Kolegij:

Baze podataka I.

Projekt:

Sustav za upravljanje knjižnicom

Autor:

Daniel Katić

JMBAG 0303123347

# Uvod

## Kontekst i motivacija

Knjižnica godišnje obrađuje više od 12000 fizičkih posudbi, te konstantno raste broj korisnika. Trenutačno se podaci o zaduženjima i vraćanjima vode u Excel tablicama i u papirnatim evidencijama, što otežava provjeru dostupnosti knjiga, kao i razna praćenja statistika, poput naplata kazni za kašnjenje, ili praćenje popularnost određenih naslova. Uz to, nedostaje pouzdana analitika koja bi pomogla nabavi pri odabiru novih knjiga i planiranju budžeta.

Iz tih je razloga odlučeno razviti cjeloviti sustav upravljanja knjižnicom baziran na relacijskoj bazi podataka i naprednim SQL upitima. Prelaskom na digitalni sustav baze podataka riješiti će se navedeni problemi i otvoriti će se mogućnost napredne analitike.

## Minimalni funkcionalni zahtjevi

1. Sustav mora omogućiti unos, ažuriranje i brisanje zapisa o članovima knjižnice.
2. Sustav mora održavati katalog izdavača i njihove osnovne kontakte.
3. Moraju se evidentirati sve posudbe i povrati knjiga, uz automatsko praćenje roka vraćanja.
4. Mora se obračunavati kazna od 0,50 € po danu kašnjenja te voditi ukupni prihod od kazni.
5. Sustav mora omogućiti registraciju i upravljanje zaposlenicima koji izdaju ili zaprimaju knjige.
6. Potrebno je generirati izvještaje i analitičke upite (Top liste, postotak kašnjenja, trend rasta članova, i tako dalje).
7. Svi podaci moraju biti konzistentni putem primarnih i stranih ključeva te ograničenja (NOT NULL, CHECK).

## Cilj mog modula

U timu sam zadužen za dizajn i implementaciju četiriju ključnih tablica – Izdavači, Članovi, Zaposlenici i Posudbe – te niza naprednih upita koji nad njima izvode poslovno korisne analize.

Modul omogućuje:

* Centralizirano čuvanje podataka o nakladnicima i korisnicima.
* Praćenje svake transakcije posudbe od izdavanja do povrata.
* Automatski obračun i izvještavanje o kaznama.
* Statistiku popularnosti izdavača, učestalost kašnjenja po grupama korisnika, vrijeme do posudbe i druge metrike koje pomažu nabavi, marketingu i vodstvu knjižnice.

## Tehnologije

* MySQL – relacijski sustav za upravljanje bazom podataka.
* MySQL Workbench – modeliranje sheme (ER-dijagram) i izvršavanje skripti.
* Visual Studio Code + proširenje SQL Tools – pisanje koda.
* Git + GitHub – timska kontrola inačica i kontinuirana integracija.
* Microsoft Word – pisanje dokumentacije.

## Struktura dokumenta

U nastavku rada najprije prikazujem konceptualni ER-dijagram i logičku shemu baze s objašnjenjem svih ograničenja.

Slijedi detaljan opis tablica i atributa, nakon čega iznosim poslovna pravila koja su implementirana u bazi.

Središnji dio čini pregled SQL upita s objašnjenjem rezultata.

Dokument završava zaključkom u kojem navodim ostvarene ciljeve i prijedloge za buduća unaprjeđenja sustava.

