

Sveučilište Jurja Dobrile u Puli

Ulica Alda Negrija 6, 52100 Pula, Hrvatska

Fakultet Informatike u Puli

Tim 8:Sustav za upravljanje rasporedom radnog vremena zaposlenika

Datum predaje\*\*\*

Juraj Crljenko

Stjepan Srdarević

Luka Hušak

Luka Catela

Nikol Buzećan

Mateo Šegon

Smjer : Informatika

Kolegij : Baze podataka II

Mentor : doc. Dr. sc. Goran Oreški

Asistent : mag. inf. Romeo Šajina

Sadržaj

[1. Uvod 2](#_Toc1319891336)

[2. Opis poslovnog procesa 2](#_Toc390790689)

[3. ER dijagram 3](#_Toc690407488)

[4. Opis ER dijagrama 4](#_Toc172359043)

[5. Shema tablica 6](#_Toc994528672)

[6. EER dijagram 7](#_Toc1662580328)

[7. Tablice 8](#_Toc78736328)

[7.1. Tablica zaposlenik 8](#_Toc1484345038)

[7.2 Tablica Odjela 10](#_Toc1814205616)

[7.3 Tablica Smjene 10](#_Toc620938271)

[7.4 Tablica vrsta smjene 11](#_Toc1770834167)

[7.4 Tablica Bolovanje 12](#_Toc395263864)

[7.5 Tablica Raspored rada 13](#_Toc1395818098)

[7.6 Tablica Evidencija rada 14](#_Toc310279169)

[7.7 Tablica place 15](#_Toc127797113)

[7.8 Tablica Godisnji odmor 16](#_Toc475155207)

[7.9 Tablica Zahtjev prekovremeni 17](#_Toc2042167906)

[7.10 Tablica Preferencije smjena 18](#_Toc1590947358)

[7.11 Tablica sluzbena putovanja 19](#_Toc6832727)

[7.12 Tablica Dopust 20](#_Toc1848158746)

[7.13 Tablica projekti 21](#_Toc1627281015)

[7.14 Tablica Zadaci 22](#_Toc548995340)

[7.15 Tablica Napomene 24](#_Toc778229990)

[8. Pogledi 25](#_Toc1617927095)

[8.1 Pogled zaposlenici i odjeli 25](#_Toc501925732)

[8.2 Pogled aktivnih zaposlenika i osnovnih informacija o njima 25](#_Toc208699648)

[8.3 Pogled mjesečnih plaća po zaposleniku 26](#_Toc1805627875)

[8.4 Pogled aktivnih projekata i zadataka 26](#_Toc1704836588)

[8.5 Pogled troškova službenih putovanja 27](#_Toc849385023)

[9. Upiti 27](#_Toc1313509553)

[9.1 Zaposlenici s najviše odrađenih sati protekli mjesec 27](#_Toc682908549)

[9.2 Upit svih zaposlenika s njihovim odjelima i ukupnom plaćom u određenom mjesecu 28](#_Toc389274080)

[9.3 Upit koji prikazuje koji su zaposlenici radili više od 15 prekovremenih sati određeni mjesec 28](#_Toc1551550389)

[9.4 Upit s projektima koji ističu u sljedećih 7 dana 29](#_Toc1438933601)

[9.5 Upit liste prekovremenih zahtjeva koji su odbijeni s razlozima 29](#_Toc1688756096)

[9.6 Upit zaposlenika koji su trenutno na bolovanju 29](#_Toc1116238587)

[9.7 Upit ukupan broj sati rada po zaposleniku za zadani mjesec 30](#_Toc415920253)

[9.8 Upit zaposlenika s najviše pozitivnih napomena 30](#_Toc1927714509)

[9.9 Upit ukupna plaća i broj prekovremenih sati po odjelu za određeni mjesec 31](#_Toc1894998076)

[10. Procedure 31](#_Toc1426142340)

[10.1 Procedura dodaj zaposelnika 31](#_Toc405174879)

[10.2 Procedura briši zaposlenika 32](#_Toc257308483)

[10.3 Procedura dodavanje projekta 33](#_Toc527866953)

[10.4 Procedura ukupan izračun troškova puta 33](#_Toc1074767930)

[10.5 Procedura dodavanje godišnjeg odmora za zaposlenika 34](#_Toc688220041)

[10.6 Procedura prikazivanja svih smjena zaposlenika za određeni dan 34](#_Toc1327983498)

[10.7 Procedura dodavanja bilješka za zaposlenika 35](#_Toc793129397)

[10.8 Procedura odobravanja godišnjeg 35](#_Toc999177069)

[10.9 Procedura dodavanja zaposlenika u smjene 37](#_Toc1335856826)

[10.10 Procedura preraspodjele zaposlenika za godišnji odmor 38](#_Toc158811594)

[10.11 Procedura korisnik prihvaća godišnji odmor 41](#_Toc831751583)

[10.12 Procedura generiranja rasporeda 41](#_Toc1310718865)

[10.13 Procedura dodavanja prekovremenih sati 43](#_Toc1943287305)

[10.14 Procedura prijenos projekta drugom zaposleniku 44](#_Toc922698607)

[10.15 Procedura ukupni izvještaj troškova po mjesecu 45](#_Toc198232369)

[11. Funkcije 46](#_Toc892632000)

[11.1. Funkcija računanje mjesečne plaće 46](#_Toc839329613)

[11.2. Funkcija kalkuliranja troškova službenih putovanja po odjelu 48](#_Toc1105112169)

[11.3. Funkcija provjera dostupnosti zaposlenika 48](#_Toc1247285427)

[12.Okidači 50](#_Toc765780122)

[12.1. Okidač za ažuriranje statusa prekovremenog 50](#_Toc1921478636)

[12.2. Okidač za ažuriranje statusa projekta 50](#_Toc933402223)

[13.Inicijalizacija i konfiguracija 51](#_Toc1266029037)

[14.Rute i funkcionalnosti 51](#_Toc485006991)

# 1. Uvod

Sustav za upravljanje rasporedom radnog vremena zaposlenika osmišljen je kako bi pojednostavio i poboljšao proces planiranja, organizacije i praćenja radnog vremena unutar tvrtke. Ovaj sustav omogućuje automatizaciju kreiranja rasporeda te i njegove brze promjene. Zaposlenicima omogućuje postavljanje zahtjeva za smjene, godišnje odmore, praćenje bolovanja, evidenciju prisutnosti i uvid u plaće. Istovremeno, sustav olakšava upravljanje radnim procesima, čime donosi učinkovitijem i skladnijem poslovanju unutar tvrtke, olakšavajući posao zaposlenicima i poslodavcima.

Uz pomoć relacijske baze podataka, sustav osigurava ispravnost i konzistentnost podataka te omogućuje njihovu efikasnu obradu. Korisnicima će pružiti mogućnost unosa, izmjene i brisanja podataka o zaposlenicima, njihovim radnim smjenama i odsutnostima. Dodatno, omogućit će generiranje izvještaja koji podržavaju donošenje odluka vezanih uz upravljanje ljudskim resursima.

Razvoj sustava temelji se na tehnikama rada s bazama podataka, uključujući složene upite, procedure, čime se osigurava funkcionalnost i praktičnost u stvarnim poslovnim okruženjima. Tim je samostalno obradio ovu temu kako bi stvorio rješenje koje ima stvarnu primjenu u poslovanju te ujedno osigurao da sustav u potpunosti ispunjava tehničke i funkcionalne zahtjeve definirane projektom.

# 2. Opis poslovnog procesa

Poslovni proces upravljanja rasporedom radnog vremena zaposlenika obuhvaća planiranje, organizaciju i praćenje radnih smjena, godišnjih odmora, bolovanja, pauza, te evidenciju prisutnosti i odsutnosti unutar organizacije. Cilj ovog procesa je osigurati da se ljudski resursi koriste na najbolji mogući način, da se poštuju radni propisi te da se zadovolje potrebe zaposlenika i same tvrtke.

Proces započinje prikupljanjem podataka o zaposlenicima, uključujući njihovo ime, prezime, kontakt podatke, poziciju u organizaciji, radne smjene te eventualne specifične zahtjeve poput preferencija za rad u određenim smjenama ili godišnjih odmora. Nakon što su svi potrebni podaci evidentirani, voditelji odjela ili odgovorne osobe koriste sustav za kreiranje inicijalnog rasporeda radnih smjena.

Sustav omogućuje:

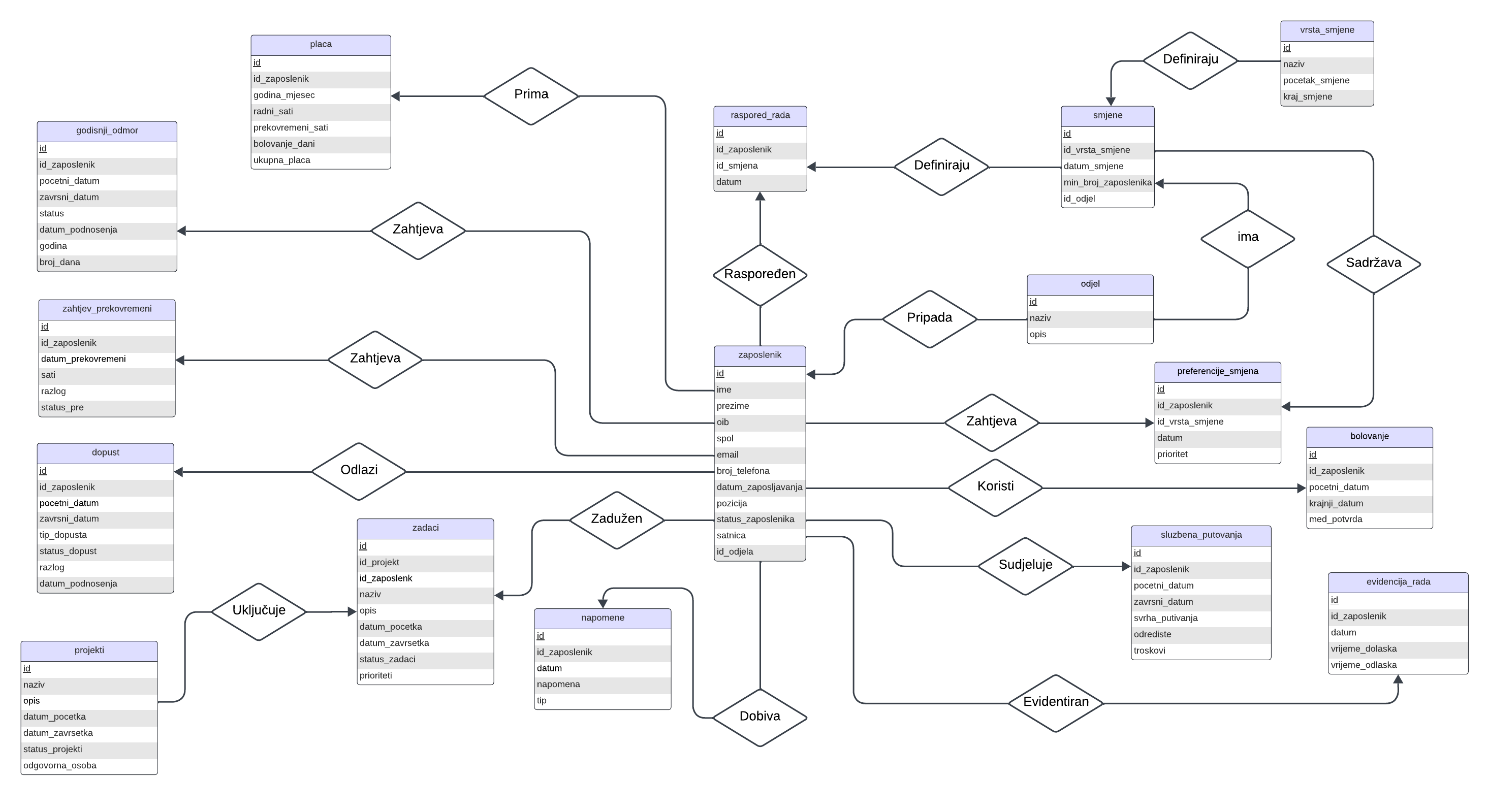
- Planiranje: Unos i automatsko generiranje rasporeda prema poslovnim pravilima te prilagodbu smjena prema potrebama.

- Organizaciju: Evidenciju odsutnosti i usklađivanje rasporeda s uvjetima zaposlenika.

- Praćenje i ažuriranje: Evidenciju radnih sati i brze izmjene rasporeda

Na kraju sustav omogućuje generiranje izvješća o radnim satima, prekovremenom radu i drugim ključnim informacijama. Implementacija ovog poslovnog procesa u sustavu donosi brojne koristi, uključujući povećanje učinkovitosti u planiranju, smanjenje administrativnog opterećenja te veću transparentnost i zadovoljstvo zaposlenika.

# 3. ER dijagram



# 4. Opis ER dijagrama

Zaposlenik **ima** odjel  
Svaki zaposlenik pripada jednom odjelu, no jedan odjel može imati više zaposlenika.  
Veza: 1:M

Zaposlenik **prima** plaću  
Zaposlenik može primati više plaća tijekom godine, no svaka plaća pripada samo jednom zaposleniku.  
Veza: 1:M

Zaposlenik **ima** raspored rada  
Zaposlenik može raditi prema više rasporeda rada, no svaki raspored rada uključuje samo jednog zaposlenika.  
Veza: 1:M

Smjene **definiraju** raspored rada  
Svaka smjena može biti uključena u više rasporeda rada, no svaki raspored rada pripada samo jednoj smjeni.  
Veza: 1:M

Zaposlenik **ima** preferencije smjena  
Zaposlenik može imati više preferencija za određene smjene, no svaka preferencija je povezana s jednim zaposlenikom i jednom vrstom smjene.  
Veza: 1:M

Zaposlenik **odlazi** na bolovanje  
Zaposlenik može biti na bolovanju više puta, no svako bolovanje pripada samo jednom zaposleniku.  
Veza: 1:M

Zaposlenik podnosi **zahtjev** za godišnji odmor  
Zaposlenik može podnijeti više zahtjeva za godišnji odmor, no svaki zahtjev pripada samo jednom zaposleniku.  
Veza: 1:M

Zaposlenik može podnijeti **zahtjev** za prekovremene sate  
Zaposlenik može podnijeti više zahtjeva za prekovremene sate, no svaki zahtjev pripada samo jednom zaposleniku.  
Veza: 1:M

Zaposlenik **sudjeluje** u službenim putovanjima  
Zaposlenik može sudjelovati u više službenih putovanja, no svako službeno putovanje vezano je za samo jednog zaposlenika.  
Veza: 1:M

Projekti **uključuju** zadatke  
Jedan projekt može sadržavati više zadataka, no svaki zadatak pripada samo jednom projektu.  
Veza: 1:M

Zaposlenik **zadužen** za zadatke  
Zaposlenik može biti zadužen za više zadataka, no svaki zadatak može biti dodijeljen samo jednom zaposleniku.  
Veza: 1:M

Zaposlenik može **dobiti** napomene  
Zaposlenik može imati više napomena tijekom rada, no svaka napomena odnosi se samo na jednog zaposlenika.  
Veza: 1:M

Zaposlenik je **evidentiran** u evidenciji rada  
Zaposlenik može imati više unosa u evidenciji rada (po danima), dok je svaki unos vezan samo za jednog zaposlenika.  
Veza: 1:M

# 5. Shema tablica

**Zaposlenik**  
id, ime, prezime, oib, spol, email, broj\_telefona, datum\_zaposlenja, pozicija, status\_zaposlenika, satnica, id\_odjel

