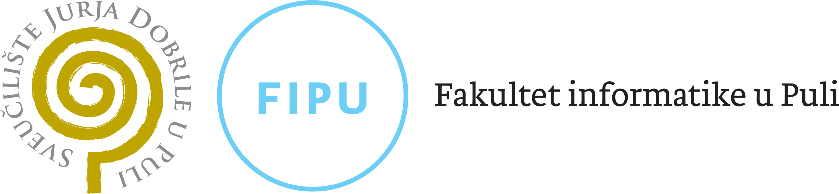
Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli



**TIM 10: AUTOMOBILIZAM**

David Tarčuki

Patrik Fabijanić

David Kokalović

Smjer : Informatika

Kolegij : Baze podataka II

Mentor : doc. Dr. sc. Goran Oreški

Asistent : mag. inf. Romeo Šajina

Pula, 2023/2024. godina

**SADRŽAJ**

[**1.** **UVOD** 1](#_Toc156220677)

[**2.** **ER-DIJAGRAM** 2](#_Toc156220678)

[**3.** **EER-DIJAGRAM (MYSQL WORKBENCH)** 4](#_Toc156220679)

[**4.** **TABLICE** 5](#_Toc156220680)

[**4.1.** **Tablica sponzor** 5](#_Toc156220681)

[**4.2.** **Tablica mehanicar** 5](#_Toc156220682)

[**4.3.** **Tablica staza** 6](#_Toc156220683)

[**4.4.** **Tablica nagrada** 6](#_Toc156220684)

[**4.5.** **Tablica tim** 7](#_Toc156220685)

[**4.6.** **Tablica prvenstvo** 8](#_Toc156220686)

[**4.7.** **Tablica timski\_mehanicar** 9](#_Toc156220687)

[**4.8.** **Tablica motor** 9](#_Toc156220688)

[**4.9.** **Tablica trener** 10](#_Toc156220689)

[**4.10.** **Tablica timski\_sponzor** 11](#_Toc156220690)

[**4.11.** **Tablica automobil** 11](#_Toc156220691)

[**4.12.** **Tablica utrka** 12](#_Toc156220692)

[**4.13.** **Tablica automobil\_kod\_mehanicara** 13](#_Toc156220693)

[**4.14.** **Tablica vozac** 14](#_Toc156220694)

[**4.15.** **Tablica rezultat** 15](#_Toc156220695)

[**4.16.** **Tablica vozacev\_automobil** 16](#_Toc156220696)

[**4.17.** **Tablica vozi\_krug** 16](#_Toc156220697)

[**4.18.** **Tablica se\_kvalificira** 17](#_Toc156220698)

[**4.19.** **Tablica ostvaruje\_rezultat** 18](#_Toc156220699)

[**5.** **VIEW** 19](#_Toc156220700)

[**5.1.** **View** 19](#_Toc156220701)

[**5.2.** **View** 20](#_Toc156220702)

[**5.3.** **View** 20](#_Toc156220703)

[**5.4.** **View** 21](#_Toc156220704)

[**5.5.** **View** 21](#_Toc156220705)

[**6.** **PROCEDURE** 22](#_Toc156220706)

[**6.1.** **Procedura** 22](#_Toc156220707)

[**6.2.** **Procedura** 23](#_Toc156220708)

[**6.3.** **Procedura** 25](#_Toc156220709)

[**6.4.** **Procedura** 26](#_Toc156220710)

[**6.5.** **Procedura** 27](#_Toc156220711)

[**6.6.** **Procedura** 28](#_Toc156220712)

[**6.7.** **Procedura** 29](#_Toc156220713)

[**6.8.** **Procedura** 30](#_Toc156220714)

[**6.9.** **Procedura** 31](#_Toc156220715)

[**6.10.** **Procedura** 33](#_Toc156220716)

[**6.11.** **Procedura** 34](#_Toc156220717)

[**6.12.** **Procedura** 35](#_Toc156220718)

[**7.** **OKIDAČI** 36](#_Toc156220719)

[**7.1.** **Okidač** 36](#_Toc156220720)

[**7.2.** **Okidač** 37](#_Toc156220721)

[**7.3.** **Okidač** 38](#_Toc156220722)

[**7.4.** **Okidač** 39](#_Toc156220723)

[**7.5.** **Okidač** 39](#_Toc156220724)

[**7.6.** **Okidač** 40](#_Toc156220725)

[**7.7.** **Okidač** 41](#_Toc156220726)

[**7.8.** **Okidač** 41](#_Toc156220727)

[**7.9.** **Okidač** 42](#_Toc156220728)

[**7.10.** **Okidač** 43](#_Toc156220729)

[**8.** **FUNKCIJE** 44](#_Toc156220730)

[**8.1.** **Funkcija** 44](#_Toc156220731)

[**8.2.** **Funkcija** 45](#_Toc156220732)

[**8.3.** **Funkcija** 45](#_Toc156220733)

[**8.4.** **Funkcija** 46](#_Toc156220734)

[**8.5.** **Funkcija** 46](#_Toc156220735)

[**8.6.** **Funkcija** 47](#_Toc156220736)

[**8.7.** **Funkcija** 48](#_Toc156220737)

[**8.8.** **Funkcija** 48](#_Toc156220738)

[**8.9.** **Funkcija** 49](#_Toc156220739)

[**8.10.** **Funkcija** 50](#_Toc156220740)

[**8.11.** **Funkcija** 50](#_Toc156220741)

[**9.** **TRANSAKCIJA** 52](#_Toc156220742)

[**9.1.** **Transakcija** 52](#_Toc156220743)

[**9.2.** **Transakcija** 53](#_Toc156220744)

[**9.3.** **Transakcija** 54](#_Toc156220745)

[**9.4.** **Transakcija** 55](#_Toc156220746)

[**9.5.** **Transakcija** 56](#_Toc156220747)

[**9.6.** **Transakcija** 57](#_Toc156220748)

[**10.** **INICIJALIZACIJA I KONFIGURACIJA** 58](#_Toc156220749)

[**11.** **RUTE I FUNKCIONALNOSTI** 58](#_Toc156220750)

[**11.1.** **Rute za Sponzore** 59](#_Toc156220751)

[**11.2.** **Rute za Timove** 59](#_Toc156220752)

[**11.3.** **Rute za Vozače** 59](#_Toc156220753)

[**11.4.** **Rute za Automobile** 59](#_Toc156220754)

[**11.5.** **Rute za Utrke** 60](#_Toc156220755)

[**11.6.** **Rute za Staze** 60](#_Toc156220756)

[**11.7.** **Rute za Nagrade** 60](#_Toc156220757)

[**11.8.** **Rute za Prvenstva** 60](#_Toc156220758)

[**11.9.** **Rute za Rezultate** 61](#_Toc156220759)

[**11.10.** **Dodatne Rute** 61](#_Toc156220760)

[**12.** **INTERAKCIJA S BAZOM PODATAKA** 61](#_Toc156220761)

[**13.** **PORUKE I REDIREKCIJA** 61](#_Toc156220762)

[**14.** **RENDERIRANJE HTML PREDLOŽAKA** 62](#_Toc156220763)

[**15.** **POKRETANJE APLIKACIJE** 62](#_Toc156220764)

[**16.** **VIEW POCETNE\_POZICIJE** 62](#_Toc156220765)

[**17.** **PROCEDURA OBRACUN\_BODOVA** 63](#_Toc156220766)

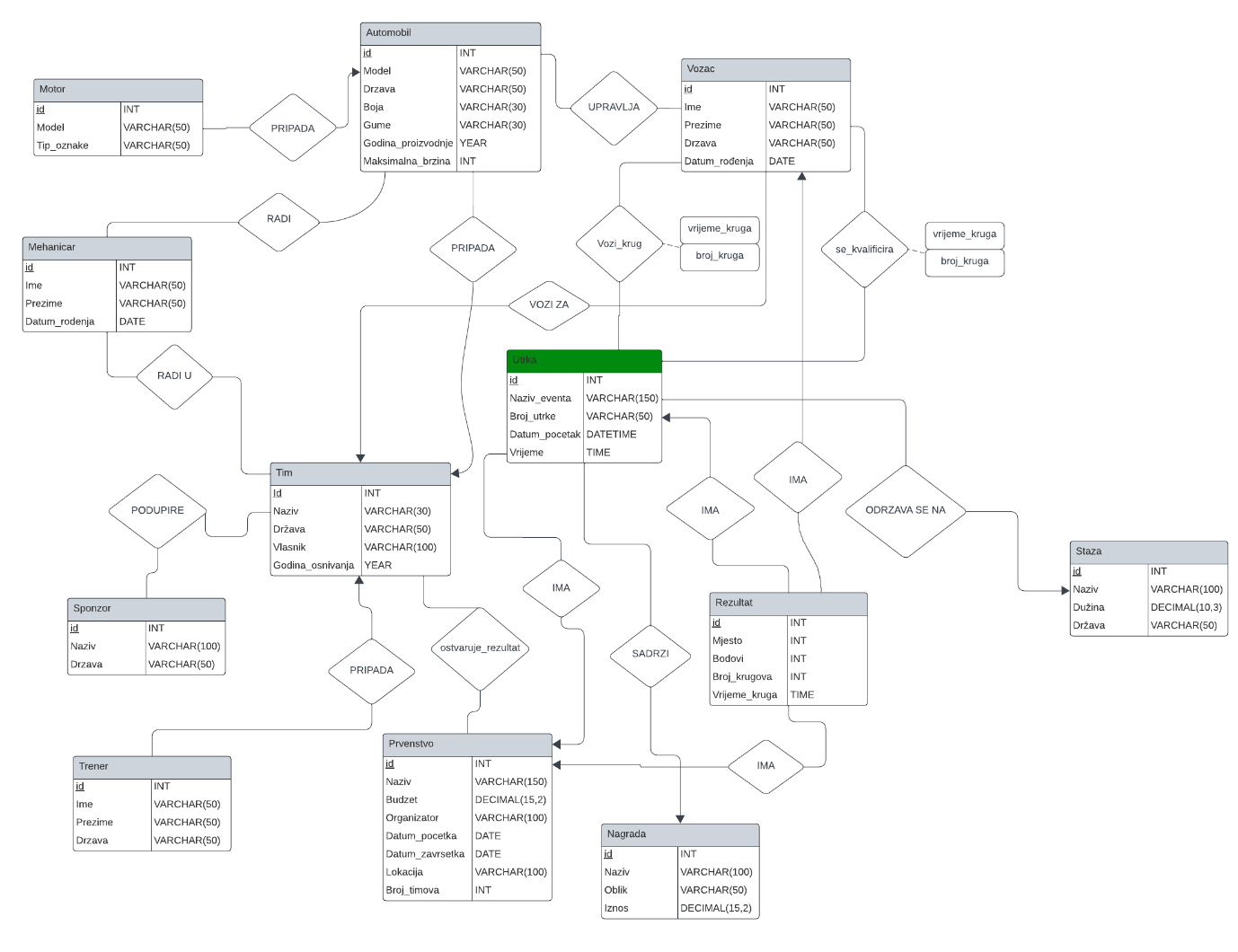
# **UVOD**

U ovoj dokumentaciji detaljno ćemo predstaviti naš projekt iz kolegija Baze podataka 2, pod nazivom 'Automobilizam'. Cilj projekta je razviti sveobuhvatnu i efikasnu bazu podataka specijaliziranu za automobilske utrke, koja obuhvaća sve ključne komponente ovog uzbuđujućeg sporta - od timova, vozača, staza, do sponzora i mehaničara.

Ovaj projekt je zamišljen kao temelj za upravljanje, organizaciju i analizu široke palete podataka povezanih s automobilizmom. Fokusirajući se na detaljnu strukturu i organizaciju, naša baza podataka dizajnirana je da efikasno obrađuje i analizira različite aspekte ovog dinamičnog sporta. Kroz ovaj projekt, istražili smo i definirali osnovne relacije koje uključuju specifikacije automobila, informacije o trkaćim stazama, te podatke o događanjima u svijetu automobilizma.

Poseban naglasak stavljamo na tehnički aspekt projekta, gdje ćemo objasniti strukturiranje našeg SQL koda za kreiranje tablica i veza. Također, predstavit ćemo kako su funkcije, okidači i transakcije integrirani u sustav za osiguranje njegove efikasnosti i pouzdanosti.

# **ER-DIJAGRAM**



ER dijagram prikazuje strukturu baze podataka. Svaki pravokutnik označava entitet ili tablicu u bazi podataka, dok linije između pravokutnika predstavljaju veze između entiteta. Ključni entiteti i njihovi atributi su:

**Automobili**: Sadrži informacije o automobilima koji sudjeluju u utrkama. Atributi uključuju ID, model, državu, boju, snagu, godinu proizvodnje i maksimalnu brzinu.

**Vozač**: Ovaj entitet sadrži podatke o vozačima, uključujući njihov ID, ime, prezime, državu i datum rođenja.

**Mehaničar**: Sadrži informacije o mehaničarima koji rade na automobilima, s atributima poput ID-a, imena, prezimena i datuma rođenja.

**Utrka**: Centralni entitet koji predstavlja događaj utrke, s atributima kao što su ID utrke, naziv, broj krugova, datum početka i vrijeme trajanja.

**Tim**: Entitet koji predstavlja timove uključene u utrke s atributima kao što su ID, naziv, država i osnivačka godina.

**Sponzor**: Entitet za sponzore s atributima poput ID-a, naziva i države.

**Trener**: Entitet koji sadrži podatke o trenerima, uključujući njihov ID, ime, prezime i državu.

**Prvenstvo**: Entitet koji predstavlja prvenstvo u kojem utr

**Prvenstvo**: Entitet koji predstavlja prvenstvo u kojem utrke imaju mjesto, sa atributima kao što su ID prvenstva, naziv, budžet, organizator, datumi početka i završetka, te lokacija.

**Staza**: Ovaj entitet opisuje staze na kojima se utrke održavaju, s atributima kao što su ID, naziv, duljina i država.

**Rezultat**: Entitet koji sadrži rezultate utrka, uključujući ID, mjesto koje je vozač zauzeo, broj bodova, broj krugova i vrijeme kruga.

