

# MYSQL 2

---

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος  
Επιμέλεια διαφανειών: Δ. Ζήνδρος

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών



# Στόχος της ώρας

- Περισσότερα θέματα MySQL
  - Ταξινόμηση
  - Περιορισμοί
  - Παράδειγμα δημιουργίας σχήματος
  - Τελεστές LIKE, IN, BETWEEN
  - Κωδικοποίηση
  - Ψευδώνυμα
  - Ημερομηνίες
  - Απλή ένωση
  - Αριστερή ένωση

# Ταξινόμηση αποτελέσματος

- **ORDER BY**: Ταξινομεί το αποτέλεσμα με βάση **ένα πεδίο**
  - **ASC**: Αύξουσα ταξινόμηση (προεπιλογή)
  - **DESC**: Φθίνουσα ταξινόμηση
- Ο όρος **ORDER BY** ακολουθεί τον όρο **WHERE**
  - (Το **WHERE** δεν απαιτείται)
- Μορφή:

```
SELECT πεδία FROM πίνακας WHERE συνθήκη  
ORDER BY πεδίο ASC;
```

```
SELECT πεδία FROM πίνακας WHERE συνθήκη  
ORDER BY πεδίο DESC;
```

# Ταξινόμηση αποτελέσματος

Αποθηκευμένος πίνακας **students**

student_id	name	surname	dob	email
1	Διονύσης	Ζήνδρος	1987-11-30	<a href="mailto:dionyziz@kamibu.com">dionyziz@kamibu.com</a>
2	Πέτρος	Αγγελάτος	1991-10-10	<a href="mailto:petros@kamibu.com">petros@kamibu.com</a>
3	Γιάννης	Παναρέτου	1985-09-07	<a href="mailto:george@gmail.com">george@gmail.com</a>
4	Ανέστης	Πατάκης	1990-01-01	<a href="mailto:a.patakis@ptakis.gr">a.patakis@ptakis.gr</a>
5	Γιώργος	Τζούμας	1980-07-27	<a href="mailto:joomla@hotmail.com">joomla@hotmail.com</a>
6	Οδυσσέας	Παππάς	1995-12-31	<a href="mailto:odycore@gmail.com">odycore@gmail.com</a>
8	Χρήστος	Πιπεράγκας	1996-01-14	<a href="mailto:chris14@hotmail.com">chris14@hotmail.com</a>

```
SELECT
    student_id, name, surname, dob
FROM
    students
ORDER BY
    dob ASC;
```

student_id	name	surname	dob
5	Γιώργος	Τζούμας	1980-07-27
3	Γιάννης	Παναρέτου	1985-09-07
1	Διονύσης	Ζήνδρος	1987-11-30
4	Ανέστης	Πατάκης	1990-01-01
2	Πέτρος	Αγγελάτος	1991-10-10
6	Οδυσσέας	Παππάς	1995-12-31
8	Χρήστος	Πιπεράγκας	1996-01-14

```
SELECT
    student_id, name, surname, dob
FROM
    students
ORDER BY
    dob DESC;
```

student_id	name	surname	dob
8	Χρήστος	Πιπεράγκας	1996-01-14
6	Οδυσσέας	Παππάς	1995-12-31
2	Πέτρος	Αγγελάτος	1991-10-10
4	Ανέστης	Πατάκης	1990-01-01
1	Διονύσης	Ζήνδρος	1987-11-30
3	Γιάννης	Παναρέτου	1985-09-07
5	Γιώργος	Τζούμας	1980-07-27

```
SELECT
    student_id, name, surname, dob
FROM
    students
WHERE
    name = 'Χρήστος' OR name = 'Οδυσσέας'
ORDER BY
    dob;
```

student_id	name	surname	dob
6	Οδυσσέας	Παππάς	1995-12-31
8	Χρήστος	Πιπεράγκας	1996-01-14

