



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Εφαρμογή συντήρησης οχημάτων με την χρήση  
των Blazor και Microsoft SQL Server**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

του

**Χριστόφορου Υφαντίδη**

(ΑΕΜ: 2744)

**Επιβλέπων : Δημήτριος Ι. Βέργαδος**  
**Επίκουρος Καθηγητής**

Καστοριά 04 - 2023

Η παρούσα σελίδα σκοπίμως παραμένει λευκή



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

# Εφαρμογή συντήρησης οχημάτων με την χρήση των Blazor και Microsoft SQL Server

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

**Χριστόφορου Υφαντίδη**

(ΑΕΜ: 2744)

**Επιβλέπων :** Δημήτριος Ι. Βέργαδος  
Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 07/04/2023

.....  
Ον/μο Μέλους  
Ιδιότητα Μέλους

.....  
Ον/μο Μέλους  
Ιδιότητα Μέλους

.....  
Ον/μο Μέλους  
Ιδιότητα Μέλους

Καστοριά 04 - 2023

Copyright © 2023 – ΧΡΙΣΤΟΦΟΡΟΣ ΥΦΑΝΤΙΔΗΣ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

## Περίληψη

---

Από την αρχική κυκλοφορία του .NET Framework πριν από 20 χρόνια, η ανάπτυξη των εφαρμογών για υπολογιστές έχει μειωθεί σημαντικά. Εκείνη την εποχή, οι περισσότερες νέες εφαρμογές αναπτυσσόταν για πρώτη φορά σε Windows περιβάλλον επειδή ήταν μακράν το πιο ευρέως διαδεδομένο λειτουργικό σύστημα. Στη σύγχρονη εποχή, είναι πολύ πιο συνηθισμένο να δημιουργείται μια εφαρμογή ιστού, διότι θα λειτουργεί σε όλα τα λειτουργικά συστήματα. Αυτό έχει ως αποτέλεσμα την στόχευση ενός ευρύτερου πλήθους χρηστών. Παρόλα αυτά, η ανάπτυξη εφαρμογών υπολογιστή θεωρείται μια καλή ιδέα, σε περιπτώσεις όπου για παράδειγμα η απόδοση ή και η ασφάλεια θέτονται ως προτεραιότητα. Σκοπός της πτυχιακής εργασίας αυτής είναι, η ανάλυση και χρησιμότητα της τεχνολογίας .NET Framework ως εργαλείο ανάπτυξης εφαρμογών υπολογιστή, όπως επίσης και όλων των υπολοίπων τεχνολογιών και εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής διαχείρισης δεδομένων.

**Λέξεις Κλειδιά:** *.NET Framework, C#, Windows*

## Abstract

---

Since the initial release of the .NET Framework 20 years ago, desktop application development has been reduced significantly. At the time, most new applications were first developed in a Windows environment because it was by far the most widely used operating system. In modern times, it is much more common to create a web application because it will work on all operating systems. This results in targeting a wider range of users. However, desktop application development is considered a good idea in cases where, for example, performance or security are prioritized. The purpose of this thesis is the analysis and significance of the .NET Framework technology as a desktop application development tool, as well as all the other technologies and tools that were used to develop the data management application.

**Key Words:** *.NET Framework, C#, Windows*

## Πίνακας Περιεχομένων

---

Εισαγωγή.....	1
<b>1. Λειτουργικά συστήματα .....</b>	<b>2</b>
1.1 Ο ρόλος του λειτουργικού συστήματος .....	3
1.2 Από την οπτική γωνία του χρήστη .....	3
1.3 Από την οπτική γωνία του υπολογιστή.....	3
1.4 Ο όρος του λειτουργικού συστήματος.....	5
1.5 Σημαντικότητα μελέτης των λειτουργικών συστημάτων.....	6
1.6 Λειτουργίες και είδη λειτουργικών συστημάτων.....	7
1.6.1 Σύστημα μοναδικών και πολλαπλών εργασιών .....	7
1.6.2 Σύστημα μοναδικών και πολλαπλών χρηστών .....	8
1.6.3 Κατανεμημένο σύστημα.....	8
1.6.4 Ενσωματωμένο σύστημα.....	8
1.6.6 Συστήματα ελεύθερου και ανοικτού κώδικα .....	9
1.6.6.1 Ιστορία του ελεύθερου λογισμικού .....	10
1.6.6.2 Συστήματα ανοικτού κώδικα ως εργαλεία μάθησης.....	11
1.6.7 Διαχείριση πόρων .....	11
1.7 Ιστορική αναδρομή.....	11
1.8 Σύγχρονα λειτουργικά συστήματα .....	12
1.9 Μερίδιο χρήσης λειτουργικών συστημάτων .....	13
1.10 Windows 10.....	13
1.11 Mac OS X.....	14
1.12 Linux.....	14
<b>2. .NET Framework .....</b>	<b>15</b>
2.1 Αρχιτεκτονική .....	16
2.1.1 Κοινή γλωσσική υποδομή .....	16
2.1.2 Κοινό περιβάλλον εκτέλεσης γλωσσών.....	17
2.1.3 Assemblies.....	17
2.1.4 Βιβλιοθήκη κλάσεων .....	18
2.1.5 C++/CLI.....	19
2.2 Σχεδιαστικές αρχές .....	19
2.2.1 Διαλειτουργικότητα.....	19

2.2.2 Γλωσσική ανεξαρτησία .....	20
2.2.3 Ασφάλεια τύπων.....	20
2.2.4 Φορητότητα .....	20
2.2.5 Ασφάλεια .....	21
2.2.6 Διαχείριση μνήμης.....	21
2.2.7 Επιδόσεις.....	22
2.3 Ιστορική αναδρομή.....	23
3. Microsoft Visual Studio .....	24
3.1 Αρχιτεκτονική .....	26
3.2 Χαρακτηριστικά .....	27
3.2.1 Επεξεργαστής κώδικα.....	27
3.2.2 Αποσφραματωτής.....	28
3.2.3 Σχεδιαστής.....	29
3.2.4 Άλλα εργαλεία .....	30
3.3 Επεκτασιμότητα .....	32
3.4 Υποστηριζόμενα προϊόντα .....	33
3.5 Εκδόσεις.....	35
4. Microsoft SQL Server.....	36
4.1 Αρχιτεκτονική .....	36
4.2 Ιστορική αναδρομή.....	37
4.3.1 Διαχείριση απομονωτών .....	38
4.3.2 Συγχρονισμός και κλείδωμα.....	39
4.4 Ανάκτηση δεδομένων και δυνατότητα προγραμματισμού.....	40
4.4.1 T-SQL.....	41
4.4.2 SQL Server Native Client (SNAC).....	42
4.4.3 SQL CLR .....	42
4.5 Υπηρεσίες.....	43
4.5.1 Υπηρεσίες μηχανικής μάθησης.....	43
4.5.2 Διαμεσολαβητής υπηρεσιών.....	44
4.5.3 Υπηρεσίες αντιγραφής .....	44
4.5.4 Υπηρεσίες ανάλυσης.....	45
4.5.5 Υπηρεσίες αναφοράς.....	46
4.5.6 Υπηρεσίες ειδοποίησης .....	46
4.5.7 Υπηρεσίες ενσωμάτωσης .....	46



<b>4.5.8 Υπηρεσία αναζήτησης πλήρους κειμένου</b> .....	47
<b>4.5.9 SQLCMD</b> .....	48
<b>4.5.10 SQL Server Management Studio</b> .....	48
<b>4.5.11 Azure Data Studio</b> .....	49
<b>4.5.12 Business Intelligence Development Studio</b> .....	49
<b>4.6 Εκδόσεις</b> .....	49
<b>5. Εισαγωγή στην C#</b> .....	52
<b>5.1 Στόχοι σχεδιασμού της C#</b> .....	53
<b>5.2 Ιστορική αναδρομή</b> .....	53
<b>5.3 Σύνταξη</b> .....	54
<b>5.4 Χαρακτηριστικά</b> .....	54
<b>5.5 Τύποι δεδομένων C#</b> .....	59
<b>5.6 Βιβλιοθήκες</b> .....	60
<b>6. Εισαγωγή στην SQL</b> .....	60
<b>6.1 Ιστορική αναδρομή</b> .....	61
<b>6.2 Σύνταξη</b> .....	61
<b>6.3 Διαλειτουργικότητα</b> .....	62
<b>6.3.1 Λόγοι ασυμβατότητας</b> .....	63
<b>6.3.2 Τύποι δεδομένων SQL</b> .....	63
<b>7. Σχεδίαση της εφαρμογής</b> .....	65
<b>8. Ανάπτυξη της εφαρμογής</b> .....	67
<b>9. Εκτέλεση της εφαρμογής</b> .....	68
Παράρτημα Κώδικα .....	70
Συμπεράσματα.....	71
Βιβλιογραφία .....	72

## Λίστα Εικόνων

---

Εικόνα 1. Η σχέση μεταξύ χρήστη-εφαρμογής-ΛΣ-υλικού .....	4
Εικόνα 2. Μερίδιο χρήσης λειτουργικών συστημάτων .....	13
Εικόνα 3. Τα τρία δημοφιλέστερα λειτουργικά συστήματα υπολογιστή .....	15
Εικόνα 4. Common Language Runtime (CLR).....	17
Εικόνα 5. Εικόνα του Microsoft Visual Studio .....	25
Εικόνα 6. Εικόνα του Microsoft SQL Server.....	36
Εικόνα 7. Εικόνα της γλώσσας C#.....	53
Εικόνα 8. Παράδειγμα ενός απλού SQL Query.....	62
Εικόνα 9. Use Case Diagram .....	65
Εικόνα 10. Sequence Diagram .....	66
Εικόνα 11. Αρχικό interface της εφαρμογής .....	68
Εικόνα 12. Διαγραφή πελάτη .....	69
Εικόνα 13. Αναζήτηση πελάτη .....	69

## Εισαγωγή

---

Μια εφαρμογή υπολογιστή είναι ένα πρόγραμμα λογισμικού που μπορεί να εκτελεστεί σε έναν αυτόνομο υπολογιστή για να εκτελέσει μια συγκεκριμένη εργασία από έναν τελικό χρήστη. Ορισμένες εφαρμογές υπολογιστή, όπως το πρόγραμμα επεξεργασίας κειμένου, η εφαρμογή επεξεργασίας φωτογραφιών και το πρόγραμμα αναπαραγωγής πολυμέσων σας επιτρέπουν να εκτελείτε διαφορετικές εργασίες, ενώ άλλες, όπως οι εφαρμογές παιχνιδιών, έχουν αναπτυχθεί αποκλειστικά για ψυχαγωγία. Όταν αγοράζετε έναν υπολογιστή ή έναν φορητό υπολογιστή, υπάρχει ένα σύνολο εφαρμογών που είναι ήδη εγκατεστημένες στην επιφάνεια εργασίας σας. Μερικά παραδείγματα ορισμένων εφαρμογών υπολογιστή είναι, η εφαρμογή επεξεργασίας κειμένου Microsoft Word, η εφαρμογή ψυχαγωγίας Ναρκαλιευτής ή η εφαρμογή αναπαραγωγής πολυμέσων Windows Media Player.

Τώρα που γνωρίζετε τι είναι μια εφαρμογή υπολογιστή ας δούμε μερικά από τα βασικά χαρακτηριστικά τους:

- **Ανεξαρτησία συνδεσιμότητας διαδικτύου**  
Η πρόσβαση στο διαδίκτυο δεν είναι απαραίτητη όταν χρησιμοποιείτε μια εφαρμογή υπολογιστή, καθώς είναι εγκατεστημένη στο σύστημα και μπορείτε να έχετε πρόσβαση σε αυτήν ανά πάσα στιγμή.
- **Ιδιωτικότητα και ασφάλεια**  
Οι εφαρμογές υπολογιστή προσφέρουν την μέγιστη ιδιωτικότητα και ασφάλεια καθώς τα δεδομένα αποθηκεύονται τοπικά στον υπολογιστή και όχι σε κάποιον server. Κατά συνέπεια είναι πολύ ασφαλέστερη η χρήση των εφαρμογών υπολογιστή καθώς τα δεδομένα δεν βρίσκονται εκτεθειμένα στο διαδίκτυο.
- **Επιδόσεις**  
Οι εφαρμογές υπολογιστή είναι πιο γρήγορες. Το λογισμικό σε συνδυασμό με έναν ισχυρό υπολογιστή είναι ικανό να εκτελεί πιο σύνθετες εργασίες σε συντομότερο χρόνο από τις εφαρμογές που βασίζονται στο διαδίκτυο.

Όπως είναι εμφανές, τα οφέλη μιας εφαρμογής υπολογιστή είναι πολλά. Ωστόσο, προτού αρχίσει κάποιος την συγγραφή κώδικα μιας εφαρμογής υπολογιστή, θα πρέπει πρώτα να επιλέξει ένα λειτουργικό σύστημα στο οποίο θα προγραμματιστεί η εφαρμογή αυτή. Κατά συνέπεια θα πρέπει να γνωρίζει τι είναι ένα λειτουργικό σύστημα, πως λειτουργεί, ποιες οι λειτουργίες και οι δυνατότητες του.

## 1. Λειτουργικά συστήματα

Το λειτουργικό σύστημα είναι λογισμικό που διαχειρίζεται το υλικό ενός υπολογιστή. Παρέχει επίσης μια βάση για τα προγράμματα εφαρμογών και ενεργεί ως ενδιάμεσος μεταξύ του χρήστη του υπολογιστή και του υλικού του υπολογιστή. Μια καταπληκτική πτυχή των λειτουργικών συστημάτων είναι ο τρόπος με τον οποίο ποικίλλουν ως προς την εκτέλεση αυτών των καθηκόντων σε μια μεγάλη ποικιλία υπολογιστικών περιβαλλόντων. Τα λειτουργικά συστήματα βρίσκονται παντού, από αυτοκίνητα και οικιακές συσκευές που περιλαμβάνουν συσκευές "Internet of Things", μέχρι έξυπνα τηλέφωνα, προσωπικούς υπολογιστές, υπολογιστές επιχειρήσεων και περιβάλλοντα υπολογιστικού νέφους. Προκειμένου να διερευνήσουμε το ρόλο ενός λειτουργικού συστήματος σε ένα σύγχρονο υπολογιστικό περιβάλλον, είναι σημαντικό να κατανοήσουμε πρώτα την οργάνωση και την αρχιτεκτονική του υλικού των υπολογιστών. Αυτό περιλαμβάνει την ΚΜΕ, τη μνήμη και τις συσκευές εισόδου/εξόδου, καθώς και την αποθήκευση. Μια θεμελιώδης ευθύνη ενός λειτουργικού συστήματος είναι να κατανέμει αυτούς τους πόρους στα προγράμματα. Επειδή ένα λειτουργικό σύστημα είναι μεγάλο και πολύπλοκο, πρέπει να δημιουργείται κομμάτι-κομμάτι. Κάθε ένα από αυτά τα κομμάτια πρέπει να είναι ένα καλά οριοθετημένο τμήμα του συστήματος, με προσεκτικά καθορισμένες εισόδους, εξόδους και λειτουργίες. Σε αυτό το κεφάλαιο, παρέχουμε μια γενική επισκόπηση των κύριων στοιχείων ενός σύγχρονου συστήματος υπολογιστή, καθώς και των λειτουργιών που παρέχει το λειτουργικό σύστημα. Επιπλέον, καλύπτουμε διάφορα θέματα που βοηθούν να προετοιμάσουμε το έδαφος για το υπόλοιπο κείμενο: δομές δεδομένων που χρησιμοποιούνται στα λειτουργικά συστήματα, υπολογιστικά περιβάλλοντα και λειτουργικά συστήματα ανοικτού κώδικα και ελεύθερα λειτουργικά συστήματα. Ένα λειτουργικό σύστημα ενεργεί ως ενδιάμεσος μεταξύ του χρήστη ενός υπολογιστή και του υλικού του υπολογιστή. Σκοπός ενός λειτουργικού συστήματος είναι να παρέχει ένα περιβάλλον στο οποίο ο χρήστης μπορεί να εκτελεί προγράμματα με βολικό και αποτελεσματικό τρόπο. Ένα λειτουργικό σύστημα είναι λογισμικό που διαχειρίζεται το υλικό του υπολογιστή. Το υλικό πρέπει να παρέχει κατάλληλους μηχανισμούς για να διασφαλίζει τη σωστή λειτουργία του συστήματος του υπολογιστή και να εμποδίζει τα προγράμματα να παρεμβαίνουν στην ορθή λειτουργία του συστήματος. Εσωτερικά, τα λειτουργικά συστήματα διαφέρουν πολύ ως προς τη σύνθεσή τους, δεδομένου ότι είναι οργανωμένα κατά μήκος πολλών διαφορετικών γραμμών. Ο σχεδιασμός ενός νέου λειτουργικού συστήματος είναι ένα σημαντικό έργο, και είναι σημαντικό οι στόχοι του συστήματος να είναι καλά καθορισμένοι πριν από την έναρξη του σχεδιασμού. Επειδή ένα λειτουργικό σύστημα είναι μεγάλο και πολύπλοκο, πρέπει να δημιουργηθεί κομμάτι-κομμάτι. Κάθε ένα από αυτά τα κομμάτια πρέπει να αποτελεί ένα καλά οριοθετημένο τμήμα του συστήματος, με προσεκτικά καθορισμένες εισόδους, εξόδους και λειτουργίες.

## 1.1 Ο ρόλος του λειτουργικού συστήματος

Ξεκινάμε τη συζήτησή μας εξετάζοντας το ρόλο του λειτουργικού συστήματος στο συνολικό σύστημα υπολογιστή. Ένα υπολογιστικό σύστημα μπορεί να χωριστεί χονδρικά σε τέσσερα στοιχεία: το υλικό, το λειτουργικό σύστημα, τα προγράμματα εφαρμογών και ένας χρήστης. Το υλικό - η κεντρική μονάδα επεξεργασίας (CPU), η μνήμη και οι συσκευές εισόδου/εξόδου (I/O) - παρέχει τους βασικούς υπολογιστικούς πόρους για το σύστημα. Τα προγράμματα εφαρμογών -όπως οι επεξεργαστές κειμένου, τα λογιστικά φύλλα, οι μεταγλωττιστές και οι φυλλομετρητές ιστού- καθορίζουν τους τρόπους με τους οποίους χρησιμοποιούνται αυτοί οι πόροι για την επίλυση των υπολογιστικών προβλημάτων των χρηστών. Το λειτουργικό σύστημα ελέγχει το υλικό και συντονίζει τη χρήση του από τα διάφορα προγράμματα εφαρμογών για τους διάφορους χρήστες. Μπορούμε επίσης να θεωρήσουμε ότι ένα υπολογιστικό σύστημα αποτελείται από υλικό, λογισμικό και δεδομένα. Το λειτουργικό σύστημα παρέχει τα μέσα για τη σωστή χρήση αυτών των πόρων κατά τη λειτουργία του συστήματος υπολογιστών. Ένα λειτουργικό σύστημα είναι παρόμοιο με μια κυβέρνηση. Όπως μια κυβέρνηση, δεν επιτελεί καμία χρήσιμη λειτουργία από μόνο του. Παρέχει απλώς ένα περιβάλλον μέσα στο οποίο άλλα προγράμματα μπορούν να κάνουν χρήσιμη εργασία. Για να κατανοήσουμε πληρέστερα το ρόλο του λειτουργικού συστήματος, θα εξερευνήσουμε στη συνέχεια τα λειτουργικά συστήματα από δύο οπτικές γωνίες: αυτή του χρήστη και αυτή του συστήματος.

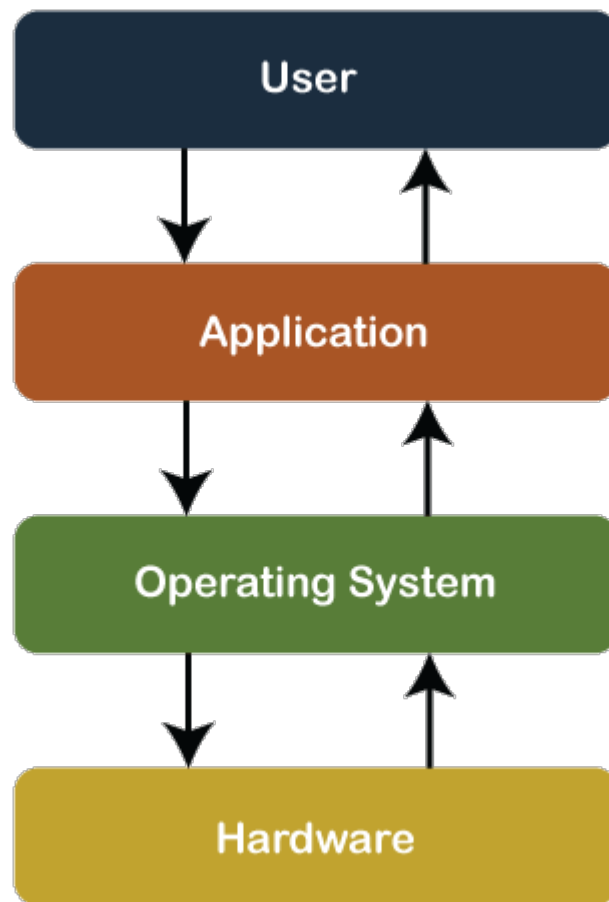
## 1.2 Από την οπτική γωνία του χρήστη

Η χρήση του λειτουργικού που κάνει ο χρήστης εξαρτάται από τη διεπαφή του συστήματος που χρησιμοποιείται. Ορισμένα συστήματα έχουν σχεδιαστεί ώστε ένας μόνο χρήστης να χρησιμοποιεί πλήρως τους πόρους για να μεγιστοποιήσει την εργασία του. Σε αυτές τις περιπτώσεις, το ΛΣ έχει σχεδιαστεί κυρίως για ευκολία στη χρήση, με μικρή έμφαση στην ποιότητα και καμία στη χρήση πόρων. Από την οπτική γωνία του χρήστη, ο χρήστης εστιάζει στον τρόπο με τον οποίο αλληλεπιδρά με το λειτουργικό σύστημα μέσω της χρήσης διαφόρων προγραμμάτων εφαρμογών. Αντιθέτως, από την οπτική σκοπιά του συστήματος, το λειτουργικό σύστημα εστιάζει στον τρόπο με τον οποίο το υλικό αλληλεπιδρά με αυτό για την ολοκλήρωση διαφόρων εργασιών.

## 1.3 Από την οπτική γωνία του υπολογιστή

Από τη σκοπιά του υπολογιστή, το λειτουργικό σύστημα είναι το πρόγραμμα που εμπλέκεται πιο στενά με το υλικό. Σε αυτό το πλαίσιο, μπορούμε να θεωρήσουμε ένα λειτουργικό σύστημα ως έναν καταναεμητή πόρων. Ένα υπολογιστικό σύστημα διαθέτει πολλούς πόρους που μπορεί να απαιτούνται για την επίλυση ενός προβλήματος: χρόνος CPU, χώρος μνήμης, χώρος αποθήκευσης, συσκευές εισόδου/εξόδου κ.ο.κ. Το λειτουργικό σύστημα ενεργεί ως διαχειριστής αυτών των πόρων. Αντιμετωπίζοντας πολυάριθμες και ενδεχομένως αντικρουόμενες αιτήσεις για πόρους, το λειτουργικό

σύστημα πρέπει να αποφασίσει πώς θα τους καταναίμει σε συγκεκριμένα προγράμματα και χρήστες, ώστε να μπορεί να λειτουργήσει το σύστημα του υπολογιστή αποτελεσματικά και δίκαια. Μια ελαφρώς διαφορετική θεώρηση ενός λειτουργικού συστήματος δίνει έμφαση στην ανάγκη ελέγχου των διαφόρων συσκευών εισόδου/εξόδου και των προγραμμάτων χρηστών. Ένα λειτουργικό σύστημα είναι ένα πρόγραμμα ελέγχου. Ένα πρόγραμμα ελέγχου διαχειρίζεται την εκτέλεση των προγραμμάτων χρήστη για την αποφυγή σφαλμάτων και ακατάλληλης χρήσης του υπολογιστή. Ασχολείται ιδιαίτερα με τη λειτουργία και τον έλεγχο των συσκευών εισόδου/εξόδου.



Εικόνα 1. Η σχέση μεταξύ χρήστη-εφαρμογής-ΛΣ-υλικού

Πηγή: <https://www.javatpoint.com/history-of-operating-system>

## 1.4 Ο όρος του λειτουργικού συστήματος

Ο όρος λειτουργικό σύστημα καλύπτει πολλούς ρόλους και λειτουργίες. Αυτό συμβαίνει, τουλάχιστον εν μέρει, λόγω των αναρίθμητων σχεδίων και χρήσεων των υπολογιστών. Υπολογιστές υπάρχουν μέσα σε τoστιέρες, αυτοκίνητα, πλοία, διαστημόπλοια, σπίτια και επιχειρήσεις. Αποτελούν τη βάση για παιχνιδιομηχανές, δέκτες καλωδιακής τηλεόρασης και βιομηχανικά συστήματα ελέγχου. Για να εξηγήσουμε αυτή την ποικιλομορφία, μπορούμε να στραφούμε στην ιστορία των υπολογιστών. Αν και οι υπολογιστές έχουν σχετικά σύντομη ιστορία, έχουν εξελιχθεί ραγδαία. Η πληροφορική ξεκίνησε ως πείραμα για να διαπιστωθεί τι μπορεί να γίνει και γρήγορα μετακινήθηκε σε συστήματα σταθερού σκοπού για στρατιωτικές χρήσεις, όπως το σπάσιμο κωδικών και η χάραξη τροχιών, και σε κυβερνητικές χρήσεις, όπως ο υπολογισμός απογραφών. Αυτοί οι πρώτοι υπολογιστές εξελίχθηκαν σε κεντρικούς υπολογιστές γενικής χρήσης, πολλαπλών λειτουργιών, και τότε γεννήθηκαν τα λειτουργικά συστήματα. Στη δεκαετία του 1960, ο νόμος του Moore προέβλεπε ότι ο αριθμός των τρανζίστορ σε ένα ολοκληρωμένο κύκλωμα θα διπλασιαζόταν κάθε 18 μήνες, και η πρόβλεψη αυτή επαληθεύτηκε. Οι υπολογιστές απέκτησαν περισσότερες λειτουργίες και συρρικνώθηκαν σε μέγεθος, με αποτέλεσμα να υπάρχει ένας τεράστιος αριθμός χρήσεων και ένας τεράστιος αριθμός και ποικιλία λειτουργικών συστημάτων. Πώς, λοιπόν, μπορούμε να ορίσουμε τι είναι ένα λειτουργικό σύστημα; Σε γενικές γραμμές, δεν έχουμε έναν απολύτως επαρκή ορισμό ενός λειτουργικού συστήματος. Τα λειτουργικά συστήματα υπάρχουν επειδή προσφέρουν έναν λογικό τρόπο επίλυσης του προβλήματος της δημιουργίας ενός εύχρηστου υπολογιστικού συστήματος. Ο θεμελιώδης στόχος των υπολογιστικών συστημάτων είναι να εκτελούν προγράμματα και να διευκολύνουν την επίλυση προβλημάτων των χρηστών. Το υλικό των υπολογιστών κατασκευάζεται προς την κατεύθυνση αυτού του στόχου. Δεδομένου ότι το γυμνό υλικό από μόνο του δεν είναι ιδιαίτερα εύκολο στη χρήση, αναπτύσσονται προγράμματα εφαρμογών. Τα προγράμματα αυτά απαιτούν ορισμένες κοινές λειτουργίες, όπως αυτές που ελέγχουν τις συσκευές εισόδου/εξόδου. Οι κοινές λειτουργίες του ελέγχου και της κατανομής των πόρων συγκεντρώνονται στη συνέχεια σε ένα κομμάτι λογισμικού: το λειτουργικό σύστημα. Επιπλέον, δεν έχουμε έναν καθολικά αποδεκτό ορισμό για το τι αποτελεί μέρος του λειτουργικού συστήματος. Μια απλή άποψη είναι ότι περιλαμβάνει όλα όσα ένας πωλητής παραδίδει όταν παραγγέλνετε "το λειτουργικό σύστημα". Τα χαρακτηριστικά που περιλαμβάνονται, ωστόσο, διαφέρουν σημαντικά μεταξύ των συστημάτων. Ορισμένα συστήματα καταλαμβάνουν λιγότερο από ένα megabyte χώρου και δεν διαθέτουν ούτε καν έναν επεξεργαστή πλήρους οθόνης, ενώ άλλα απαιτούν gigabytes χώρου και βασίζονται εξ ολοκλήρου σε γραφικά συστήματα παραθύρων. Ένας πιο συνηθισμένος ορισμός, και αυτός που συνήθως ακολουθούμε, είναι ότι το λειτουργικό σύστημα είναι το μοναδικό πρόγραμμα που εκτελείται ανά πάσα στιγμή στον υπολογιστή - συνήθως ονομάζεται πυρήνας. Μαζί με τον πυρήνα, υπάρχουν δύο άλλοι τύποι προγραμμάτων: τα προγράμματα συστήματος, τα οποία σχετίζονται με το λειτουργικό σύστημα αλλά δεν αποτελούν απαραίτητα μέρος του πυρήνα, και τα προγράμματα εφαρμογών, τα οποία περιλαμβάνουν όλα τα προγράμματα που δεν σχετίζονται με τη λειτουργία του συστήματος. Το θέμα του τι συνιστά ένα λειτουργικό σύστημα έγινε όλο και πιο σημαντικό καθώς οι προσωπικοί υπολογιστές διαδόθηκαν όλο

και περισσότερο και τα λειτουργικά συστήματα έγιναν όλο και πιο εξελιγμένα. Το 1998, το Υπουργείο Δικαιοσύνης των Ηνωμένων Πολιτειών κατέθεσε αγωγή κατά της Microsoft, ισχυριζόμενο ουσιαστικά ότι η Microsoft συμπεριέλαβε υπερβολικά πολλές λειτουργίες στα λειτουργικά της συστήματα και έτσι εμπόδιζε τους προμηθευτές εφαρμογών να ανταγωνίζονται. (Για παράδειγμα, ένα πρόγραμμα περιήγησης στο διαδίκτυο αποτελούσε αναπόσπαστο μέρος των λειτουργικών συστημάτων της Microsoft). Ως αποτέλεσμα, η Microsoft κρίθηκε ένοχη για τη χρήση του μονοπωλίου της στα λειτουργικά συστήματα για τον περιορισμό του ανταγωνισμού. Σήμερα, ωστόσο, αν εξετάσουμε τα λειτουργικά συστήματα για κινητές συσκευές, βλέπουμε ότι και πάλι ο αριθμός των χαρακτηριστικών που αποτελούν το λειτουργικό σύστημα αυξάνεται. Τα κινητά λειτουργικά συστήματα συχνά περιλαμβάνουν όχι μόνο έναν πυρήνα αλλά και ενδιάμεσο λογισμικό - ένα σύνολο πλαισίων λογισμικού που παρέχουν πρόσθετες υπηρεσίες στους προγραμματιστές εφαρμογών. Για παράδειγμα, καθένα από τα δύο πιο γνωστά λειτουργικά συστήματα κινητών τηλεφώνων -το iOS της Apple και το Android της Google- διαθέτει έναν πυρήνα μαζί με ενδιάμεσο λογισμικό που υποστηρίζει βάσεις δεδομένων, πολυμέσα και γραφικά (για να αναφέρουμε μόνο μερικά). Συνοπτικά, για τους σκοπούς μας, το λειτουργικό σύστημα περιλαμβάνει τον πυρήνα που εκτελείται πάντα, πλαίσια ενδιάμεσου λογισμικού που διευκολύνουν την ανάπτυξη εφαρμογών και παρέχουν δυνατότητες, και προγράμματα συστήματος που βοηθούν στη διαχείριση του συστήματος κατά τη διάρκεια της λειτουργίας του. Το μεγαλύτερο μέρος αυτού του κειμένου ασχολείται με τον πυρήνα των λειτουργικών συστημάτων γενικής χρήσης, αλλά εξετάζονται και άλλα συστατικά στοιχεία όπως απαιτείται για την πλήρη επεξήγηση του σχεδιασμού και της λειτουργίας του λειτουργικού συστήματος.

## 1.5 Σημαντικότητα μελέτης των λειτουργικών συστημάτων

Αν και υπάρχουν πολλοί επαγγελματίες της επιστήμης των υπολογιστών, μόνο ένα μικρό ποσοστό από αυτούς θα ασχοληθεί με τη δημιουργία ή την τροποποίηση ενός λειτουργικού συστήματος. Γιατί, λοιπόν, να μελετήσετε τα λειτουργικά συστήματα και τον τρόπο λειτουργίας τους; Απλούστατα επειδή, καθώς σχεδόν όλος ο κώδικας εκτελείται πάνω σε ένα λειτουργικό σύστημα, η γνώση του τρόπου λειτουργίας των λειτουργικών συστημάτων είναι ζωτικής σημασίας για τον σωστό, αποδοτικό, αποτελεσματικό και ασφαλή προγραμματισμό. Η κατανόηση των βασικών αρχών των λειτουργικών συστημάτων, του τρόπου με τον οποίο οδηγούν το υλικό των υπολογιστών και του τι παρέχουν στις εφαρμογές δεν είναι μόνο απαραίτητη για όσους τα προγραμματίζουν, αλλά και εξαιρετικά χρήσιμη για όσους γράφουν προγράμματα σε αυτά και τα χρησιμοποιούν.



## 1.6 Λειτουργίες και είδη λειτουργικών συστημάτων

Τώρα που συζητήσαμε βασικές πληροφορίες σχετικά με την οργάνωση και την αρχιτεκτονική των συστημάτων υπολογιστών, είμαστε έτοιμοι να μιλήσουμε για τα λειτουργικά συστήματα. Ένα λειτουργικό σύστημα παρέχει το περιβάλλον μέσα στο οποίο εκτελούνται τα προγράμματα. Εσωτερικά, τα λειτουργικά συστήματα ποικίλλουν σε μεγάλο βαθμό, καθώς είναι οργανωμένα κατά μήκος πολλών διαφορετικών γραμμών. Υπάρχουν, ωστόσο, πολλά κοινά σημεία, τα οποία εξετάζουμε σε αυτή την ενότητα. Για να αρχίσει να λειτουργεί ένας υπολογιστής -για παράδειγμα, όταν ενεργοποιείται ή επανεκκινείται- πρέπει να έχει ένα αρχικό πρόγραμμα για να εκτελεστεί. Όπως σημειώθηκε προηγουμένως, αυτό το αρχικό πρόγραμμα, ή πρόγραμμα εκκίνησης, τείνει να είναι απλό. Συνήθως, είναι αποθηκευμένο στο υλικό του υπολογιστή σε υλικολογισμικό. Αρχικοποιεί όλες τις πτυχές του συστήματος, από τους καταχωρητές της CPU έως τους ελεγκτές συσκευών και τα περιεχόμενα της μνήμης. Το πρόγραμμα bootstrap πρέπει να γνωρίζει πώς να φορτώσει το λειτουργικό σύστημα και πώς να αρχίσει να εκτελεί αυτό το σύστημα. Για να επιτευχθεί αυτός ο στόχος, το πρόγραμμα bootstrap πρέπει να εντοπίσει τον πυρήνα του λειτουργικού συστήματος και να τον φορτώσει στη μνήμη. Μόλις ο πυρήνας φορτωθεί και εκτελεστεί, μπορεί να αρχίσει να παρέχει υπηρεσίες στο σύστημα και στους χρήστες του. Ορισμένες υπηρεσίες παρέχονται εκτός του πυρήνα από προγράμματα του συστήματος που φορτώνονται στη μνήμη κατά την εκκίνηση.

