



## **EPL342 –Databases**

# **Lecture 10: RA III**

## **RA Advanced + Examples**

(Chapter 8.4-8.5, Elmasri-Navathe 7ED)

**Demetris Zeinalipour**

<http://www.cs.ucy.ac.cy/courses/EPL342>

# Επισκόπηση Τελεστών Σχεσιακής Άλγεβρας που Καλύφθηκαν



- Η Σχεσιακή Άλγεβρα παρέχει τους τελεστές (operators):
  - Μοναδιαίοι Σχεσιακοί Τελεστές (Unary Relational Ops)
    - Επιλογή (Select,  $\sigma$  (sigma))
    - Προβολή (Project,  $\pi$  (pi))
    - Μετονομασία (Rename,  $\rho$  (rho))
  - Σχεσιακοί Τελεστές από την Θεωρία Συνόλων
    - Ένωση (UNION,  $\cup$ ), Τομή (INTERSECTION,  $\cap$ ), Διαφορά Συνόλων (DIFFERENCE ή MINUS,  $-$ )
    - Καρτεσιανό Γινόμενο (CARTESIAN PRODUCT,  $\times$ )
  - Δυαδικοί Σχεσιακοί Τελεστές (Binary Relational Ops)
    - Συνένωση (JOIN,  $\bowtie$ ) (υπάρχουν πολλαπλές εκδοχές)
      - Εφόσον το  $\bowtie$  δεν υποστηρίζεται σαν σύμβολο θα χρησιμοποιείται το  $\otimes$
    - Διάρθρωση (DIVISION,  $/$ )
  - Επιπλέον Σχεσιακοί Τελεστές
    - Συναρτήσεις Συνάθροισης AGGREGATE FUNCTIONS (π.χ., SUM, COUNT, AVG, MIN, MAX)
    - Εξωτερική Συνένωση (OUTER JOINS).

# Περιεχόμενο Διάλεξης



## Κεφάλαιο 8: Προχωρημένη Σχεσιακή Άλγεβρα

- **8.3) Δυαδικοί Σχεσιακοί Τελεστές: Διαίρεση, Συμβολισμός για Δένδρα Επερωτήσεων**
- **8.4) Επιπλέον Σχεσιακές Πράξεις:**
  - **Συναθροιστικές Συναρτήσεις (Aggregate Functions)**
  - **Γενικευμένη Προβολή (Generalized Projection)**
  - **Ομαδοποίηση (Grouping)**
  - **Πράξεις Αναδρομικής Κλειστότητας (Recursive Closure)**
  - **Πράξεις Εξωτερικής Συνένωσης (Outer Join)**
- **8.5) Παραδείγματα Επερωτήσεων σε Σχεσιακή Άλγεβρα με το Σχήμα Βάσης University και το Σχήμα Sailors-Reserve-Boats**

# Διαδικοί Σχεσιακοί Τελεστές

## Διαίρεση (Division)



- Ο τελεστής της **Διαίρεσης (Division, /)**, παρόλο που **ΔΕΝ** υποστηρίζεται σε **πραγματικές γλώσσες βάσεων δεδομένων\***, είναι χρήσιμος στο πλαίσιο της Σχεσιακής Άλγεβρας για να διατυπώσει επερωτήσεις της μορφής:

*Βρες τους Suppliers που προσφέρουν ΌΛΑ τα Parts (όπως προσδιορίζονται τα Parts).*

Π.χ.,

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

PARTS

pno
p2
p4

/

=

sno
s1
s4

\* Στην **συνέχεια** θα δούμε πως ο τελεστής **μπορεί να υλοποιηθεί** με χρήση **10-4** υπαρκτών τελεστών της **SQL** (δηλ., αντίστοιχους τελεστές των **σ-π-χ**)

# Διαδικοί Σχεσιακοί Τελεστές

## Διαίρεση (Division)



Διαιρετέος (nominator)

Διαιρέτες (denominators)

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

**A**

/

pno
p2

**B1**

Απάντηση;

sno
s1
s2
s3
s4

**A/B1**

/

pno
p2
p4

**B2**

Απάντηση;

sno
s1
s4

**A/B2**

/

pno
p1
p2
p4

**B3**

Απάντηση;

sno
s1

**A/B3**

Αναπαράσταση Διαίρεσης με Ποσοδείκτες

$$A / B = \{ x \mid \exists(x,y) \in A \ \forall y \in B \}$$

# Expressing A/B Using Basic Operators



- Σημειώστε ότι η διαίρεση (/) ΔΕΝ είναι **βασικός τελεστής** αλλά αποτελεί **βολική συντομογραφία**
  - Το ίδιο ισχύει και για την **Συνένωση** ( $\otimes$ ), θυμηθείτε ότι  **$R \otimes_{\theta} S = \sigma_{\theta}(R \times S)$** .

