

# Κεφάλαιο 3

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ B' Μέρος

## Το Σχεσιακό Μοντέλο – ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

- **Σχεσιακή Άλγεβρα**
  - Επέκταση της Σχεσιακής Άλγεβρας
  - Παραδείγματα
- **Όψεις – Κανόνες Ακεραιότητας**
- **Πράξεις Αλλαγών / Ενημερώσεων**

## ΤΥΠΙΚΟΣ (Formal) ΟΡΙΣΜΟΣ ΕΚΦΡΑΣΗΣ ΣΤΗ ΣΧΕΣΙΑΚΗ ΑΛΓΕΒΡΑ

- Μία έκφραση στη Σχεσιακή Άλγεβρα αποτελείται από ένα εκ των:
  - Μία Σχέση στη Βάση Δεδομένων
  - Μια σταθερή Σχέση
- Αν  $E_1$  και  $E_2$  είναι εκφράσεις στη Σχεσιακή Άλγεβρα, τότε και τα παρακάτω είναι όλα εκφράσεις στη Σχεσιακή Άλγεβρα:
  - $E_1 \cup E_2$
  - $E_1 - E_2$
  - $E_1 \times E_2$
  - $\sigma_P(E_1)$ ,  $P$  είναι συνθήκη σε γνωρίσματα του  $E_1$
  - $\Pi_S(E_1)$ ,  $S$  είναι μια λίστα από γνωρίσματα του  $E_1$
  - $\rho_x(E_1)$ ,  $x$  είναι το νέο όνομα του αποτελέσματος του  $E_1$

# Example Queries

Find the names of all customers who have a loan at the Perryridge branch.

**loan** (loan-number, branch-name, amount)

**borrower** (customer-name, loan-number)

$\pi_{\text{customer-name}}(\sigma_{\text{loan.loan-number} = \text{borrower.loan-number}}(\sigma_{\text{branch-name} = \text{“Perryridge”}}(\text{borrower} \times \text{loan})))$   
**loan x borrower**

<i>customer-name</i>	<i>borrower.loan-number</i>	<i>loan.loan-number</i>	<i>branch-name</i>	<i>amount</i>
Adams	L-16	L-11	Round Hill	900
Adams	L-16	L-14	Downtown	1500
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Adams	L-16	L-17	Downtown	1000
Adams	L-16	L-23	Redwood	2000
Adams	L-16	L-93	Mianus	500
Curry	L-93	L-11	Round Hill	900
Curry	L-93	L-14	Downtown	1500
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Curry	L-93	L-17	Downtown	1000
Curry	L-93	L-23	Redwood	2000
Curry	L-93	L-93	Mianus	500
Hayes	L-15	L-11		900
Hayes	L-15	L-14		1500
Hayes	L-15	L-15		1500
Hayes	L-15	L-16		1300
Hayes	L-15	L-17		1000
Hayes	L-15	L-23		2000
Hayes	L-15	L-93		500
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Smith	L-23	L-11	Round Hill	900
Smith	L-23	L-14	Downtown	1500
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Smith	L-23	L-17	Downtown	1000
Smith	L-23	L-23	Redwood	2000
Smith	L-23	L-93	Mianus	500
Williams	L-17	L-11	Round Hill	900
Williams	L-17	L-14	Downtown	1500
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300
Williams	L-17	L-17	Downtown	1000
Williams	L-17	L-23	Redwood	2000
Williams	L-17	L-93	Mianus	500

$\sigma_{\text{branch-name} = \text{“Perryridge”}}(\text{borrower} \times \text{loan})$

<i>customer-name</i>	<i>borrower.loan-number</i>	<i>loan.loan-number</i>	<i>branch-name</i>	<i>amount</i>
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Hayes	L-15	L-15	Perryridge	1500
Hayes	L-15	L-16	Perryridge	1300
Jackson	L-14	L-15	Perryridge	1500
Jackson	L-14	L-16	Perryridge	1300
Jones	L-17	L-15	Perryridge	1500
Jones	L-17	L-16	Perryridge	1300
Smith	L-11	L-15	Perryridge	1500
Smith	L-11	L-16	Perryridge	1300
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300

result

<i>customer-name</i>
Adams
Hayes

## Example Queries

Find the names of all customers who have a loan at the Perryridge branch.