# Περιορισμός αποτελέσματος

- **LIMIT**: Περιορίζει το πλήθος των εγγραφών στον πίνακα αποτελέσματος.
- Όρος ακολουθεί τον όρο **ORDER BY**
- Το **ORDER BY** δεν απαιτείται
- Κρατάει τα πρώτα  $n$  στοιχεία

Μορφή:

```
SELECT πεδία FROM πίνακας WHERE συνθήκη  
ORDER BY πεδίο LIMIT n;
```



```
SELECT
    student_id, name, surname, dob
FROM
    students
ORDER BY
    dob DESC
LIMIT 3;
```

student_id	name	surname	dob
8	Χρήστος	Πιπεράγκας	1996-01-14
6	Οδυσσέας	Παππάς	1995-12-31
2	Πέτρος	Αγγελάτος	1991-10-10

3



Οι επόμενες εγγραφές **παραλείπονται** από τον πίνακα αποτελέσματος

# Ποιο είναι το ερώτημα;

- Το επώνυμο του νεότερου ατόμου;

```
SELECT
```

```
    surname
```

```
FROM
```

```
    students
```

```
ORDER BY
```

```
    dob DESC
```

```
LIMIT 1;
```

# Περιορισμοί

- Το **LIMIT** εφαρμόζεται και σε ερωτήματα **UPDATE** και **DELETE**
- Καλή πρακτική: **LIMIT 1** σε ερωτήματα που περιμένουμε να ενημερώσουν/διαγράψουν **μία** εγγραφή
- Θα σώσει το τομάρι σου αν ξεχάσεις ένα **WHERE**

```
DELETE FROM
    students
WHERE
    student_id = 5
LIMIT 1;
```

# Περιορισμοί

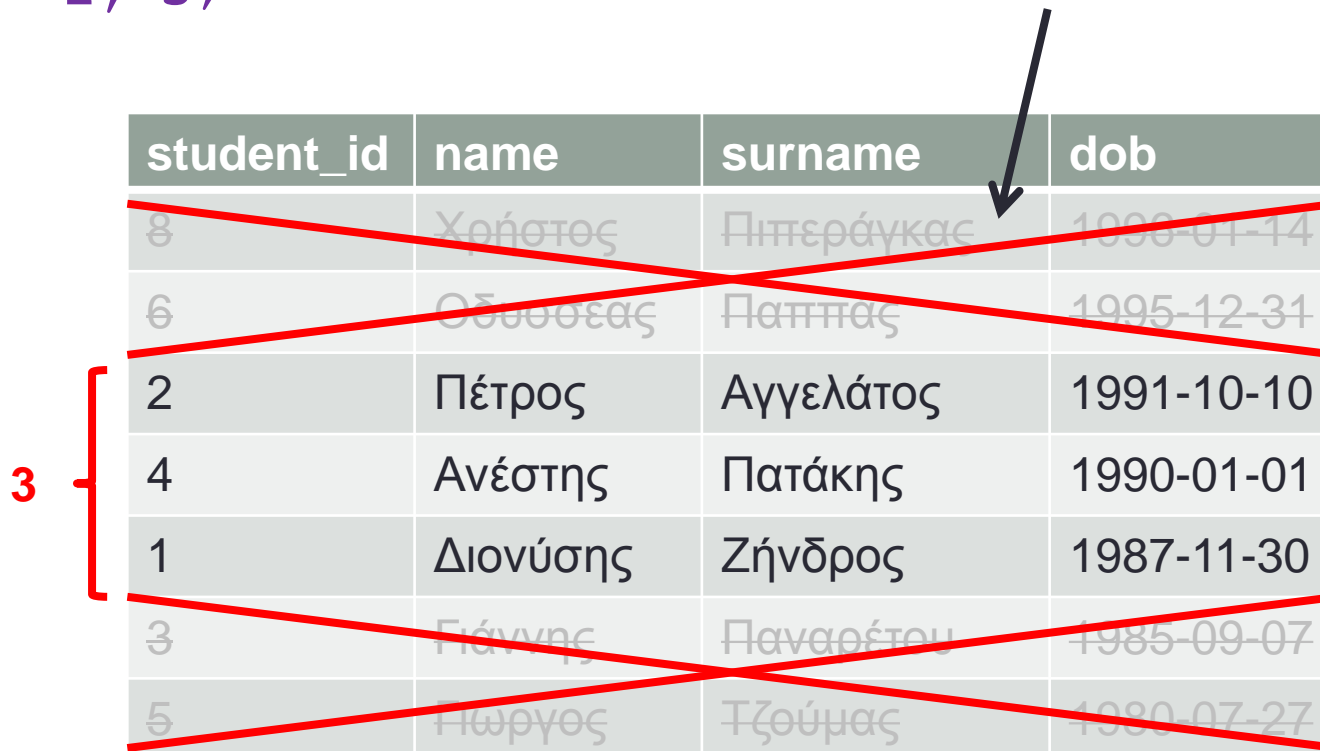
- Πλήρες LIMIT:
  - Παραλείπει εγγραφές από την αρχή του αποτελέσματος
  - Περιορίζει το πλήθος των εγγραφών
- Μορφή:

