2.2 Χαρακτηριστικά ενός ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει την έννοια της συναλλαγής
- να διακρίνει τις απαιτήσεις αξιοπιστίας ACID
- να διακρίνει τις απαιτήσεις αξιοπιστίας με βάση το θεώρημα GAP

Την ανάγκη για αξιολόγηση της αξιοπιστίας σε οτιδήποτε αφορά τις συναλλαγές στη βάση δεδομένων, έρχεται να εγγυηθεί η αποδοχή των απαιτήσεων **ACID** (Atomicity, Consistency, Isolation, Durability) - (Ατομικότητα, Συνέπεια, Απομόνωση, Μονιμότητα). Σαν **συναλλαγή** θα θεωρήσουμε οποιαδήποτε λογική ενέργεια που σχετίζεται με τα δεδομένα. Π.χ. μια μεταφορά κεφαλαίου από ένα λογαριασμό μιας τράπεζας σε έναν άλλο, όσες κινήσεις και να χρειάζεται τελικά είναι μια συναλλαγή.

Η **Ατομικότητα** απαιτεί η τροποποίηση που θα γίνει στην ΒΔ να τηρεί τον κανόνα **όλα ή τίποτα**, αν δηλαδή ένα μέρος της συναλλαγής αποτύχει, αποτυγχάνει όλη η συναλλαγή και η ΒΔ μένει όπως ήταν πριν εκτελεστεί η συναλλαγή. Η Ατομικότητα σημαίνει ότι οι χρήστες είναι απαλλαγμένοι από τον φόβο μη ολοκληρωμένων συναλλαγών.

Η **Συνέπεια** διασφαλίζει ότι η ΒΔ διατηρείται σε μια συνεπή κατάσταση, συγκεκριμένα λέει ότι κάθε συναλλαγή θα οδηγεί την βάση δεδομένων από τη μια συνεπή κατάσταση στην άλλη. Σε περίπτωση που μια συναλλαγή παραβιάζει κάποιο κανόνα της συνέπειας, ανακαλείται προκειμένου η ΒΔ να έχει μόνο έγκυρα δεδομένα. Π.χ. αν σε μια ΒΔ ένα πεδίο είναι μόνο για ακέραιους αριθμούς τότε το ΣΔΒΔ μπορεί είτε να απορρίψει απόπειρες για είσοδο δεκαδικών αριθμών είτε να τους στρογγυλοποιήσει. Και οι δυο αυτές ενέργειες διατηρούν την συνέπεια. Υπάρχουν τρία είδη συνέπειας, ισχυρή, ασθενής και ενδεχόμενη, η μελέτη τους όμως ξεφεύγει από το πλαίσιο του παρόντος βιβλίου.

Η **Απομόνωση** αναφέρεται στην απαίτηση ότι όλες οι ενέργειες δεν μπορούν να έχουν πρόσβαση ή να δουν δεδομένα τα οποία τροποποιούνται εκείνη την στιγμή από μια συναλλαγή η οποία δεν έχει ακόμα ολοκληρωθεί.

Η **Μονιμότητα** εγγυάται στον χρήστη ότι αν τελειώσει μια συναλλαγή επιτυχώς τότε τα αποτελέσματα της δεν θα χαθούν. Οι αλλαγές που έχει κάνει η συναλλαγή δεν θα χαθούν σε περίπτωση απώλειας ρεύματος ή άλλης καταστροφής.

Όταν έχουμε ένα πολύπλοκα διαμοιρασμένο σύστημα αποθήκευσης πληροφοριών είναι φυσικό να αναπτυχθεί και αντίστοιχος προβληματισμός σε θεωρητικό επίπεδο. Αποτέλεσμα αυτής της θεωρητικής μελέτης είναι το **θεώρημα CAP** (Consistency, Availability, Partition tolerance) (Συνέπεια, Διαθεσιμότητα, ανοχή Διαμερισμού), γνωστό επίσης και ως θεώρημα Brewer. Σύμφωνα με αυτό είναι **αδύνατο** για ένα καταμεμημένο σύστημα υπολογιστών να παρέχει ταυτόχρονα και τις τρεις εγγυήσεις:

Συνέπεια, αναπτύχθηκε βέβαια προηγουμένως αλλά κατά τον Brewer η κεντρική ιδέα είναι ότι όλοι οι κόμβοι βλέπουν τα ίδια δεδομένα την ίδια στιγμή.

Διαθεσιμότητα, εγγύηση ότι κάθε αίτημα παίρνει μια απάντηση σχετικά με το αν πέτυχε ή απέτυχε.

ανοχή Διαμερισμού, εγγύηση ότι το σύστημα συνεχίζει να λειτουργεί παρά τον διαμερισμό που οφείλεται σε βλάβες του δικτύου.



Ερωτήσεις / Δραστηριότητες

1. Ποιες είναι οι απαιτήσεις ACID;
2. Τι περιλαμβάνει το θεώρημα GAP;

2.3 Επισκόπηση της Εξέλιξης των ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να διακρίνει τα διαφορετικά μοντέλα ΣΔΒΔ
- να αναφέρει το βασικό χαρακτηριστικό κάθε κατηγορίας ΣΔΒΔ
- να έχει κατανοήσει σε πρώτο βαθμό τη λειτουργία του Σχεσιακού Συστήματος Διαχείρισης Βάσεων Δεδομένων

Η εμφάνιση των Βάσεων Δεδομένων από τα μέσα της δεκαετίας του 1960 κυρίως, συμπίπτει χρονικά με τη δυνατότητα της άμεσης αποθήκευσης σε σκληρούς δίσκους και δισκέτες. Αυτό επιτρέπει την διαδραστική χρήση σε αντίθεση με τα *tapedbased* συστήματα του παρελθόντος. Έχουμε ήδη αναφέρει ότι εκείνη την εποχή ο βασικός στόχος ήταν να γίνουν τα δεδομένα ανεξάρτητα από τη λογική της εφαρμογής, αποκλειστικά σε κάποια προγράμματα, έτσι ώστε να μπορούν να διατεθούν σε διαφορετικές εφαρμογές.

Η πρώτη γενιά των συστημάτων βάσεων δεδομένων ήταν βάσεις πλοήγησης (*navigational*), όπου οι εφαρμογές συνήθως είχαν πρόσβαση σε δεδομένα ακολουθώντας δείκτες από μία εγγραφή σε μία άλλη. Τα δύο **κύρια (προ σχεσιακά) μοντέλα** αυτή την εποχή ήταν:

- το **ιεραρχικό** μοντέλο (*Hierarchical Model*) το οποίο έχει μια ιεραρχική δομή που θυμίζει δένδρο
- το **δικτυωτό** μοντέλο *Network Model*

Για να καταλάβουμε την δομή του ιεραρχικού μοντέλου χρειάζεται να ξεκαθαρίσουμε την έννοια της οντότητας. Με τον όρο **οντότητα** (*entity*) αναφερόμαστε σε κάθε αντικείμενο, έννοια, πρόσωπο ή κατάσταση που έχει μια ανεξάρτητη ύπαρξη. Είναι κάτι ξεχωριστό και μπορούμε να συγκεντρώσουμε πληροφορίες γι' αυτό.

Οι οντότητες στο ιεραρχικό μοντέλο μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες. Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο ξεκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μία και μόνο μία οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίστηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

Στο **δικτυωτό** (*network*) μοντέλο, υπάρχει επίσης ένα επίπεδο ιεραρχίας αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σ' ένα κατώτερο ή σ' ένα ανώτερο επίπεδο.

Το **Σχεσιακό Μοντέλο** - *Relational Model*, προτάθηκε για πρώτη φορά το 1970 από τον Edgar F. Codd, ο οποίος σκέφτηκε ότι οι αιτήσεις του χρήστη θα πρέπει να αναζητούν δεδομένα με **βάση το περιεχόμενο** και όχι όπως πριν ακολουθώντας απλά συνδέσμους. Τα σχεσιακά συστήματα χρειάστηκαν για την υλοποίηση τους σοβαρές ανάγκες όσον αφορά τους πόρους και κυρίως τις δυνατότητες επεξεργασίας με αποτέλεσμα να καθυστερήσει η ανάπτυξη τους μέχρι τα μέσα της δεκαετίας του 1980. Τότε το υλικό των υπολογιστών αναπτύχθηκε ανάλογα και έγινε ικανό να τα υποστηρίξει. Από τις αρχές του 1990, τα σχεσιακά συστήματα ήταν κυρίαρχα σε όλες τις εφαρμογές μεγάλης κλίμακας επεξεργασίας στοιχείων και παραμένουν κυρίαρχα μέχρι και σήμερα, εκτός από εξειδικευμένες περιοχές. Η κυρίαρχη γλώσσα των Βάσεων Δεδομένων είναι το πρότυπο SQL (*Structured Query Language*).

Επειδή το Σχεσιακό μοντέλο είναι μεν μία καλή βάση για μία γλώσσα ερωταπαντήσεων αλλά υστερεί σαν γλώσσα μοντελοποίησης, αναπτύχθηκε το μοντέλο **Οντοτήτων - Συσχετίσεων** (*EntityRelationship model*) που κέρδισε τη μάχη για το σχεδιασμό βάσεων δεδομένων.

Από το 1970 και μετά, η τεχνολογία βάσεων δεδομένων συμβαδίζει με τις προοπτικές που προσφέρουν η αύξηση της χωρητικότητας και της ταχύτητας των υπολογιστών. Στο Σχεσιακό μοντέλο όλα τα δεδομένα αποθηκεύονται σε πίνακες με συγκεκριμένη δομή γραμμών και στηλών, που όσο αυξάνονται οι απαιτήσεις όλο και περισσότερο θεωρείται ως περιορισμός κατά τον χειρισμό πληροφοριών.

Η δυσκολία μοντελοποίησης που θίξαμε στην προηγούμενη παράγραφο, μεταφράζεται σε αδυναμία αναπαράστασης του φυσικού κόσμου, αφού σήμερα απαιτείται η δυνατότητα ορισμού νέων τύπων δεδομένων και λειτουργιών για την υλοποίηση π.χ. μίας εφαρμογής GIS η οποία χρειάζεται υποστήριξη για γεωμετρικά αντικείμενα (γραμμές, πολύγωνα) και πράξεις μεταξύ των γεωμετρικών αντικειμένων (τομές γραμμών, επικάλυψη πολυγώνων κλπ.). Τα παραδοσιακά σχεσιακά ΣΔΒΔ δεν επιτρέπουν τον ορισμό νέων τύπων δεδομένων και λειτουργιών, οπότε η διαχείριση των αντικειμένων εμπεριέχεται στη λογική της εφαρμογής και όχι στο ΣΔΒΔ.

Διάφορες προσπάθειες έχουν γίνει για να αντιμετωπιστεί αυτό το πρόβλημα, πολλές από αυτές οδηγούν σε λύσεις όπως οι «**μετάσχεσιακές**» (**αντικειμενοστραφή, NoSQL, NewSQL**).

Σε ένα **αντικειμενοστραφές** ΣΔΒΔ η πληροφορία προσεγγίζεται με την μορφή των αντικειμένων που χρησιμοποιούνται στον αντικειμενοστραφή προγραμματισμό. Οι ΒΔ δηλαδή κρατούν τα δεδομένα τους σαν αντικείμενα, σύμφωνα με τις αρχές του αντικειμενοστραφούς προγραμματισμού.

Οι **NoSQL** βάσεις δεδομένων χρησιμοποιούνται όλο και περισσότερο για την διαχείριση τεράστιου όγκου δεδομένων και σε web εφαρμογές πραγματικού χρόνου. Τα συστήματα NoSQL ονομάζονται επίσης "Not only SQL" για να τονιστεί η δυνατότητα υποστήριξης SQL. Μια NoSQL βάση δεδομένων παρέχει ένα μηχανισμό για την αποθήκευση και την ανάκτηση των δεδομένων που διαμορφώνεται σε άλλα μέσα εκτός των πινάκων, που στηρίζουν τις παραδοσιακές σχεσιακές βάσεις δεδομένων.

Αξιοσημείωτο ότι δεν ακολουθεί το πακέτο ACID ως εγγύηση για την αξιοπιστία της ΒΔ αλλά το θεώρημα CAP.

Η απάντηση των σχεσιακών συστημάτων διαχείρισης ΒΔ στο παραπάνω μοντέλο, είναι η κατηγορία **NewSQL**. Αποτελούν σύγχρονα σχεσιακά ΣΔΒΔ που επιδιώκουν να παρέχουν την ίδια απόδοση των NoSQL συστημάτων για τις ηλεκτρονικές συναλλαγές, διατηρώντας τις ACID εγγυήσεις αξιοπιστίας ενός παραδοσιακού συστήματος βάσης δεδομένων.



Ερωτήσεις / Δραστηριότητες

1. Τι είναι οντότητα;
2. Τι είναι ιδιότητες ή χαρακτηριστικά των οντοτήτων;
3. Τι είναι οι συσχετίσεις μεταξύ των οντοτήτων;
4. Ποια είναι τα κύρια μοντέλα των Βάσεων Δεδομένων;
5. Ποια είναι τα προσχεσιακά μοντέλα ΣΔΒΔ;
6. Τι ξέρετε για τα σχεσιακά μοντέλα ΣΔΒΔ;
7. Σε τι επικεντρώνονται τα μετασχεσιακά ΣΔΒΔ;
8. Ποιο πακέτο για τις εγγυήσεις αξιοπιστίας ακολουθεί η κατηγορία **NewSQL**;

2.4 Αρχιτεκτονική και Συστατικά ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει την αρχιτεκτονική των τριών επιπέδων (ANSI/SPARC)
- να απαριθμεί τις διάφορες κατηγορίες χρηστών ενός ΣΔΒΔ
- να διακρίνει τα σημαντικότερα υποσυστήματα ενός ΣΔΒΔ
- να αντιλαμβάνεται τη σημασία ύπαρξης του υλικού (hardware) ενός ΣΔΒΔ

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

Αυτή η κοινή διεπαφή (interface) των εφαρμογών με τα αρχεία ονομάζεται λογική διεπαφή. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν κ.λπ. Αυτό αποτελεί και τον ορισμό της ανεξαρτησίας δεδομένων. Πρακτικά σημαίνει ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων.

Στα ΣΔΒΔ έχει επικρατήσει η αρχιτεκτονική των εξής **τριών επιπέδων (ANSI/SPARC)**:

Εσωτερικό επίπεδο (internal level), έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάσταση τους.

Εξωτερικό επίπεδο (external level), έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και οι διαχειριστές της βάσης δεδομένων.

Εννοιολογικό επίπεδο (conceptual level), είναι το ενδιάμεσο επίπεδο που συνδέει τα δύο άλλα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων.

Οι δυνατότητες του ΣΔΒΔ αξιοποιούνται από διάφορες κατηγορίες χρηστών. Οι κατηγορίες αυτές έχουν ως εξής:

Τελικοί Χρήστες. Στην ουσία οι πελάτες της ΒΔ. **Διακρίνονται σε:**

Απλούς χρήστες. Στην κατηγορία αυτή ανήκουν οι χρήστες που χρησιμοποιούν κάποιο πρόγραμμα εφαρμογής για να αποκτήσουν πρόσβαση στα δεδομένα της βάσης. Τα δικαιώματα των απλών χρηστών είναι περιορισμένα, ενώ οι λειτουργίες που επιτρέπεται να πραγματοποιήσουν είναι προκαθορισμένες και ρυθμίζονται αναλόγως με τη λειτουργία του συστήματος. Για παράδειγμα, ένας χρήστης του φορολογικού συστήματος TAXIS, ο οποίος έχει υποβάλλει ηλεκτρονικά τη φορολογική του δήλωση, μπορεί να διαβάσει και να τροποποιήσει μόνο τα δικά του δεδομένα και όχι άλλων χρηστών. Οι απλοί χρήστες τις περισσότερες φορές αγνοούν την ύπαρξη του ΣΔΒΔ.

Προχωρημένοι Χρήστες. Στην κατηγορία αυτή ανήκουν οι χρήστες που έχουν επίγνωση της ύπαρξης του ΣΔΒΔ καθώς και της δομής της βάσης. Έχουν τη δυνατότητα να χρησιμοποιούν τη Δομημένη Γλώσσα ερωτημάτων (SQL) για να επικοινωνούν με τη βάση, ενώ σε πολλές περιπτώσεις γράφουν τα δικά τους προγράμματα εφαρμογής.

Προγραμματιστές Εφαρμογών. Ο προγραμματιστής εφαρμογών καλείται να υλοποιήσει τμήματα λογισμικού, τα οποία επικοινωνούν με το ΣΔΒΔ για προσπέλαση και ενημέρωση δεδομένων. Οι εφαρμογές

υλοποιούνται χρησιμοποιώντας μία γλώσσα προγραμματισμού (π.χ. C++, Java) ή χρησιμοποιώντας κάποιο περιβάλλον ανάπτυξης εφαρμογών, το οποίο παρέχει ο κατασκευαστής του ΣΔΒΔ.

Σχεδιαστές Βάσεων Δεδομένων. Η αναπαράσταση του πραγματικού κόσμου απαιτεί την προσεκτική σχεδίαση της Βάσης Δεδομένων. Αρχικά θα πρέπει να προσδιορισθούν οι απαραίτητες οντότητες και οι συσχετίσεις που υπάρχουν μεταξύ τους. Στη συνέχεια, η πληροφορία αυτή χρησιμοποιείται για την κατασκευή της εσωτερικής αναπαράστασης της πληροφορίας στη Βάση Δεδομένων του συστήματος. Η διαδικασία αυτή εκτελείται από εξειδικευμένους αναλυτές και σχεδιαστές Βάσεων Δεδομένων, οι οποίοι καλούνται να περιγράψουν τα αντικείμενα του πραγματικού κόσμου με αντικείμενα της Βάσης Δεδομένων. Οι σχεδιαστές έχουν στη διάθεσή τους εργαλεία λογισμικού με τα οποία αυτοματοποιείται η διαδικασία περιγραφής των δεδομένων.

Διαχειριστές Βάσεων Δεδομένων. Στην κατηγορία αυτή ανήκουν οι χρήστες που έχουν επιπλέον δικαιώματα που αφορούν στον τρόπο οργάνωσης των δεδομένων, στον τρόπο λειτουργίας του ΣΔΒΔ, στη διαχείριση των χρηστών και στην αποκατάσταση των δεδομένων σε περιπτώσεις απώλειας δεδομένων λόγω βλαβών του υλικού του συστήματος. Οι διαχειριστές έχουν στη διάθεσή τους εξειδικευμένες εφαρμογές με τις οποίες ελέγχουν την ορθότητα των δεδομένων και φροντίζουν για τη βέλτιστη λειτουργία του συστήματος.

Ένα ακόμη συστατικό προς μελέτη αποτελεί η **γλώσσα Βάσεων Δεδομένων**. Αποτελείται από δυο τμήματα που είναι σχεδιασμένα για διαφορετικούς σκοπούς. Η **γλώσσα ορισμού δεδομένων** (Data Definition Language, DDL) χρησιμοποιείται για τον ορισμό των οντοτήτων και των μεταξύ τους σχέσεων. Με τη γλώσσα αυτή δηλώνουμε τα χαρακτηριστικά που έχει κάθε οντότητα και τους αντίστοιχους τύπους δεδομένων (π.χ., ακέραιος αριθμός, σειρά χαρακτήρων). Η **γλώσσα χειρισμού δεδομένων** (Data Manipulation Language, DML) χρησιμοποιείται για την επεξεργασία, την ενημέρωση, την εισαγωγή και τη διαγραφή δεδομένων. Αξίζει να επισημανθεί ότι μία γλώσσα Βάσεων Δεδομένων έχει τεράστια διαφορά από μία γλώσσα προγραμματισμού. Ενώ η γλώσσα προγραμματισμού είναι ενισχυμένη με πολλές δυνατότητες για τη διευκόλυνση του προγραμματιστή εφαρμογών, μία γλώσσα Βάσεων Δεδομένων είναι απλούστερη στη σύνταξή της και κατά κύριο λόγο είναι δηλωτική (declarative). Αυτό σημαίνει ότι δεν προσδιορίζουμε τον τρόπο εκτέλεσης μίας εργασίας, αλλά απλώς δηλώνουμε τι θέλουμε στην απάντηση. Η μετάφραση, αξιολόγηση και επεξεργασία των ερωτημάτων είναι άλλη μια ευθύνη του ΣΔΒΔ.

Να σημειωθεί εδώ ότι το ΣΔΒΔ δεν είναι απόλυτο και άκαμπτο λογισμικό. Αποτελείται από πολλά υποσυστήματα που συνεργάζονται αρμονικά μεταξύ τους προκειμένου να πετύχουν αποτελεσματική και αποδοτική διαχείριση των δεδομένων της βάσης. Μερικές από τις βασικές απαιτήσεις του ΣΔΒΔ καλύπτονται από τις λειτουργίες του λειτουργικού συστήματος. Οι πιο σύνθετες και πολύπλοκες όμως διεργασίες υποστηρίζονται από το ΣΔΒΔ χρησιμοποιώντας ως βάση τις λειτουργίες του λειτουργικού συστήματος. Επομένως, η σωστή σχεδίαση ενός ΣΔΒΔ προϋποθέτει τη μελέτη της επικοινωνίας μεταξύ ΣΔΒΔ και λειτουργικού συστήματος. Τα σημαντικότερα υποσυστήματα είναι τα παρακάτω:

Διαχειριστής Αρχείων (file manager). Σε ένα αρχείο αποθηκεύονται δεδομένα, βοηθητικές δομές δεδομένων, το λεξικό δεδομένων και άλλες πληροφορίες απαραίτητες για το ΣΔΒΔ. Ο τρόπος οργάνωσης των αρχείων αυτών ορίζεται από το διαχειριστή αρχείων.

Τα **ευρετήρια** (index files) είναι επιπλέον βοηθητικές δομές προσπέλασης ενός αρχείου εγγραφών. Αποτελούν δευτερεύοντες δρόμους αναζήτησης σε αρχεία εγγραφών, που προσφέρουν στους χρήστες εναλλακτικούς τρόπους προσπέλασης χωρίς να επηρεάζουν την φυσική τοποθέτηση των εγγραφών στο δίσκο.

Διαχειριστής Αποθήκης (storage manager). Το υποσύστημα αυτό είναι υπεύθυνο για τον τρόπο φυσικής αποθήκευσης των δεδομένων στη δευτερεύουσα μνήμη και για τον έλεγχο της μετακίνησης δεδομένων από και προς την κύρια μνήμη του συστήματος. Καθορίζει τη θέση αποθήκευσης των δεδομένων.

Διαχειριστής Συναλλαγών (transaction manager). Μία συναλλαγή αποτελείται από μία ή περισσότερες λειτουργίες που πρέπει να ολοκληρωθούν. Αν μία από τις λειτουργίες αποτύχει, τότε πρέπει να ακυρωθεί ολόκληρη η συναλλαγή. Χαρακτηριστικό παράδειγμα είναι η ανάληψη ενός ποσού από έναν τραπεζικό λογαριασμό. Η συναλλαγή αυτή αποτελείται από δύο λειτουργίες: α) ανάληψη χρημάτων και β) ενημέρωση του υπολοίπου. Αν αποτύχει οποιαδήποτε από τις δύο λειτουργίες πρέπει να ακυρωθεί και η άλλη.

Μεταγλωττιστής Ερωτημάτων Data Manipulation Language (DML query compiler). Πρόκειται για το υποσύστημα που είναι υπεύθυνο για τη μεταγλώττιση του ερωτήματος του χρήστη. Το ερώτημα δίνεται σε γλώσσα χειρισμού δεδομένων και μετατρέπεται σε εσωτερική αναπαράσταση κατάλληλη για επεξεργασία. Οι μεταγλωττισμένες εντολές προωθούνται στη μηχανή εκτέλεσης.

Αξίζει να αναφέρουμε επιπλέον και τα υποσυστήματα που ασχολούνται με:

Μεταγλώττιση Εντολών Data Definition Language (DDL compiler).

Έλεγχος Ταυτοχρονισμού (concurrency control manager).

Διαχείριση Επανάκτησης και Ημερολογίου (logging and recovery manager). Διαχείριση προσωρινής μνήμης (buffer manager).

Διαχείριση Αποθήκευσης δεδομένων (storage manager).

Κλείνουμε την ενότητα με το **υλικό** (hardware) ενός ΣΔΒΔ, τα μηχανήματα δηλαδή που είναι εγκατεστημένο το λογισμικό. Αυτό αποτελείται από το σύνολο των φυσικών ηλεκτρονικών συσκευών όπως υπολογιστές συνοδευόμενοι με τις απαραίτητες συσκευές εισόδου/εξόδου όπως π.χ. μονάδες δίσκου, συσκευές αποθήκευσης, πιθανά κανάλια εισόδου/εξόδου, και γενικά οποιεσδήποτε συσκευές ή συστήματα που μπορούν να υλοποιήσουν διασύνδεση μεταξύ υπολογιστών και πραγματικού κόσμου κ.ο.κ.



Ερωτήσεις / Δραστηριότητες

1. Ποια είναι τα επίπεδα στην αρχιτεκτονική των τριών επιπέδων;
2. Σχολιάστε τις διαφορετικές κατηγορίες χρηστών σε ένα ΣΔΒΔ.
3. Ποια είναι τα τμήματα που αποτελούν την γλώσσα Βάσεων Δεδομένων.
4. Αναφέρατε τα σημαντικότερα υποσυστήματα που συνεργάζονται μεταξύ τους προκειμένου να πετύχουν αποτελεσματική και αποδοτική διαχείριση των δεδομένων της βάσης

2.5 Κατηγορίες ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να προσδιορίζει κατηγορίες ΣΔΒΔ
- να αναφέρει βασικά χαρακτηριστικά κάθε κατηγορίας ΣΔΒΔ
- να αντιλαμβάνεται την παράλληλη εξέλιξη κάθε κατηγορίας με την αντίστοιχη της θεωρίας των Υπολογιστών

Τα **κεντρικά** ΣΔΒΔ υλοποιούνται όταν τα στοιχεία τους βρίσκονται, αποθηκεύονται και διατηρούνται σε μια ενιαία θέση. Αυτή η θέση είναι πολύ συχνά ένας κεντρικός υπολογιστής. Στις περισσότερες περιπτώσεις, μια κεντρική βάση δεδομένων θα χρησιμοποιηθεί από έναν οργανισμό ή ένα ίδρυμα (π.χ. κάποια εταιρεία επιχειρήσεων ή ένα πανεπιστήμιο.) Οι χρήστες έχουν πρόσβαση σε μια κεντρική βάση δεδομένων μέσω ενός δικτύου υπολογιστών.

Όταν άρχισε η εξέλιξη και διάδοση των προσωπικών υπολογιστών με παράλληλη μείωση του κόστους τους και αναπτύχθηκαν τα δίκτυα επικοινωνίας δεδομένων, τα ΣΔΒΔ άρχισαν να αναπτύσσονται πάνω σε ένα νέο μοντέλο, το μοντέλο **πελάτη - εξυπηρετητή** (client - server). Σύμφωνα με το μοντέλο αυτό, τόσο ο κεντρικός υπολογιστής όσο και οι υπολογιστές που χρησιμοποιούν οι χρήστες είναι προσωπικοί υπολογιστές και συνδέονται μεταξύ τους μέσω τοπικού δικτύου. Συνήθως, ο κεντρικός υπολογιστής είναι ισχυρότερος και διαθέτει μεγάλες αποθηκευτικές μονάδες. Στον κεντρικό υπολογιστή είναι αποθηκευμένα τα δεδομένα της βάσης και είναι εγκατεστημένο το ΣΔΒΔ και ονομάζεται **εξυπηρετητής** (server). Στους υπόλοιπους υπολογιστές (σταθμούς εργασίας), "τρέχουν" οι εφαρμογές επεξεργασίας και εκμετάλλευσης των δεδομένων της βάσης. Οι εφαρμογές αυτές, που μπορεί να είναι εφαρμογές ενσωματωμένες στο ΣΔΒΔ ή ανεξάρτητες εφαρμογές που έχει αναπτύξει τρίτος κατασκευαστής, ονομάζονται **πελάτες** (clients).

Μια **Κατανεμημένη** Βάση Δεδομένων (ΚΒΔ) είναι η βάση δεδομένων στην οποία τμήματά της αποθηκεύονται σε πολλούς υπολογιστές σε ένα δίκτυο. Οι χρήστες έχουν πρόσβαση στο τμήμα της βάσης δεδομένων στην τοποθεσία τους, έτσι ώστε να έχουν πρόσβαση στα δεδομένα που σχετίζονται με τα καθήκοντά τους, χωρίς να επηρεάζεται η εργασία των άλλων χρηστών. Ένα κεντρικό κατανεμημένο σύστημα διαχείρισης βάσεων δεδομένων ΚΣΔΒΔ διαχειρίζεται την βάση δεδομένων σαν να ήταν όλα αποθηκευμένα στον ίδιο υπολογιστή. Το ΚΣΔΒΔ συγχρονίζει όλα τα δεδομένα περιοδικά και σε περίπτωση που πολλοί χρήστες πρέπει να έχουν πρόσβαση στα ίδια δεδομένα, διασφαλίζει ότι οι διαγραφές και ενημερώσεις σε μια τοποθεσία, αυτόματα αντικατοπτρίζονται στα δεδομένα που είναι αποθηκευμένα οπουδήποτε αλλού.

Με την εμφάνιση και χρήση της αρχιτεκτονικής συστημάτων πολυεπεξεργαστών, έχουμε πολλούς επεξεργαστές να μοιράζονται είτε κοινή μνήμη είτε κοινό δίσκο. Η επικοινωνία έτσι γίνεται χωρίς την επιπλέον επίπονη εργασία ανταλλαγής μηνυμάτων στο δίκτυο. Τα συστήματα διαχείρισης βάσεων δεδομένων που αναπτύχθηκαν με χρήση των παραπάνω αρχιτεκτονικών ονομάζονται **παράλληλα** συστήματα διαχείρισης βάσεων δεδομένων αφού χρησιμοποιούν την τεχνολογία παράλληλων επεξεργαστών.



Ερωτήσεις / Δραστηριότητες

1. Ποια είναι η κεντρική ιδέα σε ένα κεντρικό ΣΔΒΔ;
2. Σε τι μοντέλο αναπτύσσεται το μοντέλο **πελάτη - εξυπηρετητή** (clientserver);
3. Τι καινοτομία εισάγεται με την **Κατανεμημένη** Βάση Δεδομένων;

2.6 Διαδεδομένα ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να προσδιορίζει τα διαδεδομένα ΣΔΒΔ
- να διακρίνει τις διαφορετικές μεθόδους βαθμολόγησης ενός ΣΔΒΔ ανάλογα με την προτεινόμενη βαθμονόμηση

Στην εποχή μας όλα αλλάζουν με εξαιρετικά μεγάλη ταχύτητα. Θα ήταν λοιπόν μάλλον άκαιρο να μιλήσουμε για κατάταξη βάσει βαθμολόγησης των ΣΔΒΔ. Ακολουθώντας όμως την μηνιαία προσπάθεια που καταγράφεται στο <http://db-engines.com/en/ranking> παρατηρούμε για τον μήνα Ιούνιο 2015 την δημοτικότητα των ΣΔΒΔ βάσει της μεθόδου που έχει επιλεγεί και κατατίθεται εδώ, http://db-engines.com/en/ranking_definition

Κατάταξη			ΣΔΒΔ
Ιούνιος 2015	Μάιος 2015	Ιούνιος 2014	
1.	1.	1.	Oracle
2.	2.	2.	MySQL
3.	3.	3.	Microsoft SQL Server
4.	5.	4.	PostgreSQL
5.	4.	5.	MongoDB
6.	6.	6.	DB2
7.	7.	7.	Microsoft Access
8.	8.	9.	Cassandra
9.	9.	8.	SQLite
10.	10.	12.	Redis
11.	11.	10.	SAP Adaptive Server
12.	12.	11.	Solr
13.	13.	13.	Teradata
14.	14.	17.	Elasticsearch
15.	15.	15.	HBase
16.	16.	14.	FileMaker
17.	17.	18.	Hive
18.	18.	20.	Splunk
19.	19.	16.	Informix
20.	21.	23.	SAP HANA
21.	20.	19.	Memcached
22.	22.	22.	Neo4j
23.	23.	21.	CouchDB
24.	24.	26.	Couchbase
25.	25.	28.	MariaDB
26.	26.	25.	Firebird
27.	27.	24.	Netezza
28.	28.	27.	Microsoft Azure SQL Database
29.	30.	32.	Amazon DynamoDB



Ερωτήσεις / Δραστηριότητες

1. Αξιολογήστε την μέθοδο ταξινόμησης http://db-engines.com/en/ranking_definition
2. Δείτε την τρέχουσα ταξινόμηση των διαδεδομένων ΣΔΒΔ στο <http://db-engines.com/en/ranking>

κεφάλαιο

3

Σχεδιασμός Βάσεων Δεδομένων

3.1 Εισαγωγή

Τα συστήματα βάσεων δεδομένων έχουν σχεδιαστεί για να διαχειρίζονται μεγάλους όγκους πληροφοριών. Οι μεγάλοι αυτοί όγκοι πληροφοριών αποτελούν μέρος της λειτουργίας μίας επιχείρησης ή ενός οργανισμού. Το τελικό προϊόν της επιχείρησης ή του οργανισμού μπορεί να είναι η παροχή πληροφοριών από τη βάση δεδομένων ή κάποια συσκευή ή υπηρεσία. Στην τελευταία περίπτωση η βάση δεδομένων έχει μόνο υποστηρικτικό ρόλο. Το κεφάλαιο αυτό επικεντρώνεται στον σχεδιασμό βάσεων δεδομένων και ειδικότερα στον εννοιολογικό και λογικό σχεδιασμό της βάσης δεδομένων.

Πρόσθετο υποστηρικτικό υλικό υπάρχει στο παλιό βιβλίο για τις ΒΔ, των Ζαχαρή Κωνσταντίνου, Κασιμάτη Νικολάου, Κουνιάκη Χριστόφορου, Μανωλόπουλος Ιωάννη, Οικονόμου Θεόδωρου (http://www.pischools.gr/content/index.php?lesson_id=1&ep=224&c_id=752).

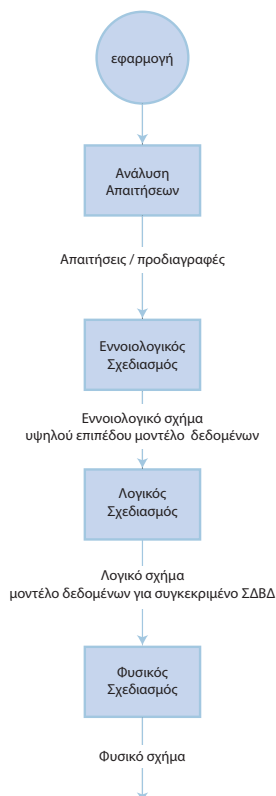


Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει τα βήματα σχεδιασμού βάσεων δεδομένων
- να διακρίνει τη σπουδαιότητα της ανάλυσης απαιτήσεων
- να διακρίνει τις διαφορές μεταξύ εννοιολογικού, λογικού και φυσικού σχεδιασμού

Για να είναι δυνατή η υλοποίηση μιας εφαρμογής με μεγάλη πολυπλοκότητα απαιτείται ο σωστός σχεδιασμός της. Για παράδειγμα για τον σωστό σχεδιασμό ενός αυτοκινήτου απαιτείται να προηγηθούν αρκετά σχέδια τα οποία απεικονίζουν διαφορετικές απόψεις του αντικειμένου. Οι βάσεις δεδομένων αποτελούν και αυτές πολύπλοκα συστήματα τα οποία μπορούν να περιγραφούν με διαφορετικούς σχεδιασμούς. Για τη Σχεδίαση μιας Βάσης δεδομένων απαιτούνται τα παρακάτω βήματα:



Εικόνα 3.1 Στάδια σχεδιασμού βάσης δεδομένων

3.1.1 Ανάλυση Απαιτήσεων

Απαίτηση είναι η περιγραφή μιας υπηρεσίας που θα πρέπει να παρέχει ένα σύστημα, μιας διεργασίας που θα πραγματοποιεί ή μιας συνθήκης που θα πρέπει να ικανοποιεί. Η απαίτηση απαντάει στο ερώτημα «τι κάνει το σύστημα;» και δεν περιλαμβάνει τον τρόπο που θα γίνει αυτό. Αποτελεί μια προσπάθεια αποσαφήνισης του προβλήματος που στοχεύει να λύσει η εφαρμογή που θα δημιουργηθεί και όχι στην περιγραφή κάποιας λύσης.

Η απαίτηση καταγράφεται και τεκμηριώνεται με τέτοιο τρόπο ώστε να είναι κατανοητή από όλα τα ενδιαφερόμενα μέρη στην ανάπτυξη της εφαρμογής. Επιπλέον η απαίτηση είναι μια σημαντική σταθερά που παραμένει ως ένας παράγοντας αξιολόγησης σε όλη τη διάρκεια του κύκλου ζωής ενός προϊόντος. Είναι δηλαδή ένας είδος συμβολαίου που ελέγχεται ακόμα και μετά το τέλος της ανάπτυξης του προϊόντος.

Η ανάλυση απαιτήσεων είναι μια διαδικασία αποτύπωσης όλων των απαιτήσεων, που πρέπει να πληροί η εφαρμογή. Η διαδικασία προσδιορισμού απαιτήσεων αποτελεί ένα δύσκολο έργο καθώς σε πολλές περιπτώσεις η επιτυχία μίας εφαρμογής στηρίζεται στην σωστή ανάλυση των απαιτήσεων. Για το σκοπό αυτό ο αναλυτής του συστήματος προετοιμάζει ένα έγγραφο απαιτήσεων που περιλαμβάνει πληροφορίες όπως:

- Ποια δεδομένα αποθηκεύονται και που
- Ποιες εφαρμογές θα χτιστούν πάνω από τα δεδομένα;
- Ποιες διαδικασίες πρέπει να υποστηριχθούν
- Ποιες είναι οι απαιτήσεις επίδοσης

Εφόσον έχει ολοκληρωθεί η διαδικασία προσδιορισμού απαιτήσεων το επόμενο βήμα αποτελεί ο σχεδιασμός της βάσης δεδομένων. Σε αυτό το βήμα είναι βασικό στοιχείο η περιγραφή της βάσης μέσα από τρεις διαφορετικούς σχεδιασμούς.

3.1.2. Εννοιολογικός, Λογικός και Φυσικός Σχεδιασμός

Εννοιολογικός Σχεδιασμός

Στον Εννοιολογικό Σχεδιασμό περιγράφονται οι οντότητες του πραγματικού κόσμου καθώς και οι συσχετίσεις μεταξύ τους. Για παράδειγμα σε μια εφαρμογή μαθητολογίου σε ένα σχολείο οντότητες μπορεί να είναι οι μαθητές, οι καθηγητές, τα μαθήματα, οι βαθμοί κ.α. Η οντότητα μαθητής συσχετίζεται με την οντότητα μάθημα και την οντότητα βαθμοί. Η οντότητα μαθητής έχει ιδιότητες όπως αριθμός μητρώου, επώνυμο, όνομα κ.α. Επίσης, περιγράφονται οι περιορισμοί που υπάρχουν στα δεδομένα (π.χ. δεν επιτρέπονται βαθμοί μικρότεροι του 0) και οι συσχετίσεις μεταξύ τους. Για κάθε οντότητα καθορίζεται το πρωτεύον κλειδί που η τιμή του είναι μοναδική για παράδειγμα στην οντότητα μαθητής πρωτεύον κλειδί μπορεί να είναι ο αριθμός μητρώου. Θα αναφερθούμε πιο αναλυτικά σε αυτά στην ενότητα 3.4.

Για τον σωστό εννοιολογικό σχεδιασμό έχουν αναπτυχθεί διαγράμματα όπως τα Διαγράμματα Οντοτήτων Συσχετίσεων (ΔΟΣ). Για παράδειγμα, σε αυτή τη φάση προσδιορίζονται οι οντότητες και οι συσχετίσεις, στη συνέχεια καθορίζονται οι ιδιότητες των οντοτήτων και συσχετίσεων, οι τύποι των συσχετίσεων και προσδιορίζονται τα πρωτεύοντα κλειδιά κάθε οντότητας. Η φάση αυτή ολοκληρώνεται με την κατασκευή και τον έλεγχο του ΔΟΣ.

Ο εννοιολογικός σχεδιασμός είναι ανεξάρτητος από τη φυσική οργάνωση των δεδομένων, με αποτέλεσμα να μην εξαρτάται από κάποιο συγκεκριμένο τρόπο αποθήκευσης.

Λογικός Σχεδιασμός

Ο Λογικός Σχεδιασμός αφορά τον σχεδιασμό του Σχεσιακού Μοντέλου το οποίο προκύπτει από την μεταφορά του ΔΟΣ που έχει κατασκευαστεί από τον εννοιολογικό σχεδιασμό. Για την σωστή σχεδίαση του Σχεσιακού Μοντέλου εφαρμόζονται μέθοδοι κανονικοποίησης και γίνεται έλεγχος πλεονασμού και ακεραιότητας δεδομένων. Ο σχεδιασμός ολοκληρώνεται με την ανασκόπηση όλης της διαδικασίας και την πρόβλεψη μελλοντικών αναγκών.

Φυσικός Σχεδιασμός

Ο Φυσικός σχεδιασμός περιγράφει τη φυσική οργάνωση της βάσης. Καθορίζεται ο τρόπος με τον οποίο αποθηκεύονται τα δεδομένα στο φυσικό μέσο αποθήκευσης (π.χ. σκληρό δίσκο) και ο τρόπος με τον οποίο πραγματοποιείται η προσπέλαση στα δεδομένα. Σε αυτή τη φάση γίνεται έλεγχος ότι η βάση ικανοποιεί τις απαιτήσεις επίδοσης/φόρτου κ.α, τα οποία επιβάλλουν οι απαιτήσεις του εγγράφου απαιτήσεων.



Ερωτήσεις / Δραστηριότητες

1. Για ποιους λόγους κατά τη γνώμη σας χρειάζονται οι τρεις διαφορετικοί σχεδιασμοί (εννοιολογικός, λογικός και φυσικός);
2. Ποια η διαφορά του εννοιολογικού σχεδιασμού από τον φυσικό σχεδιασμό;
3. Για ποιους λόγους είναι απαραίτητη η ανάλυση απαιτήσεων για τον σχεδιασμό μιας βάσης δεδομένων;

3.2 Μοντέλα Δεδομένων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει τον τρόπο που εξασφαλίζεται η ανεξαρτησία δεδομένων από την οργάνωση ενός ΣΔΒΔ σε 3 επίπεδα.
- να αναγνωρίζει την σημασία του μοντέλου δεδομένων
- να περιγράφει τη διαφορά του εννοιολογικού από το σχεσιακό μοντέλο

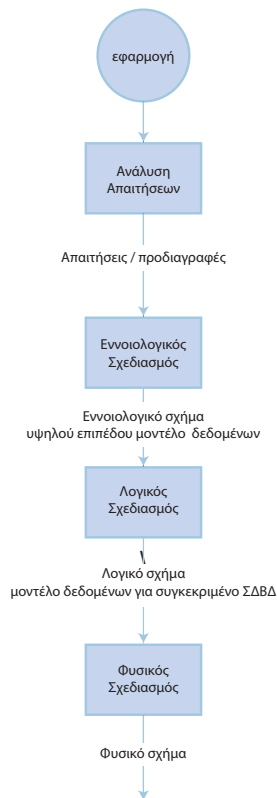
Η περιγραφή των δεδομένων μιας βάσης δεδομένων πραγματοποιείται από το μοντέλο δεδομένων (data models). Αποτελεί ένα σύνολο εργαλείων για την περιγραφή των σχέσεων των δεδομένων, της σημασίας των δεδομένων και των περιορισμών συνέπειας.

Τα μοντέλα δεδομένων περιγράφουν το σχήμα της ΒΔ σε αρκετά υψηλό επίπεδο, χωρίς λεπτομέρειες υλοποίησης. Θα πρέπει να τονιστεί ότι λάθη ή παραλήψεις στο μοντέλο δεδομένων έχουν άμεσο αντίκτυπο στα αποθηκευμένα δεδομένα όσο και στις λειτουργίες επεξεργασίας δεδομένων. Η αλλαγή του μοντέλου δεδομένων συνεπάγεται αλλαγές σε πολλά τμήματα της ΒΔ.

Έχουν αναπτυχθεί δύο μοντέλα δεδομένων τα οποία θα περιγραφούν σε αυτό το κεφάλαιο, το Μοντέλο Οντοτήτων Συσχετίσεων και το Σχεσιακό Μοντέλο. Η ανεξαρτησία των δεδομένων περιγράφεται στην ενότητα 3.2.1 Εννοιολογικό και Λογικό μοντέλο.

3.2.1. Εννοιολογικό, Λογικό και Φυσικό Σχήμα (Μοντέλο)

Για να είναι χρήσιμη μια βάση δεδομένων πρέπει να μπορεί να ανακαλεί τα δεδομένα της αποτελεσματικά. Οι σχεδιαστές των βάσεων για την αποτελεσματική ανάκληση δεδομένων χρησιμοποιούν πολύπλοκες δομές δεδομένων, οι οποίες χρησιμοποιούνται για την αναπαράσταση των δεδομένων της βάσης δεδομένων. Όμως, οι χρήστες μιας ΒΔ δεν είναι ανάγκη να είναι έμπειροι προγραμματιστές, για αυτό το λόγο οι σχεδιαστές κρύβουν την πολυπλοκότητα από τους χρήστες μέσω διαφόρων επιπέδων αφαιρετικότητας. Για να επιτευχθεί η απόκρυψη της πολυπλοκότητας από τους χρήστες χρησιμοποιούνται τρία σχήματα (βλέπε εικόνα 3.1) το εννοιολογικό σχήμα, το λογικό σχήμα και το φυσικό σχήμα. Στη βιβλιογραφία το εννοιολογικό και το λογικό σχήμα αναφέρονται πολλές φορές ως ένα σχήμα για αυτό το λόγο στην παρούσα ενότητα το λογικό σχήμα αναφέρεται και ως λογικό (εννοιολογικό) σχήμα.



Εικόνα 3.1 Εννοιολογικό, Λογικό και Φυσικό σχήμα

Το χαμηλότερο επίπεδο αφαιρετικότητας (**φυσικό σχήμα**) περιγράφει τον τρόπο αποθήκευσης των δεδομένων. Σε αυτό περιγράφονται με λεπτομέρεια οι πολύπλοκες δομές που χρησιμοποιούνται για την αποθήκευση και την ανάκτηση των δεδομένων.

Το **λογικό (εννοιολογικό) σχήμα** έχει υψηλότερο επίπεδο αφαιρετικότητας και περιγράφει ποια δεδομένα αποθηκεύονται και ποιες είναι οι σχέσεις μεταξύ τους. Στην ουσία, το λογικό σχήμα περιγράφει τη βάση δεδομένων με απλές δομές. Αυτό εξασφαλίζει ότι αν και μπορεί να απαιτούνται πολύπλοκες δομές στο φυσικό σχήμα ο χρήστης της ΒΔ δεν χρειάζεται να τις γνωρίζει.

Το υψηλότερο επίπεδο αφαιρετικότητας (**εννοιολογικό σχήμα**) περιγράφει μόνο ένα κομμάτι της ΒΔ. Αν και στο λογικό σχήμα τα δεδομένα περιγράφονται με απλές δομές ο χρήστης δεν χρειάζεται να τις γνωρίζει. Με αυτόν τον τρόπο η συνδιαλλαγή με το σύστημα της ΒΔ γίνεται πιο απλή.

Διαφορετικοί χρήστες έχουν προσπέλαση σε διαφορετικά δεδομένα της ΒΔ, ανάλογα με τους περιορισμούς και τους κανόνες προσπέλασης που εφαρμόζονται.

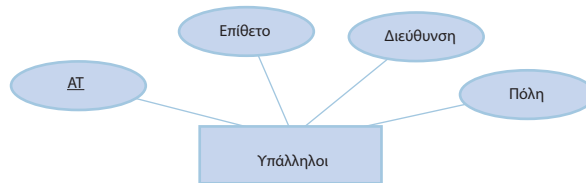
Ένας από τους πιο σημαντικούς λόγους για την χρήση ΣΔΒΔ είναι η υποστήριξη της **ανεξαρτησίας** των δεδομένων. Αυτό σημαίνει ότι ο κώδικας δεν επηρεάζεται από μεταβολές στον τρόπο οργάνωσης και καταχώρισης των δεδομένων. Η ανεξαρτησία των δεδομένων γίνεται μέσα από τα τρία επίπεδα αφαιρετικότητας που έχουν περιγραφεί. Ο χρήστης προστατεύεται από αλλαγές στη λογική δομή των δεδομένων μέσω της λογικής ανεξαρτησίας των δεδομένων. Με ανάλογο τρόπο το λογικό σχήμα απομονώνει τον χρήστη από αλλαγές που συμβαίνουν στο φυσικό σχήμα (για παράδειγμα, αλλαγές που αφορούν την αποθήκευση των δεδομένων στην περιφερειακή μνήμη, επιλογές ευρετηρίων, κ.α). Όσο το εννοιολογικό σχήμα μένει ανεπηρέαστο, η υλοποίηση των παραπάνω μπορεί να αλλάζει χωρίς να χρειάζεται αλλαγή στον κώδικα. Βέβαια, αυτές οι αλλαγές μπορεί να έχουν αρνητικές επιπτώσεις στις επιδόσεις του συστήματος.

3.2.2. Μοντέλο Οντοτήτων Συσχετίσεων

Το μοντέλο οντοτήτων συσχετίσεων βασίζεται στο πως αντιλαμβάνεται κάποιος τον φυσικό κόσμο. Αποτελείται από ένα σύνολο αντικειμένων του φυσικού κόσμου που ονομάζονται οντότητες στο συγκεκριμένο μοντέλο και από τις σχέσεις των αντικειμένων. Εκτός από τις οντότητες και τις σχέσεις, κάθε μοντέλο οντοτήτων συσχετίσεων αντιπροσωπεύει και κάποιους περιορισμούς που υπάρχουν και πρέπει τα περιεχόμενα της βάσης να υπακούουν. Μία οντότητα περιγράφεται από ένα σύνολο γνωρισμάτων/χαρακτηριστικών που την χαρακτηρίζουν.

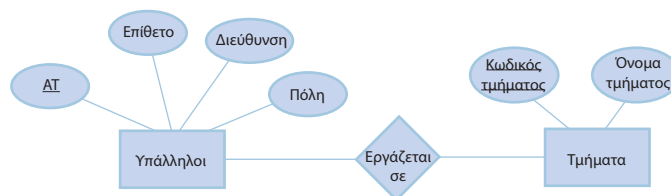
Το μοντέλο οντοτήτων συσχετίσεων αποτελεί το στάδιο του εννοιολογικού σχεδιασμού μιας βάσης δεδομένων (ενότητα 3.1) και περιγράφει το λογικό (εννοιολογικό) σχήμα της βάσης δεδομένων.

Εμείς στο παρόν κεφάλαιο θα παρουσιάσουμε μόνο το σχεσιακό μοντέλο ΣΔΒΔ, οπότε ο λογικός σχεδιασμός σημαίνει μετασχηματισμό του διαγράμματος οντοτήτων συσχετίσεων σε σχήμα σχεσιακής βάσης δεδομένων. Το μοντέλο οντοτήτων συσχετίσεων περιγράφεται από το διάγραμμα οντοτήτων συσχετίσεων που περιγράφεται λεπτομερώς στην ενότητα 3.3. Παράδειγμα διαγραμματικής απεικόνισης του μοντέλου παρουσιάζεται στην εικόνα 3.2 που αφορά το σύνολο οντοτήτων Υπάλληλοι με γνωρίσματα όνομα, Αριθμός Αστυνομικής ταυτότητας ΑΤ, διεύθυνση και πόλη κατοικίας.



Εικόνα 3.2 Σχηματική αναπαράσταση της οντότητας Υπάλληλοι

Οι συσχετίσεις ανάμεσα στις οντότητες αναπαριστώνται με ρόμβο. Παράδειγμα παρουσιάζεται στην εικόνα 3.3.



Εικόνα 3.3 Σχηματική αναπαράσταση της συσχέτισης

Στην εικόνα 3.3 περιγράφονται οι οντότητες Υπάλληλοι και Τμήματα. Τα τμήματα αντιπροσωπεύουν τα τμήματα που εργάζονται οι υπάλληλοι στον φυσικό κόσμο (Λογιστήριο, Τμήμα προσωπικού, τμήμα πωλήσεων κ.α.). Η σχέση ανάμεσα στις δύο οντότητες περιγράφεται από τη συσχέτιση "Εργάζεται σε". Το μοντέλο οντοτήτων συσχετίσεων χρησιμοποιείται ευρέως στην ανάλυση των απαιτήσεων μιας βάσης δεδομένων.

