

Κεφάλαιο 3

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Το Σχεσιακό Μοντέλο – ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

- Άτυπος και Τυπικός Ορισμός του Μοντέλου
Δομές, Πράξεις / Λειτουργίες, Δομικοί Περιορισμοί
- Σχεσιακή Άλγεβρα
 - » Επέκταση της Σχεσιακής Άλγεβρας
 - » Παραδείγματα
- Όψεις – Κανόνες Ακεραιότητας
- Πράξεις Αλλαγών / Ενημερώσεων
- Σχεσιακός Λογισμός
 - » Παραδείγματα
- Σχεσιακή Πληρότητα

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Άτυπος Ορισμός

- Προτάθηκε το 1970 από τον E.F. Codd (“*A relational model for large shared data banks*”, CACM), σαν **Θεωρία** για Μοντέλα Δεδομένων
- Έδωσε έναυσμα για πολλές ερευνητικές προσπάθειες και κατέληξε να είναι το πλέον δημοφιλές μοντέλο. Σήμερα, η συντριπτική πλειοψηφία των DBMS είναι Σχεσιακά και διατίθενται σε ΟΛΕΣ τις Υπολογιστικές Πλατφόρμες.
- **Μια Σχεσιακή Βάση Δεδομένων είναι ένα σύνολο από ΣΧΕΣΕΙΣ**
- **ΣΧΕΣΗ (RELATION):** Ένας Πίνακας τιμών. Κάθε **στήλη** στον πίνακα έχει επικεφαλίδα (όνομα), που ονομάζεται **γνώρισμα (attribute -field)**. Κάθε **γραμμή** ονομάζεται μια **πλειάδα (Tuple)** και παριστά τα χαρακτηριστικά μιας οντότητας στο μοντέλο.

Τυπικός Ορισμός (Formal)

■ ΔΟΜΕΣ

- Μόνο ένα είδος:: *relations (σχέσεις)* (που έχουν όνομα)

Ένα *πεδίο τιμών (domain)* D είναι ένα σύνολο τιμών ,
 $D = \{d_1, d_2, \dots, d_n\}$

π.χ., DOMAIN OF NAMES = το σύνολο όλων των ονομάτων

DOMAIN of WEIGHT = το σύνολο όλων των τιμών για βάρη

CHAR STRINGS from 1 to 10 in length, κλπ.

Ένα *γνώρισμα (attribute)* A ονοματίζει μια ιδιότητα (με ενδιαφέρον)
σε μία Σχέση και παίρνει τιμές από το αντίστοιχο πεδίο τιμών
 $D(A)$.

π.χ., EMPLOYEE_NAME, WEIGHT, κλπ.

Δηλαδή, τα Γνωρίσματα είναι οι επικεφαλίδες σε μια Σχέση
(Συμβολισμός: **$R.A$** , ή **$R[A]$** όπου R είναι το όνομα της Σχέσης)

Ορισμοί Δομών (2)

Ένα **Σχεσιακό Σχήμα R** είναι το όνομα και τα γνωρίσματα σε μια Σχέση, μαζί με τα αντίστοιχα πεδία τιμών για τα γνωρίσματα. Όταν είναι προφανές, τα πεδία τιμών παραλείπονται.

Συμβολισμός: **$R(A_1, A_2, \dots, A_n)$**

π.χ., **$STUDENT(Name, SSN, BirthDate, Address)$**

Ο **βαθμός (degree) n** μιας Σχέσης R είναι ο αριθμός των γνωρισμάτων στην Σχέση

Ένα **Σχήμα Βάσης Δεδομένων S** είναι ένα σύνολο Σχέσεων.

Συμβολισμός: **$S = \{R_1, R_2, \dots, R_m\}$**

π.χ., **$COMPANY = \{EMPLOYEE, PROJECT, \dots\}$**

Ορισμοί Δομών (3)

- Μια **πλειάδα (Tuple)** t μιας Σχέσης $R(A_1, A_2, \dots, A_n)$ είναι μια (διατεταγμένη) λίστα τιμών $t = \langle v_1, v_2, \dots, v_n \rangle$, όπου κάθε τιμή v_i είναι ένα στοιχείο του πεδίου $D(A_i)$.
- Ένα **στιγμιότυπο σχέσης (relation instance)** $r(R)$, πιο απλά, **σχέση (relation)**, είναι ένα σύνολο πλειάδων
$$r(R) = \{ t_1, t_2, \dots, t_k \}$$
εναλλακτικά, είναι ένα υποσύνολο του Καρτεσιανού Γινομένου
$$r(R) \subset D(A_1) \times D(A_2) \times \dots \times D(A_n)$$
- Η **πληθικότητα (cardinality)** της R είναι ο αριθμός των πλειάδων στην $r(R)$, και συμβολίζεται με $CARD_R$
- Μια **σχεσιακή βάση (relational database)** είναι ένα σύνολο σχέσεων

Χαρακτηριστικά των Σχέσεων

- Η ΔΙΑΤΑΞΗ των γνωρισμάτων σε μια σχέση είναι σημαντική
- Η ΔΙΑΤΑΞΗ των πλειάδων σε μια σχέση δεν είναι σημαντική
- Κάθε πλειάδα αποθηκεύεται ΜΙΑ ΦΟΡΑ σε μια σχέση (σύνολο)
- Μια τιμή μπορεί να παρουσιάζεται ΠΟΛΛΕΣ ΦΟΡΕΣ σε μια στήλη και είναι ΑΤΟΜΙΚΗ (μη διασπάσιμη) – συχνά αυτό αναφέρεται σαν First Normal Form (1-NF) σχέση
- Μια ειδική τιμή, ονομαζόμενη ΚΕΝΟ (NULL), χρησιμοποιείται για την να παραστήσει στη βάση μιας τιμή που είναι **μη εφαρμόσιμος** (non-applicable) ή **άγνωστη** (unknown).
Π.χ, η τιμή για τον ΑριθμόΤηλεφώνου κάποιου που δεν έχει τηλέφωνο,
Η τιμή για την Διεύθυνση κάποιου που δεν έχει προσκομίσει διεύθυνση
- Συμβολισμός: *τιμή γνωρίσματος για μια πλειάδα* t , $t[A_i] = v_i$

ΔΟΜΙΚΟΙ ΠΕΡΙΟΡΙΣΜΟΙ

■ Δομικοί Περιορισμοί

- Υπάρχουν τρία είδη περιορισμών *που είναι έμφυτοι στο μοντέλο*: **ΚΛΕΙΔΙ (KEY)**, **ΑΚΕΡΑΙΟΤΗΤΑ ΟΝΤΟΤΗΤΑΣ (ENTITY INTEGRITY)**, και **ΑΝΑΦΟΡΙΚΗ ΑΚΕΡΑΙΟΤΗΤΑ (REFERENTIAL INTEGRITY.)**
- Υπάρχουν τρία είδη *ρητών περιορισμών*: **ΠΕΔΙΟ ΤΙΜΩΝ (DOMAIN)**, **ΣΤΗΛΩΝ (COLUMN)** και **ΟΡΙΖΟΜΕΝΩΝ** από τον **ΧΡΗΣΤΗ (USER-DEFINED)** – κάποιοι άλλοι ρητοί περιορισμοί, π.χ., οι Συναρτησιακές Εξαρτήσεις (*Functional Dependencies*,) θα αναφερθούν αργότερα

■ Περιορισμοί Κλειδιών: Τα διαφορετικά κλειδιά, όπως ορίστηκαν στο μοντέλο E-R, ισχύουν και στο Σχεσιακό Μοντέλο.

