

Κεφάλαιο 3

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ Α' ΜΕΡΟΣ

Το Σχεσιακό Μοντέλο – ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

- **Άτυπος και Τυπικός Ορισμός του Μοντέλου**
 - Δομές, Πράξεις / Λειτουργίες, Δομικοί Περιορισμοί
- **Σχεσιακή Άλγεβρα**
 - Βασικές Πράξεις της Σχεσιακής Άλγεβρας
 - Παραδείγματα

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.

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ: Άτυπος Ορισμός

- Μια Σχεσιακή Βάση Δεδομένων είναι ένα σύνολο από σχέσεις
- **Σχέση (Relation):**
 - Ένας Πίνακας τιμών.
 - Κάθε **στήλη** στον πίνακα έχει επικεφαλίδα (όνομα), που ονομάζεται **γνώρισμα (attribute)**.
 - Κάθε **γραμμή** ονομάζεται μια **πλειάδα (tuple)** και παριστά τα χαρακτηριστικά μιας οντότητας στο μοντέλο.

ΣΧΕΣΕΙΣ

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

Τυπικός Ορισμός

- Δομές
 - Μόνο ένα είδος: **relations (σχέσεις)** που έχουν όνομα
- Τυπικά, δεδομένων των συνόλων D_1, D_2, \dots, D_n μια σχέση r είναι ένα υποσύνολο του $D_1 \times D_2 \times \dots \times D_n$
Επομένως μια σχέση είναι ένα σύνολο από n -πλειάδες (a_1, a_2, \dots, a_n) όπου κάθε $a_i \in D_i$

π.χ. $customer-name = \{Jones, Smith, Curry, Lindsay\}$

$customer-street = \{Main, North, Park\}$

$customer-city = \{Harrison, Rye, Pittsfield\}$

Τότε $r = \{$
 $(Jones, Main, Harrison),$
 $(Smith, North, Rye),$
 $(Curry, North, Rye),$
 $(Lindsay, Park, Pittsfield)\}$

είναι μια σχέση στο $customer-name \times customer-street \times customer-city$

Τύποι γνωρισμάτων

- Κάθε γνώρισμα έχει ένα όνομα
- Το σύνολο των επιτρεπόμενων τιμών για κάθε γνώρισμα A λέγεται *πεδίο τιμών (domain)* $D(A)$ του γνωρίσματος
 - $D(A) = \{d_1, d_2, \dots, d_n\}$
- Οι τιμές των γνωρισμάτων πρέπει (συνήθως) να είναι ατομικές, δηλ αδιαίρετες
 - Π.χ., πλειότιμα γνωρίσματα δεν είναι ατομικά
 - Π.χ., σύνθετα γνωρίσματα δεν είναι ατομικά
- Η ειδική τιμή *null* είναι μέλος οποιουδήποτε domain
 - τιμή που είναι μη εφαρμόσιμη (non-applicable) ή άγνωστη, π.χ, Αριθμός Τηλεφώνου κάποιου που δεν έχει τηλέφωνο
 - Η τιμή null value δημιουργεί επιπλοκές στον ορισμό κάποιων πράξεων

Ορισμοί Δομών (1)

- 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, ... }

Ορισμοί Δομών (2)

- Μια *πλειάδα (Tuple)* t μιας Σχέσης $R(A_1, A_2, \dots, A_n)$ είναι μια (διατεταγμένη) λίστα τιμών $\mathbf{t} = \langle v_1, v_2, \dots, v_n \rangle$, 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
$$\mathbf{r(R)} = \{ \mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_k \}$$
εναλλακτικά, είναι ένα υποσύνολο του Καρτεσιανού Γινομένου
$$\mathbf{r(R)} \subset \mathbf{D(A_1)} \times \mathbf{D(A_2)} \times \dots \times \mathbf{D(A_n)}$$
- Η *πληθικότητα (cardinality)* της R είναι ο αριθμός των πλειάδων στην $r(R)$, και συμβολίζεται με \mathbf{CARD}_R
- Μια *σχεσιακή βάση (relational database)* είναι ένα σύνολο σχέσεων

Στιγμιότυπο Σχέσης

- Σχέση customer = πίνακας
- Attributes = ονόματα στηλών
- Tuples = γραμμές του πίνακα

customer-name	customer-street	customer-city
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

customer

attributes
(or columns)

tuples
(or rows)

