Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Μεταφορά δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων

Χριστίνα Δ. Γκριτζάπη

Μεταπτυχιακή Εργασία

Ηράκλειο, Νοέμβριος 1995

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημων
Τμήμα Επιστήμης Υπολογιστών

Μεταφορά δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων

Εργασία που υποβλήθηκε από την
Χριστίνα Δ. Γκριτζάπη
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Χριστίνα Δ. Γκριτζάπη
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Πάνος Κωνσταντόπουλος,Αναπληρωτής Καθηγητής, Επόπτης

Γεώργιος Γεωργακόπουλος,Επίκουρος Καθηγητής, Μέλος

Πάνος Τραχανιάς,Επίκουρος Καθηγητής, Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος
Συντονιστής Μεταπτυχιακών Σπουδών

Ηράκλειο, Νοέμβριος 1995

Μεταφορά δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων

Χριστίνα Δ. Γκριτζάπη

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

ΠΕΡΙΛΗΨΗ

Ενας από τους ενεργούς τομείς στην περιοχή των βάσεων δεδομένων είναι οι ετερογενείς βάσεις δεδομένων. Ετερογενείς βάσεις δεδομένων λέγονται οι βάσεις δεδομένων που διαφέρουν ως προς τα συστήματα διαχείρισής τους, το σχήμα, τους τύπους των δεδομένων τους κ.τ.λ. Ενα παράδειγμα ετερογενών βάσεων δεδομένων είναι οι σχεσιακές και οι οντοκεντρικές. Η κάθε κατηγορία προσφέρει μοναδικά πλεονεκτήματα και έχουν κατασκευαστεί πολλά εμπορικά συστήματα που στηρίζονται στη μία ή στην άλλη. Επειδή συχνά παρουσιάζεται η ανάγκη να συνεργαστούν τέτοιου είδους συστήματα, είναι επιθυμητός ένας γενικός τρόπος ανταλλαγής δεδομένων ανάμεσα σε μια σχεσιακή και μια οντοκεντρική βάση δεδομένων.

Αντικείμενο της εργασίας αυτής είναι το πρόβλημα της μεταφοράς δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων. Ειδικότερα αναπτύσσεται ένα σύστημα για μεταφορά δεδομένων από μια οποιαδήποτε σχεσιακή βάση σε μια οντοκεντρική βάση που βασίζεται στο μοντέλο δεδομένων της γλώσσας παράστασης γνώσης Telos. Δύο είναι τα κύρια αποτελέσματα της εργασίας αυτής :
Πρώτον, η εύρεση μεθόδου καθορισμού απεικονίσεων από το σχεσιακό στο οντοκεντρικό σχήμα. Οι απεικονίσεις αυτές είναι πολύ σημαντικές γιατί είναι απαραίτητο να γνωρίζουμε **ποιά** δεδομένα θα μεταφερθούν από τη σχεσιακή βάση και **που** θα αποθηκευτούν στην οντοκεντρική.
Δεύτερον, η κατασκευή ενός συστήματος που μεταφέρει δεδομένα από μια σχεσιακή σε μια οντοκεντρική βάση χρησιμοποιώντας τις απεικονίσεις που ορίστηκαν.

Επόπτης : Πάνος Κωνσταντόπουλος
Αναπληρωτής Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Data transfer from a relational to an object-oriented database

Christina D. Gritzapi

Master of Science Thesis

Department of Computer Science
University of Crete

ABSTRACT

Nowadays, an active topic in the databases area is heterogeneous databases. Heterogeneous databases are called the databases which differ in terms of management systems, schema, data types etc. An example of heterogeneous databases is relational and object-oriented databases. Each category has unique advantages and many commercial systems, which are based on either of them,have been constructed. There is often a need for such kind of systems to cooperate , so a general method for data exchange among a relational and an object-oriented database is desired .

The topic of this thesis is the problem of data transfer from a relational to an object-oriented database. In particularly a system is developed for data transfer from a relational database to an object-oriented database based on the data model of the Telos knowledge representation language. Two are the main results of this thesis :

First, the development of a method for the definition of mappings from a relational to an object-oriented schema. These mappings are very important because it is necessary to know **which** data are going to be transfered from the relational database and **where** they will be stored in the object-oriented database.

Second, the development of a system which transfers data from a relational to an object-oriented database using the defined mappings.

Supervisor : Panos Constantopoulos
Associate Professor of Computer Science
University of Crete

Ευχαριστίες

Ευχαριστώ τον επόπτη καθηγητή μου κ. Πάνο Κωνσταντόπουλο που μου εμπιστεύθηκε αυτή την εργασία, καθώς και για την καθοδήγηση και τις συμβουλές του.

Θα ήθελα να ευχαριστήσω όλα τα μέλη της Ομάδας Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για τη συνεργασία και τη συμπαράστασή τους.

Επίσης ευχαριστώ το Ινστιτούτο Πληροφορικής για την υλικοτεχνική υποδομή και την οικονομική ενίσχυση που μου παρείχε κατά τη διάρκεια των μεταπτυχιακών μου σπουδών.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τους γονείς μου για τη βοήθεια και την συμπαράστασή τους σε όλη τη διάρκεια των σπουδών μου.

Ακόμη, τον Δημήτρη Δασκαλάκη για τη συμπαράσταση του κατά τη διάρκεια των μεταπτυχιακών μου σπουδών, καθώς και τους φίλους μου Δέσποινα Βαμβακά και Αναστάσιο Στογιαννίδη για την πολύτιμη βοήθειά τους.

Περιεχόμενα

Περίληψη **i**

Abstract **ii**

Ευχαριστίες **iii**

1 Εισαγωγή **1**

1.1 Το πρόβλημα 1

1.2 Αντικείμενο της εργασίας 2

1.3 Εφαρμογή σε ένα ολοκληρωμένο πληροφοριακό σύστημα μουσείων 4

1.4 Η οργάνωση της εργασίας 4

2 Ετερογενείς βάσεις δεδομένων **7**

2.1 Ορισμοί 7

2.2 Ταξινόμηση ετερογενών βάσεων 8

2.3 Αρχιτεκτονική ομοσπονδιακών συστημάτων 11

2.4 Προβλήματα συνδυασμού ετερογενών βάσεων 12

2.5 Κοινωνία δεδομένων 13

2.6 Συστήματα για ετερογενείς βάσεις δεδομένων 14

3 Σύστημα μεταφοράς δεδομένων **17**

3.1 Ορισμός προβλήματος 17

3.2 Η ιδέα του συστήματος 18

3.2.1 Στάδιο αρχικοποίησης συστήματος 19

3.2.2 Στάδιο μεταφοράς δεδομένων 19

3.3 Αρχιτεκτονική του συστήματος 20

3.3.1 Μηχανισμός αρχικοποίησης οντοκεντρικής βάσης 22

3.3.1.1 Συνέπεια βάσεων κατά την επικοινωνία 23

3.3.2 Μηχανισμός καθορισμού σχεσιακού σχήματος 24

3.3.2.1 Ο Μηχανισμός καθορισμού σχεσιακού σχήματος στα πλαίσια της επικοινωνίας MITOS/CM-ΚΛΕΙΩ . . . 26
3.3.3 Μηχανισμός εξαγωγής δεδομένων . . . 28
Αρχείο δεδομένων . . . 29
3.3.4 Μηχανισμός καθορισμού απεικονίσεων . . . 29
3.3.5 Μηχανισμός μετάφρασης και μεταφοράς δεδομένων . . . 30
3.3.5.1 Λόγοι αποτυχίας της μεταφοράς των δεδομένων . . . 32
3.3.5.2 Σύγχρονη ή ασύγχρονη επικοινωνία . . . 33
3.3.6 Μηχανισμός ανάκτησης δεδομένων οντοκεντρικής βάσης . . . 33
Ολική ανάκτηση δεδομένων . . . 34
Μερική ανάκτηση δεδομένων . . . 34
3.4 Διαδικασία επικοινωνίας . . . 35
Φάση I - Ορισμός σχεσιακού σχήματος . . . 35
Φάση II - Καθορισμός απεικονίσεων . . . 35
Φάση III - Αρχικοποίηση της οντοκεντρικής βάσης . . . 35
Φάση IV - Μεταφορά δεδομένων από τη σχεσιακή στην οντοκεντρική βάση 35
Φάση V - Ανάκτηση δεδομένων της οντοκεντρικής βάσης . . . 36

4 Τελεστές απεικόνισης 37
4.1 Το πρόβλημα της μετάφρασης σχημάτων . . . 37
4.1.1 Προτάσεις από τη βιβλιογραφία . . . 38
4.1.2 Ισοδυναμία σχημάτων . . . 41
4.2 Εισαγωγικές έννοιες για τον ορισμό των τελεστών απεικόνισης . . . 41
4.2.1 Αποθήκευση σχεσιακού σχήματος στην οντοκεντρική βάση . . . 43
4.2.2 Χρήση προθήματος και επιθήματος κατά τον ορισμό των απεικονίσεων . . . 45
4.2.3 Χρήση της έννοιας της συνθήκης κατά τον ορισμό των απεικονίσεων . . . 46
4.3 Ορισμός τελεστών απεικόνισης . . . 47
4.3.1 Τελεστής BECOMES . . . 49
4.3.2 Τελεστής ISA . . . 52
4.3.3 Τελεστής BECOMES_LINK . . . 55
4.3.4 Τελεστής BECOMES_DEPENDENT_OBJECT . . . 57
4.4 Ορισμός απεικονίσεων μέσω ξένων κλειδιών . . . 60
4.5 Ονοματοδοσία . . . 62
4.6 Περιορισμοί . . . 64
4.6.1 Μορφή σχεσιακού σχήματος . . . 64
4.6.2 Εφαρμογή τελεστών σε πίνακες χωρίς κύριο κλειδί . . . 66

5 Μετάφραση πράξεων στα δεδομένα **69**

5.1 Εισαγωγή . . . 70

5.1.1 Η εισαγωγή σε σχέση με την απεικόνιση BECOMES . . . 70

5.1.2 Η εισαγωγή σε σχέση με την απεικόνιση ISA . . . 70

5.1.3 Η εισαγωγή σε σχέση με την απεικόνιση BECOMES_LINK . . . 70

5.1.4 Η εισαγωγή σε σχέση με την απεικόνιση BECOMES_DEPENDENT_OBJECT . . . 70

5.2 Διαγραφή . . . 71

5.2.1 Η διαγραφή σε σχέση με την απεικόνιση BECOMES . . . 71

5.2.2 Η διαγραφή σε σχέση με την απεικόνιση ISA . . . 72

5.2.3 Η διαγραφή σε σχέση με την απεικόνιση BECOMES_LINK . . . 72

5.2.4 Η διαγραφή σε σχέση με την απεικόνιση BECOMES_DEPENDENT_OBJECT . . . 72

5.3 Ενημέρωση . . . 72

5.3.1 Η ενημέρωση σε σχέση με την απεικόνιση BECOMES . . . 73

5.3.2 Η ενημέρωση σε σχέση με την απεικόνιση ISA . . . 74

5.3.3 Η ενημέρωση σε σχέση με την απεικόνιση BECOMES_LINK . . . 75

5.3.4 Η ενημέρωση σε σχέση με την απεικόνιση BECOMES_DEPENDENT_OBJECT . . . 75

5.4 Θέματα συνέπειας δεδομένων . . . 76

5.4.1 Σειρά εισαγωγής δεδομένων στην οντοκεντρική βάση . . . 76

5.4.2 Ελεγχόμενο λεξιλόγιο . . . 77

6 Βοήθεια προς το χρήστη για καθορισμό απεικονίσεων **79**

6.1 Αντληση πληροφορίας για καθορισμό απεικονίσεων με τη βοήθεια του σχεσιακού και του οντοκεντρικού σχήματος . . . 79

7 Επίλογος **83**

7.1 Συμπεράσματα . . . 83

7.2 Επεκτάσεις-Βελτιώσεις . . . 84

A Ευρετήριο όρων **87**

B Ορισμός σχεσιακού σχήματος **91**

B.1 Συντακτικό ορισμού σχεσιακού σχήματος . . . 91

B.2 Μοντέλο αποθήκευσης σχεσιακού σχήματος στην Telos . . . 92

Γ Ορισμός απεικονίσεων **95**

Γ.1 Συντακτικό ορισμού απεικονίσεων . . . 95

Γ.2 Μοντέλο αποθήκευσης απεικονίσεων σε Telos . . . 97

Βιβλιογραφία **98**

Κατάλογος Σχημάτων

2.1 Ομοσπονδιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων 8
2.2 Ταξινόμηση πολυβάσεων δεδομένων ως προς την αυτονομία συσχέτισης των βάσεων-συνιστωσών τους 11
2.3 Αρχιτεκτονική 5-επιπέδων ενός Ομοσπονδιακού Συστήματος Βάσεων Δεδομένων 12

3.1 Μεταφορέας δεδομένων 18
3.2 Αρχιτεκτονική Συστήματος 21
3.3 Μηχανισμός αρχικοποίησης της οντοκεντρικής βάσης 22
3.4 Μηχανισμός καθορισμού σχεσιακού σχήματος 24
3.5 Μηχανισμός εξαγωγής δεδομένων 28
3.6 Μηχανισμός καθορισμού απεικονίσεων 30
3.7 Μηχανισμός μετάφρασης και μεταφοράς δεδομένων 31
3.8 Μηχανισμός ανάκτησης δεδομένων της οντοκεντρικής βάσης 34
3.9 Διάγραμμα λειτουργίας συστήματος μεταφοράς δεδομένων 36

4.1 Μοντέλο αποθήκευσης σχεσιακού σχήματος στην οντοκεντρική βάση 44
4.2 Μοντέλο αποθήκευσης απεικονίσεων στην οντοκεντρική βάση 47
4.3 Παράδειγμα (1) χρήσης τελεστή BECOMES 51
4.4 Παράδειγμα (2) χρήσης τελεστή BECOMES 52
4.5 Σχέση ευρύτερου-στενότερου όρου στην οντοκεντρική βάση 54
4.6 Παράδειγμα χρήσης τελεστή ISA 55
4.7 Σχέση ανάμεσα στην κλάση Αυτοκίνητο και την κλάση Χρώμα στην οντοκεντρική βάση 57
4.8 Παράδειγμα χρήσης τελεστή BECOMES_LINK 58
4.9 Γενέθλια 60
4.10 Παράδειγμα χρήσης τελεστή BECOMES_DEPENDENT_OBJECT 61
4.11 Παράδειγμα μεταφοράς δεδομένων με χρήση ξένου κλειδιού 63
4.12 Χρονολογίες στο ΚΛΕΙΩ 65
4.13 Σχέσεις κατασκευής και χρήσης μεταξύ αντικειμένων και χρονολογιών στο ΚΛΕΙΩ . . . 65
4.14 Μοντέλο Δοσοληψίας στην οντοκεντρική βάση 67

6.1 Συνοπτική άποψη της διαδικασίας καθορισμού απεικονίσεων 80

Κεφάλαιο 1

Εισαγωγή

Το κεφάλαιο αυτό είναι οργανωμένο ως εξής. Στην ενότητα 1.1 παρουσιάζεται το πρόβλημα που μας απασχολεί στην παρούσα εργασία. Στην ενότητα 1.2 παρουσιάζεται συνοπτικά το τι ακριβώς έγινε στην εργασία αυτή. Τέλος, στην ενότητα 1.3 παρουσιάζεται η εφαρμογή των αποτελεσμάτων, της παρούσας εργασίας, σε ένα ολοκληρωμένο πληροφοριακό σύστημα μουσείων.

1.1 Το πρόβλημα

Ενας από τους ενεργούς τομείς στην περιοχή των βάσεων δεδομένων σήμερα είναι οι *ετερογενείς βάσεις δεδομένων* [62]. Μια περίπτωση *ετερογενών βάσεων* είναι οι σχεσιακές σε σχέση με τις οντοκεντρικές βάσεις δεδομένων διότι έχουν πολλών ειδών διαφορές, όπως στο σχήμα, στα συστήματα διαχείρισης τους, στους τύπους των δεδομένων τους κ.τ.λ. Οι σχεσιακές βάσεις δεδομένων προσφέρουν μοναδικά πλεονεκτήματα, όπως ώριμη τεχνολογία, που έχει σχεδόν σταθεροποιηθεί. Έχουν όμως και μειονεκτήματα, όπως το ότι οι δομές, που χρησιμοποιούνται για την παράσταση της πληροφορίας, και οι πράξεις που υποστηρίζονται, είναι πολύ απλές και στις περισσότερες περιπτώσεις υπάρχει πληροφορία, που δεν είναι δυνατόν να απεικονιστεί στο σχήμα, αλλά τη γνωρίζει ο κατασκευαστής του. Τέτοιου είδους πληροφορίες είναι οι σχέσεις γενίκευσης/εξειδίκευσης και οι σχέσεις ταξινόμησης πολλαπλών επιπέδων.

Οι σχέσεις αυτές μπορούν να παρασταθούν με ευκολία σε οντοκεντρικές βάσεις δεδομένων. Επίσης οι οντοκεντρικές βάσεις δεδομένων έχουν άλλα πλεονεκτήματα, όπως γρήγορη διάσχιση μεταξύ των δομών τους και υποστήριξη περισσότερων τύπων δεδομένων.

Οι συμπληρωματικές δυνάμεις των σχεσιακών και των οντοκεντρικών βάσεων δεδομένων οδηγούν στο να είναι επιθυμητός ένας γενικός τρόπος ανταλλαγής δεδομένων ανάμεσα τους. Η ανταλλαγή δεδομένων ανάμεσα τους ανήκει στο γενικότερο πρόβλημα της *κοινωνίας δεδομένων* ετερογενών βάσεων. Ενα υποπρόβλημα του προβλήματος αυτού είναι η *μεταφορά δεδομένων* από μια σχεσιακή σε μια οντοκεντρική βάση.

1.2 Αντικείμενο της εργασίας

Αντικείμενο της εργασίας αυτής είναι η ανάπτυξη ενός συστήματος που θα προσφέρει λύση στο πρόβλημα της μεταφοράς δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων. Ειδικότερα, η εργασία στοχεύει να υποστηρίξει την μεταφορά δεδομένων ανάμεσα σε μια οποιαδήποτε σχεσιακή βάση και μια οντοκεντρική βάση που βασίζεται στο μοντέλο δεδομένων της γλώσσας παράστασης γνώσης Telos. Τα σχήματα των δύο βάσεων υποθέτουμε ότι προϋπάρχουν και περιγράφουν τον ίδιο κόσμο ή ότι ο κόσμος που περιγράφει το ένα είναι υποσύνολο του κόσμου που περιγράφει το άλλο[1]. Ο στόχος είναι να ενημερώνεται η οντοκεντρική βάση με τις αλλαγές (εισαγωγές-ενημερώσεις-διαγραφές) που γίνονται στα δεδομένα της σχεσιακής βάσης[2]. Την μεταφορά των δεδομένων από την μια βάση στην άλλη μπορούμε να την δούμε σαν ένα είδος επικοινωνίας των δύο βάσεων. Η επικοινωνία αυτή θα είναι *ασύγχρονη*, δηλαδή οι αλλαγές στα δεδομένα της σχεσιακής βάσης δεν θα μεταφέρονται αυτόματα στην οντοκεντρική αλλά κατόπιν αιτήσεως του χρήστη. Ετσι οι δύο βάσεις θα περιέχουν την ίδια πληροφορία για τα κοινά μέρη τους, μόνο σε κάποιες χρονικές στιγμές που θα αποφασίζονται από το χρήστη. Το σύστημα ανήκει στην κατηγορία των *ομοσπονδιακών συστημάτων διαχείρισης βάσεων δεδομένων* και συγκεκριμένα είναι ένα *χαλαρά-συνδεδεμένο ομοσπονδιακό σύστημα διαχείρισης βάσεων δεδομένων* [62].

Κατά την ανάπτυξη του συστήματος αντιμετωπίστηκαν προβλήματα όπως :

- Η εύρεση μεθόδου με την οποία θα μεταφέρονται τα δεδομένα στην σωστή θέση στην οντοκεντρική βάση, μιας και δεν μπορεί να μεταφέρονται τυχαία. Η μέθοδος αυτή πρέπει να μεταφέρει σωστά τα δεδομένα και όλες τις πιθανές σχέσεις ανάμεσα τους στα αντίστοιχα δεδομένα και σχέσεις της οντοκεντρικής βάσης.

- Η ανάπτυξη μιας αρχιτεκτονικής για το σύστημα μεταφοράς δεδομένων, έτσι ώστε να είναι ανεξάρτητο από την σχεσιακή βάση, από όπου αντλούνται τα δεδομένα.

- Η εύρεση τρόπου εξαγωγής των αλλαγών, που συμβαίνουν στα δεδομένα της σχεσιακής βάσης έτσι ώστε να εφαρμόζονται με την ίδια ακριβώς σειρά στα δεδομένα της οντοκεντρικής βάσης.

Ενα σημαντικό μέρος αυτής της εργασίας, όπως αναφέρθηκε, είναι η εύρεση μεθόδου καθορισμού απεικονίσεων από το σχεσιακό στο οντοκεντρικό σχήμα. Ο ορισμός των απεικονίσεων αυτών είναι πολύ σημαντικός για την επίτευξη της μεταφοράς δεδομένων από τη σχεσιακή στην οντοκεντρική βάση. Αυτό γιατί είναι απαραίτητο να γνωρίζουμε *ποιά* δεδομένα θα μεταφερθούν από τη σχεσιακή βάση και *που* θα πάνε στην οντοκεντρική, μια και τα δεδομένα δεν μπορούν να μεταφέρονται τυχαία από την μια βάση στην άλλη. Γενικά το πρόβλημα της εύρεσης απεικονίσεων μεταξύ σχημάτων είναι συνδεδεμένο με το πρόβλημα της μετάφρασης ενός σχήματος Σ1, εκφρασμένου σε ένα μοντέλο Μ1, σε

[1] οποιοδήποτε από τα δύο σχήματα μπορεί να είναι υποσύνολο του άλλου

[2] το αντίθετο πρόβλημα δεν αντιμετωπίζεται στα πλαίσια της παρούσας εργασίας

ένα σχήμα Σ2 εκφρασμένου σε ένα άλλο μοντέλο Μ2. Το πρόβλημα της μετάφρασης σχημάτων είναι ένα από τα κυριότερα προβλήματα στο χώρο των *ετερογενών βάσεων δεδομένων*. Τα προβλήματα, που παρουσιάζονται σε τέτοιου είδους μεταφράσεις ή απεικονίσεις είναι πολλά και οφείλονται κυρίως στα διάφορα είδη ετερογένειας, που υπάρχουν ανάμεσα στα σχήματα.

Σε αυτή την εργασία για να δημιουργηθούν οι απεικονίσεις μεταξύ των δύο σχημάτων ορίζεται ένα σύνολο τεσσάρων τελεστών απεικόνισης. Οι τελεστές αυτοί παίρνουν σαν είσοδο δομές του σχεσιακού σχήματος και τις απεικονίζουν σε δομές του οντοκεντρικού. Οι τελεστές προσφέρουν δυνατότητες απεικόνισης

1. ανεξάρτητων οντοτήτων της σχεσιακής βάσης σε αντίστοιχες, ανεξάρτητες οντότητες της οντοκεντρικής βάσης (BECOMES) ,

2. ανεξάρτητων σύνθετων οντοτήτων[3] της σχεσιακής βάσης σε μονοπάτια από ανεξάρτητες οντότητες και συνδέσμους της οντοκεντρικής βάσης (BECOMES_DEPENDENT_OBJECT),

3. σχέσεων τύπου υποσυνόλου, που ισχύουν ανάμεσα σε ανεξάρτητες οντότητες του σχεσιακού σχήματος στις ανάλογες σχέσεις τύπου γενίκευσης/εξειδίκευσης της οντοκεντρικής βάσης (ISA) και

4. σχέσεων οντότητας-γνωρίσματος, που ισχύουν ανάμεσα σε οντότητες της σχεσιακής βάσης στις ανάλογες σχέσεις τύπου οντότητας-γνωρίσματος της οντοκεντρικής βάσης (BECOMES_LINK)

Οι δυνατότητες για απεικονίσεις που προσφέρουν οι τελεστές που αναφέρθηκαν καλύπτουν ευρύ φάσμα των αναγκών που παρουσιάζονται κατά τον ορισμό απεικονίσεων από ένα σχεσιακό σε ένα οντοκεντρικό σχήμα. Παρόλα αυτά το σύνολο των τελεστών θα μπορούσε να εμπλουτιστεί και με άλλους τελεστές όπως θα δούμε και στην ενότητα 7.2. Στην προσέγγιση, που προτείνουμε, τις απεικονίσεις τις ορίζει ο ίδιος ο χρήστης στηριζόμενος στη γνώση του για το σχεσιακό και το οντοκεντρικό σχήμα. Οι απεικονίσεις αυτές δεν επιφέρουν αλλαγή σε κανένα από τα δύο αρχικά σχήματα[4] και προσφέρουν τη δυνατότητα μερικής ή ολικής απεικόνισης. Τέλος, πρέπει να τονιστεί ότι οι τελεστές αυτοί πρέπει να εφαρμόζονται σε σχεσιακά σχήματα, που βρίσκονται τουλάχιστον σε τρίτη κανονική μορφή (3NF). Η εφαρμογή των τελεστών σε σχήματα, που δεν είναι σε τέτοια μορφή, δεν θα έχει τα επιθυμητά αποτελέσματα. Βέβαια στην εργασία αυτή γίνεται κάποια πρόβλεψη για αντιμετώπιση μη-κανονικών σχημάτων, η οποία όμως είναι εξειδικευμένη και δεν προσφέρει λύσεις σε πολλές περιπτώσεις.

Συνοψίζοντας, τα αποτελέσματα αυτής της εργασίας είναι : η ανάπτυξη ενός συνόλου τελεστών απεικόνισης για τον καθορισμό απεικονίσεων από ένα σχεσιακό σε ένα οντοκεντρικό σχήμα και

[3] Με τον όρο σύνθετες οντότητες εννοούμε οντότητες, που αποτελούνται από πολλά τμήματα π.χ ημερομηνίες.

[4] Για να μπορούν να οριστούν απεικονίσεις ανάμεσα στα δύο σχήματα πρέπει να υπάρχουν ήδη ορισμένες σε αυτά οι δομές που θα αντιστοιχιστούν. Για παράδειγμα αν στο ένα σχήμα υπάρχει η δομή *Διεύθυνση* και στο άλλο δεν υπάρχει η ανάλογη δομή καμία απεικόνιση δεν μπορεί να οριστεί.

η κατασκευή ενός συστήματος μεταφοράς δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων, στο μοντέλο δεδομένων της γλώσσας παράστασης γνώσης Telos.

1.3 Εφαρμογή σε ένα ολοκληρωμένο πληροφοριακό σύστημα μουσείων

Το σύστημα εφαρμόστηκε στην μεταφορά δεδομένων από το σύστημα *MITOS/CM* στο σύστημα *ΚΛΕΙΩ*. Το *ΜΙΤΟΣ* είναι ένα ολοκληρωμένο πληροφοριακό σύστημα μουσείων, το οποίο αναπτύχθηκε[5] από το ΙΤΕ, το Μουσείο Μπενάκη και την εταιρεία Epsilon Software A.E. Το *ΜΙΤΟΣ* έχει δύο υποσυστήματα το *MITOS/CM* και το *ΚΛΕΙΩ*.

Το *MITOS/CM* είναι το υποσύστημα του *ΜΙΤΟΣ* για μηχανογράφηση της διαχείρισης συλλογών και αναπτύχθηκε κυρίως σε συνεργασία με την εταιρεία Epsilon Software A.E. και στηρίζεται στο σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων SYBASE. Στη σημερινή του μορφή, το σύστημα υποστηρίζει διαδικασίες ελέγχου θέσης, μετακίνησης και έκθεσης των αντικειμένων, όπως επίσης και την εισαγωγή και αλλαγή βασικών πληροφοριών αντικειμένων και την επιλογή τους με συγκεκριμένα κριτήρια.

Το *ΚΛΕΙΩ* [3] είναι ένα σύστημα πολιτισμικής τεκμηρίωσης στο οποίο παριστάνονται έννοιες ύλης, χρόνου, χώρου,υπόστασης, ποσότητας, ανθρωπολογικές και πνευματικής δημιουργίας καθώς και οι μεταξύ τους σχέσεις. Έχει υλοποιηθεί σε γλώσσα Telos, από την Μαρία Χριστοφοράκη στα πλαίσια της μεταπτυχιακής της εργασίας στο Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας (Ι.Τ.Ε.). Η έκδοση της γλώσσας Telos [5], που χρησιμοποιήθηκε, αναπτύχθηκε από το Ινστιτούτο Πληροφορικής του Ι.Τ.Ε [43], [17], [2].

1.4 Η οργάνωση της εργασίας

Στο δεύτερο κεφάλαιο γίνεται ανασκόπηση στη βιβλιογραφία πάνω στο θέμα των ετερογενών βάσεων δεδομένων και των προβλημάτων, που παρουσιάζονται σε αυτές, ένα από τα οποία είναι και η κοινωνία δεδομένων τους.

Στο τρίτο κεφάλαιο παρουσιάζεται το σύστημα, που κατασκευάστηκε στα πλαίσια της εργασίας, για μεταφορά δεδομένων από μία σχεσιακή σε μία οντοκεντρική βάση δεδομένων.

Στο τέταρτο κεφάλαιο ορίζεται ένα σύνολο τελεστών απεικόνισης, με τους οποίους μπορούν να οριστούν απεικονίσεις ανάμεσα σε ένα σχεσιακό και ένα οντοκεντρικό σχήμα.

Στο πέμπτο κεφάλαιο παρουσιάζεται το πως ακριβώς μεταφράζονται οι διάφορες πράξεις, που εφαρμόζονται στα δεδομένα της σχεσιακής βάσης κατά τη μεταφορά τους στην οντοκεντρική, από το σύστημα μεταφοράς δεδομένων.

[5]στο πλαίσιο του προγράμματος STRIDE (1992-1993), έργο ''Ελληνική Δράση για Συστήματα και Εφαρμογές Πολυμέσων'' δραστηριότητα ''Πληροφοριακό Σύστημα Μουσείου''

Στο έκτο κεφάλαιο δίνονται κάποιες γενικές οδηγίες, που θα βοηθήσουν το χρήστη στην προσπάθεια του να εντοπίσει το σύνολο των απεικονίσεων, που είναι κατάλληλο για τις εφαρμογές του.

Στο έβδομο κεφάλαιο υπάρχει ο επίλογος και προτάσεις για επεκτάσεις και βελτιώσεις του συστήματος μεταφοράς δεδομένων και του συνόλου των τελεστών.

Στο παράρτημα Α περιέχεται ένα αγγλο-ελληνικό ευρετήριο όρων, που χρησιμοποιούνται στην εργασία.

Στο παράρτημα Β παρουσιάζεται το συντακτικό με το οποίο ο χρήστης, μπορεί να ορίσει το σχεσιακό σχήμα, που θα χρησιμοποιήσει για να κάνει τις απεικονίσεις με το οντοκεντρικό σχήμα.

Τέλος, στο παράρτημα Γ παρουσιάζεται το συντακτικό με το οποίο ο χρήστης, μπορεί να ορίσει τις απεικονίσεις ανάμεσα στο σχεσιακό και το οντοκεντρικό σχήμα.

Κεφάλαιο 2

Ετερογενείς βάσεις δεδομένων

Στο κεφάλαιο αυτό γίνεται μια ανασκόπηση της βιβλιογραφίας πάνω στο θέμα των ετερογενών βάσεων δεδομένων. Η ανασκόπηση αυτή είναι απαραίτητη για να οριοθετηθεί ο χώρος μέσα στον οποίο κινείται αυτή η εργασία και να καταλάβουμε ποιό ακριβώς είναι το πρόβλημα που θα μας απασχολήσει στα επόμενα κεφάλαια.

Το κεφάλαιο αυτό είναι οργανωμένο ως εξής. Στην ενότητα 2.1 δίνονται ορισμοί γύρω από τις ετερογενείς βάσεις δεδομένων. Στην ενότητα 2.2 παρουσιάζονται διάφορες ταξινομήσεις ετερογενών βάσεων δεδομένων ως προς ορισμένα χαρακτηριστικά τους. Στην ενότητα 2.3 παρουσιάζεται η αρχιτεκτονική των ομοσπονδιακών συστημάτων διαχείρισης βάσεων δεδομένων, τα οποία είναι μια υποκατηγορία των ετερογενών βάσεων δεδομένων. Στην ενότητα 2.4 παρουσιάζονται προβλήματα που εμφανίζονται κατά το συνδυασμό ετερογενών βάσεων δεδομένων. Στην ενότητα 2.5 παρουσιάζεται το πρόβλημα κοινωνίας δεδομένων ετερογενών βάσεων. Τέλος στην ενότητα 2.6 παρουσιάζονται συνοπτικά μερικά συστήματα για ετερογενείς βάσεις δεδομένων.

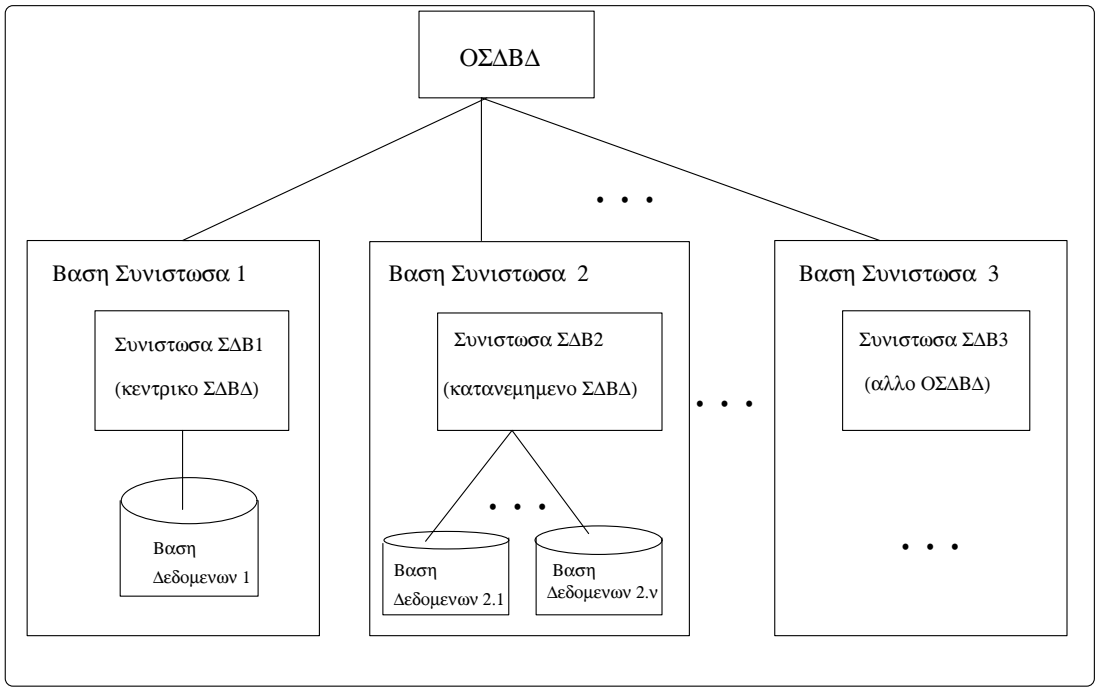
2.1 Ορισμοί

Με τον όρο *ετερογενείς βάσεις δεδομένων* εννοούμε βάσεις που διαφέρουν μεταξύ τους στο μοντέλο, στην ερωτηματική γλώσσα ή στο σχήμα [62],[24],[61],[59],[57],[51].

Μια *ομοσπονδιακή βάση δεδομένων* (ΟΒΔ) είναι μια συλλογή από αυτόνομες, ετερογενείς βάσεις δεδομένων που συνεργάζονται.

Το λογισμικό που παρέχει συντονισμό των βάσεων που συμμετέχουν ονομάζεται *ομοσπονδιακό σύστημα διαχείρισης βάσεων δεδομένων* (ΟΣΔΒΔ) και η μορφή του φαίνεται στο Σχήμα 2.1. Το κυριότερο χαρακτηριστικό μιας *ομοσπονδίας συστημάτων βάσεων δεδομένων* είναι η συνεργασία ανάμεσα σε ανεξάρτητα συστήματα.

Κάθε βάση που συμμετέχει στην ομοσπονδία ονομάζεται *βάση-συνιστώσα*. Μια βάση-συνιστώσα της ομοσπονδίας μπορεί να συμμετέχει σε περισσότερες από μια ομοσπονδίες και το διαχειριστικό



Σχήμα 2.1: Ομοσπονδιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων

της σύστημα μπορεί να είναι ή κεντρικό ή κατανεμημένο ή ένα άλλο ΟΣΔΒΔ. Ενα σημαντικό χαρακτηριστικό σε ένα ΟΣΔΒΔ είναι ότι μια βάση-συνιστώσα μπορεί να συνεχίζει τις τοπικές εργασίες της, την ίδια ώρα που συμμετέχει στην ομοσπονδία.

Τα συστήματα διαχείρισης ομοσπονδιακών βάσεων δεδομένων είναι υποσύνολα μίας γενικότερης κατηγορίας συστημάτων, των *συστημάτων πολυβάσεων δεδομένων* (ΣΠΒΔ). Ενα ΣΠΒΔ υποστηρίζει πράξεις πάνω σε πολλές βάσεις-συνιστώσες, όπου κάθε βάση-συνιστώσα έχει ένα διαφορετικό σύστημα διαχείρισης. Ενα ΣΠΒΔ ονομάζεται *ομογενές ΣΠΒΔ*, εάν τα ΣΔΒΔ όλων των βάσεων-συνιστώσων είναι τα ίδια, αλλιώς ονομάζεται *ετερογενές ΣΠΒΔ*.

Τέλος, ένα σύστημα που επιτρέπει περιοδική ανταλλαγή δεδομένων ανάμεσα σε πολλές βάσεις ονομάζεται *σύστημα ανταλλαγής δεδομένων*.

2.2 Ταξινόμηση ετερογενών βάσεων

Τα συστήματα πολυβάσεων δεδομένων μπορούν να χαρακτηριστούν σε τρεις άξονες : κατανομή, ετερογένεια και αυτονομία [62].

Κατανομή

Τα δεδομένα μπορεί να είναι κατανεμημένα σε πολλές βάσεις δεδομένων. Οι βάσεις αυτές μπορεί να είναι στο ίδιο ή σε διαφορετικά υπολογιστικά συστήματα. Το πλεονέκτημα της κατανομής των δεδομένων

είναι η αύξηση της ευκολίας προσπέλασης τους. Στην περίπτωση των ομοσπονδιακών βάσεων, η αιτία της κατανομής των δεδομένων, είναι η ύπαρξη των βάσεων πριν τη δημιουργία της ομοσπονδίας τους.

Ετερογένεια

Πολλοί τύποι ετερογένειας οφείλονται σε τεχνολογικές διαφορές, για παράδειγμα, έχουμε διαφορές στο υλικό, στο λογισμικό, στο λειτουργικό σύστημα κ.τ.λ. Ερευνητές και τεχνικοί που αναπτύσσουν λογισμικό έχουν ασχοληθεί με αυτές τις ετερογένειες, έτσι σήμερα υπάρχουν πολλά εμπορικά συστήματα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) που λειτουργούν σε ετερογενή περιβάλλοντα χωρίς πρόβλημα. Γενικά οι ετερογένειες στα ΣΔΒΔ μπορούν να διαχωριστούν σε αυτές που οφείλονται σε διαφορές των ΣΔΒΔ και σε αυτές που οφείλονται σε διαφορές στη σημασιολογία των δεδομένων.

Στις ετερογένειες, που οφείλονται σε διαφορές των ΣΔΒΔ, περιλαμβάνονται διαφορές στις δομές που χρησιμοποιούνται, διαφορές στους περιορισμούς που μπορούν να επιβληθούν και διαφορές στις ερωτηματικές γλώσσες που χρησιμοποιούνται.

Στις ετερογένειες, που οφείλονται σε διαφορές στη σημασιολογία των δεδομένων περιλαμβάνονται διαφορές στην απόδοση σημασίας, στην μετάφραση ή τη χρησιμοποίηση των ίδιων ή σχετιζόμενων δεδομένων. Ο εντοπισμός αυτών των ετερογενειών είναι ένα δύσκολο πρόβλημα [22].

Τέλος, πρέπει να αναφερθεί, ότι ένα βασικό πρόβλημα στην επίτευξη της συνεργασίας ετερογενών βάσεων είναι η λύση των ετερογενειών που εμφανίζονται ανάμεσα τους.

