

**ΠΕΡΙΕΧΟΜΕΝΑ:**

1. CROSS JOIN
2. INNER JOIN
 1. Σε 2 πίνακες
 2. Παρατηρήσεις για το Συντακτικό
 3. Σε 3+ πίνακες
 4. JOINS και NULLs
 5. EQUI-JOIN κ.α.

Βασίλης Σαγώνας

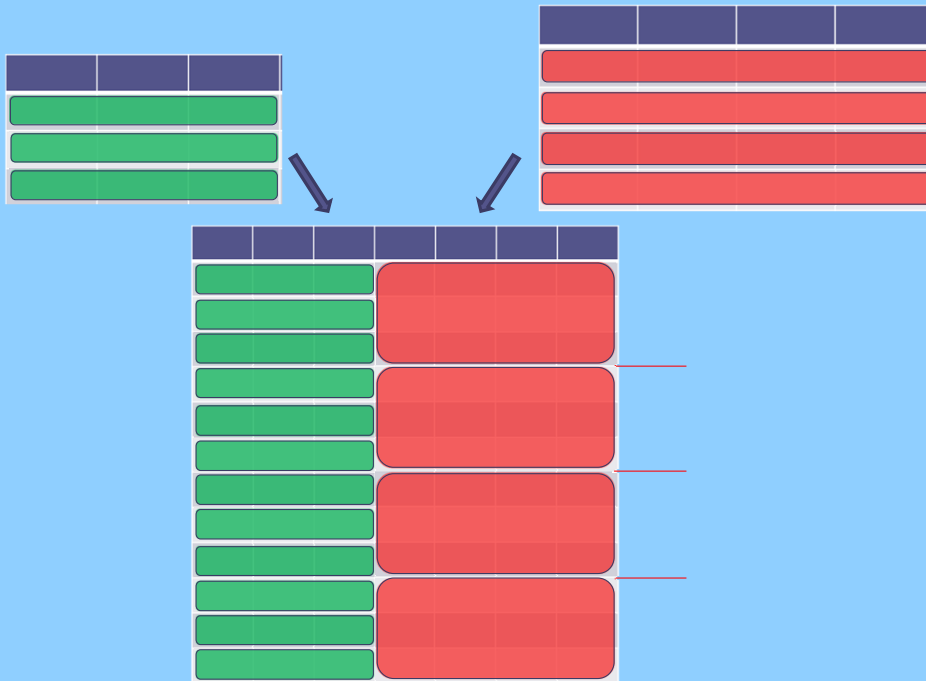
Σμαραγδένιος Χορηγός Μαθήματος

Πάνος Γ.

Ασημένιος Χορηγός Μαθήματος

CROSS JOIN:

- Γράφοντας:
FROM table1, table2
- Πραγματοποιείται το καρτεσιανό γινόμενο (cross product) των εγγραφών των δύο πινάκων,
 - δηλαδή κάθε εγγραφή του πρώτου πίνακα , συνδυάζεται με κάθε εγγραφή του δεύτερου πίνακα
 - και κατασκευάζεται έτσι ένας καινούργιος πίνακας επί του οποίου συνεχίζεται το ερώτημα SQL



1.1. Cross Join - Implicit Syntax

-- example 1.1 (script: cross.join.sql, DB: world)
SELECT *
FROM country, countrylanguage;

Εντελώς ισοδύναμο είναι το ακόλουθο συντακτικό (explicit syntax):

FROM table1 **CROSS JOIN** table2

(αυτό είναι και προτιμότερο να χρησιμοποιούμε στην πράξη)

1.2. Cross Join - Explicit Syntax

-- example 1.2
SELECT *
FROM country **CROSS JOIN** countrylanguage;

Για να λύσουμε προβλήματα ίδιας ονομασίας στηλών:

- Είτε χρησιμοποιούμε το όνομα του πίνακα ώστε να προηγείται της στήλης με την τελεία, π.χ.
SELECT country.name, city.name
FROM country **CROSS JOIN** city;
- είτε χρησιμοποιούμε συνώνυμα πινάκων (table aliases) ως εξής:
SELECT cn.name, ct.name
FROM country cn **CROSS JOIN** city ct;