# Opis poslovnog procesa

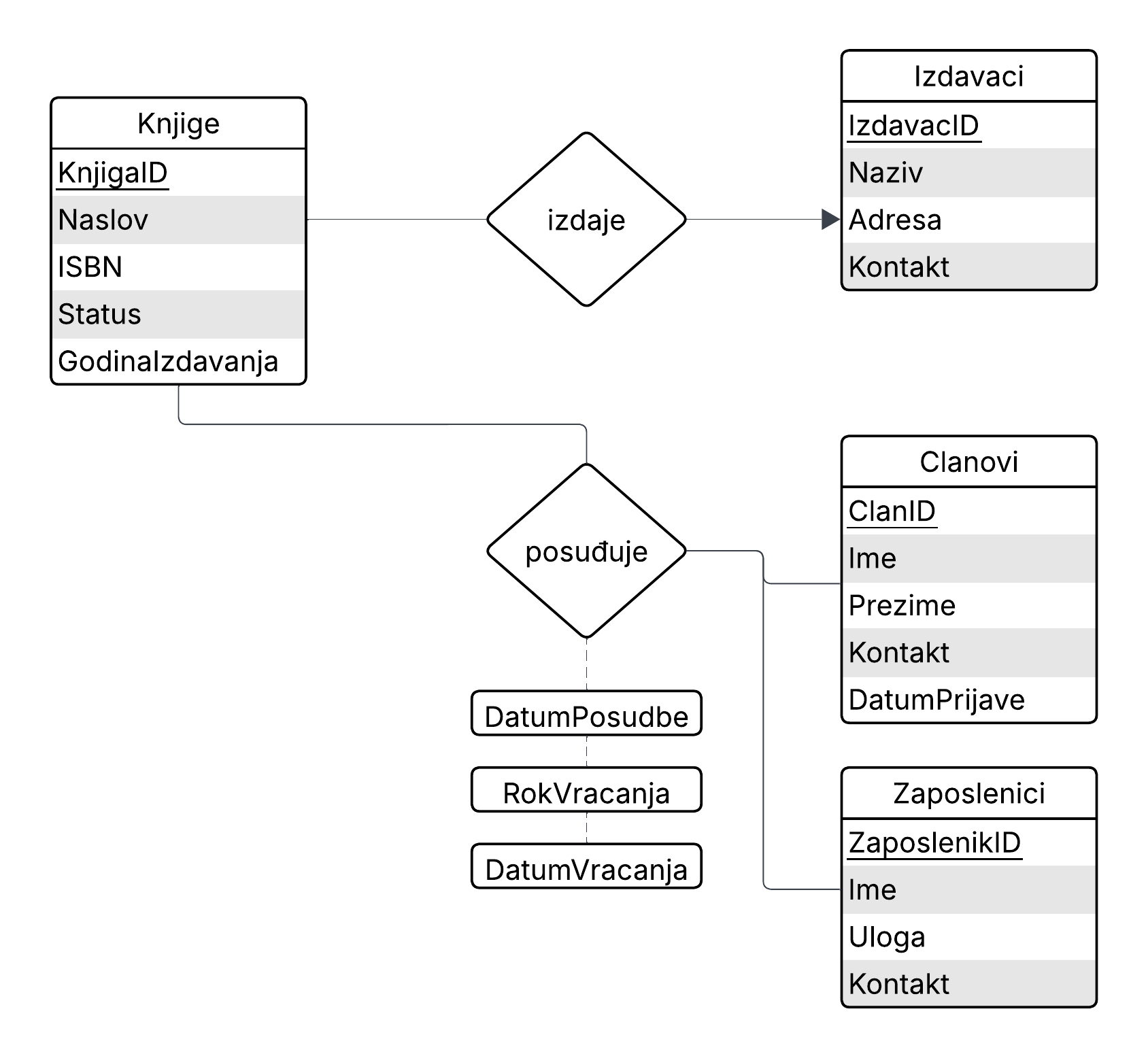
## Poslovni proces knjižnice – „život jedne knjige“

