

ΣΧΕΣΙΑΚΟΣ ΛΟΓΙΣΜΟΣ

Τι θα δούμε

- **Σχεσιακός Λογισμός**
 - Παραδείγματα
- **Σχεσιακή Πληρότητα**

Εισαγωγή στον Σχεσιακό Λογισμό

- Ο Σχεσιακός Λογισμός (**Relational Calculus** ή **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 τη γλώσσα) αναλαμβάνει το **πώς**.
- 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, **σχεσιακή άλγεβρα** και **σχεσιακός λογισμός** είναι ταυτόσημοι.

Λογισμός πλειάδων και Λογισμός πεδίων

- Ο Σχεσιακός Λογισμός χρησιμοποιεί την έννοια της μεταβλητής
- 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

- Κάθε ερώτημα είναι της μορφής
$$\{t \mid P(t)\}$$
- Όλα τα tuples t τέτοια ώστε το κατηγορημα (predicate) P να είναι αληθές
- t είναι μια μεταβλητή πλειάδας (*tuple variable*)
- $t \in r$ σημαίνει ότι t ορίζεται να παίρνει τιμές από ένα στιγμιότυπο σχέσης $r(R)$
- Οι μεταβλητές μπορεί να είναι περιορισμένες (restricted)
$$t[A], \text{ όπου } A \text{ ένα γνώρισμα της } R$$
- P είναι μια έκφραση κατηγορηματικού λογισμού (predicate calculus)

Έκφραση κατηγορηματικού λογισμού

1. Μεταβλητές (πλειάδες) και σταθερές
2. Τελεστές σύγκρισης: (π.χ., $<$, \leq , $=$, \neq , $>$, \geq)
3. Συνδετικά στοιχεία: and (\wedge), or (\vee), not (\neg)
4. Συνεπαγωγή(\Rightarrow): $x \Rightarrow y$, αν ισχύει το x , ισχύει και το y
$$x \Rightarrow y \equiv \neg x \vee y$$
5. Υπαρξιακοί (existential) και καθολικοί (universal) ποσοδείκτες (quantifiers)
 - $\exists t \in r (Q(t)) \equiv$ "υπάρχει" ένα tuple t στη σχέση r τέτοιο ώστε το $Q(t)$ να είναι true
 - $\forall t \in r (Q(t)) \equiv Q$ είναι true "για κάθε" tuple t στη σχέση r
6. Μια μεταβλητή t ονομάζεται δεσμευμένη (bound) σε μια έκφραση f αν είναι μια από μεταβλητές που ποσοδεικτούνται στην f . Αλλιώς, η t είναι ελεύθερη (free) στην f .

Παράδειγμα: t είναι ελεύθερη στην $f1 (t.City = 'london')$ και δεσμευμένη στην $f2 : (\forall t) (t.DNumber = v.Dno)$

Παράδειγμα τράπεζας

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*)

Find the *loan-number*, *branch-name*, and *amount* for loans of over \$1200

$$\{t \mid t \in \text{loan} \wedge t[\text{amount}] > 1200\}$$

Παράδειγμα τράπεζας

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)

Find the names of all customers having a loan, an account, or both at the bank

$$\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}]) \\ \vee \exists u \in \text{depositor}(t[\text{customer-name}] = u[\text{customer-name}])\}$$

Find the names of all customers who have a loan and an account at the bank

$$\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}]) \\ \wedge \exists u \in \text{depositor}(t[\text{customer-name}] = u[\text{customer-name}])\}$$

Παράδειγμα τράπεζας

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)

Find the names of all customers having a loan at the Perryridge branch

$$\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}] \\ \wedge \exists u \in \text{loan}(u[\text{branch-name}] = \text{“Perryridge”} \\ \wedge u[\text{loan-number}] = s[\text{loan-number}]))\}$$

Find the names of all customers who have a loan at the
Perryridge branch, but no account at any branch of the bank

$$\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}] \\ \wedge \exists u \in \text{loan}(u[\text{branch-name}] = \text{“Perryridge”} \wedge u[\text{loan-number}] = s[\text{loan-number}])) \\ \wedge \text{not } \exists v \in \text{depositor} (v[\text{customer-name}] = t[\text{customer-name}]) \}$$

Παράδειγμα τράπεζας

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)

Find the names of all customers having a loan from the Perryridge branch, and the cities they live in

$$\{t \mid \exists s \in \text{loan}(s[\text{branch-name}] = \text{"Perryridge"})$$
$$\wedge \exists u \in \text{borrower}(u[\text{loan-number}] = s[\text{loan-number}]$$
$$\wedge t[\text{customer-name}] = u[\text{customer-name}])$$
$$\wedge \exists v \in \text{customer}(u[\text{customer-name}] = v[\text{customer-name}]$$
$$\wedge t[\text{customer-city}] = v[\text{customer-city}])\}$$

Παράδειγμα τράπεζας

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)

Find the names of all customers who have an account at all branches located in Brooklyn:

$$\{t \mid \forall s \in \text{branch}(s[\text{branch-city}] = \text{“Brooklyn”} \Rightarrow \\ \exists u \in \text{account} (s[\text{branch-name}] = u[\text{branch-name}] \\ \wedge \exists s \in \text{depositor} (t[\text{customer-name}] = s[\text{customer-name}] \\ \wedge s[\text{account-number}] = u[\text{account-number}])))\}$$

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 .

Λογισμός πεδίων τιμών

- Μη-διαδικαστική γλώσσα ισοδύναμη του λογισμού πλειάδων
- Κάθε query είναι μια έκφραση της μορφής

$$\{ \langle x_1, x_2, \dots, x_n \rangle \mid P(x_1, x_2, \dots, x_n) \}$$

- x_1, x_2, \dots, x_n είναι μεταβλητές πεδίου
- P αντιπροσωπεύει σχέση κατηγορηματικού λογισμού

Example Queries

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)

- Find the *loan-number*, *branch-name*, and *amount* for loans of over \$1200

$$\{ \langle l, b, a \rangle \mid \langle l, b, a \rangle \in \text{loan} \wedge a > 1200 \}$$

- Find the names of all customers who have a loan of over \$1200

$$\{ \langle c \rangle \mid \exists l, b, a (\langle c, l \rangle \in \text{borrower} \wedge \langle l, b, a \rangle \in \text{loan} \wedge a > 1200) \}$$