**Odjel**  
id, naziv, opis

**Plaća**  
id, id\_zaposlenik, godina\_mjesec, radni\_sati, prekovremeni\_sati, broj\_dana\_bolovanja, ukupna\_placa

**Raspored rada**  
id, id\_zaposlenik, id\_smjena, datum

**Smjene**  
id, id\_vrsta\_smjene, minimalni\_broj\_zaposlenika, id\_odjel

**Vrsta smjene**  
id, naziv, vrijeme\_pocetka, vrijeme\_zavrsetka

**Bolovanje**  
id, id\_zaposlenik, datum\_pocetka, datum\_zavrsetka, medicinska\_potvrda

**Zahtjev za prekovremene sate**  
id, id\_zaposlenik, datum\_prekovremenog\_rada, broj\_sati, razlog, status\_zahtjeva

**Godišnji odmor**  
id, id\_zaposlenik, datum\_pocetka, datum\_zavrsetka, status, datum\_podnosenja, godina, broj\_dana

**Preferencije smjena**  
id, id\_zaposlenik, id\_vrsta\_smjene, datum, prioritet

**Evidencija rada**  
id, id\_zaposlenik, datum, vrijeme\_dolaska, vrijeme\_odlaska

**Dopust**  
id, id\_zaposlenik, datum\_pocetka, datum\_zavrsetka, tip\_dopusta, status, razlog, datum\_podnosenja

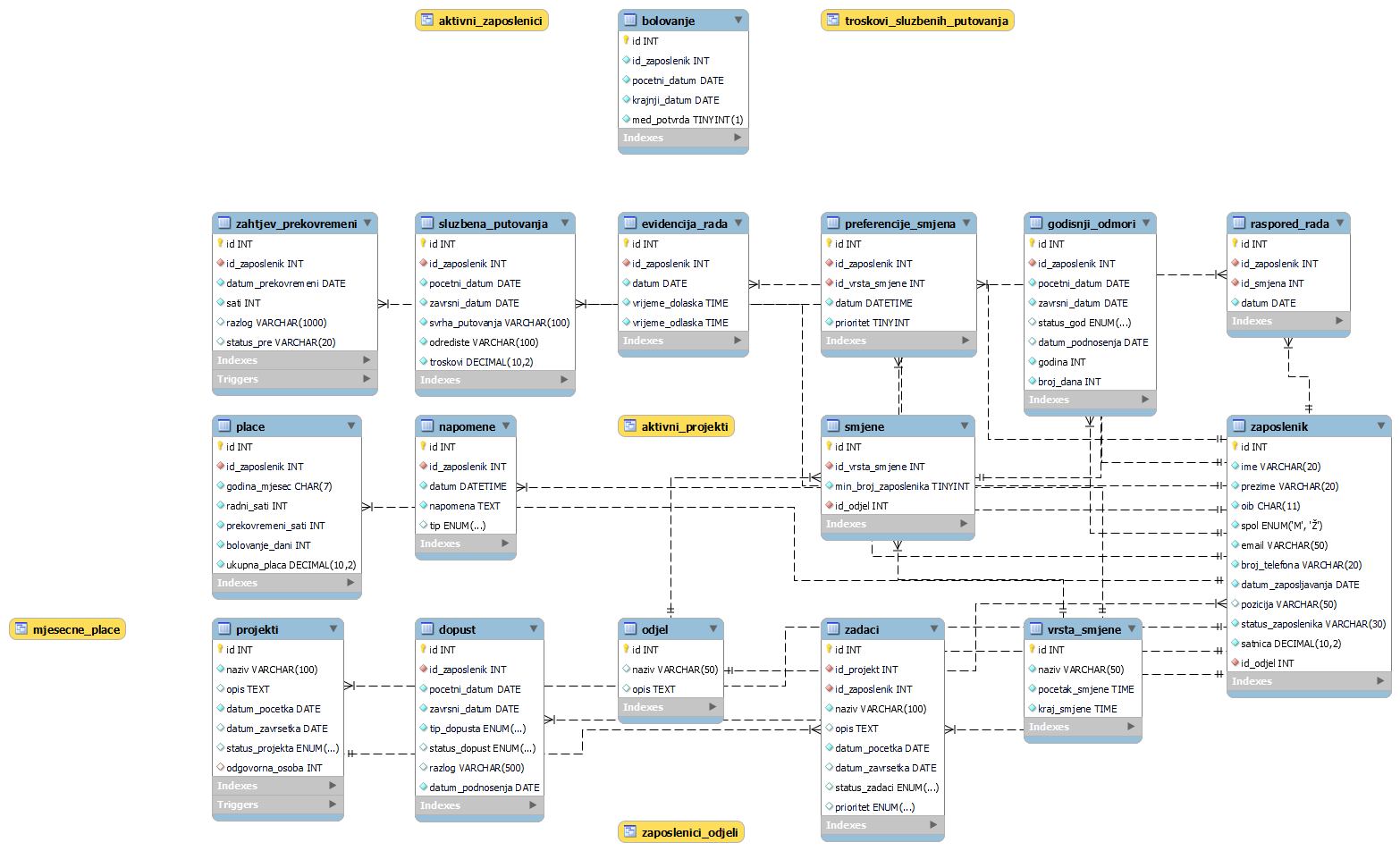
**Službena putovanja**  
id, id\_zaposlenik, datum\_pocetka, datum\_zavrsetka, svrha, odrediste, troskovi

**Projekti**  
id, naziv, opis, datum\_pocetka, datum\_zavrsetka, status, id\_odgovorna\_osoba

**Zadaci**  
id, id\_projekt, id\_zaposlenik, naziv, opis, datum\_pocetka, datum\_zavrsetka, status, prioritet

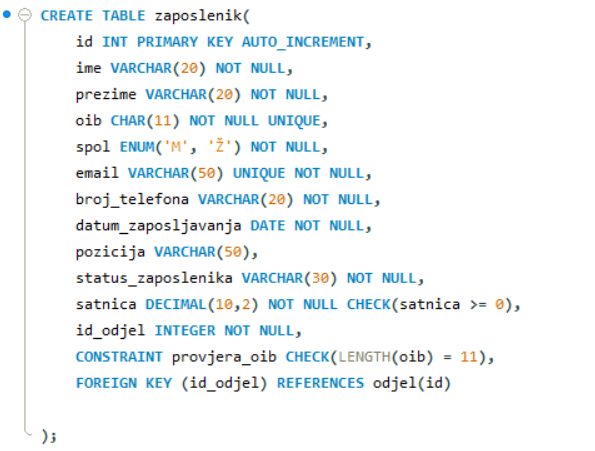
**Napomene**  
id, id\_zaposlenik, datum, napomena, tip

# 6. EER dijagram



# 7. Tablice

## 7.1. Tablica zaposlenik



**id** je primarni ključ (PRIMARY KEY) tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**ime** i **prezime** su atributi tipa VARCHAR s maksimalno 20 znakova i označavaju ime i prezime zaposlenika. Oba polja su obavezna (NOT NULL).

**oib** je atribut tipa CHAR s točno 11 znakova koji pohranjuje jedinstveni identifikacijski broj zaposlenika. Postavljena su ograničenja NOT NULL i UNIQUE kako bi svaki OIB bio obavezan i jedinstven. Dodatno ograničenje CHECK(LENGTH(oib) = 11) dodatno osigurava da je duljina točno 11 znakova.

**spol** je atribut tipa ENUM koji prihvaća samo dvije dozvoljene vrijednosti: 'M'(muški) i 'Ž' (ženski). Polje je obavezno (NOT NULL).

**email** je atribut tipa VARCHAR(50) koji pohranjuje e-mail adresu zaposlenika. Polje je obavezno (NOT NULL) i mora biti jedinstveno (UNIQUE).

**broj\_telefona** je atribut tipa VARCHAR(20) i pohranjuje telefonski broj zaposlenika. Polje je obavezno (NOT NULL).

**datum\_zaposljavanja** je atribut tipa DATE i pohranjuje datum kada je zaposlenik počeo raditi. Polje je obavezno (NOT NULL).

**pozicija** je atribut tipa VARCHAR(50) i pohranjuje naziv pozicije zaposlenika.

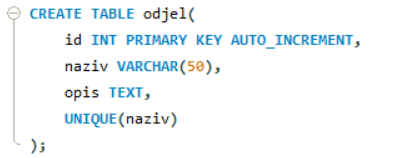
**status\_zaposlenika** je atribut tipa VARCHAR(30) i opisuje trenutni status zaposlenika (npr. "aktivan", "na probnom roku"). Polje je obavezno (NOT NULL).

**satnica** je atribut tipa DECIMAL(10,2) koji pohranjuje satnicu zaposlenika. Maksimalno dozvoljenih znamenki je 10, od čega su 2 decimalna mjesta. Postavljeno je ograničenje CHECK(satnica >= 0) koje osigurava da satnica ne može biti negativna.

**id\_odjel** je strani ključ (FOREIGN KEY) tipa INTEGER koji se povezuje s primarnim ključem id u tablici odjel. Ovo polje je obavezno (NOT NULL).

Strani ključ osigurava referencijalni integritet, tj. svaki id\_odjel u tablici zaposlenik mora postojati u tablici odjel.

## 7.2 Tablica Odjela



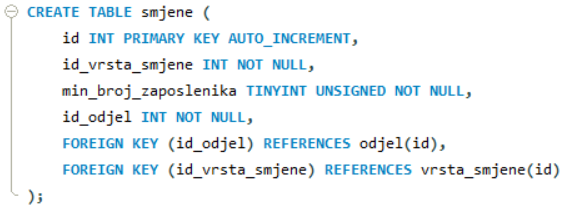
**id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa u tablici.

**naziv** je atribut tipa VARCHAR(50) koji pohranjuje naziv odjela, s maksimalno 50 znakova. Postavljeno je ograničenje UNIQUE, što znači da svaki odjel mora imati jedinstven naziv, bez dupliciranja.

**opis** je atribut tipa TEXT koji služi za pohranu dodatnih informacija o odjelu. Budući da je tip TEXT, omogućava unos većih količina teksta, ali unos ovog podatka nije obavezan.

Ova tablica omogućuje jedinstveno identificiranje odjela putem atributa **id**, dok naziv osigurava prepoznatljivost svakog odjela. Povezana je s drugim tablicama putem stranih ključeva, kao što je id\_odjel u tablici zaposlenik.

## 7.3 Tablica Smjene



**id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**id\_vrsta\_smjene** je strani ključ (FOREIGN KEY) tipa INTEGER koji se povezuje s primarnim ključem id u tablici .

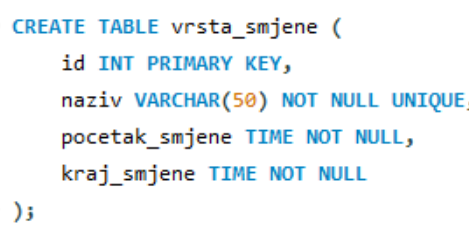
**min\_broj\_zaposlenika** je atribut tipa TINYINT UNSIGNED koji pohranjuje minimalan broj zaposlenika potrebnih za određenu smjenu. Tip TINYINT UNSIGNED omogućava samo pozitivne cijele brojeve (0 do 255). Ovo polje je obavezno (NOT NULL).

**id\_odjel** je strani ključ tipa INTEGER koji se povezuje s primarnim ključem id u tablici odjel. Ovo polje je obavezno (NOT NULL).

FOREIGN KEY (id\_vrsta\_smjene) REFERENCES vrsta\_smjene(id) osigurava integritet tako što povezuje vrijednosti iz stupca id\_vrsta\_smjene s primarnim ključem id u tablici vrsta\_smjene.

**FOREIGN KEY** (id\_odjel) REFERENCES odjel(id) osigurava da svaki zapis u atributu id\_odjel u tablici smjene mora odgovarati postojećem zapisu u tablici odjel.

## 7.4 Tablica vrsta smjene



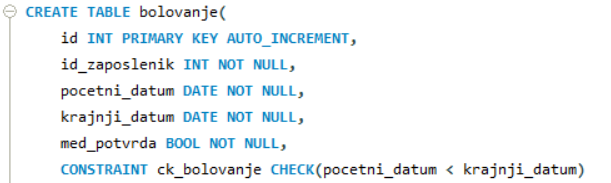
**id** je primarni ključ tipa INT koji osigurava jedinstvenost svakog zapisa.

**naziv** je atribut tipa VARCHAR u koji se može pohraniti do 50 znakova, ima ograničenje NOT NULL koje osigurava da polje mora biti popunjeno i ograničenje UNIQUE koje osigurava da je svaka nova vrijednost jedinstvena.

**pocetak\_smjene** je atribut tipa DATETIME koji bilježi točan datum i vrijeme početka smjene. Polje je obavezno (NOT NULL).

**kraj\_smjene** je atribut tipa DATETIME koji bilježi točan datum i vrijeme završetka smjene. Polje je također obavezno (NOT NULL).

## 7.4 Tablica Bolovanje



**id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se bolovanje odnosi. Ovo polje je obavezno (NOT NULL).

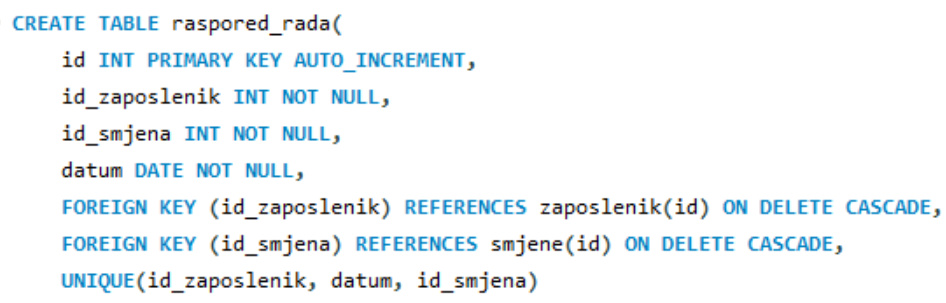
**pocetni\_datum** je atribut tipa DATE koji označava datum početka bolovanja. Unos ovog podatka je obavezan (NOT NULL).

**krajnji\_datum** je atribut tipa DATE koji označava datum završetka bolovanja. Unos ovog podatka je također obavezan (NOT NULL).

**med\_potvrda** je atribut tipa BOOL koji označava posjeduje li zaposlenik liječničku potvrdu za bolovanje. Vrijednosti mogu biti true ili false, a unos ovog podatka je obavezan (NOT NULL).

**CONSTRAINT** ck\_bolovanje CHECK(pocetni\_datum < krajnji\_datum) osigurava da datum početka bolovanja mora biti prije datuma završetka.

## 7.5 Tablica Raspored rada



**Id** je primarni ključ tipa INT koji se automatski povećava pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se zapis odnosi, a njegovo je unošenje obavezno.

**Id\_smjena** je atribut tipa INT koji označava smjenu dodijeljenu zaposleniku, također obavezno polje.

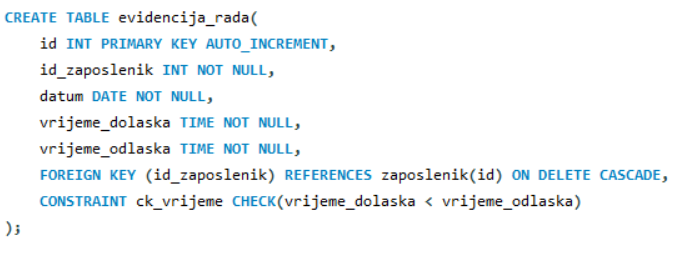
**Datum** je atribut tipa DATE koji označava dan na koji se smjena odnosi, a unos ovog podatka je neophodan.

**FOREIGN KEY** na atributu id\_zaposlenik povezan je s tablicom zaposlenik, čime se osigurava da svaki unos u ovoj tablici odgovara postojećem zaposleniku. Ako se zapis o zaposleniku izbriše, svi povezani zapisi u ovoj tablici također se brišu. Strani ključ na atributu id\_smjena povezan je s tablicom smjene, što osigurava da svaka smjena pripada postojećoj definiciji smjena.

**FOREIGN KEY** na atributu id**\_**smjena definira strani ključ povezan s tablicom smjene, a opcija ON DELETE CASCADE osigurava automatsko brisanje svih povezanih redaka u trenutnoj tablici kada se redak u tablici smjene obriše.

**UNIQUE** osigurava da ne može postojati više zapisa za istog zaposlenika na isti datum i s istom smjenom, čime se sprječava dupliciranje podataka. Tablica omogućuje precizno praćenje rasporeda smjena zaposlenika, uključujući datume i smjene na koje su dodijeljeni.

## 7.6 Tablica Evidencija rada



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se zapis odnosi, a njegovo je unošenje obavezno. Ovaj atribut je vanjski ključ koji se povezuje s tablicom zaposlenik, čime se osigurava da svaki unos u tablici evidencija\_rada odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici također će biti izbrisani zbog postavke ON DELETE CASCADE.

**Datum** je atribut tipa DATE koji označava datum evidencije dolaska i odlaska zaposlenika. Unos ovog podatka je obavezan.

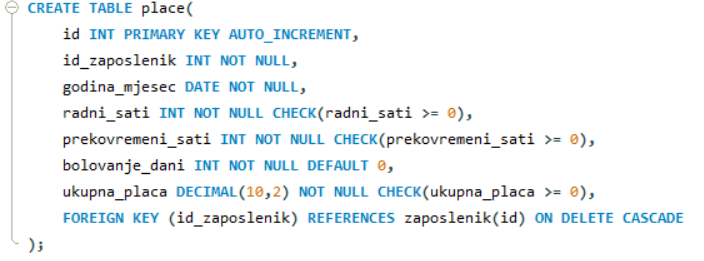
**Vrijeme\_dolaska** je atribut tipa TIME koji bilježi vrijeme kada je zaposlenik došao na posao. Ovaj podatak je također obavezan.

**Vrijeme\_odlaska** je atribut tipa TIME koji bilježi vrijeme kada je zaposlenik napustio posao. Unos ovog podatka je također obavezan.

Ograničenje **CHECK** (vrijeme\_dolaska < vrijeme\_odlaska) osigurava da vrijeme dolaska mora biti prije vremena odlaska.

**FOREIGN KEY** na atributu id\_zaposlenik povezan je s tablicom zaposlenik, čime se osigurava da svaki zapis u ovoj tablici odgovara postojećem zaposleniku. Ako se zapis o zaposleniku izbriše, svi povezani zapisi u tablici evidencija\_rada bit će automatski izbrisani.

## 7.7 Tablica place



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se zapis odnosi, a njegovo je unošenje obavezno. Ovaj atribut je vanjski ključ koji se povezuje s tablicom zaposlenik, čime se osigurava da svaki unos u tablici place odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Godina\_mjesec** je atribut tipa DATE koji označava mjesec i godinu za koji se obračunava plaća. Unos ovog podatka je obavezan.

**Radni\_sati** je atribut tipa INT koji označava ukupan broj odrađenih sati u mjesecu. Ovo polje je obavezno i ne može imati negativnu vrijednost, što osigurava CHECK (radni\_sati >= 0).