- Ωστόσο η συνένωση είναι πραγματικά χρήσιμη (για αυτό υλοποιείται από από την SQL ως ξεχωριστός τελεστής)

- *Εφόσον η διαίρεση δεν υπάρχει ως τελεστής σε πραγματικές γλώσσες, ποια **ακολουθία** τελεστών **σ-π-χ** θα μας επέστρεφε το **ίδιο λογικό αποτέλεσμα**;*

- **Έκφραση Διαίρεσης με τελεστές σ-π-χ**

$$A / B = \pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$$

- Η επόμενη διαφάνεια εξηγεί την πιο πάνω έκφραση

# Διαδικασιακοί Τελεστές

## Διαίρεση (Division)



$$A / B = \pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$$

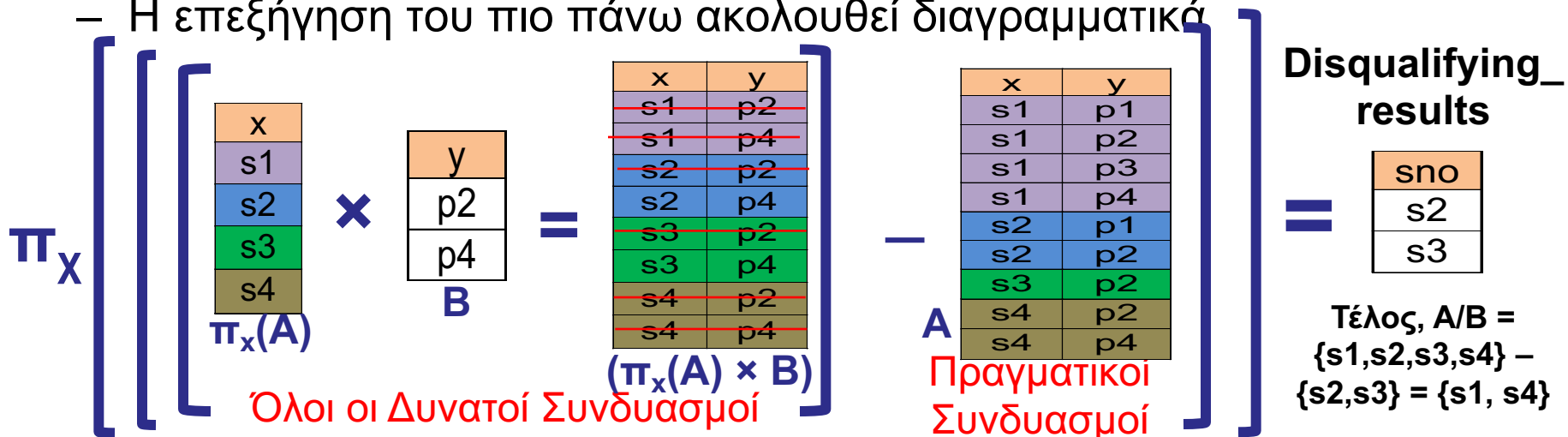
$$= \text{all\_results} - \text{disqualifying\_results}$$

- **all\_results**: Όλες οι δυνατές πλειάδες του αποτελέσματος
  - π.χ., όλοι οι suppliers  $\pi_x(A) = \{s_1, s_2, s_3, s_4\}$
- Από το **all\_results** πρέπει να αφαιρέσουμε τους ακατάλληλους suppliers (**disqualifying\_results**)
  - αυτούς που ΔΕΝ προσφέρουν **ΟΛΑ** τα parts  $y \in B$ . (δηλ.,  $s_2, s_3$ )
  - **disqualifying\_results** συμβολίζεται με  $[\pi_x((\pi_x(A) \times B) - A)]$
  - Η επεξήγηση του πιο πάνω ακολουθεί διαγραμματικά

x	y
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

**B**

y
p2
p4



# Δένδρο Επερώτησης (Query Tree)



## • Δένδρο Επερώτησης (Query Tree)

Είναι μια δενδρική δομή που αντιστοιχεί σε μια έκφραση της Σχεσιακής Άλγεβρας η οποία δείχνει με ποια ακολουθία πρέπει να εκτελεστούν οι τελεστές

