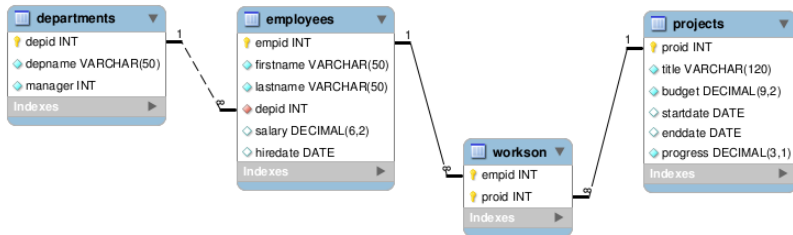


Τυπικό παράδειγμα βάσης δεδομένων



Η βάση company

- Μια εταιρεία μελετών έχει πολλούς υπαλλήλους.
- Η υπάλληλοι εργάζονται σε διάφορα τμήματα της εταιρείας.
- Κάθε τμήμα διοικείται από έναν υπάλληλο.
- Οι υπάλληλοι απασχολούνται σε διάφορα έργα μελετών.
- Σε κάθε έργο απασχολούνται πιθανά πολλοί υπάλληλοι από διάφορα τμήματα.

Ο πίνακας *employees* από τη βάση *company*

Έστω η σχέση *employees* με σχήμα:

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

empid	firstname	lastname	depid	salary	hiredate
102	Νικηφόρος	Διαμαντίδης	6	1212.50	2003-06-02
109	Μαρία	Αθανασίου	1	2787.69	2000-01-26
153	Μαρία	Αλεβιζάτου	2	1321.92	2001-05-15
172	Χρήστος	Βλάσσης	3	1101.70	2000-07-04
189	Θεόδωρος	Αγγελίνας	6	1908.28	2000-06-19
...	...	...	...	...	...

Δείγμα από τα δεδομένα του πίνακα.

Ονόματα πινάκων

- Σε αντίθεση με έναν απλό κειμενογράφο ή ένα λογιστικό φύλλο, οι πίνακες των βάσεων δεδομένων έχουν πάντα όνομα
- Δεν υπάρχει πίνακας σε βάση δεδομένων που να μην έχει όνομα
- Κάθε πίνακας έχει ένα μοναδικό όνομα σε μια βάση δεδομένων: δεν υπάρχουν δύο πίνακες με το ίδιο όνομα στην ίδια βάση δεδομένων
- Ο πίνακας που έχουμε δει έχει το όνομα: **employees**

Το πιο απλό ερώτημα

Να δοθούν όλα τα δεδομένα του πίνακα των υπαλλήλων

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

SELECT Όλα τα ερωτήματα ανάσυρσης δεδομένων αρχίζουν με **SELECT**.

* Το σύμβολο αυτό υποδηλώνει όλα τα πεδία του πίνακα.

FROM Όλα τα ερωτήματα ανάσυρσης δεδομένων έχουν τη δήλωση **FROM** ώστε να καθοριστεί ο πίνακας (ή η παράσταση πίνακα) από όπου θα γίνει η άντληση δεδομένων.

employees Ο πίνακας, τα δεδομένα του οποίου αναζητούμε.

Επιλογή πεδίου από πίνακα

Να βρεθεί το επώνυμο των εργαζομένων

```
1 SELECT lastname  
2 FROM employees;
```

- 1 Το όνομα του πεδίου τοποθετείται μετά το **SELECT**.
- 2 Μπορούμε να βάλουμε οποιοδήποτε πεδίο, αρκεί να:
 - Υπάρχει υπάρχει στον πίνακα που δηλώνεται μετά το **FROM**
 - Παραχθεί από έγκυρη παράσταση

Επιλογή περισσότερων πεδίων από πίνακα

Να βρεθεί το όνομα και το επώνυμο των υπαλλήλων

```
1 SELECT firstname, lastname  
2 FROM employees;
```

firstname	lastname
Νικηφόρος	Διαμαντίδης
Μαρία	Αθανασίου
Μαρία	Αλεβιζάτου
Χρήστος	Βλάσσης
Θεόδωρος	Αγγελίνας
Ηλίας	Μακρής

Κανόνες επιλογής περισσότερων πεδίων

```
1 SELECT firstname, lastname  
2 FROM employees;
```

- Τα ονόματα των πεδίων τοποθετούνται μετά το **SELECT**.
- Χωρίζονται μεταξύ τους με κόμματα.
- Δεν υπάρχει κόμμα μετά το τελευταίο πεδίο.
- Μπορούμε να τοποθετήσουμε τα πεδία με όποια σειρά θέλουμε.
- Τοποθετούμε το όνομα του πεδίου (πχ lastname), όχι τη θέση του στον πίνακα, πχ δεύτερο πεδίο, τέταρτο πεδίο κ.τλ.