Το χαρακτηριστικό το οποίο προσδιορίζει μοναδικά μία οντότητα καλείται πρωτεύον κλειδί. Το πρωτεύον κλειδί σημειώνεται σε ένα διάγραμμα ΟΣ με υπογράμμιση. Στην οντότητα Μαθητής ο αριθμός μητρώου προσδιορίζει μοναδικά τον κάθε μαθητή, επομένως, το πρωτεύον κλειδί είναι το χαρακτηριστικό ΑΜ. Σε πολλές περιπτώσεις όμως, δεν υπάρχει κάποιο χαρακτηριστικό που από μόνο του να αποτελεί κλειδί. Εναλλακτικά μπορούμε να χρησιμοποιήσουμε δύο ή περισσότερα χαρακτηριστικά (εάν υπάρχουν) που στο σύνολό τους προσδιορίζουν μοναδικά μία οντότητα. Στην περίπτωση αυτή το κλειδί καλείται σύνθετο κλειδί.

Όταν δεν υπάρχει κάποιο χαρακτηριστικό που να μπορεί να χρησιμοποιηθεί ως κλειδί, τότε μπορούμε να ορίσουμε κάποιο τεχνητό χαρακτηριστικό που να προσδιορίζει μοναδικά κάθε οντότητα. Το χαρακτηριστικό αυτό συνήθως ονομάζεται κωδικός ή έχει κάποιο άλλο προσδιοριστικό

Στο παράδειγμα της εικόνας 3.3. έχουμε χρησιμοποιήσει την ονομασία κωδικός τμήματος για κλειδί της οντότητας Τμήματα. Σημειώνεται ότι ένα σύνολο οντοτήτων μπορεί να έχει περισσότερα από ένα κλειδιά. Ένα από τα κλειδιά χαρακτηρίζεται ως πρωτεύον κλειδί ενώ τα υπόλοιπα ως δευτερεύοντα κλειδιά. Δευτερεύον κλειδί της οντότητας μαθητής μπορεί να είναι το επίθετο.

3.2.3 Σχεσιακό Μοντέλο

Το σχεσιακό μοντέλο χρησιμοποιείται ευρέως και πολλά συστήματα βάσης δεδομένων βασίζονται σε αυτό. Είναι σε χαμηλότερο επίπεδο αφαιρετικότητας από το μοντέλο οντοτήτων συσχετίσεων. Χαρακτηρίζεται από απλότητα και είναι ελκυστικό στη χρήση του. Το σχεσιακό μοντέλο δεδομένων (relational data model) αναπτύχθηκε από τον Codd το 1970. Τα περισσότερα εμπορικά ΣΔΒΔ υποστηρίζουν το σχεσιακό μοντέλο δεδομένων ή επεκτάσεις αυτού.

Οι βασικοί στόχοι του σχεσιακού μοντέλου είναι:

- Η υποστήριξη της ανεξαρτησίας των δεδομένων, έτσι ώστε αλλαγές στη φυσική δομή να μην απαιτούν αλλαγές στα προγράμματα εφαρμογής,
- Η αποφυγή της επανάληψης δεδομένων, όταν τα ίδια δεδομένα αποθηκεύονται πολλές φορές σε διαφορετικές περιοχές της ΒΔ.

Το σχεσιακό μοντέλο αποτελείται από πίνακες που κάθε πίνακας έχει ένα μοναδικό όνομα και προσδιορίζεται από ένα σύνολο γραμμών και ένα σύνολο στηλών. Κάθε γραμμή του πίνακα αναπαριστά μία εγγραφή δεδομένων, ενώ οι στήλες του πίνακα ορίζουν τα χαρακτηριστικά ή ιδιότητες της κάθε εγγραφής. Κάθε πίνακας περιέχει εγγραφές ενός συγκεκριμένου τύπου και κάθε τύπος εγγραφής ορίζει ένα σταθερό αριθμό πεδίων ή ιδιοτήτων. Ένας πίνακας μπορεί να χρησιμοποιηθεί τόσο για τις εγγραφές που περιγράφουν οντότητες, όσο και για τις εγγραφές που περιγράφουν συσχετίσεις. Συνήθως, θεωρούμε ότι οι στήλες είναι διατεταγμένες και συνηθίζουμε να αναφέρουμε πρώτα τις στήλες που αντιστοιχούν σε χαρακτηριστικά που είναι κλειδιά. Στην εικόνα 3.4 παρουσιάζεται παράδειγμα πίνακα σχεσιακού μοντέλου.

ΑΤ Υπαλλήλου	Επίθετο Υπαλλήλου	Διεύθυνση κατοικίας	Πόλη
AA 14355	Παπαδόπουλος	Βενιζέλου 4	Θεσσαλονίκη
AB 78155	Χαρισίδης	Τσιμισκή 19	Θεσσαλονίκη
AB 15889	Κερκινάκης	Εγνατία 151	Θεσσαλονίκη

Εικόνα 3.4 Πίνακας υπάλληλος

Στην εικόνα 3.4 παρουσιάζεται ο πίνακας υπαλλήλων μιας εταιρείας όπου ο υπάλληλος που προσδιορίζεται από τον αριθμό της Αστυνομικής ταυτότητας AB 15889 ονομάζεται Κερκινάκης, κατοικεί στην Θεσσαλονίκη και μένει στην διεύθυνση Εγνατίας 151.

Για κάθε χαρακτηριστικό/ιδιότητα υπάρχει ένα σύνολο επιτρεπτών τιμών, το οποίο καλείται πεδίο ορισμού του χαρακτηριστικού/ ιδιότητας.

Μπορούμε να δημιουργήσουμε διατάξεις στο σχεσιακό μοντέλο όπου υπάρχει πρόβλημα δηλαδή όπου εμφανίζονται διπλές πληροφορίες. Για παράδειγμα, αν θεωρήσουμε ότι στον πίνακα υπάλληλος θα πρέπει να συμπεριλάβουμε τα ονόματα των παιδιών του κάθε υπαλλήλου και ο υπάλληλος Κερκινάκης έχει δύο παιδιά, θα έπρεπε να αποθηκεύσουμε 2 γραμμές στον πίνακα όπως φαίνεται στην εικόνα 3.5. Όμως, με αυτή τη διαδικασία δημιουργούνται διπλές πληροφορίες όπως το επίθετο, η διεύθυνση και η πόλη που είναι οι ίδιες και στις δύο γραμμές. Στην ενότητα 3.4 θα μελετήσουμε πως μπορούμε να ξεχωρίσουμε τις καλές σχεδιάσεις διατάξεων από τις "κακές".

ΑΤ Υπαλλήλου	Επίθετο	Διεύθυνση	Πόλη	Όνομα παιδιού
AA 14355	Παπαδόπουλος	Βενιζέλου 4	Θεσσαλονίκη	NULL
AB 78155	Χαρισίδης	Τσιμισκή 19	Θεσσαλονίκη	Άννα
AB 15889	Κερκινάκης	Εγνατία 151	Θεσσαλονίκη	Κωνσταντίνος
AB 15889	Κερκινάκης	Εγνατία 151	Θεσσαλονίκη	Γεώργιος

Εικόνα 3.5 Πίνακας υπαλλήλων με διπλές πληροφορίες

Υπάρχουν περιπτώσεις στις οποίες δε γνωρίζουμε την τιμή ενός χαρακτηριστικού. Στην εικόνα 3.5, παρατηρούμε ότι εφόσον ο Παπαδόπουλος δεν έχει παιδιά, η ιδιότητα Όνομα παιδιού έχει την τιμή **NULL** (κενή τιμή). Η κενή τιμή είναι διαφορετική από 0 ή από τον χαρακτήρα <<SPACE>>. Μία κενή τιμή συνήθως δε δημιουργεί πρόβλημα, ωστόσο καλό είναι οι τιμές NULL να αποφεύγονται διότι δημιουργούνται δυσκολίες στη διατύπωση ερωτημάτων. Τονίζεται ότι δεν επιτρέπονται να υπάρχουν κενές τιμές στα κλειδιά ενός πίνακα.

Τα μοντέλα οντοτήτων συσχετίσεων και το σχεσιακό μοντέλο παρουσιάζουν πολλές ομοιότητες. Για παράδειγμα, μία οντότητα του μοντέλου ΟΣ αντιστοιχεί σε μία γραμμή ενός πίνακα στο σχεσιακό μοντέλο. Ένα σύνολο οντοτήτων αντιστοιχεί αντίστοιχα σε έναν πίνακα του σχεσιακού μοντέλου.

Ως συνήθως το μοντέλο οντοτήτων συσχετίσεων μεταφράζεται σε σχεσιακό μοντέλο από τους σχεδιαστές μιας βάσης δεδομένων (βλέπε κεφάλαιο 3.5.2).



Ερωτήσεις / Δραστηριότητες

1. Τι είναι ένα μοντέλο δεδομένων; Να περιγραφούν οι βασικές κατηγορίες μοντέλων και να δώσετε ένα παράδειγμα σχεσιακού μοντέλου.
2. Ποιο κλειδί ονομάζεται πρωτεύον και για ποιους λόγους κατά τη γνώμη σας είναι σημαντικό να υπάρχει;
3. Περιγράψτε τους λόγους που καθιστούν την ανεξαρτησία των δεδομένων σημαντική.
4. Εξηγήστε τις διαφορές ανάμεσα στα τρία σχήματα εννοιολογικό, λογικό και φυσικό.
5. Να δώσετε ένα παράδειγμα όπου η χρήση κενών τιμών (NULL) δημιουργεί πρόβλημα στις τιμές ενός χαρακτηριστικού / ιδιότητας.
6. Για ποιους λόγους μπορεί να δημιουργηθούν διπλές πληροφορίες; Δώστε ένα παράδειγμα.
7. Ποιες είναι οι διαφορές ανάμεσα στο Μοντέλο Οντοτήτων Συσχετίσεων και του Σχεσιακού Μοντέλου;
8. Το σχεσιακό μοντέλο θεωρείται απλό. Μπορείτε να εξηγήσετε τους λόγους;

Δραστηριότητα 1: Εννοιολογικό ή φυσικό σχήμα

Μελετήστε το παρακάτω σενάριο. Αναγνωρίστε το εννοιολογικό από το φυσικό μοντέλο. Οι συμμαθητές της Ελένης γύρω στα 24 άτομα της ετοιμάζουν πάρτι έκπληξη για τα γενέθλιά της. Θέλουν να αγοράσουν διάφορα αναψυκτικά και χυμούς για το πάρτι, όπως πορτοκαλάδες, λεμονάδες, χυμούς και νερά. Σκέφτονται να αγοράσουν 6 συσκευασίες των 4 τεμαχίων. Όταν πάνε στο σουπερμάρκετ ανακαλύπτουν ότι τα αναψυκτικά πωλούνται σε συσκευασίες των 2, 4 και 8 τεμαχίων και πρέπει να επιλέξουν την πιο συμφέρουσα λύση για αυτούς.

Πρόσθετη βοήθεια

Το εννοιολογικό σχήμα είναι κάτι σαν το όνειρο, σαν μία ιδέα που έχουμε στο νου μας. Όταν το όνειρο ή η ιδέα αυτή υλοποιείται, τότε μιλάμε για φυσικό σχήμα. Το όνειρο αυτό το οποίο έχουμε φτιάξει έχει μία δομή, δηλαδή κάποια βασικά στοιχεία που συνδέονται μεταξύ τους. Η δομή αυτή μπορεί να περιγράφει από ένα σχήμα.

Δραστηριότητα 2: Πίνακας Μαθητής

Πρόκειται να σχεδιάσετε το Μαθητολόγιο του σχολείου σας. Σας δίνεται ο παρακάτω πίνακας Μαθητής, αλλά λείπουν πληροφορίες που πρέπει να αποτυπωθούν στον πίνακα, ώστε να γίνει πληρέστερος. Μπορείτε να σκεφτείτε κάποιες από αυτές και να τις αποτυπώσετε ως χαρακτηριστικά/ιδιότητες του πίνακα;

Επίθετο	Διεύθυνση	Πόλη
Παπαδόπουλος	Βενιζέλου 4	Θεσσαλονίκη
Χαρισίδης	Τσιμισκή 19	Θεσσαλονίκη
Κερκινάκης	Εγνατία 151	Θεσσαλονίκη

Πρόσθετη βοήθεια

Μην ξεχάσετε να προσθέσετε το πρωτεύον κλειδί. Ποιο χαρακτηριστικό μπορεί να είναι μοναδικό ώστε να διαχωρίζει τις γραμμές του πίνακα;

3.3 Διαγραμματική Αναπαράσταση Συσχετίσεων Οντοτήτων

Γενικά

Το μοντέλο οντοτήτων συσχετίσεων (ERM) αποτελεί την βάση του διαγράμματος οντοτήτων συσχετίσεων (ERD). Το διάγραμμα οντοτήτων συσχετίσεων αναπαριστά με εννοιολογικό τρόπο τις βάσεις δεδομένων. Απεικονίζει τα κύρια συστατικά των βάσεων δεδομένων: οντότητες (entities), ιδιότητες γνωρίσματα (attributes) και συσχετίσεις (relationships). Η οντότητα αντιπροσωπεύει τόσο συγκεκριμένα αντικείμενα όσο και αφηρημένα γεγονότα και καταστάσεις



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να προσδιορίζει οντότητες, ιδιότητες γνωρίσματα και κλειδιά από τον χώρο της καθημερινής ζωής
- να διακρίνει οντότητα από στιγμιότυπο να διακρίνει το γνώρισμα ιδιότητα από την τιμή του
- να διακρίνει τα υποχρεωτικά και τα προαιρετικά γνωρίσματα να επιλέγει πρωτεύοντα κλειδιά για μία οντότητα
- να συσχετίζει τις συναφείς οντότητες μεταξύ τους προσδιορίζοντας τον πληθάρισμο (cardinality) και το στοιχείο του προαιρετικού (optionality)
- να επιλύει τις σχέσεις «πολλά προς πολλά» μεταξύ των οντοτήτων να αντιλαμβάνεται την δομή του μοντέλου οντοτήτων συσχετίσεων
- να προσδιορίζει οντότητες και να μοντελοποιεί τις μεταξύ τους σχέσεις καθώς και τα χαρακτηριστικά τους σε μία μελέτη περίπτωσης
- να δημιουργεί το διάγραμμα οντοτήτων συσχετίσεων

3.3.1. Οντότητες, Στιγμιότυπα, Γνωρίσματα - Ιδιότητες και Κλειδιά

Ένα **στιγμιότυπο (instance)** είναι ένα «αντικείμενο» του πραγματικού κόσμου το οποίο ξεχωρίζει από τα άλλα αντικείμενα. Για παράδειγμα, στο σχολικό περιβάλλον κάθε μαθητής ή κάθε καθηγητής είναι ένα στιγμιότυπο. Ο μαθητής «Ανδρέας Παπαϊωάννου» και ο καθηγητής «Στέφανος Λαζάρου» αποτελούν στιγμιότυπα. Κάθε στιγμιότυπο έχει ένα σύνολο από ιδιότητες, όπως όνομα, επίθετο, διεύθυνση και αριθμός μητρώου ή ταυτότητας. Οι τιμές για κάποιες από τις ιδιότητες αυτές είναι δυνατό να είναι διαφορετικές για κάθε στιγμιότυπο. Για παράδειγμα ο αριθμός μητρώου είναι μοναδικός για κάθε μαθητή. Δεν θα μπορούσαμε όμως να ισχυριστούμε το ίδιο και για το όνομα, το επίθετο ή την διεύθυνση.

Μία **οντότητα (entity)** είναι ένα σύνολο από στιγμιότυπα του ίδιου τύπου που έχουν κοινές ιδιότητες ή γνωρίσματα. Για παράδειγμα, το σύνολο των καθηγητών ενός σχολείου αποτελούν την οντότητα Καθηγητής. Ο καθηγητής «Στέφανος Λαζάρου» ανήκει στην οντότητα Καθηγητής.

Αποτελεί δηλαδή ένα στιγμιότυπο της οντότητας Καθηγητής. Παρόμοια, η οντότητα Μαθητής αναφέρεται στο σύνολο όλων των μαθητών ενός σχολείου. Ο μαθητής «Ανδρέας Παπαϊωάννου» είναι ένα στιγμιότυπο της οντότητας Μαθητής.

Μία οντότητα μπορεί να είναι κάτι που έχει μία φυσική υπόσταση όπως ένα πρόσωπο ή ένα βιβλίο ή μπορεί να είναι κάτι πιο γενικό, πιο αφηρημένο όπως ένα μάθημα, μία συναυλία ή μία δεξιότητα.

Μία οντότητα είναι κάτι που είναι «σημαντικό» για μία επιχείρηση ή έναν οργανισμό και για το οποίο η επιχείρηση ή ο οργανισμός θέλει να ξέρει ορισμένα βασικά πράγματα. Για παράδειγμα, ένα σχολείο χρειάζεται να αποθηκεύσει δεδομένα σχετικά με τους Μαθητές, τους Καθηγητές, τα Μαθήματα και τους Βαθμούς. Οι Μαθητές, οι Καθηγητές, τα Μαθήματα και οι Βαθμοί για το σχολείο θεωρούνται οντότητες, βασικά δηλαδή πράγματα για τα οποία υπάρχει ανάγκη να αποθηκεύσουμε κάποιες πληροφορίες. Πρέπει να γνωρίζουμε δηλαδή ποιοι είναι οι μαθητές μας και ποιά μαθήματα παρακολουθούν. Η οντότητα μπορεί να θεωρηθεί και ως σύνολο από στιγμιότυπα. Σε ένα σχολείο η οντότητα Μαθητής μπορεί να έχει ως στιγμιότυπα τον Παύλο, την Ελένη, τον Νίκο και τον Άλεξ.

Στον παρακάτω πίνακα παρουσιάζονται παραδείγματα οντοτήτων και αντίστοιχων στιγμιότυπων.

Οντότητες	Στιγμιότυπα
Πρόσωπο	Μέγας Αλέξανδρος, Θόδωρος Κολοκοτρώνης, Μαρία Παπαπέτρου
Προϊόν	Nikes Tzampadan
Είδος Προϊόντος	Αθλητικά
Εργασία	Μηχανικός, Γιατρός
Αγορά	Το βιβλίο που αγόρασα χθες
Ζώο	Γάτα, Σκύλος, Λιοντάρι
Εκλογές	Για το ευρωκοινοβούλιο το επόμενο φθινόπωρο

Εικόνα 3.3.1: Παραδείγματα οντοτήτων και αντίστοιχων στιγμιότυπων

Θα πρέπει να σημειώσουμε ότι το προσδιοριστικό που δίνουμε σε ένα αντικείμενο για παράδειγμα αν θα είναι οντότητα ή στιγμιότυπο εξαρτάται κάθε φορά από το πλαίσιο στο οποίο αναφερόμαστε. Για παράδειγμα, ο σκύλος είναι οντότητα ή στιγμιότυπο; Εξαρτάται. Αν αναφερόμαστε στα διάφορα είδη ζώων, τότε έχει νόημα να ορίσουμε την οντότητα Ζώο που θα έχει ως στιγμιότυπα τον Σκύλο, την Γάτα και άλλα. Αν όμως αναφερόμαστε σε μία επιχείρηση εκτροφής και αναπαραγωγής σκύλων, τότε είναι φυσικό να ορίσουμε την οντότητα Σκύλος η οποία μπορεί να έχει ως στιγμιότυπα το Τεριέ, το Κανίς, το Λαμπραντόρ και άλλα.

Κάθε οντότητα περιγράφεται από ένα σύνολο από **γνωρίσματα (attributes)** ή **ιδιότητες**. Για παράδειγμα

μα, η οντότητα Μαθητής έχει ως γνωρίσματα τον αριθμό μητρώου, το όνομα, το επίθετο, την διεύθυνση και το τμήμα. Όταν τα γνωρίσματα αυτά πάρουν συγκεκριμένες τιμές, τότε αναφερόμαστε στα στιγμιότυπα της αντίστοιχης οντότητας. Για παράδειγμα, ένας συγκεκριμένος μαθητής, δηλαδή ένα στιγμιότυπο της οντότητας Μαθητής, μπορεί να έχει ως αριθμό μητρώου το 4589543, ως όνομα το Ανδρέας, ως επίθετο το Παπαϊωάννου, ως διεύθυνση το Εγνατίας 12 και ως τμήμα το Β1. Δηλαδή, κάθε στιγμιότυπο μιας οντότητας έχει συγκεκριμένες **τιμές (values)** για τα αντίστοιχα γνωρίσματα της οντότητας.

Τα γνωρίσματα για τα οποία είναι υποχρεωτικό να έχουν τιμή, ονομάζονται **υποχρεωτικά γνωρίσματα (mandatory attributes)**. Για παράδειγμα, στις περισσότερες επιχειρήσεις είναι απαραίτητο να είναι γνωστά τα ονοματεπώνυμα των ατόμων που διαδραματίζουν κάποιο ρόλο στις διαδικασίες της επιχείρησης. Άλλα γνωρίσματα, όπως είναι για παράδειγμα το κινητό τηλέφωνο, είναι προαιρετικό να έχουν μία τιμή, εκτός και αν πρόκειται για εταιρείες κινητής τηλεφωνίας. Τα γνωρίσματα αυτά λέγονται **προαιρετικά γνωρίσματα (optional attributes)**.

Στον παρακάτω πίνακα παρουσιάζονται παραδείγματα οντοτήτων και αντίστοιχων γνωρισμάτων.

Οντότητες	Γνωρίσματα
Αυτοκίνητο	Μοντέλο, Ιπποδύναμη, Τιμή
Εργασία	Τίτλος, Περιγραφή
Συναλλαγή	Ποσό, Ημερομηνία
Εργασία	Μηχανικός, Γιατρός
Υπάλληλος	Ονοματεπώνυμο, Διεύθυνση, Κινητό, Αριθμός Ταυτότητας, email

Εικόνα 3.3.2: Παραδείγματα οντοτήτων και αντίστοιχων γνωρισμάτων

Αν τιμές που μπορεί να πάρει ένα γνώρισμα είναι μοναδικές για κάθε στιγμιότυπο, τότε το γνώρισμα αυτό λέγεται **πρωτεύον κλειδί (unique identifier)**. Για παράδειγμα, ο αριθμός μητρώου για την οντότητα Μαθητής αποτελεί το κλειδί της οντότητας αυτής, διότι δεν υπάρχουν δύο μαθητές με τον ίδιο αριθμό μητρώου.



Ερωτήσεις / Δραστηριότητες

Δραστηριότητα: Στιγμιότυπο (instance) ή οντότητα (entity);

Σκοπός

Σκοπός αυτής της δραστηριότητας είναι να μάθετε να ξεχωρίζετε την οντότητα, από το γνώρισμα και το στιγμιότυπο.

Οδηγίες

Ποια από τις παρακάτω έννοιες που παρουσιάζονται στον πίνακα που ακολουθεί είναι οντότητα, γνώρισμα ή στιγμιότυπο; Αν νομίζετε ότι μία έννοια αποτελεί οντότητα, τότε δώστε και ένα αντίστοιχο παράδειγμα από ένα στιγμιότυπο αυτής της οντότητας. Αν νομίζετε ότι μία έννοια αποτελεί γνώρισμα ή στιγμιότυπο, τότε δώστε ένα παράδειγμα αντίστοιχης οντότητας. Συμπληρώστε τις τελευταίες τρεις γραμμές με αντίστοιχες έννοιες.

Έννοια	Οντότητα/Γνώρισμα/ Στιγμιότυπο	Παράδειγμα οντότητας ή στιγμιότυπου
Πρόεδρος		
Θόδωρος Κολοκοτρώνης		
Γάτα		
Ζώο		
Ύψος		
Κινητό τηλέφωνο		
	Οντότητα	Μπισκότο
	Γνώρισμα Μπισκότο	Μπισκότο
	Στιγμιότυπο Μπισκότο	Μπισκότο

Εικόνα 3.3.3: Παραδείγματα οντοτήτων και αντίστοιχων γνωρισμάτων και στιγμιότυπων

Πρόσθετη βοήθεια

Θυμηθείτε ότι την οντότητα ΜΑΘΗΤΗΣ. Στο τμήμα σας είστε όλοι μαθητές. Όμως, κάθε ένας από εσάς αποτελεί ένα μοναδικό στιγμιότυπο της οντότητας αυτής.

Επέκταση

Θυμηθείτε την τελευταία φορά που πήγατε με τους φίλους σας σε ένα καφέ για να πιείτε για παράδειγμα ένα καφέ, για να φάτε ένα σάντουιτς, για να παίξετε ένα επιτραπέζιο παιχνίδι, για να παρακολουθήσετε ένα ποδοσφαιρικό αγώνα ή για να «μπείτε» στο διαδίκτυο. Συζητήστε με τους συμμαθητές σας για τις προσφερόμενες υπηρεσίες.

Ονομάστε τις οντότητες που αποτελούν τα κύρια συστατικά της επιχείρησης. Δώστε παραδείγματα στιγμιότυπων από κάθε οντότητα.

3.3.2. Συσχετίσεις μεταξύ Οντοτήτων - Τύποι Συσχετίσεων

Ποια είναι η σχέση ανάμεσα σε εσάς και την θεία σας, τον θείο σας, τα ξαδέρφια σας, τη γιαγιά σας κλπ; Πώς θα μπορούσατε να εξηγήσετε σε κάποιον τρίτο ότι η Ιωάννα είναι ξαδέλφη σας; Είναι απαραίτητο για την κοινωνία μας να κατηγοριοποιεί τις σχέσεις μεταξύ των ανθρώπων ή της αρκεί να αναγνωρίζει τα άτομα μόνο με το όνομά τους;

Για παράδειγμα, φανταστείτε ότι θέλετε να συστήσετε σε κάποιον την ξαδέλφη σας την Ιωάννα, χωρίς την αναφορά της λέξης «ξαδέλφη» ή κάποια άλλη παρεμφερή. Θα λέγατε τότε «Αυτή είναι η Ιωάννα, το παιδί της γυναίκας που έχει τους ίδιους γονείς με τους γονείς μου». Πολύ μεγάλη εισαγωγή και κουραστική. Δεν θα ήταν προτιμότερο να πείτε «Αυτή είναι η Ιωάννα, η κόρη της θείας μου» ή πιο απλά ακόμη «Αυτή είναι η Ιωάννα, ξαδέλφη μου»;

Το παραπάνω καθημερινό παράδειγμα των οικογενειακών σχέσεων συναντάται και στον χώρο των επιχειρήσεων και των οργανισμών. Γενικά, οι **σχέσεις (relationships)** μας βοηθούν να δούμε πώς τα διάφορα μέρη ενός συστήματος επηρεάζουν το ένα το άλλο. Για παράδειγμα, οι οντότητες ΜΑΘΗΤΗΣ (STUDENT) και ΜΑΘΗΜΑ (COURSE) συνδέονται μεταξύ τους. Κάθε μαθητής μπορεί να επιλέξει, για παράδειγμα το μάθημα «Εφαρμογές Πληροφορικής». Για να μοντελοποιήσουμε με ακρίβεια την επιχείρηση ή τον οργανισμό, οι σχέσεις μεταξύ των οντοτήτων είναι εξίσου σημαντικές με τις ίδιες τις οντότητες.

Στη συνέχεια παραθέτουμε κάποια παραδείγματα σχέσεων μεταξύ διαφόρων οντοτήτων:

Οι ΕΡΓΑΖΟΜΕΝΟΙ έχουν ΘΕΣΕΙΣ ΕΡΓΑΣΙΑΣ

Οι ΘΕΣΕΙΣ ΕΡΓΑΣΙΑΣ κατέχονται από ΕΡΓΑΖΟΜΕΝΟΥΣ

Τα ΠΡΟΪΟΝΤΑ κατατάσσονται σύμφωνα με τον ΤΥΠΟ ΤΟΥ ΠΡΟΪΟΝΤΟΣ

Ο ΤΥΠΟΣ ΤΟΥ ΠΡΟΪΟΝΤΟΣ αποτελεί μια κατηγοριοποίηση για ένα ΠΡΟΪΟΝ Οι ΑΝΘΡΩΠΟΙ κάνουν ΚΡΑΤΗΣΕΙΣ ΕΙΣΙΤΗΡΙΩΝ

Οι ΚΡΑΤΗΣΕΙΣ ΕΙΣΙΤΗΡΙΟΥ γίνονται από ΑΝΘΡΩΠΟΥΣ

Οι σχέσεις είναι χαρακτηρίζονται ως **υποχρεωτικές (mandatory)** ή **προαιρετικές (optional)**. Θεωρήστε τις δύο οντότητες ΕΡΓΑΖΟΜΕΝΟΣ (EMPLOYEE) και ΘΕΣΗ ΕΡΓΑΣΙΑΣ (JOB). Για να μπορέσουμε να καθορίσουμε το είδος της σχέσης που διέπει τις παραπάνω οντότητες θα πρέπει να απαντήσουμε στα παρακάτω δύο ερωτήματα:

- Πρέπει κάθε εργαζόμενος να έχει μια θέση εργασίας; Με άλλα λόγια, αυτό είναι ένα υποχρεωτικό ή προαιρετικό για κάθε εργαζόμενο;
- Πρέπει κάθε θέση εργασίας να αντιστοιχεί σε έναν υπάλληλο; Με άλλα λόγια, αυτό είναι ένα υποχρεωτικό ή προαιρετικό για κάθε θέση εργασίας;

Σε μία σχέση ένα ή περισσότερα στιγμιότυπα μιας οντότητας μπορούν ή πρέπει να συσχετισθούν με ένα στιγμιότυπο της άλλης οντότητας. Το αν θα είναι ένα ή περισσότερα στιγμιότυπα εκφράζει αυτό που ονομάζουμε ως **πληθάριθος (cardinality)** μιας σχέσης. Ο πληθάριθος μιας σχέσης μετράει την ποσότητα από κάτι, απαντάει δηλαδή στην ερώτηση «Πόσα;». Για παράδειγμα:

- Σε πόσες θέσεις εργασίας μπορεί ένας υπάλληλος να εργάζεται; Σε μία θέση εργασίας μόνο ή σε περισσότερες από μία;
- Πόσοι υπάλληλοι μπορούν να κατέχουν μια συγκεκριμένη θέση εργασίας; Ένας υπάλληλος μόνο ή περισσότεροι από έναν εργαζόμενο;

Ο πληθάριθος μιας σχέσης απαντάει στο ερώτημα αν το πλήθος είναι ένα ή περισσότερα από ένα. Δεν δίνει απάντηση με ένα συγκεκριμένο πλήθος για παράδειγμα τέσσερα (4).

Οι παραπάνω ερωτήσεις είναι πολύ σημαντικές και έχουν αντίκτυπο στον σχεδιασμό της βάσης δεδομένων

και συγκεκριμένα στο μοντέλο συσχετίσεων οντοτήτων. Στο προαναφερθέν παράδειγμα επιλέγουμε τις παρακάτω παραδοχές για τις οποίες θα δούμε στην επόμενη ενότητα πώς απεικονίζονται με διαγραμματικό τρόπο:

Κάθε εργαζόμενος **υποχρεωτικά (mandatory)** πρέπει να έχει **ακριβώς μία (cardinality)** θέση εργασίας.

Μια θέση εργασίας **μπορεί (optionality)** να κατέχεται από **περισσότερους από έναν εργαζόμενους (cardinality)**. Δηλαδή για μία θέση εργασίας μπορεί να έχουμε κανέναν, έναν ή περισσότερους από έναν εργαζόμενους.

Γενικά, τα μοντέλα οντοτήτων συσχετίσεων χρησιμοποιούν τρεις τύπους σχέσεων:

Ένα προς πολλά 1:M

Πολλά προς πολλά M:M

Ένα προς ένα 1:1

Ας μελετήσουμε μερικά παραδείγματα από τις τρεις παραπάνω τύπους σχέσεων.

Ένα προς πολλά 1:M

Ένας ζωγράφος ζωγραφίζει πολλούς πίνακες, αλλά κάθε πίνακας αποτελεί δημιούργημα ενός μόνο ζωγράφου. Η σχέση «Ο ζωγράφος ζωγραφίζει πίνακες» είναι μία σχέση «ένα προς πολλά». Το «ένα» είναι από την πλευρά της οντότητας «ΖΩΓΡΑΦΟΣ», ενώ το «πολλά» είναι από την πλευρά της οντότητας «ΠΙΝΑΚΑΣ».

Πολλά προς πολλά M:M

Ένας εργαζόμενος μπορεί να μάθει πολλές δεξιότητες και κάθε δεξιότητα μπορεί να κατακτηθεί από πολλούς εργαζόμενους. Η σχέση «Ο εργαζόμενος μαθαίνει μία δεξιότητα» είναι μία σχέση «πολλά προς πολλά». Το «πολλά» είναι και από την πλευρά της οντότητας «ΕΡΓΑΖΟΜΕΝΟΣ», αλλά και από την πλευρά της οντότητας «ΔΕΞΙΟΤΗΤΑ».

Ένα προς ένα 1:1

Μία εταιρεία λιανικού εμπορίου αναθέτει την διαχείριση για κάθε ένα από τα καταστήματά της σε ένα υπάλληλο. Κάθε υπεύθυνος καταστήματος διευθύνει ένα μόνο κατάστημα. Η σχέση «Ο εργαζόμενος διευθύνει ένα κατάστημα» είναι μία σχέση «ένα προς ένα». Το «ένα» είναι και από την πλευρά της οντότητας «ΕΡΓΑΖΟΜΕΝΟΣ», αλλά και από την πλευρά της οντότητας «ΚΑΤΑΣΤΗΜΑ».

Επιπλέον **περιορισμοί** που έχουν να κάνουν με τους κανόνες που διέπουν μία επιχείρηση ή έναν οργανισμό, όπως για παράδειγμα οι βαθμοί ενός μαθητή πρέπει να κυμαίνονται μεταξύ 8 και 20, δεν μπορούν να αναπαρασταθούν από έναν εννοιολογικό μοντέλο. Στην περίπτωση αυτή χρειάζεται να γραφτεί κώδικας που θα ικανοποιεί τις απαιτήσεις της επιχείρησης ή του οργανισμού.



Ερωτήσεις / Δραστηριότητες

Δραστηριότητα: Αναγνώριση Τύπων Συσχετίσεων

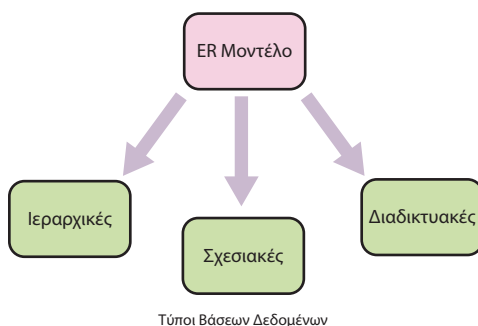
Προσδιορίστε τους διάφορους τύπους σχέσεων από τις παρακάτω προτάσεις:

- Ένας εκπαιδευτής σκι μπορεί να εκπαιδεύσει έναν ή περισσότερους σκιέρ
- Ένα ποδήλατο μπορεί να ανήκει σε ένα παιδί
- Οι μαρκαδόροι μπορούν να χρησιμοποιηθούν από όλους τους μαθητές της τάξης
- Ένα διαβατήριό ανήκει σε ένα πρόσωπο
- Ένα θηλυκός ελέφαντας γεννάει έναν μικρό ελέφαντα

3.3.3 Μοντελοποίηση Συσχετίσεων Οντοτήτων και ER Διάγραμμα

Το **μοντέλο συσχετίσεων οντοτήτων (Entity Relationship model)** είναι ένα εννοιολογικό μοντέλο το οποίο χρησιμοποιείται ευρέως στον σχεδιασμό βάσεων δεδομένων για την κατανόηση και αναπαράσταση των απαιτήσεων δεδομένων μιας επιχείρησης ή ενός οργανισμού.

Η **μοντελοποίηση συσχετίσεων οντοτήτων (Entity Relationship modeling)** είναι ανεξάρτητη από το υλικό ή το λογισμικό που χρησιμοποιείται ή θα χρησιμοποιηθεί στην υλοποίηση, όπως φαίνεται στο παρακάτω σχήμα. Παρόλο που το μοντέλο συσχετίσεων οντοτήτων (Entity Relationship Model) αποτελεί τον θεμελιώδη λίθο για την δημιουργία κάθε τύπου βάσης δεδομένων, όπως των ιεραρχικών (hierarchical databases), των διαδικτυακών (network databases) και των σχεσιακών βάσεων δεδομένων (relational databases) συνδέεται περισσότερο όμως με τις σχεσιακές βάσεις δεδομένων.



Σχήμα 3.3.4: Το μοντέλο οντοτήτων συσχετίσεων είναι ανεξάρτητο από την υλοποίηση

Ένα **μοντέλο συσχετίσεων οντοτήτων (ERM)** είναι μία λίστα οντοτήτων καθώς και σχέσεων μεταξύ των οντοτήτων αυτών. Παρέχει βασικές πληροφορίες, όπως περιγραφές οντοτήτων, τύπους δεδομένων και περιορισμούς. Το **διάγραμμα συσχετίσεων οντοτήτων (ERD)** αποτελεί γραφική αναπαράσταση του μοντέλου συσχετίσεων οντοτήτων. Χρησιμοποιεί μία σειρά από γραφικά στοιχεία τα οποία θα μελετήσουμε στη συνέχεια. Τα γραφικά αυτά στοιχεία διαφέρουν σε ορισμένα σημεία από τα αντίστοιχα που παρουσιάστηκαν στις προηγούμενες ενότητες, όπως είναι αυτό της αναπαράστασης της συσχέτισης. Και στις δύο όμως περιπτώσεις η μεθοδολογία και η προσέγγιση παραμένει η ίδια.

Απεικόνιση Οντότητας

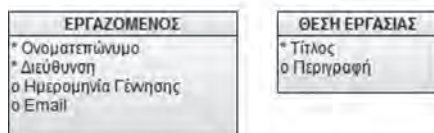
Στο ER διάγραμμα οι οντότητες αναπαρίστανται από πλαίσια μέσα στα οποία αναγράφεται το όνομα της οντότητας, όπως φαίνεται στο παρακάτω σχήμα:



Σχήμα 3.3.5: Απεικόνιση οντοτήτων σε ER διάγραμμα

Απεικόνιση Ιδιοτήτων - Γνωρισμάτων Οντότητας

Τα γνωρίσματα απεικονίζονται μέσα στο πλαίσιο της οντότητας, όπως φαίνεται στο παρακάτω σχήμα:

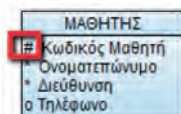


Σχήμα 3.3.6: Απεικόνιση ιδιοτήτων γνωρισμάτων οντοτήτων σε ER διάγραμμα

Τα σύμβολα * και o μπροστά από τα γνωρίσματα υποδηλώνουν την αναγκαιότητα ή μη ύπαρξης τιμής αντίστοιχα στα γνωρίσματα, όταν δημιουργούνται τα στιγμιότυπα της οντότητας. Για παράδειγμα, όταν εισάγεται στο πληροφοριακό σύστημα ένας νέος εργαζόμενος θα πρέπει να είναι γνωστό το ονοματεπώνυμό του και η διεύθυνσή του. Η ημερομηνία γέννησης και το email του μπορούν να συμπληρωθούν και αργότερα. Οι ανάγκες της επιχείρησης απαιτούν την καταγραφή του ονοματεπώνυμου και της διεύθυνσης του εργαζόμενου με την εισαγωγή ενός νέου εργαζομένου.

Απεικόνιση Κλειδιού

Εκείνο το γνώρισμα σε μία οντότητα που έχει τον ρόλο του κλειδιού, δηλαδή προσδιορίζει μονοσήμαντα την οντότητα στο ER διάγραμμα σημειώνεται με το σύμβολο «#», όπως φαίνεται στο παρακάτω σχήμα:



Σχήμα 3.3.7: Απεικόνιση κλειδιού οντότητας σε ER διάγραμμα

Απεικόνιση Συσχετίσεων Οντοτήτων

Οι συσχετίσεις ή σχέσεις μεταξύ των οντοτήτων απεικονίζονται από μια γραμμή, που συνδέει τις δύο οντότητες. Στα δύο άκρα της γραμμής αναγράφονται οι σχέσεις μεταξύ των οντοτήτων, δηλαδή πώς «βλέπει» η κάθε οντότητα τον εαυτό της σε σχέση με την άλλη οντότητα. Στο παρακάτω σχήμα απεικονίζονται οι εξής σχέσεις:

Κάθε εργαζόμενος έχει μία θέση εργασίας

Κάθε θέση εργασίας απασχολεί εργαζομένους



Σχήμα 3.3.8: Απεικόνιση συσχετίσεων οντοτήτων από το χώρο των επιχειρήσεων

Απεικόνιση Πληθάριθμου Σχέσης

Οι σχέσεις μεταξύ των οντοτήτων είναι «ένα» ή «πολλά». Το «ένα» σημαίνει ακριβώς ένα. Τα «πολλά» σημαίνουν ένα ή περισσότερα. Το «ένα» απεικονίζεται με ένα ευθύγραμμο τμήμα στην «απέναντι» οντότητα. Τα «πολλά» απεικονίζεται με ένα τρίποδο στην «απέναντι» οντότητα. Το παράδειγμα που απεικονίζεται στο παρακάτω σχήμα διαβάζεται ως εξής:

Κάθε εργαζόμενος κατέχει ακριβώς μία θέση εργασίας.



Σχήμα 3.3.9: Απεικόνιση πληθάριθμου σχέσης «ένα»

Παρόμοια, το παράδειγμα που απεικονίζεται στο παρακάτω σχήμα διαβάζεται ως εξής:
Σε κάθε θέση εργασίας εργάζονται περισσότεροι από ένας εργαζόμενοι.



Σχήμα 3.3.10: Απεικόνιση πληθάριθμου σχέσης «πολλά»

Απεικόνιση Υποχρεωτικής/Προαιρετικής Σχέσης

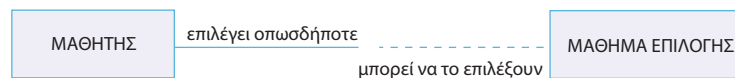
Οι σχέσεις, όπως έχουμε δει, μπορεί να είναι υποχρεωτικές ή προαιρετικές, όπως συμβαίνει και με τα γνωρίσματα. Οι υποχρεωτικές σχέσεις απεικονίζονται από μία συμπαγή γραμμή, ενώ οι προαιρετικές από μία διακεκομμένη γραμμή. Το παρακάτω σχήμα διαβάζεται ως εξής:

- Κάθε εργαζόμενος έχει οπωσδήποτε μία θέση εργασίας.
- Κάθε εργασίας μπορεί να απασχολεί εργαζόμενους. Δεν είναι υποχρεωτικό δηλαδή για μία θέση εργασίας να υπάρχουν οπωσδήποτε εργαζόμενοι».



Σχήμα 3.3.11: Υποχρεωτικές και προαιρετικές συσχετίσεις από το χώρο των επιχειρήσεων

Παρόμοια παραδείγματα συναντάμε και στον σχολικό χώρο με τα μαθήματα επιλογής. Κάθε μαθητής υποχρεούται να πάρει ένα μάθημα επιλογής. Υπάρχουν όμως μαθήματα επιλογής που δεν τα έχει επιλέξει κανείς. Η περίπτωση αυτή μπορεί να αναπαρασταθεί από το παρακάτω σχήμα:



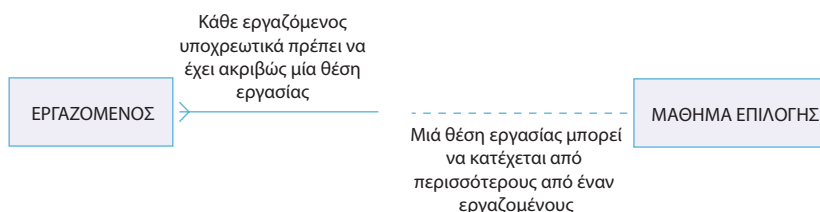
Σχήμα 3.3.12: Υποχρεωτικές και προαιρετικές συσχετίσεις από τον σχολικό χώρο

Ολοκληρωμένη Απεικόνιση Συσχετίσεων Οντοτήτων

Αν συνδυάσουμε όλα τα παραπάνω σε ένα διάγραμμα θα πάρουμε το παρακάτω σχήμα, το οποίο διαβάζεται ως εξής:

Κάθε εργαζόμενος υποχρεωτικά (mandatory) πρέπει να έχει ακριβώς μία (cardinality) θέση εργασίας.

Μια θέση εργασίας μπορεί (optionality) να κατέχεται από περισσότερους από έναν εργαζόμενους (cardinality). Δηλαδή για μία θέση εργασίας μπορεί να έχουμε κανέναν, έναν ή περισσότερους από έναν εργαζόμενους.



Σχήμα 3.3.13: ER διάγραμμα



Ερωτήσεις / Δραστηριότητες

Δραστηριότητα: Μοντελοποίηση αποχώρησης παιδιών από τον παιδικό σταθμό.

Διαβάστε το παρακάτω σενάριο, που συσχετίζει τις οντότητες γονέας και παιδί και μοντελοποιήστε με γραφικό τρόπο τις σχέσεις αυτές. Δώστε δηλαδή κατάλληλα ονόματα και στα δύο άκρα της γραμμής σχέσης και σχεδιάστε την γραμμή σχέσης κατά αυτόν τον τρόπο που υποδηλώνει τυχόν προαιρετικές ή προαιρετικές σχέσεις καθώς και την πληθάρημο της σχέσης.

«Στο τέλος της κάθε ημέρας, οι γονείς πρέπει να πάρουν τα παιδιά τους από τον παιδικό σταθμό. Όλα τα παιδιά πρέπει να έχουν φύγει το αργότερο μέχρι τις έξι το απόγευμα. Ένα παιδί μπορεί να έχει δύο γονείς, αλλά χρειαζόμαστε μόνο έναν από τους δύο για να έρθει να παραλάβει το παιδί του. Κάθε παιδί δίνεται μόνο στον γονέα του και σε κανέναν άλλο».

Δραστηριότητα: Μοντελοποίηση ενισχυτικής διδασκαλίας

Διαβάστε το παρακάτω σενάριο, που συσχετίζει τις οντότητες Καθηγητής και Μαθητής και μοντελοποιήστε με γραφικό τρόπο τις σχέσεις αυτές. Δώστε δηλαδή κατάλληλα ονόματα και στα δύο άκρα της γραμμής σχέσης και σχεδιάστε την γραμμή σχέσης κατά αυτόν τον τρόπο που υποδηλώνει τυχόν προαιρετικές ή προαιρετικές σχέσεις καθώς και την πληθάρημο της σχέσης.

«Μερικοί μαθητές ζητούν βοήθεια σε ορισμένα μαθήματα, όπως είναι τα μαθηματικά. Μπορούμε να ορίσουμε ένα καθηγητή να βοηθήσει τον μαθητή μετά το πέρας των μαθημάτων. Το σχολείο αναθέτει σε μερικούς καθηγητές την ενισχυτική αυτή διδασκαλία. Εάν περισσότεροι μαθητές χρειάζονται υποστήριξη σε ένα μάθημα, τότε μπορούμε να τους αντιστοιχίσουμε στον ίδιο καθηγητή. Αν ένας μαθητής χρειάζεται βοήθεια σε περισσότερα από ένα μαθήματα, τότε μάλλον θα ανατεθεί σε διάφορους καθηγητές».

3.3.4 Επίλυση Συσχετίσεων Πολλά προς Πολλά

Κατά την διαδικασία της μοντελοποίησης είναι δυνατό να ορίσετε αρκετές σχέσεις μεταξύ των οντοτήτων του τύπου «πολλά προς πολλά» (M:M). Οι περισσότερες από αυτές τις σχέσεις «πολλά προς πολλά» «κρύβουν» μέσα τους οντότητες τις οποίες θα χρειαστούμε στη συνέχεια για να τις προσδώσουμε συγκεκριμένα γνωρίσματα. Γνωρίσματα τα οποία απαραίτητα για μία πληρέστερη εννοιολογική απόδοση των αναγκών μίας επιχείρησης ή ενός οργανισμού.

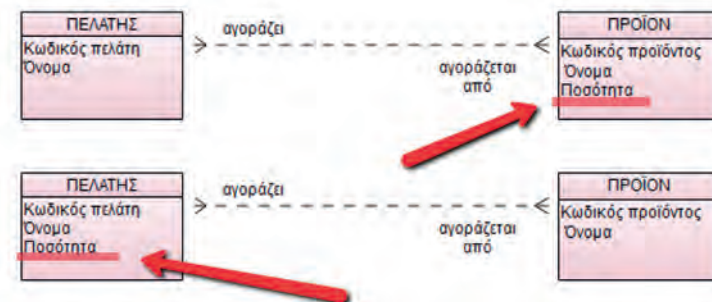
Ένα τυπικό παράδειγμα, αποτελεί αυτό της σχέσης μεταξύ ΠΕΛΑΤΗ και ΠΡΟΪΟΝ, όπως παρουσιάζεται στο παρακάτω διάγραμμα οντοτήτων-συσχετίσεων. Μία εταιρεία λιανικού εμπορίου πουλάει διάφορα προϊόντα σε πελάτες της. Ας υποθέσουμε ότι γίνονται δεκτοί στο σύστημα και μελλοντικοί πελάτες. Το ER διάγραμμα διαβάζεται ως εξής:

- Ένας πελάτης μπορεί να αγοράσει ένα ή περισσότερα προϊόντα.
- Ένα προϊόν μπορεί να αγοραστεί από έναν ή περισσότερους πελάτες.



Σχήμα 3.3.14: Οι σχέσεις «πολλά προς πολλά» κάτι κρύβουν

Για παράδειγμα, ο πελάτης Νίκος Χατζηπέτρου αγοράζει δύο μπλούζες. Το όνομα «Νίκος Χατζηπέτρου» είναι το όνομα του πελάτη και «μπλούζα» είναι το όνομα του προϊόντος. Τίθεται το εξής ερώτημα: Η ποσότητα 2 ως πληροφορία σε ποια από τις δύο οντότητες θα «αποθηκευτεί»; Στον ΠΕΛΑΤΗ ή στο ΠΡΟΪΟΝ;



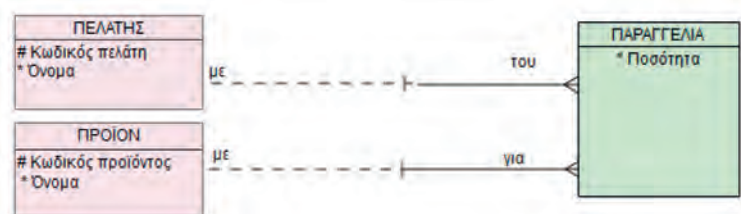
Σχήμα 3.3.15: Η ποσότητα είναι γνώρισμα της

Είναι ξεκάθαρο ότι η πληροφορία «Ποσότητα» δεν αποτελεί γνώρισμα ούτε της οντότητας ΠΕΛΑΤΗ αλλά ούτε και της οντότητας ΠΡΟΪΟΝ. Περισσότερο φαίνεται να είναι η πληροφορία «Ποσότητα» γνώρισμα της σχέσης που συνδέει τον ΠΕΛΑΤΗ με το ΠΡΟΪΟΝ.



Σχήμα 3.3.16: Η ποσότητα είναι γνώρισμα της σχέσης

Οι συσχετίσεις όμως δεν έχουν και δεν μπορούν να έχουν γνωρίσματα. Προφανώς λείπει μία οντότητα της οποίας ένα γνώρισμα είναι η «Ποσότητα». Εισάγεται στο ER διάγραμμα η οντότητα «ΠΑΡΑΓΓΕΛΙΑ», την οποία ονομάζουμε οντότητα «τομής» (intersection entity).



