SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1713

OBLIKOVANJE SUSTAVA ZA PRAĆENJE I ANALIZU PERFORMANSI VESLAČA

Zlatko Hrastić

Zagreb, lipanj 2018.

**Sadržaj**

[1. Uvod 1](#_Toc516992671)

[2. Glavne karakteristike veslačkih natjecanja 2](#_Toc516992672)

[2.1 Starosne kategorije 2](#_Toc516992673)

[2.2 Vrste čamaca 3](#_Toc516992674)

[2.3 Kategorije prema težini 4](#_Toc516992675)

[3. Definicija zahtjeva na sustav 5](#_Toc516992676)

[3.1 Definicija zahtjeva izbornika reprezentacije 5](#_Toc516992677)

[3.2 Definicija zahtjeva trenera 7](#_Toc516992678)

[3.3 Definicije zahtjeva veslačkih klubova 8](#_Toc516992679)

[3.4 Definicija zahtjeva organizatora natjecanja 8](#_Toc516992680)

[4. Prikupljanje podataka 9](#_Toc516992681)

[4.1 Odabir tehnologije i načina prikupljanja podataka 9](#_Toc516992682)

[4.2 Aplikacija za prikupljanje podataka 10](#_Toc516992683)

[4.3 Interakcija s bazom podataka 13](#_Toc516992684)

[5. Modeliranje baze podataka 15](#_Toc516992685)

[5.1 Relacije 15](#_Toc516992686)

[5.2 Pogledi 21](#_Toc516992687)

[5.3 Imenovanje objekata 26](#_Toc516992688)

[5.4 Sustav za upravljanje bazom podataka 27](#_Toc516992689)

[6. Vizualizacija podataka 29](#_Toc516992690)

[6.1 Semantika i tipovi podataka 30](#_Toc516992691)

[6.2 Vizualizacija vremenski zavisnih podataka 31](#_Toc516992692)

[6.3 Vizualizacija geoprostornih podataka 32](#_Toc516992693)

[6.4 Vizualizacija temeljena na broju ključeva 35](#_Toc516992694)

[6.4.1 Vizualizacija podataka s jednim ključem 35](#_Toc516992695)

[6.4.2 Vizualizacija podataka s dva ključa 39](#_Toc516992696)

[6.5 Vizualizacija podataka korištenjem tablica 41](#_Toc516992697)

[6.6 Boja u grafovima 42](#_Toc516992698)

[6.6.1 Prostori boja 42](#_Toc516992699)

[6.6.2 Uloga boje u vizualizaciji 44](#_Toc516992700)

[6.7 Vizualizacija uz pomoć alata Power BI 46](#_Toc516992701)

[7. Zaključak 50](#_Toc516992702)

[8. Literatura 51](#_Toc516992703)

# Uvod

Veslanje je sport u kojem se čamac pokreće uz pomoć vesala(1). U veslanju se za pokretanje čamca koristi cijelo tijelo uključujući ruke, leđa i noge. Veslanje je vrlo zdrav sport koji zahtjeva visoku fizičku i mentalnu spremu. Profesionalni veslači moraju proći rigorozne treninge kako bi bili uspješni u veslanju. Zbog ovakvih uvjeta treneri moraju vremenski planirati fizičku spremu veslača i uspoređivati je s drugim veslačima kako bi mogli prilagoditi treninge. Prilikom planiranja fizičke spreme veslača najčešće se veslača trenira kako bi utrku u određenom dijelu godine odveslao u planiranom vremenu. Ovo je proces koji traje nekoliko mjeseci pa čak i preko par godina. U tu svrhu trenerima u Hrvatskoj bi od velike koristi bio sustav koji bi mogao čuvati i analizirati podatke njihovih veslača te ih uspoređivati s drugim veslačima. Ovakav sustav u Hrvatskoj trenutno ne postoji. Iz tog razloga potrebno je izraditi sustav za praćenje i analizu performansi veslača. Ovaj sustav mora ispuniti zahtjeve ljudi na različitim pozicijama veslačke struke. Naglasak sustava treba biti na efektivnom i olakšanom pristupu informacijama. Također je potrebno analizirati prikupljene podatke i prikazati ih na razumljiv način.

# Glavne karakteristike veslačkih natjecanja

Veslačko natjecanje naziva se regata. Regate se održavaju na lokacijama koje je odobrio nadležni veslački savez. Na regatama sudjeluju veslači i posade klubova koji su pred zakonom registrirani kao veslački klubovi. Veslačka regata je natjecanje kada se sastoji od jedne ili više disciplina podijeljenih prema potrebi u više utrka, u jednom ili više tipova čamaca, za veslače podijeljene u različite kategorije prema spolu, starosti ili težini(3).

## Starosne kategorije

Različita natjecanja mogu imati raspisane različite starosne kategorije. U pravilniku Hrvatskog veslačkog saveza postoje 4 starosne kategorije: kadeti, juniori, seniori i veterani.

Starosna kategorija kadeta raspodijeljena je na 2 potkategorije: mlađi kadeti (kadeti B) i kadeti (kadeti A). U potkategoriju mlađih kadeta spadaju veslači koji do 31. prosinca, godine u kojoj se održava regata, navrše 12 godina. Veslači spadaju u potkategoriju kadeta ako do 31. prosinca, godine u kojoj se održava regata navrše 14 godina. Najduža dopuštena staza za mlađe kadete je 500 metara, a za kadete ona iznosi 1000 metara.

Veslači i kormilari koji više nisu kadeti, smatraju se juniorima i to do 31. prosinca godine u kojoj navršavaju 18 godina života(3). Ova starosna kategorija također ima dvije potkategorije, a to su: mlađi juniori (juniori B) i juniori (juniori A). Veslači spadaju u kategoriju mlađih juniora ako do 31. prosinca navrše 16 godina, inače spadaju u kategoriju juniora. Klasična duljina staze za ovu i naredne kategorije je 2000 metara.

Veslač koji nije junior klasificirat će se kao senior «U23» do 31. prosinca godine u kojoj navršava 22 godine života. Po isteku godine u kojoj je navršio 22 godine života veslač se kategorizira kao senior. Kategorija „senior U23“ ne koristi se ni na jednom službenom natjecanju u Hrvatskoj.

Posljednja starosna kategorija je kategorija veterana. U kategoriji veterana mogu se natjecati veslači koji u godini natjecanja navršavaju 27 godina. Kategorija veterana podijeljena je na 11 potkategorija koje se označavaju slovima A-K a svaka predstavlja veslače određene starosti.

Sve navedene starosne kategorije jednako se primjenjuju i na muške i na ženske posade. Kategorije su uz starost raspodijeljene i prema spolu.

## Vrste čamaca

Hrvatski veslački savez priznaje sljedeće veslačke regatne čamce po disciplinama: samac, dvojac na pariće, dvojac bez kormilara, dvojac s kormilarom, četverac na pariće, četverac bez kormilara, četverac s kormilarom i osmerac.

Samac je čamac u kojem se nalazi 1 osoba. Ta osoba vesla s 2 vesla. Oznaka čamca samac je „1x“. Dvojac na pariće je čamac u kojem se nalaze 2 osobe, svaka s po 2 vesla. Oznaka čamca je „2x“. Dvojac bez kormilara je čamac u kojem se nalaze 2 osobe, svaka s po 1 veslom. Oznaka čamca je „2-“. Dvojac s kormilarom je čamac u kojem se nalaze 3 osobe. Od te 3 osobe 2 su veslača i 1 kormilar. Svaki veslač ima po 1 veslo. Oznaka čamca je „2+“. Četverac na pariće je čamac u kojem se nalaze 4 osobe svaka s po 2 vesla. Oznaka čamca je „4x“. Četverac bez kormilara je čamac u kojem se nalaze 4 osobe, svaka s po 1 veslom. Oznaka čamca je „4-“. Četverac s kormilarom je čamac u kojem se nalazi 5 osoba. Od tih 5 osoba 4 su veslača i 1 kormilar. Svaki veslač ima po 1 veslo. Oznaka čamca je „4+“. Osmerac je čamac u kojem se nalazi 9 osoba. Od tih 9 osoba 8 je veslača i 1 kormilar. Svaki veslač ima po 1 veslo. Oznaka čamca je „8+“.

Svaki čamac ima propisane određene odredbe koje mora zadovoljavati, a te su odredbe raspisane od strane svjetske veslačke federacije FISA (franc. Fédération Internationale des Sociétés d’Aviron)(4).

## Kategorije prema težini

Kategorije prema težini koriste se samo na većim Hrvatskim i međunarodnim natjecanjima koja se održavaju u Hrvatskoj. Podjela prema težini postoji samo u kategorijama juniora i seniora te juniorki i seniorki.

U kategoriji seniora prosječna težina muške posade (bez kormilara) ne smije biti veća od 70 kg, pri čemu pojedini veslač ne može biti teži od 72,5 kg(3). Samac ne može težiti više od 72,5 kg. Prosječna težina ženske posade (bez kormilarke) ne smije biti veća od 57,5kg, pri čemu pojedina veslačica ne može biti teža od 59 kg. Samac (seniorka) ne može težiti više od 59 kg. Veslači se vežu u veslačkom trikou i to ne manje od jednog i ne više od dva sata prije prve utrke dana.

U kategoriji juniora raspisuju se samo kategorije „samac za lake juniore“ i „samac za lake juniorke“. Dopuštena maksimalna težina lakih juniora je 67,5 kg za veslače i 57,5 kg za veslačice.

# Definicija zahtjeva na sustav

Sustav za praćenje i analizu performansi veslača mora zadovoljiti zahtjeve postavljene od strane osoba koje se nalaze u različitim ulogama unutar veslanja. Sustav mora zadovoljiti zahtjeve izbornika reprezentacije, trenera i organizatora natjecanja. S obzirom na to da je potrebno zadovoljiti zahtjeve s tri različite zainteresirane strane potrebno je izlučiti najbitnije zahtjeve svake strane i integrirati ih u zajedničko rješenje.

## Definicija zahtjeva izbornika reprezentacije

Izbornik reprezentacije je osoba koja prati sve regate kupa u Hrvatskoj. Također prati i prvenstvo Republike Hrvatske te sve međunarodne regate gdje veslaju natjecatelji iz Hrvatske. Njegova je uloga uočiti pojedince koji se ističu na treninzima ili natjecanjima i pokušati ih pripremiti za natjecanja gdje će umjesto kluba predstavljati Hrvatsku reprezentaciju. Izbornik također mora kombinirati veslače iz različitih klubova kako bi pokušao dobiti optimalne posade za međunarodna natjecanja.

Proces kojim započinje odabir veslača za reprezentaciju je proces pregleda vremena na svakoj pojedinoj regati u Hrvatskoj. Zbog toga je u sustavu potrebno osigurati lagan i intuitivan pregled izveslanih vremena za svaku utrku unutar svake regate u Hrvatskoj. Za utrku se mora moči odrediti je li ona kvalifikacijska ili finalna utrka. Također vremena posada moraju biti lako preglediva i bitne su i najmanje razlike u izveslanim vremenima. Za pregled vremena unutar pojedine utrke vrlo je bitno da je moguće vidjeti tko je u posadi koja je veslala traženo vrijeme.

Nakon što se određeni veslač iskazao u jednoj ili više različitih utrka potrebno je vidjeti detalje o veslanju tog veslača. Najbitniji slučaj korištenja je pregled svih rezultata koje je veslač postigao. Iz ovih rezultata mora biti vidljivo gdje je rezultat ostvaren i unutar koje kategorije. Vrlo bitna informacija koja mora biti vidljiva je u kojoj kategoriji se veslač najčešće natječe kako bi se znalo u kojim čamcima ostvaruje najbolje rezultate. U slučaju da veslač ne vesla u samcu potrebno je znati s kime je veslač veslao i koliko često. Ova informacija je potrebna kako bi se veslači koji veslaju veće čamce mogli kao grupa testirati za veslanje u reprezentaciji.

Kako bi izbornik imao pristup određenom veslaču potrebno je znati gdje se on trenutno nalazi. Najlakši način za kontaktiranje veslača je kontaktiranje preko njegovog matičnog kluba. Iz ovog razloga je potrebno znati kojem klubu veslač trenutno pripada. Također bi trebalo moći pratiti kojim klubovima je veslač prije pripadao kako bi se mogao pratiti njegov veslački put.

U slučaju da je veslač izabran za pristup reprezentaciji mora biti moguće pratiti rezultate standardiziranih treninga i testova koje izbornik propisuje. Na jednom mjestu moraju biti vidljivi rezultati svih standardiziranih testova kroz vrijeme. Na temelju pojedinih standardiziranih testova moguće je napraviti predviđanje regatnih vremena koja će biti izveslana na vremenski bliskim regatama. Ovi podaci moraju također biti vidljivi. Standardizirani testovi i treninzi koje je potrebno pratiti su: predikcijski treninzi na 2000 metara i 500 metara te intervalni trening na dionicama od 8000, 6000 i 4000 metara.

Predikcijski trening na 2000 metara uključuje veslanje dvije dionice duljine 2000 metara. Oba vremena moraju biti vidljiva kog prikaza rezultata treninga. Na temelju bržeg rezultata računa se predviđeno vrijeme na sljedećem natjecanju na način da se na rezultat doda vrijednost od 10 sekundi.

Predikcijski trening na 500 metara uključuje veslanje jedne dionice od 250 metara i 2 dionice od 500 metara. Vrijeme dionice od 250 metara se ne treba prikazivati dok je prikaz 2 dionice od 500 metara bitan. Na temelju ovog treninga također se predviđa vrijeme iduće utrke. Vrijeme se predviđa na način da se na vrijeme izveslano na drugoj dionici od 500 metara doda 7 sekundi i zatim se dobiveno vrijeme pomnoži s 4 kako bi se skaliralo na standardnu duljinu staze od 2000 metara.