Query 1

$$\pi_{\text{customer-name}}(\sigma_{\text{branch-name} = \text{"Perryridge"}}(\sigma_{\text{borrower.loan-number} = \text{loan.loan-number}}(\text{borrower} \times \text{loan})))$$

Query 2

$$\pi_{\text{customer-name}}(\sigma_{\text{loan.loan-number} = \text{borrower.loan-number}}((\sigma_{\text{branch-name} = \text{"Perryridge"}}(\text{loan})) \times \text{borrower}))$$

## Example Queries

Find the largest account balance

- Rename *account* relation as *d*
- The query is:

$$\pi_{\text{balance}}(\text{account}) - \pi_{\text{account.balance}}(\sigma_{\text{account.balance} < \text{d.balance}}(\text{account} \times \rho_{\text{d}}(\text{account})))$$

account

<i>account-number</i>	<i>branch-name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

$\sigma_{\text{account.balance} < \text{d.balance}}$

<i>balance</i>
500
400
700
750
350

result

<i>balance</i>
900

## Additional Operations

Ορίζουμε επιπλέον πράξεις που δεν προσθέτουν κάτι στη δύναμη της σχεσιακής άλγεβρας αλλά απλοποιούν συχνά queries

- Set intersection
- Natural join
- Division
- Assignment

## Set-Intersection Operation

- Βάλτε όλες τις κοινές πλειάδες των δύο Σχέσεων σε μια Σχέση
  - **Συμβολισμός:**  $R \cap S$
  - **Τυπικά:**  $R \cap S = \{t \mid t \text{ is in } R \text{ and } t \text{ is in } S\}$
  - **Παράδειγμα:**  $\text{STUDENT} \cap \text{INSTRUCTOR}$
- Assume:
  - attributes of  $r$  and  $s$  are compatible
- Note:  $r \cap s = r - (r - s)$



## Set-Intersection Operation - Example

- Relation  $r, s$ :

A	B
$\alpha$	1
$\alpha$	2
$\beta$	1

$r$

A	B
$\alpha$	2
$\beta$	3

$s$

- $r \cap s$

A	B
$\alpha$	2

## Παράδειγμα Τράπεζας

### Customers With Both an Account and a Loan at the Bank

$$\Pi_{\text{customer-name}}(\text{depositor}) \cap \Pi_{\text{customer-name}}(\text{borrower})$$

borrower

<i>customer-name</i>	<i>loan-number</i>
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

depositor

<i>customer-name</i>	<i>account-number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

result

<i>customer-name</i>
Hayes
Jones
Smith

## Φυσική Συνένωση – Natural Join

- Συνδύασε πλειάδες δύο Σχέσεων που ταιριάζουν (έχουν την ίδια τιμή) σε *όλα τα κοινά γνωρίσματα*. Στο αποτέλεσμα, τα κοινά γνωρίσματα κρατούνται μόνο μια φορά

– **Συμβολισμός**:  $R \bowtie S$  ή  $R [ \mathbf{X} = \mathbf{X} ] S$

– Αν  $R = (A, B, C, D)$

$S = (E, B, D)$

Result schema =  $(A, B, C, D, E)$

– Παρατηρήστε ότι  $R \bowtie S$  ορίζεται ως:

$\Pi_{R.A, R.B, R.C, R.D, S.E} (\sigma_{R.B = S.B \wedge R.D = S.D} (r \times s))$

– **Παράδειγμα**: borrower  $\bowtie$  loan

## Natural Join Operation – Example

### ■ Relations r, s:

A	B	C	D
$\alpha$	1	$\alpha$	a
$\beta$	2	$\gamma$	a
$\gamma$	4	$\beta$	b
$\alpha$	1	$\gamma$	a
$\delta$	2	$\beta$	b

r

B	D	E
1	a	$\alpha$
3	a	$\beta$
1	a	$\gamma$
2	b	$\delta$
3	b	$\epsilon$

s

$r \bowtie s$

A	B	C	D	E
$\alpha$	1	$\alpha$	a	$\alpha$
$\alpha$	1	$\alpha$	a	$\gamma$
$\alpha$	1	$\gamma$	a	$\alpha$
$\alpha$	1	$\gamma$	a	$\gamma$
$\delta$	2	$\beta$	b	$\delta$

## Παράδειγμα Τράπεζας (1)

- Βρες το δάνειο και το αντίστοιχο ποσό που έχει δανειστεί κάθε πελάτης
- $\Pi_{customer-name, loan-number, amount} (borrower \bowtie loan)$

*borrower*

<i>customer-name</i>	<i>loan-number</i>
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

*loan*

<i>loan-number</i>	<i>branch-name</i>	<i>amount</i>
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

*result*

<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Adams	L-16	1300
Curry	L-93	500
Hayes	L-15	1500
Jackson	L-14	1500
Jones	L-17	1000
Smith	L-23	2000
Smith	L-11	900
Williams	L-17	1000

## Παράδειγμα Τράπεζας (2)

- Σε ποιο υποκατάστημα έχει λογαριασμό ο κάθε πελάτης
- Result of  $\Pi_{customer-name, branch-name}(depositor \bowtie account)$

account

account-number	branch-name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

depositor

customer-name	account-number
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

result

customer-name	branch-name
Hayes	Perryridge
Johnson	Downtown
Johnson	Brighton
Jones	Brighton
Lindsay	Redwood
Smith	Mianus
Turner	Round Hill

## Διαίρεση - Division

- Όταν δίνονται οι σχέσεις

$$R = (A_1, \dots, A_m, B_1, \dots, B_n)$$

$$S = (B_1, \dots, B_n)$$

Το αποτέλεσμα της διαίρεσης  $R/S$  είναι μια σχέση στο

$R - S = (A_1, \dots, A_m)$  τέτοια ώστε για κάθε  $t_s$  στην  $S$  υπάρχει κάποιο  $t$  στην  $R/S$  τέτοιο ώστε  $t t_s$  να ανήκει στο  $R$ .

