

SQL: το μεγάλο όπλο των ΒΔ

SQL: μια ειδική «γλώσσα»

- Σχεσιακή άλγεβρα: πλήρης συμβολισμός για όλες τις ενέργειες
 - Αφηρημένος όμως
- Χρειάζεται μια formal γλώσσα για την επεξεργασία σε υπολογιστή

SQL

Structured Query Language

- Το εμπορικό standard
- Όλοι το ακολουθούν

SQL: μια μακρά ιστορία

- αρχική έκδοση: San Jose Research Laboratory της IBM
- Πρώτη εκδοχή: **Sequel** (αρχές δεκαετίας 70)
- Τεράστια εξέλιξη – νέο όνομα SQL (Structured Query Language, Γλώσσα Δομημένων Ερωτημάτων)
- 1986: αποδεκτό σαν standard από το American National Standards Institute (ANSI)
- Όλα τα εμπορικά προϊόντα την υποστηρίζουν

Τι θέλουμε εδώ

- Βασικές αρχές
- Κανόνες και δομή των εντολών
- ΔΕΝ ΧΡΕΙΑΖΕΤΑΙ απομνημόνευση – υπάρχουν τα manuals 😊
- Τι δουλεύει – τι είναι λάθος

Βασικοί κανόνες (1/2)

- Συγκεκριμένη δομή – η εντολή με λάθη δεν τρέχει
- Λέξεις reserved: δεν μπορούν να χρησιμοποιηθούν για ονόματα πινάκων κτλ (SELECT, INSERT, FROM, ...)
- Λάθη: πάντα υπάρχει εξήγηση – όχι πάντα η «σωστή»
- Θέλουμε να μάθουμε να τα «διαβάζουμε»
- Δεν έχουν σημασία κεφαλαία και μικρά γράμματα
- Συνήθης κώδικας: με κεφαλαία τα τμήματα των εντολών, με μικρά τα υπόλοιπα

Βασικοί κανόνες (2/2)

- Υπάρχουν διάφορα μικρά «τουβλάκια»
- Χρησιμοποιούνται όπου χρειάζεται η δουλειά που κάνουν
- WHERE <συνθήκη> : όπου ισχύει η συνθήκη κάνε ...
- Χρησιμοποιείται στην SELECT, την UPDATE, την DELETE
- Η εντολή μπορεί να γραφεί σε μια γραμμή ή σε πολλά μικρά κομμάτια
- Δεν έχει καμιά διαφορά, μέχρι να «πούμε» ότι η εντολή τελείωσε και πρέπει να τρέξει (συνήθως με ;)

Εντολή Select

- Δομή της εντολής:
Select <λίστα εκφράσεων>
From <λίστα πινάκων>
[Where <συνθήκη>]
[Order By <λίστα πεδίων>]
[Group By <λίστα πεδίων>
[Having <συνθήκη>]]
- Απαραίτητα μόνο τα πρώτα
- Η σειρά είναι συγκεκριμένη

Εντολή Select

- Λειτουργία:
 - FROM: πάρε αυτούς τους πίνακες (θυμηθείτε: πρώτα η παρένθεση)
 - WHERE: κράτησε μόνο αυτές τις γραμμές
 - SELECT: Επίλεξε αυτές τις εκφράσεις
- Δεν αλλάζει κάτι στον αποθηκευμένο πίνακα
- Σχηματίζεται ένας νέος πίνακας
- «Εξαφανίζεται» μετά το κλείσιμο
- Μπορεί όμως να ξανασχηματιστεί τρέχοντας την εντολή
- Εκφράσεις: χαρακτηριστικά, συναρτήσεις, απλές τιμές
- Όλα τα χαρακτηριστικά: *

Εντολή Select: συντομεύσεις

- IN: «ανήκει» σε ένα σύνολο με συγκεκριμένες τιμές
 - Acc-no IN (1, 4, 8)
- Μία από τις **συγκεκριμένες** τιμές
- Ισοδύναμο με Acc-no = 1 OR Acc-no = 4 OR ...
- BETWEEN: τιμές σε ένα διάστημα
 - Balance BETWEEN 100 AND 200
- LIKE: ομοιότητα με μια σειρά χαρακτήρων (% για πολλούς, _ για ένα)
 - Cust-name LIKE 'Πέτ%'
 - Cust-name LIKE 'Πέτρ_ _'
 - Cust-name LIKE '%τρ_ _'