## LIMIT *offset*, N

- Παραλείπει *offset* γραμμές από την αρχή
  - $offset = 0$ : καμία παράληψη
- Κρατάει *N* γραμμές

```
SELECT
    student_id, name, surname, dob
FROM
    students
ORDER BY
    dob DESC
LIMIT
    2, 3;
```

2 εγγραφές παραλείπονται από την αρχή



student_id	name	surname	dob
8	Χρήστος	Πιπεράγκας	1998-01-14
6	Οδυσσεας	Παππας	1995-12-31
2	Πέτρος	Αγγελάτος	1991-10-10
4	Ανέστης	Πατάκης	1990-01-01
1	Διονύσης	Ζήνδρος	1987-11-30
3	Γιάννης	Παναρέτου	1985-09-07
5	Γωργος	Τζούμας	1980-07-27

# Ποιο είναι το ερώτημα;

- Το επώνυμο του *δεύτερου* νεότερου ατόμου;

```
SELECT
```

```
    surname
```

```
FROM
```

```
    students
```

```
ORDER BY
```

```
    dob DESC
```

```
LIMIT 1, 1;
```

# Αναζήτηση μέσα σε αλφαριθμητικά

- Τελεστής **LIKE**
- Μορφή:
- **αλφαριθμητικό LIKE έκφραση**
- **έκφραση:**
- Αλφαριθμητικό που περιέχει % ως χαρακτήρα-μπαλαντέρ
  - Σημαίνει «οποιοσδήποτε χαρακτήρας»

# Εκφράσεις με μπαλαντέρ

- `"Hello%world"`: Αρχίζει από "Hello" και τελειώνει σε "world"
  - Hello, world **NAI**
  - Hello, what a wonderful world **NAI**
  - Hello – the world is a great place **OXI**
- `"an%"`: Αρχίζει από "an"
  - anastasios **NAI**
  - ANASTASIOS **NAI**
  - aanastasios **OXI**



# Ποιο είναι το ερώτημα;

- Το επώνυμο ενός μαθητή που αρχίζει από “Z”;

```
SELECT
    surname
FROM
    students
WHERE
    surname LIKE 'Z%'
LIMIT 1;
```

# Τελεστής IN

- Ελέγχει αν μία τιμή **ανήκει** σε ένα σύνολο

- Μορφή:

τιμή IN ( στοιχείο1, στοιχείο2, ..., στοιχείοN )

```
SELECT
    student_id, name, surname, dob
FROM
    students
WHERE
    name IN ( 'Γιάννης', 'Γιώργος', 'Χρήστος' );
```

student_id	name	surname	dob
5	Γιώργος	Τζούμας	1980-07-27
3	Γιάννης	Παναρέτου	1985-09-07
8	Χρήστος	Πιπεράγκας	1996-01-14

