Kolegij:

Baze podataka I.

Projekt:

Sustav za upravljanje knjižnicom

Tim 31:

Daniel Katić (0303123347)

Lana Kohut (0009069158)

Josip Orešković (0067224448)

Petra Tuškan (0058206931)

Maja Kovačević (0009074625)

# 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 ili autora. 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).

## 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 prikazujemo konceptualni ER-dijagram i logičku shemu baze s objašnjenjem svih ograničenja.

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

Središnji dio čini pregled SQL upita svakog člana tima s objašnjenjem rezultata.

Dokument završava zaključkom u kojem navodimo ostvarene ciljeve i prijedloge za buduća unaprijeđenja sustava.

# Opis poslovnog procesa

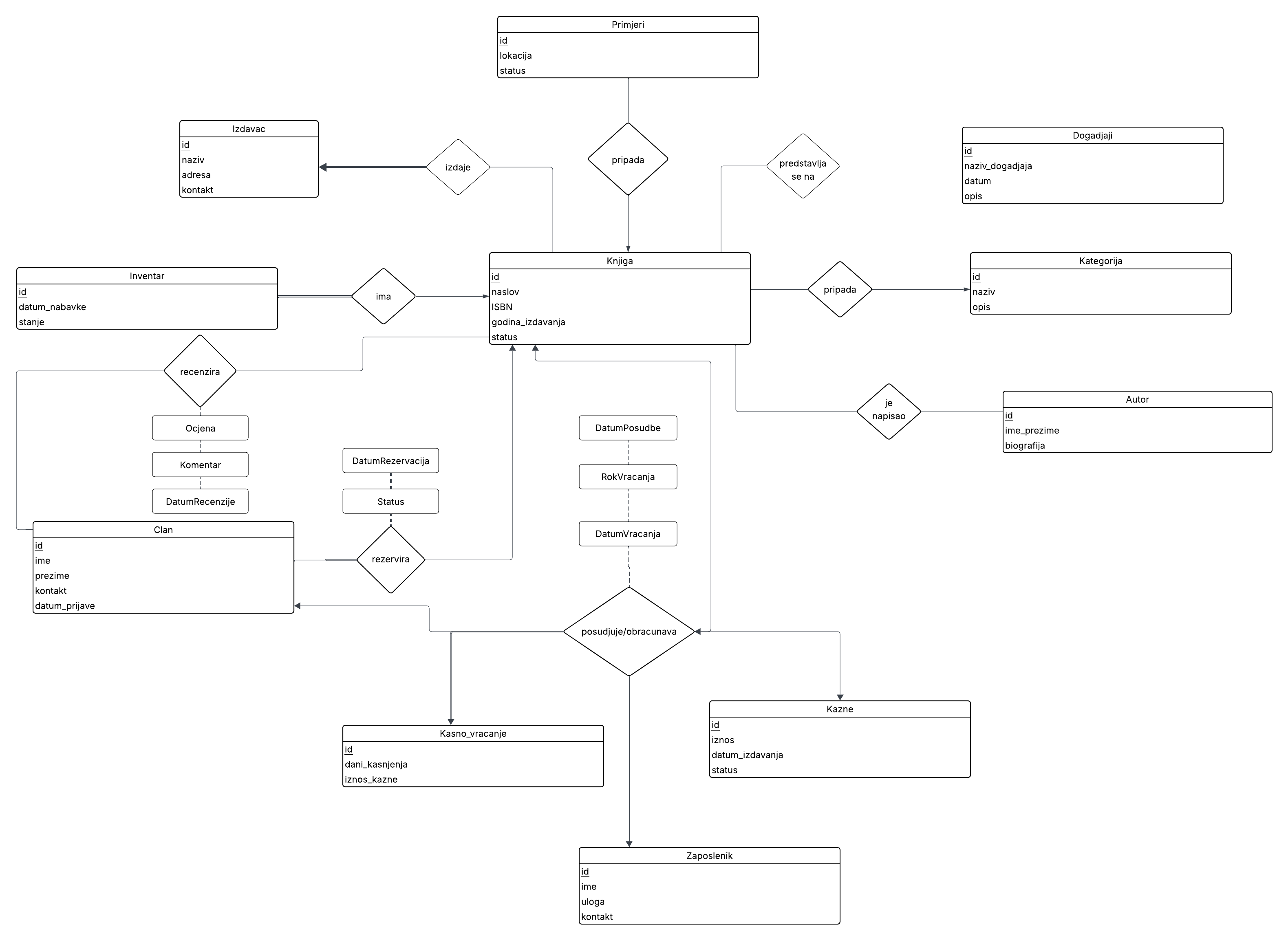
## Poslovni proces knjižnice – „život jedne knjige“

|  |  |  |
| --- | --- | --- |
| **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, GodinaIzdavanja,Status, *IzdavacID. KategorijaID*)  
Izdavaci (IzdavacID, Naziv, Adresa, Kontakt)   
Clanovi (ClanID, Ime, Prezime, Kontakt, DatumPrijave)  
Zaposlenici (ZaposlenikID, Ime, Uloga, Kontakt)  
Posuđuje (KnjigaID, ClanID, ZaposlenikID, DatumPosudbe, RokVracanja, DatumVracanja)

Inventar (InventarID, KnjigaID, DatumNabavke, Stanje)

Dogadaji (DogadajID, NazivDogadaja, Datum, Opis)

KnjigaDogadaji (KnjigaID, DogadajID)

Clanovi (ClanID, Ime, Prezime, Kontakt, DatumPrijave)

