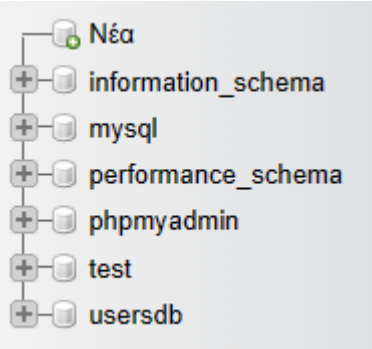
ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ – ΟΔΗΓΙΕΣ ΚΑΤΑΣΚΕΥΗΣ

Ενεργοποιώ τον XAMPP. Ανοίγω το περιβάλλον PhpMyAdmin στον XAMPP[1] και εφόσον δεν υπάρχει ήδη δημιουργώ μια νέα Βάση Δεδομένων με το όνομα **usersdb.** Στη συνέχεια πατώ στον πιο κάτω σύνδεσμο http://users.sch.gr/mfanarioti/usersdb.sql για να κατεβάσω το σχετικό αρχείο.

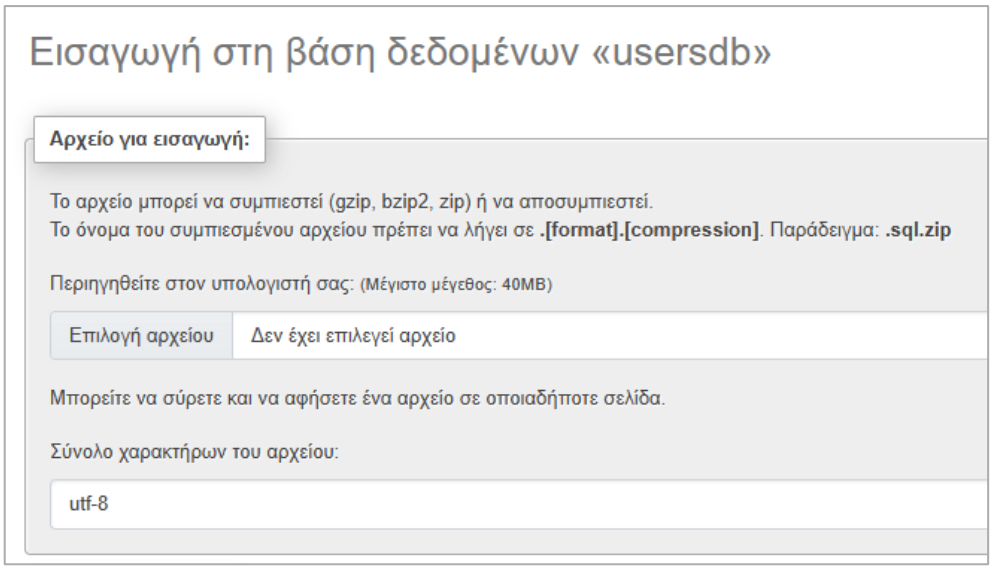
Στο phpMyAdmin και στη λίστα που βρίσκεται στα αριστερά κάνω κλικ στο όνομα της βάσης **usersdb**.



Στη συνέχεια πατώ στο κουμπί **Εισαγωγή (Import)**



Στην πιο κάτω καρτέλα πατώ στο κουμπί **Επιλογή Αρχείου**. Εντοπίζω μέσα στον υπολογιστή μου το αρχείο **usersdb.sql** που κατέβασα πιο πριν και πατώ στο κουμπί **Εισαγωγή** που βρίσκεται στο τέλος της καρτέλας.



[1] Για να ανοίξω το περιβάλλον phpMyAdmin στη γραμμή διεύθυνσης του Browser πληκτρολογώ http://localhost/phpmyadmin

Με το τέλος της διαδικασίας μέσα στη βάση πρέπει να έχει εισαχθεί ένας πίνακας με το όνομα **users** και τα στοιχεία που φαίνονται πιο κάτω:

id	username	password	firstname	lastname
2	user2	user2pass	John	Papas
3	user3	user3pass	Boy	George
4	user4	user4pass	Celine	Dion
11	user7	user7pass	Nicolas	Cage
12	user6	user6pass	Brad	Pitt
13	user8	user8pass	Kifer	Saderlant
17	user5	user5pass	Adam	Smith
18	user1	user1pass	Mary	Konsta

Ο πίνακας έχει 8 εγγραφές με κάθε εγγραφή να αποτελείται από 5 πεδία τα **id** (το πρωτεύον κλειδί του πίνακα), **username**, **password**, **firstname** και **lastname**.

Στην άσκηση αυτή θα κατασκευάσουμε μια **σελίδα διαχείρισης** δηλ. μια σελίδα μέσω της οποίας ο διαχειριστής της βάσης θα μπορεί κατ' αρχήν να βλέπει τα στοιχεία των χρηστών. Αρχικά τα στοιχεία θα πρέπει να αντλούνται μέσα από τη βάση και να παρατάσσονται σε έναν πίνακα μέσα στη σελίδα με τη μορφή που φαίνεται πιο κάτω:

Manage Users

Username	Password	Firstname	Lastname
user1	user1pass	Mary	Konsta
user2	user2pass	John	Papas
user3	user3pass	Boy	George
user4	user4pass	Celine	Dion
user5	user5pass	Adam	Smith
user6	user6pass	Brad	Pitt
user7	user7pass	Nicolas	Cage
user8	user8pass	Kifer	Saderlant

Στη συνέχεια θα δώσουμε στο διαχειριστή τη δυνατότητα μέσω της σελίδας να διαγράφει χρήστες μέσα από τη βάση, να προσθέτει νέους και να επεξεργάζεται τα στοιχεία των χρηστών με κάθε αλλαγή που θα κάνει να αντικατοπτρίζεται μέσα στη σελίδα.

