



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρότυπο Σύστημα Ολοκλήρωσης Σχημάτων με Υποστήριξη
Οντολογιών για Κατανεμημένα Περιβάλλοντα Αυτόνομων
Κόμβων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΜΑΡΙΑΣ ΕΙΡΗΝΗΣ Π. ΠΟΛΙΤΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρότυπο Σύστημα Ολοκλήρωσης Σχημάτων με Υποστήριξη Οντολογιών για Κατανεμημένα Περιβάλλοντα Αυτόνομων Κόμβων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΜΑΡΙΑΣ ΕΙΡΗΝΗΣ Π. ΠΟΛΙΤΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13^η Σεπτεμβρίου 2007.

(Υπογραφή)

.....
Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Νεκτάριος Κοζύρης
Επικ. Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2007

(Υπογραφή)

.....
MARIA ΕΙΡΗΝΗ Π. ΠΟΛΙΤΟΥ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Μαρία Ειρήνη, Π. Πολίτου, 2007

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κ. Τίμο Σελλή, για την επιστημονική του προσφορά και το ενδιαφέρον που μου έδειξε, καθώς και για την καθοδήγηση του για τις μελλοντικές μου επιλογές. Επίσης, θα ήθελα να ευχαριστήσω την συνεπιβλέπουσα, υπ. διδάκτωρ Βηρένα Καντερέ, για τον χρόνο που μου αφιέρωσε κατά την διάρκεια της εκπόνησης της εργασίας αυτής και την πολύτιμη και καθοριστική της βοήθεια. Τέλος, ευχαριστώ την οικογένειά μου, που στάθηκε δίπλα μου με αγάπη και κατανόηση.

Περίληψη

Η ανάπτυξη του Σημασιολογικού Ιστού και η ανάγκη για έναν πιο ευέλικτο και αποδοτικό τρόπο ανταλλαγής δεδομένων με καλώς ορισμένο νόημα, οδήγησε στα συστήματα ομότιμων κόμβων βασισμένα σε σχήμα (schema-based peer-to-peer systems). Στα συστήματα αυτά κάθε κόμβος χρησιμοποιεί ένα σχήμα βάσης δεδομένων με βάση το οποίο οργανώνει τα τοπικά διαθέσιμα δεδομένα. Το ζήτημα λοιπόν που προκύπτει είναι πώς θα μπορούν οι κόμβοι να αναζητούν και να ανταλλάσσουν πληροφορίες μεταξύ τους, διατηρώντας όμως την αυτονομία τους. Μία μη ευέλικτη λύση απαιτεί την ύπαρξη ενός κεντρικού σχήματος το οποίο θα χρησιμοποιούν όλοι οι κόμβοι. Μία δεύτερη λύση είναι κάθε κόμβος να έχει ένα δικό του μοναδικό σχήμα. Όμως, λόγω της ανομοιογένειας των σχημάτων των ΒΔ, για την ανταλλαγή πληροφοριών απαιτείται η μετάφραση των ερωτημάτων που τίθενται από έναν κόμβο σε έναν άλλο. Η μετάφραση αυτή πραγματοποιείται με τη βοήθεια αντιστοιχίσεων (mappings) μεταξύ των σχημάτων ΒΔ.

Σκοπός της εργασίας είναι η ανάπτυξη ενός πρότυπου συστήματος διαχείρισης ολοκλήρωσης σχημάτων βάσεων δεδομένων για δίκτυα ομότιμων κόμβων. Συγκεκριμένα, θα υλοποιήσουμε ένα σύστημα που εξάγει ένα κεντρικό ή καθολικό σχήμα για ένα σύνολο κόμβων, όπου ο καθένας έχει το δικό του τοπικό σχήμα ΒΔ. Το κεντρικό αυτό σχήμα προκύπτει από την ενοποίηση των τοπικών σχημάτων με τη βοήθεια κανόνων αντιστοίχισης, οι οποίοι ορίζουν σημασιολογικές σχέσεις μεταξύ των τοπικών σχημάτων. Με τη βοήθεια του καθολικού σχήματος και των αντιστοιχίσεων μεταξύ των τοπικών και του καθολικού σχήματος γίνεται δυνατή μία πιο αποδοτική και ευέλικτη μετάφραση και αποτίμηση ερωτημάτων σε οποιονδήποτε κόμβο το δικτύου.

Λέξεις Κλειδιά: σύστημα ομότιμων βάσεων δεδομένων, σύστημα ομότιμων κόμβων βασισμένο σε σχήμα, καθολικό σχήμα, σημασιολογική ολοκλήρωση δεδομένων, ολοκλήρωση σχημάτων, οντολογίες, αντιστοιχίσεις, συγχώνευση οντολογιών

Abstract

The evolution of the Semantic Web and the need for a more flexible and efficient way of exchanging data with well-defined concept, led to the evolution of the schema-based peer-to-peer systems. In these systems, every node uses a database schema to organise its local data. The matter that arises is how can the nodes search and exchange information, keeping their autonomy. One solution, which is not flexible, requires the existence of a central schema that all nodes use. A second solution is every node to have its own unique schema. However, due to the heterogeneity of the DB schemas, the exchange of information requires the translation of the queries that are posed by a node to an other one. The query translation is obtained by using mappings between the DB schemas.

The purpose of this thesis is the development of a database schema integration management system for peer-to-peer networks, and particularly the implementation of a system which creates a central or global schema for a set of nodes, each of which has its own local DB schema. This global schema is generated by merging the local schemas using mappings, which define semantic relations between the local schemas. Using the global schema and the mappings between the local schemas and the global one, a more efficient and flexible query translation and evaluation in any node of the network becomes possible.

Keywords: database peer-to-peer system, schema-based peer-to-peer system, global schema, semantic data integration, schema integration, ontologies, mappings, ontology merging

Πίνακας περιεχομένων

Κατάλογος Σχημάτων	15
1 Εισαγωγή	17
1.1 Αντικείμενο διπλωματικής.....	18
1.2 Οργάνωση του τόμου.....	18
2 Θεωρητικό υπόβαθρο	21
2.1 Δίκτυα Ομότιμων Κόμβων (Peer to Peer - P2P).....	21
2.1.1 <i>Ανάγκη δημιουργίας P2P συστημάτων</i>	21
2.1.2 <i>Περιγραφή P2P συστημάτων</i>	22
2.1.3 <i>Πλεονεκτήματα και μειονεκτήματα</i>	22
2.1.4 <i>Εφαρμογές</i>	23
2.2 P2P και Βάσεις Δεδομένων.....	24
2.2.1 <i>Είδη βάσεων δεδομένων</i>	24
2.2.2 <i>P2P δίκτυα βάσεων δεδομένων (P2P DB)</i>	25
2.2.3 <i>Συστήματα διαχείρισης δεδομένων Peers (Peer Data Management Systems - PDMSs)</i>	26
2.2.4 <i>Τοπικό σχεσιακό μοντέλο (Local Relational Model - LRM)</i>	27
2.3 Σημασιολογική Ολοκλήρωση Δεδομένων (Semantic Data Integration)	28
2.3.1 <i>Εισαγωγή</i>	28
2.3.2 <i>Συστήματα ολοκλήρωσης δεδομένων (Data integration systems)</i>	30
2.3.3 <i>Ενδιάμεσο / καθολικό σχήμα (Mediated / global schema)</i>	31
2.3.4 <i>Αντιστοίχιση σχημάτων (Schema mapping)</i>	32
2.3.5 <i>Ταίριασμα σχημάτων (Schema matching)</i>	32
2.3.6 <i>Προσεγγίσεις μοντελοποίησης καθολικού σχήματος</i>	34
2.3.7 <i>Επεξεργασία ερωτήματος (Query processing)</i>	40
2.3.8 <i>Μετάφραση ερωτήματος (Query translation)</i>	41
2.4 Οντολογίες (Ontologies)	41
2.4.1 <i>Χρήση</i>	42
2.4.2 <i>Γλώσσες Οντολογιών</i>	44

2.4.3	<i>Αντιστοιχίση οντολογίας (Ontology mapping)</i>	46
2.4.4	<i>Οντολογίες και σημασιολογική ολοκλήρωση</i>	47
2.4.5	<i>Ρόλος των οντολογιών στη σημασιολογική ολοκλήρωση δεδομένων</i>	47
3	Περιγραφή και ανάλυση του προβλήματος	53
3.1	Γενικό πλαίσιο του προβλήματος	53
3.2	Αναλυτική περιγραφή προβλήματος – Ορισμοί εννοιών.....	55
3.2.1	<i>Δομή βάσεων δεδομένων στους κόμβους (peers)</i>	55
3.2.2	<i>Το πρόβλημα της ετερογένειας για την αποτίμηση ερωτημάτων</i>	56
3.2.3	<i>Συσχετίσεις και αντιστοιχίσεις μεταξύ στοιχείων των σχημάτων</i>	56
3.2.4	<i>Εννοιολογικός γράφος</i>	60
4	Ολοκλήρωση σχημάτων με υποστήριξη οντολογιών	61
4.1	Γενικά	61
4.2	Σύντομη περιγραφή.....	62
4.3	Αναλυτική περιγραφή και αλγόριθμοι για την ολοκλήρωση σχημάτων	64
4.3.1	<i>Αναπαράσταση σχήματος από γράφο</i>	64
4.3.2	<i>Συσχετίσεις (correspondences) και κανόνες αντιστοίχισης (mappings rules)</i> ...	64
4.3.3	<i>Αυτόματη εξαγωγή σημασιολογικών σχέσεων από τους κανόνες αντιστοίχισης</i> . 64	
4.3.4	<i>Ενοποίηση οντολογιών και σχημάτων</i>	65
4.3.5	<i>Εξαγωγή κεντρικού σχήματος</i>	66
4.4	Συμπύεση κεντρικού σχήματος.....	66
4.5	Τελικές αντιστοιχίσεις	67
4.6	Παραδείγματα	69
5	Υλοποίηση	89
5.1	Πλατφόρμες και προγραμματιστικά εργαλεία	89
5.2	Λεπτομέρειες Υλοποίησης.....	89
5.2.1	<i>public class Peer</i>	89
5.2.2	<i>public class Schema</i>	90
5.2.3	<i>public class Relation</i>	90
5.2.4	<i>public class Attribute</i>	91
5.2.5	<i>public class Correspondence</i>	91
5.2.6	<i>public class Condition</i>	92

5.2.7	<i>public class Query</i>	92
5.2.8	<i>public class Mapping</i>	93
5.2.9	<i>public class Node</i>	93
5.2.10	<i>public class Edge</i>	94
5.2.11	<i>public class Graph</i>	94
5.2.12	<i>public class Main</i>	96
6	Αξιολόγηση - Πειράματα	105
6.1	Παράμετροι αξιολόγησης	105
6.2	Οργάνωση πειραμάτων	106
6.2.1	Ομοιότητα καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων.....	107
6.2.2	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων	110
6.2.3	Σύγκριση ομοιοτήτων με βάση το βαθμό συμπίεσης.....	111
6.3	Αποτελέσματα.....	112
6.3.1	Ομοιότητα καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων.....	112
6.3.2	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων	133
6.3.3	Σύγκριση ασυμπίεστου καθολικού με συμπιεσμένο καθολικό	147
6.3.4	Σύγκριση ομοιοτήτων με βάση το βαθμό συμπίεσης.....	152
6.4	Συμπεράσματα	167
6.4.1	Ομοιότητα καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων.....	167
6.4.2	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων	168
6.4.3	Σύγκριση ομοιοτήτων με βάση το βαθμό συμπίεσης.....	170
6.5	Σχήματα παραδειγμάτων.....	171
6.5.1	<i>P2P Δίκτυο νοσοκομείων</i>	171
6.5.2	<i>P2P Δίκτυο πανεπιστημίων</i>	198
7	Επίλογος	225
7.1	Σύνοψη και συμπεράσματα.....	225

7.2	Μελλοντικές επεκτάσεις	226
8	Βιβλιογραφία.....	229

Κατάλογος Σχημάτων

Σχήμα 2.1 Προσεγγίσεις ταιριάσματος σχημάτων	33
Σχήμα 2.2 Επέκταση των πηγών για το παράδειγμα.....	39
Σχήμα 2.3 Προσεγγίσεις σχετικές με το ρόλο των οντολογιών.....	49
Σχήμα 4.1 Εννοιολογικός γράφος Σχήματος S1 με HASA συσχετίσεις.....	71
Σχήμα 4.2 Εννοιολογικός γράφος Σχήματος S2 με HASA συσχετίσεις.....	71
Σχήμα 4.3 Εννοιολογικός γράφος συγχώνευσης των σχημάτων S1 και S2 (βήμα 1 του αλγορίθμου 4.3.4).....	73
Σχήμα 4.4 Προσθήκη ακμών Related στον γράφο συγχώνευσης (βήμα 2 του αλγορίθμου 4.3.4).....	74
Σχήμα 4.5 Συγχώνευση στον γράφο συγχώνευσης κόμβων που σχετίζονται με ID (βήμα 3 του αλγορίθμου 4.3.4).....	75
Σχήμα 4.6 Δημιουργία καθολικού γράφου με την προσθήκη ISA σχέσεων στον γράφο συγχώνευσης (βήμα 4 του αλγορίθμου 4.3.4).....	76
Σχήμα 6.1 Συνάρτηση f1: Ομοιότητα κλειδιών συναρτήσει ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων.....	116
Σχήμα 6.2 Συνάρτηση f2: Ομοιότητα κλειδιών συναρτήσει ομοιότητας σχέσεων για τα P2P δίκτυα των πανεπιστημίων.....	120
Σχήμα 6.3 Μεταβολή ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων και των Πανεπιστημίων.....	121
Σχήμα 6.4 Μεταβολή ομοιότητας ιδιοτήτων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων.....	126
Σχήμα 6.5 Μεταβολή ομοιότητας κλειδιών για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων.....	131
Σχήμα 6.6 Διασπορά κλειδιών P2P δικτύου H8.....	132

Σχήμα 6.7 Διασπορά κλειδιών P2P δικτύου H9.....	132
Σχήμα 6.8 Διασπορά κλειδιών P2P δικτύου U9.....	132
Σχήμα 6.9 Διασπορά κλειδιών P2P δικτύου U9.....	133
Σχήμα 6.10 Μεταβολή ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων.....	137
Σχήμα 6.11 Μεταβολή ομοιότητας ιδιοτήτων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων.....	142
Σχήμα 6.12 Μεταβολή ομοιότητας κλειδιών για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων.....	147
Σχήμα 6.13 Κέρδος συμπίεσμένου καθολικού ως προς τις σχέσεις.....	148
Σχήμα 6.14 Απώλεια συμπίεσμένου καθολικού ως προς τις σχέσεις.....	149
Σχήμα 6.15 Κέρδος συμπίεσμένου καθολικού ως προς τα κλειδιά.....	150
Σχήμα 6.16 Απώλεια συμπίεσμένου καθολικού ως προς τα κλειδιά.....	151
Σχήμα 6.17 Ομοιότητα σχέσεων αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων.....	155
Σχήμα 6.18 Ομοιότητα ιδιοτήτων αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων.....	159
Σχήμα 6.19 Ομοιότητα κλειδιών αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων.....	163
Σχήμα 6.20 Ομοιότητα σχέσεων συμπίεσμένου με καθολικό ως προς τις σχέσεις συναρτήσεως του βαθμού συμπίεσης.....	164
Σχήμα 6.21 Ομοιότητα συμπίεσμένου με καθολικό ως προς τις ιδιότητες συναρτήσεως του βαθμού συμπίεσης.....	165
Σχήμα 6.22 Ομοιότητα συμπίεσμένου με καθολικό ως προς τα κλειδιά συναρτήσεως του βαθμού συμπίεσης.....	166

1

Εισαγωγή

Η εξάπλωση του Παγκόσμιου Ιστού που συνεπάγεται μεγάλες απαιτήσεις ανταλλαγής πληροφοριών οδήγησε στην ανάγκη για μία ευέλικτη, δυναμική και περισσότερο κατενεμημένη αρχιτεκτονική συστήματος ανταλλαγής δεδομένων από το κλασικό μοντέλο πελάτη-εξυπηρετητή (client-server). Έτσι, δημιουργήθηκαν τα συστήματα ομότιμων κόμβων (Peer to Peer systems). Επιπλέον, η εξέλιξη του Παγκόσμιου Ιστού σε Σημαιολογικό Ιστό, μέσα στον οποίο διακινείται πληροφορία με καλώς ορισμένο νόημα, δημιούργησε την ανάγκη για πιο εκφραστικές λειτουργίες και οδήγησε στην ανάπτυξη συστημάτων ομότιμων κόμβων βασισμένων σε σχήμα (schema-based peer-to-peer systems).

Στα συστήματα αυτά κάθε κόμβος χρησιμοποιεί ένα σχήμα βάσης δεδομένων με βάση το οποίο οργανώνει τα τοπικά διαθέσιμα δεδομένα. Το ζήτημα λοιπόν που προκύπτει είναι πώς θα μπορούν οι κόμβοι να αναζητούν και να ανταλλάσσουν πληροφορίες μεταξύ τους, διατηρώντας όμως την αυτονομία τους. Για τη λύση του προβλήματος αυτού υπάρχουν δύο προσεγγίσεις. Η πρώτη απαιτεί να υπάρχει ένα κεντρικό σχήμα το οποίο θα χρησιμοποιούν όλοι οι κόμβοι. Αυτή η προσέγγιση δεν είναι ευέλικτη γιατί δεν επιτρέπει τη χρήση πολλών σχημάτων. Η δεύτερη προσέγγιση είναι κάθε κόμβος να έχει ένα δικό του μοναδικό σχήμα. Όμως, λόγω της ανομοιογένειας των σχημάτων των ΒΔ, για την ανταλλαγή πληροφοριών απαιτείται η μετάφραση των ερωτημάτων που τίθενται από έναν κόμβο σε έναν άλλο. Η μετάφραση αυτή πραγματοποιείται με τη βοήθεια αντιστοιχίσεων (mappings) μεταξύ των σχημάτων ΒΔ.

1.1 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη ενός πρότυπου συστήματος διαχείρισης ολοκλήρωσης σχημάτων βάσεων δεδομένων για καταναμημένα περιβάλλοντα αυτόνομων κόμβων, π.χ. δίκτυα ομότιμων κόμβων (P2P schema-based systems). Συγκεκριμένα, σκοπός είναι η υλοποίηση ενός συστήματος που εξάγει ένα κεντρικό ή καθολικό σχήμα (global schema) για ένα σύνολο κόμβων, όπου ο καθένας έχει το δικό του τοπικό σχήμα (local schema) βάσης δεδομένων. Τα τοπικά αυτά σχήματα θεωρούνται υποσύνολα-όψεις (views) του κεντρικού. Το κεντρικό σχήμα δεν είναι ένα σχήμα που υπάρχει εκ των προτέρων και χρησιμοποιούν όλοι οι κόμβοι, όπως απαιτεί η πρώτη προσέγγιση που αναφέραμε παραπάνω, αλλά προκύπτει από την ενοποίηση των τοπικών σχημάτων με τη βοήθεια κανόνων αντιστοίχισης, οι οποίοι ορίζουν σημασιολογικές σχέσεις μεταξύ των τοπικών σχημάτων.

Με τη βοήθεια του καθολικού σχήματος και των αντιστοιχίσεων μεταξύ των τοπικών και του καθολικού σχήματος γίνεται δυνατή μία πιο αποδοτική και ευέλικτη μετάφραση και αποτίμηση ερωτημάτων, σε σχέση και με τις δύο προαναφερθείσες προσεγγίσεις, όπως θα γίνει κατανοητό σε επόμενα κεφάλαια.

1.2 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε 8 κεφάλαια.

Το παρόν κεφάλαιο αποτελεί μία συνοπτική εισαγωγή στο αντικείμενο της εργασίας και την οργάνωση του τόμου.

Στο δεύτερο κεφάλαιο δίνεται το θεωρητικό υπόβαθρο που σχετίζεται με το αντικείμενο της εργασίας. Συγκεκριμένα, περιγράφονται βασικές έννοιες, όπως τα P2P δίκτυα, τα συστήματα διαχείρισης δεδομένων peers, η σημασιολογική ολοκλήρωση δεδομένων και οι οντολογίες.

Το τρίτο κεφάλαιο ασχολείται με την περιγραφή και ανάλυση του προβλήματος με το οποίο θα ασχοληθεί η διπλωματική, καθώς και με την επεξήγηση εννοιών που σχετίζονται άμεσα με το χώρο του προβλήματος και την υλοποίηση της εργασίας.

Στο τέταρτο κεφάλαιο περιγράφεται η λύση που προτείνουμε για την επίλυση του προβλήματος που περιγράφηκε στο τρίτο κεφάλαιο. Συγκεκριμένα δίνεται αναλυτική περιγραφή των διαδικασιών και των αλγορίθμων που χρησιμοποιούμε για την υλοποίηση του συστήματος ολοκλήρωσης σχημάτων.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα εργαλεία υλοποίησης και αναλύεται ο κώδικας που υλοποιεί το σύστημά μας.

Στο έκτο κεφάλαιο αναλύονται τα πειράματα που έγιναν και παρατίθενται τα αποτελέσματα και τα συμπεράσματα που απορρέουν απ' αυτά.

Το έβδομο κεφάλαιο αποτελεί τον επίλογο της εργασίας. Εδώ γίνεται μία σύνοψη των αποτελεσμάτων και συμπερασμάτων της υλοποίησης και δίνονται κατευθύνσεις για μελλοντική επέκταση του συστήματός μας.

Τέλος, στο όγδοο κεφάλαιο παρουσιάζεται η βιβλιογραφία που χρησιμοποιήθηκε για την εκπόνηση της εργασίας.

2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό θα γίνει μία περιγραφή διαφόρων εννοιών που σχετίζονται με την παρούσα εργασία.

2.1 Δίκτυα Ομότιμων Κόμβων (Peer to Peer - P2P)

2.1.1 Ανάγκη δημιουργίας P2P συστημάτων

Σε κατακευματμένα συστήματα μεγάλων διαστάσεων, όπως ο Παγκόσμιος Ιστός, η κλασική αρχιτεκτονική πελάτη-εξυπηρετητή (client-server) εμφανίζει σημαντικά προβλήματα συμφόρησης (bottleneck), λόγω του αυξημένου αριθμού πελατών που επιδιώκουν πρόσβαση στους πόρους των εξυπηρετητών σε αποδεκτό χρόνο απόκρισης, καθώς και λόγω του περιορισμένου εύρους ζώνης. Το πρόβλημα αυτό αποτέλεσε κίνητρο για τη δημιουργία κατακευματμένων πληροφοριακών συστημάτων κόμβων, όπως τα συστήματα ομότιμων κόμβων (P2P Systems).

2.1.2 Περιγραφή P2P συστημάτων

Ένα Peer-to-Peer σύστημα είναι ένα ανοικτό δίκτυο κατανεμημένων υπολογιστικών ή ομότιμων κόμβων, οι οποίοι μπορούν να ανταλλάσσουν δεδομένα και/ή υπηρεσίες με άλλους κόμβους. Οι συνιστώντες κόμβοι του συστήματος αυτού συμμετέχουν σε παρόμοιους ρόλους, έχοντας ίδιες δυνατότητες και δικαιώματα, και για το λόγο αυτό αναφέρονται ως ομότιμοι μεταξύ τους κόμβοι.

Σε ένα P2P δίκτυο, οι κόμβοι έχουν ίδια λειτουργικότητα για να παρέχουν και να δέχονται δεδομένα και υπηρεσίες. Κάθε κόμβος μπορεί να ανταλλάσσει πληροφορίες με ένα σύνολο από άλλους κόμβους, που ονομάζονται γείτονες ή γνωστοί (acquaintances). Οι κόμβοι είναι πλήρως αυτόνομοι στην επιλογή των γειτόνων τους [BGK+02]. Παρόλο που οι κόμβοι έχουν παρόμοιους ρόλους, μπορεί να υπάρχει κάποια δομή στο P2P σύστημα, το οποίο έχει συνήθως κάποιο βαθμό αυτοοργάνωσης, όπου κάθε κόμβος βρίσκει τους ομότιμους του και βοηθάει στη διατήρηση της δομής του συστήματος. Επιπλέον, σε περιβάλλον «καθαρών» P2P, οι κόμβοι είναι σε μεγάλο βαθμό ανεξάρτητοι ο ένας από τον άλλο, και δεν υπάρχει καθολικός (global) έλεγχος υπό τη μορφή καθολικής καταχώρησης, ή διαχείρισης καθολικών πηγών, ούτε καθολικό σχήμα ή αποθήκη δεδομένων [Zai03].

Τα P2P μπορούν να θεωρηθούν μία αποκεντρωμένη αρχιτεκτονική δικτύου. Οι συμμετέχοντες κόμβοι βασίζονται ο ένας στον άλλο, και όχι σε μία καθιερωμένη και κεντρική υποδομή, για να λάβουν υπηρεσίες. Σε αντίθεση με την αρχιτεκτονική πελάτη-εξυπηρετητή (client-server), η οποία υποδηλώνει μία έντονη διάκριση ανάμεσα στους πελάτες που απαιτούν και δέχονται υπηρεσίες, και στους εξυπηρετητές που τις παρέχουν, όλοι οι κόμβοι ενός P2P δικτύου μπορούν να αναλάβουν και τους δύο ρόλους. Η ιδιότητα μέλους ενός P2P συστήματος είναι σχετικά απρόβλεπτη: οι υπηρεσίες παρέχονται από κόμβους που τυχαίνει να συμμετέχουν την κάθε δεδομένη στιγμή. Ένα μεγάλης διάρκειας δόγμα των κατανεμημένων συστημάτων είναι ότι η δύναμη ενός κατανεμημένου συστήματος μπορεί να μεγαλώσει καθώς περισσότεροι χρήστες συμμετέχουν σ' αυτό. Κάθε συμμετέχων μπορεί να συνεισφέρει δεδομένα και υπολογιστικές πηγές (όπως αχρησιμοποίητοι κύκλοι CPU και αποθηκευτικός χώρος) στο συνολικό σύστημα, και ο πλούτος της κοινωνίας μπορεί να κλιμακωθεί με τον αριθμό των συμμετεχόντων [GHI01].

2.1.3 Πλεονεκτήματα και μειονεκτήματα

Το μεγάλο όραμα — μία αποκεντρωμένη κοινωνία από μηχανές που συγκεντρώνουν και οργανώνουν τις πηγές τους για το όφελος όλων — ωθείται για πολλούς λόγους: προσαρμοστικότητα, ευρωστία, έλλειψη ανάγκης για διαχείριση, αντοχή σε σφάλματα και

ακόμα ανωνυμία και αντοχή στη λογοκρισία [GHI01]. Οι τρεις αρχές στις οποίες βασίζονται τα P2P συστήματα κάνουν φανερά τα πλεονεκτήματα. Η αρχή του μοιράσματος πόρων, όπως αποθηκευτικού χώρου, υπολογιστικής δύναμης, εύρους ζώνης, υπηρεσιών και δεδομένων, μεταξύ των κόμβων κάνει το δίκτυο πιο ευέλικτο και αποδοτικό. Η αρχή της αποκέντρωσης δίνει λύση στο πρόβλημα της συμφόρησης και τέλος η αρχή της αυτοοργάνωσης δίνει τη δυνατότητα πιο ευέλικτης οργάνωσης και επέκτασης του δικτύου.

Ωστόσο, η ad hoc φύση ενός P2P δικτύου, η υψηλού επιπέδου αυτονομία των κόμβων και η απουσία καθολικού συγκεντρωτισμού, που αποτελούν τις αιτίες των πλεονεκτημάτων του P2P, οδηγούν επίσης σε μερικά μειονεκτήματα. Το κύριο μειονέκτημα σχετίζεται με το πρόβλημα της ανακάλυψης πηγών (resources discovery) σε ένα P2P δίκτυο. Ένα P2P δίκτυο δεν έχει μηχανισμούς συγκέντρωσης ή συντονισμού στη διάδοση της απαίτησης για ανακάλυψη πηγών. Σε ένα τέτοιο περιβάλλον η απαίτηση διαδίδεται σε βάση «στέλνω-σε-όλους τους γείτονες» (“sendto-all-acquaintances”), επομένως η διάδοση παίρνει αλυσιδωτή μορφή, περιλαμβάνοντας όλο και περισσότερους κόμβους σε κάθε επακόλουθη διάδοση. Αυτό, ασφαλώς, στις περισσότερες περιπτώσεις οδηγεί σε ένα μεγάλο ποσοστό από άσχετα αποτελέσματα και μπορεί να οδηγήσει σε υπερφόρτωση δικτύου. Δεν υπάρχει εγγύηση για την ποιότητα της υπηρεσίας (Quality of the Service) (QoS) που παρέχεται από το P2P δίκτυο. Ένα άλλο πρόβλημα είναι το πρόβλημα των μεταδεδομένων (meta-data), δηλαδή της εμβέλειας, της διατήρησης, της διαχείρισης και της απαρχαίωσης τους. Για να γίνει η χρήση των πηγών του δικτύου αποδοτική, χρειάζεται να παραχθούν και να διατηρηθούν πολλά μεταδεδομένα. Λόγω της ισχυρής δυναμικής των P2P δικτύων αυτό είναι ένα κρίσιμο και δύσκολο ζήτημα. Όσο υψηλότερο είναι το επίπεδο διαχείρισης των μεταδεδομένων στο δίκτυο τόσο υψηλότερο είναι το QoS και τόσο δυσκολότερη είναι η διατήρησή τους. Αντίθετα, όσο πιο κακώς ορισμένα είναι τα μεταδεδομένα, τόσο ευκολότερη είναι η διατήρησή τους, αλλά το QoS θα πληγεί σοβαρά. Πρέπει ασφαλώς να υπάρχει μία βέλτιστη σχέση ανάμεσα στις προσπάθειες που γίνονται στη διαχείριση των μεταδεδομένων και στην απόδοση του δικτύου. [Zai03]

2.1.4 Εφαρμογές

Τα P2P έλαβαν την προσοχή της βιομηχανίας και της ακαδημαϊκής κοινότητας ως μία έκδοση κατανεμημένου υπολογισμού που βρίσκεται μεταξύ των παραδοσιακών κατανεμημένων συστημάτων και του διαδικτύου. Τα υπάρχοντα P2P συστήματα έχουν επικεντρωθεί σε συγκεκριμένες περιοχές εφαρμογών σε συστήματα διαμοιράσματος αρχείων, σε κατανεμημένους υπολογισμούς, σε συστήματα αμέσων μηνυμάτων, σε συστήματα διαμοιράσματος αρχείων, στην παροχή δυνατοτήτων όπως των συστημάτων αρχείων κα.

Αυτά τα συστήματα αγνοούν τη σημασιολογία των δεδομένων. Χαρακτηριστικά παραδείγματα αποτελούν συστήματα ανταλλαγής αρχείων, όπως τα Napster και Gnutella. Το πρώτο είναι πλούσιο σε υπηρεσίες αλλά απαιτεί σημαντικές λειτουργικές δαπάνες για να τεθεί σε κυκλοφορία και έχει μία σχετικά στατική, ελεγχόμενη αρχιτεκτονική. Το δεύτερο είναι μία δυναμική, «καθένας-προς-καθένα» (anyone-to-anyone) αρχιτεκτονική με μικρό αρχικό κόστος αλλά με περιορισμένες υπηρεσίες. Γενικά, τα P2P προσφέρουν μία εξελισσόμενη αρχιτεκτονική όπου οι κόμβοι έρχονται και φεύγουν, επιλέγουν με ποιον θα ασχοληθούν, και απολαμβάνουν μερικές παραδοσιακά καταναμημένες υπηρεσίες με λιγότερο αρχικό κόστος[BGK+02].

Η κοινότητα των βάσεων δεδομένων ερευνά τα P2P ως ένα νέο παράδειγμα για καταναμημένη διαχείριση δεδομένων, που δίνει ισχυρό πλεονέκτημα στα υπάρχοντα συστήματα βάσεων δεδομένων, και επίσης δημιουργεί προκλήσεις για σχολαστική έρευνα στην περιοχή αυτή.

Πολλά παραδείγματα από P2P συστήματα έχουν προκύψει πρόσφατα, τα περισσότερα από τα οποία είναι συστήματα ευρείας περιοχής και μεγάλης κλίμακας που παρέχουν μοίρασμα περιεχομένων, υπηρεσίες αποθήκευσης, ή υπολογισμούς καταναμημένου “πλέγματος”. Επίσης υπάρχουν P2P συστήματα μικρότερης κλίμακας, όπως federated, συστήματα αρχείων χωρίς εξυπηρετητή και συνεργατικά εργαλεία.

2.2 P2P και Βάσεις Δεδομένων

2.2.1 Είδη βάσεων δεδομένων

Θεωρούμε την γραμμή εξέλιξης των συστημάτων βάσεων δεδομένων: οι **παραδοσιακές** κεντρικές βάσεις δεδομένων ήταν ορισμένες και διαχειριζόμενες κεντρικά. Με την εμφάνιση των **καταναμημένων** βάσεων δεδομένων το επίπεδο επικέντρωσης μειώθηκε: η διαχείριση των βάσεων δεδομένων μπορεί να χειριστεί με ένα φυσικά καταναμημένο τρόπο πάνω από ένα δίκτυο υπολογιστών, παρόλο που είχε υποτεθεί ότι οι βάσεις δεδομένων ήταν ορισμένες τοπικά. Ακόμα, η κατανομή των δεδομένων πάνω στο δίκτυο ήταν σε μεγάλο βαθμό διαφανής για τους τελικούς χρήστες. Το παραδοσιακό σύστημα καταναμημένων βάσεων δεδομένων έχει ένα καθολικό σχήμα και το ίδιο DBMS για όλες τις συμπεριλαμβανόμενες βάσεις. Αν ένα στοιχείο τοπικής αυτονομίας εισαχθεί, τότε ένα τέτοιο σύστημα λέγεται συχνά **federated database system ή federation**. Αλλά επίσης, ένα federation έχει την έννοια του καθολικού ορισμένου σχήματος. Με την παρουσία ετερογένειας ανάμεσα στα σχήματα των περιλαμβανομένων βάσεων δεδομένων, το σύστημα γίνεται ένα **multi-database** σύστημα.

Είναι εύκολο να δούμε ότι η τάση είναι να μετακινηθούμε από τις κεντρικά ορισμένες και διαχειριζόμενες βάσεις προς ένα καταναμημένο σύστημα βάσεων δεδομένων, όπου κάθε βάση κυβερνάται τοπικά, αλλά η οποία μπορεί ακόμα να θεωρηθεί ως μία μονολιθική βάση που αντιπροσωπεύεται μέσω ενός καθολικού σχήματος. Το τελευταίο κάνει την εργασία με το σύστημα βάσεων δεδομένων διαφανή για τον τελικό χρήστη, αναφορικά με την υποκείμενη πολυπλοκότητα της κατανομής των βάσεων και των εξαρτήσεων μέσα στο σχήμα.

2.2.2 P2P δίκτυα βάσεων δεδομένων (P2P DB)

Τα P2P παραδείγματα εμφανίζονται πολύ συνεκτικά με αυτή τη γραμμή εξέλιξης των βάσεων δεδομένων. Τα Peer-to-Peer δίκτυα βάσεων δεδομένων (P2P DB) μπορεί να είναι μία καλή επιλογή για τις επόμενες γενιές συστημάτων βάσεων δεδομένων. Οι βάσεις δεδομένων μπορούν να βρίσκονται και να **διοικούνται τοπικά** στους κόμβους και μπορούν να έχουν μεγάλη **αυτονομία** αναφορικά με το περιβάλλον των βάσεων σε άλλους κόμβους. Αυτονομία σημαίνει ότι οι κόμβοι αποφασίζουν μόνοι τους πως θα αναπτύξουν τις βάσεις τους, τι DBMS θα χρησιμοποιήσουν, πως θα αποθηκεύουν δεδομένα, κτλ [Zai03].

Αλλά από την άλλη, για να είναι συνεπείς με τις ιδέες ολοκλήρωσης δεδομένων, οι οποίες αξιοποιούνται σημαντικά στα σύγχρονα συστήματα βάσεων δεδομένων, οι κόμβοι των βάσεων πρέπει να έχουν καλά αναπτυγμένους μηχανισμούς για να συντονίζουν τις βάσεις τους όταν απαντούν ερωτήματα ή εκτελούν ενημερώσεις. Αυτό αφορά περισσότερο την αποτίμηση και διάδοση των ερωτημάτων, αφού η απάντηση ερωτήσεων είναι η θεμελιώδης λειτουργία που απαιτείται από ένα σύστημα βάσεων δεδομένων. Ιδανικά, η εργασία του τελικού χρήστη ενός P2P DB δικτύου δεν πρέπει να είναι διαφορετική από αυτή ενός συστήματος πολλών βάσεων δεδομένων, για παράδειγμα. Αυτό σημαίνει ότι ο αριθμός των κόμβων, ο τρόπος που τα δεδομένα δεσμεύονται σ' αυτούς τους κόμβους και οι σημασιολογικές διαεξαρτήσεις μεταξύ τους πρέπει να είναι άσχετες από το χρήστη.

Ωστόσο τρεις θεμελιώδεις διαφορές μπορούν να διακριθούν στη **διαδικασία απάντησης ερωτημάτων (query answering)** [Zai03]:

- a) όσο το P2P δίκτυο εξελίσσεται, το ίδιο ερώτημα μπορεί να φέρει διαφορετικά αποτελέσματα αν τεθεί σε διαφορετική χρονική στιγμή,
- b) ο χρόνος απόκρισης του ερωτήματος μπορεί να είναι περισσότερος και τα αποτελέσματα μπορεί να «ξεχύνονται» συνεχώς καθώς φτάνει σε περισσότερους κόμβους, και

- c) τα αποτελέσματα μπορεί να μην είναι τόσο σχετικά λόγω του ότι η ποιότητα των συνδέσεων μεταξύ των κόμβων μπορεί να ποικίλει πολύ.

2.2.3 Συστήματα διαχείρισης δεδομένων Peers (*Peer Data Management Systems - PDMSs*)

Μέσα στην ποικιλία των P2P εφαρμογών που έχουν προταθεί, τα συστήματα διαχείρισης δεδομένων των peers (*Peer Data Management Systems - PDMSs*) παίζουν σημαντικό ρόλο στο διαμοιρασμό σημασιολογικά πλούσιας πληροφορίας. Ο στόχος των PDMS είναι η χρήση μίας αποκεντρωμένης, εύκολα επεκτάσιμης αρχιτεκτονικής διαχείρισης δεδομένων στην οποία κάθε χρήστης μπορεί να συνεισφέρει καινούρια δεδομένα, σχήματα πληροφορίας, ή ακόμα και αντιστοιχίσεις μεταξύ σχημάτων άλλων κόμβων. Τα PDMS παρουσιάζουν ένα φυσικό βήμα πέρα από τα συστήματα ολοκλήρωσης δεδομένων, αντικαθιστώντας το μοναδικό λογικό σχήμα τους με μία διασυνδεδεμένη συλλογή από **σημασιολογικές αντιστοιχίσεις (semantic mappings)** μεταξύ των σχημάτων ξεχωριστών κόμβων. Οι αντιστοιχίσεις αυτές συνδέουν στοιχεία (ονόματα πινάκων και ιδιοτήτων) ενός σχήματος με στοιχεία του άλλου.

Σε ένα PDMS, κάθε κόμβος είναι μία αυτόνομη πηγή που έχει ένα τοπικό σχήμα (*local schema*), διαφορετικό από των άλλων. Οι πηγές αποθηκεύουν και διαχειρίζονται τα δεδομένα τους τοπικά, αποκαλύπτοντας μόνο τμήμα από τα σχήματα τους στους υπόλοιπους κόμβους. Λόγω της έλλειψης καθολικού σχήματος (*global schema*), εκφράζουν και απαντούν ερωτήματα βασιζόμενοι στο τοπικό σχήμα τους. Επίσης, οι κόμβοι παρουσιάζουν τοπική συνεργασία με τους γείτονες τους στη διάταξη.

Κατά τη διάρκεια της **διαδικασίας της γνωριμίας**, οι δύο κόμβοι ανταλλάσσουν πληροφορίες για τα τοπικά του σχήματα και δημιουργούν ενδιάμεσα σχήματα ημιαυτόματα. Η εγκαθίδρυση της γνωριμίας συνεπάγεται μία συμφωνία για την λειτουργία της συνεργασίας των δεδομένων ανάμεσα στους γείτονες, η οποία βασίζεται στο αντίστοιχο **σχήμα αντιστοίχισης (schema mapping)**. Ωστόσο, οι κόμβοι δεν πρέπει να συμμορφωθούν σε κανενός είδους μετασχηματισμούς δεδομένων ή σχήματος για να αποκτήσουν γείτονες και να συμμετέχουν στο σύστημα.

Για την επικοινωνία των κόμβων απαιτείται **μετάφραση των ερωτημάτων (query translation)**, αφού αυτά τίθενται με βάση το σχήμα της βάσης του κόμβου που τα θέτει, κι έτσι δεν μπορούν να απαντηθούν από άλλους κόμβους που έχουν διαφορετικό σχήμα. Η σαφής **διαδικασία επεξεργασίας ερωτημάτων (query processing)**.

Σε τέτοια συστήματα, για να καταστεί δυνατός ο αποδοτικός διαμοιρασμός πληροφορίας μεταξύ ετερογενών πηγών, εφαρμόζονται οι ιδιότητες των **κοινωνικών δικτύων (social networks)**: όπως οι άνθρωποι, οι κόμβοι προσπαθούν να αναγνωρίσουν άλλους συμμετέχοντες στη διάταξη με ενδιαφέροντα που ταιριάζουν με τα δικά τους. Τα υπολογιστικά κοινωνικά δίκτυα (computing social networks) αποτελούνται από ένα δίκτυο από διασυνδεδεμένους κόμβους (peers). Αρχικά κάθε κόμβος είναι συνδεδεμένος σε ένα τυχαίο υποσύνολο από κόμβους. Σταδιακά, οι κόμβοι γνωρίζονται μεταξύ τους, με τις νέες συνδέσεις να υποδηλώνουν σημασιολογική συγγένεια.

Στα συστήματα που περιγράψαμε μέχρι τώρα (**συστήματα διαμοιρασμού δεδομένων ή data sharing systems**), η μετάφραση των ερωτημάτων γινόταν μέσω των αντιστοιχίσεων μεταξύ των σχημάτων των κόμβων και χωρίς την παρουσία ενδιάμεσου σχήματος. Έχοντας ένα οργανισμό κοινωνικού δικτύου σε ένα PDMS, ένα ενδιαφέρον ερώτημα είναι πώς να δημιουργήσουμε αυτόματα μία σύνοψη για τα κοινά ενδιαφέροντα μιας ομάδας από σημασιολογικά συσχετιζόμενους κόμβους. Αυτό γίνεται στα **συστήματα ολοκλήρωσης δεδομένων (data integration systems)**, με χρήση ενός **ενδιάμεσου σχήματος (mediated schema)** που αντιπροσωπεύει την ομάδα μέσω των αντιστοιχίσεων (mappings) του με τις τοπικές βάσεις δεδομένων. Τα ερωτήματα μπορούν έπειτα να εκφραστούν πάνω σε αυτό το ενδιάμεσο σχήμα και στη συνέχεια να μεταφραστούν στο σχήμα των κόμβων που θα τα απαντήσουν. Αυτή η λειτουργία είναι επιθυμητή για πολλούς λόγους [KTS07]:

- a) επιτρέπει στα ερωτήματα να θέτονται σε ένα και έγκυρο σχήμα,
- b) επιταχύνει τη γνωριμία μεταξύ σημασιολογικά σχετικών κόμβων, και
- c) ελαχιστοποιεί την ανθρώπινη ανάμιξη στη διαδικασία του σχηματισμού/ενημέρωσης του ομαδικού σχήματος.

Πολλές από τις έννοιες που αναφέρθηκαν εδώ, θα εξεταστούν αναλυτικότερα στις επόμενες παραγράφους.

2.2.4 Τοπικό σχεσιακό μοντέλο (Local Relational Model - LRM)

Στα P2P συστήματα χρησιμοποιείται το Local Relational Model (LRM), ένα μοντέλο δεδομένων ειδικά κατασκευασμένο για P2P εφαρμογές. Το LRM ορίζει ένα τυπικό τρόπο, που βασίζεται σε λογική πρώτης τάξης, για να περιγράψουμε ένα σύνολο από βάσεις δεδομένων σε ένα P2P δίκτυο και ένα σύνολο από φόρμουλες συντονισμού που είναι ελαφρείς περιορισμοί μέσα στις βάσεις [Zai03]. Το LRM υποθέτει ότι το σύνολο όλων των δεδομένων στο P2P δίκτυο αποτελείται από τοπικές (σχεσιακές) βάσεις δεδομένων, καθεμία με ένα σύνολο γειτόνων, το οποίο ορίζει την τοπολογία του P2P δικτύου. Για κάθε

γνωστοποιημένο σύνδεσμο, οι σχέσεις περιοχής (domain relations) ορίζουν κανόνες μετάφρασης μεταξύ των τεμαχίων των δεδομένων, και οι φόρμουλες συντονισμού ορίζουν σημασιολογικές εξαρτήσεις μεταξύ των δύο βάσεων. Οι κύριοι σκοποί του μοντέλου δεδομένων είναι να παίρνει τα μέτρα του για ασυνεπείς βάσεις και να υποστηρίζει σημασιολογική διαλειτουργικότητα (δηλαδή, ικανότητα να ανταλλάσσει και να χρησιμοποιεί πληροφορίες) στην απουσία καθολικού σχήματος [BGK+02].

2.3 Σημασιολογική Ολοκλήρωση Δεδομένων (Semantic Data Integration)

2.3.1 Εισαγωγή

Λόγω της ευρείας εξάπλωσης και χρήσης του διαδικτύου, διατίθεται διαμέσου του ένας αυξανόμενος αριθμός δεδομένων, από την προσπάθεια να καλυφθούν οι πληροφοριακές ανάγκες ποικίλων ομάδων χρηστών. Έτσι, έχουν αναπτυχθεί πληροφοριακά συστήματα, τα οποία χαρακτηρίζονται από μεγάλο βαθμό αυτονομίας σε ποικίλα επίπεδα, όπως οι διαφορετικές δυνατότητες αναζήτησης-ανάκτησης δεδομένων σε κάθε ένα από αυτά. Η αυτονομία στο σχεδιασμό των πληροφοριακών συστημάτων οδηγεί στην εμφάνιση **ετερογένειας** σε τέσσερα διαφορετικά επίπεδα [ΜΓ06]:

1. ετερογένεια συστημάτων (system heterogeneity)
2. ετερογένεια στη σύνταξη (syntactic heterogeneity), λόγω χρήσης διαφορετικών μοντέλων ή γλωσσών
3. ετερογένεια σχημάτων (schematic heterogeneity), λόγω δομικών διαφορών
4. σημασιολογική ετερογένεια (semantic heterogeneity), λόγω των διαφορετικών τρόπων έκφρασης και των ποικίλων ερμηνειών που μπορούν να δοθούν στη σημασία των δεδομένων

Αποτέλεσμα των παραπάνω είναι η ενιαία αναζήτηση και ανάκτηση δεδομένων να γίνεται εξαιρετικά δύσκολα από τους χρήστες, οι οποίοι δεν αναγνωρίζουν τις διαφορές και τα επίπεδα ετερογένειας των πληροφοριακών συστημάτων. Αντίθετα, ενδιαφέρονται για «ενοποιημένες διαδρομές» αναζήτησης και ανάκτησης σε ποικίλες πηγές με στόχο να καλύψουν τις πληροφοριακές τους ανάγκες.

Τις ανάγκες αυτές έρχονται να καλύψουν τα **συστήματα ολοκλήρωσης δεδομένων (data integration systems)**, παρέχοντας στο χρήστη πρόσβαση σε συλλογή αυτόνομων πηγών, σαν αυτές να αποτελούν ως σύνολο μία πηγή δεδομένων. Σ' αυτά τα συστήματα υπάρχει ένα

σχήμα διαμεσολάβησης (mediated schema), ή αλλιώς **καθολικό σχήμα (global schema)**, στο οποίο ο χρήστης υποβάλλει το ερώτημα. Οι σχέσεις ανάμεσα στο καθολικό σχήμα και στις τοπικές πηγές εκφράζονται μέσα από όψεις (views).

Τα βασικά θέματα που πρέπει να αντιμετωπιστούν σε ένα σύστημα ολοκλήρωσης δεδομένων είναι:

1. ο τρόπος που ορίζουμε τις σημασιολογικά σχετικές πληροφορίες των πηγών με το καθολικό σχήμα, και
2. η διαδικασία εκτέλεσης ενός ερωτήματος και ο μετασχηματισμός του σε επιμέρους ερωτήματα προς κάθε τοπική πηγή.

Για να αντιμετωπιστούν τα προβλήματα σημασιολογικής ετερογένειας ανάμεσα στα τοπικά δεδομένα και το καθολικό σχήμα, ορίζονται **κανόνες αντιστοίχισης (mapping rules)**. Οι κανόνες αντιστοίχισης ορίζουν μία σχέση ανάμεσα σε δύο μοντέλα· το πρώτο είναι το πηγαίο σχήμα (source schema), π.χ. μία τοπική πηγή δεδομένων, και το δεύτερο είναι το τελικό σχήμα (target schema), π.χ. το σχήμα διαμεσολάβησης. Πιο αναλυτικά, διατηρούν τη σημασιολογική σχέση ανάμεσα στις πιθανές ερμηνείες των πεδίων των δύο μοντέλων.

Αναφορικά με τη διαδικασία της ερώτησης, ένα ερώτημα υποβάλλεται ως προς το καθολικό σχήμα και στη συνέχεια μετασχηματίζεται σε ερωτήματα στις επιμέρους πηγές (**μετασχηματισμός/μετάφραση ερωτήματος – query translation**). Στη διαδικασία αυτή αξιοποιούνται οι κανόνες αντιστοίχισης, οι οποίοι όσο πιο πολύπλοκοι είναι στην έκφρασή τους τόσο πιο δύσκολη γίνεται η διαδικασία του μετασχηματισμού των ερωτημάτων. Επιπλέον ζήτημα στη **διαδικασία επεξεργασίας της ερώτησης (query processing)** είναι η μετατροπή του ερωτήματος από μία γλώσσα επερώτησης σε μία άλλη, σε περίπτωση που οι τοπικές πηγές είναι σε διαφορετική σύνταξη και κωδικοποίηση σε σχέση με το καθολικό σχήμα. Έτσι, το πρόβλημα της **αναδιατύπωσης ενός ερωτήματος (query reformulating)** ενός χρήστη, που έχει τεθεί πάνω στο ενδιάμεσο σχήμα, σε ένα ερώτημα που αναφέρεται άμεσα στα σχήματα πηγών μετατρέπεται σε πρόβλημα **απάντησης των ερωτημάτων (query answering)** χρησιμοποιώντας όψεις.

Συνεπώς, το πρόβλημα της σύγκρουσης σχήματος σε ένα P2P δίκτυο μπορεί να λυθεί μεταφράζοντας δεδομένα από την τοπική αναπαράσταση στην καθολική αναφορά. Οι κόμβοι απαντούν διαφορετικά ερωτήματα ανάλογα με το σχήμα τους και τον τρόπο που παρουσιάζονται τα δεδομένα. Όπως σε όλα τα συστήματα ολοκλήρωσης, ένα σύστημα P2P παίρνει επίσης προσεγγιστικές απαντήσεις στο ερώτημα του χρήστη. Οι κόμβοι μπορούν να απαντήσουν το ίδιο ερώτημα ακόμα κι αν τα δεδομένα είναι αποθηκευμένα σε διαφορετικές μορφές. Οι διαφορές στον τύπο των δεδομένων δεν έχουν μεγάλο αντίκτυπο στην αποστολή της επεξεργασίας ερωτημάτων στους κόμβους. Αλλά διαφορές στον τύπο των δεδομένων μπορεί να επιστρέψουν αποτελέσματα σε διαφορετικές μορφές. Η επίδοση του συστήματος

μπορεί να βελτιωθεί, ορίζοντας το είδος των ερωτημάτων που ένας κόμβος είναι ικανός να απαντήσει, επειδή ένα ερώτημα δε θα σταλεί σε ένα κόμβο αν δεν υπάρχει ένδειξη ότι δεν θα είναι δυνατό να απαντηθεί. Αυτό μειώνει αποφεύξιμες συνδέσεις σε κόμβους.

Η παραδοσιακή Ολοκλήρωση Δεδομένων αποτελεί μία ανοιχτή ερευνητική περιοχή στο πεδίο των Βάσεων Δεδομένων, αλλά πρόσφατα, το ερευνητικό ενδιαφέρον μεταφέρεται από την ολοκλήρωση δεδομένων στη **σημασιολογική ολοκλήρωση δεδομένων** σε ποικίλες ερευνητικές περιοχές, όπως είναι η διαχείριση της πληροφορίας (information management) και οι οντολογίες (ontologies). Η συγκεκριμένη τάση επηρεάζεται άμεσα από τη πρόθεση «μετασχηματισμού» του καθολικού ιστού σε πλέον πλούσιες σημασιολογικές φόρμες με τη χρήση τεχνολογιών του Σημασιολογικού Ιστού (Semantic Web). Η σημασιολογική ολοκλήρωση δεδομένων καλείται να αντιμετωπίσει προβλήματα σημασιολογικής ετερογένειας (semantic heterogeneity) τα οποία προκύπτουν από τη χρήση διαφορετικών όρων για την αποτύπωση της ίδιας έννοιας τόσο σε επίπεδο σχημάτων κωδικοποίησης δεδομένων (schema level) όσο και στο επίπεδο των δεδομένων (data level). Στο επίπεδο του σχήματος, τα δεδομένα μπορούν να διαφέρουν στις σχέσεις, στα ονόματα πεδίων και ιδιοτήτων, στο επίπεδο λεπτομέρειας και στην κάλυψη ενός συγκεκριμένου τομέα. Στο επίπεδο των δεδομένων, συναντούνται διαφορετικές παρουσιάσεις της ίδιας έννοιας, όπου οι διαφορές αυτές αντιμετωπίζονται με ποικίλες μεθόδους, όπως η δημιουργία συνδέσμων μεταξύ τους (record linkage).

2.3.2 Συστήματα ολοκλήρωσης δεδομένων (Data integration systems)

Το επιστημονικό πεδίο της Ολοκλήρωσης Δεδομένων ασχολείται με το πρόβλημα του συνδυασμού δεδομένων που προέρχονται από διαφορετικές πηγές, διαφορετικά συστήματα, περιγράφονται και μοντελοποιούνται από διαφορετικά σχήματα, και την ενοποίησή τους σε ένα ομογενοποιημένο σύνολο αποτελεσμάτων, διευκολύνοντας ταυτόχρονα την πρόσβαση του χρήστη, ο οποίος διατυπώνει ερωτήματα στα ενδιάμεσα δεδομένα σαν να διατύπωνε ερώτημα σε μία βάση δεδομένων. Τα συστήματα ολοκλήρωσης δεδομένων υποστηρίζουν πλούσια ερωτήματα πάνω σε ένα μεγάλο αριθμό από αυτόνομες, ετερογενείς πηγές δεδομένων εκμεταλλευόμενα τις σημασιολογικές σχέσεις μεταξύ των διαφορετικών σχημάτων των πηγών. Το πλεονέκτημα της υλοποίησης συστημάτων ολοκλήρωσης δεδομένων είναι ότι τα δεδομένα μπορούν να διαχειρίζονται και να ενημερώνονται τοπικά αλλά να είναι διαμοιρασμένα [NB05].