– Αναλογία με την αριθμητική:

Το **πηλίκιο**  $q$  της  $a/b$  είναι ο μεγαλύτερος αριθμός έτσι ώστε  $qb \leq a$

Το **πηλίκιο**  $Q$  της  $R / S$  είναι η μέγιστη σχέση (σε πληθυσμό) έτσι ώστε

$$Q \times S \subseteq R$$

Χρήσιμη Πράξη για ερωτήσεις που περιέχουν το «για όλα»

## Διαίρεση – Παράδειγμα (1)

Relations r, s:

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ε	6
ε	1
β	2

B
1
2

s

r / s:

A
α
β

r

Θέλουμε τις τιμές του A που στην r συνδυάζονται με όλες τις τιμές του B της s



## Διαίρεση – Παράδειγμα (2)

Relations r, s:

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

r

D	E
a	1
b	1

s

r / s:

A	B	C
α	a	γ
γ	a	γ

## Εκχώρηση - Assignment

- Η Λειτουργία εκχώρησης ( $\leftarrow$ ) προσφέρει μια διευκόλυνση στην έκφραση πολύπλοκων επερωτήσεων, γράφοντας αυτές σαν σειριακά προγράμματα που αποτελούνται από
  - Μια σειρά εκχωρήσεων
  - Ακολουθούμενη από μια έκφραση η τιμή της οποίας παρουσιάζεται ως το αποτέλεσμα της επερώτησης
- ΠΑΡΑΔΕΙΓΜΑ:

$$temp1 \leftarrow \Pi_{R-S}(r)$$

$$temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$$

$$result = temp1 - temp2$$

- Το αποτέλεσμα στα δεξιά του  $\leftarrow$  εκχωρείται στη σχεσιακή μεταβλητή στα αριστερά του  $\leftarrow$ .
- Οι επόμενες εκφράσεις ΜΠΟΡΟΥΝ να τη χρησιμοποιήσουν

## Εκτεταμένες πράξεις Σχεσιακής Άλγεβρας

- Generalized Projection
- Aggregate Functions
- Outer Join

## Γενικευμένη Προβολή - Generalized Projection

- Επέκταση του projection που επιτρέπει αριθμητικές συναρτήσεις στη λίστα των γνωρισμάτων

$$\Pi_{F_1, F_2, \dots, F_n}(E)$$

- $E$  είναι οποιαδήποτε έκφραση σχεσιακής άλγεβρας
- Κάθε  $F_1, F_2, \dots, F_n$  είναι αριθμητικές εκφράσεις που συμπεριλαμβάνουν σταθερές και γνωρίσματα στο σχήμα του  $E$
- Αν θεωρήσουμε σχέση  $credit-info(customer-name, limit, credit-balance)$ , βρες πόσα ακόμα μπορεί να ξοδέψει κάθε πελάτης

$$\Pi_{customer-name, limit - credit-balance}(credit-info)$$

## Συν-αθροιστικές Συναρτήσεις και Πράξεις

- Μια **Aggregation function** δέχεται μια συλλογή από τιμές και επιστρέφει μια τιμή ως αποτέλεσμα.

**avg:** average value

**min:** minimum value

**max:** maximum value

**sum:** sum of values

**count:** number of values

- Μια **Συν-αθροιστική Πράξη** στη σχεσιακή άλγεβρα:

$$G_1, G_2, \dots, G_n \quad \mathcal{g} \quad F_1(A_1), F_2(A_2), \dots, F_n(A_n) \quad (E)$$

- $E$  έκφραση στη σχεσιακή άλγεβρα
- $G_1, G_2, \dots, G_n$  λίστα γνωρισμάτων για ομαδοποίηση
- Κάθε  $F_i$  είναι μια συν-αθροιστική συνάρτηση
- Κάθε  $A_i$  είναι το όνομα ενός γνωρίσματος

## Aggregate Operation – Παράδειγμα

- Σχέση  $r$ :

$A$	$B$	$C$
$\alpha$	$\alpha$	7
$\alpha$	$\beta$	7
$\beta$	$\beta$	3
$\beta$	$\beta$	10

$g_{\text{sum}(c)}(r)$

$\text{sum-C}$
27

## Aggregate Operation – Example

- Σχέση *account* grouped by (ομαδοποιημένο σε) *branch-name*:

<i>branch-name</i>	<i>account-number</i>	<i>balance</i>
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

*branch-name*  $\mathcal{g}$  *sum(balance)* (*account*)

<i>branch-name</i>	<i>balance</i>
Perryridge	1300
Brighton	1500
Redwood	700

## Συνένωση - Join

- Υπάρχουν πολλές μορφές Συνένωσης – όλες συνδυάζουν δύο σχέσεις για την δημιουργία μιας τρίτης
  - *(theta) join, equality join, semi-join, outer join*



