

Εργαστήριο #10

Από τα προηγούμενα εργαστήρια...

...θα χρειαστείτε ορισμένες από τις οδηγίες μορφοποίησης CSS (ανατρέξτε στις εκφωνήσεις του 8^{ου} και 9^{ου} εργαστηρίου).

Οδηγίες

Στη δυναμική δημιουργία ιστοσελίδων σχεδόν πάντα χρησιμοποιείται ένα **σύστημα σχεσιακής βάσης δεδομένων** (relational database) για την αποθήκευση της πληροφορίας που εμφανίζεται. Η PHP διαθέτει τις κατάλληλες βιβλιοθήκες-επεκτάσεις (extensions) για να συνδεθείτε σε μια τέτοια βάση και να ανακτήσετε την επιθυμητή πληροφορία.



Οι σχεσιακές βάσεις δεδομένων καλύπτονται εδώ κατά στοιχειώδη τρόπο και μόνο. Την πλήρη θεωρία θα διδαχθείτε σε μεγαλύτερα εξάμηνα. Μπορείτε να ανατρέξετε στην ύλη της “Εισαγωγής στην Επιστήμη των Η/Υ” για μια εισαγωγή στο αντικείμενο των βάσεων δεδομένων και της γλώσσας SQL.

Στο εργαστήριο θα χρησιμοποιηθεί η βάση δεδομένων MySQL, η οποία βρίσκεται εγκατεστημένη στον server δοκιμών του εργαστηρίου. Κωδικοί πρόσβασης στη βάση θα δοθούν κατά τη διεξαγωγή του εργαστηρίου.

Στο σημερινό εργαστήριο θα ανακτήσετε πληροφορία που υπάρχει ήδη στη βάση δεδομένων σε κωδικοποίηση UTF-8: αρκεί λοιπόν μετά την ανάκτηση να απεικονίσετε την πληροφορία αυτή ως ιστοσελίδα!

⇒ 1. Μια επανάληψη-αστραπή για τα χαρακτηριστικά των σχεσιακών βάσεων δεδομένων και τα πολύ απλά ερωτήματα SQL που θα χρειαστείτε στο σημερινό εργαστήριο: διαβάστε τα **Παράρτηματα Α και Β!**

⇒ 2. Διαβάστε το **Παράρτημα Γ** για να μάθετε πώς μπορείτε να ανακτήσετε δεδομένα από τη βάση μέσω PHP. Στη συνέχεια, δημιουργήστε από τα δεδομένα αυτά ιστοσελίδα με έναν πίνακα HTML με τις εξής στήλες: Τίτλος, Συγγραφείς, Έκδοση, Εκδότης, ISBN. **Μορφοποιήστε την ιστοσελίδα και τον πίνακα όπως ακριβώς στο 8^ο εργαστήριο.**

⇒ 3. Τα δεδομένα δεν πρέπει υποχρεωτικά να εμφανίζονται ως πίνακας! Δείτε το **Παράρτημα Δ** για οδηγίες δημιουργίας μιας εναλλακτικής ιστοσελίδας με τα δεδομένα των βιβλίων. Συμβουλευτείτε επίσης την εκφώνηση του 9^{ου} εργαστηρίου.

Παράρτημα Α: Σχεσιακές (relational) βάσεις δεδομένων.

Στο εργαστήριο θα χρησιμοποιήσετε μια σχεσιακή βάση δεδομένων, όπου η πληροφορία μπορεί να αναπαρασταθεί σε μορφή πινάκων, όπως στο **σχήμα 1**:

Πίνακας: books

id	title	authors	isbn
...
...
...
...

