ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εξόρυξη online θέσεων εργασίας από τον παγκόσμιο ιστό και ανάλυσή τους.

Θεοφάνης Παπαδόπουλος

Επιβλέπων καθηγητής: Τζήμας Γιάννης

Πάτρα – Φεβρουάριος 2020

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Περιεχόμενα

Λίστα Εικόνων

Πρόλογος

Στην διπλωματική εργασία αυτή πραγματευόμαστε την συλλογή δεδομένων από τον παγκόσμιο ιστό. Αναλυτικά ως στόχο η εργασία είχε την συλλογή και ανάλυση θέσεων εργασίας για το έτος 2021. Έχοντας ως γνώμονα το γεγονός πως για να γίνει μια σωστή και επαρκής ανάλυση θέσεων εργασίας χρειάζεται το λιγότερο ένας χρόνος με σκοπό τον διαχωρισμό των εργασιών σε εποχές κλπ. όμως λόγο χρονικών περιορισμών(6 μήνες) κατέστη αδύνατο αυτό καθώς χρειάστηκε στο χρονικό περιθώριο των 6 μηνών να το χωρίσουμε διότι έπρεπε να δημιουργήσουμε τους μηχανισμούς και ότι άλλο χρειαζόμαστε για να γίνει η ανάλυση και η συλλογή των δεδομένων. Αναλυτικά ξεκινήσαμε από την διαδικασία της αξιολόγησης των ιστοσελίδων δηλαδή χρησιμοποιώντας λέξεις κλειδιά στην μηχανή αναζήτησης (google) όπως (‘εύρεσή εργασίας’, ‘εύρεση δουλεία’, ‘θέσεις εργασίας’) στην ελέγξαμε την σχετικότητα των ιστοσελίδων με το θέμα μας και τα αξιολογήσαμε στους μηχανισμούς Alexa, AHREF Authority και MOZ Authority. Στην συνέχεια εφόσον καταλήξαμε στην ιστοσελίδα με την οποία και ασχοληθήκαμε δημιουργήσαμε το πρώτο πρόγραμμα το οποίο κρατάει όλες τις αγγελίες εργασίας που παίρνουμε από την ιστοσελίδα, έπειτα φτιάξαμε μία βάση δεδομένων στην οποία αποθηκεύουμε τον τίτλος της κάθε αγγελίας, το HTML αρχείο και την ημερομηνία που έγινε η εξαγωγή. Επιπροσθέτως επειδή χρειαζόταν να ασχοληθούμε με την ‘κρυμμένη’ πληροφορία που υπάρχει στο HTML αρχείο χρειάστηκε να κατασκευάσουμε και ένα regular expression με το οποίο «τραβάμε» το HTML αρχείο από την βάση δεδομένων στην οποία το αποθηκεύσαμε και το επεξεργαζόμαστε με σκοπό να εξάγουμε τις πληροφορίες που «κρύβει» μέσα του, τις οποίες χρειαστήκαμε ώστε να γίνει η ανάλυση των δεδομένων. Τέλος αφού δημιουργήσαμε όλα τα παραπάνω κατασκευάσαμε άλλη μία βάση δεδομένων για να «κρατάμε» τον τίτλο της επιχείρησης, την ημερομηνία δημοσίευσης της αγγελίας, την περιοχή οπού είναι η επιχείρηση κτλ. και όλα αυτά με την βοήθεια του Power Bi τα αναλύσαμε με την χρήση των διαγραμμάτων που μας προσφέρει όπως θα δείτε και στο κεφάλαιο 3.

Αρχικά στο 1ο κεφάλαιο επεξηγούμε τις τεχνολογίες που χρειάστηκε να χρησιμοποιήσουμε ώστε να φτάσουμε στον τελικό μας στόχο, πιο συγκεκριμένα εξηγούμε ποιες είναι αυτές, που μας χρειάστηκαν καθώς και την χρησιμότητα τους σε γενικότερο επίπεδο τέλος αναφέρουμε την ιστορία τους δηλαδή ποιος τις εφέυρε, πότε και ποιες εταιρίες τις «καλύπτουν» αυτή την στιγμή.

Αναλυτικά στο 2ο κεφάλαιο γίνεται θεωρητική ανάλυση των διαφόρων τεχνολογιών που απαρτίζουν το συνολικό σύστημα για την εξαγωγή των αγγελιών από την ιστοσελίδα και πώς καταλήξαμε σε αυτήν. Αρχικά, μελετάται η ανάλυση της ιστοσελίδας και πως ακριβώς καταλήξαμε στην συγκεκριμένη, στην συνέχεια αναπτύσσονται οι τεχνολογίες και οι τεχνικές που χρησιμοποιήθηκαν για να συλλέξουμε τα δεδομένα από την ιστοσελίδα και ο τρόπος με

τον οποίο αποθηκεύονται στην βάση δεδομένων μας, πιο συγκεκριμένα εξηγούμε επακριβώς την διαδικασία που ακολουθήσαμε για να υλοποιήσουμε τον web crawler καθώς εξηγείται αναλυτικά ο τρόπος λειτουργίας της κάθε συνάρτησης που χρησιμοποιήσαμε.

Στο 3ο κεφάλαιο γίνεται θεωρητική ανάλυση για τον προγραμματισμό του regular expression(regex). Πιο αναλυτικά εξηγούμε την διαδικασία που ακολουθήσαμε για να υλοποιήσουμε το regex και πως αυτό δουλεύει με στόχο το ξεσκαρτάρισμά της πληροφορίας και στο τέλος να μας μείνει μόνο ότι χρειαζόμαστε να αναλύσουμε στο Power Bi. Τέλος εξηγείται και ο τρόπος με τον οποίο αποθηκεύουμε την πληροφορία μας από τον ένα πίνακα(columns) στον οποίο αποθηκεύουμε ότι μας εξάγει ο ανιχνευτής στον άλλο πίνακα(tester) ο οποίος είναι ο πίνακας με τον οποίο θα ασχοληθούμε και στο 3ο κεφάλαιο διότι από αυτόν τον πίνακα παίρναμε την πληροφορία μας στο Power Bi ώστε να την αναλύσουμε.

Στο 4ο κεφάλαιο στο οποίο γίνεται η ανάλυση όσων αποθηκεύσαμε μέσο του power bi, αρχικά «περάσαμε» όλη την πληροφορία που είχαμε αποθηκεύσει στην βάση δεδομένων την οποία είχαμε σύνδεση με το regular expression, στην συνέχεια με την δημιουργία πινάκων, διαγραμμάτων, καρτών, χάρτη καθώς και την χρήση φίλτρων αναλύσαμε όσα είχαμε αποθηκεύσει σε δύο σελίδες, στην πρώτη σελίδα η οποία ονομάζεται General παρουσιάζουμε μια γενική εικόνα για τις πληροφορίες που έχουμε, πιο αναλυτικά παρουσιάζουμε τις συνολικές θέσης εργασίας, τις συνολικές ημέρες που κάναμε crawl, τις συνολικές εταιρίες που ανιχνεύσαμε με τον crawler, πόσες αγγελίες έχουμε τις τελευταίες 10 ημέρες και πόσες τις τελευταίες 30 όλα αυτά τα παρουσιάζουμε με την χρήση των καρτών, έπειτα με την δημιουργία πινάκων, διαγραμμάτων και του χάρτη βλέπουμε τους μήνες καθώς και αναλυτικά τις πληροφορίες για την κάθε εταιρεία, τον τίτλο της αγγελίας και την ακριβή ημερομηνία δημοσίευσή της αγγελίας. Στην συνέχεια στην δεύτερη και τελευταία σελίδα η οποία ονομάζεται Trending παρουσιάζουμε τις top εμφανίσεις, δηλαδή παρουσιάζουμε τις 10 εταιρείες με τις περισσότερες δημοσιευμένες αγγελίες, του 10 τίτλους εργασίας με την μεγαλύτερη απήχηση, τις 10 καλύτερες περιοχές με την μεγαλύτερη ζήτηση ατόμων για εργασία, τους 3 μήνες με την περισσότερη ζήτηση ατόμων για να εργαστούν και τέλος τον τύπο εργασίας που ζητήθηκε πιο πολύ από τους εργοδότες.

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάλυση θέσεων εργασίας στην Ελλάδα για το έτος 2021, η διαδικασία με την οποία επιτυγχάνουμε τον σκοπό μας είναι η εξής. Αρχικά αναζητήσαμε ιστοσελίδες(websites) οι οποίες είναι σχετικές με το θέμα μας, τις αναλύσαμε ώστε να ελέγξουμε ποια μας καλύπτει με βάση κάποιους κανόνες και κάποιους μηχανισμούς, επιλέξαμε μία από όλες με την οποία και εργαστήκαμε. Στην συνέχεια δημιουργήσαμε μία βάση δεδομένων με τρείς πίνακες, στον πρώτο πίνακα κρατήσαμε όσα μας επιστρέφει ο web crawler, τον δεύτερο πίνακα τον χρειαστήκαμε για να λειτουργήσει ως counter δηλαδή αποθηκεύουμε την τιμή που μας επιστρέφει ο regex κάθε φόρα που ανιχνεύει id στον πίνακα column μεγαλύτερο της τιμής value που υπάρχει στον πίνακα tester, ο λόγος που χρειάστηκε ο δεύτερος πίνακας είναι για να μας βοηθήσει να δημιουργήσουμε μία συνάρτηση που θα λειτουργεί ως block deduplication για το regular expression, στον τρίτο και τελευταίο πίνακα αποθηκεύουμε ότι μας επιστρέφει το regular expression. Έπειτα προγραμματίσαμε τον web crawler με τον οποίο «τραβήξαμε» από την κάθε μία θέση εργασίας ξεχωριστά το URL και τον HTML κώδικα της. Το URL το χρειαστήκαμε ως μέθοδο block duplicate δηλαδή με βάση το URL καταφέραμε η κάθε θέση εργασίας να αποθηκεύεται μόνο μία φορά στην βάση δεδομένων μας, το HTML αρχείο το χρειαστήκαμε στην συνέχεια, οπού μέσα σε αυτό είναι «κρυμμένη» όλη η πληροφορία που θέλαμε να αναλύσουμε με το regex (π.χ. Τίτλος εργασίας, Το όνομα της εταιρίας, Περιοχή κτλ.). Επίσης δημιουργήσαμε το regular expression, το regular expression μας βοηθά με τύπους βασισμένους σε ακολουθία λογικών χαρακτήρων να εντοπίσουμε τίς πληροφορίες που χρειαζόμαστε για το τρίτο μέρος της διπλωματικής εργασίας στο οποίο γίνεται η ανάλυση των δεδομένων που παίρνουμε από το regular expression με την χρήση του power bi. Τέλος όσες αγγελίες πήραμε από το regular expression τα μεταφέραμε στο power bi με σκοπό να τα αναλύσουμε με την χρήση διαγραμμάτων και πινάκων, δηλαδή με την χρήση του power bi κατορθώσαμε να βγάλουμε μια γενική εικόνα σχετικά με τις θέσεις εργασίας που δημοσιεύθηκαν καθώς και τις τάσεις.

Abstract

The present thesis discusses the analysis of job position in Greece in the year of 2021. The procedure with which our goal was accomplished is the following. First of all we searched for websites relating to our subject, we analyzed them in order to examine which one Is ideal base on certain rules and mechanisms and we opted for one that we continued to work with. After that, we created a data base with three boards. We used the first board for what the web crawler returns. The second one was required to act as we save the value returned by regex each time it detects an id in the column table greater than value that exists in the tester table, the reason the second one was need is to help us create a function that will act as a block deduplication for the regular expression. In the third and last board we store whatever the regular expression returns. Then, we programmed the web crawler which we used to extract the URL and HTML code separately from each job position. The URL was used as a block duplicate method. Specifically, using the URL, we managed the capture each position once in our data base. The HTML file was required later on, since it contained all the information needed for analyzing the regex(e.g. thesis title, company's name, area, etc.) We also created the regular expression, the normal expression we help with formulas based on a sequence of logical characters to find the information we need for the third part of the dissertation in which the data we get from the normal expression is analyzed using power bi. Finally, we transferred the ads that received the normal expression to power bi in order to analyze them using charts and tables, that is, with the use of power we managed to get an overview of the jobs that were published as well as the trends.

1 Τεχνολογίες

Για την διπλωματική εργασία αρχικά χρησιμοποιήσαμε την MySQL στην συνέχεια την Python και τέλος ασχοληθήκαμε με την τεχνολογία RegEx καθώς και με XAMPP, η κάθε μία τεχνολογία ξεχωριστά ήταν χρήσιμη για διαφορετικό σκοπό, δηλαδή με την MySQL δημιουργήσαμε την βάση δεδομένων μας στην οποία και αποθηκεύσαμε ότι πήραμε από τον crawler, με την Python προγραμματίσαμε τον crawler ώστε να εντοπίζει την ιστοσελίδα στο διαδίκτυο ενώ στην συνέχεια «έπαιρνε» όλες τις αγγελίες που υπήρχαν στην ιστοσελίδα και τέλος της αποθήκευε στην βάση δεδομένων μας, επιπρόσθετος με την Python δημιουργήσαμε και το RegEx με το οποίο ξεχωρίσαμε τις πληροφορίες που μας «έδωσε» ο crawler σε χρήσιμες και μη χρήσιμες με σκοπό να τις αναλύσουμε τέλος με την τεχνολογία XAMPP δημιουργήσαμε έναν τοπικό Server στον υπολογιστή μας ώστε να αποθηκεύονται online οι πληροφορίες στην βάση δεδομένων.

1.1 Python

Η Python είναι διερμηνευόμενη (interpreted), γενικού σκοπού (general-purpose) και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (*Imperative programming*) και υποστηρίζει τόσο το διαδικαστικό (*procedural programming*) όσο και το αντικειμενοστραφές (*object-oriented programming*) προγραμματιστικό υπόδειγμα (*programming paradigm*). Είναι δυναμική γλώσσα προγραμματισμού (dynamically typed) και υποστηρίζει συλλογή απορριμμάτων(*garbage collection ή GC*).

Δημιουργήθηκε από τον Ολλανδό Γκιντο βαν Ροσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989[1] και κυκλοφόρησε για πρώτη φορά το 1991.

Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησης της. Μειονεκτεί στο ότι επειδή είναι διερμηνευόμενη είναι πιο αργή από τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++. Για αυτόν τον λόγο δεν είναι κατάλληλη για γραφή λειτουργικών συστημάτων.

Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Χρησιμοποιώντας εργαλεία τρίτων, όπως το Py2exe ή το Pyinstaller, ο κώδικας της Python μπορεί να πακεταριστεί σε αυτόνομα εκτελέσιμα προγράμματα για μερικά από τα πιο δημοφιλή λειτουργικά συστήματα, επιτρέποντας τη διανομή του βασισμένου σε Python λογισμικού για χρήση σε αυτά τα περιβάλλοντα χωρίς να απαιτείται εγκατάσταση του διερμηνευτή της Python.

Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο.

Η Python είναι μια γλώσσα προγραμματισμού πολλών παραδειγμάτων. Ο αντικειμενοστραφής προγραμματισμός και ο δομημένος προγραμματισμός υποστηρίζονται πλήρως και πολλά από τα χαρακτηριστικά του υποστηρίζουν λειτουργικό προγραμματισμό και προγραμματισμό με όψη (συμπεριλαμβανομένου του μεταπρογραμματισμού και των μετα-αντικειμένων (μαγικές μέθοδοι)). Πολλά άλλα παραδείγματα υποστηρίζονται μέσω επεκτάσεων, συμπεριλαμβανομένου του σχεδιασμού με σύμβαση και του λογικού προγραμματισμού.

Η Python χρησιμοποιεί δυναμική πληκτρολόγηση και συνδυασμό καταμέτρησης αναφοράς και συλλέκτη σκουπιδιών ανίχνευσης κύκλου για διαχείριση μνήμης. Διαθέτει επίσης δυναμική ανάλυση ονόματος (καθυστερημένη δέσμευση), η οποία δεσμεύει ονόματα μεθόδων και μεταβλητών κατά την εκτέλεση του προγράμματος.

Ο σχεδιασμός της Python προσφέρει κάποια υποστήριξη για λειτουργικό προγραμματισμό στην παράδοση Lisp. Διαθέτει λειτουργίες φίλτρου, μείωσης και μείωσης. απαριθμήστε κατανοήσεις, λεξικά, σύνολα και εκφράσεις γεννήτρια. Η τυπική βιβλιοθήκη διαθέτει δύο ενότητες (itertools και functools) που υλοποιούν λειτουργικά εργαλεία που δανείστηκαν από το Haskell και το Standard ML.

Η βασική φιλοσοφία της γλώσσας συνοψίζεται στο έγγραφο The Zen of Python (PEP 20), το οποίο περιλαμβάνει αφορισμούς όπως:

Το όμορφο είναι καλύτερο από το άσχημο.

Το ρητό είναι καλύτερο από το σιωπηρό.

Το απλό είναι καλύτερο από το σύνθετο.

Το σύνθετο είναι καλύτερο από το περίπλοκο.

Η αναγνωσιμότητα μετράει.

Αντί να έχει ενσωματωμένη όλη τη λειτουργικότητά του στον πυρήνα του, η Python σχεδιάστηκε για να είναι εξαιρετικά επεκτάσιμη (με μονάδες). Αυτή η συμπαγής αρθρωτότητα το έχει κάνει ιδιαίτερα δημοφιλές ως μέσο προσθήκης προγραμματιζόμενων διεπαφών σε υπάρχουσες εφαρμογές. Το όραμα του Van Rossum για μια μικρή βασική γλώσσα με μεγάλη τυπική βιβλιοθήκη και εύκολα επεκτάσιμο διερμηνέα προήλθε από τις απογοητεύσεις του με το ABC, που υποστήριζε την αντίθετη προσέγγιση.

Η Python προσπαθεί για μια απλούστερη, λιγότερο ακατάστατη σύνταξη και γραμματική, ενώ δίνει στους προγραμματιστές μια επιλογή στη μεθοδολογία κωδικοποίησης. Σε αντίθεση με το σύνθημα του Perl "υπάρχουν περισσότεροι από ένας τρόποι για να το κάνουμε", η Python υιοθετεί μια φιλοσοφία σχεδιασμού "θα πρέπει να υπάρχει ένας - και κατά προτίμηση μόνο ένας" προφανής τρόπος να το κάνουμε ". Ο Alex Martelli, συνεργάτης στο Python Software Foundation και συγγραφέας βιβλίων Python, γράφει ότι "Το να περιγράφεις κάτι ως" έξυπνο "δεν θεωρείται φιλοφρόνηση στην κουλτούρα της Python."

Οι προγραμματιστές της Python προσπαθούν να αποφύγουν την πρόωρη βελτιστοποίηση και απορρίπτουν τις ενημερώσεις κώδικα σε μη κρίσιμα μέρη της εφαρμογής αναφοράς CPython που θα προσέφεραν οριακές αυξήσεις στην ταχύτητα με κόστος σαφήνειας. Όταν η ταχύτητα είναι σημαντική, ένας προγραμματιστής Python μπορεί να μετακινήσει κρίσιμες για το χρόνο λειτουργίες σε μονάδες επέκτασης γραμμένες σε γλώσσες όπως η C, ή να χρησιμοποιήσει PyPy, έναν μεταγλωττιστή απλώς σε χρόνο. Διατίθεται επίσης Cython, το οποίο μεταφράζει ένα σενάριο Python σε C και πραγματοποιεί απευθείας κλήσεις API σε επίπεδο C στον διερμηνέα Python.