ΒΑΣΙΚΗ ΔΙΑΤΑΞΗ ΤΗΣ ΣΕΛΙΔΑΣ

Ξεκινώ δημιουργώντας τη βασική διάταξη της τελικής σελίδας. Ανοίγω το σημειωματάριο και εισάγω το βασικό σκελετό:

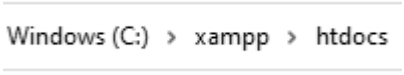
```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>

</body>
</html>
```

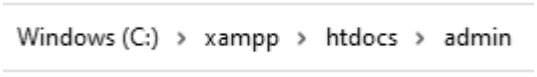
Στο εσωτερικό της διπλής ετικέτας **<body></body>** εισάγω την φράση **Manage Users** την οποία μετατρέπω σε **επικεφαλίδα επιπέδου 1**.

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>
 <h1>Manage Users</h1>
</body>
</html>
```

Μέσα στο σκληρό δίσκο **C:** εντοπίζω το φάκελο **xampp** και μέσα σ' αυτόν τον φάκελο **htdocs**



Μέσα στον φάκελο **htdocs** δημιουργώ έναν νέο φάκελο με το όνομα **admin**

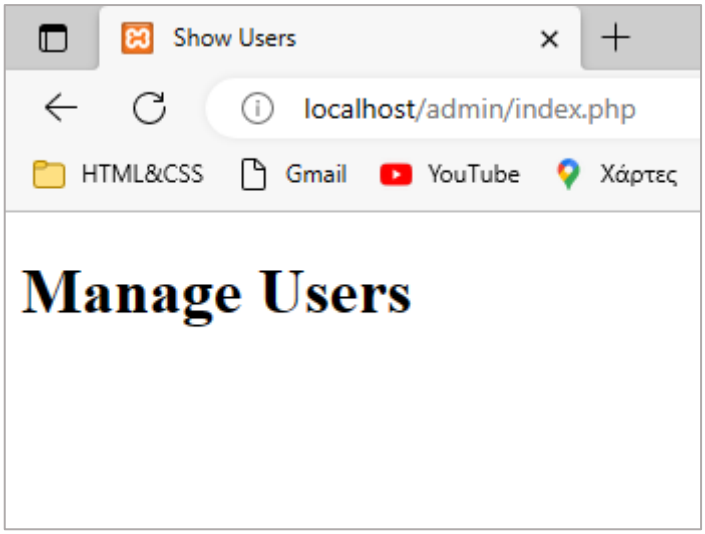


Επιστρέφω στο σημειωματάριο και σώζω τη σελίδα που δημιούργησα μέσα στο φάκελο **admin** με το όνομα **index.php**.

Στη συνέχεια ανοίγω τον browser και ζητώ από το server να μου στείλει τη σελίδα που έφτιαξα ώστε να τη δω, πληκτρολογώντας στην γραμμή διεύθυνσης του browser την ακόλουθη διεύθυνση:

http://localhost/admin/index.php

Αν όλα έχουν πάει καλά θα πρέπει να δω την πιο κάτω σελίδα:



Επιστρέφω στο σημειωματάριο και φτιάχνω το σκελετό του πίνακα. Βλέπω ότι ο πίνακας θέλω να αποτελείται από 4 στήλες και τόσες γραμμές όσες οι εγγραφές του πίνακα users της βάσης. Επειδή δεν μπορεί να γνωρίζω εκ των προτέρων πόσες είναι οι εγγραφές ξεκινώ δημιουργώντας έναν πίνακα με 4 στήλες και 1 γραμμή με υποδειγματικό περιεχόμενο.

Στο σημειωματάριο και κάτω ακριβώς από την επικεφαλίδα προσθέτω τη διπλή ετικέτα **<table></table>** με την οποία πληροφορώ τον browser ότι σ' αυτό το σημείο θέλω να εμφανίζεται ένας πίνακας.

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>
 <h1>Manage Users</h1>
 <table>

 </table>
</body>
</html>
```

Συνεχίζω εισάγοντας την πρώτη γραμμή του πίνακα. Αυτή θέλω να αποτελείται από 5 κελιά με τα ονόματα των στηλών.

Username	Password	Firstname	Lastname

Μέσα στον κώδικα και στο εσωτερικό της διπλής ετικέτας **<table></table>** προσθέτω αρχικά τη διπλή ετικέτα **<tr></tr>** με την οποία δηλώνω ότι επιθυμώ τη δημιουργία μίας νέας γραμμής στον πίνακα.

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>

  </tr>
 </table>
</body>
</html>
```

Και μέσα στην διπλή ετικέτα **<tr></tr>** προσθέτω τόσα ζευγαράκια **<th></th>** όσα τα κελιά της γραμμής (εδώ 4).

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th></th>
   <th></th>
   <th></th>
   <th></th>
  </tr>
 </table>
</body>
</html>
```

Συνεχίζω τοποθετώντας στο εσωτερικό κάθε μιας από τις ετικέτες **<th></th>** τους τίτλους των αντίστοιχων κελιών.

