

Εργαστήριο Βάσεων Δεδομένων 2010-2011

Μάθημα 1^ο

Σκοπός του 1^{ου} εργαστηριακού μαθήματος

Σκοπός του πρώτου εργαστηριακού μαθήματος είναι να μελετήσουμε ερωτήματα επιλογής δεδομένων από μία μόνο σχέση.

Μορφή ενός απλού SQL ερωτήματος

SELECT A_1, A_2, \dots, A_n

FROM R_1, R_2, \dots, R_m

WHERE Condition



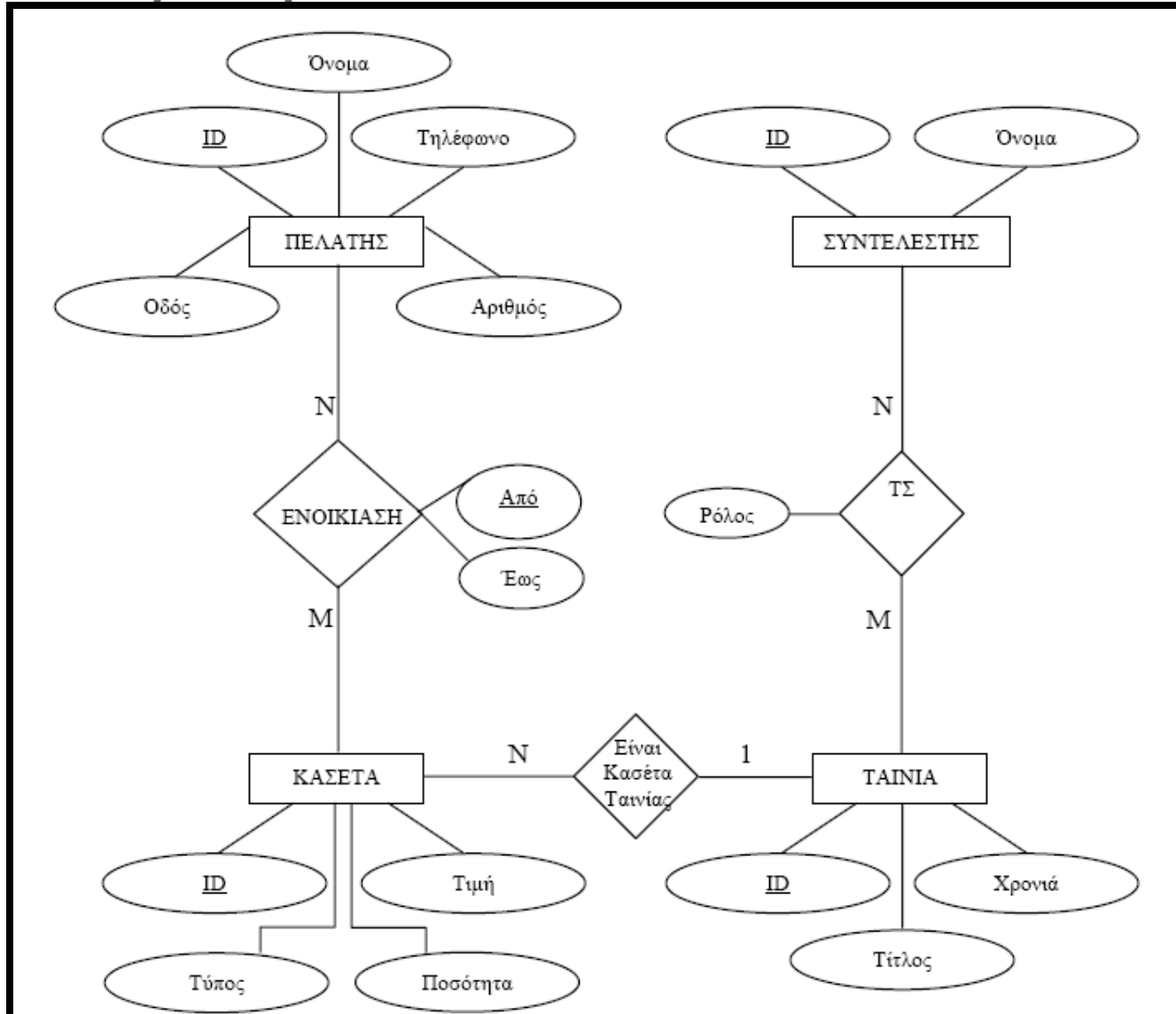
Προαιρετικό

- **SELECT:** τα χαρακτηριστικά που μας ενδιαφέρουν
- **FROM:** το σύνολο των σχέσεων στις οποίες γίνεται η αναζήτηση για να σχηματιστεί το τελικό αποτέλεσμα
- **WHERE:** περιορισμοί που πρέπει να ικανοποιεί το αποτέλεσμα

