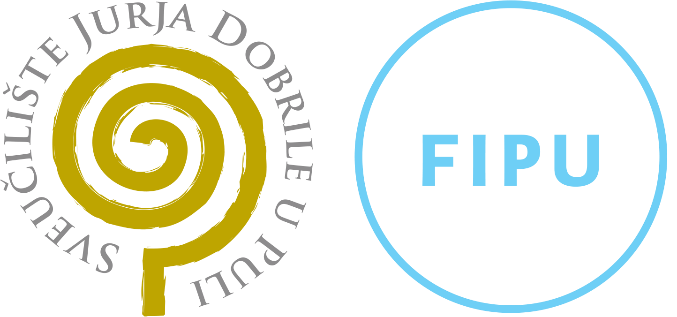
**Sveučilište Jurja Dobrile u Puli**

**Fakultet informatike u Puli**



Dokumentacija uz projektni zadatak

„**Tvrtka za najam automobila**“

**Tim 2**

* Marinela Blažek
* Mario Blažeka
* Sebastijan Dominis
* Vedrana Plenča
* Mirela Ražnjević Moravčić

**Smjer:**  Diplomski sveučilišni studij Informatika

**Kolegij:** Baze podataka I

**Mentor:** doc. dr. sc. Goran Oreški

Pula, svibanj, 2024. godine

**SADRŽAJ:**

[1. Uvod 4](#_Toc168053658)

[2. Opis poslovnog procesa 4](#_Toc168053659)

[3. Entity Relationship (ER) dijagram 7](#_Toc168053660)

[4. Veze entiteta prema ER dijagramu 7](#_Toc168053661)

[5. Relacijski model (scheme) 10](#_Toc168053662)

[6. EER dijagram (MySQL) 12](#_Toc168053663)

[7. Tablice 12](#_Toc168053664)

[7.1. TABLICA lokacija 12](#_Toc168053665)

[7.2. TABLICA pravna\_osoba 13](#_Toc168053666)

[7.3. TABLICA klijent 13](#_Toc168053667)

[7.4. TABLICA zanimanje 14](#_Toc168053668)

[7.5. TABLICA zaposlenik 14](#_Toc168053669)

[7.6. TABLICA kontakt\_klijenta 15](#_Toc168053670)

[7.7. TABLICA prihod 16](#_Toc168053671)

[7.8. TABLICA transakcija 16](#_Toc168053672)

[7.9. TABLICA prihod\_za\_zaposlenika 17](#_Toc168053673)

[7.10. TABLICA popust 18](#_Toc168053674)

[7.11. TABLICA popust\_za\_klijenta 18](#_Toc168053675)

[7.12. TABLICA poslovni\_trosak 19](#_Toc168053676)

[7.13. TABLICA gotovinsko\_placanje 19](#_Toc168053677)

[7.14. TABLICA karticno\_placanje 20](#_Toc168053678)

[7.15. TABLICA kriptovalutno\_placanje 21](#_Toc168053679)

[7.16. TABLICA kontakt\_pravne\_osobe 21](#_Toc168053680)

[7.17. TABLICA serija 22](#_Toc168053681)

[7.18. TABLICA vozilo 23](#_Toc168053682)

[7.19. TABLICA najam\_vozila 23](#_Toc168053683)

[7.20. TABLICA slika\_vozila 24](#_Toc168053684)

[7.21. TABLICA tip\_osiguranja 25](#_Toc168053685)

[7.22. TABLICA osiguranje 25](#_Toc168053686)

[7.23. TABLICA steta 26](#_Toc168053687)

[7.24. TABLICA naknada\_stete 27](#_Toc168053688)

[7.25. TABLICA punjenje 28](#_Toc168053689)

[7.26. TABLICA odrzavanje 28](#_Toc168053690)

[7.27. TABLICA rezervacija 29](#_Toc168053691)

[7.28. TABLICA oprema 29](#_Toc168053692)

[7.29. TABLICA oprema\_na\_najmu 30](#_Toc168053693)

[7.30. TABLICA oprema\_na\_rezervaciji 30](#_Toc168053694)

[7.31. TABLICA vozilo\_na\_rezervaciji 31](#_Toc168053695)

[7.32. TABLICA crna\_lista 31](#_Toc168053696)

[8. UPITI 32](#_Toc168053697)

[8.1. Upiti – Marinela 32](#_Toc168053698)

[8.2. Upiti – Mirela 35](#_Toc168053699)

[8.3. Upiti – Sebastijan 41](#_Toc168053700)

[8.4. Upiti – Vedrana 47](#_Toc168053701)

[9. Korišteni alati i metode 59](#_Toc168053702)

[10. ZAKLJUČAK 59](#_Toc168053703)

# Uvod

Ovaj je projekt započeo kao ideja o stvaranju baze podataka pomoću SQL-a za malu kompaniju koja se bavi najmom automobila, no s vremenom se proširio te sada predstavlja bazu međunarodne kompanije za najam vozila. Nadamo se da smo u pokušaju stvaranja te sveobuhvatne baze i uspjeli. Za komunikaciju pri radu na ovome projektu najviše smo koristili WhatsApp i Zoom, a ubrzo smo zaključili kako bi nam korištenje GitHuba bilo korisno pa smo i to implementirali u naš rad. Projekt je rađen u MySql Workbenchu, tj. tamo smo stvarali našu bazu podataka, unosili podatke za istu i napravili EER dijagram. Što se tiče samih podataka, oni su stvoreni u skladu sa potrebama naše firme za najam vozila koja posluje na prostoru srednje i jugoistočne Europe. Svi su podaci izmišljeni, no na način da djeluju stvarno. Količina podataka nije stvarna, odnosno značajno je manja u odnosu na količinu koju bi ova baza imala kada bi ju neka kompanija stvarno koristila. Razlog tomu je sama potreba i fokus našega projekta. Fokusirali smo se na kvalitetu modela, tj. relacija i njihovih atributa, kao i na stvaranje upita koji testiraju kvalitetu same baze te samim time broj n-torki nije igrao presudnu ulogu. Za pomoć pri generiranju nekih podataka koristili smo C++ i ChatGPT, kao i standardno pretraživanje interneta. Naša baza obuhvaća podatke o zaposlenicima, klijentima, vozilima, najmu, i svemu što je povezano sa navedenim. Namjera nam je bila da baza sadrži model relacija i atributa za sve podatke koji bi našoj izmišljenoj kompaniji mogli biti potrebni, a da pritom ne stvaramo nepotrebne relacije koje bi gušile bazu, odnosno da ona bude razumna i normalizirana u skladu sa postojećim standardima. Sljedeća će poglavlja detaljnije opisati sam poslovni proces naše fiktivne firme, predstaviti i opisati naš ER dijagram, shemu relacijskog modela i EER dijagram. Potom ćemo prikazati naš kod za pokretanje SQL skripti te opisati relacije naše baze podataka. Opisati ćemo i način na koji smo generirali podatke, a zatim ćemo ispisati i opisati upite kojima smo testirali kvalitetu našega modela, kao i zaključak ovoga projekta.

# Opis poslovnog procesa

Poslovni proces koji smo u sklopu ovoga projekta modelirali svakako nije jednostavan. Ipak, do koje je on mjere složen ostavljamo na interpretaciju čitatelja. Ponavljamo kako je cilj bio stvaranje baze podataka koja obuhvaća sve može biti važne informacije za našu firmu, a da pritom nije natrpana nevažnima te sa time na umu nastavljamo sa opisom.

Naša firma za najam vozila bavi se najmom automobila, motocikala i kamiona, a pritom po potrebi iznajmljuje i dodatnu opremu. Budući da vozila mogu unajmiti fizičke i pravne osobe, stvorili smo po jednu relaciju za obje. One su i povezane na način da se pravna osoba nalazi kao strani ključ u relaciji klijent, a razlog tomu je što fizička osoba iznajmljuje vozilo u ime pravne osobe i za to dobiva R1 račun. Također, zanimaju nas kontakti i za fizičke i za pravne osobe te je stoga stvorena relacija kontakt, koja ovisno o tome kojom se relacijom dolazi do nje daje email, telefonski broj i mobilni broj fizičke ili pravne osobe.

Što se tiče naših zaposlenika, zanima nas više informacija od onih koje tražimo od klijenata, poput informacije o tome tko je nekom zaposleniku nadređen. Zanima nas i njihovo zanimanje, a posto više zaposlenika može imati isto zanimanje (npr. vozač kamiona, računovođa, zaštitar…), za tu je potrebu stvorena i nova relacija koja opisuje ta zanimanja. Budući da poslujemo međunarodno, imamo i podružnice u drugim gradovima i državama, a u Zagrebu imamo dva mjesta poslovanja, stvorena je relacija grad koja nam pokazuje lokaciju našeg središta ili podružnice te je povezana sa zaposlenikom, kako bismo znali gdje on/ona radi. Naši zaposlenici ne primaju samo redovne plaće, nego i razne bonuse i nagrade. Relacija prihod nam detaljno pokazuje isplatu plaća i svih ostalih prihoda koje naši zaposlenici primaju.

Budući da smo velika tvrtka, kod nas se događa mnogo raznoraznih transakcija. Stoga smo stvorili novu relaciju koja nam daje osnovne informacije o bilo kakvom prijenosu novca. Budući da se najam naših vozila može platiti na tri načina, stvorene su tri relacije koje to opisuju te se usporedbom id-eva može doći do informacije o načinu plaćanja nekog klijenta. Omogućujemo gotovinsko, kartično i kriptovalutno plaćanje, a unosimo i informacije o tome kojoj banci pripada kartica klijenta koji je istom platio najam te na ovaj način možemo doći i do kontaktnih informacija banke, budući da se informacije o istoj nalaze u već postojećoj relaciji pravna osoba.

Zaposlenik klijentu putem transakcije daje vozilo na najam. Vozilo ima osiguranje, a može se raditi o automobilu, motociklu ili kamionu. Možemo imati više 'istih' vozila, primjerice VW Golfa 7, te samim time podatke o seriji automobila, kamiona i motocikala pišemo u posebnim relacijama. Željeni podaci o automobilima i kamionima su jednaki te se samim time nalaze u istoj relaciji, budući da nastojimo normalizirati našu bazu podataka. U nekim našim situacijama mogu zanimati i slike naših vozila te smo za tu potrebu stvorili nove relacije, no u većini slučajeva naše slike neće zanimati, i to je razlog iz kojeg se one nalaze u posebnim relacijama.

Redovito plaćamo osiguranja za naša vozila, što brojimo kao transakcije, a plaćamo ih osiguravajućim kućama, odnosno pravnim osobama. Naša se vozila mogu oštetiti, a podatke o svim nesrećama i kvarovima pohranjujemo u relaciju šteta. Prilikom naknade štete, popravke plaćamo uz pomoć osiguranja te i na taj proces gledamo kao na transakciju. Popravke naših vozila plaćamo kasko osiguranjem, dok popravke ostalih vozila koja se oštete prilikom nesreće plaćamo obaveznim osiguranjem. Osim ovih oblika trošenja na naša postojeća vozila, ista održavamo mijenjanjem guma, čišćenjem i dolijevanjem ulja i vode. Sve to obavljaju zaposlenici te se bilježi kao održavanje koje je zahtijevalo određenu transakciju. Naposljetku, naša vozila zahtijevaju punjenje, a može se raditi o punjenju benzina, dizela ili električne energije, budući da posjedujemo razne vrste vozila.

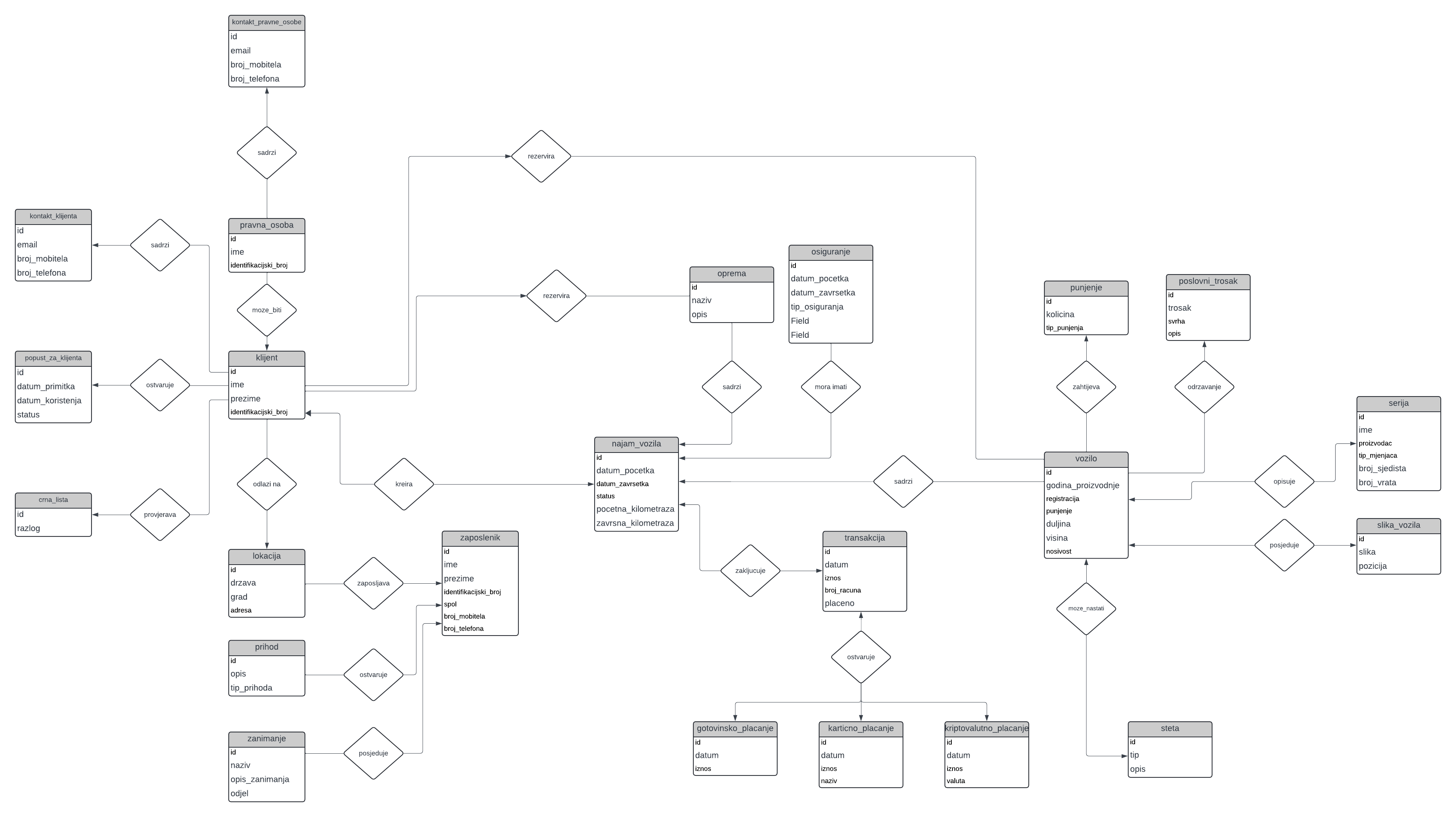
Osim svih navedenih troškova, naša firma ulaže u marketinške kampanje i kupuje nova vozila i opremu. Sve se to bilježi kao poslovni trošak za koji je bila potrebna transakcija.

Naši klijenti mogu ostvariti razne popuste, odnosno one koje stječu putem marketinških kampanja i one koje stječu zbog lojalnosti. Zbog postojanja više vrsta popusta, oni su upisani u jednu relaciju te je stvorena nova relacija, popust za klijenta, koja nam govori o tome na kojega se klijenta neki popust odnosi, je li ga klijent iskoristio, itd.

