

Εισαγωγή στις Βάσεις Δεδομένων (Κεφάλαιο 1)

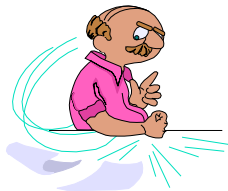
Μανόλης Γεργατσούλης

Καθηγητής

Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας &
Μουσειολογίας

Ιόνιο Πανεπιστήμιο

manolis@ionio.gr



Τι είναι το DBMS;



- **Βάσεις δεδομένων:** Ιδιαίτερα μεγάλες ολοκληρωμένες συλλογές δεδομένων.
- Αφορούν μοντέλα πραγματικών **οργανισμών**. Καταχωρούν πληροφορία για:
 - **Οντότητες** (π.χ., φοιτητές, μαθήματα, διδάσκοντες).
 - **Συσχετίσεις** (π.χ., Ο Νίκος έχει εγγραφεί στο μάθημα Βάσεις Δεδομένων).
- Το **Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS)** είναι λογισμικό ειδικά σχεδιασμένο για να διευκολύνει την αποθήκευση και τη διαχείριση βάσεων δεδομένων.



Μανόλης Γεργατσούλης
Οκτώβριος 2016

2

Γιατί να χρησιμοποιούμε DBMS;

- Ανεξαρτησία δεδομένων:
 - Οι εφαρμογές δεν εξαρτώνται από λεπτομέρειες αναπαράστασης & αποθήκευσης των δεδομένων.
- Γρήγορη πρόσβαση στα δεδομένα.
- Μείωση του χρόνου ανάπτυξης εφαρμογών.
 - Διάθεση πλήθους υποσυστημάτων γενικής χρήσης που ενσωματώνονται εύκολα σε κάθε εφαρμογή.
- Ακεραιότητα δεδομένων και ασφάλεια.
 - Επιβολή περιορισμών, έλεγχοι πρόσβασης κ.λ.π.
- Ομοιόμορφη διαχείριση των δεδομένων.
- Ταυτόχρονη πρόσβαση, επαναφορά από βλάβες.
 - Πολλοί χρήστες ταυτόχρονα, αποφυγή απώλειας δεδομένων, κ.λ.π.



Μανόλης Γεργατσούλης
Οκτώβριος 2016

3

Γιατί να μελετούμε Βάσεις Δεδομένων;

- Μετακίνηση από τον **υπολογισμό** στην **πληροφορία**
 - στο “κάτω άκρο”: η εξέλιξη του web (ακαταστασία!)
 - στο “άνω άκρο”: επιστημονικές εφαρμογές.
- Συλλογές δεδομένων με συνεχώς αυξανόμενη ποικιλομορφία και όγκο.
 - Ψηφιακές Βιβλιοθήκες, αλληλεπιδραστικό βίντεο.
- Το DBMS περιλαμβάνει τα περισσότερα από αυτά που σχετίζονται με την Επιστήμη των Υπολογιστών
 - ΛΣ, γλώσσες, θεωρία, AI, πολυμέσα, λογική.



Μανόλης Γεργατσούλης
Οκτώβριος 2016

4

Βάσεις Δεδομένων & Επιστήμη της Πληροφορίας (1/2)

- Οι βιβλιοθήκες, τα αρχεία και τα μουσεία διαχειρίζονται τεράστιο όγκο πληροφοριών.
- Βιβλιοθήκες:
 - Το συμβατικό υλικό των βιβλιοθηκών ψηφιοποιείται, αποθηκεύεται και διατίθεται ηλεκτρονικά.
 - Πολλά περιοδικά και πρακτικά επιστημονικών συνεδρίων εκδίδονται σήμερα μόνο σε ψηφιακή μορφή.
 - Μεγάλος όγκος μεταδεδομένων που αφορούν βιβλία, περιοδικά, πρακτικά συνεδρίων, διδακτορικές διατριβές, κ.α. παράγονται με τη χρήση προτύπων βιβλιογραφικών μεταδεδομένων όπως το MARC, το MODS, το Dublin Core, κ.α. τα οποία διευκολύνουν την γρήγορη αναζήτηση και ανεύρεση της πληροφορίας που ικανοποιεί τις απαιτήσεις των χρηστών.

Βάσεις Δεδομένων και Επιστήμες της Πληροφορίας (2/2)

- Αρχεία:
 - Όπως και στις βιβλιοθήκες, τα έγγραφα πολλών αρχείων ψηφιοποιούνται ώστε να είναι διαθέσιμα στους μελετητές σε ψηφιακή μορφή.
 - Με τη χρήση προτύπων όπως το ISAD, EAD κ.α. παράγεται μεγάλος όγκος μεταδεδομένων (εργαλεία αναζήτησης) για να διευκολύνουν την αναζήτηση και εύκολη πρόσβαση των χρηστών στην αρχαιακή πληροφορία.
- Μουσεία:
 - Virtual Museums: Μεγάλος όγκος από πολυμεσικά δεδομένα καθώς και μεταδεδομένα (εικόνες, κείμενο κ.λ.π.) πρέπει να οργανωθεί και να αποθηκευτεί με αποδοτικό τρόπο.

Μοντέλα Δεδομένων

- Μοντέλο δεδομένων είναι μια συλλογή από έννοιες για την περιγραφή των δεδομένων.
- Το σχήμα είναι η περιγραφή μιας συγκεκριμένης συλλογής δεδομένων, με τη χρήση δοσμένου μοντέλου δεδομένων.
- Το σχεσιακό μοντέλο δεδομένων είναι το πιο διαδεδομένο μοντέλο σήμερα.
 - Βασική έννοια: σχέση, (ουσιαστικά είναι ένας πίνακας με γραμμές και στήλες).
 - Κάθε σχέση έχει ένα σχήμα το οποίο περιγράφει τις στήλες ή πεδία ή γνωρίσματα ή χαρακτηριστικά.

Σχεσιακό μοντέλο: Παράδειγμα

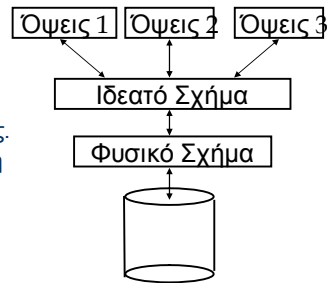
- Σχήμα:
 - Φοιτητής (AM:string, Όνομα:string, Επώνυμο:string, Ηλικία:

AM	Όνομα	Επώνυμο	Ηλικία
53123	Νίκος	Νικολάου	18
45123	Πέτρος	Γεωργίου	20
93122	Μαρία	Πέτρου	19
11234	Σοφία	Ανδρέου	21
99876	Γιώργος	Αντωνίου	20

Επίπεδα Αφαίρεσης

- Πολλές **όψεις (εξωτερικό σχήμα)**, ένα **ιδεατό (λογικό) σχήμα** και ένα **φυσικό σχήμα**.

- Οι όψεις περιγράφουν πως βλέπουν τα δεδομένα οι χρήστες.
- Το ιδεατό σχήμα ορίζει τη λογική δομή.
- Το φυσικό σχήμα περιγράφει τα αρχεία και τα ευρετήρια που χρησιμοποιούνται.



- ☒ Τα σχήματα ορίζονται με τη βοήθεια της DDL,
- ☒ Τα δεδομένα τροποποιούνται / ερωτούνται με τη βοήθεια της DML.

Παράδειγμα: Βάση Δεδομένων Πανεπιστημίου

- **Ιδεατό Σχήμα:**
 - φοιτητές(AM: string, Όνομα: string, Επώνυμο: string, Ηλικία: integer)
 - μαθήματα(KM: string, Ονομασία:string, Κατηγορία:integer)
 - εγγραφές(AM:string, KM:string, Βαθμός:integer)
- **Φυσικό Σχήμα:**
 - Σχέσεις αποθηκευμένες σαν μη ταξινομημένα αρχεία.
 - Ευρετήριο πάνω στη πρώτη στήλη του πίνακα φοιτητές.
- **Εξωτερικό Σχήμα (Όψη):**
 - εγγεγραμμένοι(KM:string, Πλήθος:integer)

Ανεξαρτησία Δεδομένων

- Τα επίπεδα αφαίρεσης εξασφαλίζουν ανεξαρτησία δεδομένων.
- Κάθε επίπεδο απομονώνει τις αλλαγές που γίνονται στο πιο κάτω επίπεδο.
- Οι πίνακες του εξωτερικού σχήματος προκύπτουν από τους πίνακες του ιδεατού (λογικού) σχήματος.
- **Λογική ανεξαρτησία δεδομένων:** Μπορούμε να αλλάζουμε την μορφή των πινάκων του ιδεατού σχήματος χωρίς να αλλάζει η μορφή των πινάκων του εξωτερικού σχήματος.
- **Φυσική ανεξαρτησία δεδομένων:** Το ιδεατό σχήμα απομονώνει το χρήστη από τις αλλαγές που γίνονται στο φυσικό σχήμα, δηλαδή αλλαγές που αφορούν στην καταχώρηση των δεδομένων στην περιφερειακή μνήμη.

Τα Αιτήματα στο DBMS

- Ευκολία ανάκτησης δεδομένων από τη βάση.
 - Ποιο είναι το ονοματεπώνυμο του φοιτητή με Αριθμό Μητρώου (AM) 12345;
 - Πόσοι φοιτητές έχουν εγγραφεί στο μάθημα με κωδικό C1234;
- Γλώσσα αιτημάτων.
- Τυπικές Γλώσσες αιτημάτων:
 - **Σχεσιακός Λογισμός:** βασίζεται στη **μαθηματική λογική**.
 - **Σχεσιακή Άλγεβρα:** χρήση συνόλου **τελεστών**.

Έλεγχος Ταυτοχρονισμού

- Η ταυτόχρονη εκτέλεση προγραμμάτων χρηστών είναι ουσιώδης για την καλή λειτουργία του DBMS.
 - Επειδή η προσπέλαση των δίσκων είναι συχνή και σχετικά αργή, είναι σημαντικό να επιτρέπεται στην ΚΜΕ να εκτελεί πολλά προγράμματα χρηστών ταυτόχρονα.
- Οι εναλλασσόμενες ενέργειες διαφορετικών προγραμμάτων μπορεί να οδηγήσουν σε ασυνέπειες: π.χ., υπολογισμός συνόλου καταθέσεων τη στιγμή που μεταφέρεται ποσό σε άλλο λογαριασμό.
- Το DBMS εξασφαλίζει ότι δεν προκύπτουν τέτοια προβλήματα: οι χρήστες συμπεριφέρονται σαν να χρησιμοποιούν ένα σύστημα ενός χρήστη.

Συναλλαγή: Εκτέλεση ενός Προγράμματος ΒΔ

- Σημαντική η έννοια της **συναλλαγής**: ακολουθία **ατομικών** ενεργειών στη ΒΔ (διάβασμα /γράψιμο).
- Η πλήρης εκτέλεση μιας συναλλαγής πρέπει να αφήνει την ΒΔ σε μια **συνεπή κατάσταση** αν η ΒΔ είναι συνεπής πριν την έναρξη της συναλλαγής.
 - Ο χρήστης μπορεί να ορίσει **περιορισμούς ακεραιότητας** στα δεδομένα. Το DBMS θα επιβάλει τους περιορισμούς.
 - Το DBMS δεν κατανοεί στην πραγματικότητα τη σημασία των δεδομένων. (π.χ., δεν κατανοεί πως υπολογίζεται ο τόκος σε ένα τραπεζικό λογαριασμό).
 - Έτσι, η εξασφάλιση ότι μια συναλλαγή (όταν εκτελείται μόνη της) διατηρεί την συνέπεια είναι ευθύνη του **χρήστη!**

Ταυτόχρονη Εκτέλεση Συναλλαγών

- Το DBMS εξασφαλίζει ότι η εκτέλεση των συναλλαγών $\{T_1, \dots, T_n\}$ είναι ισοδύναμη με κάποια **σειριακή** εκτέλεση των T_1', \dots, T_n' .
 - Πριν το γράψιμο/διάβασμα ενός αντικειμένου, η συναλλαγή απαιτεί από το DBMS μια κλειδαριά για το αντικείμενο και αναμένει να της παραχωρηθεί. Οι κλειδαριές απελευθερώνονται στο τέλος της συναλλαγής.
 - **Ιδέα:** Αν μια ενέργεια της T_i (π.χ., γράψιμο του X) επηρεάζει την T_j (που πιθανά διαβάζει το X), μια από αυτές, έστω η T_i , θα αποκτήσει τη κλειδαριά στο X πρώτη, ενώ η T_j εξαναγκάζεται να περιμένει ως ότου ολοκληρωθεί η T_i . Αυτό βάζει σε μια σειρά τις συναλλαγές.
 - Περισσότερα για κλειδαριές σε επόμενα κεφάλαια!

Ημιτελείς Συναλλαγές-Βλάβες Συστήματος

- DBMS εξασφαλίζει **ατομικότητα** (όλα-ή-τίποτα) ακόμη και όταν το σύστημα υποστεί βλάβη στο μέσω μιας συναλλαγής.
- **Ιδέα:** Τήρηση **ιστορικού** όλων των ενεργειών του DBMS κατά την εκτέλεση της συναλλαγής:
 - Πριν γίνει κάποια μεταβολή στη ΒΔ, μια καταχώρηση στο ιστορικό τοποθετείται σε ασφαλή θέση (**έγκαιρη καταγραφή ιστορικού - WAL**).
 - Μετά από μια βλάβη, τα αποτελέσματα των ημιτελών συναλλαγών **αναιρούνται** με τη βοήθεια του ιστορικού.

Το Ιστορικό



- Στο ιστορικό καταχωρούνται οι ακόλουθες ενέργειες:
 - Η Τι γράφει ένα αντικείμενο:* η παλιά και η νέα τιμή.
 - Η εγγραφή του ιστορικού πρέπει να καταχωρείται στον δίσκο πριν από την αλλαγμένη σελίδα!
 - Η Τι ολοκληρώνεται / διακόπτεται:* μια εγγραφή που δηλώνει την ενέργεια.
- Οι καταχωρήσεις του ιστορικού συνδέονται (αλυσιδωτά) με τη ταυτότητα μιας ενέργειας, με αποτέλεσμα την εύκολη αναίρεση της ενέργειας αυτής (π.χ., επίλυση αδιεξόδου).
- Το ιστορικό συχνά *αντιγράφεται* και *καταχωρείται* σε "σταθερό" αποθηκευτικό μέσο.
- Όλες οι σχετιζόμενες με το ιστορικό δραστηριότητες (στην πραγματικότητα, όλες οι δραστηριότητες που σχετίζονται με ταυτόχρονη εκτέλεση όπως κλείδωμα /ξεκλείδωμα, και με αδιέξοδα κ.λ.π.) διαχειρίζονται από το DBMS.

Ποιοι ασχολούνται με τις Βάσεις Δεδομένων...

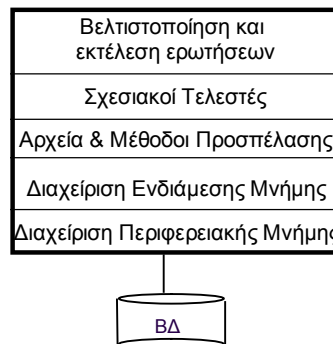
- Κατασκευαστές Βάσεων Δεδομένων (εργάζονται σε εταιρίες όπως η Oracle, IBM κ.λ.π)
- Τελικοί χρήστες και σχεδιαστές DBMS.
- Προγραμματιστές Εφαρμογών ΒΔ.
- Διαχειριστής Συστήματος ΒΔ (DBA)**
 - Σχεδιάζει το ιδεατό /φυσικό σχήμα.
 - Διαχειρίζεται της ασφάλεια και την εξουσιοδότηση.
 - Διαθεσιμότητα δεδομένων, επαναφορά από βλάβες.
 - Ρυθμίσεις στη ΒΔ, εξελισσόμενες απαιτήσεις χρηστών.



Πρέπει να κατανοούν τη λειτουργία ενός DBMS!