Υπάρχουν διαφορές και ομοιότητες ανάμεσα στο πρόβλημα ανταλλαγής δεδομένων και την ολοκλήρωση δεδομένων. Η διαφορά είναι ότι, στην ολοκλήρωση δεδομένων, τα δεδομένα δεν μετακινούνται από το ένα σχήμα στο άλλο, αλλά τα ερωτήματα διατυπώνονται σαν να

ήταν μία βάση. Στην ανταλλαγή δεδομένων, τα δεδομένα μετακινούνται από το σχήμα πηγή στο σχήμα στόχος. Μία από τις ομοιότητες είναι ότι και τα δύο χρειάζονται αντιστοίχιση σχημάτων (schema mapping) για να συσχετίσουν δεδομένα από το ένα σχήμα στο άλλο.

2.3.3 Ενδιάμεσο / καθολικό σχήμα (Mediated / global schema)

Όπως αναφέραμε παραπάνω, στα συστήματα ολοκλήρωσης δεδομένων υπάρχει ένα **σχήμα διαμεσολάβησης (mediated schema)**, ή αλλιώς **καθολικό σχήμα (global schema)**, στο οποίο ο χρήστης υποβάλλει το ερώτημα και ένα σύνολο από **τοπικές πηγές (local sources)**. Οι πηγές περιέχουν τα πραγματικά δεδομένα, ενώ το καθολικό σχήμα παρέχει μία ενοποιημένη και εικονική όψη των πηγών. Ένας διαχειριστής ορίζει το καθολικό ενδιάμεσο σχήμα για την περιοχή εφαρμογής και καθορίζει σημασιολογικές αντιστοιχίσεις ανάμεσα στις τοπικές πηγές και το ενδιάμεσο σχήμα. Το σχήμα αυτό δεν περιλαμβάνει δεδομένα από τις τοπικές πηγές αποθηκευμένα σε αυτό, αλλά αποτελεί επί της ουσίας έναν εικονικό πυρήνα έκφρασης των δεδομένων που αυτές διαθέτουν. Οι σχέσεις ανάμεσα στο καθολικό σχήμα και στις τοπικές πηγές εκφράζονται μέσα από όψεις (views).

Το κρίσιμο μειονέκτημα των συστημάτων αυτών είναι ότι βασίζονται σε ένα καθολικό σχήμα, όπου στα καθαρά P2P περιβάλλοντα η υπόθεση της ύπαρξης ενός καθολικού σχήματος δεν μπορεί να γίνει αποδεκτή. Παρόλα αυτά, οι φορμαλισμοί ολοκλήρωσης δεδομένων και οι σχετικές τεχνικές επεξεργασίας ερωτημάτων μπορεί να ξαναχρησιμοποιηθούν για να εκφράσουν διαεξαρτήσεις (interdependencies) μεταξύ δύο κόμβων ή μεταξύ ενός κόμβου και ενός (μικρού) υποσυνόλου κόμβων καθώς και για τη μετάφραση ερωτημάτων μεταξύ τους.

Θεωρούμε τη «δυνατή» σημασιολογία που απαιτείται από πολλές εφαρμογές, και ότι οι πηγές δεδομένων μπορούν να εξελιχθούν ανεξάρτητα και σχετικά ευέλικτα. Επίσης, στην πραγματικότητα, το ενδιάμεσο σχήμα, το ολοκληρωμένο (integrated) μέρος του συστήματος που πράγματι διευκολύνει τη διαμοιρασμό πληροφοριών, αποτελεί ανασχετικό παράγοντα για τη διαδικασία. Ο σχεδιασμός του ενδιάμεσου σχήματος πρέπει να γίνεται προσεκτικά και καθολικά: οι πηγές των βάσεων δεν μπορεί να αλλάζουν σημαντικά, διαφορετικά μπορεί να παραβιάσουν τις αντιστοιχίσεων στο ενδιάμεσο σχήμα, και έννοιες μπορούν να προστίθενται μόνο από τον κεντρικό διαχειριστή. Η ad hoc επεκτασιμότητα του διαδικτύου λείπει, και σαν αποτέλεσμα πολλές φυσικές, μικρής κλίμακας, εργασίες διαμοιρασμού πληροφοριών είναι δύσκολο να επιτευχθούν.

2.3.4 Αντιστοίχιση σχημάτων (*Schema mapping*)

Το πρόβλημα του διαμοιρασμού προϋπαρχόντων ετερογενών δεδομένων είναι εφαρμόσιμο σε πολλές περιοχές των βάσεων δεδομένων. Αυτές είναι η ολοκλήρωση πληροφοριών, η διαχείριση P2P δεδομένων, η ανταλλαγή δεδομένων και η αποθήκευση δεδομένων. Όλα αυτά τα προβλήματα χρειάζονται **αντιστοίχιση σχημάτων (schema mapping)**. Αντιστοίχιση σχημάτων είναι το πρόβλημα περιγραφής της σχέσης μεταξύ των σχημάτων βάσεων δεδομένων και περιγράφει μόνο τη σχέση μεταξύ σχημάτων χωρίς να θεωρεί δομικές συγκρούσεις και συγκρούσεις αναπαράστασης. Το πρόβλημα της αντιστοίχισης σχημάτων είναι εφαρμόσιμο όταν διατυπώνονται ερωτήματα από διαφορετικές πηγές. Η παραδοσιακή προσέγγιση είναι να αρχίζουμε προσδιορίζοντας τις αντιστοιχίσεις μεταξύ των σχημάτων των πηγών δεδομένων. Μία αρχιτεκτονική προτείνει χρήση XML για την αποτύπωση σχημάτων βάσεων δεδομένων από διαφορετικές πηγές. Σ' αυτή την αρχιτεκτονική όλα τα τοπικά σχήματα μετατρέπονται σε XML και το καθολικό σχήμα σχηματίζεται ενοποιώντας τοπικά σχήματα.

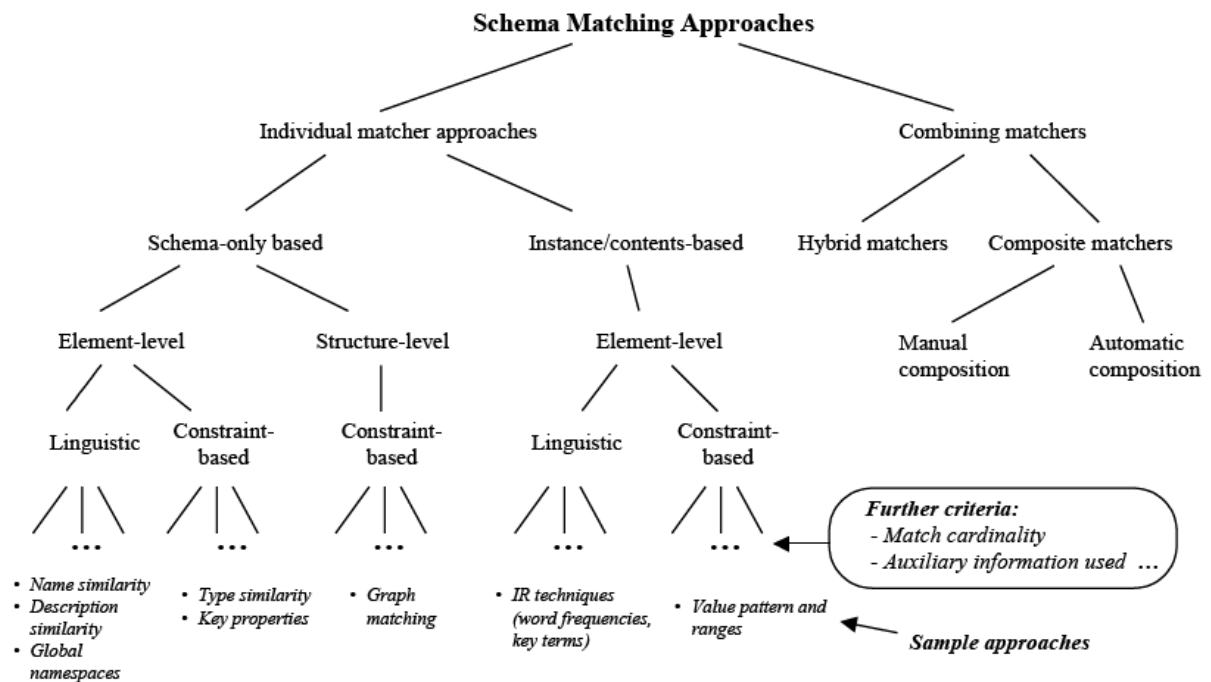
Το πρόβλημα που αντιμετωπίζουμε όταν ολοκληρώνουμε (ενοποιούμε) δεδομένα από διαφορετικές πηγές είναι ότι οι πηγές έχουν διαφορετικά μοντέλα δεδομένων, γλώσσες ερωτημάτων και ετερογένεια σχημάτων βάσεων δεδομένων. Η παραδοσιακή προσέγγιση είναι να έχουμε **μεσολαβητή (mediator)** και **συστατικά περιτύλιξης (wrapper components)** [NB05]. Ο mediator ορίζει ένα κοινό μοντέλο δεδομένων μέσα στο οποίο μεταφράζονται όλα τα τοπικά σχήματα. Το πλεονέκτημα του κοινού μοντέλου δεδομένων είναι ότι είναι πολύ ευκολότερο να γράψουμε ερωτήματα χρησιμοποιώντας ένα μοντέλο. Ο μεταφραστής ερωτημάτων (query translator) είναι μέρος των συστατικών του mediator. Ο μεταφραστής ερωτημάτων είναι υπεύθυνος για την επεξεργασία ερωτημάτων τα οποία είναι έτοιμα να περάσουν σε διαφορετικές πηγές δεδομένων. Ένα ερώτημα είναι αποκρίσιμο από ένα mediator αν είναι αποκρίσιμο από τουλάχιστον μία πηγή από τις πηγές που συμμετέχουν στην σύνδεση. Ο μεταφραστής μεταφράζει όλα τα ερωτήματα χρησιμοποιώντας μία κοινή γλώσσα χειρισμού δεδομένων η οποία χρησιμοποιείται μαζί με το κοινό μοντέλο δεδομένων. Τα συστατικά του wrapper είναι ειδικευμένα σε κάποια πηγή. Είναι υπεύθυνα για την μετάφραση ερωτημάτων από το κοινό μοντέλο δεδομένων των mediators πίσω στη γλώσσα ερωτημάτων της πηγής στην οποία ανήκει ο wrapper. Τα συστατικά του mediator μπορούν να επεκταθούν για να περιλάβουν πολλά υποστοιχεία τα οποία βελτιστοποιούν την απόδοση της αρχιτεκτονικής.

2.3.5 Ταίριασμα σχημάτων (*Schema matching*)

Με δεδομένο ένα σύνολο από ανεξάρτητα αναπτυγμένα σχήματα, κατασκευάζεται μία καθολική όψη. Αφού τα σχήματα είναι ανεξάρτητα, έχουν συχνά διαφορετική δομή και

ονοματολογία. Αυτό μπορεί προφανώς να συμβεί όταν τα σχήματα είναι από διαφορετικές περιοχές της πραγματικής ζωής. Ωστόσο, συμβαίνει επίσης ακόμα κι αν μοντελοποιούν την ίδια περιοχή του πραγματικού κόσμου, αλλά επειδή αναπτύχθηκαν από διαφορετικούς ανθρώπους σε διαφορετικά πλαίσια της πραγματικής ζωής.

Ένα πρώτο βήμα στην ενοποίηση των σχημάτων είναι να ταυτοποιήσουμε και να χαρακτηρίσουμε αυτές τις σχέσεις μέσα στα σχήματα. Αυτή είναι μία διαδικασία **ταιριάσματος σχήματος (schema matching)**. Το ταίριασμα σχήματος σε ένα P2P δίκτυο μπορεί αν γίνει συσχετίζοντας όλα τα σχήματα στο καθολικό σχήμα και ορίζοντας κανόνες για τη μετατροπή δεδομένων από το περιεχόμενο του κόμβου στο καθολικό περιεχόμενο. Αφού έχουν ταυτοποιηθεί, μπορούν τα συσχετιζόμενα στοιχεία να ενοποιηθούν κάτω από ένα συνεκτικό, ολοκληρωμένο σχήμα ή όψη. Κατά τη διάρκεια αυτής της ολοκλήρωσης, ή μερικές φορές σαν ξεχωριστό βήμα, δημιουργούνται προγράμματα ή ερωτήματα που επιτρέπουν την μετάφραση δεδομένων από το πρωτότυπο σχήμα στην ολοκληρωμένη απεικόνιση. Μία παραλλαγή του προβλήματος ολοκλήρωσης σχήματος είναι να ολοκληρωθεί ένα ανεξάρτητα αναπτυγμένο σχήμα με ένα δοθέν εννοιολογικό σχήμα. Αυτό απαιτεί, πάλι, συμφιλίωση της δομής και της ονοματολογίας των δύο σχημάτων, η οποία περιλαμβάνει ταίριασμα σχήματος. [NB05]



Σχήμα 2.1 Προσεγγίσεις ταϊριάσματος σχημάτων [RB01]

2.3.6 Προσεγγίσεις μοντελοποίησης καθολικού σχήματος

Υπάρχουν δύο τρόποι έκφρασης των σχέσεων ανάμεσα στο καθολικό σχήμα και τις τοπικές πηγές. Στην πρώτη προσέγγιση, το καθολικό σχήμα εκφράζεται ως μία όψη των τοπικών σχημάτων, οπότε και οι σχέσεις ανάμεσα στα δύο είναι άμεσες (Global-As-View (GAV)). Στη δεύτερη προσέγγιση, το καθολικό σχήμα εκφράζεται ανεξάρτητα από τις τοπικές πηγές και οι σχέσεις ανάμεσα τους καθιερώνονται εκφράζοντας κάθε τοπική πηγή ως όψη του καθολικού σχήματος (Local-As-View (LAV)). Τέλος, υπάρχει άλλη μία προσέγγιση, η Global-Local-As-View (GLAV), όπου οι σχέσεις μεταξύ του καθολικού σχήματος και τις πηγές εγκαθιδρύονται με GAV και LAV ισχυρισμούς [Len02].

Έρευνα πάνω στην ολοκλήρωση δεδομένων έχει παράγει ένα σύνολο από πλούσιες και κατανοητές γλώσσες ενδιάμεσου σχήματος πάνω στις οποίες μπορεί να κατασκευαστεί η μεσολάβηση (mediation) στα PDMSs. Ο PPL (Peer-Programming Language) είναι ένας πολύ ευέλικτος φορμαλισμός για μεσολάβηση μεταξύ σχημάτων κόμβων, ο οποίος χρησιμοποιεί τους GAV και LAV φορμαλισμούς για να καθορίσουν τοπικές αντιστοιχίσεων [HIS+03].

Στη συνέχεια περιγράφουμε αναλυτικότερα τις προσεγγίσεις αυτές [Len02]:

2.3.6.1 LAV

Σε ένα σύστημα ολοκλήρωσης δεδομένων $I = (G, S, M)$ (όπου G το καθολικό σχήμα, S το σχήμα πηγής και M η αντιστοίχιση (mapping) μεταξύ G και S), βασισμένο στην προσέγγιση LAV, η αντιστοίχιση M συνδέει σε κάθε στοιχείο s του σχήματος πηγής S ένα ερώτημα q_G πάνω στο G . Με άλλα λόγια, η γλώσσα διατύπωσης ερωτημάτων $L_{M,S}$ επιτρέπει μόνο εκφράσεις που αποτελούνται από ένα σύμβολο του αλφαβήτου A_S . Επομένως, μια αντιστοίχιση LAV είναι ένα σύνολο ισχυρισμών, ένας για κάθε στοιχείο s του S της μορφής $s \rightarrow q_G$.

Από την άποψη της μοντελοποίησης, η προσέγγιση LAV είναι βασισμένη στην ιδέα ότι το περιεχόμενο κάθε πηγής s πρέπει να χαρακτηρίζεται στα πλαίσια μιας όψης q_G πάνω στο καθολικό σχήμα. Μια αξιοσημείωτη περίπτωση αυτού του τύπου είναι όταν το σύστημα ολοκλήρωσης δεδομένων είναι βασισμένο σε ένα επιχειρηματικό πρότυπο, ή μια οντολογία. Αυτή η ιδέα είναι αποτελεσματική όταν το σύστημα ολοκλήρωσης δεδομένων είναι βασισμένο σε ένα καθολικό σχήμα που είναι σταθερό και καλά καθιερωμένο (well established) στον οργανισμό. Η LAV ευνοεί την επεκτασιμότητα του συστήματος: η προσθήκη μίας νέας πηγής συνεπάγεται απλά τον εμπλουτισμό των αντιστοιχίσεων (mappings) με ένα νέο ισχυρισμό.

Για να χαρακτηρίσουν καλύτερα κάθε πηγή όσον αφορά το καθολικό σχήμα, διάφοροι συγγραφείς έχουν προτείνει περιπλοκότερους ισχυρισμούς στη αντιστοίχιση LAV, και συγκεκριμένα με στόχο την καθιέρωση της υπόθεσης που ισχύει για τις διάφορες επεκτάσεις πηγής. Τυπικά, αυτό σημαίνει ότι στην αντιστοίχιση LAV, μια νέα προδιαγραφή, $as(s)$, συνδέεται με κάθε στοιχείο πηγής s . Η προδιαγραφή $as(s)$ καθορίζει πόσο ακριβής είναι η γνώση σχετικά με τα δεδομένα που ικανοποιούν τις πηγές, δηλ., πόσο ακριβής είναι η πηγή όσον αφορά τη σχετική όψη q_G . Υπάρχουν τρεις δυνατότητες:

1. **Σωστές όψεις (Sound views):** Όταν μια πηγή s είναι sound ($as(s) = \text{sound}$), η επέκτασή της παρέχει οποιοδήποτε υποσύνολο πλειάδων που ικανοποιεί την αντίστοιχη όψη q_G . Με άλλα λόγια, δοθείσας μιας βάσης δεδομένων πηγής D , από το γεγονός ότι μία πλειάδα είναι στο s^D μπορούμε να καταλήξουμε στο συμπέρασμα ότι ικανοποιεί την αντίστοιχη όψη πάνω από το καθολικό σχήμα, ενώ από το γεγονός ότι μία πλειάδα δεν είναι στο s^D δεν μπορούμε να καταλήξουμε στο συμπέρασμα ότι δεν ικανοποιεί την αντίστοιχη όψη. Τυπικά, όταν $as(s) = \text{sound}$, μια βάση δεδομένων B ικανοποιεί τον ισχυρισμό $s \rightarrow q_G$ όσον αφορά τη D εάν $s^D \subseteq q_G^B$. Σημειώνουμε ότι, από λογική άποψη, μια sound πηγή s με πολλαπλότητα n μοντελοποιείται μέσω του ισχυρισμού πρώτης τάξης $\forall x, s(x) \rightarrow q_G(x)$, όπου x υποδηλώνει τις μεταβλητές x_1, \dots, x_n .
2. **Πλήρεις απόψεις (Complete views):** Όταν μια πηγή s είναι πλήρης ($as(s) = \text{complete}$), η επέκτασή της παρέχει οποιοδήποτε υπερσύνολο από πλειάδες που ικανοποιεί την αντίστοιχη όψη. Με άλλα λόγια, από το γεγονός ότι μία πλειάδα είναι στο s^D δεν μπορούμε να καταλήξουμε στο συμπέρασμα ότι μία τέτοια πλειάδα ικανοποιεί την αντίστοιχη όψη. Αφ' ετέρου, από το γεγονός ότι μία πλειάδα δεν είναι στο s^D μπορούμε να καταλήξουμε στο συμπέρασμα ότι μία τέτοια πλειάδα δεν ικανοποιεί την όψη. Τυπικά, όταν $as(s) = \text{complete}$, μια βάση δεδομένων B ικανοποιεί τον ισχυρισμό $s \rightarrow q_G$ όσον αφορά τη D εάν $s^D \supseteq q_G^B$. Από λογική άποψη, μια complete πηγή s με πολλαπλότητα n μοντελοποιείται μέσω του ισχυρισμού πρώτης τάξης $\forall x, q_G(x) \rightarrow s(x)$.
3. **Ακριβείς όψεις (Exact views):** Όταν μια πηγή s είναι ακριβής ($as(s) = \text{exact}$), η επέκτασή (extension) της είναι ακριβώς το σύνολο πλειάδων των αντικειμένων που ικανοποιούν την αντίστοιχη όψη. Τυπικά, όταν $as(s) = \text{exact}$, μια βάση δεδομένων B ικανοποιεί τον ισχυρισμό $s \rightarrow q_G$ όσον αφορά τη D εάν $s^D = q_G^B$. Από λογική άποψη, μια exact πηγή s με πολλαπλότητα n μοντελοποιείται μέσω του ισχυρισμού πρώτης τάξης $\forall x, s(x) \rightarrow q_G(x)$.

Όσον αφορά την διαδικασία επεξεργασίας ερωτήματος, όταν απαντάμε σε μια ερώτηση πέρα από το καθολικό σχήμα βάσει μιας αντιστοίχισης LAV, ξέρουμε μόνο τις επεκτάσεις των όψεων που συνδέονται στις πηγές, και αυτό παρέχει σε μας μόνο τις μερικές πληροφορίες για την καθολική βάση δεδομένων. Όπως παρατηρήσαμε ήδη, γενικά, υπάρχουν πολλές πιθανές καθολικές βάσεις δεδομένων που είναι νόμιμες για το σύστημα ολοκλήρωσης δεδομένων όσον αφορά μια δεδομένη βάση δεδομένων πηγής. Αυτή η παρατήρηση ισχύει ακόμη και για ένα περιβάλλον όπου μόνο sound όψεις επιτρέπονται στην αντιστοίχιση. Το πρόβλημα είναι ακόμα πιο περίπλοκο όταν μπορούν οι πηγές να μοντελοποιηθούν σαν πλήρεις (complete) ή ακριβείς (exact) όψεις. Συγκεκριμένα, η ασχολία με ακριβείς πηγές σημαίνει ουσιαστικά την εφαρμογή του ισχυρισμού κλειστού κόσμου στις αντίστοιχες όψεις.

Στη συνέχεια παραθέτουμε ένα παράδειγμα: Θεωρούμε σύστημα ολοκλήρωσης δεδομένων I με καθολικό σχεσιακό σχήμα G που περιέχει (μεταξύ άλλων σχέσεων) μία δυαδική σχέση couple, και δύο σταθερές Ann και Bill. Θεωρούμε επίσης δύο πηγές female και male, αντίστοιχα με τις σχετικές όψεις

$$\text{female}(f) \rightarrow \{ f, m \mid \text{couple}(f, m) \}$$

$$\text{male}(m) \rightarrow \{ f, m \mid \text{couple}(f, m) \}$$

και θεωρούμε μια βάση δεδομένων πηγής D με $\text{female}^D = \{\text{Ann}\}$ και $\text{male}^D = \{\text{Bill}\}$, και υποθέτουμε ότι δεν υπάρχει κανένας περιορισμός που να επιβάλλεται από κάποιο σχήμα. Εάν και οι δύο πηγές είναι sound, ξέρουμε μόνο ότι κάποιο ζεύγος έχει το Ann ως θηλυκό (female) και το Bill ως αρσενικό (male) συστατικό του. Επομένως, η ερώτηση

$$Q = \{ x, y \mid \text{couple}(x, y) \}$$

που ζητάει όλα τα ζεύγη, θα επέστρεφε μια κενή απάντηση, δηλ., $Q_C^{I,D} = \emptyset$. Εντούτοις, εάν και οι δύο πηγές είναι exact, μπορούμε να συμπεράνουμε ότι όλα τα ζεύγη έχουν το Ann ως θηλυκό και το και Bill ως αρσενικό συστατικό τους, και ως εκ τούτου ότι (Ann,Bill) είναι το μόνο ζεύγος, δηλ., $Q_C^{I,D} = \{ (\text{Ann}, \text{Bill}) \}$.

2.3.6.2 GAV

Στην προσέγγιση GAV, η αντιστοίχιση M συνδέει σε κάθε στοιχείο g στο G ένα ερώτημα q_S πάνω από το S. Με άλλα λόγια, η γλώσσα διατύπωσης ερωτημάτων $L_{M,G}$ επιτρέπει μόνο τις εκφράσεις που αποτελούνται από ένα σύμβολο του αλφάβητου A_G . Επομένως, μια αντιστοίχιση GAV είναι ένα σύνολο ισχυρισμών, ένας για κάθε στοιχείο g του G, της μορφής $g \rightarrow q_S$.

Από την άποψη μοντελοποίησης, η προσέγγιση GAV είναι βασισμένη στην ιδέα ότι το περιεχόμενο κάθε στοιχείου g του καθολικού σχήματος πρέπει να χαρακτηριστεί στα πλαίσια μιας όψης q_s πάνω από τις πηγές. Υπό κάποια έννοια, η αντιστοίχιση λέει ρητά στο σύστημα πώς να ανακτήσει τα στοιχεία όταν κάποιος θέλει να αποτιμήσει τα διάφορα στοιχεία του καθολικού σχήματος. Αυτή η ιδέα είναι αποτελεσματική όταν το σύστημα ολοκλήρωσης δεδομένων είναι βασισμένο σε ένα σύνολο πηγών που είναι σταθερό. Σημειώνουμε ότι, σε γενικές γραμμές, η προσέγγιση GAV ευνοεί το σύστημα στην εκτέλεση της επεξεργασίας των ερωτημάτων, επειδή λέει στο σύστημα πώς να χρησιμοποιήσει τις πηγές για να ανακτήσει τα στοιχεία. Εντούτοις, η επέκταση του συστήματος με μια νέα πηγή είναι τώρα ένα πρόβλημα: η νέα πηγή μπορεί πράγματι να ασκήσει επίδραση στον καθορισμό των διάφορων στοιχείων του καθολικού σχήματος, οι σχετικές όψεις του οποίου πρέπει να επαναπροσδιοριστούν.

Για να χαρακτηρίσουν καλύτερα κάθε στοιχείο του καθολικού σχήματος όσον αφορά τις πηγές, οι περιπλοκότεροι ισχυρισμοί στη αντιστοίχιση GAV μπορούν να χρησιμοποιηθούν, στο ίδιο πνεύμα όπως είδαμε για τη LAV. Τυπικά, αυτό σημαίνει ότι στη αντιστοίχιση GAV, μια νέα προδιαγραφή, $as(g)$ (είτε *sound*, *complete*, είτε *exact*) συνδέεται σε κάθε στοιχείο g του καθολικού σχήματος. Όταν $as(g) = \text{sound}$ (αντίστοιχα, *complete*, *exact*), μια βάση δεδομένων B ικανοποιεί τον ισχυρισμό $g \rightarrow q_s$ όσον αφορά μια βάση δεδομένων πηγής D εάν $q_s^D \subseteq g^B$ (αντίστοιχα $q_s^D \supseteq g^B$, $q_s^D = g^B$).

Ο λογικός χαρακτηρισμός των *sound* και των *complete* όψεων σε GAV είναι επομένως μέσω των ισχυρισμών πρώτης τάξης $\forall x, q_s(x) \rightarrow g(x)$, $\forall x, g(x) \rightarrow q_s(x)$ αντίστοιχα.

Είναι ενδιαφέρον να παρατηρηθεί ότι η αυτονόητη υπόθεση σε πολλές προτάσεις GAV είναι αυτή των *exact* όψεων. Πράγματι, σε ένα περιβάλλον όπου όλες οι όψεις είναι *exact*, δεν υπάρχει κανένας περιορισμός στο καθολικό σχήμα, και μια γλώσσα διατύπωσης ερωτημάτων πρώτης τάξης χρησιμοποιείται ως $L_{M,G}$, ένα σύστημα ολοκλήρωσης στοιχείων GAV απολαμβάνει αυτό που μπορούμε να καλέσουμε «απλή ιδιοκτησία βάσεων δεδομένων» (“single database property”), δηλ., χαρακτηρίζεται από μια ενιαία βάση δεδομένων, δηλαδή την καθολική βάση δεδομένων που λαμβάνεται με την ένωση σε κάθε στοιχείο του συνόλου των πλειάδων που υπολογίζονται από την αντίστοιχη όψη πάνω από τις πηγές. Αυτό παρακινεί την ευρέως κοινή διαίσθηση ότι η επεξεργασία ερωτημάτων σε GAV είναι ευκολότερη απ' ό,τι σε LAV. Εντούτοις, πρέπει να τονίσουμε ότι η απλή ιδιοκτησία βάσεων δεδομένων ισχύει μόνο σε ένα τέτοιο περιορισμένο περιβάλλον.

Συγκεκριμένα, η δυνατότητα καθορισμού των περιορισμών στο G ενισχύει πολύ τη δύναμη μοντελοποίησης των συστημάτων GAV, ειδικά σε εκείνες τις καταστάσεις όπου το καθολικό σχήμα προορίζεται να εκφραστεί στα πλαίσια ενός εννοιολογικού προτύπου δεδομένων, ή στα πλαίσια μιας οντολογίας. Σε αυτές τις περιπτώσεις, η γλώσσα L_G είναι στην

πραγματικότητα αρκετά ισχυρή για να επιτρέψει να διευκρινιστούν, είτε κρυφά είτε ρητά, διάφορες μορφές περιορισμών ακεραιότητας στην καθολική βάση δεδομένων.

Όσον αφορά την διαδικασία επεξεργασίας ερωτήματος, τα περισσότερα συστήματα ολοκλήρωσης δεδομένων GAV δεν επιτρέπουν τους περιορισμούς ακεραιότητας στο καθολικό σχήμα, και υποθέτουν ότι οι όψεις είναι ακριβείς (exact). Είναι εύκολο να δει κανείς ότι, σε αυτές τις περιπτώσεις, η επεξεργασία ερωτημάτων μπορεί να βασιστεί σε μια απλή στρατηγική ξεδιπλώματος. Όταν έχουμε ένα ερώτημα q πάνω στο αλφάβητο A_G του καθολικού σχήματος, κάθε στοιχείο του A_G αντικαθίσταται με την αντίστοιχη ερώτηση πάνω στις πηγές, και το προκύπτον ερώτημα αποτιμάται έπειτα στις πηγές. Όπως είπαμε πριν, μια τέτοια στρατηγική αρκεί κυρίως επειδή το σύστημα ολοκλήρωσης δεδομένων απολαμβάνει την απλή ιδιοκτησία βάσεων δεδομένων. Ειδικότερα, η ίδια στρατηγική ισχύει επίσης στην περίπτωση των sound όψεων.

Εντούτοις, όταν η γλώσσα L_G που χρησιμοποιείται για την έκφραση του καθολικού σχήματος, επιτρέπει τους περιορισμούς ακεραιότητας, και οι όψεις είναι sound, τότε η επεξεργασία ερώτησης στα συστήματα GAV γίνεται πιο σύνθετη. Πράγματι, σε αυτήν την περίπτωση, οι περιορισμοί ακεραιότητας μπορούν σε γενικές γραμμές να χρησιμοποιηθούν προκειμένου να υπερνικηθεί η μη πληρότητα των δεδομένων στις πηγές. Το ακόλουθο παράδειγμα δείχνει ότι, λαμβάνοντας υπόψη περιορισμούς ξένου κλειδιού, μπορεί κανείς να λάβει τις απαντήσεις που θα έλειπαν με απλό «ξεδίπλωμα» του ερωτήματος του χρήστη.

Έστω $I = (G,S,M)$ ένα σύστημα ολοκλήρωσης δεδομένων, όπου το G αποτελείται από τις σχέσεις:

employee(Ecode, Ename, Ecity)

company(Ccode, Cname)

employed(Ecode, Ccode)

και τους περιορισμούς:

key(employee) = {Ecode}

key(company) = {Ccode}

employed[Ecode] \subseteq employee[Ecode]

employed[Ccode] \subseteq company[Ccode]

Το σχήμα S πηγής αποτελείται από τρεις πηγές. Η πηγή s_1 , με πολλαπλότητα 4, περιέχει πληροφορίες για τους υπαλλήλους με τον κώδικα, το όνομα, την πόλη, και την ημερομηνία γέννησής τους. Η πηγή s_2 , με πολλαπλότητα 2, περιέχει τους κώδικες και τα ονόματα των επιχειρήσεων. Τέλος, η πηγή s_3 , με πολλαπλότητα 2, περιέχει πληροφορίες για την απασχόληση στις επιχειρήσεις. Η αντιστοίχιση M καθορίζεται από:

employee $\rightarrow: \{x,y,z \mid s_1(x,y,z,w)\}$

company $\rightarrow: \{x,y \mid s_2(x,y)\}$

employed $\rightarrow: \{x,w \mid s_3(x,w)\}$

Θεωρούμε τώρα το ακόλουθο ερώτημα χρήστη q , που ζητάει κωδικούς υπαλλήλων:

$\{x \mid \text{employee}(x,y,z)\}$

Υποθέτουμε ότι τα στοιχεία που αποθηκεύονται στη βάση δεδομένων πηγής D είναι εκείνα που απεικονίζονται στο σχήμα 1: απλά ξεδιπλώνοντας το q λαμβάνουμε την απάντηση $\{12\}$. Ωστόσο, λόγω του περιορισμού ακεραιότητας $\text{employed}[\text{Ecode}] \subseteq \text{employee}[\text{Ecode}]$, ξέρουμε ότι το 16 είναι ο κωδικός ενός προσώπου, ακόμα κι αν δεν εμφανίζεται στο s_1^D . Η σωστή απάντηση στο q είναι επομένως $\{12,16\}$. Παρατηρούμε ότι δεν ξέρουμε οποιαδήποτε τιμή για τις ιδιότητες (attributes) του υπαλλήλου του οποίου ο Ecode είναι 16.

$s_1^D :$	12	<i>calvin</i>	<i>rome</i>	21
	15	<i>alice</i>	<i>hong kong</i>	24
$s_2^D :$	<i>AF</i>	<i>hotdog corp.</i>		
	<i>BN</i>	<i>banana ltd.</i>		
$s_3^D :$	12	<i>AF</i>		
	16	<i>BN</i>		

Σχήμα 2.2 Επέκταση των πηγών για το παράδειγμα

Δοθείσας μιας βάσης δεδομένων πηγής D , ονομάζουμε «ανακτημένη καθολική βάση δεδομένων» τη καθολική βάση δεδομένων που λαμβάνεται από «γέμισμα» (populating) κάθε σχέσης r στο καθολικό σχήμα σύμφωνα με τη αντιστοίχιση, δηλ., με «γέμισμα» της r με τις πλειάδες που λαμβάνονται με την αποτίμηση του ερωτήματος που η αντιστοίχιση συνδέει με το q . Γενικά, οι περιορισμοί ακεραιότητας μπορούν να παραβιαστούν στην ανακτημένη καθολική βάση δεδομένων (π.χ., η ανακτημένη καθολική βάση δεδομένων για το ανωτέρω παράδειγμα). Όσον αφορά τους περιορισμούς κλειδιού, υποθέτουμε ότι το ερώτημα που η αντιστοίχιση συνδέει σε κάθε σχέση r καθολικού σχήματος είναι τέτοιο που τα δεδομένα που ανακτώνται για την r δεν παραβιάζουν τον περιορισμό κλειδιού της r . Με άλλα λόγια, η διαχείριση των περιορισμών κλειδιού αφήνεται στο σχεδιαστή. Από την άλλη, η διαχείριση των περιορισμών ξένου κλειδιού δεν μπορούν να αφηθούν στον σχεδιαστή, δεδομένου ότι συσχετίζεται έντονα με τη μη πληρότητα των πηγών. Επιπλέον, δεδομένου ότι τα ξένα κλειδιά είναι περιορισμοί αντιστοίχισης, δεν μπορούν να ασχοληθούν με τη αντιστοίχιση GAV, η οποία, εξ ορισμού, λειτουργεί σε κάθε καθολική σχέση απομονωμένα.

2.3.6.3 Σύγκριση LAV και GAV

Όπως αναφέραμε και παραπάνω, η LAV εννοεί την επεκτασιμότητα του συστήματος: προσθέτοντας μία νέα πηγή συνεπάγεται απλά τον εμπλουτισμό των αντιστοιχίσεων (mappings) με ένα νέο ισχυρισμό (assertion). Αντίθετα η GAV υποφέρει από την αναγκαιότητα αλλαγής του καθολικού σχήματος (σε χρόνο σχεδίασης) αν υπάρχει μία αλλαγή στο τοπικό σχήμα. Η νέα πηγή μπορεί, δηλαδή, να έχει επίδραση στον ορισμό ορισμένων στοιχείων του καθολικού σχήματος, του οποίου οι συσχετιζόμενες όψεις πρέπει να επαναοριστούν. Η GAV εννοεί το σύστημα να εκτελέσει την διαδικασία ερωτημάτων (query processing), γιατί λέει στο σύστημα πώς να χρησιμοποιεί τις πηγές για να ανακτήσει τα δεδομένα. Αντίθετα, η LAV οδηγεί στο πρόβλημα της περίπλοκης υλοποίησης τεχνικών επεξεργασίας ερωτημάτων.

2.3.7 Επεξεργασία ερωτήματος (Query processing)

Η επεξεργασία σημασιολογικού ερωτήματος είναι ένα σενάριο χρόνου εκτέλεσης, όπου ένας χρήστης καθορίζει το αποτέλεσμα ενός ερωτήματος (π.χ. ο όρος SELECT στην SQL), και το σύστημα υπολογίζει πώς να παράγει αυτό το αποτέλεσμα (π.χ. καθορίζοντας τους FROM και WHERE όρους στην SQL). Οι προδιαγραφές του χρήστη προκαθορίζονται στα πλαίσια εννοιών οικείων με αυτή, η οποία μπορεί να μην είναι ίδια με τα ονόματα των στοιχείων που έχουν καθοριστεί στο σχήμα της βάσης δεδομένων. Επομένως, στην πρώτη φάση της επεξεργασίας του ερωτήματος, το σύστημα πρέπει να συσχετίσει τις έννοιες που έχουν καθοριστεί από το χρήστη στο αποτέλεσμα του ερωτήματος με τα στοιχεία του σχήματος. Αυτό είναι επίσης μία φυσική εφαρμογή λειτουργίας ταιριάσματος. Μετά το συσχετισμό του αποτελέσματος του ερωτήματος με τα στοιχεία του σχήματος, το σύστημα πρέπει να αποκομίζει ένα προσόν (π.χ. ένα όρο WHERE) που δίνει τη σημασιολογία του συσχετισμού. Για το λόγο αυτό έχουν αναπτυχθεί τεχνικές.

Για την επεξεργασία ερωτήματος βασισμένη σε όψεις, υπάρχουν δύο προσεγγίσεις, η **αναδιατύπωση ερωτήματος βασισμένη σε όψεις (view-based query rewriting)** και η **απάντηση ερωτήματος βασισμένη σε όψεις (view-based query answering)** [Len02].

Το πρόβλημα της αναδιατύπωσης ερωτήματος με χρήση όψεων είναι το πρόβλημα αναδιατύπωσης ενός δοθέντος ερωτήματος χρησιμοποιώντας μόνο τις σχέσεις και τις ιδιότητες (attributes) στο δοθέν σύνολο όψεων. Στην αναδιατύπωση ερωτήματος, η επεξεργασία ερωτήματος διαιρείται σε δύο βήματα, όπου το πρώτο επανεκφράζει το ερώτημα στα πλαίσια μίας δοθείσας γλώσσας ερωτημάτων πάνω στο αλφάβητο των ονομάτων των όψεων, και το δεύτερο αποτιμά την αναδιατύπωση ερωτήματος πάνω στις επεκτάσεις των

όψεων. Η αναδιατύπωση ερωτήματος χρησιμοποιώντας όψεις είναι εφαρμόσιμη σε πολλές περιοχές των βάσεων δεδομένων: ολοκλήρωση δεδομένων, βελτιστοποίηση ερωτημάτων και αποθήκευση δεδομένων. Στην βελτιστοποίηση ερωτημάτων η αναδιατύπωση ερωτημάτων χρησιμοποιείται για να πάρουμε ένα ισοδύναμο ερώτημα με αποδοτικά σχέδια ερωτήματος. Στην ολοκλήρωση δεδομένων είναι εφαρμόσιμο στην μετάφραση ερωτημάτων.

Στην απάντηση ερωτήματος βασισμένη σε όψεις, εκτός από το ερώτημα και τους ορισμούς των όψεων, δίνονται και οι επεκτάσεις των όψεων. Σκοπός είναι να υπολογίσουμε το σύνολο των πλειάδων (tuples), ώστε η γνώση στις επεκτάσεις των όψεων να υποδηλώνει ότι το σύνολο των πλειάδων είναι μία απάντηση στο ερώτημα. Εδώ δεν θέτουμε περιορισμούς στο πως θα γίνει η επεξεργασία των ερωτημάτων και ο μόνος σκοπός είναι να εκμεταλλευτούμε όλη τη δυνατή πληροφορία, και συγκεκριμένα τις επεκτάσεις των όψεων, για να υπολογίσουμε την απάντηση στο ερώτημα.

2.3.8 *Μετάφραση ερωτήματος (Query translation)*

Η μετάφραση ερωτημάτων σε ένα P2P σύστημα είναι το πρόβλημα διατύπωσης ερωτήματος σε δεδομένα αποθηκευμένα σε διαφορετικούς κόμβους [NB05]. Η μετάφραση ερωτημάτων σε ένα P2P δίκτυο διαμοιράσματος αρχείων είναι ευκολότερη από ένα βάσεων δεδομένων. Αυτό συμβαίνει επειδή τα δεδομένα στο διαμοιρασμό βάσεων δεδομένων είναι σημασιολογικά πλούσια και αυτό δίνει στους χρήστες την ικανότητα να κάνουν πλούσια ερωτήματα. Τα προβλήματα που προκύπτουν σαν αποτέλεσμα του διαμοιρασμού σημασιολογικά πλούσιων και ετερογενών δεδομένων είναι δομικές, ονομαστικές και σημασιολογικές συγκρούσεις. Ο σκοπός της μετάφρασης ερωτημάτων είναι να χρησιμοποιηθεί ένα ερώτημα για να ερωτηθούν πολλαπλές βάσεις. Αυτές οι βάσεις είναι φυσικά κατανεμημένες και έχουν ετερογενή σχήματα. Γι' αυτό το λόγο το πρόβλημα σχετίζεται με το πρόβλημα ολοκλήρωσης δεδομένων. Η προτεινόμενη λύση στο πρόβλημα μετάφρασης ερωτημάτων βασίζεται στην κατάληψη αρκετών μεταδεδομένων για τα δεδομένα που είναι αποθηκευμένα σε κάθε κόμβο.

2.4 *Οντολογίες (Ontologies)*

Μία οντολογία είναι μία κατηγορηματική, τυπική προδιαγραφή μίας διαμοιρασμένης εννοιολογικής αναπαράστασης (conceptualization) μερικών περιοχών ενδιαφέροντος [BGM05]. Η μελέτη της γνώσης του κόσμου επιτυγχάνεται με μία αφηρημένη θεωρία γνώσης, η οποία τυποποιεί τις γνώσεις αναφορικά με αντικείμενα, κλάσεις, σχέσεις,

συναρτήσεις και αξιώματα. Συνήθως μία δηλωτική γλώσσα αναπαράστασης γνώσης με μία οντολογική δέσμευση διευκολύνει το μοίρασμα γνώσης μεταξύ των πρακτόρων [bilkent]. Μία οντολογία είναι ένας φορμαλισμός που δημιουργήθηκε για να περιγράψει τη σημασιολογία των δεδομένων (metadata) και να λειτουργεί ως ένας μηχανισμός διαλειτουργικότητας ανάμεσα σε ανθρώπους, φορείς και συστήματα, που τους επιτρέπει να μοιράζονται γνώση σε ένα δομημένο τρόπο. Με αυτό, αναφερόμαστε στην οντολογία σαν μία θεωρία κωδικοποίησης (theory of classification) [bilkent]. Για παράδειγμα, μία αντικειμενοστρεφής γλώσσα μπορεί να θεωρηθεί μία οντολογία κώδικα, αφού έχει τυποποιημένες δομές και σχέσεις, και έτσι επιτρέπει το μοίρασμα και την επαναχρησιμοποίηση κώδικα.

Μία κοινή χρήση των οντολογιών είναι η τυποποίηση και σύλληψη στοιχείων μέσω μιας επίσημης κατανοητής από τη μηχανή γλώσσας οντολογίας. Για παράδειγμα, το καθολικό σχήμα σε ένα σύστημα ολοκλήρωσης δεδομένων μπορεί να είναι μία οντολογία, η οποία δρα σαν ένας μεσολαβητής (mediator) για την εξάλειψη των ετερογενειών μεταξύ των διαφόρων πηγών.

Η οντολογία μπορεί επίσης να αναφέρεται σε μία συγκεκριμένη κωδικοποίηση σε κάποιες έννοιες, δηλαδή να αναπαριστά εννοιολογικά ένα τομέα. Εφόσον μπορεί να αναπαραστήσει εννοιολογικά έναν τομέα, μπορεί να αξιοποιηθεί αποτελώντας μία «ομπρελά» όρων και σημασιών που εκφράζουν την ίδια έννοια. Με βάση αυτή τους την ιδιότητα, οι οντολογίες λειτουργούν ως φορέας επικοινωνίας μεταξύ διαφορετικών πληροφοριακών συστημάτων, με διαφορετικούς χρήστες, παρέχοντας μία κοινή βάση ανάμεσα σε αυτά, αναπαριστώντας και αναλύοντας τις οντότητες που περιγράφουν τα δεδομένα τους. Τα είδη των εννοιών, οι συναρτήσεις (ή ιδιότητες) τους, οι σχέσεις μεταξύ τους, τα στιγμιότυπα τους και τα αξιώματα (δηλώσεις για την περιοχή που αληθεύουν) που εκφράζουν τη γνώση μας πάνω σε μία περιοχή αποτελούν μία οντότητα αυτής της περιοχής. Μία οντολογία καθορίζει τι είναι απαραίτητο για την εννοιολογική αναπαράσταση της γνώσης. Πρέπει να μας δίνει ένα τρόπο να εκφράζουμε τη γνώση μας.

2.4.1 Χρήση

Οι οντολογίες χρησιμοποιούνται όλο και περισσότερο σε ποικίλους τομείς, όπως η διαχείριση γνώσης, η εξαγωγή πληροφοριών και ο σημασιολογικός ιστός. Η αποτίμηση οντολογιών είναι το πρόβλημα εκτίμησης μίας δοθείσας οντολογίας από την οπτική ενός συγκεκριμένου κριτηρίου εφαρμογής, τυπικά για να καθορίσει ποια από αρκετές οντολογίες θα ταίριαζε καλύτερα για ένα συγκεκριμένο σκοπό.

Το ενδιαφέρον των σύγχρονων πληροφοριακών συστημάτων έχει μετακινηθεί από την επεξεργασία δεδομένων (data processing) στην επεξεργασία εννοιών (concept processing), που σημαίνει ότι η βασική μονάδα επεξεργασίας είναι όλο και λιγότερο ένα ατομικό κομμάτι δεδομένων και γίνεται περισσότερο μία σημασιολογική έννοια η οποία φέρει μία ερμηνεία και υπάρχει σε ένα ευρύτερο πλαίσιο με άλλες έννοιες [BGM05]. Η οντολογία χρησιμοποιείται κοινά σαν μία δομή που αιχμαλωτίζει γνώση για μία συγκεκριμένη περιοχή με το να παρέχει σχετικές έννοιες και σχέσεις μεταξύ τους. Ένας παράγοντας κλειδί που κάνει ένα συγκεκριμένο κλάδο επιστημονικό είναι η ικανότητα να εκτιμά και να συγκρίνει τις ιδέες μέσα στην περιοχή.

Το ίδιο ισχύει επίσης για την έρευνα του σημασιολογικού ιστού όταν ενδιαφερόμαστε για αφαιρέσεις στη μορφή οντολογιών. Ο σημασιολογικός ιστός είναι μία επέκταση του διαδικτύου στην οποία οι πληροφορίες δίνονται καλώς ορισμένες [GM03]. Αυτό σημαίνει ότι δίνεται η δυνατότητα στους υπολογιστές και στους ανθρώπους να δουλεύουν σε συνεργασία. Αυτή η συνεργασία μπορεί να επιτευχθεί χρησιμοποιώντας μοιραζόμενα γνωστικά συστατικά. Η οντολογία είναι μια θεμελιώδης δομή δεδομένων για να αντιλαμβανόμαστε τη γνώση, αλλά είμαστε γενικά ικανοί να κατασκευάσουμε πολλές διαφορετικές οντολογίες σχηματίζοντας την εικόνα του ίδιου σώματος γνώσης και πρέπει να είμαστε σε θέση να πούμε σε ποιες από αυτές ταιριάζει καλύτερα κάποιο προκαθορισμένο κριτήριο. Κατά συνέπεια, η **αξιολόγηση οντολογίας** είναι ένα σημαντικό ζήτημα που πρέπει να αντιμετωπιστεί εάν οι οντολογίες πρόκειται να υιοθετηθούν ευρέως στο σημασιολογικό Ιστό και σε άλλες σημασιολογικά ενήμερες εφαρμογές [BGM05]. Οι χρήστες που αντιμετωπίζουν ένα πλήθος οντολογιών πρέπει να έχουν έναν τρόπο να τις εκτιμούν και να αποφασίζουν ποια ταιριάζει καλύτερα στις απαιτήσεις τους. Επιπλέον, οι άνθρωποι που κατασκευάζουν μια οντολογία χρειάζονται έναν τρόπο να αξιολογήσουν την προκύπτουσα οντολογία και ενδεχομένως να καθοδηγήσουν τη διαδικασία κατασκευής και οποιαδήποτε βήματα καθαρισμού.

Οι οντολογίες αντιπροσωπεύουν γνώση στατικής περιοχής. Γι' αυτό το λόγο, είναι απαραίτητο να αναπτυχθούν μέθοδοι και τεχνικές που επιτρέπουν να μειωθεί η προσπάθεια που απαιτείται για τη διαδικασία απόκτησης γνώσης, το οποίο είναι ο στόχος της **εκμάθησης οντολογιών (ontology learning)** [GM03]. Η απόκτηση γνώσεων περιοχής για την κατασκευή οντολογιών απαιτεί πολύ χρόνο και πολλές πηγές. Με αυτή την έννοια, μπορούμε να ορίσουμε την εκμάθηση οντολογιών σαν ένα σύνολο από μεθόδους και τεχνικές που χρησιμοποιούνται για να κατασκευαστεί μία οντολογία από την αρχή, εμπλουτίζοντας ή προσαρμόζοντας μία υπάρχουσα οντολογία σε ένα ημι-αυτόματο τρόπο χρησιμοποιώντας μερικές πηγές. Οι αυτοματοποιημένες ή ημιαυτόματες τεχνικές εκμάθησης οντολογίας απαιτούν επίσης αποτελεσματικά μέτρα αξιολόγησης, τα οποία μπορούν να χρησιμοποιηθούν

για να επιλέξουν τη «καλύτερη» οντολογία από πολλούς υποψηφίους, για να επιλέξουν τις τιμές μεταβλητών παραμέτρων του αλγορίθμου εκμάθησης, ή για να κατευθύνουν την ίδια τη διαδικασία εκμάθησης (εάν το τελευταίο διατυπώνεται ως πορεία μέσω ενός διαστήματος αναζήτησης). Για την αναφορά σε ημι-αυτόματες κατασκευές οντολογιών χρησιμοποιούνται επίσης και άλλοι όροι, όπως παραγωγή οντολογιών (ontology generation), εξόρυξη οντολογιών (ontology mining), εξαγωγή οντολογιών (ontology extraction), κα. Μερικές προσεγγίσεις υπάρχουν για την μερική αυτοματοποίηση της διαδικασίας απόκτησης γνώσης. Για να εκτελεστεί αυτή η αυτοματοποίηση, μπορούν να χρησιμοποιηθούν ανάλυση φυσικής γλώσσας και τεχνικές εκμάθησης μηχανής (machine learning techniques).

2.4.2 Γλώσσες Οντολογιών

Στην παράγραφο αυτή θα περιγράψουμε σύντομα μερικές από τις υπάρχουσες γλώσσες οντολογιών. [CX05]

2.4.2.1 Σχήμα XML

Για να κυριολεκτήσουμε, το σχήμα XML είναι μια σημασιολογική γλώσσα σήμανσης για τα στοιχεία Ιστού. Οι συμβατοί με τη βάση δεδομένων τύποι δεδομένων που υποστηρίζονται από το XML σχήμα παρέχουν έναν τρόπο να διευκρινιστεί ένα ιεραρχικό μοντέλο. Εντούτοις, δεν υπάρχει κανένα ρητό κατασκεύασμα για τον καθορισμό των κλάσεων και των ιδιοτήτων στο XML σχήμα, επομένως οι ασάφειες μπορούν να προκύψουν κατά την αντιστοίχιση (mapping) ενός βασισμένου σε XML μοντέλου δεδομένων σε ένα σημασιολογικό μοντέλο.

2.4.2.2 RDF και RDFS

Το RDF (πλαίσιο περιγραφής πόρων) είναι ένα μοντέλο δεδομένων που αναπτύχθηκε από το W3C για την περιγραφή των πηγών του Ιστού. Το RDF επιτρέπει την προδιαγραφή της σημασιολογίας των δεδομένων με έναν τυποποιημένο, διαλειτουργικό τρόπο. Στο RDF, ένα ζευγάρι των πηγών (κόμβοι) που συνδέονται με μια ιδιότητα (άκρη) διαμορφώνει μια δήλωση: (πηγή, ιδιότητα, αξία). Το RDFS (RDF σχήμα) είναι μια γλώσσα περιγραφής των λεξιλογίων των RDF δεδομένων από την άποψη των πρωτόγονων στοιχείων όπως τα `rdfs:Κλάση`, `rdf:Ιδιότητα`, `rdfs:περιοχή`, και `rdfs:σειρά`. Με άλλα λόγια, το RDFS χρησιμοποιείται για να καθορίσει τις σημασιολογικές σχέσεις μεταξύ των ιδιοτήτων και των πόρων.

2.4.2.3 DAML+OIL

Η DAML-OIL (γλώσσα σήμανσης πρακτόρων DARPA – γλώσσα διεπαφών οντολογίας) είναι μια βασισμένη σε πλήρη-fledgedWeb γλώσσα οντολογίας που αναπτύχθηκε πάνω από RDFS.7, που χαρακτηρίζει μια βασισμένη σε XML σύνταξη και μια αρχιτεκτονική σε στρώσεις. Η DAML-OIL παρέχει τους πρωτόγονους μοντελοποίησης που χρησιμοποιούνται συνήθως σε βασισμένες σε πλαίσιο προσεγγίσεις στην εφαρμοσμένη μηχανική οντολογίας (ontology engineering), και την επίσημη υποστήριξη σημασιολογίας και συλλογισμού που βρίσκεται σε προσεγγίσεις λογικής περιγραφής. Ενσωματώνει επίσης τους τύπους δεδομένων σχημάτων XML για τη σημασιολογική διαλειτουργικότητα σε XML.

2.4.2.4 OWL

Η OWL (Web Ontology Language - γλώσσα οντολογίας Ιστού) είναι μια σημασιολογική γλώσσα σήμανσης για την έκδοση και το διαμοιρασμό οντολογιών στον Ιστό. Αναπτύχθηκε ως μία επέκταση λεξιλογίου του RDF και προέρχεται από τη DAML+OIL. Άλλες γλώσσες οντολογίας είναι οι SHOE (Simple HTML Ontology Extensions), η XOL (Ontology Exchange Language - γλώσσα ανταλλαγής οντολογίας), και η UML (Unified Modeling Language - ενοποιημένη γλώσσα διαμόρφωσης). Μεταξύ όλων αυτών των γλωσσών οντολογίας, ενδιαφερόμαστε για το σχήμα XML και το RDFS για τους ιδιαίτερους ρόλους τους στην ολοκλήρωση δεδομένων και το «σημασιολογικό Ιστό». Πιο συγκεκριμένα, το σχήμα XML και το RDFS χρησιμοποιούν την ίδια σύνταξη και μπορούν να χρησιμοποιηθούν για τη μοντελοποίηση δεδομένων και την αντιπροσώπευση οντολογίας. Αλλά έχουν τα ιδιαίτερα χαρακτηριστικά γνωρίσματά τους υπό την έννοια ότι τα XML δεδομένα έχουν δομή εγγράφου (document structure) από άποψη των φωλιασμένων στοιχείων σε ένα μεμονωμένο έγγραφο XML, ενώ τα RDF δεδομένα έχουν δομή περιοχής (domain structure) διαμορφωμένη από τις έννοιες και τις σχέσεις μεταξύ των εννοιών.

