

# Βάσεις Δεδομένων

---

Εργαστήριο 4: Ερωτήματα SQL με σύζευξη πινάκων

Τσούρος Δήμος

# Το σχήμα της βάσης company

*departments*(depid, depname, manager)

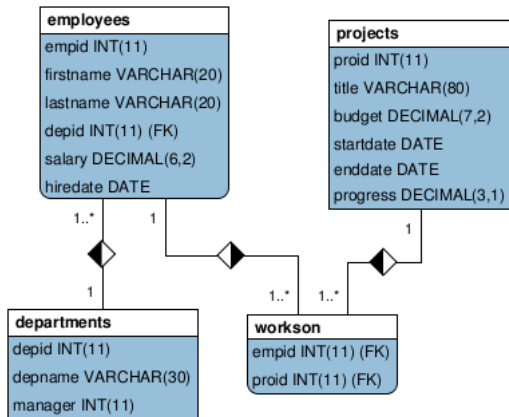
*employees*(empid, firstname, lastname, depid, salary, hiredate)

*projects*(proid, title, budget, startdate, enddate, progress)

*workson*(empid, proid)

- *departments*, τα τμήματα της εταιρείας
- *employees*, οι υπάλληλοι της εταιρείας
- *projects*, τα έργα που εκτελεί η εταιρεία
- *workson*, η απασχόληση των υπαλλήλων στα έργα

# Το σχήμα της βάσης company



- *departments*, τα τμήματα της εταιρείας
- *employees*, οι υπάλληλοι της εταιρείας
- *projects*, τα έργα που εκτελεί η εταιρεία
- *workson*, η απασχόληση των υπαλλήλων στα έργα

# Πρωτεύοντα και ξένα κλειδιά

- 1 Κάθε πίνακας έχει ένα **πρωτεύον κλειδί**.
- 2 Το **πρωτεύον κλειδί** μπορεί να είναι απλό (ένα πεδίο), ή σύνθετο (συνδυασμός πεδίων).
- 3 Κάθε εγγραφή ενός πίνακα μπορεί να προσδιοριστεί με τη χρήση του πρωτεύοντος κλειδιού.
- 4 Η σύνδεση δεδομένων από διαφορετικούς πίνακες **σύζευξη** γίνεται (συνήθως) με τη χρήση του **ξένου κλειδιού**,
- 5 Ένας πίνακας μπορεί να έχει πολλά ξένα κλειδιά ή να μην έχει κανένα.

## Συσχέτιση departments και employees N:1

- Ο πίνακας *departments* έχει **πρωτεύον κλειδί** το πεδίο *depid*.
- Ο πίνακας *employees* έχει **πρωτεύον κλειδί** το πεδίο *empid*.
- Ο πίνακας *employees* έχει **ξένο κλειδί** το πεδίο *depid*, το οποίο παίρνει τιμές που **αναφέρονται** στις τιμές του πεδίου *depid* του πίνακα *departments*:

$$departments.depid = employees.depid$$

- Το πεδίο *employees.depid* **δεν είναι πρωτεύον κλειδί** και δεν παίρνει μοναδικές τιμές: πολλοί υπάλληλοι εργάζονται στο ίδιο τμήμα.
- Οι πίνακες *departments* και *employees* συσχετίζονται μεταξύ τους με συσχέτιση **Πολλά προς Ένα**.

# Συσχέτιση departments - employees

employees - company1 - OpenOffice.org Base: Table Data

Αρχείο Επεξεργασία Προβολή Εισαγωγή Εργαλεία Παράθυρο Βοήθεια

|  | empid | firstname | lastname     | depid | salary  | hiredate  |
|--|-------|-----------|--------------|-------|---------|-----------|
|  | 102   | Νικηφόρος | Διαμαντίδης  | 6     | 1212.50 | 2/6/2003  |
|  | 109   | Μαρία     | Αθανασίου    | 1     | 2787.69 | 26/1/2000 |
|  | 153   | Μαρία     | Αλεβιζάτου   | 2     | 1321.92 | 15/5/2001 |
|  | 172   | Χρήστος   | Βλάσης       | 3     | 1101.70 | 4/7/2000  |
|  | 189   | Θεόδωρος  | Αγγελίνας    | 6     | 1908.28 | 19/6/2000 |
|  | 205   | Ηλίας     | Μακρής       | 3     | 1050.96 | 23/2/2001 |
|  | 206   | Νίκος     | Βλάχος       | 4     | 1102.04 | 3/12/2002 |
|  | 230   | Βαγγέλης  | Χριστόπουλος | 4     |         | 3/12/2002 |

