



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

## **ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

### **ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**Π.Μ.Σ. “ΕΦΑΡΜΟΣΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ”**

#### **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce**

**Αποστολόπουλος Βασίλειος**

**Εισηγητής: Δρ Κωνσταντίνος Κουκουλέτσος, Καθηγητής**

**ΑΘΗΝΑ  
ΜΑΡΤΙΟΣ 2019**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce**

**Βασίλειος Χ. Αποστολόπουλος**

**A.M. AIS0133**

**Εισηγητής:**

**Δρ Κωνσταντίνος Κουκουλέτσος, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης**



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος **ΑΠΟΣΤΟΛΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ**, του **ΧΡΗΣΤΟΥ**, με αριθμό μητρώου **AIS0133** φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

**Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»**



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της γλώσσας προγραμματισμού Java και των πλατφορμών διαχείρισης ηλεκτρονικού εμπορίου (e-commerce). Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου για την συμπαράστασή της τόσο καθ' όλη τη διάρκεια της φοίτησής μου στο παρόν μεταπτυχιακό, όσο και κατά τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.





## ΠΕΡΙΛΗΨΗ

Οι πλατφόρμες ανοικτού κώδικα για e-commerce προσφέρουν γρήγορη και ευέλικτη λύση στην παρουσία μίας επιχείρησης στο διαδίκτυο. Όμως ένα από τα μειονεκτήματά τους είναι η έλλειψη διασύνδεσης με την Java που είναι μία από τις πιο δημοφιλείς αντικειμενοστραφείς γλώσσες προγραμματισμού με μεγάλο πλεονέκτημα την ανεξαρτησία από το λειτουργικό σύστημα του υπολογιστή. Η διπλωματική θα εξετάσει τις κυριότερες πλατφόρμες e-commerce και θα παρουσιάσει μία διασύνδεση μεταξύ της Βάσης Δεδομένων Microsoft SQL Server και μίας από τις e-commerce πλατφόρμες χρησιμοποιώντας τη Java, ώστε να εντοπιστούν τα προβλήματα συμβατότητας και να εξεταστεί η δυνατότητα δημιουργίας προτύπων για την διασύνδεση αυτή.

## ABSTRACT

Open source e-commerce platforms provide a fast, flexible solution to the presence of an online business. But one of their drawbacks is the lack of interfacing with Java, which is one of the most popular object-oriented programming languages with great advantage of independence from the computer's operating system. The diploma will examine the major e-commerce platforms and present an interface between the Microsoft SQL Server Database and one of the e-commerce platforms using Java to identify compatibility problems and examine the ability to create standards for this interface.

**ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ:** Πλατφόρμες Ηλεκτρονικού Εμπορίου

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** e-commerce, e-shop, ηλεκτρονικό κατάστημα, Βάσεις Δεδομένων, Microsoft SQL Server, MySQL, Java, WooCommerce, Magento, PrestaShop, OpenCart, Apache2

## Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1 .....	12
ΕΙΣΑΓΩΓΗ .....	12
1.1 Η έννοια του ηλεκτρονικού εμπορίου .....	12
1.2 Το ηλεκτρονικό εμπόριο σήμερα .....	12
1.3 Περιγραφή του αντικειμένου .....	13
ΚΕΦΑΛΑΙΟ 2 .....	15
ΟΙ ΤΕΧΝΟΛΟΓΙΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	15
2.1 Βάση Δεδομένων – SQL .....	15
2.2 Microsoft SQL Server .....	16
2.3 MySQL.....	17
2.3.1 Τι είναι η MySQL.....	17
2.3.2 Τι είναι μια βάση MySQL .....	17
2.4 Apache Web Server .....	17
2.5 Η γλώσσα προγραμματισμού Java.....	18
2.5.1 Οι στόχοι της Java.....	18
2.5.2 Απλή Αντικειμενοστραφής και Εύκολη .....	19
2.5.3 Ισχυρή και Ασφαλής .....	20
2.5.4 Ανεξαρτησία και φορητότητα.....	21
2.5.5 Υψηλή απόδοση συστήματος.....	22
2.6 Content Management System .....	22
ΚΕΦΑΛΑΙΟ 3.....	24
ΚΑΤΗΓΟΡΙΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ (e-commerce).....	24
3.1 Επιχείρηση με Καταναλωτή (Business to Consumer ή B2C).....	24
3.2 Επιχείρηση με Επιχείρηση (Business to Business ή B2B) .....	26
3.3 Κοινού με Κράτος (C2G ή B2G) .....	27
3.4 Καταναλωτή προς Καταναλωτή (Consumer to Consumer ή C2C).....	29
3.5 Χαρακτηριστικά e-shop .....	29
ΚΕΦΑΛΑΙΟ 4.....	33
ΟΙ ΔΗΜΟΦΙΛΕΣΤΕΡΕΣ ΠΛΑΤΦΟΡΜΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ .....	33
4.1 WooCommerce.....	34
4.1.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ WooCommerce.....	35
4.2 Magento.....	37
4.2.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ Magento38	

4.3 PrestaShop.....	41
4.3.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ PrestaShop .....	42
4.4 OpenCart.....	45
4.4.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ OpenCart .....	46
ΚΕΦΑΛΑΙΟ 5.....	49
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΗ ΤΩΝ ΕΡΓΑΛΕΙΩΝ .....	49
5.1 Εγκατάσταση και ρύθμιση του Microsoft SQL Server σε Ubuntu 18.04.....	49
5.2 Ρύθμιση του Web Server Apache2 σε Ubuntu 18.04.....	51
5.3 Λήψη και εγκατάσταση του OpenCart 3.0.3.1.....	54
5.4 Ρύθμιση των παραμέτρων του OpenCart 3.0.3.1 για το Demo Βιβλιοπωλείο .....	55
ΚΕΦΑΛΑΙΟ 6.....	61
ΔΙΑΣΥΝΔΕΣΗ Microsoft SQL Server ΚΑΙ OpenCart.....	61
6.1 Η Βάση Δεδομένων sqlbookstore .....	61
6.2 Η Βάση Δεδομένων bookstore .....	62
6.3 Η εφαρμογή BookStore .....	64
6.3.1 Οι βιβλιοθήκες της εφαρμογής .....	64
6.3.2 Οι κλάσεις της εφαρμογής .....	65
6.4 Το Demo Βιβλιοπωλείο.....	69
ΚΕΦΑΛΑΙΟ 7.....	72
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ .....	72
7.1 Σύνοψη της πτυχιακής εργασίας.....	72
7.2 Συμπεράσματα - προοπτικές.....	73
ΠΑΡΑΡΤΗΜΑ .....	74
Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	74
LoginFrm.java.....	74
MainFrm.java.....	84
InsertProducts.frm.....	88
ComsFrm.java.....	104
Actions.java .....	108
Βιβλιογραφία .....	119



## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό γίνεται αναφορά στην έννοια του ηλεκτρονικού εμπορίου (e-commerce) και αναλύεται το αντικείμενο της πτυχιακής εργασίας.

#### 1.1 Η έννοια του ηλεκτρονικού εμπορίου

**Ηλεκτρονικό εμπόριο (Η.Ε.)** το οποίο είναι ευρέως γνωστό και ως *e-commerce*, *eCommerce* ή *e-comm*, ορίζεται το εμπόριο παροχής αγαθών και υπηρεσιών που πραγματοποιείται εξ αποστάσεως με ηλεκτρονικά μέσα. Βασίζεται στην ηλεκτρονική μετάδοση δεδομένων και δεν είναι αναγκαία η φυσική παρουσία των συμβαλλομένων μερών, δηλαδή του πωλητή και του αγοραστή. Το σύνολο των διαδικτυακών διαδικασιών που περιλαμβάνει είναι η ανάπτυξη, προώθηση, πώληση, παράδοση, εξυπηρέτηση και πληρωμής για τα προϊόντα και τις υπηρεσίες που επιθυμεί ο αγοραστής. Στην καθημερινότητα, οι ανταλλαγές που διεξάγονται ηλεκτρονικά, έχουν αυξηθεί ξεπερνώντας κάθε προσδοκία λόγω της αλματώδους αύξησης της χρήσης του Διαδικτύου. Το ηλεκτρονικό εμπόριο περιλαμβάνει καινοτόμες διαδικασίες όπως η ηλεκτρονική μεταφορά χρηματικών πόρων, η διαχείριση της εφοδιαστικής αλυσίδας (*supply chain management*), το διαδικτυακό μάρκετινγκ (*Internet marketing*), η διεκπεραίωση διαδικτυακών διαδικασιών (*online transaction processing*), η ανταλλαγή ηλεκτρονικών δεδομένων (*electronic data interchange, EDI*), η καταγραφή συστημάτων διοίκησης (*inventory management*) και η αυτοματοποίηση συστημάτων συγκέντρωσης δεδομένων.

#### 1.2 Το ηλεκτρονικό εμπόριο σήμερα

Σήμερα όλο και περισσότεροι πολίτες της χώρας μας πραγματοποιούν τις αγορές τους online, ενώ αντίστοιχα όλο και περισσότεροι έμποροι, ελεύθεροι επαγγελματίες, και εταιρίες εγχώριες και ξένες, προσφέρουν τα προϊόντα και τις υπηρεσίες τους ηλεκτρονικά. Σύμφωνα με το ΣΕΠΕ (Σύνδεσμος Επιχειρήσεων Πληροφορικής και Επικοινωνιών Ελλάδας), “ο κύκλος εργασιών του ηλεκτρονικού εμπορίου στην Ελλάδα αγγίζει πλέον το 2% του ΑΕΠ, ενώ σε απόλυτα μεγέθη

ανέρχεται σε €4 δις, τη στιγμή που στη χώρα υπάρχουν περισσότερα από 6.500 ελληνικά e-shops”.

Σύμφωνα με τα αποτελέσματα έρευνας που παρουσίασε το 2017 ο GR.EC.A (Ελληνικός Σύνδεσμος Ηλεκτρονικού Εμπορίου - Greek ECommerce Association) σε σχετική εκδήλωση, “ο κλάδος του ηλεκτρονικού εμπορίου στην Ελλάδα καταγράφει συνεχή και σημαντική ανάπτυξη. Σήμερα, στην Ελλάδα οι “online” καταναλωτές ξεπερνούν τα 3 εκατ. (σε σύνολο μεγαλύτερο από 7 εκατ. χρήστες internet), ξοδεύουν δε κατά μέσο όρο σε ετήσια βάση περί τα €1.300. Ο αριθμός αυτός αντιστοιχεί στο 40% του “online” πληθυσμού, ενώ σημαντική είναι και η αύξηση των αγορών από ελληνικά e-shops (περισσότεροι από το 80% των e-shoppers). Μάλιστα σε σχετική έρευνα που εκπονήθηκε, το 30% των ερωτηθέντων απάντησε πως δίνει το 50% του budget για την απόκτηση αγαθών και υπηρεσιών μέσω Διαδικτύου”.

Επίσης εντυπωσιακό είναι το γεγονός ότι ενώ το 2016, σύμφωνα με τα στοιχεία έρευνας που παρουσίασε το ELTRUN (Εργαστήριο Ηλεκτρονικού Εμπορίου του Οικονομικού Πανεπιστημίου Αθηνών) για το ηλεκτρονικό εμπόριο στην Ελλάδα, “παρά την αλματώδη ανάπτυξη της χρήσης καρτών τα δύο τελευταία χρόνια, η αντικαταβολή παραμένει για τους Έλληνες η δημοφιλέστερη μέθοδος πληρωμής για τις online αγορές τους”. Όπως έδειξαν τότε τα στοιχεία, υπάρχει σύνδεση του υψηλού ποσοστού της αντικαταβολής στην Ελλάδα με το φόβο των καταναλωτών για θέματα ασφάλειας και διαφύλαξης προσωπικών δεδομένων (διαδικτυακή απάτη), όπως επίσης και ανησυχία για το προϊόν που τελικά θα παραλάβουν (ποιότητα, δυνατότητα επιστροφής ή αλλαγής). Σήμερα τρία χρόνια μετά, οι επιφυλάξεις αυτές των Ελλήνων καταναλωτών δείχνουν σιγά σιγά να μειώνονται.

### 1.3 Περιγραφή του αντικειμένου

Αντικείμενο της παρούσης διπλωματικής εργασίας, είναι κατ’ αρχάς η παρουσίαση των προδιαγραφών που πρέπει να έχει μια πλατφόρμα ηλεκτρονικού εμπορίου (e-commerce) ανοικτού κώδικα. Ακολούθως γίνεται αναφορά στις κυριότερες από τις πλατφόρμες και η επιλογή μιας η οποία θα χρησιμοποιηθεί ως πρότυπο, έτσι ώστε να παρουσιαστεί η διασύνδεση μεταξύ αυτής και μιας custom Βάσης Δεδομένων, χρησιμοποιώντας τη γλώσσα προγραμματισμού Java, έτσι

ώστε να εντοπιστούν τυχόν προβλήματα συμβατότητας μεταξύ των διαφορετικών πλατφορμών.

Τέλος θα εξεταστεί η δυνατότητα δημιουργίας προτύπων έτσι ώστε τέτοιου είδους διασυνδέσεις να μπορούν να χρησιμοποιηθούν στο μέλλον αυτούσιες, ή με τη μικρότερη δυνατή παραμετροποίηση, για όσες πλατφόρμες ηλεκτρονικού εμπορίου ακολουθήσουν αυτή την προτυποποίηση.

Η custom Βάση Δεδομένων είναι η **Microsoft SQL Server 2017** και περιλαμβάνει 180.000 τίτλους βιβλίων που εκδόθηκαν στη χώρα μας, καθώς επίσης και 100.000 δημιουργούς (συγγραφείς, μεταφραστές, επιμελητές, επιμελητές σειράς, εικονογράφους, φωτογράφους, ζωγράφους, κ.ά.). Η διασύνδεση που θα υλοποιηθεί με τη **γλώσσα προγραμματισμού java**, θα δίνει τη δυνατότητα στο χρήστη να επιλέξει τίτλους βιβλίων ενός ή περισσότερων από τους εκδότες που βρίσκονται καταχωρημένοι στην custom Βάση Δεδομένων και να τους μεταφέρει αυτόματα στην Βάση της πλατφόρμας ηλεκτρονικού εμπορίου η οποία έχει επιλεγεί ως πρότυπο.

**Η πλατφόρμα ηλεκτρονικού εμπορίου η οποία έχει επιλεγθεί προκειμένου να υλοποιηθούν τα παραπάνω, είναι η OpenCart CMS Έκδοση 3.0.2.0, που στηρίζεται στο Σύστημα Διαχείρισης Βάσης Δεδομένων MySql 5.8 και φιλοξενείται σε Web Server Apache2.**

## ΚΕΦΑΛΑΙΟ 2

### ΟΙ ΤΕΧΝΟΛΟΓΙΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Στο κεφάλαιο αυτό γίνεται αναφορά στις τεχνολογίες οι οποίες χρησιμοποιήθηκαν για την υλοποίηση της εργασίας. Χρησιμοποιήθηκε Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα (ΕΛ/ΛΑΚ) δηλαδή λογισμικό που *“διανέμεται ελεύθερα και ο καθένας μπορεί να το χρησιμοποιήσει, να το αντιγράψει, να το τροποποιήσει ανάλογα με τις ανάγκες του και να το επαναδιανέμει”*.

#### 2.1 Βάση Δεδομένων – SQL

Με τον όρο **βάση δεδομένων** εννοούμε μία συλλογή δεδομένων που βρίσκονται κατάλληλα αποθηκευμένη σε έναν υπολογιστή, έτσι ώστε το άτομο που τη διαχειρίζεται να μπορεί με ευκολία να την προσπελάσει, να την διαχειριστεί και να την ενημερώσει. Οι ενέργειες αυτές γίνονται από τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (DataBase Management Systems - DBMS), που είναι εφοδιασμένα με τους κατάλληλους μηχανισμούς, έτσι ώστε να μπορούν να προσπελαίνουν αυτές τις οργανωμένες δομές δεδομένων και να τις επεξεργάζονται, χωρίς να υπάρχει κίνδυνος παραποίησης ή φθοράς των δεδομένων.

Κοινό σημείο αναφοράς διαφορετικών Συστημάτων Διαχείρισης Βάσεων Δεδομένων είναι η γλώσσα SQL (Structured Query Language). Η γλώσσα SQL, μπορεί μέσα από δικές της εντολές, να προσπελαίνει τα δομημένα δεδομένα μια Βάσης Δεδομένων. Ουσιαστικά λοιπόν, ένα Σύστημα Διαχείρισης Βάσης Δεδομένων μπορεί να δεχτεί και εκτελέσει εντολές SQL με σκοπό τη διαχείριση των δεδομένων της Βάσης.

Μια οργανωμένη δομή δεδομένων αποθηκεύει όχι μόνο το σύνολο δεδομένων, αλλά και τον τρόπο με τον οποίο αυτά συσχετίζονται. Όταν η λογική με την οποία τα δεδομένα μιας Βάσης οργανώνονται είναι η συσχέτιση (σχέση), τότε αυτές οι βάσεις αποκαλούνται Σχεσιακές Βάσεις Δεδομένων και τα συστήματα διαχείρισης Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (RDBMS - Relational Database Managements Systems). Άλλοι τύποι Συστημάτων Διαχείρισης Βάσεων Δεδομένων είναι τα Ιεραρχικά ΗDBMS (Hierarchical DBMS). Σε ένα Ιεραρχικό Σύστημα Διαχείρισης Βάσης Δεδομένων, τα δεδομένα αποθηκεύονται με τη μορφή δέντρου



ή σαν γράφος. Τότε η ίδια η δομή της βάσης υποδηλώνει και τη συσχέτιση των δεδομένων.

## 2.2 Microsoft SQL Server

Ο Microsoft SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύχθηκε και εξελίσσεται από τη Microsoft. Για πρώτη φορά κυκλοφόρησε το 1989 σε συνεργασία με την Sybase. Χρησιμοποιεί σαν κύριες γλώσσες την **T-SQL** και την **ANSI SQL**.

Τα δεδομένα, αποθηκεύονται σε μια βάση, (Βάση Δεδομένων), η οποία αποτελείται από πίνακες και κώδικα.

Ο Microsoft SQL Server υποστηρίζει όλους τους διαφορετικούς τύπους δεδομένων, όπως ακέραιους αριθμούς, αριθμούς κινητής υποδιαστολής, δεκαδικούς, αλφαριθμητικά δεδομένα, Varchar (string χαρακτήρων μεταβλητού μήκους), δυαδικούς αριθμούς (για αναπαράσταση μη δομημένων δεδομένων), κείμενα (text).

Επιτρέπεται επίσης η χρήση UDTs, δηλαδή σύνθετων τύπων δεδομένων, καθορισμένων από το χρήστη, οι οποίοι βασίζονται στους βασικούς τύπους αλλά μπορούν εύκολα να τροποποιηθούν.

Γίνεται χρήση της μνήμης RAM (buffering) όπου αποθηκεύονται εκεί οι σελίδες, με σκοπό της ελαχιστοποίησης της μεταφοράς δεδομένων από και προς τον σκληρό δίσκο.

Τα δεδομένα ανακτώνται από τη Βάση με τη χρήση ερωτήσεων (queries). Αυτός είναι και ο κύριος τρόπος ανάκτησης. Η σύνταξη των ερωτήσεων γίνεται με τη χρήση της γλώσσας T-SQL. Η T-SQL, είναι μια παραλλαγή της κλασσικής SQL, που αναπτύχθηκε από την Microsoft και Sybase και σε αντίθεση άλλες όπως η PL-SQL της Oracle, εφαρμόζει όλα τα ANSI standards που έχουν καθιερωθεί διεθνώς.

Επιτρέπονται επίσης και οι stored procedures. Οι stored procedures (αποθηκευμένες ερωτήσεις) είναι ερωτήσεις αποθηκευμένες στον SQL Server που συντάσσονται με τους κανόνες της T-SQL. Μπορούν να δεχτούν τιμές ως παραμέτρους καθώς και τα αποτελέσματά τους μπορούν να χρησιμοποιηθούν ως αποτελέσματα σε άλλες stored procedures. Εκτελούνται ταχύτερα σε σχέση με τις απλές εντολές SQL και μπορούν να καλέσουν functions, ή άλλες stored procedures.

## 2.3 MySQL

### 2.3.1 Τι είναι η MySQL

Η MySQL είναι γλώσσα προγραμματισμού για Συστήματα Διαχείρισης Βάσεων Δεδομένων (DBMS). Αποτελείται από μία διανομή (έκδοση) της γλώσσας SQL και ένα MySQL Σύστημα Διαχείρισης Βάσεων Δεδομένων, δηλαδή ένα σύστημα που δέχεται εντολές MySQL. Αντίθετα με άλλα Συστήματα Διαχείρισης Βάσεων Δεδομένων που η χρήση τους γίνεται επί πληρωμή, η MySQL είναι ελεύθερο λογισμικό και μπορεί να χρησιμοποιηθεί είτε ερασιτεχνικά είτε επαγγελματικά. Εκτός από τη MySQL, άλλα γνωστά Συστήματα Διαχείρισης Βάσεων Δεδομένων είναι ο SQL Server της Microsoft που είδαμε αναλυτικά παραπάνω, ο Oracle SQL server, η DB2 κ.ά.

### 2.3.2 Τι είναι μια βάση MySQL

Μια βάση δεδομένων MySQL είναι ένας κατάλληλα διαμορφωμένος χώρος όπου αποθηκεύονται δεδομένα με σχεσιακό τρόπο.

Ένα παράδειγμα μέσα από το οποίο μπορούμε να δούμε τον τρόπο με τον οποίο μπορεί να αποτελέσει χρήση της MySQL καθημερινή πρακτική στο θέμα των ιστοσελίδων είναι το ακόλουθο:

Αν έχουμε μία ιστοσελίδα η οποία περιέχει μία φόρμα παραγγελίας προϊόντων, τότε στη βάση μας μπορούμε να έχουμε έναν πίνακα (πλειάδα) των χρηστών όπου θα αποθηκεύονται οι πελάτες μας και έναν πίνακα (πλειάδα) των παραγγελιών όπου θα αποθηκεύονται οι παραγγελίες τους. Οι σχέση των δύο αυτών πινάκων είναι 1 προς πολλά (1:N) επειδή ένα πελάτης μπορεί να κάνει πολλές παραγγελίες.

## 2.4 Apache Web Server

Ο Apache, είναι ένας ανοιχτού κώδικα διακομιστής Διαδικτύου. Δημιουργήθηκε και κυκλοφόρησε το 1995 από τον αμερικανικό Robert McCool κερδίζοντας γρήγορα την πρώτη θέση στην παγκόσμια αγορά των διακομιστών Διαδικτύου αφού στις αρχές του 21ου αιώνα, οι διακομιστές Apache κατείχαν πάνω από το 50% του περιεχομένου του Internet. Σήμερα παρέχει υπηρεσίες σε “κολοσσούς” του Διαδικτύου, όπως οι εταιρίες Wikimedia, Wikipedia, Google κ.α.

Ανταποκρινόμενος στο ρόλο του διακομιστή διαδικτύου, το Apache είναι υπεύθυνο για την αποδοχή από διαδικτυακούς χρήστες αιτημάτων τύπου HTTP και την αποστολή σε αυτούς των επιθυμητών πληροφοριών με μορφή αρχείων ή ιστοσελίδων. Μεγάλο μέρος κώδικα του λογισμικού των διαδικτυακών εφαρμογών, έχει σχεδιαστεί έτσι ώστε να συμπεριλαμβάνει και να λειτουργεί τα χαρακτηριστικά του Apache, ενώ δίνεται η δυνατότητα στους προγραμματιστές κώδικα διαδικτύου να μπορούν να χρησιμοποιήσουν μια έκδοση Apache για προεπισκόπηση και δοκιμή.

Διαθέτει επίσης αυξημένα επίπεδα ασφάλειας, ενώ παρέχει την ασφαλή δυνατότητα κοινής χρήσης αρχείων. Οι χρήστες μπορούν να τοποθετούν αρχεία στον ριζικό κατάλογο του Apache με σκοπό τον διαμοιρασμό τους με άλλους χρήστες.

Η Apache Software Foundation είναι η εταιρεία που διανέμει τη μοναδική άδεια χρήσης του λογισμικού Apache. Η συντήρηση και η ανάπτυξη του γίνεται από την παγκόσμια κοινότητα προγραμματιστών ανοικτού κώδικα, από τις συνεισφορές χρηστών παγκοσμίως, καθώς και από την Apache Software Foundation.

## **2.5 Η γλώσσα προγραμματισμού Java**

### **2.5.1 Οι στόχοι της Java**

Η γλώσσα προγραμματισμού Java έχει σχεδιαστεί έτσι ώστε να ανταποκρίνεται στις ανάγκες των προγραμματιστών, που αναπτύσσουν εφαρμογές σε ετερογενή και κατακευματισμένα συστήματα σε περιβάλλον δικτύου. Οι προδιαγραφές των εφαρμογών αυτών μεταξύ άλλων περιλαμβάνουν, την ελάχιστη κατανάλωση των πόρων του συστήματος, την λειτουργία τους σε οποιαδήποτε πλατφόρμα υλικού και λογισμικού και την δυνατότητα τους για δυναμική επέκταση. Η γλώσσα προγραμματισμού Java δημιουργήθηκε αποτελώντας ένα από τα κύρια εργαλεία ενός ερευνητικού έργου, που σκοπό είχε την ανάπτυξη προηγμένου λογισμικού, για μεγάλη ποικιλία συσκευών δικτύου καθώς και ενσωματωμένων συστημάτων. Στόχος ήταν η ανάπτυξη μιας πλατφόρμας πραγματικού χρόνου, που τα κύρια χαρακτηριστικά της θα ήταν το μικρό μέγεθος, η αξιοπιστία, η φορητότητα, η κατάτμηση και η λειτουργικότητα.

Όπως ήταν φυσιολογικό η γλώσσα που επιλέχθηκε αρχικά, ήταν η C. Στην πορεία του έργου όμως διαπιστώθηκε, ότι οι δυσκολίες της γλώσσας C δημιούργησαν προβλήματα τέτοια, που θα μπορούσαν να αντιμετωπιστούν καλύτερα με τη δημιουργία μιας εντελώς καινούργιας γλώσσας προγραμματισμού. Οι προδιαγραφές αρχιτεκτονικής και σχεδιασμού της νέας γλώσσας προέρχονταν από διάφορες γλώσσες όπως η Eiffel, η SmallTalk, η C και η Cedar / Mesa και αυτό είχε σαν αποτέλεσμα η καινούργια γλωσσική πλατφόρμα να είναι ιδανική για ανάπτυξη ασφαλών, κατανεμημένων, δικτυακών εφαρμογών τελικών χρηστών, σε περιβάλλοντα συσκευών που είναι ενσωματωμένες σε δίκτυο, στο Διαδίκτυο αλλά και στην επιφάνεια εργασίας.

Οι απαιτήσεις στη σχεδίαση της γλώσσας προγραμματισμού Java, στηρίζονται στις προδιαγραφές των υπολογιστικών περιβαλλόντων για τα οποία προορίζεται το λογισμικό. Η ευρεία χρήση του Διαδικτύου και του Παγκόσμιου Ιστού απαιτεί μια εντελώς νέα θεώρηση του τρόπου με τον οποίο αναπτύσσεται και διανέμεται πλέον το λογισμικό. Έτσι η τεχνολογία Java πρέπει να μπορεί να υποστηρίξει ανάπτυξη ασφαλών εφαρμογών, εξαιρετικά ισχυρών και υψηλής απόδοσης, που να μπορούν να τρέχουν σε πολλαπλές πλατφόρμες ετερογενών και κατανεμημένων δικτύων. Για να ανταποκριθεί στις σύγχρονες προκλήσεις, η γλώσσα προγραμματισμού Java θα έπρεπε να είναι αρχιτεκτονικά ουδέτερη, φορητή και να προσαρμόζεται δυναμικά. Έτσι έπρεπε να έχει τα εξής χαρακτηριστικά:

- **Απλότητα**, στη χρήση,
- **Ευκολία**, στην εκμάθηση,
- **Αντικειμενοστραφής**, ώστε να μπορεί να ανταποκριθεί στις σύγχρονες τεχνικές ανάπτυξης λογισμικού και να προσαρμοστεί σε κατανεμημένες εφαρμογές client-server,
- **Multithreaded**, για υψηλή απόδοση στην απαίτηση για εκτέλεση πολλαπλών ταυτόχρονων δραστηριοτήτων και
- **Ανεξαρτησία**, για φορητότητα σε διαφορετικά συστήματα και συσκευές.

### 2.5.2 Απλή Αντικειμενοστραφής και Εύκολη

Η γλώσσα προγραμματισμού Java είναι απλή και μπορεί να προγραμματιστεί εύκολα, χωρίς να χρειάζεται ιδιαίτερη εκπαίδευση των προγραμματιστών. Οι θεμελιώδεις έννοιες της Java είναι εύκολα κατανοητές και ένας προγραμματιστής μπορεί γρήγορα να τη χρησιμοποιήσει και να γράψει κώδικα.

Έχει σχεδιαστεί από την αρχή ώστε να είναι αντικειμενοστραφής. Η αντικειμενοστραφής τεχνική, αποτελεί το βασικότερο ίσως στοιχείο μιας σύγχρονης γλώσσας προγραμματισμού. Τα κατανεμημένα συστήματα πελάτη-εξυπηρετητή περιλαμβάνουν ενθυλακωμένα τμήματα λογισμικού που στηρίζονται στα αντικείμενα. Μια εφαρμογή, για να μπορέσει να λειτουργήσει ικανοποιητικά μέσα σε ένα πολύπλοκο δικτυακό περιβάλλον, πρέπει να έχει σχεδιαστεί έτσι ώστε να είναι αντικειμενοστραφής.

Η τεχνολογία Java παρέχει τη δυνατότητα σε έναν προγραμματιστή να είναι ιδιαίτερα αποδοτικός στην ανάπτυξη. Του δίνεται η δυνατότητα να έχει πρόσβαση και να χρησιμοποιεί υπάρχουσες βιβλιοθήκες έτοιμων και ήδη δοκιμασμένων αντικειμένων που περιλαμβάνουν από τους βασικούς τύπους δεδομένων, λειτουργίες I/O και διεπαφών δικτύου, έως γραφικά εργαλεία διεπαφής χρήστη. Αυτές οι βιβλιοθήκες μπορούν εύκολα να επεκταθούν και να προσαρμοστούν έτσι ώστε να παρέχουν νέα συμπεριφορά που να καλύπτει πλήρως τις ανάγκες του προγραμματιστή.

Παρόλο που η C απορρίφθηκε ως γλώσσα εφαρμογής, η γλώσσα προγραμματισμού Java περιλαμβάνει αρκετά από τα χαρακτηριστικά της C, αλλά η εξοικείωση του προγραμματιστή με τη Java γίνεται πολύ πιο εύκολα αφού δεν είναι τόσο περίπλοκη όσο η C, ενώ διατηρεί σχεδόν όλα τα αντικειμενοστραφή χαρακτηριστικά της και τα πλεονεκτήματα του περιβάλλοντος της C δίνοντας έτσι τη δυνατότητα στους προγραμματιστές να μπορούν να εξοικειώνονται εύκολα και να γίνονται γρήγορα παραγωγικοί.

### **2.5.3 Ισχυρή και Ασφαλής**

Η γλώσσα προγραμματισμού Java έχει σχεδιαστεί με σκοπό την παραγωγή αρκετά αξιόπιστου λογισμικού. Παρέχει δυνατότητες εκτεταμένου ελέγχου μεταγλώττισης, ενώ συνοδεύεται από έναν ισχυρό debugger. Τα χαρακτηριστικά της γλώσσας είναι τέτοια ώστε να καθοδηγούν τους προγραμματιστές να προγραμματίζουν με βάση τις σύγχρονες μεθόδους προγραμματισμού. Τα αντικείμενα δημιουργούνται με νέο τρόπο κάνοντας τη διαχείριση της μνήμης εξαιρετικά αποδοτική. Δεν χρειάζεται σαφής ορισμός από τον προγραμματιστή στη δημιουργία αντικειμένων τύπου δείκτη (pointer data types), γίνεται αυτόματη συλλογή απορριμμάτων (garbage collection), εξαλείφοντας έτσι ολόκληρες κατηγορίες σφαλμάτων που παρουσιάζονταν έως τώρα από τη C και την C++. Ο

προγραμματιστής είναι σίγουρος ότι δεν θα προκύψουν ιδιαίτερα προβλήματα μετά τη διανομή του κώδικα αφού το σύστημα πλέον μπορεί να ανιχνεύσει και να χειριστεί πολλά από τα σφάλματα.