### 1.6.1 Σύστημα μοναδικών και πολλαπλών εργασιών

Ένα σύστημα μίας εργασίας μπορεί να εκτελέσει μόνο ένα πρόγραμμα κάθε φορά, ενώ ένα λειτουργικό σύστημα πολλαπλών εργασιών επιτρέπει την ταυτόχρονη εκτέλεση περισσότερων από ένα προγραμμάτων. Αυτό επιτυγχάνεται με την κοινή χρήση χρόνου, όπου ο διαθέσιμος χρόνος του επεξεργαστή διαιρείται μεταξύ πολλαπλών διεργασιών. Αυτές οι διεργασίες διακόπτονται κατ' επανάληψη σε χρονικά τμήματα από ένα υποσύστημα προγραμματισμού εργασιών του λειτουργικού συστήματος. Το multi-tasking μπορεί να χαρακτηριστεί σε προληπτικούς και συνεργατικούς τύπους. Στο προληπτικό multitasking, το λειτουργικό σύστημα κόβει το χρόνο της CPU και αφιερώνει μια υποδοχή σε καθένα από τα προγράμματα. Λειτουργικά συστήματα τύπου Unix, όπως το Linux—καθώς και μη Unix, όπως το AmigaOS—υποστηρίζουν προληπτικές πολλαπλές εργασίες. Η συνεργασία πολλαπλών εργασιών επιτυγχάνεται βασιζόμενη σε κάθε διαδικασία για να παρέχει χρόνο στις άλλες διεργασίες με καθορισμένο τρόπο. Οι εκδόσεις 16-bit των Microsoft Windows χρησιμοποιούσαν συνεργατική πολλαπλή εργασία. Οι εκδόσεις 32 bit των Windows NT και Win9x χρησιμοποιούσαν προληπτικές πολλαπλές εργασίες.

## 1.6.2 Σύστημα μοναδικών και πολλαπλών χρηστών

Τα λειτουργικά συστήματα ενός χρήστη δεν διαθέτουν ευκολίες για να διακρίνουν τους χρήστες, αλλά μπορεί να επιτρέπουν σε πολλά προγράμματα να εκτελούνται ταυτόχρονα. Ένα λειτουργικό σύστημα πολλών χρηστών επεκτείνει τη βασική έννοια της πολλαπλής εργασίας με εγκαταστάσεις που προσδιορίζουν διεργασίες και πόρους, όπως ο χώρος στο δίσκο, που ανήκει σε πολλούς χρήστες, και το σύστημα επιτρέπει σε πολλούς χρήστες να αλληλεπιδρούν με το σύστημα ταυτόχρονα. Τα λειτουργικά συστήματα χρονομερισμού προγραμματίζουν εργασίες για αποτελεσματική χρήση του συστήματος και μπορεί επίσης να περιλαμβάνουν λογιστικό λογισμικό για την κατανομή κόστους του χρόνου του επεξεργαστή, τη μαζική αποθήκευση, την εκτύπωση και άλλους πόρους σε πολλούς χρήστες.

## 1.6.3 Κατανεμημένο σύστημα

Ένα κατανεμημένο λειτουργικό σύστημα είναι λογισμικό συστήματος πάνω από μια συλλογή ανεξάρτητου λογισμικού, δικτυωμένων, επικοινωνούντων και φυσικώς χωριστών υπολογιστικών κόμβων. Διαχειρίζονται εργασίες που εξυπηρετούνται από πολλαπλές CPU. Κάθε μεμονωμένος κόμβος διαθέτει ένα συγκεκριμένο υποσύνολο λογισμικού του παγκόσμιου συγκεντρωτικού λειτουργικού συστήματος. Κάθε υποσύνολο είναι ένα σύνθετο από δύο διαφορετικούς παρόχους υπηρεσιών. Ο πρώτος είναι ένας πανταχού παρόν ελάχιστος πυρήνας, ή μικροπυρήνας, που ελέγχει άμεσα το υλικό αυτού του κόμβου. Το δεύτερο είναι μια συλλογή υψηλότερου επιπέδου στοιχείων διαχείρισης συστήματος που συντονίζουν τις ατομικές και συνεργατικές δραστηριότητες του κόμβου. Αυτά τα στοιχεία αφαιρούν λειτουργίες μικροπυρήνα και υποστηρίζουν εφαρμογές χρήστη.

## 1.6.4 Ενσωματωμένο σύστημα

Ένα ενσωματωμένο λειτουργικό σύστημα είναι ένα λειτουργικό σύστημα για ενσωματωμένα συστήματα υπολογιστών. Τα ενσωματωμένα λειτουργικά συστήματα είναι συστήματα υπολογιστών που έχουν σχεδιαστεί για να αυξάνουν τη λειτουργικότητα και την αξιοπιστία για την επίτευξη μιας συγκεκριμένης εργασίας. Η αποδοτικότητα των πόρων έχει το κόστος της απώλειας κάποιας λειτουργικότητας ή ευκρίνειας που παρέχουν τα μεγαλύτερα λειτουργικά συστήματα υπολογιστών, συμπεριλαμβανομένων των λειτουργιών που ενδέχεται να μην χρησιμοποιούνται από τις εξειδικευμένες εφαρμογές που εκτελούνται. Ανάλογα με τη μέθοδο που χρησιμοποιείται για την εκτέλεση πολλαπλών εργασιών, αυτός ο τύπος λειτουργικού συστήματος θεωρείται συχνά λειτουργικό σύστημα σε πραγματικό χρόνο ή RTOS. Τα ενσωματωμένα συστήματα χρησιμοποιούνται κυρίως ως λειτουργικά συστήματα σε πραγματικό χρόνο.

### 1.6.5 Σύστημα σε πραγματικό χρόνο

Ένα λειτουργικό σύστημα σε πραγματικό χρόνο (RTOS) είναι ένα λειτουργικό σύστημα για εφαρμογές σε πραγματικό χρόνο που επεξεργάζεται δεδομένα και συμβάντα που έχουν κριτικά καθορισμένους χρονικούς περιορισμούς. Ένα RTOS διαφέρει από ένα λειτουργικό σύστημα κοινής χρήσης χρόνου, όπως το Unix, το οποίο διαχειρίζεται την κοινή χρήση των πόρων του συστήματος με έναν προγραμματιστή, buffer δεδομένων ή καθορισμένη ιεράρχηση εργασιών σε περιβάλλον πολλαπλών εργασιών ή πολυπρογραμματισμού. Οι απαιτήσεις χρόνου επεξεργασίας πρέπει να είναι πλήρως κατανοητές και να δεσμεύονται και όχι απλώς να διατηρούνται στο ελάχιστο. Όλη η επεξεργασία πρέπει να γίνεται εντός των καθορισμένων περιορισμών. Τα λειτουργικά συστήματα σε πραγματικό χρόνο βασίζονται σε συμβάντα και προληπτικά, πράγμα που σημαίνει ότι το λειτουργικό σύστημα είναι ικανό να παρακολουθεί τη σχετική προτεραιότητα των ανταγωνιστικών εργασιών και να κάνει αλλαγές στην προτεραιότητα εργασίας. Τα συστήματα που βασίζονται σε συμβάντα αλλάζουν μεταξύ εργασιών με βάση τις προτεραιότητές τους, ενώ τα συστήματα χρονομερισμού αλλάζουν την εργασία με βάση τις διακοπές ρολογιού.

### 1.6.6 Συστήματα ελεύθερου και ανοικτού κώδικα

Η μελέτη των λειτουργικών συστημάτων έχει γίνει ευκολότερη χάρη στη διαθεσιμότητα ενός τεράστιου αριθμού εκδόσεων ελεύθερου λογισμικού και ανοικτού κώδικα. Τόσο τα ελεύθερα λειτουργικά συστήματα όσο και τα λειτουργικά συστήματα ανοικτού κώδικα είναι διαθέσιμα σε μορφή πηγαίου κώδικα και όχι ως μεταγλωττισμένος δυαδικός κώδικας. Σημειώστε, ωστόσο, ότι το ελεύθερο λογισμικό και το λογισμικό ανοικτού κώδικα είναι δύο διαφορετικές ιδέες που υποστηρίζονται από διαφορετικές ομάδες ανθρώπων. Το ελεύθερο λογισμικό (που μερικές φορές αναφέρεται ως ελεύθερο/ελεύθερο λογισμικό) όχι μόνο καθιστά διαθέσιμο τον πηγαίο κώδικα αλλά και αδειοδοτείται ώστε να επιτρέπει τη χρήση, την αναδιανομή και την τροποποίηση χωρίς κόστος. Το λογισμικό ανοικτού κώδικα δεν προσφέρει απαραίτητα τέτοια αδειοδότηση. Έτσι, αν και όλο το ελεύθερο λογισμικό είναι ανοικτού κώδικα, κάποιο λογισμικό ανοικτού κώδικα δεν είναι "ελεύθερο". Το GNU/Linux είναι το πιο διάσημο λειτουργικό σύστημα ανοικτού κώδικα, με ορισμένες διανομές να είναι ελεύθερες και άλλες μόνο ανοικτού κώδικα (<http://www.gnu.org/distros/>). Τα Microsoft Windows είναι ένα γνωστό παράδειγμα της αντίθετης προσέγγισης κλειστού κώδικα. Τα Windows είναι ιδιόκτητο λογισμικό - η Microsoft τα κατέχει, περιορίζει τη χρήση τους και προστατεύει προσεκτικά τον πηγαίο κώδικά τους. Το λειτουργικό σύστημα macOS της Apple περιλαμβάνει μια υβριδική προσέγγιση. Περιέχει έναν πυρήνα ανοικτού κώδικα με την ονομασία Darwin, αλλά περιλαμβάνει και ιδιόκτητα στοιχεία κλειστού κώδικα. Ξεκινώντας με τον πηγαίο κώδικα, ο προγραμματιστής μπορεί να παράγει δυαδικό κώδικα που μπορεί να εκτελεστεί σε ένα σύστημα. Το αντίθετο - η αντίστροφη μηχανική του πηγαίου κώδικα από τα δυαδικά αρχεία - είναι αρκετή δουλειά και χρήσιμα στοιχεία, όπως τα σχόλια, δεν ανακτώνται ποτέ. Η εκμάθηση λειτουργικών συστημάτων με την

εξέταση του πηγαίου κώδικα έχει και άλλα οφέλη. Με τον πηγαίο κώδικα στα χέρια του, ένας μαθητής μπορεί να τροποποιήσει το λειτουργικό σύστημα και στη συνέχεια να μεταγλωττίσει και να εκτελέσει τον κώδικα για να δοκιμάσει αυτές τις αλλαγές, πράγμα που αποτελεί ένα εξαιρετικό εργαλείο μάθησης. Το παρόν κείμενο περιλαμβάνει έργα που περιλαμβάνουν την τροποποίηση του πηγαίου κώδικα λειτουργικών συστημάτων, ενώ παράλληλα περιγράφει αλγορίθμους σε υψηλό επίπεδο, ώστε να είναι βέβαιο ότι καλύπτονται όλα τα σημαντικά θέματα λειτουργικών συστημάτων. Σε όλο το κείμενο, παρέχουμε δείκτες σε παραδείγματα κώδικα ανοικτού κώδικα για βαθύτερη μελέτη. Τα λειτουργικά συστήματα ανοικτού κώδικα έχουν πολλά πλεονεκτήματα, όπως μια κοινότητα ενδιαφερόμενων (και συνήθως μη αμειβόμενων) προγραμματιστών που συμβάλλουν στον κώδικα βοηθώντας στη συγγραφή του, στην αποσφαλμάτωσή του, στην ανάλυσή του, στην παροχή υποστήριξης και στην υποβολή προτάσεων για αλλαγές. Αναμφισβήτητα, ο κώδικας ανοικτού κώδικα είναι πιο ασφαλής από τον κώδικα κλειστού κώδικα επειδή τον βλέπουν πολύ περισσότερα μάτια. Σίγουρα, ο κώδικας ανοικτού κώδικα έχει σφάλματα, αλλά οι υποστηρικτές του ανοικτού κώδικα υποστηρίζουν ότι τα σφάλματα τείνουν να εντοπίζονται και να διορθώνονται ταχύτερα λόγω του αριθμού των ανθρώπων που χρησιμοποιούν και βλέπουν τον κώδικα. Οι εταιρείες που κερδίζουν έσοδα από την πώληση των προγραμμάτων τους συχνά διστάζουν να ανοίξουν τον κώδικά τους, αλλά η Red Hat και μυριάδες άλλες εταιρείες κάνουν ακριβώς αυτό και δείχνουν ότι οι εμπορικές εταιρείες επωφελούνται, αντί να υποφέρουν, όταν ανοίγουν τον κώδικά τους. Τα έσοδα μπορούν να προκύψουν από συμβόλαια υποστήριξης και από την πώληση υλικού στο οποίο τρέχει το λογισμικό.

### 1.6.6.1 Ιστορία του ελεύθερου λογισμικού

Για να αντιμετωπίσει την κίνηση να περιοριστεί η χρήση και η αναδιανομή του λογισμικού, ο Richard Stallman το 1984 άρχισε να αναπτύσσει ένα ελεύθερο, συμβατό με το UNIX λειτουργικό σύστημα με την ονομασία GNU (το οποίο είναι ένα αναδρομικό ακρωνύμιο για το "GNU's Not Unix!"). Για τον Stallman, το "ελεύθερο" αναφέρεται στην ελευθερία χρήσης, όχι στην τιμή. Το κίνημα του ελεύθερου λογισμικού δεν αντιτίθεται στην ανταλλαγή ενός αντιγράφου με ένα χρηματικό ποσό, αλλά θεωρεί ότι οι χρήστες δικαιούνται τέσσερις συγκεκριμένες ελευθερίες: (1) να εκτελούν ελεύθερα το πρόγραμμα, (2) να μελετούν και να αλλάζουν τον πηγαίο κώδικα και να δίνουν ή να πωλούν αντίγραφα είτε (3) με είτε (4) χωρίς αλλαγές. Το 1985, ο Stallman δημοσίευσε το Μανιφέστο GNU, το οποίο υποστηρίζει ότι όλο το λογισμικό πρέπει να είναι ελεύθερο. Δημιούργησε επίσης το Ίδρυμα Ελεύθερου Λογισμικού (FSF) με στόχο την ενθάρρυνση της χρήσης και της ανάπτυξης του ελεύθερου λογισμικού. Το FSF χρησιμοποιεί τα πνευματικά δικαιώματα στα προγράμματά του για να εφαρμόσει το "copyleft", μια μορφή αδειοδότησης που επινόησε ο Stallman. Το copylefting ενός έργου δίνει σε οποιονδήποτε κατέχει ένα αντίγραφο του έργου τις τέσσερις βασικές ελευθερίες που καθιστούν το έργο ελεύθερο, με την προϋπόθεση ότι η αναδιανομή πρέπει να διατηρεί αυτές τις ελευθερίες. Η Γενική Άδεια Δημόσιας Χρήσης GNU (GPL) είναι μια κοινή άδεια με την οποία κυκλοφορεί το ελεύθερο λογισμικό. Βασικά, η GPL απαιτεί να διανέμεται ο πηγαίος

κώδικας μαζί με οποιαδήποτε δυαδικά αρχεία και όλα τα αντίγραφα (συμπεριλαμβανομένων των τροποποιημένων εκδόσεων) να κυκλοφορούν με την ίδια άδεια GPL. Η άδεια Creative Commons "Attribution Sharealike" είναι επίσης μια άδεια copyleft- το "sharealike" είναι ένας άλλος τρόπος να δηλωθεί η ιδέα του copyleft.

### **1.6.6.2 Συστήματα ανοικτού κώδικα ως εργαλεία μάθησης**

Το κίνημα του ελεύθερου λογισμικού ωθεί λεγεώνες προγραμματιστών να δημιουργήσουν χιλιάδες έργα ανοικτού κώδικα, συμπεριλαμβανομένων των λειτουργικών συστημάτων. Ιστοσελίδες όπως οι <http://freshmeat.net/> και <http://distrowatch.com/> παρέχουν πύλες για πολλά από αυτά τα έργα. Όπως αναφέραμε προηγουμένως, τα έργα ανοικτού κώδικα επιτρέπουν στους μαθητές να χρησιμοποιούν τον πηγαίο κώδικα ως εργαλείο μάθησης. Μπορούν να τροποποιήσουν προγράμματα και να τα δοκιμάσουν, να βοηθήσουν στην εύρεση και διόρθωση σφαλμάτων και να εξερευνήσουν με άλλο τρόπο ώριμα, πλήρως εξοπλισμένα λειτουργικά συστήματα, μεταγλωττιστές, εργαλεία, διεπαφές χρήστη και άλλους τύπους προγραμμάτων. Η διαθεσιμότητα του πηγαίου κώδικα για ιστορικά έργα, όπως το Multics, μπορεί να βοηθήσει τους μαθητές να κατανοήσουν αυτά τα έργα και να αποκτήσουν γνώσεις που θα βοηθήσουν στην υλοποίηση νέων έργων. Ένα άλλο πλεονέκτημα της εργασίας με λειτουργικά συστήματα ανοικτού κώδικα είναι η ποικιλομορφία τους. Το GNU/Linux και το BSD UNIX είναι και τα δύο λειτουργικά συστήματα ανοικτού κώδικα, για παράδειγμα, αλλά το καθένα έχει τους δικούς του στόχους, χρησιμότητα, αδειοδότηση και σκοπό. Μερικές φορές, οι άδειες δεν αποκλείουν η μία την άλλη και παρατηρείται διασταύρωση, επιτρέποντας ταχείες βελτιώσεις σε έργα λειτουργικών συστημάτων. Για παράδειγμα, πολλά σημαντικά στοιχεία του OpenSolaris έχουν μεταφερθεί στο BSD UNIX. Τα πλεονεκτήματα του ελεύθερου λογισμικού και του ανοικτού κώδικα είναι πιθανό να αυξήσουν τον αριθμό και την ποιότητα των έργων ανοικτού κώδικα, οδηγώντας σε αύξηση του αριθμού των ατόμων και των εταιρειών που χρησιμοποιούν αυτά τα έργα.

### **1.6.7 Διαχείριση πόρων**

Ένα λειτουργικό σύστημα είναι ένας διαχειριστής πόρων. Η CPU του συστήματος, ο χώρος μνήμης, ο χώρος αποθήκευσης αρχείων και οι συσκευές εισόδου/εξόδου είναι μεταξύ των πόρων που πρέπει να διαχειρίζεται το λειτουργικό σύστημα.

## **1.7 Ιστορική αναδρομή**

Οι πρώτοι υπολογιστές κατασκευάστηκαν για να εκτελούν μια σειρά από μεμονωμένες εργασίες, όπως μια αριθμομηχανή. Τα βασικά χαρακτηριστικά του λειτουργικού συστήματος αναπτύχθηκαν τη δεκαετία του 1950, όπως οι λειτουργίες

μόνιμης οθόνης που μπορούσαν να εκτελούν αυτόματα διαφορετικά προγράμματα διαδοχικά για να επιταχύνουν την επεξεργασία. Τα λειτουργικά συστήματα δεν υπήρχαν στις σύγχρονες και πιο περίπλοκες μορφές τους μέχρι τις αρχές της δεκαετίας του 1960. Προστέθηκαν χαρακτηριστικά υλικού, που επέτρεψαν τη χρήση βιβλιοθηκών χρόνου εκτέλεσης, διακοπές και παράλληλη επεξεργασία. Όταν οι προσωπικοί υπολογιστές έγιναν δημοφιλείς στη δεκαετία του 1980, τα λειτουργικά συστήματα κατασκευάστηκαν για αυτούς παρόμοια στην ιδέα με αυτά που χρησιμοποιούνται σε μεγαλύτερους υπολογιστές. Στη δεκαετία του 1940, τα πρώτα ηλεκτρονικά ψηφιακά συστήματα δεν είχαν λειτουργικά συστήματα. Τα ηλεκτρονικά συστήματα αυτής της εποχής προγραμματιζόνταν σε σειρές μηχανικών διακοπών ή με καλώδια βραχυκυκλωτήρα σε πίνακες. Αυτά ήταν συστήματα ειδικού σκοπού που, για παράδειγμα, δημιουργούσαν πίνακες βαλλιστικών για τον στρατό ή έλεγχαν την εκτύπωση επιταγών μισθοδοσίας από δεδομένα σε διάτρητες χάρτινες κάρτες. Αφού εφευρέθηκαν προγραμματιζόμενοι υπολογιστές γενικής χρήσης, εισήχθησαν γλώσσες μηχανών (αποτελούμενες από σειρές των δυαδικών ψηφίων 0 και 1 σε διάτρητη χαρτοταινία) που επιτάχυναν τη διαδικασία προγραμματισμού (Stern, 1981). Στις αρχές της δεκαετίας του 1950, ένας υπολογιστής μπορούσε να εκτελεί μόνο ένα πρόγραμμα κάθε φορά. Κάθε χρήστης είχε την αποκλειστική χρήση του υπολογιστή για περιορισμένο χρονικό διάστημα και θα έφτανε σε μια προγραμματισμένη ώρα με το πρόγραμμα και τα δεδομένα του σε διάτρητες χάρτινες κάρτες ή διάτρητη ταινία. Το πρόγραμμα θα φορτωνόταν στο μηχάνημα και το μηχάνημα θα ρυθμιζόταν να λειτουργεί μέχρι να ολοκληρωθεί ή να διακοπεί το πρόγραμμα. Τα προγράμματα θα μπορούσαν γενικά να εντοπιστούν σφαλμάτων μέσω ενός μπροστινού πίνακα χρησιμοποιώντας διακόπτες εναλλαγής και φώτα πίνακα. Λέγεται ότι ο Άλαν Τούρινγκ ήταν κύριος σε αυτό στην πρώιμη μηχανή του Μάντσεστερ Mark 1, και έβγαζε ήδη την πρωτόγονη σύλληψη ενός λειτουργικού συστήματος από τις αρχές της καθολικής μηχανής Turing. Στα τέλη της δεκαετίας του 1950 άρχισαν να εμφανίζονται προγράμματα που θα αναγνώριζε κανείς ως λειτουργικό σύστημα. Συχνά αναφέρεται ως το παλαιότερο αναγνωρίσιμο παράδειγμα είναι το GM-NAA I/O, το οποίο κυκλοφόρησε το 1956 στο IBM 704. Το πρώτο γνωστό παράδειγμα που στην πραγματικότητα αναφέρεται ήταν το SHARE λειτουργικό, μια εξέλιξη του GM-NAA I/O, που κυκλοφόρησε το 1959.

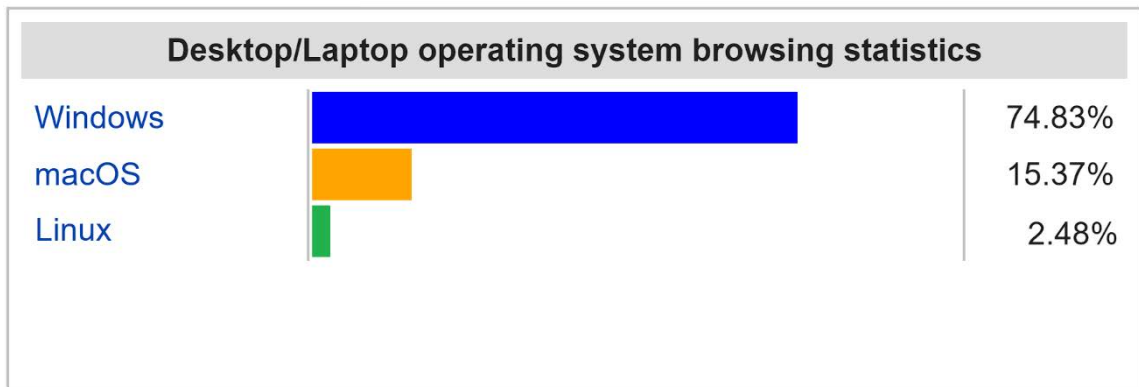
## 1.8 Σύγχρονα λειτουργικά συστήματα

Στην τρέχουσα αγορά λογισμικού, τα τρία πιο συχνά χρησιμοποιούμενα λειτουργικά συστήματα παγκοσμίως ενός ηλεκτρονικού υπολογιστή είναι τα παρακάτω:

- Microsoft Windows
- Apple Mac OS
- Linux

## 1.9 Μερίδιο χρήσης λειτουργικών συστημάτων

Το μερίδιο χρήσης των λειτουργικών συστημάτων είναι το ποσοστό των υπολογιστικών συσκευών που εκτελούν κάθε λειτουργικό σύστημα σε οποιαδήποτε συγκεκριμένη στιγμή. Όλα αυτά τα στοιχεία είναι εκτιμήσεις, επειδή είναι δύσκολο να ληφθούν δεδομένα σχετικά με το μερίδιο του λειτουργικού συστήματος. Υπάρχουν λίγες αξιόπιστες βασικές πηγές και δεν υπάρχουν συγκεκριμένες μεθοδολογίες για τη συλλογή του. Τα λειτουργικά συστήματα χρησιμοποιούνται σε πολλούς τύπους συσκευών, από ενσωματωμένες συσκευές χωρίς οθόνη έως υπερυπολογιστές.



Εικόνα 2. Μερίδιο χρήσης λειτουργικών συστημάτων

Πηγή: <https://gs.statcounter.com/os-market-share>

## 1.10 Windows 10

Τα Windows 10 είναι μια σημαντική έκδοση του λειτουργικού συστήματος Windows NT της Microsoft. Είναι ο άμεσος διάδοχος των Windows 8.1, που κυκλοφόρησε σχεδόν δύο χρόνια νωρίτερα. Κυκλοφόρησε στην κατασκευή στις 15 Ιουλίου 2015 και αργότερα στη λιανική στις 29 Ιουλίου 2015. Τα Windows 10 έγιναν διαθέσιμα για λήψη μέσω MSDN και TechNet, ως δωρεάν αναβάθμιση για αντίγραφα λιανικής των χρηστών Windows 8 και Windows 8.1 μέσω του Windows Store και σε χρήστες Windows 7 μέσω του Windows Update. Τα Windows 10 λαμβάνουν νέες εκδόσεις σε συνεχή βάση, οι οποίες είναι διαθέσιμες χωρίς επιπλέον κόστος για τους χρήστες, επιπλέον των πρόσθετων δοκιμαστικών εκδόσεων των Windows 10, οι οποίες είναι διαθέσιμες στους Windows Insiders. Οι συσκευές σε εταιρικά περιβάλλοντα μπορούν να λαμβάνουν αυτές τις ενημερώσεις με πιο αργό ρυθμό ή να χρησιμοποιούν ορόσημα μακροπρόθεσμης υποστήριξης που λαμβάνουν μόνο κρίσιμες ενημερώσεις, όπως ενημερώσεις κώδικα ασφαλείας, κατά τη διάρκεια της δεκαετούς διάρκειας εκτεταμένης υποστήριξής τους. Η Microsoft έχει δηλώσει ότι η εκτεταμένη υποστήριξη για εκδόσεις των Windows 10 που δεν βρίσκονται στο Κανάλι Μακροπρόθεσμης Εξυπηρέτησης θα λήξει στις 14 Οκτωβρίου 2025.

## 1.11 Mac OS X

Το macOS (παλαιότερα Mac OS X και αργότερα μετονομάστηκαν OS X) είναι ένα λειτουργικό σύστημα Unix που αναπτύχθηκε και διατίθεται στην αγορά από την Apple Inc. από το 2001. Είναι το κύριο λειτουργικό σύστημα για υπολογιστές Mac της Apple. Στην αγορά των επιτραπέζιων και φορητών υπολογιστών είναι το δεύτερο πιο ευρέως χρησιμοποιούμενο επιτραπέζιο λειτουργικό σύστημα, μετά τα Microsoft Windows και μπροστά από το ChromeOS. Το macOS διαδέχθηκε το κλασικό Mac OS, ένα λειτουργικό σύστημα Macintosh με εννέα εκδόσεις από το 1984 έως το 1999. Κατά τη διάρκεια αυτής της περιόδου, ο συνιδρυτής της Apple, Steve Jobs, είχε εγκαταλείψει την Apple και ξεκίνησε μια άλλη εταιρεία, τη NeXT, αναπτύσσοντας την πλατφόρμα NeXTSTEP που αργότερα θα εξαγοραζόταν από την Apple για να σχηματίσει τη βάση του macOS. Η πρώτη έκδοση για υπολογιστές, Mac OS X 10.0, κυκλοφόρησε τον Μάρτιο του 2001, με την πρώτη της ενημέρωση, την 10.1, να έρχεται αργότερα το ίδιο έτος. Όλες οι εκδόσεις από το Mac OS X 10.5 Leopard και μετά έχουν πιστοποίηση UNIX 03, με εξαίρεση το OS X 10.7 Lion.

## 1.12 Linux

Το Linux είναι ένα λειτουργικό σύστημα βασισμένο στο Unix, το οποίο δημιουργήθηκε λειτουργικό σύστημα ανοιχτού κώδικα τύπου Unix που βασίζεται στον πυρήνα Linux, ένας πυρήνας λειτουργικού συστήματος που κυκλοφόρησε για πρώτη φορά στις 17 Σεπτεμβρίου, 1991, από τον Linus Torvalds. Το Linux συσκευάζεται συνήθως ως διανομή Linux. Οι διανομές περιλαμβάνουν τον πυρήνα Linux και το λογισμικό υποστήριξης του συστήματος και τις βιβλιοθήκες, πολλές από τις οποίες παρέχονται από το έργο GNU. Πολλές διανομές Linux χρησιμοποιούν τη λέξη "Linux" στο όνομά τους, αλλά το Ίδρυμα Ελεύθερου Λογισμικού χρησιμοποιεί το όνομα "GNU/Linux" για να τονίσει τη σημασία του λογισμικού GNU, προκαλώντας κάποιες διαμάχες. Το Linux αναπτύχθηκε αρχικά για προσωπικούς υπολογιστές με βάση την αρχιτεκτονική Intel x86, αλλά έκτοτε έχει μεταφερθεί σε περισσότερες πλατφόρμες από οποιοδήποτε άλλο λειτουργικό σύστημα. Λόγω της κυριαρχίας του Android που βασίζεται σε Linux σε smartphone, το Linux, συμπεριλαμβανομένου του Android, έχει τη μεγαλύτερη εγκατεστημένη βάση από όλα τα λειτουργικά συστήματα γενικής χρήσης, από τον Μάιο του 2022. Αν και το Linux χρησιμοποιείται, από τον Μάιο του 2022, μόνο από περίπου το 2,3 τοις εκατό των επιτραπέζιων υπολογιστών, το Chromebook, το οποίο εκτελεί το Chrome OS που βασίζεται σε πυρήνα Linux, κυριαρχεί στην αγορά εκπαίδευσης K-12 των ΗΠΑ και αντιπροσωπεύει σχεδόν το 20 τοις εκατό των υπο - Πωλήσεις notebook \$300 στις ΗΠΑ. Το Linux είναι το κορυφαίο λειτουργικό σύστημα στους διακομιστές (πάνω από το 96,4% των λειτουργικών συστημάτων του 1 εκατομμυρίου κορυφαίων διακομιστών Ιστού είναι Linux), ηγείται άλλων μεγάλων σιδερένιων συστημάτων, όπως οι υπολογιστές μεγάλου μεγέθους, και είναι το μόνο λειτουργικό σύστημα που χρησιμοποιείται σε υπερυπολογιστές TOP500 (από Νοεμβρίου 2017, έχοντας αποκλείσει σταδιακά όλους τους ανταγωνιστές). Το Linux εκτελείται



επίσης σε ενσωματωμένα συστήματα, δηλαδή συσκευές των οποίων το λειτουργικό σύστημα είναι συνήθως ενσωματωμένο στο υλικολογισμικό και είναι ιδιαίτερα προσαρμοσμένο στο σύστημα. Το Linux είναι ένα από τα πιο χαρακτηριστικά παραδείγματα συνεργασίας ελεύθερου λογισμικού και λογισμικού ανοιχτού κώδικα. Ο πηγαίος κώδικας μπορεί να χρησιμοποιηθεί, να τροποποιηθεί και να διανεμηθεί εμπορικά ή μη από οποιονδήποτε σύμφωνα με τους όρους των αντίστοιχων αδειών του, όπως η Γενική Δημόσια Άδεια GNU (GPL). Ο πυρήνας Linux, για παράδειγμα, έχει άδεια χρήσης σύμφωνα με το GPLv2, με ειδική εξαίρεση για κλήσεις συστήματος, καθώς χωρίς την εξαίρεση κλήσης συστήματος, οποιοδήποτε πρόγραμμα καλεί τον πυρήνα θα θεωρείται παράγωγο και επομένως η GPL θα πρέπει να ισχύει για αυτό το πρόγραμμα.



Εικόνα 3. Τα τρία δημοφιλέστερα λειτουργικά συστήματα υπολογιστή

Πηγή: <https://computertechpro.net/linux-on-a-desktop/>

## 2. .NET Framework

Το .NET Framework (προφέρεται ως "dot net") είναι ένα ιδιόκτητο πλαίσιο λογισμικού που αναπτύχθηκε από τη Microsoft και εκτελείται κυρίως στα Microsoft Windows. Ήταν η κυρίαρχη υλοποίηση της υποδομής κοινής γλώσσας (CLI) μέχρι να αντικατασταθεί από το έργο .NET πολλαπλών πλατφορμών. Περιλαμβάνει μια μεγάλη βιβλιοθήκη κλάσεων που ονομάζεται Framework Class Library (FCL) και παρέχει γλωσσική διαλειτουργικότητα (κάθε γλώσσα μπορεί να χρησιμοποιήσει κώδικα γραμμένο σε άλλες γλώσσες) σε διάφορες γλώσσες προγραμματισμού. Τα προγράμματα που γράφονται για το .NET Framework εκτελούνται σε ένα περιβάλλον λογισμικού (σε αντίθεση με ένα περιβάλλον υλικού) που ονομάζεται Common Language Runtime (CLR). Το CLR είναι μια εικονική μηχανή εφαρμογών που παρέχει υπηρεσίες όπως η

ασφάλεια, η διαχείριση μνήμης και ο χειρισμός εξαιρέσεων. Ως εκ τούτου, ο κώδικας υπολογιστή που έχει γραφτεί με το .NET Framework ονομάζεται "διαχειριζόμενος κώδικας". Το FCL και το CLR μαζί αποτελούν το .NET Framework. Το FCL παρέχει τη διεπαφή χρήστη, την πρόσβαση σε δεδομένα, τη συνδεσιμότητα βάσεων δεδομένων, την κρυπτογραφία, την ανάπτυξη εφαρμογών ιστού, τους αριθμητικούς αλγορίθμους και τις επικοινωνίες δικτύου. Οι προγραμματιστές παράγουν λογισμικό συνδυάζοντας τον πηγαίο τους κώδικα με το .NET Framework και άλλες βιβλιοθήκες. Το πλαίσιο προορίζεται να χρησιμοποιηθεί από τις περισσότερες νέες εφαρμογές που δημιουργούνται για την πλατφόρμα των Windows. Η Microsoft παράγει επίσης ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού .NET που ονομάζεται Visual Studio. Το .NET Framework ξεκίνησε ως ιδιόκτητο λογισμικό, αν και η εταιρεία εργάστηκε για την τυποποίηση της στοίβας λογισμικού σχεδόν αμέσως, ακόμη και πριν από την πρώτη του έκδοση. Παρά τις προσπάθειες τυποποίησης, οι προγραμματιστές, κυρίως αυτοί που ανήκουν στις κοινότητες ελεύθερου και ανοικτού λογισμικού, εξέφρασαν τη δυσφορία τους για τους επιλεγμένους όρους και τις προοπτικές οποιασδήποτε εφαρμογής ελεύθερου και ανοικτού λογισμικού, ιδίως όσον αφορά τις πατέντες λογισμικού. Εκτοτε, η Microsoft άλλαξε την ανάπτυξη του .NET ώστε να ακολουθεί περισσότερο το σύγχρονο μοντέλο ενός έργου λογισμικού που αναπτύσσεται από την κοινότητα, συμπεριλαμβανομένης της έκδοσης μιας ενημερωμένης έκδοσης των πατεντών της που υπόσχεται να αντιμετωπίσει τις ανησυχίες.

