

# Σχεσιακή Άλγεβρα Relational Algebra (RA)

# Σχεσιακή Άλγεβρα (RA)

- Άλγεβρα βασισμένη σε αυστηρά καθορισμένη σημασιολογία
- Όλα τα ερωτήματα τα μετατρέπει ο optimizer σε πλάνα σχεσιακής άλγεβρας
  - Μετά εφαρμόζει μετασχηματισμούς για βελτιστοποίηση
- Αποτελέσματα τελεστών της είναι σετ
  - Δεν υπάρχουν διπλότυπα
  - Σε αντίθεση με SQL
  - Σε αυτήν θα επικεντρωθούμε στις διαφάνειες
- Υπάρχει και Σχεσιακή Άλγεβρα πάνω σε bags (που επιτρέπει διπλότυπα – αυτή χρησιμοποιείται στην SQL)

# Όνομα Σχέσης

Το όνομα μίας σχέσης επιστρέφει ένα αντίγραφο της σχέσης

Student

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Select: Κριτήρια Περιορισμού

## Tuples Αποτελέσματος

$\sigma_{\text{ΣΥΝΘΗΚΕΣ}}$  RelationName ή  $\sigma_{\text{ΣΥΝΘΗΚΕΣ}}$  (Expression)

➤ Φοιτητές με μέσο όρο (GPA) > 3.5

$\sigma_{\text{GPA} > 3.5}$  (Student)

➤ Φοιτητές με μέσο όρο (GPA) > 3.5 και hs < 1000

$\sigma_{\text{GPA} > 3.5 \wedge \text{hs} < 1000}$  (Student)

➤ Αιτήσεις στο CS του TUC

$\sigma_{\text{cName} = \text{'TUC'} \wedge \text{major} = \text{'CS'}}$  (Apply)

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Project: Επιλογή Συγκεκριμένων Κολώνων (Attributes)

Γενική μορφή

$\pi_{A_1, A_2, \dots, A_n}$  RelationName ή  $\pi_{A_1, A_2, \dots, A_n}$  (Expression)

sID και απόφαση για κάθε αίτηση

$\pi_{sID, dec}$  (Apply)

Πόσες εγγραφές θα επιστραφούν για κάποιον που έκανε πολλές αιτήσεις;

sID	dec

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Συνδυασμός Project + Select

sID, sName μαθητών με GPA > 3.5

$$\pi_{sID, sName} (\sigma_{GPA > 3.5} (\text{Student}))$$

Ισοδύναμο SQL ερώτημα:

```
SELECT DISTINCT sID, sName  
FROM Student WHERE GPA > 3.5
```

Σημείωση: Το DISTINCT είναι μόνο για την άλγεβρα σε sets

Student

sID	sName	GPA	hs
1	John	3.7	1200
2	Mary	3.9	1700
3	Mary	3.6	700
4	Jim	3.4	700
5	Carl	3.8	900

Result

sID	sName
1	John
2	Mary
3	Mary
5	Carl

# Cross-Product / Cartesian Product

## Καρτεσιανό Γινόμενο: Συνδυασμός 2 σχέσεων R, S

- Πλειάδες παράγονται «κολλώντας» tuples της R με αυτά της S
- Αποτέλεσμα με τόσες κολώνες όσες το ΑΘΡΟΙΣΜΑ των κολώνων των R,S
- Τόσες εγγραφές όσες το ΓΙΝΟΜΕΝΟ του αριθμού εγγραφών των R,S

### Student x Apply

Παράδειγμα Καρτεσιανού γινομένου στην SQL:

```
SELECT * FROM Student, Apply
```

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Cross-Product / Cartesian Product

Student x Apply

Παρατηρήστε τη μετονομασία γνωρισμάτων με ίδιο όνομα

Student.sID	sName	GPA	hs	Apply.sID	cName	major	dec

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec



# Cross-Product - Παράδειγμα

Student x Apply

Student.sID	sName	GPA	hs	Apply.sID	cName	major	dec
1	John	3.7	1200	1	MIT	CS	NO
1	John	3.7	1200	2	EMP	CS	NO
1	John	3.7	1200	2	TUC	CS	YES
2	Mary	3.9	700	1	MIT	CS	NO
2	Mary	3.9	700	2	EMP	CS	NO
2	Mary	3.9	700	2	TUC	CS	YES