Εντολή Select: τεχνικές λεπτομέρειες

- Σχεδόν πάντα μονά εισαγωγικά- πάντα πρέπει να κλείνουν
- Λογικοί τελεστές AND, OR, NOR, NOT, NAND, XOR
- Λίστες πάντα σε παρένθεση
- Διαχωρισμός λιστών με κόμματα (,)
- Παρενθέσεις: βοηθούν – δεν είναι απαραίτητες – δεν κάνουν ζημιά

Εντολή Select: παραδείγματα

- Ονόματα καταστημάτων της Αθήνας

```
SELECT br-name
```

```
FROM branch
```

```
WHERE br-city = 'Αθήνα'
```

- Υπόλοιπα λογαριασμών και ονόματα των πελατών Παύλου και Πέτρου

```
SELECT balance, cust-name
```

```
FROM deposit
```

```
WHERE (cust-name = 'Παύλου' OR cust-name = 'Πέτρου' )
```

Εντολή Select: παραδείγματα

- Αριθμοί δανείων και τόκοι για κάθε δάνειο με ποσό μεταξύ 10000 και 20000

```
SELECT loan-no, 0.08*amount
```

```
FROM borrow
```

```
WHERE amount>10000 AND amount < 20000
```

- Όλα τα στοιχεία των πελατών από την Αθήνα, την Λάρισα και την Πάτρα

```
SELECT *
```

```
FROM Customer
```

```
WHERE br-city IN ('Αθήνα', 'Λάρισα', 'Πάτρα')
```

Εντολή Select: βγάζει πίνακα

- Η select δεν αλλάζει κάτι στα αποθηκευμένα δεδομένα

Αλλά

- Δημιουργεί έναν καινούριο πίνακα
- Διαθέσιμος όσο δουλεύουμε με αυτόν
- Μπορεί να χρησιμοποιηθεί σε άλλη select

Παραδείγματα

- Είδαμε «Όλα τα στοιχεία των πελατών από την Αθήνα, την Λάρισα και την Πάτρα»
- «Μικρή» αλλαγή: δεν ξέρουμε τις πόλεις (αλλά μπορούμε να τις βρούμε)

Όλα τα στοιχεία των πελατών από πόλεις που έχουν κατάστημα «Κέντρο»

```
SELECT *  
FROM Customer  
WHERE cust-city IN ( SELECT br-city FROM branch WHERE br-name =  
'Κέντρο')
```

Παραδείγματα

- Ο πίνακας που υπολογίστηκε στην παρένθεση μπορεί να «μεταφέρεται» όπου χρειαστεί
- Προσοχή: πρέπει ο πίνακας να είναι συμβατός με τη δουλειά που κάνει
- Αν ψάχνουμε μια λίστα τιμών πόλεων, πρέπει να φροντίσουμε η εντολή στην παρένθεση να βγάζει κάτι τέτοιο
- Π.χ. όχι

```
SELECT * FROM Customer  
WHERE cust-city IN ( SELECT br-city, br-name FROM branch WHERE br-name = 'Κέντρο' )
```
- Δεν μπορώ να ψάξω **μια** μεταβλητή σε **ζευγάρια** τιμών

Διάταξη

- Νέα σειρά των γραμμών με βάση συγκεκριμένα πεδία
ORDER BY acc-no
- Διατάσσουμε τις γραμμές με αύξουσα σειρά του πεδίου acc-no
- Προφανώς η γραμμή ακολουθεί ολόκληρη 😊

Cust-name	Acc-no	balance
Πέτρου	158	1240
Παύλου	217	957
Αθανασίου	93	10002
Χρήστου	421	957



Cust-name	Acc-no	balance
Αθανασίου	93	10002
Πέτρου	158	1240
Παύλου	217	957
Χρήστου	421	957