## 2.1 Αρχιτεκτονική

### 2.1.1 Κοινή γλωσσική υποδομή

Η Υποδομή Κοινής Γλώσσας (CLI) είναι μια ανοικτή προδιαγραφή και τεχνικό πρότυπο που αναπτύχθηκε αρχικά από τη Microsoft και τυποποιήθηκε από το ISO/IEC (ISO/IEC 23271) και το Ecma International (ECMA 335) το οποίο περιγράφει εκτελέσιμο κώδικα και ένα περιβάλλον εκτέλεσης που επιτρέπει τη χρήση πολλαπλών γλωσσών υψηλού επιπέδου σε διαφορετικές πλατφόρμες υπολογιστών χωρίς να χρειάζεται να ξαναγραφούν για συγκεκριμένες αρχιτεκτονικές. Αυτό σημαίνει ότι είναι ανεξάρτητη από τις πλατφόρμες. Τα .NET Framework, .NET και Mono είναι υλοποιήσεις του CLI. Η μορφή μεταδεδομένων χρησιμοποιείται επίσης για τον προσδιορισμό των ορισμών API που εκτίθενται από το Windows Runtime. Το CLI παρέχει μια γλωσσικά ουδέτερη πλατφόρμα για την ανάπτυξη και εκτέλεση εφαρμογών. Με την υλοποίηση των βασικών πυλών του .NET Framework εντός του πεδίου εφαρμογής του CLI, οι λειτουργίες αυτές δεν θα είναι συνδεδεμένες με μια γλώσσα αλλά θα είναι διαθέσιμες σε όλες τις πολλές γλώσσες που υποστηρίζονται από το πλαίσιο.

## 2.1.2 Κοινό περιβάλλον εκτέλεσης γλωσσών

Το Common Language Runtime (CLR), το συστατικό της εικονικής μηχανής του Microsoft .NET Framework, διαχειρίζεται την εκτέλεση των προγραμμάτων .NET. Η μεταγλώττιση Just-in-time μετατρέπει τον διαχειριζόμενο κώδικα (μεταγλωττισμένους κώδικας ενδιάμεσης γλώσσας) σε εντολές μηχανής, οι οποίες στη συνέχεια εκτελούνται στη CPU του υπολογιστή. Το CLR παρέχει πρόσθετες υπηρεσίες, όπως διαχείριση μνήμης, ασφάλεια τύπων, χειρισμό εξαιρέσεων, συλλογή σκουπιδιών, ασφάλεια και διαχείριση νημάτων. Όλα τα προγράμματα που έχουν γραφτεί για το .NET Framework, ανεξάρτητα από τη γλώσσα προγραμματισμού, εκτελούνται στο CLR. Όλες οι εκδόσεις του .NET Framework περιλαμβάνουν το CLR. Η ομάδα του CLR ξεκίνησε στις 13 Ιουνίου 1998. Το .NET Framework περιλαμβάνει το Common Language Runtime (CLR). Χρησιμεύει ως μηχανή εκτέλεσης του .NET Framework και προσφέρει πολλές υπηρεσίες, όπως διαχείριση μνήμης, ασφάλεια τύπων, χειρισμό εξαιρέσεων, συλλογή σκουπιδιών, ασφάλεια και διαχείριση νημάτων. Όλα τα προγράμματα που έχουν γραφτεί για το .NET Framework εκτελούνται από το CLR. Τα προγράμματα που γράφονται για το .NET Framework μεταγλωττίζονται σε κώδικα κοινής ενδιάμεσης γλώσσας (CIL), σε αντίθεση με την απευθείας μεταγλώττιση σε κώδικα μηχανής. Κατά τη διάρκεια της εκτέλεσης, ένας μεταγλωττιστής just-in-time (JIT), ειδικής αρχιτεκτονικής, μετατρέπει τον κώδικα CIL σε κώδικα μηχανής.



Εικόνα 4. Common Language Runtime (CLR)

Πηγή: <https://www.javatpoint.com/net-framework>

## 2.1.3 Assemblies

Ο μεταγλωττισμένος κώδικας CIL αποθηκεύεται σε σύνολα CLI. Ένα assembly στην Υποδομή Κοινής Γλώσσας (CLI) είναι μια βιβλιοθήκη μεταγλωττισμένου κώδικα που χρησιμοποιείται για την ανάπτυξη, την έκδοση και την ασφάλεια. Όπως επιβάλλει η προδιαγραφή, τα σύνολα αποθηκεύονται σε μορφή αρχείου Portable Executable (PE), κοινή στην πλατφόρμα των Windows για όλες τις βιβλιοθήκες δυναμικής σύνδεσης (DLL) και τα εκτελέσιμα αρχεία EXE. Κάθε συγκρότημα αποτελείται από ένα ή περισσότερα αρχεία, ένα από τα οποία πρέπει να περιέχει ένα μανιφέστο με τα μεταδεδομένα του συγκροτήματος. Το πλήρες όνομα μιας συνέλευσης (δεν πρέπει να συγχέεται με το όνομα του αρχείου στο δίσκο) περιέχει το απλό όνομα κειμένου, τον αριθμό έκδοσης, την κουλτούρα και το διακριτικό δημόσιου κλειδιού. Τα σύνολα θεωρούνται ισοδύναμα αν έχουν το ίδιο πλήρες όνομα. Ένα ιδιωτικό κλειδί μπορεί επίσης να χρησιμοποιηθεί από

τον δημιουργό της συνέλευσης για ισχυρή ονομασία. Το διακριτικό δημόσιου κλειδιού προσδιορίζει με ποιο ιδιωτικό κλειδί έχει υπογραφεί ένα συγκρότημα. Μόνο ο δημιουργός του ζεύγους κλειδιών (συνήθως το άτομο που υπογράφει τη συναρμολόγηση) μπορεί να υπογράψει συναρμολογήσεις που έχουν το ίδιο ισχυρό όνομα με μια συναρμολόγηση προηγούμενης έκδοσης, καθώς ο δημιουργός κατέχει το ιδιωτικό κλειδί. Η ισχυρή ονομασία απαιτείται για την προσθήκη συγκροτημάτων στην Global Assembly Cache. Ξεκινώντας με το Visual Studio 2015, η τεχνολογία μεταγλώττισης .NET Native επιτρέπει τη μεταγλώττιση του κώδικα .NET των εφαρμογών της πλατφόρμας Universal Windows Platform απευθείας σε κώδικα μηχανής και όχι σε κώδικα CIL, αλλά η εφαρμογή πρέπει να είναι γραμμένη είτε σε C# είτε σε Visual Basic.NET.

## 2.1.4 Βιβλιοθήκη κλάσεων

Το .NET Framework περιλαμβάνει μια υλοποίηση των θεμελιωδών τυποποιημένων βιβλιοθηκών CLI. Οι Πρότυπες Βιβλιοθήκες είναι ένα σύνολο βιβλιοθηκών που περιλαμβάνονται στην Υποδομή Κοινής Γλώσσας (CLI) με σκοπό την ενθυλάκωση πολλών κοινών λειτουργιών, όπως η ανάγνωση και εγγραφή αρχείων, ο χειρισμός εγγράφων XML, ο χειρισμός εξαιρέσεων, η παγκοσμιοποίηση εφαρμογών, η επικοινωνία μέσω δικτύου, η νηματοποίηση και η αντανάκλαση, γεγονός που διευκολύνει τη δουλειά του προγραμματιστή. Η βιβλιοθήκη κλάσεων του .NET Framework (FCL) είναι οργανωμένη σε μια ιεραρχία χώρων ονομάτων. Οι περισσότερες από τις ενσωματωμένες διασυνδέσεις προγραμματισμού εφαρμογών (APIs) αποτελούν μέρος είτε των χώρων ονομάτων `System.*` είτε των χώρων ονομάτων `Microsoft.*`. Αυτές οι βιβλιοθήκες κλάσεων υλοποιούν πολλές κοινές λειτουργίες, όπως η ανάγνωση και εγγραφή αρχείων, η απόδοση γραφικών, η αλληλεπίδραση με βάσεις δεδομένων και ο χειρισμός εγγράφων XML. Οι βιβλιοθήκες κλάσεων είναι διαθέσιμες για όλες τις γλώσσες που είναι συμβατές με το CLI. Η FCL υλοποιεί τη Βιβλιοθήκη Κλάσεων Βάσης CLI (BCL) και άλλες βιβλιοθήκες κλάσεων - ορισμένες καθορίζονται από το CLI και άλλες είναι συγκεκριμένες για τη Microsoft. Η BCL περιλαμβάνει ένα μικρό υποσύνολο ολόκληρης της βιβλιοθήκης κλάσεων και είναι το βασικό σύνολο κλάσεων που χρησιμεύει ως το βασικό API του CLR. Για το .NET Framework, οι περισσότερες κλάσεις που θεωρούνται μέρος της BCL βρίσκονται στα `microsoft.dll`, `System.dll` και `System.Core.dll`. Οι κλάσεις BCL είναι διαθέσιμες στο .NET Framework καθώς και στις εναλλακτικές υλοποιήσεις του, συμπεριλαμβανομένων των .NET Compact Framework, Microsoft Silverlight, .NET Core και Mono. Η FCL αναφέρεται σε ολόκληρη τη βιβλιοθήκη κλάσεων που συνοδεύει το .NET Framework. Περιλαμβάνει ένα διευρυμένο σύνολο βιβλιοθηκών, συμπεριλαμβανομένων των BCL, Windows Forms, ASP.NET και Windows Presentation Foundation (WPF), αλλά και επεκτάσεις στις βασικές βιβλιοθήκες κλάσεων ADO.NET, Language Integrated Query (LINQ), Windows Communication Foundation (WCF) και Workflow Foundation (WF). Η FCL είναι πολύ μεγαλύτερη σε πεδίο εφαρμογής από τις τυπικές βιβλιοθήκες για γλώσσες όπως η C++ και συγκρίσιμη σε πεδίο εφαρμογής με τις τυπικές βιβλιοθήκες της Java. Με την εισαγωγή εναλλακτικών

υλοποιήσεων (π.χ. Silverlight), η Microsoft εισήγαγε την έννοια των φορητών βιβλιοθηκών κλάσεων (Portable Class Libraries - PCL) που επιτρέπουν σε μια βιβλιοθήκη κατανάλωσης να εκτελείται σε περισσότερες από μία πλατφόρμες. Με τον περαιτέρω πολλαπλασιασμό των πλατφορμών .NET, η προσέγγιση PCL απέτυχε να κλιμακωθεί (οι PCL είναι καθορισμένες διασταυρώσεις της επιφάνειας API μεταξύ δύο ή περισσότερων πλατφορμών). Ως το επόμενο εξελικτικό βήμα της PCL, η .NET Standard Library δημιουργήθηκε αναδρομικά με βάση τα βασισμένα στο `System.Runtime.dll` APIs που βρίσκονται στο UWP και το Silverlight. Οι νέες πλατφόρμες .NET ενθαρρύνονται να υλοποιήσουν μια έκδοση της τυποποιημένης βιβλιοθήκης που τους επιτρέπει να επαναχρησιμοποιούν υπάρχουσες βιβλιοθήκες τρίτων κατασκευαστών για να εκτελούνται χωρίς νέες εκδόσεις τους. Η τυποποιημένη βιβλιοθήκη .NET επιτρέπει την ανεξάρτητη εξέλιξη των επιπέδων βιβλιοθηκών και μοντέλων εφαρμογών εντός της αρχιτεκτονικής .NET. Το NuGet είναι ο διαχειριστής πακέτων για όλες τις πλατφόρμες .NET. Χρησιμοποιείται για την ανάκτηση βιβλιοθηκών τρίτων σε ένα έργο .NET με μια παγκόσμια τροφοδοσία βιβλιοθηκών στο NuGet.org. Οι ιδιωτικές τροφοδοσίες μπορούν να διατηρούνται ξεχωριστά, π.χ. από έναν διακομιστή κατασκευής ή έναν κατάλογο του συστήματος αρχείων.

### 2.1.5 C++/CLI

Η Microsoft εισήγαγε το C++/CLI στο Visual Studio 2005, το οποίο είναι μια γλώσσα και ένα μέσο μεταγλώττισης προγραμμάτων Visual C++ για να εκτελούνται εντός του .NET Framework. Ορισμένα τμήματα του προγράμματος C++ εξακολουθούν να εκτελούνται μέσα σε ένα μη διαχειριζόμενο Visual C++ Runtime, ενώ ειδικά τροποποιημένα τμήματα μεταφράζονται σε κώδικα CIL και εκτελούνται με το CLR του .NET Framework. Τα σύνολα που μεταγλωττίζονται με τη χρήση του μεταγλωττιστή C++/CLI ονομάζονται σύνολα μικτού τρόπου λειτουργίας, καθώς περιέχουν εγγενή και διαχειριζόμενο κώδικα στο ίδιο DLL. Τέτοια συγκροτήματα είναι πιο πολύπλοκα στην αντίστροφη μηχανική, δεδομένου ότι το .NET decompilers όπως ο .NET Reflector αποκαλύπτουν μόνο τον διαχειριζόμενο κώδικα.

## 2.2 Σχεδιαστικές αρχές

### 2.2.1 Διαλειτουργικότητα

Επειδή τα συστήματα υπολογιστών απαιτούν συνήθως αλληλεπίδραση μεταξύ νεότερων και παλαιότερων εφαρμογών, το .NET Framework παρέχει μέσα πρόσβασης σε λειτουργίες που υλοποιούνται σε νεότερα και παλαιότερα προγράμματα που εκτελούνται εκτός του περιβάλλοντος .NET. Η πρόσβαση σε ένα πρότυπο δυαδικής διασύνδεσης για στοιχεία λογισμικού που ονομάζεται Μοντέλο αντικειμένου στοιχείου (Component Object Model - COM) παρέχεται στο System. Runtime. InteropServices και System. Enterprise Services του πλαισίου. Η πρόσβαση σε άλλες λειτουργίες γίνεται μέσω των



Platform Invocation Services (P/Invoke) που είναι ένα χαρακτηριστικό των υλοποιήσεων της Common Language Infrastructure, όπως το Common Language Runtime της Microsoft, το οποίο επιτρέπει στον διαχειριζόμενο κώδικα να καλεί εγγενή κώδικα. Η πρόσβαση σε συναρτήσεις .NET από εγγενείς εφαρμογές γίνεται μέσω της αντίστροφης λειτουργίας P/Invoke. Η διαλειτουργικότητά σας επιτρέπει να διατηρήσετε και να επωφεληθείτε από τις υπάρχουσες επενδύσεις σε μη διαχειριζόμενο κώδικα. Ο κώδικας που εκτελείται υπό τον έλεγχο του common language runtime (CLR) ονομάζεται διαχειριζόμενος κώδικας, ενώ ο κώδικας που εκτελείται εκτός του CLR ονομάζεται μη διαχειριζόμενος κώδικας.

### 2.2.2 Γλωσσική ανεξαρτησία

Το .NET Framework εισάγει ένα Common Type System (CTS) που ορίζει όλους τους πιθανούς τύπους δεδομένων και τις προγραμματιστικές δομές που υποστηρίζονται από το CLR και τον τρόπο με τον οποίο μπορούν ή δεν μπορούν να αλληλεπιδρούν σύμφωνα με τις προδιαγραφές του CLI. Εξαιτίας αυτού του χαρακτηριστικού, το .NET Framework υποστηρίζει την ανταλλαγή τύπων και περιπτώσεων αντικειμένων μεταξύ βιβλιοθηκών και εφαρμογών που έχουν γραφτεί χρησιμοποιώντας οποιαδήποτε συμμορφούμενη γλώσσα .NET.

### 2.2.3 Ασφάλεια τύπων

Το CTS και το CLR που χρησιμοποιείται στο .NET Framework επιβάλλουν επίσης την ασφάλεια τύπων. Αυτό αποτρέπει τις κακώς καθορισμένες εκτροπές, τις λανθασμένες κλήσεις μεθόδων και τα προβλήματα μεγέθους μνήμης κατά την πρόσβαση σε ένα αντικείμενο. Αυτό καθιστά επίσης τις περισσότερες γλώσσες CLI στατικά τυποποιημένες (με ή χωρίς εξαγωγή συμπερασμάτων τύπου). Ωστόσο, ξεκινώντας από το .NET Framework 4.0, το Dynamic Language Runtime επέκτεινε το CLR, επιτρέποντας την υλοποίηση δυναμικά τυποποιημένων γλωσσών πάνω στο CLI.

### 2.2.4 Φορητότητα

Παρόλο που η Microsoft δεν έχει υλοποιήσει ποτέ το πλήρες πλαίσιο σε κανένα σύστημα εκτός από τα Microsoft Windows, έχει σχεδιάσει το πλαίσιο ώστε να είναι διαπλατφορμικό, και είναι διαθέσιμες υλοποιήσεις για άλλα λειτουργικά συστήματα (βλέπε Silverlight και § Εναλλακτικές υλοποιήσεις). Η Microsoft υπέβαλε τις προδιαγραφές για το CLI (το οποίο περιλαμβάνει τις βασικές βιβλιοθήκες κλάσεων, το CTS και το CIL), C# και C++/CLI τόσο στην Ecma International (ECMA) όσο και στον Διεθνή Οργανισμό Τυποποίησης (ISO), καθιστώντας τις διαθέσιμες ως επίσημα πρότυπα. Αυτό δίνει τη δυνατότητα σε τρίτους να δημιουργήσουν συμβατές υλοποιήσεις του πλαισίου και των γλωσσών του σε άλλες πλατφόρμες.

## 2.2.5 Ασφάλεια

Το .NET Framework διαθέτει τον δικό του μηχανισμό ασφαλείας με δύο γενικά χαρακτηριστικά: Ασφάλεια πρόσβασης στον κώδικα (CAS) και επικύρωση και επαλήθευση. Το CAS βασίζεται σε αποδεικτικά στοιχεία που σχετίζονται με μια συγκεκριμένη συναρμολόγηση. Συνήθως τα αποδεικτικά στοιχεία είναι η πηγή του συγκροτήματος (αν είναι εγκατεστημένο στον τοπικό υπολογιστή ή αν έχει ληφθεί από το Διαδίκτυο). Το CAS χρησιμοποιεί αποδεικτικά στοιχεία για να καθορίσει τα δικαιώματα που χορηγούνται στον κώδικα. Άλλοι κώδικες μπορούν να απαιτήσουν να χορηγηθεί στον καλούντα κώδικα μια συγκεκριμένη άδεια. Η απαίτηση αναγκάζει το CLR να εκτελέσει μια περιδιάβαση στη στοίβα κλήσεων: κάθε συνέλευση κάθε μεθόδου στη στοίβα κλήσεων ελέγχεται για την απαιτούμενη άδεια- εάν σε κάποια συνέλευση δεν χορηγηθεί η άδεια, δημιουργείται μια εξαίρεση ασφαλείας. Ο διαχειριζόμενος bytecode CIL είναι ευκολότερο να αντιστραφεί σε σχέση με τον εγγενή κώδικα, εκτός αν είναι συγκεκαλυμμένος. .NET decompiler προγράμματα επιτρέπουν σε προγραμματιστές χωρίς δεξιότητες αντίστροφης μηχανικής να βλέπουν τον πηγαίο κώδικα πίσω από μη αποκρυπτογραφημένα .NET assemblies. Αντίθετα, οι εφαρμογές που έχουν μεταγλωττιστεί σε εγγενή κώδικα μηχανής είναι πολύ πιο δύσκολο να αντιστραφούν και ο πηγαίος κώδικας δεν παράγεται σχεδόν ποτέ με επιτυχία, κυρίως λόγω των βελτιστοποιήσεων του μεταγλωττιστή και της έλλειψης αντανάκλασης. Αυτό δημιουργεί ανησυχίες στην επιχειρηματική κοινότητα σχετικά με την πιθανή απώλεια εμπορικών μυστικών και την παράκαμψη των μηχανισμών ελέγχου αδειών χρήσης. Για τον μετριασμό αυτού του προβλήματος, η Microsoft έχει συμπεριλάβει Dotfuscator Community Edition με το Visual Studio .NET από το 2002. Υπάρχουν επίσης διαθέσιμα εργαλεία συσκοτισμού τρίτων κατασκευαστών από προμηθευτές όπως η VMware, V.i. Labs, Turbo και Red Gate Software. Εργαλεία κρυπτογράφησης σε επίπεδο μεθόδου για κώδικα .NET είναι διαθέσιμα από προμηθευτές όπως η SafeNet.

## 2.2.6 Διαχείριση μνήμης

Το CLR απαλλάσσει τον προγραμματιστή από το βάρος της διαχείρισης της μνήμης (κατανομή και αποδέσμευση όταν τελειώσει)- χειρίζεται το ίδιο τη διαχείριση της μνήμης ανιχνεύοντας τότε η μνήμη μπορεί να αποδεσμευτεί με ασφάλεια. Οι ενσαρκώσεις των τύπων .NET (αντικείμενα) κατανέμονται από τον διαχειριζόμενο σωρό, μια δεξαμενή μνήμης που διαχειρίζεται το CLR. Εφόσον υπάρχει αναφορά σε ένα αντικείμενο, η οποία μπορεί να είναι είτε άμεση είτε μέσω ενός γράφου αντικειμένων, το αντικείμενο θεωρείται ότι χρησιμοποιείται. Όταν δεν υπάρχει καμία αναφορά σε ένα αντικείμενο και δεν μπορεί να προσεγγιστεί ή να χρησιμοποιηθεί, μετατρέπεται σε σκουπίδι, επιλέξιμο για αποκομιδή. Το .NET Framework περιλαμβάνει έναν συλλέκτη σκουπιδιών (garbage collector - GC), ο οποίος εκτελείται περιοδικά, σε ξεχωριστό νήμα από το νήμα της εφαρμογής, που απαριθμεί όλα τα άχρηστα αντικείμενα και ανακτά τη μνήμη που τους έχει διατεθεί. Πρόκειται για έναν μη ντετερμινιστικό, συμπιεστικό, mark-and-sweep garbage collector. Ο GC εκτελείται μόνο όταν έχει χρησιμοποιηθεί μια καθορισμένη

ποσότητα μνήμης ή όταν υπάρχει αρκετή πίεση για μνήμη στο σύστημα. Δεδομένου ότι δεν είναι εγγυημένο τότε επιτυγχάνονται οι συνθήκες για την ανάκτηση μνήμης, οι εκτελέσεις του GC είναι μη ντετερμινιστικές. Κάθε εφαρμογή .NET έχει ένα σύνολο από ρίζες, οι οποίες είναι δείκτες σε αντικείμενα στο διαχειριζόμενο σωρό (*managed objects*). Αυτές περιλαμβάνουν αναφορές σε στατικά αντικείμενα, αντικείμενα που ορίζονται ως τοπικές μεταβλητές ή παράμετροι μεθόδων που βρίσκονται επί του παρόντος σε εμβέλεια και αντικείμενα στα οποία αναφέρονται οι καταχωρητές της CPU. Όταν εκτελείται το GC, διακόπτει την εφαρμογή και στη συνέχεια, για κάθε αντικείμενο που αναφέρεται στη ρίζα, απαριθμεί αναδρομικά όλα τα αντικείμενα που είναι προσβάσιμα από τα αντικείμενα της ρίζας και τα χαρακτηρίζει ως προσβάσιμα. Χρησιμοποιεί τα μεταδεδομένα CLI και την αντανάκλαση για να ανακαλύψει τα αντικείμενα που περικλείονται από ένα αντικείμενο και στη συνέχεια τα περιδιαβαίνει αναδρομικά. Στη συνέχεια, απαριθμεί όλα τα αντικείμενα στο σωρό (τα οποία αρχικά κατανεμήθηκαν συνεχόμενα) χρησιμοποιώντας την αντανάκλαση. Όλα τα αντικείμενα που δεν χαρακτηρίζονται ως προσβάσιμα είναι σκουπίδια. Αυτή είναι η φάση *σήμανσης*. Δεδομένου ότι η μνήμη που κατέχουν τα σκουπίδια δεν έχει καμία σημασία, θεωρείται ελεύθερος χώρος. Ωστόσο, αυτό αφήνει κομμάτια ελεύθερου χώρου μεταξύ αντικειμένων που αρχικά ήταν συνεχόμενα. Στη συνέχεια, τα αντικείμενα συμπίεζονται μεταξύ τους για να γίνει ο ελεύθερος χώρος στον διαχειριζόμενο σωρό και πάλι συνεκτικός. Κάθε αναφορά σε ένα αντικείμενο που ακυρώνεται με τη μετακίνηση του αντικειμένου ενημερώνεται από το GC για να αντικατοπτρίζει τη νέα θέση. Η εφαρμογή συνεχίζεται μετά το τέλος της συλλογής σκουπιδιών. Η τελευταία έκδοση του πλαισίου .NET χρησιμοποιεί ταυτόχρονη συλλογή σκουπιδιών μαζί με τον κώδικα του χρήστη, καθιστώντας τις παύσεις απαρατήρητες, επειδή γίνονται στο παρασκήνιο. Ο συλλέκτης σκουπιδιών που χρησιμοποιείται από το .NET Framework είναι επίσης *γεννητικός*. Στα αντικείμενα ανατίθεται μια *γενιά*. Τα νεοδημιουργηθέντα αντικείμενα χαρακτηρίζονται ως *Generation 0*. Τα αντικείμενα που επιβιώνουν από μία συλλογή σκουπιδιών χαρακτηρίζονται ως *Generation 1*. Τα αντικείμενα γενιάς 1 που επιβιώνουν από άλλη συλλογή είναι *γενιάς 2*. Το πλαίσιο χρησιμοποιεί αντικείμενα έως και γενιάς 2. Τα αντικείμενα υψηλότερης γενιάς συλλέγονται σκουπίδια λιγότερο συχνά από τα αντικείμενα χαμηλότερης γενιάς. Αυτό αυξάνει την αποδοτικότητα της συλλογής σκουπιδιών, καθώς τα παλαιότερα αντικείμενα τείνουν να έχουν μεγαλύτερη διάρκεια ζωής από τα νεότερα αντικείμενα. Αγνοώντας τα παλαιότερα αντικείμενα στις περισσότερες εκτελέσεις συλλογής, απαιτούνται συνολικά λιγότεροι έλεγχοι και λειτουργίες συμπίεσης.

### 2.2.7 Επιδόσεις

Όταν μια εφαρμογή εκκινείται για πρώτη φορά, το .NET Framework μεταγλωττίζει τον κώδικα CIL σε εκτελέσιμο κώδικα χρησιμοποιώντας τον μεταγλωττιστή *just-in-time* και αποθηκεύει το εκτελέσιμο πρόγραμμα στην κρυφή μνήμη .NET Native Image Cache. Λόγω της προσωρινής αποθήκευσης, η εφαρμογή εκκινείται ταχύτερα για τις επόμενες εκκινήσεις, αν και η πρώτη εκκίνηση είναι συνήθως πιο αργή. Για να επιταχύνουν την



πρώτη εκκίνηση, οι προγραμματιστές μπορούν να χρησιμοποιήσουν το βοηθητικό πρόγραμμα Native Image Generator για να μεταγλωττίσουν και να αποθηκεύουν στην προσωρινή μνήμη κάθε εφαρμογή .NET χειροκίνητα πριν από τον χρόνο. Ο συλλέκτης σκουπιδιών, ο οποίος είναι ενσωματωμένος στο περιβάλλον, μπορεί να εισάγει απρόβλεπτες καθυστερήσεις στην εκτέλεση, επί των οποίων ο προγραμματιστής έχει ελάχιστο άμεσο έλεγχο. "Σε μεγάλες εφαρμογές, ο αριθμός των αντικειμένων με τα οποία πρέπει να εργαστεί ο συλλέκτης σκουπιδιών μπορεί να γίνει πολύ μεγάλος, πράγμα που σημαίνει ότι μπορεί να χρειαστεί πολύ μεγάλο χρονικό διάστημα για να τα επισκεφθεί και να τα αναδιατάξει όλα". Το .NET Framework παρέχει υποστήριξη για την κλήση των Streaming SIMD Extensions (SSE) μέσω διαχειριζόμενου κώδικα από τον Απρίλιο του 2014 στο Visual Studio 2013 Update 2. Ωστόσο, η Mono παρέχει υποστήριξη για SIMD Extensions από την έκδοση 2.2 στο πλαίσιο του χώρου ονομάτων Mono.Simd το 2009. Ο επικεφαλής προγραμματιστής της Mono Miguel de Icaza έχει εκφράσει την ελπίδα ότι αυτή η υποστήριξη SIMD θα υιοθετηθεί από το πρότυπο ECMA του CLR. Οι Streaming SIMD Extensions είναι διαθέσιμες στις CPUs x86 από την εισαγωγή του Pentium III. Ορισμένες άλλες αρχιτεκτονικές, όπως η ARM και η MIPS, διαθέτουν επίσης επεκτάσεις SIMD. Σε περίπτωση που η CPU δεν υποστηρίζει αυτές τις επεκτάσεις, οι εντολές προσομοιώνονται σε λογισμικό.

## 2.3 Ιστορική αναδρομή

Η Microsoft άρχισε να αναπτύσσει το .NET Framework στα τέλη της δεκαετίας του 1990, αρχικά με την ονομασία Next Generation Windows Services (NGWS), στο πλαίσιο της στρατηγικής .NET. Στις αρχές του 2000, κυκλοφόρησαν οι πρώτες beta εκδόσεις του .NET 1.0. Τον Αύγουστο του 2000, η Microsoft και η Intel εργάστηκαν για την τυποποίηση της κοινής γλωσσικής υποδομής (CLI) και της C#. Τον Δεκέμβριο του 2001, και οι δύο επικυρώθηκαν Ecma International (ECMA) πρότυπα. Τον Απρίλιο του 2003 ακολούθησε ο Διεθνής Οργανισμός Τυποποίησης (ISO). Η τρέχουσα έκδοση των προτύπων ISO είναι τα ISO/IEC 23271:2012 και ISO/IEC 23270:2006. Ενώ η Microsoft και οι συνεργάτες της κατέχουν διπλώματα ευρεσιτεχνίας για το CLI και την C#, η ECMA και το ISO απαιτούν να διατίθενται όλα τα διπλώματα ευρεσιτεχνίας που είναι απαραίτητα για την υλοποίηση υπό "εύλογους και αμερόληπτους όρους". Οι εταιρείες συμφώνησαν να τηρήσουν αυτούς τους όρους και να διαθέσουν τα διπλώματα ευρεσιτεχνίας χωρίς δικαιώματα. Ωστόσο, αυτό δεν ίσχυε για το μέρος του .NET Framework που δεν καλύπτεται από τα πρότυπα ECMA-ISO, το οποίο περιλάμβανε τα Windows Forms, ADO.NET και ASP.NET. Τα διπλώματα ευρεσιτεχνίας που κατέχει η Microsoft σε αυτούς τους τομείς ενδέχεται να έχουν αποτρέψει τις υλοποιήσεις του πλήρους πλαισίου από μη μέλη της Microsoft. Στις 3 Οκτωβρίου 2007, η Microsoft ανακοίνωσε ότι ο πηγαίος κώδικας για τις βιβλιοθήκες του .NET Framework 3.5 θα γινόταν διαθέσιμος υπό την άδεια χρήσης Microsoft Reference Source License (Ms-RSL). Το αποθετήριο πηγαίου κώδικα έγινε διαθέσιμο στο διαδίκτυο στις 16 Ιανουαρίου 2008 και περιλάμβανε τις βιβλιοθήκες BCL, ASP.NET, ADO.NET, Windows Forms, WPF και XML. Ο Scott Guthrie της Microsoft υποσχέθηκε ότι θα προστίθεντο

βιβλιοθήκες LINQ, WCF και WF. Οι παραλλαγές .NET Compact Framework και .NET Micro Framework του .NET Framework παρείχαν υποστήριξη για άλλες πλατφόρμες της Microsoft, όπως τα Windows Mobile, τα Windows CE και άλλες ενσωματωμένες συσκευές με περιορισμένους πόρους. Το Silverlight παρείχε υποστήριξη για προγράμματα περιήγησης ιστού μέσω plug-ins. Τον Νοέμβριο του 2014, η Microsoft παρουσίασε επίσης μια επικαιροποίηση των παραχωρήσεων διπλωμάτων ευρεσιτεχνίας, η οποία επεκτείνει περαιτέρω το πεδίο εφαρμογής πέραν των προηγούμενων δεσμεύσεων της. Προηγούμενα έργα όπως το Mono βρίσκονταν σε μια νομική γκρίζα ζώνη, επειδή οι προηγούμενες παραχωρήσεις της Microsoft αφορούσαν μόνο την τεχνολογία στις "καλυπτόμενες προδιαγραφές", συμπεριλαμβανομένων αυστηρά των 4ων εκδόσεων των ECMA-334 και ECMA-335. Ωστόσο, η νέα υπόσχεση διπλώματος ευρεσιτεχνίας δεν θέτει ανώτατο όριο στην έκδοση των προδιαγραφών και επεκτείνεται ακόμη και σε οποιεσδήποτε τεχνολογίες χρόνου εκτέλεσης .NET που τεκμηριώνονται στο MSDN και δεν έχουν καθοριστεί επίσημα από την ομάδα ECMA, εάν ένα έργο επιλέξει να τις υλοποιήσει. Αυτό επιτρέπει στο Mono και σε άλλα έργα να διατηρούν την ισοτιμία των χαρακτηριστικών με τα σύγχρονα χαρακτηριστικά του .NET που έχουν εισαχθεί από τη δημοσίευση της 4ης έκδοσης χωρίς να κινδυνεύουν από δικαστικές διαμάχες για διπλώματα ευρεσιτεχνίας σχετικά με την υλοποίηση αυτών των χαρακτηριστικών. Η νέα παραχώρηση διατηρεί τον περιορισμό ότι κάθε υλοποίηση πρέπει να διατηρεί ελάχιστη συμμόρφωση με τα υποχρεωτικά μέρη των προδιαγραφών CLI. Στις 31 Μαρτίου 2016, η Microsoft ανακοίνωσε στο Microsoft Build ότι θα παραχωρήσει εκ νέου πλήρη άδεια χρήσης του Mono με άδεια MIT, ακόμη και σε περιπτώσεις όπου προηγουμένως χρειαζόταν εμπορική άδεια χρήσης. Η Microsoft συμπλήρωσε επίσης την προηγούμενη υπόσχεσή της για τις πατέντες για το Mono, δηλώνοντας ότι δεν θα διεκδικήσει καμία "εφαρμοστέα πατέντα" κατά των μερών που "χρησιμοποιούν, πωλούν, προσφέρουν προς πώληση, εισάγουν ή διανέμουν το Mono". Ανακοινώθηκε ότι το έργο Mono εισφέρθηκε στο .NET Foundation. Οι εξελίξεις αυτές ακολούθησαν την εξαγορά της Xamarin, η οποία ξεκίνησε τον Φεβρουάριο του 2016 και ολοκληρώθηκε στις 18 Μαρτίου 2016. Στο δελτίο τύπου της Microsoft τονίζεται ότι η δέσμευση για διαπλατφορμική χρήση επιτρέπει πλέον μια πλήρως ανοιχτού κώδικα, σύγχρονη στοίβα .NET για την πλευρά του διακομιστή. Η Microsoft κυκλοφόρησε τον πηγαίο κώδικα για τα WPF, Windows Forms και WinUI στις 4 Δεκεμβρίου 2018.