Ισοδύναμα:

```
SELECT
```

```
    student_id, name, surname, dob
```

```
FROM
```

```
    students
```

```
WHERE
```

```
    name = 'Γιάννης'
```

```
    OR name = 'Γιώργος'
```

```
    OR name = 'Χρήστος' ;
```

# Σύγκριση

- Οι τελεστές  $=$ ,  $<$ ,  $>$ ,  $\leq$ ,  $\geq$  λειτουργούν σε:
  - Αριθμούς
    - $3 < 4$
    - $5 = 5$
    - $12.2 > -9$
  - Αλφαριθμητικά (λεξικογραφικά)
    - $\text{"Αγγελάτος"} \geq \text{"Ζήνδρος"}$
  - Ημερομηνίες (χρονολογικά)
    - $\text{"2001-09-11"} > \text{"1987-11-30"}$
    - $\text{"2010-11-22 00:00:00"} \leq \text{"2010-11-22 01:01:01"}$

# Τελεστής BETWEEN... AND

- Ελέγχει αν μία τιμή **βρίσκεται ανάμεσα** σε δύο φράγματα
- (μεγαλύτερο ή ίσο / μικρότερο ή ίσο)

- Μορφή:

`τιμή BETWEEN κάτω_φράγμα AND άνω_φράγμα`

```
SELECT
    student_id, name, surname, dob
FROM
    students
WHERE
    surname BETWEEN
        "Αγγελάτος" AND "Πατάκης"
```

student_id	name	surname	dob
3	Γιάννης	Παναρέτου	1985-09-07
1	Διονύσης	Ζήνδρος	1987-11-30
4	Ανέστης	Πατάκης	1990-01-01
2	Πέτρος	Αγγελάτος	1991-10-10
6	Οδυσσέας	Παππάς	1995-12-31

# Ψευδώνυμα στηλών

- Αλλάζουν το όνομα μίας στήλης του αποθηκευμένου πίνακα στον πίνακα αποτελέσματος
- Εισάγονται με το **AS**
- Μορφή: **στήλη AS ψευδώνυμο**



```
SELECT
    student_id AS am,
    name, surname,
    dob AS date_of_birth
FROM
    students
LIMIT 5;
```

am	name	surname	date_of_birth
3	Γιάννης	Παναρέτου	1985-09-07
1	Διονύσης	Ζήνδρος	1987-11-30
4	Ανέστης	Πατάκης	1990-01-01
2	Πέτρος	Αγγελάτος	1991-10-10
6	Οδυσσέας	Παππάς	1995-12-31

# NULL

- Ειδική τιμή
- Αντιπροσωπεύει την απουσία τιμής
- Ένα πεδίο **μπορεί** να έχει την δυνατότητα να αποθηκεύσει **NULL**;
  - Και ναι και όχι
  - Το καθορίζουμε στο σχήμα
- Έλεγχος για **NULL** με τον τελεστή **IS NULL**

# Πράξεις κατά την επιλογή

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

```
SELECT
    (student_id + 100) AS points,
    name, surname, dob
FROM
    students
LIMIT 3;
```

points	name	surname	dob
108	Χρήστος	Πιπεράγκας	1996-01-14
106	Οδυσσέας	Παππάς	1995-12-31
102	Πέτρος	Αγγελάτος	1991-10-10

# Κωδικοποίηση

- Η κωδικοποίηση στην MySQL μπορεί να γίνει πονοκέφαλος
  - Επίπεδα κωδικοποίησης:
    - Βάση δεδομένων
    - Πίνακας
    - Πεδίο
    - Σύνδεση
    - Καθεαυτά δεδομένα
- Collation:**  
Meta-ιδιότητα που «θυμάται» τι είδους κωδικοποίηση έχουμε αποθηκεύσει

# Κωδικοποίηση

- Φροντίζουμε:
  - Τα **δεδομένα** που στέλνουμε στην MySQL να είναι **UTF-8**
    - Δηλαδή το **αλφαριθμητικό του ερωτήματος**
  - Το **σχήμα** να έχει collation **utf8\_unicode\_ci**
  - Η **σύνδεση** να έχει κωδικοποίηση UTF-8
    - Με το ερώτημα: **SET NAMES utf8;**
    - Τρέχει πριν από όλα τα άλλα