Μετονομασία πεδίου στο αποτέλεσμα

Να δοθεί το επώνυμο των εργαζομένων ως στήλη με όνομα `FamilyName`

Εμβέλεια αλλαγής ονόματος πεδίου

ΟΝΟΜΑ αντί για firstname

```
1 SELECT lastname AS FamilyName
2 FROM employees;
```

FamilyName

Αθανασίου
Μαροπούλου
Ρούσσης
.....

- Το πεδίο δηλώνεται κανονικά μετά το **SELECT**.
- Ακολουθεί ο τελεστής μετονομασίας **AS**.
- Η ισχύς του νέου ονόματος περιορίζεται μόνο στο **αποτέλεσμα του ερωτήματος**, δεν αλλάζει κάτι στον ορισμό του πίνακα δεδομένων.

Να βρεθούν οι εργαζόμενοι του τμήματος 1

Περιορισμός

Να βρεθούν οι εργαζόμενοι του τμήματος 1

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

empid	firstname	lastname	depid	salary	hiredate
109	Μαρία	Αθανασίου	1	2787.69	2000-01-26
502	Κρινιώ	Μαροπούλου	1	1754.67	2001-03-07
901	Κυριάκος	Ρούσσης	1	1852.99	2001-11-01

Στη σχεσιακή άλγεβρα

$$\sigma_{depid=1}(employees)$$

Ο όρος **WHERE** σε εντολές **SQL**

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

- Τοποθετείται πάντοτε μετά το **FROM**.
- Είναι προαιρετική, πολλές εντολές **SQL** δεν έχουν καθόλου **WHERE**.
- Αν υπάρχει, ακολουθείται από μια παράσταση που μπορεί να αποτιμηθεί σε **TRUE** ή **FALSE**.
- Μόνο οι εγγραφές για τις οποίες η παράσταση συνθήκης αποδίδει την τιμή **TRUE** εμφανίζονται στο αποτέλεσμα.
- Τα πεδία που ακολουθούν τη δήλωση **WHERE** πρέπει να υπάρχουν στην παράσταση πίνακα που έχει δηλωθεί μετά το **FROM**.

Τελεστές σύγκρισης

Τελεστής	SQL	Σημασία
=	=	ίσο με
≠	! = ή <>	διάφορο από
>	>	μεγαλύτερο από
≥	>=	μεγαλύτερο ή ίσο με
<	<	μικρότερο από
≤	<=	μικρότερο ή ίσο με

Συνδυασμός προβολής και περιορισμού

Να βρεθεί το όνομα, ο μισθός και το τμήμα του υπαλλήλου με κωδικό 243

Συνδυασμός προβολής και περιορισμού

Να βρεθεί το όνομα, ο μισθός και το τμήμα του υπαλλήλου με κωδικό 243

$\Pi_{firstname, lastname, salary, depid}(\sigma_{empid=243}(employees))$

```
1 SELECT firstname, lastname, salary, depid
2 FROM employees
3 WHERE empid = 243;
```

firstname	lastname	salary	depid
Δέσποινα	Παπαδοπούλου	1609.52	2

Αναζήτηση με ημερομηνίες

Να βρεθεί το όνομα, ο μισθός και το τμήμα των υπαλλήλων που προσλήφθηκαν μετά την 1/6/2006

Αναζήτηση με ημερομηνίες

Να βρεθεί το όνομα, ο μισθός και το τμήμα των υπαλλήλων που προσλήφθηκαν μετά την 1/6/2006