Σχήμα 3.3.17: Επίλυση συσχέτισης «πολλά προς πολλά». Εισάγεται μια καινούρια οντότητα η «ΠΑΡΑΓΓΕΛΙΑ»

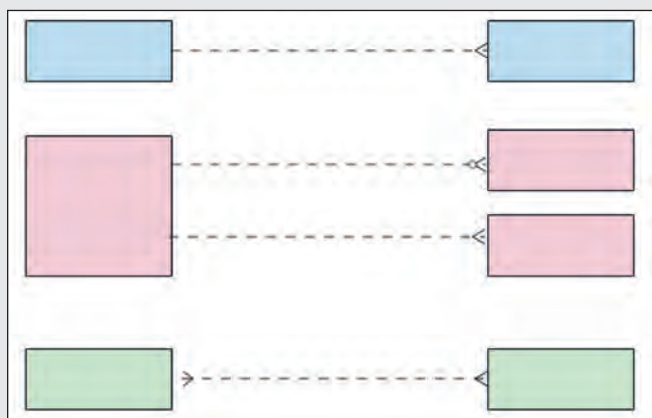
Με την δημιουργία μία τρίτης βοηθητικής οντότητας, όπως είναι η οντότητα «ΠΑΡΑΓΓΕΛΙΑ» στο προηγούμενο παράδειγμα, επιλύεται το πρόβλημα των συσχετίσεων «πολλά προς πολλά». Η συσχέτιση «πολλά προς πολλά» παραχωρεί την θέση της σε δύο συσχετίσεις «ένα προς πολλά». Το μοναδικό αναγνωριστικό ή κλειδί της τρίτης οντότητας προκύπτει από τον συνδυασμό των κλειδιών των δύο αρχικών οντοτήτων. Τώρα, στο ER διάγραμμα οι συσχετίσεις που ενώνουν τις δύο αρχικές οντότητες με την τρίτη βοηθητική οντότητα είναι «ένα προς πολλά».



Ερωτήσεις / Δραστηριότητες

Δραστηριότητα: Βρείτε ένα πλαίσιο

Με βάση τα ακόλουθα ER διαγράμματα βρείτε ένα πλαίσιο το οποίο να ικανοποιεί το αντίστοιχο μοντέλο.



Σχήμα 3.3.18: ER διαγράμματα

Δραστηριότητα: Ονομάστε την οντότητα τομής

Επιλύστε τις σχέσεις «πολλά προς πολλά» (M:M) μεταξύ των οντοτήτων, όπως φαίνεται στο παρακάτω σχήμα. Βρείτε ένα αποδεκτό όνομα για την οντότητα «τομής» (intersection entity). Για κάθε οντότητα τομής προσθέσετε ένα επιπλέον γνώρισμα, το οποίο έχει νόημα για την αντίστοιχη σχέση. Ποιο είναι το κλειδί για την οντότητα τομής;



Σχήμα 3.3.19: Επίλυση συσχέτισης «πολλά προς πολλά»

3.4 Μοναδικά Κλειδιά και Κανονικοποίηση (Normalization)



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να προσδιορίζει τα πρωτεύοντα, τα δευτερεύοντα και τα σύνθετα κλειδιά σε μια βάση δεδομένων
- να αναφέρει τους λόγους για τους οποίους είναι χρήσιμη η κανονικοποίηση
- να εφαρμόζει τεχνικές κανονικοποίησης (Πρώτη Κανονική Μορφή)

Ένα βασικό θέμα στον σχεδιασμό μιας βάσης δεδομένων είναι ο προσδιορισμός των κλειδιών με τα οποία γίνεται η αναζήτηση πληροφοριών στη βάση δεδομένων. Τα κλειδιά μπορεί να είναι πρωτεύοντα (μοναδικά για κάθε εγγραφή), δευτερεύοντα (για παράδειγμα το επίθετο ενός υπαλλήλου, μαθητή) ή σύνθετα (ένωση δύο ή και περισσότερων χαρακτηριστικών). Στην παρούσα ενότητα θα αναφερθούμε στον προσδιορισμό των κλειδιών μέσω παραδειγμάτων.

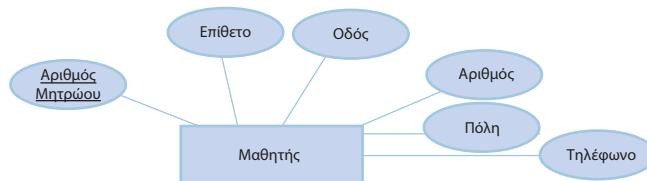
Ένα δεύτερο θέμα που θα μας απασχολήσει στην ενότητα αυτή, είναι η **κανονικοποίηση** (normalization), που αποτελεί ένα πολύ σημαντικό βήμα στον σχεδιασμό μίας βάσης δεδομένων. Οι πίνακες που προκύπτουν μετά τη μετατροπή του μοντέλου ΟΣ σε σχεσιακό, όπως αναφέρθηκε στην ενότητα 3.2, περιέχουν ως συνήθως πληροφορίες που επαναλαμβάνονται.. Η επαναλαμβανόμενη πληροφορία οδηγούν σε πολλαπλές γραμμές στον σχεδιασμό του αντίστοιχου πίνακα, γεγονός που οδηγεί σε σημαντικά προβλήματα. Η κανονικοποίηση σχέσεων αφορά στην απλοποίηση των σχέσεων με στόχο την εξάλειψη της επαναληπτικότητας των δεδομένων.

3.4.1 Πρωτεύον, Σύνθετο και Δευτερεύον κλειδί

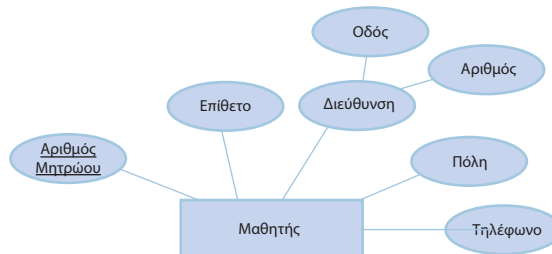
Κάθε οντότητα έχει ένα σύνολο χαρακτηριστικών/ ιδιοτήτων. Αναλυτικά, έχουμε τους εξής τύπους χαρακτηριστικών:

Απλό ή Σύνθετο.

Ένα απλό χαρακτηριστικό δεν μπορεί να διαχωριστεί σε μικρότερα τμήματα, ενώ τα σύνθετα χαρακτηριστικά μπορούν να χωριστούν σε μικρότερα τμήματα σε άλλα απλούστερα χαρακτηριστικά. Η χρήση συνθέτων χαρακτηριστικών είναι καλύτερα να γίνεται όταν επιθυμούμε σε κάποιες περιπτώσεις να αναφερόμαστε σε ολόκληρο το χαρακτηριστικό, ενώ σε κάποιες άλλες περιπτώσεις επιθυμούμε να αναφερόμαστε στα επιμέρους τμήματά του. Για παράδειγμα στο σχήμα 3.4.1, θα μπορούσαμε να αντικαταστήσουμε τα χαρακτηριστικά οδός και αριθμός του συνόλου οντοτήτων Μαθητής με το σύνθετο χαρακτηριστικό διεύθυνση όπως παρουσιάζεται στο σχήμα 3.4.2. Τα σύνθετα χαρακτηριστικά μας βοηθούν να συγκεντρώσουμε συσχετιζόμενα χαρακτηριστικά σε ομάδες, διευκολύνοντας το σχεδιασμό της ΒΔ.



Σχήμα 3.4.1 Οντότητα Μαθητής



Σχήμα 3.4.2 Οντότητα Μαθητής

Απλής Τιμής ή Πολλαπλής Τιμής.

Τα χαρακτηριστικά απλής τιμής έχουν μία μόνο τιμή. Για παράδειγμα, το σύνολο Μαθητής έχει το χαρακτηριστικό επίθετο το οποίο είναι ένα για κάθε οντότητα. Υπάρχουν όμως χαρακτηριστικά με δυο ή περισσότερες τιμές. Αυτά ονομάζονται χαρακτηριστικά πολλαπλής τιμής. Για παράδειγμα, θεωρήστε το σύνολο Μαθητής το οποίο έχει το χαρακτηριστικό τηλέφωνο. Το χαρακτηριστικό τηλέφωνο μπορεί να έχει δύο ή και περισσότερες τιμές εφόσον ο μαθητής μπορεί να έχει σταθερό και κινητό τηλέφωνο. Στα χαρακτηριστικά πολλαπλής τιμής μπορούμε να ορίσουμε κατώτερο και ανώτερο όριο τιμών. Για παράδειγμα, μπορούμε να περιορίσουμε τον αριθμό των τηλεφωνικών αριθμών που επιθυμούμε να καταγράψουμε σε δύο μόνο (π.χ. ένα αριθμό τηλεφώνου σπιτιού και ένα για κινητό). Τα χαρακτηριστικά αυτά αναπαριστώνται με διπλή έλλειψη σε ένα διάγραμμα οντοτήτων συσχετίσεων.

Κενό.

Μια **κενή** τιμή χρησιμοποιείται όταν κάποια οντότητα δεν έχει τιμή για κάποιο χαρακτηριστικό. Για παράδειγμα, εάν κάποιος μαθητής δηλώσει ότι δεν έχει τηλέφωνο, τότε η τιμή του χαρακτηριστικού Τηλέφωνο για τον συγκεκριμένο μαθητή θα είναι Κενό.

Ένα χαρακτηριστικό/ιδιότητα ενός πίνακα ονομάζεται **πρωτεύον κλειδί** εάν μπορεί να διαχωρίζει τις διαφορετικές γραμμές του πίνακα. Για παράδειγμα, ο αριθμός αστυνομικής ταυτότητας ή ο αριθμός φορολογικού μητρώου μπορούν να θεωρηθούν ως πρωτεύοντα κλειδιά για τον πίνακα Υπάλληλος εφόσον είναι μοναδικά για κάθε υπάλληλο. Αντίθετα, το επίθετο δε θεωρείται πρωτεύον κλειδί, διότι μπορεί να υπάρχουν πολλοί υπάλληλοι που να έχουν το ίδιο επίθετο (για παράδειγμα το επίθετο Παπαδόπουλος είναι κοινό και μπορεί να το έχουν πολλοί υπάλληλοι μιας εταιρείας ή ενός οργανισμού). Ένας πίνακας μπορεί να έχει περισσότερα από ένα πρωτεύοντα κλειδιά (για παράδειγμα ο αριθμός αστυνομικής ταυτότητας και ο αριθμός φορολογικού μητρώου). Ωστόσο, ένα μόνο επιλέγεται να χρησιμοποιηθεί. Το κλειδί που επιλέγεται καλείται **πρωτεύον κλειδί (primary key)** ενώ τα υπόλοιπα καλούνται **δευτερεύοντα (secondary)**. Ως δευτερεύον κλειδί στον πίνακα υπάλληλος μπορεί να επιλεγεί το επίθετο έτσι ώστε να διευκολύνεται ο χρήστης στην αναζήτηση κάποιου υπαλλήλου με τη χρήση του επιθέτου. Σε πολλές περιπτώσεις το κλειδί ενός πίνακα μπορεί να είναι **σύνθετο (composite)** δηλαδή να αποτελείται από δύο ή και περισσότερα χαρακτηριστικά. Για παράδειγμα, στον πίνακα 3.4.3 παρουσιάζεται ένα σύνθετο κλειδί αποτελούμενο από τα χαρακτηριστικά, Επίθετο, Όνομα και Όνομα Πατρός. Με αυτόν τον τρόπο διαχωρίζονται οι γραμμές του πίνακα και εύκολα ξεχωρίζει η εγγραφή Παπαδόπουλος Ιωάννης του Ιωάννη από την εγγραφή Παπαδόπουλος Ιωάννης του Χρήστου.

Επίθετο	Όνομα	Όνομα Πατρός	Διεύθυνση	Πόλη
Παπαδόπουλος	Ιωάννης	Δημήτριος	Βενιζέλου 4	Θεσσαλονίκη
Παπαδόπουλος	Ιωάννης	Χρήστος	Τσιμισκή 19	Θεσσαλονίκη
Χαρισίδης	Νικόλαος	Μενέλαος	Εγνατία 151	Θεσσαλονίκη
Κερκινάκης	Δημήτριος	Γεώργιος		

Σχήμα 3.4.3 Σύνθετο κλειδί

3.4.2 Κανονικοποίηση και Πρώτη Κανονική Μορφή

Στην ενότητα 3.2 αναφερθήκαμε στην επαναληπτικότητα των δεδομένων και ότι αυτή δημιουργεί προβλήματα στους πίνακες που προκύπτουν από το σχεσιακό μοντέλο. Αναλογιστήκαμε όμως τους λόγους που καθιστούν ανεπιθύμητη την επαναληπτικότητα;

Οι κύριοι λόγοι είναι:

Πρόβλημα Ενημέρωσης Δεδομένων.

Αν χρειαστεί να γίνει αλλαγή των δεδομένων κατά τη χρήση της ΒΔ, αυτά πρέπει να αλλάξουν όπου εμφανίζονται. Για παράδειγμα, η αλλαγή διεύθυνσης ενός μαθητή οδηγεί σε ενημέρωση δύο γραμμών του πίνακα Μαθητής (Σχήμα 3.4.4). Οι ενημερώσεις, δημιουργούν καθυστερήσεις και αν για κάποιους λόγους δεν ενημερωθούν όλες οι γραμμές του πίνακα οδηγούμαστε σε ασυνέπεια δεδομένων (άλλες γραμμές να έχουν τη νέα διεύθυνση και άλλες την παλιά).

Πρόβλημα Διαγραφής Δεδομένων.

Αν για κάποιους λόγους, ένας μαθητής καταργήσει όλους τους τηλεφωνικούς του αριθμούς, δηλώνοντας ότι δεν έχει τηλέφωνο, τι μπορούμε να κάνουμε;

Οι εναλλακτικές λύσεις που έχουμε αλλά καμία δεν μπορεί να εφαρμοστεί είναι οι εξής:

- 1) να θέσουμε κενό (NULL) σε όλες τις τιμές της στήλης τηλέφωνο, κάτι το οποίο δεν μπορεί να γίνει διότι η στήλη μπορεί να συμμετέχει στο πρωτεύον κλειδί του πίνακα, και
- 2) να διαγράψουμε τις γραμμές του πίνακα, που επίσης δεν μπορεί να γίνει διότι θα χάσουμε τις υπόλοιπες πληροφορίες για το συγκεκριμένο μαθητή.

Πρόβλημα Εισαγωγής Δεδομένων.

Κάθε φορά που εισάγουμε στη βάση δεδομένων έναν νέο μαθητή, πρέπει να εισάγουμε και κάποιον αριθμό τηλεφώνου, κάτι το οποίο δεν είναι απαραίτητο να γίνει στην αρχή.

ΑΜ Μαθητή	Επίθετο	Διεύθυνση	Πόλη	Τηλέφωνο
14355	Παπαδόπουλος	Βενιζέλου 4	Θεσσαλονίκη	23105555555
14355	Παπαδόπουλος	Βενιζέλου 4	Θεσσαλονίκη	69999999999
14582	Χαρισίδης	Τσιμισκή 1	Θεσσαλονίκη	23106666666
14695	Κερκινάκης	Εγνατία 151	Θεσσαλονίκη	23107777777

Σχήμα 3.4.4 Πίνακας με επαναλαμβανόμενες πληροφορίες

Η λύση που υπάρχει για την επίλυση των παραπάνω προβλημάτων είναι η κατασκευή δύο διαφορετικών πινάκων. Στο παραπάνω σχήμα 3.4.4 παρουσιάζεται ο πίνακας Μαθητής με επαναλαμβανόμενες πληροφορίες λόγω της ύπαρξης του χαρακτηριστικού πολλαπλής τιμής Τηλέφωνο. Μετά την κατασκευή δύο πινάκων, η νέα μορφή της βάσης δεδομένων που προκύπτει παρουσιάζεται στο Σχήμα 3.4.5.

ΑΜ Μαθητή	Επίθετο	Διεύθυνση	Πόλη
14355	Παπαδόπουλος	Βενιζέλου 4	Θεσσαλονίκη
14582	Χαρισίδης	Τσιμισκή 1	Θεσσαλονίκη
14695	Κερκινάκης	Εγνατία 151	Θεσσαλονίκη

ΑΜ Μαθητή	Τηλέφωνο
14355	23105555555
14355	69999999999
14582	23106666666
14695	23107777777

Σχήμα 3.4.5 Κατασκευή δύο πινάκων για την αποφυγή της επαναληπτικότητας των δεδομένων

Βασικός στόχος της κανονικοποίησης είναι ο προσδιορισμός ενός συνόλου πινάκων με στόχο την ελαχιστοποίηση της επαναληπτικότητας δεδομένων. Με τη μέθοδο της κανονικοποίησης πραγματοποιούνται διαδοχικές διασπάσεις των πινάκων σε πιο απλές μορφές. Η διάσπαση ακολουθεί κάποιους κανόνες. Από τον αρχικό πίνακα απομακρύνονται κάποια χαρακτηριστικά και τοποθετούνται σε νέους πίνακες ώστε να είναι δυνατή η ανασύνδεση των αρχικών πληροφοριών μέσω των νέων σχέσεων.

Έχουν προταθεί πέντε κανονικές μορφές που όμως δεν θα αναλυθούν στην παρούσα ενότητα εκτός από την πρώτη κανονική μορφή.

Πρώτη Κανονική Μορφή (1NF)

Ένας πίνακας βρίσκεται στην πρώτη κανονική μορφή όταν οι τιμές που λαμβάνει κάθε στήλη του πίνακα είναι ατομικές. Επομένως, η πρώτη κανονική μορφή απαγορεύει την ύπαρξη στηλών οι οποίες περιέχουν χαρακτηριστικά πολλαπλής τιμής. Υπάρχουν δύο τρόποι για να μετασχηματίσουμε έναν πίνακα που περιέχει πολλαπλές τιμές στην πρώτη κανονική μορφή.

Ο ένας τρόπος είναι να επαναλάβουμε τα δεδομένα πολλές φορές, έτσι ώστε τελικά ο πίνακας που θα προκύψει να περιέχει μόνο ατομικές τιμές στις στήλες. Ο δεύτερος τρόπος είναι να χρησιμοποιήσουμε ξεχωριστό πίνακα για τα χαρακτηριστικά πολλαπλών τιμών. Και οι δύο τρόποι θεωρούνται σωστοί, όμως ο δεύτερος μειώνει σημαντικά την επαναληπτικότητα των δεδομένων.

Παράδειγμα πρώτης κανονικής μορφής

Θεωρείστε ότι σχεδιάζεται το Μαθητολόγιο του σχολείου σας. Κάθε μαθητής παρακολουθεί μαθήματα ανάλογα με την τάξη. Για κάθε μαθητή θα υπάρχουν τα στοιχεία του καθώς και τα μαθήματα τα οποία παρακολουθεί.

ΑΜ Μαθητή	Επίθετο	Όνομα	Όνομα Πατρός	Διεύθυνση	Μαθήματα
14355	Παπαδόπουλος	Ιωάννης	Δημήτριος	Βενιζέλου 4	ΣΔΒΔ
14666	Παπαδόπουλος	Ιωάννης	Χρήστος	Εθν. Αμύνης 8	Μαθηματικά
14582	Χαρισίδης	Νικόλαος	Μενέλαος	Τσιμισκή 19	Φυσική Αγωγή
14695	Κερκινάκης	Δημήτριος	Γεώργιος	Εγνατία 151	Φυσική

Παρατηρούμε ότι το χαρακτηριστικό Μαθήματα είναι πολλαπλής τιμής και θα αναγκαστούμε να διασπάσουμε τον πίνακα ώστε να κανονικοποιηθεί.

ΑΜ Μαθητή	Επίθετο	Όνομα	Όνομα Πατρός	Διεύθυνση
14355	Παπαδόπουλος	Ιωάννης	Δημήτριος	Βενιζέλου 4
14666	Παπαδόπουλος	Ιωάννης	Χρήστος	Εθν. Αμύνης 8
14582	Χαρισίδης	Νικόλαος	Μενέλαος	Τσιμισκή 19
14695	Κερκινάκης	Δημήτριος	Γεώργιος	Εγνατία 151

ΑΜ Μαθητή	Μαθήματα
14355	ΣΔΒΔ
14355	Μαθηματικά
14355	Φυσική Αγωγή κ.α
14355	ΣΔΒΔ
14666	Νέα Ελληνικά
14582	Φυσική
14695	Μαθηματικά



Ερωτήσεις / Δραστηριότητες

1. Πότε ένα χαρακτηριστικό ονομάζεται σύνθετο και πότε πολλαπλής τιμής; Δώστε από ένα παράδειγμα για την κάθε περίπτωση.
2. Ποια είναι χρησιμότητα του κλειδιού σε ένα πίνακα;
3. Ποια η χρησιμότητα του δευτερεύοντος κλειδιού; Δώστε ένα παράδειγμα.
4. Εξηγείστε τι εννοούμε με τον όρο επανάληψη πληροφοριών.
5. Περιγράψτε τις ανωμαλίες ενημέρωσης που προκύπτουν από την επαναληπτικότητα των δεδομένων.
6. Εξηγείστε τι εννοούμε με τον όρο κανονικοποίηση.

Δραστηριότητα 1: Μαθητολόγιο

Πρόκειται να σχεδιάσετε το Μαθητολόγιο του σχολείου σας. Σας δίνονται οι πίνακες του παραδείγματος 3.4.1. Μπορείτε να τους συμπληρώσετε ώστε να περιλαμβάνονται και οι βαθμοί των μαθητών σε κάθε μάθημα;

Σημείωση: Μην ξεχάσετε να εφαρμόσετε κανονικοποίηση

Δραστηριότητα 2: Μαθητολόγιο

Προσδιορίστε τα κλειδιά των πινάκων που δημιουργήσατε στην Δραστηριότητα 1 (πρωτεύον, δευτερεύον ή σύνθετο);

3.5 Μετασχηματισμός από το Εννοιολογικό Μοντέλο στο Σχεσιακό Μοντέλο

Το επόμενο βήμα στον σχεδιασμό των βάσεων δεδομένων, πριν την φυσική υλοποίηση, αποτελεί ο μετασχηματισμός του μοντέλου οντοτήτων συσχετίσεων στο σχεσιακό μοντέλο. Αυτό σημαίνει ότι τα «αντικείμενα» του ER διαγράμματος που γνωρίσαμε στις προηγούμενες υποενότητες θα μετατραπούν στα αντίστοιχα «αντικείμενα» του σχεσιακού μοντέλου.

Αυτή η ενότητα περιγράφει μερικές βασικές αρχές των σχεσιακών βάσεων δεδομένων (relational databases) και παρουσιάζει διάφορες τεχνικές που μπορούμε να χρησιμοποιήσουμε για να μετατρέψουμε το μοντέλο οντοτήτων συσχετίσεων (Entity Relationship model) στο σχεσιακό μοντέλο.



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να αναγνωρίζει σε ένα πίνακα την σημασία της γραμμής, της στήλης, του πρωτεύοντος και του ξένου κλειδιού
- να διακρίνει το εννοιολογικό από το σχεσιακό μοντέλο
- να αντιλαμβάνεται την δομή του σχεσιακού μοντέλου
- να μετατρέπει μία οντότητα σε διαγραμματικό πίνακα
- να μετατρέπει τις σχέσεις μεταξύ των οντοτήτων σε στήλες που έχουν τον ρόλο του ξένου κλειδιού

3.5.1 Εισαγωγή στις έννοιες των Σχεσιακών Βάσεων Δεδομένων

Το επόμενο βήμα στον σχεδιασμό των βάσεων δεδομένων αποτελεί ο μετασχηματισμός του μοντέλου οντοτήτων συσχετίσεων στο σχεσιακό μοντέλο. Αυτό σημαίνει ότι οι οντότητες, τα γνωρίσματα, οι συσχετίσεις και τα μοναδικά αναγνωριστικά του ER διαγράμματος που γνωρίσαμε στις προηγούμενες υποενότητες θα μετατραπούν στα αντίστοιχα «αντικείμενα» του σχεσιακού μοντέλου.

Θεωρείστε τον μετασχηματισμό αυτόν παρόμοιο με αυτόν που ακολουθεί ο σχεδιαστής ρούχων όταν με βάση το σχέδιο του στο χαρτί δημιουργεί το αντίστοιχο ρούχο από ύφασμα. Η μετατροπή αυτή απαιτεί γνώσεις ραπτικής από την πλευρά του σχεδιαστή. Έτσι και εμείς στις επόμενες υποενότητες θα εφοδιαστούμε με γνώσεις σχετικές με την δομή του σχεσιακού μοντέλου. Το σχεσιακό μοντέλο αποτελεί ένα βήμα πριν την υλοποίηση.

Ας μελετήσουμε τώρα την δομή του σχεσιακού μοντέλου. Ένα σχεσιακό μοντέλο μπορεί να ειδωθεί ως ένα σύνολο από δισδιάστατους πίνακες. Στους δισδιάστατους αυτούς πίνακες οργανώνονται και αποθηκεύονται τα δεδομένα. Στο παρακάτω σχήμα παρουσιάζεται ο πίνακας ΕΡΓΑΖΟΜΕΝΟΙ, όπου αποθηκεύονται πληροφορίες σχετικά με τους εργαζόμενους.

Πίνακας: ΕΡΓΑΖΟΜΕΝΟΙ

Κωδικός_Εργαζομένου	Όνομα	Επίθετο	Κωδικός_Τμήματος
100	Μαρία	Χατζηπέτρου	10
310	Πέτρος	Θεοδώρου	15
210	Κώστας	Παπαδημητρίου	10
405	Αλίκη	Ιωαννίδου	22

Σχήμα 3.5.1: Πίνακας ΕΡΓΑΖΟΜΕΝΟΙ

Κάθε πίνακας περιέχει γραμμές και στήλες. Κάθε γραμμή περιγράφει και έναν εργαζόμενο ή ένα στιγμιότυπο της οντότητας ΕΡΓΑΖΟΜΕΝΟΣ. Οι γραμμές του πίνακα είναι τέσσερις. Όσοι και οι εργαζόμενοι. Κάθε στήλη χρησιμεύει για να αποθηκεύει συγκεκριμένο τύπο τιμών όπως είναι ο κωδικός του εργαζομένου, το ονοματεπώνυμό του και ο κωδικός του τμήματος στο οποίο εργάζεται.

Το πρωτεύον κλειδί (primary key (PK)) είναι μία στήλη ή ένας συνδυασμός από στήλες που προσδιορίζει με μοναδικό τρόπο κάθε γραμμή του πίνακα. Στο παρακάτω σχήμα, το πρωτεύον κλειδί του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ είναι ο κωδικός του εργαζομένου δηλαδή η στήλη Κωδικός_Εργαζομένου. Αυτό σημαίνει ότι δεν μπορούμε να έχουμε δύο εργαζόμενους με τον ίδιο κωδικό. Κάθε εργαζόμενος προσδιορίζεται από ένα μη μηδενικό κωδικό ο οποίος είναι μοναδικός για τον εργαζόμενο αυτόν.

Πίνακας: ΕΡΓΑΖΟΜΕΝΟΙ

Κωδικός_Εργαζομένου	Όνομα	Επίθετο	Κωδικός_Τμήματος
100	Μαρία	Χατζηπέτρου	10
310	Πέτρος	Θεοδώρου	15
210	Κώστας	Παπαδημητρίου	10
405	Αλίκη	Ιωαννίδου	22

Σχήμα 3.5.2: Το πρωτεύον κλειδί του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ είναι η στήλη Κωδικός_Εργαζομένου

Αντίθετα στον πίνακα ΛΟΓΑΡΙΑΣΜΟΙ το πρωτεύον κλειδί είναι ο συνδυασμός του κωδικού της τράπεζας και του κωδικού του λογαριασμού. Στην περίπτωση αυτή, δεν αρκεί μόνο μία στήλη για να προσδιοριστεί μοναδικά κάθε λογαριασμός. Διότι είναι δυνατό σε δύο διαφορετικές τράπεζες όπως συμβαίνει στην πρώτη και πέμπτη γραμμή του πίνακα, ο κωδικός λογαριασμού να είναι ο ίδιος. Άρα, χρειαζόμαστε και μία δεύτερη στήλη για να προσδιορίσουμε το πρωτεύον κλειδί. Αυτή η δεύτερη στήλη είναι ο κωδικός της τράπεζας.

Πίνακας: ΛΟΓΑΡΙΑΣΜΟΙ

Κωδικός_Τράπεζας	Κωδικός_Λογαριασμού	Υπόλοιπο	Ημερομηνία_Έναρξης
104	75760	12,0050.00	21-OCT-89
104	77956	100.10	
105	89570	55,775.00	15-JAN-85
103	55890	15,001.85	10-MAR-91
105	75760	5.00	22-SEP-03

Πρωτεύον Κλειδί: Συνδυασμός από δύο στήλες

Σχήμα 3.5.3: Το πρωτεύον κλειδί του πίνακα ΛΟΓΑΡΙΑΣΜΟΙ είναι ο συνδυασμός των στηλών Κωδικός_Τράπεζας και Κωδικός_Λογαριασμού

Ένα **ξένο κλειδί (foreign key (FK))** είναι μία στήλη ή ένας συνδυασμός από στήλες ενός πίνακα που μας επιτρέπει να «συνδεθούμε» με μία γραμμή ενός άλλου πίνακα.

Για παράδειγμα στο παρακάτω σχήμα από τον πίνακα ΕΡΓΑΖΟΜΕΝΟΙ (EMPLOYEES) γνωρίζουμε ότι Jennifer Whalen εργάζεται στο τμήμα με κωδικό 10. Αν θέλουμε να μάθουμε περισσότερα για το τμήμα στο οποίο εργάζεται η Jennifer Whalen, θα αναζητήσουμε στον πίνακα ΤΜΗΜΑΤΑ (DEPARTMENTS) εκείνη την γραμμή του πίνακα που έχει DEPARTMENT_ID το 10. Το τμήμα με κωδικό 10 έχει όνομα Administration. Άρα, η Jennifer Whalen εργάζεται στο τμήμα Administration.

Πρέπει να σημειώσουμε ότι το DEPARTMENT_ID ορίζεται ως το πρωτεύον κλειδί για τον πίνακα ΤΜΗΜΑΤΑ (DEPARTMENTS) ενώ για τον πίνακα ΕΡΓΑΖΟΜΕΝΟΙ (EMPLOYEES) ο κωδικός τμήματος DEPARTMENT_ID ορίζεται ως το ξένο κλειδί. Κάθε τιμή της στήλης DEPARTMENT_ID του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ (EMPLOYEES) πρέπει να «ταιριιάζει» με μία τιμή από την στήλη DEPARTMENT_ID του πίνακα ΤΜΗΜΑΤΑ (DEPARTMENTS).

Οι δύο αυτές λοιπόν συσχετιζόμενες μεταξύ τους πληροφορίες «Jennifer Whalen» και

«Administration» βρίσκονται σε διαφορετικούς πίνακες και η εύρεση της συσχέτισής τους γίνεται μέσω των ξένων και πρωτευόντων κλειδιών των αντίστοιχων πινάκων.

Πίνακας: ΕΡΓΑΖΟΜΕΝΟΙ

Κωδικός_Εργαζομένου	Όνομα	Επίθετο	Κωδικός_Τμήματος
100	Μαρία	Χατζηπέτρου	90
101	Πέτρος	Θεοδώρου	90
102	Κώστας	Παπαδημητρίου	90
200	Αλίκη	Ιωαννίδου	10
205	Μιράντα	Κερ	110

Πίνακας: ΤΜΗΜΑΤΑ

Κωδικός_Τμήματος	Όνομα_Τμήματος
10	Διοίκηση
20	Εμπορικό Τμήμα
50	Πωλήσεις

Πρωτεύον Κλειδί

ξένο κλειδί

συσχετίζεται με

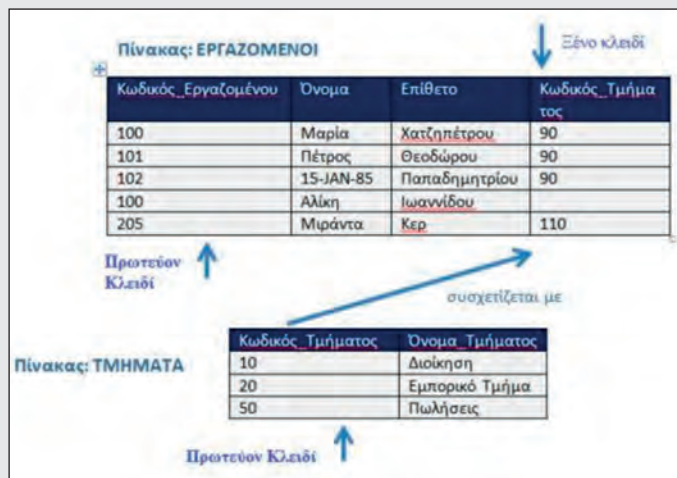
Σχήμα 3.5.4: Το ξένο κλειδί ενός πίνακα «συνδέεται» με το πρωτεύον κλειδί ενός άλλου πίνακα



Ερωτήσεις / Δραστηριότητες

Δραστηριότητα: Παραβίαση κανόνων

Δίνεται το παρακάτω σχεσιακό μοντέλο. Παρατηρείτε λάθη στο σχεσιακό αυτό μοντέλο;



Σχήμα 3.5.5: Σχεσιακό μοντέλο

Δραστηριότητα: Μελέτη σχεσιακού μοντέλου

Δίνεται το παρακάτω σχεσιακό μοντέλο. Προσπαθήστε να απαντήσετε στις παρακάτω ερωτήσεις:

- Ο εργαζόμενος Πέτρος Θεοδώρου σε ποιο τμήμα δουλεύει;
- Στο εμπορικό τμήμα πόσοι εργαζόμενοι απασχολούνται;
- Ο εργαζόμενος με κωδικό 100 σε ποιο τμήμα δουλεύει;



Σχήμα 3.5.6: Σχεσιακό μοντέλο

3.5.2 Διαδικασία Μετασχηματισμού από το Εννοιολογικό Μοντέλο στο Σχεσιακό Μοντέλο

Στην υποενότητα αυτή θα μελετήσουμε τους κανόνες που διέπουν τον μετασχηματισμό του μοντέλου οντοτήτων συσχετίσεων στο σχεσιακό μοντέλο. Η υλοποίηση του σχεσιακού μοντέλου είναι μία σχεσιακή βάση δεδομένων, όπως φαίνεται στο παρακάτω σχήμα, με τον (σχεσιακό) πίνακα ΜΑΘΗΤΕΣ.

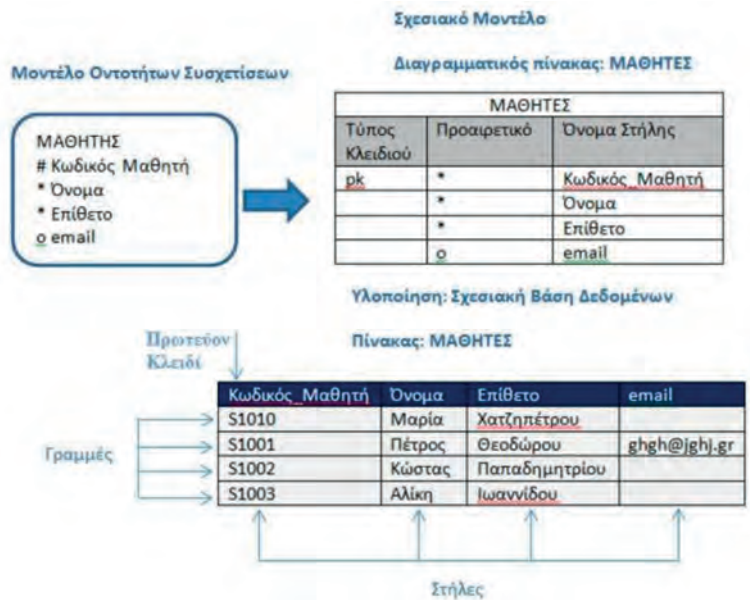
Ο (σχεσιακός) πίνακας ΜΑΘΗΤΕΣ είναι ένας δισδιάστατος πίνακας. Αποτελείται από γραμμές και στήλες. Έχει τέσσερις στήλες Κωδικός_Μαθητή, Όνομα, Επίθετο και email. Κάθε γραμμή του πίνακα αυτού περιγράφει ένα στιγμιότυπο της οντότητας ΜΑΘΗΤΗΣ. Για παράδειγμα, υπάρχει καταχωρημένος στον πίνακα ο μαθητής με όνομα «Πέτρος», επίθετο «Θεοδώρου», email

«ghgh@jghj.gr» και κωδικό μαθητή «S1001». Η στήλη Κωδικός_Μαθητή αποτελεί το πρωτεύον κλειδί του πίνακα. Κάθε μαθητής έχει ένα μοναδικό κωδικό. Δεν μπορούμε, δηλαδή, να συναντήσουμε δύο μαθητές με τον ίδιο κωδικό.

Όλες οι παραπάνω αυτές πληροφορίες σχετικά με τον (σχεσιακό) πίνακα ΜΑΘΗΤΕΣ, όπως αυτές καθορίζονται από το αντίστοιχο μοντέλο οντοτήτων συσχετίσεων, αποθηκεύονται με έναν συγκεκριμένο τρόπο στον διαγραμματικό πίνακα ΜΑΘΗΤΕΣ. Ο διαγραμματικός αυτός πίνακας ονομάζεται ΜΑΘΗΤΕΣ όπως και η αντίστοιχη οντότητα. Συνήθως προτιμούμε τον πληθυντικό αριθμό. Δηλαδή ΜΑΘΗΤΕΣ αντί για ΜΑΘΗΤΗΣ. Ο διαγραμματικός πίνακας αποτελείται από τρεις στήλες. Τον «Τύπο Κλειδιού», το «Προαιρετικό» και το «Όνομα Στήλης».

Ο διαγραμματικός αυτός πίνακας ΜΑΘΗΤΕΣ μας λέει ότι για την οντότητα ΜΑΘΗΤΗΣ θα δημιουργηθεί κατά την υλοποίηση ο (σχεσιακός) πίνακας ΜΑΘΗΤΕΣ που θα έχει τέσσερις στήλες: Κωδικός_Μαθητή, Όνομα, Επίθετο και email, όπως φαίνεται από την στήλη «Όνομα Στήλης» του διαγραμματικού πίνακα ΜΑΘΗΤΕΣ. Οι τρεις πρώτες πληροφορίες για τον μαθητή είναι υποχρεωτικές, όπως φαίνεται από την στήλη «Προαιρετικό» του διαγραμματικού πίνακα ΜΑΘΗΤΕΣ. Δηλαδή, κάθε φορά που θα εισάγουμε έναν μαθητή πρέπει υποχρεωτικά να καταγράψουμε τον κωδικό του, το όνομα του και το επίθετό του. Η καταγραφή του email του μαθητή είναι προαιρετική. Η στήλη Κωδικός_Μαθητή θα αποτελεί το πρωτεύον κλειδί του (σχεσιακού) πίνακα ΜΑΘΗΤΕΣ όπως φαίνεται από την στήλη «Τύπος κλειδιού» του διαγραμματικού πίνακα ΜΑΘΗΤΕΣ. Δηλαδή, στον σχεσιακό πίνακα ΜΑΘΗΤΕΣ όλα τα κελιά της στήλης Κωδικός_Μαθητή θα έχουν διαφορετικές τιμές.

Ας μελετήσουμε στη συνέχεια πιο αναλυτικά τον μετασχηματισμό του μοντέλου οντοτήτων συσχετίσεων, που έχει ως αποτέλεσμα την δημιουργία του διαγραμματικού πίνακα.



Σχήμα 3.5.7: Μετασχηματισμός μοντέλου οντοτήτων συσχετίσεων

Οι κανόνες που ακολουθούνται για τον μετασχηματισμό ενός ER διαγράμματος στο σχεσιακό μοντέλο είναι οι εξής:

- Κάθε οντότητα γίνεται ένας πίνακας Κάθε γνώρισμα γίνεται στήλη
- Κάθε μοναδικό αναγνωριστικό γίνεται πρωτεύον κλειδί
- Κάθε συσχέτιση μετασχηματίζεται σε ξένο κλειδί.
- Κάθε στιγμιότυπο γίνεται γραμμή του πίνακα (στην φυσική υλοποίηση)

Στο σχήμα 3.5.7. παρουσιάζεται ο μετασχηματισμός της οντότητας ΜΑΘΗΤΗΣ με βάση τους παραπάνω κανόνες με την μορφή του διαγραμματικού πίνακα ΜΑΘΗΤΕΣ. Ο διαγραμματικός πίνακας ΜΑΘΗΤΕΣ αποτελεί την αναπαράσταση της δομής μιας σχεσιακής βάσης. Ο (σχεσιακός) πίνακας ΜΑΘΗΤΕΣ αποτελεί την μετέπειτα υλοποίηση του διαγραμματικού πίνακα ΜΑΘΗΤΕΣ.

Όπως έχουμε εξηγήσει και παραπάνω ένας διαγραμματικός πίνακας αποτελείται από τρεις στήλες. Τον «Τύπο Κλειδιού», το «Προαιρετικό» και το «Όνομα Στήλης». Η πρώτη γραμμή του διαγραμματικού πίνακα περιέχει το όνομα του σχεσιακού πίνακα. Τα κελιά στην στήλη «Όνομα Στήλης» (δηλαδή Κωδικός_Μαθητή, Όνομα, Επίθετο και email) αντιστοιχούν στα γνώρισμα της αντίστοιχης οντότητας, καθώς και στις στήλες του αντίστοιχου σχεσιακού πίνακα. Για κάθε κελί που περιέχεται στη στήλη «Όνομα Στήλης» σημειώνεται στο αντίστοιχο κελί της στήλης «Τύπος Κλειδιού» το σύμβολο «pk» ή «fk» ανάλογα με το αν το αντίστοιχο γνώρισμα αποτελεί πρωτεύον ή ξένο κλειδί. Κατά επέκταση και η αντίστοιχη στήλη του σχεσιακού πίνακα θα αποτελεί πρωτεύον ή ξένο κλειδί. Η στήλη «Προαιρετικό» περιέχει τα σύμβολα «*» ή «ο» ανάλογα με το αν το αντίστοιχο κελί στην στήλη «Όνομα Στήλης» αναφέρεται σε γνώρισμα που είναι υποχρεωτικό ή προαιρετικό.

Ας θεωρήσουμε ένα δεύτερο παράδειγμα, την συσχέτιση Εργαζόμενος – Τμήμα και ας προσπαθήσουμε να την μετασχηματίσουμε ακολουθώντας τους κανόνες, όπως φαίνεται στο παρακάτω σχήμα.

Κάθε οντότητα γίνεται ένας πίνακας

Η οντότητα ΕΡΓΑΖΟΜΕΝΟΣ γίνεται διαγραμματικός πίνακας με το όνομα ΕΡΓΑΖΟΜΕΝΟΙ. Παρόμοια, η οντότητα ΤΜΗΜΑ γίνεται διαγραμματικός πίνακας με το όνομα ΤΜΗΜΑΤΑ.

Κάθε γνώρισμα γίνεται στήλη (του αντίστοιχου σχεσιακού πίνακα)

Τα γνώρισμα Κωδικός Εργαζομένου, Όνομα και Επίθετο της οντότητας ΕΡΓΑΖΟΜΕΝΟΣ μετατρέπονται στα

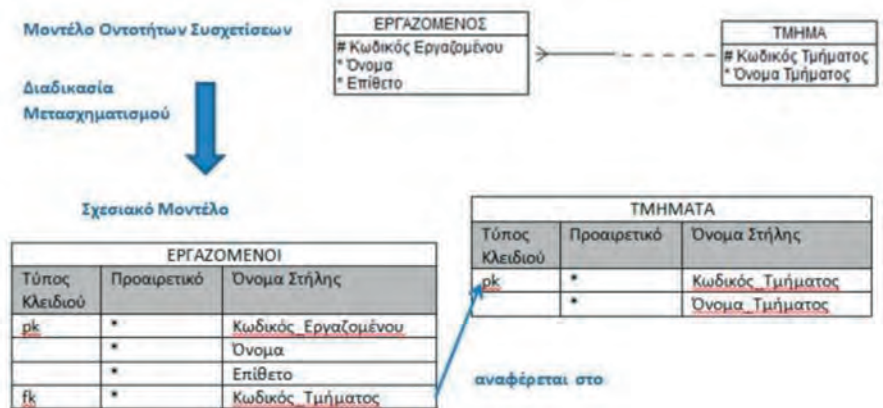
αντίστοιχα κελιά Κωδικός_Εργαζομένου, Όνομα και Επίθετο της στήλης «Όνομα Στήλης» του διαγραμματικού πίνακα ΕΡΓΑΖΟΜΕΝΟΙ. Παρόμοια και για τα γνωρίσματα της οντότητας ΤΜΗΜΑ. Τα σύμβολα του υποχρεωτικού (*) ή του προαιρετικού (ο) των γνωρισμάτων ακολουθούν και τα αντίστοιχα κελιά στην στήλη «Προαιρετικό».

Κάθε μοναδικό αναγνωριστικό γίνεται πρωτεύον κλειδί

Αν ένα γνώρισμα αποτελεί μοναδικό αναγνωριστικό (πρωτεύον κλειδί) για την οντότητα, τότε στο αντίστοιχο κελί της στήλης «Τύπος Κλειδί» σημειώνεται το σύμβολο pk. Στο παράδειγμά μας δηλαδή, τα κελιά Κωδικός_Εργαζομένου και Κωδικός_Τμήματος θα είναι πρωτεύοντα κλειδιά αντίστοιχα για τους (σχεσιακούς) πίνακες ΕΡΓΑΖΟΜΕΝΟΙ και ΤΜΗΜΑΤΑ.

Κάθε συσχέτιση μετασχηματίζεται σε ξένο κλειδί.

Το ξένο κλειδί τοποθετείται σε εκείνον τον διαγραμματικό πίνακα που βρίσκεται στη θέση του «πολλά». Η συσχέτιση μεταξύ ΤΜΗΜΑ και ΕΡΓΑΖΟΜΕΝΟΣ είναι «ένα προς πολλά». Δηλαδή σε ένα τμήμα είναι δυνατό να εργάζονται περισσότεροι από ένας εργαζόμενοι. Το «πολλά» αντιστοιχεί στην οντότητα ΕΡΓΑΖΟΜΕΝΟΣ. Για αυτό και στον διαγραμματικό πίνακα ΕΡΓΑΖΟΜΕΝΟΙ δημιουργείται ένα επιπλέον κελί με το όνομα Κωδικός_Τμήματος που έχει τον ρόλο του ξένου κλειδιού. Η στήλη «Τύπος Κλειδί» στο αντίστοιχο κελί περιέχει το σύμβολο «fk» για να δηλώσουμε ότι είναι ξένο κλειδί. Τα δύο κλειδιά ξένο και πρωτεύον στους δύο διαγραμματικούς πίνακες συνδέονται μεταξύ τους με ένα βέλος για να δείξουμε την μεταξύ τους σχέση.



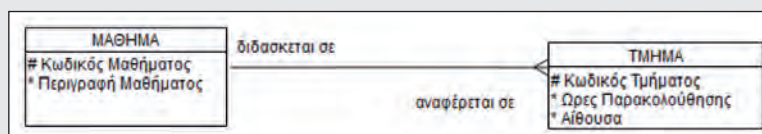
Σχήμα 3.5.8: ο μετασχηματισμός ενός ER διαγράμματος σε σχεσιακό μοντέλο.



Ερωτήσεις / Δραστηριότητες

Δραστηριότητα: Μετασχηματισμός ER διαγράμματος

Μετασχηματίστε το ακόλουθο ER διάγραμμα που απεικονίζει την σχέση μεταξύ «μαθήματος» και «τμήμα» σε σχεσιακό μοντέλο. Δημιουργήστε τόσο τους διαγραμματικούς πίνακες όσο και τους σχεσιακούς πίνακες. «Γεμίστε» τους σχεσιακούς πίνακες με ενδεικτικά στοιχεία. Οι φοιτητές στο πανεπιστήμιο παρακολουθούν διάφορα μαθήματα. Για παράδειγμα, στο πρώτο εξάμηνο διδάσκεται το μάθημα «Γλώσσα Προγραμματισμού Java». Λόγου του μεγάλου αριθμού των φοιτητών, οι φοιτητές χωρίζονται σε δύο τμήματα. Το ένα τμήμα παρακολουθεί το μάθημα αυτό κάθε Δευτέρα και Τετάρτη και ώρα 10.00 – 12.00, ενώ το άλλο τμήμα κάθε Τετάρτη και Παρασκευή και ώρα 17.00 – 19.00.



Σχήμα 3.5.9: ER διάγραμμα

κεφάλαιο

4

Η Γλώσσα Βάσεων Δεδομένων SQL

Εισαγωγή



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να κατανοεί τη σημασία των γλωσσών βάσεων δεδομένων
- να κατανοεί τη σημασία μιας πρότυπης γλώσσας βάσεων δεδομένων
- να κατανοεί τη διαφορά ανάμεσα στον ορισμό και τον χειρισμό των δεδομένων
- να αναγνωρίζει τη σημασία της SQL στην εξέλιξη και επιτυχία των σχεσιακών βάσεων δεδομένων

Η γλώσσα **SQL (Structured Query Language)** είναι γλώσσα διαχείρισης Σχεσιακών Βάσεων Δεδομένων. Η SQL αναπτύχθηκε στην εταιρεία IBM στις αρχές της δεκαετίας του 1970 και αρχικά ονομαζόταν SEQUEL. Η γλώσσα SQL θεωρείται μια από τις κύριες αιτίες της εμπορικής επιτυχίας των Σχεσιακών Βάσεων Δεδομένων, επειδή ήταν απλή και εύκολη στη χρήση της.



SQL = Structured Query Language
(προφέρεται és-qu-él ή síkwel)

Το 1986 η SQL έγινε πρότυπο (standard) από τον αμερικανικό οργανισμό προτύπων ANSI (American National Standards Institute) και το 1987 από το διεθνή οργανισμό ISO (International Organization for Standardization). Από τότε έχουν γίνει επτά (7) αναθεωρήσεις της SQL με πιο πρόσφατη αυτή του 2011.

Η καθιέρωση της SQL ως προτύπου σημαίνει ότι όλοι μπορούν να την χρησιμοποιούν με τον ίδιο ακριβώς τρόπο σε οποιοδήποτε σύστημα την υποστηρίζει. Έτσι, αν οι σχεδιαστές ή οι διαχειριστές Βάσεων Δεδομένων προγραμματίσουν σε SQL, οι εντολές που θα γράψουν θα μπορούν εύκολα να μεταφερθούν σε οποιοδήποτε σύστημα υποστηρίζει το πρότυπο της SQL. Θα μπορούν δηλαδή να μεταφέρουν εύκολα τον κώδικά τους από το ένα σύστημα στο άλλο, π.χ. από MySQL σε Postgres ή αντίστροφα, κλπ. Επιπλέον, αν ένας προγραμματιστής εφαρμογών χρησιμοποιήσει τη γλώσσα SQL για να γράψει ερωτήσεις ανάκτησης δεδομένων, τότε θα μπορεί να κάνει τις ίδιες ακριβώς ερωτήσεις σε διαφορετικές βάσεις δεδομένων που υποστηρίζουν το πρότυπο της SQL, για να αντλεί τα δεδομένα των εφαρμογών του.

Η SQL αποτελείται από δύο βασικά μέρη: τη **γλώσσα SQL Ορισμού Δεδομένων (SQL DDL)** και τη **γλώσσα SQL χειρισμού δεδομένων (SQL DML)**. Με την γλώσσα ορισμού δεδομένων ορίζεται το σχήμα των δεδομένων, δηλαδή η μορφή τους, ενώ με τη γλώσσα χειρισμού δεδομένων ορίζονται οι λειτουργίες αποθήκευσης και ανάκτησης των δεδομένων. Επομένως η γλώσσα SQL περιέχει εντολές για τη δημιουργία και την επεξεργασία της μορφής των πινάκων (ορισμός δεδομένων), αλλά και εντολές για την εισαγωγή, ενημέρωση, διαγραφή και ανάκτηση των δεδομένων που είναι αποθηκευμένα σε πίνακες (χειρισμός δεδομένων).



DDL = Data Definition Language, δηλ.
Γλώσσα Ορισμού Δεδομένων
DML = Data Manipulation Language, δηλ.
Γλώσσα Χειρισμού Δεδομένων

Αν για παράδειγμα θέλουμε να αποθηκεύσουμε τα στοιχεία των μαθητών ενός σχολείου σε σχεσιακή βάση δεδομένων, θα χρησιμοποιήσουμε τη γλώσσα SQL DDL για να δημιουργήσουμε τους σχεσιακούς πίνακες και κατόπιν θα χρησιμοποιήσουμε τη γλώσσα SQL DML για να εισάγουμε τα στοιχεία των μαθητών και για να κάνουμε ερωτήσεις που θα ανακτούν δεδομένα από τη βάση. Το παράδειγμα του Μαθητολογίου αποτελεί το βασικό παράδειγμα της συγκεκριμένης ενότητας και παρουσιάζεται αναλυτικά παρακάτω.