Επειδή η τεχνολογία Java έχει σχεδιαστεί για να λειτουργεί σε κατανεμημένα περιβάλλοντα όπου η ασφάλεια είναι υψίστης σημασίας, επιτρέπει τη δημιουργία εφαρμογών ασφαλών από εξωτερικές επιθέσεις ενώ οι εφαρμογές γραμμένες σε Java μπορούν να τρέχουν σε περιβάλλον δικτύου παρέχοντας ταυτόχρονα ασφάλεια από ιούς, που μπορούν να θέσουν σε κίνδυνο το σύστημα ή τα αρχεία.

#### **2.5.4 Ανεξαρτησία και φορητότητα**

Η τεχνολογία Java υποστηρίζει εφαρμογές που προορίζονται για ετερογενή περιβάλλοντα δικτύου και πρέπει να είναι ικανές να εκτελούνται σε διαφορετικές αρχιτεκτονικές υλικού. Πρέπει επίσης, να μπορούν να εκτελούνται σε διαφορετικά λειτουργικά συστήματα και να συνεργάζονται με εφαρμογές γραμμένες σε άλλες γλώσσες και διαφορετικά περιβάλλοντα.

Για να μπορεί να ανταποκριθεί σε αυτή την ποικιλία λειτουργικών συστημάτων και αρχιτεκτονικών, ο Java Compiler παράγει bytecodes. Το bytecodes είναι μια ενδιάμεση μορφή κώδικα, ανεξάρτητη από αρχιτεκτονική, που έχει σχεδιαστεί έτσι ώστε να μπορεί να μεταφερθεί σε πολλές και διαφορετικές πλατφόρμες υλικού και λογισμικού. Σε αυτές τις διαφορετικές πλατφόρμες, βρίσκεται ο Java Interpreter που παράγει επιτόπου δυαδικό κώδικα, επιλύοντας έτσι τόσο το δυαδικό πρόβλημα διανομής, όσο και το πρόβλημα της έκδοσης. Με αυτόν τον τρόπο ο bytecode της Java εκτελείται σε οποιαδήποτε πλατφόρμα.

Εκτός από την ουδετερότητα στην αρχιτεκτονική, ένα ακόμα χαρακτηριστικό που βρίσκεται στον πυρήνα της Java είναι η φορητότητα. Τα μεγέθη των βασικών τύπων δεδομένων καθώς και η συμπεριφορά των αριθμητικών τελεστών της Java καθορίζονται έτσι, ώστε να μπορούν να συμπεριφερθούν με τον ίδιο τρόπο σε κάθε πλατφόρμα και να μην υπάρχουν ασυμβατότητες τύπου δεδομένων σε διαφορετικές αρχιτεκτονικές υλικού και λογισμικού. Αυτό το χαρακτηριστικό της τεχνολογίας Java το υλοποιεί μια αρχιτεκτονικά ουδέτερη και φορητή γλωσσική πλατφόρμα που είναι γνωστή ως Java Virtual Machine.

Η Java Virtual Machine, είναι μια αφηρημένη μηχανή για την οποία προορίζεται ο κώδικας που δημιουργούν οι προγραμματιστές της Java. Βασίζεται κυρίως στις προδιαγραφές της διεπαφής POSIX, που αποτελεί ένα βιομηχανικό πρότυπο μιας

διασύνδεσης ενός φορητού συστήματος, ενώ η υλοποίηση της σε νέες αρχιτεκτονικές είναι σχετικά απλή όταν ικανοποιούνται κάποια βασικά χαρακτηριστικά, όπως η πολυνηματική (multithread) υποστήριξη.

### **2.5.5 Υψηλή απόδοση συστήματος**

Ένα από τα κύρια χαρακτηριστικά, αν όχι το κυριότερο, ενός υπολογιστικού συστήματος, είναι η αποδοτικότητα. Ανταποκρινόμενη σε αυτό η Java παρέχει υψηλή απόδοση σε ένα σύστημα, χάρη στον interpreter, ο οποίος έχει σχεδιαστεί έτσι ώστε να μπορεί να τρέξει με πλήρη ταχύτητα χωρίς να εξαρτάται από την ταχύτητα του περιβάλλοντος εκτέλεσης.

Επίσης ο αυτόματος συλλέκτης απορριμμάτων (garbage collector), λειτουργεί ως διεργασία χαμηλής προτεραιότητας, γεγονός που εξασφαλίζει ότι η μνήμη θα είναι διαθέσιμη όταν απαιτείται, ενώ το σύστημα έχει με τον τρόπο αυτό καλύτερη απόδοση. Οι εφαρμογές που απαιτούν μεγάλη υπολογιστική ισχύ και καταλαμβάνουν πόρους του συστήματος, μπορούν εύκολα να σχεδιαστούν έτσι ώστε οι ενότητες που απαιτούν τους πόρους να μπορούν να γραφούν στον εγγενή κώδικα της μηχανής που φιλοξενούνται και να διασυνδεθούν με την πλατφόρμα Java.

Ένας χρήστης λοιπόν αντιλαμβάνεται ότι οι διαδραστικές εφαρμογές ανταποκρίνονται πολύ γρήγορα.

## **2.6 Content Management System**

Ένα CMS (Content Management System) ή Σύστημα Διαχείρισης Περιεχομένου είναι μια εφαρμογή η οποία επιτρέπει στο χρήστη την εύκολη και αποδοτική διαχείριση του περιεχομένου μιας ιστοσελίδας. Τέτοιες εφαρμογές χρησιμοποιούνται πολύ συχνά για την δημιουργία προσωπικών ιστοσελίδων, ιστοσελίδων εμπορικού περιεχομένου, blog αλλά και ηλεκτρονικών καταστημάτων eshop. Ο χρήστης μπορεί μέσω αυτών εύκολα και γρήγορα, να δημοσιεύσει άρθρα που περιέχουν κείμενο, πολυμέσα και καταλόγους προϊόντων και προσφορών σε ένα eshop.

Ο λόγος της επιτυχίας τους είναι η ευκολία που προσφέρουν στη διαχείριση του περιεχομένου, μέσω ενός ενιαίου περιβάλλοντος διαχείρισης και εμφάνισης, που περιλαμβάνει έναν κεντρικό πίνακα ελέγχου. Επιτρέπουν στους χρήστες να

δημοσιεύουν περιεχόμενο χωρίς να απαιτούνται γνώσεις προγραμματισμού HTML, ενώ μπορούν να αλλάξουν τη λειτουργικότητά τους εγκαθιστώντας κάποια πρόσθετα χαρακτηριστικά (plugins, addons, extentions) που είναι διαθέσιμα σε μια μεγάλη συλλογή.

Μία εφαρμογή CMS εγκαθίσταται στον διακομιστή που θα φιλοξενηθεί η ιστοσελίδα και γίνονται όλες οι απαραίτητες αρχικές ρυθμίσεις. Με την εγκατάσταση του CMS στο διακομιστή και κάνοντας όλες τις απαραίτητες ρυθμίσεις, σε πάρα πολύ μικρό χρονικό διάστημα, η ιστοσελίδα είναι διαθέσιμη στο κοινό. Επίσης άμεσα διαθέσιμη και εμφανής στο κοινό που επισκέπτεται την ιστοσελίδα, είναι κάθε αλλαγή που γίνεται από τον διαχειριστή της μέσω του CMS.

Τα γνωστότερα και πιο δημοφιλή Συστήματα Διαχείρισης Περιεχομένου (CMS), είναι το **WordPress**, το **Joomla** και το **Drupal** τα οποία είναι δωρεάν ανοιχτού λογισμικού εφαρμογές.



## ΚΕΦΑΛΑΙΟ 3

### ΚΑΤΗΓΟΡΙΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ (e-commerce)

Όπως αναφέρθηκε στην εισαγωγή, **ηλεκτρονικό εμπόριο** (Η.Ε.), ορίζεται το εμπόριο παροχής αγαθών και υπηρεσιών που πραγματοποιείται εξ αποστάσεως με ηλεκτρονικά μέσα. Βασίζεται στην ηλεκτρονική μετάδοση δεδομένων και δεν είναι αναγκαία η φυσική παρουσία των συμβαλλομένων μερών, δηλαδή του πωλητή και του αγοραστή.

Ανάλογα με τον τρόπο παράδοσης των προϊόντων, το ηλεκτρονικό εμπόριο διακρίνεται σε έμμεσο και άμεσο.

- **Έμμεσο** χαρακτηρίζεται το ηλεκτρονικό εμπόριο, όταν η παράδοση των προϊόντων της ηλεκτρονικής παραγγελίας γίνεται στον πελάτη μόνο με παραδοσιακούς τρόπους π.χ. ΕΛΤΑ, Courier κ.λ.π.

- **Άμεσο** χαρακτηρίζεται το ηλεκτρονικό εμπόριο, όταν η παραγγελία, η πληρωμή και η παράδοση των προϊόντων στον πελάτη, γίνεται online, δηλαδή σε ψηφιακή μορφή απευθείας από το διαδίκτυο.

Ανάλογα με τα συμβαλλόμενα μέρη των ηλεκτρονικών συναλλαγών, το ηλεκτρονικό εμπόριο διακρίνεται στις παρακάτω κατηγορίες:

- **B2C - Business to Consumer:** Όταν το ηλεκτρονικό εμπόριο γίνεται από επιχείρηση προς καταναλωτή.

- **B2B - Business to Business:** Όταν το ηλεκτρονικό εμπόριο γίνεται από επιχείρηση προς επιχείρηση.

- **G2C ή G2B – Government to Consumer ή Government to Business:** Όταν τα συναλλασσόμενα μέρη είναι η Δημόσια Διοίκηση και ένας πολίτης ή μια επιχείρηση.

- **C2C - Consumer to Consumer:** Όταν το ηλεκτρονικό εμπόριο γίνεται από καταναλωτή προς καταναλωτή

#### 3.1 Επιχείρηση με Καταναλωτή (Business to Consumer ή B2C)

Αφορά το λιανεμπόριο, δηλαδή τις πωλήσεις οι οποίες γίνονται από επιχειρήσεις σε καταναλωτές. Στην κατηγορία αυτή ανήκει οι πλειοψηφία των εφαρμογών του

Ηλεκτρονικού Εμπορίου που στόχο έχουν τις πωλήσεις προϊόντων από τις επιχειρήσεις απευθείας στους τελικούς καταναλωτές.

Μία επιχείρηση απευθύνεται στους καταναλωτές μέσω μιας ηλεκτρονικής πλατφόρμας που μπορεί να περιλαμβάνει τις παρακάτω δραστηριότητες - δυνατότητες:

- Ηλεκτρονική διαφήμιση και προώθηση προϊόντων ή υπηρεσιών
- Ηλεκτρονική παραγγελία
- Ηλεκτρονική πώληση προϊόντων ή υπηρεσιών
- Ηλεκτρονική πληρωμή
- Ηλεκτρονική υποστήριξη των πωλήσεων
- Ηλεκτρονική παρακολούθηση ροής αγαθών
- Ηλεκτρονική ενημέρωση και ψυχαγωγία

Την ανάπτυξη της αυτή η κατηγορία του ηλεκτρονικού εμπορίου, την οφείλει στην ανάπτυξη του παγκόσμιου ιστού και των τεχνολογιών των ασφαλών και αξιόπιστων πληρωμών μέσω Internet. Ο καταναλωτής συνδέεται online, δηλαδή απευθείας με τον υπολογιστή του στον υπολογιστή της επιχείρησης μέσω Internet, χωρίς την εμπλοκή κάποιας ενδιάμεσης υπηρεσίας. Η ολοκλήρωση όλων των συναλλαγών γίνονται ηλεκτρονικά σε πραγματικό χρόνο (real time).

Πολλά ηλεκτρονικά καταστήματα που προσφέρουν στον υποψήφιο πελάτη καταναλωτικά αγαθά, ηλεκτρονικές συσκευές, είδη ένδυσης, κοσμήματα κ.λ.π., αλλά και άυλες υπηρεσίες και προγράμματα (software), του δίνουν την δυνατότητα δοκιμής μέσω ειδικών προγραμμάτων από την οθόνη του υπολογιστή του, χωρίς καν να βγει από το σπίτι του κερδίζοντας έτσι πολύτιμο χρόνο και κόπο. Οι εταιρίες οι οποίες πρώτες εισέβαλαν στο χώρο του ηλεκτρονικού εμπορίου ήταν αυτές τις πληροφορικής, οι οποίες άνοιξαν διάπλατα την πόρτα του ηλεκτρονικού εμπορίου και στις υπόλοιπες.

Το Ηλεκτρονικό Εμπόριο αποτελεί έναν συμπληρωματικό τρόπο αγοράς προϊόντων, που όμως δεν καταργεί τον παραδοσιακό τρόπο της αγοράς, δηλαδή τη φυσική παρουσία των καταναλωτών στον χώρο της επιχείρησης. Αντίθετα μάλιστα, μπορεί και να τον ενισχύσει, αφού μια καλαίσθητη πλατφόρμα ηλεκτρονικού εμπορίου στο Internet, αποτελεί για την επιχείρηση έναν ακόμα καλύτερο τρόπο διαφήμισης. Το ανταγωνιστικό πλεονέκτημα που προσφέρει το ηλεκτρονικό εμπόριο στους καταναλωτές είναι:

- Χαμηλό Κόστος
- Ταχύτητα και Ποιότητα Εξυπηρέτησης
- Ευκολία Ανεύρεσης συγκεκριμένων προϊόντων



Εικόνα 3.1: Ο κύκλος του B2C

### 3.2 Επιχείρηση με Επιχείρηση (Business to Business ή B2B)

Αφορά το χονδρεμπόριο, δηλαδή τις συναλλαγές οι οποίες γίνονται μεταξύ των επιχειρήσεων. Ουσιαστικά πρόκειται για την αγορά προμηθειών και πρώτων υλών από επιχειρήσεις. Χαρακτηριστικό παράδειγμα είναι το σύστημα παραγγελιών πρώτων υλών ενός εργοστασίου ή μιας βιοτεχνίας από τους προμηθευτές τους, προκειμένου το εργοστάσιο ή η βιοτεχνία να παράξει το τελικό προϊόν, προς διάθεση στον τελικό καταναλωτή. Οι εταιρείες, οι βιοτεχνίες, ακόμα και οι εμπορικές επιχειρήσεις, χρησιμοποιούν ένα δίκτυο προκειμένου να προβούν σε παραγγελίες προϊόντων ή πρώτων υλών από προμηθευτές τους. Μέσω αυτού του δικτύου που μπορεί να είναι μια ηλεκτρονική πλατφόρμα, πραγματοποιούν τις παραγγελίες τους, ενημερώνονται για την εξέλιξη τους, προβαίνουν σε πληρωμές, εκτυπώνουν τα εκδιδόμενα τιμολόγια, και γενικά διευκολύνουν και επιταχύνουν τη ροή των εργασιών τους έχοντας όποια στιγμή το θελήσουν την απαραίτητη πληροφόρηση.

Επιχειρήσεις που δεν θα ακολουθήσουν αυτό τον τρόπο των ηλεκτρονικών παραγγελιών των πρώτων υλών τους από τους προμηθευτές τους, θα βρεθούν σε μειονεκτική θέση σε σχέση με τους ανταγωνιστές τους, αφού δεν θα μπορέσουν να

ανταποκριθούν με ταχύτητα στις αυξημένες απαιτήσεις των πελατών τους. Είναι όμως όλες οι επιχειρήσεις έτοιμες να ακολουθήσουν αυτό το σύστημα των ηλεκτρονικών συναλλαγών; Για να μπορέσει μια επιχείρηση να ανταποκριθεί πλήρως σε αυτή την καινοτομία θα πρέπει προηγουμένως να ικανοποιεί ορισμένες ενέργειες όπως:

- Προσαρμοσμένες και εστιασμένες έρευνες συμπεριφοράς καταναλωτών & επιχειρήσεων (πελατών & προμηθευτών)
- Μελέτη και προετοιμασία των εσωτερικών πληροφοριακών συστημάτων της ίδιας της επιχείρησης



Εικόνα 3.2: Η ροή πληροφοριών στο B2B

### 3.3 Κοινού με Κράτος (C2G ή B2G)

Μια κατηγορία ηλεκτρονικού εμπορίου η οποία αναπτύχθηκε στην Ελλάδα τα τελευταία χρόνια και συνεχώς αναπτύσσεται, είναι γνωστή ως ηλεκτρονική διακυβέρνηση (e-government). Πρόκειται για την παροχή της δυνατότητας ευρείας προσβασιμότητας σε πληροφορίες της Δημόσιας Διοίκησης με τη χρήση νέων τεχνολογιών και το Διαδίκτυο και περιλαμβάνει δύο μορφές ηλεκτρονικού εμπορίου:

- Η πρώτη αφορά συναλλαγές μεταξύ Πολίτη και Δημόσιας Διοίκησης (C2G). Στην Ελλάδα δείχνει να ξεπέρασε τα πρώτα στάδια και εξελίσσεται ραγδαία. Διάφοροι φορείς, κρατικοί ή μη αναπτύσσουν εφαρμογές με σκοπό την εξυπηρέτηση των πολιτών με τους δημόσιους φορείς.

Οι υπηρεσίες ηλεκτρονικής διοίκησης μπορούν να διακριθούν σε 4 επίπεδα:

1. Υπηρεσίες πληροφόρησης (παροχή ταξινομημένων πληροφοριών)
2. Υπηρεσίες επικοινωνίας (π.χ. διάθεση εντύπων)

3. Υπηρεσίες αμφίδρομης επικοινωνίας (π.χ. υποβολή δικαιολογητικών)
4. Υπηρεσίες διεκπεραίωσης (ολοκληρωμένες συναλλαγές)

Χαρακτηριστικό παράδειγμα αυτού του είδους των συναλλαγών, είναι το σύστημα TAXIS, το οποίο χρόνο με το χρόνο αναπτύσσεται ραγδαία, συμπεριλαμβάνοντας όλο και περισσότερες επιμέρους εφαρμογές, δίνοντας τη δυνατότητα στους πολίτες να πραγματοποιούν μέσω αυτού όλες σχεδόν τις φορολογικές τους συναλλαγές με το κράτος (φορολογικές δηλώσεις, φορολογικές ενημερότητες, έντυπο Ε9, τέλη κυκλοφορίας κλπ).

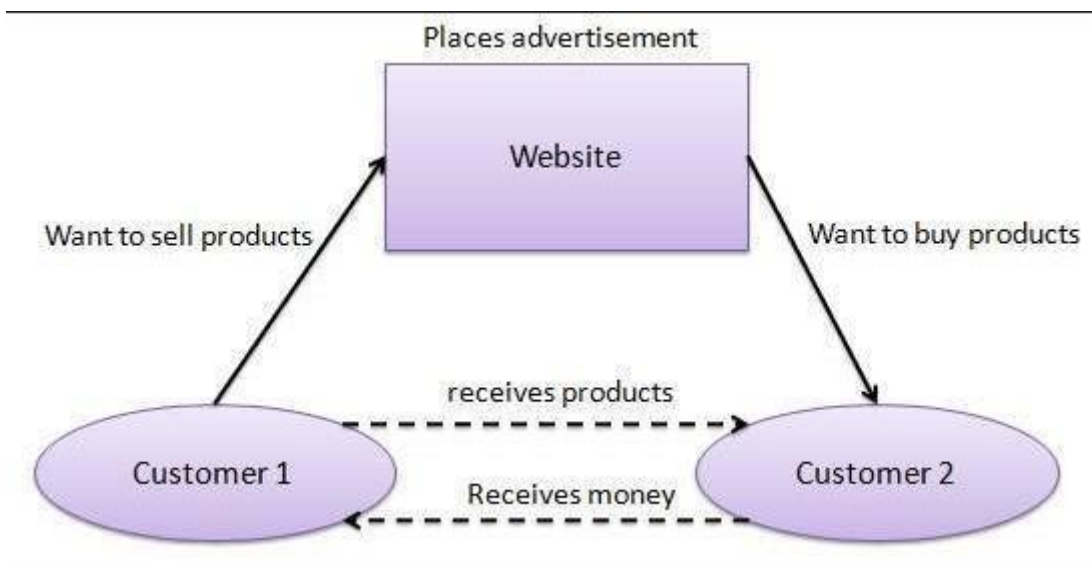
- Η δεύτερη μορφή αφορά συναλλαγές μεταξύ Επιχειρήσεων και Δημόσιας Διοίκησης (B2G). Πρόκειται στην ουσία για μια μορφή ηλεκτρονικής επικοινωνίας μεταξύ επιχειρήσεων και κράτους, τόσο για την ηλεκτρονική διεκπεραίωση φορολογικών ή άλλων υποχρεώσεων, όσο και για την αυτοματοποίηση της διαδικασίας των δημόσιων προμηθειών. Ουσιαστικά οι επιχειρήσεις ενημερώνονται ηλεκτρονικά από το κράτος για δημοπρασίες, διαγωνισμούς, προκηρύξεις, προκειμένου να υποβάλλουν και αυτές με τη σειρά τους ηλεκτρονικά τις αιτήσεις τους. Υπάρχει πλέον η δυνατότητα ηλεκτρονικής έκδοσης και υποβολής εντύπων, αυτοματοποίησης των συναλλαγών, ηλεκτρονικές πληρωμές, εκδόσεις πιστοποιητικών, ηλεκτρονικές πληρωμές οφειλών του δημοσίου μέσω e-banking κ.α. Η ανάπτυξη αυτής της μορφής του ηλεκτρονικού εμπορίου έχει πλεονεκτήματα όπως, μείωση λειτουργικών εξόδων, ταχύτητα και ποιότητα στην παροχή των υπηρεσιών, αποτελεσματικότερο έλεγχο εσόδων, διαφάνεια. Χαρακτηριστικό παράδειγμα και σε αυτή την περίπτωση, αποτελεί για τη χώρα μας το σύστημα TAXIS, με τη δυνατότητα που δίνει στις επιχειρήσεις για αυτοματοποιημένες ενέργειες όπως, η ηλεκτρονική υποβολή δηλώσεων Φ.Π.Α, e-παράβολο, υποβολή Φ.Μ.Υ, κ.α



Εικόνα 3.3: Η αλληλεπίδραση του Κοινού με το Κράτος

### 3.4 Καταναλωτή προς Καταναλωτή (Consumer to Consumer ή C2C)

Αυτή η κατηγορία ηλεκτρονικού εμπορίου, δίνει την δυνατότητα στον καταναλωτή της απευθείας πώλησης σε άλλους, προϊόντων, υπηρεσιών, ακόμα και προσωπικών αντικειμένων. Βρίσκει εφαρμογή σε ειδικούς διαδικτυακούς τόπους, όπου υπάρχουν αγοραπωλησίες αυτοκινήτων, μοτοσυκλετών, ακινήτων, μικρές αγγελίες για εύρεση ή προσφορά εργασίας, υπηρεσιών κλπ. Χαρακτηριστικά παραδείγματα αποτελούν οι διαδικτυακοί τόποι δημοπρασιών, όπου ο ενδιαφερόμενος μπορεί να καταχωρήσει και να πουλήσει ή να αγοράσει οτιδήποτε.



Εικόνα 3.4: Σύστημα C2C

### 3.5 Χαρακτηριστικά e-shop

Ένα ηλεκτρονικό κατάστημα, που αφορά την πώληση προϊόντων ή/και υπηρεσιών μέσω του διαδικτύου, πρέπει να διαθέτει τα ακόλουθα χαρακτηριστικά:

- **Κατάλογος προϊόντων/υπηρεσιών**

Απαραίτητη προϋπόθεση είναι να διαθέτει σε ηλεκτρονική μορφή, κατάλογο προϊόντων της επιχείρησης, που να περιλαμβάνει **τίτλο, περιγραφή, χαρακτηριστικά, υψηλής ανάλυσης εμπειριστατωμένη φωτογραφία, τιμή**, με ταξινόμηση των στοιχείων, προκειμένου να διευκολύνεται ο πελάτης στην αναζήτηση. Ο σημασία της ύπαρξης **υψηλής ανάλυσης εμπειριστατωμένης φωτογραφίας** για κάθε προϊόν είναι τεράστια, αφού βοηθά τον χρήστη να επιλέξει ευκολότερα τα προϊόντα που ικανοποιούν τις ανάγκες του. Ο κατάλογος πρέπει να

αντιστοιχίζεται σε ένα ή περισσότερα μενού, να είναι δομημένος σε κατηγορίες όπου η κάθε μία να περιλαμβάνει επιμέρους υποκατηγορίες, να βελτιώνεται και να ενημερώνεται διαρκώς. Πρέπει να συμπεριλαμβάνονται προϊόντα που υπάρχουν σε απόθεμα και όχι αυτά των οποίων η ποσότητα έχει εξαντληθεί. Σύμφωνα με τους Chen & Barnes (2007), *“η καθοδήγηση και η βοήθεια σε αγορές προϊόντων είναι ζωτικής σημασίας, καθώς με αυτό τον τρόπο αποκτάται η εμπιστοσύνη και χτίζεται η αφοσίωση των πελατών του καταστήματος”*.

#### • Αναζήτηση

Σημαντικό εργαλείο για τον δυνητικό πελάτη ενός ηλεκτρονικού καταστήματος, είναι η ύπαρξη-ενσωμάτωση μηχανής αναζήτησης, ώστε να μπορούν οι επισκέπτες να αναζητήσουν, μέσω λέξεων-κλειδιών (keywords) ή χαρακτηριστικών που τα περιγράφουν, και να βρουν άμεσα όλα τα διαθέσιμα προϊόντα που τους ενδιαφέρουν.

#### • Αναλυτική παρουσίαση

Πριν ο πελάτης ολοκληρώσει μια αγορά, θέλει απαραίτητα να γνωρίζει κάποια σημαντικά στοιχεία της “εν δυνάμει” παραγγελίας του, όπως η τιμή των προϊόντων που έχει επιλέξει, το συνολικό κόστος της παραγγελίας του, οι τρόποι πληρωμής της, ο φόρος που αντιστοιχεί, τα έξοδα αποστολής κ.α. Χρειάζεται όμως ιδιαίτερη προσοχή, γιατί τόσο η ελλιπής όσο και η υπερβολική πληροφόρηση, μπορεί να έχει τα ακριβώς αντίθετα από τα επιδιωκόμενα για τα ηλεκτρονικά καταστήματα αποτελέσματα και να οδηγήσουν αποτρεπτικά τον δυνητικό πελάτη να συνεχίσει στην ιστοσελίδα της επιχείρησης.

#### • Το καλάθι αγορών

Ένα από τα σημαντικότερα χαρακτηριστικά ενός ηλεκτρονικού καταστήματος, είναι το “καλάθι αγορών”. Η χρήση και η έννοιά του στα ηλεκτρονικά καταστήματα, είναι αντίστοιχη με εκείνη των φυσικών πολυκαταστημάτων. Και στις δύο περιπτώσεις “τοποθετούνται” τα επιλεγμένα από τον καταναλωτή προϊόντα. Διατηρώντας αυτή τη λογική του “καλαθιού αγορών” ένα ηλεκτρονικό κατάστημα, επιτρέπει στους καταναλωτές να τοποθετούν αρχικά σε αυτό τα προϊόντα της επιλογής τους και κατόπιν να πραγματοποιούν την αγορά τους. Πρέπει να δίνεται η δυνατότητα προσθήκης νέων προϊόντων καθώς και αφαίρεση ήδη τοποθετημένων, με ταυτόχρονη προβολή της ποσότητας του κάθε είδους και της συνολικής τιμής τους με και χωρίς Φ.Π.Α.

### • **Λογαριασμός πελάτη**

Πολλά ηλεκτρονικά καταστήματα, προκειμένου να επιτρέψουν στο χρήστη-πελάτη να πραγματοποιήσει μια αγορά, απαιτούν από αυτόν να δημιουργήσει έναν “λογαριασμό χρήστη”. Για να γίνει αυτό, απαιτείται από το χρήστη να καταχωρήσει ορισμένα στοιχεία, όπως το όνομα, το επώνυμο, το e-mail, τη διεύθυνσή του, τη χώρα κατοικίας του, τηλέφωνο επικοινωνίας κ.α. Τα στοιχεία αυτά, στέλνονται αρχικά σαν αίτημα για απόκτηση λογαριασμού από το χρήστη στο ηλεκτρονικό κατάστημα και κατόπιν αφού το αίτημα αυτό εγκριθεί και δημιουργηθεί ο λογαριασμός, τότε και μόνο μπορεί να ολοκληρωθεί η αγορά. Σύμφωνα με έρευνες, η εγγραφή του πελάτη και η δημιουργία λογαριασμού στην ηλεκτρονική πλατφόρμα, προσδίδει αίσθημα ασφάλειας στους χρήστες-πελάτες.

### • **Πραγματοποίηση On-line αγοράς**

Μετά την τοποθέτηση από τον χρήστη των προϊόντων της επιλογής του στο καλάθι αγορών, το επόμενο βήμα που πρέπει να γίνει είναι η αγορά τους. Για την αγορά των προϊόντων του καλάθιού αγορών, πρέπει να γίνουν τα εξής βήματα:

- Έλεγχος των προϊόντων και του συνολικού κόστους αυτών χωρίς και με ΦΠΑ.
- Επιλογή ενός από τους διαθέσιμους τρόπους αποστολής και παραλαβής του προϊόντος.
- Έλεγχος του συνολικού κόστους της παραγγελίας συμπεριλαμβανομένου και το κόστος αποστολής.
- Επιλογή ενός από τους διαθέσιμους τρόπους πληρωμής (π.χ. αντικαταβολή, χρήση πιστωτικής κάρτας, χρήση χρεωστικής κάρτας, paypal κ.λ.π.).
- Πραγματοποίηση της πληρωμής και εκτύπωση των στοιχείων της παραγγελίας.

### • **Ιστορικό παραγγελιών πελάτη**

Μία υπηρεσία που παρέχεται επίσης στους πελάτες από όλα σχεδόν τα σύγχρονα ηλεκτρονικά καταστήματα, είναι το “ΙΣΤΟΡΙΚΟ ΠΑΡΑΓΓΕΛΙΩΝ ΠΕΛΑΤΗ”. Για να μπορεί ο πελάτης να έχει πρόσβαση σε αυτό, θα πρέπει πριν, για λόγους ασφαλείας, να έχει συνδεθεί στο λογαριασμό που ο ίδιος έχει δημιουργήσει προκειμένου να πραγματοποιήσει τις αγορές του στην ηλεκτρονική πλατφόρμα. Εκεί έχει τη δυνατότητα ελέγχου:



- Ημιτελών παραγγελιών, δηλαδή προϊόντων που να μην έχουν προστεθεί στο καλάθι αγορών, αλλά δεν έχει πραγματοποιηθεί ακόμα την αγορά τους.
- Εξέλιξη εκκρεμών παραγγελιών, δηλαδή αγορών που έχει ήδη πραγματοποιήσει και εκκρεμεί η παράδοση των προϊόντων.
- Παλαιότερων αγορών, δηλαδή το ιστορικό όλων των αγορών που έχει πραγματοποιήσει από το συγκεκριμένο κατάστημα, καθώς και εκτύπωση των παραστατικών τους.

#### • Ασφάλεια Συναλλαγών

Ένα σημαντικό στοιχείο το οποίο απασχολεί τόσο το ίδιο το ηλεκτρονικό κατάστημα όσο και τους πελάτες του, είναι η ασφάλεια των ηλεκτρονικών συναλλαγών. Λαμβάνοντας υπόψη αυτό, ένα ηλεκτρονικό κατάστημα είναι υποχρεωμένο να τηρεί όλους τους όρους ασφαλείας, αφού οι πελάτες τόσο κατά την δημιουργία του λογαριασμού τους, όσο και κατά την πραγματοποίηση των αγορών τους καλούνται να καταχωρήσουν σε αυτό ευαίσθητα προσωπικά δεδομένα όπως το ονοματεπώνυμό τους, διεύθυνση κατοικίας τους, τηλέφωνα, ΑΦΜ, στοιχεία πιστωτικών καρτών κλ.π.

