



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Εισαγωγή στις Βάσεις Δεδομένων II

Ενότητα: Εισαγωγή II - Σχεδίαση και Αρχιτεκτονική

Κωστής Πηγουνάκης

Τμήμα Οικονομικών Επιστημών

2. Αρχιτεκτονική και Σχεδίαση

2.1. Αρχές Σωστής Σχεδίασης

Για να λειτουργεί μια βάση δεδομένων σωστά και αποδοτικά πρέπει, τόσο η ίδια, όσο και το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) που τη φιλοξενεί να είναι σωστά σχεδιασμένα. Αυτό ανάγεται σε μια σειρά από απαιτήσεις :

- Η βάση δεδομένων να μην περιλαμβάνει **περιττά δεδομένα (redundant data)**. Τα ίδια δεδομένα δεν θα πρέπει να καταχωρούνται στη βάση δύο φορές. Εάν συμβαίνει κάτι τέτοιο, τότε ανακύπτουν δυο σοβαρά προβλήματα. Το πρώτο και προφανές πρόβλημα, είναι ότι σπαταλούμε άσκοπα αποθηκευτικό χώρο. Το δεύτερο και σοβαρότερο πρόβλημα, είναι ότι υπάρχει ο κίνδυνος δημιουργίας **ασυνεπών δεδομένων (inconsistent data)**. Πράγματι, εάν κρατάμε δύο φορές την ίδια πληροφορία και η πληροφορία αυτή σε κάποια χρονική στιγμή υποστεί κάποιο είδος επεξεργασίας – για παράδειγμα την τροποποιήσουμε ή τη διαγράψουμε – τότε η επεξεργασία αυτή θα πρέπει να εφαρμοσθεί και στις δύο καταχωρήσεις που αφορούν το ίδιο δεδομένο, διότι διαφορετικά, η βάση θα περιέχει δεδομένα που δεν είναι συνεπή. Για το λόγο αυτό, ένας από τους πρώτους ελέγχους που πραγματοποιούμε στη βάση αμέσως μετά το σχεδιασμό της, είναι ο έλεγχος παρουσίας επαναλαμβανόμενων πεδίων, και η απομάκρυνσή τους, εφόσον υπάρχουν.
- Η βάση δεδομένων πρέπει να είναι σχεδιασμένη έτσι ώστε τα δεδομένα που περιλαμβάνει να ανακτώνται εύκολα και γρήγορα. Μια κακή σχεδίαση της δομής της βάσης, οδηγεί στη δημιουργία μιας βάσης η οποία είναι εξαιρετικά δυσκίνητη και αναποτελεσματική. Η σωστή σχεδίαση μιας βάσης δεδομένων, επιτυγχάνεται με εφαρμογή επί της δομής της βάσης, μιας τεχνικής, η οποία ονομάζεται **κανονικοποίηση (normalization)**. Η περιγραφή αυτής της τεχνικής παρουσιάζεται διεξοδικά σε παρακάτω.
- Το ΣΔΒΔ να χαρακτηρίζεται από **σύστημα ασφάλειας (security system)** που να απαγορεύει την πρόσβαση στα δεδομένα μη εξουσιοδοτημένων ατόμων. Αυτό ισχύει κυρίως όπως θα δούμε, σε μεγάλες βάσεις δεδομένων με πολλούς χρήστες, και η τεχνική που συνήθως εφαρμόζεται, συνίσταται στον καθορισμό **ομάδων χρηστών (user groups)** με διαφορετικά δικαιώματα πρόσβασης στον καθένα από αυτούς. Ο κάθε χρήστης λαμβάνει ένα κωδικό πρόσβασης (password) και τα καθήκοντα που μπορεί να επιτελέσει είναι εντελώς συγκεκριμένα και καθορισμένα εκ των προτέρων.

- Το ΣΔΒΔ να παρέχει **έλεγχο σε ταυτόχρονες προσπελάσεις (concurrency control)** πάνω στα ίδια δεδομένα.
- Το ΣΔΒΔ να διαθέτει σύστημα δημιουργίας **αντιγράφων ασφαλείας (backups)** των δεδομένων που είναι καταχωρημένα σε αυτή. Η ταυτόχρονη αποθήκευση των δεδομένων σε περισσότερους από ένα δίσκους, είναι μια εργασία επιβεβλημένη, προκειμένου να είναι δυνατή η ανάκτησή τους σε περιπτώσεις κατάρρευσης της βάσης για οποιοδήποτε λόγο.

Ειδικότερα, μια καλά σχεδιασμένη βάση δεδομένων σε ένα λειτουργικό και αξιόπιστο ΣΔΒΔ διακρίνεται από τα εξής χαρακτηριστικά:

- Είναι *απλή και κατανοητή* ως προς τη *δομή* της
- Διασφαλίζεται η *ακεραιότητα δεδομένων (data integrity)*
- Αντεπεξέρχεται σε *αναμενόμενα (planned)* αλλά και σε *μη αναμενόμενα (ad hoc) ερωτήματα (queries)*
- Ανταποκρίνεται στο σκοπό για τον οποίο δημιουργείται και να έχει *βιωσιμότητα*
- *Ελαχιστοποιείται η εξάρτησή* της από τις εφαρμογές που θα την προσπελούν

2.2. Αρχιτεκτονικές Συστημάτων Διαχείρισης Βάσεων Δεδομένων

2.2.1. Βαθμίδες και Αρχιτεκτονικές

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων είναι εφαρμογές λογισμικού που λειτουργούν συνήθως μέσω ενός δικτύου ή του διαδικτύου. Οι κοινοί τρόποι σχεδιασμού του λογισμικού σε ό,τι αφορά τις επιμέρους *μονάδες (modules)* και τη μεταξύ τους επικοινωνία χαρακτηρίζονται ως *αρχιτεκτονικά μοντέλα ή αρχιτεκτονικές*.