Σε ένα Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων:

- Δημιουργώ πίνακες με τη Γλώσσα Ορισμού Δεδομένων (DDL)
- Εισάγω δεδομένα στους σχεσιακούς πίνακες με τη Γλώσσα Χειρισμού Δεδομένων (DML)
- Ενημερώνω/διαγράφω τα δεδομένα των πινάκων με τη Γλώσσα Χειρισμού Δεδομένων (DML)
- Εκτελώ ερωτήματα ανάκτησης για τα δεδομένα των πινάκων με τη Γλώσσα Χειρισμού Δεδομένων (DML)

Φυσικά η δημιουργία πινάκων και η εισαγωγή/ενημέρωση/διαγραφή δεδομένων σε αυτούς μπορεί να γίνει και με γραφικά εργαλεία στα περισσότερα ΣΔΒΔ, όμως η χρήση της γλώσσας SQL εξασφαλίζει ανεξαρτησία από τον κατασκευαστή του συστήματος και εύκολη μεταφορά των προγραμμάτων ανάμεσα σε συστήματα κατασκευαστών.

Η ενότητα [4.1](#) παρουσιάζει βασικές λειτουργίες ορισμού δεδομένων σε σχεσιακή βάση με την SQL, ενώ η ενότητα [4.2](#) παρουσιάζει λειτουργίες χειρισμού δεδομένων.



Ερωτήσεις / Δραστηριότητες

1. Σε ένα σχολείο, οι ανάγκες μηχανογράφησης επιβάλλουν τη χρήση σχεσιακής βάσης δεδομένων για την διαχείριση των στοιχείων μαθητών, μαθημάτων, εκπαιδευτικών, αναθέσεων, βαθμολογιών κλπ. Γι' αυτό το σκοπό και για λόγους οικονομίας, το σχολείο αποφασίζει την εγκατάσταση δωρεάν (free) ΣΔΒΔ. Ειδικός σε ΣΔΒΔ αναλαμβάνει την εγκατάσταση και προετοιμασία (διαχείριση) του συστήματος ώστε αυτό να μπορεί να χρησιμοποιηθεί από τη σχολική κοινότητα. Ενέργειες στις οποίες πρέπει να προβεί είναι ο ορισμός των δεδομένων και ο καθορισμός των λειτουργιών χειρισμού δεδομένων που ανταποκρίνονται στις ανάγκες του προβλήματος και των χρηστών του.
 - α) Εντοπίστε και περιγράψτε ενέργειες ορισμού δεδομένων (δημιουργία πινάκων) που πρέπει να γίνουν από τον διαχειριστή του συστήματος πριν αυτό διατεθεί προς χρήση.
 - β) Εντοπίστε και περιγράψτε ενέργειες χειρισμού δεδομένων (εισαγωγές, ενημερώσεις, διαγραφές) που θα πραγματοποιούνται από τους χρήστες του συστήματος.

4.1 SQL Σχεσιακή Γλώσσα Ορισμού Δεδομένων (DDL)



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια να είναι ικανός/ή:

- να περιγράφει τα χαρακτηριστικά της Γλώσσας Ορισμού Δεδομένων (DDL)
- να διακρίνει τη σημασία των Γλωσσών Ορισμού Δεδομένων στην οργάνωση και διαχείριση των Βάσεων Δεδομένων

Μία **Γλώσσα Ορισμού Δεδομένων (Data Definition Language DDL)** χρησιμοποιείται από τους σχεδιαστές προγραμματιστές και από τους διαχειριστές Βάσεων Δεδομένων για τον ορισμό του σχήματος των δεδομένων, δηλαδή για να καθοριστεί η μορφή με την οποία θα «βλέπουν» οι χρήστες και τα προγράμματα τα αποθηκευμένα δεδομένα.

Τα ΣΔΒΔ διαθέτουν μεταφραστή της γλώσσας DDL που υποστηρίζουν, ώστε να μετατρέπεται η περιγραφή από DDL σε μορφή που μπορεί να αποθηκευτεί.

Τα σχεσιακά ΣΔΒΔ χρησιμοποιούν κατά κανόνα τη γλώσσα SQL για τον ορισμό του σχεσιακού σχήματος των δεδομένων.

Η γλώσσα SQL DDL περιγράφει τους πίνακες, τα πεδία (ή χαρακτηριστικά ή στήλες) από τα οποία αποτελούνται οι πίνακες, το είδος ή τύπο δεδομένων κάθε πεδίου και τους περιορισμούς για τα πεδία των πινάκων.

Στην υποενότητα [4.1.1](#) παρουσιάζονται οι τύποι δεδομένων που υποστηρίζει η SQL για τη δημιουργία των πεδίων πινάκων, ενώ στην υποενότητα [4.1.2](#) παρουσιάζονται οι εντολές δημιουργίας πινάκων σε SQL.

4.1.1 Τύποι Δεδομένων της SQL



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει βασικούς τύπους δεδομένων της γλώσσας SQL και να δίνει παραδείγματα τιμών για κάθε τύπο
- να διακρίνει τις διαφορές μεταξύ των τύπων δεδομένων της SQL και να αντιστοιχίζει χαρακτηριστικά οντοτήτων και πεδία σχέσεων με τους τύπους δεδομένων της SQL
- να περιγράφει πεδία σχέσεων με τους αντίστοιχους SQL τύπους τους για συγκεκριμένα προβλήματα

Οι περισσότερες γλώσσες προγραμματισμού έχουν **τύπους δεδομένων**, δηλαδή συγκεκριμένους τρόπους αναπαράστασης των δεδομένων. Αυτό σημαίνει ότι δίνουν στον προγραμματιστή συγκεκριμένες επιλογές για το είδος των στοιχείων που θα επεξεργαστούν τα προγράμματά του. Για παράδειγμα, σε ένα πρόγραμμα που επεξεργάζεται βαθμούς μαθητών, ο προγραμματιστής θα ορίσει τους προφορικούς βαθμούς των τετραμήνων ως ακέραιους αριθμούς. Με άλλα λόγια, ο τύπος δεδομένων των προφορικών βαθμών θα είναι ακέραιος αριθμός.

Η SQL που είναι γλώσσα προγραμματισμού Βάσεων Δεδομένων, έχει τους δικούς της τύπους δεδομένων, οι οποίοι χρησιμοποιούνται για να καθοριστεί το είδος των στοιχείων που θα αποθηκεύονται στους σχεσιακούς πίνακες.

Για να δημιουργηθεί ένας πίνακας με SQL, πρέπει πρώτα να καθοριστούν τα πεδία από τα οποία θα αποτελείται ο πίνακας. Τα πεδία καθορίζονται από το όνομά τους και τον τύπο τους.

Το όνομα κάθε πεδίου το επιλέγει ο σχεδιαστής και είναι μία λέξη, π.χ. Τίτλος, Ονοματεπώνυμο, Ημερομηνία_γέννησης, Έτος_δημιουργίας, Βαθμός, κλπ.

Ο τύπος ενός πεδίου καθορίζει το είδος των δεδομένων που μπορούν να αποθηκευτούν σε αυτό το πεδίο. Ο τύπος αυτός είναι ένας από τους τύπους δεδομένων που υποστηρίζει η γλώσσα SQL. Πριν μάθουμε επομένως πώς δημιουργούμε έναν πίνακα σε SQL, πρέπει να μάθουμε ποιοι είναι οι τύποι δεδομένων της SQL.

Η SQL υποστηρίζει διάφορους τύπους δεδομένων: αριθμητικούς, αλφαριθμητικούς, ημερομηνίας, ώρας, κλπ οι οποίοι επιλέγονται από το σχεδιαστή της βάσης ανάλογα με τις ανάγκες. Για παράδειγμα, σε μια βάση δεδομένων που αποθηκεύει στοιχεία για ταινίες κινηματογράφου, ο τίτλος της ταινίας είναι μια φράση που αποτελείται από αλφαριθμητικά στοιχεία, δηλαδή γράμματα και σύμβολα. Ο τύπος του πεδίου «Τίτλος» θα είναι επομένως αλφαριθμητικός. Το κόστος δημιουργίας μιας ταινίας όμως, είναι αριθμητικού τύπου. Στην Εικόνα 41 βλέπουμε τον πίνακα «Ταινία» που αποτελείται από τις στήλες «Τίτλος», «Έτος_δημιουργίας», «Σκηνοθέτης» και «Κόστος» καθώς και τιμές για την ταινία «Το κύμα (Die Welle)».

Τίτλος	Έτος_δημιουργίας	Σκηνοθέτης	Κόστος
Το κύμα (Die Welle)	2008	Dennis Gansel	5.000.000

Εικόνα 4-1. Ο σχεσιακός πίνακας «Ταινία» με δεδομένα για ταινίες κινηματογράφου

Σε ένα διαφορετικό παράδειγμα, αν θέλουμε να αποθηκεύσουμε σε έναν πίνακα τιμές προϊόντων, η τιμή κάθε προϊόντος σε ευρώ πρέπει να αποθηκευτεί ως αριθμός με υποδιαστολή, π.χ. 4,25 .

Στα σχολεία, για να διευκολυνθεί η λειτουργία τους, υπάρχει η ανάγκη αποθήκευσης δεδομένων για τους καθηγητές, τους μαθητές, τα μθήματα, τα μαθήματα, τις απουσίες, τους βαθμούς των μαθητών, κλπ. Τα στοιχεία των μαθητών περιλαμβάνουν το ονοματεπώνυμο και το πατρώνυμό τους. Οπότε μπορούμε να πούμε για αρχή, ότι ο πίνακας με τα στοιχεία των μαθητών θα μοιάζει με τον εξής:

Επώνυμο	Όνομα	Πατρώνυμο
Ομήρου	Οδύσσεια	Όμηρος
Πηλείδης	Αχιλλέας	Πηλέας

Εικόνα 4-2. Σχεσιακός πίνακας με το επώνυμο, το όνομα και το πατρώνυμο των μαθητών

Αριθμητικοί τύποι δεδομένων της SQL

Δύο βασικοί αριθμητικοί τύποι δεδομένων που υποστηρίζονται από την SQL είναι οι ακέραιοι και οι πραγματικοί αριθμοί. Οι ακέραιοι αριθμοί δεν έχουν υποδιαστολή, ενώ οι πραγματικοί έχουν. Η λέξη που χρησιμοποιεί η SQL για τον ορισμό πεδίων ακέραιου τύπου είναι INTEGER, ενώ για τον ορισμό πεδίων πραγματικού τύπου είναι FLOAT. Ο Πίνακας 41 συνοψίζει δύο βασικούς αριθμητικούς τύπους δεδομένων της SQL για ακέραιους και πραγματικούς αριθμούς.

Τύπος δεδομένων	Περιγραφή
INTEGER ή INT	Ακέραιος αριθμός (χωρίς υποδιαστολή)
FLOAT	Αριθμός με υποδιαστολή

Πίνακας 4-1. Αριθμητικοί τύποι δεδομένων FLOAT, INTEGER της SQL

Στο παράδειγμα των ταινιών κινηματογράφου επομένως που αναφέρθηκε παραπάνω, το πεδίο «Κόστος» είναι ακέραιου τύπου, δηλαδή τύπου INTEGER. Στο παράδειγμα με τα προϊόντα, το πεδίο «Τιμή_προϊόντος» είναι πραγματικού τύπου, δηλαδή τύπου FLOAT γιατί οι τιμές έχουν και δεκαδικό μέρος.

Σε μια βάση δεδομένων Μαθητολογίου για την αποθήκευση και διαχείριση των δεδομένων του σχολείου, καταγράφεται η πληροφορία για τα μαθήματα του προγράμματος σπουδών. Έτσι δημιουργείται ο πίνακας «Μάθημα» που αποτελείται από τις στήλες «Όνομα» και «Ώρες» και σε αυτόν αποθηκεύονται οι πληροφορίες για όλα τα μαθήματα που παρακολουθούν οι μαθητές του σχολείου. Οι εβδομαδιαίες ώρες διδασκαλίας των μαθημάτων που αποθηκεύονται στη στήλη «Ώρες», είναι ακέραιος αριθμός.

Όνομα	Ώρες
Συστήματα Διαχείρισης Βάσεων Δεδομένων και εφαρμογές τους στο Web	5

Εικόνα 4-3. Ο σχεσιακός πίνακας «Μάθημα»

Στην Εικόνα 4-3 βλέπουμε τον πίνακα «Μάθημα», στον οποίον έχει αποθηκευτεί η πληροφορία για το μάθημα «Συστήματα Διαχείρισης Βάσεων Δεδομένων και εφαρμογές τους στο Web».

Αλφαριθμητικοί τύποι δεδομένων της SQL

Οι αλφαριθμητικοί τύποι δεδομένων υποστηρίζουν την αποθήκευση κειμένου, δηλαδή συμβολοσειρών ή σειρών από χαρακτήρες, γι' αυτό και συχνά ονομάζονται τύποι 'χαρακτήρων'. Η λέξη που χρησιμοποιεί η SQL για τον ορισμό πεδίων τύπου χαρακτήρα (αλφαριθμητικού) είναι CHARACTER. Η SQL δίνει τη δυνατότητα ορισμού του μεγέθους μιας σειράς χαρακτήρων με την έκφραση CHARACTER(N), όπου στη θέση του N τοποθετείται ένας ακέραιος αριθμός που δηλώνει το μέγεθος του πεδίου, δηλαδή περιορίζει τις τιμές σε μέγεθος N. Ο τύπος CHARACTER VARYING(N) ή VARCHAR(N) χρησιμοποιείται για τη δημιουργία πεδίων τύπου χαρακτήρων μεταβλητού μεγέθους N. Ο Πίνακας 42 συνοψίζει τους τύπους CHAR και VARCHAR της SQL.

Τύπος δεδομένων	Περιγραφή
CHARACTER(N) ή CHAR(N)	Σειρά χαρακτήρων (συμβόλων) σταθερού μήκους N. Το μήκος N αναφέρεται στο μέγιστος μήκος της τιμής. Οι τιμές για πεδία αυτού του τύπου πρέπει να γράφονται μέσα σε μονά εισαγωγικά ``, π.χ. `Γιώργος`
CHARACTER VARYING(N) ή VARCHAR(N)	Συμβολοσειρά χαρακτήρων μεταβλητού μεγέθους, μέγιστου N.

Πίνακας 4-2. Αλφαριθμητικοί τύποι δεδομένων CHAR, VARCHAR της SQL

Στο παράδειγμα των μαθημάτων της προηγούμενης παραγράφου, το πεδίο «Όνομα_μαθήματος» είναι αλφαριθμητικού τύπου και επειδή κάθε όνομα έχει διαφορετικό πλήθος χαρακτήρων, το συγκεκριμένο πεδίο θα μπορούσε να οριστεί ως τύπου VARCHAR(100) ώστε να μπορέσει να αποθηκεύσει ονόματα μεταβλητού μεγέθους και αρκετά μεγάλα ονόματα, μέχρι 100 χαρακτήρων. Στο Μαθητολόγιο του σχολείου, ένα άλλο πεδίο που αποθηκεύεται για τους μαθητές είναι το φύλο τους. Το φύλο μπορεί να πάρει τις τιμές 'Άρρεν' και 'Θήλυ' ή σε συντομία 'Α' ή 'Θ'. Σε αυτή την περίπτωση το φύλο μπορεί να οριστεί ως τύπου CHARACTER(1).

Τύποι δεδομένων για την ημερομηνία και την ώρα

Πεδία που αποθηκεύουν ημερομηνία ή ώρα μπορούν να οριστούν ως τύπου DATE, TIME ή TIMESTAMP. Ο τύπος DATE μπορεί να αποθηκεύσει δεδομένα που αποτελούνται από το έτος (YEAR) με τιμές από 0001 ως 9999, το μήνα (MONTH) και την ημέρα (DAY). Ο τύπος TIME μπορεί να αποθηκεύσει τιμές που αποτελούνται από την ώρα (HOUR), τα λεπτά (MINUTE) και τα δευτερόλεπτα (SECOND). Ο τύπος TIMESTAMP μπορεί να αποθηκεύσει τιμές που αποτελούνται από το έτος, το μήνα, την ημέρα, την ώρα, τα λεπτά και τα δευτερόλεπτα. Ο Πίνακας 43 συνοψίζει τους τύπους DATE, TIME και TIMESTAMP.

Τύπος δεδομένων	Περιγραφή
DATE, TIME, TIMESTAMP	Τύποι δεδομένων για την ημερομηνία και την ώρα. Αποτελούνται από ακέραιους αριθμούς (YEAR, MONTH, DAY, HOUR, MINUTE, SECOND) και παριστάνουν απόλυτες χρονικές στιγμές.

Πίνακας 4-3. Τύποι δεδομένων DATE, TIME και TIMESTAMP της SQL

Επομένως, τιμές όπως '24062015' μπορούν να αποθηκευτούν σε πεδία τύπου DATE, τιμές όπως '15:02:34' μπορούν να αποθηκευτούν σε πεδία τύπου TIME, ενώ τιμές όπως '24062015 15:02:34'. Στο παράδειγμα του Μαθητολογίου, η ημερομηνία γέννησης ενός μαθητή μπορεί να αποθηκευτεί σε πεδίο τύπου DATE. Στον Πίνακα 43 μπορούμε να δούμε μια εγγραφή στον πίνακα «Μαθητής» με το πεδίο το «Επώνυμο», το «Όνομα», το «Πατρώνυμο» και την «Ημερομηνία_γέννησης» του μαθητή/της μαθήτριας.

Επώνυμο	Όνομα	Πατρώνυμο	Ημερομηνία_Γέννησης
Ομήρου	Οδύσσεια	Όμηρος	01051999
Πηλείδης	Αχιλλέας	Πηλέας	24061999

Εικόνα 4-4. Ο σχεσιακός πίνακας «Μαθητής» (Επώνυμο, Όνομα, Πατρώνυμο, Ημερομηνία_Γέννησης)

Οι τιμές τύπου ημερομηνίας και ώρας μπορούν να συγκριθούν μεταξύ τους αλλά και να αφαιρεθούν ή προστεθούν μεταξύ τους.

Στο ΠΑΡΑΡΤΗΜΑ 4.1: Οι Τύποι Δεδομένων της SQL γίνεται απαρίθμηση όλων των τύπων δεδομένων που υποστηρίζει η SQL.



Ερωτήσεις / Δραστηριότητες

- 1) Σε βάση δεδομένων ταινιών κινηματογράφου, τι είδους δεδομένα αποθηκεύονται: α) στο έτος δημιουργίας της ταινίας, β) στο ονοματεπώνυμο του σκηνοθέτη και γ) στο κόστος δημιουργίας της ταινίας;
- 2) Περιγράψτε τον τύπο των δεδομένων για έναν πίνακα «Βιβλίο» που θα αποτελείται από τα εξής πεδία : Τίτλος, Συγγραφέας, Εκδοτικός Οίκος, Τιμή.

Δραστηριότητα

- 1) Δίνεται ο παρακάτω πίνακας με δεδομένα που αφορούν σε βιβλία:

Ιλιάδα	Όμηρος	2003	18,37
Ερωτόκριτος	Κορνάρος, Βιτσέντζος	1985	12,75

- α) Αποδώστε ονόματα στα πεδία του πίνακα.
- β) Διακρίνετε τα πεδία του πίνακα σε αριθμητικού, αλφαριθμητικού τύπου και τύπου ημερομηνίας-ώρας.
- γ) Αποδώστε τύπους δεδομένων της SQL σε καθένα από τα πεδία του πίνακα.

4.1.2 Δημιουργία Πίνακα



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να δημιουργεί πίνακες σε περιβάλλον ΣΣΔΒΔ
- καθορίζει περιορισμούς στα πεδία σχεσιακών πινάκων
- αναγνωρίζει και να καθορίζει το κύριο κλειδί σε σχεσιακούς πίνακες και να τους δημιουργεί με κύριο κλειδί σε περιβάλλον ΣΣΔΒΔ
- να αναγνωρίζει τα ξένα κλειδιά σε σχεσιακούς πίνακες και να δημιουργεί περιορισμούς ξένων κλειδίων σε περιβάλλον ΣΣΔΒΔ

Μετά την παρουσίαση βασικών τύπων δεδομένων μπορούμε να δημιουργήσουμε πίνακες χρησιμοποιώντας τη γλώσσα SQL. Η εντολή της SQL που μας δίνει αυτή τη δυνατότητα είναι η CREATE TABLE, η οποία χρησιμοποιείται ως εξής:

```
CREATE TABLE <όνομα_πίνακα>
(
    <όνομα_στήλης 1>      <τύπος_στήλης 1>,
    <τύπος_στήλης 2>,      <όνομα_στήλης 2>,
    ...
    <όνομα_στήλης N>      <τύπος_στήλης N>
)
```

Δηλαδή, ξεκινάμε δηλαδή με τη φράση CREATE TABLE, ακολουθούμενη από έναν ή περισσότερους κενούς χαρακτήρες και κατόπιν γράφουμε το όνομα του πίνακα που θέλουμε να δημιουργήσουμε. Κατόπιν ανοίγουμε παρένθεση '(' και γράφουμε τα ονόματα και τους τύπους δεδομένων των στηλών του πίνακα. Για κάθε στήλη (πεδίο) του πίνακα καθορίζουμε το όνομά της και μετά από ένα ή περισσότερα κενά τον τύπο δεδομένων της στήλης. Οι τύποι δεδομένων είναι αυτοί που παρουσιάστηκαν στην προηγούμενη ενότητα. Οι ορισμοί των στηλών μεταξύ τους διαχωρίζονται με κόμμα ','. Όταν ολοκληρωθεί ο ορισμός των στηλών κλείνουμε την παρένθεση ')'. Κενός χαρακτήρας πρέπει να υπάρχει ανάμεσα στο όνομα και τον τύπο κάθε στήλης.

Αν επιστρέψουμε στα παραδείγματά μας, για να δημιουργήσουμε τον πίνακα Μάθημα, θα γράψουμε την εξής εντολή:

```
CREATE TABLE Μάθημα
(
    Όνομα VARCHAR(100),
    Ωρες INTEGER
)
```

Ο πίνακας Μάθημα αποτελείται από δύο πεδία: Όνομα και Ωρες. Ο τύπος του πεδίου "Όνομα" είναι συμβολοσειρά μέχρι 100 χαρακτήρων (VARCHAR(100)), ενώ ο τύπος του πεδίου "Ωρες" είναι ακέραιος αριθμός (INTEGER).

Για να δημιουργήσουμε τον πίνακα Μαθητής όπως παρουσιάζεται στην Εικόνα 44, προσθέτοντας το πεδίο «Φύλο», θα γράψουμε την εξής εντολή:

```
CREATE TABLE Μαθητής
(
  Επώνυμο VARCHAR(50),
  Όνομα VARCHAR (50),
  Πατρώνυμο VARCHAR (50),
  Ημερομηνία_Γέννησης DATE,
  Φύλο CHAR(1)
)
```

Τα πεδία αλφαριθμητικού τύπου «Επώνυμο», «Όνομα» και «Πατρώνυμο» είναι τύπου χαρακτήρα μεταβλητού μεγέθους με μήκος έως 50, δηλαδή VARCHAR(50). Το πεδίο «Ημερομηνία_Γέννησης» είναι τύπου ημερομηνίας, δηλαδή DATE, ενώ το πεδίο «Φύλο» είναι ένας χαρακτήρας ('Α' ή 'Θ').

Προκαθορισμένες τιμές και περιορισμοί στις τιμές των πεδίων

Κατά τη δημιουργία πινάκων με την εντολή CREATE TABLE μπορούν να καθοριστούν προκαθορισμένες τιμές και περιορισμοί για τα πεδία των πινάκων.

Προκαθορισμένη τιμή πεδίων

Κατά την εισαγωγή δεδομένων σε έναν πίνακα, είναι πιθανό κάποιες τιμές πεδίων να μην είναι γνωστές, οπότε να μην εισάγονται. Σε αυτή την περίπτωση, το πεδίο θα έχει την τιμή NULL, που σημαίνει ότι δεν έχει εισαχθεί τιμή.

Για παράδειγμα όταν γίνονται εγγραφές μαθητών στο Μαθητολόγιο του σχολείου, μπορεί να μην εισάγεται η ημερομηνία γέννησης του μαθητή μέχρι εκείνος να προσκομίσει το πιστοποιητικό γέννησης. Σε αυτή την περίπτωση η ημερομηνία γέννησης έχει την τιμή NULL μέχρι να γίνει εισαγωγή μιας έγκυρης ημερομηνίας.

Προκαθορισμένη τιμή (default value) σε ένα πεδίο σημαίνει ότι αν δεν καθοριστεί άλλη τιμή κατά την εισαγωγή δεδομένων, χρησιμοποιείται μία τιμή που έχει καθορίσει ο σχεδιαστής της Βάσης Δεδομένων.

Για παράδειγμα, αν στο Μαθητολόγιο καταχωρείται η πόλη καταγωγής των μαθητών ενός σχολείου, το πεδίο αυτό θα μπορούσε να έχει προκαθορισμένη τιμή συγκεκριμένη πόλη π.χ. «Αθήνα» αν πρόκειται για σχολείο της Αθήνας και αναμένεται οι περισσότεροι μαθητές να έχουν καταγωγή από την Αθήνα. Σε αυτή την περίπτωση η προκαθορισμένη τιμή διευκολύνει την εισαγωγή δεδομένων εφόσον δεν είναι απαραίτητο να εισάγεται με κάθε νέο μαθητή, αλλά δημιουργείται αυτόματα από το σύστημα.

Στην γλώσσα SQL, οι προκαθορισμένες τιμές περιγράφονται στην εντολή CREATE TABLE με την σύνταξη DEFAULT <τιμή> δίπλα από την περιγραφή της αντίστοιχης στήλης.

Το παράδειγμα δημιουργίας του πίνακα Μαθητής θα γινόταν επομένως :

```
CREATE TABLE Μαθητής
(
  Επώνυμο VARCHAR(50),
  Όνομα VARCHAR (50),
  Πατρώνυμο VARCHAR (50),
  Ημερομηνία_Γέννησης DATE,
  Φύλο CHAR(1),
  Πόλη VARCHAR(50) DEFAULT 'Αθήνα'
)
```

Περιορισμοί

Συχνά είναι αναγκαίο να επιβληθούν περιορισμοί στα δεδομένα για να εξασφαλιστεί η ορθότητα και η ακεραιότητά τους. Οι περιορισμοί με τους οποίους θα ασχοληθούμε σε αυτήν την υποενότητα είναι: περιορισμοί υπαρξιακής ακεραιότητας, περιορισμοί κύριου κλειδιού και περιορισμοί αναφορικής ακεραιότητας.

Περιορισμοί υπαρξιακής ακεραιότητας

Οι περιορισμοί αυτοί μπορεί να αφορούν στην ύπαρξη ή μη τιμών σε κάποιο πεδίο πίνακα και εκφράζονται σε γλώσσα SQL με τις φράσεις NULL και NOT NULL. Η λέξη «NULL» σημαίνει καμία τιμή και χρησιμοποιείται για πεδία που είναι αποδεκτό να μην έχουν τιμή, ενώ η φράση «NOT NULL» χρησιμοποιείται για πεδία που δεν επιτρέπεται να μην έχουν τιμή.

Στο παράδειγμα του Μαθητολογίου, η καταχώρηση μαθημάτων χωρίς τίτλους δεν είναι αποδεκτή, οπότε η δημιουργία του πίνακα Μάθημα θα έπρεπε να γραφτεί προσθέτοντας δίπλα στο πεδίο

«Όνομα» τη φράση NOT NULL, ως εξής:

```
CREATE TABLE Μάθημα(  
Όνομα VARCHAR(100) NOT NULL,  
Ωρες INT  
)
```

Παρόμοια, στον πίνακα Μαθητής, η καταχώρηση μαθητών χωρίς το ονοματεπώνυμό τους δεν είναι αποδεκτή, οπότε και στα δύο πεδία καθορίζεται ο περιορισμός NOT NULL ώστε να αποφευχθεί η εισαγωγή μαθητών χωρίς ονοματεπώνυμο.

```
CREATE TABLE Μάθημα(  
Επώνυμο VARCHAR(50) NOT NULL,  
Όνομα VARCHAR(50) NOT NULL,  
Πατρώνυμο VARCHAR(50),  
Ημερομηνία_Γέννησης DATE,  
Φύλο CHAR(1),  
Πόλη VARCHAR(50) DEFAULT 'Αθήνα'  
)
```

Περιορισμοί κύριου κλειδιού

Το κύριο κλειδί ενός σχεσιακού πίνακα είναι ένα πεδίο ή συνδυασμός πεδίων που για κάθε γραμμή του πίνακα έχει μοναδική τιμή. Ο περιορισμός του κύριου κλειδιού μπορεί να καθοριστεί στην εντολή CREATE TABLE με τη σύνταξη PRIMARY KEY (<πεδίο ή πεδία>). Η φράση PRIMARY KEY (<πεδίο ή πεδία>) προστίθεται μετά τον ορισμό των στηλών του πίνακα στην εντολή CREATE TABLE.

Ας επανέλθουμε στο παράδειγμα του Μαθητολογίου και στον πίνακα Μάθημα που δημιουργήθηκε προηγουμένως. Ένας ορθός σχεδιασμός της βάσης δεδομένων επιβάλλει τον καθορισμό κύριου κλειδιού σε όλους τους πίνακες που θα δημιουργηθούν. Άρα στον πίνακα Μάθημα πρέπει να προστεθεί κύριο κλειδί. Σε αυτή την περίπτωση επιλέγουμε ως κύριο κλειδί το όνομα του μαθήματος, το οποίο θεωρούμε ότι είναι μοναδικό, π.χ. «Φυσική Α' Τάξης», «Μαθηματικά Γ' Τάξης», κ.ο.κ. Άρα η δημιουργία του πίνακα Μάθημα θα συμπληρωθεί ως εξής:

```
CREATE TABLE Μάθημα
(
Όνομα VARCHAR(100) NOT NULL,
Ωρες INTEGER,
PRIMARY KEY (Όνομα)
)
```

Στο παράδειγμα του Μαθητολογίου, θέλουμε οι μαθητές επίσης να διακρίνονται μεταξύ τους με μοναδικό τρόπο. Αυτό εφαρμόζεται και στην πράξη στο μητρώο μαθητών του σχολείου. Κατά την εγγραφή του στο βιβλίο του μητρώου μαθητών, κάθε μαθητής αποκτά έναν μοναδικό Αριθμό Μητρώου. Στην Βάση Δεδομένων Μαθητολόγιο αυτό υλοποιείται με την προσθήκη του πεδίου AM στον πίνακα Μαθητής και τον καθορισμό του πεδίου ως κύριου κλειδιού ως εξής:

```
CREATE TABLE Μαθητής (
AM INT,
Επώνυμο VARCHAR(50) NOT NULL,
Όνομα VARCHAR (50) NOT NULL,
Πατρώνυμο VARCHAR (50),
Ημερομηνία_Γέννησης DATE,
Φύλο CHAR(1),
Πόλη VARCHAR(50) DEFAULT 'Αθήνα',
PRIMARY KEY (AM)
)
```

Αυτό σημαίνει ότι στον πίνακα δε μπορούν να εισαχθούν δύο μαθητές με τον ίδιο Αριθμό Μητρώου. Περιορισμοί αναφορικής ακεραιότητας

Οι περιορισμοί αναφορικής ακεραιότητας καθορίζουν συσχετίσεις μεταξύ των πεδίων διαφορετικών πινάκων, δηλαδή τα ξένα κλειδιά. Ο περιορισμός ξένου κλειδιού στην γλώσσα SQL καθορίζει ότι το πεδίο ενός πίνακα B αναφέρεται σε ένα πεδίο ενός άλλου πίνακα A, επομένως οι τιμές του πεδίου στον πίνακα B πρέπει να υπάρχουν στο αντίστοιχο πεδίο στον πίνακα A.

Οι περιορισμοί αναφορικής ακεραιότητας περιγράφονται στη γλώσσα SQL με την έκφραση FOREIGN KEY (<πεδίο>) REFERENCES <πίνακας αναφοράς> (<πεδίο πίνακα αναφοράς>) .

Στο παράδειγμά μας του Μαθητολογίου, οι μαθητές του σχολείου εγγράφονται κάθε χρόνο που φοιτούν σε συγκεκριμένη τάξη (Α, Β, ή Γ). Κάθε μαθητής που θα φοιτήσει παραπάνω από ένα έτος στο ίδιο σχολείο θα εγγραφεί τουλάχιστον μία φορά σε κάθε τάξη που θα φοιτήσει. Για να καταγραφεί αυτή η πληροφορία στη σχεσιακή Βάση Δεδομένων, θα πρέπει να δημιουργηθεί ένας νέος πίνακας ο οποίος να επιτρέπει την εγγραφή μαθητών σε σχολικά έτη και τάξεις. Σε αυτόν τον πίνακα που μπορούμε να τον ονομάσουμε ΕγγραφήΜαθητή, αυτό που πρέπει να δημιουργηθεί είναι ένα πεδίο που να ταυτοποιεί κάθε μαθητή. Το πεδίο αυτό είναι ο αριθμός μητρώου (AM) του μαθητή. Ο AM μαθητή πρέπει να συνοδεύεται και από το σχολικό έτος και την τάξη στην οποία εγγράφεται ο μαθητής. Άρα ο πίνακας ΕγγραφήΜαθητή θα έχει την εξής μορφή:

AM_Μαθητή	Σχολ_έτος	Τάξη

Εικόνα 4-5. Ο σχεσιακός πίνακας «Εγγραφή Μαθητή» (AM_Μαθητή, Σχολ_έτος, Τάξη)

Το πεδίο AM_Μαθητή αναφέρεται σε μαθητές που είναι ήδη καταχωρημένοι στον πίνακα Μαθητής. Για να εξασφαλιστεί αυτή η αναφορική ακεραιότητα, το πεδίο AM_Μαθητή πρέπει να είναι ξένο κλειδί που αναφέρεται στο πεδίο AM του πίνακα Μαθητής.

Ένα επιπλέον σημείο που πρέπει να διευκρινιστεί για τον συγκεκριμένο πίνακα, είναι το κύριο κλειδί του, δηλαδή το πεδίο ή τα πεδία που πρέπει να είναι μοναδικά για κάθε εγγραφή. Ας κάνουμε μερικές παρατηρήσεις:

- Ο AM_Μαθητή δε μπορεί να είναι μοναδικός γιατί κάθε μαθητή μπορεί να γίνουν πολλές εγγραφές (στην Α', τη Β' και τη Γ' τάξη)
- Το πεδίο Σχολ_έτος δε μπορεί να είναι μοναδικό γιατί σε κάθε σχολικό έτος εγγράφονται πολλοί μαθητές στο σχολείο.

Επομένως το κύριο κλειδί θα είναι συνδυασμός κάποιων πεδίων. Συγκεκριμένα, θα είναι ο συνδυασμός των πεδίων (AM_Μαθητή, Σχολ_έτος) ώστε να εξασφαλίζεται ότι κάθε μαθητής θα εγγράφεται μόνο μία φορά ανά σχολικό έτος σε κάποια τάξη.

Άρα η δημιουργία του πίνακα μπορεί να γίνει με την εξής εντολή SQL:

```
CREATE TABLE ΕγγραφήΜαθητή  
(  
  AM_Μαθητή INT,  
  Σχολ_έτος CHAR(9),  
  Τάξη CHAR(1),  
  PRIMARY KEY (AM_Μαθητή, Σχολ_έτος),  
  FOREIGN KEY (AM_Μαθητή) REFERENCES Μαθητής(AM)  
)
```

Παρατηρούμε ότι ο τύπος του πεδίου AM_Μαθητή είναι INT, ίδιος με τον τύπο του πεδίου AM στον πίνακα Μαθητής. Αυτή είναι απαραίτητη προϋπόθεση για την δημιουργία του περιορισμού ξένου κλειδιού (FOREIGN KEY) στον πίνακα ΕγγραφήΜαθητή.

Στο ίδιο παράδειγμα, ένας μαθητής παίρνει γραπτό βαθμό στα μαθήματα της τάξης που ένας μαθητής παίρνει έναν γραπτό βαθμό στα μαθήματα της τάξης που παρακολουθεί για κάθε σχολικό έτος που φοιτά στο σχολείο. Άρα υπάρχει συσχέτιση ανάμεσα στον μαθητή και τα μαθήματα για την καταχώρηση της γραπτής βαθμολογίας. Η συσχέτιση αυτή πρέπει να υλοποιηθεί στη σχεσιακή Βάση Δεδομένων με ένα νέο πίνακα ο οποίος θα συσχετίζει τον Αριθμό Μητρώου του μαθητή με το μάθημα στο οποίο καταχωρείται η βαθμολογία του, συμπληρωμένο με το σχολικό έτος. Σε αυτόν τον πίνακα, το κύριο κλειδί, δηλαδή το πεδίο που πρέπει να είναι μοναδικό για κάθε εγγραφή στον πίνακα είναι ο συνδυασμός των πεδίων Αριθμός Μητρώου Μαθητή, Σχολικό Έτος και Όνομα Μαθήματος, έτσι ώστε:

- να είναι δυνατό να εισαχθούν βαθμοί για όλα τα (διαφορετικά) μαθήματα κάποιου μαθητή, να μην είναι δυνατόν να εισαχθεί παραπάνω από ένας βαθμός ανά μαθητή, μάθημα και σχολικό έτος,
- να είναι δυνατόν να εισαχθούν οι βαθμοί όλων των μαθητών σε κάθε μάθημα που παρακολουθούν,
- να είναι δυνατόν να εισαχθούν βαθμοί κάποιου μαθητή στο ίδιο μάθημα για διαφορετικά σχολικά έτη σε περίπτωση επανάληψης της ίδιας τάξης (κάτι που δεν είναι επιθυμητό φυσικά αλλά πρέπει να προβλεφθεί στο ΣΔΒΔ).

Επομένως η δημιουργία του πίνακα για τις γραπτές βαθμολογίες θα έχει ως εξής:

```
CREATE TABLE ΓραπτόςΒαθμόςΜαθητή
(
  ΑΜ_Μαθητή INT,
  Σχολ_έτος CHAR(9),
  Όνομα_Μαθήματος VARCHAR(100),
  Βαθμός FLOAT,
  PRIMARY KEY (ΑΜ_Μαθητή, Σχολ_έτος, Όνομα_Μαθήματος),
  FOREIGN KEY (ΑΜ_Μαθητή) REFERENCES Μαθητής(ΑΜ),
  FOREIGN KEY (Όνομα_Μαθήματος) REFERENCES Μάθημα(Όνομα)
)
```

Παρατηρούμε ότι ο τύπος του πεδίου ΑΜ_Μαθητή είναι INT, ίδιος με τον τύπο του πεδίου ΑΜ στον πίνακα Μαθητής και ο τύπος του πεδίου Όνομα_Μαθήματος είναι VARCHAR(100), ίδιος με τον τύπο του πεδίου Όνομα στον πίνακα Μάθημα.

Στο **Παράρτημα 4.2: Δημιουργία Πινάκων στο Σχεσιακό ΣΔΒΔ MySQL** παρουσιάζεται συνοπτικά η δημιουργία πινάκων στο περιβάλλον του ΣΔΒΔ MySQL με τη χρήση του εργαλείου MySQL Workbench.



Ερωτήσεις / Δραστηριότητες

1) Ανατρέξτε στην Άσκηση 1 της προηγούμενης ενότητας, με τα παρακάτω δεδομένα που αφορούν σε βιβλία:

Ιλιάδα	Όμηρος	2003	18,37
Ερωτόκριτος	Κορνάρος, Βιτσέντζος	1985	12,75

- Περιγράψτε πιθανούς περιορισμούς που να αφορούν στα παραπάνω δεδομένα.
- Εντοπίστε το κύριο κλειδί του πίνακα ή αν είναι απαραίτητο προτείνεται την προσθήκη κατάλληλου κύριου κλειδιού στον πίνακα.
- Γράψτε και εκτελέστε την εντολή της SQL για την δημιουργία του πίνακα.

4.2 SQL Σχεσιακή Γλώσσα Χειρισμού Δεδομένων (DML)



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει τα χαρακτηριστικά της Γλώσσας Χειρισμού Δεδομένων (DML)
- να διακρίνει τη σημασία των Γλωσσών Χειρισμού Δεδομένων στην οργάνωση και διαχείριση των Βάσεων Δεδομένων

Μία Γλώσσα Χειρισμού Δεδομένων (Data Manipulation Language DML) χρησιμοποιείται από τους σχεδιαστές προγραμματιστές και από τους διαχειριστές Βάσεων Δεδομένων για τον χειρισμό των δεδομένων, δηλαδή για να εισαχθούν, να ενημερωθούν, να διαγραφούν και να ανακτηθούν τα δεδομένα της βάσης.

Τα σχεσιακά ΣΔΒΔ χρησιμοποιούν κατά κανόνα τη γλώσσα SQL για τον χειρισμό των δεδομένων. Η γλώσσα SQL DML επιτρέπει την περιγραφή εντολών εισαγωγής, ενημέρωσης, διαγραφής και ανάκτησης των δεδομένων των πινάκων.

Στην ενότητα [4.2.1](#) παρουσιάζονται εντολές εισαγωγής δεδομένων που υποστηρίζει η SQL, στην ενότητα [4.2.2](#) παρουσιάζονται εντολές ενημέρωσης δεδομένων, ενώ στην ενότητα [4.2.3](#) παρουσιάζονται εντολές διαγραφής. Οι ενότητες [4.2.4](#), [4.2.5](#) και [4.2.6](#) παρουσιάζουν βασικές ερωτήσεις ανάκτησης.

4.2.1 Εισαγωγή Δεδομένων σε Πίνακα Σχεσιακής ΒΔ με SQL



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να εισάγει δεδομένα σε πίνακες Σχεσιακού ΣΔΒΔ
- να χρησιμοποιεί την εντολή INSERT INTO της SQL για την εισαγωγή δεδομένων
- να γράφει εντολές INSERT INTO για συγκεκριμένα προβλήματα

Η εισαγωγή δεδομένων σε πίνακα σχεσιακής Βάσης Δεδομένων πραγματοποιείται με την εντολή INSERT INTO της SQL. Η σύνταξη της εντολής INSERT INTO στη γενική της μορφή είναι:

```
INSERT INTO <όνομα πίνακα> VALUES (τιμή_1, τιμή_2, τιμή_3, ...)
```

Με την παραπάνω εντολή εισάγεται μια νέα γραμμή στον πίνακα που καθορίζεται στην εντολή. Οι τιμές που εισάγονται στα πεδία του πίνακα καθορίζονται μέσα στην παρένθεση χωριζόμενες με κόμμα. Οι τιμές αυτές αντιστοιχίζονται σε όλα τα πεδία του πίνακα και με τη σειρά που αυτά έχουν καθοριστεί κατά την δημιουργία του πίνακα.

Εισαγωγή εγγραφών καθορίζοντας τιμές για όλα τα πεδία του πίνακα

Για την εισαγωγή των στοιχείων του μαθητή Πηλείδη Αχιλλέα στον πίνακα Μαθητής, γράφουμε σε SQL την παρακάτω εντολή:

```
INSERT INTO Μαθητής VALUES (863, 'Πηλείδης', 'Αχιλλέας',  
'Πηλέας', '19990624', 'Α', 'Αθήνα')
```

Στο παράδειγμα αυτό θυμίζουμε ότι στην εντολή CREATE TABLE του πίνακα Μαθητής καθορίστηκαν τα πεδία με τη σειρά: AM, Επώνυμο, Όνομα, Πατρώνυμο, Ημερομηνία_γέννησης, Φύλο, Πόλη.

Παρατηρούμε ότι η τιμή της ημερομηνίας εισάγεται με τη μορφή 'έτοςμήνασημέρα'.

Μετά την εκτέλεση της παραπάνω εντολής INSERT INTO, ο πίνακας Μαθητής θα έχει την εξής μορφή:

AM	Επώνυμο	Όνομα	Πατρώνυμο	Ημερομηνία_Γέννησης	Φύλο	Πόλη
863	Πηλείδης	Αχιλλέας	Πηλέας	19990624	A	Αθήνα

Εικόνα 4-6. Ο σχεσιακός πίνακας Μαθητής μετά την εισαγωγή δεδομένων με την εντολή INSERT INTO

Εισαγωγή εγγραφών καθορίζοντας τιμές για μερικά από τα πεδία του πίνακα

Η εισαγωγή τιμών για μερικά μόνο από τα πεδία ενός πίνακα, ή για την αλλαγή της σειρά των πεδίων κατά την εισαγωγή τιμών, μπορεί να χρησιμοποιηθεί η εντολή INSERT INTO με τον εξής τρόπο:

```
INSERT INTO <όνομα πίνακα> (Πεδίο_1, Πεδίο_2, Πεδίο_3, ...)  
VALUES (τιμή_1, τιμή_2, τιμή_3, ...)
```

Η τιμή_1 θα εισαχθεί στο Πεδίο_1 της λίστας, η τιμή_2 στο πεδίο_2 κ.ο.κ.

Στο παράδειγμά μας με τον πίνακα «Μαθητής», αν δεν γνωρίζουμε ακόμα την ημερομηνία γέννησης ενός μαθητή αλλά θέλουμε να εισάγουμε τα υπόλοιπα πεδία του, μπορούμε να χρησιμοποιήσουμε την εντολή INSERT INTO ως εξής για να εισάγουμε τα στοιχεία της μαθήτριας Ομήρου Οδύσσειας:

```
INSERT INTO Μαθητής (AM, Επώνυμο, Όνομα, Πατρώνυμο, Φύλο)  
VALUES (864, 'Ομήρου', 'Οδύσσεια', 'Ομηρος', 'Θ')
```

Μετά την εκτέλεση της παραπάνω εντολής INSERT INTO <πίνακας> (<πεδία>) VALUES, ο πίνακας Μαθη-
τής θα έχει την εξής μορφή:

ΑΜ	Επώνυμο	Όνομα	Πατρώνυμο	Ημερομηνία_Γέννησης	Φύλο	Πόλη
863	Πηλείδης	Αχιλλέας	Πηλέας	1999-06-24	Α	Αθήνα
864	Ομήρου	Οδύσσεια	Όμηρος	NULL	Θ	

Εικόνα 4-7. Ο σχεσιακός πίνακας Μαθητής μετά την εισαγωγή δεδομένων με την εντολή INSERT INTO
<πίνακας> (<πεδία>) VALUES

Παρατηρούμε ότι στο πεδίο Ημερομηνία_Γέννησης για τη μαθήτρια Ομήρου Οδύσσεια, η τιμή είναι NULL
εφόσον δεν έχει εισαχθεί ακόμα. Αργότερα μπορούμε να ενημερώσουμε το πεδίο Ημερομηνία_Γέννησης για
την συγκεκριμένη μαθήτρια με την εντολή UPDATE που παρουσιάζεται στην επόμενη ενότητα.

Επίσης παρατηρούμε ότι στο πεδίο Πόλη, παρόλο που δεν εισάγαμε τιμή για την μαθήτρια Ομήρου Οδύσ-
σεια, η τιμή έχει καθοριστεί σε 'Αθήνα', εφόσον κατά τη δημιουργία του πίνακα είχαμε καθορίσει ότι θα εισά-
γεται η συγκεκριμένη προκαθορισμένη τιμή (DEFAULT).



Ερωτήσεις / Δραστηριότητες

- 1) Δημιουργήστε και εκτελέστε τις κατάλληλες SQL εντολές για να εισάγετε τις εξής εγγραφές στον
πίνακα Μαθητής:
865, 'Γεωργίου', 'Γεώργιος', 'Δημήτριος', NULL, 'Α', 'Θεσσαλονίκη'
866, 'Μηνά', 'Μαρία', 'Μίνωας', '1998-07-05', 'Θ', 'Ηράκλειο'
867, 'Χανιωτάκη', 'Αθηνά', 'Μανούσος', '1998-07-19', 'Θ'
868, 'Ρεθυμνιωτάκης', 'Ορέστης', 'Ιούλιος', '1998-07-31', 'Α'
- 2) Γράψτε και δοκιμάσετε να εκτελέσετε εντολή SQL για την εισαγωγή μαθητή με Αριθμό Μητρώου
κάποιον από τους αριθμούς που έχετε ήδη εισάγει. Τι παρατηρείτε; Είναι εφικτή η εισαγωγή;
Αιτιολογήστε την απάντησή σας.
- 3) Συλλέξτε δεδομένα μαθητών της τάξης σας ζητώντας τη βοήθεια του/της καθηγητή/ τριάς σας
για την αναζήτηση των Αριθμών Μητρώου μαθητολογίου που διατηρεί το σχολείο.
- 4) Δημιουργήστε και εκτελέστε τις εντολές SQL για την εισαγωγή των δεδομένων που συλλέξατε
στην προηγούμενη άσκηση.
- 5) Ανατρέξτε στην Άσκηση 1 της προηγούμενης ενότητας, με τα παρακάτω δεδομένα που αφορούν
σε βιβλία:

Ιλιάδα	Όμηρος	2003	18,37
Ερωτόκριτος	Κορνάρος, Βιτσέντζος	1985	12,75

Γράψτε και εκτελέστε τις εντολές της SQL για την εισαγωγή των παραπάνω δεδομένων στον
πίνακα που δημιουργήσατε στα πλαίσια της προηγούμενης ενότητας. Σημείωση: Εισάγετε δεδομένα
και για το κύριο κλειδί που εντοπίσατε και δημιουργήσατε στην προηγούμενη ενότητα.

4.2.2 Ενημέρωση Εγγραφών Πίνακα Σχεσιακής ΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να γράφει και να εκτελεί εντολές UPDATE για την ενημέρωση εγγραφών σε Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων

Η ενημέρωση των εγγραφών που έχουν αποθηκευτεί σε πίνακα Σχεσιακής Βάσης Δεδομένων πραγματοποιείται στη γλώσσα SQL με την εντολή **UPDATE**. Στην εντολή αυτή καθορίζεται ο πίνακας στον οποίον θα γίνουν ενημερώσεις καθώς και ένα σύνολο προτάσεων που περιγράφουν τις ενημερώσεις των πεδίων του πίνακα. Η γενική μορφή της εντολής UPDATE είναι η εξής:

```
UPDATE <πίνακας>  
SET <προτάσεις ενημέρωσης πεδίων του πίνακα>
```

Έστω ότι στο παράδειγμά μας (Μαθητολόγιο) και συγκεκριμένα στον πίνακα Μάθημα θέλουμε να ενημερώσουμε την τιμή του πεδίου Ώρες έτσι ώστε όλα τα καταχωρημένα μαθήματα να έχουν 3 ώρες διδασκαλίας. Η ενημέρωση αυτή γίνεται με την εντολή UPDATE ως εξής:

```
UPDATE Μάθημα  
SET Ώρες = 3
```

Στην παρακάτω εικόνα, βλέπουμε τα δεδομένα του πίνακα Μάθημα μετά την εκτέλεση της εντολής UPDATE για την ενημέρωση των ωρών των μαθημάτων:

Όνομα	Ώρες
Συστήματα Διαχείρισης Βάσεων Δεδομένων και εφαρμογές τους στο Web	3

Εικόνα 4-8. Ο σχεσιακός πίνακας Μάθημα μετά την εκτέλεση της εντολής UPDATE

Η εντολή UPDATE μπορεί να χρησιμοποιηθεί και ως ενημέρωση με αναζήτηση των εγγραφών που θα ενημερωθούν, δηλαδή ενημέρωση υπό συνθήκη. Σε αυτή την περίπτωση, μπορούμε να χρησιμοποιήσουμε την έκφραση WHERE σε γενική μορφή ως εξής:

```
UPDATE <πίνακας>  
SET <προτάσεις ενημέρωσης πεδίων του πίνακα>  
WHERE <συνθήκη αναζήτησης>
```

Στο παράδειγμα του πίνακα «Μάθημα», η ενημέρωση όλων των μαθημάτων που έχουν καταχωρηθεί με 5 ώρες διδασκαλίας ώστε να γίνει αλλαγή των ωρών σε 3 γίνεται ως εξής:

```
UPDATE Μάθημα  
SET Ώρες = 3  
WHERE Ώρες = 5
```

Με αυτή την εντολή δε θα ενημερωθούν όλες οι εγγραφές μαθημάτων, αλλά μόνο αυτές που ικανοποιούν τη συνθήκη που υπάρχει μετά τη λέξη WHERE.



Ερωτήσεις / Δραστηριότητες

- 1) Ανατρέξτε στην ενότητα 4.2.1 όπου έγινε εισαγωγή στον πίνακα Μαθητής της μαθήτριας Οδύσσειας Ομήρου χωρίς να καθοριστεί η ημερομηνία γέννησής της.
 - α) Ποια συνθήκη πρέπει να καθοριστεί για να ενημερωθεί η ημερομηνία γέννησης μόνο της συγκεκριμένης μαθήτριας;
 - β) Γράψτε και εκτελέστε την εντολή SQL που θα ενημερώσει το πεδίο Ημερομηνία_Γέννησης για την συγκεκριμένη μαθήτρια.
- 2) Γράψτε και εκτελέστε τις εντολές SQL για να ενημερώστε τα στοιχεία των μαθητών του πίνακα Μαθητής ως εξής:
 - α) Η μαθήτρια Χανιωτάκη Αθηνά έχει πόλη καταγωγής τα Χανιά.
 - β) Ο μαθητής Ρεθυμνιωτάκης Ορέστης έχει πόλη καταγωγής το Ρέθυμνο.