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
  </tr>
 </table>
</body>
</html>
```

Σώζω και βλέπω τις αλλαγές. Η σελίδα πρέπει να έχει την πιο κάτω μορφή.

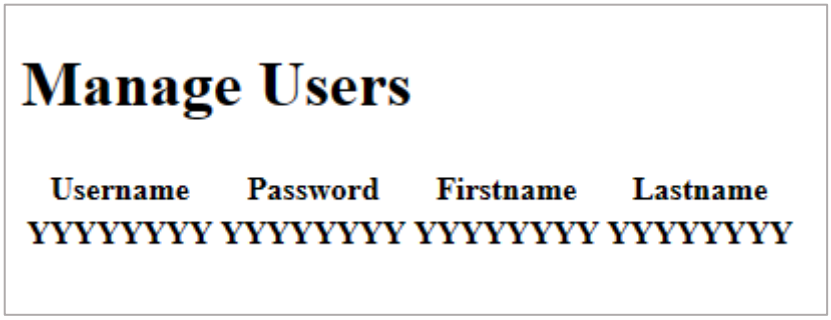


Επιστρέφω στον κώδικα και εισάγω ακριβώς κάτω από την πρώτη γραμμή μια δεύτερη γραμμή με **υποδειγματικό περιεχόμενο**. Κι αυτό γιατί προς το παρόν δεν γνωρίζω ούτε πόσες είναι οι εγγραφές του table users ούτε το ακριβές περιεχόμενό τους.

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
</head>
<body>
 <h1>Manage Users</h1>
```

```
<table>
 <tr>
  <th>Username</th>
  <th>Password</th>
  <th>Firstname</th>
  <th>Lastname</th>
 </tr>
 <tr>
  <th>YYYYYYYY</th>
  <th>YYYYYYYY</th>
  <th>YYYYYYYY</th>
  <th>YYYYYYYY</th>
 </tr>
 </table>
</body>
</html>
```

Σώζω και βλέπω τις αλλαγές.

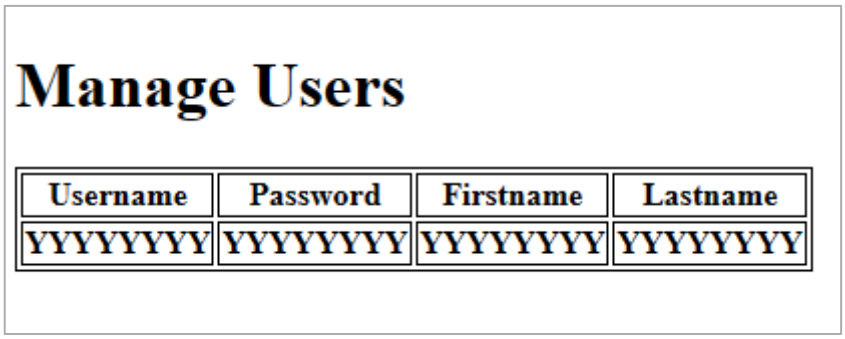


Προσέξτε ότι η διπλή ετικέτα **<th></th>** παίρνει το όνομά της από τις αγγλικές λέξεις **Table Heading** και σε αντίθεση με την **<td></td>** (που σημαίνει **Table Data**) δίνει στα περιεχόμενα του κελιού μορφή επικεφαλίδας, με έντονη γραφή και στοίχιση στο κέντρο.

Κάτω από την ετικέτα **<title>Show Users</title>** εισάγω τη διπλή ετικέτα **<style></style>** με τον ακόλουθο κανόνα μορφοποίησης ο οποίος δίνει περίγραμμα γύρω από τον πίνακα και γύρω από κάθε κελί.

```
<!DOCTYPE html>
<html>
<head>
 <title>Show Users</title>
 <style>
    table, th, td {
          border: 1px solid black;
    }
 </style>
</head>
```

Σώζω και βλέπω τις αλλαγές.



ΔΙΑΣΥΝΔΕΣΗ ΜΕ ΤΗ ΒΑΣΗ

Με το που καλείται η σελίδα θα πρέπει να λειτουργεί ένας κώδικας ο οποίος αρχικά θα εγκαθιστά μια σύνδεση με τη βάση και στη συνέχεια θα ανασύρει μέσα από τον πίνακα **users** της βάσης όλα τα στοιχεία των χρηστών και θα τα στήνει στον πίνακα της σελίδας ώστε να μπορούμε να τα δούμε.

Ξεκινούμε με τον κώδικα που εγκαθιστά τη διασύνδεση.

Επειδή η διασύνδεση με τη βάση είναι το πρώτο πράγμα που πρέπει να λαμβάνει χώρα πριν από την εμφάνιση των περιεχομένων της σελίδας ο σχετικός php κώδικας που την πραγματοποιεί θα πρέπει να τοποθετηθεί στην κορυφή της σελίδας.

Ακριβώς πάνω από τον HTML κώδικα τοποθετώ την διπλή ετικέτα **<?php ?>** όπως φαίνεται πιο κάτω:

```
<?php

?>

<!DOCTYPE html>
<html>
<head>
```

Στη συνέχεια στο εσωτερικό της διπλής ετικέτας τοποθετώ τον ακόλουθο κώδικα:

```
<?php
  $conn=new mysqli("localhost","root","","usersdb");
?>

<!DOCTYPE html>
<html>
<head>
```

Ο κώδικας εγκαθιστά μία διασύνδεση με τη βάση **usersdb** που βρίσκεται στον τοπικό server (**localhost**). Για τη σύνδεση με τη βάση θα χρησιμοποιηθεί ως **username** η λέξη **root,** δίχως password (για αυτό το τρίτο όρισμα στην πιο πάνω κλήση είναι το κενό string).

Σώζω και ανοίγω τη σελίδα.

Εφόσον δεν δω κάποιο μήνυμα λάθους η διασύνδεση με τη βάση έχει πραγματοποιηθεί.

ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΗ ΒΑΣΗ – ΥΠΟΒΟΛΗ SQL ΕΡΩΤΗΜΑΤΩΝ

Από εκεί και έπειτα μπορώ μέσω php κώδικα να στέλνω **αιτήματα** στον **MySQL server** με τα οποία να του ζητώ να ανασύρει πληροφορίες μέσα από τη βάση ή να μεταβάλει τα περιεχόμενό της. Τα αιτήματα θα υποβάλλονται με την μορφή **SQL queries** που είναι η γλώσσα που αντιλαμβάνεται ο MySQL server.