Εγγραφή 2 από 30 (1)

|  | depid | depname                | manager |
|--|-------|------------------------|---------|
|  | 1     | Διοίκησης/Επίβλεψης    | 109     |
|  | 2     | Οικονομολόγων/Λογιστών | 153     |
|  | 3     | Επιστημόνων/Μηχανικών  | 431     |
|  | 4     | Εξωτερικών συνεργατών  | 230     |
|  | 5     | Γραμματείας            | 234     |
|  | 6     | Μάνατζμεντ/Πωλήσεων    | 189     |

Εγγραφή 1 από 6 (1)

# Συσχέτιση departments και employees 1:1

- Ο πίνακας *departments* έχει **πρωτεύον κλειδί** το πεδίο *depid*.
- Ο πίνακας *employees* έχει **πρωτεύον κλειδί** το πεδίο *empid*.
- Ο πίνακας *departments* έχει **ξένο κλειδί** το πεδίο *manager*, το οποίο παίρνει τιμές που **αναφέρονται** στις τιμές του πεδίου *empid* του πίνακα *employees*:

*departments.manager* = *employees.empid*

- Το πεδίο *departments.manager* **δεν είναι πρωτεύον κλειδί** αλλά παίρνει μοναδικές τιμές (**UNIQUE**).
- Οι πίνακες *departments* και *employees* συσχετίζονται μεταξύ τους με συσχέτιση **Ένα προς Ένα**.

# Συσχέτιση employees και projects N:N

- 1 Ο πίνακας *projects* έχει **πρωτεύον κλειδί** το πεδίο *proid*.
- 2 Ο πίνακας *workson* έχει **πρωτεύον κλειδί** το συνδυασμό των πεδίων *empid* και *proid* (**σύνθετο κλειδί**).
- 3 Το πεδίο *empid* είναι **ξένο κλειδί** στον πίνακα *workson* και αναφέρεται στο πεδίο *employees.empid*.
- 4 Το πεδίο *proid* είναι **ξένο κλειδί** στον πίνακα *workson* και αναφέρεται στο πεδίο *projects.proid*.
- 5 Ένας υπάλληλος απασχολείται σε πολλά έργα, ένα έργο απασχολεί πολλούς υπαλλήλους, επομένως η συσχέτιση είναι **Πολλά προς Πολλά**.
- 6 Η σύζευξη των πινάκων *employees* και *projects* γίνεται μέσω του πίνακα *workson*.

# Καρτεσιανό γινόμενο

## Καρτεσιανό γινόμενο:

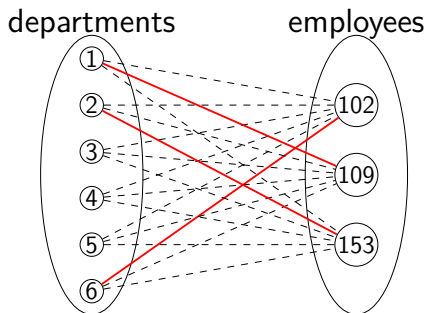
Ο συνδυασμός των πινάκων *departments* και *employees*  
 $departments \times employees$

λέγεται καρτεσιανό γινόμενο, και περιλαμβάνει όλους τους συνδυασμούς των εγγραφών των πινάκων *departments* και *employees*.

- Στο αποτέλεσμα περιλαμβάνονται όλα τα πεδία των πινάκων *departments* και *employees*.
- Οι εγγραφές του αποτελέσματος περιλαμβάνει όλους τους συνδυασμούς των εγγραφών των πινάκων *departments* και *employees*.