Nadalje, klijenti imaju mogućnost rezervacije vozila i opreme koju će koristiti pri vožnji unajmljenog vozila. Budući da klijent može, ali ne mora, iznajmiti dodatnu opremu, stvorene su nove relacije oprema na rezervaciji i vozilo na rezervaciji. To je učinjeno i zbog činjenice da klijent može unajmiti više vozila i više komada opreme. Budući da klijent na najmu može koristiti veću količinu istoga tipa opreme (npr. sjedalica za bebe), stvorena je i relacija oprema na najmu, dok relacija oprema opisuje svaku vrstu opreme koju nudimo.

# Entity Relationship (ER) dijagram

**Slika 1:** ER dijagram



*Izrada autora projekta*

# Veze entiteta prema ER dijagramu

|  |  |  |
| --- | --- | --- |
| Veza dvaju entiteta | Kardinalnost | Kratki opis |
| pravna\_osoba - klijent | One-to-One | Jedan klijent može biti pravna osoba. Jedna pravna osoba može biti jedan klijent. |
| zaposlenik - zanimanje | One-to-Many | Jedan zaposlenik obavlja jedno zanimanje. Jedno zanimanje može obavljati više zaposlenika. |
| klijent – kontakt\_klijenta | One-to-Many | Jedan klijent može imati više kontakata. Jedan kontakt pripada jednom klijentu. |
| zaposlenik – prihod | Many-to-Many | Jedan zaposlenik prima više tipova prihoda. Jedan tip prihoda može primiti više zaposlenika. |
| zaposlenik - lokacija | One-to-Many | Jedan zaposlenik radi na jednoj lokaciji. Na jednoj lokaciji radi više zaposlenika. |
| transakcija – klijent | One-to-Many | Jedan klijent obavlja više transakcija. Jednu transakciju obavlja jedan klijent. |
| transakcija – zaposlenik | One-to-Many | Jedan zaposlenik obavlja više transakcija. Jednu transakciju obavlja jedan zaposlenik. |
| prihod\_za\_zaposlenika - transakcija | One-to-One | Jedan prihod za zaposlenika je jedna transakcija. Jedna transakcija predstavlja jedan prihod za zaposlenika. |
| popust - klijent | Many-to-Many | Jedan tip popusta može primiti više klijenata. Jedan klijent može primiti više tipova popusta. |
| poslovni\_trosak - transakcija | One-to-One | Jedan poslovni trošak predstavlja jednu transakciju. Jedna transakcija predstavlja jedan poslovni trošak. |
| transakcija - gotovinsko\_placanje | One-to-One | Jedno gotovinsko plaćanje je jedna transakcija. Jedna transakcija predstavlja jedno gotovinsko plaćanje. |
| transakcija – karticno\_placanje | One-to-One | Jedno kartično plaćanje je jedna transakcija. Jedna transakcija predstavlja jedno kartično plaćanje. |
| transakcija – kriptovalutno\_placanje | One-to-One | Jedno kriptovalutno plaćanje je jedna transakcija. Jedna transakcija predstavlja jedno kriptovalutno plaćanje. |
| karticno\_placanje – pravna\_osoba | One-to-Many | Jedno kartično plaćanje provodi jedna banka. Jedna banka provodi više kartičnih plaćanja. |
| pravna\_osoba – kontakt\_pravne\_osobe | One-to-Many | Jedna pravna osoba ima više kontakata. Jedan kontakt pripada jednoj pravnoj osobi. |
| najam\_vozila - transakcija | One-to-Many | Jedan najam vozila pripada jednoj transakciji. Jedna transakcija može obuhvaćati više najmova vozila. |
| vozilo – najam\_vozila | One-to-Many | Jedno se vozilo više puta iznajmljuje. Na jednom se najmu vozila nalazi jedno vozilo. |
| vozilo - serija | One-to-Many | U jednoj se seriji proizvodi vise vozila. Jedno vozilo pripada jednoj seriji. |
| vozilo – slika\_vozila | One-to-Many | Jedno se vozilo moze vise puta slikati. Jedna slika pripada jednom vozilu. |
| tip\_osiguranja –pravna\_osoba | One-to-Many | Jedna osiguravajuća kuća (pravna osoba) omogućuje više tipova osiguranja. Jedno osiguranje pripada jednoj osiguravajućoj kući (pravnoj osobi). |
| osiguranje - vozilo | One-to-Many | Jedno vozilo ima više osiguranja. Jedno osiguranje osigurava jedno vozilo. |
| osiguranje - transakcija | One-to-One | Jedno osiguranje je jedna transakcija. Jedna transakcija predstavlja jedno osiguranje. |
| osiguranje – tip\_osiguranja | One-to-Many | Jedan se tip osiguranja koristi za više osiguranja. Jedno je osiguranje jednoga tipa osiguranja. |
| naknada\_stete - transakcija | One-to-One | Jedna je naknada štete jedna transakcija. Jedna transakcija predstavlja jednu naknadu štete. |
| naknada\_stete - steta | One-to-Many | Jedna šteta može zahtijevati više naknada štete. Jedna naknada štete naknađuje jednu štetu. |
| steta - osiguranje | Many-to-Many | Jednim se osiguranjem može platiti više šteta. Jedna šteta može zahtijevati naknadu iz više osiguranja. |
| punjenje - transakcija | One-to-One | Jedno je punjenje jedna transakcija. Jedna transakcija predstavlja jedno punjenje. |
| punjenje - vozilo | One-to-Many | Jedno se vozilo može više puta puniti. Jednim se punjenjem puni jedno vozilo. |
| odrzavanje - transakcija | One-to-One | Jedno je održavanje jedna transakcija. Jedna transakcija predstavlja jedno održavanje. |
| rezervacija - klijent | One-to-Many | Jedan klijent može obaviti više rezervacija. Jednu rezervaciju obavlja jedan klijent. |
| oprema – najam\_vozila | Many-to-Many | Pri jednom se najmu vozila može iznajmiti više opreme. Jedna sa oprema može iznajmiti više puta. |
| oprema - rezervacija | Many-to-Many | Pri jednoj se rezervaciji može rezervirati više opreme. Jedna se oprema može rezervirati više puta. |
| vozilo - rezervacija | Many-to-Many | Jedno se vozilo može rezervirati više puta. Pri jednoj se rezervaciji može rezervirati više vozila. |
| crna\_lista - klijent | One-to-Many | Jedan se klijent može nalaziti na jednoj crnoj listi. Na jednoj se crnoj listi može nalaziti više klijenata. |

# Relacijski model (scheme)

**lokacija (**id, drzava, grad, adresa)

**pravna\_osoba** (id, ime, identifikacijski\_broj, id\_lokacija)

**klijent** (id, ime, prezime, identifikacijski\_broj, id\_pravna\_osoba)

**zanimanje** (id, naziv, opis\_zanimanja, odjel)

**zaposlenik** (id, id\_nadredeni\_zaposlenik, ime, prezime, identifikacijski\_broj, spol, broj\_telefona, broj\_mobitela, email, id\_zanimanje, id\_lokacija)

**kontakt\_klijenta** (id, email, broj\_mobitela, broj\_telefona, id\_klijent)

**prihod** (id, opis, tip\_prihoda)

**transakcija** (id, datum, iznos, broj\_racuna, placeno, id\_klijent, id\_zaposlenik)

**prihod\_za\_zaposlenika** (id, datum, id\_transakcija\_prihoda, id\_prihod)

**popust** (id, tip\_popusta)

**popust\_za\_klijenta** (id, datum\_primitka, datum\_koristenja, status, id\_klijent, id\_popust)

**poslovni\_trosak** (id, id\_transakcija\_poslovnog\_troska, svrha, opis)

**gotovinsko\_placanje** (id, id\_transakcija\_gotovina)

**karticno\_placanje** (id, tip\_kartice, id\_pravna\_osoba\_banka, id\_transakcija\_kartica)

**kriptovalutno\_placanje** (id, kriptovaluta, broj\_kripto\_novcanika, id\_transakcija\_kripto)

**kontakt\_pravne\_osobe** (id, email, broj\_mobitela, broj\_telefona, opis, id\_pravna\_osoba)

**serija (**id, ime, proizvodac, tip\_mjenjaca, broj\_sjedala, broj\_vrata)

**vozilo** (id, godina\_proizvodnje, registracijska\_tablica, tip\_punjenja, duljina, visina, nosivost, id\_serija, tip\_vozila)

**najam\_vozila** (id, id\_transakcija\_najam, id\_vozilo, datum\_pocetka, datum\_zavrsetka, status, pocetna\_kilometraza, zavrsna\_kilometraza)

**slika\_vozila** (id, id\_vozilo, slika, pozicija)

**tip osiguranja** (id, id\_osiguravajuca\_kuca, tip\_osiguranja)

**osiguranje** (id, id\_vozilo, id\_transakcija, id\_tip\_osiguranja, datum\_pocetka, datum\_zavrsetka)

**steta** (id, tip, opis)

**naknada\_stete** (id, datum\_pocetka, datum\_zavrsetka, id\_transakcija, id\_osiguranje, id\_steta)

**punjenje** (id, id\_transakcija\_punjenje, id\_vozilo, kolicina, tip\_punjenja)

**odrzavanje** (id, tip\_odrzavanja, id\_transakcija\_odrzavanje)

**rezervacija** (id, datum\_rezervacije, datum\_potvrde, id\_klijent)

**oprema** (id, naziv, opis)

**oprema\_na\_najmu** (id, id\_najam\_vozila, id\_oprema, kolicina)

**oprema\_na\_rezervaciji** (id, id\_rezervacija, id\_oprema, kolicina)

**vozilo\_na\_rezervaciji** (id, id\_rezervacija, id\_vozilo)

**crna\_lista** (id, id\_klijent, razlog)

# EER dijagram (MySQL)

**Slika 2**: EER dijagram

Slika na kojoj se prikazuje tekst, snimka zaslona, dijagram, broj

Opis je automatski generiran

*Izrada autora*

# Tablice

## TABLICA lokacija

Tablica *'lokacija'* služi za pohranu informacija o lokacijama. Sadrži atribute *id, drzava, grad, adresa.* Atribut *id* je jedinstveni identifikator odnosno primarni ključ tablice, tipa SMALLINT. Postavljen je kao AUTO\_INCREMENT što znači da će se automatski dodijeliti jedinstveni broj u svakom novom unosu i NOT NULL što znači da ne smije biti null vrijednost. *Drzava* označava državu lokacije. Atribut *drzava* definiran je kao VARCHAR(47) što znači da može sadržavati niz znakova duljine do 47 znakova. Atribut *grad* označava grad lokacije. Definiran je kao VARCHAR(100) što znači da može sadržavati niz znakova duljine do 100 znakova. Atribut *adresa* označava adresu lokacije. Definiran je kao VARCHAR(100) što znači da može sadržavati niz znakova duljine do 100 znakova.

CREATE TABLE lokacija (

id SMALLINT NOT NULL AUTO\_INCREMENT,

drzava VARCHAR(47),

grad VARCHAR(100),

adresa VARCHAR(100),

PRIMARY KEY (id)

);

## TABLICA pravna\_osoba

Tablica *'pravna\_osoba'* služi za pohranu informacija o pravnim osobama kao što su npr. tvrtke, organizacije, osiguravajuće kuće. Sadrži atribute *id\_pravna\_osoba, ime, identifikacijski\_broj, id\_lokacija*. Atribut *id* je jedinstveni identifikator odnosno primarni ključ tablice, tipa MEDIUMINT. Postavlja se kao AUTO\_INCREMENT što znači da će se automatski dodijeliti jedinstveni broj za svaki unos pravne osobe i NOT NULL što znači da ne smije biti null vrijednost. *Ime* označava naziv pravne osobe kao npr. naziv tvrtke ili organizacije. Polje je definirano tipa VARCHAR(100) što znači da može sadržavati niz znakova duljine do 100 znakova. *Identifikacijski\_broj* predstavlja id pravne osobe, koji može biti porezni broj, matični broj ili neki drugi identifikator. Polje je definirano kao VARCHAR(25) što znači da može sadržavati niz znakova duljine do 25 znakova. *Id\_lokacija* je identifikator lokacije na kojoj se nalazi pravna osoba, tipa SMALLINT, te referencira primarni ključ tablice 'lokacija' koja sadrži informacije o sjedištu svih pravnih osoba. *FOREIGN KEY (id\_lokacija)* je ograničenje stranog ključa koje osigurava da se vrijednost u polju id\_lokacija mora podudarati s primarnim ključem u tablici 'lokacija'.

CREATE TABLE pravna\_osoba (

id MEDIUMINT NOT NULL AUTO\_INCREMENT,

ime VARCHAR(100),

identifikacijski\_broj VARCHAR(25) NOT NULL UNIQUE,

id\_lokacija SMALLINT,

PRIMARY KEY (id),

FOREIGN KEY (id\_lokacija) REFERENCES lokacija (id)

);

## TABLICA klijent

Tablica klijent služi za pohranu informacija o klijentima. Sadrži atribute: id\_klijent, ime, prezime, identifikacijski\_broj i id\_pravna\_osoba. Atribut id\_klijent je primarni ključ tablice i tipa INTEGER. Ime, prezime i identifikacijski\_broj su tipa VARCHAR i služe za pohranu imena, prezimena i identifikacijskog broja klijenta. Id\_pravna\_osoba je tipa MEDIUMINT i označava kontakt informacije i pravnu osobu klijenta. Ovaj atribut je strani ključ koji referencira atribut id\_pravna\_osoba u tablici "pravna\_osoba".

CREATE TABLE klijent (

id INT NOT NULL AUTO\_INCREMENT,

ime VARCHAR (30),

prezime VARCHAR (30),

identifikacijski\_broj VARCHAR(13) NOT NULL UNIQUE,

id\_pravna\_osoba MEDIUMINT,

PRIMARY KEY (id),

FOREIGN KEY (id\_pravna\_osoba) REFERENCES pravna\_osoba (id)

);

## TABLICA zanimanje

Tablica zanimanje služi za pohranu informacija o različitim zanimanjima. Sadrži atribute: id, naziv, opis\_zanimanja, i odjel. Atribut id je primarni ključ tablice i tipa SMALLINT. Koristi se za jedinstvenu identifikaciju svakog zapisa u tablici i automatski se povećava za svaki novi unos. Atribut naziv je tipa VARCHAR i može sadržavati do 50 znakova. Služi za pohranu naziva zanimanja (npr. liječnik, inženjer, učitelj) te je definiran kao NOT NULL i UNIQUE, što znači da ne može biti prazan i mora biti jedinstven u tablici. Atribut opis\_zanimanja je tipa TEXT i služi za detaljan opis zanimanja. Ovaj atribut je također definiran kao NOT NULL, što znači da mora imati vrijednost. Atribut odjel je tipa VARCHAR i može sadržavati do 50 znakova. Služi za pohranu naziva odjela u kojem se nalazi to zanimanje (npr. Medicina, Inženjering, Obrazovanje) i definiran je kao NOT NULL, što znači da ne može biti prazan.