Primjeri (PrimjerID, KnjigaID, Lokacija, Status)

Recenzija (ClanID, KnjigaID, Ocjena, Komentar, DatumRecenzije)

Kazne (KaznaID, PosudbaID, Iznos, DatumIzdavanja, Status)   
Kategorije (KategorijaID, Naziv, Opis)

Autori (AutorID, ImePrezime, Biografija)

KnjigaAutori(KnjigaID, AutorID)

Kasnovracanje(KasnoVracanjeID, PosudbaID, DaniKasnjenja, IznosKazne)

# 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. |

**Tablica Autori**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **AutorID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID autora |
| **ImePrezime** | VARCHAR (255) NOT NULL | Ime i prezime autora |
| **Biografija** | TEXT | Biografija autora |

**Tablica KnjigaAutori**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **AutorID** | INT FK | Referenca na ID autora iz tablice Autori |
| **KnjigaID** | INT FK | Referenca na ID knjige iz tablice Knjiga |

**Tablica Knjiga**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **KnjigaID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID knjige |
| **Naslov** | VARCHAR(255) NOT NULL | Naslov knjige |
| **ISBN** | VARCHAR(20) | ISBN kod knjige |
| **GodinaIzdavanja** | YEAR | Godina izdavanja knjige |
| **Status** | VARCHAR(50) | Status knjige (posuđena, rezervirana, oštećena) |
| **IzdavacID** | FK INT | Referenca na ID izdavača iz tablice Izdavači |
| **KategorijaID** | FK INT | Referenca na ID kategorije iz tablice Kategorija |

**Tablica Primjeri**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **PrimjerID** | INT PK, AUTO\_INCREMENT, NOT NULL | Id primjerka knjige |
| **KnjigaID** | INT FK NOT NULL | Id knjige kojoj primjerak pripada |
| **Lokacija** | VARCHAR(255) | Lokacija unutar knjižnice gdje se primjerak knjige nalazi |
| **Status** | VARCHAR(50) | Opisuje dostupnost pojedinog primjerka |

**Tablica Dogadjaji**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **DogadajID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID događaja |
| **NazivDogadaja** | VARCHAR(100) NOT NULL | Naziv događaja |
| **Datum** | DATE NOT NULL | Datum održavanja događaja |
| **Opis** | TEXT | Kratki opis događaja i dodatne informacije |

**Tablica KnjigaDogadjaji**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **KnjigaID** | INT FK NOT NULL | Referenca na knjigu koja se izlaže na događaju/je dio događaja |
| **DogadajID** | INT FK NOT NULL | Referenca na događaj na kojem je knjiga/knjige predstavljena |

**Tablica Kategorije**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **KategorijaID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID kategorije |
| **Naziv** | VARCHAR(100) NOT NULL | Naziv kategorije |
| **Opis** | TEXT | Opis kategorije |

**Tablica Inventar**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **InventarID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID inventarskog zapisa |
| **KnjigaID** | INT FK NOT NULL | Referenca na knjigu |
| **DatumNabavke** | DATE NOT NULL | Datum nabavke primjerka |
| **Stanje** | VARCHAR(50) | Stanje primjerka |

**Tablica Recenzije**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **RecenzijaID** | INT PK, AUTO\_INCREMENT, NOT NULL | Id recenzije |
| **KnjigaID** | INT FK NOT NULL | Id knjige za koju je ostavljena recenzija |
| **ClanID** | INT FK NOT NULL | Id člana koji je ostavio recenziju |
| **Ocjena** | INT CHECK (Ocjena BETWEEN 1 AND 5) | Ocjena recenzije sa vrijednošću između 1 I 5 |
| **Komentar** | TEXT | Tekstualni komentar recenzije |
| **DatumRecenzije** | DATE | Datum kada je recenzija ostavljena |

**Tablica Rezervacije**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **RezervacijaID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID rezervacije |
| **ClanID** | INT FK NOT NULL | Član koji je naporavio rezervaciju |
| **KnjigaID** | INT FK NOT NULL | Knjiga koja je rrezervirana |
| **DatumRezervacije** | DATE NOT NULL | Datum rezervacije |
| **Status** | VARCHAR(50) | Odobreno/Odbijeno/Na čekanju |

**Tablica KasnoVracanje**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **KasnoVracanjeID** | INT PK, AUTO\_INCREMENT, NOT NULL | Id pojedinog zapisa o kašnjenju |
| **PosudbaID** | INT FK NOT NULL | Id posudbe na koju se odnosi kasno vraćanje |
| **DaniKasnjenja** | INT NOT NULL | Broj dana kašnjenja |
| **Iznos Kazne** | DECIMAL(10,2) NOT NULL | Novčani iznos kazne |

**Tablica Kazne**

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Domena/Ograničenja** | **Komentar** |
| **KaznaID** | INT PK, AUTO\_INCREMENT, NOT NULL | ID kazne |
| **PosudbaID** | INT FK NOT NULL | Posudba na koju se kazna odnosi |
| **Iznos** | DECIMAL(10,2) NOT NULL | Iznos kazne |
| **DatumIzdavanja** | DATE DEFAULT NULL | Datum obračuna kazne |
| **Status** | VARCHAR(50) | Plaćeno/Neplaćeno/U obradi |

# 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 – Daniel Katić (0303123347)

## 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.

## 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.

# Upiti – Lana Kohut (0009069158)

# Cilj mog modula

# U timu sam zadužena za dizajn i implementaciju triju tablica – Knjiga, KnjigaAutori i Autori te naprednih pogleda koji nad njima izvode korisne analize.