## Θ-Συνένωση – Theta join

- Συνδύασε πλειάδες δύο Σχέσεων που ταιριάζουν (πληρούν κάποια Boolean συνθήκη  $c$ ) σε κάποια προσδιοριζόμενα γνωρίσματα
  - **Συμβολισμός**:  $R \bowtie_c S$  ή  $R [c] S$
  - Μια Θ-Συνένωση είναι ισοδύναμη ενός Καρτεσιανού γινομένου που ακολουθείται από μια επιλογή με συνθήκη  $c$ .

$$R \bowtie_c S = \sigma_c (R \times S)$$

- Η Σχέση – αποτέλεσμα έχει ΟΛΑ τα γνωρίσματα της  $R$  και της  $S$

## Θ-Συνένωση - Παράδειγμα

- EMPLOYEE(eid, ename, esalary, mngrId)
- MANAGER(mid, mname, msalary)
- Βρες τα ονόματα των υπαλλήλων που βγάζουν περισσότερα από τους managers τους
- $\Pi_{\text{ename}}(\text{EMPLOYEE} \bowtie_{\text{mngrId} = \text{mid and esalary} > \text{msalary}} \text{MANAGER})$

## Equality join (ή Equijoin)

- Συνδύασε πλειάδες δύο Σχέσεων που ταιριάζουν (έχουν την ίδια τιμή) σε κάποια γνωρίσματα. Είναι ειδική περίπτωση της  $\Theta$ -Συνένωσης όπου η συνθήκη είναι *ισότητα*.

– **Συμβολισμός:**  $R \bowtie_c S$       ή       $R [c] S$

– **Παράδειγμα:**  $EMPLOYEE \bowtie_{mgrId = mid} MANAGER$

## Ημι-συνένωση – Semi-join

- Επίλεξε το υποσύνολο μιας Σχέσης που συνενώνεται με μια άλλη. Μια semi-join είναι ισοδύναμη με μια συνένωση ακολουθούμενη από μια προβολή.
  - **Συμβολισμός** :  $R \bowtie_c S$   
Ισοδυναμεί με  $\pi_{a_1, \dots, a_n}(R \bowtie S)$  αν  $a_1, \dots, a_n$  τα attributes της  $R$
  - **Παράδειγμα**: Βρες όλες τις πληροφορίες των υπαλλήλων που είναι managers σε κάποιο τμήμα

EMPLOYEE  $\bowtie_{SSN=MgrSSN}$  DEPARTMENT

## Εξωτερική Συνένωση – Outer Join

- Επέκταση του join για την αποφυγή απώλειας πληροφορίας
- *Κίνητρο*: Σε μια τυπική συνένωση, πλειάδες της μιας σχέσης που δεν ταιριάζουν με καμία πλειάδα της άλλης σχέσης *δεν παρουσιάζονται* στο αποτέλεσμα.
- Σε ένα outer join όλες οι πλειάδες πρέπει να παρουσιαστούν στο αποτέλεσμα - όταν δεν υπάρχουν ταιριάζουσες πλειάδες, τότε οι ειδικές τιμές NULL τοποθετούνται για τις τιμές γνωρισμάτων που λείπουν.
- Χωρίζονται σε:
  - Left outer join:  $R \sqsupset \bowtie S$  (όλες οι πλειάδες στην R παρουσιάζονται)
  - Right outer join:  $R \bowtie \sqsupset S$  (όλες οι πλειάδες στην S παρουσιάζονται)
  - Full outer join:  $R \sqsupset \bowtie \sqsupset S$  (όλες οι πλειάδες στην R και S παρουσιάζονται)

## Joins – Example

loan

loan-number	branch-name	amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

borrower

customer-name	loan-number
Jones	L-170
Smith	L-230
Hayes	L-155

### Inner Join

*loan* ⋈ *Borrower*

loan-number	branch-name	amount	customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

## Outer Join – Example

loan

loan-number	branch-name	amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

borrower

customer-name	loan-number
Jones	L-170
Smith	L-230
Hayes	L-155

### Left Outer Join

*loan*  *Borrower*

loan-number	branch-name	amount	customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	null

## Outer Join – Example

loan

loan-number	branch-name	amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

borrower

customer-name	loan-number
Jones	L-170
Smith	L-230
Hayes	L-155

### Right Outer Join

*loan* ⋈<sub>r</sub> *Borrower*

loan-number	branch-name	amount	customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	null	null	Hayes



## Outer Join – Example

loan

loan-number	branch-name	amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

borrower

customer-name	loan-number
Jones	L-170
Smith	L-230
Hayes	L-155

### Full Outer Join

*loan* ⋈ *Borrower*

loan-number	branch-name	amount	customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	null
L-155	null	null	Hayes

## Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (Queries)

- Ένα σύνολο ερωταποκρίσεων στην Σχεσιακή Άλγεβρα παρουσιάζεται στην συνέχεια, με χρήση μιας Βάσης Δεδομένων που περιλαμβάνει ναυτικούς (*SAILORS*) οι οποίοι κάνουν κρατήσεις (*RESERVE*) για κάποια σκάφη (*BOATS*.)

