

Ερωτήματα σε συνδεδεμένου πίνακες

Μέχρι στιγμής όλα τα ερωτήματα ήταν απλά, υπό την έννοια ότι αφορούσαν μόνο έναν πίνακα. Πολλές φορές όμως θέλουμε να πάρουμε δεδομένα από περισσότερους πίνακες. Κάτι τέτοιο έχει συνήθως νόημα όταν οι πίνακες έχουν κάποια συσχέτιση-σύνδεση, όπως τις περιγράψαμε στη πρώτη ενότητα της δημιουργίας πινάκων.

Στη δικιά μας βάση έχουμε τις εξής συσχετίσεις

Books – author (Σχέση πολλά προς ένα - N:1)

Books – publication (Σχέση πολλά προς ένα - N:1)

Books – member (Σχέση πολλά προς Πολλά - N:M)

Ας υποθέσουμε ότι θέλουμε να φτιάξουμε ένα ερώτημα για να πάρουμε όλα τα βιβλία που μπήκαν στη βάση πριν από το 2000, αλλά επίσης θέλουμε να βλέπουμε το όνομα του συγγραφέα και όχι τον κωδικό του. Τα στοιχεία των βιβλίων που μας ενδιαφέρουν είναι στο πίνακα **books**, ενώ τα ονόματα των συγγραφέων είναι στον πίνακα **author**. Άρα πρέπει να τραβήξουμε ταυτόχρονα δεδομένα και από τους δύο πίνακες. Ας δοκιμάσουμε το παρακάτω ερώτημα

```
SELECT ID, title, ISBN, name FROM books, author WHERE date_added < '2000/1/1'
```

ID	title	ISBN	name
101	Χίμαιρα	11-222-33	Διδώ Σωτηρίου
101	Χίμαιρα	11-222-33	Νίκος Καββαδίας
101	Χίμαιρα	11-222-33	M. Καραγάτσης
101	Χίμαιρα	11-222-33	Αλέξανδρος Παπαδιαμάντης
102	Το Δέκα	11-223-15	Διδώ Σωτηρίου
102	Το Δέκα	11-223-15	Νίκος Καββαδίας
102	Το Δέκα	11-223-15	M. Καραγάτσης
102	Το Δέκα	11-223-15	Αλέξανδρος Παπαδιαμάντης
103	Ηλέκτρα	11-223-28	Διδώ Σωτηρίου
103	Ηλέκτρα	11-223-28	Νίκος Καββαδίας
103	Ηλέκτρα	11-223-28	M. Καραγάτσης
103	Ηλέκτρα	11-223-28	Αλέξανδρος Παπαδιαμάντης
104	Εντολή	11-226-45	Διδώ Σωτηρίου
104	Εντολή	11-226-45	Νίκος Καββαδίας
104	Εντολή	11-226-45	M. Καραγάτσης
104	Εντολή	11-226-45	Αλέξανδρος Παπαδιαμάντης

Αν μελετήσουμε τα αποτελέσματα θα δούμε ότι κάθε βιβλίο εμφανίζεται 4 φορές, με διαφορετικό όνομα συγγραφέα δίπλα του κάθε φορά. Αυτό προφανώς είναι λάθος. Το ερώτημα που φτιάξαμε κάνει όλους τους πιθανούς συνδυασμούς βιβλίων-συγγραφέων χωρίς απαραίτητα αυτοί οι συνδυασμοί να έχουν νόημα. Το πρόβλημα είναι ότι δεν βάλαμε

κάποιο περιορισμό ώστε τα ζευγάρια βιβλίο-συγγραφέας που θα επιστραφούν να έχουν το ίδιο ID συγγραφέα. Έτσι το ερώτημα μπορεί να διορθωθεί ως εξής

```
SELECT ID, title, ISBN, name FROM books, author
WHERE date_added < '2000/1/1' AND author=ID_auth
```

ID	title	ISBN	name
103	Ηλέκτρα	11-223-28	Διδώ Σωτηρίου
104	Εντολή	11-226-45	Διδώ Σωτηρίου
101	Χίμαιρα	11-222-33	M. Καραγάτσης
102	Το Δέκα	11-223-15	M. Καραγάτσης