Στη συνέχεια θα εξετάσουμε τις βασικές αρχιτεκτονικές των σύγχρονων ΣΔΒΔ που κατηγοριοποιούνται ανάλογα με τις **βαθμίδες (tiers)** που εμπλέκονται. Ο όρος των βαθμίδων ανάγεται στη λογική **αρχιτεκτονική P-A-D** από τα αρχικά των λέξεων *Παρουσίαση-Εφαρμογή-Δεδομένα (Presentation-Application-Data)*, και βασίζεται στην αλληλεπίδραση των τριών αυτών λογικών διαδικασιών:

- **Παρουσίαση –Presentation** : γνωστή και ως διεπαφή χρήστη (user interface)
- **Εφαρμογή – Application** : γνωστή και ως επεξεργασία ή λογική εφαρμογής
- **Δεδομένα – Data** : γνωστή και ως διαχείριση

Η ανάπτυξη μεγάλων εφαρμογών επικεντρωμένων στην ανταλλαγή δεδομένων μέσω δικτύου οδήγησε στο **περιβάλλον Πελάτη/Εξυπηρετητή (Client/Server)** στη δεκαετία του 1980. Σε αυτού του περιβάλλοντος τις αρχιτεκτονικές υπάρχει μια κεντρική εφαρμογή που λέγεται **Εξυπηρετητής (Server)** και με την οποία συνδέονται και ανταλλάσσουν πληροφορίες άλλες εφαρμογές που βρίσκονται στους υπολογιστές των χρηστών και χαρακτηρίζονται ως **Πελάτες (Clients)**. Η ανταλλαγή της πληροφορίας γίνεται μέσω

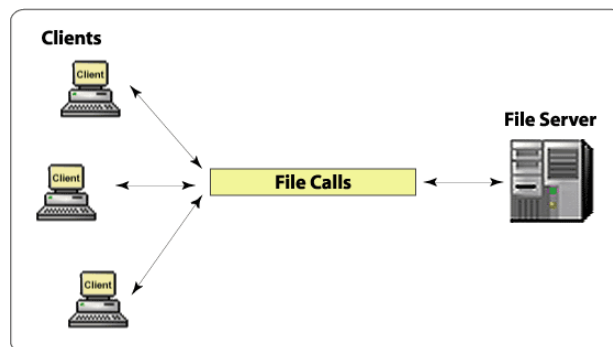
συναλλαγών (*transactions*), δηλαδή μέσω καλά ορισμένων αιτημάτων (*requests*) και αποκρίσεων (*responses*).

Το περιβάλλον Πελάτη/Εξυπηρετητή αφορά στη φυσική αρχιτεκτονική και εμφανίζει επιμέρους αρχιτεκτονικές. Όταν το μεγαλύτερο μέρος της επεξεργασίας (λογική της εφαρμογής) γίνεται στον Εξυπηρετητή και ο Πελάτης αναλαμβάνει κυρίως την παρουσίαση αποτελεσμάτων της διαδικασίας, τότε έχουμε την περίπτωση *Αδύνατου Πελάτη (Thin Client) – Παχύ Εξυπηρετητή (Fat Server)*, ενώ όταν ο Πελάτης αναλαμβάνει το μεγαλύτερο μέρος της επεξεργασίας μιλάμε για *Παχύ Πελάτη (Fat Client) – Αδύνατο Εξυπηρετητή (Thin Server)*. Στην ειδική περίπτωση του Διαδικτύου, όπου αποκλειστικά και μόνο το επίπεδο της παρουσίασης είναι στο φυλλομετρητή (*browser*), ενώ το σύνολο της επεξεργασίας είναι στον Εξυπηρετητή, μιλάμε για *Υπέρ-Παχύ Εξυπηρετητή (Super-Fat Server)*.

2.2.2. Αρχιτεκτονική μιας βαθμίδας (1-tier)

Η απλούστερη αρχιτεκτονική είναι η αρχιτεκτονική **μιας βαθμίδας (1-tier)** στην οποία ο υπολογιστής που φιλοξενεί την εφαρμογή είναι ο ίδιος με το οποίο ο χρήστης επικοινωνεί με την εφαρμογή. Τυπική περίπτωση είναι η ανάπτυξη μιας βάσης στο ΣΔΒΔ MS Access, όπου αποθηκεύει τα δεδομένα της στον τοπικό δίσκο ενός προσωπικού υπολογιστή και ταυτόχρονα παρέχει τη δυνατότητα προσπέλασης των δεδομένων μέσω διεπαφής που αποθηκεύεται στον ίδιο υπολογιστή.

