



Περιεχομενα "at a glance"

1. Βάσεις Δεδομένων
2. Πίνακες, Στήλες, Πεδία
3. Σχήματα Βάσεων Δεδομένων
4. Εισαγωγή στη γλώσσα SQL
5. SELECT, WHERE
6. INSERT INTO, DELETE, UPDATE
7. Συνδυασμός PHP/MySQL

Πρώτο Πρόγραμμα

1. Βάσεις Δεδομένων

Τι είναι Βάση Δεδομένων?

- Η Βάση Δεδομένων είναι μια συλλογή από σχετικά μεταξύ τους δεδομένα - αντικείμενα.
- Σκοπός της είναι να διευκολύνει τον χρήστη στην εισαγωγή, διαγραφή, χειρισμό και επεξεργασία μεγάλου όγκου δεδομένων.
- Τα δεδομένα που χειρίζεται είναι δυναμικά, δηλαδή αλλάζουν συνεχώς.

Χρήστες Βάσεων Δεδομένων

- Είναι όσοι χρησιμοποιούν την Βάση Δεδομένων είτε για να αποκτήσουν πληροφορίες είτε για να την συντηρήσουν.
 - ▣ Διαχειριστής Συστήματος.
 - ▣ Ιδιοκτήτης Βάσης Δεδομένων.
 - ▣ Χρήστες με διάφορες διαβαθμίσεις δικαιωμάτων.



Συστήματα Διαχείρισης Βάσεων Δεδομένων

- Είναι το λογισμικό που επιτρέπει στους χρήστες να δημιουργούν και να χρησιμοποιούν μια Βάση Δεδομένων.
- Δυνατότητες που παρέχει στους χρήστες:
 1. Ορισμός Βάσης Δεδομένων.
 2. Κατασκευή Βάσης Δεδομένων.
 3. Χρήση / Προσπέλαση Βάσης Δεδομένων.
 4. Διαγραφή Βάσης Δεδομένων.

Συστήματα Διαχείρισης Βάσεων Δεδομένων (2)

- Θα μπορούσαμε να υλοποιήσουμε μια Βάση Δεδομένων και χωρίς την χρήση Συστήματος Διαχείρισης (πχ. με χρήση αρχείων).
- Τα σημαντικότερα πλεονεκτήματά του είναι:
 1. Ευκολία στην σχεδίαση και την υλοποίηση.
 2. Γρήγορη ανάπτυξη εφαρμογών.
 3. Ακεραιότητα δεδομένων - Ασφάλεια.
 4. Επίπεδα χρηστών – Έλεγχος πρόσβασης.
 5. Ταυτόχρονη χρήση από πολλούς χρήστες.
 6. Ελέγχος ορθότητας – Πλεονασμών.
 7. Έτοιμες συναρτήσεις - Αλγόριθμοί

Δομή Σ. Δ. Βάσεων Δεδομένων

- Το Σύστημα Διαχείρισης παρέχει μια γλώσσα υποβολής ερωτήσεων προς την Βάση Δεδομένων.
- Οι ερωτήσεις αυτές επεξεργάζονται από το **Λογισμικό Χειρισμού Ερωτήσεων** και μετατρέπονται σε εντολές για ανάκληση ή επεξεργασία δεδομένων.
- Το **Λογισμικό Αποθήκευσης και Ανάκλησης Δεδομένων** αναλαμβάνει να μεταφέρει τις εντολές του χρήστη στο επίπεδο του υλικού.



Αρχιτεκτονική Βάσεων Δεδομένων

1. Εξωτερικό Επίπεδο

- ▣ Παρουσίαση δεδομένων στον χρήστη της Βάσης.

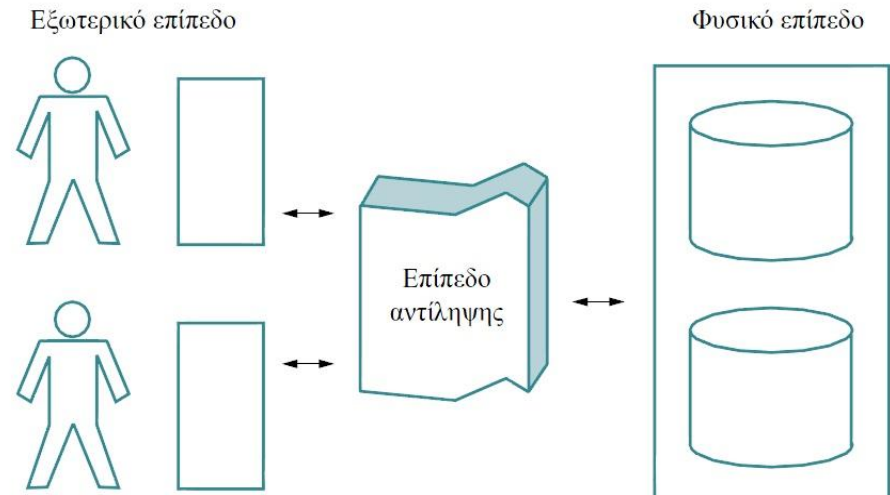
2. Επίπεδο Αντίληψης

- ▣ Συνδέει το φυσικό επίπεδο με το εξωτερικό επίπεδο.
- ▣ Επεξεργάζεται τις εντολές του χρήστη και βρίσκει την κατάλληλη πληροφορία στην Βάση.
- ▣ Μεταφράζει τις εντολές του χρήστη σε εντολές κατανοητές από το **Μηχανισμό Ανάκλησης και Αποθήκευσης Δεδομένων**

Αρχιτεκτονική Βάσεων Δεδομένων

3. Φυσικό Επίπεδο

- Το επίπεδο όπου τα δεδομένα είναι αποθηκευμένα σε μορφή bytes στον σκληρό δίσκο.
- Περιλαμβάνει τις έννοιες
 1. Πίνακας
 2. Εγγραφή
 3. Πεδίο
 4. Τιμή Πεδίου
 5. Τύπος Δεδομένων



Trailers

1. Βάσεις Δεδομένων ✓
2. Πίνακες, Στήλες, Πεδία

Σχεσιακή Βάση Δεδομένων

- Τα δεδομένα της Βάσης αποθηκεύονται σε **πίνακες**.
- Ένας **πίνακας** αντιπροσωπεύει μια **σχέση**.
- Κάθε **σχέση** αποτελείται από **M στοιχεία**.
 - ▣ Κάθε **στοιχείο** ανήκει σε ένα **Τύπο Δεδομένων**.
- **Εγγραφή**: Κάθε M-άδα για την οποία αληθεύει η σχέση.
- **Πεδίο (ή Στήλη)**: Κάθε ένα στοιχείο της M-αδας.
- **Τιμή Πεδίου**: Η τιμή ενός πεδίου για μια συγκεκριμένη εγγραφή.

5 Best Selling Albums Of All Time

Όνομα Πίνακα

Όνομα	Καλλιτέχνης	Έτος	Είδος	Πωλήσεις
Thriller	Michael Jackson	1982	Pop / R & B	112M
Back In Black	AC/DC	1980	Hard Rock	49M
Dark Side Of The Moon	Pink Floyd	1980	Progressive Rock	45M
The Bodyguard OST	Whitney Houston	1992	Soundtrack	44M
Bat Out Of Hell	Meat Loaf	1977	Classic Rock	43M

Πίνακας

5 Best Selling Albums Of All Time

Τίτλος Πεδίου

Όνομα	Καλλιτέχνης	Έτος	Είδος	Πωλήσεις
Thriller	Michael Jackson	1982	Pop / R & B	112M
Back In Black	AC/DC	1980	Hard Rock	49M
Dark Side Of The Moon	Pink Floyd	1980	Progressive Rock	45M
The Bodyguard OST	Whitney Houston	1992	Soundtrack	44M
Bat Out Of Hell	Meat Loaf	1977	Rock	43M

Πεδίο

Τύποι Πεδίου

- Κάθε πεδίο αποθηκεύει συγκεκριμένο τύπο δεδομένων
 - int: Ακέραιος αριθμός.
 - float: Δεκαδικός αριθμός.
 - text: Αλφαριθμητικό.
 - char(n): Αλφαριθμητικό σταθερού μεγέθους n.
 - varchar(n): Αλφαριθμητικό μέγιστου μεγέθους n.
 - date: Ημερομηνία.
 - datetime: Ημερομηνία και ώρα.

Και πολλοί άλλοι...

NULL

- Ειδική τιμή
- Αντιπροσωπεύει την απουσία τιμής

5 Best Selling Albums Of All Time

Όνομα	Καλλιτέχνης	Έτος	Είδος	Πωλήσεις
Thriller	Michael Jackson	1982	Pop / R & B	112M
Back In Black	AC/DC	1980	Hard Rock	49M
Dark Side Of The Moon	Pink Floyd	1980	Progressive Rock	45M
The Bodyguard OST	Whitney Houston	1992	Soundtrack	44M
Bat Out Of Hell	Meat Loaf	1977	Rock	43M

Εγγραφή



5 Best Selling Albums Of All Time

Όνομα	Καλλιτέχνης	Έτος	Είδος	Πωλήσεις
Thriller	Michael Jackson	1982	Pop / R & B	112M
Back In Black	AC/DC	1980	Hard Rock	49M
Dark Side Of The Moon	Pink Floyd	1980	Progressive Rock	45M
The Bodyguard OST	Whitney Houston	1992	Soundtrack	44M
Bat Out Of Hell	Meat Loaf	1977	Rock	43M

Τιμή Πεδίου



Σσσο, αρχίζει...