2.4.2.5 Datalog

Η Datalog είναι μια γλώσσα που χρησιμοποιείται για την αναπαράσταση συζευκτικών ερωτημάτων (conjunctive queries) και κυρίως SPJ (select – project – join) ερωτημάτων. Ένα ερώτημα (ή οψη) έχει τη μορφή:

$$M(X) :- T_1(X_1), T_2(X_2), \dots, T_n(X_n)$$

Το αριστερό μέλος λέγεται κεφαλή (head) και το δεξί σώμα (body). Τα στοιχεία (atoms) του ερωτήματος λέγονται κατηγορήματα (predicates) και αποτελούνται από έναν πίνακα (δηλαδή μία σχέση στο σχεσιακό μοντέλο) της βάσης δεδομένων και ένα σύνολο ιδιοτήτων.

Το όνομα του κατηγορήματος της κεφαλής (M) είναι το όνομα της όψης. Οι αντιστοιχίσεις (mappings) που χρησιμοποιούμε σε P2P βάσεις δεδομένων είναι όψεις. Τα ονόματα των κατηγορημάτων του σώματος (T_1, T_2, \dots, T_n) είναι τα ονόματα των πινάκων που συμμετέχουν σε μία αντιστοίχιση. Το X είναι το σύνολο ιδιοτήτων της αντιστοίχισης και τα X_1, X_2, \dots, X_n τα σύνολα ιδιοτήτων των πινάκων.

Σημαντικό στο συμβολισμό της Datalog είναι το γεγονός ότι τα ονόματα των ιδιοτήτων κάθε πίνακα θεωρούνται γνωστά και δεν εμφανίζονται στο Datalog ερώτημα. Οι ιδιότητες ενός πίνακα διακρίνονται μεταξύ τους από τη θέση στην οποία βρίσκονται μέσα στο κατηγορήμα που αντιστοιχεί στον πίνακα. Η τιμή κάθε ιδιότητας αναπαρίσταται από μία μεταβλητή, ή μία σταθερά, εάν θέλουμε αυτή να έχει μία συγκεκριμένη τιμή. Με τον τρόπο αυτό, οι σύνδεσμοι (joins) ανάμεσα στους πίνακες εκφράζονται σαν πολλαπλές εμφανίσεις της ίδιας μεταβλητής στις θέσεις των αντίστοιχων ιδιοτήτων. Οι ιδιότητες της αντιστοίχισης παίρνουν τιμές από μεταβλητές που εμφανίζονται σαν ιδιότητες πινάκων στο σώμα του ερωτήματος. Έτσι, για να έχει η αντιστοίχιση νόημα πρέπει το σύνολο X να είναι υποσύνολο (όχι γνήσιο) της ένωσης των συνόλων X_1, X_2, \dots, X_n .

Εκτός από του παραπάνω τύπου κατηγορήματα, το σώμα ενός ερωτήματος μπορεί να περιέχει κατηγορήματα με αριθμητικές συγκρίσεις. Τα κατηγορήματα αυτά έχουν τη μορφή: $x_i \text{ op } \text{const}$, όπου op: ένας οποιοσδήποτε τελεστής και const: μία σταθερά, ανάλογου τύπου με αυτόν της μεταβλητής x_i , η οποία ανήκει σε κάποιο ή κάποια από τα X_1, X_2, \dots, X_n . Τα κατηγορήματα κάθε τύπου του σώματος ενός ερωτήματος Datalog λέγονται subgoals. Τα κατηγορήματα των αριθμητικών συγκρίσεων λέγονται comparison subgoals.

2.4.3 Αντιστοίχιση οντολογίας (Ontology mapping)

Η αντιστοίχιση οντολογίας είναι η διαδικασία με την οποία δύο οντολογίες συσχετίζονται σημασιολογικά σε εννοιολογικό επίπεδο, και τα στιγμιότυπα οντολογίας πηγής μετασχηματίζονται στις οντότητες οντολογίας στόχων σύμφωνα με εκείνες τις σημασιολογικές σχέσεις.

Υπάρχουν τρεις διαστάσεις της αντιστοίχισης οντολογίας:

1. Ανακάλυψη: με το χέρι, αυτόματα ή ημιαυτόματα καθορίζοντας τις σχέσεις μεταξύ των οντολογιών

2. Αναπαράσταση: Μια γλώσσα για να αντιπροσωπεύσει τις σχέσεις μεταξύ των οντολογιών
3. Εκτέλεσης: Μεταβάλλοντας ένα στιγμιότυπο μιας οντολογίας-πηγής σε ένα στιγμιότυπο οντολογίας-στόχο.

2.4.4 Οντολογίες και σημασιολογική ολοκλήρωση

Το πρόβλημα του να έρθουν κοντά ετερογενή και διανεμημένα συστήματα ηλεκτρονικών υπολογιστών είναι γνωστό ως πρόβλημα διαλειτουργικότητας (interoperability). Προκειμένου να επιτευχθεί η σημασιολογική διαλειτουργικότητα σε ένα ετερογενές σύστημα πληροφοριών, η έννοια των πληροφοριών που ανταλλάσσονται πρέπει να γίνει κατανοητή στα συστήματα. Οι σημασιολογικές συγκρούσεις εμφανίζονται όποτε δύο πλαίσια δεν χρησιμοποιούν την ίδια ερμηνεία των πληροφοριών. Τρεις κύριες αιτίες για τη σημασιολογική ετερογένεια είναι [WVV+01]:

1. Οι συγκρούσεις σύγχυσης (confounding conflicts) εμφανίζονται όταν τα στοιχεία πληροφοριών φαίνονται να έχουν την ίδια έννοια, αλλά διαφέρουν στην πραγματικότητα, π.χ. εξ αιτίας των διαφορετικών χρονικών πλαισίων.
2. Οι συγκρούσεις κλίμακας (scaling conflicts) εμφανίζονται όταν χρησιμοποιούνται διαφορετικά συστήματα αναφοράς για να μετρήσουν μια αξία. Παραδείγματα αποτελούν τα διαφορετικά νομίσματα.
3. Οι συγκρούσεις ονόματος (naming conflicts) εμφανίζονται όταν τα σχήματα ονομάτων των πληροφοριών διαφέρουν σημαντικά. Ένα συχνό φαινόμενο είναι η παρουσία ομωνύμων και συνωνύμων.

Η χρήση των οντολογιών για την ερμηνεία της υπονοούμενης και κρυμμένης γνώσης είναι μια πιθανή προσέγγιση για να υπερνικήσει το πρόβλημα της σημασιολογικής ετερογένειας. Η διαλειτουργικότητα είναι βασική εφαρμογή των οντολογιών, και πολλές προσεγγίσεις βασισμένες σε οντολογία έχουν αναπτυχθεί στην ολοκλήρωση πληροφοριών προκειμένου να επιτευχθεί η διαλειτουργικότητα.

2.4.5 Ρόλος των οντολογιών στη σημασιολογική ολοκλήρωση δεδομένων

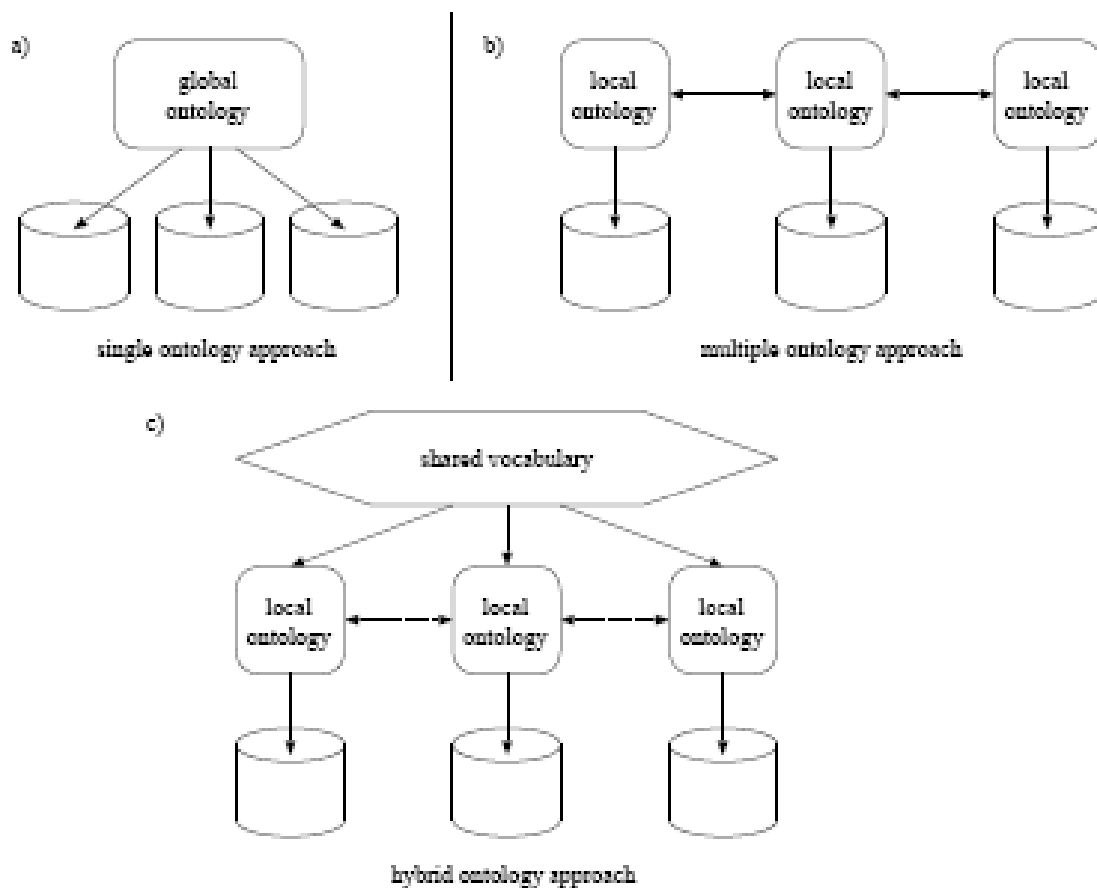
Η σημασιολογική ολοκλήρωση δεδομένων είναι η διαδικασία της χρήσης εννοιολογικών αναπαραστάσεων των δεδομένων και των σχέσεών τους με στόχο την εξάλειψη της ετερογένειας. Εφόσον οι οντολογίες επιτρέπουν την πολύπλοκη έκφραση εννοιών και των

σχέσεών τους, ο ρόλος τους στη σημασιολογική ολοκλήρωση δεδομένων είναι ιδιαίτερα ενεργός.

Μία οντολογία επιλέγεται ως καθολικό σχήμα διότι μπορεί να ορίσει πολύπλοκες σημασιολογικές σχέσεις ενός θεματικού χώρου. Παράλληλα, είναι διατυπωμένη σε αυστηρό μαθηματικό φορμαλισμό, ο οποίος επιτρέπει την εξαγωγή συμπερασμάτων, για παράδειγμα τον ορισμό επιπρόσθετων σχέσεων ανάμεσα στις έννοιες. Στο πλαίσιο αυτό έχουν αναπτυχθεί από ερευνητικές ομάδες συγκεκριμένες οντολογίες οι οποίες ορίζουν έννοιες και σχέσεις διαφόρων τομέων, π.χ. πολιτιστική κληρονομιά, υπολογιστική γλωσσολογία και γνωστική επιστήμη, οικονομικά.

Σε ένα σενάριο ολοκλήρωσης δεδομένων, υπάρχουν τρεις προσεγγίσεις σχετικές με το ρόλο των οντολογιών (οι αρχιτεκτονικές αυτές φαίνονται στο σχήμα 3) [WVV+01]:

1. **προσέγγιση απλής οντολογίας (single ontology approach):** Μία καθολική οντολογία (global ontology) παρέχει ένα διαμοιρασμένο λεξιλόγιο (shared vocabulary) για τον ορισμό των σημασιών αυτόνομων πηγών δεδομένων οι οποίες σχετίζονται με αυτήν. Η καθολική οντολογία περιγράφει ένα συγκεκριμένο τομέα, οπότε η συγκεκριμένη προσέγγιση εφαρμόζεται ιδιαίτερα σε πηγές δεδομένων που παρουσιάζουν διαφορετικές «όψεις» του ίδιου τομέα.
2. **προσέγγιση πολλαπλών οντολογιών (multiple ontology approach):** Στην προσέγγιση αυτή κάθε τοπικό σύστημα δεδομένων περιγράφεται από μία ξεχωριστή τοπική οντολογία (local ontology). Αντί να χρησιμοποιείται μία κοινή οντολογία, οι τοπικές οντολογίες συσχετίζονται μεταξύ τους. Η απουσία μίας καθολικής οντολογίας διευκολύνει την αυτόνομη ανάπτυξη τοπικών οντολογιών, οι οποίες εκφράζουν αναλυτικά και με συνέπεια τις έννοιες και τις σχέσεις κάθε τοπικού συστήματος δεδομένων. Εν τούτοις, το θετικό αυτό χαρακτηριστικό αποτελεί παράλληλα και ανοικτό ερευνητικό πρόβλημα κυρίως αναφορικά με τον *ορισμό των κανόνων αντιστοίχισης* μεταξύ των οντολογιών (ontology mapping)
3. **προσέγγιση υβριδικής οντολογίας (hybrid ontology approach):** Η συγκεκριμένη προσέγγιση συνδυάζει χαρακτηριστικά από τις δύο προαναφερθείσες προσεγγίσεις. Κάθε τοπική πηγή περιγράφεται από μία ξεχωριστή τοπική οντολογία, η οποία δεν είναι συσχετισμένη με άλλες τοπικές οντολογίες, αλλά με μία καθολική διαμοιραζόμενη οντολογία, και της οποίας η δημιουργία έχει βασιστεί είτε στις πρωτογενείς έννοιες (λεξιλόγιο) μίας καθολικής οντολογίας είτε στη μετατροπή της τοπικής πηγής σε οντολογία. Το γεγονός αυτό διευκολύνει την αντιστοίχιση των τοπικών οντολογιών και κατά συνέπεια των τοπικών πηγών δεδομένων. Νέες πηγές μπορούν εύκολα να προστίθενται χωρίς την ανάγκη αλλαγής των υπαρχουσών αντιστοιχίσεων.



Σχήμα 2.3 Προσεγγίσεις σχετικές με το ρόλο των οντολογιών [WVV+01]

Η απλή και η υβριδική προσέγγιση είναι κατάλληλες για την κατασκευή κεντρικών συστημάτων ολοκλήρωσης δεδομένων, η πρώτη μάλιστα είναι καταλληλότερη για GAV συστήματα και η δεύτερη για LAV. Ένα υβριδικό P2P σύστημα, όπου υπάρχει μία καθολική οντολογία σε έναν υπερ-κόμβο (super-peer), μπορεί να χρησιμοποιεί επίσης την προσέγγιση υβριδικής οντολογίας. Η προσέγγιση πολλαπλής οντολογίας μπορεί να χρησιμοποιηθεί καλύτερα για να κατασκευαστούν καθαρά P2P συστήματα ολοκλήρωσης δεδομένων, όπου δεν υπάρχουν υπερ-κόμβοι.

Μερικές προσεγγίσεις χρησιμοποιούν τις οντολογίες όχι μόνο για την ικανοποιημένη ερμηνεία, αλλά και είτε ως καθολικό μοντέλο ερώτησης είτε για την επαλήθευση της (καθορισμένης από το χρήστη ή από το σύστημα παραγόμενης) περιγραφής της ολοκλήρωσης. Έτσι, υπάρχουν κάποιοι πρόσθετοι ρόλοι των οντολογιών [WVV+01]:

1. **Μοντέλο ερωτήματος (query model):** Οι ολοκληρωμένες πηγές πληροφοριών παρέχουν κανονικά μια ολοκληρωμένη καθολική άποψη. Μερικές προσεγγίσεις ολοκλήρωσης χρησιμοποιούν την οντολογία ως καθολικό σχήμα ερώτησης.

2. **Επιβεβαίωση (verification):** Κατά τη διάρκεια της διαδικασίας ολοκλήρωσης διάφορες αντιστοιχίσεων (mappings) πρέπει να διευκρινιστούν από ένα καθολικό σχήμα στο τοπικό σχήμα πηγής. Η ακρίβεια τέτοιων αντιστοιχίσεων μπορεί να θεωρηθεί ικανώς βελτιωμένη εάν αυτές μπορούν να ελεγχθούν αυτόματα. Ένα υποερώτημα είναι σωστό αναφορικά με μια καθολική ερώτηση εάν το τοπικό υποερώτημα παρέχει ένα μέρος των ερωτημένων απαντήσεων, δηλ. τα υποερωτήματα πρέπει να περιληφθούν στην καθολική ερώτηση (αναχαίτιση ερώτησης). Δεδομένου ότι μια οντολογία περιέχει μια (πλήρη) προδιαγραφή της έννοιας, οι αντιστοιχίσεις μπορούν να ελεγχθούν σχετικά με τις οντολογίες.

Αναγνωρίζουμε τις ακόλουθες πέντε χρήσεις των οντολογιών στην ολοκλήρωση δεδομένων:

1. Αναπαράσταση μεταδεδομένων (metadata representation): Τα μεταδεδομένα (π.χ. σχήματα πηγών) σε κάθε πηγή δεδομένων μπορούν να αναπαρασταθούν σαφώς από μία τοπική οντολογία, χρησιμοποιώντας μία απλή γλώσσα.
2. Καθολική εννοιολογική αναπαράσταση (global conceptualization): Η καθολική οντολογία παρέχει μία εννοιολογική όψη πάνω στα σχηματικά ετερογενή σχήματα πηγών.
3. Υποστήριξη για υψηλού επιπέδου ερωτήματα (support for high-level queries): Δοθείσας μίας υψηλού επιπέδου όψης των πηγών, όπως παρέχεται από μία καθολική οντολογία, ο χρήστης μπορεί να σχηματίσει ένα ερώτημα χωρίς ειδική γνώση των διαφορετικών πηγών δεδομένων. Το ερώτημα επαναδιατυπώνεται στη συνέχεια σε ερωτήματα πάνω στις πηγές, βασιζόμενο στις σημασιολογικές αντιστοιχίσεις μεταξύ των καθολικών και των τοπικών αντιστοιχίσεων.
4. Δηλωτική μεσολάβηση (declarative mediation): Η επεξεργασία των ερωτημάτων σε ένα υβριδικό P2P σύστημα χρησιμοποιεί την καθολική οντολογία σαν έναν δηλωτικό μεσολαβητή (mediator) για την επαναδιατύπωση ερωτημάτων μεταξύ των κόμβων.
5. Υποστήριξη αντιστοίχισης (mapping support): Ένας θησαυρός, τυποποιημένος στα πλαίσια μιας οντολογίας, μπορεί να χρησιμοποιηθεί για τη διαδικασία αντιστοίχισης για να διευκολύνει την αυτοματοποίησή της.

Η πλειοψηφία των τοπικών πηγών δεν είναι δομημένη σύμφωνα με τις γλώσσες οντολογιών αλλά σε XML μορφή ή σε σχεσιακό μοντέλο (βάσεις δεδομένων). Ως συνέπεια, δεν είναι δυνατή η άμεση αντιστοίχισή τους με τις κλάσεις και τις ιδιότητες της καθολικής οντολογίας και απαιτούνται διαδικασίες μετατροπής.

Τα τελευταία χρόνια, ένας διαρκώς αυξανόμενος όγκος δεδομένων και μεταδεδομένων δομείται και διατίθεται σύμφωνα με τη γλώσσα κωδικοποίησης XML. Στόχος αυτής της επιλογής αποτελεί η ευκολότερη έκφραση, διάθεση, μεταφορά και αποθήκευση των

δεδομένων. Παράλληλα, η χρήση της XML ως γλώσσας έκφρασης δεδομένων λύνει προβλήματα ετερογένειας στη κωδικοποίηση δεδομένων (ετερογένεια στη σύνταξη). Στη διεθνή ερευνητική βιβλιογραφία παρατηρείται ιδιαίτερη ενασχόληση με τη δημιουργία κανόνων αντιστοίχισης ανάμεσα σε XML τοπικές πηγές και γλώσσες οντολογιών, όπως είναι η RDF και η OWL, με στόχο τη σημασιολογική ολοκλήρωσή τους. Για την έκφραση των κανόνων αυτών χρησιμοποιούνται πλούσιες σημασιολογικά γλώσσες, π.χ. οι γλώσσες οντολογιών, καθώς και γλώσσες που προέρχονται από τις βάσεις δεδομένων όπως είναι η Datalog και η Frame Logic (F-Logic). Σε γενικές γραμμές μπορούμε να πούμε ότι υπάρχουν δύο διαφορετικές προσεγγίσεις για τη δημιουργία κανόνων αντιστοίχισης XML δεδομένων με γλώσσες οντολογιών [WVV+01].

Η πρώτη προσέγγιση αντιμετωπίζει τους κανόνες αντιστοίχισης με «διαδικαστικό» (operational) τρόπο. Συγκεκριμένα, οι κανόνες αντιστοίχισης στην προσέγγιση αυτή περιγράφουν τη διαδικασία μετατροπής των XML δεδομένων στις κλάσεις και τις ιδιότητες μίας οντολογίας.

Η δεύτερη προσέγγιση αντιμετωπίζει τους κανόνες συσχέτισης με «δηλωτικό» (declarative) τρόπο. Πιο αναλυτικά, οι κανόνες αντιστοίχισης στην προσέγγιση αυτή περιγράφουν τη (σημασιολογική) σχέση που πρέπει να υπάρχει μεταξύ των XML δεδομένων και των κλάσεων και ιδιοτήτων μίας οντολογίας, χωρίς να παρέχουν κατά ανάγκη και κάποια διαδικασία μετασχηματισμού. Η δεύτερη προσέγγιση φαίνεται να είναι πλέον αξιοποιήσιμη σε χώρους που έχουν ήδη αναπτυχθεί εννοιολογικά μοντέλα για το σημασιολογικό τους ορισμό.

3

Περιγραφή και ανάλυση του προβλήματος

Στο κεφάλαιο αυτό, αναλύεται το πρόβλημα που διαπραγματεύεται η διπλωματική.

3.1 Γενικό πλαίσιο του προβλήματος

Τα κατανεμημένα περιβάλλοντα αυτόνομων κόμβων προσφέρουν μια ευέλικτη αρχιτεκτονική υποστήριξης ανταλλαγής δεδομένων στον Ιστό. Δημοφιλές παράδειγμα είναι τα δίκτυα ομότιμων κόμβων (peer-to-peer systems). Ένα δίκτυο ομότιμων κόμβων αποτελείται από ένα σύνολο αυτόνομων υπολογιστικών κόμβων στο Διαδίκτυο, οι οποίοι συνεργάζονται με σκοπό την ανταλλαγή δεδομένων. Τα συστήματα ομότιμων κόμβων που χρησιμοποιούνται ευρέως σήμερα κυρίως για την ανταλλαγή αρχείων μουσικής, έχουν πολύ μικρές δυνατότητες διαχείρισης δεδομένων. Η αναζήτηση πληροφορίας γίνεται με χρήση λέξεων κλειδιών (keyword-based search). Η ανάγκη λοιπόν για πιο εκφραστικές λειτουργίες, σε συνδυασμό με την ανάπτυξη του Σημασιολογικού Ιστού, οδήγησε στα συστήματα ομότιμων κόμβων με δεδομένα δομημένα με βάση κάποιο σχήμα (schema-based peer-to-peer systems), όπως θα περιγραφεί παρακάτω.

Στα συστήματα αυτά κάθε κόμβος χρησιμοποιεί ένα σχήμα βάσης δεδομένων με βάση το οποίο οργανώνει τα τοπικά διαθέσιμα δεδομένα. Το ζήτημα λοιπόν που προκύπτει είναι πώς θα μπορούν οι κόμβοι να αναζητούν και να ανταλλάσσουν πληροφορίες μεταξύ τους,

διατηρώντας όμως την αυτονομία τους. Για τη λύση του προβλήματος αυτού υπάρχουν δύο προσεγγίσεις.

1. Η πρώτη προσέγγιση λύσης του προβλήματος είναι η απαίτηση ύπαρξης ενός κεντρικού σχήματος το οποίο θα χρησιμοποιούν όλοι οι κόμβοι. Οι ερωτήσεις διατυπώνονται και αποτιμούνται με βάση το ίδιο σχήμα. Μια τέτοια προσέγγιση θα ήταν καλή για περιβάλλοντα με καθορισμένα και περιορισμένα γεωγραφικά και διοικητικά όρια, όπως για παράδειγμα το τοπικό δίκτυο ενός οργανισμού ή μιας εταιρίας. Όμως σε ένα ανοιχτό περιβάλλον, όπως είναι ο Παγκόσμιος Ιστός, χρειάζεται ένα πιο ευέλικτο μοντέλο που να επιτρέπει την χρήση πολλών διαφορετικών σχημάτων.
2. Σύμφωνα με τη δεύτερη προσέγγιση, κάθε κόμβος χρησιμοποιεί ένα δικό του, συνήθως διαφορετικό από των άλλων, σχήμα βάσης δεδομένων με βάση το οποίο οργανώνει τα τοπικά διαθέσιμα δεδομένα. Κάθε κόμβος μπορεί να λάβει πληροφορίες από τη δική του βάση δεδομένων, θέτοντας ερωτήματα (queries) σ' αυτήν. Αυτό γίνεται επειδή γνωρίζει το σχήμα, δηλαδή τον τρόπο με τον οποίο είναι δομημένα τα δεδομένα στη βάση δεδομένων. Κάτι τέτοιο δεν μπορεί να γίνει μεταξύ διαφορετικών κόμβων. Για να λυθεί το πρόβλημα αυτό της ετερογένειας των δεδομένων ενός δικτύου ομοτίμων κόμβων, χρησιμοποιούνται κανόνες αντιστοίχισης (mapping rules) μεταξύ των διαφορετικών σχημάτων, μέσω των οποίων γίνεται μετάφραση/μετασχηματισμός των ερωτημάτων (query translation/reformulation). Δηλαδή, ένα ερώτημα που διατυπώνεται με βάση ένα σχήμα μεταφράζεται με τη βοήθεια των κανόνων αντιστοίχισης σε ερώτημα που μπορεί να αποτιμηθεί από ένα άλλο γειτονικό σχήμα. Έτσι, το αρχικό ερώτημα διαδίδεται στο δίκτυο μεταξύ των γειτονικών κόμβων μέχρι ένα ορισμένο βάθος. Σε κάθε βήμα της διαδρομής, το ερώτημα μεταφράζεται στο σχήμα του νέου κόμβου, με τη βοήθεια των αντίστοιχων αντιστοιχίσεων των γειτόνων. Εκτός από τις πολλαπλές μεταφράσεις (λόγω διάδοσης του ερωτήματος) που κάνουν την αποτίμηση ενός ερωτήματος χρονοβόρα και μη βέλτιστη, ένα άλλο πρόβλημα είναι η δυσκολία προσαρμογής του συστήματος σε μεταβολές του δικτύου (π.χ. προσθαφαιρέσεις κόμβων), οι οποίες στα P2P δίκτυα είναι συχνές. Για παράδειγμα, κάθε φορά που εισέρχεται ένας νέος κόμβος πρέπει να ενημερωθούν όλοι οι γείτονές του για τις αντιστοιχίσεις, πράγμα που μειώνει την απόδοση του συστήματος, αν λάβουμε υπόψη ότι οι αντιστοιχίσεις αυτές δημιουργούνται χειρωνακτικά.

Στην εργασία αυτή, θα προτείνουμε μία λύση του προβλήματος της ανταλλαγή δεδομένων μεταξύ κόμβων, με διαφορετικά τοπικά σχήματα, ώστε η μετάφραση και αποτίμηση των ερωτημάτων να γίνεται πιο αποδοτικά.

3.2 Αναλυτική περιγραφή προβλήματος – Ορισμοί εννοιών

Στην ενότητα αυτή περιγράφουμε πιο συγκεκριμένα το πρόβλημα και δίνουμε ορισμούς και περιγραφές κάποιων εννοιών που θα χρησιμοποιηθούν στη συνέχεια..

3.2.1 Δομή βάσεων δεδομένων στους κόμβους (peers)

Θεωρούμε ένα **δίκτυο ομότιμων κόμβων (peer-to-peer system)**, το οποίο είναι επίπεδο (flat), δηλαδή δεν έχει υπερκόμβους. Σε κάθε κόμβο υπάρχει μία αυτόνομη βάση δεδομένων.

Τα δεδομένα στις βάσεις δεδομένων είναι οργανωμένα σύμφωνα με το σχεσιακό μοντέλο (relational model). Κάθε βάση δεδομένων έχει ένα **σχήμα (schema)** που περιέχει **σχέσεις (relations)**. Κάθε σχέση περιέχει ένα αριθμό **ιδιοτήτων (attributes)**, κάποιο ή κάποια από τα οποία αποτελούν το **κλειδί (key)** της σχέσης. Το κλειδί μιας σχέσης αποτελείται από μία ή περισσότερες ιδιότητες οι οποίες ορίζουν μονοσήμαντα μία εγγραφή στη βάση δεδομένων. Αυτό σημαίνει ότι το κλειδί κάθε εγγραφής είναι μοναδικό. Στη συνέχεια, θα αναφερόμαστε συνολικά στις σχέσεις και τις ιδιότητες ενός σχήματος βάσης δεδομένων με τον όρο **στοιχεία (elements)** του σχήματος.

Μία σχέση με τις ιδιότητες και το κλειδί της αναπαρίσταται ως εξής:

Relation1 (Attribute1, ..., AttributeX, AttributeY, ..., AttributeN)

Η λέξη πριν την παρένθεση είναι μία σχέση του σχήματος, ενώ αυτές μέσα στην παρένθεση οι ιδιότητες. Οι υπογραμμισμένες λέξεις αποτελούν το κλειδί της σχέσης αυτής.

Παράδειγμα 3.1: σχήμα μιας βάσης δεδομένων με πληροφορίες για ένα πανεπιστήμιο (σχήμα S1)

Θεωρούμε το σχήμα της βάσης δεδομένων ενός κόμβου του δικτύου. Η βάση αυτή περιέχει πληροφορίες για ένα πανεπιστήμιο. Η μορφή του σχήματος είναι:

Σχήμα S1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Salary, Responsibility)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

HasCourse (SID, CID, Grade)

Name (FirstName, Surname, CourseName)

3.2.2 Το πρόβλημα της ετερογένειας για την αποτίμηση ερωτημάτων

Κάθε κόμβος λαμβάνει πληροφορίες από τη βάση δεδομένων του, θέτοντας ερωτήματα (queries) σ' αυτήν. Τα ερωτήματα με τα οποία θα ασχοληθούμε είναι ερωτήματα SPJ (select-project-join) και έχουν τη μορφή:

```
SELECT {attribute1,...,attributeN}
FROM {relation1▷◁...▷◁relationM}
WHERE {condition1,..., conditionK}
```

Οι κόμβοι του δικτύου αυτού περιέχουν διαφορετικά μεταξύ τους σχήματα βάσεων δεδομένων. Όπως φαίνεται από το παραπάνω ερώτημα, κάθε κόμβος μπορεί να εξάγει πληροφορίες από τη δική του μόνο βάση δεδομένων, εφόσον δεν γνωρίζει το σχήμα της βάσης δεδομένων κάθε άλλου κόμβου, για να θέσει τα κατάλληλα ερωτήματα. Επειδή, όμως, οι κόμβοι έχουν ανάγκη την μεταξύ τους ανταλλαγή πληροφοριών, απαιτείται η εύρεση ενός τρόπου για την εξάλειψη αυτής της ετερογένειας.

3.2.3 Συσχετίσεις και αντιστοιχίσεις μεταξύ στοιχείων των σχημάτων

Για την εξάλειψη της ετερογένειας θα χρησιμοποιήσουμε τις συσχετίσεις (correspondences) και τις αντιστοιχίσεις (mappings), οι οποίες καθορίζονται από τον διαχειριστή (administrator) του συστήματος.

3.2.3.1 Συσχετίσεις (correspondences)

Μία **συσχέτιση (correspondence)** είναι μία διμερής σχέση, τα μέλη της οποίας αποτελούνται από στοιχεία ενός ή περισσότερων σχημάτων. Θεωρούμε τα ακόλουθα είδη συσχετίσεων:

1. $E1 \xrightarrow{ISA} E2$: κατευθυνόμενη συσχέτιση που δηλώνει ότι το στοιχείο E1 κληρονομεί από το στοιχείο E2.
2. $E1 \xleftarrow{ID} E2$: μη κατευθυνόμενη συσχέτιση που δηλώνει ότι τα στοιχεία E1 και E2 αντιπροσωπεύουν την ίδια έννοια.

3. $E1 \xrightarrow{HASA} E2$: κατευθυνόμενη συσχέτιση που δηλώνει ότι το στοιχείο E2 είναι μέρος του E1. Για παράδειγμα, μία σχέση συνδέεται με όλες τις ιδιότητές της με συσχέτιση HASA.
4. $E1 \xrightarrow{Related} E2$: κατευθυνόμενη συσχέτιση που δηλώνει ότι το στοιχείο E1 σχετίζεται με το στοιχείο E2. Χρησιμοποιείται σε περιπτώσεις, όπου το E2 είναι μέρος ενός συνόλου από στοιχεία και το E1 δεν σχετίζεται με όλα τα στοιχεία αυτά. Η χρήση του θα φανεί στη συνέχεια.

Φυσικά δεν χρειάζεται όλες οι πιθανές σωστές συσχετίσεις μεταξύ των στοιχείων των σχημάτων να οριστούν. Αυτό συμβαίνει, επειδή οι συσχετίσεις χρησιμοποιούνται προκειμένου να διευκρινίσουν εννοιολογικές σχέσεις. Επίσης, συνδέονται με τις αντιστοιχίες, που θα περιγράψουμε παρακάτω. Για παράδειγμα, οι διαχειριστές μπορεί να ορίσουν ένα κανόνα που δηλώνει ότι όλες οι λέξεις κλειδιά έχουν την ίδια σημασία σε δύο γείτονες, εκτός αν δηλωθεί διαφορετικά. Επομένως, δεν χρειάζεται να ορίσουμε την συσχέτιση $Grade \xleftarrow{ID} Grade$, γιατί είναι περιττό: αυτές είναι οι ασήμαντες συσχετίσεις ταυτοποίησης που μπορούν να υπονοηθούν από τις αντιστοιχίες.

Παράδειγμα 3.2: παράδειγμα συσχετίσεων για το σχήμα του παραδείγματος 1 (σχήμα S1)

Για το σχήμα S1, μερικές συσχετίσεις θα μπορούσαν να είναι οι ακόλουθες:

Faculty \xrightarrow{ISA} Person
 Student \xrightarrow{ISA} Person
 Staff \xrightarrow{ISA} Person
 Publication \xleftarrow{ID} Book
 Person \xrightarrow{HASA} ID
 Person \xrightarrow{HASA} Name
 Person \xrightarrow{HASA} Role
 Person \xrightarrow{HASA} Address
 Person \xrightarrow{HASA} Phone
 Person \xrightarrow{HASA} Email
 Person \xrightarrow{HASA} Sex
 Person \xrightarrow{HASA} Age

...

Person $\xrightarrow{\text{Related}}$ FirstName

Person $\xrightarrow{\text{Related}}$ Surname

Course $\xrightarrow{\text{Related}}$ CourseName

Παράδειγμα 3.3: παράδειγμα συσχετίσεων μεταξύ των διαφορετικών σχημάτων S1 και S2

Έστω, τώρα ότι υπάρχει το σχήμα S2 σε έναν γειτονικό κόμβο του κόμβου με το σχήμα S1:

Σχήμα S2

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Μερικές συσχετίσεις μεταξύ των σχημάτων S1 και S2 μπορεί να είναι οι εξής:

Faculty $\xrightarrow{\text{ISA}}$ Person

Course.Name $\xleftarrow{\text{ID}}$ Course.CName

Student $\xleftarrow{\text{ID}}$ Student1

Student $\xleftarrow{\text{ID}}$ Student2

Course $\xleftarrow{\text{ID}}$ Teached

3.2.3.2 Ερωτήματα (queries)

Ένα SPJ **ερώτημα (query)**, σαν αυτό που αναφέρθηκε στην παράγραφο 3.2.2, δηλαδή της μορφής:

```
SELECT {attribute1,...,attributeN}  
FROM {relation1 ▷◁ ... ▷◁ relationM}  
WHERE {condition1,..., conditionK}
```

μπορεί να αναπαρασταθεί ως συζευκτικό ερώτημα (conjunctive query):

relation1(attribute1,..., attributeX) ▷◁ ... ▷◁ relationM(attributeX,..., attributeN),
condition1,..., conditionK

ή

Q(attribute1,..., attributeX) :- relation1(attribute1,..., attributeX), ... ,
relationM(attributeX,..., attributeN),
condition1,..., conditionK

σύμφωνα με τη γλώσσα Datalog, που περιγράφηκε στην παράγραφο 2.4.2.5, όπου τα relation1, ..., relationM είναι τα subgoals και τα condition1,..., conditionK είναι τα comparison subgoals.

Παράδειγμα 3.4: παράδειγμα ενός ερωτήματος για το σχήμα του παραδείγματος 1

Για το σχήμα του παραδείγματος 1, ένα SPJ ερώτημα, το οποίο ζητάει την εμφάνιση των ιδιοτήτων ID, Name και Role της σχέσης Person μόνο των εγγραφών για τις οποίες η ιδιότητα Role έχει τιμή Faculty, είναι το εξής:

```
SELECT {ID,Name,Role}
FROM {Person}
WHERE {Role = Faculty}
```

Το ερώτημα αυτό μπορεί να αναπαρασταθεί ως εξής:

Q(ID, Name, Role) :- Person (ID, Name, Role), Role = Faculty

3.2.3.3 Αντιστοιχίσεις (mappings)

Μία **αντιστοιχίση (mapping)** είναι μία διμερής σχέση, τα μέλη της οποίας είναι τα σώματα των ερωτημάτων (queries) εκφραζόμενα σε γλώσσα Datalog. Έτσι, μία αντιστοιχίση έχει τη μορφή: query1 :- query2.

Παράδειγμα 5: παράδειγμα αντιστοιχίσεων των σχημάτων S1 και S2

Μερικές αντιστοιχίσεις των σχημάτων S1 και S2 μπορεί να είναι οι εξής:

Person (ID, Name, Role), Role = Faculty :- Faculty (FID, Name, _, _, _)

Person (ID, Name, Role), Role = Faculty :- Faculty (FID, Name, _, _, _)

Person (ID, Name, Role), Role = Staff :- Staff (ID, Name, _, _)

Person (ID, Name, Role), Role = Student :- Student (SID, Name, _)

Person (Name, __, __, __, __, __, __), Name = Name.FirstName :- Name (FirstName, __, __)

Person (Name, __, __, __, __, __, __), Name = Name.Surname :- Name (Surname, __, __)

Course (Name, __, __, __, __, __, __), Name = Name.CourseName :- Name (CourseName, __, __)

Publication (Title, FID, Description) :- Book (Title, FID, Description)

3.2.4 Εννοιολογικός γράφος

Στο σημείο αυτό θα παραθέσουμε μερικά στοιχεία της θεωρίας των γράφων, που θα μας φανούν χρήσιμα στη συνέχεια.

Γράφος (graph) (ή γράφημα) ονομάζεται ένα διατεταγμένο ζεύγος συνόλων (V, E) , όπου V είναι μη κενό σύνολο στοιχείων και E ένα σύνολο μη διατεταγμένων ζευγών του V . V είναι το σύνολο των **κόμβων (nodes)** ή κορυφών και E το σύνολο των **ακμών (edges)** ή πλευρών του γράφου.

Ένας γράφος ονομάζεται **κατευθυνόμενος (directed)** αν οι ακμές του έχουν προσανατολισμό, δηλαδή κατευθύνονται από ένα κόμβο, ο οποίος ονομάζεται **πηγή (source)** σε έναν άλλο, ο οποίος ονομάζεται **προορισμός (target)**.

Δύο κόμβοι ονομάζονται **γείτονες (acquaintances)** αν υπάρχει ακμή που να τους ενώνει.

Ορίζουμε ως **συγχώνευση (merging)** μία διαδικασία που παίρνει ως είσοδο 2 γράφους και ένα αριθμό σημασιολογικών συσχετίσεων μεταξύ των κόμβων και επιστρέφει ένα γράφο, που περιέχει κόμβους κι από τους δύο γράφους, συγχωνευμένους κατάλληλα, σύμφωνα με τις σημασιολογικές συσχετίσεις. Ο γράφος που προκύπτει ονομάζεται **γράφος συγχώνευσης**.

Τέλος, ορίζουμε την έννοια του **εννοιολογικού γράφου (conceptual graph)**. Ο εννοιολογικός γράφος είναι ένας κατευθυνόμενος γράφος, οι ακμές του οποίου συνδέουν σημασιολογικά τους κόμβους που ενώνουν. Αποτελεί μια μορφή οντολογίας, με τη βοήθεια της οποίας απεικονίζουμε ένα σχήμα βάσης δεδομένων. Κάθε κόμβος του γράφου αναπαριστά κάποια σχέση ή ιδιότητα του σχήματος. Οι ακμές, που ενώνουν δύο κόμβους, έχουν κάποιο τύπο, που δηλώνει τη σημασιολογική σχέση που φέρουν. Ο τύπος των ακμών μπορεί να είναι: HASA, ID, ISA και Related, όπως τα είδη των συσχετίσεων. Έτσι, οι ακμές απεικονίζουν τη σημασιολογική συσχέτιση μεταξύ των κόμβων που ενώνουν. Με αυτό τον τρόπο, μπορούμε να χειριστούμε ευκολότερα τα σχήματα με τα στοιχεία και τις σημασιολογικές τους συσχετίσεις.

4

Ολοκλήρωση σχημάτων με υποστήριξη οντολογιών

Στο κεφάλαιο αυτό, θα περιγράψουμε τη διαδικασία που προτείνουμε για την επίλυση του προβλήματος, που ορίσαμε στο προηγούμενο κεφάλαιο.

4.1 Γενικά

Για τη λύση του προβλήματος που περιγράφηκε στο προηγούμενο κεφάλαιο, δηλαδή την ανταλλαγή δεδομένων μέσω ερωτημάτων μεταξύ κόμβων, όπου καθένας έχει το δικό του **τοπικό σχήμα** (local schema) βάσης δεδομένων, προτείνουμε ένα τρόπο εξαγωγής ενός **κεντρικού ή καθολικού σχήματος G** (global schema) για το σύνολο των κόμβων αυτών. Τα τοπικά αυτά σχήματα αποτελούν υποσύνολα-όψεις (views) του κεντρικού σχήματος. Δεν απαιτείται η εκ των προτέρων ύπαρξη ενός κεντρικού σχήματος το οποίο χρησιμοποιούν όλοι οι κόμβοι, όπως απαιτεί η πρώτη προσέγγιση που αναφέραμε στην παράγραφο 3.1. Το κεντρικό σχήμα προκύπτει από την ενοποίηση των τοπικών σχημάτων με τη βοήθεια κανόνων αντιστοίχισης, οι οποίοι ορίζουν σημασιολογικές σχέσεις μεταξύ των τοπικών σχημάτων.

Με τη βοήθεια του καθολικού σχήματος, μπορούμε να αποτιμήσουμε ερωτήματα από ένα κόμβο προς κάθε άλλο. Τα ερωτήματα δεν διαδίδονται πλέον μεταξύ των σχημάτων, όπως απαιτεί η δεύτερη προσέγγιση της παραγράφου 3.1. Αντίθετα, εκφράζονται στο καθολικό σχήμα και στη συνέχεια μεταφράζονται σε ερωτήματα στα σχήματα των επιμέρους κόμβων που θα τα απαντήσουν. Η μετάφραση αυτή γίνεται με χρήση κάποιων τελικών αντιστοιχίσεων μεταξύ του κεντρικού σχήματος και του κάθε τοπικού, τις οποίες ορίζει ο διαχειριστής. Τέλος, επιστρέφεται το αποτέλεσμα στον κόμβο που έθεσε το ερώτημα, προσαρμοσμένο, μέσω των αντιστοιχίσεων, στο σχήμα του κόμβου αυτού.

Έτσι, δεν χρειάζεται ένας κόμβος να έχει κρατάει αντιστοιχίσεις με κάθε γείτονά του, ώστε να διατυπώσει τα κατάλληλα ερωτήματα για να αποσπάσει απ' αυτούς τα επιθυμητά δεδομένα και δεν έχουμε διαδοχικές μεταφράσεις. Επίσης, το σύστημα ολοκλήρωσης αυτό είναι πιο ευέλικτο σε μεταβολές του δικτύου, π.χ. εισαγωγής νέων κόμβων. Σε τέτοιες περιπτώσεις, δεν είναι ανάγκη να ενημερωθούν όλοι οι γειτονικοί κόμβοι για τις αλλαγές, ενημερώνεται μόνο το καθολικό σχήμα.

Συνεπώς, με τη βοήθεια του καθολικού σχήματος η μετάφραση και αποτίμηση των ερωτημάτων, και γενικά η διαδικασία της ανταλλαγής πληροφοριών μεταξύ των κόμβων γίνεται πιο αποδοτικά και το δίκτυο είναι πιο ευέλικτο.

4.2 Σύντομη περιγραφή

Αρχικά, κάνουμε τις εξής υποθέσεις για τα σχήματα των κόμβων (peers):

1. Οι αντιστοιχίσεις μεταξύ των κόμβων είναι GAV/LAV και εκφράζονται σαν συζευκτικά ερωτήματα (conjunctive queries).
2. Τα σχήματα των κόμβων έχουν επουσιώδεις εσωτερικές αντιστοιχίσεις, όπως σχέσεις εξωτερικών κλειδιών.
3. Μεταξύ δύο κόμβων υπάρχουν επιπρόσθετες συσχετίσεις (correspondences) μεταξύ των στοιχείων (σχέσεις ή ιδιότητες) των σχημάτων.
4. Όσον αφορά ένα σχήμα:
 - a. Προτιμάμε λιγότερες σχέσεις (relations) με περισσότερες ιδιότητες (attributes), παρά περισσότερες σχέσεις με λιγότερες ιδιότητες.
 - b. Η συνάφεια μεταξύ δύο σχέσεων είναι ανάλογη με το πλήθος των αντιστοιχίσεων μεταξύ των συνόλων των ιδιοτήτων τους.

- c. Θεωρούμε καλύτερες σχέσεις που δεν αναμένεται να περιέχουν πολλές κενές τιμές (null values) (π.χ. δεν προέρχονται από συγχώνευση σχέσεων με διαφορετικές ιδιότητες).
 - d. Δύο σχέσεις, των οποίων τα κλειδιά είναι πλήρως αντιστοιχισμένα, θεωρούνται περισσότερο συσχετιζόμενες, π.χ. είναι και οι δύο προβολές μίας άλλης σχέσης.
5. Υπάρχει μία προκαθορισμένη σταθερά που αντιπροσωπεύει το μέγιστο αριθμό σχέσεων που το σχήμα επιτρέπεται να έχει.
 6. Υπάρχει μία προκαθορισμένη σταθερά που αντιπροσωπεύει το μέγιστο διάστημα για την αποθήκευση τελικού σχήματος.

Για την κατασκευή του κεντρικού σχήματος από τα τοπικά ακολουθήσαμε τα παρακάτω βήματα, τα οποία θα εξετάσουμε αναλυτικότερα στη συνέχεια:

1. Χρήση του μοντέλου γράφου για την αναπαράσταση των τοπικών σχημάτων ΒΔ:
Για διευκόλυνση ενοποίησης των τοπικών σχημάτων ΒΔ, αναπαριστούμε κάθε σχήμα με ένα εννοιολογικό κατευθυνόμενο γράφο, ώστε να αξιοποιήσουμε την ευκολία χειρισμών που μας προσφέρουν οι γράφοι.
2. Χειρωνακτική προσθήκη κανόνων αντιστοίχισης (mapping rules) που συσχετίζουν τα συστατικά των σχημάτων ΒΔ:
Ο διαχειριστής (administrator) του συστήματος ολοκλήρωσης δημιουργεί συσχετίσεις μεταξύ στοιχείων ενός ή περισσότερων σχημάτων και αντιστοιχίσεις μεταξύ στοιχείων διαφορετικών σχημάτων.
3. Αυτόματη εξαγωγή σημασιολογικών σχέσεων από τα σχήματα ΒΔ και τους κανόνες αντιστοίχισης:
Από τα σχήματα και τις αντιστοιχίσεις παράγονται συσχετίσεις ανάμεσα στα στοιχεία των σχημάτων. Η εξαγωγή αυτή των σημασιολογικών σχέσεων γίνεται σύμφωνα με συγκεκριμένους κανόνες, που θα παρουσιαστούν στη συνέχεια.
4. Ενοποίηση (merging) των οντολογιών σε μια κεντρική οντολογία:
Η ενοποίηση των σχημάτων γίνεται μέσω ενοποίησης (συγχώνευσης) των εννοιολογικών γράφων τους, που αποτελούν μία μορφή οντολογίας, σύμφωνα με συγκεκριμένους κανόνες, που θα περιγράψουν στη συνέχεια.
5. Εξαγωγή κεντρικού σχήματος G που αντιστοιχεί στην κεντρική οντολογία και αποτελεί την ολοκλήρωση (integration) των τοπικών σχημάτων ΒΔ των κόμβων:
Για την κατασκευή του κεντρικού σχήματος, εξετάζουμε τον καθολικό γράφο, δηλαδή το γράφο που έχει προκύψει από την ενοποίηση των τοπικών γράφων, και από αυτόν εξάγουμε το κεντρικό σχήμα, με τρόπο που θα περιγραφεί παρακάτω.

Το κεντρικό σχήμα που παράγεται με την παραπάνω διαδικασία μπορεί να έχει πολύ μεγάλο μέγεθος. Για το λόγο αυτό είναι πολλές φορές επιθυμητό να το υποβάλλουμε σε **συμπίεση (compression)**.

Ως μετρική του μεγέθους ενός σχήματος ΒΔ θα θεωρήσουμε τις σχέσεις (relations) του σχήματος της βάσης δεδομένων. Η συμπίεση γίνεται αφαιρώντας σχέσεις, ώστε ο συνολικός αριθμός τους να μην ξεπερνάει ένα ορισμένο κατώφλι (threshold).

Η συμπίεση του καθολικού σχήματος γίνεται μέσω του καθολικού γράφου, ο οποίος συμπίεζεται σύμφωνα με συγκεκριμένους κανόνες, ώστε οι σχέσεις που αφαιρούνται να έχουν λιγότερη σημασία από τις υπόλοιπες, δηλαδή να μη χάνεται σημαντική πληροφορία κατά την αφαίρεση τους. Οι κανόνες αυτοί διευκρινίζονται παρακάτω.

4.3 Αναλυτική περιγραφή και αλγόριθμοι για την ολοκλήρωση σχημάτων

4.3.1 Αναπαράσταση σχήματος από γράφο

Για την αναπαράσταση, δημιουργούμε έναν κόμβο για κάθε σχέση και κάθε ιδιότητα του σχήματος. Οι ακμές του γράφου αυτού αντιπροσωπεύουν συσχετίσεις μεταξύ των κόμβων που ενώνουν. Οπότε, δημιουργούμε μία ακμή για κάθε συσχέτιση που υπάρχει ανάμεσα σε δύο κόμβους.

4.3.2 Συσχετίσεις (correspondences) και κανόνες αντιστοίχισης (mappings rules)

Κατ' αρχήν, από ένα σχήμα παράγονται HASA συσχετίσεις, και συγκεκριμένα ανάμεσα σε μία σχέση και κάθε ιδιότητά της. Συνήθως, σχέσεις ή ιδιότητες με παρόμοιο όνομα ταυτίζονται σημασιολογικά, οπότε δημιουργείται μία συσχέτιση ID. Αυτό, βέβαια, έγκειται στην κρίση του διαχειριστή του συστήματος, όπως και οι υπόλοιπες συσχετίσεις και αντιστοιχίσεις.

4.3.3 Αυτόματη εξαγωγή σημασιολογικών σχέσεων από τους κανόνες αντιστοίχισης

Η εξαγωγή συσχετίσεων από τις αντιστοιχίσεις γίνεται σύμφωνα με τον ακόλουθο αλγόριθμο:

1. Αν τα δύο μέλη της αντιστοίχισης έχουν σχέσεις με τα ίδια ονόματα και ίδια κλειδιά, μεταξύ των σχέσεων αυτών δημιουργείται συσχέτιση τύπου ID.
2. Αν κάποια σχέση ενός μέλους του κανόνα αντιστοίχισης έχει το ίδιο κλειδί με μία σχέση του άλλου μέλους και υπάρχει, σε ένα από τα δύο μέλη, κάποια συνθήκη της μορφής: ιδιότητα = τιμή ιδιότητας (string), τότε δημιουργείται συσχέτιση τύπου ISA προς τη σχέση που βρίσκεται στο μέλος που υπάρχει η συνθήκη αυτή.
3. Αν οι σχέσεις στα δύο μέλη του κανόνα αντιστοίχισης έχουν ίδιες όλες τις ιδιότητες, τότε δημιουργείται μεταξύ τους συσχέτιση τύπου ID.
4. Αν κάποιο μέλος έχει μία συνθήκη, όπου τα δύο μέλη της (B και Γ) είναι ιδιότητες σχέσεων (A και B) που βρίσκονται αντίστοιχα στα δύο μέλη της αντιστοίχισης, και η ιδιότητα του πρώτου μέλους της συνθήκης (B) έχει το ίδιο όνομα με σχέση του δεύτερου μέλους της αντιστοίχισης, τότε δημιουργείται συσχέτιση τύπου Related μεταξύ της σχέσης (A), που περιέχει την ιδιότητα B, και της ιδιότητας Γ.

π.χ. $A(B), A.B = B.\Gamma :- B(\Gamma) \rightarrow A \xrightarrow{Related} \Gamma$

Κατά τη δημιουργία μίας νέας συσχέτισης, γίνεται έλεγχος αν αυτή έρχεται σε αντίθεση με τις ήδη υπάρχουσες. Αν μία συσχέτιση ανάμεσα σε δύο στοιχεία υπάρχει ήδη, δεν ξαναδημιουργείται, ενώ αν δύο στοιχεία συνδέονται σε αρχική συσχέτιση διαφορετικού είδους απ' αυτή που δημιουργήθηκε από την αντιστοίχιση, υπερισχύει η μία, σύμφωνα με την εξής σειρά φθίνουσας προτεραιότητας: ID, ISA, Related, HASA.

4.3.4 Ενοποίηση εννοιολογικών γράφων

Τα βήματα του αλγορίθμου που ακολουθείται για την ενοποίηση των εννοιολογικών γράφων είναι τα ακόλουθα:

1. Συγχώνευση γράφων σχημάτων:
Οι γράφοι των τοπικών σχημάτων περιέχουν μόνο ακμές τύπου HASA. Γίνεται συγχώνευση των κόμβων με το ίδιο όνομα (ίδιο όνομα σχέσεων ή ιδιοτήτων που αντιστοιχούν στους κόμβους) και προσαρμόζονται κατάλληλα οι ακμές HASA.
2. Προσθήκη ακμών τύπου Related σύμφωνα με τις αντίστοιχες συσχετίσεις μεταξύ των σχημάτων.
3. Συγχώνευση κόμβων, που ταυτίζονται σημασιολογικά σύμφωνα με τις ID συσχετίσεις, και κατάλληλη προσαρμογή των ακμών τους.