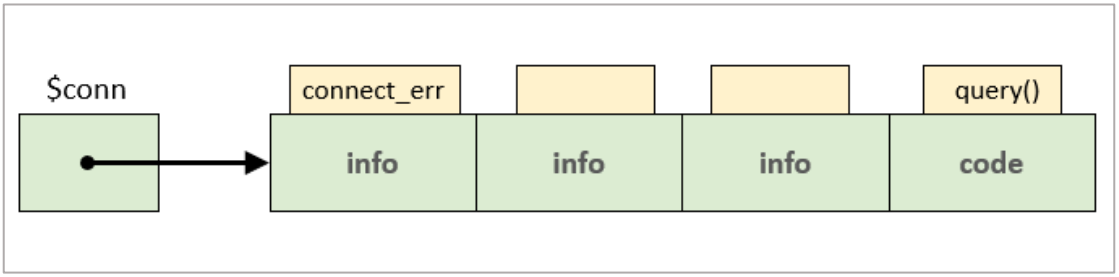
Εκτός από τη διασύνδεση η πιο πάνω εντολή δημιουργεί μέσα στη μνήμη του server και ένα νέο αντικείμενο στο οποίο δίνουμε το όνομα **$conn** (ή όποιο άλλο όνομα θέλουμε).

Το αντικείμενο αυτό είναι μια δομή όπως αυτή που φαίνεται πιο κάτω. Μοιάζει με μια **λίστα** τα στοιχεία της οποίας δεν αριθμούνται αντίθετα έχουν **ετικέτες** (labels).

Ορισμένα από αυτά τα στοιχεία περιέχουν **πληροφορία** που μπορούμε άμεσα να την προσπελάσουμε όπως εδώ το στοιχείο με ετικέτα '**connect_err'** το οποίο δημιουργείται στην περίπτωση που κάτι πήγε στραβά και περιέχει τον τύπο του σφάλματος. Τα στοιχεία αυτά λέγονται **ιδιότητες**.

Για να προσπελάσουμε τις ιδιότητες χρησιμοποιούμε μια έκφραση όπως η πιο κάτω η οποία λέει στο server να εντοπίσει μέσα στη μνήμη του το αντικείμενο με το όνομα **$conn** και μέσα σ' αυτό το στοιχείο με την ετικέτα **connect_err**. Το στοιχείο εντοπίζεται και επιστρέφεται το περιεχόμενό του που στο πιο κάτω παράδειγμα είναι ο τύπος του σφάλματος.

```
$conn→connect_err
```



Αυτό όμως που κάνει τα αντικείμενα να διαφέρουν από τις λίστες που έχουμε δει μέχρι τώρα, είναι η δυνατότητα να περιέχουν στοιχεία που στο εσωτερικό τους βρίσκεται κώδικας τον οποίο μπορούμε να καλέσουμε ώστε να εκτελεστεί με απλή αναφορά στο όνομα του στοιχείου. Τα στοιχεία αυτά λέγονται **μέθοδοι** και λειτουργούν όπως οι **συναρτήσεις**. Τις καλούμε με το όνομά τους ακολουθούμενο από δύο παρενθέσεις.

Πολλές φορές θα χρειαστεί να φωνάξουμε τη μέθοδο και να της δώσουμε και κάποιες επιπλέον πληροφορίες προκειμένου να δουλέψει. Αυτές τις πληροφορίες θα πρέπει να τις δώσουμε μέσα στην παρένθεση όπως ακριβώς κάνουμε και με τις συναρτήσεις. Μια τέτοια μέθοδος είναι η **query()**. Την μέθοδο αυτή θα πρέπει να καλούμε κάθε φορά που θέλουμε να εκτελέσουμε ένα SQL ερώτημα στη βάση.

Η κλήση στη μέθοδο πρέπει να γίνεται ως εξής:

```
$conn→query()
```

Μέσα στις παρενθέσεις θα πρέπει να δίνω σε μορφή **string** το SQL ερώτημα που θέλω να εκτελεστεί. Εδώ θέλω να μου επιστραφούν όλα τα περιεχόμενα του πίνακα **users** οπότε θα πρέπει να γράψω τα πιο κάτω:

```
$conn->query("SELECT * FROM users");
```

Ο καλύτερος τρόπος είναι να δημιουργώ μια νέα μεταβλητή στη μνήμη του server με το όνομα **$sql** στην οποία να αποθηκεύω το ερώτημα με τη μορφή **string**. Στη συνέχεια να καλώ τη **μέθοδο query** του αντικειμένου **$conn** και να της περνάω το string **$sql** όπως φαίνεται πιο κάτω:

```
$sql="SELECT * FROM users";
$conn->query($sql);
```

Η μέθοδος **query()** στέλνει το SQL ερώτημα στον MySQL Server ο οποίος αναλαμβάνει να το εκτελέσει.

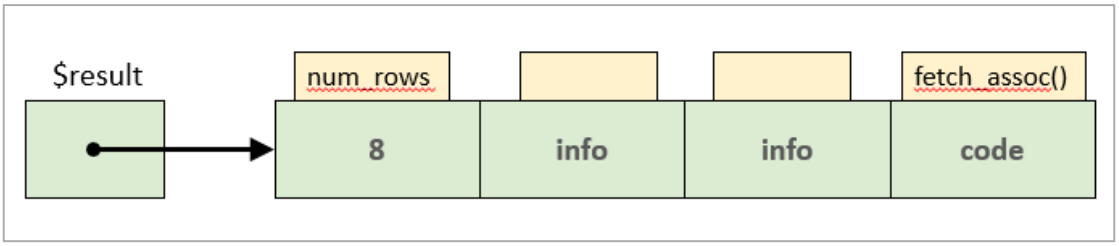
Στην περίπτωση που πρόκειται για ένα **ερώτημα αναζήτησης** μέσα στη βάση (**SELECT query**) ο MySQL server επιστρέφει πίσω έναν **πίνακα (table)**. Σ' αυτή την περίπτωση η μέθοδος **query()** δημιουργεί ένα νέο αντικείμενο στη μνήμη του server με πληροφορίες που αφορούν τον πίνακα που επιστρέφεται αλλά και μια σημαντική μέθοδο για τη διαχείρισή του.