$\Pi_{firstname, lastname, salary, depid} (\sigma_{hiredate > '2004-06-01'}(employees))$

```
1 SELECT firstname, lastname, salary, depid
2 FROM employees
3 WHERE hiredate > '2004-06-01';
```

firstname	lastname	salary	depid
Πέτρος	Αρβανιτάκης	1323.80	2
Αντώνης	Παύλου	1231.62	3
Κυριάκος	Ρούσσης	1852.99	1
Βαγγέλης	Χριστόπουλος	NULL	4
Πάυλος	Περίδης	NULL	4
Βασιλική	Περδίκη	1053.54	3

Μαρία – Μαρία

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα Μαρία

Μαρία – Μαρία

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα Μαρία

$\sigma_{firstname='Μαρία'}(employees)$

```
1 SELECT *  
2   FROM employees  
3   WHERE firstname = 'Μαρία';
```

empid	firstname	lastname	depid	salary	hiredate
109	Μαρία	Αθανασίου	1	2787.69	2003-08-29
153	Μαρία	Αλεβιζάτου	2	1321.92	2001-05-15

Σύγκριση με κείμενο

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων, που το επώνυμό τους είναι Παπαδοπούλου και μετά (στη σειρά καταλόγου)

Σύγκριση με κείμενο

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων, που το επώνυμό τους είναι Παπαδοπούλου και μετά (στη σειρά καταλόγου)

$\Pi_{salary}(\sigma_{lastname \geq 'Παπαδοπούλου'}(employees))$

```
1 SELECT *  
2 FROM employees  
3 WHERE lastname >= 'Παπαδοπούλου'
```

empid	firstname	lastname	depid	salary	hiredate
243	Δέσποινα	Παπαδοπούλου	2	1609.52	1999-03-05
412	Αριστείδης	Περρής	3	1320.83	1999-11-12
435	Αντώνης	Πάυλου	3	1231.62	2005-02-05
835	Αθανάσιος	Πετράκης	6	1322.01	2001-02-23
901	Κυριάκος	Ρούσσος	1	1852.99	2006-11-01

Γενικευμένη προβολή

Αλγεβρικές παραστάσεις σε αριθμητικά πεδία

$$salary \times 1.05$$

Η πράξη εφαρμόζεται σε όλες τις εγγραφές του πίνακα:
αύξηση του μισθού κατά 5%.

Λογιστικά φύλλα: «σύρσιμο» τύπου προς τα κάτω.

Γλώσσες προγραμματισμού: βρόγχος επανάληψης.

Εφαρμογή συναρτήσεων

- Αριθμητικές, πχ **SQRT(salary)**
- Κειμένου, πχ **LEFT(firstname, 1)**
- Ημερομηνίας πχ **YEAR(hiredate)**

Γενικευμένη προβολή και SQL

Εφαρμογή σε οποιοδήποτε όρο

- **SELECT**
- **FROM**
- **WHERE**
- **GROUP BY**
- **HAVING**
- **ORDER BY**

Παράδειγμα γενικευμένης προβολής – 1

Να βρεθεί το όνομα και ο νέος μισθός όλων των υπαλλήλων που εργάζονται στο τμήμα 2 μετά από αύξηση 150 € στο μισθό τους

Παράδειγμα γενικευμένης προβολής – 1

Να βρεθεί το όνομα και ο νέος μισθός όλων των υπαλλήλων που εργάζονται στο τμήμα 2 μετά από αύξηση 150 € στο μισθό τους

$\Pi_{firstname, lastname, salary+150}(employees)$

```
1 SELECT firstname, lastname, salary+150
2   FROM employees
3   WHERE depid = 2;
```

firstname	lastname	salary+150
Δέσποινα	Παπαδοπούλου	1759.52
Πέτρος	Αρβανιτάκης	1473.80
Μαριλένα	Κρέσπα	1255.04
Μαρία	Αλεβιζάτου	1471.92

Παράδειγμα γενικευμένης προβολής – 2

Να βρεθεί το όνομα και ο μισθός των υπαλλήλων που ο μισθός τους θα ανέβει περισσότερο από 80 € αν πάρουν αύξηση 4%

Παράδειγμα γενικευμένης προβολής – 2

Να βρεθεί το όνομα και ο μισθός των υπαλλήλων που ο μισθός τους θα ανέβει περισσότερο από 80 € αν πάρουν αύξηση 4%