4. Προσθήκη ακμών τύπου ISA σύμφωνα με τις αντίστοιχες συσχετίσεις. Εκτός από την προσθήκη ακμών, αφαιρούνται οι ακμές που συνδέουν την πηγή μιας ακμής τύπου ISA με κόμβους οι οποίοι συνδέονται με τον προορισμό αυτής της ISA ακμής.

4.3.5 Εξαγωγή κεντρικού σχήματος

Η κατασκευή του κεντρικού σχήματος από τον καθολικό γράφο, που προτείνουμε είναι η εξής:

1. Για κάθε κόμβο που έχει προέρθει από σχέση (relation), δημιουργούμε μία σχέση του κεντρικού σχήματος με το ίδιο όνομα.
2. Για τον κόμβο κάθε σχέσης που δημιουργήθηκε, ελέγχουμε τις HASA ακμές, στις οποίες είναι πηγή και δημιουργούμε για κάθε προορισμό ομώνυμη ιδιότητα για τη συγκεκριμένη σχέση του κεντρικού σχήματος. Όσες ιδιότητες ήταν κλειδιά στη συγκεκριμένη σχέση σε κάποιο από τα τοπικά σχήματα, τίθεται σαν κλειδί και στη σχέση του κεντρικού σχήματος.
3. Για κάθε σχέση του κεντρικού σχήματος, ελέγχουμε αν ο αντίστοιχος κόμβος είναι πηγή ISA ακμής. Αν αυτό ισχύει, η πηγή αυτή αποκτάει το κλειδί του προορισμού.
4. Αν υπάρχει κάποια ιδιότητα B μίας σχέσης A, το οποίο έχει το ίδιο όνομα με κάποια σχέση Γ του καθολικού σχήματος, η ιδιότητα αυτή (B) αντικαθίσταται με το κλειδί της αντίστοιχης ομώνυμης σχέσης Γ. Στην περίπτωση που η σχέση A συνδέεται με σχέσεις Related με κάποιες ιδιότητες της σχέσης Γ, η ιδιότητα B αντικαθίσταται μόνο με τις ιδιότητες αυτές.

4.4 Συμπίεση κεντρικού σχήματος

Όπως αναφέραμε στην παράγραφο 4.2, η συμπίεση του καθολικού σχήματος γίνεται μέσω του καθολικού γράφου, ο οποίος συμπιέζεται σύμφωνα με τον ακόλουθο αλγόριθμο:

1. Αφαιρούμε σχέσεις, των οποίων ο αντίστοιχος κόμβος στο γράφο έχει εισερχόμενες ακμές ISA. Αυτή η επιλογή, έναντι των HASA ακμών, γίνεται επειδή η αφαίρεση HASA συσχετίσεων θα προκαλούσε αλλοίωση του νοηματικού περιεχομένου του σχήματος. Δηλαδή θα υπήρχε απώλεια πληροφοριών των ΒΔ, εφόσον η αφαίρεση HASA συσχετίσεων συνεπάγεται αφαίρεση ιδιοτήτων από σχέσεις.
2. Από τις σχέσεις με κόμβο με εισερχόμενες ακμές ISA, αφαιρούνται κατά προτίμηση αυτές που δεν έχουν κόμβο με εισερχόμενες ακμές HASA.

3. Από τις σχέσεις με κόμβους χωρίς εισερχόμενες ακμές HASA (ή σε περίπτωση που δεν υπάρχουν τέτοιες, από τις σχέσεις με εισερχόμενες ακμές HASA), αφαιρούνται κατά προτίμηση αυτές με τις λιγότερες ISA ακμές. Σε περίπτωση που οι κόμβοι δύο σχέσεων έχουν ίδιο αριθμό εισερχομένων ακμών ISA, αφαιρούνται κατά προτίμηση οι σχέσεις με τον μικρότερο αριθμό ιδιοτήτων.

Στη συνέχεια, από τον συμπιεσμένο, πλέον, καθολικό γράφο, εξάγουμε το συμπιεσμένο καθολικό σχήμα, με τη διαδικασία που περιγράφηκε στην παράγραφο 4.3.5.

Η διαδικασία αυτή της συμπίεσης επαναλαμβάνεται, αφαιρώντας μία σχέση κάθε φορά. Μετά το τέλος κάθε επανάληψης γίνεται έλεγχος, και αν ο αριθμός των σχέσεων δεν υπερβαίνει πλέον το κατώφλι ή αν δεν μπορεί να γίνει άλλη συμπίεση στον γράφο σύμφωνα με τους παραπάνω κανόνες, η συμπίεση σταματάει.

4.5 Τελικές αντιστοιχίες

Με το καθολικό σχήμα που έχουμε δημιουργήσει μπορούμε να αποτιμήσουμε ερωτήματα από ένα κόμβο προς κάθε άλλο. Αυτό γίνεται με τη βοήθεια κάποιων τελικών αντιστοιχίσεων, οι οποίες, όπως αναφέραμε και στην παράγραφο 4.1, συνδέουν τις σχέσεις κάθε τοπικού σχήματος με το καθολικό. Έτσι, όταν ένας κόμβος θέτει ένα ερώτημα στο καθολικό σχήμα, αυτό αποτιμάται σε όλους τους κόμβους με τη βοήθεια των τελικών αυτών αντιστοιχίσεων. Τέλος, επιστρέφεται το αποτέλεσμα στον κόμβο που έθεσε το ερώτημα, προσαρμοσμένο, και πάλι μέσω των τελικών αντιστοιχίσεων, στο σχήμα του κόμβου αυτού.

Οι τελικές αντιστοιχίες κατασκευάζονται από τον διαχειριστή του συστήματος και προέρχονται από τις αρχικές συσχετίσεις και αντιστοιχίες που χρησιμοποιήθηκαν για την κατασκευή του καθολικού σχήματος.

Για κάθε σχέση κάθε τοπικού σχήματος δημιουργείται μία αντιστοίχιση που τη συνδέει με την αντίστοιχη σχέση του καθολικού σχήματος. Η κατασκευή των τελικών αυτών αντιστοιχίσεων γίνεται με τους εξής κανόνες:

1. Για απλές σχέσεις (χωρίς ISA και Related συσχετίσεις):

Μία σχέση ενός τοπικού σχήματος αντιστοιχίζεται με αυτή του καθολικού που έχει το ίδιο όνομα (default ID συσχέτιση) ή με αυτή που έχει το ίδιο όνομα με σχέση άλλου αρχικού σχήματος με την οποία υπάρχει αρχική ID συσχέτιση. Επίσης, οι ιδιότητες της σχέσης του καθολικού σχήματος που συμμετέχουν στην αντιστοίχιση είναι αυτές που συνδέονται με ID συσχέτιση με αυτές του τοπικού σχήματος.

π.χ.

Αν S1 και S2 είναι τοπικά σχήματα με σχέσεις S1.A(a, b, c) και S2.B(a, b, d, e, f) με συσχετίσεις $A \xleftarrow{ID} B$ και $c \xleftarrow{ID} e$, και η αντίστοιχη σχέση του καθολικού σχήματος G είναι G.A(a, b, c, d, f), τότε οι τελικές αντιστοιχίες είναι

S1.A(a, b, c) :- G.A(a, b, c, _, _) και S2.B(a, b, d, e, f) :- G.A(a, b, e, d, f)

2. Για σχέσεις που έχουν ISA συσχετίσεις:

Κάθε σχέση A ενός τοπικού σχήματος που αποτελεί πρώτο μέλος ISA συσχέτισης, δηλαδή κληρονομεί από μία σχέση B ($A \xrightarrow{ISA} B$), αντιστοιχίζεται με το σύνδεμο (join) της σχέσης του καθολικού σχήματος που έχει το ίδιο όνομα A και της σχέσης του καθολικού σχήματος που έχει το ίδιο όνομα με αυτήν από την οποία κληρονομεί (B).

π.χ.

Αν S1 και S2 είναι τοπικά σχήματα με σχέσεις S1.A(a, b, c, d) και S1.B(e, c, d, f, g) με συσχετίσεις $A \xrightarrow{ISA} B$, και οι αντίστοιχες σχέσεις του καθολικού σχήματος G είναι G.A(a, b, e) και G.B(e, c, d, f, g), τότε οι τελικές αντιστοιχίες είναι

S1.A(a, b, c, d) :- G.A(a, b, e), G.B(e, c, d, _, _) και

S1.B(e, c, d, f, g) :- G.B(e, c, d, f, g)

3. Για σχέσεις που έχουν Related συσχετίσεις:

Για κάθε σχέση A ενός τοπικού σχήματος που αποτελεί πρώτο μέλος Related συσχέτισης δηλαδή ($A \xrightarrow{Related} B$), δημιουργείται αντιστοίχιση με την αντίστοιχη σχέση του καθολικού σχήματος, όπου, όμως, η ιδιότητα του τοπικού που ευθύνεται για τη συσχέτιση, έχει κάποια σχέση με το B, πο

π.χ.

Αν σε ένα σχήμα υπάρχουν οι σχέσεις:

S1.Person(ID, Name, Address),

S1.Course(CID, Name, Semester),

S1.Name(FirstName, Surname, CourseName)

και οι συσχετίσεις:

Person $\xrightarrow{Related}$ FirstName,

Person $\xrightarrow{Related}$ Surname,

Course $\xrightarrow{Related}$ CourseName

και στο καθολικό υπάρχουν οι σχέσεις:

G.Person(ID, FirstName, Surname, Address),

G.Course(CID, CourseName, Semester),

G.Name(FirstName, Surname, CourseName),

τότε οι τελικές αντιστοιχίες είναι

S1.Person (ID, Name, Address) :- G.Person (ID, Address, FirstName, Surname),

Name = FirstName append SurName

S1. Course(CID, Name, Semester) :- G. Course(CID, Name, Semester)

Με όμοιο τρόπο δημιουργούνται και οι τελικές αντιστοιχίες με το συμπιεσμένο καθολικό σχήμα.

4.6 Παραδείγματα

Παράδειγμα 4.1:

Θεωρούμε το ακόλουθο παράδειγμα ενός κατανεμημένου περιβάλλοντος αυτόνομων κόμβων με βάσεις δεδομένων πανεπιστημίων, όπως αυτό που χρησιμοποιήσαμε στο προηγούμενο κεφάλαιο. Έστω ότι δύο κόμβοι έχουν αντίστοιχα τα παρακάτω σχήματα:

Σχήμα S1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Salary, Responsibility)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

HasCourse (SID, CID, Grade)

Name (FirstName, Surname, CourseName)

Σχήμα S2

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Για τα σχήματα αυτά κατασκευάζουμε τις ακόλουθες συσχετίσεις (correspondences):

Faculty \xrightarrow{ISA} Person

Student \xrightarrow{ISA} Person

Staff \xrightarrow{ISA} Person

Course.Name \xleftarrow{ID} Course.CName

Student \xleftarrow{ID} Student1

Student \xleftarrow{ID} Student2

Course \xleftarrow{ID} Teached

Και τις παρακάτω αντιστοιχίσεις (mappings):

Person (ID, Name, Role), Role = Faculty :- Faculty (FID, Name, _, _, _, _)

Person (ID, Name, Role), Role = Staff :- Staff (ID, Name, _, _)

Person (ID, Name, Role), Role = Student :- Student (SID, Name, _)

Person (Name, _, _, _, _, _, _), Name = Name.FirstName :- Name (FirstName, _, _)

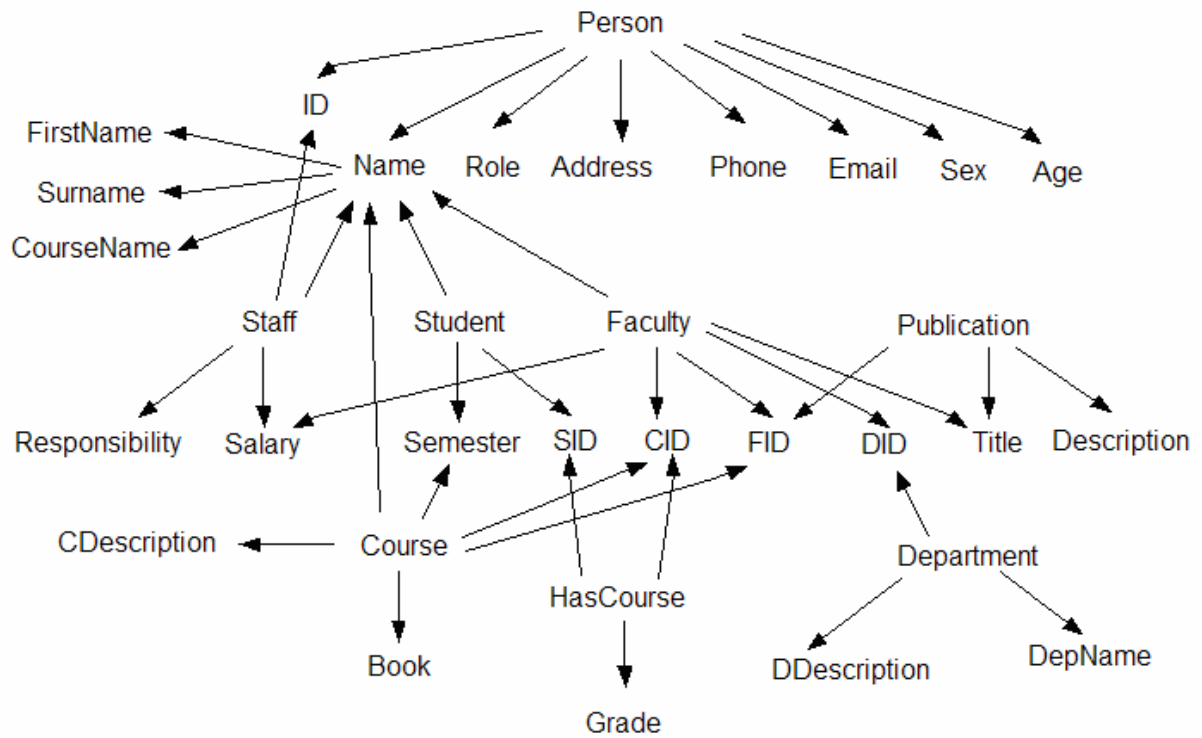
Person (Name, _, _, _, _, _, _), Name = Name.Surname :- Name (Surname, _, _)

Course (Name, _, _, _, _, _, _), Name = Name.CourseName :- Name (CourseName, _, _)

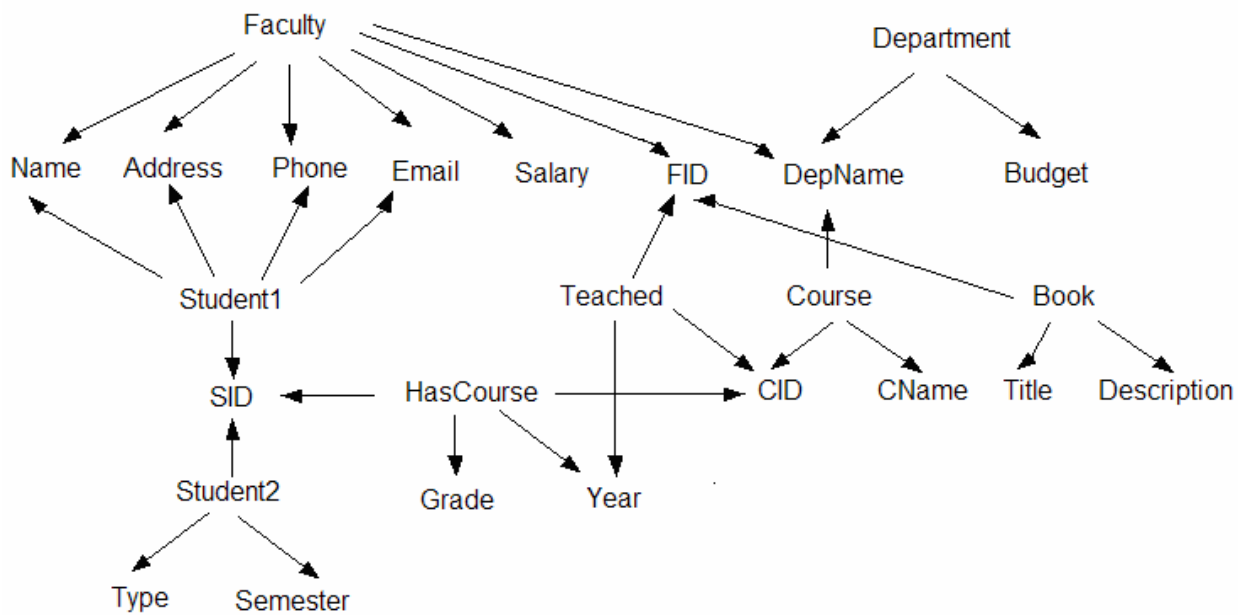
Publication (Title, FID, Description) :- Book (Title, FID, Description)

Για συντομία, οι συσχετίσεις και οι αντιστοιχίσεις αυτές δεν αναφέρουν ανάμεσα στις σχέσεις ποιων σχημάτων είναι. Αυτό γίνεται γιατί ορίζονται για τις σχέσεις όλων των σχημάτων με τα αντίστοιχα ονόματα.

Για τα σχήματα αυτά προκύπτουν οι εξής εννοιολογικοί γράφοι, στους οποίους έχουμε θεωρήσει μόνο τις HASA συσχετίσεις.



Σχήμα 4.1 Εννοιολογικός γράφος Σχήματος S1 με HASA συσχετίσεις



Σχήμα 4.2 Εννοιολογικός γράφος Σχήματος S2 με HASA συσχετίσεις

Από τις αντιστοιγήσεις εξάγονται αντίστοιχα οι ακόλουθες συσχετίσεις:

Person (ID, Name, Role), Role = Faculty :- Faculty (FID, Name, __, __, __, __)

→ Faculty \xrightarrow{ISA} Person

Person (ID, Name, Role), Role = Staff :- Staff (ID, Name, __, __)

→ Staff \xrightarrow{ISA} Person

Person (ID, Name, Role), Role = Student :- Student (SID, Name, __)

→ Student \xrightarrow{ISA} Person

Person (Name, __, __, __, __, __, __), Person.Name=Name.FirstName :- Name (FirstName, __, __)

→ Person $\xrightarrow{Related}$ FirstName

Person (Name, __, __, __, __, __, __), Person.Name = Name.Surname :- Name (Surname, __, __)

→ Person $\xrightarrow{Related}$ Surname

Course(Name, __, __, __, __, __, __), Person.Name = Name.CourseName :-

Name(CourseName, __, __)

→ Course $\xrightarrow{Related}$ CourseName

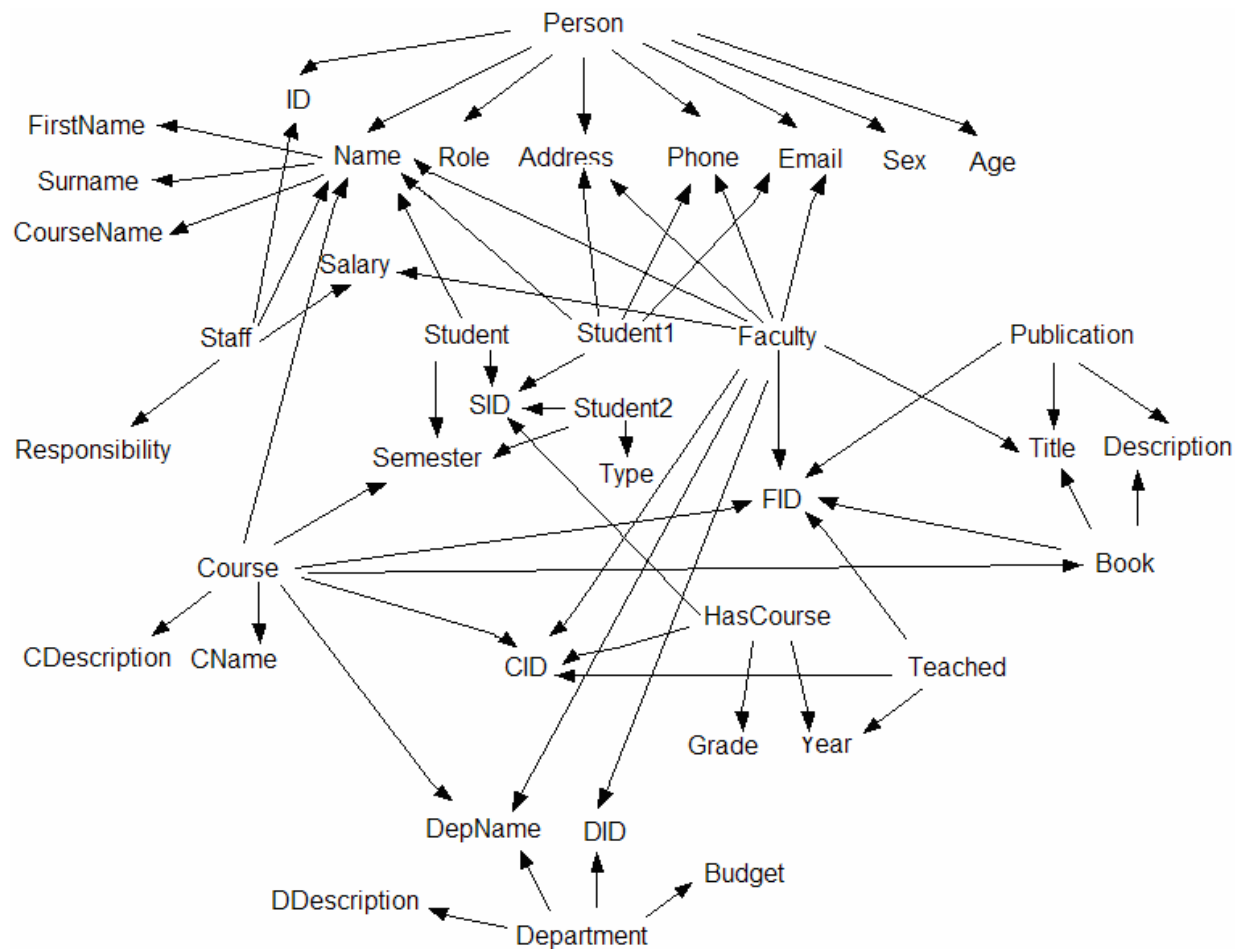
Publication (Title, FID, Description) :- Book (Title, FID, Description)

→ Publication \xleftarrow{ID} Book

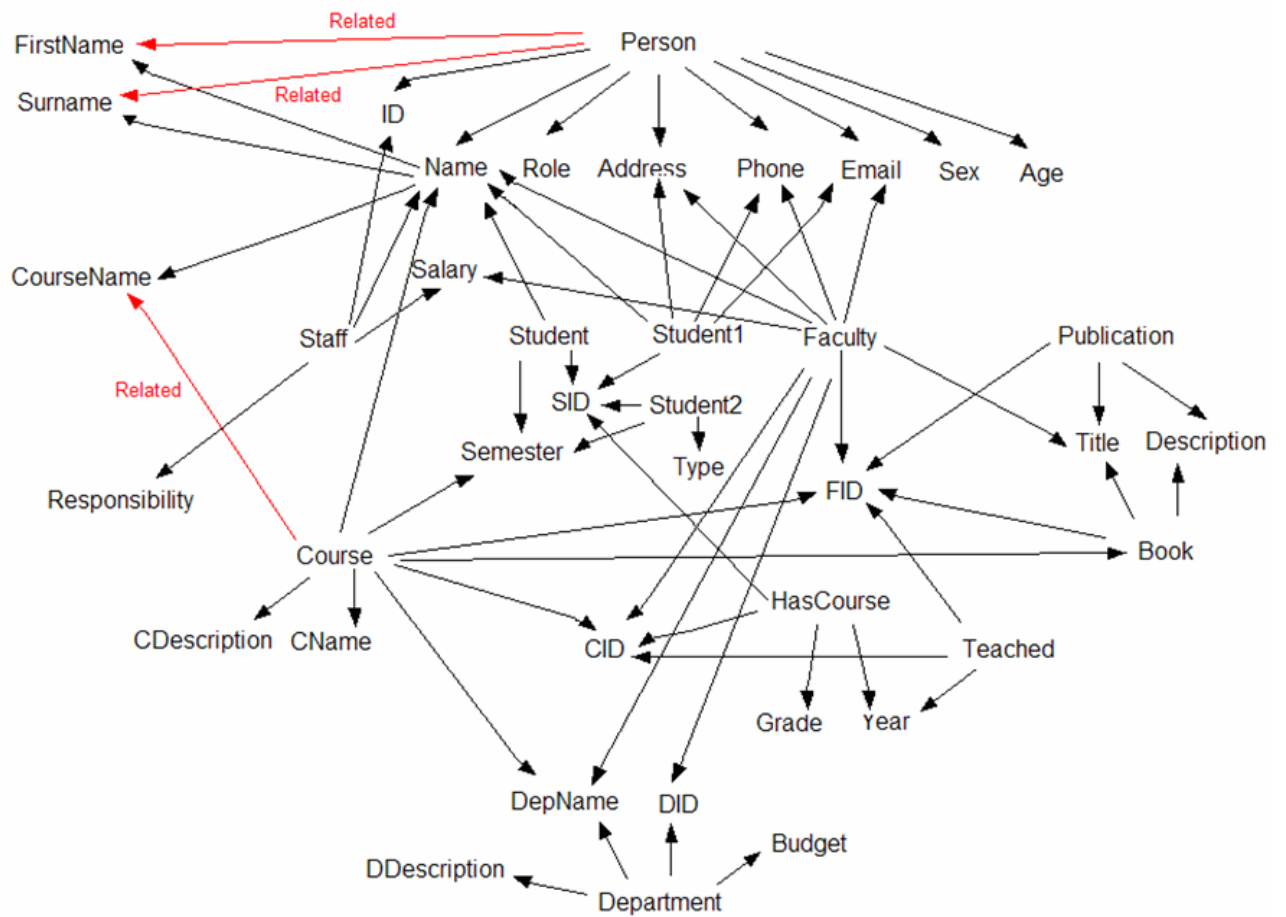
Αξίζει να σημειώσουμε ότι οι ISA συσχετίσεις προκύπτουν από τιμές που παίρνει μία ιδιότητα. Στο παράδειγμά μας, η ιδιότητα Role παίρνει τις τιμές Faculty, Staff, Student, οπότε η σχέση Person σχετίζεται με ISA με τις αντίστοιχες σχέσεις.

Παρατηρούμε, επίσης, ότι οι τρεις πρώτες συσχετίσεις που δημιουργούνται υπάρχουν ήδη, οπότε δεν ξαναδημιουργούνται.

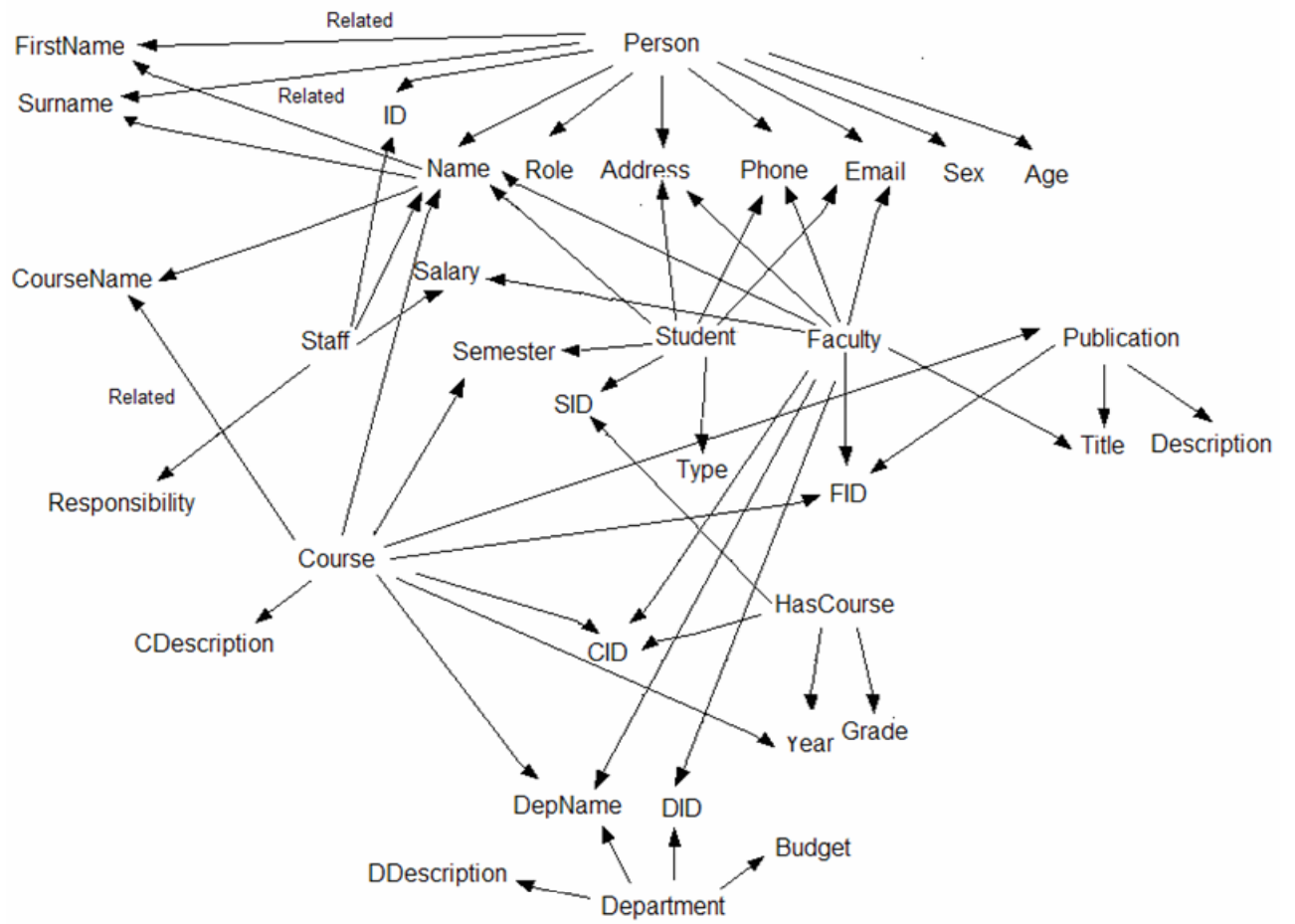
Η ενοποίηση των σχημάτων γίνεται σύμφωνα με τους κανόνες της παραγράφου 4.3.4. Για κάθε βήμα του αλγορίθμου κατασκευάζονται οι ακόλουθοι γράφοι. Στους γράφους αυτούς, στις ακμές που έχουν τύπο συσχέτισης HASA, αυτό δεν γράφεται πάνω τους, για απλοποίηση του σχήματος.



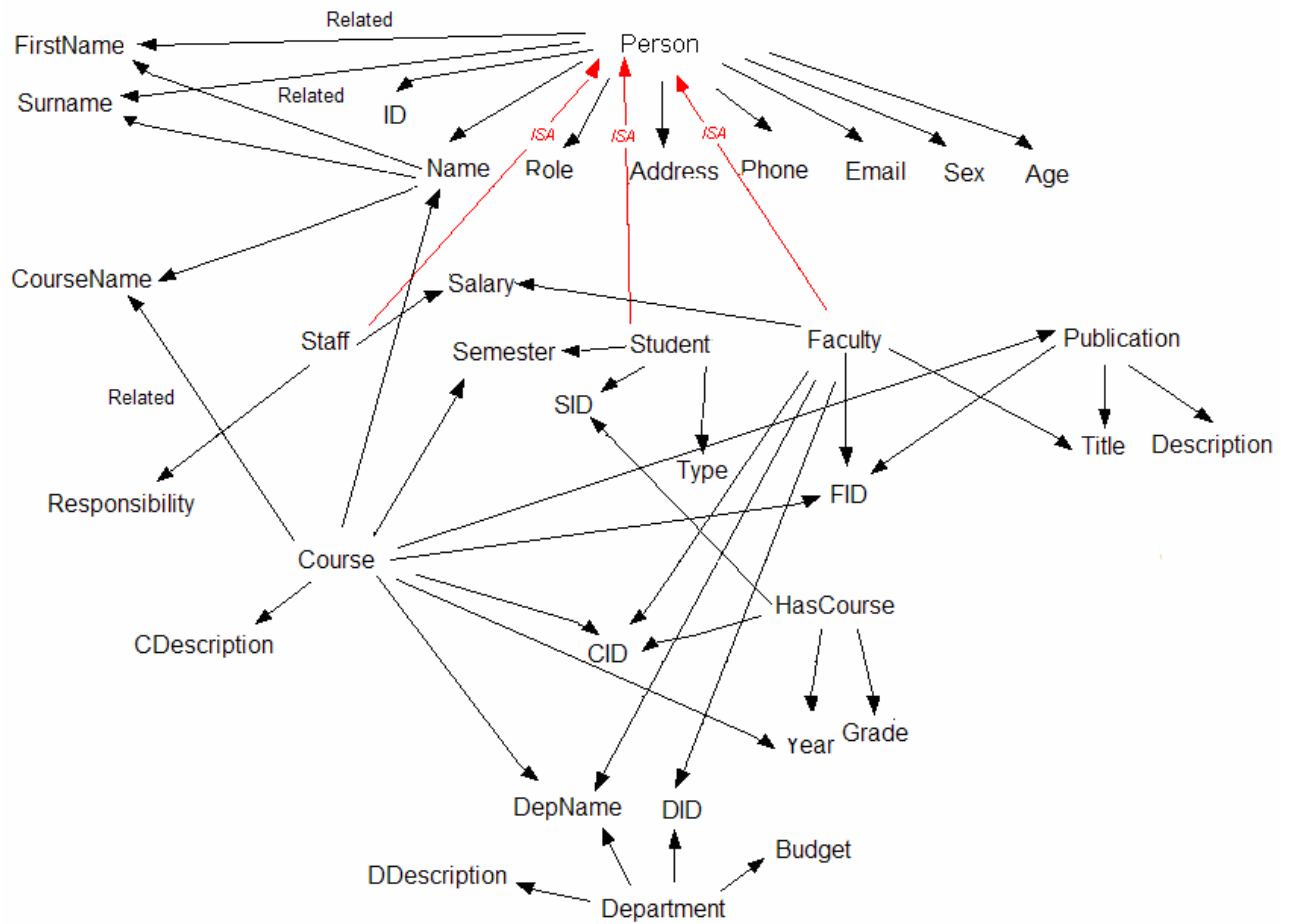
Σχήμα 4.3 Εννοιολογικός γράφος συγχώνευσης των σχημάτων S1 και S2
(βήμα 1 του αλγορίθμου 4.3.4)



**Σχήμα 4.4 Προσθήκη ακμών Related στον γράφο συγχώνευσης
(βήμα 2 του αλγορίθμου 4.3.4)**



**Σχήμα 4.5 Συγχώνευση στον γράφο συγχώνευσης κόμβων που σχετίζονται με ID
(βήμα 3 του αλγορίθμου 4.3.4)**



Σχήμα 4.6 Δημιουργία καθολικού γράφου με την προσθήκη ISA συσχετίσεων στον γράφο συγγώνευσης (βήμα 4 του αλγορίθμου 4.3.4)

Από τον καθολικό γράφο (σχήμα 4.6) του παραδείγματος μας προκύπτει το καθολικό (κεντρικό) σχήμα. Το σχήμα που παράγεται σε κάθε βήμα της παραπάνω μεθόδου κατασκευής είναι:

Global Schema (μετά τα βήματα 1&2)

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Name (FirstName, Surname, CourseName)

Faculty (FID, DID, CID, Title, Salary, DepName)

Student (SID, Semester, Type)

Staff (Salary, Responsibility)

Department (DID, DepName, DDescription, Budget)

Course (CID, Name, FID, Semester, CDescription, DepName, Year, Publication)

Publication (Title, FID, Description)

HasCourse (SID, CID, Grade, Year)

Εδώ δημιουργήθηκαν οι σχέσεις με τις ιδιότητες και τα κλειδιά τους.

Global Schema (μετά το βήμα 3)

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Name (FirstName, Surname, CourseName)

Faculty (FID, DID, CID, Title, Salary, DepName, ID)

Student (SID, Semester, Type, ID)

Staff (Salary, Responsibility, ID)

Department (DID, DepName, DDescription, Budget)

Course (CID, Name, FID, Semester, CDescription, DepName, Year, Publication)

Publication (Title, FID, Description)

HasCourse (SID, CID, Grade, Year)

Στο σημείο αυτό, έχουν προστεθεί τα κλειδιά των σχέσεων-προορισμών ακμών ISA στις σχέσεις-πηγές. Δηλαδή το κλειδί ID της σχέσης Person προστέθηκε στις σχέσεις Faculty, Student, Staff.

Global Schema (μετά το βήμα 4)

Person (ID, Role, Address, Phone, Email, Sex, Age, FirstName, Surname)

Name (FirstName, Surname, CourseName)

Faculty (FID, DID, CID, Title, Salary, DepName, ID)

Student (SID, Semester, Type, ID)

Staff (Salary, Responsibility, ID)

Department (DID, DepName, DDescription, Budget)

Course (CID, FID, Semester, CDescription, DepName, Year, CourseName, Title)

Publication (Title, FID, Description)

HasCourse (SID, CID, Grade, Year)

Τέλος, προστέθηκαν οι Related ιδιότητες στις αντίστοιχες σχέσεις. Δηλαδή η ιδιότητα Name στη σχέση Person αντικαταστήθηκε από τις ιδιότητες FirstName και Surname, ενώ η ίδια ιδιότητα στη σχέση Course αντικαταστήθηκε από την CourseName. Επίσης, οι ιδιότητες μιας σχέσης που αποτελούν σχέσεις (υπάρχει σχέση με το ίδιο όνομα) αντικαταστάθηκαν με το κλειδί των σχέσεων αυτών. Για παράδειγμα, η ιδιότητα Publication στη σχέση Course αντικαταστάθηκε από το κλειδί Title της σχέσης Publication.

Οι τελικές αντιστοιχίσεις μεταξύ των σχέσεων των τοπικών σχημάτων και του καθολικού είναι:

S1.Person (ID, Name, Role, Address, Phone, Email, Sex, Age) :-

G.Person (ID, Role, Address, Phone, Email, Sex, Age, FirstName, Surname),

Name = FirstName append Surname

S1.Faculty (FID, Name, DID, CID, Title, Salary) :-

G.Faculty (FID, DID, CID, Title, Salary, _, ID),

G.Person (ID, "Faculty", _, _, _, _, FirstName, Surname),

Name = FirstName append Surname

S1.Student (SID, Name, Semester) :-

G.Student (SID, Semester, _, ID),

G.Person (ID, "Student", _, _, _, _, FirstName, Surname),

Name = FirstName append Surname

S1.Staff (ID, Name, Salary, Responsibility) :-

G.Staff (Salary, Responsibility, ID),

G.Person (ID, “Staff”, _, _, _, _, FirstName, Surname),
 Name = FirstName append Surname

S1.Department (DID, DepName, DDescription) :-
 G.Department (DID, DepName, DDescription, _)

S1.Course (CID, CN, FID, Semester, CDescription, BT) :-
 G.Course (CID, FID, Semester, CDescription, _, _, CN, BT),

S1.Publication (Title, FID, Description) :- G.Publication (Title, FID, Description)

S1.HasCourse (SID, CID, Grade) :- G. HasCourse (SID, CID, Grade, _)

S1.Name(FirstName, Surname, CourseName) :- G.Name (FirstName, Surname, CourseName)

S2.Faculty (FID, Name, DepName, Address, Phone, Email, Salary) :-
 G.Faculty (FID, _, _, _, Salary, DepName, ID),
 G.Person (ID, “Faculty”, Address, Phone, Email, _, _, FirstName, Surname),
 Name = FirstName append Surname

S2.Student1(SID, Name, Address, Phone, Email) :-
 G.Student (SID, _, _, ID),
 G.Person (ID, “Student”, Address, Phone, Email, _, _, FirstName, Surname),
 Name = FirstName append Surname

S2.Student2 (SID, Type, Semester) :- G.Student (SID, Semester, Type, _)

S2.Course (CID, CN, DepName) :- G.Course (CID, _, _, _, DepName, _, CN, _)

S2.Teached (CID, FID, Year) :- G.Course (CID, FID, _, _, _, Year, _, _)

S2.HasCourse (CID, SID, Year, Grade) :- G.HasCourse (SID, CID, Grade, Year)

S2. Department (DepName, Budget) :- G.Department (_, DepName, _, Budget)

S2. Book (Title, FID, Description) :- G. Publication (Title, FID, Description)

Αν συμπιέσουμε το καθολικό σχήμα του παραδείγματός μας, το οποίο έχει 9 σχέσεις, θέτοντας το κατώφλι συμπίεσης ίσο με 7, παρατηρούμε ότι πραγματοποιείται μία μόνο συμπίεση. Αυτό συμβαίνει γιατί υπάρχει μόνο μία σχέση (Person) με εισερχόμενες ακμές ISA στον αντίστοιχο γράφο. Έτσι, αφαιρείται η σχέση Person και οι ιδιότητες της προστίθενται στις σχέσεις, των οποίων οι κόμβοι στον γράφο είναι οι πηγές των ISA ακμών που εισέρχονται στη σχέση Person. Το συμπιεσμένο καθολικό σχήμα που προκύπτει είναι το ακόλουθο:

Compressed Global Schema

Name (FirstName, Surname, CourseName)

Faculty (FID, DID, CID, Title, Salary, DepName, ID, Role, Address, Phone, Email, Sex, Age, FirstName, Surname)

Student (SID, Semester, Type, ID, Role, Address, Phone, Email, Sex, Age, FirstName, Surname)

Staff (Salary, Responsibility, ID, Role, Address, Phone, Email, Sex, Age, FirstName, Surname)

Department (DID, DepName, DDescription, Budget)

Course (CID, FID, Semester, CDescription, DepName, Year, CourseName, Title)

Publication (Title, FID, Description)

HasCourse (SID, CID, Grade, Year)

Παρατηρούμε ότι οι ιδιότητες Role, Address, Phone, Email, Sex, Age, FirstName και Surname της σχέσης Person προστέθηκαν στις σχέσεις Faculty, Student και Staff. Η ιδιότητα ID υπήρχε ήδη σ' αυτές.

Παράδειγμα 4.2:

Θεωρούμε το ακόλουθο παράδειγμα ενός καταναμημένου περιβάλλοντος αυτόνομων κόμβων με βάσεις δεδομένων νοσοκομείων. Έστω ότι δύο κόμβοι έχουν αντίστοιχα τα παρακάτω σχήματα:

Σχήμα S1

Hospital (HID, Name, Place, Capacity)

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, Name, Category, Description)

Σχήμα S2

Hospital (HID, HName, Place, Capacity)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)

Nurse (NID, ID, Name, HID, Salary)
InnerPatient (PID, ID, Name, NoHospitalisations)
OuterPatient (PID, ID, Name, NoVisits)
Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)
Illness (IID, IName, Category, Description)

Σχήμα S3

Hospital (Hospital_id, Address, Web-site)
Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)
Nurse (Nurse_id, Name, Salary, Age, Sex)
Staff (Staff_id, Duty, Name, Salary, Age, Sex)
Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)
Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Σχήμα S4

HospitalInfo (HID, Name, Type, Capacity)
HospitalContact (HID, Address, Web-site, Phone, Fax)
Doctor (DID, Name, SName, Salary, Speciality)
Nurse (NID, Name, SName, Salary)
Staff (SID, Name, Salary)
Patient (PID, Name, SName, History)
Sector (SName, Speciality, NoBeds, NoNurses)
Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Σχήμα S5

Hospital (HID, Name, Place, Employee)
Doctor (DID, Name, Speciality, Phone, Email)
Nurse (NID, Name)
Therapy (TID, ThName, ThDescription, Cost)
Illness (IID, IName, IDescription)
Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

Θεωρούμε τις ακόλουθες συσχετίσεις (correspondences):

Hospital.Name \xleftrightarrow{ID} Hospital.HName

Hospital.HID \xleftrightarrow{ID} Hospital.Hospital_id

Hospital.Address \xleftrightarrow{ID} Hospital.Place

Illness.Name \xleftrightarrow{ID} Illness.IName

Illness.IID \xleftrightarrow{ID} Illness.Illness_id

Illness.Category \xleftrightarrow{ID} Illness.IllnessCategory

Illness.Description \xleftrightarrow{ID} Illness.IDescription

Patient.PID \xleftrightarrow{ID} Patient.Patient_id

Doctor.DID \xleftrightarrow{ID} Doctor.Doctor_id

Nurse.NID \xleftrightarrow{ID} Nurse.Nurse_id

Staff.SID \xleftrightarrow{ID} Staff.Staff_id

Hospital \xleftrightarrow{ID} Hospital.HospitalInfo

Hospital \xleftrightarrow{ID} Hospital.HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Και τις παρακάτω αντιστοιχίσεις (mappings):

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Hospitalisation (PID, HID, DateOfHospitalisation, IID, DateOut, DID, Bed, Days)

Illness (IID, Name, Category, Description, Symptoms, TID)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, Speciality, SName, Email, ID, EID)

Nurse (NID, SName, ID, EID)

InnerPatient (NoHospitalisations, ID, PID)

OuterPatient (NoVisits, ID, PID)

Staff (Duty, SID, ID, EID)

Therapy (TID, ThName, ThDescription, Cost)

Sector (SName, Speciality, NoBeds, NoNurses)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

Αν συμπίεσουμε το καθολικό σχήμα, το οποίο έχει 16 σχέσεις, θέτοντας το κατώφλι συμπίεσης αρκετά μικρό, για παράδειγμα ίσο με 7, ώστε να γίνουν όλες οι δυνατές συμπίεσεις, παρατηρούμε ότι πραγματοποιούνται τρεις συμπίεσεις. Τα συμπεσμένα καθολικά σχήματα που προκύπτουν σε κάθε επανάληψη της διαδικασίας της συμπίεσης παρουσιάζονται παρακάτω.

1. 1^η Συμπίεση

Αρχικά, οι κόμβοι που έχουν εισερχόμενες ακμές τύπου ISA στον καθολικό γράφο και είναι υποψήφιες να αφαιρεθούν απ' αυτόν είναι οι: Person, Employee και Patient. Απ' αυτές μόνο οι Person και Patient δεν έχουν εισερχόμενες ακμές τύπου HASA. Οπότε, ο κόμβος Employee απορρίπτεται. Από τους δύο πλέον υποψήφιους για αφαίρεση κόμβους, ο Patient έχει λιγότερες εισερχόμενες ακμές ISA, οπότε είναι αυτός που θα αφαιρεθεί στην πρώτη επανάληψη. Οι ιδιότητες της αντίστοιχης σχέσης Patient προστίθενται στις σχέσεις, των οποίων οι κόμβοι στον γράφο είναι οι πηγές των ISA ακμών που εισέρχονται στην Patient. Συγκεκριμένα, οι ιδιότητες Hospitalisation_needed, DateOfHospitalisation, SName, History και IID προστίθενται στις σχέσεις InnerPatient και OuterPatient.

1st Compressed Global Schema

Hospital (HID, Name, Capacity, Address, Web-site, Type, Phone, Fax, EID, ID)

Employee (EID, HID, Job, Salary, ID)

Hospitalisation (PID, HID, DateOfHospitalisation, IID, DateOut, DID, Bed, Days)

Illness (IID, Name, Category, Description, Symptoms, TID)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, Speciality, SName, Email, ID, EID)

Nurse (NID, SName, ID, EID)

InnerPatient (NoHospitalisations, PID, Hospitalisation_needed, DateOfHospitalisation, SName, History, IID, ID)

OuterPatient (NoVisits, PID, Hospitalisation_needed, DateOfHospitalisation, SName, History, IID, ID)

Staff (Duty, SID, ID, EID)

Therapy (TID, ThName, ThDescription, Cost)

Sector (SName, Speciality, NoBeds, NoNurses)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

2. 2^η Συμπύεση

Στη συνέχεια, ελέγχεται το συμπιεσμένο καθολικό σχήμα, που προέκυψε από την πρώτη συμπύεση. Οι κόμβοι Person και Employee έχουν εισερχόμενες ακμές τύπου ISA, όμως μόνο ο Person δεν έχει και εισερχόμενο HASA. Οπότε είναι αυτός που θα αφαιρεθεί. Έτσι, οι ιδιότητες Name, Role, Address, Phone, Age και Sex της σχέσης Person προστίθενται στις σχέσεις Employee, Doctor, Nurse, InnerPatient, OuterPatient και Staff.

2nd Compressed Global Schema

Hospital (HID, Name, Capacity, Address, Web-site, Type, Phone, Fax, EID, ID)

Employee (EID, HID, Job, Salary, ID, Name, Role, Address, Phone, Age, Sex)

Hospitalisation (PID, HID, DateOfHospitalisation, IID, DateOut, DID, Bed, Days)

Illness (IID, Name, Category, Description, Symptoms, TID)

Doctor (DID, Speciality, SName, Email, ID, Name, Role, Address, Phone, Age, Sex, EID)

Nurse (NID, SName, ID, Name, Role, Address, Phone, Age, Sex, EID)

InnerPatient (NoHospitalisations, PID, Hospitalisation_needed, DateOfHospitalisation,

SName, History, IID, ID, Name, Role, Address, Phone, Age, Sex)
 OuterPatient (NoVisits, PID, Hospitalisation_needed, DateOfHospitalisation, SName,
 History, IID, ID, Name, Role, Address, Phone, Age, Sex)
 Staff (Duty, SID, ID, Name, Role, Address, Phone, Age, Sex, EID)
 Therapy (TID, ThName, ThDescription, Cost)
 Sector (SName, Speciality, NoBeds, NoNurses)
 Diagnosis (PID, Date, DID, MID, Comments)
 Medication (MID, PID, Date, TID, IID, MedDescription)
 Cures (IID, TID)

3. 3^η Συμπύεση

Για την επόμενη συμπύεση, μόνο ο κόμβος Employee έχει εισερχόμενη ακμή ISA και επομένως μπορεί να αφαιρεθεί. Ο κόμβος αυτός έχει, όμως, και εισερχόμενη ακμή HASA. Αλλά, επειδή δεν υπάρχει άλλος υποψήφιος για διαγραφή, αναγκαστικά να αφαιρέσουμε αυτόν. Έτσι, οι ιδιότητες HID, Job και Salary προστίθενται στις σχέσεις Doctor, Nurse και Staff.

3rd Compressed Global Schema

Hospital (HID, Name, Capacity, Address, Web-site, Type, Phone, Fax, EID, Job, Salary, ID,
 Role, Age, Sex)
 Hospitalisation (PID, HID, DateOfHospitalisation, IID, DateOut, DID, Bed, Days)
 Illness (IID, Name, Category, Description, Symptoms, TID)
 Doctor (DID, Speciality, SName, Email, ID, Name, Role, Address, Phone, Age, Sex, EID,
 HID, Job, Salary)
 Nurse (NID, SName, ID, Name, Role, Address, Phone, Age, Sex, EID, HID, Job, Salary)
 InnerPatient (NoHospitalisations, PID, Hospitalisation_needed, DateOfHospitalisation,
 SName, History, IID, ID, Name, Role, Address, Phone, Age, Sex)
 OuterPatient (NoVisits, PID, Hospitalisation_needed, DateOfHospitalisation, SName,
 History, IID, ID, Name, Role, Address, Phone, Age, Sex)
 Staff (Duty, SID, ID, Name, Role, Address, Phone, Age, Sex, EID, HID, Job, Salary)
 Therapy (TID, ThName, ThDescription, Cost)
 Sector (SName, Speciality, NoBeds, NoNurses)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

5

Υλοποίηση

Στο κεφάλαιο αυτό θα περιγράψουμε τον τρόπο με τον οποίο υλοποιήσαμε το σύστημα.

5.1 Πλατφόρμες και προγραμματιστικά εργαλεία

Η εργασία αυτή υλοποιήθηκε σε γλώσσα προγραμματισμού Java, έκδοση 1.5, στο περιβάλλον Eclipse, έκδοση 3.1. και σε λειτουργικό σύστημα Windows XP Professional. Ο υπολογιστής που χρησιμοποιήθηκε για την υλοποίηση έχει CPU Intel Pentium 4 στα 2.40GHZ και μνήμη RAM 512MB.

5.2 Λεπτομέρειες Υλοποίησης

Στο κεφάλαιο αυτό περιγράφουμε τις κλάσεις που απαρτίζουν το σύστημα.

5.2.1 public class Peer

Η κλάση αυτή υλοποιεί έναν ομότιμο κόμβο του δικτύου και περιέχει τις μεταβλητές:

- *String peerName*: όνομα κόμβου
- *String schemaName*: όνομα σχήματος ΒΔ που βρίσκεται στον κόμβο
- *boolean isConnected*: δείχνει αν ο κόμβος συνδέεται με άλλους
- *ArrayList <Peer> acquaintances*: λίστα με γειτονικούς (συνδεδεμένους) κόμβους

Και τις μεθόδους:

- *public void setConnected(Peer peer)*: ενώνει δύο κόμβους (τους κάνει γειτονικούς)
- *public void printConnectedPeers()*: εκτυπώνει τους γειτονικούς κόμβους ενός κόμβου

5.2.2 *public class Schema*

Υλοποιεί ένα σχήμα ΒΔ και περιέχει τις μεταβλητές:

- *String schemaName*: όνομα σχήματος
- *ArrayList <Relation> relations*: λίστα από σχέσεις

Και τις μεθόδους:

- *public void setRelation(Relation relation_)*: προσθέτει στη λίστα relations τη σχέση relation_
- *public void printRelations()*: εκτυπώνει όλες τις σχέσεις του σχήματος
- *public void printSchema()*: εκτυπώνει όλες τις σχέσεις του σχήματος με τα χαρακτηριστικά τους
- *public Relation getRelation(String relationName_)*: δοθέντος του ονόματος μιας σχέσης επιστρέφει τη σχέση με το συγκεκριμένο όνομα στο σχήμα
- *public void createHasCorrespondence(ArrayList <Correspondence> list)*: δημιουργεί τις HASA συσχετίσεις του σχήματος, μεταξύ της κάθε σχέσης με κάθε ένα από τα χαρακτηριστικά της και τις προσθέτει στη λίστα list
- *public void createNodes(Graph g)*: κατασκευάζει ένα κόμβο στο γράφο g για κάθε σχέση και κάθε χαρακτηριστικό, χρησιμοποιώντας την μέθοδο addNodeToGraph(String nodeName_, Node node_) της κλάσης Graph, που θα εξετάσουμε στη συνέχεια.