Στο αντικείμενο αυτό δίνουμε συνήθως το όνομα **$result** (ή οποιοδήποτε άλλο όνομα) ως εξής:

```
$sql="SELECT * FROM users";
$result=$conn->query($sql);
```

ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ ΠΙΝΑΚΑ ΠΟΥ ΕΠΙΣΤΡΕΦΕΤΑΙ ΑΠΟ ΕΝΑ SELECT QUERY

Πιο κάτω φαίνεται η μορφή του αντικειμένου **$result** στη μνήμη του server.



αλλά και ο πίνακας που επιστρέφεται στη μνήμη μετά την εκτέλεση του ερωτήματος

id	username	password	firstname	lastname
2	user2	user2pass	John	Papas
3	user3	user3pass	Boy	George
4	user4	user4pass	Celine	Dion
11	user7	user7pass	Nicolas	Cage
12	user6	user6pass	Brad	Pitt
13	user8	user8pass	Kifer	Saderlant
17	user5	user5pass	Adam	Smith
18	user1	user1pass	Mary	Konsta

Η εκτέλεση του ερωτήματος **«SELECT * FROM users»** επιστρέφει όλα τα περιεχόμενα του πίνακα **users**. Δηλαδή έναν πίνακα αποτελούμενο από **8** εγγραφές. Μια από τις ιδιότητες του αντικειμένου **$result** έχει το όνομα **num_rows** και τιμή το πλήθος των εγγραφών του πίνακα που επιστρέφεται στη μνήμη μετά την εκτέλεση του ερωτήματος.

Για να προσπελάσω το **πλήθος των εγγραφών** θα χρησιμοποιήσω την επόμενη έκφραση:

```
$result→num_rows
```

Η διαχείριση των στοιχείων του πίνακα γίνεται πάντοτε μέσω της μεθόδου **fetch_assoc()** ως εξής:

Ο πίνακας που επιστρέφεται στη μνήμη μετά την εκτέλεση του **SELECT query** συνοδεύεται από ένα δείκτη που δείχνει στην πρώτη εγγραφή.



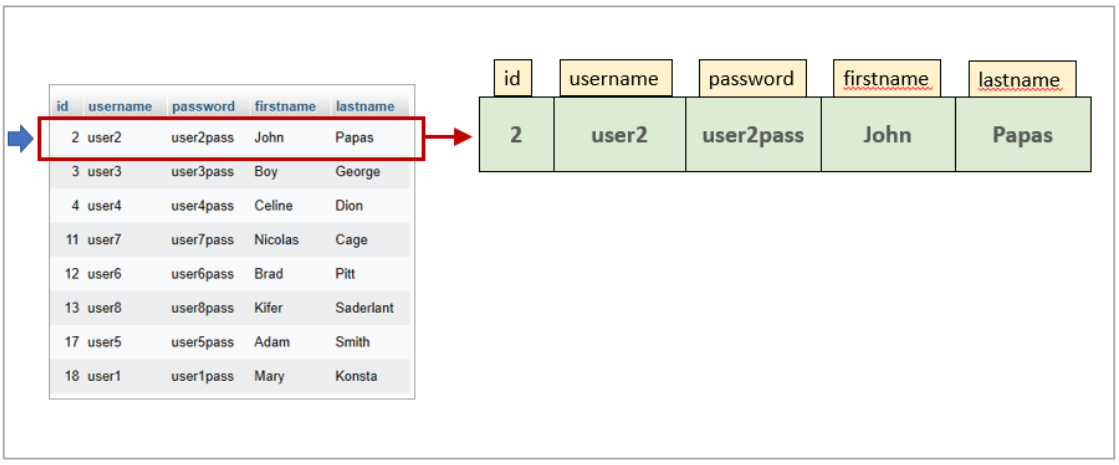
id	username	password	firstname	lastname
2	user2	user2pass	John	Papas
3	user3	user3pass	Boy	George
4	user4	user4pass	Celine	Dion
11	user7	user7pass	Nicolas	Cage
12	user6	user6pass	Brad	Pitt
13	user8	user8pass	Kifer	Saderlant
17	user5	user5pass	Adam	Smith
18	user1	user1pass	Mary	Konsta

Η κλήση στην μέθοδο **fetch_assoc()** του αντικειμένου **$result** που φαίνεται πιο κάτω

```
$result→fetch_assoc()
```

Έχει ως αποτέλεσμα τα στοιχεία της εγγραφής που βρίσκεται στη θέση του δείκτη να μεταφερθούν μέσα στη μνήμη σε μια λίστα (**associative array**) αποτελούμενη από τόσα στοιχεία όσα τα πεδία της εγγραφής με κάθε στοιχείο να φέρει ως ετικέτα το όνομα του αντίστοιχου πεδίου της εγγραφής.

Πιο κάτω φαίνεται το αποτέλεσμα της εκτέλεσης της μεθόδου **fetch_assoc()** για πρώτη φορά.