Αυτονομία

Μια βάση-συνιστώσα που συμμετέχει σε ένα ΟΣΔΒΔ μπορεί να επιδείξει πολλών ειδών αυτονομίες.

- *Αυτονομία Σχεδιασμού*
  είναι η δυνατότητα μιας βάσης-συνιστώσας να επιλέξει το σχεδιασμό της, όπως τα δεδομένα, τον τρόπο αναπαράστασης δεδομένων, την σημασιολογία δεδομένων, τους περιορισμούς που ισχύουν στα δεδομένα κ.τ.λ.

- *Αυτονομία Επικοινωνίας*
  είναι η δυνατότητα μιας βάσης-συνιστώσας να αποφασίσει πότε θα επικοινωνήσει με άλλες βάσεις-συνιστώσες ή τα ΟΣΔΒΔ που ανήκει.

- *Αυτονομία Εκτέλεσης*
  είναι η δυνατότητα μιας βάσης-συνιστώσας να εκτελεί τοπικά πράξεις, χωρίς να επηρεάζεται από τις εξωτερικές πράξεις που στέλνονται από άλλες βάσεις-συνιστώσες ή το ΟΣΔΒΔ. Αυτό συνεπάγεται ότι μπορεί να απορρίψει οποιαδήποτε εξωτερική πράξη δεν τηρεί τους τοπικούς περιορισμούς της.

- *Αυτονομία Συσχέτισης*
  είναι η δυνατότητα μιας βάσης-συνιστώσας να αποφασίσει πότε και πόσο θα μοιραστεί τη

λειτουργία της και της πηγές της με άλλους. Αυτό περιλαμβάνει τη δυνατότητα να συσχετίζει ή όχι τον εαυτό της με την ομοσπονδία και τη δυνατότητα να συμμετέχει σε μία ή περισσότερες ομοσπονδίες.

Τα *συστήματα πολυβάσεων δεδομένων* μπορούν να ταξινομηθούν σε δύο τύπους με βάση την αυτονομία συσχέτισης των βάσεων-συνιστωσών τους : *μη-ομοσπονδιακά συστήματα βάσεων δεδομένων* και *ομοσπονδιακά συστήματα βάσεων δεδομένων*.

Ενα *μη-ομοσπονδιακό σύστημα βάσεων δεδομένων* είναι ένας συνδυασμός από ΣΔΒΔ τα οποία **δεν** είναι αυτόνομα. Εχει ένα επίπεδο διαχείρισης και όλες οι πράξεις εφαρμόζονται ομοιόμορφα σε όλες τις βάσεις. Σε ένα τέτοιο σύστημα δεν διαχωρίζονται οι τοπικοί από τους μη-τοπικούς χρήστες. Ενα συγκεκριμένο είδος μη-ομοσπονδιακού συστήματος βάσεων δεδομένων, όπου όλες οι βάσεις είναι πλήρως συνδυασμένες μπορεί να ονομαστεί *ενοποιημένο σύστημα πολυβάσεων δεδομένων*.

Ενα *ομοσπονδιακό σύστημα βάσεων δεδομένων* αποτελείται από ΣΔΒΔ τα οποία αν και είναι αυτόνομα συμμετέχουν στην ομοσπονδία επιτρέποντας μερικό και ελεγχόμενο διαμοιρασμό των δεδομένων τους. Τα ΟΣΔΒΔ αποτελούν ένα συμβιβασμό μεταξύ κανενός συνδυασμού και ολικού συνδυασμού ετερογενών ή μη ΣΔΒΔ. Τα ΟΣΔΒΔ μπορούν να διαχωριστούν σε *χαλαρά-συνδεδεμένα* και *στενά-συνδεδεμένα* ανάλογα με το ποιός ελέγχει την ομοσπονδία και το πως έχουν συνδυαστεί οι βάσεις-συνιστώσες.

Σε ένα *χαλαρά-συνδεδεμένο* ΟΣΔΒΔ είναι ευθύνη του χρήστη να δημιουργήσει και να διαχειρίζεται την ομοσπονδία και δεν ασκείται έλεγχος από το ΟΣΔΒΔ και τους διαχειριστές του. Αλλοι όροι που χρησιμοποιούνται για χαλαρά-συνδεδεμένα ΟΣΔΒΔ είναι *διαλειτουργικά ΣΒΔ* και *συστήματα πολυβάσεων δεδομένων*. Σε ένα χαλαρά-συνδεδεμένο ΟΣΔΒΔ υποστηρίζονται πολλαπλά ομοσπονδιακά σχήματα[1].

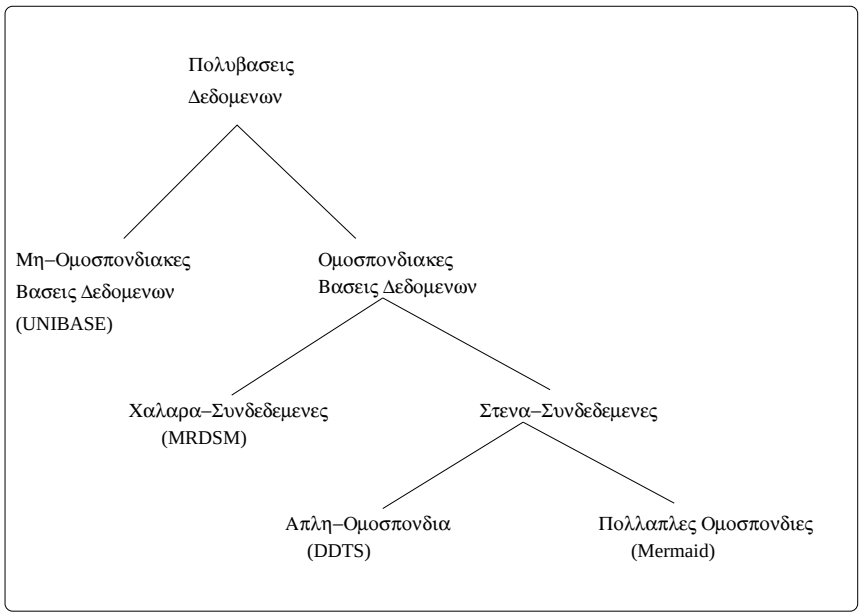
Σε ένα *στενά-συνδεδεμένο* ΟΣΔΒΔ, η ίδια η ομοσπονδία και οι διαχειριστές της έχουν την ευθύνη της δημιουργίας και της διαχείρισης της ομοσπονδίας. Επίσης έχουν την ευθύνη της άσκησης ελέγχου στις βάσεις-συνιστώσες. Τα στενά-συνδεδεμένα ΟΣΔΒΔ μπορεί να έχουν ένα ή περισσότερα ομοσπονδιακά σχήματα. Εαν έχουν ένα τότε λέγεται ότι υπάρχει *απλή ομοσπονδία* αλλιώς λέγεται ότι υπάρχουν *πολλαπλές ομοσπονδίες*.

Στο Σχήμα 2.2 φαίνεται η ταξινόμηση των *συστημάτων πολυβάσεων δεδομένων* που περιγράφηκε, καθώς επίσης αναφέρεται ένα ενδεικτικό παράδειγμα για κάθε είδος.

Μια αναλυτική ταξινόμηση ετερογενών βάσεων δεδομένων στους τρεις άξονες της ετερογένειας, κατανομής και αυτονομίας παρουσιάζεται στο [51].

Τέλος στο [45] παρουσιάζεται μια ακόμα ταξινόμηση κατηγοριών ετερογενών συστημάτων. Η ταξινόμηση αυτή κατατάσσει τα συστήματα από τα πιο στενά συνδεδεμένα προς τα πιο χαλαρά συνδεδεμένα και περιλαμβάνει συνοπτικά : κατανεμημένα συστήματα, συστήματα πολυβάσεων

[1] Η δημιουργία ενός ΟΣΔΒΔ έχει σαν αποτέλεσμα την κατασκευή ενός ή περισσοτέρων ομοσπονδιακών σχημάτων ανάλογα με το ποιές λειτουργίες θα εφαρμόζονται από κάθε χρήστη ή εφαρμογή.



Σχήμα 2.2: Ταξινόμηση πολυβάσεων δεδομένων ως προς την αυτονομία συσχέτισης των βάσεων-συνιστωσών τους

δεδομένων με καθολικό σχήμα, ομοσπονδιακά συστήματα, γλωσσικά συστήματα πολυβάσεων δεδομένων, ομογενή γλωσσικά συστήματα πολυβάσεων δεδομένων και διαλειτουργικά συστήματα.

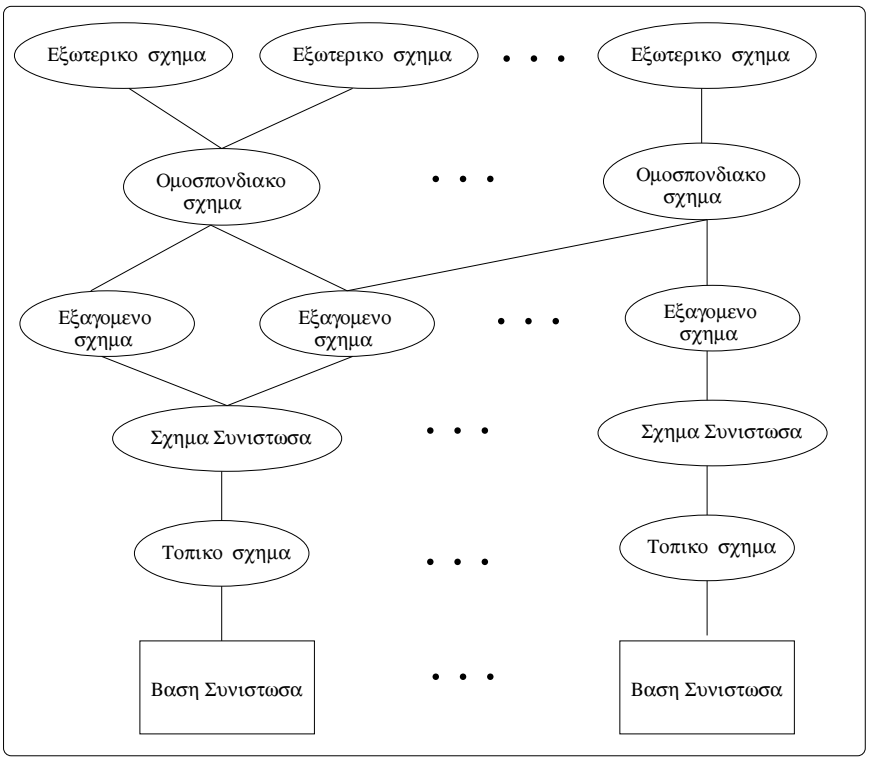
2.3 Αρχιτεκτονική ομοσπονδιακών συστημάτων

Η αρχιτεκτονική ενός ομοσπονδιακού συστήματος (Σχήμα 2.3) έχει πέντε επίπεδα.

- *Τοπικό Σχήμα*
  Το τοπικό σχήμα είναι το σχήμα μιας βάσης-συνιστώσας εκφρασμένο στο μοντέλο του συστήματος διαχείρισης που την υποστηρίζει.

- *Σχήμα Συνιστώσα*
  Το σχήμα συνιστώσα παράγεται από την μετάφραση των τοπικών σχημάτων στο μοντέλο δεδομένων του ΟΣΔΒΔ που ονομάζεται *κανονικό ή κοινό μοντέλο* (ΚΜΔ). Δύο είναι οι λόγοι για τον ορισμό των σχημάτων συνιστωσών στο ΚΜΔ : (1) να επιτευχθεί η παράσταση όλων των σχημάτων που μετέχουν σε ένα κοινό μοντέλο και (2) να προστεθεί η τυχόν σημασιολογία που λείπει από τα τοπικά σχήματα.

- *Εξαγόμενο Σχήμα*
  Το εξαγόμενο σχήμα είναι το υποσύνολο ενός σχήματος συνιστώσας με το οποίο θα συμμετέχει η βάση-συνιστώσα στην ομοσπονδία. Αυτό το σχήμα ορίζεται γιατί η βάση-συνιστώσα μπορεί να μην θέλει να συμμετέχει με όλα τα δεδομένα της στην ομοσπονδία.



Σχήμα 2.3: Αρχιτεκτονική 5-επιπέδων ενός Ομοσπονδιακού Συστήματος Βάσεων Δεδομένων

- *Ομοσπονδιακό Σχήμα*
  Το ομοσπονδιακό σχήμα είναι ο συνδυασμός πολλών εξαγομένων σχημάτων. Μπορεί να υπάρχουν πολλά ομοσπονδιακά σχήματα, ένα για κάθε κλάση χρηστών ή εφαρμογών της ομοσπονδίας.

- *Εξωτερικό Σχήμα*
  Ενα εξωτερικό σχήμα είναι ένα ειδικό σχήμα για ένα χρήστη ή/μία εφαρμογή. Λόγοι για τη χρησιμοποίηση εξωτερικών σχημάτων είναι ότι : (1) το ομοσπονδιακό σχήμα ίσως είναι πολύ μεγάλο και πολύπλοκο για να έρθει ο απλός χρήστης σε επαφή μαζί του, (2) ίσως πρέπει να προστεθούν επιπλέον περιορισμοί και (3) πρέπει να επιτευχθεί έλεγχος προσπέλασης στις βάσεις-συνιστώσες.

Υπάρχουν περιπτώσεις ΟΣΔΒΔ που ορισμένα από τα επίπεδα που αναφέρονται συγχωνεύονται ή παραλείπονται ανάλογα με τις συνθήκες.

2.4 Προβλήματα συνδυασμού ετερογενών βάσεων

Οι ετερογένειες στο σχήμα και στη σημασιολογία είναι ένα σοβαρό πρόβλημα στη δημιουργία και χρησιμοποίηση *συστημάτων πολυβάσεων δεδομένων* [62], [45],[54],[22]. Στην ενότητα αυτή

θα αναφερθούμε συνοπτικά σε ορισμένες ταξινομήσεις ειδών ετερογένειας που αναφέρονται στη βιβλιογραφία και αποτελούν προβλήματα στο συνδυασμό ετερογενών βάσεων.

Στο [14] παρουσιάζεται μια ταξινόμηση διαφόρων ετερογενειών που εμφανίζονται ανάμεσα στα σχήματα ετερογενών βάσεων.

1. Το πρώτο είδος είναι οι *σημασιολογικές ετερογένειες*. Το είδος αυτό αναφέρεται στο γεγονός ότι διαφορετικοί σχεδιαστές βάσεων δεδομένων δεν αντιλαμβάνονται τα ίδια αντικείμενα με τον ίδιο τρόπο αλλά τα φαντάζονται σαν επικαλυπτόμενα σύνολα
   π.χ. Φοιτητές και Φοιτητές του Τμήματος Υπολογιστών.

2. Το δεύτερο είδος είναι οι *ετερογένειες περιγραφής*. Εδώ το ίδιο αντικείμενο περιγράφεται από διαφορετικούς σχεδιαστές βάσεων δεδομένων με διαφορετικά σύνολα ιδιοτήτων.

3. Στην τρίτη κατηγορία ανήκουν οι ετερογένειες που προέρχονται από το γεγονός ότι έχουν χρησιμοποιηθεί διαφορετικά μοντέλα για την παράσταση των δεδομένων.

4. Τέλος υπάρχουν και οι περιπτώσεις που χρησιμοποιείται το ίδιο μοντέλο για την παράσταση των δεδομένων αλλά χρησιμοποιούνται διαφορετικές δομές για την παράσταση των αντικειμένων. Αυτές οι ετερογένειες λέγονται *δομικές ετερογένειες*.

Στο [32] παρουσιάζεται μια ολοκληρωμένη απαρίθμηση και ταξινόμηση των τύπων ετερογενειών, που παρουσιάζονται σε συστήματα πολυβάσεων δεδομένων, που στηρίζονται αποκλειστικά στο σχεσιακό μοντέλο. Οι ετερογένειες χωρίζονται σε *ετερογένειες σχήματος* και *ετερογένειες δεδομένων*. Οι λόγοι για την ύπαρξη *ετερογενειών σχήματος* είναι (1) η χρήση διαφορετικών δομών (πινάκων, γνωρισμάτων) για την παράσταση της ίδιας πληροφορίας και (2) οι διαφορετικές προδιαγραφές που ακολουθούνται για τις ίδιες δομές (ονόματα, περιορισμοί, τύποι δεδομένων). Οι *ετερογένειες σχήματος* χωρίζονται σε ετερογένειες πίνακα-με-πίνακα (table-versus-table conflicts), ετερογένειες γνωρίσματος-με-γνώρισμα (attribute-versus-attribute conflicts) και ετερογένειες πίνακα-με-γνώρισμα (table-versus-attribute conflicts). Οι *ετερογένειες δεδομένων* περιλαμβάνουν λανθασμένα δεδομένα και διαφορετικές παραστάσεις των ίδιων δεδομένων.

Τέλος στο [46] μπορεί να βρεθεί μια ακόμα καταγραφή ετερογενειών ανάμεσα στα σχήματα ετερογενών βάσεων.

2.5 Κοινωνία δεδομένων

Ενας χρήστης, είναι φυσικό, να αντιμετωπίζει πολλά προβλήματα σε περιβάλλοντα που είναι ετερογενή. Αυτό συμβαίνει διότι δεν είναι δυνατόν να γνωρίζει πολλές λεπτομέρειες για κάθε είδους βάση που συμμετέχει στο ετερογενές περιβάλλον που θέλει να εργαστεί. Ετσι η ομοιόμορφη, ενοποιημένη προσπέλαση στα δεδομένα των βάσεων-συνιστωσών αποτελεί έναν από τους κυριότερους σκοπούς

ενός ομοσπονδιακού συστήματος [38], [25],[67],[62]. Το πρόβλημα της ομοιόμορφης και ενοποιημένης προσπέλασης στα δεδομένα των βάσεων-συνιστωσών αναφέρεται σαν *κοινωνία δεδομένων*. Στη βιβλιογραφία αναφέρονται συνοπτικά τρία επίπεδα για την επίτευξη συνδυασμού ετερογενών βάσεων στο θέμα της κοινωνίας δεδομένων.

1. Μεταφορά όλων των δεδομένων από μια βάση σε μια άλλη βάση. Σε αυτήν την περίπτωση η βάση δεδομένων μετατρέπεται σε μία άλλη ισοδύναμη βάση δεδομένων εκφρασμένη σε ένα άλλο μοντέλο. Αυτό επιτυγχάνεται με τη χρήση ενός ειδικού εργαλείου που ονομάζεται *μετατροπέας βάσης δεδομένων* [25]. Αυτό που γίνεται σε αυτήν την περίπτωση είναι ότι : (1) δημιουργείται ένα αντίγραφο της αρχικής βάσης, (2) το αντίγραφο μετατρέπεται στο μοντέλο της τελικής βάσης και (3) το αποτέλεσμα δίνεται στο χρήστη.

2. Δημιουργία ενδιάμεσου σχήματος και μεταφορά δεδομένων σε αυτό, για κάθε βάση που συμμετέχει. Σε αυτή την περίπτωση αυτό που γίνεται είναι ότι δημιουργείται ένα σχήμα σε ένα κοινό μοντέλο, το οποίο αποτελεί συνδυασμό των σχημάτων όλων των βάσεων που συμμετέχουν. Με το σχήμα αυτό εφοδιάζεται μια νέα κοινή-βάση στην οποία μεταφέρονται τα δεδομένα όλων των βάσεων που συμμετέχουν. Το αποτέλεσμα που δίνεται στο χρήστη είναι αυτή η κοινή-βάση.

3. Δημιουργία ενός ενδιάμεσου σχήματος, από όλα τα σχήματα των βάσεων που συμμετέχουν και ανάπτυξη κατάλληλων μηχανισμών για προσπέλαση των δεδομένων απευθείας στις βάσεις που βρίσκονται μέσω του σχήματος αυτού. Η περίπτωση αυτή μοιάζει με την περίπτωση 2 μόνο που δεν δημιουργείται μια νέα κοινή-βάση στην οποία μεταφέρονται όλα τα δεδομένα. Εδώ τα δεδομένα παραμένουν στις βάσεις όπου βρίσκονται και ο χρήστης μπορεί να τα προσπελάσει με τους μηχανισμούς που του παρέχονται.

Το πρόβλημα σε όλες τις παραπάνω περιπτώσεις είναι ότι το αρχικό και τελικό σχήμα μπορεί να μην είναι ισοδύναμα (ενότητα 4.1.2).

2.6 Συστήματα για ετερογενείς βάσεις δεδομένων

Στην προσπάθεια υποστήριξης συνδυασμού ετερογενών βάσεων δεδομένων έχουν δημιουργηθεί πολλά συστήματα [24],[59],[45]. Μερικά από αυτά τα συστήματα έχουν δημιουργηθεί για εμπορικές εφαρμογές και άλλα σαν πρωτότυπα συστήματα. Σε αυτήν την ενότητα θα αναφερθούμε συνοπτικά σε ορισμένα από αυτά που συναντήθηκαν στην βιβλιογραφία.

Το σύστημα **ADDS** (Amoco Distibuted Database System) [24],[8] δημιουργήθηκε από την Amoco Production Company. Η κατασκευή του συστήματος ξεκίνησε στα τέλη του 1983. Το ADDS παρέχει ομοιόμορφη προσπέλαση σε προϋπάρχουσες ετερογενείς βάσεις δεδομένων και χρησιμοποιεί για το συνδυασμό τους το σχεσιακό μοντέλο. Ανήκει στην κατηγορία των στενά-συνδεδεμένων ομοσπονδιακών συστημάτων και υποστηρίζει πολλαπλά ομοσπονδιακά σχήματα. Υποστηρίζει

συνδυασμό ιεραρχικών, σχεσιακών και δικτυωτών μοντέλων. Στο ADDS διατηρείται η αυτονομία των βάσεων-συνιστωσών και δεν απαιτούνται αλλαγές στα διαχειριστικά τους συστήματα.

Το σύστημα **DATAPLEX** [24],[15] είναι ένα ετερογενές κατανεμημένο σύστημα διαχείρισης βάσεων δεδομένων που δημιουργήθηκε από την General Motors Corporation. Το σχεσιακό μοντέλο χρησιμοποιείται σαν το κοινό μοντέλο δεδομένων του DATAPLEX. Το DATAPLEX ανήκει στην κατηγορία των στενά-συνδεδεμένων ομοσπονδιακών συστημάτων και υποστηρίζει πολλαπλά ομοσπονδιακά σχήματα. Η SQL χρησιμοποιείται σαν γλώσσα ομοιόμορφης προσπέλασης των δεδομένων των βάσεων-συνιστωσών του. Το DATAPLEX αποτελείται από 14 κύρια κομμάτια και επιτρέπει σε ερωτήσεις και δοσοληψίες να ανακτούν και να ενημερώνουν τα δεδομένα των βάσεων-συνιστωσών του.

Το σύστημα **IMDAS** (Integrated Manufacturing Data Administrator System) [24] αναπτύχθηκε στο National Institute of Standards and Technology της Φλώριδας. Σκοπός του ήταν να παρέχει προσπέλαση σε πολλά συστήματα, σε πολλές πηγές βιομηχανικών δεδομένων και να μπορεί να συνεργαστεί με παλιές και νέες εφαρμογες διαχείρισης βάσεων δεδομενών. Το IMDAS ανήκει στην κατηγορία των στενά-συνδεδεμένων ομοσπονδιακών συστημάτων και υποστηρίζει ένα μόνο ολικό ομοσπονδιακό σχήμα. Το κοινό μοντέλο δεδομένων του IMDAS είναι ένα σημασιολογικό δίκτυο.

Το σύστημα **Mermaid** [24] άρχισε να αναπτύσσεται το 1980 στα εργαστήρια της System Development Corporation κατόπιν παραγγελίας του Υπουργείου Αμύνης των Ηνωμένων Πολιτειών. Σκοπός του ήταν η προσπέλαση και ο συνδυασμός δεδομένων που υπήρχαν σε διαφορετικές βάσεις. Το Mermaid ανήκει στην κατηγορία των στενά-συνδεδεμένων ομοσπονδιακών συστημάτων και υποστηρίζει πολλαπλά ομοσπονδιακά σχήματα.

Το σύστημα **MULTIBASE** [24] άρχισε να αναπτύσσεται το 1980 στα εργαστήρια της Xerox. Σκοπός του ήταν να παρέχει μια ομοιόμορφη και ολοκληρωμένη επαφή χρήσης για ανάκτηση δεδομένων σε ετερογενείς, κατανεμημένες βάσεις που προϋπήρχαν. Το σύστημα είναι γενικό και δεν έχει σχεδιαστεί για την υποστήριξη συγκεκριμένης εφαρμογής. Μπορεί να χειρίζεται ιεραρχικές, σχεσιακές και δικτυωτές βάσεις δεδομένων όπως και συστήματα αρχείων. Το MULTIBASE χρησιμοποιεί τη γλώσσα ορισμού και διαχείρισης δεδομένων DAPLEX και ανήκει στην κατηγορία των στενά-συνδεδεμένων ομοσπονδιακών συστημάτων με πολλαπλά ομοσπονδιακά σχήματα.

Μερικά ακόμα συστήματα είναι τα KODIM [53], HD-DBMS [9], InterBase [50],[49], DARPA [66] και το [65]. Στο [45] παρουσιάζονται συνοπτικοί πίνακες με συστήματα που αναφέρονται στη βιβλιογραφία έως το 1992. Οι πίνακες αυτοί περιέχουν και άλλες πληροφορίες για τα συστήματα, όπως το στάδιο ανάπτυξης στο οποίο βρίσκονταν τότε και ορισμένα γενικά χαρακτηριστικά τους.

Κεφάλαιο 3

Σύστημα μεταφοράς δεδομένων

Στο κεφάλαιο αυτό παρουσιάζεται το σύστημα που κατασκευάστηκε στα πλαίσια αυτής της εργασίας. Σκοπός του συστήματος είναι η επίτευξη μεταφοράς δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων.

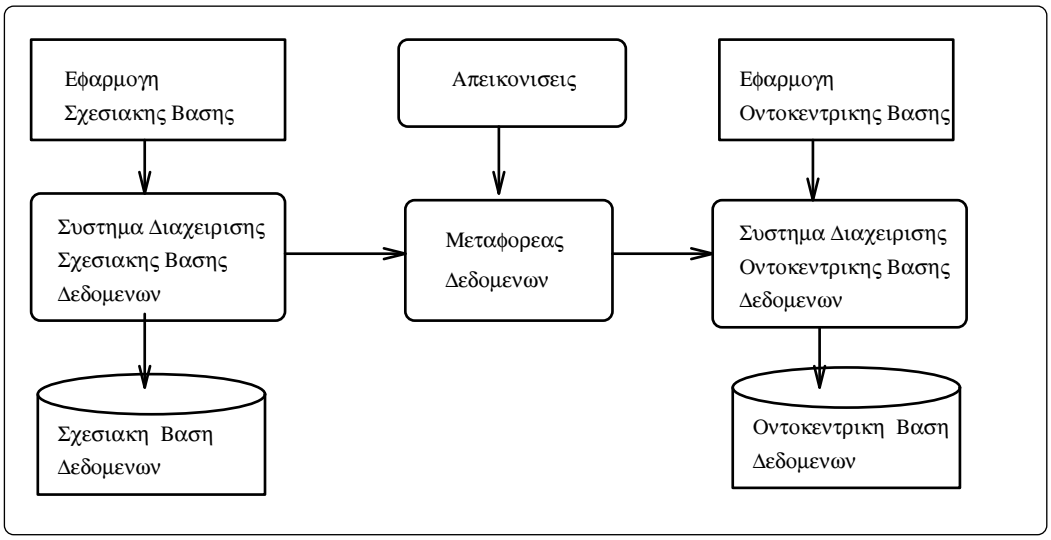
Το κεφάλαιο αυτό είναι οργανωμένο ως εξής. Στην ενότητα 3.1 παρουσιάζεται το πρόβλημα. Στην ενότητα 3.2 παρουσιάζεται η ιδέα του συστήματος. Στην ενότητα 3.3 παρουσιάζεται η αρχιτεκτονική του συστήματος, οι μηχανισμοί που την αποτελούν και θέματα συγχρονισμού των βάσεων και ανάκτησης δεδομένων. Τέλος στην ενότητα 3.4 παρουσιάζεται η διαδικασία επικοινωνίας της σχεσιακής με την οντοκεντρική βάση με τη χρήση του συστήματος αυτού.

3.1 Ορισμός προβλήματος

Το σύστημα που περιγράφεται σε αυτό το κεφάλαιο είναι μια προσέγγιση στο πρόβλημα της κοινωνίας δεδομένων ετερογενών βάσεων και ειδικότερα στο πρόβλημα της μεταφοράς δεδομένων όπως ορίστηκε στην ενότητα 2.5.

Το σύστημα λειτουργεί ανάμεσα σε δύο ετερογενείς βάσεις δεδομένων, μια σχεσιακή και μια οντοκεντρική. Οι βάσεις αυτές υποθέτουμε ότι έχουν δεδομένο σχήμα η κάθε μια. Επίσης υποθέτουμε ότι τα σχήματα των δύο βάσεων περιγράφουν ακριβώς τον ίδιο κόσμο ή ότι ο κόσμος που περιγράφει το ένα είναι υποσύνολο του κόσμου που περιγράφει το άλλο (δεν έχει σημασία ποιό από τα δύο σχήματα είναι υποσύνολο του άλλου). Ο στόχος του συστήματος είναι να ενημερώσει την οντοκεντρική βάση με τις αλλαγές (εισαγωγές-ενημερώσεις-διαγραφές) που γίνονται στα δεδομένα της σχεσιακής βάσης, έτσι ώστε οι δύο βάσεις να περιέχουν την ίδια πληροφορία για τα κοινά μέρη τους. Η ενημέρωση αυτή δεν γίνεται αμέσως μόλις συμβεί μια αλλαγή στη σχεσιακή βάση αλλά όποτε το ζητήσει ο χρήστης[1]. Επίσης δεν γίνεται τυχαία αλλά με τη καθοδήγηση απεικονίσεων που ορίζονται από το χρήστη. Το σύστημα ανήκει στην κατηγορία των *ομοσπονδιακών συστημάτων διαχείρισης βάσεων δεδομένων* (FDBMS) και

[1]Με τον όρο χρήστης εννοούμε σε αυτό το κεφάλαιο κάποιον που έχει καλή γνώση της σχεσιακής και της οντοκεντρικής βάσης.



Σχήμα 3.1: Μεταφορέας δεδομένων

συγκεκριμένα είναι ένα *χαλαρά-συνδεδεμένο ομοσπονδιακό σύστημα διαχείρισης βάσεων δεδομένων* όπως ορίστηκε στο κεφάλαιο 2.

Την διαδικασία μεταφοράς δεδομένων από τη μία βάση στην άλλη μπορούμε να τη δούμε σαν ένα είδος επικοινωνίας ανάμεσα τους, κατά την οποία η οντοκεντρική βάση επικοινωνεί με τη σχεσιακή για να ενημερωθεί με νέα δεδομένα. Για αυτό το λόγο συχνά αναφέρεται σαν *''επικοινωνία των δύο βάσεων''* αντί για *''μεταφορά δεδομένων από τη σχεσιακή στην οντοκεντρική βάση''*.

3.2 Η ιδέα του συστήματος

Η ιδέα του συστήματος είναι η κατασκευή ενός *μεταφορέα δεδομένων*. Αυτός ο *μεταφορέας δεδομένων* θα στέκεται ανάμεσα στις δύο βάσεις και θα μεταφέρει δεδομένα από τη σχεσιακή στην οντοκεντρική όταν ζητηθεί από το χρήστη. Επίσης θα είναι ανεξάρτητος από τις εφαρμογές που χειρίζονται την σχεσιακή και την οντοκεντρική βάση. Η μορφή του μεταφορέα φαίνεται στο Σχήμα 3.1. Η ιδέα της κατασκευής *μεταφορέα δεδομένων* για μεταφορά δεδομένων ανάμεσα σε δύο βάσεις δεν είναι καινούργια. Στο [6] παρουσιάζεται ένα σύστημα για μεταφορά δεδομένων από μια ιεραρχική βάση IMS σε μία σχεσιακή βάση δεδομένων DB2 (και αντιστρόφως), το οποίο κατασκευάστηκε από την IBM[2] για την Τράπεζα της Ελβετίας[3]. Ιδέα του συστήματος αυτού είναι η κατασκευή ενός συστήματος μεταφοράς δεδομένων, που ονομάζεται *IBM Data Propagator MVS/ESA*. Το σύστημα αυτό στέκεται ανάμεσα στις δύο βάσεις και στέλνει ό,τι αλλαγές γίνονται στη μία στην άλλη, με σύγχρονο ή ασύγχρονο τρόπο χωρίς να επηρεάζει η λειτουργία του καμία από τις δύο εφαρμογές. Επίσης μπορεί να υποστηρίξει

[2]IBM USA, IBM Switzerland
[3]Swiss Bank Corporation

ταυτόχρονη μεταφορά δεδομένων και προς τις δύο κατευθύνσεις, μόνο που κάτι τέτοιο δεν συνίσταται γιατί μπορεί να αλλοιωθεί η συνέπεια των δύο βάσεων. Στην Τράπεζα της Ελβετίας ο μεταφορέας αυτός χειρίζεται τις κύριες βάσεις πελατών και λογαριασμών των πέντε υποκαταστημάτων της στην Ελβετία χωρίς να επηρεάζει αισθητά τους χρόνους απόκρισης των κανονικών εφαρμογών.

Το σύστημα που παρουσιάζεται στην παρούσα εργασία στοχεύει στο να υποστηρίξει την επικοινωνία ανάμεσα σε μια οποιαδήποτε σχεσιακή βάση και μια οντοκεντρική βάση υλοποιημένη στην γλώσσα Telos [5]. Επειδή όμως τα διαχειριστικά συστήματα των σχεσιακών βάσεων διαφέρουν μεταξύ τους είναι προφανές ότι ένα και μόνο σύστημα είναι πολύ δύσκολο να μπορεί να καλύψει τις ανάγκες όλων. Για αυτό το λόγο, η ιδέα που υποστηρίζεται από την αρχιτεκτονική του συστήματος, είναι ορισμένα μέρη της να παραμένουν σταθερά, ανεξάρτητα από το είδος της σχεσιακής βάσης, που έχουμε σαν πηγή δεδομένων. Ενώ ορισμένα άλλα, να υποστούν τροποποιήσεις ανάλογα με το διαχειριστικό σύστημα της σχεσιακής βάσης που θέλουμε να υποστηρίξουμε.

Αναλυτικότερα τώρα το πρόβλημα της μεταφοράς δεδομένων από μια σχεσιακή σε μια οντοκεντρική βάση δεδομένων αντιμετωπίζεται από το δικό μας σύστημα, αλλά και σε άλλες παρόμοιες εργασίες [6],[56], σε δύο στάδια : *στάδιο αρχικοποίησης συστήματος* και το *στάδιο μεταφοράς δεδομένων.*

3.2.1 Στάδιο αρχικοποίησης συστήματος

Είναι το πρώτο στάδιο λειτουργίας του συστήματος, κατά το οποίο το σύστημα ενημερώνεται για τις παραμέτρους εισόδου του. Παραμέτρους εισόδου του συστήματος αποτελούν :

- Ολόκληρο το σχεσιακό σχήμα ή μέρος του αποτελούμενο από ορισμούς πινάκων,πεδίων του κάθε πίνακα, συναρτησιακών εξαρτήσεων μεταξύ πινάκων (ενότητα 4.2.1), ορισμό κύριου κλειδιού για κάθε πίνακα και ορισμό καρτεσιανών γινομένων σχέσεων μεταξύ πινάκων.

- Για να μεταφερθούν δεδομένα από τη μια βάση στην άλλη πρέπει πρώτα να καθοριστεί *ποιά δεδομένα* θα μεταφερθούν από τη σχεσιακή βάση και *που* ακριβώς θα πάνε στην οντοκεντρική βάση, να δημιουργηθεί δηλαδή μια απεικόνιση ανάμεσα στις δύο βάσεις, την οποία θα ακολουθούν τα δεδομένα (κεφάλαιο 4). Ο καθορισμός αυτών των απεικονίσεων αποτελεί παράμετρο εισόδου του συστήματος και γίνεται από το χρήστη.

- Τέλος το σύστημα δέχεται σαν είσοδο το οντοκεντρικό σχήμα.

Αφού το σύστημα ενημερωθεί με τα παραπάνω μπορεί να προχωρήσει στο δεύτερο στάδιο, το *στάδιο μεταφοράς δεδομένων.*

3.2.2 Στάδιο μεταφοράς δεδομένων

Στο στάδιο αυτό γίνεται η ενημέρωση της οντοκεντρικής βάσης με τις εισαγωγές,ενημερώσεις ή διαγραφές δεδομένων που έχουν εφαρμοστεί στη σχεσιακή βάση. Το στάδιο αυτό περιλαμβάνει :

- *αρχικοποίηση της οντοκεντρικής βάσης*, κατά την οποία η σχεσιακή βάση ενημερώνει την οντοκεντρική με τα δεδομένα που ήδη περιέχει
- *μετάφραση και μεταφορά δεδομένων* και
- *ανάκτηση δεδομένων της οντοκεντρικής βάσης* σε περίπτωση βλάβης της.

Στις ενότητες που ακολουθούν περιγράφονται αναλυτικά αυτές οι διαδικασίες.

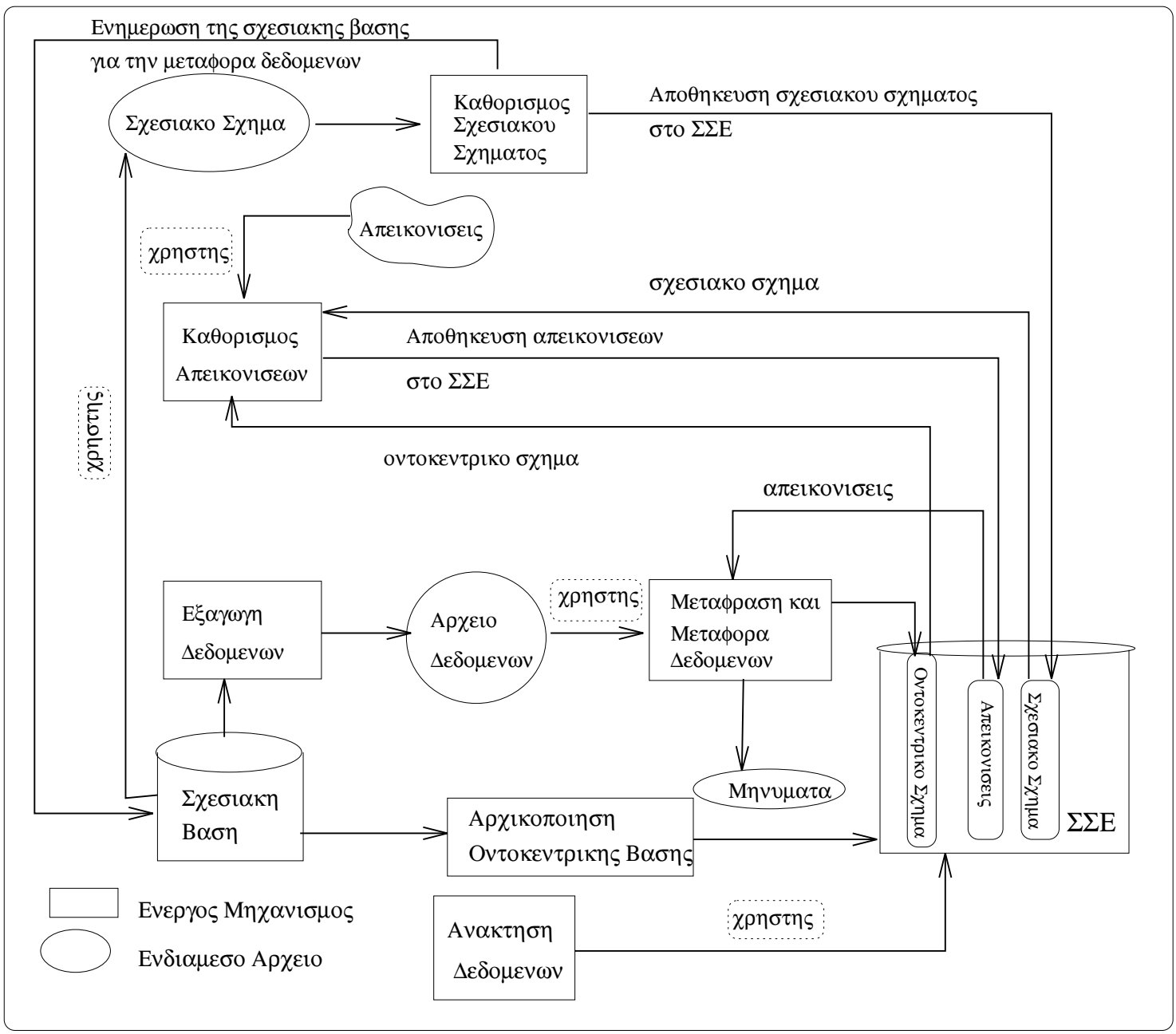
3.3 Αρχιτεκτονική του συστήματος

Η αρχιτεκτονική του συστήματος φαίνεται αναλυτικά στο Σχήμα 3.2. Το σύστημα περιέχει έξι ανεξάρτητους μηχανισμούς :

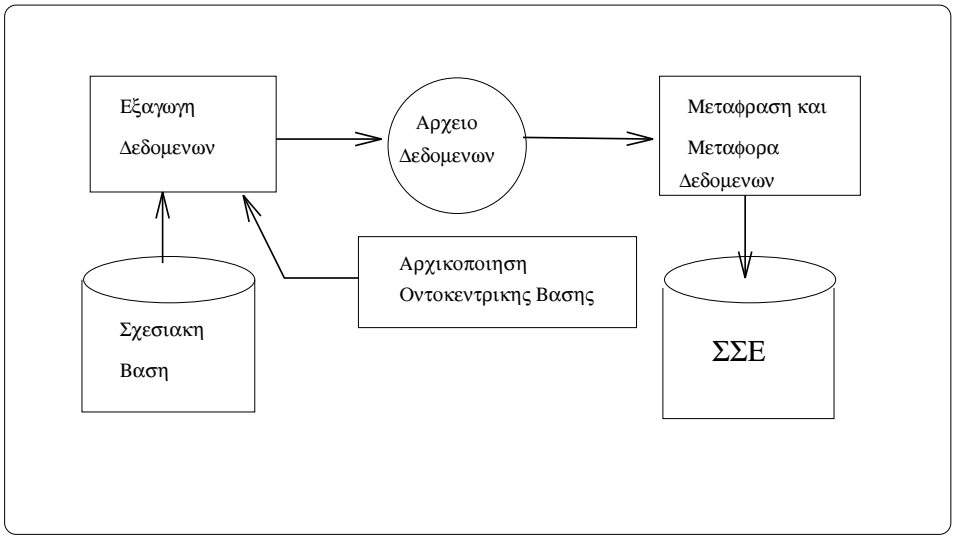
- *Μηχανισμό καθορισμού και αποθήκευσης σχεσιακού σχήματος*,
- *Μηχανισμό καθορισμού και αποθήκευσης απεικονίσεων* ανάμεσα στο σχεσιακό και το οντοκεντρικό σχήμα,
- *Μηχανισμό εξαγωγής δεδομένων* από τη σχεσιακή βάση,
- *Μηχανισμό μετάφρασης και μεταφοράς δεδομένων* από την σχεσιακή στην οντοκεντρική βάση,
- *Μηχανισμό αρχικοποίησης της οντοκεντρικής βάσης* και
- *Μηχανισμό ανάκτησης δεδομένων της οντοκεντρικής βάσης*.

