

Θεωρία Κανονικοποίησης



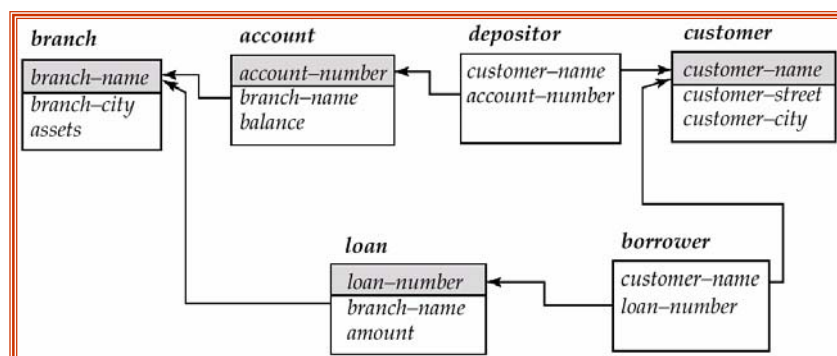
- Πρώτη Κανονική Μορφή (1NF)
- Αποσύνθεση - Συναρτησιακές Εξαρτήσεις
- Δεύτερη (2NF) και Τρίτη Κανονική Μορφή (3NF)
- Boyce-Codd Κανονική Μορφή (BCNF)
- Καθολική Διαδικασία Σχεδίασης ΒΔ

Βασική πηγή διαφανειών: Silberschatz et al., "Database System Concepts", 4/e
Εργαστήριο Πληροφοριακών Συστημάτων, Παν/μιο Πειραιώς (<http://infolab.cs.unipi.gr/>)
έκδοση: Φεβ. 2010

Το πρόβλημα ...



- Γιατί αυτοί οι πίνακες και όχι κάποιοι άλλοι (περισσότεροι ή λιγότεροι, με διαφορετικό μοίρασμα χαρακτηριστικών);



Στόχοι στη σχεδίαση σχεσιακών ΒΔ

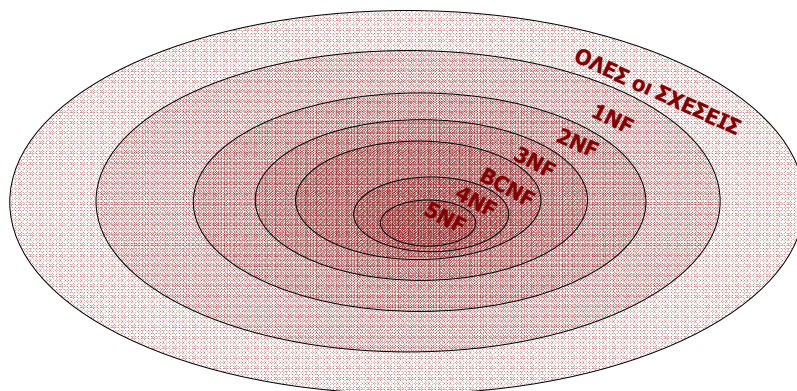


- Στόχος της σχεδίασης σχεσιακών ΒΔ είναι να βρούμε μια "καλή" συλλογή τύπων σχέσεων. Κακή σχεδίαση μπορεί να σημαίνει:
 - Επανάληψη / πλεονασμό πληροφορίας
 - Αδυναμία αναπαράστασης κάποιας πληροφορίας
- Σχεδιαστικοί στόχοι:
 - Αποφυγή πλεονασμού στα δεδομένα
 - Εξασφάλιση ότι αναπαριστώνται όλες οι συσχετίσεις μεταξύ των οντοτήτων
 - Διευκόλυνση των ελέγχων για τυχόν παραβίαση των περιορισμών ορθότητας
- Το «εργαλείο»: **κανονικοποίηση ΒΔ**
 - Υπακοή σε κάποια κανονική μορφή (1^η, 2^η, 3^η, ...)