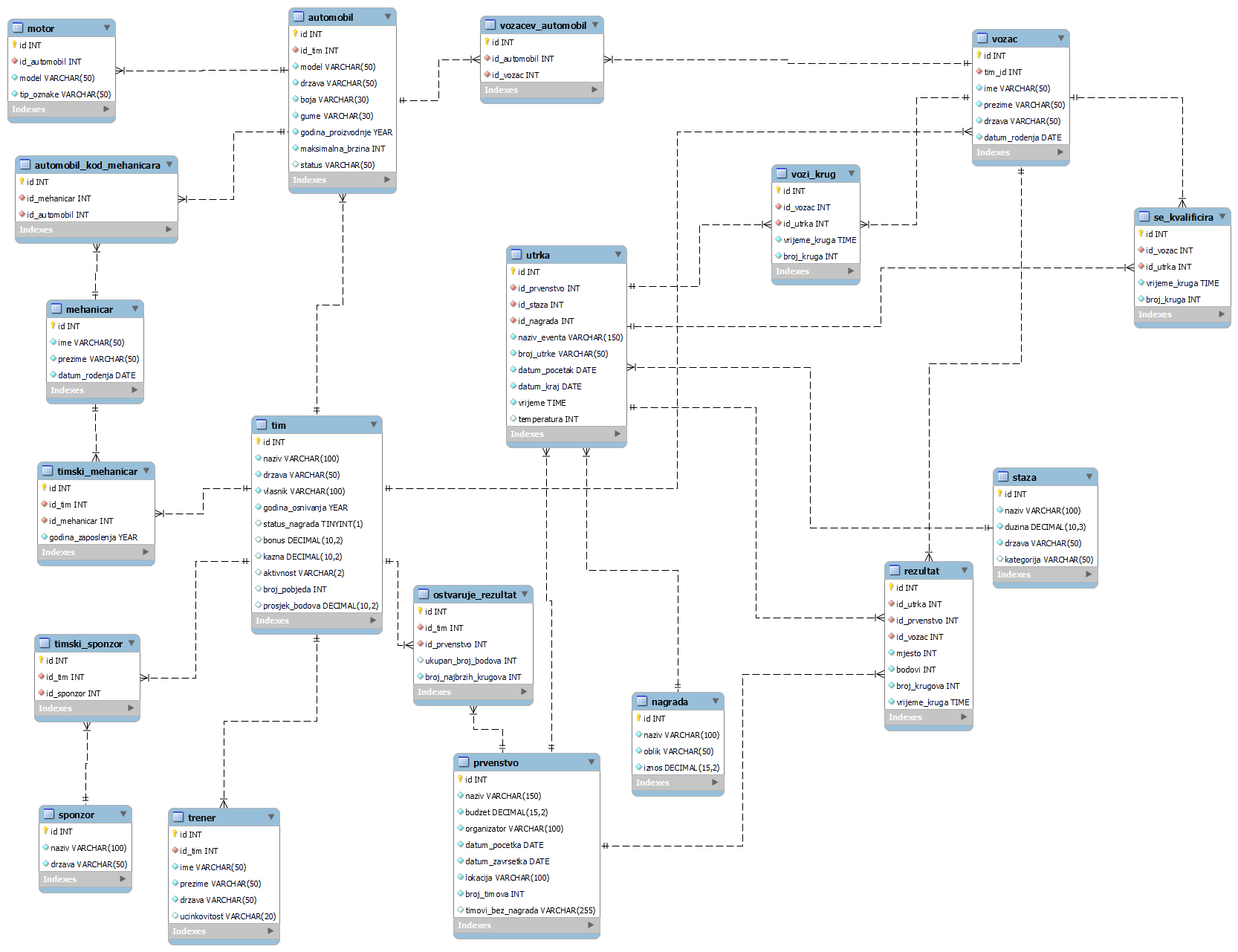
**Nagrada**: Sadrži informacije o nagradama koje se dodjeljuju, s atributima poput ID-a, naziva, oblika i iznosa.

Veze između entiteta ukazuju na to kako se podaci u jednom entitetu povezuju s podacima u drugom. Na primjer, veza između "Vozač" i "Automobili" (upravlja) sugerira da svaki vozač upravlja određenim automobilom. Isto tako, veza između "Tim" i "Mehaničar" (radi u) pokazuje da mehaničari rade za određene timove.

Kardinaliteti veza, koje su označene na linijama (poput jedan na mnogo, mnogo na jedan), govore nam o prirodi odnosa. Na primjer, jedan tim može imati više mehaničara, a jedan mehaničar radi za jedan tim, što je označeno kao "radi u" veza.

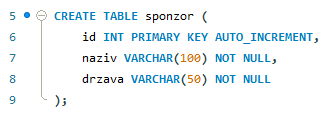
Ove informacije pružaju pregled organizacije podataka u bazi i povezivanja tablica putem različitih veza. To omogućuje kompleksne upite i analize, poput dobivanja svih automobila voženih određenim vozačem ili pregleda rezultata koje je postigao jedan tim tijekom različitih utrka.

# **EER-DIJAGRAM (MYSQL WORKBENCH)**



# **TABLICE**

## **Tablica sponzor**



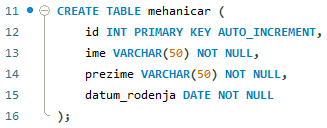
id: Jedinstveni identifikacijski broj svakog sponzora, koji se automatski povećava za svaki novi zapis.

naziv: Ime sponzora, do 100 znakova.

drzava: Država sponzora, do 50 znakova.

Stupci naziv i drzava moraju biti popunjeni (**NOT** **NULL**).

## **Tablica mehanicar**



id: Jedinstveni identifikacijski broj svakog mehaničara, koji se automatski povećava za svaki novi zapis.

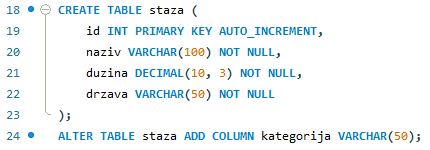
ime: Ime mehaničara, ograničeno na 50 znakova.

prezime: Prezime mehaničara, ograničeno na 50 znakova.

datum\_rodenja: Datum rođenja mehaničara.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

## **Tablica staza**



id: Jedinstveni identifikacijski broj svake staze, koji se automatski povećava za svaki novi zapis.

naziv: Ime staze, ograničeno na 100 znakova.

duzina: Duljina staze, predstavljena decimalnim brojem s do 10 znamenki ukupno i 3 decimalna mjesta.

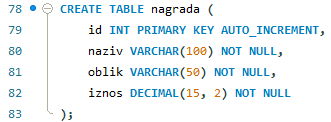
drzava: Država u kojoj se staza nalazi, ograničena na 50 znakova.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **ALTER** **TABLE** za dodavanje još stupaca:

* kategorija: Kategorija staze, ograničena na 50 znakova.

## **Tablica nagrada**



id: Jedinstveni identifikacijski broj za svaku nagradu, koji se automatski povećava za svaki novi zapis.

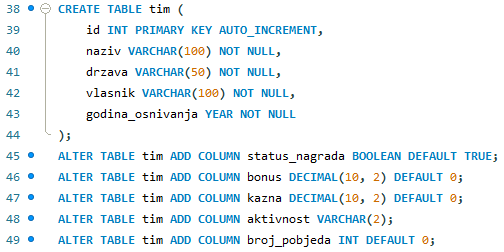
naziv: Ime nagrade, ograničeno na 100 znakova.

oblik: Opis oblika nagrade, ograničen na 50 znakova.

iznos: Vrijednost nagrade u monetarnom iznosu, predstavljena kao decimalni broj s maksimalno 15 znamenki i 2 decimalna mjesta.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

## **Tablica tim**



id: Jedinstveni identifikacijski broj za svaki tim, koji se automatski povećava za svaki novi zapis.

naziv: Ime tima, ograničeno na 100 znakova.

drzava: Država porijekla tima, ograničena na 50 znakova.

vlasnik: Ime vlasnika tima, ograničeno na 100 znakova.

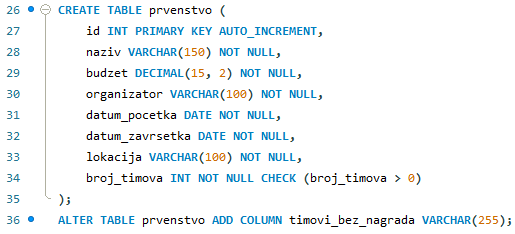
godina\_osnivanja: Godina kada je tim osnovan, predstavljena kao godina (**YEAR**).

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **ALTER TABLE** za dodavanje još stupaca:

* status\_nagrada: Booleov stupac (**BOOLEAN**) koji označava status nagrada tima. Ako je **TRUE**, tim je primio nagrade. Ako je **FALSE**, nije.
* bonus: Decimalni broj koji predstavlja bonus iznos tima, s maksimalno 10 znamenki i 2 decimalna mjesta. Zadana vrijednost je 0.
* kazna: Decimalni broj koji predstavlja kazne iznos timu, s maksimalno 10 znamenki i 2 decimalna mjesta. Zadana vrijednost je 0.
* aktivnost: Kratki tekstualni opis aktivnosti tima, ograničen na 2 znaka.
* broj\_pobjeda: Broj pobjeda tima, predstavljen kao cijeli broj. Zadana vrijednost je 0.

## **Tablica prvenstvo**



id: Jedinstveni identifikacijski broj za svako prvenstvo, koji se automatski povećava za svaki novi zapis.

naziv: Ime prvenstva, ograničeno na 150 znakova.

budzet: Ukupni budžet prvenstva, predstavljen kao decimalni broj s maksimalno 15 znamenki i 2 decimalna mjesta.

organizator: Ime organizatora prvenstva, ograničeno na 100 znakova.

datum\_pocetka: Datum početka prvenstva.

datum\_zavrsetka: Datum završetka prvenstva.

lokacija: Mjesto održavanja prvenstva, ograničeno na 100 znakova.

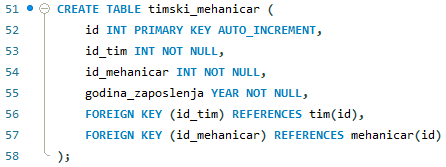
broj\_timova: Ukupan broj timova koji sudjeluju u prvenstvu, predstavljen kao cijeli broj. Postavljen je uvjet da mora biti veći od 0.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **ALTER TABLE** za dodavanje još stupaca:

* timovi\_bez\_nagrada: Tekstualni stupac, ograničen na 255 znakova.

## **Tablica timski\_mehanicar**



id: Jedinstveni identifikacijski broj za svaku poveznicu između tima i mehaničara, koji se automatski povećava za svaki novi zapis.

id\_tim: Identifikacijski broj tima, povezan s tablicom "tim".

id\_mehanicar: Identifikacijski broj mehaničara, povezan s tablicom "mehanicar".

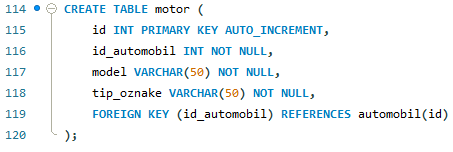
godina\_zaposlenja: Godina kada je mehaničar zaposlen u timu.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN** **KEY** povezuje id\_tim sa stupcem id u tablici "tim", osiguravajući da se svaki zapis u "timski\_mehanicar" odnosi na postojeći tim.

Naredba **FOREIGN** **KEY** povezuje id\_mehanicar sa stupcem id u tablici "mehanicar", osiguravajući da se svaki zapis u "timski\_mehanicar" odnosi na postojećeg mehaničara.

## **Tablica motor**



id: Jedinstveni identifikacijski broj za svaki motor, koji se automatski povećava za svaki novi zapis.

id\_automobil: Identifikacijski broj automobila, povezan s tablicom "automobil".

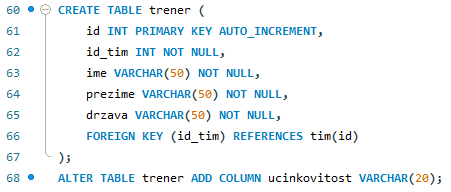
model: Naziv modela motora, ograničen na 50 znakova.

tip\_oznake: Oznaka tipa motora, ograničena na 50 znakova.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN** **KEY** povezuje id\_automobil sa stupcem id u tablici "automobil", osiguravajući da svaki motor u tablici "motor" odgovara postojećem automobilu iz tablice "automobil".

## **Tablica trener**



id: Jedinstveni identifikacijski broj za svakog trenera, koji se automatski povećava za svaki novi zapis.

id\_tim: Identifikacijski broj tima, povezan s tablicom "tim".

ime: Ime trenera, ograničeno na 50 znakova.

prezime: Prezime trenera, ograničeno na 50 znakova.

drzava: Država podrijetla trenera, ograničena na 50 znakova.

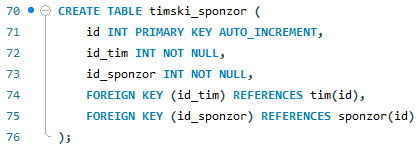
Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN** **KEY** povezuje id\_tim sa stupcem id u tablici "tim", osiguravajući da svaki trener u tablici "trener" pripada postojećem timu iz tablice "tim".

Naredba **ALTER TABLE** za dodavanje još stupaca:

* ucinkovitost: Tekstualni stupac, ograničen na 20 znakova.

## **Tablica timski\_sponzor**



id: Jedinstveni identifikacijski broj za svaki zapis o timskom sponzoru, koji se automatski povećava za svaki novi zapis.

id\_tim: Identifikacijski broj tima, povezan s tablicom "tim".

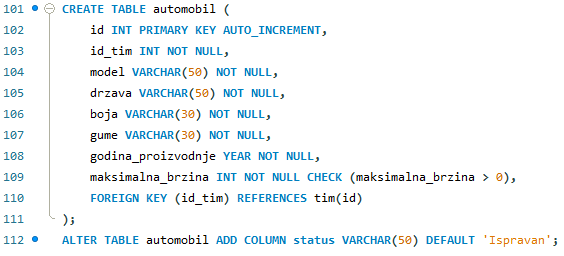
id\_sponzor: Identifikacijski broj sponzora, povezan s tablicom "sponzor".

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN** **KEY** povezuje id\_tim sa stupcem id u tablici "tim", osiguravajući da svaki zapis o timskom sponzoru pripada postojećem timu.

Naredba **FOREIGN** **KEY** povezuje id\_sponzor sa stupcem id u tablici "sponzor", osiguravajući da svaki zapis o timskom sponzoru pripada postojećem sponzoru.

## **Tablica automobil**



id: Jedinstveni identifikacijski broj za svaki automobil, koji se automatski povećava za svaki novi zapis.

id\_tim: Identifikacijski broj tima, povezan s tablicom "tim".

model: Model automobila, ograničen na 50 znakova.

drzava: Država podrijetla automobila, ograničena na 50 znakova.

boja: Boja automobila, ograničena na 30 znakova.

gume: Vrsta guma koje koristi automobil, ograničena na 30 znakova.

godina\_proizvodnje: Godina proizvodnje automobila.

maksimalna\_brzina: Maksimalna brzina koju automobil može postići, predstavljena kao cijeli broj. Postavljen je uvjet da mora biti veći od 0.

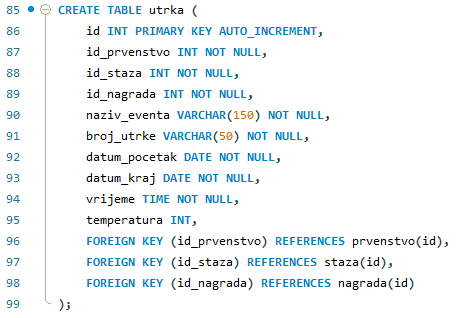
Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

**FOREIGN** **KEY** povezuje id\_tim sa stupcem id u tablici "tim", osiguravajući da svaki automobil u tablici "automobil" pripada postojećem timu iz tablice "tim".

Naredba **ALTER TABLE** za dodavanje još stupaca:

* status: Tekstualni stupac, ograničen na 50 znakova, s vrijednošću 'Ispravan'.

## **Tablica utrka**



id: Jedinstveni identifikacijski broj za svaku utrku, koji se automatski povećava za svaki novi zapis.

id\_prvenstvo: Identifikacijski broj prvenstva, povezan s tablicom "prvenstvo".

id\_staza: Identifikacijski broj staze, povezan s tablicom "staza".

id\_nagrada: Identifikacijski broj nagrade, povezan s tablicom "nagrada".

naziv\_eventa: Ime utrke, ograničeno na 150 znakova.

broj\_utrke: Jedinstveni identifikacijski broj utrke, ograničen na 50 znakova.

datum\_pocetak: Datum početka utrke.

datum\_kraj: Datum završetka utrke.

vrijeme: Vrijeme početka utrke.

temperatura: Temperatura zraka tijekom utrke, predstavljena kao cijeli broj.