Μια επέκταση αυτής της αρχιτεκτονικής είναι όταν ένας υπολογιστής – *εξυπηρετητής αρχείων (file server)* φιλοξενεί τα αρχεία αποθήκευσης του ΣΔΒΔ (αλλά όχι κατ' ανάγκη το ίδιο) και βρίσκεται σε τοπικό δίκτυο με άλλους υπολογιστές. Οι άλλοι υπολογιστές έχουν δυνατότητα *ΆΜΕΣΗΣ* προσπέλασης των αρχείων του με τη μορφή *διαμοιρασμού (file-sharing)* και *ΟΛΟΙ* έχουν εγκατεστημένη την εφαρμογή του ΣΔΒΔ. Με αυτόν τον τρόπο διαφορετικοί υπολογιστές έχουν πρόσβαση στην ίδια βάση δεδομένων, αλλά με αρχιτεκτονική ίδια με αυτήν του ενός υπολογιστή.



Εικόνα 2-1 : Μοντέλο Client/Server 1-tier

Πλεονεκτήματα:

- Διαμοιρασμός δεδομένων σε πολλαπλούς χρήστες γίνεται με απλό τρόπο αν και οι δυνατότητες είναι περιορισμένες.

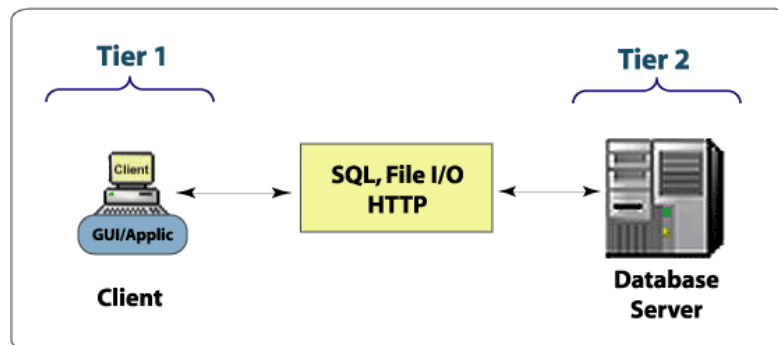
- Το κόστος αποθήκευσης είναι χαμηλό και μοιράζεται στους διασυνδεδεμένους χρήστες.
- Επειδή η ίδια εφαρμογή ΣΔΒΔ εγκαθίσταται πολλές φορές, ενδεχομένως επιτυγχάνεται χαμηλότερη τιμή αγοράς ανά εγκατάσταση (site license).

Μειονεκτήματα:

- Μικρή δυνατότητα κλιμάκωσης.
- Αδυναμία ταυτόχρονης χρήσης.
- Προβλήματα ασφάλειας

2.2.3. Αρχιτεκτονική δύο βαθμίδων (2-tier)

Πρόκειται για την περίπτωση που ένα κεντρικό σύστημα, πολλές φορές με αποκλειστική αρμοδιότητα (server), παρέχει τη λειτουργικότητα για πολλά συστήματα χρηστών (clients). Το ΣΔΒΔ είναι στον εξυπηρετητή, με τον οποίο επικοινωνούν οι εφαρμογές –πελάτες.



Εικόνα 2-2 : Μοντέλο Client/Server 2-tier

Πλεονεκτήματα

- Κλιμάκωση, οριζόντια (αύξηση πελατών) και κάθετη (αύξηση εξυπηρετητών).
- Αποτελεσματική χρήση των υπολογιστικών πόρων με καλό επίπεδο ασφάλειας.
- Εύκολη ενσωμάτωση νέας τεχνολογίας.
- Μαζί με τον client μπορεί να υπάρχουν και άλλες εφαρμογές.

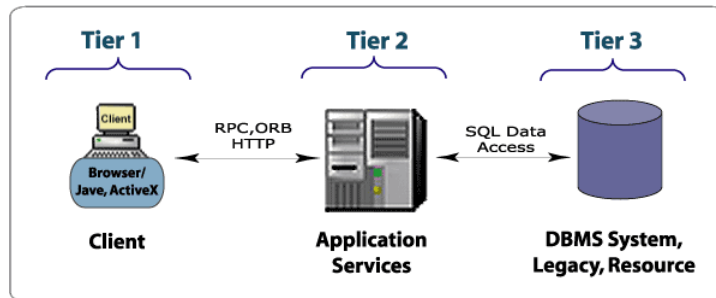
Μειονεκτήματα

- Η client εφαρμογή εγκαθίσταται σε πολλά συστήματα, πράγμα που προκαλεί δυσκολία αλλαγής της (π.χ. αναβάθμισης) και, για το λόγο αυτό, το κόστος συντήρησης αυξάνεται σημαντικά.

2.2.4. Αρχιτεκτονική πολλαπλών βαθμίδων (n-tier)

Η λειτουργικότητα της εφαρμογής του ΣΔΒΔ χωρίζεται σε διαφορετικά επίπεδα που διασυνδέονται λογικά και μοιράζονται πληροφορίες. Κάθε ενδιάμεσο επίπεδο αποτελεί ένα μοντέλο client/server. Η πιο συνηθής υλοποίηση είναι αυτή των τριών βαθμίδων (3-tier),

όπου υπάρχει σαφής φυσικός διαχωρισμός της επεξεργασίας από τη βάση δεδομένων και την παρουσίαση.