γραμμή (row):
πληροφορίες για ένα βιβλίο

π.χ. η ιδιότητα "title" για το συγκεκριμένο βιβλίο

Σχήμα 1

Συνοπτικά:

- Κάθε **γραμμή** του πίνακα αντιστοιχεί σε μια ξεχωριστή **οντότητα πληροφορίας**, π.χ. αν ο πίνακας περιγράφει **πληροφορίες βιβλίων**, τότε κάθε γραμμή (row) μπορεί να περιέχει πληροφορία για **ένα βιβλίο**.
- Στις **στήλες** του πίνακα υπάρχουν οι **ιδιότητες** κάθε οντότητας, π.χ. στον ίδιο πίνακα **βιβλίων** μπορεί να υπάρχει μια στήλη με τον **τίτλο** του κάθε βιβλίου, μια στήλη με τους **συγγραφείς** του κλπ.
- Συνήθως μια στήλη περιέχει έναν **αναγνωριστικό αριθμό (id)**, διαφορετικό για κάθε γραμμή. Αυτή η αρίθμηση μπορεί να χρησιμοποιηθεί για τη συγκρότηση **σχέσεων** μεταξύ πινάκων (στο επόμενο εργαστήριο).

Στο σημερινό εργαστήριο θα χρησιμοποιήσετε έναν υπάρχοντα πίνακα πληροφοριών για διδακτικά βιβλία που μπορείτε να επιλέξετε στο Α' εξάμηνο. Ο πίνακας ονομάζεται books και κάθε γραμμή του περιγράφει ένα βιβλίο. Οι στήλες του πίνακα books είναι:

1. id: αναγνωριστικό γραμμής (δεν θα το χρησιμοποιήσετε στο εργαστήριο αυτό).
2. title: ο τίτλος του βιβλίου.
3. code: ο κωδικός του βιβλίου στον "Εύδοξο".
4. authors: οι συγγραφείς του βιβλίου.
5. publisher: ο εκδοτικός οίκος του βιβλίου.
6. isbn: ο αριθμός ISBN του βιβλίου.
7. edition: ο αριθμός έκδοσης του βιβλίου.
8. imagelink: η διεύθυνση URL μιας εικόνας (thumbnail) του βιβλίου.

Παράρτημα Β: Structured Query Language (SQL).

Μέσω της PHP μπορείτε να συνδεθείτε σε μια βάση δεδομένων για να αναζητήσετε πληροφορίες ή να εκτελέσετε εντολές. Η πρότυπη γλώσσα για την εισαγωγή ερωτημάτων και εκτέλεση εντολών σε σχεσιακές βάσεις δεδομένων ονομάζεται **SQL (Structured Query Language)**.

Παραδείγματα απλών ερωτήσεων ανάκτησης πληροφορίας:

Εάν θέλετε να δείτε **όλες τις στήλες** του πίνακα με όνομα books για **όλες τις γραμμές** του, η σύνταξη της ερώτησης SQL θα ήταν:

```
SELECT * FROM books;
```

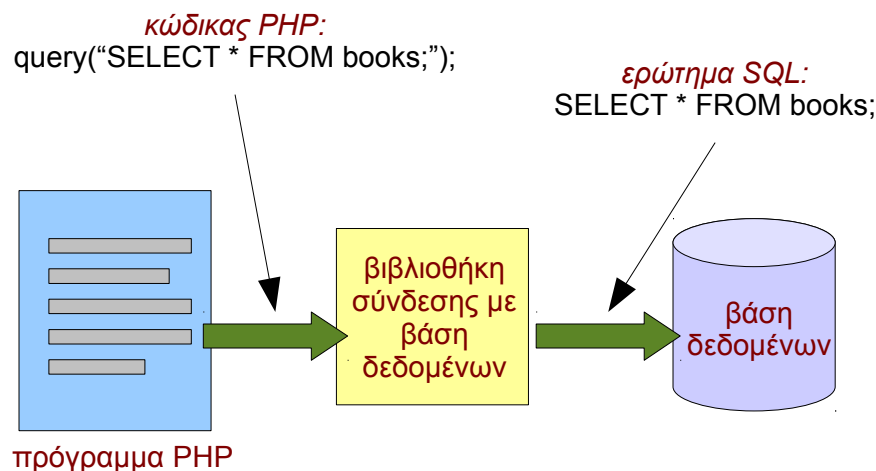
Συνήθως δεν θέλουμε την πληροφορία κάθε στήλης, αλλά μόνο ορισμένων από αυτές. Για να δείτε π.χ. μόνο τις στήλες title και authors, για όλες τις γραμμές του πίνακα books:

```
SELECT title,authors FROM books;
```

Και στις δύο προηγούμενες περιπτώσεις ερωτήσεων, η βάση δεδομένων θα επιστρέφει το σύνολο των γραμμών που υπάρχουν μέσα στον πίνακα, τη μια γραμμή μετά την άλλη. Στο επόμενο εργαστήριο θα δούμε πώς μπορούμε να επιλέξουμε ορισμένες γραμμές μόνο, με βάση κάποιο κριτήριο.