Τα αποτελέσματα από αυτή την εντολή έχουν ποια τη σωστή αντιστοίχιση. **Αυτό που ουσιαστικά κάναμε είναι να τονίσουμε με ποιο πεδίο συνδέονται οι δυο πίνακες.**

Πολλές φορές τα ονόματα των πεδίων που εμφανίζονται στο ερώτημα είναι ίδια, γιατί προέρχονται από διαφορετικούς πίνακες. Σε μια τέτοια περίπτωση πρέπει να προσδιορίσουμε από ποιον πίνακα προέρχεται το κάθε πεδίο. Μ' αυτό τον τρόπο τα ερώτημα γίνεται και πιο κατανοητό.

Ας υποθέσουμε ότι στο προηγούμενο ερώτημα θέλουμε να εμφανίσουμε και τον εκδοτικό οίκο, αλλά όχι μόνο σαν κωδικό, αλλά το όνομά του. Το όνομα του εκδοτικού οίκου είναι αποθηκευμένο στο πίνακα **publication** στο πεδίο **name**. Κάποιος θα μπορούσε να απαντήσει αμέσως ότι θα συμπεριλάβουμε τον πίνακα **publication** στο ερώτημα, ως εξής

```
SELECT ID, title, ISBN, name, name FROM books, author, publication
WHERE date_added < '2000/1/1' AND author=ID_auth AND publication=ID_publ
```

Η απάντηση που θα πάρουμε είναι ότι «**το πεδίο name είναι ασαφές**» διότι δεν ξεκαθαρίζουμε από ποιον πίνακα προέρχεται. Έτσι πρέπει να διευκρινίσουμε από ποιον πίνακα είναι το κάθε ένα πεδίο **name** που υπάρχει στο ερώτημα. Το ερώτημα μετατρέπεται σε

```
SELECT ID, title, ISBN, author.name, publication.name
FROM books, author, publication
WHERE date_added < '2000/1/1' AND author=ID_auth AND publication=ID_publ
```

Το ερώτημα πλέον είναι αποδεκτό και τα αποτελέσματά του είναι τα εξής:

ID	title	ISBN	name	name
103	Ηλέκτρα	11-223-28	Διδώ Σωτηρίου	Κέδρος
104	Εντολή	11-226-45	Διδώ Σωτηρίου	Κέδρος
101	Χίμαιρα	11-222-33	M. Καραγάτσης	Κέδρος
102	Το Δέκα	11-223-15	M. Καραγάτσης	Κέδρος

Παρατηρήστε το τίτλο της κάθε στήλης στα αποτελέσματα. Και τα δυο ονομάζονται **name**, κάτι που δεν κάνει σαφή τα αποτελέσματα και μπορεί να δημιουργήσει προβλήματα. Εδώ μπορούμε να χρησιμοποιήσουμε τη δήλωση «**AS**» για να μετονομάσουμε τα πεδία αυτά.

```
SELECT ID, title, ISBN, author.name AS author, publication.name AS publication
FROM books, author, publication
WHERE date_added < '2000/1/1' AND author=ID_auth AND publication=ID_publ
```

ID	title	ISBN	author	publication
103	Ηλέκτρα	11-223-28	Διδώ Σωτηρίου	Κέδρος
104	Εντολή	11-226-45	Διδώ Σωτηρίου	Κέδρος
101	Χίμαιρα	11-222-33	M. Καραγάτσης	Κέδρος
102	Το Δέκα	11-223-15	M. Καραγάτσης	Κέδρος

Όταν τα πεδία με κοινό όνομα είναι πολλά, τότε το να γράφουμε ολόκληρο το όνομα του πίνακα μπροστά από κάθε πεδίο είναι κουραστικό και κάνει το ερώτημα μεγάλο και δυσανάγνωστο. Σε αυτή τη περίπτωση μπορούμε να δώσουμε ένα σύντομο όνομα σε κάθε πίνακα και να το χρησιμοποιούμε ως πρόθεση στα πεδία. Έτσι αντί για «**publication.name**» θα μπορούσαμε να γράφουμε «**p.name**». Το ερώτημα γίνεται ως εξής:

```
SELECT ID, title, ISBN, a.name AS author, p.name AS publication
FROM books, author a, publication p
WHERE date_added < '2000/1/1' AND author=ID_auth AND publication=ID_publ
```