Οι προγραμματιστές της Python στοχεύουν να διατηρήσουν τη γλώσσα διασκεδαστική στη χρήση. Αυτό αντικατοπτρίζεται στο όνομά του - ένα αφιέρωμα στη βρετανική κωμική ομάδα Monty Pythonκαι σε περιστασιακά παιχνιδιάρικες προσεγγίσεις σε σεμινάρια και υλικά αναφοράς, όπως παραδείγματα που αναφέρονται σε ανεπιθύμητα μηνύματα και αυγά (αναφορά σε σκίτσο των Monty Python) του τυπικού foo και bar.

Ένας κοινός νεολογισμός στην κοινότητα Python είναι ο pythonic, ο οποίος μπορεί να έχει ένα ευρύ φάσμα σημασιών που σχετίζονται με το στυλ του προγράμματος. Το να πούμε ότι ο κώδικας είναι πυθωνικός σημαίνει ότι χρησιμοποιεί καλά ιδιώματα Python, ότι είναι φυσικό ή δείχνει ευχέρεια στη γλώσσα, ότι συμμορφώνεται με τη μινιμαλιστική φιλοσοφία της Python και την έμφαση στην αναγνωσιμότητα. Αντίθετα, ο κώδικας που είναι δύσκολο να κατανοηθεί ή διαβάζεται σαν μια πρόχειρη μεταγραφή από άλλη γλώσσα προγραμματισμού ονομάζεται μη πυθωνικός.

Οι χρήστες και οι θαυμαστές της Python, ειδικά εκείνοι που θεωρούνται γνώστες ή έμπειροι, συχνά αναφέρονται ως Pythonistas.

Η Python χρησιμοποιείται συνήθως για την ανάπτυξη ιστότοπων και λογισμικού, αυτοματοποίηση εργασιών, ανάλυση δεδομένων και οπτικοποίηση δεδομένων. Δεδομένου ότι είναι σχετικά εύκολο να μάθει, η Python υιοθετήθηκε από πολλούς μη προγραμματιστές όπως λογιστές και επιστήμονες, για μια ποικιλία καθημερινών εργασιών, όπως η οργάνωση οικονομικών

Από το 2003, η Python κατατάσσεται σταθερά στις δέκα πιο δημοφιλείς γλώσσες προγραμματισμού στον Δείκτη Κοινότητας Προγραμματισμού TIOBE όπου, από τον Φεβρουάριο του 2021, είναι η τρίτη πιο δημοφιλής γλώσσα (πίσω από την Java και την C). Επιλέχθηκε Γλώσσα Προγραμματισμού της Χρονιάς (για «τη μεγαλύτερη άνοδο στις αξιολογήσεις σε ένα έτος») το 2007, το 2010, το 2018 και το 2020 (η μόνη γλώσσα που το έκανε τέσσερις φορές).

Μια εμπειρική μελέτη διαπίστωσε ότι οι γλώσσες δέσμης ενεργειών, όπως η Python, είναι πιο παραγωγικές από τις συμβατικές γλώσσες, όπως η C και η Java, για προβλήματα προγραμματισμού που περιλαμβάνουν χειρισμό συμβολοσειρών και αναζήτηση σε ένα λεξικό και διαπίστωσε ότι η κατανάλωση μνήμης ήταν συχνά "καλύτερη από την Java και όχι πολύ χειρότερο από το C ή το C ++ ».

Οι μεγάλοι οργανισμοί που χρησιμοποιούν Python περιλαμβάνουν Wikipedia, Google, Yahoo!, CERN NASA, Facebook, Amazon, Instagram, Spotify και μερικές μικρότερες οντότητες όπως ILM και ITA. Ο ιστότοπος κοινωνικής δικτύωσης Reddit γράφτηκε κυρίως σε Python.

Η Python μπορεί να χρησιμεύσει ως γλώσσα δέσμης ενεργειών για εφαρμογές ιστού, π.χ. μέσω mod_wsgi για τον διακομιστή Ιστού Apache. Με το Web Server Gateway Interface, έχει δημιουργηθεί ένα τυπικό API για τη διευκόλυνση αυτών των εφαρμογών. Πλαίσια Ιστού όπως Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle και Zope υποστηρίζουν προγραμματιστές στο σχεδιασμό και τη συντήρηση πολύπλοκων εφαρμογών. Το Pyjs και το IronPython μπορούν να χρησιμοποιηθούν για την ανάπτυξη του πελάτη των εφαρμογών που βασίζονται στον Ajax. Το SQLAlchemy μπορεί να χρησιμοποιηθεί ως αντιστοιχιστής δεδομένων σε σχεσιακή βάση δεδομένων. Το Twisted είναι ένα πλαίσιο για τον προγραμματισμό επικοινωνιών μεταξύ υπολογιστών και χρησιμοποιείται (για παράδειγμα) από το Dropbox.

Βιβλιοθήκες όπως οι NumPy, SciPy και Matplotlib επιτρέπουν την αποτελεσματική χρήση της Python στον επιστημονικό υπολογισμό, με εξειδικευμένες βιβλιοθήκες όπως το Biopython και το Astropy που παρέχουν λειτουργικότητα συγκεκριμένου τομέα. Το SageMath είναι ένα σύστημα άλγεβρας υπολογιστών με μια διεπαφή φορητού υπολογιστή που μπορεί να προγραμματιστεί σε Python: η βιβλιοθήκη του καλύπτει πολλές πτυχές των μαθηματικών,

συμπεριλαμβανομένης της άλγεβρας, των συνδυαστικών, των αριθμητικών μαθηματικών, της θεωρίας αριθμών και του λογισμού. Το OpenCV διαθέτει συνδέσεις Python με ένα πλούσιο σύνολο δυνατοτήτων για όραση υπολογιστή και επεξεργασία εικόνας.

Η Python χρησιμοποιείται συνήθως σε έργα τεχνητής νοημοσύνης και έργα μηχανικής μάθησης με τη βοήθεια βιβλιοθηκών όπως TensorFlow, Keras, Pytorch και Scikit-learn. Ως γλώσσα δέσμης ενεργειών με αρθρωτή αρχιτεκτονική, απλή σύνταξη και πλούσια εργαλεία επεξεργασίας κειμένου, η Python χρησιμοποιείται συχνά για επεξεργασία φυσικής γλώσσας.

Η Python έχει ενσωματωθεί επιτυχώς σε πολλά προϊόντα λογισμικού ως γλώσσα δέσμης ενεργειών, συμπεριλαμβανομένου του λογισμικού με μέθοδο πεπερασμένων στοιχείων, όπως Abaqus, 3D παραμετρικού μοντελιστή όπως το FreeCAD, πακέτα τρισδιάστατων κινούμενων σχεδίων όπως 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo , MotionBuilder, Softimage, ο συνθέτης οπτικών εφέ Nuke, προγράμματα απεικόνισης 2D όπως το GIMP, Inkscape, Scribus and Paint Shop Pro, και προγράμματα μουσικής σημειογραφίας όπως σεναριογράφος και capella. Το GNU Debugger χρησιμοποιεί την Python ως έναν όμορφο εκτυπωτή για να εμφανίσει πολύπλοκες δομές όπως τα κοντέινερ C ++. Η Esri προωθεί την Python ως την καλύτερη επιλογή για τη συγγραφή σεναρίων στο ArcGIS. Έχει επίσης χρησιμοποιηθεί σε πολλά βιντεοπαιχνίδια, και έχει υιοθετηθεί ως η πρώτη από τις τρεις διαθέσιμες γλώσσες προγραμματισμού στο Google App Engine, οι άλλες δύο είναι η Java και η Go.

Πολλά λειτουργικά συστήματα περιλαμβάνουν την Python ως τυπικό στοιχείο. Αποστέλλεται με τις περισσότερες διανομές Linux, AmigaOS 4 (χρησιμοποιώντας Python 2.7), FreeBSD (ως πακέτο), NetBSD, OpenBSD (ως πακέτο) και macOS και μπορεί να χρησιμοποιηθεί από τη γραμμή εντολών (τερματικό). Πολλές διανομές Linux χρησιμοποιούν προγράμματα εγκατάστασης γραμμένα σε Python: Το Ubuntu χρησιμοποιεί το πρόγραμμα εγκατάστασης Ubiquity, ενώ το Red Hat Linux και το Fedora χρησιμοποιούν το πρόγραμμα εγκατάστασης Anaconda. Το Gentoo Linux χρησιμοποιεί την Python στο σύστημα διαχείρισης πακέτων, Portage.

Η Python χρησιμοποιείται εκτενώς στη βιομηχανία ασφάλειας πληροφοριών, συμπεριλαμβανομένης της ανάπτυξης εκμετάλλευσης.

Το μεγαλύτερο μέρος του λογισμικού Sugar για το One Laptop per Child XO, που αναπτύχθηκε τώρα στο Sugar Labs, είναι γραμμένο σε Python. Το έργο υπολογιστή Raspberry Pi με έναν πίνακα υιοθέτησε την Python ως την κύρια γλώσσα προγραμματισμού χρήστη.

Το LibreOffice περιλαμβάνει την Python και σκοπεύει να αντικαταστήσει την Java με την Python. Ο παροχέας Python Scripting είναι ένα βασικό χαρακτηριστικό από την έκδοση 4.0 από τις 7 Φεβρουαρίου 2013.

1.1.1 Η ιστορία της Python

Η Python σχεδιάστηκε στα τέλη της δεκαετίας του 1980] από τον Guido van Rossum στο Centrum Wiskunde & Informatica (CWI) στις Κάτω Χώρες ως διάδοχος της γλώσσας προγραμματισμού ABC, η οποία ήταν εμπνευσμένη από το SETL,] ικανό να χειρίζεται και να αλληλεπιδρά με εξαίρεση Λειτουργικό σύστημα Amoeba. Η εφαρμογή του ξεκίνησε τον Δεκέμβριο του 1989. Ο Van Rossum ανέλαβε την αποκλειστική ευθύνη για το έργο, ως τον κύριο προγραμματιστή, μέχρι τις 12 Ιουλίου 2018, όταν ανακοίνωσε τις «μόνιμες διακοπές» του από τις ευθύνες του ως Benevolent Dictator For Life της Python, τίτλος που του χάρισε η κοινότητα Python μακροπρόθεσμη δέσμευση ως ο κύριος υπεύθυνος λήψης αποφάσεων του έργου. Τον Ιανουάριο του 2019, οι ενεργοί προγραμματιστές της Python εξέλεξαν ένα 5μελές "Steering Council" για να ηγηθεί του έργου. Από το 2021, τα σημερινά μέλη αυτού του συμβουλίου είναι οι Barry Warsaw, Brett Cannon, Carol Willing, Thomas Wouters και Pablo Galindo Salgado.

Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου 2000, με πολλές σημαντικές νέες δυνατότητες, συμπεριλαμβανομένου ενός συλλέκτη σκουπιδιών που ανιχνεύει τον κύκλο και υποστήριξη για το Unicode.

Η Python 3.0 κυκλοφόρησε στις 3 Δεκεμβρίου 2008. wasταν μια σημαντική αναθεώρηση της γλώσσας που δεν είναι εντελώς συμβατή προς τα πίσω. Πολλά από τα κύρια χαρακτηριστικά του αναφέρθηκαν στη σειρά εκδόσεων Python 2.6.x και 2.7.x. Οι εκδόσεις του Python 3 περιλαμβάνουν το βοηθητικό πρόγραμμα 2to3, το οποίο αυτοματοποιεί (τουλάχιστον εν μέρει) τη μετάφραση του κώδικα Python 2 σε Python 3.

Η ημερομηνία λήξης ζωής του Python 2.7 ορίστηκε αρχικά το 2015 και στη συνέχεια αναβλήθηκε για το 2020 λόγω ανησυχίας ότι ένα μεγάλο μέρος του υπάρχοντος κώδικα δεν θα μπορούσε εύκολα να μεταφερθεί προς τα εμπρός στην Python 3. Δεν θα κυκλοφορήσουν άλλα μπαλώματα ασφαλείας ή άλλες βελτιώσεις. Με το τέλος της ζωής του Python 2, υποστηρίζονται μόνο Python 3.6.x και νεότερες εκδόσεις.

Οι Python 3.9.2 και 3.8.8 επιταχύνθηκαν καθώς όλες οι εκδόσεις του Python (συμπεριλαμβανομένου του 2.7) είχαν προβλήματα ασφαλείας, που οδήγησαν σε πιθανή απομακρυσμένη εκτέλεση κώδικα και δηλητηρίαση στην κρυφή μνήμη ιστού.

1.2 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων ανοιχτού κώδικα (RDBMS). Το όνομά του είναι ένας συνδυασμός του "My", το όνομα της κόρης του συνιδρυτή Michael Widenius, και του "SQL", η συντομογραφία για Structured Query Language. Μια σχεσιακή βάση δεδομένων οργανώνει δεδομένα σε έναν ή περισσότερους πίνακες δεδομένων στους οποίους οι τύποι δεδομένων μπορεί να σχετίζονται μεταξύ τους. Αυτές οι σχέσεις βοηθούν στη δομή των δεδομένων. Το SQL είναι μια γλώσσα που χρησιμοποιούν οι προγραμματιστές για τη δημιουργία, τροποποίηση και εξαγωγή δεδομένων από τη σχεσιακή βάση δεδομένων, καθώς και τον έλεγχο της πρόσβασης των χρηστών στη βάση δεδομένων. Εκτός από τις σχεσιακές βάσεις δεδομένων και το SQL, ένα RDBMS όπως το MySQL συνεργάζεται με ένα λειτουργικό σύστημα για την εφαρμογή μιας σχεσιακής βάσης δεδομένων στο σύστημα αποθήκευσης ενός υπολογιστή, διαχειρίζεται τους χρήστες, επιτρέπει την πρόσβαση στο δίκτυο και διευκολύνει τον έλεγχο της ακεραιότητας της βάσης δεδομένων και τη δημιουργία αντιγράφων ασφαλείας.

Η MySQL είναι δωρεάν λογισμικό ανοιχτού κώδικα σύμφωνα με τους όρους της GNU General Public License, και είναι επίσης διαθέσιμο με διάφορες άδειες ιδιοκτησίας. Η MySQL ανήκε και χορηγήθηκε από τη σουηδική εταιρεία MySQL AB, η οποία αγοράστηκε από την Sun Microsystems (τώρα Oracle Corporation). Το 2010, όταν η Oracle απέκτησε την Sun, ο Widenius πήρε το έργο MySQL ανοιχτού κώδικα για να δημιουργήσει το MariaDB.

Η MySQL διαθέτει αυτόνομους πελάτες που επιτρέπουν στους χρήστες να αλληλεπιδρούν απευθείας με μια βάση δεδομένων MySQL χρησιμοποιώντας SQL, αλλά πιο συχνά, η MySQL χρησιμοποιείται με άλλα προγράμματα για την εφαρμογή εφαρμογών που χρειάζονται δυνατότητα σχεσιακής βάσης δεδομένων. Το MySQL είναι ένα συστατικό της στοίβας λογισμικού εφαρμογών LAMP (και άλλων), το οποίο είναι ακρωνύμιο για Linux, Apache, MySQL, Perl/PHP/Python. Το MySQL χρησιμοποιείται από πολλές διαδικτυακές εφαρμογές που βασίζονται σε βάσεις δεδομένων, συμπεριλαμβανομένων των Drupal, Joomla, phpBB και WordPress. Το MySQL χρησιμοποιείται επίσης από πολλούς δημοφιλείς ιστότοπους, συμπεριλαμβανομένου του Facebook, Flickr, MediaWiki, Twitter, και YouTube.

Η MySQL μπορεί να δημιουργηθεί και να εγκατασταθεί με μη αυτόματο τρόπο από τον πηγαίο κώδικα, αλλά συχνότερα εγκαθίσταται από ένα δυαδικό πακέτο, εκτός εάν απαιτούνται ειδικές προσαρμογές. Στις περισσότερες διανομές Linux, το σύστημα διαχείρισης πακέτων μπορεί να κατεβάσει και να εγκαταστήσει MySQL με ελάχιστη προσπάθεια, αν και συχνά απαιτείται περαιτέρω διαμόρφωση για την προσαρμογή των ρυθμίσεων ασφάλειας και βελτιστοποίησης.Πακέτο λογισμικού LAMP, που εμφανίζεται εδώ μαζί με το Squid.

Αν και η MySQL ξεκίνησε ως εναλλακτική λύση χαμηλού επιπέδου σε ισχυρότερες ιδιόκτητες βάσεις δεδομένων, σταδιακά εξελίχθηκε για να υποστηρίξει και ανάγκες υψηλότερης κλίμακας. Εξακολουθεί να χρησιμοποιείται συχνότερα σε εφαρμογές μικρού και μεσαίου μεγέθους ενός διακομιστή, είτε ως συστατικό σε εφαρμογή Ιστού που βασίζεται σε LAMP είτε ως αυτόνομος διακομιστής βάσης δεδομένων. Μεγάλο μέρος της έκκλησης της MySQL προέρχεται από τη σχετική απλότητα και ευκολία χρήσης της, η οποία ενεργοποιείται από ένα οικοσύστημα εργαλείων ανοιχτού κώδικα όπως το phpMyAdmin. Στο μεσαίο εύρος, το MySQL μπορεί να κλιμακωθεί με την ανάπτυξη του σε πιο ισχυρό υλικό, όπως διακομιστή πολλαπλών επεξεργαστών με μνήμη gigabytes.

Υπάρχουν, ωστόσο, όρια στο πόσο μπορεί να κλιμακωθεί η απόδοση σε έναν μόνο διακομιστή («κλιμάκωση»), επομένως σε μεγαλύτερες κλίμακες, απαιτούνται εφαρμογές πολλαπλών διακομιστών MySQL («κλιμάκωση») για να παρέχουν βελτιωμένη απόδοση και αξιοπιστία. Μια τυπική διαμόρφωση υψηλού επιπέδου μπορεί να περιλαμβάνει μια ισχυρή κύρια βάση δεδομένων που χειρίζεται λειτουργίες εγγραφής δεδομένων και αναπαράγεται σε πολλούς υποτελείς που χειρίζονται όλες τις λειτουργίες ανάγνωσης. Ο κύριος διακομιστής σπρώχνει συνεχώς τα συμβάντα binlog σε συνδεδεμένους σκλάβους, ώστε σε περίπτωση αποτυχίας, ένας σκλάβος μπορεί να προωθηθεί ώστε να γίνει ο νέος κύριος, ελαχιστοποιώντας τον χρόνο διακοπής λειτουργίας. Περαιτέρω βελτιώσεις στην απόδοση μπορούν να επιτευχθούν με την προσωρινή αποθήκευση των αποτελεσμάτων από ερωτήματα βάσης δεδομένων στη μνήμη χρησιμοποιώντας memcached, ή με ανάλυση μιας βάσης δεδομένων σε μικρότερα κομμάτια που ονομάζονται θραύσματα, τα οποία μπορούν να εξαπλωθούν σε πολλά κατανεμημένα συμπλέγματα διακομιστών

1.2.1 Η Ιστορία της MySQL

Η MySQL δημιουργήθηκε από μια σουηδική εταιρεία, την MySQL AB, που ιδρύθηκε από τους Σουηδούς David Axmark, Allan Larsson και τον Φινλανδό Σουηδό Michael "Monty" Widenius. Η αρχική ανάπτυξη της MySQL από τους Widenius και Axmark ξεκίνησε το 1994. Η πρώτη έκδοση του MySQL εμφανίστηκε στις 23 Μαΐου 1995. Αρχικά δημιουργήθηκε για προσωπική χρήση από το mSQL με βάση τη γλώσσα χαμηλού επιπέδου ISAM, την οποία οι δημιουργοί θεώρησαν πολύ αργή και άκαμπτη. Δημιούργησαν μια νέα διεπαφή SQL, διατηρώντας παράλληλα το ίδιο API με το mSQL. Διατηρώντας το API συνεπές με το σύστημα mSQL, πολλοί προγραμματιστές μπόρεσαν να χρησιμοποιήσουν το MySQL αντί του προηγούμενου (αποκλειστικά αδειοδοτημένου) mSQL. Ο πρωταρχικός σκοπός ήταν η παροχή αποτελεσματικών και αξιόπιστων επιλογών διαχείρισης δεδομένων σε οικιακούς και επαγγελματικούς χρήστες. Πάνω από μισή ντουζίνα εκδόσεις alpha και beta της πλατφόρμας

κυκλοφόρησαν μέχρι το 2000. Αυτές οι εκδόσεις ήταν συμβατές με όλες σχεδόν τις μεγάλες πλατφόρμες. Αρχικά ιδιοκτησία της MySQL AB, η πλατφόρμα έγινε ανοιχτή πηγή το 2000 και άρχισε να ακολουθεί τους όρους της GPL. Η ανοικτή πηγή οδήγησε σε σημαντική πτώση των εσόδων, τα οποία, ωστόσο, ανακτήθηκαν τελικά. Η φύση ανοικτού κώδικα της MySQL το έκανε ανοιχτό για τις συνεισφορές τρίτων προγραμματιστών. Η MySQL κέρδισε σταθερή δημοτικότητα μεταξύ οικιακών και επαγγελματικών χρηστών και μέχρι το 2001, η πλατφόρμα είχε 2 εκατομμύρια ενεργές εγκαταστάσεις. Το 2002, η εταιρεία διεύρυνε την εμβέλειά της και άνοιξε τα κεντρικά γραφεία των ΗΠΑ εκτός από τα σουηδικά. Την ίδια χρονιά, ανακοινώθηκε ότι η συμμετοχή των πλατφορμών ξεπερνά τα 3 εκατομμύρια χρήστες με έσοδα ύψους 6,5 εκατομμυρίων δολαρίων. Η MySQL AB αντιμετώπισε επίσης την πρώτη της μεγάλη αγωγή τον Ιούνιο του 2001 όταν η NuSphere άσκησε μήνυση στο Περιφερειακό Δικαστήριο των ΗΠΑ στη Βοστώνη. Οι χρεώσεις περιλάμβαναν παραβίαση συμβάσεων τρίτων και αθέμιτο ανταγωνισμό. Σε κατάσταση αναμονής, η MySQL AB μήνυσε τη NuSphere το 2002 για παραβίαση πνευματικών δικαιωμάτων και εμπορικών σημάτων. Και οι δύο εταιρείες κατέληξαν σε διακανονισμό μετά από προκαταρκτική ακρόαση στις 27 Φεβρουαρίου 2002.