Σημείωση: μπορείτε να χρησιμοποιήσετε είτε το κύριο κλειδί στη συνθήκη για την ενημέρωση των παραπάνω εγγραφών, είτε το ονοματεπώνυμό τους. Ποια είναι η διαφορά; Σε ποια περίπτωση εξασφαλίζεται η ορθότητα της ενημέρωσης;

4.2.3 Διαγραφή Εγγραφών Πίνακα Σχεσιακής ΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να γράφει και να εκτελεί εντολές DELETE για την διαγραφή εγγραφών σε Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων

Η διαγραφή εγγραφών από πίνακα Σχεσιακής Βάσης Δεδομένων πραγματοποιείται με την εντολή **DELETE** της SQL. Η εντολή DELETE συντάσσεται στην απλούστερη μορφή της ως εξής:

```
DELETE FROM <πίνακας>
```

Αν για παράδειγμα θέλουμε να διαγράψουμε όλα τα μαθήματα που είναι καταχωρημένα στον πίνακα Μάθημα, γράφουμε και εκτελούμε την εντολή:

```
DELETE FROM Μάθημα
```

Με τον παραπάνω τρόπο διαγράφονται όλες οι γραμμές του πίνακα Μάθημα, οπότε πρέπει να είμαστε προσεκτικοί στη χρήση της εντολής DELETE.

Συχνά χρειάζεται να διαγράψουμε μερικές μόνο εγγραφές οι οποίες ικανοποιούν κάποια συνθήκη και πρέπει να αναζητηθούν από το σύστημα μέσα στον πίνακα ώστε να διαγραφούν. Σε αυτήν την περίπτωση θα χρησιμοποιήσουμε την εντολή DELETE μαζί με συνθήκη, δηλαδή μαζί με πρόταση WHERE, ως εξής:

```
DELETE FROM <πίνακας>  
WHERE <συνθήκη αναζήτησης>
```

Αν στο παράδειγμα του Μαθητολογίου επιθυμούμε να διαγράψουμε τους γραπτούς βαθμούς του σχολικού έτους 20142015, τότε μπορούμε να γράψουμε την εξής εντολή σε SQL:

```
DELETE FROM ΓραπτόςΒαθμόςΜαθητή  
WHERE Σχολ_έτος = '20142015'
```

Η χρήση της εντολής DELETE απαιτεί προσοχή στην περίπτωση όπου έχουμε περιορισμούς αναφοράς, δηλαδή ξένα κλειδιά. Τι θα συμβεί για παράδειγμα αν προσπαθήσουμε να διαγράψουμε έναν μαθητή για τον οποίον έχουν ήδη καταχωρηθεί γραπτοί βαθμοί; Τα στοιχεία του μαθητή είναι αποθηκευμένα στον πίνακα Μαθητής, ενώ οι βαθμοί του είναι καταχωρημένοι στον πίνακα ΓραπτόςΒαθμόςΜαθητή. Οι βαθμοί έχουν καταχωρηθεί με τον Αριθμό Μητρώου (ΑΜ) του μαθητή, ο οποίος αναφέρεται στον πίνακα Μαθητής. Τι θα συμβεί λοιπόν αν προσπαθήσουμε να διαγράψουμε τον μαθητή από τον πίνακα Μαθητής; Θα πρέπει είτε να διαγραφούν και όλοι οι βαθμοί του στον πίνακα ΓραπτόςΒαθμόςΜαθητή εφόσον θα αφορούν πλέον έναν μαθητή που δεν υπάρχει, είτε δε θα πρέπει να επιτραπεί η διαγραφή του μαθητή από το σύστημα ώστε να διατηρηθεί η αναφορική ακεραιότητα των δεδομένων.



Ερωτήσεις / Δραστηριότητες

- 1) Διαγράψτε όλες τις εγγραφές του πίνακα Μάθημα εκτελώντας την κατάλληλη εντολή σε SQL.
- 2) Ποια συνθήκη αναζήτησης θα καθορίζατε για να διαγράψετε την εγγραφή του μαθητή Πηλείδη Αχιλλέα από τον πίνακα Μαθητής;
- 3) Διαγράψτε την εγγραφή του μαθητή Πηλείδη Αχιλλέα από τον πίνακα Μαθητής, εκτελώντας την κατάλληλη SQL εντολή.
- 4) Γράψτε και εκτελέστε τις εντολές εισαγωγής εγγραφών για τους μαθητές τα δεδομένα των οποίων συλλέξατε και εισάγατε στον πίνακα ΕγγραφήΜαθητή στα πλαίσια της Άσκησης 4 της ενότητας 4.2.1. Κατόπιν διαγράψτε μία από τις εγγραφές των μαθητών εκτελώντας την κατάλληλη εντολή SQL.

4.2.4 Βασικές Ερωτήσεις Ανάκτησης



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να χρησιμοποιεί την εντολή SELECT ώστε να αναζητά συγκεκριμένες πληροφορίες σε πίνακα
- να χρησιμοποιεί ψευδώνυμα στηλών στην παρουσίαση των αποτελεσμάτων της αναζήτησης
- να δημιουργεί συσχετισμένες στήλες
- να χρησιμοποιεί τη φράση DISTINCT για την απαλοιφή πολλαπλών εμφανίσεων
- να καθορίζει συνθήκες επιλογής εγγραφών στη φράση WHERE
- να χρησιμοποιεί τη φράση FROM στην εντολή SELECT ώστε να αναζητά πληροφορίες από περισσότερους πίνακες
- να χρησιμοποιεί την εντολή SELECT ώστε να αναζητά συγκεκριμένες πληροφορίες σε διαθέσιμες βάσεις δεδομένων για την επίλυση προβλημάτων

Η Εντολή Select

Η γλώσσα ερωτήσεων της SQL βασίζεται σε πράξεις συνόλου και σχέσεων. Η εντολή Select, χρησιμοποιείται για την εμφάνιση (ανάκληση) δεδομένων από τη βάση δεδομένων. Μια χαρακτηριστική ερώτηση σε SQL για την αναζήτηση δεδομένων έχει την εξής μορφή/ σύνταξη:

```
SELECT Πεδίο_1, Πεδίο_2, ... Πεδίο_n  
FROM Πίνακας_1, Πίνακας_2, ..., Πίνακας_m  
WHERE συνθήκη;
```

Η φράση **Select** ακολουθείται από μια λίστα πεδίων, τα οποία θα εμφανίζονται στο αποτέλεσμα. Τα πεδία (στήλες) χωρίζονται μεταξύ τους με κόμματα.

Η φράση **From** δηλώνει τις σχέσεις (πίνακες) στους οποίους θα γίνει η αναζήτηση, ορίζονται δηλαδή ο/οι πίνακες/ες της ΒΔ από όπου θα ανακτηθούν οι πληροφορίες.

Η φράση **Where** (δεν είναι υποχρεωτική) καθορίζει τη συνθήκη/ ες επιλογής κάποιων εγγραφών.



Select, αντιστοιχεί στην πράξη της προβολής της σχεσιακής άλγεβρας

from, αντιστοιχεί στην πράξη του καρτεσιανού γινομένου της σχεσιακής άλγεβρας

where, αντιστοιχεί στη συνθήκη της πράξης της επιλογής στη σχεσιακή άλγεβρα

Θα εξετάσουμε μέσα από παραδείγματα διάφορες μορφές τις εντολής Select

Απλή μορφή

Επιλογή και παρουσίαση πληροφοριών από συγκεκριμένα πεδία πίνακα

```
SELECT Επώνυμο, Όνομα  
FROM Μαθητής
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής
Επιλογή και παρουσίαση πληροφοριών όλων των πεδίων του πίνακα

```
SELECT *  
FROM Μαθητής
```

Δηλαδή επέλεξε ΟΛΑ τα πεδία και εμφάνισε όλες τις πληροφορίες από τον πίνακα Μαθητής

Με τη φράση Select μπορούμε να συνδυάσουμε ένα σύνολο από χαρακτηριστικά, ώστε να έχουμε επιλογές στην εμφάνιση των αποτελεσμάτων, όπως:

A. Αριθμητικές εκφράσεις

Συνδυασμός μιας ή και περισσοτέρων πράξεων (+ , , * , /), ακολουθώντας την προτεραιότητά τους. Τα πεδία που συμμετέχουν στις αριθμητικές παραστάσεις θα πρέπει να είναι και αυτά αριθμητικά.

```
SELECT Όνομα, Ώρες*3
FROM Μάθημα
```

Δηλαδή επέλεξε τα πεδία Όνομα και τις αντίστοιχες Ώρες πολλαπλασιασμένες με το 3 (που εκφράζει τον αριθμό των τμημάτων), από τον πίνακα Μάθημα

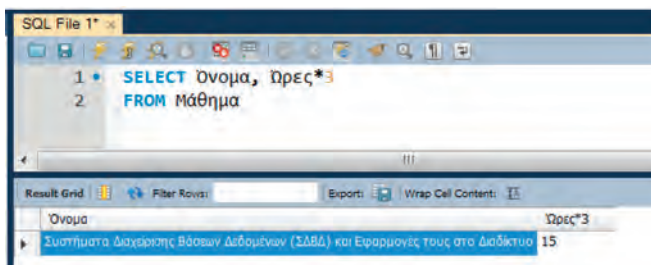
B. Ψευδώνυμα Στηλών

Επιλέγει και παρουσιάζει μια στήλη (πεδίο) με εναλλακτική επικεφαλίδα στην έξοδο. Καθορίζουμε το ψευδώνυμο μετά την στήλη, χρησιμοποιώντας το χαρακτήρα κενό.

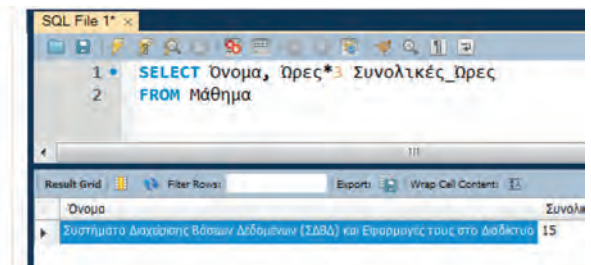
Παρατήρηση: Ψευδώνυμα βάζουμε μόνο στην φράση SELECT.

```
SELECT Όνομα, Ώρες*3 Συνολικές_Ώρες
FROM Μάθημα
```

Δηλαδή επέλεξε τα πεδία Όνομα και τις αντίστοιχες Ώρες πολλαπλασιασμένες με το 3 (που εκφράζει τον αριθμό των τμημάτων), από τον πίνακα Μάθημα με όνομα της δεύτερης στήλης το Συνολικές_Ώρες



Εικόνα 4.2.4.1 Αριθμητική πράξη: πεδίου με αριθμό (Ώρες*3)



Εικόνα 4.2.4.2 Χρησιμοποίηση του Ψευδώνυμου ώστε να χαρακτηρίσουμε την πληροφορία (Συνολικές_Ώρες)

Γ. Συσχετισμένες στήλες

Η πράξη συσχέτισης (concat) επιτρέπει μια στήλη να συνδεθεί με μια άλλη είτε αριθμητική είτε χαρακτήρων. Έτσι φτιάχνεται μία μόνο στήλη.

```
SELECT concat(Επώνυμο, Όνομα) Ονοματεπώνυμο
FROM Μαθητής
```

Δηλαδή επέλεξε και εμφάνισε τα δεδομένα στα πεδία Επώνυμο και Όνομα με όνομα στήλης το Ονοματεπώνυμο (ψευδώνυμο), από τον πίνακα Μαθητής. Τα δεδομένα εμφανίζονται χωρίς κενό μεταξύ τους

Δ. Εκφράσεις

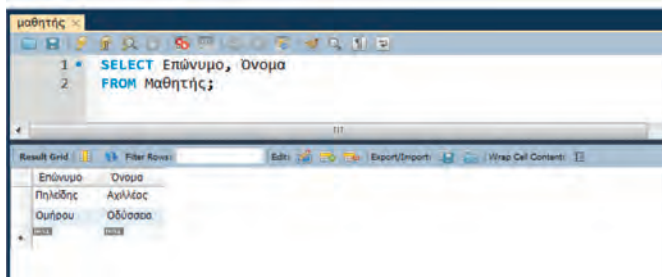
Είναι κάθε χαρακτήρας, έκφραση, αριθμός που μπαίνει στην φράση SELECT και η οποία δεν είναι όνομα στήλης ούτε ψευδώνυμο.

Οι εκφράσεις πρέπει να είναι μέσα σε μονά ``, ενώ για τους αριθμούς δεν χρειάζονται.

```
SELECT concat(Επώνυμο, ' ' Όνομα) Ονοματεπώνυμο
FROM Μαθητής
```

Σε σχέση με το παραπάνω ερώτημα, στην εμφάνιση των αποτελεσμάτων, το Επώνυμο θα χωρίζεται με το

Όνομα με ένα κενό

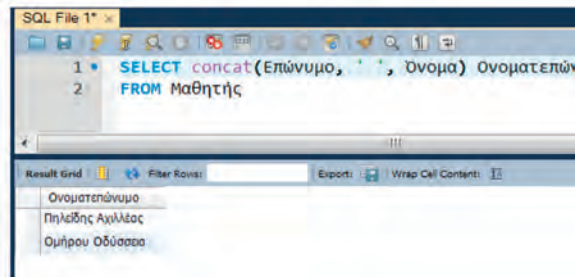


SQL File 1*

```
1 SELECT Επώνυμο, Όνομα
2 FROM Μαθητής;
```

Επώνυμο	Όνομα
Πηλοδης	Αχιλλος
Ομήρου	Οδύσσεια

Εικόνα 4.2.4.3 Εμφάνιση των περιεχομένων των στηλών Επώνυμο & Όνομα



SQL File 1*

```
1 SELECT concat(Επώνυμο, ' ', Όνομα) Ονοματεπώνυμο
2 FROM Μαθητής;
```

Ονοματεπώνυμο
Πηλοδης Αχιλλος
Ομήρου Οδύσσεια

Εικόνα 4.2.4.4 Συσχέτιση των στηλών Επώνυμο & χρησιμοποίηση της έκφρασης 'κενό', ώστε να διαχωρίζει το Ψευδώνυμο (Ονοματεπώνυμο) για να χαρακτηρίσουμε την πληροφορία

Ε. Απαλοιφή πολλαπλών εμφανίσεων –distinct

Για να πραγματοποιηθεί η διαγραφή των διπλοτύπων, εισάγεται τη λέξη κλειδί distinct μετά τη select Το ερώτημα

```
SELECT AM
FROM ΕγγραφήΜαθητή
```

Δηλαδή θα μας εμφανίζει όλους τους Αριθμούς Μητρώων (AM) των μαθητών μετά από την εγγραφή τους στα διαφορετικά σχολικά έτη, από τον πίνακα ΕγγραφήΜαθητή

Για να απαλείψουμε τις πολλαπλές εμφανίσεις των AM των μαθητών χρησιμοποιούμε τη λέξη κλειδί distinct

```
SELECT DISTINCT AM
FROM ΕγγραφήΜαθητή
```

Δηλαδή επέλεξε και εμφάνισε μόνο τα διαφορετικά AM, από τον πίνακα ΕγγραφήΜαθητή

```
SELECT DISTINCT AM, Τάξη
FROM ΕγγραφήΜαθητή
```

Δηλαδή επέλεξε και εμφάνισε μόνο τους διαφορετικούς συνδυασμούς AM και Τάξη από τον πίνακα ΕγγραφήΜαθητή

Η φράση WHERE

Με τη φράση Where καθορίζουμε συνθήκες επιλογής κάποιων εγγραφών του πίνακα με βάση τιμές των πεδίων.

Οι συνθήκες καθορίζονται από τη σύγκριση τιμών πεδίων του πίνακα και μπορεί να είναι, τιμές αριθμητικές, αλφαβητικές, αριθμητικές εκφράσεις και συναρτήσεις.

Μια συνθήκη προσδιορίζεται από:



Το **Όνομα πεδίου**,
τον **Τελεστή σύγκρισης** και
το **Όνομα πεδίου** ή τη **σταθερή τιμή** ή τη **λίστα τιμών**

Οι τελεστές καθορίζουν την σχέση μεταξύ των τιμών που συγκρίνονται και είναι:

ΤΕΛΕΣΤΗΣ	ΛΕΙΤΟΥΡΓΙΑ
=	ίσο με
<>	διάφορο του
>	μεγαλύτερο από
<	μικρότερο από
>=	μεγαλύτερο ή ίσο από
<=	μικρότερο ή ίσο από
BETWEEN...AND...	μεταξύ δύο τιμών
IN (LIST)	μία από τις τιμές της λίστας
LIKE	σύγκριση με ακολουθία χαρακτήρων
IS NULL	κενό πεδίο (ανύπαρκτη τιμή)

Στις ακολουθίες χαρακτήρων μπορεί να γίνεται χρήση ειδικών χαρακτήρων που παίρνουν τη θέση οποιονδήποτε άλλων. Για τον τελεστή LIKE, αυτοί είναι:

	ΤΕΛΕΣΤΗΣ LIKE
% (ποσοστό)	οποιοδήποτε χαρακτήρες (0, 1, ή πολλοί)
- (κάτω παύλα)	οποιοσδήποτε ένας χαρακτήρας

Λογικοί τελεστές

Μπορούμε να χρησιμοποιήσουμε περισσότερες από μια συνθήκες με τη χρήση των λογικών τελεστών **AND**, **OR** και **NOT**. Στις δημιουργημένες συνθήκες η προτεραιότητα ακολουθεί τη σειρά: NOT, AND, OR



Στις συγκρίσεις γίνεται διάκριση ανάμεσα σε κεφαλαία και πεζά.

ΠΑΡΑΔΕΙΓΜΑΤΑ:

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα='Γιώργος'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή είναι 'Γιώργος'

```
SELECT Όνομα, Ώρες
FROM Μάθημα
WHERE Ώρες >=4
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα και Ώρες από τον πίνακα Μάθημα, όταν οι ώρες του μαθήματος είναι μεγαλύτερες ή ίσες με 4

```
SELECT Όνομα, Ώρες
FROM Μάθημα
WHERE Ώρες <>4
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα και Ώρες από τον πίνακα Μάθημα, όταν οι ώρες του μαθήματος είναι διαφορετικές από 4

```
SELECT Όνομα, Ώρες
FROM Μάθημα
WHERE Ώρες BETWEEN 4 AND 5
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα και Ώρες από τον πίνακα Μάθημα, όταν οι ώρες του μαθήματος είναι μεταξύ 4 και 5 ωρών, δηλαδή μεγαλύτερες ή ίσες με 4 και μικρότερες ή ίσες με 5

```
SELECT Όνομα, Ώρες
FROM Μάθημα
WHERE Ώρες IN (4, 5)
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα και Ώρες από τον πίνακα Μάθημα, όταν οι ώρες του μαθήματος είναι 4 ή 5, δηλαδή ικανοποιούν τις τιμές που περιλαμβάνονται στην παρένθεση

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα LIKE "Μ%"
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή αρχίζει από "Μ"

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα LIKE "Μ_____"
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή αρχίζει από "Μ" και περιλαμβάνει 5 χαρακτήρες

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Ημερομηνία_Γέννησης IS NULL
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν η Ημερομηνία_Γέννησης δεν είναι συμπληρωμένη

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα NOT LIKE "Μ%"
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή δεν αρχίζει από "Μ"

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Ημερομηνία_Γέννησης IS NOT NULL
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν η Ημερομηνία_Γέννησης είναι συμπληρωμένη

Σημείωση: Ποτέ δεν γίνεται σύγκριση με Null. Πάντα χρησιμοποιούμε τις φράσεις is Null ή is not Null.

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα LIKE "Μ%" AND Πόλη='Χανιά'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή αρχίζει από "Μ" και είναι από τα Χανιά

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα LIKE "Μ%" AND Πόλη='Χανιά' OR Πόλη='Ρέθυμνο'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή αρχίζει από "Μ" και είναι από τα Χανιά ή το Ρέθυμνο ανεξαρτήτως Ονόματος μαθητή

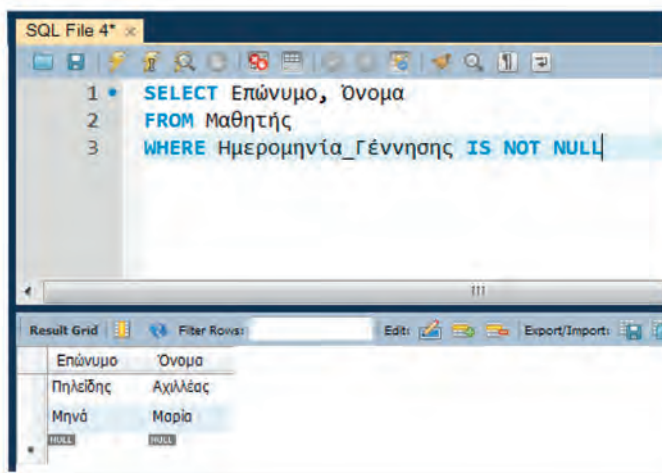
```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα LIKE "Μ%" AND (Πόλη='Χανιά' OR Πόλη='Ρέθυμνο')
```

ή

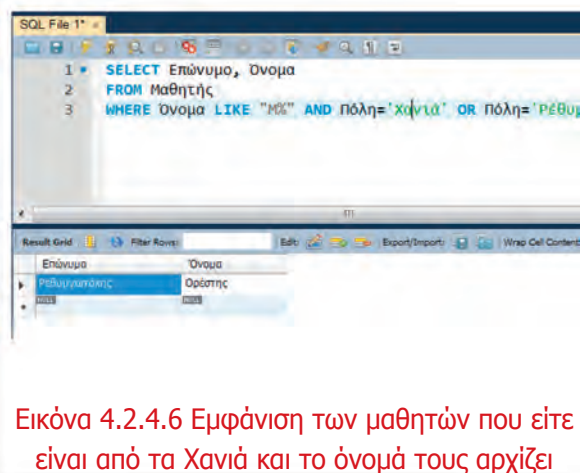
```
SELECT Επώνυμο, Όνομα
FROM Μαθητής
WHERE Όνομα LIKE "Μ%" OR Πόλη='Χανιά' AND Πόλη='Ρέθυμνο'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής, όταν το όνομα του μαθητή αρχίζει από "Μ" και είναι από τα Χανιά ή το Ρέθυμνο

Σημείωση: Ισχύει και στις λογικές πράξεις η προτεραιότητα των πράξεων (Not, And, Or)



Εικόνα 4.2.4.5 Εμφάνιση των μαθητών που έχει συμπληρωθεί η Ημερομηνία γέννησης



Εικόνα 4.2.4.6 Εμφάνιση των μαθητών που είτε είναι από τα Χανιά και το όνομά τους αρχίζει από Μ, είτε είναι από το Ρέθυμνο (ο Μαθητής είναι από το Ρέθυμνο)

Δεδομένα από περισσότερους από ένα πίνακα

Η φράση FROM

Οι εγγραφές ενός πίνακα μπορούν να συνδυαστούν με τις εγγραφές ενός άλλου πίνακα, εφόσον υπάρχουν κοινά πεδία στους δύο πίνακες. Ο συνδυασμός γίνεται βάσει των τιμών του κοινού πεδίου. Στη φράση from ορίζονται οι πίνακες.

Για να μπορέσουμε να εμφανίζουμε τις πληροφορίες των πινάκων, στη φράση where θα πρέπει να συνδέσουμε τα κοινά πεδία. Η αναφορά μας σε ένα πεδίο ενός πίνακα γίνεται με το συμβολισμό:

<όνομα-πίνακα>.<όνομα-πεδίου>

Η πρόταση from αντιστοιχεί στην πράξη του καρτεσιανού γινομένου της σχεσιακής άλγεβρας. Δίνει τη λίστα

των σχέσεων που πρέπει να σαρωθούν κατά την αξιολόγηση της έκφρασης.

Όταν το ίδιο γνώρισμα εμφανίζεται στο σχήμα περισσότερων από μια σχέσεων, τότε γίνεται διάκριση βάσει του συμβολισμού:

<όνομα-σχέσης>.<όνομα-γνωρίσματος>

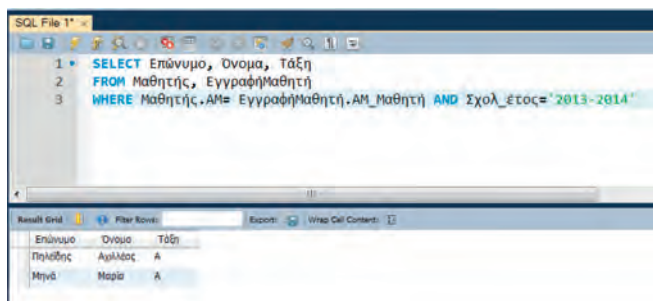
ΠΑΡΑΔΕΙΓΜΑΤΑ

```
SELECT Επώνυμο, Όνομα, Τάξη
FROM Μαθητής, ΕγγραφήΜαθητή
WHERE Μαθητής.AM= ΕγγραφήΜαθητή.AM_Μαθητή
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα και την Τάξη από τους πίνακες Μαθητής και ΕγγραφήΜαθητή, για όλους τους μαθητές. Στο αποτέλεσμα, εμφανίζονται όλοι οι δυνατοί συνδυασμοί των δύο πινάκων που έχουν κοινό πεδίο τον AM

```
SELECT Επώνυμο, Όνομα, Τάξη
FROM Μαθητής, ΕγγραφήΜαθητή
WHERE Μαθητής.AM= ΕγγραφήΜαθητή.AM_Μαθητή AND Σχολ_έτος='2013-2014'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα και την Τάξη από τους πίνακες Μαθητής και ΕγγραφήΜαθητή, για τους μαθητές στο σχολικό έτος 2013-2014. Στο αποτέλεσμα, περιορίζονται οι συνδυασμοί των δύο πινάκων στο σχολικό έτος 2013-2014



Εικόνα 4.2.4.7 Γίνεται αρχικά το join condition (Μαθητής.AM= ΕγγραφήΜαθητή.AM_Μαθητή) και στη συνέχεια η εμφάνιση των μαθητών το σχολικό έτος 2013-2014

Υποερωτήματα

Τα υποερωτήματα είναι ερωτήματα που περιέχονται στη φράση WHERE άλλων ερωτημάτων. Τα αποτελέσματα των υποερωτημάτων δεν εμφανίζονται αλλά απλά δρουν βοηθητικά, ώστε να συνδεθούν τα στοιχεία των πινάκων.

```
SELECT Επώνυμο, Όνομα
FROM Μαθητής, ΕγγραφήΜαθητή
WHERE Μαθητής.AM= ΕγγραφήΜαθητή.AM_Μαθητή AND AM IN (SELECT AM
FROM ΕγγραφήΜαθητή
WHERE Σχολ_έτος='2013-2014')
```

Παρατηρούμε ότι έχουμε το ίδιο αποτέλεσμα με το προηγούμενο ερώτημα, δηλαδή επέλεξε και εμφάνισε τις πληροφορίες για το Επώνυμο, Όνομα και την Τάξη από τους πίνακες Μαθητής και ΕγγραφήΜαθητή, για τους μαθητές στο σχολικό έτος 2013-2014. Στο αποτέλεσμα, περιορίζονται οι συνδυασμοί των δύο πινάκων στο σχολικό έτος 2013-2014)



Ερωτήσεις / Δραστηριότητες

1. Προσδιορίστε και αναπτύξτε τις αντίστοιχες εντολές select, ώστε να απαντηθούν τα ερωτήματα:
 - Εμφανίστε για όλους τους μαθητές τις πληροφορίες Επώνυμο, Όνομα, Πατρώνυμο, Ημερομηνία_Γέννησης
 - Εμφανίστε για τα αγόρια μαθητές τις πληροφορίες Επώνυμο, Όνομα, Πατρώνυμο, Ημερομηνία_Γέννησης
 - Εμφάνισε τους διαφορετικούς μαθητές της Β τάξης
 - Ποια είναι τα μαθήματα της Α τάξης;
 - Ποιος ο γραπτός βαθμός του μαθητή Μαρινάκη στο μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές στο Διαδίκτυο το σχολικό έτος 2014-2015;
 - Ποιοι μαθητές δεν έχουν βαθμό στο μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές στο Διαδίκτυο το σχολικό έτος 2014-2015;

4.2.5 Ταξινόμηση Αποτελεσμάτων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να χρησιμοποιεί τη φράση `order by` ώστε να ταξινομεί τις πληροφορίες στην αναζήτηση
- να επιλέγει αύξουσα ή φθίνουσα ταξινόμηση των εγγραφών

Διάταξη των εγγραφών – ORDER BY

Στην SQL χρησιμοποιείται η φράση `order by` ώστε οι εγγραφές στο αποτέλεσμα να είναι ταξινομημένες (π.χ. με αλφαβητική σειρά) με βάση το αντίστοιχο πεδίο.

```
SELECT Επώνυμο, Όνομα  
FROM Μαθητής  
ORDER BY Επώνυμο
```

Δηλαδή επέλεξε και εμφάνισε ταξινομημένες σε αύξουσα σειρά, βάση του Επωνύμου τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής

Η προεπιλογή είναι η αύξουσα διάταξη, αλλά μπορούμε να επιλέξουμε διάταξη χρησιμοποιώντας το `asc` (αύξουσα) ή το `desc` (φθίνουσα). Επίσης, μπορούμε να κάνουμε ταξινόμηση χρησιμοποιώντας πολλά πεδία.

```
SELECT Επώνυμο, Όνομα  
FROM Μαθητής  
ORDER BY Επώνυμο DESC, Όνομα ASC
```

Δηλαδή επέλεξε και εμφάνισε ταξινομημένες σε φθίνουσα σειρά, βάση του Επωνύμου και στη συνέχεια στις εγγραφές με κοινό Επώνυμο σε αύξουσα βάση του Ονόματος, τις πληροφορίες για το Επώνυμο, Όνομα από τον πίνακα Μαθητής.

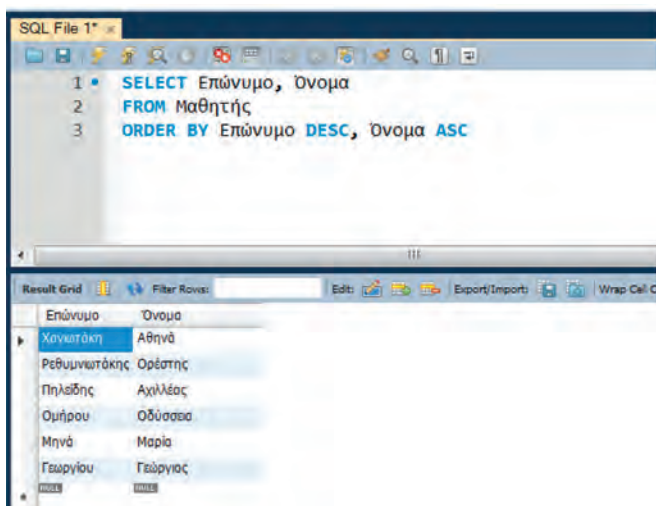
Ταξινόμηση εγγραφών μπορούμε να έχουμε και όταν η επιλογή των πεδίων είναι από περισσότερους από έναν πίνακες.

```
SELECT Επώνυμο, Όνομα, Τάξη  
FROM Μαθητής, ΕγγραφήΜαθητή  
WHERE Μαθητής.AM= ΕγγραφήΜαθητή.AM_Μαθητή  
ORDER BY Τάξη
```

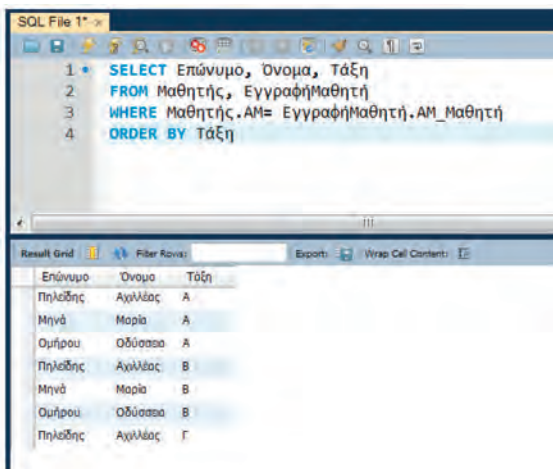
Δηλαδή επέλεξε και εμφάνισε ταξινομημένες σε αύξουσα σειρά τις πληροφορίες για το Επώνυμο, Όνομα και την Τάξη από τους πίνακες Μαθητής και ΕγγραφήΜαθητή, για όλους τους μαθητές. Στο αποτέλεσμα, εμφανίζονται όλοι οι δυνατοί συνδυασμοί των δύο πινάκων που έχουν κοινό πεδίο τον AM, ταξινομημένοι ανά τάξη.

Σημείωση

Η ταξινόμηση μπορεί να γίνει και με πεδία που δεν εμφανίζονται στο `Select`, ενώ δεν επιτρέπεται η ταξινόμηση με βάση τις πράξεις μεταξύ πεδίων.



Εικόνα 4.2.5.1 Ταξινόμηση με δύο πεδία, Φθίνουσα στο Επώνυμο και Αύξουσα στο Όνομα



Εικόνα 4.2.5.2 Ταξινόμηση σε Αύξουσα σειρά της Τάξης δύο πίνακες



Ερωτήσεις / Δραστηριότητες

1. Προσδιορίστε και αναπτύξτε τις αντίστοιχες εντολές select, ώστε να απαντηθούν τα ερωτήματα:
 - Εμφανίστε ταξινομημένα σε αύξουσα σειρά Επωνύμου και Ονόματος, τα στοιχεία των μαθητών που έχουν κάνει εγγραφή στην Α τάξη το σχολικό έτος 20152016
 - Εμφανίστε ταξινομημένα σε αύξουσα σειρά τα μαθήματα. Μπορεί να γίνει ταξινόμηση των μαθημάτων ανά τάξη;
 - Εμφανίστε σε φθίνουσα σειρά του γραπτού βαθμού τους μαθητές της Α τάξης που έγραψαν στο μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές στο Διαδίκτυο το σχολικό έτος 2014-2015, βαθμό >10.

4.2.6 Συναρτήσεις Συνάθροισης και Ομαδοποίησης



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να χρησιμοποιεί τις βασικές συναθροιστικές συναρτήσεις στην παρουσίαση των δεδομένων αναζήτησης
- να χρησιμοποιεί τη φράση group by για ομαδοποίηση των εγγραφών στα δεδομένα της αναζήτησης

Συναθροιστικές Συναρτήσεις

Η SQL έχει 5 ενσωματωμένες συναθροιστικές συναρτήσεις. Οι ακόλουθες συναρτήσεις εφαρμόζονται στο σύνολο των τιμών μιας στήλης ενός πίνακα, και επιστρέφουν μία τιμή: Αυτές επιδρούν σε ομάδα εγγραφών (γραμμών) του πίνακα και είναι:

Συναρτήσεις	Ερμηνεία
AVG (A)	Μέσος όρος τιμών (μόνο σε αριθμούς)
COUNT(A)	Πλήθος τιμών
MAX (A)	Μέγιστη τιμή
MIN (A)	Ελάχιστη τιμή
SUM (A)	Άθροισμα τιμών (μόνο σε αριθμούς)
(A): αποτελεί τη στήλη	Όλες οι συναρτήσεις αγνοούν τις τιμές NULL εκτός από την COUNT(*).

ΠΑΡΑΔΕΙΓΜΑΤΑ:

```
SELECT Όνομα_Μαθήματος, AVG(Βαθμός)
FROM ΓραπτόςΒαθμόςΜαθητή
WHERE Όνομα_Μαθήματος='Συστήματα Διαχείρισης Βάσεων
Δεδομένων (ΣΔΒΔ) και Εφαρμογές στο Διαδίκτυο'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα_Μαθήματος και το Μέσο Όρο του βαθμού από τον πίνακα ΓραπτόςΒαθμόςΜαθητή, για το μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) και Εφαρμογές στο Διαδίκτυο

```
SELECT Όνομα_Μαθήματος, MIN(Βαθμός)
FROM ΓραπτόςΒαθμόςΜαθητή
WHERE Όνομα_Μαθήματος='Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) και
Εφαρμογές στο Διαδίκτυο'
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα_Μαθήματος και το Μικρότερο βαθμό από τον πίνακα Γραπτός Βαθμός Μαθητή, για την για το μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) και Εφαρμογές στο Διαδίκτυο

```
SELECT Όνομα_Μαθήματος, COUNT(*)
FROM ΓραπτόςΒαθμόςΜαθητή
WHERE Όνομα_Μαθήματος='Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ)
και Εφαρμογές στο Διαδίκτυο' AND Βαθμός<10
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα_Μαθήματος και το Πλήθος των μαθητών από τον

πίνακα ΓραπτόςΒαθμόςΜαθητή, για το μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) και Εφαρμογές στο Διαδίκτυο που ο βαθμός τους είναι μικρότερος από 10

Η φράση GROUP BY

Μπορούμε να εφαρμόσουμε τις συναρτήσεις όχι μόνο σε ένα σύνολο από εγγραφές, αλλά σε ομάδες από σύνολα εγγραφών.

Οι ομάδες προσδιορίζονται χρησιμοποιώντας το group by και οι εγγραφές μιας ομάδας περιέχουν την ίδια τιμή σε κάποιο δεδομένο πεδίο.

Η φράση αυτή μπαίνει ΠΑΝΤΑ στο τέλος του ερωτήματος.

ΠΑΡΑΔΕΙΓΜΑΤΑ

```
SELECT Τάξη, COUNT(*)  
FROM ΕγγραφήΜαθητή  
GROUP BY Τάξη
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Τάξη και το Πλήθος των μαθητών, ανά τάξη από τον πίνακα ΕγγραφήΜαθητή

```
SELECT Τάξη, COUNT(*)  
FROM ΕγγραφήΜαθητή  
WHERE Σχολ_έτος='2014-2015'  
GROUP BY Τάξη
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Τάξη και το Πλήθος των μαθητών, ανά τάξη από τον πίνακα ΕγγραφήΜαθητή, για το Σχολικό έτος είναι 2014-2015

```
SELECT Τάξη, COUNT(*)  
FROM ΕγγραφήΜαθητή  
WHERE Τάξη<>'Α' AND Σχολ_έτος='2014-2015'  
GROUP BY Τάξη
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Τάξη και το Πλήθος των μαθητών, ανά τάξη, εκτός της Α, για το Σχολικό έτος είναι 2014-2015, από τον πίνακα ΕγγραφήΜαθητή. Η φράση where, αποκλείει ομάδες.

Η Φράση HAVING

Μπορούμε να εφαρμόσουμε μια συνθήκη σε μια συγκεκριμένη ομάδα από εγγραφές χρησιμοποιώντας το having. Η συνθήκη στην φράση having εφαρμόζεται αφού σχηματιστούν οι ομάδες (group by) και υπολογιστούν οι συναθροιστικές συναρτήσεις.

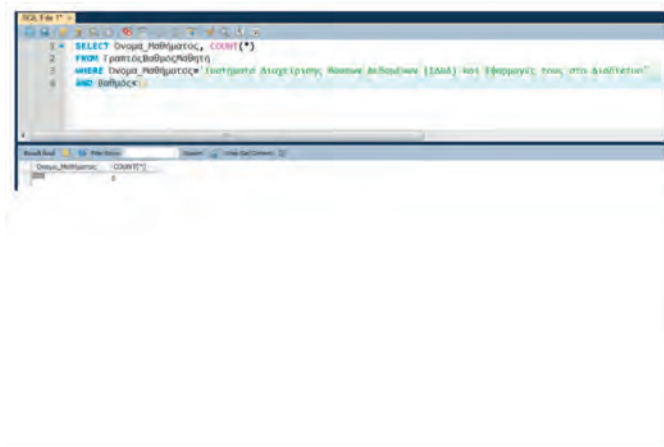
ΠΑΡΑΔΕΙΓΜΑΤΑ

```
SELECT Όνομα_Μαθήματος, AVG(Βαθμός)  
FROM ΓραπτόςΒαθμόςΜαθητή  
GROUP BY Όνομα_Μαθήματος  
HAVING AVG(Βαθμός)>15
```

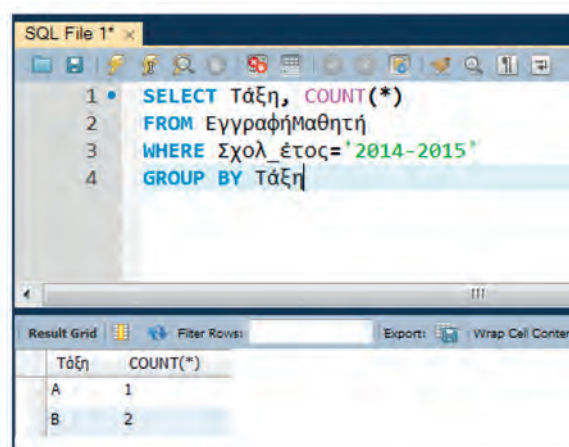
Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Όνομα_Μαθήματος και το μέσο όρο του βαθμού, ανά μάθημα από τον πίνακα ΓραπτόςΒαθμόςΜαθητή για τα μαθήματα που ο βαθμός τους είναι μεγαλύτερος από 15

```
SELECT Τάξη, Σχολ_έτος, AVG(Βαθμός)  
FROM ΓραπτόςΒαθμόςΜαθητή, ΕγγραφήΜαθητή  
WHERE ΓραπτόςΒαθμόςΜαθητή.ΑΜ=ΕγγραφήΜαθητή.ΑΜ  
GROUP BY Τάξη, Σχολ_έτος  
HAVING AVG(Βαθμός)>15
```

Δηλαδή επέλεξε και εμφάνισε τις πληροφορίες Τάξη, Σχολικό έτος και το μέσο όρο του βαθμού, ανά τάξη και Σχολικό έτος, από τους πίνακες ΓραπτόςΒαθμόςΜαθητή, ΕγγραφήΜαθητή για τα μαθήματα που ο βαθμός τους είναι μεγαλύτερος από 15



Εικόνα 4.2.6.1 Χρήση της συνάρτησης count() για το πλήθος των εγγραφών



Εικόνα 4.2.6.2 Χρήση της συνάρτησης count() για το των εγγραφών, ανά τάξη



Ερωτήσεις / Δραστηριότητες

1. Προσδιορίστε και αναπτύξτε τις αντίστοιχες εντολές select, ώστε να απαντηθούν τα ερωτήματα: Πόσα αγόρια και πόσα κορίτσια φοιτούν το σχολικό έτος 2014-2015 στην Α τάξη; (χρησιμοποιήστε ψευδώνυμο στήλης για την παρουσίαση των αποτελεσμάτων) Πόσοι μαθητές δεν έχουν γραπτό βαθμό στο μάθημα Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές στο Διαδίκτυο το σχολικό έτος 2014-2015; (χρησιμοποιήστε ψευδώνυμο στήλης για την παρουσίαση των αποτελεσμάτων) Ποιος είναι ο Μέσος Όρος των γραπτών βαθμών ανά μάθημα το σχολικό έτος 2014-2015; (χρησιμοποιήστε ψευδώνυμο στήλης για την παρουσίαση των αποτελεσμάτων)

Δραστηριότητα Κεφαλαίου 4

Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να εφαρμόζει τις γνώσεις που έχει ήδη αποκτήσει, σε ένα νέο πρόβλημα
- να επεκτείνει την λύση προβλημάτων που έχει ήδη αντιμετωπίσει
- να λαμβάνει αποφάσεις σχετικά με την υλοποίηση και διαχείριση Βάσεων Δεδομένων
- για συγκεκριμένα προβλήματα

Η βάση δεδομένων του μαθητολογίου πρέπει να συμπληρωθεί με τα στοιχεία των καθηγητών και την ανάθεση μαθημάτων σε αυτούς. Γι' αυτό το λόγο πρέπει στην βάση δεδομένων να γίνουν οι εξής ενέργειες:

1. Εισαγωγές περισσότερων μαθημάτων που διδάσκονται στο συγκεκριμένο σχολείο.
2. Δημιουργία πίνακα για την αποθήκευση των στοιχείων των καθηγητών όπως Επώνυμο, Όνομα και Ειδικότητα. Σημείωση: προτείνεται να δημιουργηθεί κύριο κλειδί που θα είναι αριθμός.
3. Εισαγωγές δεδομένων στον πίνακα με τα στοιχεία των καθηγητών.
4. Δημιουργία του πίνακα Αναθέσεις όπου θα καταγράφονται τα μαθήματα που ανατίθενται σε κάθε καθηγητή. Ο πίνακας αυτός πρέπει να συσχετίζει τα στοιχεία των καθηγητών με τα στοιχεία των μαθημάτων. Σημείωση: προτείνεται να γίνει η απλουστευτική θεώρηση ότι κάθε μάθημα το αναλαμβάνει μόνο ένας καθηγητής σε όλα τα τμήματα.
5. Εισαγωγές δεδομένων στον πίνακα Αναθέσεις. Επιπλέον είναι αναγκαίο να υλοποιηθούν τα παρακάτω ερωτήματα που θα διατεθούν στους χρήστες του συστήματος:
6. Ανάκτηση όλων των αναθέσεων μαθημάτων.

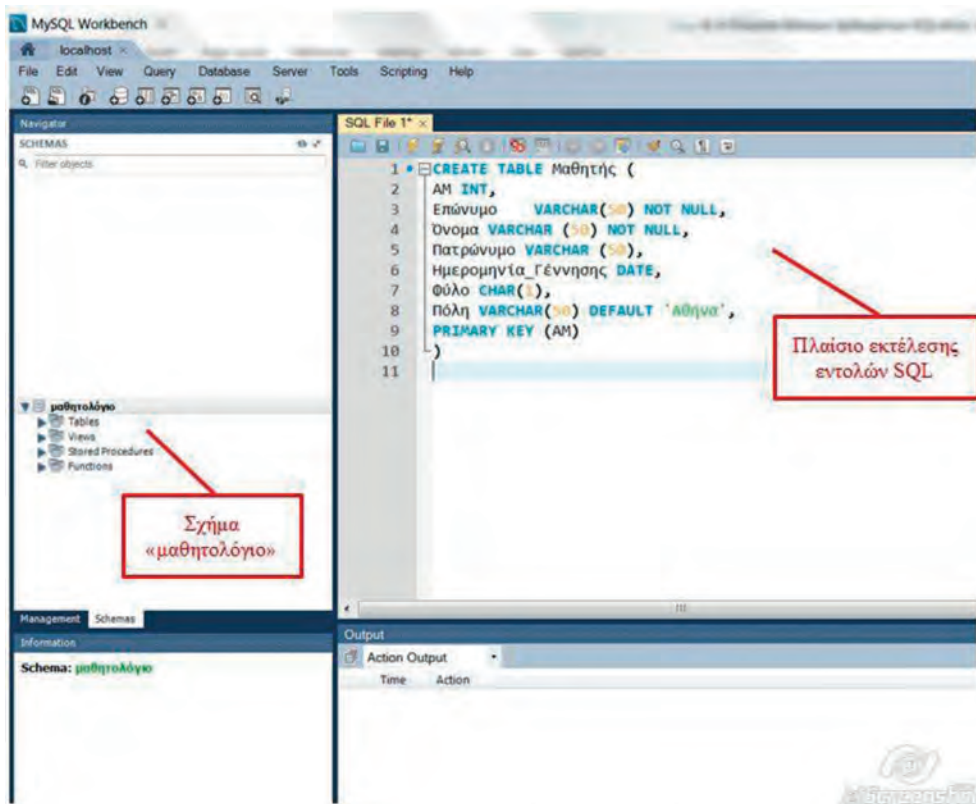
ΠΑΡΑΡΤΗΜΑ 4.1: Οι Τύποι Δεδομένων της Γλώσσας SQL

INTEGER	Ακέραιος αριθμός (χωρίς υποδιαστολή)
BIGINT	Ακέραιος αριθμός με ακρίβεια 19 ψηφία.
INTEGER(p)	Ακέραιος αριθμός με ακρίβεια p ψηφία.
SMALLINT	Ακέραιος αριθμός με ακρίβεια 5 ψηφία.
BINARY(n)	Σταθερού μήκους δυαδική συμβολοσειρά, μέγιστου μήκους n.
BOOLEAN	Τιμές TRUE ή FALSE.
BINARY VARYING(n), VARBINARY(n)	Μεταβλητού μήκους δυαδική συμβολοσειρά μέγιστου μήκους n.
FLOAT	Αριθμός με υποδιαστολή
DECIMAL(p, s)	Δεκαδικός αριθμός ακρίβειας p κλίμακας s. Η κλίμακα s δε μπορεί να υπερβαίνει την ακρίβεια p. Η ακρίβεια καθορίζει το πλήθος των ψηφίων αριστερά της υποδιαστολής, ενώ η κλίμακα το πλήθος των ψηφίων δεξιά της υποδιαστολής.
NUMERIC(p, s)	Το ίδιο με το DECIMAL.
FLOAT(p)	Προσεγγιστικός αριθμός με συντελεστή (mantissa) ακρίβειας p. Κινητής υποδιαστολής αριθμός σε εκθετική μορφή με βάση το 10.
REAL	Προσεγγιστικός αριθμός με συντελεστή (mantissa) ακρίβειας 7.
DOUBLE PRECISION	Προσεγγιστικός αριθμός με συντελεστή (mantissa) ακρίβειας 16.
CHARACTER(N)	Σειρά χαρακτήρων (συμβόλων) σταθερού μήκους N. Το μήκος N αναφέρεται στο μέγιστος μήκος της τιμής. Οι τιμές για πεδία αυτού του τύπου πρέπει να γράφονται μέσα σε μονά εισαγωγικά ``, π.χ. 'Γιώργος'.
CHARACTER VARYING(N), VARCHAR(N)	Συμβολοσειρά χαρακτήρων μεταβλητού μεγέθους, μέγιστου N.
DATE, TIME, TIMESTAMP	Τύποι που αποτελούνται από έναν αριθμό ακέραιων πεδίων που παριστάνουν μια απόλυτη χρονική στιγμή, ανάλογα με τον επιμέρους τύπο.
INTERVAL	Τύπος που αποτελείται από έναν αριθμό ακέραιων πεδίων που παριστάνουν μια χρονική περίοδο, ανάλογα με τον τύπο του διαστήματος.
COLLECTION (ARRAY, MULTISET)	Ο πίνακας ARRAY (προστέθηκε στο πρότυπο SQL99) είναι μια διατεταγμένη συλλογή στοιχείων. διατεταγμένη συλλογή στοιχείων. Ο τύπος MULTISET (προστέθηκε στο πρότυπο SQL2003) είναι μεταβλητού μήκους και μη διατεταγμένη συλλογή στοιχείων. Τα στοιχεία και των δύο τύπων πρέπει να ανήκουν σε έναν από τους προκαθορισμένους τύπους της SQL.
XML	Τύπος δεδομένων που αποθηκεύει δεδομένα μορφής XML.

ΠΑΡΑΡΤΗΜΑ 4.2: Δημιουργία Πινάκων στο Σχεσιακό ΣΔΒΔ MySQL

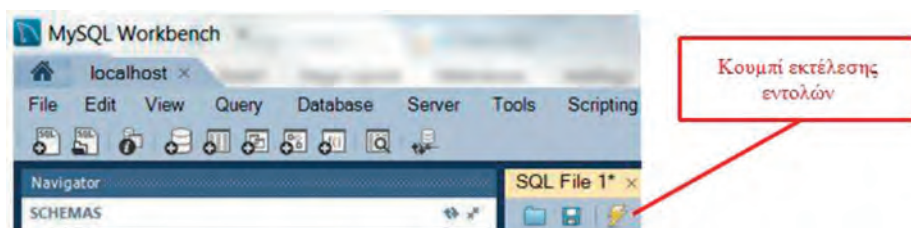
Το δωρεάν λογισμικό ΣΔΒΔ MySQL (έκδοση Community Server 5.6.21) υποστηρίζει τη δημιουργία πινάκων μέσω του εργαλείου γραφικής διαχείρισης MySQL Workbench. Το παρόν υλικό παρουσιάζει συνοπτικά την υλοποίηση των ενεργειών δημιουργίας πινάκων στο συγκεκριμένο περιβάλλον.

Στην Εικόνα 1, απεικονίζεται το βασικό περιβάλλον του εργαλείου MySQL Workbench καθώς και η εντολή CREATE TABLE για τη δημιουργία του πίνακα Μαθητής στο σχήμα «μαθητολόγιο».