5.2.3 *public class Relation*

Υλοποιεί μία σχέση (relation) και περιέχει τις μεταβλητές:

- *String relationName*: όνομα σχέσης

- *ArrayList <Attribute> attributes*: λίστα από χαρακτηριστικά (attributes) της σχέσης

Και τις μεθόδους:

- *public void setAttribute(Attribute attribute_)*: προσθέτει το χαρακτηριστικό attribute_ στη λίστα attributes
- *public void deleteAttribute(Attribute attribute_)*: διαγράφει το χαρακτηριστικό attribute_ από τη λίστα attributes
- *public Attribute getAttribute(String attributeName_)*: δοθέντος του ονόματος ενός χαρακτηριστικού επιστρέφει το χαρακτηριστικό με το συγκεκριμένο όνομα στη σχέση
- *public void printAttributes()*: εκτυπώνει όλα τα χαρακτηριστικά της σχέσης

5.2.4 *public class Attribute*

Υλοποιεί ένα χαρακτηριστικό (attribute) και περιέχει τις μεταβλητές:

- *String attributeName*: όνομα χαρακτηριστικού
- *String value*: τιμή του χαρακτηριστικού
- *boolean key*: ορίζει αν το χαρακτηριστικό είναι κλειδί

5.2.5 *public class Correspondence*

Υλοποιεί μία συσχέτιση μεταξύ στοιχείων (σχέσεων και χαρακτηριστικών) των σχημάτων και περιέχει τις μεταβλητές:

- *String correspondenceName*: όνομα συσχέτισης
- *Relation relation1*: σχέση αριστερού μέλους συσχέτισης
- *Relation relation2*: σχέση δεξιού μέλους συσχέτισης
- *Attribute attribute1*: χαρακτηριστικό αριστερού μέλους συσχέτισης
- *Attribute attribute2*: χαρακτηριστικό δεξιού μέλους συσχέτισης
- *String relationship*: είδος συσχέτισης (HASA, ID, ISA ή Related)

Κάθε μέλος είναι είτε σχέση είτε χαρακτηριστικό. Έτσι, η μία από τις μεταβλητές relation1 και attribute1 έχει τιμή και η άλλη είναι null. Το ίδιο ισχύει και για τις relation2 και attribute2.

5.2.6 *public class Condition*

Υλοποιεί μία συνθήκη ενός ερωτήματος. Ουσιαστικά μία συνθήκη δείχνει τη σχέση μεταξύ δύο χαρακτηριστικών ή ενός χαρακτηριστικού και μιας συμβολοσειράς (string). Γι' αυτό το λόγο, το δεξί μέλος της συνθήκης μπορεί να είναι είτε χαρακτηριστικό είτε συμβολοσειρά.

Η κλάση αυτή περιέχει τις μεταβλητές:

- *Attribute attPrefix*: χαρακτηριστικό ως πρόθεμα (αριστερό μέρος) της συνθήκης
- *Attribute attSuffix*: χαρακτηριστικό ως επίθεμα (δεξί μέρος) της συνθήκης
- *String sSuffix*: συμβολοσειρά ως επίθεμα (δεξί μέρος) της συνθήκης
- *String symbol*: σύμβολο συνθήκης

Και τη μέθοδο:

- *public void printCondition()*: εκτύπωση συνθήκης

5.2.7 *public class Query*

Υλοποιεί ένα ερώτημα, το οποίο θα χρησιμοποιηθεί για την έκφραση των αντιστοιχίσεων (mappings), που θα περιγραφούν παρακάτω.

Περιέχει τις μεταβλητές:

- *String queryName*: όνομα ερωτήματος
- *ArrayList <Relation> relations*: λίστα από σχέσεις που συνδέονται με join
- *ArrayList <Attribute> attributes*: λίστα από χαρακτηριστικά των σχέσεων της λίστας relations που χρησιμοποιούνται από το ερώτημα
- *ArrayList <Condition> conditions*: λίστα από συνθήκες

Ουσιαστικά η κλάση αυτή υλοποιεί ένα ερώτημα της μορφής:

```
SELECT {attribute1,...,attributeN}  
FROM {relation1 ▷<...▷<relationM}  
WHERE {condition1,..., conditionK}
```

Οι μέθοδοι που περιέχονται είναι:

- *public void setRelation(Relation relation_)*: προσθέτει την σχέση relation_ στην λίστα relations
- *public Relation getRelation(String relationName_)*: δοθέντος του ονόματος μιας σχέσης από τη λίστα relations, επιστρέφει την σχέση αυτή

- *public void setAttribute(Attribute attribute_)*: προσθέτει την σχέση *attribute_* στην λίστα *attributes*
- *public Attribute getAttribute(String attributeName_)*: δοθέντος του ονόματος ενός χαρακτηριστικού από τη λίστα *attributes*, επιστρέφει το χαρακτηριστικό αυτό
- *public void setCondition(Condition condition_)*: προσθέτει μία συνθήκη στην λίστα *conditions*
- *public void printRelations()*: εκτυπώνει τις σχέσεις με τα χαρακτηριστικά που έχουν στο συγκεκριμένο ερώτημα
- *public void printAttributes()*: εκτυπώνει τα χαρακτηριστικά που υπάρχουν στη λίστα *attributes*
- *public void printConditions()*: εκτυπώνει τις συνθήκες της λίστας *conditions*
- *public void printQuery()*: εκτυπώνει το ερώτημα στη μορφή: *queryName :- relations, conditions*
- *public void printContentsOfQuery()*: εκτυπώνει το ερώτημα χωρίς το όνομα του, δηλαδή στη μορφή: *relations, conditions*

5.2.8 *public class Mapping*

Υλοποιεί μία αντιστοίχιση μεταξύ στοιχείων (σχέσεων και χαρακτηριστικών) των σχημάτων και περιέχει τις μεταβλητές:

- *Query query1*: το ερώτημα του αριστερού μέλους της αντιστοίχισης
- *Query query2*: το ερώτημα του δεξιού μέλους της αντιστοίχισης

Και της μέθοδο:

- *public void printMapping()*: εκτυπώνει την αντιστοίχιση στη μορφή *query1 :- query2*

5.2.9 *public class Node*

Υλοποιεί ένα κόμβο ενός γράφου, με τη βοήθεια του οποίου αναπαριστούμε ένα σχήμα. Περιέχει τις μεταβλητές:

- *String nodeName*: όνομα κόμβου (συμπίπτει συνήθως με το όνομα της σχέσης ή του χαρακτηριστικού που αντιπροσωπεύει)
- *int relationCounter*: μετρητής του αριθμού των σχέσεων που αντιπροσωπεύουν τον κόμβο

- *int attributeCounter*: μετρητής του αριθμού των χαρακτηριστικών που αντιπροσωπεύουν τον κόμβο
- *ArrayList <Node> to*: λίστα από κόμβους προς τους οποίους ο συγκεκριμένος κόμβος έχει ακμές (λίστα από κόμβους προορισμούς)
- *ArrayList <Node> from*: λίστα από κόμβους από τους οποίους ο συγκεκριμένος κόμβος έχει ακμές (λίστα από κόμβους πηγές)
- *ArrayList <String> otherNames*: λίστα από επιπρόσθετα ονόματα, δηλαδή ονόματα των κόμβων που συγχωνεύθηκαν με τον κόμβο αυτό

Οι μέθοδοι της κλάσης είναι:

- *public void printOtherNames()*: εκτύπωση των επιπρόσθετων ονομάτων ενός κόμβου

5.2.10 *public class Edge*

Υλοποιεί μία ακμή ενός γράφου αναπαράστασης ενός σχήματος και περιέχει τις μεταβλητές:

- *Node source*: κόμβος που αποτελεί πηγή για την ακμή (από αυτόν ξεκινάει η ακμή)
- *Node target*: κόμβος που αποτελεί προορισμό για την ακμή (σ' αυτόν καταλήγει η ακμή)
- *String edgeCorrespondence*: είδος συσχέτισης (ISA, ID, HASA, Related)

5.2.11 *public class Graph*

Η κλάση αυτή υλοποιεί ένα γράφο που χρησιμοποιείται για την αναπαράσταση σχήματος. Περιέχει τις μεταβλητές:

- *String graphName*: όνομα γράφου
- *ArrayList <Node> nodes*: λίστα από κόμβους
- *ArrayList <Edge> edges*: λίστα από ακμές

Και τις μεθόδους:

- *public Edge getEdge(Node source_, Node target_, String edgeCor)*: επιστρέφει την ακμή που έχει ως πηγή τον κόμβο *source_*, ως προορισμό τον κόμβο *target_* και είδος συσχέτισης *edgeCor*
- *public Node getNode(String nodeName_)*: δοθείσας μιας συμβολοσειράς επιστρέφει τον κόμβο του γράφου που έχει ως όνομα τη συμβολοσειρά αυτή
- *public void addNodeToGraph(String nodeName_, Node node_)*: προσθέτει στον γράφο ένα κόμβο με όνομα *nodeName*

- *public void addNodeToGraph(Relation relation_, Node node_)*: προσθέτει στον γράφο ένα κόμβο που αντιπροσωπεύει τη σχέση *relation_*
- *public void addNodeToGraph(Attribute attribute_, Node node_)*: προσθέτει στον γράφο ένα κόμβο που αντιπροσωπεύει το χαρακτηριστικό *attribute_*
- *public void addEdgeToGraph(Edge edge_, Node source_, Node target_, String edgeCorrespondence_)*:
- *public void deleteEdgeFromGraph(Edge edge_)*: διαγράφει την ακμή *edge_* από το γράφο
- *public void printTargetNodes(Node source_)*: εκτυπώνει τους κόμβους που είναι προορισμοί σε ακμές που έχουν ως πηγή τον κόμβο *source_*
- *public void printTargetNodesWithCor(Node source_)*: εκτυπώνει τους κόμβους που είναι προορισμοί σε ακμές που έχουν ως πηγή τον κόμβο *source_*, καθώς και το είδος συσχέτισης της ακμής
- *public void printTargetNodes(String sourceName_)*: εκτυπώνει τους κόμβους που είναι προορισμοί σε ακμές που έχουν ως πηγή τον κόμβο με όνομα *sourceName_*
- *public void printNodes()*: εκτυπώνει όλους τους κόμβους του γράφου
- *public void printEdges()*: εκτυπώνει όλες τις ακμές του γράφου
- *public void printGraph()*: εκτυπώνει όλους τους κόμβους του γράφου και τους κόμβους με τους οποίους συνδέονται
- *public void printGraphWithCor()*: εκτυπώνει όλους τους κόμβους του γράφου και τους κόμβους με τους οποίους συνδέονται, καθώς και το είδος της μεταξύ τους συσχέτισης
- *public void mergeNodes(Node node1, Node node2)*: συγχωνεύει τους κόμβους *node1* και *node2*. Κατά τη συνένωση, ο κόμβος *node2* συνενώνεται με τον *node1* και διαγράφεται. Έτσι, ο τελικός γράφος που προκύπτει από τη συνένωση είναι ο *node1*. Ο αλγόριθμος λειτουργεί ως εξής: διατρέχει όλες τις ακμές του γράφου, βρίσκει τις ακμές που ξεκινούν από τον κόμβο *node2*, τις διαγράφει και προσθέτει νέες, αν δεν υπάρχουν, που ξεκινούν από τον κόμβο *node1* και έχουν τον ίδιο προορισμό και το ίδιο είδος συσχέτισης με τις αντίστοιχες που σβήστηκαν. Το ίδιο συμβαίνει για τις ακμές που καταλήγουν στον κόμβο *node2*. Οι ακμές που θα προστεθούν αποθηκεύονται προσωρινά στη λίστα *newEdges*, ενώ αυτές που θα αφαιρεθούν στη λίστα *oldEdges*.
- *public void copy(Graph g2)*: αντιγράφει τον γράφο, δημιουργώντας ένα γράφο (*g2*) όμοιο με τον αρχικό

5.2.12 *public class Main*

Στην κλάση αυτή περιέχεται η μέθοδος `main` που υλοποιεί το ζητούμενο σύστημα, καθώς και άλλες βοηθητικές μέθοδοι.

5.2.12.1 *public static void main(String[] args)*

Η μέθοδος αυτή υλοποιεί το σύστημα διαχείρισης ολοκλήρωσης σχημάτων ΒΔ με χρήση οντολογιών. Στο σημείο αυτό, θα περιγράψουμε τη διαδικασία που ακολουθείται.

1. Κατασκευή δικτύου P2P, σχημάτων των βάσεων δεδομένων, συσχετίσεων και αντιστοιχίσεων

Τα στοιχεία των σχημάτων, δηλαδή οι σχέσεις με τα χαρακτηριστικά και τα κλειδιά τους, καθώς και οι συσχετίσεις (correspondences) και οι αντιστοιχίσεις (mappings) μεταξύ τους, εισάγονται μέσω ενός αρχείου εισόδου. Αρχικά, δημιουργούνται οι κόμβοι του δικτύου P2P, τα σχήματα, οι συσχετίσεις (ISA και ID) και οι αντιστοιχίσεις (ISA, ID και Related) σύμφωνα με το αρχείο εισόδου. Οι συσχετίσεις αποθηκεύονται στη λίστα `corList` και οι αντιστοιχίσεις στη λίστα `mapList`. Ακολούθως, δημιουργούνται οι συσχετίσεις τύπου HASA, μεταξύ όλων των σχέσεων και των χαρακτηριστικών τους, με κλήση της μεθόδου `createHasCorrespondence(corList)` της κλάσης `Schema`. Επίσης, δημιουργούνται συσχετίσεις από τις αντιστοιχίσεις με χρήση των μεθόδων `createIDCor`, `createISACor` και `createRelatedCor` που περιγράφονται παρακάτω, και σύμφωνα με τους κανόνες που αναφέρονται στην παράγραφο 4.2.3.

2. Κατασκευή γράφων αναπαράστασης

Στη συνέχεια, κατασκευάζεται για κάθε σχήμα ένας γράφος αναπαράστασης. Συγκεκριμένα, κατασκευάζονται οι κόμβοι, με χρήση της μεθόδου `createGraph` της κλάσης `Schema`, και οι ακμές τύπου HASA, μεταξύ των κόμβων που αναπαριστούν σχέσεις και αυτών που αναπαριστούν τα αντίστοιχα χαρακτηριστικά.

3. Κατασκευή γράφου συγχώνευσης και εμπλουτισμός του με σημασιολογικές σχέσεις

Ακολούθως, γίνεται η συγχώνευση των γράφων, με χρήση της μεθόδου `merge(g1,g2)` η οποία συγχωνεύει δύο γράφους. Κάθε γράφος συγχωνεύεται με το αποτέλεσμα της συγχώνευσης των προηγούμενων, που είναι ο γράφος `g1`. Έτσι, ο γράφος `g1` είναι ο τελικός

γράφος που προκύπτει από τη συγχώνευση των γράφων όλων των σχημάτων. Ο γράφος αυτός περιέχει μόνο ακμές τύπου HASA, όπως και οι γράφοι αναπαράστασης των σχημάτων. Η συγχώνευση συνεχίζεται και ο γράφος εμπλουτίζεται και με τις υπόλοιπες σημασιολογικές σχέσεις (ID, ISA, Related) με χρήση των συσχετίσεων.

Συγκεκριμένα, προστίθενται ακμές τύπου Related μεταξύ των κόμβων που αναπαριστούν σχέσεις ή χαρακτηριστικά τα οποία συνδέονται με συσχετίσεις τύπου Related. Στη συνέχεια, συγχωνεύονται οι κόμβοι που αντιπροσωπεύουν σχέσεις ή χαρακτηριστικά που συσχετίζονται με συσχέτιση ID, δηλαδή ταυτίζονται σημασιολογικά. Αυτό γίνεται με χρήση της μεθόδου `mergeID(g1,corList)`, που περιγράφεται παρακάτω. Ακολουθεί η προσθήκη ακμών τύπου ISA και προσαρμογή των γειτονικών ακμών, όπως περιγράφεται στην παράγραφο 4.2.4, με χρήση της μεθόδου `mergeISA(g1,corList)`. Έτσι, έχει κατασκευαστεί ο τελικός γράφος συγχώνευσης, ο οποίος είναι εμπλουτισμένος με όλες τις σημασιολογικές σχέσεις μεταξύ των στοιχείων των σχημάτων των βάσεων δεδομένων των κόμβων του P2P δικτύου.

4. Κατασκευή καθολικού σχήματος

Στην επόμενη φάση, κατασκευάζεται το κεντρικό/καθολικό σχήμα με κλήση της μεθόδου `createGlobalSchema(g1,globalSchema,schemaList)`, η λειτουργία της οποίας θα περιγραφεί στη συνέχεια.

5. Συμπίεση καθολικού σχήματος

Τέλος, γίνεται η συμπίεση του καθολικού σχήματος. Αρχικά, ορίζεται ένα κατώφλι για το επιθυμητό πλήθος των σχέσεων και κατασκευάζεται ένα αντίγραφο `g2` του γράφου `g1`. Στη συνέχεια, ελέγχεται αν το πλήθος των σχέσεων του γράφου υπερβαίνει το κατώφλι και αν αυτό ισχύει, αρχίζει η διαδικασία της συμπίεσης. Ο έλεγχος και η συμπίεση γίνονται επαναληπτικά σε κάθε συμπιεσμένο σχήμα που προκύπτει.

Η διαδικασία της συμπίεσης γίνεται ως εξής: Αρχικά, καλείται η μέθοδος `toDeleteForCompression(g2)`, η οποία επιστρέφει το κόμβο που πρέπει να αφαιρεθεί στη συγκεκριμένη φάση της συμπίεσης, σύμφωνα με τους κανόνες που περιγράφηκαν στην παράγραφο 4.3. Αν ο κόμβος που επέστρεψε η προαναφερθείσα μέθοδος δεν είναι null, καλείται η μέθοδος `compressionDelete(nodeToDelete,g2)`, που διαγράφει στα πλαίσια της συμπίεσης τον κόμβο `nodeToDelete` από το γράφο `g2`. Από τον συμπιεσμένο γράφο που προκύπτει κατασκευάζουμε το κεντρικό σχήμα με χρήση της μεθόδου `createGlobalSchema(g2,compressedGlobalSchema,schemaList)`. Η διαδικασία της συμπίεσης επαναλαμβάνεται, ελέγχοντας κάθε φορά το νέο συμπιεσμένο γράφο που προκύπτει.

Μέρος του κώδικα, που υλοποιεί τον επαναληπτικό έλεγχο και συμπίεση είναι το ακόλουθο:

```
int relNumber = globalSchema.relations.size();
while ((relNumber>threshold) && !finish)
{
    String nodeToDelete = toDeleteForCompression(g2);
    if (!nodeToDelete.equalsIgnoreCase(""))
    {
        compressionDelete(nodeToDelete,g2);
        Schema compressedGlobalSchema = new Schema("Global");
        createGlobalSchema(g2,compressedGlobalSchema,
        schemaList);
        relNumber=compressedGlobalSchema.relations.size();
    }
    else
        finish = true;
}
```

Ακολούθως, θα περιγράψουμε τις προαναφερθείσες μεθόδους που χρησιμοποιεί η μέθοδος main.

5.2.12.2 *public static int checkMapCorrs(Correspondence cor, ArrayList corList)*

Η μέθοδος αυτή καλείται από τις μεθόδους createIDCor, createISACor, createRelatedCor και ελέγχει αν μία συσχέτιση (correspondence), που έχει μόλις παραχθεί από μία αντιστοίχιση (mapping), υπάρχει ήδη ή έρχεται σε αντίθεση με τις ήδη υπάρχουσες. Αν συμβαίνει κάτι τέτοιο επιστρέφει 0, διαφορετικά επιστρέφει 1. Μία συσχέτιση έρχεται σε αντίθεση με μία άλλη, αν σχετίζουν τις ίδιες σχέσεις ή χαρακτηριστικά, αλλά το είδος της συσχέτισης (ID, ISA, HASA, Related) είναι διαφορετικό. Σε περίπτωση που δύο συσχετίσεις έρχονται σε αντίθεση, θέλουμε να υπερισχύει αυτή που υπήρχε ήδη, οπότε η νέα συσχέτιση να μην προστίθεται στη λίστα των συσχετίσεων.

5.2.12.3 *public static void createIDCor(ArrayList mapList,ArrayList corList)*

Η μέθοδος αυτή ελέγχει τις αντιστοιχίσεις (mappings) που υπάρχουν στη λίστα mapList και αν ισχύουν οι κανόνες 1 και 3, που αναφέρθηκαν στην παράγραφο 4.2.3, δημιουργεί

συσχετίσεις τύπου ID. Στη συνέχεια, ελέγχει αν οι νέες συσχετίσεις έρχονται σε αντίθεση με τις ήδη υπάρχουσες, καλώντας τη μέθοδο `checkMapCorrs` για κάθε νέα συσχέτιση. Αν η τελευταία επιστρέψει 0, η συγκεκριμένη συσχέτιση δεν προστίθεται στη λίστα των συσχετίσεων.

5.2.12.4 *public static void createISACor(ArrayList mapList, ArrayList corList)*

Ομοίως με την προηγούμενη, η μέθοδος αυτή ελέγχει τις αντιστοιχίσεις (mappings) που υπάρχουν στη λίστα `mapList` και αν ισχύει ο κανόνας 2 της παραγράφου 4.2.3, δημιουργεί συσχετίσεις τύπου ISA. Στη συνέχεια, ελέγχει αν οι νέες συσχετίσεις έρχονται σε αντίθεση με τις ήδη υπάρχουσες, καλώντας τη μέθοδο `checkMapCorrs` για κάθε νέα συσχέτιση. Αν η τελευταία επιστρέψει 0, η συγκεκριμένη συσχέτιση δεν προστίθεται στη λίστα των συσχετίσεων.

5.2.12.5 *public static void createRelatedCor(ArrayList mapList, ArrayList corList)*

Ομοίως με τις προηγούμενες, η μέθοδος αυτή ελέγχει τις αντιστοιχίσεις (mappings) που υπάρχουν στη λίστα `mapList` και αν ισχύει ο κανόνας 4 της παραγράφου 4.2.3, δημιουργεί συσχετίσεις τύπου Related. Στη συνέχεια, ελέγχει αν οι νέες συσχετίσεις έρχονται σε αντίθεση με τις ήδη υπάρχουσες, καλώντας τη μέθοδο `checkMapCorrs` για κάθε νέα συσχέτιση. Αν η τελευταία επιστρέψει 0, η συγκεκριμένη συσχέτιση δεν προστίθεται στη λίστα των συσχετίσεων.

5.2.12.6 *public static void merge(Graph g1, Graph g2)*

Η μέθοδος αυτή συγχωνεύει δύο γράφους `g1` και `g2`. Ουσιαστικά προσθέτει τους κόμβους και τις αντίστοιχες ακμές του γράφου `g2` στον `g1`. Οπότε ο γράφος συγχώνευσης που προκύπτει είναι ο `g1`. Η διαδικασία της συγχώνευσης είναι η εξής:

Αρχικά, διατρέχονται όλοι οι κόμβοι του `g2` και γίνεται έλεγχος για τον καθένα, για να βρεθεί αν υπάρχει στον `g1` κόμβος με το ίδιο όνομα. Έτσι, προστίθενται μόνο όσοι κόμβοι δεν υπάρχουν ήδη στον `g1`. Η προσθήκη των κόμβων γίνεται με χρήση της μεθόδου `g1.addNodeToGraph(name,newNode)`, που περιγράψαμε παραπάνω.

Στη συνέχεια, διατρέχονται οι ακμές του `g2` και γίνεται έλεγχος για καθεμία αν υπάρχει στον `g1` ίδια ακμή, δηλαδή ακμή με την ίδια πηγή, τον ίδιο προορισμό και το ίδιο είδος

συσχέτισης. Προστίθενται μόνο οι ακμές που δεν υπάρχουν ήδη στον `g1` με χρήση της μεθόδου `g1.addEdgeToGraph(newEdge,source,target,correspondence)`.

5.2.12.7 *public static void mergeID(Graph g1, ArrayList corList)*

Η μέθοδος αυτή διατρέχει τη λίστα των συσχετίσεων `corList`. Για κάθε συσχέτιση τύπου ID που βρίσκει, συγχωνεύει τους αντίστοιχους κόμβους στο γράφο, με κλήση της μεθόδου `g1.mergeNodes(node1,node2)`.

5.2.12.8 *public static void mergeISA(Graph g1,ArrayList corList)*

Η μέθοδος αυτή διατρέχει τη λίστα των συσχετίσεων `corList`. Για κάθε συσχέτιση τύπου ISA που βρίσκει, δημιουργεί μία νέα ακμή τύπου ISA ανάμεσα στους αντίστοιχους κόμβους, με κλήση της μεθόδου `g1.addEdgeToGraph(newEdge,node1,node2,"ISA")`. Επιπλέον, αφαιρεί τις ακμές που συνδέουν την πηγή μιας ISA ακμής με κόμβους οι οποίοι συνδέονται με τον προορισμό αυτής της ISA ακμής.

5.2.12.9 *public static void createGlobalSchema(Graph g1, Schema globalSchema, ArrayList schemaList)*

Η μέθοδος αυτή κατασκευάζει το καθολικό σχήμα από τον καθολικό γράφο, που προέκυψε από την συγχώνευση των τοπικών γράφων σε ένα καθολικό γράφο και τον σημασιολογικό εμπλουτισμό του. Αρχικά, ελέγχει τη μεταβλητή `relationCounter` κάθε κόμβου του καθολικού γράφου και αν αυτή είναι διάφορη του μηδενός, δηλαδή αν ο συγκεκριμένος κόμβος έχει προκύψει από μία ή περισσότερες σχέσεις (`relations`), δημιουργεί μία συνώνυμη με τον κόμβο σχέση του καθολικού σχήματος. Στη συνέχεια δημιουργούνται τα χαρακτηριστικά και τα κλειδιά κάθε σχέσης, με κλήση της μεθόδου `createGlobalAttributesWithHASA(rel,g1,no,schemaList)`. Επίσης, με κλήση της μεθόδου `createGlobalKeysForISA(g1,globalSchema)`, προστίθεται το κλειδί των σχέσεων των οποίων οι κόμβοι είναι προορισμοί ακμών τύπου ISA, στις σχέσεις των οποίων οι κόμβοι είναι πηγές των αντίστοιχων ακμών. Τέλος, αν υπάρχει κάποια σχέση με χαρακτηριστικό το οποίο έχει ίδιο όνομα με μία άλλη σχέση, το χαρακτηριστικό αυτό της πρώτης σχέσης αντικαθίσταται με το κλειδί της δεύτερης. Αυτό γίνεται με χρήση της μεθόδου `relAtt(g1,globalSchema)`.

5.2.12.10 *public static void createGlobalAttributesWithHASA(Relation rel,
Graph g1, Node no, ArrayList schemaList)*

Η μέθοδος αυτή κατασκευάζει τα καθολικά χαρακτηριστικά μιας σχέσης *rel* του καθολικού σχήματος, η οποία αντιπροσωπεύεται από τον κόμβο *no* στον καθολικό γράφο *g1*. Η δημιουργία των χαρακτηριστικών γίνεται με τη βοήθεια των ακμών τύπου *HASA* του καθολικού γράφου. Συγκεκριμένα, για κάθε *HASA* ακμή που έχει πηγή κόμβο που αντιπροσωπεύει μία σχέση του καθολικού σχήματος, δημιουργείται χαρακτηριστικό με το όνομα του κόμβου προορισμού της ακμής αυτής.

Για κάθε χαρακτηριστικό που δημιουργείται, ελέγχεται αν αποτελεί κλειδί σε συνώνυμη σχέση κάποιου από τα αρχικά σχήματα, και αν αυτό ισχύει, το χαρακτηριστικό αυτό γίνεται κλειδί και της συγκεκριμένης σχέσης του καθολικού σχήματος.

Αν το χαρακτηριστικό δεν βρεθεί ως κλειδί σε συνώνυμη σχέση, ελέγχεται αν ο κόμβος του χαρακτηριστικού αυτού έχει τη μεταβλητή *otherNames* διάφορη του *null*, δηλαδή έχει συγχωνευτεί με άλλα χαρακτηριστικά. Αν αυτό ισχύει, εξετάζει αν κάποιο από τα χαρακτηριστικά αυτά αποτελεί κλειδί σε συνώνυμη σχέση κάποιου τοπικού σχήματος και ανάλογα θέτει ή όχι το χαρακτηριστικό ως κλειδί της συγκεκριμένης σχέσης του καθολικού σχήματος. Ο έλεγχος αν το χαρακτηριστικό με όνομα *name* στη σχέση *rel* κάποιου τοπικού σχήματος υπάρχει ως κλειδί, γίνεται με τη βοήθεια της μεθόδου *checkGlobalKey(schemaList,rel,name)*.

Αν δεν βρεθεί ως κλειδί ούτε από τα συνώνυμα αυτά χαρακτηριστικά, ελέγχεται αν υπάρχει ως κλειδί σε κάποια σχέση της λίστας *otherNames*, δηλαδή σε κάποια σχέση των τοπικών σχημάτων που συγχωνεύτηκε με τη συγκεκριμένη σχέση.

5.2.12.11 *public static int checkGlobalKey(ArrayList schemaList, Relation rel,
String attName)*

Η μέθοδος αυτή ελέγχει αν το χαρακτηριστικό με όνομα *name* στη σχέση *rel* υπάρχει ως κλειδί σε κάποιο τοπικό σχήμα. Αν ισχύει αυτό επιστρέφει 1, διαφορετικά 0.

5.2.12.12 *public static void createGlobalKeysForISA(Graph g1, Schema
globalSchema)*

Η μέθοδος αυτή ελέγχει κάθε σχέση του καθολικού γράφου *g1* και αν ο κόμβος της αποτελεί πηγή ακμής ISA, αποκτάει ως κλειδί, το κλειδί της σχέσης της οποίας ο κόμβος είναι προορισμός της ακμής αυτής. Αν το χαρακτηριστικό που είναι κλειδί του προορισμού υπάρχει στην πηγή, τίθεται απλά ως κλειδί στην πηγή. Διαφορετικά, το χαρακτηριστικό δημιουργείται εξ αρχής και τίθεται ως κλειδί.

5.2.12.13 *public static void relAtt(Graph g1, Schema globalSchema)*

Η μέθοδος αυτή ελέγχει αν υπάρχει κάποια σχέση (A) με χαρακτηριστικό (B) το οποίο έχει ίδιο όνομα με μία άλλη σχέση (Γ). Αν ισχύει αυτό, ελέγχεται αν υπάρχει ακμή τύπου Related ανάμεσα στους κόμβους των A και Γ αντίστοιχα. Αν υπάρχει τέτοια ακμή, το χαρακτηριστικό B διαγράφεται από χαρακτηριστικό της σχέσης A, ενώ προστίθεται ως χαρακτηριστικό της το Γ. Στην περίπτωση που δεν υπάρχει ακμή τύπου Related, το χαρακτηριστικό B της πρώτης σχέσης A αντικαθίσταται με το κλειδί της δεύτερης σχέσης Γ.

5.2.12.14 *public static String toDeleteForCompression (Graph g1)*

Η μέθοδος αυτή επιστρέφει το όνομα (συμβολοσειρά) της σχέσης που θα διαγραφεί στη συγκεκριμένη φάση (επανάληψη) της διαδικασίας της συμπίεσης. Η απόφαση για το ποια σχέση θα διαγραφεί λαμβάνεται σύμφωνα με τους κανόνες της παραγράφου 4.3.

Αρχικά, ελέγχεται αν υπάρχουν σχέσεις, των οποίων οι κόμβοι έχουν εισερχόμενες ακμές ISA (κανόνας 1). Όλες αυτές οι σχέσεις αποθηκεύονται στη λίστα *toDeleteList1*.

Στη συνέχεια, ελέγχονται όλες οι σχέσεις της λίστας *toDeleteList1* και όσων απ' αυτές οι κόμβοι δεν έχουν εισερχόμενες ακμές HASA (κανόνας 2), αποθηκεύονται στη λίστα *toDeleteList2*.

Το επόμενο βήμα είναι να εφαρμοστεί ο κανόνας 3. Για το σκοπό αυτό καλείται η μέθοδος *rule3(toDeleteList2,g1)*, η οποία επιστρέφει το όνομα (συμβολοσειρά) της σχέσης που θα διαγραφεί στη φάση αυτή.

Σε περίπτωση που ο κανόνας 2 δεν ισχύει, δηλαδή δεν υπάρχουν κόμβοι με εισερχόμενες ακμές ISA και χωρίς εισερχόμενες ακμές HASA, εφαρμόζεται ο κανόνας 3 για τις σχέσεις με κόμβους με εισερχόμενες ακμές ISA και εισερχόμενες ακμές HASA, δηλαδή καλείται η παραπάνω μέθοδος ως εξής: *rule3(toDeleteList1,g1)*.

Σε περίπτωση που δεν ισχύει ο κανόνας 1, δηλαδή δεν υπάρχουν κόμβοι με εισερχόμενες ακμές ISA, επιστρέφεται null και η διαδικασία της συμπίεσης σταματάει.

5.2.12.15 *public static String rule3(ArrayList toDeleteList, Graph g1)*

Η μέθοδος αυτή υλοποιεί τον κανόνα 3 της διαδικασίας της συμπίεσης. Ελέγχει όλους τους κόμβους με ονόματα από τη λίστα toDeleteList και επιστρέφει αυτόν που έχει το μικρότερο αριθμό εισερχομένων HASA ακμών.

5.2.12.16 *public static void compressionDelete(String nodeToDelete, Graph g2)*

Η μέθοδος αυτή διαγράφει στα πλαίσια της συμπίεσης τον κόμβο nodeToDelete από το γράφο g2. Επίσης, ενώνει τους κόμβους που έχουν εξερχόμενες ακμές προς τον nodeToDelete, με τους κόμβους που έχουν εισερχόμενες από τον nodeToDelete ακμές, αν φυσικά δεν υπάρχουν ήδη ίδιες ακμές. Αυτές οι εισερχόμενες και εξερχόμενες ακμές του nodeToDelete διαγράφονται. Οι νέες ακμές που δημιουργούνται είναι του ίδιου τύπου με αυτές που είχαν πηγή τον nodeToDelete και προορισμό τους κόμβους-προορισμούς.

6

Αξιολόγηση - Πειράματα

Στο κεφάλαιο αυτό θα παρουσιάσουμε πειράματα αξιολόγησης των σχημάτων που προκύπτουν από την τεχνική ολοκλήρωσης που προτείναμε, σε σχέση με τα σχήματα εισόδου.

6.1 Παράμετροι αξιολόγησης

Ως παραμέτρους αξιολόγησης της μεθόδου μας θα χρησιμοποιήσουμε την ομοιότητα μεταξύ των αρχικών και του τελικού σχήματος. Το ποσοστό ομοιότητάς τους αποτελεί ένδειξη του πόσο όμοια δομημένες είναι οι πληροφορίες που φυλάσσονται σ' αυτά, και επομένως του πόσο όμοια είναι τα αποτελέσματα αποτίμησης ενός ερωτήματος στα αρχικά και το τελικό σχήμα.

Ως παραμέτρους για τον υπολογισμό της ομοιότητας μεταξύ σχημάτων θα χρησιμοποιήσουμε τις σχέσεις, τις ιδιότητες και τα κλειδιά των σχημάτων, και συγκεκριμένα το πλήθος των στοιχείων αυτών που είναι διαφορετικά μεταξύ τους. Η ομοιότητα μεταξύ σχημάτων θα υπολογίζεται κάθε φορά ως προς μία από τις παραμέτρους ομοιότητας.

Έτσι, θα παρουσιάσουμε πειράματα που εξετάζουν την ομοιότητα των αρχικών τοπικών σχημάτων με το καθολικό σχήμα, καθώς και με το συμπιεσμένο καθολικό σχήμα, που παράγονται με τη διαδικασία ολοκλήρωσης που περιγράφηκε στα προηγούμενα κεφάλαια.

6.2 Οργάνωση πειραμάτων

Εκτελέσαμε πειράματα τριών διαφορετικών κατηγοριών:

1. Στην πρώτη, εξετάζεται η **μεταβολή της ομοιότητας του καθολικού σχήματος με τα αρχικά**, καθώς μεταβάλλεται η ομοιότητα μεταξύ των αρχικών σχημάτων. Η ομοιότητα ανάμεσα στα σχήματα υπολογίζεται ως προς κάθε μία από τις τρεις παραμέτρους ομοιότητας (σχέσεις, ιδιότητες ή κλειδιά).
2. Στη δεύτερη, αντί την ομοιότητα του καθολικού σχήματος με τα αρχικά, μελετάμε την **ομοιότητα του συμπιεσμένου καθολικού σχήματος με τα αρχικά**, με τον ίδιο τρόπο όπως στην πρώτη κατηγορία πειραμάτων. Η ομοιότητα υπολογίζεται, και πάλι, ως προς κάθε μία από τις τρεις παραμέτρους.
3. Τέλος, στην τρίτη κατηγορία πειραμάτων, εξετάζουμε το ρόλο που παίζει ο **βαθμός συμπίεσης** του τελικού σχήματος. Συγκεκριμένα, στο πρώτο μέρος των πειραμάτων συγκρίνουμε την ομοιότητα των αρχικών σχημάτων με το τελικό (καθολικό ή συμπιεσμένο) για διάφορους βαθμούς συμπίεσης. Στο δεύτερο μέρος, συγκρίνουμε την ομοιότητα των συμπιεσμένων καθολικών σχημάτων με τα ασυμπιεστα καθολικά σχήματα, καθώς μεταβάλλεται ο βαθμός συμπίεσης και για τις τρεις παραμέτρους ομοιότητας.

Παρακάτω παρουσιάζεται αναλυτικά ο τρόπος με τον οποίο πραγματοποιήθηκαν τα πειράματα κάθε κατηγορίας.

Τα παραπάνω πειράματα διενεργήθηκαν για δύο περιπτώσεις δικτύων ομότιμων κόμβων (P2P) με διαφορετικού περιεχομένου βάσεις δεδομένων. Συγκεκριμένα:

1. η πρώτη σειρά πειραμάτων έγινε για ένα αριθμό σχημάτων βάσεων δεδομένων με πληροφορίες για νοσοκομεία και
2. η δεύτερη για σχήματα βάσεων δεδομένων πανεπιστημίων.

Για κάθε περίπτωση κατασκευάσαμε ένα πρότυπο P2P δίκτυο με σχήματα βάσεων δεδομένων. Στη συνέχεια, κάνοντας κατάλληλες αλλαγές στα πρότυπα αυτά σχήματα (προσθαφαίρεση στοιχείων των σχημάτων ή σημασιολογικών σχέσεων μεταξύ τους), δημιουργήσαμε νέα δίκτυα P2P με διαφορετικά σχήματα βάσεων δεδομένων. Τα σχήματα αυτά κάθε δικτύου P2P αποτέλεσαν την είσοδο των πειραμάτων μας. Ως έξοδο, πήραμε το καθολικό σχήμα που προκύπτει από αυτά, καθώς και τα συμπιεσμένα καθολικά σχήματα, διαφόρων βαθμών συμπίεσης.

6.2.1 Ομοιότητα καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων

Στην κατηγορία πειραμάτων αυτή, συγκρίναμε την ομοιότητα μεταξύ καθολικού και αρχικών σχημάτων για έναν αριθμό δικτύων P2P με σχήματα βάσεων δεδομένων, συναρτήσεως της ομοιότητας μεταξύ των αρχικών σχημάτων. Πραγματοποιήθηκαν τρεις σειρές πειραμάτων, σε κάθε μία από τις οποίες, μεταβάλλαμε μία από τις τρεις παραμέτρους ομοιότητας (σχέσεις, ιδιότητες και κλειδιά), διατηρώντας τις άλλες δύο σταθερές ή, όπου αυτό δεν ήταν απόλυτα εφικτό, διατηρώντας τις σχεδόν σταθερές ή αφήνοντάς τις να ακολουθούν κάποια συνάρτηση. Ακολουθεί αναλυτική περιγραφή των τριών αυτών σειρών πειραμάτων.

6.2.1.1 Μεταβολή ομοιότητας σχέσεων (relations) των αρχικών σχημάτων

Για τη διεξαγωγή των πειραμάτων αυτών, δημιουργήσαμε από το πρότυπο P2P δίκτυο κάθε περίπτωσης (νοσοκομεία και πανεπιστήμια), τρία νέα P2P δίκτυα με διαφορετικά σχήματα βάσεων δεδομένων, αυξάνοντας ή μειώνοντας τον αριθμό των σχέσεων κάποιων από τα σχήματα των προτύπων. Συγκεκριμένα, για την περίπτωση των νοσοκομείων κατασκευάσαμε P2P δίκτυα με ποσοστά ομοιότητας σχέσεων μεταξύ των αρχικών σχημάτων: 88,5% (δίκτυο H1), 76,8% (πρότυπο δίκτυο H), 56,9% (H2) και 35,7% (H3) και για την περίπτωση των πανεπιστημίων δίκτυα με ποσοστά: 89,4% (U1), 70,2% (U2), 56,7% (πρότυπο U) και 39,3% (U3). Τα σχήματα των δικτύων αυτών παραθέτονται στην ενότητα 6.5.

Όπως είναι λογικό, η προσθήκη σε ένα σχήμα νέων σχέσεων που δεν υπάρχουν σε ένα δεύτερο, μειώνει το ποσοστό ομοιότητας των δύο σχημάτων, ενώ η προσθήκη σχέσεων που υπάρχουν στο δεύτερο, αυξάνει το ποσοστό αυτό. Αντίθετα, η αφαίρεση σχέσεων από το πρώτο σχήμα που υπάρχουν και στο δεύτερο, μειώνει το ποσοστό ομοιότητάς τους, ενώ η αφαίρεση μοναδικών σχέσεων αυξάνει το ποσοστό αυτό.

Η ομοιότητα μεταξύ των αρχικών σχημάτων ως προς τις σχέσεις υπολογίζεται ως εξής:

1. Αρχικά, υπολογίζουμε για κάθε αρχικό σχήμα, το ποσοστό των σχέσεων του που υπάρχουν και σε οποιαδήποτε άλλη σχέση. Για τον υπολογισμό αυτό, μετράμε μία μόνο φορά κάθε σχέση ανεξάρτητα από το πόσες φορές υπάρχει σε ένα σχήμα και κάθε σχέση που ταυτίζεται με μία άλλη σχέση του συγκεκριμένου σχήματος μέσω συσχέτισης ID. Επίσης, μία σχέση ενός σχήματος θεωρείται ότι υπάρχει σε ένα άλλο σχήμα, αν το τελευταίο έχει σχέση με το ίδιο όνομα ή σχέση που ταυτίζεται σημασιολογικά (μέσω συσχέτισης ID) με αυτή του πρώτου σχήματος.

2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των ποσοστών που υπολογίστηκαν στο προηγούμενο βήμα. Ο μέσος όρος αυτός αποτελεί το ποσοστό ομοιότητας ως προς τις σχέσεις μεταξύ των αρχικών σχημάτων.

Η ομοιότητα μεταξύ του καθολικού και των αρχικών σχημάτων ως προς τις σχέσεις υπολογίζεται ως εξής:

1. Υπολογίζουμε τι ποσοστό των σχέσεων του καθολικού σχήματος αποτελούν οι σχέσεις κάθε αρχικού σχήματος. Οι όμοιες σχέσεις μετρικούνται κι εδώ μία μόνο φορά.
2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των ποσοστών που υπολογίστηκαν στο προηγούμενο βήμα. Ο μέσος όρος αυτός αποτελεί το ποσοστό ομοιότητας ως προς τις σχέσεις του καθολικού με τα αρχικά σχήματα.

Είναι προφανές ότι καθώς μεταβάλλεται ο αριθμός των σχέσεων ενός σχήματος, μεταβάλλεται και ο αριθμός των ιδιοτήτων και των κλειδιών που αυτές περιέχουν. Παρόλα αυτά, ο αριθμός των ιδιοτήτων και επομένως και η ομοιότητα των σχημάτων ως προς αυτές, μπορεί να διατηρηθεί περίπου σταθερός, κατασκευάζοντας κατάλληλα τα σχήματα, δηλαδή προσθαφαιρώντας ιδιότητες όπου χρειάζεται. Αντίθετα, με τα κλειδιά είναι δύσκολο να συμβεί κάτι παρόμοιο, αφού είναι λογικό ότι καθώς μειώνονται οι σχέσεις, αφαιρούνται και τα αντίστοιχα κλειδιά. Γι' αυτό το λόγο, δε θεωρούμε ότι η ομοιότητα των κλειδιών παραμένει σταθερή, αλλά ότι είναι συνάρτηση της ομοιότητας των αρχικών σχημάτων.

6.2.1.2 Μεταβολή ομοιότητας ιδιοτήτων (*attributes*) των αρχικών σχημάτων

Όπως και στην προηγούμενη κατηγορία πειραμάτων, δημιουργήσαμε για κάθε περίπτωση (νοσοκομεία και πανεπιστήμια), τρία νέα δίκτυα κόμβων με διαφορετικά σχήματα βάσεων δεδομένων από τα πρότυπα. Αυτή τη φορά, προσθέσαμε ή αφαιρέσαμε ιδιότητες των σχέσεων των σχημάτων, χωρίς όμως να αλλάξουμε τις ίδιες τις σχέσεις ή τα κλειδιά τους, και επομένως χωρίς να μεταβληθεί η ομοιότητα των σχημάτων ως προς τις δύο αυτές παραμέτρους. Συγκεκριμένα, για την περίπτωση των νοσοκομείων κατασκευάσαμε P2P δίκτυα με ποσοστά ομοιότητας ιδιοτήτων μεταξύ των αρχικών σχημάτων: 87% (δίκτυο H4), 70,4% (πρότυπο δίκτυο H), 54,2% (H5) και 38,3% (H6) και για την περίπτωση των πανεπιστημίων δίκτυα με ποσοστά: 80,2% (U4), 64,5% (U5), 48,2% (πρότυπο U) και 32,4% (U6). Τα σχήματα των δικτύων αυτών παραθέτονται στην ενότητα 6.5.

Είναι προφανές ότι η προσθήκη σε ένα σχήμα νέων ιδιοτήτων που δεν υπάρχουν σε ένα άλλο σχήμα, μειώνει το ποσοστό ομοιότητας τους ως προς τις ιδιότητες, ενώ η προσθήκη ιδιοτήτων που υπάρχουν ήδη στο δεύτερο, αυξάνει το ποσοστό αυτό. Το αντίθετο συμβαίνει κατά την αφαίρεση ιδιοτήτων.

Η ομοιότητα μεταξύ των αρχικών σχημάτων ως προς τις ιδιότητες υπολογίζεται με όμοιο τρόπο με αυτόν του υπολογισμού της ομοιότητας ως προς τις σχέσεις:

1. Αρχικά, υπολογίζουμε για κάθε αρχικό σχήμα, το ποσοστό των ιδιοτήτων του που υπάρχουν και σε οποιαδήποτε άλλη σχέση. Για τον υπολογισμό αυτό, μετράμε μία μόνο φορά κάθε ιδιότητα, ανεξάρτητα από το πόσες φορές υπάρχει σε ένα σχήμα και τις ιδιότητες που ταυτίζονται μεταξύ τους, μέσω συσχέτισης ID. Επίσης, μία ιδιότητα θεωρείται ότι υπάρχει σε ένα άλλο σχήμα, αν το τελευταίο έχει ιδιότητα με το ίδιο όνομα ή ιδιότητα που ταυτίζεται σημασιολογικά (μέσω συσχέτισης ID) με αυτή του πρώτου σχήματος.
2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των ποσοστών που υπολογίστηκαν στο προηγούμενο βήμα. Ο μέσος όρος αυτός αποτελεί το ποσοστό ομοιότητας ως προς τις ιδιότητες μεταξύ των αρχικών σχημάτων.

Ομοίως με την ομοιότητα μεταξύ του καθολικού και των αρχικών σχημάτων ως προς τις σχέσεις, υπολογίζεται και η αντίστοιχη ομοιότητα ως προς τις ιδιότητες:

1. Υπολογίζουμε τι ποσοστό των ιδιοτήτων του καθολικού σχήματος αποτελούν οι ιδιότητες κάθε αρχικού σχήματος. Οι όμοιες ιδιότητες μετριοούνται κι εδώ μία μόνο φορά.
2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των ποσοστών που υπολογίστηκαν στο προηγούμενο βήμα. Ο μέσος όρος αυτός αποτελεί το ποσοστό ομοιότητας ως προς τις ιδιότητες του καθολικού με τα αρχικά σχήματα.

6.2.1.3 Μεταβολή ομοιότητας κλειδιών (*keys*) των αρχικών σχημάτων

Όπως και στις προηγούμενες κατηγορίες πειραμάτων, δημιουργήσαμε για κάθε περίπτωση (νοσοκομεία και πανεπιστήμια), νέα δίκτυα κόμβων με διαφορετικά σχήματα βάσεων δεδομένων από τα πρότυπα. Για τη δημιουργία των νέων σχημάτων δεν μεταβάλλαμε τις σχέσεις και τις ιδιότητες των σχημάτων, και επομένως και την ομοιότητα ως προς αυτές. Αντίθετα, αλλάξαμε ορισμένα κλειδιά και συσχετίσεις μεταξύ κλειδιών, ώστε να ταυτίζονται περισσότερα κλειδιά στις περιπτώσεις που θέλουμε να αυξήσουμε την ομοιότητα ως προς αυτά. Συγκεκριμένα, για την περίπτωση των νοσοκομείων κατασκευάσαμε P2P δίκτυα με ποσοστά ομοιότητας κλειδιών μεταξύ των αρχικών σχημάτων: 90,3% (δίκτυο H7), 72,8% (πρότυπο δίκτυο H), 60% (H8) και 50% (H9) και για την περίπτωση των πανεπιστημίων δίκτυα με ποσοστά: 85,7% (U7), 70% (πρότυπο U) , 61,4% (U8) και 47,1% (U9). Τα σχήματα των δικτύων αυτών παραθέτονται στην ενότητα 6.5.

Η ομοιότητα μεταξύ των αρχικών σχημάτων ως προς τα κλειδιά υπολογίζεται με όμοιο τρόπο με αυτόν του υπολογισμού της ομοιότητας ως προς τις σχέσεις και τις ιδιότητες:

1. Αρχικά, υπολογίζουμε για κάθε αρχικό σχήμα, το ποσοστό των κλειδιών του που υπάρχουν και σε οποιαδήποτε άλλη σχέση. Για τον υπολογισμό αυτό, μετράμε μία μόνο φορά κάθε κλειδί, ανεξάρτητα από το πόσες φορές εμφανίζεται σε ένα σχήμα καθώς και τα κλειδιά που ταυτίζονται μεταξύ τους, μέσω συσχέτισης ID. Επίσης, ένα κλειδί θεωρείται ότι υπάρχει σε ένα άλλο σχήμα, αν το τελευταίο έχει κλειδί με το ίδιο όνομα ή κλειδί που ταυτίζεται σημασιολογικά (μέσω συσχέτισης ID) με αυτό του πρώτου σχήματος.
2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των ποσοστών που υπολογίστηκαν στο προηγούμενο βήμα. Ο μέσος όρος αυτός αποτελεί το ποσοστό ομοιότητας ως προς τα κλειδιά μεταξύ των αρχικών σχημάτων.

Ομοίως με την ομοιότητα μεταξύ του καθολικού και των αρχικών σχημάτων ως προς τις σχέσεις και τις ιδιότητες, υπολογίζεται και η αντίστοιχη ομοιότητα ως προς τα κλειδιά:

1. Υπολογίζουμε τι ποσοστό των κλειδιών του καθολικού σχήματος αποτελούν τα κλειδιά κάθε αρχικού σχήματος. Τα όμοια κλειδιά μετριοούνται κι εδώ μία μόνο φορά.
2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των ποσοστών που υπολογίστηκαν στο προηγούμενο βήμα. Ο μέσος όρος αυτός αποτελεί το ποσοστό ομοιότητας ως προς τα κλειδιά του καθολικού με τα αρχικά σχήματα.

6.2.2 Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων

Σ' αυτή την κατηγορία πειραμάτων, εργαστήκαμε ακριβώς όπως στην προηγούμενη κατηγορία που περιγράψαμε στην παράγραφο 6.2.1. Η μοναδική διαφορά είναι ότι εδώ συγκρίναμε την ομοιότητα των αρχικών σχημάτων με το συμπιεσμένο καθολικό σχήμα και όχι απλά με το καθολικό. Ως συμπιεσμένο καθολικό σχήμα θεωρήσαμε αυτό που προέκυψε μετά από πολλαπλές συμπιέσεις, μέχρι να σταματήσει μόνος του ο αλγόριθμος συμπίεσης (παράγραφος 4.4). Δηλαδή δεν περιορίσαμε τη διαδικασία των επαναλαμβανόμενων συμπιέσεων ορίζοντας ένα σχετικά μεγάλο κατώφλι συμπίεσης. Αντίθετα, θέσαμε αρκετά μικρό το κατώφλι αυτό, ώστε οι συμπιέσεις να τελειώσουν όταν δεν υπάρχουν άλλες σχέσεις, οι οποίες να μπορούν να αφαιρεθούν, σύμφωνα με τον αλγόριθμο της παραγράφου 4.4.

Έτσι, συγκρίναμε την ομοιότητα μεταξύ συμπιεσμένου καθολικού και αρχικών σχημάτων για έναν αριθμό δικτύων κόμβων με σχήματα βάσεων δεδομένων, συναρτήσεως της ομοιότητας μεταξύ των αρχικών σχημάτων. Πραγματοποιήθηκαν τρεις σειρές πειραμάτων, σε κάθε μία από τις οποίες, μεταβάλλαμε μία από τις τρεις παραμέτρους ομοιότητας (σχέσεις, ιδιότητες και κλειδιά), διατηρώντας τις άλλες δύο σταθερές ή σχεδόν σταθερές, όπως περιγράψαμε

παραπάνω. Οι τρεις αυτές σειρές πειραμάτων έγιναν ακριβώς με τον ίδιο τρόπο με αυτές της προηγούμενης κατηγορίας πειραμάτων, και οι οποίες περιγράφηκαν στις παραγράφους 6.2.1.1 – 6.2.1.3.

6.2.3 Σύγκριση ομοιοτήτων με βάση το βαθμό συμπίεσης

6.2.3.1 Ομοιότητα αρχικών σχημάτων με το τελικό (καθολικό ή συμπιεσμένο) για διάφορους βαθμούς συμπίεσης

Στο πρώτο μέρος της κατηγορίας πειραμάτων αυτής, συγκρίναμε την ομοιότητα των αρχικών σχημάτων με το καθολικό και τα συμπιεσμένα αυξανόμενων βαθμών συμπίεσης. Ο βαθμός συμπίεσης δείχνει το πλήθος των σχέσεων που αφαιρούνται ώστε να μειωθεί το μέγεθος του σχήματος. Έγιναν τρεις σειρές πειραμάτων, μία για κάθε παράμετρο ομοιότητας.

6.2.3.2 Ομοιότητα συμπιεσμένου με καθολικό καθώς μεταβάλλεται ο βαθμός συμπίεσης

Σ' αυτό το μέρος της κατηγορίας, συγκρίναμε την ομοιότητα μεταξύ ενός συμπιεσμένου καθολικού σχήματος και του αντίστοιχου ασυμπιεστού καθολικού σχήματος από το οποίο προήλθε, καθώς μεταβάλλεται ο βαθμός συμπίεσης.

Για τη διεξαγωγή των πειραμάτων αυτών, χρησιμοποιήσαμε το πρότυπο P2P δίκτυο κόμβων της περίπτωσης των νοσοκομείων, το οποίο μπορεί από κατασκευής του να συμπιεστεί πάνω από μία φορά, και συγκεκριμένα 3 φορές.

Εκτελέσαμε τρεις σειρές μετρήσεων, σε κάθε μία από τις οποίες ο υπολογισμός της ομοιότητας μεταξύ του συμπιεσμένου καθολικού σχήματος και του αντίστοιχου ασυμπιεστού, έγινε ως προς μία από τις τρεις παραμέτρους ομοιότητας. Η ομοιότητα αυτή είναι το ποσοστό των σχέσεων, ιδιοτήτων ή κλειδιών (ανάλογα με την περίπτωση) του καθολικού σχήματος, τα οποία υπάρχουν και στο συμπιεσμένο.

6.3 Αποτελέσματα

Στην παράγραφο αυτή παρουσιάζουμε τα αποτελέσματα που προέκυψαν από τα πειράματα που εκτελέσαμε, με τον τρόπο που περιγράφηκε στις προηγούμενες παραγράφους.

6.3.1 Ομοιότητα καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων

Στους πίνακες που ακολουθούν, η δεύτερη και τρίτη στήλη αποτελούν τα ποσοστά εισόδου των πειραμάτων και η τρίτη και τέταρτη τα ποσοστά εξόδου.