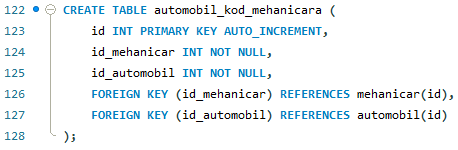
Svi stupci osim id\_utrka i temperatura moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_prvenstvo sa stupcem id u tablici "prvenstvo", osiguravajući da svaka utrka pripada postojećem prvenstvu.

Naredba **FOREIGN KEY** povezuje id\_staza sa stupcem id u tablici "staza", osiguravajući da se svaka utrka održava na postojećoj stazi.

Naredba **FOREIGN KEY** povezuje id\_nagrada sa stupcem id u tablici "nagrada", osiguravajući da se za svaku utrku dodjeljuje postojeća nagrada.

## **Tablica automobil\_kod\_mehanicara**



id: Jedinstveni identifikacijski broj za svaki zapis, koji se automatski povećava za svaki novi unos.

id\_mehanicar: Identifikacijski broj mehaničara, povezan s tablicom "mehanicar".

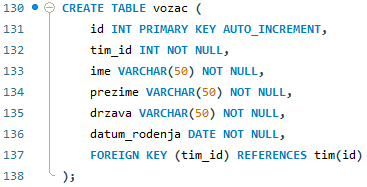
id\_automobil: Identifikacijski broj automobila, povezan s tablicom "automobil".

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_mehanicar sa stupcem id u tablici "mehanicar", osiguravajući da svaki zapis odgovara postojećem mehaničaru.

Naredba **FOREIGN KEY** povezuje id\_automobil sa stupcem id u tablici "automobil", osiguravajući da svaki zapis odgovara postojećem automobilu.

## **Tablica vozac**



id: Jedinstveni identifikacijski broj za svakog vozača, koji se automatski povećava za svaki novi zapis.

tim\_id: Identifikacijski broj tima kojem vozač pripada, povezan s tablicom "tim".

ime: Ime vozača, ograničeno na 50 znakova.

prezime: Prezime vozača, također ograničeno na 50 znakova.

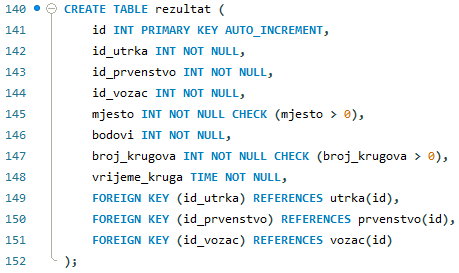
drzava: Država iz koje vozač dolazi, ograničena na 50 znakova.

datum\_rodenja: Datum rođenja vozača.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje tim\_id sa stupcem id u tablici "tim", osiguravajući da svaki vozač ima pripadnost jednom timu.

## **Tablica rezultat**



id: Jedinstveni identifikacijski broj za svaki rezultat, koji se automatski povećava za svaki novi zapis.

id\_utrka: Identifikacijski broj utrke, povezan s tablicom "utrka".

id\_prvenstvo: Identifikacijski broj prvenstva, povezan s tablicom "prvenstvo".

id\_vozac: Identifikacijski broj vozača, povezan s tablicom "vozac".

mjesto: Pozicija koju je vozač zauzeo na utrci, predstavljena kao cijeli broj. Postavljen je uvjet da mora biti veći od 0.

bodovi: Broj bodova koje je vozač osvojio na utrci, predstavljen kao cijeli broj.

broj\_krugova: Ukupan broj krugova koje je vozač završio, predstavljen kao cijeli broj. Postavljen je uvjet da mora biti veći od 0.

vrijeme\_kruga: Najbolje vrijeme koje je vozač postigao u jednom krugu, predstavljeno kao vremenski zapis.

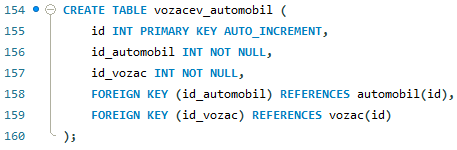
Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_utrka sa stupcem id u tablici "utrka", osiguravajući povezanost rezultata s određenom utrkom.

Naredba **FOREIGN KEY** povezuje id\_prvenstvo sa stupcem id u tablici "prvenstvo", omogućujući povezivanje rezultata s određenim prvenstvom.

Naredba **FOREIGN KEY** povezuje id\_vozac sa stupcem id u tablici "vozac", osiguravajući da svaki zapis u ovoj tablici odgovara postojećem vozaču.

## **Tablica vozacev\_automobil**



id: Jedinstveni identifikacijski broj za svaki zapis, koji se automatski povećava za svaki novi zapis.

id\_automobil: Identifikacijski broj automobila, povezan s tablicom "automobil".

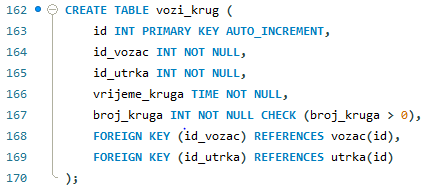
id\_vozac: Identifikacijski broj vozača, povezan s tablicom "vozac".

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_automobil sa stupcem id u tablici "automobil", osiguravajući da svaki zapis odgovara postojećem automobilu.

Naredba **FOREIGN KEY** povezuje id\_vozac sa stupcem id u tablici "vozac", osiguravajući da svaki zapis odgovara postojećem vozaču.

## **Tablica vozi\_krug**



id: Jedinstveni identifikacijski broj za svaki zapis u tablici, koji se automatski povećava za svaki novi zapis.

id\_vozac: Identifikacijski broj vozača, povezan s tablicom "vozac".

id\_utrka: Identifikacijski broj utrke, povezan s tablicom "utrka".

vrijeme\_kruga: Vrijeme potrebno za završetak jednog kruga u utrci, predstavljeno u formatu vremena (TIME).

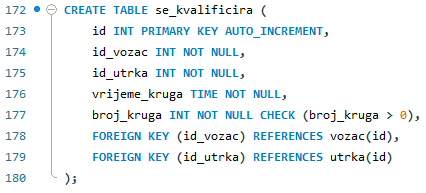
broj\_kruga: Broj kruga koji vozač vozi, predstavljen kao cijeli broj. Postavljen je uvjet da mora biti veći od 0.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_vozac sa stupcem id u tablici "vozac", osiguravajući da svaki zapis odgovara postojećem vozaču.

Naredba **FOREIGN KEY** povezuje id\_utrka sa stupcem id u tablici "utrka", osiguravajući da svaki zapis odgovara postojećoj utrci.

## **Tablica se\_kvalificira**



id: Jedinstveni identifikacijski broj za svaki zapis u tablici, koji se automatski povećava za svaki novi zapis.

id\_vozac: Identifikacijski broj vozača, povezan s tablicom "vozac".

id\_utrka: Identifikacijski broj utrke, povezan s tablicom "utrka".

vrijeme\_kruga: Vrijeme koje je vozaču potrebno za završetak određenog kruga tijekom kvalifikacija, predstavljeno u formatu vremena (TIME).

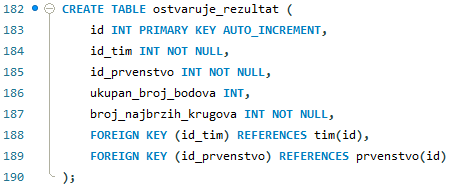
broj\_kruga: Redni broj kruga koji vozač vozi tijekom kvalifikacija, predstavljen kao cijeli broj. Postavljen je uvjet da mora biti veći od 0.

Svi stupci osim id moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_vozac sa stupcem id u tablici "vozac", osiguravajući da svaki zapis odgovara postojećem vozaču.

Naredba **FOREIGN KEY** povezuje id\_utrka sa stupcem id u tablici "utrka", osiguravajući da svaki zapis odgovara postojećoj utrci.

## **Tablica ostvaruje\_rezultat**



id: Jedinstveni identifikacijski broj za svaki zapis u tablici, koji se automatski povećava s dodavanjem svakog novog zapisa.

id\_tim: Identifikacijski broj tima, povezan s tablicom "tim".

id\_prvenstvo: Identifikacijski broj prvenstva, povezan s tablicom "prvenstvo".

ukupan\_broj\_bodova: Ukupan broj bodova koje je tim osvojio u okviru prvenstva.

broj\_najbrzih\_krugova: Broj najbržih krugova koje je tim ostvario tijekom prvenstva.

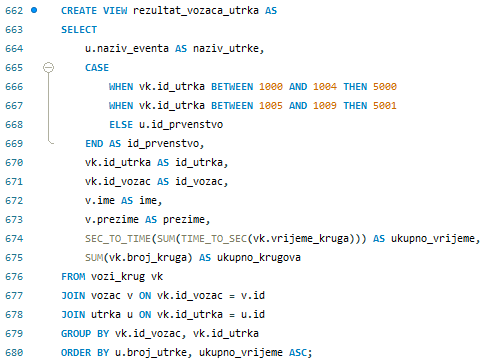
Svi stupci osim id i ukupan\_broj\_bodova moraju biti popunjeni (**NOT** **NULL**).

Naredba **FOREIGN KEY** povezuje id\_tim sa stupcem id u tablici "tim", osiguravajući da svaki zapis odgovara postojećem timu.

Naredba **FOREIGN KEY** povezuje id\_prvenstvo sa stupcem id u tablici "prvenstvo", osiguravajući da svaki zapis odgovara postojećem prvenstvu.

# **VIEW**

## **View**



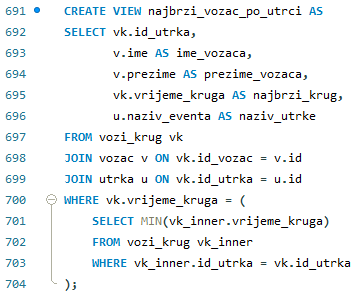
Ovaj view nazvan rezultat\_vozaca\_utrka je kreiran za prikaz organiziranih informacija o rezultatima vozača u utrkama. On objedinjuje podatke iz više tablica vozi\_krug, vozac i utrka i to čini na sljedeći način:

Prvo se selektiraju osnovni podaci kao što su naziv utrke, identifikacijski brojevi prvenstva, utrke i vozača, te imena i prezimena vozača. Kod identifikacijskog broja prvenstva koristi se poseban uvjet: ako je ID utrke između 1000 i 1004, dodjeljuje se ID prvenstva 5000, a za utrke između 1005 i 1009, ID prvenstva je 5001. U svim ostalim slučajevima koristi se ID prvenstva iz tablice utrka.

Zatim view koristi funkcije za sumiranje i obradu vremena kako bi izračunao ukupno vrijeme provedeno u trci i ukupan broj odvoženih krugova za svakog vozača. To omogućava bolji uvid u performanse svakog vozača.

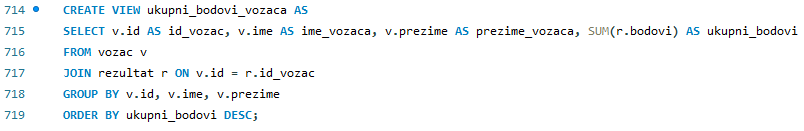
Podaci su grupirani po identifikacijskom broju vozača i utrke, što omogućava jasno razdvajanje rezultata po svakom vozaču i svakoj utrci. Na kraju, rezultati su sortirani po broju utrke i ukupnom vremenu, pružajući jasan pregled performansi vozača u svakoj utrci.

## **View**



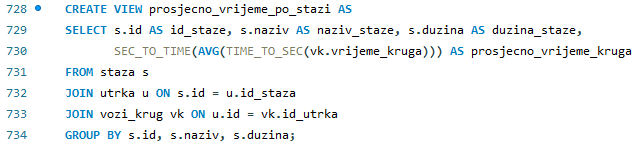
View najbrzi\_vozac\_po\_utrci je kreiran kako bi prikazao koji vozač je imao najbrži krug u svakoj utrci. On spaja podatke iz tablica vozi\_krug, vozac i utrka. U view-u se prikazuje identifikacijski broj svake utrke, ime i prezime vozača, vrijeme njihovog najbržeg kruga te naziv utrke. Za određivanje najbržeg kruga u svakoj utrci koristi se podupit koji traži minimalno vrijeme kruga unutar te utrke. Na ovaj način, view pruža jasnu informaciju o najbržem vozaču u svakoj pojedinoj utrci, skupa sa svim važnim detaljima o vozačima i utrkama.

## **View**



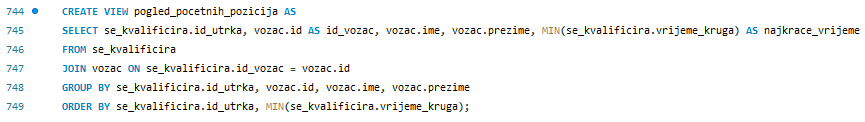
View ukupni\_bodovi\_vozaca je dizajniran da prikaže ukupan broj bodova koje je svaki vozač osvojio. Ovaj view kombinira podatke iz tablica vozac i rezultat. Za svakog vozača, prikazuje se njegov identifikacijski broj, ime i prezime, kao i suma bodova koje je osvojio. Podaci su grupirani po identifikacijskom broju, imenu i prezimenu vozača, čime se osigurava da se bodovi svakog vozača pravilno zbroje. Na kraju, view sortira vozače prema ukupnom broju osvojenih bodova, od najvećeg prema najmanjem.

## **View**



View prosjecno\_vrijeme\_po\_stazi je kreiran kako bi se prikazalo prosječno vrijeme kruga po svakoj stazi. On povezuje podatke iz tablica staza, utrka i vozi\_krug. Za svaku stazu, prikazuju se njen identifikacijski broj, naziv i dužina. Korištenjem funkcija za rad s vremenom, izračunava se prosječno vrijeme kruga na svakoj stazi. Ovo se postiže uzimanjem prosjeka vremena svih krugova voženih na toj stazi. Podaci su grupirani po identifikacijskom broju, nazivu i dužini staze, što omogućava precizno izračunavanje prosječnog vremena za svaku pojedinu stazu.