$\Pi_{lastname,firstname,salary}(\sigma_{salary*0.04>80}(employees))$

```
1 SELECT *  
2 FROM employees  
3 WHERE salary*0.04 > 80;
```

firstname	lastname	salary
-----------	----------	--------

Δέσποινα	Παπαδοπούλου	1609.52
----------	--------------	---------

Αδαμαντία	Αθανασίου	2787.69
-----------	-----------	---------

Θεόδωρος	Αγγελίνας	1908.28
----------	-----------	---------

Υπολογισμοί με ποσοστά

Αν:

- s_1 είναι η αρχική τιμή του μισθού, όπως είναι αποθηκευμένη στο πεδίο *salary* του πίνακα *employees*,
- s_2 είναι η νέα τιμή, αυτή που προκύπτει μετά την μεταβολή,
- v είναι το ποσό της μεταβολής,
- σ είναι η σχετική μεταβολή, και
- π είναι η σχετική ποσοστιαία μεταβολή

τότε ισχύουν οι σχέσεις:

$$s_2 = s_1 + v$$

$$\sigma = \frac{s_2 - s_1}{s_1}$$

$$\pi = \frac{(s_2 - s_1)}{s_1} 100$$

$$s_2 = (1 + \sigma)s_1$$

$$s_2 = \left(1 + \frac{\pi}{100}\right)s_1$$

Παράδειγμα γενικευμένης προβολής – 3

Να βρεθεί ο κωδικός, ο μισθός, ο νέος μισθός των υπαλλήλων μετά από αύξηση 80 € αν ο μισθός μεταβληθεί περισσότερο από 4% με φθίνουσα ταξινόμηση ως προς το ποσοστό αύξησης του μισθού.

Παράδειγμα γενικευμένης προβολής – 3

Να βρεθεί ο κωδικός, ο μισθός, ο νέος μισθός των υπαλλήλων μετά από αύξηση 80 € αν ο μισθός μεταβληθεί περισσότερο από 4% με φθίνουσα ταξινόμηση ως προς το ποσοστό αύξησης του μισθού.

```
1  SELECT empid, salary, salary+80, 80*100/salary
2  FROM employees
3  WHERE salary+80 > salary*1.04
4  ORDER BY 80*100/salary DESC;
```

empid	salary	salary+80	80*100/salary
205	1050.96	1130.96	7.6121
490	1051.92	1131.92	7.6051
...	...	...	...
811	1323.98	1403.98	6.0424
342	1351.96	1431.96	5.9173

ORDER BY column [ASC | DESC]

- 1 Είναι προαιρετική.
- 2 Αν υπάρχει **ORDER BY** στην πρόταση **SQL**, τότε τοποθετείται πάντα στο τέλος.
- 3 Μπορεί να περιέχει ονόματα πεδίων που είτε έχουν, είτε δεν έχουν δηλωθεί στο **SELECT**, αρκεί τα πεδία να υπάρχουν στην έκφραση πίνακα που ακολουθεί το **FROM**.
- 4 Μπορεί να γίνει ταξινόμηση σε ένα πεδίο μετά από κάποιο είδος υπολογισμού (αλγεβρική παράσταση ή κλήση συνάρτησης).
- 5 Αν ο προσδιορισμός αύξουσας (**ASC**) ή φθίνουσας (**DESC**) ταξινόμησης παραληφθεί τότε γίνεται αύξουσα ταξινόμηση (**ASC**).

Ταξινόμηση ως προς το τμήμα

Να βρεθεί το όνομα, ο μισθός, και το τμήμα των υπαλλήλων, με αύξουσα ταξινόμηση ως προς το τμήμα

Ταξινόμηση ως προς το τμήμα

Να βρεθεί το όνομα, ο μισθός, και το τμήμα των υπαλλήλων, με αύξουσα ταξινόμηση ως προς το τμήμα

```
1  SELECT firstname, lastname, salary, depid
2      FROM employees
3  ORDER BY depid;
```

firstname	lastname	salary	depid
Μαρία	Αθανασίου	2787.69	1
Κρινιώ	Μαροπούλου	1754.67	1
Κυριάκος	Ρούσσης	1852.99	1
Δέσποινα	Παπαδοπούλου	1609.52	2
Πέτρος	Αρβανιτάκης	1323.80	2
Μαριλένα	Κρέσπα	1105.04	2

.....