6.3.1.1 Μεταβολή ομοιότητας σχέσεων (relations) των αρχικών σχημάτων

6.3.1.1.1 P2P Δίκτυο νοσοκομείων

Όπως αναφέραμε στην παράγραφο 6.2.1.1, για την περίπτωση των νοσοκομείων, κατασκευάσαμε από το P2P δίκτυο H, τα H1, H2 και H3, τα σχήματα των οποίων παραθέτονται στην ενότητα 6.5. Η ομοιότητα των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκε περίπου σταθερή και ίση με $70,4 \pm 1\%$, ενώ η ομοιότητα των κλειδιών των αρχικών σχημάτων τους είναι συνάρτηση της ομοιότητας των αρχικών σχέσεων (f1). Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

H1

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	4/5	80,0	5/16	31,3
Schema 2	5/8	62,5	8/16	50,0
Schema 3	6/6	100,0	6/16	37,5
Schema 4	11/11	100,0	11/16	68,8
Schema 5	10/10	100,0	10/16	62,5
MO		88,5		50,0

Πίνακας 6.1 Δίκτυο P2P H1: Ομοιότητα ως προς τις σχέσεις

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	11/14	78,6	4/5	80,0
Schema 2	13/18	72,2	3/7	42,9
Schema 3	13/18	72,2	6/6	100,0
Schema 4	27/40	67,5	10/11	90,9
Schema 5	22/36	61,1	9/10	90,0
MO		70,3		80,8

Πίνακας 6.2 Δίκτυο P2P H1: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

H

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	4/5	80,0	5/16	31,3
Schema 2	5/8	62,5	8/16	50,0
Schema 3	6/6	100,0	6/16	37,5
Schema 4	6/7	85,7	7/16	43,8
Schema 5	5/9	55,6	9/16	56,3
MO		76,8		43,8

Πίνακας 6.3 Δίκτυο P2P H: Ομοιότητα ως προς τις σχέσεις

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	13/15	86,7	4/5	80,0
Schema 2	15/20	75,0	3/7	42,9
Schema 3	15/19	78,9	6/6	100,0
Schema 4	13/22	59,1	6/7	85,7
Schema 5	11/21	52,4	5/9	55,6
MO		70,4		72,8

Πίνακας 6.4 Δίκτυο P2P H: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

H2

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	3/4	75,0	4/18	22,2
Schema 2	3/6	50,0	6/18	33,3
Schema 3	4/6	66,7	6/18	33,3
Schema 4	3/6	50,0	6/18	33,3
Schema 5	3/7	42,9	7/18	38,9
MO		56,9		32,2

Πίνακας 6.5 Δίκτυο P2P H2: Ομοιότητα ως προς τις σχέσεις

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	11/14	78,6	3/4	75,0
Schema 2	13/18	72,2	2/5	40,0
Schema 3	13/18	72,2	4/6	66,7
Schema 4	12/17	70,6	4/5	80,0
Schema 5	8/14	57,1	3/7	42,9
MO		70,1		60,9

Πίνακας 6.6 Δίκτυο P2P H2: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

H3

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	1/2	50,0	2/18	11,1
Schema 2	1/5	20,0	5/18	27,8
Schema 3	2/5	40,0	5/18	27,8
Schema 4	2/5	40,0	5/18	27,8
Schema 5	2/7	28,6	7/18	38,9
MO		35,7		26,7

Πίνακας 6.7 Δίκτυο P2P H3: Ομοιότητα ως προς τις σχέσεις

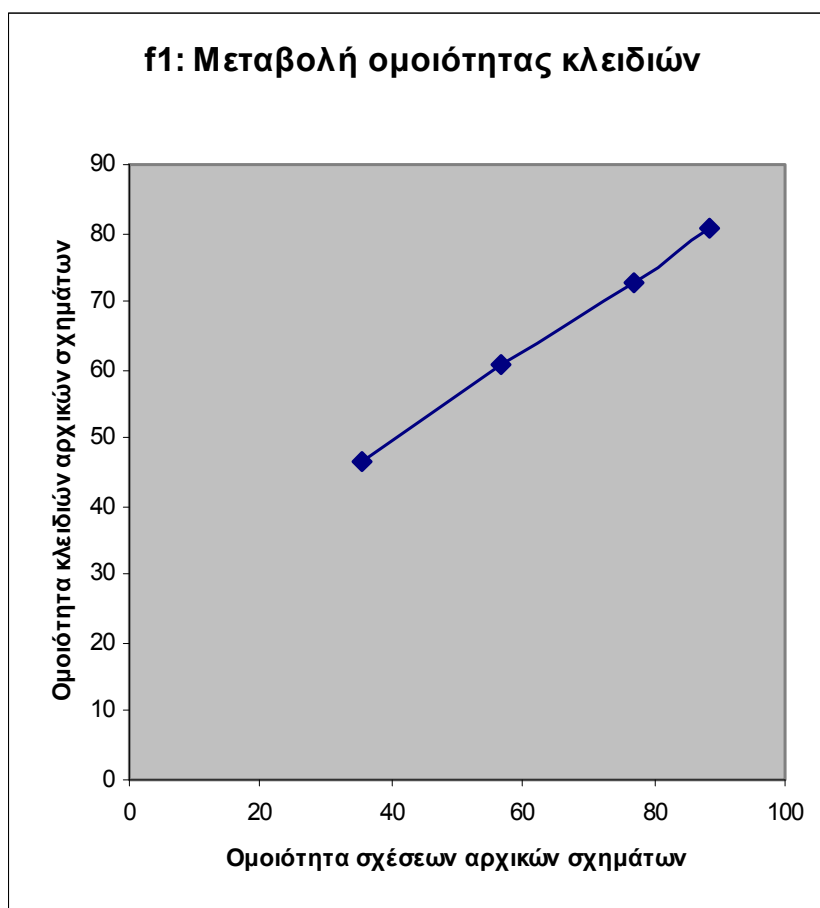
	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	8/10	80,0	1/2	50,0
Schema 2	12/15	80,0	1/4	25,0
Schema 3	8/15	53,3	2/5	40,0
Schema 4	11/15	73,3	3/4	75,0
Schema 5	11/8	61,1	3/7	42,9
MO		69,6		46,6

Πίνακας 6.8 Δίκτυο P2P H3: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

Παρατηρούμε ότι πράγματι το ποσοστό ομοιότητας των ιδιοτήτων των αρχικών σχημάτων διατηρείται σχεδόν σταθερό (70,4%) με απόκλιση μικρότερη από $\pm 1\%$. Αντίθετα η ομοιότητα των κλειδιών των αρχικών σχημάτων συναρτήσκει της ομοιότητας των σχέσεων δίνεται από τη συνάρτηση f_1 . Παρουσιάζουμε τον πίνακα και τη γραφική παράσταση της συνάρτησης αυτής. Όπως φαίνεται, το γράφημα έχει τη μορφή ευθείας, δηλαδή η ομοιότητα των κλειδιών αυξάνεται ανάλογα με αυτή των σχέσεων.

Ποσοστό ομοιότητας σχέσεων (%)	Ποσοστό ομοιότητα κλειδιών (%)
88,5	80,8
76,8	72,8
56,9	60,9
35,7	46,6

Πίνακας 6.9 Ομοιότητα κλειδιών συναρτήσκει ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων



Σχήμα 6.1 Συνάρτηση f1: Ομοιότητα κλειδιών συναρτήσεως ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων

6.3.1.1.2 P2P Δίκτυο πανεπιστημίων

Όπως αναφέραμε στην παράγραφο 6.2.1.1, για την περίπτωση των πανεπιστημίων, κατασκευάσαμε από το P2P δίκτυο U, τα U1, U2 και U3, τα σχήματα των οποίων παραθέτονται στην ενότητα 6.5. Η ομοιότητα των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκε περίπου σταθερή και ίση με $48,2 \pm 1\%$, ενώ η ομοιότητα των κλειδιών των αρχικών σχημάτων τους είναι συνάρτηση της ομοιότητας των αρχικών σχέσεων (f2). Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

U1

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	7/10	70,0	10/13	76,9
Schema 2	7/8	87,5	8/13	61,5
Schema 3	5/5	100,0	5/13	38,5
Schema 4	7/7	100,0	7/13	53,8
MO		89,4		57,7

Πίνακας 6.10 Δίκτυο P2P U1: Ομοιότητα ως προς τις σχέσεις

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	18/38	47,4	6/8	75,0
Schema 2	17/38	44,7	7/8	87,5
Schema 3	8/23	34,8	5/6	83,3
Schema 4	18/26	69,2	6/7	85,7
MO		49,0		82,9

Πίνακας 6.11 Δίκτυο P2P U1: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

U2

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	6/9	66,7	9/16	56,3
Schema 2	5/7	71,4	7/16	43,8
Schema 3	4/7	57,1	7/16	43,8
Schema 4	6/7	85,7	7/16	43,8
MO		70,2		46,9

Πίνακας 6.12 Δίκτυο P2P U2: Ομοιότητα ως προς τις σχέσεις

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	17/32	53,1	6/8	75,0
Schema 2	14/33	42,4	5/7	71,4
Schema 3	6/23	26,1	4/5	80,0
Schema 4	16/22	72,7	5/7	71,4
MO		48,6		74,5

Πίνακας 6.13 Δίκτυο P2P U2: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

U

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	4/8	50,0	8/18	44,4
Schema 2	4/7	57,1	7/18	38,9
Schema 3	4/7	57,1	7/18	38,9
Schema 4	5/8	62,5	8/18	44,4
MO		56,7		41,7

Πίνακας 6.14 Δίκτυο P2P U: Ομοιότητα ως προς τις σχέσεις

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	17/32	57,7	5/7	71,4
Schema 2	14/33	36,4	4/7	57,1
Schema 3	6/23	26,1	4/5	80,0
Schema 4	16/22	72,7	5/7	71,4
MO		48,2		70,0

Πίνακας 6.15 Δίκτυο P2P U: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

U3

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	3/7	42,9	7/20	35,0
Schema 2	3/7	42,9	7/20	35,0
Schema 3	2/7	28,6	7/20	35,0
Schema 4	3/7	42,9	7/20	35,0
MO		39,3		35,0

Πίνακας 6.16 Δίκτυο P2P U3: Ομοιότητα ως προς τις σχέσεις

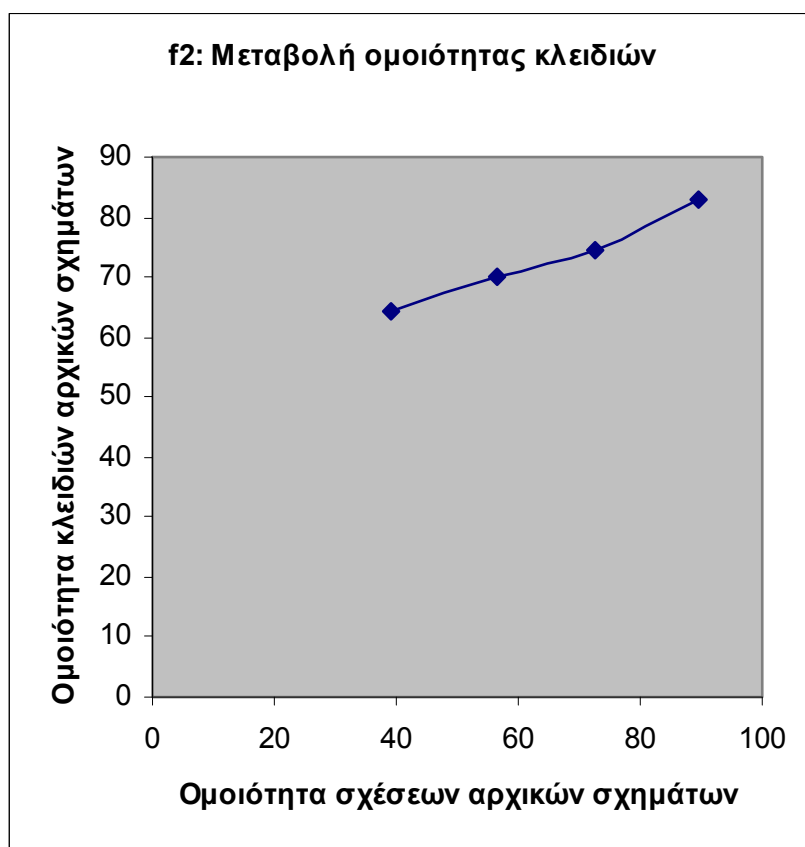
	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)
Schema 1	14/24	58,3	5/7	71,4
Schema 2	12/33	36,4	3/7	42,9
Schema 3	5/23	21,7	3/5	60,0
Schema 4	15/20	75,0	5/6	83,3
MO		47,9		64,4

Πίνακας 6.17 Δίκτυο P2P U3: Ομοιότητα αρχικών σχημάτων ως προς τις ιδιότητες και τα κλειδιά

Παρατηρούμε ότι πράγματι το ποσοστό ομοιότητας των ιδιοτήτων των αρχικών σχημάτων διατηρείται σχεδόν σταθερό (48,2%) με απόκλιση μικρότερη από $\pm 1\%$. Αντίθετα η ομοιότητα των κλειδιών των αρχικών σχημάτων συναρτήσκει της ομοιότητας των σχέσεων δίνεται από τη συνάρτηση f_2 . Παρουσιάζουμε τον πίνακα και τη γραφική παράσταση της συνάρτησης αυτής. Όπως φαίνεται, και αυτό το γράφημα έχει τη μορφή ευθείας, δηλαδή η ομοιότητα των κλειδιών αυξάνεται ανάλογα με αυτή των σχέσεων.

Ποσοστό ομοιότητας σχέσεων (%)	Ποσοστό ομοιότητα κλειδιών (%)
89,4	82,9
72,8	74,5
56,7	70,0
39,3	64,4

Πίνακας 6.18 Ομοιότητα κλειδιών συναρτήσει ομοιότητας σχέσεων για τα P2P δίκτυα των πανεπιστημίων



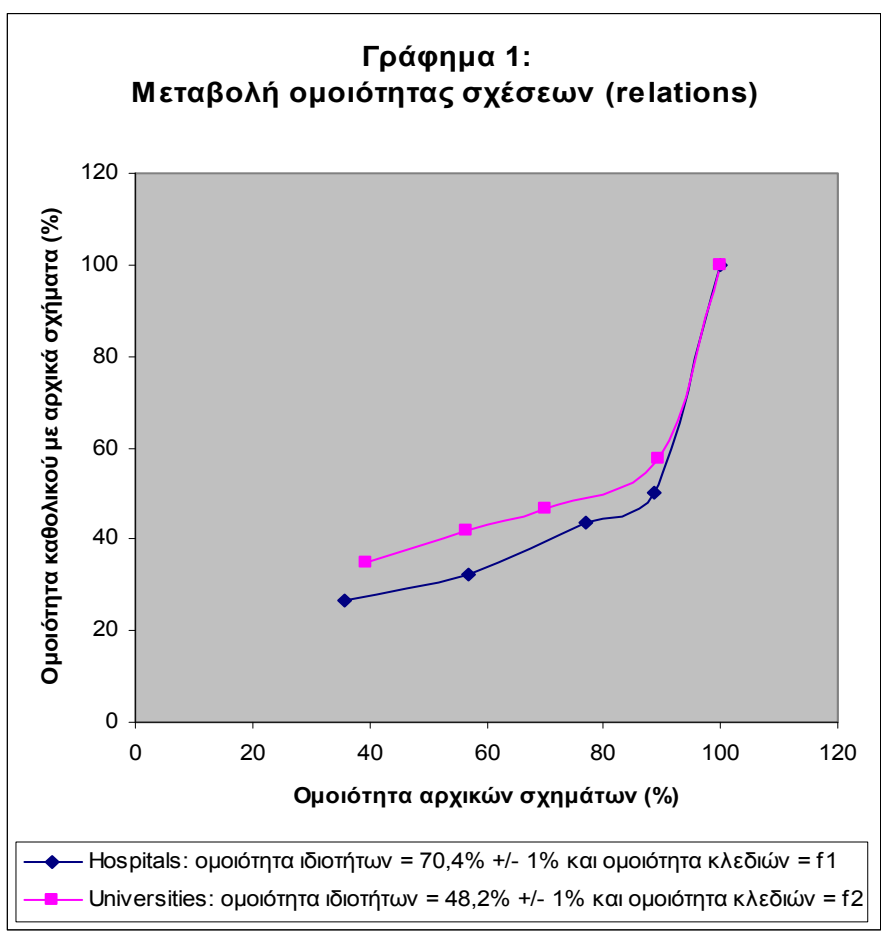
Σχήμα 6.2 Συνάρτηση f2: Ομοιότητα κλειδιών συναρτήσει ομοιότητας σχέσεων για τα P2P δίκτυα των πανεπιστημίων

6.3.1.1.3 Αποτελέσματα σύγκρισης ομοιοτήτων ως προς τις σχέσεις

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του καθολικού με τα αρχικά σχήματα ως προς τις σχέσεις, καθώς μεταβάλλεται η ομοιότητα των αρχικών σχημάτων μεταξύ τους ως προς τις σχέσεις:

Ομοιότητα αρχικών σχημάτων νοσοκομείων (%)	Ομοιότητα καθολικού με αρχικά σχήματα νοσοκομείων (%)	Ομοιότητα αρχικών σχημάτων πανεπιστημίων (%)	Ομοιότητα καθολικού με αρχικά σχήματα πανεπιστημίων (%)
100	100,0	100,0	100,0
88,5	50,0	89,4	57,7
76,8	43,8	70,2	46,9
56,9	32,2	56,7	41,7
35,7	26,7	39,3	35,0

Πίνακας 6.19 Ομοιότητα σχέσεων αρχικών σχημάτων και καθολικού για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων



Σχήμα 6.3 Μεταβολή ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

6.3.1.2 Μεταβολή ομοιότητας ιδιοτήτων (attributes) των αρχικών σχημάτων

6.3.1.2.1 P2P Δίκτυο νοσοκομείων

Όπως αναφέραμε στην παράγραφο 6.2.1.2, για την περίπτωση των νοσοκομείων, κατασκευάσαμε από το δίκτυο κόμβων H, τα H4, H5 και H6, τα σχήματα των οποίων παραθέτονται στην ενότητα 6.5. Οι ομοιότητες των σχέσεων και των κλειδιών των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 76,8% και 72,8% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

H4

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	13/15	86,7	15/29	51,7
Schema 2	15/17	88,2	17/29	58,6
Schema 3	15/16	93,8	16/29	55,2
Schema 4	13/14	92,9	14/29	48,3
Schema 5	11/15	73,3	15/29	51,7
MO		87,0		53,1

Πίνακας 6.20 Δίκτυο P2P H4: Ομοιότητα ως προς τις ιδιότητες

H

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	13/15	86,7	15/44	34,1
Schema 2	15/20	75,0	20/44	45,5
Schema 3	15/19	78,9	19/44	43,2
Schema 4	13/22	59,1	22/44	50,0
Schema 5	11/21	52,4	21/44	47,7
MO		70,4		44,1

Πίνακας 6.21 Δίκτυο P2P H: Ομοιότητα ως προς τις ιδιότητες

H5

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	11/20	55,0	20/69	29,0
Schema 2	15/23	65,2	23/69	33,3
Schema 3	14/22	63,6	22/69	31,9
Schema 4	12/24	50,0	24/69	34,8
Schema 5	10/27	37,0	27/69	39,1
MO		54,2		33,6

Πίνακας 6.22 Δίκτυο P2P H5: Ομοιότητα ως προς τις ιδιότητες

H6

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	10/25	40,0	24/107	22,4
Schema 2	13/30	43,3	29/107	27,1
Schema 3	13/30	43,3	30/107	28,0
Schema 4	12/32	37,5	30/107	28,0
Schema 5	9/33	27,3	33/107	30,8
MO		38,3		27,3

Πίνακας 6.23 Δίκτυο P2P H6: Ομοιότητα ως προς τις ιδιότητες

6.3.1.2.2 P2P Δίκτυο πανεπιστημίων

Όπως αναφέραμε στην παράγραφο 6.2.1.2, για την περίπτωση των πανεπιστημίων, κατασκευάσαμε από το δίκτυο κόμβων U, τα U4, U5 και U6, τα σχήματα των οποίων παραθέτονται στην ενότητα 6.5. Οι ομοιότητες των σχέσεων και των κλειδιών των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 56,7% και 70% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

U4

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	16/21	76,2	21/35	60,0
Schema 2	16/21	76,2	22/35	62,9
Schema 3	16/19	84,2	19/35	54,3
Schema 4	16/19	84,2	19/35	54,3
MO		80,2		57,9

Πίνακας 6.24 Δίκτυο P2P U4: Ομοιότητα ως προς τις ιδιότητεςU5

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	15/22	68,2	22/38	57,9
Schema 2	12/20	60,0	20/38	52,6
Schema 3	6/12	50,0	12/38	31,6
Schema 4	16/20	80,0	20/38	52,6
MO		64,5		48,7

Πίνακας 6.25 Δίκτυο P2P U5: Ομοιότητα ως προς τις ιδιότητεςU

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	15/26	57,7	26/70	37,1
Schema 2	12/33	36,4	33/70	47,1
Schema 3	6/23	26,1	23/70	32,9
Schema 4	16/22	72,7	22/70	31,4
MO		48,2		37,1

Πίνακας 6.26 Δίκτυο P2P U: Ομοιότητα ως προς τις ιδιότητες

U6

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	8/25	32,0	25/70	35,7
Schema 2	5/26	19,2	26/70	37,1
Schema 3	5/22	22,7	22/70	31,4
Schema 4	10/18	55,6	18/70	25,7
MO		32,4		32,5

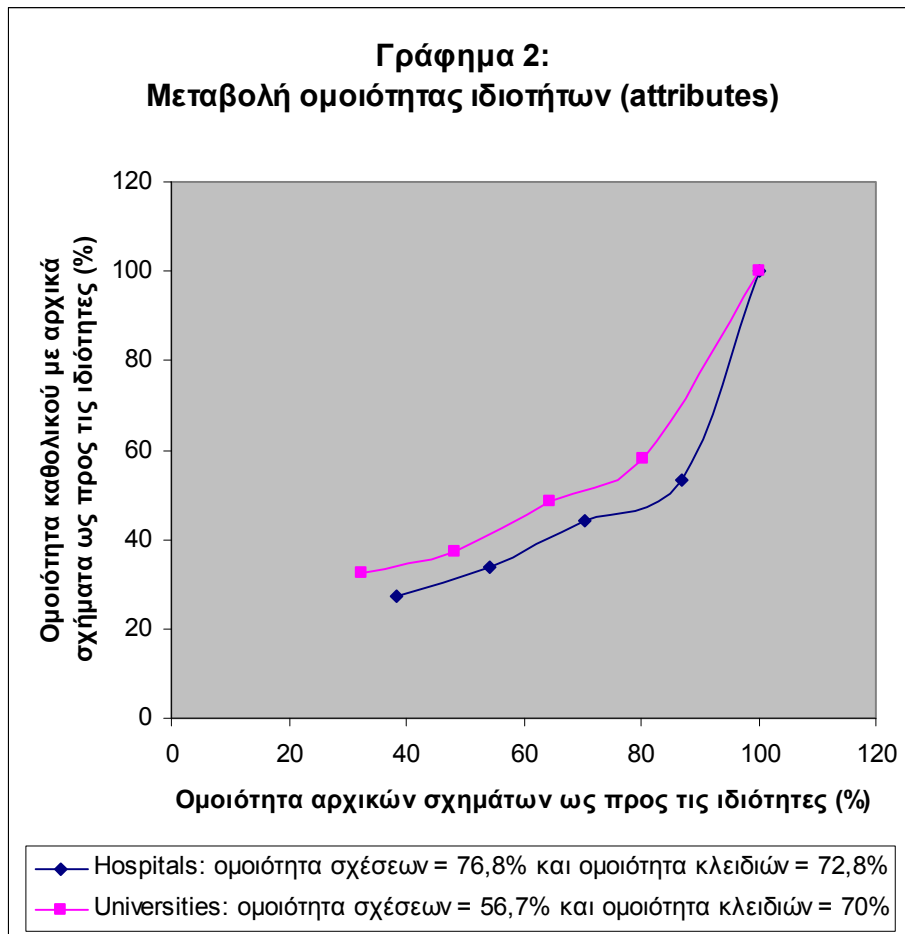
Πίνακας 6.27 Δίκτυο P2P U6: Ομοιότητα ως προς τις ιδιότητες

6.3.1.2.3 Αποτελέσματα σύγκρισης ομοιοτήτων ως προς τις ιδιότητες

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του καθολικού με τα αρχικά σχήματα ως προς τις ιδιότητες, καθώς μεταβάλλεται η ομοιότητα των αρχικών σχημάτων μεταξύ τους ως προς τις ιδιότητες:

Ομοιότητα αρχικών σχημάτων νοσοκομείων (%)	Ομοιότητα καθολικού με αρχικά σχήματα νοσοκομείων (%)	Ομοιότητα αρχικών σχημάτων πανεπιστημίων (%)	Ομοιότητα καθολικού με αρχικά σχήματα πανεπιστημίων (%)
100,0	100,0	100,0	100,0
87,0	53,1	80,2	57,9
70,4	44,1	64,5	48,7
54,2	33,6	48,2	37,1
38,3	27,3	32,4	32,5

Πίνακας 6.28 Ομοιότητα ιδιοτήτων αρχικών σχημάτων και καθολικού για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων



Σχήμα 6.4 Μεταβολή ομοιότητας ιδιοτήτων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

6.3.1.3 Μεταβολή ομοιότητας κλειδιών (keys) των αρχικών σχημάτων

6.3.1.3.1 P2P Δίκτυο νοσοκομείων

Όπως αναφέραμε στην παράγραφο 6.2.1.3, για την περίπτωση των νοσοκομείων, κατασκευάσαμε από το δίκτυο κόμβων H, τα H7, H8 και H9, τα σχήματα των οποίων παραθέτονται στην ενότητα 6.5. Οι ομοιότητες των σχέσεων και των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 76,8% και 70,4% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

H7

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	4/11	36,4
Schema 2	6/7	85,7	5/11	45,5
Schema 3	6/6	100,0	3/11	27,3
Schema 4	7/7	100,0	4/11	36,4
Schema 5	9/9	100,0	3/11	27,3
MO		93,1		34,5

Πίνακας 6.29 Δίκτυο P2P H7: Ομοιότητα ως προς τα κλειδιάH

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	3/14	21,4
Schema 2	3/7	42,9	5/14	35,7
Schema 3	6/6	100,0	2/14	14,3
Schema 4	6/7	85,7	3/14	21,4
Schema 5	5/9	55,6	6/14	42,9
MO		72,8		27,1

Πίνακας 6.30 Δίκτυο P2P H: Ομοιότητα ως προς τα κλειδιάH8

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	3/14	21,4
Schema 2	3/7	42,9	4/14	28,6
Schema 3	3/6	50,0	2/14	14,3
Schema 4	5/7	71,4	3/14	21,4
Schema 5	5/9	55,6	6/14	42,9
MO		60,0		25,7

Πίνακας 6.31 Δίκτυο P2P H8: Ομοιότητα ως προς τα κλειδιά

H9

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	1/14	7,1
Schema 2	3/7	42,9	2/14	14,3
Schema 3	0/6	0,0	0/14	0,0
Schema 4	5/7	71,4	2/14	14,3
Schema 5	5/9	55,6	4/14	28,6
MO		50,0		12,9

Πίνακας 6.32 Δίκτυο P2P H9: Ομοιότητα ως προς τα κλειδιά

6.3.1.3.2 P2P Δίκτυο πανεπιστημίων

Όπως αναφέραμε στην παράγραφο 6.2.1.3, για την περίπτωση των πανεπιστημίων κατασκευάσαμε από το δίκτυο κόμβων U, τα U7, U8 U9 και U10, τα σχήματα των οποίων παραθέτονται στην ενότητα 6.5. Οι ομοιότητες των σχέσεων και των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 56,7% και 48,2% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

U7

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	5/7	71,4	6/11	54,5
Schema 2	6/7	85,7	4/11	36,4
Schema 3	5/5	100,0	4/11	36,4
Schema 4	6/7	85,7	5/11	45,5
MO		85,7		43,2

Πίνακας 6.33 Δίκτυο P2P U7: Ομοιότητα ως προς τα κλειδιά

U

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	5/7	71,4	4/13	30,8
Schema 2	4/7	57,1	4/13	30,8
Schema 3	4/5	80,0	3/13	23,1
Schema 4	5/7	71,4	4/13	30,8
MO		70,0		28,8

Πίνακας 6.34 Δίκτυο P2P U: Ομοιότητα ως προς τα κλειδιάU8

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	5/7	71,4	4/14	28,6
Schema 2	3/7	42,9	4/14	28,6
Schema 3	3/5	60,0	4/14	28,6
Schema 4	5/7	71,4	4/14	28,6
MO		61,4		28,6

Πίνακας 6.35 Δίκτυο P2P U8: Ομοιότητα ως προς τα κλειδιάU9

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/7	57,1	4/15	26,7
Schema 2	0/7	0,0	4/15	26,7
Schema 3	3/5	60,0	4/15	26,7
Schema 4	5/7	71,4	4/15	26,7
MO		47,1		26,7

Πίνακας 6.36 Δίκτυο P2P U9: Ομοιότητα ως προς τα κλειδιά

U10

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/7	57,1	4/16	25,0
Schema 2	0/7	0,0	4/16	25,0
Schema 3	0/5	0,0	3/16	18,8
Schema 4	4/7	57,1	3/16	18,8
MO		28,6		21,9

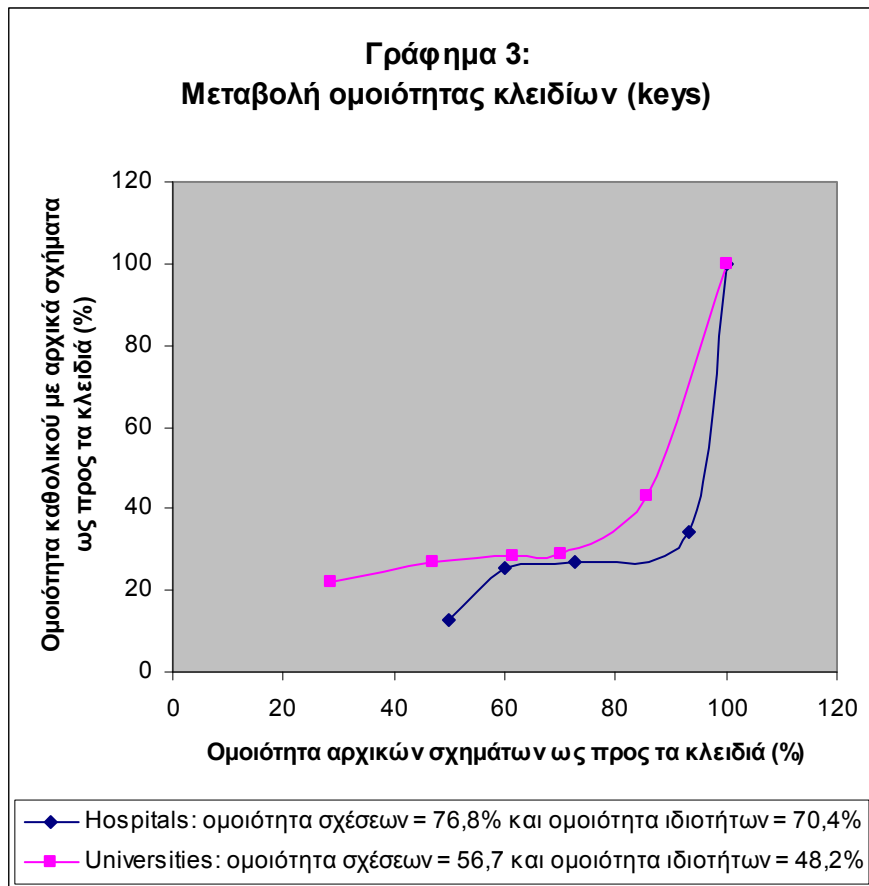
Πίνακας 6.37 Δίκτυο P2P U10: Ομοιότητα ως προς τα κλειδιά

6.3.1.3.3 Αποτελέσματα σύγκρισης ομοιοτήτων ως προς τα κλειδιά

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του καθολικού με τα αρχικά σχήματα ως προς τα κλειδιά, καθώς μεταβάλλεται η ομοιότητα των αρχικών σχημάτων μεταξύ τους ως προς τα κλειδιά:

Ομοιότητα αρχικών σχημάτων νοσοκομείων (%)	Ομοιότητα καθολικού με αρχικά σχήματα νοσοκομείων (%)	Ομοιότητα αρχικών σχημάτων πανεπιστημίων (%)	Ομοιότητα καθολικού με αρχικά σχήματα πανεπιστημίων (%)
100,0	100,0	100,0	100,0
93,1	34,5	85,7	43,2
72,8	27,1	70,0	28,8
60,0	25,7	61,4	28,6
50,0	12,9	47,1	26,7
		28,6	21,9

Πίνακας 6.38 Ομοιότητα κλειδιών αρχικών σχημάτων και καθολικού για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

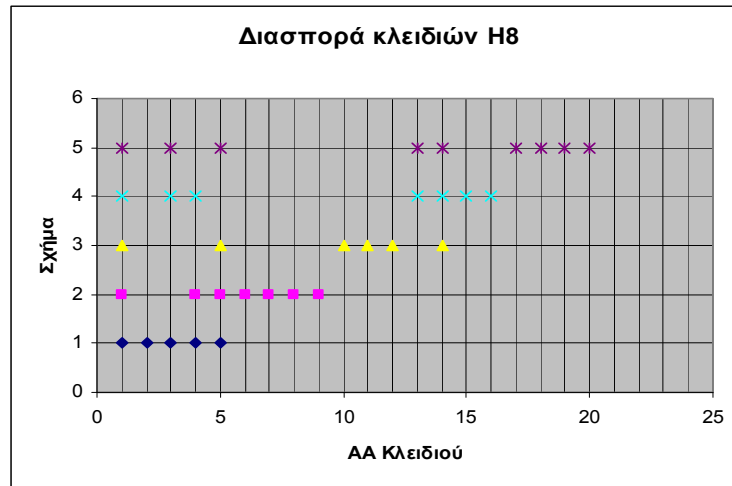


Σχήμα 6.5 Μεταβολή ομοιότητας κλειδιών για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

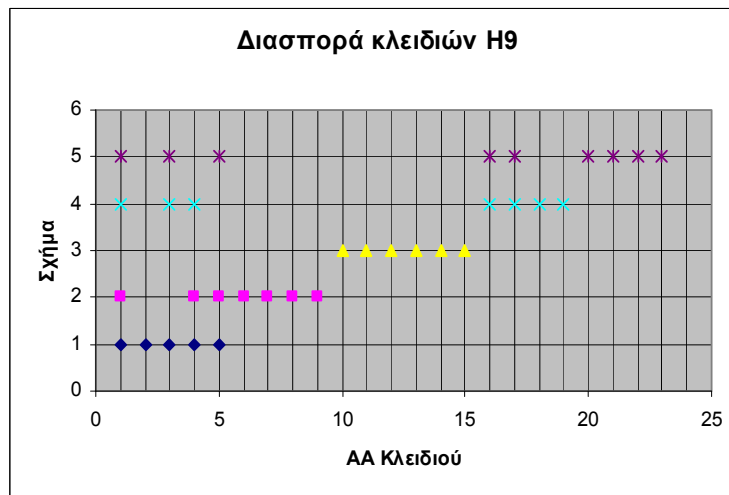
6.3.1.3.4 Διασπορά κλειδιών

Στο σημείο αυτό παραθέτουμε μερικά διαγράμματα που απεικονίζουν τη διασπορά των κλειδιών στα σχήματα μερικών P2P δικτύων. Συγκεκριμένα, στον άξονα των x, απεικονίζονται τα διαφορετικά κλειδιά που υπάρχουν σε ένα P2P δίκτυο και στον άξονα των y τα διάφορα σχήματα του δικτύου αυτού. Όταν ένα συγκεκριμένο κλειδί υπάρχει σε ένα σχήμα,, αυτό απεικονίζεται με ένα σημείο.

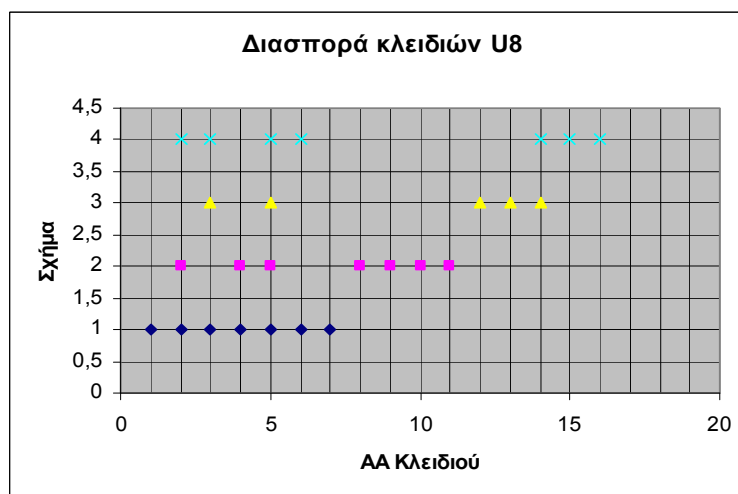
Παρατηρούμε ότι το P2P δίκτυο H8 που έχει μεγαλύτερο ποσοστό ομοιότητας αρχικών σχέσεων από το H9, εμφανίζει μεγαλύτερη επικάλυψη κλειδιών στα σχήματα, δηλαδή περισσότερες σχέσεις έχουν κοινά κλειδιά, ενώ τα κλειδιά του H9 είναι πιο «δισεπαρμένα» στις σχέσεις. Το ίδιο ισχύει και για τα P2P δίκτυα U8 και U9, όπου το πρώτο έχει μεγαλύτερο ποσοστό ομοιότητας αρχικών σχέσεων.



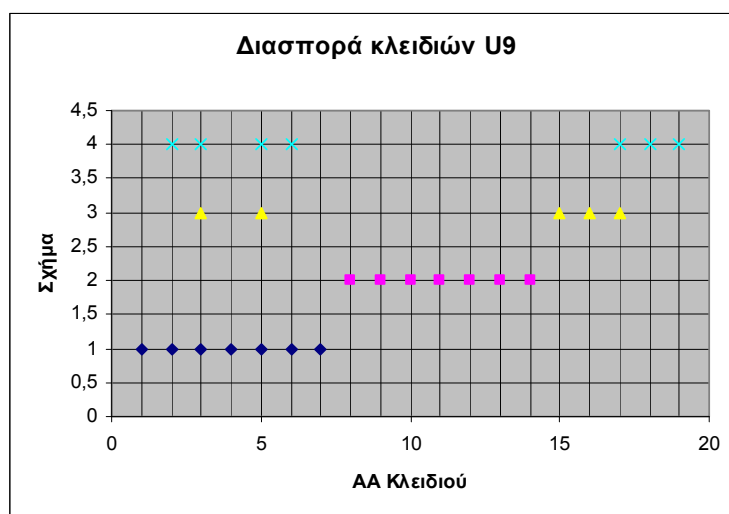
Σχήμα 6.6 Διασπορά κλειδιών P2P δικτύου H8



Σχήμα 6.7 Διασπορά κλειδιών P2P δικτύου H9



Σχήμα 6.8 Διασπορά κλειδιών P2P δικτύου U9



Σχήμα 6.9 Διασπορά κλειδιών P2P δικτύου U9

6.3.2 Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων

Στους πίνακες που ακολουθούν, η δεύτερη και τρίτη στήλη αποτελούν τα ποσοστά εισόδου των πειραμάτων και η τρίτη και τέταρτη τα ποσοστά εξόδου. Όπως αναφέραμε και στην παράγραφο 6.2.2, η συμπίεση εδώ είναι η μέγιστη συμπίεση που μπορεί να γίνει σε κάθε καθολικό σχήμα.

6.3.2.1 Μεταβολή ομοιότητας σχέσεων (relations) των αρχικών σχημάτων

6.3.2.1.1 P2P Δίκτυο νοσοκομείων

Για την περίπτωση των νοσοκομείων, χρησιμοποιήσαμε και σε αυτή την κατηγορία πειραμάτων τα δίκτυα κόμβων H, H1, H2 και H3. Η ομοιότητα των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών είναι ίση με $70,4 \pm 1\%$, ενώ η ομοιότητα των κλειδιών των αρχικών σχημάτων τους είναι συνάρτηση της ομοιότητας των αρχικών σχέσεων (f1), όπως και στο πείραμα 6.2.1.1, εφόσον χρησιμοποιούμε τα ίδια σχήματα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

H1

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	4/5	80,0	5/13	38,5
Schema 2	5/8	62,5	8/13	61,5
Schema 3	6/6	100,0	6/13	46,2
Schema 4	11/11	100,0	11/13	84,6
Schema 5	10/10	100,0	10/13	76,9
MO		88,5		61,5

Πίνακας 6.39 Δίκτυο P2P H1: Ομοιότητα ως προς τις σχέσειςH

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	4/5	80,0	5/13	38,5
Schema 2	5/8	62,5	8/13	61,5
Schema 3	6/6	100,0	6/13	46,2
Schema 4	6/7	85,7	7/13	53,8
Schema 5	5/9	55,6	9/13	69,2
MO		76,8		53,8

Πίνακας 6.40 Δίκτυο P2P H: Ομοιότητα ως προς τις σχέσειςH2

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	3/4	75,0	4/15	26,7
Schema 2	3/6	50,0	6/15	40,0
Schema 3	4/6	66,7	6/15	40,0
Schema 4	3/6	50,0	6/15	40,0
Schema 5	3/7	42,9	7/15	46,7
MO		56,9		38,7

Πίνακας 6.41 Δίκτυο P2P H2: Ομοιότητα ως προς τις σχέσεις

H3

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	1/2	50,0	2/15	13,3
Schema 2	1/5	20,0	5/15	33,3
Schema 3	2/5	40,0	5/15	33,3
Schema 4	2/5	40,0	5/15	33,3
Schema 5	2/7	28,6	7/15	46,7
MO		35,7		32,0

Πίνακας 6.42 Δίκτυο P2P H3: Ομοιότητα ως προς τις σχέσεις

6.3.2.1.2 P2P Δίκτυο πανεπιστημίων

Για την περίπτωση των πανεπιστημίων χρησιμοποιήσαμε και σε αυτή την κατηγορία πειραμάτων τα δίκτυα κόμβων U, U1, U2 και U3. Η ομοιότητα των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών είναι ίση με $48,2 \pm 1\%$, ενώ η ομοιότητα των κλειδιών των αρχικών σχημάτων τους είναι συνάρτηση της ομοιότητας των αρχικών σχέσεων (f_2), όπως και στο πείραμα 6.2.1.1, εφόσον χρησιμοποιούμε τα ίδια σχήματα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

U1

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	7/10	70,0	10/12	83,3
Schema 2	7/8	87,5	8/12	66,7
Schema 3	5/5	100,0	5/12	41,7
Schema 4	7/7	100,0	7/12	58,3
MO		89,4		62,5

Πίνακας 6.43 Δίκτυο P2P U1: Ομοιότητα ως προς τις σχέσεις

U2

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	6/9	66,7	9/15	60,0
Schema 2	5/7	71,4	7/15	46,7
Schema 3	7/7	57,1	7/15	46,7
Schema 4	6/7	85,7	7/15	46,7
MO		70,2		50,0

Πίνακας 6.44 Δίκτυο P2P U2: Ομοιότητα ως προς τις σχέσειςU

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	4/8	50,0	8/17	47,1
Schema 2	4/7	57,1	7/17	41,2
Schema 3	4/7	57,1	7/17	41,2
Schema 4	5/8	62,5	8/17	47,1
MO		56,7		44,1

Πίνακας 6.45 Δίκτυο P2P U: Ομοιότητα ως προς τις σχέσειςU3

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις σχέσεις (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις σχέσεις (%)
Schema 1	3/7	42,9	7/19	36,8
Schema 2	3/7	42,9	7/19	36,8
Schema 3	2/7	28,6	7/19	36,8
Schema 4	3/7	42,9	7/19	36,8
MO		39,3		36,8

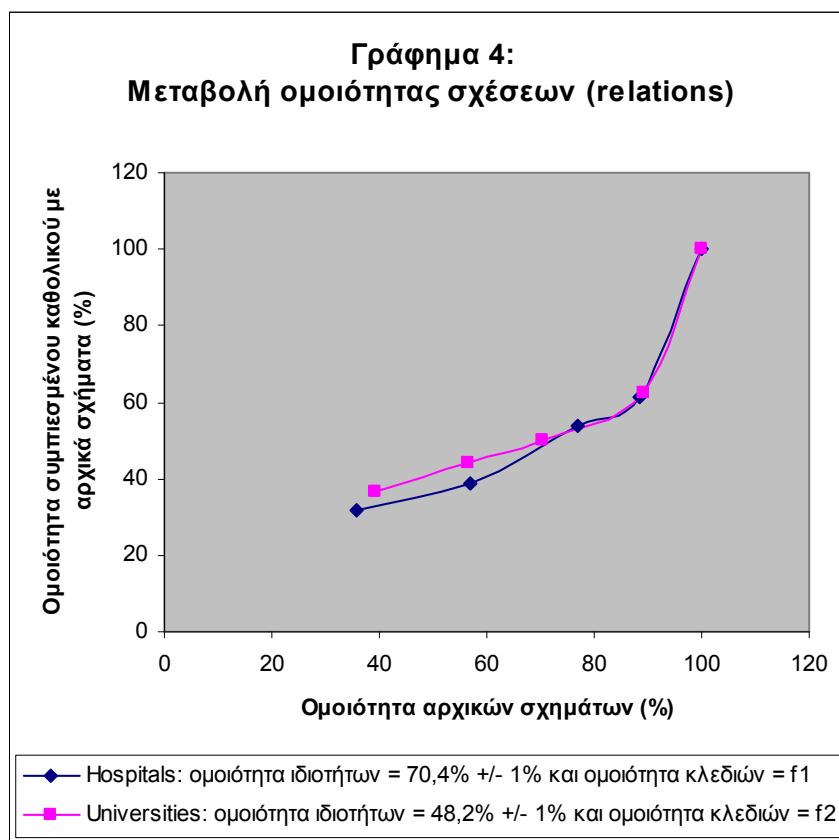
Πίνακας 6.46 Δίκτυο P2P U3: Ομοιότητα ως προς τις σχέσεις

6.3.2.1.3 Αποτελέσματα σύγκρισης ομοιοτήτων ως προς τις σχέσεις

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του συμπιεσμένου καθολικού με τα αρχικά σχήματα ως προς τις σχέσεις, καθώς μεταβάλλεται η ομοιότητα των αρχικών σχημάτων μεταξύ τους ως προς τις σχέσεις:

Ομοιότητα αρχικών σχημάτων νοσοκομείων (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα νοσοκομείων (%)	Ομοιότητα αρχικών σχημάτων πανεπιστημίων (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα πανεπιστημίων (%)
100,0	100,0	100,0	100,0
88,5	61,5	89,4	62,5
76,8	53,8	70,2	50,0
56,9	38,7	56,7	44,1
35,7	32,0	39,3	36,8

Πίνακας 6.47 Ομοιότητα σχέσεων αρχικών σχημάτων και συμπιεσμένου καθολικού για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων



Σχήμα 6.10 Μεταβολή ομοιότητας σχέσεων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

6.3.2.2 Μεταβολή ομοιότητας ιδιοτήτων (attributes) των αρχικών σχημάτων

6.3.2.2.1 P2P Δίκτυο νοσοκομείων

Για την περίπτωση των νοσοκομείων, χρησιμοποιήσαμε και σε αυτή την κατηγορία πειραμάτων τα δίκτυα κόμβων H, H4, H5 και H6. Οι ομοιότητες των σχέσεων και των κλειδιών των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 76,8% και 72,8% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

H4

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	13/15	86,7	15/29	51,7
Schema 2	15/17	88,2	17/29	58,6
Schema 3	15/16	93,8	16/29	55,2
Schema 4	13/14	92,9	14/29	48,3
Schema 5	11/15	73,3	15/29	51,7
MO		87,0		53,1

Πίνακας 6.48 Δίκτυο P2P H4: Ομοιότητα ως προς τις ιδιότητες

H

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	13/15	86,7	15/44	34,1
Schema 2	15/20	75,0	20/44	45,5
Schema 3	15/19	78,9	19/44	43,2
Schema 4	13/22	59,1	22/44	50,0
Schema 5	11/21	52,4	21/44	47,7
MO		70,4		44,1

Πίνακας 6.49 Δίκτυο P2P H: Ομοιότητα ως προς τις ιδιότητες

H5

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	11/20	55,0	20/69	29,0
Schema 2	15/23	65,2	23/69	33,3
Schema 3	14/22	63,6	22/69	31,9
Schema 4	12/24	50,0	24/69	34,8
Schema 5	10/27	37,0	27/69	39,1
MO		54,2		33,6

Πίνακας 6.50 Δίκτυο P2P H5: Ομοιότητα ως προς τις ιδιότητες

H6

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	10/25	40,0	24/107	22,4
Schema 2	13/30	43,3	29/107	27,1
Schema 3	13/30	43,3	30/107	28,0
Schema 4	12/32	37,5	30/107	28,0
Schema 5	9/33	27,3	33/107	30,8
MO		38,3		27,3

Πίνακας 6.51 Δίκτυο P2P H6: Ομοιότητα ως προς τις ιδιότητες

6.3.2.2 P2P Δίκτυο πανεπιστημίων

Για την περίπτωση των πανεπιστημίων, χρησιμοποιήσαμε και σε αυτή την κατηγορία πειραμάτων τα δίκτυα κόμβων U, U4, U5 και U6. Οι ομοιότητες των σχέσεων και των κλειδιών των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 56,7% και 70% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

U4

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	16/21	76,2	21/35	60,0
Schema 2	16/21	76,2	22/35	62,9
Schema 3	16/19	84,2	19/35	54,3
Schema 4	16/19	84,2	19/35	54,3
MO		80,2		57,9

Πίνακας 6.52 Δίκτυο P2P U4: Ομοιότητα ως προς τις ιδιότητες

U5

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	15/22	68,2	22/38	57,9
Schema 2	12/20	60,0	20/38	52,6
Schema 3	6/12	50,0	12/38	31,6
Schema 4	16/20	80,0	20/38	52,6
MO		64,5		48,7

Πίνακας 6.53 Δίκτυο P2P U5: Ομοιότητα ως προς τις ιδιότητες

U

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	15/26	57,7	26/70	37,1
Schema 2	12/33	36,4	33/70	47,1
Schema 3	6/23	26,1	23/70	32,9
Schema 4	16/22	72,7	22/70	31,4
MO		48,2		37,1

Πίνακας 6.54 Δίκτυο P2P U: Ομοιότητα ως προς τις ιδιότητες

U6

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τις ιδιότητες (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τις ιδιότητες (%)
Schema 1	8/25	32,0	25/70	35,7
Schema 2	5/26	19,2	26/70	37,1
Schema 3	5/22	22,7	22/70	31,4
Schema 4	10/18	55,6	18/70	25,7
MO		32,4		32,5

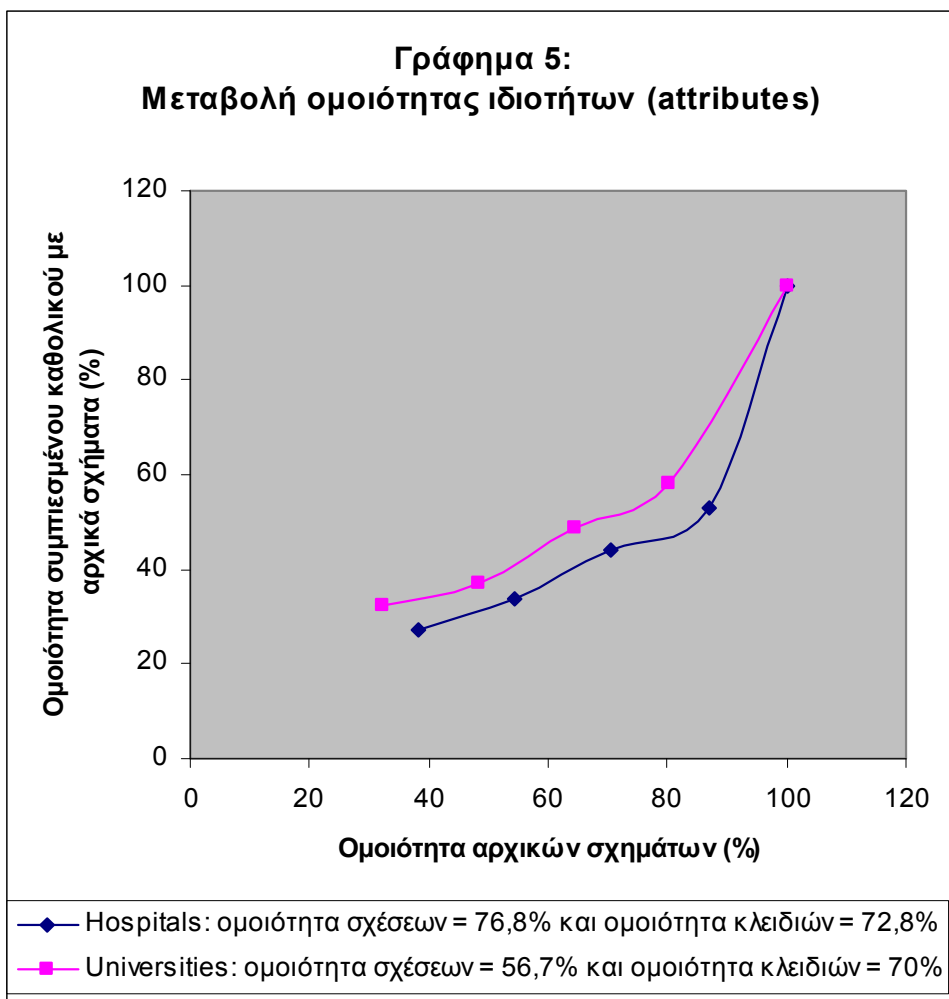
Πίνακας 6.55 Δίκτυο P2P U6: Ομοιότητα ως προς τις ιδιότητες

6.3.2.2.3 Αποτελέσματα σύγκρισης ομοιοτήτων ως προς τις ιδιότητες

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του συμπιεσμένου καθολικού με τα αρχικά σχήματα ως προς τις ιδιότητες, καθώς μεταβάλλεται η ομοιότητα των αρχικών σχημάτων μεταξύ τους ως προς τις ιδιότητες:

Ομοιότητα αρχικών σχημάτων νοσοκομείων (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα νοσοκομείων (%)	Ομοιότητα αρχικών σχημάτων πανεπιστημίων (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα πανεπιστημίων (%)
100,0	100,0	100,0	100,0
87,0	53,1	80,2	57,9
70,4	44,1	64,5	48,7
54,2	33,6	48,2	37,1
38,3	27,3	32,4	32,5

Πίνακας 6.56 Ομοιότητα ιδιοτήτων αρχικών σχημάτων και συμπιεσμένου καθολικού για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων



Σχήμα 6.11 Μεταβολή ομοιότητας ιδιοτήτων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

6.3.2.3 Μεταβολή ομοιότητας κλειδιών (keys) των αρχικών σχημάτων

6.3.2.3.1 P2P Δίκτυο νοσοκομείων

Για την περίπτωση των νοσοκομείων, χρησιμοποιήσαμε και σε αυτή την κατηγορία πειραμάτων τα δίκτυα κόμβων H, H7, H8 και H9. Οι ομοιότητες των σχέσεων και των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 76,8% και 70,4% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

H7

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	4/9	44,4
Schema 2	6/7	85,7	4/9	44,4
Schema 3	6/6	100,0	3/9	33,3
Schema 4	7/7	100,0	4/9	44,4
Schema 5	9/9	100,0	3/9	33,3
MO		93,1		40,0

Πίνακας 6.57 Δίκτυο P2P H7: Ομοιότητα ως προς τα κλειδιάH

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	3/12	25,0
Schema 2	3/7	42,9	4/12	33,3
Schema 3	6/6	100,0	2/12	16,7
Schema 4	6/7	85,7	3/12	25,0
Schema 5	5/9	55,6	6/12	50,0
MO		72,8		30,0

Πίνακας 6.58 Δίκτυο P2P H: Ομοιότητα ως προς τα κλειδιάH8

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	3/12	25,0
Schema 2	3/7	42,9	3/12	25,0
Schema 3	3/6	50,0	2/12	26,7
Schema 4	5/7	71,4	3/12	25,0
Schema 5	5/9	55,6	6/12	50,0
MO		60,0		28,3

Πίνακας 6.59 Δίκτυο P2P H8: Ομοιότητα ως προς τα κλειδιά

H9

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/5	80,0	1/12	8,3
Schema 2	3/7	42,9	1/12	8,3
Schema 3	0/6	0,0	0/12	0,0
Schema 4	5/7	71,4	2/12	16,7
Schema 5	5/9	55,6	4/12	33,3
MO		50,0		13,3

Πίνακας 6.60 Δίκτυο P2P H9: Ομοιότητα ως προς τα κλειδιά*6.3.2.3.2 P2P Δίκτυο πανεπιστημίων*

Για την περίπτωση των πανεπιστημίων, χρησιμοποιήσαμε και σε αυτή την κατηγορία πειραμάτων τα δίκτυα κόμβων U, τα U7, U8 U9 και U10. Οι ομοιότητες των σχέσεων και των ιδιοτήτων των αρχικών σχημάτων των κόμβων αυτών διατηρήθηκαν σταθερές και ίσες με 56,7% και 48,2% αντίστοιχα. Στο σημείο αυτό, παραθέτουμε τα ποσοστά ομοιότητας που προέκυψαν για κάθε ένα από τα δίκτυα κόμβων.