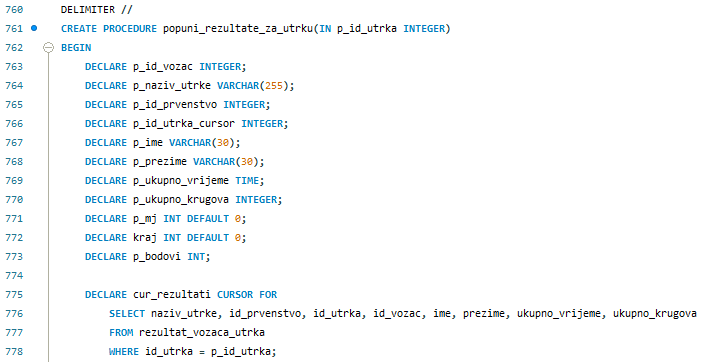
## **View**

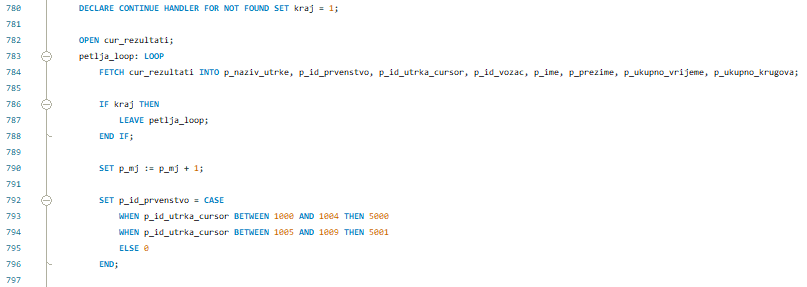


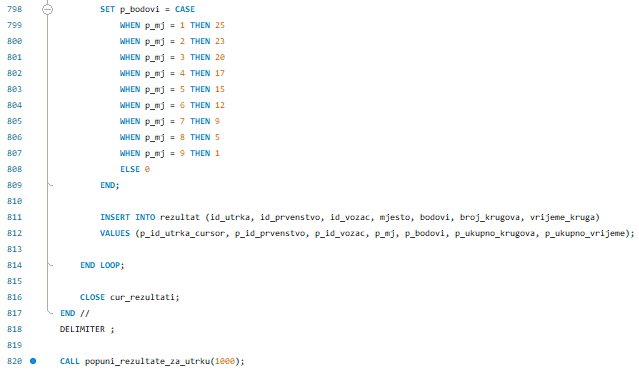
View pogled\_pocetnih\_pozicija je kreiran da prikaže najkraća vremena kruga koje su vozači postigli u kvalifikacijama za svaku utrku. On povezuje podatke iz tablica se\_kvalificira i vozac. U view-u se za svaku utrku prikazuju identifikacijski broj utrke, identifikacijski broj vozača, ime i prezime vozača, te njihovo najkraće vrijeme kruga ostvareno u kvalifikacijama. Podaci su grupirani po identifikacijskom broju utrke i vozača, kao i po imenu i prezimenu vozača, omogućavajući prikaz najboljeg rezultata svakog vozača po utrkama. Na kraju, view sortira podatke prvo po identifikacijskom broju utrke, a zatim po najkraćem vremenu kruga, što daje pregled kvalifikacijskih rezultata i početnih pozicija vozača za svaku utrku.

# **PROCEDURE**

## **Procedura**



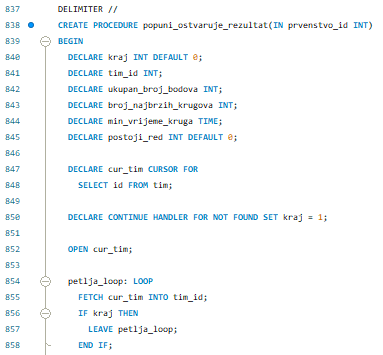


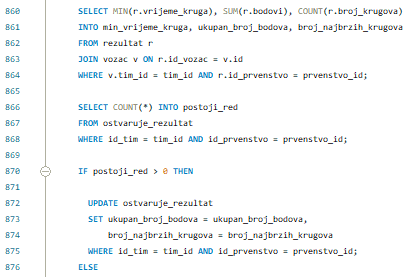


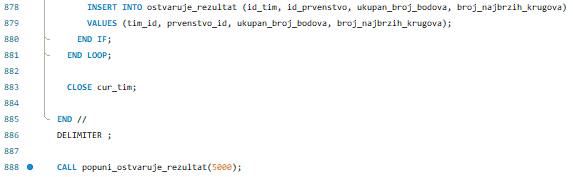
Procedura popuni\_rezultate\_za\_utrku služi za automatsko popunjavanje tablice rezultata za određenu utrku. Procedura započinje deklariranjem niza varijabli koje će koristiti za pohranu podataka o utrci i vozačima. Zatim koristi cursor za prolazak kroz svaki redak dobiven iz view-a rezultat\_vozaca\_utrka koji odgovara zadanoj utrci. U svakom prolasku kroz petlju, procedura preuzima informacije o vozačima i njihovim rezultatima, izračunava ID prvenstva na temelju ID-a utrke i bodove prema mjestu koje je vozač zauzeo. Nakon toga, rezultati se umetnu u tablicu rezultat, bilježeći detalje kao što su ID utrke, ID prvenstva, ID vozača, mjesto u utrci, bodovi, broj odvoženih krugova i ukupno vrijeme kruga. Na kraju, nakon obrade svih podataka, cursor se zatvara.

Korištenjem CALL popuni\_rezultate\_za\_utrku(1000), procedura bi automatski popunila rezultate za utrku s ID-om 1000.

## **Procedura**







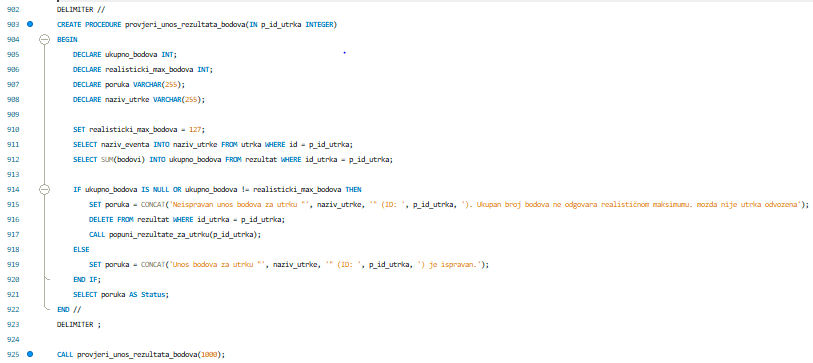
Procedura popuni\_ostvaruje\_rezultat je namijenjena za popunjavanje tablice ostvaruje\_rezultat s rezultatima svih timova za određeno prvenstvo. Procedura započinje postavljanjem varijabli, uključujući cursor za prolazak kroz sve timove u tablici tim.

Procedura zatim ulazi u petlju gdje za svaki tim dohvaća minimalno vrijeme kruga, ukupan broj bodova i broj najbržih krugova koje su ostvarili vozači tog tima u određenom prvenstvu. Ove informacije se dohvaćaju iz tablice rezultat gdje su povezane s vozačima koji pripadaju tom timu.

Nakon toga, procedura provjerava postoji li već red u tablici ostvaruje\_rezultat za taj tim i to prvenstvo. Ako postoji, procedura ažurira postojeće podatke. Ako ne postoji, procedura umetne nove podatke u tablicu.

Kada se obrade svi timovi, petlja završava i cursor se zatvara. Korištenjem CALL popuni\_ostvaruje\_rezultat(5000) će popuniti podatke za prvenstvo s ID-om 5000.

## **Procedura**

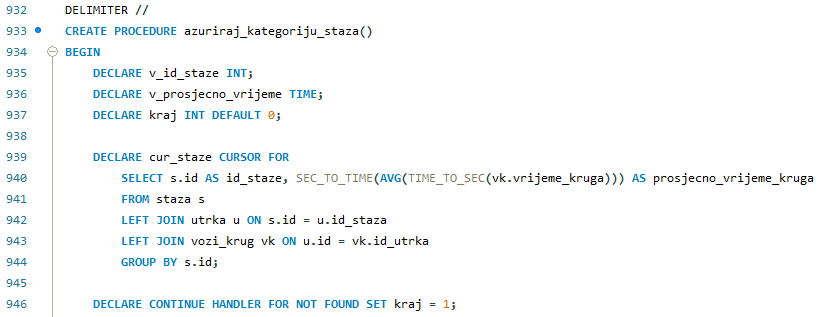


Procedura provjeri\_unos\_rezultata\_bodova služi za provjeru točnosti unesenih bodova za pojedinu utrku. Procedura prvo postavlja maksimalan realističan broj bodova koji se može osvojiti u utrci. Zatim dohvaća naziv te utrke pomoću njenog ID-a.

Procedura sumira sve bodove osvojene u toj utrci. Ako ukupan broj bodova ne odgovara postavljenom maksimumu, smatra se da je došlo do greške u unosu. U tom slučaju, svi rezultati za tu utrku se brišu, a procedura popuni\_rezultate\_za\_utrku se pokreće kako bi se ispravno popunili rezultati. Uz to, generira se poruka o neispravnom unosu.

Ako je broj bodova točan, procedura stvara poruku koja potvrđuje točnost unosa. Na kraju, procedura vraća poruku koja obavještava o statusu provjere. Korištenjem CALL provjeri\_unos\_rezultata\_bodova(1000), provjeravaju se i ispravljaju bodovi za utrku s ID-om 1000.

## **Procedura**





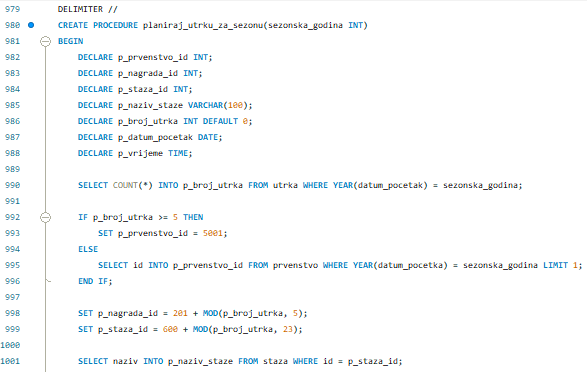
Procedura azuriraj\_kategoriju\_staza služi za ažuriranje kategorije svih staza na temelju njihovog prosječnog vremena kruga. Procedura prolazi kroz sve staze i, na temelju izračunatog prosječnog vremena kruga za svaku stazu, ažurira njihovu kategoriju.

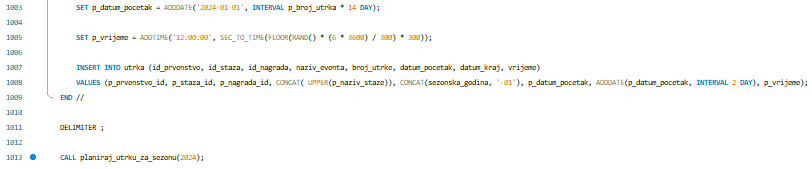
Procedura prvo otvara cursor koji prolazi kroz sve staze i izračunava prosječno vrijeme kruga za svaku od njih. Za svaku stazu, procedura zatim provjerava je li prosječno vrijeme kruga poznato. Ako nije, kategorija staze se postavlja na 'Nepoznata'.

Ako je prosječno vrijeme kruga poznato, procedura provjerava u koju kategoriju staza pripada. Ako je prosječno vrijeme manje od 1 sat i 15 minuta i 30 sekundi, staza se kategorizira kao 'Brza'. Ako je prosječno vrijeme između 1 sata 16 minuta 30 sekundi i 1 sat 19 minuta, staza se kategorizira kao 'Srednja'. U svim ostalim slučajevima, staza se kategorizira kao 'Spora'.

Nakon obrade svake staze, cursor se zatvara. Korištenjem CALL azuriraj\_kategoriju\_staza() pokreće se procedura za ažuriranje kategorija svih staza.

## **Procedura**





Procedura planiraj\_utrku\_za\_sezonu služi za planiranje i dodavanje nove utrke u kalendar utrka za određenu sezonu. Procedura uzima godinu sezone kao ulazni parametar i na temelju toga izrađuje plan za novu utrku.

Procedura počinje provjerom koliko je utrka već planirano za tu sezonu. Ako je broj utrka pet ili više, ID prvenstva se postavlja na 5001. Ako je manje, procedura odabire ID prvenstva koji odgovara godini sezone.

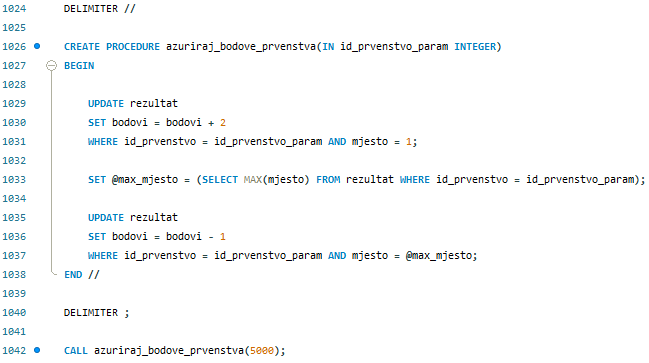
Nakon toga, procedura određuje ID nagrade i ID staze za novu utrku, koristeći matematičke formule koje ovisi o broju već planiranih utrka.

Također, procedura dohvaća naziv staze koja odgovara odabranom ID-u staze. Datum početka nove utrke postavlja se na datum koji je 14 dana nakon svake već planirane utrke u toj sezoni. Vrijeme početka utrke postavlja se nasumično unutar određenog vremenskog okvira.

Na kraju, procedura dodaje novu utrku u tablicu utrka s izračunatim detaljima kao što su ID prvenstva, ID staze, ID nagrade, naziv utrke, broj utrke, datum i vrijeme početka.

Korištenjem CALL planiraj\_utrku\_za\_sezonu(2024) pokreće se procedura koja planira i dodaje novu utrku u kalendar za sezonu 2024.

## **Procedura**



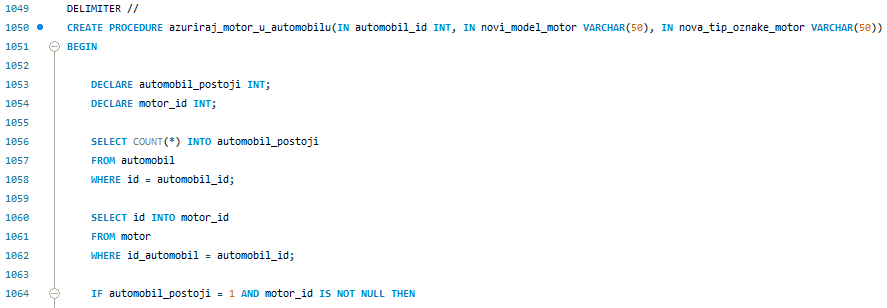
Procedura azuriraj\_bodove\_prvenstva služi za ažuriranje bodova vozača u određenom prvenstvu. Procedura kao ulazni parametar uzima ID prvenstva za koje treba ažurirati bodove.

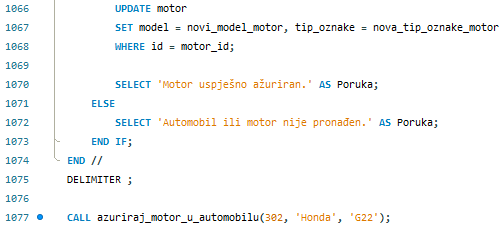
Procedura prvo povećava bodove za 2 bodova za vozača koji je zauzeo prvo mjesto u svim utrkama tog prvenstva. To se postiže putem UPDATE naredbe koja mijenja bodove za svaki rezultat gdje je ID prvenstva jednak predanom ID-u prvenstva i gdje je mjesto vozača 1.

Nakon toga, procedura dohvaća maksimalni broj mjesta ostvaren u tom prvenstvu (najslabiji rezultat). Ovo se radi kako bi se pronašla posljednja pozicija na kojoj su vozači osvojili bodove.

Zatim, procedura smanjuje bodove za 1 bod za vozača koji je zauzeo to maksimalno mjesto (najslabiji rezultat) u svim utrkama tog prvenstva. Korištenjem CALL azuriraj\_bodove\_prvenstva(5000) pokreće se procedura koja ažurira bodove za prvenstvo s ID-om 5000.