Παράρτημα Γ: Προσπέλαση βάσης δεδομένων μέσω PHP.

Η PHP διαθέτει μια σειρά από βιβλιοθήκες-επεκτάσεις (libraries-extensions) για την επικοινωνία με σχεσιακές βάσεις δεδομένων. Σκοπός των βιβλιοθηκών αυτών είναι να παρέχουν ένα API (σετ συναρτήσεων) που θα επιτρέψουν στο πρόγραμμά μας να συνδεθεί και να στείλει εντολές SQL στη βάση δεδομένων, καθώς και να λάβει τα δεδομένα που επιστρέφονται από τη βάση (σχήμα 2).



Σχήμα 2

Αν και η SQL είναι πρότυπη γλώσσα και υλοποιείται σε όλα τα συστήματα σχεσιακών βάσεων δεδομένων, κάθε βάση έχει τις δικές της “ιδιομορφίες”. Έτσι, η PHP διαθέτει ξεχωριστές βιβλιοθήκες ή drivers για κάθε γνωστό σύστημα βάσης.

Η βιβλιοθήκη PHP Data Objects (PDO).

(<http://www.php.net/manual/en/book.pdo.php>)

Η βιβλιοθήκη-επέκταση **PDO** της PHP παρέχει μια μορφή “αφαίρεσης” κατά την πρόσβαση διαφορετικών βάσεων δεδομένων: προσφέρει ένα **κοινό API** για κάθε γνωστό σύστημα βάσης δεδομένων. Φυσικά, υπάρχουν ιδιαιτερότητες που δεν μπορούν να καλυφθούν κατά έναν κοινό τρόπο· τέτοιες ιδιαιτερότητες όμως δεν θα συναντήσουμε στο εργαστήριο.

Η βιβλιοθήκη PDO είναι γραμμένη με **αντικειμενοστρεφή** τρόπο (ναι, η PHP διαθέτει και τέτοια χαρακτηριστικά!). Στις οδηγίες που ακολουθούν θα δείτε τη χρήση αντικειμένων και μεθόδων κλάσεων. Ένας πολύ γρήγορος οδηγός χρήσης (όχι συγγραφής!) κλάσεων ακολουθεί:

- Πώς δημιουργούμε το αντικείμενο μιας κλάσης;
`$obj = new aClassName (param1 , param2 , . . .) ;`
- Πώς καλούμε μια συνάρτηση (μέθοδο) ενός αντικειμένου;
`$obj->funcName () ;`

1. Σύνδεση με τη βάση δεδομένων.

Πριν οποιαδήποτε ερώτηση ή άλλη εντολή προς τη βάση δεδομένων, πρέπει να ζητήσουμε από την PHP να συνδεθεί με τη βάση, η οποία εδώ ονομάζεται labdb:

```
$db = new PDO('mysql:host=localhost;dbname=labdb',  
              'username', 'password');
```

Ο πιο πάνω κώδικας δημιουργεί ένα νέο αντικείμενο της κλάσης **PDO**, το οποίο αντιπροσωπεύει τη σύνδεση με την επιθυμητή βάση δεδομένων. Τα ορίσματα για την κατασκευή του αντικειμένου είναι:

- Το πρώτο όρισμα (“Data Source Name” - DSN) περιγράφει τη βάση που θα συνδεθείτε: το είδος της βάσης (MySQL), τον υπολογιστή (ίδιο με τον webserver, localhost) και το όνομα της βάσης για το εργαστήριο (labdb) .
- Ο κωδικός χρήστη της βάσης (δεν σχετίζεται απαραίτητα με λογαριασμό χρήστη στο σύστημα, **αντικαταστήστε το με το όνομα που θα σας δοθεί**).
- Το κωδικό κλειδί για τον χρήστη της βάσης (παρατηρήστε ότι το κλειδί αναγράφεται μη κωδικοποιημένο! Πρέπει να εξασφαλίσετε ότι δεν μπορούν να δουν άλλοι το πρόγραμμα PHP! **Αντικαταστήστε το με το όνομα που θα σας δοθεί**).