U7

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	5/7	71,4	6/11	54,5
Schema 2	6/7	85,7	4/11	36,4
Schema 3	5/5	100,0	4/11	36,4
Schema 4	6/7	85,7	5/11	45,5
MO		85,7		43,2

Πίνακας 6.61 Δίκτυο P2P U7: Ομοιότητα ως προς τα κλειδιά

U

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	5/7	71,4	4/13	30,8
Schema 2	4/7	57,1	4/13	30,8
Schema 3	4/5	80,0	3/13	23,1
Schema 4	5/7	71,4	4/13	30,8
MO		70,0		28,8

Πίνακας 6.62 Δίκτυο P2P U: Ομοιότητα ως προς τα κλειδιάU8

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	5/7	71,4	4/14	28,6
Schema 2	3/7	42,9	4/14	28,6
Schema 3	3/5	60,0	4/14	28,6
Schema 4	5/7	71,4	4/14	28,6
MO		61,4		28,6

Πίνακας 6.63 Δίκτυο P2P U8: Ομοιότητα ως προς τα κλειδιάU9

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/7	57,1	4/15	26,7
Schema 2	0/7	0,0	4/15	26,7
Schema 3	3/5	60,0	4/15	26,7
Schema 4	5/7	71,4	4/15	26,7
MO		47,1		26,7

Πίνακας 6.64 Δίκτυο P2P U9: Ομοιότητα ως προς τα κλειδιά

U10

	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών σχημάτων ως προς τα κλειδιά (%)	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
Schema 1	4/7	57,1	4/16	25,0
Schema 2	0/7	0,0	4/16	25,0
Schema 3	0/5	0,0	3/16	18,8
Schema 4	4/7	57,1	3/16	18,8
MO		28,6		21,9

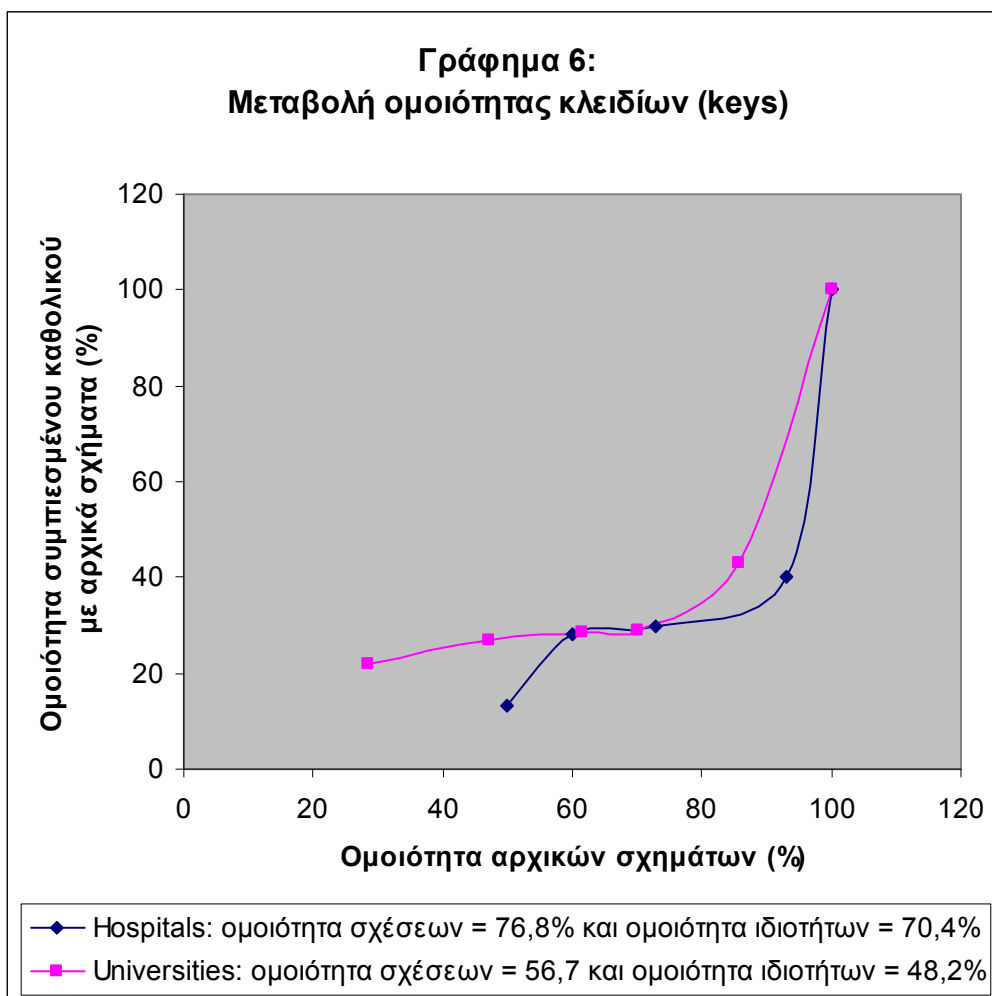
Πίνακας 6.65 Δίκτυο P2P U10: Ομοιότητα ως προς τα κλειδιά

6.3.2.3.3 Αποτελέσματα σύγκρισης ομοιοτήτων ως προς τα κλειδιά

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του συμπιεσμένου καθολικού με τα αρχικά σχήματα ως προς τα κλειδιά, καθώς μεταβάλλεται η ομοιότητα των αρχικών σχημάτων μεταξύ τους ως προς τα κλειδιά:

Ομοιότητα αρχικών σχημάτων νοσοκομείων (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα νοσοκομείων (%)	Ομοιότητα αρχικών σχημάτων πανεπιστημίων (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα πανεπιστημίων (%)
100,0	100,0	100,0	100,0
93,1	40,0	85,7	43,2
72,8	30,0	70,0	28,8
60,0	28,3	61,4	28,6
50,0	13,3	47,1	26,7
		28,6	21,9

Πίνακας 6.66 Ομοιότητα κλειδιών αρχικών σχημάτων και συμπιεσμένου καθολικού για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων



Σχήμα 6.12 Μεταβολή ομοιότητας κλειδίων για τα P2P δίκτυα των νοσοκομείων και των πανεπιστημίων

6.3.3 Σύγκριση ασυμπέστου καθολικού με συμπίεμένο καθολικό

Στο σημείο αυτό θα συγκρίνουμε τα αποτελέσματα που προέκυψαν από τα πειράματα 6.3.1 και 6.3.2, δηλαδή την ομοιότητα του (ασυμπέστου) καθολικού με τα αρχικά σχήματα με την ομοιότητα του συμπίεσμένου (καθολικού) με τα αρχικά. Αυτό θα γίνει για καθεμία από τις παραμέτρους ομοιότητας. Ουσιαστικά θα συγκρίνουμε τα γραφήματα: 1 (σχήμα 6.3) με 4 (σχήμα 6.10), 2 (σχήμα 6.4) με 5 (σχήμα 6.11) και 3 (σχήμα 6.5) με 6 (σχήμα 6.12). Η διαδικασία αυτή θα γίνει μόνο για την περίπτωση των νοσοκομείων, που είναι και πιο περίπλοκη.

Για τη σύγκριση των παραπάνω ομοιοτήτων, θα θεωρήσουμε την διαφορά της ομοιότητας του συμπίεσμένου με την ομοιότητα του καθολικού. Αυτή η διαφορά δείχνει κατά πόσο βελτιώθηκε το ποσοστό ομοιότητας του τελικού σχήματος μετά τη συμπίεση, γι' αυτό θα την ονομάσουμε **κέρδος**. Το κέρδος αυτό οφείλεται στο γεγονός ότι από το συμπίεμένο σχήμα

έχουν αφαιρεθεί σχέσεις, οι οποίες (σύμφωνα με τα κριτήρια συμπίεσης) δεν υπήρχαν σε πολλά από τα αρχικά σχήματα. Επομένως, το συμπιεσμένο σχήμα έχει λιγότερες σχέσεις, οι οποίες εμφανίζονται περισσότερες φορές στα αρχικά σχήματα. Για το λόγο αυτό, ενισχύεται η ομοιότητα του τελικού σχήματος με τα αρχικά.

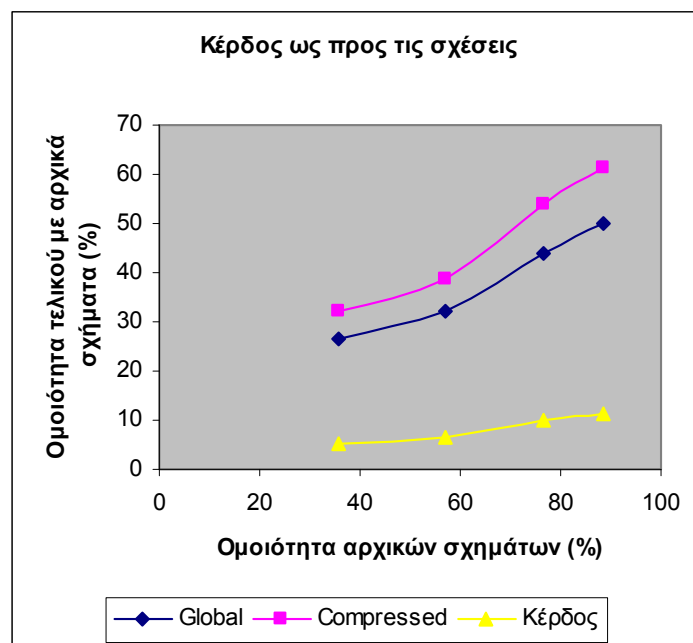
Όμως, με την συμπίεση του καθολικού σχήματος έχουμε και **απώλεια** πληροφορίας λόγω της αφαίρεσης σχέσεων. Αυτή την απώλεια, την υπολογίζουμε με τον εξής τρόπο:

1. Αρχικά, υπολογίζουμε για κάθε σχήμα το ποσοστό απώλειάς του, δηλαδή το λόγο των σχέσεων του σχήματος που αφαιρέθηκαν κατά τη συμπίεση, προς τον συνολικό αριθμό των σχέσεων που αφαιρέθηκαν από το καθολικό σχήμα.
2. Στη συνέχεια, υπολογίζουμε τον μέσο όρο των παραπάνω ποσοστών, ο οποίος αποτελεί τη ζητούμενη απώλεια.

6.3.3.1 Ομοιότητα ως προς τις σχέσεις (relations)

Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου καθολικού με αρχικά σχήματα (%)	Κέρδος
88,5	50,0	61,5	11,5
76,8	43,8	53,8	10,0
56,9	32,2	38,7	6,5
35,7	26,7	32,0	5,3

Πίνακας 6.67 Κέρδος συμπιεσμένου καθολικού ως προς τις σχέσεις

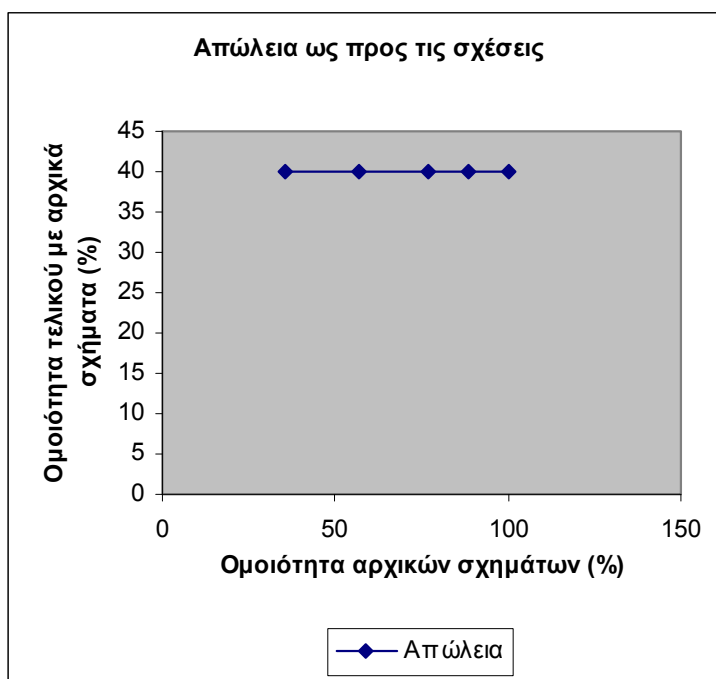


Σχήμα 6.13 Κέρδος συμπιεσμένου καθολικού ως προς τις σχέσεις

Η απώλεια για τις σχέσεις είναι η ίδια για όλα τα P2P δίκτυα που εξετάσαμε. Αυτό οφείλεται στην κατασκευή τους. Συγκεκριμένα, τα δίκτυα αυτά έχουν κατασκευαστεί έτσι ώστε οι σχέσεις που αφαιρούνται κατά την συμπίεση να υπάρχουν σε όλα στο ίδιο πλήθος. Οπότε το αποτέλεσμα αυτό δεν είναι ενδεικτικό.

Ποσοστό ομοιότητας αρχικών με συμπίεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπίεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
2/3	66,7
1/3	33,3
1/3	33,3
1/3	33,3
1/3	33,3
MO	40,0

Πίνακας 6.68 Απώλεια συμπίεσμένου καθολικού ως προς τις σχέσεις



Σχήμα 6.14 Απώλεια συμπίεσμένου καθολικού ως προς τις σχέσεις

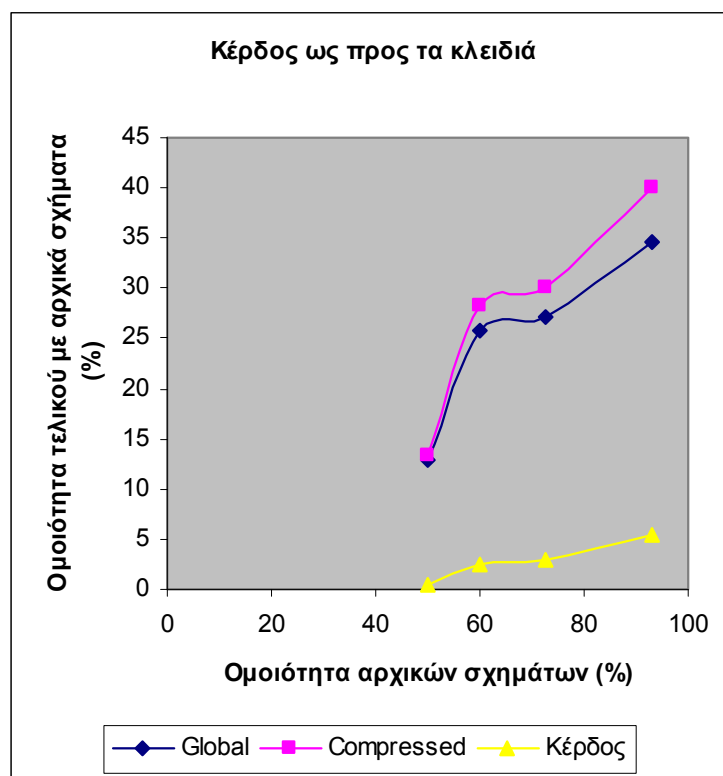
6.3.3.2 Ομοιότητα ως προς τις ιδιότητες (attributes)

Συγκρίνοντας τα γραφήματα 2 (σχήμα 6.4) και 5 (σχήμα 6.11), παρατηρούμε ότι η ομοιότητα των ιδιοτήτων του τελικού με τα αρχικά σχήματα δεν αλλάζει από το καθολικό στο συμπιεσμένο. Αυτό συμβαίνει, επειδή αφαιρώντας μία σχέση κατά τη συμπίεση, οι ιδιότητες της δεν χάνονται, αλλά εισάγονται κατάλληλα σε άλλες σχέσεις, ώστε να μην έχουμε απώλεια της σχετικής πληροφορίας.

6.3.3.3 Ομοιότητα ως προς τα κλειδιά (key)

Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου με αρχικά σχήματα (%)	Κέρδος
93,1	34,5	40,0	5,5
72,8	27,1	30,0	2,9
60,0	25,7	28,3	2,6
50,0	12,9	13,3	0,4

Πίνακας 6.69 Κέρδος συμπιεσμένου καθολικού ως προς τα κλειδιά

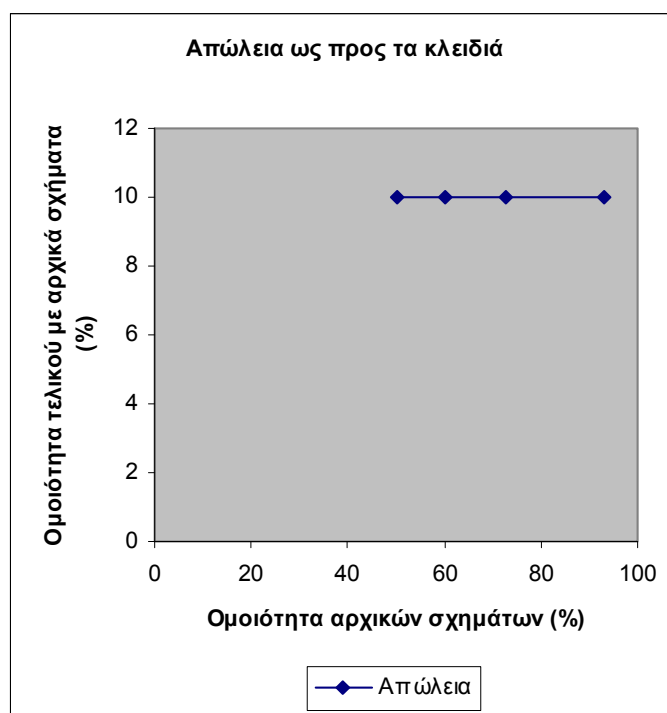


Σχήμα 6.15 Κέρδος συμπιεσμένου καθολικού ως προς τα κλειδιά

Η απώλεια για τα κλειδιά είναι η ίδια για όλα τα P2P δίκτυα που εξετάσαμε, λόγω της κατασκευής τους, όπως εξηγήθηκε παραπάνω.

Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά	Ποσοστό ομοιότητας αρχικών με συμπιεσμένο καθολικό σχήμα ως προς τα κλειδιά (%)
0/2	0,0
1/2	50,0
0/2	0,0
0/2	0,0
0/2	0,0
ΜΟ	10,0

Πίνακας 6.70 Απώλεια συμπιεσμένου καθολικού ως προς τα κλειδιά



Σχήμα 6.16 Απώλεια συμπιεσμένου καθολικού ως προς τα κλειδιά

6.3.4 Σύγκριση ομοιοτήτων με βάση το βαθμό συμπίεσης

6.3.4.1 Ομοιότητα αρχικών σχημάτων με το τελικό (καθολικό ή συμπιεσμένο) για διάφορους βαθμούς συμπίεσης

Στο πρώτο μέρος της κατηγορίας πειραμάτων, συγκρίναμε την ομοιότητα των αρχικών σχημάτων με το καθολικό και τα συμπιεσμένα αυξανόμενων βαθμών συμπίεσης. Εκτελέσαμε τρεις σειρές μετρήσεων, σε κάθε μία από τις οποίες ο υπολογισμός της ομοιότητας έγινε ως προς μία από τις τρεις παραμέτρους ομοιότητας.

Στους πίνακες που ακολουθούν, η πρώτη στήλη αποτελεί τα ποσοστά εισόδου των πειραμάτων και οι υπόλοιπες τα ποσοστά εξόδου.

6.3.4.1.1 Ομοιότητα ως προς τις σχέσεις (relations)

H1

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	4/5	5/16	5/15	5/14	5/13
Schema 2	5/8	8/16	8/15	8/14	8/13
Schema 3	6/6	6/16	6/15	6/14	6/13
Schema 4	11/11	11/16	11/15	11/14	11/13
Schema 5	10/10	10/16	10/15	10/14	10/13

Πίνακας 6.71 Δίκτυο P2P H1: Ομοιότητα ως προς τις σχέσεις

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	80,0	31,3	33,3	35,7	38,5
Schema 2	62,5	50,0	53,3	57,1	61,5
Schema 3	100,0	37,5	40,0	42,9	46,2
Schema 4	100,0	68,8	73,3	78,6	84,6
Schema 5	100,0	62,5	66,7	71,4	76,9
MO	88,5	50,0	53,3	57,1	61,5

Πίνακας 6.72 Δίκτυο P2P H1: Ομοιότητα (%) ως προς τις σχέσεις

H

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	4/5	5/16	5/15	5/14	5/13
Schema 2	5/8	8/16	8/15	8/14	8/13
Schema 3	6/6	6/16	6/15	6/14	6/13
Schema 4	6/7	7/16	7/15	7/14	7/13
Schema 5	5/9	9/16	9/15	9/14	9/13

Πίνακας 6.73 Δίκτυο P2P H: Ομοιότητα ως προς τις σχέσεις

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	80,0	31,3	33,3	35,7	38,5
Schema 2	62,5	50,0	53,3	57,1	61,5
Schema 3	100,0	37,5	40,0	42,9	46,2
Schema 4	85,7	43,8	46,7	50,0	53,8
Schema 5	55,6	56,3	60,0	64,3	69,2
MO	76,8	43,8	46,7	50,0	53,8

Πίνακας 6.74 Δίκτυο P2P H: Ομοιότητα (%) ως προς τις σχέσειςH2

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	3/4	4/18	4/17	4/16	4/15
Schema 2	3/6	6/18	6/17	6/16	6/15
Schema 3	4/6	6/18	6/17	6/16	6/15
Schema 4	3/6	6/18	6/17	6/16	6/15
Schema 5	3/7	7/18	7/17	7/16	7/15

Πίνακας 6.75 Δίκτυο P2P H2: Ομοιότητα ως προς τις σχέσεις

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	75,0	22,2	23,5	25,0	26,7
Schema 2	50,0	33,3	35,3	37,5	40,0
Schema 3	66,7	33,3	35,3	37,5	40,0
Schema 4	50,0	33,3	35,3	37,5	40,0
Schema 5	42,9	38,9	41,2	43,8	46,7
MO	56,9	32,2	34,1	36,3	38,7

Πίνακας 6.76 Δίκτυο P2P H2: Ομοιότητα (%) ως προς τις σχέσεις

H3

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	1/2	2/18	2/17	2/16	2/15
Schema 2	1/5	5/18	5/17	5/16	5/15
Schema 3	2/5	5/18	5/17	5/16	5/15
Schema 4	2/5	5/18	5/17	5/16	5/15
Schema 5	2/7	7/18	7/17	7/16	7/15

Πίνακας 6.77 Δίκτυο P2P H3 Ομοιότητα ως προς τις σχέσεις

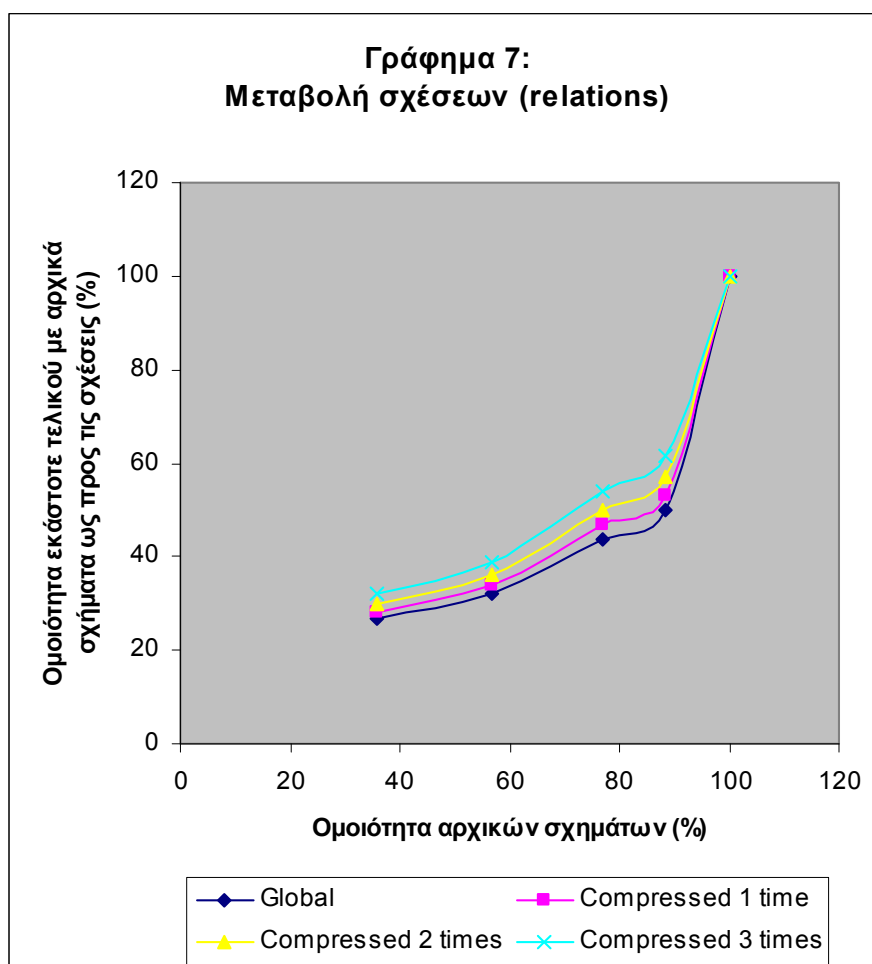
	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	50,0	11,1	11,8	12,5	13,3
Schema 2	20,0	27,8	29,4	31,3	33,3
Schema 3	40,0	27,8	29,4	31,3	33,3
Schema 4	40,0	27,8	29,4	31,3	33,3
Schema 5	28,6	38,9	41,2	43,8	46,7
MO	35,7	26,7	28,2	30,0	32,0

Πίνακας 6.78 Δίκτυο P2P H3 Ομοιότητα (%) ως προς τις σχέσεις

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του εκάστοτε τελικού (καθολικού ή συμπιεσμένων) σχήματος με τα αρχικά σχήματα ως προς τις σχέσεις:

Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
100,0	100,0	100,0	100,0	100,0
88,5	50,0	53,3	57,1	61,5
76,8	43,8	46,7	50,0	53,8
56,9	32,2	34,1	36,3	38,7
35,7	26,7	28,2	30,0	32,0

Πίνακας 6.79 Ομοιότητα σχέσεων αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων



Σχήμα 6.17 Ομοιότητα σχέσεων αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων

6.3.4.1.2 Ομοιότητα ως προς τις ιδιότητες (attributes)

H4

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	13/15	15/29	15/29	15/29	15/29
Schema 2	15/17	17/29	17/29	17/29	17/29
Schema 3	15/16	16/29	16/29	16/29	16/29
Schema 4	13/14	14/29	14/29	14/29	14/29
Schema 5	11/15	15/29	15/29	15/29	15/29

Πίνακας 6.80 Δίκτυο P2P H4 Ομοιότητα ως προς τις ιδιότητες

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	86,7	51,7	51,7	51,7	51,7
Schema 2	88,2	58,6	58,6	58,6	58,6
Schema 3	93,8	55,2	55,2	55,2	55,2
Schema 4	92,9	48,3	48,3	48,3	48,3
Schema 5	73,3	51,7	51,7	51,7	51,7
MO	87,0	53,1	53,1	53,1	53,1

Πίνακας 6.81 Δίκτυο P2P H4 Ομοιότητα (%) ως προς τις ιδιότητες

H

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	13/15	15/44	15/44	15/44	15/44
Schema 2	15/20	20/44	20/44	20/44	20/44
Schema 3	15/19	19/44	19/44	19/44	19/44
Schema 4	13/22	22/44	22/44	22/44	22/44
Schema 5	11/21	21/44	21/44	21/44	21/44

Πίνακας 6.82 Δίκτυο P2P H Ομοιότητα ως προς τις ιδιότητες

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	86,7	34,1	34,1	34,1	34,1
Schema 2	75,0	45,5	45,5	45,5	45,5
Schema 3	78,9	43,2	43,2	43,2	43,2
Schema 4	59,1	50,0	50,0	50,0	50,0
Schema 5	52,4	47,7	47,7	47,7	47,7
MO	70,4	44,1	44,1	44,1	44,1

Πίνακας 6.83 Δίκτυο P2P Η Ομοιότητα (%) ως προς τις ιδιότητες

H5

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	11/20	20/69	20/69	20/69	20/69
Schema 2	15/23	23/69	23/69	23/69	23/69
Schema 3	14/22	22/69	22/69	22/69	22/69
Schema 4	12/24	24/69	24/69	24/69	24/69
Schema 5	10/27	27/69	27/69	27/69	27/69

Πίνακας 6.84 Δίκτυο P2P H5 Ομοιότητα ως προς τις ιδιότητες

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	55,0	29,0	29,0	29,0	29,0
Schema 2	65,2	33,3	33,3	33,3	33,3
Schema 3	63,6	31,9	31,9	31,9	31,9
Schema 4	50,0	34,8	34,8	34,8	34,8
Schema 5	37,0	39,1	39,1	39,1	39,1
MO	54,2	33,6	33,6	33,6	33,6

Πίνακας 6.85 Δίκτυο P2P H5 Ομοιότητα (%) ως προς τις ιδιότητες

H6

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	10/25	24/107	24/107	24/107	24/107
Schema 2	13/30	29/107	29/107	29/107	29/107
Schema 3	13/30	30/107	30/107	30/107	30/107
Schema 4	12/32	30/107	30/107	30/107	30/107
Schema 5	9/33	33/107	33/107	33/107	33/107

Πίνακας 6.86 Δίκτυο P2P H6 Ομοιότητα ως προς τις ιδιότητες

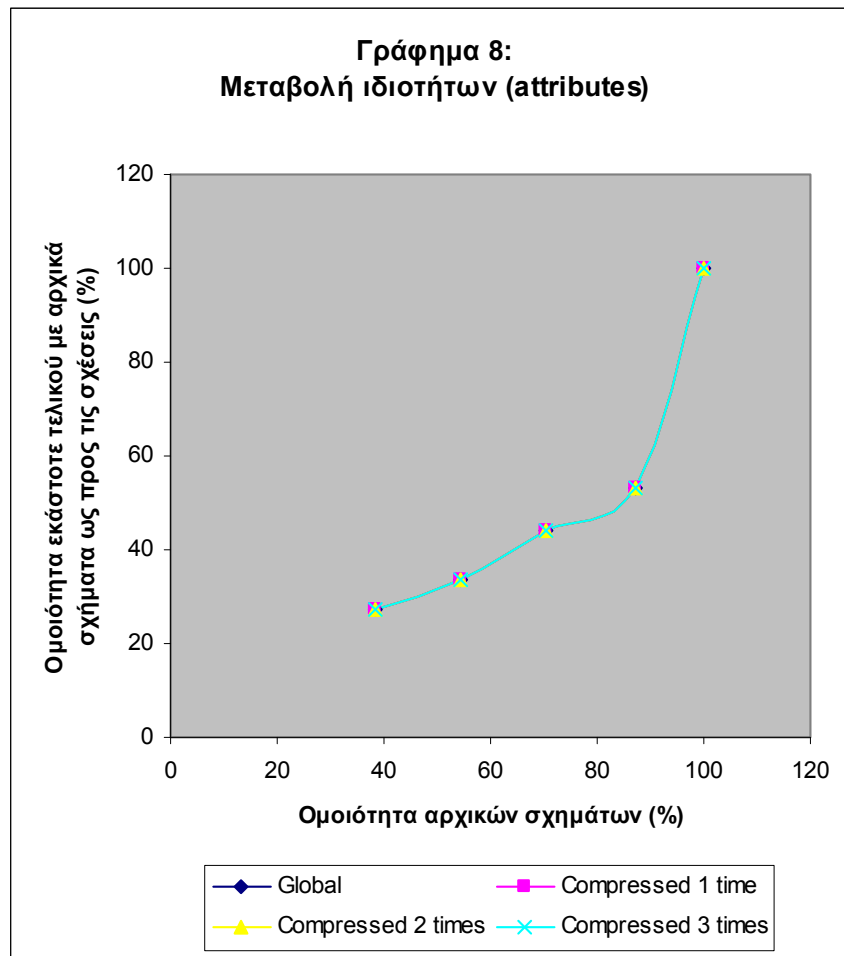
	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	40,0	22,4	22,4	22,4	22,4
Schema 2	43,3	27,1	27,1	27,1	27,1
Schema 3	43,3	28,0	28,0	28,0	28,0
Schema 4	37,5	28,0	28,0	28,0	28,0
Schema 5	27,3	30,8	30,8	30,8	30,8
MO	38,3	27,3	27,3	27,3	27,3

Πίνακας 6.87 Δίκτυο P2P H6 Ομοιότητα (%) ως προς τις ιδιότητες

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του εκάστοτε τελικού (καθολικού ή συμπιεσμένων) σχήματος με τα αρχικά σχήματα ως προς τις ιδιότητες:

Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
100,0	100,0	100,0	100,0	100,0
87,0	53,1	53,1	53,1	53,1
70,4	44,1	44,1	44,1	44,1
54,2	33,6	33,6	33,6	33,6
38,3	27,3	27,3	27,3	27,3

Πίνακας 6.88 Ομοιότητα ιδιοτήτων αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων



Σχήμα 6.18 Ομοιότητα ιδιοτήτων αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων

6.3.4.1.3 Ομοιότητα ως προς τα κλειδιά (keys)

H7

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	4/5	4/11	4/11	4/10	4/9
Schema 2	6/7	5/11	5/11	4/10	4/9
Schema 3	6/6	3/11	3/11	3/10	3/9
Schema 4	7/7	4/11	4/11	4/10	4/9
Schema 5	9/9	3/11	3/11	3/10	3/9

Πίνακας 6.89 Δίκτυο P2P H7 Ομοιότητα ως προς τις ιδιότητες

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	80,0	36,4	36,4	40,0	44,4
Schema 2	85,7	45,5	45,5	40,0	44,4
Schema 3	100,0	27,3	27,3	30,0	33,3
Schema 4	100,0	36,4	36,4	40,0	44,4
Schema 5	100,0	27,3	27,3	30,0	33,3
MO	93,1	34,5	34,5	36,0	40,0

Πίνακας 6.90 Δίκτυο P2P H7 Ομοιότητα (%) ως προς τις ιδιότητες

H

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	4/5	3/14	3/14	3/13	3/12
Schema 2	3/7	5/14	5/14	4/13	4/12
Schema 3	6/6	2/14	2/14	2/13	2/12
Schema 4	6/7	3/14	3/14	3/13	3/12
Schema 5	5/9	6/14	6/14	6/13	6/12

Πίνακας 6.91 Δίκτυο P2P H Ομοιότητα ως προς τις ιδιότητες

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	80,0	21,4	21,4	23,1	25,0
Schema 2	42,9	35,7	35,7	30,8	33,3
Schema 3	100,0	14,3	14,3	15,4	16,7
Schema 4	85,7	21,4	21,4	23,1	25,0
Schema 5	55,6	42,9	42,9	46,2	50,0
MO	72,8	27,1	27,1	27,7	30,0

Πίνακας 6.92 Δίκτυο P2P H Ομοιότητα (%) ως προς τις ιδιότητες

H8

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	4/5	3/14	3/14	3/13	3/12
Schema 2	3/7	4/14	4/14	3/13	3/12
Schema 3	6/6	2/14	2/14	2/13	2/12
Schema 4	6/7	3/14	3/14	3/13	3/12
Schema 5	5/9	6/14	6/14	6/13	6/12

Πίνακας 6.93 Δίκτυο P2P H8 Ομοιότητα ως προς τις ιδιότητες

	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	80,0	21,4	21,4	23,1	25,0
Schema 2	42,9	28,6	28,6	23,1	25,0
Schema 3	50,0	14,3	14,3	15,4	16,7
Schema 4	71,4	21,4	21,4	23,1	25,0
Schema 5	55,6	42,9	42,9	46,2	50,0
MO	60,0	25,7	25,7	26,2	28,3

Πίνακας 6.94 Δίκτυο P2P H8 Ομοιότητα (%) ως προς τις ιδιότητεςH9

	Ομοιότητα αρχικών σχημάτων	Ομοιότητα καθολικού με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα
Schema 1	4/5	1/14	1/14	1/13	1/12
Schema 2	3/7	2/14	2/14	1/13	1/12
Schema 3	6/6	0/14	0/14	0/13	0/12
Schema 4	6/7	2/14	2/14	2/13	2/12
Schema 5	5/9	4/14	4/14	4/13	4/12

Πίνακας 6.95 Δίκτυο P2P H9 Ομοιότητα ως προς τις ιδιότητες

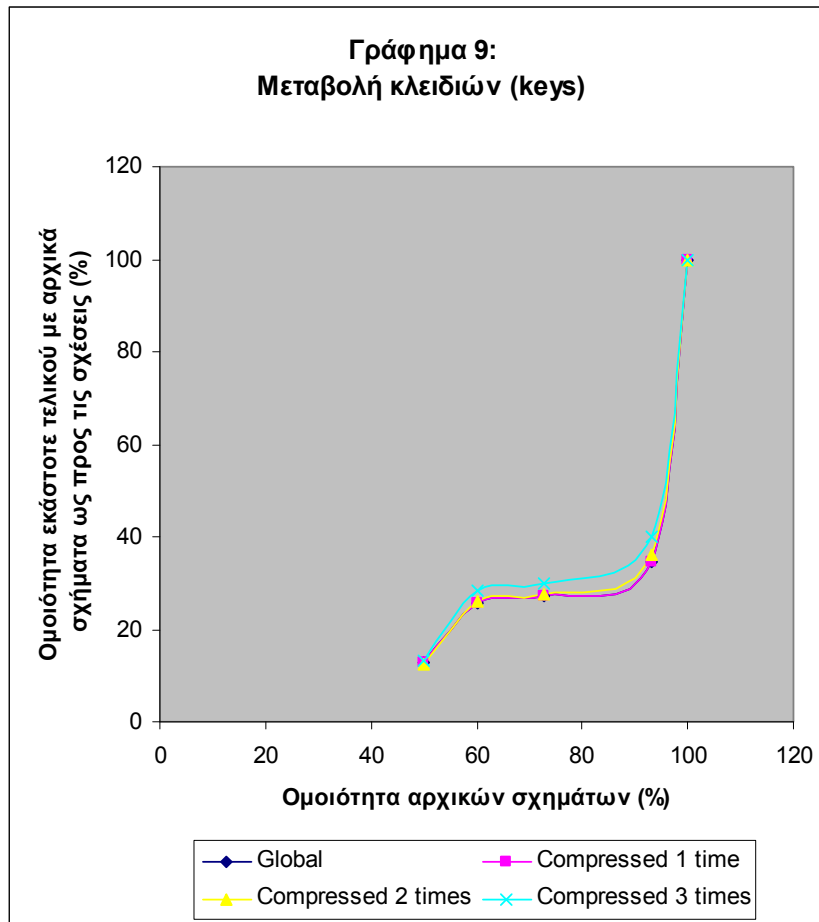
	Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
Schema 1	80,0	7,1	7,1	7,7	8,3
Schema 2	42,9	14,3	14,3	7,7	8,3
Schema 3	0,0	0,0	0,0	0,0	0,0
Schema 4	71,4	14,3	14,3	15,4	16,7
Schema 5	55,6	28,6	28,6	30,8	33,3
MO	50,0	12,9	12,9	12,3	13,3

Πίνακας 6.96 Δίκτυο P2P H9 Ομοιότητα (%) ως προς τις ιδιότητες

Από τα παραπάνω δεδομένα κατασκευάζουμε τον εξής πίνακα και γραφική παράσταση, που δείχνει πως μεταβάλλεται η ομοιότητα του εκάστοτε τελικού (καθολικού ή συμπιεσμένου) σχήματος με τα αρχικά σχήματα ως προς τα κλειδιά:

Ομοιότητα αρχικών σχημάτων (%)	Ομοιότητα καθολικού με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 1) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 2) με αρχικά σχήματα (%)	Ομοιότητα συμπιεσμένου (βαθμός συμπίεσης 3) με αρχικά σχήματα (%)
100,0	100,0	100,0	100,0	100,0
93,1	34,5	34,5	36,0	40,0
72,8	27,1	27,1	27,7	30,0
60,0	25,7	25,7	26,2	28,3
50,0	12,9	12,9	12,3	13,3

Πίνακας 6.97 Ομοιότητα κλειδιών αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων



Σχήμα 6.19 Ομοιότητα κλειδιών αρχικών και εκάστοτε τελικών για τα P2P δίκτυα των νοσοκομείων

6.3.4.2 Ομοιότητα συμπιεσμένου με καθολικό καθώς μεταβάλλεται ο βαθμός συμπίεσης

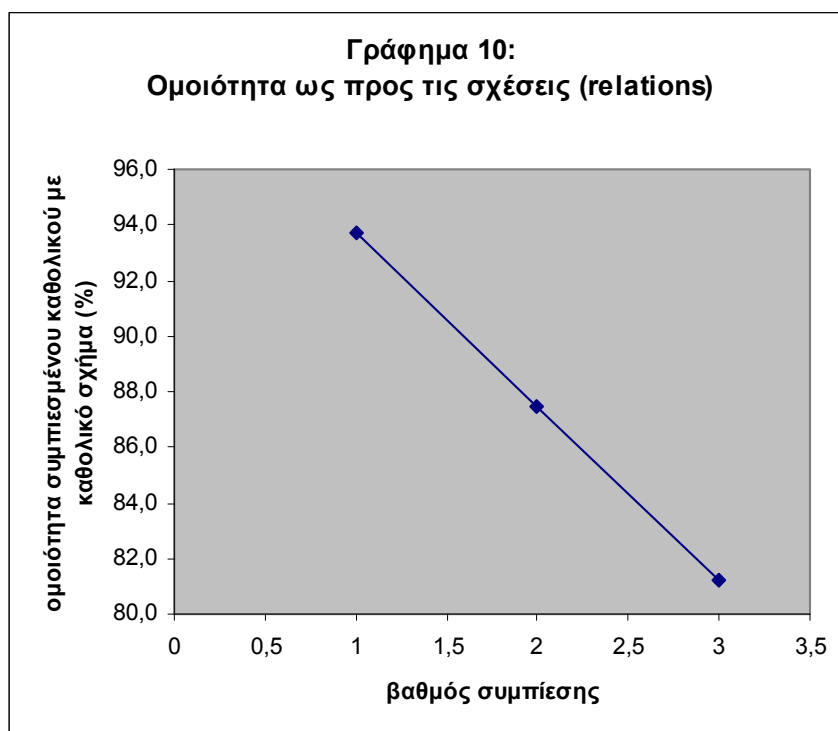
Στο πείραμα αυτό συγκρίναμε την ομοιότητα μεταξύ ενός συμπιεσμένου καθολικού σχήματος και του αντίστοιχου ασυμπιεστού καθολικού σχήματος από το οποίο προήλθε, καθώς μεταβάλλεται ο βαθμός συμπίεσης. Για την εκτέλεση χρησιμοποιήθηκε το πρότυπο δίκτυο κόμβων των νοσοκομείων Η. Εκτελέσαμε τρεις σειρές μετρήσεων, σε κάθε μία από τις οποίες ο υπολογισμός της ομοιότητας έγινε ως προς μία από τις τρεις παραμέτρους ομοιότητας.

Στη συνέχεια παρουσιάζονται τα ποσοστά ομοιότητας μεταξύ του καθολικού και του συμπιεσμένου σχήματος, που προέκυψαν μετά από κάθε συμπίεση, καθώς και οι γραφικές παραστάσεις για κάθε περίπτωση.

6.3.4.2.1 Ομοιότητα ως προς τις σχέσεις (relations)

Βαθμός συμπίεσης	Ομοιότητα του συμπεσμένου με καθολικό σχήμα	Ομοιότητα του συμπεσμένου με καθολικό σχήμα (%)
1	15/16	93,8
2	14/16	87,5
3	13/16	81,3

Πίνακας 6.98 Ομοιότητα συμπεσμένου με καθολικό ως προς τις σχέσεις για διάφορους βαθμούς συμπίεσης

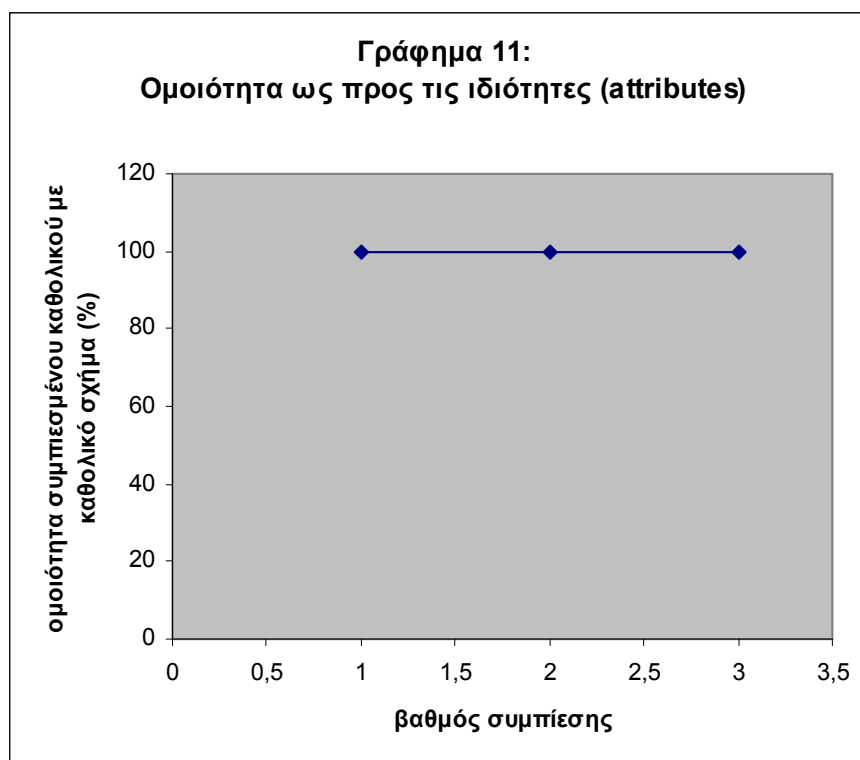


Σχήμα 6.20 Ομοιότητα σχέσεων συμπεσμένου με καθολικό ως προς τις σχέσεις συναρτήσει του βαθμού συμπίεσης

6.3.4.2.2 Ομοιότητα ως προς τις ιδιότητες (attributes)

Βαθμός συμπίεσης	Ομοιότητα του συμπιεσμένου με καθολικό σχήμα (%)
1	100,0
2	100,0
3	100,0

Πίνακας 6.99 Ομοιότητα συμπιεσμένου με καθολικό ως προς τις ιδιότητες για διάφορους βαθμούς συμπίεσης

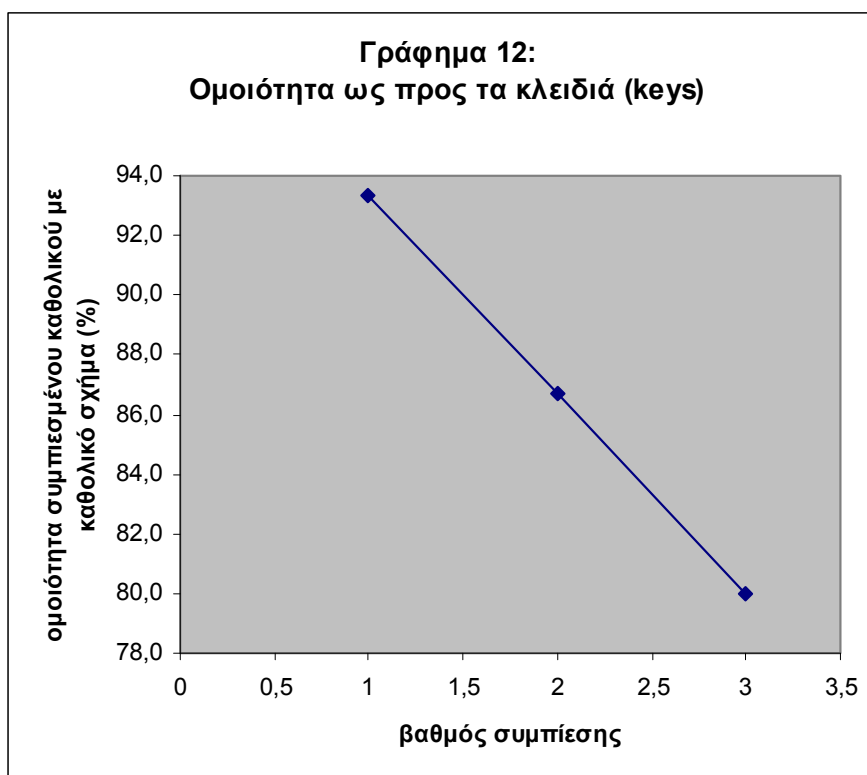


Σχήμα 6.21 Ομοιότητα συμπιεσμένου με καθολικό ως προς τις ιδιότητες συναρτήσει του βαθμού συμπίεσης

6.3.4.2.3 Ομοιότητα ως προς τα κλειδιά (keys)

Βαθμός συμπίεσης	Ομοιότητα του συμπιεσμένου με καθολικό σχήμα	Ομοιότητα του συμπιεσμένου με καθολικό σχήμα (%)
1	14/15	93,3
2	13/15	86,7
3	12/15	80,0

Πίνακας 6.100 Ομοιότητα συμπιεσμένου με καθολικό ως προς τα κλειδιά για διάφορους βαθμούς συμπίεσης



Σχήμα 6.22 Ομοιότητα συμπιεσμένου με καθολικό ως προς τα κλειδιά συναρτήσει του βαθμού συμπίεσης

6.4 Συμπεράσματα

6.4.1 Ομοιότητα καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων

Από τα γραφήματα 1, 2 και 3 (σχήματα 6.3, 6.4 και 6.5 αντίστοιχα), παρατηρούμε ότι οι γραφικές παραστάσεις μεταβολής ομοιότητας των ιδιοτήτων (γράφημα 2) τείνουν προς τη διχοτόμο της γωνίας των αξόνων του συστήματος συντεταγμένων. Αντίθετα, οι γραφικές παραστάσεις μεταβολής ομοιότητας των σχέσεων (γράφημα 1) αποκλίνουν περισσότερο από τη διχοτόμο και αυτές των κλειδιών (γράφημα 3) ακόμα πιο πολύ. Όταν μία γραφική παράσταση είναι κοντά στη διχοτόμο σημαίνει ότι μία μεταβολή των τιμών των τεταγμένων (άξονας x) επηρεάζει σε μεγάλο βαθμό τις τιμές των τεταγμένων (άξονας y). Έτσι, από τα γραφήματα συμπεραίνουμε ότι η ομοιότητα των ιδιοτήτων του καθολικού με τα αρχικά σχήματα επηρεάζεται περισσότερο από αυτή των σχέσεων, και η τελευταία περισσότερο από αυτή των κλειδιών, σε μία μεταβολή της ομοιότητας των αντίστοιχων παραμέτρων των αρχικών σχημάτων. Αυτό είναι λογικό, αν αναλογιστούμε ότι οι σχέσεις αποτελούν μια πιο ευρεία έννοια και έχουν μεγαλύτερη βαρύτητα και σημασία σε μία βάση δεδομένων από τις ιδιότητες. Έτσι, έχουμε μεγαλύτερες απαιτήσεις στις σχέσεις για να γίνει μία αλλαγή στο καθολικό σχήμα. Δηλαδή, απαιτείται οι σχέσεις να βρίσκονται σε όσο το δυνατόν περισσότερα σχήματα (μεγάλη ομοιότητα). Έτσι, η έλλειψη, για παραδειγμα, μίας σχέσης από ένα σχήμα επηρεάζει περισσότερο το καθολικό από την έλλειψη μίας ιδιότητας. Κατ' ανάλογο τρόπο επιδρούν και τα κλειδιά, τα οποία έχουν μεγαλύτερη βαρύτητα από τις σχέσεις, γιατί είναι αυτά που καθορίζουν το περιεχόμενό τους.

Μία επιπλέον παρατήρηση, που μπορεί να γίνει, είναι ότι η κλίση όλων των γραφικών παραστάσεων είναι αρχικά μικρή και αυξάνεται πολύ καθώς πλησιάζουμε την ομοιότητα 100%. Αυτό συμβαίνει επειδή όσο η ομοιότητα μεταξύ των αρχικών σχημάτων αυξάνεται, τόσο περισσότερο θα μοιάζει και το καθολικό σχήμα με αυτά. Υπενθυμίζουμε ότι η ομοιότητα του καθολικού σχήματος με τα αρχικά υπολογίστηκε ως ο μέσος όρος των ομοιοτήτων του καθολικού με καθένα από τα αρχικά. Οπότε, όσο λιγότερο μοιάζουν τα αρχικά σχήματα μεταξύ τους, τόσο μικρότερο θα είναι και το ποσοστό ομοιότητας του καθολικού με κάθε ένα από τα σχήματα αυτά χωριστά.

Επίσης, συγκρίνοντας σε κάθε γράφημα τις δύο καμπύλες για τις δύο περιπτώσεις (νοσοκομεία και πανεπιστήμια), όπου υπάρχει διαφορετικό ποσοστό ομοιότητας των αρχικών

σχημάτων, ως προς τις άλλες παραμέτρους απ' αυτές που εξετάζονται, παρατηρούμε ότι έχουν γενικά την ίδια μορφή. Η σχέση των δύο γραφικών παραστάσεων (ποια είναι πάνω από την άλλη) δεν εξαρτάται από τα σταθερά ποσοστά ομοιότητας των άλλων παραμέτρων, αφού αυτές είναι ανεξάρτητες από τις παραμέτρους ως προς τις οποίες γίνεται η σύγκριση των ομοιοτήτων.