1.3 Regex

Η φράση κανονικές εκφράσεις, ή regex, χρησιμοποιείται συχνά για να σημαίνει τη συγκεκριμένη, τυπική σύνταξη κειμένου για την αναπαράσταση μοτίβων για αντιστοίχιση κειμένου, σε αντίθεση με τη μαθηματική σημειογραφία που περιγράφεται παρακάτω. Κάθε χαρακτήρας σε μια κανονική έκφραση (δηλαδή, κάθε χαρακτήρας στη συμβολοσειρά που περιγράφει το πρότυπό του) είναι είτε ένας μεταχαρακτήρας, που έχει ένα ιδιαίτερο νόημα, είτε ένας κανονικός χαρακτήρας που έχει κυριολεκτική σημασία. Για παράδειγμα, στο regex b., Το "b" είναι ένας κυριολεκτικός χαρακτήρας που ταιριάζει μόνο με το "b", ενώ "." είναι ένας μεταχαρακτήρας που ταιριάζει με κάθε χαρακτήρα εκτός από μια νέα γραμμή. Επομένως, αυτό το regex ταιριάζει, για παράδειγμα, με το "b%" ή το "bx" ή το "b5". Μαζί, οι μεταχαρακτήρες και οι κυριολεκτικοί χαρακτήρες μπορούν να χρησιμοποιηθούν για τον προσδιορισμό του κειμένου ενός δεδομένου μοτίβου ή την επεξεργασία μιας σειράς περιπτώσεών του. Οι αντιστοιχίσεις μοτίβου μπορεί να διαφέρουν από μια ακριβή ισότητα σε μια πολύ γενική ομοιότητα, όπως ελέγχεται από τους μεταχαρακτήρες. Για παράδειγμα, . είναι ένα πολύ γενικό μοτίβο, το [a-z] (ταιριάζει με όλα τα πεζά γράμματα από 'a' έως 'z') είναι λιγότερο γενικό και το b είναι ένα ακριβές μοτίβο (ταιριάζει μόνο με το 'b'). Η σύνταξη μεταχαρακτήρα έχει σχεδιαστεί ειδικά για να αντιπροσωπεύει καθορισμένους στόχους με συνοπτικό και ευέλικτο τρόπο για να κατευθύνει την αυτοματοποίηση της επεξεργασίας κειμένου μιας ποικιλίας

δεδομένων εισόδου, σε μια μορφή εύκολη στην πληκτρολόγηση χρησιμοποιώντας ένα τυπικό πληκτρολόγιο ASCII.

Μια πολύ απλή περίπτωση μιας κανονικής έκφρασης σε αυτή τη σύνταξη είναι να εντοπίσετε μια λέξη που γράφεται με δύο διαφορετικούς τρόπους σε έναν επεξεργαστή κειμένου, η κανονική έκφραση seriali [sz] e ταιριάζει τόσο με το "serialise" όσο και με το "serialize". Οι χαρακτήρες μπαλαντέρ επιτυγχάνουν επίσης αυτό, αλλά είναι πιο περιορισμένοι σε αυτό που μπορούν να σχεδιάσουν, καθώς έχουν λιγότερους μεταχαρακτήρες και μια απλή βάση γλώσσας.

Το συνηθισμένο πλαίσιο των χαρακτήρων μπαλαντέρ είναι η παγκοσμιοποίηση παρόμοιων ονομάτων σε μια λίστα αρχείων, ενώ τα regex συνήθως χρησιμοποιούνται σε εφαρμογές που ταιριάζουν γενικά με συμβολοσειρές κειμένου. Για παράδειγμα, το regex ^[\ t]+| [\ t]+$ αντιστοιχεί σε υπερβολικό κενό διάστημα στην αρχή ή στο τέλος μιας γραμμής. Μια προηγμένη κανονική έκφραση που ταιριάζει με κάθε αριθμό είναι [+-]? (\ D+(\. \ D+)? | \. \ D+) ([eE] [+-]? \ D+) ?.

Ένας επεξεργαστής regex μεταφράζει μια κανονική έκφραση στην παραπάνω σύνταξη σε μια εσωτερική αναπαράσταση που μπορεί να εκτελεστεί και να ταιριάξει με μια συμβολοσειρά που αντιπροσωπεύει το κείμενο που αναζητείται. Μια πιθανή προσέγγιση είναι ο αλγόριθμος κατασκευής του Thompson για την κατασκευή ενός μη οριστικοποιημένου πεπερασμένου αυτόματος (NFA), το οποίο Κατόπιν καθίσταται ντετερμινιστική και το προκύπτον ντετερμινιστικό πεπερασμένο αυτόματο (DFA) εκτελείται στη συμβολοσειρά κειμένου στόχου για να αναγνωρίσει υποσύμβολα που ταιριάζουν με την κανονική έκφραση. Η εικόνα δείχνει το σχήμα NFA N (s*) που λαμβάνεται από την κανονική έκφραση s*, όπου το s δηλώνει με τη σειρά του μια απλούστερη κανονική έκφραση, η οποία έχει ήδη μεταφραστεί αναδρομικά στα NFA N (s).

1.3.1 Ιστορία του Regex

Οι κανονικές εκφράσεις ξεκίνησαν το 1951, όταν ο μαθηματικός Stephen Cole Kleene περιέγραψε κανονικές γλώσσες χρησιμοποιώντας τη μαθηματική του σημειογραφία που ονομάζεται κανονικά γεγονότα. Αυτά προέκυψαν στη θεωρητική επιστήμη των υπολογιστών, στα υποπόδια της θεωρίας αυτομάτων (μοντέλα υπολογισμού) και στην περιγραφή και ταξινόμηση των επίσημων γλωσσών. Άλλες πρώτες εφαρμογές της αντιστοίχισης προτύπων περιλαμβάνουν τη γλώσσα SNOBOL, η οποία δεν χρησιμοποιούσε κανονικές εκφράσεις, αλλά αντίθετα τις δικές της κατασκευές που ταιριάζουν με μοτίβα.

Οι κανονικές εκφράσεις εισήλθαν στη δημοφιλή χρήση από το 1968 σε δύο χρήσεις: αντιστοίχιση προτύπων σε επεξεργαστή κειμένου και λεξική ανάλυση σε μεταγλωττιστή. Μεταξύ των πρώτων εμφανίσεων κανονικών εκφράσεων σε μορφή προγράμματος ήταν όταν ο Ken Thompson ενσωμάτωσε το συμβολισμό του Kleene στον επεξεργαστή QED ως μέσο αντιστοίχισης μοτίβων σε αρχεία κειμένου. Για ταχύτητα, ο Thompson εφάρμοσε την τακτική αντιστοίχιση με αντιστοίχιση (JIT) στον κωδικό IBM 7094 στο Συμβατό Σύστημα Χρονικής Ανταλλαγής, ένα σημαντικό πρώιμο παράδειγμα συλλογής JIT. Πρόσθεσε αργότερα αυτή τη δυνατότητα στο πρόγραμμα επεξεργασίας Unix, το οποίο οδήγησε τελικά στο δημοφιλές εργαλείο αναζήτησης τη χρήση κανονικών εκφράσεων από το grep ("grep" είναι μια λέξη που προέρχεται από την εντολή για κανονική αναζήτηση έκφρασης στον επεξεργαστή ed: g/re/p meaning "Παγκόσμια αναζήτηση γραμμών τακτικής έκφρασης και εκτύπωσης"). Περίπου την ίδια εποχή που ο Thompson ανέπτυξε το QED, μια ομάδα ερευνητών συμπεριλαμβανομένου του Douglas T. Ross εφάρμοσε ένα εργαλείο βασισμένο σε κανονικές εκφράσεις που χρησιμοποιείται για λεξική ανάλυση στο σχεδιασμό του μεταγλωττιστή.

Πολλές παραλλαγές αυτών των πρωτότυπων μορφών κανονικών εκφράσεων χρησιμοποιήθηκαν σε προγράμματα Unix στα Bell Labs τη δεκαετία του 1970, συμπεριλαμβανομένων των vi, lex, sed, AWK και expr και σε άλλα προγράμματα όπως το Emacs. Τα Regexes υιοθετήθηκαν στη συνέχεια από ένα ευρύ φάσμα προγραμμάτων, με αυτές τις πρώτες μορφές τυποποιημένες στο πρότυπο POSIX.2 το 1992.

Στη δεκαετία του 1980 εμφανίστηκαν τα πιο περίπλοκα regex στο Perl, τα οποία προέρχονταν αρχικά από μια βιβλιοθήκη regex γραμμένη από τον Henry Spencer (1986), ο οποίος αργότερα έγραψε μια εφαρμογή Advanced Regular Expressions για Tcl. Η βιβλιοθήκη Tcl είναι υβριδική υλοποίηση NFA/DFA με βελτιωμένα χαρακτηριστικά απόδοσης. Τα έργα λογισμικού που έχουν υιοθετήσει την κανονική εφαρμογή έκφρασης Tcl του Spencer περιλαμβάνουν το PostgreSQL. Ο Perl αργότερα επεκτάθηκε στην αρχική βιβλιοθήκη του Spencer για να προσθέσει πολλά νέα χαρακτηριστικά. Μέρος της προσπάθειας στο σχεδιασμό του Raku (που στο παρελθόν ονομαζόταν Perl 6) είναι η βελτίωση της ενσωμάτωσης του Perl στο regex και η αύξηση του πεδίου και των δυνατοτήτων τους για να επιτρέψει τον ορισμό της ανάλυσης γραμματικών εκφράσεων. Το αποτέλεσμα είναι μια μίνι γλώσσα που ονομάζεται κανόνες Raku, οι οποίοι χρησιμοποιούνται για τον καθορισμό της γραμματικής Raku καθώς και για την παροχή ενός εργαλείου στους προγραμματιστές στη γλώσσα. Αυτοί οι κανόνες διατηρούν τα υπάρχοντα χαρακτηριστικά του Perl 5.x regex, αλλά επιτρέπουν επίσης τον ορισμό του στυλ BNF ενός αναδρομικού αναλυτή καταγωγής μέσω υπο-κανόνων.

Η χρήση των regex σε πρότυπα δομημένης πληροφόρησης για τη μοντελοποίηση εγγράφων και βάσεων δεδομένων ξεκίνησε τη δεκαετία του 1960 και επεκτάθηκε τη δεκαετία του 1980, όταν τα βιομηχανικά πρότυπα όπως το ISO SGML (που προηγήθηκαν από την ANSI

"GCA 101-1983") ενοποιήθηκαν. Ο πυρήνας των προτύπων γλώσσας προδιαγραφών δομής αποτελείται από regexes. Η χρήση του είναι εμφανής στη σύνταξη της ομάδας στοιχείων DTD.

Ξεκινώντας το 1997, ο Philip Hazel ανέπτυξε το PCRE (Perl Compatible Regular Expressions), το οποίο προσπαθεί να μιμηθεί στενά τη λειτουργικότητα του Perl και χρησιμοποιείται από πολλά σύγχρονα εργαλεία, συμπεριλαμβανομένου του PHP και του Apache HTTP Server.

Σήμερα, τα regex υποστηρίζονται ευρέως σε γλώσσες προγραμματισμού, προγράμματα επεξεργασίας κειμένου (ιδιαίτερα lexers), προηγμένους επεξεργαστές κειμένου και ορισμένα άλλα προγράμματα. Η υποστήριξη Regex είναι μέρος της τυπικής βιβλιοθήκης πολλών γλωσσών προγραμματισμού, συμπεριλαμβανομένων των Java και Python, και είναι ενσωματωμένη στη σύνταξη άλλων, συμπεριλαμβανομένων των Perl και ECMAScript. Οι εφαρμογές της λειτουργικότητας regex συχνά ονομάζονται μηχανές regex και πολλές βιβλιοθήκες είναι διαθέσιμες για επαναχρησιμοποίηση. Στα τέλη της δεκαετίας του 2010, αρκετές εταιρείες άρχισαν να προσφέρουν υλισμικό, FPGA, GPU υλοποιήσεις συμβατών μηχανών regex με PCRE που είναι ταχύτερες σε σύγκριση με τις εφαρμογές της CPU.

1.4 XAMPP

Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP προϋποθέτει μόνο τα λογισμικά συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης. Το XAMPP έχει δυνατότητα αναβάθμισης σε νέες εκδόσεις του εξυπηρετητή ιστοσελίδων http Apache, της βάσης δεδομένων MySQL, της γλώσσας PHP και Perl. Το XAMPP συμπεριλαμβάνει επίσης τα πακέτα OpenSSL και το phpMyAdmin.

Επίσης το XAMPP είναι ένα πακέτο στοίβας λύσεων διακομιστή ιστού δωρεάν και ανοιχτού κώδικα που δημιουργήθηκε από τους Apache Friends, που αποτελείται κυρίως από τον Apache HTTP Server, τη βάση δεδομένων MariaDB και διερμηνείς για σενάρια γραμμένα σε γλώσσες προγραμματισμού PHP και Perl. Δεδομένου ότι οι περισσότερες πραγματικές αναπτύξεις διακομιστή ιστού χρησιμοποιούν τα ίδια στοιχεία με το XAMPP, καθιστά δυνατή τη μετάβαση από έναν τοπικό δοκιμαστικό διακομιστή σε έναν ζωντανό διακομιστή.

Η ευκολία ανάπτυξης του XAMPP σημαίνει ότι μια στοίβα WAMP ή LAMP μπορεί να εγκατασταθεί γρήγορα και απλά σε ένα λειτουργικό σύστημα από έναν προγραμματιστή, με το

πλεονέκτημα ότι οι κοινές πρόσθετες εφαρμογές όπως το WordPress και το Joomla! μπορεί επίσης να εγκατασταθεί με παρόμοια ευκολία χρησιμοποιώντας το Bitnami.

Το πιο προφανές χαρακτηριστικό του XAMPP είναι η ευκολία με την οποία μπορεί να αναπτυχθεί και να δημιουργηθεί μια στοίβα διακομιστή WAMP.Αργότερα, ορισμένες κοινές συσκευασμένες εφαρμογές που μπορούσαν να εγκατασταθούν εύκολα παρέχονται από το Bitnami.

Επισήμως, οι σχεδιαστές του XAMPP το προορίζουν για χρήση μόνο ως εργαλείο ανάπτυξης, για να επιτρέψουν στους σχεδιαστές ιστοσελίδων και προγραμματιστές να δοκιμάσουν τη δουλειά τους στους δικούς τους υπολογιστές χωρίς πρόσβαση στο Διαδίκτυο. Για να γίνει αυτό όσο το δυνατόν πιο εύκολο, πολλές σημαντικές λειτουργίες ασφαλείας απενεργοποιούνται από προεπιλογή. Το XAMPP έχει τη δυνατότητα να εξυπηρετεί ιστοσελίδες στον Παγκόσμιο Ιστό. Παρέχεται ειδικό εργαλείο για την προστασία με κωδικό πρόσβασης των πιο σημαντικών τμημάτων του πακέτου.

Το XAMPP παρέχει επίσης υποστήριξη για τη δημιουργία και τον χειρισμό βάσεων δεδομένων σε MariaDB και SQLite μεταξύ άλλων.

Μόλις εγκατασταθεί το XAMPP, είναι δυνατό να αντιμετωπιστεί ένα localhost σαν απομακρυσμένος κεντρικός υπολογιστής συνδέοντας χρησιμοποιώντας ένα πρόγραμμα - πελάτη FTP. Η χρήση ενός προγράμματος όπως το FileZilla έχει πολλά πλεονεκτήματα κατά την εγκατάσταση ενός συστήματος διαχείρισης περιεχομένου (CMS) όπως το Joomla ή το WordPress [απαιτείται περαιτέρω εξήγηση]. Είναι επίσης δυνατό να συνδεθείτε στο localhost μέσω FTP με έναν επεξεργαστή HTML.

2 Διαδικασία Crawling

Στο κεφάλαιο δύο θα προσεγγίσουμε με θεωρητικό τρόπο την συνολική διαδικασία που ακολουθήσαμε στο πρώτο μέρος της διπλωματικής εργασίας. Αρχικά θα αναλύσουμε την διαδικασία που ακολουθήσαμε για να επιλέξουμε την συγκεκριμένη ιστοσελίδα για να πάρουμε τις πληροφορίες για τις αγγελίες εργασίας. Στην συνέχεια θα εξετάσουμε τον λόγο που χρειαστήκαμε την SQL βάση δεδομένων καθώς και την χρησιμότητα του κάθε πίνακα που δημιουργήσαμε ξεχωριστά. Τέλος θα αναλύσουμε βήμα προς βήμα την διαδικασία που ακολουθήσαμε ώστε να προγραμματίσουμε τον crawler με σκοπό να «τραβάει» όλες τις αγγελίες εργασίας που δημοσιεύονται καθημερινά στην ιστοσελίδα και ο τρόπος με τον οποίο αποθηκεύονται στην βάση δεδομένων μας.