Χαρακτηριστικά των Σχέσεων

- Η διάταξη των γνωρισμάτων σε μια σχέση είναι σημαντική
- Η διάταξη 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 διασπασίμη) – συχνά αυτό αναφέρεται σαν First Normal Form (1-NF) σχέση
- Συμβολισμός: τιμή γνωρίσματος A_i για μια πλειάδα t

$$t[A_i] = v_i$$

Δομικοί Περιορισμοί

■ Έμφυτοι περιορισμοί

- Κλειδιών (key)
 - » Τα διαφορετικά κλειδιά, όπως ορίστηκαν στο μοντέλο E-R, ισχύουν και στο Σχεσιακό Μοντέλο.
 - » Υπενθύμιση: το κλειδί είναι ιδιότητα του Σχεσιακού Σχήματος, όχι του στιγμιότυπου της Σχέσης (δηλαδή, ισχύει για ΟΛΑ τα στιγμιότυπα)
- Ακεραιότητα οντότητας (Entity integrity)
- Αναφορική ακεραιότητα (Referential integrity)

■ Ρητοί περιορισμοί

- Πεδίου τιμών (domain)
- Στηλών (column)
- Οριζόμενων από τον χρήστη (user-defined)
- άλλοι ρητοί περιορισμοί, π.χ., οι Συναρτησιακές Εξαρτήσεις (Functional Dependencies,) θα αναφερθούν αργότερα

Περιορισμοί Κλειδιού

- Έστω $K \subseteq R$
- K είναι **υπέρ-κλειδί (superkey)** του σχήματος σχέσης R αν οι τιμές του K είναι αρκετές για να ταυτοποιήσουν μια μοναδική πλειάδα για κάθε πιθανή σχέση $r(R)$
 - Παράδειγμα: $\{customer-name, customer-street\}$ και $\{SSN\}$ είναι υπερκλειδιά του *Customer*
- 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 κλειδί.
 - Π.χ., SSN είναι υποψήφιο κλειδί στο *Customer*, 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.

Περιορισμοί Ακεραιότητας Οντότητας

- Το κύριο κλειδί PK στο σχήμα της σχέσης R δεν μπορεί να έχει κενές (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.

Περιορισμοί Αναφορικής Ακεραιότητας

- 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:

$$t_1[\text{FK}] = t_2[\text{FK}]$$

π.χ., EMPLOYEE (SSN, Name, BirthDate, Address, Sex, Salary, DNumber)

PROJECT (PNumber, PName, Location, DNumber)

WORKS_ON (SSN, PNumber, HoursPW)

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)

