

Τα πρώτα ερωτήματα SQL

Αθανάσιος Σταυρακούδης

<http://stavrakoudis.econ.uoi.gr>

Άνοιξη 2016



Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.



Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.
- Είναι μια διαδικαστική γλώσσα δεδομένων με δύο μέρη:
 - α) **DDL**: γλώσσα ορισμού δεδομένων
 - β) **DML**: γλώσσα χειρισμού δεδομένων.



Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.
- Είναι μια διαδικαστική γλώσσα δεδομένων με δύο μέρη:
 - α) **DDL**: γλώσσα ορισμού δεδομένων
 - β) **DML**: γλώσσα χειρισμού δεδομένων.
- Πρώτη εμπορική υλοποίηση ως System R από την IBM (1970s).



Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.
- Είναι μια διαδικαστική γλώσσα δεδομένων με δύο μέρη:
 - α) **DDL**: γλώσσα ορισμού δεδομένων
 - β) **DML**: γλώσσα χειρισμού δεδομένων.
- Πρώτη εμπορική υλοποίηση ως System R από την IBM (1970s).
- Η εταιρεία Relational Software Inc. κατασκευάζει το λογισμικό που αργότερα θα γίνει γνωστό ως **Oracle**



Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.
- Είναι μια διαδικαστική γλώσσα δεδομένων με δύο μέρη:
 - α) **DDL**: γλώσσα ορισμού δεδομένων
 - β) **DML**: γλώσσα χειρισμού δεδομένων.
- Πρώτη εμπορική υλοποίηση ως System R από την IBM (1970s).
- Η εταιρεία Relational Software Inc. κατασκευάζει το λογισμικό που αργότερα θα γίνει γνωστό ως **Oracle**
- **πόλεμος πάντων μὲν πατήρ ἐστι (Ηράκλειτος)**



Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.
- Είναι μια διαδικαστική γλώσσα δεδομένων με δύο μέρη:
 - α) **DDL**: γλώσσα ορισμού δεδομένων
 - β) **DML**: γλώσσα χειρισμού δεδομένων.
- Πρώτη εμπορική υλοποίηση ως System R από την IBM (1970s).
- Η εταιρεία Relational Software Inc. κατασκευάζει το λογισμικό που αργότερα θα γίνει γνωστό ως **Oracle**
- **πόλεμος πάντων μὲν πατήρ ἐστι (Ηράκλειτος)**
- **Dialects: SQL-86, SQL-89, SQL-99, ..., SQL-2011, ...**

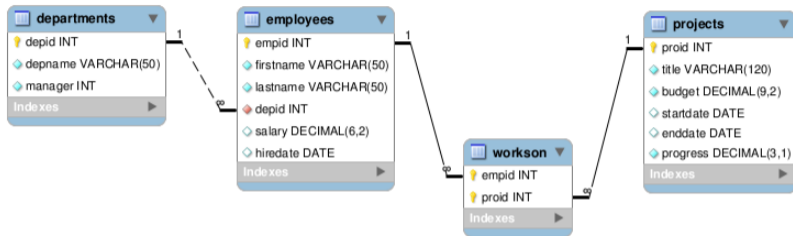


Δυο λόγια για την SQL

- Υλοποίηση του σχεσιακού μοντέλου δεδομένων του E.F. Codd, αρχικά ως DSI/Alpha, αργότερα ως **SQUARE** και **SEQUEL**.
- Είναι μια διαδικαστική γλώσσα δεδομένων με δύο μέρη:
 - α) **DDL**: γλώσσα ορισμού δεδομένων
 - β) **DML**: γλώσσα χειρισμού δεδομένων.
- Πρώτη εμπορική υλοποίηση ως System R από την IBM (1970s).
- Η εταιρεία Relational Software Inc. κατασκευάζει το λογισμικό που αργότερα θα γίνει γνωστό ως **Oracle**
- **πόλεμος πάντων μὲν πατήρ ἐστι (Ηράκλειτος)**
- **Dialects: SQL-86, SQL-89, SQL-99, ..., SQL-2011, ...**
- **NoSQL : Not only SQL**



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.