CREATE TABLE zanimanje (

id SMALLINT NOT NULL AUTO\_INCREMENT,

naziv VARCHAR(50) NOT NULL UNIQUE,

opis\_zanimanja TEXT NOT NULL,

odjel VARCHAR(50) NOT NULL,

PRIMARY KEY (id)

);

## TABLICA zaposlenik

Tablica zaposlenik služi za pohranu informacija o zaposlenicima. Sadrži atribute: id\_zaposlenik, id\_nadredeni\_zaposlenik, ime, prezime, identifikacijski\_broj, spol, broj\_telefona, broj\_mobitela, email i id\_zanimanje. Atribut id\_zaposlenik je primarni ključ tablice i tipa MEDIUMINT. Id\_nadredeni\_zaposlenik je tipa MEDIUMINT i označava nadređenog zaposlenika. Ime i prezime su tipa VARCHAR i služe za pohranu imena i prezimena zaposlenika. Identifikacijski\_broj je također tipa VARCHAR i označava identifikacijski broj zaposlenika, taj atribut je UNIQUE. Spol je tipa CHAR i označava spol zaposlenika, moguće vrijednosti su 'M' (muško) i 'F' (žensko). Broj\_telefona, broj\_mobitela i email su atributi tipa VARCHAR i označavaju broj telefona, broj mobitela i email adresu zaposlenika. Strani ključ id\_zanimanje koji je tipa SMALLINT, referencira atribut id\_zanimanje u tablici zanimanje, omogućujući povezivanje informacija o zanimanju zaposlenika s podacima u tablici zaposlenik. Strani ključ id\_lokacija je tipa SMALLINT i referencira atribut id\_lokacija u tablici lokacija, omogućujući povezivanje informacija o lokaciji zaposlenika s podacima u tablici zaposlenik dok strani ključ id\_nadredeni\_zaposlenik referencira atribut id\_nadredeni\_zaposlenik, omogućujući povezivanje informacija o nadređenom zaposleniku s podacima u tablici zaposlenik.

CREATE TABLE zaposlenik (

id MEDIUMINT NOT NULL AUTO\_INCREMENT,

id\_nadredeni\_zaposlenik MEDIUMINT,

ime VARCHAR (30) NOT NULL,

presume VARCHAR (30) NOT NULL,

identifikacijski\_broj VARCHAR (15) NOT NULL UNIQUE,

spol CHAR (1) CHECK (spol IN ('M', 'F')),

broj\_telefona VARCHAR (20) UNIQUE,

broj\_mobitela VARCHAR (20) UNIQUE,

email VARCHAR (320) UNIQUE,

id\_zanimanje SMALLINT NOT NULL,

id\_lokacija SMALLINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_zanimanje) REFERENCES zanimanje (id),

FOREIGN KEY (id\_lokacija) REFERENCES lokacija (id)

FOREIGN KEY (id\_nadredeni\_zaposlenik) REFERENCES zaposlenik (id)

);

## TABLICA kontakt\_klijenta

Tablica kontakt\_klijenta služi za pohranu kontaktnih informacija. Sadrži atribute: id, email, broj\_mobitela i broj\_telefona. Atribut id\_ kontakt\_klijenta je primarni ključ tablice i tipa INTEGER. Email, broj\_mobitela i broj\_telefona su tipa VARCHAR i služe za pohranu informacija o kontaktu poput e-mail adrese, broja mobitela i broja telefona, definirani su kao UNIQUE atributi, što znači da svaki od tih atributa mora biti jedinstven u tablici. Strani ključ u tablici id\_klijent referencira odgovarajući atribut u tablici "klijent". Osim toga, ima ograničenje ON DELETE CASCADE, što znači da će se svi kontakti povezani s klijentom automatski izbrisati kada se klijent izbriše.

CREATE TABLE kontakt\_klijenta (

id INT NOT NULL AUTO\_INCREMENT,

email VARCHAR (320) UNIQUE,

broj\_mobitela VARCHAR (20) UNIQUE,

broj\_telefona VARCHAR (20) UNIQUE,

id\_klijent INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_klijent) REFERENCES klijent (id) ON DELETE CASCADE

);

## TABLICA prihod

Tablica prihod služi za pohranu informacija o prihodima. Sadrži atribute: *id, opis, i tip\_prihoda*. Atribut *id\_prihod* je primarni ključ tablice i tipa TINYINT. Atribut *opis* je tipa TEXT i može sadržavati do 200 znakova, a služi za pohranu opisa prihoda. Atribut *tip\_prihoda* je tipa VARCHAR i može sadržavati do 25 znakova, a služi za pohranu tipa prihoda (npr. prodaja, najam, usluga) te je definiran kao NOT NULL, što znači da ne može biti prazan.

CREATE TABLE prihod (

id TINYINT NOT NULL AUTO\_INCREMENT,

opis TEXT(200),

tip\_prihoda VARCHAR(25) NOT NULL,

PRIMARY KEY (id)

);

## TABLICA transakcija

Tablica transakcija služi za pohranu informacija o transakcijama. Sadrži atribute: *id, datum, iznos, broj\_racuna, placeno, id\_klijent i id\_zaposlenik*. Atribut *id\_transakcija* je primarni ključ tablice i tipa BIGINT. Datum je tipa DATE i označava datum transakcije. Iznos i placeno su tipa DECIMAL i označavaju iznos transakcije i plaćen iznos, te iznos koji ostaje tj duguje kad se oduzme plaćen iznos. Broj\_racuna je tipa VARCHAR i označava broj računa. Strani ključ id\_klijent koji je tipa INTEGER, referencira atribut id\_klijent u tablici klijent, omogućujući povezivanje informacija o klijentu s podacima u tablici transakcija. Strani ključ id\_zaposlenik je tipa MEDIUMINT, i referencira atribut id\_zaposlenik u tablici zaposlenik, što označava zaposlenika povezanog s transakcijom.

CREATE TABLE transakcija (

id BIGINT NOT NULL AUTO\_INCREMENT,

datum DATE,

iznos DECIMAL (12,2),

broj\_racuna VARCHAR (7),

placeno DECIMAL (12,2),

id\_klijent INT,

id\_zaposlenik MEDIUMINT,

PRIMARY KEY (id),

FOREIGN KEY (id\_klijent) REFERENCES klijent (id),

FOREIGN KEY (id\_zaposlenik) REFERENCES zaposlenik (id)

);

## TABLICA prihod\_za\_zaposlenika

Tablica *'prihod\_za\_zaposlenika'* služi za pohranu i praćenje informacija o prihodima koje zaposlenici ostvaruju kroz neki vid transakcije, povezujući ih s odgovarajućim zaposlenicima, transakcijama i vrstama prihoda. Sadrži atribute *id, datum, id\_transakcija\_prihoda, id\_prihod.* *Id\_prihod\_za\_zaposlenika* jedinstveni je identifikator za svaki unos prihoda za zaposlenika u tablici, tipa INTEGER i primarni je ključ tablice. Postavlja se kao AUTO\_INCREMENT što znači da će se automatski dodijeliti jedinstveni broj za svaki novi unos prihoda za zaposlenika i NOT NULL što znači da ne smije biti null vrijednost. *Datum* označava datum kada je prihod ostvaren. Polje je tipa DATE i služi za pohranu datuma u formatu '%Y-%m-%d' gdje '%Y' predstavlja godinu u četveroznamenkastom formatu, '%m' mjesec u dvoznamenkastom formatu i '%d' dan u dvoznamenkastom formatu. Uz pomoć MySQL funkcije 'STR\_TO\_DATE' pretvara niz znakova u datum u skladu s odabranim formatom. *Id\_transakcija\_prihoda* je identifikator transakcije koja je rezultirala prihodom. Polje je tipa BIGINT i referencira primarni ključ tablice *'transakcija'* koja sadrži informacije o svim transakcijama. *Id\_prihod* je identifikator vrste prihoda koji je ostvaren. Polje je tipa TINYINT i referencira primarni ključ tablice '*prihod'.* *FOREIGN KEY* (*id\_transakcija\_prihoda)* je ograničenje stranog ključa koje osigurava da se vrijednost u polju *id\_transakcija\_prihoda* mora podudarati s primarnim ključem u tablici '*transakcija'.* Ima ograničenje ON DELETE CASCADE, što znači da će se svi prihodi zaposlenika povezani s transakcijom automatski izbrisati kada se izbriše transakcija. *FOREIGN KEY (id\_prihod)* predstavlja ograničenje stranog ključa koje osigurava da se vrijednost u polju *id\_prihod* mora podudarati s primarnim ključem u tablici '*prihod'.*

CREATE TABLE prihod\_za\_zaposlenika (

id INT NOT NULL AUTO\_INCREMENT,

datum DATE,

id\_transakcija\_prihoda BIGINT NOT NULL,

id\_prihod TINYINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_transakcija\_prihoda) REFERENCES transakcija(id) ON DELETE CASCADE,

FOREIGN KEY (id\_prihod) REFERENCES prihod(id)

);

## TABLICA popust

Tablica popust služi za pohranu informacija o raznim vrstama popusta. Sadrži atribute: id i tip\_popusta. Atribut id je primarni ključ tablice i tipa TINYINT. Atribut tip\_popusta je tipa VARCHAR i može sadržavati do 100 znakova. Ovaj atribut je definiran kao NOT NULL, što znači da ne može biti prazan, te služi za pohranu naziva ili opisa tipa popusta (npr. sezonski popust, popust na količinu, promotivni popust).

CREATE TABLE popust (

id TINYINT NOT NULL AUTO\_INCREMENT,

tip\_popusta VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

);

## TABLICA popust\_za\_klijenta

Tablica popust\_za\_klijenta služi za pohranu informacija o popustima dodijeljenim klijentima. Sadrži atribute: id, datum\_primitka, datum\_koristenja, status, id\_klijent i id\_popust. Atribut id je primarni ključ tablice i tipa INT. Koristi se za jedinstvenu identifikaciju svakog zapisa u tablici i automatski se povećava za svaki novi unos. Atribut datum\_primitka je tipa DATE i služi za pohranu datuma kada je klijent primio popust. Nadalje, atribut datum\_koristenja je tipa DATE i služi za pohranu datuma kada je klijent iskoristio popust. Atribut status je tipa VARCHAR i može sadržavati do 50 znakova. Služi za pohranu statusa popusta (npr. aktivan, iskorišten, istekao). Atribut id\_klijent je tipa INT i služi kao strani ključ koji referencira atribut id u tablici klijent. Ovaj atribut je definiran kao NOT NULL, što znači da mora imati vrijednost. Također ima ograničenje ON DELETE CASCADE, što znači da će se svi popusti povezani s klijentom automatski izbrisati kada se klijent izbriše. Atribut id\_popust je tipa TINYINT i služi kao strani ključ koji referencira atribut id u tablici popust. Ovaj atribut je definiran kao NOT NULL, što znači da mora imati vrijednost.

CREATE TABLE popust\_za\_klijenta (

id INT NOT NULL AUTO\_INCREMENT,

datum\_primitka DATE,

datum\_koristenja DATE,

status VARCHAR(50),

id\_klijent INT NOT NULL,

id\_popust TINYINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_klijent) REFERENCES klijent (id) ON DELETE CASCADE,

FOREIGN KEY (id\_popust) REFERENCES popust (id)

);

## TABLICA poslovni\_trosak

Tablica poslovni\_trosak služi za pohranu informacija o poslovnim troškovima. Sadrži atribute: id, id\_transakcija\_poslovnog\_troska, svrha, i opis. Atribut id je primarni ključ tablice i tipa INT. Koristi se za jedinstvenu identifikaciju svakog zapisa u tablici i automatski se povećava za svaki novi unos. Atribut id\_transakcija\_poslovnog\_troska je tipa BIGINT i služi kao strani ključ koji referencira atribut id u tablici transakcija. Ovaj atribut je definiran kao NOT NULL, što znači da mora imati vrijednost. Ima ograničenje ON DELETE CASCADE, što znači da će se svi poslovni troškovi povezani s transakcijom automatski izbrisati kada se transakcija izbriše. Atribut svrha je tipa VARCHAR i može sadržavati do 40 znakova. Služi za pohranu svrhe poslovnog troška (npr. putni troškovi, uredski materijal). Atribut opis

CREATE TABLE poslovni\_trosak (

id INT NOT NULL AUTO\_INCREMENT,

id\_transakcija\_poslovnog\_troska BIGINT NOT NULL,

svrha VARCHAR(40),

opis TEXT(200),

PRIMARY KEY (id),

FOREIGN KEY (id\_transakcija\_poslovnog\_troska) REFERENCES transakcija (id) ON DELETE CASCADE

);

## TABLICA gotovinsko\_placanje

Tablica *'gotovinsko\_placanje'* služi za pohranu i praćenje informacija o transakcijama koje su izvršene gotovinskim plaćanjem, a omogućuje nam povezivanje specifičnih situacija transakcija s odgovarajućim gotovinskim plaćanjem. Sadrži atribut *id* koji je ujedno primarni ključ i jedinstveni identifikator, za svaku transakciju koja je izvršena gotovinskim plaćanjem. Atribut je tipa BIGINT, a definiran je kao NOT NULL što znači da ne smije biti null vrijednost. *FOREIGN KEY (id\_gotovinsko\_placanje)* predstavlja primarni ključ u tablici *'gotovinsko\_placanje'*, a istovremeno je i strani ključ koji se referencira na tablicu *'transakcija'.* Na ovaj način je osiguran integritet podataka između ove dvije tablice. Ovo koristimo jer želimo osigurati da svaki unos u tablici '*gotovinsko\_placanje'* mora odgovarati unosu u tablici '*transakcija'*.

CREATE TABLE gotovinsko\_placanje (

id BIGINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id) REFERENCES transakcija (id)

## TABLICA karticno\_placanje

Ova tablica se koristi za praćenje informacija o transakcijama koje su izvršene plaćanjem karticom te vođenje evidencije o transakcijama kartičnog plaćanja kako bismo vidjeli koji su tipovi kartica korišteni za transakcije. Sadrži atribute *id, tip\_kartice, id\_pravna\_osoba\_banka, id\_transakcija\_kartica.* Atribut *id\_karticno\_placanje* predstavlja jedinstveni identifikator za svaku transakciju kartičnog plaćanja, tipa je BIGINT i primarni je ključ tablice. Definiran je kao NOT NULL, što znači da ne smije biti null vrijednost. Atribut *tip\_kartice* je polje koje predstavlja tip kartice koja je korištena za transakciju. Definirano je kao VARCHAR(20) što znači da može sadržavati niz znakova duljine do 20 znakova. *Id\_pravna\_osoba\_banka* predstavlja identifikator pravne osobe (banke) koja je izdala karticu. Definiran je kao MEDIUMINT i referencira primarni ključ tablice *'pravna\_osoba',* koja sadrži informacije o pravnim osobama, u ovom slučaju banci. *FOREIGN KEY (id)* predstavlja primarni ključ u tablici *'karticno\_placanje'*, a istovremeno je i strani ključ koji se referencira na tablicu *'transakcija'.* Na ovaj način je osiguran integritet podataka između ove dvije tablice. Ovo koristimo jer želimo osigurati da svaki unos u tablici '*karticno\_placanje'* mora odgovarati unosu u tablici '*transakcija'*. *FOREIGN KEY (id\_pravna\_osoba)* je ograničenje stranog ključa koje osigurava da se vrijednost u polju *'id\_pravna\_osoba\_banka'* mora podudarati s primarnim ključem u tablici '*pravna\_osoba'.*

CREATE TABLE karticno\_placanje (

id BIGINT NOT NULL,

tip\_kartice VARCHAR(20),

id\_pravna\_osoba\_banka MEDIUMINT,

PRIMARY KEY (id),

FOREIGN KEY (id) REFERENCES transakcija (id),

FOREIGN KEY (id\_pravna\_osoba\_banka) REFERENCES pravna\_osoba (id)

);

## TABLICA kriptovalutno\_placanje

Ova tablica se koristi za praćenje informacija o transakcijama koje su izvršene kriptovalutnim plaćanjem te vođenje evidencije o transakcijama kriptovalutnog plaćanja kako bismo vidjeli koji su kriptovalute korištene za transakcije. Sadrži atribute *id, kriptovaluta, broj\_kripto\_novcanika, id\_transakcija\_kripto*. Atribut *id\_kriptovalutno\_placanje* predstavlja jedinstveni identifikator za svaku kriptovalutnu transakciju i polje je tipa BIGINT. Definiran je kao NOT NULL, što znači da ne smije biti null vrijednost. Primarni je ključ tablice. *Kriptovaluta* je polje koje predstavlja naziv kriptovalute koja je korištena za transakciju. Definirano je kao VARCHAR(50) NOT NULL što znači da može sadržavati niz znakova duljine do 50 znakova i ne smije biti null vrijednost.. *Broj\_kripto\_novcanika* predstavlja broj kripto novčanika (wallet-a) povezanog s transakcijom. Definiran je kao VARCHAR(60) NOT NULL što znači da može sadržavati niz znakova duljine do 60 znakova i ne smije biti null vrijednost. *FOREIGN KEY (id)* je primarni ključ u tablici *'kriptovalutno\_placanje'*, a istovremeno je i strani ključ koji se referencira na tablicu *'transakcija'.* Na ovaj način je osiguran integritet podataka između ove dvije tablice. Ovo koristimo jer želimo osigurati da svaki unos u tablici '*kriptovalutno\_placanje'* mora odgovarati unosu u tablici '*transakcija'*.

