

# Mrežna mjesta unutar hr domene kroz prizmu relacijske baze podataka

---

**Kroflin, Sven**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:142:662212>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-06**



*Repository / Repozitorij:*

[FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek](#)



Sveučilište J. J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Sveučilišni prijediplomski jednopredmetni studij Informatologija

Sven Kroflin

**Mrežna mjesta unutar hr domene kroz prizmu relacijske baze podataka**

Završni rad

Mentor: izv. prof. dr. sc. Tomislav Jakopec

Osijek, 2024.

Sveučilište J. J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Sveučilišni prijediplomski jednopredmetni studij Informatologija

Sven Kroflin

**Mrežna mjesta unutar hr domene kroz prizmu relacijske baze podataka**

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti, informacijski sustavi  
i informatologija

Mentor: izv. prof. dr. sc. Tomislav Jakopec

Osijek, 2024

**Prilog: Izjava o akademskoj čestitosti i o suglasnosti za javno objavljivanje**

Obveza je studenta da donju Izjavu vlastoručno potpiše i umetne kao treću stranicu završnoga, odnosno diplomskog rada.

**IZJAVA**

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napisao/napisala te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s navođenjem izvora odakle su preneseni.

Svojim vlastoručnim potpisom potvrđujem da sam suglasan/suglasna da Filozofski fakultet u Osijeku trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta u Osijeku, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku 23. kolovoza, 2024.

Sven Kroflin, 0122238819

Ime i prezime studenta, JMBAG



## Sažetak

Ovaj završni rad pruža pregled oblikovanja mrežnih stranica, kako bi se istaknuo razvoj i napredak tehnologije primijenjeno u istome. Predstavlja se razvoj mrežnih stranica i mjesta kroz vizualni dojam korisnika primjenom metoda i načina u raspoređivanju i baratanju sadržajem. Kako se razvijaju mrežna mjesta, razvijaju se alati namijenjeni pohrani podataka samih mrežnih mjesta i korisnika koji prebivaju na njima, u obliku baze podataka. Baze podataka predstavljaju skup svih povezanih podataka, a budući da je riječ o ogromnoj količini podataka koji su nepregledni i iznimno nemogući procesuirati u očima programera kamo li korisnika, razvijaju se posebni sustavi namijenjeni upravljanju istoga, odnosno sustavi za upravljanje bazama podataka (SUBP). Postoje mnoštvo SUBP-a, svaka uz svoje pogodnosti i nepogodnosti. Uz procese globalizacije i rasprostranjivanjem medija diljem svijeta, ista se tehnologija prožima na područje Republike Hrvatske, te dolazi do razvitka hrvatskih mrežnih mjesta. Primjenom metoda rudarenja podataka, analize podataka, MariaDB i alata DBeaver, navedeni će rad analizirati bazu podataka koja sadržava mnoštvo podataka o aktualnim hrvatskim mrežnim mjestima, koristeći strukturirani upitni jezik sa specifičnim uvjetima kako bi konačno dohvaćeni podaci odgovarali potrebama rada i u konačnici doveo zaključak o hrvatskim mrežnim mjestima.

**Ključne riječi:** mrežno mjesto, mrežna stranica, baza podataka, relacijska baza podataka, domena

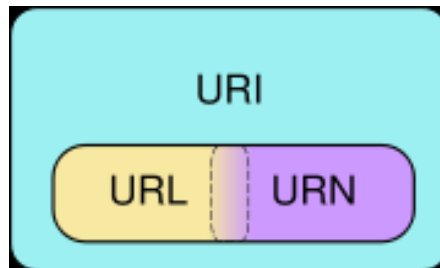
## Sadržaj

1. Uvod.....	1
2. Povijesni pregled mrežnih stranica .....	3
2.1. Implementacija CSS-a .....	3
3. Baze podataka .....	5
3.1. Vrste SUBP-a .....	6
4. Relacijski model.....	9
4.1. Primarni i vanjski ključ.....	10
5. ERA dijagram.....	11
6. Rad s bazom.....	12
6.1. Osnovni preduvjeti za rad .....	12
6.2. Podešavanje okruženja za rad .....	13
6.3. ERA dijagram i tablice baze podataka .....	15
6.4. Rudarenje i analiza podataka .....	20
6.4.1. Upiti .....	21
7. Rasprava.....	29
8. Zaključak.....	31
Literatura.....	32

## 1. Uvod

Tijekom svog rada u Europskom vijeću za nuklearna istraživanja, Tim Berners Lee je predložio način za učinkovitu diseminaciju i dijeljenje informacije koristeći tehnologiju poznatu kao hipertekst. 1989. Tim Berners Lee predlaže prvi koncept, koji postavlja temelje za prvobitnu inačicu web-a putem dokumenta poznat kao „Information Management: A Proposal“.<sup>1</sup>

Pri razvoju samog web-a, ističu se tri ključne tehnologije: HTML, URI i HTTP protokol.<sup>2</sup> HyperText Markup Language (HTML) je oblik označiteljskog jezika koji se smatra najosnovnijim oblikom bloka namijenjen u izradi mrežnih stranica; on definira značenje i strukturu web sadržaja.<sup>3</sup> Koncept hiperteksta se referira na poveznice koja spaja mrežnu stranicu sa drugom i potom kova pojam mrežnog mjesta. URI ili Uniform Resource Identifier je znakovni niz koji se odnosi na određen resurs; najčešći oblici su URL (Uniform Resource Locator) koji identificira izvor pružanjem sjedišta izvora na samome web-u i URN (Uniform Resource Name) koji se odnosi na resurs putem nazivlja ovisno o imenskom prostoru (Slika 1).<sup>4</sup>



Slika 1: Odnos URI-a s URL-om i URN-om.

Hypertext Transfer Protocol (HTTP) je protokol aplikacijskog sloja za prijenos hipermedijskih datoteka, poput HTML-a. Namijenjen je kao oblik komunikacije ili posredništva između preglednika i web servera. Kao najuobičajeniji protokol on funkcionira na temelju *klijent-*

---

<sup>1</sup> Usp. History of the Web. // World Wide Web Foundation. URL: <https://webfoundation.org/about/vision/history-of-the-web/> (2024-06-10)

<sup>2</sup> Usp. Isto.

<sup>3</sup> Usp. HTML // Mozilla Developer. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (2024-06-10)

<sup>4</sup> Usp. URI // Mozilla Developer. URL: <https://developer.mozilla.org/en-US/docs/Glossary/URI> (2024-06-10)

*server* arhitekture, gdje postoji jedno središnje računalo koje poslužuje ostala računala pomoću vlastitih izvora i daje isto na korištenje.<sup>5</sup>

No prije razvitka mrežnih stranica i mjesta, postojale su zbirke strukturiranih podataka koje se obično pohranjuju elektronički unutar računalnog sustava, poznatije kao baze podataka.<sup>6</sup> Uz baze podataka postoje i sustavi namijenjeni njihovim upravljanjem (SUBP), a uz navedene sustave postoje i strukturirani jezici za upite (SQL) namijenjeni za zapisivanje i postavljanje upita.

Kako bi se bolje shvatio koncept mrežnih stranica i mjesta te samih baza podataka, ovaj završni rad pruža pregled mrežnih stranica, povijest, mijenjanje koncepta i oblikovanja, vizualnog prikaza i srodno; koncept baza podataka, vrste istoga i kakve relacije postoje. Također pomoću oblikovanje upita unutar MariaDB-a i koncepta *rudarenje podataka*, analizirat će se baza podataka koja sačinjava podatke o mrežnim mjestima na području Republike Hrvatske i stječi bolji dojam o aktualnim domenama i poddomenama istoga.

---

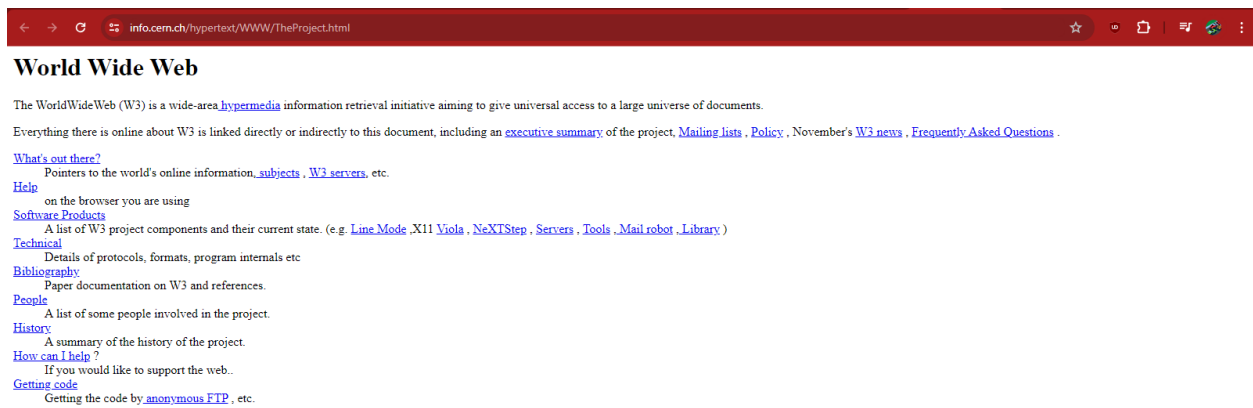
<sup>5</sup> Usp. Halonja, Antun; Mihaljević, Milica. Nazivlje računalnih mreža. // Institut za hrvatski jezik i jezikoslovlje, 29/1 (2003), str. 88.

<sup>6</sup> Usp. Što je baza podataka // Oracle Croatia. URL: <https://www.oracle.com/hr/database/what-is-database/> (2024-06-10)



## 2. Povijesni pregled mrežnih stranica

Prvu mrežnu stranicu objavio je Tim Berners Lee, stranica je sadržavala informacije o WWW-u i formate, protokole, te kako oblikovat vlastitu mrežnu stranicu. URL adresa same mrežne stranice jest „<http://info.cern.ch/hypertext/WWW/TheProject.html>“, a 1992. izgledala je kao u priloženoj slici (Slika 2).<sup>7</sup>



Slika 2: Prva mrežna stranica.

Temelj prvih mrežnih stranice je bio tekst, s obzirom na tadašnju tehnologiju koja nije predstavljala iznimno veliku brzinu kao sadašnja. Iz priloga, evidentan je nedostatak uporabe stila i dekorativnog prikaza elemenata, jer su tadašnje mrežne stranice počivale na najosnovnijim HTML elementima za poglavlje, paragraf i poveznice.