# Τελεστές σε ημερομηνίες

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

# 5<sup>η</sup> Εργασία

- Βελτίωση της 4<sup>ης</sup> εργασίας
- Παράδοση μέσω SSH (Secure SHell)
  - Ανεβάστε .php, .html, .css, εικόνες κλπ. στο home σας
  - Ανεβάστε ένα Export του σχήματός σας σε μορφή .sql
  - Μέχρι **6/12/2010**
- **Να βελτιώσετε το file uploader σας**
  - Απλό σύστημα χρηστών
  - Register – Login – Logout
  - Βάση δεδομένων
    - Αποθήκευση χρηστών
      - Όνομα & κωδικός πρόσβασης
    - Αποθήκευση meta-πληροφοριών αρχείων
      - Ημερομηνία αποστολής
      - Χρήστης
- Θα χρειαστεί να εγκαταστήσετε MySQL + phpMyAdmin

```
SELECT
    name, surname,
    dob + INTERVAL 10 YEAR
FROM
    students
LIMIT 3;
```

name	surname	dob
Χρήστος	Πιπεράγκας	2006-01-14
Οδυσσέας	Παππάς	2005-12-31
Πέτρος	Αγγελάτος	2001-10-10



# Τελεστές σε ημερομηνίες

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

**NOW ()** : “2010-11-22 17:49:00”

# Online now

- Πώς θα δείξω ποιοι χρήστες είναι online;
- 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();
```

# Ένωση πινάκων

- Συνδυασμός δεδομένων από πολλούς αποθηκευμένους πίνακες
- Ο πίνακας αποτελέσματος έχει στοιχεία από όλους

# Καρτεσιανό γινόμενο

$$A \times B$$

κάθε εγγραφή του πίνακα A  
συνδυάζεται με  
κάθε εγγραφή του πίνακα B

# Καρτεσιανό γινόμενο

student_id	firstname	surname	school
5	Γιώργος	Τζούμας	1
3	Γιάννης	Παναρέτου	1
8	Χρήστος	Πιπεράγκας	3

**people**

**x**

school_id	schoolname
1	ΕΜΠ
2	ΑΠΘ
3	ΑΣΟΕΕ

**schools**

# Καρτεσιανό γινόμενο

=

student_id	firstname	surname	school	school_id	schoolname
5	Γιώργος	Τζούμας	1	1	ΕΜΠ
5	Γιώργος	Τζούμας	1	2	ΑΠΘ
5	Γιώργος	Τζούμας	1	3	ΑΣΟΕΕ
3	Γιάννης	Παναρέτου	1	1	ΕΜΠ
3	Γιάννης	Παναρέτου	1	2	ΑΠΘ
3	Γιάννης	Παναρέτου	1	3	ΑΣΟΕΕ
8	Χρήστος	Πιπεράγκας	3	1	ΕΜΠ
8	Χρήστος	Πιπεράγκας	3	2	ΑΠΘ
8	Χρήστος	Πιπεράγκας	3	3	ΑΣΟΕΕ

# Καρτεσιανό γινόμενο

- Στον όρο **FROM** γράφουμε τους δύο πίνακες χωρισμένους με κόμματα:

```
SELECT  
    surname, schoolname  
FROM  
    students, schools;
```

<b>surname</b>	<b>schoolname</b>
Τζούμας	ΕΜΠ
Τζούμας	ΑΠΘ
Τζούμας	ΑΣΟΕΕ
Παναρέτου	ΕΜΠ
Παναρέτου	ΑΠΘ
Παναρέτου	ΑΣΟΕΕ
Πιπεράγκας	ΕΜΠ
Πιπεράγκας	ΑΠΘ
Πιπεράγκας	ΑΣΟΕΕ

# Καρτεσιανό γινόμενο

- Από μόνο του **άχρηστο**
- Συνδυασμένο με έναν **όρο WHERE** χρήσιμο