CREATE TABLE kriptovalutno\_placanje (

id BIGINT NOT NULL,

kriptovaluta VARCHAR(50) NOT NULL,

broj\_kripto\_novcanika VARCHAR(60) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id) REFERENCES transakcija (id)

);

## TABLICA kontakt\_pravne\_osobe

Ova tablica se koristi za pohranu kontaktnih informacija o pravnim osobama ili osobama koje su povezane s pravnim osobama, a omogućuje povezivanje tih osoba s odgovarajućom pravnom osobom. Sadrži atribute *id, email, broj\_mobitela, broj\_telefona, opis, id\_pravna\_osoba*. Atribut *id\_kontakt\_pravne\_osobe* je primarni ključ tablice odnosno jedinstveni identifikator za svaki unos kontaktnog podatka, definiran kao AUTO\_INCREMENT polje tipa INTEGER, što znači da će se automatski dodijeliti jedinstveni broj za svaki novi unos, i NOT NULL što znači da ne smije biti null vrijednost. *Email* polje predstavlja e-mail adresu kontakta. Definirano je kao VARCHAR(320) kako bi se mogle pohraniti e-mail adrese koje su u skladu s RFC 5321 standardom, a označeno je kao UNIQUE kako bi se osigurala jedinstvenost e-mail adresa u tablici. Atributi b*roj\_mobitela i broj\_telefona* su polja koje predstavlja brojeve mobilnog odnosno fiksnog telefona kontakta, tipa su VARCHAR(20) što je dovoljno za pohranu raznih formata fiksnih i mobilnih brojeva. Atribut *opis* je polje koje omogućuje dodavane kratkog opisa kontakta, definirano kao VARCHAR(100) što bi trebalo biti dovoljno za opisivanje kontakta. *Id\_pravna\_osoba* predstavlja identifikator pravne osobe s kojom je kontakt povezan, definiran kao MEDIUMINT, a referencira primarni ključ tablice *'pravna\_osoba'. FOREIGN KEY (id\_pravna\_osoba)* predstavlja ograničenje stranog ključa koje osigurava da se vrijednost u polju *'id\_pravna\_osoba'* mora podudarati primarnim ključem u tablici *'pravna\_osoba'*.

CREATE TABLE kontakt\_pravne\_osobe (

id INT NOT NULL AUTO\_INCREMENT,

email VARCHAR(320) UNIQUE,

broj\_mobitela VARCHAR(20),

broj\_telefona VARCHAR(20),

opis VARCHAR(100),

id\_pravna\_osoba MEDIUMINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_pravna\_osoba) REFERENCES pravna\_osoba(id) ON DELETE CASCADE

);

## TABLICA serija

Tablica koja služi za pohranu informacija o serijama automobila ili kamiona te je korisna za vođenje evidencije o različitim modelima vozila. Sadrži atribute *id, ime, proizvođač, tip mjenjača, broj\_sjedala, broj\_vrata*. Atribut  *id\_serija* je jedinstveni identifikator za svaku seriju automobila/kamiona i primarni je ključ tablice Definiran je kao AUTO\_INCREMENT polje tipa SMALLINT, što znači da će se automatski dodijeliti jedinstveni broj za svaki novi unos, te NOT NULL što znači da ne smije biti null vrijednost. *Ime* predstavlja ime serije automobila/kamiona, definirano kao VARCHAR(100) NUT NULL UNIQUE što znači da može sadržavati niz znakova duljine do 100 znakova, da ne smije biti null vrijednost i da mora biti jedinstveno ime. *Proizvodac* je polje koje predstavlja proizvođača serije automobila/kamiona, definirano kao VARCHAR(40) što znači da može sadržavati niz znakova duljine do 40 znakova. *Tip\_mjenjaca* predstavlja tip mjenjača serije automobila/kamiona, definiran kao VARCHAR(10) što znači da može sadržavati niz znakova duljine do 10 znakova. *Broj\_sjedala* predstavlja broj sjedala automobila ili kamiona (može biti 2, 3 ili 5), definiran kao TINYINT. *Broj\_vrata* predstavlja broj vrata automobila ili kamiona (može biti 2, 3 ili 5), definiran kao TINYINT.

CREATE TABLE serija (

id SMALLINT NOT NULL AUTO\_INCREMENT,

ime VARCHAR(100) NOT NULL UNIQUE,

proizvodac VARCHAR(40),

tip\_mjenjaca VARCHAR(10),

broj\_vrata TINYINT,

broj\_vrata TINYINT,

PRIMARY KEY (id)

);

## TABLICA vozilo

Tablica vozilo služi za pohranu informacija o vozilima koja se iznajmljuju. Sadrži atribute: id\_vozilo, godina\_proizvodnje, registracijska\_tablica i tip\_punjenja. Atribut id\_vozilo je primarni ključ tablice i tipa INTEGER. Godina\_proizvodnje je tipa YEAR i označava godinu proizvodnje vozila. Registracijska\_tablica je tipa VARCHAR i služi za pohranu registracijske tablice vozila. Tip\_punjenja je također tipa VARCHAR i označava način punjenja goriva (npr. benzin, dizel, električni). Duljina je tipa NUMERIC i označava duljinu vozila, visina je tipa NUMERIC i označava visinu vozila. Nosivost je tipa NUMERIC i označava nosivost vozila, id\_serija je tipa SMALLINT i strani je ključ koji referencira id u tablici serija. Tip\_vozila je tipa CHAR i može imati vrijednosti 'K' (kamion), 'M' (motocikl), ili 'A' (automobil). Ovaj atribut ima CHECK ograničenje koje osigurava da vrijednost bude jedna od navedenih.

CREATE TABLE vozilo (

id INT NOT NULL AUTO\_INCREMENT,

godina\_proizvodnje YEAR,

registracijska\_tablica VARCHAR (15) NOT NULL,

tip\_punjenja VARCHAR (10),

duljina DECIMAL (4, 2),

visina DECIMAL (3, 2),

nosivost DECIMAL (7, 2),

id\_serija SMALLINT,

tip\_vozila CHAR (1) CHECK (tip\_vozila IN ('K', 'M', 'A')),

PRIMARY KEY (id),

FOREIGN KEY (id\_serija) REFERENCES serija (id)

);

## TABLICA najam\_vozila

Tablica najam\_vozila služi za praćenje najma vozila. Sadrži atribute: id\_najam\_vozila, id\_transakcija\_najam, id\_vozilo, datum\_pocetka, datum\_zavrsetka, status, pocetna\_kilometraza i zavrsna\_kilometraza. Atribut id\_najam\_vozila je primarni ključ tablice i tipa INTEGER. Id\_transakcija\_najam je tipa BIGINT, ne može biti NULL, i označava transakciju koja je povezana s najmom vozila. Id\_vozilo je tipa INT, također ne može biti NULL i označavaju vozilo koje povezano s najmom. Datum\_pocetka i datum\_zavrsetka su tipa DATE i označavaju datum početka i završetka najma. Status je tipa VARCHAR, ne može biti NULL i označava trenutni status najma vozila. Pocetna\_kilometraza i zavrsna\_kilometraza su tipa DECIMAL i označavaju početnu i završnu kilometražu vozila tijekom najma.. Svaki strani ključ u tablici "najam\_vozila" referencira odgovarajući atribut u svojoj tablici: id\_transakcija\_najam referencira id u tablici "transakcija" i id\_vozilo referencira id u tablici "vozilo".

CREATE TABLE najam\_vozila (

id INT NOT NULL AUTO\_INCREMENT,

id\_transakcija\_najam BIGINT NOT NULL,

id\_vozilo INT NOT NULL,

datum\_pocetka DATE,

datum\_zavrsetka DATE,

status VARCHAR (15) NOT NULL,

pocetna\_kilometraza DECIMAL (10, 2),

zavrsna\_kilometraza DECIMAL (10, 2),

PRIMARY KEY (id),

FOREIGN KEY (id\_transakcija\_najam) REFERENCES transakcija (id) ON DELETE CASCADE,

FOREIGN KEY (id\_vozilo) REFERENCES vozilo (id)

);

## TABLICA slika\_vozila

Ova tablica omogućuje pohranu slika za pojedinačna vozila. Sadrži atribute id, id\_vozilo, slika, i pozicija. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao INT NOT NULL AUTO\_INCREMENT što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Id\_vozilo* predstavlja identifikator vozila na koje se odnosi slika. Definiran je kao INT NOT NULL i referencira primarni ključ tablice vozilo. Ovo polje omogućuje povezivanje svake slike s odgovarajućim vozilom. *Slika* je atribut tipa MEDIUMBLOB, što omogućuje pohranu binarnih podataka do 16 MB. Ovo polje služi za pohranu same slike vozila. *Pozicija* je atribut tipa VARCHAR(30) koji opisuje položaj ili vrstu slike (npr. prednja strana, stražnja strana, unutrašnjost). Tablica također sadrži ograničenje stranog ključa FOREIGN KEY (id\_vozilo) REFERENCES vozilo(id) ON DELETE CASCADE, koje osigurava da se svi zapisi povezani s vozilom automatski brišu ako se izbriše odgovarajuće vozilo.

CREATE TABLE slika\_vozila (

id INT NOT NULL AUTO\_INCREMENT,

id\_vozilo INT NOT NULL,

slika MEDIUMBLOB,

pozicija VARCHAR(30),

PRIMARY KEY (id),

FOREIGN KEY (id\_vozilo) REFERENCES vozilo (id) ON DELETE CASCADE

);

## TABLICA tip\_osiguranja

Ova tablica omogućuje pohranu podataka o različitim vrstama osiguranja koja su dostupna. Sadrži atribute *id, id\_osiguravajuca\_kuca, i tip\_osiguranja*. *Id\_tip\_osiguranja* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao SMALLINT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu i da ne smije biti null vrijednost. Ovo polje služi kao primarni ključ tablice. *Id\_osiguravajuca\_kuca* predstavlja identifikator osiguravajuće kuće koja nudi osiguranje. Definiran je kao MEDIUMINT i referencira primarni ključ tablice pravna\_osoba. *Tip\_osiguranja* je atribut tipa VARCHAR(20) koji opisuje vrstu osiguranja (npr. osnovno, prošireno). Tablica također sadrži ograničenje stranog ključa *FOREIGN KEY (id\_osiguravajuca\_kuca)* REFERENCES pravna\_osoba(id), koje osigurava referencijalni integritet podataka.

CREATE TABLE tip\_osiguranja (

id SMALLINT NOT NULL AUTO\_INCREMENT,

id\_osiguravajuca\_kuca MEDIUMINT,

tip\_osiguranja VARCHAR(20),

PRIMARY KEY (id),

FOREIGN KEY (id\_osiguravajuca\_kuca) REFERENCES pravna\_osoba (id)

);

## TABLICA osiguranje

Ova tablica služi za praćenje osiguranja vozila. Sadrži atribute *id, id\_vozilo, id\_transakcija, id\_tip\_osiguranja, datum\_pocetka, datum\_zavrsetka*. Atribut *id\_osiguranje* je primarni ključ tipa INT, što omogućuje jedinstveno identificiranje svake police osiguranja. Definiran je kao NOT NULL AUTO\_INCREMENT što znači da ne smije biti null vrijednost i da će se automatski dodijeliti jedinstveni broj za svaki novi unos osiguranja. *Id\_vozilo* je strani ključ koji referencira tablicu vozilo. Identificira vozilo koje je osigurano. Odabran je INT NOT NULL tip podataka jer se referencira na primarni ključ druge tablice vozilo i ne smije biti null vrijednost. *Id\_transakcija* je strani ključ koji referencira tablicu transakcija. Identificira transakciju vezanu uz kupnju osiguranja. Odabran je BIGINT NOT NULL tip podataka kako bi se osigurala mogućnost pohrane većeg broja zapisa i ne smije biti null vrijednost. *Id\_tip\_osiguranja* je atribut tipa SMALLINT NOT NULL koji sadrži vrstu osiguranja, poput "osiguranje od krađe" ili "obvezno osiguranje od automobilske odgovornosti" i ne smije biti null vrijednost. Datum\_pocetka i datum\_zavrsetka su atributi tipa DATE koji određuju početak i kraj trajanja police osiguranja. *FOREIGN KEY (id\_vozilo)* predstavlja ograničenje stranog ključa koje osigurava da se vrijednost u polju *id\_vozilo* mora podudarati s primarnim ključem u tablici '*vozilo'. FOREIGN KEY (id\_transakcija)* predstavlja ograničenje stranog ključa koje osigurava da se vrijednost u polju *id\_transakcija* mora podudarati s primarnim ključem u tablici '*transakcija'. FOREIGN KEY (id\_tip\_osiguranja)* predstavlja ograničenje stranog ključa koje osigurava da se vrijednost u polju *id\_tip\_osiguranja* mora podudarati s primarnim ključem u tablici '*tip\_osiguranja'.*

CREATE TABLE osiguranje (

id INT NOT NULL AUTO\_INCREMENT,

id\_vozilo INT NOT NULL,

id\_transakcija BIGINT NOT NULL,

id\_tip\_osiguranja SMALLINT NOT NULL,

datum\_pocetka DATE,

datum\_zavrsetka DATE,

PRIMARY KEY (id),

FOREIGN KEY (id\_vozilo) REFERENCES vozilo (id),

FOREIGN KEY (id\_transakcija) REFERENCES transakcija (id),

FOREIGN KEY (id\_tip\_osiguranja) REFERENCES tip\_osiguranja (id) ON DELETE CASCADE

);

## TABLICA steta

Ova tablica služi za praćenje šteta koje su se dogodile vozilima ili opremi. Atribut id je primarni ključ tipa INT, što omogućuje jedinstveno identificiranje svakog zapisa u tablici. Tip štete predstavljen je atributom tip, koji je VARCHAR(20), jer se očekuje da će sadržavati tekstualne opise različitih vrsta šteta. Opis štete je tekstualni atribut opis, koji je tipa TEXT, što omogućuje pohranu duljih opisa šteta. Ovaj tip podataka omogućuje fleksibilnost u pohrani različitih količina teksta.

CREATE TABLE steta (

id INT NOT NULL AUTO\_INCREMENT,

tip VARCHAR(20),

opis TEXT(200),

PRIMARY KEY (id)

);

## TABLICA naknada\_stete

Ova tablica služi za praćenje naknada za štete koje su prijavljene i odobrene. Atribut id je primarni ključ tipa INT, što omogućuje jedinstveno identificiranje svakog zapisa u tablici. Datum početka i završetka naknade su atributi datum\_pocetka i datum\_zavrsetka, koji su tipa DATE. Odabran je DATE tip podataka jer se očekuje pohrana datuma početka i završetka naknade. Id\_transakcija, id\_osiguranje i id\_steta su strani ključevi koji referenciraju tablice transakcija, osiguranje i steta. Id\_transakcija je tipa BIGINT, a id\_osiguranje i id\_steta su tipa INT. Odabrani su ti tipovi podataka kako bi se osigurala dosljednost s primarnim ključevima drugih tablica.