Η αρχιτεκτονική του συστήματος που παρουσιάζεται έχει κοινά χαρακτηριστικά με την αρχιτεκτονική του συστήματος [6]. Και στις δύο αρχιτεκτονικές υπάρχει *Μηχανισμός καθορισμού απεικονίσεων*, *Μηχανισμός αρχικοποίησης* των βάσεων, *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* από τη μία βάση στην άλλη και *Μηχανισμός ανάκτησης δεδομένων* σε περίπτωση βλάβης των βάσεων.

Στο Σχήμα 3.2 σημειώνονται με παραλληλόγραμμα οι ενεργοί μηχανισμοί και με ελλείψεις τα αρχεία που χρησιμοποιούνται για ανταλλαγή δεδομένων μεταξύ των διαφορετικών μηχανισμών ή για επικοινωνία των μηχανισμών με το χρήστη. Επίσης σημειώνονται με την ένδειξη ‘‘χρήστης’’ όλα τα σημεία που είναι απαραίτητη η δράση του χρήστη για να συνεχίσει το σύστημα τη λειτουργία του. Το ΣΣΕ (Σημασιολογικό Σύστημα Ευρετηριασμού) [43] είναι ένα εργαλείο για περιγραφή και τεκμηρίωση μεγάλου πληθυσμού, ιδιόμορφων, εξελισσόμενων και πολλαπλώς συνδεδεμένων δεδομένων. Το ΣΣΕ αναπτύχθηκε από την ομάδα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Ερευνας και χρησιμοποιεί τη γλώσσα απαράστασης γνώσης Telos.



Σχήμα 3.2: Αρχιτεκτονική Συστήματος



Σχήμα 3.3: Μηχανισμός αρχικοποίησης της οντοκεντρικής βάσης

Όλοι οι μηχανισμοί που αναφέρονται διακόπτουν τη λειτουργία τους εάν συναντήσουν λάθος και παράγουν στην οθόνη το κατάλληλο μήνυμα για το χρήστη . Το μήνυμα αυτό αποθηκεύεται στο αρχείο λαθών με όνομα *error_com* για να μην το χάσει ο χρήστης. Επίσης εάν υπάρχουν άλλα μηνύματα για το χρήστη φυλάσσονται στο αρχείο *warning_com*. Εαν δεν υπάρχουν λάθη ή μηνύματα τα αρχεία αυτά είναι κενά.

Όλοι οι μηχανισμοί διαβάζουν ή γράφουν σε αρχεία που βρίσκονται στους καταλόγους που δηλώνουν οι μεταβλητές περιβάλλοντος *COM_WORKING_DIR* και *COM_BACKUP_DIR*. Οι μεταβλητές αυτές πρέπει να τίθενται στους σωστούς καταλόγους πριν κληθεί οποιοσδήποτε μηχανισμός. Η μεταβλητή περιβάλλοντος *COM_WORKING_DIR* πρέπει να δείχνει εκεί όπου βρίσκονται όλα τα αρχεία εισόδου/εξόδου των μηχανισμών και η *COM_BACKUP_DIR* εκεί όπου θέλουμε να αποθηκεύονται τα εφεδρικά αρχεία ασφαλείας.

Τέλος υπάρχουν δύο ακόμα σημαντικές μεταβλητές περιβάλλοντος για τη λειτουργία του συστήματος. Οι μεταβλητές αυτές καθορίζουν αν θα εφαρμοστούν οι πράξεις διαγραφής και ενημέρωσης που γίνονται στα δεδομένα της σχεσιακής βάσης και στα δεδομένα της οντοκεντρικής. Ο λόγος της ύπαρξης τους είναι ότι μπορεί ο χρήστης να μην θέλει να διαγράφεται καμία από τις πληροφορίες που αποκτά. Οι μεταβλητές αυτές είναι οι *ENABLED_DELETE* και *ENABLED_UPDATE*. Η πρώτη καθορίζει αν θα μεταφερθούν οι διαγραφές και οι δεύτερη αν θα μεταφερθούν οι ενημερώσεις. Οταν η τιμή μιας μεταβλητής είναι ένα (**1**) η αλλαγή που καθορίζει μεταφέρεται, αλλιώς αν η τιμή της είναι μηδέν (**0**) δεν μεταφέρεται.

3.3.1 Μηχανισμός αρχικοποίησης οντοκεντρικής βάσης

Ο *Μηχανισμός αρχικοποίησης της οντοκεντρικής βάσης* (Σχήμα 3.3) είναι απαραίτητος για την ομαλή μεταφορά δεδομένων ανάμεσα στις δύο βάσεις. Αυτό γιατί όταν αποφασίζεται να επικοινωνήσουν δύο βάσεις πρέπει να θεωρήσουμε ότι προϋπήρχαν της απόφασης αυτής. Οπότε η σχεσιακή βάση είναι πιθανό να περιέχει ήδη ορισμένα δεδομένα που αφορούν και την οντοκεντρική. Τα δεδομένα αυτά θα πρέπει να μεταφερθούν στην οντοκεντρική βάση για να είναι συνεπείς ανάμεσα τους οι δύο βάσεις. Το σκοπό αυτό επιτελεί ο *Μηχανισμός αρχικοποίησης της οντοκεντρικής βάσης.*

Οταν ενεργοποιηθεί ο *Μηχανισμός αρχικοποίησης της οντοκεντρικής βάσης* :

1. Φέρνει τη σχεσιακή βάση σε κατάσταση που δεν επιτρέπονται ενημερώσεις, έτσι ώστε να μην μπορούν να εφαρμοστούν αλλαγές κατά την ώρα που θα πραγματοποιείται η μεταφορά των δεδομένων (συγχρονισμός των δύο βάσεων) .

2. Διαβάζει τα δεδομένα της σχεσιακής βάσης και παράγει με τη βοήθεια του *Μηχανισμού εξαγωγής δεδομένων* ένα αρχείο που περιέχει όλα τα δεδομένα, με τα οποία πρέπει να ενημερωθεί η οντοκεντρική βάση. Τα δεδομένα αυτά είναι όσα περιέχουν οι πίνακες της σχεσιακής βάσης, που συμμετέχουν στην επικοινωνία, μέχρι εκείνη τη στιγμή.

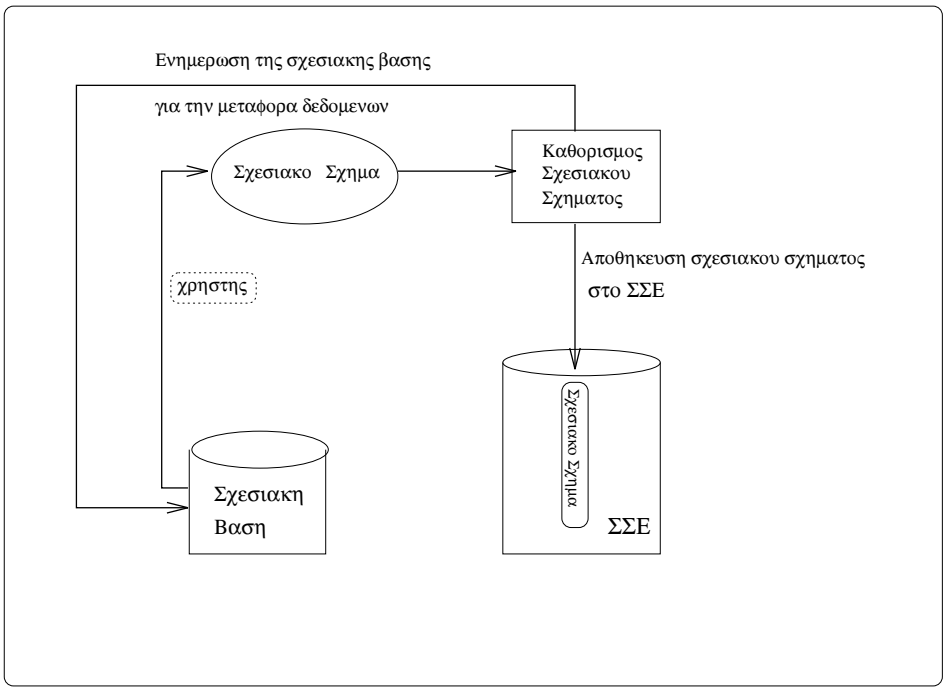
3. Τέλος καλεί τον *Μηχανισμό μετάφρασης και μεταφοράς δεδομένων* και τα δεδομένα που υπολογίστηκαν στο προηγούμενο βήμα εισάγονται στην οντοκεντρική βάση όπως θα περιγραφεί παρακάτω.

Ο *Μηχανισμός αρχικοποίησης της οντοκεντρικής βάσης* πρέπει να κληθεί μια φορά πριν αρχίσει η διαδικασία μεταφοράς των δεδομένων. Μετά μπορεί να χρησιμοποιηθεί για την ανάκτηση των δεδομένων της οντοκεντρικής βάσης, σε περίπτωση που αυτή υποστεί ολοκληρωτική καταστροφή (ενότητα 3.3.6).

3.3.1.1 Συνέπεια βάσεων κατά την επικοινωνία

Το θέμα της συνέπειας των δεδομένων των δύο βάσεων είναι πολύ σημαντικό. Αυτό γιατί σε περίπτωση που η οντοκεντρική βάση δεν περιέχει τα δεδομένα που θα έπρεπε, τότε είναι σχεδόν σίγουρη η μελλοντική εμφάνιση λάθους κατά τη μεταφορά των δεδομένων.

Η συνέπεια των βάσεων, πριν αρχίσει η διαδικασία της μεταφοράς δεδομένων, επιτυγχάνεται με κλήση του *Μηχανισμού αρχικοποίησης της οντοκεντρικής βάσης.* Κατόπιν κατά την μεταφορά των δεδομένων ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* φροντίζει να παίρνει πάντα από τη σχεσιακή βάση όλες τις αλλαγές που έχουν γίνει από την τελευταία φορά που κλήθηκε. Επίσης φροντίζει όλες αυτές οι αλλαγές να γίνονται γνωστές στην οντοκεντρική βάση. Εαν ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* ανακαλύψει ασυνέπεια στα δεδομένα της οντοκεντρικής βάσης ειδοποιεί το χρήστη με το κατάλληλο μήνυμα. Σε αυτή την περίπτωση ο χρήστης καλείται να αποκαταστήσει τη συνέπεια των δεδομένων για να μπορέσει να συνεχιστεί και να ολοκληρωθεί ομαλά η μεταφορά των δεδομένων (ενότητα 3.3.5.1).



Σχήμα 3.4: Μηχανισμός καθορισμού σχεσιακού σχήματος

Ο καλύτερος τρόπος για να διατηρούνται πάντα συνεπή τα δεδομένα των δύο βάσεων είναι να μην επεμβαίνει σε αυτά ο χρήστης παρά μόνο αν του ζητηθεί από το σύστημα μεταφοράς των δεδομένων.

3.3.2 Μηχανισμός καθορισμού σχεσιακού σχήματος

Ο *Μηχανισμός καθορισμού σχεσιακού σχήματος* (Σχήμα 3.4) έχει σαν σκοπό να ενημερώσει :

- την οντοκεντρική βάση με μια περιγραφή του σχεσιακού σχήματος, την οποία θα χρησιμοποιήσει αργότερα ο *Μηχανισμός καθορισμού απεικονίσεων* και ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* και

- την σχεσιακή βάση με πληροφορίες, όπως νέους, βοηθητικούς πινάκες και ρουτίνες, που θα χρησιμεύσουν για την εξαγωγή των δεδομένων, από τη σχεσιακή βάση, κατά τη λειτουργία του συστήματος.

Η περιγραφή του σχεσιακού σχήματος περιλαμβάνει καθορισμό του ονόματος της σχεσιακής βάσης, ορισμούς πινάκων, πεδίων του κάθε πίνακα, συναρτησιακών εξαρτήσεων μεταξύ πινάκων (ενότητα 4.2.1), ορισμό κύριου κλειδιού για κάθε πίνακα που συμμετέχει και ορισμό καρτεσιανών γινομένων μεταξύ πινάκων. Για να αποκτηθεί η παραπάνω περιγραφή ο χρήστης καλείται να την ορίσει σε απλό συντακτικό το οποίο περιγράφεται στο παράρτημα Β.

Οταν ενεργοποιηθεί ο *Μηχανισμός καθορισμού σχεσιακού σχήματος* κάνει τις παρακάτω ενέργειες:

1. Ζητάει από το χρήστη να του προμηθεύσει ένα αρχείο με την περιγραφή του σχεσιακού σχήματος. Το αρχείο αυτό, ο χρήστης πρέπει να το έχει ήδη ετοιμάσει με τη βοήθεια ενός κειμενογράφου και πρέπει να περιέχει πληροφορία για το σχεσιακό σχήμα παρόμοια με αυτήν που προαναφέρθηκε.

2. Διαβάζει το αρχείο αυτό και ελέγχει την ύπαρξη των πινάκων και των πεδίων που έχουν αποδοθεί σε κάθε πίνακα της σχεσιακής βάσης. Οταν τελειώσει παράγει στο συντακτικό της γλώσσας εισαγωγής δεδομένων της οντοκεντρικής βάσης τον κατάλληλο κώδικα για να την ενημερώσει αργότερα[4].

3. Ενημερώνει την οντοκεντρική βάση με την πληροφορία που απέκτησε.

4. Τέλος ενημερώνει τη σχεσιακή βάση, με πληροφορία που θα βοηθήσει την φύλαξη των αλλαγών στα δεδομένα και την εξαγωγή τους από τη βάση, όταν ζητηθεί η μεταφορά τους.

Πριν περάσουμε στην περιγραφή του επόμενου μηχανισμού θα ήταν σκόπιμο να αναφερθούμε στην τελευταία ενέργεια που κάνει ο *Μηχανισμός καθορισμού σχεσιακού σχήματος* δηλαδή *''την ενημέρωση της σχεσιακής βάσης με πληροφορία που θα βοηθήσει την φύλαξη των αλλαγών στα δεδομένα και την εξαγωγή τους από τη βάση''*. Για να μιλήσουμε για αυτό το θέμα πρέπει να δούμε πρώτα ποιά είναι η ακριβής διαδικασία για να επιτευχθεί η απόκτηση και μεταφορά των δεδομένων :

- Καταρχήν είναι χρήσιμο να απομονώνονται κάθε φορά για μεταφορά, μόνο όσα δεδομένα είναι νέες εισαγωγές ή έχουν υποστεί αλλαγή και να μην μεταφέρονται κάθε φορά στην οντοκεντρική βάση όλα τα δεδομένα της σχεσιακής [1]. Ο λόγος που είναι επιθυμητή μια τέτοια συμπεριφορά είναι ότι μειώνεται σημαντικά ο συνολικός χρόνος της μεταφοράς των δεδομένων, όπως και ο χώρος που απαιτείται για την φύλαξη τους.

- Δεύτερον πρέπει η σειρά με την οποία εφαρμόζονται οι αλλαγές στην οντοκεντρική βάση να είναι η ίδια ακριβώς με τη σειρά που συμβαίνουν στη σχεσιακή βάση (ενότητα 5.4.1) για να διατηρούνται τα δεδομένα των δύο βάσεων πάντα συνεπή. Επομένως πρέπει να φυλάσσεται και η σειρά με την οποία συμβαίνουν οι αλλαγές αυτές.

- Τέλος, πρέπει να βρεθεί τρόπος για να επιτευχθεί η εξαγωγή των δεδομένων από τη σχεσιακή βάση, έτσι ώστε να γίνει δυνατή η επεξεργασία τους και κατόπιν η εισαγωγή τους στην οντοκεντρική βάση.

Σε αυτή τη φάση λοιπόν που το σύστημα μαθαίνει ποιοί πίνακες της σχεσιακής θα χρησιμοποιηθούν και τις σχέσεις μεταξύ τους, πρέπει να φροντίσει να δημιουργήσει το υπόβαθρο για να μπορέσουν να εξαχθούν οι αλλαγές στα δεδομένα από τη σχεσιακή βάση όταν ζητηθεί. Τα παραπάνω αποτελούν ένα κομμάτι του *Μηχανισμού καθορισμού σχεσιακού σχήματος*. Το κομμάτι αυτό είναι μεταβλητό και πρέπει να προσαρμόζεται στο εκάστοτε διαχειριστικό σύστημα.

[4]Το μοντέλο, με το οποίο αποθηκεύεται στην οντοκεντρική βάση το σχεσιακό σχήμα, περιγράφεται στο κεφάλαιο 4 και στο παράρτημα Β.

3.3.2.1 Ο Μηχανισμός καθορισμού σχεσιακού σχήματος στα πλαίσια της επικοινωνίας MITOS/CM-ΚΛΕΙΩ

Σε αυτήν την ενότητα θα μιλήσουμε πιο ειδικά για το πως υλοποιήθηκε το κομμάτι του *Μηχανισμού καθορισμού σχεσιακού σχήματος* που περιγράφεται παραπάνω στα πλαίσια της επικοινωνία του συστήματος *MITOS/CM*[5] με το σύστημα *ΚΛΕΙΩ*. Σε αυτή την περίπτωση επιλέχθηκε η δημιουργία και χρήση ειδικών πινάκων οι οποίοι ενημερώνονται κάθε φορά που γίνεται μια αλλαγή στη σχεσιακή βάση με το είδος της αλλαγής[6] και τον αύξων αριθμό της, από ορισμένες επίσης ειδικές ρουτίνες που δημιουργούνται σε αυτή τη φάση. Οι πίνακες αυτοί λέγονται *πίνακες επικοινωνίας* και οι ρουτίνες *ρουτίνες επικοινωνίας*. *Πίνακες και ρουτίνες επικοινωνίας* αποθηκεύονται στη σχεσιακή βάση.

Πίνακες Επικοινωνίας

Για κάθε πίνακα που αναφέρεται στην περιγραφή του σχεσιακού σχήματος που δίνει ο χρήστης δημιουργείται ένας νέος πίνακας, ο *πίνακας επικοινωνίας*. Ο πίνακας αυτός έχει ίδιο όνομα με τον αρχικό πίνακα με τη διαφορά ότι προστίθεται ο χαρακτήρας C[7] στην αρχή του. Στον πίνακα αυτό εισάγονται αρχικά δύο πεδία, τα *id* και *action*, τα οποία είναι αντίστοιχα ο αύξων αριθμός αλλαγής[8] και το είδος της αλλαγής που συνέβη στα δεδομένα. Κατόπιν προστίθενται στον πίνακα τόσα πεδία όσα και του αρχικού πίνακα με τους ίδιους ακριβώς ορισμούς.

Ετσι για τον πίνακα *objtypes*[9], ο οποίος περιγράφει τη σχέση αντικειμένων με τους τύπους τους, που αποτελείται από τα πεδία :

objcode	objno	objsuffix	typeno	detail

θα δημιουργηθεί ο πίνακας επικοινωνίας Cobjtypes με πεδία :

id	action	objcode	objno	objsuffix	typeno	detail

Οπως αναφέρθηκε ο χρήστης καθορίζει επίσης και τα καρτεσιανά γινόμενα που ισχύουν μεταξύ των πινάκων. Έτσι γνωρίζουμε ότι ο πίνακας *objtypes* συνδέεται με τον πίνακα *types*[10] μέσω του πεδίου *typeno*[11]. Μέσω αυτού λοιπόν του πεδίου ορίζεται μια σχέση μεταξύ των πινάκων με σκοπό την απόκτηση της τιμής του πεδίου *typename* που ανήκει στον πίνακα *types* και περιγράφει το όνομα του τύπου.

Ενα τέτοιο σενάριο, οδηγεί στο να προστεθεί στον πίνακα επικοινωνίας *Cobjtypes* ένα ακόμα πεδίο με όνομα *Jtypename* το οποίο θα ενημερώνεται κάθε φορά που προστίθεται μια γραμμή στον πίνακα

[5] Όπως έχουμε ήδη αναφέρει το σύστημα MITOS/CM στηρίζεται στο διαχειριστικό σύστημα βάσεων δεδομένων SYBASE.
[6] Το είδος της αλλαγής μπορεί να είναι εισαγωγή,διαγραφή ή ενημέρωση.
[7] Ο χαρακτήρας C υποδηλώνει επικοινωνία-communication.
[8] Σαν αύξων αριθμό αλλαγής ορίζουμε τη σειρά με την οποία συμβαίνει μια αλλαγή στην οντοκεντρική βάση.
[9] Το παράδειγμα προέρχεται από την εφαρμογή του συστήματος μεταφοράς δεδομένων από το σχεσιακό σύστημα MITOS/CM στο ΚΛΕΙΩ.
[10] Ο πίνακας types περιγράφει τους διαθέσιμους τύπους αντικειμένων της βάσης MITOS/CM.
[11] Το πεδίο typeno είναι κύριο κλειδί του πίνακα *types* και ξένο κλειδί (foreigh key) του πίνακα *objtypes* (ενότητα 4.4)

επικοινωνίας, με το αποτέλεσμα του υπολογισμού του κατάλληλου καρτεσιανού γινομένου. Το πεδίο αυτό όπως φαίνεται έχει ίδιο όνομα με αυτό του πεδίου που θέλουμε να υπολογιστεί μόνο που έχει προστεθεί ο χαρακτήρας J[12] στην αρχή του. Ο ορισμός του πεδίου είναι ίδιος με αυτόν του πεδίου *typename* στον πίνακα *types*. Ετσι αν ο πίνακας *types* είναι ο παρακάτω

types	typeno	typename	typedescr	intype

Ο πίνακας επικοινωνίας Cobjtypes θα γίνει

Cobjtypes	id	action	objcode	objno	objsuffix	typeno	detail	Jtypename

Η διαδικασία επαναλαμβάνεται για κάθε καρτεσιανό γινόμενο που έχει οριστεί για τον πίνακα και μπορεί να του προσθέσει ένα ή και περισσότερα πεδία. Τέλος στον πίνακα προστίθενται ακόμα μερικά πεδία τα οποία είναι αντίγραφα αυτών που του έχουν ήδη προστεθεί. Τα πεδία αυτά έχουν ονόματα ίδια με αυτά των υπαρχόντων πεδίων μόνο που στην αρχή τους έχει προστεθεί ο χαρακτήρας O[13]. Τα πεδία αυτά χρησιμεύουν για να κρατάνε τις παλιές τιμές των δεδομένων σε περίπτωση ενημέρωσης, ενώ οι νέες τιμές των δεδομένων συμπληρώνονται στα πρώτα πεδία που αναφέραμε. Στις περιπτώσεις εισαγωγής και διαγραφής δεδομένα συμπληρώνονται μόνο στα πρώτα πεδία. Η τελική μορφή που παίρνει ο πίνακας επικοινωνίας Cobjtypes είναι:

Cobjtypes	id	action	objcode	objno	objsuffix	typeno	details	Jtypename
			Oobjcode	Oobjno	Oobjsuffix	Otypeno	Odetails	OJtypename

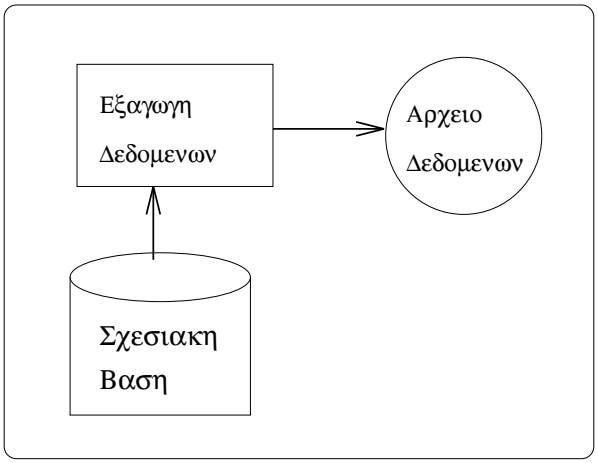
Στον πίνακα αυτόν θα αποθηκεύεται κάθε αλλαγή δεδομένων που αφορά τον αρχικό πίνακα με τη βοήθεια των ρουτινών επικοινωνίας.

Ρουτίνες Επικοινωνίας

Για κάθε πίνακα που αναφέρεται στην περιγραφή του σχεσιακού σχήματος δημιουργούνται εκτός από τον πίνακα επικοινωνίας και τέσσερις ρουτίνες. Οι τρεις από αυτές τις ρουτίνες[14] είναι υπεύθυνες για την ενημέρωση του πίνακα επικοινωνίας σε περίπτωση εισαγωγής,ενημέρωσης ή διαγραφής δεδομένων του κανονικού πίνακα. Οι ρουτίνες αυτές ενημερώνουν τον πίνακα επικοινωνίας με

- τον αύξων αριθμό πράξης,
- το είδος πράξης, I για εισαγωγή (insert), D για διαγραφή (delete) και U για ενημέρωση (update).
- τις τιμές των πεδίων του πίνακα, στον οποίο έγινε η αλλαγή

[12] Ο χαρακτήρας J υποδηλώνει join.
[13] ο χαρακτήρας O υποδηλώνει παλιό-old.
[14] Οι τρεις αυτές ρουτίνες ανήκουν στην κατηγορία ρουτινών της SYBASE που λέγονται triggers δημιουργούνται ειδικά για κάθε πίνακα, συνδέονται με μια συγκεκριμένη πράξη εισαγωγή,διαγραφή ή ενημέρωση και πυροδοτούνται αυτόματα μόλις συμβεί αυτή η πράξη στον πίνακα.



Σχήμα 3.5: Μηχανισμός εξαγωγής δεδομένων

- και τέλος υπολογίζουν τις τιμές των πεδίων που προέρχονται από τα καρτεσιανά γινόμενα του πίνακα.

Τα ονόματα των ρουτινών αυτών σχηματίζονται ως _TI≺όνομα πίνακα≻, _TD≺όνομα πίνακα≻ και _TU≺όνομα πίνακα≻, όπου ο χαρακτήρας T υποδηλώνει το είδος της ρουτίνας και οι χαρακτήρες I,D και U το είδος της πράξης (insert,delete,update).

Η τέταρτη και τελευταία ρουτίνα είναι υπεύθυνη να ενημερώσει τον πίνακα επικοινωνίας με όλα τα δεδομένα που περιέχει ο κανονικός πίνακας και χρησιμοποιείται μόνο κατά την αρχικοποίηση. Το όνομα της σχηματίζεται ως _IN≺όνομα πίνακα≻[15] και κάνει την ενημέρωση του πίνακα επικοινωνίας όπως και οι προηγούμενες ρουτίνες που περιγράφηκαν.

3.3.3 Μηχανισμός εξαγωγής δεδομένων

Ο *Μηχανισμός εξαγωγής δεδομένων* (Σχήμα 3.5) έχει σαν σκοπό να εξάγει από τη σχεσιακή βάση τα δεδομένα, που υπάρχουν στους πίνακες επικοινωνίας σε ένα ενδιάμεσο αρχείο, από το οποίο θα τα διαβάσει αργότερα ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων*.

Οταν ενεργοποιηθεί ο *Μηχανισμός εξαγωγής δεδομένων* κάνει τις παρακάτω ενέργειες :

1. Διαβάζει όλα τα δεδομένα από τους πίνακες επικοινωνίας.
2. Αντιγράφει τα δεδομένα αυτά σε ένα αρχείο με όνομα *_data_for_transfer*.
3. Τέλος για λόγους ασφάλειας παράγει ένα αντίγραφο του αρχείου των δεδομένων με όνομα *_data_for_transfer.backup*, το οποίο μπορεί να χρησιμοποιηθεί σε περίπτωση που καταστραφεί το κανονικό αρχείο (ενότητα 3.3.5.1).

[15] Οι χαρακτήρες IN προέρχονται από το initialize-αρχικοποίηση.

Αρχείο δεδομένων Η μορφή του αρχείου δεδομένων είναι συγκεκριμένη. Για κάθε γραμμή δεδομένων περιέχει

- τρεις συνεχόμενους χαρακτήρες **@**,
- το όνομα του πίνακα, στον οποίο ανήκουν τα δεδομένα,
- ένα χαρακτήρα **@**
- λίστα με τα ονόματα των πεδίων του πίνακα χωρισμένα με κόμματα
- τα δεδομένα κάθε πεδίου με τη σειρά που εμφανίζονται τα πεδία στη λίστα, με διαχωριστικό πρόθημα τρείς χαρακτήρες # και τέλος
- τρεις χαρακτήρες # για τέλος γραμμής.

Συνοπτικά η δομή του αρχείου είναι :
@@@όνομα πίνακα
@πεδίο1,...,πεδίοN
###δεδομένο πεδίου1###...###δεδομένο πεδίουN###
κ.τ.λ

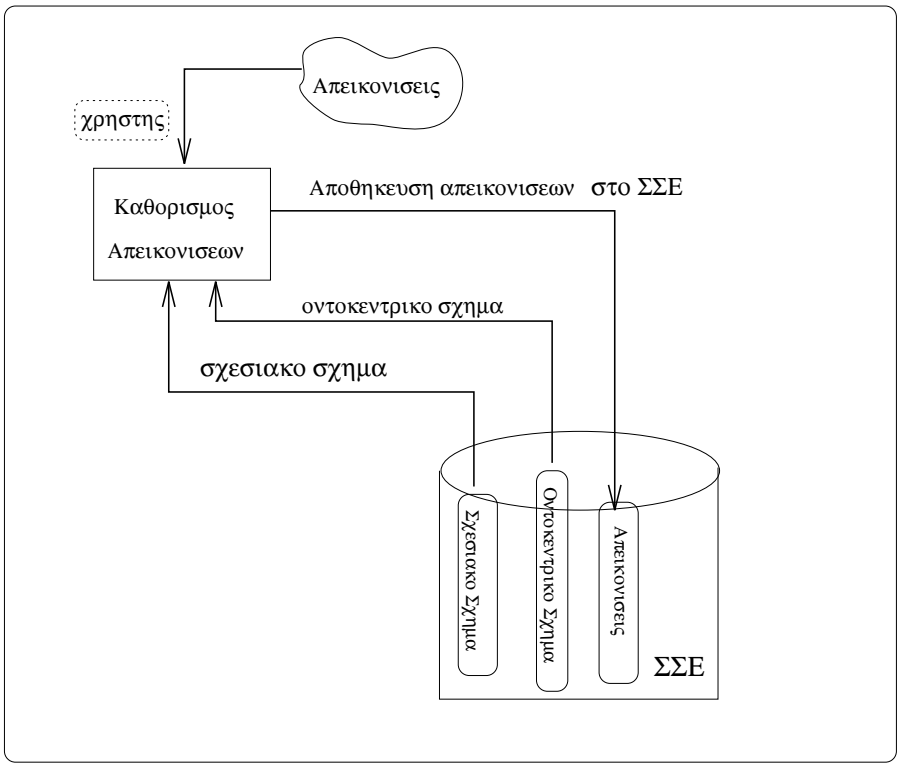
3.3.4 Μηχανισμός καθορισμού απεικονίσεων

Ο *Μηχανισμός καθορισμού απεικονίσεων* (Σχήμα 3.6) έχει σαν σκοπό να ενημερώσει το σύστημα με το **ποιά** δεδομένα θα μεταφερθούν από την σχεσιακή βάση και το **που** ακριβώς θα μεταφερθούν στην οντοκεντρική βάση. Για να γίνει όμως αυτό χρειάζεται ο χρήστης να ορίσει απεικονίσεις ανάμεσα στα δύο σχήματα.

Ο *Μηχανισμός καθορισμού απεικονίσεων* παίρνει σαν είσοδο :

1. Το σχεσιακό σχήμα, το οποίο διαβάζει από την οντοκεντρική βάση όπου έχει αποθηκευτεί από το *Μηχανισμό Καθορισμού Σχεσιακού Σχήματος*.
2. Το οντοκεντρικό σχήμα, το οποίο το διαβάζει επίσης από την οντοκεντρική βάση, όπου έχει αποθηκευτεί κατά τη δημιουργία της βάσης.
3. Τέλος ζητάει από το χρήστη να του προμηθεύσει το αρχείο με την περιγραφή των απεικονίσεων που θέλει να οριστούν ανάμεσα στα δύο αυτά σχήματα.

Κατόπιν ο *Μηχανισμός καθορισμού απεικονίσεων* ελέγχει τις απεικονίσεις που του προμήθευσε ο χρήστης για την συντακτική τους ορθότητα και για την ύπαρξη των δομών του σχεσιακού ή οντοκεντρικού σχήματος που χρησιμοποιούν. Το αποτέλεσμα αυτής της ανάλυσης είναι να παραχθεί



Σχήμα 3.6: Μηχανισμός καθορισμού απεικονίσεων

στο συντακτικό της γλώσσας εισαγωγής δεδομένων της οντοκεντρικής βάσης ο κατάλληλος κώδικας για την ενημέρωση της βάσης όταν τελειώσει η διαδικασία του ελέγχου[16].

3.3.5 Μηχανισμός μετάφρασης και μεταφοράς δεδομένων

Ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* (Σχήμα 3.7) αποτελεί τον πυρήνα του συστήματος μεταφοράς δεδομένων μιας και επιτελεί το κύριο έργο του συστήματος, τη μεταφορά των δεδομένων.

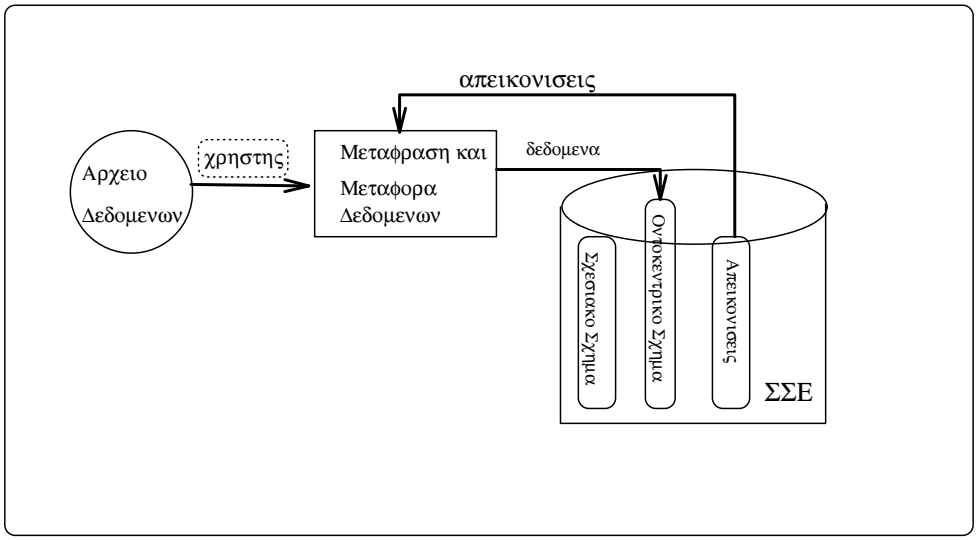
Ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* καλείται αφού έχουν οριστεί το σχεσιακό σχήμα και οι απεικονίσεις ανάμεσα στα δύο σχήματα. Εχει σαν σκοπό του, να μεταφέρει τα δεδομένα που παίρνει από τον *Μηχανισμό εξαγωγής δεδομένων* στην οντοκεντρική βάση, αφού πρώτα τα μεταφράσει στο συντακτικό της γλώσσας εισαγωγής δεδομένων της στηριζόμενος στις απεικονίσεις που έχουν οριστεί.

Οταν ενεργοποιηθεί ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* κάνει τις εξής ενέργειες.

1. Φροντίζει να έρθει η σχεσιακή βάση σε κατάσταση που να μην επιτρέπονται οι ενημερώσεις την ώρα που θα γίνεται η μεταφορά των δεδομένων (συγχρονισμός των δύο βάσεων).

2. Καλεί το *Μηχανισμό εξαγωγής δεδομένων* να εξάγει τα δεδομένα από τη σχεσιακή βάση και να

[16]Το μοντέλο, με το οποίο αποθηκεύονται στην οντοκεντρική βάση οι απεικονίσεις περιγράφεται στο κεφάλαιο 4.



Σχήμα 3.7: Μηχανισμός μετάφρασης και μεταφοράς δεδομένων

τα τοποθετήσει στο αρχείο *_data_for_transfer*.

3. Ταξινομεί το αρχείο *_data_for_transfer* σύμφωνα με τους αύξοντες αριθμούς των αλλαγών που περιέχονται. Έτσι στο νέο ταξινομημένο αρχείο τα δεδομένα ακολουθούν την ίδια ακριβώς σειρά, με την οποία παρήχθησαν στη σχεσιακή βάση.

4. Διαβάζει μια γραμμή του ταξινομημένου αρχείου.

5. Εντοπίζει το όνομα του πίνακα, στον οποίο ανήκει γραμμή

6. Εντοπίζει τη σειρά, με την οποία παρατίθενται τα δεδομένα των πεδίων του πίνακα.

7. Διαβάζει τα δεδομένα των πεδίων.

8. Εντοπίζει το είδος της πράξης (εισαγωγή,διαγραφή ή ενημέρωση).

9. Αναζητάει στην οντοκεντρική βάση τις απεικονίσεις που έχουν οριστεί για τον πίνακα αυτόν.

10. Χρησιμοποιώντας τις απεικονίσεις, που έχουν οριστεί, και το είδος της πράξης που έχει εφαρμοστεί στα δεδομένα, από το σχεσιακό σύστημα, μεταφράζει[17] τα δεδομένα και τα εισάγει στην οντοκεντρική βάση.