- Υπενθύμιση ότι το **ΚΛΕΙΔΙ** είναι ιδιότητα του Σχεσιακού Σχήματος **δεν είναι ιδιότητα του στιγμιότυπου της Σχέσης** (δηλαδή, ισχύει για **ΌΛΑ** τα στιγμιότυπα - σχέσεις)

Έμφυτοι Δομικοί Περιορισμοί (1)

- Ένα σύνολο γνωρισμάτων **SK** σχήματος σχέσης **R** για το οποίο κάθε πλειάδα σε στιγμιότυπο $r(R)$ πρέπει να έχει *μοναδιαία τιμή (τιμές)* είναι ένα **υπέρ-κλειδί (superkey)**. Δηλαδή, για διαφορετικά t_1 και t_2 , ισχύει $t_1[SK] \neq t_2[SK]$

Για παράδειγμα, SSN στο EMPLOYEE, NAME και ADDRESS στο EMPLOYEE, SSN και NAME στο EMPLOYEE, κλπ.

- Ένα **υποψήφιο κλειδί (candidate key)** **K** είναι ένα *ελάχιστο* υπέρ-κλειδί (δηλαδή, δεν υπάρχει υποσύνολο του **K** που να είναι και αυτό υπέρ-κλειδί). Το **K** ονομάζεται συνήθως **ΚΛΕΙΔΙ (key)**.

Για παράδειγμα, SSN είναι υποψήφιο κλειδί στο EMPLOYEE, αλλά ο συνδυασμός {SSN, NAME} ΔΕΝ είναι.

- Ένα **κύριο κλειδί (primary key)** **PK** είναι ένα από τα υποψήφια κλειδιά που συμφωνείται να παίξει το ρόλο του του προσδιοριστή για τις πλειάδες της σχέσης (τα κύρια κλειδιά **υπογραμμίζονται**)

Για παράδειγμα, SSN είναι το κύριο κλειδί της σχέσης EMPLOYEE.

Έμφυτοι Δομικοί Περιορισμοί (2)

- **ΑΚΕΡΑΙΟΤΗΤΑ ΟΝΤΟΤΗΤΑΣ (ENTITY INTEGRITY) :**
Το κύριο κλειδί PK στο σχήμα της σχέσης R δεν μπορεί να έχει ΚΕΝΕΣ (NULL) τιμές σε πλειάδες μιας σχέσης $r(R)$.

$t[PK] \neq \text{NULL}$, για κάθε t στην $r(R)$

- Ο λόγος πίσω από αυτό τον περιορισμό είναι ότι το κύριο κλειδί χρησιμοποιείται για τον προσδιορισμό μιας πλειάδας σε μια Σχέση.
- Σημειώνεται ότι και άλλα γνωρίσματα στην R μπορεί να περιορίζονται στο να μην έχουν NULL από ΡΗΤΟΥΣ περιορισμούς.

Έμφυτοι Δομικοί Περιορισμοί (3)

■ ΑΝΑΦΟΡΙΚΗ ΑΚΕΡΑΙΟΤΗΤΑ -REFERENTIAL INTEGRITY

Αυτός ο δομικός περιορισμός εμπλέκει ΔΥΟ σχέσεις και χρησιμοποιείται για να καταγράψει τη συνέπεια σε μια συσχέτιση μεταξύ πλειάδων των δυο σχέσεων.

■ Η πλέον συνήθης μορφή είναι αυτή των *εξωτερικών κλειδιών*.

- Ένα εξωτερικό κλειδί (foreign key) **FK** είναι ένα σύνολο γνωρισμάτων σε μια σχέση R1 που αποτελεί **κύριο κλειδί** σε μια άλλη σχέση R2.

Μια πλειάδα t_1 στην $r(R_1)$ λέγεται ότι *αναφέρεται* σε μια άλλη πλειάδα t_2 στην $r(R_2)$, εάν: $t_1[FK] = t_2[FK]$

Για παράδειγμα, για την σχέση WORKING-ON το γνώρισμα SSN είναι *foreign key* (είναι το κύριο κλειδί στην σχέση EMPLOYEE).

Ρητοί Περιορισμοί (1)

- **Περιορισμοί Πεδίου Τιμών:** Είναι οι κανόνες που ορίζονται για το πεδίο τιμών και κληρονομούνται από τις στήλες (γνωρίσματα) που παίρνουν τιμές από το πεδίο.
 - Το πεδίο μπορεί να οριστεί μαζί με κανόνες ακεραιότητας (π.χ., το πεδίο των *integers* με όλους τους κανόνες για ακέραιους). Αυτοί είναι (κυρίως) οι βασικοί **τύποι δεδομένων** (*data types*.)
 - Η καλύτερη περίπτωση είναι να έχουμε υποστήριξη για **strong data typing** (σπάνιο)
- **Περιορισμοί Στηλών:** Είναι *επιπρόσθετοι* των περιορισμών πεδίου τιμών και αναφέρονται στις τιμές για τα γνωρίσματα.
 - Για παράδειγμα., η στήλη των *small integers* ή *integers between 1 and 10*, κλπ. είναι επιπλέον περιορισμοί των ακεραίων.)
 - Σε μερικά σχεσιακά συστήματα, η υποστήριξη παρέχεται με ένα μηχανισμό που ονομάζεται **CHECK option**

Ρητοί Περιορισμοί (2)