## **Procedura**





Procedura azuriraj\_motor\_u\_automobilu služi za ažuriranje podataka o motoru koji je instaliran u određenom automobilu. Procedura uzima tri ulazna parametra: ID automobila, novi model motora i novu tip oznaku motora.

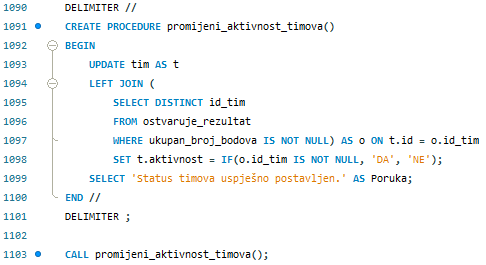
Na početku, procedura provjerava postoji li automobil s unesenim ID-om. To se radi pomoću SELECT naredbe koja broji koliko puta se uneseni ID automobila pojavljuje u tablici automobila.

Zatim, procedura traži ID motora koji je povezan s tim automobilom. Ako su i automobil i motor, procedura ažurira podatke o motoru u tablici motora s novim modelom motora i novom tip oznakom motora.

Na kraju, procedura vraća poruku koja obavještava korisnika o uspješnom ažuriranju motora, ako su svi uvjeti ispunjeni. Ako automobil ili motor nisu pronađeni, vraća se poruka koja to navodi.

Korištenjem CALL azuriraj\_motor\_u\_automobilu(302, 'Honda', 'G22'), procedura će ažurirati motor automobila s ID-om 302 novim modelom 'Honda' i novom tip oznakom 'G22', ukoliko automobil i motor postoje.

## **Procedura**



Procedura promijeni\_aktivnost\_timova koristi se za ažuriranje statusa aktivnosti svih timova. Ova procedura provjerava jesu li timovi osvojili bilo kakve bodove i na temelju toga postavlja njihov status aktivnosti.

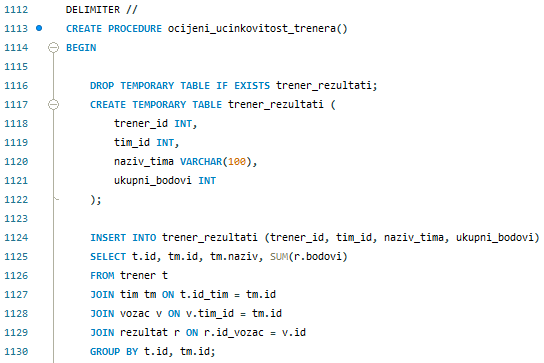
Procedura radi tako što prvo pravi privremenu listu SELECT DISTINCT id\_tim FROM ostvaruje\_rezultat WHERE ukupan\_broj\_bodova IS NOT NULL svih timova koji su osvojili bodove. Zatim se koristi LEFT JOIN za povezivanje te liste s tablicom tim.

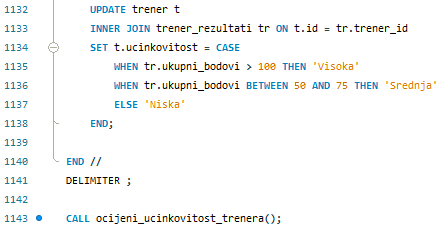
U procesu ažuriranja, procedura postavlja vrijednost kolone aktivnost u tablici tim na 'DA' za sve timove koji su na listi (tj. za one koji su osvojili bodove), a na 'NE' za sve ostale timove.

Nakon izvršenja ovog ažuriranja, procedura vraća poruku koja obavještava korisnika da je status aktivnosti timova uspješno postavljen.

Korištenjem CALL promijeni\_aktivnost\_timova(), procedura će ažurirati status aktivnosti svih timova na temelju njihovih ostvarenih bodova.

## **Procedura**





Procedura ocijeni\_ucinkovitost\_trenera služi za procjenu učinkovitosti trenera temeljem ukupnih bodova koje su osvojili timovi pod njihovim vodstvom. Procedura koristi niz koraka kako bi izračunala i postavila ocjenu učinkovitosti svakog trenera.

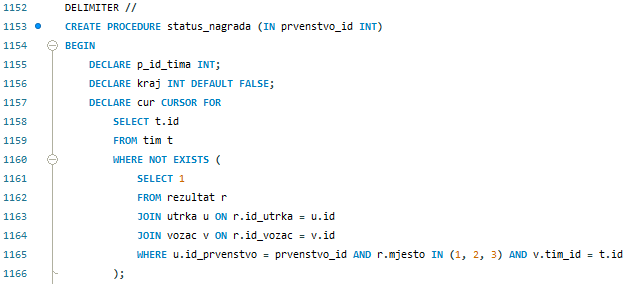
Prvo, procedura briše privremenu tablicu trener\_rezultati ako ona već postoji, a zatim stvara novu privremenu tablicu s istim imenom. Ova tablica sadrži informacije o ID-u trenera, ID-u tima, nazivu tima i ukupnim bodovima koje su timovi osvojili.

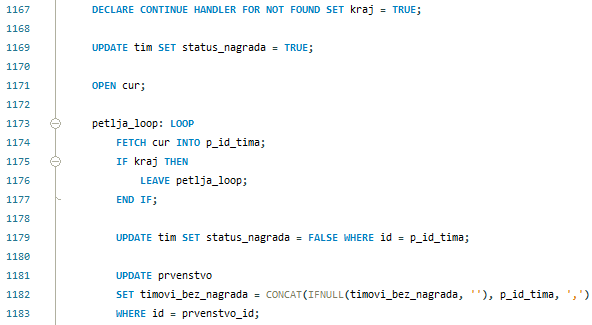
Sljedeći korak je popunjavanje te privremene tablice. Procedura to čini tako što sumira bodove osvojene od strane vozača koji pripadaju timovima podređenim određenim trenerima. Informacije se dohvaćaju iz tablica trener, tim, vozac i rezultat.

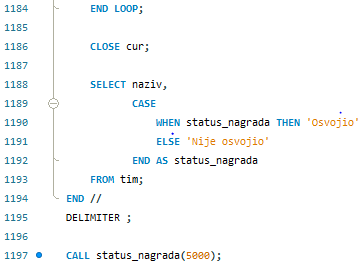
Nakon što su podaci popunjeni, procedura ažurira tablicu trener postavljajući ocjenu učinkovitosti za svakog trenera. Učinkovitost se određuje prema broju bodova koje je tim osvojio: 'Visoka' učinkovitost za više od 100 bodova, 'Srednja' za između 50 i 75 bodova, i 'Niska' za manje bodova.

Korištenjem CALL ocijeni\_ucinkovitost\_trenera(), procedura procjenjuje i ažurira učinkovitost svih trenera temeljem uspjeha njihovih timova u natjecanjima.

## **Procedura**







Procedura status\_nagrada, služi za ažuriranje i prikazivanje status nagrada za sve timove koji sudjeluju u tom prvenstvu.

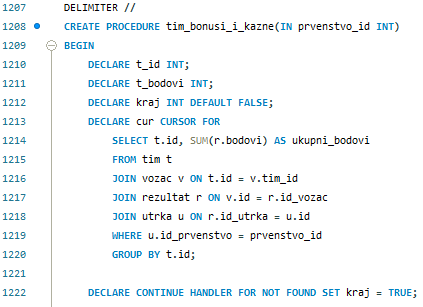
Na početku, procedura postavlja status nagrade na 'DA' za sve timove, pretpostavljajući da su svi timovi mogli osvojiti nagradu. Zatim se koristi kursor kako bi se pregledali svi timovi unutar prvenstva. Procedura posebno traži one timove koji nisu uspjeli osvojiti prva tri mjesta u bilo kojoj od utrka prvenstva. Da bi to učinila, koristi se podupit koji pretražuje rezultate utrka i identificira timove čiji vozači nisu bili među prva tri.

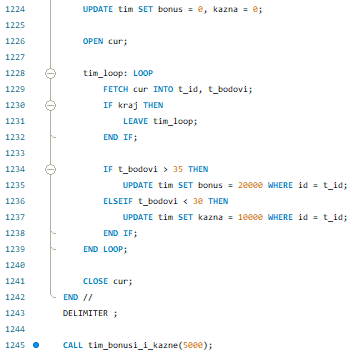
Svaki put kada procedura pronađe tim koji nije osvojio nijedno od prva tri mjesta, njezin status se ažurira u 'NE'. Ujedno, informacija o timu koji nije osvojio nagradu dodaje se u tablicu prvenstva. Ovo omogućava da se na jednom mjestu prate svi timovi koji nisu bili uspješni u osvajanju nagrada.

Nakon što procedura prođe kroz sve timove, ona zatvara kursor. Kao posljednji korak, procedura vraća popis svih timova s ažuriranim statusom nagrade.

Korištenjem CALL status\_nagrada(5000), procedura izvodi ovu provjeru i ažuriranje za prvenstvo s ID-om 5000.

## **Procedura**



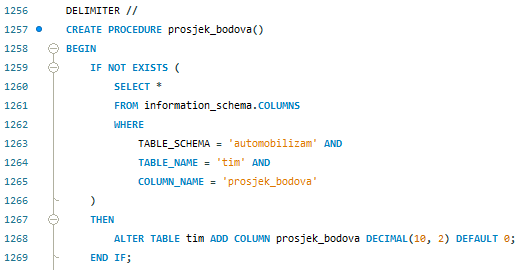


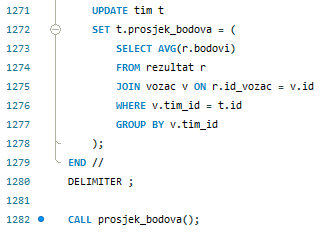
Procedura tim\_bonusi\_i\_kazne služi za dodjelu bonusa ili kazni timovima na osnovu njihovih ukupnih bodova u određenom prvenstvu.

Na početku, procedura postavlja bonuse i kazne svih timova na nulu. Zatim, koristeći kursor, prolazi kroz svaki tim u prvenstvu i izračunava njihove ukupne bodove. Ako tim ima više od 35 bodova, dodjeljuje mu se bonus od 20,000. U suprotnom, ako tim ima manje od 30 bodova, timu se dodjeljuje kazna od 10,000.

Procedura ovako prolazi kroz sve timove, ažurirajući njihov status bonusa i kazni. Na kraju, procedura zatvara kursor i završava svoje izvođenje. Korištenjem CALL tim\_bonusi\_i\_kazne(5000) pokreće se procedura za prvenstvo s ID-om 5000, gdje se bonuse i kazne dodjeljuju timovima na temelju njihovih ukupnih bodova.

## **Procedura**





Procedura prosjek\_bodova služi za izračunavanje i dodavanje prosječnog broja bodova za svaki tim u bazi podataka automobilizma. Procedura prvo provjerava postoji li već stupac 'prosjek\_bodova' u tablici 'tim'. Ako ne postoji, dodaje se novi stupac za pohranu prosječnih bodova.

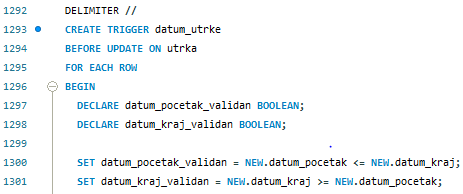
Nakon što se osigura da tablica 'tim' ima potrebni stupac, procedura ažurira svaki tim s njihovim prosječnim brojem bodova. To se postiže izračunavanjem prosjeka bodova koje su vozači svakog tima osvojili u različitim utrkama.

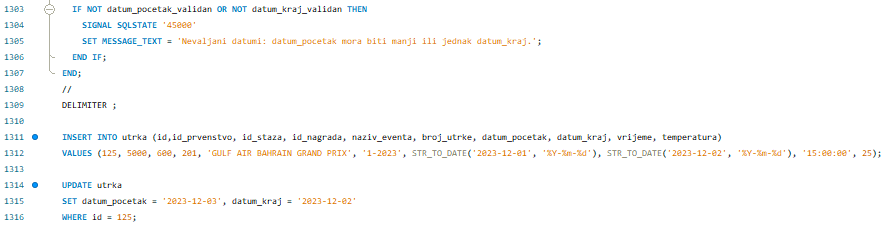
Procedura prolazi kroz sve timove, izračunava prosjek bodova za svaki tim i ažurira stupac 'prosjek\_bodova' u tablici 'tim' s tim prosječnim vrijednostima.

Korištenjem CALL prosjek\_bodova() pokreće se ova procedura, što rezultira ažuriranjem svakog tima u tablici 'tim' s njihovim prosječnim brojem bodova.

# **OKIDAČI**

## **Okidač**



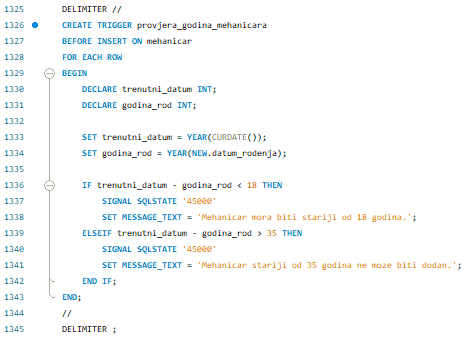


Okidač datum\_utrke je programiran za automatsku provjeru ispravnosti datuma početka i kraja utrke u bazi podataka kada god se ti podaci ažuriraju u tablici utrka. Njegova glavna funkcija je osigurati da datum početka utrke nikada nije kasniji od datuma njenog završetka.

Kada se pokrene ažuriranje u tablici utrka, ovaj okidač prvo provjerava da li je novi datum početka (NEW.datum\_pocetak) manji ili jednak novom datumu kraja (NEW.datum\_kraj). Ako je to slučaj, ažuriranje se nastavlja bez problema.

Međutim, ako je novi datum početka veći od datuma kraja, okidač signalizira grešku i šalje poruku da su datumi neispravni, čime se sprečava nevaljano ažuriranje. Time se osigurava logička dosljednost u evidenciji datuma utrka, što je ključno za točno planiranje i vođenje evidencija u svijetu automobilizma.

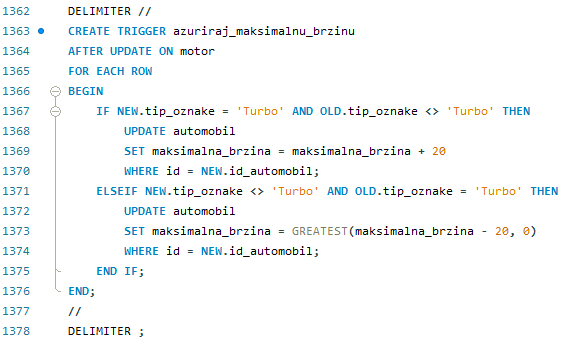
## **Okidač**



Okidač provjera\_godina\_mehanicara osigurava da svaki mehaničar koji se dodaje u bazu podataka ima odgovarajuću dob. Kada se pokuša dodati novi mehaničar, okidač automatski izračunava trenutnu godinu i godinu rođenja mehaničara te provjerava dvije stvari. Prvo, da li je mehaničar stariji od 18 godina, jer ako nije, ne može biti dodan zbog zakonskih ograničenja.