CREATE TABLE naknada\_stete (

id INT NOT NULL AUTO\_INCREMENT,

datum\_pocetka DATE,

datum\_zavrsetka DATE,

id\_transakcija BIGINT NOT NULL,

id\_osiguranje INT,

id\_steta INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_transakcija) REFERENCES transakcija (id) ON DELETE CASCADE,

FOREIGN KEY (id\_steta) REFERENCES steta (id) ON DELETE CASCADE,

FOREIGN KEY (id\_osiguranje) REFERENCES osiguranje (id)

);

## TABLICA punjenje

Ova tablica služi za praćenje punjenja goriva vozila. Atribut id je primarni ključ tipa INT, što omogućuje jedinstveno identificiranje svakog zapisa u tablici. Id\_transakcija\_punjenje, id\_vozilo i kolicina su atributi koji označavaju transakciju punjenja, vozilo na koje se odnosi punjenje i količinu goriva. Id\_transakcija\_punjenje je tipa BIGINT, dok su id\_vozilo i količina tipa INT. Odabrani su ti tipovi podataka kako bi se osigurala dosljednost s primarnim ključevima i drugim atributima drugih tablica.

CREATE TABLE punjenje (

id INT NOT NULL AUTO\_INCREMENT,

id\_transakcija\_punjenje BIGINT NOT NULL,

id\_vozilo INT NOT NULL,

kolicina DECIMAL(7,3),

tip\_punjenja VARCHAR(10),

PRIMARY KEY (id),

FOREIGN KEY (id\_transakcija\_punjenje) REFERENCES transakcija (id) ON DELETE CASCADE,

FOREIGN KEY (id\_vozilo) REFERENCES vozilo (id) ON DELETE CASCADE

);

## TABLICA odrzavanje

Ova tablica omogućuje praćenje informacija o održavanju vozila. Sadrži atribute id, tip\_odrzavanja, i id\_transakcija\_odrzavanje. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao INT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Tip\_odrzavanja* je atribut tipa VARCHAR(100) koji opisuje vrstu održavanja (npr. redovito, izvanredno). *Id\_transakcija\_odrzavanje* predstavlja identifikator transakcije povezane s održavanjem. Definiran je kao BIGINT NOT NULL i referencira primarni ključ tablice transakcija. Tablica također sadrži ograničenje stranog ključa FOREIGN KEY (id\_transakcija\_odrzavanje) REFERENCES transakcija(id) ON DELETE CASCADE, koje osigurava da se svi zapisi povezani s transakcijom automatski brišu ako se izbriše odgovarajuća transakcija.

CREATE TABLE odrzavanje (

id INT NOT NULL AUTO\_INCREMENT,

tip\_odrzavanja VARCHAR(100),

id\_transakcija\_odrzavanje BIGINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_transakcija\_odrzavanje) REFERENCES transakcija (id) ON DELETE CASCADE

);

## TABLICA rezervacija

Ova tablica omogućuje pohranu podataka o rezervacijama vozila. Sadrži atribute id, datum\_rezervacije, datum\_potvrde, i id\_klijent. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao INT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Datum\_rezervacije* je atribut tipa DATE koji predstavlja datum kada je rezervacija napravljena. *Datum\_potvrde* je atribut tipa DATE koji predstavlja datum kada je rezervacija potvrđena. *Id\_klijent* predstavlja identifikator klijenta koji je napravio rezervaciju. Definiran je kao INT NOT NULL i referencira primarni ključ tablice klijent. Tablica također sadrži ograničenje stranog ključa FOREIGN KEY (id\_klijent) REFERENCES klijent(id) ON DELETE CASCADE, koje osigurava da se svi zapisi povezani s klijentom automatski brišu ako se izbriše odgovarajući klijent.

CREATE TABLE rezervacija (

id INT NOT NULL AUTO\_INCREMENT,

datum\_rezervacije DATE,

datum\_potvrde DATE,

id\_klijent INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_klijent) REFERENCES klijent (id) ON DELETE CASCADE

);

## TABLICA oprema

Ova tablica omogućuje pohranu podataka o opremi koja se može iznajmiti ili rezervirati uz vozila. Sadrži atribute id, naziv, i opis. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao SMALLINT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Naziv* je atribut tipa VARCHAR(30) koji predstavlja naziv opreme (npr. dječja sjedalica, GPS uređaj). *Opis* je atribut tipa TEXT(200) koji daje detaljan opis opreme.

CREATE TABLE oprema (

id SMALLINT NOT NULL AUTO\_INCREMENT,

naziv VARCHAR(30),

opis TEXT(200),

PRIMARY KEY (id)

);

## TABLICA oprema\_na\_najmu

Ova tablica omogućuje pohranu podataka o opremi koja je iznajmljena uz vozila. Sadrži atribute id, id\_najam\_vozila, id\_oprema, i kolicina. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao INT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Id\_najam\_vozila* predstavlja identifikator najma vozila. Definiran je kao INT NOT NULL i referencira primarni ključ tablice najam\_vozila. *Id\_oprema* predstavlja identifikator opreme. Definiran je kao SMALLINT NOT NULL i referencira primarni ključ tablice oprema. *Kolicina* je atribut tipa TINYINT koji označava količinu opreme koja je iznajmljena. Tablica sadrži ograničenja stranog ključa FOREIGN KEY (id\_najam\_vozila) REFERENCES najam\_vozila(id) ON DELETE CASCADE i FOREIGN KEY (id\_oprema) REFERENCES oprema(id), koja osiguravaju referencijalni integritet podataka.

CREATE TABLE oprema\_na\_najmu (

id INT NOT NULL AUTO\_INCREMENT,

id\_najam\_vozila INT NOT NULL,

id\_oprema SMALLINT NOT NULL,

kolicina TINYINT,

PRIMARY KEY (id),

FOREIGN KEY (id\_najam\_vozila) REFERENCES najam\_vozila (id) ON DELETE CASCADE,

FOREIGN KEY (id\_oprema) REFERENCES oprema (id)

););

## TABLICA oprema\_na\_rezervaciji

Ova tablica omogućuje pohranu podataka o opremi koja je rezervirana uz vozila. Sadrži atribute id, id\_rezervacija, id\_oprema, i kolicina. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao INT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Id\_rezervacija* predstavlja identifikator rezervacije. Definiran je kao INT NOT NULL i referencira primarni ključ tablice rezervacija. *Id\_oprema* predstavlja identifikator opreme. Definiran je kao SMALLINT NOT NULL i referencira primarni ključ tablice oprema. *Kolicina* je atribut tipa TINYINT koji označava količinu opreme koja je rezervirana. Tablica sadrži ograničenja stranog ključa FOREIGN KEY (id\_rezervacija) REFERENCES rezervacija(id) ON DELETE CASCADE i FOREIGN KEY (id\_oprema) REFERENCES oprema(id), koja osiguravaju referencijalni integritet podataka.

CREATE TABLE oprema\_na\_rezervaciji (

id INT NOT NULL AUTO\_INCREMENT,

id\_rezervacija INT NOT NULL,

id\_oprema SMALLINT NOT NULL,

kolicina TINYINT,

PRIMARY KEY (id),

FOREIGN KEY (id\_rezervacija) REFERENCES rezervacija (id) ON DELETE CASCADE,

FOREIGN KEY (id\_oprema) REFERENCES oprema (id)

);

## TABLICA vozilo\_na\_rezervaciji

Ova tablica omogućuje pohranu podataka o vozilima koja su rezervirana. Sadrži atribute id, id\_rezervacija, i id\_vozilo. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao INT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Id\_rezervacija* predstavlja identifikator rezervacije. Definiran je kao INT NOT NULL i referencira primarni ključ tablice rezervacija. *Id\_vozilo* predstavlja identifikator vozila. Definiran je kao INT NOT NULL i referencira primarni ključ tablice vozilo. Tablica sadrži ograničenja stranog ključa FOREIGN KEY (id\_rezervacija) REFERENCES rezervacija(id) ON DELETE CASCADE i FOREIGN KEY (id\_vozilo) REFERENCES vozilo(id), koja osiguravaju referencijalni integritet podataka.

CREATE TABLE vozilo\_na\_rezervaciji (

id INT NOT NULL AUTO\_INCREMENT,

id\_rezervacija INT NOT NULL,

id\_vozilo INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_rezervacija) REFERENCES rezervacija (id) ON DELETE CASCADE

FOREIGN KEY (id\_vozilo) REFERENCES vozilo (id),

);

## TABLICA crna\_lista

Ova tablica omogućuje pohranu podataka o klijentima koji su stavljeni na crnu listu. Sadrži atribute id, id\_klijent, i razlog. *Id* je jedinstveni identifikator za svaki zapis u tablici. Definiran je kao SMALLINT NOT NULL AUTO\_INCREMENT, što znači da će se automatski dodijeliti jedinstveni broj svakom novom unosu. Ovo polje služi kao primarni ključ tablice. *Id\_klijent* predstavlja identifikator klijenta koji je na crnoj listi. Definiran je kao INT NOT NULL i referencira primarni ključ tablice klijent. *Razlog* je atribut tipa TEXT koji opisuje razlog zašto je klijent stavljen na crnu listu. Atribut ima CHECK ograničenje CHECK (LENGTH(razlog) > 30) koje osigurava da je razlog dovoljno detaljan. Tablica također sadrži ograničenje stranog ključa FOREIGN KEY (id\_klijent) REFERENCES klijent(id) ON DELETE CASCADE, koje osigurava da se svi zapisi povezani s klijentom automatski brišu ako se izbriše odgovarajući klijent.

CREATE TABLE crna\_lista (

id SMALLINT NOT NULL AUTO\_INCREMENT,

id\_klijent INT NOT NULL,

razlog TEXT CHECK (LENGTH(razlog) < 40),

PRIMARY KEY (id),

FOREIGN KEY (id\_klijent) REFERENCES klijent (id) ON DELETE CASCADE

);

# UPITI

## Upiti – Marinela

**Upit 1:** Potrebno je pronaći sve klijente koji su na crnoj listi, razloge zašto su na crnoj listi, te koliko su puta ti klijenti unajmili vozilo.

SELECT k.id, k.ime, k.prezime, k.identifikacijski\_broj, cl.razlog, COUNT(nv.id) AS broj\_najmova

FROM klijent k

JOIN crna\_lista cl ON k.id = cl.id\_klijent

LEFT JOIN transakcija t ON k.id = t.id\_klijent

LEFT JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

GROUP BY k.id, k.ime, k.prezime, k.identifikacijski\_broj, cl.razlog;

**Upit 2:** Prikaži ukupan broj iznajmljenih komada određene opreme po rezervaciji padajućim redoslijedom i prikaži o kojoj opremi se radi

SELECT r.id AS rezervacija\_id, o.naziv AS naziv\_opreme, SUM(onr.kolicina) AS ukupan\_broj\_iznajmljene\_opreme

FROM rezervacija r

JOIN oprema\_na\_rezervaciji onr ON r.id = onr.id\_rezervacija

JOIN oprema o ON onr.id\_oprema = o.id

GROUP BY r.id, o.naziv

ORDER BY ukupan\_broj\_iznajmljene\_opreme DESC;

**Upit 3:** Pregled ukupnog broja transakcija po vrsti plaćanja

SELECT 'Gotovinsko placanje' AS vrsta\_placanja,

COUNT(gp.id) AS broj\_transakcija

FROM gotovinsko\_placanje gp

JOIN transakcija t ON gp.id = t.id

UNION

SELECT 'Karticno placanje' AS vrsta\_placanja,

COUNT(kp.id) AS broj\_transakcija

FROM karticno\_placanje kp

JOIN transakcija t ON kp.id = t.id

UNION

SELECT 'Kriptovalutno placanje' AS vrsta\_placanja,

COUNT(kp.id) AS broj\_transakcija

FROM kriptovalutno\_placanje kp

JOIN transakcija t ON kp.id = t.id;

**Upit 4:** Koja oprema se najrjeđe iznajmljivala, a koja najčešće u zadnjih godinu dana? (U svrhu nabavke većeg broja opreme koja se često iznajmljuje i moguće povlačenje opreme koja se rijetko iznajmljuje)

SELECT

oprema.naziv,

oprema.opis,

broj\_iznajmljivanja

FROM

(SELECT

oprema\_na\_najmu.id\_oprema,

COUNT(\*) AS broj\_iznajmljivanja

FROM

oprema\_na\_najmu

JOIN

najam\_vozila ON najam\_vozila.id = oprema\_na\_najmu.id\_najam\_vozila

WHERE

najam\_vozila.datum\_pocetka >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

oprema\_na\_najmu.id\_oprema

) AS IznajmljivanjaUPosljednjojGodini

JOIN

oprema ON oprema.id = IznajmljivanjaUPosljednjojGodini.id\_oprema

ORDER BY

broj\_iznajmljivanja ASC

LIMIT 1;

SELECT

oprema.naziv,

oprema.opis,

broj\_iznajmljivanja

FROM

(SELECT

oprema\_na\_najmu.id\_oprema,

COUNT(\*) AS broj\_iznajmljivanja

FROM

oprema\_na\_najmu

JOIN

najam\_vozila ON najam\_vozila.id = oprema\_na\_najmu.id\_najam\_vozila

WHERE

najam\_vozila.datum\_pocetka >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

oprema\_na\_najmu.id\_oprema

) AS IznajmljivanjaUPosljednjojGodini

JOIN

oprema ON oprema.id = IznajmljivanjaUPosljednjojGodini.id\_oprema

ORDER BY

broj\_iznajmljivanja DESC

LIMIT 1;

**Upit 5:** pregled svih naknada štete po vozilima, uključujući informacije o vozilu i ukupnu naknadu za svako vozilo.

SELECT

vozilo.registracijska\_tablica,

vozilo.tip\_vozila,

SUM(transakcija.iznos) AS ukupna\_naknada,

COUNT(naknada\_stete.id) AS broj\_naknada

FROM

naknada\_stete

JOIN

transakcija ON transakcija.id = naknada\_stete.id\_transakcija

JOIN

osiguranje ON osiguranje.id = naknada\_stete.id\_osiguranje

JOIN

vozilo ON vozilo.id = osiguranje.id\_vozilo

GROUP BY

vozilo.id

ORDER BY

ukupna\_naknada DESC;

**Upit 6:** Pet klijenata s najvećim brojem rezervacija

SELECT

k.ime AS ImeKlijenta,

k.prezime AS PrezimeKlijenta,

COUNT(r.id) AS BrojRezervacija

FROM

klijent k

LEFT JOIN

rezervacija r ON k.id = r.id\_klijent

GROUP BY

k.ime, k.prezime

ORDER BY

BrojRezervacija DESC

LIMIT 5;

## Upiti – Mirela

**Upit 1**

Tvrtka analizira podatke kako bi se identificirali ključni klijenti koji su generirali najveće troškove najma opreme i vozila tijekom odabrane vremenske periode. Analiza ponašanja visoko potrošačkih klijenata može pružiti dublji uvid u trendove na tržištu najma vozila i opreme te pomoći u donošenju informiranih poslovnih odluka.

Tvrtka analizira koji su to klijenti s najviše potrošenog novca na najam vozila i opreme. Time može bolje razumjeti potrebe tih klijenata i planirati kako im bolje pružiti usluge.