Modul omogućuje:

* Pregled knjiga i autora po različitim uvjetima
* Izračun ukupnog broja knjiga po uvjetima te prosječne godine izdanja knjiga

Sve knjige od autora koji su napisali više od 2 knjige

Može se koristiti kada korisnici sustava žele vidjeti sve knjige čiji autori su napisali više od 2 knjige (na taj način se isključuju najnoviji autori sa samo jednom ili 2 knjige primjerice)

CREATE VIEW Autori\_sa\_vise\_od\_2\_knjige AS  
SELECT k.Naslov, a.ImePrezime  
FROM knjige AS k  
JOIN knjigaautori AS ka ON k.KnjigaID = ka.KnjigaID  
JOIN autori a ON ka.AutorID = a.AutorID  
WHERE a.AutorID IN (  
 SELECT AutorID  
 FROM knjigaautori  
 GROUP BY AutorID  
 HAVING COUNT(DISTINCT KnjigaID) > 2  
);

**Detalji:**

* VIEW će prikazivati sve autore koji su napisali više od 2 knjige
* U SELECTU biram naslov knjige te ime i prezime autora
* U JOINU prvo spajam tablicu Knjige sa KnjigaAutori, a onda u sljedećoj liniji tablicu Autori sa istom tablicom
* U WHERE dijelu pišem svoj uvjet, želim samo one autore koji su napisali više od 2 knjige koje su dostupne u knjižnici

Prosjecna godina kada su knjige različitih autora izdane

Može se koristiti kada korisnici sustava žele vidjeti prosječnu godinu kada su izdane knjige različitih autora. Svrha podataka bi mogla biti primjerice za blog post koji bi ujedno dao i primjer knjiga nekih autora te time dodatno poticao korisnike (nove ili stare) na čitanje.

CREATE VIEW ProsjecnaGodinaAutora AS  
SELECT a.AutorID, a.ImePrezime, ROUND(AVG(k.GodinaIzdavanja)) AS ProsjecnaGodina  
FROM autori a  
JOIN knjigaautori ka ON a.AutorID = ka.AutorID  
JOIN knjige k ON ka.KnjigaID = k.KnjigaID  
GROUP BY a.AutorID, a.ImePrezime;

**Detalji:**

* VIEW će prikazivati prosjecne godine kada su knjige različitih autora izdane
* U SELECTU biram ID Autora, Ime i Prezime autora te prosječnu godinu izdavanja (zaokruženo te prikazano u novom stupcu ProsjecnaGodina)
* U JOINU prvo spajam tablicu Autori sa KnjigaAutori, a onda u sljedećoj liniji tablicu Knjige sa istom tablicom
* U GROUPBY grupiram rezultat po autoru, tako da dobijemo traženi rezultat

Najnovije knjige po autoru

Može se koristiti kada korisnici sustava žele vidjeti najnovije izdane knjige od svakog autora. Korisno za promocije, događaje ili objave na web stranici.

CREATE VIEW NajnovijeKnjigeAutora AS  
SELECT a.AutorID, a.ImePrezime, k.Naslov, k.GodinaIzdavanja  
FROM autori a  
JOIN knjigaautori ka ON a.AutorID = ka.AutorID  
JOIN knjige k ON ka.KnjigaID = k.KnjigaID  
WHERE k.GodinaIzdavanja = (  
 SELECT MAX(k2.GodinaIzdavanja)  
 FROM knjigaautori ka2  
 JOIN knjige k2 ON ka2.KnjigaID = k2.KnjigaID  
 WHERE ka2.AutorID = a.AutorID  
);

**Detalji:**

* VIEW će prikazivati najnovije izdane knjige po autoru
* U SELECTU biram ID Autora, Ime i Prezime autora, naslov knjige te godinu izdavanja
* JOIN – isto kao i u prijašnjim upitima
* U WHERE filtriram podatke tako da dobijem samo najnovije knjige (MAX (k2.GodinaIzdavanja))

Pogled autora po kategoriji te uključen broj knjiga koje su napisali u toj kategoriji

Može se koristiti kada korisnici sustava žele vidjeti više informacija knjigama koje je autor napisao u svakoj kategoriji (žanru) te koliko ih je u toj kategoriji. Korisno za promocije autora, posebne događaje u knjižnici, itd.

CREATE VIEW AutoriPoKategoriji AS  
SELECT   
 a.AutorID,   
 a.ImePrezime,   
 k.KategorijaID,   
 kat.Naziv AS NazivKategorije,  
 COUNT(k.KnjigaID) AS BrojKnjigaUKategoriji  
FROM autori a  
JOIN knjigaautori ka ON a.AutorID = ka.AutorID  
JOIN knjige k ON ka.KnjigaID = k.KnjigaID  
JOIN kategorije kat ON k.KategorijaID = kat.KategorijaID  
GROUP BY a.AutorID, k.KategorijaID  
ORDER BY a.ImePrezime, BrojKnjigaUKategoriji DESC;

**Detalji:**

* VIEW će prikazivati autore po kategoriji (žanru) te uključen broj knjiga koje su napisali u toj kategoriji
* U SELECTU biram ID Autora, Ime i Prezime autora, ID kategorije, naziv kategorije te brojim knjige tog autora u toj kategoriji (novi stupac BrojKnjigaUKategoriji)
* JOIN – isto kao i u prijašnjim upitima, sa dodatkom da spajam vanjski ključ KategorijaID u tablici Knjiga sa primarnim ključem u tablici Kategorije te tako mogu izvući naziv kategorije koji je potreban za svrhu ovog pogleda
* U GROUP BY grupiram podatke po autorima i kategoriji
* U ORDER BY sortiram po imenu i prezimenu autora te po broju knjiga

Autori bez objavljene knjige u zadnjih 10 godina

Može se koristiti kada korisnici sustava žele vidjeti više informacija starijim knjigama te ih na jednostavan način, koristeći podatke iz pregleda, promovirati korisnicima te na taj način povećati posudbu tih knjiga.