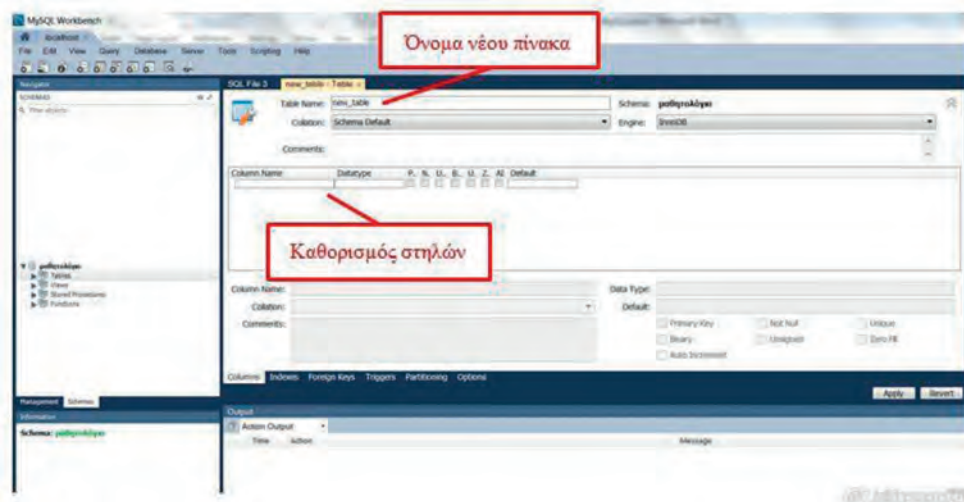
Εικόνα 1. Εντολή CREATE TABLE για τη δημιουργία του πίνακα Μαθητής στο περιβάλλον MySQL Workbench

Η εκτέλεση της εντολής πραγματοποιείται με το πάτημα του κουμπιού εκτέλεσης όπως φαίνεται στην παρακάτω εικόνα:



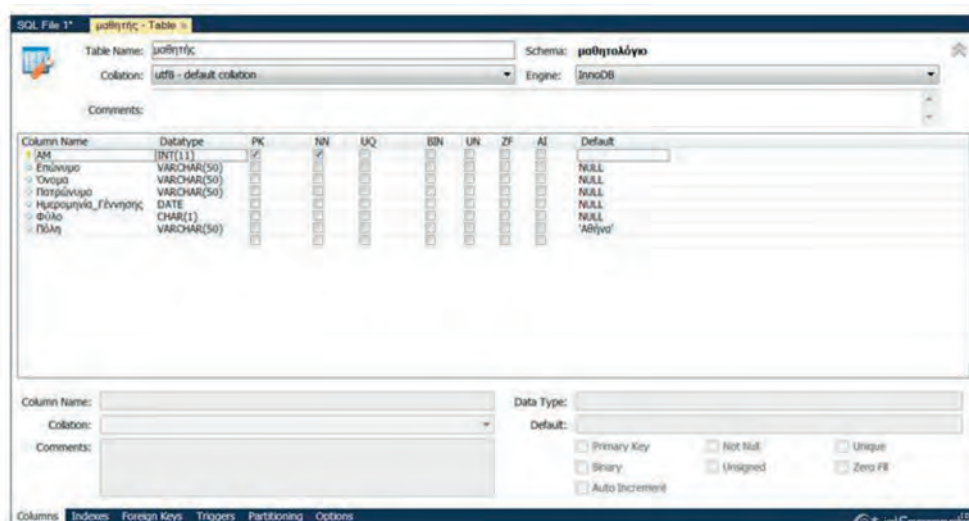
Εικόνα 2. Εκτέλεση εντολών στο περιβάλλον MySQL Workbench

Η δημιουργία πινάκων μπορεί να γίνει και με γραφικό τρόπο όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα 3. Γραφικό περιβάλλον δημιουργίας πινάκων

Στην παρακάτω εικόνα απεικονίζεται η δημιουργία/επεξεργασία του πίνακα Μαθητής με γραφικό τρόπο στο περιβάλλον MySQL Workbench.



Εικόνα 4. Δημιουργία/επεξεργασία του πίνακα Μαθητής στο περιβάλλον MySQL Workbench

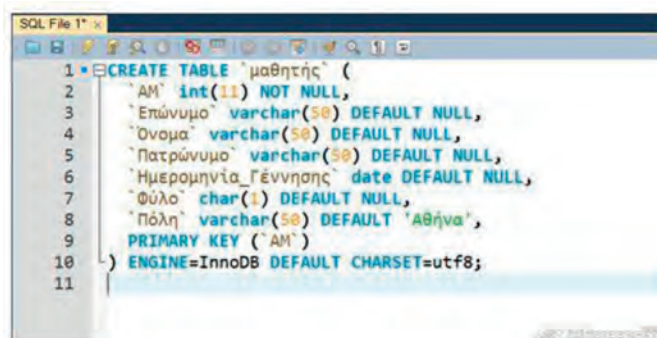
Παρατηρούμε τις στήλες του πίνακα (Column Name), τον τύπο δεδομένων τους (Datatype), το αν το κάθε πεδίο είναι κύριο κλειδί ή όχι (PK), αν επιτρέπονται ή όχι τιμές NULL σε κάθε πεδίο (NN) και ποια είναι η προκαθορισμένη τιμή (Default). Ο πίνακας δημιουργείται στο σχήμα «μαθητολόγιο», δηλαδή στη βάση δεδομένων που έχει δημιουργηθεί για τους πίνακες και τα στοιχεία του Μαθητολογίου.

Η επισκόπηση των πληροφοριών του πίνακα μετά τη δημιουργία του στο παράθυρο πλοήγησης (Navigator) του περιβάλλοντος απεικονίζεται στην Εικόνα 2.



Εικόνα 5. Επισκόπηση του πίνακα Μαθητής στο σχήμα Μαθητολόγιο

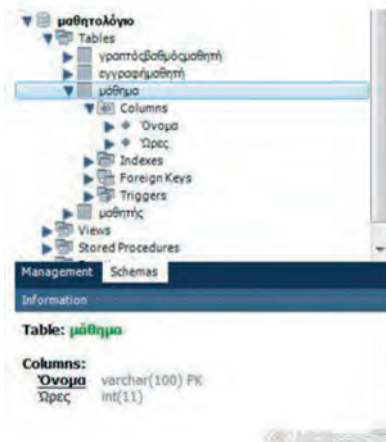
Ο κώδικας SQL για τη δημιουργία του πίνακα Μαθητής στο περιβάλλον MySQL Workbench απεικονίζεται παρακάτω.



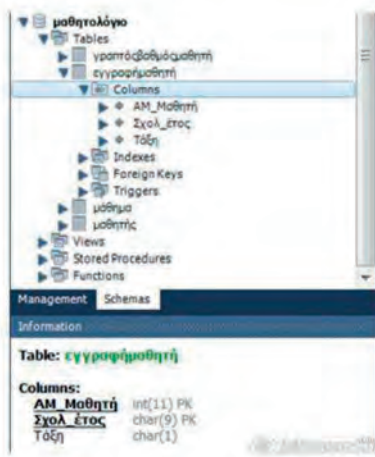
Εικόνα 6. Ο κώδικας SQL για τη δημιουργία του πίνακα Μαθητής στο σχήμα Μαθητολόγιο στο περιβάλλον MySQL Workbench

Παρατηρούμε ότι η δημιουργία της εντολής παραπάνω εντολής περιέχει προσθήκες του συγκεκριμένου ΣΔΒΔ που αφορούν στο σύνολο χαρακτήρων (DEFAULT CHARSET) για τα δεδομένα κειμένου που θα εισάγονται στον πίνακα, καθώς και τεχνικές λεπτομέρειες για τον τρόπο αποθήκευσης του πίνακα (ENGINE = InnoDB).

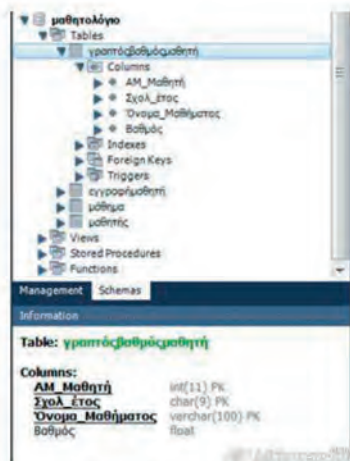
Οι παρακάτω τρεις εικόνες δείχνουν τους πίνακες Μάθημα, ΕγγραφήΜαθητή και ΓραπτόςΒαθμόςΜαθητή αντίστοιχα, μετά τη δημιουργία τους.



Εικόνα 7. Επισκόπηση του πίνακα Μάθημα στο σχήμα Μαθητολόγιο



Εικόνα 8. Επισκόπηση του πίνακα ΕγγραφήΜαθητή στο σχήμα Μαθητολόγιο



Εικόνα 9. Επισκόπηση του πίνακα ΓραπτόςΒαθμόςΜαθητή στο σχήμα Μαθητολόγιο

Ένα επιπλέον εργαλείο που έχει στη διάθεσή του ο διαχειριστής ΒΔ στο περιβάλλον MySQL, είναι το εργαλείο δημιουργίας μοντέλων, όπου έχει τη δυνατότητα να σχεδιάσει το σχήμα της βάσης δεδομένων με γραφικό τρόπο και να πραγματοποιήσει αλλαγές και προς τις δύο κατευθύνσεις: από τη σχεδίαση προς την υλοποίηση και από την υλοποίηση να ενημερώσει τη σχεδίαση.

Η παρακάτω εικόνα δείχνει το διάγραμμα πινάκων και συσχετίσεων της βάσης δεδομένων Μαθητολόγιο.



Εικόνα 10. Το διάγραμμα πινάκων και συσχετίσεων για τους πίνακες του μαθητολογίου

Οι γραμμές που συνδέουν τους πίνακες δηλώνουν τις συσχετίσεις μεταξύ τους, δηλαδή τους περιορισμούς ξένου κλειδιού (FOREIGN KEY). Παρατηρούμε ότι τα κύρια κλειδιά έχουν κίτρινη σήμανση, ενώ τα ξένα κλειδιά έχουν διαφορετική σήμανση (κόκκινο χρώμα) στο διάγραμμα.

κεφάλαιο

5

Διαχείριση Βάσεων Δεδομένων

5.1 Οι Ρόλοι του Διαχειριστή Βάσης Δεδομένων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να περιγράφει το ρόλο του Διαχειριστή Βάσης Δεδομένων
- να απαριθμεί τα καθήκοντα του Διαχειριστή Βάσης Δεδομένων
- να αναγνωρίζει τη σπουδαιότητα του έργου του Διαχειριστή Βάσης Δεδομένων να γνωρίζει τις προοπτικές απασχόλησης ενός Διαχειριστή Βάσης Δεδομένων

Ένα ΣΔΒΔ, όπως και οποιοδήποτε πληροφοριακό σύστημα, απαιτεί καθημερινή διαχείριση προκειμένου να εξασφαλιστεί η σωστή λειτουργία του. Η ανάγκη αυτή είναι αυξημένη στην περίπτωση των ΒΔ καθώς αυτές χρησιμοποιούνται από περισσότερα από ένα άτομα και εφαρμογές, αποτελούν δηλαδή μεριζόμενους πόρους. Για το σκοπό αυτό, στα τμήματα πληροφορικής προβλέπεται ο ρόλος του Διαχειριστή Βάσης Δεδομένων (Database Administrator ή DBA για συντομία).



Ο **Διαχειριστής Βάσης Δεδομένων** είναι μια από της κατηγορίες χρηστών μιας ΒΔ. Είναι υπεύθυνος για τον κεντρικό έλεγχο και την υποστήριξη της ΒΔ ώστε η ΒΔ να λειτουργεί σωστά και χωρίς προβλήματα.

Ανάλογα με το πλήθος, το μέγεθος και την πολυπλοκότητα των ΒΔ, ο ρόλος του Διαχειριστή ΒΔ μπορεί να ανατεθεί σε κάποιον παράλληλα με άλλα καθήκοντα (π.χ. διαχειριστής συστήματος, διαχειριστής δικτύου) ή να αποτελεί τα αποκλειστικά του καθήκοντα. Επίσης, δεν είναι σπάνια η περίπτωση ο ρόλος του Διαχειριστή ΒΔ να ανατίθεται από κοινού σε περισσότερα από ένα άτομα.

Οι Διαχειριστές ΒΔ είναι από τις πιο περιζήτητες και καλά αμειβόμενες θέσεις στο χώρο της Πληροφορικής, ειδικά στο εξωτερικό. Αυτό οφείλεται στη σπουδαιότητα του έργου του Διαχειριστή ΒΔ, καθώς και στις εξειδικευμένες γνώσεις και δεξιότητες που πρέπει να κατέχει ένας Διαχειριστής ΒΔ.



Τα **βασικά καθήκοντα** του Διαχειριστή ΒΔ είναι:

- εγκατάσταση και παραμετροποίηση του ΣΔΒΔ,
- ορισμός του σχήματος της ΒΔ,
- δημιουργία χρηστών της ΒΔ και απόδοση ρόλων στους χρήστες,
- καθορισμός μεθόδων προσπέλασης της ΒΔ,
- εργασίες συντήρησης της ΒΔ και αναβάθμισης του ΣΔΒΔ,

Τα καθήκοντα αυτά περιγράφονται αναλυτικά στις επόμενες ενότητες.



ΧΡΗΣΙΜΕΣ ΔΙΕΥΘΥΝΣΕΙΣ ΔΙΑΔΙΚΤΥΟΥ

Στατιστικά στοιχεία για την απασχόληση των Διαχειριστών ΒΔ

Bureau of Labor Statistics, US Department of Labor:

<http://www.bls.gov/ooh/computerandinformationtechnology/databaseadministrators.htm>, <http://stats.bls.gov/oes/current/oes151141.htm>



Ερωτήσεις / Δραστηριότητες

1. Αναζητήστε στατιστικά στοιχεία (από την Ελλάδα και το εξωτερικό) για τις προοπτικές του επαγγέλματος του Διαχειριστή ΒΔ.
2. Επισκεφθείτε ιστοσελίδες εύρεσης εργασίας από την Ελλάδα (π.χ. <http://www.skywalker.gr/>, <http://www.kariera.gr/>) ή το εξωτερικό (π.χ. <http://www.monster.com>, <http://www.dice.com/>) και αναζητήστε αγγελίες για Διαχειριστές ΒΔ. Καταγράψτε τις αρμοδιότητες της θέσης, τα προσόντα που ζητούνται και τις απολαβές που προσφέρονται.

5.1.1 Εγκατάσταση και Παραμετροποίηση του ΣΔΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να γνωρίζει τις εργασίες εγκατάστασης ενός ΣΔΒΔ
- να διακρίνει τις διαφορετικές μεθόδους εγκατάστασης ενός ΣΔΒΔ (χειροκίνητη και αυτόματη, πρόγραμμα εγκατάστασης και αποθετήριο λογισμικού)
- απαριθμεί τις βασικές παραμέτρους εγκατάστασης ενός ΣΔΒΔ
- διακρίνει τους διαφορετικούς τρόπους παραμετροποίησης ενός ΣΔΒΔ (κατά την εγκατάσταση, μετά την εγκατάσταση)

Πριν χρησιμοποιηθεί μια ΒΔ πρέπει να εγκατασταθεί το αντίστοιχο ΣΔΒΔ. Η εργασία αυτή αποτελεί ένα από τα βασικά καθήκοντα του Διαχειριστή ΒΔ. Η εγκατάσταση συνήθως πραγματοποιείται σε ισχυρούς υπολογιστές που αναλαμβάνουν το ρόλο του **εξυπηρετητή ΒΔ (database server)**. Το ΣΔΒΔ μπορεί να είναι το μοναδικό λογισμικό (μαζί με το λειτουργικό σύστημα) που θα εγκατασταθεί ή να συνυπάρχει με άλλες εφαρμογές. Η εγκατάσταση του ΣΔΒΔ μπορεί να γίνει και σε λιγότερο ισχυρούς υπολογιστές για τη δημιουργία περιβάλλοντος ανάπτυξης και ελέγχου από τους προγραμματιστές ΒΔ.

Τα αναγκαία αρχεία για την εγκατάσταση του ΣΔΒΔ προσφέρονται από τον κατασκευαστή του ΣΔΒΔ. Στο παρελθόν, τα αρχεία ήταν διαθέσιμα σε οπτικούς δίσκους (CD ή DVD), ενώ σήμερα, χάρη στις υψηλές ταχύτητες σύνδεσης στο Διαδίκτυο, είναι διαθέσιμα για μεταφόρτωση (download) από την ιστοσελίδα του κατασκευαστή. Για κάθε λειτουργικό σύστημα (Windows, UNIX, Linux, Mac OS) υπάρχει διαφορετική έκδοση των αρχείων εγκατάστασης.



Οι **εργασίες που πραγματοποιούνται κατά την εγκατάσταση του ΣΔΒΔ** είναι οι ακόλουθες:

- δημιουργία φακέλων εγκατάστασης,
- αποσυμπίεση των αρχείων εγκατάστασης,
- αντιγραφή των αποσυμπιεσμένων αρχείων στους φακέλους που δημιουργήθηκαν και σε φακέλους του λειτουργικού συστήματος,
- ρύθμιση του λειτουργικού συστήματος και των προγραμμάτων που αποτελούν το ΣΔΒΔ.

Η διαδικασία εγκατάστασης μπορεί να είναι **χειροκίνητη (manual)** ή **αυτόματη (automatic)**. Στη χειροκίνητη εγκατάσταση ο Διαχειριστής ΒΔ εργάζεται σε περιβάλλον γραμμής εντολών (command line) και ακολουθώντας τις οδηγίες του κατασκευαστή του ΣΔΒΔ πραγματοποιεί όλες τις απαραίτητες εργασίες. Η μέθοδος αυτή δεν είναι ιδιαίτερα δημοφιλής εξαιτίας της πολυπλοκότητας της. Στην αυτόματη εγκατάσταση οι απαραίτητες εργασίες γίνονται με ελάχιστη παρέμβαση του Διαχειριστή ΒΔ. Για την αυτόματη εγκατάσταση σε περιβάλλον MS Windows και Mac OS είναι διαθέσιμα **προγράμματα εγκατάστασης (installers)** που όταν εκτελεστούν ξεκινάει ένας **οδηγός εγκατάστασης (installation wizard)** σε περιβάλλον γραμμής εντολών (command line) ή σε γραφικό περιβάλλον (GUI – graphical user interface). Ο οδηγός εγκατάστασης αρχικά δίνει τη δυνατότητα στον Διαχειριστή ΒΔ να επιλέξει τις βασικές παραμέτρους εγκατάστασης και στη συνέχεια πραγματοποιούνται αυτόματα οι απαραίτητες εργασίες. Σε περιβάλλον UNIX και Linux η αυτόματη εγκατάσταση πραγματοποιείται μέσω ειδικών δικτυακών τόπων που ονομάζονται **αποθετήρια λογισμικού (software repositories)**. Ο Διαχειριστής ΒΔ συνδέεται στο αποθετήριο λογισμικού και επιλέγει τα **πακέτα λογισμικού (software packages)** του ΣΔΒΔ για μεταφόρτωση. Με την ολοκλήρωση της μεταφόρτωσης ξεκινάνε αυτόματα οι αναγκαίες εργασίες εγκατάστασης.

Κατά την εγκατάσταση του ΣΔΒΔ, χειροκίνητη ή αυτόματα, πραγματοποιείται η **αρχική παραμετροποίηση του ΣΔΒΔ, δηλαδή η ρύθμιση διαφόρων παραμέτρων που σχετίζονται με τη λειτουργία του ΣΔΒΔ**. Η παραμετροποίηση συνήθως αφορά στα ακόλουθα:

Φάκελοι εγκατάστασης: Καθορισμός φακέλων για τα αρχεία εφαρμογών (προγράμματα που αποτελούν το ΣΔΒΔ) και τα αρχεία δεδομένων (περιεχόμενα της ΒΔ). Είναι συνηθισμένη πρακτική ο φάκελος για τα αρχεία δεδομένων να βρίσκεται σε διαφορετικό διαμέρισμα (partition) του δίσκου ή σε διαφορετικό δίσκο από το λειτουργικό σύστημα και το ΣΔΒΔ. Με αυτό τον τρόπο αν γίνει επανεγκατάσταση του λειτουργικού συστήματος ή του ΣΔΒΔ τα περιεχόμενα της ΒΔ δεν θα επηρεαστούν. Επίσης αν τα αρχεία δεδομένων βρίσκονται σε άλλο δίσκο βελτιώνεται η ταχύτητα της ΒΔ.

Τύπος ΒΔ: Ανάλογα με τον σκοπό και την προβλεπόμενη χρήση της ΒΔ μπορεί να οριστεί ο ανάλογος τύπος ΒΔ που θα δημιουργείται από την εντολή **CREATE DATABASE** (αναλύεται σε επόμενη ενότητα). Η επιλογή



Το ΣΔΒΔ MySQL υποστηρίζει 9 μηχανές ΒΔ από τις οποίες οι πιο διαδεδομένες είναι η MyISAM και InnoDB.

τύπου ΒΔ καθορίζει συγκεκριμένες παραμέτρους της εγκατάστασης, όπως την κύρια μνήμη που θα δεσμεύει το ΣΔΒΔ, το μέγιστο μέγεθος που μπορεί να έχει η ΒΔ, τον τρόπο αποθήκευσης και οργάνωσης των αρχείων δεδομένων με τα περιεχόμενα της ΒΔ. Σε ορισμένα ΣΔΒΔ η επιλογή τύπου ΒΔ οδηγεί σε επιλογή κατάλληλης **μηχανής ΒΔ (database engine)**. Η μηχανή ΒΔ είναι το τμήμα του ΣΔΒΔ υπεύθυνο για την πρόσβαση στα δεδομένα της ΒΔ, δηλαδή υλοποιεί τις λειτουργίες της δημιουργίας, ανάγνωσης, ενημέρωσης και διαγραφής. **ΣΔΒΔ ως υπηρεσία:** Υπάρχει δυνατότητα να δημιουργηθεί υπηρεσία (service) του λειτουργικού συστήματος ώστε το ΣΔΒΔ να ξεκινάει αυτόματα κατά την εκκίνηση του υπολογιστή που είναι εγκατεστημένο. Εάν το ΣΔΒΔ χρησιμοποιείται σε παραγωγικό περιβάλλον τότε η αυτόματα εκκίνηση είναι επιβεβλημένη. Αντίθετα, εάν το ΣΔΒΔ χρησιμοποιείται για ανάπτυξη και έλεγχο εφαρμογών ΒΔ μπορεί να μην είναι επιθυμητή η αυτόματη εκκίνηση.

Λογαριασμός Διαχειριστή ΒΔ: Κάθε ΣΔΒΔ πρέπει να έχει τουλάχιστον έναν χρήστη με δικαιώματα Διαχειριστή ΒΔ ο οποίος να έχει απεριόριστο έλεγχο στο ΣΔΒΔ και τις ΒΔ. Ο χρήστης αυτός δημιουργείται αυτόματα κατά την εγκατάσταση του ΣΔΒΔ. Κάθε ΣΔΒΔ δημιουργεί λογαριασμό Διαχειριστή ΒΔ με συγκεκριμένο όνομα χρήστη (user name) και κωδικό τα οποία είναι γνωστά σε όλους καθώς αναφέρονται στις οδηγίες του κατασκευαστή. Γι' αυτό συστήνεται να ορίζεται για τον λογαριασμό του Διαχειριστή ΒΔ διαφορετικό όνομα και κωδικός από αυτά που προτείνονται κατά την εγκατάσταση.

Σύνολο χαρακτήρων (character set): Εάν στη ΒΔ θέλουμε να αποθηκευτούν δεδομένα που δεν χρησιμοποιούν το λατινικό αλφάβητο (π.χ. Ελληνικά), πρέπει να δηλωθεί κατά την εγκατάσταση του ΣΔΒΔ το κατάλληλο σύνολο χαρακτήρων. Αυτό είναι απαραίτητο προκειμένου το ΣΔΒΔ να μπορεί να χειριστεί σωστά τα δεδομένα, δηλαδή να μπορεί να τα εμφανίσει, να τα ταξινομήσει και να πραγματοποιεί αναζητήσεις.

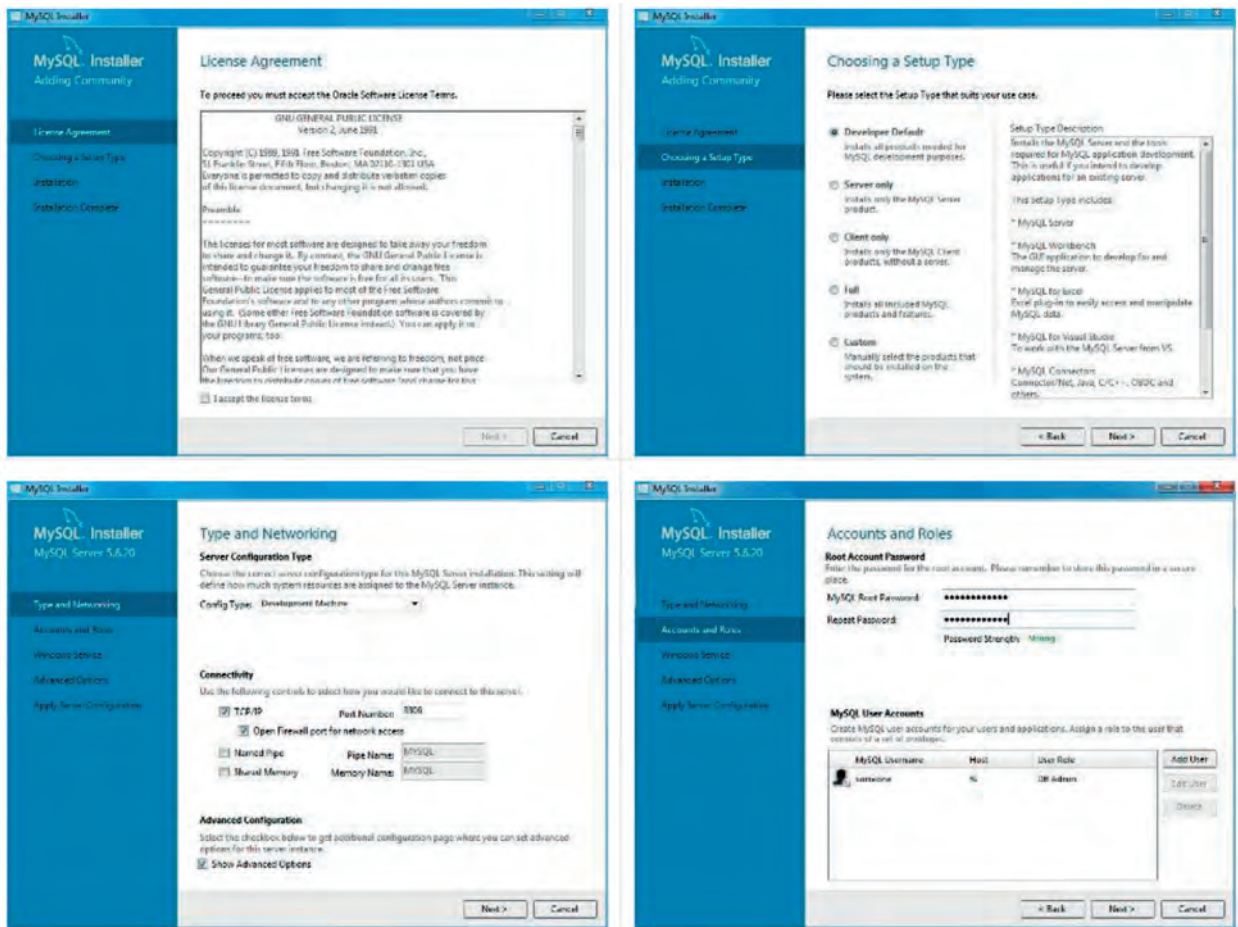
Επικοινωνία: Οι χρήστες και οι εφαρμογές που χρησιμοποιούν μια ΒΔ μπορεί να μην βρίσκονται στον ίδιο υπολογιστή με το ΣΔΒΔ. Σε αυτή την περίπτωση η επικοινωνία με τη ΒΔ είναι απομακρυσμένη και υλοποιείται μέσω του πρωτοκόλλου δικτύου TCP/IP. Κατά την εγκατάσταση του ΣΔΒΔ μπορούν να ρυθμιστούν οι λεπτομέρειες της απομακρυσμένης επικοινωνίας (π.χ. θύρα TCP/IP που θα χρησιμοποιείται) ή να απενεργοποιηθεί η δυνατότητα απομακρυσμένης επικοινωνίας αν δεν προβλέπονται απομακρυσμένοι χρήστες της ΒΔ.

Αρχεία καταγραφής (log files): Υπάρχει δυνατότητα το ΣΔΒΔ να καταγράφει στοιχεία για τη χρήση και λειτουργία της ΒΔ. Τα περιεχόμενα των αρχείων καταγραφής χρησιμοποιούνται από το Διαχειριστή ΒΔ για την παρακολούθηση της λειτουργίας της ΒΔ, τη βελτίωση της απόδοσής της και την επίλυση προβλημάτων.

Βοηθητικά εργαλεία: Εκτός από το ΣΔΒΔ μπορούν να εγκατασταθούν πρόσθετα προγράμματα για τη διαχείριση της ΒΔ ή την ανάπτυξη εφαρμογών ΒΔ.

Αντίγραφα ασφαλείας (backup) και αναπαραγωγή (replication): Κατά την εγκατάσταση του ΣΔΒΔ

μπορεί να οριστεί χρονοδιάγραμμα αυτόματης λήψης αντιγράφων ασφαλείας της ΒΔ. Επίσης υπάρχει δυνατότητα να οριστεί σχέδιο αναπαραγωγής της ΒΔ, για την αυτόματη δημιουργία αντιγράφων της ΒΔ σε άλλους υπολογιστές. Τα αντίγραφα είναι συγχρονισμένα, δηλαδή οι αλλαγές στα περιεχόμενα της ΒΔ μεταφέρονται αυτόματα στα αντίγραφα (ρεπλικές) της ΒΔ που βρίσκονται στους άλλους υπολογιστές.



Εικόνα 5.1 Γραφικός οδηγός εγκατάστασης ΣΔΒΔ MySQL: άδεια χρήσης (πάνω αριστερά), τύπος εγκατάστασης (πάνω δεξιά), ρυθμίσεις δικτύου (κάτω αριστερά), ορισμός χρήστη (κάτω δεξιά) [Πηγή: MySQL Installer GUI <http://dev.mysql.com/doc/refman/5.6/en/mysqlinstallergui.html>]

Επιπλέον παραμετροποίηση μπορεί να γίνει και μετά την εγκατάσταση του ΣΔΒΔ χρησιμοποιώντας εργαλεία που προσφέρει το ΣΔΒΔ ή εκτελώντας ξανά τον οδηγό εγκατάστασης και επιλέγοντας τροποποίηση της εγκατάστασης. Επίσης μπορεί να γίνει απευθείας αλλαγή στα περιεχόμενα των **αρχείων ρυθμίσεων (configuration files)** του ΣΔΒΔ που δημιουργήθηκαν κατά την εγκατάσταση. Η τελευταία επιλογή απαιτεί ιδιαίτερη προσοχή και αυξημένες γνώσεις καθώς υπάρχει κίνδυνος καταστροφής της ΒΔ από λανθασμένες ενέργειες.

ΧΡΗΣΙΜΕΣ ΔΙΕΥΘΥΝΣΕΙΣ ΔΙΑΔΙΚΤΥΟΥ



Αρχεία εγκατάστασης ΣΔΒΔ

MySQL: <https://dev.mysql.com/downloads/mysql/>

MS SQL Server Express: http://www.microsoft.com/enus/servercloud/products/sql_server-editions/sqlserverexpress.aspx

Oracle Database Express: <http://www.oracle.com/technetwork/database/database-technologies/expressedition/downloads/index.html>

IBM DB2 Express: http://www.ibm.com/software/data/db2/express_c/download.html

Οδηγίες εγκατάστασης ΣΔΒΔ

MySQL: <http://dev.mysql.com/doc/refman/5.6/en/installing.html> MS SQL Server Express: <https://msdn.microsoft.com/en-us/sqlserver2014express.aspx>

Oracle Database Express: Windows installation

http://docs.oracle.com/cd/E17781_01/install.112/e18803/toc.htm , Linux installation

http://docs.oracle.com/cd/E17781_01/install.112/e18802/toc.htm IBM DB2: http://www01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.qb.server.doc/lang=en



Ερωτήσεις / Δραστηριότητες

1. Ξεκινήστε τον οδηγό εγκατάστασης του ΣΔΒΔ MySQL (αν δεν υπάρχει στον υπολογιστή σας θα πρέπει να τον μεταφορτώσετε από την ιστοσελίδα <https://dev.mysql.com/downloads/mysql/>). Καταγράψτε ποιες ρυθμίσεις κάνατε μέσω του οδηγού και συγκρίνετε με όσα αναφέρονται σχετικά στο κείμενο της ενότητας.
2. Μεταφερθείτε στον φάκελο εγκατάστασης του ΣΔΒΔ MySQL. Εντοπίστε το αρχείο ρυθμίσεων (**my.ini** σε περιβάλλον **Windows** και **my.cnf** σε περιβάλλον Linux) και δείτε τα περιεχόμενά του με έναν απλό επεξεργαστή κειμένου (π.χ. Notepad σε Windows, GEdit σε Linux). Προσπαθήστε να εντοπίσετε τις γραμμές του αρχείου που αντιστοιχούν στις ρυθμίσεις που κάνατε κατά την εκτέλεση του οδηγού εγκατάστασης.
3. Επισκεφθείτε την ιστοσελίδα <https://dev.mysql.com/doc/refman/5.6/en/storage-engines.html> με την επίσημη τεκμηρίωση του ΣΔΒΔ MySQL σχετικά με τις διαθέσιμες **μηχανές ΒΔ**. Μελετήστε τον πίνακα στο κάτω μέρος της σελίδας και συγκρίνετε τις μηχανές ΒΔ ως προς το μέγεθος της ΒΔ (storage limits), την υποστήριξη συναλλαγών (transactions), υποστήριξη γεωγραφικών δεδομένων (geospatial support), δυνατότητα συμπίεσης και κρυπτογράφησης δεδομένων (compressed και encrypted data), υποστήριξη ξένων κλειδιών (foreign key).

5.1.2 Ορισμός του Σχήματος της ΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να χρησιμοποιεί την κατάλληλη εντολή SQL για να δημιουργήσει μια ΒΔ και να ορίζει το σύνολο χαρακτήρων των δεδομένων της
- να γνωρίζει τις αρμοδιότητες του Διαχειριστή ΒΔ σχετικά με τη δημιουργία ΒΔ και σχήματος ΒΔ
- να κατανοεί τους λόγους ασφάλειας που υπαγορεύουν η δημιουργία ΒΔ και σχήματος ΒΔ να γίνεται μόνο από τον Διαχειριστή ΒΔ

Πριν δημιουργηθούν πίνακες σε μια ΒΔ πρέπει να δημιουργηθεί η ΒΔ με την εντολή CREATE DATABASE. Για παράδειγμα η εκτέλεση της εντολής θα δημιουργήσει μια κενή ΒΔ με όνομα mathitologio.

```
CREATE DATABASE mathitologio;
```

Η εντολή CREATE DATABASE μπορεί να περιλαμβάνει παραμέτρους που θα καθορίζουν τεχνικές λεπτομέρειες της ΒΔ, π.χ. φάκελοι αποθήκευσης των αρχείων με τα περιεχόμενα της ΒΔ, μέγιστο μέγεθος αρχείων της ΒΔ, σύνολο χαρακτήρων που χρησιμοποιείται για αναπαράσταση των δεδομένων της ΒΔ και τρόπος ταξινόμησης των δεδομένων, χρήση αρχείων καταγραφής (log files). Αν η εντολή δεν περιλαμβάνει παραμέτρους τότε η ΒΔ δημιουργείται με τις προκαθορισμένες τιμές που δηλώθηκαν κατά την εγκατάσταση του ΣΔΒΔ.

Στο ΣΔΒΔ MySQL, κατά τη δημιουργία μιας ΒΔ μπορεί να καθοριστεί το σύνολο χαρακτήρων που χρησιμοποιούν τα δεδομένα (παράμετρος **CHARACTER SET**) και ο τρόπος ταξινόμησης για το επιλεγμένο σύνολο χαρακτήρων (παράμετρος **COLLATE**). Για παράδειγμα, η εντολή

```
CREATE DATABASE mathitologio CHARACTER SET greek COLLATE greek_general_ci;
```

θα δημιουργήσει μια κενή ΒΔ με όνομα mathitologio, τα δεδομένα της οποίας μπορούν να περιλαμβάνουν και Ελληνικούς χαρακτήρες (επιπλέον των λατινικών) σύμφωνα με την κωδικοποίηση "ISO 88597 Greek" και θα ταξινομούνται σύμφωνα με τη σειρά των γραμμάτων στο Ελληνικό αλφάβητο. Εναλλακτικά μπορεί να χρησιμοποιηθεί κωδικοποίηση Unicode που υποστηρίζει όλα τα διαθέσιμα αλφάβητα (άρα και το ελληνικό), όπως φαίνεται στο ακόλουθο παράδειγμα.

```
CREATE DATABASE mathitologio CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Ο λογαριασμός χρήστη με δικαιώματα Διαχειριστή ΒΔ που δημιουργείται κατά την εγκατάσταση του ΣΔΒΔ έχει πάντα τη δυνατότητα να δημιουργεί ΒΔ. Για λόγους ασφάλειας συνήθως η δυνατότητα δημιουργίας ΒΔ δεν δίνεται σε άλλους χρήστες.

Αφού δημιουργηθεί η ΒΔ, μπορούν με τις κατάλληλες εντολές SQL να δημιουργηθούν πίνακες. Ο λογαριασμός χρήστη με δικαιώματα Διαχειριστή ΒΔ που δημιουργείται κατά την εγκατάσταση του ΣΔΒΔ έχει πάντα αυτή τη δυνατότητα. Σε περιβάλλον ανάπτυξης και ελέγχου εφαρμογών ΒΔ η δυνατότητα αυτή μπορεί να δοθεί και στα μέλη της ομάδας ανάπτυξης. Σε παραγωγικό περιβάλλον όμως που η ΒΔ περιέχει πραγματικά δεδομένα και υποστηρίζει κάποια λειτουργία ή παρέχει κάποια υπηρεσία, αυτή τη δυνατότητα έχει συνήθως μόνο ο Διαχειριστής ΒΔ για λόγους ασφάλειας. Σε αυτή την περίπτωση ο Διαχειριστής ΒΔ θα πρέπει να εκτελέσει τις κατάλληλες εντολές SQL, ακολουθώντας τις οδηγίες της ομάδας ανάπτυξης, για να δημιουργήσει πίνακες στη ΒΔ.

Σε ορισμένα ΣΔΒΔ, μια ΒΔ μπορεί να περιέχει ένα ή περισσότερα σχήματα. Ένα **σχήμα (schema)** ομαδοποιεί πίνακες της ΒΔ, π.χ. ανά χρήστη ή εφαρμογή που χρησιμοποιεί τη ΒΔ, με σκοπό την καλύτερη διαχείριση και μεγαλύτερη ασφάλεια των περιεχομένων της ΒΔ. Για τη δημιουργία σχήματος χρησιμοποιείται η εντολή **CREATE SCHEMA** η οποία συνοδεύεται από εντολές SQL για τη δημιουργία πινάκων της ΒΔ. Αυτό είναι ιδιαίτερα χρήσιμο σε ΒΔ που περιλαμβάνουν μεγάλο πλήθος πινάκων.



Το ΣΔΒΔ **MySQL** στην τελευταία έκδοσή του (5.6.25, Μάιος 2015) δεν υποστηρίζει τη δημιουργία σχήματος και η εντολή **CREATE SCHEMA** έχει την ίδια σύνταξη και χρήση με την εντολή **CREATE DATABASE** (δηλαδή ο όρος SCHEMA χρησιμοποιείται ως συνώνυμο του όρου DATABASE).

5.1.3 Δημιουργία Χρηστών και Απόδοση Ρόλων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να χρησιμοποιεί εντολές SQL για να διαχειρίζεται λογαριασμούς του ΣΔΒΔ (δημιουργία, μετονομασία, διαγραφή, ορισμός και αλλαγή κωδικού)
- να κατανοεί την αναγκαιότητα προστασίας των λογαριασμών με κωδικό πρόσβασης
- να κατανοεί την αναγκαιότητα μετονομασίας των λογαριασμών με δικαιώματα Διαχειριστή ΒΔ που δημιουργούνται αυτόματα κατά την εγκατάσταση του ΣΔΒΔ
- να περιγράφει τα προνόμια που μπορούν να αποδοθούν σε ένα λογαριασμό
- να χρησιμοποιεί εντολές SQL για να διαχειρίζεται προνόμια (εκχώρηση, ανάκληση, εμφάνιση)
- να κατανοεί τα κενά ασφάλειας που δημιουργούνται αν επιτραπεί σε χρήστες που δεν είναι Διαχειριστές ΒΔ να εκχωρούν προνόμια
- να γνωρίζει καλές πρακτικές για τη σωστή διαχείριση των προνομίων

Κάθε άτομο ή εφαρμογή που θέλει να χρησιμοποιήσει μια ΒΔ πρέπει πρώτα να συνδεθεί στο ΣΔΒΔ χρησιμοποιώντας ένα **λογαριασμό χρήστη (user account)**. Οι συγκεκριμένοι λογαριασμοί είναι διαφορετικοί από τους λογαριασμούς του λειτουργικού συστήματος. Υπάρχουν ΣΔΒΔ που μπορούν να συνδέσουν τους δύο τύπους λογαριασμών, ώστε ο χρήστης να μην χρειάζεται να ταυτοποιηθεί δύο φορές, μία από το λειτουργικό σύστημα και μία από το ΣΔΒΔ, για να χρησιμοποιήσει τη ΒΔ.



Η σύνδεση των λογαριασμών του ΣΔΒΔ με τους λογαριασμούς του λειτουργικού συστήματος μειώνει την ασφάλεια του συστήματος καθώς αν κάποιος εισβολέας παραβιάσει ένα λογαριασμό του λειτουργικού συστήματος θα αποκτηθεί αυτόματα πρόσβαση και στο ΣΔΒΔ.

Η δημιουργία λογαριασμών χρήστη στο ΣΔΒΔ είναι από τις σπουδαιότερες αρμοδιότητες του Διαχειριστή ΒΔ καθώς από τη σωστή διαχείριση των λογαριασμών εξαρτάται η ασφάλεια της ΒΔ. Ο λογαριασμός χρήστη με δικαιώματα Διαχειριστή ΒΔ που δημιουργείται κατά την εγκατάσταση του ΣΔΒΔ είναι αρχικά ο μοναδικός που μπορεί να δημιουργήσει άλλους λογαριασμούς.

Η δημιουργία λογαριασμών χρήστη υλοποιείται με διαφορετικό τρόπο από κάθε ΣΔΒΔ, με αποτέλεσμα η σύνταξη των σχετικών εντολών να διαφέρει. Στο ΣΔΒΔ MySQL χρησιμοποιείται η εντολή **CREATE USER**. Για παράδειγμα η εντολή

```
CREATE USER 'g_pappas';
```

δημιουργεί ένα λογαριασμό με όνομα χρήστη g_pappas.

Οι λογαριασμοί χρήστη μπορούν να διαθέτουν κωδικό πρόσβασης. Στο ΣΔΒΔ MySQL, αν στην εντολή CREATE USER χρησιμοποιηθεί η παράμετρος **IDENTIFIED BY** μπορεί να οριστεί κωδικός πρόσβασης. Για παράδειγμα η εντολή

```
CREATE USER 'k_aggelou' IDENTIFIED BY 'p@$$';
```

δημιουργεί ένα λογαριασμό με όνομα χρήστη k_aggelou και κωδικό πρόσβασης p@\$\$.



Για λόγους ασφάλειας, **κάθε λογαριασμός χρήστη του ΣΔΒΔ θα πρέπει να προστατεύεται με κωδικό πρόσβασης (password)**. Από τις πρώτες ενέργειες του Διαχειριστή ΒΔ μετά την εγκατάσταση του ΣΔΒΔ πρέπει να είναι ο ορισμός κωδικών για όλους τους λογαριασμούς που δημιουργούνται αυτόματα κατά την εγκατάσταση του ΣΔΒΔ.

Τα στοιχεία των λογαριασμών χρήστη αποθηκεύονται σε ειδικό πίνακα που δημιουργείται αυτόματα κατά την εγκατάσταση του ΣΔΒΔ. Στο ΣΔΒΔ MySQL αυτός ο πίνακας ονομάζεται `mysql.user`. Με την εντολή

```
SELECT * FROM mysql.USER;
```

θα εμφανιστούν οι λεπτομέρειες όλων των λογαριασμών που έχουν δημιουργηθεί. Θα παρατηρήσετε ότι οι κωδικοί πρόσβασης είναι ακατάληπτοι. Αυτό συμβαίνει επειδή οι κωδικοί που ορίζονται με την εντολή **CREATE USER ... IDENTIFIED BY ...**; κωδικοποιούνται πριν αποθηκευτούν στον πίνακα `mysql.user`. Έτσι ακόμα κι αν ένα μη εξουσιοδοτημένο άτομο αποκτήσει πρόσβαση στον πίνακα `mysql.user` δεν θα μπορέσει να ανακτήσει τους κωδικούς.

Για να αλλάξουμε τον κωδικό ενός λογαριασμού ή να ορίσουμε κωδικό σε περίπτωση που αυτό δεν είχε γίνει κατά τη δημιουργία του λογαριασμού, στο ΣΔΒΔ MySQL χρησιμοποιούμε την εντολή **SET PASSWORD**. Για παράδειγμα η εντολή

```
SET PASSWORD FOR 'a_dimou' = PASSWORD('0p3n');
```

ορίζει κωδικό πρόσβασης 0p3n για τον λογαριασμό με όνομα χρήστη `a_dimou`. Η συνάρτηση `PASSWORD(...)` κρυπτογραφεί τον κωδικό πρόσβασης πριν αποθηκευτεί στον πίνακα `mysql.user` με τα στοιχεία των λογαριασμών.



Στο ΣΔΒΔ **MySQL** η εντολή **SET PASSWORD** δεν κωδικοποιεί τους κωδικούς – σε αντίθεση με την εντολή **CREATE USER ... IDENTIFIED BY ...**; – γι' αυτό πρέπει να χρησιμοποιηθεί η συνάρτηση **PASSWORD(...)**.

Υπάρχει δυνατότητα αλλαγής του ονόματος ενός λογαριασμού. Αυτό συνήθως γίνεται για λόγους ασφάλειας. Για παράδειγμα, στο ΣΔΒΔ MySQL το προτεινόμενο όνομα του λογαριασμού με δικαιώματα Διαχειριστή ΒΔ που δημιουργείται κατά την εγκατάσταση του ΣΔΒΔ είναι `root`. Αν χρησιμοποιηθεί αυτό το όνομα τότε μειώνεται η ασφάλεια του συστήματος καθώς από τα δύο συστατικά του λογαριασμού (όνομα και κωδικός) το ένα είναι ήδη γνωστό (όνομα) και το μόνο που χρειάζεται κάποιος εισβολέας είναι να μαντέψει το άλλο (κωδικός). Στο ΣΔΒΔ MySQL η μετονομασία ενός λογαριασμού γίνεται με την εντολή **RENAME USER**. Για παράδειγμα η εντολή

```
RENAME USER 'root' TO 'db_admin';
```

θα μετονομάσει τον λογαριασμό με όνομα `root` σε `db_admin`.



Από τις πρώτες ενέργειες του Διαχειριστή ΒΔ μετά την εγκατάσταση του ΣΔΒΔ πρέπει να είναι η **μετονομασία των λογαριασμών με δικαιώματα Διαχειριστή ΒΔ που έχουν συνηθισμένα ονόματα ή ονόματα που αναφέρονται στις οδηγίες του κατασκευαστή**.

Υπάρχουν περιπτώσεις που ένας λογαριασμός πρέπει να καταργηθεί, π.χ. επειδή το άτομο που τον χρησιμοποιούσε μετακινήθηκε σε άλλη θέση ή αποχώρησε. Για τον σκοπό αυτό χρησιμοποιείται η εντολή **DROP USER**. Για παράδειγμα η εντολή

```
RENAME USER 'm_alexiou';
```

θα διαγράψει από το ΣΔΒΔ το λογαριασμό με όνομα χρήστη `m_alexiou`.



Οι ενέργειες που μπορεί να εκτελέσει ένας λογαριασμός χρήστη του ΣΔΒΔ καθορίζονται από τα δικαιώματα που διαθέτει ο λογαριασμός. Στην ορολογία των ΒΔ, τα δικαιώματα αυτά ονομάζονται **προνόμια (privileges)**. Στην ορολογία των ΒΔ, όταν δίνεται ένα προνόμιο λέμε ότι το προνόμιο **εκχωρείται (grant)** και όταν αφαιρείται λέμε ότι το προνόμιο **ανακαλείται (revoke)**.

Ένα προνόμιο μπορεί να είναι καθολικό, δηλαδή να αφορά όλες τις ΒΔ δεδομένων του ΣΔΒΔ.

Εναλλακτικό το προνόμιο μπορεί να είναι για συγκεκριμένη ΒΔ του ΣΔΒΔ, για συγκεκριμένο πίνακα μιας ΒΔ ή για συγκεκριμένη στήλη ενός πίνακα. Τα πιο συνηθισμένα προνόμια που υποστηρίζει το ΣΔΒΔ MySQL και το επίπεδο στο οποίο μπορούν να εφαρμοστούν παρουσιάζονται στον ακόλουθο πίνακα.

Πίνακας 5.1 Κατάλογος προνομίων του ΣΔΒΔ MySQL

Προνόμιο	Επιτρεπόμενες ενέργειες	Επίπεδο
ALL	Όλα τα προνόμια εκτός από GRANT OPTION	Καθολικό, ΒΔ, Πίνακας, Στήλη
ALTER	Τροποποίηση της δομής πίνακα	Καθολικό, ΒΔ, Πίνακας
CREATE	Δημιουργία ΒΔ και πίνακα	Καθολικό, ΒΔ, Πίνακας
CREATE USER	Δημιουργία, διαγραφή και μετονομασία λογαριασμού χρήστη	Καθολικό
DELETE	Διαγραφή γραμμών από πίνακα	Καθολικό, ΒΔ, Πίνακας
DROP	Διαγραφή ΒΔ, πίνακα και όψης	Καθολικό, ΒΔ, Πίνακας
GRANT OPTION	Εκχώρηση και ανάκληση προνομίου σε λογαριασμό χρήστη	Καθολικό, ΒΔ, Πίνακας
INDEX	Δημιουργία και διαγραφή ευρετηρίου	Καθολικό, ΒΔ, Πίνακας
INSERT	Εισαγωγή γραμμών σε πίνακα	Καθολικό, ΒΔ, Πίνακας, Στήλη
REFERENCES	Δημιουργία ξένου κλειδιού	Καθολικό, ΒΔ, Πίνακας, Στήλη
SELECT	Επιλογή γραμμών από πίνακα	Καθολικό, ΒΔ, Πίνακας, Στήλη
SHUTDOWN	Τερματισμός ΣΔΒΔ	Καθολικό
SUPER	Χρήση εργαλείων διαχείρισης ΣΔΒΔ, παραμετροποίηση ΣΔΒΔ	Καθολικό
UPDATE	Ενημέρωση γραμμών πίνακα	Καθολικό, ΒΔ, Πίνακας, Στήλη



Ένας λογαριασμός χρήστη μπορεί να εκχωρήσει σε άλλο λογαριασμό μόνο προνόμια που ήδη κατέχει. Ο λογαριασμός χρήστη με δικαιώματα Διαχειριστή ΒΔ που δημιουργείται κατά την εγκατάσταση του ΣΔΒΔ κατέχει όλα τα διαθέσιμα προνόμια σε όλα τα επίπεδα και στη συνέχεια μπορεί να τα εκχωρήσει στους λογαριασμούς που θα δημιουργηθούν.

Στο ΣΔΒΔ MySQL η εκχώρηση προνομίου γίνεται με την εντολή **GRANT**, η οποία έχει την ακόλουθη σύνταξη

```
GRANT <privilege> ON <level> TO <user>;
```

όπου <privilege> το όνομα του προνομίου, <level> το επίπεδο στο οποίο εφαρμόζεται το προνόμιο και <user> το όνομα του λογαριασμού χρήστη στον οποίο εκχωρείται το προνόμιο. Το επίπεδο εφαρμογής του προνομίου μπορεί να λάβει τις ακόλουθες τιμές:

- *.*** για καθολικό προνόμιο που αφορά όλες τις ΒΔ του ΣΔΒΔ,

- db.*** για προνόμιο στη ΒΔ με όνομα db,

- db.table** για προνόμιο στον πίνακα με όνομα table που βρίσκεται μέσα στη ΒΔ με όνομα db,

- db.table(column)** για προνόμιο στη στήλη με όνομα column του πίνακα με όνομα table που βρίσκεται στη ΒΔ με όνομα db.

Με μία εντολή GRANT μπορούν να εκχωρηθούν περισσότερα από ένα προνόμια σε περισσότερους από έναν λογαριασμούς. Σε αυτή την περίπτωση η εντολή περιλαμβάνει λίστες από προνόμια και λίστες από χρήστες, με τα στοιχεία κάθε λίστας να διαχωρίζονται με κόμματα.

Ακολουθούν παραδείγματα εκχώρησης προνομίων στο ΣΔΒΔ MySQL σε όλα τα διαθέσιμα επίπεδα (καθολικό, ΒΔ, πίνακας, στήλη). **Στα παραδείγματα θεωρούμε ότι οι εντολές εκτελούνται από λογαριασμό χρήστη με δικαιώματα Διαχειριστή ΒΔ, ο οποίος κατέχει όλα τα διαθέσιμα προνόμια σε όλα τα πιθανά επίπεδα (καθολικό, ΒΔ, πίνακας, στήλη).**

```
GRANT ALL ON *.* TO 'g_pappas';
```

Εκχωρούνται όλα τα διαθέσιμα προνόμια στο λογαριασμό χρήστη με όνομα g_pappas. Ουσιαστικά ο λογαριασμός g_pappas αποκτάει δικαιώματα Διαχειριστή ΒΔ καθώς μπορεί να εκτελέσει οποιαδήποτε ενέργεια σε όλες τις ΒΔ του ΣΔΒΔ.



Το ειδικό προνόμιο **ALL** σημαίνει «όλα των διαθέσιμα προνόμια για το αντίστοιχο επίπεδο εφαρμογής (καθολικό, ΒΔ, πίνακας, στήλη)». Δηλαδή αν εφαρμοστεί καθολικά περιλαμβάνει όλα τα καθολικά προνόμια, αν εφαρμοστεί σε ΒΔ περιλαμβάνει όλα τα προνόμια για ΒΔ κ.ο.κ. Μια εντολή **GRANT** με το προνόμιο **ALL** δεν μπορεί να περιλαμβάνει άλλα προνόμια παρά μόνο το προνόμιο **GRANT OPTION**.