**Zadatak:** prikaži 3 klijenta s najviše potrošenog novca na najam vozila i opreme

SELECT k.id AS id\_klijent,

CONCAT(k.ime, ' ', k.prezime) AS ime\_prezime\_klijenta,

SUM(t.iznos) AS ukupni\_iznos

FROM klijent k

INNER JOIN transakcija t ON k.id = t.id\_klijent

INNER JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

INNER JOIN oprema\_na\_najmu onr ON nv.id = onr.id\_najam\_vozila

GROUP BY k.id

ORDER BY ukupni\_iznos DESC

LIMIT 3;

**Odabir podataka:** u upitu prvo odabirem stupce k.id (identifikator klijenta), k.ime (ime klijenta), k.prezime (prezime klijenta) koje sam spojila pomoću funkcije CONCAT u jedan stupac i nazvala ga ime\_prezime\_klijenta.

**Povezivanje tablica:** Povezujem tablice klijent, transakcija, najam\_vozila i oprema\_na\_najmu koristeći INNER JOIN na odgovarajuće ključeve..

**Grupiranje podataka:** Grupiram rezultate po identifikatoru klijenta kako bih dobila ukupni iznos potrošenog novca za svakog klijenta uz agregiranje sa funkcijom SUM.

**Sortiranje i ograničavanje rezultata:** Sortiram rezultate silazno po ukupnom iznosu potrošenog novca, a zatim ograničavam na prvih 3 rezultata.

**Upit 2**

Ako se određena vrsta opreme rijetko iznajmljuje, tvrtka može smanjiti zalihu te opreme kako bi oslobodila skladišni prostor i smanjila troškove održavanja zaliha. Također može donijeti odluke o tome treba li smanjiti količinu određene opreme ili je ukloniti iz ponude. Zatim ćemo koristiti taj pogled u našem upitu kako bismo pronašli vrstu opreme koja se najmanje iznajmljivala.

**Zadatak:** prikaz 5 vrsta dodatne opremu koja se najmanje puta iznajmila

CREATE VIEW Broj\_Najmova\_Opreme AS

SELECT o.naziv AS Vrsta\_Opreme, COUNT(onr.id\_najam\_vozila) AS Broj\_Najmova

FROM oprema o

LEFT JOIN oprema\_na\_najmu onr ON o.id = onr.id\_oprema

GROUP BY o.naziv;

SELECT Vrsta\_Opreme

FROM Broj\_Najmova\_Opreme

ORDER BY Broj\_Najmova ASC

LIMIT 5;

**Kreiranje pogleda za izračun broja najmova za svaku vrstu opreme:** Kreirala sam pogled nazvan "Broj\_Najmova\_Opreme" kako bih izračunala broj najmova za svaku vrstu opreme. U ovom pogledu koristim SELECT naredbu kako bih odabrala naziv vrste opreme (o.naziv) i povezujem tablice "oprema" s tablicom "oprema\_na\_najmu" pomoću LEFT JOIN kako bih uključila sve vrste opreme, čak i one koje se nisu iznajmljivale. Na kraju grupiram rezultate po nazivu vrste opreme pomoću GROUP BY i agregiram ih koristeći funkciju COUNT kako bih dobila Broj\_Najmova.

**Korištenje pogleda za pronalaženje vrste opreme koja se najmanje puta iznajmljivala:** Nakon što sam kreirala pogled za izračun broja najmova za svaku vrstu opreme, koristim ga u glavnom upitu kako bih pronašla vrstu opreme koja se najmanje puta iznajmljivala. Koristim SELECT naredbu za odabir naziva vrste opreme iz pogleda "Broj\_Najmova\_Opreme". Sortiram rezultate uzlazno po broju najmova (Broj\_Najmova) koristeći ORDER BY klauzulu s ASC opcijom kako bih dobila vrstu opreme koja se najmanje puta iznajmljivala na vrhu rezultata. Na kraju koristim LIMIT klauzulu kako bih ograničila prikaz na prvih 5 vrsta opreme, fokusirajući se samo na one koje su se najmanje puta iznajmljivale.

**Upit 3**

Tvrtka provjerava koja su vozila prešla više kilometara nego što je očekivano tijekom najma. To joj može pomoći da održava vozni park u dobrom stanju i izbjegava neočekivane troškove popravaka.

**Zadatak:** Provjera najma vozila koji su premašili brojku od 100 kilometara

SELECT nv.id, v.registracijska\_tablica, nv.datum\_pocetka, nv.datum\_zavrsetka,

nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza AS Prijeđeni\_Kilometri

FROM najam\_vozila nv

INNER JOIN vozilo v ON nv.id\_vozilo = v.id

WHERE nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza > 100;

Ovaj upit omogućava tvrtki provjeru vozila koja su prešla više kilometara nego što je očekivano tijekom najma. Evo detaljnog opisa koraka u izvršavanju upita:

**Odabir podataka:** Odabrala sam u SELECT djelu identifikator najma, registracijsku tablicu vozila, razliku datume početka i završetka najma koji izračunava prijeđene kilometre.

**Povezivanje tablica:** Zatim sam spojila tablice najam\_vozila i vozilo kako bi dobila potrebne informacije.

**Filtriranje rezultata:** nadalje koristim WHERE uvjet za filtriranje rezultata kako bi prikazala samo najmove vozila koji su prešli više od 100 kilometara.

**Upit 4**

Tvrtka želi analizirati svoje poslovne troškove kako bi bolje razumjela gdje se trošio novac tokom prvih 6 mjeseci poslovanja u 2024.godini i kako bi identificirala područja u kojima može uštedjeti ili optimizirati troškove.

**Zadatak:** Prikaži sve poslovne troškove za rezervacije koje su se dogodile u prvih 6 mjeseci u 2024.god

SELECT pt.\*

FROM poslovni\_trosak pt

JOIN transakcija t ON pt.id\_transakcija\_poslovnog\_troska = t.id

WHERE (MONTH(t.datum) BETWEEN 1 AND 6) AND YEAR(t.datum) = 2024;

**Odabir podataka:** Odabrala sam sve podatke o poslovnim troškovima, uzimam sve stupce iz tablice poslovni\_trosak

**Povezivanje tablica:** zatim sam spojila tablice poslovni\_trosak i transakcija kako bi dobila sve potrebne informacije preko zajedničkog stupca transakcija\_id.

**Filtriranje rezultata:** Ovdje koristim WHERE uvjet za filtriranje rezultata prema prvih 6 mjeseci u 2024. godini. Funkcija MONTH(t.datum) uzima mjesec iz datuma transakcije, a funkcija YEAR(t.datum) uzima godinu. Koristim BETWEEN operator kako bi provjerila je li mjesec u rasponu od 1 do 6 (prvih 6 mjeseci), a zatim provjeravam je li godina 2024.

Na kraju, dobivam sve poslovne troškove koji su se dogodili u prvih 6 mjeseci u 2024. godini.

**Upit 5**

Tvrtka proučava u kojim državama se nalaze tvrtke koje najviše koriste najam automobila. To joj može pomoći da shvati gdje najviše posluje i kako prilagoditi svoje usluge.

**Zadatak:** Pronađi broj najmova vozila po državi sjedišta pravne osobe

SELECT l.drzava AS Drzava, COUNT(nv.id) AS Broj\_Najmova

FROM pravna\_osoba p

INNER JOIN klijent k ON p.id = k.id\_pravna\_osoba

INNER JOIN transakcija t ON k.id = t.id\_klijent

INNER JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

INNER JOIN lokacija l ON p.id\_lokacija = l.id

WHERE YEAR(nv.datum\_pocetka) = 2024

GROUP BY l.drzava

ORDER BY Broj\_Najmova DESC;

**Odabir podataka:** Koristila sam u upitu podatke o nazivu države, broju najmova vozila po državi te prosječno trajanje najma.

**Povezivanje tablica:** Spajam tablice pravna\_osoba, klijent, transakcija, najam\_vozila i lokacija kako bi dobila potrebne informacije.

**Grupiranje podataka:** Grupirala sam rezultate po državi kako bih dobila broj najmova vozila za svaku državu.

**Sortiranje rezultata:** Rezultati se sortiraju silazno po broju najmova vozila.

**Upit 6**

Tvrtka istražuje koliko je novca svaki zaposlenik potrošio na najam automobila. To joj može pomoći da vidi tko ima najveće troškove u ovoj kategoriji i kako ih možda optimizirati.

**Zadatak:** Prikaži troškove najma vozila po zaposleniku

CREATE VIEW Ukupni\_Troskovi\_Najma\_Zaposlenika AS

SELECT z.id AS id\_zaposlenik, z.ime, z.prezime, SUM(t.iznos) AS ukupni\_troskovi

FROM zaposlenik z

INNER JOIN transakcija t ON z.id = t.id\_zaposlenik

INNER JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

GROUP BY z.id, z.ime, z.prezime;

SELECT ime, prezime, ukupni\_troskovi

FROM Ukupni\_Troskovi\_Najma\_Zaposlenika

ORDER BY ukupni\_troskovi DESC;

**Kreiranje pogleda za izračun ukupnih troškova najma vozila po zaposleniku:** Prvo sam kreirala pogled nazvan "Ukupni\_Troskovi\_Najma\_Zaposlenika" pomoću CREATE VIEW naredbe.U ovom pogledu koristila sam tri tablice: "zaposlenik" (z), "najam\_vozila" (nv) i "transakcija" (t).

Nadalje koristim se INNER JOIN kako bi se povezale tablice "zaposlenik" (z) i "najam\_vozila" (nv) prema atributu id\_zaposlenik. Zatim se povezuju tablice transakcija (t) i najam\_vozila (nv) prema atributu id\_transakcija\_najam. Rezultate sam grupirala po atributima z.id (identifikator zaposlenika), z.ime (ime zaposlenika) i z.prezime (prezime zaposlenika) kako bi se izračunala ukupne troškove najma vozila za svakog zaposlenika uz korištenje agregacije i funkcije SUM.

SUM(t.iznos) se koristi za izračun ukupnih troškova najma vozila za svakog zaposlenika.

**Korištenje pogleda u glavnom upitu:** Nakon što sam kreirala pogled, koristim rezultate tog pogleda u glavnom upitu.U SELECTU odabirem atributa koji će biti prikazani u rezultatima glavnog upita, u ovom slučaju to su ime, prezime i ukupni\_troskovi (ukupni troškovi najma vozila).

**Sortiranje rezultata:** Rezultati se sortiraju silazno po ukupnim troškovima najma vozila. Rezultati se sortiraju silazno prema ukupnim troškovima (ukupni\_troskovi) koristeći ORDER BY klauzulu s DESC opcijom, što znači da će zaposlenici s najvišim ukupnim troškovima biti prvi na popisu.

Ovaj upit omogućava prikaz ukupnih troškova najma vozila po zaposleniku, što može biti korisno za praćenje troškova najma vozila za svakog zaposlenika u tvrtki.

**Top of Form**

**Upit 7**

Tvrtka želi pratiti trajanja najma vozila kako bi bolje usmjerila marketinške kampanje i promocije. Analizom podataka o najmovima, tvrtka želi identificirati razdoblja kada su dulji najmovi rjeđi. Na temelju tih informacija, tvrtka može ponuditi posebne popuste kako bi potaknula klijente na dulje najmove tijekom tih razdoblja.

**Zadatak:** Prikaži trajanja najma vozila za klijente u 2024. godini od najdužih prema najkraćim najmovima

(SELECT k.ime, k.prezime,

nv.datum\_pocetka, nv.datum\_zavrsetka,

DATEDIFF(nv.datum\_zavrsetka, nv.datum\_pocetka) AS trajanje,

'Prvo tromjesečje 2024.' AS razdoblje

FROM najam\_vozila nv

INNER JOIN transakcija t ON nv.id\_transakcija\_najam = t.id

INNER JOIN klijent k ON t.id\_klijent = k.id

WHERE YEAR(nv.datum\_pocetka) = 2024 AND MONTH(nv.datum\_pocetka) <= 3)

UNION ALL

(SELECT k.ime, k.prezime,

nv.datum\_pocetka, nv.datum\_zavrsetka,

DATEDIFF(nv.datum\_zavrsetka, nv.datum\_pocetka) AS trajanje,

'Drugo tromjesečje 2024.' AS razdoblje

FROM najam\_vozila nv

INNER JOIN transakcija t ON nv.id\_transakcija\_najam = t.id

INNER JOIN klijent k ON t.id\_klijent = k.id

WHERE YEAR(nv.datum\_pocetka) = 2024 AND MONTH(nv.datum\_pocetka) > 3 AND MONTH(nv.datum\_pocetka) <= 6)

ORDER BY trajanje DESC;

Koristila sam UNION ALL operator kako bih kombinirala rezultate dva podupita.

**Odabir podataka**: Odabrala sam podatke o imenima i prezimenima klijenata, datumima početka i završetka najma vozila te trajanju najma. Također, koristim funkciju DATEDIFF kako bi izračunala trajanje najma vozila u danima. Dodajem i naziv razdoblja kako bi se razlikovali najmovi iz različitih tromjesečja.

**Unija podataka**: Unijom sam spojila rezultate iz dva dijela upita koji su se odnosili na prvo i drugo tromjesečje 2024. godine. Prvi dio upita izvlači podatke za prvo tromjesečje 2024. godine, dok drugi dio upita izvlači podatke za drugo tromjesečje 2024. godine.

**Filtriranje podataka**: Koristim WHERE uvjet kako bi se osigurala da se uzimaju u obzir samo podaci iz 2024. godine i iz određenog tromjesečja.

**Sortiranje podataka**: Upit završavam naredbom "ORDER BY" koja sortira rezultate prema trajanju najma vozila, od najdužih do najkraćih.

Ovaj upit omogućuje tvrtki da analizira trajanje najma vozila klijenata tijekom 2024. godine kako bi bolje usmjerila marketinške kampanje i promocije, posebno nudeći popuste tijekom razdoblja kada su dulji najmovi rjeđi.

## Upiti – Sebastijan

**Upit 1:** Koji je klijent u posljednjih mjesec dana najviše potrošio na najam svakoga od vozila (kamion, motocikl i automobil zasebno), a da pritom nije na crnoj listi ili iznajmljivao u ime pravne osobe, i koji je njegov kontakt? Tražimo upit sa ciljem da nagradimo te klijente.