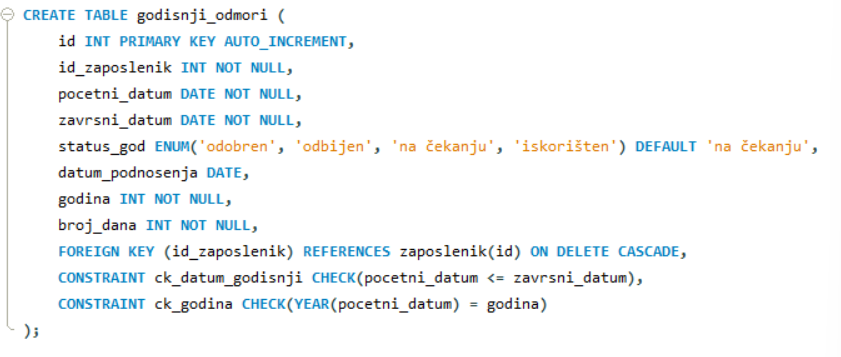
**Prekovremeni\_sati** je atribut tipa INT koji označava broj prekovremenih sati odrađenih u mjesecu. Ovo polje je obavezno i također ne može imati negativnu vrijednost, što osigurava CHECK (prekovremeni\_sati >= 0).

**Bolovanje\_dani** je atribut tipa INT koji označava broj dana bolovanja u mjesecu. Zadana vrijednost ovog atributa je 0, što znači da zaposlenik nije bio na bolovanju, osim ako nije drugačije uneseno.

**Ukupna\_placa** je atribut tipa DECIMAL(10,2) koji označava iznos isplaćene plaće za mjesec. Ovo polje je obavezno i ne može imati negativnu vrijednost, što osigurava CHECK (ukupna\_placa >= 0).

**FOREIGN KEY** na atributu id\_zaposlenik povezan je s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici place odgovara postojećem zaposleniku. Ako se zapis o zaposleniku izbriše, svi povezani zapisi u tablici place bit će automatski izbrisani.

## 7.8 Tablica Godisnji odmor



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se zapis odnosi, a njegovo je unošenje obavezno. Ovaj atribut je vanjski ključ koji se povezuje s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici godisnji\_odmori odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Pocetni\_datum** je atribut tipa DATE koji označava datum početka godišnjeg odmora. Ovo polje je obavezno.

**Zavrsni\_datum** je atribut tipa DATE koji označava datum završetka godišnjeg odmora. Ovo polje je obavezno.

**Status\_god** je atribut tipa ENUM koji označava status zahtjeva za godišnji odmor. Dozvoljene vrijednosti su 'odobren', 'odbijen', 'na čekanju' i 'iskorišten'. Zadana vrijednost je 'na čekanju'.

**Datum\_podnosenja** je atribut tipa DATE koji označava datum kada je zahtjev za godišnji odmor podnesen. Ovaj podatak nije obavezan.

**Godina** je atribut tipa INT koji označava godinu u kojoj se odmor koristi. Ovo polje je obavezno.

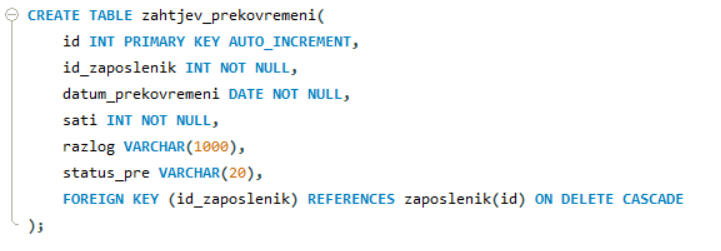
**Broj\_dana** je atribut tipa INT koji označava ukupan broj dana korištenog godišnjeg odmora. Ovo polje je obavezno.

**FOREIGN KEY** na atributu id\_zaposlenik povezan je s tablicom zaposlenik , čime se osigurava da svaki zapis u tablici godisnji\_odmori odgovara postojećem zaposleniku. Ako se zapis o zaposleniku izbriše, svi povezani zapisi u tablici godisnji\_odmori bit će automatski izbrisani.

**CONSTRAINT** ck\_datum\_godisnji CHECK (pocetni\_datum <= zavrsni\_datum): Osigurava da datum početka godišnjeg odmora mora biti prije ili jednak datumu završetka godišnjeg odmora.

**CONSTRAINT** ck\_godina CHECK (YEAR(pocetni\_datum) = godina): Osigurava da godina početnog datuma odmora mora odgovarati navedenoj godini.

## 7.9 Tablica Zahtjev prekovremeni



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se zapis odnosi, a njegovo je unošenje obavezno. Ovaj atribut je vanjski ključ koji se povezuje s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici zahtjev\_prekovremeni odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Datum\_prekovremeni** je atribut tipa DATE koji označava datum prekovremenog rada zaposlenika. Ovo polje je obavezno.

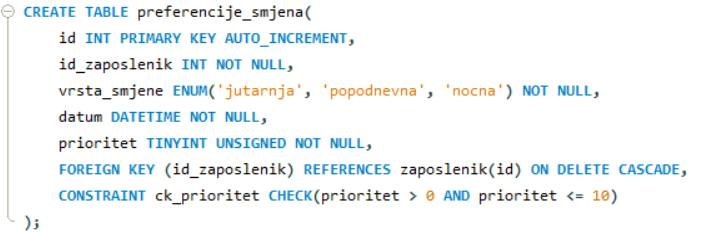
**Sati** je atribut tipa INT koji označava broj odrađenih prekovremenih sati. Ovo polje je obavezno.

**Razlog** je atribut tipa VARCHAR(1000) koji omogućava unos razloga za prekovremeni rad.

**Status\_pre** je atribut tipa VARCHAR(20) koji označava status zahtjeva za prekovremeni rad.

**FOREIGN KEY** na atributu id\_zaposlenik povezan je s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici zahtjev\_prekovremeni odgovara postojećem zaposleniku. Ako se zapis o zaposleniku izbriše, svi povezani zapisi u tablici zahtjev\_prekovremeni bit će automatski izbrisani.

## 7.10 Tablica Preferencije smjena



**id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika na kojeg se zapis odnosi, a njegovo je unošenje obavezno. Ovaj atribut je vanjski ključ koji se povezuje s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici preferencije\_smjena odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Id\_vrsta\_smjene** je atribut tipa INT koji se odnosi na id u tablici vrsta\_smjene i direktno ih povezuje putem stranog ključa, ima ograničenje koje osigurava da polje ne smije biti prazno.

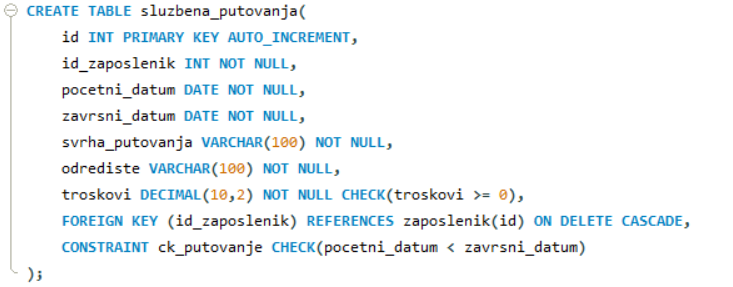
**Datum** je atribut tipa DATETIME koji označava datum i vrijeme kada zaposlenik preferira raditi određenu smjenu. Ovo polje je obavezno.

**Prioritet** je atribut tipa TINYINT UNSIGNED koji označava prioritet preferencije za smjenu. Prioritet može biti vrijednost od 1 do 10. Ovaj atribut je obavezan, a vrijednost mora biti veća od 0 i manja ili jednaka 10. Ograničenje CONSTRAINT ck\_prioritet osigurava da prioritet bude u ovom opsegu.

**FOREIGN KEY** na atributu id\_vrsta\_smjene definira strani ključ povezan s tablicom vrsta\_smjene, osiguravajući da vrijednosti u ovom stupcu odgovaraju postojećim vrijednostima u stupcu id tablice vrsta\_smjene.

**FOREIGN KEY** na atributu id\_zaposlenik povezan je s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici preferencije\_smjena odgovara postojećem zaposleniku. Ako se zapis o zaposleniku izbriše, svi povezani zapisi u tablici preferencije\_smjena bit će automatski izbrisani.

## 7.11 Tablica sluzbena putovanja



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika koji ide na službeno putovanje. Ovaj atribut je vanjski ključ povezan s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici sluzbena\_putovanja odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Pocetni\_datum** je atribut tipa DATE koji označava datum početka službenog putovanja. Ovo polje je obavezno.

**Zavrsni\_datum** je atribut tipa DATE koji označava datum završetka službenog putovanja. Ovo polje je obavezno.

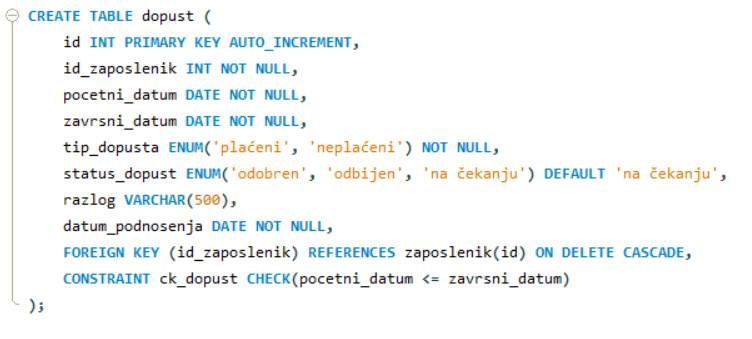
**Svrha\_putovanja** je atribut tipa VARCHAR(100) koji označava svrhu službenog putovanja. Ovo polje je obavezno.

**Odrediste** je atribut tipa VARCHAR(100) koji označava odredište službenog putovanja. Ovo polje je obavezno.

**Troskovi** je atribut tipa DECIMAL(10,2) koji označava ukupne troškove službenog putovanja. Ovo polje je obavezno, a vrijednost ne može biti negativna zbog ograničenja CHECK (troskovi >= 0).

Ograničenje **CONSTRAINT** ck\_putovanje osigurava da datum početka službenog putovanja mora biti prije datuma završetka.

## 7.12 Tablica Dopust



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT), osiguravajući jedinstvenost svakog zapisa.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika koji uzima dopust. Ovaj atribut je vanjski ključ povezan s tablicom zaposlenik, čime se osigurava da svaki zapis u tablici dopust odgovara postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Pocetni\_datum** je atribut tipa DATE koji označava datum početka dopusta. Ovo polje je obavezno.

**Zavrsni\_datum** je atribut tipa DATE koji označava datum završetka dopusta. Ovo polje je obavezno.

**Tip\_dopusta** je atribut tipa ENUM koji označava vrstu dopusta (plaćeni ili neplaćeni). Ovo polje je obavezno.

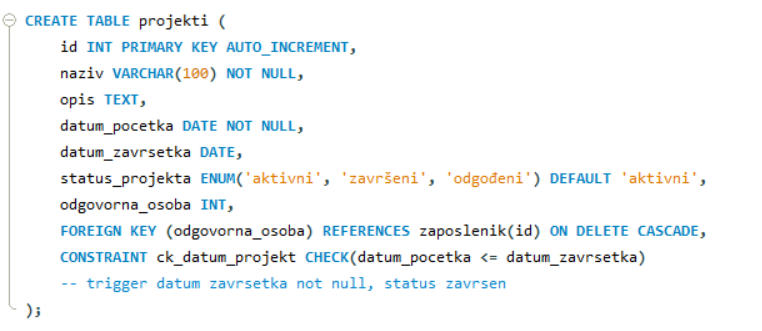
**Status\_dopust** je atribut tipa ENUM koji označava trenutni status dopusta (odobren, odbijen, na čekanju). Ovo polje ima zadanu vrijednost 'na čekanju'.

**Razlog** je atribut tipa VARCHAR(500) koji omogućava unos razloga za dopust. Ovo polje nije obavezno.

**Datum\_podnosenja** je atribut tipa DATE koji označava datum podnošenja zahtjeva za dopust. Ovo polje je obavezno.

Ograničenje **CONSTRAINT** ck\_dopust osigurava da datum početka dopusta mora biti prije ili jednak datumu završetka.

## 7.13 Tablica projekti



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Naziv** je atribut tipa VARCHAR(100) koji označava naziv projekta. Ovo polje je obavezno.

**Opis** je atribut tipa TEXT koji omogućava unos detaljnog opisa projekta. Ovo polje nije obavezno.

**Datum\_pocetka** je atribut tipa DATE koji označava datum početka projekta. Ovo polje je obavezno.

**Datum\_zavrsetka** je atribut tipa DATE koji označava datum završetka projekta. Ovo polje nije obavezno.

**Status\_projekta** je atribut tipa ENUM koji označava trenutni status projekta (aktivni, završeni, odgođeni). Ovo polje ima zadanu vrijednost 'aktivni'.

**Odgovorna\_osoba** je atribut tipa INT koji označava zaposlenika odgovornog za projekt. Ovaj atribut je vanjski ključ povezan s tablicom zaposlenik, čime se osigurava da svaki projekt ima odgovornu osobu. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

Ograničenje **CONSTRAINT** ck\_datum\_projekt osigurava da datum početka projekta mora biti prije ili jednak datumu završetka.

## 7.14 Tablica Zadaci



**Id** je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

**Id\_projekt** je atribut tipa INT koji označava projekt u kojem se zadatak obavlja. Ovaj atribut je vanjski ključ povezan s tablicom projekti, čime se osigurava da svaki zadatak pripada postojećem projektu.

**Id\_zaposlenik** je atribut tipa INT koji označava zaposlenika kojem je zadatak dodijeljen. Ovaj atribut je vanjski ključ povezan s tablicom zaposlenik, čime se osigurava da svaki zadatak bude dodijeljen postojećem zaposleniku. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

**Naziv** je atribut tipa VARCHAR(100) koji označava naziv zadatka. Ovo polje je obavezno.

**Opis** je atribut tipa TEXT koji omogućava unos detaljnog opisa zadatka. Ovo polje nije obavezno.

**Datum\_pocetka** je atribut tipa DATE koji označava datum početka zadatka. Ovo polje je obavezno.

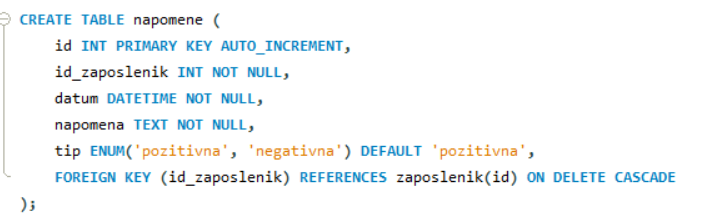
**Datum\_zavrsetka** je atribut tipa DATE koji označava datum završetka zadatka. Ovo polje nije obavezno.

**Status\_zadaci** je atribut tipa ENUM koji označava trenutni status zadatka (u tijeku, završeni, odgođeni). Ovo polje ima zadanu vrijednost 'u tijeku'.

**Prioritet** je atribut tipa ENUM koji označava prioritet zadatka (nizak, srednji, visok). Ovo polje ima zadanu vrijednost 'srednji'.

Ograničenje **CONSTRAINT** ck\_datum\_zadaci osigurava da datum početka zadatka mora biti prije ili jednak datumu završetka zadatka.

## 7.15 Tablica Napomene



Id je primarni ključ tipa INT koji se automatski povećava (AUTO\_INCREMENT) pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

Id\_zaposlenik je atribut tipa INT koji označava zaposlenika na kojeg se napomena odnosi. Ovaj atribut je vanjski ključ povezan s tablicom zaposlenik, čime se osigurava da svaka napomena bude povezana s postojećim zaposlenikom. Ako se zaposlenik izbriše, svi povezani zapisi u ovoj tablici bit će automatski izbrisani zbog postavke ON DELETE CASCADE.

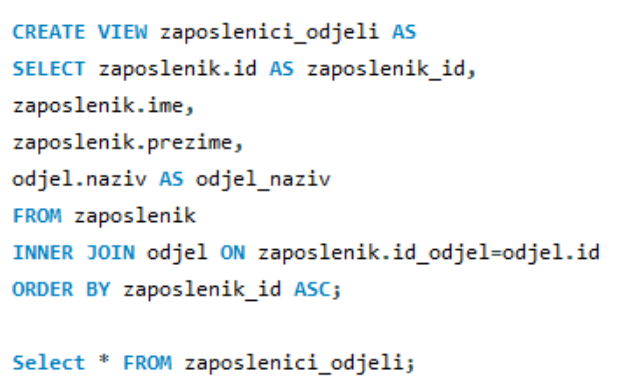
Datum je atribut tipa DATETIME koji označava datum i vrijeme kada je napomena unesena. Ovo polje je obavezno.

Napomena je atribut tipa TEXT koji omogućava unos detaljnog teksta napomene. Ovo polje je obavezno.

Tip je atribut tipa ENUM koji označava tip napomene (pozitivna ili negativna). Ovo polje ima zadanu vrijednost 'pozitivna'.