Εικόνα 2-3 : Μοντέλο Client/Server 3-tier

Πλεονεκτήματα

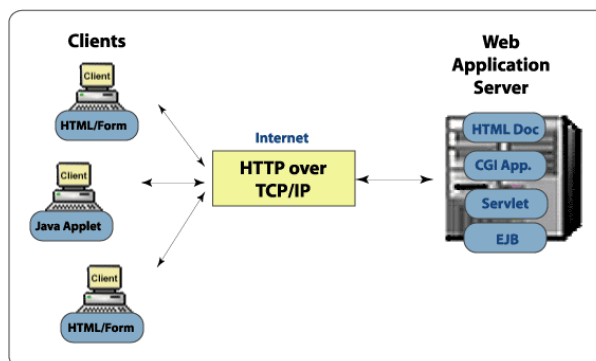
- Αυξημένη ασφάλεια, αφού ο χρήστης δεν προσπελάει άμεσα τη βάση δεδομένων
- Αυξημένη ευελιξία και έλεγχος λόγω των ενδιάμεσων επιπέδων και, επομένως, καλύτερη απόδοση
- Ελαχιστοποιημένη διαχείριση σε σχέση με το μοντέλο client/server

Μειονεκτήματα

- Πολυπλοκότητα της σχεδίασης
- Αυξημένο αρχικό κόστος υλοποίησης

Αρχιτεκτονική διαδικτύου

Στην αρχιτεκτονική πολλαπλών βαθμίδων εντάσσεται και η αρχιτεκτονική διαδικτύου. Σε αυτήν την περίπτωση το τελευταίο μέρος προς τον Πελάτη υλοποιείται μέσω του πρωτοκόλλου http και η μόνη απαίτηση από αυτόν είναι η εφαρμογή φυλλομετρητή (περίπτωση Υπερ-Παχύ Εξυπηρετητή).



Εικόνα 2-4 : Μοντέλο Client/Server Διαδικτύου

Πλεονεκτήματα

- Αυξημένη ασφάλεια, αφού ο χρήστης δεν προσπελάει άμεσα τη βάση δεδομένων
- Αυξημένη ευελιξία και έλεγχος και, επομένως, καλύτερη απόδοση

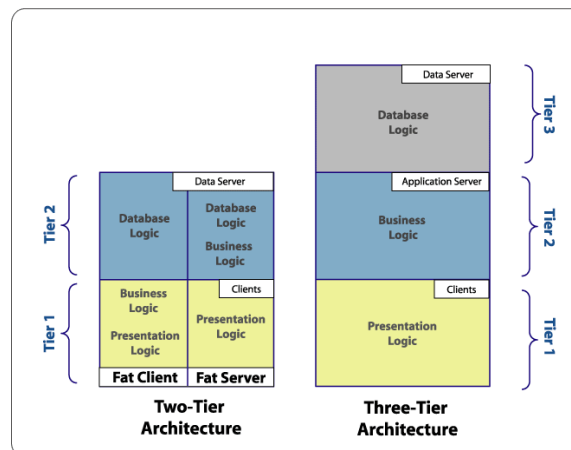
- Ελαχιστοποιημένη διαχείριση σε σχέση με το μοντέλο client/server
- Χαμηλές απαιτήσεις για την πλευρά του χρήστη

Μειονεκτήματα

- Αυξημένοι κίνδυνοι που αντιμετωπίζονται μέσω ισχυρών πολιτικών ασφάλειας (firewalls, λογισμικό ανίχνευσης εισβολών, κ.ά.)

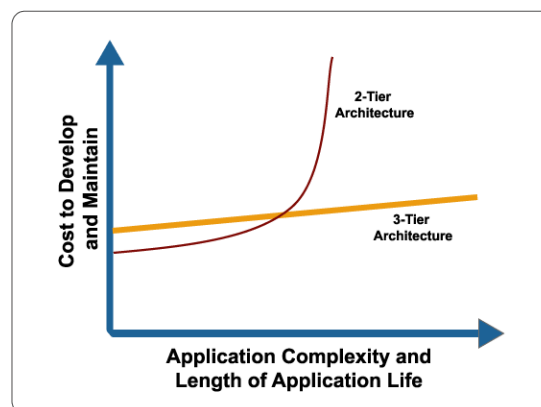
2.2.5. Σύγκριση αρχιτεκτονικών 2- και 3-βαθμίδων

Η βασική διαφορά μεταξύ των 2 και των 3 βαθμίδων είναι ο τρόπος που υλοποιείται το μέρος της *Εταιρικής Λογικής (Business Logic)*, δηλαδή η λογική της εφαρμογής κατά P-A-D. Έτσι, στην αρχιτεκτονική 2 βαθμίδων η Εταιρική Λογική αναπτύσσεται και στις δύο βαθμίδες, και στις δύο πλευρές του μοντέλου Πελάτη-Εξυπηρετητή. Στο παρακάτω σχήμα δίδεται παραστατικά η διαφορά μεταξύ των αρχιτεκτονικών 2-tier και 3-tier:



Εικόνα 2-5 : Σύγκριση αρχιτεκτονικών 2-tier και 3-tier

Συγκρίνοντας τις δύο αρχιτεκτονικές ως προς το κόστος ανάπτυξης και συντήρησης, τα συστήματα 2 βαθμίδων μπορεί να κλιμακώσουν υπερβολικά το κόστος καθώς αυξάνει η πολυπλοκότητα των εφαρμογών δηλ. η επεξεργασία και ο χρόνος ζωής της εφαρμογής. (πηγή : <http://wiki.eeng.dcu.ie/ee557/g3/296-EE.html>)