# Καρτεσιανό γινόμενο *departments* $\times$ *employees*

- Όλες οι δυνατές συνδέσεις ανάμεσα στα τμήματα και σε δείγμα 3 υπαλλήλων.
- Μόνο οι **κόκκινες γραμμές** αντιστοιχούν στην πραγματικότητα: (1, 109), (6, 102), (2,153).
- Τα καρτεσιανό γινόμενο απεικονίζει όλα τα **πιθανά ενδεχόμενα**, όχι απαραίτητα τα **πραγματικά γεγονότα**.



# Καρτεσιανό γινόμενο στην SQL

## Καρτεσιανό γινόμενο τμημάτων και υπαλλήλων

*departments*  $\times$  *employees*

```
1 SELECT *  
2 FROM departments, employees;
```

- Η σύνταξη στην **SQL** είναι απλή: γράφουμε τους πίνακες μετά τον όρο **FROM** και τους χωρίζουμε με κόμμα.
- Μπορούμε να γράψουμε περισσότερο από δύο πίνακες.
- Προσοχή! το αποτέλεσμα μπορεί να περιέχει μεγάλο όγκο εγγραφών, πχ δύο πίνακες με 6 και 30 εγγραφές αντίστοιχα δίνουν στο αποτέλεσμα  $6 \times 30 = 180$  εγγραφές.

# Καρτεσιανό γινόμενο στην SQL92, CROSS JOIN

## Με χρήση του όρου CROSS JOIN

```
1 SELECT *  
2 FROM departments CROSS JOIN employees;
```

## Η απλά JOIN

```
1 SELECT *  
2 FROM departments JOIN employees;
```

- Οι δύο εκφράσεις είναι ισοδύναμες, θα επιστρέψουν το ίδιο αποτέλεσμα.
- Προτιμούμε τον πρώτο τρόπο **CROSS JOIN**, δηλώνει με πιο καθαρό τρόπο τη σύζευξη με βάση το καρτεσιανό γινόμενο.

# Αποτέλεσμα καρτεσιανού γινομένου

```
1 SELECT *
2 FROM departments, employees;
```

| depid | depname       | manager | empid | firstname | lastname    | depid | salary  | hiredate   |
|-------|---------------|---------|-------|-----------|-------------|-------|---------|------------|
| 1     | Διοίκ./Επιβ.  | 109     | 102   | Νικηφόρος | Διαμαντίδης | 6     | 1212.50 | 2003-06-02 |
| 2     | Οικον./Λογ.   | 153     | 102   | Νικηφόρος | Διαμαντίδης | 6     | 1212.50 | 2003-06-02 |
| 3     | Επιστημ./Μηχ. | 431     | 102   | Νικηφόρος | Διαμαντίδης | 6     | 1212.50 | 2003-06-02 |
| 4     | Εξωτ. συνερ.  | 230     | 102   | Νικηφόρος | Διαμαντίδης | 6     | 1212.50 | 2003-06-02 |
| 5     | Γραμματείας   | 234     | 102   | Νικηφόρος | Διαμαντίδης | 6     | 1212.50 | 2003-06-02 |
| 6     | Μάνατζ./Πωλ.  | 189     | 102   | Νικηφόρος | Διαμαντίδης | 6     | 1212.50 | 2003-06-02 |
| 1     | Διοίκ./Επιβ.  | 109     | 109   | Μαρία     | Αθανασίου   | 1     | 2787.69 | 2000-01-26 |
| 2     | Οικον./Λογ.   | 153     | 109   | Μαρία     | Αθανασίου   | 1     | 2787.69 | 2000-01-26 |
| 3     | Επιστημ./Μηχ. | 431     | 109   | Μαρία     | Αθανασίου   | 1     | 2787.69 | 2000-01-26 |

.....