2. Τερματισμός σύνδεσης.

Όταν ολοκληρώσετε την επικοινωνία με τη βάση, καλό είναι να κλείσετε τη σύνδεση:

```
$db = null;
```

3. Χειρισμός σφαλμάτων σύνδεσης.

Η δημιουργία του αντικειμένου τύπου PDO μπορεί να προκαλέσει ένα **σφάλμα (exception)** σύνδεσης, όταν η βάση δεν είναι διαθέσιμη για οποιοδήποτε λόγο. Όταν συμβεί αυτό, η PHP διακόπτει την εκτέλεση του προγράμματος και αρχίζει να επιστρέφει από τις συναρτήσεις (αν η εκτέλεση βρίσκεται μέσα σε συνάρτηση). Αν εσείς δεν “πιάνετε” (εντολή **catch**) πουθενά το σφάλμα, το πρόγραμμα διακόπτεται οριστικά και η PHP τυπώνει ένα μήνυμα λάθους.



Επειδή το μήνυμα αυτό **δε ενδιαφέρει** τον τελικό χρήστη (περιέχει τεχνικές λεπτομέρειες) και -ακόμα χειρότερα- **μπορεί να αποκαλύψει απόρρητες πληροφορίες** στον κακόβουλο χρήστη (όπως π.χ. με ποιο username και password δοκιμάσατε να συνδεθείτε στη βάση), **πρέπει πάντα να το “πιάσετε”!**

Το πώς θα γίνει αυτό φαίνεται στο επόμενο υπόδειγμα κώδικα, **το οποίο πρέπει να αποτελέσει τη βάση για το δικό σας πρόγραμμα.**


```
try {
    $db = new PDO('mysql:host=localhost;dbname=labdb',
                 'username', 'password');

    // βάλτε εδώ οποιαδήποτε λειτουργία της βάσης

    $db = null;
}
catch (PDOException $e) {
    echo 'database error';
}
```

4. Απλά SQL ερωτήματα (SELECT).

Στο σημερινό εργαστήριο θα χρησιμοποιήσετε **στατικά** ερωτήματα **SELECT** προς τη βάση δεδομένων. Στατικά σημαίνει προκαθορισμένα ερωτήματα που **δεν εξαρτώνται από τιμές μεταβλητών ή από δεδομένα που έδωσε ο χρήστης.**



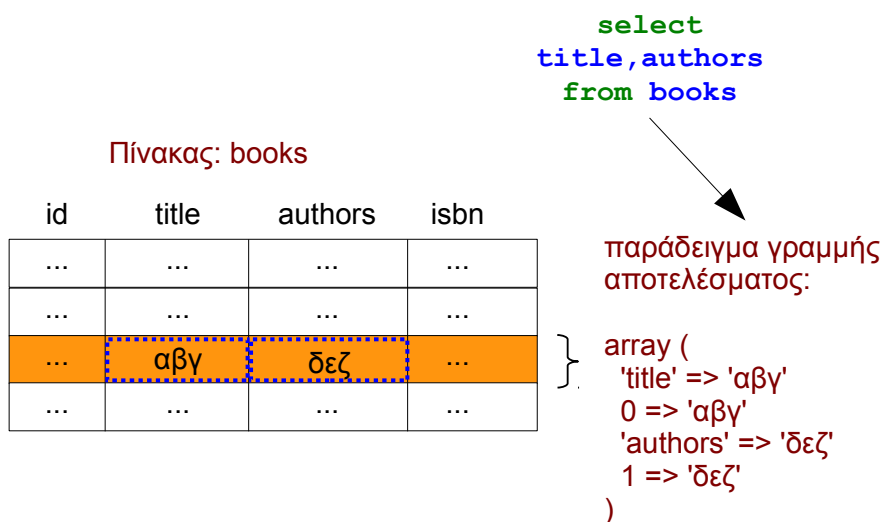
Αν το ερώτημα είναι **δυναμικό**, τότε η μέθοδος που ακολουθεί δεν είναι η πιο κατάλληλη και ίσως δημιουργεί **προβλήματα ασφάλειας!** Δυναμικά ερωτήματα θα δούμε στο επόμενο εργαστήριο.