|  |  |  |
| --- | --- | --- |
| Faza | Akteri | Kratki tok |
| 1 | Katalogizacija | Knjižničar unosi osnove podatke o naslovu i povezuje ga s postojećim Izdavačem. Ako izdavač ne postoji, najprije ga kreira. |
| 2 | Učlanjenje | Član popunjava obrazac, djelatnik mu dodjeljuje ClanID. Datum prijave bilježimo radi metrika onboarding-a. |
| 3 | Posudba | Član odabire knjigu, djelatnik otvara zapisu Posudbe gdje se bilježi DatumPosudbe i RokVracanja. |
| 4 | Povrat / Kašnjenje | Knjiga se vraća, zapis se ažurira poljem DatumVracanja. Ako je rok prekoračen, sustav obračunava kaznu (0,50 €/dan). |
| 5 | Analitika | Na temelju podataka o posudbama izvode se upiti: Top 5 izdavača, postotak kašnjenja, vrijeme na polici, i tako dalje. |

Ključni poslovi zahtjevi koji proizlaze iz toka:

* Svaka posudba mora referencirati točno jednog člana, jednog zaposlenika i jedan primjerak knjige.
* Knjiga mora imati točno jednog izdavača.
* Član može imati najviše N aktivnih posudbi (ograničenje implementirano poslovnim pravilom).
* Kazna se računa samo ako CURDATE() > RokVracanja.

# ER dijagram



**Relacijski model:**

Knjige (KnjigaID, Naslov, ISBN, Status, *IzdavacID*)  
Izdavaci (IzdavacID, Naziv, Adresa, Kontakt)   
Clanovi (ClanID, Ime, Prezime, Kontakt, DatumPrijave)  
Zaposlenici (ZaposlenikID, Ime, Uloga, Kontakt)  
Posuđuje (KnjigaID, ClanID, ZaposlenikID, DatumPosudbe, RokVracanja, DatumVracanja)

# EER Dijagram

* EER dijagram baze podataka

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

Sadržaj generiran umjetnom inteligencijom može biti netočan.

# Detaljan opis tablica, atributa i domena s komentarima

**Tablica Izdavaci**

|  |  |  |
| --- | --- | --- |
| Atribut | Domena/Ograničenja | Komentar |
| IzdavacID | INT PK, AUTO\_INCREMENT, NOT NULL | Jedinstveni ID nakladnika. |
| Naziv | VARCHAR(255) NOT NULL | Puni naziv izdavačke kuće. |
| Adresa | VARCHAR(255) | Poštanska adresa sjedišta. |
| Kontakt | VARCHAR(100) | Opći tel. ili e-mail za nabavu. |

**Tablica Clanovi**

|  |  |  |
| --- | --- | --- |
| Atribut | Domena/Ograničenja | Komentar |
| ClanID | INT PK, AUTO\_INCREMENT, NOT NULL | ID člana. |
| Ime | VARCHAR(100) NOT NULL | Ime člana. |
| Prezime | VARCHAR(100) NOT NULL | Prezime člana. |
| Kontakt | VARCHAR(100) | Telefon ili e-mail člana. |
| DatumPrijave | DATE NULL | Dan učlanjenja. |

**Tablica Zaposlenici**

|  |  |  |
| --- | --- | --- |
| Atribut | Domena/Ograničenja | Komentar |
| ZaposlenikID | INT PK, AUTO\_INCREMENT, NOT NULL | ID zaposlenika. |
| Ime | VARCHAR(100) NOT NULL | Ime knjižničara / administratora. |
| Uloga | VARCHAR(100) | Tekstualni opis uloge (npr. knjižničar, voditelj, administrator). |
| Kontakt | VARCHAR(100) | Telefon ili email. |