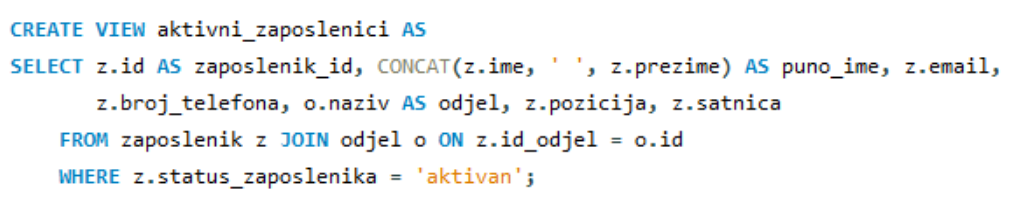
# 8. Pogledi

## 8.1 Pogled zaposlenici i odjeli



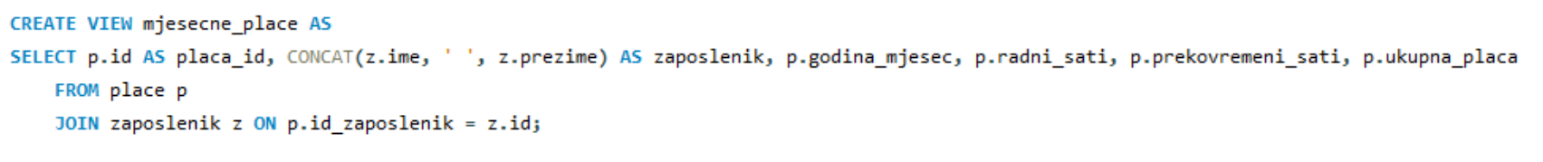
Pogled **zaposlenici\_odjeli** prikazuje povezan prikaz podataka iz tablica zaposlenik i odjel. Prvo se koristi INNER JOIN kako bi se povezali zapisi iz tablice zaposlenik i tablice odjel na temelju zajedničkog stupca id\_odjel u tablici zaposlenik i id u tablici odjel. Zatim se selektiraju četiri stupca: id iz tablice zaposlenik, koji se preimenuje u zaposlenik\_id, ime i prezime iz tablice zaposlenik, te naziv odjela (naziv), koji se preimenuje u odjel\_naziv. Rezultati se sortiraju prema zaposlenik\_id u rastućem (ASC) redoslijedu. Ovaj pogled omogućava jednostavniji pristup podacima koji kombiniraju informacije o zaposlenicima i njihovim odjelima.

## 8.2 Pogled aktivnih zaposlenika i osnovnih informacija o njima



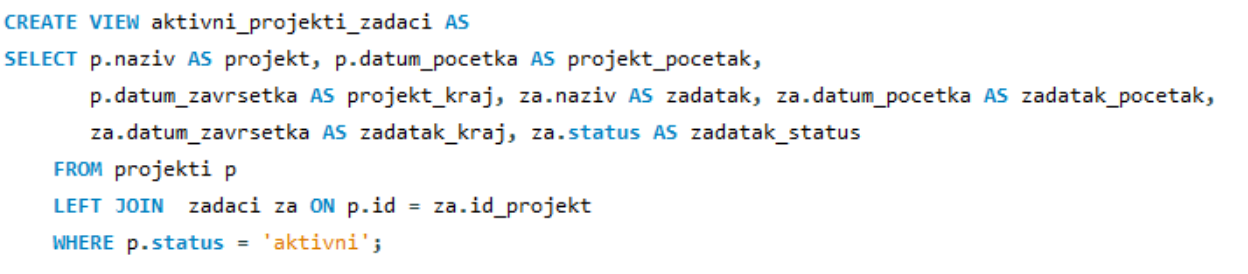
Pogled **aktivni\_zaposlenici** prikazuje podatke o zaposlenicima koji imaju status 'aktivan'. U upitu se koristi JOIN između tablica zaposlenik i odjel na temelju zajedničkog stupca id\_odjel. Odabrani su stupci: id iz tablice zaposlenik, koji je preimenovan u zaposlenik\_id, spojeno ime i prezime zaposlenika u jedno polje puno\_ime, te email, broj telefona, naziv odjela (odjel), pozicija i satnica zaposlenika. Rezultati se filtriraju uvjetom WHERE z.status\_zaposlenika = 'aktivan', tako da se prikazuju samo zaposleni koji su trenutno aktivni. Ovaj pogled omogućava lakši pristup podacima o aktivnim zaposlenicima zajedno s informacijama o njihovim odjelima.

## 8.3 Pogled mjesečnih plaća po zaposleniku



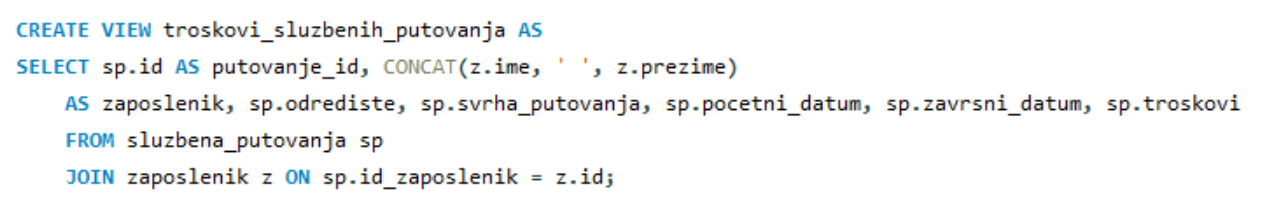
**Pogled mjesecne\_place** prikazuje informacije o mjesečnim plaćama zaposlenika. U upitu se koristi JOIN između tablica place i zaposlenik na temelju zajedničkog stupca id\_zaposlenik. Odabrani su stupci: id iz tablice place, koji je preimenovan u placa\_id, spojeno ime i prezime zaposlenika u jedno polje zaposlenik, te godina i mjesec (stupac godina\_mjesec), radni sati, prekovremeni sati i ukupna plaća (stupac ukupna\_placa). Ovaj pogled omogućava pregled svih mjesečnih plaća zaposlenika zajedno s detaljima o broju radnih i prekovremenih sati.

## 8.4 Pogled aktivnih projekata i zadataka



**Pogled aktivni\_projekti\_zadaci** prikazuje informacije o aktivnim projektima i njihovim zadacima. U prvom dijelu, koristi se LEFT JOIN između tablica projekti i zadaci na temelju zajedničkog stupca id iz tablice projekti i id\_projekt iz tablice zadaci. Odabrani su stupci: naziv projekta (projekt), datum početka i završetka projekta (projekt\_pocetak i projekt\_kraj), naziv zadatka (zadatak), datum početka i završetka zadatka (zadatak\_pocetak i zadatak\_kraj), te status zadatka (zadatak\_status). U uvjetu WHERE p.status = 'aktivni' filtriraju se samo aktivni projekti. Ovaj pogled omogućava uvid u projekte koji su aktivni zajedno s njihovim zadacima, uključujući njihove početne i završne datume te status.

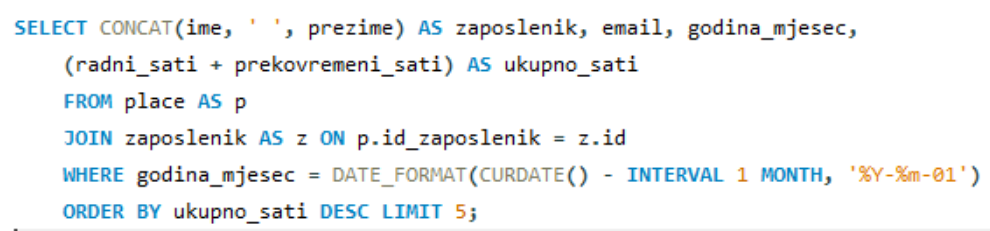
## 8.5 Pogled troškova službenih putovanja



Pogled **troskovi\_sluzbenih\_putovanja** prikazuje informacije o troškovima službenih putovanja zaposlenika. U upitu se koristi JOIN između tablica sluzbena\_putovanja i zaposlenik na temelju zajedničkog stupca id\_zaposlenik u tablici sluzbena\_putovanja i id u tablici zaposlenik. Odabrani su stupci: id iz tablice sluzbena\_putovanja, koji je preimenovan u putovanje\_id, spojeno ime i prezime zaposlenika u jedno polje zaposlenik, odredište (odrediste), svrha putovanja (svrha\_putovanja), datum početka i završetka putovanja (pocetni\_datum i zavrsni\_datum), te ukupni troškovi putovanja (troskovi). Ovaj pogled omogućava praćenje troškova službenih putovanja zajedno s informacijama o zaposlenicima koji su na tim putovanjima.

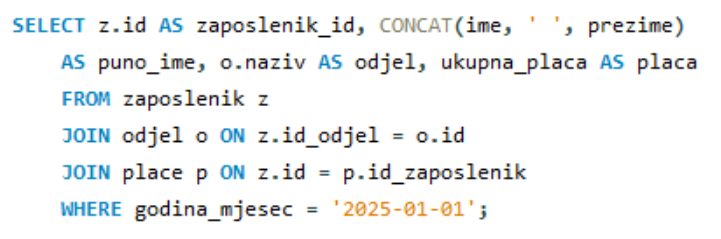
# 9. Upiti

## 9.1 Zaposlenici s najviše odrađenih sati protekli mjesec



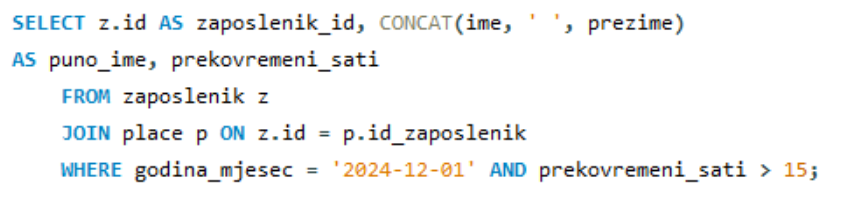
Upit prikazuje zaposlenike koji su odradili najviše sati u proteklom mjesecu. Upit koristi JOIN između tablica place i zaposlenik na temelju zajedničkog stupca id\_zaposlenik. Odabrani su stupci: kombinirano ime i prezime zaposlenika (zaposlenik), email zaposlenika, godina i mjesec (godina\_mjesec), te ukupni broj sati koji je zbroj radnih sati i prekovremenih sati. U uvjetu WHERE filtrira podatke za protekli mjesec koristeći funkciju DATE\_FORMAT i CURDATE(). Rezultati se sortiraju prema ukupnim satima u silaznom (DESC) redoslijedu, a upit prikazuje samo prvih 5 zaposlenika koji su odradili najviše sati.

## 9.2 Upit svih zaposlenika s njihovim odjelima i ukupnom plaćom u određenom mjesecu



Upit prikazuje informacije o zaposlenicima, njihovim plaćama i odjelima za specifični mjesec, u ovom slučaju za siječanj 2025. godine. Upit koristi JOIN između tablica zaposlenik, odjel i place na temelju zajedničkih stupaca: id\_odjel u tablici zaposlenik i id u tablici odjel, te id\_zaposlenik u tablici place i id u tablici zaposlenik. Odabrani su stupci: id zaposlenika, koji je preimenovan u zaposlenik\_id, spojeno ime i prezime zaposlenika u jedno polje puno\_ime, naziv odjela odjel, te ukupna plaća placa. Rezultati se filtriraju prema mjesecu 2025-01-01 za stupac godina\_mjesec, prikazujući plaće zaposlenika za siječanj 2025. godine.

## 9.3 Upit koji prikazuje koji su zaposlenici radili više od 15 prekovremenih sati određeni mjesec



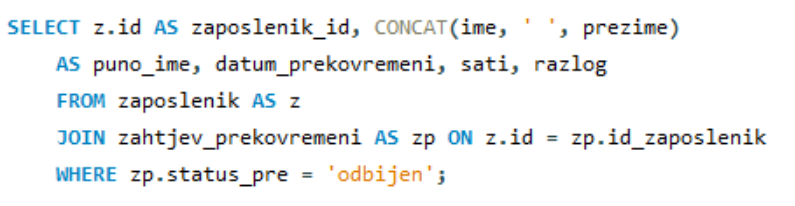
Upit prikazuje informacije o zaposlenicima koji su imali više od 15 prekovremenih sati u prosincu 2024. godine. Upit koristi JOIN između tablica zaposlenik i place na temelju zajedničkog stupca id u tablici zaposlenik i id\_zaposlenik u tablici place. Odabrani su stupci: id zaposlenika, koji je preimenovan u zaposlenik\_id, spojeno ime i prezime zaposlenika u jedno polje puno\_ime, te broj prekovremenih sati prekovremeni\_sati. Rezultati se filtriraju prema uvjetima godina\_mjesec = 2024-12-01 i prekovremeni\_sati > 15, čime se prikazuju samo oni zaposlenici koji su imali više od 15 prekovremenih sati.

## 9.4 Upit s projektima koji ističu u sljedećih 7 dana



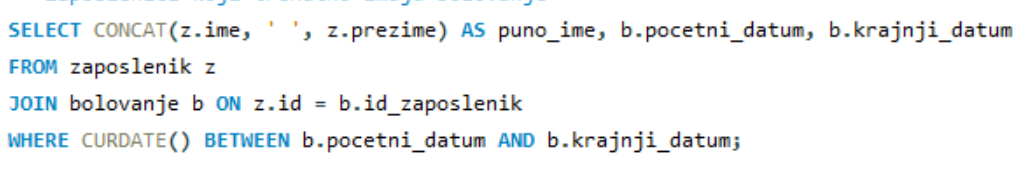
Upit prikazuje projekte koji imaju rokove koji ističu u sljedećih 7 dana. Upit odabire stupce: naziv projekta (naziv), opis projekta (opis) i datum završetka projekta (datum\_zavrsetka) iz tablice projekti. Koristi se uvjet WHERE kako bi se filtrirali projekti koji su aktivni (status 'aktivni'), a čiji je datum završetka unutar vremenskog okvira od današnjeg datuma (CURDATE()) do 7 dana od danas (DATE\_ADD(CURDATE(), INTERVAL 7 DAY)). Ovaj upit omogućava praćenje projekata čiji rokovi ističu uskoro, što pomaže u pravovremenom upravljanju rokovima.

## 9.5 Upit liste prekovremenih zahtjeva koji su odbijeni s razlozima



Upit prikazuje informacije o zaposlenicima čiji su zahtjevi za prekovremene sate odbijeni. Upit koristi JOIN između tablica zaposlenik i zahtjev\_prekovremeni na temelju zajedničkog stupca id\_zaposlenik. Odabrani su stupci: id zaposlenika, koji je preimenovan u zaposlenik\_id, spojeno ime i prezime zaposlenika u jedno polje puno\_ime, datum prekovremenih sati (datum\_prekovremeni), broj sati (sati) i razlog odbijenog zahtjeva (razlog). Rezultati se filtriraju prema uvjetu zp.status\_pre = 'odbijen', čime se prikazuju samo oni zaposlenici čiji su zahtjevi za prekovremene sate odbijeni.

## 9.6 Upit zaposlenika koji su trenutno na bolovanju



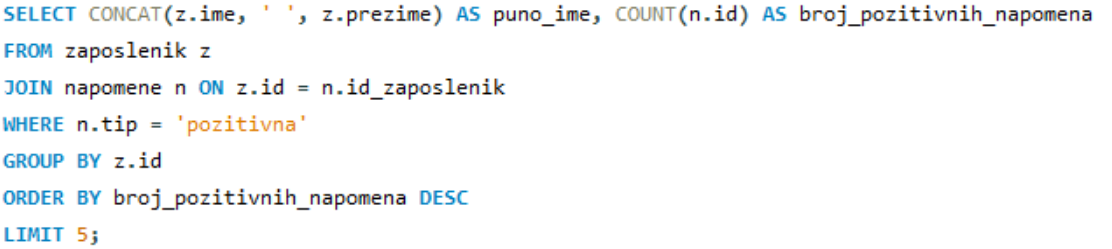
Upit dohvaća popis zaposlenika koji se trenutno nalaze na bolovanju, uključujući datume početka i završetka njihovog bolovanja. Koristi JOIN kako bi povezao tablice zaposlenik i bolovanje na temelju zajedničkog stupca id\_zaposlenik, koji povezuje svakog zaposlenika s pripadajućim zapisima bolovanja. U rezultatu, stupac CONCAT(z.ime, ' ', z.prezime) spaja ime i prezime zaposlenika u jedan prikaz, dok se za svakog zaposlenika prikazuju i datumi pocetni\_datum i krajnji\_datum njihovog bolovanja. WHERE koristi funkciju CURDATE(), koja vraća trenutni datum, kako bi se pronašli samo oni zapisi gdje trenutni datum pada unutar raspona datuma bolovanja.

## 9.7 Upit ukupan broj sati rada po zaposleniku za zadani mjesec



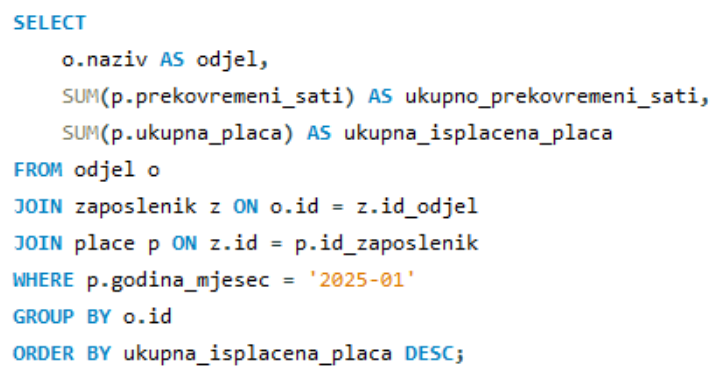
Upit izračunava ukupan broj sati rada svakog zaposlenika za određeni mjesec. Koristi JOIN kako bi povezao tablice zaposlenik i evidencija\_rada putem stupca id\_zaposlenik, koji povezuje svakog zaposlenika s njihovim zapisima u evidenciji rada. U stupcu CONCAT(z.ime, ' ', z.prezime), ime i prezime zaposlenika spajaju se u jedan formatirani prikaz. Zatim se pomoću funkcije TIMESTAMPDIFF(HOUR, e.vrijeme\_dolaska, e.vrijeme\_odlaska) računa broj sati rada za svaki zapis dolaska i odlaska, a funkcija SUM zbraja te sate za svaki zaposlenik. U uvjetu WHERE filtrira se zapis iz tablice evidencija\_rada tako da uključuje samo one koji pripadaju zadanom mjesecu. Grupiranje po stupcu z.id osigurava da se sati zbrajaju zasebno za svakog zaposlenika, dok ORDER BY ukupno\_sati DESC sortira rezultate prema ukupnom broju sati rada u silaznom redoslijedu, počevši od zaposlenika s najviše odrađenih sati.