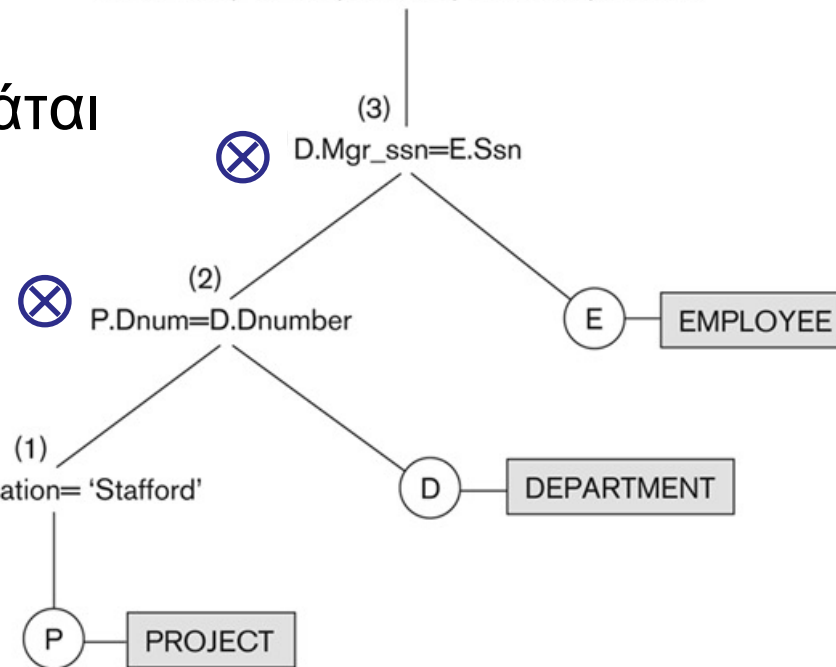
- **Τερματικοί Κόμβοι:** Σχέσεις Εισόδου  $\pi_{P.Pnumber, P.Dnum, E.Lname, E.Address, E.Bdate}$
- **Εσωτερικοί Κόμβοι:** Τελεστές
- Το δένδρο στα δεξιά, αναπαριστάται από την έκφραση Σ.Α.:

$T1 \leftarrow \sigma_{P.Plocation='Stafford'}(P)$

$T2 \leftarrow T1 \otimes_{DNum=DNumber} D$

$T3 \leftarrow T2 \otimes_{Mgr\_ssn=ssn} D$

$T4 \leftarrow \pi_{Pnumber, Dnum, Address, Bdate}(T3)$





# Δένδρο Επερώτησης (Query Tree)



- Ένα **Δένδρο Επερώτησης (Query Tree)** ονομάζεται και **Δένδρο Αποτίμησης Επερωτήσεων (Query Evaluation Tree)**
- Η βάση δεδομένων επιλεγεί συνήθως μόνο τα **αριστεροβαθή δένδρα** (αυτά στα οποία το δεξί παιδί είναι σχέση εισόδου
  - γιατί με αυτό τον τρόπο τα αποτελέσματα μπορεί να σωληνώνονται μεταξύ τελεστών (pipelined)
- Σημειώστε ότι υπάρχουν πολλά τέτοια πιθανά δένδρα  $n!$ , όπου  $n$  ο αριθμός των σχέσεων (διάταξη χωρίς επανατοποθέτηση)

# Επιπλέον Σχεσιακοί Τελεστές (Additional Relational Operators)



- Η σχεσιακή άλγεβρα, όπως προτάθηκε **δεν είχε πρόνοιες** για κάποια είδη **χρήσιμων πράξεων**, π.χ.,
  - **Συναθροιστικές Συναρτήσεις (Aggregate Functions)**
    - Εύρεση **απλών στατιστικών** των πινάκων (π.χ., μέση ηλικία, μέγιστος βαθμός φοιτητή, αριθμός φοιτητών σε μια σχέση κτλ.)
  - **Ομαδοποίηση (Grouping)** των συναθροιστικών αποτελεσμάτων (π.χ., ανά Τμήμα)
  - **Πράξεις Αναδρ. Κλειστότητας (Recursive Closure)** για εύρεση απαντήσεων αναδρομικά σε μια σχέση
  - **Πράξεις Εξωτερικής Συνένωσης (Outer Join)** για συμπερίληψη πλειάδων που δεν συνενώνονται
- Πολλές από τις πιο πάνω πράξεις υποστηρίζονται σήμερα και τόσο στο **τυπικό επίπεδο** (δηλ., στη Σχεσιακή Άλγεβρα όσο και στο **πρακτικό επίπεδο** (δηλ., στη SQL, QBE, κτλ.)

# Επιπλέον Σχεσιακοί Τελεστές

## (Συναθροιστικές Συναρτήσεις)



- **Συναθροιστικές Συναρτήσεις (Aggregate Functions):** Προσδιορίζουν μαθηματικές πράξεις πάνω σε συλλογές τιμών της βάσης:

$\mathcal{F}_{\langle \text{function-list} \rangle}(\mathbf{Relation})$

– **Function List:** **MIN, MAX, SUM, COUNT, AVERAGE, ...**

- Παράδειγμα: Βρες τη μέση ηλικία των υπάλληλων

$\mathcal{F}_{Average(age)}(\mathbf{EMPLOYEE})$

- Η συνάρτηση COUNT χρησιμοποιείται για να μετριοούνται πλειάδες ή τιμές.

– Δεν μετριοούνται τα **NULL**.

– Δεν μετριοούνται τα Διπλότυπα (εφόσον δεν υπάρχουν στην Σχεσιακή Άλγεβρα)

- Στην SQL, όπου οι σχέσεις (πίνακες) ενδέχεται να περιέχουν διπλότυπα (duplicates) χρησιμοποιείται και η έννοια του **COUNT DISTINCT** για να **ΜΗΝ** μετριοούνται ξανά τα **διπλότυπα στα πολυσύνολα**.