### 3. Microsoft Visual Studio

Το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων υπολογιστών, καθώς και ιστοσελίδων, εφαρμογών ιστού, υπηρεσιών ιστού και εφαρμογών για κινητά τηλέφωνα. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft, όπως Windows API, Windows Forms, Windows Presentation Foundation, Windows Store και Microsoft Silverlight. Μπορεί να παράγει τόσο εγγενή κώδικα όσο και διαχειριζόμενο κώδικα. Το Visual Studio περιλαμβάνει έναν επεξεργαστή κώδικα που υποστηρίζει το IntelliSense (το στοιχείο συμπλήρωσης κώδικα) καθώς και την αναδιαμόρφωση κώδικα.

Ο ενσωματωμένος αποσφαλματωτής λειτουργεί τόσο ως αποσφαλματωτής σε επίπεδο πηγής όσο και ως αποσφαλματωτής σε επίπεδο μηχανής. Άλλα ενσωματωμένα εργαλεία περιλαμβάνουν έναν αναλυτή κώδικα (code profiler), έναν σχεδιαστή για τη δημιουργία εφαρμογών GUI, έναν σχεδιαστή ιστοσελίδων, έναν σχεδιαστή κλάσεων και έναν σχεδιαστή σχημάτων βάσεων δεδομένων. Δέχεται πρόσθετα που επεκτείνουν τη λειτουργικότητα σχεδόν σε κάθε επίπεδο -συμπεριλαμβανομένης της προσθήκης υποστήριξης για συστήματα ελέγχου πηγής (όπως το Subversion και το Git) και της προσθήκης νέων συνόλων εργαλείων, όπως επεξεργαστές και οπτικοί σχεδιαστές για γλώσσες ειδικού τομέα ή σύνολα εργαλείων για άλλες πτυχές του κύκλου ζωής της ανάπτυξης λογισμικού (όπως το πρόγραμμα-πελάτης Azure DevOps: Team Explorer). Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει στον επεξεργαστή κώδικα και τον αποσφαλματωτή να υποστηρίζουν (σε διαφορετικό βαθμό) σχεδόν κάθε γλώσσα προγραμματισμού, εφόσον υπάρχει μια υπηρεσία που αφορά τη συγκεκριμένη γλώσσα. Οι ενσωματωμένες γλώσσες περιλαμβάνουν τη C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML και CSS. Υποστήριξη για άλλες γλώσσες, όπως η Python, Ruby, Node.js και M μεταξύ άλλων, είναι διαθέσιμες μέσω πρόσθετων προγραμμάτων. Η Java (και η J#) υποστηρίζονταν στο παρελθόν. Η πιο βασική έκδοση του Visual Studio, η έκδοση Community, διατίθεται δωρεάν. Το σλόγκαν της έκδοσης Visual Studio Community είναι "Δωρεάν, πλήρως εξοπλισμένο IDE για φοιτητές, προγραμματιστές ανοικτού κώδικα και μεμονωμένους προγραμματιστές". Από τις 8 Νοεμβρίου 2021, το Visual Studio 2022 είναι μια τρέχουσα έκδοση έτοιμη για παραγωγή, ενώ οι παλαιότερες εκδόσεις, όπως το 2013 και το 2015, βρίσκονται σε εκτεταμένη υποστήριξη και το 2017 και το 2019 σε κύρια υποστήριξη.



**Εικόνα 5.** Εικόνα του Microsoft Visual Studio

Πηγή: <https://visualstudio.microsoft.com/>

### 3.1 Αρχιτεκτονική

Το Visual Studio δεν υποστηρίζει εγγενώς καμία γλώσσα προγραμματισμού, λύση ή εργαλείο- αντίθετα, επιτρέπει τη σύνδεση λειτουργιών που έχουν κωδικοποιηθεί ως VSPackage. Όταν εγκατασταθεί, η λειτουργικότητα είναι διαθέσιμη ως *υπηρεσία*. Το IDE παρέχει τρεις υπηρεσίες: SVsSolution, η οποία παρέχει τη δυνατότητα απαρίθμησης έργων και λύσεων- SVsUIShell, η οποία παρέχει λειτουργικότητα παραθύρων και UI (συμπεριλαμβανομένων καρτελών, γραμμών εργαλείων και παραθύρων εργαλείων)- και SVsShell, η οποία ασχολείται με την καταχώριση των πακέτων VSPackages. Επιπλέον, το IDE είναι επίσης υπεύθυνο για το συντονισμό και την ενεργοποίηση της επικοινωνίας μεταξύ των υπηρεσιών. Όλοι οι επεξεργαστές, οι σχεδιαστές, οι τύποι έργων και άλλα εργαλεία υλοποιούνται ως πακέτα VSPackages. Το Visual Studio χρησιμοποιεί COM για την πρόσβαση στα VSPackages. Το SDK του Visual Studio περιλαμβάνει επίσης το *Managed Package Framework (MPF)*, το οποίο είναι ένα σύνολο διαχειριζόμενων περιτυλίξεων γύρω από τις διεπαφές COM που επιτρέπουν τη συγγραφή των πακέτων σε οποιαδήποτε γλώσσα συμβατή με το CLI. Ωστόσο, το MPF δεν παρέχει όλες τις λειτουργίες που εκτίθενται από τις διεπαφές COM του Visual Studio. Οι υπηρεσίες μπορούν στη συνέχεια να καταναλωθούν για τη δημιουργία άλλων πακέτων, τα οποία προσθέτουν λειτουργικότητα στο Visual Studio IDE. Η υποστήριξη για γλώσσες προγραμματισμού προστίθεται με τη χρήση ενός ειδικού πακέτου VSPackage που ονομάζεται *Υπηρεσία γλώσσας*. Μια υπηρεσία γλώσσας ορίζει διάφορες διεπαφές τις οποίες η υλοποίηση του πακέτου VSPackage μπορεί να υλοποιήσει για να προσθέσει υποστήριξη για διάφορες λειτουργίες. Οι λειτουργίες που μπορούν να προστεθούν με αυτόν τον τρόπο περιλαμβάνουν χρωματισμό σύνταξης, συμπλήρωση εντολών, αντιστοίχιση παρενθέσεων, συμβουλές πληροφοριών παραμέτρων, λίστες μελών και δείκτες σφαλμάτων για μεταγλώττιση στο παρασκήνιο. Εάν η διεπαφή υλοποιηθεί, η λειτουργικότητα θα είναι διαθέσιμη για τη γλώσσα. Οι γλωσσικές υπηρεσίες υλοποιούνται ανά γλώσσα. Οι υλοποιήσεις μπορούν να επαναχρησιμοποιούν κώδικα από τον αναλυτή ή τον μεταγλωττιστή για τη γλώσσα. Οι γλωσσικές υπηρεσίες μπορούν να υλοποιηθούν είτε σε εγγενή κώδικα είτε σε διαχειριζόμενο κώδικα. Για εγγενή κώδικα, μπορούν να χρησιμοποιηθούν είτε οι εγγενείς διεπαφές COM είτε το Babel Framework (μέρος του Visual Studio SDK). Για διαχειριζόμενο κώδικα, το MPF περιλαμβάνει περιτύλιγμα για τη συγγραφή διαχειριζόμενων γλωσσικών υπηρεσιών. Το Visual Studio δεν περιλαμβάνει ενσωματωμένη υποστήριξη ελέγχου πηγαιού κώδικα, αλλά ορίζει δύο εναλλακτικούς τρόπους ενσωμάτωσης των συστημάτων ελέγχου πηγαιού κώδικα στο IDE. Ένα πακέτο VSPackage ελέγχου πηγαιού μπορεί να παρέχει τη δική του προσαρμοσμένη διεπαφή χρήστη. Αντίθετα, ένα πρόσθετο ελέγχου πηγαιού που χρησιμοποιεί το MSSCCI (Microsoft Source Code Control Interface) παρέχει ένα σύνολο λειτουργιών που χρησιμοποιούνται για την υλοποίηση διαφόρων λειτουργιών ελέγχου πηγαιού, με μια τυπική διεπαφή χρήστη του Visual Studio. Το MSSCCI χρησιμοποιήθηκε αρχικά για την ενσωμάτωση του Visual SourceSafe στο Visual Studio 6.0, αλλά αργότερα ανοίχτηκε μέσω του Visual Studio SDK. Το Visual Studio .NET 2002 χρησιμοποίησε το MSSCCI 1.1 και το Visual Studio .NET 2003 χρησιμοποίησε το MSSCCI 1.2. Τα Visual Studio 2005, 2008 και 2010 χρησιμοποιούν την έκδοση 1.3 του



MSSCCI, η οποία προσθέτει υποστήριξη για διάδοση μετονομασίας και διαγραφής, καθώς και ασύγχρονο άνοιγμα. Το Visual Studio υποστηρίζει την εκτέλεση πολλαπλών περιπτώσεων του περιβάλλοντος (η καθεμία με το δικό της σύνολο πακέτων VSPackages). Οι περιπτώσεις χρησιμοποιούν διαφορετικές κυψέλες μητρώου (βλ. τον ορισμό του όρου "κυψέλη μητρώου" στο MSDN με την έννοια που χρησιμοποιείται εδώ) για να αποθηκεύουν την κατάσταση διαμόρφωσής τους και διαφοροποιούνται από το AppId (αναγνωριστικό εφαρμογής). Οι περιπτώσεις εκκινούνται από ένα ειδικό για το AppId .exe που επιλέγει το AppId, ορίζει τη ριζική κυψέλη και εκκινεί το IDE. Τα πακέτα VSPackages που έχουν καταχωρηθεί για ένα AppId ενσωματώνονται με άλλα πακέτα VSPackages για το συγκεκριμένο AppId. Οι διάφορες εκδόσεις προϊόντων του Visual Studio δημιουργούνται χρησιμοποιώντας τα διαφορετικά AppIds. Τα προϊόντα της έκδοσης Visual Studio Express εγκαθίστανται με τα δικά τους AppIds, αλλά τα προϊόντα Standard, Professional και Team Suite μοιράζονται το ίδιο AppId. Κατά συνέπεια, μπορεί κανείς να εγκαταστήσει τις εκδόσεις Express δίπλα-δίπλα με άλλες εκδόσεις, σε αντίθεση με τις άλλες εκδόσεις που ενημερώνουν την ίδια εγκατάσταση. Η επαγγελματική έκδοση περιλαμβάνει ένα υπερσύνολο των πακέτων VSPackages της τυπικής έκδοσης και η σουίτα Team περιλαμβάνει ένα υπερσύνολο των πακέτων VSPackages και των δύο άλλων εκδόσεων. Το σύστημα AppId αξιοποιείται από το Visual Studio Shell στο Visual Studio 2008.

## 3.2 Χαρακτηριστικά

### 3.2.1 Επεξεργαστής κώδικα

Το Visual Studio περιλαμβάνει έναν επεξεργαστή κώδικα που υποστηρίζει επισήμανση συντακτικού και συμπλήρωση κώδικα με τη χρήση IntelliSense για μεταβλητές, συναρτήσεις, μεθόδους, βρόχους και ερωτήματα LINQ. Το IntelliSense υποστηρίζεται για τις γλώσσες που περιλαμβάνονται, καθώς και για XML, Cascading Style Sheets και JavaScript κατά την ανάπτυξη ιστότοπων και εφαρμογών ιστού. Οι προτάσεις αυτόματης συμπλήρωσης εμφανίζονται σε ένα πλαίσιο λίστας χωρίς μοντέλο πάνω από το παράθυρο του επεξεργαστή κώδικα, κοντά στο δρομέα επεξεργασίας. Στο Visual Studio 2008 και μετά, μπορεί να γίνει προσωρινά ημι-διαφανές για να βλέπετε τον κώδικα που εμποδίζεται από αυτό. Ο επεξεργαστής κώδικα χρησιμοποιείται για όλες τις υποστηριζόμενες γλώσσες. Ο επεξεργαστής κώδικα του Visual Studio υποστηρίζει επίσης τον καθορισμό σελιδοδεικτών στον κώδικα για γρήγορη πλοήγηση. Άλλα βοηθήματα πλοήγησης περιλαμβάνουν την σύμπτυξη των μπλοκ κώδικα και τη σταδιακή αναζήτηση, εκτός από την κανονική αναζήτηση κειμένου και την αναζήτηση με regex. Ο επεξεργαστής κώδικα περιλαμβάνει επίσης ένα πρόχειρο πολλαπλών στοιχείων και μια λίστα εργασιών. Ο επεξεργαστής κώδικα υποστηρίζει αποσπάσματα κώδικα, τα οποία είναι αποθηκευμένα πρότυπα για επαναλαμβανόμενο κώδικα και μπορούν να εισαχθούν στον κώδικα και να προσαρμοστούν για το έργο που επεξεργάζεται. Έχει ενσωματωθεί επίσης ένα εργαλείο διαχείρισης για τα αποσπάσματα κώδικα. Αυτά τα εργαλεία εμφανίζονται ως αιωρούμενα παράθυρα, τα οποία μπορούν να ρυθμιστούν ώστε να

αποκρύπτονται αυτόματα όταν δεν χρησιμοποιούνται ή να αγκιστρώνονται στο πλάι της οθόνης. Ο επεξεργαστής κώδικα του Visual Studio υποστηρίζει επίσης την αναδιοργάνωση κώδικα, συμπεριλαμβανομένης της αναδιάταξης παραμέτρων, της μετονομασίας μεταβλητών και μεθόδων, της εξαγωγής διασυνδέσεων και της ενθυλάκωσης μελών κλάσεων μέσα σε ιδιότητες, μεταξύ άλλων.

### 3.2.2 Αποσφαλματωτής

Το Visual Studio περιλαμβάνει ένα πρόγραμμα εντοπισμού σφαλμάτων που λειτουργεί τόσο ως πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο πηγής όσο και ως πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο μηχανής. Λειτουργεί τόσο με διαχειριζόμενο κώδικα όσο και με εγγενή κώδικα και μπορεί να χρησιμοποιηθεί για την αποσφαλμάτωση εφαρμογών γραμμένων σε οποιαδήποτε γλώσσα υποστηρίζεται από το Visual Studio. Επιπλέον, μπορεί επίσης να συνδεθεί σε διεργασίες που εκτελούνται, να παρακολουθεί και να αποσφαλματώνει αυτές τις διεργασίες. Εάν είναι διαθέσιμος ο πηγαίος κώδικας για την εκτελούμενη διεργασία, εμφανίζει τον κώδικα καθώς εκτελείται. Εάν ο πηγαίος κώδικας δεν είναι διαθέσιμος, μπορεί να εμφανίσει την αποσυναρμολόγηση. Το πρόγραμμα εντοπισμού σφαλμάτων του Visual Studio μπορεί επίσης να δημιουργήσει απολήψεις μνήμης καθώς και να τις φορτώσει αργότερα για αποσφαλμάτωση. Υποστηρίζονται επίσης προγράμματα πολλαπλών νημάτων. Ο αποσφαλματωτής μπορεί να ρυθμιστεί ώστε να ξεκινά όταν μια εφαρμογή που εκτελείται εκτός του περιβάλλοντος του Visual Studio καταρρέει. Ο αποσφαλματωτής του Visual Studio επιτρέπει τον καθορισμό σημείων διακοπής (τα οποία επιτρέπουν την προσωρινή διακοπή της εκτέλεσης σε μια συγκεκριμένη θέση) και παρακολουθήσεων (οι οποίες παρακολουθούν τις τιμές των μεταβλητών καθώς εξελίσσεται η εκτέλεση). Τα σημεία διακοπής μπορούν να είναι υπό όρους, δηλαδή ενεργοποιούνται όταν ικανοποιείται η συνθήκη. Ο κώδικας μπορεί να περνάει με βήματα, δηλαδή να εκτελείται μία γραμμή (του πηγαίου κώδικα) κάθε φορά. Μπορεί είτε να βηματίσει σε συναρτήσεις για να κάνει αποσφαλμάτωση στο εσωτερικό τους, είτε να βηματίσει πάνω από αυτές, δηλ. η εκτέλεση του σώματος της συνάρτησης δεν είναι διαθέσιμη για χειροκίνητη επιθεώρηση. Ο αποσφαλματωτής υποστηρίζει την επεξεργασία και τη συνέχιση, δηλαδή επιτρέπει την επεξεργασία του κώδικα κατά την αποσφαλμάτωσή του. Κατά την αποσφαλμάτωση, εάν ο δείκτης του ποντικιού αιωρείται πάνω από οποιαδήποτε μεταβλητή, η τρέχουσα τιμή της εμφανίζεται σε μια συμβουλή εργαλείων ("data tooltips"), όπου μπορεί επίσης να τροποποιηθεί εάν είναι επιθυμητό. Κατά τη διάρκεια της κωδικοποίησης, το πρόγραμμα εντοπισμού σφαλμάτων του Visual Studio επιτρέπει τη χειροκίνητη κλήση ορισμένων λειτουργιών από το παράθυρο εργαλείων `Immediate`. Οι παράμετροι της μεθόδου παρέχονται στο παράθυρο Immediate.

### 3.2.3 Σχεδιαστής

Το Visual Studio περιλαμβάνει μια σειρά από οπτικούς σχεδιαστές που βοηθούν στην ανάπτυξη εφαρμογών. Αυτά τα εργαλεία περιλαμβάνουν:

#### 1) Σχεδιαστής Windows Forms

Ο σχεδιαστής Windows Forms χρησιμοποιείται για τη δημιουργία εφαρμογών GUI με χρήση των Windows Forms. Η διάταξη μπορεί να ελεγχθεί με τη στέγαση των στοιχείων ελέγχου μέσα σε άλλα δοχεία ή με το κλειδωμά τους στο πλάι της φόρμας. Τα στοιχεία ελέγχου που εμφανίζουν δεδομένα (όπως το πλαίσιο κειμένου, το πλαίσιο λίστας και η προβολή πλέγματος) μπορούν να συνδεθούν με πηγές δεδομένων, όπως βάσεις δεδομένων ή ερωτήματα. Τα στοιχεία ελέγχου που είναι συνδεδεμένα με δεδομένα μπορούν να δημιουργηθούν σύροντας στοιχεία από το παράθυρο Data Sources (Πηγές δεδομένων) σε μια επιφάνεια σχεδίασης. Το UI συνδέεται με τον κώδικα χρησιμοποιώντας ένα μοντέλο προγραμματισμού με γνώμονα το συμβάν. Ο σχεδιαστής παράγει είτε κώδικα C# είτε κώδικα VB.NET για την εφαρμογή.

#### 2) Σχεδιαστής WPF

Ο σχεδιαστής WPF, με την κωδική ονομασία *Cider*, εισήχθη με το Visual Studio 2008. Όπως και ο σχεδιαστής Windows Forms υποστηρίζει τη μεταφορά drag and drop. Χρησιμοποιείται για τη συγγραφή διεπαφών χρήστη με στόχο το Windows Presentation Foundation. Υποστηρίζει όλες τις λειτουργίες του WPF, συμπεριλαμβανομένης της δέσμευσης δεδομένων και της αυτόματης διαχείρισης διάταξης. Παράγει κώδικα XAML για το UI. Το παραγόμενο αρχείο XAML είναι συμβατό με το Microsoft Expression Design, το προϊόν που είναι προσανατολισμένο στον σχεδιαστή. Ο κώδικας XAML συνδέεται με τον κώδικα χρησιμοποιώντας ένα μοντέλο code-behind.

#### 3) Σχεδιαστής/ανάπτυξη ιστοσελίδων

Το Visual Studio περιλαμβάνει επίσης έναν επεξεργαστή και σχεδιαστή ιστοσελίδων που επιτρέπει τη συγγραφή ιστοσελίδων με drag and drop widgets. Χρησιμοποιείται για την ανάπτυξη εφαρμογών ASP.NET και υποστηρίζει HTML, CSS και JavaScript. Χρησιμοποιεί ένα μοντέλο code-behind για τη σύνδεση με τον κώδικα ASP.NET. Από το Visual Studio 2008 και μετά, η μηχανή διάταξης που χρησιμοποιείται από τον σχεδιαστή ιστού είναι κοινή με το καταργηθέν Expression Web. Υπάρχει επίσης υποστήριξη ASP.NET MVC για την τεχνολογία MVC ως ξεχωριστή λήψη και το έργο ASP.NET Dynamic Data διαθέσιμο από τη Microsoft.

#### 4) Σχεδιαστής κλάσεων

Ο σχεδιαστής κλάσεων χρησιμοποιείται για τη συγγραφή και την επεξεργασία των

κλάσεων (συμπεριλαμβανομένων των μελών της και της πρόσβασής τους) με τη χρήση μοντελοποίησης UML. Ο σχεδιαστής κλάσεων μπορεί να δημιουργήσει περιγράμματα κώδικα C# και VB.NET για τις κλάσεις και τις μεθόδους. Μπορεί επίσης να δημιουργήσει διαγράμματα κλάσεων από χειρόγραφες κλάσεις.

### 5) Σχεδιαστής δεδομένων

Ο σχεδιαστής δεδομένων μπορεί να χρησιμοποιηθεί για τη γραφική επεξεργασία σχημάτων βάσεων δεδομένων, συμπεριλαμβανομένων τυποποιημένων πινάκων, πρωτεύοντων και ξένων κλειδιών και περιορισμών. Μπορεί επίσης να χρησιμοποιηθεί για το σχεδιασμό ερωτημάτων από τη γραφική προβολή.

### 6) Σχεδιαστής χαρτογράφησης

Από το Visual Studio 2008 και μετά, ο σχεδιαστής χαρτογράφησης χρησιμοποιείται από το LINQ σε SQL για να σχεδιάζει την αντιστοίχιση μεταξύ των σχημάτων βάσεων δεδομένων και των κλάσεων που ενθυλακώνουν τα δεδομένα. Η νέα λύση από την προσέγγιση ORM, το ADO.NET Entity Framework, αντικαθιστά και βελτιώνει την παλιά τεχνολογία.

## 3.2.4 Άλλα εργαλεία

### 1) Πρόγραμμα περιήγησης ανοικτών καρτελών

Το πρόγραμμα περιήγησης ανοικτών καρτελών χρησιμοποιείται για τη λίστα όλων των ανοικτών καρτελών και την εναλλαγή μεταξύ τους. Η κλήση του γίνεται με τη χρήση των πλήκτρων CTRL+TAB.

### 2) Επεξεργαστής Ιδιοτήτων

Το εργαλείο *Properties Editor* χρησιμοποιείται για την επεξεργασία ιδιοτήτων σε ένα παράθυρο GUI μέσα στο Visual Studio. Απαριθμεί όλες τις διαθέσιμες ιδιότητες (τόσο αυτές που είναι μόνο για ανάγνωση όσο και αυτές που μπορούν να οριστούν) για όλα τα αντικείμενα, συμπεριλαμβανομένων των κλάσεων, των φορμών, των ιστοσελίδων και άλλων αντικειμένων.

### 3) Περιηγητής αντικειμένων

Το Object Browser είναι ένα πρόγραμμα περιήγησης χώρου ονομάτων και βιβλιοθήκης κλάσεων για το Microsoft .NET. Μπορεί να χρησιμοποιηθεί για την περιήγηση στους χώρους ονομάτων (οι οποίοι είναι διατεταγμένοι ιεραρχικά) σε



διαχειριζόμενες συγκροτήσεις. Η ιεραρχία μπορεί να αντικατοπτρίζει ή όχι την οργάνωση στο σύστημα αρχείων.

#### 4) Εξερεύνηση λύσεων

Στην ορολογία του Visual Studio, μια λύση είναι ένα σύνολο αρχείων κώδικα και άλλων πόρων που χρησιμοποιούνται για τη δημιουργία μιας εφαρμογής. Τα αρχεία μιας λύσης είναι ιεραρχικά διατεταγμένα, γεγονός που μπορεί να αντικατοπτρίζει ή να μην αντικατοπτρίζει την οργάνωση στο σύστημα αρχείων. Η *Εξερεύνηση λύσης* χρησιμοποιείται για τη διαχείριση και την περιήγηση στα αρχεία μιας λύσης.

#### 5) Εξερευνητής ομάδων

Ο *Team Explorer* χρησιμοποιείται για την ενσωμάτωση των δυνατοτήτων του Azure DevOps (είτε Azure DevOps Services είτε Azure DevOps Server) στο IDE . Εκτός από την ενσωμάτωση του ελέγχου εκδόσεων παρέχει τη δυνατότητα προβολής και διαχείρισης μεμονωμένων στοιχείων εργασίας (συμπεριλαμβανομένων ιστοριών χρηστών, σφαλμάτων, εργασιών και άλλων εγγράφων). Περιλαμβάνεται ως μέρος της εγκατάστασης του Visual Studio και είναι επίσης διαθέσιμο ως αυτόνομη λήψη.

#### 6) Εξερεύνηση δεδομένων

Ο *Data Explorer* χρησιμοποιείται για τη διαχείριση βάσεων δεδομένων σε περιπτώσεις Microsoft SQL Server. Επιτρέπει τη δημιουργία και την τροποποίηση πινάκων βάσεων δεδομένων (είτε με την έκδοση εντολών T-SQL είτε με τη χρήση του σχεδιαστή δεδομένων). Μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία ερωτημάτων και αποθηκευμένων διαδικασιών, με τις τελευταίες είτε σε T-SQL είτε σε διαχειριζόμενο κώδικα μέσω της SQL CLR. Διατίθεται επίσης υποστήριξη εντοπισμού σφαλμάτων και IntelliSense.

#### 7) Server Explorer

Το εργαλείο *Server Explorer* χρησιμοποιείται για τη διαχείριση των συνδέσεων βάσεων δεδομένων σε έναν προσβάσιμο υπολογιστή. Χρησιμοποιείται επίσης για την περιήγηση στις εκτελούμενες υπηρεσίες των Windows, στους μετρητές επιδόσεων, στο Windows Event Log και στις ουρές μηνυμάτων και τη χρήση τους ως πηγή δεδομένων.

#### 8) Πλαίσιο δημιουργίας κειμένου

Το Visual Studio περιλαμβάνει ένα πλήρες πλαίσιο δημιουργίας κειμένου με την ονομασία T4, το οποίο επιτρέπει στο Visual Studio να δημιουργεί αρχεία κειμένου από

πρότυπα είτε στο IDE είτε μέσω κώδικα.

## 9) Εργαλείο διαχείρισης ιστοτόπων ASP.NET

Το Εργαλείο διαχείρισης ιστοτόπων ASP.NET επιτρέπει τη διαμόρφωση ιστοτόπων ASP.NET.

## 10) Εργαλεία του Visual Studio για το Office

Το Visual Studio Tools for Office είναι ένα SDK και ένα πρόσθετο για το Visual Studio που περιλαμβάνει εργαλεία για την ανάπτυξη για τη σουίτα Microsoft Office. Προηγουμένως (για τα Visual Studio .NET 2003 και Visual Studio 2005) ήταν ένα ξεχωριστό SKU που υποστήριζε μόνο τις γλώσσες Visual C# και Visual Basic ή περιλαμβανόταν στην Team Suite. Με το Visual Studio 2008, δεν αποτελεί πλέον ξεχωριστή SKU, αλλά περιλαμβάνεται στην έκδοση Professional και στις υψηλότερες εκδόσεις. Για την ανάπτυξη λύσεων VSTO απαιτείται ξεχωριστό runtime.

## 3.3 Επεκτασιμότητα

Το Visual Studio επιτρέπει στους προγραμματιστές να γράφουν επεκτάσεις για το Visual Studio ώστε να επεκτείνουν τις δυνατότητές του. Αυτές οι επεκτάσεις "συνδέονται" με το Visual Studio και επεκτείνουν τη λειτουργικότητά του. Οι επεκτάσεις έχουν τη μορφή *μακροεντολών, πρόσθετων προγραμμάτων και πακέτων*. Οι μακροεντολές αντιπροσωπεύουν επαναλαμβανόμενες εργασίες και ενέργειες που οι προγραμματιστές μπορούν να καταγράψουν προγραμματιστικά για αποθήκευση, αναπαραγωγή και διανομή. Οι μακροεντολές, ωστόσο, δεν μπορούν να υλοποιήσουν νέες εντολές ή να δημιουργήσουν παράθυρα εργαλείων. Γράφονται με τη χρήση της Visual Basic και δεν μεταγλωττίζονται. Τα Add-Ins παρέχουν πρόσβαση στο μοντέλο αντικειμένων του Visual Studio και μπορούν να αλληλεπιδρούν με τα εργαλεία του IDE. Τα Add-Ins μπορούν να χρησιμοποιηθούν για την υλοποίηση νέων λειτουργιών και μπορούν να προσθέσουν νέα παράθυρα εργαλείων. Τα Add-Ins συνδέονται στο IDE μέσω COM και μπορούν να δημιουργηθούν σε οποιαδήποτε γλώσσα συμβατή με COM. Τα πακέτα δημιουργούνται χρησιμοποιώντας το Visual Studio SDK και παρέχουν το υψηλότερο επίπεδο επεκτασιμότητας. Μπορούν να δημιουργήσουν σχεδιαστές και άλλα εργαλεία, καθώς και να ενσωματώσουν άλλες γλώσσες προγραμματισμού. Το Visual Studio SDK παρέχει μη διαχειριζόμενα API καθώς και ένα διαχειριζόμενο API για την εκτέλεση αυτών των εργασιών. Ωστόσο, το διαχειριζόμενο API δεν είναι τόσο ολοκληρωμένο όσο το μη διαχειριζόμενο. Οι επεκτάσεις υποστηρίζονται στις εκδόσεις Standard (και νεότερες) του Visual Studio 2005. Οι εκδόσεις Express δεν υποστηρίζουν επεκτάσεις φιλοξενίας. Το Visual Studio 2008 εισήγαγε το Visual Studio Shell που επιτρέπει την ανάπτυξη μιας προσαρμοσμένης έκδοσης του IDE. Το Visual Studio Shell ορίζει ένα σύνολο πακέτων VSPackages που παρέχουν τη λειτουργικότητα που απαιτείται σε οποιοδήποτε IDE. Επιπλέον, μπορούν να προστεθούν και άλλα πακέτα για την προσαρμογή της

εγκατάστασης. Η απομονωμένη λειτουργία του κελύφους δημιουργεί ένα νέο AppId όπου εγκαθίστανται τα πακέτα. Αυτά πρέπει να εκκινούνται με ένα διαφορετικό εκτελέσιμο πρόγραμμα. Στοχεύει στην ανάπτυξη προσαρμοσμένων περιβαλλόντων ανάπτυξης, είτε για μια συγκεκριμένη γλώσσα είτε για ένα συγκεκριμένο σενάριο. Η λειτουργία Integrated εγκαθιστά τα πακέτα στο AppId των εκδόσεων Professional/Standard/Team System, έτσι ώστε τα εργαλεία να ενσωματώνονται σε αυτές τις εκδόσεις. Το Visual Studio Shell διατίθεται ως δωρεάν λήψη. Μετά την κυκλοφορία του Visual Studio 2008, η Microsoft δημιούργησε το Visual Studio Gallery. Χρησιμεύει ως κεντρική τοποθεσία για τη δημοσίευση πληροφοριών σχετικά με τις επεκτάσεις του Visual Studio. Οι κοινοτικοί προγραμματιστές καθώς και οι εμπορικοί προγραμματιστές μπορούν να ανεβάζουν πληροφορίες σχετικά με τις επεκτάσεις τους στο Visual Studio .NET 2002 έως το Visual Studio 2010. Οι χρήστες του ιστότοπου μπορούν να βαθμολογούν και να αξιολογούν τις επεκτάσεις για να βοηθήσουν στην αξιολόγηση της ποιότητας των επεκτάσεων που αναρτώνται. Μια επέκταση αποθηκεύεται σε ένα αρχείο VSIX. Εσωτερικά ένα αρχείο VSIX είναι ένα αρχείο ZIP που περιέχει ορισμένα αρχεία XML και ενδεχομένως ένα ή περισσότερα DLL. Ένα από τα κύρια πλεονεκτήματα αυτών των επεκτάσεων είναι ότι δεν απαιτούν δικαιώματα διαχειριστή για την εγκατάστασή τους.

### 3.4 Υποστηριζόμενα προϊόντα

#### 1) Microsoft Visual C#

Το Microsoft Visual C#, η υλοποίηση της γλώσσας C# από τη Microsoft, στοχεύει στο .NET Framework, μαζί με τις υπηρεσίες γλώσσας που επιτρέπουν στο Visual Studio IDE να υποστηρίζει έργα C#. Ενώ οι υπηρεσίες γλώσσας αποτελούν μέρος του Visual Studio, ο μεταγλωττιστής διατίθεται ξεχωριστά ως μέρος του .NET Framework. Οι μεταγλωττιστές Visual C# 2008, 2010 και 2012 υποστηρίζουν τις εκδόσεις 3.0, 4.0 και 5.0 των προδιαγραφών της γλώσσας C#, αντίστοιχα. Η Visual C# υποστηρίζει, μεταξύ άλλων, τον σχεδιαστή κλάσεων του Visual Studio, τον σχεδιαστή φορμών και τον σχεδιαστή δεδομένων.