1. Βάσεις Δεδομένων
2. Πίνακες, Στήλες, Πεδία
3. Σχήματα Βάσεων Δεδομένων

Σχήμα - Ονοματολογία

- Σχήμα ονομάζεται η δομή των πινάκων και των στηλών μιας Βάσης Δεδομένων.
- Για την δημιουργία σχήματος σκεφτόμαστε:
 - ▣ Τι είδους αντικείμενα θα αποθηκεύσουμε? => Πίνακες
 - ▣ Τι ιδιότητες έχει κάθε τέτοιο αντικείμενο? => Πεδία
- Ονοματολογία Βάσεων – Πινάκων – Πεδίων
 - ▣ Αρχίζουν από γράμμα.
 - ▣ Περιέχουν λατινικά γράμματα, αριθμούς, underscore (_).
 - ▣ Είναι case sensitive.

Παράδειγμα Σχήματος

- Μουσική Εγκυκλοπαίδεια
- Τι είδους αντικείμενα χρειαζόμαστε?
 - ▣ Τραγούδια
 - ▣ **Καλλιτέχνες**
 - ▣ Μπάντες
 - ▣ Αλμπουμ
- Τι ιδιότητες χρειαζόμαστε για κάθε αντικείμενο?

Παράδειγμα Σχήματος (2)

- **Καλλιτέχνης (Artist)**
 - Όνομα (name)
 - Επίθετο (lastname)
 - Ημερομηνία Γέννησης (birthdate)
 - Χώρα (country)
 - Είδος (genre)

Παράδειγμα Σχήματος (3)

name	lastname	birthdate	country	genre
Michael	Jackson	29/08/1959	USA	Pop / R & B
Whitney	Houston	09/08/1963	USA	Soul / R & B
Tarja	Turunen	17/08/1977	Finland	Metal
Σώτης	Βολάνης	10/11/1971	Ελλάδα	Γύφτικο



Μοναδικό Αναγνωριστικό (id)

- Σε κάθε πίνακα έχουμε μια στήλη-αναγνωριστικό.
- Συνήθως την ονομάζουμε
 - ▣ id
 - ▣ Όνομα πίνακα + id => artistid
- Είναι θετικοί φυσικοί αριθμοί
- Ξεκινούν από το 1.
- Κάθε νέα εγγραφή έχει τον επόμενο αριθμό.
 - ▣ Αυτό ρυθμίζεται για να γίνεται αυτόματα από το σύστημα της Βάσης Δεδομένων με `AUTO_INCREMENT`.

Μοναδικό Αναγνωριστικό (id)

- Με αυτόν τον τρόπο μπορούμε να
 - **Αναφερθούμε σύντομα σε μια εγγραφή**
 - Ο καλλιτέχνης με id 10.
 - Το τραγούδι με id 123.
 - Το άλμπουμ με id 666 (“Number of The Beast”).
 - **Ξεχωρίσουμε δύο εγγραφές με ίδια στοιχεία**
 - Καλλιτέχνης **Tom Jones** από **USA**, id 3
 - Καλλιτέχνης **Tom Jones** από **USA**, id 305
- Αυτό το πεδίο ονομάζεται **primary key** (πρωτεύον κλειδί).



Παράδειγμα Σχήματος (4)

Primary Key

id	name	lastname	birthdate	country	genre
1	Michael	Jackson	29/08/1959	USA	Pop / R & B
2	Whitney	Houston	09/08/1963	USA	Soul / R & B
3	Tarja	Turunen	17/08/1977	Finland	Metal
4	Σώτης	Βολάνης	10/11/1971	Ελλάδα	Γύφτικο

Στήλη

AUTO_INCREMENT

Παράδειγμα Σχήματος (5)

□ Album

- albumid
- title
- year
- studioname
- studiolocation
- studiofunded

□ Band

- bandid
- name
- country
- year
- studioname
- studiolocation
- studiofunded

Πίνακας Album

albumid	title	year	studioName	studioLocation	studioFunded
1	Thriller	1982	Westlake	USA	1974
2	Off The Wall	1979	Westlake	USA	1974
3	Bad	1984	Westlake	USA	1974
4	Τσικουλάτα	2004	Vasipap	Greece	2001

Ανεπιθύμητη επανάληψη

Αποφυγή Επανάληψης

- Για να αποφύγουμε την επανάληψη περίσσειας πληροφορίας στους πίνακες μας, φτιάχνουμε ένα πίνακα ξεχωριστά για τα Recording Studios.
- Studio
 - ▣ studioid
 - ▣ name
 - ▣ location
 - ▣ funded

Αναφορές Πινάκων

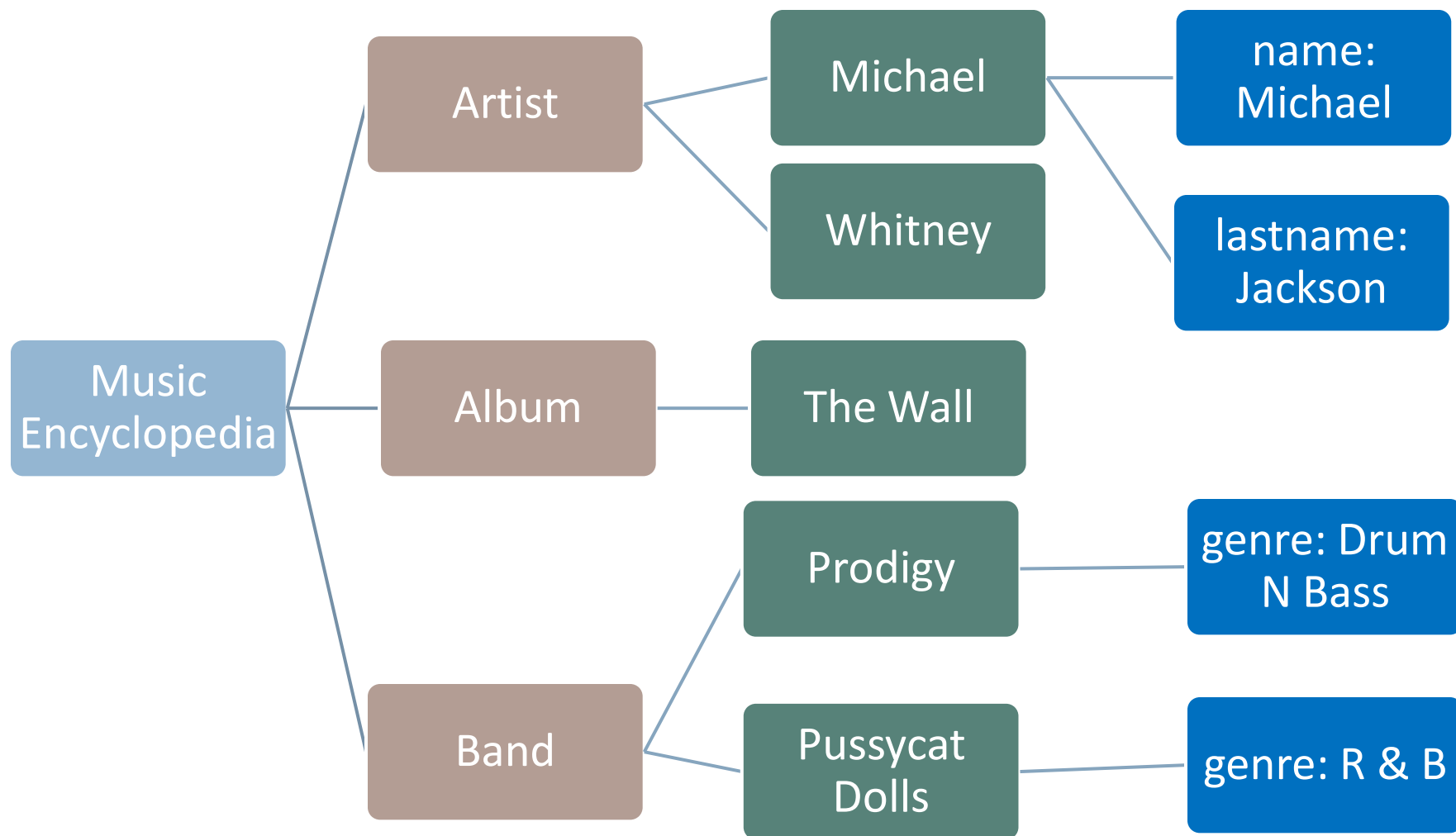
studioid	name	location	funded
1	Westlake	USA	1974
2	Vasipap	Greece	2001

αναφέρεται σε

αναφέρεται σε

albumid	title	year	studioid
1	Thriller	1982	1
2	Off The Wall	1979	1
3	Bad	1984	1
4	Τσικουλάτα	2004	2