Κανονικές μορφές



- Έχουν οριστεί 5 **κανονικές μορφές** (normal forms - NF) που η επόμενη είναι πιο αυστηρή από την προηγούμενη.
 - Μεταξύ 3^{ης} και 4^{ης} κανονικής μορφής, έχει οριστεί η κανονική μορφή Boyce-Codd (BCNF) που πρακτικά αποτελεί το 'στόχο' της διαδικασίας σχεδίασης ΒΔ





- ➔ ▪ Πρώτη Κανονική Μορφή (1NF)
- Αποσύνθεση - Συναρτησιακές Εξαρτήσεις
- Δεύτερη (2NF) και Τρίτη Κανονική Μορφή (3NF)
- Boyce-Codd Κανονική Μορφή (BCNF)
- Καθολική Διαδικασία Σχεδίασης ΒΔ

Πρώτη Κανονική Μορφή



- Οι τιμές του πεδίου ορισμού ενός χαρακτηριστικού είναι **ατομικές** εάν θεωρούνται ως αδιαίρετες μονάδες
 - Αντιπαδείγματα: σύνολα ονομάτων, σύνθετα χαρακτηριστικά, κωδικοί (π.χ. CS101) που σπάζουν σε τμήματα
- Ένας τύπος σχέσης R βρίσκεται στην **1^η κανονική μορφή (first normal form – 1NF)** εάν τα πεδία τιμών όλων των χαρακτηριστικών του R είναι **ατομικά**
- Μη ατομικές τιμές επιφέρουν δυσκολίες στην αποθήκευση και πλεονασμό δεδομένων
 - μαζί με κάθε πελάτη να αποθηκεύονται τα στοιχεία όλων των λογαριασμών του, μαζί με κάθε λογαριασμό να αποθηκεύονται τα στοιχεία όλων των κατόχων του κλπ.

Πρώτη Κανονική Μορφή (συν.)



- Η ατομικότητα ουσιαστικά έχει να κάνει με τον τρόπο με τον οποίο χρησιμοποιούνται οι τιμές των χαρακτηριστικών π.χ.
 - οι συμβολοσειρές συνήθως θεωρούνται αδιαίρετες. Αν όμως ο ΑΜ του φοιτητή είναι μια συμβολοσειρά της μορφής CS0012 και οι δύο πρώτοι χαρακτήρες χρησιμοποιούνται για να βρούμε το Τμήμα του φοιτητή, τότε καταστρατηγούμε την ατομικότητα
 - Αυτό είναι κακή ιδέα γιατί μας οδηγεί σε συγγραφή κώδικα σε κάποιο πρόγραμμα εφαρμογής (έξω από τη βάση δεδομένων)



- Πρώτη Κανονική Μορφή (1NF)
- Αποσύνθεση - Συναρτησιακές Εξαρτήσεις
- Δεύτερη (2NF) και Τρίτη Κανονική Μορφή (3NF)
- Boyce-Codd Κανονική Μορφή (BCNF)
- Καθολική Διαδικασία Σχεδίασης ΒΔ



Παράδειγμα «προβληματικού» σχήματος



- Έστω το σχήμα:
 $Lending-schema = (branch-name, branch-city, assets, customer-name, loan-number, amount)$

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

- Πλεονασμός:
 - Επανάληψη των δεδομένων *branch-name*, *branch-city*, *assets* για κάθε δάνειο που δίνει το συγκεκριμένο υποκατάστημα → σπατάλη χώρου
 - Δυσκολία στην τροποποίηση δεδομένων, πιθανότητα ασυνεπών τιμών
- Τιμές Null
 - Δεν μπορούμε να αποθηκεύσουμε τα στοιχεία ενός υποκαταστήματος αν δεν έχει δοθεί κανένα δάνειο από αυτό
 - Μπορούμε να χρησιμοποιήσουμε τιμές null αλλά είναι δύσκολος ο χειρισμός τους.

ΒΔ: [4] Θεωρία Κανονικοποίησης

9

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Αποσύνθεση (decomposition)



- Αποσύνθεση του σχήματος

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

σε

 $Branch-schema = (branch-name, branch-city, assets)$
 $Loan-info-schema = (customer-name, loan-number, branch-name, amount)$

- Όλα τα χαρακτηριστικά του αρχικού τύπου σχέσης (R) πρέπει να διατηρούνται στην αποσύνθεση (R_1, R_2): $R = R_1 \cup R_2$
- Στόχος της σχεδίασης σχεσιακών βάσεων δεδομένων: να αποφασίσουμε αν ένας τύπος σχέσης R είναι σε "καλή" μορφή.
- Το ζητούμενο είναι: **αποσύνθεση χωρίς απώλεια πληροφορίας (lossless decomposition)**
 - Φορμαλιστικά:

$$\forall \text{ σχέση } r \text{ τύπου } R, r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$$