Drugo, okidač provjerava da li je mehaničar stariji od 35 godina, jer ako jest, prema unaprijed određenim pravilima, takva osoba ne može biti dodana u bazu. Ako mehaničar ne zadovoljava ove uvjete, okidač zaustavlja unos i šalje poruku o grešci, čime se osigurava da se u bazu podataka dodaju samo osobe koje ispunjavaju propisane dobne kriterije.

## **Okidač**

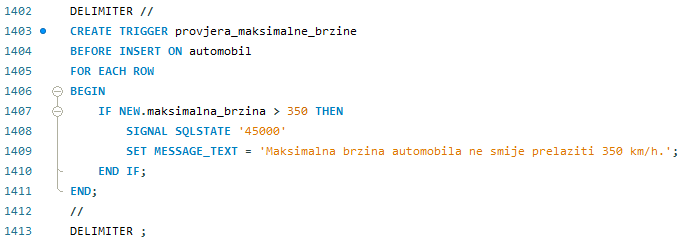


Okidač azuriraj\_maksimalnu\_brzinu se aktivira nakon svake promjene podataka u tablici motor. Njegova glavna svrha je ažurirati maksimalnu brzinu povezanog automobila ovisno o tome ima li motor tip oznake 'Turbo'.

Ako se tip oznake motora promijeni na 'Turbo', a prije toga nije bio 'Turbo', okidač automatski povećava maksimalnu brzinu automobila povezanog s tim motorom za 20 jedinica. To odražava povećanu performansu koju 'Turbo' motor donosi automobilu.

S druge strane, ako se tip oznake motora promijeni s 'Turbo' na nešto drugo, maksimalna brzina automobila se smanjuje za 20 jedinica, ali ne ispod nule. Ovo smanjenje odražava gubitak performansi zbog uklanjanja 'Turbo' motora.

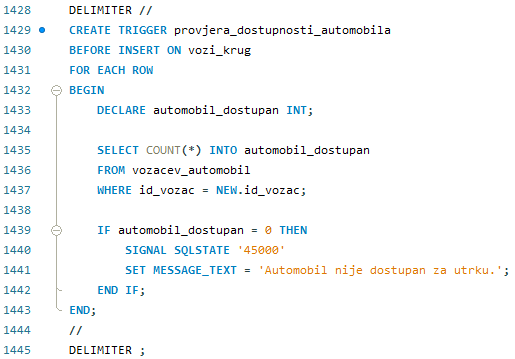
## **Okidač**



Okidač provjera\_maksimalne\_brzine se koristi za provjeru maksimalne brzine automobila prije nego što se unesu u bazu podataka. Ovaj okidač djeluje prije svakog unosa u tablicu automobil.

Kada se pokuša dodati novi automobil, okidač provjerava je li navedena maksimalna brzina automobila veća od 350 km/h. Ako je to slučaj, okidač zaustavlja unos i šalje poruku o grešci, obavještavajući da maksimalna brzina automobila ne smije prelaziti 350 km/h.

## **Okidač**

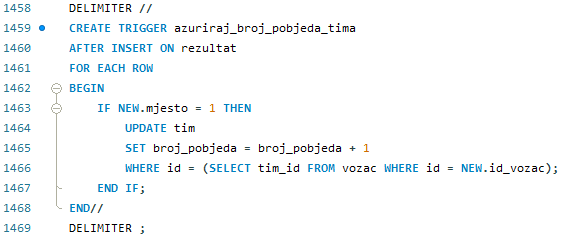


Okidač provjera\_dostupnosti\_automobila provjerava je li automobil dostupan vozaču prije nego što se zabilježi njegov krug u utrci. Ovaj okidač se aktivira prije svakog unosa u tablicu vozi\_krug.

Kada se pokuša unijeti novi zapis o krugu koji je vozač odvozio, okidač prvo provjerava ima li vozač dodijeljen automobil. To radi tako što broji koliko puta se ID vozača pojavljuje u tablici vozacev\_automobil.

Ako vozač nema dodijeljen automobil, tj. ako je broj pronađenih zapisa jednak nuli, okidač prekida unos i šalje poruku o grešci koja govori da automobil nije dostupan za utrku.

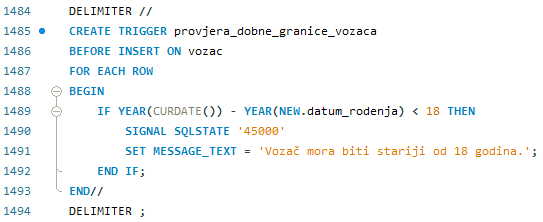
## **Okidač**



Okidač azuriraj\_broj\_pobjeda\_tima automatski ažurira broj pobjeda tima svaki put kada vozač tog tima osvoji prvo mjesto u utrci. Ovaj okidač se aktivira nakon svakog unosa u tablicu rezultat.

Kada se unese novi rezultat, okidač provjerava je li mjesto koje je vozač zauzeo prvo mjesto. Ako jest, okidač pronalazi tim kojem pripada taj vozač i povećava njihov ukupan broj pobjeda za jedan. To se postiže ažuriranjem tablice tim, gdje se broj pobjeda povećava za svaki put kada vozač tog tima osvoji prvo mjesto u utrci.

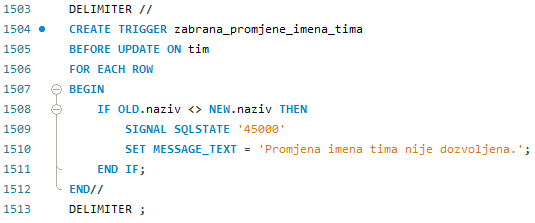
## **Okidač**



Okidač provjera\_dobne\_granice\_vozaca je postavljen za provjeru da li su vozači koji se dodaju u bazu podataka stariji od 18 godina. Ovaj okidač se aktivira prije svakog unosa u tablicu vozac.

Kada se pokuša dodati novi vozač, okidač izračunava starost vozača na osnovu njegovog datuma rođenja. Ako je vozač mlađi od 18 godina, okidač prekida proces unosa i šalje poruku o grešci, obavještavajući da vozač mora biti stariji od 18 godina.

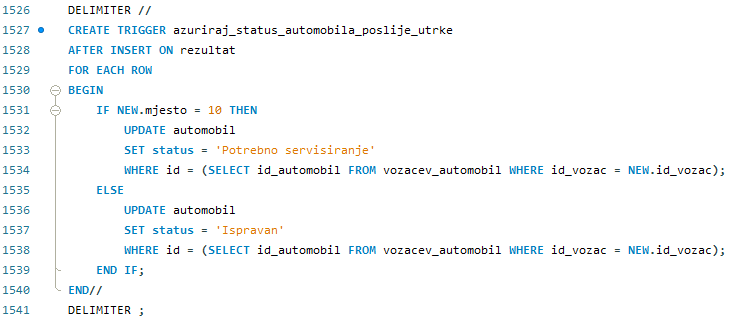
## **Okidač**



Okidač zabrana\_promjene\_imena\_tima postavljen je da spriječi promjenu imena timova u bazi podataka. Ovaj okidač se aktivira prije svake promjene u tablici tim.

Kad god se pokuša ažurirati zapis u tablici tim, okidač provjerava je li došlo do promjene u nazivu tima. Ako se pokuša promijeniti naziv tima (tj. ako su stari i novi naziv različiti), okidač prekida proces ažuriranja i javlja grešku, poručujući da promjena imena tima nije dozvoljena.

## **Okidač**

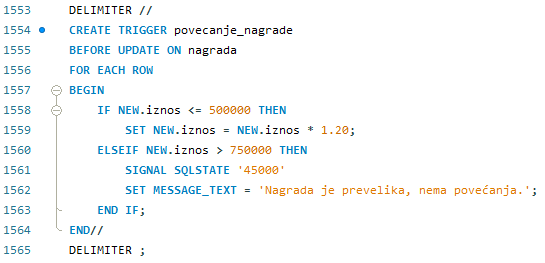


Okidač azuriraj\_status\_automobila\_poslije\_utrke automatski ažurira status automobila nakon svake utrke, ovisno o plasmanu vozača. Ovaj okidač se aktivira nakon unosa rezultata utrke u tablicu rezultat.

Ako vozač zauzme deseto mjesto u utrci, okidač postavlja status povezanog automobila na 'Potrebno servisiranje'. Ovo bi moglo odražavati politiku tima ili pravila natjecanja koja nalažu da automobili koji se plasiraju na određenim pozicijama prolaze kroz detaljan tehnički pregled ili servis nakon utrke.

U svim ostalim slučajevima, status automobila se postavlja na 'Ispravan', sugerirajući da automobil nije doživio tehničke probleme tijekom utrke i da je spreman za daljnju upotrebu.

## **Okidač**



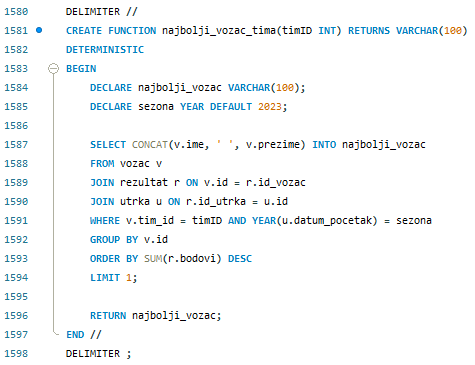
Okidač povecanje\_nagrade je postavljen da automatski poveća iznos nagrade prije svakog ažuriranja u tablici nagrada, ali i da osigura da iznos nagrade ne postane prevelik.

Ako se pokuša ažurirati iznos nagrade i ako je novi predloženi iznos manji ili jednak 500,000, okidač automatski povećava taj iznos za 20%. To znači da će se, primjerice, iznos od 500,000 povećati na 600,000.

Međutim, ako je novi iznos nagrade veći od 750,000, okidač zaustavlja ažuriranje i šalje poruku o grešci, obavještavajući da nagrada neće biti povećana jer je već prevelika.

# **FUNKCIJE**

## **Funkcija**



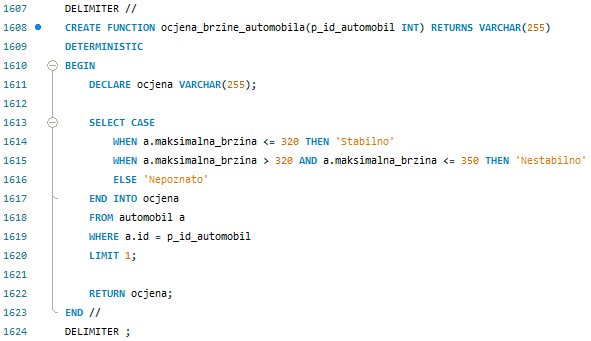
Funkcija najbolji\_vozac\_tima je kreirana kako bi odredila najboljeg vozača unutar određenog tima za specifičnu sezonu. Ova funkcija vraća ime i prezime najboljeg vozača temeljem ukupno osvojenih bodova tijekom sezone.

Funkcija prima ID tima kao ulazni parametar i koristi fiksiranu godinu sezone za određivanje relevantnih utrka i rezultata.

Unutar funkcije, izvršava se SQL upit koji se spaja s tablicama vozac, rezultat i utrka. Upit filtrira vozače koji pripadaju zadanom timu i utrke koje su se odvijale tijekom definirane sezone. Grupiranje po ID-u vozača omogućava sumiranje bodova svakog vozača.

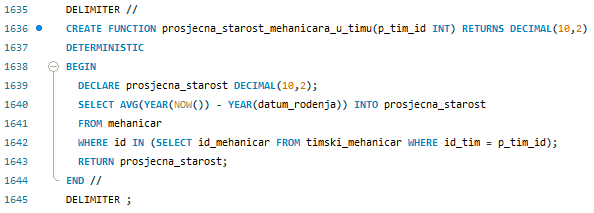
Na kraju, upit naređuje vozače po ukupnom broju osvojenih bodova u silaznom redoslijedu i ograničava rezultat na samo jednog vozača - onog s najvećim brojem bodova, koji se smatra najboljim vozačem tima. Funkcija zatim vraća ime i prezime tog vozača.

## **Funkcija**



Funkcija ocjena\_brzine\_automobila daje ocjenu brzine za automobil na temelju njegove maksimalne brzine. Kad se unese ID automobila, funkcija provjerava njegovu maksimalnu brzinu. Ako je ta brzina 320 km/h ili manje, automobil se smatra stabilnim i funkcija vraća ocjenu 'Stabilno'. Ako je brzina između 320 i 350 km/h, smatra se da automobil može biti nestabilan pa funkcija vraća ocjenu 'Nestabilno'. Za sve brzine iznad 350 km/h ili ako podaci nisu jasni, funkcija vraća 'Nepoznato'.

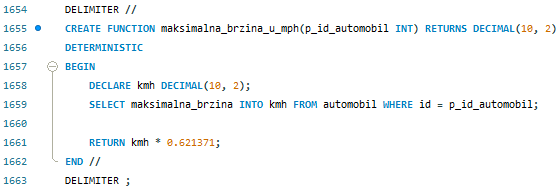
## **Funkcija**



Funkcija prosjecna\_starost\_mehanicara\_u\_timu računa prosječnu starost mehaničara koji rade u određenom timu. Kad se unese ID tima, funkcija izračunava prosjek godina svih mehaničara povezanih s tim timom.

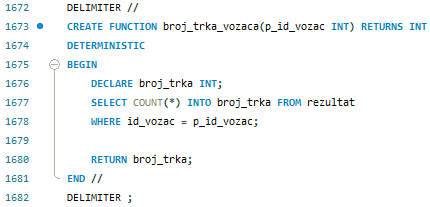
Funkcija prvo traži sve mehaničare koji su povezani s timom koristeći ID tima. Zatim izračunava prosječnu starost tih mehaničara tako što oduzima godinu rođenja svakog mehaničara od trenutne godine. Na kraju, funkcija vraća izračunatu prosječnu starost kao decimalni broj s dvije decimalne točke.

## **Funkcija**



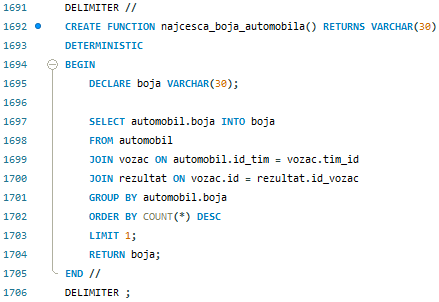
Funkcija maksimalna\_brzina\_u\_mph pretvara maksimalnu brzinu automobila iz kilometara na sat (km/h) u milje na sat (mph). Kada se unese ID automobila, funkcija prvo dohvaća njegovu maksimalnu brzinu izraženu u km/h iz tablice automobil. Zatim tu brzinu pretvara u mph množeći ju s faktorom pretvorbe 0.621371. Rezultat je maksimalna brzina tog automobila izražena u mph, koja se vraća kao decimalni broj s dvije decimalne točke.

## **Funkcija**



Funkcija broj\_trka\_vozaca služi za izračunavanje ukupnog broja utrka u kojima je sudjelovao određeni vozač. Kada se unese ID vozača, funkcija pretražuje tablicu rezultat i broji koliko puta se pojavljuje ID tog vozača, što odgovara broju utrka u kojima je sudjelovao. Nakon što izračuna ukupan broj, funkcija vraća taj broj kao cijeli broj.