Εικόνα 2-6 : Κλιμάκωση κόστους αρχιτεκτονικών 2-tier και 3-tier

2.3. Σχεδίαση Βάσεων Δεδομένων

2.3.1. Στάδια Ζωής Βάσεων Δεδομένων

Στον κύκλο ζωής μιας βάσης δεδομένων μπορούμε να διακρίνουμε τρία στάδια. Το πρώτο Στάδιο προηγείται χρονικά και πρέπει να ολοκληρωθεί για να προχωρήσουμε στα επόμενα, ενώ για τα Στάδια 2 και 3 υπάρχει χρονική αλληλοεπικάλυψη. Παρακάτω αναλύονται λεπτομερώς τα τρία στάδια

- **Στάδιο 1: Σχεδίαση και υλοποίηση της βάσης:** Στο στάδιο αυτό αναλύονται οι απαιτήσεις που έχουμε για τη βάση δεδομένων, πραγματοποιείται ο σχεδιασμός και υλοποιείται η βάση δεδομένων χρησιμοποιώντας τα κατάλληλο λογισμικό και εξειδικευμένο προσωπικό. Το στάδιο αυτό είναι καίριο γιατί δεσμεύει την εξέλιξη της βάσης και πραγματοποιείται άπαξ. Λάθη κατά τη σχεδίαση είναι πολύ πιθανόν να οδηγήσουν σε αναποτελεσματική διαχείριση των δεδομένων, κάτι που μακροπρόθεσμα θα κοστίσει, τόσο χρόνο, όσο και χρήμα.
- **Στάδιο 2: Καταχώρηση των δεδομένων στη βάση (data entry):** Η καταχώρηση δεδομένων δεν απαιτεί την απασχόληση εξειδικευμένου προσωπικού. Γίνεται είτε αυτοματοποιημένα, είτε με συμπλήρωση φορμών σε ειδικές εφαρμογές *διεπαφής με το χρήστη (user interfaces – UI)*.
- **Στάδιο 3: Διαχείριση της βάσης δεδομένων και επεξεργασία των δεδομένων:** Η διαχείριση των δεδομένων της βάσης πραγματοποιείται ανά πάσα στιγμή μετά την υλοποίησή της και εφαρμόζεται στα δεδομένα εκείνα που έχουν ήδη καταχωρηθεί. Η κυριότερη μορφή διαχείρισης των δεδομένων αφορά στην προβολή δεδομένων που πληρούν συγκεκριμένα κριτήρια αναζήτησης. Τα κριτήρια αυτά μπορεί, είτε να έχουν τεθεί εξ αρχής σε μια εφαρμογή που προσπελαύνει τη βάση δεδομένων (3-tier) και επιστρέφει το αποτέλεσμα με τυποποιημένη μορφοποίηση σε μορφή αναφοράς, είτε να τίθεται κατά τη διάρκεια εκτέλεσης της εφαρμογής. Άλλη μορφή διαχείρισης, εξ ίσου σημαντική, είναι η αλλαγή των δεδομένων (τροποποίηση / ενημέρωση / διαγραφή).

Το ενδιαφέρον μας εστιάζεται στο πρώτο Στάδιο και ιδιαίτερα στη σχεδίαση των βάσεων δεδομένων.

2.3.2. Επίπεδα Σχεδίασης Βάσεων Δεδομένων

Η σχεδίαση μιας βάσης δεδομένων μπορεί να χωριστεί γενικά σε τρία διακριτά εννοιολογικά επίπεδα: το φυσικό επίπεδο, το λογικό επίπεδο, και το επίπεδο όψης.

Στο **φυσικό επίπεδο (physical level)**, σχεδιάζεται και αναλύεται ο τρόπος με τον οποίο τα δεδομένα της βάσης θα αποθηκεύονται στο ΣΔΒΔ. Στο επίπεδο αυτό, καθορίζονται, μεταξύ άλλων, οι τύποι δεδομένων που θα χρησιμοποιηθούν.

Στο **εννοιολογικό ή λογικό επίπεδο (conceptual / logical level)**, καθορίζεται ο τρόπος που τα δεδομένα και οι σχέσεις που υφίστανται ανάμεσά τους θα μοντελοποιηθούν στη βάση του συστήματος.

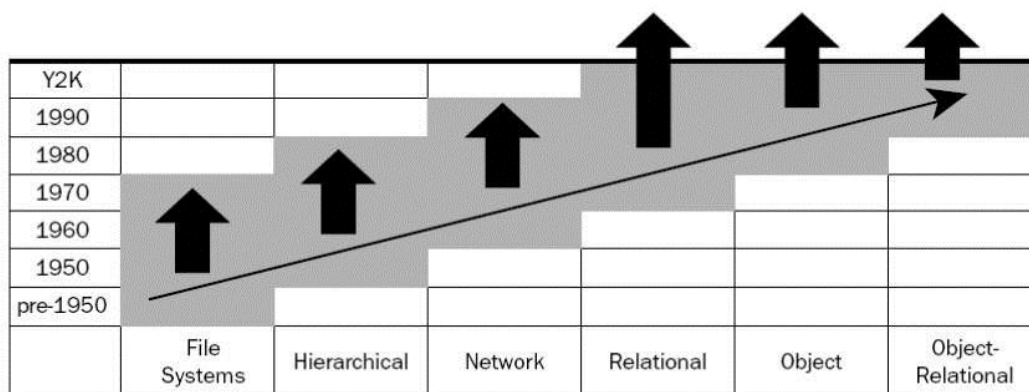
Τέλος, στο **επίπεδο όψης (view level)**, καθορίζεται ο τρόπος που ένα μέρος των δεδομένων θα προβάλλεται στο χρήστη σύμφωνα με καθορισμένα κριτήρια που εκείνος θέτει με άμεσο ή έμμεσο τρόπο.