ΒΔ: [4] Θεωρία Κανονικοποίησης

10

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Συναρτησιακές Εξαρτήσεις



- Περιορισμοί πάνω στο σύνολο των έγκυρων σχέσεων
- Απαιτούν η τιμή ορισμένων χαρακτηριστικών να προσδιορίζει μοναδικά την τιμή άλλων χαρακτηριστικών
- Γενίκευση της έννοιας του κλειδιού
- Ορισμός: Έστω R ένας τύπος σχέσης, με $\alpha \subseteq R$ and $\beta \subseteq R$. Η **συναρτησιακή εξάρτηση (functional dependence - FD)** $\alpha \rightarrow \beta$ ισχύει πάνω στον R εάν και μόνο εάν

$$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$

- Παραδείγματα:
 $branch-name \rightarrow assets, branch-city$
 $loan-number \rightarrow amount, branch-name$

Συναρτησιακές Εξαρτήσεις (συν.)



- Το σύνολο χαρακτηριστικών K αποτελεί **υπερκλειδί (superkey)** για τον τύπο σχέσης R εάν και μόνο εάν $K \rightarrow R$
- Το K αποτελεί **υποψήφιο κλειδί (candidate key)** για τον R εάν και μόνο εάν
 - $K \rightarrow R$, και
 - δεν υπάρχει $\alpha \subset K$, $\alpha \rightarrow R$
- Οι συναρτησιακές εξαρτήσεις επιτρέπουν να εκφράσουμε περιορισμούς που δεν μπορούν να εκφραστούν μέσω των υπερκλειδιών.
 - Για παράδειγμα, ας θεωρήσουμε τον τύπο σχέσης:
 $loan-info-schema = (customer-name, loan-number, branch-name, amount)$
 περιμένουμε να ισχύουν αυτές οι συναρτησιακές εξαρτήσεις:
 $loan-number \rightarrow amount$ και $loan-number \rightarrow branch-name$
 αλλά όχι αυτή: $loan-number \rightarrow customer-name$

Συναρτησιακές Εξαρτήσεις (συν.)



- Χρησιμοποιούμε συναρτησιακές εξαρτήσεις για να:
 - Ελέγχουμε την εγκυρότητα μιας σχέσης r σύμφωνα με ένα δοθέν σύνολο συναρτησιακών εξαρτήσεων F (οπότε λέμε ότι η r **ικανοποιεί** την F).
 - Προδιαγράφουμε περιορισμούς F πάνω σε έναν τύπο σχέσης R (οπότε λέμε ότι η F **ισχύει για** τον R)
- Μια συναρτησιακή εξάρτηση καλείται **τετριμμένη** εάν ικανοποιείται οπωσδήποτε σε κάθε στιγμιότυπο ενός τύπου σχέσης. π.χ.
 - $customer-name, loan-number \rightarrow customer-name$
 - Γενικά, $\alpha \rightarrow \beta$ είναι τετριμμένη εάν $\beta \subseteq \alpha$

Κανονικοποίηση με χρήση συναρτησιακών εξαρτήσεων



- Όταν αποσυνθέτουμε ένα τύπο σχέσης R με ένα σύνολο συναρτησιακών εξαρτήσεων F στους τύπους σχέσεων $R_1, R_2, \dots, R_n$ έχουμε 3πλο στόχο:
 - Στόχος 1: **όχι απώλεια πληροφορίας**
 - Η Αποσύνθεση πρέπει να είναι χωρίς απώλεια πληροφορίας (lossless decomposition)
 - Στόχος 2: **όχι πλεονασμός πληροφορίας**
 - Οι τύποι σχέσεων R_i που προκύπτουν από την αποσύνθεση, κατά προτίμηση να είναι σε Boyce-Codd Κανονική Μορφή ή 3^η Κανονική Μορφή (θα τις ορίσουμε σε λίγο ...)
 - Στόχος 3: **διατήρηση εξαρτήσεων**
 - Έστω F_i το σύνολο των εξαρτήσεων που περιέχουν μόνο χαρακτηριστικά του R_i . Η αποσύνθεση επιθυμούμε να διατηρεί τις εξαρτήσεις.