(SELECT k.id, k.ime, k.prezime, ck.email, ck.broj\_mobitela, ck.broj\_telefona, SUM(t.iznos) AS ukupna\_potrosnja, "Kamion" AS tip\_vozila

FROM klijent k

JOIN transakcija t ON k.id = t.id\_klijent

JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

JOIN vozilo v ON nv.id\_vozilo = v.id

LEFT JOIN crna\_lista cl ON k.id = cl.id\_klijent

LEFT JOIN kontakt\_klijenta ck ON k.id = ck.id\_klijent

WHERE v.tip\_vozila = 'K'

AND t.datum BETWEEN DATE\_SUB(CURDATE(), INTERVAL 1 MONTH) AND CURDATE()

AND cl.id\_klijent IS NULL

AND k.id\_pravna\_osoba IS NULL

GROUP BY k.id, k.ime, k.prezime, ck.email, ck.broj\_mobitela, ck.broj\_telefona

ORDER BY ukupna\_potrosnja DESC

LIMIT 1)

UNION

(SELECT k.id, k.ime, k.prezime, ck.email, ck.broj\_mobitela, ck.broj\_telefona, SUM(t.iznos) AS ukupna\_potrosnja, "Automobil" AS tip\_vozila

FROM klijent k

JOIN transakcija t ON k.id = t.id\_klijent

JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

JOIN vozilo v ON nv.id\_vozilo = v.id

LEFT JOIN crna\_lista cl ON k.id = cl.id\_klijent

LEFT JOIN kontakt\_klijenta ck ON k.id = ck.id\_klijent

WHERE v.tip\_vozila = 'A'

AND t.datum BETWEEN DATE\_SUB(CURDATE(), INTERVAL 1 MONTH) AND CURDATE()

AND cl.id\_klijent IS NULL

AND k.id\_pravna\_osoba IS NULL

GROUP BY k.id, k.ime, k.prezime, ck.email, ck.broj\_mobitela, ck.broj\_telefona

ORDER BY ukupna\_potrosnja DESC

LIMIT 1)

UNION

(SELECT k.id, k.ime, k.prezime, ck.email, ck.broj\_mobitela, ck.broj\_telefona, SUM(t.iznos) AS ukupna\_potrosnja, "Motocikl" AS tip\_vozila

FROM klijent k

JOIN transakcija t ON k.id = t.id\_klijent

JOIN najam\_vozila nv ON t.id = nv.id\_transakcija\_najam

JOIN vozilo v ON nv.id\_vozilo = v.id

LEFT JOIN crna\_lista cl ON k.id = cl.id\_klijent

LEFT JOIN kontakt\_klijenta ck ON k.id = ck.id\_klijent

WHERE v.tip\_vozila = 'M'

AND t.datum BETWEEN DATE\_SUB(CURDATE(), INTERVAL 1 MONTH) AND CURDATE()

AND cl.id\_klijent IS NULL

AND k.id\_pravna\_osoba IS NULL

GROUP BY k.id, k.ime, k.prezime, ck.email, ck.broj\_mobitela, ck.broj\_telefona

ORDER BY ukupna\_potrosnja DESC

LIMIT 1);

**Upit 2:** Koja se serija automobila najviše rezervirala unutar zadnjih godinu dana? Istražujemo popularnost automobila

SELECT

serija.id AS id\_serija\_automobila,

serija.ime AS ime\_serije,

serija.proizvodac AS proizvodac\_serije,

COUNT(vozilo\_na\_rezervaciji.id) AS broj\_rezervacija

FROM

vozilo

JOIN

vozilo\_na\_rezervaciji ON vozilo.id = vozilo\_na\_rezervaciji.id\_vozilo

JOIN

serija ON vozilo.id\_serija = serija.id

JOIN

rezervacija ON vozilo\_na\_rezervaciji.id\_rezervacija = rezervacija.id

WHERE

vozilo.tip\_vozila = 'A'

AND rezervacija.datum\_rezervacije BETWEEN DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) AND CURDATE()

GROUP BY

serija.id

ORDER BY

broj\_rezervacija DESC

LIMIT 1;

**Upit 3:** Prikaži sve transakcije (troškove) po tipu transakcije od početka 2024 do danas

SELECT

tip\_transakcije,

SUM(iznos) AS ukupno\_potroseno

FROM (

SELECT

'Najam vozila' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

transakcija

JOIN

najam\_vozila ON transakcija.id = najam\_vozila.id\_transakcija\_najam

WHERE

YEAR(transakcija.datum) = 2024

UNION ALL

SELECT

'Održavanje' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

transakcija

JOIN

odrzavanje ON transakcija.id = odrzavanje.id\_transakcija\_odrzavanje

WHERE

YEAR(transakcija.datum) = 2024

UNION ALL

SELECT

'Prihod za zaposlenika' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

prihod\_za\_zaposlenika

JOIN

transakcija ON prihod\_za\_zaposlenika.id\_transakcija\_prihoda = transakcija.id

WHERE

YEAR(transakcija.datum) = 2024

UNION ALL

SELECT

'Punjenje' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

punjenje

JOIN

transakcija ON punjenje.id\_transakcija\_punjenje = transakcija.id

WHERE

YEAR(transakcija.datum) = 2024

UNION ALL

SELECT

'Naknada stete' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

naknada\_stete

JOIN

transakcija ON naknada\_stete.id\_transakcija = transakcija.id

WHERE

YEAR(transakcija.datum) = 2024

UNION ALL

SELECT

'Osiguranje' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

osiguranje

JOIN

transakcija ON osiguranje.id\_transakcija = transakcija.id

WHERE

YEAR(transakcija.datum) = 2024

UNION ALL

SELECT

'Poslovni trosak' AS tip\_transakcije,

SUM(iznos) AS iznos

FROM

poslovni\_trosak

JOIN

transakcija ON poslovni\_trosak.id\_transakcija\_poslovnog\_troska = transakcija.id

WHERE

YEAR(transakcija.datum) = 2024

) AS transakcije

GROUP BY

tip\_transakcije;

**Upit 4 :** Koliko imamo zaposlenika odjela "Prodaja" po gradu?

SELECT

l.grad,

COUNT(DISTINCT z.id) AS broj\_zaposlenika\_u\_prodaji

FROM

lokacija l

LEFT JOIN

zaposlenik z ON l.id = z.id\_lokacija

JOIN

zanimanje zan ON z.id\_zanimanje = zan.id

WHERE

zan.odjel = 'Prodaja'

GROUP BY

l.grad;

**Upit 5:** Prikazi tri vozila na kojima je najviše potrošeno na sveukupno osiguranje unutar zadnje tri godine.

SELECT v.id AS id\_vozila,

v.registracijska\_tablica AS registracijska\_tablica\_vozila,

SUM(t.iznos) AS ukupno\_potroseno\_na\_osiguranje

FROM vozilo v

JOIN osiguranje o ON v.id = o.id\_vozilo

JOIN transakcija t ON o.id\_transakcija = t.id

WHERE YEAR(t.datum) >= YEAR(CURDATE()) - 3

GROUP BY v.id, v.registracijska\_tablica

ORDER BY ukupno\_potroseno\_na\_osiguranje DESC

LIMIT 3;

**Upit 6:** Koliko je prosječno, maksimalno i minimalno potrošeno na godišnji servis unutar zadnjih pola godine?

SELECT

tip\_odrzavanja,

AVG(iznos) AS prosjecno\_potroseno,

MAX(iznos) AS maksimalno\_potroseno,

MIN(iznos) AS minimalno\_potroseno

FROM odrzavanje

JOIN transakcija ON odrzavanje.id\_transakcija\_odrzavanje = transakcija.id

WHERE datum >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH)

AND tip\_odrzavanja = 'Godišnji servis';

**Upit 7:** Koliko je ukupno potrošeno putem kartičnih plaćanja, po tipu kartice, unutar zadnje dvije godine?

SELECT

tip\_kartice,

SUM(t.iznos) AS ukupno\_potroseno

FROM

karticno\_placanje kp

JOIN

transakcija t ON kp.id = t.id

WHERE

t.datum >= DATE\_SUB(CURDATE(), INTERVAL 2 YEAR)

GROUP BY

tip\_kartice

ORDER BY

ukupno\_potroseno DESC;

**Upit 8:** Koliko je ukupno potrošeno na punjenje, sortirano po tipu punjenja, unutar zadnjih godinu dana?

SELECT

tip\_punjenja,

SUM(t.iznos) AS ukupno\_potroseno

FROM

punjenje p

JOIN

transakcija t ON p.id\_transakcija\_punjenje = t.id

WHERE

t.datum >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

tip\_punjenja

ORDER BY

ukupno\_potroseno DESC;

## Upiti – Vedrana

**Upit 1**

Održavatelj vozila tvrtke dolazi do računovođe s pitanjem o isplativosti pojedinih vozila te želi saznati koja vozila su najviše korištena u određenom razdoblju, kako bi mogao donijeti odluku o eventualnim ulaganjima u nova vozila ili zamjeni manje isplativih vozila. Za ovu analizu potrebno je prikupiti podatke o kilometraži koju su vozila prešla tijekom najma, kao i o datumima početka i završetka najma.

**Zadatak:** Prikaži pet vozila koja su napravila najveću kilometražu, period kad su bili iznajmljeni. Rezultate grupirati po vozilu, a vrijednost kilometraže sortirati silazno.

**Kod za upit:**

SELECT

nv.id\_vozilo AS 'ID vozila',

SUM(nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza) AS 'Kilometraža',

MIN(nv.datum\_pocetka) AS 'Datum početka najma',

MAX(nv.datum\_zavrsetka) AS 'Datum završetka najma'

FROM

najam\_vozila nv

GROUP BY

nv.id\_vozilo

ORDER BY

SUM(nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza) DESC

LIMIT 5;

**Opis upita:**

* Odabrani stupci za prikaz su:
  + ID Vozila
  + Ovdje su korištene agregirane funkcije SUM, MIN, MAX kako bi se dobio rezultat za sve iznajmljene automobile
  + Kilometraža SUM – izračunava ukupnu prijeđenu kilometražu za svako vozilo tako što zbraja razlike između završne i početne kilometraže za svaki najam
  + Datum početka najma - MIN – najraniji datum početka najma za svako vozilo
  + Datum završetka najma MAX – najkasniji datum završetka najma
* Koristimo aliase radi lakšeg pisanja i čitanja upita.
* GROUP BY -grupira rezultate prema id\_vozila, agregatne funkcije (SUM, MIN i MAX) primjenjuju se na svaki skup podataka koji pripada istom vozilu
* ORDER BY SUM – sortira rezultate prema ukupnoj prijeđenoj kilometraži u DESC padajućem redoslijedu (vozila s najvećom kilometražom prva)
* LIMIT 5 – ograničenje da broj rezultata bude 5, prikazuje top 5 vozila s najvećom prijeđenom kilometražom

**Rješenje:**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, broj

Opis je automatski generiran

**Upit 2**

Marketinški stručnjak koji radi u tvrtki za iznajmljivanje vozila želi pratiti klijente koji su rezervirali vozila u turističkoj sezoni. S obzirom na to da je turistička sezona ključna za poslovanje, direktor želi dobiti uvid u podatke o klijentima koji su izvršili rezervacije u tijeku turističke sezone. Posebno ga zanima serija i tip vozila koja su rezervirana, kako bi mogao analizirati potražnju za određenim vrstama vozila.

**Zadatak:** Prikaži sve klijente koji su rezervirali automobil u periodu od 01. travnja do 31. listopada 2023. godine, s podacima o seriji i tipu vozila, podaci poredani po datumu uzlazno.

**Kod za upit:**

SELECT

k.id AS 'ID klijenta',

k.ime AS 'Ime',

k.prezime AS 'Prezime',

k.identifikacijski\_broj AS 'Identifikacijski broj',

r.datum\_rezervacije AS 'Datum rezervacije',

s.ime AS 'Serija vozila',

v.tip\_vozila AS 'Tip vozila'

FROM

rezervacija r

JOIN

klijent k ON r.id\_klijent = k.id

JOIN

vozilo\_na\_rezervaciji vr ON r.id = vr.id\_rezervacija

JOIN

vozilo v ON vr.id\_vozilo = v.id

JOIN

serija s ON v.id\_serija = s.id

WHERE

r.datum\_rezervacije BETWEEN '2023-04-01' AND '2023-10-31' AND v.tip\_vozila = 'A'

ORDER BY

r.datum\_rezervacije ASC;

**Opis upita:**

* Odabrani stupci za prikaz:
* Id klijenta
* Ime
* Prezime
* Identifikacijski broj
* Datum rezervacije
* serija vozila
* Tip vozila
* 4 INNER JOIN funkcije - povezuju one redove koji imaju odgovarajuće vrijednosti u obje tablice
  + spaja tablicu rezervacija s tablicom klijent gdje je id\_klijent = id kako bi se došlo do podataka o klijentima
  + spaja tablicu rezervacija s tablicom vozilo na rezervaciji gdje je id\_rezervacija = id kako bi se dobili podaci o vozilima na rezervaciji
  + JOIN spaja tablicu vozilo na rezervaciji s tablicom vozilo gdje je id\_vozilo = id kako bi se dobili podaci o vozilu
  + JOIN spaja tablicu vozilo s tablicom serija gdje je id serija = id kako bi se dobili podaci o seriji vozila
* Podaci o rezervacijama su filtrirani za period od 01.04. do 31.10.
* Filtrira rezultate kako bi prikazao samo rezervacije za vozila tipa A – odnosno automobile
* Sortiranje podataka prema datumu rezervacije uzlazno ASC – uzlazno od najstarije do najnovijeg

**Rješenje:**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, broj

Opis je automatski generiran

**Upit 3**

Direktor tvrtke želi dobiti detaljan uvid u upotrebu električnih vozila u tvrtki. S obzirom na rastući interes za ekološki prihvatljivijim opcijama, direktor želi analizirati koliko često se električna vozila iznajmljuju i rezerviraju, te koliki je broj dana koje provode u najmu. Ovi podaci će pomoći direktoru da donese odluke o budućim ulaganjima u električna vozila i optimizaciji njihove dostupnosti.

**Zadatak:** Prikaži vozila koja imaju električno punjenje, broj njihovih najmova i broj rezervacija, datume početka i završetka najma te podatak o broju dana najma.

**Kod za upit:**

SELECT

v.id AS 'ID vozila', s.ime AS 'Naziv serije', v.tip\_punjenja AS 'Tip punjenja',

COUNT(nv.id) AS 'Broj najmova',

(

SELECT COUNT(r.id)

FROM rezervacija r

JOIN vozilo\_na\_rezervaciji vr ON r.id = vr.id\_rezervacija

WHERE vr.id\_vozilo = v.id

) AS 'Broj rezervacija',

(

SELECT GROUP\_CONCAT(DISTINCT datum\_pocetka SEPARATOR ', ')

FROM najam\_vozila

WHERE id\_vozilo = v.id

) AS 'Datum početka najma',

(

SELECT GROUP\_CONCAT(DISTINCT datum\_zavrsetka SEPARATOR ', ')

FROM najam\_vozila WHERE id\_vozilo = v.id

) AS 'Datum završetka najma',

(

SELECT SUM(DATEDIFF(datum\_zavrsetka, datum\_pocetka)) FROM najam\_vozila

WHERE id\_vozilo = v.id

) AS 'Broj dana najma'

FROM vozilo v

JOIN serija s ON v.id\_serija = s.id

LEFT JOIN najam\_vozila nv ON v.id = nv.id\_vozilo

WHERE v.tip\_punjenja = 'Električno'

GROUP BY v.id, s.ime;

**Opis upita:**

* Odabrani stupci za prikaz:
* Id vozila
* Naziv serije
* Tip punjenja
* Broj najmova
* Broj rezervacija
* Datum početka najma
* Datum završetka najma
* Broj dana najma
* Podupit 1: Prebrojava rezervacije vozila
* Podupit 2: grupira sve datume početka najma vozila
* DISTINCT funkcija osigurava da se svaki datum pojavi samo jednom te se otklanja mogućnost dupliciranih datuma.
* CONCAT – sve prikupljene jedinstvene datume spaja u jedan niz, odvojen zarezom
* Podupit 3: grupira sve datume završetka najma vozila
* Podupit 4: računa ukupan broj dana najma vozila
* DATEDIFF – rezultat je broj dana između datuma 1 i datuma 2
* JOIN – spaja tablicu serija s tablicom vozilo preko id serija kako bi se dobili podaci o seriji vozila
* LEFT JOIN – lijevi vanjski spoj koji tablicu vozilo spaja s tablicom najam vozila na osnovu id iz tablice vozilo i id vozilo iz tablice najam vozila
* WHERE - Filtriranje podataka kako bi se prikazala samo električna vozila
* Podaci su grupirani prema id vozila i nazivu vozila

**Rješenje:**

Slika na kojoj se prikazuje tekst, snimka zaslona, broj, Font

Opis je automatski generiran

**Upit 4**

Marketinški direktor tvrtke za iznajmljivanje vozila želi dobiti detaljan uvid u slučajeve krađe vozila unutar tvrtke. Cilj je analizirati vozila koja su imala štetu zbog krađe, pregledati opis štete, trajanje naknade štete i iznose transakcija povezanih s naknadom štete. Ovi podaci su važni za procjenu sigurnosnih rizika, optimizaciju osiguravateljskih politika i planiranje budućih preventivnih mjera.