# Επιπλέον Σχεσιακοί Τελεστές

## (Συναθροιστικές Συναρτήσεις)



- Παραδείγματα Χρήσης του Τελεστή Συναθροιστικής Συναρτήσεως  $\mathcal{F}$ 
  - $\mathcal{F}_{\text{MAX}(\text{Salary})}$  (**EMPLOYEE**) επιστρέφει την μέγιστη τιμή του πεδίου **Salary** που εμφανίζεται στη σχέση **EMPLOYEE**.
  - $\mathcal{F}_{\text{MIN}(\text{Salary})}$  (**EMPLOYEE**) επιστρέφει την ελάχιστη τιμή του πεδίου **Salary** που εμφανίζεται στη σχέση **EMPLOYEE**.
  - $\mathcal{F}_{\text{SUM}(\text{Salary})}$  (**EMPLOYEE**) επιστρέφει το άθροισμα του πεδίου **Salary** που εμφανίζεται στη σχέση **EMPLOYEE**.
  - $\mathcal{F}_{\text{COUNT}(\text{SSN}), \text{AVERAGE}(\text{Salary})}$  (**EMPLOYEE**) επιστρέφει τον αριθμό των υπαλλήλων και τον μέσο όρο των μισθών τους

# Επιπλέον Σχεσιακοί Τελεστές

(Ομαδοποίηση με Συναθροιστικές Συναρτήσεις)

## Ομαδοποίηση (Grouping)

- Οι συναθροιστικές συναρτήσεις  $\mathcal{F}$  μπορεί να **εφαρμόζονται** σε επί μέρους (ομαδοποιημένα) **υποσύνολα** μιας **σχέσης**
  - Π.χ., Βρες τον Μέσο Μισθό **ανά Τμήμα** (αντί για όλη την εταιρεία).
- Ο τελεστής της συνάθροισης **επεκτείνεται** ως ακολούθως:  
 $\langle \text{grouping-list} \rangle \mathcal{F} \langle \text{function-list} \rangle$  (**Relation**)
- Το **αποτέλεσμα** περιλαμβάνει τα πεδία που προσδιορίζονται στο **function-list** και το **επιπλέον πεδίο(α)** του **grouping-list**

## Παράδειγμα

- **Ερώτηση:** Για κάθε τμήμα, ανάκτησε το DNO, τον αριθμό των υπαλλήλων και τον μέσο μισθό ανά τμήμα
- **Απάντηση:**

DNO  $\mathcal{F}$  COUNT(SSN), AVERAGE(Salary) (EMPLOYEE)

# Επιπλέον Σχεσιακοί Τελεστές

(Ομαδοποίηση με Συναθροιστικές Συναρτήσεις)



- Εάν θέλουμε μπορούμε να **μετονομάσουμε** τα αποτελέσματα έτσι ώστε αυτά να έχουν εξειδικευμένα ονόματα γνωρισμάτων

- **R(Dno, No\_of\_employees, Average\_sal)**

←  $\text{DNO } \mathcal{F} \text{ COUNT(SSN), AVERAGE(Salary) (EMPLOYEE)}$

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

R

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000



# Επιπλέον Σχεσιακοί Τελεστές

(Ομαδοποίηση + Επιλογή με Συναθρ. Συναρ.)



- **Ομαδοποίηση με Επιλογή (HAVING)**

Φιλτράρει το αποτέλεσμα μιας ομαδοποίησης

- π.χ., να βρούμε και πάλι τον αριθμό υπαλλήλων και τον μέσο όρο μισθών ανά τμήμα αλλά να τυπώσουμε μόνο τα αποτελέσματα όπου **Count>3**.
- Στη **SQL** υπάρχει εξειδικευμένος τελεστής για αυτή τη συχνή λειτουργία ο οποίος ονομάζεται **HAVING** (θα μελετηθεί αργότερα)
- Στη **Σχεσιακή Άλγεβρα** μπορεί να λυθεί με συνδυασμό **Συναθροιστικής συνάρτησης** ακολουθούμενο από **επιλογή**.
- Το παράδειγμα στην επόμενη διαφάνεια δείχνει πως...

Fname	Minit	Lname	<u>Ssn</u>	...	Salary	Super_ssn	Dno
John	B	Smith	123456789		30000	333445555	5
Franklin	T	Wong	333445555		40000	888665555	5
Ramesh	K	Narayan	666884444		38000	333445555	5
Joyce	A	English	453453453	...	25000	333445555	5
Alicia	J	Zelaya	999887777		25000	987654321	4
Jennifer	S	Wallace	987654321		43000	888665555	4
Ahmad	V	Jabbar	987987987		25000	987654321	4
James	E	Bong	888665555		55000	NULL	1

Dno	Count (*)	Avg (Salary)
5	4	33250
4	3	31000
1	1	55000

Result of Q24

Grouping EMPLOYEE tuples by the value of Dno

# Επιπλέον Σχεσιακοί Τελεστές (Ομαδοποίηση + Επιλογή με Συναθρ. Συναρ.)



EMPLOYEE									
Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPENDENT				
<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship

Selection is always a number, not a set (e.g.,  $\sigma_{Dcount \geq MAX(age)}$  is wrong!)