## 9.8 Upit zaposlenika s najviše pozitivnih napomena



Upit pronalazi zaposlenike s najviše pozitivnih napomena i prikazuje prvih pet s najvećim brojem takvih zapisa. Pomoću JOIN povezuje tablice zaposlenik i napomene putem stupca id\_zaposlenik, koji identificira zaposlenike u obje tablice. Pomoću CONCAT(z.ime, ' ', z.prezime), ime i prezime zaposlenika formatiraju se u jedan prikaz. Funkcija COUNT(n.id) broji broj pozitivnih napomena povezanih sa svakim zaposlenikom. Uvjet WHERE n.tip = 'pozitivna' osigurava da se broje samo zapisi u tablici napomene koji imaju tip "pozitivna". Rezultati se grupiraju prema z.id kako bi se brojale napomene za svakog zaposlenika pojedinačno, a pomoću ORDER BY broj\_pozitivnih\_napomena DESC sortira zaposlenike prema broju pozitivnih napomena u silaznom redoslijedu. Ograničenje LIMIT 5 prikazuje samo prvih pet zaposlenika s najviše pozitivnih napomena.

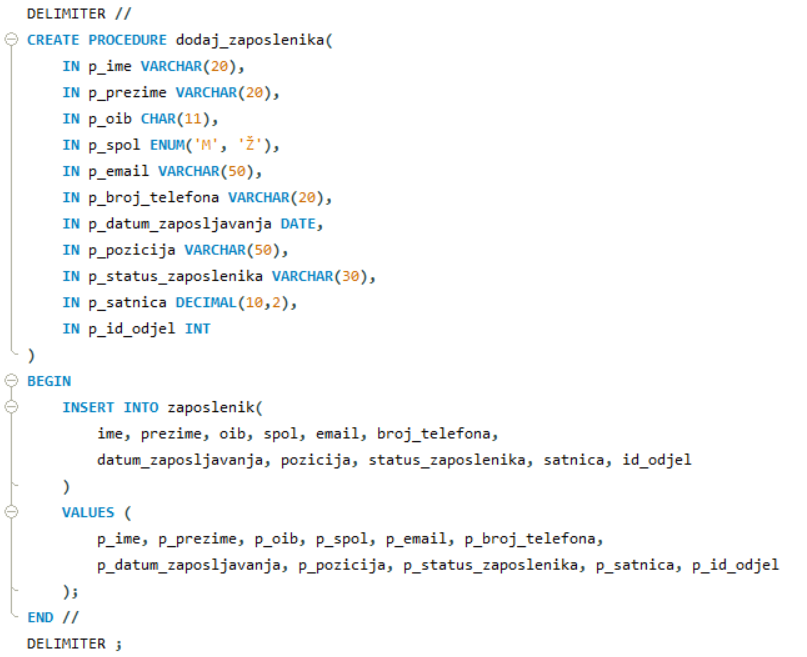
## 9.9 Upit ukupna plaća i broj prekovremenih sati po odjelu za određeni mjesec



Ovaj upit prikazuje informacije o svakom odjelu, uključujući ukupno odrađene prekovremene sate i ukupne isplaćene određeni mjesec. Pomoću JOIN povezuje tablice odjel, zaposlenik, i place, pri čemu se odjeli povezuju sa zaposlenicima putem id\_odjel, a zaposlenici s plaćama putem id\_zaposlenik. Stupac o.naziv označava naziv odjela, dok se funkcije SUM(p.prekovremeni\_sati) i SUM(p.ukupna\_placa) koriste za zbrajanje ukupnog broja prekovremenih sati i ukupnih isplaćenih plaća za sve zaposlenike u svakom odjelu. Uvjet WHERE p.godina\_mjesec = '2025-01' osigurava da se podaci filtriraju samo za siječanj 2025. Rezultati se grupiraju prema o.id, što znači da se podaci zbrajaju na razini odjela. ORDER BY ukupna\_isplacena\_placa DESC sortira rezultate prema ukupnoj isplaćenoj plaći po odjelu, u silaznom redoslijedu, tako da odjeli s najvećim troškovima plaća budu prikazani prvi.

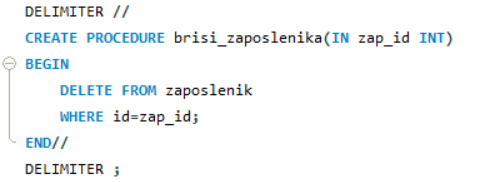
# 10. Procedure

## 10.1 Procedura dodaj zaposelnika



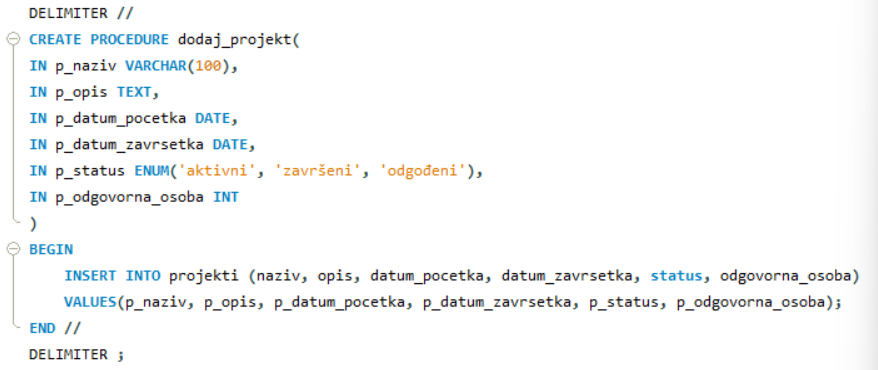
Procedura **dodaj\_zaposlenika** omogućava unos novog zaposlenika u tablicu zaposlenik. Procedura prima nekoliko ulaznih parametara za sve relevantne informacije o zaposleniku. Parametri su p\_ime, p\_prezime, p\_oib, p\_spol, p\_email, p\_broj\_telefona, p\_datum\_zaposljavanja, p\_pozicija, p\_status\_zaposlenika, p\_satnica, p\_id\_odjel. Unutar tijela procedure koristi se INSERT INTO kako bi se novi podaci unijeli u tablicu zaposlenik u odgovarajuće stupce. Vrijednosti za unos dolaze iz parametara koje je korisnik proslijedio prilikom poziva procedure. Ova procedura omogućava jednostavan unos novih zaposlenika s potrebnim informacijama bez potrebe za ručnim unosom podataka svakim upitom.

## 10.2 Procedura briši zaposlenika



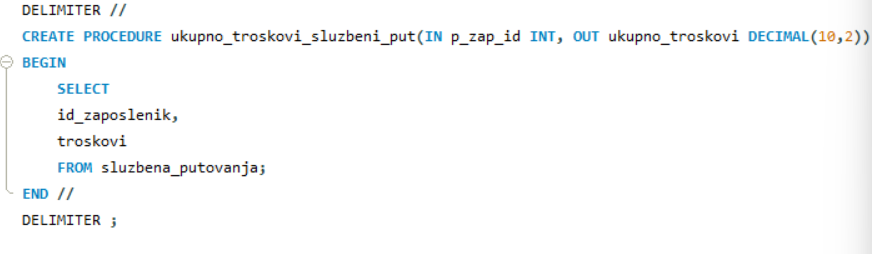
Procedura **brisi\_zaposlenika** omogućava brisanje zaposlenika iz tablice zaposlenik na temelju njegovog ID-a. Procedura prima jedan ulazni parametar, zap\_id, koji predstavlja ID zaposlenika kojeg treba obrisati. Unutar tijela procedure koristi se DELETE za uklanjanje zapisa u tablici zaposlenik gdje je id jednak vrijednosti proslijeđenoj kao parametar zap\_id. Ova procedura omogućava jednostavno upravljanje podacima zaposlenika jer omogućava njihovo brisanje pomoću jednog parametra, ID-a zaposlenika. Korištenje ove procedure čini administraciju baze podataka učinkovitijom i bržom.

## 10.3 Procedura dodavanje projekta



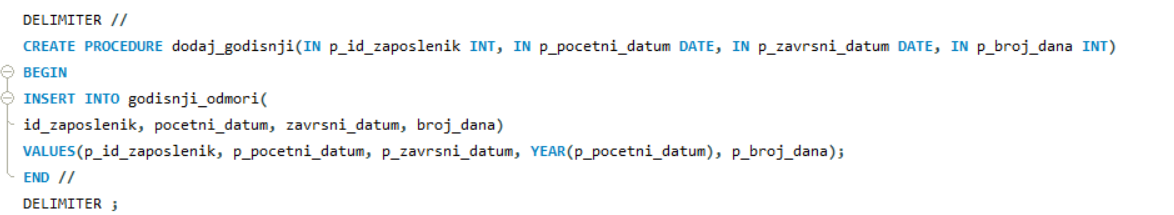
Procedura **dodaj\_projekt** omogućava unos novih projekata u tablicu projekti. Procedura prima šest ulaznih parametara: p\_naziv, p\_opis, p\_datum\_pocetka, p\_datum\_zavrsetka, p\_status, te p\_odgovorna\_osoba. Unutar procedure koristi se upit INSERT INTO projekti kako bi se uneseni podaci pohranili u odgovarajuće stupce tablice projekti. Ova procedura omogućava unos novih projekata uz zadane vrijednosti za naziv, opis, datume, status i odgovornu osobu. Na početku se koristi. Ova procedura čini unos novih projekata lakšim i bržim, jer sve informacije o projektu mogu biti unesene u jednom pozivu procedure.

## 10.4 Procedura ukupan izračun troškova puta



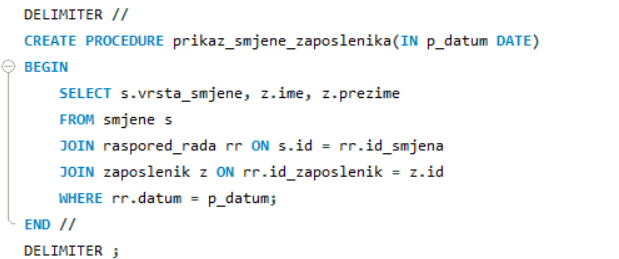
Procedura **ukupno\_troskovi\_sluzbeni\_put** služi za dohvaćanje ukupnih troškova službenih putovanja za određenog zaposlenika na temelju njegovog ID-a. Procedura prima dva parametra: p\_zap\_id, koji je ulazni parametar, te ukupno\_troskovi, koji je izlazni parametar koji će sadržavati ukupne troškove za tog zaposlenika. Unutar tijela procedure koristi se upit SELECT za dohvaćanje podataka o zaposleniku i njegovim troškovima iz tablice sluzbena\_putovanja. Upit vraća stupce id\_zaposlenik i troskovi za sve zaposlenike.

## 10.5 Procedura dodavanje godišnjeg odmora za zaposlenika



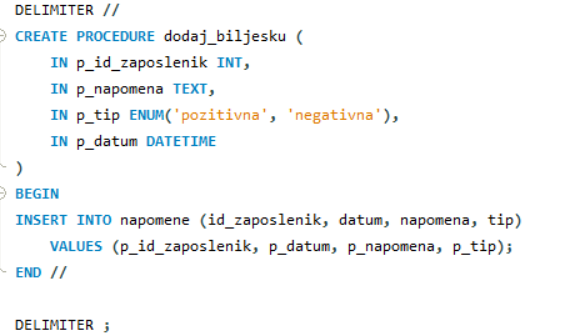
Procedura **dodaj\_godisnji** omogućava unos podataka o godišnjem odmoru zaposlenika u tablicu godisnji\_odmori. Procedura prima četiri ulazna parametra: p\_id\_zaposlenik (tipa INT), p\_pocetni\_datum (tipa DATE), p\_zavrsni\_datum (tipa DATE), i p\_broj\_dana (tipa INT). Unutar procedure koristimo INSERT INTO godisnji\_odmori kako bi se uneseni podaci pohranili u odgovarajuće stupce tablice. Vrijednosti za unos su parametri koje je korisnik proslijedio prilikom poziva procedure, uključujući početni datum, završni datum, broj dana, te ID zaposlenika.

## 10.6 Procedura prikazivanja svih smjena zaposlenika za određeni dan



Procedura **prikaz\_smjene\_zaposlenika** omogućuje pregled smjena i zaposlenika za određeni datum. Kao ulaz prima p\_datum(datum koji želimo prikaz smjena)(tip:DATE) a vraća popis smjena koje se održavaju na taj dan, zajedno s pripadajućim zaposlenicima raspoređenih na te smjene. Procedura dohvaća podatke iz tablice: smjene(informacije o vrstama smjena), raspored\_rada(povezuje smjene s određenim datumom i zaposlenicima) te i zaposlenik(informacije o zaposleniku kao ime i prezime).Procedura JOIN spaja tablice smjene i raspored\_rada pomoćuid\_smjena i raspored\_rada i zaposlenik pomoću id\_zaposlenik.

## 10.7 Procedura dodavanja bilješka za zaposlenika



Procedura **dodaj\_biljesku** omogućuje dodavanje bilješka(napomene) za određenog zaposlenika u sustavu.Kao ulazni parametri, procedura prima četiri vrijednosti: ID zaposlenika (p\_id\_zaposlenik), tekst bilješke (p\_napomena), tip bilješke (p\_tip), koji može biti pozitivna ili negativna, te datum i vrijeme kada bilješka treba biti dodana (p\_datum).Procedura izvršava **INSERT** upit u tablicu napomene gdje se spremaju podaci o bilješki.

## 10.8 Procedura odobravanja godišnjeg





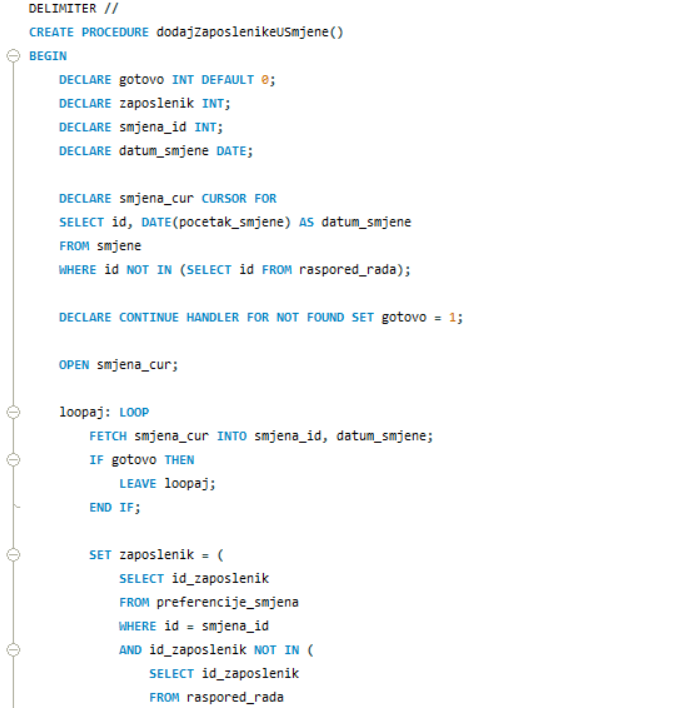
Procedura **odobrigodisnji** služi za odobravanje zahtjev za godišnji odmor zaposlenika koji su trenutno na čekanju, pod uvjetom da postoji dovoljno zaposlenika na raspolaganju tijekom traženog razdoblja. Procedura prolazi kroz sve zahtjeve za godišnji odmor koji čekaju odobrenje i, na temelju određenih uvjeta, odobrava ih.

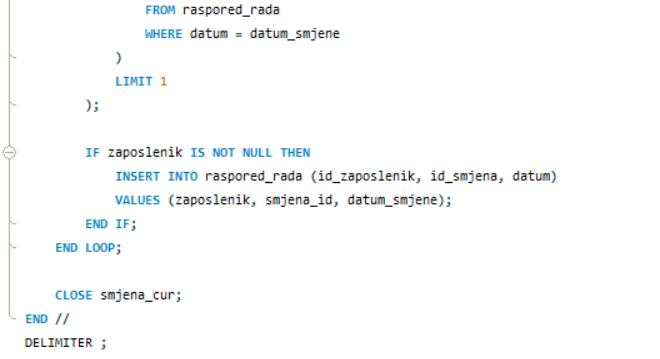
Za to koristi nekoliko varijabli: gotovo (koja označava kraj obrade podataka), unesi\_id (koji pohranjujeID zahtjeva), pocetak i kraj (koji pohranjuju datume početka i završetka godišnjeg odmora), te min\_zaposlenika (koji definira minimalan broj zaposlenika koji moraju biti prisutni na radu kako bi odobrenje bilo moguće).