Εμείς επικεντρωνόμαστε στο λογικό επίπεδο γιατί εκεί γίνεται κυρίως η μοντελοποίηση της βάσης δεδομένων.

2.3.3. Μοντέλα Σχεδίασης Εννοιολογικού-Λογικού Επιπέδου

Τα μοντέλα σχεδίασης εννοιολογικού-λογικού επιπέδου δεν αφορούν απλά στους τρόπους οργάνωσης δεδομένων, αλλά προχωρούν στο να ορίσουν και τις πράξεις που τελούνται στα δεδομένα. Έτσι, για το σχεσιακό μοντέλο, για παράδειγμα, υφίστανται οι τελεστές της επιλογής (*select*), της προβολής (*project*), της σύζευξης (*join*) και της διαίρεσης (*division*).

Η ιστορική εξέλιξη των μοντέλων φαίνεται στην ακόλουθη εικόνα :



Εικόνα 2-7 : Ιστορική Εξέλιξη των Μοντέλων ΒΔ

Επίπεδο Μοντέλο (Flat Model)

Το επίπεδο μοντέλο ή μοντέλο **πίνακα (table)** ή μοντέλο **επίπεδου αρχείου (flat file)** είναι ο παλαιότερος τρόπος οργάνωσης δεδομένων. Πρόκειται για συλλογή ακολουθιών δεδομένων, διαχωρισμένα με ειδικούς χαρακτήρες (π.χ. κενό, tab, κόμμα, ερωτηματικό) που μπορούν να διερευνηθούν σειριακά για τον εντοπισμό τους. Η σειριακή αναζήτηση δεδομένων είναι αναποτελεσματική, αλλά μπορεί να είναι χρήσιμη για μικρούς καταλόγους και για απλές δομές δεδομένων. Σε αυτό το μοντέλο δεν υπάρχει η δυνατότητα καθορισμού του τύπου των δεδομένων, καθώς η οργάνωση του κάθε αρχείου είναι ελεύθερη, χωρίς έλεγχο κανόνων από κάποιο σύστημα. Η διαχείριση γίνεται μέσω προγραμμάτων που διαβάζουν και γράφουν στα αρχεία των δεδομένων.

Προβλήματα :

- ταυτόχρονη εγγραφή από περισσότερες από μία εφαρμογές,
- απώλεια δεδομένων λόγω κατάρρευσης μιας εφαρμογής κατά τη διάρκεια προσπέλασης του αρχείου.

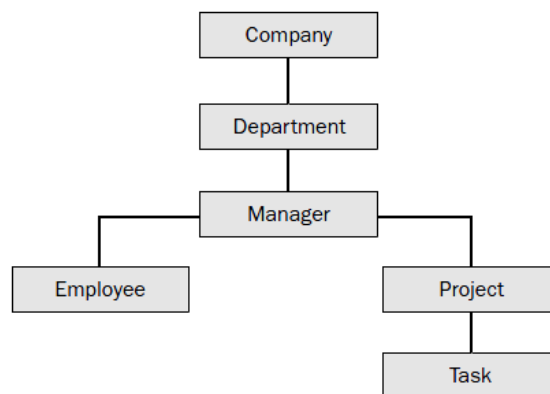
Ιεραρχικό Μοντέλο (Hierarchical Model)

Στο ιεραρχικό μοντέλο τα δεδομένα οργανώνονται σε δομές που ακολουθούν τη λογική του ανεστραμμένου δένδρου. Τις δομές αυτές μπορούμε να τις θεωρήσουμε *πίνακες εγγραφών (record tables)*, οι οποίοι χαρακτηρίζονται από *δείκτες (pointers)*. Οι σχέσεις μεταξύ των πινάκων είναι του τύπου *πρόγονος-απόγονος (parent-child)*, επειδή δημιουργούνται σχέσεις πολλαπλότητας «ένα προς πολλά». Αυτό καθίσταται δυνατό μέσω των δεικτών που φέρουν οι πίνακες. Ο πρώτος πίνακας-πρόγονος καλείται *ρίζα (root)*, και έχει έναν ή περισσότερους πίνακες – απογόνους. Κάθε πίνακας, μπορεί να έχει έναν ή περισσότερους απογόνους, αλλά **μόνο έναν πρόγονο**. Η αναζήτηση είναι **μονοσήμαντη** μέσω των «κλαδιών», ξεκινώντας από τη ρίζα. Στο ιεραρχικό μοντέλο η ακεραιότητα αναφοράς των δεδομένων είναι εγγενής και δεν χρειάζεται ιδιαίτερος μηχανισμός για να επιτευχθεί.