Intervalni trening na dionicama od 8000, 6000 i 4000 metara je trening na kojem se prati vrijeme svake izveslane dionice. Dionice moraju biti veslane jedna iza druge. Kako bi se lakše usporedili rezultati po dionicama na ovom treningu ne uspoređuje se ukupno vrijeme po pojedinoj dionici. Za ovaj trening potrebno je usporediti prosječno izveslano vrijeme na 500 metara za svaku dionicu. To se vrijeme dobiva prema fromuli x.

Gdje je prosječno izveslano vrijeme na 500 m, je ukupno vrijeme izveslano na pojedinoj dionici, a je duljina pripadne dionice u metrima. Prosječno i ukupno vrijeme u teoriji mogu biti u bilo kojoj vremenskoj jedinici ali se u praksi vrijeme prikazuje kao kombinacija minuta, sekundi, desetinki i stotinki.

## Definicija zahtjeva trenera

Zahtjevi trenera djelomično se preklapaju sa zahtjevima izbornika reprezentacije. Treneri također trebaju prikaz detalja o veslanju pojedinog veslača. Ovi detalji uključuju izveslana vremena, veslače s kojima je određeni veslač bio u posadi i kategorije u kojima je veslač veslao. Ovaj zahtjev se u potpunosti poklapa sa zahtjevom izbornika reprezentacije. Zahtjev je opisan u poglavlju 3.1.

Treneri također žele pratiti rezultate treninga i standardiziranih testova. Treneri ne žele javno dijeliti detalje o svojim treninzima i testovima. Iz ovog razloga ovaj zahtjev moguće je objediniti sa zahtjevom o treninzima i testovima koji je definirao izbornik. Zahtjev je opisan u poglavlju 3.1.

Treneri žele pratiti fizičke karakteristike veslača. Najinteresantnije karakteristike su visina i težina veslača. Također bi uz prikaz težine htjeli vidjeti granicu težine za ulazak veslača u kategoriju lakih veslača. Visina veslača je bitna kako bi se mogao pratiti tjelesni razvoj mlađih veslača i kako bi se u čamce mogli grupirati veslači sličnih tjelesnih karakteristika.

Treneri također žele praćenje lokacija gdje su veslači veslali. Zanima ih činjenica koliko je puta određeni veslač veslao na određenoj stazi. Ovaj podatak je potrebno prikazati na karti kako bi treneri znali je li veslač sudjelovao samo na regatama u Hrvatskoj ili je sudjelovao i na međunarodnim regatama izvan zemlje.

## Definicije zahtjeva veslačkih klubova

Zahtjevi veslačkih klubova predstavljaju zahtjeve osoblja u klubovima koje je zaduženo za nadgledanje veslača i rezultata kao skupine, a ne pojedinih veslača.

Glavni zahtjev veslačkih klubova je praćenje broja veslača registriranih u klubu. Potrebno je poznavati povijest broja veslača po sezonama gdje je najbitnije znati razliku broja veslača na početku i na kraju sezone. Također je potrebno znati trenutnu raspodjelu veslača prema starosnoj kategoriji kojoj pripadaju.

Uz starost veslača klub zanima povijest osvojenih medalja kako bi se znao generalni napredak kluba kroz godine.

## Definicija zahtjeva organizatora natjecanja

Organizatore natjecanja zanimaju grupni podaci o klubovima i kategorijama na regatama. Ovi podaci su bitni kako bi se znalo koje kategorije je potrebno poticati na razvoj. Također je bitno vidjeti koji su klubovi bili na regati i s kojim brojem veslača kako bi se klubove s manje veslača potaknulo na više prijava na idućim regatama.

Organizatore natjecanja također zanima koje su se kategorije i čamci veslali u zadnjoj godini na svim veslačkim natjecanjima kako bi se uočile kategorije koje su se manje veslale i kako bi se potaknuo njihov razvoj.

# Prikupljanje podataka

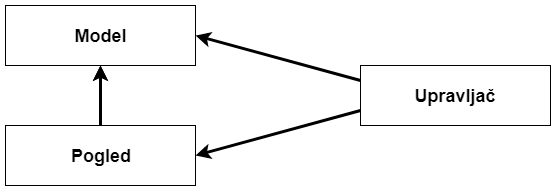
## Odabir tehnologije i načina prikupljanja podataka

Prikupljanje podataka za sustav za analizu i praćenje performansi veslača potrebno je obaviti upisivanjem u sustav od strane čovjeka. Podaci o veslačima i veslačkim natjecanjima javno su dostupni samo u obliku tekstualnih dokumenata u polu uređenom obliku. Dokumenti s rezultatima prate određenu propisanu formu, ali ova se forma često krši i zbog toga je podatke nemoguće prikupiti na temelju propisane forme dokumenata. U javno dostupnom obliku podataka veslači se identificiraju samo imenom i prezimenom koji često sadrže zatipke. Iz ovog razloga vrlo je teško napraviti automatizirani sustav koji će prikupiti podatke o veslačima i natjecanjima.

U svrhu prikupljanja podataka za analizu napravljena je webovska aplikacija. Ovaj način prikupljanja podataka odabran je kako bi se podaci mogli skupljati iz više lokacijski različitih izvora. Također sva poslovna logika potrebna za prikupljanje i pohranu podataka se u ovom načinu prikupljanja podataka nalazi na jednom mjestu.

S obzirom na odabrani način prikupljanja podataka, za tehnologiju implementacije webovske aplikacije odabrana je platforma *.NET.* Platforma *.NET* je platforma za razvoj aplikacije tvrtke Microsoft. Ova platforma omogućuje razvoj webovskih aplikacija koje koriste obrazac model-pogled-upravljač (*engl. model-view-controller*) poznatiji pod nazivom *MVC*. Obrazac *MVC* je arhitekturni obrazac koji razdvaja aplikaciju u tri glavne logičke komponente: model, pogled i upravljač(izvor MVC).

Model predstavlja svu logiku vezanu uz podatke s kojom korisnik mora raditi. Pogled predstavlja vizualnu prezentaciju modela te je točka u kojoj se prikupljaju podaci koji se kasnije obrađuju i pohranjuju u modelu. Upravljač je komponenta koja služi za komunikaciju između modela i pogleda. Upravljač je zadužen za pravilno prosljeđivanje podataka od pogleda do modela i za transformaciju podataka iz modela za pravilan prikaz u pogledu. Logička povezanost ove tri komponente nalazi se na slici x.



Slika Logička povezanost komponenti modela MVC

Strelica na slici označava ovisnost komponente iz koje izvire strelica o komponenti prema kojoj pokazuje.

Model je neovisan od pogleda i upravljača i nije logički vezan ni za jednu od druge dvije komponente. Pogled je ovisan o modelu i mora ga poznavati kako bi mogao pravilno prikazati i prikupiti podatke. Upravljač je ovisan i o modelu i o pogledu te koristi obje komponente kako bi omogućio pravilan rad aplikacije.

## Aplikacija za prikupljanje podataka

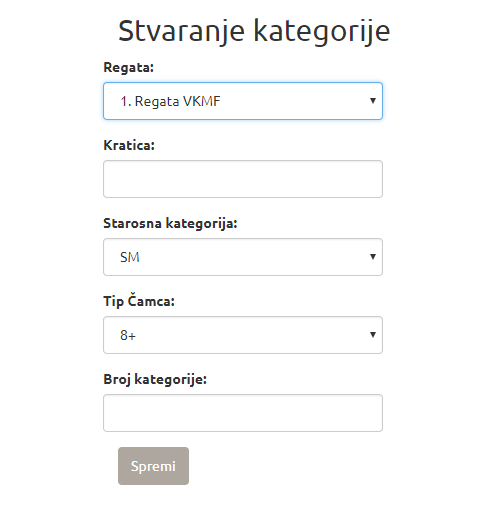
Aplikacija za unos podataka sastoji se od ekrana za unosa podataka u obliku formi i alatne trake koja omogućava prijelaz između ekrana za unos podataka. Alatna traka grupirana je prema logičkim cjelinama kako bi omogućila olakšanu navigaciju (Slika 2).



Slika Alatna traka aplikacije za unos podataka

U logičku cjelinu vezanu uz veslačke kategorije spadaju stvaranje kategorija i registracija veslačkih posada za veslanje u pojedinim kategorijama. U cjelinu vezanu uz utrke spadaju stvaranje utrka, dodavanje postojećih posada u utrke i upis i spremanje rezultata pojedine utrke. U cjelinu vezanu uz veslače spadaju stvaranje veslača te unos visine i težine za pojedine veslače. U cjelinu vezanu uz klubove spadaju stvaranje klubova i registracija veslača za određeni klub, to jest unos vremena kada je određeni veslač veslao za jedan od registriranih klubova. Cjeline za unos podataka o regatama i treninzima nemaju podcjelina.

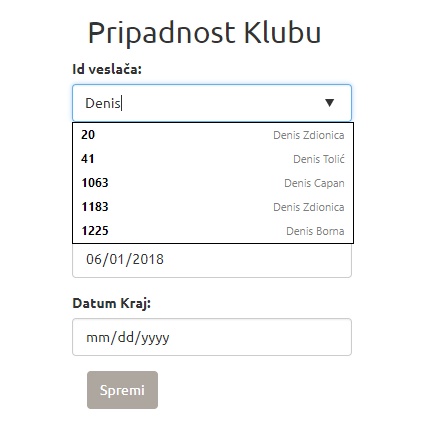
Aplikacija se sastoji od većeg broja formi za unos podataka. Primjer forme za unos podataka je forma za stvaranje nove kategorije(Slika 3).



Slika Forma za stvaranje veslačke kategorije

Forma za stvaranje kategorije sadrži polja za unos tekstualnog i brojčanog tipa. Na ovoj formi su sva polja za unos obavezna. Polja koja se kod stvaranja kategorije moraju referencirati na druge objekte iz modela predstavljena su padajućim izbornicima. Padajući izbornici omogućuju izbor već unesenih podataka na koje se trenutna forma referencira. Kod predavanja podataka iz pogleda u model, u model se ne upisuju tekstualni podaci iz padajućih izbornika već njihovi odgovarajući ključevi. Ovakav način unosa referenciranih podataka je prikladan kada referencirani podatak sadrži manji broj različitih vrijednosti.

U slučaju da je prilikom unosa podataka potrebno referencirati objekt koji može poprimiti više različitih vrijednosti potrebno je koristiti liste podataka koje je moguće filtrirati. Primjer ovakvog unosa podataka je unos podataka gdje se referencira pojedini veslač (Slika 4).



Slika Forma za unos pripadnosti veslača klubu

Na slici je moguće vidjeti listu podataka s imenima veslača koja je filtrirana prema znakovima koji su uneseni u polje za unos veslača. Odabirom pojedine vrijednosti iz liste podataka forma za unos pripadnosti klubu referencira odabranog veslača.

## Interakcija s bazom podataka

Interakcija s bazom podataka odvija se uz pomoć programske knjižnice *„Dapper“.* Knjižnica *„Dapper“* je knjižnica za objektno relacijsko preslikavanje (*engl. ORM*). Ova knjižnica spada u skupinu proizvoda pod nazivom „*micro ORM*“. Posebnost ove skupine knjižnica je ta što ne nude mogućnosti automatskog stvaranja SQL upita. Svi SQL upiti koji se koriste za interakciju s bazom podataka moraju biti spremljeni u aplikaciji ili u bazi podataka u obliku procedura. Knjižnica *„Dapper“* omogućava lakši unos parametara i lakši dohvat podataka od upita korištenjem *„ADO.NET“* knjižnice koja je preporučena knjižnica za „*.NET“* programsko okruženje. Primjer upita korištenjem knjižnice *„Dapper“* nalazi se u odsječku koda 1.

public SelectDTO DohvatiRegatu(int idRegata)

{

string query = "SELECT [IdRegata] AS Id,[Ime] AS Value

FROM [dbo].[Regata] WHERE IdRegata=@idRegata";

using (var connection = new SqlConnection(Konekcija))

{

return connection.Query<SelectDTO>

(query, new { idRegata }).FirstOrDefault();

}

}

*Odsječak koda 1 Dohvat podataka korištenjem knjižnice Dapper*

Knjižnica *„Dapper“* za interakciju s bazom podataka koristi standardni obrazac. Prvo je potrebno stvoriti konekciju na bazu podataka na način da se knjižnici predaju podaci o bazi podataka na koju je potrebno ostvariti konekciju. U ovom slučaju podaci se nalaze u varijabli *Konekcija.* Upiti nad bazom podataka izvršavaju se jednom od naredbi: *„Query“, „QuerySingle“* ili *„Execute“.*  Naredba *„Query“* koristi se dohvaćanje podataka. Naredba *„QuerySingle“* koristi se kada je potrebno dohvatiti jedan podatak primitivnog tipa. Ovo naredba najčešće se koristi kada je potrebno dohvatiti ključ novo unesenog retka. Naredba *„Execute“* koristi se kada je potrebno unesti ili izmijeniti podatke u bazi podataka. Parametri koji se predaju SQL upitu u funkciju se predaju u obliku objekta čiji nazivi atributa odgovaraju oznakama koje se postavljene u SQL upit. Oznake je moguće prepoznati prema prefiksu „@“. U odsječku koda 1 primjer oznake u SQL upitu je „@idRegata“.

Posebnost knjižnice *„Dapper“* je preslikavanje rezultata u objekte. Kao ulaz u funkcije za interakciju s bazom podataka potrebno je predati tip podataka u koji će se obaviti preslikavanje dobivenih rezultata. U ovom primjeru radi se o tipu „SelectDTO“. Knjižnica *„Dapper“* preslikavanje dobivenih ntorki iz baze podataka obavlja na sjedeći način. Svaka ntorka preslikava se u poseban objekt, objekti se zatim slažu u listu i u tom formatu se vraćaju korisniku. Pojedina ntorka u objekt se preslikava tako da se u svojstvo objekta u programu upisuje vrijednost istoimenog atributa iz ntorke. Na ovaj način zaobilazi se proceduralno čitanje svakog atributa i njegovo dodavanje u objekt. Uz jednostavniju implementaciju prednost ovog pristupa je smanjena krhkost koda. U slučaju promjene tipa atributa u bazi potrebno je napraviti samo jednu promjenu u programskom kodu kako bi se komunikacija s bazom i dalje nesmetano odvijala.