11. Η ενημέρωση της οντοκεντρικής βάσης :
    Εαν **επιτύχει**, ενημερώνεται ένα αρχείο που κρατάει το ιστορικό της μετάφρασης των δεδομένων για λόγους ασφάλειας και ανάκτησης δεδομένων και ο έλεγχος μεταφέρεται στο *βήμα 12*.
    Εαν **αποτύχει**, ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* παράγει μήνυμα λάθους

[17] Αναλυτικότερα για την μετάφραση πράξεων στα δεδομένα θα μιλήσουμε στο κεφάλαιο 5.

για ενημέρωση του χρήστη, περιορίζει το αρχείο δεδομένων στα δεδομένα που δεν έχουν ακόμα μεταφραστεί και πηγαίνει στο *βήμα 13.*

12. Εξετάζει αν το αρχείο των δεδομένων έχει φτάσει στο τέλος του. Εάν όχι επιστρέφει στο βήμα 4, αλλιώς πηγαίνει στο *βήμα 13.*

13. Κλείνει όσα αρχεία είναι ανοιχτά και σφραγίζει το αρχείο που κρατάει το ιστορικό της μετάφρασης των δεδομένων με τη χρονική σφραγίδα της ημερομηνίας και ώρας που άρχισε η μετάφραση.

14. Ενημερώνει τη σχεσιακή βάση ότι η διαδικασία της μεταφοράς δεδομένων τελείωσε.

Εαν συμβεί λάθος και ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* σταματήσει μένει η διαδικασία της μεταφοράς δεδομένων ανολοκλήρωτη. Για να ολοκληρωθεί πρέπει ο χρήστης να διορθώσει το λάθος, σύμφωνα με το μήνυμα που θα λάβει από το μηχανισμό και να καλέσει ένα υπομηχανισμό του *Μηχανισμού μετάφρασης και μεταφοράς δεδομένων* για να ενημερώσει την οντοκεντρική βάση με τα υπόλοιπα δεδομένα. Ο υπομηχανισμός αυτός περιλαμβάνει τα *βήματα 4-14* και πρέπει να κληθεί γιατί αλλιώς τα δεδομένα ανάμεσα στις δύο βάσεις δεν θα είναι συνεπή μια και δεν θα έχουν μεταφερθεί όλες οι αλλαγές της σχεσιακής στην οντοκεντρική βάση.

3.3.5.1 Λόγοι αποτυχίας της μεταφοράς των δεδομένων

Οι κυριότεροι λόγοι, για τους οποίους μπορεί να αποτύχει η μεταφορά των δεδομένων και ο τρόπος αντιμετώπισης τους είναι οι εξής.

- Τα δεδομένα που προσπαθούμε να εισαγάγουμε στην οντοκεντρική βάση στο βήμα 10, προϋποθέτουν ότι υπάρχει ήδη σε αυτήν πληροφορία, η οποία δεν μπορεί να εντοπιστεί, οπότε η οντοκεντρική βάση απορρίπτει τα δεδομένα και προκαλεί διακοπή της μεταφοράς τους. Αυτό δείχνει ότι έχει δημιουργηθεί *ασυνέπεια* των δύο βάσεων. Η ασυνέπεια αυτή είναι το πιο πιθανό να προέρχεται από το γεγονός ότι ένας χρήστης της οντοκεντρικής βάσης έχει αλλοιώσει δεδομένα που έχουν ήδη μεταφερθεί από τη σχεσιακή και τώρα δεν μπορούν να εντοπισθούν. Για να μην προκύψει ποτέ τέτοια περίπτωση ο χρήστης δεν πρέπει να αλλοιώνει τα δεδομένα που προέρχονται από τη σχεσιακή. Εαν συμβεί όμως κάτι τέτοιο ο χρήστης πρέπει να διορθώσει την ασυνέπεια και να ολοκληρώσει τη μεταφορά των δεδομένων.

- Το αρχείο των δεδομένων περιέχει δεδομένα που δεν έχουν δηλωθεί πουθενά. Αυτό είναι λάθος του χρήστη και πρέπει να το διορθώσει είτε αφαιρώντας τα δεδομένα είτε τροποποιώντας τον ορισμό του σχεσιακού σχήματος και των απεικονίσεων.

- Το αρχείο επικοινωνίας αλλοιώθηκε. Αυτό είναι λάθος του συστήματος και δεν μπορεί να προβλεφτεί. Σε αυτήν την περίπτωση ο χρήστης μπορεί να χρησιμοποιήσει το εφεδρικό αρχείο δεδομένων που έχει κρατηθεί (*_data_for_transfer.backup*).

- Δεν έχουν καθοριστεί οι μεταβλητές περιβάλλοντος που δηλώνουν το που βρίσκεται η σχεσιακή βάση, η οντοκεντρική βάση ή μία άλλη μεταβλητή που το σύστημα χρειάζεται για να λειτουργήσει. Αυτό είναι λάθος του χρήστη και διορθώνεται εύκολα με το να προμηθεύσει ο χρήστης τις μεταβλητές αυτές.

- Η οντοκεντρική βάση έχει πάθει ολική βλάβη. Αυτό μπορεί να διορθωθεί μόνο με *ολική ανάκτηση* της οντοκεντρικής βάσης (ενότητα 3.3.6).

- Η σχεσιακή βάση έχει πάθει ολική βλάβη. Αυτό δεν μπορεί να αντιμετωπιστεί προφανώς από το σύστημα μεταφοράς δεδομένων. Ο χρήστης θα πρέπει να επανακτήσει τη σχεσιακή του βάση και να ξεκινήσει την επικοινωνία από την αρχή.

3.3.5.2 Σύγχρονη ή ασύγχρονη επικοινωνία

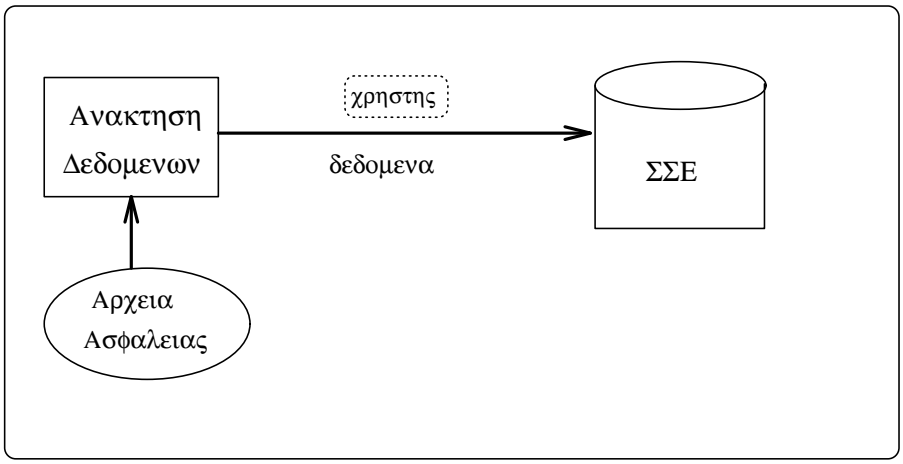
Οπως θα έχει γίνει ίσως φανερό η επικοινωνία των δύο βάσεων που υποστηρίζεται από το σύστημα είναι *ασύγχρονη. Ασύγχρονη επικοινωνία*, σημαίνει ότι οι αλλαγές στα δεδομένα της σχεσιακής δεν μεταφέρονται αμέσως στην οντοκεντρική βάση, αλλά μόνο αν το ζητήσει ο χρήστης. Το ερώτημα που τίθεται είναι γιατί η επικοινωνία να είναι ασύγχρονη και όχι σύγχρονη.

Για να απαντήσουμε το παραπάνω ερώτημα, πρέπει να δούμε πρώτα, πότε χρειάζεται να είναι η επικοινωνία σύγχρονη και πότε ασύγχρονη. Η *σύγχρονη επικοινωνία* είναι απαραίτητη όταν είναι κρίσιμο τα δύο συστήματα που επικοινωνούν να περιέχουν κάθε στιγμή τα ίδια δεδομένα. Για παράδειγμα, αν οι εφαρμογές αφορούν τις καταθέσεις πελατών μιας τράπεζας θα έπρεπε όλες οι αλλαγές που γίνονται στη μία βάση να μεταφέρονται αμέσως στην άλλη για να μην υπάρχουν ασυνέπειες. Εαν, όμως δεν μας ενδιαφέρει τα δύο συστήματα που επικοινωνούν να περιέχουν κάθε στιγμή τα ίδια δεδομένα, δεν υπάρχει πρόβλημα να γίνεται η επικοινωνία ασύγχρονα. Στην πραγματικότητα, ίσως σε μερικές περιπτώσεις να είναι καλύτερη η ασύγχρονη επικοινωνία, γιατί δεν επηρεάζονται οι χρόνοι απόδοσης των συστημάτων που συμμετέχουν.

Στην παρούσα εργασία θεωρούμε ότι δεν μας ενδιαφέρει τα δύο συστήματα που επικοινωνούν, να περιέχουν κάθε στιγμή τα ίδια δεδομένα, γιαυτό και υλοποιήθηκε ασύγχρονη επικοινωνία.

3.3.6 Μηχανισμός ανάκτησης δεδομένων οντοκεντρικής βάσης

Ο *Μηχανισμός ανάκτησης δεδομένων της οντοκεντρικής βάσης* (Σχήμα 3.8) καλείται κάθε φορά που θα διαπιστωθεί ότι οι βάσεις έχουν έρθει σε ασυνεπή κατάσταση. Η διαδικασία της επικοινωνίας των δύο βάσεων ξεκινάει πάντα με την αρχικοποίηση της οντοκεντρικής που τις φέρνει σε συνεπή κατάσταση. Από εκείνο το σημείο και μετά μεταφέρονται δεδομένα από τη σχεσιακή στην οντοκεντρική και για κάθε μεταφορά δεδομένων κρατείται από το *Μηχανισμό μετάφρασης και μεταφοράς δεδομένων* ένα αρχείο ασφαλείας που περιέχει τα δεδομένα που μεταφέρθηκαν και έχει σαν χρονική σφραγίδα την ημερομηνία μεταφοράς τους.



Σχήμα 3.8: Μηχανισμός ανάκτησης δεδομένων της οντοκεντρικής βάσης

Σε περίπτωση λοιπόν που γίνει λάθος και οι δύο βάσεις έρθουν σε ασυνεπή κατάσταση μεταξύ τους υπάρχουν δύο δυνατότητες ανάκτησης δεδομένων από την οντοκεντρική βάση : η *μερική ανάκτηση δεδομένων* και η *ολική ανάκτηση δεδομένων*, ανάλογα με το τι έχει συμβεί.

Ολική ανάκτηση δεδομένων Εάν η οντοκεντρική βάση έχει καταστραφεί ανεπανόρθωτα τότε ο χρήστης πρέπει να την επαναδημιουργήσει και κατόπιν να ξεκινήσει την διαδικασία της επικοινωνίας από την αρχή κάνοντας αρχικοποίηση της οντοκεντρικής βάσης. Για να ανακτήσει και τυχόν αλλαγές που έχουν γίνει ο χρήστης πρέπει να εισάγει στην οντοκεντρική βάση και τα αρχεία που είχαν παραχθεί ενδιάμεσα από τα *Δελτία Εισαγωγής Δεδομένων* [16],[19].

Μερική ανάκτηση δεδομένων Σε αυτή την περίπτωση απαιτείται ο συντηρητής της οντοκεντρικής βάσης να έχει πρόσφατα αντίγραφα ασφαλείας για τη βάση. Έτσι εάν απλά έχει καταστραφεί η τελευταία έκδοση της οντοκεντρικής βάσης, ο χρήστης μπορεί να την επαναδημιουργήσει στην μορφή που είχε κατά την τελευταία φορά που κρατήθηκε αντίγραφο ασφαλείας της. Κατόπιν μπορεί να την εμπλουτίσει με τις τελευταίες αλλαγές, μέσω των αρχείων ασφαλείας των μεταφορών δεδομένων, που έχουν παρεμβληθεί χρονικά από την τελευταία φορά που κρατήθηκε αυτό το αντίγραφο ασφαλείας της βάσης.

Ο εμπλουτισμός αυτός γίνεται με τη βοήθεια του *Μηχανισμού ανάκτησης δεδομένων της οντοκεντρικής βάσης*, ο οποίος παίρνει σαν είσοδο το αρχείο ασφαλείας της μεταφοράς και ενημερώνει με αυτό την οντοκεντρική βάση. Τα αρχεία ασφαλείας της μεταφοράς φυλάσσονται στον κατάλογο που δείχνει η μεταβλητή περιβάλλοντος *COM_BACKUP_DIR*.

3.4 Διαδικασία επικοινωνίας

Η διαδικασία της επικοινωνίας των δύο βάσεων περιλαμβάνει πέντε φάσεις, οι οποίες είναι : *Φάση I - Ορισμός σχεσιακού σχήματος*, *Φάση II - Καθορισμός απεικονίσεων*, *Φάση III - Αρχικοποίηση της οντοκεντρικής βάσης*, *Φάση IV - Μεταφορά δεδομένων από τη σχεσιακή στην οντοκεντρική βάση* και *Φάση V - Ανάκτηση δεδομένων από την οντοκεντρική βάση.*

Φάση I - Ορισμός σχεσιακού σχήματος Η φάση αυτή είναι πάντα η πρώτη κατά σειρά, από την οποία πρέπει να περάσει το σύστημα μεταφοράς δεδομένων. Κατά τη διάρκειά της καλείται ο *Μηχανισμός καθορισμού σχεσιακού σχήματος* και ο χρήστης δίνει την περιγραφή του σχεσιακού σχήματος.

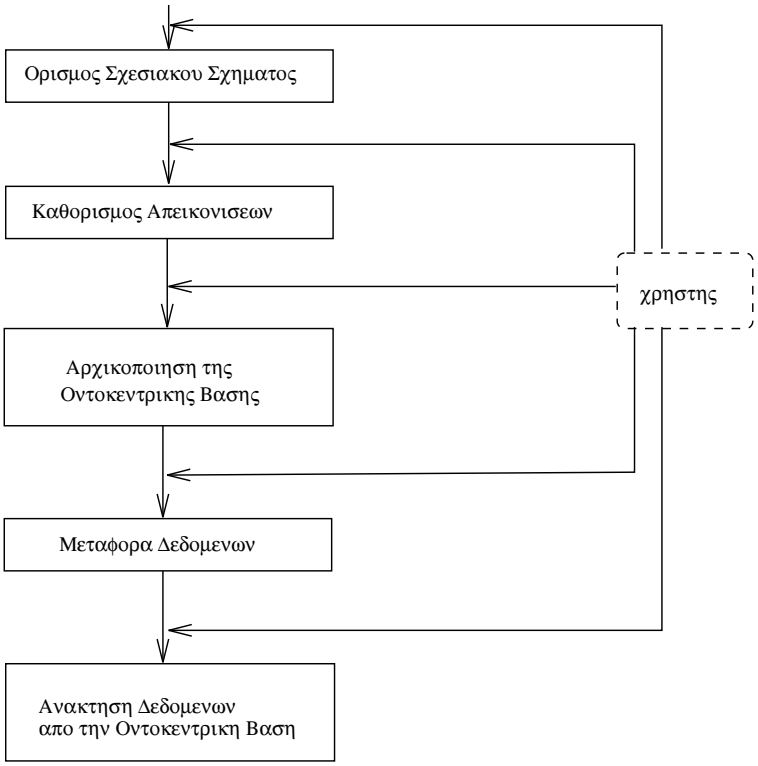
Μετά το τέλος αυτής της φάσης, οποιαδήποτε στιγμή ο χρήστης θελήσει να συμπληρώσει κάτι, που αφορά το σχεσιακό σχήμα, μπορεί να ξανακαλέσει το *Μηχανισμό καθορισμού σχεσιακού σχήματος* και να το ορίσει. Εαν, τώρα θέλησει να ακυρώσει κάτι που έχει δηλώσει πρέπει να ξεκινήσει την όλη διαδικασία της επικοινωνίας από την αρχή ή να κάνει την αλλαγή με τα *Δελτία Εισαγωγής Δεδομένων.*

Φάση II - Καθορισμός απεικονίσεων Η φάση αυτή είναι πάντα η δεύτερη κατά σειρά, από την οποία πρέπει να περάσει το σύστημα μεταφοράς δεδομένων. Κατά τη διάρκειά της καλείται ο *Μηχανισμός καθορισμού απεικονίσεων* και ο χρήστης δίνει την περιγραφή των απεικονίσεων που θέλει να οριστούν ανάμεσα στο σχεσιακό και το οντοκεντρικό σχήμα.

Μετά το τέλος αυτής της φάσης, οποιαδήποτε στιγμή ο χρήστης θέλησει να συμπληρώσει κάτι, που αφορά το σχεσιακό σχήμα, μπορεί να ξανακαλέσει το *Μηχανισμό καθορισμού απεικονίσεων* και να το ορίσει. Εαν, τώρα θέλησει να ακυρώσει κάτι που έχει δηλώσει πρέπει να ξεκινήσει την όλη διαδικασία της επικοινωνίας από την αρχή ή να κάνει την αλλαγή με τα *Δελτία Εισαγωγής Δεδομένων.*

Φάση III - Αρχικοποίηση της οντοκεντρικής βάσης Η φάση αυτή είναι πάντα η τρίτη κατά σειρά, από την οποία πρέπει να περάσει το σύστημα μεταφοράς δεδομένων. Κατά τη διάρκειά της καλείται ο *Μηχανισμός αρχικοποίησης της οντοκεντρικής Βάσης* και η οντοκεντρική βάση ενημερώνεται για τα δεδομένα, που περιέχει ήδη η σχεσιακή. Η φάση αυτή είναι πολύ σημαντική για τη συνέπεια των δεδομένων των δύο βάσεων.

Φάση IV - Μεταφορά δεδομένων από τη σχεσιακή στην οντοκεντρική βάση Η φάση αυτή επαναλαμβάνεται κάθε φορά που υπάρχουν δεδομένα να μεταφερθούν. Κάθε πετυχημένη αλλαγή στα δεδομένα της σχεσιακής φυλάσσεται έως ότου ο χρήστης ζητήσει να μεταφερθούν οι αλλαγές στην οντοκεντρική βάση. Τότε καλείται ο *Μηχανισμός μετάφρασης και μεταφοράς δεδομένων* να μεταφέρει τις αλλαγές όπως περιγράφηκε.



Σχήμα 3.9: Διάγραμμα λειτουργίας συστήματος μεταφοράς δεδομένων

Φάση V - Ανάκτηση δεδομένων της οντοκεντρικής βάσης Στη φάση αυτή μπαίνει το σύστημα μεταφοράς δεδομένων κάθε φορά που τα δεδομένα των δύο βάσεων βρεθούν ασυνεπή. Τότε καλείται ο *Μηχανισμός ανάκτησης δεδομένων της οντοκεντρικής βάσης*, ο οποίος λειτουργεί όπως περιγράφηκε (ενότητα 3.3.6).

Στο Σχήμα 3.9 μπορούμε να δούμε το συνολικό διάγραμμα λειτουργίας του συστήματος μεταφοράς δεδομένων. Οπως φαίνεται στο σχήμα, το σύστημα μεταφοράς δεδομένων μπορεί να μεταβεί σε οποιαδήποτε από τις φάσεις που αναφέρθηκαν, οποιαδήποτε στιγμή, ανάλογα με την επιλογή του χρήστη. Ο μόνος περιορισμός στην διαδοχή των φάσεων είναι ότι την πρώτη φορά, που θα λειτουργήσει το σύστημα, πρέπει να περάσει απαραίτητα από τις φάσεις I, II και III, με τη σειρά που αυτές αναφέρονται.

Κεφάλαιο 4

Τελεστές απεικόνισης

Στο κεφάλαιο 3 αναφέρθηκε, ότι για να μπορέσει να επιτευχθεί μεταφορά δεδομένων από ένα σχεσιακό σε ένα οντοκεντρικό σύστημα, πρέπει πρώτα να οριστούν απεικονίσεις ανάμεσα στα σχήματα των δύο συστημάτων. Ο ορισμός απεικονίσεων ανάμεσα σε δύο σχήματα είναι υποπρόβλημα του γενικότερου προβλήματος της μετάφρασης σχημάτων. Στο κεφάλαιο αυτό θα ασχοληθούμε με το πρόβλημα της μετάφρασης σχημάτων και τη λύση που προτείνεται στα πλαίσια αυτής της εργασίας.

Το κεφάλαιο αυτό είναι οργανωμένο ως εξής. Στην ενότητα 4.1 παρουσιάζεται το πρόβλημα της μετάφρασης σχημάτων. Στην ενότητα 4.2 παρουσιάζονται εισαγωγικές έννοιες για τον ορισμό των τελεστών απεικόνισης. Στην ενότητα 4.3 ορίζονται οι τελεστές απεικόνισης. Στην ενότητα 4.4 παρουσιάζεται ο ορισμός απεικονίσεων μέσω ξένων κλειδιών. Στην ενότητα 4.5 παρουσιάζονται θέματα ονοματοδοσίας. Τέλος στην ενότητα 4.6 παρουσιάζονται ορισμένοι περιορισμοί για την εφαρμογή των τελεστών απεικόνισης.

4.1 Το πρόβλημα της μετάφρασης σχημάτων

Ενα από τα κυριότερα προβλήματα στο χώρο των *ετερογενών βάσεων δεδομένων* είναι το πρόβλημα της μετάφρασης ενός σχήματος Σ1, εκφρασμένου σε ένα μοντέλο Μ1, σε ένα σχήμα Σ2 εκφρασμένου σε ένα άλλο μοντέλο Μ2. Στενά συνδεδεμένα με αυτό πρόβλημα είναι τα προβλήματα του συνδυασμού σχημάτων και της εύρεσης απεικονίσεων μεταξύ σχημάτων. Το πραγματικό πρόβλημα σε όλες τις περιπτώσεις είναι να εντοπιστεί εάν τα δύο σχήματα ή μέρη τους είναι ισοδύναμα.

Τα προβλήματα που παρουσιάζονται σε αυτή τη μετάφραση είναι πολλά, μερικά από αυτά αναφέρονται παρακάτω :

- Η μετάφραση ενός σχήματος από ένα μοντέλο Μ1 σε ένα άλλο μοντέλο Μ2 είναι δύσκολο να γίνει. Στην πραγματικότητα έχει γίνει πολλή δουλειά πάνω στο πρόβλημα της μετάφρασης ενός μοντέλου σε ένα άλλο. Για παράδειγμα είναι γνωστό το πως μεταφράζεται το *Μοντέλο Οντοτήτων Συσχετίσεων* στο *Σχεσιακό Μοντέλο* [13] ή το πως μεταφράζεται το *Δικτυωτό Μοντέλο*

στο *Σχεσιακό Μοντέλο* [35] αλλά δεν υπάρχουν γενικές τεχνικές μετάφρασης ενός μοντέλου σε ένα άλλο. Περισσότερες μεθοδολογίες μετάφρασης και συνδυασμού σχημάτων μπορούμε να δούμε στο [8].

- Η μετάφραση ενός σχήματος από ένα μοντέλο M1 σε ένα μοντέλο M2, που είναι λιγότερο εκφραστικό, δεν μπορεί να διατηρήσει πλήρως τη σημασιολογία του αρχικού σχήματος [58],[37].

- Τέλος δεν είναι δυνατόν να γίνει τέτοιου είδους μετάφραση χωρίς να υπάρχει σημασιολογική γνώση του σχήματος που θέλουμε να μεταφραστεί [11].

4.1.1 Προτάσεις από τη βιβλιογραφία

Το θέμα της απεικόνισης σχημάτων έχει μελετηθεί από πολλούς. Η περισσότερη δουλειά σε αυτό το θέμα έχει γίνει στο να βρεθούν απεικονίσεις από ένα πιο εκφραστικό μοντέλο σε ένα λιγότερο εκφραστικό. Συγκεκριμένα για σχήματα στο *Μοντέλο Οντοτήτων-Συσχετίσεων (ΜΟΣ)* ή επεκτάσεις αυτού (π.χ *Επεκτεταμένο Μοντέλο Οντοτήτων-Συσχετίσεων*[1] (ΕΜΟΣ)) στο *Σχεσιακό Μοντέλο* [33],[42], [63],[27],[8]. Οπως, επίσης και για σχήματα στο *Οντοκεντρικό Μοντέλο* στο *Σχεσιακό Μοντέλο* [44].

Το θέμα της απεικόνισης σχημάτων από ένα λιγότερο εκφραστικό μοντέλο σε ένα περισσότερο εκφραστικό έχει μελετηθεί λιγότερο. Επειδή στα λιγότερο εκφραστικά μοντέλα, όπως το σχεσιακό, οι δομές που χρησιμοποιούνται για την παράσταση της πληροφορίας και οι πράξεις που υποστηρίζονται είναι πολύ απλές, στις περισσότερες περιπτώσεις υπάρχει πληροφορία που δεν απεικονίζεται στο σχήμα αλλά τη γνωρίζει ο κατασκευαστής του [26]. Η πληροφορία αυτή μπορεί όμως να απεικονιστεί κατά την μεταφορά του σχήματος σε ένα περισσότερο εκφραστικό μοντέλο. Από όσο ξέρουμε σχεδόν όλες οι εργασίες που έχουν ασχοληθεί με αυτό το θέμα ασχολούνται με τη μετατροπή από το *Σχεσιακό Μοντέλο* στο *Μοντέλο Οντοτήτων-Συσχετίσεων* ή επεκτάσεις αυτού [41], [10],[48],[21],[28], [55].

Οι εργασίες που ακολουθούν, έχουν ασχοληθεί με το πρόβλημα του συνδυασμού σχημάτων διαφόρων βάσεων δεδομένων, έτσι ώστε να δημιουργηθεί μια νέα,ετερογενής βάση δεδομένων. Στις περισσότερες, από αυτές τις εργασίες, το πρόβλημα αντιμετωπίζεται σε δύο στάδια :
(1) τα σχήματα, των επιμέρους βάσεων, μεταφράζονται σε ένα εκφραστικότερο, κοινό μοντέλο [23],[62] και (2) τα σχήματα που προκύπτουν, στο κοινό μοντέλο, συνδυάζονται σε ένα σχήμα, αφού πρώτα λυθούν ότι ετερογένειες υπάρχουν ανάμεσα τους (ενότητα 2.4).
Κατά τη διάρκεια του πρώτου βήματος, παρατηρούμε ότι γίνεται προσπάθεια να εμπλουτιστεί το νέο σχήμα με πληροφορία, που δεν ήταν δυνατόν να απεικονιστεί στο αρχικό σχήμα, λόγω της περιορισμένης εκφραστικότητας του αρχικού μοντέλου.

Στην εργασία [41] αναπτύσσεται μια διαδικασία με την οποία μπορεί να αντληθεί από μία σχεσιακή βάση δεδομένων η οντοκεντρική περιγραφή της. Η διαδικασία αυτή αφορά συγκεκριμένα την

[1] Το Επεκτεταμένο Μοντέλο Οντοτήτων-Συσχετίσεων περιέχει επιπλέον των βασικών δομών του ΜΟΣ δυνατότητες για συνάθροιση και γενίκευση τις οποίες βρίσκουμε συνήθως μόνο στα οντοκεντρικά μοντέλα.

προσπάθεια μετάφρασης σχημάτων στο *Σχεσιακό Μοντέλο* σε σχήματα στο *Επεκτεταμένο Μοντέλο Οντοτήτων-Συσχετίσεων*. Πρώτα μετατρέπεται το σχεσιακό σχήμα σε μορφή κατάλληλη για να εντοπισθούν δομικά στοιχεία του ΕΜΟΣ μοντέλου. Για να επιτευχθεί αυτό εντοπίζονται και απαλείφονται αρχικά συναρτησιακές εξαρτήσεις τύπου ‘‘περιέχεται’’ που σχηματίζουν κύκλο. Επίσης εάν υπάρχουν σχέσεις που μπορεί να είχαν συγχωνευθεί χωρίζονται. Στο επόμενο στάδιο κάθε σχέση του σχήματος που έχει προκύψει μετατρέπεται στην κατάλληλη δομή του ΕΜΟΣ μοντέλου. Ο τύπος της κάθε δομής του ΕΜΟΣ μοντέλου που προκύπτει από το αρχικό σχήμα και των σχέσεων μεταξύ τους αντλείται από την εξέταση των συναρτησιακών εξαρτήσεων τύπου ‘‘περιέχεται’’ και τη δομή των κλειδιών του σχεσιακού σχήματος. Η διαδικασία αυτή μπορεί να χρησιμοποιηθεί για την ανάλυση της σημασιολογίας μιάς υπάρχουσας βάσης δεδομένων και για τη μετατροπή σχεσιακών σχημάτων σε οντοκεντρικά. Τέλος στην εργασία [40] παρουσιάζεται και η αντίστροφη διαδικασία μετάφρασης σχημάτων από το ΕΜΟΣ μοντέλο στο σχεσιακό.

Στην εργασία [37],[39] παρουσιάζεται μια γενική και ημιαυτόματη μέθοδος για μετατροπή σχεσιακών σχημάτων στο οντοκεντρικό μοντέλο BLOOM (Barcelona Object Oriented Model) [36]. Για την εφαρμογή της μεθόδου δεν χρειάζεται να γίνουν πολλές υποθέσεις και απλοποιήσεις όπως το να θεωρηθεί ότι το σχεσιακό σχήμα είναι καλά σχεδιασμένο. Η μέθοδος έχει δύο στάδια : *στάδιο απόκτησης γνώσης* και *στάδιο μετατροπής σχεσιακού σχήματος*. Στο πρώτο στάδιο προσπαθεί να ανακαλύψει την ακριβή περιγραφή των αντικειμένων του σχεσιακού σχήματος και τις σχέσεις μεταξύ τους. Για να το πετύχει αυτό (α) εξετάζει τα κλειδιά της κάθε σχέσης, (β) εξετάζει τις συναρτησιακές εξαρτήσεις που ισχύουν [12] και (γ) κανονικοποιεί τις σχέσεις σε τρίτη κανονική μορφή (3NF). Στο δεύτερο στάδιο γίνεται μετατροπή του σχεσιακού σχήματος σε οντοκεντρικό με βάση τα δεδομένα του πρώτου σταδίου και τη βοήθεια του χρήστη όπου χρειάζεται. Ανάμεσα στις εφαρμογές της μεθόδου είναι η περίπτωση βάσεων δεδομένων που θέλουμε να συνεργαστούν για να μοιράζονται τα δεδομένα τους διατηρώντας ταυτόχρονα την αυτονομία τους.

Στην εργασία [46] παρουσιάζεται μια μεθοδολογία που καλύπτει συνδυασμό σχημάτων και βάσεων δεδομένων. Χρησιμοποιεί μια αρχιτεκτονική τεσσάρων επιπέδων (local schemata,local object schemata, global schema και global view schemata) όπου κάθε επίπεδο αναπαριστά μια συνδυασμένη όψη των ιδεών που χαρακτηρίζουν το αμέσως κατώτερο του επίπεδο. Εδώ η προσθήκη σημασιολογικής γνώσης στα υπάρχοντα σχήματα γίνεται κατά την ανάπτυξη των ενδιάμεσων σχημάτων αντικειμένων (local object schemata) τα οποία έχουν σκοπό (α) να εκφραστούν σε ένα ενδιάμεσο κοινό μοντέλο όλα τα σχήματα έτσι ώστε να απλοποιηθεί η διαδικασία της συγχώνευσης τους και (β) να αποθηκευτεί σε αυτά η σημασιολογία που συλλέχθηκε για τα αντικείμενα του κατώτερου επιπέδου. Τελικά τα σχήματα συνδυάζονται σε ένα ολικό σχήμα αφού εντοπιστούν και απαλειφθούν κύκλοι ή πλεονασμοί που μπορεί να υπάρχουν.

Αυτό που παρατηρείται στις διάφορες εργασίες που έχουν γίνει πάνω σε αυτά τα θέματα είναι ότι οι περισσότερες επιζητούν να λύσουν το πρόβλημα της μετατροπής ενός σχήματος, σε κάποιο μοντέλο,

σε ένα ισοδύναμο σχήμα σε κάποιο άλλο μοντέλο έχοντας σαν δεδομένο ότι το δεύτερο σχήμα θα κατασκευαστεί από την αρχή. Υπάρχουν όμως και οι περιπτώσεις που στο μοντέλο-στόχο υπάρχει ήδη κάποιο σχήμα και αυτό που χρειάζεται είναι ένας απλός καθορισμός απεικονίσεων μεταξύ των δύο σχημάτων. Και περιπτώσεις που δεν μας ενδιαφέρει να απεικονίσουμε ολόκληρο το ένα σχήμα στο άλλο αλλά μέρη του ενός με μέρη του άλλου.

Το σύστημα Pegasus [60],[56],[18],[29] είναι ένα σύστημα διαχείρισης ετερογενών βάσεων δεδομένων το οποίο αναπτύχθηκε στα εργαστήρια της Hewlett-Packard. Οι βάσεις δεδομένων που συμμετέχουν στο Pegasus είναι αυτόνομες και βρίσκονται κάτω από τον έλεγχο του ίδιου ή ανεξάρτητων συστημάτων διαχείρισης βάσεων δεδομένων. Επίσης οι βάσεις αυτές είναι ετερογενείς σε επίπεδο συστήματος και σημασιολογίας. Το Pegasus παρέχει ένα οντοκεντρικό μοντέλο σαν βάση για την ομοιόμορφη συνεργασία των βάσεων αυτών. Το σύστημα Pegasus δίνει τη δυνατότητα στο χρήστη με ειδικούς τελεστές να απεικονίσει μέρη του σχήματος-πηγή (σχεσιακά, ιεραρχικά, δικτυωτά σχήματα) σε δομές του σχήματος του Pegasus προσθέτοντας, αν επιθυμεί, σημασιολογική πληροφορία για εμπλουτισμό του υπάρχοντος σχήματος ή αφαιρώντας ότι πληροφορία θέλει.

Η γλώσσα ορισμού και διαχείρισης δεδομένων του Pegasus είναι η HOSQL [2]. Η HOSQL παρέχει μηχανισμό για καθορισμό απεικονίσεων μεταξύ του μοντέλου βάσης-πηγής και του μοντέλου του Pegasus. Επίσης παρέχει μηχανισμό μετάφρασης ερωτήσεων εκφρασμένων σε HOSQL στην γλώσσα της βάσης-πηγής. Ο μηχανισμός για καθορισμό απεικονίσεων δίνει τη δυνατότητα κατασκευής νέων τύπων και συναρτήσεων[3] στο μοντέλο του Pegasus, των οποίων τον ορισμό αντλεί από αντίστοιχες δομές της βάσης-πηγής. Επίσης παρέχει τη δυνατότητα να μην δημιουργηθεί ένας νέος τύπος αλλά να γίνει απεικόνιση σε υπάρχοντες τύπους. Ετσι για παράδειγμα αν η βάση πηγή περιγράφει εργαζόμενους και περιέχει τον πίνακα *employee* με γνωρίσματα τον αριθμό υπαλλήλου (employeeNo) και τον μισθό του (salary) μπορούμε να αντιστοιχήσουμε τον αριθμό υπαλλήλου με ένα νέο τύπο του Pegasus *Employee*. Ο νέος αυτός τύπος θα παίρνει τα δεδομένα του από τον αντίστοιχο πίνακα στη βάση. Ομοια μπορεί να κατασκευαστεί και μια συνάρτηση υπολογισμού του μισθού του τύπου Employee που θα θα παίρνει τα δεδομένα της από τον αντίστοιχο πίνακα στη βάση και πάλι.

Οι τύποι και οι συναρτήσεις που κατασκευάζονται με αυτόν τον τρόπο ενσωματώνονται στο σχήμα του Pegasus και ο χρήστης μπορεί να συνεχίσει να επεμβαίνει πάνω τους και να ορίσει εξαρτήσεις με τους υπάρχοντες τύπους. Η απεικόνιση και η μετάφραση σχεσιακών σχημάτων μπορεί να γίνει πλήρως ή ημιαυτόματα από ειδικά εργαλεία [29] με εξέταση των συναρτησιακών εξαρτήσεων τύπου ‘‘περιέχεται’’ και των συναρτησιακών εξαρτήσεων ανάμεσα στα ξένα κλειδιά των σχέσεων.

Στο [6] παρουσιάζεται ένα σύστημα για μεταφορά δεδομένων από μια ιεραρχική βάση IMS σε μία σχεσιακή βάση δεδομένων DB2(και αντιστρόφως), το οποίο κατασκευάστηκε από την IBM σε συνεργασία με την Τράπεζα της Ελβετίας[4]. Ιδέα του συστήματος αυτού είναι η κατασκευή ενός

[2]Heterogeneous Object Structured Query Language

[3]Τύποι,συναρτήσεις και αντικείμενα (types,functions,objects) είναι οι βασικές δομές του μοντέλου του Pegasus.

[4]Swiss Bank Corporation

μεταφορέα δεδομένων, που ονομάζεται *IBM Data Propagator MVS/ESA*, ο οποίος στέκεται ανάμεσα στις δύο βάσεις και στέλνει ότι αλλαγές γίνονται στη μία, στην άλλη. Στο σύστημα αυτό πριν αρχίσει η μεταφορά των δεδομένων, ο διαχειριστής της βάσης δεδομένων πρέπει να αποφασίσει ποιά δεδομένα θα σταλούν και που. Για να το κάνει αυτό κατασκευάζει απεικονίσεις με χρήση ειδικών τελεστών και τις σώζει σε ένα ειδικό κατάλογο. Για να δημιουργήσει αυτές τις απεικονίσεις επιλέγει μέρη της ιεραρχικής βάσης και μέρη της σχεσιακής βάσης. Κατόπιν εφαρμόζει κανόνες απεικόνισης ανάμεσα τους για να τα συσχετίσει. Υπάρχουν 5 γενικοί κανόνες απεικόνισης (single segment to single relation, single segment and extensions to single relation, single segment to separate relations, segment group to separate relations, fields from segments into a single relation).

Στο [47] περιγράφεται μια γλώσσα μετάφρασης δεδομένων, η CONVERT. Η CONVERT υποστηρίζει δύο κατηγορίες πράξεων : πράξεις απεικόνισης δεδομένων και πράξεις επικύρωσης δεδομένων. Για να γίνουν οι πράξεις αυτές παρέχεται ένα σύνολο τελεστών τους οποίους υποθέτει ότι θα τους χρησιμοποιήσει κάποιος με καλή γνώση των δεδομένων και ικανός να χρησιμοποιήσει το συντακτικό της. Η CONVERT είναι κατασκευασμένη για να χειρίζεται ιεραρχικά δομημένα δεδομένα αλλά μπορεί να εφαρμοστεί και σε άλλου είδους δεδομένα.

4.1.2 Ισοδυναμία σχημάτων

Οπως αναφέρθηκε, στην εισαγωγή της ενότητας 4.1, το πραγματικό πρόβλημα στην απεικόνιση σχημάτων είναι να εντοπιστεί, εάν τα δύο σχήματα είναι ισοδύναμα. Το πρόβλημα αυτό είναι εξίσου ενδιαφέρον, στις περιπτώσεις της μετάφρασης ενός σχήματος σε ένα άλλο και στις περιπτώσεις που θέλουμε να κάνουμε συνδυασμό σχημάτων. Πάνω στο θέμα αυτό έχουν γίνει πολλές εργασίες και έχουν δοθεί πολλοί, μη-συμβατοί ορισμοί για την ισοδυναμία σχημάτων.Κάποιες αναφορές πάνω στο θέμα μπορούν να βρεθούν στα [7],[35],[52],[58].

4.2 Εισαγωγικές έννοιες για τον ορισμό των τελεστών απεικόνισης

Εχουν γίνει πολλές εργασίες πάνω στο θέμα του συνδυασμού ετερογενών βάσεων δεδομένων, έτσι ώστε να πάρουμε τελικά ένα σύστημα στο οποίο οι βάσεις αυτές να συνεργάζονται διατηρώντας την αυτονομία τους. Για να επιτευχθεί κάτι τέτοιο ακολουθούνται διάφορες προσεγγίσεις. Στις περισσότερες από αυτές αυτό που γίνεται είναι να συνδυάζονται όλα τα σχήματα σε ένα κοινό σχήμα με διάφορες μεθόδους. Η παραγωγή αυτού του σχήματος μπορεί να είναι αυτόματη,ημιαυτόματη ή να γίνεται από έναν ειδικό χρήστη. Το θέμα που εξετάζεται στην παρούσα εργασία αναφέρεται στην περίπτωση που στο σύστημα-στόχο[5] υπάρχει ήδη ένα σχήμα. Εμείς θέλουμε να ορίσουμε απεικονίσεις ανάμεσα στο σχήμα του συστήματος-πηγής[6] στο σχήμα του συστήματος-στόχου. Δεν θέλουμε να δημιουργηθεί ένα

[5]Σαν σύστημα-στόχο ορίζουμε το σύστημα στο οποίο θέλουμε να μεταφερθούν τα δεδομένα.
[6]Σαν σύστημα-πηγή ορίζουμε το σύστημα από το οποίο θέλουμε να πάρουμε τα δεδομένα.