Procedura koristi cursor **izlazni\_cur** koji se postavlja na sve zahtjeve za godišnji odmor koji su u statusu "na čekanju". Nakon što otvorimo cursor, ulazimo u petlju **citaj\_loop** koja će iterirati kroz sve te zahtjeve. Za svaki zahtjev provodi se provjera: prvo se kontrolira postoji li već odobren godišnji odmor koji se preklapa s trenutnim zahtjevom. Ako se nađe preklapanje, zahtjev se ne odobrava.

Zatim se provodi provjera koliko zaposlenika nije na godišnjem odmoru u tom razdoblju; broj tih zaposlenika mora biti veći ili jednak minimalnom broju min\_zaposlenika da bi se zahtjev odobrio. Ako su oba uvjeta zadovoljena, status zahtjeva mijenja se u "odobren". Kada se svi zahtjevi obrade, cursor izlazni\_cur se zatvara.

## 10.9 Procedura dodavanja zaposlenika u smjene

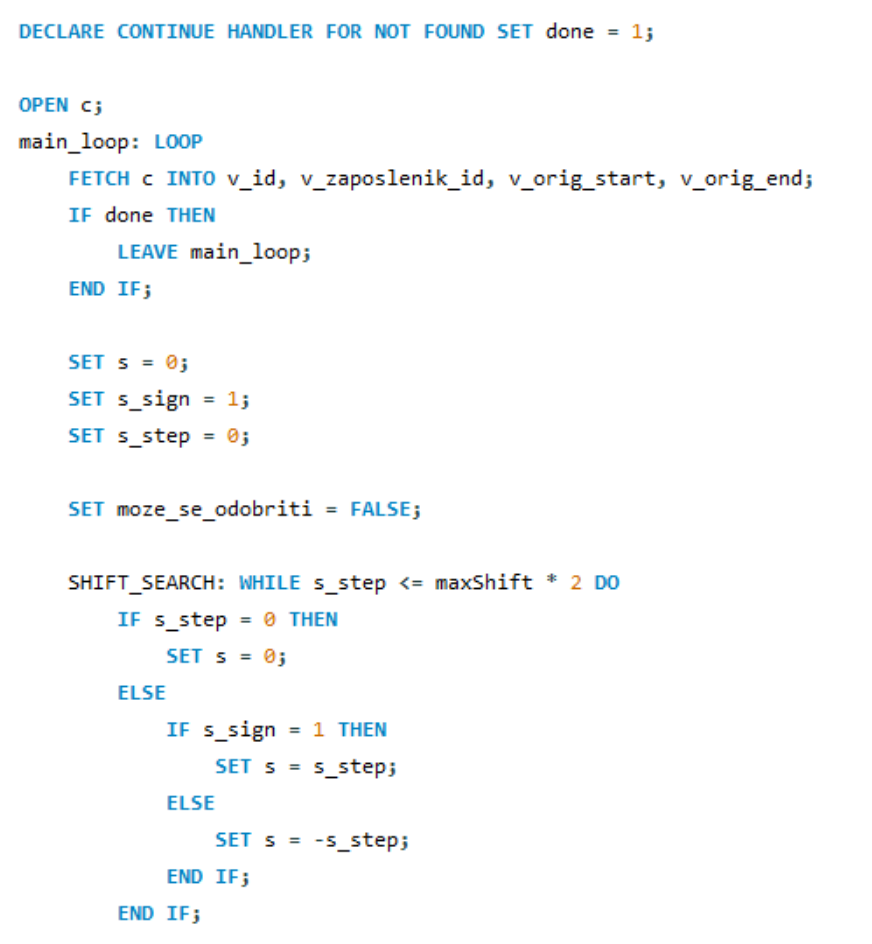
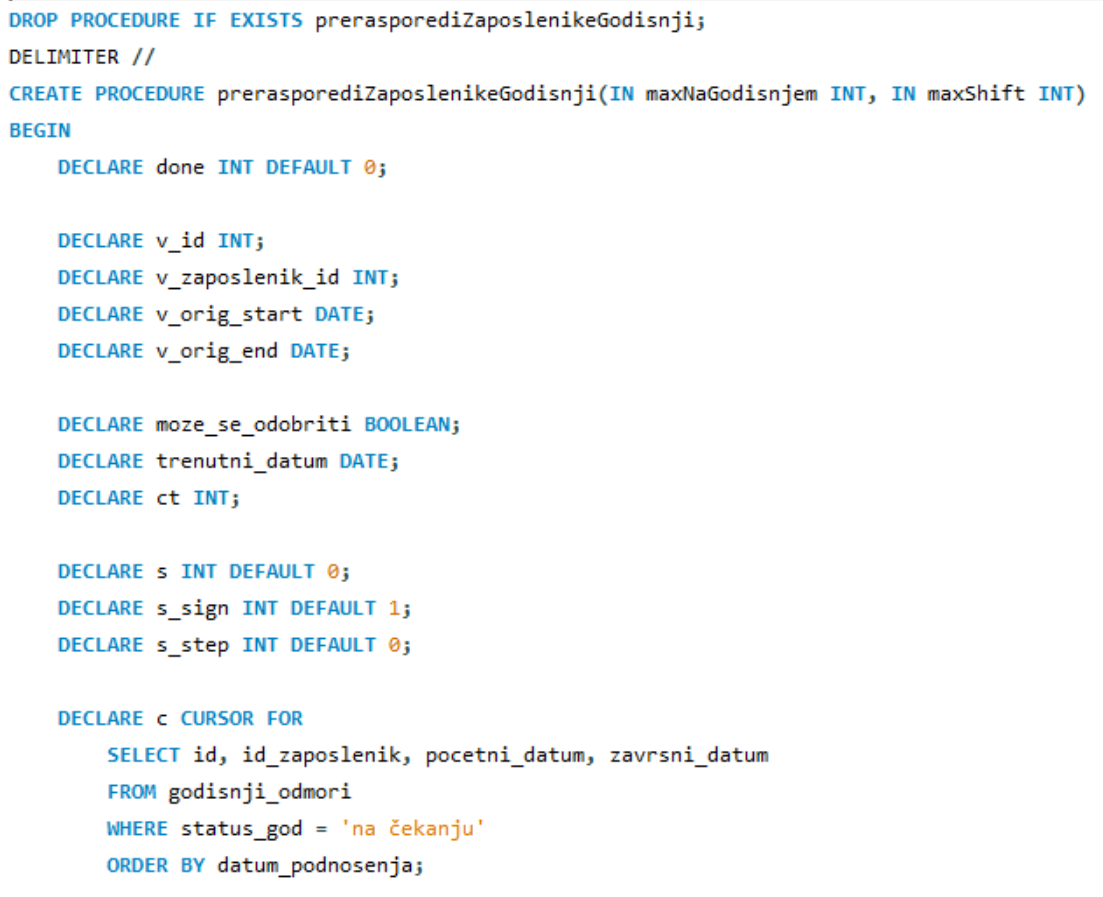


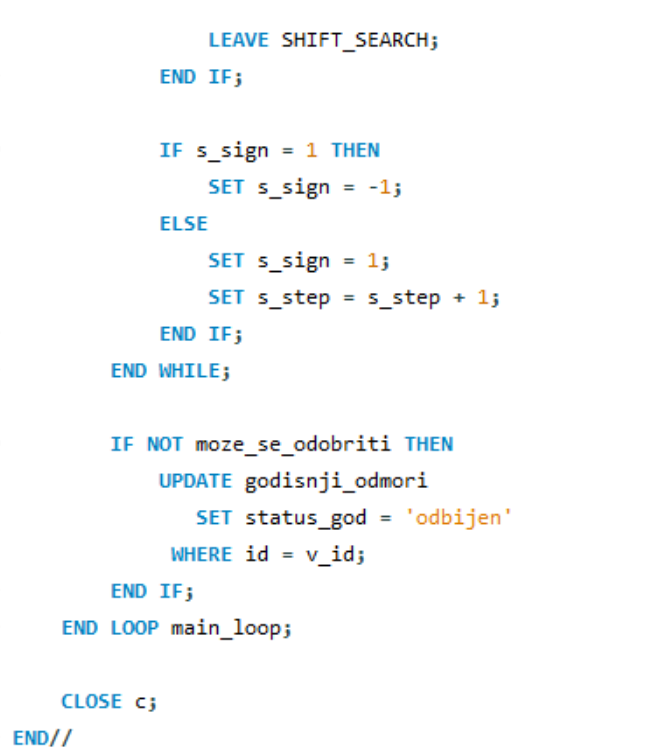
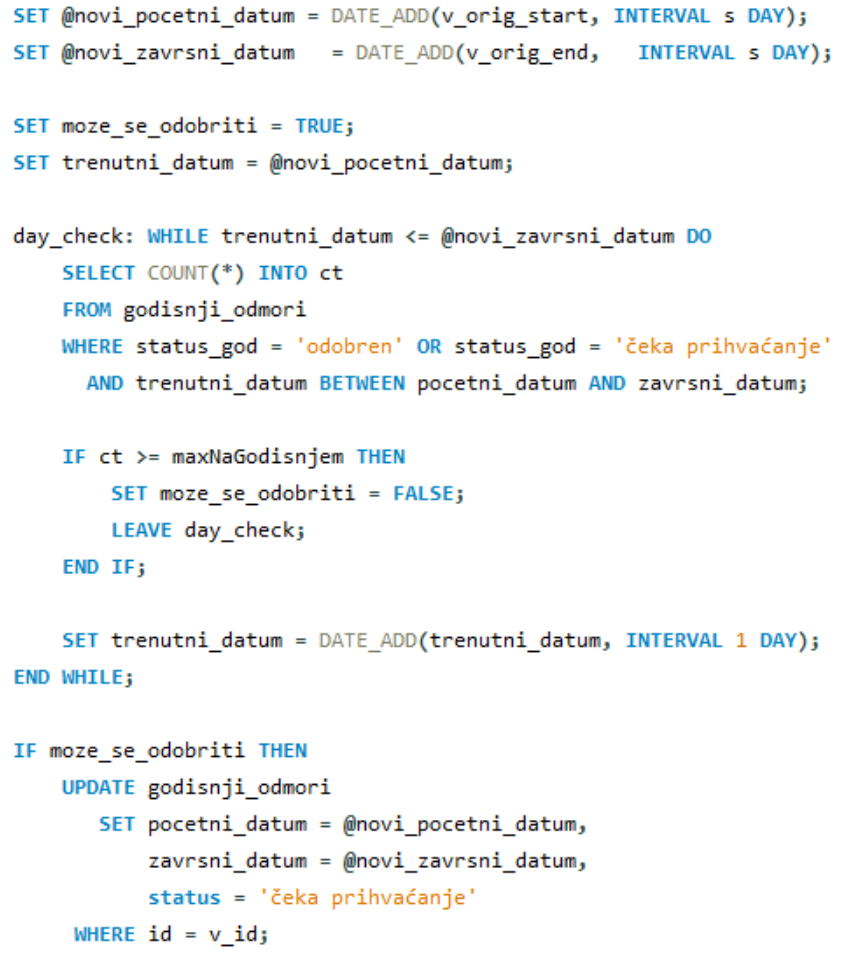


Procedura **dodajZaposlenikeUSmjene** ima za cilj dodavanje zaposlenika u smjene prema njihovim preferencijama, ali uz uvjet da se ne preklapaju s već postojećim rasporedom za taj datum. U ovoj proceduri koristi se nekoliko varijabli: gotovo (koja označava kraj obrade), zaposlenik (koja pohranjuje ID zaposlenika), smjena\_id (koja pohranjuje ID smjene), te datum\_smjene (koji označava datum smjene).

Procedura koristi cursor smjena\_cur, koji dohvaća sve smjene koje još nemaju dodijeljene zaposlenike, a to su smjene čiji ID nije prisutan u tablici raspored\_rada. Kada je cursor otvoren, ulazi se u petlju loopaj koja se izvodi dok se ne prođu svi zapisi. U svakom krugu petlje, procedura pokušava dodijeliti zaposlenika smjeni tako da prvo provjerava njegove preferencije smjena u tablici preferencije\_smjena, uz uvjet da zaposleni nije već dodijeljen za tu smjenu na istom datumu. Ako se pronađe zaposlenik koji ispunjava uvjete, tada se zaposlenik dodaje u tablicu raspored\_rada s odgovarajućim ID-em smjene i datumom. Kada se svi podaci obrade, cursor smjena\_cur se zatvara.

## 10.10 Procedura preraspodjele zaposlenika za godišnji odmor



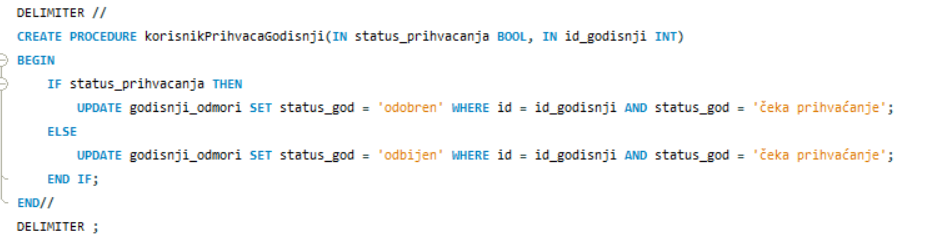


Procedura **prerasporediZaposlenikeGodisnji** osmišljena je kako bi automatski upravljala zahtjevima za godišnje odmore zaposlenika uz poštivanje ograničenja na broj zaposlenika koji mogu biti na odmoru i fleksibilno prilagođavanje termina odmora. Prima dva ulazna parametra: maxNaGodisnjem, koji definira maksimalan broj zaposlenika koji smiju biti na godišnjem odmoru u bilo kojem trenutku, i maxShift, koji određuje koliko dana se datumi odmora mogu pomaknuti kako bi se pronašao valjan raspored.

Unutar procedure se koristi CURSOR za iteraciju kroz zahtjeve za godišnji odmor koji imaju status "na čekanju", poredane prema datumu podnošenja. Za svaki zahtjev, pokušava se pronaći valjan raspored tako da se početni i završni datumi pomiču unutar zadanog raspona maxShift. Algoritam koristi logiku gdje pomiče datume naizmjenično prema naprijed i unatrag, provjeravajući pritom za svaki dan je li broj odobrenih odmora ispod limita maxNaGodisnjem. Ako je novi termin valjan, ažurira se zapis u tablici godisnji\_odmori i postavlja status "čeka prihvaćanje". Ako nije moguće pronaći valjan raspored unutar dopuštenog pomaka, zahtjev se odbija promjenom statusa na "odbijen".

Kontrola toka unutar procedure osigurava da se zahtjevi obrađuju redom, uz upotrebu petlji WHILE i LOOP, te dodatnih varijabli za izračun i provjere. Postavljen je i HANDLER za slučaj kada se obrada završi jer više nema zahtjeva za obradu. Na kraju, kursor se zatvara. Procedura je osmišljena kako bi automatizirala složen zadatak upravljanja godišnjim odmorima uz optimizaciju resursa i poštivanje pravila organizacije.

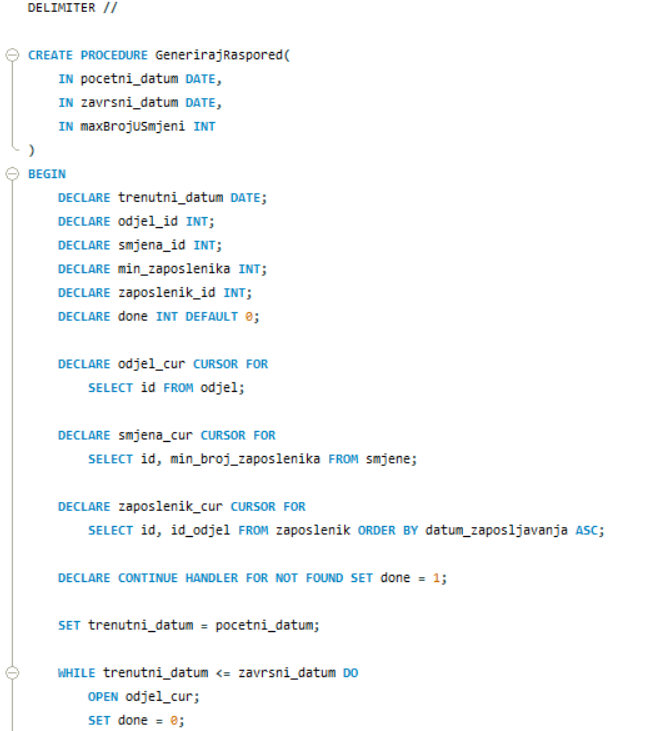
## 10.11 Procedura korisnik prihvaća godišnji odmor

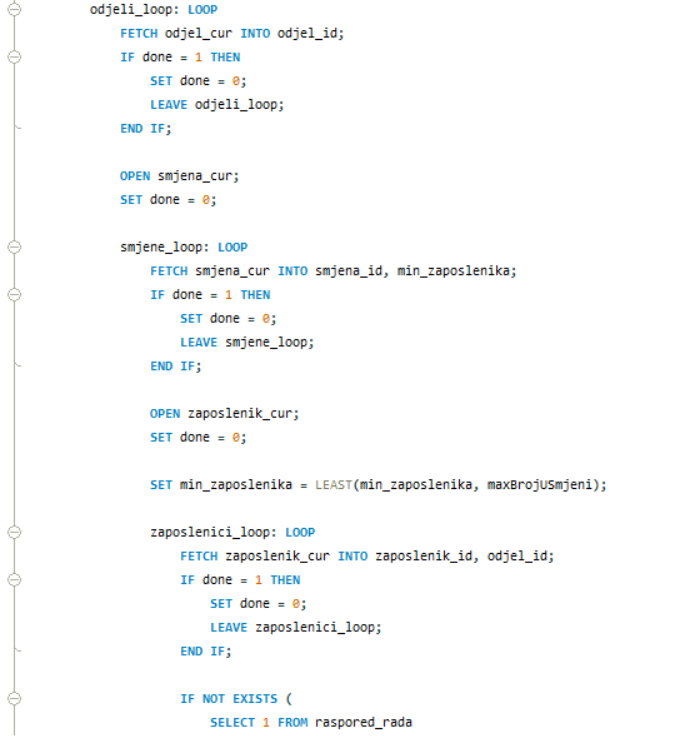


Procedura **korisnikPrihvacaGodisnji** omogućava korisniku (ili administratoru) da prihvati ili odbije zahtjev za godišnji odmor na temelju predanog statusa. Procedura prima dva ulazna parametra: status\_prihvacanja koji označava je li zahtjev prihvaćen ili odbijen (tip varijable je boolean, gdje TRUE znači prihvaćanje, a FALSE znači odbijanje), te id\_godisnji koji predstavlja ID zahtjeva za godišnji odmor u tablici godisnji\_odmori.