# Modeliranje baze podataka

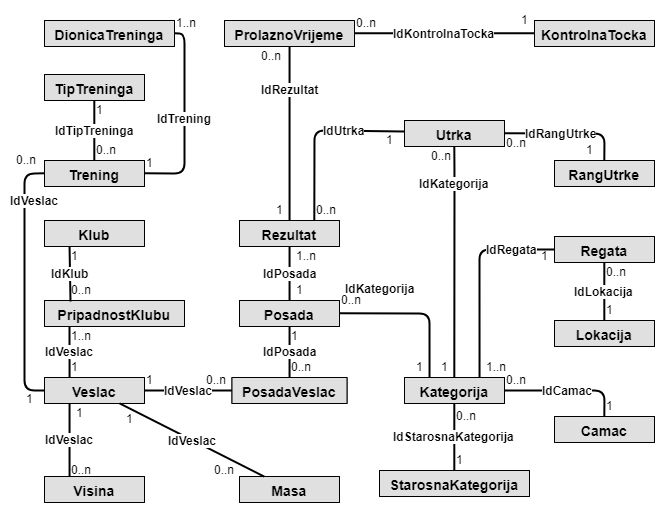
## Relacije

Relacija *Veslac* predstavlja zapis o pojedinom veslaču koji je učlanjen u određeni klub. Podaci koji se zapisuju su ime, prezime, datum rođenja i osobni identifikacijski broj (OIB). Atributi *DatumRodenja* i *OIB* nisu obavezni atributi jer nisu poznati za sve veslače. U teoriji bi ovi podaci trebali biti obavezni. Atribut *OIB* je tipa nchar(11) koji predstavlja niz znakova stalne duljine od 11 znakova. Atribut *OIB* je ovoga tipa jer je osobni identifikacijski broj podatak dugačak 11 znakova. Kada bi se moglo osigurati da je atribut OIB uvijek poznat ovaj bi atribut predstavljao prirodni ključ i na njega bi bilo potrebno postaviti ograničenje jedinstvenosti (engl. unique)(2-pogledaj opasku).

Relacija *Camac* sastoji se od sljedećih atributa: *IdCamac, Oznaka, Ime* i *BrojLjudi.* *IdCamac* je cjelobrojni tip podatka i predstavlja umjetni primarni ključ kako bi relacija mogla biti referentna relacija za strane ključeve. *Oznaka* predstavlja niz znakova koji označavaju pojedinu vrstu čamca (npr. 4X). Kao tip podatka *Oznaka* je niz znakova maksimalne duljine 10 znakova. Prosječna duljina ovog polja je 2 znaka, međutim zbog posebnih oznaka određenih čamaca u veslanju potrebno je za oznaku čamca rezervirati više od 2 znaka. Atribut *Ime* predstavlja puni naziv vrste čamca (npr. Četverac na pariče). Atribut *BrojLjudi* predstavlja broj ljudi koje se mogu nalaziti u čamcu uključujući veslače i kormilara. Ovo je cjelobrojni tip podatka koji za spremanje koristi samo 1 bajt (1.) jer u praksi može dostići maksimalno broj 9.

Relacija *StarosnaKategorija* sastoji zapise o kategorijama u kojima se veslači natječu podijeljeni prema starosti. Atribut *IdStarosnaKategorija* predstavlja umjetni primarni ključ relacije. Atribut *Oznaka* je niz znakova koji predstavljaju kraticu starosne kategorije (npr. KA). Atribut *Ime* je neograničeni niz znakova i predstavlja puni naziv starosne kategorije (npr. Kadeti). Atributi *StarostPocetak* i *StarostKraj* predstavljaju granice starosti veslača koji se natječu u određenoj kategoriji. Starosti veslača koji se nalaze u starosnoj kategoriji uključuju vrijednosti ova dva atributa.

Relacija *Kategorija* predstavlja sve raspisane kategorije na određenoj regati koja je određena atributom *IdRegata*. Atribut *IdRegata* je umjetni primarni ključ relacije. Ova relacija povezuje tip čamca i starosnu kategoriju u jedinstvenu kategoriju raspisanu na regati, ovo povezivanje ostvaraju se preko stranih ključeva na atributima *IdCamac* i *IdStarosnaKategorija*. Svaka kategorija pripada točno jednoj starosnoj kategoriji i vesla se u točno jednom čamcu. Sve veze među relacijama u modelu baze podataka i njihove brojnosti prikazuje Slika 5. Atribut *Kratica* predstavlja jedinstvenu kraticu kategorije na regati i najčešće je jednak spojenim kraticama iz relacija *Camac* i *StarosnaKategorija*. Atribut *BrojKategorije* predstavlja jedinstveni broj kategorije na regati. Ovaj broj određuje organizator natjecanja. Svaka kategorija na pojedinoj mora imati jedinstven broj kategorije i zato se nad parom *IdRegata* i *BrojKategorije* nalazi ograničenje jedinstvenosti.



Slika Veze među relacijama u bazi podataka

Relacija *Posada* predstavlja zapis o pojedinoj posadi koja je prijavljena za natjecanje u određenoj kategoriji. Sastoji se od atributa: *IdPosada, IdKategorija* i *Kratica*. *IdPosada* je umjetni primarni ključ. Preko atributa *IdKategorija* relacija se povezuje s relacijom *Kategorija* kako bi se znalo kojoj kategoriji posada pripada. Svaka posada može biti prijavljena samo u jednu kategoriju. U slučaju da je isti skup veslača prijavljen u drugu kategoriju, ti se veslači smatraju drugom posadom. Atribut *Kratica* predstavlja jedinstvenu kraticu posade unutar kategorije u kojoj se posada natječe. U ovoj relaciji se ne nalazi atribut koji bi je povezivao s klubom kojem posada pripada zbog postojanja kombiniranih posada. To jest jedna posada može pripadati više klubova, a ova se informacija može dobiti povezivanjem relacija *Posada* i *Veslac*.

*PosadaVeslac* je vezna relacija između relacija *Posada* i *Veslac*. S obzirom da više veslača može veslati u jednoj posadi i jedan veslač može veslati u više posada između ove dvije relacije postoji m:n veza. U slučaju postojanja m:n veze stvara se nova relacija. U tu relaciju su kao strani ključevi uključeni primarni ključevi dviju relacija koje se povezuju. Par tih stranih ključeva predstavlja primarni ključ nove relacije. Također svi atributi koji nisu vezani uz relacije koje se povezuju se dodaju u veznu relaciju(2). U slučaju ove relacije strani ključevi koji su dio primarnog ključa su atributi *IdPosada* i *IdVeslac*. Dodatni atribut koji se nalazi u ovoj relaciji je atribut *MjestoUCamcu* koji predstavlja mjesto sjedenja pojedinog veslača u čamcu brojeći od krme čamca. Ovaj atribut zajedno s atributom *IdPosada* predstavlja prirodni ključ jer u jednoj posadi na 1 mjestu može biti samo 1 veslač.

Relacija *Klub* opisuje pojedini veslački klub za koji se prikupljaju podaci. Sastoji se od sljedećih atributa: *IdKlub, Ime* i *Kratica*. *IdKlub* je cjelobrojni tip podatka i predstavlja umjetni primarni ključ. Atribut *Ime* je neograničeni niz znakova i predstavlja ime pojedinog kluba (npr. Mladost). Atribut *Kratica* je niz znakova maksimalne duljine 10 znakova i predstavlja kraticu pod kojom određeni klub nastupa (npr. MLA).

Relacija *KontrolnaTocka* opisuje kontrolnu točku na kojoj se odvija mjerenje vremena utrke veslača. Sastoji se samo od primarnog ključa naziva *IdKontrolnaTocka* i atributa *Udaljenost* koji predstavlja udaljenost mjerne stanice od početka veslačke staze.

Relacija *Lokacija* opisuje lokaciju na kojoj se održava veslačka regata. Sastoji se od umjetnog primarnog ključa *IdLokacija*, neograničenog niza znakova naziva imena *Naziv* koji predstavlja ime lokacije i 2 realna broja s pomičnom točkom (engl. float) koji predstavljaju zemljopisnu širinu i visinu (*GeografskaSirina* i *GeografskaVisina*). Ovaj podataka nije spremljen kao tip geografije u bazu podataka jer korišteni alat za vizualizaciju ne prepoznaje format spremljen unutar baze podataka.

Relacija *Masa* sastoji se od četiri atributa: *IdMasa, Masa, VrijemeMjerenje* i *IdVeslac*. *IdMasa* je umjetni primarni ključ. Masa je decimalni tip podatka koji ima preciznost od 5 znamenki od kojih je maksimalni broj decimalnih znamenki 2. Ovaj atribut nije spremljen u obliku broja s pomičnom točkom jer bitna njegova preciznost. Atribut masa predstavlja masu veslača izraženu u kilogramima zaokruženu na 2 decimalna mjesta. Atribut *VrijemeMjerenje* predstavlja datum i vrijeme kada je izvršeno mjerenje mase. Ovaj atribut ne predstavlja samo datum jer se na pojedinim natjecanjima masa veslača može mjeriti više od jednom dnevno. Atribut *IdVeslac* predstavlja strani ključ prema tablici Veslač. Svaki zapis o mjerenju mase pripada točno jednom veslaču. Par atributa *IdVeslac* i *VrijemeMjerenje* predstavlja prirodni ključ u ovoj tablici te ja na ovaj par postavljenog ograničenje jedinstvenosti. Ovaj je par atributa prirodni ključ jer nije moguće istovremeno mjeriti masu jednog veslača na 2 uređaja.

Relacija *Visina* ima slične atribute kao i relacija *Masa* te svi istoimeni atributi iz ove dvije relacije služe istoj svrsi. Posebni atribut *Visina* je decimalni tip preciznosti 5 znamenki sa 2 rezervirana decimalna mjesta. Ovaj atribut predstavlja visinu veslača izraženu u centimetrima.

Relacija *RangUtrke* sastoji se od umjetnog primarnog ključa *IdRangUtrke*, niza znakova *Kratica* i niza znakova *Naziv*. *Kratica* predstavlja kraticu ranga određene utrke (npr. FA), *Naziv* predstavlja puni naziv ranga utrke (npr. Finale A).

Relacija *Regata* sastoji se od sljedećih atributa: *IdRegata, Ime, DatumPocetak, DatumKraj* i *IdLokacija*. *IdRegata* je umjetni primarni ključ. *Ime* je neograničeni niz znakova koji predstavlja puni naziv regate (npr. 1. Regata veslačkog kupa Miljenka Finderlea u Zagrebu). *DatumPocetak* i *DatumKraj* su podaci tipa datum (engl. date) i predstavljaju datum kada regata počinje to jest završava. Atribut *IdLoakcija* je strani ključ na tablicu *Lokacija* i predstavlja mjesto održavanja veslačke regate. Svaka se veslačka regata održava na točno jednoj lokaciji.

Relacija *Utrka* sadrži zapise o utrkama koje se održavaju u sklopu pojedine kategorije na regati. Sadrži atribute: *IdUtrka, IdKategorija, IdRangUtrke, RedniBroj* i *StartnoVrijeme*. Atribut *IdUtrka* je umjetni primarni ključ. Atributi *IdKategorija* i *IdRangUtrke* povezuju ovu relaciju s relacijama *Kategorija* i *RangUtrke*. Svaka utrka pripada točno jednoj kategoriji i nije moguće miješanje čamaca i starosnih kategorija. Za pojedine starosne kategorije mogu veslat mlađi veslači, ali se oni za potrebe utrke smatraju kao da pripadaju u starosnu kategoriju utrke. Također svaka utrka pripada samo jednom rangu utrke, nije moguće da je utrka istovremeno kvalifikacijska i finalna utrka. Atribut *RedniBroj* predstavlja redni broj utrke na regati. Relacija se na relaciju *Regata* može povezati preko relacije *Kategorija* i zato ne sadrži posebni atribut koji je direktno povezuje s relacijom *Regata*. Atribut *StartnoVrijeme* predstavlja planirano startno vrijeme utrke. Najčešće atribut *RedniBroj* prati poredak atributa *StartnoVrijeme*, ali postoje iznimke i zbog toga se atribut *RedniBroj* ne može izračunati iz atributa *StartnoVrijeme*.

Relacija *ProlaznoVrijeme* povezuje relacije *KontrolnaTocka* i *Rezultat*. Ova relacija predstavlja zapis o vremenu koje je određena posada izveslala do određene kontrolne točke. Na primjer predstavlja izveslano vrijeme određene posade na kontrolnoj točki na udaljenosti 1000 metara od početka staze. Izveslano vrijeme se s posadom povezuje preko relacije *Rezultat*. Atribut *Vrijeme* je vremenski tip podatka koji predstavlja izveslano vrijeme u satima, minuta i sekundama do točnosti od 3 decimale u sekundama. To jest u atribut vrijeme uključuju se i desetinke, stotinke i tisućinke sekunde. Par atributa *IdKontrolnaTocka i IdRezultat* su jedinstveni jer ne smije postojati više vremena na pojedinoj kontrolnoj točki za pojedinu posadu.

Relacija *Rezultat* povezuje posade u određenoj utrci s vremenima iz relacije *ProlaznoVrijeme*. Ova relacija u sebi sadrži strane ključeve *IdUtrka* i *IdPosada* kako bi povezala posadu s utrkom. Primarni ključ relacije je umjetni ključ *IdRezultat* koji se u relaciji *ProlaznoVrijeme* koristi za povezivanje na ovu relaciju. Relacija sadrži i atribut *Staza* koji označava broj staze u kojoj je posada veslala za vrijeme utrke.