Στη δήλωση των πινάκων βλέπουμε δίπλα από κάποιους πίνακες μια συντομογραφία (δικής μας επιλογής), που μπορούμε να χρησιμοποιήσουμε μπροστά από τα ονόματα των πεδίων για να προσδιορίσουμε το πίνακα από τον οποίο προέρχονται. Αυτή η συντομογραφία μπορεί να χρησιμοποιηθεί και στην **WHERE** αν αυτό κρίνεται απαραίτητο, δηλαδή αν εμφανίζονται πεδία με το ίδιο όνομα από διαφορετικούς πίνακες.

Εμφωλευμένες SELECT

Στα προηγούμενα παραδείγματα συνδυάσαμε τους πίνακες **books**, **author** και **publication** για να πάρουμε δεδομένα και από τους τρεις ταυτόχρονα. Συγκεκριμένα από τους πίνακες **author** και **publication** θέλαμε να πάρουμε ένα μόνο πεδίο. Σε αυτές τις περιπτώσεις, που μας ενδιαφέρει ένα μόνο πεδίο από έναν δευτερεύοντα πίνακα, μπορούμε να το πάρουμε εφαρμόζοντας ένα ερώτημα SELECT μέσα στην ίδια τη SELECT. Αυτές οι SELECT ονομάζονται **εμφωλευμένες**.

Για παράδειγμα το ίδιο ερώτημα με πριν (όλα τα βιβλία που εκδόθηκαν πριν από το 2000 με το όνομα του συγγραφέα και του εκδοτικού οίκου) θα γινόταν ως εξής:

```
SELECT ID, title, ISBN,  
(SELECT name FROM author WHERE author=ID_auth) AS author,  
(SELECT name FROM publication WHERE publication =ID_publ) AS publication  
FROM books  
WHERE date_added < '2000/1/1'
```

Τα αποτελέσματα του ερωτήματος είναι ακριβώς τα ίδια με του προηγούμενου ερωτήματος. Οι **εμφωλευμένες SELECT** ακολουθούνται από τη δήλωση **AS**, αλλιώς τα ίδια τα ερωτήματα θα παρουσιάζονται ως τίτλοι στις στήλες των αποτελεσμάτων.

Σύνολα τιμών και εμφωλευμένες ερωτήματα

Οι **εμφωλευμένες SELECT** μπορούν να χρησιμοποιηθούν και στη δήλωση **WHERE**. Ας θεωρήσουμε το εξής ερώτημα. **Θέλουμε να βρούμε όλα τα βιβλία που ανήκουν σε εκδοτικούς οίκους με τουλάχιστον 3 βιβλία.** Αν κάποιος γνωρίζει τα δεδομένα της βάσης θα μπορούσε να πει ότι αυτοί οι εκδοτικοί οίκοι είναι οι 2 και 4 και να φτιάξει απευθείας το ερώτημα:

```
SELECT *  
FROM books  
WHERE publication = 2 OR publication = 4
```

Αυτό το ερώτημα δεν έχει σχέση με τη σύνδεση πινάκων. Παρατηρούμε όμως ότι το ζητούμενο είναι να βρούμε τα βιβλία που το **publication** είναι 2 ή 4. Κατά τον ίδιο τρόπο θα μπορούσε το **publication** να έχει περισσότερες πιθανές τιμές. Για παράδειγμα αν θέλαμε το **publication** να είναι 2 ή 4 ή 6 ή 7 ή 10 θα έπρεπε στο **WHERE** να γράψουμε

```
SELECT *  
FROM books  
WHERE publication = 2 OR publication = 4 OR publication = 6 OR publication = 7 OR  
publication = 10
```

Μια τέτοια γραφή κάνει το ερώτημα μεγάλο και δυσανάγνωστο. Αντί γι' αυτό θα μπορούσαμε να χρησιμοποιήσουμε τη δήλωση **IN** που χρησιμοποιείται για τον ορισμό συνόλων. Με την **IN** δηλώνουμε απευθείας ποιές είναι οι πιθανές τιμές του πεδίου ως εξής:

```
SELECT *
FROM books
WHERE publication IN (2,4,6,7,10)
```