CREATE VIEW AutoriBezNovijihIzdanja AS  
SELECT DISTINCT a.AutorID, a.ImePrezime  
FROM autori a  
WHERE a.AutorID NOT IN (  
 SELECT DISTINCT ka.AutorID  
 FROM knjigaautori ka  
 JOIN knjige k ON ka.KnjigaID = k.KnjigaID  
 WHERE k.GodinaIzdavanja >= YEAR(CURDATE()) - 10  
);

**Detalji:**

* VIEW će prikazivati autore koji nisu objavili knjigu u zadnjih 10 godina
* U SELECTU biram jedinstveni (DISTINCT) ID Autora te Ime i Prezime autora
* U WHERE tražim autore koji NISU (NOT IN) objavili knjigu u zadnjih 10 godina (WHERE k.GodinaIzdavanja >= YEAR(CURDATE()) - 10)

Knjige čija je kategorija 12,2 ili 22 (Fantastika) te ukupan broj knjiga koje su dostupne od tih autora

Može se koristiti kada korisnici sustava žele vidjeti više informacija knjigama koje je autor napisao u kategoriji fantastike te koliko knjiga je ukupno dostupno od autora. Korisno za promocije tih knjiga ili geeky događaje (sajmovi, festivali,...)

CREATE VIEW fantastika\_dostupan\_broj\_knjiga AS  
SELECT   
 k.Naslov AS NaslovKnjige,  
 a.ImePrezime,  
 (  
 SELECT COUNT(\*)   
 FROM knjigaautori AS ka2   
 WHERE ka2.AutorID = a.AutorID  
 ) AS UkupnoKnjigaOdAutora  
FROM knjige AS k  
JOIN knjigaautori ka ON k.KnjigaID = ka.KnjigaID  
JOIN autori AS a ON ka.AutorID = a.AutorID  
WHERE k.KategorijaID = 12 OR k.KategorijaID = 22 OR k.KategorijaID = 2;

**Detalji:**

* VIEW će prikazivati autore i knjige iz kategorije Fantastika te ukupan broj knjiga dostupan od pojedinog autora
* U SELECTU biram naslov knjige te ime i prezime autora
* U podupitu brojim ukupan broj knjiga dostupan od svakog autora
* JOIN – isto kao i u prva dva upita, spajanje tablica Knjiga i Autori preko KnjigaAutori
* U WHERE filtriram podatke koji pripadaju kategoriji Fantastika (12, 22 ili 2)

# Upiti – Josip Orešković (00672224448)

# Cilj mog modula

U timu sam zadužen za dizajn i implementaciju triju tablica – kazne, rezervacije i kategorije te naprednih pogleda koji nad njima izvode korisne analize.

Modul omogućuje:

* Centralizirano čuvanje podataka o kategorijama.
* Praćenje svake transakcije obračuna kazni i rezervacija knjiga.

## Ukupna kazna po članu s podacima o broju posudbi i zakašnjenjima

Pogled **KaznePoClanuDetaljno** je složen analitički SQL pogled koji prikazuje detaljan pregled članova knjižnice koji su barem jednom dobili kaznu. Njegova svrha je omogućiti osoblju knjižnice da identificira korisnike s najvećim financijskim dugovanjima i potencijalno problematičnim ponašanjem u pogledu kašnjenja.

CREATE VIEW KaznePoClanuDetaljno AS

SELECT

c.ClanID,

c.Ime,

c.Prezime,

COUNT(DISTINCT p.PosudbaID) AS BrojPosudbi,

COUNT(DISTINCT k.KaznaID) AS BrojKazni,

SUM(k.Iznos) AS UkupnaKazna

FROM kazne k

JOIN posudbe p ON k.PosudbaID = p.PosudbaID

JOIN clanovi c ON p.ClanID = c.ClanID

GROUP BY c.ClanID, c.Ime, c.Prezime

HAVING UkupnaKazna > 0

ORDER BY UkupnaKazna DESC;

**Detalji:**

* CREATE VIEW KaznePoClanuDetaljno AS

Kreira **SQL pogled** (virtualnu tablicu) s nazivom KaznePoClanuDetaljno.

* SELECT c.ClanID, c.Ime, c.Prezime

Dohvaća osnovne identifikacijske podatke svakog člana.

* COUNT(DISTINCT p.PosudbaID) AS BrojPosudbi

Broji broj **različitih posudbi** koje je član imao, ali samo za one koje su

povezane s kaznama.

* SUM(k.Iznos) AS UkupnaKazna

Izračunava **ukupan iznos kazni** koje je član prikupio. Ovo je ključna metrika za financijsko praćenje

* FROM kazne k JOIN posudbe p ON k.PosudbaID = p.PosudbaID JOIN clanovi c ON p.ClanID = c.ClanID

Vrši povezivanje tablica:

kazne → izvor podataka o novčanim kaznama.

posudbe → veza između kazne i člana (kroz posudbu).

clanovi → dohvat podataka o članu

* + GROUP BY c.ClanID, c.Ime, c.Prezime

Grupira rezultate po članu – kako bi se dobile agregirane statistike po osobi.