Kada se pozove ova procedura, izvršava se IF uvjet koji provjerava vrijednost parametra status\_prihvacanja. Ako je vrijednost TRUE (tj. korisnik prihvaća godišnji odmor), tada se ažurira status zahtjeva za godišnji odmor na 'odobren', ali samo ako je trenutni status zahtjeva 'čeka prihvaćanje'. Ako je vrijednost FALSE (korisnik odbija zahtjev), tada se status promijeni na 'odbijen', također samo ako je trenutni status 'čeka prihvaćanje'.

## 10.12 Procedura generiranja rasporeda



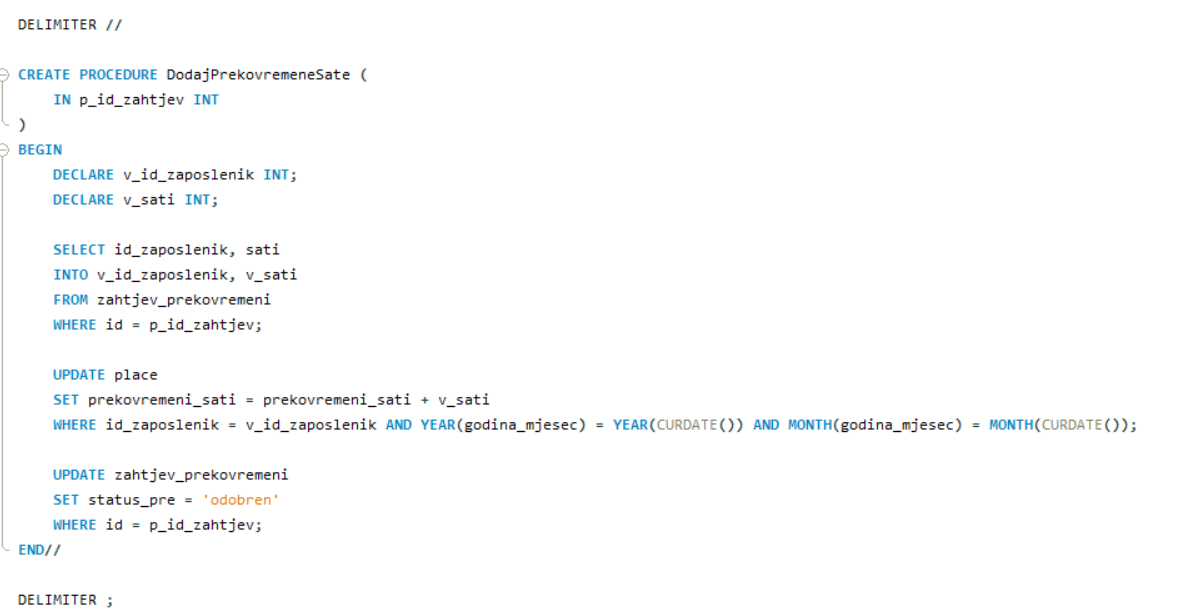




Procedura GenerirajRaspored korisiti se za automatsko generiranje rasporeda rada zaposlenika. Prima tri ulazna parametra: pocetni\_datum koji označava početak razdoblja za generiranje rasporeda, zavrsni\_datum koji označava kraj tog razdoblja i maxBrojUSmjeni koji predstavlja maksimalni broj zaposlenika dozvoljen u jednoj smjeni.

Procedura započinje deklaracijom potrebnih varijabli: trenutni\_datum za iteraciju kroz dane u zadanom razdoblju, odjel\_id za identificiranje odjela, smjena\_id za identifikaciju smjene, min\_zaposlenika za praćenje minimalnog broja potrebnih zaposlenika u smjeni, te zaposlenik\_id za identifikaciju svakog zaposlenika. Uz to, definiraju se tri kursora za iteraciju kroz odjele, smjene i zaposlenike, te handler za upravljanje situacijom kada se dosegne kraj kursora. Procedura koristi vanjsku petlju koja iterira kroz svaki dan u razdoblju od pocetni\_datum do zavrsni\_datum. Za svaki dan otvara se kursor odjel\_cur koji dohvaća popis svih odjela, a unutar te petlje otvara se kursor smjena\_cur koji dohvaća popis smjena zajedno s minimalnim brojem zaposlenika potrebnim za svaku smjenu. Maksimalni broj zaposlenika u smjeni se zatim ograničava na vrijednost manjeg između zadanog minimalnog broja i parametra maxBrojUSmjeni. Unutar petlje za smjene otvara se treći kursor zaposlenik\_cur, koji iterira kroz sve zaposlenike, sortirane prema datumu zaposlenja kako bi se prioritet dao zaposlenicima s duljim radnim stažem. Za svakog zaposlenika provjerava se je li već raspoređen na radni dan (trenutni\_datum) ili je na godišnjem odmoru u tom razdoblju. Ako zaposlenik ispunjava uvjete, dodaje se u raspored rada za određenu smjenu i dan. Nakon toga smanjuje se broj potrebnih zaposlenika za tu smjenu, a kada broj dosegne nulu, petlja za zaposlenike se prekida. Po završetku iteracije kroz zaposlenike, kursor za zaposlenike se zatvara. Isto se ponavlja za smjene, a zatim i za odjele. Kada su svi odjeli i smjene za određeni dan obrađeni, procedura prela*z*i na sljedeći datum koristeći DATE\_ADD funkciju. Na kraju, procedura osigurava da svi potrebni zaposlenici budu raspoređeni prema uvjetima smjene i dostupnosti, automatski kreira kreirajući unos u tablici raspored\_rada s informacijama o zaposleniku, smjeni i datumu.

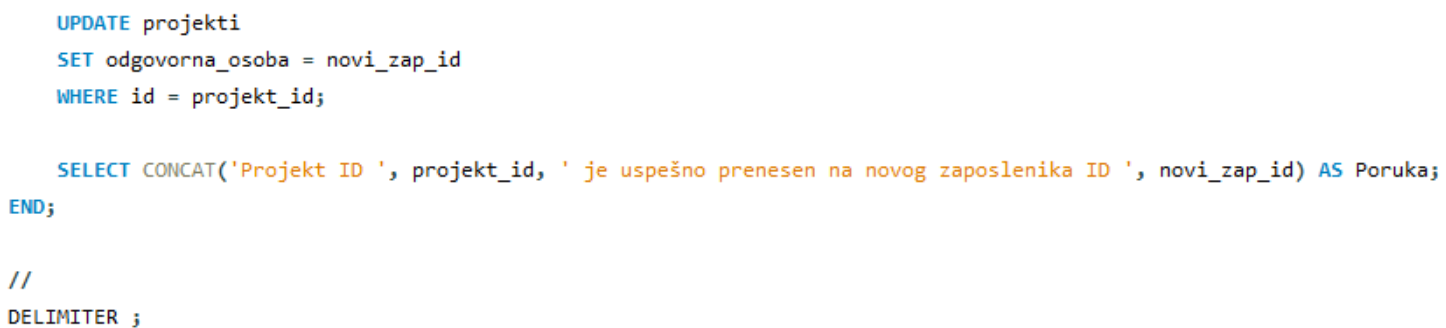
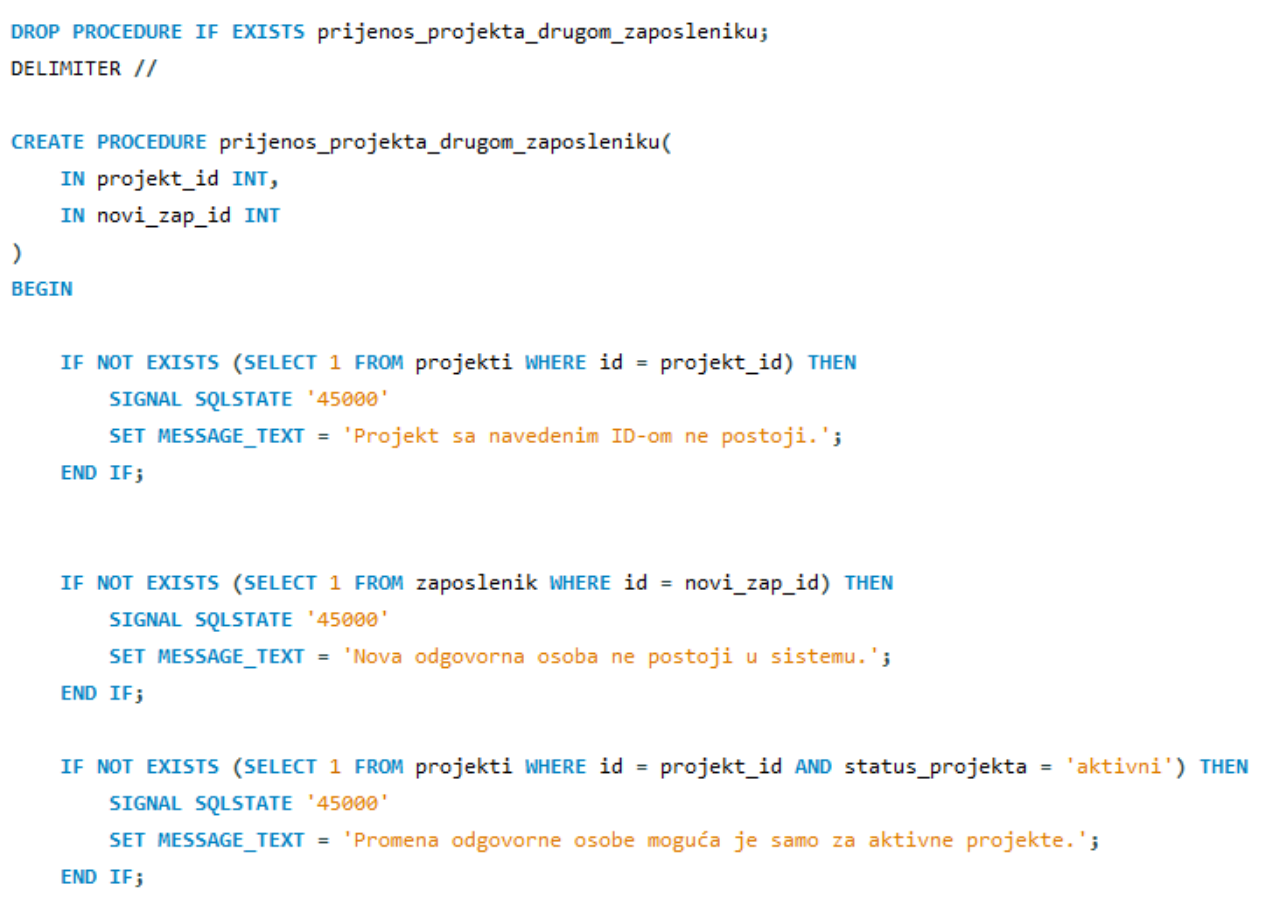
## 10.13 Procedura dodavanja prekovremenih sati



Procedura **DodajPrekovremeneSate** koristi se za odobravanje prekovremenih sati zaposlenika putem zahtjeva te potom ažuriranje u tablici plaća. Procedura prima jedan ulazni parametar p\_id\_zahtjev, koji predstavlja jedinstveni identifikator zahtjeva za prekovremenim radom.Prvo se deklariraju dvije varijable:v\_id\_zaposlenik (pohranjivanje identifikatora zaposlenika koji je podnio zahtjev) i v\_sati(broj prekovremenih sati zatraženih u zahtjevu.)Korištenjem SELECT ... INTO, procedura dohvaća vrijednosti id\_zaposlenik (ID zaposlenika) i sati (broj prekovremenih sati) iz tablice zahtjev\_prekovremeni, koristeći ulazni parametar p\_id\_zahtjev za identifikaciju specifičnog zahtjeva.Nakon toga, procedura ažurira tablicu place, dodajući broj prekovremenih sati (v\_sati) za zaposlenika v\_id\_zaposlenik. Ažuriranje se odnosi samo na zapis u tablici plaća za trenutni mjesec i godinu, što se postiže pomoću funkcija YEAR(CURDATE()) i MONTH(CURDATE()) za dohvaćanje tekuće godine i mjeseca. Vrijednost prekovremenih sati (prekovremeni\_sati) se uvećava za broj sati iz zahtjeva.

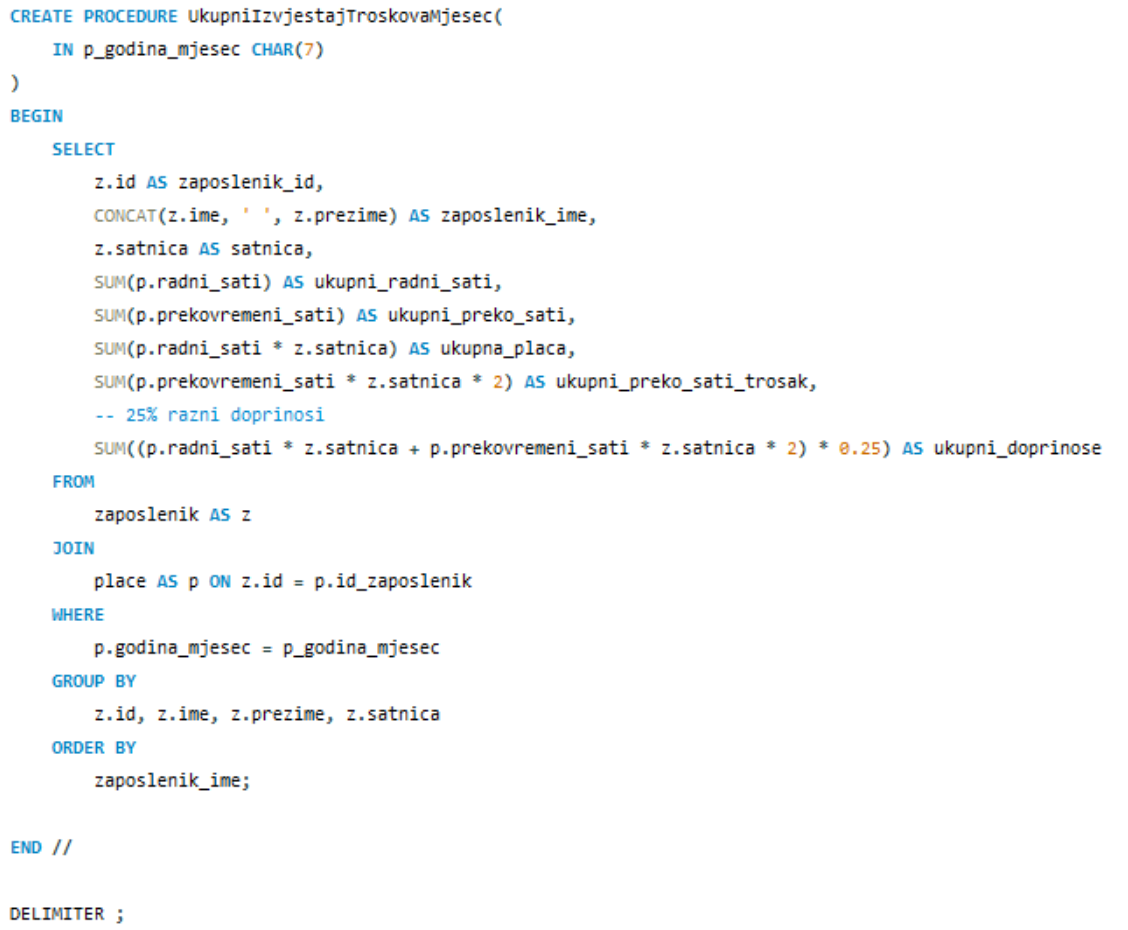
Konačno, status zahtjeva u tablici zahtjev\_prekovremeni ažurira se na 'odobren' za identificirani zahtjev (p\_id\_zahtjev).