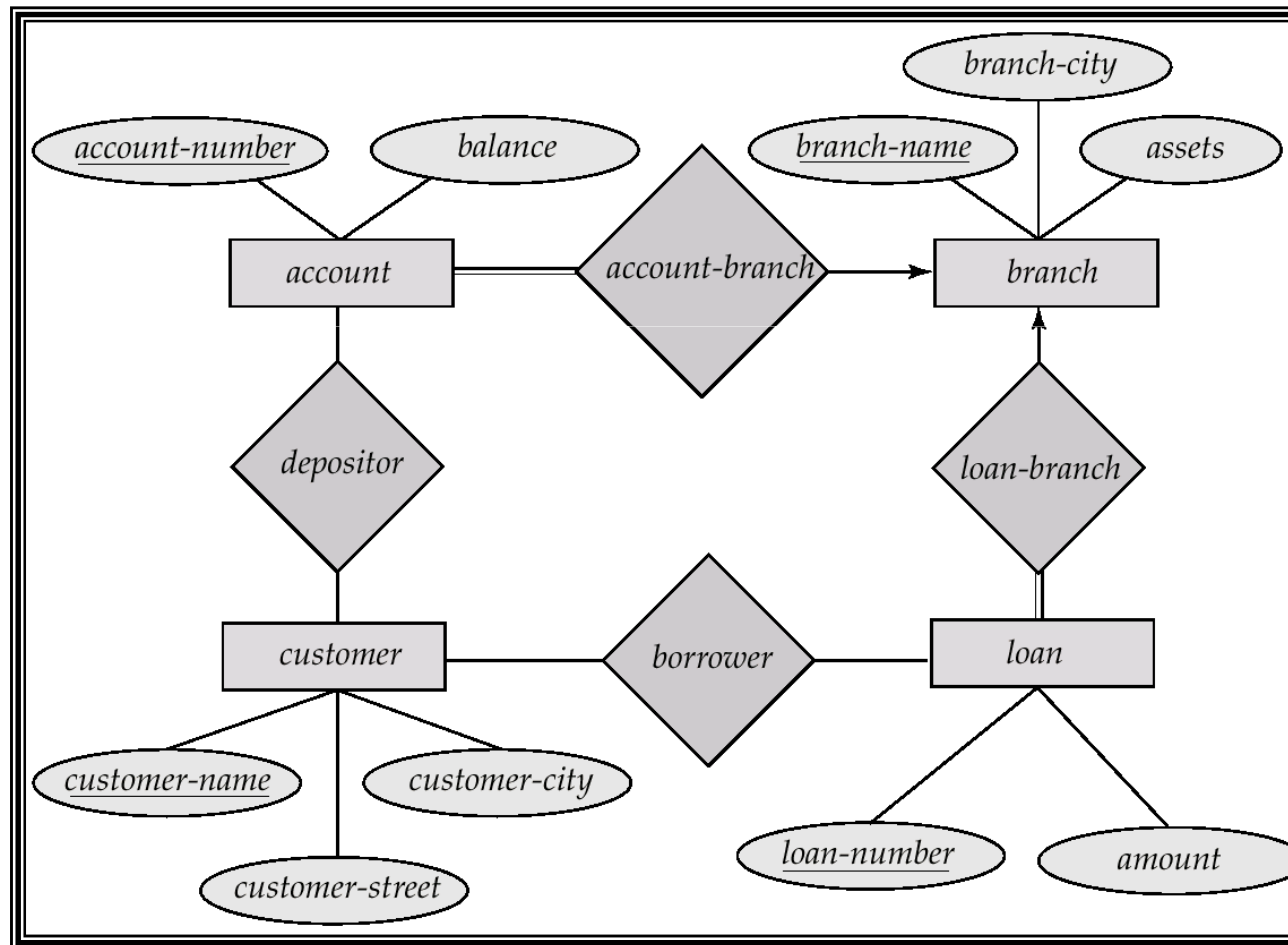
Περιορισμοί Στηλών

- 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***

Περιορισμοί Οριζόμενοι από τον Χρήστη

- 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**

E-R διάγραμμα τράπεζας



Από E/R Diagrams σε Relations

- Τύποι Οντοτήτων γίνονται Σχέσεις (Πίνακες, Relations) με τα ίδια χαρακτηριστικά (attributes)
- Τύποι Συσχετίσεων γίνονται Σχέσεις των οποίων τα Χαρακτηριστικά είναι (μόνο):
 - Τα Κλειδιά των συσχετιζόμενων Οντοτήτων
 - Χαρακτηριστικά του Τύπου Συσχέτισης

Καθορίζοντας τα κλειδιά από το E-R

■ Ισχυρός τύπος οντότητας

- Το πρωτεύον κλειδί της οντότητας γίνεται πρωτεύον κλειδί της σχέσης

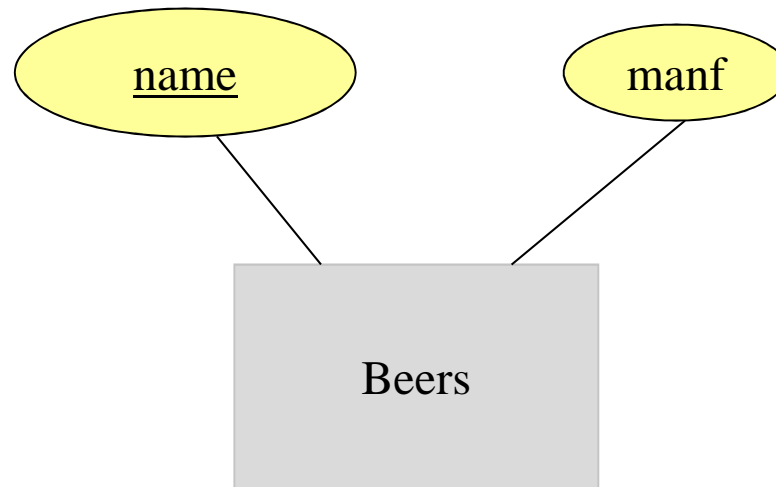
■ Ασθενής τύπος οντότητας

- Το πρωτεύον κλειδί της σχέσης αποτελείται από την ένωση του πρωτεύοντος κλειδιού του ισχυρού τύπου οντότητας και του μερικού κλειδιού του ασθενούς τύπου οντότητας