Student

sID	sName	GPA	hs
1	John	3.7	1200
2	Mary	3.9	700

Apply

sID	cName	major	dec
1	MIT	CS	NO
2	EMP	CS	NO
2	TUC	CS	YES

# Παράδειγμα

sName, GPA μαθητών με GPA > 3.5, με αίτηση στο CS που απορρίφθηκε

$\pi_{sName, GPA} (\sigma_{(Student \times Apply)})$

Result

sName	GPA
John	3.7
Mary	3.9

Student.sID = Apply.sID  
 $\wedge$  GPA > 3.5  
 $\wedge$  major = 'CS'  $\wedge$  dec = 'NO'

Student x Apply

Student.sID	sName	GPA	hs	Apply.sID	cName	major	dec
1	John	3.7	1200	1	MIT	CS	NO
1	John	3.7	1200	2	EMP	CS	NO
1	John	3.7	1200	2	TUC	CS	YES
2	Mary	3.9	700	1	MIT	CS	NO
2	Mary	3.9	700	2	EMP	CS	NO
2	Mary	3.9	700	2	TUC	CS	YES

# Natural Join / Φυσική Συνένωση

## Συνδυασμός 2 σχέσεων R, S

- Επιβάλλει ΙΣΟΤΗΤΑ σε γνωρίσματα με το ίδιο όνομα σε διαφορετικές σχέσεις
- Κρατάει μόνο 1 αντίγραφο από τα διπλά γνωρίσματα

sName, GPA μαθητών με hs > 1000, με αίτηση στο CS που απορρίφθηκε

$$\pi_{sName, GPA} \left( \sigma_{hs > 1000 \wedge major = 'CS'} (Student \bowtie Apply) \right) \wedge dec = 'NO'$$

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Natural Join - Παράδειγμα

$\pi_{sName, GPA} (\sigma_{hs > 1000 \wedge major = 'CS' \wedge dec = 'NO'} (Student \bowtie Apply))$

Student

sID	sName	GPA	hs
1	John	3.7	1200
2	Mary	3.9	700
3	Jim	3.4	1400

Apply

sID	cName	major	dec
1	MIT	CS	NO
1	TUC	CS	NO
3	EMP	CS	NO
3	TUC	MPD	YES
2	TUC	CS	NO

Student  $\bowtie$  Apply

sID	sName	GPA	hs	cName	major	dec
1	John	3.7	1200	MIT	CS	NO
1	John	3.7	1200	TUC	CS	NO
3	Jim	3.4	1400	EMP	CS	NO
3	Jim	3.4	1400	TUC	MPD	YES
2	Mary	3.9	700	TUC	CS	NO

Result

sName	GPA
John	3.7
Jim	3.4

# Natural Join / Φυσική Συνένωση

sName, GPA μαθητών με  $hs > 1000$ , με αίτηση στο CS σε κολλέγια με  $enr > 20000$  που απορρίφθηκε

$\pi_{sName, GPA} (\sigma_{hs > 1000 \wedge major = 'CS' \wedge dec = 'NO' \wedge enr > 20000} ((Student \bowtie Apply) \bowtie College))$

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# $\theta$ join & Equijoin

$Exp1 \bowtie_{\theta} Exp2$

$\theta$ : περιορισμός, παρόμοιος με ό,τι βάζουμε σε select

Οι περισσότεροι αναφέρονται στο  $\theta$  join απλώς ως join

Ισχύει ότι:

$$Exp1 \bowtie_{\theta} Exp2 \equiv \sigma_{\theta} (Exp1 \times Exp2)$$

**Equijoin:** Όταν το  $\theta$  περιέχει μόνο συνθήκες ισότητας

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Παράδειγμα $\Theta$ join

$\pi_{sName, GPA} ((Student \bowtie Apply) \bowtie College))$   
Apply.sID = Student.sID  
Apply.cName = College.cName  
 $\wedge hs > 1000$   
 $\wedge major = 'CS'$   
 $\wedge dec = 'NO'$

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Ένωση, Τομή, Διαφορά

Βάσει παραδοσιακών τελεστών σε σετ:

**Union:** Ένωση  $A \cup B$

- Το σετ από πλειάδες που ανήκουν στο  $A$  ή στο  $B$
- Για συμβατά ως προς την ένωση  $A$  και  $B$

**Difference:** Διαφορά  $A - B$

- Το σετ από πλειάδες που ανήκουν στο  $A$  αλλά όχι στο  $B$
- Για συμβατά ως προς τη διαφορά  $A$  και  $B$