```
GRANT CREATE, DROP ON mathitologio.* TO 'k_aggelou';
```

Στο λογαριασμό χρήστη με όνομα k_aggelou εκχωρείται τα προνόμια δημιουργίας πίνακα (CREATE) στη ΒΔ mathitologio και το προνόμιο διαγραφής (DROP) οποιουδήποτε πίνακα της ΒΔ mathitologio.

```
GRANT SELECT, INSERT ON mathitologio.Μάθημα  
TO 'a_dimou', 'm_alexiou';
```

Στους λογαριασμούς χρήστη με ονόματα a_dimou και m_alexiou εκχωρείται το προνόμιο επιλογής γραμμών (SELECT) και το προνόμιο εισαγωγής γραμμών (INSERT) στον πίνακα Μάθημα της ΒΔ mathitologio.

```
GRANT SELECT (ΑΜ, Επώνυμο, Όνομα), UPDATE (ΑΜ)  
ON mathitologio.Μαθητής TO 'd_georgiou';
```

Εκχωρούνται στο λογαριασμό χρήστη με όνομα d_georgiou το προνόμιο επιλογής (SELECT) στις στήλες ΑΜ, Επώνυμο και Όνομα του πίνακα Μαθητής της ΒΔ mathitologio, και το προνόμιο ενημέρωσης (UPDATE) στη στήλη ΑΜ του πίνακα Μαθητής της ΒΔ mathitologio. Δηλαδή ο χρήστης d_georgiou μπορεί με την εντολή

SELECT να εμφανίσει μόνο τα περιεχόμενα των στηλών AM, Επώνυμο και Όνομα και με την εντολή UPDATE μπορεί να ενημερώσει μόνο τα περιεχόμενα της στήλης AM.

Υπάρχει ένα ειδικό προνόμιο που ονομάζεται **GRANT OPTION** και δίνει τη δυνατότητα σε ένα λογαριασμό χρήστη να εκχωρεί προνόμια σε άλλους λογαριασμούς. Αυτό γίνεται προσθέτοντας την επιλογή **WITH GRANT OPTION** στο τέλος μιας εντολής GRANT. Για παράδειγμα, στο ΣΔΒΔ MySQL η εντολή

```
GRANT SELECT ON mathitologio.* TO 'i_petrou' WITH GRANT OPTION;
```

εκχωρεί στο λογαριασμό χρήστη με όνομα i_petrou το προνόμιο επιλογής γραμμών από όλους τους πίνακες της ΒΔ mathitologio και παράλληλα επιτρέπει στον χρήστη i_petrou να εκχωρήσει προνόμια που ήδη διαθέτει για τη ΒΔ mathitologio σε άλλο λογαριασμό χρήστη.

Ο λογαριασμός χρήστη με δικαιώματα Διαχειριστή ΒΔ που δημιουργείται κατά την εγκατάσταση του ΣΔΒΔ είναι αρχικά ο μοναδικός που μπορεί να εκχωρεί προνόμια σε άλλους λογαριασμούς.



Το προνόμιο **GRANT OPTION** πρέπει να χρησιμοποιείται με προσοχή, καθώς όταν εκχωρηθεί δίνει το δικαίωμα σε ένα χρήστη να εκχωρήσει σε κάποιον άλλο οποιοδήποτε προνόμιο **ήδη κατέχει ή θα αποκτήσει στο μέλλον**. Γι' αυτό θα πρέπει να εκχωρείται με μεγάλη προσοχή καθώς μπορεί να χρησιμοποιηθεί για να παρακαμφθούν οι κανόνες ασφάλειας που έχει θέσει ο Διαχειριστής ΒΔ.

Για να γίνουν κατανοητά τα προηγούμενα, έστω ότι ο Διαχειριστής ΒΔ εκχωρεί στον χρήστη user1 τα προνόμια SELECT, INSERT, UPDATE, DELETE με GRANT OPTION, και στον χρήστη user2 τα προνόμια SELECT, UPDATE. Ο user1 έχει το προνόμιο GRANT OPTION οπότε μπορεί να εκχωρήσει στον user2 τα προνόμια INSERT, DELETE παρόλο που ο Διαχειριστής ΒΔ είχε αποφασίσει ότι ο user2 δεν έπρεπε να τα κατέχει. Έστω ότι στη συνέχεια ο Διαχειριστής ΒΔ εκχωρεί στον user1 το προνόμιο DROP (χωρίς την επιλογή GRANT OPTION). Παρόλο που ο user1 απέκτησε το προνόμιο DROP χωρίς την επιλογή GRANT OPTION, μπορεί να εκχωρήσει το προνόμιο DROP στον user2 καθώς ήδη διαθέτει το προνόμιο GRANT OPTION από προηγούμενη εντολή. Γι' αυτό το λόγο, σε παραγωγικό περιβάλλον που η ΒΔ περιέχει πραγματικά δεδομένα, μόνο ο Διαχειριστής ΒΔ θα πρέπει να διαθέτει το προνόμιο GRANT OPTION.



Για την προστασία της δομής και των περιεχομένων μιας ΒΔ προτείνονται οι ακόλουθοι **κανόνες για τη διαχείριση των προνομίων**:

- Σε κάθε λογαριασμό να δίνονται τα ελάχιστα προνόμια που χρειάζεται ο χρήστης του λογαριασμού για να χρησιμοποιήσει τη ΒΔ. Δηλαδή αν κάποιος απλώς αναζητάει δεδομένα τότε ο λογαριασμός του θα πρέπει να έχει μόνο το προνόμιο SELECT.
- Τα προνόμια να δίνονται για συγκεκριμένη ΒΔ και αν είναι δυνατό για συγκεκριμένο πίνακα της ΒΔ. Καθολικά προνόμια να διαθέτει μόνο ο Διαχειριστής ΒΔ.
- Το προνόμιο GRANT OPTION να διαθέτει μόνο ο Διαχειριστής ΒΔ.

Υπάρχουν περιπτώσεις που κάποια προνόμια πρέπει να ανακληθούν από ένα λογαριασμό, επειδή για παράδειγμα άλλαξαν οι αρμοδιότητες του χρήστη. Η ανάκληση προνομίων γίνεται με την εντολή **REVOKE**. Για παράδειγμα, στο ΣΔΒΔ MySQL με την εντολή

```
REVOKE SELECT, INSERT ON mathitologio.mathima TO 'a_dimou';
```

θα ανακληθεί από το λογαριασμό χρήστη με όνομα a_dimou το προνόμιο επιλογής γραμμών (SELECT) και το

προνόμιο εισαγωγής γραμμών (INSERT) στον πίνακα mathima της ΒΔ mathitologio.



Για να ανακαλέσει ένας χρήστης προνόμια από κάποιον άλλο χρήστη θα πρέπει να διαθέτει το προνόμιο **GRANT OPTION** καθώς και τα προνόμια που προσπαθεί να ανακαλέσει.

Όπως γίνεται κατανοητό, αν το πλήθος των χρηστών μιας ΒΔ είναι μεγάλο και γίνονται συχνά αλλαγές στα προνόμια, δεν είναι εύκολο να θυμάται ο Διαχειριστής ΒΔ τα προνόμια κάθε χρήστη. Γι' αυτό υπάρχει η εντολή **SHOW GRANTS** που εμφανίζει τα προνόμια ενός λογαριασμού. Για παράδειγμα, η εντολή

```
SHOW GRANTS FOR 'm_alexiou';
```

θα εμφανίσει τις εντολές GRANT που απαιτήθηκαν για να αποκτήσει ο λογαριασμός χρήστη με όνομα m_alexiou τα προνόμια που διαθέτει.

Είναι συνηθισμένο σε μια ΒΔ να υπάρχουν πολλοί χρήστες που λόγω της εργασίας τους πρέπει να έχουν τα ίδια ακριβώς προνόμια. Για τις περιπτώσεις αυτές, ορισμένα ΣΔΒΔ υποστηρίζουν τη δημιουργία **ρόλων (roles)**. Ένας ρόλος δημιουργείται με την εντολή **CREATE ROLE** στην οποία προσδιορίζεται το όνομα του ρόλου. Στη συνέχεια εκχωρούνται προνόμια στο ρόλο με μια εντολή GRANT, στην οποία αντί για όνομα λογαριασμού αναφέρεται το όνομα του ρόλου. Τέλος, εκχωρείται ο ρόλος σε λογαριασμούς με μια εντολή GRANT, στην οποία αντί για προνόμια αναφέρεται ο ρόλος. Με αυτό τον τρόπο οι λογαριασμοί αποκτούν τα προνόμια που διαθέτει ο ρόλος. Η ύπαρξη ρόλων διευκολύνει το έργο του Διαχειριστή ΒΔ καθώς οι αλλαγές προνομίων γίνονται στους ρόλους αντί σε κάθε λογαριασμό ξεχωριστά.



Το ΣΔΒΔ **MySQL** στην τελευταία έκδοσή του (5.6.25, Μάιος 2015) **δεν** υποστηρίζει ρόλους.



Ερωτήσεις / Δραστηριότητες

1. Συνδεθείτε στο ΣΔΒΔ MySQL με λογαριασμό χρήστη που διαθέτει τα προνόμια CREATE USER και GRANT OPTION.
 - Δημιουργήστε ένα νέο λογαριασμό χρήστη χρησιμοποιώντας ένα όνομα της επιλογής σας.
 - Στον λογαριασμό που δημιουργήσατε, ορίστε ως κωδικό πρόσβασης τη λέξη secret.
 - Στον λογαριασμό που δημιουργήσατε, εκχωρήστε το προνόμιο SELECT στη ΒΔ mathitologio, το προνόμιο INSERT στον πίνακα Μάθημα της ΒΔ mathitologio και το προνόμιο UPDATE στις στήλες Επώνυμο και Όνομα του πίνακα Μαθητής της ΒΔ mathitologio.
 - Χρησιμοποιήστε την κατάλληλη εντολή SQL για να εμφανίσετε τα προνόμια του λογαριασμού που δημιουργήσατε.
 - Συνδεθείτε με τον λογαριασμό που δημιουργήσατε και υποβάλετε διάφορα ερωτήματα SQL για να επιβεβαιώσετε ότι ο λογαριασμός διαθέτει τα προνόμια που εκχωρήθηκαν.

5.1.4 Καθορισμός Μεθόδων Προσπέλασης



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να κατανοεί την αλλαγή που προκαλεί η ύπαρξη ευρετηρίου στη μέθοδο προσπέλασης των περιεχόμενα της ΒΔ
- να εντοπίζει τις στήλες ενός πίνακα για τις οποίες πρέπει να δημιουργηθεί ευρετήριο
- να χρησιμοποιεί εντολές SQL για να διαχειρίζεται ευρετήρια (δημιουργία, διαγραφή)
- να κατανοεί πως επηρεάζεται ο χρόνος εκτέλεσης διαφορετικών τύπων ερωτημάτων (ανάκτηση, εισαγωγή, ενημέρωση, διαγραφή) από την ύπαρξη ευρετηρίου

Ο τρόπος που γίνεται η προσπέλαση των περιεχομένων μιας ΒΔ όταν υποβάλλονται ερωτήματα έχει μεγάλη σημασία, καθώς επηρεάζει το **χρόνο απόκρισης της ΒΔ, δηλαδή το χρόνο που απαιτείται για να εμφανιστούν τα αποτελέσματα ενός ερωτήματος**. Καθώς τα περισσότερα ερωτήματα αφορούν συνήθως ανάκτηση δεδομένων, είναι σημαντικό ο χρόνος εκτέλεσης των εντολών SELECT να μην είναι μεγάλος. Υπάρχουν περιπτώσεις που για να εκτελεστεί μια εντολή SELECT το ΣΔΒΔ πρέπει να διαβάσει κάθε γραμμή ενός πίνακα προκειμένου να επιλέξει τις γραμμές που θα επιστραφούν ως αποτέλεσμα του ερωτήματος. Αυτό συμβαίνει όταν η εντολή SELECT:

- περιέχει συνθήκη (WHERE),
- περιέχει συναρτήσεις για εύρεση μέγιστης (MAX) και ελάχιστης (MIN) τιμής, απαιτεί ταξινόμηση των αποτελεσμάτων (ORDER BY),
- απαιτεί ομαδοποίηση των αποτελεσμάτων (GROUP BY).

Για παράδειγμα, στη ΒΔ mathitologio υπάρχει ο πίνακας Μάθημα(Όνομα, Ώρες) με τις εβδομαδιαίες ώρες διδασκαλίας κάθε μαθήματος. Για να εμφανίσουμε τα μαθήματα που διδάσκονται 4 ώρες την εβδομάδα, θα υποβάλουμε το ακόλουθο ερώτημα:

```
SELECT *  
FROM Μάθημα  
WHERE Ώρες=4;
```

Για το συγκεκριμένο ερώτημα, το ΣΔΒΔ θα διαβάσει κάθε γραμμή του πίνακα Μάθημα και για κάθε γραμμή θα ελέγξει αν η τιμή της στήλης Ώρες είναι 4. Όσες γραμμές ικανοποιούν αυτή τη συνθήκη θα συμπεριληφθούν στα αποτελέσματα τους ερωτήματος.

Συνεχίζοντας το προηγούμενο παράδειγμα, αν θέλουμε να βρούμε το μέγιστο αριθμό ωρών που διδάσκεται ένα μάθημα θα υποβάλουμε το ακόλουθο ερώτημα:

```
SELECT MAX(Ώρες)  
FROM Μάθημα;
```

Για το συγκεκριμένο ερώτημα, το ΣΔΒΔ θα διαβάσει από κάθε γραμμή του πίνακα Μάθημα την τιμή της στήλης Ώρες ώστε να βρει τη μέγιστη τιμή της στήλης.

Γίνεται λοιπόν κατανοητό ότι όσο περισσότερες γραμμές περιέχει ένας πίνακας τόσο αυξάνεται ο χρόνος εκτέλεσης μιας εντολής SELECT. Αν ο χρόνος αυτός δεν είναι αποδεκτός από τους χρήστες της ΒΔ τότε η ΒΔ δεν θα χρησιμοποιηθεί. Φανταστείτε για παράδειγμα τι θα συμβεί αν η ΒΔ που εξυπηρετεί τα ΑΤΜ μιας τράπεζας χρειάζεται μερικά λεπτά (αντί για δευτερόλεπτα) για να επιστρέψει αποτελέσματα σχετικά με το υπόλοιπο ενός λογαριασμού.

Ο χρόνος εκτέλεσης μιας εντολής SELECT μπορεί να μειωθεί αν χρησιμοποιηθεί **ευρετήριο (index)**. Η έν-

νοια του ευρετηρίου είναι γνωστή καθώς χρησιμοποιείται και σε τομείς εκτός της Πληροφορικής. Για παράδειγμα, αρκετά βιβλία στο τέλος έχουν ευρετήριο όρων που αποτελείται από ζεύγη λέξεων-αριθμών σελίδας. Έτσι αν κάποιος ψάχνει τις σελίδες που αναφέρονται σε κάποιο θέμα δεν χρειάζεται να διαβάσει ολόκληρο το βιβλίο από την πρώτη μέχρι την τελευταία σελίδα, αλλά μπορεί να συμβουλευτεί το ευρετήριο και να μεταφερθεί στην αντίστοιχη σελίδα. Με ανάλογο τρόπο **στις ΒΔ μπορούν να δημιουργηθούν ευρετήρια για τις τιμές που περιέχει μία ή περισσότερες στήλες ενός πίνακα ώστε να επιταχύνονται ερωτήματα που αφορούν τις συγκεκριμένες στήλες.**



Σε όλα τα ΣΔΒΔ **δημιουργείται αυτόματα ευρετήριο για τη στήλη ή τις στήλες που αποτελούν το κύριο κλειδί κάθε πίνακα.** Επιπλέον ευρετήρια θα πρέπει να δημιουργηθούν για στήλες ή συνδυασμό στηλών στις οποίες πραγματοποιούνται συχνά ερωτήματα.

Η δημιουργία ευρετηρίου γίνεται με την εντολή CREATE INDEX. Για παράδειγμα, η εντολή

```
CREATE INDEX Ωρες_idx  
ON Μάθημα(Ωρες);
```

θα δημιουργήσει ένα ευρετήριο για τη στήλη Ωρες του πίνακα Μάθημα. Έτσι εντολές SELECT που αφορούν τη στήλη Ωρες θα εκτελούνται πιο γρήγορα καθώς το ΣΔΒΔ θα χρησιμοποιεί το ευρετήριο αντί να ελέγχει τις τιμές της στήλης Ωρες για κάθε γραμμή του πίνακα.

Ένα ευρετήριο μπορεί να αφορά περισσότερες από μία στήλες ενός πίνακα. Για παράδειγμα, στη ΒΔ mathitologio υπάρχει ο πίνακας Μαθητής(ΑΜ, Επώνυμο, Όνομα, Πατρώνυμο, Ημερομηνία_Γέννησης, Φύλο, Πόλη) με τα στοιχεία των μαθητών. Με την εντολή

```
CREATE INDEX ΕπώνυμοΌνομα_idx  
ON Μαθητής(Επώνυμο, Όνομα);
```

θα δημιουργήσουμε ένα ευρετήριο για τις στήλες Επώνυμο και Όνομα. Το ευρετήριο αυτό θα επιταχύνει εντολές SELECT που αφορούν μόνο τη στήλη Επώνυμο ή το συνδυασμό των στηλών Επώνυμο και Όνομα. Εντολές SELECT όμως που αφορούν μόνο τη στήλη Όνομα δεν θα ωφεληθούν από το συγκεκριμένο ευρετήριο. Για να γίνει αυτό κατανοητό, έστω τα ακόλουθα ερωτήματα:

```
SELECT *  
FROM Μαθητής  
WHERE Επώνυμο='Παπαδόπουλος';  
  
SELECT *  
FROM Μαθητής  
WHERE Επώνυμο='Παπαδόπουλος' AND Όνομα='Ιωάννης';  
  
SELECT *  
FROM Μαθητής  
WHERE Όνομα='Ιωάννης';
```

Τα πρώτα δύο ερωτήματα θα αξιοποιήσουν το ευρετήριο για τις στήλες Επώνυμο και Όνομα, ενώ το τρίτο δεν θα μπορέσει να το χρησιμοποιήσει.

Σε ορισμένα ΣΔΒΔ υπάρχει δυνατότητα στην εντολή CREATE INDEX να οριστούν διάφορες παράμετροι, όπως για παράδειγμα η μέθοδος ταξινόμησης των περιεχομένων του ευρετηρίου (αύξουσα ή φθίνουσα), ο τρόπος οργάνωσης των περιεχομένων του ευρετηρίου (π.χ. B+ δέντρο, πίνακας κατακερματισμού), το μέγεθος και ο τρόπος αποθήκευσης των αρχείων που αποτελούν το ευρετήριο.

Η απόφαση για τη δημιουργία ευρετηρίων λαμβάνεται αρχικά από την ομάδα ανάπτυξης μιας εφαρμογής ΒΔ, με βάση τα ερωτήματα που θα υποβάλλονται στη ΒΔ. Επιπλέον ευρετήρια μπορούν να δημιουργηθούν κατά την παραγωγική λειτουργία μιας ΒΔ. Ειδικότερα, ο Διαχειριστής ΒΔ πρέπει να παρακολουθεί τη λειτουργία της ΒΔ για να εντοπίσει ερωτήματα με μεγάλους χρόνους εκτέλεσης (ο τρόπος που γίνεται αυτό εξηγείται σε επόμενη ενότητα). Σε αυτή την περίπτωση, ο Διαχειριστής ΒΔ ακολουθώντας τις οδηγίες της ομάδας ανάπτυξης θα δημιουργήσει τα κατάλληλα ευρετήρια και στη συνέχεια θα κάνει μετρήσεις για να επιβεβαιώσει ότι ο χρόνος απόκρισης της ΒΔ βελτιώθηκε μετά τη δημιουργία των ευρετηρίων. Είναι προφανές ότι για λόγους ασφάλειας, όταν η ΒΔ βρίσκεται σε παραγωγικό περιβάλλον και περιέχει πραγματικά δεδομένα, η δημιουργία ευρετηρίων πρέπει να γίνεται μόνο από τον Διαχειριστή ΒΔ.

Υπάρχουν περιπτώσεις που η ύπαρξη ενός ευρετηρίου δεν είναι επιθυμητή, π.χ. επειδή διαπιστώθηκε ότι το ευρετήριο δεν βελτίωσε τον χρόνο απόκρισης της ΒΔ ή επειδή δεν υποβάλλονται πλέον ερωτήματα που να αξιοποιούν το συγκεκριμένο ευρετήριο. Η διαγραφή ενός ευρετηρίου γίνεται με την εντολή **DROP INDEX**. Για παράδειγμα, με την εντολή

```
DROP INDEX ωρες_idx  
ON Μάθημα;
```

θα διαγραφεί το ευρετήριο στη στήλη 'Ωρες' του πίνακα Μάθημα που δημιουργήθηκε σε προηγούμενο παράδειγμα.



Η δημιουργία ευρετηρίων **μειώνει** το χρόνο εκτέλεσης των εντολών ανάκτησης γραμμών (SELECT), αλλά **αυξάνει** το χρόνο εκτέλεσης των εντολών εισαγωγής (INSERT), ενημέρωσης (UPDATE) και διαγραφής (DELETE) γραμμών καθώς εκτός από τον πίνακα πρέπει να ενημερωθούν και τα αντίστοιχα ευρετήρια. Γι' αυτό ο Διαχειριστής ΒΔ θα πρέπει να παρακολουθεί τους χρόνους εκτέλεσης των ερωτημάτων και να διαχειρίζεται κατάλληλα τα ευρετήρια.



Ερωτήσεις / Δραστηριότητες

1. Συνδεθείτε στο ΣΔΒΔ MySQL με λογαριασμό χρήστη που διαθέτει το προνόμιο INDEX.

- Μελετήστε τους πίνακες της ΒΔ mathitologio που δημιουργήσατε στο Κεφάλαιο 4.
- Συζητήστε πιθανά ερωτήματα SQL που μπορεί να υποβάλλονται για να εντοπίσετε στήλες (ή συνδυασμούς στηλών) για τις οποίες θα μπορούσαν να δημιουργηθούν ευρετήρια.
- Δημιουργήστε τα ευρετήρια που σχεδιάσατε στο προηγούμενο βήμα.

5.1.5 Εργασίες Συντήρησης και Αναβάθμισης



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να γνωρίζει τις καθημερινές εργασίες για τις οποίες είναι υπεύθυνος ο Διαχειριστής ΒΔ
- να αναγνωρίζει τη σπουδαιότητα ενεργειών διαχείρισης του ΣΔΒΔ (παρακολούθηση λειτουργίας, δημιουργία και επαναφορά αντιγράφων ασφαλείας, αναβάθμιση)
- να χρησιμοποιεί εντολές για την εκκίνηση και τερματισμό του ΣΔΒΔ
- να χρησιμοποιεί τα αρχεία καταγραφής του ΣΔΒΔ για να παρακολουθεί τη λειτουργία του ΣΔΒΔ
- να αξιολογεί το πλάνο εκτέλεσης ενός ερωτήματος και να το βελτιώνει με τη χρήση ευρετηρίων
- να χρησιμοποιεί εντολές για τη δημιουργία και επαναφορά αντιγράφων ασφαλείας μιας ΒΔ
- να εφαρμόζει καλές πρακτικές για την αναβάθμιση του ΣΔΒΔ

Η σημαντικότερη αρμοδιότητα ενός Διαχειριστή ΒΔ είναι η συνεχής παρακολούθηση και υποστήριξη της ΒΔ ώστε να εξασφαλιστεί η λειτουργία της ΒΔ χωρίς διακοπές και χωρίς προβλήματα.



Για την καλή λειτουργία της ΒΔ, ο Διαχειριστής ΒΔ είναι υπεύθυνος για τις ακόλουθες εργασίες, πολλές από τις οποίες πραγματοποιούνται σε καθημερινή βάση:

- έναρξη και τερματισμός ΣΔΒΔ,
- παρακολούθηση λειτουργίας ΣΔΒΔ και των ΒΔ,
- εντοπισμός και επίλυση προβλημάτων του ΣΔΒΔ και των ΒΔ,
- δημιουργία και επαναφορά αντιγράφων ασφαλείας,
- αναβάθμιση ΣΔΒΔ.

Ο τερματισμός του ΣΔΒΔ δεν πραγματοποιείται συχνά, ιδιαίτερα σε περιβάλλον παραγωγής όπου το ΣΔΒΔ φιλοξενεί ΒΔ με πραγματικά δεδομένα. Υπάρχουν περιπτώσεις που ο τερματισμός είναι αναγκαίος, όπως για παράδειγμα για την πραγματοποίηση εργασιών συντήρησης (π.χ. αναβάθμιση στο λογισμικό του ΣΔΒΔ, αναβάθμιση στο λειτουργικό σύστημα του υπολογιστή που είναι εγκατεστημένο του ΣΔΒΔ, μεταφορά του ΣΔΒΔ και των ΒΔ σε άλλο υπολογιστή), για την αντιμετώπιση προβλημάτων (π.χ. αντικατάσταση ελαττωματικών στοιχείων υλικού), για τη λήψη ή επαναφορά αντιγράφου ασφαλείας του ΣΔΒΔ και των ΒΔ που φιλοξενεί. Για λόγους ασφάλειας, σε περιβάλλον παραγωγής η εκκίνηση και ο τερματισμός του ΣΔΒΔ γίνεται μόνο από τον Διαχειριστή ΒΔ.

Ο τερματισμός του ΣΔΒΔ MySQL γίνεται με την εντολή `mysqld stop` και η εκκίνηση με την εντολή **mysql start**. Οι εντολές αυτές πρέπει να εκτελεστούν σε περιβάλλον γραμμής εντολών του λειτουργικού συστήματος, στον φάκελο που έχει εγκατασταθεί το ΣΔΒΔ MySQL.

Σε περιβάλλον **Windows** ο φάκελος εγκατάστασης του ΣΔΒΔ MySQL είναι ο `C:\mysql\bin`. Οπότε η εντολή για τον **τερματισμό** του ΣΔΒΔ είναι:

```
C:\mysql\bin\mysqld stop
```

Αντίστοιχα για την **εκκίνηση** του ΣΔΒΔ η εντολή είναι:

```
C:\mysql\bin\mysqld
```

Σε περιβάλλον **Linux** ο φάκελος εγκατάστασης του ΣΔΒΔ MySQL είναι ο `/etc/init.d`. Οπότε για τον **τερματισμό** του ΣΔΒΔ χρησιμοποιείται η εντολή:

```
/etc/init.d/mysqld stop
```

Αντίστοιχα για την **εκκίνηση** του ΣΔΒΔ χρησιμοποιείται η εντολή:

```
/etc/init.d/mysqld
```



Για να μπορέσει ένας χρήστης να τερματίσει το ΣΔΒΔ **MySQL** πρέπει να διαθέτει το καθολικό πρό- νόμιο **SHUTDOWN**.

Μια από τις καθημερινές εργασίες του Διαχειριστή ΒΔ είναι η παρακολούθηση της λειτουργίας του ΣΔΒΔ προκειμένου να εντοπιστούν δυσλειτουργίες ή προβλήματα. Βασικό εργαλείο για την παρακολούθηση του ΣΔΒΔ είναι τα αρχεία καταγραφής.



Τα **αρχεία καταγραφής (log files)** του ΣΔΒΔ είναι αρχεία κειμένου που περιέχουν εγγραφές για διάφορα συμβάντα. Κάθε εγγραφή περιλαμβάνει ημερομηνία και ώρα του συμβάντος καθώς και μια σύντομη περιγραφή του συμβάντος.



Στο ΣΔΒΔ **MySQL** είναι διαθέσιμα τα ακόλουθα **αρχεία καταγραφής**:

- αρχείο καταγραφής σφαλμάτων (**error log**)
- αρχείο καταγραφής ερωτημάτων (**query log**)
- αρχείο καταγραφής αργών ερωτημάτων (**slow query log**)

Τα αρχεία καταγραφής βρίσκονται στον ίδιο φάκελο με τα αρχεία που περιέχουν τα δεδομένα των ΒΔ.

Το **αρχείο καταγραφής σφαλμάτων** έχει την κατάληξη **.err**. Στο αρχείο αυτό καταγράφονται **πληρο- φορίες για την εκκίνηση ή τερματισμό του ΣΔΒΔ καθώς και κρίσιμα σφάλματα που παρουσιάστη- καν κατά τη λειτουργία του ΣΔΒΔ**. Ελέγχοντας τα περιεχόμενα του αρχείου ο Διαχειριστής ΒΔ μπορεί να εντοπίσει σφάλματα στη λειτουργία του ΣΔΒΔ τα οποία πρέπει να επιδιορθώσει.

Το **αρχείο καταγραφής ερωτημάτων** έχει την κατάληξη **.log**. Σε αυτό καταγράφονται όλες **οι συνδέ- σεις που γίνονται από τους χρήστες στο ΣΔΒΔ και τα ερωτήματα που υποβάλουν**. Το συγκεκριμένο αρχείο προσφέρει πλήρη εικόνα της δραστηριότητας στις ΒΔ του ΣΔΒΔ. Έτσι, αν προκύψει κάποιο πρόβλημα στο ΣΔΒΔ ο Διαχειριστής ΒΔ μπορεί να συμβουλευτεί το συγκεκριμένο αρχείο για να εντοπίσει πιθανές αιτίες του προβλήματος.

Το **αρχείο καταγραφής αργών ερωτημάτων** έχει όνομα που τελειώνει σε **—slow.log**. Το αρχείο περιέχει **ερωτήματα που ο χρόνος εκτέλεσής τους ξεπερνάει μια προκαθορισμένη τιμή** (η τιμή μπορεί να αλ- λαχθεί από τον Διαχειριστή ΒΔ). Ο Διαχειριστής ΒΔ πρέπει να ελέγχει αυτό το αρχείο για να εντοπίζει ερωτήμα- τα με μεγάλο χρόνο εκτέλεσης καθώς αυτά τα ερωτήματα μπορεί να προκαλέσουν γενικότερες καθυστερήσεις στη λειτουργία του ΣΔΒΔ.

Ο Διαχειριστής ΒΔ πρέπει να αναλύσει το **πλάνο εκτέλεσης (execution plan)** των ερωτημάτων με με- γάλο χρόνο εκτέλεσης ώστε να εντοπίσει που οφείλονται οι καθυστερήσεις. Όλα τα ΣΔΒΔ διαθέτουν μια ειδική εντολή που παρουσιάζει το πλάνο εκτέλεσης ενός ερωτήματος. Στο ΣΔΒΔ MySQL αυτό γίνεται αν χρησιμοποι- ηθεί η εντολή **EXPLAIN** στην αρχή ενός ερωτήματος. Για παράδειγμα, με την εντολή

```
EXPLAIN SELECT *  
FROM Μαθητής  
WHERE Επώνυμο='Παπαδόπουλος' AND Όνομα='Ιωάννης';
```

θα εμφανιστούν πληροφορίες για τον τρόπο εκτέλεσης του ερωτήματος. Οι σημαντικότερες πληροφορίες είναι τα κλειδιά που θα χρησιμοποιηθούν για την εκτέλεση του ερωτήματος και μια εκτίμηση του πλήθους των γραμμών που θα ελέγξει το ΣΔΒΔ για να εντοπίσει τις γραμμές που περιλαμβάνει το αποτέλεσμα. Αν από το πλάνο εκτέλεσης προκύψει ότι ένα ερώτημα με μεγάλο χρόνο εκτέλεσης δεν χρησιμοποιεί κάποιο κλειδί, ο Διαχειριστής ΒΔ θα πρέπει να δημιουργήσει κατάλληλα ευρετήρια (όπως εξηγήθηκε σε προηγούμενη ενότητα) για να μειωθεί ο χρόνος εκτέλεσης.

Στις καθημερινές εργασίες του Διαχειριστή ΒΔ περιλαμβάνεται και η **δημιουργία αντιγράφου ασφαλείας (backup) της ΒΔ**. Τα αντίγραφα ασφαλείας μπορούν να χρησιμοποιηθούν για την **επαναφορά (restore) της ΒΔ σε προηγούμενη κατάσταση** σε περίπτωση καταστροφικού προβλήματος (π.χ. σφάλμα στο δίσκο του υπολογιστή που είναι εγκατεστημένο το ΣΔΒΔ, καταστροφή των αρχείων με τα περιεχόμενα της ΒΔ, ιός κ.λπ.). Επίσης η δημιουργία αντιγράφου ασφαλείας είναι επιβεβλημένη πριν ξεκινήσουν εργασίες αναβάθμισης λογισμικού (π.χ. λειτουργικό σύστημα, ΣΔΒΔ).

Στο ΣΔΒΔ MySQL η δημιουργία αντιγράφου ασφαλείας γίνεται με την εφαρμογή **mysqldump** που είναι διαθέσιμη σε περιβάλλον γραμμής εντολών του λειτουργικού συστήματος. Η εφαρμογή mysqldump δεν δημιουργεί **φυσικό αντίγραφο (physical backup)** της ΒΔ, δηλαδή δεν αντιγράφει τα αρχεία δεδομένων με τα περιεχόμενα της ΒΔ. Αντίθετα δημιουργεί **λογικό αντίγραφο (logical backup)**, δηλαδή παράγει εντολές SQL για να δημιουργηθούν ξανά η ΒΔ, οι πίνακες και τα ευρετήρια (CREATE DATABASE, CREATE TABLE, CREATE INDEX) και για να εισαχθούν τα δεδομένα στους πίνακες (INSERT). Το πλεονέκτημα του λογικού αντιγράφου είναι ότι το αρχείο που παράγεται είναι πολύ μικρό σε μέγεθος. Η εφαρμογή mysqldump πρέπει να εκτελεστεί στον φάκελο εγκατάστασης του ΣΔΒΔ MySQL. Για παράδειγμα, αν θέλουμε να δημιουργήσουμε αντίγραφο της ΒΔ mathitologio, θα εκτελέσουμε την ακόλουθη εντολή:

Σε περιβάλλον **Windows**

```
C:\mysql\bin\mysqldump -u root -p mathitologio > mathitologio_bak.sql
```

Σε περιβάλλον **Linux**

```
/etc/init.d/mysqldump -u root -p mathitologio > mathitologio_bak.sql
```

Με την επιλογή **-u** καθορίζεται ο χρήστης που θα εκτελέσει την εντολή (στο παράδειγμα ο χρήστης είναι ο root) και η επιλογή **-p** καθορίζει ότι θα ζητηθεί κωδικός πριν ξεκινήσει η διαδικασία. Το αποτέλεσμα της εντολής θα αποθηκευτεί στο αρχείο mathitologio_bak.sql (το σύμβολο **>** καθορίζει που θα αποθηκευτεί το αποτέλεσμα). Το αποτέλεσμα είναι ένα αρχείο κειμένου με τις απαραίτητες εντολές SQL για να δημιουργηθεί ξανά η ΒΔ.

Ένα ΣΔΒΔ μπορεί να φιλοξενεί περισσότερες από μια ΒΔ. Σε αυτή την περίπτωση ο Διαχειριστής ΒΔ μπορεί να δημιουργήσει αντίγραφο ασφαλείας για όλες τις ΒΔ χρησιμοποιώντας την επιλογή **all databases**. Για παράδειγμα, για να δημιουργήσουμε αντίγραφο όλων των ΒΔ θα εκτελέσουμε την επόμενη εντολή:

Σε περιβάλλον **Windows**

```
C:\mysql\bin\mysqldump -u root -p alldatabases > full_bak.sql
```

Σε περιβάλλον **Linux**

```
/etc/init.d/mysqldump -u root -p alldatabases > full_bak.sql
```

Το αρχείο full_bak.sql που θα προκύψει θα περιέχει τις αναγκαίες εντολές SQL για να δημιουργηθούν ξανά όλες οι ΒΔ που φιλοξενεί το ΣΔΒΔ.

Η επαναφορά αντιγράφου ασφαλείας στο ΣΔΒΔ MySQL γίνεται επίσης με την εφαρμογή mysqldump. Η σύνταξη είναι ακριβώς ίδια με τη λήψη αντιγράφου με μόνη διαφορά ότι χρησιμοποιείται το σύμβολο < για να καθορίσει από που θα γίνει η εισαγωγή των στοιχείων. Για παράδειγμα, για να επαναφέρουμε το αντίγραφο της ΒΔ mathitologio θα εκτελεστεί η εντολή:

Σε περιβάλλον **Windows**

```
C:\mysql\bin\mysqldump -u root -p new_mathitologio < mathitologio_bak.sql
```

Σε περιβάλλον **Linux**

```
/etc/init.d/mysqldump -u root -p new_mathitologio < mathitologio_bak.sql
```

Όπως φαίνεται, χρησιμοποιείται το αρχείο mathitologio_bak.sql που δημιουργήθηκε προηγουμένως, για να γίνει επαναφορά της ΒΔ mathitologio στη ΒΔ new_mathitologio. Ειδικότερα, οι εντολές που περιέχει το αρχείο mathitologio_bak.sql θα εκτελεστούν ώστε να δημιουργηθεί ξανά η ΒΔ. Η ΒΔ στην οποία θα γίνει η επαναφορά πρέπει να έχει δημιουργηθεί με την εντολή CREATE DATABASE πριν ξεκινήσει η διαδικασία.

Η επαναφορά αντιγράφου όλων των ΒΔ του ΣΔΒΔ, θα γίνει με την εντολή:

Σε περιβάλλον **Windows**

```
C:\mysql\bin\mysqldump -u root -p < full_bak.sql
```

Σε περιβάλλον **Linux**