Δείτε τα πλήρη περιεχόμενα εδώ:

<http://stavrakoudis.econ.uoi.gr/stavrakoudis/?iid=400>



Όνόματα 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**



Πρόσβαση στη βάση company

Πρόσβαση στη βάση company

- Διαδικτυακή πρόσβαση από: <http://bit.ly/SQLcmp>
 - **username: dbuser**
 - **password: econ98**
- Διαθέσιμη για μεταφόρτωση από την ιστοσελίδα <http://bit.ly/xxy896>
σε μορφή:
 - **MS Access**
 - **Open/Libre Office**
 - **SQL installation script**
 - **csv files**



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`



Μετονομασία πεδίου στο αποτέλεσμα

Να δοθεί το επώνυμο των εργαζομένων ως στήλη με όνομα FamilyName

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



Μετονομασία πεδίου στο αποτέλεσμα

Να δοθεί το επώνυμο των εργαζομένων ως στήλη με όνομα FamilyName

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

FamilyName

Αθανασίου
Μαροπούλου
Ρούσσης
.....



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



Να βρεθούν οι εργαζόμενοι του τμήματος 1

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



Να βρεθούν οι εργαζόμενοι του τμήματος 1

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

empid	firstname	lastname	deptid	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



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



Απλές σχεσιακές πράξεις

Προβολή

Επιλογή μέρους από το σύνολο των γνωρισμάτων της σχέσης, πχ:

$$\Pi_{lastname, salary}(employees)$$

Περιορισμός

Επιλογή μέρους από τις πλειάδες της σχέσης, πχ:

$$\sigma_{empid=243}(employees)$$

Συνδυασμός

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


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

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



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;
```



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

```
Π firstname,lastname,salary,deptid ( $\sigma_{hiredate>'2004-06-01'}$  (employees))
```



Αναζήτηση με ημερομηνίες

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

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

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



Αναζήτηση με ημερομηνίες

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

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

```
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
Παύλος	Περίδης	NULL	4
Βασιλική	Περδίκη	1053.54	3



Μαρία – Μαρία

Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα Μαρία



Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα Μαρία

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



Να βρεθούν όλες οι λεπτομέρειες των υπαλλήλων με μικρό όνομα Μαρία

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

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

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



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'
```



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	Παύλου	3	1231.62	2005-02-05
835	Αθανάσιος	Πετράκης	6	1322.01	2001-02-23
901	Κυριάκος	Ρούσσος	1	1852.99	2006-11-01



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.



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)**



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

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



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;
```



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

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

```
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%

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



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



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;
5
```



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]



ORDER BY column [ASC | DESC]

- 1 Είναι προαιρετική.



ORDER BY column [ASC | DESC]

- 1 Είναι προαιρετική.
- 2 Αν υπάρχει **ORDER BY** στην πρόταση **SQL**, τότε τοποθετείται πάντα στο τέλος.



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**.



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).



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;
4
```



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;
4
```



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;
5
```



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;
4
```



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;
```

```
4
5 empid salary depid
```

```
6 -----
```

```
7 109 2787.69 1
```

```
8 901 1852.99 1
```

```
9 502 1754.67 1
```

```
10 243 1609.52 2
```

```
11 419 1323.80 2
```

```
12 .....
```



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;
```



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;
```

```
4
5 empid salary depid
```

```
6 -----
7 109 2787.69 1
```

```
8 801 2312.56 3
```

```
9 189 1908.28 6
```

```
10 901 1852.99 1
```

```
11 502 1754.67 1
```

```
12 .....
```



Περίληψη μαθήματος

- 1 **SELECT .. FROM .. WHERE** : ανάσυρση δεδομένων.



Περίληψη μαθήματος

- 1 **SELECT .. FROM .. WHERE** : ανάσυρση δεδομένων.
- 2 Ο όρος **SELECT** πρέπει να ακολουθείται από τουλάχιστον ένα πεδίο.



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.



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.



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.



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.



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**.



Σας ευχαριστώ για την προσοχή σας

Είμαι στη διάθεσή σας για σχόλια, απορίες και ερωτήσεις