**Intersection:** Τομή  $A \cap B$

- Το σετ από πλειάδες που ανήκουν και στο  $A$  και στο  $B$
- Για συμβατά ως προς την τομή  $A$  και  $B$
- $A \cap B \equiv A - (A - B)$

# Παράδειγμα: Union

Δώστε τα ονόματα κολλεγίων ή μαθητών

Σχεδόν σωστό...

$$\pi_{sName}(\text{Student}) \cup \pi_{cName}(\text{College})$$

Η ένωση απαιτεί το ίδιο σχήμα (ονόματα/τύπο attributes)  
Θα το λύσουμε αυτό αργότερα με τελεστή μετονομασίας

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Παράδειγμα: Difference

Δώστε τα sID των μαθητών που δεν έκαναν αίτηση  
 $\pi_{sID}(\text{Student}) - \pi_{sID}(\text{Apply})$

Πώς βρίσκουμε όλα τα στοιχεία των μαθητών που δεν  
έκαναν καμία αίτηση;

$$(\pi_{sID}(\text{Student}) - \pi_{sID}(\text{Apply})) \bowtie \text{Student}$$

Αποτέλεσμα: Εγγραφές της Student με μαθητές χωρίς αίτηση

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Παράδειγμα: Difference

Δώστε τα sID και τα ονόματα των μαθητών που δεν έκαναν  
καμία αίτηση

$$\pi_{sID, sName} \left( (\pi_{sID}(\text{Student}) - \pi_{sID}(\text{Apply})) \bowtie \text{Student} \right)$$

Αποτέλεσμα: Εγγραφές της Student με μαθητές χωρίς αίτηση

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Rename: Τελεστής Μετονομασίας

Πολλές μορφές...

- $\rho_{R(A_1, \dots, A_n)}(E)$  αλλαγή και ονόματος σχέσης και attributes
- $\rho_R(E)$  αλλαγή ονόματος σχέσης – ίδια attributes
- $\rho_{A_1, \dots, A_n}(E)$  αλλαγή ονόματος attributes μόνο
- $\rho_{\text{παλιό} \rightarrow \text{νέο}}(E)$  αλλαγή σε 1 μόνο attribute
- $\rho_{\text{παλιό}_1, \text{παλιό}_2 \rightarrow \text{νέο}_1, \text{νέο}_2}(E)$  αλλαγή σε πολλά attributes
- $\rho_{\text{παλιό}_{1,3} \rightarrow \text{νέο}_1, \text{νέο}_2}(E)$  αναφορά και με τη θέση ενός attribute προς μετονομασία

$$\rho_{\text{name}}(\pi_{\text{sName}}(\text{Student})) \cup \rho_{\text{name}}(\pi_{\text{cName}}(\text{College}))$$

# OUTER JOIN

Συμπληρώνουμε με null πλειάδες της μίας σχέσης που ΔΕ συνενώνονται με κάποια πλειάδα της άλλης σχέσης

□ **LEFT OUTER JOIN:**  $R \bowtie_{\text{LEFT}} S$  ή  $R \Join S$

Μόνο tuples της R συμπληρώνονται με null

Όλα τα tuples της R συμμετέχουν στο join

□ **RIGHT OUTER JOIN:**  $R \bowtie_{\text{RIGHT}} S$  ή  $R \Join S$

Μόνο tuples της S συμπληρώνονται με null

Όλα τα tuples της S συμμετέχουν στο join

□ **FULL OUTER JOIN:**  $R \bowtie_{\text{FULL}} S$  ή  $R \Join S$

Tuples της R και της S συμπληρώνονται με null

Όλα τα tuples της R,S συμμετέχουν στο join

# Παραδείγματα – LEFT OUTER JOIN

R

ename	street	city
C	T	H
R	U	C
S	R	D
W	S	S

S

sname	branch	salary
C	M	1500
R	E	1300
G	R	5300
W	D	1500

$R \bowtie_{\text{ename=sname}} S$

ename	street	city	sname	branch	salary
C	T	H	C	M	1500
R	U	C	R	E	1300
S	R	D	null	null	null
W	S	S	W	D	1500