2.1 Επιλογή Web-Site

Η επιλογή της ιστοσελίδας γίνεται με κάποιες συγκεκριμένες τεχνικές και μηχανισμούς. Πρώτα από όλα θα πρέπει να κάνουμε αναζήτηση στο google τις λέξεις: " Εύρεση εργασίας", "Εύρεση δουλείας" και "Θέσεις εργασίας", στην συνέχεια θα κρατήσουμε όσα site μας εμφανίσει σαν αποτέλεσμα και εξετάζουμε την σχετικότητα αυτών σε σχέση με το αντικείμενο μας. Εφόσον διαπιστώσουμε συσχέτιση τότε αξιολογούμε το website στο Alexa, AHREF authority και MOZ authority. Τέλος για κάθε site στο Alexa, υπάρχει η ενότητα SIMILAR SITES BY AUDIENCE OVERLAP, ακολουθούμε την συγκεκριμένη ενότητα μέχρι 2 επίπεδα δηλαδή κρατάμε τα similar sites και τα similar των similar και κάνουμε ξανά έλεγχο για σχετικότητα με το αντικείμενο μας, να είναι δηλαδή site στα οποία υπάρχουν αγγελίες εργασίας και δεν είναι άρθρα.

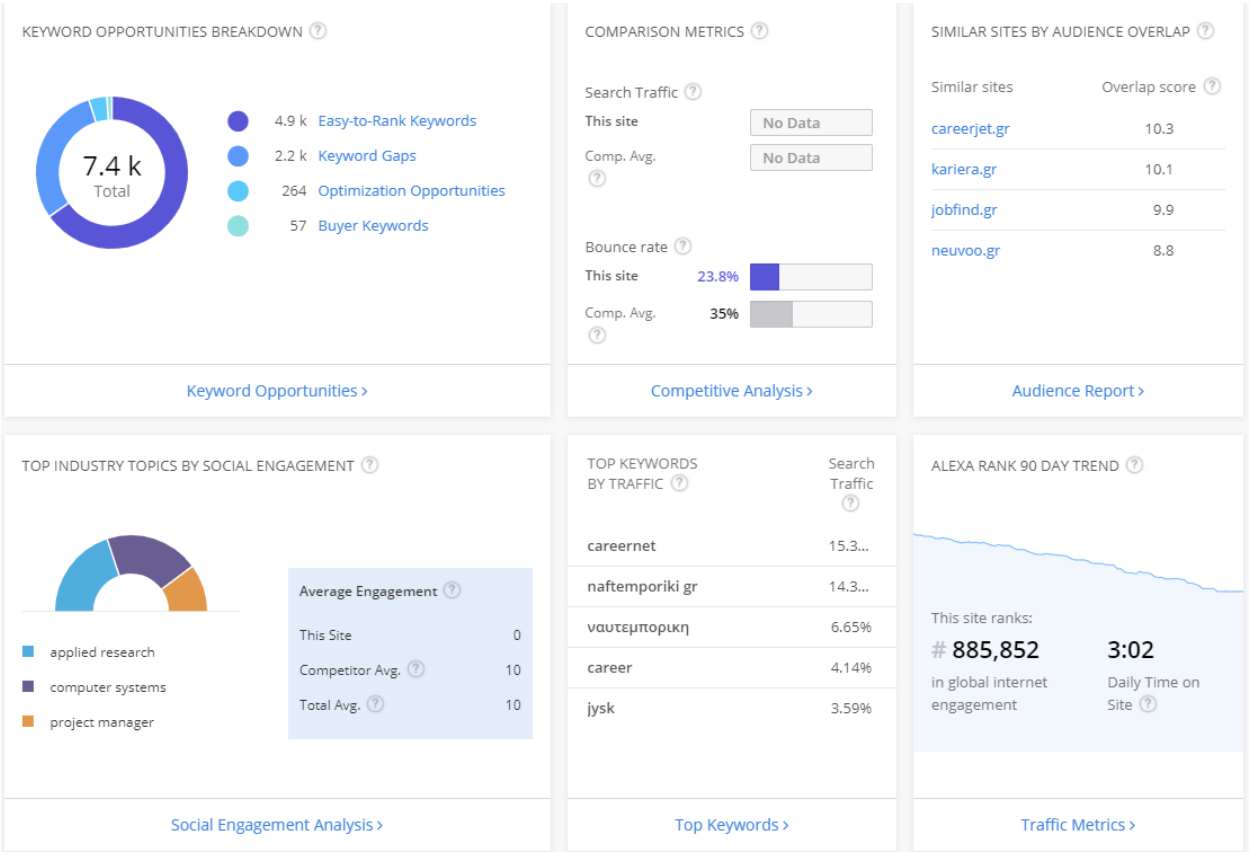
2.1.1 Alexa rank

Η Alexa Internet, Inc. είναι μια αμερικανική εταιρεία ανάλυσης επισκεψιμότητας ιστού με έδρα το Σαν Φρανσίσκο. Είναι θυγατρική εξ ολοκλήρου της Amazon.

Η Alexa ιδρύθηκε ως ανεξάρτητη εταιρεία το 1996 και εξαγοράστηκε από την Amazon το 1999 για 250 εκατομμύρια δολάρια σε απόθεμα. Η Alexa παρέχει δεδομένα επισκεψιμότητας ιστού, παγκόσμια κατάταξη και άλλες πληροφορίες σε πάνω από 30 εκατομμύρια ιστότοπους. Η Alexa εκτιμά την επισκεψιμότητα του ιστοτόπου με βάση ένα δείγμα εκατομμυρίων χρηστών του Διαδικτύου που χρησιμοποιούν επεκτάσεις προγράμματος περιήγησης, καθώς και από ιστότοπους που έχουν επιλέξει να εγκαταστήσουν ένα σενάριο Alexa. Από το 2020, ο ιστότοπός του επισκέπτεται πάνω από 400 εκατομμύρια άτομα κάθε μήνα. Περιστασιακές διαφωνίες σχετικά με τις αξιώσεις του συμβαίνουν, αλλά είναι συνολικά πολύ περιεκτικό.

Η Alexa Traffic Rank βασίζεται στον αριθμό της επισκεψιμότητας που καταγράφεται από χρήστες που έχουν εγκαταστήσει τη γραμμή εργαλείων Alexa σε διάστημα τριών μηνών. Η

κατάταξη ενός ιστότοπου βασίζεται σε ένα συνδυασμένο μέτρο μοναδικών επισκεπτών και προβολών σελίδων. Οι μοναδικοί επισκέπτες καθορίζονται από τον αριθμό των μοναδικών χρηστών της Alexa που επισκέπτονται έναν ιστότοπο μια δεδομένη ημέρα. Οι προβολές σελίδας είναι ο συνολικός αριθμός αιτημάτων URL χρήστη Alexa για έναν ιστότοπο. Οι Βαθμοί επισκεψιμότητας της Alexa προορίζονται μόνο για τομείς και δεν δίνουν ξεχωριστές βαθμολογίες για υποσέλιδές σε έναν τομέα ή υποτομείς.



Εικόνα 2—1 Alexa Rank

2.1.2 AHREF Authority

Η "εξουσιοδότηση ιστότοπου" είναι μια έννοια SEO που αναφέρεται στην "δύναμη" ενός δεδομένου τομέα.

Μερικοί άνθρωποι ονομάζουν αυτό "εξουσιοδότηση τομέα", το οποίο δεν πρέπει να συγχέεται με τη μέτρηση του Domain Authority (DA) της Moz. Όταν μιλάμε για εξουσιοδότηση

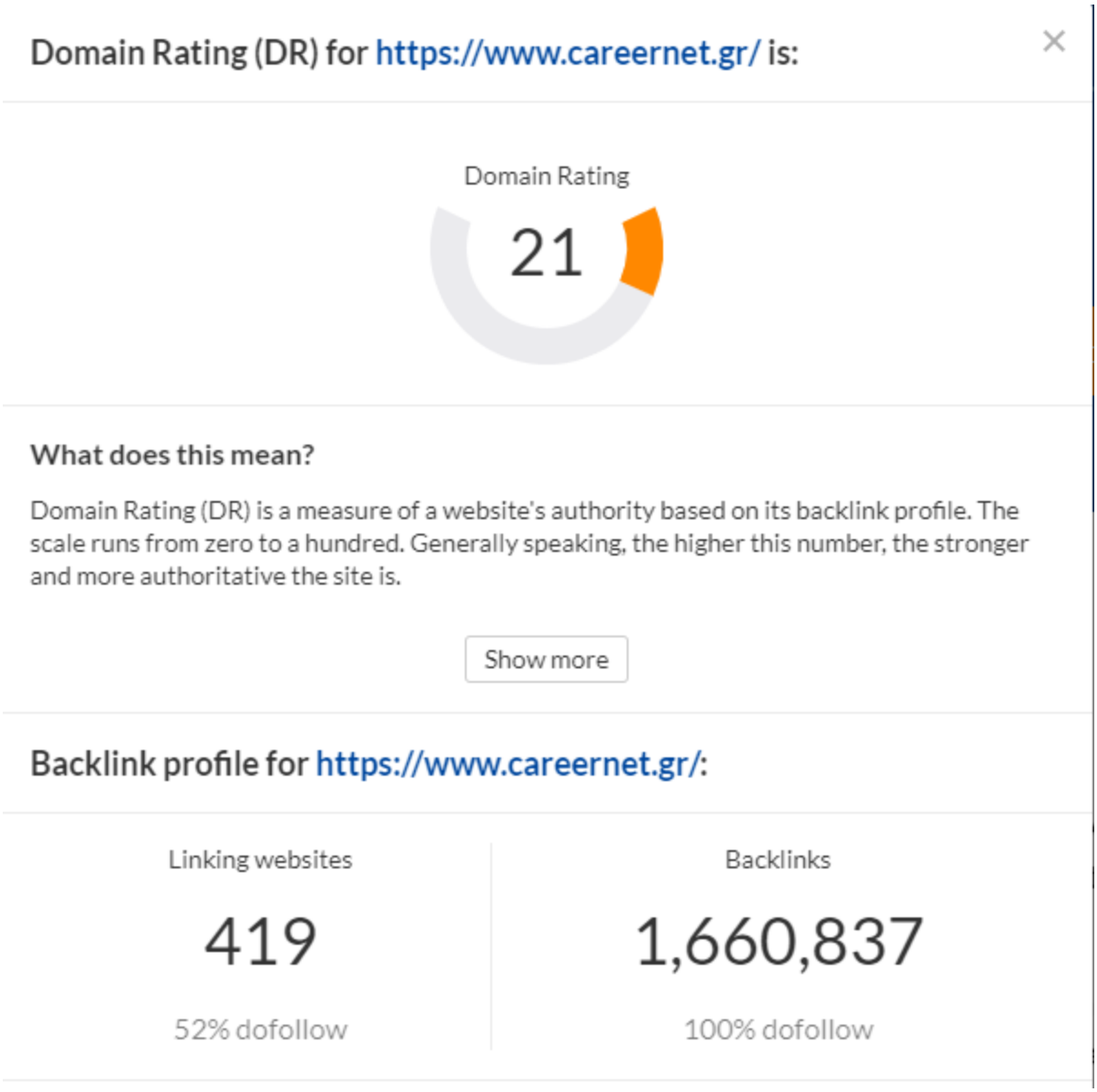
τομέα, μιλάμε για μια γενική έννοια SEO που είναι συνώνυμη με τον όρο "αρχή του ιστότοπου".

Εδώ στο Ahrefs, έχουμε μια δική μας μέτρηση αρχής ιστότοπου που ονομάζεται Αξιολόγηση τομέα. Λειτουργεί σε κλίμακα από μηδέν έως εκατό. Όσο υψηλότερη είναι η αξιολόγηση τομέα (DR) ενός ιστότοπου, τόσο ισχυρότερη και πιο έγκυρη είναι.

Το παραπάνω δωρεάν εργαλείο εμφανίζει την "εξουσία" του ιστότοπού σας όπως υπολογίστηκε από το Ahrefs (δηλαδή, Αξιολόγηση τομέα).

2.1.2.1 Πώς υπολογίζεται η βαθμολογία Domain Rating(DR)

- Κοιτάξτε πόσοι μοναδικοί τομείς συνδέονται με τον ιστότοπο προορισμού.
- Κοιτάξτε την "εξουσία" αυτών των τομέων σύνδεσης.
- Λάβετε υπόψη σας πόσους μοναδικούς τομείς συνδέονται καθένας από αυτούς τους ιστότοπους
- Εφαρμόστε κάποια μαθηματική και μαγική κωδικοποίηση για να υπολογίσετε «ακατέργαστες» βαθμολογίες DR.
- Σχεδιάστε αυτές τις βαθμολογίες σε κλίμακα 100 βαθμών



Εικόνα 2—2 AHREF authority

2.1.3 MOZ Authority

Η αρχή τομέα (αναφέρεται επίσης ως ηγετική σκέψη) ενός ιστότοπου περιγράφει τη συνάφειά του για μια συγκεκριμένη θεματική περιοχή ή βιομηχανία. Το Domain Authority είναι μια βαθμολογία κατάταξης μηχανών αναζήτησης που αναπτύχθηκε από τη Moz. Αυτή η συνάφεια έχει άμεσο αντίκτυπο στην κατάταξή της από τις μηχανές αναζήτησης, προσπαθώντας να αξιολογήσει την εξουσία τομέα μέσω αυτοματοποιημένων αναλυτικών

αλγορίθμων. Η συνάφεια της εξουσιοδότησης τομέα με την καταχώριση ιστότοπων στα SERP των μηχανών αναζήτησης οδήγησε στη δημιουργία μιας ολόκληρης βιομηχανίας παρόχων Black Hat SEO, προσπαθώντας να προσποιηθεί ένα αυξημένο επίπεδο εξουσίας τομέα. Η κατάταξη από τις μεγάλες μηχανές αναζήτησης, π.χ., το Page Rank της Google είναι αγνωστική για συγκεκριμένους κλάδους ή θεματικούς τομείς και αξιολογεί έναν ιστότοπο στο πλαίσιο του συνόλου των ιστότοπων στο Διαδίκτυο. Τα αποτελέσματα στη σελίδα SERP ορίζουν το Page Rank στο πλαίσιο μιας συγκεκριμένης λέξης -κλειδιού. Σε λιγότερο ανταγωνιστικό θέμα, ακόμη και ιστότοποι με χαμηλό Page Rank μπορούν να επιτύχουν υψηλή προβολή στις μηχανές αναζήτησης, καθώς οι ιστότοποι με την υψηλότερη κατάταξη που ταιριάζουν με συγκεκριμένες λέξεις αναζήτησης τοποθετούνται στις πρώτες θέσεις των SERP.

Top Pages by Links

The site's most important pages based on Page Authority (PA), an algorithm of link metrics.

Page/URL	PA
careernet.gr/	35
www.careernet.gr/aggelies	34
www.careernet.gr/	32
www.careernet.gr/ergodotes/proslavanoun	27
www.careernet.gr/aggelies/naftilia	26
www.careernet.gr/login?returnUrl=%2F	26
marine.careernet.gr/	25

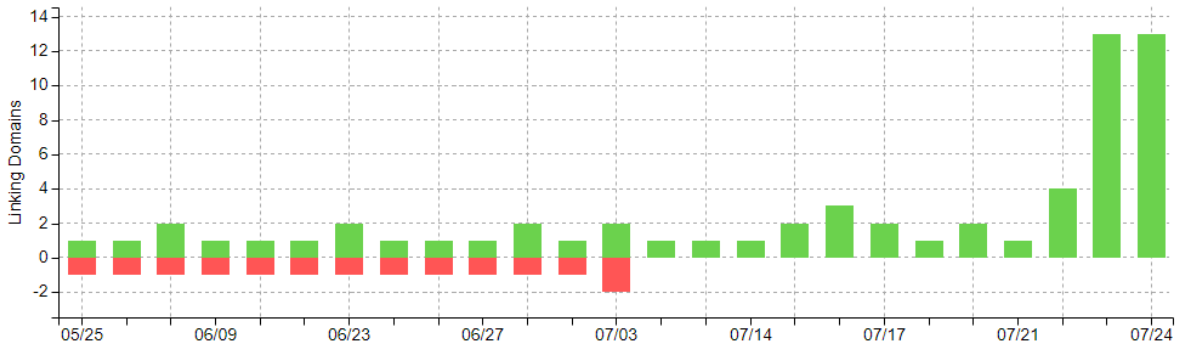
Top Linking Domains

The top linking domains based on Domain Authority (DA), a metric which predicts ranking potential based on links.

Domain	DA
msn.com	95
bit.ly	94
meetup.com	92
naftemporiki.gr	76
auth.gr	68
worldwidetopsite.com	65
faucre.com	63

Discovered and Lost Linking Domains

Track when we found new linking domains over the past 60 days.



Εικόνα 2—3 MOZ authority

2.2 Βάση δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά μορφοποιημένα σχετιζόμενα δεδομένα στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση.

Ειδικότερα, στην επιστήμη της πληροφορικής και στην καθημερινή χρήση των ηλεκτρολογικών υπολογιστών, με τον όρο βάσεις δεδομένων αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων δεδομένων ηλεκτρονικά και ψηφιακά αποθηκευμένων, στο λογισμικό που χειρίζεται τέτοιες συλλογές(Σύστημα Διαχείρισης Βάσεων Δεδομένων, ή DBMS) και στο γνωστικό πεδίο που μελετά. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων μέσω του σχεδιασμού και του τρόπου ιεράρχησης της, τα αποκαλούμενα σύστημα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

Η SQL είναι μια γλώσσα υπολογιστώ στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα σχεσιακών βάσεων δεδομένων(Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στην σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL ήταν μία από τίς πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

Ο SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φόρα στην αγορά το 1989 σε συνεργασία με την Sybase. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11.000.000 εγκαταστάσεις. Έλαβε το όνομα της από την κόρη του Μόντυ Βιντένιους, τη Μάι. Το πρόγραμμα τρέχει έναν εξυπηρετητή(server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία μοναδική κερδοσκοπική εταιρία, τη Σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

2.2.1 Υλοποίηση Βάσης δεδομένων

Ο λόγος που χρειαστήκαμε τις βάσεις δεδομένων στην διπλωματική εργασία αυτή είναι, αρχικά χρειάστηκε να αποθηκεύσουμε όλες τις θέσεις εργασίας που δημοσιεύονται καθημερινά σε μια σωστά δομημένη μορφή δηλαδή, στην συνέχεια χρειάστηκε μέσα από τις αποθηκευμένες πληροφορίες του web-crawler να κρατήσουμε τα χρήσιμα κομμάτια και γι' αυτό τον λόγο χρειαστήκαμε τρείς πίνακες στους οποίους αποθηκεύσαμε τις πληροφορίες

μας. Στον πρώτο πίνακα που τον ονομάσαμε ως columns κρατήσαμε τις πληροφορίες που πήραμε από τον crawler, στον δεύτερο πίνακα που τον ονομάσαμε tester κρατήσαμε τις πληροφορίες που πήραμε από το regular expression ενώ τον τρίτο πίνακα που τον ονομάσαμε options τον χρειαστήκαμε ως μέσο ώστε να μην υπάρχουν διπλότυπες εγγραφές στον πίνακα με όνομα tester.