Ταξινόμηση ως προς το μισθό

Να βρεθεί το όνομα και ο μισθός όλων των υπαλλήλων με αύξουσα ταξινόμηση ως προς το μισθό

Ταξινόμηση ως προς το μισθό

Να βρεθεί το όνομα και ο μισθός όλων των υπαλλήλων με αύξουσα ταξινόμηση ως προς το μισθό

```
1  SELECT firstname, lastname, salary
2      FROM employees
3  ORDER BY salary ASC;
```

firstname	lastname	salary
Βαγγέλης	Χριστόπουλος	NULL
Ηρακλής	Μανωλάκης	NULL
Πάυλος	Περίδης	NULL
Βασιλική	Περδίκη	1053.54
Περικλής	Κιτσάκης	1100.13
Κώστας	Παπαδόπουλος	1100.23

.....

Φθίνουσα ταξινόμηση ημερομηνιών

Να βρεθεί το όνομα, ο κωδικός, και η ημερομηνία πρόσληψης των υπαλλήλων που προσελήφθησαν μετά την 1/1/2004, με φθίνουσα ταξινόμηση ως προς την ημερομηνία πρόσληψης

Φθίνουσα ταξινόμηση ημερομηνιών

Να βρεθεί το όνομα, ο κωδικός, και η ημερομηνία πρόσληψης των υπαλλήλων που προσελήφθησαν μετά την 1/1/2004, με φθίνουσα ταξινόμηση ως προς την ημερομηνία πρόσληψης

```
1  SELECT firstname, lastname, empid, hiredate
2      FROM employees
3      WHERE hiredate > '2004-01-01'
4  ORDER BY hiredate DESC;
```

firstname	lastname	empid	hiredate
Βαγγέλης	Χριστόπουλος	230	2007-12-03
Κυριάκος	Ρούσσης	901	2006-11-01
Πέτρος	Αρβανιτάκης	419	2006-07-17
Αντώνης	Παύλου	435	2005-02-05
.....			

Διπλή ταξινόμηση: τμήμα και μισθός

Να βρεθεί ο κωδικός, ο μισθός και το τμήμα των υπαλλήλων με αύξουσα ταξινόμηση ως προς το τμήμα και φθίνουσα ταξινόμηση ως προς το μισθό

Διπλή ταξινόμηση: τμήμα και μισθός

Να βρεθεί ο κωδικός, ο μισθός και το τμήμα των υπαλλήλων με
αύξουσα ταξινόμηση ως προς το τμήμα και φθίνουσα ταξινόμηση ως
προς το μισθό

```
1  SELECT empid, salary, depid
2  FROM employees
3  ORDER BY depid ASC, salary DESC;
```

empid	salary	depid
109	2787.69	1
901	1852.99	1
502	1754.67	1
243	1609.52	2
419	1323.80	2
.....		

Διπλή ταξινόμηση: μισθός και τμήμα

Να βρεθεί ο κωδικός, ο μισθός και το τμήμα των υπαλλήλων με αύξουσα ταξινόμηση ως προς το τμήμα και φθίνουσα ταξινόμηση ως προς το μισθό

Διπλή ταξινόμηση: μισθός και τμήμα

Να βρεθεί ο κωδικός, ο μισθός και το τμήμα των υπαλλήλων με
αύξουσα ταξινόμηση ως προς το τμήμα και φθίνουσα ταξινόμηση ως
προς το μισθό

```
1  SELECT empid, salary, depid
2  FROM employees
3  ORDER BY salary DESC, depid ASC;
```

empid	salary	depid
109	2787.69	1
801	2312.56	3
189	1908.28	6
901	1852.99	1
502	1754.67	1
.....		

Περίληψη μαθήματος

- 1 **SELECT .. FROM .. WHERE** : ανάσυρση δεδομένων.
- 2 Ο όρος **SELECT** πρέπει να ακολουθείται από τουλάχιστον ένα πεδίο.
- 3 Το * υποδηλώνει την επιλογή όλων των πεδίων των πινάκων που δηλώνονται στον όρο FROM.
- 4 Ο όρος **WHERE** τοποθετείται μετά το **FROM**, στην περίπτωση όπου είναι επιθυμητό να γίνει περιορισμός των εγγραφών του αποτελέσματος.
- 5 Αν υπάρχει όρος **WHERE**, τότε μόνο οι εγγραφές για τις οποίες η παράσταση δίνει τιμή **TRUE** εμφανίζονται στο αποτέλεσμα του ερωτήματος.
- 6 Οι ημερομηνίες ή το κείμενο σε παραστάσεις σύγκρισης απαιτούν τη χρήση εισαγωγικών.
- 7 Τα αποτελέσματα ενός ερωτήματος μπορούν να ταξινομηθούν με τον όρο **ORDER BY**.