* + HAVING UkupnaKazna > 0

Prikazuje **samo članove koji imaju barem jednu kaznu** (ne prikazuje one s 0 iznosa).

* + ORDER BY UkupnaKazna DESC;

Sortira rezultate **od člana s najvećim ukupnim dugovanjem prema dolje**.

| **Problem** | **Kako ga pogled rješava** |
| --- | --- |
| Praćenje financijskih dugovanja članova | Prikazuje točan iznos kazni po članu |
| Identifikacija problematičnih korisnika | Broji broj kazni i povezanih posudbi |
| Analiza korisničkog ponašanja | Kombinira broj posudbi i kazni – omjer može ukazivati na navike |
| Prioriteti za naplatu ili kontaktiranje | Sortira članove po visini dugovanja |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ClanID** | **Ime** | **Prezime** | **BrojPosudbi** | **BrojKazni** | **UkupnaKazna** |
| 11 | Ljubica | Pešić | 2 | 2 | 16797 |
| 185 | Jasminka | Turina | 1 | 2 | 11812 |
| 88 | Zvonko | Miše | 2 | 2 | 9936 |
| 166 | Marina | Kovaček | 1 | 1 | 9351 |
| 91 | Jelena | Hrvojić | 1 | 3 | 9222 |
| 95 | Branka | Jurlina | 1 | 1 | 8289 |
| 197 | Marica | Ljubetić | 1 | 1 | 7632 |
| 136 | Pero | Milovac | 1 | 1 | 7563 |
| 148 | Mara | Karagić | 1 | 1 | 7484 |
| 170 | Lovre | Meić | 1 | 1 | 7287 |
| 86 | Milena | Dautović | 1 | 1 | 7280 |

Kazne po mjesecu i godini

Pogled **KaznePoMjesecu** je analitički SQL pogled koji omogućuje mjesečno praćenje kazni izdanih u knjižnici. Služi kao alat za financijsku i operativnu analizu, pomažući knjižničarima i administraciji da razumiju kada se kazne najčešće izdaju i koliki su ukupni iznosi.

CREATE VIEW KaznePoMjesecu AS

SELECT

YEAR(k. DatumIzdavanja) AS Godina,

MONTH(k. DatumIzdavanja) AS Mjesec,

COUNT(\*) AS BrojKazni,

SUM(k.Iznos) AS UkupanIznos

FROM kazne k

GROUP BY Godina, Mjesec

ORDER BY Godina DESC, Mjesec DESC;

**Detalji:**

* CREATE VIEW KaznePoMjesecu AS

Kreira **virtualnu tablicu** (pogled) pod nazivom KaznePoMjesecu

* SELECT YEAR(k.DatumKazne) AS Godina, MONTH(k.DatumKazne) AS Mjesec

Ekstrahira godinu i mjesec iz stupca DatumKazne (datum kada je kazna izdana).

Ove dvije vrijednosti omogućuju grupiranje podataka po vremenskim periodima, konkretno po godini i mjesecu.

* COUNT(\*) AS BrojKazni,

Broji **ukupan broj kazni** izdanih u tom mjesecu

* SUM(k.Iznos) AS UkupanIznos

Zbraja ukupni **iznos svih kazni** u tom mjesecu – ključan podatak za **praćenje prihoda** od kazni ili identificiranje mjeseci u kojima korisnici najviše kasne.

* FROM kazne k

Koristi tablicu kazne kao izvor podataka.

* GROUP BY Godina, Mjesec

Grupira zapise po godini i mjesecu, omogućujući **agregaciju** (zbrajanje i brojanje) za svaki mjesec posebno.

| **Problem / Potreba** | **Kako pogled pomaže** |
| --- | --- |
| Praćenje učestalosti kašnjenja korisnika | Prikazuje broj kazni po mjesecima |
| Praćenje financijskog prihoda od kazni | Daje ukupan iznos naplaćenih kazni po mjesecu |
| Sezonska analiza korisničkog ponašanja | Omogućuje usporedbu među mjesecima/godinama |
| Predviđanje potrebe za pojačanim nadzorom | Ako se broj kazni povećava u određenim mjesecima |
| Laka integracija s grafovima i dashboardima | Prikladan format za dijagrame (npr. linijski graf u Power BI-u) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Godina** | **Mjesec** | **BrojKazni** | **UkupanIznos** |
| 2025 | 2 | 1 | 7484 |
| 2025 | 1 | 5 | 27464 |
| 2024 | 12 | 2 | 11360 |
| 2024 | 11 | 1 | 1721 |
| 2024 | 10 | 2 | 9338 |
| 2024 | 9 | 1 | 4974 |
| 2024 | 8 | 7 | 26298 |
| 2024 | 7 | 3 | 14175 |
| 2024 | 6 | 3 | 23719 |
| 2024 | 5 | 1 | 7563 |
| 2024 | 4 | 2 | 10447 |
| 2024 | 3 | 2 | 15584 |

Rezervacije koje nisu realizirane posudbom (neaktivne rezervacije)

Pogled NerealiziraneRezervacije prikazuje rezervacije koje nisu realizirane, tj. slučajeve u kojima je korisnik rezervirao knjigu, ali je nije posudio u roku 7 dana od datuma rezervacije. Ovaj pogled je vrlo koristan za praćenje neučinkovitih rezervacija, identificiranje korisnika koji ne preuzimaju rezervirane knjige, te optimizaciju rada knjižnice.