Η χρήση της **IN** είναι αναπόφευκτη όταν το σύνολο τιμών, που θέλουμε να ανήκει το πεδίο, δεν μπορεί να προσδιοριστεί. Για παράδειγμα στο προηγούμενο ζητούμενο (**να βρούμε όλα τα βιβλία που ανήκουν σε εκδοτικούς οίκους με τουλάχιστον 3 βιβλία**) δεν θα μπορούσαμε να γνωρίζουμε ποιοι είναι οι εκδοτικοί οίκοι αν η βάση είχε χιλιάδες βιβλία και εκατοντάδες εκδοτικούς οίκους. Σε αυτή τη περίπτωση θα ορίζαμε το σύνολο των εκδοτικών οίκων με την **IN** και ένα εμφωλευμένο ερώτημα. Το ερώτημα για την εύρεση των εκδοτικών οίκων θα ήταν:

```
SELECT ID_publ FROM publication
WHERE (SELECT count(*) FROM books where publication=ID_publ) >= 3
```

Ένα τέτοιο ερώτημα επιστρέφει ένα σύνολο από ID εκδοτικών οίκων που πληρούν τις προδιαγραφές μας, δηλαδή έχουν εκδώσει τουλάχιστον 3 βιβλία.

	ID_publ
<input type="checkbox"/> Edit Copy Delete	2
<input type="checkbox"/> Edit Copy Delete	4

Οπότε το τελικό ερώτημα θα προέκυπτε βάζοντας το προηγούμενο ερώτημα στη δήλωση **IN** ως εξής:

```
SELECT *
FROM books
WHERE publication IN (
    SELECT ID_publ FROM publication
    WHERE (
        SELECT count(*) FROM books WHERE publication=ID_publ
    ) >= 3
)
```

Παρατηρούμε ότι στο τελικό ερώτημα έχουμε τρεις **SELECT** τη μια μέσα στην άλλη διαδοχικά. Η κάθε μια επιστρέφει ένα αποτέλεσμα που χρησιμοποιείται από τη γονική της. Η χρήση της **IN** και η χρήση **εμφωλευμένων SELECT** θεωρείται ότι **οδηγούν σε βαριά και**

αργά ερωτήματα όταν η βάση είναι μεγάλη, γι' αυτό πρέπει να χρησιμοποιούνται μόνο όταν είναι απαραίτητο.

Σύνδεση πινάκων με την JOIN

Μέχρι στιγμής είδαμε δυο τρόπους για να αξιοποιήσουμε τη σύνδεση των πινάκων. Ο ένας ήταν με τη προσθήκη μιας επιπλέον συνθήκης στο τμήμα **WHERE** που δηλώνει ποια είναι τα πεδία στα οποία συνδέονται δυο πίνακες **(a)** και ο άλλος είναι η χρήση εμφωλευμένων **SELECT (b)**.

- a) **SELECT** ID, title, ISBN, name **AS** author **FROM** books, author
WHERE date_added < '2000/1/1' **AND** author=ID_auth
- b) **SELECT** ID, title, ISBN,
(SELECT name **FROM** author **WHERE** author=ID_auth) **AS** author
FROM books **WHERE** date_added < '2000/1/1'

Τα δύο ερωτήματα **a** και **b** μπορεί να φαίνονται ισοδύναμα αλλά δεν είναι. Αν θεωρήσουμε ότι (για κάποιο λόγο που δεν μας ενδιαφέρει) το βιβλίο με ID 104 («Εντολή»), αλλάζει και στο πεδίο ID_auth παίρνει τη τιμή 0. Αυτό σημαίνει ότι δεν είναι πια συνδεδεμένο με κάποιον συγγραφέα. Σε αυτή τη περίπτωση αν εκτελέσουμε τα προηγούμενα ερωτήματα θα πάρουμε τα εξής αποτελέσματα, όπου στο ένα το βιβλίο 104 δεν υπάρχει:

ID	title	ISBN	author
103	Ηλέκτρα	11-223-28	Διδώ Σωτηρίου
101	Χίμαιρα	11-222-33	M. Καραγάτσης
102	Το Δέκα	11-223-15	M. Καραγάτσης
104	Εντολή	11-226-45	NULL

(a)

(b)

Αυτή η διαφορά οφείλεται στο γεγονός ότι το ερώτημα **(a)** ψάχνει τα ζευγάρια από τους πίνακες **books** και **author** όπου το ID του συγγραφέα ταυτίζεται. Για το βιβλίο **104** δε βρέθηκε αντιστοίχιση στο πίνακα των συγγραφέων και γι' αυτό δεν συμπεριλαμβάνεται στα αποτελέσματα.