Τέλος, αξίζει να σημειωθεί μία λεπτομέρεια για τα κλειδιά. Τα κλειδιά είναι γενικά ανεξάρτητα από τις σχέσεις, αφού ένα σύνολο από ιδιότητες που αποτελεί ένα κλειδί σε μία σχέση μπορεί να βρίσκεται ως κλειδί και σε κάποια άλλη σχέση. Έτσι, η διασπορά των κλειδιών σε ένα σχήμα ποικίλει, και γενικά εξαρτάται από την κατασκευή του σχήματος. Επίσης, κατά τη δημιουργία του καθολικού σχήματος, τα κλειδιά των σχέσεων μεταβάλλονται - για παράδειγμα, κατά τη συγχώνευση δύο σχέσεων, η τελική σχέση περιέχει ένα νέο κλειδί που ουσιαστικά αποτελεί ένωση των κλειδιών των αρχικών σχέσεων. Έτσι πολλά απ' αυτά δεν είναι πλέον όμοια με τα αρχικά.. Για το λόγο αυτό, τα αποτελέσματα είναι κατά μεγάλο ποσοστό τυχαία και μπορεί να μην ακολουθούν συγκεκριμένους κανόνες. Για παράδειγμα, όπως αναφέραμε και παραπάνω, παρατηρούμε ότι τα δύο πρώτα σημεία της γραφικής παράστασης των νοσοκομείων του γραφήματος 3 (σχήμα 6.5) έχουν μεγάλη διαφορά, δηλαδή η κλίση της γραφικής παράστασης είναι πολύ μεγάλη στο συγκεκριμένο σημείο. Αυτό οφείλεται στην κατασκευή των συγκεκριμένων συνόλων σχημάτων (P2P δικτύων). Ειδικότερα, τα κλειδιά, κατά τα οποία διαφέρουν τα σχήματα των P2P δικτύων H8 και H9, που αντιπροσωπεύουν τα δύο σημεία, υπάρχουν ως κλειδιά στο καθολικό σχήμα, κι έτσι επηρεάζουν πολύ τα ποσοστά ομοιότητας καθολικού με αρχικά. Αυτό δεν συμβαίνει στα αντίστοιχα σημεία της γραφικής παράστασης των πανεπιστημίων, όπου η ομοιότητα καθολικού με αρχικά δεν μεταβάλλεται ιδιαίτερα. Συγκεκριμένα, τα περισσότερα από τα κλειδιά, κατά τα οποία διαφέρουν τα σχήματα των δικτύων των δύο πρώτων σημείων της γραφικής αυτής παράστασης, δεν υπάρχουν στο καθολικό σχήμα, κι έτσι η προσθήκη ή αφαίρεση τους δεν αλλάζει σε μεγάλο ποσοστό την ομοιότητα καθολικού με αρχικά.

6.4.2 Ομοιότητα συμπίεσμένου καθολικού με αρχικά σχήματα με μεταβολή μίας παραμέτρου ομοιότητας των αρχικών σχημάτων

Συγκρίνοντας τα γραφήματα 1 (σχήμα 6.3) με 4 (σχήμα 6.10), 2 (σχήμα 6.4) με 5 (σχήμα 6.11) και 3 (σχήμα 6.5) με 6 (σχήμα 6.12), παρατηρούμε ότι έχουν την ίδια μορφή, δηλαδή η ομοιότητα του συμπίεσμένου σχήματος με τα αρχικά μεταβάλλεται ως προς την ομοιότητα των αρχικών, με τον ίδιο τρόπο που μεταβάλλεται η αντίστοιχη ομοιότητα του ασυμπίεστου καθολικού σχήματος. Συγκεκριμένα, παρατηρούμε ότι τα γραφήματα 2 και 5 είναι όμοια. Αυτό οφείλεται στο γεγονός ότι κατά την αφαίρεση σχέσεων λόγω της συμπίεσης, δεν

αφαιρούνται οι ιδιότητες των σχέσεων αυτών, αλλά προστίθενται κατάλληλα σε άλλες σχέσεις. Έτσι το πλήθος τους δεν μεταβάλλεται, και επομένως ούτε τα ποσοστά ομοιότητας τους.

Από τη γραφική παράσταση του σχήματος 6.13, παρατηρούμε ότι το κέρδος αυξάνεται καθώς αυξάνεται η ομοιότητα των αρχικών σχημάτων. Αυτό συμβαίνει επειδή, όπως αναφέραμε και στην παράγραφο 6.3.3, η αφαίρεση σχέσεων, κατά την συμπίεση, ενισχύει την ομοιότητα τελικού με αρχικά σχήματα, γιατί οι σχέσεις που αφαιρούνται δεν εμφανίζονται σε μεγάλο ποσοστό στα αρχικά σχήματα. Έτσι, οι σχέσεις που παραμένουν στο συμπιεσμένο σχήμα, και οι οποίες συμβάλλουν θετικά στην ομοιότητα του τελικού με τα αρχικά σχήματα, λόγω της συχνής εμφάνισής τους σ' αυτά, συμβάλλουν ακόμα θετικότερα στην ομοιότητα αυτή στο συμπιεσμένο απ' ότι στο ασυμπιεστο, στο οποίο υπάρχουν και οι σχέσεις που αφαιρέθηκαν. Για τον λόγο αυτό, όσο πιο όμοια είναι τα αρχικά σχήματα μεταξύ τους, τόσο περισσότερο θα ενισχύεται και η ομοιότητα του τελικού με αυτά. Το ίδιο ισχύει και για την ομοιότητα ως προς τα κλειδιά, όπως φαίνεται στη γραφική παράσταση του σχήματος 6.15. Για τις ιδιότητες δεν ορίζεται κέρδος αφού, όπως εξηγήθηκε παραπάνω, τα ποσοστά ομοιότητας του συμπιεσμένου με τα αρχικά ταυτίζονται με τα αντίστοιχα του ασυμπιεστού.

Η μελέτη των απωλειών δεν εξετάζεται στην παρούσα εργασία, διότι, όπως περιγράφηκε παραπάνω, τα P2P δίκτυα που κατασκευάσαμε για τα πειράματα, κατασκευάστηκαν έτσι ώστε οι σχέσεις που αφαιρούνται κατά τη συμπίεση να υπάρχουν σε όλα στο ίδιο πλήθος. Γενικά, όμως, η συμπεριφορά που αναμένεται είναι καθώς αυξάνεται η ομοιότητα μεταξύ των αρχικών σχημάτων, να αυξάνονται και οι απώλειες που εισάγονται κατά τη συμπίεση. Αυτό οφείλεται στο γεγονός ότι όσο πιο όμοια είναι τα σχήματα μεταξύ τους, τόσο πιο σημαντικό είναι κάθε στοιχείο τους, και η αφαίρεση του επηρεάζει πολύ την ομοιότητα του τελικού με τα αρχικά. Έτσι, όσο πιο όμοια είναι τα αρχικά σχήματα μεταξύ τους, τόσο καλύτερο καθολικό και τόσο χειρότερο συμπιεσμένο σχήμα δημιουργείται, δηλαδή η συμπίεση δεν μας προσφέρει θετικά αποτελέσματα. Φυσικά ισχύει και το αντίθετο. Όσο πιο ανόμοια είναι τα αρχικά σχήματα μεταξύ τους, τόσο χειρότερο είναι το καθολικό σχήμα και τόσο καλύτερη συμπίεση μπορεί να γίνει, δηλαδή η συμπίεση είναι χρήσιμη.

6.4.3 Σύγκριση ομοιοτήτων με βάση το βαθμό συμπίεσης

6.4.3.1 Ομοιότητα αρχικών σχημάτων με το τελικό (καθολικό ή συμπιεσμένο) για διάφορους βαθμούς συμπίεσης

6.4.3.1.1 Ομοιότητα ως προς τις σχέσεις (relations)

Από το γράφημα 7 (σχήμα 6.17), παρατηρούμε ότι καθώς αυξάνεται η συμπίεση, αυξάνεται και η ομοιότητα των σχέσεων του τελικού με τα αρχικά σχήματα. Αυτό συμβαίνει διότι, όσο αυξάνεται ο βαθμός συμπίεσης αφαιρούνται σχέσεις, οι οποίες δεν εμφανίζονται σε μεγάλο ποσοστό στα αρχικά σχήματα, οπότε συμβάλλουν λιγότερο από τις υπόλοιπες στην ομοιότητα τελικού με αρχικών σχημάτων. Έτσι, οι σχέσεις που παραμένουν στο συμπιεσμένο σχήμα είναι λιγότερες και συμβάλλουν περισσότερο στην ομοιότητα απ' αυτές που αφαιρέθηκαν. Συνεπώς, η ομοιότητα του τελικού με τα αρχικά σχήματα ενισχύεται καθώς αυξάνεται ο βαθμός συμπίεσης.

6.4.3.1.2 Ομοιότητα ως προς τις ιδιότητες (attributes)

Από το γράφημα 8 (σχήμα 6.18), παρατηρούμε ότι η ομοιότητα του τελικού με τα αρχικά σχήματα ως προς τις ιδιότητες δεν επηρεάζεται από τη συμπίεση. Αυτό οφείλεται στο γεγονός ότι κατά τη συμπίεση αφαιρούνται μόνο σχέσεις, και μάλιστα οι ιδιότητες των σχέσεων που αφαιρούνται προστίθενται κατάλληλες σε άλλες σχέσεις, οπότε δε χάνονται.

6.4.3.1.3 Ομοιότητα ως προς τα κλειδιά (keys)

Από το γράφημα 9 (σχήμα 6.19), παρατηρούμε ότι γενικά καθώς αυξάνεται η συμπίεση, αυξάνεται και η ομοιότητα των κλειδιών του τελικού με τα αρχικά σχήματα. Αυτό συμβαίνει επειδή, καθώς αφαιρούνται σχέσεις κατά τις συμπιέσεις, αφαιρούνται πολλές φορές και τα αντίστοιχα κλειδιά. Ωστόσο, αυτό δεν ισχύει πάντα. Ισχύει μόνο στην περίπτωση που το κλειδί της σχέσης που αφαιρείται δεν υπάρχει σε καμιά άλλη σχέση. Όπως σχολιάσαμε και στην παράγραφο 6.4.1, τα κλειδιά είναι γενικά ανεξάρτητα από τις σχέσεις. Η διασπορά και μεταβολή τους κατά τη δημιουργία των τελικών σχημάτων εξαρτώνται από την κατασκευή του κάθε σχήματος. Έτσι, στο γράφημα 9 παρατηρούμε ότι οι καμπύλες της ομοιότητας του καθολικού και του πρώτου συμπιεσμένου ταυτίζονται, το οποίο σημαίνει ότι το κλειδί της σχέσης που αφαιρέθηκε υπήρχε και σε άλλη σχέση του τελικού σχήματος. Αντίθετα, τα

κλειδιά των σχέσεων που αφαιρέθηκαν κατά την δεύτερη και τρίτη συμπίεση ήταν μοναδικά, και για το λόγο αυτό η ομοιότητα των κλειδιών του τελικού με τα αρχικά μεταβλήθηκε.

6.4.3.2 Ομοιότητα συμπιεσμένου με καθολικό καθώς μεταβάλλεται ο βαθμός συμπίεσης

Από το γράφημα 10 (σχήμα 6.20), παρατηρούμε ότι η γραφική παράσταση έχει μορφή ευθείας, δηλαδή έχει σταθερή κλίση. Αυτό είναι αναμενόμενο γιατί σε κάθε συμπίεση αφαιρείται μία μόνο σχέση.

Από το γράφημα 11 (σχήμα 6.21), παρατηρούμε ότι η ομοιότητα του καθολικού με το συμπιεσμένο σχήμα είναι 100% για κάθε βαθμό συμπίεσης. Αυτό είναι λογικό γιατί, σε κάθε συμπίεση, αφαιρείται μία σχέση, αλλά οι ιδιότητες της δεν αφαιρούνται, αλλά προστίθενται κατάλληλα σε άλλες σχέσεις. Έτσι, ο αριθμός των ιδιοτήτων (υπολογίζοντας μία φορά πολλαπλές ή όμοιες σχέσεις) δεν μεταβάλλεται.

Όπως και στην ομοιότητα ως προς τις σχέσεις, το γράφημα 12 (σχήμα 6.22) έχει μορφή ευθείας, δηλαδή έχει σταθερή κλίση. Αυτό συμβαίνει γιατί σε κάθε συμπίεση αφαιρείται μία μόνο σχέση, οπότε σε κάθε βήμα συμπίεσης, μπορεί να αφαιρεθεί το πολύ ένα κλειδί, δηλαδή το κλειδί της σχέσης που αφαιρέθηκε. Φυσικά, στην περίπτωση που το κλειδί της σχέσης που αφαιρείται είναι όμοιο με το κλειδί μίας άλλης σχέσης, ο αριθμός των κλειδιών παραμένει σταθερός.

6.5 Σχήματα παραδειγμάτων

Στην ενότητα αυτή θα παραθέσουμε τα σχήματα βάσεων δεδομένων των κόμβων των P2P δικτύων, που χρησιμοποιήθηκαν για τα πειράματα, και τα αποτελέσματα των οποίων παρουσιάστηκαν στην ενότητα 6.3.

6.5.1 P2P Δίκτυο νοσοκομείων

6.5.1.1 H

Schema 1

Hospital (HID, Name, Place, Capacity)

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, Name, Category, Description)

Schema 2

Hospital (HID, HName, Place, Capacity)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)

Nurse (NID, ID, Name, HID, Salary)

InnerPatient (PID, ID, Name, NoHospitalisations)

OuterPatient (PID, ID, Name, NoVisits)

Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)

Illness (IID, IName, Category, Description)

Schema 3

Hospital (Hospital_id, Address, Web-site)

Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)

Nurse (Nurse_id, Name, Salary, Age, Sex)

Staff (Staff_id, Duty, Name, Salary, Age, Sex)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)

Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Schema 4

HospitalInfo (HID, Name, Type, Capacity)

HospitalContact (HID, Address, Web-site, Phone, Fax)

Doctor (DID, Name, SName, Salary, Speciality)

Nurse (NID, Name, SName, Salary)

Staff (SID, Name, Salary)

Patient (PID, Name, SName, History)

Sector (SName, Speciality, NoBeds, NoNurses)

Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Schema 5

Hospital (HID, Name, Place, Employee)

Doctor (DID, Name, Speciality, Phone, Email)

Nurse (NID, Name)

Therapy (TID, ThName, ThDescription, Cost)

Illness (IID, IName, IDescription)

Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

Correspondences

Name \xleftarrow{ID} HName

HID \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} IName

IID \xleftarrow{ID} Illness_id

Category \xleftarrow{ID} IllnessCategory

Description \xleftarrow{ID} IDescription

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

NID \xleftarrow{ID} Nurse_id

SID \xleftarrow{ID} Staff_id

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.2 HI

Schema 1

Hospital (HID, Name, Place, Capacity)

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, Name, Category, Description)

Schema 2

Hospital (HID, HName, Place, Capacity)

Person (ID, Name, Role, Address, Phone, Age, Sex)
Doctor (DID, ID, Name, Speciality, Salary)
Nurse (NID, ID, Name, HID, Salary)
InnerPatient (PID, ID, Name, NoHospitalisations)
OuterPatient (PID, ID, Name, NoVisits)
Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)
Illness (IID, IName, Category, Description)

Schema 3

Hospital (Hospital_id, Address, Web-site)
Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)
Nurse (Nurse_id, Name, Salary, Age, Sex)
Staff (Staff_id, Duty, Name, Salary, Age, Sex)
Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)
Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Schema 4

HospitalInfo (HID, Name, Type, Capacity)
HospitalContact (HID, Address, Web-site, Phone, Fax)
Doctor (DID, Name, SName, Salary, Speciality)
Nurse (NID, Name, SName, Salary)
Staff (SID, Name, Salary)
Patient (PID, Name, SName, History)
Sector (SName, Speciality, NoBeds, NoNurses)
Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)
Therapy (TID, ThName, ThDescription, Cost, T1, T2, T3, T4, T5)
Diagnosis (PID, Date, DID, MID, Comments, D6, D7, D8, D9, D10)
Medication (MID, PID, Date, TID, Illness, MedDescription)
Cures (Illness, TID)

Schema 5

Hospital (HID, Name, Place, Employee)

Doctor (DID, Name, Speciality, Phone, Email)

Nurse (NID, Name)

Therapy (TID, ThName, ThDescription, Cost)

Illness (IID, IName, IDescription)

Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID, C1, C2, C3, C4, C5, C6)

Sector (SName, Speciality, NoBeds, NoNurses, S7, S8, S9, S10, S11, S12)

Correspondences

Name \xleftarrow{ID} HName

HID \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} IName

IID \xleftarrow{ID} Illness_id

Category \xleftarrow{ID} IllnessCategory

Description \xleftarrow{ID} IDescription

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

NID \xleftarrow{ID} Nurse_id

SID \xleftarrow{ID} Staff_id

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.3 H2

Schema1

Hospital (HID, Name, Place, Capacity)

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Illness (IID, Name, Category, Description)

Schema2

Hospital (HID, HName, Place, Capacity)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)
InnerPatient (PID, ID, Name, NoHospitalisation)
OuterPatient (PID, ID, Name, NoVisits)
Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, Illness)

Schema3

Hospital (Hospital_id, Address, Web-site)
Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)
Nurse (Nurse_id, Name, Salary, Age, Sex, Hospital_id)
Staff (Staff_id, Duty, Name, Salary, Age, Sex)
Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)
Illness (Illness_id, IllnessCategory, Therapy)

Schema4

HospitalInfo (HID, Name, Type, Capacity)
HospitalContact (HID, Address, Web-site, Phone)
Doctor (DID, Name, SName, Salary, Speciality)
Patient (PID, Name, SName, History)
Sector (SName, Speciality, NoBeds, NoNurses)
Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness)

Schema5

Hospital (HID, Name, Place, Employee)
Doctor (DID, Name, Speciality, Phone)
Therapy (TID, ThDescription)
Patient (PID, Name, Address, Phone)
Diagnosis (PID, Date, DID, MID)
Medication (MID, PID, Date, TID, Illness, MedDescription)
Cures (Illness, TID)

Correspondences

Name \xleftarrow{ID} HName

HID \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

IID \xleftarrow{ID} Illness_id

Category \xleftarrow{ID} IllnessCategory

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.4 H3

Schema1

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Schema2

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)

InnerPatient (PID, ID, Name)

OuterPatient (PID, ID, Name)

Hospitalisation (PID, HID, DateOfHospitalisation, DID, DateOut, Illness)

Schema3

Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)

Nurse (Nurse_id, Name, Salary, Age, Sex, Hospital)

Staff (Staff_id, Duty, Name, Salary, Age, Sex)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)

Illness (Illness_id, IllnessCategory, Therapy)

Schema4

HospitalInfo (HID, Name, Type, Capacity)

HospitalContact (HID, Address, Web-site, Phone)

Doctor (DID, Name, SName, Salary, Speciality)

Patient (PID, Name, SName, History)

Sector (SName, Speciality, NoBeds, NoNurses)

Schema5

Hospital (HID, Name, Place, Employee, NoBeds, NoNurses, Web-site)

Doctor (DID, Name, Speciality, Phone)

Therapy (TID, ThName, ThDescription)

Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID)

Medication (MID, PID, Date, TID, Illness, MedDescription)

Cures (Illness, TID)

Correspondences

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.5 H4

Schema1

Hospital (HID, Name, Place, Capacity)

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, Name, Category, Description)

Schema2

Hospital (HID, HName, Place, Capacity)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)

Nurse (NID, ID, Name, HID, Salary)

InnerPatient (PID, ID, Name)

OuterPatient (PID, ID, Name)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, IName, Category, Description)

Schema3

Hospital (Hospital_id, Address, Web-site)

Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)

Nurse (Nurse_id, Name, Salary, Age, Sex)

Staff (Staff_id, Name, Salary, Age, Sex)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)

Illness (Illness_id, IllnessCategory)

Schema4

HospitalInfo (HID, Name, Capacity)

HospitalContact (HID, Address, Web-site, Phone)
Doctor (DID, Name, SName, Salary, Speciality)
Nurse (NID, Name, SName, Salary)
Staff (SID, Name, Salary)
Patient (PID, Name, SName)
Sector (SName, Speciality)
Hospitalisation (PID, HID, DateOfHospitalisation, DID)

Schema5

Hospital (HID, Name, Place, Employee)
Doctor (DID, Name, Speciality, Phone)
Nurse (NID, Name)
Therapy (TID, ThDescription)
Illness (IID, IName, IDescription)
Patient (PID, Name, Address, Phone)
Diagnosis (PID, Date, DID)
Medication (MID, PID, Date, TID, IID)
Cures (IID, TID)

Correspondences

Name- \xleftarrow{ID} HName
HIDI \xleftarrow{ID} Hospital_id
Address \xleftarrow{ID} Place
Name \xleftarrow{ID} IName
IID \xleftarrow{ID} Illness_id
Category \xleftarrow{ID} IllnessCategory
Description \xleftarrow{ID} IDescription
PID \xleftarrow{ID} Patient_id
DID \xleftarrow{ID} Doctor_id
NID \xleftarrow{ID} Nurse_id

SID \xleftarrow{ID} Staff_id

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.6 H5

Schema1

Hospital (HID, Name, Place, NoDoctors, OpenHours, NightDuty)

Employee (EID, Name, HID, Job, Salary, DateOfEmployment, Contract, HoursPerWeek)

Patient (PID, Name, Phone, Age, ChronicDisease)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, Name, Category, Description)

Schema2

Hospital (HID, HName, Place, Expanse, TaxID)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)

Nurse (NID, ID, Name, HID, Salary, Grade)

InnerPatient (PID, ID, Name, NoHospitalisations)

OuterPatient (PID, ID, Name, NoVisits)

Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)

Illness (IID, IName, Description, IsChronic)

Schema3

Hospital (Hospital_id, Address, Web-site, Ownership, NoEmployees)

Doctor (Doctor_id, Name, Age, Speciality)

Nurse (Nurse_id, Name, Experience)

Staff (Staff_id, Duty, Name, Education)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)

Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Schema4

HospitalInfo (HID, Name, Type, Capacity, NoSectors)

HospitalContact (HID, Address, Web-site, Phone, Fax)

Doctor (DID, Name, SName, Speciality, UniversityOfStudies, IsProfessor)

Nurse (NID, Name, SName)

Staff (SID, Name)

Patient (PID, Name, SName, History)

Sector (SName, Speciality, NoBeds, NoNurses)

Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Schema5

Hospital (HID, Name, Employee)

Doctor (DID, Name, Speciality, Email)

Nurse (NID, Name)

Therapy (TID, ThName, ThDescription, Cost, MeansOfPayment)

Illness (IID, IName, IDescription)

Patient (PID, Name, Address, Phone, Insurance)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription, MeansOfTake, Duration,
Frequency)

Cures (IID, TID, Efficiency, TimeOfTherapy)

Correspondences

Name \xleftarrow{ID} HName

HID- \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} IName

IID \xleftarrow{ID} Illness_id

Category- \xleftarrow{ID} IllnessCategory

Description \xleftarrow{ID} IDescription

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

NID \xleftarrow{ID} Nurse_id

SID \xleftarrow{ID} Staff_id

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.7 H6

Schema1

Hospital (HID, Name, Place, NoDoctors, OpenHours, NightDuty, H1)

Employee (EID, Name, HID, Job, Salary, DateOfEmployment, Contract, HoursPerWeek, E1)

Patient (PID, Name, Phone, Age, ChronicDisease, P1)

Hospitalisation (PID, HID, DateOfHospitalisation, IID, Hs1)

Illness (IID, Name, Category, Description, I1)

Schema2

Hospital (HID, HName, Place, Expanse, TaxID)
Person (ID, Name, Role, Address, Phone, Age, Sex, P2)
Doctor (DID, ID, Name, Speciality, Salary)
Nurse (NID, ID, Name, HID, Salary, Grade, N2a, N2b)
InnerPatient (PID, ID, Name, NoHospitalisations, Operations, In2)
OuterPatient (PID, ID, Name, NoVisits, Out2a, Out2b)
Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)
Illness (IID, IName, IsChronic, I2)

Schema3

Hospital (Hospital_id, Address, Web-site, Ownership, NoEmployees, H3)
Doctor (Doctor_id, Name, Age, Speciality, VisitHours, Assistances, D3)
Nurse (Nurse_id, Name, Experience, Studies, N3)
Staff (Staff_id, Duty, Name, Education, S3)
Patient (Patient_id, Name, Age, Illness_id, Hospitalisation_needed, DateOfHospitalisation,
P3)
Illness (Illness_id, IllnessCategory, Symptoms, Therapy, Seriousness)

Schema4

HospitalInfo (HID, Name, Type, Capacity, NoSectors)
HospitalContact (HID, Address, Web-site, Phone, Fax)
Doctor (DID, Name, SName, Speciality, UniversityOfStudies, IsProfessor, Practice)
Nurse (NID, Name, SName, N4)
Staff (SID, Name, S4)
Patient (PID, Name, SName, History, P4)
Sector (SName, Speciality, NoBeds, NoNurses, NoPractices, NoOffices, Norooms, Director)
Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Schema5

Hospital (HID, Name, Employee)
Doctor (DID, Name, Speciality, Email)

Nurse (NID, Name)

Therapy (TID, ThName, ThDescription, Cost, MeansOfPayment, ThDuration,
EstimatedResult)

Illness (IID, IName)

Patient (PID, Name, Address, Phone, Insurance)

Diagnosis (PID, Date, DID, MID, Comments, Possibility)

Medication (MID, PID, Date, TID, IID, MedDescription, MeansOfTake, Duration,
Frequency, PharmCompany, Problems, Alternative)

Cures (IID, TID, Efficiency, TimeOfTherapy, Details)

Correspondences

Name \xleftarrow{ID} HName

HID \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} IName

IID \xleftarrow{ID} Illness_id

Category \xleftarrow{ID} IllnessCategory

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

NID \xleftarrow{ID} Nurse_id

SID \xleftarrow{ID} Staff_id

Hospital \xleftarrow{ID} HospitalInfo

Hospita \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

OuterPatient (PID, ID, Name, NoVisits)

Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)

Illness (IID, IName, Category, Description)

Schema3

Hospital (Hospital_id, Address, Web-site)

Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)

Nurse (Nurse_id, Name, Salary, Age, Sex)

Staff (Staff_id, Duty, Name, Salary, Age, Sex)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)

Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Schema4

HospitalInfo (HID, Name, Type, Capacity)

HospitalContact (HID, Address, Web-site, Phone, Fax)

Doctor (DID, Name, SName, Salary, Speciality)

Nurse (NID, Name, SName, Salary)

Staff (SID, Name, Salary)

Patient (PID, Name, SName, History)

Sector (SName, Speciality, NoBeds, NoNurses)

Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Schema5

Hospital (HID, Name, Place, Employee)

Doctor (DID, Name, Speciality, Phone, Email)

Nurse (NID, Name)

Therapy (IID, TID, ThName, ThDescription, Cost)

Illness (IID, IName, IDescription)

Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (IID, MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

Correspondences

Name \xleftarrow{ID} HName

HID \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} IName

IID \xleftarrow{ID} Illness_id

Category \xleftarrow{ID} IllnessCategory

Description \xleftarrow{ID} IDescription

PID \xleftarrow{ID} Patient_id

DID \xleftarrow{ID} Doctor_id

NID \xleftarrow{ID} Nurse_id

SID \xleftarrow{ID} Staff_id

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)

Nurse (Nurse_id, Name, Salary, Age, Sex)

Staff (Staff_id, Duty, Name, Salary, Age, Sex)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,
DateOfHospitalisation)

Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Schema4

HospitalInfo (HID, Name, Type, Capacity)

HospitalContact (HID, Address, Web-site, Phone, Fax)

Doctor (DID, Name, SName, Salary, Speciality)

Nurse (NID, Name, SName, Salary)

Staff (SID, Name, Salary)

Patient (PID, Name, SName, History)

Sector (SName, Speciality, NoBeds, NoNurses)

Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Schema5

Hospital (HID, Name, Place, Employee)

Doctor (DID, Name, Speciality, Phone, Email)

Nurse (NID, Name)

Therapy (TID, ThName, ThDescription, Cost)

Illness (IID, IName, IDescription)

Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

Correspondences

Name \xleftarrow{ID} HName

HID \xleftarrow{ID} Hospital_id

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} Name

IID \xleftarrow{ID} Illness_id

Category \xleftarrow{ID} IllnessCategory

Description \xleftarrow{ID} IDescription

NID \xleftarrow{ID} Nurse_id

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.1.10 H9

Schema1

Hospital (HID, Name, Place, Capacity)

Employee (EID, Name, HID, Job, Salary)

Patient (PID, Name, Address, Phone, Age, Sex)

Hospitalisation (PID, HID, DateOfHospitalisation, IID)

Illness (IID, Name, Category, Description)

Schema2

Hospital (HID, HName, Place, Capacity)

Person (ID, Name, Role, Address, Phone, Age, Sex)

Doctor (DID, ID, Name, Speciality, Salary)

Nurse (NID, ID, Name, HID, Salary)

InnerPatient (PID, ID, Name, NoHospitalisations)

OuterPatient (PID, ID, Name, NoVisits)

Hospitalisation (PID, HID, DateOfHospitalisation, DateOut, IID)

Illness (IID, IName, Category, Description)

Schema3

Hospital (Hospital_id, Address, Web-site)

Doctor (Doctor_id, Name, Salary, Age, Sex, Speciality)

Nurse (Nurse_id, Name, Salary, Age, Sex)

Staff (Staff_id, Duty, Name, Salary, Age, Sex)

Patient (Patient_id, Name, Age, Sex, Illness_id, Hospitalisation_needed,

DateOfHospitalisation)

Illness (Illness_id, IllnessCategory, Symptoms, Therapy)

Schema4

HospitalInfo (HID, Name, Type, Capacity)

HospitalContact (HID, Address, Web-site, Phone, Fax)

Doctor (DID, Name, SName, Salary, Speciality)

Nurse (NID, Name, SName, Salary)

Staff (SID, Name, Salary)

Patient (PID, Name, SName, History)

Sector (SName, Speciality, NoBeds, NoNurses)

Hospitalisation (PID, HID, DateOfHospitalisation, DID, Illness, Bed, Days)

Schema5

Hospital (HID, Name, Place, Employee)

Doctor (DID, Name, Speciality, Phone, Email)

Nurse (NID, Name)

Therapy (TID, ThName, ThDescription, Cost)

Illness (IID, IName, IDescription)

Patient (PID, Name, Address, Phone)

Diagnosis (PID, Date, DID, MID, Comments)

Medication (MID, PID, Date, TID, IID, MedDescription)

Cures (IID, TID)

Correspondences

Name \xleftarrow{ID} HName

Address \xleftarrow{ID} Place

Name \xleftarrow{ID} IName

Category \xleftarrow{ID} IllnessCategory

Description \xleftarrow{ID} IDescription

Hospital \xleftarrow{ID} HospitalInfo

Hospital \xleftarrow{ID} HospitalContact

InnerPatient \xrightarrow{ISA} Person

OuterPatient \xrightarrow{ISA} Person

Mappings

Person(ID, Name, Role), Role = Doctor :- Doctor(DID, Name)

Person(ID, Name, Role), Role = Doctor :- Doctor(Doctor_id, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(NID, Name)

Person(ID, Name, Role), Role = Nurse :- Nurse(Nurse_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(Staff_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(PID, Name)

Person(ID, Name, Role), Role = Patient :- Patient(Patient_id, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(DID, Name)

Employee(EID, Name, Job), Job = Doctor :- Doctor(Doctor_id, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(NID, Name)

Employee(EID, Name, Job), Job = Nurse :- Nurse(Nurse_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(Staff_id, Name)

Employee(EID, Name, Job), Job = Staff :- Staff(SID, Name)

Person(ID, Name, Role), Role = Employee :- Employee(EID, Name)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = InnerPatient :-

InnerPatient(PID, Name, ID)

Patient(Patient_id, Hospitalisation_needed), Hospitalisation_needed = OuterPatient :-

OuterPatient(PID, Name, ID)

6.5.2 P2P Δίκτυο πανεπιστημίων

6.5.2.1 U

Schema 1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth,

Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema 2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek,
HasLab, Assistancess)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema 3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema 4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

HasCourse \xleftarrow{ID} Students_courses

FID \xleftarrow{ID} Faculty_id

SID \xleftarrow{ID} Student_id

DID \xleftarrow{ID} Dept_id

CID \xleftarrow{ID} Course_id

Faculty \xleftarrow{ID} Professor

FID \xleftarrow{ID} IdP

SID \xleftarrow{ID} IdSt

CID \xleftarrow{ID} IdC

Name \xleftarrow{ID} CName

Student \xleftarrow{ID} Student1

Student \xleftarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.2 UI

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Laboratory (Lab_id, LabName, FID, OpenHours, NoCourses, Department, Equipment)

Research_Programm (Res_id, FID, ResearchTitle, Duration, FinancialSource, Results, StartDate)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab, Assurances)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Scholarship (SSID, Student_id, Amount, Organisation, Preconditions, NoStudents)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason, ApprovalDate, Certificates, Instalments,
Comments)

Correspondences

HasCourse \xleftrightarrow{ID} Students_courses

FID \xleftrightarrow{ID} Faculty_id

SID \xleftrightarrow{ID} Student_id

DID \xleftrightarrow{ID} Dept_id

CID \xleftrightarrow{ID} Course_id

Faculty \xleftrightarrow{ID} Professor

Department \xleftrightarrow{ID} Sector

FID \xleftrightarrow{ID} IdP

SID \xleftrightarrow{ID} IdSt

DID \xleftrightarrow{ID} SectorId

DepName \xleftrightarrow{ID} SectorName

CID \xleftrightarrow{ID} IdC

Course \xleftrightarrow{ID} Exam

Course \xleftrightarrow{ID} Lesson

Name \xleftrightarrow{ID} CName

Student \xleftrightarrow{ID} Student1

Student \xleftrightarrow{ID} Student2

Course \xleftrightarrow{ID} Teached

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

Publication(Title, FID, Description) :- Book(Title, FID, Description)

6.5.2.3 U2

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Laboratory (Lab_id, LabName, FID, OpenHours, NoCourses, Department, Equipment)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab,
Assistances)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

HasCourse \xleftarrow{ID} Students_courses

FID \xleftarrow{ID} Faculty_id

SID \xleftarrow{ID} Student_id

DID \xleftarrow{ID} Dept_id

CID \xleftarrow{ID} Course_id

FacultyIDProfessor

FID \xleftarrow{ID} IdP

SID \xleftarrow{ID} IdSt

CID \xleftarrow{ID} IdC

Name \xleftarrow{ID} CName

Student \xleftarrow{ID} Student1

Student \xleftarrow{ID} Student2

Course \xleftarrow{ID} Taached

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

Publication(Title, FID, Description) :- Book(Title, FID, Description)

6.5.2.4 U3

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab, Assistancess)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName, Budget)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

FID \xleftrightarrow{ID} Faculty_id

SID \xleftrightarrow{ID} Student_id

CID \xleftrightarrow{ID} Course_id

SID \xleftrightarrow{ID} IdSt

CID \xleftrightarrow{ID} IdC

Name \xleftrightarrow{ID} CName

Student \xleftrightarrow{ID} Student1

Student \xleftrightarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.5 U4

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Age, Sex, Salary, Email, Address)

Department (DID, DepName)

Course (CID, Name, FID, Semester, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id)

Faculty (Faculty_id, Dept_id, Name, Age, Sex, Salary, Email, Address)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, ExamDate, LessonHours,
LessonPlace, Book)

Laboratory (Lab_id, Dept_id, Faculty_id)

Research_Programm (Res_id, Subject, Faculty_id)

Cooperation (Co_id, University)

Schema3

Student (IdSt, Name, SectorId, Semester, Type)

Professor (IdP, Name, SectorId, IdC, Salary, Age, Sex, Address, Email, Phone)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory)

Exam (IdC, ExamDate)

Lesson (IdC, LessonHours, LessonPlace)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID)

HasCourse (CID, SID, Grade)

Department (DepName)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source)

Correspondences

HasCours \xleftarrow{ID} Students_courses

FID \xleftarrow{ID} Faculty_id

SID \xleftarrow{ID} Student_id

DID \xleftarrow{ID} Dept_id

CID \xleftarrow{ID} Course_id

Faculty \xleftarrow{ID} Professor

FID \xleftarrow{ID} IdP

SID \xleftarrow{ID} IdSt

CID \xleftarrow{ID} IdC

Name \xleftarrow{ID} CName

Student \xleftarrow{ID} Student1

Student \xleftarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)
 Person(ID, Name, Role), Role = Student :- Student(SID, Name)
 Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)
 Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)
 Person(Name), Name = FirstName :- Name(FirstName)
 Person(Name), Name = SurName :- Name(SurName)
 Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.6 U5

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)
 Faculty (FID, Name, DID, CID, Title, Salary)
 Student (SID, Name, Semester)
 Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address)
 Department (DID, DepName)
 Course (CID, Name, FID, Semester, Book)
 Publication (Title, FID, Description)
 Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id)
 Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address)
 Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)
 Course (Course_id, Faculty_id, Dept_id, Semester, Publication)
 Laboratory (Lab_id, Dept_id, Faculty_id)
 Research_Programm (Res_id, Subject, Faculty_id)
 Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId)
 Professor (IdP, Name, SectorId, IdC, Salary)

Sector (SectorId, SectorName)
Course (IdC, SectorId, IdP, Obligatory)
Exam (IdC, ExamDate)
Lesson (IdC, LessonHours, LessonPlace)
Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)
Student1 (SID, Name, Address, Phone, Email)
Student2 (SID, Type, Semester)
Course (CID, CName, DepName)
Teached (CID, FID, Year)
HasCourse (CID, SID, Year, Grade)
Department (DepName)
Book (Title, FID, Description)
Scholarship (SSID, SID, Amount, Source)

Correspondences

HasCourse \xleftrightarrow{ID} Students_courses
FID \xleftrightarrow{ID} Faculty_id
SID \xleftrightarrow{ID} Student_id
DID \xleftrightarrow{ID} Dept_id
CID \xleftrightarrow{ID} Course_id
Faculty \xleftrightarrow{ID} Professor
FID \xleftrightarrow{ID} IdP
SID \xleftrightarrow{ID} IdSt
CID \xleftrightarrow{ID} IdC
Name \xleftrightarrow{ID} CName
Student \xleftrightarrow{ID} Student1

Student \xleftrightarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.7 U6

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Studies)

Student (Student_id, Dept_id, Name)

Course (Course_id, Faculty_id, Dept_id, Publication, HoursPerWeek, HasLab, Assistances)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)
 Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)
 Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)
 Professor (IdP, Name, SectorId, IdC, UniversityOfStudy, PhD)
 Sector (SectorId, SectorName)
 Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)
 Exam (IdC, ExamDate, ExamPlace, ExamDuration)
 Lesson (IdC, LessonHours, LessonPlace, LessonDuration)
 Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Salary)
 Student1 (SID, Name)
 Student2 (SID, Type)
 Course (CID, CName, DepName)
 Teached (CID, FID, Year)
 HasCourse (CID, SID, Year, Grade)
 Department (DepName, Budget)
 Book (Title, FID, Description)
 Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

HasCourse \xleftrightarrow{ID} Students_courses
 FID \xleftrightarrow{ID} Faculty_id
 SID \xleftrightarrow{ID} Student_id
 DID \xleftrightarrow{ID} Dept_id
 CID \xleftrightarrow{ID} Course_id
 Faculty \xleftrightarrow{ID} Professor

$FID \xleftarrow{ID} IdP$

$SID \xleftarrow{ID} IdSt$

$CID \xleftarrow{ID} IdC$

$Name \xleftarrow{ID} CName$

$Student \xleftarrow{ID} Student1$

$Student \xleftarrow{ID} Student2$

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.8 U7

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab,
Assistances)

Laboratory (Faculty_id, Lab_id, LabName, Dept_id, Capacity, NoMembers)

Research_Programm (Faculty_id, Res_id, Subject, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SID, SSID, Amount, Source, Reason)

Correspondences

HasCourse \xleftrightarrow{ID} Students_courses

FID \xleftarrow{ID} Faculty_id

SID \xleftarrow{ID} Student_id

DID \xleftarrow{ID} Dept_id

CID \xleftarrow{ID} Course_id

Faculty \xleftarrow{ID} Professor

Department \xleftarrow{ID} Sector

FID \xleftarrow{ID} IdP

SID \xleftarrow{ID} IdSt

DID \xleftarrow{ID} SectorId

DepName \xleftarrow{ID} SectorName

CID \xleftarrow{ID} IdC

Name \xleftarrow{ID} CName

Student \xleftarrow{ID} Student1

Student \xleftarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.9 U8

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab, Assistances)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

HasCourse \xleftarrow{ID} Students_courses

FID \xleftarrow{ID} Faculty_id

DID \xleftarrow{ID} Dept_id

CID \xleftarrow{ID} Course_id

SID \xleftarrow{ID} IdSt

CID \xleftarrow{ID} IdC

Name \xleftarrow{ID} CName

Student \xleftarrow{ID} Student1

Student \xleftarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.10 U9

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab, Assistancess)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

HasCourse \xleftrightarrow{ID} Students_courses

SID \xleftrightarrow{ID} IdSt

CID \xleftrightarrow{ID} IdC

Name \xleftrightarrow{ID} CName

Student \xleftrightarrow{ID} Student1

Student \xleftrightarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

6.5.2.11 U10

Schema1

Person (ID, Name, Role, Address, Phone, Email, Sex, Age)

Faculty (FID, Name, DID, CID, Title, Salary)

Student (SID, Name, Semester)

Staff (ID, Name, Responsibility, Age, Sex, Salary, Email, Address, HoursPerMonth, Vacations)

Department (DID, DepName, DDescription)

Course (CID, Name, FID, Semester, CDescription, Book)

Publication (Title, FID, Description)

Name (FirstName, SurName, CourseName)

Schema2

Department (Dept_id, Web-site, NoCourses, NoLabs, NoFaculties)

Faculty (Faculty_id, Dept_id, Field, Name, Age, Sex, Salary, Email, Address, Studies)

Student (Student_id, Dept_id, Name, Age, Sex, Type, Email, Address)

Course (Course_id, Faculty_id, Dept_id, Semester, Publication, HoursPerWeek, HasLab, Assistancess)

Laboratory (Lab_id, LabName, Dept_id, Faculty_id, Capacity, NoMembers)

Research_Programm (Res_id, Subject, Faculty_id, Financing, NoResearchers)

Cooperation (Co_id, University, CooperationType)

Schema3

Student (IdSt, Name, SectorId, MatriculationYear, PreviousStudies)

Professor (IdP, Name, SectorId, IdC, Salary, UniversityOfStudy, PhD)

Sector (SectorId, SectorName)

Course (IdC, SectorId, IdP, Obligatory, Difficulty, URL, Prerequisite, MaxNoStudents)

Exam (IdC, ExamDate, ExamPlace, ExamDuration)

Lesson (IdC, LessonHours, LessonPlace, LessonDuration)

Students_courses (IdSt, IdC, Grade)

Schema4

Faculty (FID, Name, DepName, Address, Phone, Email, Salary)

Student1 (SID, Name, Address, Phone, Email)

Student2 (SID, Type, Semester)

Course (CID, CName, DepName)

Teached (CID, FID, Year)

HasCourse (CID, SID, Year, Grade)

Department (DepName, Budget)

Book (Title, FID, Description)

Scholarship (SSID, SID, Amount, Source, Reason)

Correspondences

HasCourse \xleftrightarrow{ID} Students_courses

Name- \xleftrightarrow{ID} CName

Student \xleftrightarrow{ID} Student1

Student \xleftrightarrow{ID} Student2

Mappings

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(Faculty_id, Name)

Person(ID, Name, Role), Role = Faculty :- Faculty(FID, Name)

Person(ID, Name, Role), Role = Staff :- Staff(ID, Name)

Person(ID, Name, Role), Role = Student :- Student(SID, Name)

Person(ID, Name, Role), Role = Student :- Student(Student_id, Name)

Person(ID, Name, Role), Role = Student :- Student(IdSt, Name)

Person(Name), Name = FirstName :- Name(FirstName)

Person(Name), Name = SurName :- Name(SurName)

Course(Name), Name = CourseName :- Name(CourseName)

7

Επίλογος

7.1 Σύνοψη και συμπεράσματα

Στα πλαίσια της διπλωματικής αυτής εργασίας ασχοληθήκαμε με την ανάπτυξη ενός πρότυπου συστήματος διαχείρισης ολοκλήρωσης σχημάτων βάσεων δεδομένων για δίκτυα ομότιμων κόμβων (P2P schema-based systems). Συγκεκριμένα υλοποιήσαμε ένα σύστημα για την εξαγωγή ενός κεντρικού ή καθολικού σχήματος (global schema) για ένα σύνολο κόμβων, όπου ο καθένας έχει το δικό του τοπικό σχήμα (local schema) βάσης δεδομένων. Το κεντρικό σχήμα προκύπτει από την ενοποίηση των τοπικών σχημάτων με τη βοήθεια κανόνων αντιστοίχισης, οι οποίοι ορίζουν σημασιολογικές σχέσεις μεταξύ των τοπικών σχημάτων.

Με τη βοήθεια του καθολικού σχήματος και των αντιστοιχίσεων μεταξύ των τοπικών σχημάτων και του καθολικού σχήματος γίνεται δυνατή μία πιο αποδοτική και ευέλικτη μετάφραση και αποτίμηση ερωτημάτων σε οποιονδήποτε κόμβο το δικτύου.

Το σύστημα ελέγχθηκε με μία σειρά πειραμάτων, όπου αξιολογήθηκε το τελικό σχήμα που δημιουργείται, σε σχέση με τα αρχικά σχήματα εισόδου και τις παραμέτρους σύγκρισης των σχημάτων (σχέσεις, ιδιότητες, κλειδιά).

Τα συμπεράσματα που εξήχθησαν συνοψίζονται στα ακόλουθα:

1. Η ομοιότητα των ιδιοτήτων του καθολικού με τα αρχικά σχήματα επηρεάζεται περισσότερο από αυτή των σχέσεων, και η τελευταία περισσότερο από αυτή των

κλειδιών, σε μία μεταβολή της ομοιότητας των αντίστοιχων παραμέτρων των αρχικών σχημάτων.

2. Όσο αυξάνεται η ομοιότητα μεταξύ των αρχικών σχημάτων, τόσο περισσότερο αυξάνεται και η ομοιότητα του καθολικού σχήματος με αυτά.
3. Τα κλειδιά είναι γενικά ανεξάρτητα από τις σχέσεις. Η διασπορά και μεταβολή τους κατά τη δημιουργία των τελικών σχημάτων εξαρτώνται από την κατασκευή των σχημάτων. Έτσι, τα αποτελέσματα που λαμβάνουμε για τη συμπεριφορά της ομοιότητας των κλειδιών είναι κατά μεγάλο ποσοστό τυχαία και γενικά δεν ακολουθούν συγκεκριμένους κανόνες.
4. Η ομοιότητα του συμπιεσμένου σχήματος με τα αρχικά μεταβάλλεται ως προς την ομοιότητα των αρχικών, με τον ίδιο τρόπο που μεταβάλλεται η αντίστοιχη ομοιότητα του ασυμπιεστού καθολικού σχήματος.
5. Η ομοιότητα των ιδιοτήτων δεν μεταβάλλεται κατά τη συμπίεση.
6. Το κέρδος (σχετικά με το ποσοστό ομοιότητας τελικού με αρχικά σχήματα) που εισάγεται στο συμπιεσμένο σχήμα κατά τη συμπίεση αυξάνεται, καθώς αυξάνεται η ομοιότητα των αρχικών σχημάτων.
7. Οι απώλειες (απώλεια πληροφορίας) που εισάγονται στο συμπιεσμένο σχήμα κατά τη συμπίεση αυξάνονται, με την αύξηση της ομοιότητας μεταξύ των αρχικών σχημάτων.
8. Όσο μεγαλύτερη είναι η ομοιότητα μεταξύ των αρχικών σχημάτων, τόσο καλύτερο καθολικό και τόσο χειρότερο συμπιεσμένο σχήμα δημιουργείται, δηλαδή η συμπίεση δεν μας προσφέρει θετικά αποτελέσματα. Και αντίστροφα, όσο μικρότερο είναι το ποσοστό ομοιότητας των αρχικών σχημάτων, τόσο χειρότερο είναι το καθολικό σχήμα και τόσο καλύτερη συμπίεση μπορεί να γίνει, δηλαδή η συμπίεση είναι χρήσιμη.
9. Καθώς αυξάνεται η συμπίεση, αυξάνεται και η ομοιότητα των σχέσεων του τελικού με τα αρχικά σχήματα.
10. Η μεταβολή της ομοιότητας του συμπιεσμένου καθολικού σχήματος με το ασυμπιεστο ως προς τις σχέσεις ή τα κλειδιά είναι γραμμικά αντιστρόφως ανάλογη του βαθμού συμπίεσης.

7.2 Μελλοντικές επεκτάσεις

Στην ενότητα αυτή, παρουσιάζουμε μερικές προτάσεις για περαιτέρω επέκταση και βελτίωση του συστήματος.

Αρχικά, ως προς τις σχεδιαστικές αποφάσεις που λάβαμε για τη δημιουργία του καθολικού σχήματος, μία πιθανή επέκταση είναι η εισαγωγή περισσότερων ειδών σημασιολογικών συσχετίσεων μεταξύ των στοιχείων των σχημάτων, ώστε οι πληροφορίες να οργανώνονται με πιο αποδοτικά, δημιουργώντας καθολικό σχήμα μικρότερου μεγέθους.

Σχετικά με τη συμπίεση του καθολικού σχήματος, στην εργασία αυτή θεωρήσαμε ως μετρική του μεγέθους μόνο τις σχέσεις, με αφαίρεση των οποίων και υλοποιήθηκε η συμπίεση. Το πλήθος των ιδιοτήτων, όμως, επηρεάζει σημαντικά το μέγεθος ενός σχήματος, ειδικά αν το τελευταίο έχει μικρό αριθμό σχέσεων. Επίσης, ο τύπος των ιδιοτήτων, που καθορίζει τον χώρο που καταλαμβάνουν στη μνήμη, παίζει σπουδαίο ρόλο. Συνεπώς, μία πιο αντιπροσωπευτική μετρική του μεγέθους θα μπορούσε να είναι ο αριθμός των bytes που καταλαμβάνει το σχήμα στη μνήμη. Έτσι, θα μπορούσε να γίνεται έλεγχος κατά πόσο αναγκαία είναι η αφαίρεση ιδιοτήτων και αν αυτή μπορεί να γίνει χωρίς σημαντικές απώλειες πληροφορίας. Σ' αυτή την κατεύθυνση θα βοηθούσε η εισαγωγή νέων ειδών σημασιολογικών συσχετίσεων μεταξύ των ιδιοτήτων, ώστε η πληροφορίες τους να μπορούν να οργανωθούν στο καθολικό σχήμα χρησιμοποιώντας λιγότερες ιδιότητες. Ένα παράδειγμα είναι η προσθήκη ISA συσχετίσεων μεταξύ των ιδιοτήτων, οι οποίες στην εργασία μας χρησιμοποιήθηκαν μόνο μεταξύ σχέσεων. Επίσης, μία επιπρόσθετη επέκταση είναι η συμπίεση να μην γίνεται μόνο σε σχέσεις με ISA συσχετίσεις, αλλά πιθανώς, θεωρώντας νέα κριτήρια, και σε άλλες σχέσεις, ώστε να μη σταματάει η συμπίεση τόσο γρήγορα αν δεν υπάρχουν άλλες ISA ενώ το σχήμα είναι μεγαλύτερο από το κατώφλι.

Ως προς την υλοποίηση, επεκτάσεις μπορούν να γίνουν υλοποιώντας περιπλοκότερες αντιστοιχίσεις από τις απλές που πραγματοποιήθηκαν εδώ.

Ακόμα μία πιθανή επέκταση είναι το σύστημα να μην δίνει ως έξοδο μόνο το τελικό σχήμα (καθολικό ή συμπιεσμένο), αλλά και τις τελικές αντιστοιχίσεις που επιτρέπουν την αποδοτική μετάφραση των ερωτημάτων. Η εξαγωγή των αντιστοιχίσεων αυτών θα γίνεται με τη χρήση κατάλληλων κανόνων, όπως αυτών που περιγράφονται στην παράγραφο 4.5.

Τέλος, υπάρχουν πολλά περιθώρια για περαιτέρω πειραματική μελέτη του συστήματος, σχετικά με τη διασπορά των κλειδιών στα σχήματα, τις απώλειες που εισάγονται κατά τη συμπίεση του καθολικού σχήματος, κ.α.

8

Βιβλιογραφία

- [bilkent] <http://www.cs.bilkent.edu.tr/~erayo/ontology/html/background.html>
- [BGK+02] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu, Data Management for Peer-to-Peer Computing: A Vision, In Proc. of the 5th International Workshop on the Web and Databases (WebDB), June 2002
- [BGM05] Janez Brank, Marko Grobelnik and Dunja Mladenić, A Survey of Ontology Evaluation Technique, In Conference on Data Mining and Data Warehouses (SIKDD 2005), October 2005
- [CX05] Isabel F. Cruz and Huiyong Xiao, The Role of Ontologies in Data Integration, In International journal of engineering intelligent systems for electrical engineering and communications, 2005
- [DMD02] AnHai Doan, Jayant Madhavan, Pedro Domingos and Alon Halevy, Learning to Map between Ontologies on the Semantic Web, In the 11th international conference on World Wide Web, 2002
- [GHI01] Steven Gribble, Alon Halevy, Zachary Ives, Maya Rodrig and Dan Suciu, What Can Databases Do for Peer-to-Peer?, 3rd WebDB Workshop, 2001
- [GM03] Asunción Gómez-Pérez, David Manzano-Macho, A survey on ontology learning methods and technics, IST Project IST-2000-29243, 2003

- [HIS+03] Alon Y. Halevy, Zachary G. Ives, Dan Suciu and Igor Tatarinov, Schema Mediation in Peer Data Management Systems, University of Washington, 2003
- [KTS07] Verena Kantere, Dimitrios Tsoumakos, and Timos Sellis, Semantic Grouping of Social Networks in P2P Database Settings, Technical Report, 2007
- [Len02] Maurizio Lenzerini, Data Integration: A Theoretical Perspective, In ACM PODS 2002, 2002
- [LZ] Yang Li and Ming Zhang, Discussion of Peer-to-Peer Database
- [NB05] Mduduzi Nxumalo and Sonia Berman, Query Translation in Database sharing Peer to Peer networks, Technical Report, Department of Computer Science, University of Cape Town, 2005
- [NSE84] S. B. Navathe, T. Sashidhar and R. Elmasri, Relationship Merging in Schema Integration, In the Tenth International Conference on Very Large Data Bases, 1984
- [PH01] Rachel Pottinger, Alon Halevy, MiniCon: A scalable algorithm for answering queries using views, In The VLDB Journal (2001) 10: 182-198, 2001
- [RB01] Erhard Rahm and Philip A. Bernstein, A survey of approaches to automatic schema matching, In The VLDB Journal 10 :334-350, December 2001
- [wikipedia] <http://en.wikipedia.org/wiki>
- [WPH06] Taowei David Wang, Bijan Parsia and James Hendler, A Survey of the Web Ontology Landscape, International Semantic Web Conference, 2006
- [WVV+01] H.Wache, T. Vogege, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hubner, Ontology-Based Integration of Information — A Survey of Existing Approaches, IJCAI-01 Workshop, 2001
- [Zai03] Ilya Zaihrayeu, Query Answering in Peer-to-Peer Database Networks, Technical Report, March 2003
- [ΜΓ06] Λίνα Μπουντούρη και Μανόλης Γεργατσούλης, Σημασιολογική Ολοκλήρωση με χρήση Οντολογιών, 15^ο Πανελλήνιο Συνέδριο Ακαδημαϊκών Βιβλιοθηκών, 2006
- [Ορφ07] Γεώργιος Ι. Ορφανουδάκης, Υλοποίηση Μηχανισμού Ερωταποκρίσεων για Δίκτυο Ομότιμων Βάσεων Δεδομένων, Διπλ. εργασία Γεωργίου Ι. Ορφανουδάκη, ΕΜΠ 2007