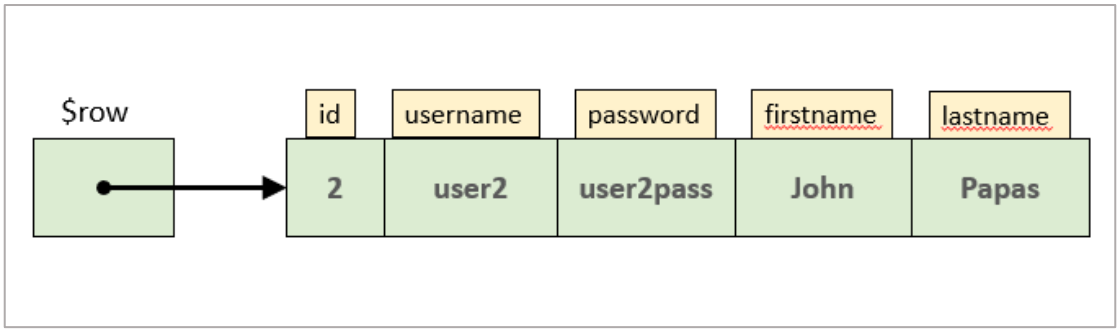


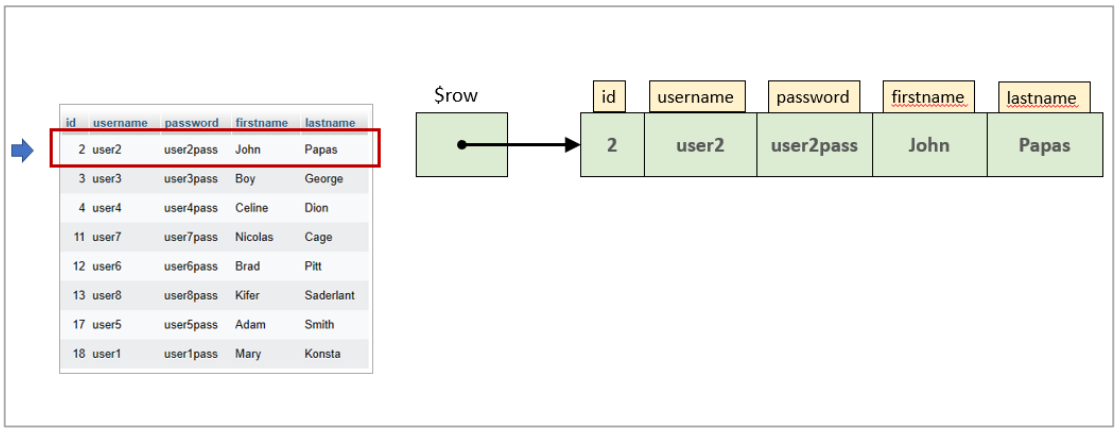
Προσέξτε ότι μέσα στη μνήμη του server δημιουργείται μια λίστα με 5 στοιχεία όσα τα πεδία της εγγραφής. Κάθε ένα από τα στοιχεία της λίστας γεμίζει με την τιμή του αντίστοιχου πεδίου της εγγραφής, ενώ για ετικέτα τοποθετείται το όνομα που έφερε το αντίστοιχο πεδίο στην εγγραφή. Δημιουργείται δηλαδή μέσα στη μνήμη του server μια λίστα πανομοιότυπη με την εγγραφή του πίνακα. Στην λίστα αυτή δίνουμε ένα **όνομα** (συνήθως το **$row**) προκειμένου να μπορούμε να τη διαχειριστούμε, δηλ. να αναφερθούμε στα στοιχεία της για να επεξεργαστούμε τις πληροφορίες που αυτά περιέχουν.

Η απόδοση του ονόματος γίνεται ως εξής:

```
$row=$result→fetch_assoc()
```

Η δομή **$row** που δημιουργείται μέσα στη μνήμη φαίνεται πιο κάτω:





Μετά την εκτέλεση της μεθόδου ο δείκτης μεταφέρεται στην εγγραφή που ακολουθεί:

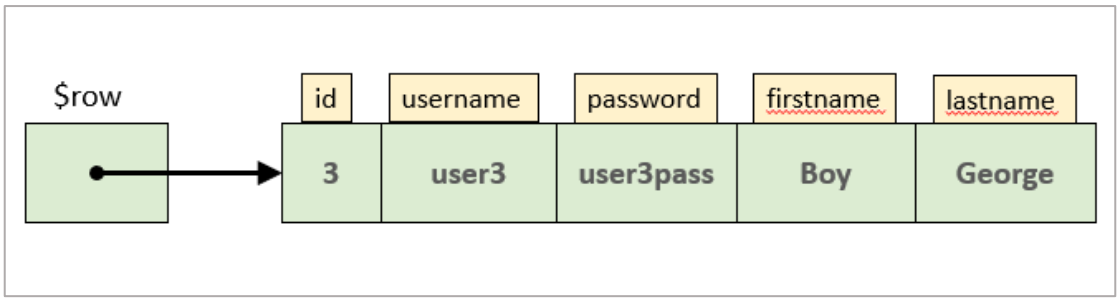


id	username	password	firstname	lastname
2	user2	user2pass	John	Papas
3	user3	user3pass	Boy	George
4	user4	user4pass	Celine	Dion
11	user7	user7pass	Nicolas	Cage
12	user6	user6pass	Brad	Pitt
13	user8	user8pass	Kifer	Saderlant
17	user5	user5pass	Adam	Smith
18	user1	user1pass	Mary	Konsta

Νέα κλήση στη μέθοδο fetch_assoc() με τη μορφή που φαίνεται πιο κάτω

$row=$result→fetch_assoc()

θα έχει ως αποτέλεσμα τη δημιουργία νέας λίστας στη μνήμη του server με τα στοιχεία της δεύτερης εγγραφής με τον δείκτη προς τη νέα λίστα να αποθηκεύεται μέσα στη μεταβλητή **$row**.



Καλώντας λοιπόν διαδοχικά την μέθοδο **fetch_assoc()** του αντικειμένου **$result** καταφέρνω να μεταφέρω κάθε φορά μία ολόκληρη εγγραφή του table που έχει επιστραφεί μέσα σε μια λίστα στη μνήμη του server. Από αυτή τη λίστα μπορώ να τραβώ τα περιεχόμενα της εγγραφής και να τα διαχειρίζομαι.