**Tablica Posudbe**

|  |  |  |
| --- | --- | --- |
| Atribut | Domena/Ograničenja | Komentar |
| PosudbaID | INT PK, AUTO\_INCREMENT, NOT NULL | ID posudbe. |
| ClanID | INT FK NOT NUL -> Clanovi(ClanID) | Tko je posudio knjigu. |
| KnjigaID | INT FK NOT NULL -> Knjige(KnjigaID) | Koja je knjiga posuđena. |
| DatumPosudbe | DATE NOT NULL | Datum izdavanja knjige. |
| RokVracanja | DATE NOT NULL | Predviđeni povrat (npr. +21 dan). |
| DatumVracanja | DATE NULL | Popunjava se pri vraćanju, NULL znači da je još uvijek zaduženo. |
| ZaposlenikID | INT FK NOT NULL -> Zaposlenici(ZaposlenikID) | Djelatnik koji je obradio posudbu/povrat. |

# Poslovna pravila i ograničenja baze

## Primarni i strani ključevi

|  |  |  |
| --- | --- | --- |
| Pravilo | Objašnjenje | Implementacija |
| Jednoznačna identifikacija | Svaka tablica ima sintetski PK (AUTO\_INCREMENT) pa ni jedan red ne može imati duplicirani ID. | PRIMARY KEY (IzdavacID / ClanID / ZaposlenikID, PosudbaID) |
| Referencijalni integritet | Nijedna posudba ne smije postojati bez valjanog člana, knjige i zaposlenika. Svaka knjiga mora imati izdavača. | FOREIGN KEY ograničenja sa ON DELETE RESTRICT |

Naredbe:

-- Sprječava brisanje člana ako ima posudbe  
ALTER TABLE Posudbe ADD CONSTRAINT fk\_posudbe\_clan FOREIGN KEY (ClanID) REFERENCES Clanovi(ClanID);  
  
-- Sprječava brisanje knjige ako postoji posudba te knjige Ž  
-- (posudbe moraju ostati sačuvane u povijesti)  
ALTER TABLE Posudbe ADD CONSTRAINT fk\_posudbe\_knjiga FOREIGN KEY (KnjigaID)  
REFERENCES Knjige (KnjigaID) ON DELETE RESTRICT ON UPDATE CASCADE;

-- Sprječava brisanje zaposlenika ako je odradio neku posudbu  
-- (čuva se informacija tko je izdao knjigu)  
ALTER TABLE Posudbe ADD CONSTRAINT fk\_posudbe\_zaposlenik FOREIGN KEY (ZaposlenikID) REFERENCES Zaposlenici (ZaposlenikID) ON DELETE RESTRICT ON UPDATE CASCADE;

-- Sprječava brisanje izdavača ako postoje knjige tog izdavača  
-- (zadržava se veza knjiga -> izdavač)  
ALTER TABLE Knjige ADD CONSTRAINT fk\_knjige\_izdavac FOREIGN KEY (IzdavacID) REFERENCES Izdavaci (IzdavacID) ON DELETE RESTRICT ON UPDATE CASCADE;

## Obvezni podaci (NOT NULL)

|  |  |
| --- | --- |
| Stupac | Objašnjenje |
| Naziv (Izdavaci) | Bez naziva izdavača zapis nema smisla. |
| Naslov, ISBN (Knjige) | Ključni atributi. |
| DatumPosudbe, RokVracanja (Posudbe) | Posudba mora imati datum izdavanja i definirani rok. |
| Ime, Prezime (Clanovi, Zaposlenici) | Minimalni identitet korisnika / djelatnika. |

## Kontrolna pravila (CHECK)

|  |  |  |
| --- | --- | --- |
| Naziv | Izraz | Poslovna logika |
| chk\_rok\_posudbe | RokVracanja > DatumPosudbe | Rok mora biti u budućnosti u odnosu na datum izdavanja. |
| chk\_datum\_vracanja | DatumVracanja IS NULL OR DatumVracanja >= DatumPosudbe | Vraćanja ne može biti prije izdavanja. |
| chk\_status\_knjige | Status IN ('dostupna', 'posuđena', 'otpisana') | Zaštita kad se ne koristi pravi ENUM. |

Naredbe:

/\* 1) Rok vraćanja mora biti u budućnosti u odnosu na datum posudbe \*/  
  
ALTER TABLE Posudbe  
ADD CONSTRAINT chk\_rok\_posudbe  
CHECK (RokVracanja > DatumPosudbe);