Relacija *PripadnostKlubu* sadržava zapise o tome kada je određeni veslač bio registriran u određenom klubu. Ove se relacija uz pomoć stranih ključeva *IdVeslac* i *IdKlub* povezuje s relacijama *Veslac* i *Klub*. Vrlo bitan aspekt ove relacije je praćenje pripadnosti veslača klubovima u vremenu. Potrebno je moći odgovoriti na pitanje kada je veslač bio registriran u određenom klubu uz ograničenje da veslač ne može biti registriran u više od jednog kluba istovremeno. S obzirom na to da je potrebno moći rekonstruirati potpunu povijest pripadnosti veslača klubovima za ovu relaciju koristi se koncept potpunog vremenskog označavanja (engl. fully temporalising)(3). S obzirom na nepostojanje tipa koji bi predstavljao interval trajanja u korištenoj bazi podataka u ovoj relaciji se za predstavljanje intervala koriste 2 atributa. Atribut *DatumPocetak* predstavlja datum kada je veslač bio registriran za određeni klub. Atribut *DatumKraj* predstavlja datum kada je veslač prestao biti član određenog kluba. U slučaju da je vrijednost atributa *DatumKraja* jednaka NULL znači da je veslač trenutno registriran u pripadnom klubu. Za oba atributa vrijedi pravilo da su granice intervala uključene u sam interval. Zbog ograničenja domene na ova dva atributa postoji ograničenje da je atribut *DatumKraj* uvijek veće vrijednosti od atributa *DatumPočetak*. S obzirom na to da u korištenoj bazi podataka ne postoji tip indeksa koji bi osigurao ograničenje da veslač ne može biti registriran u više od jednog kluba istovremeno, ograničenje je potrebno provesti uz pomoć okidača (engl. triggers).

Relacija *TipTreninga* sastoji se od atributa: *IdTipTreninga* i *NazivTreninga. IdTipTreninga* je umjetni ključ ove relacije. Atribut *NazivTreninga* je znakovni niz koji predstavlja ime tipa treninga za kojeg se spremaju podaci. Ova relacija je potrebna kako bi se dalo značenje podacima koji se spremaju u relacije koje su vezane za relaciju *Trening*. Tipovi treninga koji se upisuju u bazu podataka moraju biti bazirani na mjerenju vremena jedne ili više dionica koje je veslač izveslao na treningu.

U relaciji *Trening* nalaze se osnovni podaci o treninzima koje su veslači izvršili. Atribut *IdTrening* umjetni je prirodni ključ na ovoj relaciji. Atribut *IdVeslac* povezuje ovu relaciju s relacijom *Veslac*. Svaki veslač može odveslati više treninga, ali rezultati svakog treninga pripadaju jednom veslaču. *VrijemeTreninga* je atribut koji govori kada se trening dogodio. *IdTipTreninga* povezuje ovu relaciju s relacijom *TipTreninga*. Relacija trening nije zamišljena da se u njoj spremaju podaci o svim treninzima nego samo podaci o posebnim vrstama treninga koji su popisani u relaciji *TipTreninga*.

Relacija *DionicaTreninga* sadrži podatke o pojedinim izveslanim vremenima koja su dio jednog treninga. Duljina izveslane dionice se ne pohranjuje jer je implicitno zadana tipom treninga. Relacija sadrži umjetni ključ naziva *IdDionicaTreninga*. Uz pomoć atributa *IdTrening* relacija se povezuje s relacijom *Trening* i na taj način se grupiraju sve relacije koje su dio istog treninga. Atribut *BrojDionice* predstavlja redni broj dionice koja je izveslana u sklopu pojedinog treninga. Par atributa *IdTrening* i *BrojDionice* mora biti jedinstven jer se dionice unutar treninga ne ponavljaju. Moguće je odveslati dionicu istih karakteristika, ali se to smatra novom dionicom. Atribut *Vrijeme* pohranjuje podatak o vremenu koje je veslaču bilo potrebno da savlada određenu dionicu.

## Pogledi

Pogled je pretraživi objekt u bazi podataka koji je definiran SQL upitom.(izvor view). Poglede se koriste u da bi se osigurao integritet poslovnih pravila, konzistentnost, sigurnost i jednostavnost. Pogledi si koriste kada je potrebno osigurati integritet poslovnih pravila u slučajevima kada je na podatke potrebno primijeniti poslovnu logiku koja nije u potpunosti razumljiva korisnicima. Koristeći poglede moguće je osigurati da svi korisnici kao rezultat upita dobiju istu interpretaciju poslovne logike. Ovaj primjer također je i primjer osiguranja konzistentnosti. Pogledi se mogu koristiti kao sigurnosni mehanizmi u slučaju da određeni korisnik u posebnoj situaciji treba veća prava nego što ih ima. Na primjer korisnik želi znati imena veslača, moguće mu je osigurati čitanje tablice u kojoj se nalaze imena i prezimena veslača. Međutim u istoj tablici mogu se nalaziti osjetljivi podaci kojima korisnik ne smije pristupiti. Uz pomoć pogleda moguće je dohvatiti samo podatke kojima korisnik smije imati pristup i onemogućiti mu direktno čitanje iz tablice s podacima. Pogledi se koriste kada se želi osigurati jednostavnost pristupa podacima. U slučaju da je za određeni upit potrebno napraviti veći broj povezivanja tablica, a taj se upit često koristi, lakše je napraviti pogled koji će izvršavati složeniji upit uvijek na isti način. Pogledi se u slučaju vizualizacije i u ovome radu koriste kako bi se osigurala jednostavnost i konzistentnost pristupa podacima. S obzirom na činjenicu da je podatke potrebno prikazati proizvoljnim alatom za vizualizaciju uz pomoć pogleda je potrebno osigurati da se povezivanje podataka događa unutar baze podataka. U sustavu za praćenje i analizu performansi veslača pogledi se koriste za olakšanu vizualizaciju podataka.

Pogled *BrojVeslacaPoKlubovimaPoGodinama* prikazuje povijesne podatke za broj veslača koji su bili registrirani u klubovima 1. siječnja u zadnjih pet godina. Atribut *pocetakSezone* označava godinu za koju pogled prikazuje podatke. Atribut *idKlub* služi kao veza prema relaciji *Klub* i označava klub za koji se prikazuje broj veslača. Atribut *brojLjudi*  predstavlja broj ljudi koji su bili registrirani za pojedini klub 1. siječnja traženih godina. Pogled prikazuje podatke za sve klubove i sve godine čak i ako je broj prijavljenih veslača 0. Pogled prikazuje podatke za sve klubove i sve godine čak i ako je broj prijavljenih veslača 0.

Pogled *MedaljePoGodinama* prikazuje broj medalja koje je pojedini klub osvojio u zadnjih 5 godina. Atribut *Mjesto* je brojčana prezentacija osvojenog mjesta i poprima vrijednosti 1, 2 ili 3. Na temelju ove vrijednosti moguće je odrediti osvojenu medalju. Atribut *Godina* označava godinu za koju se dohvaća broj osvojenih medalja. Atribut *idKlub* služi kao veza prema relaciji *Klub* i označava klub za koji se prikazuje broj medalja. Atribut *BrojMedalja* predstavlja broj osvojenih medalja u određenoj godini od strane određenog kluba. Pogled prikazuje podatke za sve klubove i sve tipove medalja čak i ako je broj osvojih medalja 0.

Pogledi *PredikcijskiTrening2000* i *PredikcijskiTrening500* su srodni pogledi. Ovi pogledi prikazuju podatke o tipu treninga koji se nalazi u njihovim imenima. Atribut *IdVeslac* predstavlja vezu prema relaciji *Veslač* te predstavlja veslača koji je odveslao trening. Atribut *VrijemeTreninga* predstavlja vrijeme i datum kada je započeo traženi trening. Atribut *BrojDionice* predstavlja redni broj odveslane dionice u određenom treningu. Duljina dionice implicitno je određena tipom treninga, a to je 2000 metara za *PredikcijskiTrening2000* i 500 metara za *PredikcijskiTrening500*. Atribut *Vrijeme* predstavlja vrijeme koje je veslaču bilo potrebno da odvesla određenu dionicu.

Pogledi *PredikcijskiTrening2000Predikcija* i *PredikcijskiTrening500Predikcija* su srodni pogledi i prikazuju predviđeno vrijeme koje veslač može odveslati na vremenski bliskoj utrci. Ovi pogledi predviđanja temelje na podacima koje prikazuju pogledi *PredikcijskiTrening2000* i *PredikcijskiTrening50.* Atribut *IdVeslac* predstavlja vezu prema relaciji *Veslač* te predstavlja veslača za kojeg se predviđa vrijeme veslanja regate. Atribut *VrijemeTreninga* predstavlja vrijeme i datum kada je započeo traženi trening na temelju kojeg se radi predviđanje vremena veslanja regate. Atribut *PredvidenoVrijeme* predstavlja predviđeno vrijeme koje bi veslač mogao odveslati na vremenski bliskoj regati. Ovaj atribut predstavlja glavnu razliku između promatranih pogleda.

U slučaju pogleda *PredikcijskiTrening2000Predikcija* računa se kao:

U slučaju pogleda *PredikcijskiTrening500Predikcija*  atribut *PredvidenoVrijeme*  računa se kao:

Sva prikazana vremena moraju biti prikazana u sekundama. Oznaka T je predviđeno vrijeme veslanja. Oznake t1 i t2 predstavljaju vremena veslanja pojedinih dionica.

Pogled *RankPoPosadamaPoGodinama* predstavlja sličan pogled kao i *MedaljePoGodinama.* Ovaj pogled prikazuje broj osvojenih mjesta u finalnim utrkama na svim regatama grupiranim prema posadama. Atribut *IdPosada* povezuje ovaj pogled s relacijom *Posada* kako bi se znalo koje je mjesto pojedina posada osvojila. Atribut *DatumRezultata* predstavlja datum kada je osvojeno određeno mjesto u utrci. Atribut *Rank* predstavlja mjesto koje je određena posada osvojila na određenoj regati.

Pogled *RegataBrojPosadaPoKategorijama* prikazuje broj posada koje su veslale na određenoj regati grupirane prema kategorijama. Atributi *IdRegata* i *IdKategorija* povezuju pogled s relacijama *Regata* i *Kategorija.* Prema ovim atributima obavlja se grupiranje rezultata pogleda. Atribut *Broj posada*  predstavlja broj posada koje su veslale u određenoj kategoriji na pojedinoj regati.

Pogled *RegataBrojVeslacaPoKlubovima*  srodan je prethodnom pogledu i predstavlja broj veslača koji su na određenoj regati veslali za pojedini klub. Na relacije *Klub* i *Regata* pogled se referencira preko atributa *IdKlub* i *IdRegata.* Atribut *Broj veslaca*  predstavlja broj veslača koji su veslali za određeni klub na pojedinoj regati.

Pogled *ZbirnaStartnaLista* sadrži zapise iz baze podataka koji povezani tvore zbrinu startnu listu. To je lista zapisa o utrkama koje se veslaju na određenoj regati sortirana prema rednom broju koji je dodijeljen svakoj pojedinoj utrci. Atributi *IdKategorija, IdUtrka* i *IdRegata* povezuju pogled s pripadajućim relacijama. Atribut *RedniBrojUtrke* predstavlja redni broj utrke sklopu jedne regate, atribut *Kategorija* predstavlja kraticu kategorije kojoj određeni zapis pripada, a atribut *StartnoVrijeme* predstavlja startno vrijeme pojedine utrke. Atribut *Rang* predstavlja kraticu ranga kojem utrka pripada na primjer Finale A.

Pogled *RegataVremenaPoUtrkama*  je pogled s najviše atributa u sustavu za praćenje i analizu veslača. Ovaj pogled predstavlja izveslana vremena na regatama za sve posade u svim kategorijama. Koristeći atribute *IdRegata, IdKategorija, IdUtrka* i *IdPosada* pogled se povezuje s tablicama gdje su ovi atributi primarni ključevi. Atribut *Staza* označava stazu u kojoj je posada u određenoj utrci odveslala utrku. Atribut *Kratica* označava kraticu pod kojom je posada veslala utrku. Atribut *Vrijeme*  predstavlja vrijeme koje je određena posada izveslala u određenoj utrci. Atribut *Udaljenost*  predstavlja udaljenost kontrolne točke na kojoj se mjeri izveslano vrijeme od početka veslačke staze.

Pogled *VeslaciPoStarosnimKategorijama* prikazuje broj veslača koji se nalaze u klubovima grupirano prema starosnim kategorijama. Atribut *IdStarosnaKategorija* povezuje se na relaciju *StarosnaKategorija* i označava kojoj starosnoj kategoriji pripadaju veslači u klubu. Atribut *IdKlub* povezuje pogled na pripadajući klub. Atribut *BrojVeslaca* i predstavlja broj veslača u određenom klubu grupiranih prema starosnoj kategoriji.

Pogled *VeslaciUParu* prikazuje parove veslača koji su veslali skupa u posadi. Za svakog veslača u posadi u ovom pogledu postoje posebni zapisi koji ga povezuju s ostalim veslačima iz posade. Pogled također prikazuje koliko su puta veslači veslali skupa. Veslači su prikazani atributima *IdVeslac, VeslaoSaId* i *VeslaoSa*. Prva dva atributa povezuju pogled s relacijom *Veslač.* Atribut *VeslaoSa* predstavlja ime i prezime drugog veslača u paru. Ovaj atribut nalazi se u pogledu zbog ograničenja određenih sustava za vizualizaciju podataka. Neki sustav ne mogu imati entitete koji se više puta povezuju na isti atribut drugog entiteta. U ovom specifičnom slučaju nije moguće povezati entitet *VeslaciUParu* na atribut *IdVeslac* entiteta *Veslac*. Atribut *VeslaliPuta* predstavlja broj puta koji je određeni par veslača veslao skupa u posadi.

Pogled *VeslacPoKategoriji* predstavlja zapise o broju puta koji je određeni veslač veslao u određenoj kategoriji. Uz pomoć atributa *IdVeslac* rezultati se grupiraju prema veslaču, a preko atributa *Kratica* grupiraju se prema kategoriji. Atribut *PutaVeslao* predstavlja broj puta koji je određeni veslač veslao u određenoj kategoriji.