- **ΠΕΡΙΟΡΙΣΜΟΙ ΟΡΙΖΟΜΕΝΟΙ από τον ΧΡΗΣΤΗ (USER-DEFINED) :** Κάθε περιορισμός ακεραιότητας, πέραν αυτών που έχουν ήδη αναφερθεί, καλείται *user-defined*.
- Για την υποστήριξη επιχειρηματικών κανόνων, απαιτούνται περιορισμοί ακεραιότητας με σημαντική πολυπλοκότητα
- Αυτοί προσδιορίζονται είτε *διαδικαστικά* ή *δηλωτικά* (με προτίμηση)
- Μια σειρά μηχανισμών χρησιμοποιούνται για την υποστήριξη τέτοιων κανόνων σε ένα σχεσιακό σύστημα:
 - *stored procedures, triggers, methods* (για *object-oriented systems*)
- Σημαντική ομάδα περιορισμών είναι οι *περιορισμοί σημασιολογικής ακεραιότητας (semantic integrity constraints)*
- **Γενικά, τα DBMS είναι αδύνατα σε υποστήριξη περιορισμών**

Η Βάση Δεδομένων COMPANY στο Σχεσιακό Μοντέλο - ΣΧΗΜΑ

EMPLOYEE (SSN, Name, BirthDate, Address, Sex, Salary, SupSSN, DNumber)

DEPARTMENT (DNumber, DName, MgrSSN, MgrStartDate)

PROJECT (PNumber, PName, Location, DNumber)

DEPT_LOCATION (DNumber, DLocation)

WORKS_ON (SSN, PNumber, HoursPW)

DEPENDENT (SSN, DependName, Sex, BirthDate, Relationship)

Η Βάση Δεδομένων COMPANY στο Σχεσιακό Μοντέλο - ΣΤΙΓΜΙΟΤΥΠΟ

EMPLOYEE

SSN	Name	BDate	Address	Sex	Salary	SupSSN	DNumber
1234	john	9.1.55	kifisia	m	30000	3344	5
3344	frank	8.9.45	athina	m	55000	8886	5
9998	alice	7.6.50	ekali	f	25000	9876	4
9876	jenny	2.6.41	patra	f	43000	8886	4
6668	rama	5.8.56	korinth	m	38000	3334	5
4534	joyce	3.7.62	kiato	f	25000	3334	5
9879	jack	2.3.59	maroussi	m	25000	9876	4
8886	james	1.1.40	psihico	m	60000	NULL	1

DEPARTMENT

DNumber	DName	MgrSSN	MgrSD
5	research	3334	22.3.78
4	admin	9876	14.1.85
1	headqrtr	8886	28.6.73

DEPT_LOCATION

DNumber	DLocation
1	kolonaki
4	zografou
5	gkizi
5	patisia
5	kolonaki

Η Βάση Δεδομένων COMPANY στο Σχεσιακό Μοντέλο - ΣΤΙΓΜΙΟΤΥΠΟ

WORKS_ON

SSN	PNumber	HoursPW
1234	1	32
1234	2	7
3344	2	10
3344	3	10
3344	10	10
3344	20	10
9998	30	30
9998	10	10
9876	30	20
9876	20	15
6668	3	40
4534	1	20
4534	2	20
9879	10	35
9879	30	5
8886	20	NULL

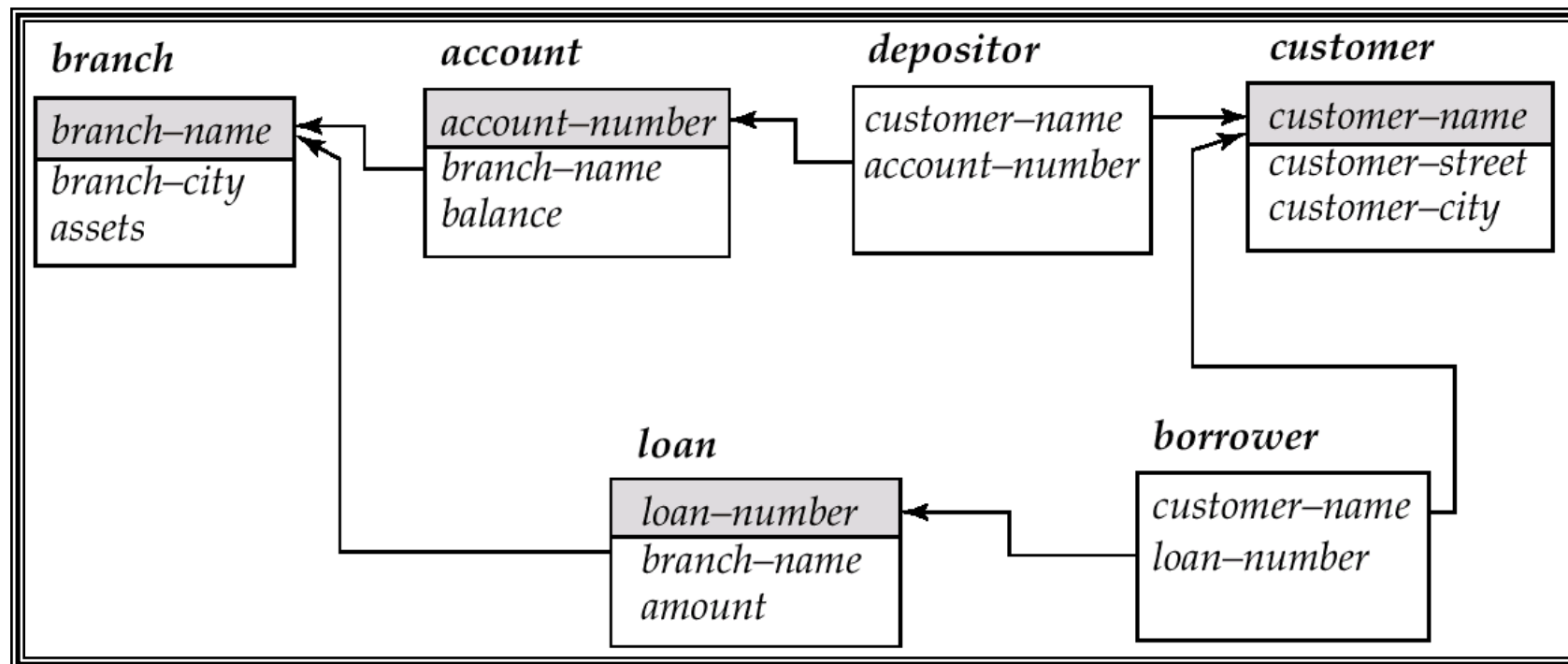
PROJECT

PNumber	PName	Location	DNumber
1	prodx	gkizi	5
2	prody	patisia	5
3	prodz	kolonaki	5
10	computer	zografou	4
20	reorganize	kolonaki	1
30	benefits	zografou	4

DEPENDENT

ESSN	Name	Sex	BDate	Relation
3334	aliki	f	1976	daught
3334	theo	m	1973	son
3334	joy	f	1948	spouse
9876	abner	m	1932	spouse
1234	mike	m	1978	son
1234	aliki	f	1979	daught
1234	elisa	f	1957	spouse