**Zadatak:** Prikaži sva vozila koja su imala štetu, ali da je tip štete krađa. Također prikaži datum početka i završetka nastanka štete te iznosa koji je pokrio štetu, sortiran po iznosu transakcije silazno.

**Kod za upit**

SELECT

vozilo.id AS 'ID vozila',

serija.ime AS 'Naziv vozila',

steta.tip AS 'Tip štete',

steta.opis AS 'Opis štete',

naknada\_stete.id AS 'ID naknade štete',

naknada\_stete.datum\_pocetka AS 'Datum početka',

naknada\_stete.datum\_zavrsetka AS 'Datum završetka',

transakcija.iznos AS 'Iznos transakcije'

FROM

(SELECT DISTINCT id, id\_serija FROM vozilo) vozilo

JOIN

najam\_vozila ON vozilo.id = najam\_vozila.id\_vozilo

JOIN

naknada\_stete ON najam\_vozila.id = naknada\_stete.id\_transakcija

JOIN

steta ON naknada\_stete.id\_steta = steta.id

JOIN

serija ON vozilo.id\_serija = serija.id

JOIN

transakcija ON naknada\_stete.id\_transakcija = transakcija.id

WHERE

steta.tip = 'Krađa'

ORDER BY

transakcija.iznos DESC;

**Opis upita:**

* Odabir stupaca:
* Id vozila
* Naziv vozila
* Tip štete
* Opis štete
* Id naknada štete
* Datum početka
* Datum završetka
* Iznos transakcije
* FROM – ovdje funkcija FROM koristi SELECT DISTINCT – podupit koji nam definira privremenu (izvedenu) tablicu koja će se kasnije koristiti u glavnom upitu, a rezultat je skup jedinstvenih kombinacija vrijednosti id i id serija iz tablice vozilo
* PET INNER JOIN funkcija – povezuju one redove koji imaju odgovarajuće vrijednosti u obje tablice
* Povezuje tablicu najam vozila s izvedenom tablicom vozilo
* Povezuje tablicu najam vozila s tablicom naknada štete
* Povezuje tablicu naknada štete i šteta
* Povezuje tablicu vozilo s tablicom serija
* Povezuje tablicu naknada štete s tablicom transakcija
* WHERE – uvjet koji filtrira rezultat tako da uključi samo one redove gdje je tip štete krađa
* ORDER by – sortira rezultate prema iznosu transakcije u silaznom redoslijedu DESC

**Rješenje:**

Slika na kojoj se prikazuje tekst, Font, broj, crta

Opis je automatski generiran

**Upit 5**

Vlasnik tvrtke za iznajmljivanje vozila želi prepoznati zaposlenike koji su ostvarili najveći prihod u zadnjoj godini dana. Cilj je identificirati najprofitabilnije zaposlenike, odnosno one koji ostvaruju najviše transakcija i najmova. Ovi podaci su važni za nagrađivanje zaposlenika, planiranje treninga, te optimizaciju poslovnih strategija.

**Zadatak:** Prikaži 5 zaposlenika koji su ostvarili najveći prihod u zadnjoj godini dana, sortiraj po ukupnom prihodu silazno. Prikaži i dodatne stupce s brojem transakcija i najmova koje su ostvarili.

**Kod za upit**

SELECT

z.id AS zaposlenik\_id,

z.ime AS zaposlenik\_ime,

z.prezime AS zaposlenik\_prezime,

z.email AS zaposlenik\_email,

l.drzava AS lokacija država,

l.grad AS lokacija\_grad,

l.adresa AS lokacija\_adresa,

SUM(t.iznos) AS ukupni\_prihod,

COUNT(t.id) AS broj\_transakcija,

COUNT(DISTINCT t.id) AS broj\_najmova

FROM

zaposlenik z

JOIN

transakcija t ON z.id = t.id\_zaposlenik

JOIN

prihod\_za\_zaposlenika pz ON t.id = pz.id\_transakcija\_prihoda

JOIN

prihod p ON pz.id\_prihod = p.id

JOIN

lokacija l ON z.id\_lokacija = l.id

WHERE

t.datum >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

z.id, z.ime, z.prezime, z.email, l.drzava, l.grad, l.adresa

ORDER BY

ukupni\_prihod DESC

LIMIT 5;

**Opis upita:**

* Odabrani stupci za prikaz: selektirani stupci koji će se vratiti, s aliasima za lakše čitanje i pisanje koda
* FROM – funkcija koja nam pokazuje da će se sve druge tablice spajati s tablicom zaposlenik
* 4 INNER JOINA - povezuju one redove koji imaju odgovarajuće vrijednosti u obje tablice
  + Povezuje relaciju zaposlenika s tablicom transakcija na temelju id u tablici zaposlenik i id\_zaposlenik u relaciji transakcija
  + Povezuje transakciju s relacijom prihod za zaposlenika s tablicom na temelju id iz relacije transakcija i id\_transakcija prihoda u relaciji prihod za zaposlenika
  + Povezuje prihod za zaposlenika s relacijom prihod na temelju id\_prihod iz relacije prihod za zaposlenika i id iz relacije prihod
  + Povezuje relaciju zaposlenik s relacijom lokacija na temelju id\_lokacija iz relacije zaposlenik i id u relaciji lokacija
* WHERE – uvjet koji filtrira transakcije da bi se prikazale samo one koje su se dogodile u zadnjih godinu dana / CURDATE – trenutni datum sustava
* GROUP BY – grupira rezultate po jedinstvenim kombinacijama zaposlenika
* ORDER BY – sortiranje rezultate po DESC – silaznom redoslijedu
* LIMIT 5 – ograničava rezultat na prvih pet zaposlenika

**Rješenje**



**Upit 6**

Direktor računovodstva za iznajmljivanje vozila želi analizirati koliko klijenti koriste kriptovalute za plaćanje. Cilj je dobiti uvid u popularnost kriptovalutnih transakcija među klijentima kako bi tvrtka mogla prilagoditi svoje usluge i ponuditi dodatne pogodnosti za kripto korisnike.

**Zadatak:** Prikaži sva plaćanja izvršena kriptovalutnim načinom plaćanja, datum kad su izvršena i od strane kojeg klijenta. Sortiraj po datumu silazno.

**Kod za upit:**

SELECT

t.id,

t.datum,

t.iznos,

t.broj\_racuna,

t.placeno,

t.id\_klijent,

t.id\_zaposlenik,

k.kriptovaluta,

k.broj\_kripto\_novcanika,

CONCAT(kl.ime,' ', kl.prezime) AS puno\_ime\_klijenta

FROM

transakcija t

JOIN

kriptovalutno\_placanje k ON t.id = k.id

JOIN

klijent kl ON t.id\_klijent = kl.id

ORDER BY

t.datum DESC;

**Opis upita:**

* Selektirani su podatci iz navedenih tablica koje upit treba vratiti (korištenje aliasa za lakše čitanje)
* FROM navodi da će se podaci dohvatiti iz tablice transakcija gdje su id iz tablice transakcija i id iz tablice klijent jednaki
* 2 INNER JOINA - - povezuju one redove koji imaju odgovarajuće vrijednosti u obje tablice
* Povezuje tablicu transakcija s tablicom kriptovalutno plaćanje gdje je id transakcija jednak id kriptovalutno placanje
* Povezuje tablicu transakcija s tablicom klijent gdje je id klijent u transakciji jednak id klijent
* ORDER BY – sortira rezultate po datumu transakcije u silaznom DESC redoslijedu prikazujući najnovije transakcije prve

**Rješenje:**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, broj

Opis je automatski generiran

**Upit 7**

Marketing menadžer tvrtke za iznajmljivanje vozila želi analizirati kupce koji najčešće koriste popuste. Cilj je identificirati najlojalnije kupce, razumjeti koliko često koriste popuste i planirati buduće promocije temeljene na ovim podacima.

**Zadatak:** Prikaži klijente koji su ostvarili najveći broj popusta, i posebno tablicu u kojoj je vidljivo koliko je aktivnih, a koliko neaktivnih popusta

**Kod za upit:**

SELECT

CONCAT(k.ime, ' ', k.prezime) AS klijent,

COUNT(pk.id) AS broj\_popusta,

SUM(CASE WHEN pk.status = 'aktivan' THEN 1 ELSE 0 END) AS broj\_aktivnih\_popusta,

SUM(CASE WHEN pk.status = 'iskorišten' THEN 1 ELSE 0 END) AS broj\_iskorištenih\_popusta

FROM

klijent k

JOIN

popust\_za\_klijenta pk ON k.id = pk.id\_klijent

GROUP BY

k.ime, k.prezime

ORDER BY

broj\_popusta DESC

LIMIT 5;

**Opis upita:**

* SELECT – specificira stupce koje će upit vratiti (ime, prezime, broj popusta, broj aktivnih i broj neaktivnih popusta) CONCAT – funkcija koja spaja ime i prezime u jedan string
* CASE WHEN – slučaj kad je status aktivan ili neaktivan
* FROM – navodi da će se podaci dohvatiti iz tablice klijent
* INNER JOIN – spaja relacije popust za klijenta i klijent preko id klijent
* GROUP BY – grupirano po imenu i prezimenu
* ORDER BY – sortirano po broju popusta silazno – od najvećeg prema manjem
* LIMIT 5 – postavljeno ograničenje na 5 klijenata

**Rješenje:**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, crta

Opis je automatski generiran

**Upit 8**

Rukovoditelj ljudskih resursa tvrtke za iznajmljivanje vozila želi analizirati radne uvjete ženskih zaposlenica, s ciljem čestitanja Međunarodnog dana žena i osiguravanja bolje ravnoteže između posla i privatnog života. Cilj je prikupiti podatke o zaposlenicama, njihovim radnim mjestima i broju transakcija koje su obavile u zadnjih 6 mjeseci.

**Zadatak:** Prikaži sve zaposlenice ženskog spola, države u kojoj rade, njihova zanimanja, e-mail i broj transakcija u zadnjih 6 mjeseci. Sve sortirano po imenu zaposlenica.

**Kod za upit:**

CREATE VIEW v\_podaci\_zaposlenika AS

SELECT z.ime, z.prezime, l.drzava, zanimanje.naziv AS zanimanje,

COUNT(t.id) AS broj\_transakcija, z.email

FROM

zaposlenik z

JOIN

zanimanje ON z.id\_zanimanje = zanimanje.id

JOIN

lokacija l ON z.id\_lokacija = l.id

LEFT JOIN

transakcija t ON z.id = t.id\_zaposlenik AND t.datum >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH)

WHERE

z.spol = 'F'

GROUP BY

z.ime, z.prezime, l.drzava, zanimanje.naziv, z.email

ORDER BY

z.ime ASC;

SELECT \* FROM v\_podaci\_zaposlenika;

**Opis upita:**

* Kroz CREATE VIEW 'podaci\_zaposlenika' – virtualna tablica koja uzima podatke o zaposlenicima koji me zanimaju (ime, prezime, drzava, zanimanje.naziv i email
* SELECT: specificira stupce koje će upit vratiti, COUNT – računa broj transakcija koje je zaposlenik izvršio u zadnjih 6 mjeseca
* FROM – navodi da će se podaci dohvatiti iz tablice zaposlenik
* 2 INNER JOINA –
  + Spaja tablice zaposlenik sa zanimanjem i lokacijom i vraća vrijednosti koje seu obje tablice koje se spajaju
* LEFT JOIN
  + spaja tablicu zaposlenik s tablicom transakcija na način da uzima podatke iz tablice zaposlenik i spaja s odgovarajućim podacima iz tablice transakcija, a ako nema podataka vraća NULL vrijednost
  + CURDATE – trenutni datum sustava,
  + DATE\_SUB(CURDATE(), INTERVAL 6 MONTH) - od trenutnog datuma se oduzima 6 mjeseci
* WHERE – postavlja uvjet da se rezultat ograniči samo na zaposlenice ženskog spola
* GROUP BY – po imenu, prezimenu, država, zanimanje.naziv i email
* ORDER BY – sortira po imenu uzlaznim redoslijedom ASC

**Rješenje:**

Slika na kojoj se prikazuje tekst, snimka zaslona, broj, Font

Opis je automatski generiran

# Korišteni alati i metode

|  |  |
| --- | --- |
| WhatsApp | * brza razmjena poruka, dokumenata i fotografija |
| Zoom | * održavanje virtualnih sastanaka * suradnja * dogovori oko zadataka |
| GitHub | * verzioniranje i dijeljenje koda * praćenje zadataka i promjena u kodu |
| MySQL Workbench | * stvaranje baze podataka * izrada EER dijagrama * pisanje i testiranje koda * stvaranje upita |
| Notepad++ | * generiranje, uređivanje, prikaz i provjera podataka |
| ChatGPT | * generiranje vrijednosti i pronalaženje grešaka |
| Lucidchart | * izrada ER dijagrama |

# ZAKLJUČAK

Početna ideja ovog projekta bila je stvaranje baze podataka koja će sadržavati 10 relacija što je bilo dovoljno za praćenje osnovnih segmenata poslovanje naše tvrtke. Međutim, kad smo počeli malo intenzivnije razmišljati o potrebama korisnika i cijelom procesu iznajmljivanja, shvatili smo da je za efikasno upravljanje ovakvom tvrtkom potrebno puno više detalja. Stoga je baza podataka bitno proširena i sada obuhvaća 32 relacije.

Proširivanjem baze podataka postigli smo nekoliko bitnih prednosti:

* detaljniju evidenciju svih aspekata poslovanja (različiti tipovi vozila, različiti načini plaćanja, informacije o klijentima i zaposlenicima),
* poboljšali smo efikasnost tvrtke (prati se najam vozila, obračuni plaća, upravljanje popustima),
* s više relacija podaci su bolje organizirani i točniji (pouzdanost baze),
* proširena baza omogućava lakše prilagođavanje budućim potrebama (ako se ukaže potreba za dodatnim informacijama ili novim poslovnim procesima, baza je dovoljno fleksibilna i lako se može proširiti).

Dok smo stvarali ovu proširenu bazu podataka, naišli smo na dosta problema i izazova. U početku nam je najveći problem predstavljalo povezivanje velikog broja relacija što je zahtijevalo složene odnose između tablica, ali i stvaranje novih tablica. Sljedeći problem na koji smo naišli je konstantno ispravljanje grešaka, teškoće u usklađivanju dijelova projekta i uspostavljanje jasne komunikacije. Iako je kontinuirano ispravljanje grešaka ključno za uspješan razvoj baze podataka mi smo se često našli u problemu jer smo analizirali gotovo svaku grešku kako bismo poboljšali našu bazu kroz seriju manjih koraka i ažuriranja.

Ukupno gledano, cijeli ovaj proces izrade projekta, ne samo da je utjecao na to da ova baza bude u skladu s našim idejama i potrebama korisnika, već je utjecao na cijeli naš tim i način na koji radimo zajedno.

Unatoč svim izazovima i naporima, svjesni činjenice da je za izradu ovako zahtjevnog projekta bio potrebno puno više vremena da se 'izbrusi' do najsitnijih detalja, ipak vjerujemo da smo izradili sustav koji je pouzdan, robustan i učinkovit i koji će pouzdano podržavati poslovne procese naše tvrtke. Uvjereni smo da će naša baza podataka ostati relevantna i konkurenta i u budućnosti te da će uspješno odgovoriti zahtjevima modernog poslovanja. Također smo uvjereni da se i nadalje može nastaviti ulagati u njen razvoj i nadogradnju dodavanjem novih funkcionalnosti (korištenje Cloud usluga, rad s velikim količinama podataka - Big Data, upotreba naprednih algoritama strojnog učenja i umjetne inteligencije), kako bi odgovorila zahtjevima tržišta i ostala u skladu s najnovijim trendovima i inovacijama u području baza podataka.