Σύμφωνα με τους Hoffman, Novak & Peralta (1999), *“Η ασφάλεια στις συναλλαγές και των προσωπικών δεδομένων, συμβάλουν στη δημιουργία εμπιστοσύνης μεταξύ ηλεκτρονικού καταστήματος και πελάτη”*.

#### • Λειτουργίες διευκόλυνσης αγορών

Οι επιχειρήσεις, προκειμένου να ανταπεξέλθουν στον διαρκώς αυξανόμενο ανταγωνισμό και προκειμένου να προσελκύσουν τους πελάτες τους, υιοθετούν τη χρήση διαφόρων εργαλείων τα οποία τους βοηθούν στην επιλογή του κατάλληλου γι αυτούς προϊόντος. Τέτοια εργαλεία είναι, η παρουσίαση στην αρχική σελίδα νέων προϊόντων, η δυνατότητα σύγκρισης με άλλα ομοειδή, σύντομα διαφημιστικά σποτάκια, παρουσίαση φωτογραφιών από πολλές οπτικές γωνίες κάποιου προϊόντος, εργαλεία μεγέθυνσης, εύρεση κατάλληλου μεγέθους κ.α.

## ΚΕΦΑΛΑΙΟ 4

### ΟΙ ΔΗΜΟΦΙΛΕΣΤΕΡΕΣ ΠΛΑΤΦΟΡΜΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

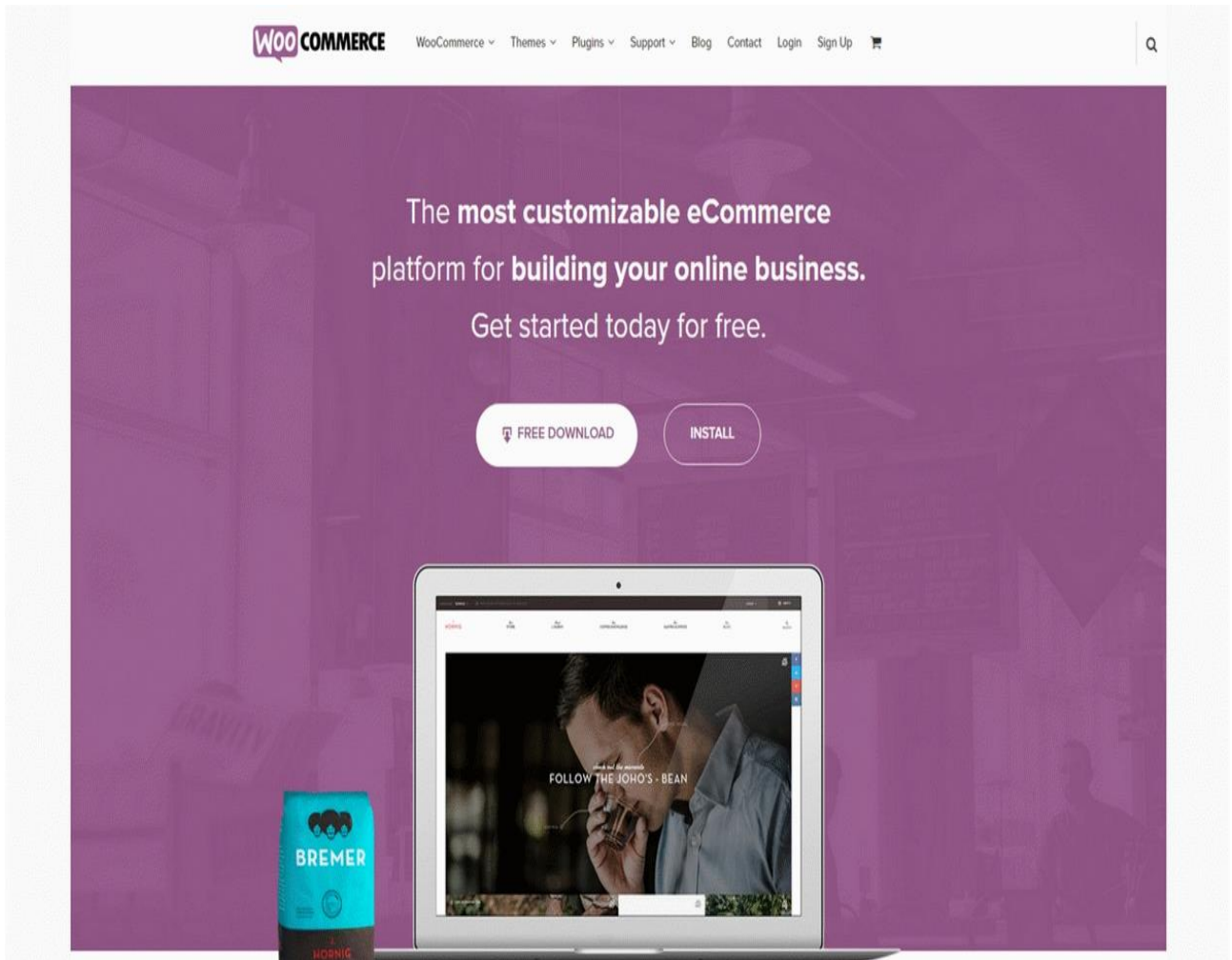
Στο κεφάλαιο αυτό, θα εξεταστούν οι δημοφιλέστερες πλατφόρμες ηλεκτρονικού εμπορίου ανοικτού κώδικα (open source software), καθώς και η δομή των πινάκων τους που περιέχουν τα ΠΡΟΪΟΝΤΑ, τις ΚΑΤΗΓΟΡΙΕΣ και τους ΚΑΤΑΣΚΕΥΑΣΤΕΣ. Οι οντότητες αυτές είναι οι βασικότερες, αφού περιέχουν τα στοιχεία εκείνα τα οποία συνήθως αναζητούν οι πελάτες που τις επισκέπτονται, προκειμένου να ενημερωθούν γι αυτά και να πραγματοποιήσουν τις αγορές τους.

Συχνά ένας επιχειρηματίας, έρχεται αντιμέτωπος με το εξής ερώτημα: Ποια πλατφόρμα να επιλέξω για την κατασκευή ενός ηλεκτρονικού καταστήματος (e-shop) με σκοπό την προώθηση και πώληση των προϊόντων μου ή των υπηρεσιών μου; Το ερώτημα προκύπτει από τη στιγμή που υπάρχουν αρκετές επιλογές στην αγορά, που μπορούν να αποδώσουν στην πραγματοποίηση αυτού του εγχειρήματος. Αρκετοί είναι και αυτοί που θεωρούν ότι ένα ηλεκτρονικό κατάστημα, εκτός από το να ενισχύσει το φυσικό τους κατάστημα, αποτελεί έναν εξωστρεφή τρόπο λειτουργίας της επιχείρησης, που μπορεί να του δώσει τρόπους διεξόδου από τυχόν αγκυλώσεις, που φέρνει η φυσική έδρα. Για όλους αυτούς τους λόγους, ένας μεγάλος αριθμός επιχειρηματιών, κυρίως μικρών αλλά και μεσαίων επιχειρήσεων, επιδιώκουν την αύξηση των πωλήσεών τους στοχεύοντας σε πελάτες από την αγορά του διαδικτύου. Αξίζει να σημειωθεί ότι, σε καιρούς κρίσης αυτή η αγορά, παρουσιάζει πρωτοφανή αύξηση ενδιαφέροντος από την πλευρά του αγοραστικού κοινού.

Παρακάτω, παρατίθενται οι πιο δημοφιλείς πλατφόρμες δημιουργίας ηλεκτρονικού καταστήματος ανοικτού κώδικα, οι οποίες μπορούν να ανταποκριθούν σε κάθε επιχειρηματικό σχέδιο.

#### 4.1 WooCommerce

Το WooCommerce είναι ίσως η πιο δημοφιλής πλατφόρμα ηλεκτρονικού εμπορίου στον κόσμο. Είναι ένα πρόσθετο (plugin) του επίσης δημοφιλούς Συστήματος Διαχείρισης Περιεχομένου (CMS), WordPress και αποτελεί μια εξαιρετικά ευέλικτη λύση, κατάλληλη για μικρά και μεσαίου μεγέθους eshop και online καταλόγους προϊόντων. Χρησιμοποιείται παγκοσμίως από το 42% των eshop ποσοστό αρκετά υψηλό και αποτελεί την δημοφιλέστερη πλατφόρμα ecommerce.



Εικόνα 4.1: Αρχική σελίδα WooCommerce

#### 4.1.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ

##### WooCommerce

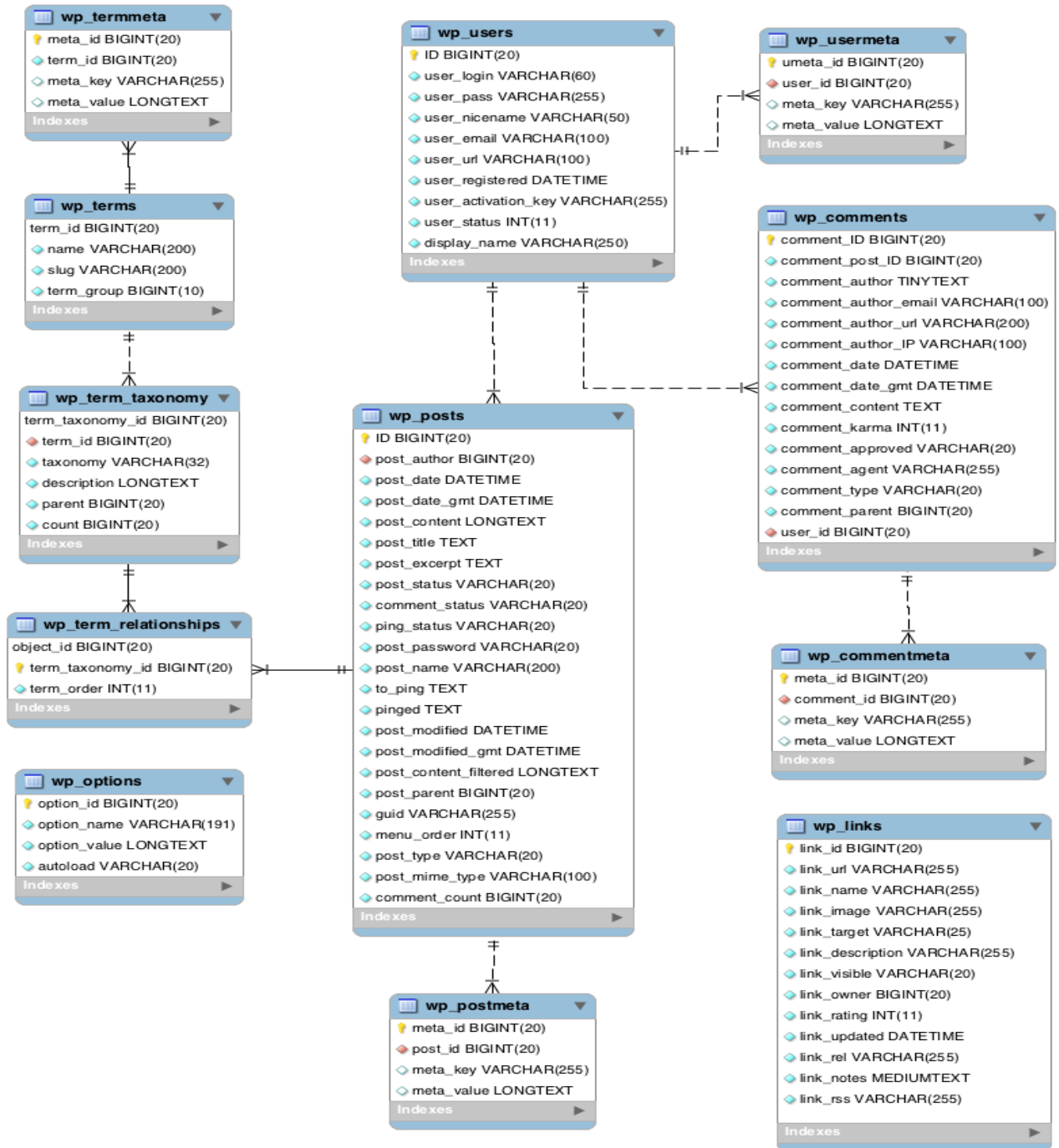
Τα προϊόντα, οι κατηγορίες και οι κατασκευαστές στην πλατφόρμα WordPress – WooCommerce, βρίσκονται κυρίως σε δύο πίνακες:

- **wp\_posts table** όπου το πεδίο `post_type` περιέχει την τιμή “product” για το προϊόν, ή την τιμή “product\_variation” για κάποια παραλλαγή ενός συγκεκριμένου κωδικού προϊόντος
- **wp\_postmeta table** όπου το πεδίο `post_id` περιέχει τον κωδικό προϊόντος (product ID).

Οι τύποι των προϊόντων (product types), οι κατηγορίες (categories), οι υποκατηγορίες (subcategories), οι ετικέτες (tags), τα χαρακτηριστικά (attributes) και οποιαδήποτε άλλη ταξινόμηση (custom taxonomy) αφορά συγκεκριμένο προϊόν και μπορεί να οριστεί από το χρήστη (π.χ. ο κατασκευαστής), περιέχεται στους παρακάτω πίνακες:

- **wp\_terms**
- **wp\_termmeta**
- **wp\_term\_taxonomy**
- **wp\_term\_relationships**
- **wp\_woocommerce\_termmeta**
- **wp\_woocommerce\_attribute\_taxonomies** (χαρακτηριστικά προϊόντος).

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce



Εικόνα 4.2: Δομή πινάκων WordPress

## 4.2 Magento

Η Magento δημιουργήθηκε και αναπτύχθηκε από τον όμιλο εταιρειών eBay, με σκοπό να παρέχει ένα ευέλικτο και εύκολα προσαρμόσιμο στις απαιτήσεις των επιχειρήσεων, ηλεκτρονικό κατάστημα. Μπορεί εύκολα να επεκταθεί ώστε να παρουσιάζονται, να διαχειρίζονται και να πωλούνται από 10 έως... 10.000 προϊόντα τουλάχιστον, σε διαρκώς αυξανόμενο αριθμό πελατών. Επίσης, το περιβάλλον διαχείρισής του (administrator panel) είναι πολύ καλά οργανωμένο, με εύκολη πλοήγηση, γεγονός που το κάνει να θεωρείται ένα από τα πιο ευκολονόητα και χρηστικά σε σύγκριση με αντίστοιχες e-commerce πλατφόρμες. Συνοψίζοντας όλα τα παραπάνω, μπορεί κάποιος να πει, ότι η ευελιξία μαζί με την προσαρμοστικότητα και την επεκτασιμότητά του, αποτελούν τα μεγαλύτερα πλεονεκτήματα της πλατφόρμας Magento. Όλα αυτά τα χαρακτηριστικά τη φέρνουν στην δεύτερη θέση παγκοσμίως με ποσοστό χρήσης 13% και στην πρώτη θέση σε μεγάλα e-shop.

**INTRODUCING MAGENTO ANALYTICS**  
Breaking news: Magento acquires RJMetrics. Optimize your business with cloud analytics for commerce.

[READ THE PRESS RELEASE](#) [LEARN MORE](#)

**#1 PLATFORM IR TOP 1000 - 2015**  
For the third year in a row, Magento has powered more Internet Retailer Top 1000 merchants than anyone else—more than double the amount of any

**#1 PLATFORM IR B2B 300 - 2015**  
Magento has been named the top eCommerce platform on the inaugural Internet Retailer B2B eCommerce 300 guide, with 41 merchants that list

**#1 PLATFORM IR EUROPE 500 - 2016**  
Magento powers 2X more large, fast growing companies in Europe than anyone else—53 to be exact, versus 26 for the nearest platform.

Εικόνα 4.3: Αρχική σελίδα Magento

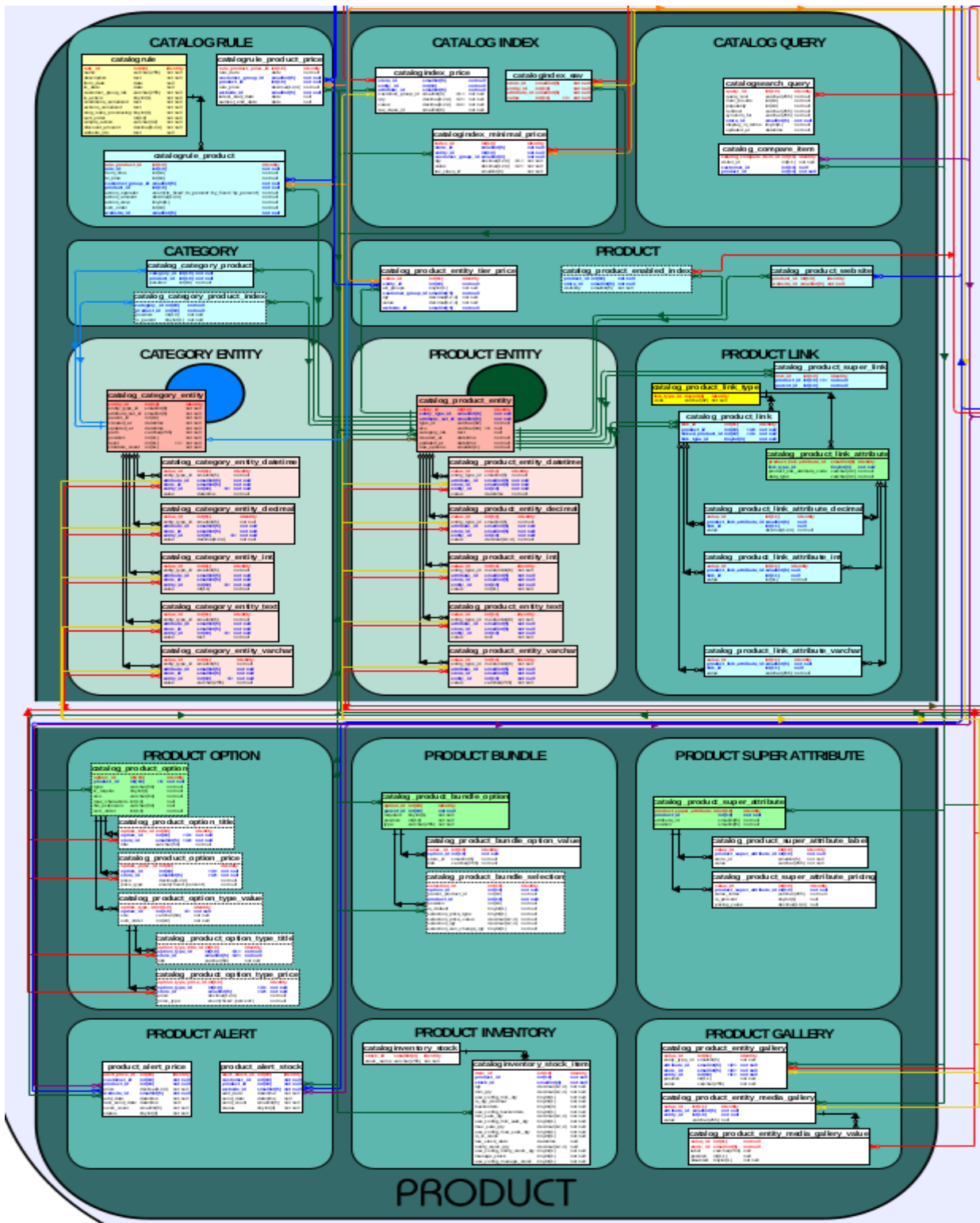
#### 4.2.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ Magento

Τα προϊόντα, οι κατηγορίες και οι κατασκευαστές στην πλατφόρμα Magento, βρίσκονται στους ακόλουθους πίνακες:

- **catalog\_product\_entity** - ο κύριος πίνακας προϊόντων. Περιέχει το id, sku, το σύνολο χαρακτηριστικών και μερικές άλλες λεπτομέρειες.
- **catalog\_product\_entity\_int | varchar | text | datetime | decimal** - περιέχουν τιμές χαρακτηριστικών. Κάθε χαρακτηριστικό έχει έναν τύπο και με βάση αυτόν τον τύπο η τιμή εισάγεται στον σωστό πίνακα.
- **catalog\_product\_website** - περιέχει τη σύνδεση προϊόντων σε ιστότοπους (id προϊόντος, id ιστότοπου). Αν δεν υπάρχουν προϊόντα σε τουλάχιστον έναν ιστότοπο, δεν υπάρχουν οι αντίστοιχες εγγραφές σε αυτόν τον πίνακα.
- **catalog\_category\_product** - περιέχει τη συσχέτιση μεταξύ προϊόντων και κατηγοριών (id προϊόντος, αναγνωριστικό κατηγορίας). Αν δεν υπάρχουν προϊόντα που αντιστοιχούν σε κατηγορίες, δεν υπάρχουν οι αντίστοιχες εγγραφές σε αυτόν τον πίνακα.
- **catalog\_category\_product\_index** - διατηρεί επίσης τη συσχέτιση μεταξύ προϊόντων και κατηγοριών και συμπληρώνεται αυτόματα μετά από ενημέρωση (reindexing).
- **catalog\_product\_entity\_gallery, catalog\_product\_entity\_media\_gallery, catalog\_product\_entity\_media\_gallery\_value** - περιέχουν αναφορές σε εικόνες προϊόντων.
- **catalog\_product\_link** - δεσμοί (cross-sells, up-sells) μεταξύ προϊόντων και συναφών προϊόντων (π.χ. παιδικά), καθώς και για ομάδες προϊόντων (product groups).
- **catalog\_product\_link\_attribute, catalog\_product\_link\_attribute\_decimal | int | varchar** - θέσεις και τιμές χαρακτηριστικών που μπορεί να σχετίζονται μεταξύ τους (up-sells, cross-sells).
- **catalog\_product\_entity\_tier\_price** - περιέχει τις τιμές των προϊόντων
- **catalog\_product\_option** - περιέχει προσαρμοσμένες επιλογές για προϊόντα
- **catalog\_product\_option\_price** - περιέχει τιμές για τις προσαρμοσμένων επιλογές

- **catalog\_product\_option\_title** - περιέχει τίτλους για τις προσαρμοσμένες επιλογές
- **catalog\_product\_option\_type\_value** - περιέχει τύπους για τις προσαρμοσμένες επιλογές
- **catalog\_product\_option\_type\_price** - περιέχει τιμές για τους τύπους των προσαρμοσμένων επιλογών
- **catalog\_product\_relation** - διατηρεί ένα είδος σχέσης γονέα-παιδιού για προϊόντα
- **catalog\_product\_index\_\*** - διατήρηση τιμών για τα χαρακτηριστικά του προϊόντος μετά από ενημέρωση (reindexing).
- **catalog\_product\_bundle\_\*** - σχέσεις και επιλογές για δέσμες προϊόντων
- **downloadable\_\*** - δεδομένα σχετικά με τα προϊόντα που μπορούν να γίνουν download

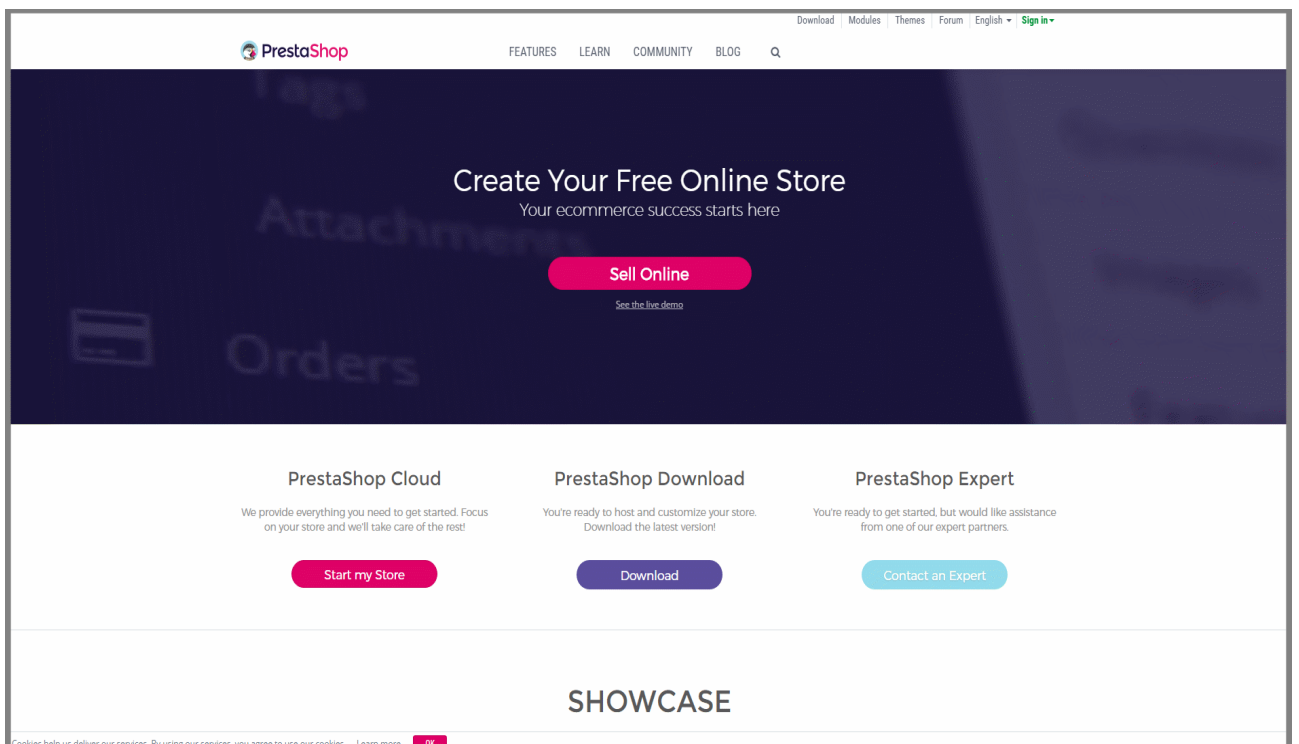




Εικόνα 4.4: Δομή πινάκων προϊόντων Magento

### 4.3 PrestaShop

Με ρίζες από τη Γαλλία, το PrestaShop αποτελεί την ευρωπαϊκή απάντηση στο Magento. Αυτό δικαιολογείται καθώς με βάση το ποσοστό χρήσης του παγκοσμίως, βρίσκεται στην τρίτη θέση με 9%, πλησιάζει τα ποσοστά χρήσης του Magento. Είναι ένα από τα κορυφαία software για eshop, αφού λόγω των προηγμένων λειτουργιών του το επιλέγει αρκετά μεγάλος αριθμός επιχειρήσεων. Εύκολα προσαρμόσιμο, μπορεί να υποστηρίξει από πολύ μικρά έως πολύ μεγάλου μεγέθους καταστήματα. Περιέχει έναν αριθμό ενσωματωμένων λειτουργιών, τις οποίες ο διαχειριστής μπορεί εύκολα μέσω του διαχειριστικού πίνακα (administration panel), να ενεργοποιήσει ή να απενεργοποιήσει. Διατίθεται σε πάνω από 40 γλώσσες, γεγονός που το κάνει να υποστηρίζεται αποδοτικά, με συνέπεια και διάρκεια από μια μεγάλη κοινότητα χρηστών. Μια επιχείρηση χρησιμοποιώντας το PrestaShop, μπορεί να έχει ένα οικονομικό, σύγχρονο και ενημερωμένο e-shop.



Εικόνα 4.5: Αρχική σελίδα PrestaShop

### 4.3.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ PrestaShop

Τα προϊόντα, οι κατηγορίες και οι κατασκευαστές στην πλατφόρμα PrestaShop, βρίσκονται στους ακόλουθους πίνακες:

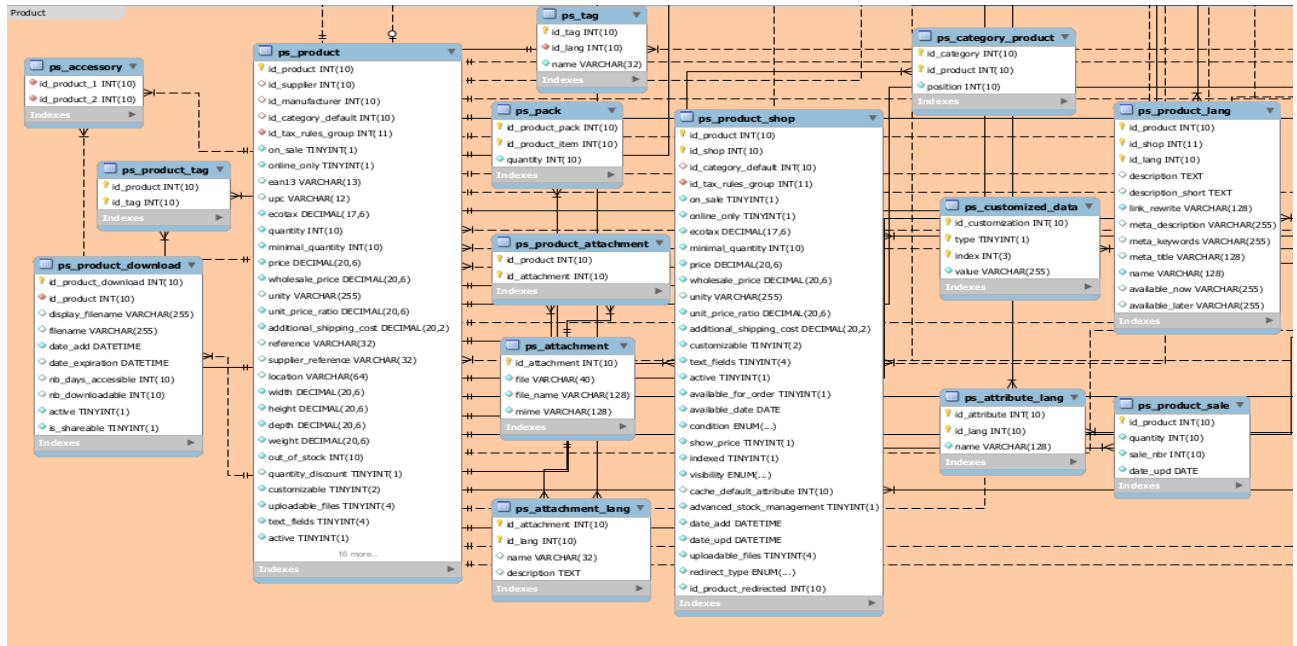
Πίνακας 4.1 Δομή πινάκων PrestaShop

Πίνακας	Περιγραφή
ps_accessory	Αξεσουάρ προϊόντων.
ps_address	Διευθύνσεις Κατασκευαστών.
ps_attachment	Παρατήματα προϊόντων.
ps_attachment_lang	Ονόματα παραρτημάτων.
ps_attribute	Χαρακτηριστικά προϊόντων.
ps_attribute_group	Ομάδες χαρακτηριστικών.
ps_attribute_group_lang	Ονόματα ομάδων χαρακτηριστικών.
ps_attribute_impact	Βάρη και τιμές ανά χαρακτηριστικό προϊόντος.
ps_attribute_lang	Ονόματα χαρακτηριστικών προϊόντων.
ps_category	Κατηγορίες προϊόντων.
ps_category_group	Ομάδες κατηγοριών προϊόντων.
ps_category_lang	Ονόματα ομάδων κατηγοριών προϊόντων.
ps_category_product	Προϊόντα ανά κατηγορία.
ps_customization_field	Προσαρμοσμένα πεδία προϊόντων.