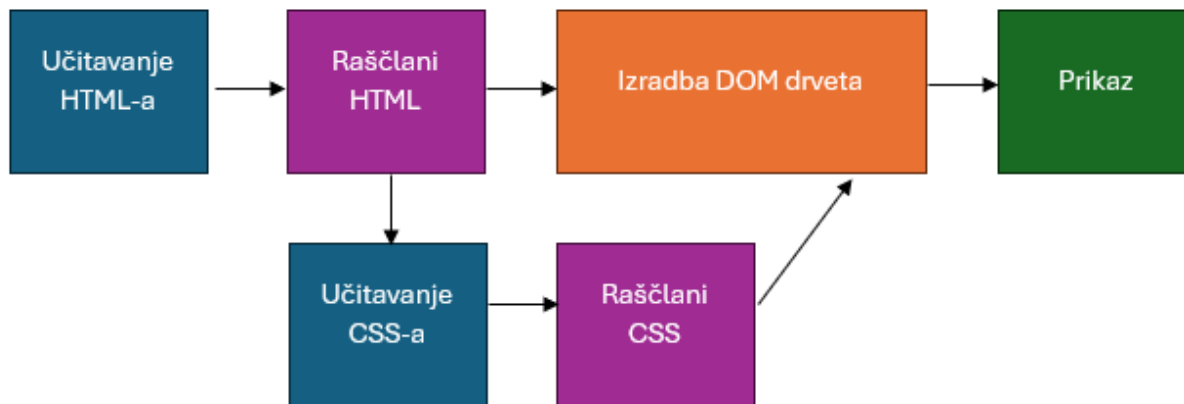
### 2.1. Implementacija CSS-a

Početak novog stoljeća donosi i novu eru, gdje počinje prevladavati CSS (*Cascading Style Sheets*). CSS je stilski jezik preko kojeg se opisuje način prikaza i predstavljanja dokumenta, koji je napisan u HTML-u; CSS opsuje kako će se elementi predočiti na ekranu i srodnim medijima.<sup>8</sup> Preglednik učitava HTML, gdje ga konvertira u DOM (*Document Object Model*) koji predstavlja dokument unutar računalne memorije; preglednik potom dohvaća sve resurse i sredstva koja su povezana s HTML dokumentom, među kojima je i CSS. Zatim slijedi korak stabla renderiranja, gdje preglednik analizira dohvaćeni CSS dokument i organizira pravila prema njihovim vrstama

<sup>7</sup> Usp. Ottervig, Vegard. Websites: past and present, 2022. URL: <https://www.enonic.com/blog/websites-past-and-present> (2024-06-11)

<sup>8</sup> Usp. CSS / Mozilla Developer. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (2024-06-11)

selektora u različita spremišta odnosno „kante“, te na temelju pronađenih selektora utvrđuje koja se pravila trebaju primijeniti na koje određene čvorove unutar DOM-a. Renderirano stablo je priloženo u namijenjenoj strukturi i vizualni prikaz stranice je prikazan (Slika 4).<sup>9</sup>



Slika 4: Način učitavanja CSS-a putem DOM-a.

Uz mogućnost definiranja stila i načina ponašanja određenih komponenti, posao web developera je poboljšán i olakšan. Mrežne stranice postaju lakše za upravljat i održavat, budući da web developeri mogu jednostavno ažurirat stilski aspekt bez potrebe modificiranja priloženog sadržaja. Također, CSS je ostvario mogućnost preglednijih i pristupačnijih mrežnih stranica za korisnike sa određenim poteškoćama ili invaliditetom. Kako se razdvaja stilski aspekt sa aspekta sadržaja u jednoj datoteci, smanjuje se veličina datoteka, što dovodi do bržeg učitavanja i renderiranja komponenti.<sup>10</sup>

<sup>9</sup> Usp. How CSS works. // Mozilla Developer. URI: [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/How\\_CSS\\_works](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works) (2024-06-11)

<sup>10</sup> Usp. The Evolution of Web Design: A Historical Perspective. // emd:digital. URL: <https://www.emddigital.com.au/blog-posts/the-evolution-of-web-design-a-historical-perspective> (2024-06-11)

### 3. Baze podataka

Kako dolazi do razvitka tehnologije, dolazi do gubitka vitalnih podataka, te problematika trajnog pohranjivanja veće količine istoga u vanjskoj memoriji računala. Kako bi se uklonila poteškoća uobičajene automatske obrade podataka, u drugoj polovici 20. stoljeća dolazi do razvitka baze podataka.<sup>11</sup> Primjenom baza podataka unutar poslovanja poboljšana je produktivnost, kvaliteta i pouzdanost u razvitku aplikativnih rješenja, mrežnih mjesta i srodnog, glede pohranjivanja i pretraživanja podataka.

Baze podataka ukratko funkcioniraju na temelju uporabe zajedničkih i objedinjenih kolekcije podataka, kako određena aplikativna rješenja i sustavi ne bi kreirali vlastitu datoteku na disk. Baza podataka upravlja podacima u posrednom obliku, gdje se primjenjuje specijaliziran programska podrška, poznat kao sustav za upravljanje bazom podataka te SUBP (*DBMS – Database Managment System*), koji je namijenjen rukovanju zajedničkoj kolekciji podataka.<sup>12</sup> SUBP služi kao posrednik nalik sučelja, između baze podataka i korisnika ili programa; omogućuje se dohvat te prikaz, ažuriranje, brisanje, upravljanje načinom organiziranja i optimiziranja podataka.<sup>13</sup> Pruža mogućnost podržavanja raznolikih baza, sa iznimkom svake baze koja posjeduje vlastitu logičku strukturu, te je u skladu s istim modelom.

---

<sup>11</sup> Usp. Manger, Robert. Baze podataka. Drugo izdanje. Zagreb: 2001. Str. 9.

<sup>12</sup> Usp. Isto.

<sup>13</sup> Usp. Što je baza podataka? // Oracle. URL: <https://www.oracle.com/hr/database/what-is-database/> (2024-06-11)

### 3.1. Vrste SUBP-a

Kako bi baza podataka imala logičku, jasnu i preglednu strukturu, model podataka je neophodan. Model podataka predstavlja niz pravila koji određuju način logičkog strukturiranja baze podataka.<sup>14</sup> Zbog široke raznolikosti određenih poslovanja, postoje i različiti sustavi namijenjeni upravljanju bazama podatka, te svaki sustav pojedinačno predstavlja mnoštvo pogodnosti u vršenju svoje dužnosti. Kako svako poslovanje ima svoj vlastiti način djelovanja, tako se mogu primijeniti i različiti modeli za određeno poslovanje, a modeli su sljedeći:

- Relacijski model
- Hijerarhijski model
- Mrežni model
- Objektno-orijentirani model
- Model entitet-odnos
- Objektno-relacijski model

Objektni se model baze podataka temelji na konceptu objektno-orijentiranog programiranja, gdje je baza koncipirana kao skup trajno pohranjenih objekata, a svaki zasebni objekt sadržava svoje interne podatke u obliku atributa i operacije, te metode za rukovanje istim podacima.<sup>15</sup>

Najraniji model podataka je bio hijerarhijski model, koji predstavlja obrnuto stablo.<sup>16</sup> Odnos datoteka predstavlja odnos roditelja-potomka (eng. *parent-child*), gdje jedan roditelj može imati više djece, ali svako dijete može imati samo jednog roditelja. Iako pruža jednostavan prizor odnosa jedan prema više (1:n), najveću poteškoću pak ima sa odnosa više prema više (n:m), jer nije fleksibilan u dodavanju novih veza, što kasnije zahtijeva promjene cijele strukture i aplikacije.<sup>17</sup>

Daljnijim razvitkom hijerarhijskog modela i otklanjanjem problema fleksibilnosti, dolazi do razvitka novog mrežnog modela podataka.<sup>18</sup> Kako je u hijerarhijskom modelu najveću poteškoću

---

<sup>14</sup> Usp. Manger, Robert. Baze podataka. Drugo izdanje. Zagreb: 2001. Str. 10.

<sup>15</sup> Usp. Isto.

<sup>16</sup> Usp. Understanding the Hierarchical Database Model. // MariaDB. URL: <https://mariadb.com/kb/en/understanding-the-hierarchical-database-model/> (2024-06-12)

<sup>17</sup> Usp. Isto.

<sup>18</sup> Usp. Understanding the Network Database Model. // MariaDB. URL: <https://mariadb.com/kb/en/understanding-the-network-database-model/> (2024-06-12)

predstavljao odnos više prema više te mogućnost potomka više roditelja, navedeni model omogućuje više roditelja. Potomci predstavljaju *članove*, a roditelji *vlasnike*. Spram primjene novijeg odnosa, mrežni model svakako ima svoje mane, a to su implementacija i održavanje. Relacije ne mogu biti zadovoljene pripisivanjem novog vlasnika.<sup>19</sup>

Najuobičajeniji model je relacijski model koji uvrštava podatke u tablice, koje se nazivaju relacijama. Svaka relacija sadržava svoje stupce i redove, a svaki stupac navodi atribut entiteta, kao što je primjerice naziv, serijski broj ili datum nastanka. Kako su relacije pogodne za prikaz određenih odnosa, istoimeni model se često primjenjuje kod većine SUBP-a i tako tvori sustav za upravljanje relacijskim bazama podataka (SURBP ili RDBMS – *Relational Database Management System*). Kao programska podrška primjenjuje se u izradi, ažuriranju i upravljanje relacijskim bazama podatka.<sup>20</sup>

Uz razvitak objektnog modela baze podataka, razvijaju se drukčije ne-relacijske baze podataka i tako nastaje NoSQL. NoSQL („not only SQL“) primjenjuje ne-tablični format za pohranu podataka naspram relacijskih tablica. NoSQL implementira fleksibilni model sheme koji podržava široki raspon nestrukturiranih podataka, poput dokumenata, ključ-vrijednosti, grafove itd.<sup>21</sup>

DB2 je proizvod poduzeća IBM, namijenjen z/OS operacijskom sustavu; DB2 je oblikovan kako bi sukladno djelovao uz centralnu jedinicu IBM sustava z.<sup>22</sup> Kao krucijalna komponenta IBM proizvoda, DB2 za z/OS sustav implementira strukturirani upitni jezik (SQL – *Structured Query Language*) i relacijski model baze podataka.

Oracle je proizvod istoimene tvrtke, te služi kao platforma za izgradnju i uspješno upravljanje sustavima relacijskih baza podataka. Zbog pogodnosti glede pohranjivanja i pregledavanja podataka, mnoštvo poduzeća ulaže u ovakav oblik podatkovnog sustava.<sup>23</sup> Funkcionira s korisničkim i poslužiteljskim sučeljem, te radnja se zahtjeva ili izvodi, a sam poslužitelj daje

---

<sup>19</sup> Usp. Understanding the Network Database Model. // MariaDB. URL: <https://mariadb.com/kb/en/understanding-the-network-database-model/> (2024-06-12)

<sup>20</sup> Usp. What is a relation database? // URL: <https://cloud.google.com/learn/what-is-a-relational-database> (2024-08-14)

<sup>21</sup> Usp. What is a NoSQL database? // URL: <https://cloud.google.com/discover/what-is-nosql> (2024-08-14)

<sup>22</sup> Usp. Introduction to DB2 for z/OS. // IBM. URL: <https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-introduction-db2> (2024-06-12)

<sup>23</sup> Usp. Što je Oracle i za što radi? Kako ga razviti? // Vida Bytes. URL: <https://vidabytes.com/hr/%C5%A1to-je-proro%C4%8Di%C5%A1te/> (2024-06-12)

odgovor omogućujući slanje zahtjeva od strane korisnika. Sustav sam po sebi nema opseg spram ostalih baza podataka; glavna značajka jest mogućnost rada s različitim razinama i količinom podataka.<sup>24</sup> Glede posredništva i komunikacije primjenjuje se PL / SQL (Proceduralno jezična ekstenzija za strukturirani upitni jezik).

MS SQL Server je Microsoft-ov proizvod namijenjen poslužiteljskim računalima koje funkcioniraju i djeluju na temelju operacijskog sustava MS Windows. Srodan kao i ostali SURBP, SQL Server se temelji na SQL-u kako bi komunicirao sa bazom podataka. SQL je Server povezan sa T-SQL-om, Microsoft-ova implementacija i konstrukcija SQL-a, koja uključuje skup vlasničkih programskih konstrukcija.<sup>25</sup>

MySQL je najpoznatiji SURBP otvorenog pristupa, a razvija, distribuira i održava ga Oracle.<sup>26</sup> MySQL je iznimno jednostavan za koristiti i upravljat mnoštvom baza. Glede pouzdanosti on je najpouzdaniji i najrasprostranjeniji sustav, s obzirom na poduzeća koje se oslanjaju na njega glede pokretanja i održavanja poslovnih aplikacija. S obzirom na optimizaciju zahtjeva pristupačnosti aplikacija, MySQL-ova izvorna arhitektura replikacije omogućuje velikim poduzećima i organizacijama skaliranje aplikacija kako bi podržavali veliki broj korisnika.<sup>27</sup> Nakon što je MySQL preuzeo Oracle, osnivač MySQL-a Michael Widenius razvija alternativu i potom dolazi do osnutka MariaDB.<sup>28</sup> Kao i njezin prethodnik, MariaDB isto naglašava važnost optimizacije i skaliranja, tako što dijeli opterećenja baze podataka na mnoštvo poslužitelja. Također MariaDB održava visoku razinu kompatibilnosti sa svojim prethodnikom MySQL, te također mnoštvo aktualnih i popularnih aplikacija koje primjenjuju MySQL glede pohrane podataka, mogu istovremeno primijeniti i MariaDB.<sup>29</sup>

---

<sup>24</sup> Usp. Isto.