Η Δομή ενός DBMS

- Ένα τυπικό DBMS έχει πολυεπίπεδη αρχιτεκτονική.
- Στο σχήμα δε φαίνονται τα τμήματα για έλεγχο ταυτοχρονισμού και επαναφορά.
- Αυτή είναι μια από τις πιθανές αρχιτεκτονικές. Κάθε σύστημα έχει τις δικές του παραλλαγές.



Αυτά τα επίπεδα πρέπει να ασχολούνται και με τον έλεγχο ταυτοχρονισμού και την επαναφορά

Περίληψη

- Πλεονεκτήματα DBMS: επανάκτηση από βλάβες, ταυτόχρονη προσπέλαση, γρήγορη ανάπτυξη εφαρμογών, ακεραιότητα και η ασφάλεια των δεδομένων κ.α.
- Τα επίπεδα αφαίρεσης εξασφαλίζουν ανεξαρτησία των δεδομένων.
- Το DBMS έχει συνήθως αρχιτεκτονική πολλών επιπέδων.
- DBA: υπεύθυνη και καλοπληρωμένη εργασία!
- Τα DBMS χρησιμοποιούνται για διατήρηση μεγάλων συλλογών δεδομένων και υποβολή αιτημάτων.
- Το R&D για DBMS είναι ένα από τα ευρύτερα και πιο ενδιαφέροντα πεδία της Επιστήμης των Υπολογιστών.



Σύγχρονες Εξελίξεις στις ΒΔ

- Έρευνα για σχεδιασμό ισχυρότερων γλωσσών αιτημάτων και εμπλουτισμό των μοντέλων δεδομένων.
- Δυνατότητες για καταχώρηση νέων δομών (εικόνες, ελεύθερο κείμενο, XML κ.λ.π.).
- Αποθήκες Δεδομένων, Εξόρυξη Δεδομένων.
- Λογισμικό για προγραμματισμό πόρων (ERP) και διαχείριση πόρων (MRP) επιχείρησης .
 - Ενίσχυση των DBMS με εφαρμογές που επικάθονται σε αυτό.
- Είσοδος DBMS στην εποχή του Internet:
 - Καταχώρηση δεδομένων σε DBMS και ανάκτηση μέσω του λογισμικού πλοήγησης.
 - Ερωτήσεις μέσα από φόρμες του διαδικτύου, μορφοποίηση αποτελεσμάτων σε HTML.
- Ολοκλήρωση δεδομένων (άντληση πληροφοριών με ενιαίο τρόπο από ετερογενείς βάσεις δεδομένων).
- Ανταλλαγή δεδομένων.

Το Μοντέλο Οντότητας-Συσχέτισης (Κεφάλαιο 2)

Μανόλης Γεργατσούλης

Καθηγητής

Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας &
Μουσειολογίας

Ιόνιο Πανεπιστήμιο

manolis@ionio.gr



Επισκόπηση Σχεδιασμού ΒΔ

- Ανάλυση Απαιτήσεων.
 - Κατανόηση-καταγραφή είδους δεδομένων, επισήμανση λειτουργιών που συμβαίνουν συχνότερα και επηρεάζουν την απόδοση του συστήματος, καταγραφή απαιτήσεων χρηστών.
- Σχεδιασμός Ιδεατής ΒΔ.
 - Χρήση ER μοντέλου.
- Σχεδιασμός της Λογικής ΒΔ.
 - Επιλογή DBMS (έχουν επικρατήσει τα σχεσιακά), δημιουργία λογικού σχήματος.
- Τελειοποίηση του Σχήματος.
 - «Κανονικοποίηση» των πινάκων ώστε να πληρούνται κάποιες επιθυμητές ιδιότητες.
- Σχεδιασμός του Φυσικού Σχήματος.
- Σχεδιασμός της Πολιτικής Ασφάλειας.
 - Κατάταξη των χρηστών σε ομάδες με διακριτούς ρόλους και διαφορετικές δυνατότητες και δικαιώματα πρόσβασης στα διάφορα τμήματα της ΒΔ.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

2

Επισκόπηση Σχεδιασμού Ιδεατής ΒΔ

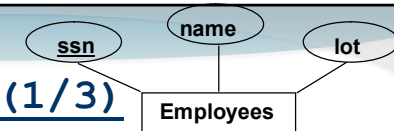
- Σχεδιασμός Ιδεατής ΒΔ: (σ' αυτή τη φάση χρησιμοποιείται το **Μοντέλο Οντότητας-Συσχέτισης** ή **ER Μοντέλο**.)
 - Ποιες είναι οι **οντότητες** και οι **συσχετίσεις**;
 - Ποιες πληροφορίες σχετικές με αυτές τις οντότητες και τις συσχετίσεις πρέπει να αποθηκευτούν στη βάση δεδομένων;
 - Ποιοι είναι οι **περιορισμοί ακεραιότητας** και οι **επιχειρηματικοί κανόνες λειτουργίας** που ισχύουν;
 - Ένα 'σχήμα' βάσης δεδομένων στο ER Μοντέλο μπορεί να παρασταθεί γραφικά (**ER διάγραμμα**).
 - Ένα ER διάγραμμα μπορεί να απεικονιστεί σε ένα σχεσιακό σχήμα.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

3

Το ER Μοντέλο (1/3)



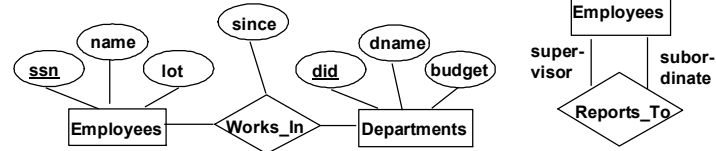
- **Οντότητα**: Αντικείμενο του πραγματικού κόσμου διακριτό από τα άλλα αντικείμενα. Μια οντότητα περιγράφεται στη ΒΔ χρησιμοποιώντας ένα σύνολο **γνωρισμάτων**.
- **Σύνολο Οντοτήτων**: Συλλογή ομοειδών οντοτήτων. Π.χ., όλοι οι εργαζόμενοι (*Employees*).
 - Όλες οι οντότητες σε ένα σύνολο οντοτήτων έχουν το ίδιο σύνολο γνωρισμάτων. (Τουλάχιστον μέχρι να μιλήσουμε για ISA ιεραρχίες!)
 - Κάθε σύνολο οντοτήτων έχει ένα **κλειδί** (ελάχιστος αριθμός γνωρισμάτων που προσδιορίζουν μονοσήμαντα μια οντότητα).
 - Μπορεί να υπάρχουν περισσότερα του ενός **υποψήφια κλειδιά**. Ένα από αυτά επιλέγεται ως **κύριο κλειδί**.
 - Κάθε γνώρισμα έχει ένα **πεδίο ορισμού**.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

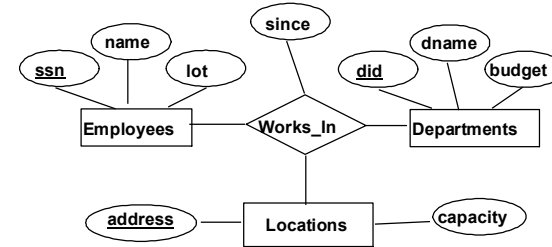
4

Το ER Μοντέλο (2/3)



- **Συσχέτιση:** Διασύνδεση μεταξύ δύο ή περισσότερων οντοτήτων. Π.χ. η Μαρία εργάζεται στο τμήμα καλλυντικών.
- **Σύνολο Συσχετίσεων:** Συλλογή ομοειδών συσχετίσεων.
 - Ένα n -αδικό σύνολο συσχετίσεων R συσχετίζει n σύνολα οντοτήτων E_1, \dots, E_n : κάθε συσχέτιση του R περιλαμβάνει οντότητες $e_1 \in E_1, \dots, e_n \in E_n$.
 - Το ίδιο σύνολο οντοτήτων μπορεί να συμμετέχει σε διαφορετικά σύνολα συσχετίσεων ή σε διαφορετικούς 'ρόλους' στο ίδιο σύνολο.
 - Μια συσχέτιση μπορεί να εμπεριέχει και περιγραφικά γνωρίσματα.
 - Στα πιο πάνω παραδείγματα έχουμε **δ्वαδικές συσχετίσεις**.

Το ER Μοντέλο (3/3)



- Στο παράδειγμα του σχήματος η συσχέτιση Works_In εμφανίζεται ως **τριαδική συσχέτιση** αφού συνδέει τρία σύνολα οντοτήτων.

Πρόσθετα χαρακτηριστικά του ER μοντέλου

- Το ER μοντέλο διαθέτει δομές που καθιστούν δυνατή την απεικόνιση ουσιαστικών ιδιοτήτων των δεδομένων. Τέτοιες δομές είναι:
 - Περιορισμοί Κλειδιού
 - Περιορισμοί Συμμετοχής
 - Αδύναμες Οντότητες
 - Ιεραρχίες Οντοτήτων
 - Συνυπολογισμός

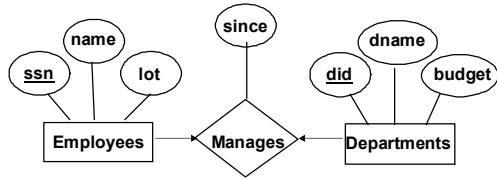
Περιορισμοί Κλειδιού (1/2)

- Έστω η Works_In: Ένας εργαζόμενος μπορεί να εργάζεται σε πολλά τμήματα. Ένα τμήμα μπορεί να έχει πολλούς εργαζόμενους (Πολλά προς Πολλά).
- Αντιθέτως, κάθε τμήμα έχει το πολύ ένα διευθυντή, σύμφωνα με τον **περιορισμό κλειδιού** στη σχέση Manages (Ένα προς Πολλά). Προσοχή: ένας υπάλληλος μπορεί να διευθύνει πολλά τμήματα.
- Ο περιορισμός κλειδιού αναπαριστάται με κατευθυνόμενη γραμμή (από την οντότητα προς τη συσχέτιση).

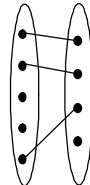
Πολλά-προς-πολλά

1-προς-Πολλά

Περιορισμοί Κλειδιού (2/2)



- Αν προσθέσω επιπλέον τον περιορισμό (κλειδιού) ότι κάθε υπάλληλος μπορεί να διευθύνει το πολύ ένα τμήμα τότε έχω συσχέτιση Ένα-προς-Ένα.



1-προς-1



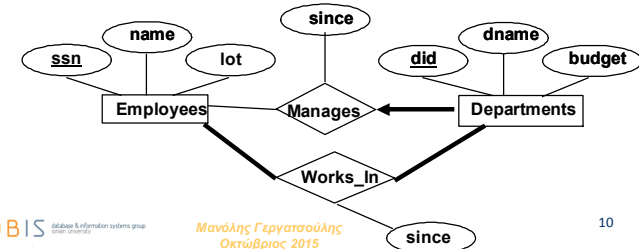
database information systems group

Μανώλης Γεργασιούλης
Οκτώβριος 2015

9

Περιορισμοί Συμμετοχής

- Έχουν όλα τα τμήματα διευθυντή;
 - Αν ναι, τότε έχουμε **περιορισμό συμμετοχής**: η συμμετοχή της Departments στην Manages ονομάζεται **ολική** (σε διαφορετική περίπτωση θα λέγεται **μερική** συμμετοχή).
 - Κάθε τιμή του did στον πίνακα Departments πρέπει να εμφανίζεται σε μια γραμμή του πίνακα Manages (με τιμή του ssn διάφορη του null!).
 - Ο περιορισμός συμμετοχής αναπαρίσταται με έντονη γραμμή στο ER διάγραμμα.
 - Έντονη γραμμή με κατεύθυνση αναπαριστά την ταυτόχρονη ύπαρξη περιορισμών κλειδιού και συμμετοχής.



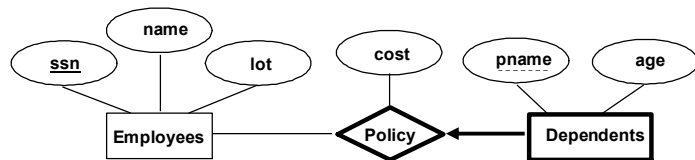
database information systems group

Μανώλης Γεργασιούλης
Οκτώβριος 2015

10

Αδύναμες Οντότητες

- Στο παράδειγμα οι υπάλληλοι έχουν τη δυνατότητα να συνάπτουν ασφαλιστικά συμβόλαια για τα εξαρτώμενα μέλη των οικογενειών τους.
- Μια **αδύναμη οντότητα** μπορεί να προσδιοριστεί μοναδικά μόνο μέσω του πρωτεύοντος κλειδιού μιας άλλης οντότητας (**προσδιορίζων ιδιοκτήτης**).
 - Πρέπει να υπάρχει μια συσχέτιση τύπου ένα-προς-πολλά ανάμεσα στο σύνολο οντοτήτων του προσδιορίζοντος ιδιοκτήτη και στο σύνολο των αδύναμων οντοτήτων (ένας ιδιοκτήτης, πολλές αδύναμες οντότητες).
 - Το σύνολο αδύναμων οντοτήτων **πρέπει να έχει ολική συμμετοχή στο αντίστοιχο σύνολο των συσχετίσεων προσδιορισμού**.



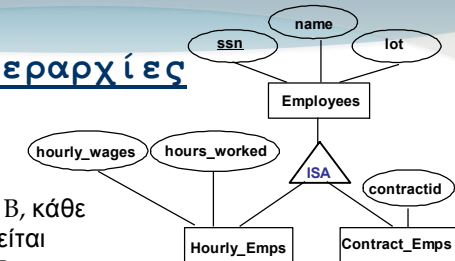
database information systems group

Μανώλης Γεργασιούλης
Οκτώβριος 2015

11

ISA ('is a') Ιεραρχίες

- Εδώ τα γνωρίσματα κληρονομούνται.
- Αν δηλώσουμε A ISA B, κάθε οντότητα του A θεωρείται επίσης οντότητα του B.



- Περιορισμοί επικάλυψης**: Μπορεί ο John να ανήκει ταυτόχρονα και στο σύνολο οντοτήτων Hourly_Emps και στο Contract_Emps; (Επιτρέπεται/απαγορεύεται)
- Περιορισμοί κάλυψης**: Πρέπει κάθε οντότητα Employees να ανήκει υποχρεωτικά είτε στο Hourly_Emps είτε στο Contract_Emps; (Ναι/όχι)
- Λόγοι για να χρησιμοποιούμε ISA:
 - Για να ορίσουμε ένα νέο γνώρισμα που αφορά μόνο μερικά από τα μέλη ενός συνόλου οντοτήτων (π.χ. hourly_wages).
 - Για να προσδιορίζουμε το σύνολο των οντοτήτων που μπορούν να συμμετέχουν σε μια συσχέτιση (π.χ. διευθυντές μόνο συμβασιούχοι.)



database information systems group

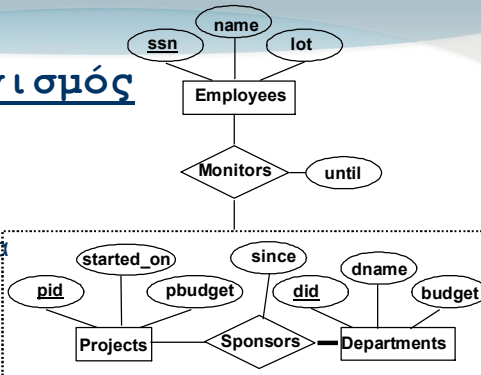
Μανώλης Γεργασιούλης
Οκτώβριος 2015

12

Συνυπολογισμός

- Χρησιμοποιείται όταν έχουμε να αναπαραστήσουμε μια συσχέτιση που περιλαμβάνει (σύνολο οντοτήτων και) ένα σύνολο συσχέτισης.

- Συνυπολογισμός** επιτρέπει να αντιμετωπίσουμε ένα σύνολο συσχέτισεων σαν ένα σύνολο οντοτήτων για λόγους συμμετοχής σε (άλλες) συσχέτισεις.