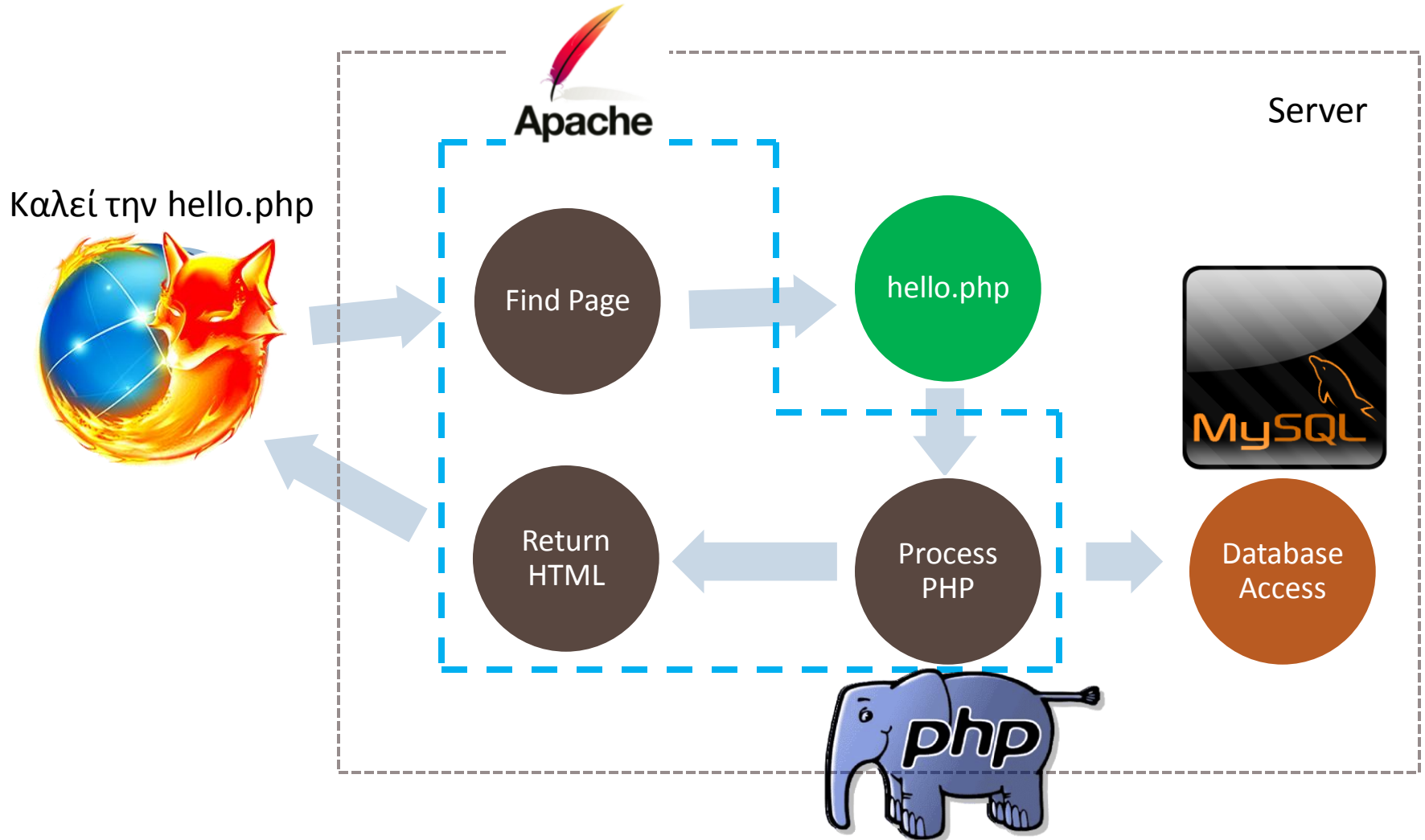
Γραφική Αναπαράσταση Βάσης



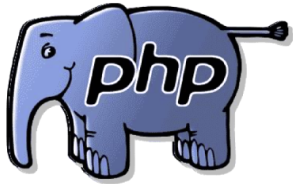
MySQL Server

- Η MySQL είναι ένα πρόγραμμα Server.
- Συνήθως τρέχει στον ίδιο υπολογιστή με τον Apache.
 - ▣ Σε μεγάλα sites τρέχει σε διαφορετική φυσική τοποθεσία για μεγαλύτερη ασφάλεια και σταθερότητα.
- Η PHP συνδέεται στην MySQL
 - ▣ Στέλνει ερωτήματα σε βάσεις.
 - ▣ Δέχεται τις απαντήσεις.
- Η PHP είναι ένας MySQL client
 - ▣ Ενώ ταυτόχρονα συνεχίζει να τρέχει στον HTTP Server (Apache).

Διαδρομή Λειτουργιών



PHP + MySQL = B.F.F.E.



MySQL Client

MySQL Server



→ Φέρε τα στοιχεία του Michael Jackson

← Πάρε τα στοιχεία του Βασιλιά!

→ Προσθεσε στη Βάση την Ke\$ha

← Φίνο γκομενάκι μου φαίνεται! ΟΚ!

→ Διάγραψε από τη Βάση τους βρωμιάρηδες τους Nickelback

← ENNOEITAI! Επιτέλους!

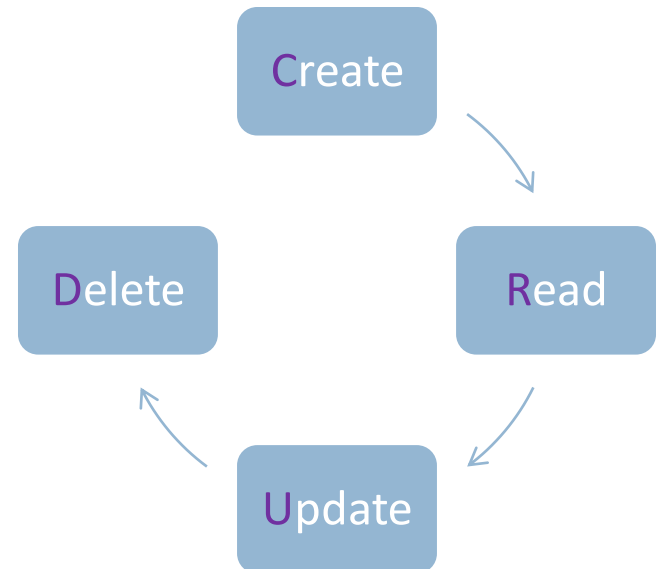


Διάλλειμα

1. Βάσεις Δεδομένων ✓
2. Πίνακες, Στήλες, Πεδία ✓
3. Σχήματα Βάσεων Δεδομένων ✓
4. Εισαγωγή στη γλώσσα SQL

Ερώτημα

- Μια εντολή προς το Σύστημα Διαχείρισης Βάσεων Δεδομένων.
- Ο τρόπος που επικοινωνούμε με τις Βάσεις.
- Μπορούμε να ζητήσουμε
 - ▣ Δημιουργία Εγγραφών (Create)
 - ▣ Ανάγνωση Εγγραφών (Read)
 - ▣ Ενημέρωση Εγγράφων (Update)
 - ▣ Διαγραφή Εγγραφών (Delete)



Ερώτημα (2)

- Σε αντίθεση με το σχήμα της Βάσης που το φτιάχνουμε off-line με την phpMyAdmin:
 - ▣ Τα ερωτήματα τρέχουν στον χρόνο εκτέλεσης, πχ. όταν τρέχει ένα PHP Script.
- Παραδείγματα
 - ▣ Ο χρήστης θέλει να δημιουργήσει έναν λογαριασμό πατώντας το κατάλληλο κουμπί στο site.
 - ▣ Καλείτε το PHP Script register.php με HTTP POST.
 - ▣ Τρέχει το ερώτημα προσθήκης των στοιχείων του νέου μέλους.

Γλώσσα T-SQL

- Χρησιμοποιείται για να γράψουμε τα **ερωτήματα** προς την Βάση.
- Περιγράφει **τι** θέλουμε να συμβεί.
- Οχι **το πως** θέλουμε να συμβεί.
- Κάθε ερώτημα τελειώνει με semicolon (;).
- Space insensitive (τα κενά δεν παίζουν ρόλο).

Ερώτημα	Απάντηση
Δημιουργία	<ul style="list-style-type: none">• Εντάξει ή Όχι εντάξει• Τιμή πρωτεύοντος κλειδιού
Ανάγνωση	<ul style="list-style-type: none">• Δεδομένα
Επεξεργασία	<ul style="list-style-type: none">• Εντάξει ή Όχι εντάξει• Πλήθος εγγραφών
Διαγραφή	<ul style="list-style-type: none">• Εντάξει ή Όχι εντάξει• Πλήθος εγγραφών

Η συνέχεια...

1. Βάσεις Δεδομένων
2. Πίνακες, Στήλες, Πεδία
3. Σχήματα Βάσεων Δεδομένων
4. Εισαγωγή στη γλώσσα SQL
5. **SELECT, WHERE**

Ανάγνωση Εγγραφών

□ Ερώτημα *SELECT*

▣ Σύνταξη

SELECT

πεδίο1, πεδίο2, ... , πεδίοN

FROM

όνομα Πίνακα;

- ▣ Επιστρέφει συγκεκριμένα πεδία από όλες τις εγγραφές ενός πίνακα.

□ Παράδειγμα

SELECT

name, lastname

FROM

Artist;



name	lastname
Michael	Jackson
Whitney	Houston
Tarja	Turunen
Σώτης	Βολάνης

Φιλτράρισμα Εγγραφών

□ Εντολή *WHERE*

□ Σύνταξη

WHERE λογική πρόταση

□ Οι λογικές προτάσεις:

□ Παίρνουν τιμή **true (1)** ή **false (0)**.

□ Μπορούν να περιέχουν

■ **Ονόματα πεδίων:** artistid, name, location.

■ **Σταθερές:** 5, "Hello World!", "20/11/2011".

□ Αληθεύουν πιθανώς για κάποιες εγγραφές.

□ Δεν αληθεύουν πιθανώς για άλλες εγγραφές.