Αντίθετα στο ερώτημα **(b)** ψάχνουμε όλα τα βιβλία που εκδόθηκαν πριν το 2000 και για κάθε ένα από αυτά «τραβάμε» το όνομα από τον πίνακα **author**. Το βιβλίο **104** ικανοποίησε τη συνθήκη της ημερομηνίας αλλά το εμφωλευμένο ερώτημα για τη λήψη του ονόματος απέτυχε να βρει τον συγγραφέα με ID μηδέν. Έτσι το βιβλίο συμπεριλαμβάνεται στα αποτελέσματα αλλά με κενό (**null**) το όνομα του συγγραφέα. Η ύπαρξη δηλαδή αντίστοιχου συγγραφέα δεν ήταν προαπαιτούμενο στο ερώτημα (a), όπως ήταν στο ερώτημα (a).

Συμπεραίνουμε λοιπόν ότι οι δυο μέθοδοι για τον συνδυασμό συνδεδεμένων πινάκων διαφέρουν και μπορούν να χρησιμοποιηθούν σε διαφορετικές περιπτώσεις. Επιπλέον η χρήση εμφωλευμένων ερωτημάτων κάνει τα ερωτήματα «βαριά» και μπορεί να χρησιμοποιηθεί μόνο όταν θέλουμε να «τραβήξουμε» ένα πεδίο από συνδεδεμένο πίνακα. Γι' αυτούς τους λόγους έχει ορισθεί και τυποποιηθεί μια ακόμη μέθοδος για την άντληση δεδομένων από συνδεδεμένους πίνακες, η δήλωση **JOIN** (<http://dev.mysql.com/doc/refman/5.7/en/join.html>).

Ας θεωρήσουμε σε αυτό το σημείο ότι προσθέτουμε έναν ακόμα συγγραφέα για τον οποίο δεν υπάρχει καταχωρημένο βιβλίο.

```
INSERT INTO `author` SET `name`='Νίκος Καζαντζάκης'
```

Ο πίνακας author θα έχει στο εξής τα παρακάτω στοιχεία. Άρα αυτή τη στιγμή έχουμε στη βάση ένα βιβλίο χωρίς συγγραφέα (το 104) και έναν συγγραφέα χωρίς βιβλίο (τον 5).

	ID_auth	name
<input type="checkbox"/> Edit Copy Delete	1	Διδώ Σωτηρίου
<input type="checkbox"/> Edit Copy Delete	2	Νίκος Καββαδίας
<input type="checkbox"/> Edit Copy Delete	3	M. Καραγάσης
<input type="checkbox"/> Edit Copy Delete	4	Αλέξανδρος Παπαδιαμάντης
<input type="checkbox"/> Edit Copy Delete	5	Νίκος Καζαντζάκης

Με τη δήλωση **JOIN** μπορούμε αν ορίσουμε πως ακριβώς θέλουμε να συνδυαστούν οι εγγραφές των συνδεδεμένων πινάκων. Για τη χρήση της **JOIN** πρέπει να ορίσουμε το τύπο της ένωσης και το πεδίο σύνδεσης των πινάκων. Αν για παράδειγμα θέλουμε όλα τα βιβλία μαζί με τα αντίστοιχα στοιχεία του συγγραφέα θα γράφαμε το ερώτημα

```
SELECT * FROM books JOIN author ON author=ID_auth
```

Μετά τη δήλωση **JOIN** γράφουμε το όνομα του δεύτερου πίνακα και ύστερα τη δήλωση **ON** στην οποία θα προσδιορίσουμε με ποια πεδία των δυο πινάκων θα υλοποιηθεί η ένωση.