<sup>25</sup> Usp. What is SQL Server? // SQLServer Tutorial.net. URL: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> (2024-06-12)

<sup>26</sup> Usp. What is MySQL? // MySQL. URL: <https://dev.mysql.com/doc/refman/8.4/en/what-is-mysql.html> (2024-06-12)

<sup>27</sup> Usp. Isto.

<sup>28</sup> Usp. MariaDB Server. // MariaDB. URL: <https://mariadb.org/en/> (2024-06-12)

<sup>29</sup> Usp. Isto.

## 4. Relacijski model

Edgar Codd razvija teoretski koncept relacijskog modela podataka u drugoj polovici 20. stoljeća. Prve su primjene modela bile iznimno spore i nepraktične, no zahvaljujući intenzivnom istraživanju i razvoju, funkcionalnost relacijskog modela doseže veću stopu uporabe i primjene.<sup>30</sup>

Bazu podataka čine skup tablica, te relacije. Svaka relacija je jedinstvena po svom imenu, a jedan stupac u svakoj relaciji sadržava vrijednost jednog atributa, te entitet ili vezu. Atribut ima svoje ime po kojem ga razlikujemo od istoga u samoj relaciji; postoji mogućnost da dvije relacije imaju istoimene attribute, no tada se podrazumijeva da je riječ o atributima s istim značenjem u različitim entitetima.<sup>31</sup> Vrijednost atributa mora biti jednostruka i jednostavna, te ne smije biti ponovljena i rastavljena na manje dijelove cjeline; jedan redak relacije predstavlja jedan primjer entiteta i može predstavljati vezu između dva ili više entiteta. Redak tablice jest *n-torka*, u jednoj relaciji ne smije biti dvije jednake *n-torke*, budući da relaciju tumačimo kao skup *n-torki*.<sup>32</sup>

Primjerice relacija auto, s atributima identifikator, godište, boja, marka i model. Navedena relacija sadrži podatke o autima koje je vlasnik imao (Slika 5). Atribut identifikator služi za jedinstvenu identifikaciju auta, s obzirom na mogućnost postojanja dva ili više auta istog godišta, boje, marke i modela.

identifikator	godiste	boja	marka	model
1	1992.	Crna	Mercedes	SL 300 24V
2	2016.	Bijela	Škoda	2,0 TDI DSG
3	2021.	Plava	BMW	Seriya 3 / M340D
4	2005	Metalik siva	Toyota	Corolla 2.0

Slika 5: Relacija s podacima o autima.

Spram primjene stručne terminologije u mnoštvu literature, češće se upotrebljavaju neposredni i konvencionalni termini, te tablica umjesto relacija, redak umjesto *n-torka* i stupac umjesto atribut.<sup>33</sup>

<sup>30</sup> Usp. Manger, Robert. Baze podataka. Drugo izdanje. Zagreb: 2001. Str. 43.

<sup>31</sup> Usp. Isto.

<sup>32</sup> Usp. Isto.

<sup>33</sup> Usp. Manger, Robert. Baze podataka. Drugo izdanje. Zagreb: 2001. Str. 44.

## 4.1. Primarni i vanjski ključ

Primarni ključevi omogućuju jedinstveno identificiranje svakog retka unutar tablice. Svrha primarnih ključeva jest dohvaćanje svakog podatka, koji je unesen u bazu podataka. Ako su primarni ključevi jedinstveni za svaki redak, onda u potpunosti znamo da dohvaćamo redak s podacima koji je potreban.<sup>34</sup> Uz jedinstvenost, primarni ključ ne smije sadržavat vrijednost *null*. Null je samorazumljiv, on je posebna vrijednost baze podataka koja referira na „nepoznato“. Jedan primarni ključ s null vrijednošću je prihvatljiv, jer još uvijek imamo koncept jedinstvenosti; međutim, ukoliko imamo još jedan ili više primarnih ključeva s istom vrijednošću, gubimo svojstvo jedinstvenosti.

Kada tablica sadrži stupac, koji je sličan primarnom ključu tablice u bazi podataka, navedeni stupac se naziva *vanjskim ključem*. Identičnost vrijednosti vanjskog i primarnog ključa predstavlja podatkovni odnos u relacijskoj bazi podataka.<sup>35</sup> Primjerice tablica *odjel* i tablica *djelatnik*. Vanjski ključ u tablici *djelatnik* (*id\_odjel*) može služiti kao poveznica na tablicu *odjel* kako bi se *djelatnik* mogao povezati s njegovim pripadajućim *odjelom*.

*vlasnik* (*id\_vlasnik*, *ime*, *prezime*, *datum\_rodenja*, *oib*)

*proizvodac* (*id\_proizvodac*, *naziv*, *drzava*, *godina\_osnutka*)

*model* (*id\_model*, *naziv*, *id\_proizvodac*)

*auto* (*id*, *boja*, *godiste*, *model*, *id\_vlasnik*, *id\_proizvodac*)

Relacijski SUBP primjenjuje implicirane odnose povezujući podatke između primarnih i vanjskih ključeva.<sup>36</sup>

---

<sup>34</sup> Usp. Harrington, Jan, L. *Relational Database Design and Implementation*, Burlington, MA, USA: Morgan Kaufman. 2009. Str. 92-93.

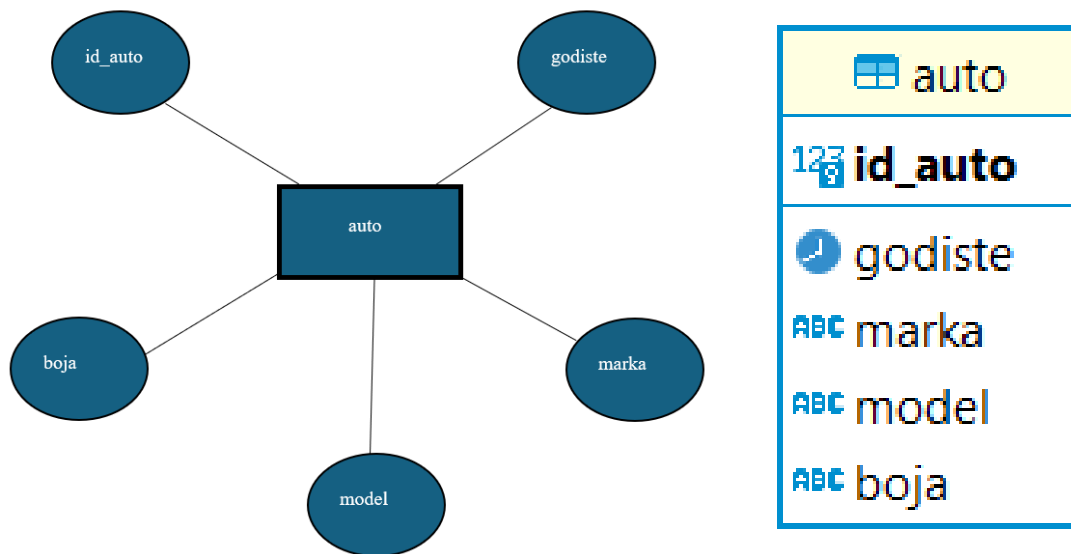
<sup>35</sup> Usp. Isto, str. 98.

<sup>36</sup> Usp. Isto, str. 99.



## 5. ERA dijagram

Glede preglednosti i prikaz odnosa među entitetima koncipirani su dijagrami, koji omogućuju shvaćanje entiteta koji prebivaju unutar baze uz attribute koji opisuju isto. ERA dijagram predstavlja grafički prikaz entiteta-atributa-relacije (*entity-relationship ER*).<sup>37</sup> Trenutačno postoje tri moguća prikaza: Chen model, informacijsko inženjerstvo i jedinstveni jezik za modeliranje (*unified modeling language*). Navedeni prikazi primjenjuju pravokutnike za predstavljanje entiteta, a kružnice za predstavljanje atributa (Slika 6).<sup>38</sup>



Slika 6: Primjer Chen model dijagrama s autom i ERA dijagrama tablice iz DBEaver notacije.

Glede preglednosti i prikaza manje zgušćenih dijagrama, te fleksibilnosti često se primjenjuju information engineering (IE ili „crows feet“) i unified modeling language (UML) prikazi naspram Chenov-og modela.<sup>39</sup> UML dijagram omogućuje jednostavnije razumijevanje i shvaćanje kompleksnih ideja i sustava, pruža jednostavniji pregled šireg aspekta programa i omogućuje osobama koje se ne bave poslovima razvoja aplikativnih rješenja, lakše razumijevanje procesa i funkcionalnosti programske podrške.<sup>40</sup>

<sup>37</sup> Usp. Harrington, Jan, L. Relational Database Design and Implementation, Burlington, MA, USA: Morgan Kaufman. 2009. Str. 61.

<sup>38</sup> Usp. Isto, str. 62.

<sup>39</sup> Usp. Harrington, Jan, L. Relational Database Design and Implementation, Burlington, MA, USA: Morgan Kaufman. 2009. Str. 62.

<sup>40</sup> Usp. What is a UML diagram? // Miro. URL: <https://miro.com/diagramming/what-is-a-uml-diagram/#benefits-of-uml-diagrams-> (2024-06-24)

## 6. Rad s bazom

### 6.1. Osnovni preduvjeti za rad

Kako bi uspješno radili s bazom koja posjeduje podatke o aktualnim mrežnim mjestima te domenama i poddomenama unutar .hr domene, neophodna nam je sljedeća tehnologija: XAMPP, DBeaver. XAMPP je više platformi mrežni poslužitelj koji je besplatan i dostupan u otvorenom pristupu. Dolazi u paketu s Apache serverom, MariaDB, PHP i Perl programskim jezicima, omogućujući programerima pisanje i testiranje vlastitog koda na lokalnom poslužitelju. Iako je MariaDB dio XAMPP-a, ona se također može koristiti izvan XAMPP-a kao neovisna tehnologija.<sup>41</sup>

DBeaver je besplatan, grafički alat za upravljanje bazama podataka dostupan u otvorenom pristupu, namijenjen razvojnim programerima i administratorima baza podataka. DBeaver se može primjenjivati u izradi i upravljanju bazama podataka kroz širok raspon SUBP, te on funkcionira s najpopularnijim SUBP kao što su MySQL, PostgreSQL, MariaDB, SQLite, Oracle, DB2, SQL Server, Microsoft Access itd.<sup>42</sup> DBeaver ima grafičko korisničko sučelje (eng. prijevod GUI), koji omogućuje prikazivanja velikog broja baza podataka i njihovih objekata, te ubrzava proces istoga. Nudi se mogućnost ažuriranja i uređivanja podataka unutar tablice i izrada vizualnih dijagrama objekata baze podataka. Također, DBeaver podržava mnoštvo značajki koje su specifične za svaki pojedinačni SUBP.<sup>43</sup>