Λογικές Προτάσεις

Τελεστές	Λειτουργία
<code>+, -, *, /</code>	Αριθμητικές πράξεις
<code>OR, AND, NOT</code>	Λογικές πράξεις: ή, και, όχι
<code>=, !=, <, >, <=, >=</code>	Σύγκριση

- Παραδείγματα λογικών προτάσεων
 - ▣ `WHERE name = "Michael"`
 - ▣ `WHERE lastname != "Houston"`
 - ▣ `WHERE artistid < 10`
 - ▣ `WHERE name = "Michael" AND lastname = "Jordan"`
 - ▣ Έλεγχος για `NULL` με τον τελεστή `IS NULL`
 - Π.χ. `WHERE birthdate IS NULL`

Ανάγνωση Εγγραφών (2)

□ Ερώτημα *SELECT WHERE*

▣ Σύνταξη

SELECT

πεδίο1, πεδίο2, ... , πεδίοN

*(ή *)*

FROM

όνομα Πίνακα

WHERE

λογική πρόταση;

- ▣ Επιστρέφει συγκεκριμένα πεδία από όλες τις εγγραφές ενός πίνακα που ικανοποιούν την λογική πρόταση.
- ▣ Με το αστεράκι επιστρέφουμε όλα τα πεδία.

□ Παράδειγμα

SELECT

name, lastname

FROM

Artist

WHERE

country = "USA";



<i>name</i>	<i>lastname</i>
Michael	Jackson
Whitney	Houston

Πίνακας Αποτελέσματος

- Ο πίνακας που είναι αποθηκευμένος και ο πίνακας που επιστρέφει το ερώτημα μπορεί να είναι διαφορετικοί.
- Διαφορετικά πεδία
 - ▣ Η απάντηση έχει συνήθως μόνο κάποια πεδία.
- Διαφορετικές εγγραφές
 - ▣ Η απάντηση συνήθως έχει μόνο κάποιες εγγραφές.
- Ο προσωρινός πίνακας που επιστρέφεται ονομάζεται **πίνακας αποτελέσματος**.

Παράδειγμα

□ Αποθηκευμένος Πίνακας

albumid	title	year	studioid
1	Thriller	1982	1
2	Off The Wall	1979	1
3	Bad	1984	1
4	Τσικουλάτα	2004	2

□ Πίνακας Αποτελέσματος

title	year
Thriller	1982
Off The Wall	1979
Bad	1984

```
SELECT  
    title, year  
FROM  
    Album  
WHERE  
    studioid = 1;
```

Αυτός που αλλού έχει παίξει?

1. Βάσεις Δεδομένων ✓
2. Πίνακες, Στήλες, Πεδία ✓
3. Σχήματα Βάσεων Δεδομένων ✓
4. Εισαγωγή στη γλώσσα SQL ✓
5. SELECT, WHERE ✓
6. INSERT INTO, DELETE, UPDATE

Δημιουργία Εγγραφής

□ Ερώτημα *INSERT INTO*

▣ Σύνταξη

INSERT INTO

όνομα Πίνακα

SET

πεδίο1 = τιμή1,

πεδίο2 = τιμή2,

...

πεδίοN = τιμήN;

▣ Εναλλακτική Σύνταξη

INSERT INTO

όνομα Πίνακα

(πεδίο1 , πεδίο2 ,..., πεδίοN)

Values

(τιμή1, τιμή2 ,..., τιμήN);

Παράδειγμα Δημιουργίας

INSERT INTO

Artist

SET

name = "Michael",

lastname = "Jackson",

birthdate = "29/08/1959";

- Παρατηρούμε ότι παραλείπουμε την τιμή για το πεδίο `id`, που το έχουμε ορίσει `AUTO_INCREMENT`.
- Όσα πεδία παραλείπονται παίρνουν:
 - Την προεπιλεγμένη τιμή που ορίζουμε στο σχήμα μας.
 - Την προεπιλεγμένη τιμή σύμφωνα με τον τύπο τους αν δεν έχουμε ορίσει εμείς μια.

Ενημέρωση Εγγραφών

□ Ερώτημα *UPDATE*

□ Σύνταξη

UPDATE

όνομα Πίνακα

SET

πεδίο1 = τιμή1,

πεδίο2 = τιμή2,

...

πεδίοN = τιμήN;

□ Αλλάζει όλες τις εγγραφές του πίνακα.

□ Τα πεδιά που δεν εμφανίζονται δεν αλλάζουν.

□ Εναλλακτική Σύνταξη

UPDATE

όνομα Πίνακα

SET

πεδίο1 = τιμή1,

πεδίο2 = τιμή2,

...

πεδίοN = τιμήN

WHERE

λογική έκφραση

Παράδειγμα Ενημέρωσης

UPDATE

Artist

SET

lastname = "Bolton"

WHERE

name = "Michael";



Διαγραφή Εγγραφών

□ Ερώτημα *DELETE*

□ Σύνταξη

DELETE FROM

όνομα Πίνακα

- Διαγράφει όλες τις εγγραφές του πίνακα.

□ Εναλλακτική Σύνταξη

DELETE FROM

όνομα Πίνακα

WHERE

λογική έκφραση

- Φιλτράρουμε τις εγγραφές που θα διαγραφούν με την *WHERE*.

Παράδειγμα Ενημέρωσης

DELETE FROM

Artist

WHERE

name = "Σώτης";



Κοντά στο Happy End!

1. Βάσεις Δεδομένων ✓
2. Πίνακες, Στήλες, Πεδία ✓
3. Σχήματα Βάσεων Δεδομένων ✓
4. Εισαγωγή στη γλώσσα SQL ✓
5. SELECT, WHERE ✓
6. INSERT INTO, DELETE, UPDATE ✓
7. Συνδυασμός PHP/MySQL

Συνδυασμός PHP και MySQL

- Απαιτείται η επέκταση MySQL της PHP
- Πάλι καλά που τα εγκαταστήσαμε όλα μαζί...😊
- Τα SQL ερωτήματα για την PHP είναι **αλφαριθμητικά**.
- Η βιβλιοθήκη MySQL για PHP περιέχει **συναρτήσεις** που τις καλούμε για να πετύχουμε επικοινωνία PHP και MySQL μέσω των ερωτημάτων.
- Τα ερωτήματα περνούν ως **παράμετροι**.

Σύνδεση στη Βάση

□ Εντολή *mysql_connect()*

□ Σύνταξη

mysql_connect(διεύθυνση , χρήστης , κωδικός);

□ Διεύθυνση

- Σε ποιόν υπολογιστή τρέχει ο MySQL Server.
- Συνήθως localhost.

□ Χρήστης – Κωδικός

- Είναι απαραίτητα για την πρόσβαση στην Βάση
- Ρυθμίζονται κατά την εγκατάσταση της MySQL ή από το phpMyAdmin.

□ Η σύνδεση κλείνει αυτόματα στο τέλος του script.

Ερωτήματα σε Βάση

□ Εντολή *mysql_select_db()*

▣ Σύνταξη

mysql_select_db(όνομα βάσης)

▣ Επιλέγει σε ποια βάση θα τρέξουν τα ερωτήματα.

- Ο MySQL Server μπορεί να αποθηκεύει πολλές βάσεις.
- Συνήθως έχουμε μια για κάθε web εφαρμογή.

□ Εντολή *mysql_query()*

▣ Σύνταξη

\$res = mysql_query(ερώτημα)

▣ Εκτελεί το ερώτημα και επιστρέφει την απάντηση.

- True / False σε περίπτωση ερωτημάτων Δημιουργίας, Ενημέρωσης, Διαγραφής.
- Ένα resource (πίνακας συνήθως) που μπορεί να χρησιμοποιηθεί σε περίπτωση Ανάγνωσης.

Εκτέλεση Ερωτήματος

```
mysql_connect("localhost","Anty","Stewie");  
mysql_select_db( "Music" );  
$success = mysql_query(  
    "UPDATE  
    Artist  
    SET  
    lastname = 'Bolton'  
    WHERE  
    name = 'Michael'; "  
);
```

Χρήσιμες Εντολές PHP

- *mysql_insert_id()*
 - ▣ Επιστρέφει το *AUTO_INCREMENT* της εγγραφής που δημιουργήθηκε στο πιο πρόσφατο ερώτημα.
- *mysql_affected_rows()*
 - ▣ Επιστρέφει το πλήθος των γραμμών που άλλαξαν από την εκτέλεση του πιο πρόσφατου ερωτήματος.
- *mysql_num_rows(\$res)*
 - ▣ Παίρνει ως παράμετρο το **resource** που επιστρέφει η *mysql_query()*
 - ▣ Επιστρέφει το πλήθος των εγγραφών του αποτελέσματος.

Παράδειγμα Εισαγωγής Εγγραφής

□ database.php

```
mysql_connect("localhost","Anty","Stewie");
```

```
mysql_select_db( "Music" );
```

□ Συνήθως χρησιμοποιούμε ένα αρχείο php για την σύνδεση μας με την βάση, το οποίο το κάνουμε include στο script που κάνουμε το ερώτημα.

Παράδειγμα Εισαγωγής Εγγραφής (2)

```
include "database.php";  
mysql_query(  
    "INSERT INTO  
        Artist  
    SET  
        name = 'Mariah',  
        lastname = 'Carey',  
        birthdate = '27/03/1970',  
        country = 'USA'; "  
);  
$id = mysql_insert_id();  
echo "Καταχωρήθηκε ο καλλιτέχνης $id";
```



Παράδειγμα Διαγραφής Δεδομένων

```
include "database.php";
mysql_query(
    "DELETE FROM
        Album
    WHERE
        title = 'Τσικουλάτα'; "
);
if (mysql_affected_rows() == 1){
    echo "Μπούλο ο Ξανθιώτης";
}
else {
    echo "Αιντι μάλε τσικουλάτα τσικιτα, τσικι τσικι τσικιτά";
}
```

Παράδειγμα Ανάγνωσης Δεδομένων

```
include "database.php";  
$res = mysql_query(  
    "SELECT  
        title  
    FROM  
        Album  
    WHERE  
        studioid = 2; "  
);  
echo "Έχουν ηχογραφηθεί " . mysql_num_rows( $res );  
echo "δισκάρες στη Vasipar!";
```



Έχουν ηχογραφηθεί 10 δισκάρες στην Vasipar!