CREATE VIEW NerealiziraneRezervacije AS

SELECT

r.RezervacijaID,

r.DatumRezervacije,

c.Ime,

c.Prezime,

k.Naslov

FROM rezervacije r

JOIN clanovi c ON r.ClanID = c.ClanID

JOIN knjige k ON r.KnjigaID = k.KnjigaID

LEFT JOIN posudbe p

ON r.ClanID = p.ClanID

AND r.KnjigaID = p.KnjigaID

AND DATE(p.DatumPosudbe) BETWEEN DATE(r.DatumRezervacije) AND DATE\_ADD(r.DatumRezervacije, INTERVAL 7 DAY)

WHERE p.PosudbaID IS NULL

ORDER BY r.DatumRezervacije DESC;

**Detalji:**

* CREATE VIEW NerealiziraneRezervacije AS

Kreira se novi **pogled** s imenom NerealiziraneRezervacije

* SELECT r.RezervacijaID, r.DatumRezervacije,c.Ime, c.Prezime, k.Naslov

Izbor podataka koji se prikazuju:

* RezervacijaID: jedinstveni identifikator rezervacije
* DatumRezervacije: kada je knjiga rezervirana
* Ime, Prezime: ime i prezime člana koji je rezervirao knjigu
* Naslov: naslov knjige koja je rezervirana
* FROM rezervacije r JOIN clanovi c ON r.ClanID = c.ClanID JOIN knjige k ON r.KnjigaID = k.KnjigaID

Spajanje tablica:

* rezervacije → osnovna tablica
* clanovi → dohvaćanje podataka o korisniku
* knjige → dohvaćanje naslova knjige
* LEFT JOIN posudbe p ON r.ClanID = p.ClanID AND r.KnjigaID = p.KnjigaID AND DATE(p.DatumPosudbe) BETWEEN DATE(r.DatumRezervacije) AND DATE\_ADD(r.DatumRezervacije, INTERVAL 7 DAY)

Koristi se **LEFT JOIN** kako bi pronašao postoji li posudba za tu rezervaciju.

Spajanje se događa **samo ako je ista knjiga posuđena od istog člana unutar 7 dana od rezervacije**.

Ako se posudba nije dogodila u tom roku, rezultat iz posudbe će biti NULL.

To znači: rezervacija je **nerealizirana ako korisnik nije posudio knjigu u roku od 7 dana**.

* WHERE p.PosudbaID IS NULL

Filtrira samo one rezervacije gdje **nije pronađena odgovarajuća posudba** – znači da rezervacija **nije realizirana**.

* ORDER BY r.DatumRezervacije DESC

Rezultati su sortirani prema datumu rezervacije (najnovije na vrhu).

| **Problem / potreba** | | | **Kako pogled pomaže** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Neučinkovito korištenje rezervacija | | | Otkriva rezervacije koje su prošle bez posudbe | | | |
| Zauzimanje knjiga koje nitko ne preuzima | | | Omogućuje knjižničarima da brzo oslobode rezervacije | | | |
| Identifikacija korisnika koji često rezerviraju uzalud | | | Olakšava praćenje ponašanja korisnika | | | |
| Optimizacija pravila rezervacija | | | Pomaže odlučiti je li 7 dana predug/prekratak rok | | | |
| Automatizirano čišćenje ili slanje podsjetnika | | | Može se koristiti kao podloga za e-mail obavijesti | | | |
| **RezervacijaID** | **DatumRezervacije** | **Ime** | | **Prezime** | **Naslov** |  |
| 23 | 12.03.2025 | Anton | | Blažičko | Sjene nad Balkanom |  |
| 39 | 12.03.2025 | Biljana | | Marijanović | Razum i osjećaji |  |
| 154 | 11.03.2025 | Anto | | Kramarić | Noćni vlak za Lisabon |  |
| 74 | 10.03.2025 | Sara | | Botić | Gospodin Mercedes |  |
| 101 | 08.03.2025 | Petra | | Stipanović | Plavi anđeo |  |
| 183 | 08.03.2025 | Marta | | Harapin | Otok s blagom |  |
| 72 | 07.03.2025 | Jasna | | Bušljeta | Pustolovine Huckleberryja Finna |  |
| 173 | 07.03.2025 | Anica | | Barnaba | Oluja mačeva |  |
| 176 | 05.03.2025 | Zoran | | Raspor | Jantarni teleskop |  |
| 96 | 01.03.2025 | Zdenka | | Jelavić | Gozba vrana |  |
| 130 | 01.03.2025 | Damir | | Barnaba | Pepeljuga |  |
| 169 | 01.03.2025 | Nada | | Nikolić | Anđeli i demoni |  |
| 120 | 28.02.2025 | Andrea | | Mirosavljević | Noćni vlak za Lisabon |  |
| 37 | 27.02.2025 | Hana | | Čulina | Derviš i smrt |  |
| 79 | 27.02.2025 | Dalibor | | Mimica | Alisa u zemlji čudesa |  |
| 114 | 27.02.2025 | Zoran | | Sinožić | Gospodin Nobody |  |
| 17 | 25.02.2025 | Siniša | | Pleše | Mali princ |  |

Popularnost kategorija

Pogled PopularnostKategorija služi za analizu popularnosti knjižnih kategorija na temelju broja posudbi i rezervacija knjiga koje pripadaju tim kategorijama. Ovaj pogled omogućuje knjižnici da prati interes korisnika za različite žanrove ili tematske skupine knjiga i na temelju toga donosi informirane odluke o nabavi i upravljanju fondom.