Παράδειγμα αποσύνθεσης



- $R = (A, B, C)$
 $F = \{A \rightarrow B, B \rightarrow C\}$
 - Η αποσύνθεση μπορεί να γίνει με δύο εναλλακτικούς τρόπους
- $R_1 = (A, B), R_2 = (B, C)$
 - Αποσύνθεση με απώλεια πληροφορίας:
 $R_1 \cap R_2 = \{B\}$ and $B \rightarrow BC$
 - Διατήρηση εξαρτήσεων? → Ναι
- $R_1 = (A, B), R_2 = (A, C)$
 - Αποσύνθεση χωρίς απώλεια πληροφορίας :
 $R_1 \cap R_2 = \{A\}$ and $A \rightarrow AB$
 - Διατήρηση εξαρτήσεων? → Όχι
(δεν μπορούμε να ελέγξουμε αν ισχύει $B \rightarrow C$ χωρίς να εκτελέσουμε τη σύνδεση $R_1 \bowtie R_2$)

- Πρώτη Κανονική Μορφή (1NF)
- Αποσύνθεση - Συναρτησιακές Εξαρτήσεις
- ➔ ▪ Δεύτερη (2NF) και Τρίτη Κανονική Μορφή (3NF)
- Boyce-Codd Κανονική Μορφή (BCNF)
- Καθολική Διαδικασία Σχεδίασης ΒΔ



Δεύτερη Κανονική Μορφή



- Ένας τύπος σχέσης R είναι σε **2^η Κανονική Μορφή (2nd Normal Form - 2NF)** εάν είναι σε 1NF και επιπλέον:

- Κανένα χαρακτηριστικό (από αυτά που δεν συμμετέχουν σε υποψήφιο κλειδί) δεν **προσδιορίζεται** από **τμήμα** ενός υποψήφιου κλειδιού K

- Φορμαλιστικά:

$$\nexists (a \rightarrow b) \text{ in FD: } a \subset K \wedge b \notin K$$

- Παράδειγμα:

- $R = (\text{customer-name}, \text{loan-number}, \text{amount})$
- Καταρχήν η R είναι σε 1NF
- Όμως, ισχύει η εξάρτηση $\text{loan-number} \rightarrow \text{amount}$ και το χαρακτηριστικό loan-number είναι μέρος του υποψήφιου κλειδιού $\{\text{customer-name}, \text{loan-number}\}$.
- Άρα η R δεν είναι σε 2NF

Δεύτερη Κανονική Μορφή (συν.)



- Πώς γίνεται αποσύνδεση σε 2NF:
 - Απαλλάσσουμε την R από την 'προβληματική' εξάρτηση, η οποία με τη σειρά της σχηματίζει ένα νέο τύπο σχέσης.

- Παράδειγμα:

- $R = (\text{customer-name}, \text{loan-number}, \text{amount})$
- Η 'προβληματική' εξάρτηση είναι η $\text{loan-number} \rightarrow \text{amount}$
- Άρα: $R_1 = (\text{customer-name}, \text{loan-number})$, $R_2 = (\text{loan-number}, \text{amount})$
- Τόσο η R_1 όσο και η R_2 είναι σε 2NF

Τρίτη Κανονική Μορφή



- Ένας τύπος σχέσης R είναι σε **3^η Κανονική Μορφή (3rd Normal Form - 3NF)** εάν είναι σε 2NF και επιπλέον:
 - Κανένα χαρακτηριστικό (από αυτά που δεν συμμετέχουν σε υποψήφιο κλειδί) δεν **προσδιορίζεται μεταβατικά** από ένα υποψήφιο κλειδί K
 - Η μεταβατική εξάρτηση ισχύει όταν παρεμβάλλεται χαρακτηριστικό από αυτά που δεν συμμετέχουν σε υποψήφιο κλειδί
- Φορμαλιστικά:

$$\nexists (a \rightarrow b), (b \rightarrow c) \text{ in FD: } a = K \wedge b \notin K \wedge c \notin K$$
- Παράδειγμα:
 - $R = (\text{loan-number}, \text{amount}, \text{branch-name}, \text{branch-city})$
 - Καταρχήν η R είναι σε 2NF. Όμως:
 - $\text{loan-number} \rightarrow \text{branch-name}$ και $\text{branch-name} \rightarrow \text{branch-city}$, οπότε το branch-city προσδιορίζεται μεταβατικά από το loan-number μέσω του branch-name .
 - Άρα η R δεν είναι σε 3NF

Τρίτη Κανονική Μορφή (συν.)