Σειρά εκτέλεσης των εντολών

1. Σχηματίζεται το καρτεσιανό γινόμενο των σχέσεων R_1, R_2, \dots, R_m της **FROM**
2. Επιλέγονται πλειάδες από το καρτεσιανό γινόμενο που ικανοποιούν την συνθήκη της **WHERE**
3. Προβάλλονται στο αποτέλεσμα μόνο τα χαρακτηριστικά που βρίσκονται στην εντολή **SELECT**
4. Το αποτέλεσμα του ερωτήματος είναι ένα σύνολο πλειάδων της μορφής $\langle A_1, A_2, \dots, A_n \rangle$

Το σχήμα της βάσης πάνω στην οποία θα δουλέψουμε



Οι σχέσεις του σχήματος(1)

Πελάτης: περιλαμβάνει τα στοιχεία των πελατών του video club.

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

Οι σχέσεις του σχήματος(2)

Ταινία: περιλαμβάνει τις ταινίες που υπάρχουν διαθέσιμες στο video-club

<u>ID</u>	Τίτλος	Χρονιά
1	Rear Window	1954
2	Psycho	1960
3	Ben-Hur	1959

Οι σχέσεις του σχήματος(3)

Συντελεστής: περιλαμβάνει τους συντελεστές μίας ταινίας

<u>ID</u>	Όνομα
1	Alfred Hitchcock
2	Grace Kelly
3	Anthony Perkins

Οι σχέσεις του σχήματος(4)

Κασέτα: αντιστοιχεί σε ένα μέσο αποθήκευσης. DVD ή VHS, μίας ταινίας

<u>ID</u>	<u>IDΤαινίας</u>	Τύπος	Ποσότητα	Τιμή
1	1	VHS	3	2
2	1	DVD	1	3
3	2	VHS	2	2

Οι σχέσεις του σχήματος(5)

Ενοικίαση: περιλαμβάνει ποιοι πελάτες νοίκιασαν ποιες κασέτες καθώς και τις ημερομηνίες ενοικίασης και επιστροφής των κασετών

<u>IDΠελάτη</u>	<u>IDΚασέτας</u>	Από	Εώς
1	1	07/10/2006	09/10/2006
1	2	09/20/2006	11/20/2006
2	1	09/10/2006	NULL

Οι σχέσεις του σχήματος(6)

ΤΣ: περιλαμβάνει ποιοι συντελεστές συμμετείχαν σε ποιες ταινίες καθώς και τον ρόλο συμμετοχής αυτών

<u>IDΤαινίας</u>	<u>IDΣυντελεστή</u>	Ρόλος
1	1	Σκηνοθέτης
1	2	Ηθοποιός
2	1	Σκηνοθέτης
2	3	Ηθοποιός

Δύο απλά ερωτήματα

Ερώτημα: Να δοθούν τα ονόματα όλων των πελατών

```
SELECT Όνομα  
FROM ΠΕΛΑΤΗΣ
```

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

Δύο απλά ερωτήματα

Ερώτημα: Να δοθούν τα τηλέφωνα όλων των πελατών

```
SELECT Τηλέφωνο  
FROM ΠΕΛΑΤΗΣ
```

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

Η εντολή DISTINCT

- Στο προηγούμενο ερώτημα το τηλ. 246801 εμφανίζεται δύο φορές (δόθηκαν οι παρακάτω πλειάδες { < 246801 >, < 246801 >, < 987654 >} ως απάντηση)
- Με την εντολή DISTINCT δηλώνουμε πως το επιθυμητό αποτέλεσμα ΔΕΝ θα περιλαμβάνει δύο όμοιες πλειάδες.
- Πότε δύο πλειάδες είναι όμοιες?

DISTINCT Παράδειγμα-1

```
SELECT DISTINCT Τηλέφωνο  
FROM ΠΕΛΑΤΗΣ
```

{ < 246801 >, ~~< 246801 >~~, < 987654 > }


<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

DISTINCT Παράδειγμα-2

SELECT DISTINCT Όνομα, Τηλέφωνο
FROM ΠΕΛΑΤΗΣ;

{ < Perkins, 246801 >, <Καντακουζηνός, 246801 >, < Παλαιολόγος,987654>}

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654



Το *

Το * δίπλα στην εντολή SELECT σημαίνει ότι οι πλειάδες του αποτελέσματος θα περιλαμβάνουν όλα τα χαρακτηριστικά των σχέσεων που βρίσκονται μετά την εντολή FROM