νέο σχήμα σαν συνδυασμός των δύο σχημάτων που ήδη υπάρχουν αλλά μια απεικόνιση ανάμεσά τους. Σαν μοντέλο-πηγή θεωρούμε το σχεσιακό μοντέλο και σαν μοντέλο-στόχο το οντοκεντρικό [20],[31],[30],[64].

Ο καθορισμός απεικονίσεων από ένα σχεσιακό σε ένα οντοκεντρικό μοντέλο ανήκει στην κατηγορία όπου το μοντέλο-πηγή είναι λιγότερο εκφραστικό από το μοντέλο-στόχο. Για αυτό το λόγο είναι χρήσιμο να προηγηθεί μια φάση όπου το οντοκεντρικό σύστημα θα πρέπει να αποκτήσει γνώση για το σχεσιακό σχήμα και τη σημασιολογία του, πριν καθοριστούν οι απεικονίσεις μεταξύ των δύο σχημάτων. Ο χρήστης[7] με βάση την γνώση του για το σύστημα θα καθορίσει τις παραπάνω πληροφορίες και τις σημασιολογικές λεπτομέρειες που δεν είναι προφανείς. Για να ορισθούν τώρα οι απεικονίσεις μεταξύ των δύο σχημάτων παρέχεται ένα σύνολο τελεστών απεικόνισης, οι οποίοι αυτό που κάνουν είναι να παίρνουν σαν είσοδο δομές του σχεσιακού σχήματος και να τις απεικονίζουν σε δομές του οντοκεντρικού. Στην προσέγγιση που προτείνουμε τις απεικονίσεις τις κάνει ο ίδιος ο χρήστης με βάση το σχεσιακό σχήμα, τη σημασιολογία του σχεσιακού σχήματος και το οντοκεντρικό σχήμα. Οι απεικονίσεις αυτές δεν επιφέρουν καμία αλλαγή σε κανένα από τα δύο αρχικά σχήματα. Βέβαια για να μπορούν να οριστούν απεικονίσεις ανάμεσα στα δύο σχήματα πρέπει να υπάρχουν ήδη ορισμένες σε αυτά οι δομές που θα αντιστοιχιστούν. Για παράδειγμα αν στο ένα σχήμα υπάρχει η δομή *Διεύθυνση* και στο άλλο δεν υπάρχει η ανάλογη δομή καμία απεικόνιση δεν μπορεί να οριστεί. Τέλος πρέπει να τονιστεί ότι το σχεσιακό σχήμα στο οποίο θα εφαρμοστούν οι τελεστές πρέπει να είναι κανονικοποιημένο σε κανονική μορφή τουλάχιστον 3NF για να είναι σίγουρη η απόκτηση απεικονίσεων που θα εκφράζουν αυτό ακριβώς που θέλει ο χρήστης. Η εφαρμογή των τελεστών σε μη κανονικοποιημένα σχήματα δεν είναι αδύνατη αλλά δεν είναι σίγουρο ότι θα έχει τα επιθυμητά αποτελέσματα (ενότητα 4.6.1).

Παρόμοιες εργασίες με αυτήν που παρουσιάζεται, σε αυτό το κεφάλαιο, για τον καθορισμό των τελεστών μπορούμε να δούμε στο σύστημα [6] (ενότητα 4.1.1) όπου παρέχεται ένα σύνολο τελεστών για απεικονίσεις ανάμεσα στο ιεραρχικό και το σχεσιακό μοντέλο. Επίσης σε άλλα συστήματα, όπως το Pegasus, δίνεται η δυνατότητα με ειδικούς τελεστές να επεκταθεί το σχήμα του συστήματος με την εισαγωγή νέων σχημάτων (ενότητα 4.1.1). Τέλος μια ακόμα σχετική εργασία αποτελεί το [37],[39] (ενότητα 4.1.1). Στην εργασία αυτή παρουσιάζεται μια γενική και ημιαυτόματη μέθοδος για μετατροπή σχεσιακών σχημάτων στο οντοκεντρικό μοντέλο BLOOM. Η διαφορά της με την παρούσα εργασία είναι οτι οι απεικονίσεις δεν ορίζονται από το χρήστη, με τη χρήση ενός συνόλου τελεστών, αλλά εξάγονται ημιαυτόματα με βάση τις συναρτησιακές εξαρτήσεις που ισχύουν, τα κλειδιά των πινάκων κ.τ.λ. Βέβαια και σε αυτή την εργασία η παρουσία του χρήστη είναι απαραίτητη σε περιπτώσεις που το σύστημα δεν μπορεί να αποφασίσει.

Στην ανάπτυξη του συνόλου των τελεστών έπαιξε μεγάλο ρόλο η συνεργασία με το Μουσείο Μπενάκη. Στα πλαίσια της συνεργασίας αυτής [1], [4] εντοπίστηκαν διάφορες ανάγκες και προβλήματα

[7]Με τον όρο χρήστης εννοούμε σε αυτό το κεφάλαιο κάποιον που έχει καλή γνώση της σχεσιακής και της οντοκεντρικής βάσης.

κατά την προσπάθεια να επικοινωνήσουν δύο πραγματικά πληροφοριακά συστήματα. Οι ανάγκες και τα προβλήματα αυτά λήφθηκαν υπόψην στη σχεδίαση των τελεστών.

4.2.1 Αποθήκευση σχεσιακού σχήματος στην οντοκεντρική βάση

Για να εφοδιάσουμε το οντοκεντρικό σύστημα με τη γνώση του σχεσιακού σχήματος και της σημασιολογίας του απεικονίζουμε αρχικά το σχεσιακό σχήμα στο οντοκεντρικό μοντέλο. Η απεικόνιση αυτή περιλαμβάνει καθορισμό :

- της σχεσιακής βάσης από όπου προέρχεται το σχήμα,
- των πινάκων του σχεσιακού σχήματος που συμμετέχουν στην επικοινωνία,
- των πεδίων του κάθε πίνακα,
- του κύριου κλειδιού κάθε πίνακα ,
- των σχέσεων ή καρτεσιανών γινομένων που ισχύουν ανάμεσα στους πίνακες και
- των σημασιολογικών συσχετίσεων που ισχύουν ανάμεσα στους πίνακες. Οι συσχετίσεις αυτές μπορεί να είναι τριών ειδών [46] :

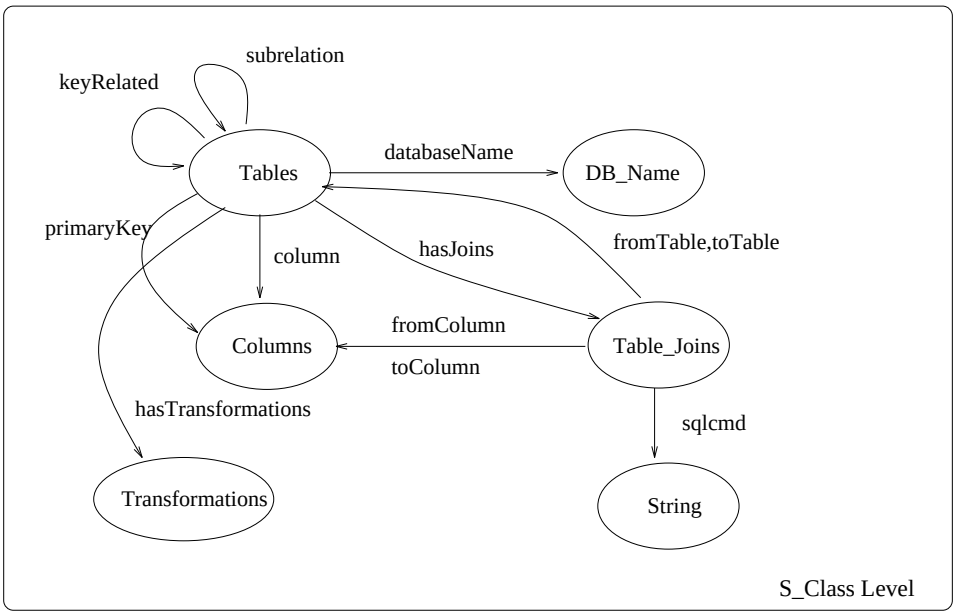
  1. *Σχέση ισοδυναμίας*
     αφορά πίνακες και σημαίνει ότι ένας πίνακας είναι ισοδύναμος με έναν άλλο.
  2. *Σχέση υποσυνόλου*
     αφορά πίνακες, σημαίνει ότι ένας πίνακας είναι υποσύνολο ενός άλλου και εκφράζει συναρτησιακές εξαρτήσεις τύπου *''περιέχεται''*
     π.χ FACULTY $\subset$ EMPLOYEE .
  3. *Σχέση Κλειδιών*
     αφορά γνωρίσματα πινάκων και εκφράζει συναρτησιακές εξαρτήσεις κλειδιών, όπου το κύριο κλειδί ενός πίνακα χρησιμοποιείται ως ξένο κλειδί σε έναν άλλο(-ους) πίνακα(-ες).

Στην παρούσα εργασία θα θεωρήσουμε ότι στις σχεσιακές βάσεις που εξετάζουμε ισχύουν μόνο οι συσχετίσεις (2) και (3) μια και είναι πλεονασμός σε μια βάση δεδομένων να ισχύει η (1).

Για την αποθήκευση του σχεσιακού σχήματος στην οντοκεντρική βάση δημιουργήσαμε το μοντέλο που φαίνεται στο Σχήμα 4.1. Οπως φαίνεται στο σχήμα το μοντέλο έχει περιγραφεί ολόκληρο σε S_Class επίπεδο [5].

Στο μοντέλο αυτό η κλάση *DB_Name* περιγράφει τα ονόματα των σχεσιακών βάσεων, των οποίων τα μοντέλα έχουν εισαχθεί στην οντοκεντρική βάση.

Η κλάση *Tables* περιγράφει τους πίνακες της σχεσιακής βάσης και έχει τα γνωρίσματα :



Σχήμα 4.1: Μοντέλο αποθήκευσης σχεσιακού σχήματος στην οντοκεντρική βάση

- *column*
  συνδέει τον πίνακα με τα πεδία που τον αποτελούν και παίρνει τιμές στην κλάση *Columns.*

- *primaryKey*
  συνδέει τον πίνακα με τα πεδία που αποτελούν το κύριο κλειδί του και παίρνει τιμές στην κλάση *Columns.*

- *databaseName*
  συνδέει τον πίνακα με τη βάση δεδομένων από όπου προέρχεται και παίρνει τιμές στην κλάση *DB_Name.*

- *subrelation*
  δείχνει τη σχέση του πίνακα με άλλους πίνακες μέσω της *σχέσης υποσυνόλου* και παίρνει τιμές στην κλάση *Tables.*

- *keyRelated*
  δείχνει τη σχέση του πίνακα με άλλους πίνακες μέσω της *σχέσης κλειδιών* και παίρνει τιμές στην κλάση *Tables.*

- *hasJoins*
  συνδέει τον πίνακα με τα καρτεσιανά γινόμενα που έχουν οριστεί από αυτόν προς άλλους πίνακες και παίρνει τιμές στην κλάση *Table_Joins.*

- *hasTransformations*
  συνδέει τον πίνακα με τις απεικονίσεις που έχουν οριστεί από αυτόν σε δομές του οντοκεντρικού σχήματος και παίρνει τιμές στην κλάση *Transformations.*

Η κλάση *Table_Joins* περιγράφει τα καρτεσιανά γινόμενα που έχουν δηλωθεί ότι ισχύουν ανάμεσα σε πίνακες της σχεσιακής βάσης και έχει τα γνωρίσματα :

- *fromTable*
  δείχνει τον πίνακα από όπου αρχίζει ο υπολογισμός του καρτεσιανού γινομένου και παίρνει τιμές στην κλάση *Tables.*

- *fromColumn*
  δείχνει το πεδίο ή τα πεδία, για τα οποία θα υπολογιστεί το καρτεσιανό γινόμενο και παίρνει τιμές στην κλάση *Columns.*

- *toTable*
  δείχνει τον πίνακα όπου καταλήγει ο υπολογισμός του καρτεσιανού γινομένου και παίρνει τιμές στην κλάση *Tables.*

- *toColumn*
  δείχνει το πεδίο ή τα πεδία, τα οποία θα υπολογιστούν από το καρτεσιανό γινόμενο και παίρνει τιμές στην κλάση *Columns.*

- *sqlcmd*
  δείχνει σε κείμενο που είναι κώδικας γραμμένος σε SQL και υπολογίζει το καρτεσιανό γινόμενο αν εφαρμοστεί στη σχεσιακή βάση.

Η διατύπωση του μοντέλου αυτού σε Telos φαίνεται στο παράρτημα Β, όπου φαίνεται επίσης και το συντακτικό, με το οποίο ο χρήστης καθορίζει το σχεσιακό σχήμα.

Με βάση το μοντέλο, που περιγράφηκε, μπορούν να αποθηκευτούν παραπάνω από ένα σχεσιακά σχήματα, στην οντοκεντρική βάση και για το καθένα από αυτά να οριστούν οι δικές του απεικονίσεις στο οντοκεντρικό σχήμα.

4.2.2 Χρήση προθήματος και επιθήματος κατά τον ορισμό των απεικονίσεων

Ο ορισμός όλων των απεικονίσεων υποστηρίζει τη χρήση *προθήματος* και *επιθήματος.* Ο ορισμός *προθήματος* και *επιθήματος* είναι προαιρετικός και μπορεί να οριστεί μόνο το ένα από τα δύο. Ορίζονται από το χρήστη μαζί με την απεικόνιση στην οποία ανήκουν.

Η εισαγωγή *προθήματος* και *επιθήματος* είναι χρήσιμη κατά την μεταφορά δεδομένων γιατί :

1. Βοηθάει στον έλεγχο των στοιχείων [4].

2. Ενας ακόμα λόγος που υποστηρίζεται η χρήση *προθήματος* και *επιθήματος* είναι ότι βοηθάνε στην αποφυγή ‘‘ταυτωνυμιών’’ [4]. Συγκεκριμένα, τα αντικείμενα στην οντοκεντρική βάση πρέπει να έχουν μοναδικό λογικό όνομα. Κατά την ανάπτυξη όμως των τελεστών βρέθηκαν περιπτώσεις που αυτό παραβιάζονταν. Για παράδειγμα, κατά τη συνεργασία με το Μουσείο Μπενάκη και τη συλλογή απαιτήσεων για την επίτευξη της επικοινωνίας, διαπιστώθηκε ότι υπήρχαν δεδομένα που έπρεπε να εισαχθούν πολλές φορές στην οντοκεντρική βάση, με το ίδιο λογικό όνομα, αντιπροσωπεύοντας όμως κάτι διαφορετικό κάθε φορά. Ετσι παρουσιάστηκε η ανάγκη το δεδομένο ‘‘γλυπτό’’ να πρέπει να εισαχθεί σαν τεχνική και σαν είδος. Με την προσθήκη, όμως προθήματος το πρόβλημα λύνεται και έχουμε π.χ είδος:γλυπτό και τεχνική:γλυπτό.

3. Τέλος, με τη χρήση *προθήματος* ή *επιθήματος* στο όνομα των δεδομένων (ενότητα 4.5) γίνεται πιο εύκολη η αναγνώριση ενός δεδομένου και του τι αντιπροσωπεύει.

Το συντακτικό ορισμού *προθήματος* και *επιθήματος* παρουσιάζεται στο παράρτημα Γ αλλά αναφέρεται και κατά των ορισμό των απεικονίσεων στις επόμενες ενότητες.

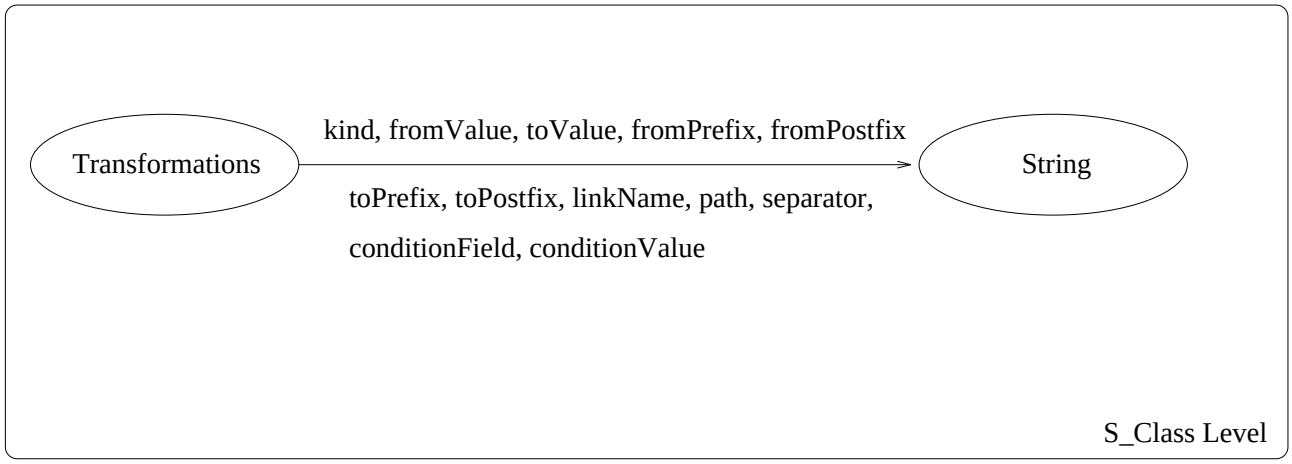
4.2.3 Χρήση της έννοιας της συνθήκης κατά τον ορισμό των απεικονίσεων

Ο ορισμός των απεικονίσεων υποστηρίζει την έννοια της *συνθήκης*. Η χρήση της έννοιας αυτής είναι απαραίτητη σε βάσεις δεδομένων που τυχαίνει σε έναν ή περισσότερους πίνακες οι τιμές ενός πεδίου να εξαρτώνται από τις τιμές ενός άλλου πεδίου του ίδιου πίνακα. Το φαινόμενο αυτό συμβαίνει σε βάσεις που το σχήμα τους δεν βρίσκεται σε τρίτη κανονική μορφή (3NF).

Επίσης μπορεί να εμφανιστεί η ανάγκη, τα δεδομένα ενός ή περισσοτέρων πεδίων ενός πίνακα, να πρέπει να μεταφερθούν σε διαφορετικά σημεία της οντοκεντρικής βάσης, ανάλογα με τις τιμές που παίρνει ένα άλλο πεδίο, του ίδιου πίνακα. Για παράδειγμα, στον πίνακα περιγραφής χρονολογιών *chronologies* του συστήματος MITOS/CM [1] ανάλογα με τις τιμές του πεδίου *chronkind* (είδος χρονολογίας) τα δεδομένα των υπολοίπων πεδίων του πίνακα μπορεί να είναι περίοδοι, χρονολογίες προ Χριστού (π.Χ) ή χρονολογίες μετά Χριστό (μ.Χ) και πρέπει να μεταφερθούν σε διαφορετικά σημεία της οντοκεντρικής βάσης.

Η καλύτερη αλλά και πιο επίπονη λύση σε αυτές τις περιπτώσεις είναι να γίνει κανονικοποίηση της βάσης έτσι ώστε να λυθούν τέτοιου είδους προβλήματα. Επειδή όμως αυτό μπορεί να είναι δύσκολο να γίνει, αποφασίστηκε να υποστηριχθεί η έννοια της *συνθήκης* για να αντιμετωπίσει ο χρήστης παρόμοιες περιπτώσεις. Βέβαια, όπως θα δούμε στην ενότητα 4.6.1, ακόμα και με τη χρήση της έννοιας της *συνθήκης* δε λύνονται πάντα όλα τα προβλήματα κακού σχεδιασμού μιας βάσης δεδομένων.

Ο ορισμός της *συνθήκης* γίνεται μαζί με τον ορισμό της αντίστοιχης απεικόνισης. Επιτρέπεται η δήλωση μίας μόνο *συνθήκης* ανά απεικόνιση. Οι *συνθήκες* που μπορούν να οριστούν περιλαμβάνουν μόνο ισότητα και δεν υποστηρίζουν τους λογικούς τελεστές ΚΑΙ (AND),Η (OR) και ΟΧΙ (NOT). Το συντακτικό ορισμού της *συνθήκης* είναι το ίδιο για όλες τις απεικονίσεις και είναι το εξής :



Σχήμα 4.2: Μοντέλο αποθήκευσης απεικονίσεων στην οντοκεντρική βάση

IF ≺*ονομα πεδίου*≻ = *τιμή*

Αυτό σημαίνει ότι για εφαρμοστεί η απεικόνιση θα εξεταστεί αν η συνθήκη είναι αληθής. Ο έλεγχος της *συνθήκης* γίνεται από το *Μηχανισμός Μετάφρασης και Μεταφοράς Δεδομένων* (ενότητα 3.3.5).

4.3 Ορισμός τελεστών απεικόνισης

Για τον καθορισμό απεικόνισης από το σχεσιακό στο οντοκεντρικό σχήμα, ορίζεται από την παρούσα εργασία ένα σύνολο τελεστών απεικόνισης. Οπως θα δούμε, στη συνέχεια, οι τελεστές αυτοί μας βοηθούν να ορίσουμε μια απεικόνιση μεταξύ στοιχείων του σχεσιακού και στοιχείων του οντοκεντρικού σχήματος. Η απεικόνιση αυτή δεν είναι αμφιμονοσήμαντη διότι δεν μπορούμε να ισχυριστούμε ότι στην γενική περίπτωση θα απεικονίζονται τα στοιχεία του ενός σχήματος, στα στοιχεία του άλλου σχήματος ικανοποιώντας τις ιδιότητες 1-1 και επί.

Για την αποθήκευση των απεικονίσεων που ορίζονται στην οντοκεντρική βάση δημιουργήσαμε το μοντέλο που φαίνεται στο Σχήμα 4.2. Οπως φαίνεται στο σχήμα, το μοντέλο έχει περιγραφεί ολόκληρο σε S_Class επίπεδο.

Στο μοντέλο αυτό η κλάση *Transformations* περιγράφει τις απεικονίσεις που ορίζονται με τους τελεστές. Όλα τα γνωρίσματα της κλάσης *Transformations* παίρνουν τιμές στην κλάση Telos_String και είναι τα παρακάτω :

- *kind*
  δηλώνει το είδος του τελεστή και παίρνει τιμές (όπως θα δούμε αργότερα) BECOMES, ISA, BECOMES_LINK και BECOMES_DEPENDENT_OBJECT.

- *fromValue*

δείχνει στις δομές του σχεσιακού σχήματος από όπου θα πάρουμε τα δεδομένα προς μεταφορά.

- *toValue*
  δείχνει το πού θα μεταφερθούν τα δεδομένα στην οντοκεντρική βάση.
- *fromPrefix, fromPostfix, toPrefix, toPostfix*
  δείχνουν στα προθήματα ή επιθήματα που έχουν οριστεί.
- *linkName*
  δείχνει στο όνομα του συνδέσμου που ορίζεται στην περίπτωση του τελεστή τύπου BECOMES_LINK.
- *path*
  δείχνει στα μονοπάτια που ορίζονται στην περίπτωση του τελεστή τύπου BECOMES_DEPENDENT_OBJECT.
- *separator*
  δείχνει στο διαχωριστικό που έχει οριστεί στην περίπτωση του τελεστή τύπου BECOMES_DEPENDENT_OBJECT.
- *conditionField*
  δείχνει στο πεδίο που έχει οριστεί σαν συνθήκη.
- *conditionValue*
  δείχνει στην τιμή που πρέπει να έχει το πεδίο για να ικανοποιείται η συνθήκη.

Οι τελεστές απεικόνισης, που περιγράφονται αμέσως μετά, είναι τέτοιοι ώστε ο καθένας από αυτούς να εκφράζει μια και μόνο απλή πράξη. Επίσης κάθε τελεστής μπορεί να εφαρμοστεί κάθε φορά σε ένα και μόνο πίνακα. Ετσι οι απεικονίσεις που δημιουργούνται αφορούν και αυτές έναν πίνακα κάθε φορά. Ουσιαστικά η λειτουργία κάθε τελεστή είναι να ορίσει ένα μονοπάτι, το οποίο θα πρέπει να ακολουθήσουν τα δεδομένα για να εισαχθούν στο κατάλληλο σημείο στην οντοκεντρική βάση. Τα μονοπάτια αυτά εκφράζουν διαφόρων ειδών σχέσεις ανάμεσα στις δομές του σχήματος-πηγής που μεταφράζονται στις αντίστοιχες σχέσεις ανάμεσα στις δομές του σχήματος-στόχου.

Οι τελεστές αναφέρονται μαζί με τις παραμέτρους εισόδου/εξόδου τους, την λειτουργία που επιτελούν και άλλες πληροφορίες που αφορούν την εφαρμογή τους και τον τρόπο καθορισμού τους από τον χρήστη του συστήματος. Στον καθορισμό όλων των τελεστών, που θα αναφερθούν στη συνέχεια, είναι κοινή η παρακάτω πληροφορία :

- FROM_PREFIX, FROM_POSTFIX
  το πρόθημα και το επίθημα που μπορεί να ορίσει ο χρήστης για να προστεθεί στο όνομα των δεδομένων που προέρχονται από τις παραμέτρους $\Pi_1, ..., \Pi_v$ κατά την εισαγωγή τους στην οντοκεντρική βάση.

- πρόθημα1,επίθημα1
  οι τιμές που παίρνουν αντίστοιχα τα FROM_PREFIX και FROM_POSTFIX και μπορούν να αποτελούνται από αγγλικούς, ελληνικούς χαρακτήρες και τον χαρακτήρα υπογράμμιση.

- TO_PREFIX,TO_POSTFIX
  το πρόθημα και το επίθημα που μπορεί να χρησιμοποιηθεί μόνο στην περίπτωση που η απεικόνιση ορίζεται με την μορφή (α) του τελεστή *BECOMES* για να καθοριστεί το όνομα με το οποίο έχουν εισαχθεί στην οντοκεντρική βάση τα δεδομένα των παραμέτρων $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$.

- πρόθημα2,επίθημα2
  οι τιμές που παίρνουν αντίστοιχα τα TO_PREFIX και TO_POSTFIX και μπορούν να αποτελούνται από αγγλικούς, ελληνικούς χαρακτήρες και τον χαρακτήρα υπογράμμιση.

- IF ≺όνομα πεδίου≻ = τιμή
  καθορίζει την συνθήκη, αν θέλει να ορίσει μία ο χρήστης.

4.3.1 Τελεστής BECOMES

Με τον τελεστή αυτό μεταφέρονται ανεξάρτητες οντότητες του σχήματος-πηγής σε ανεξάρτητες οντότητες στο σχήμα-στόχο. Ο τελεστής *BECOMES* μπορεί να οριστεί με δύο μορφές.

(α) $\Pi_1, ..., \Pi_\nu$ *BECOMES* $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ ή
(β) $\Pi_1, ..., \Pi_\nu$ *BECOMES* K

Οταν ορίζεται με τη μορφή (α) δέχεται ως *είσοδο* τις παραμέτρους $\Pi_1, ..., \Pi_\nu$ και $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$. Ενω όταν ορίζεται με τη μορφή (β) δέχεται ως *είσοδο* τις παραμέτρους $\Pi_1, ..., \Pi_\nu$ και την παράμετρο K. Οι παράμετροι $\Pi_1, ..., \Pi_\nu$,$\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ πρέπει να είναι πεδία πίνακα του σχεσιακού σχήματος. Ενώ η παράμετρος K πρέπει να είναι κλάση του οντοκεντρικού σχήματος.

Η *έξοδος* που παράγεται από αυτήν την απεικόνιση είναι σχέσεις τύπου ‘‘περίπτωσης’’ από τις οντότητες που δημιουργούνται από τα δεδομένα των πεδίων $\Pi_1, ..., \Pi_\nu$ προς την κλάση που αντιστοιχούν οι παράμετροι $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ ή την κλάση K.

Περιορισμοί

- Οι παράμετροι $\Pi_1, ..., \Pi_\nu$ πρέπει να ανήκουν στον ίδιο πίνακα του σχεσιακού σχήματος και να αποτελούν κλειδί του. Αυτό γιατί η ταυτότητα των αντικειμένων που εισάγονται στην οντοκεντρική βάση πρέπει να είναι μοναδική όπως είναι μοναδικός για κάθε γραμμή του πίνακα ο συνδυασμός των πεδίων που αποτελούν το κλειδί του. Πρέπει λοιπόν τα $\Pi_1, ..., \Pi_\nu$ να καθορίζουν μοναδικά την οντότητα που θα δημιουργηθεί.

- Αν η απεικόνιση οριστεί με την μορφή (α) του τελεστή BECOMES πρέπει οι παράμετροι $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ να αποτελούν κλειδί του πίνακα αυτού ή κάποιου άλλου πίνακα για τον λόγο που προαναφέρθηκε.
- Οι παράμετροι $\Pi_1, ..., \Pi_\nu$ και $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ πρέπει να ανήκουν στον ίδιο πίνακα.
- Ο τελεστής *BECOMES* για να μπορεί να εφαρμοστεί χρειάζεται τουλάχιστον μια παράμετρο από τις $\Pi_1, ..., \Pi_\nu$ και μια παράμετρο από τις $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$, αν οριστεί με τη μορφή (α). Ενω αν οριστεί με τη μορφή (β) χρειάζεται τουλάχιστον μια παράμετρο από τις $\Pi_1, ..., \Pi_\nu$ και την παράμετρο K.
- Αν η απεικόνιση οριστεί με την μορφή (α) του τελεστή *BECOMES* πρέπει τα δεδομένα που αντιστοιχούν στις παραμέτρους $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ να έχουν εισαχθεί στην οντοκεντρική βάση πριν ορισθούν οι σχέσεις τύπου ''περίπτωσης'' μεταξύ αυτών και των δεδομένων των παραμέτρων$\Pi_1, ..., \Pi_\nu$.
- Αν η απεικόνιση οριστεί με την μορφή (β) του τελεστή *BECOMES* το όνομα της κλάσης K δεν πρέπει να αλλαχθεί διότι δεν θα μπορεί να εφαρμοστεί η απεικόνιση που ορίστηκε.

Ο καθορισμός της απεικόνισης που ορίζεται με τον τελεστή *BECOMES* γίνεται με το παρακάτω συντακτικό :

FOR TABLE ≺*όνομα πίνακα*≻
$\Pi_1, ..., \Pi_\nu$ *BECOMES* $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ *ή K*
FROM_PREFIX : πρόθημα1
FROM_POSTFIX : επίθημα1
TO_PREFIX : πρόθημα2
TO_POSTFIX : επίθημα2
IF ≺*όνομα πεδίου*≻ *= τιμή*

Οπου, εκτός από τα πεδία που αναφέρθηκαν στην εισαγωγή της ενότητας 4.3, το ≺όνομα πίνακα≻ είναι ο πίνακας όπου ανήκουν τα πεδία $\Pi_1, ..., \Pi_\nu, \Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$.

Για να δούμε ένα παράδειγμα χρήσης του τελεστή *BECOMES* έστω ότι εξετάζουμε μια σχεσιακή βάση που περιέχει τον πίνακα *ζώα* που περιγράφει διάφορα ζώα και ορίζεται σαν :

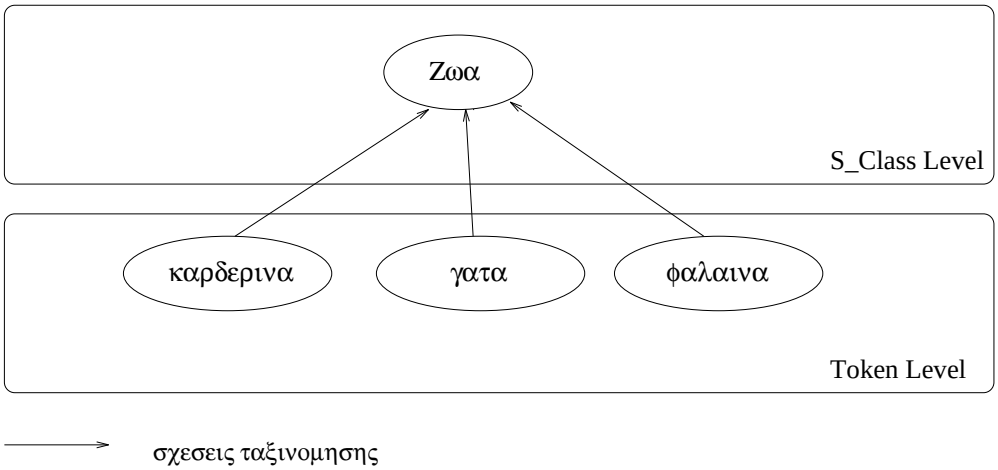
ζώα

όνομαΖώου	είδοςΖώου

όπου *όνομαΖώου* το όνομα του ζώου και *είδοςΖώου* το είδος του ζώου το οποίο μπορεί να είναι πουλί, θηλαστικό ή ψάρι. Εστω τώρα ότι τα δεδομένα που περιέχει ο πίνακας είναι :

ζώα

όνομαΖώου	είδοςΖώου
καρδερίνα	πουλί
γάτα	θηλαστικό
φάλαινα	ψάρι



Σχήμα 4.3: Παράδειγμα (1) χρήσης τελεστή BECOMES

Τέλος έστω ότι στην οντοκεντρική βάση έχει οριστεί η κλάση *Ζώα* και θέλουμε τα δεδομένα του πίνακα να μεταφερθούν σαν περιπτώσεις της κλάσης αυτής. Η απεικόνιση που πρέπει να οριστεί είναι :

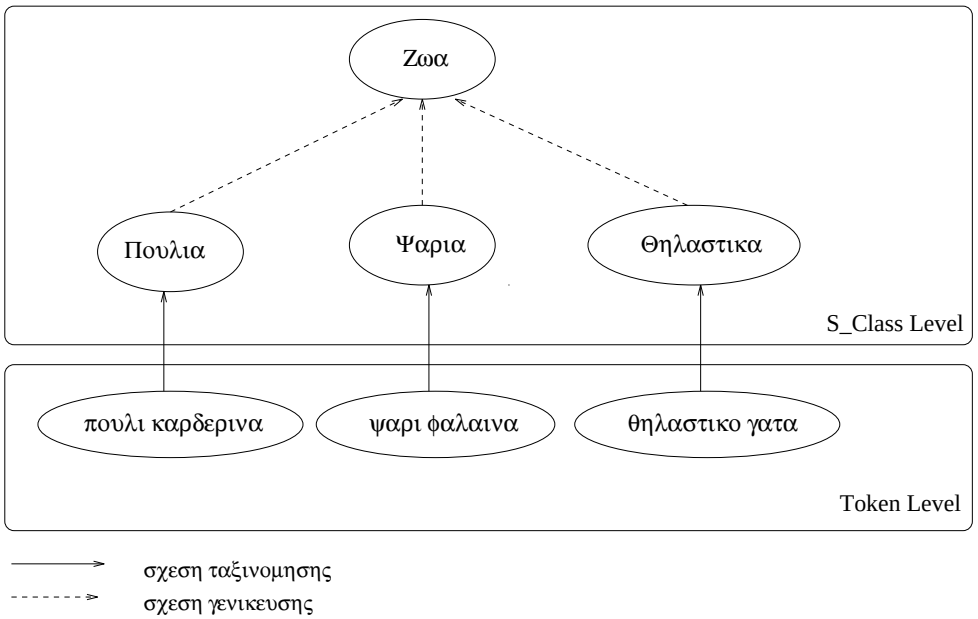
FOR TABLE ζώα
όνομαΖώου BECOMES Ζώα

Η απεικόνιση αυτή θα έχει σαν αποτέλεσμα τα δεδομένα ‘‘καρδερίνα’’,‘‘γάτα’’ και ‘‘φάλαινα’’ να μεταφερθούν σαν περιπτώσεις της κλάσης *Ζώα* όπως φαίνεται στο Σχήμα 4.3.

Εστω τώρα ότι έχουμε ορίσει στην οντοκεντρική βάση και τις κλάσεις *Ψάρια*, *Πουλιά* και *Θηλαστικά* σαν υποκλάσεις της κλάσης *Ζώα* και θέλουμε τα δεδομένα του πίνακα *ζώα* να μεταφερθούν κάτω από την κλάση που ανήκει το κάθε ζώο. Ταυτόχρονα θέλουμε να εισάγεται μπροστά από το όνομα του κάθε ζώου και ένα πρόθημα που θα δηλώνει το είδος του. Για να πετύχουμε το πρώτο από αυτά που θέλουμε πρέπει να χρησιμοποιήσουμε συνθήκη που θα βασίζεται στις τιμές του πεδίου *είδοςΖώου*. Και για να πετύχουμε το δεύτερο πρέπει να ορίσουμε το κατάλληλο πρόθημα που θα εισάγεται σε κάθε περίπτωση μπροστά από το όνομα του ζώου. Η απεικόνιση που πρέπει να οριστεί είναι :

FOR TABLE ζώα
όνομαΖώου BECOMES Ψάρια
FROM_PREFIX : ψάρι
IF είδοςΖώου = ψάρι

FOR TABLE ζώα
όνομαΖώου BECOMES Πουλιά
FROM_PREFIX : πουλί
IF είδοςΖώου = πουλί



Σχήμα 4.4: Παράδειγμα (2) χρήσης τελεστή BECOMES

FOR TABLE ζώα
όνομαΖώου BECOMES Θηλαστικά
FROM_PREFIX : θηλαστικό
IF είδοςΖώου = θηλαστικό

Αυτό θα είχε σαν αποτέλεσμα να εισαχθεί σαν περίπτωση της κλάσης *Ψάρια* το ‘‘ψάρι‘φάλαινα’’, σαν περίπτωση της κλάσης *Θηλαστικά* το ‘‘θηλαστικό‘γάτα’’ και σαν περίπτωση της κλάσης *Πουλιά* το ‘‘πουλί‘καρδερίνα’’ όπως φαίνεται στο Σχήμα 4.4.

4.3.2 Τελεστής ISA

Με τον τελεστή αυτό μεταφέρονται σχέσεις υποσυνόλου, που ισχύουν ανάμεσα σε ανεξάρτητες οντότητες του σχήματος-πηγής, στις αντίστοιχες σχέσεις υποσυνόλου στο σχήμα-στόχος. Ο τελεστής *ISA* μπορεί να οριστεί με δύο μορφές.

(α) $\Pi_1, ..., \Pi_\nu$ *ISA* $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ ή
(β) $\Pi_1, ..., \Pi_\nu$ *ISA* K

Οταν ορίζεται με τη μορφή (α) δέχεται ως *είσοδο* τις παραμέτρους $\Pi_1, ..., \Pi_\nu$ και $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$. Ενω όταν ορίζεται με τη μορφή (β) δέχεται ως *είσοδο* τις παραμέτρους $\Pi_1, ..., \Pi_\nu$ και την παράμετρο K. Οι παράμετροι $\Pi_1, ..., \Pi_\nu$,$\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ πρέπει να είναι πεδία πίνακα του σχεσιακού σχήματος. Ενώ η παράμετρος K πρέπει να είναι κλάση του οντοκεντρικού σχήματος.

Η *έξοδος* που παράγεται από αυτήν την απεικόνιση είναι σχέσεις τύπου γενίκευσης/εξειδίκευσης (isA) από τις οντότητες που δημιουργούνται από τα δεδομένα των πεδίων $\Pi_1, ..., \Pi_\nu$ προς την κλάση που αντιστοιχούν οι παράμετροι $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ ή την κλάση K.

Περιορισμοί

- Οι παράμετροι $\Pi_1, ..., \Pi_\nu$ πρέπει να ανήκουν στον ίδιο πίνακα του σχεσιακού σχήματος και να αποτελούν κλειδί του πίνακα αυτού ή κάποιου άλλου πίνακα. Αυτό γιατί η ταυτότητα των αντικειμένων που θέλουμε να εντοπίσουμε στην οντοκεντρική βάση πρέπει να είναι μοναδική όπως είναι μοναδικός για κάθε γραμμή του πίνακα ο συνδυασμός των πεδίων που αποτελούν το κλειδί του. Πρέπει λοιπόν τα $\Pi_1, ..., \Pi_\nu$ να καθορίζουν μοναδικά την οντότητα που θα γίνει υποκλάση μιας άλλης οντότητας.

- Αν η απεικόνιση οριστεί με την μορφή (α) του τελεστή *ISA* πρέπει οι παράμετροι $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ να αποτελούν κλειδί του πίνακα αυτού ή κάποιου άλλου πίνακα για τον λόγο που προαναφέρθηκε.