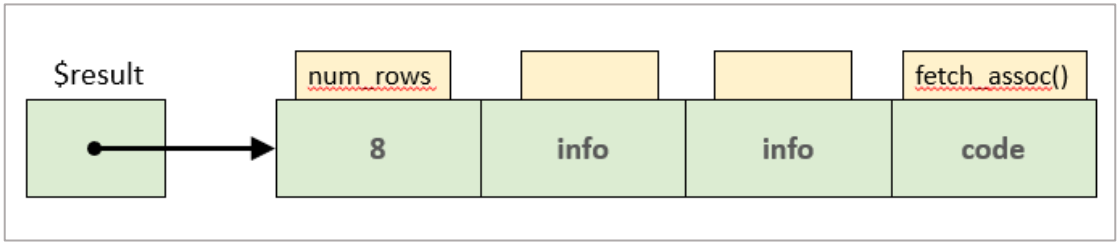
ΠΡΟΣΟΧΗ! Δεν θα δημιουργώ τόσες λίστες μέσα στη μνήμη όσες οι εγγραφές. Αντίθετα κάθε φορά θα μεταφέρω μία εγγραφή σε μια λίστα και αφού αξιοποιήσω τα περιεχόμενά της μετά θα περάσω να διαβάσω τα περιεχόμενα της επόμενης εγγραφής στην ίδια λίστα κ.ο.κ.

ΥΠΟΒΟΛΗ ΕΡΩΤΗΜΑΤΟΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ TABLE ΠΟΥ ΕΠΙΣΤΡΕΦΕΙ Ο MYSQL SERVER

Επιστρέφω στο σημειωματάριο και ακριβώς κάτω από την πρώτη γραμμή του πίνακα στο σημείο που θέλω να έρχονται τα στοιχεία μέσα από τη βάση και να τοποθετούνται σε γραμμές εισάγω τη διπλή ετικέτα **<?php ?>** ακολουθούμενη από το ερώτημα που περιεγράφηκε πιο πάνω:

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
  </tr>
  <?php
      $sql="SELECT * FROM users";
      $result=$conn->query($sql);
  ?>
  <tr>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
  </tr>
 </table>
</body>
</html>
```

Η εκτέλεση του PHP κώδικα θα έχει ως αποτέλεσμα την εκτέλεση του query, την επιστροφή του πίνακα users και την δημιουργία του αντικειμένου $result που περιγράψαμε πριν.



Το πρώτο βήμα θα είναι να δούμε πόσες εγγραφές περιέχει ο πίνακας. Οι εγγραφές έχουν καταμετρηθεί και το πλήθος τους έχει φυλαχτεί ως τιμή της ιδιότητας **num_rows** του αντικειμένου **$result**.

Η πρώτη μας δουλειά θα είναι να μεταφέρουμε το πλήθος των εγγραφών σε μια νέα μεταβλητή έστω **$N** μέσω της εντολής:

```
$N=$result→num_rows;
```

την οποία συμπληρώνω στον πιο πάνω κώδικα ακριβώς στη συνέχεια:

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
  </tr>
  <?php
      $sql="SELECT * FROM users";
      $result=$conn->query($sql);

      $N=$result→num_rows;
  ?>
  <tr>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
  </tr>
 </table>
</body>
</html>
```

Αφού γνωρίζω τώρα πόσες εγγραφές έχουν επιστραφεί αρκεί να καλέσω τόσες φορές τη μέθοδο **fetch_assoc()** όσο το πλήθος των εγγραφών προκειμένου να μπορέσω να μεταφέρω το περιεχόμενό τους στην μνήμη για να το επεξεργαστώ. Πρόκειται για μια διαδικασία που πρέπει να **επαναλάβω** τόσες φορές όσες είναι οι εγγραφές που έχουν επιστραφεί. Άρα θα χρειαστώ μια εντολή **for**.

Στο προηγούμενο κώδικα συμπληρώνω την πιο κάτω εντολή:

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
 </tr>
 <?php
      $sql="SELECT * FROM users";
      $result=$conn->query($sql);

      $N=$result→num_rows;
      for ($i=1; $i<=$N; $i=$i+1) {

      }
  ?>
 <tr>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
 </tr>
 </table>
</body>
</html>
```

Τώρα μέσα στη for θα πρέπει να κωδικοποιήσω σε php τις πιο κάτω ενέργειες:

A. Μεταφορά της τρέχουσας εγγραφής σε μια λίστα μέσα στη μνήμη του server

Ακολουθεί ο σχετικός κώδικας

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
  </tr>
  <?php
      $sql="SELECT * FROM users";
      $result=$conn->query($sql);

      $N=$result→num_rows;
      for ($i=1; $i<=$N; $i=$i+1) {
         $row=$result->fetch_assoc();
      }
   ?>
  <tr>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
  </tr>
 </table>
</body>
</html>
```

Β. Μεταφορά των στοιχείων της λίστας σε νέες μεταβλητές που είναι πιο εύκολα προσβάσιμες.