/\* 2) Vraćanje ne može biti prije izdavanja (NULL je dopušten dok je primjerak još zadužen) \*/  
  
ALTER TABLE Posudbe  
ADD CONSTRAINT chk\_datum\_vracanja  
CHECK (DatumVracanja IS NULL  
OR DatumVracanja >= DatumPosudbe);

/\* 3) Status knjige ograničen na tri dopuštene vrijednosti – koristi se kada stupac NIJE definiran kao ENUM \*/  
  
ALTER TABLE Knjige  
ADD CONSTRAINT chk\_status\_knjige  
CHECK (Status IN ('dostupna','posuđena','otpisana'));

## Poslovna pravila ostvarena kodom / proceduralno

|  |  |
| --- | --- |
| Pravilo | Način provedbe |
| Kazna 0.50 € po danu kašnjenja | Dinamički se računa u analitičkom upitu. Nema potrebe za fizičkim stupcem, čime se izbjegava redundancija. |
| Limit aktivnih posudbi (npr. Max 5 po članu) | BEFORE INSERT/UPDATE trigger trg\_posudbe\_max\_5. Odbija transakciju ako SELECT COUNT(\*) iz aktivnih posudbi > 4. |
| Automatsko punjenje RokVracanja | DEFAULT vrijednost: DEFAULT (DATE\_ADD(CURDATE(), INTERVAL 21 DAY)) ili BEFORE INSERT trigger. |

**Primjeri:**

-- Dinamički izračun kazne (primjer SELECT upita)

SELECT p.PosudbaID, p.ClanID, p.DatumVracanja, p.RokVracanja,  
CASE WHEN p.DatumVracanja > p.RokVracanja THEN DATEDIFF(p.DatumVracanja, p.RokVracanja) \* 0.50 ELSE 0.00 END AS IznosKazne FROM Posudbe p;

-- BEFORE INSERT trigger provjerava broj nevracenih posudbi za člana, ako je >= 5, prekida se transakcija

DELIMITER $$

CREATE TRIGGER trg\_posudbe\_max\_5 BEFORE INSERT ON Posudbe  
FOR EACH ROW   
BEGIN  
DECLARE broj\_posudbi INT;  
SELECT COUNT(\*)  
INTO broj\_posudbi  
FROM Posudbe  
WHERE ClanID = NEW.ClanID  
AND DatumVracanja IS NULL;  
IF broj\_posudbi >= 5 THEN  
SIGNAL SQLSTATE '45000'  
SET MESSAGE\_TEXT = 'Član ima već 5 aktivnih posudbi.';  
END IF;  
END $$  
  
DELIMITER ;

-- Automatsko punjenje RokVracanja (+21 dan), najfleksibilnije, radi i s NULL  
  
DELIMITER $$  
  
CREATE TRIGGER trg\_autorok\_posudbe  
BEFORE INSERT ON Posudbe  
FOR EACH ROW  
BEGIN  
IF NEW.RokVracanja IS NULL THEN  
SET NEW.RokVracanja = DATE\_ADD(NEW.DatumPosudbe, INTERVAL 21 DAY);  
END IF;  
END $$  
  
DELIMITER ;

-- Pregled aktivnih trigger-a

SHOW TRIGGERS LIKE 'Posudbe';

## Kardinalnosti i kaskadna pravila brisanja

|  |  |  |
| --- | --- | --- |
| Veza | Kardinalnost | ON DELETE / UPDATE |
| Izdavaci - Knjige | 1 : N | ON DELETE RESTRICT (sprječava brisanje izdavača ako postoje knjige). |
| Knjige – Posudbe | 1 : N | ON DELETE RESTRICT (povijesni podaci moraju ostati). |
| Clanovi – Posudbe | 1 : N | ON DELETE RESTRICT (čuva revizijski trag) |
| Zaposlenici – Posudbe | 1 : N | ON DELETE SET NULL (opcionalno) – ako djelatnik ode, zadržava se transakcija s NULL referencom. |