- Οι παράμετροι $\Pi_1, ..., \Pi_\nu$ και $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ πρέπει να ανήκουν στον ίδιο πίνακα.

- Ο τελεστής *ISA* για να μπορεί να εφαρμοστεί χρειάζεται τουλάχιστον μια παράμετρο από τις $\Pi_1, ..., \Pi_\nu$ και μια παράμετρο από τις $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$, αν οριστεί με τη μορφή (α). Ενω αν οριστεί με τη μορφή (β) χρειάζεται τουλάχιστον μια παράμετρο από τις $\Pi_1, ..., \Pi_\nu$ και την παράμετρο K.

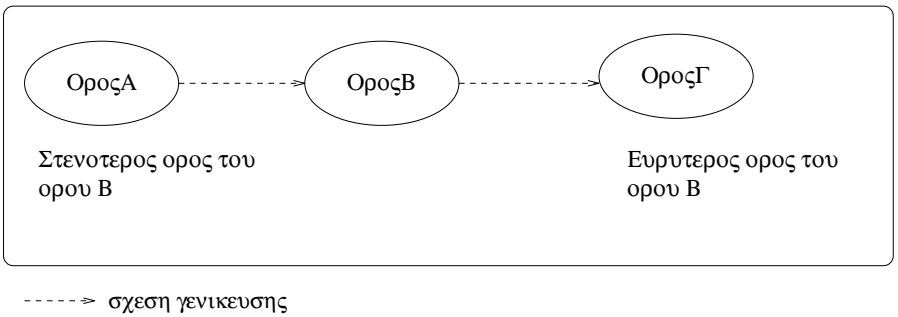
- Αν η απεικόνιση οριστεί με την μορφή (α) του τελεστή *ISA* πρέπει τα δεδομένα που αντιστοιχούν στις παραμέτρους $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ να έχουν εισαχθεί στην οντοκεντρική βάση πριν ορισθούν σχέσεις *ISA* μεταξύ αυτών και των δεδομένων των παραμέτρων $\Pi_1, ..., \Pi_\nu$.

- Αν η απεικόνιση οριστεί με την μορφή (α) του τελεστή *ISA* πρέπει οι οντότητες που έχουν δημιουργηθεί από τα δεδομένα των παραμέτρων $\Pi_1, ..., \Pi_\nu$ και $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ ή K στο οντοκεντρικό σύστημα να είναι στο ίδιο επίπεδο ταξινόμησης βλ.[43],[5] για να μπορεί να ορισθεί σχέση κλάσης-υποκλάσης ανάμεσα τους.

- Αν η απεικόνιση οριστεί με την μορφή (β) του τελεστή *ISA* το όνομα της κλάσης K δεν πρέπει να αλλαχθεί διότι δεν θα μπορεί να εφαρμοστεί η απεικόνιση που ορίστηκε.

Ο καθορισμός της απεικόνισης που ορίζεται με τον τελεστή *ISA* γίνεται με το παρακάτω συντακτικό

FOR TABLE ≺*όνομα πίνακα*≻
$\Pi_1, ..., \Pi_\nu$ *ISA* $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ *ή K*
FROM_PREFIX : πρόθημα1
FROM_POSTFIX : επίθημα1
TO_PREFIX : πρόθημα2



Σχήμα 4.5: Σχέση ευρύτερου-στενότερου όρου στην οντοκεντρική βάση

TO_POSTFIX : επίθημα2
IF ≺όνομα πεδίου≻ = τιμή

Οπου, εκτός από τα πεδία που αναφέρθηκαν στην εισαγωγή της ενότητας 4.3, το ≺όνομα πίνακα≻, είναι ο πίνακας όπου ανήκουν τα πεδία $\Pi_1, ..., \Pi_\nu, \Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$.

Για να δούμε ένα παράδειγμα χρήσης του τελεστή *ISA* έστω ότι εξετάζουμε μια σχεσιακή βάση που περιγράφει λεξικογραφικό θησαυρό. Η βάση αυτή περιέχει τον πίνακα *ευρύτεροιΟροι* που περιγράφει τους ευρύτερους όρους κάποιου όρου και ορίζεται σαν :

ευρύτεροιΟροι

όρος	ευρύτεροςΟρος

όπου *όρος* κάποιος όρος και *ευρύτεροςΟρος* ένας ευρύτερος όρος του όρου *όρος*. Εστω τώρα ότι τα δεδομένα που περιέχει ο πίνακας είναι :

ευρύτεροιΟροι

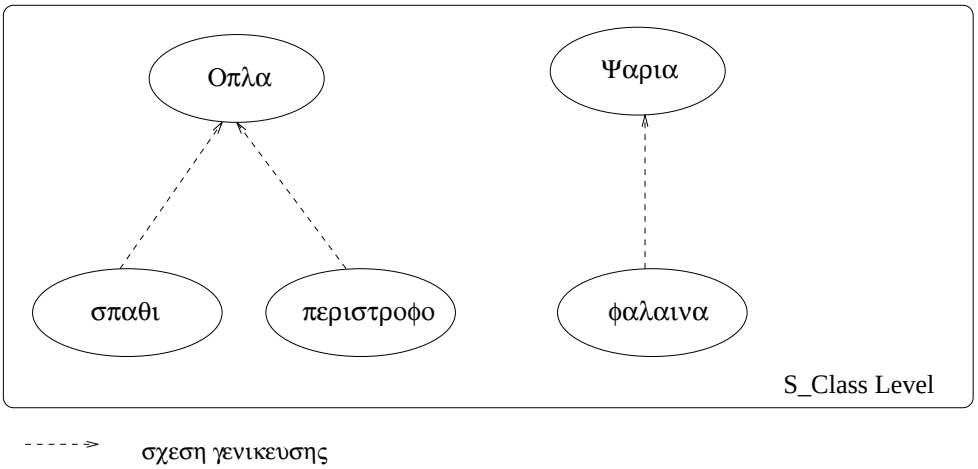
όρος	ευρύτεροςΟρος
σπαθί	Οπλα
περίστροφο	Οπλα
φάλαινα	Ψάρια

Η σχέση που περιγράφει τον ευρύτερο όρο ενός άλλου όρου στο οντοκεντρικό μοντέλο φαίνεται στο Σχήμα 4.5 όπου βλέπουμε ότι αποδίδεται με τη σχέση γενίκευσης/εξειδίκευσης.

Τέλος έστω ότι στην οντοκεντρική βάση έχει εισαχθεί η κλάση *Οπλα* και η κλάση *Ψάρια* και θέλουμε τα δεδομένα του πεδίου *όρος* του πίνακα *ευρύτεροιΟροι* να μεταφερθούν σαν υποκλάσεις των κλάσεων που περιγράφει το πεδίο *ευρύτεροςΟρος* . Η απεικόνιση που πρέπει να οριστεί είναι :

FOR TABLE ευρύτεροιΟροι
όρος ISA ευρύτεροςΟρος

Η απεικόνιση αυτή θα έχει σαν αποτέλεσμα το ‘‘σπαθί’’ και το ‘‘περίστροφο’’ να εισαχθούν σαν υποκλάσεις της κλάσης *Οπλα* ενώ το ‘‘φάλαινα’’ θα εισαχθεί σαν υποκλάση της κλάσης *Ψάρια* όπως φαίνεται στο Σχήμα 4.6.



Σχήμα 4.6: Παράδειγμα χρήσης τελεστή ISA

4.3.3 Τελεστής BECOMES_LINK

Με τον τελεστή αυτό μεταφέρονται σχέσεις οντότητας-γνωρίσματος, που ισχύουν ανάμεσα σε οντότητες του σχήματος-πηγή, στις αντίστοιχες σχέσεις οντότητας-γνωρίσματος ανάμεσα σε οντότητες στο σχήμα-στόχο. Ο τελεστής *BECOMES_LINK* ορίζεται σαν :

FROM : $\Pi_1, ..., \Pi_\nu$
TO : $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$
BECOMES_LINK ΟνομαΣυνδέσμου

Δέχεται ως *είσοδο* τις παραμέτρους $\Pi_1, ..., \Pi_\nu, \Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ που είναι πεδία πίνακα του σχεσιακού σχήματος και την παράμετρο ΟνομαΣυνδέσμου που είναι σύνδεσμος στο οντοκεντρικό σχήμα.

Η *έξοδος* που παράγεται από αυτήν την απεικόνιση είναι σύνδεσμοι τύπου ΟνομαΣυνδέσμου ανάμεσα στις οντότητες που δημιουργήθηκαν από τα δεδομένα των πεδίων $\Pi_1, ..., \Pi_\nu$ και τις οντότητες που δημιουργήθηκαν από τα δεδομένα των πεδίων $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$. Συγκεκριμένα τα δεδομένα των πεδίων $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ γίνονται γνωρίσματα στις οντότητες που δημιουργήθηκαν από τα δεδομένα των πεδίων $\Pi_1, ..., \Pi_\nu$.

Περιορισμοί

- Οι παράμετροι $\Pi_1, ..., \Pi_\nu, \Pi_\nu, \Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ πρέπει να ανήκουν στον ίδιο πίνακα του σχεσιακού σχήματος.
- Ο τελεστής *BECOMES_LINK* για να μπορεί να εφαρμοστεί χρειάζεται τουλάχιστον μια παράμετρο από τις $\Pi_1, ..., \Pi_\nu$, μια παράμετρο από τις $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ και την παράμετρο ΟνομαΣυνδέσμου.
- Τα δεδομένα που αντιστοιχούν στις παραμέτρους $\Pi_1, ..., \Pi_\nu$ και $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$ πρέπει να έχουν εισαχθεί στην οντοκεντρική βάση πριν ορισθούν οι σχέσεις γνωρίσματος μεταξύ τους.

- Το ΟνομαΣυνδέσμου δεν πρέπει να αλλαχθεί διότι τότε δεν θα μπορεί να εφαρμοστεί η απεικόνιση που ορίστηκε.

Ο καθορισμός της απεικόνισης που ορίζεται με τον τελεστή *BECOMES_LINK* γίνεται με το παρακάτω συντακτικό :

FOR TABLE ≺*όνομα πίνακα*≻
FROM : $\Pi_1, ..., \Pi_\nu$
TO : $\Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$
BECOMES_LINK ≺*όνομα συνδέσμου*≻
FROM_PREFIX : πρόθημα1
FROM_POSTFIX : επίθημα1
TO_PREFIX : πρόθημα2
TO_POSTFIX : επίθημα2
IF ≺*όνομα πεδίου*≻ *= τιμή*

Οπου τα παραπάνω, εκτός από τα πεδία που αναφέρθηκαν στην εισαγωγή της ενότητας 4.3, είναι :

- ≺όνομα πίνακα≻
  ο πίνακας όπου ανήκουν τα πεδία $\Pi_1, ..., \Pi_\nu, \Pi_{\nu+1}, ..., \Pi_{\nu+\mu}$.

- FROM,TO
  το FROM καθορίζει από ποιές οντότητες θα ξεκινάει ο σύνδεσμος που θα δημιουργηθεί και το TO σε ποιές οντότητες θα καταλήγει.

- ≺όνομα συνδέσμου≻
  καθορίζει το όνομα του συνδέσμου που θα δημιουργηθεί.

Για να δούμε ένα παράδειγμα χρήσης του τελεστή έστω ότι εξετάζουμε μια σχεσιακή βάση που περιγράφει αυτοκίνητα. Η βάση αυτή περιέχει τον πίνακα *χρώμαΑυτοκινήτου* που περιγράφει το χρώμα των αυτοκινήτων που ανήκουν στη βάση και ορίζεται σαν :

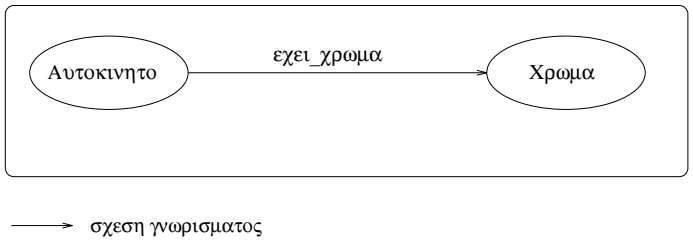
χρώμαΑυτοκινήτου

κωδικόςΑυτοκινήτου	χρώμα

όπου *κωδικόςΑυτοκινήτου* καθορίζει το αυτοκίνητο και *χρώμα* καθορίζει το χρώμα του αυτοκινήτου. Εστω τώρα ότι τα δεδομένα που περιέχει ο πίνακας είναι :

χρώμαΑυτοκινήτου

κωδικόςΑυτοκινήτου	χρώμα
ΗΚ9898	κόκκινο
ΗΚ9090	πράσινο



Σχήμα 4.7: Σχέση ανάμεσα στην κλάση Αυτοκίνητο και την κλάση Χρώμα στην οντοκεντρική βάση

Τέλος έστω ότι στην οντοκεντρική βάση έχει εισαχθεί η κλάση *Αυτοκίνητο* και η κλάση *Χρώμα* και έχει οριστεί ανάμεσα τους ο σύνδεσμος *έχει_χρώμα* όπως φαίνεται στο Σχήμα 4.7.

Θέλουμε να εισαχθούν στην οντοκεντρική βάση οι σχέσεις τύπου *έχει_χρώμα* ανάμεσα στα δεδομένα του πεδίου *κωδικόςΑυτοκινήτου* και του πεδίου *χρώμα* που περιγράφει ο πίνακας *χρώμαΑυτοκινήτου.* Επίσης γνωρίζουμε ότι τα αυτοκίνητα έχουν εισαχθεί με το πρόθημα ‘‘αυτοκίνητο’’ και τα χρώματα με το πρόθημα ‘‘χρώμα’’, και αυτό πρέπει να το λάβουμε υπόψη μας στον καθορισμό της απεικόνισης. Η απεικόνιση που πρέπει να οριστεί είναι :

FOR TABLE χρώμαΑυτοκινήτου
FROM : κωδικόςΑυτοκινήτου
TO : χρώμα
BECOMES_LINK έχει_χρώμα
FROM_PREFIX : αυτοκίνητο
TO_PREFIX : χρώμα

Η απεικόνιση αυτή θα έχει σαν αποτέλεσμα να δημιουργηθεί ένας σύνδεσμος με όνομα *έχει_χρώμα* ανάμεσα στο ‘‘αυτοκίνητο‘ΗΚ9898’’ και το ‘‘χρώμα‘κόκκινο’’ και ένας ακόμα σύνδεσμος ανάμεσα στο ‘‘αυτοκίνητο‘ΗΚ9090’’ και το ‘‘χρώμα‘πράσινο’’ όπως φαίνεται στο Σχήμα 4.8.

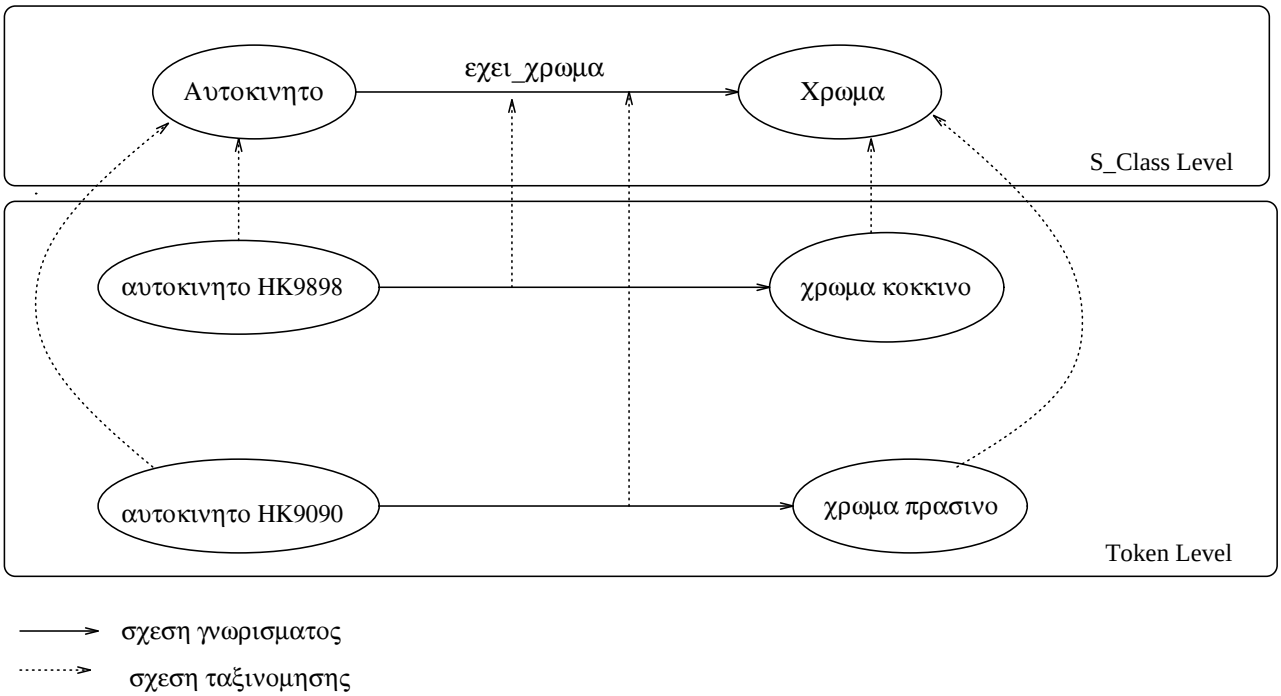
4.3.4 Τελεστής BECOMES_DEPENDENT_OBJECT

Με τον τελεστή αυτό μεταφέρονται σύνθετες οντότητες του σχήματος-πηγής σε μονοπάτια από ανεξάρτητες οντότητες και συνδέσμους στο σχήμα-στόχος.
Ο τελεστής *BECOMES_DEPENDENT_OBJECT* ορίζεται σαν :
Π_1 *BECOMES_DEPENDENT_OBJECT* $M_1, .., M_\mu$

Δέχεται ως *είσοδο* την παράμετρο Π_1, που είναι πεδίο πίνακα του σχεσιακού σχήματος και τις παραμέτρους $M_1, ..., M_\mu$ που είναι μονοπάτια στην οντοκεντρική βάση.

Η μορφή των μονοπατιών αυτών M_ι, όπου ι=1,...,μ είναι

Αντικείμενο1.Σύνδεσμος1- Αντικείμενο2.Σύνδεσμος2- Αντικείμενο3...Σύνδεσμος(N-1)- ΑντικείμενοN .



Σχήμα 4.8: Παράδειγμα χρήσης τελεστή BECOMES_LINK

Το κάθε ένα από τα $M_ι$ καθορίζει το πλήρες μονοπάτι, που πρέπει να ακολουθηθεί, για να μεταβούμε από την κλάση Αντικείμενο1, της οντοκεντρικής βάσης, στην κλάση ΑντικείμενοΝ.

Η *έξοδος* που παράγεται από αυτήν την απεικόνιση είναι οντότητες και σύνδεσμοι έτσι ώστε να σχηματισθούν στην οντοκεντρική βάση οι περιπτώσεις των $M_ι$, με ι=1,...,μ.

Ο λόγος που χρειάζεται, μια τέτοια απεικόνιση, είναι ότι πολλές φορές παρουσιάζεται η ανάγκη, ένα πεδίο μιας σχέσης να μετασχηματίζεται σε περισσότερα από ένα αντικείμενα και συνδέσμους στην οντοκεντρική βάση, λόγω του υπάρχοντος οντοκεντρικού σχήματος. Ο τελεστής αυτός έχει σαν σκοπό του να παράγει όλες τις οντότητες και τους συνδέσμους που χρειάζονται για να δημιουργηθούν οι περιπτώσεις του μονοπατιού $M_ι$. Ο τελεστής αυτός αποτελεί ουσιαστικά μια σύνθεση των τελεστών *BECOMES* και *BECOMES_LINK* με μερικές ακόμα προσθήκες που αναφέρονται παρακάτω.

Σε ορισμένες περιπτώσεις τυχαίνει το πεδίο που θέλουμε να μεταφερθεί να είναι σύνθετο. Να αποτελείται δηλαδή από πολλά τμήματα, καθένα από αυτά τα οποία πρέπει να μετασχηματιστεί από μόνο του σε ένα ολόκληρο μονοπάτι. Παράδειγμα είναι οι ημερομηνίες του τύπου ΗΗ-ΜΜ-ΧΧ ή ΗΗ/ΜΜ/ΧΧ ή ΗΗ.ΜΜ.ΧΧ κ.τ.λ., όπου

το ΗΗ πρέπει να μετασχηματιστει σε Ημερομηνία.μέρα- Μέρα,

το ΜΜ πρέπει να μετασχηματιστει σε Ημερομηνία.μήνας- Μήνας και

το ΧΧ πρέπει να μετασχηματιστει σε Ημερομηνία.έτος- Ετος.

Για να αντιμετωπισθούν λοιπόν οι ανάγκες τέτοιου είδους πεδίων δίνεται η δυνατότητα να καθοριστούν περισσότερα από ένα μονοπάτια, ένα για κάθε τμήμα του πεδίου. Το πως χωρίζονται τα τμήματα αυτά μεταξύ τους το καθορίζει ο χρήστης. Γενικά επιτρέπονται δύο είδη διαχωριστικών που είναι τα εξής : ≺ / ≻ και ≺ - ≻. Τέλος, για κάθε τμήμα του πεδίου πρέπει να οριστεί και ένα μονοπάτι.

Περιορισμοί

- Εαν το πεδίο που θέλουμε να μεταφερθεί αποτελείται από περισσότερα από ένα τμήματα, πρέπει να οριστεί από ένα μονοπάτι για κάθε τμήμα.
- Οπως είπαμε το κάθε μονοπάτι έχει την παρακάτω μορφή :

  Αντικείμενο1.Σύνδεσμος- Αντικείμενο2.Σύνδεσμος...

  Απαιτείται το πρώτο αντικείμενο των μονοπατιών που ορίζονται σε μια απεικόνιση να είναι το ίδιο για όλα. Δηλαδή όλα τα μονοπάτια να έχουν κοινή αρχή.

Ο καθορισμός της απεικόνισης που ορίζεται με τον τελεστή *BECOMES_DEPENDENT_OBJECT* γίνεται με το παρακάτω συντακτικό :

FOR TABLE ≺*όνομα πίνακα*≻
Π_1 *BECOMES_DEPENDENT_OBJECT*
PATH : $M_1, ..., M_\mu$
FROM_PREFIX : πρόθημα1
FROM_POSTFIX : επίθημα1
IF ≺*όνομα πεδίου*≻ *= τιμή*
SEPARATOR = ≺ */ ή -* ≻

Οπου, εκτός από τα πεδία που αναφέρθηκαν στην εισαγωγή της ενότητας 4.3, το ≺όνομα πίνακα≻, είναι ο πίνακας όπου ανήκει το πεδίο Π_1.

Για να δούμε ένα παράδειγμα χρήσης του τελεστή *BECOMES_DEPENDENT_OBJECT* έστω ότι εξετάζουμε μια σχεσιακή βάση που περιέχει τον πίνακα *γενέθλια*. Ο πίνακας αυτός περιγράφει τα γενέθλια των ανθρώπων της βάσης και ορίζεται σαν :

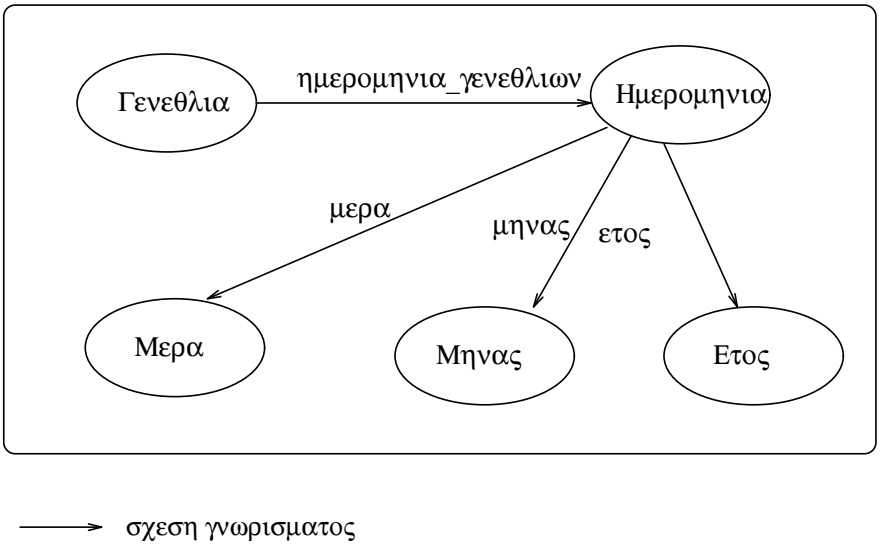
γενέθλια

ημερομηνία

όπου *ημερομηνία*, ημερομηνία γενεθλίων κάποιου ανθρώπου. Εστω τώρα ότι τα δεδομένα που περιέχει ο πίνακας είναι :

γενέθλια

ημερομηνία
12/3/1990



Σχήμα 4.9: Γενέθλια

Τέλος έστω ότι τα γενέθλια περιγράφονται στο οντοκεντρικό μοντέλο όπως φαίνεται στο Σχήμα 4.9.

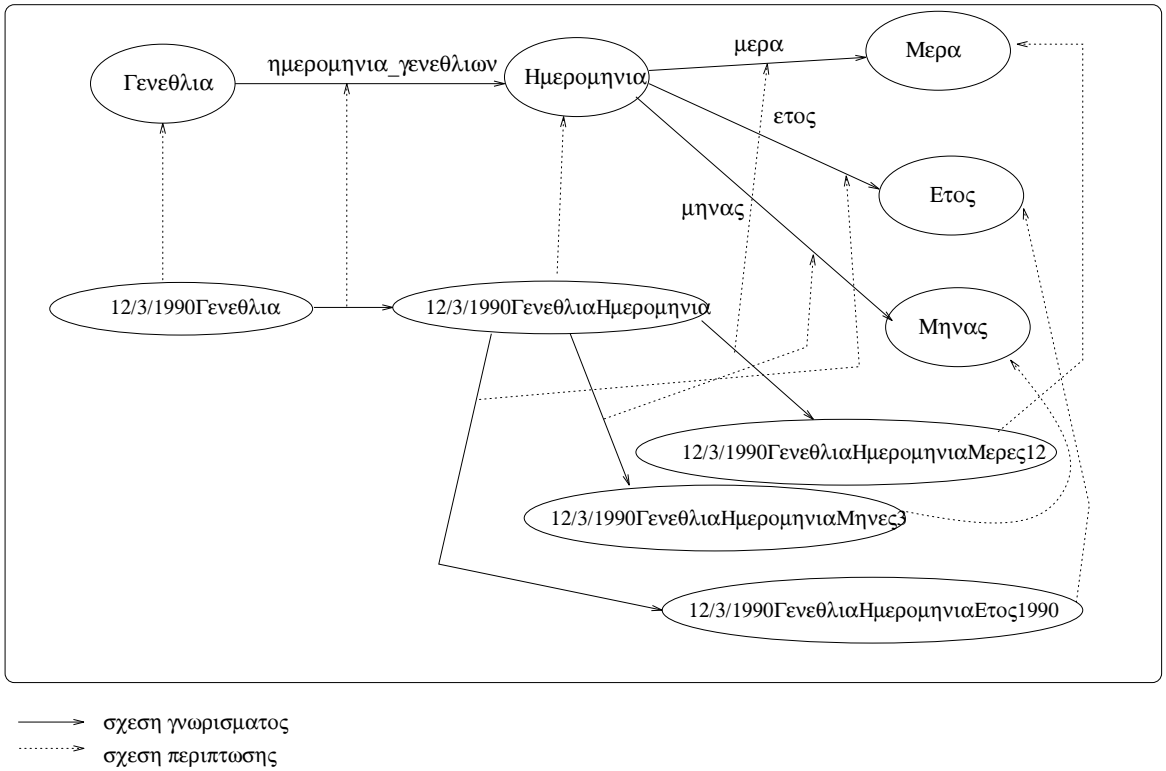
Θέλουμε να εισαγάγουμε στην οντοκεντρική βάση τις ημερομηνίες γενεθλίων που περιγράφονται στον πίνακα *γενέθλια* για να τις συνδέσουμε αργότερα με τους αντίστοιχους ανθρώπους. Η απεικόνιση που πρέπει να οριστεί είναι :

FOR TABLE γενέθλια
ημερομηνία BECOMES_DEPENDENT_OBJECT
PATH : Γενέθλια.ημερομηνία_γενεθλίων-Ημερομηνία.μέρα-Μέρα,
Γενέθλια.ημερομηνία_γενεθλίων-Ημερομηνία.μήνας-Μήνας,
Γενέθλια.ημερομηνία_γενεθλίων-Ημερομηνία.ετος-Ετος
*SEPARATOR = *

Η απεικόνιση αυτή θα έχει σαν αποτέλεσμα να δημιουργηθούν οι οντότητες ‘‘12/3/1990‘Γενέθλια’’, ‘‘12/3/1990‘Γενέθλια‘Ημερομηνία’’,‘‘12/3/1990‘Γενέθλια‘Ημερομηνία‘Μέρα‘12’’, ‘‘12/3/1990‘Γενέθλια‘Ημερομηνία‘Μήνας‘3’’ και ‘‘12/3/1990‘Γενέθλια‘Ημερομηνία‘Ετος‘1990’’ όπως επίσης και οι κατάλληλοι σύνδεσμοι μεταξύ τους όπως φαίνεται στο Σχήμα 4.10.

4.4 Ορισμός απεικονίσεων μέσω ξένων κλειδιών

Στις σχεσιακές βάσεις δεδομένων είναι διαδεδομένη η έννοια των *ξένων κλειδιών*. Ξένο κλειδί ενός πίνακα λέγεται ένα πεδίο του ή ο συνδυασμός ορισμένων πεδίων του που αποτελούν για έναν άλλο πίνακα κύριο κλειδί. Η χρήση ξένων κλειδιών γίνεται συνήθως σε πίνακες που εκφράζουν σχέσεις



Σχήμα 4.10: Παράδειγμα χρήσης τελεστή BECOMES_DEPENDENT_OBJECT

ανάμεσα στα δεδομένα άλλων πινάκων. Ετσι για παράδειγμα αν θεωρήσουμε τους πίνακες

ζώα

ονομαΖώου	κωδικόςΖώου

είδη

είδος	κωδικόςΕίδους

ζώα_είδη

κωδικόςΖώου	κωδικόςΕίδους

Ο πίνακας *ζώα* περιγράφει ζώα και έχει πεδία το *ονομαΖώου*, που είναι το όνομα του ζώου και *κωδικόςΖώου*, που είναι μια ταυτότητα που καθορίζει μοναδικά το ζώο και αποτελεί κλειδί του πίνακα. Ο πίνακας *είδη* περιγράφει είδη ζώων και έχει πεδία το *είδος*, που είναι το όνομα του είδους και *κωδικόςΕίδους*, που είναι μια ταυτότητα που καθορίζει μοναδικά το είδος και αποτελεί κλειδί του πίνακα. Τέλος ο πίνακας *ζώα_είδη* συσχετίζει τα ζώα με τα είδη τους και έχει πεδία τα *κωδικόςΕίδους* και *κωδικόςΖώου*. Τα πεδία *κωδικόςΕίδους* και *κωδικόςΖώου* αποτελούν κύριο κλειδί του *ζώα_είδη* αλλά το κάθε ένα ξεχωριστά αποτελεί ξένο κλειδί.

Αυτό που θέλουμε είναι να μεταφερθούν στην οντοκεντρική βάση οι σχέσεις των ζώων με τα είδη τους που περιγράφονται από τον πίνακα *ζώα_είδη*. Για να γίνει αυτό ορίζουμε την απεικόνιση

FOR TABLE ζώα_είδη
FROM : κωδικόςΖώου
TO : κωδικόςΕίδους
BECOMES_LINK έχει_είδος

υποθέτοντας ότι τα ζώα συνδέονται με τα είδη τους στην οντοκεντρική βάση με συνδέσμους τύπου *έχει_είδος*.

Αυτό που μεταφέρεται στην οντοκεντρική βάση στην συγκεκριμένη περίπτωση δεν είναι οι σχέσεις ανάμεσα στα πεδία *κωδικόςΖώου* και *κωδικόςΕίδους* αλλά οι σχέσεις ανάμεσα στις οντότητες που αντιπροσωπεύουν αυτά τα πεδία δηλαδή σχέσεις ανάμεσα στα δεδομένα του πεδίου *ονομαΖώου* του πίνακα *ζώα* και του πεδίου *είδος* του πίνακα *είδη*. Ετσι στην οντοκεντρική βάση δεν μεταφέρονται σχέσεις ανάμεσα σε αριθμούς, μια και τα *κωδικόςΖώου* και *κωδικόςΕίδους* είναι αριθμοί, αλλά σχέσεις ανάμεσα σε ονόματα ζώων και τα είδη τους. Για γίνει βέβαια αυτή η αντικατάσταση πρέπει να έχει δηλωθεί. Στην συγκεκριμένη περίπτωση πρέπει να έχει δηλωθεί ότι το *κωδικόςΖώου* αντιπροσωπεύει το πεδίο *ονομαΖώου* του πίνακα *ζώα* και το *κωδικόςΕίδους* αντιπροσωπεύει το πεδίο *είδος* του πίνακα *είδη*. Η αντιπροσώπευση αυτή δηλώνεται στη φάση καθορισμού του σχεσιακού σχήματος μέσω των σχέσεων καρτεσιανού γινομένου που ορίζονται ανάμεσα στους πίνακες. Το σύστημα κατά την λειτουργία του αναζητά τέτοιου είδους σχέσεις και τις εφαρμόζει. Ετσι αν οι πίνακες περιέχουν τα παρακάτω δεδομένα

ζώα

ονομαΖώου	κωδικόςΖώου
γάτα	15

είδη

είδος	κωδικόςΕίδους
Θηλαστικά	103
Σαρκοφάγα	175

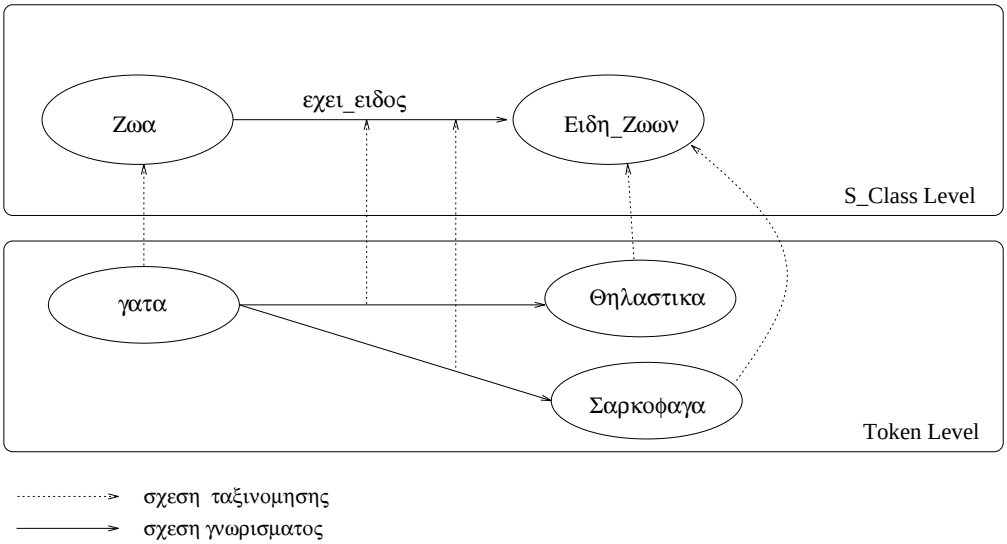
ζώα_είδη

κωδικόςΖώου	κωδικόςΕίδους
15	103
15	175

αυτό που θα μεταφερθεί στην οντοκεντρική βάση είναι σύνδεσμοι ανάμεσα στο ‘‘γάτα’’ και τα ‘‘Θηλαστικά’’,‘‘Σαρκοφάγα’’, όπως φαίνεται στο Σχήμα 4.11.

4.5 Ονοματοδοσία

Στην ενότητα αυτή θα ασχοληθούμε με το πως παίρνουν το όνομα τους τα δεδομένα κατά τη μεταφορά τους στην οντοκεντρική βάση. Γενικά το όνομα αυτό (που αποτελεί το λογικό τους όνομα για το



Σχήμα 4.11: Παράδειγμα μεταφοράς δεδομένων με χρήση ξένου κλειδιού

οντοκεντρικό σύστημα) είναι το ίδιο με το περιεχόμενο τους συμπληρωμένο από το πρόθημα ή το επίθημα (ενότητα 4.2.2) που έχει ορίσει ο χρήστης. Ετσι το τελικό όνομα σχηματίζεται σαν :

Τελικό Ονομα = Πρόθημα + Δεδομένο + Επίθημα

Αυτό ισχύει για όλους τους τελεστές εκτός από τον *BECOMES_DEPENDENT_OBJECT*.

Στην περίπτωση του τελεστή *BECOMES_DEPENDENT_OBJECT* επειδή τα δεδομένα που παράγονται όταν οριστεί τέτοιου είδους απεικόνιση είναι σύνθετα και η μετάφραση γίνεται αυτόματα, τα ονόματα των οντοτήτων που σχηματίζονται είναι και αυτά σύνθετα. Η μέθοδος που ακολουθείται είναι η εξής :

Αντικείμενο1.Σύνδεσμος1- Αντικείμενο2.Σύνδεσμος2- Αντικείμενο3...

Εαν το παραπάνω είναι η δομή του μονοπατιού (ενότητα 4.3.4) τότε το πρώτο αντικείμενο που κατασκευάζεται και γίνεται περίπτωση της κλάσης *Αντικείμενο1* παίρνει όνομα *Δεδομένο1'Αντικείμενο1*. Το επόμενο αντικείμενο που κατασκευάζεται και γίνεται περίπτωση της κλάσης *Αντικείμενο2* παίρνει όνομα το όνομα του προηγούμενου αντικειμένου συν *Δεδομένο2'Αντικείμενο2* δηλαδή τελικά παίρνει όνομα *Δεδομένο1'Αντικείμενο1'Δεδομένο2'Αντικείμενο2*.

Αυτός ο τρόπος ονοματοδοσίας συνεχίζεται για όλα τα αντικείμενα που δημιουργούνται κατά το σχηματισμό του μονοπατιού. Μόνο για το τελευταίο αντικείμενο του μονοπατιού γίνεται έλεγχος αν είναι βασικός τύπος της οντοκεντρικής βάσης. *Εαν είναι* παίρνει σαν τιμή το εκάστοτε τμήμα του δεδομένου π.χ αν είναι το δεδομένο ημερομηνία με μορφή 12-3-95 και το 12 αντιστοιχεί σε μέρες και είναι ακέραιος στην οντοκεντρική βάση, θα πάρει το τελευταίο κομμάτι τιμή 12. *Εαν δεν είναι* βασικός τύπος της οντοκεντρικής βάσης ονομάζεται όπως και τα προηγούμενα αντικείμενα.

4.6 Περιορισμοί

4.6.1 Μορφή σχεσιακού σχήματος

Είναι γνωστό από τη θεωρία των βάσεων δεδομένων τι είναι κανονικοποίηση [64],[34],[20]. Για να μπορούν να εφαρμοστούν με επιτυχία οι τελεστές που ορίστηκαν είναι απαραίτητο το σχεσιακό σχήμα να είναι κανονικοποιημένο σε τουλάχιστον τρίτη κανονική μορφή (3NF). Αυτό σημαίνει ότι σε ένα πίνακα δεν πρέπει η τιμή ενός πεδίου να εξαρτάται από την τιμή ενός άλλου πεδίου. Κάτι τέτοιο θέλουμε να συμβαίνει γιατί οι τελεστές που ορίστηκαν μπορούν να εφαρμόζονται σε ένα και μόνο πίνακα και δεν μπορούν να λάβουν υπόψη τους τις σχέσεις αυτού του πίνακα με άλλους. Ενα παράδειγμα θα κάνει πιο φανερό τι προβλήματα μπορούν να δημιουργηθούν σε περίπτωση που η τιμή ενός πεδίου εξαρτάται από τις τιμές ενός άλλου πεδίου. Το παράδειγμα προέρχεται από την επικοινωνία ΜΙΤΟΣ/CM-ΚΛΕΙΩ.