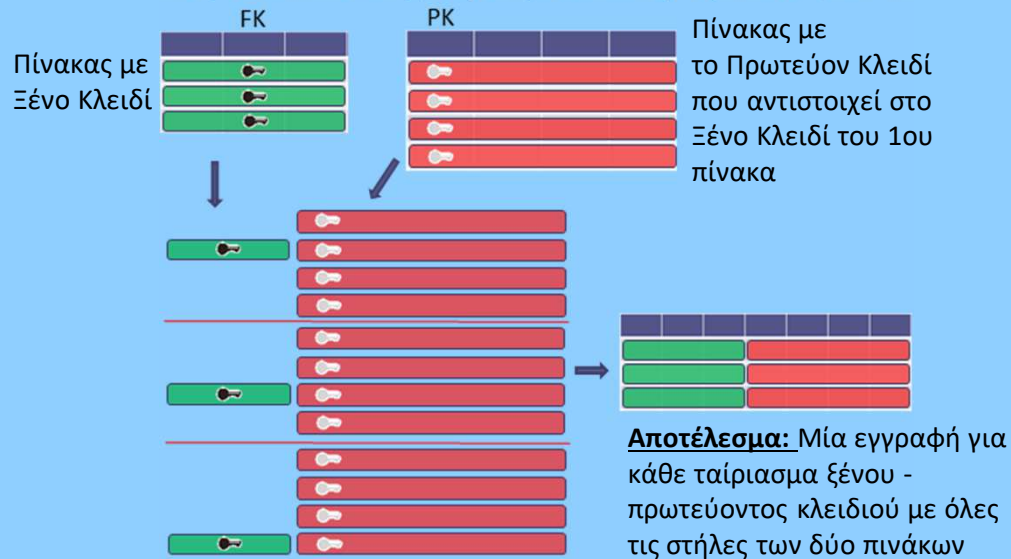
Σημείωση: Τα cross joins, σπάνια είναι χρήσιμα στην πράξη.

INNER JOIN (εσωτερική σύνδεση/συνένωση)

- Συνηθισμένη χρήση/σύνταξη:

```
FROM table1 t1 [INNER] JOIN table2 t2
ON t1.fk = t2.pk
```

- Πραγματοποιείται ταίριασμα στη σύνδεση που υπάρχει μεταξύ:
 - του ξένου κλειδιού του πρώτου πίνακα
 - με το πρωτεύον κλειδί του δεύτερου πίνακα
- και επιστρέφονται όλες οι γραμμές με βάση το παραπάνω ταίριασμα (Το ίδιο γίνεται κι αν γράψουμε αντίστροφα τους πίνακες)



INNER JOIN στους δύο πίνακες: Για κάθε εγγραφή:

- Για την τιμή του ξένου κλειδιού στον 1ο πίνακα
- Αναζητείται και εντοπίζεται η εγγραφή του 2ου πίνακα που ταιριάζει στο ξένο κλειδί (έχει ίδια τιμή στο πρωτεύον κλειδί)

Παράδειγμα:

- Η πόλη έχει ως ξένο κλειδί, τον κωδικό της χώρας.

country			
Code	Name	Continent	...
ABW	Aruba	North America	...
AFG	Afghanistan	Asia	...
AGO	Angola	Africa	...
AIA	Anguilla	North America	...

city			
ID	Name	CountryCode	...
1	Kabul	AFG	...
2	Qandahar	AFG	...
3	Herat	AFG	...

- Το ερώτημα:

-- example 2.1 (script: inner.join1.sql, DB: world)

```
SELECT country.name AS country, city.name AS city
FROM city JOIN country
ON city.CountryCode = country.Code
ORDER BY 1, 2;
```

- έχει ως αποτέλεσμα:

city	Country
Kabul	Afghanistan
Qandahar	Afghanistan
Herat	Afghanistan

Παρατηρήσεις για το Συντακτικό:

- Κάθε τύπος JOIN (τόσο το CROSS, το INNER και οι υπόλοιποι τύποι που θα δούμε στο επόμενο μάθημα) συντάσσονται με δύο τρόπους:

- Τη **ρητή σύνταξη (explicit syntax)** που χρησιμοποιεί το ON αμέσως μετά τη δήλωση (π.χ. για INNER JOIN):

```
FROM table1 t1 [INNER] JOIN table2 t2
ON t1.fk = t2.pk
```