Pogled *VeslanjeNaLokaciji* predstavlja zapise o lokacijama na kojima je veslao određeni veslač. Preko atributa *IdVeslac* i *IdLokacija* pogled se povezuje s pripadajućim relacijama. Atribut *PutaNaLokaciji* predstavlja broj puta koji je pojedini veslač veslao na određenoj lokaciji.

Pogled *VremenaPoVeslacu* sadrži zapise o svim vremenima koje je određeni veslač veslao na regata. Pogled uključuje vremena koja je veslač izveslao sam i kao dio posade. Atributi *Ime, Prezime* i *IdVeslac* predstavljaju podatke o veslaču. Atributi *ImeRegate* i *DatumPocetak* predstavljaju podatke o regati na kojoj je vrijeme izveslano. Atribut *DatumPocetak* se nalazi u ovom pogledu kako bi se rezultati mogli sortirati vremenski po datumu veslanja. Atribut *Vrijeme*  predstavlja izveslano vrijeme u određenoj utrci. Atribut *Kratica* označava u kojoj je kategoriji postignut rezultat, a atribut *Udaljenost* predstavlja kontrolnu točku na kojoj je izmjereno izveslano vrijeme.

## Imenovanje objekata

Imenovanje objekata u bazi podataka moguće je napraviti na više različitih načina. Imenovanje je bitno kako bi se jednostavnije povezali ili pronašli entiteti u bazi podataka. Konzistentnost u imenovanju objekata je bitna radi preglednosti sustava i moguće buduće nadogradnje. Sustav za praćenje i analizu performansi veslača ima definirana pravila prema kojima se imenuju objekti unutar baze podataka.

Relacije se imenuju imenima koja označavaju tip podataka koji se u njima nalaze. Imena relacija moraju biti u prvom licu jednine u nominativu. U slučaju da se ime relacije sastoji od dvije riječi, barem jedna riječ mora biti u nominativu. Svaka riječ u imenu relacije mora započeti velikim slovom (*engl. upper camel case*). Vezne relacije imenuju se tako da se spoje imena relacija na koje se odnose.

Pogledi se imenuju prema tipu podataka koji se u njima nalazi. Pogledi slijede ista pravila kao i relacije. Pogledi u svojim imenima mogu sadržavati priloge i prijedloge.

Svi primarni ključevi u bazi podataka moraju biti imenovani. Imena primarnih ključeva sadrže prefiks *„PK\_“*  koji slijedi ime tablice u kojoj se primarni ključ nalazi. Primjer imenovanog primarnog ključa je „*PK\_PosadaVeslac*“.

Svi strani ključevi u bazi podataka moraju biti imenovani. Imena stranih ključeva sadrže prefiks *„FK\_“* koji slijedi ime tablice u kojoj se nalazi ključ iza kojeg se nalazi ime referencirane tablice odvojene donjom crtom. Primjer imenovanog stranog ključa je „*FK\_Posada\_Kategorija*“.

Ograničenja jedinstvenosti moraju biti imenovana. Imena ograničenja jedinstvenosti sadrže prefiks *„UQ\_“* iza kojeg slijede imena svih atributa koji su uključeni u ograničenje razdvojenih donjom crtom. Primjer imenovanog ograničenja jedinstvenosti je „*UQ\_IdPosada\_MjestoUCamcu*“

Ograničenja s provjerom (*engl. check constraints*) sadrže prefiks *„CH\_“* koji slijede imena atributa uključenih u provjeru odvojenih donjom crtom. Primjer imenovanog ograničenja s provjerom je „*CH\_DatumPocetak\_DatumKraj*“

## Sustav za upravljanje bazom podataka

Sustav za analizu i praćenje performansi veslača kao sustav za upravljanje bazom podataka koristi sustav „*SQL Server*“. Sustav „*SQL Server*“ proizvod je tvrtke Microsoft. Ovaj sustav za upravljanje bazom podataka postoji u nekoliko inačica. Najčešće korištene inačice ovog sustava zovu se *„SQL Server Standard“, „SQL Server Express“* i *„LocalDB“.* Ove tri inačice razlikuju se prema broju usluga koje pružaju, kompliciranosti instalacije i resursima koje troše.

Sustav *„SQL Server Standard“* je najnapredniji od tri promatrana sustava. Ovo je sustav koji se najčešće koristi za produkcijske baze podataka. „SQL Server“ nudi najveći spektar mogućnosti, ali je zbog toga i najteži za instaliranje i postavljanje. Najbitnije mogućnosti ovog sustava su: mogućnost udaljenog pristupa bazi podataka, mogućnost pokretanja sustava za bazu podataka kao servis i mogućnost stvaranja šifriranih sigurnosnih kopija. Pokretanje sustava kao servis omogućuje da se sustav za upravljanje bazom podataka pokreće kod paljenja računala i da za njegovo pokretanje nije potrebna korisnička interakcija. Mogućnost stvaranja šifriranih sigurnosnih kopija omogućuje da su svi prikupljeni osjetljivi podaci zaštićeni čak i u obliku sigurnosnih kopija. Ova činjenica je bitna zbog zakona Europskog zakona o zaštiti osobnih podataka.

Sustav *„SQL Express“* omogućuje slabije performanse od sustava *„SQL Server Standard“.* Uz slabije performanse ovaj sustav ne omogućava stvaranje šifriranih sigurnosnih kopija. Ovaj je sustav omogućava udaljeni pristup bazama podataka i omogućuje pokretanje sustava za upravljanje bazom podataka kao servis. Ova inačica sustava je besplatna.

Sustav *„LocalDB“* je sustav namijenjen za korištenje pri razvoju programskih rješenja. Ovaj sustav ne omogućava udaljeni pristup, ne omogućava šifrirane sigurnosne kopije i ne omogućava pokretanje sustava kao servis. Iako ne posjeduje nijednu od promatranih naprednih mogućnosti ovaj je sustav najlakši za instalaciju i postavljanje. Svi koncepti i sve operacije koje je moguće napraviti na sustavu *„LocalDB“* moguće je napraviti i na drugim inačicama sustava „*SQL Server*“.

Prilikom razvoja sustava za analizu i praćenje performansi veslača korištena je inačica sustava *„LocalDB“*. Ova je inačica korištena zato što se baza podataka, sustav za vizualizaciju i sustav za unos podataka nalaze na istom računalu. Također baza podataka mora biti dostupna samo za vrijeme korištenja aplikacije za unos i za vrijeme korištenja alata za vizualizaciju i iz tog razloga nije potrebno imati mogućnost pokretanja sustava kao servisa. Prilikom puštanja sustava u javnost preporuča se korištenje inačice *„SQL Server Standard“.* Ova se inačica preporuča kako bi se omogućio istovremeni pristup sustavu s više udaljenih računala i kako i bi se osjetljivi podaci mogli najefikasnije zaštititi. Migracija podataka s inačice *„LocalDB“* na jednu od razvijenijih inačica je vrlo jednostavno jer sve inačice dijele isti upitni jezik i međusobno su kompatibilne za prijenos podataka.

# Vizualizacija podataka

Računalni sustav za vizualizaciju podataka pružaju vizualnu reprezentaciju skupova podataka koji su dizajnirani kako bi ljudi efikasnije izvršavali dane zadatke.(Peters str1). Vizualizaciju je prikladno koristiti kada je potrebno promijeniti ljudsko viđenje podataka. Vizualizacija se ne koristi kako bi računala donosila odluke. Dizajn vizualizacija je veliko područje koje uključuje oblikovanje i interakciju s vizualiziranim podacima. Dizajniranje vizualnih reprezentacija podataka težak je posao jer većina mogućih prikaza podataka neefektivna u prenošenju potrebnih informacija. Zbog toga je vrlo važno dobro dizajnirati i provjeriti efektivnost vizualiziranih podataka. Osoba koja izrađuje vizualnu reprezentaciju podataka mora uzeti u obzir ograničenja koja posjeduju ljudi, računala i ekrani.

Današnji računalni sustavi imaju sve jaču računalnu moć. S ubrzavanjem računalnih sustava i razvojem novih algoritama za računanje moguće je pretpostaviti da se većina odluka može prepustiti računalima. Računala su adekvatna kada je potrebno odgovoriti na unaprijed poznata pitanja. Matematika i računala dovoljno su napredni da bi odgovorili na najčešća poznata pitanja. S druge strane u mnogim područjima postoji potreba za stvaranjem korisnog znanja bez poznavanja točnog pitanja na koje je potrebno odgovoriti. Vizualizacija podataka može ponuditi rješenje na ovaj problem. S obzirom na to da vizualizacija prikazuje podatke uz pomoć slika i grupira ih, ljudima je lakše uočiti pravila u podacima nego kad se oni nalaze u drugim oblicima. Ljudski faktor je ovdje presudan jer čovjek donosi zaključke, a ne računalo.

## Semantika i tipovi podataka

Veliki broj aspekata vizualizacije proizlazi iz tipova podataka koji su dostupni i koje je potrebno vizualizirati. Međutim sami podaci ne daju dovoljno informacija za njihovo razumijevanje. Kako bi se iz podataka moglo dobiti korisno znanje potrebno je poznavati njihova dva aspekta: semantiku podataka i tip podataka.

Semantika se definira kao značenje i korištenje podataka. U informacijskim sustavima semantika se može definirati kao mapiranje objekta koji se nalazi u informacijskom sustavu i stvarnog objekta kojeg predstavlja. (izvor semantika) Iako su ove definicije općenite obje govore da je semantika postavljanje podatka u određeni kontekst. Tek kada je poznata semantika podataka iz podataka se mogu početi stvarati novi podaci i korisno znanje.

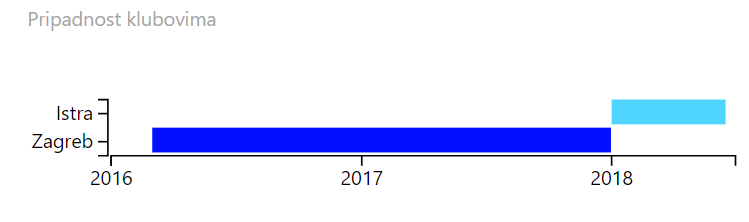
Tip podatka predstavlja njegov oblik ili strukturu. On opisuje podatak na najnižoj razini. Tip podataka odgovara na pitanje kako podatak izgleda i koja su mu prirodna ograničenja. Na primjer tip podatka govori o broju znamenki koje se koriste za prikaz realnog broja ili pak o broju slova koji se koristi za opis određenog objekta. Tip podatka ne mora biti jednostavan, složeni tipovi podataka mogu se sastojati od više jednostavnih tipova ili pak od cijelih hijerarhija tipova podataka.

Potrebno je poznavati i tip i semantiku podataka kako bi se iz podataka dobilo korisno znanje. I tip podatka i semantika nose određeno znanje i ograničenja. Na primjer ako postoji podatak cjelobrojnog tipa iz matematike je poznato da se nad takvim tipovima može izvršiti operacija zbrajanja. Međutim semantika podatka kaže da taj cjelobrojni tip označava poštanski broj i s tom informacijom može se zaključiti da zbrajane takvih cjelobrojnih tipova ne može stvoriti nikakvo korisno znanje.

## Vizualizacija vremenski zavisnih podataka

Vremenski atribut je bilo koja vrsta informacije koja se odnosi na vrijeme. Podaci koji se odnose na vrijeme vrlo su komplicirani za obradu jer je vrijeme složena hijerarhijska struktura. Vrlo je važna semantika vremenskih podataka kako bi se znalo na kojoj se vremenskoj skali radi s podacima. Podaci koji sadrže informaciju o nano sekundama najčešće nisu korisni ako se vrijeme računa na tisućljetnoj skali.

Drugi problem kod vizualizacije podataka je što vremenski podaci mogu predstavljati i ključ i vrijednost ovisno o kontekstu u kojem se nalaze. S obzirom na to je li vremenski podatak ključ ili vrijednost s njim se treba postupati na drugačiji način. U slučaju da je vremenski podatak ključ moguće su 2 mogućnosti: ključ i vrijednost su diskretne vrijednosti te ključ i vrijednost su nediskretne vrijednosti te nose kontinuiranu informaciju. Za drugi slučaj vizualizacije vremenskih podataka koriste se vremenske crte (engl. timeline) koje prikazuju diskretne vrijednosti na kontinuiranoj vremenskoj skali (Slika 6).



Slika Pripadnost veslača veslačkim klubovima po godinama

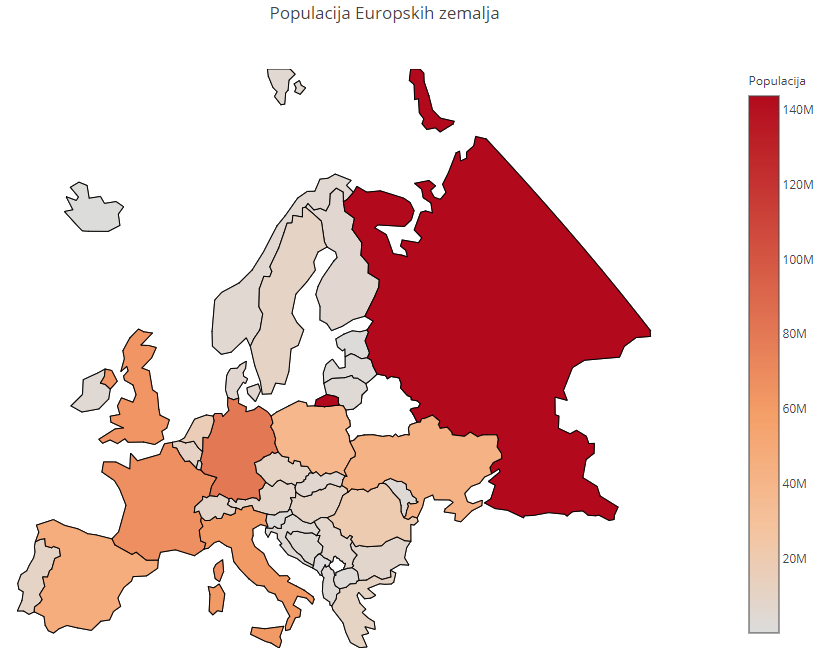
Slika 6 prikazuje vremensku pripadnost veslača određenom klubu. Vrijeme u ovom slučaju kontinuirani period, a vrijednosti su diskretni nizovi slova koji predstavljaju nazive klubova.