---

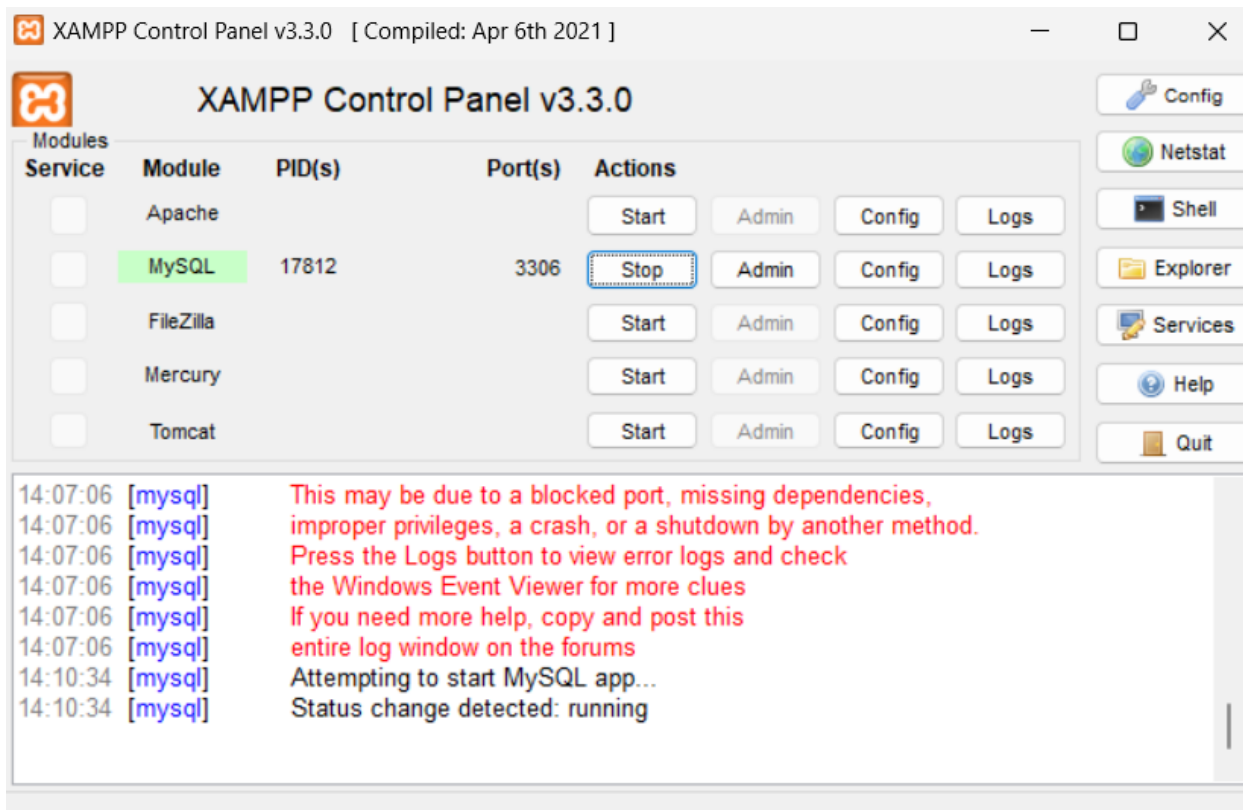
<sup>41</sup> Usp. What is XAMPP? // Educba. URL: <https://www.educba.com/what-is-xampp/> (2024-06-27)

<sup>42</sup> Usp. What is DBeaver? // Database.Guide. URL: <https://database.guide/what-is-dbeaver/> (2024-06-27)

<sup>43</sup> Usp. Isto.

## 6.2. Podešavanje okruženja za rad

Kako bi uspješno radili i upravljali MariaDB SURBP potrebno je pokrenut XAMPP i podesiti kontrolnu ploču, te MySQL<sup>44</sup> bazu podataka (Slika 7). Apache server nije nužan za pokretanje, budući da sam rad nije koncipiran na tehnologiji kojoj je navedeni server neophodan; stoga njegovo pokretanje nije uopće potrebno.

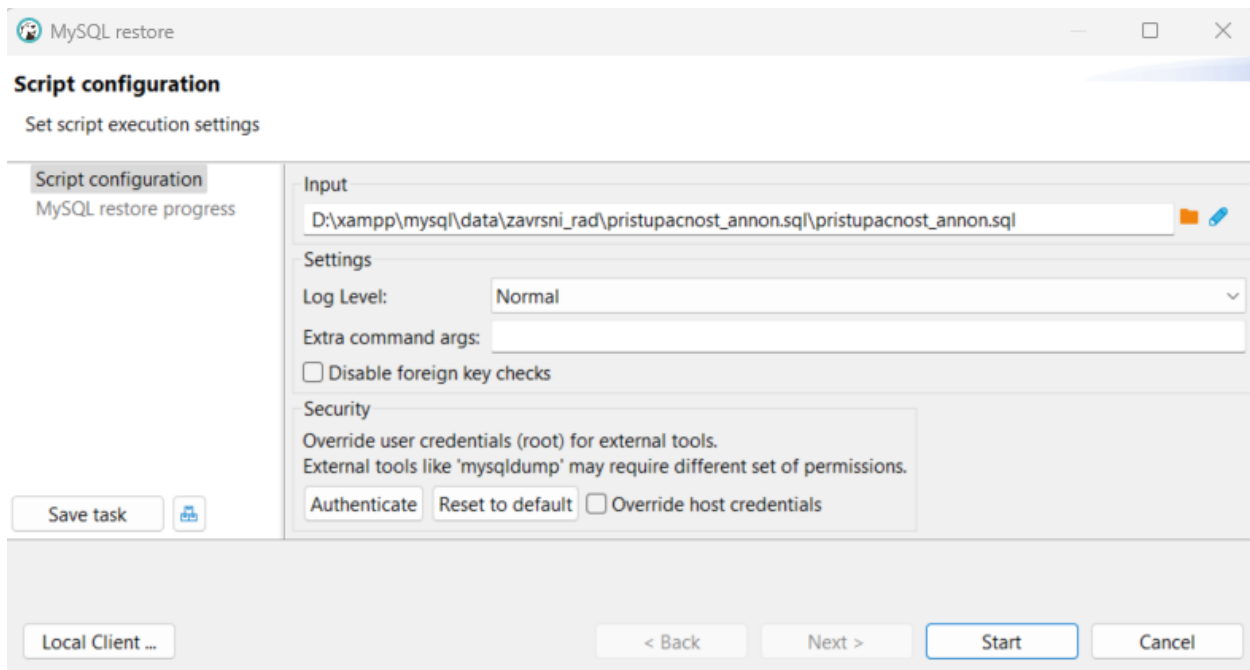


Slika 7: XAMPP kontrolna ploča s pokrenutim apache serverom i MySQL bazom podataka.

Potom pokrećemo DBeaver. Na lijevoj strani možemo identificirati trenutno dostupne veze, među kojima je i naša veza koja je omogućena putem XAMPP-a. Unutar nje, nalaze se baze podataka, a tamo će se smjestiti naša baza s njezinim pripadajućim podacima.

Nakon raspakiranja obnavljamo bazu koristeći mogućnost *restore database*, gdje nam se obnavljaju sve tablice i njezini pripadajući podaci (vidi Slika 8). Pri završetku obnove baze možemo započeti s analizom i „rudarenjem“ podataka.

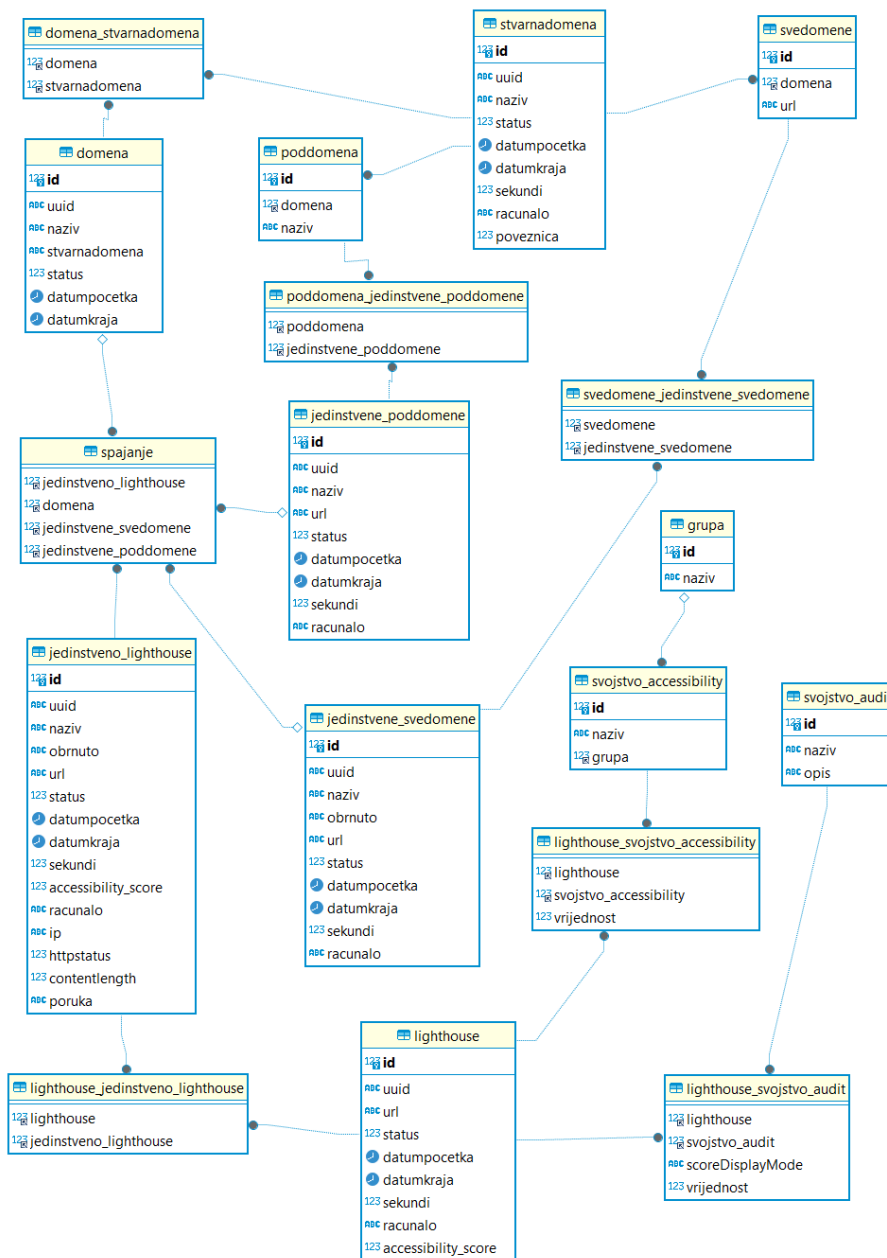
<sup>44</sup> XAMPP se prebacio s MySQL na MariaDB, a budući da je MariaDB zamjena za MySQL, vidjet će se neki određeni direktoriji s imenom mysql, ali zapravo je MariaDB iza njih.



Slika 8: Proces obnavljanje baze, tablice i pripadajućih podataka.

### 6.3. ERA dijagram i tablice baze podataka

Glede preglednosti i boljeg razumijevanja baze, te njezine strukture izrađen je ERA dijagram kako bi se bolje razumjela svaka pojedinačna tablica i relacija istoga (Slika 9). Tehnologija koja je primijenjena za izradu navedene baze i njezinih tablica je MariaDB.



Slika 9: ERA model baze podataka „Mrežna mjesta RH“.