Παραδείγματα

- Μπορώ να υπολογίσω και μία μοναδική τιμή και να την χρησιμοποιήσω σε συνθήκες
- Κωδικοί προμηθευτή με στάτους μεγαλύτερο από 2 (εύκολο)
`SELECT S# FROM S WHERE Status > 2`
- Τροποποίηση: Κωδικοί προμηθευτή με στάτους μεγαλύτερο από αυτό του S1 (θα χρειαστεί να το υπολογίσουμε)
- Ποιο είναι το στάτους του S1? → `SELECT Status FROM S WHERE S# = 'S1'`
Άρα τελικά
`SELECT S# FROM S WHERE Status > (SELECT Status FROM S WHERE S# = 'S1')`

διάταξη

- Περισσότερα πεδία
ORDER BY balance, acc-no
- Πρώτα διάταξη με το πρώτο πεδίο

Cust-name	Acc-no	balance
Πέτρου	158	1240
Παύλου	421	957
Αθανασίου	93	10002
Χρήστου	217	957



Cust-name	Acc-no	balance
Παύλου	421	957
Χρήστου	217	957
Πέτρου	158	1240
Αθανασίου	93	10002

- Αν υπάρχει ισοπαλία, η σειρά παραμένει όπως ήταν
- Μετά κρίνεται με βάση το δεύτερο

Διάταξη

Cust-name	Acc-no	balance
Χρήστου	217	957
Παύλου	421	957
Πέτρου	158	1240
Αθανασίου	93	10002

Προσοχή: μόνο οι γραμμές που είναι ισόπαλες επηρεάζονται

Διάταξη φθίνουσα

ORDER BY balance DESC

- Η σειρά αντιστρέφεται
- Η default σειρά είναι η αύξουσα (ASC, αλλά δεν χρειάζεται)
- Κάθε πεδίο μπορεί να έχει διαφορετική σειρά

ORDER BY balance DESC, cust-name ASC, acc-no

- Πρώτα με βάση το υπόλοιπο με φθίνουσα σειρά, αν έχουμε το ίδιο υπόλοιπο με αλφαβητική σειρά του ονόματος, αν έχουμε και το ίδιο όνομα με αύξουσα σειρά του αρ. λογαριασμού

Διάφορα παραδείγματα

- Ονόματα πελατών από Αθήνα και Θεσσαλονίκη με αλφαβητική σειρά
SELECT Cust-name FROM Customer
WHERE cust-city = 'Αθήνα' OR Cust-city = 'Θεσσαλονίκη'
ORDER BY Cust-name
- Εναλλακτικά WHERE cust-city IN ('Αθήνα' , 'Θεσσαλονίκη')
- Μικρή παραλλαγή: με αλφαβητική σειρά, αλλά πρώτα τους Θεσσαλονικείς και μετά τους Αθηναίους → Πρέπει να διατάξουμε πρώτα ως προς την πόλη, και μάλιστα με φθίνουσα σειρά!
SELECT Cust-name FROM Customer
WHERE cust-city = 'Αθήνα' OR Cust-city = 'Θεσσαλονίκη'
ORDER BY Cust-city DESC, Cust-name
- Προσοχή: δεν χρειάζεται τα πεδία του ORDER BY να εμφανίζονται και στο SELECT
- Το ίδιο ισχύει και για το WHERE

Διάφορα παραδείγματα

- Αρ. Δανείου και κατάστημα για δάνεια του πελάτη Πέτρου με ποσό μεταξύ 10000 και 20000 με φθίνουσα σειρά υπολοίπου

```
SELECT loan-no, br-name FROM Borrow
```

```
WHERE cust-name = 'Πέτρου' AND (amount BETWEEN 10000 AND 20000)
```

```
ORDER BY amount DESC
```

- Εναλλακτικά (amount <20000 AND amount >10000)
- Εδώ συμβαίνει να μην εμφανίζονται στο SELECT ούτε τα πεδία του WHERE ούτε το πεδίο του ORDER BY