```
SELECT
    surname, schoolname
FROM
    students, schools
WHERE
    schoolid = school;
```

surname	schoolname
Τζούμας	ΕΜΠ
Παναρέτου	ΕΜΠ
Πιπεράγκας	ΑΣΟΕΕ



student_id	firstname	surname	school	school_id	schoolname
5	Γιώργος	Τζούμας	1	1	ΕΜΠ
5	Γιώργος	Τζούμας	1	2	ΑΠΘ
5	Γιώργος	Τζούμας	1	3	ΑΣΟΕΕ
3	Γιάννης	Παναρέτου	1	1	ΕΜΠ
3	Γιάννης	Παναρέτου	1	2	ΑΠΘ
3	Γιάννης	Παναρέτου	1	3	ΑΣΟΕΕ
8	Χρήστος	Πιπεράγκας	3	1	ΕΜΠ
8	Χρήστος	Πιπεράγκας	3	2	ΑΠΘ
8	Χρήστος	Πιπεράγκας	3	3	ΑΣΟΕΕ

# Αναφορά σε στήλες πίνακα

- Δύο στήλες μπορεί να έχουν το ίδιο όνομα
- Αναφερόμαστε σε συγκεκριμένη πίνακα με την σύνταξη:

πίνακας.στήλη

student_id	name	surname	school_id
5	Γιώργος	Τζούμας	1
3	Γιάννης	Παναρέτου	1
8	Χρήστος	Πιπεράγκας	3

school_id	name
1	ΕΜΠ
2	ΑΠΘ
3	ΑΣΟΕΕ

```
SELECT
```

```
    students.name AS firstname,  
    surname,
```

```
    schools.name AS schoolname
```

```
FROM
```

```
    students, schools
```

```
WHERE
```

```
    schools.school_id = students.school_id;
```

# Εναλλακτική σύνταξη

Αντί για:

```
FROM
```

```
    A, B
```

```
WHERE A.x = B.y
```

Είναι πιο ευανάγνωστο να γράφουμε:

```
A CROSS JOIN B on A.x = B.y
```

Έτσι το **WHERE** δεσμεύεται για φιλτράρισμα

```
SELECT
```

```
    students.name AS firstname,  
    surname,
```

```
    schools.name AS schoolname
```

```
FROM
```

```
    students CROSS JOIN schools ON
```

```
    schools.school_id = student.school_id;
```

# Ποιο είναι το ερώτημα;

- Το σχολείο με τον μαθητή που έχει την μεγαλύτερη ηλικία;

```
SELECT
    schools.name
FROM
    schools CROSS JOIN students ON
    school.school_id = students.school_id
ORDER BY
    dob
LIMIT 1;
```

student_id	name	surname	school_id	school_id	name
5	Γιώργος	Τζούμας	1	1	ΕΜΠ
3	Γιάννης	Παναρέτου	1	2	ΑΠΘ
8	Χρήστος	Πιπεράγκας	3	3	ΑΣΟΕΕ

Ονόματα όλων των μαθητών και το σχολείο τους;