SAILORS (Sid, SName, Rating)

BOATS (Bid, BName, Color)

RESERVE (Sid, Bid, Date)

## Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (2)

SAILORS (Sid, SName, Rating)

BOATS (Bid, BName, Color)

RESERVE (Sid, Bid, Date)

- QUERY1: Βρες τα ονόματα των SAILORS που έχουν κρατήσει το BOAT με νούμερο 2

$\pi_{SName} (\sigma_{Bid=2} RESERVE \bowtie_{Sid=Sid} SAILORS)$

- QUERY2: Βρες τα ονόματα των SAILORS που έχουν κρατήσει BOAT με χρώμα κόκκινο

$\pi_{SName} (\sigma_{Color=red} BOAT \bowtie_{Bid=Bid} RESERVE \bowtie_{Sid=Sid} SAILORS)$

## Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (3)

SAILORS (Sid, SName, Rating)

BOATS (Bid, BName, Color)

RESERVE (Sid, Bid, Date)

- QUERY3: Βρες τα χρώματα των σκαφών που έχει κρατήσει η ναυτικός με το όνομα Ελένη

$\pi_{\text{Color}} ( \sigma_{\text{SName}=\text{eleni}} \text{SAILORS} \bowtie_{\text{Sid}=\text{Sid}} \text{RESERVE} \bowtie_{\text{Bid}=\text{Bid}} \text{BOATS} )$

- QUERY4: Βρες τα ονόματα των ΝΑΥΤΙΚΩΝ που έχουν κρατήσει τουλάχιστον ένα ΣΚΑΦΟΣ

$\pi_{\text{SName}} ( \text{RESERVE} \bowtie_{\text{Sid}=\text{Sid}} \text{SAILORS} )$

## Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (4)

SAILORS (Sid, SName, Rating)

BOATS (Bid, BName, Color)

RESERVE (Sid, Bid, Date)

- QUERY5: Βρες τα ονόματα των ΝΑΥΤΙΚΩΝ που έχουν κρατήσει ένα κόκκινο ή ένα πράσινο ΣΚΑΦΟΣ

$$\pi_{SName} ((\sigma_{Color=red} BOATS \cup \sigma_{Color=green} BOATS) \bowtie_{Bid=Bid} RESERVE \bowtie_{Sid=Sid} SAILORS)$$

- QUERY6: Βρες τα ονόματα των ΝΑΥΤΙΚΩΝ που έχουν κρατήσει ένα κόκκινο και ένα πράσινο ΣΚΑΦΟΣ

$$\pi_{SName} ((\pi_{Sid} (\sigma_{Color=red} BOATS \bowtie_{Bid=Bid} RESERVE) \cap \pi_{Sid} (\sigma_{Color=green} BOATS \bowtie_{Bid=Bid} RESERVE)) \bowtie_{Sid=Sid} SAILORS)$$

## Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (5)

SAILORS (Sid, SName, Rating)

BOATS (Bid, BName, Color)

RESERVE (Sid, Bid, Date)

- QUERY7: Βρες τα ονόματα των ΝΑΥΤΙΚΩΝ που έχουν κρατήσει όλα τα ΣΚΑΦΗ

$$\pi_{\text{SName}} (( \pi_{\text{Sid, Bid}} \text{RESERVE} / \pi_{\text{Bid}} \text{BOATS} ) \bowtie_{\text{Sid=Sid}} \text{SAILORS})$$

- QUERY8: Βρες τα ονόματα και διαβαθμίσεις των ΝΑΥΤΙΚΩΝ που έχουν κρατήσει όλα τα κόκκινα ΣΚΑΦΗ

$$\pi_{\text{SName, Rating}} (( \pi_{\text{Sid, Bid}} \text{RESERVE} / \pi_{\text{Bid}} (\sigma_{\text{Color=red}} \text{BOATS}) ) \bowtie_{\text{Sid=Sid}} \text{SAILORS})$$

## Σχεσιακή Άλγεβρα: ΣΧΟΛΙΑ (1)

- Υπάρχουν διάφορες ιδιότητες που ισχύουν σε μια έκφραση της Σχεσιακής Άλγεβρας (μεταβατική, αντιμεταθετική, προσεταιριστική, κλπ.)

– **Παραδείγματα:**

$$\sigma_{c1}(\pi_L(R)) = \pi_L(\sigma_{c1}(R))$$

$$\sigma_{c1}(R \bowtie_{c2} S) = \sigma_{c1}(R) \bowtie_{c2} S$$

$$\sigma_{c1}(R \cup S) = \sigma_{c1}(R) \cup \sigma_{c1}(S)$$

....