Απαλοιφή διπλοεγγραφών

Να βρεθεί σε ποια τμήματα απασχολούνται οι υπάλληλοι

Χωρίς απαλοιφή
διπλοεγγραφών

```
1 SELECT depid  
2 FROM employees;
```

depid
1
1
1
2
2
...

Με απαλοιφή
διπλοεγγραφών

```
1 SELECT DISTINCT depid  
2 FROM employees;
```

depid
1
2
3
4
5
6

Συνδυασμός πεδίων

Να δοθεί το τμήμα και ο μισθός των υπαλλήλων

```
1 SELECT DISTINCT depid, salary
2 FROM employees;
```

depid	salary
6	1212.50
1	2787.69
2	1321.92
3	1101.70
6	1908.28
...	...

- Εδώ ελέγχεται ο συνδυασμός των τιμών στα πεδία **depid** και **salary**
- Έτσι, μια τιμή στο πεδίο **depid** μπορεί να εμφανιστεί δεύτερη (ή πολλαπλή) επειδή αντιστοιχίζεται με διαφορετική τιμή στο πεδίο **salary**
- Αν το ζεύγος τιμών τμήμα-μισθός είναι ίδιο για δύο διαφορετικές εγγραφές τότε θα εμφανιστεί στο αποτέλεσμα μόνο μία φορά (λόγω **DISTINCT**)

DISTINCT και προβολή στη σχεσιακή άλγεβρα

Η σχεσιακή πράξη της προβολής, πχ:

$$\Pi_{depid}(employees)$$

εκφράζεται στην SQL πάντα με DISTINCT:

```
1 SELECT DISTINCT depid  
2 FROM employees;
```

και όχι χωρίς αυτό, πχ:

```
1 SELECT depid  
2 FROM employees;
```

Ο λόγος είναι απλός: η σχεσιακή άλγεβρα αφορά πράξεις συνόλων, οπότε δε μπορεί να υπάρχουν διπλοεγγραφές, τα στοιχεία ενός συνόλου είναι μοναδικά.

Και κάτι για εξάσκηση

Να βρεθούν τα τμήματα στα οποία ο μισθός των υπαλλήλων θα ανέβει πάνω από 80 € αν πάρουν αύξηση 3.5%

$\Pi_{depid}(\sigma_{salary*0.035>80}(employees))$

```
1 SELECT DISTINCT depid
2   FROM employees
3  WHERE salary*0.035 > 80;
```

depid
1
3

Μόνο στα τμήματα 1 και 3 υπάρχουν υπάλληλοι που αν ο μισθός τους αυξηθεί κατά 3.5% η αύξηση θα είναι μεγαλύτερη από 80 €. Στα τμήματα αυτά, πιθανά να υπάρχουν περισσότεροι από ένας υπάλληλος με τέτοια αύξηση. Αλλά αυτό που ενδιαφέρει είναι σε **ποιο τμήμα**, όχι σε ποιους και πόσους υπαλλήλους.

Χειρισμός τιμών NULL

- 1 Δε μπορούμε να χειριστούμε τιμές **NULL** με τελεστές σύγκρισης
- 2 Μια τιμή είναι (**IS**) ή δεν είναι (**IS NOT**) **NULL**
- 3 Ο έλεγχος για τιμές **NULL** αποδίδει πάντοτε **TRUE** ή **FALSE**
- 4 Δύο τιμές **NULL** δεν είναι ίσες μεταξύ τους, η μεταξύ τους σύγκριση δεν έχει νόημα

Παραδείγματα με NULL

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων χωρίς καταχωρημένο μισθό

```
1 SELECT *  
2     FROM employees  
3     WHERE salary IS NULL;
```

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με καταχωρημένο μισθό

```
1 SELECT *  
2     FROM employees  
3     WHERE salary IS NOT NULL;
```