2.2.2 Database Columns

Στον πίνακα columns αποθηκεύουμε τις πληροφορίες που παίρνουμε από τον crawler, πιο συγκεκριμένα κρατάμε το id, το οποίο αυξάνεται μόνο κάθε φορά που αποθηκεύεται μια αγγελία στην βάση μας, το URL της αγγελίας, το HTML αρχείο της και την ημερομηνία που περάστηκε στην βάση δεδομένων μας. Το URL το χρειαζόμαστε για την δημιουργία μια συνάρτησης(function) με την οποία ελέγχουμε εάν η αγγελία αυτή είναι ήδη περασμένη ή όχι, πιο αναλυτικά το URL είναι χρήσιμο σε αυτό το στάδιο καθώς μας βοηθά στην συνάρτηση του crawler στην οποία ελέγχουμε μία-μία τις αγγελίες που είναι ήδη περασμένες στον πίνακα columns και τις συγκρίνουμε ξανά μία προς μία με τα URL που είναι δημοσιευμένα, το HTML αρχείο το χρειαστήκαμε καθώς εκεί βρίσκεται όλη η χρήσιμη πληροφορία με την οποία και ασχοληθήκαμε, πιο συγκεκριμένα εκεί είναι «κρυμμένη» όλη η πληροφορία που κρύβει ένα HTML αρχείο, δηλαδή στην συγκεκριμένη περίπτωση αυτό που εμείς χρειαζόμασταν με στόχο να φτάσουμε στο τελικό στάδιο που ήταν η ανάλυση των δεδομένων μας ήταν, ο τίτλος της αγγελίας, η επωνυμία της επιχείρησης που την δημοσίευσε, η διεύθυνση καθώς και η περιοχή για την οποία προορίζεται η αγγελία. Τέλος το id μας ήταν χρήσιμο στο regular expression καθώς με βάση το τελευταίο id που είχαμε στον πίνακα columns γινόταν έλεγχος στον πίνακα options ώστε εάν η τιμή της στήλης value ήταν ίδια με αυτή της τιμής του id στον πίνακα columns.

id ▲ 1	job_cat	job_url	job_description	processed	Date
1		https://www.careernet.gr/aggelia/30893/payroll-con...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
2		https://www.careernet.gr/aggelia/25016/sumboulos-p...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
3		https://www.careernet.gr/aggelia/31237/hr-sales-co...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
4		https://www.careernet.gr/aggelia/25114/stelexos-te...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
5		https://www.careernet.gr/aggelia/23817/proswpikos-...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
6		https://www.careernet.gr/aggelia/31950/customer-ca...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
7		https://www.careernet.gr/aggelia/31165/sumboulos-p...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
8		https://www.careernet.gr/aggelia/25111/ekproswpoi-...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
9		https://www.careernet.gr/aggelia/25884/stelexos-ti...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
10		https://www.careernet.gr/aggelia/23584/sumbouloi-p...	<!DOCTYPE html> <html data-adman-async=true> <he...	0	2021-05-07
11		https://www.careernet.gr/aggelia/24040/sumboulos-e...	<!DOCTYPE html> <html data-adman-async=true>	0	2021-05-07

Εικόνα 2—4 Table columns

2.2.3 Database Options

Τον πίνακα options τον χρειαστήκαμε στο regular expression καθώς αυτός ο πίνακας έδρασε ως μέθοδος deduplication στο πρόγραμμα μας, δηλαδή σε αυτόν τον πίνακα μας ενδιέφερε η γραμμή που είχε id = 7 και η τιμή της στήλης value αλλά και πάλι για id = 7, η τιμή της στήλης value αυξάνεται κάθε φόρα που έχουμε μία νέα εγγραφή στον πίνακα tester(ο πίνακας στον οποίο αποθηκεύονται οι επεξεργασμένες πληροφορίες) όμως η στήλη value έχει την ιδιότητα να κάνει και έλεγχο πέραν από το να αυξάνεται και αυτός είναι κ σκοπός για τον οποίο αυξάνεται όταν έχουμε νέα εγγραφή, πιο αναλυτικά η τιμή που κρατάει η στήλη value είναι για να γίνει ο έλεγχος στο regular expression, δηλαδή η τιμή της στήλης value συσχετίζεται με την τιμή του id του πίνακα columns, εάν η τιμή της στήλης value είναι ίση με την τιμή του id

στον πίνακα columns τότε αυτό σημαίνει πως δεν έχουμε κάποια καινούρια εγγραφή στον πίνακα columns, σε άλλη περίπτωση δηλαδή εάν η τιμή της value είναι μικρότερη από την τιμή του id στον πίνακα columns τότε «λέει» στο πρόγραμμα να επιστρέψει την τιμή του id από τον columns που θα είναι ίση με value+1.

id	cr_option	value
1	last-fetch	117471
2	last-value	398298
3	matching_esco	851380
4	hierarchy	101578
5	last-processed	644127
6	deduplication_opt	1
7	careernet	3080
49617	last-is-active-offset	1446
3816372	onet-last-is-active-offset	420

Εικόνα 2—5 Table options

2.2.4 Database Tester

Στον πίνακα tester αποθηκεύουμε τις πληροφορίες που «κρατάμε» από το regular expression, πιο συγκεκριμένα αποθηκεύουμε τις πληροφορίες που πήραμε από το HTML αρχείο της κάθε αγγελίας όλα αυτά που αποθηκεύουμε στον πίνακα αυτό είναι όσα υπήρχαν στο HTML αρχείο και ήταν «κρυμμένα» σε διάφορες εντολές της HTML σελίδας και είναι αυτές με τις οποίες ασχοληθήκαμε κατά την ανάλυση στην οποία έγινε. Με βάση την στήλη value που υπάρχει στον πίνακα options έγινε ο έλεγχος όπως αναφέραμε και στο κεφάλαιο 1.2.3 και έτσι η κάθε αγγελία είναι μοναδική στον πίνακα tester. Επιπροσθέτως, στον πίνακα tester αποθηκεύουμε τον τίτλος της επιχείρησης, τον τίτλος της αγγελίας, τα στοιχεία της αγγελίας, την διεύθυνση, περιοχή, χώρα για τα οποία προορίζεται η αγγελία και τέλος ημερομηνία δημοσίευσης καθώς και διαγραφής της αγγελίας.

id	title	description	employ_type	company_name	address_country	address_region	address_locality	address_street	date_posted	date_end
1	Payroll Consultant	<p style=text-align: center;>Payroll Consu...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
2	Σύμβουλος Προώθησης Υπηρεσιών Ενέργειας (Τηλεργασί...	Σύμβουλος Προώθησης Υπηρεσιών Ενέρ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
3	HR Sales Consultant	HR Sales Consultant <st...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
4	Στέλεχος Τεχνικής Εξυπηρέτησης Πελατών / Forthnet	Στέλεχος Τεχνικής Εξυπηρέτησης Πελ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
5	Προσωπικός Σύμβουλος High Value Πελατών / Vodafone	Προσωπικός Σύμβουλος High Value Πε...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
6	Customer Care Representative	ICAP Employment Solutions, part of ICAP Group, ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
7	Σύμβουλος Προώθησης Τηλεπικοινωνιακών Υπηρεσιών	Σύμβουλος Προώθησης Τηλεπικοινωνιακών Υ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
8	Εκπρόσωποι Τεχνικής Υποστήριξης Πελατών / 1st Leve...	<p style=text-align: center;>Εκπρόσωποι Τε...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
9	Στέλεχος Τηλεφωνικής Εξυπηρέτησης Πελατών σε Τραπε...	Στέλεχος Τηλεφωνικής Εξυ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
10	Σύμβουλοι Προώθησης Υπηρεσιών Ηλεκτρικής Ενέργειας	Σύμβουλοι Προώθησης Υπηρεσιών Ηλεκ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Ελληνικό	Ελληνικό	2021-05-07	2021-06-11
11	Σύμβουλος εξυπηρέτησης πελατών εισερχόμενων κλήσεω...	Σύμβουλος εξυπηρέτησης πελατών εισ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
12	Σύμβουλος Εξυπηρέτησης και Διακράτησης Πελατών / F...	Σύμβουλος Εξυπηρέτησης και Διακράτ...	FULL_TIME	ICAP Employment Solutions	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-06-11
13	Operations Manager	Operations Manager for Dry Bul...	FULL_TIME	Shipping company	Ελλάδα	Αθήνα	Αθήνα	Αθήνα	2021-05-07	2021-05-22

Εικόνα 2—6 Table tester

2.3 Web Crawler

Οι web crawlers (ή αλλιώς spiders όπως είναι ευρέως γνωστοί), είναι αυτοματοποιημένα προγράμματα που διαπερνούν το παγκόσμιο ιστό με κάποια συγκεκριμένη τακτική. Η διαδικασία που επιτελεί ένας web crawler καλείται web crawling και είναι μία διαδικασία που χρησιμοποιείται κατά κύριο λόγο από τις υπηρεσίες δεικτοδότησης ώστε να «κατεβάσουν» τις σελίδες του διαδικτύου. Γενικά, ένας crawler ξεκίνα από μία λίστα URLs που πρόκειται να επισκεφτεί. Συνεχίζει αναδρομικά βρίσκοντας τα links στις σελίδες που επισκέπτεται και τερματίζει αφού καλυφθούν κάποιες παράμετροι, π.χ. πόσα web-pages έχει το website ώστε να φτάσει στο τελευταίο, κλπ.

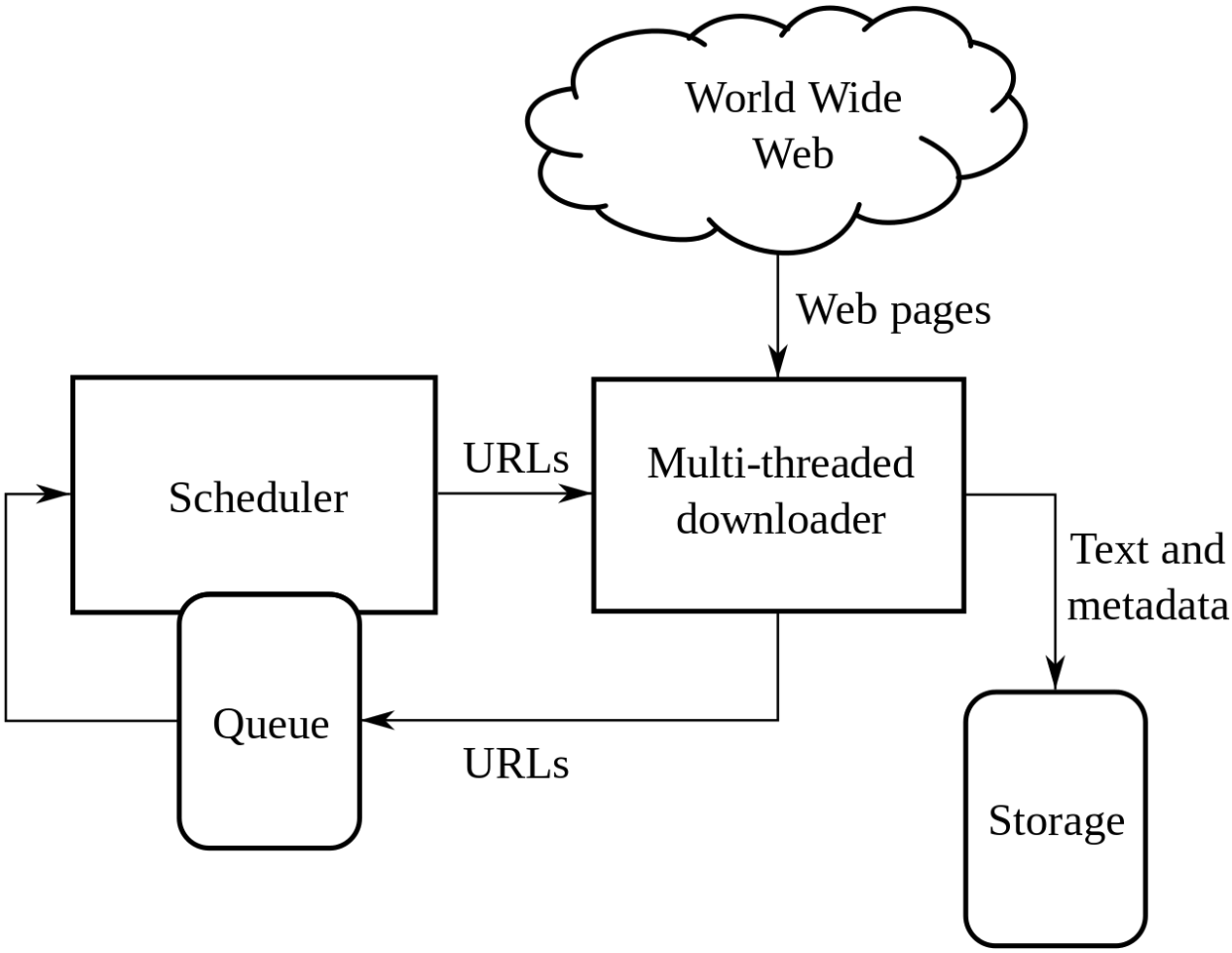
Το πλήθος των web crawlers είναι αρκετά μεγάλο και αν εξαιρέσουμε τους εξειδικευμένους web crawler παρατηρούμε πως οι περισσότεροι έχουν σαν σκοπό να συλλέξουν όλες τις HTML σελίδες από τις οποίες απαρτίζεται ένας δικτυακός τόπος μαζί με τα βοηθητικά αρχεία(pdf, video, css, javascript) και ουσιαστικά να δημιουργήσουν ένα offline-instance του δικτυακού τόπου τον οποίο προελαύνουν.

Ο web crawler, πρόκειται για έναν από τους πρώτους crawlers που κατασκευάστηκαν από τον Pinkerton το 1994. Βασίστηκε στη βιβλιοθήκη WWW, προκειμένου να είναι σε θέση να κατεβάζει σελίδες από το διαδίκτυο ενώ χρησιμοποιούσε ένα δεύτερο πρόγραμμα προκειμένου να διαβάζει τα URLs τα οποία πρέπει να προσπελάσει. Ο αλγόριθμος

προσπέλασης ήταν κατά πλάτος αναζήτηση του γραφήματος μίας ιστοσελίδας σε συνδυασμό με αποφυγή των σελίδων που έχει ήδη επισκεφθεί. Ένα αξιοσημείωτο στοιχείο ήταν η δυνατότητα να ακολουθεί συγκεκριμένα μόνο links σε ένα δικτυακό τόπο και όχι όλα βάση του ερωτήματος που έθετε ο χρήστης. Ήταν κάτι σαν ένας crawler πραγματικού χρόνου που φυσικά μπορούσε να ανταποκριθεί πλήρως λόγω του μικρού μεγέθους που είχε το διαδίκτυο. Ο web crawler το 2001 είναι τμήμα της Infospace η οποία τον χρησιμοποιεί ως βάση για την ομώνυμη μεταμηχανή αναζήτησης.

Ένα πρόγραμμα ανίχνευσης Ιστού (web crawler) ξεκινά με μια λίστα διευθύνσεων URL για επίσκεψη, που ονομάζονται seed. Καθώς το πρόγραμμα ανίχνευσης επισκέπτεται αυτές τις διευθύνσεις URL, προσδιορίζει όλους τους υπερσύνδεσμους στις σελίδες και τις προσθέτει στη λίστα των διευθύνσεων URL για επίσκεψη, που ονομάζονται σύνορα ανίχνευσης. Οι διευθύνσεις URL από τα σύνορα επισκέπτονται αναδρομικά σύμφωνα με ένα σύνολο πολιτικών. Εάν το πρόγραμμα ανίχνευσης εκτελεί αρχειοθέτηση ιστότοπων (ή αρχειοθέτηση ιστού), αντιγράφει και αποθηκεύει τις πληροφορίες όσο περνά. Τα αρχεία αποθηκεύονται συνήθως με τέτοιο τρόπο ώστε να μπορούν να προβληθούν, να διαβαστούν και να προηγηθούν σαν αν ήταν στον ζωντανό ιστό, αλλά διατηρούνται ως «στιγμιότυπα».

Το αρχείο είναι γνωστό ως αποθετήριο και έχει σχεδιαστεί για την αποθήκευση και διαχείριση της συλλογής ιστοσελίδων. Το αποθετήριο αποθηκεύει μόνο σελίδες HTML και αυτές οι σελίδες αποθηκεύονται ως ξεχωριστά αρχεία. Ένα αποθετήριο είναι παρόμοιο με οπουδήποτε άλλο σύστημα που αποθηκεύει δεδομένα, όπως μια σύγχρονη βάση δεδομένων. Η μόνη διαφορά είναι ότι ένα αποθετήριο δεν χρειάζεται όλε τις λειτουργίες που προσφέρει ένα σύστημα βάσης δεδομένων. Το αποθετήριο αποθηκεύει την πιο πρόσφατη έκδοση της ιστοσελίδας που ανακτήθηκε από το πρόγραμμα ανίχνευσης.



Εικόνα 2—7 Architecture of a web crawler

2.3.1 Κατασκευή του Crawler

Για την κατασκευή του crawler αρχικά δημιουργήσαμε μια συνάρτηση(function) με την οποία «παίρνουμε» όλες τις θέσεις εργασίας μία-μία με βάση το URL τους. Στην συνέχεια ελέγχουμε ένα-ένα τα URL που υπάρχουν στην ιστοσελίδα με αυτά που ήδη υπάρχουν στην βάση δεδομένων μας, στην πρώτη περίπτωση που το URL είναι ήδη καταχωρημένο στην βάση δεδομένων τότε πάει στο επόμενο URL, ενώ σε αντίθετη περίπτωση εάν δηλαδή το URL δεν είναι ήδη καταχωρημένο στην βάση δεδομένων μας τότε ο crawler καλεί μια άλλη συνάρτηση(function) με την οποία «παίρνουμε» το URL και τον HTML κώδικα της αγγελίας και

έπειτα ξανά καλούμε την προηγούμενη συνάρτηση, αυτό επαναλαμβάνεται για όλες τις θέσεις εργασίας μέχρις ότου ο crawler να φτάσει στην τελευταία σελίδα και στην τελευταία αγγελία.