* Παράδειγμα

SELECT *
FROM ΠΕΛΑΤΗΣ

{ < 1,Perkins, 246801 >, <2,Καντακουζηνός, 246801 >, < 3,Παλαιολόγος,987654>} , <4,Καντακουζηνός, 246801>

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654
4	Καντακουζηνός	24681

To keyword AS

- Το keyword **AS** χρησιμοποιείται τόσο για τη μετονομασία των χαρακτηριστικών των πλειάδων του αποτελέσματος, όσο και για τη μετονομασία των σχέσεων μετά την εντολή FROM.
- Η μετονομασία έχει τοπική ισχύ, δηλαδή ισχύει μόνον για το τρέχον ερώτημα και δεν μεταβάλλει την ονομασία των σχέσεων ή των χαρακτηριστικών της βάσης

As Παράδειγμα-1

**SELECT ID AS Κωδικός, Όνομα, τηλέφωνο AS τηλ.
FROM ΠΕΛΑΤΗΣ**

Δεν αλλάζει τίποτα ως προς
το αποτέλεσμα.
Αλλάζουν μόνον τα
ονόματα των
χαρακτηριστικών των
πλειάδων εξόδου

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654
4	Καντακουζηνός	24681

<u>Κωδικός</u>	όνομα	Τηλ.
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654
4	Καντακουζηνός	24681

As Παράδειγμα-2

```
SELECT *  
FROM ΠΕΛΑΤΗΣ AS P
```

Το αποτέλεσμα δεν διαφέρει από το αποτέλεσμα του ερωτήματος

```
SELECT *  
FROM ΠΕΛΑΤΗΣ
```

Ωστόσο, στην πρώτη περίπτωση, μετά την μετονομασία της σχέσης ΠΕΛΑΤΗΣ σε P, οποιαδήποτε άλλη φορά χρειαστεί να αναφερθούμε στη σχέση ΠΕΛΑΤΗΣ μέσα στο ίδιο ερώτημα θα χρησιμοποιείται μόνον το P.

Μεταβολή των τιμών των χαρακτηριστικών των πλειάδων εξόδου

Με την εντολή `select` μπορούμε να μεταβάλλουμε τις τιμές των χαρακτηριστικών των πλειάδων εξόδου. Πχ.

- πρόσθεση (+)/αφαίρεση (-) σταθερών ή εφαρμογή άλλων μαθηματικών συναρτήσεων (`sqrt`, `sin`, `mod`) σε τιμές των χαρακτηριστικών των πλειάδων εξόδου που είναι `numerical` τύπου (πχ. `int`, `float`, `double`)
- `appending` αλφαριθμητικών σε τιμές των χαρακτηριστικών των πλειάδων εξόδου που είναι αλφαριθμητικού τύπου
- απομόνωση της μέρας, του μήνα ή και του χρόνου της τιμής ενός χαρακτηριστικού που είναι τύπου `datetime` (ημερομηνία), μέσω των συναρτήσεων `DAY`, `MONTH`, `YEAR`.

Περίπτωση numerical Παράδειγμα-1

```
SELECT ID + 100  
FROM ΠΕΛΑΤΗΣ
```

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654
4	Καντακουζηνός	24681



<u>ID</u>
101
102
103
104

Περίπτωση αλφαριθμητικού Παράδειγμα-2

```
SELECT ('2310'+τηλέφωνο)  
FROM ΠΕΛΑΤΗΣ
```

<u>ID</u>	όνομα	τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654
4	Καντακουζηνός	24681



2310246801
2310246801
2310987654
231024681

Ο όρος WHERE

Όπως είδαμε και παραπάνω η εντολή WHERE περιέχει μία συνθήκη που πρέπει να ικανοποιούν οι πλειάδες του αποτελέσματος

Όταν ένα χαρακτηριστικό που βρίσκεται στην εντολή SELECT είναι τύπου numerical μπορούμε να χρησιμοποιήσουμε τους τελεστές <, >, =, <> για να το συγκρίνουμε με κάποια σταθερά.

Όταν είναι αλφαριθμητικού τύπου μπορούμε να χρησιμοποιήσουμε τον τελεστή LIKE για να το συγκρίνουμε με κάποιο άλλο αλφαριθμητικό.