## **Funkcija**



Funkcija najcesca\_boja\_automobila služi za utvrđivanje koje je boje automobil najčešće korišten u utrkama. Funkcija pretražuje bazu podataka, spaja tablice automobil, vozac i rezultat, te broji koliko puta se automobili različitih boja pojavljuju u utrkama.

Na osnovu toga, funkcija određuje koja je boja najčešće korištena, grupirajući podatke po boji automobila i brojeći broj pojavljivanja svake boje. Rezultati se zatim naređuju od najčešće do najrjeđe korištene boje, a funkcija vraća boju koja se pojavljuje najviše puta.

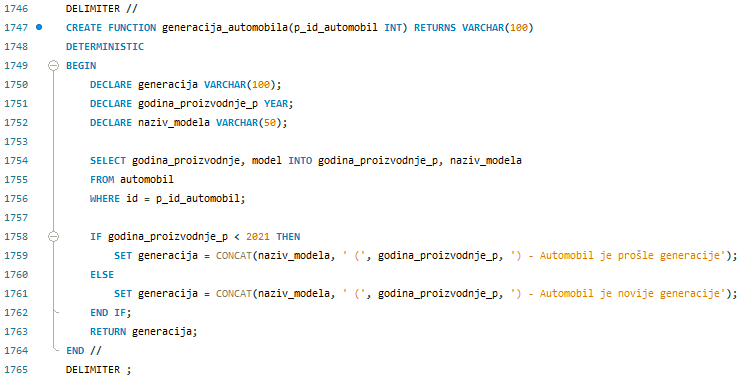
## **Funkcija**



Funkcija kontinent\_sponzora određuje kontinent na kojem se nalazi sponzor na temelju njegove države. Funkcija prima ID sponzora kao ulazni parametar, a zatim traži državu tog sponzora u bazi podataka.

Nakon što se pronađe država sponzora, funkcija uspoređuje tu državu s unaprijed definiranom listom država po kontinentima. Ako država pripada Sjedinjenim Američkim Državama, Kanadi ili Meksiku, funkcija vraća 'Amerika' kao kontinent sponzora. Ako država pripada skupini europskih zemalja poput Njemačke, Francuske, Italije, Španjolske, Velike Britanije, Švedske ili Švicarske, funkcija vraća 'Europa'. Za sve ostale države, funkcija vraća 'Drugi kontinent'.

## **Funkcija**



Funkcija generacija\_automobila služi za određivanje generacije automobila na temelju njegove godine proizvodnje. Kad se unese ID automobila, funkcija dohvaća godinu proizvodnje i model tog automobila iz baze podataka.

Ako je godina proizvodnje automobila manja od 2021, funkcija zaključuje da je automobil prošle generacije i vraća naziv modela s godinom proizvodnje, dodajući opis da se radi o automobilu prošle generacije. U suprotnom, ako je godina proizvodnje 2021 ili kasnije, funkcija zaključuje da je automobil novije generacije te vraća naziv modela s godinom proizvodnje, uz napomenu da je automobil novije generacije.

## **Funkcija**

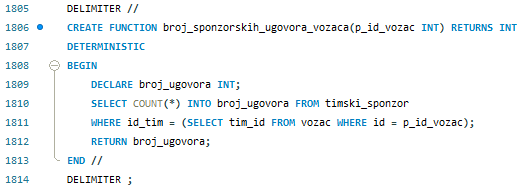


Funkcija uvjeti\_za\_voznju procjenjuje uvjete za vožnju na automobilističkoj stazi tijekom određene utrke. Ova funkcija uzima ID utrke kao ulazni parametar, a zatim dohvaća podatke o temperaturi i vremenu utrke, kao i naziv staze.

Na temelju dobivenih informacija, funkcija određuje jesu li uvjeti za vožnju dobri ili loši. Ako je temperatura između 15 i 25 stupnjeva celzijusa i utrka se održava između 7 ujutro i 19 sati, funkcija zaključuje da su uvjeti dobri. U tom slučaju vraća poruku koja opisuje dobre uvjete za vožnju, navodeći naziv staze, temperaturu i vrijeme.

Ukoliko uvjeti ne zadovoljavaju ove kriterije, funkcija zaključuje da su uvjeti loši te vraća poruku koja opisuje loše uvjete, također navodeći naziv staze, temperaturu i vrijeme.

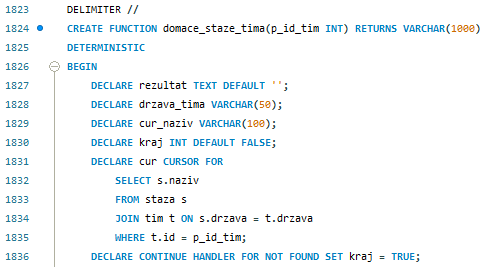
## **Funkcija**

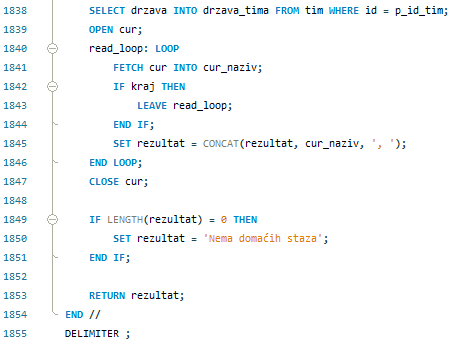


Funkcija broj\_sponzorskih\_ugovora\_vozaca služi za izračunavanje broja sponzorskih ugovora koje ima tim u kojem vozi određeni vozač. Kada se unese ID vozača, funkcija traži tim kojem taj vozač pripada i zatim broji koliko sponzorskih ugovora ima taj tim.

Dakle, funkcija prvo identificira tim vozača pretražujući tablicu vozac kako bi pronašla ID tima povezanog s tim vozačem. Nakon toga, funkcija pretražuje tablicu timski\_sponzor da vidi koliko ugovora ima taj tim. Na kraju, funkcija vraća ukupan broj sponzorskih ugovora kao cijeli broj.

## **Funkcija**



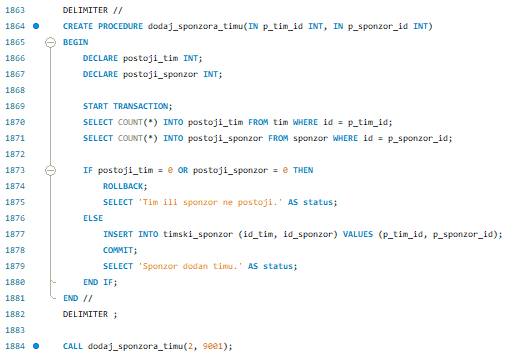


Funkcija domace\_staze\_tima pruža popis staza koje se nalaze u istoj državi kao i određeni automobilistički tim. Kada se unese ID tima, funkcija traži sve staze koje se nalaze u istoj državi u kojoj je registriran taj tim.

Funkcija najprije dohvaća državu u kojoj se tim nalazi, a zatim koristi kursor za prolazak kroz sve staze u toj državi. Za svaku pronađenu stazu, njezin naziv se dodaje u tekstualni niz rezultat. Ako ne pronađe nijednu stazu, funkcija postavlja rezultat na poruku 'Nema domaćih staza'. Nakon što se prođe kroz sve staze, funkcija vraća tekstualni niz koji sadrži imena svih domaćih staza za taj tim.

# **TRANSAKCIJA**

## **Transakcija**

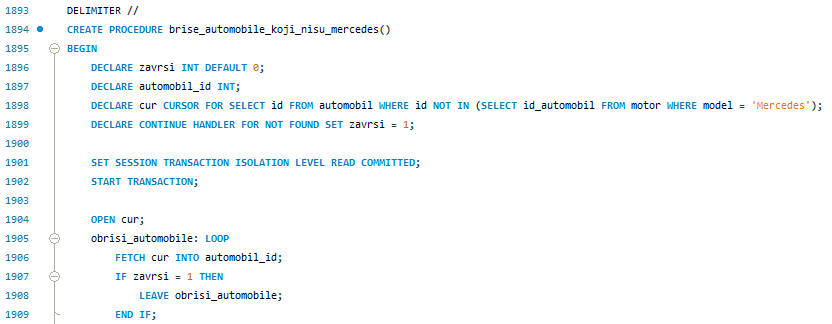


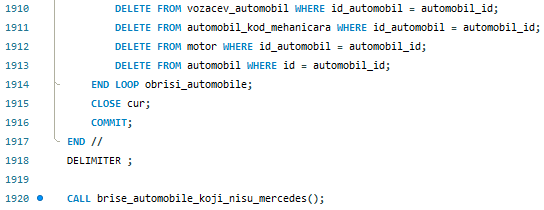
Procedura dodaj\_sponzora\_timu koristi se za dodavanje novog sponzora automobilističkom timu. Prvo provjerava postoje li u bazi podataka uneseni tim i sponzor prema njihovim ID-ovima. Ako jedan od njih ne postoji, procedura se zaustavlja i vraća poruku da tim ili sponzor ne postoji.

U slučaju da i tim i sponzor postoje, procedura dodaje sponzora timu zapisom u tablicu timski\_sponzor. Nakon uspješnog dodavanja, procedura potvrđuje transakciju i vraća poruku da je sponzor uspješno dodan timu.

Pozivom CALL dodaj\_sponzora\_timu(2, 9001) pokreće se procedura za dodavanje sponzora s ID-om 9001 timu s ID-om 2, ali samo ako oba entiteta postoje u bazi podataka.

## **Transakcija**





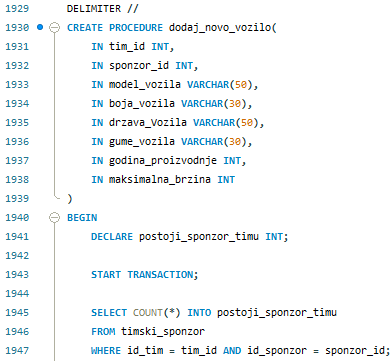
Procedura brise\_automobile\_koji\_nisu\_mercedes služi za brisanje svih automobila iz baze podataka koji nisu povezani s Mercedes motorima. Procedura prolazi kroz sve automobile i briše one koji ne sadrže motor marke Mercedes.

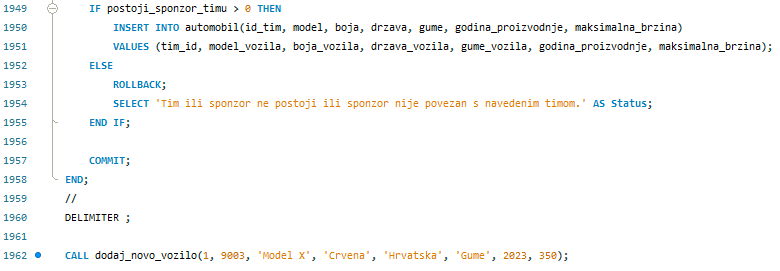
Prvo, procedura postavlja razinu izolacije transakcija na 'READ COMMITTED' kako bi se izbjegli problemi s istovremenim pristupima podacima. Zatim započinje transakciju i otvara kursor koji prolazi kroz sve automobile koji nisu povezani s Mercedes motorima.

Za svaki pronađeni automobil koji ne ispunjava kriterij (da ima Mercedes motor), procedura izvršava niz operacija brisanja. Briše sve veze tog automobila s tablicom vozacev\_automobil, automobil\_kod\_mehanicara i motor, te na kraju briše sam automobil iz tablice automobil.

Nakon što se obrade svi takvi automobili, kursor se zatvara, a transakcija se potvrđuje. Pozivom CALL brise\_automobile\_koji\_nisu\_mercedes() pokrenut će se procedura koja je definirana kao brise\_automobile\_koji\_nisu\_mercedes. Ova procedura će obaviti zadatak brisanja svih automobila iz baze podataka koji nisu opremljeni Mercedes motorima.

## **Transakcija**





Procedura dodaj\_novo\_vozilo koristi se za dodavanje novog automobila u bazu podataka, ali samo ako postoji veza između unesenog tima i sponzora

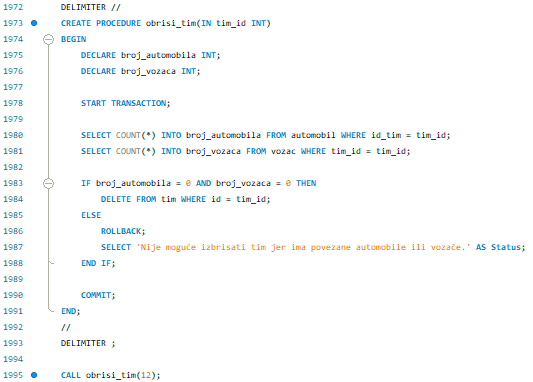
Procedura prima različite parametre poput ID-a tima i sponzora, modela vozila, boje, države porijekla vozila, tipa guma, godine proizvodnje i maksimalne brzine. Započinje transakcijom i provjerava postoji li veza između unesenog tima i sponzora. Ako veza postoji, dodaje novi automobil s unesenim specifikacijama u tablicu automobil.

Međutim, ako veza između tima i sponzora ne postoji, procedura prekida transakciju i vraća poruku da tim ili sponzor ne postoji ili da sponzor nije povezan s timom. Pozivom CALL dodaj\_novo\_vozilo(1, 9003, 'Model X', 'Crvena', 'Hrvatska', 'Gume', 2023, 350) procedura će pokušati dodati automobil 'Model X' u boji 'Crvena', s porijeklom iz Hrvatske, gumama 'Gume', godinom proizvodnje 2023 i maksimalnom brzinom od 350, za tim s ID-om 1 i sponzorom s ID-om 9003.

Procedura će uspjeti samo ako postoji veza između tog tima i sponzora. Ako takva veza postoji, automobil će biti dodan u bazu podataka s navedenim karakteristikama. To znači da će se u tablicu automobil dodati novi zapis koji sadrži sve specificirane detalje o automobilu, povezane s odabranim timom i sponzorom.

Ako, s druge strane, veza između tima s ID-om 1 i sponzora s ID-om 9003 ne postoji u bazi podataka, procedura neće dodati automobil i umjesto toga će vratiti poruku koja upozorava da ili tim ili sponzor ne postoji, ili da sponzor nije povezan s timom.

## **Transakcija**

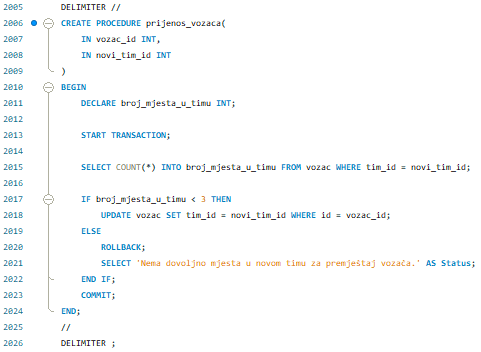


Procedura obrisi\_tim koristi se za brisanje tima iz baze podataka, ali s važnim uvjetom. Procedura se poziva s ID-em tima koji želite izbrisati. Prije nego što izvrši brisanje, procedura provodi provjeru kako bi osigurala da u tom timu nema povezanih automobila ili vozača. To znači da procedura prvo pretražuje bazu podataka da vidi ima li bilo kakvih automobila ili vozača koji su registrirani pod tim timom.