```
SELECT
    students.name AS firstname,
    school.name AS schoolname
FROM
    students
    CROSS JOIN schools
ON
    students.school_id = schools.school_id;
```

firstname	schoolname
Γιώργος	ΕΜΠ
Γιάννης	ΕΜΠ
Χρήστος	ΑΣΟΕΕ



# Αριστερή ένωση

- Τι γίνεται αν δεν έχουμε πληροφορίες για ένα σχολείο;

Καρτεσιανό γινόμενο:

student_id	name	surname	school_id	school_id	name
5	Γιώργος	Τζούμας	1	1	ΕΜΠ
5	Γιώργος	Τζούμας	1	2	ΑΠΘ
3	Γιάννης	Παναρέτου	1	1	ΕΜΠ
3	Γιάννης	Παναρέτου	1	2	ΑΠΘ
8	Χρήστος	Πιπεράγκας	3	1	ΕΜΠ
8	Χρήστος	Πιπεράγκας	3	2	ΑΠΘ

- Πίνακας αποτελέσματος:

firstname	schoolname
Γιώργος	ΕΜΠ
Γιάννης	ΕΜΠ

- Ο μαθητής λείπει **μόνο και μόνο επειδή λείπει το σχολείο του!**
- Η λογική πρόταση είναι ψευδής με όποια γραμμή και να κάνουμε συνδυασμό

Όμως θέλουμε τις πληροφορίες του **ακόμη** και αν δεν βρεθεί το σχολείο

# Αριστερή ένωση

- Αλλάζουμε το CROSS JOIN σε LEFT JOIN
- Αν **ταιριάζει** μία ή περισσότερες εγγραφές:
  - κάνει το **ίδιο** με την απλή ένωση
- Αν **καμία** 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

student_id	name	surname	school_id	school_id	name
5	Γιώργος	Τζούμας	1	1	ΕΜΠ
5	Γιώργος	Τζούμας	1	2	ΑΠΘ
3	Γιάννης	Παναρέτου	1	1	ΕΜΠ
3	Γιάννης	Παναρέτου	1	2	ΑΠΘ
8	Χρήστος	Πιπεράγκας	3	1	ΕΜΠ
8	Χρήστος	Πιπεράγκας	3	2	ΑΠΘ

Καμία εγγραφή του **δεξιού** πίνακα δεν ικανοποιεί την σχέση

**students.school\_id = schools.school\_id**

για τον μαθητή Χρήστο!

```
SELECT
```

```
    students.name AS firstname,  
    school.name AS schoolname
```

```
FROM
```

```
    students
```

```
        LEFT JOIN schools
```

```
ON
```

```
students.school_id = schools.school_id;
```

firstname	schoolname
Γιώργος	ΕΜΠ
Γιάννης	ΕΜΠ
Χρήστος	NULL

# Πολλαπλές ενώσεις

- Ενώνουμε το *αποτέλεσμα* της ένωσης με άλλο πίνακα
- Έχουμε ένωση τριών ή περισσότερων πινάκων

# Παράδειγμα σχήματος

users:

- userid
- username
- password
- email

blogs:

- blogid
- title
- text
- userid

comments:

- commentid
- text
- blogid
- userid
- created

# Ποιο είναι το ερώτημα;

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

```
SELECT
    text, title
FROM
    blogs
WHERE
    userid = 5;
```



# Ποιο είναι το ερώτημα;

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

```
SELECT
    comment.text
FROM
    comments CROSS JOIN blogs
    ON comments.blogid = blogs.blogid
WHERE
    blogs.userid = 5;
```

# Ποιο είναι το ερώτημα;

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

```
SELECT
    comment.text, users.name
FROM
    comments CROSS JOIN blogs
        ON comments.blogid = blogs.blogid
    CROSS JOIN users
        ON users.userid = comment.userid
WHERE
    blogs.userid = 5;
```

# Ποιο είναι το ερώτημα;

- 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
    comments CROSS JOIN blogs
    ON comments.blogid = blogs.blogid
ORDER BY
    comments.created DESC
LIMIT 1
```

# Μάθαμε

- Ταξινόμηση
- Περιορισμοί
- Παράδειγμα δημιουργίας σχήματος
- Τελεστές LIKE, IN, BETWEEN
- Κωδικοποίηση
- Ψευδώνυμα
- Ημερομηνίες
- Απλή ένωση
- Αριστερή ένωση

# Αξιολόγηση

- Πείτε μας τη γνώμη σας 😊

# Συγχαρητήρια!

- Μάθατε MySQL.
- Μπορείτε να χειριστείτε αποθηκευμένα δεδομένα!



# Την επόμενη φορά...

- Προχωρημένα θέματα PHP:
  - Εμβάθυνση με **πολλά** παραδείγματα
  - Λίγη θεωρία, περισσότερη πράξη