Εστω ο πίνακας *chronologies* που περιγράφει χρονολογίες και έχει τη μορφή

chronologies	chronid	chronbegin	chronend	chronlikely	chronology	chronkind

Οπου *chronid* το κύριο κλειδί του πίνακα, *chronbegin* και *chronend* χρονολογίες αρχής και τέλους περιόδου αντίστοιχα, *chronlikely* η πιθανότερη χρονολογία σε μια περίοδο, *chronology* το όνομα της περιόδου και *chronkind* το είδος χρονολογίας που μπορεί να είναι περίοδος,ημερομηνία π.Χ ή ημερομηνία μ.Χ.

Οι απεικονίσεις που θέλουμε να ορίσουμε είναι :

- το πεδίο *chronology* να μεταφερθεί στην οντοκεντρική βάση σαν περίοδος (Period), αν το *chronkind* είναι περίοδος, με αρχή και τέλος περιόδου τις χρονολογίες των *chronbegin* και *chronend.*
- το πεδίο *chronology* να μεταφερθεί στην οντοκεντρική βάση σαν ημερομηνία π.Χ (JulianDateBC) αν το *chronkind* είναι ημερομηνία π.Χ ή σαν ημερομηνία μ.Χ (JulianDateAD) αν το *chronkind* είναι ημερομηνία μ.Χ.

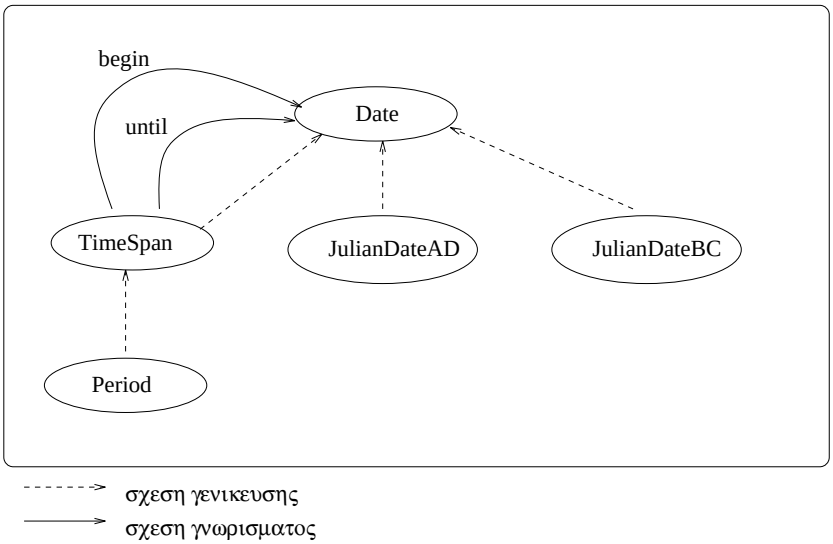
Ο τρόπος με τον οποίο περιγράφονται οι χρονολογίες στο μοντέλο του ΚΛΕΙΩ [3] φαίνεται στο Σχήμα 4.12.

Συνεχίζοντας το παράδειγμα βλέπουμε ότι οι χρονολογίες είναι συνδεδεμένες με αντικείμενα της βάσης μέσω του πίνακα *objchrons* που έχει τη μορφή :

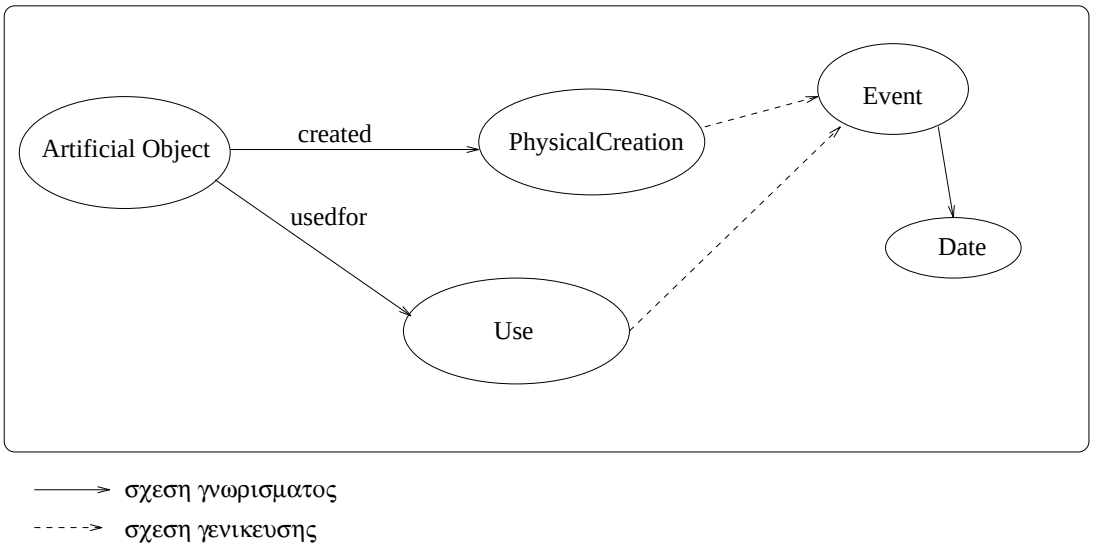
objchrons	chronid	objectid	chronkind

Οπου *chronid* καθορίζει την χρονολογία, *objectid* καθορίζει το αντικείμενο που έχει συνδεθεί με την χρονολογία και *chronkind* καθορίζει το είδος της σχέσης του αντικειμένου με την χρονολογία το οποίο μπορεί να είναι κατασκευή αντικειμένου (creation) ή χρήση αντικειμένου (use). Οι σχέσεις αυτές μεταξύ αντικειμένου και χρονολογίας στο μοντέλο του ΚΛΕΙΩ [3] φαίνονται στο Σχήμα 4.13.

Θέλουμε τώρα να συσχετίσουμε τις χρονολογίες με τα αντικείμενα με βάση τα περιεχόμενα του πίνακα *objchrons* και υποθέτοντας ότι και οι χρονολογίες και τα αντικείμενα έχουν εισαχθεί στην



Σχήμα 4.12: Χρονολογίες στο ΚΛΕΙΩ



Σχήμα 4.13: Σχέσεις κατασκευής και χρήσης μεταξύ αντικειμένων και χρονολογιών στο ΚΛΕΙΩ

κατάλληλη θέση στην οντοκεντρική βάση. Εστω ότι θέλουμε να συσχετίσουμε την χρονολογία κατασκευής ενός αντικειμένου με το αντικείμενο. Εδώ ακριβώς αρχίζει το πρόβλημα. Για να γίνει η συσχέτιση πρέπει να αναφερθούμε στην πίνακα *objchrons* αλλά από αυτόν γνωρίζουμε ότι πρόκειται για την χρονολογία κατασκευής του αντικειμένου και όχι το είδος της χρονολογίας αυτής. Ετσι δεν ξέρουμε αν η χρονολογία αυτή είναι περίοδος ή χρονολογία π.Χ ή χρονολογία μ.Χ. Οπότε δεν ξέρουμε πως να αναφερθούμε σε αυτήν για να την εντοπίσουμε στην οντοκεντρική βάση και να τη συνδέσουμε με το αντικείμενο.

Το πρόβλημα αυτό δεν θα υπήρχε αν το σχήμα της βάσης ήταν κανονικοποιημένο και υπήρχαν τρεις πίνακες περιγραφής χρονολογιών : ένας για χρονολογίες που είναι περίοδοι, ένας για χρονολογίες που είναι π.Χ και ένας για χρονολογίες που είναι μ.Χ. Και επίσης υπήρχαν τρεις πίνακες συσχέτισης αντικειμένων με το αντίστοιχο είδος χρονολογίας.

Παρόλο που οι τελεστές ορίστηκαν έτσι ώστε να μπορούν να αντιμετωπίσουν κακό σχεδιασμό του σχεσιακού σχήματος (ενότητα 4.2.3), η βοήθεια που μπορούν να προσφέρουν σε τέτοιες περιπτώσεις είναι περιορισμένη. Για να είναι λοιπόν σίγουρα επιτυχής η εφαρμογή των τελεστών και να μην υπάρχουν απρόσμενα αποτελέσματα πρέπει πρώτα να κανονικοποιηθεί το σχεσιακό σχήμα σε μορφή τουλάχιστον 3NF.

4.6.2 Εφαρμογή τελεστών σε πίνακες χωρίς κύριο κλειδί

Στις σχεσιακές βάσεις δεδομένων μπορεί να βρεθούν πίνακες που δεν έχουν κλειδί, τέτοιοι πίνακες λέγονται *ασθενείς οντότητες*. Η εφαρμογή των τελεστών σε τέτοιου είδους πίνακες δεν θα έχει σωστά αποτελέσματα, παρά μόνο κατά τύχη. Για να διαπιστώσουμε το γιατί συμβαίνει αυτό ας εξετάσουμε το παράδειγμα που ακολουθεί.

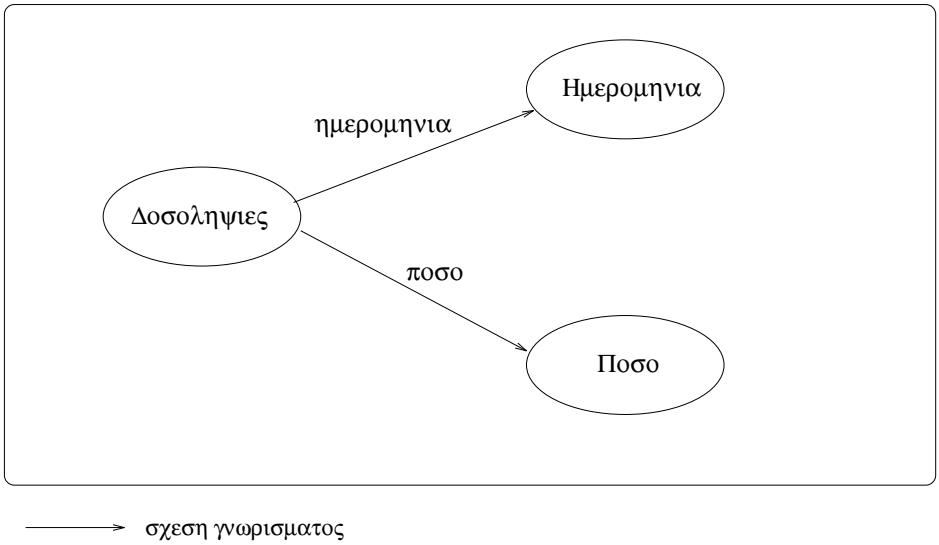
Εστω ο πίνακας *δοσοληψίες* με τρία πεδία *αριθμός-δοσοληψίας*, *ημερομηνία* και *ποσό*. Οπου *αριθμός-δοσοληψίας* είναι αριθμός δοσοληψίας, *ημερομηνία*, η ημερομηνία που έγινε η δοσοληψία και *ποσό* το ποσό της δοσοληψίας. Ο πίνακας αυτός δεν έχει κλειδί γιατί οι τιμές κανενός από τα πεδία του δεν μπορούν να καθορίσουν μοναδικά κάθε γραμμή του. Για να αποκτήσει κλειδί ο πίνακας αυτός θα έπρεπε να του προστεθεί ένα ακόμα πεδίο, ο αριθμός λογαριασμού της δοσοληψίας, οπότε ο συνδυασμός αριθμού λογαριασμού και αριθμού δοσοληψίας θα αποτελούσαν κλειδί.

Εστω ότι ο πίνακας έχει τα παρακάτω δεδομένα

δοσοληψίες

αριθμός-δοσοληψίας	ημερομηνία	ποσό
11	12/05/90	5000
11	03/11/91	1.000.000

Επίσης έστω ότι στην οντοκεντρική βάση έχουμε το μοντέλο που φαίνεται στο Σχήμα 4.14, όπου υποθέτουμε ότι κάθε περίπτωση της κλάσης *Δοσοληψίες* πρέπει να είναι μοναδική με μία ημερομηνία



Σχήμα 4.14: Μοντέλο Δοσοληψίας στην οντοκεντρική βάση

και ένα ποσό. Εαν θελήσουμε να μεταφέρουμε τα δεδομένα του πίνακα *δοσοληψίες* σε αυτή την οντοκεντρική βάση, θα πρέπει να ορίσουμε τις απεικονίσεις :

FOR TABLE δοσοληψίες
FROM : αριθμός-δοσοληψίας
TO : ημερομηνία
BECOMES_LINK ημερομηνία

FOR TABLE δοσοληψίες
FROM : αριθμός-δοσοληψίας
TO : ποσό
BECOMES_LINKποσό

Το αποτέλεσμα θα είναι να μεταφερθούν στη οντοκεντρική βάση δύο δοσοληψίες με τον ίδιο αριθμό και διαφορετικές ημερομηνίες και ποσά. Δηλαδή θα μεταφερθεί η δοσοληψία με κωδικό 11 με δύο ημερομηνίες 12/05/90 και 03/11/91 και δύο ποσά 5000 και 1.000.000. Κάτι τέτοιο θα οδηγήσει την οντοκεντρική βάση σε ασυνεπή κατάσταση.

Παρόμοιες καταστάσεις μπορούν να εμφανιστούν και για άλλες περιπτώσεις εφαρμογής των τελεστών σε πίνακες χωρίς κύριο κλειδί. Η λύση σε αυτές τις περιπτώσεις είναι να προστεθεί στον πίνακα το κατάλληλο γνώρισμα ώστε να αποκτήσει κλειδί και κατόπιν να εφαρμοστούν οι τελεστές.

Στην παρούσα εργασία θεωρούμε ότι όλοι οι πίνακες που συμμετέχουν στην επικοινωνία έχουν κύριο κλειδί το οποίο πρέπει να δηλώνεται όταν ορίζεται το σχεσιακό σχήμα.

Κεφάλαιο 5

Μετάφραση πράξεων στα δεδομένα

Στο κεφάλαιο 3 αναφερθήκαμε στο σύστημα μεταφοράς των δεδομένων και στο κεφάλαιο 4 αναφερθήκαμε στα είδη των απεικονίσεων που μπορούν να οριστούν ανάμεσα σε δύο σχήματα. Στο κεφάλαιο αυτό θα αναφερθούμε στο πως ακριβώς μεταφράζονται οι διάφορες αλλαγές, που εφαρμόζονται στα δεδομένα της σχεσιακής βάσης,από το σύστημα μεταφοράς δεδομένων, κατά τη μεταφορά τους στην οντοκεντρική βάση. Η μετάφραση των αλλαγών αυτών εξαρτάται από το είδος των απεικονίσεων που έχουν οριστεί για τα δεδομένα. Οι αλλαγές αυτές προκύπτουν από την εφαρμογή των πράξεων της *εισαγωγής, ενημέρωσης* και *διαγραφής* στα δεδομένα της σχεσιακής βάσης. Όποια από αυτές συμβεί στη σχεσιακή βάση, σε πίνακα που συμμετέχει στην επικοινωνία των δύο βάσεων, πρέπει να εφαρμοστεί και στα δεδομένα της οντοκεντρικής βάσης.

Για να είναι πάντα επιτυχείς οι μεταφράσεις των αλλαγών που έχουν προκύψει από πράξεις ενημέρωσης και διαγραφής απαιτείται ο χρήστης του οντοκεντρικού συστήματος να μην αλλοιώνει τα δεδομένα που εισάγονται από τη σχεσιακή βάση. Σε διαφορετική περίπτωση θα δημιουργείται ασυνέπεια μεταξύ των δύο βάσεων, την οποία το σύστημα μεταφοράς δεδομένων δεν μπορεί να ελέγξει. Ο χρήστης του οντοκεντρικού συστήματος μπορεί να αναγνωρίσει τα δεδομένα που έχουν προέλθει από τη σχεσιακή βάση από το γνώρισμα με όνομα *comes_from_base*[1] το οποίο έχει σαν τιμή το όνομα της βάσης από όπου προέρχονται.

Στις επόμενες ενότητες θα μελετηθεί η κάθε πράξη ξεχωριστά και η εφαρμογή της σε σχέση με τις απεικονίσεις που έχουν οριστεί. Πρέπει να αναφερθεί ότι ο *Μηχανισμός Μετάφρασης και Μεταφοράς Δεδομένων* (ενότητα 3.3.5), πριν μεταφέρει οποιαδήποτε αλλαγή έχει εφαρμοστεί στα δεδομένα της σχεσιακής βάσης στην οντοκεντρική, κάνει τα εξής :

1. Εξετάζει αν έχει οριστεί συνθήκη (*ενότητα 4.2.3*) για την εφαρμογή της απεικόνισης και αν έχει, εξετάζει αν ισχύει πριν προχωρήσει στην παραγωγή κώδικα για μεταφορά της αλλαγής στην οντοκεντρική βάση.

[1] Το γνώρισμα *comes_from_base* προστίθεται σε όλα τα δεδομένα που προέρχονται από τη σχεσιακή βάση κατά τη μεταφορά τους στην οντοκεντρική.

2. Εξετάζει για την εφαρμογή των πράξεων της διαγραφής και της ενημέρωσης αν έχει επιτραπεί από το χρήστη η μεταφορά τους (*ενότητα 3.3*).

3. Σε περίπτωση δεδομένων για τα οποία δεν έχει οριστεί απεικόνιση, αγνοούνται οι αλλαγές που τους έχουν συμβεί και δεν γίνεται καμία μεταφορά στην οντοκεντρική βάση.

5.1 Εισαγωγή

Κατά την πράξη της *εισαγωγής* εισάγονται δεδομένα ή σχέσεις μεταξύ δεδομένων στη βάση. Η διαδικασία που ακολουθείται σε κάθε περίπτωση ανάλογα με την απεικόνιση φαίνεται παρακάτω.

5.1.1 Η εισαγωγή σε σχέση με την απεικόνιση BECOMES

Οπως είδαμε στην *ενότητα 4.3.1*, με τον τελεστή *BECOMES* μπορούμε να καθορίσουμε απεικονίσεις, με τις οποίες δεδομένα πεδίων ενός πίνακα γίνονται περιπτώσεις κάποιας κλάσης του οντοκεντρικού σχήματος. Σε αυτήν την περίπτωση για να γίνει η μεταφορά ελέγχεται εάν τα δεδομένα που περιέχουν αυτά τα πεδία είναι κενά. Εαν δεν είναι κενά, παράγεται κώδικας στην γλώσσα εισαγωγής δεδομένων της οντοκεντρικής βάσης που τα κάνει περιπτώσεις της καθορισμένης κλάσης.

5.1.2 Η εισαγωγή σε σχέση με την απεικόνιση ISA

Οπως είδαμε στην *ενότητα 4.3.2*, με τον τελεστή *ISA* μπορούμε να καθορίσουμε απεικονίσεις, με τις οποίες δεδομένα πεδίων ενός πίνακα γίνονται υποκλάσεις κάποιας κλάσης του οντοκεντρικού σχήματος. Σε αυτήν την περίπτωση για να γίνει η μεταφορά ελέγχεται αν τα δεδομένα που περιέχουν αυτά τα πεδία είναι κενά. Εαν δεν είναι κενά, παράγεται κώδικας στην γλώσσα εισαγωγής δεδομένων της οντοκεντρικής βάσης που τα κάνει υποκλάσεις της καθορισμένης κλάσης.

5.1.3 Η εισαγωγή σε σχέση με την απεικόνιση BECOMES_LINK

Οπως είδαμε στην *ενότητα 4.3.3*, με τον τελεστή *BECOMES_LINK* μπορούμε να καθορίσουμε απεικονίσεις, με τις οποίες δεδομένα πεδίων ενός πίνακα ενώνονται με συνδέσμους κάποιας συγκεκριμένης κατηγορίας με άλλα πεδία του ίδιου πίνακα. Εαν λοιπόν οι τιμές των πεδίων από όπου ξεκινάει και των πεδίων όπου καταλήγει ο σύνδεσμος που θα φτιαχτεί δεν είναι κενές, παράγεται κώδικας στην γλώσσα εισαγωγής δεδομένων της οντοκεντρικής βάσης που δημιουργεί τον καθορισμένο σύνδεσμο ανάμεσα στα δεδομένα.

5.1.4 Η εισαγωγή σε σχέση με την απεικόνιση BECOMES_DEPENDENT_OBJECT

Οπως είδαμε στην *ενότητα 4.3.4*, με τον τελεστή *BECOMES_DEPENDENT_OBJECT* μπορούμε να καθορίσουμε απεικονίσεις, με τις οποίες τα δεδομένα ενός πεδίου απεικονίζονται στην οντοκεντρική

βάση σε ένα μονοπάτι αποτελούμενο από οντότητες και συνδέσμους. Εαν λοιπόν η τιμή του πεδίου αυτού δεν είναι κενή, παράγεται κώδικας στην γλώσσα εισαγωγής δεδομένων της οντοκεντρικής βάσης που δημιουργεί τις οντότητες και τους σύνδεσμους ανάμεσα τους, που αποτελούν το μονοπάτι που έχει οριστεί.

5.2 Διαγραφή

Η πράξη της *διαγραφής* είναι αυτή κατά την οποία διαγράφουμε δεδομένα ή σχέσεις μεταξύ δεδομένων. Η πράξη αυτή προϋποθέτει ότι τα δεδομένα που προσπαθούμε να διαγράψουμε έχουν εισαχθεί στην οντοκεντρική βάση γιατί αλλιώς θα σημειωθεί λάθος. Επειδή η *διαγραφή* είναι πράξη που πρέπει να εφαρμόζεται με μεγάλη προσοχή, το σύστημα δεν πραγματοποιεί τελικά όλες τις διαγραφές που συναντάει κατά τη μεταφορά των δεδομένων. Αυτό που κάνει είναι ότι καλεί το χρήστη να αποφασίσει, αν θέλει όντως να γίνουν διαγραφές και του δίνει τον τρόπο να το κάνει. Η διαδικασία που ακολουθείται σε κάθε περίπτωση ανάλογα με την απεικόνιση φαίνεται παρακάτω.

5.2.1 Η διαγραφή σε σχέση με την απεικόνιση BECOMES

Διαγραφή, στην περίπτωση που έχει οριστεί απεικόνιση *BECOMES*, σημαίνει ότι το δεδομένο πρέπει να διαγραφεί από περίπτωση της αντίστοιχης κλάσης. Το δεδομένο όμως αυτό σαν περίπτωση της συγκεκριμένης κλάσης μπορεί να έχει συνδέσμους που ξεκινάνε από αυτό και τους έχει κληρονομήσει από αυτήν την κλάση. Για να μπορεί λοιπόν να διαγραφεί σαν περίπτωση της, σύμφωνα με τους κανόνες της γλώσσας εισαγωγής δεδομένων της οντοκεντρικής βάσης, πρέπει πρώτα να διαγραφούν αυτοί οι σύνδεσμοι. Αυτό που κάνει το σύστημα σε αυτή την περίπτωση είναι :

1. Ελέγχει αν το αντικείμενο που έχει προέλθει από τα δεδομένα που προσπαθούμε να σβήσουμε έχει εισαχθεί στην οντοκεντρική βάση. Αν έχει, συνεχίζει στο επόμενο βήμα, αν δεν έχει, σταματάει τη λειτουργία του και ενημερώνει τον χρήστη για την ασυνέπεια.

2. Εντοπίζει όλους τους συνδέσμους που ξεκινάνε από αυτό το αντικείμενο και τους κληρονομεί από την κλάση, από την οποία θέλουμε να το διαγράψουμε.

3. Παράγει κώδικα για τη διαγραφή των συνδέσμων και του δεδομένου σαν περίπτωση της συγκεκριμένης κλάσης.

4. Γράφει τον κώδικα αυτό σε ένα αρχείο με το όνομα του δεδομένου και ενημερώνει το χρήστη να εφαρμόσει το αρχείο αυτό στην οντοκεντρική βάση, αν θέλει να κάνει τις αλλαγές. Η ενημέρωση του χρήστη γίνεται με μήνυμα στην οθόνη που τον παραπέμπει να εξετάσει το αρχείο *warning_com*. Στο αρχείο αυτό δίνονται ακριβείς οδηγίες στο χρήστη για το τι πρέπει να κάνει.

Υπάρχει η περίπτωση το δεδομένο, μετά από αυτή τη διαγραφή, να μην είναι περίπτωση ή υποκλάση καμίας άλλης κλάσης και κανείς σύνδεσμος να μην δείχνει ή ξεκινάει από αυτό. Αυτό σημαίνει ότι το δεδομένο αυτό δεν συνδέεται με κανένα άλλο στην οντοκεντρική βάση και πιθανώς πρέπει να σβηστεί. Το σύστημα ελέγχει αυτήν την περίπτωση στο *βήμα 2* και αν την συναντήσει παράγει κώδικα που σβήνει τελείως το δεδομένο από την οντοκεντρική βάση. Τέλος στο *βήμα 4* ενημερώνει το χρήστη και για αυτό εκτός από τα άλλα.

5.2.2 Η διαγραφή σε σχέση με την απεικόνιση ISA

Διαγραφή στην περίπτωση που έχει οριστεί απεικόνιση *ISA* σημαίνει ότι το δεδομένο πρέπει να διαγραφεί από υποκλάση της αντίστοιχης κλάσης. Το δεδομένο όμως αυτό σαν υποκλάση της συγκεκριμένης κλάσης μπορεί να έχει συνδέσμους που ξεκινάνε από αυτό, που τους έχει κληρονομήσει από αυτήν την κλάση. Για να μπορεί λοιπόν να διαγραφεί σαν υποκλάση της, σύμφωνα με τους κανόνες της γλώσσας εισαγωγής δεδομένων της οντοκεντρικής βάσης, πρέπει πρώτα να διαγραφούν αυτοί οι σύνδεσμοι. Σε αυτήν την περίπτωση το σύστημα λειτουργεί με τον ίδιο τρόπο όπως στην περίπτωση του *BECOMES*.

5.2.3 Η διαγραφή σε σχέση με την απεικόνιση BECOMES_LINK

Διαγραφή στην περίπτωση που έχει οριστεί απεικόνιση *BECOMES_LINK* σημαίνει ότι πρέπει να σβηστεί ένας συγκεκριμένος σύνδεσμος ανάμεσα σε οντότητες που έχουν προέλθει από τα δεδομένα της σχεσιακής. Αυτή είναι η πιο απλή περίπτωση διαγραφής. Το σύστημα παράγει κώδικα για τη διαγραφή του συνδέσμου στη γλώσσα εισαγωγής δεδομένων της οντοκεντρικής βάσης και την ενημερώνει αμέσως χωρίς να ερωτηθεί ο χρήστης.

5.2.4 Η διαγραφή σε σχέση με την απεικόνιση BECOMES_DEPENDENT_OBJECT

Διαγραφή στην περίπτωση που έχει οριστεί απεικόνιση *BECOMES_DEPENDENT_OBJECT* σημαίνει ότι πρέπει να διαγραφεί ολόκληρο το μονοπάτι που είχε παραχθεί από το συγκεκριμένο δεδομένο. Πρέπει δηλαδή να διαγραφούν οι οντότητες και οι σύνδεσμοι που αποτελούν το μονοπάτι. Το σύστημα λειτουργεί σε αυτή την περίπτωση με τον ίδιο τρόπο όπως στην περίπτωση του *BECOMES* μόνο που εξετάζει ξεχωριστά κάθε αντικείμενο του μονοπατιού και τέλος καλείται ο χρήστης να αποφασίσει τι θέλει να κάνει.

5.3 Ενημέρωση

Η πράξη της *ενημέρωσης* είναι αυτή κατά την οποία αλλάζουμε το όνομα κάποιου αντικειμένου της οντοκεντρικής βάσης ή τη σχέση του με άλλα αντικείμενα. Η *ενημέρωση* προϋποθέτει ότι τα δεδομένα που προσπαθούμε να αλλάξουμε έχουν εισαχθεί στην οντοκεντρική βάση. Επειδή μεταφέρουμε

δεδομένα που προέρχονται από πίνακες και η ενημέρωση είναι ιδιόμορφη πράξη είναι απαραίτητο να εξετάσουμε την ενημέρωση σε συνάρτηση με τις αλλαγές που μπορεί να συμβαίνουν σε ένα πίνακα. Εστω λοιπόν ο παρακάτω *πίνακαςX* :

Ονομα_στήλης1	Ονομα_στήλης2	Ονομα_στήλης3	...
A	B	Γ	...

Η διαδικασία που ακολουθείται σε κάθε περίπτωση ανάλογα με την απεικόνιση είναι η εξής.

5.3.1 Η ενημέρωση σε σχέση με την απεικόνιση BECOMES

Εστω ότι έχουμε ορίσει την απεικόνιση

FOR TABLE tableX
A **BECOMES** B

δηλαδή τα δεδομένα του πεδίου A να γίνουν περιπτώσεις των αντίστοιχων δεδομένων του πεδίου B.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,-[2],γ1,...), η οποία ενημερώνεται σε (α1,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *εισαγωγή*. Εισάγεται το α1 σαν περίπτωση του β1. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν εισαγωγή.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α1,-,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *διαγραφή*. Διαγράφεται το α1 σαν περίπτωση του β1. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν διαγραφή.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α1,β2,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση μια *ενημέρωση*. Διαγράφεται το α1 σαν περίπτωση του β1 και εισάγεται σαν περίπτωση του β2. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι η πράξη αυτή αποτελεί ενημέρωση και για τις δύο βάσεις.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α2,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *αλλαγή του ονόματος* του α1 σε α2.

Εστω ότι είχε οριστεί απεικόνιση

FOR TABLE tableX
A **BECOMES** K

[2]Το σύμβολο παύλα (-) συμβολίζει κενό δεδομένο.

που μεταφέρει τα δεδομένα του πεδίου A να γίνουν περιπτώσεις μιας σταθερής κλάσης K στην οντοκεντρική βάση. Τότε θα είχαμε :

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α2,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *αλλαγή του ονόματος* του α1 σε α2.

5.3.2 Η ενημέρωση σε σχέση με την απεικόνιση ISA

Εστω ότι έχουμε ορίσει την απεικόνιση

FOR TABLE tableX
A **ISA** B

δηλαδή τα δεδομένα του πεδίου A να γίνουν υποκλάσεις των αντίστοιχων δεδομένων του πεδίου B.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,-,γ1,...), η οποία ενημερώνεται σε (α1,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *εισαγωγή*. Εισάγεται το α1 σαν υποκλάση του β1. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν εισαγωγή.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α1,-,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *διαγραφή*. Διαγράφεται το α1 σαν υποκλάση του β1. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν διαγραφή.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α1,β2,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση μια *ενημέρωση*. Διαγράφεται το α1 σαν υποκλάση του β1 και εισάγεται σαν υποκλάση του β2. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι η πράξη αυτή αποτελεί ενημέρωση και για τις δύο βάσεις.

Εαν είχε οριστεί η απεικόνιση

FOR TABLE tableX
A **ISA** K

δηλαδή τα δεδομένα του πεδίου A να γίνουν υποκλάσεις μιας σταθερής κλάσης στην οντοκεντρική βάση τότε η μόνη περίπτωση που θα είχαμε είναι:

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (-,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *διαγραφή*. Διαγράφεται το α1 σαν υποκλάση της K.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (-,β1,γ1,...), η οποία ενημερώνεται σε (α1,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *εισαγωγή*. Εισάγεται το α1 σαν υποκλάση της Κ. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν εισαγωγή.

5.3.3 Η ενημέρωση σε σχέση με την απεικόνιση BECOMES_LINK

Εστω ότι έχουμε ορίσει την απεικόνιση

FOR TABLE tableX
FROM : A
TO : B
BECOMES_LINK linkX

δηλαδή τα δεδομένα του πεδίου Α να συνδεθούν με τα αντίστοιχα δεδομένα του πεδίου Β μέσω του συνδέσμου με όνομα linkX.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,-,γ1,...), η οποία ενημερώνεται σε (α1,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *εισαγωγή*. Εισάγεται από το α1 ένας σύνδεσμος προς το β1. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν εισαγωγή.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α1,-,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση *διαγραφή*. Διαγράφεται ο σύνδεσμος που δείχνει από το α1 στο β1. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι μια πράξη που για την σχεσιακή βάση είναι ενημέρωση στην οντοκεντρική μεταφράζεται σαν διαγραφή.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α1,β2,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση μια *ενημέρωση*. Διαγράφεται ο σύνδεσμος που δείχνει από το α1 στο β1 και εισάγεται ο σύνδεσμος που δείχνει από το α1 στο β2. Βλέπουμε λοιπόν σε αυτήν την περίπτωση ότι η πράξη αυτή αποτελεί ενημέρωση και για τις δύο βάσεις.

5.3.4 Η ενημέρωση σε σχέση με την απεικόνιση BECOMES_DEPENDENT_OBJECT

Εστω ότι έχουμε ορίσει την απεικόνιση

FOR TABLE tableX
A **BECOMES_DEPENDENT_OBJECT**
PATH = Π1,Π2,..,ΠΝ
SEPARATOR = -

δηλαδή τα δεδομένα του πεδίου Α να δημιουργήσουν στην οντοκεντρική βάση τα μονοπάτια Π1,Π2,..,ΠΝ.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (-,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση διαγραφή των μονοπατιών που δημιουργήθηκαν από το α1.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (-,β1,γ1,...), η οποία ενημερώνεται σε (α1,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση εισαγωγή των μονοπατιών που δημιουργούνται από το α1.

- Εστω ότι έχει εισαχθεί στον πίνακα *tableX* η γραμμή (α1,β1,γ1,...), η οποία ενημερώνεται σε (α2,β1,γ1,...). Η ενημέρωση αυτή οδηγεί στο να γίνει στην οντοκεντρική βάση διαγραφή των μονοπατιών που δημιουργήθηκαν από το α1 και εισαγωγή των μονοπατιών που δημιουργούνται από το α2.

5.4 Θέματα συνέπειας δεδομένων

5.4.1 Σειρά εισαγωγής δεδομένων στην οντοκεντρική βάση

Ενα ακόμα θέμα που αφορά την μεταφορά δεδομένων από τη σχεσιακή στην οντοκεντρική βάση είναι η σειρά με την οποία εισάγονται οι αλλαγές στα δεδομένα στην οντοκεντρική. Όπως αναφέρθηκε στο *κεφάλαιο 3* πρέπει η σειρά με την οποία εφαρμόζονται οι αλλαγές στην οντοκεντρική βάση να είναι η ίδια ακριβώς με τη σειρά που συμβαίνουν στη σχεσιακή βάση για να διατηρούνται τα δεδομένα των δύο βάσεων πάντα συνεπή. Για το λόγο αυτό φυλάσσεται και η σειρά που συμβαίνουν οι αλλαγές στη σχεσιακή βάση. Ας δούμε όμως γιατί έχει σημασία η σειρά των αλλαγών :

- Στην περίπτωση της *εισαγωγής* η σειρά των αλλαγών έχει σημασία γιατί εάν θελήσεις να εισαγάγεις ένα σύνδεσμο από ένα αντικείμενο Α σε ένα αντικείμενο Β (Α->Β) προφανώς το Β πρέπει να έχει ήδη εισαχθεί αλλιώς θα σημειωθεί λάθος.

- Στην περίπτωση της *διαγραφής* η σειρά των αλλαγών έχει σημασία γιατί εάν θελήσεις να διαγράψεις ένα αντικείμενο Α σαν περίπτωση μιας κλάσης Β ή σαν υποκλάση μιας κλάσης Γ ή ένα σύνδεσμο από το αντικείμενο Α σε ένα αντικείμενο Δ, πρέπει πρώτα οποιοδήποτε από αυτά να έχει ήδη εισαχθεί αλλιώς θα σημειωθεί λάθος.

- Τέλος στην περίπτωση της *ενημέρωσης* η σειρά των αλλαγών έχει σημασία γιατί εάν θεωρήσουμε τις αλλαγές Α,Β,Γ πάνω σε κάποιο αντικείμενο το αποτέλεσμα που θα πάρουμε δεν θα είναι το ίδιο αν τις εφαρμόσουμε σαν Α,Β,Γ ή Α,Γ,Β ή Γ,Α,Β κτλ.

5.4.2 Ελεγχόμενο λεξιλόγιο

Με τον όρο *ελεγχόμενο λεξιλόγιο* εννοούμε στις σχεσιακές βάσεις ένα μηχανισμό που περιορίζει το χρήστη κατά την επικοινωνία του με τη βάση να χρησιμοποιεί ορισμένους μόνο όρους που έχουν ήδη δηλωθεί. Εάν τώρα στη σχεσιακή βάση έχουμε ελεγχόμενο λεξιλόγιο δεν μπορούμε να ισχυριστούμε ότι μετά τη μεταφορά των δεδομένων θα διατηρηθεί και στην οντοκεντρική.

Για παράδειγμα έστω ότι στην σχεσιακή βάση έχουμε δύο πίνακες, έναν που περιγράφει τύπους αντικειμένων και είναι ο *types*, και έναν που περιγράφει σχέσεις των αντικειμένων με τους τύπους τους και είναι ο *objtypes* και η περιγραφή των πινάκων είναι η παρακάτω.

types

typename

objtypes

objectName	typename

Εστω επίσης ότι έχουμε ελεγχόμενο λεξιλόγιο στον καθορισμό του τύπου των αντικειμένων, πράγμα που σημαίνει ότι ο τύπος που δίνει ένας χρήστης σε ένα αντικείμενο πρέπει πρώτα να έχει εισαχθεί στον πίνακα των τύπων *types* αλλιώς το σύστημα δεν του επιτρέπει να συνδέσει με αυτόν το αντικείμενο.

Εαν η μεταφορά των δεδομένων των παραπάνω πινάκων στην οντοκεντρική βάση δεν γίνει με κάποιο τρόπο που να διατηρεί τις παραπάνω σχέσεις αυτές θα χαθούν.

Κεφάλαιο 6

Βοήθεια προς το χρήστη για καθορισμό απεικονίσεων

Για να μπορέσει να λειτουργήσει το σύστημα μεταφοράς δεδομένων ικανοποιητικά, πρέπει ο χρήστης να ορίσει το κατάλληλο σύνολο απεικονίσεων για τη μεταφορά των δεδομένων. Στο κεφάλαιο αυτό θα προσπαθήσουμε να δώσουμε κάποιες γενικές οδηγίες που θα βοηθήσουν το χρήστη στην προσπάθεια του να βρει το σύνολο των απεικονίσεων που είναι κατάλληλο για τις εφαρμογές του.

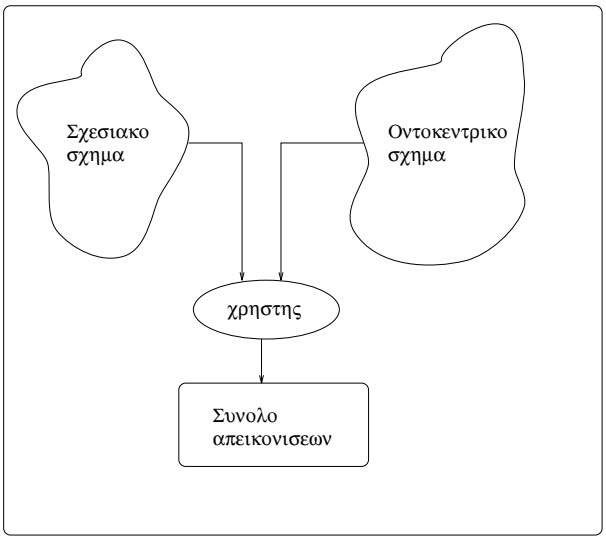
6.1 Αντληση πληροφορίας για καθορισμό απεικονίσεων με τη βοήθεια του σχεσιακού και του οντοκεντρικού σχήματος

Η υπόθεση που έχει γίνει σε όλη την εργασία είναι ότι ο χρήστης που θα ορίσει τις απεικονίσεις έχει καλή γνώση του σχεσιακού και του οντοκεντρικού σχήματος. Εαν δεν έχει αυτές τις γνώσεις ένας και μόνο χρήστης, οι απεικονίσεις μπορούν να οριστούν με τη συνεργασία δύο ή περισσότερων χρηστών που έχουν το κατάλληλο σύνολο γνώσεων.

Μπορούμε να φανταστούμε το χρήστη σαν ένα σύστημα που παίρνει σαν είσοδο το σχεσιακό και το οντοκεντρικό σχήμα και αποφασίζει τι απεικονίσεις πρέπει να ορίσει ανάμεσα τους (Σχήμα 6.1). Προσπαθώντας λοιπόν να διατυπώσουμε κάποια μέθοδο που πρέπει να ακολουθήσει ο χρήστης για να διευκολυνθεί στο να ορίσει τις απεικονίσεις αυτές σημειώνουμε τα παρακάτω.