■ Τύπος συσχέτισης

- Η ένωση των πρωτευόντων κλειδιών των σχετιζόμενων οντοτήτων γίνεται υπερκλειδί της σχέσης
- Για δυαδικές N:1 συσχετίσεις, το πρωτεύον κλειδί της οντότητας στην πλευρά του N γίνεται πρωτεύον κλειδί της σχέσης
- Για 1:1 συσχετίσεις, το πρωτεύον κλειδί της σχέσης μπορεί να είναι το πρωτεύον κλειδί οποιασδήποτε από τις 2 εμπλεκόμενες οντότητες
- Για N:M συσχετίσεις, το πρωτεύον κλειδί της σχέσης είναι η ένωση των πρωτευόντων κλειδιών των σχετιζόμενων οντοτήτων

Entity Set -> Relation



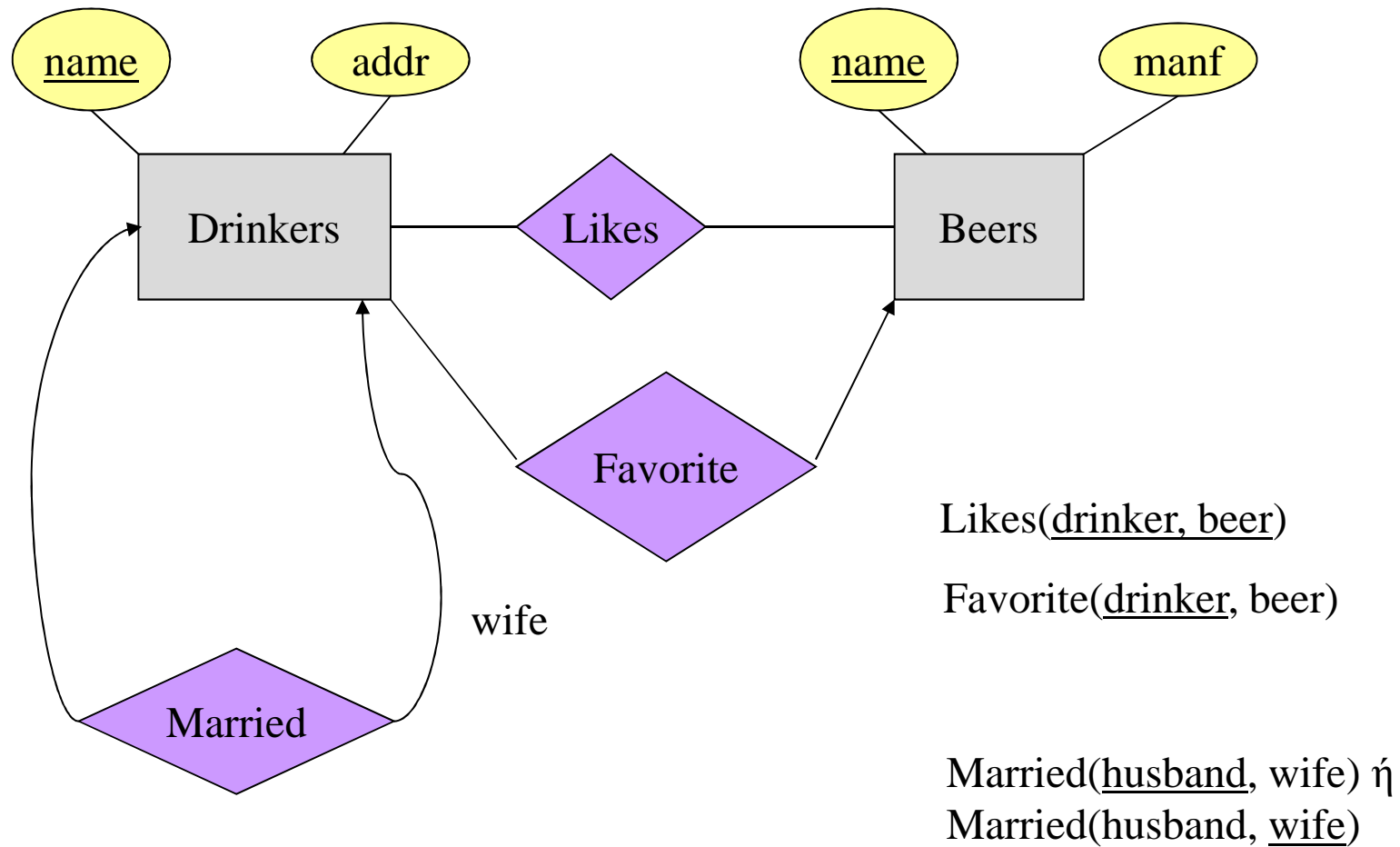
Relation: Beers(name, manf)