- Αυτές οι ιδιότητες είναι εξαιρετικά χρήσιμες στην βελτιστοποίηση των ερωταποκρίσεων (*query optimization*)

## Σχεσιακή Άλγεβρα: ΣΧΟΛΙΑ (2)

### ■ ΠΛΗΡΕΣ ΣΥΝΟΛΟ ΠΡΑΞΕΩΝ

– Το σύνολο των πράξεων  $\{ \sigma, \pi, \cup, -, X, \rho \}$  ονομάζεται **πλήρες** σύνολο πράξεων της Σχεσιακής Άλγεβρας. Η επαγωγή είναι ότι ΟΛΕΣ οι άλλες πράξεις μπορεί να περιγραφούν ως μια ακολουθία πράξεων από το σύνολο αυτό (το σύνολο αυτό έχουμε δει στον τυπικό ορισμό της έκφρασης σχεσιακής άλγεβρας )

– Για παράδειγμα, η *διαίρεση* περιγράφεται ως:

$$R / S = \pi_X(R) - ( (\pi_X(R) \times S) - R )$$

όπου  $X$  είναι τα μη κοινά γνωρίσματα στην  $R$  με την  $S$

– Ισοδυνάμως, περιγράφεται ως:

$$(R / S) [X] = R[X] - ( (R[X] \times S) - R ) [X]$$



## Σχεσιακή Άλγεβρα: ΠΛΗΡΟΤΗΤΑ

- Υπάρχουν αρκετοί συνδυασμοί πράξεων που συγκροτούν ένα πλήρες σύνολο πράξεων.
- Κάθε Γλώσσα Ερωταποκρίσεων ισοδύναμη με ένα πλήρες σύνολο πράξεων ονομάζεται Σχεσιακά Πλήρης (RELATIONALLY COMPLETE)

## Null Values (Κενές Τιμές)

- Είναι δυνατόν να έχουμε κενές τιμές για κάποια από τα γνωρίσματα σε μια πλειάδα (null)
- Το *null* υποδηλώνει ότι η τιμή είναι ΑΓΝΩΣΤΗ ή ότι ΔΕΝ ΥΠΑΡΧΕΙ
- Το αποτέλεσμα μιας αριθμητικής πράξης που έχει μέσα το *null* είναι *null*.
- Οι συν-αθροιστικές συναρτήσεις αγνοούν το null
  - Αυτή είναι μια τυχαία απόφαση (θα μπορούσαν ενδεχομένως να επιστρέφουν σαν αποτέλεσμα το null).
- Δυο null θεωρούνται ότι είναι ίσα

## Κενές Τιμές

- Συγκρίσεις με `null` επιστρέφουν την τιμή *unknown*
  - Αν επέστρεφε *false* αντί για *unknown*, τότε  $\text{not } (A < 5)$   
θα ήταν ισοδύναμο με  $A \geq 5$
- Εφαρμόζεται μια Λογική τριών τιμών (`true`, `false`, *unknown*):
  - OR: (*unknown* **or** *true*) = *true*,  
(*unknown* **or** *false*) = *unknown*  
(*unknown* **or** *unknown*) = *unknown*
  - AND: (*true* **and** *unknown*) = *unknown*,  
(*false* **and** *unknown*) = *false*,  
(*unknown* **and** *unknown*) = *unknown*
  - NOT: (**not** *unknown*) = *unknown*

## Αλλαγές των τιμών στη Βάση Δεδομένων

- Το περιεχόμενο της Βάσης αλλάζει με τις παρακάτω πράξεις:
  - Deletion
  - Insertion
  - Updating
- Οι πράξεις αυτές εκφράζονται με τον τελεστή ASSIGNMENT (εκχώρηση)

## Διαγραφή (DELETION)

- Μια Διαγραφή εκφράζεται όπως μια επερώτηση, με τη διαφορά ότι αντί να παρουσιαστεί το αποτέλεσμα, αυτό διαγράφεται.
- Διαγράφονται ΜΟΝΟ Πλειάδες (όχι μεμονωμένες τιμές)
- Στη Σχεσιακή Άλγεβρα, η Διαγραφή παρίσταται με:

$$r \leftarrow r - E$$

όπου  $r$  είναι μια σχέση και  $E$  είναι το αποτέλεσμα μιας σχεσιακής έκφρασης.

## Παραδείγματα ΔΙΑΓΡΑΦΩΝ

**branch** (branch-name, branch-city, assets)

**customer** (customer-name, customer-street, customer-city)

**account** (account-number, branch-name, balance)

**loan** (loan-number, branch-name, amount)

**depositor** (customer-name, account-number)

**borrower** (customer-name, loan-number)

- Διάγραψε όλες τις account εγγραφές στο Perryridge branch.

$account \leftarrow account - \sigma_{branch-name = "Perryridge"}(account)$

- Διάγραψε όλες τις loan εγγραφές με ποσό μεταξύ 0 και 50

$loan \leftarrow loan - \sigma_{amount \geq 0 \text{ and } amount \leq 50}(loan)$

## Παραδείγματα ΔΙΑΓΡΑΦΩΝ

- **branch** (branch-name, branch-city, assets)
  - **customer** (customer-name, customer-street, customer-city)
  - **account** (account-number, branch-name, balance)
  - **loan** (loan-number, branch-name, amount)
  - **depositor** (customer-name, account-number)
  - **borrower** (customer-name, loan-number)
- Διάγραψε όλα τα accounts σε branches του Needham.