180 rows in set (0.00 sec)

# Επιλογή πεδίων από πίνακες

```
1  SELECT departments.depid, depname, empid, lastname
2  FROM departments, employees;
```

| depid | depname       | empid | lastname    |
|-------|---------------|-------|-------------|
| 1     | Διοίκ./Επιβ.  | 102   | Διαμαντίδης |
| 2     | Οικον./Λογ.   | 153   | Διαμαντίδης |
| 3     | Επιστημ./Μηχ. | 431   | Διαμαντίδης |
| 4     | Εξωτ. συνερ.  | 230   | Διαμαντίδης |
| 5     | Γραμματείας   | 234   | Διαμαντίδης |
| 6     | Μάνατζ./Πωλ.  | 189   | Διαμαντίδης |
| 1     | Διοίκ./Επιβ.  | 109   | Αθανασίου   |
| 2     | Οικον./Λογ.   | 153   | Αθανασίου   |
| 3     | Επιστημ./Μηχ. | 431   | Αθανασίου   |

.....  
180 rows in set (0.00 sec)

# Επιλογή πεδίων από πίνακες

```
1 SELECT departments.depid, depname, empid, lastname  
2 FROM departments, employees;
```

## Τι προσέχουμε:

- 1 Οποιοδήποτε όνομα πεδίου/στήλης υπάρχει στους πίνακες που ακολουθούν τον όρο **FROM** μπορούν να τοποθετηθούν μετά τον όρο **SELECT**.
- 2 Σε περίπτωση που το όνομα πεδίου είναι μοναδικό σε όλους τους πίνακες μπορούμε να το γράψουμε ως έχει, πχ *depname* ή *lastname*.
- 3 Αν το ίδιο όνομα πεδίου υπάρχει σε δύο διαφορετικούς πίνακες, τότε πρέπει να γραφεί με τη μορφή **πίνακας.πεδίο**.

# Μετονομασία πινάκων

## Τι προσέχουμε:

```
1 SELECT d.depid, d.depname, e.empid, e.lastname  
2 FROM departments d, employees e;
```

- ❶ Η χρήση ψευδωνύμων είναι προαιρετική.
- ❷ Βολεύει όταν το όνομα του πίνακα γράφεται πολλές φορές.

## Ίδιοι κανόνες στην SQL92:

```
1 SELECT d.depid, d.depname, e.empid, e.lastname  
2 FROM departments d CROSS JOIN employees e;
```

# Καρτεσιανό γινόμενο υπαλλήλων και έργων

```
1  SELECT e.empid, p.proid  
2  FROM employees e, projects p;
```

```
4  | empid | proid |  
5  -----
```

```
6  | 102 | 5 |  
7  | 102 | 12 |  
8  | 102 | 14 |  
9  | 102 | 21 |  
10 | 102 | 38 |  
11 | 102 | 43 |  
12 | 109 | 5 |  
13 | 109 | 12 |
```

```
14 .....  
15 180 rows in set (0.00 sec)
```

# Καρτεσιανού γινόμενο υπαλλήλων και έργων

```
1 SELECT e.empid, p.proid  
2 FROM employees e, projects p;
```

- 1 Το αποτέλεσμα συνδέει **όλους** τους υπαλλήλους με **όλα** τα έργα.
- 2 Είναι **πιθανό** ένας υπάλληλος να απασχολείται σε όλα τα έργα, αλλά αυτό δεν συμβαίνει **υποχρεωτικά** για όλους τους υπαλλήλους.
- 3 Είναι **πιθανό** ένα έργο να απασχολεί όλους τους υπαλλήλους, αλλά αυτό δεν συμβαίνει **υποχρεωτικά** για όλα τα έργα.
- 4 Το καρτεσιανό γινόμενο μας δίνει όλα τα **πιθανά ενδεχόμενα**, δεν μας αποκαλύπτει το «τι συμβαίνει».

# Σύζευξη θήτα στην SQL

## Σύζευξη τμημάτων και υπαλλήλων

```
1 SELECT *  
2   FROM departments, employees  
3   WHERE departments.depid = employees.depid;
```

Προσέξτε τη γραφή **πίνακας.πεδίο**.

Στη φράση **WHERE** μπορούν να προστεθούν επιπλέον περιορισμοί με λογική σύζευξη (**AND**).

Η θήτα σύζευξη είναι δυνατό να πραγματοποιηθεί και με πεδία που δεν έχουν το ίδιο όνομα.