2.3.2 Function remove_html_tags

Η συνάρτηση(function) remove_html_tags χρησιμεύει στο να καθαρίζουμε κατά την διαδικασία του crawling τις αγγελίες από τα html tags δηλαδή καθαρίζουμε την πληροφορία από τις δηλώσεις τις οποίες κάνουμε σε ένα html αρχείο για τον καθορισμό τις μεταβλητής αυτής (π.χ. <li class….<\li>) με στόχο η πληροφορία να μας μένει «καθαρή» από αυτά.

```
def remove_html_tags(text):
    """Remove html tags from a string"""
    p = re.compile(r'<.*?>')
    data = p.sub('', text)
    data = data.strip()
    return data
```

Εικόνα 2—8 Function remove html tags

2.3.3 Function parse

Στην συνάρτηση parse αρχικά ορίζουμε τα μονοπάτια(paths) στα οποία ο crawler θα «πατήσει» ώστε να βρει τις αγγελίες που υπάρχουν στην ιστοσελίδα, στην συνέχεια ελέγχουμε το URL το οποίο βρίσκει ο crawler διότι υπάρχει περίπτωση να μην έχουν την ίδια μορφή, οπότε σε αυτή την περίπτωση θα πρέπει να το «φτιάξουμε» εμείς, εφόσον έχουμε τελειώσει και την διαδικασία ελέγχου του URL θα πρέπει να ελέγχουμε εάν το URL αυτό υπάρχει ήδη στον πίνακα columns ή όχι αρά γι’ αυτό τον λόγο «παίρνουμε» όλες τις αγγελίες που υπάρχουν ήδη στον συγκεκριμένο πίνακα και τις ελέγχουμε με αυτή που θέλουμε να αποθηκεύσουμε, στην περίπτωση που η αγγελία αυτή υπάρχει ήδη στην βάση μας τότε ξανά τρέχει το πρόγραμμα από την αρχή, αλλιώς στην άλλη στην περίπτωση, δηλαδή στην περίπτωση όπου το URL δεν είναι καταχωρημένο στον πίνακα, τότε καλούμε την συνάρτηση parse_job και μέσα από αυτή όταν γίνουν οι απαραίτητης ενέργειες με σκοπό να κρατήσουμε όσα χρειαζόμαστε ξανά καλούμε την συνάρτηση parse, τέλος εφόσον ο crawler έχει φτάσει και στην τελευταία αγγελία της σελίδας αλλάζει την σελίδα (π.χ. εάν είναι στην σελίδα 1 και έχει ελέγξει και τις 16 αγγελίες που υπάρχουν σε αυτή τότε θα αλλάξει και θα πάει στην σελίδα 2) και ξανά ξεκινάει από την αρχή τις διαδικασίες που περιγράψαμε παραπάνω έως ότου φτάσει στην σελίδα που

εμείς του έχουμε ορίσει στην συγκεκριμένη περίπτωση ο crawler θα σταματήσει όταν φτάσει στην σελίδα 94.

```
def parse(self, response):
    # find job path
    all_the_jobs = response.xpath('//article')

    # parse all jobs
    for job in all_the_jobs:
        job_url = job.xpath('.//header/a/@href').extract_first()

        if '/aggelia/' not in job_url:
            job_url = '/aggelia/' + job_url
        job_url_final = 'https://www.careernet.gr' + str(job_url)
        self.conn.cursor(prepared=True)
        print(job_url_final)
        self.cursor.execute("Select job_url from collumns where job_url like '" + str(job_url_final) + "'")
        select_query = self.cursor.fetchall()
        if len(select_query) == 0:
            yield scrapy.Request(job_url_final, callback=self.parse_job)

    # go to next page
    next_page = 'https://www.careernet.gr/aggelies?page=' + str(CarrernetSpider.page_number)
    if CarrernetSpider.page_number <= 94:
        CarrernetSpider.page_number += 1
        yield response.follow(next_page, callback=self.parse)
```

Εικόνα 2—9 Function parse

2.3.4 Function parse_job

Η συνάρτηση parse_job καλείται μόνο στην περίπτωση οπού ο crawler ανιχνεύει με βάση την συνάρτηση που αναφέραμε στο κεφάλαιο 2.6.3 ένα URL το οποίο δεν είναι ήδη καταχωρημένο στον πίνακα columns, πιο αναλυτικά εφόσον έχουν γίνει οι απαραίτητες ενέργειες και οι απαραίτητοι έλεγχοι τα οποία αναφέραμε στο κεφάλαιο 2.6.3 τότε καλείται η συνάρτηση parse_job και μέσω αυτής «εισβάλουμε» στην εκατοστέ αγγελία και με διάφορες συναρτήσεις κρατάμε από την αγγελία τον τίτλος της, το HTML αρχείο και το URL της, στην συνέχεια δημιουργούμε μια σύνδεση μεταξύ του crawler και της βάσης δεδομένων και καταχωρούμε τα στοιχεία που πήραμε από την αγγελία στον πίνακα columns, τέλος εφόσον έχουμε καταχωρήσει όσα χρειαζόμαστε ξανά καλούμε την συνάρτηση parse για να ξανά γίνει η διαδικασία αυτή με την επόμενη αγγελία έως ότου φτάσουμε στην τελευταία σελίδα και στην τελευταία αγγελία.

```
def parse_job(self, response):

    # parse information from job url
    title = response.xpath('//h1[@class="col-lg-11 col-md-11 col-sm-11 col-xs-12 aggelia-title"]').extract()
    job_url = response.url
    job_html = response.body.decode(response.encoding)
    timestamp = time.strftime('%Y-%m-%d %H:%M:%S')
    print(timestamp)
    print('Url:', job_url)
    title = CarrernetSpider.remove_html_tags(title[0])
    print('Title:', title)
    job_html = str(job_html)
    # mysql connection
    try:
        conn = mysql.connector.connect(host="snf-876565.vm.okeanos.grnet.gr",
                                       port="3306",
                                       user="theofanis",
                                       database="testcrawler",
                                       password="9.^9#M<4*k7tN%d,")
        cursor = conn.cursor(prepared=True)
        sql_insert_query = """INSERT INTO
                                            `collumns`(
                                            `job_url`,
                                            `job_description`,
                                            `Date`
                                            )VALUES (%s,%s,%s)"""
        insert_tuple = (job_url, job_html, timestamp)
        result = self.cursor.execute(sql_insert_query, insert_tuple)

        self.conn.commit()
    except mysql.connector.Error as error:
        self.conn.rollback()
        print("Failed to insert into MySQL table{}".format(error))
```

Εικόνα 2—10 Function parse job

3 Εξαγωγή Δεδομένων

Στο κεφάλαιο τρία θα αναπτύξουμε επακριβώς την διαδικασία που ακολουθήσαμε για την εξαγωγή των δεδομένων. Πιο συγκεκριμένα το πρώτο βήμα για την εξαγωγή των δεδομένων από τον πίνακα columns ήταν να δημιουργήσαμε τρείς συναρτήσεις, στην πρώτη συνάρτηση δημιουργήσαμε μια σύνδεση μεταξύ του προγράμματος και του πίνακα options ώστε να δημιουργήσουμε μια μέθοδο deduplication, στην δεύτερη συνάρτηση δημιουργήσαμε μια σύνδεση μεταξύ του προγράμματος και του πίνακα columns με σκοπό να ζητάμε από το πίνακα μια-μια τις ήδη υπάρχουσες εγγραφές με σκοπό να τις ξεσκαρτίσουμε και τέλος σαν τρίτο βήμα και εφόσον έχουν ολοκληρωθεί τα δύο πρώτα, το πρόγραμμα αναζητάει στον HTML κώδικα με βάση τα μονοπάτια που του ορίσαμε τις λέξεις κλειδιά και τις αποθηκεύει στον πίνακα tester με στόχο την ανάλυση τους στο Power Bi.

3.1 Προγραμματισμός του RegEx

Αρχικά για τον προγραμματισμό του regular expression χρειάστηκε να δημιουργήσουμε 3 συναρτήσεις. Πρώτα απ' όλα ελέγχουμε εάν η εκατοστέ αγγελία έχει ξανά αποθηκευτεί στην βάση δεδομένων, στην συνέχεα εάν δεν έχει ξανά αποθηκευτεί τότε «τρέχει» το πρόγραμμα μας, σαν πρώτο στάδιο «τραβάμε» όλο το HTML αρχείο από τον πίνακα columns, στην συνέχεια με διάφορους τύπους ή αλλιώς μονοπάτια π.χ.(re.search(r"(?<=[validThrough :])?\d{4}-\d{2}-\d{2}(?=[T])) «οδηγούμε» το πρόγραμμα μας στα κομμάτια του HTML κώδικα της αγγελίας οπού εμείς θέλουμε να ανιχνεύσουμε με σκοπό να τα αποθηκεύσουμε ώστε να τα αναλύσουμε, τέλος εφόσον κρατήσουμε τις πληροφορίες που θέλουμε από το HTML αρχείο, τις αποθηκεύουμε στον πίνακα tester, ωστόσο για την ολοκλήρωση του Regex χρειάστηκε εμείς οι ίδιοι να δημιουργήσουμε δύο βιβλιοθήκες, στην μία υπάρχει σύνδεση μεταξύ των πινάκων που έχουμε στην βάση δεδομένων μας ενώ στην άλλη πραγματοποιείται ο έλεγχος για διπλότυπες εγγραφές.

3.2 Function id

Με την συνάρτηση id ζητάμε από τον πίνακα options να μας επιστρέψει την τελευταία τιμή της στήλης value. Η τιμή αυτή μας βοηθά ώστε να μην έχουμε διπλότυπες εγγραφές στον πίνακα tester, δηλαδή η συνάρτηση αυτή εφόσον τραβήξει από τον πίνακα options την τιμή της στήλης value την ελέγχει εάν αυτή η τιμή είναι ίδια με την τιμή της στήλης id του πίνακα

columns, σε περίπτωση οπού η τιμή της στήλης value είναι ίση με την τιμή που έχει η στήλη id δηλαδή ισχύει η συνθήκη **value = id** τότε το πρόγραμμα σταματάει εκεί και δεν προχωράει στις παρακάτω ενέργειες, όμως σε αντίθετη περίπτωση ένα δηλαδή η τιμή της στήλης value δεν είναι ίση με την τιμή της στήλης id του πίνακα columns συγκεκριμένα γίνεται μόνο να είναι μικρότερη ή και ίση η τιμή της στήλης value σε σχέση με την τιμή της στήλης id δηλαδή ισχύει η συνθήκη **value < id** τότε το πρόγραμμα προχωράει στις επόμενες συναρτήσεις, έπειτα η τιμή της στήλης value αυξάνεται κατά ένα αυτή η διαδικασία επαναλαμβάνεται έως ότου ισχύσει η συνθήκη **value = id.**

```
#check option id next id to process
def id(self):
    sql_id_deduplication_option = 'SELECT `value` FROM `options` WHERE `id` = 7'
    next_id = self.data_base.query(sql_id_deduplication_option)
    return next_id
```

Εικόνα 3—1 Function id

3.3 Function data_id

Με την συνάρτηση data_id ζητάμε από τον πίνακα columns να μας επιστρέψει την τελευταία τιμή της στήλης id που έχουμε καταχωρήσει με την χρήση του web-crawler, με βάση το id αυτό γίνεται και ο έλεγχος που αναφέραμε και στο κεφάλαιο 3.1.2 έτσι ώστε να γίνει ο έλεγχος για την αποτροπή διπλοτύπων εγγραφών, το id αυτό είναι το κλειδί για κάθε αγγελία στον πίνακα columns και μόνο έτσι το RegEx μπορεί να καταλάβει πώς υπάρχει μια εγγραφή την οποία δεν την έχουμε αποθηκεύσει στον πίνακα tester αλλά ούτε και την έχουμε επεξεργαστεί, έτσι με βάση την συνάρτηση id (κεφάλαιο 3.1.2) και την συνάρτηση data_id γίνονται οι απαραίτητες ενέργειες ώστε να προχωρήσει το πρόγραμμα στα επόμενα βήματα.

```
def data_id(self,id):
    sql_id_deduplication_option = 'SELECT `job_url`,`job_description` FROM `collumns` WHERE `id` = "' + str(id)+'"'
    next_id = self.data_base.query_all(sql_id_deduplication_option)
    return next_id
```

Εικόνα 3—2 Function data id

3.4 Function data_clean

Εφόσον έχουν γίνει τα βήματα που αναφέραμε στα κεφάλαια 3.1.2 και 3.1.3 τότε το πρόγραμμα προχωράει στην τελική συνάρτηση, στην συνάρτηση με όνομα data_clean, σε αυτή την συνάρτηση έχουμε ορίσει όσα χρειαζόμαστε από το regular expression να μας επιστρέψει, δηλαδή να μας επιστρέψει τον τίτλο της αγγελίας, την επωνυμία της επιχείρησης, τις διευθύνσεις καθώς και τις ημερομηνίες δημοσίευσης και διαγραφής, αυτό πραγματοποιείται με κάποιους τύπους βασισμένους σε μια ακολουθία λογικών χαρακτήρων (π.χ. για να ανίχνευση την ημερομηνία διαγράφης της κάθε αγγελίας γίνεται με βάση την συγκεκριμένη ακολουθία (r"(?<=[ValidThrough :]) ?\d{4}-\d{2}-\d{2}(?=[T] εικόνα 3-3) οπού σημαίνει επέστρεψε μου όσα αναγράφονται μεταξύ των λέξεων ValidThrough και του γράμματος T και έχει την μορφή 4-2-2 δηλαδή Χρονολογία, Μήνας, Ημέρα) τέλος εφόσον τα εντοπίσει όλα όσα του ζητάμε προχωράει παρακάτω και φτάνει στο στάδιο της αποθηκεύσεις, εκεί έχουμε δημιουργήσει μια σύνδεση μεταξύ του προγράμματος και του πίνακα tester με σκοπό να «περνάει» όσα βρίσκει με την σειρά που τα θέλουμε, όμως δεν γίνεται να τα περάσουμε με ανακατωμένη σειρά καθώς η σειρά που αναγράφεται στην εικόνα 3-3 είναι και η σειρά με την οποία είναι δομημένες οι στήλες στον πίνακα tester άρα δεν θα ήταν σωστό να έχουμε πρώτα το πεδίο company_name ενώ στον πίνακα πρώτα αναγράφεται ο τιτλός γίατι θα περνούσαν λάθος τα δεδομένα (π.χ. στο πεδίο title θα αποθήκευε RAND STAND HELLAS A.E ενώ το σωστό θα ήταν Βοηθός Λογιστικής).

```
def data_clean(self,data):

    print(data)
    title = re.search(r"title\" : \"(.*?)\"", str(data[1])).group()
    description = re.search(r"description\" : \"(.*?)\"", str(data[1])).group()
    employ_type = re.search(r"employmentType\" : \"(.*?)\",  str(data[1])).group()
    company_name = re.search(r"name\" : \"(.*?)\",  str(data[1])).group()
    address_country = re.search(r"addressCountry\": \"(.*?)\",  str(data[1])).group()
    address_region = re.search(r"addressRegion\": \"(.*?)\", str(data[1])).group()
    address_locality = re.search(r"addressLocality\": \"(.*?)\",  str(data[1])).group()
    address_street = re.search(r"streetAddress\": \"(.*?)\", str(data[1])).group()
    date_posted = re.search(r"\d{4}-\d{2}-\d{2}", str(data[1])).group()
    date_end = re.search(r"(?<=[validThrough : ])?\d{4}-\d{2}-\d{2}(?=[T])", str(data[1])).group()

    print(title[10:len(title)-1])
    print(description[16:len(description)-1])
    print(employ_type[19:len(employ_type)-1])
    print(company_name[9:len(company_name)-1])
    print(address_country[18:len(address_country)-1])
    print(address_region[17:len(address_region)-1])
    print(address_locality[21:len(address_locality)-1])
    print(address_street[17:len(address_street)-1])
    print(date_posted)
    print(date_end,id)

    try:

        conn = mysql.connector.connect(host="snf-876565.vm.okeanos.grnet.gr",
                                       port="3306",
                                       user="theofanis",
                                       database="testcrawler",
                                       password="9.^9#M<4*k7tN%d,")
        cursor = conn.cursor(prepared=True)
        sql_insert_query = """INSERT INTO
                                            `tester`(
                                              `title`,
                                              `description`,
                                              `employ_type`,
                                              `company_name`,
                                              `address_country`,
```

Εικόνα 3—3 Function data clean

4 Ανάλυση Δεδομένων

Για την ανάλυση των δεδομένων χρησιμοποιήσαμε το πρόγραμμα της Microsoft το **Power Bi.** Το **Power BI** είναι μια συλλογή από υπηρεσίες λογισμικού, εφαρμογές και συνδέσεις που συνεργάζονται για να μετατρέψουν τις μη σχετιζόμενες προελεύσεις δεδομένων σας σε συνεκτικές, οπτικά καθηλωτικές και διαδραστικές πληροφορίες. Τα δεδομένα σας μπορεί να είναι ένα υπολογιστικό φύλλο Excel ή μια συλλογή υβριδικών αποθηκών δεδομένων cloud ή εσωτερικής εγκατάστασης. Το Power BI σάς επιτρέπει να συνδέσετε εύκολα τις προελεύσεις δεδομένων σας, να απεικονίσετε και να ανακαλύψετε τα σημαντικά στοιχεία και να τα μοιραστείτε με όσα άτομα θέλετε. Επίσης το Power BI είναι μια υπηρεσία επιχειρηματικών αναλυτικών στοιχείων της Microsoft. Παρέχει διαδραστικές απεικονίσεις και δυνατότητες επιχειρηματικής ευφυΐας με μια διεπαφή που η Microsoft λέει ότι είναι αρκετά απλή για τους τελικούς χρήστες να δημιουργούν αναφορές και πίνακες ελέγχου.

4.1 Η ιστορία του Power Bi

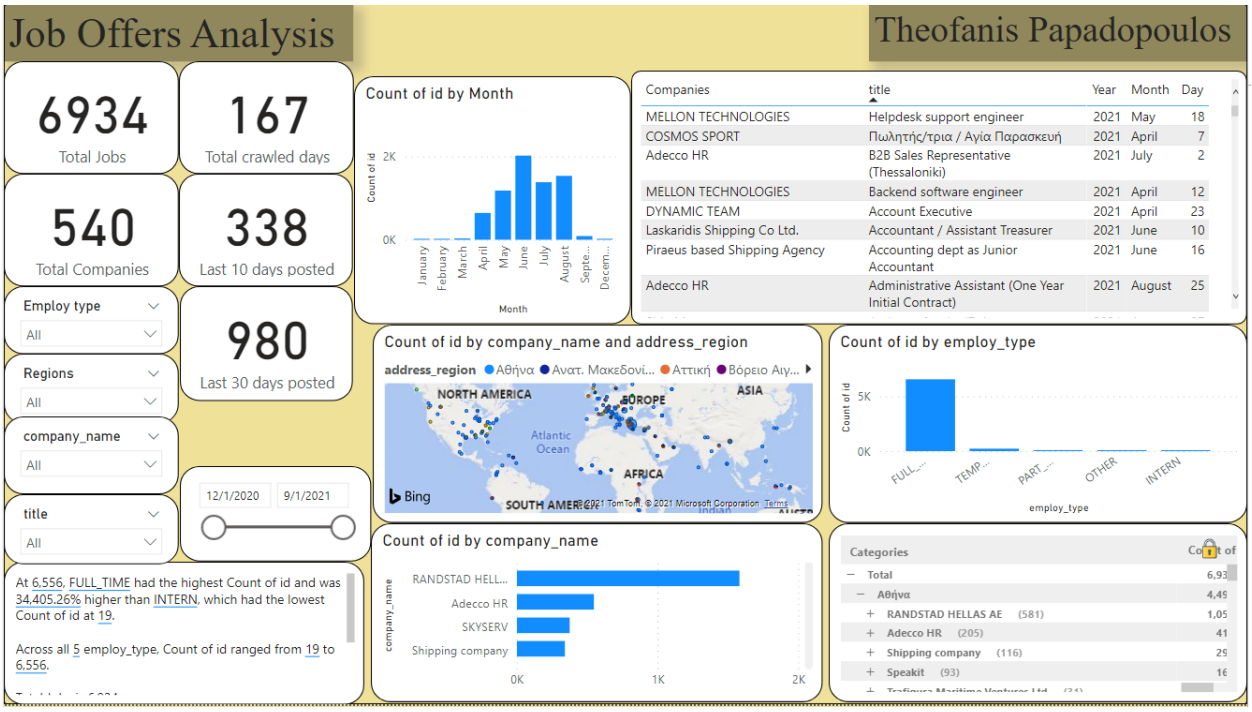
Αυτή η εφαρμογή σχεδιάστηκε αρχικά από τους Thierry D'Hers και Amir Netz της ομάδας αναφοράς υπηρεσιών SQL Server της Microsoft. Αρχικά σχεδιάστηκε από τον Ron George το καλοκαίρι του 2010 και ονομάστηκε Project Crescent. Το Project Crescent ήταν αρχικά διαθέσιμο για δημόσια λήψη στις 11 Ιουλίου 2011 σε συνδυασμό με τον κωδικό όνομα SQL Server Denali. Αργότερα μετονομάστηκε σε Power BI και στη συνέχεια παρουσιάστηκε από τη Microsoft τον Σεπτέμβριο του 2013 ως Power BI για το Office 365. Η πρώτη κυκλοφορία του Power BI βασίστηκε στα πρόσθετα που βασίζονται στο Microsoft Excel: Power Query, Power Pivot και Power View. Με την πάροδο του χρόνου, η Microsoft πρόσθεσε επίσης πολλές πρόσθετες δυνατότητες, όπως ερωτήσεις και απαντήσεις, δυνατότητα σύνδεσης δεδομένων επιχειρήσεων και επιλογές ασφάλειας μέσω του Power BI Gateways. Το Power BI κυκλοφόρησε για πρώτη φορά στο ευρύ κοινό στις 24 Ιουλίου 2015.

Τον Φεβρουάριο του 2019, η Gartner.com, μια εταιρεία αναθεώρησης λογισμικού, επιβεβαίωσε τη Microsoft ως ηγέτη στο "2019 Gartner Magic Quadrant for Analytics and Business Intelligence Platform" ως αποτέλεσμα των δυνατοτήτων της πλατφόρμας Power BI. Αυτό αντιπροσώπευε το 12ο συνεχόμενο έτος αναγνώρισης της Microsoft ως κορυφαίου προμηθευτή σε αυτήν την κατηγορία Magic Quadrant (ξεκινώντας 3 χρόνια πριν καν δημιουργηθεί αυτό το εργαλείο).