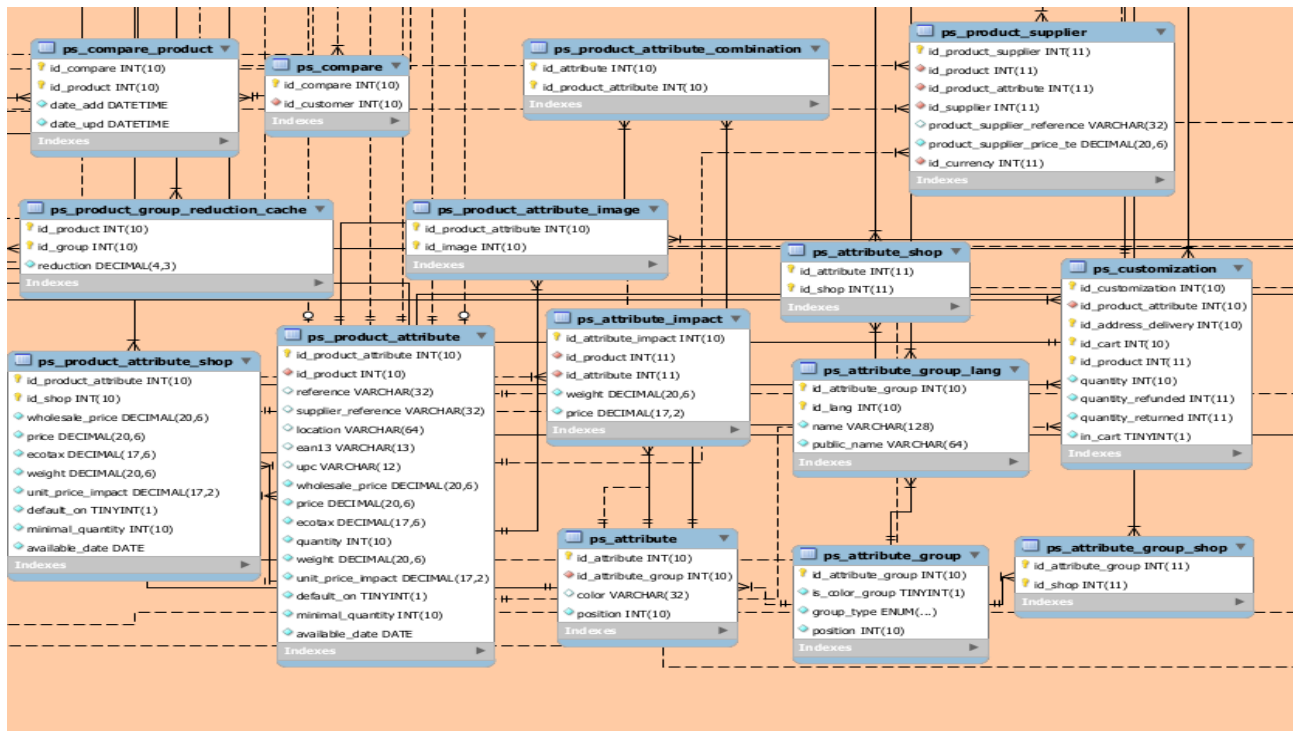
Πίνακας	Περιγραφή
ps_customization_field_lang	Ονόματα προσαρμοσμένων πεδίων προϊόντων.
ps_feature	Ιδιότητες προϊόντων.
ps_feature_lang	Ονόματα ιδιοτήτων προϊόντων.
ps_feature_product	Σύνδεση ιδιοτήτων με προϊόντα.
ps_feature_value	Τιμές ιδιοτήτων προϊόντων.
ps_feature_value_lang	Ονόματα τιμών ιδιοτήτων.
ps_image	Φωτογραφίες προϊόντων.
ps_image_lang	Περιγραφές εικόνων προϊόντων.
ps_image_type	Διαστάσεις μικρογραφιών εικόνων προϊόντων.
ps_manufacturer	Κατασκευαστές.
ps_manufacturer_lang	Περιγραφές κατασκευαστών.
ps_product	Πληροφορίες προϊόντων.
ps_product_attribute	Πληροφορίες χαρακτηριστικών προϊόντων.
ps_product_attribute_combination	Συνδυασμός χαρακτηριστικών προϊόντων.
ps_product_attribute_image	Εικόνες χαρακτηριστικών προϊόντων.
ps_product_download	Προϊόντα διαθέσιμα για download.

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

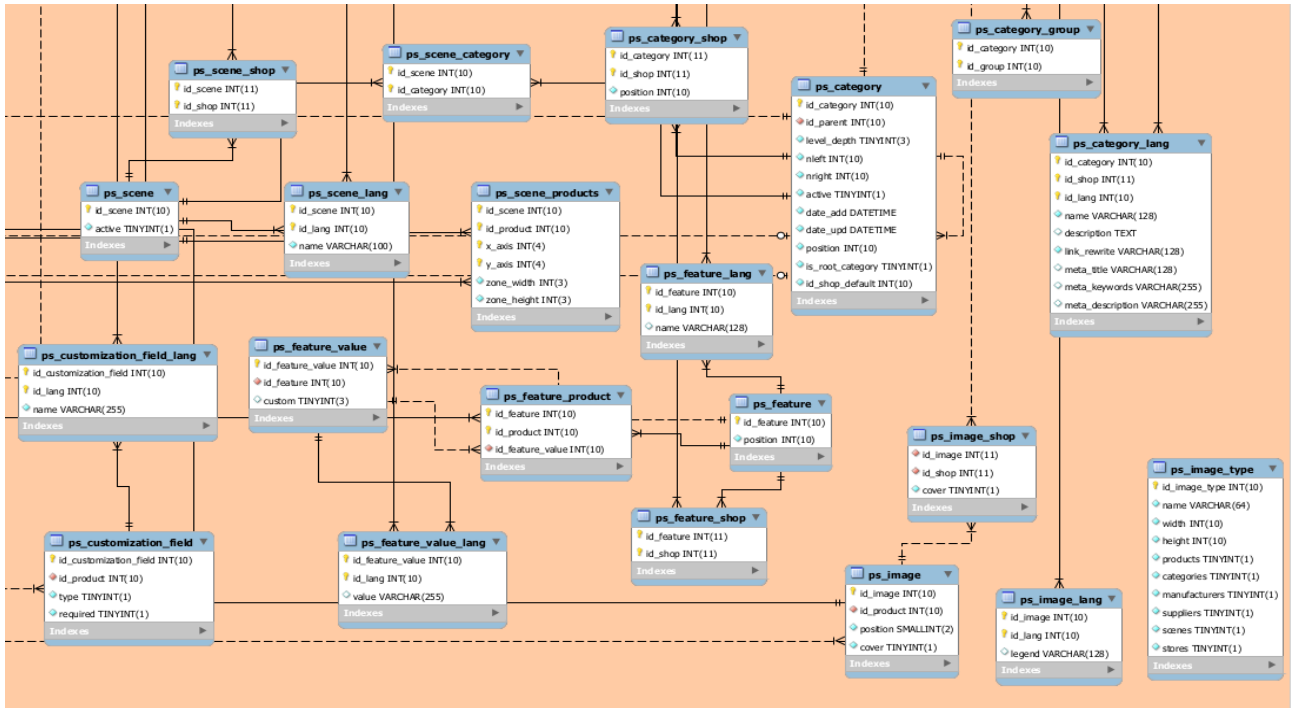
Πίνακας	Περιγραφή
ps_product_lang	Ονόματα και περιγραφές προϊόντων.
ps_product_tag	Ετικέτες προϊόντων.



Εικόνα 4.6.1: Δομή πινάκων προϊόντων PrestaShop



Εικόνα 4.6.2: Δομή πινάκων προϊόντων PrestaShop

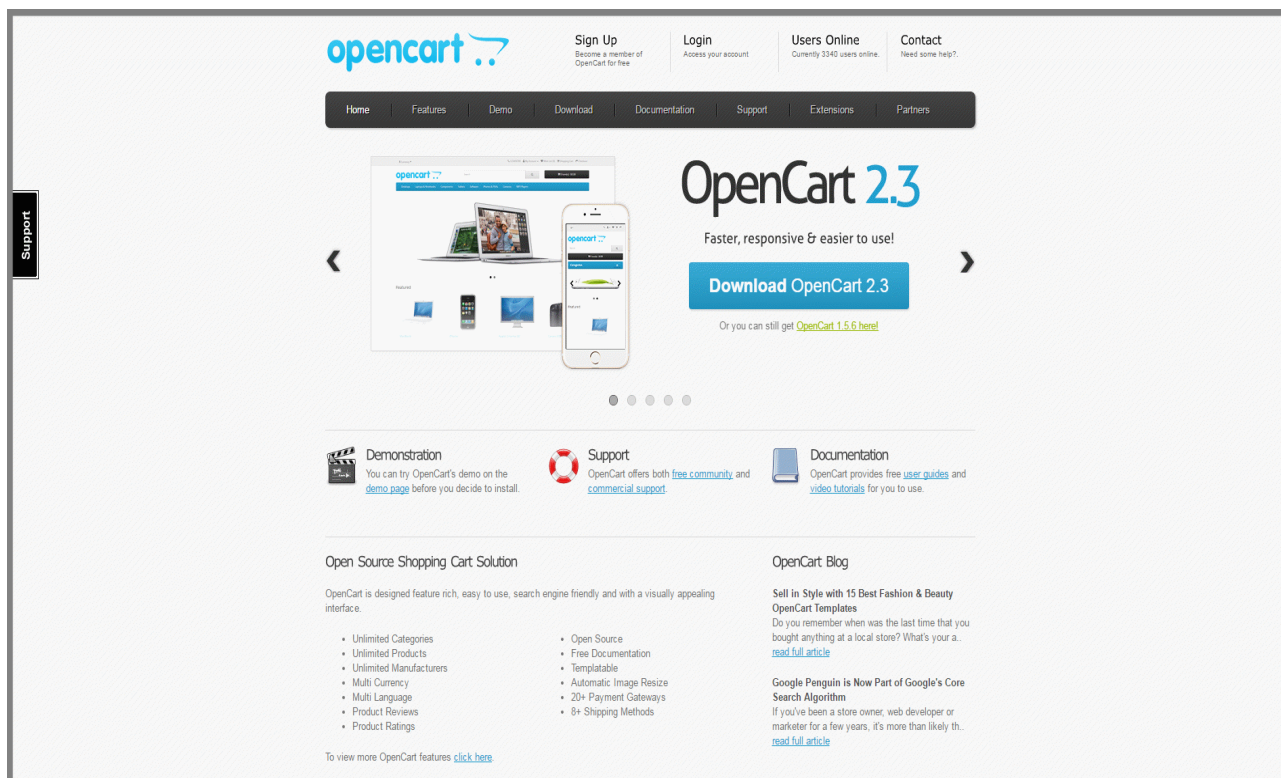


Εικόνα 4.6.3: Δομή πινάκων προϊόντων PrestaShop

#### 4.4 OpenCart

Το OpenCart, λόγω της απλότητας, της καλής σχεδίασης, και των δυνατοτήτων που παρέχει στην εύκολη διαχείριση προϊόντων και κατηγοριών, θεωρείται μία από τις κορυφαίες επιλογές για την κατασκευή ενός μικρού e-shop. Τα πρόσθετα (plugins) και οι επεκτάσεις του, είναι πολύ πιο φθηνά σε σύγκριση με το Magento, θεωρείται αρκετά “ελαφρύ” και μπορεί να φιλοξενηθεί ακόμα και σε μικρούς servers δίνοντας έτσι μια πολύ οικονομική επιλογή. Έχει μια ιδιαίτερα δραστήρια κοινότητα χρηστών, η οποία αυξάνεται συνεχώς. Τα templates και τα modules που μπορεί να βρει κάποιος, είναι υψηλής ποιότητας και πολλά από αυτά διατίθενται δωρεάν. Η δομή και οι διαδικασίες του είναι απλές και κατανοητές, η εκμάθησή του πολύ εύκολη, πράγμα που σημαίνει ότι ένας νέος χρήστης μπορεί να αρχίσει να χρησιμοποιεί το OpenCart άμεσα. Ο εύκολος διαχειριστικός πίνακας (administration panel), αποτελεί επίσης ένα από τα σημαντικότερα πλεονεκτήματα αυτής της πλατφόρμας.





Εικόνα 4.7: Αρχική σελίδα OpenCart

#### 4.4.1. Δομή πινάκων ΠΡΟΪΟΝΤΩΝ ΚΑΤΗΓΟΡΙΩΝ ΚΑΤΑΣΚΕΥΑΣΤΩΝ OpenCart

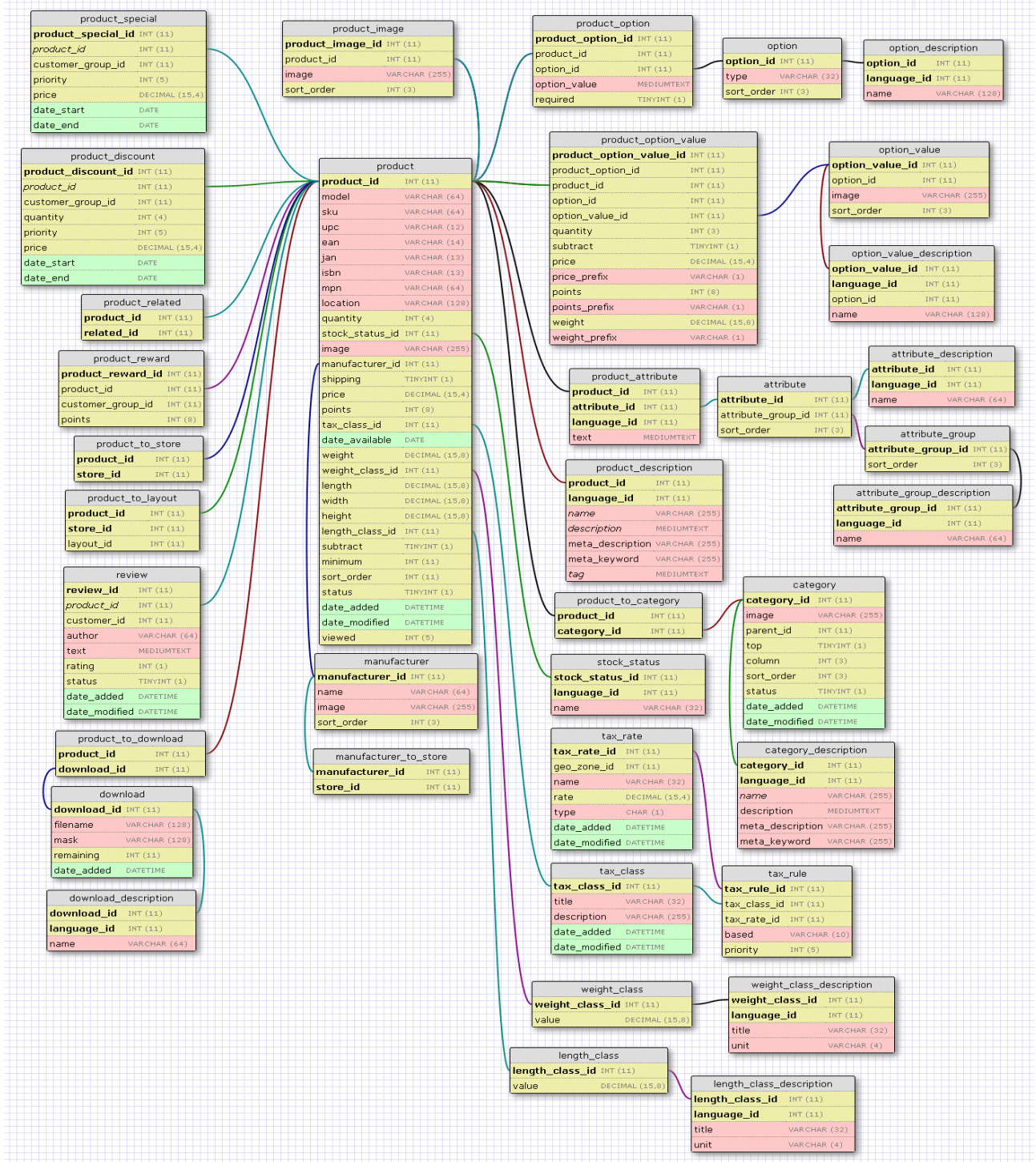
Τα προϊόντα, οι κατηγορίες και οι κατασκευαστές στην πλατφόρμα OpenCart, βρίσκονται στους ακόλουθους πίνακες:

- **category** – περιέχει τα id των κατηγοριών των προϊόντων. Περιέχει επίσης και τα `parent_id`, αν μια κατηγορία είναι υποκατηγορία άλλης κατηγορίας.
- **category\_description** – περιέχει τους τίτλους και τις περιγραφές των κατηγοριών των προϊόντων
- **category\_filter** – αντιστοιχία κατηγοριών σε φίλτρα, βάση των οποίων θα εμφανίζονται τα προϊόντα κάποιας κατηγορίας.
- **category\_path** – χρησιμοποιείται για την εμφάνιση των κατηγοριών – υποκατηγοριών σε δενδροειδή μορφή στο menu
- **category\_to\_layout** – αντιστοιχία μητρικών κατηγοριών σε επιλογή/ες του αρχικού menu
- **category\_to\_store** – αντιστοιχία κατηγοριών – υποκατηγοριών σε κατάστημα.
- **manufacturer** – περιέχει τα στοιχεία των κατασκευαστών (κωδικός, όνομα, εικόνα κ.λ.π.)

- **manufacturer\_to\_store** – αντιστοιχία κατασκευαστών σε κατάστημα.
- **product** – περιέχει ID's όλων των προϊόντων, model, sku, isbn, διαστάσεις, βάρος, τιμή απόθεμα, διαθεσιμότητες, κωδικοί φόρων, κ.λ.π.
  - **product\_attribute** – χαρακτηριστικά προϊόντων οριζόμενα από τον διαχειριστή, βάση των οποίων μπορεί ο πελάτης να επιλέξει προϊόντα.
  - **product\_description** – περιέχει τίτλους και περιγραφές των προϊόντων.
  - **product\_discount** – περιέχει εκπτώσεις για συγκεκριμένα προϊόντα ή ομάδες πελατών.
  - **product\_filter** – αντιστοιχία προϊόντων σε φίλτρα, βάση των οποίων θα εμφανίζονται.
  - **product\_image** – περιέχει τις φωτογραφίες των προϊόντων.
  - **product\_option** – πρόσθετες επιλογές που μπορεί να κάνει ένας πελάτης στη σελίδα του προϊόντος προτού το προσθέσει στο καλάθι αγορών.
    - **product\_option\_value** – τιμές των πρόσθετων επιλογών του πελάτη.
    - **product\_recurring** – επαναλαμβανόμενες πληρωμές για προϊόντα (συνδρομές κ.λ.π.)
    - **product\_related** – σχετικά προϊόντα που θα εμφανίζονται στον πελάτη όταν επιλέγει κάποιο προϊόν.
    - **product\_reward** – πόντοι ανταμοιβής πελατών για την αγορά προϊόντων από το κατάστημα.
    - **product\_special** – περιέχει ειδικές προσφορές για συγκεκριμένα προϊόντα ή ομάδες πελατών.
    - **product\_to\_category** – αντιστοιχία προϊόντων σε κατηγορίες του πίνακα category.
    - **product\_to\_download** – προϊόντα που είναι διαθέσιμα για download.
    - **product\_to\_layout** – αντιστοιχία προϊόντων σε επιλογή/ες του αρχικού menu.
    - **product\_to\_store** – αντιστοιχία προϊόντων σε κατάστημα.



## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce



Εικόνα 4.8: Δομή πινάκων προϊόντων OpenCart

## ΚΕΦΑΛΑΙΟ 5

### ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΗ ΤΩΝ ΕΡΓΑΛΕΙΩΝ

Στο κεφάλαιο αυτό, θα εξεταστεί η εγκατάσταση και η ρύθμιση των εργαλείων που απαιτούνται, προκειμένου να γραφεί και να εκτελεστεί ο απαραίτητος κώδικας σε java, για τη διασύνδεση μιας Βάσης Δεδομένων Microsoft SQL Server και της πλατφόρμας ηλεκτρονικού εμπορίου OpenCart.

Θα χρησιμοποιηθεί η Βάση Δεδομένων **Microsoft SQL Server Έκδοση 2017**, η πλατφόρμα ηλεκτρονικού εμπορίου **Opencart Έκδοση 3.0.3.1**, και τα προεγκατεστημένα στο **Ubuntu 18.04**, **PHP Version 7.2.15-0ubuntu0.18.04.1**, η **Βάση Δεδομένων MySql Έκδοση 14.14 Distrib 5.7.25 for Linux (x86\_64)** και **openjdk Έκδοση "10.0.2" 2018-07-17**. Η πλατφόρμα ηλεκτρονικού εμπορίου **Opencart Έκδοση 3.0.2.0**, θα εγκατασταθεί σε έναν **Web Server Apache2 Έκδοση Apache/2.4.29 (Ubuntu)**, ο οποίος είναι επίσης προεγκατεστημένος σε ένα σύστημα **Ubuntu 18.04**.

#### 5.1 Εγκατάσταση και ρύθμιση του Microsoft SQL Server σε Ubuntu 18.04

Για την εγκατάσταση και ρύθμιση των παραμέτρων του MS SQL Server Έκδοση 2017 σε ένα σύστημα Ubuntu 18.04, πρέπει να εκτελεστούν οι παρακάτω εντολές από το τερματικό, προκειμένου να εγκατασταθεί το πακέτο **mssql-server**:

1. Εισαγωγή δημόσιων κλειδιών (public repository GPG keys) της Microsoft:

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
```

2. Καταχώρηση του Microsoft SQL Server Ubuntu repository:

```
sudo add-apt-repository "$(wget -qO- https://packages.microsoft.com/config/ubuntu/18.04/mssql-server-2017.list)"
```

3. Εκτέλεση των ακόλουθων εντολών προκειμένου να γίνει η εγκατάσταση του SQL Server:

```
sudo apt-get update
```

```
sudo apt-get install -y mssql-server
```

4. Μετά την εγκατάσταση του SQL Server, εκτελείται η εντολή mssql-conf setup προκειμένου να οριστεί το SA password και να επιλεγεί η έκδοση:

```
sudo /opt/mssql/bin/mssql-conf setup
```

5. Μετά την εγκατάσταση και τη ρύθμιση εκτελείται η ακόλουθη εντολή προκειμένου να ελεγχθεί ότι ο MS SQL Server έχει εγκατασταθεί στο σύστημά μας και τρέχει χωρίς προβλήματα:

**systemctl status mssql-server**

**ΣΗΜΕΙΩΣΗ:** Αν η βάση δεδομένων MS SQL βρίσκεται σε απομακρυσμένο εξυπηρετητή και θέλουμε να έχουμε απομακρυσμένη σύνδεση, πρέπει να ελέγξουμε τις ρυθμίσεις του firewall ώστε να είναι ανοικτή η SQL Server TCP port (default 1433).

Για τη δημιουργία βάσεων δεδομένων σε έναν MS SQL Server, πρέπει να γίνει σύνδεση με ένα εργαλείο που μπορεί να εκτελεί εντολές Transact-SQL στον SQL Server. Το εργαλείο αυτό ονομάζεται **mssql-tools**, περιέχει τα **sqlcmd** και **bcp** και τα παρακάτω βήματα εγκαθιστούν αυτά εργαλεία γραμμής εντολών που χρησιμοποιούνται για αυτόν το σκοπό:

1. Εισαγωγή δημόσιων κλειδιών (public repository GPG keys):

**curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -**

2. Καταχώρηση του Microsoft Ubuntu repository:

**curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list | sudo tee /etc/apt/sources.list.d/msprod.list**

3. Ενημέρωση της λίστας πηγών (sources list) και εκτέλεση της εγκατάστασης με το unixODBC developer package:

**sudo apt-get update**

**sudo apt-get install mssql-tools unixodbc-dev**

**ΣΗΜΕΙΩΣΗ:** Για την ενημέρωση της τελευταίας έκδοσης του πακέτου *mssql-tools*, απαιτείται η εκτέλεση των παρακάτω εντολών:

**sudo apt-get update**

**sudo apt-get install mssql-tools**

Προκειμένου να συνδεθούμε τοπικά στον νέο SQL Server που μόλις εγκαταστήσαμε, τρέχουμε το **sqlcmd** με παραμέτρους το όνομα του Server (-S), το όνομα του χρήστη (-U), και τον κωδικό πρόσβασης (-P).

Αν θέλουμε να συνδεθούμε τοπικά το όνομα του Server είναι το localhost, ενώ το όνομα χρήστη είναι το SA και ο κωδικός πρόσβασης ο κωδικός που ορίστηκε για το χρήστη SA κατά τη διάρκεια της εγκατάστασης.

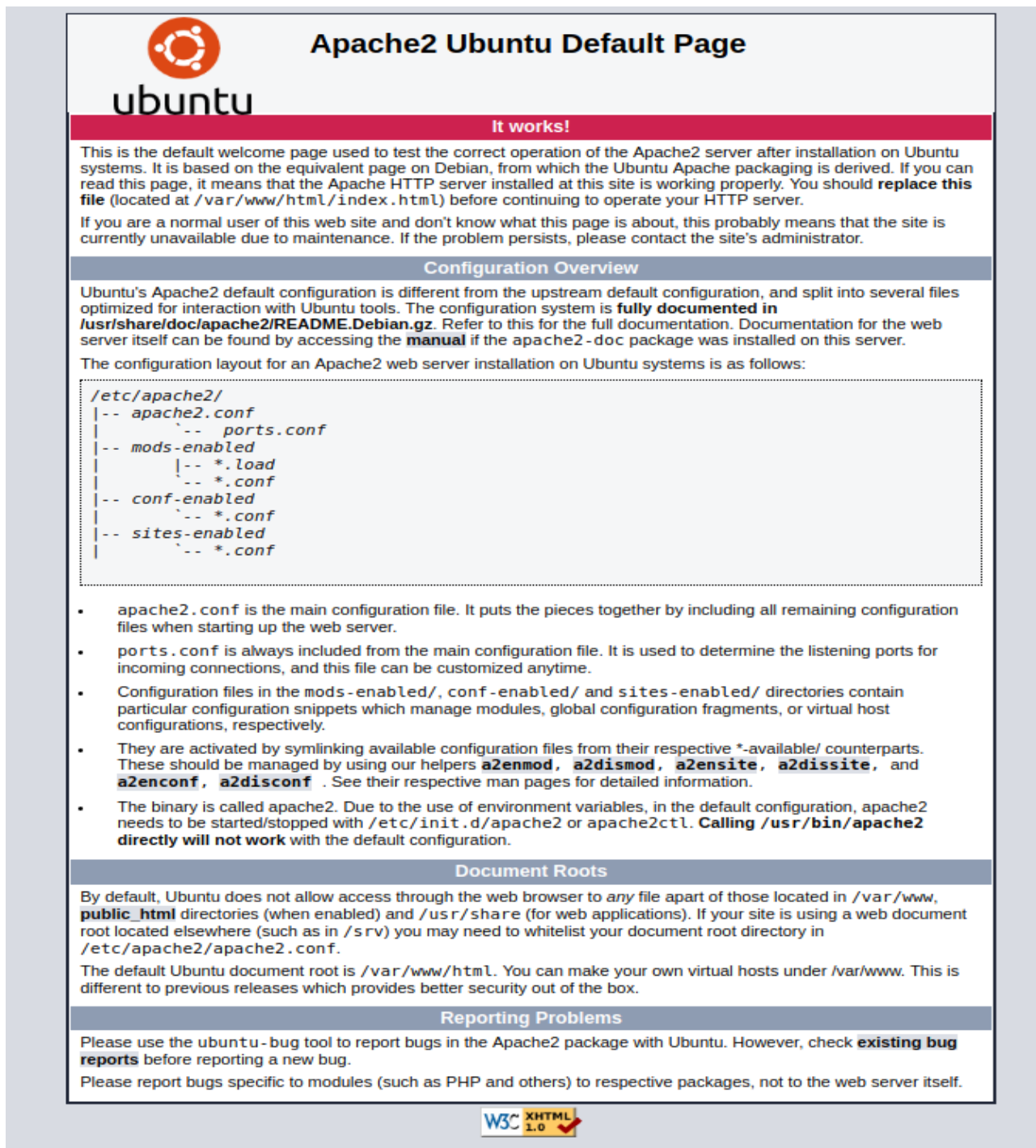
## 5.2 Ρύθμιση του Web Server Apache2 σε Ubuntu 18.04.

Με την εγκατάσταση του **Ubuntu 18.04** εγκαθίσταται και ο **Web Server Apache2** με τις ακόλουθες ρυθμίσεις οι οποίες περιλαμβάνονται στο αρχείο **000-default.conf**, στο φάκελο **etc/apache2/sites-available**:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName localhost
    ServerAlias localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Οι ρυθμίσεις αυτές περιλαμβάνουν το **όνομα του εξυπηρετητή (ServerName)**, καθώς και ένα **ψευδώνυμο (ServerAlias)**, το οποίο και στις δύο περιπτώσεις είναι το **localhost**. Επίσης περιλαμβάνουν και την διαδρομή του **αρχικού φακέλου (DocumentRoot)** που περιέχονται τα αρχεία του ιστότοπου. Όπως φαίνεται στο αρχείο 000-default.conf, η διαδρομή του DocumentRoot είναι η **var/www/html**. Μέσα σε αυτό το φάκελο περιέχονται δύο αρχεία.

- Το **index.html** που περιέχει την αρχική σελίδα του Apache2 και το
- **index.php**, που δείχνει όλες τις ρυθμίσεις που έχουν γίνει στον εξυπηρετητή, συμπεριλαμβανομένων και των ρυθμίσεων της **php** και **mysql**



Εικόνα 5.1: `http://localhost` - Αρχική σελίδα του Apache2

PHP Version 7.2.15-0ubuntu0.18.04.1	
<b>System</b>	Linux vasilis-P61-DS3-B3 4.15.0-46-generic #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 2019 x86_64
<b>Build Date</b>	Feb 8 2019 14:54:22
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/7.2/apache2
<b>Loaded Configuration File</b>	/etc/php/7.2/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/7.2/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-bcmath.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-soap.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
<b>PHP API</b>	20170718
<b>PHP Extension</b>	20170718
<b>Zend Extension</b>	320170718
<b>Zend Extension Build</b>	API320170718,NTS
<b>PHP Extension Build</b>	API20170718,NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	enabled
<b>Zend Memory Manager</b>	enabled
<b>Zend Multibyte Support</b>	provided by mbstring
<b>IPv6 Support</b>	enabled
<b>DTrace Support</b>	available, disabled
<b>Registered PHP Streams</b>	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar, zip
<b>Registered Stream Socket Transports</b>	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
<b>Registered Stream Filters</b>	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

Εικόνα 5.2: <http://localhost/index.php> – ρυθμίσεις Apache2. PHP, MySQL.

Για να είναι δυνατή η αναγνώριση του localhost σαν όνομα εξυπηρετητή, θα πρέπει, σε ένα σύστημα Ubuntu, να εισαχθεί στην αρχή η γραμμή “127.0.0.1 localhost”, στο αρχείο **hosts** που βρίσκεται στο **φάκελο /etc** έτσι ώστε να υπάρχει αντιστοιχία του επιθυμητού ονόματος του εξυπηρετητή, στην IP διεύθυνση 127.0.0.1.

Το αρχείο hosts θα έχει την παρακάτω μορφή:

```
127.0.0.1    localhost
127.0.1.1    vasilis-P61-DS3-B3
```

```
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

### 5.3 Λήψη και εγκατάσταση του OpenCart 3.0.3.1

1. Από τη διεύθυνση <https://www.opencart.com/index.php?route=cms/download>, της επίσημης ιστοσελίδας του OpenCart κατεβάζουμε στην τελευταία έκδοση του OpenCart (αρχείο opencart-3.0.3.1.zip).