- Πώς γίνεται αποσύνδεση σε 3NF:
 - Απαλλάσσουμε την R από την 'προβληματική' εξάρτηση, η οποία με τη σειρά της σχηματίζει ένα νέο τύπο σχέσης.
- Παράδειγμα:
 - $R = (\text{loan-number}, \text{amount}, \text{branch-name}, \text{branch-city})$
 - Η 'προβληματική' εξάρτηση είναι η $\text{branch-name} \rightarrow \text{branch-city}$
 - Άρα: $R_1 = (\text{loan-number}, \text{amount}, \text{branch-name})$, $R_2 = (\text{branch-name}, \text{branch-city})$
 - Τόσο η R_1 όσο και η R_2 είναι σε 3NF
- Αποδεικνύεται ότι **πάντοτε** μπορεί να βρεθεί μια αποσύνθεση σε 3NF που να είναι αποσύνθεση χωρίς απώλεια πληροφορίας και να διατηρεί τις εξαρτήσεις.



- Πρώτη Κανονική Μορφή (1NF)
- Αποσύνθεση - Συναρτησιακές Εξαρτήσεις
- Δεύτερη (2NF) και Τρίτη Κανονική Μορφή (3NF)
- ➔ ▪ Boyce-Codd Κανονική Μορφή (BCNF)
- Καθολική Διαδικασία Σχεδίασης ΒΔ

Boyce-Codd Κανονική Μορφή



- Ένας τύπος σχέσης R είναι σε **Boyce-Codd Κανονική Μορφή (Boyce-Codd Normal Form - BCNF)** εάν για όλες τις συναρτησιακές εξαρτήσεις $\alpha \rightarrow \beta$ που ισχύουν στην R , το α αποτελεί υποψήφιο κλειδί της R
 - Προσοχή: δεν ξεκινάμε τον έλεγχο με την παραδοχή ότι η R είναι σε 3NF
 - Απλά, μπορεί να αποδειχθεί ότι αν η R είναι σε BCNF τότε είναι και σε 3NF.
- Φορμαλιστικά: $\nexists (a \rightarrow b) \text{ in FD: } a \neq K$
- Παράδειγμα:
 - Έστω $R = (A, B, C)$, ισχύουν οι συναρτησιακές εξαρτήσεις $F = \{A \rightarrow B, B \rightarrow C\}$ και $\{A\}$ είναι το κλειδί
 $\Rightarrow$ Η R δεν είναι σε BCNF (μάλιστα, δεν είναι ούτε σε 3NF !!)
 - Έστω ότι γίνεται η αποσύνθεση της R σε $R_1 = (A, B)$, $R_2 = (B, C)$
 $\Rightarrow$ οι R_1 και R_2 είναι σε BCNF

Boyce-Codd Κανονική Μορφή (συν.)



- Πώς γίνεται αποσύνδεση σε 3NF:
 - Κάθε συναρτησιακή εξάρτηση σχηματίζει ένα νέο τύπο σχέσης.
- Παράδειγμα:
 - $R = (branch-name, branch-city, assets, \underline{customer-name}, \underline{loan-number}, amount)$

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

- συναρτησιακές εξαρτήσεις:
 - $branch-name \rightarrow assets, branch-city$
 - $loan-number \rightarrow amount, branch-name$
- Άρα: $R_1 = (branch-name, branch-city, assets)$, $R_2 = (branch-name, \underline{loan-number}, amount)$, $R_3 = (\underline{customer-name}, \underline{loan-number})$

BCNF και Διατήρηση Εξαρτήσεων



- Ιδιότητες της BCNF:
 - Αποσύνθεση χωρίς απώλεια πληροφορίας $\rightarrow$ ΝΑΙ
 - Διατήρηση εξαρτήσεων $\rightarrow$ ΙΣΩΣ (!)
 - Με άλλα λόγια, δεν είναι πάντοτε δυνατό να πάρουμε μια αποσύνθεση BCNF που να διατηρεί τις εξαρτήσεις !!
- Παράδειγμα:
 - $R = (J, K, L)$, $F = \{JK \rightarrow L, L \rightarrow K\}$, δύο υποψήφια κλειδιά JK και JL
 $\Rightarrow R$ είναι σε 3NF αλλά δεν είναι σε BCNF
 - Οποιαδήποτε αποσύνθεση της R αδυνατεί να διατηρήσει την εξάρτηση $JK \rightarrow L$
- Τότε καταφεύγουμε στη λύση της 3NF (μπορεί να αποδειχθεί ότι αν ένας τύπος σχέσης R είναι σε BCNF τότε είναι και σε 3NF)
 - Αυτή η λύση επιτρέπει κάποιες μορφές πλεονασμό (με συνεπαγόμενα προβλήματα όμως)

BCNF και Διατήρηση Εξαρτήσεων (συν.)



