

ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ  
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ / ΜΕΣΟΛΟΓΓΙ

## **Πτυχιακή εργασία**

ΔΗΜΙΟΥΡΓΙΑ ΥΠΗΡΕΣΙΑΣ OLAP ΣΕ ΒΑΣΗ  
ΔΕΔΟΜΕΝΩΝ ΜΕ ΣΤΟΙΧΕΙΑ ΑΞΙΟΛΟΓΗΣΗΣ

Καρβέλης Νικόλαος

Μεσολόγγι 2017



ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ  
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ / ΜΕΣΟΛΟΓΓΙ

## **Πτυχιακή εργασία**

ΔΗΜΙΟΥΡΓΙΑ ΥΠΗΡΕΣΙΑΣ OLAP ΣΕ ΒΑΣΗ  
ΔΕΔΟΜΕΝΩΝ ΜΕ ΣΤΟΙΧΕΙΑ ΑΞΙΟΛΟΓΗΣΗΣ

Καρβέλης Νικόλαος

Επιβλέπων καθηγητής  
Καρούσος Νικόλαος

# Μεσολόγγι 2017

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Διοίκησης Επιχειρήσεων/Μεσολογγίου του ΤΕΙ Δυτικής Ελλάδας δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

## Περιεχόμενα

1	Περίληψη.....	1
2	Ιστορική αναδρομή.....	2
3	Το OLAP αναλυτικά .....	4
3.1	Τι είναι το OLAP;.....	4
3.2	Γιατί χρειαζόμαστε το OLAP;.....	5
3.2.1	Αύξηση των αποθηκευμένων δεδομένων.....	5
3.2.2	Δεδομένα εναντίον Πληροφοριών.....	5
3.2.3	Διάταξη των δεδομένων .....	6
4	Βασικές αρχές του OLAP.....	7
4.1	Τι είναι ένας κύβος;.....	7
4.2	Πολυδιαστατικότητα .....	9
4.2.1	Τέσσερις διαστάσεις και πάνω .....	11
4.2.2	Ένθετες Διαστάσεις .....	12
4.3	Ιεραρχίες και ομαδοποιήσεις .....	13
4.4	Λειτουργίες του OLAP.....	14
4.4.1	Τεμαχισμός και κομμάτιασμα (λειτουργίες Slicing και Dicing).....	14
4.4.2	Συναθροιστική άνοδος, Αναλυτική κάθοδος και Οριζόντια ανάλυση πρωτογενών στοιχείων(λειτουργίες Drill-down, Drill-up & Drill-across) .....	14
4.5	Συγκεντρωτικά δεδομένα.....	15
4.5.1	Προϋπολογισμένα εναντίον Κατά παραγγελία.....	16
4.5.2	Σποραδικά δεδομένα.....	16
4.6	Αποθήκευση των δεδομένων.....	18
4.6.1	ROLAP .....	18
4.6.2	MOLAP .....	18
4.6.3	HOLAP.....	18
4.6.4	Σύγκριση μεταξύ ROLAP, HOLAP και MOLAP.....	19

5	Το εργαλείο διαχείρισης δεδομένων Pentaho .....	20
6	Μελέτη περίπτωσης: Χρησιμοποιώντας OLAP σε βάση δεδομένων με αποτελέσματα αξιολόγησης.....	21
6.1	Τα δεδομένα αξιολόγησης .....	21
6.2	Εγκατάσταση OLAP (Pentaho) και εισαγωγή δεδομένων .....	22
6.3	Σχήμα OLAP .....	48
6.4	Παραδείγματα χρήσης OLAP .....	50
7	Συμπεράσματα και μελλοντικές κατευθύνσεις .....	62

# 1 Περίληψη

Η παρούσα πτυχιακή εργασία ασχολείται με την δημιουργία μιας υπηρεσίας OLAP(Onlineanalyticalprocessing) σε βάση δεδομένων και την αξιολόγηση των δεδομένων της. Το OLAP είναι μια μέθοδος με την οποία απαντάμε πολυδιάστατα ερωτήματα γρήγορα. Τα εργαλεία OLAP επιτρέπουν στους χρήστες να αναλύσουν πολύπλοκα δεδομένα από πολλές οπτικές γωνίες.

Ο σκοπός της εργασίας αυτής είναι η αποσαφήνιση των βασικών λειτουργιών του λογισμικού διαχείρισης βάσεων δεδομένων OLAP, καθώς και οι συγκριτική ανάλυση αυτών με τις σχεσιακές βάσεις δεδομένων, ώστε να έχουμε την βέλτιστη απόδοση αποτελεσμάτων κατά την εξαγωγή τους με το εν λόγω λογισμικό από την βάση δεδομένων του Ελληνικού Ανοικτού Πανεπιστημίου (ΕΑΠ) που έχουμε ως παράδειγμα.

Στα παρακάτω κεφάλαια θα δούμε τι είναι το OLAP και πως αυτό χρησιμοποιείται. Θα αναλύσουμε τις βασικές έννοιες του OLAP όπως τον κύβο και τις διαστάσεις. Έπειτα θα δούμε κάποιες βασικές λειτουργίες του OLAP ώστε να μάθουμε πως λειτουργεί.

Στη συνέχεια θα δούμε αναλυτικά όλα τα βήματα για την εγκατάσταση των απαραίτητων προγραμμάτων που επιλέξαμε για να μπορέσουμε να υλοποιήσουμε μια μελέτη περίπτωσης και τον τρόπο με τον οποίο θα στήσουμε και θα αξιολογήσουμε τη βάση δεδομένων. Τέλος θα δούμε κάποια παραδείγματα χρήσης του OLAP.

## 2 Ιστορική αναδρομή

Από όταν ξεκινήσαμε να μηχανογραφούμε δεδομένα για την διαχείριση τους, η δυνατότητα χρήσης ηλεκτρονικών υπολογιστών για την ανάλυση των δεδομένων ήταν εμφανής, για τις επιχειρήσεις. Ωστόσο, για να δημιουργηθεί κάποιο query για τα δεδομένα με τα πρώτα εργαλεία ανάλυσης χρειαζόταν η βοήθεια του τμήματος IT (Information Technology). Δεν ήταν καθόλου διαδραστικά και απαιτούσαν ειδικές γνώσεις πάνω στους υπολογιστές. Από τα μέσα της δεκαετίας του '80, τα συστήματα υποστήριξης διοίκησης εισήγαγαν νέα γραφικά περιβάλλοντα που δεν χρειάζονταν πληκτρολόγιο (όπως οθόνες αφής). Ωστόσο, τα στελέχη χρειάζονταν ακόμα επαγγελματίες πληροφορικής για τον ορισμό των queries, και οι τιμές του απαιτούμενου λογισμικού και υλικού, ήταν απαγορευτικές για τις μικρές επιχειρήσεις. Τελικά, τα φθηνότερα και εύκολα στη χρήση λογιστικά φύλλα έγιναν πολύ δημοφιλής στις επιχειρήσεις, αλλά σύντομα έγινε σαφές ότι δεν ήταν κατάλληλα για χρήση και διαχείριση τεράστιων ποσοτήτων δεδομένων. Έτσι, το 1993 ο Edgar Codd επινόησε τον όρο OLAP [1]. Στην εν λόγω έκθεση, οι συντάκτες ορίζουν 12 κανόνες που πρέπει να έχει ένα εργαλείο ώστε να θεωρείται εργαλείο OLAP. Οι κανόνες αυτοί προκάλεσαν έντονη αντιπαράθεση, και δεν πέτυχαν τη γενική αναγνώριση όπως η αντίστοιχη έκθεση για Συστήματα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων του Codd [1]. Παρ' όλα αυτά, το όνομα OLAP έγινε πολύ δημοφιλές και χρησιμοποιείται ευρέως.

Αν και το όνομα OLAP πρωτοεμφανίστηκε το 1993 και η ιδέα για αυτό υπήρχε από την δεκαετία του '80, δεν υπάρχει τυπικός ορισμός για την έννοια αυτή ακόμα. Όπως προτείνεται από τον Nigel Pendse στο [2] τα εργαλεία του OLAP θα πρέπει να περάσουν το FASMI (Fast Analysis of Shared Multidimensional Information) τεστ. Που σημαίνει γρήγορη ανάλυση των κοινόχρηστων πολυδιάστατων δεδομένων. Έτσι, τα εργαλεία θα πρέπει:

- Να είναι αρκετά γρήγορα για να επιτρέπουν διαδραστικά queries.
- Να βοηθούν στην ανάλυση, παρέχοντας ευελιξία στη χρήση των στατιστικών εργαλείων και σεναρίων What-if.
- Να παρέχουν μηχανισμούς ασφάλειας (τόσο με την έννοια της εμπιστευτικότητας όσο και της ακεραιότητας) που να επιτρέπουν την ανταλλαγή δεδομένων.
- Να παρέχουν μια πολυδιάστατη προβολή, έτσι ώστε η μεταφορά του κύβου δεδομένων να μπορεί να χρησιμοποιηθεί από τους χρήστες.
- Να είναι σε θέση να διαχειριστούν μεγάλο όγκο δεδομένων και μεταδεδομένων.

Ωστόσο, δεν υπάρχουν μέτρα και όρια για όλα αυτά τα χαρακτηριστικά, ώστε να είναι σε θέση να διαπιστωθεί αν ένα από αυτά έχει εκπληρωθεί ή όχι, και ως εκ τούτου μπορεί πάντα να αμφισβητηθεί αν ένα συγκεκριμένο εργαλείο τα ικανοποιεί. Παρ' όλα αυτά, είναι γενικά αποδεκτό ότι, προκειμένου να θεωρηθεί ένα εργαλείο, «εργαλείο OLAP», πρέπει να προσφέρει μια πολυδιάστατη προβολή των δεδομένων.

Από την αρχή της δημιουργίας τους, τα εργαλεία OLAP έχουν γίνει μικρότερα και έχουν μειώσει το κόστος, αλλά προσφέρουν περισσότερες λειτουργίες, καλύτερη επαφή με τον χρήστη και ευκολότερη διαχείριση. Έτσι, μπορεί κάποιος να υποθέσει πως σήμερα έχει έρθει η ώρα για τις μικρές επιχειρήσεις να χρησιμοποιήσουν το OLAP. Μπορούν πλέον να αντέξουν οικονομικά να χρησιμοποιήσουν τεχνολογίες OLAP. Μέρος της βιομηχανίας OLAP σχετίστηκε με το Συμβούλιο του OLAP (που δημιουργήθηκε τον Ιανουάριο του



1995), στόχος του οποίου ήταν η προώθηση και η τυποποίηση της ορολογίας και της τεχνολογίας του OLAP. Ωστόσο, ορισμένοι μεγάλοι πωλητές ποτέ δεν έγιναν μέλη του συμβουλίου αυτού, έτσι ώστε τελικά να εξαφανιστεί το 1999. Σήμερα, δεν υπάρχει τέτοιο τυποποιημένο ίδρυμα αφιερωμένο ειδικά στο OLAP. Ως εκ τούτου, φαίνεται δύσκολο να έχουμε ένα πρότυπο μοντέλο δεδομένων και μια γλώσσα για queries στο εγγύς μέλλον, παρά το γεγονός ότι είναι σαφώς επιθυμητό.

### 3 Το OLAP αναλυτικά

Αυτό το κεφάλαιο εξηγεί το OLAP. Γιατί υπάρχει, και γιατί χρησιμοποιείται αντί των παραδοσιακών μεθόδων αποθήκευσης και ανάλυσης δεδομένων.

#### 3.1 Τι είναι το OLAP;

OLAP σημαίνει πολλά διαφορετικά πράγματα σε διαφορετικούς ανθρώπους, αλλά οι ορισμοί συνήθως περιλαμβάνουν τους όρους «κύβου», «πολυδιάστατο», «Slice and Dice» και «Ταχεία απόκριση». Το OLAP είναι όλα αυτά τα πράγματα και ακόμα περισσότερα. Είναι επίσης ένας παρεξηγημένος όρος, εν μέρει επειδή καλύπτει τόσα πολλά θέματα. Για αρχή, θα εξηγήσουμε το ορισμό και την προέλευση του OLAP.

Το OLAP είναι ένα ακρώνυμο για το "On-Line Analytical Processing" (Διαδικτυακή επεξεργασία αναλύσεων) [3]. Αυτό, μόνο του, δεν παρέχει πολύ ακριβή περιγραφή του OLAP, αλλά το ξεχωρίζει από το OLTP ή "On-Line Transactional Processing" (Διαδικτυακή επεξεργασία συναλλαγών).

Ο όρος **OLTP** καλύπτει, όπως υποδηλώνει το όνομά του, τις εφαρμογές που λειτουργούν με συναλλαγές ή «ατομικά» δεδομένα. Τα μεμονωμένα αρχεία που περιέχονται σε μια βάση δεδομένων. Οι εφαρμογές **OLTP** συνήθως μόνο ανακτούν ομάδες αρχείων και τις παρουσιάζουν στον τελικό χρήστη, για παράδειγμα, ο κατάλογος των λογισμικών που πωλούνται σε ένα συγκεκριμένο κατάστημα κατά τη διάρκεια μιας ημέρας. Αυτές οι εφαρμογές χρησιμοποιούν συνήθως σχεσιακές βάσεις δεδομένων, με ένα γεγονός ή τα δεδομένα του πίνακα που περιέχουν μεμονωμένες συναλλαγές που συνδέονται με τους πίνακες που περιέχουν metadata με δεδομένα σχετικά με τους πελάτες και τις λεπτομέρειες του προϊόντος.

Οι εφαρμογές **OLAP** προσφέρουν στον τελικό χρήστη όχι μόνο δεδομένα, αλλά και πληροφορίες. Κάνουν εύκολο στους χρήστες να εντοπίζουν μοτίβα ή τάσεις στα δεδομένα πολύ γρήγορα, χωρίς την ανάγκη για αναζήτηση μέσα από βουνά «ακατέργαστων» δεδομένων. Συνήθως η ανάλυση αυτή καθοδηγείται από την ανάγκη να απαντήσουμε σε ερωτήσεις των επιχειρήσεων, όπως «Πώς είναι οι πωλήσεις μας αυτό το μήνα στη Βόρεια Ελλάδα;». Από εκεί, οι εφαρμογές OLAP προχώρησαν σε τομείς όπως η πρόβλεψη και η εξόρυξη δεδομένων, επιτρέποντας στους χρήστες να απαντήσουν σε ερωτήματα όπως «Ποιο είναι το προβλεπόμενο κόστος για την επόμενη χρονιά;» και «Δείξε μου τον πιο επιτυχημένο πωλητή μας».

Οι εφαρμογές **OLAP** διαφέρουν από τις εφαρμογές **OLTP** με τον τρόπο που αποθηκεύουν τα δεδομένα, τον τρόπο που αναλύουν τα δεδομένα και τον τρόπο που παρουσιάζουν τα στοιχεία στον τελικό χρήστη. Είναι αυτές οι θεμελιώδεις διαφορές (όπως παρουσιάζοντα στο [4]) που επιτρέπουν στις εφαρμογές OLAP να απαντήσουν σε πιο εξελιγμένες ερωτήσεις των επιχειρήσεων.

## 3.2 Γιατί χρειαζόμαστε το OLAP;

Με μια πρώτη ματιά το OLAP δεν φαίνεται τόσο χρήσιμο διότι ότι πληροφορίες μπορεί να χρειαστούμε σε μια επιχείρηση, παρέχονται και από τα δεδομένα συναλλαγών. Αυτό σημαίνει ότι κάθε απαίτηση καλύπτεται από τις εφαρμογές OLTP. Αυτό βέβαια δεν είναι αλήθεια διότι οι εφαρμογές OLTP έχουν κάποιους περιορισμούς. Τα όρια των εφαρμογών OLTP βρίσκονται σε τρεις τομείς.

### 3.2.1 Αύξηση των αποθηκευμένων δεδομένων

Οι επιχειρήσεις αποθηκεύουν όλο και περισσότερο δεδομένα, και η ανάκτηση αυτών των πολλών χιλιάδων δεδομένων για άμεση ανάλυση από πολλούς χρήστες ταυτόχρονα είναι δαπανηρή διαδικασία. Οι περισσότερες μηχανές βάσεων δεδομένων δυσκολεύονται όταν καλούνται να επιστέψουν μεγάλα αποτελέσματα για να απαντήσουν σε queries σε χιλιάδες χρήστες ταυτόχρονα.

Το caching συχνά αναζητά δεδομένα σε προσωρινούς πίνακες και η αποθήκευση των δεδομένων αυτών μπορεί να βοηθήσει λίγο την κατάσταση, αλλά λύνει το πρόβλημα μόνο μέχρι ένα σημείο, ιδίως εάν κάθε χρήστης απαιτεί ένα ελαφρώς διαφορετικό σύνολο δεδομένων.

Σε μια σύγχρονη βάση δεδομένων όπου τα απαιτούμενα στοιχεία μπορεί να εξαπλώνονται σε πολλούς πίνακες ένα πολύπλοκο query απαιτεί περισσότερους πόρους συστήματος και προκαλεί καθυστερήσεις στο σύστημα. Έτσι πρέπει να δαπανηθούν περισσότερα χρήματα σε servers για τις βάσεις δεδομένων ώστε να καλυφθούν οι απαιτήσεις του χρήστη.

### 3.2.2 Δεδομένα εναντίον Πληροφοριών

Οι χρήστες στις επιχειρήσεις για να πάρουν τις κατάλληλες αποφάσεις χρειάζονται τις πληροφορίες που κρύβονται μέσα στα δεδομένα που τους παρέχει η επιχείρηση. Ένας υπεύθυνος αποθεμάτων σε ένα υπερκατάστημα μπορεί να θέλει τον πλήρη κατάλογο όλων των αγαθών που πωλούνται προκειμένου να ελέγξει τα επίπεδα των αποθεμάτων, αλλά ο διαχειριστής μπορεί να θέλει να γνωρίζει μόνο την ποσότητα των φρούτων και κατεψυγμένων προϊόντων που πωλούνται. Ή ακόμα καλύτερα η τάση των πωλήσεων των κατεψυγμένων προϊόντων τους τελευταίους τρεις μήνες.

Για να απαντήσουμε στο ερώτημα «Πόσα κατεψυγμένα προϊόντα πουλήσαμε σήμερα;», μια εφαρμογή OLTP πρέπει να ανακτήσει όλες τις πωλήσεις των κατεψυγμένων προϊόντων για την ημέρα και στη συνέχεια, να τις μετρήσει, παρουσιάζοντας μόνο τις συνοπτικές πληροφορίες στον τελικό χρήστη. Για να κάνουμε μια σύγκριση για τρεις μήνες, η διαδικασία αυτή πρέπει να επαναληφθεί για πολλές ημέρες. Αν πολλαπλασιάσουμε το πρόβλημα με αρκετές εκατοντάδες καταστήματα, έτσι ώστε ο διευθύνων σύμβουλος να δει πώς το σύνολο της εταιρείας λειτουργεί, είναι εύκολο να δούμε ότι το πρόβλημα απαιτεί σημαντικές ποσότητες επεξεργαστικής ισχύς για να δώσει απαντήσεις στα λίγα δευτερόλεπτα που ο χρήστης θα ήταν διατεθειμένος να περιμένει.

Οι μηχανές των βάσεων δεδομένων δεν μπορούν να αποδώσουν καλά όταν τους ζητηθεί να κάνουν κάτι τέτοιο. Μια εφαρμογή OLTP θα είναι πάντα σε θέση να παρέχει τις απαντήσεις, αλλά όχι στα τυπικά λίγα δευτερόλεπτα απόκρισης που απαιτούνται από τους χρήστες.

Ούτε και εδώ βοηθά το caching γιατί θα χρειαστεί να αποθηκευτεί στην προσωρινή μνήμη μια πολύ μεγάλη ποσότητα δεδομένων ώστε να γίνει αυτό.

### 3.2.3 Διάταξη των δεδομένων

Το σχεσιακό μοντέλο βάσης δεδομένων έχει σχεδιαστεί για επεξεργασία συναλλαγών και δεν είναι πάντα ο καλύτερος τρόπος για να αποθηκεύσουμε τα δεδομένα όταν προσπαθούμε να απαντήσουμε ερωτήσεις όπως «Οι πωλήσεις ηλεκτρονικών υπολογιστών ανά περιοχή» ή «Ο όγκος των συναλλαγών με πιστωτικές κάρτες ανά μήνα.». Αυτοί οι τύποι ερωτημάτων απαιτούν τεράστιες ποσότητες δεδομένων για να ανακτηθούν και να συγκεντρωθούν κάθε φορά που κάποιος τα ζητά, κάτι που απαιτεί χρόνο και πόρους συστήματος για να επιτευχθεί.

Αναλυτικότερα, ερωτήματα όπως «πωλήσεις κατεψυγμένων προϊόντων ανά περιοχή» και «Περιφέρειες διαχωρισμένες από τις πωλήσεις των προϊόντων» απαιτούν ξεχωριστά ερωτήματα για να πραγματοποιηθούν στο ίδιο σύνολο δεδομένων. Η απάντηση στους περιορισμούς του OLTP δεν είναι να δαπανούμε περισσότερα και περισσότερα χρήματα για την μεγαλύτερη και πιο γρήγορη βάση δεδομένων, αλλά να χρησιμοποιήσουμε μια διαφορετική προσέγγιση συνολικά για το πρόβλημα και αυτή η προσέγγιση είναι το OLAP.

Οι εφαρμογές OLAP αποθηκεύουν τα δεδομένα με διαφορετικό τρόπο από το παραδοσιακό μοντέλο σχέσεων, που τους επιτρέπει να εργάζονται με σύνολα δεδομένων που έχουν σχεδιαστεί για να εξυπηρετούν μεγαλύτερο αριθμό χρηστών παράλληλα. Σε αντίθεση με τις βάσεις δεδομένων, τα δεδομένα του OLAP έχουν σχεδιαστεί για να λειτουργούν με συγκεντρωτικά στοιχεία, που τους επιτρέπει να απαντήσουν γρήγορα σε ερωτήσεις υψηλού επιπέδου σχετικά με τα δεδομένα μιας εταιρείας, ενώ εξακολουθεί να επιτρέπει στους χρήστες να έχουν πρόσβαση στα αρχικά δεδομένα συναλλαγών, όταν απαιτείται.

## 4 Βασικές αρχές του OLAP

Οι εφαρμογές OLAP που χρησιμοποιούνται για την επίλυση των καθημερινών επιχειρηματικών queries όπως «Πόσα αυτοκίνητα πουλήσαμε στην Ευρώπη τον περασμένο μήνα;» ή «Πετάνε τα καταστήματα μας στην βόρεια Ελλάδα περισσότερα ελαττωματικά προϊόντα έτος με το έτος;» Για να απαντήσουμε αυτά τα ερωτήματα, πρέπει να ανακτηθούν μεγάλα τμήματα των συναλλαγών ή των δεδομένων και στη συνέχεια να γίνουν οι κατάλληλες πράξεις. Ουσιαστικά, απαιτούν μια διαφορετική προσέγγιση για την αποθήκευση και την ανάκτηση των δεδομένων.

Το κάθε εργαλείο OLAP χρησιμοποιεί διαφορετικές τεχνολογίες αλλά όλα παρουσιάζουν τα δεδομένα χρησιμοποιώντας την ίδια βασική έννοια. Αυτή του πολυδιάστατου κύβου. Οι κύβοι αυτοί μπορεί να φαίνονται αρκετά περίπλοκοι διότι διαφέρουν από τις βάσεις δεδομένων αλλά στην πραγματικότητα είναι εύκολο να κατανοηθούν. Παρακάτω αναλύουμε τι είναι κύβος και ποιες είναι οι θεμελιώδεις διαφορές του από τις σχεσιακές βάσεις δεδομένων. Η αποθήκευση δεδομένων σε κύβους εισάγει νέους όρους και έννοιες [5].

### 4.1 Τι είναι ένας κύβος;

Ο κύβος είναι ο προκαταρκτικός σχεδιασμός για την αποθήκευση δεδομένων στο κέντρο όλων των εφαρμογών OLAP. Αν και τα βασικά δεδομένα θα μπορούσαν να αποθηκευτούν χρησιμοποιώντας πολλές διαφορετικές μεθόδους, ο κύβος είναι ο λογικός σχεδιασμός, με τον οποία γίνεται αναφορά στα δεδομένα.

Ο ευκολότερος τρόπος για να εξηγήσουμε έναν κύβο είναι να συγκρίνουμε την αποθήκευση δεδομένων σε ένα κύβο με την αποθήκευσή σε έναν πίνακα βάσης δεδομένων [6].

Products	Store	Volume
Bulbs	Uptown	40
Bulbs	Midtown	52
Bulbs	Downtown	36
Batteries	Uptown	104
Batteries	Midtown	22
Batteries	Downtown	78
Fuses	Uptown	56
Fuses	Midtown	31
Fuses	Downtown	58

Σχ 4.1.1 Ένας σχεσιακός πίνακας που περιέχει τα αρχεία πωλήσεων.

Το σχήμα 4.1.1 παρουσιάζει μια σειρά από αρχεία πωλήσεων από τρία καταστήματα ηλεκτρικών που εμφανίζονται σε πίνακα μιας βάσης δεδομένων. Υπάρχουν δύο στήλες πεδίων "Product" και "Store", που περιέχουν πληροφορίες σε μορφή κειμένου για κάθε εγγραφή δεδομένων και μια τρίτη στήλη τιμών "Volume". Αυτό το είδος της διάταξης του πίνακα συχνά αποκαλείται «πίνακας γεγονόσ».

Οι στήλες σε ένα πίνακα καθορίζουν τα δεδομένα που αποθηκεύονται στον πίνακα. Οι σειρές των πληροφοριών και οι αριθμητικές τιμές είναι απλά παραδείγματα των δεδομένων.

Κάθε σειρά είναι ένα μόνο σημείο δεδομένων. Ένα μεγαλύτερο σύνολο δεδομένων θα εμφανιζόταν ως ένας πίνακας με ένα μεγαλύτερο αριθμό γραμμών.

*Διάσταση Products*

→

↓

*Διάσταση Store*

Store	Bulbs	Batteries	Fuses
Uptown	40	104	56
Midtown	52	22	31
Downtown	36	78	58

Σχ. 4.1.2 Τα δεδομένα από το σχήμα 4.1.1 ως ένας διδιάστατος κύβος.

Το σχήμα 4.1.2 παρουσιάζει τα ίδια δεδομένα τώρα τοποθετημένα σε ένα «κύβο». Ο όρος «κύβος» είναι κάπως γενικός, καθώς στην πραγματικότητα αυτό είναι μια διδιάστατη διάταξη, που συχνά αναφέρεται ως μια «προβολή φύλλου» διότι μοιάζει με ένα τυπικό υπολογιστικό φύλλο.

Οι άξονες του κύβου περιέχουν τις στήλες του πίνακα της βάσης δεδομένων. Κάθε άξονας σε ένα κύβο αναφέρεται ως μια «διάσταση». Σε αυτόν τον κύβο, η οριζόντια διάσταση περιέχει τα ονόματα των προϊόντων και αναφέρεται ως η "Διάσταση Product». Η κάθετη διάσταση περιέχει τα ονόματα των καταστημάτων και αναφέρεται ως «Διάσταση Store».

Στον πίνακα της βάσης δεδομένων, μια ενιαία γραμμή αντιπροσωπεύει ένα μόνο σημείο δεδομένων. Στον κύβο η τομή μεταξύ των πεδίων καθορίζει ένα σημείο δεδομένων. Σε αυτόν τον κύβο, το κελί στην διασταύρωση των Fuses και Midtown περιέχει τον αριθμό των Fuses που πουλήθηκαν στο κατάστημα Midtown(σε αυτή την περίπτωση, 31 κουτιά). Δεν υπάρχει καμία ανάγκη να αναφέρουμε "Τιμές", καθώς ολόκληρος ο κύβος περιέχει δεδομένα τιμών.

Αυτή η ιδέα συλλογής δεδομένων που βασίζεται στον συνδυασμό διαστάσεων είναι ο λόγος για τον οποίο δεν μπορούμε απλά να αγνοούμε μία από τις διαστάσεις του κύβου. Για παράδειγμα, η ερώτηση «Πόσα bulbs πουλήσαμε;» δεν έχει νόημα σε αυτόν τον κύβο, εκτός αν προσδιορίσουμε ζητώντας δεδομένα για ένα συγκεκριμένο κατάστημα.

Ο όρος "πεδίο" χρησιμοποιείται για να αναφέρουμε μεμονωμένα μέλη μιας διάστασης, έτσι για παράδειγμα, Uptown είναι ένα πεδίο στη διάσταση Store. Παρατηρούμε ότι οι δύο διαστάσεις περιέχουν φαινομενικά άσχετα πεδία. Οι διαστάσεις συνήθως αποτελούνται από αντικείμενα της ίδιας κατηγορίας. Σε αυτό το παράδειγμα όλα τα προϊόντα είναι σε μια διάσταση και όλα τα καταστήματα είναι σε μια άλλη. Το να προσπαθήσουμε να αναμειξουμε πεδία μεταξύ των δύο διαστάσεων δεν θα είχε νόημα, διότι δεν θα ήταν δυνατό να δημιουργηθεί ένα μοναδικό κελί για κάθε σημείο δεδομένων.