* **Συνυπολογισμός ή τριαδική συσχέτιση:**
 v Η *Monitors* είναι ξεχωριστή συσχέτιση, με ένα περιγραφικό γνώρισμα.
 v Επίσης, μπορούμε να πούμε ότι κάθε Επιχορήγηση επιοπτεύεται από το πολύ ένα εργαζόμενο.

Σχεδίαση με το ER Μοντέλο

- Επιλογές Σχεδίασης:**
 - Χρησιμοποιούμε οντότητα ή γνώρισμα για τη μοντελοποίηση μιας έννοιας;
 - Χρησιμοποιούμε οντότητα ή συσχέτιση;
 - Προσδιορισμός συσχέτισεων: Διαδικές ή τριαδικές; Συνυπολογισμός;
- Περιορισμοί στο ER Μοντέλο:**
 - Πολλά σημασιολογικά χαρακτηριστικά των δεδομένων πρέπει (και μπορούν) να αναπαριστούνται.
 - Κάποιοι περιορισμοί δεν μπορούν να αναπαρασταθούν στα ER διαγράμματα.

Βασικά δομικά στοιχεία του ER μοντέλου

□ οντότητα

○ γνώρισμα

◇ συσχέτιση

— Συνδέει γνωρίσματα με την αντίστοιχη οντότητα και οντότητες με συσχέτισεις

→ Από οντότητα προς συσχέτιση. Απεικονίζει περιορισμό κλειδιού (δηλ. κάθε οντότητα συμμετέχει το πολύ μια φορά στη συσχέτιση). Π.χ. κάθε τμήμα έχει το πολύ ένα διευθυντή.

— Υποδηλώνει περιορισμό συμμετοχής (π.χ. κάθε υπάλληλος εργάζεται σε ένα τμήμα

→ Περιορισμός κλειδιού + περιορισμός συμμετοχής (π.χ. κάθε τμήμα έχει ακριβώς ένα διευθυντή)

ISA Ιεραρχίες Οντοτήτων

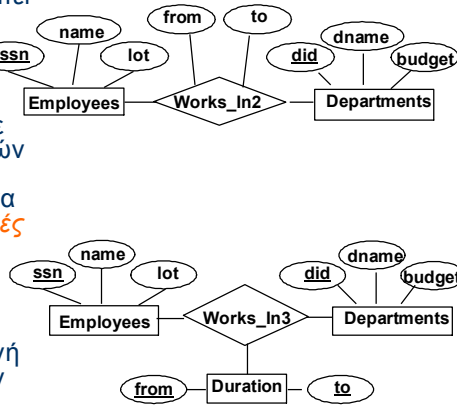
◁ Αδύναμη Οντότητα

Οντότητα ή Γνώρισμα (1/2)

- Θα έπρεπε η *address* να είναι γνώρισμα της *Employees* ή οντότητα (συνδεμένη με την *Employees* μέσω μιας συσχέτισης);
- Εξαρτάται από τη χρήση που θα κάνουμε στη διεύθυνση, και τη σημασιολογία των δεδομένων:
 - Αν έχουμε πολλές διευθύνσεις ανά εργαζόμενο, η *address* πρέπει να είναι οντότητα (αφού οι τιμές των γνωρισμάτων δεν μπορεί να είναι σύνολα).
 - Όταν η δομή της διεύθυνσης (πόλη, οδός, κ.λ.π.) μας ενδιαφέρει, π.χ., επιθυμούμε να ανακτούμε αυτούς που εργάζονται σε συγκεκριμένη πόλη, η *address* πρέπει να παρασταθεί σαν οντότητα (αφού οι τιμές των γνωρισμάτων είναι ατομικές).

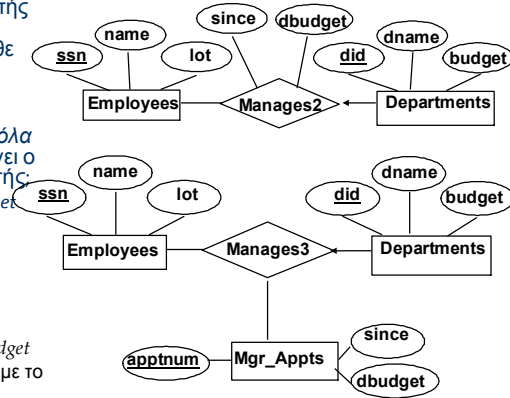
Οντότητα ή Γνωρίσματα (2/2)

- Η *Works_In2* δεν επιτρέπει σε εργαζόμενο να εργάζεται σε ένα τμήμα για δύο ή περισσότερες περιόδους.
- Παρόμοιο πρόβλημα με καταχώρηση πολλαπλών διευθύνσεων για έναν εργαζόμενο: θέλουμε να καταχωρούμε **πολλαπλές τιμές περιγραφικού γνωρίσματος για κάθε στιγμιότυπο της συσχέτισης**.
- Αντιμετώπιση: Εισαγωγή του συνόλου οντοτήτων *Duration*.



Οντότητα ή Συσχέτιση

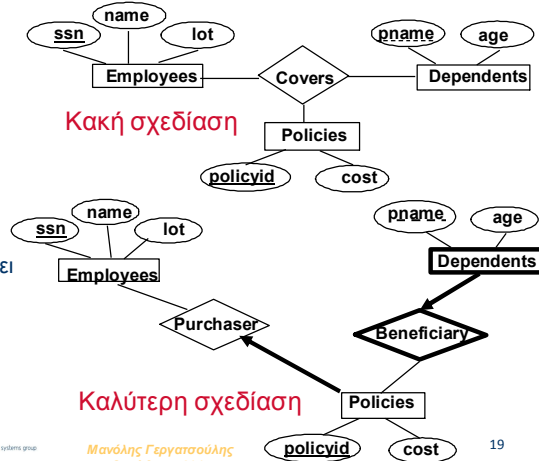
- Το πρώτο ER διάγραμμα ικανοποιεί αν ο διευθυντής αντιστοιχίζεται σε έναν προϋπολογισμό για κάθε τμήμα που διευθύνει.
- Τι γίνεται αν ο προϋπολογισμός που καταχωρείται καλύπτει όλα τα τμήματα που διευθύνει ο συγκεκριμένος διευθυντής;
 - Πλεονασμός: το *dbudget* καταχωρείται για κάθε τμήμα που διευθύνει ο συγκεκριμένος διευθυντής;



Παρανόηση: ενδέχεται το *dbudget* να συνδεθεί από παρανόηση με το διαχειριζόμενο τμήμα.

Διαδικές ή Τριαδικές Συσχετίσεις (1/2)

- Αν κάθε *policy* αναφέρεται σε ακριβώς 1 *employee*:
 - Οι περιορισμοί κλειδιού στον *Policies* σημαίνουν ότι ασφαλιστική κάλυψη καλύπτει μόνο 1 *dependent*!
- Ποιοι είναι οι επιπλέον περιορισμοί στο 2ο διάγραμμα;



Διαδικές ή Τριαδικές Συσχετίσεις (2/2)

- Το προηγούμενο παράδειγμα δείχνει μια περίπτωση που δύο διαδικές συσχετίσεις είναι προτιμότερες από μια τριαδική συσχέτιση.
- Παράδειγμα στην αντίθετη κατεύθυνση: η τριαδική σχέση *Contracts* συσχετίζει τα σύνολα οντοτήτων *Parts*, *Departments* και *Suppliers*, και έχει το περιγραφικό γνώρισμα *qty*. Κανένας συνδυασμός δυαδικών συσχετίσεων δεν αποτελεί ικανοποιητικό υποκατάστατο:
 - Π.χ. Οι S “can-supply” P, D “needs” P, και D “deals-with” S δεν υπονοεί ότι ο D συμφώνησε να αγοράσει το P από τον S.
 - Πως καταχωρούμε το *qty*;

Περίληψη του ER Μοντέλου (1/3)

- Ο σχεδιασμός ιδεατής ΒΔ ακολουθεί την ανάλυση απαιτήσεων.
 - Παρέχει υψηλού επιπέδου περιγραφή των δεδομένων που πρόκειται να αποθηκευτούν.
- Το ER μοντέλο είναι δημοφιλές για σχεδιασμό ιδεατής ΒΔ.
 - Χρησιμοποιεί εκφραστικά δομικά στοιχεία, κοντά στον τρόπο που σκέφτονται οι άνθρωποι για τις εφαρμογές.
- Βασικά δομικά στοιχεία: *οντότητες*, *συσχετίσεις*, και *γνωρίσματα* (οντοτήτων και συσχετίσεων).
- Επιπρόσθετα δομικά στοιχεία: *αδύναμες οντότητες*, *ISA ιεραρχίες*, και *συνυπολογισμός*.
- Σημείωση: Υπάρχουν πολλές παραλλαγές του ER μοντέλου.

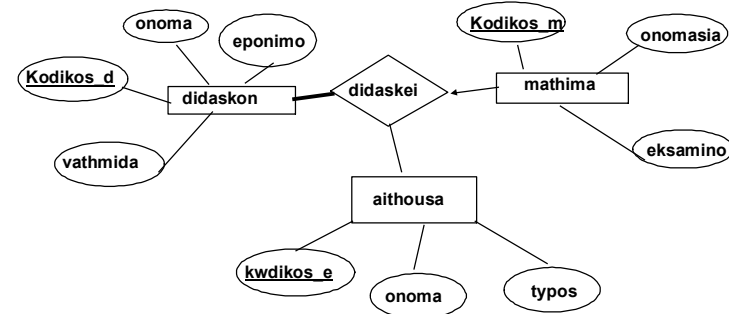
Περίληψη του ER Μοντέλου (2/3)

- Διάφορα είδη περιορισμών ακεραιότητας μπορούν να εκφραστούν στο ER μοντέλο: *περιορισμοί κλειδιού*, *περιορισμοί συμμετοχής*, και *περιορισμοί επικάλυψης/κάλυψης* για ISA ιεραρχίες. Κάποιοι *περιορισμοί ξένου κλειδιού* μπορούν επίσης να περιγραφούν έμμεσα στον ορισμό ενός συνόλου συσχετίσεων.
 - Κάποιοι περιορισμοί (όπως οι, *συναρτησιακές εξαρτήσεις*) δεν μπορούν να εκφραστούν στο ER μοντέλο.
 - Οι περιορισμοί παίζουν σημαντικό ρόλο στον προσδιορισμό της βέλτιστης σχεδίασης της βάσης δεδομένων ενός οργανισμού.

Περίληψη του ER Μοντέλου (3/3)

- Η σχεδίαση του ER διαγράμματος είναι *υποκειμενική*. Υπάρχουν συνήθως πολλοί τρόποι να μοντελοποιηθεί ένα σενάριο! Η ανάλυση των εναλλακτικών λύσεων είναι περίπλοκη κυρίως για μεγάλους οργανισμούς. Οι συνήθεις επιλογές περιλαμβάνουν:
 - Οντότητα ή χαρακτηριστικά, οντότητα ή συσχέτιση, δυαδική ή ν-αδική συσχέτιση, χρήση ή όχι ιεραρχιών ISA, χρήση ή όχι συνυπολογισμού.
- Εξασφάλιση καλής σχεδίασης της ΒΔ: το σχεσιακό μοντέλο που θα προκύψει πρέπει να αναλυθεί και να βελτιωθεί παραπέρα. Η πληροφορία για τις συναρτησιακές συσχετίσεις και οι τεχνικές κανονικοποίησης είναι ιδιαίτερα χρήσιμες.

ER-διάγραμμα: Ένα ακόμη παράδειγμα



Το Σχεσιακό Μοντέλο (Κεφάλαιο 3)

Μανόλης Γεργατσούλης

Καθηγητής

Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας
& Μουσειολογίας

Ιόνιο Πανεπιστήμιο

manolis@ionio.gr



Γιατί Μελετάμε το Σχεσιακό Μοντέλο;

- Είναι το πιο διαδεδομένο μοντέλο.
 - Σχεσιακά DBMS: DB2 της IBM, Oracle, Informix, MS-ACCESS και MS-SQLServer της Microsoft, Sybase, κ.λ.π.
- Τελευταίος ανταγωνιστής: αντικειμενοστραφές μοντέλο.
 - ObjectStore, Versant, Ontos
 - Πρόσφατη σύνθεση των δύο μοντέλων: *αντικείμενο-σχεσιακό μοντέλο*
 - Informix Universal Server, UniSQL, O2, Oracle, DB2



Μανόλης Γεργατσούλης
Οκτώβριος 2015

2

Σχεσιακές Βάσεις Δεδομένων: Ορισμοί

- **Σχεσιακή ΒΔ**: ένα σύνολο *σχέσεων (πινάκων)*
- **Σχέση**: αποτελείται από 2 συστατικά:
 - **Στιγμιότυπο**: ένας *πίνακας*, με γραμμές και στήλες.
#Γραμμές = πληθυσμός, #πεδία = βαθμός.
 - **Σχήμα**: προσδιορίζει το όνομα της σχέσης, και τα ονόματα και τους τύπους κάθε στήλης.
 - Π.χ. Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real).
- Μια σχέση μπορεί να θεωρηθεί ως *σύνολο γραμμών* ή *πλειάδων* (οι γραμμές είναι διαφορετικές μεταξύ τους).



Μανόλης Γεργατσούλης
Οκτώβριος 2015

3

Παράδειγμα: Στιγμιότυπο της Σχέσης Students

Πεδία, γνωρίσματα, στήλες

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Ονόματα πεδίων

Πλειάδες (εγγραφές γραμμές)

- ❖ Πληθυσμός = 3, βαθμός = 5, διακριτές γραμμές
- ❖ Πρέπει όλες οι στήλες σε ένα στιγμιότυπο μιας σχέσης να είναι διακριτές;



Μανόλης Γεργατσούλης
Οκτώβριος 2015

4

Σχεσιακές Γλώσσες Αιτημάτων

- Ισχυρό πλεονέκτημα του σχεσιακού μοντέλου: υποστηρίζει την υποβολή απλών αλλά ισχυρών **αιτημάτων** πάνω στα δεδομένα.
- Τα αιτήματα μπορούν να διατυπωθούν διαισθητικά και το DBMS είναι υπεύθυνο για τον αποδοτικό υπολογισμό τους.
 - Σημείο κλειδί: η ακριβής σημασιολογία των σχεσιακών αιτημάτων.
 - Επιτρέπει στο βελτιστοποιητή την εκτενή αναδιάταξη των λειτουργιών, εξασφαλίζοντας παράλληλα ότι η απάντηση δεν αλλάζει.



Η Γλώσσα Αιτημάτων SQL

- Αναπτύχθηκε από την IBM (system R) τη δεκαετία του 1970.
- Απαιτήση για πρότυπα αφού χρησιμοποιείται από πολλούς κατασκευαστές λογισμικού.
- Πρότυπα:
 - SQL-86
 - SQL-89 (μικρές βελτιώσεις)
 - SQL-92 (σημαντικές βελτιώσεις, το τρέχον πρότυπο)
 - SQL-99 (ουσιαστικές επεκτάσεις).