Ανάγνωση Δεδομένων

- Εντολή *mysql_fetch_array()*
 - ▣ Σύνταξη
mysql_fetch_array(\$res)
 - ▣ Παίρνει ως παράμετρο το resource που επέστρεψε το *mysql_query()*.
 - ▣ Αν ο πίνακας αποτελέσματος έχει **N** εγγραφές, επιστρέφει
 - Την πρώτη φορά, την πρώτη εγγραφή του πίνακα αποτελέσματος.
 - Την δεύτερη φορά, την δεύτερη εγγραφή του πίνακα αποτελέσματος.
 - ...
 - Την N-οστή φορά, την τελευταία εγγραφή του πίνακα αποτελέσματος.
 - Την (N+1)-οστή φορά που θα κληθεί, false.

Ανάγνωση Δεδομένων (2)

- Η πρώτη κλήση της `mysql_fetch_array($res)` επιστρέφει ένα λεξικό.
- Αντιπροσωπεύει την **πρώτη** εγγραφή του αποτελέσματος.
- Κλειδιά: τα **ονόματα των πεδίων**
- Τιμές: οι τιμές των συγκεκριμένων πεδίων για την συγκεκριμένη εγγραφή.
- κτλ. για τις πρώτες N κλήσεις.
- Μέχρι να εξαντληθεί ο πίνακας αποτελέσματος.

name	lastname
Michael	Jackson
Whitney	Houston
Tarja	Turunen

```
$sql = "SELECT
        name, lastname
FROM
Artist
WHERE
artistid < 4;";
```

```
include "database.php";
```

```
$res = mysql_query( $sql );
```

```
$first = mysql_fetch_array( $res );
```

```
$second = mysql_fetch_array( $res );
```

```
$third = mysql_fetch_array( $res );
```

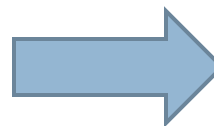
```
$end = mysql_fetch_array( $res );
```

```
echo $first[ "name" ] . "<br />";
```

```
echo $second[ "name" ] . " ";
```

```
echo $second[ "lastname" ] . "<br />";
```

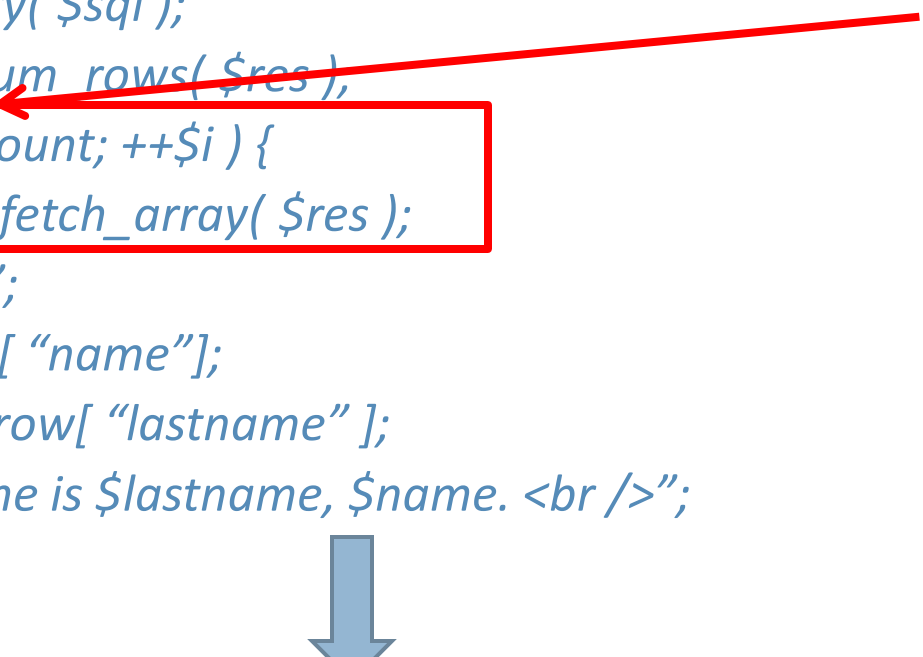
```
echo $third[ "lastname" ] . "<br />";
```



```
Michael
Whitney Houston
Turunen
```

Παράδειγμα Ανάγνωσης

```
include "database.php";
$sql = "...";
$res = mysql_query( $sql );
$count = mysql_num_rows( $res );
for ( $i = 0; $i < $count; ++$i ) {
    $row = mysql_fetch_array( $res );
    echo "Row $i:";
    $name = $row[ "name" ];
    $lastname = $row[ "lastname" ];
    echo "The name is $lastname, $name. <br />";
    ?>
}
```



Row 0: The name is Jackson, Michael.
Row 1: The name is Houston, Whitney.
Row 2: The name is Turunen, Tarja.



Διαλέχτε, καλέ κυρία, Διαλέχτε

1. Ταξινόμηση
2. Περιορισμοί
3. Τελεστές LIKE, IN, BETWEEN
4. Ψευδώνυμα - Ημερομηνίες
5. Ενώσεις Πινάκων

Έχω πρώτο πράγμα...

1. Ταξινόμηση

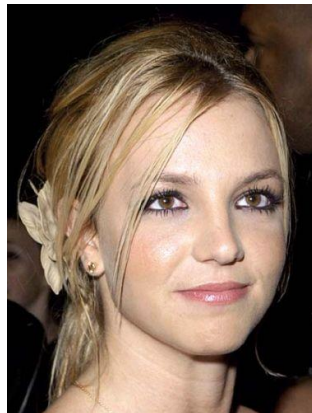
Ταξινόμηση

- Εντολή *ORDER BY*
 - ▣ Ταξινομεί το αποτέλεσμα με βάση **ένα πεδίο**.
 - *ASC*: Αύξουσα Ταξινόμηση (προεπιλογή).
 - *DESC*: Φθίνουσα Ταξινόμηση.
 - ▣ Ακολουθεί πάντα τον όρο *WHERE*.
 - Είτε υπάρχει είτε όχι.
- Παράδειγμα
*Select πεδία FROM πίνακας WHERE συνθήκη
ORDER BY πεδιο ASC (DESC);*

Ταξινόμηση (2)

□ Αποθηκευμένος πίνακας **Artist**

artistid	name	lastname	birthdate	country	genre
1	Tori	Amos	1963-08-22	USA	Pop / Jazz
2	Britney	Spears	1981-12-02	USA	Pop
3	Bruce	Dickinson	1958-08-07	England	Metal
4	Μιχαλάρας	Ρακιντζής	1962-01-05	Greece	Λίγο από όλα



Ταξινόμηση (3)

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
ORDER BY
    birthdate ASC;
```

□ Πίνακας αποτελέσματος

Artistid	name	lastname	birthdate
3	Bruce	Dickinson	1958-08-07
4	Μιχαλάρας	Ρακιντζής	1962-01-05
1	Tori	Amos	1963-08-22
2	Britney	Spears	1981-12-02

Ταξινόμηση (4)

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
ORDER BY
    birthdate DESC;
```

□ Πίνακας αποτελέσματος

Artistid	name	lastname	birthdate
2	Britney	Spears	1981-12-02
1	Tori	Amos	1963-08-22
4	Μιχαλάρας	Ρακιντζής	1962-01-05
3	Bruce	Dickinson	1958-08-07

Ταξινόμηση (5)

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
WHERE
    name = 'Britney' OR lastname = 'Amos'
ORDER BY
    birthdate DESC;
```

□ Πίνακας αποτελέσματος

Artistid	name	lastname	birthdate
2	Britney	Spears	1981-12-02
1	Tori	Amos	1963-08-22

Ότι πάρεις ένα ευρώ, ότι πάρεις.

1. Ταξινόμηση ✓
2. Περιορισμοί

Περιορισμοί

□ Εντολή *LIMIT*

- ▣ Περιορίζει τον πλήθος των εγγραφών του πίνακα αποτελέσματος.
- ▣ Ο όρος αυτός ακολουθεί το *ORDER BY*.
 - Είτε υπάρχει, είτε δεν υπάρχει.
- ▣ Κρατάει τα πρώτα **N** στοιχεία.

□ Παράδειγμα

*Select πεδία FROM πίνακας WHERE συνθήκη
ORDER BY πεδίο LIMIT N;*

Περιορισμοί (2)

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
ORDER BY
    birthdate ASC
LIMIT 3;
```

□ Πίνακας αποτελέσματος

Artistid	name	lastname	birthdate
3	Bruce	Dickinson	1958-08-07
4	Μιχαλάρας	Ρακιντζής	1962-01-05
1	Tori	Amos	1963-08-22

Τρία αποτελέσματα

← Οι υπόλοιπες εγγραφές παραλείπονται.

Περιορισμοί (2)

- Ποια είναι τα στοιχεία του γηραιότερου καλλιτέχνη της βάσης?

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
ORDER BY
    birthdate ASC
LIMIT 1;
```



Artistid	name	lastname	birthdate
3	Bruce	Dickinson	1958-08-07

Περιορισμοί (3)

- Το *LIMIT* εφαρμόζεται και σε ερωτήματα *UPDATE* και *DELETE*.
- **ΚΑΛΗ ΠΡΑΚΤΙΚΗ!**
 - ▣ *LIMIT 1* σε ερωτήματα που περιμένουμε να αλλάξουν / διαγράψουν μόνο μια εγγραφή.
 - ▣ Θα σώσουν την κατάσταση αν ξεχάσεις το *WHERE*.