Να σημειωθεί ότι αποφύγαμε τη χρήση του όρου σειράς και στήλης. Αν και ένας κύβος φαίνεται να έχει σειρές και στήλες ακριβώς όπως ένας πίνακας, είναι πολύ διαφορετικές από τις σειρές και τις στήλες σε μια βάση δεδομένων. Σε μια βάση δεδομένων, η σειρά και η στήλη αναφέρονται σε συγκεκριμένα στοιχεία. Σε ένα κύβο, απλά περιγράφουν τον τρόπο με τον οποίο ο κύβος παρουσιάζει τα δεδομένα. Για παράδειγμα, ο κύβος στο σχήμα 4.1.2 μπορεί επίσης να απεικονιστεί όπως στο σχήμα 4.1.3, με τις διαστάσεις αντεστραμμένες.

*Διάσταση Store*

→

<i>Διάσταση Products</i>	Products	Uptown	Midtown	Downtown
	Bulbs	40	52	36
	Batteries	104	22	78
	Fuses	56	31	58

↓

Σχ. 4.1.3 Οι δύο διαστάσεις του κύβου αντεστραμμένες.

Και τα δύο σχήματα (4.1.2 και 4.1.3) είναι σωστές διατάξεις. Το σημαντικό σημείο είναι ότι το πρώτο σχήμα δείχνει «Προϊόντα ανά κατάσταση» και το δεύτερο δείχνει «Καταστάματα ανά Προϊόν». Αυτό είναι ένα από τα πλεονεκτήματα του κύβου ως μέσο αποθήκευσης δεδομένων. Τα δεδομένα μπορούν να αναδιατάσσονται γρήγορα για να απαντήσουμε σε πολλές ερωτήσεις μια επιχείρησης, χωρίς να χρειάζεται να γίνουν νέοι υπολογισμοί. Ένα δεύτερο πλεονέκτημα είναι ότι τα δεδομένα θα μπορούσαν να είναι ταξινομημένα είτε κάθετα είτε οριζόντια, επιτρέποντας τα δεδομένα να διευθετηθούν από το κατάστημα ή το προϊόν ανεξάρτητα από τον προσανατολισμό του κύβου.

Από αυτόν τον απλό δισδιάστατο κύβο, μπορούμε τώρα να εξηγήσουμε κάποιες περαιτέρω έννοιες.

## 4.2 Πολυδιαστατικότητα

Στην προηγούμενη ενότητα, εξετάσαμε έναν απλό δισδιάστατο κύβο και είδαμε ότι δεν μας προσφέρει κάτι παραπάνω από μια απλή βάση δεδομένων, παρά μόνο έναν διαφορετικό τρόπο παρουσίασης των δεδομένων. Οι δυνατότητες ενός κύβου φαίνονται καλύτερα όταν επεκτείνουμε το σχεδιασμό σε περισσότερες διαστάσεις. Η πολυδιαστατικότητα είναι ίσως το πιο δύσκολο στοιχείο του σχεδιασμού ενός κύβου καθώς μερικές φορές είναι δύσκολο να προβλεφθεί. Είναι καλύτερο να το εξηγήσουμε αρχίζοντας με ένα τρισδιάστατο παράδειγμα.

Μένοντας στο σύνολο δεδομένων που χρησιμοποιήσαμε στην προηγούμενη ενότητα, μπορούμε τώρα να φέρουμε περισσότερα δεδομένα, όπως τα έσοδα και κόστος. Τα σχήματα 4.2.1 & 4.2.2 δείχνουν τους διαφορετικούς τρόπους με τους οποίους τα νέα δεδομένα θα μπορούσαν να αποθηκευτούν σε έναν πίνακα.

Οι μετρήσεις απλώνονται σε πολλαπλές στήλες

→

Products	Store	Revenue	Cost	Volume
Bulbs	Uptown	\$250	\$115	40
Bulbs	Midtown	\$325	\$145	52
Bulbs	Downtown	\$225	\$155	36
Batteries	Uptown	\$416	\$255	104
Batteries	Midtown	\$88	\$55	22
Batteries	Downtown	\$312	\$180	78

Σχ 4.2.1 Μια «εκφυλισμένη» μορφή πίνακα

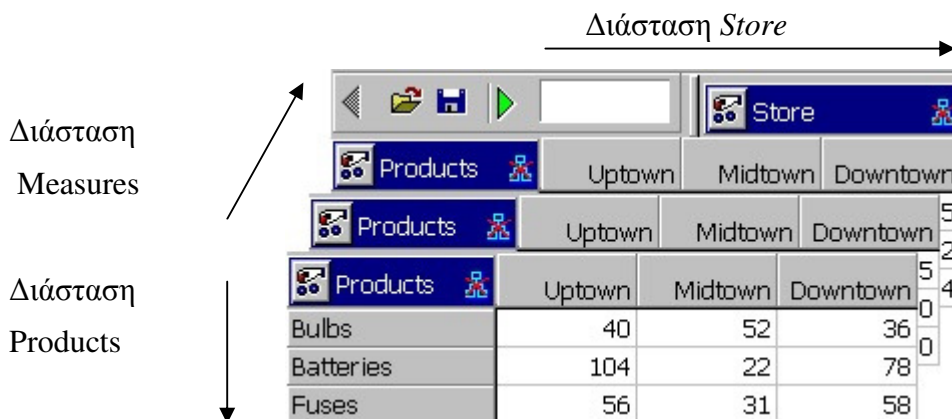
*Όλες οι τιμές σε μία μόνο στήλη*

Products	Store	Measures	Value
Bulbs	Uptown	Revenue	\$250
Bulbs	Uptown	Cost	\$115
Bulbs	Uptown	Volume	40
Bulbs	Midtown	Revenue	\$325
Bulbs	Midtown	Cost	\$145
Bulbs	Midtown	Volume	52

Σχ 4.2.2 Η "κανονική" διάταξη του πίνακα

Όπως φαίνεται, η «εκφυλισμένη» διάταξη οδηγεί σε ένα μεγαλύτερο πίνακα με λιγότερες σειρές ενώ το κανονικό μοντέλο οδηγεί σε έναν μικρότερο πίνακα με περισσότερες σειρές. Καμία από τις δύο διατάξεις δεν είναι ιδιαίτερα εύκολο να διαβαστεί όταν προβάλλονται άμεσα.

Η απλούστερη διάταξη OLAP είναι να δημιουργηθούν τρεις ξεχωριστοί, δύο διαστάσεων κύβοι για κάθε τμήμα των δεδομένων. Ένας για τις τιμές των εσόδων, ένας για το κόστος και ένας για τα πλήθη. Ενώ είναι χρήσιμη, αυτή η διάταξη χάνει τα πλεονεκτήματα που έχουμε συνδυάζοντας τα δεδομένα σε ένα τρισδιάστατο κύβο. Ο τρισδιάστατος κύβος φτιάχνεται πολύ απλά αν τοποθετήσουμε τα τρία δύο διαστάσεων "φύλλα" του πλήθους, του κόστους και των εσόδων (Volume, Cost, Revenue) το ένα πάνω από το άλλο.



Σχ. 4.2.3 Ο τρισδιάστατος κύβος.

Όπως φαίνεται από το σχήμα 4.2.3, η τρισδιάστατη διάταξη γίνεται εμφανής αμέσως μόλις τα τρία "φύλλα" τοποθετηθούν το ένα πάνω στο άλλο. Η τρίτη διάσταση, «μετρήσεις» (Measures) είναι ορατή ως ο τρίτος άξονας του κύβου, με κάθε φύλλο που αντιστοιχεί στο σχετικό πεδίο (πλήθους, κόστους ή εσόδων).



Τα πραγματικά σημεία δεδομένων βρίσκονται πάλι με συνδυασμό των διαστάσεων. Σε αυτό το παράδειγμα, κάθε κελί είναι μια τιμή για τα έσοδα, το κόστος ή ο πλήθος ενός συγκεκριμένου προϊόντος που πωλείται σε ένα συγκεκριμένο κατάστημα. Όπως και πριν, τα δεδομένα μπορούν να επαναπροσανατολιστούν και να αναδιαταχθούν, αλλά αυτή τη φορά, πιο εξελιγμένα. Για παράδειγμα, η προβολή από τη μπροστά πλευρά του κύβου στην εικόνα 3.6 παρουσιάζει τα έσοδα, το κόστος και τον πλήθος για όλα τα προϊόντα που πωλούνται στο κατάστημα στο κέντρο της πόλης. Η προβολή από πάνω δείχνει τα στοιχεία για τα έσοδα, το κόστος και το πλήθος για τις λάμπες (Bulbs) σε όλες τις τρεις αποθήκες.

Αυτή η ικανότητα να προβάλλουμε διαφορετικές όψεις ενός κύβου επιτρέπει στην επιχείρησή ερωτήματα όπως «Το προϊόν με τις καλύτερες επιδόσεις σε όλα τα καταστήματα» να μπορούν να απαντηθούν γρήγορα αλλάζοντας τη διάταξη των δεδομένων χωρίς να κάνουμε οποιουδήποτε νέους υπολογισμούς. Καταλήγοντας έτσι, σε μια σημαντική βελτίωση των επιδόσεων κατά της μεθόδου του παραδοσιακού πίνακα σε μια σχεσιακή βάση δεδομένων.

#### 4.2.1 Τέσσερις διαστάσεις και πάνω

Η λέξη «κύβος» μας κάνει να σκαφτόμαστε ένα τρισδιάστατο αντικείμενο, όμως στο OLAP ο κύβος δεν περιορίζεται σε 3 διαστάσεις. Πολλές εφαρμογές OLAP χρησιμοποιούν σχεδιαγράμματα κύβου που περιέχουν έως και δέκα διαστάσεις, αλλά το να προσπαθήσουμε να απεικονίσουμε ένα πολυδιάστατο κύβο είναι πολύ δύσκολο. Το πρώτο βήμα είναι να κατανοήσουμε γιατί το να δημιουργήσουμε έναν κύβο με περισσότερες από τρεις διαστάσεις είναι εφικτό και τι πλεονεκτήματα έχει.

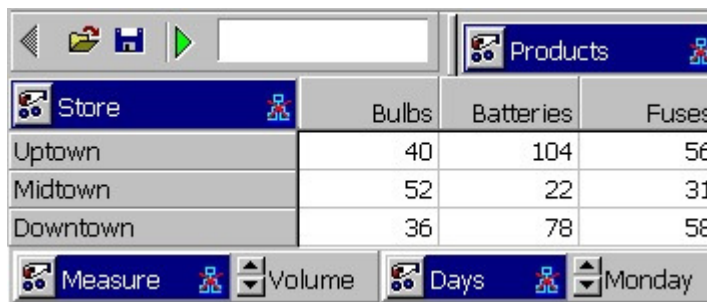
Όπως είδαμε στην προηγούμενη ενότητα, το να δημιουργήσουμε ένα τρισδιάστατο κύβο είναι αρκετά απλό, ιδιαίτερα αφού είχαμε ένα σύνολο στοιχείων που μπορούσε να παρουσιαστεί σε μια διάταξη τριών διαστάσεων. Τώρα φανταστείτε ότι έχουμε αρκετούς τρισδιάστατους κύβους, τον καθένα να περιέχει τις ίδιες διαστάσεις προϊόντος, καταστήματος και μετρήσεων (Product, Store, Measures) όπως και πριν, αλλά με το καθένα να έχει δεδομένα για μια διαφορετική μέρα. Θα μπορούσαμε να επεκτείνουμε μία από τις διαστάσεις, για παράδειγμα, η διάσταση των μετρήσεων θα μπορούσε να έχει τα πεδία "κόστος της Δευτέρας" και "κόστος της Τρίτης», αλλά δεν θα ήταν εύκολο να εργαστούμε σε αυτόν το σχεδιασμό και θα χάναμε τα πλεονεκτήματα μιας πολυδιάστατης διάταξης. Η καλύτερη λύση είναι να δημιουργούμε μια τέταρτη διάσταση. Στην περίπτωση αυτή η διάσταση «Ημέρες» (Days) και να την προσθέσουμε στον κύβο. Δεν είναι εύκολο να προβληθεί ένας τέτοιος κύβος, αλλά μπορούμε εύκολα να δούμε τη χρησιμότητά του. Όπως αναφέρθηκε προηγουμένως, κάθε σημείο δεδομένων είναι αποθηκευμένο σε ένα μόνο κελί το οποίο μπορεί να αναφέρεται με μοναδικό τρόπο. Σε έναν σχεδιασμό τεσσάρων διαστάσεων, μπορούμε ακόμα να αναφερθούμε σε μια συγκεκριμένη τιμή, για παράδειγμα, η τιμή για τα έσοδα από λάμπες που πουλήθηκαν στο κατάστημα Uptown τη Δευτέρα. Αυτό είναι ένα σημείο αναφοράς τεσσάρων διαστάσεων καθώς απαιτεί ένα πεδίο από τέσσερις διαστάσεις για να το εντοπίσουμε:

1. Το πεδίο Έσοδα από τη διάσταση "Measures".
2. Το πεδίο λάμπες (Bulbs) από τη διάσταση "Product".
3. Το Uptown πεδίο από τη διάσταση "Store".

#### 4. Το πεδίο της Δευτέρας από τη διάσταση "Days".

Χωρίς στην πραγματικότητα να χρειάζεται να σχεδιάσουμε ή να απεικονίσουμε το σύνολο του κύβου, είναι αρκετά εύκολο να ανακτήσουμε και να δουλέψουμε με ένα σύνολο δεδομένων με τέσσερις διαστάσεις αν σκεφτούμε τα στοιχεία που ζητούνται για ένα συγκεκριμένο κελί. Βέβαια παρά το γεγονός ότι ο κύβος μπορεί να έχει τρεις (ή περισσότερες) διαστάσεις, οι περισσότερες εφαρμογές παρουσιάζουν μόνο μια δισδιάστατη προβολή των δεδομένων τους. Για να δούμε αυτό το σύνολο δεδομένων σε μόνο δύο διαστάσεις, οι άλλες διαστάσεις θα πρέπει να «μειωθούν». Αυτή είναι μια διαδικασία παρόμοια με την έννοια του φιλτραρίσματος όταν δημιουργούμε ένα SQL query.

Έχοντας σχεδιάσει ένα τεσσάρων διαστάσεων κύβο, ένας χρήστης μπορεί να θέλει μόνο να δει την αρχική δισδιάστατη διάταξη από το σχήμα 4.1.2. Προϊόντα ανά Κατάστημα. Για να εμφανίσουμε αυτήν την προβολή, πρέπει να κάνουμε κάτι για τις υπόλοιπες δύο διαστάσεις Measure και Days. Δεν έχει νόημα απλά να τις αποβάλουμε, καθώς χρησιμοποιούνται για να εντοπίσουμε τα δεδομένα. Απλά επιλέγουμε ένα πεδίο από την κάθε μια διάσταση, που μας επιτρέπει να παρουσιάσουμε μία ενιαία όψη του κύβου.



Store	Bulbs	Batteries	Fuses
Uptown	40	104	56
Midtown	52	22	31
Downtown	36	78	58

Measure: Volume Days: Monday

Βλεπουμε μονο το πλήθος \_\_\_\_\_ ↑  
για την ημέρα Δευτέρα ↑

Σχ 4.2.4 Δισδιάστατη όψη ενός τεσσάρων διαστάσεων δομή.

Ένας πίνακας συχνά περιγράφεται με τον αριθμό των στηλών και των σειρών που έχει, ενώ ένας κύβος ορίζεται από τον αριθμό των διαστάσεων και τον αριθμό των πεδίων σε κάθε διάσταση. Η παρουσίαση όμως κύβων με πάνω από τρεις διαστάσεις είναι κάπως δύσκολη αλλά συνήθως παρουσιάζονται με σχήμα αστεριού [7].

#### 4.2.2 Ένθετες Διαστάσεις

Παρά το γεγονός ότι τα δεδομένα μπορούν να προβληθούν μόνο σε μια δισδιάστατη ή επίπεδη διάταξη, δεν σημαίνει ότι μπορούν να εμφανίζονται μόνο δύο διαστάσεις ταυτόχρονα. Μπορούμε να εμφανίσουμε περισσότερες από μία διαστάσεις σε κάθε άξονα.

Για παράδειγμα, ο χρήστης μπορεί να θέλει να δει τα στοιχεία εσόδων και του κόστους για όλα τα προϊόντα που πωλούνται σε κάθε κατάσταση, την ημέρα Δευτέρα. Αντί να εμφανίζει τα έσοδα και το κόστος χωριστά, η διάσταση των μετρήσεων (Measure) μπορεί να «φωλιάζει» στο εσωτερικό της διάστασης των καταστημάτων (Store), εμφανίζοντας τα

έσοδα και το κόστος ώστε να μπορούμε να τα συγκρίνουμε άμεσα. Αυτή η διάταξη μπορεί να φανεί στο σχήμα 4.2.5

		Products		
Store	Measure	Bulbs	Batteries	Fuses
Uptown	Revenue	250	416	160
	Cost	115	255	95
Midtown	Revenue	325	88	140
	Cost	145	55	80
Downtown	Revenue	225	312	90
	Cost	155	180	75

Days: Monday

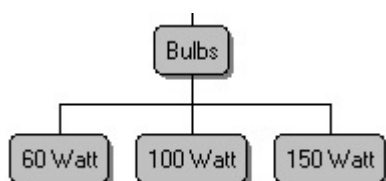
Σχ. 4.2.5 Δισδιάστατη άποψη με ένθετες διαστάσεις.

### 4.3 Ιεραρχίες και ομαδοποιήσεις

Μέχρι στιγμής, έχουμε δει μόνο απλές διαστάσεις, που η καθεμιά περιέχει μόνο μερικά πεδία. Τα πραγματικά σύνολα δεδομένων μπορούν να δημιουργήσουν διαστάσεις με πολλές χιλιάδες πεδία και αυτό δημιουργεί δύο προβλήματα. Πρώτον, οι διαστάσεις με μεγάλο αριθμό πεδίων δεν είναι εύκολο να διαχωριστούν και οι εφαρμογές απαιτούν έναν τρόπο διαίρεσης των πεδίων σε ομάδες. Και δεύτερον, πολλοί χρήστες δεν θέλουν τις ξεχωριστές τιμές για κάθε πεδίο αλλά την συνολική αξία όλων των πεδίων, ή για μια συγκεκριμένη ομάδα. Και τα δύο αυτά προβλήματα όμως λύνονται με την ομαδοποίηση των πεδίων ώστε να δημιουργήσουμε «ιεραρχικές» διαστάσεις.

Τις αποκαλούμε ιεραρχικές διαστάσεις διότι ορισμένα πεδία είναι σε υψηλότερο επίπεδο και κάποια σε χαμηλότερο, σε μια διάσταση. Για παράδειγμα, ένα κατάστημα μπορεί να πουλά τριών ειδών λάμπες (Bulbs), 60 Watt, 100 Watt και 150 Watt. Αντί για να έχει πάντα μια λίστα δεδομένων για κάθε τύπο λάμπας, ένας χρήστης ίσως να θέλει να ομαδοποιήσει τις λάμπες μαζί για να απλοποιήσει την προβολή δεδομένων και έτσι να μπορεί να απαντήσει σε ερωτήσεις όπως «Πόσες λάμπες (Bulbs) πουλήσαμε σήμερα?».

Η μέθοδος που χρησιμοποιείται από όλες τις εφαρμογές OLAP είναι να δημιουργηθεί ένα «πεδίο-γονέας» που ομαδοποιεί μαζί τα άλλα πεδία. Για παράδειγμα, το Σχήμα 4.2.6 δείχνει το πεδίο-γονέας " Bulbs" (λάμπες) και τα πεδία-παιδιά που το συνθέτουν. Αυτό το πεδίο-γονέας είναι σε υψηλότερο επίπεδο στη διάσταση και συχνά περιγράφεται ως το υψηλότερο ιεραρχικό επίπεδο.



Σχ. 4.2.6 Η ιεραρχία των Bulbs μέσα στην διάσταση Προϊόντων.

Μόλις έχει οριστεί το πεδίο-γονέας, αυτό κάνει αυτόματα τα πεδία κάτω από αυτό πεδία-παιδιά. Αν τα παιδιά του πεδίου-γονέα δεν έχουν κανένα δικό τους πεδίο-παιδί, τότε τα δεδομένα θα αναφέρονται ως πεδία βάσης στην διάσταση καθώς δεν υπάρχουν άλλα επίπεδα κάτω.

Η ομαδοποίηση αυτή επιτρέπει να λυθούν και τα δύο προβλήματα μαζί. Μια εφαρμογή μπορεί τώρα να αθροίσει μαζί τα δεδομένα και για τα τρία είδη λάμπας και να τα αποθηκεύσει στο νέο πεδίο "Bulbs" (Λάμπες). Χωρίς το νέο πεδίο, ένας κύβος δεν έχει πουθενά να αποθηκεύσει τις τιμές που δημιουργήθηκαν αθροίζοντας μαζί τις τιμές για κάθε μεμονωμένο τύπο λάμπας. Επίσης έτσι μπορεί μια εφαρμογή να απλοποιήσει τη διάταξη της παρουσιάζοντας μόνο τα πεδία-γονείς για το χρήστη που συνήθως είναι ένας πολύ μικρότερος κατάλογος από τον πλήρη κατάλογο των πεδίων βάσης.

Οι ιεραρχίες βέβαια δεν περιορίζονται σε ένα επίπεδο. Για παράδειγμα ένα παγκόσμιο κατάσταση πώλησης υπολογιστών θα μπορούσε να αρχίσει με την ομαδοποίηση των καταστημάτων σε πόλεις. Αυτές οι πόλεις στη συνέχεια θα αθροιστούν σε χώρες και οι χώρες σε ηπείρους. Ένα τελικό πεδίο «Υφήλιος» θα παρουσιάζει την συνολική εικόνα.

## 4.4 Λειτουργίες του OLAP

Το OLAP έχει ένα φιλικό προς το χρήστη περιβάλλον που παρέχει κάποιες λειτουργίες που υλοποιούν διαφορετικές προβολές δεδομένων, επιτρέποντας την διαδραστική ανάλυση των δεδομένων [8]. Παρακάτω αναλύουμε τις πιο δημοφιλείς λειτουργίες που υποστηρίζονται από τις εφαρμογές πολυδιάστατων λογιστικών φύλλων.

### 4.4.1 Τεμαχισμός και κομμάτισμα (λειτουργίες Slicing και Dicing)

Αυτή είναι μια φράση που χρησιμοποιείται για να περιγράψει δύο διαδικασίες που χρησιμοποιούνται για την ανάκτηση και την προβολή δεδομένων που είναι αποθηκευμένα σε έναν κύβο OLAP. Δεδομένου ότι τα στοιχεία μπορούν αποτελεσματικά να εμφανίζονται μόνο σε δισδιάστατη μορφή, ο πολυδιάστατος κύβος πρέπει να περιορίζεται σε επίπεδες «φέτες» δεδομένων, και αυτό γίνεται με το "**Slicing**". Ουσιαστικά είναι σαν ένα ειδικό φίλτρο για ένα συγκεκριμένο πεδίο σε μια συγκεκριμένη διάσταση [9]. Το "**Dicing**" είναι παρόμοιο με το "**Slicing**" αλλά αντί για να τεμαχίζει τον κύβο σε «φέτες», είναι πιο πολύ σαν να μεγεθύνει ένα συγκεκριμένο κομμάτι του κύβου. Έτσι δημιουργείται ένας «υπο-κύβος» όπου μας δείχνει ένα συγκεκριμένο πεδίο σε όλες τις διαστάσεις [9].

### 4.4.2 Συναθροιστική άνοδος, Αναλυτική κάθοδος και Οριζόντια ανάλυση πρωτογενών στοιχείων(λειτουργίες Drill-down, Drill-up & Drill-across)

Αν μια εφαρμογή εμφανίζει μόνο το πεδίο-γονέα σε έναν κύβο, όπως τα δεδομένα για ολόκληρη την Ευρώπη και ο χρήστης θέλει να προβάλλονται τα δεδομένα για ένα

συγκεκριμένο κράτος, τότε το να αναλύσει το πεδίο-γονέα Ευρώπη στα πεδία παιδιά του ονομάζεται **"Drilling-down"**. Αντιστρόφως, το **"Drilling-up"** αναφέρεται στη διαδικασία της επιλογής ενός πεδίου-παιδιού και η εμφάνιση του πεδίο-γονέα του.

Το **"Drill-across"** αλλάζει τα πεδία που αναλύονται στον κύβο, αλλάζοντας τα δεδομένα που προβάλλονται χωρίς να αλλάξει τις διαστάσεις. Για παράδειγμα αν ένας κύβος περιέχει στοιχεία για πωλήσεις, ένας χρήστης μπορεί να χρησιμοποιήσει αυτή τη λειτουργία για να αναλύσει τα δεδομένα για τα προϊόντα με τις ίδιες διαστάσεις.

#### 4.5 Συγκεντρωτικά δεδομένα

Μια άλλη από τις βασικές περιγραφές του OLAP είναι η έννοια της «ταχείας απόκρισης», που σημαίνει ο χρόνος που χρειάζεται ένας χρήστης για να λάβει τα δεδομένα που απαιτεί. Όπως έχουμε ήδη δει, αν και οι μηχανές των βάσεων δεδομένων μπορούν να επεξεργαστούν δεδομένα, όπως ο συνολικός αριθμός των πωλήσεων σε μια περιοχή, δυσκολεύονται να το κάνουν διότι δεν έχουν σχεδιαστεί για να απαντούν σε τέτοια queries και έτσι, δεν αποθηκεύουν τα δεδομένα με τέτοιο τρόπο ώστε να τα απαντούν αποτελεσματικά.

Οι εφαρμογές OLAP, από τη φύση τους, υποχρεούνται να παρέχουν συγκεντρωτικά δεδομένα. Αυτά τα δεδομένα προκύπτουν από την ομαδοποίηση σχετικών πεδίων για να παρέχουν πληροφορίες υψηλότερου επιπέδου. Όπως έχουμε δει, οι εφαρμογές OLAP δημιουργούν νέα πεδία για να ομαδοποιούν τα υπάρχοντα βασικά πεδία σε λογικές, διαχειρίσιμες ομάδες (ιεραρχίες). Τα δεδομένα υψηλού επιπέδου που προκύπτουν από αυτά τα ιεραρχικά πεδία σπάνια υπάρχουν στα αρχικά σύνολα δεδομένων, και έτσι οι τιμές πρέπει να υπολογιστούν από την εφαρμογή. Παρακάτω θα δούμε την τεχνική υπολογισμού των τιμών αυτών.

Store	Bulbs	Batteries	Fuses	All Products
Uptown	40	104	56	200
Midtown	52	22	31	105
Downtown	36	78	58	172
All Stores	128	204	145	477

↓  
συγκεντρωτικό  
σύνολο

Σχ 4.5.1 Ένας δύο διαστάσεων κύβος με ενοποιημένα δεδομένα.

Σκεφτείτε τον δύο διαστάσεων κύβο στο σχήμα 4.5.1. Οι τιμές στα γκριζα κουτιά είναι οι αρχικές βασικές τιμές δεδομένων και οι τιμές των λευκών κελιών είναι οι επιπλέον τιμές υψηλού επιπέδου που δημιουργούνται από την εφαρμογή OLAP. Αυτά τα πεδία-γονείς συνήθως αντιπροσωπεύουν το άθροισμα των τιμών των πεδίων-παιδιών τους, και έτσι μπορούν να υπολογιστούν προσθέτοντας μαζί τις τιμές από το πιο κάτω επίπεδο.

Παρατηρούμε ότι κελί με το συγκεντρωτικό σύνολο (σημειώνεται με έντονους χαρακτήρες) μπορεί να υπολογιστεί με την πρόσθεση των πεδίων-παιδιών είτε κατά στήλη ή κατά τη

γραμμή, αν και οι δύο ιεραρχικές διαστάσεις είναι απλά αθροίσματα. Προφανώς είναι γρηγορότερο να χρησιμοποιήσουμε τον υπολογισμό που απαιτεί τις λιγότερες προσθέσεις. Αυτό είναι άλλο ένα πλεονέκτημα της αποθήκευσης δεδομένων του OLAP. Η επεξεργασία των δεδομένων που έχουμε στους κύβους γίνεται πιο γρήγορα όταν δημιουργούμε συγκεντρωτικά δεδομένα.

#### 4.5.1 Προϋπολογισμένα εναντίον Κατά παραγγελία

Η πολυδιάστατη αποθηκεύει δεδομένων επιτρέπει στις εφαρμογές OLAP να υπολογίσουν μεγάλες ποσότητες δεδομένων πολύ γρήγορα. Συνήθως πολύ πιο γρήγορα από μια βάση δεδομένων. Ωστόσο, επειδή τα αποθηκευμένα δεδομένα του OLAP κάποιες φορές είναι πολύ μεγάλα, ο υπολογισμός του συνόλου των τιμών μπορεί να πάρει κάποια λεπτά ή ακόμα και ώρες για να ολοκληρωθεί. Έτσι ένας χρήστης μπορεί ή να έχει υπολογίσει ήδη τις τιμές και να τις έχει αποθηκεύσει ή να υπολογίζει κάθε φορά τις τιμές που χρειάζεται εκείνη τη στιγμή. Συνήθως, η επιλογή ανάμεσα στις δύο μεθόδους έχει να κάνει με τις απαιτήσεις των χρηστών, με βάση την ώρα απόκρισης

Όταν τα σύνολα δεδομένων των εφαρμογών ήταν μικρότερα και οι υπολογιστές ήταν λιγότερο ισχυροί, σχεδόν όλες οι εφαρμογές OLAP ακλουθούσαν την πρώτη μέθοδο (προϋπολογισμό δεδομένων). Αυτό σήμαινε ότι ο κύβος OLAP θα κατασκευαζόταν σε τακτά χρονικά διαστήματα (συνήθως ημερήσια ή εβδομαδιαία) για να συμπεριλάβει πρόσφατες αλλαγές και στη συνέχεια θα αποθηκευόταν κάπου για γρήγορη ανάκτηση των δεδομένων.