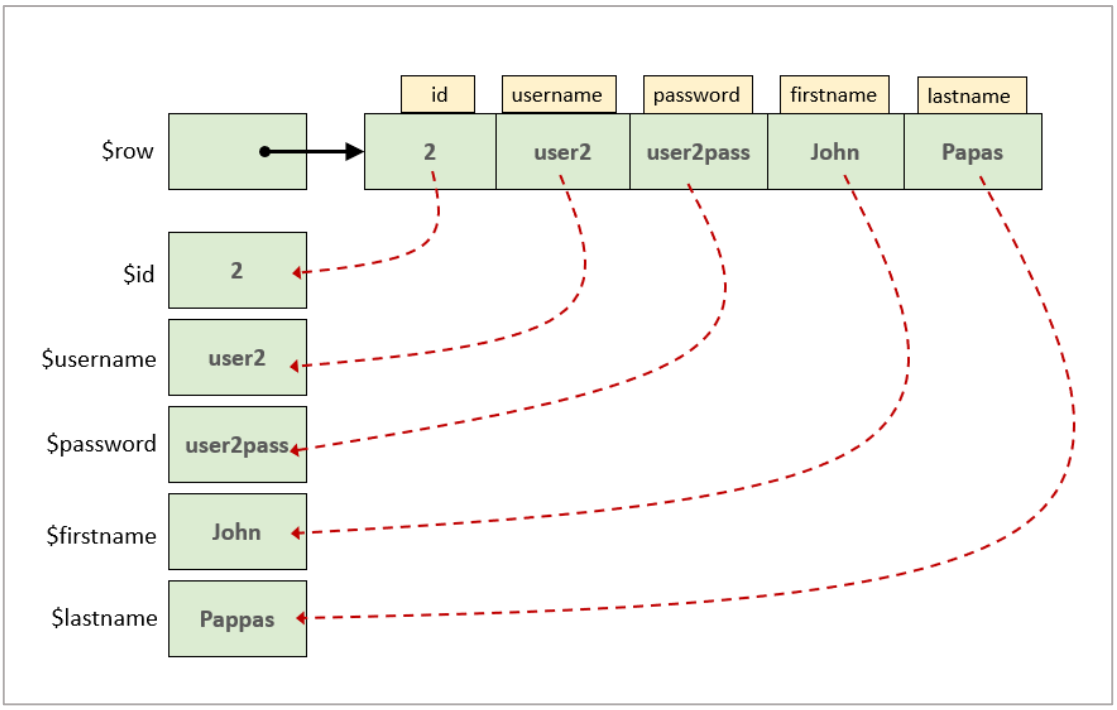
Ακολουθεί ο σχετικός κώδικας

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
  </tr>
  <?php
      $sql="SELECT * FROM users";
      $result=$conn->query($sql);

      $N=$result→num_rows;
      for ($i=1; $i<=$N; $i=$i+1) {
          $row=$result->fetch_assoc();

          $id=$row["id"];
          $username=$row["username"];
          $password=$row["password"];
          $firstname=$row["firstname"];
          $lastname=$row["lastname"];
      }
   ?>
  <tr>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
   <th>YYYYYYYY</th>
  </tr>
 </table>
</body>
</html>
```

Στο επόμενο σχήμα φαίνεται το αποτέλεσμα της εκτέλεσης των πιο πάνω εντολών μέσα στη μνήμη.



Γ. Δημιουργία του κατάλληλου HTML κώδικα ο οποίος προσθέτει μια νέα γραμμή στον υπάρχοντα πίνακα με τα στοιχεία του χρήστη που μόλις ανασύρθηκαν και βρίσκονται στις μεταβλητές που περιγράφηκαν πιο πάνω.

Στον προηγούμενο κώδικα ξεκινώ μεταφέροντας τον html κώδικα που προσθέτει νέα γραμμή μέσα στην ετικέτα της php και τον κλείνω μέσα σε μια εντολή **echo** .

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
 </tr>
 <?php
     $sql="SELECT * FROM users";
     $result=$conn->query($sql);

     $N=$result→num_rows;
     for ($i=1; $i<=$N; $i=$i+1) {
         $row=$result->fetch_assoc();
        $id=$row["id"];
```

```
        $username=$row["username"];
        $password=$row["password"];
        $firstname=$row["firstname"];
        $lastname=$row["lastname"];

        echo "
          <tr>
            <td> YYYYYYYY </td>
            <td> YYYYYYYY </td>
            <td> YYYYYYYY </td>
            <td> YYYYYYYY </td>
          </tr>
        ";
    }
  ?>
 </table>
</body>
</html>
```

Η εντολή **echo** όταν εκτελεστεί θα συμπληρώσει τον κώδικα της γραμμής μέσα στην υπάρχουσα html. Στη συνέχεια αντικαθιστώ το υποδειγματικό περιεχόμενο κάθε κελιού με τα στοιχεία του χρήστη που προηγουμένως ανέσυρα μέσα από την εγγραφή και τα έχω μεταφέρει σε μεταβλητές μέσα στη μνήμη.

```
<body>
 <h1>Manage Users</h1>
 <table>
  <tr>
   <th>Username</th>
   <th>Password</th>
   <th>Firstname</th>
   <th>Lastname</th>
  </tr>
  <?php
      $sql="SELECT * FROM users";
      $result=$conn->query($sql);

      $N=$result→num_rows;
      for ($i=1; $i<=$N; $i=$i+1) {
          $row=$result->fetch_assoc();
          $id=$row["id"];
```

```
        $username=$row["username"];
        $password=$row["password"];
        $firstname=$row["firstname"];
        $lastname=$row["lastname"];

        echo "
          <tr>
            <td> $username</td>
            <td> $password</td>
            <td> $firstname</td>
            <td> $lastname </td>
          </tr>
        ";
    }

  ?>
 </table>
</body>
</html>
```

Σώζω και ανοίγω τη σελίδα στον browser

Manage Users

Username	Password	Firstname	Lastname
user2	user2pass	John	Papas
user3	user3pass	Boy	George
user4	user4pass	Celine	Dion
user7	user7pass	Nicolas	Cage
user6	user6pass	Brad	Pitt
user8	user8pass	Kifer	Saderlant
user5	user5pass	Adam	Smith
user1	user1pass	Mary	Konsta

Αν θέλω τα περιεχόμενα του πίνακα να εμφανίζονται ταξινομημένα ως προς το username θα πρέπει να κάνω στο SELECT query την πιο κάτω προσθήκη:

```
$sql="SELECT * FROM users ORDER BY username";
```

Σώζω και ανοίγω τη σελίδα στον browser

Manage Users

Username	Password	Firstname	Lastname
user1	user1pass	Mary	Konsta
user2	user2pass	John	Papas
user3	user3pass	Boy	George
user4	user4pass	Celine	Dion
user5	user5pass	Adam	Smith
user6	user6pass	Brad	Pitt
user7	user7pass	Nicolas	Cage
user8	user8pass	Kifer	Saderlant