NULL και DISTINCT

Χωρίς DISTINCT

```
1  SELECT salary
2      FROM employees
3      WHERE salary IS NULL;
4  | salary |
5  -----
6  |  NULL  |
7  |  NULL  |
8  |  NULL  |
```

Με DISTINCT

```
1  SELECT DISTINCT salary
2      FROM employees
3      WHERE salary IS NULL;
4  | salary |
5  -----
6  |  NULL  |
```

Λάθη στον έλεγχο για NULL

Με τον τελεστή =

```
1 SELECT *  
2   FROM employees  
3   WHERE salary = NULL;
```

```
4  
5 Empty set (0.00 sec)
```

Ως κείμενο

```
1 SELECT *  
2   FROM employees  
3   WHERE salary = 'NULL';
```

```
4  
5 Empty set, 1 warning (0.00 sec)
```

NULL και αλγεβρικές πράξεις

Πρόσθεση 1000 € στο μισθό των υπαλλήλων χωρίς μισθό

```
1  SELECT empid, salary, salary+1000
2      FROM employees
3      WHERE salary IS NULL;
```

empid	salary	salary+1000
230	NULL	NULL
483	NULL	NULL
593	NULL	NULL

Ταίριασμα κειμένου

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα Νίκη

```
1 SELECT *  
2     FROM employees  
3     WHERE firstname = 'Νίκη';
```

Μπορούμε να κάνουμε συγκρίσεις αλφαριθμητικών με τελεστές σύγκρισης όπως και με αριθμούς.
Ωστόσο, το αλφαριθμητικό τοποθετείται πάντα μέσα σε εισαγωγικά.

Τελεστές ανισότητας και σύγκριση κειμένου

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό μετά τη Νίκη

```
1 SELECT *  
2     FROM employees  
3     WHERE firstname > 'Νίκη';
```

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα μετά το Ν

```
1 SELECT *  
2     FROM employees  
3     WHERE firstname > 'N';
```

Ταίριασμα κειμένου με τον τελεστή LIKE

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα που αρχίζει από N

```
1 SELECT *  
2     FROM employees  
3     WHERE firstname LIKE 'N%';
```

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα που αρχίζει από A και τελειώνει σε A

```
1 SELECT *  
2     FROM employees  
3     WHERE firstname LIKE 'A%A';
```

LIKE: χαρακτήρες υποκατάστασης

- % (ή *) : υποκαθιστά από κανένα ως πολλούς χαρακτήρες
- _ (ή ?) : υποκαθιστά ακριβώς ένα χαρακτήρα
- Οι χαρακτήρες υποκατάστασης μπορούν αν συνδυαστούν μεταξύ τους

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα που έχει 5 χαρακτήρες και τελειώνει σε A

```
1 SELECT *  
2     FROM employees  
3     WHERE firstname LIKE '____A';
```

Ασκήσεις με ταίριασμα κειμένου

- 1 Να βρεθούν οι υπάλληλοι που το όνομά τους αρχίζει από Α.
- 2 Να βρεθούν οι υπάλληλοι που το όνομά τους τελειώνει σε ς.
- 3 Να βρεθούν οι υπάλληλοι που το όνομά τους έχει δεύτερο χαρακτήρα το α.
- 4 Να βρεθούν οι υπάλληλοι που το όνομά τους έχει δεύτερο χαρακτήρα το α και τελειώνει σε ς.

Ο τελεστής AND

- 1 Δύο παραστάσεις στον όρο **WHERE** μπορούν να συνδυαστούν μεταξύ τους με το λογικό τελεστή **AND**.
- 2 Χρησιμοποιούμε τον τελεστή **AND** όταν θέλουμε να ελέγξουμε την τιμή αληθείας και των δύο παραστάσεων ταυτόχρονα.
- 3 Οι βάσεις δεδομένων ακολουθούν την τριαδική λογική: **TRUE**, **FALSE**, **UNK**.
- 4 Στο αποτέλεσμα του ερωτήματος εισέρχονται μόνο οι εγγραφές που αποδίδουν **TRUE**.