#### 2) Microsoft Visual C++

Το Microsoft Visual C++ είναι η μερική υλοποίηση της Microsoft του μεταγλωττιστή C και η πλήρης υλοποίηση του μεταγλωττιστή C++ και των σχετικών γλωσσών-υπηρεσιών και ειδικών εργαλείων για την ενσωμάτωση με το Visual Studio IDE. Μπορεί να κάνει μεταγλώττιση είτε σε λειτουργία C είτε σε λειτουργία C++. Για τη C++, από την έκδοση 15.7 συμμορφώνεται με την C++17. Η υλοποίηση C του Visual Studio 2015 εξακολουθεί να μην υποστηρίζει το πλήρες πρότυπο-συγκεκριμένα, η επικεφαλίδα σύνθετων αριθμών `complex.h` που εισήχθη στην C99 δεν υποστηρίζεται. Το Visual C++ υποστηρίζει την προδιαγραφή C++/CLI για τη συγγραφή διαχειριζόμενου κώδικα, καθώς

και κώδικα μικτής λειτουργίας (ένα μείγμα εγγενούς και διαχειριζόμενου κώδικα). Η Microsoft τοποθετεί το Visual C++ για ανάπτυξη σε εγγενή κώδικα ή σε κώδικα που περιέχει τόσο εγγενή όσο και διαχειριζόμενα στοιχεία. Η Visual C++ υποστηρίζει την COM καθώς και τη βιβλιοθήκη MFC. Για την ανάπτυξη MFC, παρέχει μια σειρά από οδηγούς για τη δημιουργία και την προσαρμογή του boilerplate κώδικα της MFC και τη δημιουργία εφαρμογών GUI με χρήση της MFC. Η Visual C++ μπορεί επίσης να χρησιμοποιήσει τον σχεδιαστή φορμών του Visual Studio για να σχεδιάσει γραφικά το UI. Η Visual C++ μπορεί επίσης να χρησιμοποιηθεί με το API των Windows. Υποστηρίζει επίσης τη χρήση *εγγενών συναρτήσεων*, οι οποίες είναι συναρτήσεις που αναγνωρίζονται από τον ίδιο τον μεταγλωττιστή και δεν υλοποιούνται ως βιβλιοθήκη. Οι ενδογενείς συναρτήσεις χρησιμοποιούνται για την έκθεση του συνόλου εντολών SSE των σύγχρονων CPUs. Η Visual C++ περιλαμβάνει επίσης την προδιαγραφή OpenMP (έκδοση 2.0).

### 3) Microsoft Visual Basic

Η Microsoft Visual Basic είναι η υλοποίηση της γλώσσας VB.NET και των σχετικών εργαλείων και γλωσσικών υπηρεσιών της Microsoft. Εισήχθη με το Visual Studio .NET (2002). Η Microsoft έχει τοποθετήσει τη Visual Basic για την ταχεία ανάπτυξη εφαρμογών. Η Visual Basic μπορεί να χρησιμοποιηθεί για τη συγγραφή τόσο εφαρμογών κονσόλας όσο και εφαρμογών GUI. Όπως και η Visual C#, η Visual Basic υποστηρίζει επίσης, μεταξύ άλλων, τον σχεδιαστή κλάσεων του Visual Studio, τον σχεδιαστή φορμών και τον σχεδιαστή δεδομένων. Όπως και η C#, ο μεταγλωττιστής VB.NET είναι επίσης διαθέσιμος ως μέρος του .NET Framework, αλλά οι υπηρεσίες γλώσσας που επιτρέπουν την ανάπτυξη έργων VB.NET με το Visual Studio, είναι διαθέσιμες ως μέρος του τελευταίου.

### 4) Microsoft Visual Web Developer

Το Microsoft Visual Web Developer χρησιμοποιείται για τη δημιουργία δικτυακών τόπων, εφαρμογών ιστού και υπηρεσιών ιστού με χρήση ASP.NET. Μπορούν να χρησιμοποιηθούν είτε οι γλώσσες C# είτε VB.NET. Το Visual Web Developer μπορεί να χρησιμοποιήσει το Visual Studio Web Designer για να σχεδιάσει γραφικά τις διατάξεις των ιστοσελίδων.

### 5) Azure DevOps

Το Azure DevOps προορίζεται για συνεργατικά έργα ανάπτυξης λογισμικού και παρέχει έλεγχο εκδόσεων, προγραμματισμό και παρακολούθηση εργασιών, συλλογή δεδομένων και υποβολή εκθέσεων. Περιλαμβάνει επίσης το *Team Explorer* το οποίο είναι ενσωματωμένο μέσα στο Visual Studio.

## 3.5 Εκδόσεις

Το Microsoft Visual Studio διατίθεται στις ακόλουθες εκδόσεις ή SKU:

### 1) Community

Η έκδοση Community ανακοινώθηκε στις 12 Νοεμβρίου 2014, ως μια νέα δωρεάν έκδοση, με παρόμοια λειτουργικότητα με το Visual Studio Professional. Πριν από αυτή την ημερομηνία, οι μόνες δωρεάν εκδόσεις του Visual Studio ήταν οι περιορισμένων δυνατοτήτων παραλλαγές Express. Σε αντίθεση με τις παραλλαγές Express, το Visual Studio Community υποστηρίζει πολλές γλώσσες και παρέχει υποστήριξη για επεκτάσεις. Οι μεμονωμένοι προγραμματιστές δεν έχουν περιορισμούς στη χρήση της έκδοσης Community. Οι ακόλουθες χρήσεις επιτρέπουν επίσης την απεριόριστη χρήση: συνεισφορά σε έργα ανοικτού κώδικα, ακαδημαϊκή έρευνα, σε περιβάλλον μάθησης σε αίθουσα διδασκαλίας και για την ανάπτυξη και δοκιμή προγραμμάτων οδήγησης συσκευών για το λειτουργικό σύστημα Windows. Όλες οι άλλες χρήσεις από έναν οργανισμό εξαρτώνται από το αν κατατάσσετε ως επιχείρηση (περισσότεροι από 250 εργαζόμενοι ή ετήσια έσοδα άνω του 1 εκατομμυρίου USD, σύμφωνα με τη Microsoft). Οι μη επιχειρήσεις μπορούν να χρησιμοποιούν μέχρι 5 αντίγραφα χωρίς περιορισμό, ο αριθμός χρηστών 6 και άνω απαιτεί εμπορική άδεια χρήσης- οι οργανισμοί Enterprise απαιτούν εμπορική άδεια χρήσης για χρήση εκτός των αναφερόμενων εξαιρέσεων. Το Visual Studio Community απευθύνεται σε μεμονωμένους προγραμματιστές και μικρές ομάδες.

### 2) Professional

Από το Visual Studio 2010, η έκδοση Professional είναι η εμπορική έκδοση εισαγωγικού επιπέδου του Visual Studio. (Προηγουμένως, ήταν διαθέσιμη μια έκδοση Standard με πιο περιορισμένες δυνατότητες). Παρέχει ένα IDE για όλες τις υποστηριζόμενες γλώσσες ανάπτυξης. Η υποστήριξη MSDN είναι διαθέσιμη ως MSDN Essentials ή ως πλήρης βιβλιοθήκη MSDN ανάλογα με την αδειοδότηση. Υποστηρίζει επεξεργασία XML και XSLT και μπορεί να δημιουργήσει πακέτα ανάπτυξης που χρησιμοποιούν μόνο ClickOnce και MSI. Περιλαμβάνει εργαλεία όπως ο Server Explorer και ενσωμάτωση με τον Microsoft SQL Server επίσης. Η υποστήριξη ανάπτυξης Windows Mobile περιλαμβανόταν στο Visual Studio 2005 Standard, ωστόσο, με το Visual Studio 2008, είναι διαθέσιμη μόνο στις εκδόσεις Professional και υψηλότερες

### 3) Enterprise

Εκτός από τα χαρακτηριστικά που παρέχονται από την έκδοση Professional, η έκδοση Enterprise παρέχει ένα νέο σύνολο εργαλείων ανάπτυξης λογισμικού, ανάπτυξης βάσεων δεδομένων, συνεργασίας, μετρήσεων, αρχιτεκτονικής, δοκιμών και αναφορών.

## 4. Microsoft SQL Server

Ο Microsoft SQL Server είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που αναπτύχθηκε από τη Microsoft. Ως διακομιστής βάσεων δεδομένων, είναι ένα προϊόν λογισμικού με πρωταρχική λειτουργία την αποθήκευση και ανάκτηση δεδομένων όπως ζητούνται από άλλες εφαρμογές λογισμικού - οι οποίες μπορεί να εκτελούνται είτε στον ίδιο υπολογιστή είτε σε άλλον υπολογιστή μέσω δικτύου (συμπεριλαμβανομένου του Διαδικτύου). Η Microsoft διαθέτει στην αγορά τουλάχιστον δώδεκα διαφορετικές εκδόσεις του Microsoft SQL Server, οι οποίες απευθύνονται σε διαφορετικά κοινά και για φόρτους εργασίας που κυμαίνονται από μικρές εφαρμογές ενός μηχανήματος έως μεγάλες εφαρμογές που απευθύνονται στο Διαδίκτυο με πολλούς ταυτόχρονους χρήστες.



Εικόνα 6. Εικόνα του Microsoft SQL Server

Πηγή: <https://www.sqlservertutorial.net/>

### 4.1 Αρχιτεκτονική

Το επίπεδο πρωτοκόλλου υλοποιεί την εξωτερική διασύνδεση με τον SQL Server. Όλες οι λειτουργίες που μπορούν να κληθούν στον SQL Server κοινοποιούνται σε αυτόν μέσω μιας μορφής που ορίζεται από τη Microsoft και ονομάζεται Tabular Data Stream (TDS). Το TDS είναι ένα πρωτόκολλο επιπέδου εφαρμογής, που χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ ενός διακομιστή βάσης δεδομένων και ενός πελάτη. Αρχικά σχεδιάστηκε και αναπτύχθηκε από τη Sybase Inc. για τη μηχανή σχεσιακής βάσης



δεδομένων Sybase SQL Server το 1984 και αργότερα από τη Microsoft στο Microsoft SQL Server, τα πακέτα TDS μπορούν να εγκλωβιστούν σε άλλα πρωτόκολλα που εξαρτώνται από τη φυσική μεταφορά, όπως το TCP/IP, οι ονομαστικές σωλήνες και η κοινή μνήμη. Κατά συνέπεια, η πρόσβαση στον SQL Server είναι διαθέσιμη μέσω αυτών των πρωτοκόλλων. Επιπλέον, το API του SQL Server εκτίθεται επίσης μέσω υπηρεσιών ιστού.

## 4.2 Ιστορική αναδρομή

Η ιστορία του Microsoft SQL Server αρχίζει με το πρώτο προϊόν Microsoft SQL Server - τον SQL Server 1.0, έναν διακομιστή 16-bit για το λειτουργικό σύστημα OS/2 το 1989 - και εκτείνεται μέχρι σήμερα. Το όνομά του είναι απολύτως περιγραφικό, καθώς πρόκειται για λογισμικό *διακομιστή* που ανταποκρίνεται σε ερωτήματα στη γλώσσα *SQL*.

## 4.3 Αποθήκευση δεδομένων

Η αποθήκευση δεδομένων γίνεται με την βοήθεια της βάσης δεδομένων, η οποία είναι μια συλλογή πινάκων με τυποποιημένες στήλες. Ο SQL Server υποστηρίζει διάφορους τύπους δεδομένων, μεταξύ των οποίων και πρωτόγονους τύπους όπως *Integer*, *Float*, *Decimal*, *Char* (συμπεριλαμβανομένων των συμβολοσειρών χαρακτήρων), *Varchar* (συμβολοσειρές χαρακτήρων μεταβλητού μήκους), *binary* (για μη δομημένες μάζες δεδομένων), *Text* (για δεδομένα κειμένου) μεταξύ άλλων. Η στρογγυλοποίηση των *float* σε ακέραιους αριθμούς χρησιμοποιεί είτε συμμετρική αριθμητική στρογγυλοποίηση είτε συμμετρική στρογγυλοποίηση προς τα κάτω (*fix*) ανάλογα με τα ορίσματα: `SELECT Round(2.5, 0)` δίνει 3. Ο Microsoft SQL Server επιτρέπει επίσης τον ορισμό και τη χρήση σύνθετων τύπων που καθορίζονται από τον χρήστη (UDT). Διαθέτει επίσης στατιστικά στοιχεία του διακομιστή ως εικονικούς πίνακες και προβολές (που ονομάζονται *Dynamic Management Views* ή *DMVs*). Εκτός από τους πίνακες, μια βάση δεδομένων μπορεί επίσης να περιέχει άλλα αντικείμενα, συμπεριλαμβανομένων των προβολών, των αποθηκευμένων διαδικασιών, των δεικτών και των περιορισμών, μαζί με ένα αρχείο καταγραφής συναλλαγών. Μια βάση δεδομένων του SQL Server μπορεί να περιέχει το πολύ 231 αντικείμενα και μπορεί να καλύπτει πολλαπλά αρχεία επιπέδου λειτουργικού συστήματος με μέγιστο μέγεθος αρχείου 260 bytes (1 exabyte). Τα δεδομένα της βάσης δεδομένων αποθηκεύονται σε αρχεία πρωτογενών δεδομένων με επέκταση `. mdf`. Τα δευτερεύοντα αρχεία δεδομένων, τα οποία αναγνωρίζονται με την επέκταση `. ndf`, χρησιμοποιούνται για να επιτρέψουν την κατανομή των δεδομένων μιας ενιαίας βάσης δεδομένων σε περισσότερα από ένα αρχεία και προαιρετικά σε περισσότερα από ένα συστήματα αρχείων. Τα αρχεία καταγραφής αναγνωρίζονται με την επέκταση `. ldf`. Ο αποθηκευτικός χώρος που διατίθεται σε μια βάση δεδομένων χωρίζεται σε σελίδες με διαδοχική αρίθμηση, μεγέθους 8 KB η κάθε μία. Μια σελίδα είναι η βασική μονάδα εισόδου/εξόδου για τις λειτουργίες του SQL Server. Μια σελίδα επισημαίνεται με μια επικεφαλίδα 96 byte η οποία αποθηκεύει μεταδεδομένα σχετικά με τη σελίδα, συμπεριλαμβανομένων του αριθμού σελίδας, του τύπου σελίδας, του ελεύθερου χώρου

στη σελίδα και του αναγνωριστικού του αντικειμένου στο οποίο ανήκει. Ο τύπος σελίδας καθορίζει τα δεδομένα που περιέχονται στη σελίδα. Τα δεδομένα αυτά περιλαμβάνουν: δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων, ένα ευρετήριο, έναν χάρτη κατανομής, ο οποίος περιέχει πληροφορίες σχετικά με τον τρόπο κατανομής των σελίδων σε πίνακες και ευρετήρια, και έναν χάρτη αλλαγών, ο οποίος περιέχει πληροφορίες σχετικά με τις αλλαγές που έχουν γίνει σε άλλες σελίδες από το τελευταίο αντίγραφο ασφαλείας ή την τελευταία καταγραφή ή περιέχουν μεγάλους τύπους δεδομένων, όπως εικόνα ή κείμενο. Ενώ μια σελίδα είναι η βασική μονάδα μιας λειτουργίας εισόδου/εξόδου, η διαχείριση του χώρου γίνεται στην πραγματικότητα με όρους έκτασης που αποτελείται από 8 σελίδες. Ένα αντικείμενο βάσης δεδομένων μπορεί είτε να καλύπτει και τις 8 σελίδες σε μια έκταση ("ομοιόμορφη έκταση") είτε να μοιράζεται μια έκταση με έως και 7 άλλα αντικείμενα ("μικτή έκταση"). Μια γραμμή σε έναν πίνακα βάσης δεδομένων δεν μπορεί να καλύψει περισσότερες από μία σελίδες, οπότε περιορίζεται σε μέγεθος 8 KB. Ωστόσο, εάν τα δεδομένα υπερβαίνουν τα 8 KB και η γραμμή περιέχει *nvarchar* ή *varbinary* δεδομένα, τα δεδομένα σε αυτές τις στήλες μετακινούνται σε μια νέα σελίδα (ή ενδεχομένως σε μια ακολουθία σελίδων, που ονομάζεται *μονάδα κατανομής*) και αντικαθίστανται με έναν δείκτη στα δεδομένα. Για τη φυσική αποθήκευση ενός πίνακα, οι γραμμές του χωρίζονται σε μια σειρά καταταμίσεων (αριθμημένες από το 1 έως το n). Το μέγεθος των καταταμίσεων καθορίζεται από τον χρήστη- εξ ορισμού, όλες οι γραμμές βρίσκονται σε μία μόνο κατάτμηση. Ένας πίνακας χωρίζεται σε πολλαπλά διαμερίσματα προκειμένου να διαμοιραστεί μια βάση δεδομένων σε μια συστάδα υπολογιστών. Οι γραμμές σε κάθε κατάτμηση αποθηκεύονται είτε σε δομή B-tree είτε σε δομή σωρού. Εάν ο πίνακας έχει ένα σχετικό, ομαδοποιημένο ευρετήριο για να επιτρέψει τη γρήγορη ανάκτηση των γραμμών, οι γραμμές αποθηκεύονται με τη σειρά σύμφωνα με τις τιμές του ευρετηρίου τους, με ένα δέντρο B να παρέχει το ευρετήριο. Τα δεδομένα βρίσκονται στον κόμβο-φύλλο των φύλλων και άλλοι κόμβοι που αποθηκεύουν τις τιμές του δείκτη για τα δεδομένα του φύλλου είναι προσβάσιμοι από τους αντίστοιχους κόμβους. Εάν το ευρετήριο είναι μη ομαδοποιημένο, οι γραμμές δεν ταξινομούνται σύμφωνα με τα κλειδιά του ευρετηρίου. Μια προβολή με ευρετήριο έχει την ίδια δομή αποθήκευσης με έναν πίνακα με ευρετήριο. Ένας πίνακας χωρίς ομαδοποιημένο ευρετήριο αποθηκεύεται σε μια μη ταξινομημένη δομή σωρού. Ωστόσο, ο πίνακας μπορεί να έχει μη ομαδοποιημένους δείκτες για να επιτρέψει τη γρήγορη ανάκτηση των γραμμών. Σε ορισμένες περιπτώσεις η δομή σωρού έχει πλεονεκτήματα απόδοσης έναντι της δομής με συστάδες. Τόσο οι σωροί όσο και τα δέντρα B μπορούν να καλύπτουν πολλαπλές μονάδες κατανομής.

### 4.3.1 Διαχείριση απομονωτών

Ο SQL Server αποθηκεύει σελίδες σε προσωρινή μνήμη RAM για την ελαχιστοποίηση της εισόδου/εξόδου στο δίσκο. Οποιαδήποτε σελίδα 8 KB μπορεί να αποθηκευτεί στη μνήμη και το σύνολο όλων των σελίδων που είναι αποθηκευμένες επί του παρόντος ονομάζεται προσωρινή μνήμη απομόνωσης (buffer cache). Η ποσότητα μνήμης που διαθέτει ο SQL Server αποφασίζει πόσες σελίδες θα αποθηκευτούν στη



μνήμη. Η διαχείριση της προσωρινής μνήμης απομονωμένης μνήμης γίνεται από το *Buffer Manager*. Είτε η ανάγνωση από είτε η εγγραφή σε οποιαδήποτε σελίδα την αντιγράφει στην προσωρινή μνήμη buffer. Οι επακόλουθες αναγνώσεις ή εγγραφές ανακατευθύνονται στο αντίγραφο στη μνήμη και όχι στην έκδοση στο δίσκο. Η σελίδα ενημερώνεται στο δίσκο από το Buffer Manager μόνο εάν δεν έχει γίνει αναφορά στην κρυφή μνήμη για κάποιο χρονικό διάστημα. Κατά την εγγραφή σελίδων πίσω στο δίσκο, χρησιμοποιείται ασύγχρονη E/E, με την οποία η λειτουργία E/E πραγματοποιείται σε ένα νήμα στο παρασκήνιο, έτσι ώστε άλλες λειτουργίες να μην χρειάζεται να περιμένουν την ολοκλήρωση της λειτουργίας E/E. Κάθε σελίδα γράφεται μαζί με το άθροισμα ελέγχου της κατά την εγγραφή της. Κατά την ανάγνωση της σελίδας πίσω, το άθροισμα ελέγχου υπολογίζεται ξανά και ταυτίζεται με την αποθηκευμένη έκδοση για να διασφαλιστεί ότι η σελίδα δεν έχει υποστεί ζημία ή αλλοίωση στο μεταξύ.

### 4.3.2 Συγχρονισμός και κλείδωμα

Ο SQL Server επιτρέπει σε πολλούς πελάτες να χρησιμοποιούν ταυτόχρονα την ίδια βάση δεδομένων. Ως εκ τούτου, πρέπει να ελέγχει την ταυτόχρονη πρόσβαση σε κοινόχρηστα δεδομένα, για να διασφαλίζει την ακεραιότητα των δεδομένων - όταν πολλαπλοί πελάτες ενημερώνουν τα ίδια δεδομένα ή όταν οι πελάτες προσπαθούν να διαβάσουν δεδομένα που βρίσκονται σε διαδικασία αλλαγής από άλλον πελάτη. Ο SQL Server παρέχει δύο τρόπους ελέγχου ταυτόχρονης πρόσβασης: την απαισιόδοξη ταυτόχρονη πρόσβαση και την αισιόδοξη ταυτόχρονη πρόσβαση. Όταν χρησιμοποιείται απαισιόδοξος έλεγχος ταυτόχρονης πρόσβασης, ο SQL Server ελέγχει την ταυτόχρονη πρόσβαση χρησιμοποιώντας κλειδαριές. Οι κλειδαριές μπορούν να είναι είτε κοινές είτε αποκλειστικές. Το αποκλειστικό κλείδωμα χορηγεί στον χρήστη αποκλειστική πρόσβαση στα δεδομένα - κανένας άλλος χρήστης δεν μπορεί να έχει πρόσβαση στα δεδομένα όσο κρατείται το κλείδωμα. Τα κοινόχρηστα κλειδώματα χρησιμοποιούνται όταν γίνεται ανάγνωση ορισμένων δεδομένων - πολλοί χρήστες μπορούν να διαβάσουν από δεδομένα που έχουν κλειδωθεί με κοινόχρηστο κλείδωμα, αλλά δεν μπορούν να αποκτήσουν αποκλειστικό κλείδωμα. Οι τελευταίοι θα πρέπει να περιμένουν να απελευθερωθούν όλα τα κοινόχρηστα κλειδώματα. Οι κλειδαριές μπορούν να εφαρμοστούν σε διάφορα επίπεδα λεπτομέρειας - σε ολόκληρους πίνακες, σελίδες ή ακόμη και ανά σειρά σε πίνακες. Για ευρετήρια, μπορεί να είναι είτε σε ολόκληρο το ευρετήριο είτε σε φύλλα ευρετηρίου. Το επίπεδο λεπτομέρειας που θα χρησιμοποιηθεί ορίζεται ανά βάση δεδομένων από τον διαχειριστή της βάσης δεδομένων. Παρόλο που ένα σύστημα κλειδώματος με λεπτή διαβάθμιση επιτρέπει σε περισσότερους χρήστες να χρησιμοποιούν τον πίνακα ή το ευρετήριο ταυτόχρονα, απαιτεί περισσότερους πόρους, οπότε δεν αποδίδει αυτόματα υψηλότερες επιδόσεις. Ο SQL Server περιλαμβάνει επίσης δύο πιο ελαφριές λύσεις αμοιβαίου αποκλεισμού -latches και spinlock- οι οποίες είναι λιγότερο ισχυρές από τις κλειδαριές αλλά λιγότερο απαιτητικές σε πόρους. Ο SQL Server τις χρησιμοποιεί για DMVs και άλλους πόρους που συνήθως δεν είναι απασχολημένοι. Ο SQL Server παρακολουθεί επίσης όλα τα νήματα εργασίας που αποκτούν κλειδώματα για να διασφαλίσει ότι δεν καταλήγουν σε αδιέξοδα - σε περίπτωση που καταλήξουν, ο SQL

Server λαμβάνει διορθωτικά μέτρα, τα οποία σε πολλές περιπτώσεις είναι να σκοτώσει ένα από τα νήματα που έχουν εμπλακεί σε αδιέξοδο και να ανατρέψει τη συναλλαγή που ξεκίνησε. Για την υλοποίηση του κλειδώματος, ο SQL Server περιέχει το *Lock Manager*. Ο Lock Manager διατηρεί έναν πίνακα στη μνήμη που διαχειρίζεται τα αντικείμενα της βάσης δεδομένων και τα κλειδώματα, εάν υπάρχουν, σε αυτά μαζί με άλλα μεταδεδομένα σχετικά με το κλείδωμα. Η πρόσβαση σε οποιοδήποτε κοινόχρηστο αντικείμενο διαμεσολαβείται από τον διαχειριστή κλειδώματος, ο οποίος είτε χορηγεί πρόσβαση στον πόρο είτε τον μπλοκάρει. Ο SQL Server παρέχει επίσης τον αισιόδοξο μηχανισμό ελέγχου ταυτόχρονης εκτέλεσης, ο οποίος είναι παρόμοιος με τον μηχανισμό πολλαπλών εκδόσεων/έλεγχος ταυτόχρονης χρήσης που χρησιμοποιείται σε άλλες βάσεις δεδομένων. Ο μηχανισμός επιτρέπει τη δημιουργία μιας νέας έκδοσης μιας γραμμής κάθε φορά που η γραμμή ενημερώνεται, σε αντίθεση με την αντικατάσταση της γραμμής, δηλαδή μια γραμμή αναγνωρίζεται επιπλέον από το αναγνωριστικό της συναλλαγής που δημιούργησε την έκδοση της γραμμής. Τόσο η παλιά όσο και η νέα έκδοση της γραμμής αποθηκεύονται και διατηρούνται, αν και οι παλιές εκδόσεις μεταφέρονται από τη βάση δεδομένων σε μια βάση δεδομένων του συστήματος που προσδιορίζεται ως Tempdb. Όταν μια γραμμή βρίσκεται σε διαδικασία ενημέρωσης, οποιαδήποτε άλλα αιτήματα δεν μπλοκάρονται (σε αντίθεση με το κλείδωμα) αλλά εκτελούνται στην παλαιότερη έκδοση της γραμμής. Εάν το άλλο αίτημα είναι μια δήλωση ενημέρωσης, θα οδηγήσει σε δύο διαφορετικές εκδόσεις των γραμμών - και οι δύο θα αποθηκευτούν από τη βάση δεδομένων, αναγνωρισμένες από τα αντίστοιχα αναγνωριστικά συναλλαγής τους.

#### 4.4 Ανάκτηση δεδομένων και δυνατότητα προγραμματισμού

Ο κύριος τρόπος ανάκτησης δεδομένων από μια βάση δεδομένων του SQL Server είναι η υποβολή ερωτημάτων. Το ερώτημα εκφράζεται χρησιμοποιώντας μια παραλλαγή της SQL που ονομάζεται T-SQL, μια διάλεκτος που μοιράζεται ο Microsoft SQL Server με τον Sybase SQL Server λόγω της κληρονομιάς του. Το ερώτημα καθορίζει δηλωτικά τι πρέπει να ανακτηθεί. Επεξεργάζεται από τον επεξεργαστή ερωτημάτων, ο οποίος υπολογίζει την ακολουθία των βημάτων που θα χρειαστούν για την ανάκτηση των ζητούμενων δεδομένων. Η ακολουθία των ενεργειών που απαιτούνται για την εκτέλεση ενός ερωτήματος ονομάζεται σχέδιο ερωτήματος. Μπορεί να υπάρχουν πολλαπλοί τρόποι επεξεργασίας του ίδιου ερωτήματος. Για παράδειγμα, για ένα ερώτημα που περιέχει ένα join δήλωση και μια δήλωση select, η εκτέλεση join και στους δύο πίνακες και στη συνέχεια η εκτέλεση select στα αποτελέσματα θα έδινε το ίδιο αποτέλεσμα με την επιλογή από κάθε πίνακα και στη συνέχεια την εκτέλεση του join, αλλά θα οδηγούσε σε διαφορετικά σχέδια εκτέλεσης. Σε μια τέτοια περίπτωση, ο SQL Server επιλέγει το σχέδιο που αναμένεται να αποδώσει τα αποτελέσματα στον συντομότερο δυνατό χρόνο. Αυτό ονομάζεται βελτιστοποίηση ερωτήματος και εκτελείται από τον ίδιο τον επεξεργαστή ερωτημάτων. Ο SQL Server περιλαμβάνει έναν βελτιστοποιητή ερωτημάτων με βάση το κόστος, ο οποίος προσπαθεί να βελτιστοποιήσει το κόστος, όσον αφορά τους πόρους που θα χρειαστούν για την εκτέλεση του ερωτήματος. Δεδομένου ενός ερωτήματος, τότε ο βελτιστοποιητής ερωτημάτων εξετάζει το σχήμα της βάσης

δεδομένων, τα στατιστικά στοιχεία της βάσης δεδομένων και το φορτίο του συστήματος εκείνη τη στιγμή. Στη συνέχεια αποφασίζει ποια σειρά πρόσβασης στους πίνακες που αναφέρονται στο ερώτημα, ποια σειρά εκτέλεσης των πράξεων και ποια μέθοδος πρόσβασης θα χρησιμοποιηθεί για την πρόσβαση στους πίνακες. Για παράδειγμα, αν ο πίνακας έχει σχετικό ευρετήριο, αν το ευρετήριο πρέπει να χρησιμοποιηθεί ή όχι: αν το ευρετήριο αφορά μια στήλη η οποία δεν είναι μοναδική για τις περισσότερες στήλες (χαμηλή "επιλεκτικότητα"), μπορεί να μην αξίζει τον κόπο να χρησιμοποιηθεί το ευρετήριο για την πρόσβαση στα δεδομένα. Τέλος, αποφασίζει αν θα εκτελεστεί το ερώτημα ταυτόχρονα ή όχι. Ενώ η ταυτόχρονη εκτέλεση είναι πιο δαπανηρή από την άποψη του συνολικού χρόνου του επεξεργαστή, επειδή η εκτέλεση στην πραγματικότητα κατανέμεται σε διαφορετικούς επεξεργαστές μπορεί να σημαίνει ότι θα εκτελεστεί ταχύτερα. Μόλις δημιουργηθεί ένα σχέδιο ερωτήματος για ένα ερώτημα, αποθηκεύεται προσωρινά στην προσωρινή μνήμη. Για περαιτέρω κλήσεις του ίδιου ερωτήματος, χρησιμοποιείται το αποθηκευμένο σχέδιο. Τα αχρησιμοποιήτα σχέδια απορρίπτονται μετά από κάποιο χρονικό διάστημα. Ο SQL Server επιτρέπει επίσης τον ορισμό αποθηκευμένων διαδικασιών. Οι αποθηκευμένες διαδικασίες είναι παραμετροποιημένα ερωτήματα T-SQL, τα οποία αποθηκεύονται στον ίδιο το διακομιστή (και δεν εκδίδονται από την εφαρμογή-πελάτη, όπως συμβαίνει με τα γενικά ερωτήματα). Οι αποθηκευμένες διαδικασίες μπορούν να δέχονται τιμές που αποστέλλονται από τον πελάτη ως παράμετροι εισόδου και να στέλνουν πίσω αποτελέσματα ως παράμετροι εξόδου. Μπορούν να καλέσουν καθορισμένες συναρτήσεις και άλλες αποθηκευμένες διαδικασίες, συμπεριλαμβανομένης της ίδιας αποθηκευμένης διαδικασίας (μέχρι έναν καθορισμένο αριθμό φορών). Μπορούν να έχουν επιλεκτική πρόσβαση σε. Σε αντίθεση με άλλα ερωτήματα, οι αποθηκευμένες διαδικασίες έχουν ένα σχετικό όνομα, το οποίο χρησιμοποιείται κατά το χρόνο εκτέλεσης για την επίλυση στα πραγματικά ερωτήματα. Επίσης, επειδή ο κώδικας δεν χρειάζεται να αποστέλλεται από τον πελάτη κάθε φορά (καθώς μπορεί να γίνει πρόσβαση με βάση το όνομα), μειώνεται η κυκλοφορία του δικτύου και βελτιώνεται κάπως η απόδοση. Τα σχέδια εκτέλεσης για τις αποθηκευμένες διαδικασίες αποθηκεύονται επίσης στην προσωρινή μνήμη, εφόσον είναι απαραίτητο.

#### 4.4.1 T-SQL

Η T-SQL (Transact-SQL) είναι η ιδιόκτητη επέκταση της διαδικαστικής γλώσσας της Microsoft για τον SQL Server. Παρέχει εντολές REPL (Read-Eval-Print-Loop) που επεκτείνουν το σύνολο εντολών της τυπικής SQL για εντολές χειρισμού δεδομένων (DML) και ορισμού δεδομένων (DDL), συμπεριλαμβανομένων των ειδικών ρυθμίσεων του SQL Server, της ασφάλειας και της διαχείρισης στατιστικών στοιχείων της βάσης δεδομένων. Εκθέτει λέξεις-κλειδιά για τις λειτουργίες που μπορούν να εκτελεστούν στον SQL Server, συμπεριλαμβανομένης της δημιουργίας και τροποποίησης σχημάτων βάσεων δεδομένων, της εισαγωγής και επεξεργασίας δεδομένων στη βάση δεδομένων, καθώς και της παρακολούθησης και διαχείρισης του ίδιου του διακομιστή. Οι εφαρμογές-πελάτες που καταναλώνουν δεδομένα ή διαχειρίζονται το διακομιστή θα αξιοποιήσουν τη λειτουργικότητα του SQL Server στέλνοντας ερωτήματα και δηλώσεις T-SQL, τα οποία

στη συνέχεια επεξεργάζονται από το διακομιστή και τα αποτελέσματα (ή τα σφάλματα) επιστρέφονται στην εφαρμογή-πελάτη. Για το σκοπό αυτό εκθέτει πίνακες μόνο για ανάγνωση από τους οποίους μπορούν να διαβαστούν στατιστικά στοιχεία του διακομιστή. Η λειτουργικότητα διαχείρισης εκτίθεται μέσω αποθηκευμένων διαδικασιών που ορίζονται από το σύστημα και οι οποίες μπορούν να κληθούν από ερωτήματα T-SQL για την εκτέλεση της λειτουργίας διαχείρισης. Είναι επίσης δυνατή η δημιουργία συνδεδεμένων εξυπηρετητών με τη χρήση T-SQL. Οι συνδεδεμένοι διακομιστές επιτρέπουν σε ένα μόνο ερώτημα να επεξεργάζεται λειτουργίες που εκτελούνται σε πολλαπλούς διακομιστές.

#### 4.4.2 SQL Server Native Client (SNAC)

Ο SQL Server Native Client είναι η εγγενής βιβλιοθήκη πρόσβασης σε δεδομένα από την πλευρά του πελάτη για τον Microsoft SQL Server, από την έκδοση 2005 και μετά. Υλοποιεί εγγενώς την υποστήριξη των χαρακτηριστικών του SQL Server, συμπεριλαμβανομένης της υλοποίησης Tabular Data Stream, της υποστήριξης για κατοπτρικές βάσεις δεδομένων SQL Server, της πλήρους υποστήριξης όλων των τύπων δεδομένων που υποστηρίζονται από τον SQL Server, των ασύγχρονων λειτουργιών, των ειδοποιήσεων ερωτημάτων, της υποστήριξης κρυπτογράφησης, καθώς και της λήψης πολλαπλών συνόλων αποτελεσμάτων σε μία μόνο συνεδρία βάσης δεδομένων. Το SQL Server Native Client χρησιμοποιείται κάτω από την κουκούλα από τα πρόσθετα του SQL Server για άλλες τεχνολογίες πρόσβασης σε δεδομένα, όπως το ADO ή το OLE DB. Το SQL Server Native Client μπορεί επίσης να χρησιμοποιηθεί απευθείας, παρακάμπτοντας τα γενικά επίπεδα πρόσβασης σε δεδομένα. Στις 28 Νοεμβρίου 2011, κυκλοφόρησε μια έκδοση προεπισκόπησης του προγράμματος οδήγησης SQL Server ODBC για Linux.