Η Γλώσσα Αιτημάτων SQL

- Για να βρούμε όλους τους φοιτητές ηλικίας 18 ετών γράφουμε:

```
SELECT *  
FROM Students S  
WHERE S.age=18
```

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2

Αποτέλεσμα

- Για να πάρουμε μόνο τα ονόματα και τα *logins*, αντικαθιστούμε την πρώτη γραμμή με:

```
SELECT S.name, S.login
```



Αιτήματα σε Πολλαπλές Σχέσεις

- Τι υπολογίζει η ακόλουθη ερώτηση;

```
SELECT S.name, E.cid  
FROM Students S, Enrolled E  
WHERE S.sid=E.sid AND E.grade='A'
```

Όταν δίνεται το ακόλουθο στιγμιότυπο της *Enrolled* (είναι δυνατό όταν το DBMS εξασφαλίζει **αναφορική ακεραιότητα**);

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Enrolled

sid	cid	grade
53831	Carnatic101	C
53831	Reggae203	B
53650	Topology112	A
53666	History105	B

αποτέλεσμα

S.name	E.cid
Smith	Topology112



Δημιουργία Σχέσεων με την SQL

- Δημιουργεί τη σχέση *Students*. Παρατηρήστε ότι προσδιορίζεται και ο τύπος (**domain**) κάθε πεδίου, και επιβάλλεται από το DBMS κάθε φορά που προστίθενται ή τροποποιούνται πλειάδες.
- Άλλο παράδειγμα: ο πίνακας *Enrolled* διατηρεί πληροφορίες σχετικά με τα μαθήματα που παίρνουν οι φοιτητές.

```
CREATE TABLE Students
(sid CHAR(20),
name CHAR(20),
login CHAR(10),
age INTEGER,
gpa REAL)
```

```
CREATE TABLE Enrolled
(sid CHAR(20),
cid CHAR(20),
grade CHAR(2))
```

Ακύρωση και Μεταβολή Σχέσεων

```
DROP TABLE Students
```

- Ακυρώνει τη σχέση *Students*. Η πληροφορία του σχήματος και οι πλειάδες διαγράφονται.

```
ALTER TABLE Students
ADD COLUMN firstYear integer
```

- ❖ Το σχήμα της σχέσης *Students* μεταβάλλεται και προστίθεται ένα νέο πεδίο. Κάθε πλειάδα στο τρέχον στιγμιότυπο επεκτείνεται με τη τιμή *null* στο νέο πεδίο.

Εισαγωγή και Διαγραφή Πλειάδων

- Μπορούμε να εισάγουμε μια πλειάδα ως εξής:

```
INSERT INTO Students (sid, name, login, age, gpa)
VALUES (53688, 'Smith', 'smith@ee', 18, 3.2)
```

- ❖ Μπορούμε να διαγράψουμε όλες τις πλειάδες που ικανοποιούν μια συνθήκη (π.χ., *name = 'Smith'*):

```
DELETE
FROM Students S
WHERE S.name = 'Smith'
```

- ☒ Παρέχονται επίσης ισχυρές παραλλαγές αυτών των εντολών. Περισσότερα αργότερα!

Περιορισμοί Ακεραιότητας (ICs)

- IC: συνθήκη που πρέπει να αληθεύει για κάθε στιγμιότυπο της ΒΔ (π.χ. περιορισμοί πεδίου ορισμού).
 - Τα ICs προσδιορίζονται όταν ορίζεται το σχήμα.
 - Τα ICs ελέγχονται όταν τροποποιούνται οι σχέσεις.
- **Έγκυρο** στιγμιότυπο μιας σχέσης είναι εκείνο που ικανοποιεί όλα τα ICs.
 - Τα DBMS δεν επιτρέπουν μη έγκυρα στιγμιότυπα.
- Αν ένα DBMS ελέγχει τα ICs, τα αποθηκευμένα δεδομένα είναι πιστότερα στη σημασία που έχουν στον πραγματικό κόσμο.
 - Επίσης αποφεύγονται λάθη στην εισαγωγή στοιχείων!

Περιορισμοί Κύριου Κλειδιού

- Ένα σύνολο πεδίων είναι **κλειδί** για μια σχέση αν:
 - Δεν επιτρέπεται σε δύο διαφορετικές πλειάδες να έχουν τις ίδιες τιμές σε όλα τα πεδία του κλειδιού, και
 - Το προηγούμενο δεν ισχύει για κανένα υποσύνολο του κλειδιού.
 - Αν το 2 δεν ισχύει τότε έχουμε ένα **εμπλουτισμένο κλειδί**.
 - Αν υπάρχουν περισσότερα από 1 **υποψήφια κλειδιά** για μια σχέση, τότε επιλέγεται (από τον DBA) ένα από αυτά να είναι το **κύριο κλειδί**.
- Π.χ., το *sid* είναι κλειδί για τη *Students*. (Τι λέτε για το *name*;) Το σύνολο {*sid*, *gra*} είναι ένα εμπλουτισμένο κλειδί.



Κύρια Κλειδιά στην SQL

- Η δήλωση του κλειδιού ενός πίνακα γίνεται με την **PRIMARY KEY** ενώ ακολουθεί το όνομα ή τα ονόματα των πεδίων που θα αποτελέσουν το κλειδί σε παρένθεση.

```
CREATE TABLE Students  
(sid CHAR(20),  
name CHAR(20),  
login CHAR(10),  
age INTEGER,  
gra REAL,  
PRIMARY KEY (sid))
```



Κύρια και Υποψήφια Κλειδιά στην SQL

- Πιθανώς πολλά **υποψήφια κλειδιά** (δηλώνεται με τη **UNIQUE**), ένα από αυτά επιλέγεται σαν **κύριο κλειδί**.
- “Για συγκεκριμένο ζευγάρι φοιτητή-μάθημα υπάρχει ένας μοναδικός βαθμός.” ή “Κάθε φοιτητής μπορεί να πάρει μόνο ένα μάθημα, και να λάβει ένα μοναδικό βαθμό σ’ αυτό. Επίσης, δεν υπάρχουν δύο φοιτητές με τον ίδιο βαθμό στο ίδιο μάθημα.”

```
CREATE TABLE Enrolled  
(sid CHAR(20),  
cid CHAR(20),  
grade CHAR(2),  
PRIMARY KEY (sid,cid))
```
- Απρόσεκτη χρήση ενός IC μπορεί να εμποδίσει την αποθήκευση χρήσιμων στιγμιότυπων!

```
CREATE TABLE Enrolled  
(sid CHAR(20),  
cid CHAR(20),  
grade CHAR(2),  
PRIMARY KEY (sid),  
UNIQUE (cid, grade))
```



Ξένο Κλειδιά, Αναφορική Ακεραιότητα

- Ξένο κλειδί**: Σύνολο πεδίων μιας σχέσης το οποίο χρησιμοποιείται για να `αναφέρεται` σε μια πλειάδα άλλης σχέσης σαν `λογικός δείκτης` (αντιστοιχεί στο κύριο κλειδί μιας άλλης σχέσης).
- Π.χ., το *sid* είναι ξένο κλειδί στη σχέση *Enrolled* αναφερόμενο στη σχέση *Students*:
 - Enrolled*(*sid*: string, *cid*: string, *grade*: string)
 - Με την επιβολή όλων των περιορισμών ξένου κλειδιού πετυχαίνουμε **αναφορική ακεραιότητα**, δηλ. αποφεύγονται αναφορές σε ανύπαρκτες πλειάδες.
 - Μοντέλο δεδομένων που δεν έχει αναφορική ακεραιότητα:
 - Σύνδεσμοι στην HTML!



Ένα Κλειδιά στην SQL

- Μόνο οι φοιτητές που έχουν καταχωρηθεί στη σχέση *Students* επιτρέπεται να εγγράφονται σε μαθήματα.

```
CREATE TABLE Enrolled
(sid CHAR(20), cid CHAR(20), grade CHAR(2),
PRIMARY KEY (sid,cid),
FOREIGN KEY (sid) REFERENCES Students )
```

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8



Επιβολή Αναφορικής Ακεραιότητας

- Έστω οι σχέσεις *Students* και *Enrolled*. Το *sid* της *Enrolled* είναι ξένο κλειδί και αναφέρεται στη *Students*.
- Τι πρέπει να γίνει αν γίνει προσπάθεια να εισαχθεί πλειάδα της *Enrolled* με ανύπαρκτο κωδικό φοιτητή (*id*); (*Να απορριφθεί!*)
- Τι πρέπει να γίνεται κατά τη διαγραφή πλειάδας της *Students*; Δυνατές επιλογές:
 - Να διαγραφούν όλες οι πλειάδες της *Enrolled* που αναφέρονται σ' αυτήν.
 - Να απαγορευτεί η διαγραφή της αναφερόμενης πλειάδας της *Students*.
 - Να τοποθετηθεί μια *προαποφασισμένη τιμή* στο *sid* όλων των πλειάδων της *Enrolled* που αναφέρονται σ' αυτήν.
 - Να τοποθετηθεί η ειδική τιμή *null*, που δηλώνει 'άγνωστο' ή 'μη εφαρμοσимо', στο *sid* των πλειάδων της *Enrolled* που αναφέρονται σ' αυτήν. Αυτό δεν επιτρέπεται εδώ γιατί το *sid* συμμετέχει στο κύριο κλειδί της *Enrolled*.
- Παρόμοια για την ενημέρωση του κύριου κλειδιού της *Students*.



Αναφορική Ακεραιότητα στην SQL/92 (1/2)

- Η SQL/92 υποστηρίζει και τις 4 δυνατότητες στη διαγραφή και την ενημέρωση.
 - Προαποφασισμένο είναι το **NO ACTION** (η *delete/update* απορρίπτεται)
 - CASCADE** (διαγράφει επίσης όλες τις πλειάδες που αναφέρονται σ' αυτήν που διαγράφεται)
 - SET NULL / SET DEFAULT** (δίνει τιμή στο ξένο κλειδί της πλειάδας – η προαποφασισμένη τιμή καθορίζεται κατά τη δήλωση του *sid* στην *Enrolled*)

```
CREATE TABLE Enrolled
(sid CHAR(20),
cid CHAR(20),
grade CHAR(2),
PRIMARY KEY (sid,cid),
FOREIGN KEY (sid)
REFERENCES Students
ON DELETE CASCADE
ON UPDATE SET DEFAULT)
```



Αναφορική Ακεραιότητα στην SQL/92 (2/2)

- Η διαγραφή πλειάδας του *Students* οδηγεί αυτόματα και στη διαγραφή των πλειάδων της *Enrolled* που αναφέρονται σε αυτήν.

```
CREATE TABLE Enrolled
(sid CHAR(20),
cid CHAR(20),
grade CHAR(2),
PRIMARY KEY (sid,cid),
FOREIGN KEY (sid)
REFERENCES Students
ON DELETE CASCADE
ON UPDATE CASCADE)
```

- Η ενημέρωση πλειάδας του *Students* με νέα τιμή *sid* οδηγεί αυτόματα και στην ενημέρωση όλων των πλειάδων της *Enrolled* που αναφέρονται σε αυτήν με τη νέα τιμή του *sid*.



Από που πηγάζουν οι περιορισμοί ακεραιότητας (ICs) ;

- Τα ICs βασίζονται στη σημασιολογία της επιχείρησης που περιγράφεται στις σχέσεις της ΒΔ.
- Μπορούμε να ελέγξουμε αν ένα στιγμιότυπο της ΒΔ παραβιάζει τα ICs, δεν μπορούμε όμως **ΠΟΤΕ** να συμπεράνουμε, εξετάζοντας το στιγμιότυπο, ότι ένα IC είναι αληθές.
 - Ένα IC αναφέρεται σε *όλα τα πιθανά* στιγμιότυπα!
 - Από το παράδειγμα γνωρίζουμε ότι το *name* δεν είναι κλειδί, όμως μας δηλώνεται ότι το *sid* είναι κλειδί.
- Τα ICs κλειδιού & ξένου κλειδιού χρησιμοποιούνται πιο συχνά, όμως υποστηρίζονται και πιο γενικά ICs.

Όψεις

- Μια **όψη** είναι μια σχέση, αλλά αντί για σύνολο πλειάδων, αποθηκεύουμε έναν **ορισμό (δήλωση)**.

```
CREATE VIEW YoungActiveStudents (name, grade)
AS SELECT S.name, E.grade
FROM Students S, Enrolled E
WHERE S.sid = E.sid AND S.age < 21
```

- ❖ Οι όψεις μπορούν να ακυρωθούν με την εντολή **DROP VIEW**.

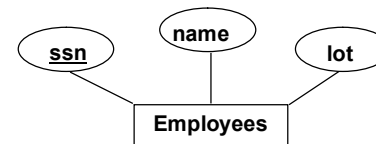
- Πως θα εκτελεστεί **DROP TABLE** αν υπάρχει όψη σ' αυτόν τον πίνακα;
 - Η εντολή **DROP TABLE** παρέχει επιλογές που επιτρέπουν στο χρήστη να το καθορίσει.

Όψεις και Ασφάλεια

- Οι όψεις μπορούν να χρησιμοποιηθούν για την παρουσίαση της αναγκαίας πληροφορίας (ή περιλήψεων), αποκρύπτοντας ταυτόχρονα τις λεπτομέρειες του σχέσεων που βασίζονται.
 - Όταν δίνεται η σχέση *YoungActiveStudents*, όχι όμως και οι σχέσεις *Students* και *Enrolled*, μπορούμε να βρούμε τα ονόματα των φοιτητών που έχουν εγγραφεί, αλλά όχι τα *cid's* των μαθημάτων στα οποία αυτοί έχουν εγγραφεί.

Λογικός Σχεδιασμός Βάσης Δεδομένων: Από το ER στο Σχεσιακό Σχήμα

- Από τα σύνολα οντοτήτων στους πίνακες.

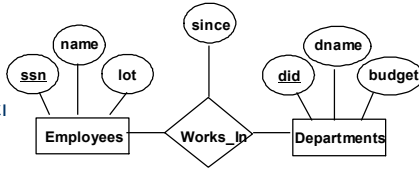


- Μια οντότητα μετατρέπεται σε έναν πίνακα με στήλες τα γνωρίσματα της οντότητας:

```
CREATE TABLE Employees
(ssn CHAR(11),
name CHAR(20),
lot INTEGER,
PRIMARY KEY (ssn))
```

Από τις Συσχετίσεις στους Πίνακες

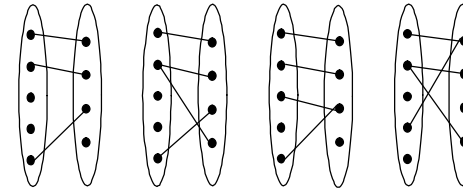
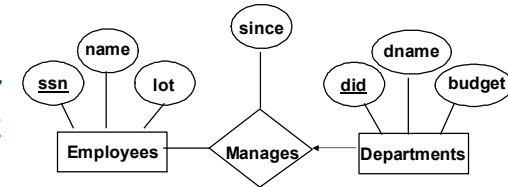
- Κατά τη μετατροπή συνόλου συσχετίσεων σε σχέση, τα γνωρίσματα της σχέσης πρέπει να περιλαμβάνουν:
 - Κλειδιά για κάθε σύνολο οντοτήτων που συμμετέχει (σαν ξένα κλειδιά).
 - Αυτό το σύνολο γνωρισμάτων αποτελεί ένα **εμπλουτισμένο κλειδί** για τη σχέση.
 - Όλα τα περιγραφικά γνωρίσματα της συσχέτισης.



```
CREATE TABLE Works_In(
  ssn CHAR(11),
  did INTEGER,
  since DATE,
  PRIMARY KEY (ssn, did),
  FOREIGN KEY (ssn)
    REFERENCES Employees,
  FOREIGN KEY (did)
    REFERENCES Departments)
```

Επισκόπηση: Περιορισμοί Κλειδιού

- Κάθε τμήμα έχει το πολύ ένα διευθυντή, με βάση τον **περιορισμό κλειδιού** στη σχέση **Manages**.