# Upiti

## Top 5 izdavača prema broju posudbi u zadnjih 12 mjeseci

CREATE VIEW top\_5\_nakladnika AS  
SELECT   
 i.IzdavacID, -- ID nakladnika  
 i.Naziv, -- naziv nakladnika  
 COUNT(\*) AS broj\_posudbi -- broj posudbi u zadnjih 12 mjeseci  
FROM (  
 SELECT PosudbaID, KnjigaID  
 FROM Posudbe  
 WHERE DatumPosudbe >= DATE\_SUB(CURDATE(), INTERVAL 12 MONTH)  
) AS zadnja\_godina -- subquery   
  
JOIN Knjige k ON k.KnjigaID = zadnja\_godina.KnjigaID  
JOIN Izdavaci i ON i.IzdavacID = k.IzdavacID  
GROUP BY i.IzdavacID, i.Naziv  
ORDER BY broj\_posudbi DESC  
LIMIT 5;

**Detalji:**

* **zadnja\_godina** je alias za **ugnježđeni SELECT** koji dohvaća sve posudbe u zadnjih 12 mjeseci.
* **Subquery filtrira posudbe** pomoću DATE\_SUB(CURDATE(), INTERVAL 12 MONTH) – dohvaća samo one koje su izdane unutar zadnjih godinu dana.
* **JOIN Knjige i Izdavaci** omogućuju da se svaka posudba poveže s knjigom i njezinim izdavačem.
* **GROUP BY i.IzdavacID, i.Naziv** agregira broj posudbi po nakladniku.
* **ORDER BY broj\_posudbi DESC + LIMIT 5** prikazuje samo **Top 5** nakladnika s najviše posudbi u tom razdoblju.

## Aktivne posudbe s kaznama

CREATE VIEW aktivne\_posudbe\_s\_kaznama AS  
SELECT   
 p.PosudbaID, -- ID posudbe  
 c.ClanID, -- ID člana  
 CONCAT(c.Ime, ' ', c.Prezime) AS Clan, -- ime i prezime člana  
 k.Naslov, -- naslov knjige  
 p.DatumPosudbe, -- datum kada je knjiga posuđena  
 p.RokVracanja, -- predviđeni rok vraćanja  
 CASE  
 WHEN CURDATE() > p.RokVracanja  
 THEN TIMESTAMPDIFF(DAY, p.RokVracanja, CURDATE()) \* 0.50  
 ELSE 0  
 END AS IznosKazneEUR -- dinamički izračun kazne u €  
FROM Posudbe p  
JOIN Clanovi c ON c.ClanID = p.ClanID  
JOIN Knjige k ON k.KnjigaID = p.KnjigaID  
WHERE p.DatumVracanja IS NULL;

**Detalji:**

* DatumVracanja IS NULL: filtrira samo **aktivne posudbe** — one koje još nisu vraćene. Ove posudbe su kandidati za kašnjenje.
* CONCAT(c.Ime, ' ', c.Prezime) AS Clan: kombinira ime i prezime člana u jedan prikaz, čineći rezultat čitljivijim.
* JOIN Clanovi c ON c.ClanID = p.ClanID: povezuje posudbu s članom koji je zadužio knjigu.
* JOIN Knjige k ON k.KnjigaID = p.KnjigaID: dohvaća naslov knjige za svaku posudbu.
* CASE ... WHEN CURDATE() > p.RokVracanja ...:  
  koristi se uvjetna logika — ako je današnji datum **nakon** roka vraćanja, izračunava se kazna.
* TIMESTAMPDIFF(DAY, p.RokVracanja, CURDATE()): računa **broj dana kašnjenja** u odnosu na rok.
* \* 0.50: svaki dan kašnjenja nosi kaznu od **0,50 €**, što daje ukupni iznos kazne.
* ELSE 0: ako nije probijen rok, kazna je **nula**.
* AS IznosKazneEUR: rezultatu se daje jasan naziv u eurima.