Τύποι Οντοτήτων με υποκλάσεις

■ 3 τρόποι:

1. *Object-oriented* : Μια σχέση για κάθε υποκλάση, με όλα τα attributes των υπερκλάσεών της. Υποθέτουμε ότι κάθε οντότητα ανήκει σε μία υποκλάση.
2. *Χρήση nulls* : 1 σχέση, οντότητες με null τιμές σε attributes που δεν τις αφορούν
3. *E/R style* : 1 σχέση για κάθε subclass, με μόνο τα key attributes και τα attributes που αφορούν την κάθε subclass. Κάθε οντότητα αντιπροσωπεύεται σε όλες τις σχέσεις των κλάσεων που ανήκει

Relationship -> Relation



Συνδυασμοί Σχέσεων

- Είναι αποδεκτό (και πολλές φορές, επιθυμητό) να συνδυάζονται η Σχέση για ένα Τύπο Οντοτήτων E με τη Σχέση R για την N προς 1 Συσχέτιση της E με κάποιον άλλον Τύπο Οντοτήτων.

- Παράδειγμα:

Drinkers(name, addr) και Favorite(drinker, beer)

Συνδυάζονται στο:

Drinker1(name, addr, favBeer)

Risk with Many-Many Relationships

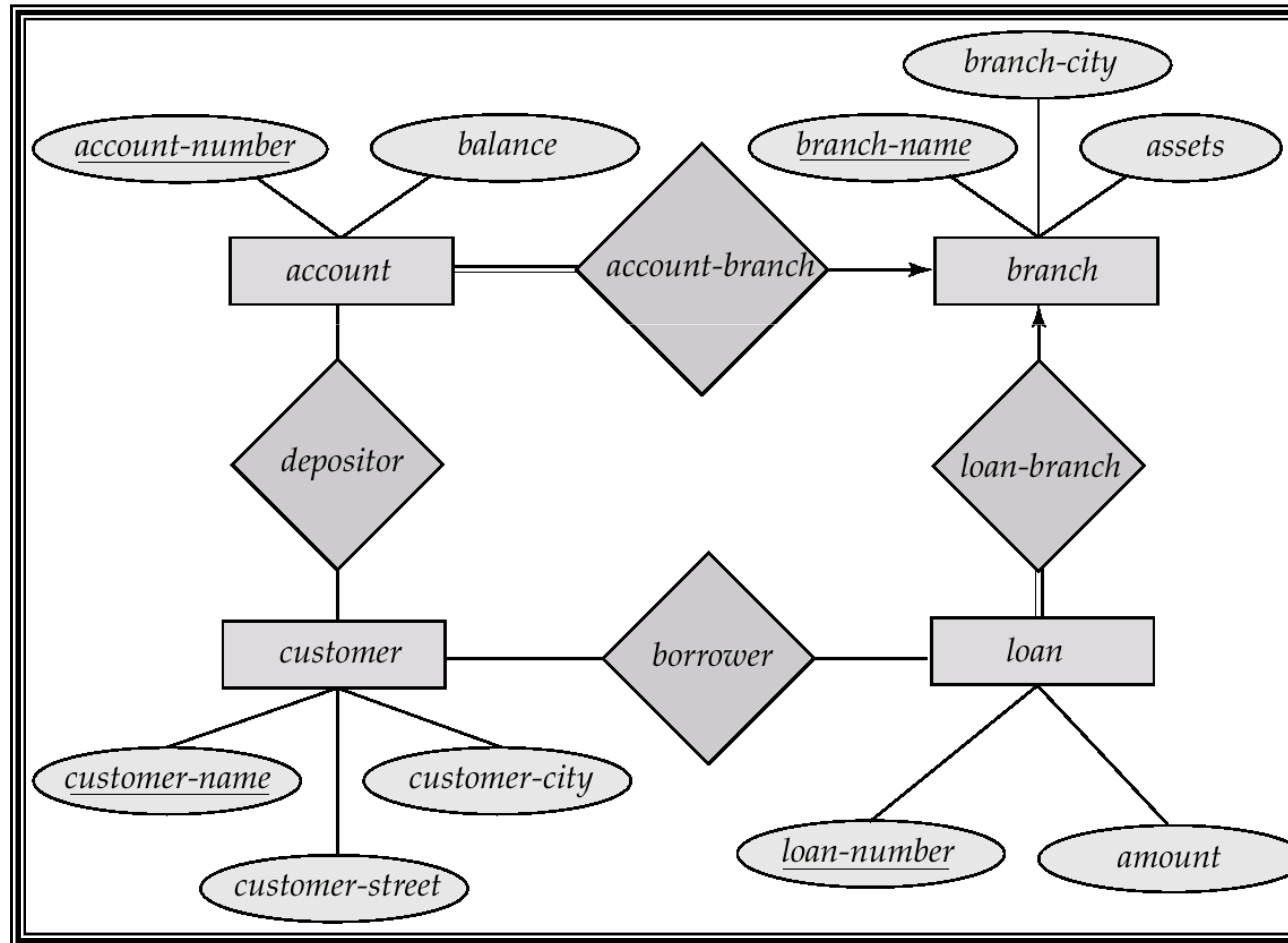
- Συνδυάζοντας το Drinkers με το Likes είναι λάθος μια και οδηγεί σε φαινόμενα όπως