Τώρα τελευταία, με τους ταχύτερους επεξεργαστές και την εμφάνιση των όλο και μεγαλύτερων κύβων δεδομένων που είναι δύσκολο να αποθηκευτούν, πολλές εφαρμογές επιλέγουν την μέθοδο κατά παραγγελίας, όπου οι τιμές υπολογίζονται μόνο όταν απαιτούνται από τους χρήστες. Αυτή είναι μια πιο περίπλοκη μέθοδος εργασίας, καθώς απαιτεί συνήθως όλοι οι σχετικοί υπολογισμοί να ολοκληρωθούν μέσα σε λίγα δευτερόλεπτα, έτσι ώστε οι χρήστες να μην γνωρίζουν ότι οι τιμές που ζητούν δεν υπήρχαν λίγα δευτερόλεπτα νωρίτερα.

Υποθέτοντας ότι μπορούμε να κρατήσουμε τον χρόνο απόκρισης σε αυτά τα λίγα δευτερόλεπτα, τότε η κατά παραγγελία μέθοδος προσφέρει το απλό πλεονέκτημα της μαζικής μείωσης του απαιτούμενου αποθηκευτικού χώρου που μεταφράζεται άμεσα σε εξοικονόμηση κόστους για τις αγορές αποθηκευτικών δίσκων. Κάποια εξελιγμένα εργαλεία OLAP επιτρέπουν στην εφαρμογή να προϋπολογίσει κάποια τμήματα από τα συγκεντρωτικά στοιχεία και να αφήσει άλλα να υπολογιστούν κατά παραγγελία. Αυτό επιτρέπει στην εφαρμογή να έχει τα πλεονεκτήματα και από τις δύο μεθόδους.

Τελικά, η επιλογή βασίζεται σε διάφορους παράγοντες. Στην ποσότητα των δεδομένων που αποθηκεύεται, την ισχύ του server στον οποίο λειτουργεί η εφαρμογή και τις απαιτήσεις που έχουν οι χρήστες. Μόλις αυτοί οι παράγοντες έχουν ληφθεί υπόψη, μπορούμε να επιλέξουμε τη μέθοδο που θα χρησιμοποιήσουμε για τον υπολογισμό των δεδομένων.

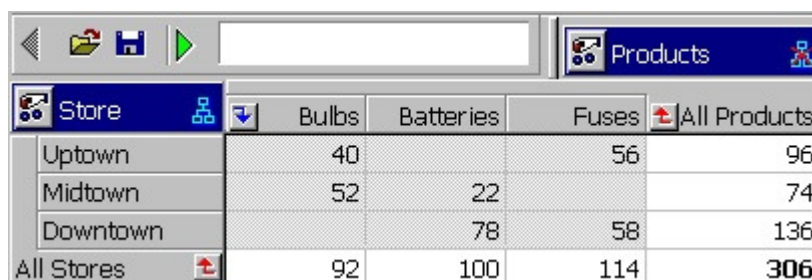
#### 4.5.2 Σποραδικά δεδομένα

Τα δεδομένα στους κύβους είναι διασταυρώσεις δύο ή περισσότερων διαστάσεων. Στην πραγματικότητα όμως πολλές από αυτές τις διασταυρώσεις δεν έχουν τιμές. Και όσο

περισσότερες διαστάσεις έχουμε τόσο πιο «αραιός» θα είναι ο κύβος. Στην πραγματικότητα τα σποραδικά δεδομένα είναι δεδομένα με «τρύπες».

Αυτές οι «τρύπες» προκύπτουν επειδή δεν είναι όλοι οι συνδυασμοί των πεδίων στις διαστάσεις εφικτοί. Τα σποραδικά δεδομένα εμφανίζονται συνήθως σε εφαρμογές που εμφανίζουν πληροφορίες για προϊόντα και καταστήματα. Σε μια μεγάλη επιχείρηση, δεν θα πουλούν όλα τα καταστήματα όλα τα προϊόντα στην γκάμα τους, έτσι όταν ένα όνομα καταστήματος συμπίπτει με ένα προϊόν που δεν πωλείται εκεί, η τιμή απλά παραμένει κενή (ή μηδενική).

Οι σχεσιακές βάσεις δεδομένων είναι και αυτές σποραδικές, απλά δεν φαίνεται τόσο εύκολα. Αυτό οφείλεται στο γεγονός ότι ένας πίνακας βάσης δεδομένων εμφανίζει μόνο τις εγγραφές που υπάρχουν. Οποιαδήποτε δεδομένα δεν υπάρχουν, συνήθως δεν εμφανίζονται στον πίνακα, και έτσι οι κενές τιμές δεν είναι άμεσα εμφανείς. Το σχήμα 4.5.2 δείχνει απλά των δύο διαστάσεων κύβο μας, αλλά αυτή τη φορά με μερικούς αριθμούς να έχουν αφαιρεθεί. Αυτά τα κενά κελιά κάνουν έναν κύβο, σποραδικό.



Store	Bulbs	Batteries	Fuses	All Products
Uptown	40		56	96
Midtown	52	22		74
Downtown		78	58	136
All Stores	92	100	114	306

Σχ. 4.5.2 Ένας σποραδικός κύβος δεδομένων.

Είναι σημαντικό να σημειωθεί ότι αυτά τα κενά κελιά δεν είναι μηδενικές τιμές, είναι απλά τρύπες στα δεδομένα. Για παράδειγμα, στο σχήμα 4.5.2 Η κενή τιμή κατά τη διασταύρωση των Bulbs (λάμπες) και Downtown δείχνει ότι το κατάστημα στο κέντρο της πόλης δεν πουλά λάμπες.

Η ικανότητα μιας εφαρμογής OLAP να χειριστεί σποραδικά δεδομένα είναι ένα σημαντικό μέρος της επίδοσης της. Αν οι τιμές των κελίων δεν υπάρχουν, τότε δεν θα πρέπει να καταλαμβάνουν χώρο στον κύβο, αλλά η θέση τους στον κύβο θα εξακολουθεί να ισχύει.

Η ιδέα ότι τα κενά κελιά δεν καταλαμβάνουν κανένα φυσικό χώρο είναι πολύ σημαντική για τις σύγχρονες εφαρμογές OLAP επειδή ένας μεγάλος κύβος OLAP έχει τη δυνατότητα να είναι τεράστιος. Μπορούμε δηλαδή να έχουμε έναν κύβο με τρισεκατομμύρια κελιά, που ουσιαστικά να έχει τιμές μόνο σε κάποιες χιλιάδες κελιά, και μόνο αυτά τα κελιά καταλαμβάνουν αποθηκευτικό χώρο.

Το θέμα του μέγιστου μεγέθους για ένα κύβο υπάρχει επειδή σε ένα κύβο, τα πεδία είναι όλα σταθερά, έτσι μπορεί να υπάρχει μόνο ένας πεπερασμένος αριθμός κελιών. Σε μια βάση δεδομένων, μπορούμε να προσθέτουμε και άλλα πεδία στις στήλες, όταν απαιτείται, με αποτέλεσμα τη σταδιακή αύξηση του μεγέθους του πίνακα.

Για παράδειγμα, ένας πίνακας μιας βάσης δεδομένων είναι σαν ένα μπαλόνη. Μπορεί να επεκταθεί και να μειωθεί, όταν προσθέτουμε ή διαγράφουμε δεδομένα. Ένας κύβος είναι σαν ένα κουτί, το σχήμα του είναι αυστηρά καθορισμένη και τα δεδομένα μπορούν να προστεθούν μέχρι να γεμίσει κουτί.

## 4.6 Αποθήκευση των δεδομένων

Υπάρχουν τρεις διαφορετικοί τρόποι φυσικής αποθήκευση των πολυδιάστατων δεδομένων που υπάρχουν μέσα σε ένα κύβο OLAP. Κάθε μέθοδος προσπαθεί να παρουσιάσει τα δεδομένα σε ένα κύβο, αλλά χρησιμοποιεί διαφορετική τεχνολογία για την επίτευξη αυτού του αποτελέσματος.

### 4.6.1 ROLAP

Προέρχεται από το "Relational OLAP" που σημαίνει σχεσιακό OLAP. Ο όρος αυτός περιγράφει τις εφαρμογές OLAP που αποθηκεύουν όλα τα δεδομένα του κύβου, τόσο τα στοιχεία βάσης όσο και τα υψηλού επιπέδου σε σχεσιακούς πίνακες. Η εφαρμογή κρύβει την παρουσία των πινάκων με την παρουσίαση των δεδομένων σε μια διάταξη κύβου. Οι χρήστες που έχουν μόνο μια σχεσιακή μηχανή βάσης δεδομένων στη διάθεσή τους πρέπει να έχουν αυτή τη μέθοδο αποθήκευσης δεδομένων.

Οι πολυδιάστατες προβολές που δημιουργούνται από τους πίνακες βασικών και συγκεντρωτικών στοιχείων μαζί με την SQL, συχνά έχουν ως αποτέλεσμα την κακή επίδοση μαζί με τη δυσκολία της διατήρησης των πινάκων συγκεντρωτικών στοιχείων, προκειμένου να βελτιωθεί ο χρόνος απόκριση.

### 4.6.2 MOLAP

Προέρχεται από το "Multidimensional OLAP" που σημαίνει πολυδιάστατο OLAP. Ο όρος αυτός περιγράφει εφαρμογές OLAP που αποθηκεύουν όλα τα δεδομένα κύβου τόσο τα στοιχεία βάσης όσο και τα υψηλού επιπέδου σε πολυδιάστατα αρχεία δεδομένων. Η εφαρμογή αντιγράφει τα δεδομένα βάσης από τον πίνακα σε μια πολυδιάστατη μορφή δεδομένων (συνήθως ένα δυαδικό αρχείο δεδομένων) και, στη συνέχεια υπολογίζει τις συγκεντρωτικές τιμές.

Οι πολυδιάστατες προβολές δεδομένων υπάρχουν από την αρχή σε αυτή τη μέθοδο και η επίδοση είναι συχνά πολύ γρήγορη, ιδιαίτερα εάν οι κύβοι είναι αρκετά μικροί ώστε να χωράνε στη μνήμη RAM. Αλλιώς, τα δεδομένα αποθηκεύονται σε μεγάλα αρχεία στο δίσκο. Το μεγαλύτερο μειονέκτημα αυτής της μεθόδου είναι ο διπλασιασμός των δεδομένων της βάσης, που γίνεται όταν αντιγράφεται στο κύβο, και χρειάζονται περισσότερο χώρο στο δίσκο και τη περισσότερη ώρα επεξεργασίας.

### 4.6.3 HOLAP

Προέρχεται από το "Hybrid OLAP" που σημαίνει υβριδικό OLAP. Ο όρος αυτός περιγράφει τις εφαρμογές OLAP που αποθηκεύουν δεδομένων υψηλού επιπέδου σε αρχεία πολυδιάστατων δεδομένων, αλλά αφήνουν τα βασικά δεδομένα στους αρχικούς πίνακες.

Αυτή η μέθοδος έχει το μεγάλο πλεονέκτημα ότι δεν απαιτείται η επανάληψη των δεδομένων βάσης, με αποτέλεσμα να αποταμιεύει χώρο και χρόνο στο δίσκο. Ο κύβος οδηγεί τις πολυδιάστατες προβολές, έτσι ώστε η εφαρμογή να απαιτεί έναν σύνδεσμο μεταξύ των



πολυδιάστατων αρχείων δεδομένων και του σχεσιακού πίνακα που αποθηκεύει τα δεδομένα βάσης.

#### 4.6.4 Σύγκριση μεταξύ ROLAP, HOLAP και MOLAP

Οι κάθε μια από αυτές τις μεθόδους έχει τα δικά της πλεονεκτήματα και μειονεκτήματα. Στον παρακάτω πίνακα μπορούμε να δούμε μια σύγκριση και των τριών μεθόδων μαζί από το [10].

	<b>ROLAP</b>	<b>HOLAP</b>	<b>MOLAP</b>
<b>Επίδοση των queries</b>	Πιο αργή	Μέτρια	Πιο γρήγορη
<b>Φόρτωση δεδομένων</b>	Δεν χρειάζεται φόρτωση	Μέτρια	Δεν είναι λειτουργικό για μεγάλα σύνολα δεδομένων
<b>Επεκτασιμότητα (χειρισμός μεγάλων διαστάσεων)</b>	Μεγίστη επεκτασιμότητα με ισχυρές εγκαταστάσεις RDBMS	Εξαρτάτε από τις επιλογές σχεδιασμού	Έχει προβλήματα με πολλές και μεγάλες διαστάσεις
<b>Ευκολία πρόσβασης δεδομένων</b>	Οποιοδήποτε εργαλείο SQL	Μόνο εργαλεία OLAP	Μόνο εργαλεία OLAP
<b>Χειρισμός μη-συγκεντρωτικών στοιχείων</b>	Καλύτερος	Ευέλικτο, βάση της επιλογής σχεδιασμού	Χαμηλή επίδοση
<b>Ασφάλεια</b>	Ισχυρές εγκαταστάσεις ασφαλείας RDBMS	Πιο αδύναμη από του ROLAP	Πιο αδύναμη από του ROLAP
<b>Αποθηκευτικός χώρος που χρειάζεται</b>	Χρειάζεται περισσότερο αποθηκευτικό χώρο από το MOLAP	Μέτριος	Το λιγότερο δυνατό (με ειδικές τεχνικές συμπίεσης)
<b>Πλεονασμός δεδομένων</b>	Κανένας	Μερικώς πλεονασμός δεδομένων	Διπλασιάζει όλο το σύνολο δεδομένων

## 5 Το εργαλείο διαχείρισης δεδομένων Pentaho

Σε αυτό το κεφάλαιο θα δούμε εργαλεία διαχείρισης δεδομένων που μπορούν να χρησιμοποιήσουν OLAP. Στην σημερινή αγορά ο όγκος των δεδομένων όλο και αυξάνεται γι αυτό το λόγο και χρησιμοποιούμε όλο και περισσότερο αυτά τα προγράμματα για να αναλύσουμε και να παρουσιάσουμε τα δεδομένα μας με τέτοιο τρόπο ώστε να μπορούμε να τα αξιολογήσουμε καλύτερα. Το πρόγραμμα με το οποίο θα ασχοληθούμε εμείς είναι το Pentaho.

Το Pentaho προσφέρει αρκετά εργαλεία ανάλυσης και εξόρυξης δεδομένων, έναν διακομιστή OLAP και ερωτήματα πολυδιάστατων δεδομένων. Το Pentaho δίνει πολλές δυνατότητες να εξερευνήσεις κρυμμένα μοτίβα στα δεδομένα σου. Είναι δωρεάν και έχει μια πολύ καλή κοινότητα προγραμματιστών να δουλεύουν σε αυτό. Έτσι προσφέρει αρκετές επεκτάσεις ανοιχτού λογισμικού και πολύτιμη βοήθεια και υποστήριξη για προγραμματιστικές ανάγκες.

Το Pentaho είναι μια πλατφόρμα ενσωμάτωσης δεδομένων που μπορεί να προσφέρει ακριβή δεδομένα έτοιμα για ανάλυση. Αυτά τα δεδομένα μπορεί να τα δει ο χρήστης πολύ εύκολα χωρίς να χρειάζεται να γράψει κώδικα. Η εταιρεία Pentaho παρέχει λύσεις για ενσωμάτωση και ανάλυση δεδομένων από το 2004.

Το εργαλείο αυτό είναι ιδανικό για μικρές, μεσαίες και μεγάλες επιχειρήσεις ανεξαρτήτως περιεχομένου. Κάποιες από τις εταιρείες που χρησιμοποιούν το Pentaho είναι η Mozilla και η Swissport. Το Pentaho έχει ένα φιλικό προς το χρήστη περιβάλλον με το οποίο οι χρήστες μπορούν να έχουν εύκολη και γρήγορη πρόσβαση στα δεδομένα τους και τις αναφορές τους.

Τα πλεονεκτήματα του Pentaho:

- Το Pentaho προσφέρει ευέλικτη υποστήριξη για όλα τα δεδομένα.
- Τα εργαλεία ανάλυσης δεδομένων του Pentaho δεν απαιτούν γνώσεις προγραμματισμού.
- Το Pentaho προσφέρει 24ωρη διαδικτυακή και τηλεφωνική εξυπηρέτηση πελατών.
- Το εργαλείο ανάλυσης δεδομένων είναι φιλικό προς το χρήστη και έχει εύκολη πρόσβαση σε εταιρικούς χρήστες.

Τα μειονεκτήματα του Pentaho:

Το εργαλείο ανάλυσης δεδομένων του Pentaho μπορεί να απορροφά πολλούς πόρους συστήματος όταν έχουμε να επεξεργαστούμε πολλά δεδομένα

Μια εναλλακτική του Pentaho είναι το icCube. Το icCube είναι ένα υψηλής απόδοσης πρόγραμμα ανάλυσης δεδομένων. Είναι φτιαγμένο από μια ομάδα ανθρώπων που έχουν σαν κίνητρο την ποιότητα και την ικανοποίηση του πελάτη. Είναι σχεδιασμένο για μεγάλο όγκο δεδομένων και υποστηρίζει αναφορές πολυδιάστατων δεδομένων σε MDX (προέρχεται από το MultiDimensional eXpressions και είναι γλώσσα προγραμματισμού για ερωτήματα όπως η SQL.). Ένα θετικό είναι το γεγονός πως υποστηρίζει XMLA, μια συνήθης μέθοδος για πρόσβαση δεδομένων σε συστήματα ανάλυσης όπως το OLAP, και μπορεί να συνδεθεί σε διάφορα εργαλεία αναφοράς (για παράδειγμα Excel). Το πρόγραμμα αυτό απευθύνεται σε μικρές και μεσαίες επιχειρήσεις.

## 6 Μελέτη περίπτωσης: Χρησιμοποιώντας OLAP σε βάση δεδομένων με αποτελέσματα αξιολόγησης

### 6.1 Τα δεδομένα αξιολόγησης

Η βάση δεδομένων που θα ασχοληθούμε έχει δημιουργηθεί από ερωτηματολόγια που έχουν απαντήσει τα τελευταία χρόνια οι φοιτητές του Ελληνικού Ανοικτού Πανεπιστημίου (ΕΑΠ). Σε αυτά τα ερωτηματολόγια οι φοιτητές κλήθηκαν να απαντήσουν σε διάφορες ερωτήσεις και να βαθμολογήσουν τους καθηγητές σε όλο το εκπαιδευτικό τους έργο, για τις χρονιές 2012-2016. Για ευκολία εμείς έχουμε συγκεντρώσει αυτά τα στοιχεία και για κάθε καθηγητή έχουμε βγάλει έναν μέσο όρο ώστε να έχουμε μια τιμή για κάθε χρονιά. Τα δεδομένα αυτά είναι επίτηδες διαφοροποιημένα από τα πραγματικά για λόγους εμπιστευτικότητας αλλά έχουμε κρατήσει τη μορφή τους ώστε τα αποτελέσματα να είναι ακριβή. Στο σχήμα 6.1.1 βλέπουμε ένα μικρό μέρος των δεδομένων μας για ένα συγκεκριμένο έτος.

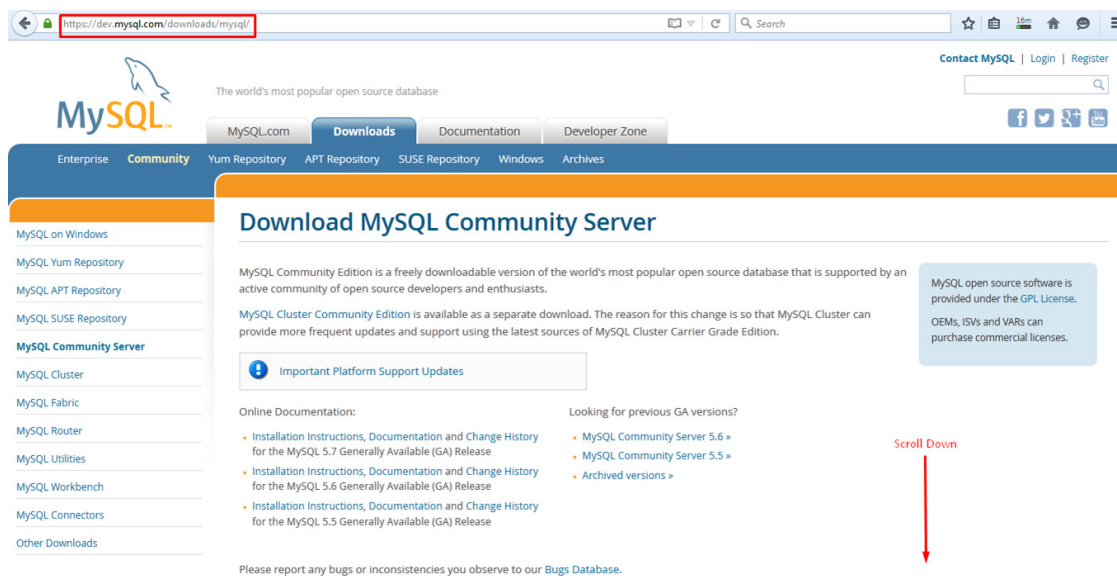
	A	B	C	D	E	F
1	ΣΧΟΛΗ	ΠΡΟΓΡΑΜΜΑ_ΣΠΟΥΔΩΝ	ΜΑΘΗΜΑ	ΤΑΞΗ	ΚΑΘΗΓΗΤΗΣ	ΒΑΘΜΟΣ
2	ΘΕΤ	ΠΣΕ	ΠΣΕ51	ΑΘΗ1	ΝΙΚΟΛΑΟΣ ΠΑΡΑΝΥΧΙΔΑΣ	4,34
3	ΑΣ	ΓΑΛ	ΓΑΛ50	ΑΘΗ2	ΣΟΦΙΑ ΣΑΧΛΟΥ	2,14
4	ΘΕΤ	ΠΣΕ	ΠΣΕ50	ΘΕΣ2	ΝΙΚΟΛΑΟΣ ΜΑΝΩΛΗΣ	2,84
5	ΑΣ	ΓΕΡ	ΓΕΡ51	ΑΘΗ1	ΓΕΩΡΓΙΑ ΑΛΤΗ	2,66
6	ΘΕΤ	ΠΣΕ	ΠΣΕ51	ΑΘΗ1	ΝΙΚΟΛΑΟΣ ΠΑΡΑΝΥΧΙΔΑΣ	3,15
7	ΚΕ	ΜΒΑ	ΜΒΑ51	ΘΕΣ2	ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ	2,62
8	ΘΕΤ	ΠΣΕ	ΠΣΕ51	ΑΘΗ1	ΝΙΚΟΛΑΟΣ ΠΑΡΑΝΥΧΙΔΑΣ	4,8
9	ΑΣ	ΑΓΓ	ΑΓΓ52	ΘΕΣ2	ΦΩΤΕΙΝΗ ΔΗΜΗΤΡΙΟΥ	1,15
10	ΘΕΤ	ΠΛΗ	ΠΛΗ10	ΠΑΤ1	ΝΙΚΟΛΑΟΣ ΚΑΥΓΑΣ	4,18
11	ΚΕ	ΤΡΑ	ΤΡΑ50	ΘΕΣ2	ΦΩΤΙΟΣ ΣΑΚΗΣ	4,56
12	ΑΣ	ΑΓΓ	ΑΓΓ53	ΑΘΗ1	ΒΑΣΙΛΕΙΑ ΚΑΖΟΥ	1,1
13	ΚΕ	ΜΒΑ	ΜΒΑ50	ΘΕΣ2	ΙΩΑΝΝΗΣ ΚΑΡΑΓΑΣ	4,38
14	ΚΕ	ΜΒΑ	ΜΒΑ51	ΘΕΣ2	ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ	3,34
15	ΑΣ	ΑΓΓ	ΑΓΓ53	ΘΕΣ1	ΒΑΣΙΛΕΙΑ ΚΑΖΑΔΥΟ	4,81
16	ΚΕ	ΜΒΑ	ΜΒΑ51	ΘΕΣ2	ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ	4,48
17	ΘΕΤ	ΠΛΗ	ΠΛΗ10	ΘΕΣ1	ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΣ	4,97
18	ΑΣ	ΓΕΡ	ΓΕΡ51	ΑΘΗ1	ΓΕΩΡΓΙΑ ΑΛΤΗ	4,03
19	ΘΕΤ	ΠΛΗ	ΠΛΗ10	ΘΕΣ1	ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΣ	4,02
20	ΚΕ	ΜΒΑ	ΜΒΑ51	ΘΕΣ2	ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ	4,58
21	ΑΣ	ΓΑΛ	ΓΑΛ51	ΑΘΗ1	ΟΛΓΑ ΠΑΤΕΡΑ	1,97
22	ΚΕ	ΜΒΑ	ΜΒΑ51	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΑΡΓΗΣ	1,89
23	ΚΕ	ΤΡΑ	ΤΡΑ50	ΑΘΗ1	ΕΥΑΓΓΕΛΟΣ ΜΑΝΟΣ	3,85
24	ΘΕΤ	ΠΛΣ	ΠΛΣ50	ΑΘΗ2	ΙΩΑΝΝΗΣ ΡΕΦΑΝΗΣ	4,56
25	ΘΕΤ	ΠΣΕ	ΠΣΕ51	ΑΘΗ2	ΔΗΜΗΤΡΗΣ ΝΙΚΟΛΑΟΥ	2,91

Σχ 6.1.1 Τα δεδομένα μας σε μορφή Excel

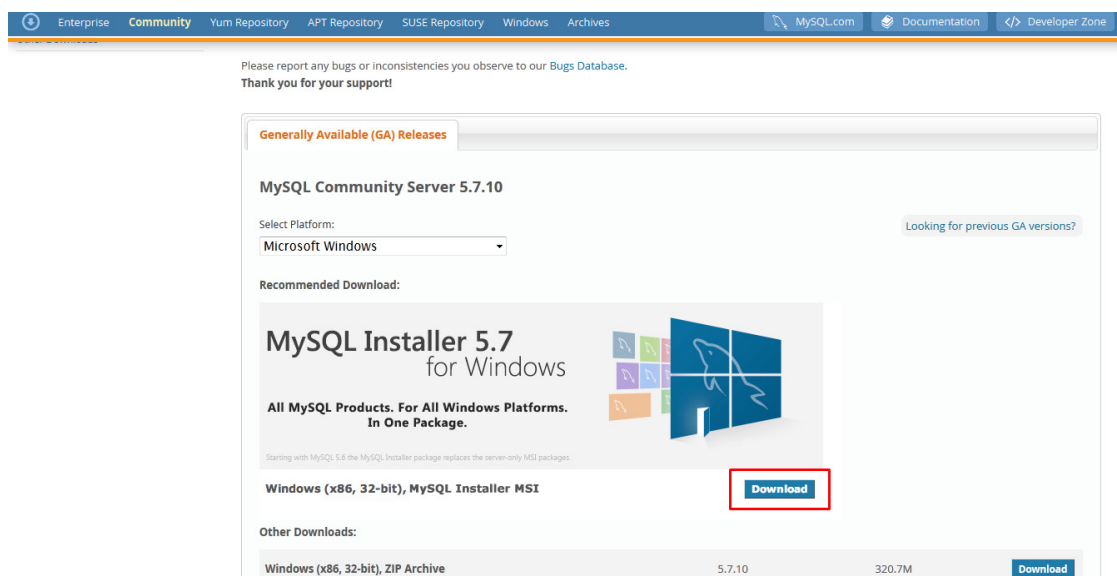
## 6.2 Εγκατάσταση OLAP (Pentaho) και εισαγωγή δεδομένων

Για να μπορέσουμε να κάνουμε ερωτήματα σε OLAP εμείς επιλέξαμε το Pentaho. Θα χρειαστεί να καταχωρήσουμε τα δεδομένα μας σε μια βάση δεδομένων σε SQL που θα συνδεθεί με το Pentaho μέσω ενός σχήματος Mondrian XML. Ένα απλό πρόγραμμα για τη βάση δεδομένων μας είναι το MySQL Workbench. Παρακάτω θα δούμε πως μπορούμε να:

- Εγκαταστήσουμε το Workbench
- Εισάγουμε τα δεδομένα
- Εγκαταστήσουμε το Pentaho
- Δημιουργήσουμε το σχήμα μας
- Συνδέσουμε την βάση δεδομένων με το Pentaho



Σχ 6.2.1 Μπαίνουμε στο site [www.mysql.com](http://www.mysql.com)



Σχ 6.2.2 Επιλέγουμε το «Download».