## Rizični članovi – više od 3 aktivne posudbe

CREATE VIEW rizicni\_clanovi AS  
SELECT   
 c.ClanID, -- ID korisnika  
 CONCAT(c.Ime, ' ', c.Prezime) AS Clan, -- ime + prezime  
 COUNT(\*) AS aktivne\_posudbe -- broj trenutno zaduženih knjiga  
FROM Posudbe p  
JOIN Clanovi c ON c.ClanID = p.ClanID -- pridruži podatke o članu  
WHERE p.DatumVracanja IS NULL -- samo posudbe koje još traju  
GROUP BY c.ClanID, c.Ime, c.Prezime -- grupiramo po članu  
HAVING aktivne\_posudbe > 3 -- filtriramo na >3 aktivne  
ORDER BY aktivne\_posudbe DESC;

**Detalji:**

* DatumVracanja IS NULL: dohvaćamo samo aktivne posudbe (još nisu vraćene).
* JOIN Clanovi c ON c.ClanID = p.ClanID: povezuje svaku posudbu s članom koji ju je zadužio.
* COUNT(\*) AS aktivne\_posudbe: brojimo koliko nevraćenih knjiga svaki član trenutno ima.
* GROUP BY c.ClanID, c.Ime, c.Prezime: agregiramo broj aktivnih posudbi po članu.
* HAVING aktivne\_posudbe > 3: koristimo HAVING jer filtriramo prema agregatnoj vrijednosti (COUNT). Ovdje biramo samo članove s više od 3 aktivne posudbe.
* ORDER BY aktivne\_posudbe DESC: sortiramo tako da su članovi s najviše aktivnih posudbi na vrhu — najveći “rizik” prvi.

## Mjesečni prihodi od kazni – pivot s ukupnim zbrojem

CREATE VIEW mjesecne\_kazne AS  
SELECT  
 -- godina (ili trenutna ako nije vraćeno)  
 YEAR(COALESCE(DatumVracanja, CURDATE())) AS god,   
 -- mjesec (ili trenutni)  
 MONTH(COALESCE(DatumVracanja, CURDATE())) AS mj,   
 ROUND(SUM(  
 CASE  
 WHEN (DatumVracanja IS NULL AND CURDATE() > RokVracanja)  
 THEN TIMESTAMPDIFF(DAY, RokVracanja, CURDATE()) \* 0.50  
 WHEN DatumVracanja > RokVracanja  
 THEN TIMESTAMPDIFF(DAY, RokVracanja, DatumVracanja) \* 0.50  
 ELSE 0  
 END  
 ), 2) AS ukupno\_kazni\_eur  
FROM Posudbe  
GROUP BY   
 YEAR(COALESCE(DatumVracanja, CURDATE())),  
 MONTH(COALESCE(DatumVracanja, CURDATE()));

-- SELECT NAREDBA:

SELECT \* FROM mjesecne\_kazne ORDER BY god DESC, mj ASC;

**Detalji:**

* COALESCE(DatumVracanja, CURDATE()): Ako knjiga još **nije vraćena**, koristi se **današnji datum** kao privremeni datum povrata, kako bi kazna bila točna do danas.
* CASE ... THEN ... ELSE:
  + Ako knjiga **nije vraćena** i prošao je **RokVracanja**, izračunava se kazna:  
    broj dana kašnjenja × 0,50 €
  + Ako je knjiga **vraćena s kašnjenjem**, kazna se računa od roka do stvarnog datuma vraćanja
  + Ako nije bilo kašnjenja → **kazna je 0**
* ROUND(SUM(...), 2): Ukupni iznos kazni za svaki mjesec, zaokružen na **dvije decimale**.
* GROUP BY YEAR, MONTH: Grupira rezultat po godini i mjesecu (bilo stvarnog vraćanja ili današnjeg datuma ako knjiga nije vraćena).
* ORDER BY god DESC, mj ASC: Prikaz počinje od najnovije godine prema starijima, a unutar svake godine od siječnja prema prosincu. Ovo pomaže vizualnom uvidu u trendove kazni tijekom vremena.