#### 4.4.3 SQL CLR

Ο Microsoft SQL Server 2005 περιλαμβάνει ένα συστατικό που ονομάζεται SQL CLR ("Common Language Runtime") μέσω του οποίου ενσωματώνεται στο .NET Framework. Σε αντίθεση με τις περισσότερες άλλες εφαρμογές που χρησιμοποιούν το .NET Framework, ο ίδιος ο SQL Server φιλοξενεί το .NET Framework runtime, δηλαδή οι απαιτήσεις μνήμης, νηματοποίησης και διαχείρισης πόρων του .NET Framework ικανοποιούνται από το ίδιο το SQLOS και όχι από το υποκείμενο λειτουργικό σύστημα των Windows. Το SQLOS παρέχει επίσης υπηρεσίες ανίχνευσης και επίλυσης αδιεξόδων για κώδικα .NET. Με το SQL CLR, οι αποθηκευμένες διαδικασίες και τα triggers μπορούν να γραφούν σε οποιαδήποτε διαχειριζόμενη γλώσσα .NET, συμπεριλαμβανομένων των C# και VB.NET. Ο διαχειριζόμενος κώδικας μπορεί επίσης να χρησιμοποιηθεί για τον ορισμό UDT's (user defined types), οι οποίοι μπορούν να παραμείνουν στη βάση δεδομένων. Ο διαχειριζόμενος κώδικας μεταγλωττίζεται σε σύνολο CLI και αφού ελεγχθεί για την ασφάλεια τύπου, καταχωρείται στη βάση δεδομένων. Μετά από αυτό, μπορούν να κληθούν όπως οποιαδήποτε άλλη διαδικασία. Ωστόσο, μόνο ένα υποσύνολο της Βιβλιοθήκης Βασικών Κλάσεων είναι



διαθέσιμο, όταν εκτελείται κώδικας υπό SQL CLR. Τα περισσότερα API που σχετίζονται με τη λειτουργικότητα της διεπαφής χρήστη δεν είναι διαθέσιμα. Κατά τη συγγραφή κώδικα για το SQL CLR, τα δεδομένα που είναι αποθηκευμένα σε βάσεις δεδομένων του SQL Server μπορούν να προσπελαστούν με τη χρήση των APIs ADO.NET όπως οποιαδήποτε άλλη διαχειριζόμενη εφαρμογή που προσπελαίνει δεδομένα του SQL Server. Ωστόσο, με τον τρόπο αυτό δημιουργείται μια νέα σύνδεση βάσης δεδομένων, διαφορετική από αυτή στην οποία εκτελείται ο κώδικας. Για να αποφευχθεί αυτό, ο SQL Server παρέχει ορισμένες βελτιώσεις στον πάροχο ADO.NET που επιτρέπουν την ανακατεύθυνση της σύνδεσης στην ίδια συνεδρία που φιλοξενεί ήδη τον εκτελούμενο κώδικα. Τέτοιες συνδέσεις ονομάζονται συνδέσεις πλαισίου και καθορίζονται με τον καθορισμό της παραμέτρου σύνδεσης πλαισίου σε true στη συμβολοσειρά σύνδεσης. Ο SQL Server παρέχει επίσης αρκετές άλλες βελτιώσεις στο ADO.NET API, συμπεριλαμβανομένων κλάσεων για την εργασία με δεδομένα σε μορφή πίνακα ή με μία μόνο γραμμή δεδομένων, καθώς και κλάσεων για την εργασία με εσωτερικά μεταδεδωμένα σχετικά με τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων. Παρέχει επίσης πρόσβαση στα χαρακτηριστικά XML του SQL Server, συμπεριλαμβανομένης της υποστήριξης XQuery. Αυτές οι βελτιώσεις είναι επίσης διαθέσιμες στις T-SQL Procedures ως συνέπεια της εισαγωγής του νέου τύπου δεδομένων XML (query, value, nodes functions).

## 4.5 Υπηρεσίες

Ο SQL Server περιλαμβάνει επίσης μια σειρά από πρόσθετες υπηρεσίες. Αν και αυτές δεν είναι απαραίτητες για τη λειτουργία του συστήματος βάσεων δεδομένων, παρέχουν υπηρεσίες προστιθέμενης αξίας πάνω από το βασικό σύστημα διαχείρισης βάσεων δεδομένων. Οι υπηρεσίες αυτές εκτελούνται είτε ως μέρος κάποιου στοιχείου του SQL Server είτε εκτός διεργασίας ως υπηρεσία των Windows και παρουσιάζουν το δικό τους API για τον έλεγχο και την αλληλεπίδραση μαζί τους.

### 4.5.1 Υπηρεσίες μηχανικής μάθησης

Οι υπηρεσίες μηχανικής μάθησης του SQL Server λειτουργούν εντός της περίπτωσης του SQL Server, επιτρέποντας στους ανθρώπους να κάνουν μηχανική μάθηση και ανάλυση δεδομένων χωρίς να χρειάζεται να στέλνουν δεδομένα μέσω δικτύου ή να περιορίζονται από τη μνήμη των δικών τους υπολογιστών. Οι υπηρεσίες συνοδεύονται από τις διανομές R και Python της Microsoft, οι οποίες περιέχουν κοινώς χρησιμοποιούμενα πακέτα για την επιστήμη των δεδομένων, μαζί με ορισμένα ιδιόκτητα πακέτα (π.χ. revoscalepy, RevoScaleR, microsoftml) που μπορούν να χρησιμοποιηθούν για τη δημιουργία μηχανικών μοντέλων σε κλίμακα. Οι αναλυτές μπορούν είτε να ρυθμίσουν το μηχάνημα-πελάτη τους να συνδεθεί σε έναν απομακρυσμένο διακομιστή SQL και να προωθήσουν τις εκτελέσεις των σεναρίων σε αυτόν, είτε να εκτελέσουν ένα σενάριο R ή Python ως εξωτερικό σενάριο μέσα σε ένα ερώτημα T-SQL. Το εκπαιδευμένο μοντέλο μηχανικής μάθησης μπορεί να αποθηκευτεί μέσα σε μια βάση

δεδομένων και να χρησιμοποιηθεί για βαθμολόγηση.

## 4.5.2 Διαμεσολαβητής υπηρεσιών

Χρησιμοποιείται μέσα σε ένα παράδειγμα, σε περιβάλλον προγραμματισμού. Για cross-instance εφαρμογές, ο Service Broker επικοινωνεί μέσω TCP/IP και επιτρέπει το συγχρονισμό των διαφόρων στοιχείων, μέσω της ανταλλαγής μηνυμάτων. Ο Service Broker, ο οποίος εκτελείται ως μέρος της μηχανής βάσης δεδομένων, παρέχει μια αξιόπιστη πλατφόρμα ανταλλαγής μηνυμάτων και ουράς μηνυμάτων για τις εφαρμογές του SQL Server.

Οι υπηρεσίες μεσίτη υπηρεσιών αποτελούνται από τα ακόλουθα μέρη:

- τύποι μηνυμάτων
- συμβάσεις
- ουρές
- προγράμματα υπηρεσιών
- διαδρομές

Ο τύπος μηνύματος καθορίζει τη μορφή δεδομένων που χρησιμοποιείται για το μήνυμα. Αυτό μπορεί να είναι ένα αντικείμενο XML, απλό κείμενο ή δυαδικά δεδομένα, καθώς και ένα μηδενικό σώμα μηνύματος για ειδοποιήσεις. Το συμβόλαιο ορίζει ποια μηνύματα χρησιμοποιούνται σε μια συνομιλία μεταξύ υπηρεσιών και ποιος μπορεί να τοποθετήσει μηνύματα στην ουρά. Η ουρά λειτουργεί ως πάροχος αποθήκευσης για τα μηνύματα. Υλοποιούνται εσωτερικά ως πίνακες από τον SQL Server, αλλά δεν υποστηρίζουν λειτουργίες εισαγωγής, ενημέρωσης ή διαγραφής. Το πρόγραμμα υπηρεσιών λαμβάνει και επεξεργάζεται τα μηνύματα του διαμεσολαβητή υπηρεσιών. Συνήθως το πρόγραμμα υπηρεσίας υλοποιείται ως αποθηκευμένη διαδικασία ή εφαρμογή CLR. Οι διαδρομές είναι διευθύνσεις δικτύου στις οποίες βρίσκεται ο μεσίτης υπηρεσιών στο δίκτυο. Επίσης, ο διαμεσολαβητής υπηρεσιών υποστηρίζει χαρακτηριστικά ασφαλείας, όπως έλεγχο ταυτότητας δικτύου (χρησιμοποιώντας NTLM, Kerberos ή πιστοποιητικά εξουσιοδότησης), έλεγχο ακεραιότητας και κρυπτογράφηση μηνυμάτων.

## 4.5.3 Υπηρεσίες αντιγραφής

Οι υπηρεσίες αντιγραφής του SQL Server χρησιμοποιούνται από τον SQL Server για την αντιγραφή και τον συγχρονισμό των αντικειμένων της βάσης δεδομένων, είτε στο σύνολό τους είτε σε ένα υποσύνολο των αντικειμένων που υπάρχουν, σε πράκτορες αντιγραφής, οι οποίοι μπορεί να είναι άλλοι διακομιστές βάσεων δεδομένων στο δίκτυο ή κρυφές μνήμες βάσεων δεδομένων στην πλευρά του πελάτη. Οι υπηρεσίες αντιγραφής ακολουθούν ένα μοντέλο εκδότη/συνδρομητή, δηλαδή οι αλλαγές αποστέλλονται από έναν διακομιστή βάσης δεδομένων ("εκδότη") και λαμβάνονται από άλλους ("συνδρομητές"). Ο SQL Server υποστηρίζει τρεις διαφορετικούς τύπους αντιγραφής:

## 1) Αντιγραφή συναλλαγών

Κάθε συναλλαγή που πραγματοποιείται στη βάση δεδομένων του εκδότη (κύρια βάση δεδομένων) συγχρονίζεται στους συνδρομητές, οι οποίοι ενημερώνουν τις βάσεις δεδομένων τους με τη συναλλαγή. Η αντιγραφή συναλλαγών συγχρονίζει τις βάσεις δεδομένων σχεδόν σε πραγματικό χρόνο.

## 2) Αναπαραγωγή συγχώνευσης

Οι αλλαγές που πραγματοποιούνται τόσο στις βάσεις δεδομένων του εκδότη όσο και στις βάσεις δεδομένων των συνδρομητών παρακολουθούνται και περιοδικά οι αλλαγές συγχρονίζονται αμφίδρομα μεταξύ του εκδότη και των συνδρομητών. Εάν τα ίδια δεδομένα έχουν τροποποιηθεί με διαφορετικό τρόπο τόσο στις βάσεις δεδομένων του εκδότη όσο και στις βάσεις δεδομένων του συνδρομητή, ο συγχρονισμός θα οδηγήσει σε σύγκρουση η οποία πρέπει να επιλυθεί, είτε χειροκίνητα είτε με τη χρήση προκαθορισμένων πολιτικών. rowguid πρέπει να ρυθμιστεί σε μια στήλη εάν έχει ρυθμιστεί η αντιγραφή συγχώνευσης.

## 3) Αντιγραφή στιγμιότυπων

Η αντιγραφή στιγμιότυπου δημοσιεύει ένα αντίγραφο ολόκληρης της βάσης δεδομένων (το τότε στιγμιότυπο των δεδομένων) και αναπαράγεται στους συνδρομητές. Περαιτέρω αλλαγές στο στιγμιότυπο δεν παρακολουθούνται.

### 4.5.4 Υπηρεσίες ανάλυσης

Οι υπηρεσίες ανάλυσης του Microsoft SQL Server (SSAS) είναι ένα εργαλείο διαδικτυακής αναλυτικής επεξεργασίας (OLAP) και εξόρυξης δεδομένων στον Microsoft SQL Server. Το SSAS χρησιμοποιείται ως εργαλείο από οργανισμούς για την ανάλυση και την αξιοποίηση πληροφοριών που ενδεχομένως είναι διασκορπισμένες σε πολλαπλές βάσεις δεδομένων ή σε ανομοιογενείς πίνακες ή αρχεία. Η Microsoft έχει συμπεριλάβει μια σειρά υπηρεσιών στον SQL Server που σχετίζονται με την επιχειρηματική ευφυΐα και την αποθήκευση δεδομένων. Οι υπηρεσίες αυτές περιλαμβάνουν τις Υπηρεσίες ολοκλήρωσης, τις Υπηρεσίες αναφοράς και τις Υπηρεσίες ανάλυσης. Οι υπηρεσίες ανάλυσης περιλαμβάνουν μια ομάδα δυνατοτήτων OLAP και εξόρυξης δεδομένων και διατίθενται σε δύο εκδοχές, την πολυδιάστατη και την επιτραπέζια, όπου η διαφορά μεταξύ των δύο είναι ο τρόπος παρουσίασης των δεδομένων. Σε ένα επιτραπέζιο μοντέλο, οι πληροφορίες είναι τοποθετημένες σε δισδιάστατους πίνακες, οι οποίοι μπορούν έτσι να είναι πιο ευανάγνωστοι για έναν άνθρωπο. Ένα πολυδιάστατο μοντέλο μπορεί να περιέχει πληροφορίες με πολλούς βαθμούς ελευθερίας και πρέπει να ξεδιπλωθεί για να αυξηθεί η αναγνωσιμότητα από έναν άνθρωπο.

### 4.5.5 Υπηρεσίες αναφοράς

Το *SQL Server Reporting Services (SSRS)* είναι ένα σύστημα λογισμικού δημιουργίας εκθέσεων που βασίζεται σε διακομιστή από τη Microsoft. Αποτελεί μέρος μιας σουίτας υπηρεσιών του Microsoft SQL Server, συμπεριλαμβανομένων των SSAS (SQL Server Analysis Services) και SSIS (SQL Server Integration Services). Διαχειρίζεται μέσω διεπαφής Web και μπορεί να χρησιμοποιηθεί για την προετοιμασία και την παράδοση μιας ποικιλίας διαδραστικών και εκτυπωμένων αναφορών. Η υπηρεσία SSRS παρέχει μια διασύνδεση στο Microsoft Visual Studio, ώστε οι προγραμματιστές καθώς και οι διαχειριστές SQL να μπορούν να συνδεθούν με βάσεις δεδομένων SQL και να χρησιμοποιούν τα εργαλεία SSRS για τη διαμόρφωση αναφορών SQL με πολλούς και πολύπλοκους τρόπους. Παρέχει επίσης ένα εργαλείο "Report Builder" για λιγότερο τεχνικούς χρήστες για τη μορφοποίηση αναφορών SQL μικρότερης πολυπλοκότητας. Το SSRS ανταγωνίζεται το Crystal Reports και άλλα εργαλεία επιχειρηματικής ευφυΐας.

### 4.5.6 Υπηρεσίες ειδοποίησης

Το SQL Server Notification Services είναι μια πλατφόρμα που αναπτύχθηκε από τη Microsoft για την ανάπτυξη και την εγκατάσταση εφαρμογών ειδοποίησης που βασίζονται στην τεχνολογία SQL Server και το Microsoft .NET Framework. Το Notification Services προσφέρει μια κλιμακούμενη μηχανή διακομιστή στην οποία εκτελούνται εφαρμογές ειδοποιήσεων, με δυνατότητα πολλαπλών διακομιστών - παρέχοντας ευελιξία και κλιμάκωση για την ανάπτυξη εφαρμογών. Το Notification Services σχεδιάστηκε για να διευκολύνει την ανάπτυξη και την εγκατάσταση εφαρμογών ειδοποιήσεων που δημιουργούν εξατομικευμένες, έγκαιρες πληροφορίες για τους συνδρομητές. Για να σχεδιάσετε, να κωδικοποιήσετε και να δοκιμάσετε όλα τα στοιχεία που συνθέτουν μια ισχυρή εφαρμογή Υπηρεσιών ειδοποίησης - όπως ο προγραμματισμός ειδοποιήσεων, η ανίχνευση αποτυχίας, η λογική επανάληψης, η διαχείριση χρονικών ζωνών, η ομαδοποίηση ειδοποιήσεων και η διαχείριση ουρών αναμονής - η προσθήκη Υπηρεσιών ειδοποίησης σε εφαρμογές λογισμικού μπορεί να είναι ένα δύσκολο έργο.

### 4.5.7 Υπηρεσίες ενσωμάτωσης

Οι υπηρεσίες ολοκλήρωσης Microsoft SQL Server Integration Services (SSIS) είναι ένα συστατικό του λογισμικού βάσεων δεδομένων Microsoft SQL Server που μπορεί να χρησιμοποιηθεί για την εκτέλεση ενός ευρέος φάσματος εργασιών μετάβασης δεδομένων. Το SSIS είναι μια πλατφόρμα για εφαρμογές ολοκλήρωσης δεδομένων και ροής εργασιών. Διαθέτει ένα εργαλείο αποθήκευσης δεδομένων που χρησιμοποιείται για εξαγωγή, μετασχηματισμό και φόρτωση δεδομένων (ETL). Το εργαλείο μπορεί επίσης να χρησιμοποιηθεί για την αυτοματοποίηση της συντήρησης των βάσεων δεδομένων του SQL Server και των ενημερώσεων των δεδομένων πολυδιάστατων κύβων. Το SSIS κυκλοφόρησε για πρώτη φορά με τον Microsoft SQL Server 2005 και αντικατέστησε τις υπηρεσίες μετασχηματισμού δεδομένων, οι οποίες αποτελούσαν χαρακτηριστικό του



SQL Server από την έκδοση 7.0. Σε αντίθεση με το DTS, το οποίο περιλαμβανόταν σε όλες τις εκδόσεις, το SSIS είναι διαθέσιμο μόνο στις εκδόσεις "Standard", "Business Intelligence" και "Enterprise". Με το "Visual Studio Dev Essentials" της Microsoft είναι πλέον δυνατή η χρήση του SSIS με το Visual Studio 2017 χωρίς κόστος, αρκεί να γίνεται μόνο για σκοπούς ανάπτυξης και εκμάθησης.

#### 4.5.8 Υπηρεσία αναζήτησης πλήρους κειμένου

Η υπηρεσία αναζήτησης πλήρους κειμένου του SQL Server είναι μια εξειδικευμένη υπηρεσία ευρετηρίασης και αναζήτησης για μη δομημένο κείμενο που είναι αποθηκευμένο σε βάσεις δεδομένων του SQL Server. Το ευρετήριο αναζήτησης πλήρους κειμένου μπορεί να δημιουργηθεί σε οποιαδήποτε στήλη με δεδομένα κειμένου που βασίζονται σε χαρακτήρες. Επιτρέπει την αναζήτηση λέξεων στις στήλες κειμένου. Ενώ μπορεί να πραγματοποιηθεί με τον τελεστή SQL LIKE, η χρήση της υπηρεσίας αναζήτησης πλήρους κειμένου του SQL Server μπορεί να είναι πιο αποτελεσματική. Η υπηρεσία Full επιτρέπει την ανακριβή αντιστοίχιση της αρχικής συμβολοσειράς, η οποία υποδεικνύεται από μια τιμή Rank (Κατάταξη) που μπορεί να κυμαίνεται από 0 έως 1000 - μια υψηλότερη κατάταξη σημαίνει ακριβέστερη αντιστοίχιση. Επιτρέπει επίσης γλωσσικό ταίριασμα ("αναζήτηση με κλίση"), δηλαδή, γλωσσικές παραλλαγές μιας λέξης (όπως ένα ρήμα σε διαφορετικό χρόνο) θα είναι επίσης ένα ταίριασμα για μια δεδομένη λέξη (αλλά με χαμηλότερη κατάταξη από ένα ακριβές ταίριασμα). Υποστηρίζονται επίσης αναζητήσεις εγγύτητας, δηλαδή, αν οι λέξεις που αναζητούνται δεν εμφανίζονται με τη σειρά που καθορίζονται στο ερώτημα αλλά βρίσκονται κοντά η μία στην άλλη, θεωρούνται επίσης ταίριασμα. Η T-SQL εκθέτει ειδικούς τελεστές που μπορούν να χρησιμοποιηθούν για την πρόσβαση στις δυνατότητες του FTS. Η μηχανή αναζήτησης πλήρους κειμένου χωρίζεται σε δύο διεργασίες: τη διεργασία Filter Daemon (msftefd.exe) και τη διεργασία αναζήτησης (msftesql.exe). Αυτές οι διεργασίες αλληλεπιδρούν με τον SQL Server. Η διεργασία Search περιλαμβάνει τον indexer (που δημιουργεί τα ευρετήρια πλήρους κειμένου) και τον επεξεργαστή ερωτημάτων πλήρους κειμένου. Ο ευρετηριαστής σαρώνει τις στήλες κειμένου στη βάση δεδομένων. Μπορεί επίσης να ευρετηριάσει μέσω δυαδικών στηλών και να χρησιμοποιήσει iFilters για την εξαγωγή ουσιαστικού κειμένου από τη δυαδική σταγόνα (για παράδειγμα, όταν ένα έγγραφο του Microsoft Word είναι αποθηκευμένο ως μη δομημένο δυαδικό αρχείο σε μια βάση δεδομένων). Τα φίλτρα iFilters φιλοξενούνται από τη διεργασία Filter Daemon. Μόλις εξαχθεί το κείμενο, η διεργασία Filter Daemon το αναλύει σε μια ακολουθία λέξεων και το παραδίδει στον ευρετηριαστή. Ο ευρετηριαστής φιλτράρει τις λέξεις θορύβου, δηλαδή λέξεις όπως A, And, κ.λπ., οι οποίες εμφανίζονται συχνά και δεν είναι χρήσιμες για την αναζήτηση. Με τις υπόλοιπες λέξεις, δημιουργείται ένα ανεστραμμένο ευρετήριο, συσχετίζοντας κάθε λέξη με τις στήλες στις οποίες βρέθηκε. Ο ίδιος ο SQL Server περιλαμβάνει ένα στοιχείο Gatherer που παρακολουθεί τις αλλαγές στους πίνακες και καλεί τον ευρετηριαστή σε περίπτωση ενημερώσεων. Όταν ένα ερώτημα πλήρους κειμένου λαμβάνεται από τον επεξεργαστή ερωτημάτων του SQL Server, παραδίδεται στον επεξεργαστή ερωτημάτων FTS κατά τη διαδικασία αναζήτησης. Ο επεξεργαστής

ερωτημάτων FTS αναλύει το ερώτημα στις λέξεις που το αποτελούν, φιλτράρει τις λέξεις θορύβου και χρησιμοποιεί έναν ενσωματωμένο θησαυρό για να βρει τις γλωσσικές παραλλαγές για κάθε λέξη. Στη συνέχεια, οι λέξεις ερωτώνται με βάση το ανεστραμμένο ευρετήριο και υπολογίζεται μια κατάταξη της ακρίβειάς τους. Τα αποτελέσματα επιστρέφονται στον πελάτη μέσω της διαδικασίας του SQL Server.

#### 4.5.9 SQLCMD

Το SQLCMD είναι μια εφαρμογή γραμμής εντολών που συνοδεύει τον Microsoft SQL Server και εκθέτει τις δυνατότητες διαχείρισης του SQL Server. Επιτρέπει τη συγγραφή και εκτέλεση ερωτημάτων SQL από τη γραμμή εντολών. Μπορεί επίσης να λειτουργήσει ως γλώσσα σεναρίων για τη δημιουργία και εκτέλεση ενός συνόλου εντολών SQL ως σενάριο. Τέτοιες δέσμες ενεργειών αποθηκεύονται ως αρχείο .sql και χρησιμοποιούνται είτε για τη διαχείριση των βάσεων δεδομένων είτε για τη δημιουργία του σχήματος της βάσης δεδομένων κατά την ανάπτυξη μιας βάσης δεδομένων. Το SQLCMD εισήχθη με τον SQL Server 2005 και συνεχίστηκε με τις εκδόσεις του SQL Server 2008, 2008 R2, 2012, 2014, 2016 και 2019. Ο προκάτοχός του για τις προηγούμενες εκδόσεις ήταν οι OSQL και ISQL, οι οποίες ήταν λειτουργικά ισοδύναμες όσον αφορά την εκτέλεση TSQL, και πολλές από τις παραμέτρους της γραμμής εντολών είναι πανομοιότυπες, αν και το SQLCMD προσθέτει επιπλέον ευελιξία.

#### 4.5.10 SQL Server Management Studio

Το SQL Server Management Studio είναι ένα εργαλείο GUI που περιλαμβάνεται με τον SQL Server 2005 και μεταγενέστερες εκδόσεις για τη διαμόρφωση, διαχείριση και διαχείριση όλων των στοιχείων του Microsoft SQL Server. Το εργαλείο περιλαμβάνει τόσο επεξεργαστές σεναρίων όσο και γραφικά εργαλεία που εργάζονται με αντικείμενα και χαρακτηριστικά του διακομιστή. Το SQL Server Management Studio αντικαθιστά το Enterprise Manager ως την κύρια διεπαφή διαχείρισης για τον Microsoft SQL Server από τον SQL Server 2005 και μετά. Μια έκδοση του SQL Server Management Studio είναι επίσης διαθέσιμη για την έκδοση SQL Server Express Edition, για την οποία είναι γνωστή ως *SQL Server Management Studio Express* (SSMSE). Ένα κεντρικό χαρακτηριστικό του SQL Server Management Studio είναι η Εξερεύνηση αντικειμένων, η οποία επιτρέπει στο χρήστη να περιηγηθεί, να επιλέξει και να ενεργήσει σε οποιοδήποτε από τα αντικείμενα εντός του διακομιστή. Μπορεί να χρησιμοποιηθεί για την οπτική παρατήρηση και ανάλυση των σχεδίων ερωτημάτων και τη βελτιστοποίηση της απόδοσης της βάσης δεδομένων, μεταξύ άλλων. Το SQL Server Management Studio μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία μιας νέας βάσης δεδομένων, την τροποποίηση οποιουδήποτε υπάρχοντος σχήματος βάσης δεδομένων με την προσθήκη ή την τροποποίηση πινάκων και δεικτών ή την ανάλυση της απόδοσης. Περιλαμβάνει τα παράθυρα ερωτημάτων τα οποία παρέχουν μια διεπαφή βασισμένη σε γραφικό περιβάλλον εργασίας για τη συγγραφή και εκτέλεση ερωτημάτων.

### 4.5.11 Azure Data Studio

Το Azure Data Studio είναι ένας επεξεργαστής ερωτημάτων πολλαπλών πλατφορμών που διατίθεται ως προαιρετική λήψη. Το εργαλείο επιτρέπει στους χρήστες να γράφουν ερωτήματα, να εξάγουν αποτελέσματα ερωτημάτων, να δεσμεύουν σενάρια SQL σε αποθετήρια Git και να εκτελούν βασικές διαγνωστικές εργασίες διακομιστή. Το Azure Data Studio υποστηρίζει συστήματα Windows, Mac και Linux. Κυκλοφόρησε σε γενική διαθεσιμότητα τον Σεπτέμβριο του 2018. Πριν από την κυκλοφορία η έκδοση προεπισκόπησης της εφαρμογής ήταν γνωστή ως SQL Server Operations Studio.

### 4.5.12 Business Intelligence Development Studio

Το Business Intelligence Development Studio (BIDS) είναι το IDE της Microsoft που χρησιμοποιείται για την ανάπτυξη λύσεων ανάλυσης δεδομένων και επιχειρηματικής ευφυΐας με χρήση των Microsoft SQL Server Analysis Services, Reporting Services και Integration Services. Βασίζεται στο περιβάλλον ανάπτυξης του Microsoft Visual Studio, αλλά προσαρμόζεται με τις ειδικές επεκτάσεις και τύπους έργων για τις υπηρεσίες του SQL Server, συμπεριλαμβανομένων εργαλείων, στοιχείων ελέγχου και έργων για εκθέσεις (με χρήση των Reporting Services), κύβους και δομές εξόρυξης δεδομένων (με χρήση των Analysis Services). Για τον SQL Server 2012 και μεταγενέστερα, αυτό το IDE έχει μετονομαστεί σε SQL Server Data Tools (SSDT).

## 4.6 Εκδόσεις

Η Microsoft διαθέτει τον SQL Server σε πολλαπλές εκδόσεις, με διαφορετικά σύνολα χαρακτηριστικών και απευθυνόμενη σε διαφορετικούς χρήστες. Οι εκδόσεις αυτές είναι οι εξής:

### Κύριες Εκδόσεις

#### 1) Enterprise

Η έκδοση Enterprise του SQL Server περιλαμβάνει τόσο τη βασική μηχανή βάσης δεδομένων όσο και πρόσθετες υπηρεσίες, με μια σειρά εργαλείων για τη δημιουργία και τη διαχείριση ενός cluster του SQL Server. Μπορεί να διαχειριστεί βάσεις δεδομένων μεγέθους 524 petabytes και να απευθύνεται σε 12 terabytes μνήμης και υποστηρίζει 640 λογικούς επεξεργαστές (πυρήνες CPU).

## 2) Standard

Η έκδοση Standard του SQL Server περιλαμβάνει τον πυρήνα της μηχανής βάσεων δεδομένων, μαζί με τις αυτόνομες υπηρεσίες. Διαφέρει από την έκδοση Enterprise στο ότι υποστηρίζει λιγότερες ενεργές περιπτώσεις (αριθμός κόμβων σε ένα cluster) και δεν περιλαμβάνει ορισμένες λειτουργίες υψηλής διαθεσιμότητας, όπως η μνήμη hot-add (που επιτρέπει την προσθήκη μνήμης ενώ ο διακομιστής εξακολουθεί να λειτουργεί) και τα παράλληλα ευρετήρια.

## 3) Web

Η έκδοση Web του SQL Server είναι μια επιλογή χαμηλού κόστους-οφέλους για φιλοξενία στο Web.

## 4) Business Intelligence

Εισήχθη στον SQL Server 2012 και επικεντρώνεται στην αυτοεξυπηρέτηση και την εταιρική επιχειρηματική ευφυΐα. Περιλαμβάνει τις δυνατότητες της Standard Edition και τα εργαλεία Business Intelligence: Power Pivot, Power View, το σημασιολογικό μοντέλο BI, τις υπηρεσίες Master Data Services, τις υπηρεσίες ποιότητας δεδομένων και τις αναλύσεις μνήμης xVelocity.

## 5) Workgroup

Η έκδοση Workgroup του SQL Server περιλαμβάνει τις βασικές λειτουργίες της βάσης δεδομένων, αλλά δεν περιλαμβάνει τις πρόσθετες υπηρεσίες. Σημειώστε ότι αυτή η έκδοση έχει αποσυρθεί στον SQL Server 2012.

## 6) Express

Η έκδοση SQL Server Express είναι μια μειωμένη, δωρεάν έκδοση του SQL Server, η οποία περιλαμβάνει τη βασική μηχανή βάσης δεδομένων. Ενώ δεν υπάρχουν περιορισμοί στον αριθμό των βάσεων δεδομένων ή των χρηστών που υποστηρίζονται, περιορίζεται στη χρήση ενός επεξεργαστή, 1 GB μνήμης και 10 GB αρχείων βάσης δεδομένων (4 GB αρχείων βάσης δεδομένων πριν από τον SQL Server Express 2008 R2). Προορίζεται ως αντικαταστάτης του MSDE. Δύο πρόσθετες εκδόσεις παρέχουν ένα υπερσύνολο χαρακτηριστικών που δεν περιλαμβάνονται στην αρχική έκδοση Express Edition. Η πρώτη είναι η *SQL Server Express with Tools*, η οποία περιλαμβάνει το SQL Server Management Studio Basic. Η έκδοση *SQL Server Express with Advanced Services* προσθέτει δυνατότητα αναζήτησης πλήρους κειμένου και υπηρεσίες αναφοράς.

## Εξειδικευμένες εκδόσεις

### 1) Azure

Η Microsoft Azure SQL Database είναι η βασισμένη στο cloud έκδοση του Microsoft SQL Server, η οποία παρουσιάζεται ως προσφορά πλατφόρμας ως υπηρεσίας στο Microsoft Azure.

### 2) Azure MPP

Η Azure SQL Data Warehouse είναι η βασισμένη στο σύννεφο έκδοση του Microsoft SQL Server σε αρχιτεκτονική MPP (μαζικά παράλληλη επεξεργασία) για φόρτους εργασίας ανάλυσης, η οποία παρουσιάζεται ως προσφορά πλατφόρμας ως υπηρεσίας στο Microsoft Azure.

### 3) Compact (SQL CE)

Η Compact έκδοση είναι μια ενσωματωμένη μηχανή βάσης δεδομένων. Σε αντίθεση με τις άλλες εκδόσεις του SQL Server, η μηχανή SQL CE βασίζεται στην SQL Mobile (αρχικά σχεδιασμένη για χρήση σε φορητές συσκευές) και δεν μοιράζεται τα ίδια δυαδικά αρχεία. Λόγω του μικρού μεγέθους της (αποτύπωμα DLL 1 MB), έχει αισθητά μειωμένο σύνολο δυνατοτήτων σε σύγκριση με τις άλλες εκδόσεις. Για παράδειγμα, υποστηρίζει ένα υποσύνολο των τυποποιημένων τύπων δεδομένων, δεν υποστηρίζει αποθηκευμένες διαδικασίες ή προβολές ή δέσμες πολλαπλών δηλώσεων (μεταξύ άλλων περιορισμών). Περιορίζεται σε 4 GB μέγιστο μέγεθος βάσης δεδομένων και δεν μπορεί να εκτελεστεί ως υπηρεσία των Windows, η Compact Edition πρέπει να φιλοξενείται από την εφαρμογή που τη χρησιμοποιεί. Η έκδοση 3.5 περιλαμβάνει υποστήριξη για υπηρεσίες συγχρονισμού ADO.NET. Η SQL CE δεν υποστηρίζει συνδεσιμότητα ODBC, σε αντίθεση με τον πραγματικό SQL Server.

### 4) Developer

Η έκδοση SQL Server Developer περιλαμβάνει τις ίδιες δυνατότητες με την έκδοση SQL Server Enterprise Edition, αλλά περιορίζεται από την άδεια χρήσης ώστε να χρησιμοποιείται μόνο ως σύστημα ανάπτυξης και δοκιμών και όχι ως διακομιστής παραγωγής. Από τις αρχές του 2016, η Microsoft διέθεσε αυτή την έκδοση δωρεάν στο κοινό.

### 5) Embedded (SSEE)

Η SQL Server 2005 Embedded έκδοση είναι μια ειδικά διαμορφωμένη ονομασμένη περίπτωση της μηχανής βάσεων δεδομένων SQL Server Express, στην οποία μπορούν να έχουν πρόσβαση μόνο ορισμένες υπηρεσίες των Windows.

## 6) Evaluation

Η έκδοση Evaluation του SQL Server, γνωστή και ως *δοκιμαστική έκδοση*, διαθέτει όλες τις δυνατότητες της έκδοσης Enterprise Edition, αλλά περιορίζεται σε 180 ημέρες, μετά τις οποίες τα εργαλεία θα συνεχίσουν να εκτελούνται, αλλά οι υπηρεσίες του διακομιστή θα σταματήσουν.

## 7) Fast Track

Η έκδοση Fast Track του SQL Server προορίζεται ειδικά για αποθήκευση αποθηκών δεδομένων σε επιχειρησιακή κλίμακα και επεξεργασία επιχειρηματικής ευφυΐας και εκτελείται σε υλικό αρχιτεκτονικής αναφοράς που είναι βελτιστοποιημένο για την έκδοση Fast Track.