4.2 Γενική ανάλυση

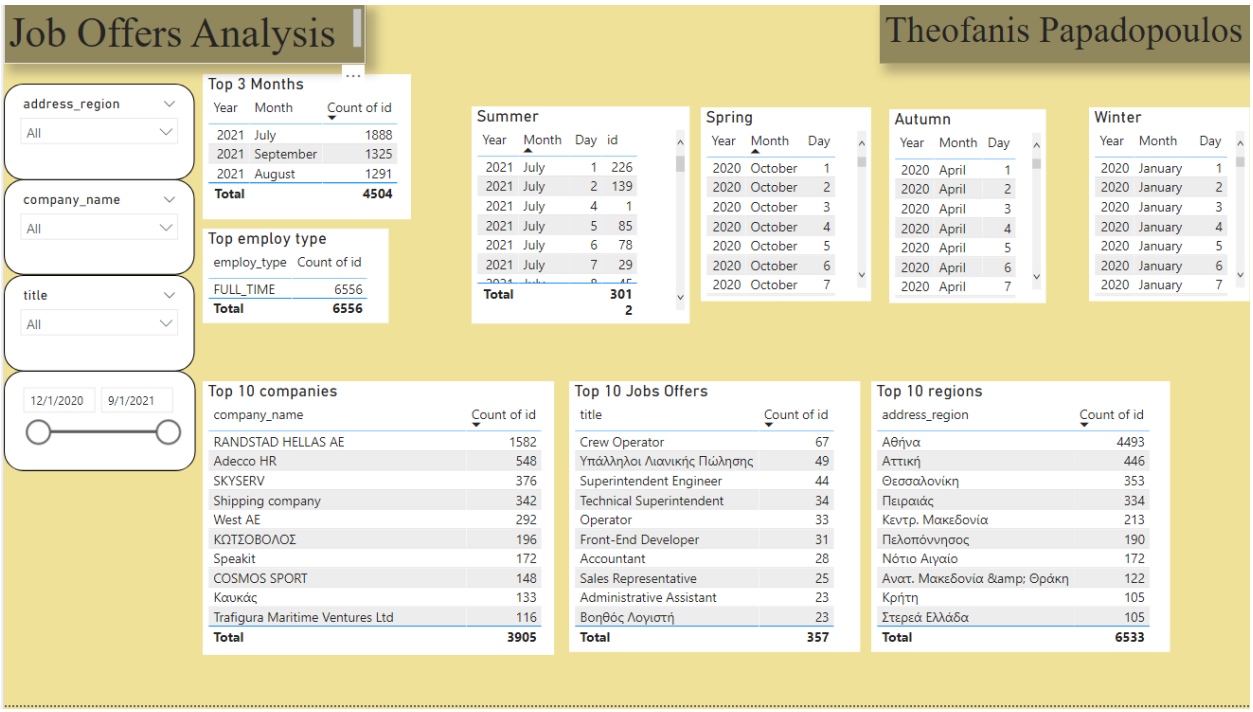
Αρχικά θα σας παρουσιάσουμε τα γενικά αποτελέσματα τα οποία πήραμε από την ανάλυση που κάναμε, δηλαδή θα σας παρουσιάσουμε πόσες και ποιες εταιρίες έχουμε «κρατήσει», ποιες ημέρες και τα συνολικά αποτελέσματα.

Στην εικόνα 4-1 παρατηρούμε πως οι συνολικές θέσεις που κρατήσαμε ανέρχονται στις 6.934, παρατηρούμε πως συνολικά βρήκαμε 540 εταιρίες, οι συνολικές ημέρες που κάναμε crawl είναι 167, τις τελευταίες 10 ημέρες δημοσιευθήκαν 403 αγγελίες και πως στις τελευταίες 30 ημέρες δημοσιεύθηκαν 1020, επιπλέον με την χρήση διαφόρων πινάκων καθώς και σχεδιαγραμμάτων παρατηρούμε πως τον μήνα Ιούνιο είχαμε τις περισσότερες αγγελίες, στην συνέχεια με την χρήση του πίνακα βλέπουμε ποιες εταίρες δημοσίευσαν αγγελίες, τι θέση εργασίας ανοίχτηκε και πότε, τέλος με την χρήση των διαγραμμάτων βλέπουμε τι τύπος εργασίας ζητείτε ποιο συχνά δηλαδή full time, part time κτλ. και στο τελευταίο σχήμα βλέπουμε πως η RANDSTAD HELLAS A.E έχει δημοσίευση τις πιο πολλές αγγελίες μεταξύ των διαστημάτων από 12/1/2020 έως 9/1/2021.



Εικόνα 4—1 General Page

Στην εικόνα 4-2 παρουσιάζονται οι 10 πρώτες εταιρίες με βάση τον αριθμό των αγγελιών που δημοσίευσαν, οι 10 πρώτες κατηγορίες εργασίας και οι 10 πρώτες περιοχές με βάση τον αριθμό των αγγελιών. Με βάση τους πίνακες καταλήξαμε στο συμπέρασμα πώς από τις 12/1/2020 έως τις 9/1/2021 η εταιρία RANDSTAND HELLAS A.E είχε δημοσιεύσει τις περισσότερες αγγελίες οι οποίες ανέρχονται στον αριθμό των 1582 αγγελιών, η αγγελία με το όνομα Crew Operator κατέχει την πρώτη θέση στα ίδια χρονικά διαστήματα καθώς μετράει 67 αγγελίες με αυτόν τον τίτλο και η Αθήνα κατέχει την πρώτη θέση καθώς δημοσιεύθηκαν 4493 αγγελίες οι οποίες αφορούν την Αθήνα έπειτα δεύτερη έρχεται η Αττική με 446. Τέλος οι 3 μήνες οι οποίοι είχαν την μεγαλύτερη ζήτηση ατόμων για εργασία είναι, πρώτος έρχεται ο μήνας Ιούλιος του 2021 με 1888 αγγελίες, δεύτερος έρχεται ο μήνας Σεπτέμβριος του 2021 και τρίτος ο Αύγουστος του 2021, εδώ επίσης διαπιστώσαμε πως την καλοκαιρινή εποχή είχαμε την μεγαλύτερη ζήτηση ατόμων για εργασία.

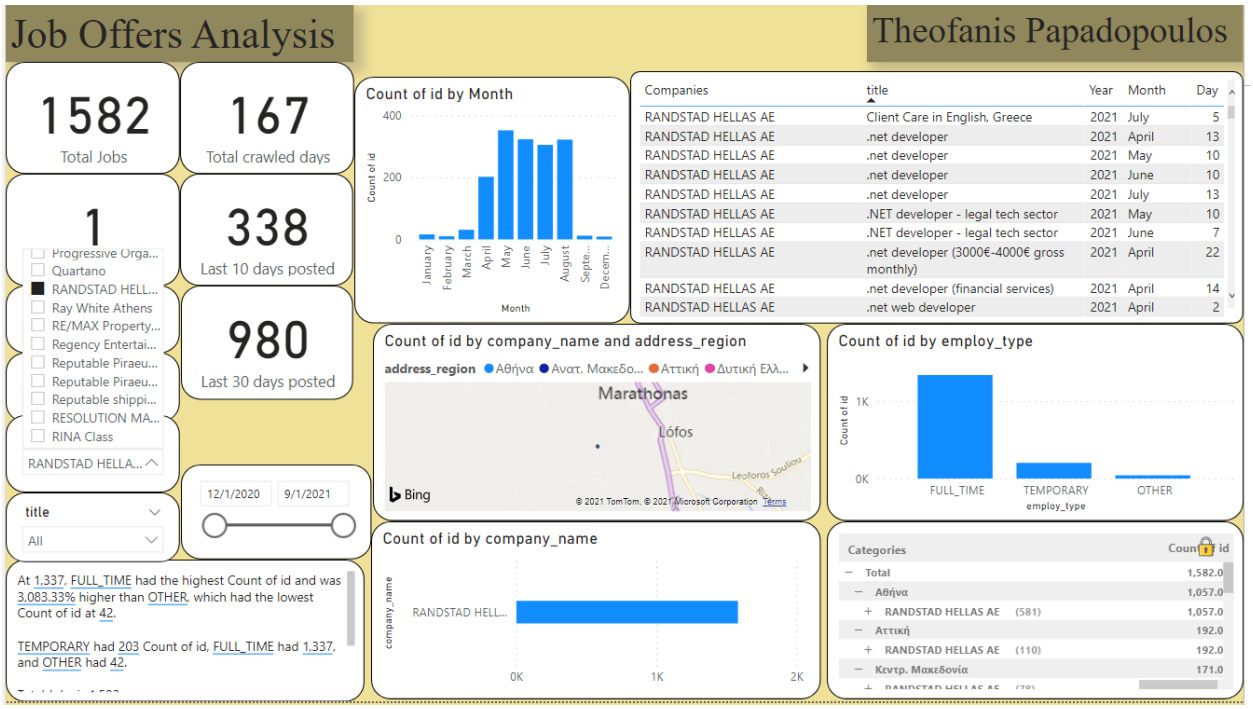


Εικόνα 4—2 Trending Page

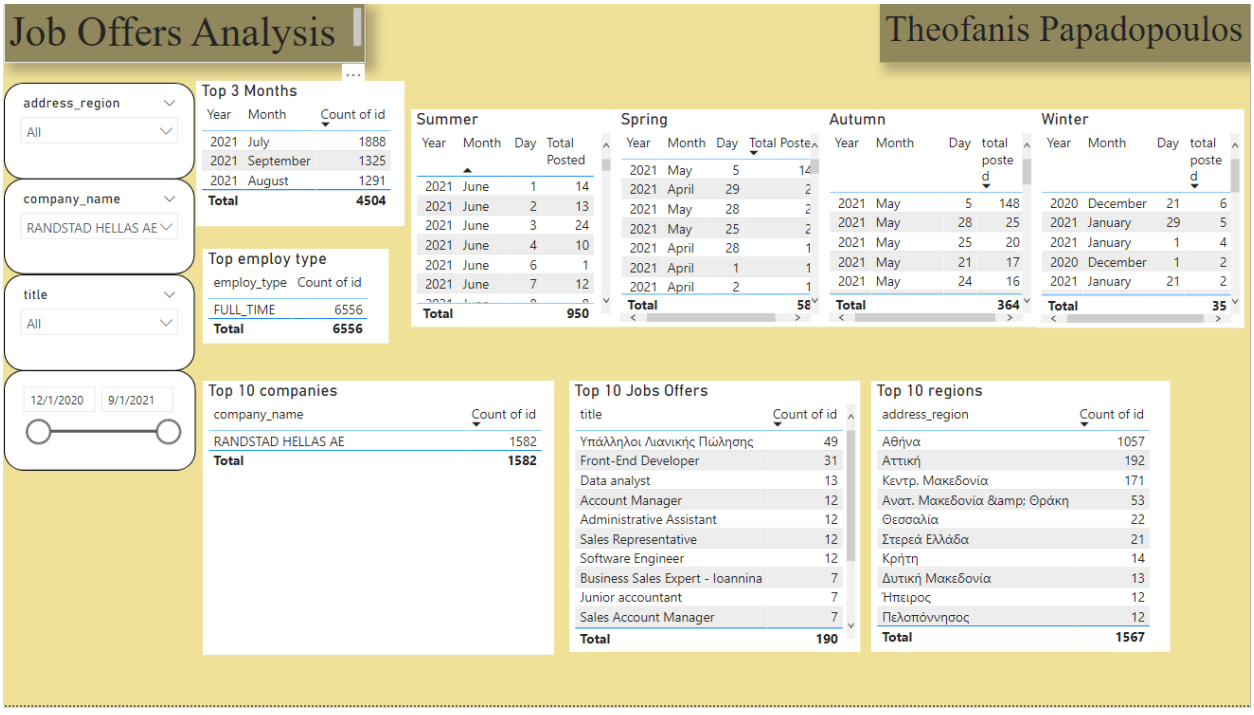
4.3 Αποτελέσματα

Όπως θα δούμε και στα σχήματα παρακάτω αλλάζοντας κάποιες μεταβλητές αναζήτησης όπως οι ημερομηνίες για τις οποίες ενδιαφερόμαστε ή ψάχνοντας μια συγκεκριμένη εταιρία, μια συγκεκριμένη πόλη/περιοχή, έναν συγκεκριμένο τίτλο εργασίας ή ακόμα και έναν συγκεκριμένο τύπο εργασίας τότε αλλάζουν και τα αποτελέσματα τα οποία έχουμε.

1) Έστω πως μας ενδιαφέρει να μάθουμε σχετικά για την εταιρία με το όνομα RANDSTAD HELLAS A.E.
   a) Με την βοήθεια της εικόνας 4-3 παρατηρούμε πως η συγκεκριμένη εταιρία έχει δημοσίευσή 1582 αγγελίες, με την βοήθεια του διαγράμματος διαπιστώνουμε πώς τον μήνα Μάιο είχε δημοσιεύσει τις περισσότερες αγγελίες και πως ο τύπος εργασίας Full time ανέρχεται στον αριθμό των 1.337 αγγελιών.
   b) Στην εικόνα 4-4 παρατηρούμε πώς για την συγκεκριμένη εταιρία πως οι περισσότερες αγγελίες που δημοσίευσε για εργασία αφορούσαν Υπάλληλους Λιανικής Πώλησης οι οποίες ανέρχονται στις 49, η Αθήνα για την συγκεκριμένη εταιρία βρίσκεται στην πρώτή θέση καθώς δημοσίευσε 1057 αγγελίες που αφορούσαν την συγκεκριμένη περιοχή.



Εικόνα 4—3 RANDSTAD HELLAS AE, General

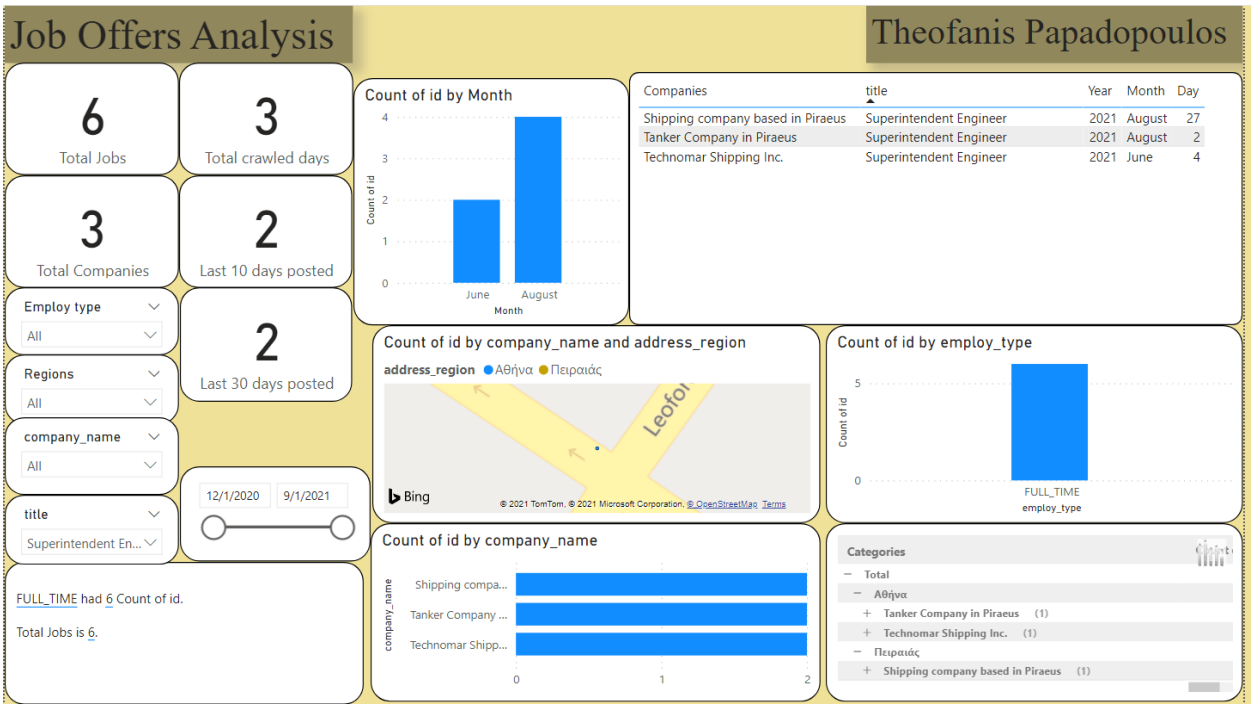


Εικόνα 4—4 RANDSTAD HELLAS AE, Trending

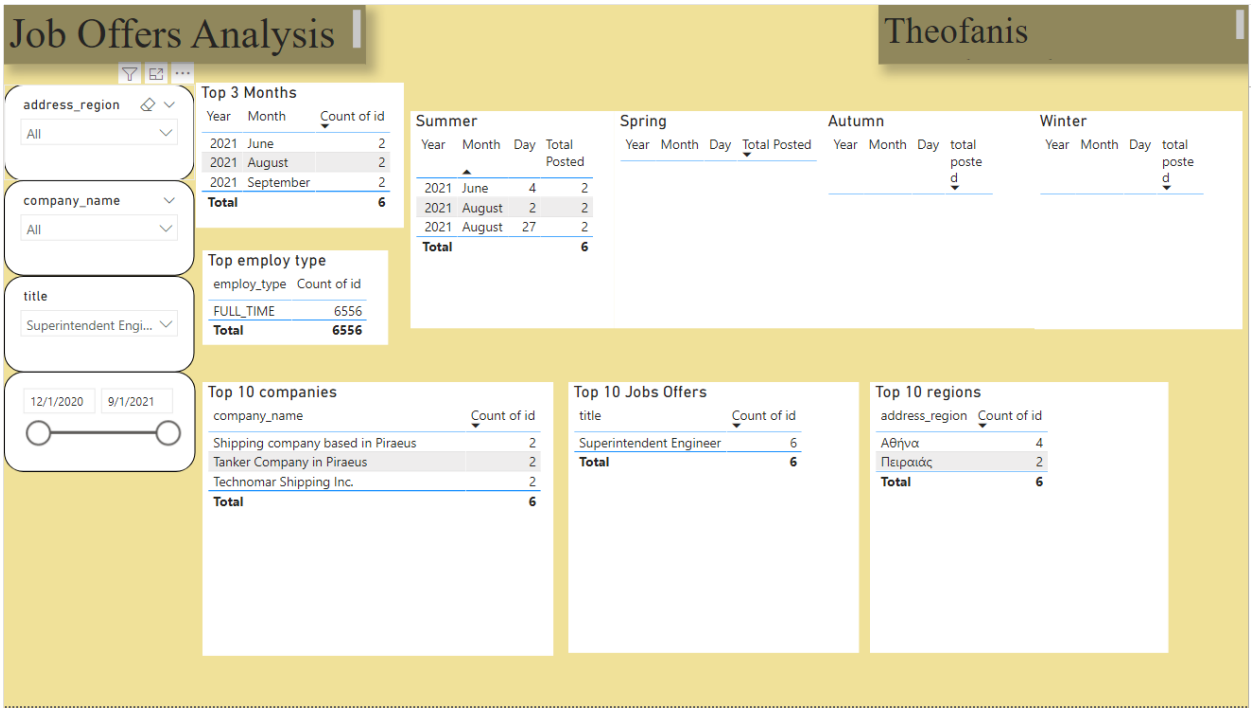
2) Έστω πως ενδιαφερόμαστε για την εργασία με τίτλο Superintendent Engineer.

a) Αρχικά με την χρήση της εικόνας 4-5 διαπιστώνουμε πώς οι συνολικές θέσεις εργασίας που αφορούν τον συγκεκριμένο κλάδο εργασίας ανέρχονται στις 6, επίσης διαπιστώνουμε πώς τις τελευταίες 30 ημέρες βρέθηκαν 2 αγγελίες που αφορούσαν τον συγκεκριμένο κλάδο εξίσου και τις τελευταίες 10 ημέρες είχαμε τον ίδιο αριθμό ζήτησης, οι συνολικές εταιρίες που δημοσίευσαν αγγελίες με τον τίτλο Superintendent Engineer ανέρχονται στις 3 ενώ και οι 6 αγγελίες που δημοσιευθήκαν ήταν πλήρης απασχόλησης, τέλος οι συγκεκριμένη θέση εργασίας ζητήθηκε μόνο τους μήνες Ιούνιο και Αύγουστο.

b) Στην συνέχεια με την χρήση της εικόνας 4-6 διαπιστώνουμε πώς οι περιοχές που ζητούν τον συγκεκριμένο κλάδο εργασίας είναι η Αθήνα και ο Πειραιάς, με πρώτη την Αθήνα η οποία είχε τις 4 αγγελίες από τις 6, επίσης οι ακριβής ημερομηνίες οι οποίες δημοσιεύθηκαν οι αγγελίες είναι : Α) 4 Ιουλίου 2021 με 2 αγγελίες, Β) 2 Αυγούστου 2021 με 2 αγγελίες και Γ) 27 Αυγούστου 2021 με 2 αγγελίες ξανά.