- Προβλήματα που μπορεί να προκύψουν λόγω πλεονασμού στην 3NF

$$R = (J, K, L)$$

$$F = \{JK \rightarrow L, L \rightarrow K\}$$

J	L	K
j_1	l_1	k_1
j_2	l_1	k_1
j_3	l_1	k_1
null	l_2	k_2

- Επανάληψη πληροφορίας (π.χ., η συσχέτιση l_1, k_1)
- Ανάγκη για χρήση τιμών null (π.χ., για να αναπαρασταθεί η συσχέτιση l_2, k_2 όταν δεν υπάρχει αντίστοιχη τιμή για το J).

Τέταρτη (4NF) και Πέμπτη (5NF) Κανονική Μορφή



- Με τη χρήση των συναρτησιακών εξαρτήσεων (functional dependencies - FD) μπορούμε να ορίσουμε 2NF, 3NF και BCNF
- Έχουν μελετηθεί και άλλες εξαρτήσεις:
 - πλειότιμες εξαρτήσεις (Multi-Valued Dependencies - MVD),
 - εξαρτήσεις συνένωσης (Join Dependencies),
 - εξαρτήσεις εγκλεισμού (Inclusion Dependencies), ...
- Αυτές οι εξαρτήσεις μας πάνε σε περαιτέρω κανονικές μορφές, 4NF και 5NF
 - Έχουν μελετηθεί θεωρητικά αλλά ...
 - ... δεν είναι στους σκοπούς του μαθήματος !!



- Πρώτη Κανονική Μορφή (1NF)
- Αποσύνθεση - Συναρτησιακές Εξαρτήσεις
- Δεύτερη (2NF) και Τρίτη Κανονική Μορφή (3NF)
- Boyce-Codd Κανονική Μορφή (BCNF)
- ➔ ▪ Καθολική Διαδικασία Σχεδίασης ΒΔ

Καθολική διαδικασία σχεδίασης ΒΔ



- Οι στόχοι στη σχεδίαση σχεσιακών ΒΔ είναι:
 - BCNF
 - Σύνδεση χωρίς απώλεια πληροφορίας
 - Διατήρηση εξαρτήσεων
- Αν δεν μπορούμε να τα πετύχουμε, κάνουμε υποχωρήσεις:
 - Μη διατήρηση κάποιων εξαρτήσεων
 - 3NF αντί BCNF (πλεονασμός ...)
- Ας σημειωθεί ότι η SQL δεν παρέχει κάποιον άμεσο τρόπο να προδιαγραφούν οι συναρτησιακές εξαρτήσεις, πέρα από τους περιορισμούς του κλειδιού.
 - Μπορούν να προδιαγραφούν με assertions, αλλά ο έλεγχός τους κοστίζει

Καθολική διαδικασία σχεδίασης ΒΔ (συν.)



- Κάνουμε την παραδοχή ότι ο τύπος σχέσης R :
 - Θα μπορούσε να προκύψει όταν μετατρέπουμε ένα διάγραμμα E-R σε ένα σύνολο πινάκων.
 - Θα μπορούσε να θεωρηθεί ως μία και μόνη σχέση που περιέχει **όλα** τα χαρακτηριστικά που μας ενδιαφέρουν στη ΒΔ (ονομάζεται **καθολική σχέση - universal relation**).
 - Θα μπορούσε να είναι το αποτέλεσμα κάποιων σχεδιαστικών αποφάσεων στην πορεία ελέγχου ή μετατροπής μιας σχέσης σε κάποια κανονική μορφή