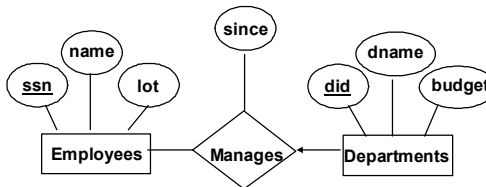


1-προς-1 1-προς Πολλά Πολλά-προς-1 Πολλά-προς-Πολλά

Πως μετατρέπονται στο σχεσιακό μοντέλο;

Μετατρέποντας ER διαγράμματα με Περιορισμούς Κλειδιού (1/2)

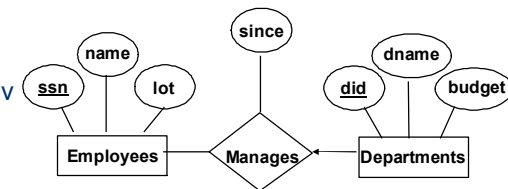
- 1ος τρόπος** (όπως και πριν):
 - Η συσχέτιση απεικονίζεται σαν πίνακας:
 - Ξεχωριστοί πίνακες *Employees* και *Departments*.
 - Προσοχή: το *did* είναι τώρα το κλειδί!



```
CREATE TABLE Manages(
  ssn CHAR(11),
  did INTEGER,
  since DATE,
  PRIMARY KEY (did),
  FOREIGN KEY (ssn) REFERENCES Employees,
  FOREIGN KEY (did) REFERENCES Departments)
```

Μετατρέποντας ER διαγράμματα με Περιορισμούς Κλειδιού (2/2)

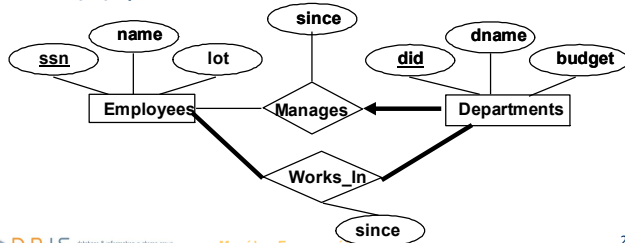
- 2ος τρόπος** Αφού κάθε τμήμα έχει έναν μοναδικό διευθυντή (το πολύ), θα μπορούσαμε εναλλακτικά να συνδυάσουμε τους *Manages* και *Departments* δημιουργώντας έναν ενιαίο πίνακα.



```
CREATE TABLE Dept_Mgr(
  did INTEGER,
  dname CHAR(20),
  budget REAL,
  ssn CHAR(11),
  since DATE,
  PRIMARY KEY (did),
  FOREIGN KEY (ssn) REFERENCES Employees)
```

Επισκόπηση: Περιορισμοί Συμμετοχής

- Έχει κάθε τμήμα οπωσδήποτε ένα διευθυντή;
 - Αν ναι, τότε αυτό αποτελεί **περιορισμό συμμετοχής**: η συμμετοχή του πίνακα *Departments* στο *Manages* ονομάζεται **ολική** (σε αντίθεση με τη **μερική**).
 - Κάθε τιμή του *did* στον πίνακα *Departments* πρέπει να εμφανίζεται σε μια γραμμή του *Manages* (με τιμή του *ssn* διάφορη του null!)



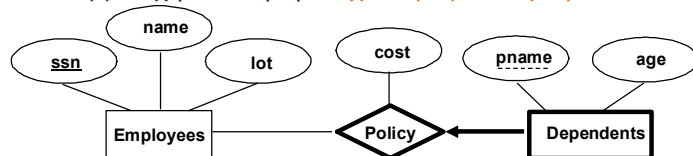
Περιορισμοί Συμμετοχής στην SQL

- Μπορούμε να ορίσουμε περιορισμούς συμμετοχής που περιλαμβάνουν ένα σύνολο οντοτήτων σε μια δυαδική συσχέτιση, όχι όμως πολλά πέρα από αυτά (χωρίς προσφυγή σε CHECK περιορισμούς).

```
CREATE TABLE Dept_Mgr(
  did INTEGER,
  dname CHAR(20),
  budget REAL,
  ssn CHAR(11) NOT NULL,
  since DATE,
  PRIMARY KEY (did),
  FOREIGN KEY (ssn) REFERENCES Employees,
  ON DELETE NO ACTION)
```

Επισκόπηση: Αδύναμες Οντότητες

- Μια **αδύναμη οντότητα** μπορεί να προσδιοριστεί μοναδικά μόνο δια μέσου του κύριου κλειδιού μια άλλης οντότητας (**προσδιορίζων ιδιοκτήτης**).
 - Το σύνολο οντοτήτων ιδιοκτήτη και αδύναμων οντοτήτων πρέπει να συμμετέχουν σε σύνολο συσχετίσεων 1-προς-πολλά (1 ιδιοκτήτης, πολλές αδύναμες οντότητες).
 - Το σύνολο αδύναμων οντοτήτων πρέπει να έχει ολική συμμετοχή σ' αυτή τη **συσχέτιση προσδιορισμού**.

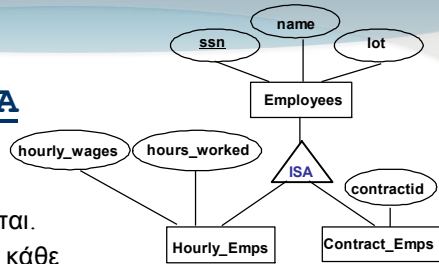


Μετατροπή Συνόλου Αδύναμων Οντοτήτων

- Ένα σύνολο αδύναμων οντοτήτων από κοινού με το προσδιορίζων σύνολο συσχετίσεων μετατρέπονται σε έναν πίνακα.
 - Όταν ο προσδιορίζων ιδιοκτήτης διαγράφεται, όλες οι αδύναμες οντότητες που του ανήκουν πρέπει να διαγραφούν.

```
CREATE TABLE Dep_Policy (
  pname CHAR(20),
  age INTEGER,
  cost REAL,
  ssn CHAR(11) NOT NULL,
  PRIMARY KEY (pname, ssn),
  FOREIGN KEY (ssn) REFERENCES Employees,
  ON DELETE CASCADE)
```

Επισκόπηση: Ιεραρχίες ISA



❖ Όπως και στη C++, τα γνωρίσματα κληρονομούνται.

❖ Αν δηλώσουμε A ISA B, κάθε οντότητα του A θεωρείται επίσης οντότητα του B.

- **Περιορισμοί επικάλυψης:** Μπορεί ο John να ανήκει ταυτόχρονα και στο σύνολο οντοτήτων *Hourly_Emps* και στο *Contract_Emps*; (Επιτρέπεται/απαγορεύεται)
- **Περιορισμοί κάλυψης:** Πρέπει κάθε οντότητα *Employees* να ανήκει υποχρεωτικά είτε στο *Hourly_Emps* είτε στο *Contract_Emps*; (Ναι/όχι)

Μετατροπή Ιεραρχιών ISA σε Σχέσεις

▪ Γενική προσέγγιση:

- 3 σχέσεις: *Employees*, *Hourly_Emps* και *Contract_Emps*.
 - *Hourly_Emps*: Κάθε εργαζόμενος καταχωρείται στον *Employees*. Για τους ωρομισθίους, καταχωρείται επιπλέον πληροφορία στον *Hourly_Emps* (*hourly_wages*, *hours_worked*, *ssn*).
 - Διαγραφή πλειάδας του πίνακα *Employees* οδηγεί σε διαγραφή και της πλειάδας του *Hourly_Emps* που αναφέρεται σ' αυτήν.
 - Διευκολύνονται τα αιτήματα που αναφέρονται σε όλους τους εργαζόμενους, ενώ αυτά που αναφέρονται μόνο στον *Hourly_Emps* απαιτούν χρήση του join.
- Εναλλακτικά: Μόνο *Hourly_Emps* και *Contract_Emps*.
 - *Hourly_Emps*: *ssn*, *name*, *lot*, *hourly_wages*, *hours_worked*.
 - Κάθε εργαζόμενος πρέπει να ανήκει σε μια από αυτές τις υποκατηγορίες.

Σχεσιακό Μοντέλο: Περίληψη

- Αναπαράσταση των δεδομένων υπό μορφή πινάκων.
- Απλό και διαισθητικό, το πιο διαδεδομένο σήμερα.
- Περιορισμοί ακεραιότητας (ICs) που βασίζονται στη σημασία των εφαρμογών μπορούν να ορίζονται από τον DBA. Το DBMS ελέγχει για παραβιάσεις.
 - Δύο ενδιαφέρουσες κατηγορίες ICs: περιορισμοί κύριου κλειδιού και περιορισμοί ξένου κλειδιού.
 - Επίσης, έχουμε πάντα περιορισμούς πεδίου ορισμού.
- Υπάρχουν ισχυρές και φυσικές γλώσσες αιτημάτων.
- Υπάρχουν κανόνες για μετατροπή από ER σε σχεσιακό μοντέλο.

Στοιχεία Σχεσιακής Άλγεβρας (Κεφάλαιο 4α)

Μανόλης Γεργατσούλης

Καθηγητής

Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας &
Μουσειολογίας

Ιόνιο Πανεπιστήμιο

manolis@ionio.gr



Σχεσιακές Γλώσσες Αιτημάτων

- **Γλώσσες αιτημάτων:** Επιτρέπουν το χειρισμό και την ανάκτηση των δεδομένων από μια βάση δεδομένων.
- Το σχεσιακό μοντέλο υποστηρίζει απλές, αλλά ισχυρές γλώσσες αιτημάτων:
 - Διαθέτουν αυστηρό μαθηματικό υπόβαθρο που βασίζεται στη (μαθηματική) λογική.
 - Επιτρέπουν τη βελτιστοποίηση των αιτημάτων.
- Οι Γλώσσες αιτημάτων διαφέρουν από τις γλώσσες προγραμματισμού:
 - Οι γλώσσες αιτημάτων (ΓΑ) δεν αναμένεται να είναι "Turing complete".
 - Οι ΓΑ δεν φτιάχτηκαν για την υποστήριξη πολύπλοκων υπολογισμών.
 - Οι ΓΑ υποστηρίζουν την εύκολη και αποδοτική προσπέλαση σε μεγάλες συλλογές δεδομένων.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

2

Τυπικές Σχεσιακές Γλώσσες Αιτημάτων

- Δύο μαθηματικές γλώσσες αιτημάτων αποτελούν τη βάση για τις "πραγματικές" γλώσσες (π.χ. SQL), καθώς και για τις υλοποιήσεις των γλωσσών αυτών:
 1. **Σχεσιακή άλγεβρα:** Πιο διαδικαστική, ιδιαίτερα χρήσιμη για την αναπαράσταση σχεδίων επεξεργασίας αιτημάτων.
 2. **Σχεσιακός λογισμός:** Επιτρέπει στους χρήστες να περιγράφουν τι θέλουν, αντί για το πως θα υπολογιστεί αυτό (*Μη-διαδικαστική, δηλωτική γλώσσα*).

✉ Η κατανόηση Σχεσιακής Άλγεβρας και Λογισμού είναι το κλειδί για την κατανόηση της SQL, και της επεξεργασίας των αιτημάτων!



Μανόλης Γεργατσούλης
Οκτώβριος 2015

3

Προκαταρκτικά

- Ένα αίτημα εφαρμόζεται σε **στιγμιότυπα σχέσεων**, και το αποτέλεσμα του αιτήματος είναι επίσης στιγμιότυπο σχέσης.
- Το **σχήμα των σχέσεων εισόδου** ενός αιτήματος είναι **αμετάβλητο** (αλλά το αίτημα θα "τρέχει" ανεξάρτητα από το στιγμιότυπο!)
 - Το **σχήμα του αποτελέσματος** δεδομένου αιτήματος είναι επίσης **αμετάβλητο!** Καθορίζεται εξ ορισμού από τις συντακτικές δομές της γλώσσας αιτημάτων.
- Η αναπαράσταση των πεδίων μπορεί να βασίζεται στη θέση ή στην ονομασία των πεδίων:
 - Η αναπαράσταση που βασίζεται στη θέση διευκολύνει τους αυστηρούς ορισμούς, ενώ η αναπαράσταση που βασίζεται στις ονομασίες των πεδίων είναι πιο ευανάγνωστη.
 - Και οι δύο χρησιμοποιούνται στην SQL.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

4

Παραδείγματα Στιγμιότυπων (1/2)

R1	sid	bid	day
	22	101	10/10/96
	58	103	11/12/96

- Στιγμιότυπα των σχέσεων "Sailors" και "Reserves" για τις S1 ανάγκες των παραδειγμάτων μας.
- Θα χρησιμοποιήσουμε και τις δύο αναπαραστάσεις για τα πεδία (ονομασία, θέση), και θα υποθέσουμε ότι τα ονόματα των πεδίων στα αποτελέσματα 'κληρονομούνται' από τα ονόματα των πεδίων στις σχέσεις εισόδου των αιτημάτων.

S1	sid	sname	rating	age
	22	dustin	7	45.0
	31	lubber	8	55.5
	58	rusty	10	35.0

S2	sid	sname	rating	age
	28	yuppy	9	35.0
	31	lubber	8	55.5
	44	guppy	5	35.0
	58	rusty	10	35.0



Παραδείγματα Στιγμιότυπων (2/2)

- Στα παραδείγματά μας θα χρησιμοποιήσουμε επίσης τον πίνακα "Boats" που περιλαμβάνει πληροφορίες για τις βάρκες (κωδικός, όνομα, χρώμα).

B1	bid	bname	color
	101	Interlake	blue
	102	Interlake	red
	103	Clipper	green
	104	Marine	red



Σχεσιακή Άλγεβρα

- Βασικοί τελεστές:
 - Επιλογή** (σ) Επιλέγει ένα υποσύνολο γραμμών από μια σχέση.
 - Προβολή** (π) Διαγράφει ανεπιθύμητες στήλες από μια σχέση.
 - Καρτεσιανό γινόμενο** (\times) Επιτρέπει να συνδυάσουμε δύο σχέσεις παίρνοντας μια νέα σχέση.
 - Διαφορά** (\rightarrow) Πλειάδες στη σχέση 1, αλλά όχι στη σχέση 2.
 - Ένωση** (\cup) Πλειάδες στη σχέση 1 και στη σχέση 2.
- Επιπλέον τελεστές:
 - Τομή, σύζευξη (join), διαίρεση, μετονομασία:** Όχι απαραίτητοι, αλλά (πολύ!) χρήσιμοι.
- Επειδή κάθε πράξη επιστρέφει μια σχέση, επιτρέπεται η **σύνθεση** τελεστών! (Η άλγεβρα είναι "κλειστή".)



Προβολή

- Διαγράφει τα πεδία που δεν ανήκουν στη λίστα προβολής.
- Το **σχήμα** του αποτελέσματος περιέχει ακριβώς τα πεδία της λίστας προβολής, με τα ίδια ονόματα τα οποία είχαν στην αρχική σχέση.
- Απαλείφονται οι **διπλοεγγραφές** από το αποτέλεσμα!
 - Σημείωση: τα πραγματικά συστήματα συνήθως δεν απαλείφουν τις πολλαπλές εμφανίσεις εκτός αν το ζητήσει ο χρήστης.

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

$\pi_{sname, rating}(S2)$

age
35.0
55.5

$\pi_{age}(S2)$



Επιλογή

- Επιλέγει τις πλειάδες που ικανοποιούν τη **συνθήκη επιλογής**.
- Απαλείφονται οι διπλοεγγραφές από το αποτέλεσμα!
- Το **σχήμα** του αποτελέσματος είναι ταυτόσημο με το σχήμα της σχέσης εισόδου.
- Η σχέση που προκύπτει σαν αποτέλεσμα μπορεί να είναι **είσοδος** σε μια άλλη πράξη της σχεσιακής άλγεβρας! (Σύνθεση τελεστών.)

sid	sname	rating	age
28	yuppy	9	35.0
58	rusty	10	35.0

$$\sigma_{rating > 8}(S2)$$

sname	rating
yuppy	9
rusty	10

$$\pi_{sname, rating}(\sigma_{rating > 8}(S2))$$



Ένωση, Τομή, Διαφορά

- Εφαρμόζονται σε δύο σχέσεις, που πρέπει να είναι **συμβατές-ως-προς-την-ένωση**:
 - Ίδιος αριθμός πεδίων.
 - Τα αντίστοιχα πεδία έχουν τον ίδιο τύπο.
- Ποιο είναι το **σχήμα** του αποτελέσματος;

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$$S1 \cup S2$$

sid	sname	rating	age
22	dustin	7	45.0

$$S1 - S2$$

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

$$S1 \cap S2$$



Καρτεσιανό Γινόμενο

- Κάθε γραμμή του S1 συνδυάζεται με κάθε γραμμή του R1.
- Σχήμα αποτελέσματος** περιλαμβάνει ένα πεδίο για κάθε πεδίο του S1 και του R1, τα ονόματα των πεδίων 'κληρονομούνται' αν είναι δυνατόν.
 - Σύγκρουση ονομασίας**: S1 και R1 έχουν πεδίο με όνομα *sid*.