```
DELETE FROM  
  Artist  
WHERE  
  artistid = 3  
LIMIT 1;
```

Περιορισμοί (4)

- Πλήρης Σύνταξη του *LIMIT*
 - *LIMIT offset, N*
- Σε αυτή τη μορφή
 - Παραλείπει εγγραφές από την αρχή του αποτελέσματος ίσες με το offset.
 - Αν $\text{offset} = 0 \rightarrow$ καμία παράλειψη.
 - Περιορίζει το πλήθος των εγγραφών του αποτελέσματος κατά N .

Παράδειγμα LIMIT με offset

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
ORDER BY
    birthdate DESC
LIMIT 1, 2;
```

1 εγγραφή από την αρχή παραλείπεται

□ Πίνακας αποτελέσματος

Artistid	name	lastname	birthdate
2	Britney	Spears	1981-12-02
1	Tori	Amos	1963-08-22
4	Μιχαλάρας	Ρακιντζής	1962-??-??
3	Bruce	Dickinson	1958-08-07



Δύο έγγραφές

Παράδειγμα LIMIT με offset (2)

- Ποια είναι τα στοιχεία του δεύτερου γηραιότερου καλλιτέχνη της βάσης?

```
SELECT
    artistid, name, lastname, birthdate
FROM
    Artist
ORDER BY
    birthdate ASC
LIMIT 1,1;
```

Artistid	name	lastname	birthdate
4	Μιχαλάρας	Ρακιντζής	1962-01-05



Καρέκλες και τραπέζια και καρέκλες έχω!!!

1. Ταξινόμηση ✓

2. Περιγραφισμοί ✓

3. Τελεστές LIKE, IN, BETWEEN

Αναζήτηση μέσα σε Αλφαριθμητικά

□ Τελεστής *LIKE*

□ Σύνταξη

WHERE πεδίο LIKE έκφραση

□ Έκφραση

- Αλφαριθμητικό που περιέχει % ως χαρακτήρα-μπαλαντέρ.
- Σημαίνει οποιοσδήποτε χαρακτήρας.

□ Παραδείγματα

“Kiss%”: **Αρχίζει από “Kiss”**

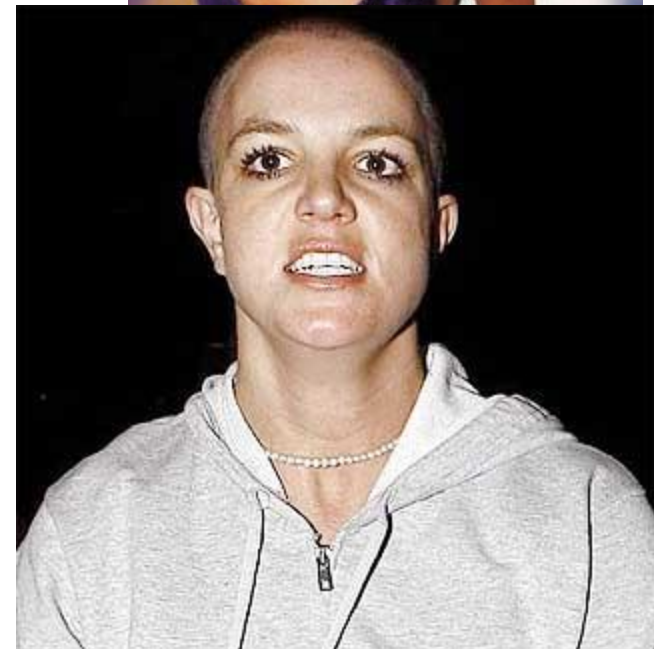
- “Kiss Me” **OK**
- “Gimme A Kiss Kiss” **NOT OK**

“Kiss%You”: **Αρχίζει από “Kiss” και τελειώνει με “You”**

- “Kissing You” **OK**
- “Kiss You and your sister!!!” **NOT OK**

Παράδειγμα LIKE

```
SELECT  
    artistid, name, lastname, birthdate  
FROM  
    Artist  
WHERE  
    lastname LIKE 'S%'  
LIMIT 1;
```



Artistid	name	lastname	birthdate
2	Britney	Spears	1981-12-02

Αναζήτηση μέσα σε Αλφαριθμητικά (2)

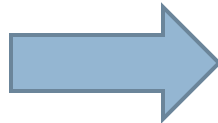
□ Τελεστής *IN*

□ Σύνταξη

Πεδίο IN (τιμή1, τιμή2, ... , τιμήN)

- Ελέγχει αν μια τιμή ενός πεδίου ανήκει σε ένα σύνολο τιμών.

```
SELECT  
    artistid, name, lastname  
FROM  
    Artist  
WHERE  
    name IN ('Tori', 'Bruce');
```



artistid	name	lastname
1	Tori	Amos
3	Bruce	Dickinson

Σύγκριση

- Οι τελεστές $=$, $<$, $>$, $<=$, $>=$ λειτουργούν σε:
 - Αριθμούς
 - $3 < 4$
 - $5 = 5$
 - $12.2 > -9$
 - Αλφαριθμητικά (λεξικογραφικά)
 - $'Tori' >= 'Bruce'$
 - Ημερομηνίες (χρονολογικά)
 - $'2001-09-11' > '1987-11-30'$
 - $'2011-11-22 00:00:00' <= '2011-11-22 01:01:01'$

Αναζήτηση μέσα σε Αλφαριθμητικά (3)

□ Τελεστής *BETWEEN... AND*

▣ Σύνταξη

Πεδίο BETWEEN κάτω_φράγμα AND άνω_φράγμα

- ▣ Ελέγχει αν μια τιμή βρίσκεται μεταξύ των δύο φραγμάτων (μεγαλύτερο ίσο / μικρότερο ίσο).

```
SELECT
    name, lastname
FROM
    Artist
WHERE
    name BETWEEN 'Tori' AND 'Zed';
```

name	lastname
Tori	Amos



Τα δύο μισή τιμή σε λέω...

1. Ταξινόμηση ✓
2. Περιγραφισμοί ✓
3. Τελειστάς LIKE, IN, BETWEEN ✓
4. Ψευδώνυμα - Ημερομηνίες

Ψευδώνυμα Στηλών

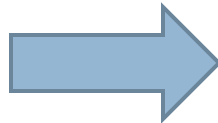
- Εισάγονται με το *AS*

- Σύνταξη

πεδίο AS ψευδώνυμο

- Αλλάζουν το όνομα μιας στήλης του αποθηκευμένου πίνακα στον πίνακα αποτελέσματος.

```
SELECT
    artistid AS code, name, lastname
FROM
    Artist
WHERE
    name IN ('Tori', 'Bruce');
```



code	name	lastname
1	Tori	Amos
3	Bruce	Dickinson

Τελεστές Ημερομηνιών

- Λέξη-κλειδί *INTERVAL*
- Λέξεις-κλειδιά διάρκειας:
 - *SECOND*
 - *MINUTE*
 - *HOUR*
 - *DAY*
 - *MONTH*
 - *YEAR*
- Μορφή: *INTERVAL N διάρκεια*
- Μπορεί να προστεθεί / αφαιρεθεί από ημερομηνία για να δώσει μία άλλη

Τελεστές Ημερομηνιών (2)

- Συνάρτηση *NOW()*
 - ▣ Επιστρέφει την τωρινή ημερομηνία/ώρα.

NOW(): '2010 – 11 – 30 20:00:21'

Παράδειγμα NOW()

- Πως θα δείξω ποίοι είναι μέσα στην σελίδα μου?
 - ▣ Στήλη lastactive στον πίνακα users
 - ▣ Ενημέρωση κάθε φορά που ανοίγει ένα PHP αρχείο:

```
UPDATE
    users
SET
    lastactive = NOW()
WHERE
    userid = 5
LIMIT 1;
```

- ▣ Ερώτημα για να πάρω τους online χρήστες:

```
SELECT
    username
FROM
    users
WHERE
    lastactive + INTERVAL 5 MINUTE > NOW();
```

Πράξεις κατά την Επιλογή

- Το SELECT επιτρέπει και πράξεις κατά την επιλογή
- Δηλαδή τα πεδία στην πραγματικότητα είναι παραστάσεις.

SELECT

(artistid + 10) AS code, name, lastname

FROM

Artist

WHERE

name IN ('Tori', 'Bruce');

code	name	lastname
11	Tori	Amos
13	Bruce	Dickinson

SELECT

name, lastname, (birthdate + INTERVAL 100 YEARS) AS death

FROM

Artist

WHERE

name IN ('Tori', 'Bruce');

name	lastname	death
Tori	Amos	2063-08-22
Bruce	Dickinson	2058-08-07

Κληρώνει δεν λερώνει...

1. Ταξινόμηση ✓
2. Περιγραφισμοί ✓
3. Τελειστάς LIKE, IN, BETWEEN ✓
4. Ψευδώνυμα – Ημερομηνίες ✓
5. Ενώσεις Πινάκων

ΕΝΩΣΕΙΣ ΠΙΝΑΚΩΝ

- Συνδυασμός δεδομένων από πολλούς διαθέσιμους αποθηκευμένους πίνακες.
- Ο πίνακας αποτελέσματος μπορεί να περιλαμβάνει δεδομένα από όποιους του ζητήσουμε.
- Μπορεί επίσης να φιλτράρει τα αποτελέσματα με χρήση πεδίων από όσους από αυτούς θέλουμε.
- Πολλάπλές ενώσεις, ενώνοντας το αποτέλεσμα μιας ένωσης πινάκων με άλλον πίνακα.