2. Δημιουργούμε έναν φάκελο με όνομα **bookstore** στη διαδρομή **var/www/html** που βρίσκεται το DocumentRoot του Apache2 και αποσυμπιέζουμε εκεί τα περιεχόμενα από τον φάκελλο **upload** του αρχείου **opencart-3.0.3.1.zip**. Έτσι τώρα έχουμε τη διαδρομή **var/www/html/bookstore** με τα περιεχόμενα του φακέλλου upload σύμφωνα με την παρακάτω εικόνα:

Όνομα	Μέγεθος	Τύπος	Τροποποιήθηκε
admin	23,3 MB	Φάκελος	07 Ιανουαρίου 2019, 15:04
catalog	12,9 MB	Φάκελος	05 Ιανουαρίου 2019, 16:16
image	5,6 MB	Φάκελος	05 Ιανουαρίου 2019, 16:16
install	2,3 MB	Φάκελος	05 Ιανουαρίου 2019, 16:16
system	6,9 MB	Φάκελος	05 Ιανουαρίου 2019, 16:16
.htaccess.txt	2,4 KB	Έγγραφο a...	05 Ιανουαρίου 2019, 16:16
config-dist.php	0 bytes	Δέσμη ενερ...	05 Ιανουαρίου 2019, 16:16
index.php	293 bytes	Δέσμη ενερ...	07 Ιανουαρίου 2019, 15:05
php.ini	418 bytes	Configurati...	05 Ιανουαρίου 2019, 16:16

Εικόνα 5.3: Περιεχόμενα του φακέλλου bookstore στην διαδρομή `/var/www/html/bookstore/`

3. Μετονομάζουμε το αρχείο **config-dist.php** που βρίσκεται στο **φάκελλο bookstore** σε **config.php**. Κάνουμε το ίδιο και με το αρχείο **config-dist.php** βρίσκεται στον υποφάκελλο **admin** του φακέλλου **bookstore**.

4. Αντιγράφουμε το αρχείο **000-default.conf** που βρίσκεται στο **φάκελλο etc/apache2/sites-available** σε αρχείο με όνομα **bookstore.conf** και αλλάζουμε τα **ServerName**, **ServerAlias** και **DocumentRoot** σύμφωνα με τα παρακάτω:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName bookstore
    ServerAlias bookstore
    DocumentRoot /var/www/html/bookstore
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

5. Προσθέτουμε κάτω από τη γραμμή “127.0.0.1 localhost” του αρχείου **hosts** που βρίσκεται στο φάκελο **/etc**, τη γραμμή “127.0.0.1 bookstore” έτσι ώστε να υπάρχει αντιστοιχία του ονόματος εξυπηρετητή **bookstore**, στην IP διεύθυνση 127.0.0.1. Έτσι το αρχείο **hosts** θα έχει την παρακάτω μορφή:

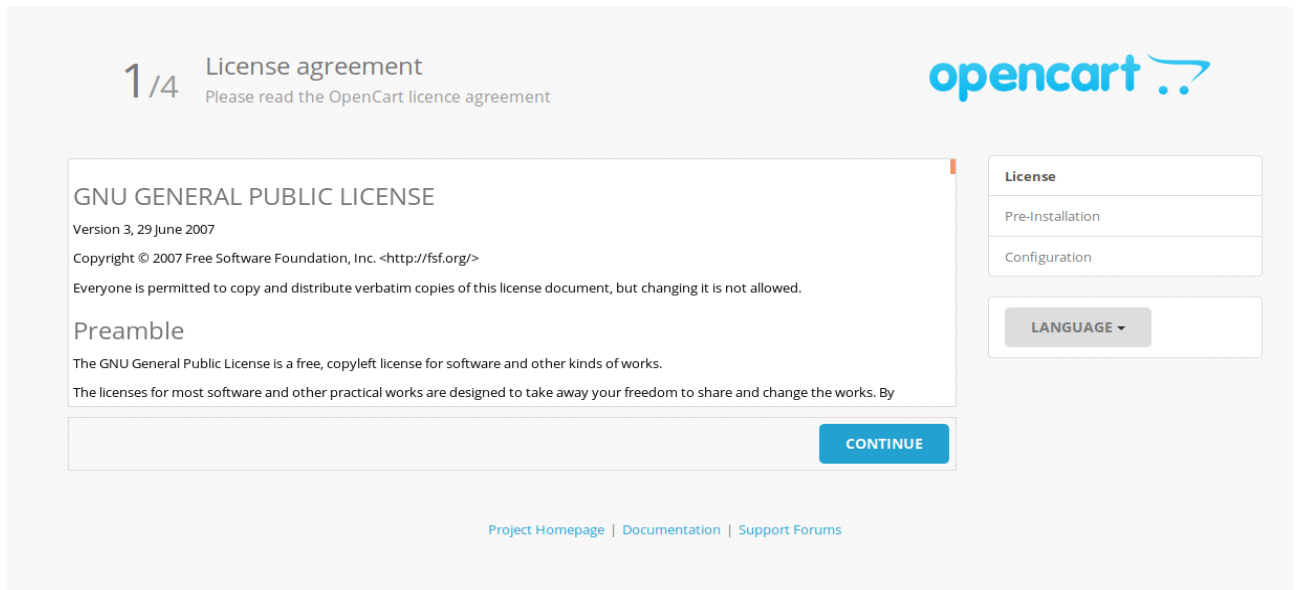
```
127.0.0.1    localhost
127.0.0.1    bookstore
127.0.1.1    vasilis-P61-DS3-B3
# The following lines are desirable for IPv6 capable hosts
::1    ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

6. Εκτελούμε από τη γραμμή εντολών, την εντολή “**a2ensite bookstore**” προκειμένου να ενεργοποιήσουμε τον νέο ιστότοπο με όνομα **bookstore** και κάνουμε επανεκκίνηση του υπολογιστή.

#### **5.4 Ρύθμιση των παραμέτρων του OpenCart 3.0.3.1 για το Demo Βιβλιοπωλείο**


Μετά την επανεκκίνηση, πληκτρολογώντας στη γραμμή url ενός browser τη διεύθυνση **bookstore/** εμφανίζεται η αρχική σελίδα για την ρύθμιση των παραμέτρων του **opencart**.





Εικόνα 5.4: Αρχική σελίδα εγκατάστασης OpenCart

Πατώντας CONTINUE εμφανίζεται η δεύτερη σελίδα στην οποία γίνεται ο έλεγχος για το αν έχει ρυθμιστεί σωστά ο εξυπηρετητής για την εγκατάσταση του OpenCart και δίνονται πληροφορίες για τυχόν επιπλέον ρυθμίσεις που απαιτούνται.

2/4 Pre-Installation
Check your server is set-up correctly


1. Please configure your PHP settings to match requirements listed below.

PHP Settings	Current Settings	Required Settings	Status
PHP Version	7.2.15-Ubuntu0.18.04.1	5.4+	✔
Register Globals	Off	Off	✔
Magic Quotes GPC	Off	Off	✔
File Uploads	On	On	✔
Session Auto Start	Off	Off	✔

License

Pre-Installation

Configuration

LANGUAGE ▾

2. Please make sure the PHP extensions listed below are installed.

Extension Settings	Current Settings	Required Settings	Status
Database	On	On	✔
GD	On	On	✔
cURL	On	On	✔
OpenSSL	On	On	✔
ZLIB	On	On	✔
ZIP	On	On	✔

4. Please make sure you have set the correct permissions on the files list below.

Files	Status
/var/www/html/bookstore/config.php	Writable
/var/www/html/bookstore/admin/config.php	Writable

3. Please make sure you have set the correct permissions on the directories list below.

Directories	Status
/var/www/html/bookstore/image/	Writable
/var/www/html/bookstore/image/cache/	Writable
/var/www/html/bookstore/image/catalog/	Writable
/var/www/html/bookstore/system/storage/cache/	Writable
/var/www/html/bookstore/system/storage/logs/	Writable
/var/www/html/bookstore/system/storage/downloads/	Writable
/var/www/html/bookstore/system/storage/uploads/	Writable
/var/www/html/bookstore/system/storage/modification/	Writable

BACK
CONTINUE

[Project Homepage](#) | [Documentation](#) | [Support Forums](#)

Εικόνα 5.5: Δεύτερη σελίδα εγκατάστασης Opencart

Πατώντας πάλι CONTINUE εμφανίζεται η τρίτη σελίδα εγκατάστασης, όπου ζητείται να συμπληρωθούν τα πεδία με το \* που περιέχουν τα στοιχεία σύνδεσης στη Βάση Δεδομένων, το όνομα της Βάσης που θα εγκατασταθούν οι πίνακες, καθώς και τα στοιχεία σύνδεσης του διαχειριστή στη σελίδα admin του opencart.

3/4 Configuration  
Enter your database and administration details

opencart

1. Please enter your database connection details.

DB Driver: MySQL

\* Hostname: localhost

\* Username: root

Password: .....

\* Database: bookstore

\* Port: 3306

Prefix: OC\_

License

Pre-Installation

Configuration

LANGUAGE

2. Please enter a username and password for the administration.

\* Username: vapostolo

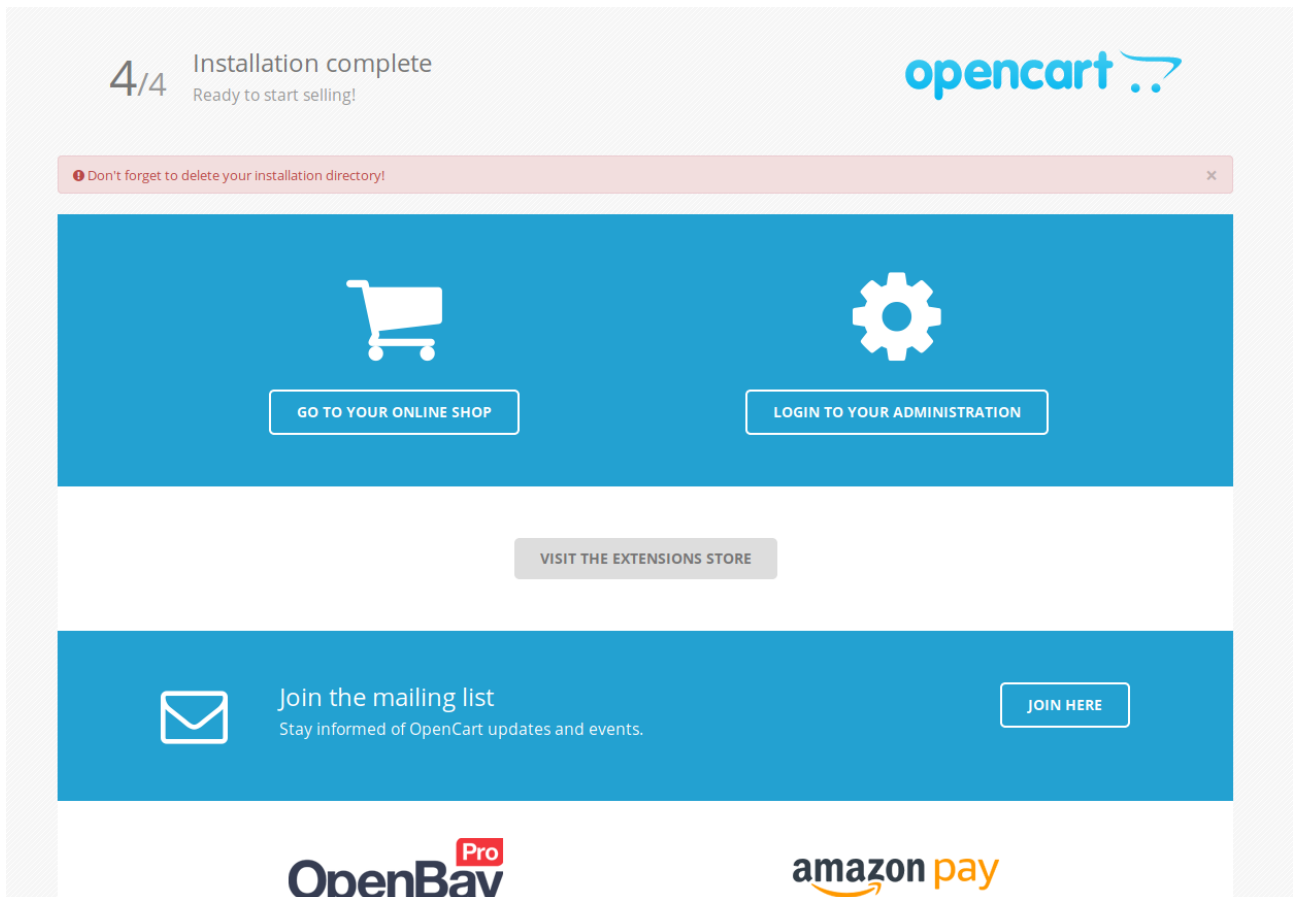
\* Password: mypassword

\* E-Mail: my@email.com

BACK CONTINUE

Εικόνα 5.6: Τρίτη σελίδα εγκατάστασης OpenCart

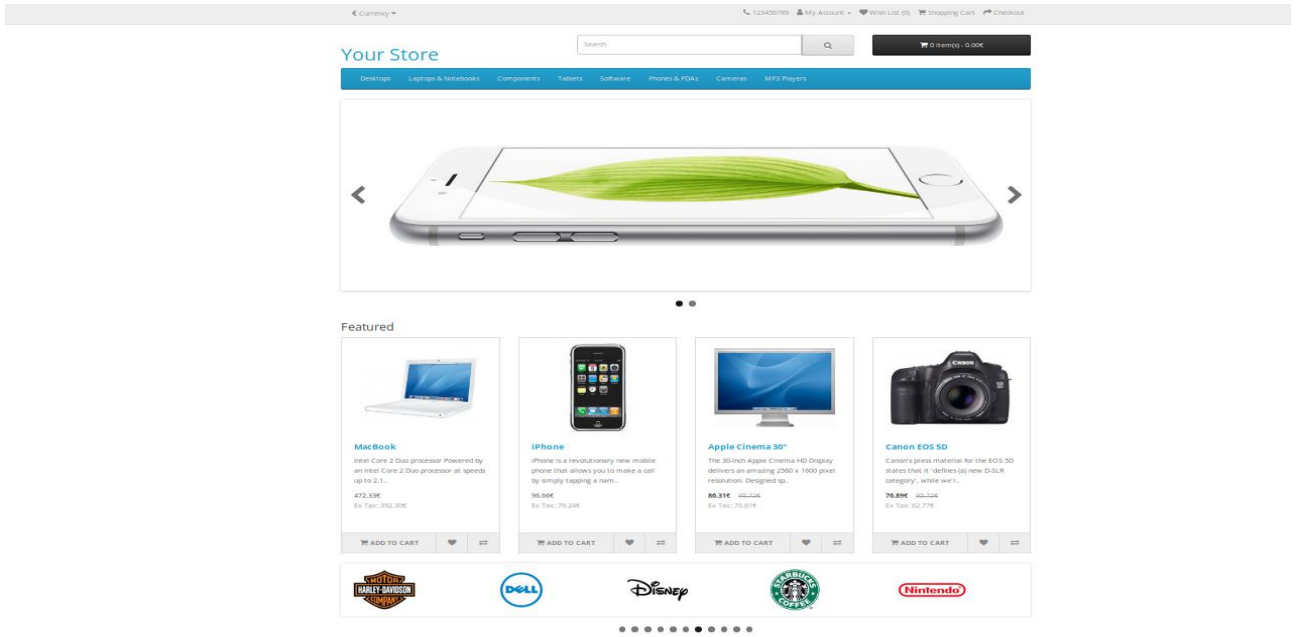
Αφού συμπληρωθούν τα στοιχεία και πατηθεί το CONTINUE, η τέταρτη και τελευταία σελίδα, δηλώνει την επιτυχή εγκατάσταση και μας ενημερώνει ότι, πρέπει πλέον για λόγους ασφαλείας, να διαγραφεί ο υποφάκελος installation που βρίσκεται στον φάκελλο bookstore του ιστοτόπου.



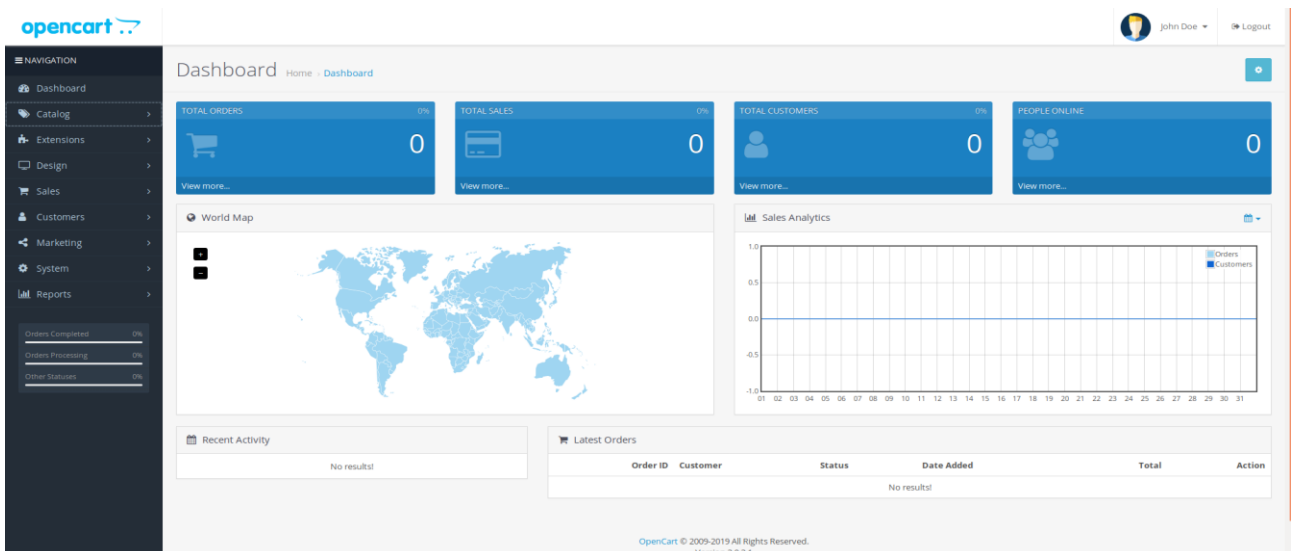
Εικόνα 5.7: Τέταρτη σελίδα εγκατάστασης OpenCart

Μπορούμε τώρα πλέον, είτε να επισκεφτούμε την αρχική σελίδα του ηλεκτρονικού καταστήματος πληκτρολογώντας το url bookstore/ (εικόνα 5.8), είτε, πληκτρολογώντας το url bookstore/admin και δίνοντας το username και το password του administration όπως τα δώσαμε στην τρίτη σελίδα εγκατάστασης του openCart, να επισκεφτούμε την σελίδα διαχείρισης (εικόνα 5.9).

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce



Εικόνα 5.8: Η αρχική σελίδα του ηλεκτρονικού καταστήματος “Demo Βιβλιοπωλείο”



Εικόνα 5.9: Η αρχική σελίδα διαχείρισης του ηλεκτρονικού καταστήματος “Demo Βιβλιοπωλείο”

## ΚΕΦΑΛΑΙΟ 6

### ΔΙΑΣΥΝΔΕΣΗ Microsoft SQL Server ΚΑΙ OpenCart

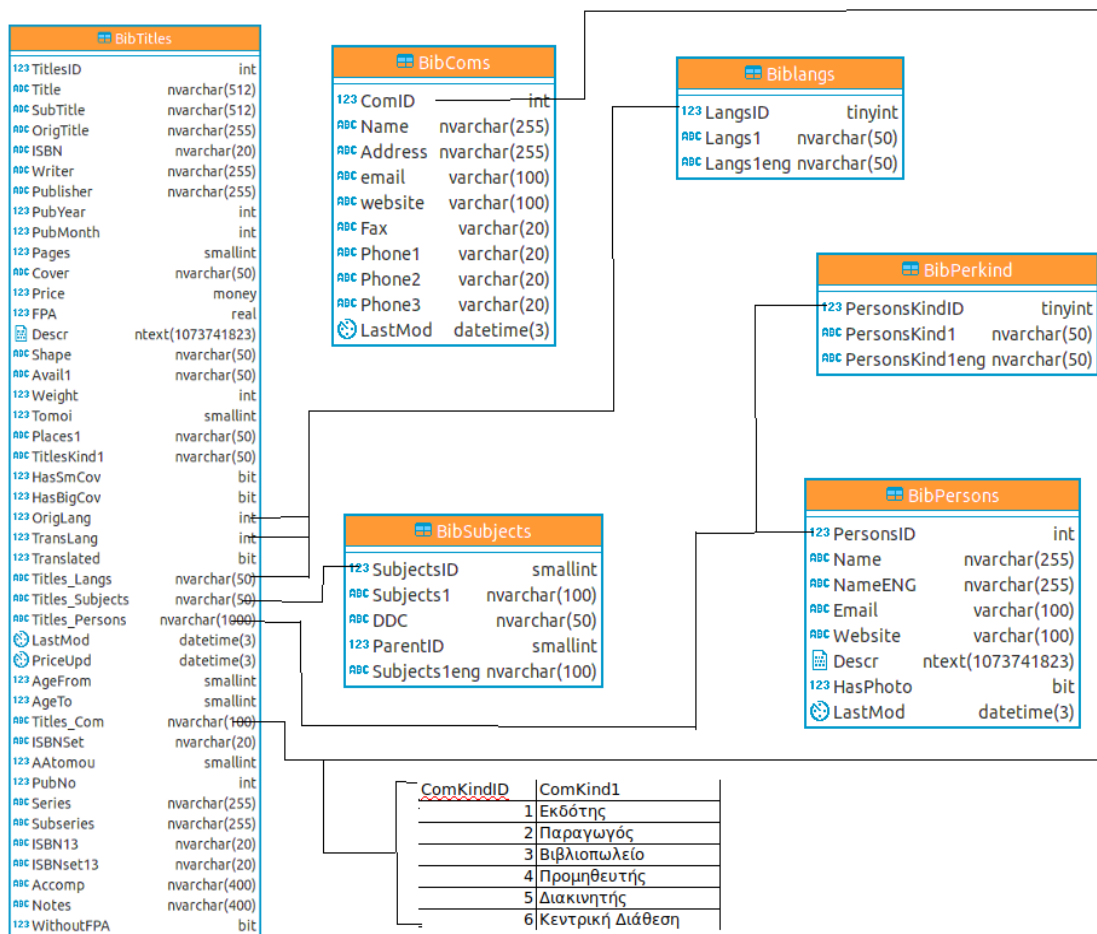
Στο κεφάλαιο αυτό, θα εξεταστεί η διασύνδεση μιας Βάσης Δεδομένων Microsoft SQL Server και της πλατφόρμας ηλεκτρονικού εμπορίου OpenCart χρησιμοποιώντας τη Java, ώστε να εντοπιστούν τα προβλήματα συμβατότητας και να εξεταστεί η δυνατότητα δημιουργίας προτύπων για την διασύνδεση αυτή.

#### 6.1 Η Βάση Δεδομένων sqlbookstore

Πρόκειται για τη Βάση Δεδομένων Microsoft SQL, η οποία βρίσκεται εγκατεστημένη στον αντίστοιχο Microsoft SQL Server και περιέχει τα “προς μεταφορά” δεδομένα. (Εικόνα 6.1)

Αποτελείται από τους πίνακες:

- **BibComs** – Περιέχει τους πωλητές (Εκδότης, Παραγωγός, Βιβλιοπωλείο, Προμηθευτής, Διακινητής, Κεντρική Διάθεση) με τα στοιχεία τους, όπως ID, Όνομα, Διεύθυνση, Τηλέφωνα, Email κ.λ.π. **Είναι ο πίνακας βάση του οποίου θα επιλεγούν οι “προς μεταφορά” εγγραφές από τον πίνακα των τίτλων BibTitles.**
- **BibPersons** - Περιέχει τους δημιουργούς με τα στοιχεία τους, όπως ID, Όνομα, Διεύθυνση, Τηλέφωνα, Email, Βιογραφικό κ.λ.π.
- **BibPerkind** – Περιέχει την ιδιότητα του δημιουργού π.χ. Συγγραφέας, Ζωγράφος, Συνθέτης, Γλύπτης κ.λ.π.
- **BibSubjects** – Περιέχει τις κατηγορίες και τις υποκατηγορίες που εντάσσεται ο κάθε τίτλος έργου π.χ. Μυθιστόρημα, Παιδικό, Επιστήμες, Γεωγραφία κ.λ.π.
- **Biblangs** – Περιέχει τις γλώσσες που προέρχεται ή μεταφράστηκε ο κάθε τίτλος.
- **BibTitles** – Ο κύριος πίνακας που περιέχει όλους τους τίτλους βιβλίων ή έργων που θα μεταφερθούν, με κριτήριο την επιλογή πωλητή από τον πίνακα BibComs, στην πλατφόρμα ηλεκτρονικού εμπορίου Opencart.



Εικόνα 6.1: Οι πίνακες της Βάσης Δεδομένων sqlbookstore και οι σχέσεις τους

## 6.2 Η Βάση Δεδομένων bookstore

Πρόκειται για τη Βάση Δεδομένων MySql, που αποτελεί τη ΒΔ του Demo Βιβλιοπωλείο (Opencart) και είναι αυτή που θα ενημερωθεί μέσω της Java με τα επιλεγμένα δεδομένα από την sqlbookstore.

Οι πίνακες που θα ενημερωθούν είναι:

- **manufacturer** – τα πεδία **manufacturer\_id** και **name**, αντίστοιχα από τα πεδία **ComID** και **Name**, του πίνακα **BibComs** της Βάσης Δεδομένων **sqlbookstore**. Το πεδίο **sort\_order** με default τιμή **0**.
- **manufacturer\_to\_store** – το πεδίο **manufacturer\_id** από το πεδίο **ComID** του πίνακα **BibComs** της Βάσης Δεδομένων **sqlbookstore**. Το πεδίο **store\_id** με default τιμή **0**.
- **product** – τα πεδία, **product\_id** και **model** ενημερώνονται από το πεδίο **TitlesID**, το **isbn** από το **ISBN**, το **location** από το **Places1**, το **manufacturer\_id**

από το ComID, τα **price** και **weight** από τα πεδία Price και Weight του πίνακα **BidTitles** αντίστοιχα της Βάσης Δεδομένων **sqlbookstore**. Τα υπόλοιπα πεδία του πίνακα **product** ενημερώνονται με τις default τιμές τους.

- **product\_description** – τα πεδία **product\_id**, **name**, **description**, **meta\_title** και **meta\_description**, από τα πεδία TitlesID, Title, Descr και SubTitle του πίνακα **BidTitles** της Βάσης Δεδομένων **sqlbookstore**.

- **product\_to\_store** - το πεδίο **product\_id** από το πεδίο **TitlesID** του πίνακα **BidTitles** της Βάσης Δεδομένων **sqlbookstore** και το πεδίο **store\_id** με default τιμή **0**.

- **product\_to\_layout** - το πεδίο **product\_id** από το πεδίο **TitlesID** του πίνακα **BidTitles** της Βάσης Δεδομένων **sqlbookstore** και τα πεδία **store\_id** και **layout\_id**, με default τιμή **0**.

- **product\_to\_category** - το πεδίο **product\_id** από το πεδίο **TitlesID** του πίνακα **BidTitles** της Βάσης Δεδομένων **sqlbookstore** και το πεδίο **category\_id** με τις τιμές:

1. τιμή “**59**” που αντιστοιχεί στην κατηγορία “**Είδη Βιβλίων**” αν πρόκειται για **Κύρια Κατηγορία**,
2. την τιμή “**60**” που αντιστοιχεί στην κατηγορία “**Συγγραφείς**” αν πρόκειται για **Δημιουργό**,
3. την τιμή από το πεδίο **category\_id** του πίνακα **category** της Βάσης Δεδομένων **bookstore** αν πρόκειται για **Υποκατηγορία**.

- **category** – τα πεδία **category\_id** και **parent\_id**, όπως προηγουμένως το πεδίο **category\_id** του πίνακα **product\_to\_category**.

- **category\_description** – το πεδίο **category\_id** από το αντίστοιχο πεδίο του πίνακα **category** και τα πεδία **name**, **description**, **meta\_title** και **meta\_description**, από το πεδίο Subjects1 του πίνακα **BidSubjects** της Βάσης Δεδομένων **sqlbookstore** αν πρόκειται για **Υποκατηγορία**, ή από το πεδίο Writers του πίνακα **BidTitles** της ίδιας Βάσης Δεδομένων αν πρόκειται για **Δημιουργό**.

- **category\_path** – τα πεδία **category\_id** και **path\_id** αντίστοιχα από τα πεδία **category\_id**, και **parent\_id** του πίνακα **category**.

- **category\_to\_layout** - το πεδίο **category\_id** από το πεδίο **category\_id** του πίνακα **category** και τα πεδία **store\_id** και **layout\_id**, με default τιμή **0**.



- **category\_to\_store** - το πεδίο **category\_id** από το πεδίο **category\_id** του πίνακα **category** και το πεδίο **store\_id** με default τιμή **0**.

### 6.3 Η εφαρμογή BookStore

Ενημερώνει τους πίνακες της ΒΔ bookstore που αναφέρθηκαν προηγουμένως και που αποτελεί την κύρια ΒΔ του OpenCart εγκατεστημένη σε ένα MySql server. Η ενημέρωση γίνεται με τη μεταφορά εγγραφών από την ΒΔ sqlbookstore, η οποία βρίσκεται εγκατεστημένη στον αντίστοιχο Microsoft SQL Server και περιλαμβάνει 180.000 τίτλους βιβλίων που εκδόθηκαν στη χώρα μας, καθώς επίσης και 100.000 δημιουργούς (συγγραφείς, μεταφραστές, επιμελητές, επιμελητές σειράς, εικονογράφους, φωτογράφους, ζωγράφους, κ.ά.). Το κριτήριο της μεταφοράς, είναι η επιλογή ενός ή περισσότερων εκδοτών, που βάση συμβολαίου έχουν το δικαίωμα πώλησης του αντίστοιχου έργου. Μετά την επιλογή, η ΒΔ bookstore του OpenCart ενημερώνεται με όλα τα έργα που αντιστοιχούν στο συγκεκριμένο εκδότη και είναι πλέον διαθέσιμα προς πώληση, άμεσα προσβάσιμα από το ευρύ κοινό, που αποτελεί τους εν δυνάμει πελάτες του ηλεκτρονικού καταστήματος.

#### 6.3.1 Οι βιβλιοθήκες της εφαρμογής

Οι βιβλιοθήκες που χρησιμοποιούνται (**Εικόνα 6.2**) για την εφαρμογή BookStore είναι οι:

- **MySQL JDBC Driver:** Ο επίσημος οδηγός JDBC για σύνδεση με MySQL. Χρησιμοποιείται από την εφαρμογή, για τη σύνδεση της Java με τη Βάση Δεδομένων bookstore.
- **Mssql-jdbc:** Ο οδηγός JDBC που παρέχει η Microsoft, για σύνδεση της Java με SQL Server και Azure SQL Database. Χρησιμοποιείται από την εφαρμογή, για σύνδεση της Java με τη Βάση Δεδομένων sqlbookstore.
- **JDK 1.8:** Το JDK είναι ένα περιβάλλον ανάπτυξης για την κατασκευή εφαρμογών, μικροεφαρμογών και στοιχείων, χρησιμοποιώντας τη γλώσσα προγραμματισμού Java. Περιλαμβάνει εργαλεία χρήσιμα για την ανάπτυξη και τη δοκιμή προγραμμάτων γραμμένα στη γλώσσα προγραμματισμού Java, που εκτελούνται στην πλατφόρμα Java.

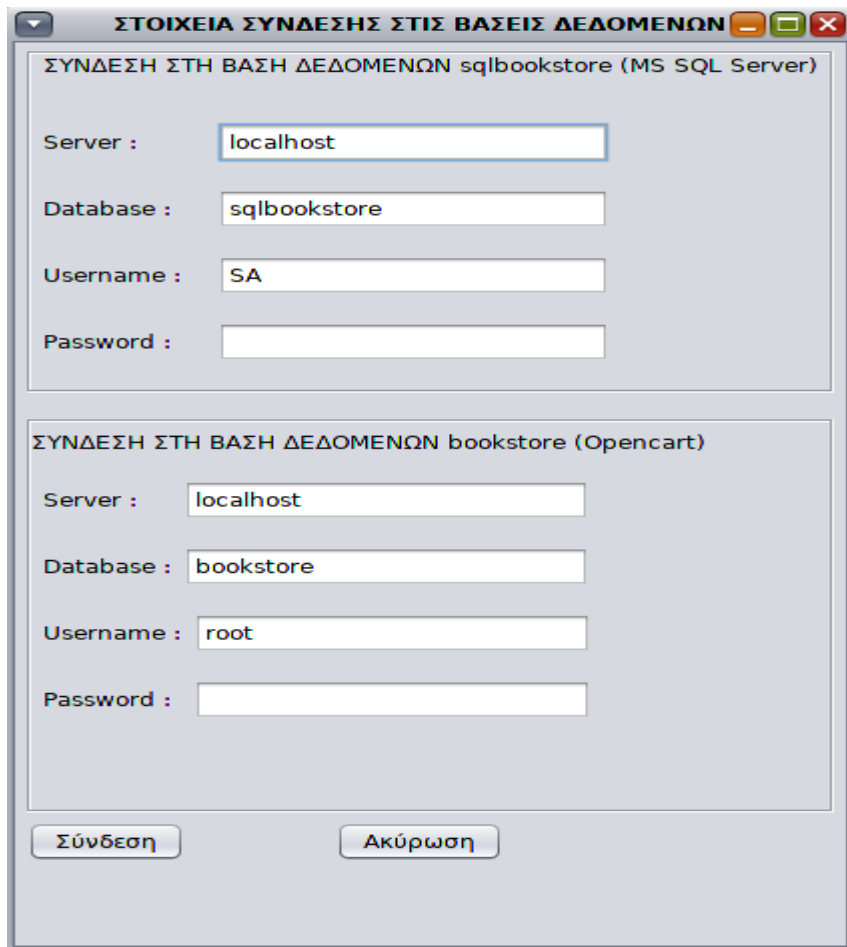


Εικόνα 6.2: Προγράμματα και Βιβλιοθήκες της εφαρμογής BookStore

### 6.3.2 Οι κλάσεις της εφαρμογής

Όπως φαίνεται στην **Εικόνα 6.2**, η εφαρμογή **BookStore** περιλαμβάνει τις εξής κλάσεις που βρίσκονται στο “πακέτο” **bookstore\_utilities**:

- **LoginFrm.java**: Πραγματοποιεί την είσοδο στην εφαρμογή και τη σύνδεση στις Βάσεις Δεδομένων sqlbookstore και bookstore (**Εικόνα 6.3**). Απαιτούνται τόσο για τον MS SQL Server που βρίσκεται η Βάση Δεδομένων sqlbookstore, όσο και για τον MySQL server που βρίσκεται η Βάση Δεδομένων bookstore: **α)** Το όνομα του server (localhost για τοπική σύνδεση ή το πλήρες URL αν πρόκειται για απομακρυσμένο εξυπηρετητή), **β)** το όνομα της αντίστοιχης Βάσης Δεδομένων, **γ)** το όνομα χρήστη και **δ)** ο κωδικός πρόσβασης.



ΣΤΟΙΧΕΙΑ ΣΥΝΔΕΣΗΣ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΣΥΝΔΕΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ sqlbookstore (MS SQL Server)

Server : localhost

Database : sqlbookstore

Username : SA

Password :

ΣΥΝΔΕΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ bookstore (Opencart)

Server : localhost

Database : bookstore

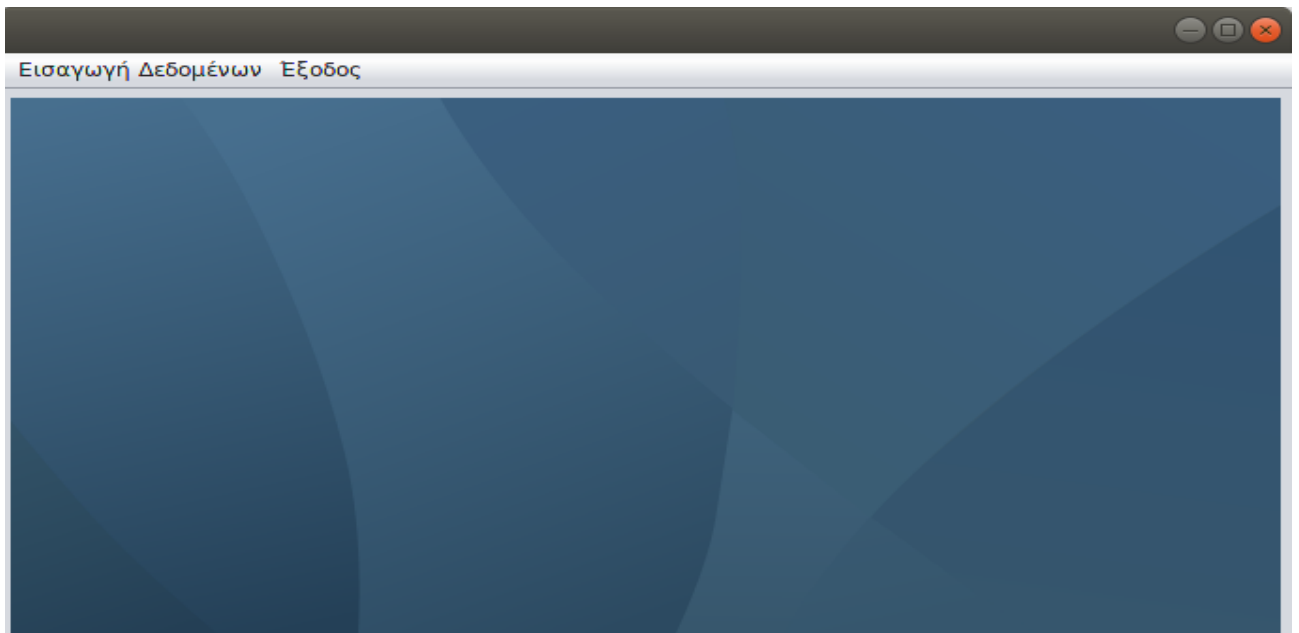
Username : root

Password :

Σύνδεση      Ακύρωση

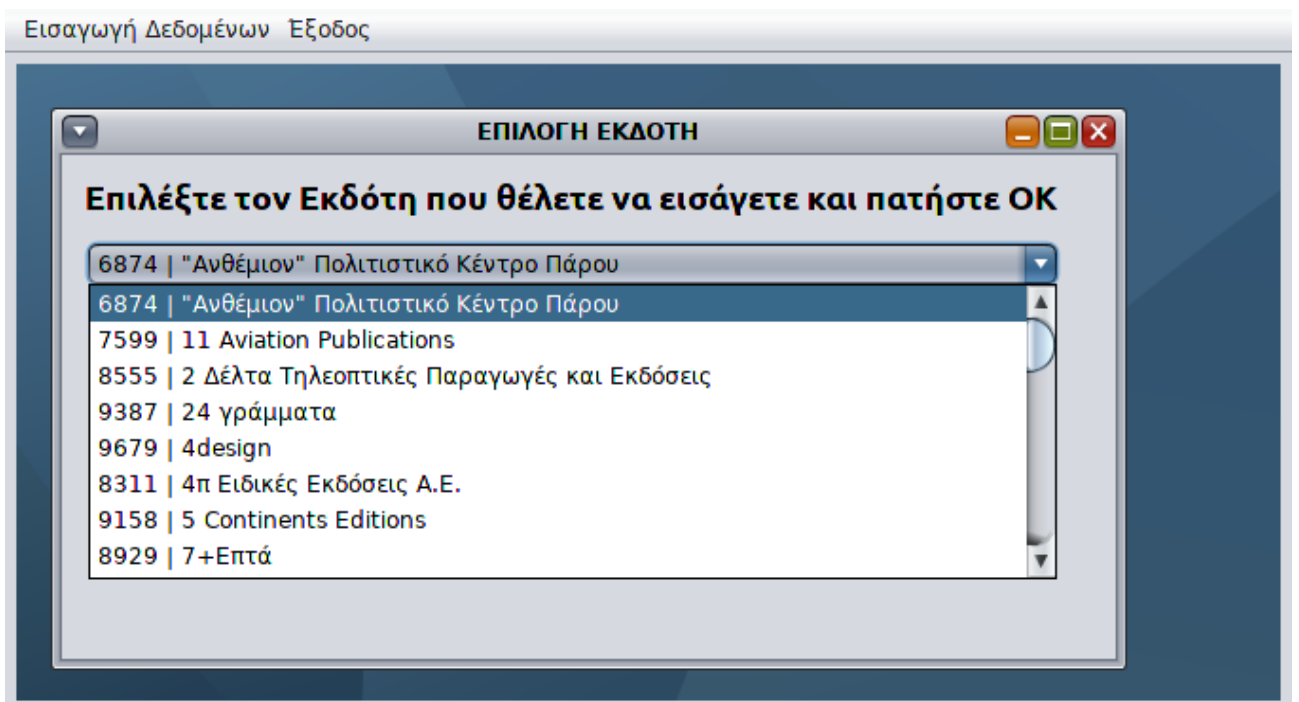
Εικόνα 6.3: LoginFrm.java – Στοιχεία Σύνδεσης στις Βάσεις Δεδομένων

- **MainFrm.java:** Είναι η “**Main Class**” της εφαρμογής, δηλαδή η κλάση που εκτελείται πρώτη και καλεί την LoginFrm.java. Μετά την επιτυχή είσοδο στην εφαρμογή εμφανίζει το μενου επιλογών, όπου ο χρήστης μπορεί να προχωρήσει κάνοντας την ανάλογη επιλογή από το μενου, είτε στην “**Εισαγωγή Δεδομένων**”, είτε στην “**Εξοδος**” από το πρόγραμμα. (Εικόνα 6.4)



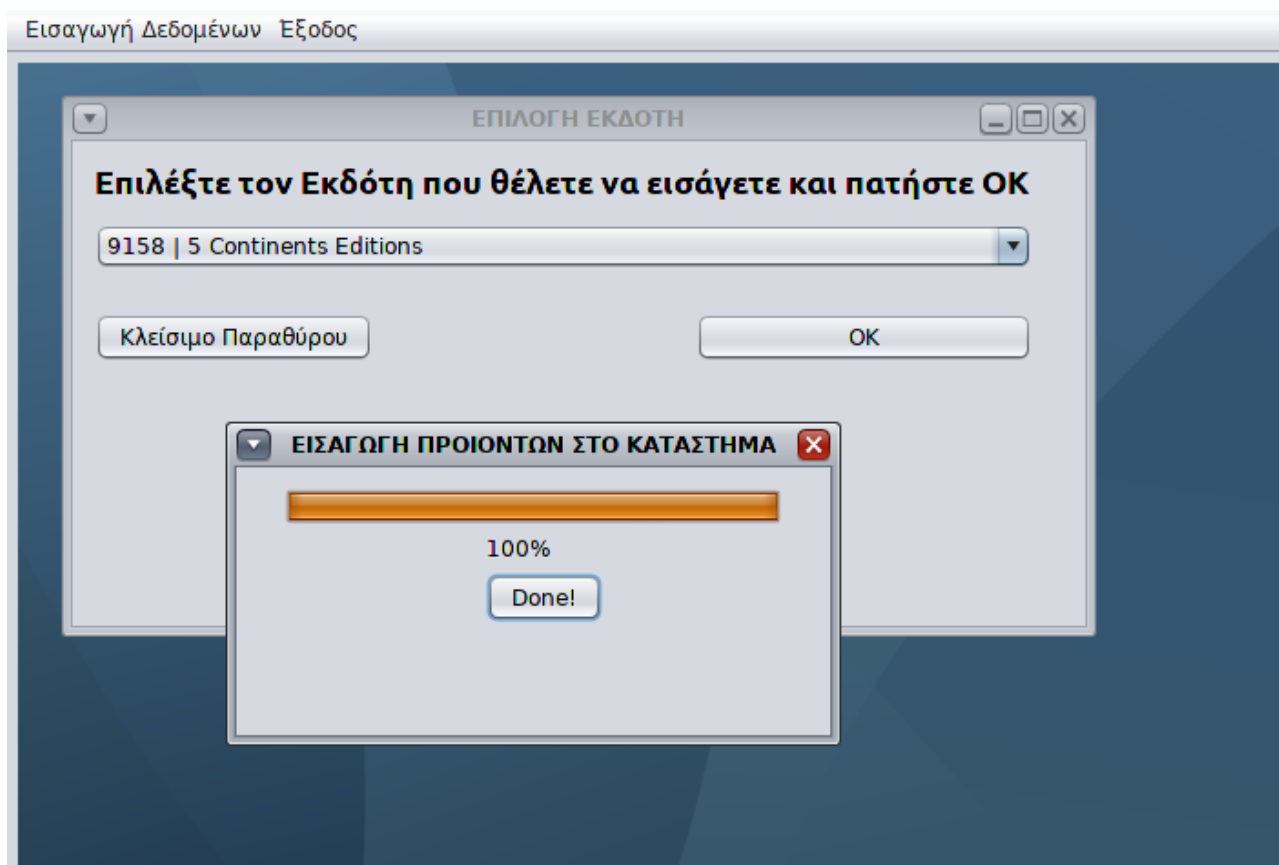
Εικόνα 6.4: MainFrm.java – Το menu της εφαρμογής

- **ComsFrm.java:** Χρησιμοποιεί μία “πτυσσόμενη λίστα” (Εικόνα 6.5), με όλες τις εγγραφές του πίνακα **BibComs** της Βάσης Δεδομένων **sqlbookstore**. Είναι η λίστα των **διαθέσιμων Εκδοτών της Βάσης**, που η επιλογή ενός θα αποτελέσει το κριτήριο για την μεταφορά των εγγραφών από τον πίνακα **BibTitles** της Βάσης Δεδομένων **sqlbookstore**, στους πίνακες της Βάσης Δεδομένων **bookstore**.



Εικόνα 6.5: ComsFrm.java – Επιλογή Εκδότη

- **InsertProductsFrm.java:** Μετά την επιλογή Εκδότη από την πτυσσόμενη λίστα, η κλάση InsertProductsFrm **αναλαμβάνει την μεταφορά των εγγραφών από τον πίνακα BibTitles της Βάσης Δεδομένων sqlbookstore, στους πίνακες της Βάσης Δεδομένων bookstore.** Εμφανίζεται η “μπάρα προόδου εργασιών”, καθώς και το ποσοστό των εγγραφών που έχουν μεταφερθεί τη συγκεκριμένη χρονική στιγμή (**Εικόνα 6.6**).



Εικόνα 6.6: InsertProductsFrm.java – Εισαγωγή προϊόντων στο κατάστημα

- **Actions.java:** Είναι η κλάση που περιέχει τις βοηθητικές στατικές μεθόδους που χρησιμοποιούνται από την εφαρμογή. Οι κύριες μέθοδοι είναι τρεις και καλούνται από την κλάση **InsertProductsFrm.java**:
  1. **Insert\_Writer:** Η μέθοδος αυτή εισάγει στην Βάση Δεδομένων bookstore, τους **Συγγραφείς** των έργων που αντιστοιχούν στον εκδότη ο οποίος αποτελεί το κριτήριο μεταφοράς από την “**Επιλογή Εκδότη**” και τους εμφανίζει στο αρχικό menu του Demo Βιβλιοπωλείου (OpenCart).
  2. **Insert\_Categories:** Η μέθοδος αυτή δημιουργεί στην Βάση Δεδομένων bookstore, την **Ιεραρχική Δομή Κατηγοριών-Υποκατηγοριών** των έργων

που αντιστοιχούν στον εκδότη της επιλογής. Η δομή αυτή των κατηγοριών, εμφανίζεται στο αρχικό menu του Demo Βιβλιοπωλείου (OpenCart).

3. **Insert\_Category:** Η μέθοδος αυτή εισάγει στην Βάση Δεδομένων bookstore, την **κατηγορία του έργου** που εισάγεται κάθε στιγμή, αφού πρώτα έχει ενημερωθεί η Ιεραρχική Δομή Κατηγοριών-Υποκατηγοριών και εντάσσει το έργο στη συγκεκριμένη υποκατηγορία.

#### 6.4 Το Demo Βιβλιοπωλείο

Μετά την “**Επιλογή Εκδότη**” από την εφαρμογή **BookStore** και την εισαγωγή εγγραφών στο **Demo Βιβλιοπωλείο**, το online κατάστημα θα έχει τη μορφή που εμφανίζεται στις Εικόνες 6.7 και 6.8.

Στην **Εικόνα 6.7** εμφανίζονται τα έργα ανά κατηγορία, ενώ στην **Εικόνα 6.8** εμφανίζονται οι συγγραφείς των έργων, βάση των οποίων ο πελάτης μπορεί να επιλέξει και να εμφανίσει τα έργα του συγγραφέα της επιλογής του.

Ακολουθεί η **Εικόνα 6.9** που εμφανίζονται αναλυτικά όλα τα στοιχεία του έργου, περιγραφή, τιμή, αξιολογήσεις κ.λ.π. Εδώ δίνεται η δυνατότητα στον πελάτη του ηλεκτρονικού καταστήματος, να ελέγξει τη διαθεσιμότητα, να γράψει τη δική του αξιολόγηση για το προϊόν, να επιλέξει την ποσότητα που θέλει και να το προσθέσει στο καλάθι αγορών.

€ Νόμισμα
123456789 | Ο λογαριασμός μου | Λίστα αγαπημένων (0) | Καλάθι Αγορών | Αγορά

## DEMO Βιβλιοπωλείο

🔍

🛒 0 προϊόν(τα) - 0,00€

Είδη Βιβλίων
Συγγραφείς

🏠
Είδη Βιβλίων
Θρησκεία
Θρησκεία - Γενικά έργα
Θρησκευτικά - Σπουδή και διδασκαλία
Θρησκευτικά - Σπουδή και διδασκαλία (Στοιχειώδης)

**Είδη Βιβλίων (117)**

- Γενικά βιβλία (21)
- Γλώσσα (22)
- Εφαρμοσμένες επιστήμες (1)
- Θρησκεία (4)
- Ιστορία, Γεωγραφία (18)
- Καλές τέχνες (6)
- Κοινωνικές επιστήμες (47)
- Λογοτεχνία (18)
- Φιλοσοφία, Παραφυσικά φαινόμενα, Ψυχολογία (2)
- Φυσικές και θετικές επιστήμες (8)
- Συγγραφείς (117)

### Θρησκευτικά - Σπουδή και διδασκαλία (Στοιχειώδης)

🛒 Σύγκριση Προϊόντων (0)
Ταξινόμηση: Προεπιλογή
Εμφάνιση: 15

Ιστορία, θρησκευτικά, μελέτη περιβάλλοντος Γ' δημοτικού

Ιστορία, θρησκευτικά, μελέτη περιβάλλοντος Γ' δημοτικού

Ένα χρηστικό εκπαιδευτικό βοήθημα που καλύπτει πλήρως την ύλη της "Ιστορίας", των "Θρησκευτικών" και..

17,90€

Χωρίς ΦΠΑ: 17,90€

🛒 ΚΑΛΑΘΙ
❤️
⚖️

Ιστορία, θρησκευτικά, μελέτη περιβάλλοντος Δ' δημοτικού

Ιστορία, θρησκευτικά, μελέτη περιβάλλοντος Δ' δημοτικού

Ένα χρηστικό εκπαιδευτικό βοήθημα που καλύπτει πλήρως την ύλη της "Ιστορίας", των "Θρησκευτικών" και..

17,90€

Χωρίς ΦΠΑ: 17,90€

🛒 ΚΑΛΑΘΙ
❤️
⚖️

Φυσικά, θρησκευτικά Ε' δημοτικού

Φυσικά, θρησκευτικά Ε' δημοτικού

Ένα χρηστικό βοήθημα που καλύπτει πλήρως την ύλη των "Φυσικών" και των "Θρησκευτικών" των αντίστοιχων..

17,90€

Χωρίς ΦΠΑ: 17,90€

🛒 ΚΑΛΑΘΙ
❤️
⚖️

Εμφάνιση 1 έως 4 από 4 (1 Σελ.)

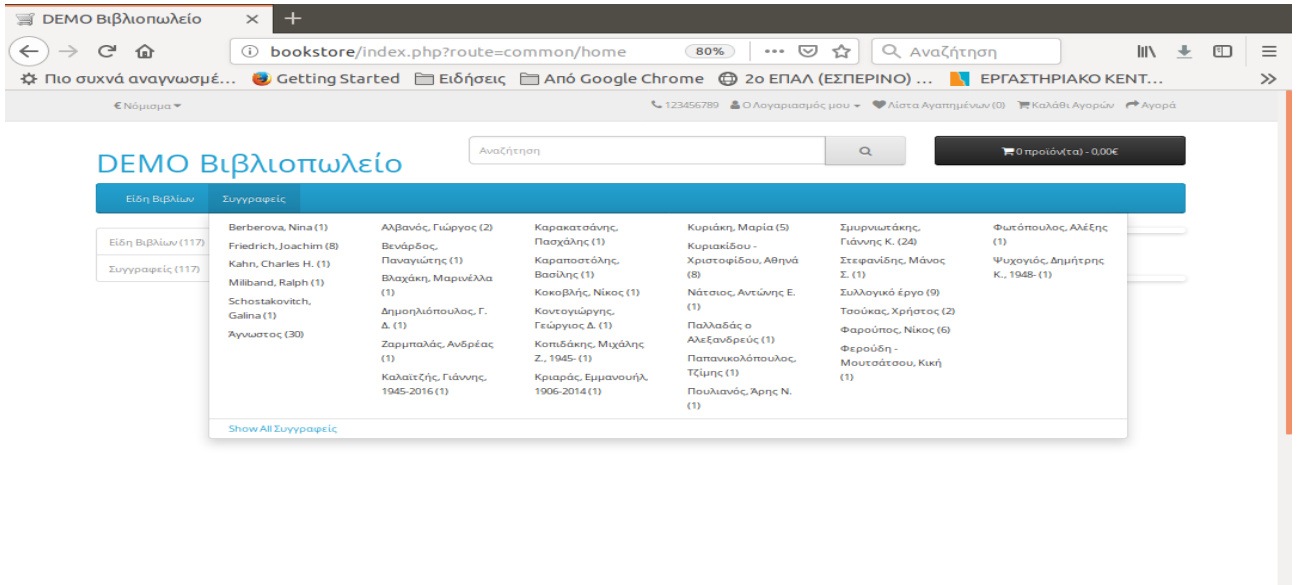
<p><b>Πληροφορίες</b></p> <ul style="list-style-type: none"> <li>About Us</li> <li>Delivery Information</li> <li>Privacy Policy</li> </ul>	<p><b>Εξυπηρέτηση Πελατών</b></p> <ul style="list-style-type: none"> <li>Επικοινωνήστε μαζί μας</li> <li>Επιστροφές</li> <li>Χάρτης Ιστότοπου</li> </ul>	<p><b>Περισσότερα</b></p> <ul style="list-style-type: none"> <li>Ευρετήριο Κατασκευαστών</li> <li>Αγορά Δωροεπιταγής</li> <li>Πρόγραμμα Συνεργατών</li> </ul>	<p><b>Ο Λογαριασμός μου</b></p> <ul style="list-style-type: none"> <li>Ο λογαριασμός μου</li> <li>Ιστορικό Παραγγελιών</li> <li>Λίστα Αγαπημένων</li> </ul>
--	--	---	---

Εικόνα 6.7 Έργα ανά κατηγορία

Αποστολόπουλος Βασίλειος

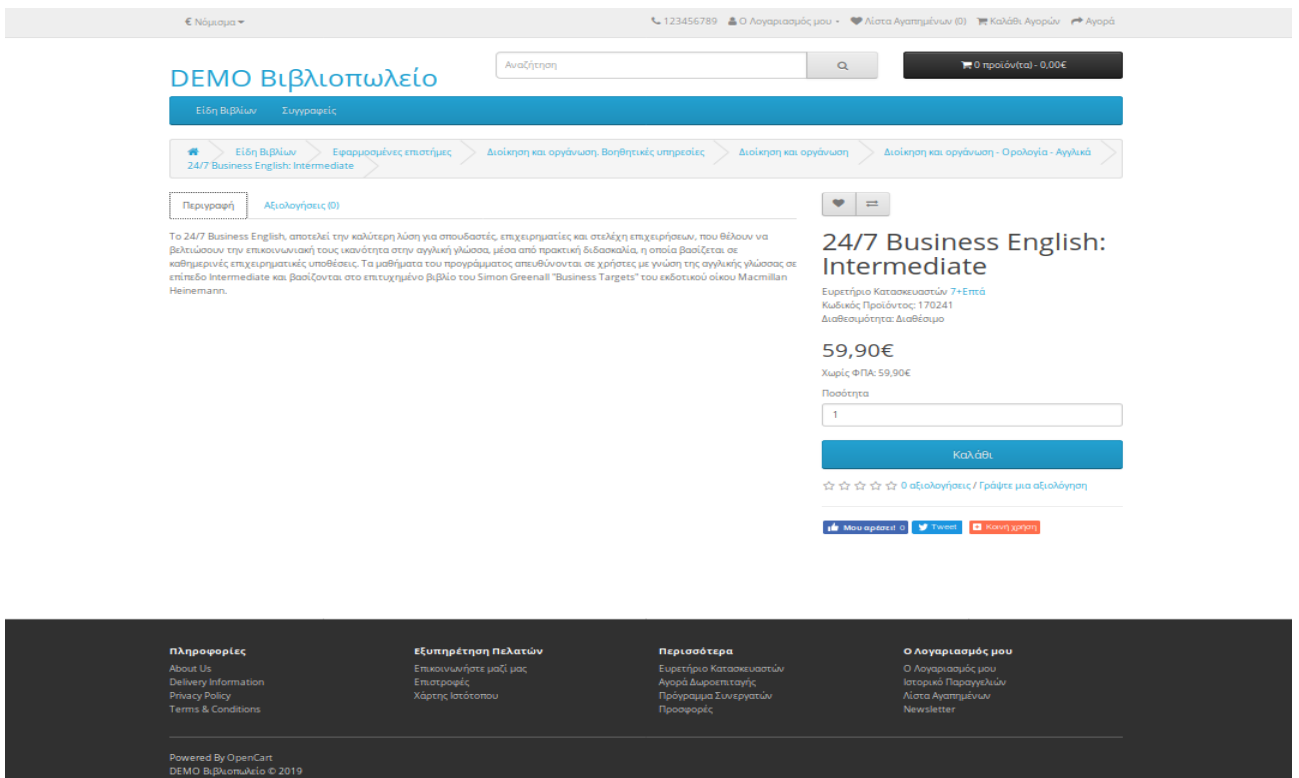
70

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce



The screenshot shows the 'DEMO Βιβλιοπωλείο' website. The search bar contains 'Αναζήτηση'. Below the search bar, there is a table of authors and their book counts. The table has two columns: 'Είδη Βιβλίων' and 'Συγγραφείς'. The authors listed include Berberova, Nina (1), Friedrich, Joachim (8), Kahn, Charles H. (1), Miliband, Ralph (1), Schostakovitch, Galina (1), Άγνωστος (30), Αλβανός, Γιώργος (2), Βενάφδης, Παναγιώτης (1), Βλαχάκη, Μαριέλλα (1), Δημητριάδης, Γ. Δ. (1), Ζαμπταλάς, Ανδρέας (1), Καλαϊτζής, Γιάννης, 1945-2016 (1), Καρακατσάνης, Πασχάλης (1), Καραματσόλης, Βασίλης (1), Κοκοβλής, Νίκος (1), Κοντογιώργης, Γεώργιος Δ. (1), Κοπιδάκης, Μιχάλης Ζ., 1945- (1), Κριαράς, Εμμανουήλ, 1906-2014 (1), Κυριακή, Μαρία (5), Κυριακίδου - Χριστοφίδου, Αθηνά (8), Νάτσος, Αντώνης Ε. (1), Παλλάδας ο Αλέξανδρeύς (1), Παπακοκόπουλος, Τζιμής (1), Πουλιανός, Άρης Ν. (1), Σιμνιωτάκης, Γιάννης Κ. (24), Στεφανίδης, Μάνος Σ. (1), Συλλογικό έργο (9), Τσούκας, Χρήστος (2), Φαρούκος, Νίκος (6), Φερούδη - Μουτσάτσου, Κική (1), Φωτόπουλος, Αλέξης (1), Ψυχγιός, Δημήτρης Κ., 1948- (1). There is a 'Show All Συγγραφείς' link at the bottom of the table.

Εικόνα 6.8 Επιλογή Συγγραφέα



The screenshot shows the product page for '24/7 Business English: Intermediate'. The page includes a breadcrumb trail: 'Είδη Βιβλίων > Εφαρμοσμένες επιστήμες > Διοίκηση και οργάνωση, Βοηθητικές υπηρεσίες > Διοίκηση και οργάνωση > Διοίκηση και οργάνωση - Ορολογία - Αγγλικά > 24/7 Business English: Intermediate'. The product description states: 'Το 24/7 Business English, αποτελεί την καλύτερη λύση για σπουδαστές, επιχειρηματίες και στελέχη επιχειρήσεων, που θέλουν να βελτιώσουν την επικοινωνιακή τους ικανότητα στην αγγλική γλώσσα, μέσα από πρακτική διδασκαλία, η οποία βασίζεται σε καθημερινές επιχειρηματικές υποθέσεις. Τα μαθήματα του προγράμματος απευθύνονται σε χρήστες με γνώση της αγγλικής γλώσσας σε επίπεδο Intermediate και βασίζονται στο επιτυχημένο βιβλίο του Simon Greenall "Business Targets" του εκδοτικού οίκου Macmillan Heinemann.' The price is 59,90€ (χωρίς ΦΠΑ: 59,90€). The quantity is set to 1. There is a 'Καλάθι' button and a star rating system. The footer contains links for 'Πληροφορίες', 'Εξυπηρέτηση Πελατών', 'Περισσότερα', and 'Ο λογαριασμός μου'.

Εικόνα 6.9 Αναλυτικά στοιχεία του έργου



## ΚΕΦΑΛΑΙΟ 7

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

Στο κεφάλαιο αυτό γίνεται μια σύνοψη της πτυχιακής εργασίας, αναφέρονται τα συμπεράσματα που προέκυψαν κατά την εκπόνησή της, καθώς και οι προοπτικές οι οποίες μπορούν να υπάρξουν, προκειμένου ένα παρόμοιο σύστημα να έχει εφαρμογή, αν όχι σε όλες, τουλάχιστον στις δημοφιλέστερες πλατφόρμες ηλεκτρονικού εμπορίου.

#### 7.1 Σύνοψη της πτυχιακής εργασίας

Στην παρούσα πτυχιακή εργασία, έγινε μια σύντομη παρουσίαση της έννοιας του **ηλεκτρονικού εμπορίου**, καθώς και μια αναφορά στην θέση που κατέχει σήμερα τόσο στην ελληνική, όσο και στην παγκόσμια αγορά. Έγινε εκτενής αναφορά στις κατηγορίες του, καθώς και στην ανάλυση των χαρακτηριστικών που πρέπει να έχει μια πλατφόρμα ηλεκτρονικού εμπορίου.

Αφού έγινε μια σύντομη περιγραφή του τι είναι ένα **Σύστημα Διαχείρισης Περιεχομένου CMS (Content Management System)**, παρουσιάστηκαν οι δημοφιλέστερες πλατφόρμες ηλεκτρονικού εμπορίου, δίνοντας ιδιαίτερη έμφαση στη δομή των πινάκων τους στη Βάση Δεδομένων. Επιλέχτηκε μία από αυτές, η **OpenCart**, παρουσιάστηκε η εγκατάστασή της και δημιουργήθηκε μια εφαρμογή γραμμένη σε **Java**, η οποία ενημέρωνε την Βάση Δεδομένων της πλατφόρμας με τα προϊόντα που προέρχονταν από μία άλλη εξωτερική Βάση Δεδομένων.

Περιγράφηκαν επίσης οι τεχνολογίες **Ελεύθερου Λογισμικού / Λογισμικού Ανοικτού Κώδικα (ΕΛ/ΛΑΚ)** που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας. Έγινε σύντομη αναφορά στους όρους **Βάση Δεδομένων** και **Εξυπηρετητής Διαδικτύου (Web Server)** και παρουσιάστηκαν αναλυτικά τα Συστήματα Διαχείρισης Βάσεων Δεδομένων, **Microsoft SQL Server** και **MySQL** που αποτέλεσαν τις πηγές λήψης και αποθήκευσης των δεδομένων.

Παρουσιάστηκαν τα κύρια χαρακτηριστικά και οι δυνατότητες της γλώσσας προγραμματισμού **Java** που ήταν το κύριο εργαλείο συγγραφής κώδικα κατά την εκπόνηση της πτυχιακής.

## 7.2 Συμπεράσματα - προοπτικές

Παρατηρώντας τη δομή των πινάκων στις Βάσεις Δεδομένων που αποθηκεύονται τα ΠΡΟΪΟΝΤΑ, οι ΚΑΤΗΓΟΡΙΕΣ και οι ΚΑΤΑΣΚΕΥΑΣΤΕΣ, στις δημοφιλέστερες πλατφόρμες ηλεκτρονικού εμπορίου (ΚΕΦΑΛΑΙΟ 4), διαπιστώνεται η διαφορετικότητα στο σχεδιασμό τους. Αυτό έχει σαν αποτέλεσμα την “ασυμβατότητα” μεταξύ τους, με συνέπεια τη δυσκολία της μεταφοράς των δεδομένων, όταν κριθεί για κάποιους λόγους σκόπιμο να γίνει “μεταπήδηση” ενός καταστήματος από τη μια πλατφόρμα στην άλλη.

Η ίδια δυσκολία παρατηρείται όταν χρειαστεί να μεταφερθούν και εγγραφές που αφορούν τις παραπάνω οντότητες, από μία ιδιωτική (custom) Βάση Δεδομένων ενός καταστήματος, σε μια πλατφόρμα ηλεκτρονικού εμπορίου, με αποτέλεσμα η δυνατότητα αυτή καθ’ αυτής της μεταφοράς να αποτελεί ένα επιπλέον κριτήριο για την επιλογή μιας πλατφόρμας.

Η “προτυποποίηση” τέτοιων πινάκων στις Βάσεις Δεδομένων των πλατφορμών ηλεκτρονικού εμπορίου, ιδιαίτερα όσον αφορά τα βασικά στοιχεία των ΠΡΟΪΟΝΤΩΝ, ΚΑΤΗΓΟΡΙΩΝ και ΚΑΤΑΣΚΕΥΑΣΤΩΝ ίσως είναι η λύση στο παραπάνω πρόβλημα. Μια τέτοια “προτυποποίηση” μπορεί να αποτελέσει ακόμα και οδηγό, για το σχεδιασμό ιδιωτικών (custom) βάσεων δεδομένων, κάνοντας με αυτόν τον τρόπο σχετικά εύκολη την μεταφορά προϊόντων και υπηρεσιών σε μια πλατφόρμα ηλεκτρονικού εμπορίου με σκοπό την προσέλκυση από ένα κατάστημα “διαδικτυακών” πελατών.

## ΠΑΡΑΡΤΗΜΑ

### Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

#### LoginFrm.java