# Παραδείγματα – FULL OUTER JOIN

R

ename	street	city
C	T	H
R	U	C
S	R	D
W	S	S

S

sname	branch	salary
C	M	1500
R	E	1300
G	R	5300
W	D	1500

$R \bowtie_{\text{ename=sname}} S$

ename	street	city	sname	branch	salary
C	T	H	C	M	1500
R	U	C	R	E	1300
S	R	D	null	null	null
W	S	S	W	D	1500
null	null	null	G	R	5300

# Division (Διαίρεση)

□  $R / S$  ή  $R \div S$

Έστω  $A1$ : τα attributes της  $R$

Έστω  $A2$ : τα attributes της  $S$ , με  $A2 \subset A1$

**$R / S$  περιέχει ως attributes τα  $A1 - A2$**

Περιέχεται κάθε tuple του  $A1-A2$  που συνδυάζεται στην  $R$  με ΌΛΑ τα tuples της  $S$

# Παράδειγμα Διαίρεσης

R

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

S

C	D
c	d
e	f

R / S

A	B
a	b
e	d

# Παράδειγμα Διαίρεσης

Ποιοι προμηθευτές προμηθεύουν όλα τα προϊόντα;

SUPPLIER\_PART /  $\pi_{P\#}$  SUPPLIER\_PART

SUPPLIER\_PART

S#	P#
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S4	P5
S1	P5

Τι επιστρέφει;  
Προσέξτε: 5 προϊόντα...

# Aggregate Functions

- Συναθροιστικές συναρτήσεις πάνω σε attribute A μιας σχέσης R

$\text{sum}_A(\mathbf{R}), \text{avg}_A(\mathbf{R}), \text{max}_A(\mathbf{R}), \text{min}_A(\mathbf{R}), \text{count}_A(\mathbf{R})$

Πολύ χρήσιμο συχνά και το  $\text{count-distinct}_A(\mathbf{R})$

- Συνάθροιση με ομαδοποίηση: Έστω  $F_i$  κάποια από τις παραπάνω συναθροιστικές συναρτήσεις

$G_1, G_2, \dots, G_n \text{ } \mathbf{G}_{F_1(A_1), F_2(A_2), \dots, F_m(A_m)}(\mathbf{R}) \quad \text{ή}$

$G_1, G_2, \dots, G_n \text{ } \mathbf{G}_{F_1(A_1), F_2(A_2), \dots, F_m(A_m)}(\mathbf{R}) \quad \text{ή}$

$G_1, G_2, \dots, G_n \text{ } \mathbf{G}_{F_1(A_1) \text{ as } Y_1, F_2(A_2) \text{ as } Y_2, \dots, F_m(A_m) \text{ as } Y_m}(\mathbf{R})$

Για κάθε συνδυασμό τιμών των  $G_1, G_2, \dots, G_n$  υπολόγισε τις συναθροιστικές συναρτήσεις  $F_1, \dots, F_m$

# Παραδείγματα Aggregate Functions

$G_{\text{avg}(\text{salary})}(\text{instructor}) \equiv \text{avg}_{\text{salary}}(\text{instructor})$

instructor

ID	name	dept_name	salary
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

avg_salary
74833

dept\_name  $G_{\text{avg}(\text{salary})}(\text{instructor})$

dept_name	avg_salary
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

# Σχεσιακή Άλγεβρα σε Bags

- Επιτρέπει διπλότυπα – Πχ,  $\{1,2,1,1,3\}$  είναι ένα bag
- Οι περισσότεροι τελεστές δουλεύουν
  - $\sigma, \rho$  δεν επηρεάζονται
  - $\pi$  δεν αφαιρεί διπλότυπα
  - Όλοι οι τελεστές συνόλων θέλουν αλλαγές
    - $\{1, 1, 2, 3\} \cup \{2, 2, 3, 4\} = \{1, 1, 2, 3, 2, 2, 3, 4\}$
    - $\{1, 1, 2, 3\} \cap \{2, 2, 3, 4\} = \{2, 3\}$ 

Εξήγηση: Το κάθε στοιχείο εμφανίζεται με το  $\min$  των εμφανίσεών του στα 2 σετ όπου εφαρμόζουμε την τομή
    - $\{1, 1, 2, 3\} - \{1, 2, 3, 4\} = \{1\}$ 

Εξήγηση: Για κάθε στοιχείο του πρώτου σετ, αφαιρούμε τον αριθμό εμφανίσεών του στο 2ο σετ