Η ένωση με το απλό **JOIN** ισοδυναμεί με τη ένωση τύπου **INNER JOIN**, που σημαίνει ότι θα επιστραφούν όλες τα ζευγάρια εγγραφών από τους δυο πίνακες όπου το πεδίο του συγγραφέα ταυτίζεται.

ID	ISBN	title	author	publication	price	status	date_added	ID_auth	name
103	11-223-28	Ηλέκτρα	1	2	22.00	1	1997-04-02	1	Διδώ Σωτηρίου
105	11-223-36	Μαραμπού	2	1	11.40	1	2001-11-02	2	Νίκος Καββαδίας
110	11-222-85	Θα σας τηλεφωνώ	2	4	17.30	0	2015-03-29	2	Νίκος Καββαδίας
101	11-222-33	Χίμαιρα	3	4	15.42	1	1995-05-15	3	M. Καραγάσης
102	11-223-15	Το Δέκα	3	4	18.50	1	1996-03-28	3	M. Καραγάσης
106	11-245-35	Όνειρο στο κύμα	4	2	21.60	0	2008-06-15	4	Αλέξανδρος Παπαδιαμάντης
107	11-225-38	Η φόνισσα	4	3	15.80	1	2009-07-30	4	Αλέξανδρος Παπαδιαμάντης
108	11-224-98	Η γυφτοπούλα	4	4	9.90	1	2012-01-28	4	Αλέξανδρος Παπαδιαμάντης
109	11-229-78	Η μετανάστρια	4	3	23.10	1	2014-02-16	4	Αλέξανδρος Παπαδιαμάντης

Στη προηγούμενη εικόνα φαίνονται τα αποτελέσματα του ερωτήματος. Προσέξτε ότι το βιβλίο 104 και ο συγγραφέας 5 απουσιάζουν τελείως από τα αποτελέσματα, αφού δεν υπάρχει ζευγάρι γι' αυτά.

Έστω ότι θέλουμε να βρούμε όλα τα βιβλία και τους συγγραφείς τους, ανεξάρτητα από το αν υπάρχει ή όχι συγγραφέας. Σε αυτή τη περίπτωση θέλουμε να δώσουμε βαρύτητα στον πίνακα **books**, ο οποίος βρίσκεται στα αριστερά της δήλωσης **JOIN**. Έτσι προκύπτει η ένωση **LEFT JOIN**, με την οποία θα πάρουμε στα αποτελέσματα και το βιβλίο 104. Το ερώτημα και τα αποτελέσματά του λοιπόν φαίνονται παρακάτω. Προσέξτε στη τελευταία γραμμή ότι το βιβλίο 104 συμπεριλαμβάνεται στα αποτελέσματα παρά το γεγονός ότι δεν έχει συγγραφέα.

```
SELECT * FROM books LEFT JOIN author ON author=ID_auth
```

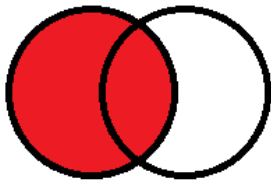
ID	ISBN	title	author	publication	price	status	date_added	ID_auth	name
103	11-223-28	Ηλέκτρα	1	2	22.00	1	1997-04-02	1	Διδώ Σωτηρίου
105	11-223-36	Μαραμπού	2	1	11.40	1	2001-11-02	2	Νίκος Καββαδίας
110	11-222-85	Θα σας τηλεφωνώ	2	4	17.30	0	2015-03-29	2	Νίκος Καββαδίας
101	11-222-33	Χίμαιρα	3	4	15.42	1	1995-05-15	3	Μ. Καραγάτσης
102	11-223-15	Το Δέκα	3	4	18.50	1	1996-03-28	3	Μ. Καραγάτσης
106	11-245-35	Όνειρο στο κύμα	4	2	21.60	0	2008-06-15	4	Αλέξανδρος Παπαδιαμάντης
107	11-225-38	Η φόνισσα	4	3	15.80	1	2009-07-30	4	Αλέξανδρος Παπαδιαμάντης
108	11-224-98	Η γυφτοπούλα	4	4	9.90	1	2012-01-28	4	Αλέξανδρος Παπαδιαμάντης
109	11-229-78	Η μετανάστρια	4	3	23.10	1	2014-02-16	4	Αλέξανδρος Παπαδιαμάντης
104	11-226-45	Εντολή	0	2	7.78	1	1999-10-25	NULL	NULL