## 10.14 Procedura prijenos projekta drugom zaposleniku



Procedura **prijenos\_projekta\_drugom\_zaposleniku** omogućuje prijenos odgovornosti za određeni projekt s trenutne odgovorne osobe na drugog zaposlenika, koristeći ID projekta i ID novog zaposlenika kao ulazne parametre. Prvo provjerava postoji li projekt s danim projekt\_id, a ako ne postoji, pokreće grešku s odgovarajućom porukom. Zatim provjerava je li novi zaposlenik s danim novi\_zap\_id prisutan u sustavu; ako nije, generira novu grešku. Sljedeća provjera osigurava da se promjena odgovorne osobe vrši samo za projekte koji imaju status 'aktivni', što sprječava promjene za projekte koji su završeni ili odgođeni. Ako sve provjere prođu, naredba UPDATE ažurira polje odgovorna\_osoba u tablici projekti tako da sadrži ID novog zaposlenika. Nakon uspješnog ažuriranja, procedura vraća poruku koja informira korisnika da je projekt uspješno prenesen na novog zaposlenika, uključujući ID-ove projekta i novog zaposlenika.

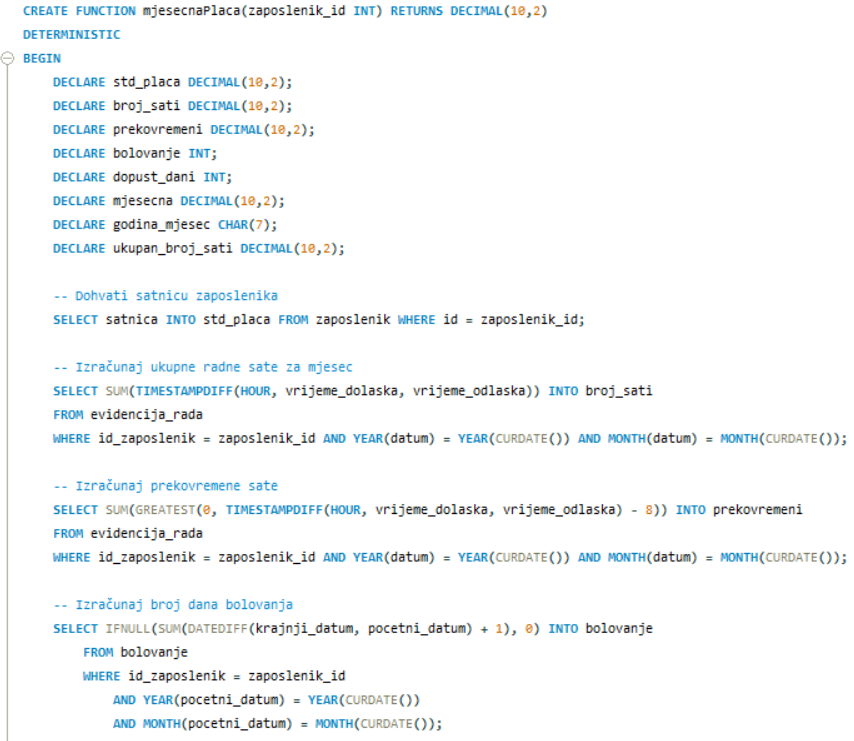
## 10.15 Procedura ukupni izvještaj troškova po mjesecu



Procedura **UkupniIzvjestajTroskovaMjesec** generira izvještaj o troškovima rada za određeni mjesec, na temelju unesenog parametra \_godina\_mjesec u formatu YYYY-MM. Izvještaj uključuje podatke o svakom zaposleniku, njegovim radnim i prekovremenim satima, ukupnoj zaradi, troškovima prekovremenih sati, te doprinosima koji se temelje na ukupnim plaćama. Unutar procedure, izvršava se SELECT upit koji spaja tablice zaposlenik i place kako bi se dohvatili podaci o zaposleniku i njegovim satima rada za navedeni mjesec. Za svakog zaposlenika računa se ukupni broj radnih i prekovremenih sati, ukupna zarada temeljena na satnici, dodatni troškovi prekovremenih sati (koji su plaćeni dvostruko), te doprinosi koji čine 25% od ukupne bruto plaće. Podaci se grupiraju prema ID-u zaposlenika i osnovnim podacima o zaposleniku (ime, prezime, satnica) kako bi se osiguralo pravilno agregiranje. Rezultati su poredani abecedno prema imenu zaposlenika kako bi izvještaj bio pregledniji. Procedura pruža detaljan pregled troškova rada za svakog zaposlenika, olakšavajući financijsku analizu i donošenje poslovnih odluka.

# 11. Funkcije

## 11.1. Funkcija računanje mjesečne plaće





Funkcija **mjesecnaPlaca** izračunava mjesečnu plaću zaposlenika na temelju radnih sati, prekovremenih sati, bolovanja i osnovne satnice zaposlenika. Funkcija prima jedan ulazni parametar zaposlenik\_id, koji označava ID zaposlenika čiju plaću treba izračunati, i vraća iznos mjesečne plaće tipa DECIMAL(10,2).

U prvom koraku funkcija dohvaća osnovnu satnicu zaposlenika iz tablice zaposlenik koristeći njegov ID. Ova vrijednost se pohranjuje u varijablu std\_placa. Zatim, funkcija izračunava ukupne radne sate zaposlenika za trenutni mjesec. Za to koristi funkciju TIMESTAMPDIFF, koja računa razliku u satima između vremena dolaska i vremena odlaska zaposlenika, te zbraja sve radne sate za tog zaposlenika, pohranjujući ih u varijablu broj\_sati.

Nakon toga, funkcija izračunava prekovremene sate te i bolovanja. Na temelju prethodnih varijabli, funkcija izračunava ukupnu mjesečnu plaću prema formuli:

mjesecna = (broj\_sati \* std\_placa) +

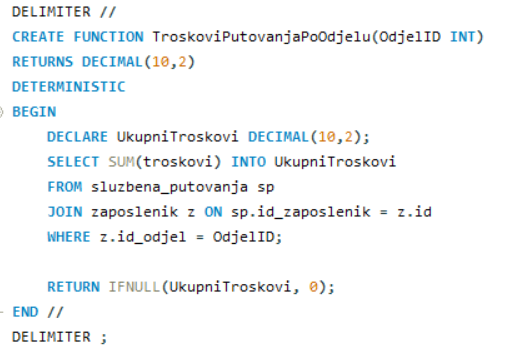
(prekovremeni \* std\_placa \* 2) -

(bolovanje \* 3 \* std\_placa).

Ova formula uključuje osnovnu plaću za odrađene sate, dodatak za prekovremene sate (koji je dvostruko veći od osnovne satnice), te odbitak za dane bolovanja (koji se množe s tri osnovne satnice po danu bolovanja).

Izračunata plaća zatim se pohranjuje u tablicu place zajedno s informacijama o radnim satima, prekovremenim satima, bolovanju i ukupnoj plaći za tog zaposlenika za trenutni mjesec. Ako već postoji zapis za tog zaposlenika u tablici za taj mjesec, koristi se naredba ON DUPLICATE KEY UPDATE za ažuriranje postojećeg zapisa s novim iznosom plaće.

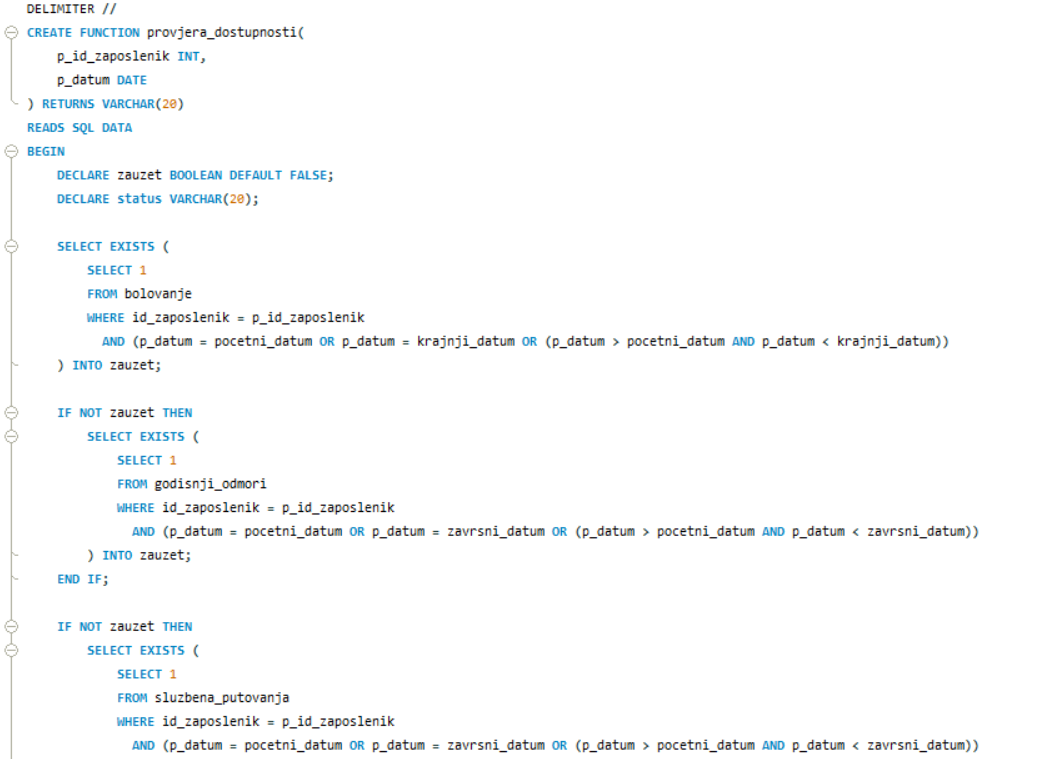
## 11.2. Funkcija kalkuliranja troškova službenih putovanja po odjelu



Funkcija **TroskoviPutovanjaPoOdjelu** izračuna ukupne troškove službenih putovanja za određeni odjel.Fznkcija prima ulazni parametar OdjelID, koji predstavlja ID odjela koji se želi izračunati ukupni troškovi putovanja.Rezultat funkcije je prikazan tipa DECIMAL(10,2).

Unutar funkcije je deklarirana(DECLARE) je varijabla UkupniTroskovi, koja će pohraniti izračunate troškove. Funkcija koristi SELECT ... INTO naredbu za dohvaćanje i zbrajanje svih troškova iz tablice sluzbena\_putovanja. Uvjet WHERE z.id\_odjel = OdjelID osigurava da se zbrajaju samo troškovi službenih putovanja zaposlenika iz navedenog odjela. Funkcija zatim koristi SUM(troskovi) za izračun ukupnog iznosa troškova putovanja. Ako nema podataka o službenim putovanjima za taj odjel (tj. ako rezultat zbrajanja bude NULL), funkcija koristi IFNULL kako bi vratila vrijednost 0, čime se izbjegava vraćanje NULL kao rezultata.

## 11.3. Funkcija provjera dostupnosti zaposlenika



Funkcija **provjera\_dostupnosti** koristi ze za provjeru dostupnosti određenog zaposlenika za određeni datum i vraća rezultat u obliku teksta ‘Dostupan’ ili ‘Nedostupan’.FUnkcija ima dva ulazna parametra:p\_id\_zaposlenik, koji predstavlja ID zaposlenika, te p\_datum,koji predstavlja datum koji korinik provjerava dostupnost zaposlenika.U funkciji su definirane dvije pomoćne varijable: zauzet(tip:BOOLEAN) i pocetno postavljanje FALSE, te status(tipa:VARCHAR(20)) koji služi za spremanje konačnog rezultata funkcije.

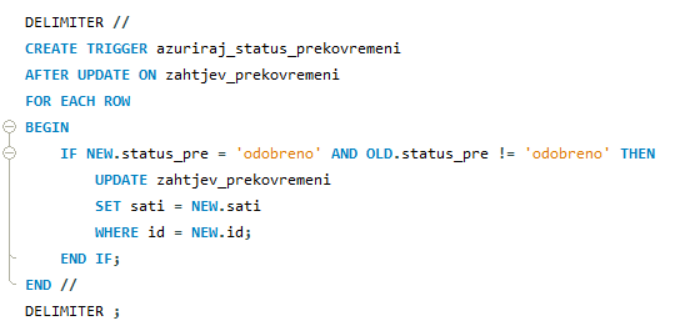
Sljedeći korak je provjera dali zaposlenik je na bolovanju, godišnjem odmoru, službeno putovanju ili dopustu. Funkcija koristi naredbu SELECT EXISTS kako bi provjerila postoji li zapis u svim tablicama za navedenog zaposlenika i datuma. U provjeri se uzima u obzir je li p.datum jednak početnom datumu, krajnjem datumu ili unutar raspona između početnog i krajnje datuma. Ako se pronađe odgovarajući zapis, varijabla zauzet postavlja se na TRUE.

U provijeri tablice dopust vrijedi li isti uvjet poklapanja datuma kao u pethodnim slučajevima,ali s dodatnim uvjetom da je status dopusta jednak ‘odobren’.

Na kraju, nakon svih provjera, funkcija određuje vrijednost varijable status. Ako je varijabla zauzet postavljena na TRUE, tada se u status sprema vrijednost 'Nedostupan'. Ako je zauzet ostao FALSE, vrijednost varijable status postavlja se na 'Dostupan'. Funkcija zatim vraća vrijednost varijable status kao rezultat.

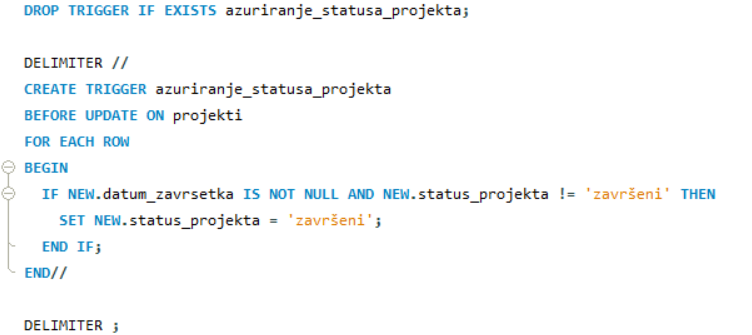
# 12.Okidači

## 12.1. Okidač za ažuriranje statusa prekovremenog



Okidač **azuriraj\_status\_prekovremeni** autmomatski se aktivira nakon svake UPDATE operacije na tablici zahtjev\_prekovremeni. Unutar okidača provodi se provjera koja ispituje da li je novi status zahtjeva (NEW.status\_pre) postao 'odobreno' i je li prethodni status (OLD.status\_pre) bio različit od 'odobreno'. Ako su oba uvjeta zadovoljena, okidač izvršava naredbu koja ažurira broj sati u zahtjevu. Konkretno, vrijednost stupca sati za odabrani zapis (određen prema njegovom id polju) postavlja se na novu vrijednost iz NEW.sati.

## 12.2. Okidač za ažuriranje statusa projekta



Okidač **azuriranje\_statusa\_projekta** automatski se aktivira prije svake UPDATE operacije na tablici projekti. Prilikom ažuriranja bilo kojeg zapisa u tablici projekti, okidač provjerava dva uvjeta. Prvi uvjet je da polje datum\_zavrsetka novog zapisa (NEW.datum\_zavrsetka) nije NULL, što znači da je projekt dobio završni datum. Drugi uvjet je da trenutni status projekta (NEW.status\_projekta) već nije 'završeni'. Ako su oba uvjeta zadovoljena, okidač automatski postavlja vrijednost NEW.status\_projekta na 'završeni'.

# 13.Inicijalizacija i konfiguracija

**Flask(\_\_name\_\_)**: Kreira instancu Flask aplikacije, što omogućuje definiciju ruta i povezivanje s predlošcima te statičkim datotekama.

**app.config**: Postavlja konfiguraciju za povezivanje s MySQL bazom podataka koristeći SQLAlchemy. Povezivanje uključuje URL baze podataka (mysql+pymysql), korisničko ime (root), lozinku (luka23012005), host (127.0.0.1), port (3306) i naziv baze podataka (bp\_2\_projekt).

**SQLAlchemy**: Inicijalizira se objekt db, koji omogućuje rad s bazom podataka na praktičan način, koristeći ORM ili ručne SQL upite.

# 14.Rute i funkcionalnosti

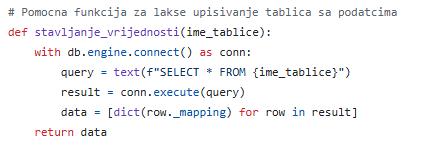
Glavna ruta (/) Ruta / prikazuje početnu stranicu aplikacije (index.html) koristeći render\_template.

Rute za dohvaćanje podataka iz baze

Korištenjem pomoćne funkcije stavljanje\_vrijednosti, implementirane su rute koje dohvaćaju podatke iz tablica baze podataka i vraćaju ih u JSON formatu.



Funkcija stavljanje\_vrijednosti omogućuje univerzalno dohvaćanje podataka iz bilo koje tablice. Izvršava SQL upit SELECT \* FROM ime\_tablice, dohvaća rezultate, te ih pretvara u JSON format koristeći dict(row.\_mapping).



Pomocna funkcija stavljanje\_vrijednosti omogućava dohvaćanje podataka iz tablice u bazi podataka na strukturiran način. Kao argument prima naziv tablice, a zatim otvara konekciju prema bazi koristeći SQLAlchemy mehanizam. Unutar konekcije generira SQL upit za dohvaćanje svih redaka iz navedene tablice (SELECT \* FROM ime\_tablice) i izvršava ga. Dobiveni rezultati pretvaraju se u popis rječnika, gdje svaki rječnik predstavlja jedan red tablice, pri čemu se nazivi stupaca koriste kao ključevi.