- **Q1: Βρες τα ονόματα όλων των υπαλλήλων με δυο ή περισσότερους εξαρτώμενους (dependents).**

// Καταμέτρηση (COUNT) Εξαρτωμένων κάθε ατόμου

1.  $T1(Ssn, Dcount) \leftarrow \pi_{ESSN, COUNT(Dependent\_name)}(DEPENDENT)$

// Φιλτράρισμα Συναρθροιστικού Αποτελέσματος με Επιλογή

2.  $T2 \leftarrow \sigma_{Dcount \geq 2}(T1)$

// Προβολή Αποτελεσμάτων (μέσω φυσικής συνένωσης πάνω στο SSN)

3.  $RESULT \leftarrow \pi_{LNAME, FNAME}(T2 * EMPLOYEE)$



# Επιπλέον Σχεσιακοί Τελεστές (Γενικευμένη Προβολή)



- **Γενικευμένη Προβολή (Generalized Projection):** Επεκτείνει την πράξη της προβολής επιτρέποντας να συμπεριληφθούν στη λίστα της προβολής **συναρτήσεις γνωρισμάτων, δηλ.,**

**$\pi_{F_1, F_2, \dots, F_n}(R)$ , όπου  $F_i (i \leq n)$  μπορεί να είναι συνάρτηση γνωρίσματος ή σταθερά.**

- Π.χ., Υποθέστε το ακόλουθο Σχήμα:  
EMPLOYEE(Ταυτότητα, Μισθός, Αποκοπές,  
Χρόνια\_Υπηρεσίας)

Π Ταυτότητα, Μισθός-Αποκοπές, 2000\*Χρόνια\_Υπηρεσίας, 0.25\*Μισθός (EMPLOYEE)

Απλό  
Γνώρισμα

Καθαρός  
Μισθός

Bonus

Φορολογία

# Επιπλέον Σχεσιακοί Τελεστές (Αναδρομική Κλειστότητα)



- **Αναδρομική Κλειστότητα (Recursive or Transitive Closure Operations):** Επεξήγηση με Παράδειγμα:
  - Supervisor(123456789) → 333445555
  - Supervisor(333445555) → 888665555
  - Supervisor(888665555) → NULL
- Supervisor(123456789) σε όλα τα πιο πάνω επίπεδα;
  - Απάντηση A = {333445555, 888665555, NULL}
- Το A ορίζει την αναδρομική κλειστότητα του Supervisor(123456789)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

# Επιπλέον Σχεσιακοί Τελεστές (Αναδρομική Κλειστότητα)

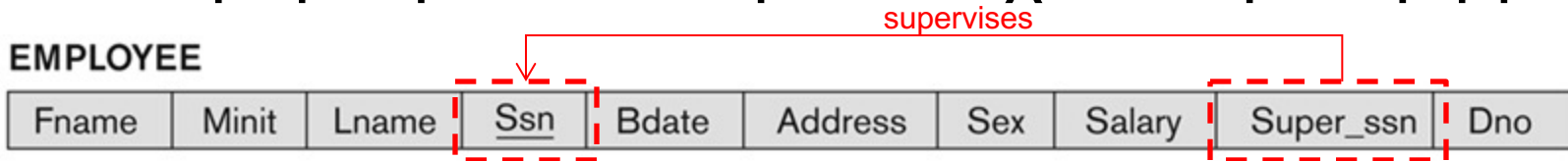


- Αναδρομική Κλειστότητα (Recursive Closure Operations)
  - Στη Σχεσιακή Άλγεβρα **δεν υπάρχει τελεστής** που να υποστηρίζει αυτή την πράξη.
  - Μπορούμε ωστόσο να το **υποστηρίξουμε** με μια **σειρά συνενώσεων** (δες επόμενη διαφάνεια).
  - Σε SQL3, που θα δούμε αργότερα, θα είναι δυνατό να υλοποιηθεί με κάποιο **διαδικαστικό τρόπο**, με χρήση επαναλήψεων, αλλά όχι μέσω κάποιου **εξειδικευμένου τελεστή**.
    - Μια ιδέα είναι η χρήση του αλγορίθμου **Floyd-Warshall**, το οποίο βρίσκει το **ελάχιστο μονοπάτι μεταξύ οποιονδήποτε κορυφών** (και κατ' επέκταση όλα τα δυνατά ζεύγη)

# Επιπλέον Σχεσιακοί Τελεστές (Αναδρομική Κλειστότητα)



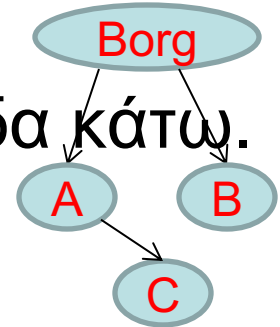
- Αναδρομική Κλειστότητα σε Σχεσιακή Άλγεβρα



- Ερώτημα (Q5):** Βρες τους υφιστάμενους (supervisees) του James Borg μέχρι 2 επίπεδα κάτω.