Osnove navedene baze su tablice:

1. domena
2. domena\_stvarnadomena
3. grupa
4. jedinstvene\_poddomene
5. jedinstvene\_svedomene
6. jedinstveno\_lighthouse
7. lighthouse
8. lighthouse\_jedinstveno\_lighthouse
9. lighthouse\_svojstvo\_accessibility
10. lighthouse\_svojstvo\_audit
11. poddomene
12. poddomene\_jedinstvene\_poddomene
13. spajanje, stvarnadomena
14. svedomene
15. svedomene\_jedinstvene\_svedomene
16. svojstvo\_accessibility, svojstvo\_audit.

Kako bi se uspostavila neovisnost među podacima samim tablicama, implementira se relacijska veza između tablica, kako bi se postiglo povezivanje među tablicama i osigurala preglednost relacija.

Tablica „domena“ sadrži podatke o domenama, a njezini stupci su id, uuid, stvarnadomena, status, datumpocetka i datumkraja. Navedena tablica se povezuje s prethodno spomenutom tablicom „stvarnadomena“ putem atributa stvarnadomena. Također se povezuje s tablicom „svedomene“ preko atributa domena. Podaci koje tablica predstavljaju incijalno dobiveni popis od CARNET-a (Hrvatska akademska i istraživačka mreža ili Croatian Academic and Research Network), koji je ovlašten dati iste podatke. Sami podaci su iz siječnja 2024. godine.

Tablica „stvarnadomena“ sadrži podatke o svim stvarnim domenama, te sadrži stupce id, uuid, naziv, status, datumpocetka, datumkraja, sekundi, racunalo i poveznica. Njezin odnos je predstavljen u obliku poveznice sa tablicom „domena“ preko atributa stvarnadomena u istoj tablici.

Veza je n:m, gdje jedna domena može imati više stvarnih domena dok se jedna stvarna domena može odnositi na više domena.

Tablica „poddomena“ sadrži podatke o poddomenama i stupce id, domena i naziv. Povezuje se sa tablicom „domena“ u obliku odnosa 1:n, gdje jedna domena može imati više poddomena, dok svaka poddomena pripada jednoj domeni.

Tablica „jedinственe\_poddomene“ sadrži podatke o jedinstvenim poddomenama, a njezini stupci su id, uuid, naziv, url, status, datumpocetka, datumkraja, sekundi i racunalo. Kako bi uspostavila odnos s drugim tablicama povezana je sa tablicom „poddomena\_jedinственe\_poddomene“ preko atributa jedinstvene\_poddomene, te također je povezana s tablicom spajanje preko atributa jedinstvene\_poddomene. Njezina veza je n:m, gdje jedna poddomena može imati više jedinstvenih poddomena, a svaka jedinstvena poddomena može biti asocirana sa više poddomena.

Tablica „poddomena\_jedinственe\_poddomene“ služi kao poveznica između poddomene i jedinstvenih poddomena. Sadrži stupce poddomena i jedinstvene\_poddomene, a preko navedenih stupaca povezuje tablice poddomena i jedinstvene\_poddomene. Navedena tablica omogućuje fleksibilnu vezu, gdje poddomene mogu imati više jedinstvenih identifikatora.

Tablica „spajanje“ služi kao veza između jedinstvenih svedomena, jedinstvenih poddomena i jedinstvenog lighthouse-a. Sadrži stupce jedinstveno\_lighthouse, domena, jedinstvene\_svedomene i jedinstvene\_poddomene, a sama tablica služi kako bi povezala tablice jedinstvene\_svedomene, jedinstvene\_poddomene i jedinstveno\_lighthouse. Veza je n:m, gdje sve jedinstvene sve domene mogu biti povezane s više jedinstvenih poddomena i imati više jedinstvenih Lighthouse rezultata i obratno. Navedena veza podržava kompleksne asocijacije među ovim entitetima.

Tablica „jedinственe\_svedomene“ sadrži podatke o svim jedinstvenim domenama. Njezini stupci su id, uuid, naziv, obrnuto, url, status, datumpocetka, datumkraja, sekundi i racunalo. Povezuje se sa tablicom „svedomene\_jedinственe\_svedomene“ preko atributa „jedinственe\_svedomene“, te također se povezuje sa tablicom „spajanje“ preko atributa „jedinственe\_svedomene“. Veza je n:m, jedna sva domena može biti povezana sa više jedinstvenih

sve domena, te svaka jedinstvena sva domena može biti asocirana sa više sve domena. Olakšava sveobuhvatno mapiranje sažetih domena na jedinstvene identifikatore.

Tablica „svedomene“ sadrži podatke o svim domenama, a navedena tablica sadrži stupce id, domena i url. Odnos je uspostavljen putem tablice „domena“, tako što je povezana preko atributa domena tablice svedomene, a istovremeno je povezana sa tablicom „svedomene\_jedinstvene\_svedomene“ preko atributa svedomene navedene tablice. Veza je 1:n, jedna domena može sadržavati više svih domena, a više svih domena pripada jednoj domeni.

Tablica „svedomene\_jedinstvene\_svedomene“ služi kao veza između tablice „svedomene“ i „jedinstvene\_svedomene“. Omogućuje vezu n:n između.

Tablica „lighthouse“ sadrži podatke o Lighthouse rezultatima. Lighthouse je automatizirani alat u otvorenom pristupu, koji omogućuje poboljšavanje kvalitete mrežnih stranica, a nudi audite za izvedbu, pristupačnost, progresivne web aplikacije (PWA), optimizacija tražilica i više.<sup>45</sup> Lighthouse ocjena pristupačnosti je izračunat prosjek svih audita pristupačnosti. Ocjena pristupačnosti može biti od 0 do 100. Ako je ocjena između 90 i 100, upućuje na odlično optimizirano korisničko iskustvo stranice. Sve ispod 90 može se odnositi na određen broj resursa koji osporavaju samu stranicu.<sup>46</sup> Navedena tablica ima stupce id, uuid, url, status, datumpocetka, datumkraja, sekundi, racunalo i accessibility\_score. Povezuje se sa tablicama „jedinstveno\_lighthouse“ preko atributa id, „lighthouse\_jedinstveno\_lighthouse“ preko atributa id i „lighthouse\_svojstvo\_accessibility“ preko atributa id. Veza je 1:n, gdje svaki Lighthouse rezultat može imati više jedinstvenih Lighthouse rezultata, ali svaki jedinstveni Lighthouse unos povezan je s jednim Lighthouse rezultatom.

Tablica „jedinstveno\_lighthouse“ sadrži podatke o jedinstvenim Lighthouse rezultatima. Stupci su id, uuid, naziv, obrnuto, url, status, datumpocetka, datumkraja, sekundi, racunalo, ip, httpstatus, contentlength i poruka. Povezuje se s tablicama „spajanje“ preko atributa id i „lighthouse\_jedinstveno\_lighthouse“ preko atributa id.

---

<sup>45</sup> Usp. Lighthouse. // Chrome for Developers. URL: <https://developer.chrome.com/docs/lighthouse/overview> (2024-07-03)

<sup>46</sup> Usp. Pol, Tushar. Google Lighthouse: What It Is & How to Use It?. // Semrush Blog. URL: <https://www.semrush.com/blog/google-lighthouse/> (2024-08-21)



Tablica „lighthouse\_jedinstveno\_lighthouse“ služi kao veza između tablice „lighthouse“ i „jedinstveno\_lighthouse“, te sadrži istoimene stupce. Veza je n:m, svaki Lighthouse rezultat može se odnositi na više unosa jedinstvenih Lighthouse rezultata i obratno, što omogućuje višestruke asocijacije između Lighthouse rezultata i njihovih jedinstvenih identifikatora.

Tablica „svojstvo\_accessibility“ sadrži podatke o svojstvima pristupačnosti, a sadrži stupce id, naziv i grupa. Odnos se ostvaruje povezivanjem sa tablicom „lighthouse\_svojstvo\_accessibility“ putem atributa id. Veza je 1:n, gdje svako svojstvo pristupačnosti (svojstvo\_accessibility) može biti asocirano s više unosa svojstva pristupačnosti Lighthouse-a; no svaki se unos povezuje nazad na jedno svojstvo pristupačnosti. Ovo podrazumijeva svako svojstvo utječe na više Lighthouse rezultata.

Tablica „lighthouse\_svojstvo\_accessibility“ služi kao veza između Lighthouse-a i svojstva pristupačnosti, odnosno tablica „lighthouse“ i „svojstvo\_accessibility“. Sadrži stupce lighthouse, svojstvo\_accessibility i vrijednost.

Tablica „svojstvo\_audit“ sadrži informacije o svojstvima audita. Stupce koje posjeduje su id, naziv i opis. Odnos se ostvaruje povezivanjem tablice „lighthouse\_svojstvo\_audit“ preko atributa id. Veza je 1:n, svako svojstvo audita utječe na više unosa svojstva Lighthouse audita; no svaki je unos odgovoran jednom svojstvu audita, što ukazuje na primjenu svakog svojstva audita u više Lighthouse audita.

Tablica „lighthouse\_svojstvo\_audit“ služi kao veza između Lighthouse-a i svojstva audita, te istoimenih tablica „lighthouse“ i „svojstvo\_audit“. Sadrži stupce „lighthouse“, „svojstvo\_audit“, „scoreDisplayMode“ i „vrijednost“.

Tablica „grupa“ sadrži podatke o grupama, a stupci su id i naziv. Povezuje se sa tablicom „svojstvo\_accessability“ preko vlastitog atributa id. Veza je 1:n, svaka grupa može uključiti više unosa svojstva pristupačnosti, no svaki unos svojstva pristupačnosti pripada jednoj grupi. Grupa može imati više asociranih svojstava.

## 6.4. Rudarenje i analiza podataka

Podaci baze su prikupljeni u sklopu istraživanja pobiranja mrežnih mjesta .hr domene u siječnju 2024; podaci su anonimizirani i rezultati daju numeričke vrijednosti bez konkretnih tekstualnih. Kako bi se prikupili sami podaci i stekli bolji uvid u njihovo značenje, primijeniti će se koncept rudarenja podataka i analiza istoga. Analiza podataka ili podatkovna analiza predstavlja skup aktivnosti sistematskog prikupljanja, transformacije, opisivanja, modeliranja i interpretacije podataka, primjenom statističkih tehnika. Tehnike navedene analize su neophodne u stjecanju uvida samih skupova podataka, koji potom mogu doprinijeti pri donošenju poslovnih ili znanstvenih odluka.<sup>47</sup>

Rudarenje podatka je proces organizacije i sortiranja kroz golemu količinu podataka, kako bi se identificirali odnosi koji mogu doprinijeti poslovnim problemima primjenom analize podataka. Rudarenje podataka je ključna komponenta podatkovnoj analizi, te ujedno jedna od glavnih disciplina podatkovne znanosti. Postupak rudarenja oslanja se na uspješnu implementaciju skupa podataka, skladištenje i procesiranje istoga: može se primijeniti u opisivanju određenog skupa podataka, predviđanje ishoda, te osvješćivanje o korisnicima i njihovim navikama.<sup>48</sup>

Kako bi se stekao bolji uvid o navedenoj bazi, primijeniti će se navedeni principi analize i rudarenja podataka. Rudarenje je primijenjeno u obliku formuliranih upita koristeći naredbu *select* uz dodavanje određenih uvjeta kako bi upit odgovarao postavljenom pitanju. Potom se vrši analiza podataka, gdje se stječe dojam i informacije o stanju hrvatskih mrežnih mjesta, te domena, poddomena i više.