$r_1 \leftarrow \sigma_{branch\text{-}city = \text{“Needham”}} (account \bowtie branch)$

$r_2 \leftarrow \Pi_{branch\text{-}name, account\text{-}number, balance} (r_1)$

$r_3 \leftarrow \Pi_{customer\text{-}name, account\text{-}number} (r_2 \bowtie depositor)$

$account \leftarrow account - r_2$

$depositor \leftarrow depositor - r_3$

## ΕΙΣΑΓΩΓΗ (Insertion)

- Για εισαγωγή δεδομένων σε μια Σχέση, είτε:
  - Παρουσιάζουμε μια νέα πλειάδα
  - Ή γράφουμε μια επερώτηση το αποτέλεσμα της οποίας θα εισαχθεί
- Στη σχεσιακή άλγεβρα, η Εισαγωγή συμβολίζεται με:

$$r \leftarrow r \cup E$$

όπου  $r$  είναι μια σχέση και  $E$  είναι μια έκφραση



## Παραδείγματα Εισαγωγής

- Να εισαχθούν τα δεδομένα: Smith has \$1200 in account A-973 at the Perryridge branch.

$$account \leftarrow account \cup \{(\text{"Perryridge"}, \text{A-973}, 1200)\}$$
$$depositor \leftarrow depositor \cup \{(\text{"Smith"}, \text{A-973})\}$$

- Κάνε δώρο σε όλους τους loan customers στο Perryridge branch, ένα \$200 savings account.

$$r_1 \leftarrow (\sigma_{branch-name = \text{"Perryridge"}}(borrower \bowtie loan))$$
$$account \leftarrow account \cup \Pi_{branch-name, account-number, 200}(r_1)$$
$$depositor \leftarrow depositor \cup \Pi_{customer-name, loan-number}(r_1)$$

## ΕΝΗΜΕΡΩΣΗ (Updating)

- Ένας μηχανισμός για αλλαγές επιλεκτικά σε ΤΙΜΕΣ μιας Πλειάδας ΧΩΡΙΣ να αλλάζουν ΟΛΕΣ οι τιμές της ΠΛΕΙΑΔΑΣ
- Χρησιμοποιείται ο συμβολισμός:

$$r \leftarrow \prod_{F_1, F_2, \dots, F_I} (r)$$

όπου τα  $F_i$  είναι ένα εκ των:

- το  $i$ th γνώρισμα του  $r$ , αν το  $i$ th γνώρισμα δεν αλλάζει, ή,
- αν το  $i$ th γνώρισμα πρόκειται να αλλάξει, τότε το  $F_i$  είναι μια έκφραση, που περιλαμβάνει μόνο σταθερές και τα γνωρίσματα του  $r$ , που δίνουν τη νέα τιμή για το γνώρισμα

## Παραδείγματα Ενημέρωσης

- Πλήρωσε τους τόκους αυξάνοντας όλα τα υπόλοιπα λογαριασμών με 5 %

$account \leftarrow \Pi_{AN, BN, BAL * 1.05} (account)$

όπου  $AN$ ,  $BN$  και  $BAL$  είναι το *account-number*, *branch-name* και *balance*, αντίστοιχα.

- Πλήρωσε όλους με υπόλοιπο πάνω από \$10,000 6 % τόκους και Πλήρωσε όλους τους άλλους 5 %

$account \leftarrow \Pi_{AN, BN, BAL * 1.06} (\sigma_{BAL > 10000} (account))$   
 $\cup \Pi_{AN, BN, BAL * 1.05} (\sigma_{BAL \leq 10000} (account))$

## ΏΨΕΙΣ / ΠΡΟΒΟΛΕΣ (VIEWS)

- Στις περισσότερες περιπτώσεις, δεν είναι επιθυμητό να βλέπουν όλοι οι χρήστες ολόκληρο το λογικό μοντέλο (δηλαδή, όλες τις σχέσεις που αποθηκεύονται στη βάση δεδομένων.)
- Ενδεικτικά, θεωρήστε ότι κάποιος θέλει να ξέρει τον αριθμό δανείου για έναν πελάτη, ΑΛΛΑ (π.χ., για λόγους εχεμύθειας ή ασφαλείας) δεν πρέπει να δει το ποσό του δανείου. Αυτός μπορεί να «βλέπει» μια σχέση που περιγράφεται στη σχεσιακή άλγεβρα ως:

$$\Pi_{customer-name, loan-number} (borrower \bowtie loan)$$

- Κάθε σχέση που δεν είναι στο εννοιολογικό (conceptual) μοντέλο αλλά είναι ορατή σε χρήστες ως μια ΙΔΕΑΤΗ ΣΧΕΣΗ (“virtual relation”) ονομάζεται **ΟΨΗ (view)**.