The screenshot shows the MySQL website's 'Downloads' section for the MySQL Installer. The page includes a navigation menu with 'Downloads' selected. The main content area is titled 'Download MySQL Installer' and contains the following text:

MySQL Installer provides an easy to use, wizard-based installation experience for all your MySQL software needs. Included in the product are the latest versions of:

- MySQL Server
- MySQL Connectors
- MySQL Workbench and sample models
- Sample Databases
- MySQL for Excel
- MySQL Notifier
- MySQL for Visual Studio
- Documentation

**Choosing the right file:**

- If you have an online connection while running the MySQL Installer, choose the `mysql-installer-web-community` file.
- If you do NOT have an online connection while running the MySQL Installer, choose the `mysql-installer-community` file.

**Note: MySQL Installer is 32 bit, but will install both 32 bit and 64 bit binaries.**

Online Documentation

- MySQL Installer Documentation and Change History

Please report any bugs or inconsistencies you observe to our [Bugs Database](#).  
Thank you for your support!

A red arrow labeled 'Scroll' points downwards on the right side of the page.

Σχ 6.2.3 Προχωράμε τη σελίδα προς τα κάτω.

The screenshot shows the 'Generally Available (GA) Releases' section for MySQL Installer 5.7.10. The page displays a table of available releases for Microsoft Windows. The table has the following data:

Platform	Version	Size	Action
Windows (x86, 32-bit), MSI Installer	5.7.10	1.6M	<a href="#">Download</a>
Windows (x86, 32-bit), MSI Installer	5.7.10	371.7M	<a href="#">Download</a>

The 'Download' buttons for both releases are highlighted with a red border. Below the table, there is a note: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download."

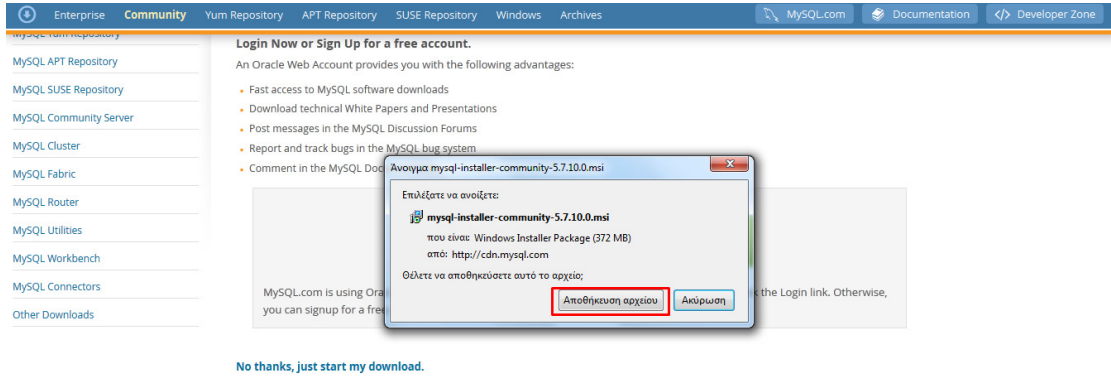
Σχ 6.2.4 Επιλέγουμε το κουμπί «Download» όπως φαίνεται στο σχήμα.

The screenshot shows the MySQL website's download page. At the top, there is a navigation bar with 'MySQL.com', 'Downloads', 'Documentation', and 'Developer Zone'. Below this is a secondary navigation bar with 'Enterprise', 'Community', 'Yum Repository', 'APT Repository', 'SUSE Repository', 'Windows', and 'Archives'. The main content area is titled 'Begin Your Download - mysql-installer-community-5.7.10.0.msi'. It features a section for 'Login Now or Sign Up for a free account.' with a list of advantages: fast access to downloads, technical papers, discussion forums, bug reporting, and documentation comments. Two buttons are present: 'Login » using my Oracle Web account' and 'Sign Up » for an Oracle Web account'. A red arrow labeled 'Scroll' points downwards from the sign-up button area.

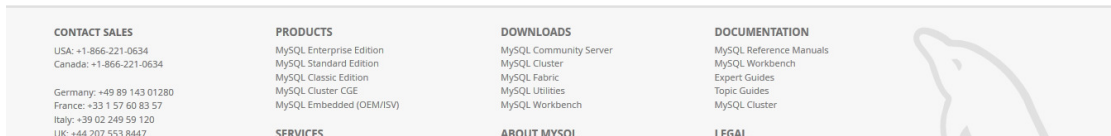
Σχ 6.2.5 Προχωράμε τη σελίδα προς τα κάτω.

This screenshot shows the same MySQL website page as the previous one, but with the 'No thanks, just start my download.' link highlighted in a red box. The page content is identical, including the navigation bars, the download title, the account prompt, the list of advantages, and the login/sign-up buttons. The footer contains contact information for sales, products, downloads, and documentation.

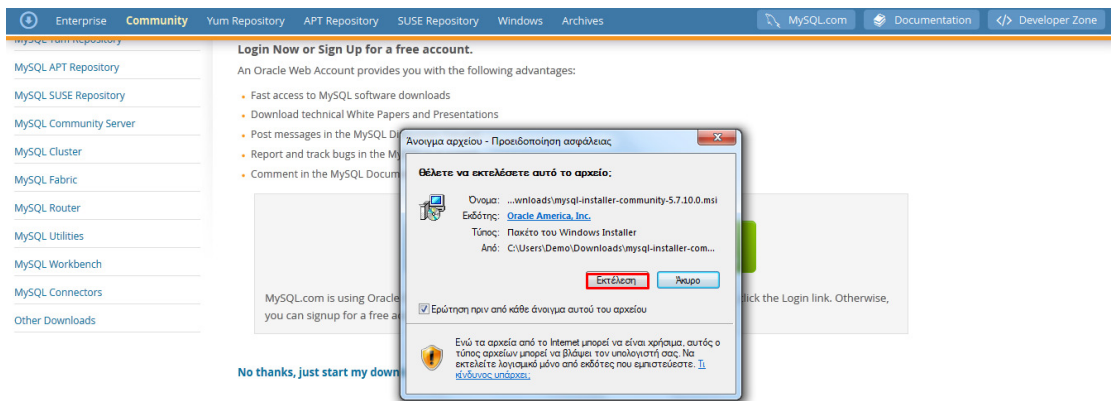
Σχ 6.2.6 Επιλέγουμε να συνεχίσουμε στο «Download» χωρίς λογαριασμό.



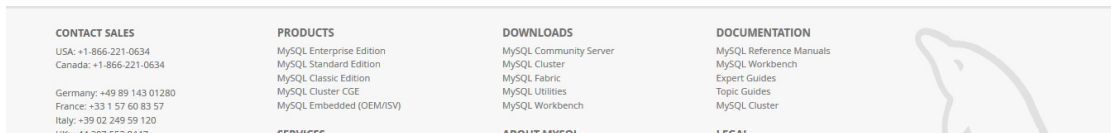
No thanks, just start my download.



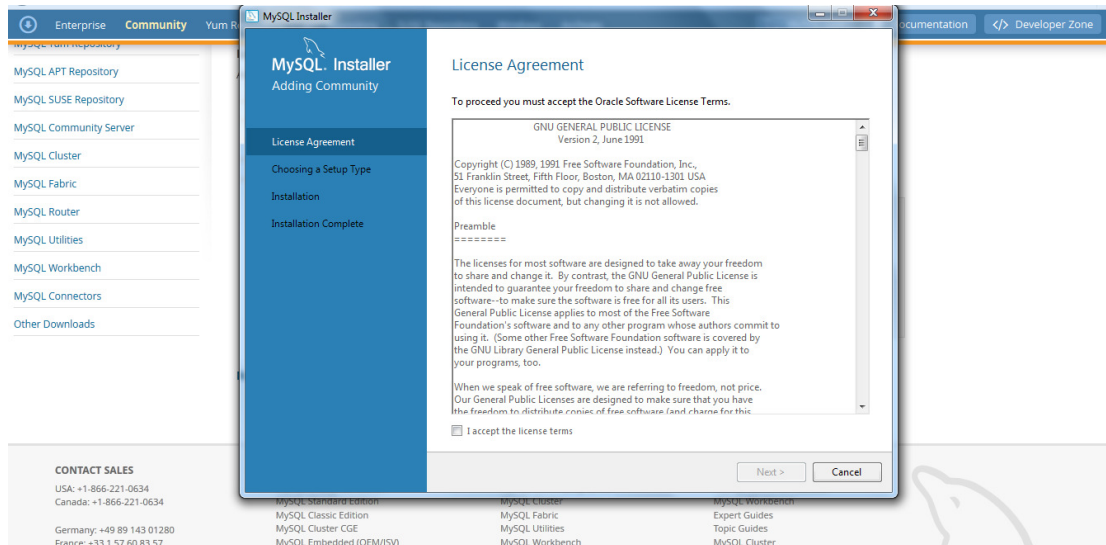
Σχ 6.2.7 Στο παράθυρο που θα εμφανιστεί επιλέγουμε «Αποθήκευση αρχείου».



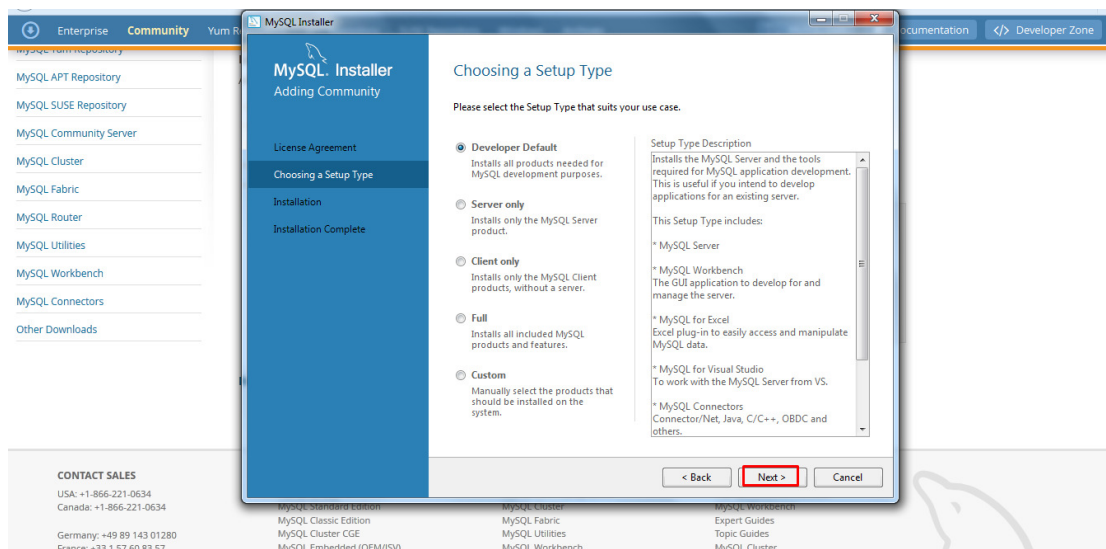
No thanks, just start my download.



Σχ 6.2.8 Ανοίγουμε το αρχείο που κατεβάσαμε και επιλέγουμε «Εκτέλεση».

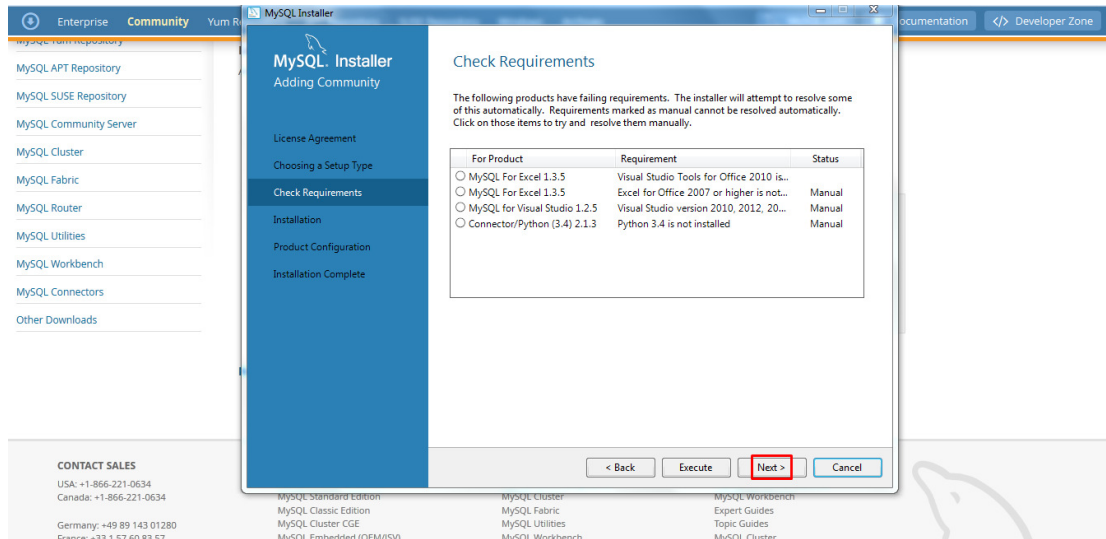


Σχ 6.2.9 Στο παράθυρο που εμφανίζεται αποδεχόμαστε τους όρους και πατάμε «Next».

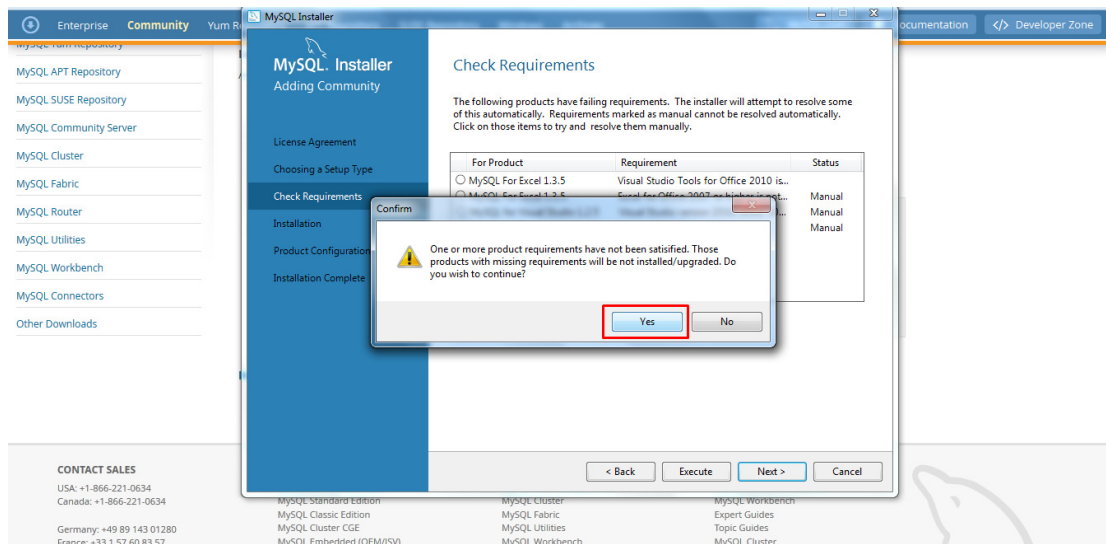


Σχ 6.2.10 Αφήνουμε την προεπιλογή και έπειτα πατάμε «Next».

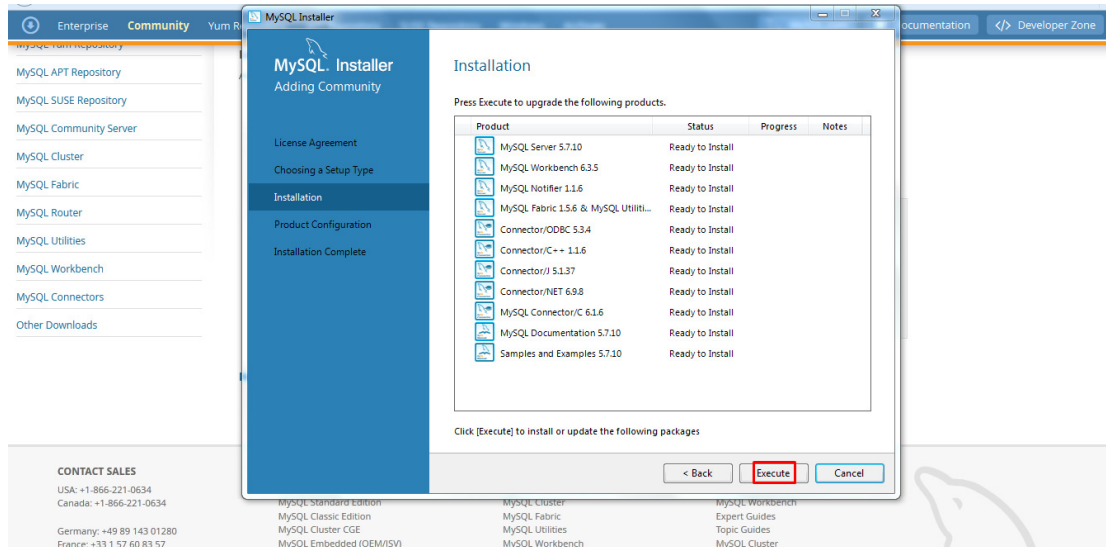




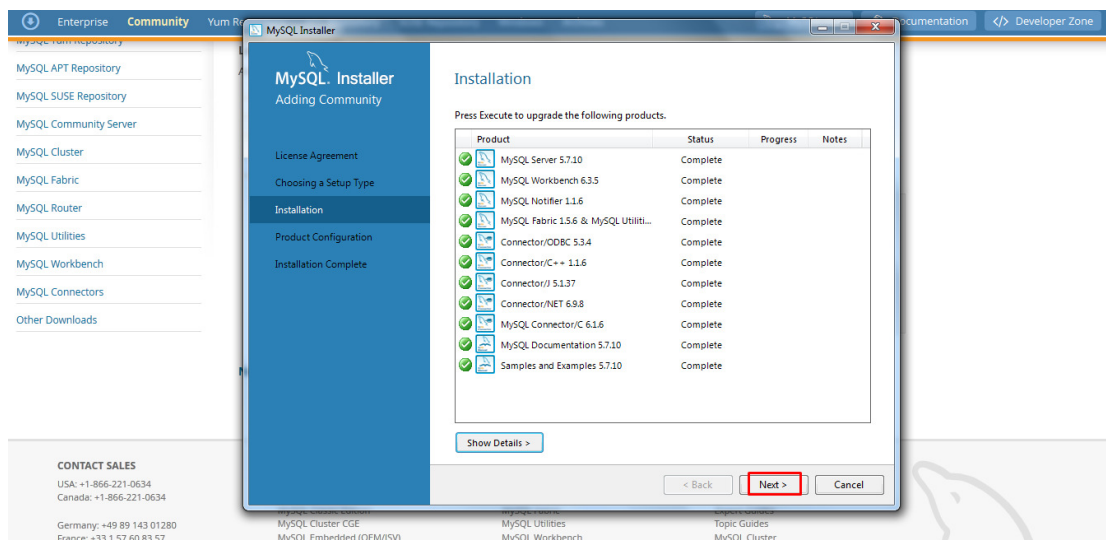
Σχ 6.2.11 Πατάμε «Next» χωρίς να επιλέξουμε κάτι άλλο.



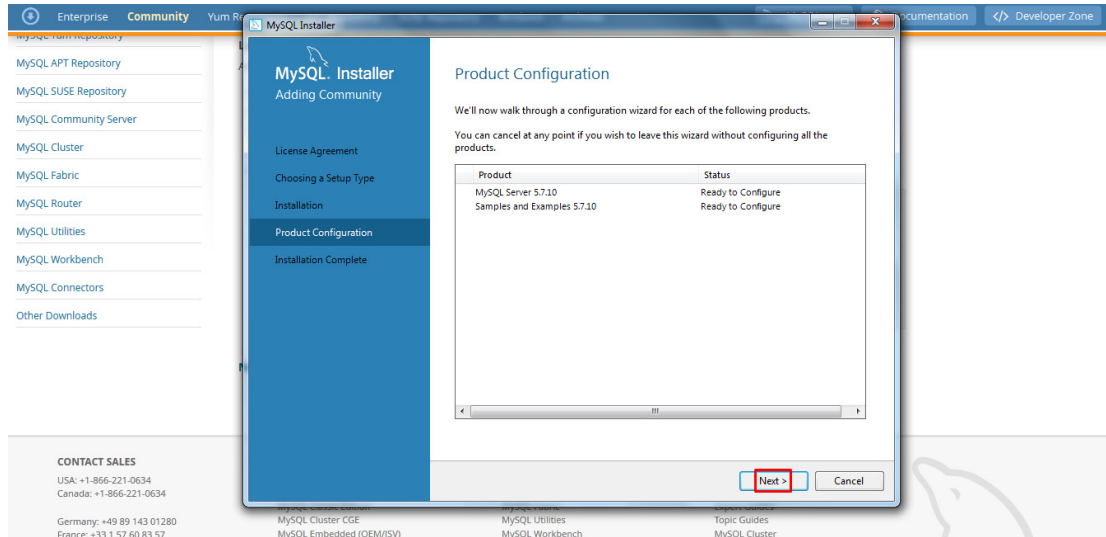
Σχ 6.2.12 Στο παράθυρο που θα εμφανιστεί επιλέγουμε «Yes».



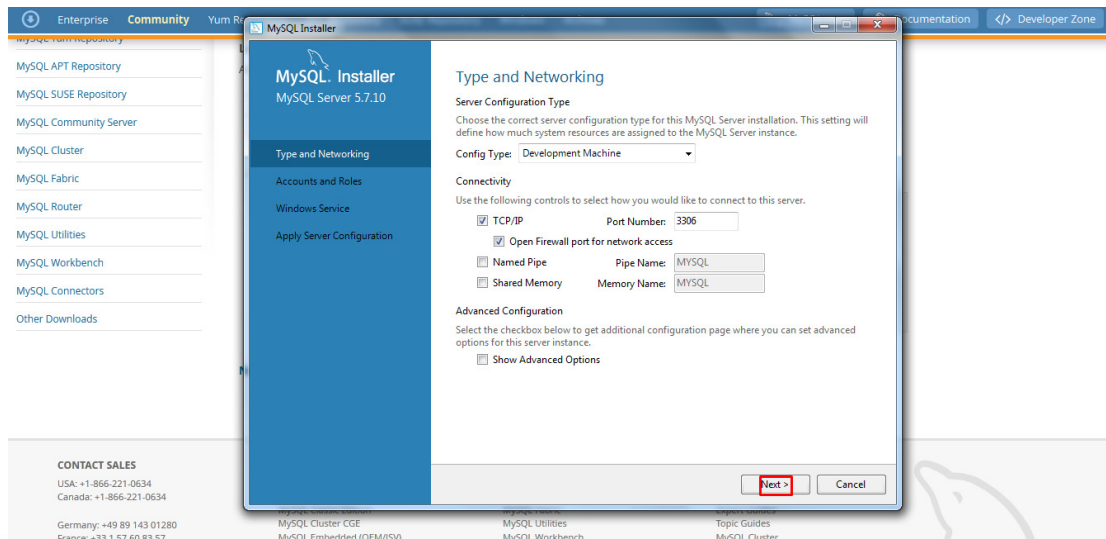
Σχ 6.2.13 Από την προεπιλογή είναι επιλεγμένα όλα και πατάμε το «Execute»



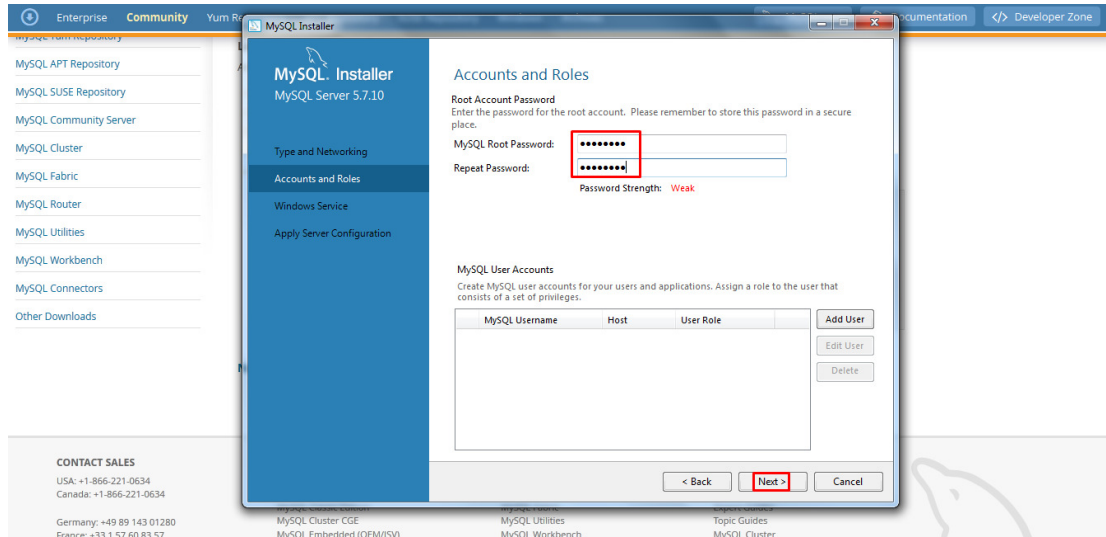
Σχ 6.2.14 Όταν ολοκληρωθεί η εγκατάσταση πατάμε «Next».



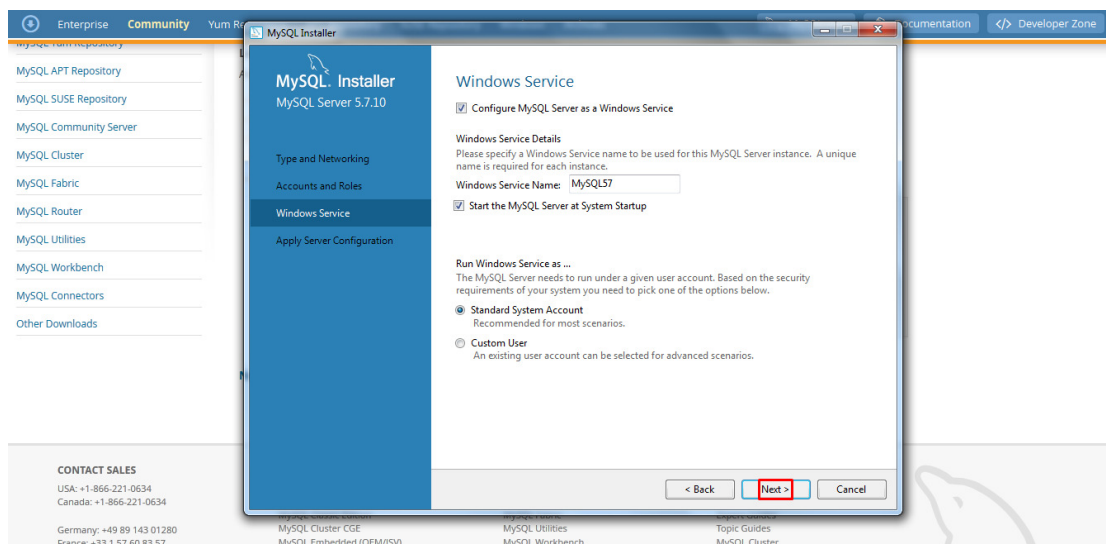
Σχ 6.2.15 Επιλέγουμε το «Next».



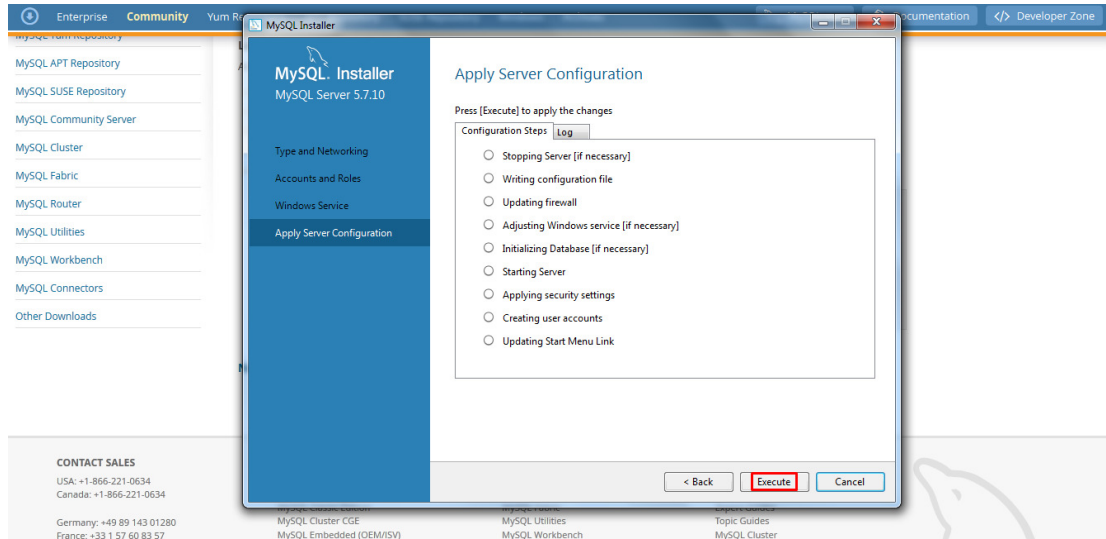
Σχ 6.2.16 Εάν επιθυμούμε αλλάζουμε το Port Number και επιλέγουμε «Next».