CREATE VIEW PopularnostKategorija AS

SELECT

kat.Naziv AS Kategorija,

COUNT(DISTINCT p.PosudbaID) AS BrojPosudbi,

COUNT(DISTINCT r.RezervacijaID) AS BrojRezervacija

FROM kategorije kat

JOIN knjige k ON kat.KategorijaID = k.KategorijaID

LEFT JOIN posudbe p ON p.KnjigaID = k.KnjigaID

LEFT JOIN rezervacije r ON r.KnjigaID = k.KnjigaID

GROUP BY kat.Naziv;

**Detalji:**

* CREATE VIEW PopularnostKategorija AS

Kreira se **pogled** s nazivom PopularnostKategorija, koji se može koristiti kao tablica u upitima, izvještajima ili dashboardima.

* SELECT kat.Naziv AS Kategorija, COUNT(DISTINCT p.PosudbaID) AS BrojPosudbi, COUNT(DISTINCT r.RezervacijaID) AS BrojRezervacija

kat.Naziv AS Kategorija: dohvaća naziv kategorije iz tablice kategorije, npr. "Povijest", "Znanstvena fantastika", "Psihologija".

COUNT(DISTINCT p.PosudbaID): broji **jedinstvene posudbe** knjiga unutar te kategorije.

COUNT(DISTINCT r.RezervacijaID): broji **jedinstvene rezervacije** knjiga unutar te kategorije.

Korištenje DISTINCT je važno jer:

Jedna knjiga može biti više puta posuđena ili rezervirana.

Broji se broj jedinstvenih transakcija, ne broj primjeraka.

* FROM kategorije kat JOIN knjige k ON kat.KategorijaID = k.KategorijaID

Povezuje kategorije s knjigama koje im pripadaju putem KategorijaID

* LEFT JOIN posudbe p ON p.KnjigaID = k.KnjigaID LEFT JOIN rezervacije r ON r.KnjigaID = k.KnjigaID

**LEFT JOIN** omogućava da se u rezultatima prikazuju i one kategorije koje trenutno **nema posudbi ili rezervacija** – korisno za praćenje i nepopularnih kategorija.

Povezuje se s tablicama posudbe i rezervacije kako bi se brojali događaji povezani s knjigama unutar svake kategorije.

* GROUP BY kat.Naziv;

Grupira sve podatke prema **nazivu kategorije**, kako bi se dobile agregirane vrijednosti po svakoj.

| **Problem / potreba** | **Kako pogled pomaže** |
| --- | --- |
| Praćenje popularnosti pojedinih žanrova | Pruža broj posudbi i rezervacija po kategoriji |
| Planiranje nabave novih knjiga | Kategorije s visokom potražnjom mogu se prioritetno obnavljati |
| Praćenje interesa korisnika kroz vrijeme | Može se uspoređivati s prethodnim pogledima (mjesečnim ili godišnjim) |
| Identifikacija nepopularnih kategorija | Niske vrijednosti mogu sugerirati zastarjelost sadržaja ili slabu vidljivost |
| Izvještavanje za upravu knjižnice | Pogled je idealan za prikaz u Power BI, Excelu, Tableau itd. |

|  |  |  |
| --- | --- | --- |
| **Kategorija** | **BrojPosudbi** | **BrojRezervacija** |
| Avantura | 23 | 25 |
| Fantastika | 7 | 11 |
| Filozofski roman | 18 | 28 |
| Horor | 14 | 15 |
| Klasici | 26 | 27 |
| Kriminalistika | 28 | 21 |
| Ljubavni roman | 32 | 26 |
| Povijesni roman | 12 | 15 |
| Psihološki triler | 21 | 12 |
| Znanstvena fantastika | 19 | 20 |

Prosječno trajanje posudbi po kategoriji

Pogled ProsjecnoZadrzavanjePoKategoriji služi za analizu prosječnog vremena zadržavanja knjiga po kategorijama, tj. koliko dana korisnici u prosjeku drže posuđene knjige prije nego što ih vrate. Ovo je vrlo koristan pokazatelj za upravljanje fondom, planiranje nabave primjeraka, kao i za podešavanje pravila posudbe u knjižnici.

CREATE VIEW ProsjecnoZadrzavanjePoKategoriji AS

SELECT

kat.Naziv AS Kategorija,

ROUND(AVG(DATEDIFF(p.DatumVracanja, p.DatumPosudbe)), 1) AS ProsjecniBrojDana

FROM kategorije kat

JOIN knjige k ON kat.KategorijaID = k.KategorijaID

JOIN posudbe p ON p.KnjigaID = k.KnjigaID

WHERE p.DatumVracanja IS NOT NULL

GROUP BY kat.Naziv;

**Detalji:**

* CREATE VIEW ProsjecnoZadrzavanjePoKategoriji AS

Kreira se novi pogled (view) koji se može koristiti kao virtualna tablica za upite, izvještaje i analize.

* SELECT kat.Naziv AS Kategorija, ROUND(AVG(DATEDIFF(p.DatumVracanja, p.DatumPosudbe)), 1) AS ProsjecniBrojDana

**kat.**Naziv AS Kategorija: prikazuje naziv kategorije knjiga, npr. "Povijest", "Roman", "Biografija".

DATEDIFF(p.DatumVracanja, p.DatumPosudbe): izračunava broj dana između datuma posudbe i datuma vraćanja knjige.

AVG(...): računa **prosjek** tih vremenskih razlika — znači prosječno koliko dana knjige iz određene kategorije ostaju kod korisnika.