Ένα στατικό SQL ερώτημα **SELECT**, για την ανάκτηση μίας ή περισσότερων γραμμών από ένα πίνακα της βάσης, εκτελείται με τη βοήθεια της μεθόδου **query()** του αντικειμένου τύπου **PDO** που έχετε δημιουργήσει κατά τη σύνδεση με τη βάση:

```
// $db είναι το αντικείμενο τύπου PDO της σύνδεσης  
$st = $db->query('select title,authors from books');
```

Η συνάρτηση (μέθοδος) **query()** δέχεται ως όρισμα το ερώτημα SQL ως string και επιστρέφει ένα αντικείμενο τύπου **PDOStatement**, μέσω του οποίου θα ανακτήσουμε τα αποτελέσματα.

Κάθε γραμμή (row) αποτελεσμάτων επιστρέφεται ως ένα array της PHP, με δείκτες τόσο **αριθμητικούς** (οι στήλες με τη σειρά που τις ζητήσατε στο ερώτημα SELECT) όσο και με το **όνομα** της στήλης, όπως δείχνει το παράδειγμα του **σχήματος 3**:



Σχήμα 3

α) Ανάκτηση όλων των αποτελεσμάτων ως λίστα.

Στο παράδειγμα που ακολουθεί, χρησιμοποιούμε τη μέθοδο **fetchAll()** του αντικειμένου **PDOStatement** για να ανακτήσουμε το σύνολο των γραμμών ως λίστα, την οποία μπορούμε να διασχίσουμε και να επεξεργαστούμε στη συνέχεια:

```
$st = $db->query('select title,authors from books');  
$result = $st->fetchAll();  
foreach ($result as $row) {  
  // επεξεργασία μιας γραμμής αποτελεσμάτων  
  // π.χ. echo $row['authors'];  
  // ή echo $row[0];  
}
```

β) Ανάκτηση των αποτελεσμάτων γραμμή προς γραμμή.