Εικόνα 4—5 Superintendent Engineer, General



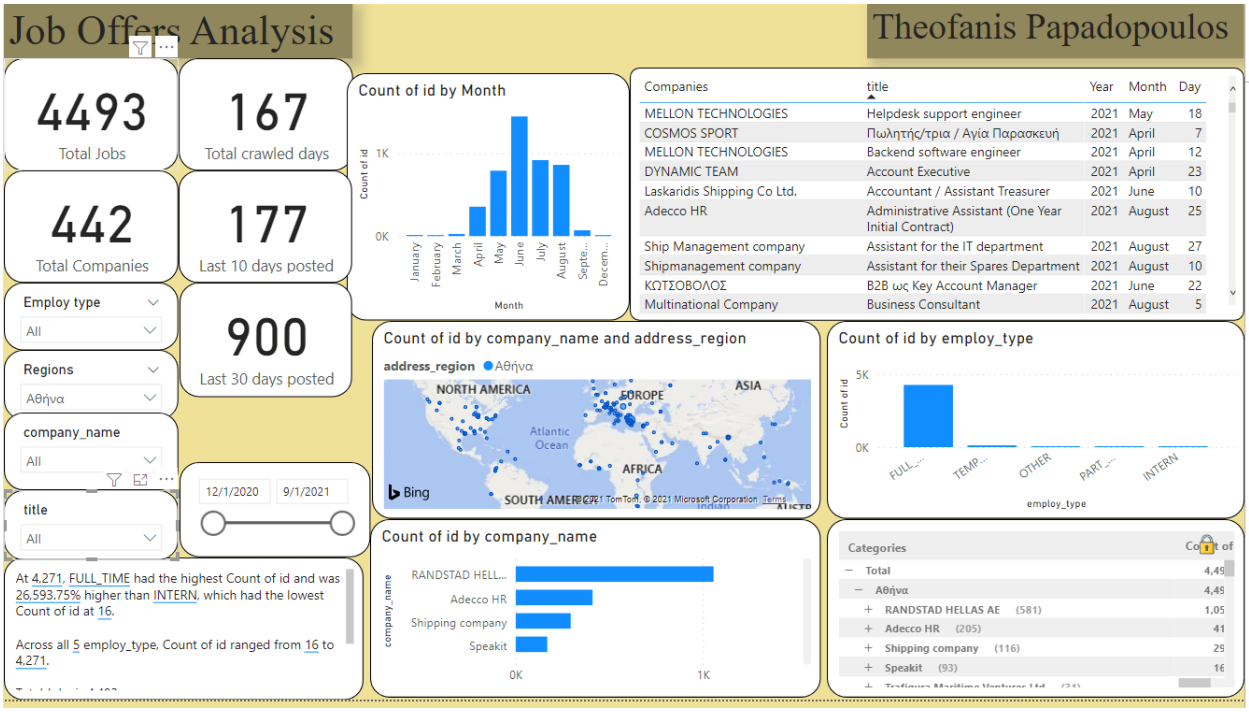
Εικόνα 4—6 Superintendent Engineer, Trending

3) Έστω πως θέλουμε να μας εμφανίσει όλες τις αγγελίες εργασίας που αφορούν την Αθήνα.

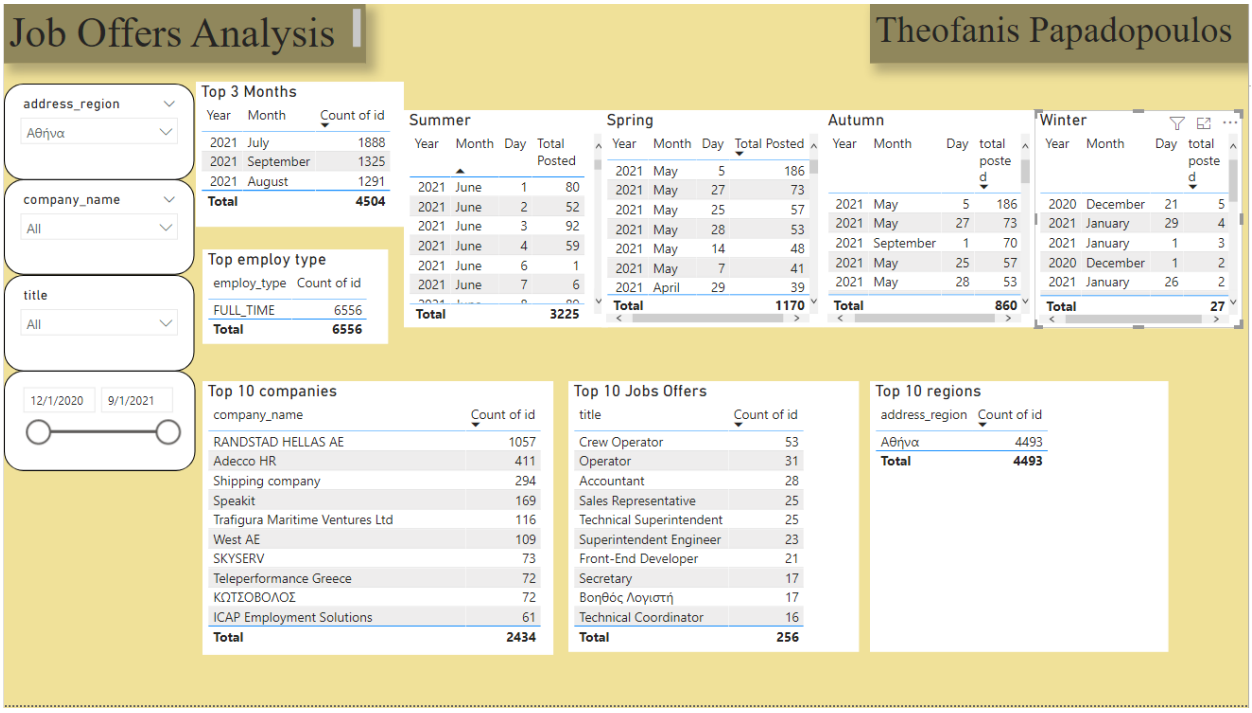
a) Πρώτα απ' όλα μέσω της εικόνας 4-7 διαπιστώνουμε πώς οι συνολικές αγγελίες που αφορούν την Αθήνα ανέρχονται στις 4493, τις τελευταίες 30 ημέρες είχαμε 900 αγγελίες που δημοσιευθήκαν ενώ τις τελευταίες 10 είχαμε 177, τον μήνα Ιούνιο είχαμε τις περισσότερες αγγελίες που δημοσιεύθηκαν για την Αθήνα επίσης η εταιρεία RANDSTAD HELLAS A.E κατέχει την πρώτη θέση με βάση το σχεδιάγραμμα Count of id by company_name.

b) Στην συνέχεια μέσω της εικόνας 4-8 διαπιστώνουμε πώς στις οι 10 εταιρείες οι οποίες δημοσίευσαν αγγελίες που αφορούσαν την Αθήνα είναι η RANDSTAND HELLAS A.E με 1057 αγγελίες, Adecco HR με 411 αγγελίες, Shipping company με 294 αγγελίες, Speakit με 169 αγγελίες, Trafigura Maritime Ventures Ltd με 116 αγγελίες, West A.E με 109 αγγελίες, skyserv με 73 αγγελίες, Teleperformance Greece με 72 αγγελίες. ΚΩΤΣΟΒΟΛΟΣ με 72 αγγελίες και τέλος η εταιρεία ICAP Employment Solutions με 61 αγγελίες επίσης οι 10 θέσεις εργασίας με την μεγαλύτερη ζήτηση εργασίας για την Αθήνα είναι η θέση με τίτλο Crew Operator με 53 αγγελίες, Operator με 31 αγγελίες,

Accountant με 28 αγγελίες, Sales Representative με 25 αγγελίες, Technical Superintendent με 25 αγγελίες, Superintendent Engineer με 23 αγγελίες, Front – End Developer με 21 αγγελίες, Secretary με 17 αγγελίες, Βοηθός Λογιστή με 17 αγγελίες και τέλος Technical Coordinator με 16 αγγελίες, επίσης ένα συγκρίνουμε και τις εποχές θα διαπιστώσουμε πώς την καλοκαιρινή περίοδο είχαμε την μεγαλύτερη ζήτηση ατόμων για εργασία στην Αθήνα καθώς το καλοκαίρι δημοσιεύθηκαν 3.225 αγγελίες ενώ στην δεύτερη θέση βρίσκεται η άνοιξη με 1170 αγγελίες, στην συνέχεια έρχεται το φθινόπωρο με 860 αγγελίες και τέλος ο χειμώνας με μόλις 27 αγγελίες.



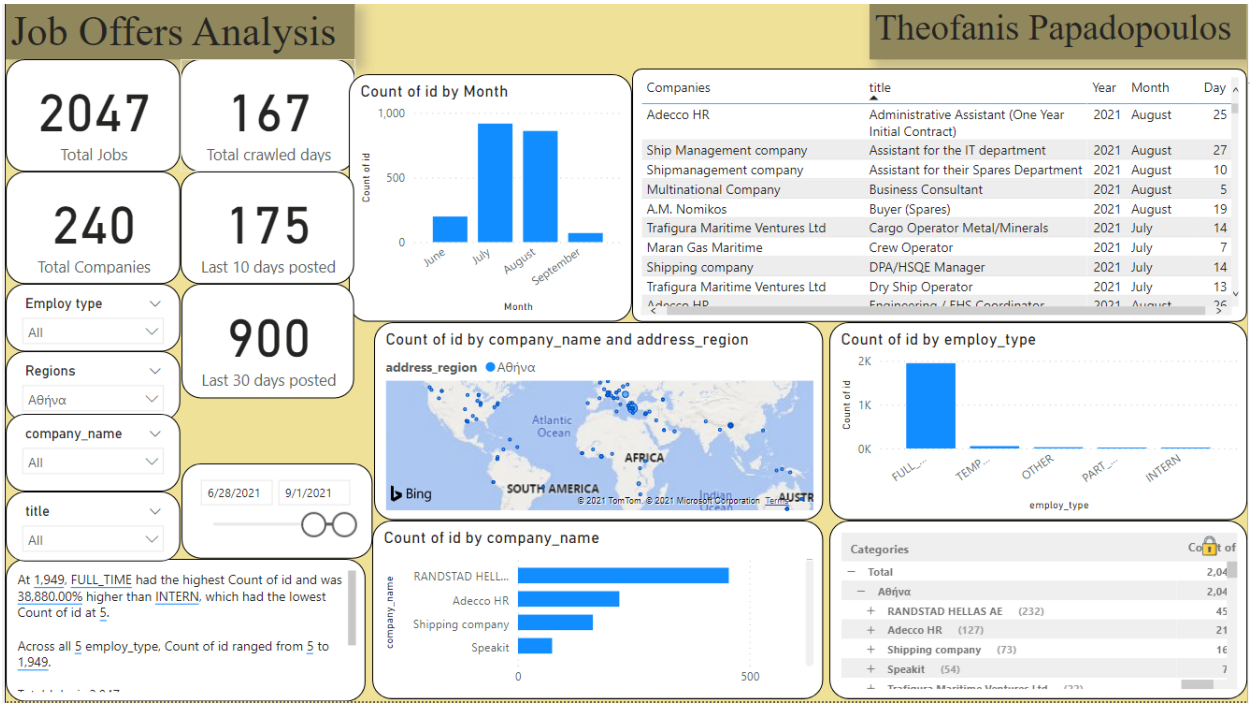
Εικόνα 4—7 Αθήνα, General



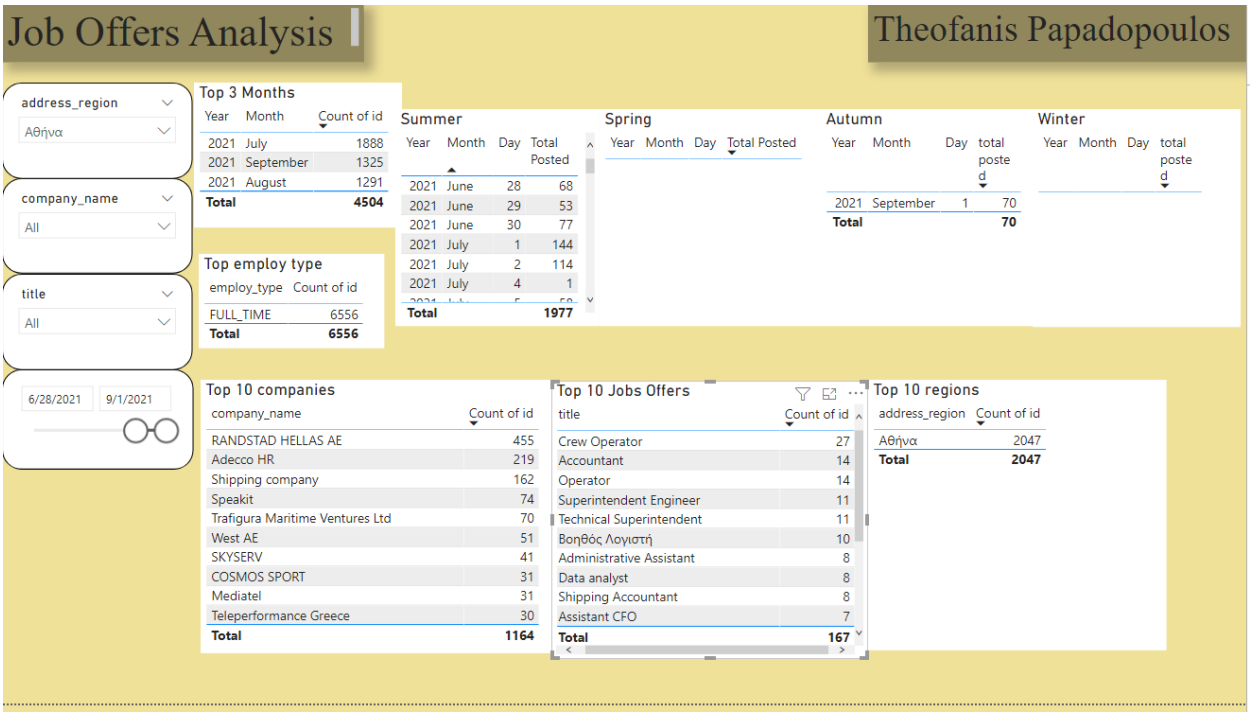
Εικόνα 4—8 Αθήνα, Trending

4) Έστω πως ενδιαφερόμαστε για συγκεκριμένα χρονικά όρια για παράδειγμα μας ενδιαφέρει να βρούμε τις εργασίες που δημοσιευθήκαν από τις 28/6/2021 έως 1/9/2021.
   a) Αρχικά με την εξέταση της εικόνας 4-9 διαπιστώνουμε πως οι συνολικές αγγελίες που δημοσιεύθηκαν το χρονικό διάστημα αυτό ανέρχονται στις 2.047, οι συνολικές εταιρίες οι οποίες δημοσίευσαν αγγελίες σε αυτό το χρονικό διάστημα είναι 240 ενώ από τις 2.047 αγγελίες οι 1.949 αφορούσαν θέσης πλήρης εργασίας.
   b) Επιπροσθέτως εξετάζοντας και την εικόνα 4-10 διαπιστώνουμε πως οι 10 εταιρίες με τις περισσότερες δημοσιευμένες αγγελίες το συγκεκριμένο χρονικό διάστημα είναι η RANDSTAND HELLAS A.E με 455 αγγελίες, δεύτερη έρχεται η Adecco HR με 219 αγγελίες και Τρίτη η Shipping company με 162, στην συνέχεια έρχεται η εταιρεία Speakit με 74 αγγελίες, η Trafigura Maritime Ventures Ltd με 70 αγγελίες, West A.E με 51 αγγελίες, SKYSERV με 41 αγγελίες, COSMOS SPORT με 31 αγγελίες, Mediatel με 31 αγγελίες και τέλος η Teleperformance Greece με 30 αγγελίες. Οι 10 κλάδοι εργασίας που ζητήθηκαν πιο πολύ την συγκεκριμένη χρονική στιγμή ήταν η Crew operator με 27 αγγελίες, Accountant με 14 αγγελίες, Operator με 14 αγγελίες, Superintendent Engineer με 11

αγγελίες, Technical Superintendent με 11 αγγελίες, Βοηθός λογιστή με 10 αγγελίες, Administrative Assistant με 8 αγγελίες, Data analyst με 8 αγγελίες, Shipping Accountant με 8 αγγελίες και τέλος Assistant CFO με 8 αγγελίες ενώ όλες οι αγγελίες που δημοσιευθήκαν αφορούσαν την Αθήνα.



Εικόνα 4—9 28/6/2021-1/9/2021, General



Εικόνα 4—10 28/6/2021-1/9/2021, Trending

Επίλογος

Η διπλωματική αυτή με βοήθησε τόσο στο να μάθω καινούριες τεχνολογίες όσο και στο να κατανοήσω σε μεγαλύτερο βάθος τεχνολογίες που ήδη γνώριζα, πιο συγκεκριμένα οι τεχνολογίες που χρησιμοποιήθηκαν ήταν η Python, Xampp, RegEx, και MySQL. Με την Python προγραμματίσαμε τον crawler καθώς και το RegEx, με την χρήση του Xampp δημιουργήσαμε έναν τοπικό διακομιστή(local server) στον ηλεκτρονικό υπολογιστή μας και τέλος με την χρήση της MySQL δημιουργήσαμε την βάση δεδομένων στην οποία αποθηκεύσαμε τα αποτελέσματα που μας επέστρεφε ο crawler καθώς και το RegEx. Τέλος ασχοληθήκαμε με το PowerBi στο οποίο έγινε και η ανάλυση των δεδομένων με την χρήση διαφόρων πινάκων και πινάκων με στόχο την ανάλυση όσων αποτελεσμάτων μας είχε επιστρέψει το RegEx.

5 Βιβλιογραφία

1 https://www.datasciencecentral.com/profiles/blogs/history-of-mysql

2 https://en.wikipedia.org/wiki/MySQL

3 https://www.seobility.net/en/wiki/RegEx

4 https://en.wikipedia.org/wiki/Python_(programming_language)

5 https://en.wikipedia.org/wiki/XAMPP

6 https://en.wikipedia.org/wiki/Web_crawler

7 https://en.wikipedia.org/wiki/Database