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

☒ **Τελεστής μετονομασίας**: $\rho(C(1 \rightarrow sid\ 1, 5 \rightarrow sid\ 2), S1 \times R1)$



Σύζευξη (1/2)

- Σύζευξη**: καρτεσιανό γινόμενο ακολουθούμενο από προβολές και επιλογές.
- Σύζευξη Συνθήκης**: $R \bowtie_c S = \sigma_c(R \times S)$

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	58	103	11/12/96

$$S1 \bowtie_{S1.sid < R1.sid} R1$$

- Σχήμα αποτελέσματος**: όπως και στο καρτεσιανό γινόμενο.
- Λιγότερες πλειάδες από το καρτεσιανό γινόμενο, πιθανόν αποδοτικότερος υπολογισμός.



Σύζευξη (2/2)

- **Σύζευξη Ισότητας:** Ειδική περίπτωση σύζευξης συνθήκης κατά την οποία η συνθήκη c περιλαμβάνει μόνο **ισότητες** (μεταξύ πεδίων των R και S).

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

$$S1 \bowtie R1 \quad S1.sid = R1.sid$$

- **Σχήμα αποτελέσματος:** όπως και στο καρτεσιανό γινόμενο, αλλά παίρνουμε μόνο ένα αντίγραφο του πεδίου για κάθε ισότητα.
- **Φυσική Σύζευξη:** Σύζευξη ισότητας πάνω σε όλα τα κοινά πεδία.



Βρείτε να ονόματα των ναυτικών που έχουν κάνει κράτηση στη βάρκα με κωδικό 103

- Λύση 1: $\pi_{sname}((\sigma_{bid=103} Reserves) \bowtie Sailors)$
- ❖ Λύση 2: $\rho(Temp1, \sigma_{bid=103} Reserves)$
 $\rho(Temp2, Temp1 \bowtie Sailors)$
 $\pi_{sname}(Temp2)$
- ❖ Λύση 3: $\pi_{sname}(\sigma_{bid=103}(Reserves \bowtie Sailors))$



Βρείτε να ονόματα των ναυτικών που έχουν κάνει κράτηση σε κόκκινη βάρκα

- Πληροφορία που αφορά το χρώμα μιας βάρκας είναι διαθέσιμη μόνο στη σχέση *Boats*, επομένως απαιτείται μια επιπλέον σύζευξη:

$$\pi_{sname}((\sigma_{color='red'} Boats) \bowtie Reserves \bowtie Sailors)$$

- ❖ Μια πιο αποδοτική λύση*:

$$\pi_{sname}(\pi_{sid}((\pi_{bid} \sigma_{color='red'} Boats) \bowtie Reserves) \bowtie Sailors)$$

☒ Ένα σύστημα βελτιστοποίησης θα μπορούσε να τη βρει όταν του δοθεί η 1η λύση!



Βρείτε να ονόματα των ναυτικών που έχουν κάνει κράτηση είτε σε κόκκινη είτε σε πράσινη βάρκα

- Μπορούμε να εντοπίσουμε όλες τις βάρκες που έχουν κόκκινο ή πράσινο χρώμα και στη συνέχεια να βρούμε τους ναυτικούς που έχουν κάνει κράτηση σε μια από αυτές:
 $\rho(Tempboats, (\sigma_{color='red'} \vee \sigma_{color='green'} Boats))$
 $\pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$
- ❖ Μπορούμε επίσης να ορίσουμε την *Tempboats* μέσω της ένωσης! (Πως;)
- ❖ Τι συμβαίνει αν αντικατασταθεί το \setminus με το $/$ στο αίτημα;



Βρείτε να ονόματα των ναυτικών που έχουν κάνει κράτηση και σε κόκκινη και σε πράσινη βάρκα

- Η προηγούμενη προσέγγιση δεν δουλεύει! Πρέπει αν εντοπίσουμε τους ναυτικούς που έχουν κάνει κράτηση σε κόκκινη βάρκα, καθώς και τους ναυτικούς που έχουν κάνει κράτηση σε πράσινη βάρκα, και να βρούμε τη τομή τους (σημειώστε ότι το *sid* είναι κλειδί για την *Sailors*):

$\rho (Tempred, \pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves))$

$\rho (Tempgreen, \pi_{sid}((\sigma_{color='green'} Boats) \bowtie Reserves))$

$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailors)$

Περίληψη

- Το σχεσιακό μοντέλο διαθέτει αυστηρά ορισμένες γλώσσες αιτημάτων οι οποίες είναι απλές αλλά ισχυρές.
- Η σχεσιακή άλγεβρα είναι πιο διαδικαστική, και χρήσιμη σαν εσωτερική αναπαράσταση σχεδίων επεξεργασίας αιτημάτων.
- Γενικά ένα αίτημα μπορεί να διατυπωθεί με πολλούς διαφορετικούς τρόπους. Ένα σύστημα βελτιστοποίησης αιτημάτων θα μπορούσε να επιλέξει την πιο αποδοτική εκδοχή.

SQL: Αιτήματα (Κεφάλαιο 5)

Μανόλης Γεργατσούλης

Καθηγητής

Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας &
Μουσειολογίας

Ιόνιο Πανεπιστήμιο

manolis@ionio.gr



Στιγμιότυπα

του

Παραδείγματος

R1	sid	bid	day
	22	101	10/10/96
	58	103	11/12/96

- Στα παραδείγματα μας θα χρησιμοποιήσουμε τα παρόντα στιγμιότυπα των σχέσεων *Sailors*, *Reserves*, και *Boats*.

S1	sid	sname	rating	age
	22	dustin	7	45.0
	31	lubber	8	55.5
	58	rusty	10	35.0

- Ποια θα ήταν η διαφορά στη σημασιολογία αν το κλειδί της σχέσης *Reserves* περιελάμβανε μόνο τα πεδία *sid* και *bid*;

B1	bid	bname	color
	101	Interlake	blue
	102	Interlake	red
	103	Clipper	green
	104	Marine	red



Μανόλης Γεργατσούλης
Οκτώβριος 2015

2

Βασικό SQL

Αίτημα

```
SELECT [DISTINCT] λίστα-επιλογής  
FROM λίστα-πινάκων  
WHERE καταλληλότητα
```

- λίστα-πινάκων:** λίστα ονομάτων πινάκων (καθένα όνομα μπορεί να ακολουθείται από μια *μεταβλητή διαστήματος*).
- λίστα-επιλογής:** λίστα από ονόματα στηλών (ή παραστάσεις που περιέχουν ονόματα στηλών) που ανήκουν σε πίνακες που περιλαμβάνονται στη *λίστα-πινάκων*.
- καταλληλότητα:** Συνθήκες της μορφής *παρασταση op παρασταση* όπου το *op* μπορεί να είναι ένας από τους *<*, *>*, *=*, *<=*, *>=*, *<>*.
παρασταση μπορεί να αποτελέσει το όνομα μιας στήλης, μια σταθερά, μια αριθμητική παράσταση ή μια συμβολοσειρά. Οι παραστάσεις συνδυάζονται με τους λογικούς τελεστές *AND*, *OR* και *NOT*.
- Η χρήση του όρου *DISTINCT* είναι προαιρετική και δείχνει ότι η απάντηση δε θα πρέπει να περιέχει επαναλήψεις. Χωρίς το *DISTINCT* οι πολλαπλές εμφανίσεις δεν απαλείφονται.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

3

Ιδεατή Στρατηγική Υπολογισμού

- Η σημασιολογία ενός αιτήματος SQL ορίζεται με βάση την ακόλουθη ιδεατή στρατηγική υπολογισμού:
 - Υπολογίζεται το καρτεσιανό γινόμενο των πινάκων στη *λίστα-πινάκων*.
 - Απαλείφονται οι πλειάδες που προκύπτουν οι οποίες δεν ικανοποιούν την *καταλληλότητα*.
 - Διαγράφονται οι στήλες που δεν βρίσκονται στην *λίστα-επιλογής*.
 - Αν το αίτημα περιλαμβάνει το *DISTINCT*, απαλείφονται οι πολλαπλές εμφανίσεις της ίδιας πλειάδας.
- Η στρατηγική αυτή είναι πιθανά η λιγότερο αποδοτική για τον υπολογισμό του αιτήματος! Ένας βελτιστοποιητής θα έβρισκε πιο αποδοτικές στρατηγικές για να υπολογίσει τις *ίδιες απαντήσεις*.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

4

Παράδειγμα Ιδεατού Υπολογισμού

Sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Reserves

sid	bid	day
22	101	10/10/96
58	103	11/12/96

```
SELECT S.sname
FROM Sailors S, Reserves R
WHERE S.sid=R.sid AND R.bid=103
```

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

S.sid=R.sid

R.bid=103



DBIS

database information systems group

Μανόλης Γεργασιούλης

Οκτώβριος 2015

5

Σχετικά με τις μεταβλητές διαστήματος

- Στην πραγματικότητα χρειάζονται μόνο όταν ο ίδιος πίνακα εμφανίζεται δύο φορές στη συνιστώσα FROM. Η προηγούμενη ερώτηση μπορεί επίσης να διατυπωθεί ως εξής:

```
SELECT S.sname
FROM Sailors S, Reserves R
WHERE S.sid=R.sid AND bid=103
```

Είναι καλό όμως, να χρησιμοποιούμε πάντα μεταβλητές διαστήματος!

```
H SELECT sname
FROM Sailors, Reserves
WHERE Sailors.sid=Reserves.sid
AND bid=103
```



DBIS

database information systems group

Μανόλης Γεργασιούλης

Οκτώβριος 2015

6

Να βρεθούν οι κωδικοί των ναυτικών που έχουν κάνει κράτηση τουλάχιστον σε μια βάρκα

Sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Reserves

sid	bid	day
22	101	10/10/96
58	103	11/12/96

```
SELECT S.sid
FROM Sailors S, Reserves R
WHERE S.sid=R.sid
```

- Θα είχε καμιά επίδραση αν προσθέταμε DISTINCT σ' αυτό το αίτημα;
- Τι θα συνέβαινε αν αντικαθιστούσαμε το S.sid με το S.sname στην εντολή SELECT; Θα είχε καμιά επίδραση αν προσθέταμε DISTINCT σ' αυτή την παραλλαγή του αιτήματος;

Αποτέλεσμα

sid
22
58



DBIS

database information systems group

Μανόλης Γεργασιούλης

Οκτώβριος 2015

7

Παραστάσεις και Συμβολοσειρές

```
SELECT S.age, age1=S.age-5, 2*S.age AS age2
FROM Sailors S
WHERE S.sname LIKE 'B_%B'
```

- Δείχνει τη χρήση των αριθμητικών παραστάσεων και του ταιριάσματος συμβολοσειρών: *Να βρείτε τριάδες (ηλικία ναυτικού και δύο πεδία που ορίζονται μέσω παραστάσεων) για τους ναυτικούς που το όνομα τους αρχίζει και τελειώνει με B και αποτελείται από τουλάχιστον τρεις χαρακτήρες.*
- AS και = είναι δύο τρόποι να δώσουμε ονόματα στα πεδία του αποτελέσματος.
- LIKE χρησιμοποιείται για το ταιρίασμα συμβολοσειρών. Το `_' αντιπροσωπεύει έναν χαρακτήρα ενώ το `%` παριστάνει 0 ή περισσότερους χαρακτήρες. Κάποια συστήματα χρησιμοποιούν το `*` αντί του `%` και το `?` αντί του `_'.



DBIS

database information systems group

Μανόλης Γεργασιούλης

Οκτώβριος 2015

8

Να βρεθούν τα sid's των ναυτικών που έχουν κάνει κράτηση σε κόκκινη ή πράσινη βάρκα

- UNION: Μπορεί να χρησιμοποιηθεί για τον υπολογισμό της ένωσης δύο συμβατών ως προς την ένωση συνόλων πλειάδων (τα οποία αποτελούν αποτελέσματα αιτημάτων SQL).
- Αν αντικαταστήσουμε το OR με το AND στην πρώτη εκδοχή, τι προκύπτει;

```
SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid
AND (B.color='red' OR B.color='green')
```

```
SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid
AND B.color='red'
```

```
UNION
SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid
AND B.color='green'
```



Να βρεθούν τα sid των ναυτικών που έχουν κάνει κράτηση σε κόκκινη και σε πράσινη βάρκα

- INTERSECT: Μπορεί να χρησιμοποιηθεί για τον υπολογισμό της τομής δύο συνόλων πλειάδων που είναι συμβατά ως προς την ένωση.
- Περιλαμβάνεται στο πρότυπο SQL/92, όμως κάποια συστήματα δεν το υποστηρίζουν.
- Συγκρίνετε τη συμμετρία των αιτημάτων που χρησιμοποιούν UNION και INTERSECT με το πόσο διαφέρουν οι αντίστοιχες εκδόσεις των αιτημάτων που δεν τα χρησιμοποιούν.

```
SELECT S.sid
FROM Sailors S, Boats B1, Reserves R1,
Boats B2, Reserves R2
WHERE S.sid=R1.sid AND R1.bid=B1.bid
AND S.sid=R2.sid AND R2.bid=B2.bid
AND (B1.color='red' AND B2.color='green')
```

Πεδίο κλειδί!

```
SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid
AND B.color='red'
```

```
INTERSECT
SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid
AND B.color='green'
```



Εμφωλευμένα Αιτήματα

Βρείτε τα ονόματα των ναυτικών που έκαναν κράτηση στη βάρκα με κωδικό 103:

```
SELECT S.sname
FROM Sailors S
WHERE S.sid IN (SELECT R.sid
FROM Reserves R
WHERE R.bid=103)
```

- Ένα ισχυρό χαρακτηριστικό της SQL: η συνιστώσα WHERE μπορεί με τη σειρά της να περιέχει ένα αίτημα SQL! (Το ίδιο ισχύει και για τις συνιστώσες FROM και HAVING.)
- Για να βρούμε τους ναυτικούς που δεν έχουν κάνει κράτηση στην βάρκα με κωδικό 103, χρησιμοποιούμε το NOT IN.
- Για να κατανοήσουμε τη σημασία των εμφωλευμένων αιτημάτων, σκεφτείτε τον υπολογισμό μιας εμφωλευμένης επανάληψης ως εξής: Για κάθε πλειάδα του πίνακα Sailors, να ελεγχθεί η καταλληλότητα υπολογίζοντας το υπό-αίτημα.



Εμφωλευμένα Αιτήματα με Αμοιβαία

Εξάρτηση

Βρείτε τα ονόματα των ναυτικών που έκαναν κράτηση στη βάρκα με κωδικό 103:

```
SELECT S.sname
FROM Sailors S
WHERE EXISTS (SELECT *
FROM Reserves R
WHERE R.bid=103 AND S.sid=R.sid)
```

- Το EXISTS είναι ένας τελεστής σύγκρισης πάνω σε σύνολα, όπως και το IN.
- Χρησιμοποιώντας το UNIQUE, και αντικαθιστώντας το * με το R.bid, βρίσκουμε τους ναυτικούς που έχουν κάνει το πολύ μια κράτηση για τη βάρκα με κωδικό 103. (το UNIQUE ελέγχει για πολλαπλές πλειάδες, ενώ το * υποδηλώνει όλα τα πεδία. Γιατί πρέπει να αντικαταστήσουμε το * με το R.bid;)
- Δείχνει γιατί, στη γενική περίπτωση, το υπό-αίτημα πρέπει να ξανά-υπολογιστεί για κάθε πλειάδα του πίνακα Sailors.