ΔΙΑΓΡΑΜΜΑ ΣΧΗΜΑΤΟΣ για το ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ



Ορισμός Σχεσιακού Μοντέλου: ΠΡΑΞΕΙΣ

■ ΠΡΑΞΕΙΣ / ΛΕΙΤΟΥΡΓΙΕΣ

- Διαχωρίζονται σε (α) ΕΝΗΜΕΡΩΣΕΙΣ, (β) ΑΝΑΚΤΗΣΕΙΣ
- Το σύνολο των πράξεων στο Σχεσιακό Μοντέλο είναι **ΚΛΕΙΣΤΟ** δηλαδή οι πράξεις ορίζονται σε Σχέσεις και έχουν αποτέλεσμα νέες Σχέσεις
- **Ενημερώσεις (UPDATE) σε Σχέσεις**
 - Εισαγωγή (INSERT) πλειάδας
 - Διαγραφή (DELETE) πλειάδας
 - Τροποποίηση (MODIFY) πλειάδας
- Οι περιορισμοί ακεραιότητας δεν πρέπει να παραβιάζονται με την εκτέλεση μιας πράξης ενημέρωσης. Για αυτό, ενημερώσεις μπορεί είτε να **απορρίπτονται ή να διορθώνουν με την εκτέλεση νέων (επιπλέον) ενημερώσεων.**
 - Π.χ., όταν μια πλειάδα του EMPLOYEE διαγράφεται, όλες οι πλειάδες στην WORKING_ON που έχουν την ίδια τιμή στο SSN διαγράφονται (ανύπαρκτοι υπάλληλοι δεν δουλεύουν σε έργα!)

Πράξεις: Σχεσιακές Γλώσσες

- **Πράξεις ΑΝΑΚΤΗΣΗΣ** στο Σχεσιακό Μοντέλο
 - Υπάρχουν δύο βασικές ομάδες πράξεων, που βασίζονται σε:
 - (α) **ΣΧΕΣΙΑΚΗ ΑΛΓΕΒΡΑ (RELATIONAL ALGEBRA)** – διαδικαστικού τύπου, προσδιορίζει πως υπολογίζεται το αποτέλεσμα
 - (β) **ΣΧΕΣΙΑΚΟΣ ΛΟΓΙΣΜΟΣ (RELATIONAL CALCULUS)** – δηλωτικού χαρακτήρα, προσδιορίζει ποιες ιδιότητες πρέπει να πληροί το αποτέλεσμα
- Κανένα σχεσιακό DBMS δεν υποστηρίζει τις δύο κατηγορίες γλωσσών στην **πλήρη θεωρητική** μορφή των. Αυτό συμβαίνει γιατί παράγοντες όπως “ευκολία χρήσης”, “διευκόλυνση”, κλπ., είναι σημαντικοί στην χρήση. Όμως, οι γλώσσες που υποστηρίζονται σε DBMS έχουν τις ρίζες τους είτε στην Σχεσιακή Άλγεβρα ή στον Σχεσιακό Λογισμό.

Σχεσιακές Γλώσσες

- Query languages: Επιτρέπουν την *επεξεργασία* και *ανάκτηση* δεδομένων σε μια Βάση Δεδομένων
- Το Σχεσιακό Μοντέλο υποστηρίζει απλές και εκφραστικές Γλώσσες Ανάκτησης (QLs):
 - Ισχυρή τυπική / θεωρητική βάση (Μαθηματική Λογική).
 - Επιτρέπουν μεγάλο βαθμό βελτιστοποιήσεις.
- Query Languages **ΔΕΝ ΕΙΝΑΙ** programming languages!
 - Οι QLs δεν αναμένονται να είναι “Turing complete”.
 - Οι QLs δεν χρησιμοποιούνται για σύνθετους υπολογισμούς.
 - Οι QLs υποστηρίζουν εύκολη και αποδοτική πρόσβαση σε μεγάλες Βάσεις Δεδομένων.

Πράξεις στη Σχεσιακή Άλγεβρα

ΣΧΕΣΙΑΚΗ ΑΛΓΕΒΡΑ

- Ένα σύνολο τελεστών που ενεργούν σε Σχέσεις και έχουν σαν αποτέλεσμα Σχέσεις (η Άλγεβρα είναι κλειστή).
- Οι πράξεις μπορούν να ΣΥΝΔΥΑΣΤΟΥΝ δημιουργώντας «εκφράσεις» στη Σχεσιακή Άλγεβρα (ο τυπικός ορισμός ακολουθεί)
- Οι τελεστές, όπως και στην αριθμητική άλγεβρα, μπορεί να φωλιάσουν (nested), εφόσον το αποτέλεσμα κάθε πράξης είναι μια Σχέση
- Υπάρχουν δύο είδη τελεστών / πράξεων:
 - **κλασσικές (regular) συνολοθεωρητικές**
 - » ένωση, τομή, διαφορά, ...
 - **εξειδικευμένες για βάσεις δεδομένων**
 - » προβολή, επιλογή, συνένωση, μετονομασία ...

Σχεσιακή Άλγεβρα : Συνολοθεωρητικές Πράξεις (1)

■ Κλασσικές Συνολοθεωρητικές Πράξεις

– *Union, Intersection, Difference, Cartesian Product*

– Για την εφαρμογή των πρώτων τριών πράξεων, πρέπει να έχουμε την ιδιότητα της ΣΥΜΒΑΤΟΤΗΤΑΣ ως προς την ΕΝΩΣΗ (UNION COMPATIBILITY) μεταξύ των δυο Σχέσεων
Δηλαδή:

$R_1 (A_1 , A_2 , \dots , A_n)$ και $R_2 (B_1 , B_2 , \dots , B_n)$ πρέπει να έχουν τον **ίδιο αριθμό** γνωρισμάτων και τα πεδία τιμών που αντιστοιχούν να είναι συμβατά, δηλαδή,

$$D(A_i) = D(B_i) , \quad \text{για} \quad i = 1, 2, \dots, n$$

Σχεσιακή Άλγεβρα: Παράδειγμα ΒΔ

Σχήμα Βάσης Δεδομένων

STUDENT (SName, SAge) ,

σχήμα R

INSTRUCTOR (IName, IAge)

σχήμα S

Στιγμιότυπο Βάσης Δεδομένων

R (STUDENT)

S (INSTRUCTOR)

SName	SAge
mary	20
jack	22
paul	25
barb	22
john	24

IName	IAge
helen	22
paul	25
barb	22
kris	24

$CARD_R = 5$

$CARD_S = 4$

Σχεσιακή Άλγεβρα : Συνολοθεωρητικές Πράξεις (2)