## Knjige s najvećim zakašnjenjem pri vraćanju

CREATE VIEW knjige\_najvece\_kasnjenje AS

SELECT

k.Naslov,

SUM(kv.DaniKasnjenja) AS UkupnoDanaKasnjenja

FROM kasnovracanje kv

JOIN posudbe p ON kv.PosudbaID = p.PosudbaID

JOIN knjige k ON p.KnjigaID = k.KnjigaID

GROUP BY k.KnjigaID, k.Naslov;

SELECT \* FROM knjige\_najvece\_kasnjenje

ORDER BY UkupnoDanaKasnjenja DESC;

**Detalji:**

* View će prikazivati naslov knjige te ukupan broj dana kašnjenja pri vraćanju te knjige.
* SUM() koristim za sumu, odnosno za izračun ukupno dana kašnjenja.
* Prvi JOIN povezuje zapis o zakašnjelom vraćanju s konkretnom posudbom u tablici posudbe, tako što uspoređuje njihov zajednički ključ PosudbaID.
* Drugi JOIN onda povezuje tu posudbu s informacijama o knjizi u tablici knjige putem polja KnjigaID, čime dobivam naslov knjige za svako zakašnjenje.

## Izdavači po broju posudbi

CREATE VIEW broj\_posudbi\_po\_izdavacu AS

SELECT

i.IzdavacID,

i.Naziv AS Izdavac,

COUNT(\*) AS BrojPosudbi

FROM posudbe p

JOIN knjige k ON p.KnjigaID = k.KnjigaID

JOIN izdavaci i ON k.IzdavacID = i.IzdavacID

GROUP BY i.IzdavacID, i.Naziv;  
  
SELECT \* FROM broj\_posudbi\_po\_izdavacu

ORDER BY BrojPosudbi DESC;

**Detalji:**

* VIEW će prikazivati Izdavače i broj posudbi po tom izdavaču.
* Sa COUNT brojimo posudbe po izdavaču.
* U SELECT dijelu koda odabirem sve izdavače i brojeve njihovih posudbi te ih poredam po broju posudbi od najviše prema najmanje.

# Zaključak

U sklopu ovog projekta implementiran je kompletan modul baze podataka za upravljanje izdavačima, članovima, zaposlenicima i posudbama u knjižnici.

Kroz dizajn i implementaciju relacijskih tablica te dodavanjem pravila integriteta i kontrolnih ograničenja, osigurana je konzistentnost i pouzdanost podataka.

Korištenjem naprednih SQL upita, izrađeni su analitički pogledi (view-i) koji omogućavaju jednostavno praćenje poslovnih metrika poput izdavača prema broju posudbi, obračuna kazni, identificiranja korisnika s višestrukim zaduženjima, te izračuna mjesečnih prihoda od zakasnina.

Projekt je dodatno obogaćen primjenom poslovnih pravila kroz CHECK ograničenja, FOREIGN KEY odnose i proceduralne mehanizme poput triggera. Time je funkcionalnost baze podignuta na razinu koja omogućuje realnu primjenu u produkcijskom okruženju.

Realizacijom ovog modula zadovoljio sam sve funkcionalne i tehničke zahtjeve, a baza je spremna za daljnji razvoj i integraciju s korisničkim sučeljem ili aplikacijom.

Kao potencijalna proširenja predlažem dodavanje povijesti promjena, statistike zaposlenika te vizualizaciju podataka putem BI alata.