Ένα τυπικό παράδειγμα οργάνωσης σύμφωνα με την ιεραρχικό μοντέλο είναι το πρότυπο καταλόγου *Lightweight Directory Access Protocol – LDAP* ή το πρότυπο *eXtended Markup Language – XML*.

Τα σοβαρότερα μειονεκτήματα είναι :

- η ανάγκη εκκίνησης της αναζήτησης από τη ρίζα
- η μεγάλη δυσκολία αλλαγής της δομής του «δέντρου», η οποία επηρεάζει, όχι μόνο την ίδια τη βάση δεδομένων, αλλά και τις εφαρμογές που τη χειρίζονται.
- λόγω της στενής σχέσης των πινάκων, όταν διαγράφεται ένας πρόγονος, πρέπει να διαγραφούν και οι απόγονοί του.

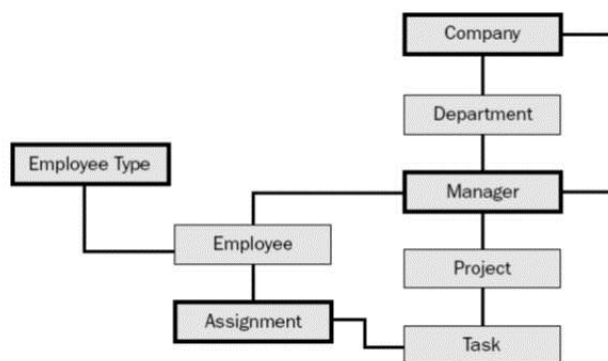


Εικόνα 2-8: Οργάνωση δεδομένων με το Ιεραρχικό Μοντέλο

Μοντέλο Δικτύου (Network Model)

Το μοντέλο δικτύου οργανώνει τα δεδομένα ακολουθώντας δύο βασικές έννοιες : τις *εγγραφές (records)* και τα *σύνολα (sets)*. Οι εγγραφές περιέχουν πεδία ενώ τα σύνολα (που

δεν σχετίζονται με τη μαθηματική έννοια του συνόλου) ορίζουν σχέσεις πολλαπλότητας «ένα προς πολλά» μεταξύ των εγγραφών. Έτσι, Οι έγγραφες συνδέονται μεταξύ τους δημιουργώντας ένα δίκτυο συνδέσεων, και ουσιαστικά συνδυάζονται μεμονωμένες ιεραρχικές βάσεις δεδομένων σε μια μεγαλύτερη και με αυτόν τον τρόπο το μοντέλο δικτύου αποτελεί επέκταση του ιεραρχικού μοντέλου γιατί επιτρέπει σε κάθε απόγονο να έχει πολλούς προγόνους, δηλ. επιτρέπει τη δημιουργία σχέσεων πολλαπλότητας «πολλά προς πολλά». Η πλοήγηση δεν απαιτεί την έναρξη από κάποια ρίζα, αλλά **μπορεί να αρχίσει από οποιαδήποτε πίνακα**.



Εικόνα 2-9: Οργάνωση δεδομένων με το Μοντέλο Δικτύου

Σχεσιακό Μοντέλο (Relational Model)

Το σχεσιακό μοντέλο βασίζεται στην Κατηγορηματική Λογική (Predicate Logic) και τη Θεωρία Συνόλων (Set Theory). Τα βασικά στοιχεία του μοντέλου είναι οι οντότητες (entities), οι σχέσεις ή συσχετίσεις (relations), τα χαρακτηριστικά (attributes) και τα πεδία τιμών τους (domains). Στο σχεσιακό μοντέλο οι οντότητες και οι συσχετίσεις αποδίδονται με πίνακες, και σε αντίθεση με το ιεραρχικό και το μοντέλο δικτύου, η έννοια της ιεραρχίας και δεν αποτελεί το κυρίαρχο χαρακτηριστικό.

Το μοντέλο αυτό θα αποτελέσει το αντικείμενο μελέτης μας στα επόμενα κεφάλαια.

Αντικειμενοστραφές μοντέλο (Object Oriented Model)

Το μοντέλο αυτό προέκυψε από την ανάγκη άμεσης πρόσβασης στο μοντέλο της βάσης δεδομένων από σύγχρονες τεχνικές αντικειμενοστραφούς προγραμματισμού (object oriented programming). Αυτό έχει σαν αποτέλεσμα την άμεση προσπέλαση των δεδομένων από της εφαρμογές και τις καθιστά **αποδοτικότερες σε νέους τύπους δεδομένων** όπως ήχου και εικόνας (σταθερής ή video). Οι αντικειμενοστραφείς ΒΔ προσπελούνται άμεσα από γλώσσες όπως η Java και η C++, πράγμα που τις κάνει **10 έως 1000 φορές ταχύτερες** από τις αντίστοιχες σχεσιακές.

Αν και σχεδιαστικά οι αντικειμενοστραφείς ΒΔ ξεπερνούν πολλές αδυναμίες των σχεσιακών, πρακτικά παρουσιάζουν το μειονέκτημα ότι η διείσδυσή τους στην αγορά είναι μικρή σε σύγκριση με τις σχεσιακές βάσεις. Αυτό σημαίνει λιγότερους εξειδικευμένους προγραμματιστές και μη διασφάλιση εκτέλεσης των εφαρμογών σε διαφορετικές πλατφόρμες λόγω μη επαρκούς τυποποίησης (standardization).