Περισσότερα Σχετικά με Τελεστές Σύγκρισης Συνόλων

- Έχουμε ήδη συναντήσει τους **IN**, **EXISTS** και **UNIQUE**. Μπορούμε ακόμη να χρησιμοποιήσουμε τους **NOT IN**, **NOT EXISTS** και **NOT UNIQUE**.
- Υπάρχουν επίσης τα: **or ANY**, **or ALL** όπου **or** είναι ένα από τα **<**, **>**, **=**, **<=**, **>=**, **<>**.
- Να βρεθούν οι ναυτικοί με τιμή διατίμησης (rating) μεγαλύτερη από εκείνη ενός ναυτικού με το όνομα Horatio:

```
SELECT *
FROM Sailors S
WHERE S.rating > ANY (SELECT S2.rating
FROM Sailors S2
WHERE S2.sname='Horatio')
```



Επαναδιατύπωση Αιτημάτων INTERSECT με χρήση του IN

Να βρεθούν τα sid των ναυτικών που έχουν κάνει κράτηση και σε κόκκινη και σε πράσινη βάρκα:

```
SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid AND B.color='red'
AND S.sid IN (SELECT S2.sid
FROM Sailors S2, Boats B2, Reserves R2
WHERE S2.sid=R2.sid AND R2.bid=B2.bid
AND B2.color='green')
```

- Παρομοίως, αιτήματα **EXCEPT** μπορούν να διατυπωθούν με χρήση του **NOT IN**.
- Για να βρούμε τα ονόματα (και όχι τα sid) των ναυτικών που έχουν κάνει κράτηση και σε κόκκινη και σε πράσινη βάρκα, αντικαθιστούμε το **S.sid** με το **S.sname** στη συνιστώσα **SELECT**. (Τι γίνεται με αιτήματα **INTERSECT**;))



Διαίρεση στην SQL

Να βρεθούν ναυτικοί που έχουν κάνει κράτηση σε όλες τις βάρκες.

- Ας δούμε το δύσκολο τρόπο, χωρίς το **EXCEPT**:

(2) SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (SELECT B.bid
FROM Boats B

Ναυτικοί S τέτοιοι ώστε ...
δεν υπάρχει βάρκα B χωρίς ...
WHERE NOT EXISTS (SELECT R.bid
FROM Reserves R
WHERE R.bid=B.bid
AND R.sid=S.sid))

για πλειάδα του Reserves που δείχνει ότι ο S έχει κάνει κράτηση στη B



(1)

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS
((SELECT B.bid
FROM Boats B)
EXCEPT
(SELECT R.bid
FROM Reserves R
WHERE R.sid=S.sid))
```

Τελεστές Συνάθροισης

- Σημαντική επέκταση της σχεσιακής άλγεβρας.

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

Απλή στήλη

```
SELECT COUNT (*)
FROM Sailors S
```

```
SELECT AVG (S.age)
FROM Sailors S
WHERE S.rating=10
```

```
SELECT S.sname
FROM Sailors S
WHERE S.rating=(SELECT MAX(S2.rating)
FROM Sailors S2)
```

```
SELECT COUNT (DISTINCT S.rating)
FROM Sailors S
WHERE S.sname='Bob'
```

```
SELECT AVG (DISTINCT S.age)
FROM Sailors S
WHERE S.rating=10
```



Να βρεθεί το όνομα και η ηλικία του πλέον ηλικιωμένου ναυτικού

- Το πρώτο αίτημα δεν είναι σωστό! (Θα δούμε αργότερα το γιατί, όταν θα συζητήσουμε για το **GROUP BY**)

```
SELECT S.sname, MAX (S.age)
FROM Sailors S
```
- Το τρίτο αίτημα είναι ισοδύναμο με το δεύτερο, και είναι επιτρεπτό στο πρότυπο SQL/92, αλλά δεν υποστηρίζεται από μερικά συστήματα.

```
SELECT S.sname, S.age
FROM Sailors S
WHERE S.age =
(SELECT MAX (S2.age)
FROM Sailors S2)
```



```
SELECT S.sname, S.age
FROM Sailors S
WHERE (SELECT MAX (S2.age)
FROM Sailors S2)
= S.age
```

GROUP BY και HAVING

- Μέχρι τώρα έχουμε εφαρμόσει τελεστές συνάθροισης σε όλες (τις προσδιοριζόμενες) πλειάδες. Μερικές φορές θέλουμε να τους εφαρμόσουμε σε κάθε μια από διάφορες ομάδες πλειάδων.
- Έστω το αίτημα: *Να βρεθεί η ηλικία του νεώτερου ναυτικού για κάθε τιμή διατίμησης (rating).*
 - Γενικά δεν γνωρίζουμε πόσες τιμές διατίμησης (ratings) υπάρχουν και ποιες είναι οι τιμές αυτές!
 - Έστω ότι γνωρίζουμε ότι οι τιμές διατίμησης κυμαίνονται μεταξύ 1 και 10; Μπορούμε να γράψουμε 10 αιτήματα τα οποία μοιάζουν με το ακόλουθο (!):

```
For i = 1, 2, ..., 10:
SELECT MIN (S.age)
FROM Sailors S
WHERE S.rating = i
```

Αιτήματα με GROUP BY και HAVING

```
SELECT [DISTINCT] λίστα-επιλογής
FROM λίστα-πινάκων
WHERE καταλληλότητα
GROUP BY λίστα-ομαδοποίησης
HAVING καταλληλότητα-ομάδας
```

- Η *λίστα-επιλογής* περιέχει (i) ονόματα πεδίων (ii) όρους που συνιστούν αποτελέσματα διαδικασιών συνάθροισης (π.χ., $\text{MIN}(S.age)$).
 - Η *λίστα-επιλογής* πρέπει να είναι υποσύνολο της *λίστας-ομαδοποίησης*. Διαισθητικά, κάθε πλειάδα της απάντησης αντιστοιχεί σε μια *ομάδα*, και τα πεδία αυτά πρέπει να έχουν μοναδική τιμή σε κάθε ομάδα. (Μια *ομάδα* είναι ένα σύνολο πλειάδων οι οποίες έχουν την ίδια τιμή για όλα τα πεδία που περιγράφονται στη *λίστα-ομαδοποίησης*.)

Ιδεατός Υπολογισμός

- Υπολογίζεται το καρτεσιανό γινόμενο της *λίστας πινάκων*, οι πλειάδες που δεν ικανοποιούν την *καταλληλότητα* απαλείφονται, τα *μη αναγκαία* πεδία διαγράφονται, και οι πλειάδες που απομένουν χωρίζονται σε ομάδες με βάση τη τιμή των πεδίων στη *λίστα-ομαδοποίησης*.
- Στη συνέχεια εφαρμόζεται η *καταλληλότητα-ομάδας* για να περιορίσει κάποιες ομάδες. Οι παραστάσεις στην *καταλληλότητα-ομάδας* πρέπει να έχουν μοναδική τιμή ανά ομάδα!
 - Επομένως, ένα πεδίο που αναφέρεται στην *καταλληλότητα-ομάδας* οφείλει επίσης να εμφανίζεται στη *λίστα-ομαδοποίησης* ή να αποτελεί ονοματισμό του αποτελέσματος ενός τελεστή συνάθροισης.
- Για κάθε ομάδα παράγεται μια πλειάδα στην απάντηση.

Να βρεθεί η ηλικία του νεότερου ναυτικού για κάθε τιμή διατίμησης (rating).

- Με τη χρήση της συνιστώσας GROUP BY διατυπώνεται ως εξής:

```
SELECT S.rating, MIN (S.age)
FROM Sailors S
GROUP BY S.rating
```

Sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	35.0
8	55.5
10	16.0

Ο πίνακας της απάντησης



Να βρεθεί η ηλικία του νεότερου ενήλικα ναυτικού (ηλικία >= 18), για κάθε τιμή διατίμησης με τουλάχιστον 2 ενήλικες ναυτικούς

```
SELECT S.rating, MIN (S.age)
FROM Sailors S
WHERE S.age >= 18
GROUP BY S.rating
HAVING COUNT (*) > 1
```

- Μόνο τα S.rating και S.age αναφέρονται στις συνιστώσες SELECT, GROUP BY και HAVING. Τα άλλα πεδία είναι 'μη αναγκαία'.
- Η 2η στήλη του αποτελέσματος είναι ανώνυμη (Μπορεί να πάρει όνομα με τη χρήση του AS).

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	45.0
7	35.0
8	55.5
10	35.0

rating	age
7	35.0

Ο πίνακας της απάντησης



Για κάθε κόκκινη βάρκα, να βρεθεί ο συνολικός αριθμός κρατήσεων που της έχουν γίνει

```
SELECT B.bid, COUNT (*) AS scount
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid AND B.color='red'
GROUP BY B.bid
```

- Εδώ έχουμε ομαδοποίηση πάνω σε σύζευξη τριών σχέσεων.
- Τι παίρνουμε αν απαλείψουμε το B.color='red' από τη συνιστώσα WHERE και προσθέσουμε μια συνιστώσα HAVING με την ίδια συνθήκη;
- Τι συμβαίνει αν απαλείψουμε τον πίνακα Sailors και τη συνθήκη που περιλαμβάνει το S.sid?



Να βρεθεί η ηλικία του νεότερου ναυτικού με ηλικία > 18, για κάθε τιμή διατίμησης με τουλάχιστον 2 ναυτικούς (οποιασδήποτε ηλικίας)

```
SELECT S.rating, MIN (S.age)
FROM Sailors S
WHERE S.age > 18
GROUP BY S.rating
HAVING 1 < (SELECT COUNT (*)
FROM Sailors S2
WHERE S.rating=S2.rating)
```

- Η συνιστώσα HAVING μπορεί επίσης να περιέχει ένα υποαίτημα.
- Να συγκρίνετε το παραπάνω με το αίτημα στο οποίο μας ενδιέφεραν μόνο ratings με δύο ενήλικες ναυτικούς (ηλικία μεγαλύτερη του 18)!
- Τι γίνεται αν η συνιστώσα HAVING αντικατασταθεί με:
 - HAVING COUNT(*) > 1




Περίληψη

- Η SQL ήταν ένας σημαντικός παράγοντας που οδήγησε στη γρήγορη υιοθέτηση του σχεσιακού μοντέλου, αφού είναι πιο φυσική από τις προγενέστερες της διαδικαστικές γλώσσες αιτημάτων.
- Με την SQL μπορούμε να εκφράσουμε όλα τα αιτήματα που μπορούν να εκφραστούν σε σχεσιακή άλγεβρα και ακόμη περισσότερα.
- Ακόμη και αιτήματα που μπορούν να εκφραστούν στη σχεσιακή άλγεβρα μπορούν συχνά να εκφραστούν με πιο φυσικό τρόπο στην SQL.
- Υπάρχουν πολλοί εναλλακτικοί τρόποι για να γραφτεί μια ερώτηση. Ο βελτιστοποιητής θα αναζητήσει στη συνέχεια ένα πιο αποδοτικό σχέδιο υπολογισμού του αιτήματος.
 - Στη πράξη, οι χρήστες πρέπει να γνωρίζουν πως βελτιστοποιούνται και υπολογίζονται τα αιτήματα έτσι ώστε να επιτυγχάνουν καλύτερα αποτελέσματα.


ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ: ΑΣΚΗΣΗ I

Μανόλης Γεργατσούλης
Καθηγητής
Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας & Μουσειολογίας
Ιόνιο Πανεπιστήμιο
manolis@ionio.gr

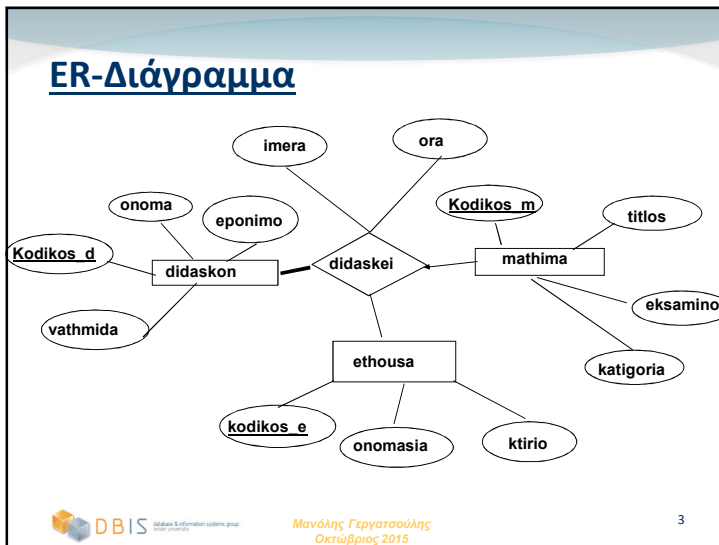


Εκφώνηση Άσκησης

- Ζητείται να σχεδιαστεί βάση δεδομένων για τη διαχείριση των αναθέσεων μαθημάτων του τμήματος Αρχειονομίας, Βιβλιοθηκονομίας και Μουσειολογίας και των αιθουσών για τη διεξαγωγή των μαθημάτων αυτών. Για τη σχεδίαση της βάσης θεωρείστε ότι ισχύουν οι ακόλουθες παραδοχές:
 - Η διδασκαλία κάθε μαθήματος ανατίθεται σε ένα μόνο καθηγητή.
 - Κάθε καθηγητής διδάσκει τουλάχιστον ένα μάθημα.



Μανόλης Γεργατσούλης
Οκτώβριος 2015



Ορισμός πινάκων σε SQL (1/2)

```

CREATE TABLE didaskon
(kodikos_d CHAR(5),
 onoma CHAR(15),
 eponimo CHAR(20),
 vathmida CHAR(2),
 PRIMARY KEY (kodikos_d));

CREATE TABLE ethousa
(kodikos_e CHAR(5),
 onomasia CHAR(15),
 ktirio CHAR(2),
 PRIMARY KEY (kodikos_e));

CREATE TABLE mathima
(kodikos_m CHAR(5),
 titlos CHAR(15),
 eksamino CHAR(2),
 katigoria CHAR(2),
 PRIMARY KEY (kodikos_m));
    
```



Μανόλης Γεργατσούλης
Οκτώβριος 2015

Ορισμός πινάκων σε SQL (2/2)

```
CREATE TABLE didaskei
(
  kodikos_d CHAR(5),
  kodikos_m CHAR(5),
  kodikos_e CHAR(5),
  imera CHAR(10),
  ora CHAR(10),
  PRIMARY KEY (kodikos_m),
  FOREIGN KEY (kodikos_m) REFERENCES mathima,
  FOREIGN KEY (kodikos_d) REFERENCES didaskon,
  FOREIGN KEY (kodikos_e) REFERENCES ethousa
);
```

ΑΙΤΗΜΑ 1:

Να βρεθούν τα ονόματα και τα επώνυμα όλων των διδασκόντων.

SELECT onoma, eponimo FROM didaskon	SELECT D.onoma, D.eponimo FROM didaskon D
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 2:

Να βρεθούν οι τίτλοι όλων των μαθημάτων του 'Δ' εξαμήνου.

SELECT titlos FROM mathima WHERE eksamino = 'Δ'	SELECT M.titlos FROM mathima M WHERE M.eksamino = 'Δ'
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 3:

Ζητείται πίνακας ο οποίος περιλαμβάνει τα ονόματα και τα επώνυμα των διδασκόντων και τα μαθήματα που διδάσκουν.

```
SELECT D1.onoma, D1.eponimo, M.titlos  
FROM didaskon D1, didaskei D2, mathima M  
WHERE D1.kodikos_d = D2.kodikos_d AND  
D2.kodikos_m = M.kodikos_m
```

ΑΙΤΗΜΑ 4:

Ζητείται πίνακας ο οποίος περιλαμβάνει τους τίτλους των μαθημάτων που διδάσκονται στην αίθουσα με κωδικό 'Α1'.

```
SELECT M.titlos
FROM didaskei D, mathima M
WHERE D.kodikos_e = 'A1' AND
      D.kodikos_m = M.kodikos_m
```

ΑΙΤΗΜΑ 5:

Ζητείται πίνακας με τους τίτλους των μαθημάτων που διδάσκει ο διδάσκων με επώνυμο 'ΓΕΡΓΑΤΣΟΥΛΗΣ' συνοδευόμενος από το όνομα των αιθουσών στις οποίες διδάσκονται τα μαθήματα αυτά.