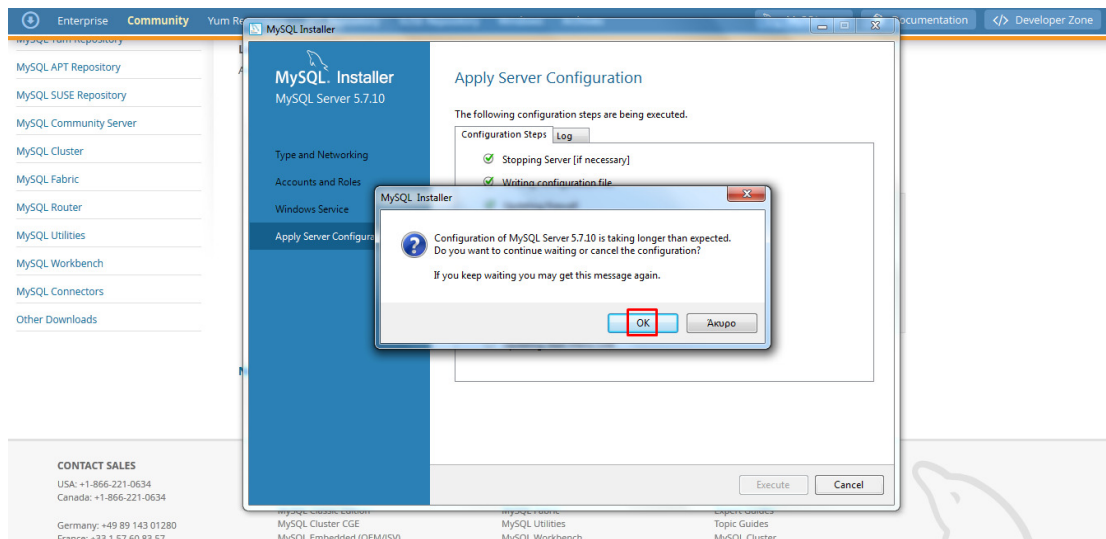
Σχ 6.2.17 Διαλέγουμε κωδικό πρόσβασης. Τον πληκτρολογούμε και στα δύο πεδία και έπειτα πατάμε «Next».



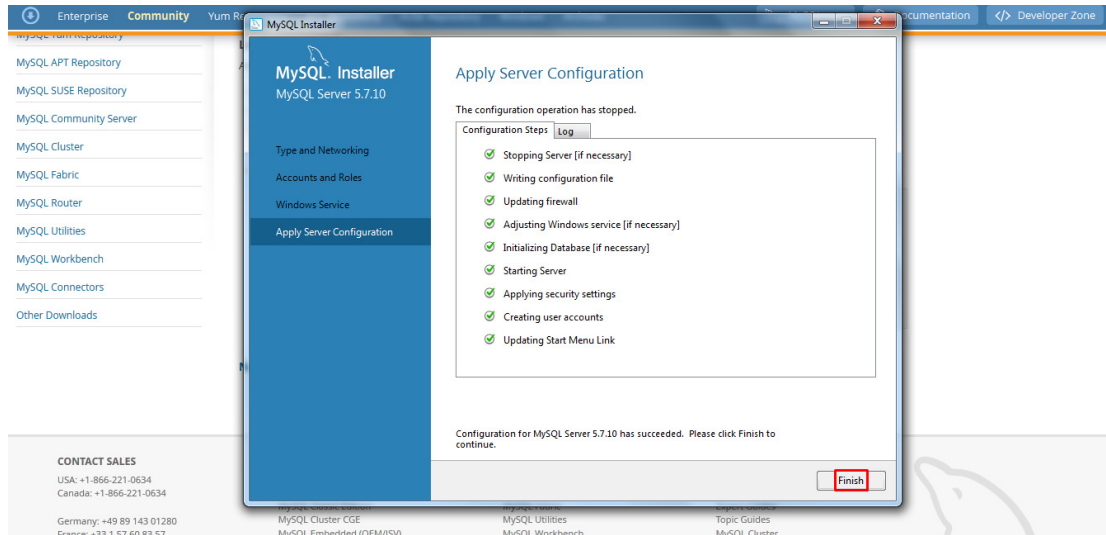
Σχ 6.2.18 Όπως είναι όλα προεπιλεγμένα πατάμε «Next».



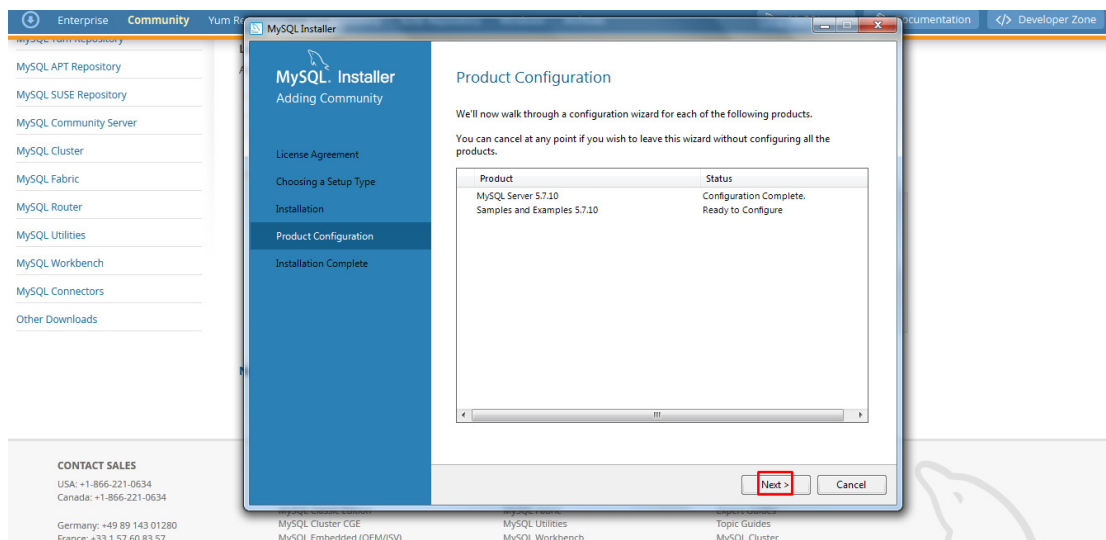
Σχ 6.2.19 Πατάμε «Execute».



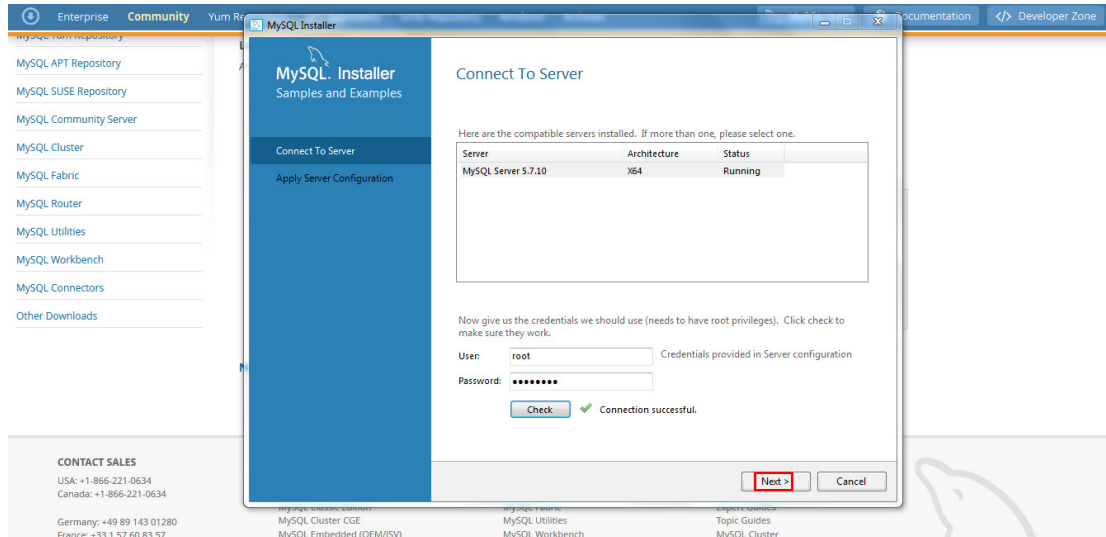
Σχ 6.2.20 Στο παράθυρο που θα εμφανιστεί στην οθόνη επιλέγουμε «OK».



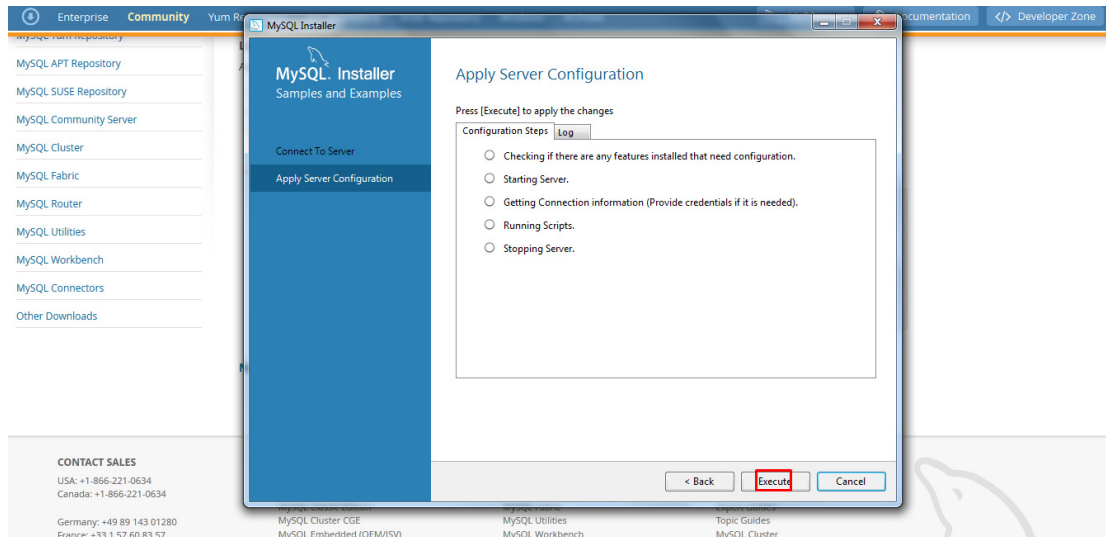
Σχ 6.2.21 Μόλις ολοκληρωθεί η διαδικασία επιλέγουμε «Finish».



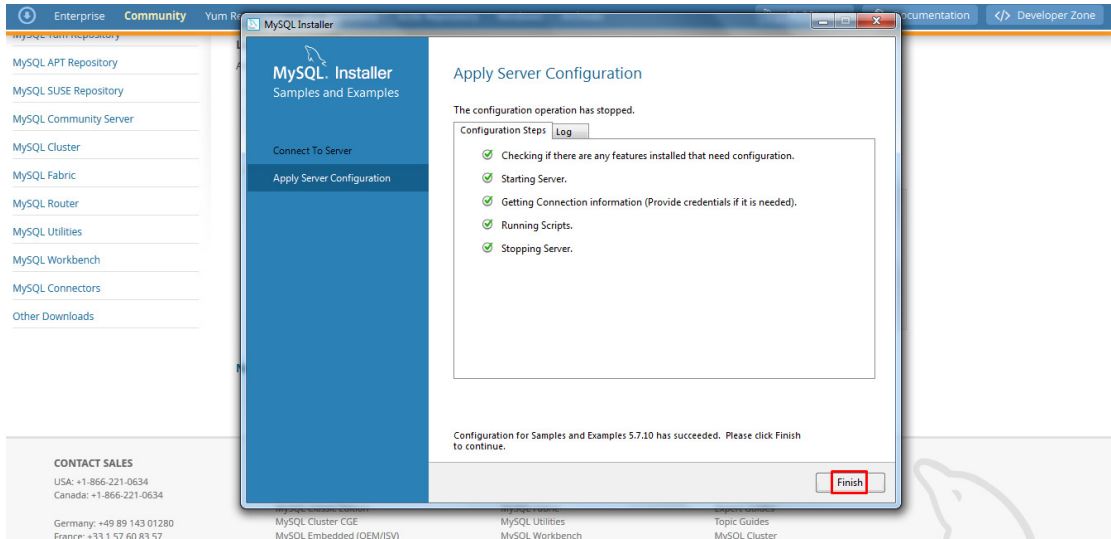
Σχ 6.2.22 Στο παράθυρο που θα εμφανιστεί επιλέγουμε «Next».



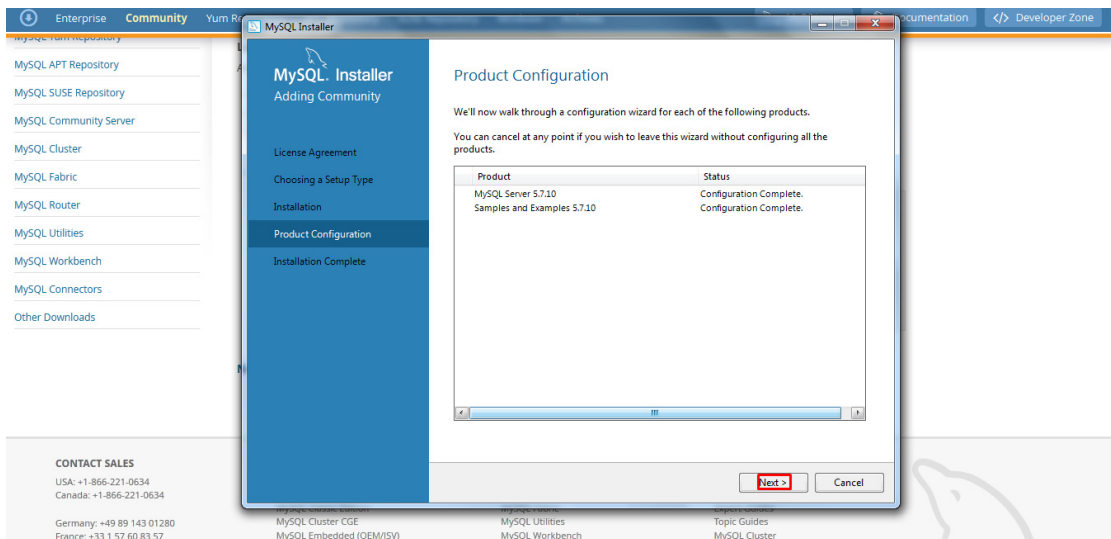
Σχ 6.2.23 Εισάγουμε τον κωδικό πρόσβασης και έπειτα επιλέγουμε «Next».



Σχ 6.2.24 Επιλέγουμε «Execute».

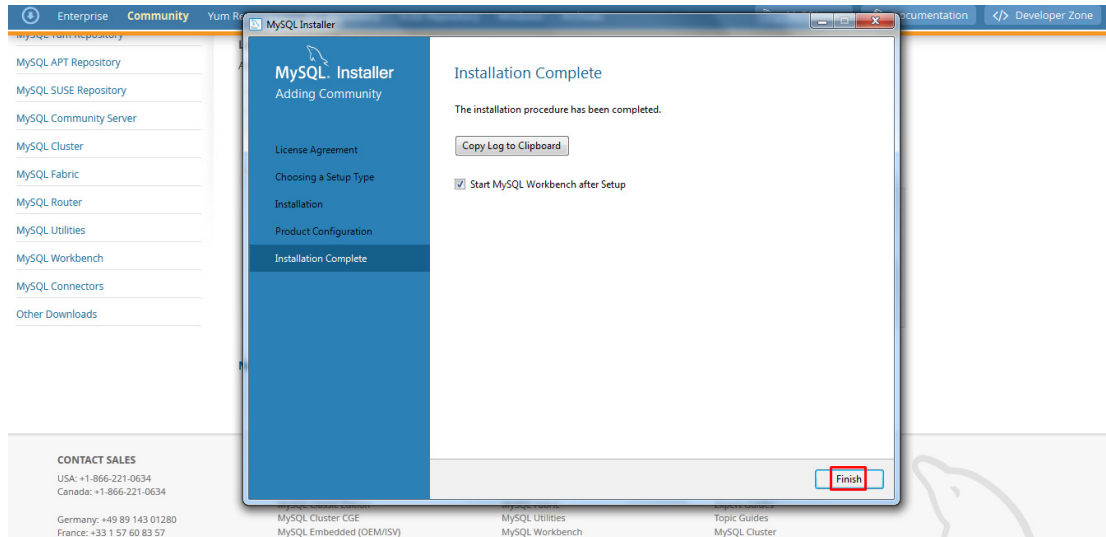


Σχ 6.2.25 Μόλις ολοκληρωθεί η διαδικασία επιλέγουμε το «Finish».

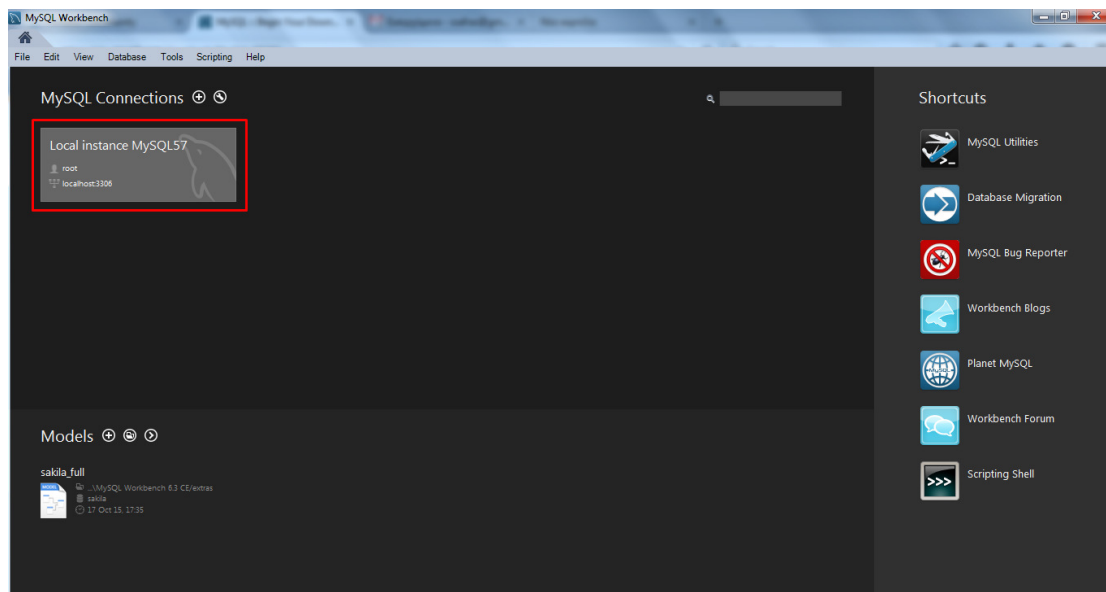


Σχ 6.2.26 Επιλέγουμε «Next».

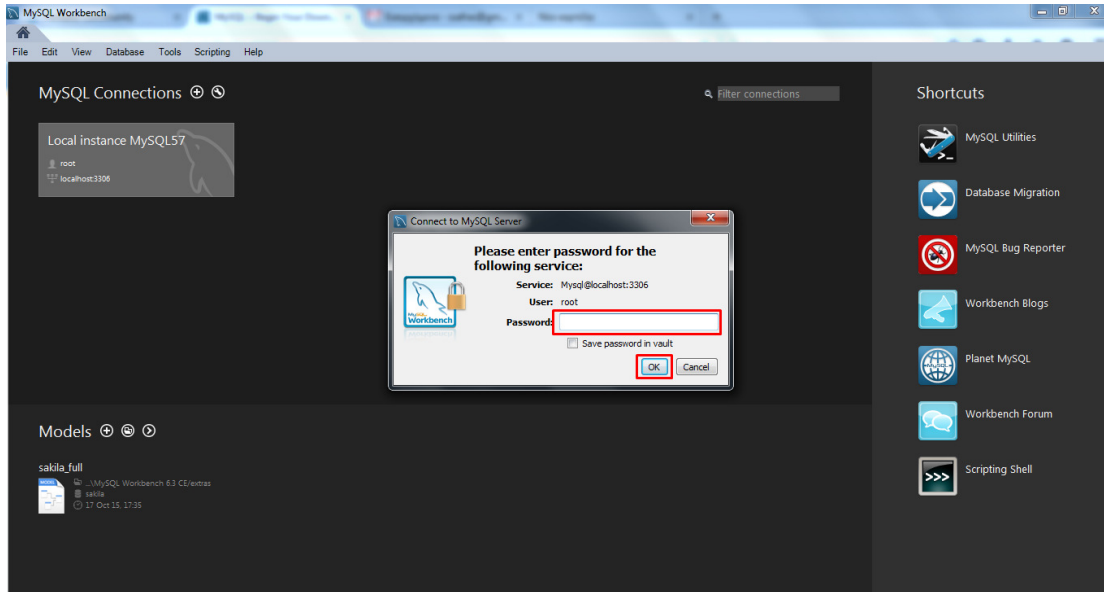




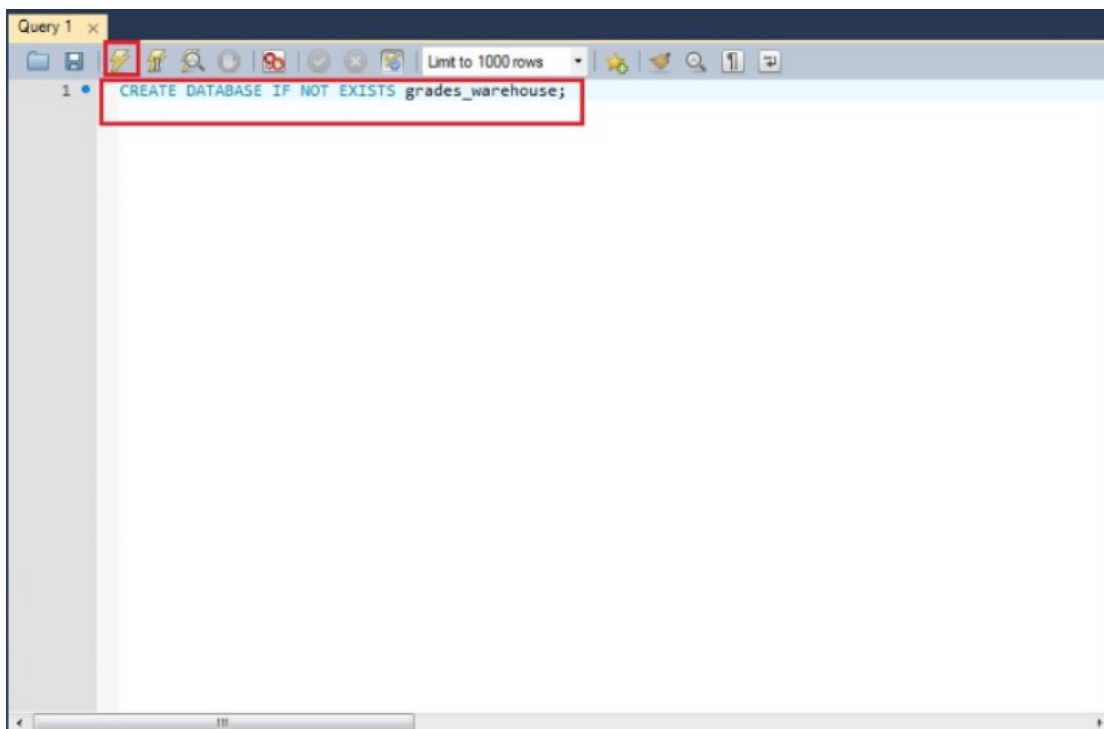
Σχ 6.2.27 Επιλέγουμε «Finish».



Σχ 6.2.28 Στο παράθυρο που θα εμφανιστεί επιλέγουμε το «Local instance MySQL57».



Σχ 6.2.29 Εισάγουμε τον κωδικό που είχαμε καθορίσει και έπειτα επιλέγουμε «OK».



Σχ 6.2.30 Δημιουργούμε την βάση δεδομένων μας.

```
Query 1 x
Limit to 1000 rows

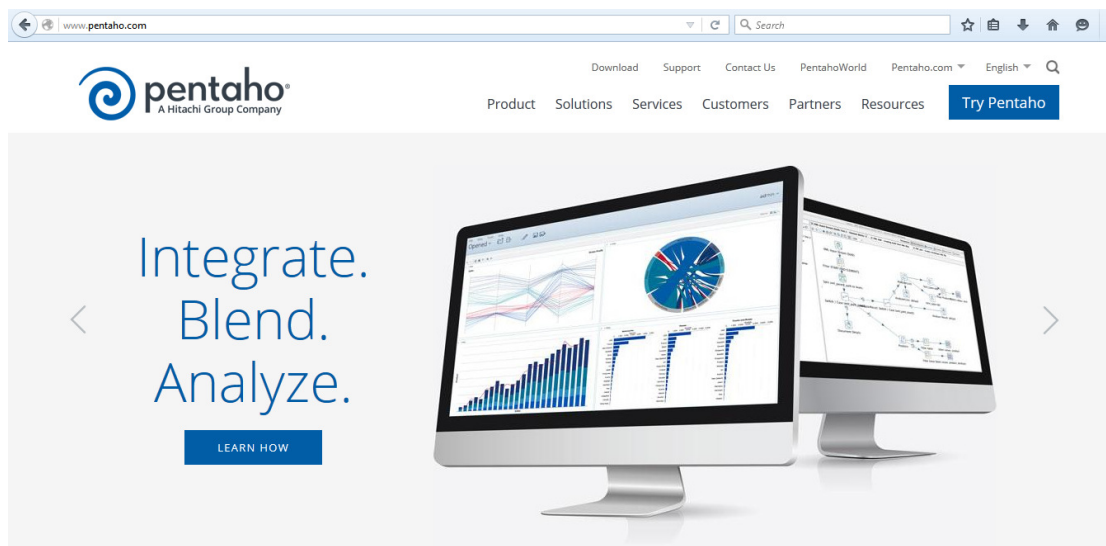
1 CREATE TABLE Years
2 (
3   yearDate YEAR NOT NULL,
4   PRIMARY KEY (yearDate)
5 );
6
7 CREATE TABLE Classes
8 (
9   class VARCHAR(10) NOT NULL,
10  PRIMARY KEY (class)
11 );
12
13 CREATE TABLE Courses
14 (
15   course VARCHAR(10) NOT NULL,
16   PRIMARY KEY (course)
17 );
18
19 CREATE TABLE Curriculums
20 (
21   curriculum VARCHAR(10) NOT NULL,
22   PRIMARY KEY (curriculum)
23 );
24
25 CREATE TABLE Departments
26 (
27   department VARCHAR(5) NOT NULL,
28   PRIMARY KEY (department)
29 );
30
31 CREATE TABLE Professors
32 (
33   professor VARCHAR(70) NOT NULL,
34   PRIMARY KEY (professor)
35 );
36
37 CREATE TABLE Professors_grades
38 (
39   Grade_ID BIGINT NOT NULL,
40   yearDate YEAR NOT NULL,
41   department VARCHAR(5) NOT NULL,
42   curriculum VARCHAR(10) NOT NULL,
43   course VARCHAR(10) NOT NULL,
44   class VARCHAR(10) NOT NULL,
45   professor VARCHAR(70) NOT NULL,
46   grade DECIMAL(10,2) NOT NULL,
47   PRIMARY KEY (Grade_ID),
48   FOREIGN KEY (yearDate) REFERENCES Years(yearDate),
49   FOREIGN KEY (department) REFERENCES Departments(department),
50   FOREIGN KEY (curriculum) REFERENCES Curriculums(curriculum),
51   FOREIGN KEY (course) REFERENCES Courses(course),
52   FOREIGN KEY (class) REFERENCES Classes(class),
53   FOREIGN KEY (professor) REFERENCES Professors(professor)
54 );
```

Σχ 6.2.31 Δημιουργούμε τους πίνακες της βάσης δεδομένων.