## 8) LocalDB

Εισήχθη στο SQL Server Express 2012, η LocalDB έκδοση είναι μια ελάχιστη, κατά παραγγελία, έκδοση του SQL Server που έχει σχεδιαστεί για προγραμματιστές εφαρμογών. Μπορεί επίσης να χρησιμοποιηθεί ως ενσωματωμένη βάση δεδομένων.

## 9) Analytics Platform Systems (APS)

Παλαιότερα ονομαζόντουσαν Parallel Data Warehouse (PDW). Είναι μια εφαρμογή μαζικής παράλληλης επεξεργασίας (MPP) του SQL Server βελτιστοποιημένη για αποθήκευση δεδομένων μεγάλης κλίμακας εκατοντάδων terabytes.

## 10) Datawarehouse Appliance

Προεγκατεστημένη και διαμορφωμένη ως μέρος μιας εφαρμογής σε συνεργασία με την Dell & HP με βάση την αρχιτεκτονική Fast Track. Αυτή η έκδοση δεν περιλαμβάνει τις υπηρεσίες SQL Server Integration Services, Analysis Services ή Reporting Services.sqlcmd

## 5. Εισαγωγή στην C#

Η C# (προφέρεται C sharp) είναι μια γενικού σκοπού, ασφαλής ως προς τον τύπο, αντικειμενοστραφής γλώσσα προγραμματισμού. Στόχος της γλώσσας είναι η παραγωγικότητα του προγραμματιστή. Για το σκοπό αυτό, η C# εξισορροπεί την απλότητα, την εκφραστικότητα και την απόδοση. Ο κύριος αρχιτέκτονας της γλώσσας από την πρώτη της έκδοση είναι ο Anders Hejlsberg (δημιουργός της Turbo Pascal και αρχιτέκτονας των Delphi). Η γλώσσα C# είναι ουδέτερη ως προς την πλατφόρμα και συνεργάζεται με μια σειρά από προγράμματα εκτέλεσης για συγκεκριμένες πλατφόρμες.





**Εικόνα 7.** Εικόνα της γλώσσας C#

Πηγή: <https://learn.microsoft.com/en-us/dotnet/csharp/>

## 5.1 Στόχοι σχεδιασμού της C#

Το πρότυπο ECMA παραθέτει τους παρακάτω στόχους σχεδιασμού για την C#:

- Η γλώσσα προορίζεται να είναι μια απλή, σύγχρονη, γενικής χρήσης, αντικειμενοστραφής γλώσσα προγραμματισμού.
- Η γλώσσα και οι υλοποιήσεις της θα πρέπει να παρέχουν υποστήριξη για αρχές μηχανικής λογισμικού, όπως έλεγχος ισχυρού τύπου, έλεγχος ορίων πίνακα, ανίχνευση προσπαθειών χρήσης μη αρχικοποιημένων μεταβλητών και αυτόματη συλλογή σκουπιδιών. Η στιβαρότητα του λογισμικού, η ανθεκτικότητα και η παραγωγικότητα του προγραμματιστή είναι σημαντικά.
- Η γλώσσα προορίζεται για χρήση στην ανάπτυξη στοιχείων λογισμικού κατάλληλων για ανάπτυξη σε καταναλωμένα περιβάλλοντα.
- Η φορητότητα είναι πολύ σημαντική για τον πηγαίο κώδικα και τους προγραμματιστές, ειδικά όσους είναι ήδη εξοικειωμένοι με τη C και τη C++.
- Η υποστήριξη για τη διεθνοποίηση είναι πολύ σημαντική.
- Το C# προορίζεται να είναι κατάλληλο για τη σύνταξη εφαρμογών τόσο για φιλοξενούμενα όσο και για ενσωματωμένα συστήματα, που κυμαίνονται από τα πολύ μεγάλα που χρησιμοποιούν εξελιγμένα λειτουργικά συστήματα, μέχρι τα πολύ μικρά με αποκλειστικές λειτουργίες.
- Αν και οι εφαρμογές C# προορίζονται να είναι οικονομικές όσον αφορά τις απαιτήσεις μνήμης και ισχύος επεξεργασίας, η γλώσσα δεν προοριζόταν να ανταγωνιστεί άμεσα την απόδοση και το μέγεθος με τη γλώσσα C ή assembly.

## 5.2 Ιστορική αναδρομή

Κατά την ανάπτυξη του .NET Framework, οι βιβλιοθήκες κλάσεων γράφτηκαν αρχικά χρησιμοποιώντας ένα σύστημα διαχειριζόμενου μεταγλωττιστή κώδικα που

ονομάζεται "Simple Managed C" (SMC). Τον Ιανουάριο του 1999, ο Anders Hejlsberg σχημάτισε μια ομάδα για να χτίσει μια νέα γλώσσα εκείνη την εποχή που ονομαζόταν Cool, η οποία σήμαινε "Γλώσσα αντικειμενοστρεφή που μοιάζει με C". Η Microsoft είχε σκεφτεί να διατηρήσει το όνομα "Cool" ως το τελικό όνομα της γλώσσας, αλλά επέλεξε να μην το κάνει για λόγους εμπορικού σήματος. Μέχρι τη στιγμή που το έργο .NET ανακοινώθηκε δημόσια στο Συνέδριο Επαγγελματιών Προγραμματιστών τον Ιούλιο του 2000, η γλώσσα είχε μετονομαστεί σε C# και οι βιβλιοθήκες κλάσεων και ο χρόνος εκτέλεσης του ASP.NET είχαν μεταφερθεί σε C#. Ο Hejlsberg είναι ο κύριος σχεδιαστής και ο κύριος αρχιτέκτονας της C# στη Microsoft και στο παρελθόν είχε ασχοληθεί με το σχεδιασμό των Turbo Pascal, Embarcadero Delphi (πρώην CodeGear Delphi, Inprise Delphi και Borland Delphi) και Visual J++. Σε συνεντεύξεις και τεχνικές εργασίες έχει δηλώσει ότι τα ελαττώματα στις περισσότερες μεγάλες γλώσσες προγραμματισμού (π.χ. C++, Java, Delphi και Smalltalk) οδήγησαν στις βασικές αρχές του Common Language Runtime (CLR), το οποίο, με τη σειρά του, οδήγησε στη σχεδίαση η ίδια η γλώσσα C#.

### 5.3 Σύνταξη

Η βασική σύνταξη της γλώσσας C# είναι παρόμοια με αυτή άλλων γλωσσών τύπου C όπως η C, η C++ και η Java, ιδιαίτερα:

- Τα ερωτηματικά χρησιμοποιούνται για να δηλώσουν το τέλος μιας πρότασης.
- Οι σγουρές αγκύλες χρησιμοποιούνται για την ομαδοποίηση των δηλώσεων. Οι δηλώσεις συνήθως ομαδοποιούνται σε μεθόδους (συναρτήσεις), μεθόδους σε κλάσεις και κλάσεις σε χώρους ονομάτων.
- Οι μεταβλητές εκχωρούνται χρησιμοποιώντας ένα σύμβολο ίσου, αλλά συγκρίνονται χρησιμοποιώντας δύο διαδοχικά σύμβολα ίσου.
- Οι αγκύλες χρησιμοποιούνται με πίνακες, τόσο για να τους δηλώσουν όσο και για να λάβουν μια τιμή σε ένα δεδομένο δείκτη σε έναν από αυτούς.

### 5.4 Χαρακτηριστικά

Μερικά αξιοσημείωτα χαρακτηριστικά της C# που τη διακρίνουν από την C, τη C++ και την Java όπου σημειώνεται, είναι:

#### 1) Φορητότητα

Από την σχεδίαση της ακόμα, η C# είναι η γλώσσα προγραμματισμού που αντικατοπτρίζει πιο άμεσα την υποκείμενη Κοινή Γλωσσική Υποδομή (CLI). Οι περισσότεροι από τους εγγενείς τύπους του αντιστοιχούν σε τύπους τιμών που υλοποιούνται από το πλαίσιο CLI. Ωστόσο, η προδιαγραφή γλώσσας δεν αναφέρει τις απαιτήσεις δημιουργίας κώδικα του μεταγλωττιστή: δηλαδή, δεν δηλώνει ότι ένας



μεταγλωττιστής C# πρέπει να στοχεύει έναν χρόνο εκτέλεσης κοινής γλώσσας ή να δημιουργεί κοινή ενδιάμεση γλώσσα (CIL) ή να δημιουργεί οποιαδήποτε άλλη συγκεκριμένη μορφή. Θεωρητικά, ένας μεταγλωττιστής C# θα μπορούσε να δημιουργήσει κώδικα μηχανής όπως οι παραδοσιακοί μεταγλωττιστές της C++ ή του Fortran.

## 2) Τύποι δεδομένων

Η C# υποστηρίζει ισχυρά, σιωπηρά πληκτρολογημένες δηλώσεις μεταβλητών με τη λέξη-κλειδί `var` και σιωπηρά πληκτρολογημένους πίνακες με τη λέξη-κλειδί `new[]` που ακολουθείται από έναν αρχικοποιητή συλλογής. Υποστηρίζει επίσης έναν αυστηρό τύπο δεδομένων `Boolean`, `bool`. Οι δηλώσεις που λαμβάνουν συνθήκες, όπως `while` και `if`, απαιτούν μια έκφραση ενός τύπου που υλοποιεί τον αληθινό τελεστή, όπως ο τύπος `Boolean`. Ενώ η C++ έχει επίσης τύπο `Boolean`, μπορεί να μετατραπεί ελεύθερα από και προς ακέραιους αριθμούς και εκφράσεις όπως το `if (a)` απαιτούν μόνο το `a` να είναι μετατρέψιμο σε `bool`, επιτρέποντας στο `a` να είναι `int` ή δείκτης. Η C# δεν επιτρέπει αυτήν την προσέγγιση "ακέραιος που σημαίνει αληθής ή ψευδής", με το σκεπτικό ότι ο εξαναγκασμός των προγραμματιστών να χρησιμοποιούν εκφράσεις που επιστρέφουν ακριβώς `bool` μπορεί να αποτρέψει ορισμένους τύπους σφαλμάτων προγραμματισμού, όπως εάν `(a = b)` (χρήση ανάθεσης = αντί ισότητας `==`). Η C# είναι πιο ασφαλές από το C++. Οι μόνες σιωπηρές μετατροπές από προεπιλογή είναι αυτές που θεωρούνται ασφαλείς, όπως η διεύρυνση ακεραίων. Αυτό επιβάλλεται κατά το χρόνο μεταγλώττισης, κατά τη διάρκεια του JIT και, σε ορισμένες περιπτώσεις, κατά το χρόνο εκτέλεσης. Δεν γίνονται σιωπηρές μετατροπές μεταξύ `Booleans` και ακεραίων, ούτε μεταξύ μελών απαρίθμησης και ακεραίων (εκτός από το κυριολεκτικό `0`, το οποίο μπορεί σιωπηρά να μετατραπεί σε οποιονδήποτε απαριθμημένο τύπο). Οποιαδήποτε μετατροπή που ορίζεται από το χρήστη πρέπει να επισημαίνεται ρητά ως ρητή ή σιωπηρή, σε αντίθεση με τους κατασκευαστές αντιγράφων C++ και τους τελεστές μετατροπής, οι οποίοι είναι και οι δύο σιωπηροί από προεπιλογή. Η C# έχει ρητή υποστήριξη για συνδιακύμανση και αντιδιακύμανση σε γενικούς τύπους, σε αντίθεση με τη C++ που έχει κάποιο βαθμό υποστήριξης για αντιδιακύμανση απλώς μέσω της σημασιολογίας των τύπων επιστροφής σε εικονικές μεθόδους. Τα μέλη της απαρίθμησης τοποθετούνται στη δική τους εμβέλεια. Η C# δεν επιτρέπει καθολικές μεταβλητές ή συναρτήσεις. Όλες οι μέθοδοι και τα μέλη πρέπει να δηλώνονται εντός κλάσεων. Τα στατικά μέλη των δημόσιων κλάσεων μπορούν να υποκαταστήσουν καθολικές μεταβλητές και συναρτήσεις. Οι τοπικές μεταβλητές δεν μπορούν να σκιάσουν τις μεταβλητές του μπλοκ που περικλείει, σε αντίθεση με τη C και τη C++.

## 3) Μεταπρογραμματισμός

Ο μεταπρογραμματισμός μπορεί να επιτευχθεί με διάφορους τρόπους:

- Ο αντικατοπτρισμός υποστηρίζεται μέσω των API .NET, τα οποία επιτρέπουν σενάρια όπως η επιθεώρηση μεταδεδομένων τύπου και η επίκληση δυναμικής μεθόδου.
- Τα δέντρα έκφρασης αντιπροσωπεύουν τον κώδικα ως αφηρημένο δέντρο σύνταξης, όπου κάθε κόμβος είναι μια έκφραση που μπορεί να επιθεωρηθεί ή να εκτελεστεί. Αυτό επιτρέπει τη δυναμική τροποποίηση του εκτελέσιμου κώδικα κατά το χρόνο εκτέλεσης. Τα δέντρα έκφρασης εισάγουν κάποια ομοιονικότητα στη γλώσσα.
- Τα χαρακτηριστικά είναι μεταδεδομένα που μπορούν να προσαρτηθούν σε τύπους, μέλη ή ολόκληρες συγκροτήσεις, ισοδύναμα με σχολιασμούς στην Java. Τα χαρακτηριστικά είναι προσβάσιμα τόσο στον μεταγλωττιστή όσο και στον κώδικα μέσω της ανάκλασης. Πολλά από αυτά τα χαρακτηριστικά αντιγράφουν τη λειτουργικότητα των οδηγιών προεπεξεργαστή του GCC και του VisualC++ που εξαρτώνται από την πλατφόρμα.
- Η πλατφόρμα μεταγλωττιστή .NET (Roslyn) παρέχει πρόσβαση API σε υπηρεσίες μεταγλώττισης γλωσσών, επιτρέποντας τη μεταγλώττιση κώδικα C# μέσα από εφαρμογές .NET. Εκθέτει API για συντακτική (λεξική) ανάλυση κώδικα, σημασιολογική ανάλυση, δυναμική μεταγλώττιση σε CIL και εκπομπή κώδικα.
- Οι γεννήτριες πηγών, ένα χαρακτηριστικό του μεταγλωττιστή Roslyn C#, επιτρέπουν τον μεταπρογραμματισμό χρόνου μεταγλώττισης. Κατά τη διάρκεια της διαδικασίας μεταγλώττισης, οι προγραμματιστές μπορούν να επιθεωρήσουν τον κώδικα που μεταγλωττίζεται με το API του μεταγλωττιστή και να περάσουν τον πρόσθετο δημιουργημένο πηγαίο κώδικα C# για μεταγλώττιση.

#### 4) Μεθόδοι και λειτουργίες

Μια μέθοδος στη C# είναι ένα μέλος μιας κλάσης που μπορεί να κληθεί ως συνάρτηση (μια ακολουθία εντολών), αντί για απλή ικανότητα διατήρησης τιμής μιας ιδιότητας κλάσης. Όπως και σε άλλες συντακτικά παρόμοιες γλώσσες, όπως η C++ και η ANSI C, η υπογραφή μιας μεθόδου είναι μια δήλωση που περιλαμβάνει κατά σειρά: τυχόν προαιρετικές λέξεις-κλειδιά προσβασιμότητας (όπως ιδιωτικές), τη ρητή προδιαγραφή του τύπου επιστροφής της (όπως int ή λέξη-κλειδί void εάν δεν επιστραφεί τιμή), το όνομα της μεθόδου και, τέλος, μια ακολουθία προδιαγραφών παραμέτρων διαχωρισμένων με κόμματα σε παρένθεση, καθεμία από τις οποίες αποτελείται από τον τύπο μιας παραμέτρου, το επίσημο όνομά της και προαιρετικά, μια προεπιλεγμένη τιμή που θα χρησιμοποιείται όποτε δεν υπάρχει καμία υπό την προϋπόθεση. Ορισμένα συγκεκριμένα είδη μεθόδων, όπως αυτά που απλώς λαμβάνουν ή ορίζουν μια ιδιότητα κλάσης με επιστροφή τιμής ή ανάθεσης, δεν απαιτούν πλήρη υπογραφή, αλλά στη γενική περίπτωση, ο ορισμός μιας κλάσης περιλαμβάνει τη δήλωση πλήρους υπογραφής των μεθόδων της. Όπως η C++, και σε αντίθεση με την Java, οι προγραμματιστές της C# πρέπει να χρησιμοποιούν τη λέξη-κλειδί τροποποιητή εμβέλειας εικονικό για να επιτρέπεται η παράκαμψη των μεθόδων από υποκλάσεις. Οι μέθοδοι επέκτασης στη C#

επιτρέπουν στους προγραμματιστές να χρησιμοποιούν στατικές μεθόδους σαν να ήταν μέθοδοι από τον πίνακα μεθόδων μιας κλάσης, επιτρέποντας στους προγραμματιστές να προσθέτουν μεθόδους σε ένα αντικείμενο που πιστεύουν ότι πρέπει να υπάρχουν σε αυτό το αντικείμενο και τα παράγωγά του. Η δυναμική τύπου επιτρέπει τη δέσμευση της μεθόδου χρόνου εκτέλεσης, επιτρέποντας κλήσεις μεθόδων τύπου JavaScript και σύνθεση αντικειμένων χρόνου εκτέλεσης. Η C# έχει υποστήριξη για δείκτες συναρτήσεων με έντονη πληκτρολόγηση μέσω της λέξης-κλειδιού `plhresousios`. Όπως το σήμα και η υποδοχή ψευδο-C++ του πλαισίου Qt, η C# έχει σημασιολογικά στοιχεία που περιβάλλουν συγκεκριμένα συμβάντα στυλ δημοσίευσης-εγγραφής, αν και η C# χρησιμοποιεί εκπροσώπους για να το κάνει.

## 5) Ιδιότητες

Η C# υποστηρίζει κλάσεις με ιδιότητες. Οι ιδιότητες μπορεί να είναι απλές συναρτήσεις πρόσβασης με πεδίο υποστήριξης ή να υλοποιούν συναρτήσεις λήψης και ρύθμισης.

## 6) Namespaces

Ένα Namespace στην C# παρέχει το ίδιο επίπεδο απομόνωσης κώδικα με ένα πακέτο Java ή ένα χώρο ονομάτων C++, με πολύ παρόμοιους κανόνες και δυνατότητες με ένα πακέτο. Οι χώροι ονομάτων μπορούν να εισαχθούν με τη σύνταξη "using".

## 7) Πρόσβαση στην μνήμη

Στην C#, οι δείκτες διεύθυνσης μνήμης μπορούν να χρησιμοποιηθούν μόνο μέσα σε μπλοκ που επισημαίνονται ειδικά ως μη ασφαλή και τα προγράμματα με μη ασφαλή κώδικα χρειάζονται τα κατάλληλα δικαιώματα για να εκτελεστούν. Το μεγαλύτερο μέρος της πρόσβασης σε αντικείμενα γίνεται μέσω αναφορών ασφαλών αντικειμένων, οι οποίες πάντα είτε δείχνουν σε ένα "ζωντανό" αντικείμενο είτε έχουν την καλά καθορισμένη μηδενική τιμή. είναι αδύνατο να λάβετε μια αναφορά σε ένα "νεκρό" αντικείμενο (ένα που έχει συλλεχθεί σκουπίδια) ή σε ένα τυχαίο μπλοκ μνήμης. Ένας μη ασφαλής δείκτης μπορεί να υποδεικνύει μια παρουσία ενός τύπου τιμής "μη διαχειριζόμενη" που δεν περιέχει αναφορές σε αντικείμενα που συλλέγονται από σκουπίδια, πίνακα, συμβολοσειρά ή μπλοκ μνήμης εκχωρημένης στοίβας. Ο κώδικας που δεν έχει επισημανθεί ως μη ασφαλής μπορεί να αποθηκεύσει και να χειριστεί δείκτες μέσω του τύπου `System.IntPtr`, αλλά δεν μπορεί να τους αποαναφέρει. Η διαχειριζόμενη μνήμη δεν μπορεί να ελευθερωθεί ρητά. Αντίθετα, συλλέγονται αυτόματα σκουπίδια. Η συλλογή σκουπιδιών αντιμετωπίζει το πρόβλημα των διαρροών μνήμης απαλλάσσοντας τον προγραμματιστή από την ευθύνη για την απελευθέρωση μνήμης που δεν χρειάζεται πλέον στις περισσότερες περιπτώσεις. Ο κώδικας που διατηρεί αναφορές σε αντικείμενα περισσότερο από ό,τι απαιτείται μπορεί να εξακολουθεί να έχει μεγαλύτερη χρήση μνήμης από ό,τι χρειάζεται, ωστόσο μόλις απελευθερωθεί η τελική αναφορά σε ένα

αντικείμενο, η μνήμη είναι διαθέσιμη για συλλογή σκουπιδιών.

## 8) Εξαιρέσεις

Μια σειρά τυπικών εξαιρέσεων είναι διαθέσιμη στους προγραμματιστές. Οι μέθοδοι σε τυπικές βιβλιοθήκες δημιουργούν τακτικά εξαιρέσεις συστήματος σε ορισμένες περιπτώσεις και το εύρος των εξαιρέσεων που εισάγονται συνήθως τεκμηριώνεται. Οι προσαρμοσμένες κλάσεις εξαίρεσης μπορούν να οριστούν για κλάσεις που επιτρέπουν την εφαρμογή χειρισμού για συγκεκριμένες περιστάσεις, όπως απαιτείται. Οι επιλεγμένες εξαιρέσεις δεν υπάρχουν στη C# (σε αντίθεση με την Java). Αυτή ήταν μια συνειδητή απόφαση που βασίζεται στα ζητήματα της επεκτασιμότητας και της δυνατότητας έκδοσης.

## 9) Πολυμορφισμός

Σε αντίθεση με τη C++, η C# δεν υποστηρίζει πολλαπλή κληρονομικότητα, αν και μια κλάση μπορεί να υλοποιήσει οποιονδήποτε αριθμό "διεπαφών" (πλήρως αφηρημένες κλάσεις). Αυτή ήταν μια σχεδιαστική απόφαση από τον κύριο αρχιτέκτονα της γλώσσας για την αποφυγή επιπλοκών και την απλοποίηση των αρχιτεκτονικών απαιτήσεων σε όλο το CLI. Κατά την υλοποίηση πολλαπλών διεπαφών που περιέχουν μια μέθοδο με το ίδιο όνομα και λαμβάνοντας παραμέτρους του ίδιου τύπου με την ίδια σειρά (δηλαδή την ίδια υπογραφή), παρόμοια με την Java, η C# επιτρέπει τόσο μια μεμονωμένη μέθοδο να καλύπτει όλες τις διεπαφές και, εάν είναι απαραίτητο, συγκεκριμένες μεθόδους για κάθε διεπαφή. Ωστόσο, σε αντίθεση με την Java, η C# υποστηρίζει υπερφόρτωση τελεστών.

## 10) Language Integrated Query (LINQ)

Η C# έχει τη δυνατότητα να χρησιμοποιεί το LINQ μέσω του .NET Framework. Ένας προγραμματιστής μπορεί να ρωτήσει μια ποικιλία πηγών δεδομένων, με την προϋπόθεση ότι η διεπαφή `IEnumerable<T>` έχει εφαρμοστεί στο αντικείμενο. Αυτό περιλαμβάνει έγγραφα XML, ένα σύνολο δεδομένων ADO.NET και βάσεις δεδομένων SQL. Η χρήση του LINQ στην C# προσφέρει πλεονεκτήματα όπως υποστήριξη IntelliSense, ισχυρές δυνατότητες φιλτραρίσματος, ασφάλεια τύπου με δυνατότητα ελέγχου σφαλμάτων μεταγλώττισης και συνέπεια για την αναζήτηση δεδομένων σε διάφορες πηγές. Υπάρχουν πολλές διαφορετικές δομές γλώσσας που μπορούν να χρησιμοποιηθούν με το C# και το LINQ και είναι εκφράσεις ερωτήματος, εκφράσεις λάμδα, ανώνυμοι τύποι, σιωπηρά πληκτρολογημένες μεταβλητές, μέθοδοι επέκτασης και αρχικοποιητές αντικειμένων.

## 11) Λειτουργικός προγραμματισμός

Αν και ήταν κυρίως επιτακτική γλώσσα, η C# 2.0 προσέφερε περιορισμένη υποστήριξη για λειτουργικό προγραμματισμό μέσω λειτουργιών πρώτης κατηγορίας και

κλεισίματος με τη μορφή ανώνυμων εκπροσώπων. Η C# 3.0 επέκτεινε την υποστήριξη για λειτουργικό προγραμματισμό με την εισαγωγή μιας ελαφριάς σύνταξης για εκφράσεις λάμδα, μεθόδων επέκτασης (μια οικονομική δυνατότητα για ενότητες) και μια σύνταξη κατανόησης λίστας με τη μορφή γλώσσας "κατανόησης ερωτημάτων". Η C# 7.0 προσθέτει χαρακτηριστικά που βρίσκονται συνήθως σε λειτουργικές γλώσσες όπως πλειάδες, τοπικές συναρτήσεις και αντιστοίχιση προτύπων. Η C# 9.0 εισάγει τη δυνατότητα εγγραφής η οποία έχει κατασκευαστεί κυρίως για την καλύτερη υποστήριξη μοντέλων αμετάβλητων δεδομένων.

## 5.5 Τύποι δεδομένων C#

Οι τύποι δεδομένων καθορίζουν τον τύπο δεδομένων που μπορεί να περιέχει μια έγκυρη μεταβλητή της C#. Η C# είναι μια έντονα τυποποιημένη γλώσσα προγραμματισμού επειδή στη C#, κάθε τύπος δεδομένων (όπως ακέραιος, χαρακτήρας, float κ.ο.κ.) είναι προκαθορισμένος ως μέρος της γλώσσας προγραμματισμού και όλες οι σταθερές ή μεταβλητές που ορίζονται για ένα συγκεκριμένο πρόγραμμα πρέπει να περιγράφονται με έναν από τους τύπους δεδομένων. Οι τύποι δεδομένων στη C# χωρίζονται κυρίως σε τρεις κατηγορίες:

- Τύποι δεδομένων τιμής
- Τύποι δεδομένων αναφοράς
- Τύποι δεδομένων δείκτη

### 1. Τύποι δεδομένων τιμής

- Οι προκαθορισμένοι τύποι δεδομένων είναι τύποι δεδομένων που είναι αποθηκευμένοι στο παγκόσμιο αποθετήριο, όπως Integer, Float, Boolean κ.λπ.
- Οι τύποι δεδομένων που ορίζονται από τον χρήστη είναι τύποι δεδομένων που προέρχονται από έναν υπάρχοντα τύπο δεδομένων, όπως οι δομές, οι απαριθμήσεις κ.λπ.

### 2. Τύποι δεδομένων αναφοράς

- Προκαθορισμένοι τύποι όπως αντικείμενα και συμβολοσειρές
- Τύποι που ορίζονται από τον χρήστη, όπως οι διεπαφές και οι κλάσεις

### 3. Τύποι δεδομένων δείκτη

Ο δείκτης στη C# είναι μια μεταβλητή, επίσης είναι γνωστή ως locator ή indicator που δείχνει σε μια διεύθυνση μιας τιμής.

## 5.6 Βιβλιοθήκες

Οι προδιαγραφές της C# περιγράφει λεπτομερώς ένα ελάχιστο σύνολο τύπων και βιβλιοθηκών κλάσεων που ο μεταγλωττιστής αναμένει να έχει διαθέσιμες. Στην πράξη, η C# χρησιμοποιείται συχνότερα με κάποια υλοποίηση του CLI. Επιπλέον, υπάρχουν πολλές εμπορικές και κοινοτικές βιβλιοθήκες που βασίζονται πάνω στις βιβλιοθήκες του .NET Framework για να παρέχουν πρόσθετη λειτουργικότητα. Η C# μπορεί να πραγματοποιήσει κλήσεις σε οποιαδήποτε βιβλιοθήκη περιλαμβάνεται στη λίστα βιβλιοθηκών του .NET Framework.

## 6. Εισαγωγή στην SQL

Η Structured Query Language (SQL) είναι μια γλώσσα ειδικού τομέα που σημαίνει ότι είναι μια γλώσσα εξειδικευμένη σε έναν συγκεκριμένο τομέα εφαρμογής. Η SQL χρησιμοποιείται στον προγραμματισμό και έχει σχεδιαστεί για τη διαχείριση δεδομένων που τηρούνται σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ή για την επεξεργασία ροής δεδομένων σε ένα σύστημα διαχείρισης ροής σχεσιακών δεδομένων (RDSMS). Είναι ιδιαίτερα χρήσιμη στη διαχείριση δομημένων δεδομένων, δηλαδή δεδομένων που ενσωματώνουν σχέσεις μεταξύ οντοτήτων και μεταβλητών. Η SQL προσφέρει δύο βασικά πλεονεκτήματα σε σχέση με τα παλαιότερα API ανάγνωσης-εγγραφής, όπως το ISAM ή το VSAM. Πρώτον, εισήγαγε την έννοια της πρόσβασης σε πολλές εγγραφές με μία μόνο εντολή. Δεύτερον, εξαλείφει την ανάγκη καθορισμού *του τρόπου* πρόσβασης σε μια εγγραφή, π.χ. με ή χωρίς δείκτη. Αρχικά βασισμένη στη σχεσιακή άλγεβρα και στον σχεσιακό λογισμό πλειάδων, η SQL αποτελείται από πολλούς τύπους δηλώσεων, οι οποίες μπορούν ανεπίσημα να ταξινομηθούν ως υπογλώσσες, συνήθως: γλώσσα ερωτημάτων δεδομένων (DQL), γλώσσα ορισμού δεδομένων (DDL), γλώσσα ελέγχου δεδομένων (DCL) και γλώσσα χειρισμού δεδομένων (DML). Το πεδίο εφαρμογής της SQL περιλαμβάνει ερωτήματα δεδομένων, χειρισμό δεδομένων (εισαγωγή, ενημέρωση και διαγραφή), ορισμό δεδομένων (δημιουργία και τροποποίηση σχήματος) και έλεγχο πρόσβασης σε δεδομένα. Αν και η SQL είναι ουσιαστικά μια δηλωτική γλώσσα (4GL), περιλαμβάνει επίσης διαδικαστικά στοιχεία. Η SQL ήταν μία από τις πρώτες εμπορικές γλώσσες που χρησιμοποίησαν το σχεσιακό μοντέλο του Edgar F. Codd. Το μοντέλο περιγράφηκε στο σημαντικό έγγραφο του 1970, "A Relational Model of Data for Large Shared Data Banks" (Ένα σχεσιακό μοντέλο δεδομένων για μεγάλες κοινές τράπεζες δεδομένων). Παρά το γεγονός ότι δεν ακολουθούσε πλήρως το σχεσιακό μοντέλο όπως περιγράφηκε από τον Codd, έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα βάσεων δεδομένων. Η SQL έγινε πρότυπο του Αμερικανικού Ινστιτούτου Εθνικών Προτύπων (ANSI) το 1986 και του Διεθνούς Οργανισμού Τυποποίησης (ISO) το 1987. Έκτοτε, το πρότυπο έχει αναθεωρηθεί για να συμπεριλάβει ένα μεγαλύτερο σύνολο χαρακτηριστικών. Παρά την ύπαρξη προτύπων, ο περισσότερος κώδικας SQL απαιτεί τουλάχιστον κάποιες αλλαγές πριν μεταφερθεί σε διαφορετικά συστήματα βάσεων δεδομένων.

## 6.1 Ιστορική αναδρομή

Η SQL αναπτύχθηκε αρχικά στην IBM από τους Donald D. Chamberlin και Raymond F. Boyce αφού έμαθαν για το σχεσιακό μοντέλο από τον Edgar F. Codd στις αρχές της δεκαετίας του 1970. Αυτή η έκδοση, που αρχικά ονομάστηκε SEQUEL (Structured English Query Language), σχεδιάστηκε για να χειρίζεται και να ανακτά δεδομένα που ήταν αποθηκευμένα στο αρχικό σύστημα διαχείρισης οιονεί σειριακών βάσεων δεδομένων της IBM, το System R, το οποίο είχε αναπτύξει μια ομάδα στο ερευνητικό εργαστήριο San Jose της IBM κατά τη διάρκεια της δεκαετίας του 1970. Η πρώτη απόπειρα των Chamberlin και Boyce για μια σχεσιακή γλώσσα βάσεων δεδομένων ήταν η SQUARE (Specifying Queries in A Relational Environment), αλλά ήταν δύσκολο να χρησιμοποιηθεί λόγω της σημειογραφίας με δείκτες/υπερδείκτες. Αφού μετακόμισαν στο ερευνητικό εργαστήριο του San Jose το 1973, άρχισαν να εργάζονται πάνω σε μια συνέχεια της SQUARE. Το όνομα SEQUEL άλλαξε αργότερα σε SQL (αφήνοντας τα φωνήεντα) επειδή το "SEQUEL" ήταν εμπορικό σήμα της εταιρείας Hawker Siddeley Dynamics Engineering Limited με έδρα το Ηνωμένο Βασίλειο. Η ετικέτα SQL έγινε αργότερα το ακρωνύμιο για τη Structured Query Language (Γλώσσα δομημένων ερωτήσεων). Αφού δοκίμασε την SQL σε δοκιμαστικούς χώρους πελατών για να καθορίσει τη χρησιμότητα και την πρακτικότητα του συστήματος, η IBM άρχισε να αναπτύσσει εμπορικά προϊόντα βασισμένα στο πρωτότυπο System R, συμπεριλαμβανομένων των System/38, SQL/DS και IBM Db2, τα οποία ήταν εμπορικά διαθέσιμα το 1979, το 1981 και το 1983, αντίστοιχα. Στα τέλη της δεκαετίας του 1970, η Relational Software, Inc. (σήμερα Oracle Corporation) είδε τις δυνατότητες των εννοιών που περιέγραψαν οι Codd, Chamberlin και Boyce και ανέπτυξε το δικό της RDBMS βασισμένο στην SQL με την προοπτική να το πουλήσει στο Ναυτικό των ΗΠΑ, στην Κεντρική Υπηρεσία Πληροφοριών και σε άλλες κυβερνητικές υπηρεσίες των ΗΠΑ. Τον Ιούνιο του 1979, η Relational Software παρουσίασε μια από τις πρώτες εμπορικά διαθέσιμες υλοποιήσεις της SQL, την Oracle V2 (Version2) για υπολογιστές VAX. Μέχρι το 1986, οι ομάδες προτύπων ANSI και ISO υιοθέτησαν επίσημα τον τυποποιημένο ορισμό της γλώσσας "Γλώσσα βάσεων δεδομένων SQL". Νέες εκδόσεις του προτύπου δημοσιεύθηκαν το 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011 και πιο πρόσφατα, το 2016.