Πρώτα από όλα είναι βασικό να αποφασίσει ο χρήστης *ποιά δεδομένα* θέλει να μεταφερθούν από τη σχεσιακή στην οντοκεντρική βάση. Για να το κάνει αυτό μπορεί να του παρέχεται από το σύστημα η δυνατότητα να βλέπει το σχεσιακό σχήμα σε κάποιο ειδικό εργαλείο παρουσίασης του σχεσιακού σχήματος. Ο χρήστης βλέποντας το σχεσιακό σχήμα και γνωρίζοντας το είδος των εφαρμογών που θέλει να εφοδιάσουν το σύστημα μεταφοράς δεδομένων μπορεί να αποφασίσει το σύνολο των δεδομένων που θέλει να μεταφερθούν.

Εχοντας τώρα υπόψη αυτό το σύνολο δεδομένων, ο χρήστης μπορεί να εντοπίσει ποιά από αυτά



Σχήμα 6.1: Συνοπτική άποψη της διαδικασίας καθορισμού απεικονίσεων

τα δεδομένα αποτελούν ανεξάρτητες οντότητες. Κατόπιν μπορεί να εντοπίσει και τις σχέσεις που υπάρχουν ανάμεσα τους. Οι σχέσεις αυτές μπορεί να είναι *σχέσεις υποσυνόλου* και *σχέσεις κλειδιών* όπως αναφέρθηκε στην ενότητα 4.2.1.

Τα δεδομένα που βρήκε ο χρήστης ότι αποτελούν ανεξάρτητες οντότητες μεταφέρονται γενικά στην οντοκεντρική βάση και πάλι σαν ανεξάρτητες οντότητες, κάτω από κλάσεις που αντιπροσωπεύουν τον τύπο τους. Οι σχέσεις υποσυνόλου μπορούν να μεταφερθούν στην οντοκεντρική βάση σαν σχέσεις γενίκευσης/ειδίκευσης ή σαν σχέσεις γνωρίσματος μεταξύ ανεξάρτητων οντοτήτων. Ενω οι σχέσεις κλειδιών μεταφράζονται πάντα σαν σχέσεις γνωρίσματος μεταξύ ανεξάρτητων οντοτήτων.

Με τις παραπάνω πληροφορίες ο χρήστης μπορεί να σχηματίσει ένα πιθανό οντοκεντρικό σχήμα για την παράσταση των δεδομένων του στην οντοκεντρική βάση. Ετσι θα έχει κάποια ιδέα για το πως περίπου πρέπει να μεταφερθούν τα δεδομένα του και θα μπορεί να κρίνει στο τέλος αν τον ικανοποιούν οι απεικονίσεις που θα ορίσει. Επίσης το σχήμα αυτό μπορεί να τον βοηθήσει να έχει μια πιο ξεκάθαρη άποψη για το τι θα πρέπει να αναζητήσει στην οντοκεντρική βάση για να ορίσει ευκολότερα τις απεικονίσεις. Ενα τέτοιο σχήμα αντί να το φτιάξει μόνος του ο χρήστης θα μπορούσε να το πάρει από ένα εργαλείο μετάφρασης σχεσιακού σχήματος στο οντοκεντρικό μοντέλο. Στο εργαλείο αυτό θα έπρεπε να δώσει σαν είσοδο το υποσύνολο του σχεσιακού σχήματος που θέλει να μεταφέρει και θα πάρει σαν έξοδο ένα πιθανό οντοκεντρικό σχήμα [29],[41],[37],[48].

Το σχήμα αυτό τώρα ο χρήστης μπορεί να το συγκρίνει με το υπάρχον σχήμα της οντοκεντρικής βάσης. Για να εξερευνήσει ευκολότερα το σχήμα της οντοκεντρικής βάσης μπορεί να του παρέχεται από το σύστημα η δυνατότητα να βλέπει μια γραφική παράσταση του σχήματός της εκφρασμένη σε κάποιο κατάλληλο εργαλείο (π.χ [17]).

Το πιο καλό σε αυτήν την προσπάθεια είναι να εντοπιστούν πρώτα οι κλάσεις του οντοκεντρικού

σχήματος, κάτω από τις οποίες θα μεταφερθούν σαν περιπτώσεις οι ανεξάρτητες οντότητες που έχουν επιλέγει. Για να γίνει αυτό ο χρήστης μπορεί ή να διερευνήσει το οντοκεντρικό σχήμα αναζητώντας τις κατάλληλες κλάσεις ή να αναζητήσει με κάποιο μηχανισμό που θα του παρέχεται από το σύστημα (όπως grep) δομές του οντοκεντρικού σχήματος που έχουν το ίδιο ή παρόμοιο όνομα με το είδος των οντοτήτων που θέλει να μεταφέρει. Οταν τελειώσει μπορεί να κάνει τις πρώτες απεικονίσεις χρησιμοποιώντας τον τελεστή *BECOMES* (ενότητα 4.3.1). Αν βρει ότι κάποια από τις ανεξάρτητες οντότητες που θέλει να μεταφέρει δεν αντιστοιχεί σε μια κλάση αλλά σε ένα ολόκληρο μονοπάτι από οντότητες και συνδέσμους πρέπει να χρησιμοποιήσει για να κάνει τις απεικονίσεις αυτές τον τελεστή *BECOMES_DEPENDENT_OBJECT* (ενότητα 4.3.4).

Αφού εντοπίσει λοιπόν ο χρήστης που θα μεταφερθούν όλες οι ανεξάρτητες οντότητες πρέπει να εξετάσει πώς μπορούν να οριστούν ανάμεσά τους και οι σχέσεις που έχει βρει ότι ισχύουν. Τις σχέσεις αυτές θα τις ορίσει χρησιμοποιώντας τους τελεστές *ISA* (ενότητα 4.3.2) και *BECOMES_LINK* (ενότητα 4.3.3). Τον τελεστή *ISA* θα τον χρησιμοποιήσει για ορισμό σχέσεων συνόλου/υποσυνόλου και τον τελεστή *BECOMES_LINK* για σχέσεις οντότητας-γνωρίσματος.

Οταν τελειώσει με τα παραπάνω ο χρήστης μπορεί να επανεξετάσει το οντοκεντρικό σχήμα και τις απεικονίσεις που έχει ορίσει. Κάνοντας αυτό μπορεί να δει αν υπάρχουν και άλλες απεικονίσεις που θα ήθελε να κάνει με δομές της οντοκεντρικής βάσης, που θα ταξινομούσαν καλύτερα τα δεδομένα που μεταφέρονται από τη σχεσιακή στην οντοκεντρική βάση.

Κεφάλαιο 7

Επίλογος

7.1 Συμπεράσματα

Στην εργασία αυτήν ορίστηκε ένα σύνολο τελεστών για απεικόνιση σχεσιακών σχημάτων σε οντοκεντρικά. Επίσης παρουσιάστηκε ένα σύστημα μεταφοράς δεδομένων, που δημιουργήθηκε για να μεταφέρει δεδομένα από μια σχεσιακή σε μια οντοκεντρική βάση χρησιμοποιώντας απεικονίσεις που ορίζονται με τους τελεστές που αναφέρθηκαν.

Το σύνολο των τελεστών, που ορίστηκε στο κεφάλαιο 4, καλύπτει την ανάγκη απεικόνισης των δομών ενός σχεσιακού σχήματος σε δομές ενός οντοκεντρικού σχήματος. Ορίστηκαν συνολικά τέσσερα είδη τελεστών. Με τους τελεστές αυτούς μπορούν να απεικονιστούν οντότητες του ενός σχήματος σε οντότητες του άλλου. Επίσης μπορούν να απεικονιστούν σχέσεις υποσυνόλου και σχέσεις κλειδιών, που ισχύουν ανάμεσα σε οντότητες του σχεσιακού στις αντίστοιχες σχέσεις στο οντοκεντρικό σχήμα. Η υπόθεση που γίνεται κατά την απεικόνιση αυτή είναι ότι το σχεσιακό και το οντοκεντρικό σχήμα προϋπάρχουν και δεν αλλοιώνονται με την εφαρμογή των τελεστών. Τις απεικονίσεις αυτές τις ορίζει ο χρήστης εξετάζοντας το σχεσιακό σχήμα, το οντοκεντρικό σχήμα και τις ανάγκες των εφαρμογών του.

Το σύστημα μεταφοράς δεδομένων που δημιουργήθηκε (κεφάλαιο 3), αποτελεί προσέγγιση στο πρόβλημα της κοινωνίας δεδομένων σε ετερογενείς βάσεις δεδομένων και ειδικότερα στο πρόβλημα της μεταφοράς δεδομένων. Το σύστημα ενεργεί πάνω σε δύο ετερογενείς βάσεις δεδομένων, μια σχεσιακή και μια οντοκεντρική. Ανήκει στην κατηγορία των *ομοσπονδιακών συστημάτων διαχείρισης βάσεων δεδομένων* και ειδικότερα είναι ένα *χαλαρά-συνδεδεμένο ομοσπονδιακό σύστημα διαχείρισης βάσεων δεδομένων*, όπως ορίστηκε στο κεφάλαιο 2. Σκοπός του συστήματος είναι να μεταφέρει δεδομένα από μια οποιαδήποτε σχεσιακή βάση σε μια οντοκεντρική βάση υλοποιημένη στην γλώσσα Telos. Για να το κάνει αυτό χρησιμοποιεί τις απεικονίσεις, που έχουν οριστεί ανάμεσα στα δύο σχήματα με τους τελεστές, που αναφέρθηκαν στην προηγούμενη παράγραφο.

Το σύστημα μεταφοράς δεδομένων εφαρμόστηκε επιτυχώς στην μεταφορά δεδομένων από το σχεσιακό σύστημα *MITOS/CM* στο οντοκεντρικό σύστημα *ΚΛΕΙΩ* για το Μουσείο Μπενάκη. Με

χρήση του συνόλου τελεστών που περιγράφηκε ορίστηκαν με επιτυχία όλες οι απεικονίσεις που χρειάζονταν ανάμεσα στα δύο σχήματα. Τα προβλήματα που παρουσιάστηκαν κατά τον ορισμό των ζητούμενων απεικονίσεων οφείλονταν κυρίως στο ότι το σχεσιακό σχήμα δεν ήταν σε τρίτη κανονική μορφή και αντιμετωπίστηκαν με κανονικοποίηση του. Αποτέλεσμα της μεταφοράς ήταν η εισαγωγή χιλίων περίπου αντικειμένων και πληροφοριών γύρω από αυτά, από το *MITOS/CM* στο *ΚΛΕΙΩ*.

7.2 Επεκτάσεις-Βελτιώσεις

Το σύνολο των τελεστών θα μπορούσε να επεκταθεί έτσι ώστε να υποστηρίζεται απεικόνιση δομών και από άλλα μοντέλα παράστασης δεδομένων στο οντοκεντρικό. Επίσης θα μπορούσαν να υποστηρίζονται και απεικονίσεις από αρχεία στο οντοκεντρικό σχήμα. Αυτό θα οδηγούσε το σύστημα μεταφοράς δεδομένων να μπορεί να λειτουργήσει και ανάμεσα σε άλλων ειδών βάσεις έχοντας βέβαια πάντα σαν στόχο την οντοκεντρική βάση.

Το σύνολο των τελεστών θα μπορούσε, επίσης, να επεκταθεί και με νέα είδη των τελεστών. Ετσι θα μπορούσαν να προστεθούν τελεστές με τους οποίους : (1) θα υποστηρίζεται ο μηχανισμός Time_primitive, (2) θα μπορεί ένας υπάρχον σύνδεσμος να γίνει περίπτωση περισσοτέρων από ένα κατηγοριών, (3) θα μπορεί ο χρήστης να δίνει όνομα στους συνδέσμους που δημιουργούνται, (4) θα μπορούν να ορίζονται ‘‘αντίστροφοι σύνδεσμοι’’ κ.τ.λ

Μια ακόμα βελτίωση θα ήταν να επεκταθεί το σύστημα μεταφοράς δεδομένων, για να υποστηρίζει και *σύγχρονη μεταφορά* δεδομένων από το ένα σύστημα στο άλλο. Ετσι οι αλλαγές, που θα συνέβαιναν στο σύστημα-πηγή θα γίνονταν αμέσως γνωστές στο σύστημα-στόχο. Μια τέτοια προσθήκη θα έκανε το σύστημα ικανό, να υποστηρίξει μεταφορά δεδομένων ανάμεσα σε εφαρμογές, όπου υπάρχει κρισιμότητα για συνέπεια των δεδομένων κάθε στιγμή.

Χρήσιμη θα ήταν και προσθήκη ενός μηχανισμού αναγνώρισης των δεδομένων, που προέρχονται από τη σχεσιακή, ο οποίος δεν θα επιτρέπει στο χρήστη να τα αλλοιώνει. Ο μηχανισμός αυτός μπορεί να στηριχθεί στην τιμή του γνωρίσματος *comes_from_base* το οποίο αποκτούν τα δεδομένα όταν προέρχονται από τη σχεσιακή βάση κατά την εισαγωγή τους στην οντοκεντρική και έχει τιμή το όνομα της βάσης από όπου προέρχονται.

Επίσης το σύστημα θα μπορούσε να επεκταθεί για να υποστηρίζει *αμφίδρομη επικοινωνία*. Αυτό σημαίνει να μπορεί και η βάση-πηγή να μαθαίνει τι αλλαγές γίνονται στα δεδομένα της οντοκεντρικής. Αν γινόταν κάτι τέτοιο ο χρήστης θα μπορούσε να επεμβαίνει στα δεδομένα, που μεταφέρονται από το ένα σύστημα στο άλλο χωρίς να δημιουργείται πρόβλημα. Για να γίνει κάτι τέτοιο απαιτείται και διαμόρφωση του συνόλου των τελεστών για να υποστηρίζουν και τις αντίστροφες απεικονίσεις.

Μια ακόμα χρήσιμη επέκταση του συστήματος θα ήταν η απόκτηση ενός μηχανισμού, που να βοηθάει το χρήστη να ορίσει τις κατάλληλες απεικονίσεις ανάμεσα στα σχήματα. Το εργαλείο αυτό, θα μπορούσε να παίρνει σαν εισόδους τα δύο σχήματα και να παράγει ένα σύνολο πιθανών απεικονίσεων,

από το οποίο ο χρήστης θα μπορούσε να επιλέξει. Επίσης θα μπορούσε να βοηθάει το χρήστη να εντοπίζει εναλλακτικές απεικονίσεις για τα ίδια δεδομένα (αν υπάρχουν) και από αυτές να επιλέγει την καλύτερη.

Παράρτημα Α

Ευρετήριο όρων

aggregation	συνάθροιση
association	συσχέτιση
autonomy	αυτονομία
browsing	εξερεύνηση
canonical data model	κανονικό μοντέλο δεδομένων
category	κατηγορία
centralized system	κεντρικό σύστημα
class-subclass relation	σχέση κλάσης-υποκλάσης
component database	βάση-συνιστώσα
component schema	σχήμα-συνιστώσα
conflict	σύγκρουση
controlled vocabulary	ελεγχόμενο λεξιλόγιο
data exchange system	σύστημα ανταλλαγής δεδομένων
data sharing	κοινωνία δεδομένων
database converter	μετατροπέας βάσης δεδομένων
descriptive conflicts	συγκρούσεις περιγραφής
distributed system	κατανεμημένο σύστημα
distribution	κατανομή
entity relationship model	μοντέλο οντοτήτων-συσχετίσεων
entry forms	δελτία εισαγωγής δεδομένων
equivalence relation	σχέση ισοδυναμίας
export schema	εξαγόμενο σχήμα
extended entity relationship model	επεκτεταμένο μοντέλο οντοτήτων-συσχετίσεων
external schema	εξωτερικό σχήμα
federated schemas	ομοσπονδιακά σχήματα

federated database	ομοσπονδιακή βάση δεδομένων
federated database management system	ομοσπονδιακό σύστημα διαχείρισης βάσεων δεδομένων
federated system	ομοσπονδιακό σύστημα
foreign key	ξένο κλειδί
generalization	γενίκευση
global schema multidatabase systems	συστήματα πολυβάσεων δεδομένων με καθολικό σχήμα
heterogeneity	ετερογένεια
heterogeneity conflicts	συγκρούσεις ετερογένειας
heterogeneous databases	ετερογενείς βάσεις δεδομένων
heterogeneous multibatabase system	ετερογενές σύστημα πολυβάσεων δεδομένων
homogeneous multibatabase system	ομογενές σύστημα πολυβάσεων δεδομένων
homogeneous multidatabase language system	ομογενή γλωσσικά συστήματα πολυβάσεων δεδομένων
inclusion dependencies	συναρτησιακές εξαρτήσεις τύπου *"περιέχεται"*
instance	μέλος, περίπτωση
instanceof relations	σχέσεις τύπου "περίπτωσης"
integration	συνδυασμός
interoperable database systems	διαλειτουργικά συστήματα βάσεων δεδομένων
interoperable systems	διαλειτουργικά συστήματα
join	καρτεσιανό γινόμενο
key dependencies	συναρτησιακές εξαρτήσεις κλειδιών
knowledge acquisition	απόκτηση γνώσης
link	σύνδεσμος
local schema	τοπικό σχήμα
loosely coupled FDBMS	χαλαρά-συνδεδεμένα ΟΣΔΒΔ
multidatabase language system	γλωσσικά συστήματα πολυβάσεων δεδομένων
multidatabase system	σύστημα πολυβάσεων δεδομένων
multiple federations	πολλαπλές ομοσπονδίες
network model	δικτυωτό μοντέλο
non-federated systems	μη-ομοσπονδιακά συστήματα
path	μονοπάτι
postfix	επίθημα
prefix	πρόθημα
primary key	κύριο κλειδί
relational model	σχεσιακό μοντέλο
schema conversion	μετατροπή σχήματος
schema mapping	απεικόνιση σχημάτων

schema translation	μετάφραση σχημάτων
semantic conflicts	σημασιολογικές συγκρούσεις
single federation	απλή ομοσπονδία
stuctural conflicts	δομικές συγκρούσεις
tightly coupled FDBMS	στενά-συνδεδεμένα ΟΣΔΒΔ
unified multidatabase system	ενοποιημένο σύστημα πολυβάσεων δεδομένων
weak entities	ασθενείς οντότητες

Παράρτημα Β

Ορισμός σχεσιακού σχήματος

Στο παράρτημα αυτό παρουσιάζεται : (1) το συντακτικό με το οποίο ο χρήστης μπορεί να ορίσει το σχεσιακό σχήμα και (2) το μοντέλο με το οποίο αποθηκεύεται το σχεσιακό σχήμα στην οντοκεντρική βάση εκφρασμένο σε γλώσσα Telos.

Β.1 Συντακτικό ορισμού σχεσιακού σχήματος

Στο συντακτικό με το οποίο ορίζεται το σχεσιακό σχήμα οι δηλώσεις γίνονται με την εξής σειρά :

1. όνομα της σχεσιακής βάσης, της οποίας το σχήμα θα δηλώσουμε
2. πίνακες της βάσης,
3. κύρια κλειδιά όλων των πινάκων,
4. καρτεσιανά γινομένα που ισχύουν ανάμεσα στους πίνακες και τέλος
5. σχέσεις τύπου υποσυνόλου ή οι σχέσεις κλειδιών που ισχύουν ανάμεσα στους πίνακες.

Από τις παραπάνω δηλώσεις αυτές, του ονόματος της βάσης, των πινάκων και του κύριου κλειδιού κάθε πίνακα είναι υποχρεωτικές, ενώ οι υπόλοιπες είναι προαιρετικές. Στο παρακάτω συντακτικό όποια λέξη είναι με κεφαλαία, τονισμένα γράμματα είναι δεσμευμένη λέξη και μπορεί να γραφτεί και με μικρά γράμματα.

<Δήλωση ονόματος σχεσιακής βάσης δεδομένων>

DATABASE <όνομα σχεσιακής βάσης>
<Δήλωση πινάκων>

[**TABLE** <όνομα πίνακα> **HAS** <όνομα πεδίου>[,<όνομα πεδίου>]*]*

<Δήλωση κύριων κλειδιών πινάκων>

[**FOR TABLE** <όνομα πίνακα>
PRIMARY_KEY = <όνομα πεδίου>[,<όνομα πεδίου>]*]*

<Δήλωση καρτεσιανών γινομένων μεταξύ πινάκων>

[**FOR TABLE** <όνομα πίνακα>
FOR FIELD <όνομα πεδίου>[,<όνομα πεδίου>]
SELECT < όνομαπεδίου >[,<όνομα πεδίου>]*
FROM <όνομα πίνακα>, <όνομα πίνακα>
WHERE <όνομα πίνακα>.<όνομα πεδίου>=<όνομα πίνακα>.<όνομα πεδίου>
[**AND** <όνομα πίνακα>.<όνομα πεδίου>=<όνομα πίνακα>.<όνομα πεδίου>]*]*

<Δήλωση σχέσεων μεταξύ πινάκων[1]>

[<όνομα πίνακα> **SUBRELATION** <όνομα πίνακα>]*
[<όνομα πίνακα> **SUBRELATION** <όνομα πίνακα>]*

Β.2 Μοντέλο αποθήκευσης σχεσιακού σχήματος στην Telos

Το μοντέλο με το οποίο αποθηκεύεται το σχεσιακό σχήμα στην οντοκεντρική βάση εκφρασμένο σε γλώσσα Telos είναι το παρακάτω :

TELL Individual DB_Name **in S_Class**
end DB_Name
TELL Individual Columns **in S_Class**
end Columns

[1]Οι σχέσεις τύπου υποσυνόλου ή κλειδιών μπορούν να δηλώνονται με οποιαδήποτε σειρά και ανακατεμένες

TELL Individual Tables **in S_Class**
with attribute
databaseName **:** DB_Name ;
column **:** Columns ;
subrelation **:** Tables ;
keyRelated **:** Tables ;
hasTransformations **:** Transformations ;
primaryKey **:** Columns ;
hasJoins **:** Table_Joins
end Tables
TELL Individual Table_Joins **in S_Class**
with attribute
fromColumn **:** Columns ;
fromTable **:** Tables ;
toColumn **:** Columns ;
toTable **:** Tables ;
sqlcmd **:** Telos_String
end Table_Joins
TELL AttributeClass comes_from_base
components
from **:** Individual
to **:** DB_Name
in S_Class
end comes_from_base

Παράρτημα Γ

Ορισμός απεικονίσεων

Στο παράρτημα αυτό παρουσιάζεται: (1) το συντακτικό, με το οποίο ο χρήστης μπορεί να ορίσει απεικονίσεις ανάμεσα σε ένα σχεσιακό και ένα οντοκεντρικό σχήμα και (2) το μοντέλο, με το οποίο αποθηκεύονται οι απεικονίσεις στην οντοκεντρική βάση. Το μοντέλο περιγράφεται στην γλώσσα Telos.

Γ.1 Συντακτικό ορισμού απεικονίσεων

Κατά τον ορισμό των απεικονίσεων, ανάμεσα σε ένα σχεσιακό και ένα οντοκεντρικό σχήμα πρώτα πρέπει να δηλώνεται το όνομα της σχεσιακής βάσης και κατόπιν να ακολουθούν οι δηλώσεις των απεικονίσεων με οποιαδήποτε σειρά. Στο παρακάτω συντακτικό όποια λέξη είναι με κεφαλαία, τονισμένα γράμματα είναι δεσμευμένη λέξη και μπορεί να γραφτεί και με μικρά γράμματα.

<Δήλωση ονόματος σχεσιακής βάσης δεδομένων>

DATABASE <όνομα σχεσιακής βάσης>

<Δήλωση απεικόνισης τύπου BECOMES>

FOR TABLE <όνομα πίνακα>
<όνομα πεδίου>[,<όνομα πεδίου>]*
BECOMES <όνομα πεδίου ή κλάσης >[,<όνομα πεδίου>]*
[**FROM_PREFIX :** πρόθημα]
[**FROM_POSTFIX :** επίθημα]
[**TO_PREFIX :** πρόθημα]
[**TO_POSTFIX :** επίθημα]
[**IF** <όνομα πεδίου> = <τιμή>]

<Δήλωση απεικόνισης τύπου ISA>

FOR TABLE <όνομα πίνακα>
<όνομα πεδίου>[,<όνομα πεδίου>]*
ISA <όνομα πεδίου ή κλάσης>[,<όνομα πεδίου>]*
[**FROM_PREFIX :** πρόθημα]
[**FROM_POSTFIX :** επίθημα]
[**TO_PREFIX :** πρόθημα]
[**TO_POSTFIX :** επίθημα]
[**IF** <όνομα πεδίου> = <τιμή>]

<Δήλωση απεικόνισης τύπου BECOMES_LINK>

FOR TABLE <όνομα πίνακα>
FROM : <όνομα πεδίου>[,<όνομα πεδίου>]*
TO : <όνομα πεδίου>[,<όνομα πεδίου>]*
BECOMES_LINK <όνομα συνδέσμου>
[**FROM_PREFIX :** πρόθημα]
[**FROM_POSTFIX :** επίθημα]
[**TO_PREFIX :** πρόθημα]
[**TO_POSTFIX :** επίθημα]
[**IF** <όνομα πεδίου> = <τιμή>]

<Δήλωση απεικόνισης τύπου BECOMES_DEPENDENT_OBJECT >

FOR TABLE <όνομα πίνακα>
<όνομα πεδίου> **BECOMES_DEPENDENT_OBJECT**
PATH : <όνομα κόμβου>**.**<όνομα συνδέσμου>^<όνομα κόμβου>
[**.**<όνομα συνδέσμου>^<όνομα κόμβου>]*
[[**,** <όνομα κόμβου>**.**<όνομα συνδέσμου>^<όνομα κόμβου>] [**.**<όνομα συνδέσμου>^<όνομα κόμβου>]*]*
[**FROM_PREFIX :** πρόθημα]
[**FROM_POSTFIX :** επίθημα]
[**IF** <όνομα πεδίου> = <τιμή>]
SEPARATOR = < \ ή - >

Γ.2 Μοντέλο αποθήκευσης απεικονίσεων σε Telos

TELL Individual Transformations **in S_Class**
with attribute
kind **:** Telos_String;
fromValue **:** Telos_String;
toValue **:** Telos_String ;
fromPrefix **:** Telos_String ;
fromPostfix **:** Telos_String ;
toPrefix **:** Telos_String ;
toPostfix **:** Telos_String ;
linkName **:** Telos_String ;
path **:** Telos_String ;
seperator **:** Telos_String;
conditionField **:** Telos_String;
conditionValue **:** Telos_String
end Transformations

Βιβλιογραφία

[1] Ιφιγένεια Διονυσιάδου και Λουκάς Καράμπελας. Καταγραφή απαιτήσεων για τη ζεύξη των ΜΙΤΟΣ/CM-ΚΛΕΙΩ. Τεχνική Εκθεση ΠΟΛΕΜΩΝ.ΜΜ.95.ΚΕ # 3, Μουσείο Μπενάκη, Απρίλιος 1995.

[2] Κώστας Νταντουρής. Βιβλιοθήκη στοιχειωδών ερωτηματικών συναρτήσεων και επεξεργασία ερωτήσεων στη γλώσσα Telos. Μεταπτυχιακή εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Ιούλιος 1993.

[3] Μαρία Χριστοφοράκη. Τεκμηρίωση πολιτιστικών αγαθών με το σύστημα ΚΛΕΙΩ. Μεταπτυχιακή εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Φεβρουάριος 1994.

[4] Ιφιγένεια Διονυσιάδου. Παρατηρήσεις για το μοντέλο και την εμφάνιση των δεδομένων στο ΜΙΤΟΣ/CM-ΚΛΕΙΩ. Τεχνική Εκθεση ΠΟΛΕΜΩΝ.ΜΜ.95.ΚΕ # 1, Μουσείο Μπενάκη, Φεβρουάριος 1995.

[5] Alex Borgida, Matthias Jarke, Manolis Koubarakis and John Mylopoulos. Telos: Representing Knowledge About Information Systems. *ACM Transactions on Information Systems*, 8(4):pp. 352--362, October 1990.

[6] Andreas Meier, Rolf Dippold, Jacky Mercerat, Alex Muriset, Jean-Claude Untersinger, Robert Eckerlin and Flavio Ferrara. Hierarchical to Relational Database Migration. *IEEE Software*, 11(3):pp. 21--27, May 1994.

[7] Paolo Atzeni and Riccardo Torlone. A metamodel approach for the management of multiple models and the translation of schemes. *Information Systems , Pergamon Press*, 18(6):pp. 349--362, September 1993.

[8] C. Batini, M. Lenzerini and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4):pp. 322--364, December 1986.

[9] Alfonso F. Cardenas. Heterogeneous Distributed Database Management:The HD-DBMS. *Proceedings of the IEEE*, 75(5):pp.588--600, May 1987.

[10] M. Casanova and J. Amarel. Designing ER Schemas from Conventional Information Systems. *Proceedings, 3rd International Conference on the Entity-Relationship Approach, New York*, 1983.

[11] Malu Castellanos. A Methodology for Semantically Enriching Interoperable Databases. *Proceedings, 11th British National Conference on Databases, Keele, 1993, pp.58-75.*

[12] Malu Castellanos and Felix Saltor. Extraction of Data Dependencies. In: Jaakkola, Kangassalo, Kitahashi and Markus (eds), Information Modelling and Knowledge Bases V (3rd European-Japanese Seminar on Information Modelling and Knowledge Bases , Budapest, 1993). IOS Press, Amsterdam,pp. 400-420,1994.

[13] P.P Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Databases Systems*, 1(1):pp.9--36, March 1976.

[14] Christine Parent, Stefano Spaccapietra and Yahu Dupont. Model Independent Assertions for Integration of Heterogeneous Schemas. *VLDB Journal,pp. 81--126*, January 1992.

[15] Chin-Wan Chung. DATAPLEX : An Access to Heterogeneous Distributed Databases. *Communications of the ACM*, 33(1):pp.70--80, January 1990.

[16] Panos Constantopoulos and Martin Doerr. Component Classification in the Software Information Base . In: O. Nierstrasz and D. Tsichritzis, eds., Object-Oriented Software Composition,Prentice-Hall, 1995.

[17] Costas Dadouris, Polykarpos Karamaounas, Nikos Prekas and Maria Theodoridou. SIS Graphical Analysis Interface User's Manual, Working Paper # 1. Information Systems and Software Technology Group, Institute of Computer Science Foundation of Research and Technology Hellas, October 1993.

[18] D. H. Fishman, D. Beech, H.P. Cate, E.C. Chow, T. Connors, J.W. Davis, N. Derrett, C.G. Hoch, W. Kent, P. Lyngbaek, B. Mahbod, M.A. Neimat, T.A. Ryan and M.C. Shan. IRIS : An Object-Oriented Database Management System. *ACM Transactions on Office Information Systems*, 5(1):pp. 48--69, January 1987.

[19] Dimitris Daskalakis and Yannis T. Tzitzikas. Customizing Data Entry Forms, Working Paper #9. Information Systems and Software Technology Group, Institute of Computer Science Foundation of Research and Technology Hellas, October 1994.

[20] C.J. Date, editor. *An Introduction to Database Systems.* Addison Wesley, Reading Massachussetts, 1986.

[21] Davis and Arora. Converting a Relational Database Model into an ER Model. *Proceedings, 6th International Conference on the Entity-Relationship Approach, New York*, 1987.

[22] Felix Saltor, Malu Castellanos and Manuel Garcia-Solaco. Overcoming Schematic Discrepancies in Interoperable Databases. In: Hsiao, Neuhold and Sacks-Davis (eds) Interoperable Database Systems (Proceedings, IFIP WG2.6 Database Semantics Conference on Interoperable Database Systems , Lorne, Victoria, Australia, 1992), 1993, pp. 191-205.

[23] Felix Saltor, Malu Castellanos and Manuel Garcia-Solaco. Suitability of Data Models as Canonical Models for Federated Databases. *ACM SIGMOD Record*, 20(4):pp. 44--48, December 1991.

[24] Gomes Thomas, Glenn R. Thompson, Chin-Wan Chung, Edward Barkmeyer, Fred Carter, Marjorie Templeton, Stephen Fox and Berl Hartman. Heterogeneous Distributed Database Systems for Production Use. *ACM Computing Surveys*, 22(3):pp. 237--266, September 1990.

[25] David K. Hsiao. Federated Databases and Systems : Part I - A Tutorial on Their Data Sharing. *VLDB Journal,pp. 127--179*, January 1991.

[26] Richard Hull. Relative information capacity of simple relational schemata. *SIAM Journal of Computing*, 15(3):pp. 856--886, 1986.

[27] Richard Hull and Roger King. Semantic Database Modeling : Survey, applications and research issues. *ACM Computing Surveys*, 19(3):pp. 201--260, September 1987.

[28] Johannesson and Kalman. A Method from Translating Relational Schemas into Conceptual Schemas. *Proceedings, 8th International Conference on the Entity-Relationship Approach ,Toronto*, 1989.

[29] Joseph Albert, Rafi Ahmed, Mohammed Ketabchi, William Kent and Ming-Chien Shan. Automatic Importation of Relational Schemas in Pegasus. In: Proceedings, 3rd International Workshop on Research Issues in Data Engineering : Interoperability in Multidatabase Systems, Vienna, Austria, pp. 105--113, 1993.

[30] Won Kim. *Introduction to Object-Oriented Databases.* MIT Press, 1990.

[31] Won Kim and Frederick H. Lochovsky, editors. *Object-Oriented Concepts, Databases and Applications.* ACM Press, Addison Wesley, 1989.

[32] Won Kim and Jungyun Seo. Classifying Schematic and Data Heterogeneity in Multidatabase Systems. *IEEE Computer*, 24(12):pp. 12--18, December 1991.

[33] S. N. Koshafian and G. P. Copeland. Object Identity. *SIGPLAN Notices (Proceedings ACM Conference on Object-Oriented Programming Systems, Languages and Applications)*, 21(11):pp. 417--423, 1986.

[34] James A. Kowal. *Analyzing Systems.* Prentice-Hall, New Jersey, 1988.

[35] Edmund Y. Lien. On the Equivalence of Database Models. *Journal of the ACM*, 29(2):pp. 333--362, April 1982.

[36] Malu Castellanos, Felix Saltor and Manuel Garcia-Solaco. A Canonical Model for the Interoperability among Object-Oriented and Relational Databases. In: Ozsu, Dayal and Valduriez (eds) Distributed Object Management (Proceedings, International Workshop on Distributed Object Management, Edmonton, Canada, 1992). Morgan Kaufmann, 1994, pp. 309--314.

[37] Malu Castellanos, Felix Saltor and Manuel Garsia-Salaco. Semantically Enriching Relational Databases into an Object-Oriented Semantic Model . In: D. Karagiannis (ed.): Database and Expert Systems Applications (5th International Conference DEXA'94, Athens, 1994). Springer Verlag, 1994, pp. 125--134.

[38] Dhamir N. Mannai and Khaled Bugrara. Enhancing Inter-Operability and Data Sharing In Medical Information Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington (May 26-28)*, volume 22, pp. 495--498, June 1993.

[39] Manuel Garcia-Solaco, Malu Castellanos and Felix Saltor. Discovering Interdatabase Resemblance of Classes for Interoperable Databases. In: Schek, Sheth and Czejdo (eds) Proceedings, 3rd International Workshop on Research Issues in Data Engineering: Interoperability in Multidatabase Systems, Vienna, Austria, 1993, pp.26--33.

[40] Victor Markowitz and Arie Shoshani. Representing Extended Entity-Relationship Structures in Relational Databases: A Modular Approach. *ACM Transactions on Database Systems*, 17(3):pp. 423--464, September 1992.

[41] Victor M. Markowitz and Johann A. Makowsky. Identifying Extended Entity-Relationship Structures in Relational Schemas. *IEEE Transactions On Software Engineering*, 16(8):pp. 77--90, August 1990.

[42] Victor M. Markowitz and Arie Shoshani. On the correctness of representing extented entity-relashionship structures in the relational model. *Proceeding 1989 SIGMOD*, pp. 430--439, 1989.

[43] Martin Doerr, Manos Theodorakis and Polivios Klimathianakis. SIS Data Entry Language Users's Manual, Working Paper #2. Information Systems and Software Technology Group, Institute of Computer Science Foundation of Research and Technology Hellas, December 1992.

[44] William Premerlani Michael Blaha and Hwa Shen. Converting OO-Models into RDBMS Schema. *IEEE Software*, 11(3):pp. 28--39, May 1994.

[45] M.N. Bright, A.R. Hurson and Simin H. Pakrad. A Taxonomy and Current Issues in Multidabase Systems. *IEEE Computer*, 25(3):pp. 50--59, March 1992.

[46] M.P. Reddy, B.E. Prasad, P.G. Reddy and A. Gupta. A Methodology for Integration of Heterogeneous Databases. *IEEE Transactions on Knwoledge and Data Engineering*, 6(6):pp. 920--933, December 1994.

[47] Nan C. Shu, Barron C. Housel and Vincent Y. Lum. CONVERT : A High Level Translation Definition Language for Data Convertion. *Communications of the ACM*, 18(10):pp. 557--567, October 1975.

[48] Navathe and Awong. Abstracting Relational and Hierarchical Data with a Semantic Data Model. *Proceedings, 6th International Conference on the Entity-Relationship Approach, New York*, 1987.

[49] Omran Bukhres, Jiansan Chen, Ahmed K. Elmagarmid, Xiangning Liu and James G. Mullen. InterBase : A Multidatabase Prototype System . In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington (May 26-28)*, volume 22, pp. 534--539, June 1993.

[50] Omran Bukhres, Jiansan Chen and Rod Pezzali. An InterBase System at BNR. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington (May 26-28)*, volume 22, pp. 426--429, June 1993.

[51] Tamer M. Ozsu and Patrick Valduriez. Distributed Database Systems : Where Are We Now ? *IEEE Computer*, 24(8):pp. 68--78, August 1991.

[52] P. Atzeni, G. Ausiello, C. Batini and M. Moscarini. Inclusion and equivalence between relational database schemata. *Theoretical Computer Science*, 19(2):pp. 267--285, 1982.

[53] P. Fankhauser, R. Busse, M. Kaul, E. Neuhold, M. Oheiner and Y. Xu. Database Research at IPSI (The KODIM Project). *SIGMOD Record*, 21(1):pp. 135, March 1992.

[54] Pam Drew, Roger King, Dennis Mcleod, Marek Rusinkiewicz and Avi Silberschatz. Report of the WorkShop on Semantic Heterogeneity and Interoperation in Multidatabase Systems. *SIGMOD Record*, 22(3):pp. 47--55, September 1993.

[55] R. Chiang, T. Barron and V. Stoney . Performance Evaluation of Reverse Engineering Relational Databases into Extended Entity-Relationship Models. *Proceedings, 12th International Conference on the Entity-Relationship Approach ,Arlington*, 1993.

[56] Rafi Ahmed, Philippe De Smeth, Weimin Dn, William Kent, Mohammed A. Ketabchi, Witold A. Litwin, Abbas Rafii and Ming-Chien Shan. The Pegasus Heterogeneous MultiDatabase System. *IEEE Computer*, 24(12):pp. 19--26, December 1991.

[57] Sudha Ram. Heterogeneous Distributed Database Systems. *IEEE Computer*, 24(12):pp. 7--10, December 1991.

[58] R.J. Miller, Y.E. Ioannidis and R. Ramakrishnan. Schema equivalence in heterogeneous systems : Bridging theory and practice. *Information Systems*, 19(1):pp. 3--31, January 1994.

[59] Marek Rusinkiewicz. Panel on Multi-Database Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois (June 1-3)*, pp. 3--5, 1988.

[60] Ming-Chien Shan. Pegasus Architecture and Design Principles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington (May 26-28)*, volume 22, pp. 422--425, June 1993.

[61] Amit P. Sheth. Semantic Issues in Multidatabase Systems . *SIGMOD Record*, 20(4):pp.5--9, December 1991.

[62] Amit P. Sheth and James A. Carson. Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Database. *ACM Computing Surveys*, 22(3):pp.183--236, September 1990.

[63] Toby J. Teorey, Dongqing Yang and James P. Fry. A logical design methodology for relational databases using the extented entiry-relashionship model. *ACM Computing Surveys*, 18(2):pp. 197--222, June 1986.

[64] Jeffrey D. Ullman. *Principles of Database And Knowledge-Base Systems Volume I : Classical Database Systems.* Computer Science Press, 1988.

[65] M. Papazoglou M. McLoughlin E. Lindsay S. Willie. Database Research at the Queensland University of Technology. *SIGMOD Record*, 21(4), December 1992.

[66] Marianne Winslett. Multiple Data Source Access System Research at DARPA. *SIGMOD Record*, 21(3):pp. 67, September 1992.

[67] Witold Litwin, Leo Mark and Nick Roussopoulos. Interoperability of Multiple Autonomous Database. *ACM Computing Surveys*, 22(3):pp. 267--293, September 1990.