Ako se u timu nalaze automobili ili vozači, procedura neće dozvoliti brisanje tima. To je mjera opreza kako bi se spriječilo uklanjanje tima koji je aktivno uključen u natjecanja. U tom slučaju, procedura vraća poruku koja vas obavještava da brisanje nije moguće zbog postojećih povezanih automobila ili vozača. S druge strane, ako tim nema povezanih automobila ili vozača, procedura će sigurno izbrisati tim iz baze podataka.

Pozivom CALL obrisi\_tim(12) procedura će pregledati postoje li automobili ili vozači povezani s timom broj 12. Ako ne postoje, tim se briše, a ako postoje, procedura će vas obavijestiti da brisanje nije moguće.

## **Transakcija**

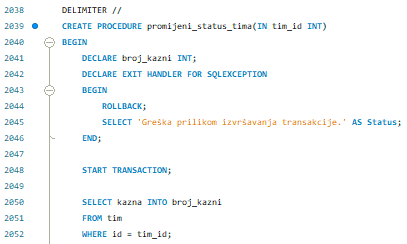


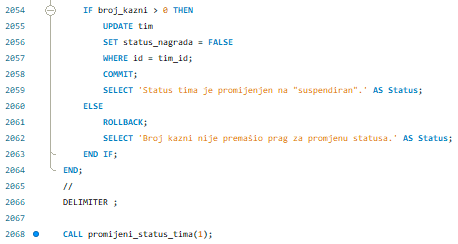
Procedura prijenos\_vozaca služi za premještanje vozača iz jednog tima u drugi. Kada se pozove s ID-em vozača i ID-em novog tima, procedura prvo provjerava ima li mjesta u novom timu za dodatnog vozača.

Procedura započinje transakcijom i broji koliko vozača već ima tim u koji se vozač namjerava premjestiti. Ako novi tim već ima tri ili više vozača, procedura ne dozvoljava premještaj jer nema dovoljno mjesta. U tom slučaju, transakcija se prekida i vraća se poruka da u novom timu nema dovoljno mjesta za novog vozača.

Međutim, ako u novom timu ima manje od tri vozača, procedura ažurira podatke vozača tako da se promijeni njihov tim na novi tim. Zatim potvrđuje transakciju. Pozivom CALL prijenos\_vozaca(103, 1) procedura će pokušati premjestiti vozača s ID-em 103 u tim s ID-em 1, ali samo ako u tom timu ima manje od tri vozača.

## **Transakcija**





Procedura promijeni\_status\_tima koristi se za promjenu statusa automobilističkog tima u bazi podataka, zasnovano na broju kazni koje taj tim ima. Kada se pozove s ID-em tima, procedura provjerava broj kazni koje je tim akumulirao.

Ako tim ima jednu ili više kazni, procedura mijenja status tima u "suspendiran" (postavlja status\_nagrada na FALSE). Ovo ukazuje na to da tim zbog akumuliranih kazni trenutno nije u dobrom statusu ili nije ispunio određene uvjete.

Međutim, ako broj kazni nije premašio određeni prag (tj. ako je broj kazni nula), procedura ne mijenja status tima i završava s porukom da broj kazni nije dovoljan za promjenu statusa.

Procedura također sadrži handler za SQL iznimke. Ako dođe do greške prilikom izvođenja transakcije, procedura će zaustaviti transakciju i vratiti poruku o grešci. Pozivom CALL promijeni\_status\_tima(1) procedura provjerava broj kazni za tim s ID-em 1 i na temelju toga odlučuje hoće li promijeniti njegov status.

# **INICIJALIZACIJA I KONFIGURACIJA**

**Flask(\_\_name\_\_):** Kreira instancu Flask aplikacije.

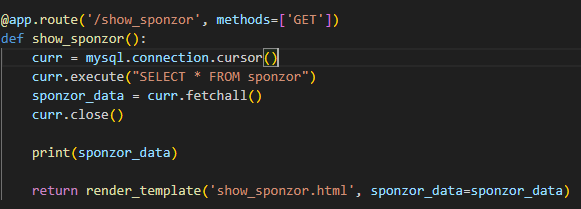
**app.secret\_key:** Postavlja tajni ključ za sesije i kolačiće, što je bitno za sigurnost aplikacije.

**app.config[..]:** Postavlja parametre za povezivanje s MySQL bazom podataka, uključujući host, korisničko ime, lozinku i naziv baze.

**MySQL(app):** Inicijalizira MySQL objekt za interakciju s bazom podataka.

# **RUTE I FUNKCIONALNOSTI**

**Glavna Ruta (/):** Vraća početnu stranicu aplikacije **(home.html).**



Primjer izgleda jedne rute, koja prikazuje tablicu sponzora, putem HTML predloška.

## **Rute za Sponzore**

**/show\_sponzor:** Prikazuje sve sponzore iz baze podataka.

**/insert\_sponzor\_form:** Prikazuje formu za dodavanje novog sponzora.

**/insert\_sponzor:** Procesira unos i dodaje novog sponzora u bazu.

**/delete\_sponzor/<int:id>:** Briše sponzora prema njegovom ID-u.

## **Rute za Timove**

**/show\_tim:** Prikazuje sve timove i povezane informacije.

**/insert\_tim\_form:** Prikazuje formu za dodavanje novog tima.

**/insert\_tim:** Dodaje novi tim u bazu.

**/delete\_tim/<int:id>:** Briše tim na temelju ID-a.

## **Rute za Vozače**

**/show\_vozaci:** Prikazuje sve vozače i njihove timove.

**/insert\_vozac\_form:** Forma za dodavanje novog vozača.

**/insert\_vozac:** Dodaje vozača u bazu.

**/delete\_vozac/<int:id>:** Briše vozača na temelju ID-a.

## **Rute za Automobile**

**/show\_automobil:** Prikazuje sve automobile.

**/insert\_automobil\_form:** Forma za dodavanje novog automobila.

**/insert\_automobil:** Dodaje automobil u bazu.

**/delete\_automobil/<int:id>:** Briše automobil na temelju ID-a.

## **Rute za Utrke**

**/show\_utrka:** Prikazuje informacije o svim utrkama, uključujući prvenstva, nagrade i staze.

## **Rute za Staze**

**/show\_staza:** Prikazuje sve staze.

**/insert\_staza\_form:** Forma za dodavanje nove staze.

**/insert\_staza:** Dodaje novu stazu u bazu.

**/delete\_staza/<int:id>:** Briše stazu na temelju ID-a.

## **Rute za Nagrade**

**/show\_nagrada:** Prikazuje sve nagrade.

**/insert\_nagrada\_form:** Forma za dodavanje nove nagrade.

**/insert\_nagrada:** Dodaje nagradu u bazu.

**/delete\_nagrada/<int:id>:** Briše nagradu na temelju ID-a.

## **Rute za Prvenstva**

**/show\_prvenstvo:** Prikazuje sva prvenstva.

**/insert\_prvenstvo\_form:** Forma za dodavanje novog prvenstva.

**/insert\_prvenstvo:** Dodaje prvenstvo u bazu.

**/delete\_prvenstvo/<int:id>:** Briše prvenstvo na temelju ID-a.

## **Rute za Rezultate**

**/show\_rezultat:** Prikazuje rezultate utrka, bodove i vremena vožnje po krugovima.

**/izracunaj\_rezultate/<int:id\_utrka>:** Izračunava i unosi rezultate za određenu utrku.

**/izbrisi\_rezultate:** Briše sve rezultate iz baze podataka.

## **Dodatne Rute**

**/pocetne\_pozicije/<int:id\_utrka>:** Prikazuje početne pozicije za odabranu utrku.

**/obracun\_bodova/<int:id\_prvenstvo>:** Obračunava bodove za određeno prvenstvo i prikazuje najbolje rezultate.

# **INTERAKCIJA S BAZOM PODATAKA**

**cursor:** Se koristi za izvođenje SQL naredbi i upita.

**fetchall():** Dohvaća sve rezultate SQL upita.

**commit():** Sprema promjene u bazu podataka.

**close():** Zatvara veze s bazom nakon upotrebe.

# **PORUKE I REDIREKCIJA**

**flash():** Služi za prikazivanje poruka korisnicima (npr. uspješno dodavanje ili brisanje podataka).

**redirect(url\_for('route\_name')):** Preusmjerava korisnika na drugu rutu nakon izvršene akcije.

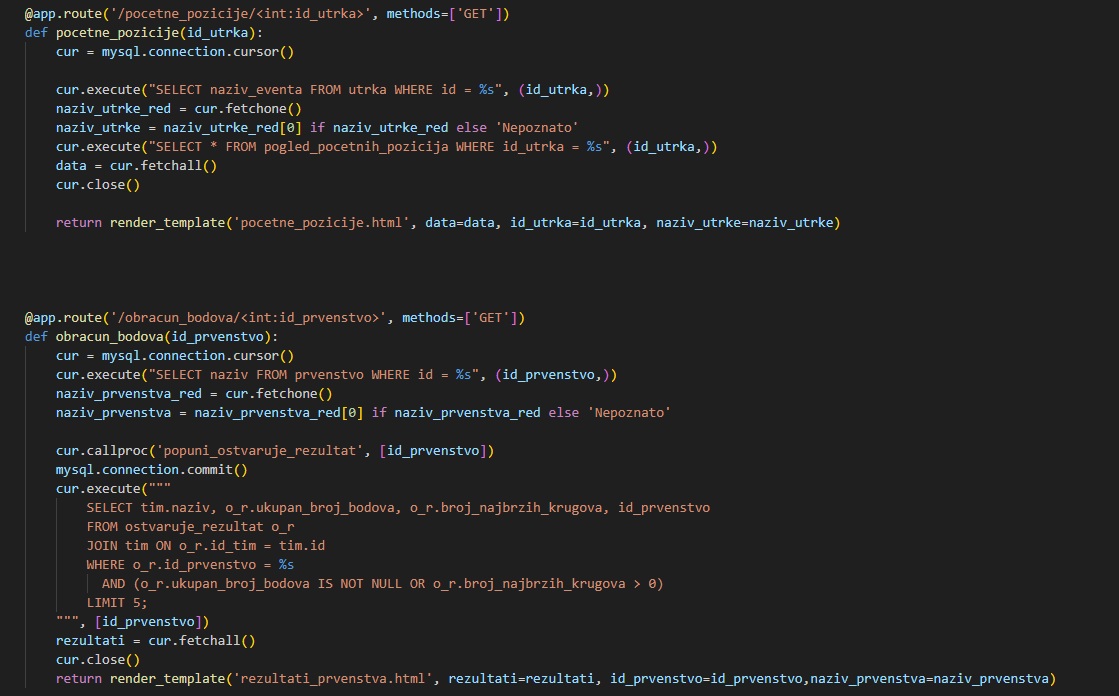
# **RENDERIRANJE HTML PREDLOŽAKA**

**render\_template:** Korišteno za prikaz HTML stranica, s mogućnošću slanja podataka iz Pythona u predložak.

# **POKRETANJE APLIKACIJE**

**app.run(debug=True, port=5000):** Pokreće Flask aplikaciju na portu 5000, s uključenim debug modom za lakše praćenje i ispravljanje grešaka.

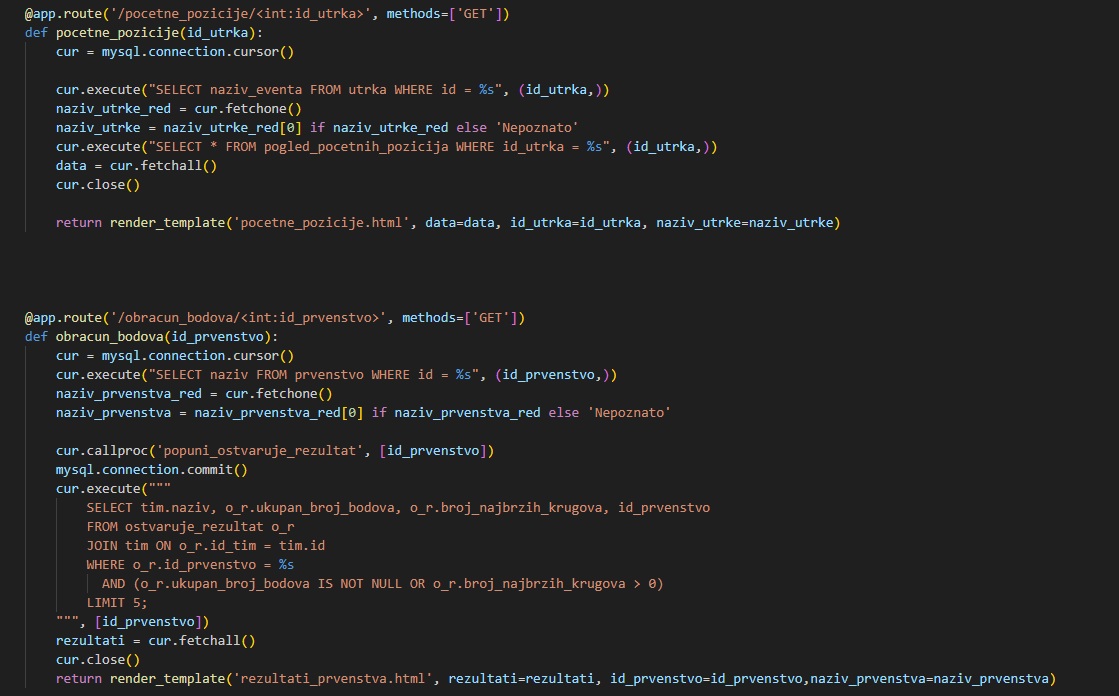
# **VIEW POCETNE\_POZICIJE**



View **pocetne\_pozicije(id\_utrka)** u Flask aplikaciji služi za prikaz početnih pozicija vozača u određenoj utrci.

Kada se zatraži URL **/pocetne\_pozicije/<int:id\_utrka>**, prvo se uspostavlja veza s bazom podataka pomoću **cursor** objekta. Zatim, view izvršava SQL upit da bi dohvatila naziv događaja iz tablice utrka prema ID-u koji je specificiran u URL-u. Ako naziv ne postoji, view će umjesto toga koristiti naziv 'Nepoznato'. Nakon toga, view dohvaća sve podatke o početnim pozicijama iz odgovarajućeg pogleda **(VIEW)** u bazi podataka koji se odnosi na tu utrku. Na kraju, veza s bazom se zatvara i dobiveni podaci se prosljeđuju HTML predlošku **pocetne\_pozicije.html** za prikaz u web pregledniku.

# **PROCEDURA OBRACUN\_BODOVA**



Procedura **obracun\_bodova(id\_prvenstvo)** radi na sličan način,ali za prvenstva.

Ona se aktivira kada korisnik zatraži URL **/obracun\_bodova/<int:id\_prvenstvo>.** Procedura prvo dohvaća naziv prvenstva iz baze podataka. Ako ne postoji, koristi 'Nepoznato' kao naziv.

Zatim poziva pohranjenu proceduru **popuni\_ostvaruje\_rezultat** koja ažurira rezultate prvenstva unutar baze. Nakon što se promjene potvrde u bazi **(commit)**, procedura izvršava dodatni SQL upit koji izvlači top pet timova prema njihovim ukupnim bodovima i broju najbržih krugova za to prvenstvo. Kad se podaci dohvate, **cursor** se zatvara, a rezultati se šalju u predložak **rezultati\_prvenstva.html** koji se renderira za prikaz na klijentskoj strani.