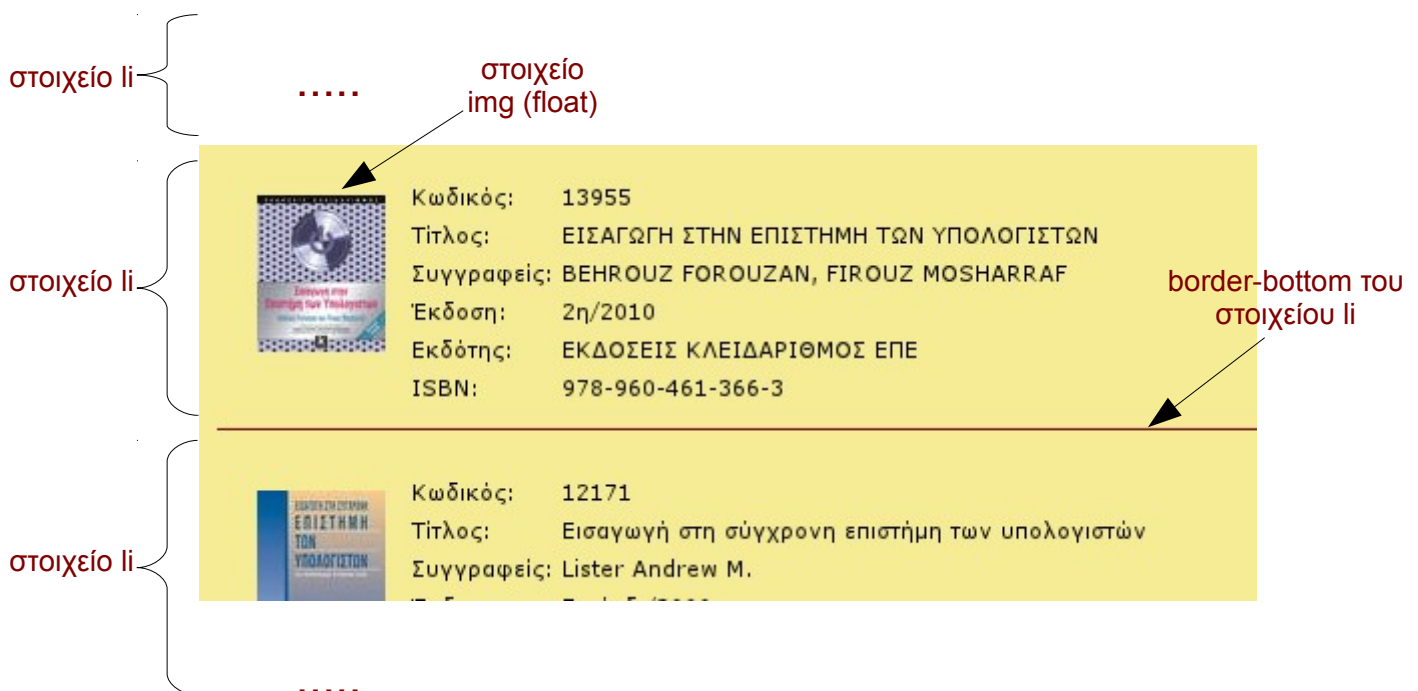
Μπορείτε επίσης να προσπελάσετε μία-μία τις γραμμές του αποτελέσματος με τη μέθοδο `fetch()`, όπως στο επόμενο παράδειγμα:

```
$st = $db->query('select title,authors from books');  
while ($row = $st->fetch()) {  
    // επεξεργασία μιας γραμμής αποτελεσμάτων  
    // π.χ. echo $row['authors'];  
    // ή echo $row[0];  
}
```

Παράρτημα Δ: Εμφάνιση δεδομένων ως λίστα HTML.

Το ζητούμενο είναι να εμφανίσετε μια ιστοσελίδα με όλα τα βιβλία της βάσης, μορφοποιημένη όπως φαίνεται στο **σχήμα 4**:

λίστα ul



Σχήμα 4

- Για να αποφύγετε την εμφάνιση κουκίδων (bullets) της λίστας `ul`, χρησιμοποιήστε την ιδιότητα CSS `list-style-type: none`.
- Επίσης, εξουδετερώστε σε μηδέν το αριστερό περιθώριο (margin) και παραγέμισμα (padding) του στοιχείου `ul`. Αυτό πρέπει να γίνει γιατί η λίστα έχει εξ'ορισμού αριστερό περιθώριο για τις κουκίδες.

Τι συμβαίνει όταν τα γράμματα είναι πολύ μικρά; Οι εικόνες ως floats βγαίνουν έξω από το στοιχείο li που τις περιέχει! Για να το αποφύγετε αυτό και να αναγκάσετε το στοιχείο li να περικλείει την εικόνα του, πρέπει να προσθέσετε ως τελευταίο στοιχείο **μέσα στο li**, ένα στοιχείο με την ιδιότητα **clear:both** (βλ. 9^ο εργαστήριο). Ποιο στοιχείο όμως; Θα μπορούσατε να βάλετε ένα κενό <div>. Συνήθως όμως εισάγεται τεχνητά ένα μη ορατό στοιχείο με τον επιλογέα **:after**, ως εξής:

```
li:after {  
  content: ".";  
  display: block;  
  height: 0;  
  clear: both;  
  visibility: hidden;  
}
```

Ο επιλογέας **li:after** εισάγει μέσα στο στοιχείο li, και αμέσως μετά το τελευταίο παιδί του, ένα μη ορατό στοιχείο απλού κειμένου με την ιδιότητα **clear:both**.