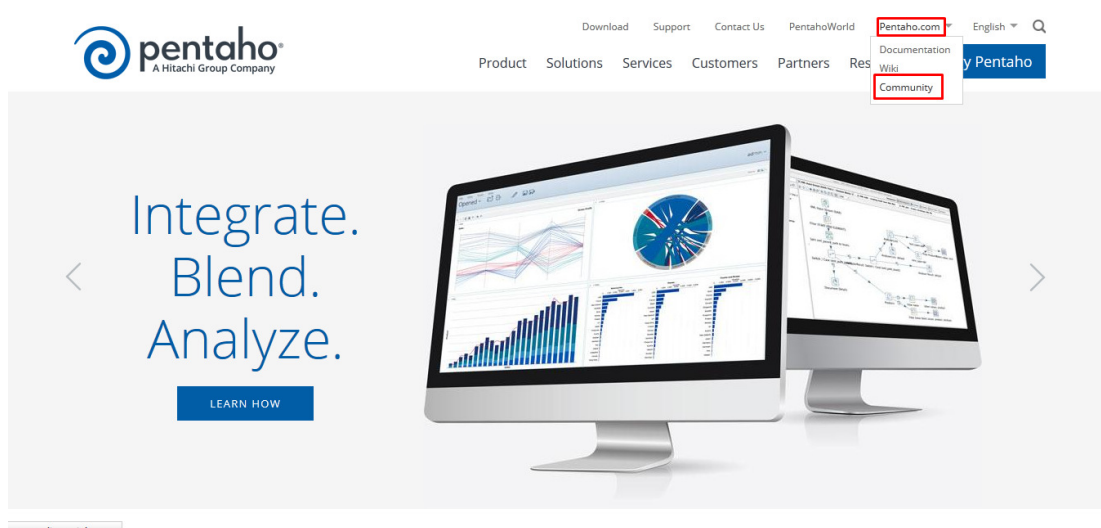
```
Query 1 x
Limit to 1000 rows
1 • LOAD DATA LOCAL INFILE 'C:/years.csv'
2 INTO TABLE years
3 FIELDS TERMINATED BY ','
4 LINES TERMINATED BY '\n';
5
6 • LOAD DATA LOCAL INFILE 'C:/class.csv'
7 INTO TABLE classes
8 FIELDS TERMINATED BY ','
9 LINES TERMINATED BY '\n';
10
11 • LOAD DATA LOCAL INFILE 'C:/course.csv'
12 INTO TABLE courses
13 FIELDS TERMINATED BY ','
14 LINES TERMINATED BY '\n';
15
16 • LOAD DATA LOCAL INFILE 'C:/curriculum.csv'
17 INTO TABLE curriculums
18 FIELDS TERMINATED BY ','
19 LINES TERMINATED BY '\n';
20
21 • LOAD DATA LOCAL INFILE 'C:/department.csv'
22 INTO TABLE departments
23 FIELDS TERMINATED BY ','
24 LINES TERMINATED BY '\n';
25
26 • LOAD DATA LOCAL INFILE 'C:/professor.csv'
27 INTO TABLE professors
28 FIELDS TERMINATED BY ','
29 LINES TERMINATED BY '\n';
30
31 • LOAD DATA LOCAL INFILE 'C:/data.csv'
32 INTO TABLE professors_grades
33 FIELDS TERMINATED BY ','
34 LINES TERMINATED BY '\n';
```

Σχ 6.2.32 Προσθέτουμε τα δεδομένα στους πίνακες.

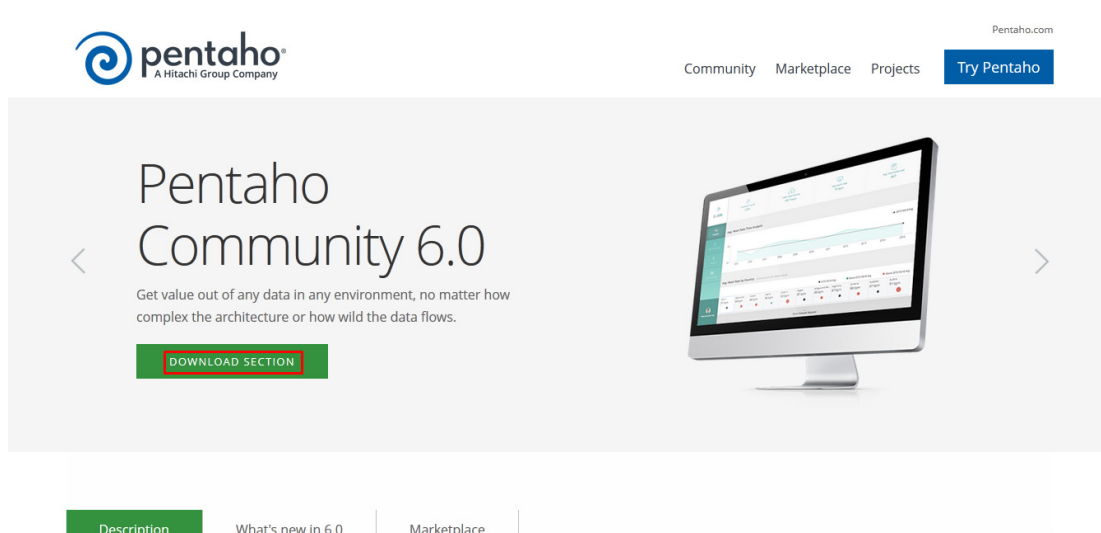
Τώρα που εγκαταστήσαμε το Workbench και δημιουργήσαμε τη βάση δεδομένων μας στην SQL θα συνεχίσουμε με την εγκατάσταση του Pentaho.



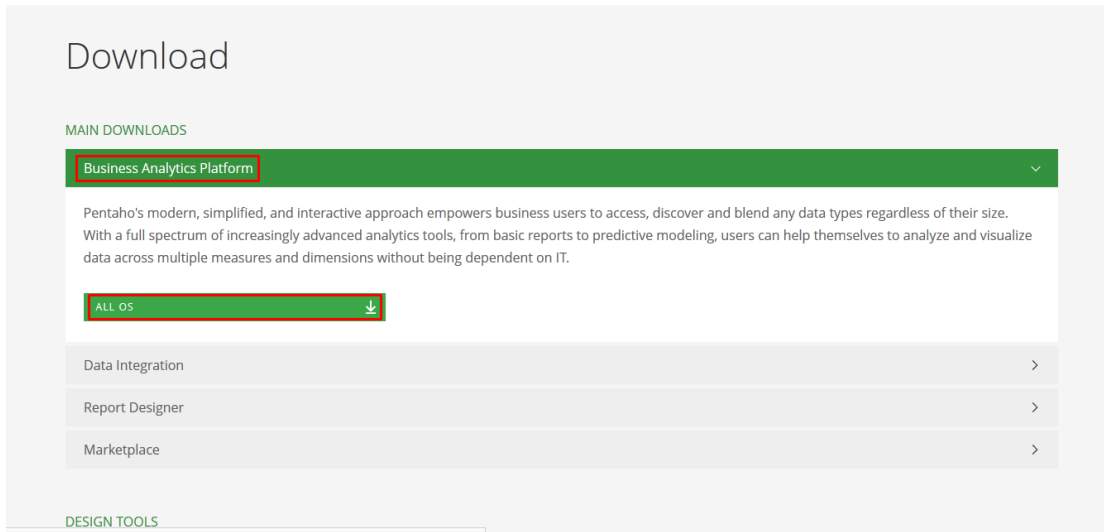
Σχ 6.2.33 Μπαίνουμε στο site [www.pentaho.com](http://www.pentaho.com)



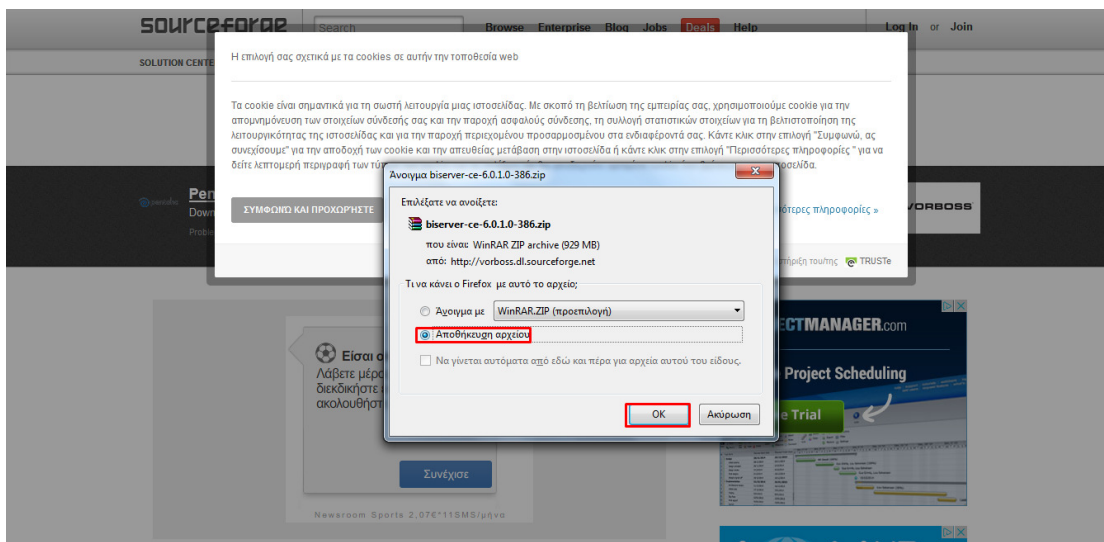
Σχ 6.2.34 Επιλέγουμε το Pentaho.com και έπειτα επιλέγουμε «Community».



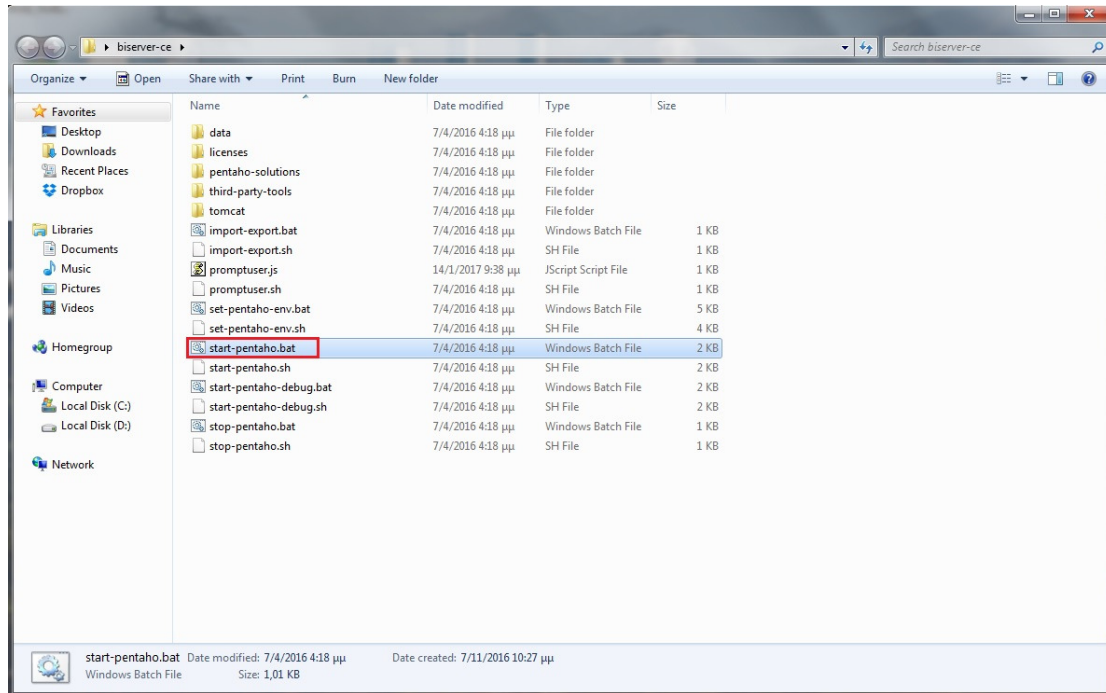
Σχ 6.2.35 Επιλέγουμε «DOWNLOAD SECTION».



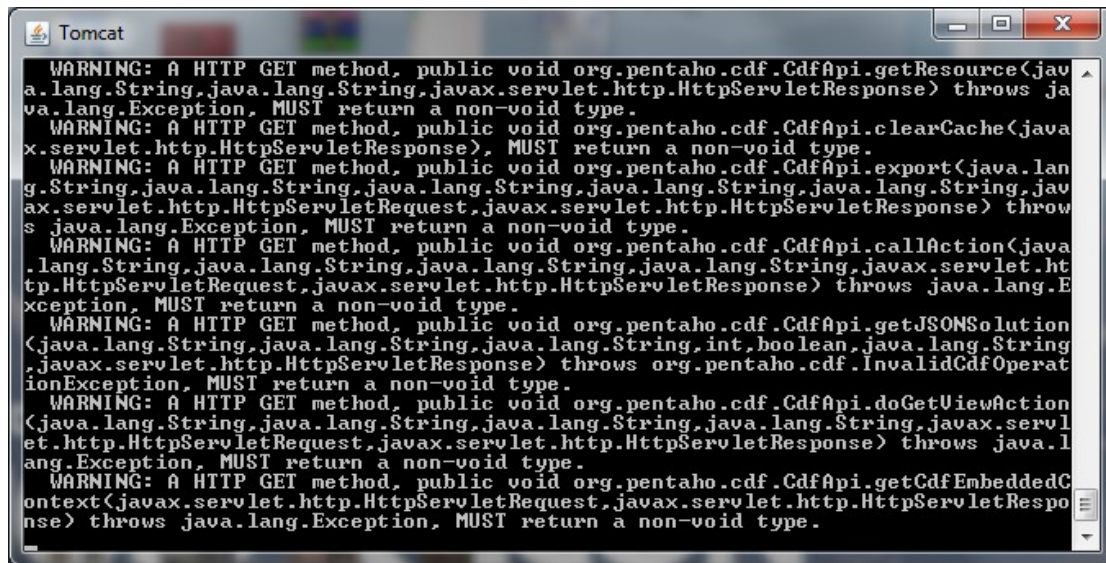
Σχ 6.2.36 Επιλέγουμε στο MAIN DOWNLOADS το Business Analytics Platform και έπειτα ALL OS.



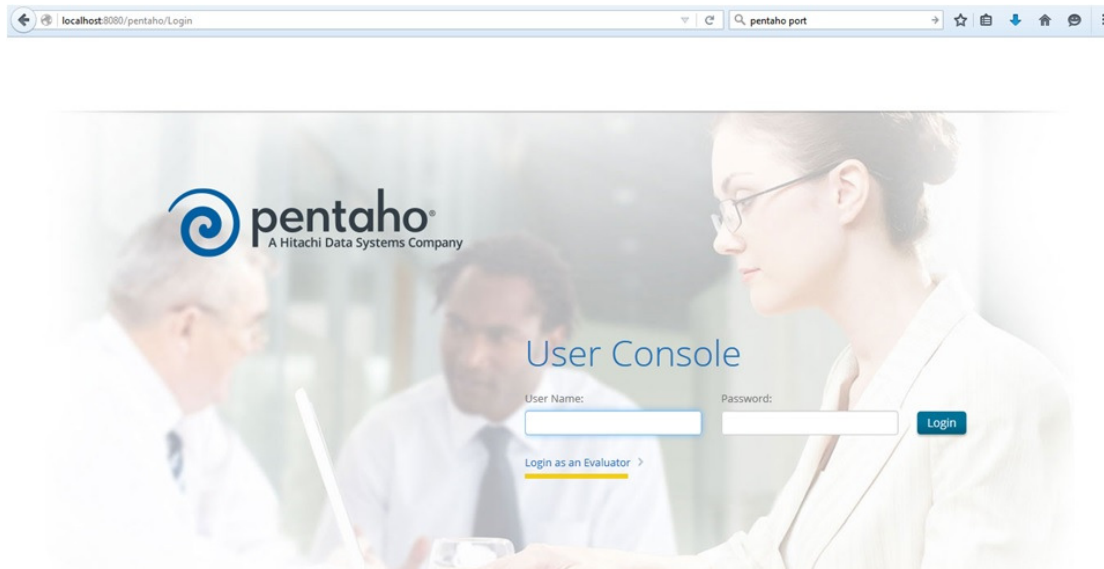
Σχ 6.2.37 Στο παράθυρο που θα εμφανιστεί επιλέγουμε «Αποθήκευση αρχείου» και έπειτα «OK».



Σχ 6.2.38 Βρίσκουμε το φάκελο που αποθηκεύσαμε στον Υπολογιστή και επιλέγουμε το αρχείο «start-pentaho.bat».



Σχ 6.2.39 Περιμένουμε τον server να ξεκινήσει.

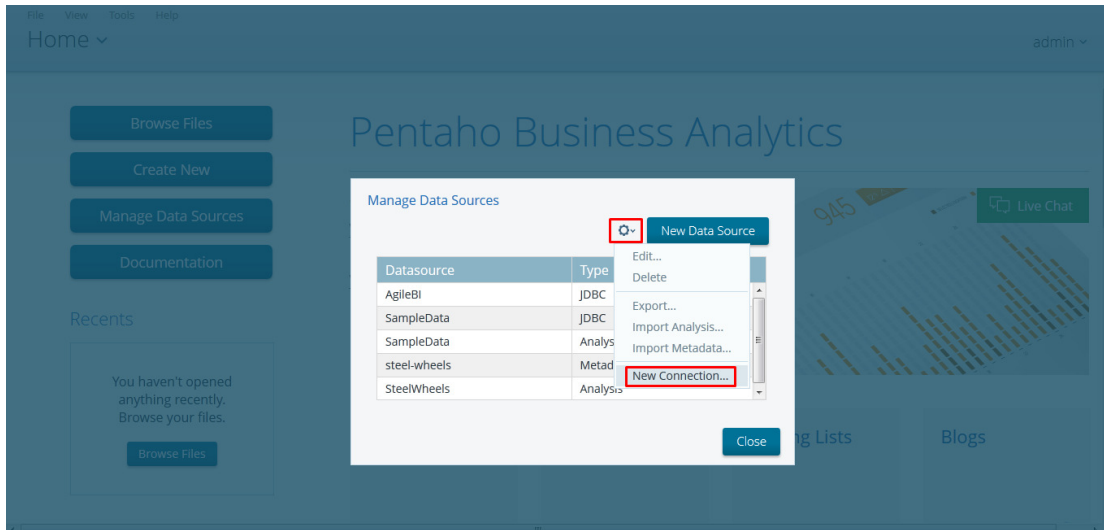


Σχ 6.2.40 Μόλις ολοκληρωθεί η διαδικασία ανοίγουμε έναν browser και εισάγουμε τη διεύθυνση localhost:8080. Στο User Name: πληκτρολογούμε admin και στο Password: password και επιλέγουμε το «Login».

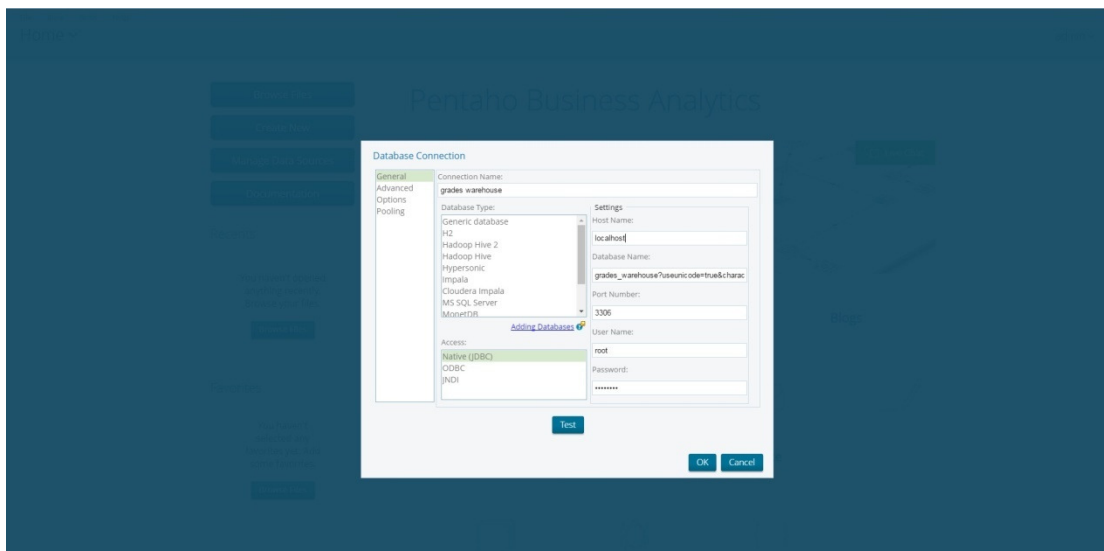


Σχ 6.2.41 Επιλέγουμε «Manage Data Sources».

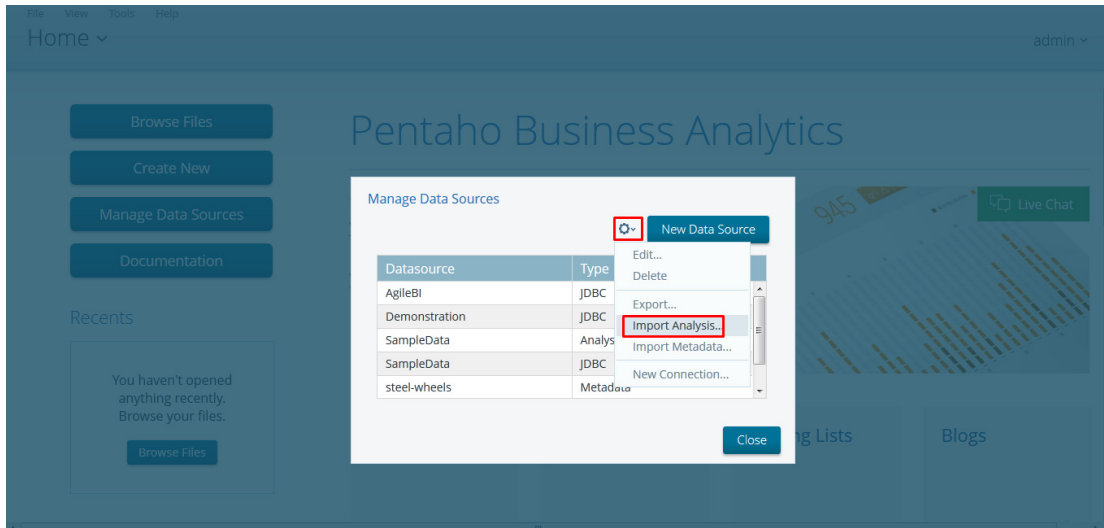




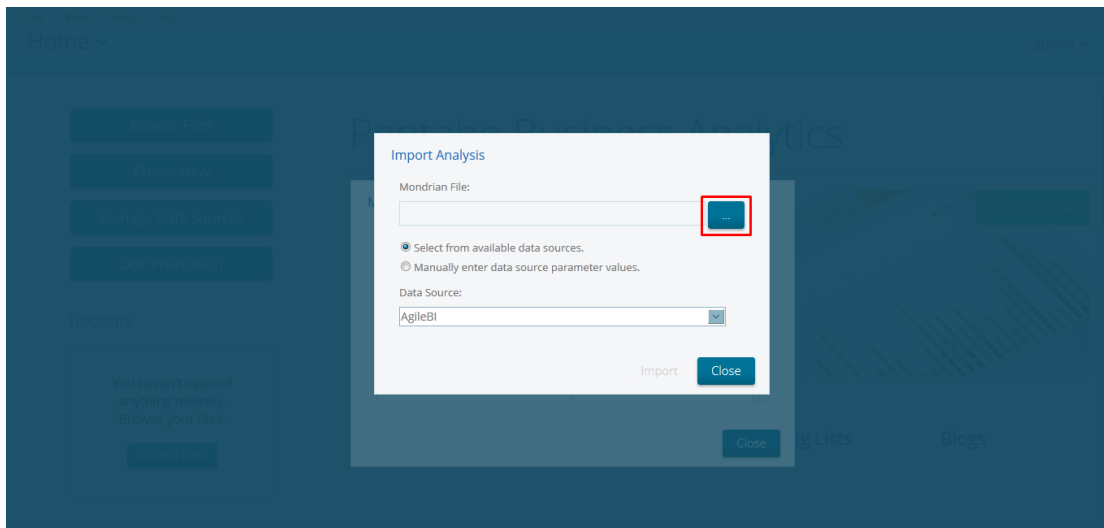
Σχ 6.2.42 Επιλέγουμε το γρανάτζι και έπειτα «New Connection».



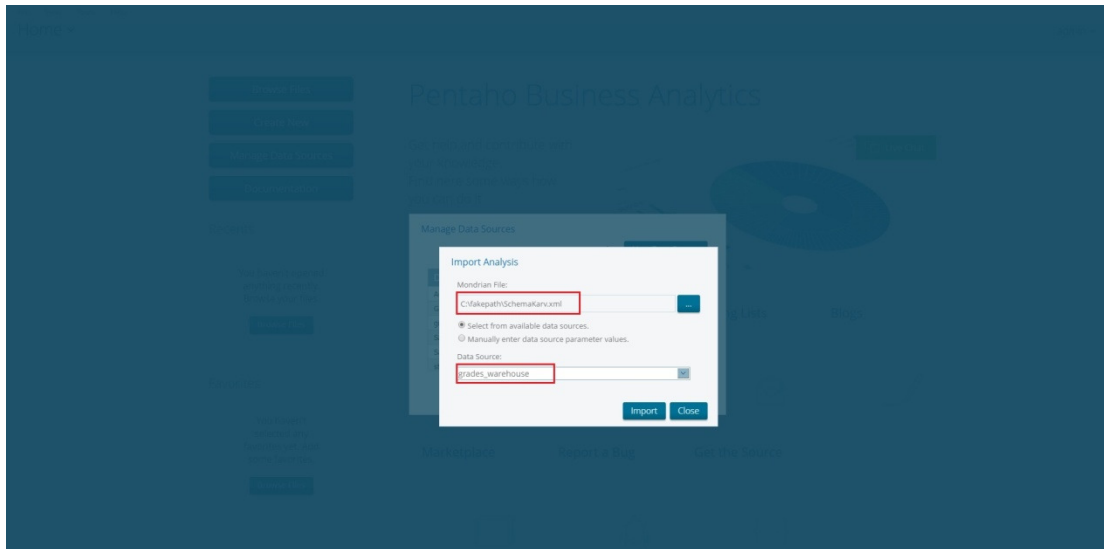
Σχ 6.2.43 Πληκτρολογούμε ένα connection name και στο Data base name πληκτρολογούμε «το όνομα που επιλέξαμε?useunicode=true&character Encoding=utf8».



Σχ 6.2.44 Επιλέξουμε το γρανάτζι και έπειτα «Import Analysis».



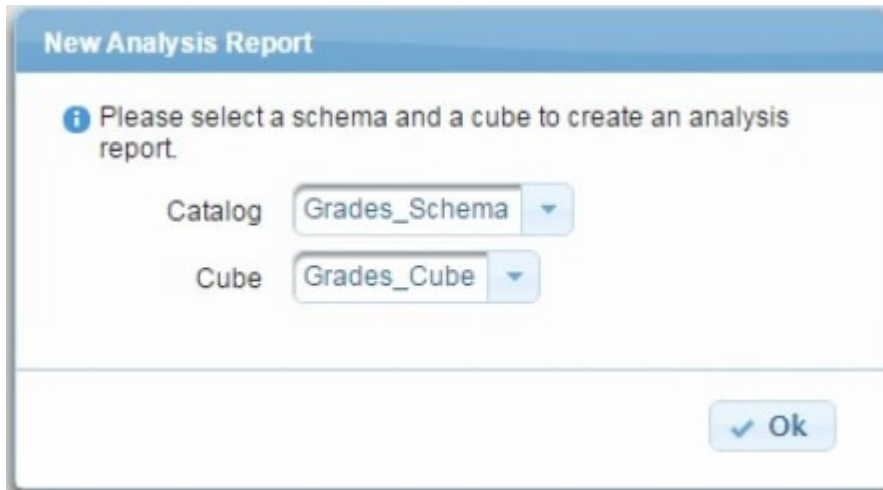
Σχ 6.2.45 Επιλέγουμε τα αποσιωπητικά και επιλέγουμε το Mondrian File μας.



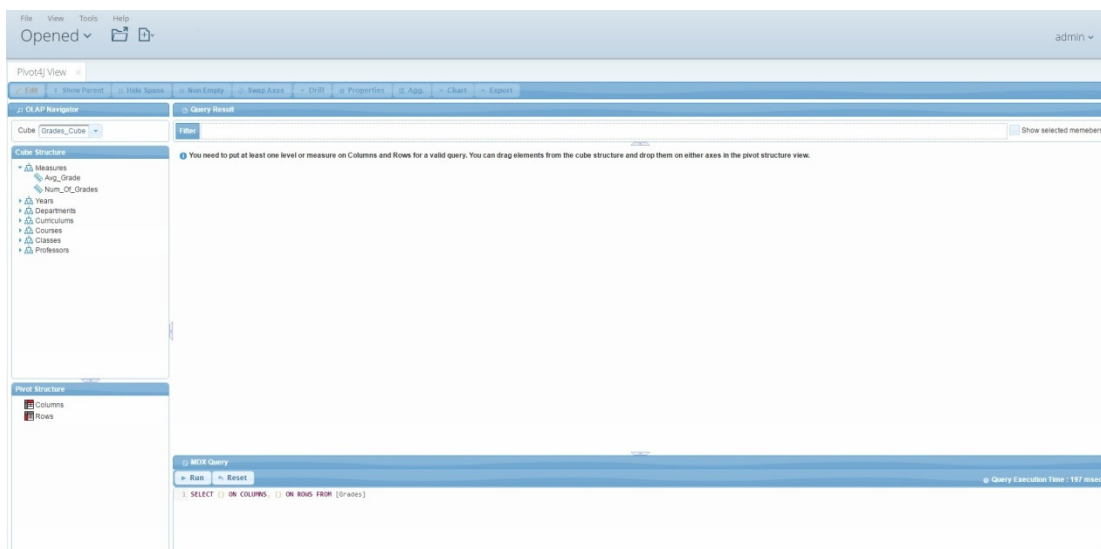
Σχ 6.2.46 Στο Data Source επιλέγουμε την βάση δεδομένων μας και πατάμε «Import».