- Απάντηση:**

- $BORG\_SSN \leftarrow \pi_{SSN}(\sigma_{Fname="James" \text{ AND } Lname="Borg"}(EMPLOYEE))$
- $SUPERVISION(SSN1, SSN2) \leftarrow \pi_{SSN, Super\_ssn}(EMPLOYEE)$



// Υφιστάμενοι του James Borg σε επίπεδο 1

- $RESULT1(SSN) = \pi_{SSN1}(SUPERVISION \otimes_{SSN2=SSN} BORG\_SSN)$

// Υφιστάμενοι των Άμεσα Υφιστάμενων του James Borg (σε επίπεδο 2)

- $RESULT2(SSN) = \pi_{SSN1}(SUPERVISION \otimes_{SSN2=SSN} RESULT1)$

- $RESULT \leftarrow RESULT2 \cup RESULT1$

# Επιπλέον Σχεσιακοί Τελεστές (Αναδρομική Κλειστότητα)



**BORG SSN: 888665555**

**Υφιστάμενοι Borg Επιπέδου 1**

(SSN) (SUPERSSN)

SUPERVISION	SSN1	SSN2
	123456789	333445555
	333445555	888665555
	999887777	987654321
	987654321	888665555
	666884444	333445555
	453453453	333445555
	987987987	987654321

RESULT 1	SSN
	333445555
	987654321

**Υφιστάμενοι Borg Επιπέδου 2**

RESULT 2	SSN
	123456789
	999887777
	666884444
	453453453
	987987987

RESULT	SSN
	123456789
	999887777
	666884444
	453453453
	987987987
	333445555
	987654321

**RESULT2  $\cup$  RESULT1**

# Επιπλέον Σχεσιακοί Τελεστές

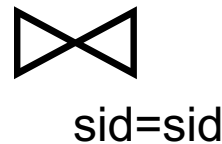
(Εξωτερική Συνένωση, Outer Join =  $\bowtie$ ,  $\bowtie$  =, =  $\bowtie$  =)



## Εξωτερική Συνένωση (Outer Join) – Παράδειγμα

**|Sailors|=5**

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



**|Reserves|=6**

sid	bid	day
28	103	12/4/21
28	103	11/3/21
31	101	10/10/21
31	102	10/12/21
31	101	10/11/21
58	103	11/12/21

- Σε μια **Θ-συνένωση** το αποτέλεσμα περιλαμβάνει **ΜΟΝΟ** τις πλειάδες που έχουν το ίδιο γνώρισμα συνένωσης (δηλ., το sid):

– **SAILOR**  $\bowtie_{\text{sid=sid}}$  **RESERVES** = {(28, yuppy, 9, 35.0, 103, 12/4/21), (28, yuppy, 9, 35.0, 103, 11/3/21), (31, lubber, 8, 55.5, 101, 10/10/21), (31, lubber, 8, 55.5, 102, 10/12/21), (31, lubber, 8, 55.5, 101, 10/11/21), (58, rusty, 10, 35.0, 103, 11/12/21)}

- Συχνά, θέλουμε να έχουμε στο αποτέλεσμα **ΟΛΕΣ** τις πλειάδες της **ΑΡΙΣΤΕΡΗΣ** σχέσης, δηλ., να περιλαμβάνει και τις πλειάδες **(22, dustin, 7, 45.0, NULL, NULL)**, **(44, guppy, 5, 35.0, NULL, NULL)**
  - Το πιο πάνω είναι παράδειγμα **Αριστερής Εξ. Συν.** (**left outer join, = $\bowtie$** )
  - Κατά αντίστοιχο τρόπο δημιουργούνται και οι έννοιες της **δεξιάς** (**right outer join,  $\bowtie$  =**) και **Πλήρης Εξωτερικής συνένωσης** (**full outer join, = $\bowtie$  =**)

# Επιπλέον Σχισιακοί Τελεστές

(Εξωτερική Συνένωση, Outer Join =  $\otimes$ ,  $\otimes=$ ,  $=\otimes=$ )



**EMPLOYEE**  $\otimes_{DNO=Dnumber}$  **DEPARTMENT\_SUB**

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	2
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	2
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	2
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	2
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	2
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Τυπικά, το σενάριο αυτό δεν είναι εφικτό λόγω του κανόνα αναφορικής ακεραιότητας, προκύπτει ωστόσο σε συνένωση μη-κλειδιών

DEPARTMENT\_SUB

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

RESULT

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

Η εξωτερική συνένωση είναι χρήσιμη για παραγωγή αποτελεσμάτων που θέλουν στο αποτέλεσμα όλες τις εγγραφές μιας οντότητας ανεξάρτητα εάν συνενώνονται ή όχι.