Η προσέγγιση της καθολικής σχέσης



- Η σχέση $r_1 \bowtie r_2 \bowtie \dots \bowtie r_n$ ονομάζεται **καθολική σχέση (universal relation)** καθώς εμπλέκει όλα τα χαρακτηριστικά της ΒΔ όπως ορίζονται από το $R_1 \cup R_2 \cup \dots \cup R_n$
- «**άκυρες**» **πλειάδες** – πλειάδες που “εξαφανίζονται” όταν υπολογίζουμε μια σύνδεση.
 - Έστω $r_1(R_1), r_2(R_2), \dots, r_n(R_n)$ ένα σύνολο από σχέσεις
 - Μια πλειάδα r της σχέσης r_i είναι «άκυρη» πλειάδα εάν η r δεν βρίσκεται μέσα στη σχέση:

$$\prod_{R_i} (r_1 \bowtie r_2 \bowtie \dots \bowtie r_n)$$

Παράδειγμα αποσύνθεσης καθολικής σχέσης



- Έστω σε εννοιολογικό επίπεδο έχουν καταγραφεί τα εξής:
 - Οντότητες: customer (customer-name), loan (loan-number, amount), branch (branch-name, branch-city, assets)
 - Συσχετίσεις: customer-loan (τύπου M:N), loan-branch (τύπου N:1)
- Το σχήμα καθολικής σχέσης που προκύπτει είναι:
 - $U = (\text{branch-name}, \text{branch-city}, \text{assets}, \text{customer-name}, \text{loan-number}, \text{amount})$
- Ένα στιγμιότυπο της σχέσης:

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>	<i>customer-name</i>	<i>loan-number</i>	<i>amount</i>
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

Έλεγχος για 2NF → ??

για 3NF → ??

για BCNF → ??

Μοντέλο E-R και Κανονικοποίηση



- Όταν το διάγραμμα E-R σχεδιάζεται προσεκτικά, με επιτυχή εντοπισμό των οντοτήτων, οι πίνακες που προκύπτουν δεν χρειάζονται περαιτέρω κανονικοποίηση
- Όμως, σε πραγματικές καταστάσεις, μπορούν να υπάρχουν συναρτησιακές εξαρτήσεις από χαρακτηριστικά που δεν είναι κλειδιά προς άλλα χαρακτηριστικά μιας οντότητας
 - π.χ. οντότητα *employee* με χαρακτηριστικά *department-number* και *department-address* και με τη συναρτησιακή εξάρτηση *department-number* → *department-address*
 - Ένας καλός σχεδιασμός θα είχε αναγνωρίσει το department ως οντότητα

Απο-κανονικοποίηση (denormalization) για λόγους απόδοσης



- Μερικές φορές για λόγους απόδοσης προτιμούμε μη κανονικοποιημένα σχήματα
 - π.χ. αν θέλουμε τα *customer-name*, *account-number* και *balance* να είναι μαζί στο αποτέλεσμα μιας ερώτησης απαιτείται *account* ⋈ *depositor*
- 1^ο εναλλακτικό: απο-κανονικοποίηση με τα χαρακτηριστικά των σχέσεων να είναι μαζί σε μία (μη κανονικοποιημένη) σχέση
 - + μεγαλύτερη ταχύτητα στις ερωτήσεις
 - μεγαλύτερες απαιτήσεις χώρου και χρόνου εκτέλεσης των αλλαγών στη ΒΔ, κίνδυνος σφαλμάτων
- 2^ο εναλλακτικό : ορισμός μιας υλοποιημένης θεώρησης (materialized view) *account* ⋈ *depositor*
 - τα ίδια υπέρ και κατά όπως πριν, εκτός του κινδύνου σφαλμάτων

Άλλα σχεδιαστικά ζητήματα



- Υπάρχουν προβλήματα που δεν οφείλονται στη (μη) κανονικοποίηση αλλά σε κακή σύλληψη:
- Παράδειγμα: αντί του *earnings(company-id, year, amount)*, να έχουμε
 - Πολλούς πίνακες: *earnings-2000*, *earnings-2001*, *earnings-2002*, κλπ., όλοι του τύπου (*company-id, earnings*).
 - Είναι μεν σε BCNF, αλλά καθιστούν δύσκολη την εκτέλεση ερωτήσεων όταν εμπλέκονται πολλά έτη και επιπλέον απαιτείται ένας νέος πίνακας κάθε έτος!
 - *company-year(company-id, earnings-2000, earnings-2001, earnings-2002)*
 - Επίσης σε BCNF, αλλά καθιστούν δύσκολη την εκτέλεση ερωτήσεων όταν εμπλέκονται πολλά έτη και επιπλέον απαιτείται ένα νέο χαρακτηριστικό κάθε έτος!
 - Αποτελεί κλασικό παράδειγμα λειτουργίας **crosstab**, όπου οι τιμές ενός χαρακτηριστικού γίνονται στήλες
 - Χρησιμοποιείται σε spreadsheets και σε εργαλεία ανάλυσης δεδομένων