WHERE Παράδειγμα-1

Ερώτημα: Να βρεθούν οι κωδικοί από όσες κασέτες έχουν διαθέσιμη ποσότητα από ένα έως δύο αντίτυπα, ή όσες η τιμή τους είναι μεγαλύτερη του 2

```
SELECT ID
FROM ΚΑΣΕΤΑ
WHERE (Ποσότητα >= 1 AND Ποσότητα <= 2) OR Τιμή > 2
```

ID	IDΤαινίας	Τύπος	Ποσότητα	Τιμή
1	1	VHS	3	2
2	1	DVD	1	3
3	2	VHS	2	2

η τιμή τους είναι μεγαλύτερη του 2

WHERE Παράδειγμα-2

Ερώτημα: Να βρεθούν οι κωδικοί των κασετών που είναι τύπου VHS και η διαθέσιμη ποσότητα τους είναι μεγαλύτερη του 2 ή η τιμή τους είναι μεγαλύτερη του 2

```
SELECT ID  
FROM ΚΑΣΕΤΑ  
WHERE Τύπος = 'VHS' AND (Ποσότητα > 2 OR Τιμή > 2)
```

ID	IDΤαινίας	Τύπος	Ποσότητα	Τιμή
1	1	VHS	3	2
2	1	DVD	1	3
3	2	VHS	2	2



απορρίπτεται καθώς Ποσότητα ≤ 2 AND Τιμή ≤ 2

Ο τελεστής LIKE

Για αλφαριθμητικά δεδομένα, χρήσιμος είναι ο τελεστής LIKE, για την σύγκριση ενός αλφαριθμητικού με κάποιο άλλο αλφαριθμητικό. Με τον τελεστή LIKE χρησιμοποιούνται δύο ειδικοί χαρακτήρες (μπαλαντέρ):

- % για ταύτιση οποιασδήποτε συμβολοσειράς
- _ για ταύτιση οποιουδήποτε χαρακτήρα

Πχ. Να βρεθούν τα ονόματα των πελατών που αρχίζουν από Κ

```
SELECT Όνομα  
FROM ΠΕΛΑΤΗΣ  
WHERE Όνομα LIKE 'Κ%'
```

Η τιμή NULL

Για να ελέγξουμε εάν ένα χαρακτηριστικό έχει την τιμή NULL **δεν** χρησιμοποιούνται τα `= NULL` και `<> NULL` αλλά **IS NULL** και **IS NOT NULL**.

Πχ. Να βρεθούν οι ενοικιάσεις για τις οποίες δεν έχει ορισθεί ημερομηνία επιστροφής

```
SELECT *  
FROM ΕΝΟΙΚΙΑΣΗ  
WHERE Έως IS NULL
```

Ταξινόμηση αποτελεσμάτων – ORDER BY

Πολλές φορές προκύπτει η ανάγκη ταξινόμησης των αποτελεσμάτων βάσει των τιμών ενός χαρακτηριστικού. Η SQL επιτρέπει την ταξινόμηση των πλειάδων του αποτελέσματος με την εντολή **ORDER BY**

Με τα keywords **ASC** και **DESC** μετά το χαρακτηριστικό ως προς το οποίο θέλουμε να γίνει η ταξινόμηση, δηλώνουμε εάν η ταξινόμηση θέλουμε να γίνει κατά **αύξουσα ή φθίνουσα σειρά**

Η εντολή **ORDER BY** (όταν χρησιμοποιείται) πρέπει να καλείται στο τέλος ενός SQL ερωτήματος και να περιλαμβάνει ένα (υπό)σύνολο των χαρακτηριστικών που βρίσκονται μετά την εντολή **SELECT**

ORDER BY Παράδειγμα-1

Ερώτημα: Να δοθούν οι κωδικοί των κασετών ταξινομημένοι ως προς τη διαθέσιμη ποσότητά τους, κατά αύξοντα τρόπο

```
SELECT ID, Ποσότητα  
FROM ΚΑΣΕΤΑ  
ORDER BY Ποσότητα;
```

ID	IDΤαινίας	Τύπος	Ποσότητα	Τιμή
1	1	VHS	3	2
2	1	DVD	1	3
3	2	VHS	2	2

ID	Ποσότητα
2	1
3	2
1	3

ORDER BY Παράδειγμα-2

Ερώτημα: Να δοθούν οι κωδικοί των κασετών ταξινομημένοι κατά φθίνοντα τρόπο ως προς την τιμή ενοικίασης. Στην περίπτωση ίσων τιμών ενοικίασης, η ταξινόμηση να γίνει κατά φθίνοντα τρόπο ως προς την ποσότητα

```
SELECT ID, Τιμή, Ποσότητα  
FROM ΚΑΣΕΤΑ  
ORDER BY Τιμή DESC, Ποσότητα ASC
```

ID	IDΤαινίας	Τύπος	Ποσότητα	Τιμή
1	1	VHS	3	2
2	1	DVD	1	3
3	2	VHS	2	2

ID	Τιμή	Ποσότητα
2	3	1
3	2	2
1	2	3