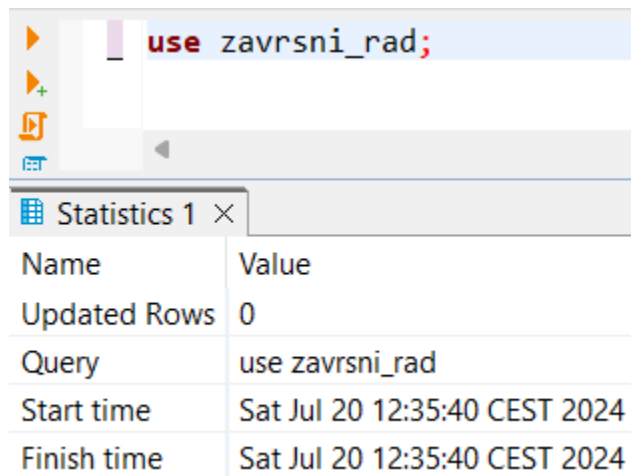
---

<sup>47</sup> Usp. Eldridge, Stephen. Data analysis. // Britannica. URL: <https://www.britannica.com/science/data-analysis> (2024-07-05)

<sup>48</sup> Usp. S. Gillis, Alexander; Stedman, Craig; Hughes, Adam. Data mining. // TechTarget. URL: <https://www.techtarget.com/searchbusinessanalytics/definition/data-mining> (2024-07-05)

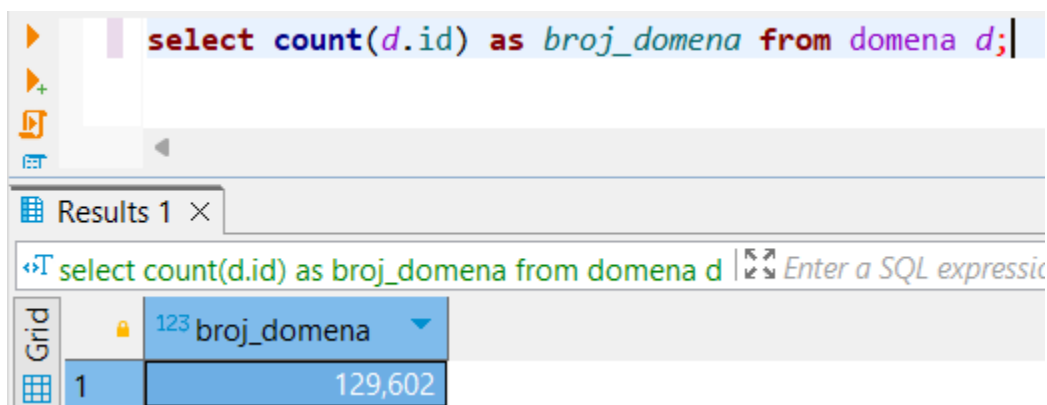
## 6.4.1. Upiti

Prvo se trebamo pozicionirati u bazu, a to ostvarujemo naredbom „use“ i potom naziv baze (Slika 10).



Slika 10: Prikaz uspješnog pozicioniranja u bazu podataka.

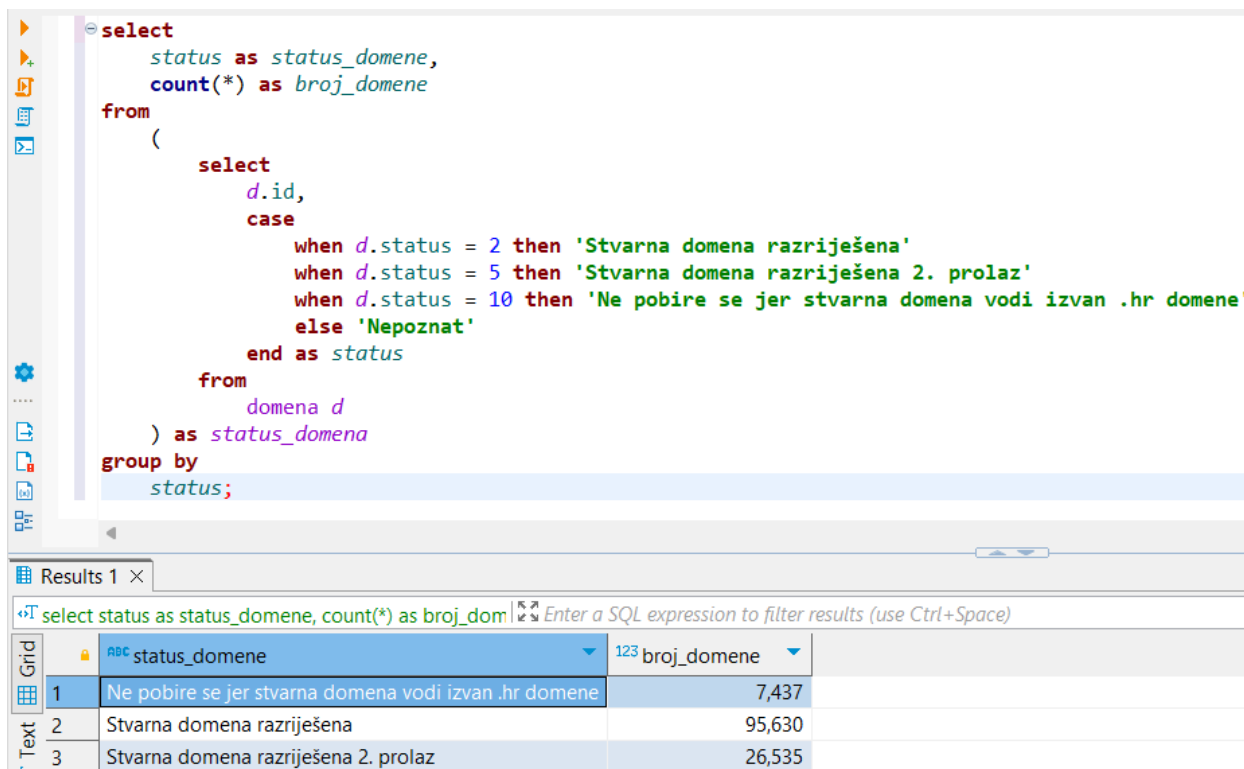
Kako bi mogli dohvatiti određen broj, primjerice vrijednosti unutar samih stupaca, primijeniti ćemo agregiranu funkciju `count()`. Navedenu funkciju primjenjujemo kako bi dohvatili broj ukupnih domena unutar tablice „domena“ (vidi Slika 11). Postavljamo pitanje „Koliki je broj domena?“.



Slika 11: Upit koji dohvaća broj domena unutar tablice „domena“.

Navedenim upitom odgovaramo na prethodno pitanje i doznajemo da broj domena na području Republike Hrvatske iznosi 129.602.

Budući da tablica „domena“ posjeduje attribute id, uuid, naziv, stvarnadomena, datumpoceka, datumkraja, te status, možemo inkorporirati podupite i potom slučajeve (eng. cases) koji funkcioniraju na temelju ako, drugo i onda izjave kako bi evaluirali uvjete i vratili vrijednost statusa kada je prvi uvjet zadovoljen (vidi Slika 12). Postavljamo pitanje „Koliki je broj domena koje imaju status 2, 5 i 10?“.



```
select
  status as status_domene,
  count(*) as broj_domene
from
(
  select
    d.id,
    case
      when d.status = 2 then 'Stvarna domena razriješena'
      when d.status = 5 then 'Stvarna domena razriješena 2. prolaz'
      when d.status = 10 then 'Ne pobire se jer stvarna domena vodi izvan .hr domene'
      else 'Nepoznat'
    end as status
  from
    domena d
) as status_domene
group by
  status;
```

Results 1 x

select status as status\_domene, count(\*) as broj\_domene | Enter a SQL expression to filter results (use Ctrl+Space)

Grid	ABC status_domene	123 broj_domene
1	Ne pobire se jer stvarna domena vodi izvan .hr domene	7,437
2	Stvarna domena razriješena	95,630
3	Stvarna domena razriješena 2. prolaz	26,535

Slika 12: Upit koji dohvaća domene iz tablice „domena“ sa svojim pripadajućim statusom.

Navedenim upitom doznajemo da ima najviše domena koje imaju status 2 čiji status predstavlja razriješenje stvarne domene, potom slijede domene koje imaju status 5 čije su stvarne domene razriješene na drugom prolazu, a najmanje ima domena sa statusom 10 koje se ne pobiru, budući da njihove stvarne domene vode izvan .hr domene. Odgovorom na postavljeno pitanje zaključujemo da najviše prevladavaju domene čije su stvarne domene razriješene, a njih ima 95.630.

Spajanjem više tablica, koristeći *join* možemo ostvariti poveznice među više tablica, a i time bogatiji upit s kojim ćemo prikupiti veću količinu podataka. Kako postoji tablica „spajanje“ sa svojim pripadajućim atributima namijenjeni za povezivanje, možemo primijeniti odnos više naprema više (n:m) između tablica „domena“ i „jedinствене\_svedomene“; tako oblikujući upit koji doznajemo domenu sa svojim identifikatorom i brojem svih jedinственih domena (vidi Slika 13). Postavljamo pitanje „Koja domena imaju najviše povezanih svih jedinственih domena?“.

```

select
    d.id as domena,
    count(distinct jsv.id) as broj_svih_jedinственih_domena
from
    domena d
    inner join spajanje s on d.id = s.domena
    inner join jedinствене_svedomene jsv on s.jedinствене_svedomene = jsv.id
group by
    d.id
order by
    2 desc
limit
    10;

```

Grid	domena	broj_svih_jedinственih_domena
1	115,481	7
2	3,665	4
3	108,340	4
4	15,557	3
5	117,502	2
6	42,446	2
7	91,359	2
8	89,359	2
9	120,874	2
10	71,697	2

Slika 13: Upit koji dohvaća identifikator svake pojedinačne domene i njezin broj svih jedinственih domena.

Domena koja ima najviše povezanih svih jedinственih domena je domena s identifikatorom 115.481, navedena domena ima 7 povezanih svih jedinственih domena.