## Vizualizacija geoprostornih podataka

Geoprostorni podaci su podvrsta prostornih podataka. Prostorni podaci su podaci koji opisuju dani dvodimenzionalni ili trodimenzionalni prostor. Geoprostorni podaci se odnose na podatke koji opisuju prostor planete zemlje. Geoprostorni podaci mogu se prikazati na razne načine, a odabir prikaza ovisi o samim podacima i znanju koje se želi sintetizirati iz podataka. Na primjer ovisno o potrebama grad Zagreb se na karti može prikazati kao točka ili pak kao geometrijski lik koji odgovara njegovim granicama. Uz sami odabir prikaza geometrijskih podataka vrlo je važno na prikladan način odabrati kako će na grafu prikazivati podaci koji su ovisni o samim geoprostornim podacima. Ova dva odabira objedinjava kartografska generalizacija. Kartografska generalizacija označava skup akcija kojima se iz početnih neobrađenih podataka dolazi do skupa geoprostornih podataka koji je prikladan za prikazivanje na karti. Dobiveni skup također mora zadovoljavati potrebe korisnika karte.

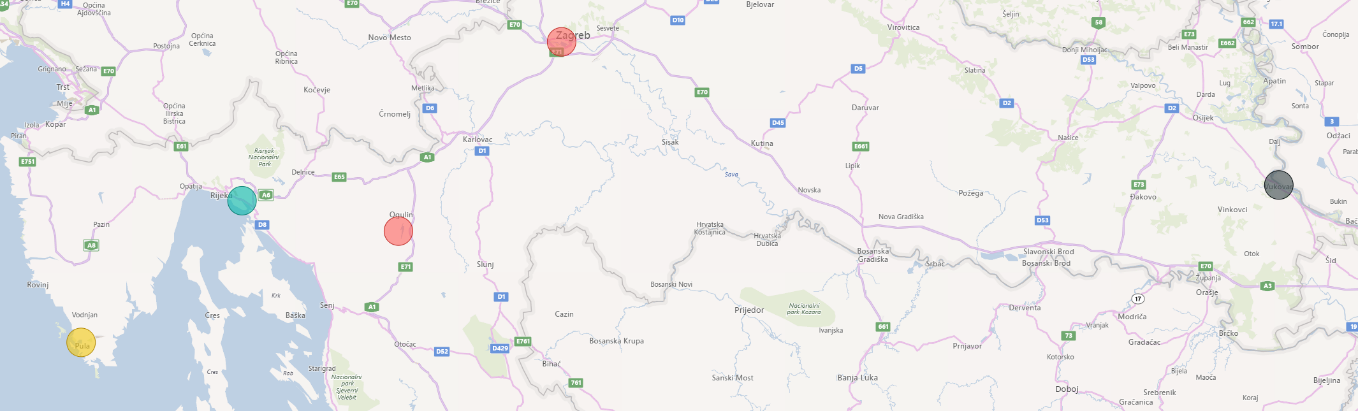
Jedan od vrlo čestih načina prikazivanja geoprostornih podataka je mapom boja (engl. choropleth map).Ovaj način prikaza za geoprostorne podatke koristi granice geometrije koja se prikazuje, na primjer granice grada Zagreba. Kako bi se prikazali podaci vezani uz određenu geometriju, prikazane geometrije bojaju se u različitim bojama ili istim bojama različitog intenziteta (Slika 7).

Ovakav način prikaza idealan je kod skalarnih podataka koji su vezani uz određeno geografsko područje. Također ovaj je prikaz prikladan kada prikazane geometrije dijele granice i kada je potrebno uspoređivati podatke prostorno bliskih geometrija.



Slika Populacija Europskih zemalja prikazana mapom boja

Prikaz geoprostornih podataka krugovima koristi se u sličnim situacijama kao i prikazivanje mapom boja. Kod prikaza krugovima kao vrijednosti vezane uz geometriju također se koriste skalarne vrijednosti. Te se vrijednosti prikazuju krugovima različitih promjera. Glavna razlika kod ova dva prikaza je ta što je prikaz krugovima bolje koristiti kada prikazane geometrije ne dijele granice ili su prostorno udaljene. Također u ovom prikazu geometrije se najčešće označavaju točkom umjesto geometrijskim likom. U tom slučaju za prikaz geometrije potrebno je poznavati geografsku dužinu i geografsku širinu. Kao i kod svih podataka geografsku visinu i širinu potrebno je transformirati u format koji je prikladan za prikaz odabranim alatom za vizualizaciju. Slika 8 prikazuje broj regata koji se održavao u pojedinom gradu. Prikaz je prikazan grafom s krugovima.



Slika Lokacije održavanja regata u Hrvatskoj

## Vizualizacija temeljena na broju ključeva

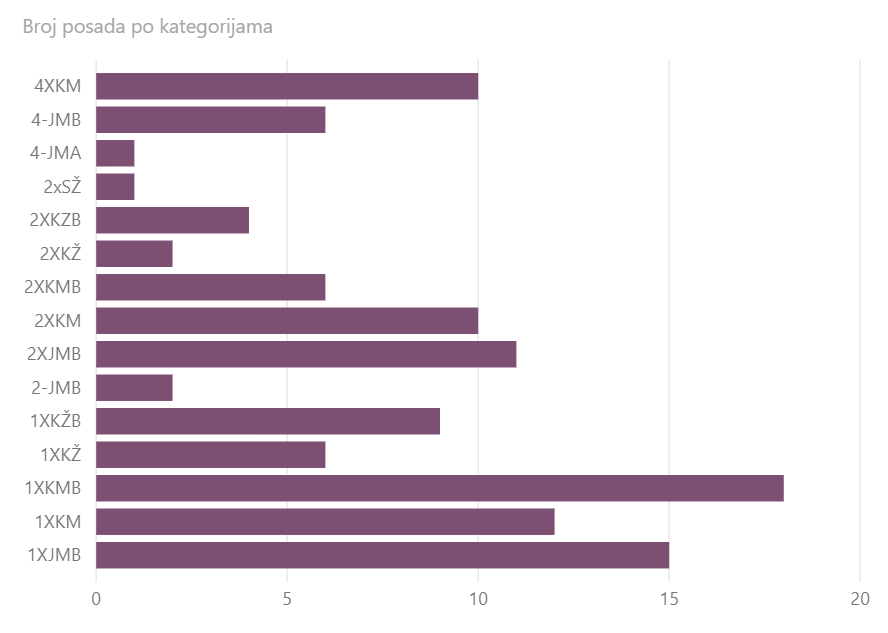
Prilikom vizualizacije tabličnih podataka grafovima potrebno je znati i razumjeti uloge atributa u tablici. Ulogu atributu daje semantika podatka i semantika same tablice. Atribut može biti ključ ili vrijednost. Ključ je nezavisni atribut koji se može koristiti kao jedinstveni indeks po kojem se mogu pretraživati vrijednosti u tablici. Vrijednost je zavisni atribut i ovisi o ključu. Ključevi mogu biti kategorijski (engl. categorical) ili redni (engl. ordinal). Kategorijski ključevi nemaju implicitni poredak i često su tekstualni. Predstavljaju atribut prema kojem su vrijednosti u tablici grupirane. Redni ključevi imaju implicitno definirani poredak i često su brojčanog tipa. Predstavljaju vrijednosti koje je logično promatrati u definiranom nizu. Vrijednosni atributi mogu biti kategorijski, redni ili količinski (engl. quantitive).

### Vizualizacija podataka s jednim ključem

Prilikom vizualizacije podataka koji imaju samo jedan ključ graf se dijeli na jednu regiju po ključu. Regija je dio grafičkog prikaza u kojem se nalaze podaci vezani za jedan ili više ključeva. Prilikom vizualizacije s jednim ključem regije se rasprostiru jednodimenzionalno u poretku liste. Poredak liste može biti horizontalni ili vertikalni.

Grafički prikaz podataka s jednim ključem je dvodimenzionalan. Ključevi su poredani na jednoj osi, a regije u kojima se prikazuju podaci raspoređene su na drugoj osi. Ovisno o tipu ključeva i vrijednosti te željenim informacijama koje se žele dobiti vizualizacija podataka s jednim ključem može se napraviti koristeći različite grofovske prikaze.

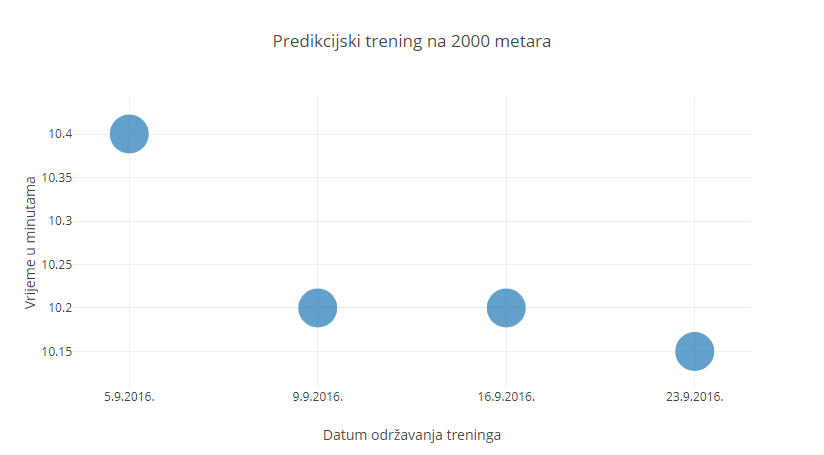
U slučaju da je potrebno vizualizirati podatke koji sadrže jedan kategorijski ključ i jednu količinsku vrijednost najbolje je koristiti graf s crtama (engl. bar chart). Podaci se na grafu s crtama kodiraju kao odvojene crte za svaku vrijednost ključa. Duljina crte ovisi o vrijednosnom atributu (Slika 9).



Slika Graf s crtama

Graf s crtama koristi se u kada je potrebno uspoređivati podatke i kada je potrebno pronaći vrijednosti u ovisnosti o ključu (engl. lookup). Graf s crtama vrlo je često korišten zbog svoje skalabilnosti. Broj različitih vrijednosti atributa koji se mogu koristiti kao ključ se kreće od nekolicine atributa do stotina atributa. Ovaj broj vrlo je ovisan o veličini samog grafa. Prilikom dizajna grafa s crtama vrlo je bitno pravilno sortirati vrijednosti ključeva. Ako je potrebno brzo pronaći vrijednost uz određeni ključ, ključevi se trebaju sortirati abecednim redom. Ovaj način sortiranja najčešće onemogućuje uočavanje trendova u podacima. U slučaju da je potrebno uočiti trendove ili sličnosti među ključevima, ključeve bi trebalo sortirati linearno prema njihovim vrijednostima. Hoće li sortiranje biti uzlazno ili silazno ovisi o domeni problema. Slika 9 prikazuje broj posada koje su veslale u određenoj kategoriji na odabranoj regati. U ovom slučaju ključ je ime kategorije, a vrijednost broj prijavljenih posada za kategoriju. Sortiranje ključa u ovom slučaju je abecedno zbog semantike ključa. Abecednim sortiranjem u ovom slučaju kategorije se sortiraju prema veličini čamca koji se vesla.

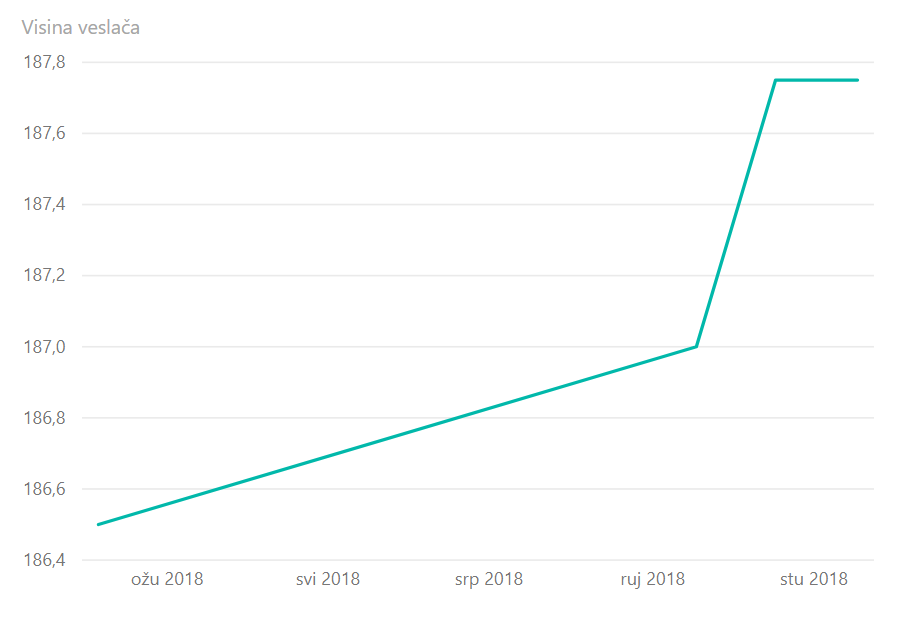
U slučaju da je potrebno vizualizirati podatke koji sadrže jedan redni ključ i jednu količinsku vrijednost preporuča se koristiti graf s točkama (engl. dot chart). Podaci se na grafu s točkama kodiraju kao točke za svaku odvojenu vrijednost ključa. Točke su u dvodimenzionalnom prostoru raspoređene u ovisnosti o svojem ključu i vrijednosti (Slika 10).



Slika Graf s točkama

Graf s točkama na jednoj osi prikazuje poredane vrijednosti iz rednog ključa a na drugoj osi vrijednosti atributa koji se promatra.

U slučaju da je na grafu potrebno uočiti trendove graf s točkama može se zamijeniti linijskim grafom. Linijski graf izgleda kao graf s točkama gdje su točke povezane linijama (Slika 11). Ovaj graf naglašava razlike između susjednih ključeva i zato se koristi kada je potrebno u podacima uočiti trendove.