Στο αποτέλεσμα εμφανίζονται μόνο οι εγγραφές για τις οποίες η συνθήκη είναι **TRUE**.

## θ σύζευξη, επιπλέον παράδειγμα

Τμήματα με υπαλλήλους με μισθό πάνω από 1500 €

```
1 SELECT DISTINCT d.depname
2   FROM departments d, employees e
3  WHERE d.depid = e.depid
4        AND e.salary > 1500;
```

Περιορισμός εγγραφών της θ σύζευξης με βάση μια παράσταση σύγκρισης.

Στο αποτέλεσμα εμφανίζονται μόνο οι εγγραφές για τις οποίες όλες οι συνθήκες είναι **TRUE**.

Η θ σύζευξη δεν έχει νόημα να ακολουθείται από λογική διάζευξη **OR**.

# Θ σύζευξη στην SQL 1:1

Να βρεθούν τα ονοματεπώνυμα των υπαλλήλων που είναι διευθυντές τμημάτων

```
1 SELECT firstname, lastname
2     FROM departments, employees
3     WHERE departments.manager = employees.empid;
```

Να βρεθεί το ονοματεπώνυμο του υπαλλήλου που διευθύνει το τμήμα 2

```
1 SELECT firstname, lastname
2     FROM departments, employees
3     WHERE departments.manager = employees.empid
4     AND departments.depid = 2;
```

# Θ σύζευξη στην SQL N:N

Να βρεθεί ο τίτλος των έργων και τα ονοματεπώνυμα των υπαλλήλων που συμμετέχουν σε αυτά.

```
1 SELECT title, firstname, lastname
2     FROM employees, workson, projects
3     WHERE employees.empid = workson.empid
4           AND workson.proid = projects.proid;
```

Να βρεθεί ο τίτλος των έργων και τα ονοματεπώνυμα των υπαλλήλων που συμμετέχουν σε αυτά.

```
1 SELECT title, firstname, lastname
2     FROM employees, projects
3     WHERE employees.empid = workson.empid
4           AND workson.proid = projects.proid;
```

# Θ ανισοσύζευξη στην SQL

Να βρεθούν τα ονοματεπώνυμα των υπαλλήλων που προσλήφθηκαν μετά την έναρξη του έργου με κωδικό 21.

```
1 SELECT firstname, lastname
2     FROM employees, workson, projects
3     WHERE employees.empid = workson.empid
4           AND workson.proid = projects.proid
5           AND employees.hiredate > projects.startdate
6           AND projects.proid = 21;
```

Να βρεθούν τα ονοματεπώνυμα των υπαλλήλων που προσλήφθηκαν μετά την έναρξη του έργου με κωδικό 21.

```
1 SELECT firstname, lastname
2     FROM employees, projects
3     WHERE employees.hiredate > projects.startdate
4           AND projects.proid = 21;
```

# Μετονομασία πινάκων

## Με χρήση του AS

```
1 FROM employees AS e
```

## Χωρίς χρήση του AS

```
1 FROM employees e
```

## Στη σύζευξη

```
1 FROM departments d, employees e
```

## Στο ερώτημα

```
1 SELECT e.firstname, e.lastname  
2 FROM employees e, projects p  
3 WHERE e.hiredate > p.startdate  
4 AND p.proid = 21;
```

# Φυσική σύζευξη στην SQL

## Σύζευξη τμημάτων και υπαλλήλων

```
1 SELECT *  
2 FROM departments NATURAL JOIN employees;
```

Απαιτείται η ύπαρξη ενός τουλάχιστον κοινού πεδίου, εδώ το πεδίο **depid**.

Μόνο οι εγγραφές όπου οι τιμές του κοινού πεδίου ταυτίζονται υπάρχουν στο αποτέλεσμα του ερωτήματος.

Το κοινό γνώρισμα υπάρχει μόνο μία φορά στο αποτέλεσμα.