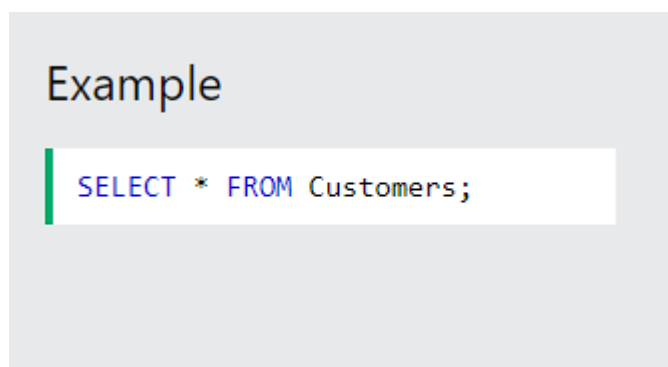
## 6.2 Σύνταξη

Η γλώσσα SQL υποδιαιρείται σε διάφορα γλωσσικά στοιχεία, μεταξύ των οποίων:

- Ρήτρες, οι οποίες αποτελούν συστατικά στοιχεία των δηλώσεων και των ερωτημάτων. (Σε ορισμένες περιπτώσεις, αυτές είναι προαιρετικές.)
- Εκφράσεις, οι οποίες μπορούν να παράγουν είτε κλιμακωτές τιμές είτε πίνακες που αποτελούνται από στήλες και γραμμές δεδομένων
- Προδιαγραφές, οι οποίες καθορίζουν συνθήκες που μπορούν να αξιολογηθούν σε τιμές αλήθειας τριών τιμών λογικής SQL (3VL) (αληθές/ψευδές/άγνωστο) ή

Boolean και χρησιμοποιούνται για τον περιορισμό των αποτελεσμάτων των εντολών και των ερωτημάτων ή για την αλλαγή της ροής του προγράμματος.

- Ερωτήματα, τα οποία ανακτούν τα δεδομένα βάσει συγκεκριμένων κριτηρίων. Αυτό είναι ένα σημαντικό στοιχείο της SQL.
- Δηλώσεις, οι οποίες μπορεί να έχουν μόνιμη επίδραση σε σχήματα και δεδομένα ή να ελέγχουν συναλλαγές, ροή προγράμματος, συνδέσεις, συνεδρίες ή διαγνωστικά.
- Οι εντολές SQL περιλαμβάνουν επίσης τον τερματισμό της εντολής με άνω τελεία (";"). Αν και δεν απαιτείται σε κάθε πλατφόρμα, ορίζεται ως τυπικό μέρος της γραμματικής της SQL.
- Τα ασήμαντα κενά διαστήματα γενικά αγνοούνται στις εντολές και τα ερωτήματα SQL, διευκολύνοντας τη μορφοποίηση του κώδικα SQL για αναγνωσιμότητα.



Εικόνα 8. Παράδειγμα ενός απλού SQL Query

Πηγή: [https://www.w3schools.com/sql/sql\\_syntax.asp](https://www.w3schools.com/sql/sql_syntax.asp)

### 6.3 Διαλειτουργικότητα

Οι υλοποιήσεις της SQL είναι ασύμβατες μεταξύ των προμηθευτών και δεν ακολουθούν απαραιτήτως πλήρως τα πρότυπα. Ειδικότερα, η σύνταξη ημερομηνίας και ώρας, η συνένωση συμβολοσειρών, τα NULL και η ευαισθησία στην πεζότητα της σύγκρισης διαφέρουν από προμηθευτή σε προμηθευτή. Ιδιαίτερες εξαιρέσεις αποτελούν η PostgreSQL και η Mimer SQL οι οποίες επιδιώκουν τη συμμόρφωση με τα πρότυπα, αν και η PostgreSQL δεν τηρεί το πρότυπο σε όλες τις περιπτώσεις. Για παράδειγμα, η αναδίπλωση των ονομάτων χωρίς εισαγωγικά σε πεζά γράμματα στην PostgreSQL δεν είναι συμβατή με το πρότυπο SQL, το οποίο λέει ότι τα ονόματα χωρίς εισαγωγικά πρέπει να αναδιπλώνονται σε κεφαλαία γράμματα. Έτσι, το Foo θα έπρεπε να ισοδυναμεί με FOO και όχι με foo σύμφωνα με το πρότυπο. Οι δημοφιλείς υλοποιήσεις της SQL συνήθως παραλείπουν την υποστήριξη βασικών χαρακτηριστικών της Standard SQL, όπως οι τύποι δεδομένων DATE ή TIME. Τα πιο προφανή τέτοια παραδείγματα, και παρεμπιπτόντως τα πιο δημοφιλή εμπορικά και ιδιόκτητα ΣΔΒΔ SQL, είναι η Oracle (της οποίας η DATE συμπεριφέρεται ως DATETIME, και δεν διαθέτει τύπο TIME) και ο MS SQL Server (πριν από την έκδοση 2008). Ως αποτέλεσμα, ο κώδικας SQL σπάνια μπορεί



να μεταφερθεί μεταξύ συστημάτων βάσεων δεδομένων χωρίς τροποποιήσεις.

### 6.3.1 Λόγοι ασυμβατότητας

Διάφοροι λόγοι για την έλλειψη συμβατότητας μεταξύ συστημάτων βάσεων δεδομένων περιλαμβάνουν:

- Η πολυπλοκότητα και το μέγεθος του προτύπου SQL σημαίνει ότι οι περισσότεροι φορείς υλοποίησης δεν υποστηρίζουν ολόκληρο το πρότυπο.
- Το πρότυπο δεν καθορίζει τη συμπεριφορά της βάσης δεδομένων σε διάφορους σημαντικούς τομείς (π.χ. δείκτες, αποθήκευση αρχείων...), αφήνοντας τις υλοποιήσεις να αποφασίσουν πώς θα συμπεριφερθούν.
- Το πρότυπο SQL προσδιορίζει επακριβώς τη σύνταξη που πρέπει να εφαρμόζει ένα σύστημα βάσεων δεδομένων που συμμορφώνεται. Ωστόσο, οι προδιαγραφές του προτύπου για τη σημασιολογία των γλωσσικών δομών είναι λιγότερο καλά καθορισμένες, γεγονός που οδηγεί σε ασάφεια.
- Όταν η νεότερη έκδοση του προτύπου SQL έρχεται σε σύγκρουση με την προηγούμενη συμπεριφορά της βάσης δεδομένων του προμηθευτή, ο προμηθευτής μπορεί να μην επιθυμεί να διακόψει την προς τα πίσω συμβατότητα.
- Οι προμηθευτές έχουν ελάχιστα εμπορικά κίνητρα για να διευκολύνουν την αλλαγή προμηθευτών βάσεων δεδομένων (βλ. κλείδωμα προμηθευτή).
- Οι χρήστες που αξιολογούν λογισμικό βάσεων δεδομένων τείνουν να θέτουν άλλους παράγοντες, όπως η απόδοση, υψηλότερα στις προτεραιότητές τους από τη συμμόρφωση με τα πρότυπα.

### 6.3.2 Τύποι δεδομένων SQL

Το πρότυπο SQL ορίζει τρία είδη τύπων δεδομένων

- Predefined
- Constructed
- User-defined

Οι Constructed τύποι δεδομένων είναι ένας από τους τύπους ARRAY, MULTISSET, REF(erence) ή ROW. Οι user-defined είναι τύποι δεδομένων συγκρίσιμοι με τις κλάσεις στην αντικειμενοστραφή γλώσσα με τους δικούς τους κατασκευαστές, παρατηρητές, μεταλλάκτες, μεθόδους, κληρονομικότητα, υπερφόρτωση, αντικατάσταση, διεπαφές κ.ο.κ. Οι predefined τύποι υποστηρίζονται φυσικά από την ίδια την υλοποίηση.

Predefined τύποι δεδομένων:

1. Τύποι χαρακτήρων

- Character (CHAR)
- Character varying (VARCHAR)
- Character large object (CLOB)

2. National τύποι χαρακτήρων

- National character (NCHAR)
- National character varying (NCHAR VARYING)
- National character large object (NCLOB)

3. Δυαδικοί τύποι

- Binary (BINARY)
- Binary varying (VARBINARY)
- Binary varying object (BLOB)

4. Αριθμητικοί τύποι

- Exact numeric types (NUMERIC, DECIMAL, SMALLINT, INTEGER, BIGINT)
- Approximate numeric types (FLOAT, REAL, DOUBLE PRECISION)
- Decimal floating-point type (DECFLOAT)

5. Τύποι Datetime (DATE, TIME, TIMESTAMP)

6. Τύπος διαστήματος (INTERVAL)

7. Boolean

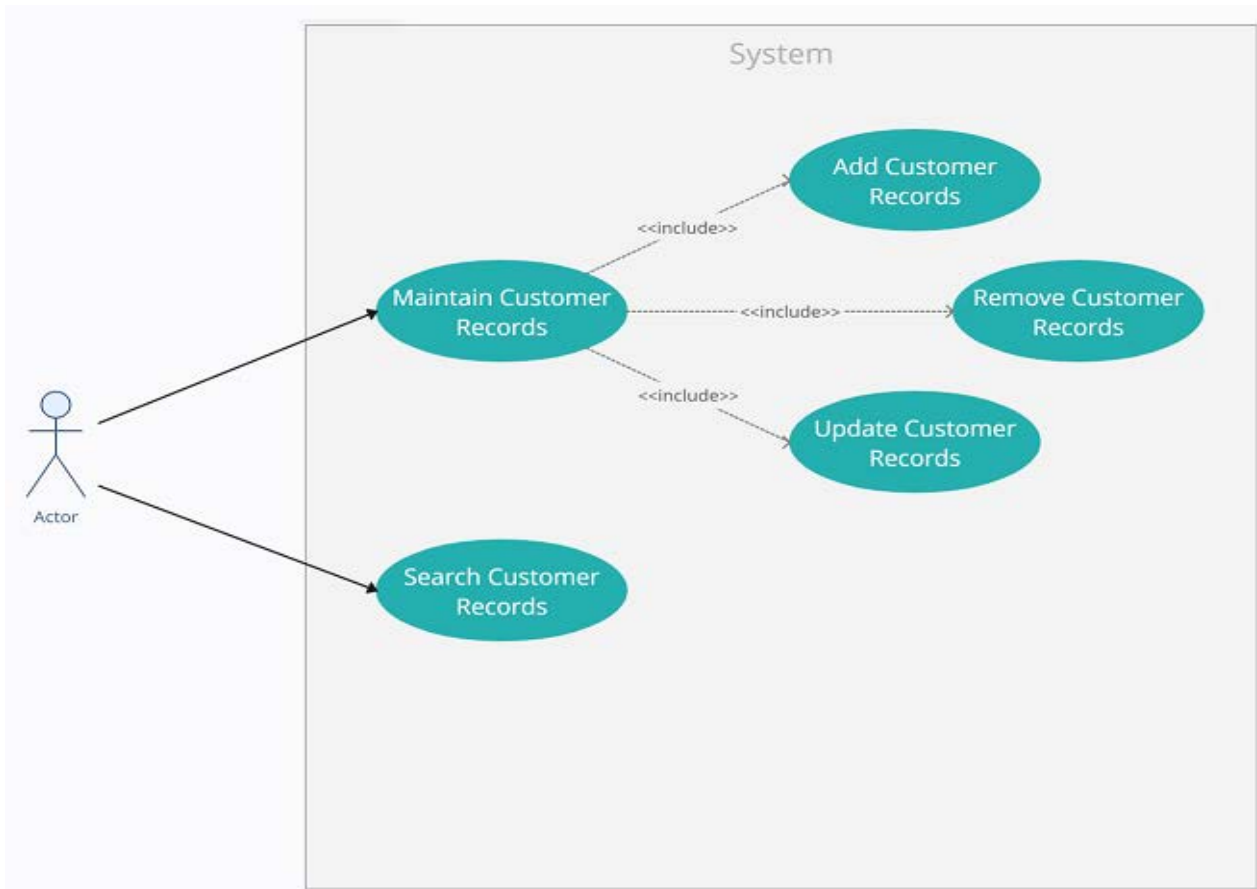
8. XML

9. JSON

## 7. Σχεδίαση της εφαρμογής

### Use Case Diagram

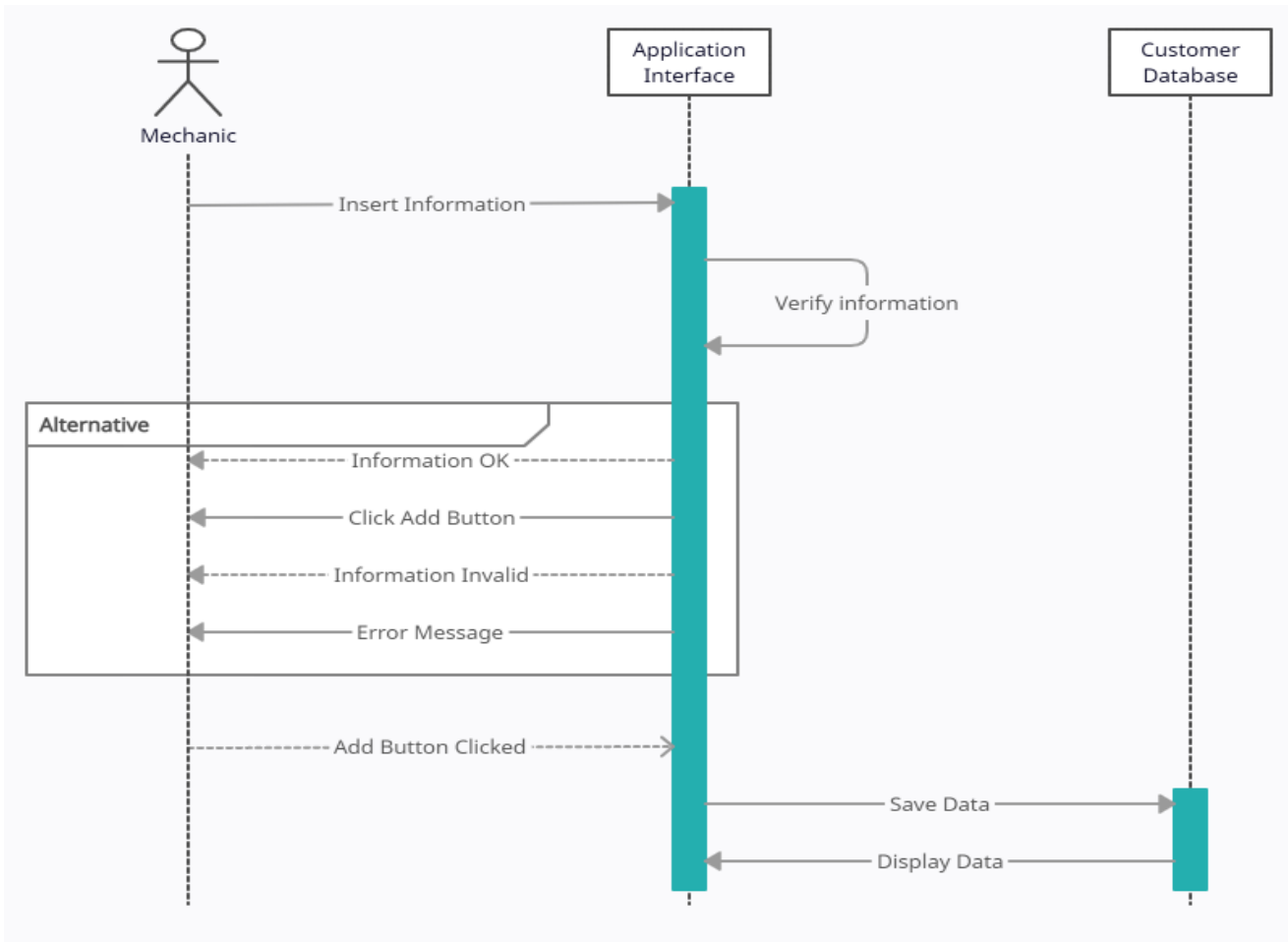
Ένα διάγραμμα περίπτωσης χρήσης είναι μια γραφική απεικόνιση των πιθανών αλληλεπιδράσεων ενός χρήστη με ένα σύστημα. Ένα διάγραμμα περίπτωσης χρήσης δείχνει διάφορες περιπτώσεις χρήσης και διαφορετικούς τύπους χρηστών που έχει το σύστημα και συχνά θα συνοδεύεται και από άλλους τύπους διαγραμμάτων. Οι περιπτώσεις χρήσης αντιπροσωπεύονται είτε με κύκλους είτε με ελλείψεις. Οι ηθοποιοί παρουσιάζονται συχνά ως φιγούρες ραβδίων.



Εικόνα 9. Use Case Diagram

## Sequence Diagram

Ένα διάγραμμα ακολουθίας δείχνει αλληλεπιδράσεις διεργασιών διατεταγμένες σε χρονική ακολουθία στον τομέα της μηχανικής λογισμικού. Απεικονίζει τις διαδικασίες που εμπλέκονται και την αλληλουχία των μηνυμάτων που ανταλλάσσονται μεταξύ των διαδικασιών που απαιτούνται για την εκτέλεση της λειτουργικότητας.



Εικόνα 10. Sequence Diagram

## 8. Ανάπτυξη της εφαρμογής

Ο πηγαίος κώδικας της εφαρμογής περιλαμβάνει την κλάση VehicleMaintenance, η οποία αποτελείται από τα παρακάτω τμήματα κώδικα:

- Το τμήμα δηλώσεων των μεταβλητών
- Την συνάρτηση VehicleMaintenance, στην οποία δημιουργούνται, διαμορφώνονται και εισάγονται όλα τα Controls στην φόρμα
- Τους παρακάτω Event Handlers, οι οποίοι εκτελούν τις ανάλογες ενέργειες και διαδικασίες όταν αυτό απαιτηθεί
  - 1) BtnAddClick
  - 2) BtnRemoveClick
  - 3) BtnEditClick
  - 4) BtnSaveClick
  - 5) DeletePressed
  - 6) TxtCSearchChanged
  - 7) CDataGrid\_DataBindingComplete
  - 8) ClickHandler
  - 9) ColumnHeaderClick
  - 10) Form1\_SizeChanged

### Οργάνωση του κώδικα σε καταλόγους

Ο κώδικας και τα αρχεία είναι οργανωμένα σε καταλόγους ως εξής:

Vehicle Maintenance

Vehicle Maintenance.exe

Resources

Database

VehicleMaintenance.mdf

VehicleMaintenance.ldf

MyIcon.ico

VehicleMaintenance.cs

## 9. Εκτέλεση της εφαρμογής

Παρακάτω φαίνεται η επιτυχής εκτέλεση της εφαρμογής σε διάφορα σενάρια.

Vehicle Maintenance

Customers

Name	Surname	Phone	Email	City	Address	Postal Code	Date of Birth	VIN	License Plates	Make	Model	Year	Trim	Color	Mileage
Achileas	Raptis	6949668244	a.raptis@gmail.com	Ptolemaida	Akrita 34	50200	24/02/1999	1GKDT13S722452177	KZN-5192	Citroen	C3	2006	XTR	Red	315.421
Alexandra	Triantafyllou	6902937978	a.triantafyllou@outlook.com	Thessaloniki	Dimokratias 40	54630	21/07/1967	WWWPD63BX4P032650	NAB-7123	Toyota	Aygo	2016	X	White	67.582
Anastasia	Nikolaou	6909018168	a.nikolaou@outlook.com	Kastoria	Korutsas 19	52100	02/10/1986	1G1YA2DW0B5104528	KTB-7906	Nissan	Qashqai	2019	SV	Black	37.702
Christina	Georgiou	6941733407	c.georgiou@outlook.com	Kozani	Irodotou 29	50010	03/08/2000	3VWD17AJ3EM300485	MNB-6349	Renault	Clio	2011	Extreme	White	209.161
Christos	Papaioannou	6909122849	c.papaioannou@gmail.com	Kozani	Irakleiou 11	50131	23/07/1967	1D3HV13T89J508425	MNA-0360	Nissan	Micra	2011	Tekna	Green	114.730
Eirini	Papadaki	6999123024	e.papadaki@gmail.com	Thessaloniki	Dios 90	54632	20/11/1980	3VWDP7AJ6DM272873	NAB-8123	Toyota	CHR	2017	XLE	Gray	46.712
Eleni	Pappa	6930122101	e.pappa@outlook.com	Kastoria	M. Alexandrou 49	52100	16/07/1978	1GNSKCKCXFR123330	KTZ-1733	Ford	Fiesta	2009	SE	Red	133.021
Georgia	Karagianni	6955128881	g.karagianni@outlook.com	Thessaloniki	Chariton 79	54639	27/02/1972	2HKYF18554H555879	NAA-1237	Hyundai	I20	2020	Magna	Blue	28.249
Giannis	Papadimitriou	6958978645	g.papadimitriou@gmail.com	Kozani	Filota 13	50131	30/12/1981	3GCPCREH8EG241669	MNB-6034	Peugeot	208	2013	GT	Black	142.810
Giorgos	Papadopoulos	6931340760	g.papadopoulos@gmail.com	Althina	Domokou 51	10445	17/05/1994	YV4952CY8B1538199	ATH-4631	Toyota	Yaris	2018	LE	Silver	62.371
Katerina	Oikonomou	6941326451	k.oikonomou@outlook.com	Kozani	Farsalon 45	50132	28/05/1979	JF2SJAAC6FG407567	MNA-0779	Fiat	Panda	2005	CL	White	243.163
Konstantinos	Papageorgiou	6945712329	k.papageorgiou@outlook.com	Ioannina	Ag. Dimitriou 22	26510	27/11/2001	JTJHA31U040011604	INA-9271	Seat	Ibiza	2015	FR Sport	Red	155.926
Konstantinos	Vlachos	6909217100	k.vlachos@gmail.com	Serres	Ilia 55	62124	20/02/1972	1HGCG5651WA013837	EPP-1203	Hyundai	I10	2010	GL	Black	183.602
Maria	Papadopoulou	6937128867	m.papadopoulou@gmail.com	Thessaloniki	25i Martiou 91	54629	14/02/1989	JNKBV61E88M262181	NAA-1911	Toyota	Avensis	2007	T4	Silver	207.493
Marios	Vasileiou	6903173566	m.vasileiou@gmail.com	Ptolemaida	25i Martiou 85	50200	07/02/1995	WDBTJ65J13F098888	KZN-9180	Citroen	C3	2006	VTR	Silver	213.651

Results: 21

Name:

Surname:

Phone:

Email:

City:

Address:

Postal Code:

Date of Birth:

VIN:

License Plates:

Make:

Model:

Year:

Trim:

Color:

Mileage:

Search by:

Add Customer Edit Customer

Remove Customer Save Customer

Εικόνα 11. Αρχικό interface της εφαρμογής

Vehicle Maintenance

Customers

Name	Surname	Phone	Email	City	Address	Postal Code	Date of Birth	VIN	License Plates	Make	Model	Year	Trim	Color	Mileage
Achileas	Raptis	69496682...	a.raptis@gmail.com	Ptolemaida	Akrita 34	50200	24/02/1999	1GKDT13S722452177	KZN-5192	Citroen	C3	2006	XTR	Red	315.421
Alexandra	Triantafyllou	69029379...	a.triantafyllou@outlook.com	Thessaloniki	Dimokratias 40	54630	21/07/1967	WVWPD63BX4P032650	NAB-7123	Toyota	Aygo	2016	X	White	67.582
Anastasia	Nikolaou	69090181...	a.nikolaou@outlook.com	Kastoria	Korutsas 19	52100	02/10/1986	1G1YA2DW0B5104528	KTB-7906	Nissan	Qashqai	2019	SV	Black	37.702
Christina	Georgiou	69417334...	c.georgiou@outlook.com	Kozani	Irodotou 29	50010	03/08/2000	3VVD17AJ3EM300485	MNB-6349	Renault	Clio	2011	Extreme	White	209.161
Christos	Papaioannou	69091228...	c.papaioannou@gmail.com	Kozani	Irakleiou 11	50131	23/07/1967	1D3HV13T89J508425	MNA-0360	Nissan	Micra	2011	Tekna	Green	114.730
Erini	Papadaki	69991230...	e.papadaki@gmail.com	Thessaloniki	Dios 90	54632	20/11/1980	3VWDP7AJ6DM272873	NAB-8123	Toyota	CHR	2017	XLE	Gray	46.712
Eleni	Pappa	69301221...	e.pappa@outlook.com	Kastoria	M. Alexandrou 49	52100	16/07/1978	1GNSKCKCXHR123330	KTZ-1733	Ford	Fiesta	2009	SE	Red	133.021
Georgia	Karagianni	69551288...	g.karagianni@outlook.com	Thessaloniki	Chariton 79	54639	27/02/1972	2HKYF18554H555879	NAA-1237	Hyundai	I20	2020	Magna	Blue	28.249
Giannis	Papadimitriou	69589786...	g.papadimitriou@gmail.com	Kozani	Filota 13	50131	30/12/1981	3GCPREH8EG241669	MNB-6034	Peugeot	208	2013	GT	Black	142.810
Giorgos	Papadopoulos	69313407...	g.papadopoulos@gmail.com	Athina	Domokou 51	10445	17/05/1994	YV4952CY8B1538199	ATH-4631	Toyota	Yaris	2018	LE	Silver	62.371
Katerina	Oikonomou	69413264...	k.oikonomou@outlook.com	Kozani	Farsalon 45	50131	14/02/1989	JNKBV61E88M262181	MNA-0779	Fiat	Panda	2005	CL	White	243.163
Konstantinos	Papageorgiou	69457123...	k.papageorgiou@outlook.com	Ioannina	Ag. Dimitriou 22	46100	14/02/1989	JNKBV61E88M262181	INA-9271	Seat	Ibiza	2015	FR Sport	Red	155.926
Konstantinos	Vlachos	69092171...	k.vlachos@gmail.com	Serres	Ilia 55	50131	14/02/1989	JNKBV61E88M262181	EPP-1203	Hyundai	I10	2010	GL	Black	183.602
Maria	Papadopoulou	69371288...	m.papadopoulou@gmail.com	Thessaloniki	25i Martiou 91	54629	14/02/1989	JNKBV61E88M262181	NAA-1911	Toyota	Avensis	2007	T4	Silver	207.493
Marios	Vasilioiu	69031735...	m.vasilioiu@gmail.com	Ptolemaida	25i Martiou 85	50200	07/02/1995	WDBTJ65J13F098888	KZN-9180	Citroen	C3	2006	VTR	Silver	213.651

Delete Customer

Are you sure you want to remove the selected customer?

Yes No

Name:  City:  VIN:  Year:  Search by:

Surname:  Address:  License Plates:  Trim:

Phone:  Postal Code:  Make:  Color:

Email:  Date of Birth:  Model:  Mileage:

Add Customer Edit Customer Remove Customer Save Customer

Results: 21

Εικόνα 12. Διαγραφή πελάτη

Vehicle Maintenance

Customers

Name	Surname	Phone	Email	City	Address	Postal Code	Date of Birth	VIN	License Plates	Make	Model	Year	Trim	Color	Mileage
Maria	Papadopoulou	6937128867	m.papadopoulou@gmail.com	Thessaloniki	25i Martiou 91	54629	14/02/1989	JNKBV61E88M262181	NAA-1911	Toyota	Avensis	2007	T4	Silver	207.493

Name:  City:  VIN:  Year:  Search by:

Surname:  Address:  License Plates:  Trim:

Phone:  Postal Code:  Make:  Color:

Email:  Date of Birth:  Model:  Mileage:

Add Customer Edit Customer Remove Customer Save Customer

Results: 1

Εικόνα 13. Αναζήτηση πελάτη



## Παράρτημα Κώδικα

Παρακάτω παρουσιάζεται το σημαντικότερο παραρτήμα κώδικα που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής διαχείρισης δεδομένων. Στην Εικόνα 9 φαίνεται η σύνδεση μεταξύ της εφαρμογής διαχείρισης δεδομένων και της βάσης δεδομένων. Το πρώτο και σημαντικότερο πράγμα το οποίο πρέπει να γίνεται σε κάθε αλληλεπίδραση με μια βάση δεδομένων είναι η δημιουργία μιας σύνδεσης. Ένα `SqlConnection` είναι ένα object όπως κάθε άλλο object της C#. Το σύνθημα είναι να δηλώνεται όπως φαίνεται παρακάτω. Στη παράμετρο `Data Source` δίνεται η πηγή δεδομένων, η οποία μπορεί να είναι το όνομα του υπολογιστή, το όνομα του χρήστη ή ακόμα και μια διεύθυνση IP. Ακολουθείτε από την παράμετρο `Initial Catalog` στην οποία συμπληρώνεται το όνομα της βάσης δεδομένων. Τέλος, η παράμετρος `Integrated Security` παίρνει τις παραμέτρους `SSPI` ή `false`. Εάν πάρει την παράμετρο `SSPI` τότε χρησιμοποιούνται τα διαπιστευτήρια του τρέχων Windows λογαριασμού, αλλιώς εάν πάρει την τιμή `false` θα πρέπει να οριστούν οι μεταβλητές `User ID` και `Password` ξεχωριστά. Ο σκοπός της δημιουργίας ενός `SqlConnection` object είναι η χρησιμότητα αυτού ως παράμετρος σε άλλα objects όπως για παράδειγμα του `SqlDataAdapter`. Η ακολουθία των λειτουργιών που λαμβάνουν χώρα κατά τη διάρκεια ζωής ενός `SqlConnection` είναι η εξής:

- Αρχικοποίηση της σύνδεσης `SqlConnection`
- Άνοιγμα της σύνδεσης
- Χρήση της σύνδεσης από άλλα αντικείμενα
- Εκτέλεση λειτουργιών βάσης δεδομένων
- Κλείσιμο της σύνδεσης

```
SqlConnection con= new SqlConnection("Data Source=Xristoforos;Initial Catalog=VehicleMaintenance;Integrated Security=SSPI;");
DataTable datatable = new DataTable();
BindingSource bindingSource = new BindingSource();
SqlDataAdapter adapter = new SqlDataAdapter("Select * From Customers ORDER BY Name ASC;",con);

bindingSource.DataSource = datatable;
adapter.Fill(datatable);
CDataGrid.DataSource = datatable;
```

## Συμπεράσματα

---

Συνοψίζοντας, μια εφαρμογή χρησιμοποιείται για να εκτελέσει κάποιες συγκεκριμένες λειτουργίες για τον τελικό χρήστη. Η εφαρμογή διαχείρισης βάσης δεδομένων αναπτύχθηκε χρησιμοποιώντας την γλώσσα προγραμματισμού C# και την βοήθεια της τεχνολογίας .NET Framework. Αυτό έχει ως αποτέλεσμα, η εφαρμογή να μπορεί να εγκατασταθεί μόνο σε Windows λειτουργικό σύστημα. Η εφαρμογή λειτουργεί εκτός σύνδεσης, δηλαδή δεν χρειάζεται κάποια σύνδεση στο διαδίκτυο για να εκτελέσει τις λειτουργίες για τις οποίες δημιουργήθηκε. Η βάση δεδομένων που χρησιμοποιήθηκε για την διαχείριση των δεδομένων είναι η Microsoft SQL Server η οποία χρησιμοποιεί την γλώσσα προγραμματισμού Structured Query Language (SQL). Η χρησιμότητα της εφαρμογής είναι η διαχείριση δεδομένων. Η διαχείριση δεδομένων είναι ένα κρίσιμο πρώτο βήμα για τη χρησιμοποίηση αποτελεσματικής ανάλυσης δεδομένων σε μια μεγάλη κλίμακα, η οποία οδηγεί σε σημαντικές πληροφορίες που προσθέτουν αξία για παράδειγμα σε ένα επιχειρησιακό περιβάλλον. Με την αποτελεσματική διαχείριση δεδομένων, μια επιχείρηση μπορεί να βρίσκει και να έχει πρόσβαση σε αξιόπιστα δεδομένα ώστε να βρίσκει τις σωστές απαντήσεις στα διάφορα ερωτήματά της. Μερικά πλεονεκτήματα μιας αποτελεσματικής διαχείρισης δεδομένων περιλαμβάνουν την αξιοπιστία, την ασφάλεια και την επεκτασιμότητα.

## Προτάσεις μελλοντικής επέκτασης

Πολλές διαφορετικές προσαρμογές, δοκιμές και πειράματα έχουν μείνει για το μέλλον ως προτάσεις μελλοντικής επέκτασης, οι οποίες έχουν ως στόχο τη βαθύτερη ανάλυση συγκεκριμένων μηχανισμών, νέες προτάσεις για δοκιμή διαφορετικών μεθόδων ή απλά την ικανοποίηση περιέργειας. Μια πρόταση μελλοντικής επέκτασης της εφαρμογής θα ήταν, η δημιουργία μιας ειδικά διαμορφωμένης διεπαφής, μέσω της οποίας θα μπορούσε ο τελικός χρήστης εισάγοντας τα κατάλληλα στοιχεία, να συνδεθεί απομακρυσμένα σε μία βάση δεδομένων αντί να συνδέεται στην τοπική βάση του υπολογιστή του. Αυτό βέβαια προϋποθέτει πως ο χρήστης θα πρέπει να γνωρίζει κάποιες συγκεκριμένες πληροφορίες για την βάση στην οποία θέλει να συνδεθεί. Ωστόσο για να επιτευχθεί η υλοποίηση αυτή θα πρέπει πρώτα να γίνουν οι κατάλληλες ρυθμίσεις και από τις δύο πλευρές. Επιπλέον, πρέπει να ληφθεί υπόψιν ότι η πρόσβαση στο διαδίκτυο είναι απαραίτητη σε αυτή την περίπτωση.

## Βιβλιογραφία

---

- [1] R. H. Arpaci-Dusseau και A. C. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, CreateSpace Independent Publishing Platform, 2018.
- [2] D. Dhamdhere, Operating Systems, McGraw-Hill, Inc., 2008.
- [3] H. M. Deitel, An introduction to operating systems (2nd ed.), Addison-Wesley Longman Publishing Co., Inc., 1990.
- [4] K. W. Tracy, Software: A Technical History, Association for Computing Machinery, 2021.
- [5] A. Silberschatz, P. B. Galvin και G. Gagne, Operating System Concepts, 10e Abridged Print Companion, John Wiley & Sons, 2018.
- [6] T. Thai και H. Lam, .NET Framework Essentials, O'Reilly Media, Inc., 2003.
- [7] J. Richter, Applied Microsoft .NET Framework, Microsoft Press, 2002.
- [8] M. Lutz και P. Laplante, C# and the .NET Framework: Ready for real time, IEEE, 2003.
- [9] B. Albahari, P. Drayton και B. Merrill, C# Essentials: Programming the .NET Framework, O'Reilly Media, Inc., 2002.
- [10] P. Ritchie, Practical Microsoft Visual Studio 2015, Apress, 2016.
- [11] J. Sharp, Microsoft Visual C# Step by Step, Microsoft Press, 2018.
- [12] R. Mistry και S. Misner, Introducing Microsoft SQL Server 2014, Microsoft Press, 2014.
- [13] D. Petković, Microsoft SQL Server 2005: A Beginner's Guide, McGraw Hill, 2005.
- [14] K. Delaney, Inside Microsoft SQL Server 2000 3rd Edition, Microsoft Press, 2000.
- [15] A. Hejlsberg, M. Torgersen, S. Wiltamuth και P. Golde, The C# Programming Language, Pearson Education, 2008.
- [16] J. Albahari, C# 9.0 in a Nutshell, O'Reilly Media, Inc., 2021.
- [17] H.-P. Halvorsen, Structured Query Language, University College of Southeast Norway, 2016.