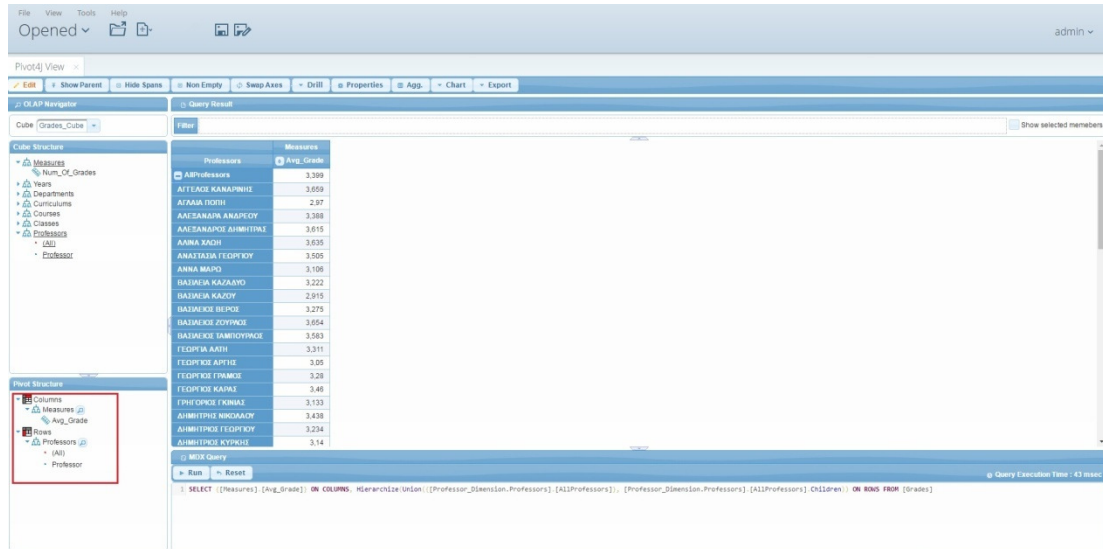
Σχ 6.2.47 Επιλέγουμε «Create New» και έπειτα «Pivot4J View».



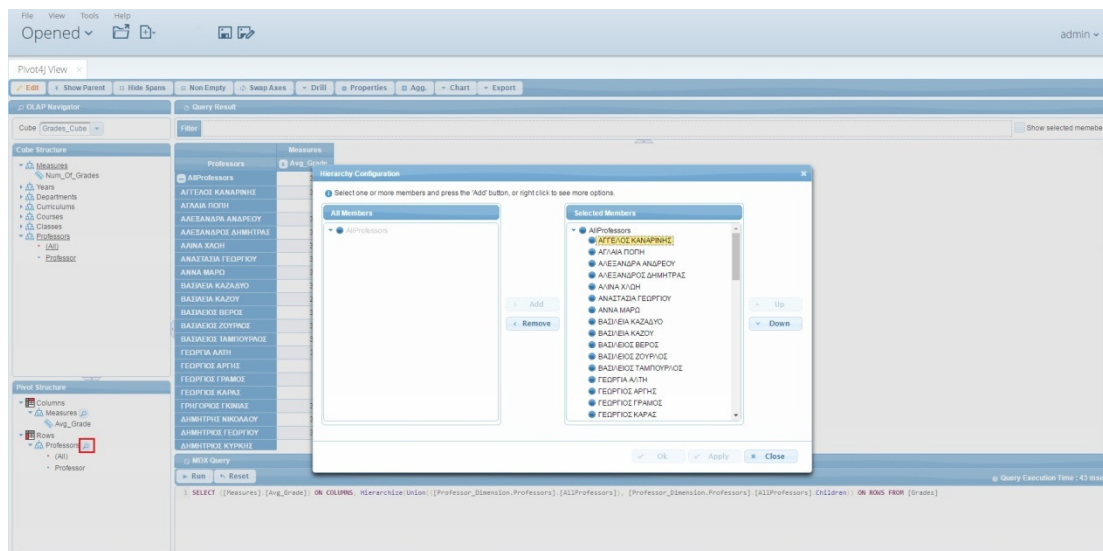
Σχ 6.2.48 Επιλέγουμε το σχήμα που φτιάξαμε και τον αντίστοιχο κύβο.



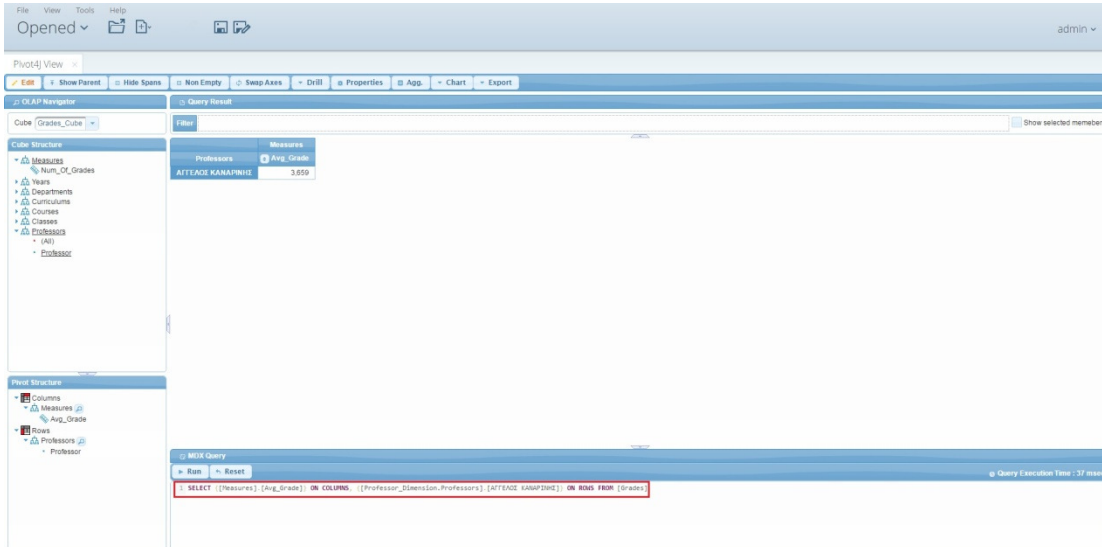
Σχ 6.2.49 Τώρα έχουμε συνδέσει την βάση δεδομένων με το Pentaho και είμαστε έτοιμοι να κάνουμε ερωτήματα.



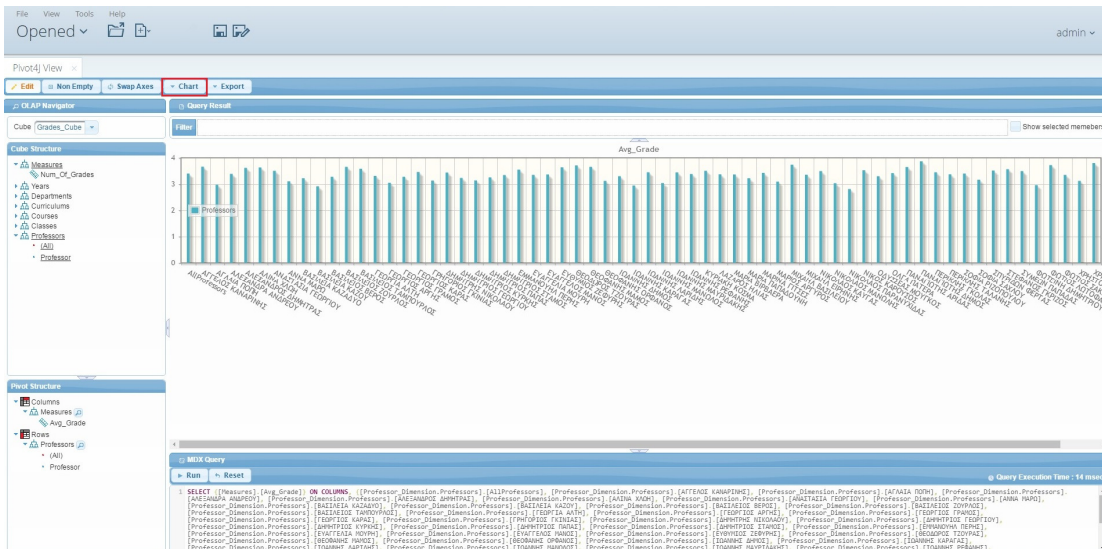
Σχ 6.2.50 Τώρα μπορούμε να επιλέξουμε διαστάσεις από τη λίστα επάνω αριστερά και να τα καταχωρήσουμε σε στήλες και γραμμές.



Σχ 6.2.51 Πατώντας το μεγεθυντικό φακό δίπλα από κάθε διάσταση μπορούμε να βάλουμε φίλτρα.



Σχ 6.2.52 Στο πλαίσιο κάτω βλέπουμε το ερώτημα που έχουμε κάνει σε γλώσσα MDX.



Σχ 6.2.53 Με το κουμπί «Charts» μπορούμε να δημιουργήσουμε διαγράμματα για το σχήμα μας.

### 6.3 Σχήμα OLAP

Σε αυτή την ενότητα θα σχεδιάσουμε το σχήμα από τα δεδομένα που έχουμε. Η βάση δεδομένων χρησιμοποιεί το σχήμα snowflake και αποτελείται από έξι διαστάσεις και μία μέτρηση.

Παρακάτω θα δούμε τις παραμέτρους που χρησιμοποιήθηκαν για να δημιουργήσουμε το σχήμα.

#### **Grade ID:**

Σε αυτό το πεδίο έχουμε την αρίθμηση των εγγραφών που έχουμε στη βάση δεδομένων.

**yearDate:**

Σε αυτό το πεδίο βλέπουμε τη χρονιά για την οποία έχει βαθμολογηθεί ο κάθε καθηγητής. Οι χρονιές που θα εξετάσουμε είναι από το 2012 έως το 2016.

**department:**

Σε αυτό το πεδίο βλέπουμε τη σχολή του ανοιχτού πανεπιστημίου την οποία εξετάζουμε όπου περιέχει κάποια προγράμματα σπουδών.

**curriculum:**

Σε αυτό το πεδίο βλέπουμε το πρόγραμμα σπουδών που εξετάζουμε.

**course:**

Σε αυτό το πεδίο βλέπουμε το μάθημα ενός συγκεκριμένου προγράμματος σπουδών.

**class:**

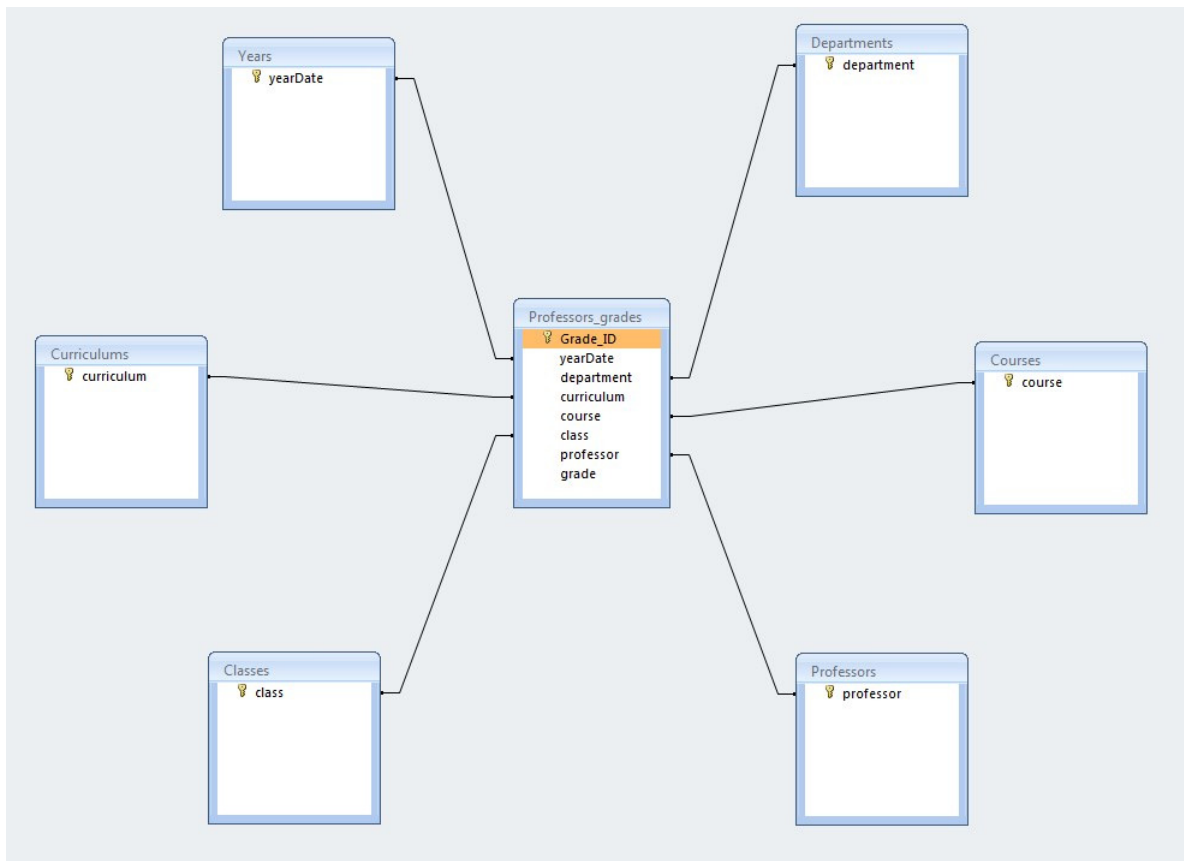
Σε αυτό το πεδίο βλέπουμε την τάξη του κάθε μαθήματος.

**professor:**

Σε αυτό το πεδίο βλέπουμε το ονοματεπώνυμο του καθηγητή που βαθμολογήθηκε.

**grade:**

Σε αυτό το πεδίο βλέπουμε τους βαθμούς με τους οποίους αξιολόγησαν οι φοιτητές τους καθηγητές. Οι βαθμολογίες αυτές είναι ένας μέσος όρος που έχει βγει από τους βαθμούς που έχουν βάλει οι φοιτητές σε όλο το εκπαιδευτικό έργο των καθηγητών και ορίζονται από ένα έως πέντε.



Σχ 6.3.1 Το σχήμα και οι σχέσεις των πινάκων.

## 6.4 Παραδείγματα χρήσης OLAP

Σε αυτό το κεφάλαιο θα δούμε κάποια παραδείγματα χρήσης του OLAP με το εργαλείο Pentaho. Η βάση δεδομένων με την οποία θα ασχοληθούμε είναι μια βάση δεδομένων με τις βαθμολογίες που έχουν οι καθηγητές βάση των αποδόσεων τους στο ανοιχτό πανεπιστήμιο για τις χρονιές 2012 έως 2016. Η συγκεκριμένη βάση δημιουργήθηκε σε MySQL και μετά συνδέθηκε με το Pentaho με ένα Mondrian XML σχήμα. Το εργαλείο που χρησιμοποιήσαμε για τα ερωτήματα είναι το Pivot4j. Το Pivot4j είναι ένα πρόσθετο για το διακομιστή του Pentaho που επεκτείνει το βασικό λογισμικό το οποίο μπορεί να χρησιμοποιηθεί και από μόνο του σαν ένα πλήρες πρόγραμμα ανάλυσης.[11]

Έχοντας φτιάξει τον κύβο μας στο Pentaho μπορούμε πλέον να επιλέξουμε ποιες διαστάσεις θέλουμε να δούμε σε κάθε άξονα και να βάλουμε διάφορα κριτήρια. Για παράδειγμα στο σχήμα 6.4.1 έχουμε βάλει απλά την διάσταση "**Professors**" στις σειρές και τους βαθμούς στις γραμμές. (Επειδή το αποτέλεσμα είναι αρκετά μεγάλο έχουμε κρατήσει μόνο τα 7 πρώτα ονόματα αλφαβητικά.)

	Measures
Professors	Avg_Grade
AllProfessors	3,399
ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	3,659
ΑΓΛΑΙΑ ΠΟΠΗ	2,97
ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	3,388
ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ	3,615
ΑΛΙΝΑ ΧΛΩΗ	3,635
ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ	3,505
ΑΝΝΑ ΜΑΡΩ	3,106

Σχ 6. 4.1 Οι βαθμοί των 7 πρώτων καθηγητών.

Τώρα μπορούμε να προσθέσουμε και άλλες διαστάσεις για να δούμε πιο συγκεκριμένα στοιχεία. Για παράδειγμα μπορούμε να προσθέσουμε την διάσταση "Years" για να δούμε αυτούς τους βαθμούς ανά χρονιά.



2012-01-01	AllProfessors	3,68
	ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	3,896
	ΑΓΛΑΙΑ ΠΟΠΗ	3,218
	ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	4,198
	ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ	4,096
	ΑΛΙΝΑ ΧΛΩΗ	4,028
	ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ	3,547
	ΑΝΝΑ ΜΑΡΩ	3,813
2013-01-01	AllProfessors	3,068
	ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	3,192
	ΑΓΛΑΙΑ ΠΟΠΗ	2,429
	ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	3,283
	ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ	3,547
	ΑΛΙΝΑ ΧΛΩΗ	3,125
	ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ	2,96
	ΑΝΝΑ ΜΑΡΩ	3,274
2014-01-01	AllProfessors	3,504
	ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	3,742
	ΑΓΛΑΙΑ ΠΟΠΗ	3,11
	ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	3,224
	ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ	3,541
	ΑΛΙΝΑ ΧΛΩΗ	3,692
	ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ	3,853
	ΑΝΝΑ ΜΑΡΩ	2,952

Σχ 6. 4.2 Οι βαθμοί των 7 πρώτων καθηγητών για τις χρονιές 2012,2013,2014.

Με την ίδια λογική μπορούμε να συνεχίσουμε να προσθέτουμε διαστάσεις και να ειδικεύουμε και άλλο τα δεδομένα μας. Μπορούμε όμως να αλλάξουμε και τον τρόπο με τον οποίο παρουσιάζονται τα δεδομένα για να τα εξετάσουμε διαφορετικά. Στο σχήμα 6.4.2 βλέπουμε τους βαθμούς που έχουν όλοι οι καθηγητές για κάθε χρόνο. Με μια απλή αλλαγή μπορούμε να δούμε στο σχήμα 6.4.3 τους βαθμούς που έχει για όλες τις χρονιές ο κάθε καθηγητής. (Επειδή το αποτέλεσμα είναι αρκετά μεγάλο έχουμε κρατήσει μόνο τα 4 πρώτα ονόματα αλφαβητικά.)

ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	AllYears	3,659
	2012-01-01	3,896
	2013-01-01	3,192
	2014-01-01	3,742
	2015-01-01	3,317
	2016-01-01	4,279
ΑΓΛΑΙΑ ΠΟΠΗ	AllYears	2,97
	2012-01-01	3,218
	2013-01-01	2,429
	2014-01-01	3,11
	2015-01-01	2,403
	2016-01-01	4,018
ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	AllYears	3,388
	2012-01-01	4,198
	2013-01-01	3,283
	2014-01-01	3,224
	2015-01-01	2,712
	2016-01-01	3,892
ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ	AllYears	3,615
	2012-01-01	4,096
	2013-01-01	3,547
	2014-01-01	3,541
	2015-01-01	3,214
	2016-01-01	4,002

Σχ 6.4.3 Οι βαθμοί των για όλες τις χρονιές των καθηγητών ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ, ΑΓΛΑΙΑ ΠΟΠΗ, ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ, ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ.

Δεν μπορούμε όμως μόνο να προσθέτουμε διαστάσεις και να αλλάζουμε τον τρόπο παρουσίασης τους. Μπορούμε επίσης να προσθέσουμε και διάφορα κριτήρια ώστε να εξετάσουμε μόνο τις εγγραφές που μας ενδιαφέρουν. Για παράδειγμα στα δεδομένα που έχουμε ήδη από το σχήμα 6.4.3, μπορούμε να ζητήσουμε μόνο τις εγγραφές που έχουν βαθμό πάνω από 4.

		Measures
Professors	Years	Avg_Grade
ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	2016-01-01	4,279
ΑΓΛΑΙΑ ΠΟΠΗ	2016-01-01	4,018
ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	2012-01-01	4,198
ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ	2016-01-01	4,002
	2012-01-01	4,096
ΑΛΙΝΑ ΧΛΩΗ	2016-01-01	4,246
	2012-01-01	4,028
ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ	2016-01-01	4,425

Σχ 6. 4.3 Οι βαθμοί για όλες τις χρονιές των 7 πρώτων καθηγητών που έχουν βαθμό πάνω από 4.

Στο σχήμα 6.4.3 παρατηρούμε ότι εμφανίζονται μόνο έξι από τους επτά πρώτους καθηγητές και αυτό γίνεται διότι απλά ο έβδομος δεν έχει καμία χρονιά βαθμό πάνω από 4.

Στα παρακάτω σχήματα βλέπουμε κάποια άλλα ερωτήματα για την ίδια βάση δεδομένων.

				Measures
Years	Courses	Classes	Professors	Avg_Grade
AllYears	ΠΛΗ10	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	3,28
	ΠΛΗ11	ΑΘΗ2	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	3,583
2012-01-01	ΠΛΗ10	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	4,198
	ΠΛΗ11	ΑΘΗ2	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	3,368
2013-01-01	ΠΛΗ10	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	3,152
	ΠΛΗ11	ΑΘΗ2	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	3,381
2014-01-01	ΠΛΗ10	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	2,959
	ΠΛΗ11	ΑΘΗ2	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	3,865
2015-01-01	ΠΛΗ10	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	2,055
	ΠΛΗ11	ΑΘΗ2	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	2,996
2016-01-01	ΠΛΗ10	ΑΘΗ1	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	4,11
	ΠΛΗ11	ΑΘΗ2	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	4,428

Σχ 6.4.4 Οι βαθμοί των καθηγητών ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ και ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ για κάθε τάξη στα μαθήματα ΠΛΗ10 και ΠΛΗ11 ανά χρόνο.

```

MDX Query
Run Reset Query Execution Time : 268 msec
1 SELECT NON EMPTY {[Measures].[Avg_Grade]} ON COLUMNS, NON EMPTY
CrossJoin(Hierarchize(Union({[Year_Dimension.Years].[AllYears]}, [Year_Dimension.Years].
[AllYears].Children)), CrossJoin({[Course_Dimension.Courses].[ΠΛΗ10],
[Course_Dimension.Courses].[ΠΛΗ11]}, CrossJoin({[Class_Dimension.Classes].[ΑΘΗ1],
[Class_Dimension.Classes].[ΑΘΗ2], [Class_Dimension.Classes].[ΘΕΣ1],
[Class_Dimension.Classes].[ΘΕΣ2], [Class_Dimension.Classes].[ΠΑΤ1]},
{[Professor_Dimension.Professors].[ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ], [Professor_Dimension.Professors].
[ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ]}))) ON ROWS FROM [Grades]

```

Σχ 6.4.5 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.4

		Measures
Professors	Years	Avg_Grade
ΠΕΡΙΚΛΗΣ ΤΑΛΑΝΗΣ	AllYears	3,373
	2012-01-01	3,517
	2014-01-01	3,603
	2016-01-01	4,463
ΣΠΥΡΙΔΩΝ ΦΕΡΤΑΣ	AllYears	3,514
	2014-01-01	3,775
	2012-01-01	3,905
	2016-01-01	4,461
ΧΡΙΣΤΙΝΑ ΚΑΛΦΑΣ	AllYears	3,802
	2013-01-01	3,232
	2015-01-01	3,562
	2012-01-01	3,888
	2014-01-01	4,132
	2016-01-01	4,297

Σχ 6.4.6 Οι βαθμοί των ΠΕΡΙΚΛΗΣ ΤΑΛΑΝΗΣ, ΣΠΥΡΙΔΩΝ ΦΕΡΤΑΣ, ΧΡΙΣΤΙΝΑ ΚΑΛΦΑΣ ανά χρόνο, που είναι πάνω από 3.

MDX Query
Close

▶ Run
↶ Reset
Query Execution Time : 268 msec

```

1 SELECT {[Measures].[Avg_Grade]} ON COLUMNS,
Order(filter(Union(Union(Union(CrossJoin({[Professor_Dimension.Professors].[ΠΕΡΙΚΛΗΣ
ΤΑΛΑΝΗΣ]}, {[Year_Dimension.Years].[AllYears]}),
CrossJoin({[Professor_Dimension.Professors].[ΠΕΡΙΚΛΗΣ ΤΑΛΑΝΗΣ]}, [Year_Dimension.Years].
[AllYears].Children)), Union(CrossJoin({[Professor_Dimension.Professors].[ΣΠΥΡΙΔΩΝ
ΦΕΡΤΑΣ]}, {[Year_Dimension.Years].[AllYears]}),
CrossJoin({[Professor_Dimension.Professors].[ΣΠΥΡΙΔΩΝ ΦΕΡΤΑΣ]}, [Year_Dimension.Years].
[AllYears].Children)), Union(CrossJoin({[Professor_Dimension.Professors].[ΧΡΙΣΤΙΝΑ
ΚΑΛΦΑΣ]}, {[Year_Dimension.Years].[AllYears]}),
CrossJoin({[Professor_Dimension.Professors].[ΧΡΙΣΤΙΝΑ ΚΑΛΦΑΣ]}, [Year_Dimension.Years].
[AllYears].Children))), [Measures].[Avg_Grade] > 3.0), [Measures].[Avg_Grade], ASC) ON
ROWS FROM [Grades]

```

Σχ 6.4.7 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.6

		Measures
Professors	Years	Num_Of_Grades
ΑΓΛΑΙΑ ΠΟΠΗ	AllYears	33
	2012-01-01	5
	2013-01-01	8
	2014-01-01	9
	2015-01-01	6
	2016-01-01	5
ΓΕΩΡΓΙΑ ΑΛΤΗ	AllYears	59
	2012-01-01	9
	2013-01-01	17
	2014-01-01	15
	2015-01-01	10
	2016-01-01	8

Σχ 6.4.8 Πόσες φορές έχουν βαθμολογηθεί οι καθηγήτριες ΑΓΛΑΙΑ ΠΟΠΗ και ΓΕΩΡΓΙΑ ΑΛΤΗ σε κάθε χρόνο, και για όλα τα χρόνια συνολικά.

```

MDX Query
Run Reset Query Execution Time : 268 msec
1 SELECT NON EMPTY {[Measures].[Num_Of_Grades]} ON COLUMNS, NON EMPTY
CrossJoin({[Professor_Dimension.Professors].[ΑΓΛΑΙΑ ΠΟΠΗ], [Professor_Dimension.Professors].[ΓΕΩΡΓΙΑ ΑΛΤΗ]}, {[Year_Dimension.Years].[AllYears], [Year_Dimension.Years].[2012-01-01], [Year_Dimension.Years].[2013-01-01], [Year_Dimension.Years].[2014-01-01], [Year_Dimension.Years].[2015-01-01], [Year_Dimension.Years].[2016-01-01]}) ON ROWS FROM [Grades]

```

Σχ 6.4.9 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.8

		Measures
Professors	Years	Avg_Grade
ΓΕΩΡΓΙΑ ΑΛΤΗ	AllYears	3,311
	2015-01-01	2,645
	2014-01-01	3,039
	2013-01-01	3,493
	2016-01-01	3,679
	2012-01-01	3,836

Σχ 6.4.10 Ο μέσος όρος των βαθμών της ΓΕΩΡΓΙΑ ΑΛΤΗ για κάθε χρόνο ταξινομημένα από το μικρότερο προς το μεγαλύτερο.

```

MDX Query
Run Reset Query Execution Time : 268 msec
1 SELECT NON EMPTY {[Measures].[Avg_Grade]} ON COLUMNS, NON EMPTY
Order(CrossJoin({[Professor_Dimension.Professors].[ΓΕΩΡΓΙΑ ΑΛΤΗ]}, {[Year_Dimension.Years].[AllYears], [Year_Dimension.Years].[2012-01-01], [Year_Dimension.Years].[2013-01-01], [Year_Dimension.Years].[2014-01-01], [Year_Dimension.Years].[2015-01-01], [Year_Dimension.Years].[2016-01-01]}), [Measures].[Avg_Grade], ASC) ON ROWS FROM [Grades]

```

Σχ 6.4.11 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.10

			Measures
Years	Departments	Professors	Avg_Grade
2016-01-01	ΚΕ	ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	4,279
		ΑΛΙΝΑ ΧΛΩΗ	4,246
		ΓΕΩΡΓΙΟΣ ΑΡΓΗΣ	3,351
		ΓΡΗΓΟΡΙΟΣ ΓΚΙΝΙΑΣ	4,105
		ΔΗΜΗΤΡΙΟΣ ΓΕΩΡΓΙΟΥ	4,015
		ΔΗΜΗΤΡΙΟΣ ΚΥΡΚΗΣ	3,847
		ΔΗΜΗΤΡΙΟΣ ΠΑΠΑΣ	3,409
		ΔΗΜΗΤΡΙΟΣ ΣΤΑΜΟΣ	3,847
		ΕΜΜΑΝΟΥΗΛ ΠΕΡΗΣ	3,93
		ΕΥΑΓΓΕΛΟΣ ΜΑΝΟΣ	4,033
		ΘΕΟΦΑΝΗΣ ΜΑΜΟΣ	3,805
		ΙΩΑΝΝΗΣ ΚΑΡΑΓΑΣ	3,036
		ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ	3,159
		ΚΥΡΙΑΚΗ ΚΟΣΜΑ	4,207
		ΜΙΧΑΗΛ ΕΙΡΗΝΗΣ	3,938
		ΠΑΝΑΓΙΩΤΗΣ ΑΡΙΔΑΣ	4,192
		ΠΕΡΙΚΛΗΣ ΓΚΟΛΑΣ	4,062
		ΣΥΜΕΩΝ ΠΑΠΑΔΑΣ	3,301
		ΦΩΤΙΟΣ ΣΑΚΗΣ	4,195
ΧΡΗΣΤΟΣ ΠΙΤΑΣ	3,501		