# Παραδείγματα Σχεσιακής Άλγεβρας (Σχήμα UNIVERSITY)



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

## DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------



# Παραδείγματα Σχεσιακής Άλγεβρας



## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

- **Q1: Ανάκτησε το name και address όλων των υπαλλήλων που δουλεύουν για το 'Research' department.**
  1. RESEARCH\_DEPT  $\leftarrow \sigma_{\text{DNAME}='Research'}$  (DEPARTMENT)
  2. RESEARCH\_EMPS  $\leftarrow$  (RESEARCH\_DEPT  $\otimes_{\text{DNUMBER}=\text{DNO}}$  EMPLOYEE)
  3. RESULT  $\leftarrow \pi_{\text{FNAME, LNAME, ADDRESS}}$  (RESEARCH\_EMPS)

\* Η σειρά των  $\sigma$ - $\pi$ - $\otimes$  θα μπορούσε να αλλάξει λαμβάνοντας πίσω το ίδιο αποτέλεσμα (π.χ., η σειρά των joins στο 2)

# Παραδείγματα Σχεσιακής Άλγεβρας



EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

PROJECT

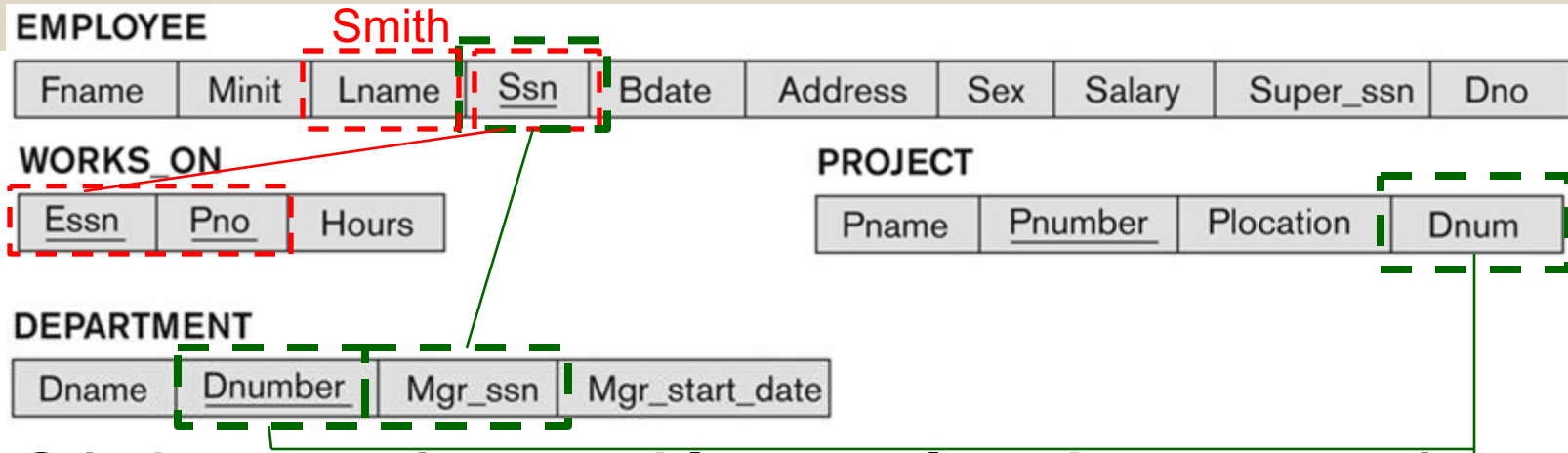
Pname	<u>Pnumber</u>	Plocation	<u>Dnum</u>
-------	----------------	-----------	-------------

- Q3: Βρες το **όνομα** των υπαλλήλων που **δουλεύουν** πάνω σε **όλα τα projects** που **ελέγχονται** από το **department 5**.

1.  $DEPT5\_PROJS(Pno) \leftarrow \pi_{Pnumber}(\sigma_{DNUM=5}(PROJECT))$
2.  $EMP\_PROJ(Ssn, Pno) \leftarrow \pi_{ESSN, Pno}(WORKS\_ON)$
3.  $RESULT\_EMP\_SSNS \leftarrow EMP\_PROJ / DEPT5\_PROJS$
4.  $RESULT \leftarrow \pi_{LNAME, FNAME}(RESULT\_EMP\_SSNS * EMPLOYEE)$

\* Natural join

# Παραδείγματα Σχεσιακής Άλγεβρας



- **Q4: Δημιουργήστε μια λίστα από projects τα οποία περιλαμβάνουν ένα υπάλληλο με το επίθετο “Smith”, ως υπάλληλο ή\* ως manager του τμήματος που ελέγχει το εν λόγω project.**