name	addr	beer
Sally	123 Maple	Bud
Sally	123 Maple	Miller

Redundancy



E-R διάγραμμα τράπεζας



Σχέσεις της Τράπεζας

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

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

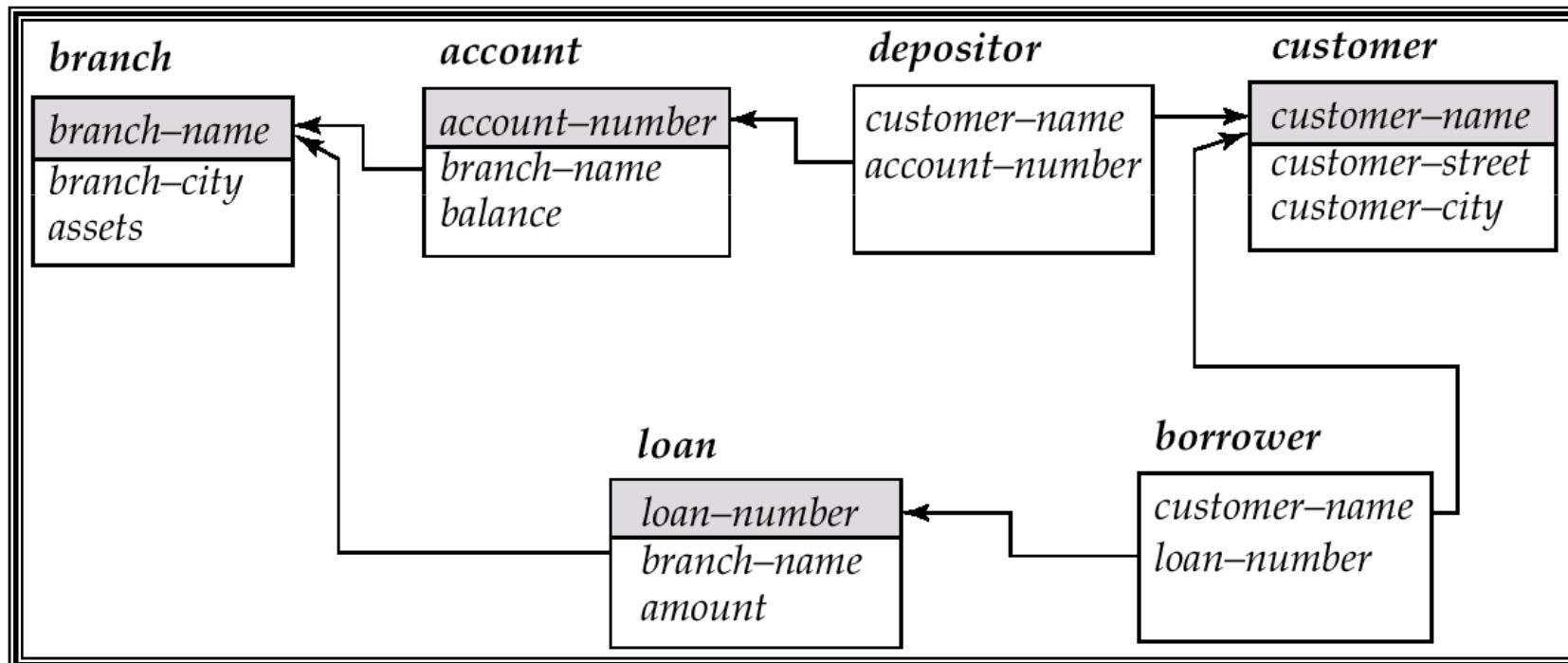
account (account-number, branch-name, balance)