# Αποτέλεσμα φυσικής σύζευξης

```
1  SELECT *
```

```
2      FROM departments NATURAL JOIN employees;
```

```
3
4  depid  depname      manager  empid  firstname  lastname      salary  hiredate
5  -----
6      1  Διοίκ./Επιβ.    109      109  Μαρία      Αθανασίου     2787.69  2000-01-26
7      1  Διοίκ./Επιβ.    109      502  Κρινιώ     Μαροπούλου    1754.67  2001-03-07
8      1  Διοίκ./Επιβ.    109      901  Κυριάκος   Ρούσσης       1852.99  2001-11-01
9      2  Οικον./Λογ.        153      153  Μαρία      Αλεβιζάτου    1321.92  2001-05-15
10     2  Οικον./Λογ.        153      243  Δέσποινα   Παπαδοπούλου  1609.52  1999-03-05
```

```
11  .....
12  30 rows in set (0.00 sec)
```

# Προβολή και φυσική σύζευξη στην SQL

Να βρεθεί ο κωδικός και το επώνυμο των υπαλλήλων, καθώς και το όνομα του τμήματος στο οποίο απασχολούνται

# Προβολή και φυσική σύζευξη στην SQL

Να βρεθεί ο κωδικός και το επώνυμο των υπαλλήλων, καθώς και το όνομα του τμήματος στο οποίο απασχολούνται

```
1 SELECT empid, lastname, depname  
2 FROM departments NATURAL JOIN employees;
```

# Φυσική σύζευξη τριών πινάκων στην SQL

Να βρεθεί το ονοματεπώνυμο των υπαλλήλων, ο κωδικός και ο τίτλος του έργου που απασχολούνται

# Φυσική σύζευξη τριών πινάκων στην SQL

Να βρεθεί το ονοματεπώνυμο των υπαλλήλων, ο κωδικός και ο τίτλος του έργου που απασχολούνται

```
1 SELECT firstname, lastname, proid, title
2     FROM employees NATURAL JOIN workson
3         NATURAL JOIN projects;
```

## Φυσική σύζευξη ή καρτεσιανό γινόμενο;

$departments \bowtie projects = departments \times projects$

```
1 SELECT depid, proid
2 FROM departments NATURAL JOIN projects;
```

| depid | proid |
|-------|-------|
| 1     | 5     |
| 2     | 5     |
| 6     | 5     |
| 4     | 5     |
| 5     | 5     |
| 3     | 5     |
| 1     | 12    |
| 2     | 12    |
| ...   | ...   |

Το αποτέλεσμα είναι ίδιο με αυτό του φυσικού γινομένου **(ειδική περίπτωση)**.

Αν δεν υπάρχει κοινό πεδίο στους πίνακες της φυσικής σύζευξης τότε φυσική σύζευξη και καρτεσιανό γινόμενο θα δώσουν το ίδιο αποτέλεσμα.

# Περιορισμός εγγραφών στη φυσική σύζευξη

Να βρεθεί ο κωδικός και το επώνυμο των υπαλλήλων που απασχολούνται στο έργο με κωδικό 5

# Περιορισμός εγγραφών στη φυσική σύζευξη

Να βρεθεί ο κωδικός και το επώνυμο των υπαλλήλων που απασχολούνται στο έργο με κωδικό 5

```
1 SELECT empid, lastname
2     FROM employees NATURAL JOIN workson
3     WHERE proid = 5;
```

Να βρεθεί ο κωδικός και το επώνυμο των υπαλλήλων που απασχολούνται στο έργο με κωδικό 5

```
1 SELECT empid, lastname
2     FROM employees
3     WHERE proid = 5;
```

# Εσωτερική σύζευξη στην SQL

## Σύζευξη τμημάτων και υπαλλήλων

```
1 SELECT *  
2   FROM departments INNER JOIN employees  
3   ON departments.depid = employees.depid;
```

- ❶ Οι δύο πίνακες ενώνονται με τον όρο **INNER JOIN**.
- ❷ Αντί για **WHERE** υπάρχει (**υποχρεωτικά**) μετά το **INNER JOIN** η φράση **ON**.
- ❸ Το πεδίο σύζευξης (εδώ το *depid*) υπάρχει δύο φορές στο αποτέλεσμα.
- ❹ Το πεδίο σύζευξης μπορεί να έχει διαφορετικό όνομα στους δύο πίνακες.