Αντίστοιχα με το **LEFT JOIN** υπάρχει η ένωση **RIGHT JOIN**, όπου δίνουμε βαρύτητα στο πίνακα που βρίσκεται στα δεξιά της δήλωσης **JOIN**. Εφαρμόζοντας το ερώτημα αυτό θα πάρουμε όλα τα βιβλία με αντιστοίχιση στους συγγραφείς αλλά και όποιους συγγραφείς δεν έχουν αντιστοίχιση σε βιβλία (προσέξτε ξανά τη τελευταία γραμμή των αποτελεσμάτων).

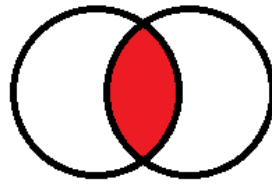
```
SELECT * FROM books LEFT JOIN author ON author=ID_auth
```

ID	ISBN	title	author	publication	price	status	date_added	ID_auth	name
101	11-222-33	Χίμαιρα	3	4	15.42	1	1995-05-15	3	Μ. Καραγάτσης
102	11-223-15	Το Δέκα	3	4	18.50	1	1996-03-28	3	Μ. Καραγάτσης
103	11-223-28	Ηλέκτρα	1	2	22.00	1	1997-04-02	1	Διδώ Σωτηρίου
105	11-223-36	Μαραμπού	2	1	11.40	1	2001-11-02	2	Νίκος Καββαδίας
106	11-245-35	Όνειρο στο κύμα	4	2	21.60	0	2008-06-15	4	Αλέξανδρος Παπαδιαμάντης
107	11-225-38	Η φόνισσα	4	3	15.80	1	2009-07-30	4	Αλέξανδρος Παπαδιαμάντης
108	11-224-98	Η γυφτοπούλα	4	4	9.90	1	2012-01-28	4	Αλέξανδρος Παπαδιαμάντης
109	11-229-78	Η μετανάστρια	4	3	23.10	1	2014-02-16	4	Αλέξανδρος Παπαδιαμάντης
110	11-222-85	Θα σας τηλεφωνώ	2	4	17.30	0	2015-03-29	2	Νίκος Καββαδίας
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	5	Νίκος Καζαντζάκης

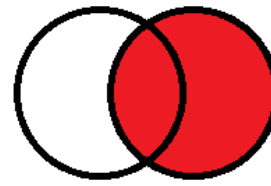
Αν φανταστούμε τις εγγραφές των δυο συνδεδεμένων πινάκων ως κύκλους, τότε θα μπορούσαμε να απεικονίσουμε τους τρεις τύπους της **JOIN** όπως στο παρακάτω σχήμα.



LEFT JOIN

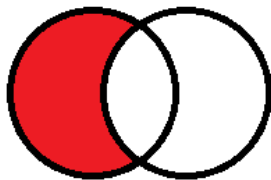


INNER JOIN

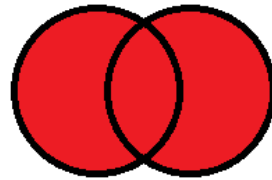


RIGHT JOIN

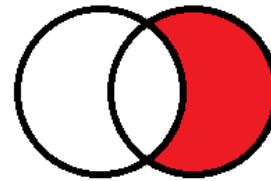
Με το ίδιο σκεπτικό θα μπορούσαμε να υλοποιήσουμε περισσότερους τύπους σύνδεσης. Σε κάποιες εκδόσεις και παλαιότερα πρότυπα της SQL, υποστηρίζεται και η δήλωση **OUTER JOIN**. Με τη δήλωση αυτή μπορούσαμε να πάρουμε διαφορετικά υποσύνολα (όπως φαίνεται στο παρακάτω σχήμα) αλλά δεν υποστηρίζεται από την MySQL, όπως και άλλες λειτουργίες συνόλων. Μπορούμε ωστόσο να υλοποιήσουμε τέτοιες ενώσεις χρησιμοποιώντας τη λειτουργία **UNION** και επιπλέον συνθήκες με τη **WHERE** στα ερωτήματα.



LEFT OUTER JOIN



FULL OUTER JOIN



RIGHT OUTER JOIN