- **ΕΝΩΣΗ (UNION)** – Βάλε όλες τις πλειάδες των δύο Σχέσεων σε μια Σχέση

– **Συμβολισμός:** $R \cup S$ $CARD_{R \cup S} \leq CARD_R + CARD_S$

– **Τυπικά:** $R \cup S = \{ t \mid t \text{ is in } R \text{ or } t \text{ is in } S \}$

– **Παράδειγμα:** $STUDENT \cup INSTRUCTOR$

SName	SAge
mary	20
jack	22
paul	25
barb	22
john	24

$\cup$

IName	IAge
helen	22
paul	25
barb	22
kris	24

=

SName	SAge
mary	20
jack	22
paul	25
barb	22
john	24
helen	22
kris	24

Σχεσιακή Άλγεβρα : Συνολοθεωρητικές Πράξεις (3)

- **ΤΟΜΗ** - Βάλε όλες τις κοινές πλειάδες των δύο Σχέσεων σε μια Σχέση

– **Συμβολισμός:** $R \cap S$ $CARD_{R \cap S} \leq \max(CARD_R, CARD_S)$

– **Τυπικά:** $R \cap S = \{t \mid t \text{ is in } R \text{ and } t \text{ is in } S\}$

– **Παράδειγμα :** $STUDENT \cap INSTRUCTOR$

SName	SAge
mary	20
jack	22
paul	25
barb	22
john	24

$\cap$

IName	IAge
helen	22
paul	25
barb	22
kris	24

=

SName	SAge
paul	25
barb	22

Σχεσιακή Άλγεβρα : Συνολοθεωρητικές Πράξεις (4)

- **ΔΙΑΦΟΡΑ** – Επέλεξε τις πλειάδες της πρώτης Σχέσης, που δεν είναι στοιχεία της δεύτερης Σχέσης
 - **Συμβολισμός:** $R - S$
 - **Τυπικά:** $R - S = \{t \mid t \text{ is in } R \text{ and } t \text{ is not in } S\}$
 - **Παράδειγμα:** STUDENT – INSTRUCTOR

SName	SAge
mary	20
jack	22
paul	25
barb	22
john	24

–

IName	IAge
helen	22
paul	25
barb	22
kris	24

=

SName	SAge
mary	20
jack	22
john	24

$$\text{CARD}_{R-S} \leq \text{CARD}_R$$

Σχεσιακή Άλγεβρα : Συνολοθεωρητικές Πράξεις (5)

- **ΚΑΡΤΕΣΙΑΝΟ ΓΙΝΟΜΕΝΟ** – Συνδύασε τις πλειάδες της μιας σχέσης με κάθε πλειάδα της άλλης
 - **Συμβολισμός** : $R \times S$
 - **Τυπικά**: $R \times S = \{ t \mid t \text{ is the concatenation of a Tuple in } R \text{ with a Tuple in } S \}$
 - **Παράδειγμα** : $\text{STUDENT} \times \text{INSTRUCTOR}$

$$\text{CARD}_{R \times S} = \text{CARD}_R \times \text{CARD}_S$$

SName	SAge	Iname	IAge
mary	20	helen	22
mary	20	paul	25
mary	20	barb	22
mary	20	kris	24
jack	22	helen	22
jack	22	paul	25
jack	22	barb	22
jack	22	kris	24
paul	25	helen	22

paul	25	paul	25
paul	25	barb	22
paul	25	kris	24
barb	22	helen	22
barb	22	paul	25
barb	22	barb	22
barb	22	kris	24
john	24	helen	22
john	24	paul	25
john	24	barb	22
john	24	kris	24

Σχεσιακή Άλγεβρα: ΕΠΙΛΟΓΗ (SELECTION)

- **ΕΠΙΛΟΓΗ** – Επέλεξε τις πλειάδες μιας σχέσης που πληρούν κάποια συνθήκη (*qualification*) c , η οποία είναι μια λογική παράσταση (Boolean Expression) στα γνωρίσματα της Σχέσης (“οριζόντιο” υποσύνολο της R)
 - **Συμβολισμός** : $\sigma_c(R)$ ή $R[c]$
 - **Τυπικά**: $\sigma_c(R) = \{ t \mid t \text{ is in } r(R) \text{ and condition } c \text{ holds for } t \}$
 - **Παραδείγματα** : $\sigma_{\text{DNumber} = 4}(\text{EMPLOYEE})$, $\sigma_{\text{Salary} > 30000}(\text{EMPLOYEE})$
 $\sigma_{(\text{Salary} > 30000 \text{ AND } \text{DNumber} = 4) \text{ OR } \text{DNumber} = 5}(\text{EMPLOYEE})$,
 $\text{EMPLOYEE} [\text{Dnumber} = 4], \text{EMPLOYEE} [\text{Salary} > 30000]$

Σχεσιακή Άλγεβρα: ΕΠΙΛΟΓΗ (2)

– Η ΕΠΙΛΟΓΗ είναι *commutative* και *associative*

- (a) $\sigma_{c1} (\sigma_{c2} (R)) = \sigma_{c2} (\sigma_{c1} (R))$
- (b) $\sigma_{c1} (\sigma_{c2} (R)) = \sigma_{c1 \text{ AND } c2} (R) = \sigma_{c1, c2} (R)$
- (c) $\sigma_{c1} (\sigma_{c2} (\sigma_{c3} (R))) = \sigma_{c2} (\sigma_{c3} (\sigma_{c1} (R)))$

– **Παράδειγμα :** $\sigma_{\text{DNumber} = 4} (\text{EMPLOYEE})$

Όλοι οι υπάλληλοι στο τμήμα με αριθμό 4

SSN	Name	BDate	Address	Sex	Salary	SupSSN	DNumber
9998	alice	7.6.50	ekali	f	25000	9876	4
9876	jenny	2.6.41	patra	f	43000	8886	4
9879	jack	2.3.59	maroussi	m	25000	9876	4

Σχεσιακή Άλγεβρα: ΠΡΟΒΟΛΗ (PROJECTION)

- **ΠΡΟΒΟΛΗ** – Κρατά στο αποτέλεσμα μόνο μερικά γνωρίσματα (που προσδιορίζονται από μια λίστα L) και διαγράφει τα άλλα γνωρίσματα της R όπως επίσης τις διπλές πλειάδες (“κάθετο” υποσύνολο της R)
 - **Συμβολισμός** : $\pi_L(R)$ ή $R[L]$
 - **Τυπικά**: $\pi_L(R) = \{ t[L] \mid t \text{ is in } r(R) \text{ and } L \subset R \}$
 - **Παράδειγμα** : $\pi_{\text{Location}}(\text{PROJECT})$, or $\text{PROJECT}[\text{Location}]$

Location
gkizi
patisia
kolonaki
zografou

Όλες οι τοποθεσίες όπου εκτελούνται έργα

Σχεσιακή Άλγεβρα: METONOMASIA (Rename)

- **METONOMASIA:** Μας επιτρέπει να ΟΝΟΜΑΖΟΥΜΕ, και κατά συνέπεια να μπορούμε να αναφερόμαστε σε αυτό, το αποτέλεσμα μιας έκφρασης στη Σχεσιακή Άλγεβρα. Επίσης μας επιτρέπει να αναφερόμαστε σε μία Σχέση με περισσότερα του ενός ονόματα.