```
SELECT M.titlos, E.onomasia
FROM didaskei D1, mathima M, ethousa E, didaskon D2
WHERE D2.eponimo = 'ΓΕΡΓΑΤΣΟΥΛΗΣ' AND
      D2.kodikos_d = D1.kodikos_d AND
      D1.kodikos_m = M.kodikos_m AND
      D1.kodikos_e = E.kodikos_e
```

ΑΙΤΗΜΑ 6:

Να βρεθεί το πλήθος των διδασκόντων που ανήκουν στην βαθμίδα 'B'.

```
SELECT COUNT(*)
FROM didaskon D
WHERE D.vathmida = 'B'
```

ΑΙΤΗΜΑ 7:

Να βρεθεί το πλήθος των μαθημάτων του 'Γ' εξαμήνου τα οποία διδάσκει ο διδάσκων με επώνυμο 'ΚΑΠΙΔΑΚΗΣ'.

```
SELECT COUNT(*)
FROM didaskon D1, didaskei D2, mathima M
WHERE D1.eponimo = 'ΚΑΠΙΔΑΚΗΣ' AND
      D1.kodikos_d = D2.kodikos_d AND
      D2.kodikos_m = M.kodikos_m AND
      M.eksamino = 'Γ'
```

ΑΙΤΗΜΑ 8:

Να βρεθεί το πλήθος των μαθημάτων του 'Γ' εξαμήνου που διδάσκονται την 'ΤΕΤΑΡΤΗ'.

```
SELECT COUNT(*)
FROM didaskei D, mathima M
WHERE D.imera = 'ΤΕΤΑΡΤΗ' AND
      D.kodikos_m = M.kodikos_m AND
      M.eksamino = 'Γ'
```

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ: ΑΣΚΗΣΗ 2

Μανόλης Γεργατσούλης

Καθηγητής

Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας & Μουσειολογίας

Ιόνιο Πανεπιστήμιο

manolis@ionio.gr



ΕΚΦΩΝΗΣΗ ΑΣΚΗΣΗΣ

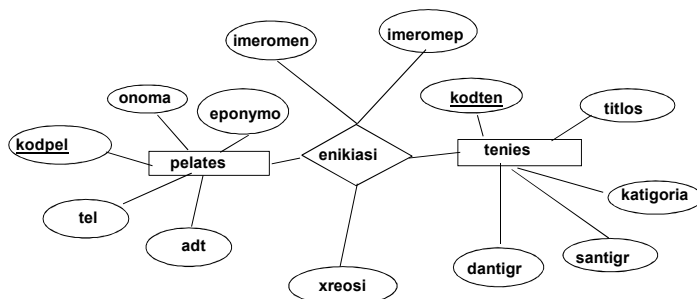
- Ζητείται να σχεδιαστεί απλή βάση δεδομένων για τη διαχείριση της ενοικίασης ταινιών ενός Video Club. Να κατασκευαστεί ER-διάγραμμα, να οριστούν οι πίνακες σε SQL, και να οριστούν μερικά αντιπροσωπευτικά αιτήματα.



Μανόλης Γεργατσούλης
Οκτώβριος 2015

2

ER-ΔΙΑΓΡΑΜΜΑ



Μανόλης Γεργατσούλης
Οκτώβριος 2015

3

Οι πίνακες της ΒΔ

Πίνακας: **pelates**

Όνομα πεδίου	Είδος δεδομένων	Εξήγηση
kodpel	χαρακτήρες	ο κωδικός πελάτη
onoma	χαρακτήρες	το όνομα του πελάτη
eponymo	χαρακτήρες	το επώνυμο του πελάτη
tel	χαρακτήρες	το τηλέφωνο του πελάτη
adt	χαρακτήρες	αριθμός ταυτότητας του πελάτη

✓ Το πεδίο **kodpel** είναι κλειδί του πίνακα **pelates**



Μανόλης Γεργατσούλης
Οκτώβριος 2015

4

Οι πίνακες της ΒΔ (συνέχεια)

Πίνακας: **tenies**

Όνομα πεδίου	Είδος δεδομένων	Εξήγηση
kodten	χαρακτήρες	ο κωδικός της ταινίας
titlos	χαρακτήρες	ο τίτλος της ταινίας
katigoria	χαρακτήρες	η κατηγορία που ανήκει η ταινία
santigr	ακέραιος αριθμός	ο συνολικός αριθμός αντιγράφων
dantigr	ακέραιος αριθμός	ο αριθμός των διαθέσιμων αντιγράφων

✓ Το πεδίο **kodten** είναι κλειδί του πίνακα **tenies**

Οι πίνακες της ΒΔ (συνέχεια)

Πίνακας: **enikiasi**

Όνομα πεδίου	Είδος δεδομένων	Εξήγηση
kodpel	χαρακτήρες	ο κωδικός πελάτη
kodten	χαρακτήρες	ο κωδικός της ταινίας
imeromen	ημερομηνία	η ημερομηνία ενοικίασης της ταινίας
imeromep	ημερομηνία	η ημερομηνία που πρέπει να επιστραφεί η ταινία
xreosi	πραγματικός αριθμός	το ποσό της χρέωσης

- ✓ Τα πεδία (**kodpel,kodten**) είναι κλειδί του πίνακα **enikiasi**
- ✓ Το πεδίο **kodpel** του πίνακα **enikiasi** είναι ξένο κλειδί (αναφέρεται στον πίνακα **pelates**)
- ✓ Το πεδίο **kodten** του πίνακα **enikiasi** είναι ξένο κλειδί (αναφέρεται στον πίνακα **tenies**)

ΟΡΙΣΜΟΣ ΠΙΝΑΚΩΝ ΣΕ SQL

```
CREATE TABLE pelates
(kodpel CHAR(5),
onoma CHAR(15),
eponymo CHAR(20),
tel CHAR(15),
adt CHAR(8),
PRIMARY KEY (kodpel));
```

```
CREATE TABLE tenies
(kodten CHAR(5),
titlos CHAR(15),
katigoria CHAR(20),
santigr INTEGER,
dantigr INTEGER,
PRIMARY KEY (kodten));
```

```
CREATE TABLE enikiasi
(kodpel CHAR(5),
kodten CHAR(5),
imeromen DATE,
imeromep DATE,
xreosi REAL,
PRIMARY KEY (kodpel,kodten),
FOREIGN KEY (kodpel) REFERENCES pelates,
FOREIGN KEY (kodten) REFERENCES tenies);
```

Στιγμιότυπο της ΒΔ (1/3)

pelates				
kodpel	onoma	eponymo	tel	adt
1232	Γιάννης	Πέτρου	3645234	M345678
2112	Γιόργος	Ιωάννου	6554432	K332255
2243	Μαρία	Σταμάτη	2934560	H223456
2345	Νίκος	Γεωργίου	2345678	Π371244
2438	Πέτρος	Νικολάου	3456789	K423420
3243	Άρης	Πέτρου	3243546	A934202
3344	Άρης	Ιωάννου	7665543	A113243
3443	Μηνάς	Στυλιανού	5670470	Ξ908765
4423	Σοφία	Μιχαήλ	4560789	P234561
5678	Άννα	Μηνά	6504526	N987654

Στιγμιότυπο της ΒΔ (2/3)

tenies				
kodten	titlos	katigoria	santigr	dantigr
123	Μανχάταν	Κομωδία	5	3
132	Το διαμάντι του Νείλου	Περπέτεια	4	4
210	Ο βασιλιάς των λιονταριών	Παιδικό	7	7
233	Ρομπέν τον δασόν	Παιδικό	6	5
238	Το μορό της Ροζμαρί	Θρίλερ	3	3
244	Μονομαχία στο Ελ Πάσο	Γουέστερν	3	2
297	Η ωραία κοιμωμένη	Παιδικό	4	3
324	Ο Διαβολάκος	Κομωδία	4	3
333	Μπαμπούλες ΑΕ	Παιδικό	5	5
423	Μουζάν	Παιδικό	6	6
432	Και οι 7 ήταν υπέροχοι	Γουέστερν	1	0
455	Τούτσι	Κομωδία	4	4
456	Ποντικομικρούλης	Παιδικό	5	5
478	Η Χιονάτη και οι επτά νάνοι	Παιδικό	4	4
511	Ταξίδι στα Κύθηρα	Κοινωνικό	3	3
512	Ο καλός ο κακός και ο άσημος	Γουέστερν	2	0
633	Η εκλογή της Σόφης	Κοινωνικό	5	5
733	Χορεύοντας με τους λύκους	Περπέτεια	3	0

Στιγμιότυπο της ΒΔ (3/3)

enikiasi				
kodpel	kodten	imeromen	imeromep	xreosi
1232	123	4/12/2002	11/12/2002	2
1232	233	4/12/2002	11/12/2002	1,5
2112	123	3/12/2002	11/12/2002	2
2112	324	3/12/2002	7/12/2002	2
2112	733	3/12/2002	7/12/2002	2
2243	432	2/12/2002	9/12/2002	1,5
2243	512	3/12/2002	10/12/2002	2
2345	297	2/12/2002	8/12/2002	1,5
2345	512	4/12/2002	11/12/2002	2
3243	244	1/12/2002	8/12/2002	2
3443	733	2/12/2002	7/12/2002	1,5
4423	733	1/12/2002	9/12/2002	2,5

ΑΙΤΗΜΑ 1:

Να βρεθούν τα στοιχεία του πελάτη με κωδικό 2438

SELECT * FROM pelates WHERE kodpel = '2438';	SELECT * FROM pelates P WHERE P.kodpel = '2438';
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 2:

Να εμφανιστεί λίστα των ταινιών που διαθέτει το Video Club η οποία να περιλαμβάνει τα στοιχεία: Τίτλος ταινίας, και Κατηγορία.

SELECT titlos, katigoria FROM tenies;	SELECT S.titlos, S.katigoria FROM tenies S;
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 3:

Να εμφανιστούν οι τίτλοι των ταινιών που διακινεί το Video Club με κατηγορία περιεχομένου «Παιδικό»

<pre>SELECT titlos FROM tenies WHERE katigoria = 'Παιδικό';</pre>	<pre>SELECT S.titlos FROM tenies S WHERE S.katigoria = 'Παιδικό';</pre>
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 4:

Να εμφανιστούν τα (πλήρη) στοιχεία των ταινιών για τις οποίες δεν υπάρχουν διαθέσιμα αντίγραφα.

<pre>SELECT * FROM tenies WHERE dantigr = 0;</pre>	<pre>SELECT * FROM tenies T WHERE T.dantigr = 0;</pre>
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 5:

Να εμφανιστούν τα (πλήρη) στοιχεία των ταινιών για τις οποίες δεν βρίσκεται νοικιασμένο κανένα αντίγραφο τους αυτήν τη στιγμή.

<pre>SELECT * FROM tenies WHERE dantigr = santigr;</pre>	<pre>SELECT * FROM tenies T WHERE T.dantigr = T.santigr;</pre>
1 ^{ος} τρόπος	2 ^{ος} τρόπος

ΑΙΤΗΜΑ 6:

Να εμφανιστούν οι τίτλοι των ταινιών και το πλήθος των αντιγράφων για τις ταινίες που τα διαθέσιμα αντίγραφα είναι περισσότερα από 2.

```
SELECT T.titlos, T.dantigr
FROM tenies AS T
WHERE T.dantigr > 2;
```

ΑΙΤΗΜΑ 7:

Να εμφανιστούν οι τίτλοι των ταινιών και το πλήθος των διαθέσιμων αντιγράφων για τις ταινίες των οποίων τα αντίγραφα που βρίσκονται στα χέρια πελατών είναι τουλάχιστον 2.

```
SELECT T.titlos, T.dantigr  
FROM tenies AS T  
WHERE T.santigr-T.dantigr >= 2;
```

ΑΙΤΗΜΑ 8:

Να εμφανιστούν οι τίτλοι των ταινιών που βρίσκονται χρωμένες στον πελάτη με κωδικό 2112.

```
SELECT T.titlos  
FROM tenies AS T, enikiasi AS E  
WHERE E.kodpel = '2112' AND T.kodten = E.kodten;
```

ΑΙΤΗΜΑ 9:

Να εμφανιστούν οι τίτλοι των ταινιών που έχει χρωθεί ο πελάτης Νίκος Γεωργίου.

```
SELECT T.titlos  
FROM tenies AS T, enikiasi AS E, pelates AS P  
WHERE P.onoma = 'Νίκος' AND P.eponymo = 'Γεωργίου'  
AND E.kodpel = P.kodpel AND T.kodten = E.kodten;
```

ΑΙΤΗΜΑ 10:

Να εμφανιστεί λίστα των πελατών με τις ταινίες που έχει χρωθεί ο καθένας. Η λίστα να περιλαμβάνει τα στοιχεία Όνομα πελάτη, Επώνυμο πελάτη, Τίτλος ταινίας.

```
SELECT P.onoma, P.eponymo, T.titlos  
FROM tenies AS T, enikiasi AS E, pelates AS P  
WHERE E.kodpel = P.kodpel and T.kodten = E.kodten;
```

AΙΤΗΜΑ 11:

Να επαναδιατυπωθεί το προηγούμενο ερώτημα κατά τρόπο ώστε στη λίστα να εμφανίζονται μόνο οι ταινίες με περιεχόμενο της κατηγορίας «Παιδικό».

```
SELECT P.onoma, P.eponymo, T.titlos
FROM tenies AS T, enikiasi AS E, pelates AS P
WHERE E.kodpel=P.kodpel AND T.kodten = E.kodten
AND T.kategoria = 'Παιδικό';
```

AΙΤΗΜΑ 12:

Να εμφανιστούν τα ονόματα και τα επώνυμα των πελατών που έχουν νοικιάσει τουλάχιστον μια ταινία. Κάθε πελάτης να εμφανίζεται μια μόνο φορά ακόμη και αν έχει νοικιάσει περισσότερες από μια ταινίες.

```
SELECT DISTINCT P.onoma, P.eponymo
FROM pelates AS P, enikiasi AS E
WHERE P.kodpel=E.kodpel;
```

AΙΤΗΜΑ 13:

Να εμφανιστούν το όνομα και το επώνυμο των πελατών που το επώνυμο τους αρχίζει από «Μ».

```
SELECT S.onoma, S.eponymo
FROM pelates AS S
WHERE S.eponymo LIKE 'M*';
```

AΙΤΗΜΑ 14:

Να υπολογιστεί το πλήθος των πελατών του Video Club.

```
SELECT count(*)
FROM pelates AS S;
```

AITHMA 15:

Να υπολογιστεί το σύνολο των αντιτύπων ταινιών που διαθέτει το Video Club.

```
SELECT sum (S.santigr)  
FROM tenies S;
```

AITHMA 16:

Να βρεθεί η μέγιστη χρέωση ανάμεσα στις καταχωρήσεις του πίνακα **enikiasi**.

```
SELECT max(E.xreosi)  
FROM enikiasi E;
```

AITHMA 17:

Να βρεθεί η μέση χρέωση ανάμεσα στις καταχωρήσεις του πίνακα **enikiasi**.

```
SELECT avg(E.xreosi)  
FROM enikiasi E;
```

AITHMA 18:

Να βρεθούν τα ονόματα και επώνυμα των πελατών που έχουν νοικιάσει την ταινία με κωδικό 123 ή την ταινία με κωδικό 512.

```
SELECT P.onoma, P.eponymo  
FROM pelates AS P, enikiasi AS E  
WHERE P.kodpel=E.kodpel AND  
(E.kodten='123' OR E.kodten='512');
```

ΑΙΤΗΜΑ 19:

Να εμφανιστεί ο τίτλος της ταινίας με τον μεγαλύτερο αριθμό διαθέσιμων αντιγράφων.

```
SELECT T.titlos  
FROM tenies AS T  
WHERE T.dantigr >= (SELECT max(S.dantigr)  
                    FROM tenies S);
```



DBIS

database & information systems group
open to visitors

Μανόλης Γεργασιούλης
Οκτώβριος 2015

29