- Την **έμμεση σύνταξη (implicit syntax)** όπου η σύνδεση καθορίζεται στη WHERE (π.χ. για INNER JOIN):

```
FROM table1 t1, table2 t2
WHERE t1.fk = t2.pk
```

- Διευκολύνσεις** για το συντακτικό:

- Είτε χρησιμοποιούμε συνώνυμα πινάκων (όπως παραπάνω)
- Είτε χρησιμοποιούμε απευθείας τα ονόματα πινάκων με την τελεία:

```
FROM table1 [INNER] JOIN table2
ON table1.fk = table2.pk
```

- Είτε (αν δεν υπάρχουν συγκρούσεις ονομάτων), χωρίς τίποτα από τα δύο, π.χ.:

```
FROM table1 [INNER] JOIN table2
ON fk = pk
```

Άσκηση 1: (DB: classicmodels)

A) Εντοπίστε τη σύνδεση:

- Πίνακας employees: FK officeCode
- Πίνακας offices: PK officeCode

Κατασκευάστε ερώτημα το οποίο θα επιστρέφει δύο στήλες: Η 1η να έχει το πλήρες ονοματεπώνυμο του υπαλλήλου και η 2η να έχει την πόλη στην οποία βρίσκεται το γραφείο:

- Χρησιμοποιώντας τη ρητή σύνταξη με συνώνυμα πινάκων
- Χρησιμοποιώντας την έμμεση σύνταξη με συνώνυμα πινάκων

B) Έπαναλάβετε τα δύο ερωτήματα, χρησιμοποιώντας τα πλήρη ονόματα των πινάκων (χωρίς συνώνυμα)

Βέλτιστη Πρακτική:

- Πάντα να χρησιμοποιούμε τη ρητή σύνταξη (οδηγεί σε λιγότερα λάθη)
- Πάντα να χρησιμοποιούμε συνώνυμα πινάκων

Παρατηρήσεις:

- Η σειρά των πινάκων δεν έχει σημασία στα INNER και CROSS JOIN
- (Υιοθετούμε, πάντως, την πολιτική να γράφουμε πρώτα τον πίνακα με το ξένο κλειδί και έπειτα τον πίνακα με το πρωτεύον κλειδί όταν κάνουμε ερωτήματα επί δύο πινάκων)

Άσκηση 2: DB: World (country, countrylanguage)

[βλ. υποχρεωτικά και το βίντεο στην άσκηση αυτή]

Κατασκευάστε ερώτημα που να επιστρέφει τις στήλες:

- Country (πλήρες όνομα χώρας - όχι κωδικός)
- Language (Γλώσσα)
- Percentage (ποσοστό ομιλούντων τη γλώσσα)

Άσκηση 3: DB ClassicModels (products, productLines)

Κατασκευάστε ερώτημα που να επιστρέφει τις στήλες:

- productName
- productLine (Αντιστοιχεί στην textDescription της productLines)
- quantityInStock

Σημαντικό:

- Το JOIN είναι καλό να το σκεφτόμαστε σαν μία επέκταση του FROM.
- Συνεπώς το πρώτο που κατασκευάζουμε είναι ένας μεγάλος πίνακας (με χρήση JOIN(s)) και έπειτα συνεχίζουμε με τον τρόπο που είδαμε στο μάθημα 1.3: WHERE => GROUP BY => HAVING => SELECT => ORDERBY => LIMIT.

Άσκηση 4: Πίνακες customers, payments

Ο μάνατζερ μας ζητάει ένα report με τις εξής στήλες:

- Όνομα Πελάτη (να ονομαστεί customer)
- Όνομα Πόλης Πελάτη (να ονομαστεί city)
- Συνολικό ποσό που έχει πληρώσει ο πελάτης (να ονομαστεί total_amount)
- Όριο πίστωσης (να ονομαστεί credit_limit)

Το report να αφορά μόνο τους πελάτες μας που είναι στις ΗΠΑ και τη Γαλλία και να εμφανίζει τις εγγραφές ταξινομημένες σε φθίνουσα σειρά ορίου πίστωσης.

Hint: Πρέπει να εμφανίζονται 47 γραμμές.