```
/etc/init.d/mysqldump -u root -p < full_bak.sql
```

Όπως φαίνεται, χρησιμοποιείται το αρχείο full_bak.sql που δημιουργήθηκε προηγουμένως, για να γίνει επαναφορά όλων των ΒΔ που φιλοξενεί το ΣΔΒΔ. Σε αυτή την περίπτωση δεν χρειάζεται να δημιουργήσουμε από πριν τις ΒΔ καθώς το αρχείο full_bak.sql αφορά περισσότερες από μία ΒΔ και γι' αυτό περιέχει τις κατάλληλες εντολές CREATE DATABASE.

Όπως συμβαίνει με όλες τις εφαρμογές λογισμικού, έτσι και τα ΣΔΒΔ εξελίσσονται και οι κατασκευαστές τους παρουσιάζουν νεότερες εκδόσεις. Η νεότερη έκδοση μπορεί απλώς να επιλύει προβλήματα (λιγότερο ή περισσότερο σημαντικά) ή να προσφέρει επιπλέον δυνατότητες. Η απόφαση για **αναβάθμιση (upgrade)** σε νεότερη έκδοση του ΣΔΒΔ είναι πολύ σημαντική, ιδιαίτερα για ΣΔΒΔ που χρησιμοποιούνται σε παραγωγικό περιβάλλον, και θα πρέπει να λαμβάνεται αφού πρώτα μελετηθούν οι απαιτήσεις της νέας έκδοσης σε υλικό και λογισμικό και η συμβατότητα με παλαιότερες εκδόσεις. Αν αποφασιστεί αναβάθμιση, τότε θα πρέπει να ακολουθηθούν πιστά οι οδηγίες του κατασκευαστή του ΣΔΒΔ, κυρίως για τον τρόπο μεταφοράς των ΒΔ στη νεότερη έκδοση του ΣΔΒΔ. Η διαδικασία της αναβάθμισης είναι αρμοδιότητα του Διαχειριστή ΒΔ. Η αναβάθμιση θα πρέπει πρώτα να δοκιμαστεί σε ΣΔΒΔ που δεν φιλοξενεί τις πραγματικές ΒΔ, ώστε να ελεγχθεί η διαδικασία και να εντοπιστούν πιθανά προβλήματα. Εφόσον η δοκιμή πετύχει τότε μπορεί να προχωρήσει η αναβάθμιση και του ΣΔΒΔ που χρησιμοποιείται σε παραγωγικό περιβάλλον.

Πριν την αναβάθμιση θα πρέπει ο Διαχειριστής ΒΔ να δημιουργήσει αντίγραφο ασφάλειας του ΣΔΒΔ και των ΒΔ που φιλοξενεί, ώστε αν παρουσιαστεί πρόβλημα να μπορέσει άμεσα να επαναφέρει το ΣΔΒΔ στην προηγούμενη κατάσταση.

Μετά την ολοκλήρωση της αναβάθμισης και πριν τεθεί το ΣΔΒΔ σε λειτουργία, ο Διαχειριστής ΒΔ χρησιμοποιεί ειδικά εργαλεία για να εξασφαλίσει τη σωστή λειτουργία των ΒΔ στην νεότερη έκδοση του ΣΔΒΔ. Στο ΣΔΒΔ MySQL μετά την αναβάθμιση πρέπει να εκτελεστεί σε περιβάλλον γραμμής εντολών του λειτουργικού συστήματος η εφαρμογή **mysql_upgrade**. Η εφαρμογή αυτή επιδιορθώνει τη δομή των πινάκων και των ευρετηρίων ώστε να εξασφαλιστεί η συμβατότητα με τη νεότερη έκδοση του ΣΔΒΔ.



Ερωτήσεις / Δραστηριότητες

1. Χρήση αρχείων καταγραφής

- Αν κατά την εγκατάσταση του ΣΔΒΔ MySQL δεν είχατε επιλέξει τη χρήση αρχείων καταγραφής, ξεκινήστε το ΣΔΒΔ με την ακόλουθη εντολή ώστε να ενεργοποιηθούν τα αρχεία καταγραφής: **mysqld --general-log --log-error --slow-query-log**.
- Συνδεθείτε στο ΣΔΒΔ MySQL και υποβάλετε μερικά ερωτήματα.
- Εντοπίστε το αρχείο καταγραφής σφαλμάτων (**.err**), το αρχείο καταγραφής ερωτημάτων (**.log**) και το αρχείο καταγραφής αργών ερωτημάτων (**-slow.log**) και δείτε τα περιεχόμενά τους με έναν απλό επεξεργαστή κειμένου (π.χ. Notepad σε Windows, GEdit σε Linux). Ελέγξτε το αρχείο καταγραφής ερωτημάτων και εντοπίστε εγγραφές που σχετίζονται με τις ενέργειες που κάνατε στο ΣΔΒΔ. Ελέγξτε το αρχείο καταγραφής σφαλμάτων για σφάλματα στη λειτουργία του ΣΔΒΔ. Ελέγξτε το αρχείο καταγραφής αργών ερωτημάτων για ερωτήματα με μεγάλο χρόνο εκτέλεσης.

2. Δημιουργία και επαναφορά αντιγράφου ασφάλειας

- Δημιουργήστε αντίγραφο ασφάλειας της ΒΔ mathitologio.
- Χρησιμοποιήστε έναν απλό επεξεργαστή κειμένου για να δείτε τα περιεχόμενα του αρχείου που παράχθηκε κατά τη δημιουργία του αντιγράφου ασφαλείας.
- Δημιουργήστε μια νέα ΒΔ. Χρησιμοποιήστε το αντίγραφο ασφαλείας που δημιουργήσατε προηγουμένως για να επαναφέρετε τη ΒΔ mathitologio στη νέα ΒΔ.

5.2 Εργαλεία Διαχείρισης ΒΔ



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να κατανοεί πως η χρήση εργαλείων διαχείρισης διευκολύνει το έργο του Διαχειριστή ΒΔ
- να απαριθμεί τις βασικές λειτουργίες που προσφέρει ένα εργαλείο διαχείρισης ΒΔ
- να γνωρίζει τα πιο διαδεδομένα εργαλεία διαχείρισης ΒΔ
- να αξιολογεί τα εργαλεία διαχείρισης ΒΔ και να επιλέγει το καταλληλότερο

Οι εργασίες διαχείρισης της ΒΔ που παρουσιάστηκαν στις προηγούμενες ενότητες, πραγματοποιήθηκαν με εντολές SQL ή με εντολές σε περιβάλλον γραμμής εντολών του λειτουργικού συστήματος. Παρόλο που ένας Διαχειριστής ΒΔ θα πρέπει να έχει τη δυνατότητα να πραγματοποιεί όλες τις εργασίες διαχείρισης με αυτόν τον τρόπο, υπάρχουν ειδικές εφαρμογές λογισμικού που κάνουν αυτές τις εργασίες ευκολότερες.



Ένα **εργαλείο διαχείρισης ΒΔ** διευκολύνει το έργο του Διαχειριστή ΒΔ παρέχοντας ένα γραφικό περιβάλλον εργασίας (GUI – graphical user interface) μέσω του οποίου μπορούν να πραγματοποιηθούν διάφορες εργασίες διαχείρισης της ΒΔ. Οι λειτουργίες που συνήθως παρέχει ένα εργαλείο διαχείρισης ΒΔ είναι:

- δημιουργία ΒΔ,
- ρύθμιση παραμέτρων του ΣΔΒΔ και των ΒΔ,
- δημιουργία και τροποποίηση πινάκων και ευρετηρίων,
- υποβολή ερωτημάτων SQL (ανάκτηση, εισαγωγή, ενημέρωση, διαγραφή) με γραφικό τρόπο,
- σχηματική απεικόνιση των σχέσεων μεταξύ των πινάκων,
- δημιουργία χρηστών και διαχείριση προνομίων,
- εισαγωγή και εξαγωγή δεδομένων,
- δημιουργία και επαναφορά αντιγράφων ασφαλείας,
- παρακολούθηση λειτουργίας του ΣΔΒΔ και των ΒΔ,
- επιδιόρθωση περιεχομένων ΒΔ.

Εργαλεία διαχείρισης ΒΔ αναπτύσσονται συνήθως από τους κατασκευαστές ΣΔΒΔ και περιλαμβάνονται στο πρόγραμμα εγκατάστασης του ΣΔΒΔ. Υπάρχουν όμως και εργαλεία διαχείρισης ΒΔ που έχουν αναπτυχθεί από τρίτους.



Τα πιο διαδεδομένα εργαλεία διαχείρισης για το ΣΔΒΔ MySQL είναι το **MySQL Workbench** και το **phpMyAdmin**. Άλλα γνωστά εργαλεία είναι τα Toad for MySQL, HeidiSQL και Sequel Pro.

(Όλα τα παραπάνω εργαλεία διατίθενται δωρεάν.)



Εικόνα 5.2
Εργαλεία διαχείρισης
για το ΣΔΒΔ MySQL:
MySQL Workbench
(αριστερά),
phpMyAdmin (δεξιά)



[Πηγή: <http://dev.mysql.com/doc/workbench/en/wbperformancedashboard.html>, <https://www.phpmyadmin.net/try/>]

Τα εργαλεία διαχείρισης ΒΔ μπορούν να κατηγοριοποιηθούν σύμφωνα με διάφορα κριτήρια.

Κόστος: Κάποια εργαλεία είναι **δωρεάν** και κάποια είναι **επί πληρωμή**. Συνήθως για τα εργαλεία επί πληρωμή διατίθενται και δωρεάν δοκιμαστικές εκδόσεις (με περιορισμούς στο χρόνο χρήσης ή στις προσφερόμενες λειτουργίες).

Άδεια χρήσης: Υπάρχουν εργαλεία **ελεύθερου/ανοιχτού κώδικα (free/open source)** στα οποία ο κώδικας της εφαρμογής είναι διαθέσιμος στο κοινό, σε ορισμένες περιπτώσεις ακόμα και για τροποποίηση. Από την άλλη υπάρχουν εργαλεία **κλειστού κώδικα (closed source ή proprietary)** που ο κώδικας δεν είναι διαθέσιμος και δεν επιτρέπεται σε κανέναν εκτός από τον κατασκευαστή να τον τροποποιήσει.

Υποστηριζόμενα λειτουργικά συστήματα: Ορισμένα εργαλεία είναι διαθέσιμα μόνο για συγκεκριμένο λειτουργικό σύστημα (π.χ. Windows, UNIX/Linux, MacOS), ενώ άλλα είναι **ανεξάρτητα πλατφόρμας (crossplatform)**, δηλαδή υπάρχουν εκδόσεις για περισσότερα λειτουργικά συστήματα.

Υποστηριζόμενα ΣΔΒΔ: Μερικά εργαλεία συνεργάζονται μόνο με συγκεκριμένο ΣΔΒΔ (π.χ. MySQL, Oracle Database, MS SQL Server, IBM DB2), ενώ άλλα συνεργάζονται με περισσότερα ΣΔΒΔ.

Τύπος εφαρμογής: Υπάρχουν εργαλεία που είναι **εφαρμογές πελάτη (client application)**, δηλαδή η εγκατάστασή τους γίνεται στον υπολογιστή του Διαχειριστή ΒΔ και συνδέονται μέσω δικτύου στον υπολογιστή που είναι εγκατεστημένο το ΣΔΒΔ. Υπάρχουν όμως και εργαλεία που είναι **εφαρμογές εξυπηρετητή (server application)**, δηλαδή η εγκατάστασή τους γίνεται σε κάποιον εξυπηρετητή (συνήθως σε αυτόν που είναι εγκατεστημένο και το ΣΔΒΔ) και ο Διαχειριστής ΒΔ χρησιμοποιεί ένα φυλλομετρητή Διαδικτύου (web browser) για να συνδεθεί στο εργαλείο.

Στον επόμενο πίνακα συγκρίνονται σύμφωνα με τα παραπάνω κριτήρια τα πιο διαδεδομένα εργαλεία διαχείρισης για διάφορα ΣΔΒΔ.

Πίνακας 5.2 Σύγκριση εργαλείων διαχείρισης ΒΔ

Ονομασία εργαλείου	Κόστος	Άδεια χρήσης	Λειτουργικό σύστημα	ΣΔΒΔ	Τύπος εφαρμογής
MySQL Workbench	Δωρεάν	Ελεύθερο/ ανοιχτού κώδικα	Windows, Linux, MacOS	MySQL	Πελάτης
phpMyAdmin	Δωρεάν	Ελεύθερο/ ανοιχτού κώδικα	Windows, Linux	MySQL	Εξυπηρετητής
Toad	Δωρεάν για MySQL, επί πληρωμή για άλλα ΣΔΒΔ	Κλειστού κώδικα	Windows	MySQL, Oracle, MS SQL Server, IBM DB2, Sybase	Πελάτης
Oracle Enterprise Manager Database Management	Επί πληρωμή	Κλειστού κώδικα	Windows, UNIX, Linux	Oracle	Εξυπηρετητής
MS SQL Server Management Studio	Επί πληρωμή	Κλειστού κώδικα	Windows	MS SQL Server	Πελάτης
IBM Data Server Manager	Επί πληρωμή	Κλειστού κώδικα	Windows, UNIX, Linux	IBM DB2	Εξυπηρετητής



ΧΡΗΣΙΜΕΣ ΔΙΕΥΘΥΝΣΕΙΣ ΔΙΑΔΙΚΤΥΟΥ

Δωρεάν εργαλεία διαχείρισης για το ΣΔΒΔ MySQL

MySQL Workbench: <http://dev.mysql.com/downloads/workbench/> phpMyAdmin: <https://www.phpmyadmin.net/>

Δοκιμαστική εγκατάσταση phpMyAdmin: http://demo.phpmyadmin.net/master_config/

Toad for MySQL: <http://software.dell.com/products/toadformysql/> HeidiSQL: <http://www.heidisql.com/>

Sequel Pro: <http://www.sequelpro.com/>



Ερωτήσεις / Δραστηριότητες

1. Επισκεφθείτε τη σελίδα της Wikipedia

https://en.wikipedia.org/wiki/Comparison_of_database_tools στην οποία γίνεται σύγκριση εργαλείων διαχείρισης ΒΔ. Με βάση τα στοιχεία της Wikipedia, δημιουργήστε αντίστοιχους πίνακες για τα εργαλεία που υποστηρίζουν το ΣΔΒΔ MySQL. Μελετήστε τους πίνακες που δημιουργήσατε και συζητήστε εάν τα εργαλεία επί πληρωμή πλεονεκτούν σημαντικά έναντι των δωρεάν εργαλείων.

2. Συνδεθείτε στο εργαλείο διαχείρισης **phpMyAdmin** (εάν δεν υπάρχει δυνατότητα εγκατάστασής του, επισκεφθείτε τη σελίδα http://demo.phpmyadmin.net/master_config/ με τη δοκιμαστική εγκατάσταση που προσφέρουν οι κατασκευαστές του εργαλείου). Εξετάστε τις δυνατότητες που προσφέρει το εργαλείο και πραγματοποιήστε κάποιες ενέργειες. Συζητήστε τις ευκολίες που παρέχει το εργαλείο συγκριτικά με την πραγματοποίηση των αντίστοιχων ενεργειών σε περιβάλλον γραμμής εντολών.

κεφάλαιο

6

Βάσεις Δεδομένων στο Διαδίκτυο

6.1 Νέες Ανάγκες και NoSQL Βάσεις Δεδομένων



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- Ο μαθητής/τρια πρέπει να είναι ικανός/ή:
- να αναγνωρίζει την αναγκαιότητα των NoSQL ΒΔ
- να αναγνωρίζει τα χαρακτηριστικά των NoSQL βάσεων δεδομένων
- να αναφέρει πλεονεκτήματα και μειονεκτήματα στη χρήση των NoSQL ΒΔ

Οι σχεσιακές βάσεις δεδομένων έχουν καθιερωθεί και χρησιμοποιούνται για την αποθήκευση δεδομένων για δεκαετίες, ενώ η SQL είναι η γλώσσα που επικράτησε να χρησιμοποιείται για την αλληλεπίδραση με αυτές. Την τελευταία δεκαετία όμως, εμφανίστηκε η ανάγκη για ανάπτυξη εφαρμογών, στο διαδίκτυο κυρίως, των οποίων οι απαιτήσεις δεν καλύπτονταν σε ικανοποιητικό βαθμό από τις σχεσιακές βάσεις δεδομένων. Οι εφαρμογές αυτές (μπορούν να χαρακτηριστούν εφαρμογές ευρείας κλίμακας *large scale applications*) έχουν την ανάγκη αποθήκευσης και επεξεργασίας μεγάλων ποσοτήτων δεδομένων. Επίσης, η χρήση τους γίνεται από μεγάλους πληθυσμούς χρηστών, οι οποίοι μπορεί να είναι διεσπαρμένοι σε όλο τον πλανήτη και είναι ιδιαίτερα απαιτητικοί όσον αφορά την ταχύτητα απόκρισης των συστημάτων αυτών. Οι σχεσιακές βάσεις δεδομένων δεν είναι σχεδιασμένες να αντιμετωπίζουν τα μεγέθη που χρειάζονται να διαχειριστούν οι σύγχρονες εφαρμογές αυτού του τύπου, δυσκολεύονται να ανταπεξέλθουν στις προκλήσεις σε ευελιξία (*agility*) που απαιτούνται και δεν μπορούν εύκολα να επωφεληθούν από τη φθηνή αποθηκευτική και επεξεργαστική ισχύ που είναι διαθέσιμες σήμερα.

Η προσπάθεια κάλυψης των αναγκών αυτών δημιούργησε και έφερε στο προσκήνιο μια νέα ομάδα βάσεων δεδομένων, με διαφορετικά χαρακτηριστικά από τις σχεσιακές, μέλη της οποίας καθιερώνονται όλο και περισσότερο. Η ομάδα των βάσεων δεδομένων αυτών ονομάστηκε **NoSQL (Not Only SQL)** και περιλαμβάνει μια ποικιλία από διαφορετικές τεχνολογίες που αναπτύχθηκαν σαν απόκριση στις νέες απαιτήσεις.

Η εμφάνιση των NoSQL βάσεων είναι σχετικά πρόσφατη και, ως εκ τούτου, είναι δύσκολο να καταλήξουμε σε ένα περιεκτικό ορισμό του τι περιλαμβάνει η τεχνολογία. Σε γενικές γραμμές, μπορεί να θεωρηθεί ως μία οικογένεια τεχνολογιών βάσεων δεδομένων που αναπτύσσονται με σκοπό το αποδοτικό χειρισμό μεγάλων ποσοτήτων αδόμητων και ημιδομημένων δεδομένων. Εταιρείες όπως η Google, η Amazon, και η Facebook κάνουν εκτεταμένη χρήση της τεχνολογίας αυτής για να στεγάσουν τις τεράστιες μάζες δεδομένων που παράγονται από τους χρήστες τους.

Για την απόφαση όμως χρήσης μιας τέτοιας βάσης αντί μιας σχεσιακής, θα πρέπει πρώτα να μελετηθούν με λεπτομέρεια οι ανάγκες της εφαρμογής που θα αναπτυχθεί και να σταθμιστούν τα όποια πλεονεκτήματα και μειονεκτήματα. Οι σχεσιακές βάσεις δεδομένων συνεχίζουν να είναι μια αξιόπιστη και αποδοτική λύση σε μια μεγάλη ομάδα εφαρμογών, ενώ οι καινοτομίες που εισαγάγουν οι NoSQL βάσεις μπορεί να δίνουν λύσεις στις ανάγκες μιας σειράς μοντέρνων εφαρμογών, αλλά δεν τις καθιστούν καταλληλότερες επιλογές για τις εφαρμογές που ήδη, για χρόνια, καλύπτονται άριστα από τις σχεσιακές βάσεις.

6.1.1 Χαρακτηριστικά NoSQL Βάσεων Δεδομένων

Σχετικά με ότι έχει αναφερθεί έως τώρα, το βασικότερο πεδίο εφαρμογών που διακρίνονται οι βάσεις NoSQL είναι αυτό των δικτυακών εφαρμογών μεγάλου εύρους. Μήπως αυτό σημαίνει ότι θα διερευνούμε την αναγκαιότητα χρήσης NoSQL βάσεων δεδομένων μόνο στις περιπτώσεις που μας απασχολούν προβλήματα κλίμακας της εφαρμογής; Φυσικά όχι! Οι NoSQL βάσεις δεδομένων έχουν πολλά περισσότερα να προσφέρουν πέρα από την επίλυση των προβλημάτων της κλίμακας.

Πιο συγκεκριμένα, όφελος από την χρήση τους μπορούμε να έχουμε εκμεταλλευόμενοι και τα παρακάτω χαρακτηριστικά τους:

Αδόμητη ή ημιδομημένη αναπαράσταση δεδομένων (schemaless data representation): Η πλειοψηφία των NoSQL βάσεων ακολουθούν αδόμητη ή ημιδομημένη αναπαράσταση δεδομένων. Αυτό σημαίνει ότι δεν χρειάζεται να προβλέψουμε το μοντέλο δεδομένων που θα χρειαστεί η εφαρμογή μας στο μακρινό μέλλον, αλλά να καθορίσουμε την δομή που είναι κατάλληλη στο παρόν και να την εξελίσσουμε στο χρόνο ανάλογα με τις ανάγκες μας, όπως για παράδειγμα να προσθέτουμε νέα “πεδία” μόνο στις “εγγραφές” που αυτό είναι απαραίτητο.

Ταχύτητα ανάπτυξης εφαρμογής: Για την χρήση μιας NoSQL βάσης δεν χρειάζεται πάντα η γνώση μιας ξεχωριστής γλώσσας, όπως η SQL, αλλά μας διατίθενται βιβλιοθήκες (APIs) η χρήση των οποίων ακολουθεί την μορφή της γλώσσας ανάπτυξης της εφαρμογής (π.χ. javascript, ή php) διευκολύνοντας έτσι τον προγραμματιστή.

Ταχύτητα απόκρισης: Η τεχνολογία ανάπτυξης των NoSQL βάσεων δεδομένων, έχοντας σαν σκοπό να καλύψει τις ανάγκες του μεγάλου αριθμού πιθανών χρηστών των διαδικτυακών εφαρμογών, πετυχαίνει ταχύτατους χρόνους απόκρισης, κάτι που κάνει την εμπειρία χρήσης αυτών των εφαρμογών περισσότερο επιτυχημένη.

Προσαρμοστικότητα στον φόρτο: Τα περισσότερα συστήματα διαχείρισης NoSQL βάσεων δεδομένων είναι σχεδιασμένα με τρόπο ώστε, χωρίς να χρειάζονται ιδιαίτερες ρυθμίσεις και πρόβλεψη διαθεσιμότητας σε υποδομές, να ανταποκρίνονται άμεσα σε μεγάλες αλλαγές στις απαιτήσεις των χρηστών. Έτσι δεν υπάρχει μεγάλη δυσκολία (και είναι οικονομικό) να αναπτύσσουμε εφαρμογές, οι οποίες να λειτουργούν το ίδιο καλά με μικρούς αλλά και μεγάλους αριθμούς χρηστών, ακόμα και αν αυτοί οι αριθμοί αλλάζουν απότομα από στιγμή σε στιγμή. Με άλλα λόγια οι NoSQL βάσεις μπορούν αν διαχειρίζονται ικανοποιητικά ξαφνικές αιχμές του φόρτου.

Η παραπάνω λίστα με τα θετικά χαρακτηριστικά των NoSQL βάσεων φυσικά δεν είναι με κανένα τρόπο πλήρης. Ένα επιπλέον θετικό σημείο που πρέπει να προσθέσουμε είναι ότι οι περισσότερες από τις NoSQL βάσεις δεδομένων είναι ΕΛ/ΛΑΚ και η εξέλιξή τους υποστηρίζεται από την παγκόσμια κοινότητα προγραμματιστών.

6.1.2 Τύποι (Κατηγορίες) NoSQL Βάσεων Δεδομένων

Οι NoSQL βάσεις δεδομένων κατηγοριοποιούνται ανάλογα με τον τρόπο που αποθηκεύονται τα δεδομένα. Λόγω της ανάγκης να παρέχουν πληροφορίες που προέρχονται από επεξεργασία μεγάλων ποσοτήτων δεδομένων σε πραγματικό χρόνο, στις περισσότερες περιπτώσεις οι NoSQL βάσεις ακολουθούν μια «οριζόντια δομή». Οι συχνότεροι τύποι που χρησιμοποιούνται αφορούν αποθήκευση σε:

- στήλες (Column-oriented)
- έγγραφα (Document Stores)
- ζεύγη κλειδί – τιμή (Key Value Stores)
- γράφους (Graph)

Στηλοκεντρικές (Columnoriented) βάσεις δεδομένων

Οι στηλοκεντρικές βάσεις δεδομένων αποθηκεύουν τα δεδομένα σε στήλες, σε αντίθεση με τις σχεσιακές που τα αποθηκεύουν σε γραμμές. Πιο συγκεκριμένα, οι σχεσιακές βάσεις αναπαριστούν τα δεδομένα σε διάστατους πίνακες που αποτελούνται από γραμμές και στήλες αλλά τα αποθηκεύουν, τα ανακτούν και τα επεξεργάζονται ανά γραμμή, ενώ αντίστοιχα οι στηλοκεντρικές ανά στήλη.

Για παράδειγμα, ας θεωρήσουμε ότι θέλουμε να αποθηκεύσουμε τα παρακάτω δεδομένα μαθητών:

Κωδικός	Όνομα	Επώνυμο	Γέννηση	Βαθμός
A1	Ιωάννης	Παπαδάκης	12/11/1998	18.2
B4	Μαρία	Μανουσάκη	11/02/1997	19.5
A8	Γεώργιος	Γνώστης	02/05/1998	17.8
E4	Αγγελική	Κοτσιφού	03/05/1999	16.9

σε μια σχεσιακή βάση τα δεδομένα θα αποθηκεύονταν εσωτερικά στην παρακάτω μορφή:

```
A1, Ιωάννης, Παπαδάκης, 12/11/1998, 18.2
B4, Μαρία, Μανουσάκη, 11/02/1997, 19.5
A8, Γεώργιος, Γνώστης, 02/05/1998, 17.8
E4, Αγγελική, Κοτσιφού, 03/05/1999, 16.9
```

ενώ σε μια στηλοκεντρική:

```
A1, B4, A8, E4
Ιωάννης, Μαρία, Γεώργιος, Αγγελική
Παπαδάκης, Μανουσάκη, Γνώστης, Κοτσιφού
12/11/1998, 11/02/1997, 02/05/1998, 03/05/1999
18.2, 19.5, 17.8, 16.9
```

Οι προηγούμενες αναπαραστάσεις εσωτερικής αποθήκευσης είναι απλουστευμένες, αλλά μας βοηθούν να καταλάβουμε καλύτερα σε επίπεδο μοντέλου τις διαφορές των βάσεων αυτών. Συνήθως τα συστήματα διαχείρισης βάσεων δεδομένων έχουν εξελιγμένους και βελτιστοποιημένους μηχανισμούς για την αποθήκευση των δεδομένων τους ώστε να λειτουργούν αποδοτικά εργασίες όπως η τμηματοποίηση, η προσωρινή αποθήκευση και η δεικτοδότηση (partitioning, caching, indexing).

Στις περιπτώσεις εισαγωγής νέων οντοτήτων στο σύστημα οι στηλοκεντρικές βάσεις δεδομένων είναι λιγότερο αποδοτικές από τις σχεσιακές μια που θα χρειαστεί ενημέρωση όλων των γραμμών της βάσης. Σε περίπτωση όμως που χρειάζεται να προσθέσουμε στο μοντέλο μας ένα νέο πεδίο, αυτό θα γίνει πολύ εύκολα και αποδοτικά, με την προσθήκη μόνο μίας γραμμής στην βάση.

Επίσης, οι στηλοκεντρικές βάσεις έχουν μεγάλο πλεονέκτημα σε απόδοση στις περιπτώσεις επεξεργασίας υποσυνόλου των πεδίων της βάσης, όπως για παράδειγμα σε πράξεις υπολογισμού μέγιστου, ελάχιστου, μέσου όρου και αθροίσματος ιδιαίτερα σε πολύ μεγάλα σύνολα δεδομένων.

Χαρακτηριστικές στηλοκεντρικές βάσεις είναι οι:

BigTable <https://cloud.google.com/bigtable/>

Hadoop/HBase <http://hbase.apache.org/>

Cassandra <http://cassandra.apache.org/>

SimpleDB <http://aws.amazon.com/simplydb/>

Cloudera <http://www.cloudera.com/>

Αποθήκες εγγράφων (Document Stores)

Οι αποθήκες εγγράφων, οι οποίες αναφέρονται και σαν εγγραφοκεντρικές βάσεις δεδομένων (document-oriented databases), επιτρέπουν την εισαγωγή, ανάκτηση και επεξεργασία ημιδομημένων δεδομένων. Οι περισσότερες βάσεις της συγκεκριμένης κατηγορίας χρησιμοποιούν JSON (JavaScript Object Notation <https://en.wikipedia.org/wiki/JSON>), BSON (Binary JSON <https://en.wikipedia.org/wiki/BSON>), και XML (Extensible Markup Language <https://en.wikipedia.org/wiki/XML>) μορφές αποθήκευσης και η πρόσβαση στα δεδομένα γίνεται συνήθως μέσω πρωτοκόλλου HTTP με την χρήση RESTful διεπαφών (). Τα έγγραφα των εγγραφοκεντρικών βάσεων είναι αντίστοιχα των εγγράφων (ή γραμμών) των σχεσιακών, αλλά είναι ημιδομημένα σε σύγκριση με τις εγγραφές.

Για παράδειγμα, δύο έγγραφα (εγγραφές) μπορούν να έχουν διαφορετικά σύνολα πεδίων (στηλών). Ακόμη και αν τα έγγραφα που δεν ακολουθούν ένα αυστηρό σχήμα, ευρετήρια μπορεί να δημιουργηθούν και να χρησιμοποιούνται στις ερωτήσεις. Ακολουθεί η πρώτη εγγραφή από το παράδειγμα της προηγούμενης ενότητας σε αναπαράσταση εγγράφου σε μορφή JSON:

```
{
  "Κωδικός": "A1",
  "Όνομα": "Ιωάννης",
  "Επώνυμο": "Παπαδόκης",
  "Γέννηση": "12/11/1998",
  "Βαθμός": "18.2"
}
```

ένα δεύτερο έγγραφο μπορεί να μην έχει όλες τις πληροφορίες του προηγούμενου (δεν έχει καθοριστεί ακόμα βαθμός):

```
{
  "Κωδικός": "B4",
  "Όνομα": "Μαρία",
  "Επώνυμο": "Μανουσάκη",
  "Γέννηση": "11/02/1997"
}
```

ενώ ένα τρίτο έγγραφο μπορεί να έχει περισσότερα πεδία, κάποια από τα οποία μπορεί περιέχουν άλλα εμφω-
λευμένα (Διεύθυνση) ή να είναι και πίνακες (Απουσίες):

```
{
  "Κωδικός": "Α8",
  "Όνομα": "Γεώργιος",
  "Επώνυμο": "Γνώστης",
  "Γέννηση": "02/05/1998",
  "Βαθμός": "17.8",
  "Διεύθυνση":
  {
    "Οδός": "Δημοκρατίας 23",
    "ΤΚ": "73100",
    "Πόλη": "Χανιά"
  }
  "ΜηνιαίεςΑπουσίες": [
    { "Μήνας": "Οκτώβριος", "Απουσίες": "12" },
    { "Μήνας": "Απρίλιος", "Απουσίες": "6" },
    { "Μήνας": "Μάιος", "Απουσίες": "18" },
  ]
}
```

Όπως θα παρατηρήσατε στα προηγούμενα παραδείγματα, τα πρώτα 2 έγγραφα είναι σχεδόν τα ίδια (στο δεύτερο δεν έχει καθοριστεί ακόμα ο "Βαθμός"), αλλά το τρίτο περιέχει ένα σύνθετο πεδίο (την "Διεύθυνση") και ένα πίνακα από αντικείμενα (τις "ΜηνιαίεςΑπουσίες").

Οι εγγραφοκεντρικές βάσεις παρέχουν αυτή την ευελιξία (δυναμικό σχήμα) και δεδομένης της απεριόριστης ευελιξίας που αυτό προσφέρει, τις καθιστούν μια από τις πιο δημοφιλείς επιλογές που χρησιμοποιούνται στις σημερινές διαδικτυακές εφαρμογές, όπου υπάρχει η ανάγκη αποθήκευσης διαφορετικών τύπων δεδομένων των οποίων τα χαρακτηριστικά αλλάζουν με τον καιρό.

Αν και η εξέλιξη στην τεχνολογία των εγγραφοκεντρικών βάσεων δεν έχει φτάσει ακόμα στο αποκορύφω-
μά της, υπάρχουν ήδη αρκετές ώριμες και δημοφιλής λύσεις διαθέσιμες στην αγορά, οι χαρακτηριστικότερες των οποίων είναι οι:

MongoDB <https://www.mongodb.org/>

CouchDB <http://couchdb.apache.org/>

RavenDB <http://ravendb.net/>

Terrastore <https://code.google.com/p/terrastore/>

BaseX <http://basex.org/>

eXist <http://exist-db.org/exist/apps/homepage/index.html/>

Αποθήκες ζευγών κλειδιών-τιμών (Key Value Stores)

Τις Αποθήκες ζευγών κλειδιών-τιμών μπορούμε να τις φανταστούμε σαν βάσεις στις οποίες υπάρχει ένας μόνο πίνακας με δύο μόνο στήλες. Η πρώτη στήλη αφορά τα κλειδιά και η δεύτερη τις τιμές. Κάθε κλειδί είναι μοναδικό. Ευρετήριο (index) έχουμε μόνο σε επίπεδο κλειδιών. Οι τιμές μπορούν να είναι διαφόρων τύπων και φυσικά δεν υπάρχει η έννοια του σχήματος σε αυτό τον τύπο βάσεων.

Η κύρια χρήση τους είναι για υλοποίηση συστημάτων προσωρινής μνήμης (caching).

Για να γίνει καλύτερα κατανοητό το μοντέλο αποθήκευσης που χρησιμοποιούν οι αποθήκες ζευγών κλειδιών-τιμών, παρακάτω παρουσιάζουμε πως τα μπορούσε να αναπαρασταθεί σε μια τέτοια βάση το παράδειγμα της προηγούμενης ενότητας:

Κλειδί	Τιμή
A1_Όνομα	Ιωάννης
A1_Επώνυμο	Παπαδάκης
A1_Γέννηση	12/11/1998
A1_Βαθμός	18.2
B4_Όνομα	Μαρία
B4_Επώνυμο	Μανουσάκη
B4_Γέννηση	11/2/1997
A8_Όνομα	Γεώργιος
A8_Επώνυμο	Γνώστης
A8_Γέννηση	02/05/1998
A8_Βαθμός	17.8
A8_Διεύθυνση_Οδός	Δημοκρατίας 23
A8_Διεύθυνση_TK	73100
A8_Διεύθυνση_Πόλη	Χανιά
A8_ΜηνιαίεςΑπουσίες_Οκτώβριος	12
A8_ΜηνιαίεςΑπουσίες_Απρίλιος	6
A8_ΜηνιαίεςΑπουσίες_Μάιος	18

Χαρακτηριστικές βάσεις της συγκεκριμένης κατηγορίας είναι οι:

Redis <http://redis.io/>

Membase <http://www.couchbase.com/>

MemcacheDB <http://memcachedb.org/>

Βάσεις δεδομένων γράφων (graph)

Οι βάσεις δεδομένων γράφων αντιπροσωπεύουν μια ειδική κατηγορία noSQL βάσεων όπου οι σχέσεις μεταξύ των οντοτήτων αναπαριστώνται ως γράφοι. Μπορούν να υπάρχουν πολλαπλές συνδέσεις ανάμεσα σε δύο κόμβους του γράφου, δίνοντας την δυνατότητα αναπαράστασης πολλαπλών σχέσεων μεταξύ των οντοτήτων που συνδέονται.

Για παράδειγμα, οι σχέσεις που αναπαριστώνται ενδέχεται να περιλαμβάνουν κοινωνικές σχέσεις μεταξύ των ανθρώπων, σύνδεση με μεταφορικά μέσα μεταξύ περιοχών, ή τοπολογίες δικτύου μεταξύ των συνδεδεμένων συστημάτων.

Συχνό φαινόμενο σε διαδικτυακές εφαρμογές είναι για την αποθήκευση των δεδομένων να χρησιμοποιείται μια εγγραφοκεντρική βάση, ενώ για την αποθήκευση των σχέσεων των οντοτήτων να χρησιμοποιείται μία βάση γράφου.

Οι βάσεις δεδομένων γράφων είναι αρκετά νέες στην αγορά, οπότε διατίθενται μόνο λίγες δοκιμασμένες λύσεις προς χρήση:

Neo4J <http://neo4j.com/>

FlockDB <https://github.com/twitter/flockdb>

InfiniteGraph <http://www.objectivity.com/products/infinitegraph/>



Ερωτήσεις / Δραστηριότητες

1. Με την ευρύτατη διάδοση του διαδικτύου, ποιες ήταν οι ανάγκες που ώθησαν την ανάπτυξη νέων βάσεων δεδομένων;
2. Ποια είναι τα βασικά χαρακτηριστικά των NoSQL βάσεων δεδομένων τα οποία τις καθιστούν ελκυστική επιλογή για την ανάπτυξη διαδικτυακών εφαρμογών;
3. Αναφέρετε τις βασικές κατηγορίες NoSQL βάσεων δεδομένων.
4. Αναφέρετε δύο χαρακτηριστικές στηλοκεντρικές βάσεις δεδομένων.
5. Αναφέρετε ένα παράδειγμα στο οποίο οι στηλοκεντρικές βάσεις δεδομένων υπερτερούν των σχεσιακών και εξηγήστε γιατί.
6. Αναφέρετε δύο χαρακτηριστικές εγγραφοκεντρικές βάσεις δεδομένων.
7. Ποιο είναι το βασικό χαρακτηριστικό που δίνει την αναγκαία ευελιξία στις εγγραφοκεντρικές βάσεις δεδομένων και τις καθιστά μια από τις πιο δημοφιλείς επιλογές που χρησιμοποιούνται στις σημερινές διαδικτυακές εφαρμογές και γιατί;
8. Αναφέρετε δύο χαρακτηριστικές βάσεις δεδομένων τύπου Αποθήκες ζευγών κλειδιών-τιμών.
9. Ποια είναι η πιο συνηθισμένη χρήση των βάσεων δεδομένων τύπου Αποθήκες ζευγών κλειδιών-τιμών;
10. Περιγράψτε τις δύο βασικές κατηγορίες στις οποίες διακρίνονται οι αρχιτεκτονικές των υπολογιστικών συστημάτων ως προς την διασύνδεση των χρηστών με το σύστημα.

6.2 Αρχιτεκτονική Εφαρμογών ΒΔ στο Διαδίκτυο



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να αναφέρει στοιχεία της αρχιτεκτονικής των εφαρμογών ΒΔ στο Διαδίκτυο

Οι αρχιτεκτονικές των υπολογιστικών συστημάτων διακρίνονται σε δύο βασικές κατηγορίες ως προς την διασύνδεση των χρηστών με το σύστημα:

Κεντρικά συστήματα (centralized systems), στα οποία οι χρήστες συνδέονται στον εξυπηρετητή μέσω τερματικών που δεν έχουν επεξεργαστική ισχύ αλλά χρησιμεύουν μόνο για προβολή. Σε αυτά τα συστήματα η επικοινωνία γίνεται μέσω δικτύων τα οποία μεταφέρουν τα μηνύματα των χρηστών προς το σύστημα και τα αποτελέσματα του συστήματος προς τους χρήστες.

Συστήματα βασικής αρχιτεκτονικής πελάτη/εξυπηρετητή (client/server architecture) όπου το λογισμικό κατανέμεται ανάμεσα σε πελάτες και εξυπηρετητές. Οι χρήστες (πελάτες) τοπικά διαθέτουν επεξεργαστική ισχύ και συνδέονται με τον εξυπηρετητή ή τους εξυπηρετητές για την εκτέλεση διάφορων λειτουργιών. Μερικές από αυτές τις λειτουργίες μπορεί να εκτελούνται και σε τοπικό επίπεδο, δηλαδή στον πελάτη, και κατά κανόνα οι πελάτες είναι διαφορετικού τύπου υπολογιστές ή συσκευές.

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων ακολούθησαν εξελικτικά μια μετάβαση από κεντρικά συστήματα σε συστήματα πελάτη/εξυπηρετητή. Τα σύγχρονα ΣΔΒΔ πελάτη/εξυπηρετητή με τη σειρά τους μπορεί επιπλέον να στηρίζονται σε **δύο ή τρία επίπεδα λογισμικού (two-tier ή three-tier architecture)**.

Τα δύο επίπεδα (two-tier) διαχωρίζουν τη διεπαφή του χρήστη (user interface) και τα προγράμματα εφαρμογών (application programs), από τον εξυπηρετητή των ερωτήσεων και συνδιαλλαγών (query & transaction server). Οι εφαρμογές του πελάτη, όταν είναι απαραίτητο, συνδέονται με τον εξυπηρετητή του ΣΔΒΔ μέσω συγκεκριμένων πρωτοκόλλων και προγραμματιστικών διεπαφών (Application Programming Interfaces – API).

Στην αρχιτεκτονική τριών επιπέδων υπάρχει και ένα ενδιάμεσο επίπεδο ανάμεσα στον πελάτη και τον εξυπηρετητή του ΣΔΒΔ. Το επίπεδο αυτό συνήθως είναι εξυπηρετητής εφαρμογών (application server) ή εξυπηρετητής Παγκόσμιου Ιστού (web server). Ο εξυπηρετητής αυτός εκτελεί εφαρμογές, κατευθύνει τα αιτήματα των πελατών για πρόσβαση στα δεδομένα (εξυπηρετητή ερωτήσεων) και εφαρμόζει κανόνες που αφορούν στη «επιχειρηματική λογική» (business logic) του συστήματος.

Οι κανόνες αυτοί για παράδειγμα μπορεί να αφορούν στην ασφάλεια των αιτημάτων των πελατών.

Στην περίπτωση που το ενδιάμεσο επίπεδο είναι εξυπηρετητής του Παγκόσμιου Ιστού, ο εξυπηρετητής αυτός έχει το ρόλο της δυναμικής δημιουργίας των ιστοσελίδων που παρουσιάζουν στους χρήστες τα δεδομένα που είναι αποθηκευμένα στη βάση.



Ερωτήσεις / Δραστηριότητες

1. Περιγράψτε την αρχιτεκτονική τριών επιπέδων στην οποία βασίζονται τα σύγχρονα ΣΔΒΔ.
2. Αναφέρετε ένα παράδειγμα εφαρμογής στο οποίο θα ήταν βολική η επιλογή βάσης δεδομένων γράφων.

κεφάλαιο

7

Εφαρμογές Βάσεων Δεδομένων

7.1 Εισαγωγή

Καθημερινά χρησιμοποιούμε διάφορες εφαρμογές του Διαδικτύου που χρησιμοποιούν Βάσεις Δεδομένων. Το κεφάλαιο αυτό αξιοποιεί τρία σύγχρονα παραδείγματα ΒΔ, οργανωμένα ως δραστηριότητες, ώστε μέσα από τη χρήση τους να εξερευνηθούν θέματα σχεδίασης και υλοποίησης Συστημάτων Βάσεων Δεδομένων και ανάπτυξης Διαδικτυακών εφαρμογών.



Στόχοι / Προσδοκώμενα Αποτελέσματα

Ο μαθητής/τρια πρέπει να είναι ικανός/ή:

- να αναφέρει και να περιγράφει εφαρμογές του Διαδικτύου που χρησιμοποιούν Βάσεις Δεδομένων
- να αναγνωρίζει την αναγκαιότητα χρήσης ΒΔ σε συγκεκριμένες εφαρμογές και εφαρμογές του Διαδικτύου
- να εντοπίζει θέματα ασφάλειας σε εφαρμογές ΒΔ στο Διαδίκτυο
- να αναφέρει ζητήματα/προβλήματα που προκύπτουν από την διασύνδεση ΒΔ με εφαρμογές του Διαδικτύου
- να διακρίνει και να επιλέγει καλές πρακτικές σε θέματα που αφορούν τη διαχείριση εφαρμογών ΒΔ

7.2 Διαδικτυακή Βάση Δεδομένων Διαθέσιμων Ραντεβού Ιατρών ΕΟΠΥΥ

Η Διαδικτυακή εφαρμογή αναζήτησης συμβεβλημένων παρόχων υγείας από την επίσημη ιστοσελίδα του Εθνικού Οργανισμού Παροχής Υπηρεσιών Υγείας (ΕΟΠΥΥ), βοηθά στον εντοπισμό Ιατρών σε όλες τις γεωγραφικές περιοχές της Ελλάδας, με συγκεκριμένα κριτήρια επιλογής.

Χρησιμοποιείται η διεύθυνση: <https://apps.ika.gr/eFindDoctor/> ή επιλογές στην επίσημη ιστοσελίδα του ΕΟΠΥΥ.

Γιατρός	Διεύθυνση	Τηλέφωνο	Διοθεσιμότητα/Πλαφόν	Μεταβολές/Ακυρώσεις	Χάρτης
ΑΒΡΑΜΙΔΗΣ ΒΑΣΙΛΕΙΟΣ	ΝΕΟΦΥΤΟΥ ΒΑΜΒΑ 4, ΑΘΗΝΑ, 10674	6945437676	190/200	0/0	Χάρτης
ΑΓΓΕΛΑΤΟΣ ΓΕΡΑΣΙΜΟΣ	ΠΑΤΗΣΙΩΝ 126 & ΙΘΑΚΗΣ, ΑΘΗΝΑ, 11257	2108824151	101/200	0/0	Χάρτης
ΑΓΚΟΣΤΟΝ ΚΑΤΑΛΙΝ	ΚΥΠΡΙΩΝ ΗΡΩΩΝ 70, ΗΛΙΟΥΠΟΛΗ, 16341	2109910551	167/200	0/0	Χάρτης
ΑΛΑΤΣΙΔΟΥ ΖΗΝΟΒΙΑ	ΣΟΛΩΝΟΣ 34, ΚΟΛΩΝΑΚΙ ΑΘΗΝΑ, 10673	2109244040 6944759665	167/200	0/0	Χάρτης
ΑΝΔΡΙΑΝΟΠΟΥΛΟΣ ΕΥΣΤΑΘΙΟΣ - ΔΗΜΟΣΘΕΝ	ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ 65, ΧΑΛΑΝΔΡΙ, 15231	2114063227 6944265526	176/200	0/0	Χάρτης
ΑΡΤΟΠΟΥΛΟΣ ΜΗΝΑΣ	ΚΡΗΤΗΣ 72, ΝΕΑ ΙΩΝΙΑ, 14231	2102717066	193/200	0/0	Χάρτης

Δραστηριότητες

1. Εντοπίστε τα στοιχεία που θεωρείτε ότι προσδιορίζουν την παραπάνω εφαρμογή σε Βάση Δεδομένων
2. Αναπτύξτε το σχεσιακό μοντέλο της Βάσης Δεδομένων. Σχηματίστε τους πιθανούς πίνακες, προσδιορίστε τα πεδία και τους τύπους δεδομένων, χαρακτηρίστε τα κλειδιά κλπ.
3. Τι πληροφορία χαρακτηρίζει και πώς αποθηκεύεται στη Βάση Δεδομένων η εμφάνιση του χάρτη
4. Πώς πραγματοποιείται η επικοινωνία του χρήστη με την εφαρμογή; Ποια τα υποχρεωτικά πεδία;
5. Ποιοι είναι οι πιθανοί χρήστες της εφαρμογής;
6. Ποια η χρησιμότητα του "Κωδικού Οπτικής Επιβεβαίωσης";
7. Ποια τα ερωτήματα που προσδιορίζονται με την "Αναζήτηση" στη χρήση της εφαρμογής;
8. Διατυπώστε τα αντίστοιχα SQL Ερωτήματα

7.3 Μελέτη Κοινωνικών Δικτύων ως προς τις Χρησιμοποιούμενες Τεχνολογίες σε ΒΔ

Καθημερινά εκατομμύρια χρήστες συνδέονται στα κοινωνικά δίκτυα για να επικοινωνήσουν με φίλους και συνεργάτες. Τα συστήματα κοινωνικών δικτύων διαχειρίζονται ένα μεγάλο όγκο δεδομένων και χρησιμοποιούν βάσεις δεδομένων για να αποθηκεύουν και να διαχειρίζονται τα δεδομένα.

Έχουμε αναρωτηθεί άραγε ποιες είναι οι χρησιμοποιούμενες τεχνολογίες των κοινωνικών δικτύων στον τομέα των Βάσεων Δεδομένων καθώς και τα χαρακτηριστικά αυτών; Ποια είναι τα πλεονεκτήματα των τεχνολογιών αυτών και ποια θέματα ασφάλειας εγείρονται; Θα μπορούσαμε να δημιουργήσουμε ή να σχεδιάσουμε ένα υποτυπώδες κοινωνικό δίκτυο έστω και στο χαρτί;

Επιλέξτε ένα σύστημα κοινωνικής δικτύωσης και προσπαθήστε να απαντήσετε στις παρακάτω ερωτήσεις. Μπορείτε να αναζητήσετε πρόσθετη βοήθεια στο διαδίκτυο. Στο τέλος παρουσιάστε τις απόψεις σας και τις προτάσεις σας τεκμηριωμένα δημιουργώντας μία παρουσίαση ή ένα βίντεο στην ολομέλεια της τάξης σας.

1. Τι υπηρεσίες παρέχει το κοινωνικό δίκτυο το οποίο επιλέξατε;
2. Ποιος είναι ο όγκος των δεδομένων που διακινούνται καθημερινώς; Ποια είναι η χωρητικότητα των αποθηκευμένων δεδομένων; Πόσοι χρήστες διατηρούν λογαριασμό και συνδέονται καθημερινά;
3. Ποιες είναι οι προκλήσεις που αντιμετωπίζει το κοινωνικό δίκτυο το οποίο επιλέξατε από τεχνολογικής πλευράς;
4. Τα δεδομένα βρίσκονται κάπου όλα μαζί συγκεντρωμένα ή είναι κατανεμημένα σε διάφορα σημεία; Θα μπορούσαν να είναι στο Υπολογιστικό Σύννεφο (cloud) και τι σημαίνει αυτό; Ποια είναι η αρχιτεκτονική που ακολουθείται στην αποθήκευση των δεδομένων και ποιες τεχνολογίες Βάσεων Δεδομένων χρησιμοποιούνται;
5. Θεωρήστε το εννοιολογικό μοντέλο μιας Βάσης Δεδομένων κοινωνική δικτύωσης.
 - Περιγράψτε τουλάχιστον πέντε οντότητες που θα μπορούσαν να αποτελούν μέρος του εννοιολογικού αυτού μοντέλου. Βρείτε τα χαρακτηριστικά τους και τις συσχετίσεις μεταξύ τους.
 - Σχεδιάστε το αντίστοιχο ΟΣ (ER) διάγραμμα και συμπεριλάβετε σε αυτό όλα όσα γνωρίζετε. Επιλύστε τυχόν θέματα συσχετίσεων που παρουσιάζονται.
 - Μετατρέψτε το ΟΣ (ER) διάγραμμα στο αντίστοιχο σχεσιακό μοντέλο.
 - Σε περίπτωση, που δημιουργούσατε ένα κοινωνικό δίκτυο για το σχολείο σας σε τι θα διέφερε και για ποιους λόγους από αυτό του κοινωνικού δικτύου που επιλέξατε; Σχεδιάστε το αντίστοιχο ΟΣ διάγραμμα.
6. Θεωρήστε την υλοποίηση του σχεσιακού μοντέλου που δημιουργήσατε στο πέμπτο ερώτημα. Γράψτε τουλάχιστον πέντε ερωτήσεις SQL που σχετίζονται με την δημιουργία και διαχείριση πινάκων.
7. Μερικοί άνθρωποι, θεωρούν ότι η αυξημένη χρήση των κοινωνικών δικτύων μπορεί να οδηγήσει σε προβλήματα, όπως είναι η πρόσβαση σε ευαίσθητα προσωπικά δεδομένα καθώς και σε νέες μορφές εγκληματικότητας.
 - Έχουν παρατηρηθεί για το κοινωνικό δίκτυο το οποίο μελετάτε παρόμοια φαινόμενα;
 - Ποια ήταν η αντίδραση της εταιρείας στα αντίστοιχα κρούσματα;
 - Με ποιόν τρόπο συνδέονται οι χρήστες στο κοινωνικό δίκτυο; Ποιο είναι το σύστημα αυθεντικοποίησης (authentication) που χρησιμοποιείται;
 - Έχετε να επισημάνετε τυχόν κενά στην ασφάλεια καθώς και στην προστασία των προσωπικών δεδομένων;
 - Ποιες είναι οι προτάσεις σας για την διασφάλιση των προσωπικών δεδομένων και την αποφυγή μορφών εγκληματικότητας;

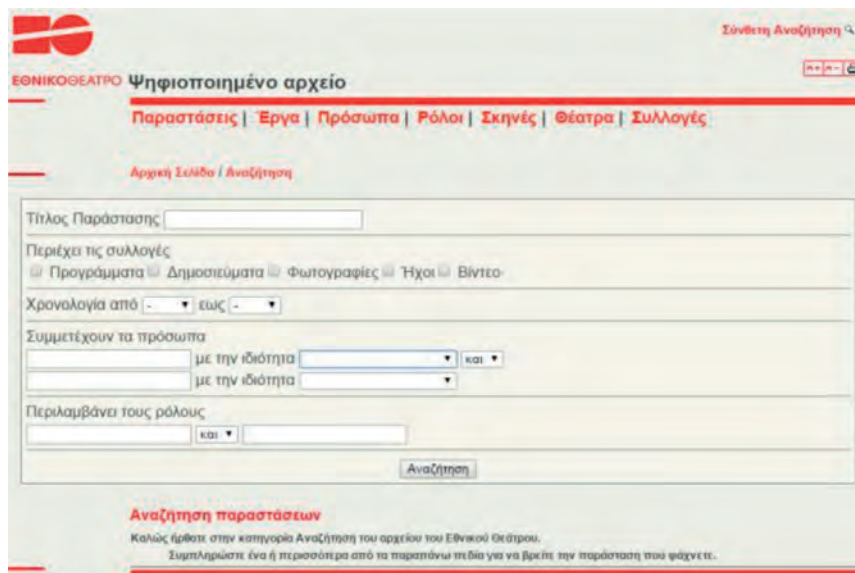
8. Τα τελευταία χρόνια παρατηρείται αυξανόμενη τάση αναζήτησης προσωπικού μέσω ιστοσελίδων κοινωνικής δικτύωσης. Η δημιουργία εταιρικών ιστοσελίδων σε κοινωνικά δίκτυα, η καταχώρηση αγγελιών νέων θέσεων εργασίας σε συγκεκριμένες ιστοσελίδες αποτελούν συνηθισμένη πρακτική. Ως εκ τούτου, αναδύονται ηθικά ερωτήματα. Νομίζετε ότι είναι ηθικό τα κοινωνικά δίκτυα να πουλάνε προσωπικές πληροφορίες χρηστών σε επιχειρήσεις έρευνας αγοράς και όχι μόνο;
9. Πολλοί γονείς φοβούνται ότι ενώ τα παιδιά τους παίζουν παιχνίδια στο Διαδίκτυο μέσω των μέσων κοινωνικής δικτύωσης, διάφοροι μπορεί να τα πλησιάσουν με κακόβουλες διαθέσεις. Ποια είναι τα μέτρα που θα λαμβάνετε ως σχεδιαστές της Βάσης Δεδομένων ώστε να αποφεύγονται τέτοια κρούσματα.

7.4 Ψηφιοποιημένο Αρχείο Εθνικού Θεάτρου

Το Εθνικό Θέατρο έχει δημιουργήσει ψηφιοποιημένο αρχείο για τις παραστάσεις από το 1932 μέχρι το 2006. Το αρχείο είναι διαθέσιμο στο ευρύ κοινό μέσω του Διαδικτύου. Για κάθε παράσταση υπάρχουν πληροφορίες για το έργο και τους συντελεστές της παράστασης, φωτογραφίες, προγράμματα παραστάσεων και αποκόμματα εφημερίδων. Για παραστάσεις μετά το 1955 υπάρχει ηχητικό υλικό ενώ για παραστάσεις μετά το 1994 υπάρχουν βιντεοσκοπήσεις.

Το ψηφιοποιημένο αρχείο του Εθνικού Θεάτρου αποτελεί χαρακτηριστικό παράδειγμα εφαρμογής ΒΔ στο Διαδίκτυο. Όλες οι πληροφορίες σε μορφή κειμένου και το πολυμεσικό υλικό (εικόνες, ήχοι, βίντεο) είναι αποθηκευμένα σε ΒΔ που φιλοξενείται σε κάποιο ΣΔΒΔ. Η διάθεση των περιεχομένων της ΒΔ γίνεται μέσω ενός εξυπηρετητή Διαδικτύου (web server) που επικοινωνεί με το ΣΔΒΔ. Ο εξυπηρετητής Διαδικτύου φιλοξενεί τη διαδικτυακή εφαρμογή (web application), η οποία έχει τη μορφή μιας φόρμας και επιτρέπει στους χρήστες να πραγματοποιούν αναζητήσεις στα περιεχόμενα του ψηφιοποιημένου αρχείου. Η πρόσβαση στη φόρμα αναζήτησης γίνεται μέσω ενός φυλλομετρητή Διαδικτύου (web browser).

1. Χρησιμοποιήστε ένα φυλλομετρητή Διαδικτύου για να επισκεφθείτε το ψηφιοποιημένο αρχείο του Εθνικού Θεάτρου στη διεύθυνση <http://www.nt-archive.gr/>.
2. Στο επάνω μέρος της σελίδας υπάρχουν σύνδεσμοι για τις φόρμες απλής αναζήτησης. Χρησιμοποιήστε αυτές τις φόρμες για να πραγματοποιήσετε αναζητήσεις για έργα και ρόλους, ώστε να αποκτήσετε μια πρώτη εικόνα για το είδος και το πλήθος των πληροφοριών που είναι διαθέσιμες. Μπορείτε να αναζητήσετε πληροφορίες για κάποιο κλασικό θεατρικό έργο που παρακολουθήσατε πρόσφατα ή για τα έργα που διδαχθήκατε στο Γυμνάσιο στο πλαίσιο του μαθήματος «**Δραματική Ποίηση**» («**Ελένη**» του **Ευριπίδη** και «**Όρνιθες**» του **Αριστοφάνη**).



Εικόνα 7.1 Ψηφιοποιημένο αρχείο Εθνικού Θεάτρου: Φόρμα σύνθετης αναζήτησης (Πηγή: <http://www.nt-archive.gr/search.aspx>)

3. Ακολουθήστε το σύνδεσμο στο άνω δεξί τμήμα της οθόνης για να επισκεφθείτε τη **φόρμα σύνθετης αναζήτησης** (βλ. εικόνα 7.1). Κάθε πεδίο της φόρμας πρέπει να αντιστοιχεί σε στήλη ενός πίνακα της ΒΔ.
 - Ποιος πρέπει να είναι ο τύπος δεδομένων της στήλης που αντιστοιχεί σε κάθε πεδίο της φόρμας;

- Εάν υπάρχουν περισσότερες από μία εναλλακτικές, καταγράψτε τους περιορισμούς που θέτει ο αντίστοιχος τύπος δεδομένων.



Εικόνα 7.2 Ψηφιοποιημένο αρχείο Εθνικού Θεάτρου: Απλοποιημένο σχήμα ΒΔ

4. Θεωρήστε ότι το σχήμα της ΒΔ που υλοποιεί το ψηφιοποιημένο αρχείο του Εθνικού Θεάτρου είναι αυτό της εικόνας 7.2 [πρόκειται για απλοποιημένο σχήμα της ΒΔ που δημιουργήθηκε για εκπαιδευτικούς σκοπούς – η πραγματική ΒΔ σίγουρα περιέχει περισσότερους πίνακες με επιπλέον στήλες].
 - Τι τύπου πρέπει να είναι οι συσχετίσεις μεταξύ των πινάκων (1:1, 1:N, N:M);
 - Τι περιορισμούς θέτει ο αντίστοιχος τύπος συσχέτισης για τα περιεχόμενα της ΒΔ;
5. Πραγματοποιήστε τις ακόλουθες αναζητήσεις μέσω της φόρμας σύνθετης αναζήτησης:
 - Τίτλος παράστασης «Ελένη».
 - Συμμετέχουν τα πρόσωπα «Αριστοφάνης» με την ιδιότητα «Θεατρικός Συγγραφέας». Χρονολογία από «1932» έως «1950» και περιλαμβάνει τους ρόλους «Ιφιγένεια» ή «Ορέστης».
 - Έχοντας υπόψη το σχήμα ΒΔ της εικόνας 7.2, ποια πρέπει να είναι η συνθήκη (WHERE) των εντολών SELECT που υλοποιούν τις παραπάνω αναζητήσεις;



ΧΡΗΣΙΜΕΣ ΔΙΕΥΘΥΝΣΕΙΣ ΔΙΑΔΙΚΤΥΟΥ

Ιστορία Εθνικού Θεάτρου: <https://www.n-t.gr/el/knowus/history/>

Πληροφορίες για το ψηφιοποιημένο αρχείο του Εθνικού Θεάτρου: <http://www.n-t.gr/el/archive/>

Ψηφιοποιημένο αρχείο Εθνικού Θεάτρου: <http://www.nt-archive.gr>

Βιβλιογραφία, Αναφορές

Ζαχαρής, Κ., Κασσιμάτης, Ν., Κουνιάκης, Χ., Μανωλόπουλος, Ι., & Οικονόμου, Θ. (2002). Βάσεις Δεδομένων, Τεχνικά Επαγγελματικά Εκπαιδευτήρια, Β' Τάξη 1ου Κύκλου, Κατεύθυνση Υποστήριξη Συστημάτων Υπολογιστών. Υπουργείο Παιδείας και Θρησκευμάτων, Παιδαγωγικό Ινστιτούτο. Αθήνα: ΟΕΔΒ.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α.Ν. (2006). Βάσεις Δεδομένων: Θεωρία και Πρακτική Εφαρμογή. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Connolly, T., Begg, C., & Strachan, A. (2006). Βάσεις Δεδομένων: Μια Πρακτική Προσέγγιση στο Σχεδιασμό την Υλοποίηση και τη Διαχείριση Συστημάτων Βάσεων Δεδομένων (4η έκδ., Τόμ. 12) (Κ. Ευμορφόπουλος, Δ. Γκαρμπολάς, & Π. Υψηλάντης, Μετάφρ.). Αθήνα: Εκδόσεις Ίων.

Delisle, M. (2012). Mastering phpMyAdmin 3.4 for Effective MySQL Management. Birmingham, UK: Packt Publishing.

Elmasri, R., & Navathe, S.B. (2011). Fundamentals of Database Systems (6th ed.). Boston, MA: AddisonWesley.

GarciaMolina, H., Ullman, J.D., & Widom, J. (2009). Database Systems: The Complete Book (2nd ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Kofler, M. (2005). The Definitive Guide to MySQL5 (3rd ed.) (D. Kramer, Trans.). New York, NY: Apress.

Kruckenberg, M., & Pipes, J. (2005). Pro MySQL. New York, NY: Apress.

Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems (3rd ed.). New York, NY: McGrawHill.

Silberschatz, A., Korth, H.F., & Sudarshan, S. (2011). Database System Concepts (6th ed.). New York, NY: McGrawHill.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Συστήματα Βάσεων Δεδομένων: Η Πλήρης Θεωρία των Βάσεων Δεδομένων (6η έκδ.) (Μ. Γκλαβά, Μετάφρ.). Αθήνα: Εκδόσεις Μ.Γκιούρδας.

Speelpenning, J., Daux, P., & Gallus, J. (2008). Data Modeling and Relational Database Design (Oracle Course). Oracle Corporation.

Gaurav Vaish, 2013. Getting Started with NoSQL. Packt Publishing

Kristina Chodorow. MongoDB: The Definitive Guide, Second Edition. O'Reilly Media, Inc.

Χαρακτηριστικές NoSQL Βάσεις Δεδομένων:

Amazon SimpleDB: <http://aws.amazon.com/simpliedb> BaseX: <http://www.basex.org>

Cassandra: <http://cassandra.apache.org>

CouchDB: <http://couchdb.apache.org>

Google Datastore: <http://developers.google.com/appengine>

HBase: <http://hbase.apache.org>

MemcacheDB: <http://memcachedb.org>

MongoDB: <http://www.mongodb.com>

Neo4j: <http://www.neo4j.org>

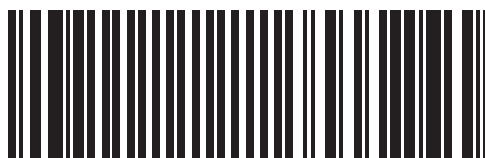
Redis: <http://redis.io>

ACID <http://el.wikipedia.org/wiki/ACID>

Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔΙΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946,108, Α').

Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήματος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας, Θρησκευμάτων και Αθλητισμού / ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ.

Κωδικός βιβλίου: 0-24-0554
ISBN 978-960-06-5155-3



(01) 000000 0 24 0554 4