# Άλλα Σύμβολα

- Ομαδοποίηση:  $G, \Gamma, \gamma$   
Διάφοροι συμβολισμοί
- Αφαίρεση διπλότυπων:  $\delta$  (σε άλγεβρα πάνω σε bags)  
 $\pi_{sID}(\text{Apply})$  περιέχει διπλότυπα  
 $\delta(\pi_{sID}(\text{Apply}))$  δεν περιέχει διπλότυπα
- Ταξινόμηση:  $\tau$   
 $\tau_{sID}(\text{Apply})$   
ταξινόμηση του Apply κατά sID, αύξουσα σειρά  
 $\tau_{-sID}(\text{Apply})$  (προσέξτε το - )  
ταξινόμηση του Apply κατά sID, φθίνουσα σειρά

# Τι δεν Καλύψαμε;

- Τα subqueries τα εκτελούν με διάφορους (και διαφορετικούς) τρόπους τα ΣΔΒΔ
- Πολλά subqueries γίνονται flattened και μετατρέπονται σε joins
- Κάποια εκτελούνται ξεχωριστά, σπάζοντας το πλάνο εκτέλεσης
- Κάποια απαιτούν nested loops για να υπολογιστούν
- Περισσότερα στο μάθημα "Προχωρημένα Θέματα Βάσεων Δεδομένων"

# Γενικευμένη Προβολή – Generalized Projection

- Εμφανίζεται σε επεκτάσεις της σχεσιακής άλγεβρας
- Ίδιο σύμβολο ( $\pi$ ) με την προβολή
- Περιέχει πιθανώς αριθμητικές συναρτήσεις στη λίστα των attributes προς προβολή

$\pi_{\text{cardNum}, (\text{creditLimit} - \text{balance}) \text{ as available}} (\text{CreditCard})$

Αντίστοιχο του rename σε αυτόν τον τελεστή  
Προφανώς, μπορείτε να πετύχετε το ίδιο με rename – ΠΩΣ;

Credit Card

cardNum	owner	creditLimit	balance

# Ανάθεση και Χρήσεις

Μεγάλη έκφραση:

$$\pi_{sID, sName}((\pi_{sID}(\text{Student}) - \pi_{sID}(\text{Apply})) \bowtie \text{Student})$$

Με ανάθεση:

$$t1 \leftarrow \pi_{sID}(\text{Student}) - \pi_{sID}(\text{Apply})$$

$$\pi_{sID, sName}(t1 \bowtie \text{Student})$$

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Ανάθεση για Εισαγωγή/Διαγραφή

Μπορούμε να κάνουμε και ανάθεση σε μία σχέση  
Μοναδικός τρόπος αλλαγής των δεδομένων της

Εισαγωγή Tuples:  $\text{Student} \leftarrow \text{Student} \cup \text{Expression}$

Διαγραφή Tuples:  $\text{Student} \leftarrow \text{Student} - \text{Expression}$

Επίσης...  $\text{Student} \leftarrow \text{Expression}$

College

cName	state	enr

Student

sID	sName	GPA	hs

Apply

sID	cName	major	dec

# Ανάθεση για Ενημέρωση

Λίγο πιο σύνθετο.

Μπορούμε να θέλουμε αλλαγές σε όλες τις εγγραφές σε κάποιες στήλες, ή σε συγκεκριμένες εγγραφές μόνο

Δώσε 10% αύξηση σε όλους τους Employees

$\text{Employee} \leftarrow \pi_{\text{ID, Name, (1.1 * salary)}} \text{Employee}$

Δώσε 10% αύξηση σε όλους τους Employees με μισθό μικρότερο του 20000

$\text{Employee} \leftarrow \pi_{\text{ID, Name, (1.1 * salary)}} (\sigma_{\text{salary} < 20000} (\text{Employee}))$