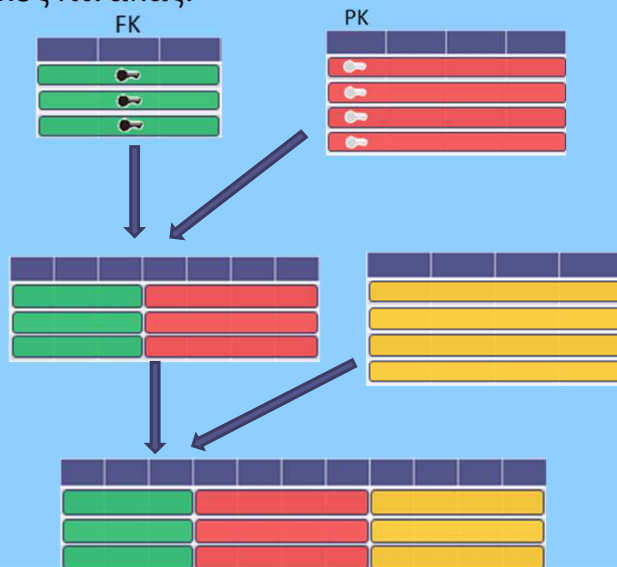
INNER JOIN για περισσότερους από 2 πίνακες

• Σύνταξη:

```
FROM table1 t1
  [INNER] JOIN table2 t2 ON t1.fk = t2.pk
  [INNER] JOIN table3 t3 ON join_condition
...
```

• όπου:

- Γίνεται το JOIN των 2 πρώτων πινάκων (παράγεται ενδιάμεσος πίνακας)
- Από τον ενδιάμεσο πίνακα και τον 3ο πίνακα παράγεται ο τελικός πίνακας.



Σημειώσεις:

- `join_condition`: μπορεί να είναι είτε σχέση FK του κοινού πίνακα με PK του 3ου πίνακα ή το αντίστροφο: FK του 3ου πίνακα με PK του κοινού πίνακα.
- Το συντακτικό επεκτείνεται αντίστοιχα σε παραπάνω από 3 πίνακες.
- και υπάρχουν όλες οι παραλλαγές του συντακτικού που είδαμε προηγούμενα (implicit, explicit, συντομογραφίες κ.λπ.)

-- example 3.1 (script: *inner.join2.sql*, DB: *world*)

```
SELECT ct.name AS city,
       ct.population as population,
       COUNT(*) AS languages_spoken
FROM city ct
  JOIN country cn ON ct.countrycode = cn.code
  JOIN countrylanguage cl ON cn.code = cl.countrycode
WHERE cn.name = 'Greece'
GROUP BY ct.name
ORDER BY population DESC
LIMIT 2
```

“Πλήθος θρησκειών και πληθυσμός, για τις 2 μεγαλύτερες σε πληθυσμό ελληνικές πόλεις”

ΜΑΘΗΜΑ 1.5: INNER JOINS

2.4. INNER JOIN και NULLs

- Αν το FK είναι NULL, τότε η γραμμή που το περιέχει δεν περιλαμβάνεται στο τελικό αποτέλεσμα.



-- example 4.1 (script: inner.join3.sql, DB: world)

```
SELECT COUNT(*)  
FROM customers;
```

```
SELECT COUNT(*)  
FROM customers  
WHERE salesRepEmployeeNumber IS NULL;
```

```
SELECT COUNT(*)  
FROM customers c JOIN employees e  
ON c.salesRepEmployeeNumber = e.employeeNumber;
```

Άσκηση 5: Πίνακες film, language (DB: sakila)

Εξετάζουμε την προέλευση των ταινιών της βάσης δεδομένων μας. Μετρήστε πόσες ταινίες υπάρχουν από κάθε γλώσσα στη βάση μας. (προσοχή μας ενδιαφέρει το language_id του πίνακα film και όχι το original_language_id). Το τελικό αποτέλεσμα να επιστρέφει δύο στήλες: Η 1η με το όνομα της γλώσσας και η 2η με το πλήθος των ταινιών στη γλώσσα αυτή.