Τέλος, υφίσταται και ένα υβριδικό μοντέλο *Συσχέτισης Αντικειμένων (Object Related Model)* που προτάθηκε από την Oracle στο τέλος της δεκαετίας του 1990. Το μοντέλο αυτό ενσωματώνει αντικειμενοστραφή χαρακτηριστικά σε μια σχεσιακή βάση δεδομένων, επιτρέποντας τη δημιουργία νέων τύπων δεδομένων, αν και η βασική αρχιτεκτονική παραμένει σχεσιακή.

2.3.4. Λειτουργίες και Γλώσσες

Ο καθορισμός και η διαχείριση της δομής της βάσης πραγματοποιούνται με τη βοήθεια ειδικών γλωσσών προγραμματισμού. Οι γλώσσες αυτές, χαρακτηρίζονται ως *γλώσσες τέταρτης γενεάς (4th Generation Languages, 4GL)*, διότι σε αντίθεση με τις συμβατικές γλώσσες προγραμματισμού δεν υλοποιούν πολύπλοκες λειτουργίες χαμηλού επιπέδου, αλλά το μόνο που κάνουν είναι να ανακτούν δεδομένα από τη βάση του συστήματος, χωρίς ο χρήστης να γνωρίζει πως λαμβάνει χώρα αυτή η διαδικασία. Εκτός από την ανάκτηση των δεδομένων, οι γλώσσες αυτές προσφέρουν και πιο προχωρημένες λειτουργίες, όπως είναι η δημιουργία νέων δομών δεδομένων μέσα στη βάση, η διαγραφή και η τροποποίηση αυτών, και ο ορισμός συσχετίσεων μεταξύ των στοιχείων της βάσης.

Οι λειτουργίες αυτές μπορούν να διαχωριστούν σε δύο κατηγορίες: σ' αυτές που τροποποιούν τη **δομή της βάσης**, και σ' εκείνες που τροποποιούν το **περιεχόμενο της βάσης**, δηλαδή τα δεδομένα που αποθηκεύονται σε αυτή. Αυτό σημαίνει ότι υφίστανται δύο επίπεδα διαχείρισης, και γι' αυτό χρησιμοποιούνται διαφορετικές γλώσσες προγραμματισμού: μια γλώσσα που είναι υπεύθυνη για τον ορισμό και την τροποποίηση της δομής της βάσης και η οποία ονομάζεται *γλώσσα ορισμού δεδομένων (Data Definition Language, DDL)*, και μια γλώσσα που είναι υπεύθυνη για τη διαχείριση των δεδομένων της βάσης, και η οποία ονομάζεται *γλώσσα διαχείρισης δεδομένων (Data Manipulation Language, DML)*.

2.4. Ερωτήσεις Κατανόησης

1. Ποια είναι η βασική αρχιτεκτονική φυσικού επιπέδου για τα σύγχρονα ΣΔΒΔ ;
2. Ποιες επιμέρους αρχιτεκτονικές υπάρχουν για τα μοντέλα Client/Server;
3. Ποια τα πλεονεκτήματα και τα μειονεκτήματα σε καθεμία από τις αρχιτεκτονικές ΣΔΒΔ;
4. Ποιες είναι οι επιπτώσεις στο κόστος κατά την κλιμάκωση μιας εφαρμογής με αρχιτεκτονική 2-tier σε σχέση με μία 3-tier ;
5. Ποια είναι τα στάδια στον κύκλο ζωής μιας ΒΔ;
6. Ποια είναι τα επίπεδα σχεδίασης και σε τι αφορά το καθένα;
7. Ποιες είναι οι αρχές της καλής σχεδίασης των ΒΔ;
8. Τι εννοούμε με τους όρους περιττά και ασυνεπή ;
9. Τι είναι η κανονικοποίηση;
10. Πώς επιτυγχάνεται η ασφάλεια των δεδομένων σε μια ΒΔ;
11. Πώς διασφαλίζεται η μη απώλεια δεδομένων από ταυτόχρονες προσπελάσεις;
12. Πώς διασφαλίζεται η μη απώλεια δεδομένων από κατάρρευση;
13. Ποια είναι τα μοντέλα λογικής σχεδίασης;

14. Τοποθετήστε χρονικά της εξέλιξη των μοντέλων λογικής σχεδίασης.
15. Ποιο το χαρακτηριστικό του μοντέλου επίπεδου αρχείου;
16. Ποια τα προβλήματα του μοντέλου επίπεδου αρχείου;
17. Ποια τα χαρακτηριστικά και τα μειονεκτήματα του ιεραρχικού μοντέλου;
18. Σε τι διαφέρει το ιεραρχικό από το μοντέλο δικτύου;
19. Ποια τα χαρακτηριστικά του σχεσιακού μοντέλου;
20. Ποια τα μειονεκτήματα και τα πλεονεκτήματα του αντικειμενοστραφούς μοντέλου έναντι του σχεσιακού;
21. Τι είναι οι γλώσσες τέταρτης γενιάς ;
22. Ποια είδη λειτουργιών εντοπίζονται σε μια σχεσιακή ΒΔ;

Σημειώματα

Σημείωμα αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Κωστής Πηγουνάκης, 2015. Κωστής Πηγουνάκης. «Εισαγωγή στις Βάσεις Δεδομένων II». Έκδοση: 1.0. Ρέθυμνο 2014.

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγο Έργο 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή δασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