Πίνακας αληθείας AND

AND	TRUE	FALSE	UNK
TRUE	TRUE	FALSE	UNK
FALSE	FALSE	FALSE	FALSE
UNK	UNK	FALSE	UNK

Ο τελεστής **AND** αποδίδει **TRUE**
μόνο όταν και οι δύο παραστάσεις είναι **TRUE**

Ένα παράδειγμα με τον τελεστή AND

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων που εργάζονται στο τμήμα 4 και ο μισθός τους είναι μεγαλύτερος από 1100 €

```
1  SELECT *  
2      FROM employees  
3      WHERE depid = 4  
4          AND salary > 1100;
```

empid	firstname	lastname	depid	salary	hiredate
206	Νίκος	Βλάχος	4	1102.04	2002-12-03
311	Νίκος	Στεργιόπουλος	4	1386.05	2002-02-01

Ο τελεστής OR

- 1 Δύο παραστάσεις στον όρο **WHERE** μπορούν να συνδυαστούν μεταξύ τους με το λογικό τελεστή **OR**.
- 2 Χρησιμοποιούμε τον τελεστή **OR** όταν θέλουμε να ελέγξουμε την τιμή αληθείας τουλάχιστον μιας από τις δύο παραστάσεις.
- 3 Οι βάσεις δεδομένων ακολουθούν την τριαδική λογική: **TRUE**, **FALSE**, **UNK**.
- 4 Στο αποτέλεσμα του ερωτήματος εισέρχονται μόνο οι εγγραφές που αποδίδουν **TRUE**.

Πίνακας αληθείας OR

OR	TRUE	FALSE	UNK
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNK
UNK	TRUE	UNK	UNK

Ο τελεστής **OR** αποδίδει **TRUE** όταν μία τουλάχιστον παράσταση είναι **TRUE**.

Ένα παράδειγμα με τον τελεστή OR

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων που εργάζονται στο τμήμα 4 ή ο μισθός τους είναι μεγαλύτερος από 1100 €

```
1 SELECT *  
2   FROM employees  
3   WHERE depid = 4  
4         OR salary >= 1100;
```

empid	firstname	lastname	depid	salary	hiredate
.....					
206	Νίκος	Βλάχος	4	1102.04	2002-12-03
230	Βαγγέλης	Χριστόπουλος	4	NULL	2002-12-03
234	Αδαμαντία	Θεοτοκάτου	5	1534.65	1999-10-01
243	Δέσποινα	Παπαδοπούλου	2	1609.52	1999-03-05
311	Νίκος	Στεργιόπουλος	4	1386.05	2002-02-01

Μακριά από παγίδες

Να βρεθούν τα ονοματεπώνυμα των υπαλλήλων που εργάζονται στα τμήματα 1 και 3

Λάθος απάντηση

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

Σωστή απάντηση

```
1 SELECT firstname, lastname
2   FROM employees
3   WHERE depid = 1
4         OR depid = 3;
```

Σύνθετες παραστάσεις και παρενθέσεις

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων που δεν προσλήφθηκαν το 2002 και εργάζονται στο τμήμα 3

Σύνθετες παραστάσεις και παρενθέσεις

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων που δεν προσλήφθηκαν το 2002 και εργάζονται στο τμήμα 3

Λάθος απάντηση

```
1 SELECT *  
2   FROM employees  
3   WHERE depid = 3  
4     AND hiredate < '2002-01-01'  
5     OR hiredate > '2002-12-31' ;
```

Σωστή απάντηση

```
1 SELECT *  
2   FROM employees  
3   WHERE depid = 3  
4     AND ( hiredate < '2002-01-01' OR  
5         hiredate > '2002-12-31' );
```

Ο τελεστής συνόλου IN

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων που εργάζονται στα τμήματα 2, 3 και 4

```
1 SELECT *  
2     FROM employees  
3     WHERE depid = 2  
4         OR depid = 3  
5         OR depid = 4;
```

Το ίδιο, αλλά πιο απλά

```
1 SELECT *  
2     FROM employees  
3     WHERE depid IN (2,3,4);
```

Άρνηση του IN

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων που δεν εργάζονται στα τμήματα 2, 3 και 4

```
1 SELECT *  
2     FROM employees  
3     WHERE depid NOT IN (2,3,4);
```

Ή, ισοδύναμα με:

```
1 SELECT *  
2     FROM employees  
3     WHERE depid != 2  
4         AND depid != 3  
5         AND depid != 4;
```