Slika Linijski graf

Kada se podaci prikazuju linijskim grafom bitno dobro odrediti omjer duljina osi s atributima. Preporuča se korištenje osi jednakih duljina jer ljudi najlakše razlikuju nagibe oko 45 stupnjeva.

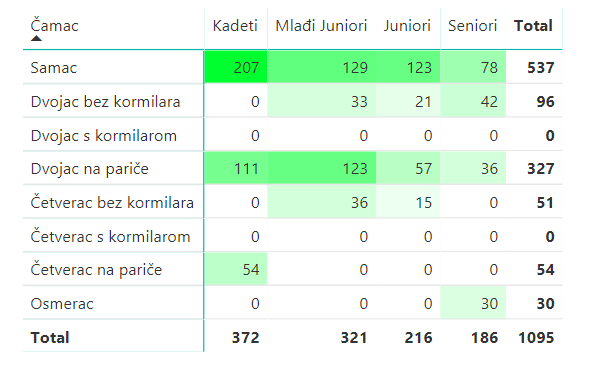
Slika 11 prikazuje ovisno visine veslača o vremenu. Ovaj način prikaza se koristi jer je vrijeme redni ključ. Za ovaj prikaz ne koristi se graf s točkama jer je važno uspoređivanje susjednih vrijednosti i uočavanje trenda promjene visine.

Linijski graf ne bi se smio koristiti kod prikaza s kategorijskim ključevima zato što ljude asocira da povezuju kategorije koje nisu nužno povezane. Graf s točkama i linijski grafu mogu imati stotine različitih vrijednosti ključeva na jednom grafičkom prikazu.

### Vizualizacija podataka s dva ključa

Vizualizacija podataka koji sadrže 2 ključa je kompleksnija od vizualizacije s 1 ključem. S obzirom na činjenicu da je na grafu potrebno prikazati dva ključa, a većina se vizualizacija prikazuje u dvodimenzionalnom prostoru, postavlja se pitanje kako je potrebno kodirati vrijednosne atribute. Prostorni kanal prikaza zauzet je prikazom ključeva i zbog toga se vrijednosti ne preporučuje prikazivati na način koji će ovisiti o prostoru. Preporučeni način prikazivanja atributa je prikaz bojama.

Najčešći prikaz podataka s dva ključa je grafom naziva žarišna mapa (engl. heatmap). Žarišna mapa je graf oblika matrice gdje se vrijednosti jednog ključa nalaze na jednoj osi, a vrijednosti drugog ključa na njoj okomitoj osi. Dobivena regija za prikaz vrijednosnog atributa je pravokutnik koji se nalazi na presjeku linija koje su određene lokacijama ključeva (Slika 12).



Slika Žarišna mapa s dodatkom brojeva

Kodiranje vrijednosti atributa radi se uz pomoć boje. Odabir načina kodiranja bojom ovisi o semantici podataka. Cilj kodiranja je lako uočavanje prostora koji su različiti od okoline. Uz boju na žarišnu mapu mogu se dodati i brojevi kako bi se lakše uočile razlike između slično obojanih područja. Dodavanje brojeva se preporuča samo ako je broj različitih regija manji od 100. Slika 12 prikazuje broj posada koje su u godinu dana veslale u Veslačkom kupu Miljenka Finderlea. Broj posada je prikazan u ovisnosti o tipu čamca i starosnoj kategoriji u kojoj je posada veslala. Na grafu je lako primijetiti da se veće vrijednosti nalaze u gornjem lijevom trokutu. Iz te činjenice može se zaključiti da se u kupu Miljenka Finderlea najviše veslaju manji čamci i da je većina veslača mlađe životne dobi.

Žarišne mape mogu prikazati veliki broj različitih vrijednosti atributa zbog činjenice da imaju dvije osi. U slučaju da svaki ključ ima 200 različitih vrijednosti na žarišnoj mapi može se prikazati 40000 različitih vrijednosti. Broj različitih vrijednosti koji se mogu prikazati limitiran je vidljivosti pojedinog pravokutnika koji prikazuje boju. Međutim s obzirom da žarišne mape prikazuju mala područja ne kontinuirane boje ljudski vidni sustav u njoj može razaznati samo 3 do 11 različitih razina vrijednosnih atributa.

## Vizualizacija podataka korištenjem tablica

Tablica je efikasan format usporedne analize objekata prema kategorijama. Najčešće se objekti koji se uspoređuju stavljaju u stupce, a kategorije u redove. (izvor tablica)Vrijednosti se zatim stavljaju na presjek retka i stupca koji se naziva čelija (Slika 13).



Slika Rezultati utrke prikazani tablicom

Prikaz tablicom prigodan je kada je potrebno usporediti vrijednosti s velikom preciznosti. Tablice često mogu zamijenit jednostavnije tipove grafova poput grafa s crtama i kružnog grafa (*engl. pie chart*). Tablice se često izbjegavaju u vizualizaciji jer ljudi često imaju dojam da se svi podaci bolje prikazuju grafovima umjesto tablicama što nije točno. U slučaju da je potrebno uočiti trendove ili grupe u podacima, tablice nisu dobar izbor jer podaci s bliskim vrijednostima nisu nužno vizualno slični ili bliski.

Slika 13 prikazuje rezultate utrke u tabličnom obliku. Koristi se tablični prikaz ovog skupa podataka zato što je atribut vrijeme potrebno razlikovati do preciznosti od milisekunde. Stupac „Kratica Posade“ u ovom slučaju predstavlja kategoriju iz zbog toga kategorija nije posebno označavana u svakom redu.

## Boja u grafovima

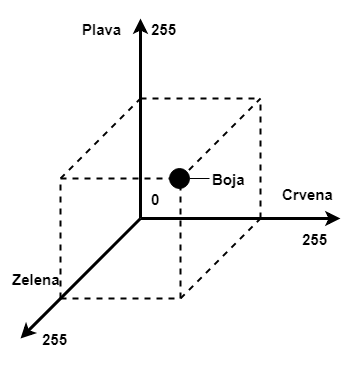
Boja je svojstvo objekta koji reflektira ili emitira svjetlost. Ljudsko oko različito doživljava različite valne duljine svjetlosti. Ta razlika u doživljajima naziva se bojom.(google-nači bolji izvor i definiciju)

Vid je osjetilo na koje se ljudi najviše oslanjaju. Iz tog razloga informacije koje čovjek najlakše percipira su informacije koje primi vidom. Boja je jedan od najvećih aspekata vida i zbog toga u vizualizaciji podataka poprima vrlo bitnu ulogu.

### Prostori boja

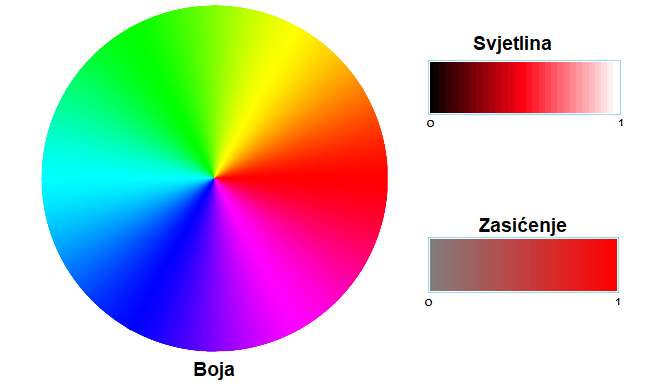
Prostor boja je skup svih boja koje određeni sustav može registrirati ili prikazati. Ljudsko oko ima 3 vrste receptora koji osjetljivi na svjetlost različitih valnih duljina(izvor ljudsko oko). I ove činjenice proizlazi zaključak da se prostor boja koje ljudsko oko doživljava najbolje može opisati kao prostor s tri različite osi. Zbog toga je većina sustava boja koji se koriste u računalnim sustavima trodimenzionalna.

U računalnoj grafici najrašireniji prostor boja je RGB prostor boja. Svaka boja prikazuje se kao triplet crvene (engl. red), zelene (engl. green) i plave (engl. blue). Iako je ovaj način prikaza boja vrlo raširen on jako je slabo usklađen s načinom na koji ljudi vide boje. Ako se svaki kanal u RGB prostoru boja prikazuje zasebno on ne nosi korisnu informaciju. RGB prostor boja često se prikazuje s tri osi na čijem se presijeku vrijednosti nalazi određena boja (Slika 14).



Slika RGB prostor boja

Prostor boja HSL intuitivniji je za korištenje kod grafičkog dizajna i sličniji je načinu na koji ljudsko oko doživljava boje. Ovaj prostor boja također se sastoji od 3 kanala. Kanali su: nijansa (engl. hue), zasićenje (engl. saturation) i svjetlina (engl. lightness). Kanal nijanse nosi informaciju koja se kolokvijalno naziva „čistom“ bojom. Ovo je boja koja nije pomiješana s nijansama crne ili bijele. Kanal zasićenja nosi informaciju koliko je bijele boje pomiješano s bojom iz kanala nijanse. Suprotno od kanala zasićenja kanal svjetline nosi informaciju o količini crne boje koja je pomiješana s „čistom“ bojom. Ovaj prostor boja često se prikazuje krugom gdje je kanal nijanse na bridovima kruga. U centru kruga nalazi se bijela boja, a pored kruga se nalaze odvojene linearne kontrole koje predstavljaju kanale zasićenja i svjetline (Slika 15).

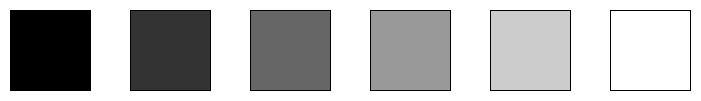


Slika HSL prosto boja prikazan krugom

### Uloga boje u vizualizaciji

Boja u vizualizaciji ima dvostruku ulogu. Ona može imati razlikovnu ili identifikacijsku ulogu. Iz ovog razloga boja na grafovima može biti izvor nejasnoće i potrebno ju je koristiti s oprezom.

Kada se boja koristi u razlikovnoj ulozi koriste se različite vrijednosti svjetline boje kako bi se razlikovali podaci. Slika 16 prikazuje sivu boju s različitim razinama svjetline.



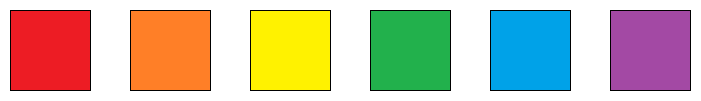
Slika Različite svjetline sive boje

Ovaj način prikaza je problematičan zbog karakteristika ljudskog vidnog sustava. Ljudski vidni sustav teško uočava razlike u svjetlini u regijama koje nisu kontinuirane. Iz ovog razloga broj diskretnih razina svjetline koji se koristi za razlikovanje podataka treba biti malen. Maksimalni preporučeni broj različitih razina svjetline koje se koriste u razlikovnoj ulozi boje je pet. U slučaju da se za prikaz koriste različite nijanse sive, preporučeni broj diskretnih razina je dvije do četiri.

Uz diskretizaciju boja svjetlinom može se koristiti i diskretizacija na temelju kanala zasićenja. Ovakav način diskretizacije ima iste probleme kao i diskretizacija svjetlinom boje. Maksimalni broj preporučenih diskretnih razina za ovaj način prikaza je 3 boje.

Kanal zasićenja često se koristi u kombinaciji s prikazom u veličini. U ovom slučaju preporučuju se samo dvije razine zasićenja. Na malim veličinama preporučuju se visoko zasićene boje dok se na velikim veličinama preporuča korištenje nisko zasićenih boja.

Kada se boja koristi u identifikacijskoj ulozi za razlike se koristi kanal nijanse   
(Slika 17). Ovakav način identificiranja vrlo je efektivan kada se podaci prikazuju u kategorijama ili kada se podaci grupiraju. Kanal nijanse najefektivniji je način kategorizacije podataka uz prostorni prikaz podataka. Kanal nijanse također je usko povezan s veličinom prikaza. Na manjim prikazima teže je uočiti razlike u nijansi dok je na većim prikazima lakše uočiti razlike. Slično kao i kod kanala koji se koriste kod razlikovne uloge ljudski vizualni sustav lakše uočava razlike u kontinuiranim nego odvojenim površinama. Također maksimalni broj različitih boja koji se preporuča za identifikacijsku ulogu boje je sedam.



Slika Boja u identifikacijskoj ulozi

Kanali zasićenja i svjetline imaju implicitni poredak i ljudi mogu boje koje se razlikuju prema ovim kanalima svrstati u red. Na primjer moguće je poredati boje prema količini bijele to jest crne koja se u njima nalazi. Zbog ove činjenice su ovi kanali pogodni za razlikovnu ulogu. S druge strane kanal nijanse nema implicitni poredak, na primjer nije intuitivno kako se u red mogu poredati plava, crvena i žuta. Zbog ove činjenice ovaj je kanal pogodan za identifikacijsku ulogu boje.