Άσκηση 6: Πίνακες film, film_actor, actor (DB: sakila)

Κατασκευάστε έναν πίνακα ο οποίος θα εμφανίζει το ονοματεπώνυμο του ηθοποιού ακολουθούμενο από πόσες ταινίες έχει παίξει, ταξινομημένα σύμφωνα με το πλήθος των ταινιών σε φθίνουσα σειρά. Οι 3 πρώτες γραμμές του πίνακα πρέπει να είναι:

ACTOR	MOVIES
GINA DEGENERES	42
WALTER TORN	41
MARY KEITEL	40
...	...

- Αν και η χρήση των INNER JOIN που θα κάνουμε στην πράξη, θα είναι τις περισσότερες φορές η σύνδεση του FK ενός πίνακα με το PK του πίνακα στο οποίο αναφέρεται,
- στην πραγματικότητα, ένα INNER JOIN μπορεί να συγκρίνει στήλες του 1ου πίνακα με στήλες του 2ου πίνακα και επιστρέφει τους συνδυασμούς εγγραφών που ικανοποιούν τις συνθήκες.
- Έτσι υπάρχουν:

- Τα **EQUI-JOIN** που είναι INNER JOIN που γίνεται έλεγχος ισότητας, π.χ.:

```
FROM table1 t1 [INNER] JOIN table2 t2
ON t1.col1 = t2.col2
```

(Παρατηρήστε ότι δεν είναι ανάγκη να αναφερόμαστε σε σχέση FK-PK)

- Τα **non-EQUI-JOIN** που είναι INNER JOIN που γίνεται με άλλο σχεσιακό τελεστή (όχι =), π.χ. <, <=, <>, >, >=
- ενώ η σύγκριση μπορεί να γίνει και με περισσότερες από δύο στήλες, π.χ.:

```
FROM table1 t1 [INNER] JOIN table2 t2
ON t1.col1 op t2.col2 AND/OR t1.col1' op t2.col2'
```

- Τέλος το **NATURAL JOIN** είναι ένα EQUI-JOIN το οποίο γίνεται σε όλες τις στήλες των δύο πινάκων που έχουν το ίδιο όνομα (δεν απαιτείται να γράψουμε το ON)

```
FROM table1 t1 NATURAL JOIN table2 t2
```

Άσκηση 7:

Εξετάστε αν τα παρακάτω ερωτήματα είναι: (A) INNER JOIN, (B) EQUI-JOIN (Γ) non-EQUI-JOIN (Δ) NATURAL JOIN:

-- 7.1

```
SELECT *
FROM payments p JOIN orders o
ON p.paymentDate > o.orderDate;
```

-- 7.2

```
SELECT *
FROM customers c JOIN employees e
ON c.contactFirstName = e.firstName AND
c.contactLastName = e.lastName;
```

-- 7.3

```
SELECT *
FROM customers c JOIN employees e
ON NOT (c.contactFirstName > e.firstName OR
c.contactLastName <> e.lastName);
```

-- 7.4

```
SELECT *
FROM employees NATURAL JOIN offices;
```


Άσκηση 8: DB Sakila: (film, film_category, category)

Έχετε το δύσκολο έργο να προτείνετε μία ταινία στην πεθερά σας. Προκειμένου να έχει πιθανότητες να της αρέσει, λάβετε υπόψη σας τα εξής στην επιλογή:

- Να μην έχει χαρακτηριστεί ακατάλληλη (R ή NC-17 στο πεδίο rating)
- Η διάρκεια της να είναι από 1 έως 1, 1/2 ώρα (πεδίο length)
- Στην περιγραφή (πεδίο description) να βρίσκεται η λέξη boring ή love ή documentary
- Το κόστος αντικατάστασής της να είναι το πολύ 14 δολάρια.
- Η κατηγορία της ταινίας (πίνακας category) να είναι Travel, Family, Comedy, Classics, ή Comedy

Άσκηση 9: Πίνακες actor, film_actor, film, film_category, category

Σε ποια είδη ταινιών έχει παίξει η γνωστή ηθοποιός JULIA FAWCETT;

Άσκηση 10:

- A) Διατυπώστε ένα ενδιαφέρον ερώτημα στα ελληνικά πάνω σε οποιαδήποτε από τις 3 βάσεις που να απαιτεί INNER JOIN
- B) Γράψτε το SQL ερώτημα
- C) Αναρτήστε τα A-B σαν σχόλιο στο βίντεο στο youtube!