# Ισοδυναμία εσωτερικής και θ σύζευξης στην SQL

## Σύζευξη τμημάτων και υπαλλήλων με εσωτερική σύζευξη

```
1 SELECT *  
2     FROM departments INNER JOIN employees  
3         ON departments.depid = employees.depid;
```

## Σύζευξη τμημάτων και υπαλλήλων με θ σύζευξη

```
1 SELECT *  
2     FROM departments JOIN employees  
3         WHERE departments.depid = employees.depid;
```

# Όχι στις υπερβολές

## Σύζευξη τμημάτων και υπαλλήλων με εσωτερική και θ σύζευξη

```
1 SELECT *  
2     FROM departments INNER JOIN employees  
3         ON departments.depid = employees.depid  
4 WHERE departments.depid = employees.depid;
```

- 1 Περιττό.
- 2 Δεν είναι συντακτικά λάθος, είναι όμως εννοιολογικά μπερδεμένο.
- 3 Μία φορά αρκεί, η πολυλογία φέρνει λάθη.
- 4 Δυσκολία συντήρησης του κώδικα.

# Μετονομασία πινάκων

## Με χρήση του AS

```
1 FROM employees AS e
```

## Χωρίς χρήση του AS

```
1 FROM employees e
```

## Στη σύζευξη

```
1 FROM departments d INNER JOIN employees e
```

## Στο ερώτημα

```
1 SELECT *  
2 FROM departments d INNER JOIN employees e;
```

## Παράδειγμα INNER JOIN – 1

Να δοθεί το επώνυμο των υπαλλήλων και το όνομα του τμήματος στο οποίο εργάζονται

# Παράδειγμα INNER JOIN – 1

Να δοθεί το επώνυμο των υπαλλήλων και το όνομα του τμήματος στο οποίο εργάζονται

```
1 SELECT employees.lastname, departments.depname
2 FROM departments INNER JOIN employees
3 ON employees.depid = departments.depid;
```

Η σύζευξη γίνεται με βάση το κοινό του πεδίο *depid*.

Η σύζευξη με βάση **πρωτεύον** και **ξένο** κλειδί είναι η πλέον συνηθισμένη περίπτωση σύζευξης.

## Παράδειγμα INNER JOIN – 2

Να δοθεί το όνομα των υπαλλήλων, το όνομα του τμήματος στο οποίο εργάζονται, και ο μισθός τους

# Παράδειγμα INNER JOIN – 2

Να δοθεί το όνομα των υπαλλήλων, το όνομα του τμήματος στο οποίο εργάζονται, και ο μισθός τους

```
1 SELECT e.firstname, e.lastname, d.depname, e.salary
2 FROM employees e INNER JOIN departments d
3 ON e.depid = d.depid;
```

| firstname | lastname     | depname                | salary  |
|-----------|--------------|------------------------|---------|
| Μαρία     | Αθανασίου    | Διοίκησης/Επίβλεψης    | 2787.69 |
| Κρινιώ    | Μαροπούλου   | Διοίκησης/Επίβλεψης    | 1754.67 |
| Κυριάκος  | Ρούσσης      | Διοίκησης/Επίβλεψης    | 1852.99 |
| Μαρία     | Αλεβιζάτου   | Οικονομολόγων/Λογιστών | 1321.92 |
| Δέσποινα  | Παπαδοπούλου | Οικονομολόγων/Λογιστών | 1609.52 |
| Πέτρος    | Αρβανιτάκης  | Οικονομολόγων/Λογιστών | 1323.80 |
| ...       | ...          | ...                    | ...     |

## Παράδειγμα INNER JOIN – 3

Να δοθεί το όνομα των υπαλλήλων, το όνομα του τμήματος στο οποίο εργάζονται και ο μισθός τους, για υπαλλήλους με μισθό μεταξύ 1050 και 1300 €

## Παράδειγμα INNER JOIN – 3

Να δοθεί το όνομα των υπαλλήλων, το όνομα του τμήματος στο οποίο εργάζονται και ο μισθός τους, για υπαλλήλους με μισθό μεταξύ 1050 και 1300 €

```
1 SELECT e.firstname, e.lastname, d.depname, e.salary
2     FROM employees e INNER JOIN departments d
3         ON e.depid = d.depid
4     WHERE e.salary BETWEEN 1050 AND 1300;
```

Ο όρος **WHERE** μπορεί να χρησιμοποιηθεί για περιορισμό των εγγραφών.