Παράδειγμα: Το $\rho_X(E)$

επιστρέφει το αποτέλεσμα της έκφρασης E με το όνομα X

Αν η έκφραση E είναι βαθμού n , τότε η

$$\rho_X(A_1, A_2, \dots, A_n)(E)$$

επιστρέφει το αποτέλεσμα της E με το όνομα X , και με τα γνωρίσματα να έχουν μετονομαστεί σε $A_1, A_2, \dots, A_n$.

Σχεσιακή Άλγεβρα: ΣΥΝΕΝΩΣΗ (JOIN)

- Υπάρχουν πολλές μορφές Συνένωσης – όλες συνδυάζουν δύο σχέσεις για την δημιουργία μιας τρίτης
 - *(theta) join, equality join, natural join, semi-join, outer join*
- **Θήτα - Συνένωση:** Συνδύασε πλειάδες δύο Σχέσεων που ταιριάζουν (πληρούν κάποια Boolean συνθήκη **c**) σε κάποια προσδιοριζόμενα γνωρίσματα
 - Μια Θήτα-Συνένωση είναι ισοδύναμη ενός Καρτεσιανού γινομένου που ακολουθείται από μια επιλογή με συνθήκη **c**.
 - **Συμβολισμός:** $R \bowtie_c S$ ή $R [c] S$
 - Η Σχέση – αποτέλεσμα έχει ΟΛΑ τα γνωρίσματα της R και της S

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

Σχεσιακή Άλγεβρα: Θ-ΣΥΝΕΝΩΣΗ (THETA JOIN)

– **Παράδειγμα** : DEPARTMENT $\bowtie_{\text{MgrSSN} > \text{MgrSSN}}$ DEPARTMENT

Όλοι οι συνδυασμοί ΤΜΗΜΑ-ΤΜΗΜΑ όπου το νούμερο του πρώτου τμήματος είναι μεγαλύτερο από αυτό του δεύτερου

DNumber	DName	MgrSSN	MgrSD
4	admin	9876	14.1.85
4	admin	9876	14.1.85
1	headqrtr	8886	28.6.73

DNumber	DName	MgrSSN	MgrSD
1	headqrtr	8886	28.6.73
5	research	3334	22.3.78
5	research	3334	22.3.78

Σχεσιακή Άλγεβρα: ΣΥΝΕΝΩΣΗ (EQUALITY JOIN)

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

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

– **Παράδειγμα:** $WORKS_ON \bowtie_{HoursPW = DNumber} PROJECT$

SSN	PNumber	HoursPW
9879	30	5
9879	30	5
9879	30	5

PNumber	PName	Location	DNumber
1	prodx	gkizi	5
2	prody	patisia	5
3	prodz	kolonaki	5

Μια Συνένωση ΧΩΡΙΣ ΝΟΗΜΑ !!!

Σχεσιακή Άλγεβρα: ΦΥΣΙΚΗ ΣΥΝΕΝΩΣΗ (NATURAL JOIN)

- **ΦΥΣΙΚΗ ΣΥΝΕΝΩΣΗ:** 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 $R[X=X]S$
 - **Παράδειγμα :** $DEPARTMENT \bowtie DEPT_LOCATION$

DNumber	DName	MgrSSN	MgrSD	DLocation
5	research	3334	22.3.78	gkizi
5	research	3334	22.3.78	patisia
5	research	3334	22.3.78	kolonaki
4	admin	9876	14.1.85	zografou
1	headqrtr	8886	28.6.73	kolonaki

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:

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

- 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
α	α	7
α	β	7
β	β	3
β	β	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 $g_{sum(balance)}$ (*account*)

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

Σχεσιακή Άλγεβρα: ΗΜΙ-ΣΥΝΕΝΩΣΗ (SEMI—JOIN)

- **SEMI-JOIN:** Επέλεξε το υποσύνολο μιας Σχέσης που συνενώνεται με μια άλλη. Μια semi-join είναι ισοδύναμη με μια συνένωση ακολουθούμενη από μια προβολή.
 - **Συμβολισμός:** $R \propto_c S$ ή $R <_c S$
 - **Παράδειγμα:** $\text{EMPLOYEE} \propto_{\text{SSN=MgrSSN}} \text{DEPARTMENT}$

SSN	Name	BDate	Address	Sex	Salary	SupSSN	DNumber
3344	frank	8.9.45	athina	m	55000	8886	5
9876	jenny	2.6.41	patra	f	43000	8886	4
8886	james	1.1.40	psihico	m	60000	NULL	1

Οι ημι-συνενώσεις είναι ΧΡΗΣΙΜΕΣ σε κατανεμημένες Βάσεις Δεδομένων

Σχεσιακή Άλγεβρα: ΕΞΩΤΕΡΙΚΗ ΣΥΝΕΝΩΣΗ (OUTER—JOIN)

- *Κίνητρα*: Σε μια τυπική συνένωση, πλειάδες στην Σχέση R ή στην S που δεν έχουν ταιριάζουσες πλειάδες στην άλλη σχέση δεν παρουσιάζονται στο αποτέλεσμα.

Σε μερικές ερωταποκρίσεις, όλες οι πλειάδες της R (ή S) πρέπει να παρουσιαστούν στο αποτέλεσμα - όταν δεν υπάρχουν ταιριάζουσες πλειάδες, τότε οι ειδικές τιμές NULL τοποθετούνται για τις τιμές γνωρισμάτων που λείπουν.