ΑΠΛΗ ΕΝΩΣΗ

- Είναι η ένωση δύο ή περισσότερων πινάκων σε ένα ερώτημα.
 - ▣ Για να πάρουμε δεδομένα από όλους.
 - ▣ Για να βάλουμε κάποια κριτήρια στα αποτελέσματά μας.
- Για να ξεχωρίσουμε τα πεδία που έχουν κοινά ονόματα κάνουμε αναφορά σε πεδίο πίνακα
 - ▣ πίνακας.πεδίο

Παράδειγμα

studioid	name	location	funded
1	Westlake	USA	1974
2	Vasipap	Greece	2001

albumid	name	year	studioid
1	Thriller	1982	1
3	Bad	1984	1
4	Τσικουλάτα	2004	2

SELECT

Album.name AS title, year, Studio.name AS recording_studio

FROM

Album, Studio

WHERE

Studio.studioid = Album.studioid;

ΑΠΛΗ ΕΝΩΣΗ (2)

- Σωστότερη σύνταξη

```
SELECT  
    πεδία  
FROM  
    A  
INNER JOIN B  
    ON A.x = B.y
```

- Έτσι το *WHERE* μπορεί να χρησιμοποιηθεί και πάλι για φιλτράρισμα και όχι για ένωση πινάκων.

Παραδείγμα (2)

SELECT

Album.name AS title, year, Studio.name AS recording_studio

FROM

Album

INNER JOIN

Studio ON Studio.studioid = Album.studioid

WHERE

Studio.studioid = 1;

title	year	recording_studio
Thriller	1982	Westlake
Bad	1984	Westlake

Δεύτερο Παραδείγμα

- Ποιο είναι το studio που ηχογραφήθηκε το πιο παλιό album?

SELECT

Studio.name AS recording_studio

FROM

Studio

INNER JOIN

Album ON Studio.studioid = Album.studioid

ORDER BY year ASC

LIMIT 1;

Τρίτο Παραδείγμα

- Όλοι οι τίτλοι των album και η τοποθεσία του studio που ηχογραφήθηκαν?

SELECT

Album.name AS title, Studio.location

FROM

Album

INNER JOIN

Studio ON Studio.studioid = Album.studioid;

name	location
Thriller	USA
Bad	USA
Τσικουλάτα	Greece

ΑΡΙΣΤΕΡΗ ΕΝΩΣΗ

- Στην περίπτωση που δεν έχουμε για όλα τα **studios** πληροφορίες.

studioid	name	location	funded
1	Westlake	USA	1974

- Τώρα αν κάνουμε το ερώτημα της απλής ένωσης, δεν θα επιστραφούν όλες οι τιμές.

```
SELECT
    Album.name AS title, Studio.location
FROM
    Album
INNER JOIN
    Studio ON Studio.studioid = Album.studioid;
```

ΑΡΙΣΤΕΡΗ ΕΝΩΣΗ (2)

□ Αποτέλεσμα

title	location
Thriller	USA
Bad	USA

- Κάποια **album** θα λείπουν επειδή λείπει το **studio** στο οποίο ηχογραφήθηκε.
- Εμείς όμως θέλουμε τις πληροφορίες του **album** ακόμα και αν δεν έχουμε στοιχεία για το **studio**.

ΑΡΙΣΤΕΡΗ ΕΝΩΣΗ (3)

- Αλλάζουμε το *INNER JOIN* με *LEFT OUTER JOIN*
 - ▣ Αν **υπάρχει** το κλειδί της ένωσης στον δεξιό πίνακα
 - Κάνει το ίδιο με την απλή ένωση.
 - ▣ Αν καμιά εγγραφή του δεξιού πίνακα **δεν ταιριάζει** με την εκάστοτε του αριστερού
 - Κρατάει τα στοιχεία της εγγραφής του αριστερού πίνακα.
 - Γεμίζει τα στοιχεία των στηλών του δεξιού πίνακα του αποτελέσματος με *NULL*.

Παράδειγμα Αριστερής Ένωσης

SELECT

Album.name AS title, Studio.location

FROM

Album

LEFT OUTER JOIN

Studio ON Studio.studioid = Album.studioid;

title	location
Thriller	USA
Bad	USA
Τσικουλάτα	NULL

Καρτεσιανό Γινόμενο

□ A x B

- Κάθε εγγραφή του πίνακα A.
- Συνδυάζεται με κάθε εγγραφή του πίνακα B.

□ Παράδειγμα

studioid	name
1	Westlake
2	Vasipap

Studio

x

albumid	title	year	studio
1	Thriller	1982	1
3	Bad	1984	1
4	Τσικουλάτα	2004	2

Album

Καρτεσιανό Γινόμενο (2)

□ Αποτέλεσμα

albumid	title	year	studio	studioid	name
1	Thriller	1982	1	1	Westlake
1	Thriller	1982	1	2	Vasipap
3	Bad	1984	1	1	Westlake
3	Bad	1984	1	2	Vasipap
4	Τσικουλάτα	2004	2	1	Westlake
4	Τσικουλάτα	2004	2	2	Vasipap

Παράδειγμα

```
SELECT  
    title, year, name  
FROM  
    Album, Studio
```

title	year	name
Thriller	1982	Westlake
Thriller	1982	Vasipap
Bad	1984	Westlake
Bad	1984	Vasipap
Τσικουλάτα	2004	Westlake
Τσικουλάτα	2004	Vasipap

Παράδειγμα (2)

- Από μόνο του **άχρηστο**.
- Πολύ πιο χρήσιμο με το *WHERE*.

```
SELECT  
  title, year, name  
FROM  
  Album, Studio  
WHERE  
  studio = studioid
```

title	year	name
Thriller	1982	Westlake
Bad	1984	Westlake
Τσικουλάτα	2004	Vasipap

Παράδειγμα Πολλαπλών Ενώσεων

user:

- userid
- username
- password
- email

blog:

- blogid
- title
- text
- userid

comment:

- commentid
- text
- blogid
- userid
- created

Παράδειγμα Πολλαπλών Ενώσεων (2)

- Όλοι οι τίτλοι και τα κείμενα από τις αναρτήσεις (postblogs) του χρήστη #5

```
SELECT  
    text, title  
FROM  
    blog  
WHERE  
    userid = 5
```


Παράδειγμα Πολλαπλών Ενώσεων (3)

- Το κείμενο από όλα τα σχόλια που έγιναν σε όλα τα postblogs του χρήστη #5

SELECT

comment.text

FROM

comment INNER JOIN blog

ON comment.blogid = blog.blogid

WHERE

blog.userid = 5

Χρησιμοποιούμε την ένωση για να πάρουμε στοιχεία από τον πίνακα comment και να φιλτράρουμε με βάση τον πίνακα blog.

Παράδειγμα Πολλαπλών Ενώσεων (4)

- Το κείμενο και ο χρήστης που έγραψε το σχόλιο από όλα τα σχόλια που έγιναν σε όλα τα postblogs του χρήστη #5

SELECT

comment.text, user.name

FROM

comment INNER JOIN blogs

ON comment.blogid = blog.blogid

INNER JOIN user

ON user.userid = comment.userid

WHERE

blog.userid = 5

Χρησιμοποιούμε την ένωση για να πάρουμε στοιχεία από τον πίνακα comment και user και να φιλτράρουμε με βάση τον πίνακα blog (ένωση στην ένωση).

Παράδειγμα Πολλαπλών Ενώσεων (5)

- Το κείμενο από το πιο **πρόσφατο** σχόλιο που έγινε σε κάποιο blog του χρήστη #5 μαζί με τον τίτλο του blog;

SELECT

blog.title, comment.text

FROM

comment INNER JOIN blog

ON comment.blogid = blog.blogid

WHERE blog.userid = 5

ORDER BY

comment.created DESC

LIMIT 1

Χρησιμοποιούμε την ένωση για να πάρουμε στοιχεία από τον πίνακα comment και comment και για να φιλτράρουμε με χρήση αναφοράς.

Τα μάθατε καλέ κυρία?

1. Ψευδώνυμα Πινάκων
2. Ένωση Πίνακα με τον εαυτό του
3. Ομαδοποίηση
4. Aggregated Functions (Συναρτήσεις Σύνοψης)
5. Υποερωτήματα

Τι καλέ???

1. Ψευδώνυμα Πινάκων

Ψευδώνυμα Πινάκων

- Εκτός από τα πεδιά, στην SQL επιτρέπεται να δώσουμε ψευδώνυμο ακόμα και σε ολόκληρο πίνακα.
- Σύνταξη
 - ▣ Εισάγονται με το *AS*.
 - ▣ *όνομα πίνακα AS ψευδώνυμο.*
- Χρησιμοποιούνται για να μας διευκολύνουν όταν έχουμε πολλαπλές ενώσεις ή πολύπλοκα ερωτήματα.

Παραδείγμα

SELECT

Album.name AS title, year, Studio.name AS recording_studio

FROM

Album

INNER JOIN

Studio ON Studio.studioid = Album.studioid

WHERE

Studio.studioid = 1;

title	year	recording_studio
Thriller	1982	Westlake
Bad	1984	Westlake

Παραδείγμα (2)

SELECT

a.name AS title, year, b.name AS recording_studio

FROM

Album AS a

INNER JOIN

Studio AS b ON b.studioid = a.studioid

WHERE

b.studioid = 1;

title	year	recording_studio
Thriller	1982	Westlake
Bad	1984	Westlake

ΝΕΑ ΓΑΡΙΔΑΚΙΑ ΛΟΤΑΡΙΑ!!!