$\cup \sigma_{\text{salary} \geq 20000} (\text{Employee})$

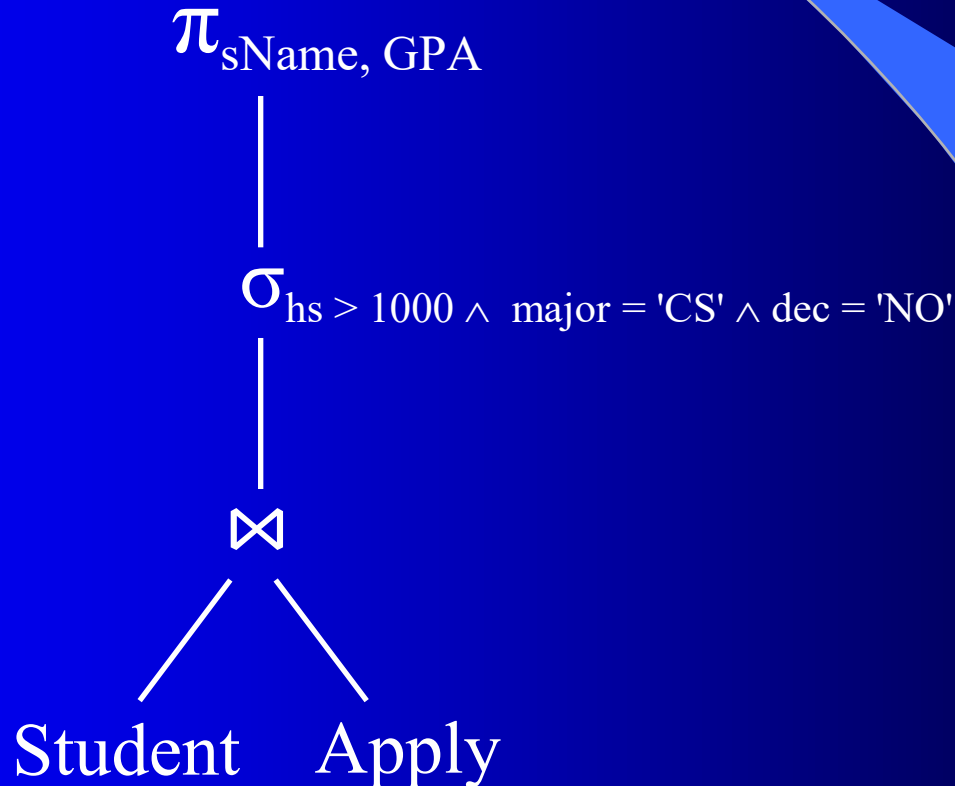
ID	Name	salary

Employee

# Expression Trees

Έκφραση ερωτήματος σε μορφή δέντρου...

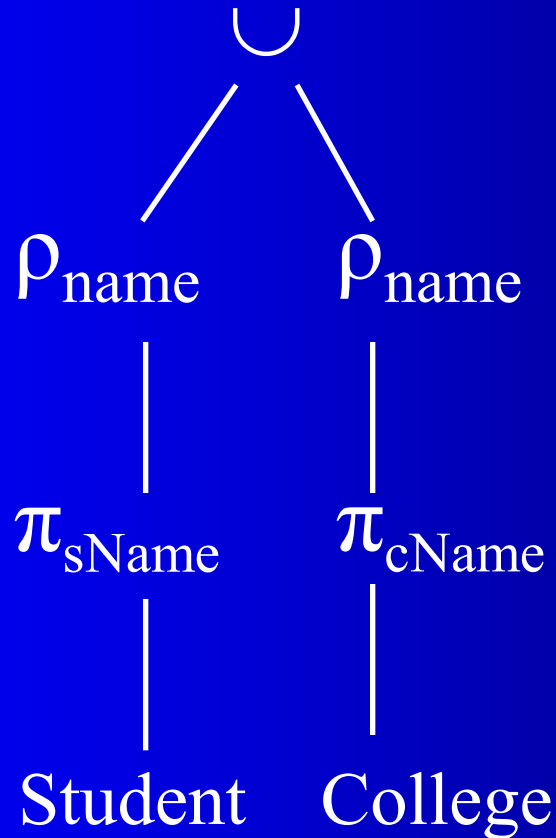
$\pi_{sName, GPA} (\sigma_{hs > 1000 \wedge major = 'CS' \wedge dec = 'NO'} (Student \bowtie Apply))$



# Expression Trees

Έκφραση ερωτήματος σε μορφή δέντρου...

$\rho_{\text{name}}(\pi_{\text{sName}}(\text{Student})) \cup \rho_{\text{name}}(\pi_{\text{cName}}(\text{College}))$



# Expression Trees

$\pi_{sName, GPA} ((Student \bowtie Apply) \bowtie College))$

Apply.sID =  
Student.sID  
 $\wedge$  hs > 1000  
 $\wedge$  major = 'CS'  
 $\wedge$  dec = 'NO'

Apply.cName =  
College.cName  
 $\wedge$  enr > 20000