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

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

Σχεσιακό Διάγραμμα Σχήματος



Branch Relation

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Borrower Relation

<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 Relation

<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

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 ανάκτησης – Σχεσιακές γλώσσες

- Η γλώσσα που χρησιμοποιείται για την επεξεργασία και ανάκτηση πληροφορίας από τη βάση
- Κατηγορίες
 - procedural
 - non-procedural
- “Pure” languages:
 - Σχεσιακή Άλγεβρα (Relational Algebra) - διαδικαστικού τύπου, προσδιορίζει πως υπολογίζεται το αποτέλεσμα
 - Σχεσιακός Λογισμός (Relational Calculus) - δηλωτικού χαρακτήρα, προσδιορίζει ποιες ιδιότητες πρέπει να πληροί το αποτέλεσμα

Σχεσιακές Γλώσσες

- Κανένα σχεσιακό DBMS δεν υποστηρίζει τις δύο κατηγορίες γλωσσών στην *πλήρη θεωρητική* μορφή των. Όμως, οι γλώσσες που υποστηρίζονται σε DBMS έχουν τις ρίζες τους είτε στην Σχεσιακή Άλγεβρα ή στον Σχεσιακό Λογισμό.
- 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.

Σχεσιακή Άλγεβρα

- Διαδικαστική γλώσσα
- 6 βασικές πράξεις (συνολοθεωρητικές ή εξειδικευμένες για βάσεις)
 - select
 - project
 - union
 - set difference
 - Cartesian product
 - rename
- Παίρνουν σαν είσοδο 1 ή περισσότερες σχέσεις και δίνουν ως αποτέλεσμα μια νέα σχέση
- Οι τελεστές, όπως και στην αριθμητική άλγεβρα, μπορεί να φωλιάσουν (**nested**), εφόσον το αποτέλεσμα κάθε πράξης είναι μια Σχέση

ΕΠΙΛΟΓΗ (SELECTION)

- Επιλέγει τις πλειάδες μιας σχέσης που πληρούν κάποια συνθήκη (qualification) c , η οποία είναι μια λογική παράσταση (Boolean Expression) στα γνωρίσματα της Σχέσης

- Συμβολισμός: $\sigma_c(\mathbf{R})$ ή $\mathbf{R}[c]$

- Τυπικά: $\sigma_c(\mathbf{R}) = \{ t \mid t \text{ is in } r(\mathbf{R}) \text{ and condition } c \text{ holds for } t \}$

όπου c τύπος που αποτελείται από terms που συνδέονται με:

\wedge (and), \vee (or), \neg (not)

Κάθε term είναι της μορφής:

$\langle \text{attribute} \rangle$ op $\langle \text{attribute} \rangle$ or $\langle \text{constant} \rangle$

όπου op είναι: $=$, \neq , $>$, \geq , $<$, \leq

- Παραδείγματα : $\sigma_{\text{branch-name}=\langle \text{Perryridge} \rangle}(\text{loan})$,
 $\text{loan}[\text{branch-name}=\langle \text{Perryridge} \rangle]$