1. Ψευδώνυμα Πινάκων
2. Ένωση Πίνακα με τον εαυτό του

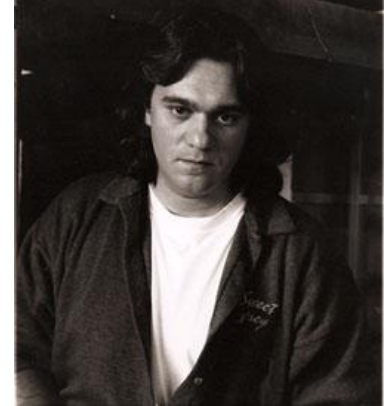
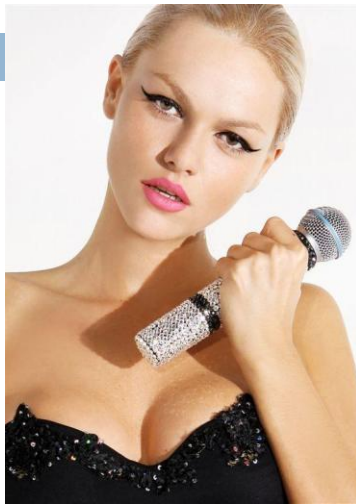
Αυτοένωση Πινάκων

- Στην SQL επιτρέπεται και η ένωση ενός πίνακα με τον εαυτό του.
- Σε αυτή την περίπτωση είναι απαραίτητο να έχουν ψευδώνυμα οι πίνακες καθώς μόνο έτσι μπορούμε να τους ξεχωρίσουμε μεταξύ τους.
- Επιτρέπεται να χρησιμοποιήσουμε όλων των ειδών τις ενώσεις (Απλή / Αριστερή Ένωση, Καρτεσιανό Γινόμενο).

Παράδειγμα

person:

- personid
- name
- surname
- fatherid



personid	name	lastname	fatherid
1	Νάταλι	Θάνου	2
2	Παναγιώτης	Τόκος	NULL
3	Ακίνδυνος	Γκίκας	2
4	Βασίλης	Λεβέντης	NULL
5	Τάκης	Γκώνιας	4
6	Ανίτα	Πάνια	5

Παράδειγμα (2)

- Εμφάνιση όλων των ατόμων μαζί με το όνομα πατρός όπου αυτό υπάρχει.

SELECT

*child.name AS name,
child.lastname AS lastname,
father.name AS fathername*

FROM

person AS child

LEFT OUTER JOIN

person AS father

ON child.fatherid = father.personid;

Αποτέλεσμα (Varona)

name	lastname	fathername
Νάταλι	Θάνου	Μπάμπης
Παναγιώτης	Τόκος	NULL
Ακίνδυνος	Γκίκας	Μπάμπης
Βασίλης	Λεβέντης	NULL
Τάκης	Γκώνιας	Βασίλης
Ανίτα	Πάνια	Τάκης

Δεύτερο Παράδειγμα

- Να εμφανίσει το ονοματεπώνυμο των ατόμων που έχουν για πατέρα κάποιον Μπάμπη.

SELECT

child.name AS name,

child.lastname AS lastname,

FROM

person AS child

INNER JOIN person AS father

ON child.fatherid = father.personid

WHERE

father.name = 'Παναγιώτης';

name	lastname
Νάταλι	Θάνου
Ακίνδυνος	Γκίκας

Τρίτο Παράδειγμα

- Να εμφανίσει το ονοματεπώνυμο των ατόμων που ξέρουμε το όνομα του πατέρα τους.

SELECT

child.name AS name,

child.lastname AS lastname,

FROM

person AS child

INNER JOIN person AS father



ON child.fatherid = father.personid

WHERE

father.name IS NOT NULL;

name	lastname
Νάταλι	Θάνου
Ακίνδυνος	Γκίκας
Τάκης	Γκώνιας
Ανίτα	Πάνια

ΓΙΟΥΠΙ!!!!

1. Ψευδοπίνακα Πινάκων 
2. Ένωση Πίνακα με τον εαυτό του 
3. Ομαδοποίηση

Ομαδοποίηση

- Στην SQL μας δίνεται η δυνατότητα να ομαδοποιούμε τα αποτελέσματά μας.
- Η ομαδοποίηση γίνεται βάσει κάποιου κοινού χαρακτηριστικού
 - ▣ Τιμή μιας στήλης ή παράστασης.
 - ▣ Ένα ολόκληρο πεδίο.
- Σύνταξη
 - ▣ *GROUP BY πεδίο*
- Εμφανίζεται μετά τον όρο WHERE και πριν τον ORDER.
*Select πεδία FROM πίνακας WHERE συνθήκη
GROUP BY πεδίο ORDER BY πεδίο;*

Παράδειγμα Λάθος Χρήσης

name	lastname
Μπάμπης	Στόκας
Μπάμπης	Τόκος
Βασίλης	Καήλας
Βασίλης	Λεβέντης
Τάκης	Τόκος

GROUP BY name



name	lastname
Μπάμπης	???
Βασίλης	???
Τάκης	Τόκος

name	lastname
Μπάμπης	Στόκας
Μπάμπης	Τόκος
Βασίλης	Καήλας
Βασίλης	Λεβέντης
Τάκης	Τόκος

GROUP BY lastname



name	lastname
Μπάμπης	Στόκας
????	Τόκος
Βασίλης	Καήλας
Βασίλης	Λεβέντης

Που είναι ρε μόρτη το λάθος???

- Όταν χρησιμοποιείς την εντολή *GROUP BY* για ομαδοποίηση, τότε τα πεδία που κάνεις *SELECT* πρέπει οπωσδήποτε
 - ▣ Είτε να είναι το πεδίο σύμφωνα με το οποίο κάνεις ομαδοποίηση.
 - ▣ Είτε να περιέχεται σε μια **Aggregated Function** (Συνάρτηση Σύνοψης).

Παράδειγμα Σωστής Χρήσης

- Να βρεθούν όλα τα διαφορετικά μικρά ονόματα των ατόμων της βάσης.

```
SELECT  
  name  
FROM  
  people  
GROUP BY  
  name;
```



name
Μπάμπης
Βασίλης
Τάκης

Τελεστής DISTINCT

- Την ίδια ακριβώς δουλειά με το προηγούμενο παράδειγμα μπορούμε να την κάνουμε χωρίς χρήση ομαδοποίησης αλλά με τον τελεστή *DISTINCT*.

- Παράδειγμα

SELECT

DISTINCT name

FROM

people



name
Μπάμπης
Βασίλης
Τάκης

ΠΑΜΕ ΝΑ ΤΑ ΑΓΟΡΑΣΟΥΜΕ ΟΛΑΑΑΑ!!!

1. Ψευδώνυμα Πινάκων ✓
2. Ένωση Πίνακα με τον εαυτό του ✓
3. Ομαδοποίηση ✓
4. Aggregated Functions (Συναρτήσεις Σύνοψης)

Aggregated Functions (Συνάρτηση Σύνοψης)

- Χρησιμοποιούνται αποκλειστικά σε ερωτήματα που υπάρχει ομαδοποίηση.
- Δίνουν διάφορες πληροφορίες για τις ομαδοποιημένες γραμμές.
 - ▣ *COUNT(*)*: πλήθος ομαδοποιημένων γραμμών.
 - ▣ *MIN(πεδίο)*: ελάχιστη τιμή των ομαδοποιημένων τιμών ενός πεδίου.
 - ▣ *MAX(πεδίο)*: μέγιστη τιμή των ομαδοποιημένων τιμών ενός πεδίου.
 - ▣ *AVG(πεδίο)*: μέσος όρος των ομαδοποιημένων τιμών ενός πεδίου.
- Για την εμφάνιση τους στον πίνακα αποτελέσματος, χρησιμοποιούμε ψευδώνυμα για κάθε μια από τις συναρτήσεις που χρησιμοποιούμε.

Παράδειγμα

name	lastname	grade	semester
Στέλιος	Μοσχογλου	9	3
Αντώνης	Χρυσόπουλος	2	3
Θεοδόσης	Σουργκούνης	3	5
Δημήτρης	Μανώλας	7	3
Θάνος	Κιντσάκης	10	5
Χριστίνα	Αυδίκου	5	5
Κυριακή	Καζά	10	5

Παράδειγμα (2)

- Ποια είναι η μέγιστη, η ελάχιστη και ο μέσος όρος βαθμολογίας κάθε εξαμήνου?

SELECT

semester,

MIN(grade) AS mingrade,

MAX(grade) AS maxgrade,

AVG(grade) AS avggrade

FROM

people

GROUP BY

semester;

Αποτέλεσμα Παραδείγματος

semester	mingrade	maxgrade	avggrade
3	2	9	6
5	3	10	7

ΣΕ ΕΠΙΛΕΓΜΕΝΑ ΠΕΡΙΠΤΕΡΑ ΚΑΙ ΣΟΥΠΕΡ ΜΑΡΚΕΤ!

1. Ψευδοκώδικας Πινάκων ✓
2. Ένωση Πίνακα με τον εαυτό του ✓
3. Ομαδοποίηση ✓
4. Aggregated Functions (Συναρτήσεις Σύνοψης) ✓
5. Υποερωτήματα

Υπερωτήματα

- Τα υπερρωτήματα χρησιμοποιούνται κατά κύριο λόγο
 - ▣ Στις ενώσεις πινάκων.
 - ▣ Στο φιλτράρισμα του πίνακα αποτελέσματος.
- Εμείς θα δούμε το πιο απλό παράδειγμα, πως μπορούμε να φιλτράρουμε δεδομένα με χρήση υπερρωτημάτων.

Παράδειγμα

```
SELECT  
  Album.name AS title, year, Studio.name AS  
  recording_studio  
FROM  
  Album  
WHERE  
  studioid IN  
(SELECT  
  studioid  
FROM  
  Studio  
WHERE location = 'Greece');
```