- **Συμβολισμός** : $R \oplus S$

- OUTER-JOINs χωρίζονται σε:

- *Left outer join* (όλες οι πλειάδες στην R παρουσιάζονται)
- *Right outer join* (όλες οι πλειάδες στην S παρουσιάζονται)
- *Full outer join* (όλες οι πλειάδες στην R και S παρουσιάζονται)

Σχεσιακή Άλγεβρα: ΔΙΑΙΡΕΣΗ (DIVISION)

- **ΔΙΑΙΡΕΣΗ:** Όταν δίνονται οι σχέσεις $R(X, Y)$ και $S(Y)$, όπου X, Y είναι σύνολα γνωρισμάτων, μια πλειάδα t είναι στοιχείο της διαίρεσης (συμβολίζεται: $(R / S)[X]$) ΑΝ για όλα τα t_S στην S υπάρχει κάποιο t_R στην R , έτσι ώστε: $t_R[Y] = t_S[Y]$ και $t_R[X] = t[X]$

– Αναλογία με την αριθμητική:

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

Το **πηλίκο** Q της R / S είναι η μέγιστη σχέση (σε πληθυσμό) έτσι ώστε $Q \times S \subseteq R$

Χρήσιμη Πράξη για ερωτήσεις της μορφής: **«Βρες τους υπαλλήλους που εργάζονται σε ΌΛΑ τα έργα που εργάζεται η Ελένη»**

Σχεσιακή Άλγεβρα: ΕΚΧΩΡΗΣΗ (Assignment)

- **ΕΚΧΩΡΗΣΗ:** 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:** Η διαίρεση r / s γράφεται ως:

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

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

$result = temp1 - temp2$

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

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

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)

- **QUERY1:** Βρες τα ονόματα των *ΝΑΥΤΙΚΩΝ* που έχουν κρατήσει το *ΣΚΑΦΟΣ* με νούμερο 2

(RESERVE [Bid=2] [Sid=Sid] SAILORS) [SName]

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

- **QUERY2:** Βρες τα ονόματα των *ΝΑΥΤΙΚΩΝ* που έχουν κρατήσει *ΣΚΑΦΗ* με χρώμα κόκκινο

(BOAT [Color=red] [Bid=Bid] RESERVE [Sid=Sid] SAILORS) [SName]

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

Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (3)

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

(SAILORS [SName=eleni] [Sid=Sid] RESERVE [Bid=Bid] BOATS) [Color]

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

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

(RESERVE [Sid=Sid] SAILORS) [SName]

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

Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (4)

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

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

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

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

Σχεσιακή Άλγεβρα: Ερωταποκρίσεις (5)

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

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

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

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

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)*
 - Σημείωση: Αυτό δεν σημαίνει απαραίτητα ότι η Γλώσσα είναι *ικανή* για όλες τις πράξεις σε μια βάση δεδομένων (ενδεικτικά, μια καλή γλώσσα πρέπει να υποστηρίζει συσσώρευση, πολλές μορφές συνένωσης, κτισμένες (built-in) συναρτήσεις,...)
- Ένας ενδιαφέρον τελεστής –που ξεπερνά την εκφραστική δύναμη των σχεσιακών πράξεων, όπως έχουν ορισθεί από τον Codd- είναι η *αναδρομική κλειστότητα (transitive closure.)* Αυτή είναι πολύ χρήσιμη πράξη σε Σχεσιακές Βάσεις.

Σχεσιακή Άλγεβρα: Αναδρομική Κλειστότητα

- Είναι μια αναδρομική συσχέτιση μεταξύ πλειάδων του ιδίου τύπου (π.χ., Υπάλληλος – Προϊστάμενος Υπάλληλος, part --- sub-part, κλπ.)
- Κλασσικές Ερωτήσεις
 - Περιπλανώμενος Πωλητής (από πόλη σε πόλη μέσω πόλεων)
 - Προϊστάμενοι σε όλα τα επίπεδα ενός Υπαλλήλου
- Η πράξη αυτή δεν υποστηρίζεται στην Σχεσιακή Άλγεβρα
- Υπάρχουν Γλώσσες ερωταποκρίσεων που τις υποστηρίζουν (π.χ., Query By Example – QBE)

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
 - Και αυτή είναι μια τυχαία απόφαση (θα μπορούσε ενδεχομένως κάθε *null* να είναι διαφορετικό)
 - Γενικά ακολουθείται εννοιολογικά για το *null* το μοντέλο της SQL

Κενές Τιμές

- Συγκρίσεις με *null* επιστρέφουν την τιμή *unknown*
 - Αν *false* επέστρεφε 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

- Διέγραψε όλες τις account εγγραφές στο Perryridge branch.

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

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

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

- Διέγραψε όλα τα accounts σε branches του Needham.

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

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

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

$account \leftarrow account - r_2$

$depositor \leftarrow depositor - r_3$

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.

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

$depositor \leftarrow depositor \cup \{("Smith", A-973)\}$

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

$r_1 \leftarrow (\sigma_{branch-name = "Perryridge"}(borrower \bowtie loan))$

$account \leftarrow account \cup \Pi_{branch-name, account-number, 200}(r_1)$

$depositor \leftarrow depositor \cup \Pi_{customer-name, loan-number}(r_1)$

ΕΝΗΜΕΡΩΣΗ (Updating)

- Ένας μηχανισμός για αλλαγές επιλεκτικά σε TIMES 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 %

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

ΌΨΕΙΣ / 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:

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

- 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

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

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

$$\Pi_{customer-name} \\ (\sigma_{branch-name = "Perryridge"} (all-customer))$$

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* $\leftarrow$ *branch-loan* $\cup$ {"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 επιλεχθεί κάποιο *loan* και *account*, και να δημιουργηθεί ένα νέο νόμισμα 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

Σχεσιακός Λογισμός: ΕΙΣΑΓΩΓΗ

- Ο **Σχεσιακός Λογισμός (RC)** είναι μια μη-διαδικαστική, τυπική γλώσσα που βασίζεται στον **κατηγορηματικό λογισμό πρώτης τάξης**
- Οι ερωταποκρίσεις στην RC προδιαγράφουν **ΤΙ** πρόκειται να ανακληθεί (δηλωτικά) ενώ το σύστημα (που υποστηρίζει την γλώσσα) αναλαμβάνει το **ΠΩΣ**
- Οι περισσότερες εμπορικές σχεσιακές γλώσσες έχουν τις ρίζες τους στον Σχεσιακό Λογισμό (**QUEL, QBE, SQL**). Αυτές οι γλώσσες δίνουν έμφαση στην ευκολία χρήσης.
- Ως προς την εκφραστικότητα, **ΣΧΕΣΙΑΚΗ ΑΛΓΕΒΡΑ** και **ΣΧΕΣΙΑΚΟΣ ΛΟΓΙΣΜΟΣ** είναι ταυτόσημοι.