## Vizualizacija uz pomoć alata Power BI

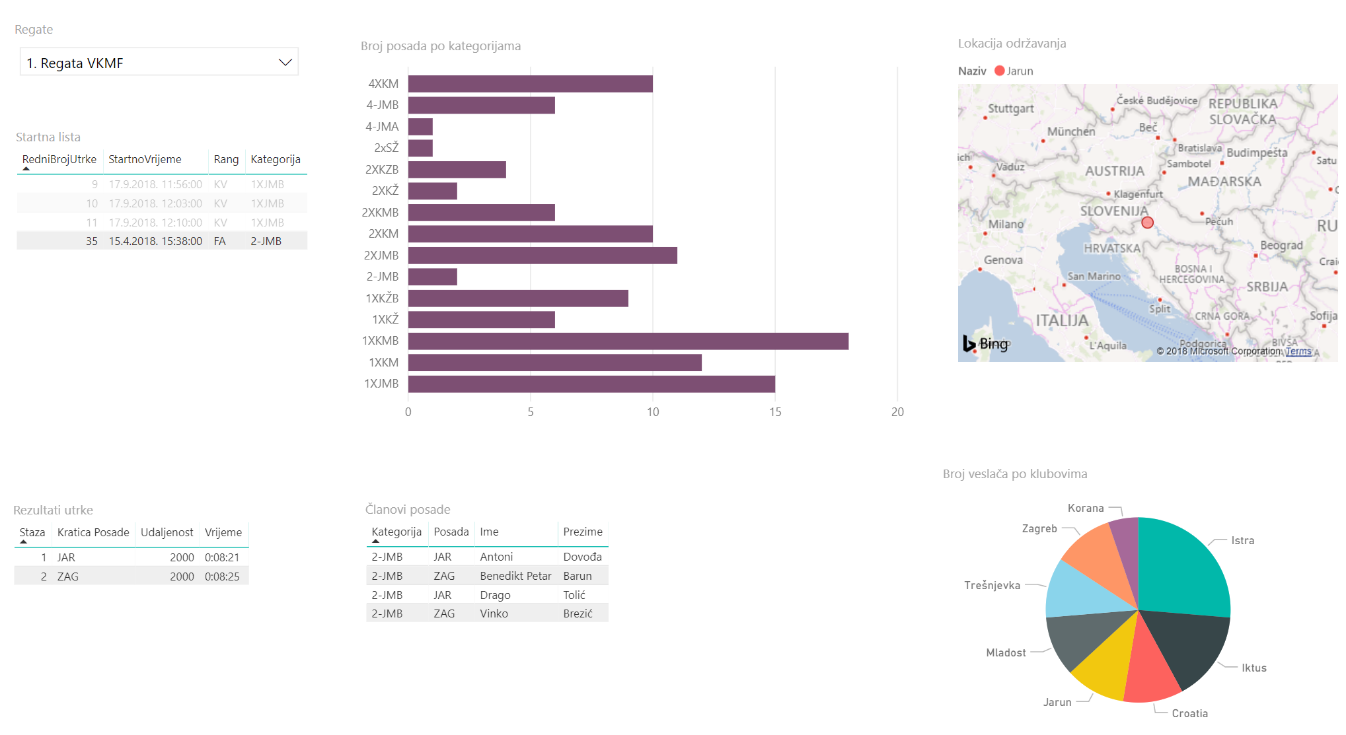
Alat *„Power BI“* je alat za vizualizaciju podataka koji je proizvela tvrtka Microsoft. Ovaj alat postoji u sedam različitih inačica. Od svih inačica besplatna je samo inačica *„Power BI Desktop“* koja je ujedno i inačica s najmanjim broj funkcionalnosti. Iako inačica *„Power BI Desktop“* imaju najmanji broj funkcionalnosti, funkcionalnosti koje posjeduje dovoljne su za kvalitetnu vizualizaciju podataka. Ova je inačica alata *„Power BI“* korištena za analizu i vizualizaciju podataka u sustavu za praćenje i analizu performansi veslača.

Alat *„Power BI“*  moguće je povezati na razne izvore podataka koji se mogu koristiti za vizualizaciju. Neki od izvora podataka na koje se *„Power BI“*  može povezati su: Microsoft Excel dokumenti, tekstualni CSV dokumenti, dokumenti koji sadrže zapise u JSON formatu, SQL server baze podataka, Access baze podataka, IBM DB2 i Informix baze podataka, MySQL baze podataka, PostgresSQL baze podataka, sustav Impala, sustav Hadoop, sustav Spark i mnogi drugi. S obzirom na činjenicu da alat *„Power BI“*  podržava veliki broj različitih tipova izvora podataka on je i vrlo raširen alat za vizualizaciju. Velika prednost ovog alata je što je moguće koristiti izvore različitog tipa unutar jednog projekta. Na taj način lagano se i intuitivno mogu vizualizirati podaci koji nisu spremljeni na istom mjestu.

Poslije učitavanja podataka s izvora *„Power BI“* nudi mogućnost stvaranja veza između dohvaćenih podataka. Veze je moguće samostalno stvoriti, ali u slučaju da su atributi u više dohvaćenih relacija istog imena alat će vezu samostalno stvoriti. Funkcionalnost automatskog stvaranja veze treba koristiti s oprezom u slučaju da izvor podatka sadrži malenu količinu podataka. Sustav će ponekad krivo procijeniti brojnost veze između dvije relacije. Ako je brojnost veze krivo procijenjena alat neće dopustiti učitavanje novih podataka koji se ne podudaraju s trenutno definiranim vezama.

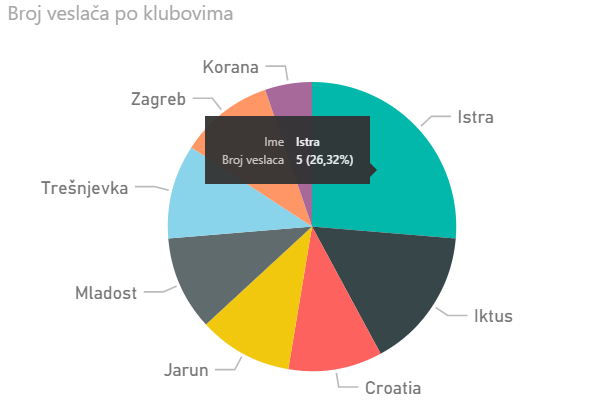
Uz veliki broj različitih izvora podataka koji se mogu koristiti, alat *„Power BI“* nudi i podršku za najčešće korištene vrste grafova. U slučaju da pred instalirana podrška za vizualizaciju nije dovoljna, *„Power BI“*  nudi mogućnost dodavanja novih vrsta grafova u projekt. Ovaj je proces olakšan pristupom trgovini grafova iz koje je moguće uvesti grafove koje je napravila programerska zajednica. Grafovi koje je napravila zajednica nude ograničenu podršku i u projekt se dodaju na vlastitu odgovornost.

Unutar *„Power BI“* projekta grafovi se u logičke i fizičke cjeline povezuju uz pomoć ploča (*engl. dashborad*). Svaka ploča predstavlja praznu stranicu na koju se mogu rasporediti grafovi i filteri (Slika 18).



Slika Ploča s podacima o pojedinoj regati

Grafovi vizualno prikazuju podatke, a filteri se koriste kako bi se prikaz podatka ograničio prema određenom ključu. Slika 18 prikazuje ploču pod nazivom „Regata“. Ova ploča prikazuje podatke koji su direktno povezani s pojmom regate. Na ovoj ploči filter je entitet „Regata“. Odabirom pojedine regate prikazuju se samo podaci za nju. Svi ostali entiteti na ploči su grafovi. Grafovi su također povezani vezama. Na primjer odabirom utrke na startnoj listi prikazuju se rezultati odabrane utrke, a odabirom pojedinog rezultata prikazuju se članovi posade koji su ostvarili odabrani rezultat. Svi grafovi unutar alata *„Power BI“* su interaktivni. Prilikom interakcije s grafovima na njima se prikazuju dodatni podaci koji se ne prikazuju u početnom stanju (Slika 19).



Slika Prikaz grafa prilikom interakcije

Slika 19 prikazuje graf s podacima o broju veslača koji su veslali za pojedini klub na pojedinoj regati. U početnom stanju ne prikazuje se broj veslača već samo omjer veslača po klubovima. Prilikom interakcije s grafičkim prikazom prikazuje se prozor koji prikazuje točan broj veslača koji je veslao na regati za traženi klub.

U sustavu za praćenje i analizu performansi veslača grafički prikazi podijeljeni su u četiri logičke cjeline. Svaka cjelina predstavljena je jednom pločom s grafovima i filterima.

Prethodno prikazana ploča pod nazivom Regata prikazuje tablice koje sadrže podatke o startnoj listi, rezultatima i članovima posade. Također sadrži grafove koji prikazuju broj veslača prema kategorijama i klubovima za koje su veslači veslali. Uz ove grafove na ploči se nalazi karta koja prikazuje lokaciju na kojoj je regata odveslana.

Ploča naziva Veslač sadrži podatke koji se odnose na pojedinog veslača. Uz filter za odabir veslača ploča sadrži grafičke prikaze visine i težine veslača kroz vrijeme, prikaz svih kategorija u kojima se veslač natjecao i graf s prikazom svih veslača s kojima je odabrani veslač veslao skupa na natjecanjima. Uz ove grafove na ovoj se ploči nalazi vremenska crta koja prikazuje povijest klubova za koje je veslač bio registriran. Uz vremensku crtu na ploči se nalazi karta koja prikazuje sve lokacije na kojima je veslač veslao te koliko je često veslao na pojedinoj lokaciji. Uz sve grafičke prikaze ova ploča sadrži i tablicu u kojoj pišu detaljna vremena koja je veslač postigao na svim dosadašnjim natjecanjima.

Ploča s imenom Klub sadrži podatke dobivene analizom podataka vezanih za veslačke klubove. Ploča sadrži filter za odabir kluba za kojeg je potrebno prikazati podatke i tri grafa. Grafovi koji se nalaze na ovoj ploči su: prikaz broja veslača koji su registrirani u klubu grupirani prema starosnoj kategoriji u kojoj se trenutno nalaze, prikaz broja osvojenih medalja na natjecanjima u zadnjih pet godina i prikaz broja veslača koji su bili registrirani u klubu na početku sezone kroz zadnjih pet godina.

Ploča imena trening sadrži podatke o treninzima za pojedine veslače. Ova ploča sadrži filter s vrijednostima veslača i 5 grafova. Jedan od grafova prikazuje vremena izveslana na predikcijskom treningu na 2000 metara, vremena su grupirana prema rednom broju izveslane dionice i prikazana na osi na kojoj se nalazi datum održavanja treninga. Uz ovaj graf nalazi se graf koji koristi iste ulazne podatke kao i prethodni graf, ali ovaj graf prikazuje predviđeno vrijeme veslanja koje se temelji na rezultatima predikcijskog treninga na 2000 metara. Uz ovaj par grafova postoji još jedan par grafova koji prikazuje istu analizu ali za predikcijski trening na 500 metara. Na ploči se također nalazi i graf koji prikazuje rezultate intervalnog treninga na 8000, 6000 i 4000 metara. Rezultati su na ovome grafu grupirani prema odveslanoj dionici i na osi su prikazani kroz datum kada je odveslan trening.

# Zaključak

Razvoj informacijskog sustava koji će biti korišten za praćenje i analizu podataka traži vrlo dobro poznavanje domene za koju se sustav radi. Zbog velikog broja zainteresiranih strana postoji i veliki broj sličnih zahtjeva. Takve zahtjeve potrebno je grupirati i spojiti u jedan ili nekoliko zahtjeva. Ovaj proces je zahtjeva jer postoji velika mogućnost da dio zahtjeva neće biti ispunjen ako se zahtjevi nekvalitetno grupiraju.

Veliki problem kod dizajniranja ovakvog tipa sustava je izvor podataka. Podaci koji se prikupljaju nisu normalizirani i ponekad je potrebno podatke ručno unositi i normalizirati što iziskuje veliki vremenski trošak. Unatoč tome nakon unosa podataka analizu je moguće kvalitetno obaviti pod uvjetom da su podaci kvalitetno normalizirani. Prilikom unosa podataka veliki problem stvaraju podaci koji odudaraju od službeno propisanog oblika u kojima bi se trebali nalaziti. Takve podatke ponekad nije ni moguće unesti u sustav, a i oni koje je moguće unesti često su upitne točnosti.

Kada su podaci uneseni u sustav nad njima se mogu provesti analiza i vizualizacija. Veći dio analize podataka najbolje je obaviti unutar same baze podataka kako bi se vizualizacija mogla lagano napraviti korištenjem različitih alata. Na ovaj način izbjegavaju se ograničenja vizualizacijskih alata. Analizu i grupiranje podataka najbolje je napraviti korištenjem SQL procedura i pogleda. Iako je dio analize napravljen u bazi podataka krajnja analiza i vizualizacija se obavljaju korištenjem alata za vizualizaciju. Vizualizaciju podataka potrebno je obaviti prema pravilima i smjernicama koje pruža teorija vizualizacije. Vizualizirati podatke je lagano, ali vizualizirati podatke pravilno je teško. Svrha vizualizacije nije prikaz podataka slikama i bojama. Odabir pravilnog tipa prikaza podataka vrlo je bitan jer se vizualizacijom pokušava promijeniti ljudska percepcija određenog problema. Vizualizacija podataka je moćan alat koji je potrebno pravilno koristiti kako bi se došlo da najboljih rezultata.

# Literatura

1. CERT, Modeli zaštite baza podataka, 25.10.2006. , *Sigurnost sustava za upravljanje bazama podataka,* <http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2006-10-171.pdf> , 10.5.2016.
2. Jesse James Garrett, Defining Ajax , 18.02.2005., *Ajax: A New Approach to Web Applications,* <https://web.archive.org/web/20080702075113/http://www.adaptivepath.com/ideas/essays/archives/000385.php> , 27.05.2016.
3. Ian Fette i Alexey Melnikov , The WebSocket Protocol, 12.2011., *The WebSocket Protocol,* [*https://tools.ietf.org/html/rfc6455*](https://tools.ietf.org/html/rfc6455) *,* 28.05.2016.
4. Time Dierks i Christopher Allen, The TLS Protocol, 01.1999. , *The TLS Protocol,* [*http://www.ietf.org/rfc/rfc2246.txt*](http://www.ietf.org/rfc/rfc2246.txt) *,* 28.05.2016.
5. SQL Injection, <http://www.w3schools.com/sql/sql_injection.asp> , 30.5.2016.
6. CERT, XSS napad, 21.01.2008., *Provjera XSS i SQL Injection ranjivosti Exploit Me skupom alata,* [*http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2008-01-215.pdf*](http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2008-01-215.pdf) *, 1.6.2016.*

**Oblikovanje sustava za praćenje i analizu performansi veslača**

**Sažetak**

Ključne riječi: vizualizacija, informacijski sustav, analiza, graf, podatak, veslanje

**Creating a system for monitoring and analyzing performance of rowers Abstract**

Keywords: visualization, information system, analysis, chart, data, rowing