Kako imamo tablicu „lighthouse“ i atribut `accessibility_score` možemo primijeniti agregiranu funkciju `avg()`, kako bi izračunali prosječnu ocjenu pristupačnosti različitih domena, primijeniti `concat()` za dodavanje „%“, te potom broj lighthouse testova koji su izvršeni primjenom `count()`. Budući da postoji vjerojatnost da neke domene nisu ostvarile `accessibility_score`, primjenjuje se `coalesce()` funkcija kako bi upit umjesto null vrijednosti prikazao vrijednost koju smo mi zadali (vidi Slika 14). Postavljamo pitanje „Koje su domene i njihovi prosječne ocjene pristupačnosti?“.

```

select
  d.id as domena,
  count(l.id) as broj_lighthouse_testova,
  coalesce(concat(avg(l.accessibility_score * 100), '%'), 'Nema rezultata')
  as prosjecni_accessibility_score
from
  domena d
  inner join spajanje s on d.id = s.domena
  inner join jedinstveno_lighthouse jl on s.jedinstveno_lighthouse = jl.id
  inner join lighthouse_jedinstveno_lighthouse ljl on jl.id = ljl.jedinstveno_lighthouse
  inner join lighthouse l on ljl.lighthouse = l.id
  inner join lighthouse_svojstvo_accessibility lsa on l.id = lsa.lighthouse
  inner join svojstvo_accessibility sa on lsa.svojstvo_accessibility = sa.id
group by
  d.id
order by
  2 desc
limit
  10;

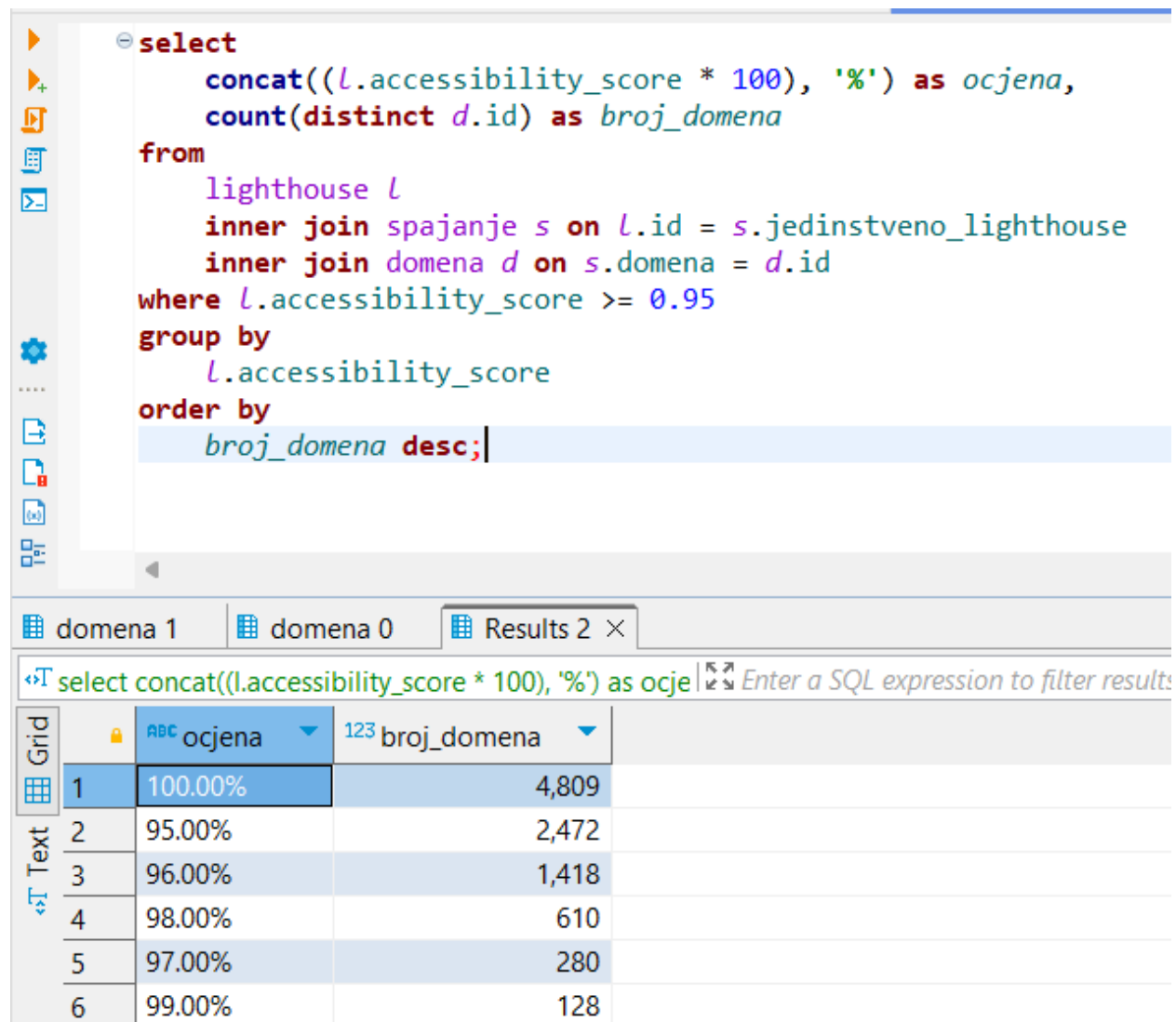
```

	domena	broj_lighthouse_testova	prosjecni_accessibility_score
1	115,481	497	79.000000%
2	3,665	284	Nema rezultata
3	108,340	284	95.000000%
4	15,557	213	83.000000%
5	91,359	142	78.000000%
6	23,913	142	73.000000%
7	101,597	142	96.000000%
8	70,480	142	86.000000%
9	96,368	142	Nema rezultata
10	108,636	142	68.000000%

Slika 14: Upit koji dohvaća identifikatore domena, broj lighthouse testova koji su provedeni i prosječnu ocjenu pristupačnosti navedenih domena.

Navedenim upitom i njegovom analizom doznajemo da domena s identifikatorom 115.481 je imala najviše lighthouse testova s prosječnom ocjenom od 79%. i provedenih 497 Lighthouse testova. Kao što je prikazano na priloženoj slici, domena s identifikatorom 3.665 nema izračunatu prosječnu ocjenu pristupačnosti i umjesto null vrijednosti prikazan je tekst „Nema rezultata“.

Uz prethodno napomenuto spajanje tablica primjenom tablice „spajanje“ za prikaz odnosa n:m tako možemo i prikazati ocjenu pristupačnosti izraženu u postocima; tako što primijenimo operator za množenje, primijeniti funkciju concat() za dodavanje '%' i dodavanja uvjeta da se prikaže tražena ocjena pristupačnosti (vidi Slika 15). Uz navedeno, postavljamo pitanje „Koje se ocjene pristupačnosti pojavljuju, a veće ili jednake su 95% i u koliko se domena pojavljuju?“.



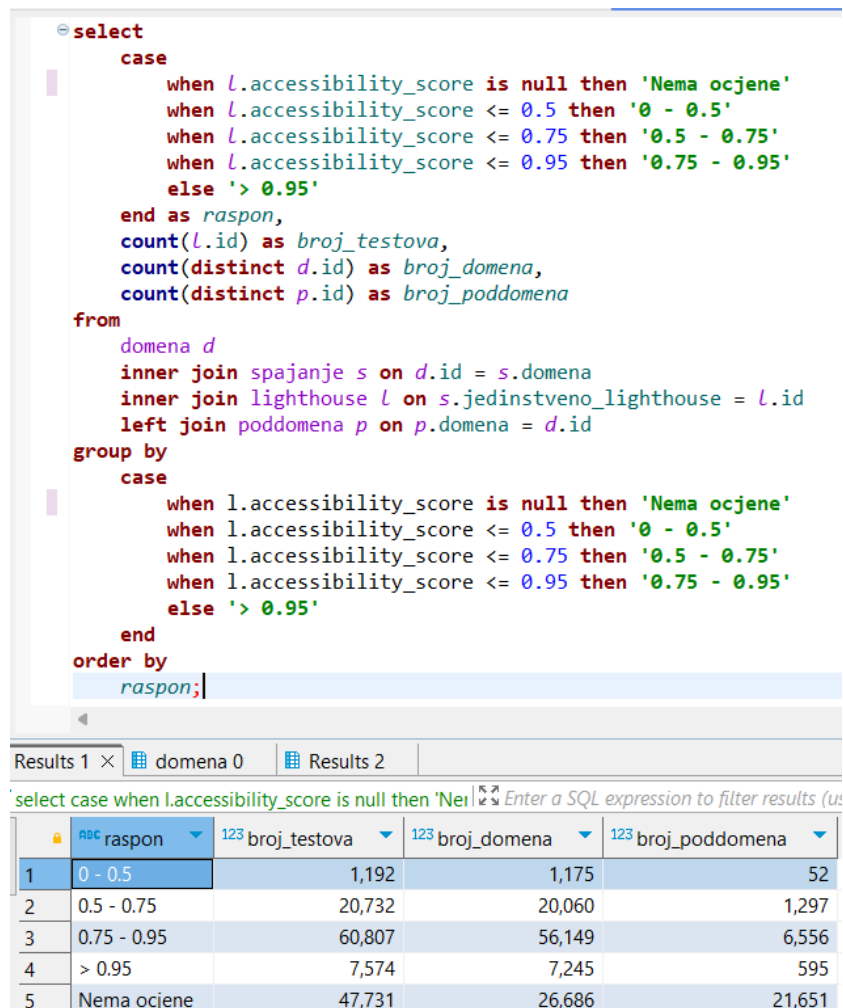
```
select
  concat((l.accessibility_score * 100), '%') as ocjena,
  count(distinct d.id) as broj_domena
from
  lighthouse l
  inner join spajanje s on l.id = s.jedinstveno_lighthouse
  inner join domena d on s.domena = d.id
where l.accessibility_score >= 0.95
group by
  l.accessibility_score
order by
  broj_domena desc;
```

Grid	ocjena	broj_domena
1	100.00%	4,809
2	95.00%	2,472
3	96.00%	1,418
4	98.00%	610
5	97.00%	280
6	99.00%	128

Slika 15: Upit koji dohvaća ocjenu pristupačnosti izraženu u postocima i broj domena u kojima se pojavljuje.

Navedenim upitom doznajemo da se ocjena pristupačnosti 100% najviše pojavljuje, a navedena ocjena se pojavljuje u 4.809 domena. Potom, ocjena 95% pojavljuje se u 2.472 domena. Najmanje se pojavljuje ocjena 99%, u svega od 128 domena.

Kako imamo mnoštvo ocjena pristupačnosti te je teško dohvatiti iste podatke, možemo kreirati slučajeve da dohvatimo raspon ocjena, te potom count() kako bi dohvatili broj lighthouse testova, broj domena i poddomena, koji spadaju u svoj zasebni raspon ocjena (vidi Slika 16). Možemo onda postaviti pitanje „Koliko je testova provedeno za različite intervale ocjene pristupačnosti u tablici lighthouse? Također koliki je broj domena i poddomena koje se nalaze u navedenim intervalima ocjene pristupačnosti?“.



Slika 16: Upit koji dohvaća broj domena i poddomena sa statusom 2, te njihovu prosječnu ocjenu pristupačnosti.

Navedenim upitom doznajemo da je najveći broj testova i domena unutar raspona od 0.75 do 0.95. Što znači da više od pola domena ima zadovoljavajuću ocjenu pristupačnosti, ako ne i više. Najveći broj poddomena spada unutar raspona „Nema ocjene“. Navedenim podatkom možemo zaključiti da većina poddomena ne zadovoljava Lighthouse testove koji su se vršili pri prikupljanju podataka, stoga nije se mogla izračunati ocjena pristupačnosti.



Nazivlje jedinstvenih poddomena možemo inkorporirati u složenije upite kako bi stekli bolji dojam o njihovoj vezi s drugim domenama, te statusom pripadajućih domena. Kako bi dobili opširniji upit i bolji dojam strukture domena i jedinstvenih poddomena, integrirat ćemo count() funkciju kako bi dobili broj domena za svaku pojedinačnu jedinstvenu poddomenu (vidi Slika 17).

```

select
  jp.id as jedinstvena_poddomena_id,
  count(distinct d.id) as broj_domena,
  d.status as status_domene
from
  jedinstvene_poddomene jp
  inner join poddomena_jedinstvene_poddomene pjp on pjp.jedinstvene_poddomene = jp.id
  inner join poddomena pd on pjp.poddomena = pd.id
  inner join domena d on pd.domena = d.id
group by
  jp.naziv, jp.status, d.status
order by
  2 desc
limit
  10;

```

jedinstvene\_poddomene(+) 1 ×

select jp.id as jedinstvena\_poddomena\_id, count(distinct d.id) as broj\_domena, d.status as status\_domene

	jedinstvena_poddomena_id	broj_domena	status_domene
1	7,488	6	2
2	29,287	5	2
3	1,984	5	2
4	29,283	5	2
5	1,988	5	2
6	29,292	5	2
7	29,288	5	2
8	29,289	5	2
9	1,986	5	2
10	29,293	5	2

Slika 17: Upit kojim doznajemo koja jedinstvena poddomena ima najviše domena.

Na temelju ovo upita doznajemo da jedinstvena poddomena s identifikatorom 7.488 ima najveći broj domena, odnosno njih 6; dok, ostale jedinstvene poddomene imaju 5 domena.