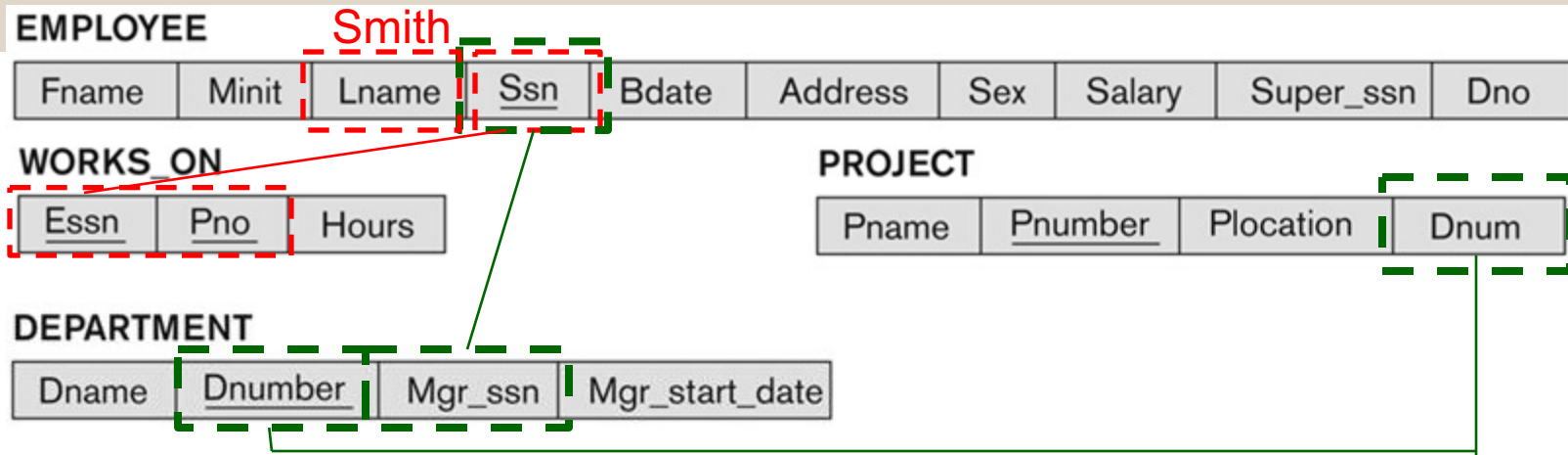
- Αναλύοντας το πιο πάνω ερώτημα βρίσκουμε ότι το αποτέλεσμα θα πρέπει να είναι της μορφής (δηλ., να είναι ένωση αποτελεσμάτων)

$$\pi_{Pnumber}(\mathbf{SMITH\_WORKER\_PROJS} \cup \mathbf{SMITH\_MGR\_PROJS})$$

- Στην επόμενη διαφάνεια δείχνουμε αναλυτικά την απάντηση.

\* Εάν το ή ήταν συμμετρική διάφορα (δηλαδή είτε...ή) τότε θα έπρεπε να εφαρμόσουμε την ισοδυναμία  $R \oplus S = (R - S) \cup (S - R)$

# Παραδείγματα Σχεσιακής Άλγεβρας



- Q4: Δημιουργήστε μια λίστα από projects τα οποία περιλαμβάνουν ένα υπάλληλο με το επίθετο “Smith”, ως υπάλληλο ή ως manager του τμήματος που ελέγχει το εν λόγω project.

$$SMITHS \leftarrow \pi_{SSN}(\sigma_{Lname='Smith'}(EMPLOYEE))$$

$$SMITH\_WORKER\_PROJS \leftarrow \pi_{Pno} (SMITHS \otimes_{SSN=ESSN} WORKS\_ON)$$

$$SMITH\_MNG\_DEPTS \leftarrow \pi_{DNumber} (SMITHS \otimes_{SSN=MGR\_SSN} DEPARTMENT)$$

$$SMITH\_MGR\_PROJS(PNO) \leftarrow \pi_{Pnumber} (SMITH\_MNG\_DEPTS \otimes_{DNumber=DNum} PROJECT)$$

$$RESULT \leftarrow \pi_{Pno} (SMITH\_WORKER\_PROJS \cup SMITH\_MGR\_PROJS(PNO))$$

# Παραδείγματα Σχεσιακής Άλγεβρας

## Sailors-Reserve-Boats



Βρες τα ονόματα των **sailors** που κράτησαν  
μια **κόκκινη βάρκα**

**Sailors**(sid:integer, sname:string, rating:integer, age:real)

**Boats**(bid:integer, bname:string, color:string)

**Reserves**(sid:integer, bid:integer, day:date)

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

❖ Μια πιο αποδοτική διατύπωση\*:

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

*Διατήρηση μόνο των απαραίτητων γνωρισμάτων*

\* Είναι πιο αποδοτικό διότι τα ενδιάμεσα αποτελέσματα είναι μικρότερα

- π.χ., στο  $\pi_{bid}(\sigma_{color='red'} Boats)$  διατηρούμε ως **ενδιάμεσο αποτέλεσμα** μόνο το **bid** **αντί** και τα τρία