Η παράσταση συνθήκης μπορεί να αφορά οποιοδήποτε πεδίο από αυτά που υπάρχουν στους δύο πίνακες.

## Παράδειγμα INNER JOIN – 4

Κωδικός και όνομα όλων των υπαλλήλων που συμμετέχουν στο έργο με κωδικό 38, με αύξουσα ταξινόμηση ως προς το επώνυμο

## Παράδειγμα INNER JOIN – 4

Κωδικός και όνομα όλων των υπαλλήλων που συμμετέχουν στο έργο με κωδικό 38, με αύξουσα ταξινόμηση ως προς το επώνυμο

```
1  SELECT e.empid, e.firstname, e.lastname
2      FROM employees e INNER JOIN workson w
3          ON e.empid = w.empid
4      WHERE w.proid = 38
5      ORDER BY e.lastname ASC;
```

Ο όρος **ORDER BY** πάντα στο τέλος.

Προσέξτε πως χρειάζεται **σύζευξη** ακόμα και αν όλα τα πεδία που εμφανίζονται μετά τον όρο **SELECT** βρίσκονται σε ένα πίνακα.

Ξαναγράψτε την απάντηση με υποερώτημα (παρακάτω μάθημα).

# Πολλά προς πολλά

Να βρεθούν τα ονόματα των υπαλλήλων και ο κωδικός και ο προϋπολογισμός των έργων στα οποία συμμετέχουν υπάλληλοι με μισθό μεγαλύτερο από €1700

# Πολλά προς πολλά

Να βρεθούν τα ονόματα των υπαλλήλων και ο κωδικός και ο προϋπολογισμός των έργων στα οποία συμμετέχουν υπάλληλοι με μισθό μεγαλύτερο από €1700

```
1 SELECT e.firstname, e.lastname, p.proid, p.budget
2     FROM (employees e INNER JOIN workson w
3           ON e.empid = w.empid)
4           INNER JOIN projects p
5           ON p.proid = w.proid
6 WHERE e.salary > 1700;
```

# Πολλά προς πολλά – Εναλλακτικός τρόπος

Να βρεθούν τα ονόματα των υπαλλήλων και ο κωδικός και ο προϋπολογισμός των έργων στα οποία συμμετέχουν υπάλληλοι με μισθό μεγαλύτερο του 1700€

```
1  SELECT e.firstname, e.lastname, p.proid, p.budget
2      FROM employees e, workson w, projects p
3  WHERE e.empid = w.empid
4         AND p.proid = w.proid
5         AND e.salary > 1700;
```

## Ερώτημα με 4 πίνακες

Να βρεθεί το όνομα των υπαλλήλων και του τμήματος των υπαλλήλων για όλους τους υπαλλήλους που προσλήφθηκαν μετά από την 1/1/2002 και απασχολούνται σε έργα με βαθμό προόδου τουλάχιστον 20%

# Ερώτημα με 4 πίνακες

Να βρεθεί το όνομα των υπαλλήλων και του τμήματος των υπαλλήλων για όλους τους υπαλλήλους που προσλήφθηκαν μετά από την 1/1/2002 και απασχολούνται σε έργα με βαθμό προόδου τουλάχιστον 20%

```
1 SELECT DISTINCT e.firstname, e.lastname, d.depname
2   FROM ((departments d INNER JOIN employees e
3         ON d.depid = e.depid)
4         INNER JOIN workson w
5         ON e.empid = w.empid)
6         INNER JOIN projects p
7         ON p.proid = w.proid
8  WHERE e.hiredate > '2002-01-01'
9        AND p.progress > 20;
```