## ΟΡΙΣΜΟΣ ΟΨΗΣ

- Μια όψη ορίζεται με την εντολή **create view** που έχει τη μορφή:  
**create view v as <query expression>**  
όπου <query expression> είναι κάθε νόμιμη έκφραση επερώτησης. Το όνομα της ΟΨΗΣ είναι το *v*.
- Από τη στιγμή που ορίζεται η Όψη, το όνομα της χρησιμοποιείται για αναφορά στην ιδεατή σχέση που η ΟΨΗ δημιουργεί.
- Ο Ορισμός της ΟΨΗΣ δεν δημιουργεί μια νέα Σχέση με πλειάδες το αποτέλεσμα της εκτέλεσης της επερώτησης.
  - Αντίθετα, ο Ορισμός της ΟΨΗΣ σημαίνει ότι δημιουργείται (και κρατιέται) μια έκφραση, που αργότερα αντικαθίσταται στις επερωτήσεις που την χρησιμοποιούν.

## Παραδείγματα ΟΨΕΩΝ

- Έστω η όψη (με ονομασία *all-customer*) που αποτελείται από branches και customers.

**create view *all-customer* as**

$$\begin{aligned} & \Pi_{branch-name, customer-name} (depositor \bowtie account) \\ & \cup \Pi_{branch-name, customer-name} (borrower \bowtie loan) \end{aligned}$$

- Μπορούμε να βρούμε όλους τους customers του Perryridge branch γράφοντας:

$$\Pi_{customer-name} (\sigma_{branch-name = \text{“Perryridge”}} (all-customer))$$

## Ενημερώσεις μέσω ΟΨΕΩΝ

- Ενημερώσεις της Βάσης Δεδομένων εκφραζόμενες μέσω όψεων πρέπει να μεταφραστούν σε ενημερώσεις των πραγματικών Σχέσεων στη Βάση.
- Για παράδειγμα, έστω ότι κάποιος θέλει να δει όλα τα δεδομένα για δάνεια στην *loan* σχέση ΕΚΤΟΣ του *amount*. Η όψη που του δίνεται, *branch-loan*, ορίζεται ως:

**create view *branch-loan* as**

$\Pi_{branch-name, loan-number}(loan)$

- Εφόσον επιτρέπουμε το όνομα μιας ΟΨΗΣ να παρουσιάζεται όπου μπορεί να παρουσιαστεί το όνομα μιας ΣΧΕΣΗΣ, τότε θα μπορούσε να γραφεί:

***branch-loan* ← *branch-loan* ∪ {"Perryridge", L-37}**

## Ενημερώσεις μέσω ΟΨΕΩΝ (συνέχεια)

- Η Εισαγωγή των στοιχείων πρέπει να μεταφραστεί σε μια εισαγωγή στοιχείων στην πραγματική σχέση *loan* από την οποία η *branch-loan* κατασκευάζεται.
- Όμως η εισαγωγή στη σχέση *loan* απαιτεί μια τιμή για το *amount*. Υπάρχουν δύο περιπτώσεις:
  - Απαγόρευσε την εισαγωγή και επέστρεψε κάποιο μήνυμα στο χρήστη
  - Βάλε την πλειάδα (“L-37”, “Perryridge”, *null*) στη σχέση *loan*
- Μερικές τροποποιήσεις μέσω όψεων είναι ΑΔΥΝΑΤΟ να μεταφραστούν σε τροποποιήσεις στις πραγματικές σχέσεις!
  - $\text{create view } v \text{ as } \sigma_{\text{branch-name} = \text{“Perryridge”}}(\text{account})$   
 $v \leftarrow v \cup (\text{L-99}, \text{Downtown}, 23)$
- Άλλες ΔΕΝ είναι δυνατό να μεταφραστούν μονοσήμαντα!
  - $\text{all-customer} \leftarrow \text{all-customer} \cup \{(\text{“Perryridge”}, \text{“John”})\}$ 
    - » Πρέπει να επιλεγθεί κάποιο *loan* και *account*, και να δημιουργηθεί ένα νέο νούμερο για *account / loan*!



## ΟΨΕΙΣ οριζόμενες μέσω άλλων ΟΨΕΩΝ

- Μία ΟΨΗ μπορεί να χρησιμοποιηθεί στην ορίζουσα έκφραση μιας άλλης ΟΨΗΣ
- Μια όψη  $v_1$  λέγεται ότι *εξαρτάται άμεσα* από μια όψη  $v_2$  αν η  $v_2$  χρησιμοποιείται στην έκφραση που ορίζει την  $v_1$
- Μια όψη  $v_1$  λέγεται ότι *εξαρτάται* από μια όψη  $v_2$  αν είτε η  $v_1$  εξαρτάται άμεσα από την  $v_2$  ή υπάρχει ένα μονοπάτι εξαρτήσεων από τη  $v_1$  στη  $v_2$
- Μια όψη  $v$  λέγεται *αναδρομική (recursive)* αν εξαρτάται από τον εαυτό της