ΕΠΙΛΟΓΗ

– Η ΕΠΙΛΟΓΗ είναι *commutative* και *associative*

(a) $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_2}(\sigma_{c_1}(R))$

(b) $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_1 \text{ AND } c_2}(R) = \sigma_{c_1, c_2}(R)$

(c) $\sigma_{c_1}(\sigma_{c_2}(\sigma_{c_3}(R))) = \sigma_{c_2}(\sigma_{c_3}(\sigma_{c_1}(R)))$

Επιλογή - Παράδειγμα

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- $\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

Παράδειγμα Τράπεζας

$\sigma_{branch-name=\langle\text{Perryridge}\rangle}(loan)$

Loan relation

<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>loan-number</i>	<i>branch-name</i>	<i>amount</i>
L-15	Perryridge	1500
L-16	Perryridge	1300

ΠΡΟΒΟΛΗ (PROJECTION)

- 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** : $\pi_{\text{loan-number, amount}}(\text{loan})$, or $\text{loan}[\text{loan-number, amount}]$

Προβολή - Παράδειγμα

■ Relation r :

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

■ $\pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Παράδειγμα Τράπεζας

$\Pi_{loan-number, amount}(\text{account})$

Loan relation

<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>loan-number</i>	<i>amount</i>
L-11	900
L-14	1500
L-15	1500
L-16	1300
L-17	1000
L-23	2000
L-93	500

Ένωση (Union)

- Βάλτε όλες τις πλειάδες των δύο Σχέσεων σε μια Σχέση
- 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,

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

- **Συμβολισμός:** $R \cup S$
- **Τυπικά:** $R \cup S = \{ t \mid t \text{ is in } R \text{ or } t \text{ is in } S \}$
- **Παράδειγμα:** find all customers with either an account or a loan
 $\pi_{\text{customer-name}} (\text{depositor}) \cup \pi_{\text{customer-name}} (\text{borrower})$

Ένωση - Παράδειγμα

- Relations r, s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r \cup s$:

A	B
α	1
α	2
β	1
β	3

Παράδειγμα Τράπεζας

$\pi_{\text{customer-name}}(\text{depositor}) \cup \pi_{\text{customer-name}}(\text{borrower})$

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>
Adams
Curry
Hayes
Jackson
Jones
Smith
Williams
Lindsay
Johnson
Turner

Διαφορά – Set Difference

- Επίλεξε τις πλειάδες της πρώτης Σχέσης, που δεν είναι στοιχεία της δεύτερης Σχέσης
- Συμβατότητα σχέσεων
 - **Συμβολισμός:** $R - S$
 - **Τυπικά:** $R - S = \{ t \mid t \text{ is in } R \text{ and } t \text{ is not in } S \}$
 - **Παράδειγμα :** Customers With An Account But No Loan
depositor - borrower

Διαφορά - Παράδειγμα

■ Relations r, s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r - s$:

A	B
α	1
β	1

Παράδειγμα Τράπεζας

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

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>
Johnson
Lindsay
Turner

Καρτεσιανό Γινόμενο (Cartesian Product)

- 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** : $\text{borrower} \times \text{loan}$

Καρτεσιανό Γινόμενο - Παράδειγμα

Relations r, s:

A	B
α	1
β	2

α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

α	10	a
β	10	a
β	20	b
γ	10	b

s

r x s:

A	B	C	D	E
---	---	---	---	---

α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Παράδειγμα Τράπεζας

borrower × *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

<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

Σύνθεση Πράξεων

- Σύνθετες εκφράσεις με πολλαπλές πράξεις

- Example: $\sigma_{A=C}(r \times s)$

- $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

- $\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	20	a
β	2	β	20	b

Παράδειγμα Τράπεζας

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

<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

Μετονομασία - Rename

- 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 .

Example Queries

Find all loans of over \$1200

$$\sigma_{\text{amount} > 1200} (\text{loan})$$

Find the loan number for each loan of an amount greater than \$1200

$$\pi_{\text{loan-number}} (\sigma_{\text{amount} > 1200} (\text{loan}))$$

Example Queries

Find the names of all customers who have a loan, an account, or both, from the bank

$$\pi_{\text{customer-name}}(\text{borrower}) \cup \pi_{\text{customer-name}}(\text{depositor})$$

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

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

Example Queries

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

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

Find the names of all customers who have a loan at the Perryridge branch but do not have an account at any branch of the bank.

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

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:

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