```

package bookstore_utilities;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Collections;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author vasilis
 */
public class Actions {
    java.sql.Timestamp mydate = new java.sql.Timestamp(new java.util.Date().getTime());

    //-----Update All Categories tree sql1.biblionet.gr-----

    /* public static void updateCategories(Connection bookstore_conn, Connection sqlbookstore_conn)
    throws SQLException {

        Integer product_id;
        String Titles_Subjects;
        String[] categoriesID;

        int metritis = 0;

        try (Statement bookstore_st = bookstore_conn.createStatement()) {
            ResultSet product_rs = bookstore_st.executeQuery("SELECT * FROM product");

            while (product_rs.next()){
                product_id = product_rs.getInt("product_id");
                metritis++;
                System.out.println(product_id + " " + Integer.toString(metritis));

                try (Statement sqlbookstore_st = sqlbookstore_conn.createStatement()) {
                    ResultSet sqlbookstore_rs = sqlbookstore_st.executeQuery("SELECT Titles_Subjects FROM
                    dbo.BibTitles WHERE TitlesID = " +
                        product_id);

                    if (sqlbookstore_rs.next()){
                        Titles_Subjects = sqlbookstore_rs.getString("Titles_Subjects");
                        categoriesID = Split(Titles_Subjects);
                    }
                }
            }
        }
    }

```

```

        Update_Category_Table(sqlbookstore_conn, bookstore_conn, categoriesID);
        Product_to_Categories(bookstore_conn, product_id, categoriesID);
    }
}
}

}

private static void Update_Category_Table(Connection sqlbookstore_conn, Connection bookstore_conn,
String[] categories_id) throws SQLException{
    String selStr;

    for (int i=0; i<categories_id.length; i++){
        Integer categoryID = Integer.parseInt(categories_id[i]);
        selStr = "SELECT * FROM category WHERE category_id = " + categoryID;
        try (Statement bookstore_st = bookstore_conn.createStatement()) {
            ResultSet category_rs = bookstore_st.executeQuery(selStr);
            if (!category_rs.next()){
                try (Statement sqlbookstore_st = sqlbookstore_conn.createStatement()) {
                    selStr = "SELECT SubjectsID, Subjects1, ParentID FROM dbo.BibSubjects WHERE SubjectsID = "
+ categoryID;
                    ResultSet BibSubjects_rs = sqlbookstore_st.executeQuery(selStr);
                    while (BibSubjects_rs.next()){
                        Insert_Category_Table(bookstore_conn, BibSubjects_rs.getInt("SubjectsID"),
                            BibSubjects_rs.getString("Subjects1"), BibSubjects_rs.getInt("ParentID"));
                    }

                    while (BibSubjects_rs.getInt("ParentID") != 0){
                        selStr = "SELECT SubjectsID, Subjects1, ParentID FROM dbo.BibSubjects WHERE SubjectsID = "
+ BibSubjects_rs.getInt("ParentID");
                        BibSubjects_rs = sqlbookstore_st.executeQuery(selStr);
                        while (BibSubjects_rs.next()){
                            selStr = "SELECT * FROM category WHERE category_id = " + categoryID;
                            category_rs = bookstore_st.executeQuery(selStr);
                            if (!category_rs.next()){
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

private static void Insert_Category_Table(Connection bookstore_conn, Integer SubjectsID, String
Subjects1, Integer ParentID) throws SQLException{
    PreparedStatement ps;

    ps = bookstore_conn.prepareStatement("INSERT INTO category (category_id, parent_id, top, status,
date_added, date_modified) VALUES(?,?,0,1,?,?)");
    ps.setInt(1, SubjectsID);
    ps.setInt(2, ParentID);
}

```

```

ps.setTimestamp(3, new java.sql.Timestamp(new java.util.Date().getTime()));
ps.setTimestamp(4, new java.sql.Timestamp(new java.util.Date().getTime()));
ps.executeUpdate();

ps = bookstore_conn.prepareStatement("INSERT INTO category_description (category_id, language_id,
name, meta_title) VALUES(?,?,?,?)");
ps.setInt(1, SubjectsID);
ps.setInt(2, 2);
ps.setNString(3, Subjects1);
ps.setNString(4, Subjects1);
ps.executeUpdate();

ps = bookstore_conn.prepareStatement("INSERT INTO category_to_store (category_id, store_id)
VALUES(?,?)");
ps.setInt(1, SubjectsID);
ps.setInt(2, 0);
ps.executeUpdate();

ps = bookstore_conn.prepareStatement("INSERT INTO category_to_layout (category_id, store_id,
layout_id) VALUES(?,?,?)");
ps.setInt(1, SubjectsID);
ps.setInt(2, 0);
ps.setInt(3, 0);
ps.executeUpdate();

}

private static void Product_to_Categories(Connection conn, Integer product_id, String[] categories_id)
throws SQLException{
    String selStr;

    for (int i=0; i<categories_id.length; i++){
        Integer categoryID = Integer.parseInt(categories_id[i]);

        while(categoryID != 0){
            selStr = "SELECT * FROM category WHERE category_id = " + categoryID;
            Statement st2;
            try (Statement st1 = conn.createStatement()) {
                ResultSet categories_rs = st1.executeQuery(selStr);
                categories_rs.first();
                Integer parentID = categories_rs.getInt("parent_id");
                st2 = conn.createStatement();
                ResultSet product_to_category_rs = st2.executeQuery("SELECT * FROM product_to_category
WHERE category_id = "
                    + categoryID + " AND product_id = " + product_id);
                if (product_to_category_rs.first() == false) {
                    PreparedStatement ps = conn.prepareStatement("INSERT INTO product_to_category
(product_id, category_id) VALUES(?,?)");
                    ps.setInt(1, product_id);
                    ps.setInt(2, categoryID);
                    ps.executeUpdate();
                } categoryID = parentID;
            }
            st2.close();
        }
    }
}

```

```

    }

}

*/
//-----Update Products from sql1.biblionet.gr-----
/*
public static void old_updateProducts(Connection bookstore_conn, Connection sqlbookstore_conn, int
ComID, String SearchStr) throws SQLException, UnsupportedEncodingException, InterruptedException {
    String SourceBigimageURL = "http://www.biblionet.gr/images/covers/b";
    String SourceSmallimageURL = "http://www.biblionet.gr/images/covers/s";
    String TargetimageURL = "/home/vasilis/www/html/eshop/image/catalog/kalokathis/";
    String DefaultimageURL = "/home/vasilis/www/html/eshop/image/catalog/default-image.png";
    java.sql.Date date = new java.sql.Date(Calendar.getInstance().getTime().getTime());

    Statement sqlbookstore_st = sqlbookstore_conn.createStatement();
    ResultSet rs = sqlbookstore_st.executeQuery("SELECT COUNT(*) AS nunrows FROM dbo.BibTitles
WHERE Publisher LIKE (N'" + SearchStr + "'");
    rs.next();
    int counter = rs.getInt("nunrows")/100;
    rs.close();

    try (ResultSet bibtitles_rs = sqlbookstore_st.executeQuery("SELECT * FROM dbo.BibTitles WHERE
Publisher LIKE (N'" + SearchStr + "'")) {
        int i = 0;

        while (bibtitles_rs.next()){
            i++;
            int id = bibtitles_rs.getInt("TitlesID");
            //String[] categoriesID;
            try (Statement bookstore_st = bookstore_conn.createStatement()) {
                ResultSet product_rs = bookstore_st.executeQuery("SELECT * FROM product WHERE product_id =
" + id);

                if (product_rs.first() == false) {

                    PreparedStatement ps = bookstore_conn.prepareStatement("INSERT INTO product (product_id,
model, sku, upc, ean, jan, isbn, location, "
                    + "quantity, stock_status_id, image, manufacturer_id, shipping, price, points,
tax_class_id, date_available, "
                    + "weight, weight_class_id, length, width, height, length_class_id, subtract, minimum,
sort_order, status, "
                    + "viewed, date_added, date_modified, mpn)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");

                    ps.setInt(1, id);
                    ps.setString(2, Integer.toString(id));
                    ps.setString(3, "");
                    ps.setString(4, "");
                    ps.setString(5, "");
                    ps.setString(6, "");

                    if (bibtitles_rs.getString("ISBN") != null) {
                        ps.setNString(7, bibtitles_rs.getString("ISBN").replaceAll("-", ""));
                    } else {
                        ps.setString(7, "");
                    }
                }
            }
        }
    }
}

```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
ps.setNString(8, bibtitles_rs.getString("Places1"));
ps.setInt(9, 1);
ps.setInt(10, 6);
ps.setString(11, DefaultimageUrl);

if (bibtitles_rs.getBoolean("HasSmCov")){
    try {
downloadUsingStream(SourceSmallimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID"))
                    , TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));
        ps.setString(11, TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));
    } catch (IOException ex) {
        ps.setString(11, DefaultimageUrl);
    }
}

if (bibtitles_rs.getBoolean("HasBigCov")){
    try {
        downloadUsingStream(SourceBigimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID"))
                            , TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));
        ps.setString(11, TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));
    } catch (IOException ex) {
        ps.setString(11, DefaultimageUrl);
    }
}

ps.setInt(12, ComID);
ps.setInt(13, 1);
ps.setFloat(14, bibtitles_rs.getFloat("Price"));
ps.setInt(15, 0);
ps.setInt(16, 11);
ps.setDate(17, date);
ps.setDouble(18, bibtitles_rs.getDouble("Weight"));
ps.setInt(19, 2);
ps.setDouble(20, 0);
ps.setDouble(21, 0);
ps.setDouble(22, 0);
ps.setInt(23, 1);
ps.setInt(24, 0);
ps.setInt(25, 1);
ps.setInt(26, 0);
ps.setInt(27, 1);
ps.setInt(28, 0);
ps.setDate(29, date);
ps.setDate(30, date);
ps.setString(31, "");

ps.executeUpdate();
ps.close();
}

Statement bookstore_st1 = bookstore_conn.createStatement();
String temp = "SELECT * FROM product_description WHERE product_id = " + id + " AND
language_id = 2";
ResultSet product_description_rs = bookstore_st1.executeQuery(temp);
PreparedStatement ps1;
```

```

if (product_description_rs.first() == false) {
    String temp1;
    if (bibtitles_rs.getString("Descr") == null){
        temp1="";
    } else {
        temp1 = bibtitles_rs.getString("Descr");
    }
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_description (product_id,
language_id, name, description, "
        + "meta_title, meta_description, tag, meta_keyword) VALUES(?,?,?,?,?,?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 2);
    ps1.setNString(3, bibtitles_rs.getString("Title"));
    ps1.setNString(4, temp1);
    ps1.setNString(5, bibtitles_rs.getString("Title"));
    if (bibtitles_rs.getString("SubTitle") == null){
        ps1.setString(6, "");
    } else {
        ps1.setNString(6, bibtitles_rs.getString("SubTitle"));
    }
    ps1.setString(7, "");
    ps1.setString(8, "");
    ps1.executeUpdate();
    ps1.close();
}
bookstore_st1.close();

bookstore_st1 = bookstore_conn.createStatement();
ResultSet product_to_store_rs = bookstore_st1.executeQuery(
    "SELECT * FROM product_to_store WHERE product_id = " + id + " AND store_id = 0");
if (product_to_store_rs.first() == false) {
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_store (product_id, store_id)
VALUES(?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 0);
    ps1.executeUpdate();
    ps1.close();
}
bookstore_st1.close();

bookstore_st1 = bookstore_conn.createStatement();
ResultSet product_to_layout_rs = bookstore_st1.executeQuery(
    "SELECT * FROM product_to_layout WHERE product_id = " + id + " AND store_id = 0");
if (product_to_layout_rs.first() == false) {
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_layout (product_id,
store_id, layout_id) VALUES(?,?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 0);
    ps1.setInt(3, 0);
    ps1.executeUpdate();
    ps1.close();
}
bookstore_st1.close();
}
}
} catch (SQLException ex) {

```



```

    Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
}

}

*/
//-----
public static void Insert_Writer(Integer parent_id, String name, Integer product_id, Integer language) {
    PreparedStatement ps;
    Integer category_id;

    try {

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category " +
            "(category.parent_id, category.top, category.column, category.status,
category.date_added, category.date_modified) " +
            "VALUES(?,?,?,?,?)", Statement.RETURN_GENERATED_KEYS);

        ps.setInt(1, parent_id);
        ps.setBoolean(2, false);
        ps.setInt(3, 1);
        ps.setBoolean(4, true);
        ps.setTimestamp(5, new java.sql.Timestamp(new java.util.Date().getTime()));
        ps.setTimestamp(6, new java.sql.Timestamp(new java.util.Date().getTime()));
        ps.executeUpdate();

        ResultSet rs = ps.getGeneratedKeys();
        rs.next();
        category_id = rs.getInt(1);

        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_description " +
            "(category_description.category_id, category_description.language_id,
category_description.name, " +
            "category_description.description, category_description.meta_title,
category_description.meta_description, category_description.meta_keyword) " +
            "VALUES(?,?,?,?,""?, ""?)");
        ps.setInt(1, category_id);
        ps.setInt(2, language);
        ps.setNString(3, name.trim());
        ps.setNString(4, name.trim());
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, parent_id);
        ps.setInt(3, 0);
        ps.executeUpdate();
        ps.close();
    }
}

```

```

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, category_id);
        ps.setInt(3, 1);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_layout (category_id,
store_id, layout_id) VALUES(?,?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, 0);
        ps.setInt(3, 0);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_store (category_id,
store_id) VALUES(?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, 0);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO product_to_category (product_id,
category_id) VALUES(?,?)");
        ps.setInt(1, product_id);
        ps.setInt(2, category_id);
        ps.executeUpdate();
        ps.close();

    } catch (SQLException ex) {
        Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public static int Insert_Categories(int cat_id, int root_category, Integer language) throws SQLException{
    int retval = 0;
    int parent = root_category;

    ArrayList<String> aCatHierarchy=new ArrayList<String>();
    ArrayList<Integer> nCatHierarchy=new ArrayList<Integer>();

    while (cat_id != 0) {
        PreparedStatement ps = MainFrm.sqlbookstore_conn.prepareStatement("SELECT ParentID,
Subjects1 FROM dbo.BibSubjects WHERE SubjectsID = ?");
        ps.setInt(1, cat_id);
        ResultSet rs = ps.executeQuery();
        while (rs.next()){
            aCatHierarchy.add(rs.getNString("Subjects1"));
            cat_id = rs.getInt("ParentID");
        }
    }
}

```

```

Collections.reverse(aCatHierarchy);
nCatHierarchy.add(parent);

for(String str:aCatHierarchy){

    PreparedStatement ps = MainFrm.bookstore_conn.prepareStatement("SELECT category_id FROM
category_description WHERE name = ?");
    ps.setNString(1, str);
    ResultSet rs = ps.executeQuery();

    boolean newCategory = true;
    while (rs.next()) {
        parent = rs.getInt("category_id");

        newCategory = false;
    }
    if (newCategory) {
        parent = Insert_Category(parent, str, language);
    }

    nCatHierarchy.add(parent);
}

retval = parent;

for (int i=1; i< nCatHierarchy.size(); i++){

    for (int j=0; j<=i; j++) {

        PreparedStatement ps = MainFrm.bookstore_conn.prepareStatement("SELECT * FROM
category_path WHERE category_id = ? AND path_id = ?");
        ps.setInt(1, nCatHierarchy.get(i));
        ps.setInt(2, nCatHierarchy.get(j));
        ResultSet rs = ps.executeQuery();

        boolean found = false;

        while (rs.next()){
            found = true;
        }

        if (!found) {
            ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
            ps.setInt(1, nCatHierarchy.get(i));
            ps.setInt(2, nCatHierarchy.get(j));
            ps.setInt(3, j);
            ps.executeUpdate();
            ps.close();
        }

    }

}

return retval;

```



```

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_store (category_id,
store_id) VALUES(?,?)");
        ps.setInt(1, retval);
        ps.setInt(2, 0);
        ps.executeUpdate();
        ps.close();

    } catch (SQLException ex) {
        Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
    }

    return retval;
}
//-----
public static String[] Split(String data) {

    String[] retData = data.split("\\*\\*");
    for (int i=0; i < retData.length; i++){
        retData[i] = retData[i].replaceAll("[*]", "");
    }
    return retData;
}

/*private static void downloadUsingStream(String urlStr, String file) throws MalformedURLException,
IOException {
    URL url = new URL(urlStr);
    try (BufferedInputStream bis = new BufferedInputStream(url.openStream()); FileOutputStream fis =
new FileOutputStream(file)) {
        byte[] buffer = new byte[1024];
        int count=0;
        while((count = bis.read(buffer,0,1024)) != -1)
        {
            fis.write(buffer, 0, count);
        }
    }
}*/
//-----
}

```

## MainFrm.java

```

package bookstore_utilities;

import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JDesktopPane;

```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
import javax.swing.JFrame;
import javax.swing.JOptionPane;

/**
 *
 * @author vasilis
 */
public class MainFrm extends javax.swing.JFrame {

    public static MainFrm main;
    public static Connection bookstore_conn, sqlbookstore_conn;

    private javax.swing.JMenu jMenuItem1;
    private javax.swing.JMenu jMenuItem2;
    private javax.swing.JMenuItem jMenuItem1;
        private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JMenuBar mainMenu;

    /**
     * Creates new form MainFrm
     */
    public MainFrm() {
        initComponents();
    }

    private void addMenu(){
        mainMenu = new javax.swing.JMenuBar();
        jMenuItem1 = new javax.swing.JMenu();
        jMenuItem1 = new javax.swing.JMenuItem();
        jMenuItem2 = new javax.swing.JMenu();
        jMenuItem2 = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jMenuItem1.setText("ΜΕΤΑΒΑΣΤΕ ΣΤΗΝ ΟΜΑΔΑ MS SQL Of OpenCart");
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });

        private void jMenuItem1ActionPerformed(ActionEvent evt) {
            //UpdateProducts up = new UpdateProducts(bookstore_conn, sqlbookstore_conn, 13,);
            /*InsertProductsFrm InsertProducts_ProgressBar = new InsertProductsFrm(6547,);
            main.jDesktopPane1.add(InsertProducts_ProgressBar);
            Dimension DimMax = Toolkit.getDefaultToolkit().getScreenSize();
            Dimension jInternalFrameSize = InsertProducts_ProgressBar.getSize();
            InsertProducts_ProgressBar.setLocation((DimMax.width - jInternalFrameSize.width)/2,
                (DimMax.height- jInternalFrameSize.height)/2 - 150);

            InsertProducts_ProgressBar.setTitle("");
            InsertProducts_ProgressBar.setVisible(true);

            InsertProducts_ProgressBar.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE)
            ;*/

            ComsFrm ComsFrmScr = new ComsFrm();
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
ComsFrmScr.setTitle("");
main.jDesktopPane1.add(ComsFrmScr);
Dimension DimMax = Toolkit.getDefaultToolkit().getScreenSize();
Dimension jInternalFrameSize = ComsFrmScr.getSize();
ComsFrmScr.setLocation((DimMax.width - jInternalFrameSize.width)/2,
    (DimMax.height - jInternalFrameSize.height)/2 - 150);
ComsFrmScr.setVisible(true);
ComsFrmScr.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

}
});

jMenu1.add(jMenuItem1);

mainMenu.add(jMenu1);

jMenu2.setText(",");

jMenuItem2.setText("");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});

private void jMenuItem2ActionPerformed(ActionEvent evt) {
    try {
        bookstore_conn.close();
        sqlbookstore_conn.close();
        System.exit(0);
    } catch (SQLException ex) {
        Logger.getLogger(MainFrm.class.getName()).log(Level.SEVERE, null, ex);
    }
}

});
jMenu2.add(jMenuItem2);

mainMenu.add(jMenu2);

setJMenuBar(mainMenu);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
private void initComponents() {

    jDesktopPane1 = new javax.swing.JDesktopPane();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
```

```

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jDesktopPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 1168,
Short.MAX_VALUE)
            .addContainerGap())
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jDesktopPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 685,
Short.MAX_VALUE)
            .addContainerGap())
        );

pack();
} // </editor-fold> // GEN-END: initComponents

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(MainFrm.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(MainFrm.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(MainFrm.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(MainFrm.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
} //</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        LoginFrm login = new LoginFrm() {
            @Override

```



## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
void Connect() {
    try {
        bookstore_conn = DriverManager.getConnection(this.getMyConnectionString());
        sqlbookstore_conn = DriverManager.getConnection(this.getMSSConnectionString());
        main.addMenu();
        this.dispose();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "");
        Logger.getLogger(MainFrm.class.getName()).log(Level.SEVERE, null, ex);
    }
}
};
main = new MainFrm();
Dimension DimMax = Toolkit.getDefaultToolkit().getScreenSize();
main.setMaximumSize(DimMax);

main.setExtendedState(JFrame.MAXIMIZED_BOTH);
main.setVisible(true);
main.jDesktopPane1.add(login);

//Dimension desktopSize = main.jDesktopPane1.getSize();
Dimension jInternalFrameSize = login.getSize();
login.setLocation((DimMax.width - jInternalFrameSize.width)/2,
    (DimMax.height - jInternalFrameSize.height)/2);
login.setTitle("");
login.setIconifiable(true);
login.setMaximizable(true);
login.setClosable(true);
login.setVisible(true);
login.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
}
});

}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JDesktopPane jDesktopPane1;
// End of variables declaration//GEN-END:variables

public static JDesktopPane getMainDesktop(){
    return main.jDesktopPane1;
}

}
```

### InsertProducts.frm

```
package bookstore_utilities;
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
import java.io.BufferedInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.net.MalformedURLException;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Calendar;

import java.util.logging.Level;

import java.util.logging.Logger;

/**
 *
 * @author vasilis
 */

public class InsertProductsFrm extends javax.swing.JInternalFrame {

    static Integer Language = 2;

    static Integer Writer_Category = 60;

    static Integer Category_Category = 59;

    static Connection bookstore_conn = MainFrm.bookstore_conn;

    static Connection sqlbookstore_conn = MainFrm.sqlbookstore_conn;

    static String SearchStr;

    static String StrManufacturer;

    /**
     * Creates new form InsertProducts
     */

    public InsertProductsFrm(String Search, String Manufacturer) {
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
initComponents();

SearchStr = Search;
StrManufacturer = Manufacturer.replaceAll("\\", "");

percent.setVisible(false);
done.setVisible(false);
}

//The thread
public static class thread1 implements Runnable{
    public void run(){
        String SourceBigimageURL = "http://www.biblionet.gr/images/covers/b";
        String SourceSmallimageURL = "http://www.biblionet.gr/images/covers/s";
        String TargetimageURL = "/home/vasilis/www/html/eshop/image/catalog/kalokathis/";
        String DefaultimageURL = "/home/vasilis/www/html/eshop/image/catalog/default-image.png";
        java.sql.Date date = new java.sql.Date(Calendar.getInstance().getTime().getTime());

        Statement sqlbookstore_st = null;

        Statement bookstore_st1 = null;

        PreparedStatement ps1;

        int counter = 0;
        int ProgressBar_Setter = 0;

        try {
            sqlbookstore_st = sqlbookstore_conn.createStatement();
        } catch (SQLException ex) {
            Logger.getLogger(InsertProductsFrm.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```
}  
  
ResultSet rs = null;  
  
try {  
    rs = sqlbookstore_st.executeQuery("SELECT COUNT(*) AS nunrows FROM dbo.BibTitles WHERE  
Titles_Com LIKE '%" + SearchStr + "%'");  
} catch (SQLException ex) {  
    Logger.getLogger(InsertProductsFrm.class.getName()).log(Level.SEVERE, null, ex);  
}  
  
try {  
    rs.next();  
} catch (SQLException ex) {  
    Logger.getLogger(InsertProductsFrm.class.getName()).log(Level.SEVERE, null, ex);  
}  
  
try {  
    counter = rs.getInt("nunrows");  
} catch (SQLException ex) {  
    Logger.getLogger(InsertProductsFrm.class.getName()).log(Level.SEVERE, null, ex);  
}  
  
try {  
    rs.close();  
} catch (SQLException ex) {  
    Logger.getLogger(InsertProductsFrm.class.getName()).log(Level.SEVERE, null, ex);  
}  
  
try (ResultSet bibtitles_rs = sqlbookstore_st.executeQuery("SELECT * FROM dbo.BibTitles WHERE  
Titles_Com LIKE '%" + SearchStr + "%'")) {  
    int i = 0;  
  
    while (bibtitles_rs.next()){  
        i++;  
        int id = bibtitles_rs.getInt("TitlesID");  
        String Writer = bibtitles_rs.getNString("Writer");  
        if (Writer == null) {  
            Writer = ",";  
        }  
    }  
}
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
}  
  
String aaa = bibtitles_rs.getNString("Titles_Subjects");  
  
if (aaa == null) {  
    aaa = "59";  
}  
  
//System.out.println(aaa + "\n");  
  
//String[] Categories = Actions.Split(bibtitles_rs.getNString("Titles_Subjects"));  
  
String[] Categories = Actions.Split(aaa);  
  
  
try (Statement bookstore_st = bookstore_conn.createStatement()) {  
    ResultSet product_rs = bookstore_st.executeQuery("SELECT * FROM product WHERE  
product_id = " + id);  
  
  
    if (product_rs.first() == false) {  
  
        bookstore_st1 = bookstore_conn.createStatement();  
        ResultSet manufacturer_rs = bookstore_st1.executeQuery(  
            "SELECT * FROM manufacturer WHERE manufacturer_id = " +  
Integer.parseInt(SearchStr));  
        if (manufacturer_rs.first() == false) {  
            ps1 = bookstore_conn.prepareStatement("INSERT INTO manufacturer (manufacturer_id,  
name, sort_order) VALUES(?,?,?)");  
            ps1.setInt(1, Integer.parseInt(SearchStr));  
            ps1.setNString(2, StrManufacturer);  
            ps1.setInt(3, 0);  
            ps1.executeUpdate();  
            ps1.close();  
        }  
        bookstore_st1.close();  
  
  
        bookstore_st1 = bookstore_conn.createStatement();  
        ResultSet manufacturer_to_store_rs = bookstore_st1.executeQuery(  

```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
"SELECT * FROM manufacturer_to_store WHERE manufacturer_id = " +
Integer.parseInt(SearchStr)
    + " AND store_id = 0");

if (manufacturer_to_store_rs.first() == false) {

    ps1 = bookstore_conn.prepareStatement("INSERT INTO manufacturer_to_store
(manufacturer_id, store_id) VALUES(?,?)");

    ps1.setInt(1, Integer.parseInt(SearchStr));

    ps1.setInt(2, 0);

    ps1.executeUpdate();

    ps1.close();

}

bookstore_st1.close();

PreparedStatement ps = bookstore_conn.prepareStatement("INSERT INTO product
(product_id, model, sku, upc, ean, jan, isbn, location, "
    + "quantity, stock_status_id, image, manufacturer_id, shipping, price, points, tax_class_id,
date_available, "
    + "weight, weight_class_id, length, width, height, length_class_id, subtract, minimum,
sort_order, status, "
    + "viewed, date_added, date_modified, mpn)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");

ps.setInt(1, id);

ps.setString(2, Integer.toString(id));

ps.setString(3, "");

ps.setString(4, "");

ps.setString(5, "");

ps.setString(6, "");

if (bibtitles_rs.getString("ISBN") != null) {

    ps.setNString(7, bibtitles_rs.getString("ISBN").replaceAll("-", ""));

} else {

    ps.setString(7, "");

}
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
if (bibtitles_rs.getString("Places1") == null) {
    ps.setNString(8, " ");
} else {
    ps.setNString(8, bibtitles_rs.getString("Places1"));
}

ps.setInt(9, 1);
ps.setInt(10, 6);
ps.setString(11, DefaultimageURL);

if (bibtitles_rs.getBoolean("HasSmCov")){
    try {

downloadUsingStream(SourceSmallimageURL+Integer.toString(bibtitles_rs.getInt("TitlesID"))
    , TargetimageURL+Integer.toString(bibtitles_rs.getInt("TitlesID")));
    ps.setString(11, TargetimageURL+Integer.toString(bibtitles_rs.getInt("TitlesID")));
    } catch (IOException ex) {
    ps.setString(11, DefaultimageURL);
    }
}

if (bibtitles_rs.getBoolean("HasBigCov")){
    try {

downloadUsingStream(SourceBigimageURL+Integer.toString(bibtitles_rs.getInt("TitlesID"))
    , TargetimageURL+Integer.toString(bibtitles_rs.getInt("TitlesID")));
    ps.setString(11, TargetimageURL+Integer.toString(bibtitles_rs.getInt("TitlesID")));
    } catch (IOException ex) {
    ps.setString(11, DefaultimageURL);
    }
}

ps.setInt(12, Integer.parseInt(SearchStr));
ps.setInt(13, 1);
ps.setFloat(14, bibtitles_rs.getFloat("Price"));
```

```
ps.setInt(15, 0);
ps.setInt(16, 11);
ps.setDate(17, date);
ps.setDouble(18, bibtitles_rs.getDouble("Weight"));
ps.setInt(19, 2);
ps.setDouble(20, 0);
ps.setDouble(21, 0);
ps.setDouble(22, 0);
ps.setInt(23, 1);
ps.setInt(24, 0);
ps.setInt(25, 1);
ps.setInt(26, 0);
ps.setInt(27, 1);
ps.setInt(28, 0);
ps.setDate(29, date);
ps.setDate(30, date);
ps.setString(31, "");

ps.executeUpdate();
ps.close();