Σχ 6.4.12 Οι βαθμοί όλων των καθηγητών της σχολής ΚΕ την χρονιά 2016

```

MDX Query
▶ Run  ◀ Reset  Query Execution Time : 268 msec

1 SELECT NON EMPTY {[Measures].[Avg_Grade]} ON COLUMNS, NON EMPTY
CrossJoin({[Year_Dimension.Years].[2016-01-01]},
CrossJoin({[Department_Dimension.Departments].[ΚΕ]}, {[Professor_Dimension.Professors].
[ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ], [Professor_Dimension.Professors].[ΑΛΙΝΑ ΧΛΩΗ],
[Professor_Dimension.Professors].[ΓΕΩΡΓΙΟΣ ΑΡΓΗΣ], [Professor_Dimension.Professors].
[ΓΡΗΓΟΡΙΟΣ ΓΚΙΝΙΑΣ], [Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΓΕΩΡΓΙΟΥ],
[Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΚΥΡΚΗΣ], [Professor_Dimension.Professors].
[ΔΗΜΗΤΡΙΟΣ ΠΑΠΑΣ], [Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΣΤΑΜΟΣ],
[Professor_Dimension.Professors].[ΕΜΜΑΝΟΥΗΛ ΠΕΡΗΣ], [Professor_Dimension.Professors].
[ΕΥΑΓΓΕΛΟΣ ΜΑΝΟΣ], [Professor_Dimension.Professors].[ΘΕΟΦΑΝΗΣ ΜΑΜΟΣ],
[Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΚΑΡΑΓΑΣ], [Professor_Dimension.Professors].
[ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ], [Professor_Dimension.Professors].[ΚΥΡΙΑΚΗ ΚΟΣΜΑ],
[Professor_Dimension.Professors].[ΜΙΧΑΗΛ ΕΙΡΗΝΗΣ], [Professor_Dimension.Professors].
[ΠΑΝΑΓΙΩΤΗΣ ΑΡΙΔΑΣ], [Professor_Dimension.Professors].[ΠΕΡΙΚΛΗΣ ΓΚΟΛΑΣ],
[Professor_Dimension.Professors].[ΣΥΜΕΩΝ ΠΑΠΑΔΑΣ], [Professor_Dimension.Professors].[ΦΩΤΙΟΣ
ΣΑΚΗΣ], [Professor_Dimension.Professors].[ΧΡΗΣΤΟΣ ΠΙΤΑΣ]})) ON ROWS FROM [Grades]

```

Σχ 6.4.13 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.12

				Measures
Departments	Curriculums	Courses	Classes	Avg_Grade
KE	MBA	MBA50	AΘΗ1	3,355
			AΘΗ2	3,548
			ΘΕΣ1	3,14
			ΘΕΣ2	2,942
			ΠΑΤ1	3,658
		MBA51	AΘΗ1	3,05
			AΘΗ2	3,65
			ΘΕΣ1	3,254
			ΘΕΣ2	3,451
			ΠΑΤ1	3,345
	TPA	TPA50	AΘΗ1	3,365
			AΘΗ2	3,121
			ΘΕΣ1	3,449
			ΘΕΣ2	3,348
			ΠΑΤ1	3,133
		TPA51	AΘΗ1	3,659
			AΘΗ2	3,635
			ΘΕΣ1	3,506
		ΘΕΣ2	3,492	
		ΠΑΤ1	3,234	

Σχ 6.4.14 Οι βαθμοί της κάθε τάξης σε κάθε μάθημα για τα προγράμματα σπουδών της σχολής KE

```

MDX Query
Run Reset Query Execution Time : 268 msec

1 SELECT NON EMPTY {[Measures].[Avg_Grade]} ON COLUMNS, NON EMPTY
CrossJoin({[Department_Dimension.Departments].[KE]},
CrossJoin({[Curriculum_Dimension.Curriculums].[ΑΓΓ], [Curriculum_Dimension.Curriculums].
[ΓΑΑ], [Curriculum_Dimension.Curriculums].[ΓΕΡ], [Curriculum_Dimension.Curriculums].[ΜΒΑ],
[Curriculum_Dimension.Curriculums].[ΠΛΗ], [Curriculum_Dimension.Curriculums].[ΠΛΣ],
[Curriculum_Dimension.Curriculums].[ΠΣΕ], [Curriculum_Dimension.Curriculums].[ΤΡΑ]},
CrossJoin({[Course_Dimension.Courses].[ΑΓΓ52], [Course_Dimension.Courses].[ΑΓΓ53],
[Course_Dimension.Courses].[ΓΑΑ50], [Course_Dimension.Courses].[ΓΑΑ51],
[Course_Dimension.Courses].[ΓΕΡ50], [Course_Dimension.Courses].[ΓΕΡ51],
[Course_Dimension.Courses].[ΜΒΑ50], [Course_Dimension.Courses].[ΜΒΑ51],
[Course_Dimension.Courses].[ΠΛΗ10], [Course_Dimension.Courses].[ΠΛΗ11],
[Course_Dimension.Courses].[ΠΛΣ50], [Course_Dimension.Courses].[ΠΛΣ51],
[Course_Dimension.Courses].[ΠΣΕ50], [Course_Dimension.Courses].[ΠΣΕ51],
[Course_Dimension.Courses].[ΤΡΑ50], [Course_Dimension.Courses].[ΤΡΑ51]},
{[Class_Dimension.Classes].[ΑΘΗ1], [Class_Dimension.Classes].[ΑΘΗ2],
[Class_Dimension.Classes].[ΘΕΣ1], [Class_Dimension.Classes].[ΘΕΣ2],
[Class_Dimension.Classes].[ΠΑΤ1]})) ON ROWS FROM [Grades]

```

Σχ 6.4.15 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.14

					Measures
Years	Departments	Curriculums	Courses	Professors	Avg_Grade
2015-01-01	ΚΕ	MBA	MBA50	ΕΜΜΑΝΟΥΗΛ ΠΕΡΗΣ	3,134
				ΘΕΟΦΑΝΗΣ ΜΑΜΟΣ	3,548
			MBA51	ΔΗΜΗΤΡΙΟΣ ΠΑΠΑΣ	3,257
				ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ	3,341
				ΠΑΝΑΓΙΩΤΗΣ ΑΡΙΔΑΣ	3,346

Σχ 6.4.16 Οι βαθμοί των καθηγητών, σε κάθε τμήμα του προγράμματος σπουδών MBA, της σχολής ΚΕ για τον χρόνο 2015, που είναι πάνω από 3

```

MDX Query
Run Reset Query Execution Time : 268 msec
1 SELECT NON EMPTY {[Measures].[Avg_Grade]} ON COLUMNS, NON EMPTY
filter(CrossJoin({[Year_Dimension.Years].[2015-01-01]},
CrossJoin({[Department_Dimension.Departments].[ΚΕ]},
CrossJoin({[Curriculum_Dimension.Curriculums].[MBA]},
CrossJoin({[Course_Dimension.Courses].[MBA50], [Course_Dimension.Courses].[MBA51]},
{[Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΚΥΡΚΗΣ], [Professor_Dimension.Professors].
[ΕΜΜΑΝΟΥΗΛ ΠΕΡΗΣ], [Professor_Dimension.Professors].[ΘΕΟΦΑΝΗΣ ΜΑΜΟΣ],
[Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΚΑΡΑΓΑΣ], [Professor_Dimension.Professors].
[ΜΙΧΑΗΛ ΕΙΡΗΝΗΣ], [Professor_Dimension.Professors].[ΓΕΩΡΓΙΟΣ ΑΡΓΗΣ],
[Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΠΑΠΑΣ], [Professor_Dimension.Professors].
[ΔΗΜΗΤΡΙΟΣ ΣΤΑΜΟΣ], [Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ],
[Professor_Dimension.Professors].[ΠΑΝΑΓΙΩΤΗΣ ΑΡΙΔΑΣ]}))))) , [Measures].[Avg_Grade] > 3.0)
ON ROWS FROM [Grades]

```

Σχ 6.4.17 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.16

				Measures
Years	Curriculums	Courses	Classes	Num_Of_Grades
2012-01-01	ΠΛΗ	ΠΛΗ10	AllClasses	32
			ΑΘΗ1	5
			ΑΘΗ2	4
			ΘΕΣ1	5
			ΘΕΣ2	13
			ΠΑΤ1	5
		ΠΛΗ11	AllClasses	41
			ΑΘΗ1	11
			ΑΘΗ2	8
			ΘΕΣ1	7
			ΘΕΣ2	6
			ΠΑΤ1	9

Σχ 6.4.18 Πόσοι φοιτητές βαθμολόγησαν καθηγητές στις τάξεις του κάθε μαθήματος για το πρόγραμμα σπουδών ΠΛΗ το 2012



```

MDX Query
Run Reset Query Execution Time : 268 msec

1 SELECT NON EMPTY {[Measures].[Num_Of_Grades]} ON COLUMNS, NON EMPTY
CrossJoin({[Year_Dimension.Years].[2012-01-01]},
CrossJoin({[Curriculum_Dimension.Curriculums].[ΠΛΗ]},
CrossJoin({[Course_Dimension.Courses].[ΠΛΗ10], [Course_Dimension.Courses].[ΠΛΗ11]},
{[Class_Dimension.Classes].[AllClasses], [Class_Dimension.Classes].[ΑΘΗ1],
[Class_Dimension.Classes].[ΑΘΗ2], [Class_Dimension.Classes].[ΘΕΣ1],
[Class_Dimension.Classes].[ΘΕΣ2], [Class_Dimension.Classes].[ΠΑΤ1]})) ON ROWS FROM
[Grades]

```

Σχ 6.4.19 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.18

		Measures
Years	Professors	Avg_Grade
2012-01-01	ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ	4,198
	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	4,198
	ΔΗΜΗΤΡΙΟΣ ΓΕΩΡΓΙΟΥ	4,17
	ΙΩΑΝΝΗΣ ΜΑΥΡΙΔΑΚΗΣ	4,129
	ΣΟΦΙΑ ΣΑΧΛΟΥ	4,163
	ΣΥΜΕΩΝ ΠΑΠΑΔΑΣ	4,29
	2014-01-01	ΧΡΙΣΤΙΝΑ ΚΑΛΦΑΣ
2016-01-01	ΑΓΓΕΛΟΣ ΚΑΝΑΡΙΝΗΣ	4,279
	ΑΛΙΝΑ ΧΛΩΗ	4,246
	ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ	4,425
	ΒΑΣΙΛΕΙΟΣ ΒΕΡΟΣ	4,372
	ΒΑΣΙΛΕΙΟΣ ΖΟΥΡΛΟΣ	4,228
	ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΟΣ	4,428
	ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ	4,11
	ΓΕΩΡΓΙΟΣ ΚΑΡΑΣ	4,21
	ΓΡΗΓΟΡΙΟΣ ΓΚΙΝΙΑΣ	4,105
		ΕΥΘΥΜΙΟΣ ΖΕΦΥΡΗΣ
	ΘΕΟΔΩΡΟΣ ΤΖΟΥΡΑΣ	4,582
	ΙΩΑΝΝΗΣ ΡΕΦΑΝΗΣ	4,178
	ΚΥΡΙΑΚΗ ΚΟΣΜΑ	4,207
	ΛΑΖΑΡΟΣ ΗΛΙΑΣ	4,261
	ΜΑΡΙΑ ΒΙΡΒΑΕΡΑ	4,252
	ΜΑΡΙΑ ΠΑΠΑΔΟΥΝΗ	4,15
	ΜΑΡΙΝΑ ΓΠΣΕΣ	4,225
	ΜΙΧΑΗΛ ΒΑΣΙΛΕΙΟΥ	4,329
	ΠΑΝΑΓΙΩΤΗΣ ΑΡΙΔΑΣ	4,192
	ΠΑΝΑΓΙΩΤΗΣ ΔΗΜΟΣ	4,56
	ΠΕΡΙΚΛΗΣ ΤΑΛΑΝΗΣ	4,463
	ΣΠΥΡΙΔΩΝ ΦΕΡΤΑΣ	4,461
	ΣΤΕΦΑΝΟΣ ΓΚΡΙΖΟΣ	4,673
	ΦΩΤΕΙΝΗ ΔΗΜΗΤΡΙΟΥ	4,141
	ΦΩΤΙΟΣ ΛΩΤΟΦΑΓΟΣ	4,361
	ΦΩΤΙΟΣ ΣΑΚΗΣ	4,195
	ΧΡΙΣΤΙΝΑ ΚΑΛΦΑΣ	4,297

Σχ 6.4.20 Πόσοι καθηγητές πήραν πάνω από 4,1 ανά χρονιά

```

MDX Query
Run Reset Query Execution Time : 268 msec

1 SELECT NON EMPTY {[Measures].[Avg_Grade]} ON COLUMNS, NON EMPTY
filter(CrossJoin({[Year_Dimension.Years].[2012-01-01], [Year_Dimension.Years].[2013-01-
01], [Year_Dimension.Years].[2014-01-01], [Year_Dimension.Years].[2015-01-01],
[Year_Dimension.Years].[2016-01-01]}, {[Professor_Dimension.Professors].[ΑΓΓΕΛΟΣ
ΚΑΝΑΡΙΝΗΣ], [Professor_Dimension.Professors].[ΑΓΛΑΪΑ ΠΟΠΗ],
[Professor_Dimension.Professors].[ΑΛΕΞΑΝΔΡΑ ΑΝΔΡΕΟΥ], [Professor_Dimension.Professors].
[ΑΛΕΞΑΝΔΡΟΣ ΔΗΜΗΤΡΑΣ], [Professor_Dimension.Professors].[ΑΛΙΝΑ ΧΛΩΗ],
[Professor_Dimension.Professors].[ΑΝΑΣΤΑΣΙΑ ΓΕΩΡΓΙΟΥ], [Professor_Dimension.Professors].
[ANNA ΜΑΡΩ], [Professor_Dimension.Professors].[ΒΑΣΙΛΕΙΑ ΚΑΖΑΔΟΥ],
[Professor_Dimension.Professors].[ΒΑΣΙΛΕΙΑ ΚΑΖΟΥ], [Professor_Dimension.Professors].
[ΒΑΣΙΛΕΙΟΣ ΒΕΡΟΣ], [Professor_Dimension.Professors].[ΒΑΣΙΛΕΙΟΣ ΖΟΥΡΛΟΣ],
[Professor_Dimension.Professors].[ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΛΟΣ],
[Professor_Dimension.Professors].[ΓΕΩΡΓΙΑ ΑΛΤΗ], [Professor_Dimension.Professors].
[ΓΕΩΡΓΙΟΣ ΑΡΓΗΣ], [Professor_Dimension.Professors].[ΓΕΩΡΓΙΟΣ ΓΡΑΜΟΣ],
[Professor_Dimension.Professors].[ΓΕΩΡΓΙΟΣ ΚΑΡΑΣ], [Professor_Dimension.Professors].
[ΓΡΗΓΟΡΙΟΣ ΓΚΙΝΙΑΣ], [Professor_Dimension.Professors].[ΔΗΜΗΤΡΗΣ ΝΙΚΟΛΑΟΥ],
[Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΓΕΩΡΓΙΟΥ], [Professor_Dimension.Professors].
[ΔΗΜΗΤΡΙΟΣ ΚΥΡΚΗΣ], [Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΠΑΠΑΣ],
[Professor_Dimension.Professors].[ΔΗΜΗΤΡΙΟΣ ΣΤΑΜΟΣ], [Professor_Dimension.Professors].
[ΕΜΜΑΝΟΥΗΛ ΠΕΡΗΣ], [Professor_Dimension.Professors].[ΕΥΑΓΓΕΛΙΑ ΜΟΥΡΗ],
[Professor_Dimension.Professors].[ΕΥΑΓΓΕΛΟΣ ΜΑΝΟΣ], [Professor_Dimension.Professors].
[ΕΥΘΥΜΙΟΣ ΖΕΦΥΡΗΣ], [Professor_Dimension.Professors].[ΘΕΟΔΩΡΟΣ ΤΖΟΥΡΑΣ],
[Professor_Dimension.Professors].[ΘΕΟΦΑΝΗΣ ΜΑΜΟΣ], [Professor_Dimension.Professors].
[ΘΕΟΦΑΝΗΣ ΟΡΦΑΝΟΣ], [Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΔΗΜΟΣ],
[Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΚΑΡΑΓΑΣ], [Professor_Dimension.Professors].
[ΙΩΑΝΝΗΣ ΛΑΡΙΔΗΣ], [Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΣ],
[Professor_Dimension.Professors].[ΙΩΑΝΝΗΣ ΜΑΥΡΙΔΑΚΗΣ], [Professor_Dimension.Professors].
[ΙΩΑΝΝΗΣ ΡΕΦΑΝΗΣ], [Professor_Dimension.Professors].[ΚΥΡΙΑΚΗ ΚΟΣΜΑ],
[Professor_Dimension.Professors].[ΛΑΖΑΡΟΣ ΗΛΙΑΣ], [Professor_Dimension.Professors].[ΜΑΡΙΑ
ΒΙΡΒΑΕΡΑ], [Professor_Dimension.Professors].[ΜΑΡΙΑ ΠΑΠΑΔΟΥΝΗ],
[Professor_Dimension.Professors].[ΜΑΡΙΝΑ ΓΙΤΣΕΣ], [Professor_Dimension.Professors].
[ΜΑΡΙΟΣ ΑΡΓΥΡΟΣ], [Professor_Dimension.Professors].[ΜΙΧΑΗΛ ΒΑΣΙΛΕΙΟΥ],
[Professor_Dimension.Professors].[ΜΙΧΑΗΛ ΕΙΡΗΝΗΣ], [Professor_Dimension.Professors].
[ΝΙΚΟΛΑΟΣ ΚΑΥΓΑΣ], [Professor_Dimension.Professors].[ΝΙΚΟΛΑΟΣ ΜΑΝΩΛΗΣ],
[Professor_Dimension.Professors].[ΝΙΚΟΛΑΟΣ ΠΑΡΑΝΥΧΙΔΑΣ],
[Professor_Dimension.Professors].[ΝΙΚΟΣ ΚΑΡΟΥΣΟΣ], [Professor_Dimension.Professors].
[ΟΔΥΣΣΕΑΣ ΜΟΥΓΚΟΣ], [Professor_Dimension.Professors].[ΟΛΓΑ ΠΑΤΕΡΑ],
[Professor_Dimension.Professors].[ΠΑΝΑΓΙΩΤΗΣ ΑΡΙΔΑΣ], [Professor_Dimension.Professors].
[ΠΑΝΑΓΙΩΤΗΣ ΔΗΜΟΣ], [Professor_Dimension.Professors].[ΠΕΡΙΚΛΗΣ ΓΚΟΛΑΣ],
[Professor_Dimension.Professors].[ΠΕΡΙΚΛΗΣ ΤΑΛΑΝΗΣ], [Professor_Dimension.Professors].
[ΣΟΦΙΑ ΡΙΖΟΠΟΥΛΟΥ], [Professor_Dimension.Professors].[ΣΟΦΙΑ ΣΑΧΛΟΥ],
[Professor_Dimension.Professors].[ΣΠΥΡΙΔΩΝ ΦΕΡΤΑΣ], [Professor_Dimension.Professors].
[ΣΤΕΦΑΝΟΣ ΓΚΡΙΖΟΣ], [Professor_Dimension.Professors].[ΣΥΜΕΩΝ ΠΑΠΑΔΑΣ],
[Professor_Dimension.Professors].[ΦΩΤΕΙΝΗ ΔΗΜΗΤΡΙΟΥ], [Professor_Dimension.Professors].
[ΦΩΤΙΟΣ ΛΩΤΟΦΑΓΟΣ], [Professor_Dimension.Professors].[ΦΩΤΙΟΣ ΣΑΚΗΣ],
[Professor_Dimension.Professors].[ΧΡΗΣΤΟΣ ΠΙΤΑΣ], [Professor_Dimension.Professors].
[ΧΡΙΣΤΙΝΑ ΚΑΛΦΑΣ])), [Measures].[Avg_Grade] > 4.1) ON ROWS FROM [Grades]

```

Σχ 6.4.21 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.20

	Years				
	2012-01-01	2013-01-01	2014-01-01	2015-01-01	2016-01-01
Measures	Measures	Measures	Measures	Measures	Measures
Professors	Avg_Grade	Avg_Grade	Avg_Grade	Avg_Grade	Avg_Grade
ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΛΟΣ	3,368	3,381	3,865	2,996	4,428
ΓΕΩΡΓΙΟΣ ΚΑΡΑΣ	3,838	2,973	3,583	2,862	4,21
ΔΗΜΗΤΡΗΣ ΝΙΚΟΛΑΟΥ	3,161	3,232	3,617	3,37	3,777

Σχ 6.4.22 Οι βαθμοί των καθηγητών ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΛΟΣ, ΓΕΩΡΓΙΟΣ ΚΑΡΑΣ και ΔΗΜΗΤΡΗΣ ΝΙΚΟΛΑΟΥ ανά χρόνο.

```
MDX Query
▶ Run  ◀ Reset  Query Execution Time : 268 msec
1 SELECT NON EMPTY CrossJoin({[Year_Dimension.Years].[2012-01-01], [Year_Dimension.Years].
[2013-01-01], [Year_Dimension.Years].[2014-01-01], [Year_Dimension.Years].[2015-01-01],
[Year_Dimension.Years].[2016-01-01]}, {[Measures].[Avg_Grade]}) ON COLUMNS, NON EMPTY
{[Professor_Dimension.Professors].[ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΟΥΡΛΟΣ], [Professor_Dimension.Professors].
[ΓΕΩΡΓΙΟΣ ΚΑΡΑΣ], [Professor_Dimension.Professors].[ΔΗΜΗΤΡΗΣ ΝΙΚΟΛΑΟΥ]} ON ROWS FROM
[Grades]
```

Σχ 6.4.23 Ο αντίστοιχος κώδικας MDX για το σχήμα 6.4.22

## 7 Συμπεράσματα και μελλοντικές κατευθύνσεις

Όπως βλέπουμε το Pentaho είναι ένα αρκετά δυνατό και εύχρηστο εργαλείο για OLAP reports. Με διάφορες επεκτάσεις όπως το Pivott4j γίνεται αρκετά απλό και μπορεί κανείς να δημιουργήσει queries χωρίς να χρειαστεί να γράψει κώδικα στην MDX, απλά χρησιμοποιώντας το γραφικό περιβάλλον. Βέβαια υπάρχει η δυνατότητα να επέμβουμε στα MDX queries που δημιουργούνται με το γραφικό περιβάλλον αλλά και να δημιουργήσουμε δικά μας από την αρχή. Για το συγκεκριμένο παράδειγμα της βάσης δεδομένων του Ελληνικού Ανοικτού Πανεπιστημίου (ΕΑΠ) το Pentaho μας εξυπηρετεί αρκετά και βοηθά στο να βγάλουμε διάφορα συμπεράσματα. Αυτό όμως δεν σημαίνει ότι το OLAP είναι καλύτερο από την SQL. Αυτά τα δύο εργαλεία είναι αρκετά διαφορετικά και πολλές φορές δεν μπορούμε καν να τα συγκρίνουμε μεταξύ τους.

Αν έχουμε συγκριμένες αναλύσεις που απαιτούν συγκριμένα σύνολα ίσως είναι καλύτερο να δημιουργήσουμε κάποιο πίνακα με αυτά τα σύνολα και να χρησιμοποιήσουμε queries της SQL σε αυτούς τους πίνακες. Για παράδειγμα αν ένα τμήμα πωλήσεων θέλει να βλέπει τις πωλήσεις κάθε μαγαζιού για κάποιους μήνες τότε ίσως είναι καλύτερο να φτιάξουμε πίνακες με αυτά τα ποσά και να εξετάσουν αυτά τα δεδομένα.

Αν όμως οι χρήστες θέλουν να εξετάζουν τα δεδομένα από πολλές διαφορετικές όψεις αυτή η μέθοδος δεν εξυπηρετεί τόσο διότι θα πρέπει να δημιουργούμε ένα διαφορετικό πίνακα με τις νέες τιμές κάθε φορά. Εδώ εξυπηρετεί το να δημιουργήσουμε έναν κύβο και οι χρήστες να μπορούν να εξετάσουν μόνοι τους τις διάφορες όψεις χρησιμοποιώντας διαφορετικές διαστάσεις κάθε φορά, απλά και γρήγορα χωρίς να χρειάζεται να δημιουργήσουμε καινούριους πίνακες με νέες τιμές.

Οπότε το ποιο από τα δύο εργαλεία θα χρησιμοποιήσουμε κάθε φορά εξαρτάται από το τι απαιτήσεις έχει ο χρήστης και τι συμπεράσματα θέλει να βγάλει από την βάση δεδομένων του.

Το OLAP χρησιμοποιείται για την εξαγωγή γνώσης από την αποθήκη δεδομένων. Ένα άλλο είδος εργαλείου που χρησιμοποιείται με αυτόν τον σκοπό είναι τα εργαλεία εξόρυξης δεδομένων. Μέχρι τώρα, οι δύο ερευνητικές κοινότητες εξελίσσονται χωριστά. Η πρώτη πρέπει να είναι διαδραστική, ενώ η δεύτερη παρουσιάζει προβλήματα υπολογιστικής πολυπλοκότητας. Ωστόσο, φαίνεται πολλά υποσχόμενη η ενοποίηση των δύο ειδών εργαλείων, έτσι ώστε να μπορούν να επωφεληθούν το ένα από το άλλο. Στην πραγματικότητα, είχε ήδη προταθεί στο [12], και μερικά εργαλεία όπως το Microsoft Analysis Services ενοποιεί τα δύο είδη κατά κάποιο τρόπο. Παρ'όλα αυτά, υπάρχει πολλή δουλειά να κάνουμε στον τομέα αυτό ακόμα.

Βέβαια, η ασφάλεια είναι συνήθως ένα ελάττωμα στην αποθήκευση τέτοιων δεδομένων. Στο παρελθόν, τα εργαλεία OLAP χρησιμοποιούνταν από λίγους χρήστες και όλοι τους είχαν μεγάλες ευθύνες στις επιχειρήσεις. Έτσι δεν υπήρχε θέμα με την εμπιστευτικότητα. Σήμερα όμως, με την αύξηση του ανθρωπίνου δυναμικού οι χρήστες των εργαλείων OLAP τόσο μέσα όσο και έξω από τις επιχειρήσεις, η ασφάλεια είναι πλέον μια προτεραιότητα. Επιπλέον, προσωπικά δεδομένα (όπως εκείνα των πελατών) αναλύονται σε όλες σχεδόν τις επιχειρήσεις. Έτσι, οι μηχανισμοί ελέγχου του περιβάλλοντος και των εργαλείων OLAP πρέπει να μελετηθούν στην εξόρυξη δεδομένων.

Άλλες ερευνητικές κατευθύνσεις στο OLAP μπορεί να είναι η βελτίωση της αλληλεπίδρασης του χρήστη και η ευελιξία στον υπολογισμό στατιστικών στοιχείων, καθώς και η ενσωμάτωση της ανάλυσης των σεναρίων what-if.

## Απόδοση όρων

Queries	Ερωτήματα βάσεων δεδομένων
SQL	Δομημένη Γλώσσα ερωτημάτων
Metadata	Μεταδεδομένα
Σενάρια what-if	Υποθετικά σενάρια
Caching	Αποθήκευση στην προσωρινή μνήμη
Server	Διακομιστής
Slice	Τεμαχισμός
Dice	Τεμαχισμός σε κύβους
Drilling data	Διάτρηση δεδομένων
Μνήμη RAM	Μνήμη τυχαίας προσπέλασης
Relational database management system (RDBMS)	Σύστημα διαχείρισης σχεσιακής βάσης δεδομένων

## Βιβλιογραφία

- [1] Edgar F. Codd, Providing OLAP to user-analysts: An IT mandate, 1993
- [2] Nigel Pendse, What is OLAP?, 2007
- [3] Επίσημο site του OLAP, <http://www.olap.com/> (Τελευταία επίσκεψη: Μάιος 2016)
- [4] OLTP vs. OLAP, <http://datawarehouse4u.info> (Τελευταία επίσκεψη: Μάιος 2016)
- [5] Introduction to OLAP, Crystal Decisions, 2002
- [6] Just What Are Cubes Anyway? (A Painless Introduction to OLAP Technology), <http://msdn.microsoft.com> (Τελευταία επίσκεψη: Μάιος 2016)
- [7] Andrei Pandre, OLAP Cubes, <https://apandre.wordpress.com> (Τελευταία επίσκεψη: Μάιος 2016)
- [8] Alberto Abelló and Oscar Romero, ON-LINE ANALYTICAL PROCESSING
- [9] <http://www.hypertextbookshop.com>
- [10] Sobhan Badiozamani, Microsoft SQL Server OLAP Solution, 2008
- [11] <http://www.pivot4j.org/> (Τελευταία επίσκεψη: Μάιος 2017)
- [12] Jiawei Han, OLAP mining: Integration of OLAP with Data Mining Morgan Kaufmann, 1997

## **Πνευματικά δικαιώματα**

Copyright © ΤΕΙ Δυτικής Ελλάδας. Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1988 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον.

Καρβέλης Νικόλαος, 2017