Daljnjom primjenom funkcije count() i spajanja tablica, možemo saznat koja domena na temelju vlastitog identifikatora, ima najveći broj stvarnih poddomena (vidi Slika 18). Postavljamo pitanje „Koje su top 5 domena na temelju broja stvarnih poddomena?“.

```

select
  d.id as domena_id,
  count(distinct pd.id) as broj_stvarnih_poddomena
from
  domena d
  left join domena_stvarnadomena ds on d.id = ds.domena
  left join stvarnadomena s on ds.stvarnadomena = s.id
  left join poddomena pd on s.id = pd.domena
group by
  d.id
order by
  broj_stvarnih_poddomena desc
limit
  5;

```

domena(+) 1 ×

select d.id as domena\_id, count(distinct pd.id) as broj\_stvarnih\_poddomena

	domena_id	broj_stvarnih_poddomena	naziv_poddomene
1	47,946	19,577	
2	71,230	1,323	
3	103,965	1,323	
4	50,016	902	
5	1,938	519	

Slika 18: Upit kojim doznajemo koje su top 5 domena sa najvećim brojem stvarnih poddomena.

Na temelju prikupljenih podataka, zaključujemo da domena s identifikatorom 47.946 ima najveći broj stvarnih poddomena u iznosu od 19.577.

## 7. Rasprava

Analiza je vršena nad skupom podataka od 129.602 domena. Prema novijim podacima, broj registriranih domena unutar .hr domene iznosi 133.567, što ukazuje na rast od 3.965 od siječnja 2024. godine do danas.<sup>49</sup> Najzastupljeniji status među domenama je status razriješene stvarne domene koji iznosi 96.630. Povezivanjem različitih objekata unutar baze podataka, dobivena je jasnija slika međusobne povezanosti domena i njihovih sastavnih dijelova. Tako je utvrđeno da domena s identifikatorom 115.481 ima najveći broj povezanih jedinstvenih domena, ukupno 7.

Uporabom Lighthouse rezultata, utvrđeno je da je ista domena (115.481) prošla kroz najviše testova pristupačnosti (497), s prosječnom ocjenom od 79%. Ova ocjena ukazuje na relativno dobru razinu, no također upućuje na postojanje određenih elementa koji nazaduju performansu i optimizaciju navedene domene, što također umanjuje korisničko iskustvo istoga. Stoga treba posebno obratiti pozornost na navedene uvjete i razriješiti ih. Dok, domena s identifikatorom 101.597 ima najvišu prosječnu ocjenu pristupačnosti (96%), a ima 142 provedena testa pristupačnosti. Ova ocjena ukazuje na odličnu optimizaciju domene za korisničko iskustvo. U kontrastu sa domenom koja ima najveći broj provedenih testova pristupačnosti (115.481), ova domena ima upola manje provedenih testova.

S druge strane, ocjena od 100% zastupljena je kod 4.809 domena, što znači da su navedene domene u potpunosti optimizirane za korisničko iskustvo i ne postoje nikakvi resursi koji usporavaju ili negiraju isto.

Daljnjom analizom, utvrđeno je da je najveći broj testova pristupačnosti, njih 60.870, završio s rezultatima unutar raspona od 75% do 95% (0.75-0.95). Ovo upućuje na zadovoljavajuću razinu pristupačnosti većine domena jer najveći broj domena prevladava u navedenom rasponu (59.149); međutim, 21.651 poddomena nije dobila nikakvu ocjenu (null vrijednost ili „Nema ocjene“). Ovo ukazuje da nisu zadovoljeni određeni preduvjeti i kriteriji Lighthouse testova pri prikupljanju istih rezultata, te teži potreba za dodatnim poboljšanjima kako bi se ostvarila cjelovita evaluacija pristupačnosti.

Analizom jedinstvenih poddomena, utvrđeno je da jedinstvena poddomena s identifikatorom 7.488 ima 6 domena sa statusom 2, što ukazuje na razrješenje stvarnih domena. Analizom broja

---

<sup>49</sup> Home. // domene.hr. URL: <https://domene.hr/portal/home> (2024-08-16)

poddomena zaključeno je da domena s identifikatorom 34.570 ima najveći broj stvarnih poddomena ukupno 19.577, što upućuje na bogat raspon stvarnih poddomena povezanih s tom domenom ili generički sustav generiranja poddomena u odnosu na sadržaj glavne domene.

Iz priloženih podataka vidljivo je da postoji značajna razlika u razini pristupačnosti i optimizacije među domenama, te da većina domena zadovoljava minimalne standarde pristupačnosti, uz iznimku da određeni dio zahtijeva dodatnu pažnju i prilagodbu. Također, rezultati ukazuju na aktivnu dinamičnost unutar .hr domene s obzirom na rast broja registracija i razrješenja stvarnih domena.

## 8. Zaključak

Ovaj završni rad pruža informacije o mrežnim mjestima unutar .hr domene kroz prizmu baze relacijske podataka. Praktičnom primjenom navedenog modela omogućen je pregled velikog skupa podataka i upravljanju istoga. Kroz prikaz strukture i sheme baze podataka putem ERA dijagrama, uporabom strukturiranog upitnog jezika za oblikovanje upita, te analiza povezanih tablica, rad je pokazao kako se relacijska baza podataka može učinkovito primijeniti za upravljanje i analiziranje velikih skupova podataka vezanih uz mrežna mjesta unutar .hr domene.

Dohvaćeni i analizirani podaci pružali su korisne uvide u brojnost i raspodjelu domena, poddomena, svih domena i jedinstvenih unutar .hr domene, njihov status i razinu pristupačnosti. Rezultati upućuju na to da većina domena zadovoljava osnovne kriterije pristupačnosti, postoje područja koja itekako zahtijevaju dodatna unaprjeđenja. Nadalje, povezivanjem podataka o domenama i rezultatima alata Lighthouse, stekao se bolji dojam o kvantiteti koja provedena kako bi se prikupio rezultat pristupačnosti i kojim ishodom je rezultirao. Rezultati testova upućuju na daljnje optimizacije domena i srodnog kako bi se unaprijedilo korisničko iskustvo i ukupna performansa.

Zaključno, ovaj rad dokumentira povijesni razvoj mrežnih mjesta i baza podataka, te pruža praktičan primjer kako se relacijski model baze podataka može učinkovito primijeniti za složeno rudarenje i analizu podataka unutar specifičnog područja, kao što je .hr domena. Primjenom navedenog modela osigurava se strukturiranost, lako dostupnost i konzistentnost podataka; što ga čini prikladnim alatom za upravljanje i analizu kompleksnih podatkovnih skupova.

## Literatura

1. Aced, Cristina. Web 2.0: the origin of the word that has changed the way we understand public relations. / Barcelona International PR Conference, 2013. Str. 1-2.
2. CSS / Mozilla Developer. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (2024-06-11)
3. Data analysis. Eldridge, Stephen. // Britannica. URL: <https://www.britannica.com/science/data-analysis> (2024-07-05)
4. DBeaver. What is DBeaver? // Database.Guide. URL: <https://database.guide/what-is-dbeaver/> (2024-06-27)
5. Dietrich W., Suzanne; Urban D., Susan. Fundamentals of Object Databases. Morgan & Claypool Publishers, 2011. Str. 3.
6. Eldridge, Stephen. Data analysis. // Britannica. URL: <https://www.britannica.com/science/data-analysis> (2024-07-05)
7. Gillis, Alexander; Stedman, Craig; Hughes, Adam. Data mining. // TechTarget. URL: <https://www.techtarget.com/searchbusinessanalytics/definition/data-mining> (2024-07-05)
8. Halonja, Antun; Mihaljević, Milica. Nazivlje računalnih mreža. // Institut za hrvatski jezik i jezikoslovlje, 29/1 (2003), str. 88.
9. Harrington, Jan, L. Relational Database Design and Implementation, Burlington, MA, USA: Morgan Kauffman. 2009. Str. 61-99.
10. History of the Web. // World Wide Web Foundation. URL: <https://webfoundation.org/about/vision/history-of-the-web/> (2024-06-10)
11. Home. // domene.hr. URL: <https://domene.hr/portal/home> (2024-08-16)
12. HTML // Mozilla Developer. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (2024-06-10)
13. How CSS works. // Mozilla Developer. URI: [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/How\\_CSS\\_works](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works) (2024-06-11)
14. IBM. Introduction to DB2 for z/OS. // URL: <https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-introduction-db2> (2024-06-12)
15. Introduction to DB2 for z/OS. // IBM. URL: <https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-introduction-db2> (2024-06-12)

16. Lighthouse. // Chrome for Developers. URL: <https://developer.chrome.com/docs/lighthouse/overview> (2024-07-03)
17. Manger, Robert. Baze podataka. Drugo izdanje. Zagreb: 2001. Str. 9-44.
18. MariaDB Server. // MariaDB. URL: <https://mariadb.org/en/> (2024-06-12)
19. MySQL. What is MySQL? // MySQL. URL: <https://dev.mysql.com/doc/refman/8.4/en/what-is-mysql.html> (2024-06-12)
20. Ottervig, Vegard. Websites: past and present, 2022. URL: <https://www.enonic.com/blog/websites-past-and-present> (2024-06-11)
21. Pol, Tushar. Google Lighthouse: What It Is & How to Use It?. // Semrush Blog. URL: <https://www.semrush.com/blog/google-lighthouse/> (2024-08-21)
22. SQLServer Tutorial.net. What is SQL Server? // URL: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> (2024-06-12)
23. SooHoo, Kaela. The role of HTML5 and Flash in Web Design. // College of Liberal Arts, 2012. URL: <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1081&context=grcsp> (2024-06-11)
24. Šojat-Bikić, Maja. Web 2.0? Da, ali nakon Weba 1.0. / Informatica museologica, 42 (1-4), 2011. Str. 113.
25. Što je baza podataka? // Oracle. URL: <https://www.oracle.com/hr/database/what-is-database/> (2024-06-11)
26. Što je Oracle i za što radi? Kako ga razviti? // Vida Bytes. URL: <https://vidabytes.com/hr/%C5%A1to-je-proro%C4%8Di%C5%A1te/> (2024-06-12)
27. The Evolution of Web Design: A Historical Perspective. // emd:digital. URL: <https://www.emddigital.com.au/blog-posts/the-evolution-of-web-design-a-historical-perspective> (2024-06-11)
28. URI // Mozilla Developer. URL: <https://developer.mozilla.org/en-US/docs/Glossary/URI> (2024-06-10)
29. Understanding the Hierarchical Database Model. // MariaDB. URL: <https://mariadb.com/kb/en/understanding-the-hierarchical-database-model/> (2024-06-12)
30. Understanding the Network Database Model. // MariaDB. URL: <https://mariadb.com/kb/en/understanding-the-network-database-model/> (2024-06-12)

31. Vida Bytes. Što je Oracle i za što radi? Kako ga razviti? // URL: <https://vidabytes.com/hr/%C5%A1to-je-proro%C4%8Di%C5%A1te/> (2024-06-12)
32. Websites: past and present, 2022. Ottervig, Vegard. URL: <https://www.enonic.com/blog/websites-past-and-present> (2024-06-11)
33. What is a UML diagram? // Miro. URL: <https://miro.com/diagramming/what-is-a-uml-diagram/#benefits-of-uml-diagrams-> (2024-06-24)
34. What is a relation database? // URL: <https://cloud.google.com/learn/what-is-a-relational-database> (2024-08-14)
35. What is a NoSQL database? // URL: <https://cloud.google.com/discover/what-is-nosql> (2024-08-14)
36. What is MySQL? // MySQL. URL: <https://dev.mysql.com/doc/refman/8.4/en/what-is-mysql.html> (2024-06-12)
37. What is Oracle i za što radi? Kako ga razviti? Vida Bytes. URL: <https://vidabytes.com/hr/%C5%A1to-je-proro%C4%8Di%C5%A1te/> (2024-06-12)
38. What is SQL Server? // SQLServer Tutorial.net. URL: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> (2024-06-12)
39. What is XAMPP? // Educba. URL: <https://www.educba.com/what-is-xampp/> (2024-06-27)