}

bookstore_st1 = bookstore_conn.createStatement();
String temp = "SELECT * FROM product_description WHERE product_id = " + id + " AND
language_id = " + Language;
ResultSet product_description_rs = bookstore_st1.executeQuery(temp);

if (product_description_rs.first() == false) {
    String temp1;
    if (bibtitles_rs.getString("Descr") == null){
        temp1="";
    } else {
```



## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
temp1 = bibtitles_rs.getString("Descr");
}

ps1 = bookstore_conn.prepareStatement("INSERT INTO product_description (product_id,
language_id, name, description, "
+ "meta_title, meta_description, tag, meta_keyword) VALUES(?,?,?,?,?,?,?)");
ps1.setInt(1, id);
ps1.setInt(2, 2);
ps1.setNString(3, bibtitles_rs.getString("Title"));
ps1.setNString(4, temp1);
ps1.setNString(5, bibtitles_rs.getString("Title"));
if (bibtitles_rs.getString("SubTitle") == null){
    ps1.setString(6, "");
} else {
    ps1.setNString(6, "");
}
ps1.setString(7, "");
ps1.setString(8, "");
ps1.executeUpdate();
ps1.close();

}

bookstore_st1.close();

bookstore_st1 = bookstore_conn.createStatement();
ResultSet product_to_store_rs = bookstore_st1.executeQuery(
    "SELECT * FROM product_to_store WHERE product_id = " + id + " AND store_id = 0");
if (product_to_store_rs.first() == false) {
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_store (product_id,
store_id) VALUES(?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 0);
    ps1.executeUpdate();
    ps1.close();
}
}
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
bookstore_st1.close();

bookstore_st1 = bookstore_conn.createStatement();
ResultSet product_to_layout_rs = bookstore_st1.executeQuery(
    "SELECT * FROM product_to_layout WHERE product_id = " + id + " AND store_id = 0");
if (product_to_layout_rs.first() == false) {
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_layout (product_id,
store_id, layout_id) VALUES(?,?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 0);
    ps1.setInt(3, 0);
    ps1.executeUpdate();
    ps1.close();
}
bookstore_st1.close();

bookstore_st1 = bookstore_conn.createStatement();
ResultSet product_to_category_rs = bookstore_st1.executeQuery(
    "SELECT * FROM product_to_category WHERE product_id = " + id + " AND category_id =
59");
if (product_to_category_rs.first() == false) {
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_category (product_id,
category_id) VALUES(?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 59);
    ps1.executeUpdate();
    ps1.close();
}
bookstore_st1.close();

PreparedStatement ps = MainFrm.bookstore_conn.prepareStatement("SELECT * FROM
category_description WHERE name = ?");
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
ps.setNString(1, Writer);

ResultSet writer_rs = ps.executeQuery();

if (writer_rs.first() == false) {
    Actions.Insert_Writer(Writer_Category, Writer, id, Language);
} else {
    Integer cat_id = writer_rs.getInt("category_id");

    ps = MainFrm.bookstore_conn.prepareStatement("SELECT * FROM product_to_category
WHERE product_id = ? AND category_id = ?");

    ps.setInt(1, id);

    ps.setInt(2, cat_id);

    writer_rs = ps.executeQuery();

    if (writer_rs.first() == false) {

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO product_to_category
(product_id, category_id) VALUES(?,?)");

        ps.setInt(1, id);

        ps.setInt(2, cat_id);

        ps.executeUpdate();

    }
}

ps.close();

for (int j=0; j<Categories.length; j++){

    ps = MainFrm.sqlbookstore_conn.prepareStatement("SELECT Subjects1 FROM
dbo.BibSubjects WHERE SubjectsID = ?");

    ps.setInt(1, Integer.parseInt(Categories[j]));

    rs = ps.executeQuery();

    while (rs.next()) {

        int cat_id = 0;

        String Subject = rs.getNString("Subjects1");

        ps = MainFrm.bookstore_conn.prepareStatement("SELECT category_id FROM
category_description WHERE name = ?");

        ps.setNString(1, Subject);

        rs = ps.executeQuery();
```

## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
boolean newCategory = true;
while (rs.next()){
    newCategory = false;
    cat_id = rs.getInt("category_id");
}
if (newCategory){
    cat_id = Actions.Insert_Categories(Integer.parseInt(Categories[j]), Category_Category,
Language);
}

ps = MainFrm.bookstore_conn.prepareStatement("SELECT * FROM product_to_category
WHERE product_id = ? AND category_id = ?");
ps.setInt(1, id);
ps.setInt(2, cat_id);
rs = ps.executeQuery();
boolean newProduct = true;
while (rs.next()){
    newProduct = false;
}

if (newProduct) {
    ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO product_to_category
(product_id, category_id) VALUES(?,?)");
    ps.setInt(1, id);
    ps.setInt(2, cat_id);
    ps.executeUpdate();
    ps.close();
}

}
}
}
```

```
if (i>=counter){
    ProgressBar_Setter = 100;
} else {
    ProgressBar_Setter = (i*100)/counter;
}

percent.setText(Integer.toString(ProgressBar_Setter)+"%");
percent.repaint();

progressbar.setValue(ProgressBar_Setter); //Set value
progressbar.repaint(); //Refresh graphics

try{Thread.sleep(1);} //Sleep 10 milliseconds
catch (InterruptedException err){}

}

done.setVisible(true);
} catch (SQLException ex) {
    Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void downloadUsingStream(String urlStr, String file) throws MalformedURLException,
IOException {
    URL url = new URL(urlStr);

    try (BufferedInputStream bis = new BufferedInputStream(url.openStream()); FileOutputStream fis =
new FileOutputStream(file)) {
        byte[] buffer = new byte[1024];
        int count=0;
        while((count = bis.read(buffer,0,1024)) != -1) {
            fis.write(buffer, 0, count);
        }
    }
}
```

```
    }  
    }  
}  
  
/**  
 * This method is called from within the constructor to initialize the form.  
 * WARNING: Do NOT modify this code. The content of this method is always  
 * regenerated by the Form Editor.  
 */  
@SuppressWarnings("unchecked")  
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents  
private void initComponents() {  
  
    jPanel1 = new javax.swing.JPanel();  
    progressbar = new javax.swing.JProgressBar();  
    percent = new javax.swing.JLabel();  
    go = new javax.swing.JButton();  
    done = new javax.swing.JButton();  
  
    setClosable(true);  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);  
  
    percent.setText("0%");  
  
    go.setText("Go...");  
    go.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            goActionPerformed(evt);  
        }  
    });  
  
    done.setText("Done!");  
    done.addActionListener(new java.awt.event.ActionListener() {
```



```
.addComponent(done)
    .addContainerGap(36, Short.MAX_VALUE)
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap(36)
        )
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        )
);

pack();
} // </editor-fold> //GEN-END:initComponents

private void goActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_goActionPerformed
    go.setVisible(false);
    percent.setVisible(true);
    this.closable = false;
    new Thread(new thread1()).start(); //Start the thread
} //GEN-LAST:event_goActionPerformed

private void doneActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_doneActionPerformed
```



```
        this.dispose();
    } //GEN-LAST:event_doneActionPerformed

    // Variables declaration - do not modify //GEN-BEGIN:variables
    private static javax.swing.JButton done;
    private javax.swing.JButton go;
    private javax.swing.JPanel jPanel1;
    private static javax.swing.JLabel percent;
    private static javax.swing.JProgressBar progressbar;
    // End of variables declaration //GEN-END:variables
}
```

## **ComsFrm.java**

```
package bookstore_utilities;

import static com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type.Int;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultComboBoxModel;

/**
 *
 * @author vasilis
 */
public class ComsFrm extends javax.swing.JInternalFrame {

    private ArrayList<String> Comslist=new ArrayList<String>();

    /**
     * Creates new form ComsFrm
     */
    public ComsFrm() {
        initComponents();
        fillCombo();
    }

    /**
```

```

* This method is called from within the constructor to initialize the form.
* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    ComsList = new javax.swing.JComboBox<>();
    jPanel3 = new javax.swing.JPanel();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();

    setClosable(true);
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setIconifiable(true);
    setMaximizable(true);

    jLabel1.setFont(new java.awt.Font("Ubuntu", 1, 18)); // NOI18N
    jLabel1.setText(" OK");

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addContainerGap(100, true))
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jButton1)
                .addComponent(jButton2))
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton2)
            )
    );

    ComsList.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Item 1", "Item 2",
    "Item 3", "Item 4" }));

    jButton1.setText("OK");
    jButton1.setToolTipText("");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });
}

```

```

});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()
            .addComponent(jButton2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 185,
                javax.swing.GroupLayout.PREFERRED_SIZE))
        );
jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(jPanel3Layout.createSequentialGroup()
                .addGap()
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton1)
                    .addComponent(jButton2))
                .addGap(58, Short.MAX_VALUE))
        );

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jPanel2Layout.createSequentialGroup()
                .addGap()
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
                    false)
                    .addComponent(comsList, 0, 515, Short.MAX_VALUE)
                    .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(23, Short.MAX_VALUE))
        );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(comsList, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 88, Short.MAX_VALUE))
        );

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(layout.createSequentialGroup()
                .addGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
        );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_jButton1ActionPerformed
    String retvalue = ComsList.getSelectedItem().toString();
    String[] splitted = retvalue.split("\\|");
    InsertProductsFrm InsertProducts_ProgressBar = new InsertProductsFrm(splitted[0].trim(),
splitted[1].trim());
    MainFrm.getMainDesktop().add(InsertProducts_ProgressBar);
    Dimension DimMax = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension jInternalFrameSize = InsertProducts_ProgressBar.getSize();
    InsertProducts_ProgressBar.setLocation((DimMax.width - jInternalFrameSize.width)/2,
(DimMax.height - jInternalFrameSize.height)/2 - 150);

    InsertProducts_ProgressBar.setTitle("");
    InsertProducts_ProgressBar.setVisible(true);

    InsertProducts_ProgressBar.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE)
;
} // GEN-LAST:event_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_jButton2ActionPerformed
    this.dispose();
} // GEN-LAST:event_jButton2ActionPerformed

// Variables declaration - do not modify // GEN-BEGIN:variables
private javax.swing.JComboBox<String> ComsList;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
// End of variables declaration // GEN-END:variables

private void fillCombo() {

```

```
try {
    fillComsList();
} catch (SQLException ex) {
    Logger.getLogger(ComsFrm.class.getName()).log(Level.SEVERE, null, ex);
}
DefaultComboBoxModel model = new DefaultComboBoxModel(Comslist.toArray());
ComsList.setModel(model);
}

private void fillComsList() throws SQLException {
    Statement sqlbookstore_st = MainFrm.sqlbookstore_conn.createStatement();
    ResultSet bibcoms_rs = sqlbookstore_st.executeQuery("SELECT ComID, Name FROM dbo.BibComs
ORDER BY Name");

    while (bibcoms_rs.next()){
        Comslist.add(Integer.toString(bibcoms_rs.getInt("ComID")) + " | " + bibcoms_rs.getString("Name"));
    }

}
}
```

## Actions.java

```
package bookstore_utilities;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Collections;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author vasilis
 */
public class Actions {
    java.sql.Timestamp mydate = new java.sql.Timestamp(new java.util.Date().getTime());

    //-----Update All Categories tree sql1.biblionet.gr-----

    /* public static void updateCategories(Connection bookstore_conn, Connection sqlbookstore_conn)
throws SQLException {

        Integer product_id;
        String Titles_Subjects;
        String[] categoriesID;

        int metritis = 0;
```

```

try (Statement bookstore_st = bookstore_conn.createStatement()) {
    ResultSet product_rs = bookstore_st.executeQuery("SELECT * FROM product");

    while (product_rs.next()){
        product_id = product_rs.getInt("product_id");
        metritis++;
        System.out.println(product_id + " " + Integer.toString(metritis));

        try (Statement sqlbookstore_st = sqlbookstore_conn.createStatement()) {
            ResultSet sqlbookstore_rs = sqlbookstore_st.executeQuery("SELECT Titles_Subjects FROM
dbo.BibTitles WHERE TitlesID = " +
                product_id);

            if (sqlbookstore_rs.next()){
                Titles_Subjects = sqlbookstore_rs.getString("Titles_Subjects");
                categoriesID = Split(Titles_Subjects);

                Update_Category_Table(sqlbookstore_conn, bookstore_conn, categoriesID);
                Product_to_Categories(bookstore_conn, product_id, categoriesID);
            }
        }
    }
}

private static void Update_Category_Table(Connection sqlbookstore_conn, Connection bookstore_conn,
String[] categories_id) throws SQLException{
    String selStr;

    for (int i=0; i<categories_id.length; i++){
        Integer categoryID = Integer.parseInt(categories_id[i]);
        selStr = "SELECT * FROM category WHERE category_id = " + categoryID;
        try (Statement bookstore_st = bookstore_conn.createStatement()) {
            ResultSet category_rs = bookstore_st.executeQuery(selStr);
            if (!category_rs.next()){
                try (Statement sqlbookstore_st = sqlbookstore_conn.createStatement()) {
                    selStr = "SELECT SubjectsID, Subjects1, ParentID FROM dbo.BibSubjects WHERE SubjectsID = "
+ categoryID;
                    ResultSet BibSubjects_rs = sqlbookstore_st.executeQuery(selStr);
                    while (BibSubjects_rs.next()){
                        Insert_Category_Table(bookstore_conn, BibSubjects_rs.getInt("SubjectsID"),
                            BibSubjects_rs.getString("Subjects1"), BibSubjects_rs.getInt("ParentID"));
                    }

                    while (BibSubjects_rs.getInt("ParentID") != 0){
                        selStr = "SELECT SubjectsID, Subjects1, ParentID FROM dbo.BibSubjects WHERE SubjectsID =
" + BibSubjects_rs.getInt("ParentID");
                        BibSubjects_rs = sqlbookstore_st.executeQuery(selStr);
                        while (BibSubjects_rs.next()){
                            selStr = "SELECT * FROM category WHERE category_id = " + categoryID;
                            category_rs = bookstore_st.executeQuery(selStr);
                            if (!category_rs.next()){
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }
    }
}

```

```

private static void Insert_Category_Table(Connection bookstore_conn, Integer SubjectsID, String
Subjects1, Integer ParentID) throws SQLException{
    PreparedStatement ps;

    ps = bookstore_conn.prepareStatement("INSERT INTO category (category_id, parent_id, top, status,
date_added, date_modified) VALUES(?,?,0,1,?,?)");
    ps.setInt(1, SubjectsID);
    ps.setInt(2, ParentID);
    ps.setTimestamp(3, new java.sql.Timestamp(new java.util.Date().getTime()));
    ps.setTimestamp(4, new java.sql.Timestamp(new java.util.Date().getTime()));
    ps.executeUpdate();

    ps = bookstore_conn.prepareStatement("INSERT INTO category_description (category_id, language_id,
name, meta_title) VALUES(?,?,?,?)");
    ps.setInt(1, SubjectsID);
    ps.setInt(2, 2);
    ps.setNString(3, Subjects1);
    ps.setNString(4, Subjects1);
    ps.executeUpdate();

    ps = bookstore_conn.prepareStatement("INSERT INTO category_to_store (category_id, store_id)
VALUES(?,?)");
    ps.setInt(1, SubjectsID);
    ps.setInt(2, 0);
    ps.executeUpdate();

    ps = bookstore_conn.prepareStatement("INSERT INTO category_to_layout (category_id, store_id,
layout_id) VALUES(?,?,?)");
    ps.setInt(1, SubjectsID);
    ps.setInt(2, 0);
    ps.setInt(3, 0);
    ps.executeUpdate();

}

```

```

private static void Product_to_Categories(Connection conn, Integer product_id, String[] categories_id)
throws SQLException{
    String selStr;

    for (int i=0; i<categories_id.length; i++){
        Integer categoryID = Integer.parseInt(categories_id[i]);

        while(categoryID != 0){
            selStr = "SELECT * FROM category WHERE category_id = " + categoryID;
            Statement st2;
            try (Statement st1 = conn.createStatement()) {

```

```

        ResultSet categories_rs = st1.executeQuery(selStr);
        categories_rs.first();
        Integer parentID = categories_rs.getInt("parent_id");
        st2 = conn.createStatement();
        ResultSet product_to_category_rs = st2.executeQuery("SELECT * FROM product_to_category
WHERE category_id = "
                + categoryID + " AND product_id = " + product_id);
        if (product_to_category_rs.first() == false) {
            PreparedStatement ps = conn.prepareStatement("INSERT INTO product_to_category
(product_id, category_id) VALUES(?,?)");
            ps.setInt(1, product_id);
            ps.setInt(2, categoryID);
            ps.executeUpdate();
        } categoryID = parentID;
    }
    st2.close();
}
}

*/
//-----Update Products from sql1.biblionet.gr-----
/*
public static void old_updateProducts(Connection bookstore_conn, Connection sqlbookstore_conn, int
ComID, String SearchStr) throws SQLException, UnsupportedEncodingException, InterruptedException {
    String SourceBigimageURL = "http://www.biblionet.gr/images/covers/b";
    String SourceSmallimageURL = "http://www.biblionet.gr/images/covers/s";
    String TargetimageURL = "/home/vasilis/www/html/eshop/image/catalog/kalokathis/";
    String DefaultimageURL = "/home/vasilis/www/html/eshop/image/catalog/default-image.png";
    java.sql.Date date = new java.sql.Date(Calendar.getInstance().getTime().getTime());

    Statement sqlbookstore_st = sqlbookstore_conn.createStatement();
    ResultSet rs = sqlbookstore_st.executeQuery("SELECT COUNT(*) AS nunrows FROM dbo.BibTitles
WHERE Publisher LIKE (N'" + SearchStr + "'");
    rs.next();
    int counter = rs.getInt("nunrows")/100;
    rs.close();

    try (ResultSet bibtitles_rs = sqlbookstore_st.executeQuery("SELECT * FROM dbo.BibTitles WHERE
Publisher LIKE (N'" + SearchStr + "'")) {
        int i = 0;

        while (bibtitles_rs.next()){
            i++;
            int id = bibtitles_rs.getInt("TitlesID");
            //String[] categoriesID;
            try (Statement bookstore_st = bookstore_conn.createStatement()) {
                ResultSet product_rs = bookstore_st.executeQuery("SELECT * FROM product WHERE product_id =
" + id);

                if (product_rs.first() == false) {

                    PreparedStatement ps = bookstore_conn.prepareStatement("INSERT INTO product (product_id,
model, sku, upc, ean, jan, isbn, location, "
                            + "quantity, stock_status_id, image, manufacturer_id, shipping, price, points,
tax_class_id, date_available, "

```



## Διασύνδεση Java με πλατφόρμες διαχείρισης e-commerce

```
+ "weight, weight_class_id, length, width, height, length_class_id, subtract, minimum,  
sort_order, status, "  
+ "viewed, date_added, date_modified, mpn)  
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
```

```
ps.setInt(1, id);  
ps.setString(2, Integer.toString(id));  
ps.setString(3, "");  
ps.setString(4, "");  
ps.setString(5, "");  
ps.setString(6, "");
```

```
if (bibtitles_rs.getString("ISBN") != null) {  
    ps.setNString(7, bibtitles_rs.getString("ISBN").replaceAll("-", ""));  
} else {  
    ps.setString(7, "");  
}  
ps.setNString(8, bibtitles_rs.getString("Places1"));  
ps.setInt(9, 1);  
ps.setInt(10, 6);  
ps.setString(11, DefaultimageUrl);
```

```
if (bibtitles_rs.getBoolean("HasSmCov")){  
    try {
```

```
downloadUsingStream(SourceSmallimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID"))  
    , TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));  
    ps.setString(11, TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));  
    } catch (IOException ex) {  
        ps.setString(11, DefaultimageUrl);  
    }  
}
```

```
if (bibtitles_rs.getBoolean("HasBigCov")){  
    try {  
        downloadUsingStream(SourceBigimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID"))  
            , TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));  
        ps.setString(11, TargetimageUrl+Integer.toString(bibtitles_rs.getInt("TitlesID")));  
    } catch (IOException ex) {  
        ps.setString(11, DefaultimageUrl);  
    }  
}
```

```
ps.setInt(12, ComID);  
ps.setInt(13, 1);  
ps.setFloat(14, bibtitles_rs.getFloat("Price"));  
ps.setInt(15, 0);  
ps.setInt(16, 11);  
ps.setDate(17, date);  
ps.setDouble(18, bibtitles_rs.getDouble("Weight"));  
ps.setInt(19, 2);  
ps.setDouble(20, 0);  
ps.setDouble(21, 0);  
ps.setDouble(22, 0);  
ps.setInt(23, 1);  
ps.setInt(24, 0);  
ps.setInt(25, 1);
```

```

        ps.setInt(26, 0);
        ps.setInt(27, 1);
        ps.setInt(28, 0);
        ps.setDate(29, date);
        ps.setDate(30, date);
        ps.setString(31, "");

        ps.executeUpdate();
        ps.close();

    }

    Statement bookstore_st1 = bookstore_conn.createStatement();
    String temp = "SELECT * FROM product_description WHERE product_id = " + id + " AND
language_id = 2";
    ResultSet product_description_rs = bookstore_st1.executeQuery(temp);
    PreparedStatement ps1;
    if (product_description_rs.first() == false) {
        String temp1;
        if (bibtitles_rs.getString("Descr") == null){
            temp1="";
        } else {
            temp1 = bibtitles_rs.getString("Descr");
        }
        ps1 = bookstore_conn.prepareStatement("INSERT INTO product_description (product_id,
language_id, name, description, "
            + "meta_title, meta_description, tag, meta_keyword) VALUES(?,?,?,?,?,?,?)");
        ps1.setInt(1, id);
        ps1.setInt(2, 2);
        ps1.setNString(3, bibtitles_rs.getString("Title"));
        ps1.setNString(4, temp1);
        ps1.setNString(5, bibtitles_rs.getString("Title"));
        if (bibtitles_rs.getString("SubTitle") == null){
            ps1.setString(6, "");
        } else {
            ps1.setNString(6, bibtitles_rs.getString("SubTitle"));
        }
        ps1.setString(7, "");
        ps1.setString(8, "");
        ps1.executeUpdate();
        ps1.close();

    }
    bookstore_st1.close();

    bookstore_st1 = bookstore_conn.createStatement();
    ResultSet product_to_store_rs = bookstore_st1.executeQuery(
        "SELECT * FROM product_to_store WHERE product_id = " + id + " AND store_id = 0");
    if (product_to_store_rs.first() == false) {
        ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_store (product_id, store_id)
VALUES(?,?)");
        ps1.setInt(1, id);
        ps1.setInt(2, 0);
        ps1.executeUpdate();
        ps1.close();
    }
    bookstore_st1.close();

```

```

bookstore_st1 = bookstore_conn.createStatement();
ResultSet product_to_layout_rs = bookstore_st1.executeQuery(
    "SELECT * FROM product_to_layout WHERE product_id = " + id + " AND store_id = 0");
if (product_to_layout_rs.first() == false) {
    ps1 = bookstore_conn.prepareStatement("INSERT INTO product_to_layout (product_id,
store_id, layout_id) VALUES(?,?,?)");
    ps1.setInt(1, id);
    ps1.setInt(2, 0);
    ps1.setInt(3, 0);
    ps1.executeUpdate();
    ps1.close();
}
bookstore_st1.close();
}
}
} catch (SQLException ex) {
    Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

}

\*/

//-----

```

public static void Insert_Writer(Integer parent_id, String name, Integer product_id, Integer language) {
    PreparedStatement ps;
    Integer category_id;

    try {

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category " +
            "(category.parent_id, category.top, category.column, category.status,
category.date_added, category.date_modified) " +
            "VALUES(?,?,?,?,?,?)", Statement.RETURN_GENERATED_KEYS);

        ps.setInt(1, parent_id);
        ps.setBoolean(2, false);
        ps.setInt(3, 1);
        ps.setBoolean(4, true);
        ps.setTimestamp(5, new java.sql.Timestamp(new java.util.Date().getTime()));
        ps.setTimestamp(6, new java.sql.Timestamp(new java.util.Date().getTime()));
        ps.executeUpdate();

        ResultSet rs = ps.getGeneratedKeys();
        rs.next();
        category_id = rs.getInt(1);

        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_description " +
            "(category_description.category_id, category_description.language_id,
category_description.name, " +

```

```

        "category_description.description, category_description.meta_title,
category_description.meta_description, category_description.meta_keyword) " +
        "VALUES(?,?,?,?,";
        ps.setInt(1, category_id);
        ps.setInt(2, language);
        ps.setNString(3, name.trim());
        ps.setNString(4, name.trim());
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, parent_id);
        ps.setInt(3, 0);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, category_id);
        ps.setInt(3, 1);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_layout (category_id,
store_id, layout_id) VALUES(?,?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, 0);
        ps.setInt(3, 0);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_store (category_id,
store_id) VALUES(?,?)");
        ps.setInt(1, category_id);
        ps.setInt(2, 0);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO product_to_category (product_id,
category_id) VALUES(?,?)");
        ps.setInt(1, product_id);
        ps.setInt(2, category_id);
        ps.executeUpdate();
        ps.close();

    } catch (SQLException ex) {
        Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public static int Insert_Categories(int cat_id, int root_category, Integer language) throws SQLException{

```

```

int retval = 0;
int parent = root_category;

ArrayList<String> aCatHierarchy=new ArrayList<String>();
ArrayList<Integer> nCatHierarchy=new ArrayList<Integer>();

while (cat_id != 0) {
    PreparedStatement ps = MainFrm.sqlbookstore_conn.prepareStatement("SELECT ParentID,
Subjects1 FROM dbo.BibSubjects WHERE SubjectsID = ?");
    ps.setInt(1, cat_id);
    ResultSet rs = ps.executeQuery();
    while (rs.next()){
        aCatHierarchy.add(rs.getNString("Subjects1"));
        cat_id = rs.getInt("ParentID");
    }
}

Collections.reverse(aCatHierarchy);
nCatHierarchy.add(parent);

for(String str:aCatHierarchy){

    PreparedStatement ps = MainFrm.bookstore_conn.prepareStatement("SELECT category_id FROM
category_description WHERE name = ?");
    ps.setNString(1, str);
    ResultSet rs = ps.executeQuery();

    boolean newCategory = true;
    while (rs.next()) {
        parent = rs.getInt("category_id");

        newCategory = false;
    }
    if (newCategory) {
        parent = Insert_Category(parent, str, language);
    }

    nCatHierarchy.add(parent);
}

retval = parent;

for (int i=1; i< nCatHierarchy.size(); i++){

    for (int j=0; j<=i; j++) {

        PreparedStatement ps = MainFrm.bookstore_conn.prepareStatement("SELECT * FROM
category_path WHERE category_id = ? AND path_id = ?");
        ps.setInt(1, nCatHierarchy.get(i));
        ps.setInt(2, nCatHierarchy.get(j));
        ResultSet rs = ps.executeQuery();

        boolean found = false;

        while (rs.next()){
            found = true;

```

```

    }

    if (!found) {
        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
        ps.setInt(1, nCatHierarchy.get(i));
        ps.setInt(2, nCatHierarchy.get(j));
        ps.setInt(3, j);
        ps.executeUpdate();
        ps.close();
    }

}

}

return retval;
}

public static int Insert_Category(int parent_id, String name, int language) {
    int retval = 0;

    PreparedStatement ps;

    try {

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category " +
"(category.parent_id, category.top, category.column, category.status,
category.date_added, category.date_modified) " +
"VALUES(?,?,?,?,?,?)", Statement.RETURN_GENERATED_KEYS);

        ps.setInt(1, parent_id);
        ps.setBoolean(2, false);
        ps.setInt(3, 1);
        ps.setBoolean(4, true);
        ps.setTimestamp(5, new java.sql.Timestamp(new java.util.Date().getTime()));
        ps.setTimestamp(6, new java.sql.Timestamp(new java.util.Date().getTime()));
        ps.executeUpdate();

        ResultSet rs = ps.getGeneratedKeys();
        rs.next();
        retval = rs.getInt(1);

        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_description " +
"(category_description.category_id, category_description.language_id,
category_description.name, " +
"category_description.description, category_description.meta_title,
category_description.meta_description, category_description.meta_keyword) " +
"VALUES(?,?,?,?," +
"?)");
        ps.setInt(1, retval);
        ps.setInt(2, language);
        ps.setNString(3, name.trim());
        ps.setNString(4, name.trim());
        ps.executeUpdate();
        ps.close();
    }
}

```

```

        /*ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_path (category_id,
path_id, level) VALUES(?,?,?)");
        ps.setInt(1, retval);
        ps.setInt(2, retval);
        ps.setInt(3, 0);
        ps.executeUpdate();
        ps.close(); */

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_layout (category_id,
store_id, layout_id) VALUES(?,?,?)");
        ps.setInt(1, retval);
        ps.setInt(2, 0);
        ps.setInt(3, 0);
        ps.executeUpdate();
        ps.close();

        ps = MainFrm.bookstore_conn.prepareStatement("INSERT INTO category_to_store (category_id,
store_id) VALUES(?,?)");
        ps.setInt(1, retval);
        ps.setInt(2, 0);
        ps.executeUpdate();
        ps.close();

    } catch (SQLException ex) {
        Logger.getLogger(Actions.class.getName()).log(Level.SEVERE, null, ex);
    }

    return retval;
}
//-----
public static String[] Split(String data) {

    String[] retData = data.split("\\\\*\\\\*");
    for (int i=0; i < retData.length; i++){
        retData[i] = retData[i].replaceAll("[*]", "");
    }
    return retData;
}

/*private static void downloadUsingStream(String urlStr, String file) throws MalformedURLException,
IOException {
    URL url = new URL(urlStr);
    try (BufferedInputStream bis = new BufferedInputStream(url.openStream()); FileOutputStream fis =
new FileOutputStream(file)) {
        byte[] buffer = new byte[1024];
        int count=0;
        while((count = bis.read(buffer,0,1024)) != -1)
        {
            fis.write(buffer, 0, count);
        }
    }
}*/

//-----
}

```

## Βιβλιογραφία

- [https://el.wikipedia.org/wiki/Ηλεκτρονικό\\_εμπόριο](https://el.wikipedia.org/wiki/Ηλεκτρονικό_εμπόριο) Ανακτήθηκε στις 26 Ιανουαρίου 2019
- <http://www.sepe.gr/gr/research-studies/article/10104941/sta-4-dis-o-tziros-tou-ilektroniku-eboriou-stin-ellada/> Ανακτήθηκε 27 Ιανουαρίου 2019
- [https://el.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://el.wikipedia.org/wiki/Microsoft_SQL_Server) Ανακτήθηκε 27 Ιανουαρίου 2019
- <https://www.britannica.com/technology/Apache-Web-server> Ανακτήθηκε 27 Ιανουαρίου 2019
- <https://www.oracle.com/technetwork/java/intro-142807.html#943> Ανακτήθηκε 27 Ιανουαρίου 2019
- <https://www.oracle.com/technetwork/java/intro-141325.html#334> Ανακτήθηκε 27 Ιανουαρίου 2019
- [https://www.ip.gr/Hosting/Τι\\_είναι\\_μια\\_βάση\\_MySQL-209.html](https://www.ip.gr/Hosting/Τι_είναι_μια_βάση_MySQL-209.html) Ανακτήθηκε 28 Ιανουαρίου 2019
- [https://www.ip.gr/el/dictionary/390-CMS\\_Content\\_Management\\_System](https://www.ip.gr/el/dictionary/390-CMS_Content_Management_System) Ανακτήθηκε 28 Ιανουαρίου 2019
- Λιόση Χριστίνα (2017), Διαδικασίες και πρακτικές ηλεκτρονικού καταστήματος, Μεταπτυχιακό Πρόγραμμα Σπουδών στη Βιομηχανική Διοίκηση και Τεχνολογία, ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
- <https://www.espa.gr/el/Pages/DictionaryFS.aspx?item=247> Αναρτήθηκε 12 Φεβρουαρίου 2019
- Βουγιουκαλάκης Εμμανουήλ (2017), Ανάπτυξη ιστοσελίδας ηλεκτρονικού καταστήματος με χρήση λογισμικού ανοικτού κώδικα, Πτυχιακή Εργασία, ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ, ΤΜΗΜΑ ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ, ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
- <https://www.synergic.gr/web/synergic-software/-/oi-demophilesteres-ecommerce-platphormes-gia-to-2016> Αναρτήθηκε 18 Φεβρουαρίου 2019