Σχεσιακός Λογισμός: ΕΙΣΑΓΩΓΗ (2)

- Ο Σχεσιακός Λογισμός χρησιμοποιεί την έννοια της ΜΕΤΑΒΛΗΤΗΣ
- Υπάρχουν δυο μορφές της Γλώσσας:
 - **Λογισμός Πλειάδων**
οι μεταβλητές αναφέρονται σε πλειάδες από μια Σχέση
π.χ., t είναι μια μεταβλητή πλειάδος που αναφέρεται σε μια πλειάδα της $r(R)$
 - **Λογισμός Πεδίων Τιμών**
οι μεταβλητές αναφέρονται σε τιμές από ένα Πεδίο
π.χ., d είναι μια μεταβλητή πεδίου που αναφέρεται σε μια τιμή, που έχει μια πλειάδα της $r(R)$ για το γνώρισμα A (από το πεδίο D)

Σχεσιακός Λογισμός Πλειάδων

- Μια **ΕΚΦΡΑΣΗ (ΔΙΑΤΥΠΩΣΗ) στον ΣΧΕΣΙΑΚΟ ΛΟΓΙΣΜΟ** ορίζει μια νέα Σχέση με βάση της υπάρχουσες σχέσεις.
- Μια **έκφραση** κατασκευάζεται από τα παρακάτω στοιχεία:
 - (1) **Μεταβλητές πλειάδων** (π.χ., $t, v, w, t_1, t_2, t_3, \dots t_n$), οι οποίες ορίζονται να παίρνουν τιμές από ένα στιγμιότυπο σχέσης $r(R)$
Οι μεταβλητές μπορεί να είναι **περιορισμένες** (restricted), όπου, $t.A$, με A ένα γνώρισμα της R , συμβολίζει το A -τμήμα της τιμής του t
Παράδειγμα: $t.Name$
 - (2) **Συνθήκες** της μορφής x **τελεστής** y , όπου x, y είναι περιορισμένες μεταβλητές πλειάδος ή σταθερές τιμές και ο **τελεστής** $\in \{=, \neq, <, >, \leq, \geq\}$
Παράδειγμα: $t.Name = 'Μαρία', \quad t.City \neq v.City$

Εκφράσεις στον Λογισμό Πλειάδων

(3) Well-Formed Formulas (WFFs) (ατομική διατύπωση), ορίζονται ως:

- ♦ μια **συνθήκη** είναι WFF
- ♦ αν ***f*** είναι μια WFF, τότε είναι και η ***(f)*** και η **$\neg(f)$**
(όπου $\neg$ είναι το Λογικό **NOT**)
- ♦ αν ***f*** και ***g*** είναι WFFs, τότε είναι και η ***(f & g)*** και η ***(f | g)***
(όπου **&**, **|**, είναι τα Λογικά **AND, OR** αντίστοιχα)
- ♦ αν ***f*** είναι WFF με ελεύθερη μεταβλητή ***t***, τότε είναι και οι **$\exists t(f)$** , **$\forall t(f)$**
(όπου **$\exists$** , **$\forall$** είναι οι **υπαρξιακός (existential)** και **καθολικός (universal)** ποσοδείκτες)

Μια μεταβλητή πλειάδος ***t*** ονομάζεται δεσμευμένη (bound) σε μια έκφραση ***f*** αν είναι μια από μεταβλητές που ποσοδεικτούνται στην ***f***. Αλλιώς, ***t*** είναι ελεύθερη (free) στην ***f***.

Παράδειγμα: ***t*** είναι free στην **$f_1 : (t.City = 'london')$** και είναι bound στην **$f_2 : (\forall t) (t.DNumber = v.Dno)$** , ενώ η ***v*** είναι free in **f_2**

Εκφράσεις στον Λογισμό Πλειάδων

- Παραδείγματα των WFFs (ατομικών διατυπώσεων)
 - $t.City = v.City$
 - $(t.City = v.City)$
 - $\neg (t.City = v.City)$
 - $(t.City \neq v.City)$
 - $(t.City = v.City) \ \& \ (w.City = London)$
 - $\exists t (t.City = v.City)$
 - $\forall t (t.City = Athens)$
- αν f είναι μια WFF, τότε μια έκφραση / αίτημα στον λογισμό πλειάδων είναι κάθε έκφραση της μορφής:

$$\{ t_i.A_j \mid t_i \in r(R_k) \text{ AND } f \}$$

όπου τα R_k είναι σχήματα σχέσεων και τα A_j είναι γνωρίσματα στις R_k

Παραδείγματα Ερωταποκρίσεων (1)

- Επανερχόμαστε στην γνωστή ΒΔ των Ναυτικών

SAILORS (Sid, SName, Rating)

BOATS (Bid, BName, Color)

RESERVE (Sid, Bid, Date)

- **QUERY1:** *Βρες τα ονόματα των ΝΑΥΤΙΚΩΝ που έχουν id μεγαλύτερο από 10 και έχουν διαβάθμιση (rating) 'α'*

RANGE of t is SAILORS

t.SName where (t.Sid > 10) & (t.Rating = 'α')

*Παρακάτω δίνεται σαν **θεωρητικά καθαρό RC query**:*

{ t.SName | t ∈ r(SAILORS) & ((t.Sid > 10) & (t.Rating = 'α')) }

Παραδείγματα Ερωταποκρίσεων (2)

- **QUERY2:** *Βρες τα ονόματα και διαβαθμίσεις των ΝΑΥΤΙΚΩΝ που έχουν κρατήσει το ΣΚΑΦΟΣ με νούμερο 3*

RANGE of t is SAILORS

RANGE of v is RESERVE

t.SName, t.Rating where $\exists v ((t.Sid = v.Sid) \ \& \ (v.Bid = 3))$

- **QUERY3:** *Βρες τα ονόματα των ΝΑΥΤΙΚΩΝ που δεν έχουν κρατήσει το ΣΚΑΦΟΣ με νούμερο 3*

RANGE of t is SAILORS

RANGE of v is RESERVE

t.SName where $\neg (\exists v ((t.Sid = v.Sid) \ \& \ (v.Bid = 3)))$

ΑΣΦΑΛΕΙΑ Εκφράσεων

- Είναι δυνατόν να γραφούν εκφράσεις στο Σχεσιακό Λογισμό που δημιουργούν ΑΠΕΙΡΕΣ Σχέσεις.
- Για παράδειγμα η:
 $\{t \mid \neg t \in r\}$
έχει σαν αποτέλεσμα μια άπειρη σχέση (αν το πεδίο τιμών για την r είναι άπειρο.)
- Για αποφυγή αυτού του προβλήματος, περιορίζουμε τις εκφράσεις σε αυτές που ονομάζονται ασφαλείς εκφράσεις (safe expressions).
- Μια έκφραση $\{t \mid P(t)\}$ στο Σχεσιακό Λογισμό Πλειάδων ονομάζεται ΑΣΦΑΛΗΣ αν κάθε συνιστώσα της t παρουσιάζεται σε μια από τις σχέσεις, πλειάδες, ή σταθερές στην P
 - ΠΡΟΣΟΧΗ: Αυτό δεν είναι μια απλή συντακτική συνθήκη.
 - » Π.χ., $\{t \mid t[A]=5 \vee \mathbf{true}\}$ δεν είναι ασφαλής ---ορίζει ένα άπειρο σύνολο με τιμές γνωρισμάτων που δεν παρουσιάζονται σε καμία σχέση, πλειάδα ή σταθερά στην P .