ROUND(..., 1): zaokružuje rezultat na **jednu decimalu** radi čitljivosti.

Ova mjera se naziva i **vrijeme zadržavanja (retention period)**.

* FROM kategorije kat JOIN knjige k ON kat.KategorijaID = k.KategorijaID

JOIN posudbe p ON p.KnjigaID = k.KnjigaID

Redoslijed spajanja:

* + kategorije i knjige: kako bi znali kojoj kategoriji knjiga pripada
  + knjige i posudbe: da bismo dobili podatke o posudbama za svaku knjigu

Svaka posudba je povezana s konkretnom knjigom, a knjiga s određenom kategorijom.

* + WHERE p.DatumVracanja IS NOT NULL

U obzir se **uzimaju samo vraćene knjige**, jer za posudbe koje još traju ne možemo izračunati trajanje (DatumVracanja je NULL).

Ovo poboljšava točnost prosjeka i sprječava greške u funkciji DATEDIFF.

* + GROUP BY kat.Naziv;

Grupira rezultate po svakoj kategoriji knjiga kako bi se izračunao zaseban prosjek zadržavanja za svaku.

| **Problem / potreba** | **Kako pogled pomaže** |
| --- | --- |
| Koje se knjige najduže zadržavaju | Kategorije s duljim prosječnim zadržavanjem mogu ukazivati na složenije ili rjeđe knjige |
| Planiranje trajanja posudbi | Knjižnica može prilagoditi maksimalno trajanje posudbe po kategoriji |
| Nabava dodatnih primjeraka | Ako korisnici dugo zadržavaju knjige u određenoj kategoriji, možda treba više primjeraka |
| Identifikacija potencijalne zloporabe | Neuobičajeno dugo zadržavanje može ukazivati na korisnike koji ne poštuju rokove |
| Usporedba čitanosti i složenosti kategorije | Dulje zadržavanje može značiti da je sadržaj zahtjevniji za čitanje |

|  |  |
| --- | --- |
| **Kategorija** | **ProsjecniBrojDana** |
| Horor | 70,7 |
| Avantura | 67,4 |
| Filozofski roman | 66,4 |
| Kriminalistika | 65,7 |
| Klasici | 60,9 |
| Fantastika | 58 |
| Psihološki triler | 57,6 |
| Povijesni roman | 57,5 |
| Ljubavni roman | 57,3 |
| Znanstvena fantastika | 49,7 |

# Upiti – Petra Tuškan (0058206931)

## Cilj mog modula

Dizajn i implementacija tablica: Recenzije, Primjeri te KasnoVracanje

Omogućeno je:

* procjena zadovoljstva članova knjižnice putem sustava za recenziranje knjiga
* uvid u fizičko stanje i dostupnost svake knjige
* praćenje kašnjenja i pripadajućih kazni

## Članovi koji su ostavili barem jednu recenziju poredano po broju recenzija i prezimenu člana

CREATE VIEW clanovi\_recenzije AS

SELECT

c.ClanID,

c.Ime,

c.Prezime,

COUNT(r.ClanID) AS broj\_recenzija

FROM

Clanovi c

INNER JOIN

Recenzije r ON c.ClanID = r.ClanID

GROUP BY

c.ClanID, c.Ime, c.Prezime

HAVING

COUNT(r.ClanID) >= 1

ORDER BY

broj\_recenzija DESC,

c.Prezime ASC;

**Detalji:**

* pomoću naredbe SELECT odabiremo ID člana knjižnice te njegovo ime i prezime iz tablice Clanovi (alias c), a pomoću COUNT agregatne funkcije brojimo recenzije koje je pojedini član ostavio
* INNER JOIN služi za spajanje tablice Recenzije s tablicom Clanovi preko zajedničkom stupca ClanID*.* Na taj način se osigurava da se u rezultat uključe isključivo oni članovi koji su napisali barem jednu recenziju
* GROUP BY agregatnu funkciju moramo koristiti nakon korištenja agregatne funkcije COUNT. Ona služi za grupiranje podataka po c.ClanID, c.Ime i c.Prezime stupcima
* HAVING naredba se koristi nakon grupiranja i služi za filtriranje rezultata tako da ostaju samo oni za koje je broj recenzija minimalno 1
* iako je u ovom konkretnom slučaju HAVING suvišan (jer s obzirom na spajanje s tablicom Recenzije, svaki član automatski ima minimalno jednu recenziju), ipak će biti ostavljen jer bi u budućnosti moglo doći do nekih promjena u uvjetima
* ORDER BY sortira rezultate na način da se prvo sortiraju po broju recenzija u silaznom redoslijedu, a ako više članova ima isti broj recenzija, tada se oni dodatno sortiraju abecednim redoslijedom prema prezimenu

Primjer korištenja u praksi:

Knjižnica želi potaknuti članove na aktivnije ostavljanje recenzija s ciljem stvaranja korisničkih preporuka i povratnih informacija o knjigama.

Jedan od načina da se to postigne je organizacija nagradne igre u kojoj sudjeluju svi članovi koji su ostavili barem jednu recenziju.

## Knjige s barem jednim kašnjenjem čiji je prosjek ocjena manji od 3

CREATE VIEW kasnjenja\_niske\_ocjene AS

SELECT

k.KnjigaID,

k.Naslov,

AVG(r.Ocjena) AS prosjecna\_ocjena,

COUNT(DISTINCT kv.KasnoVracanjeID) AS broj\_kasnjenja

FROM

Knjige k

INNER JOIN Posudbe p ON k.KnjigaID = p.KnjigaID

INNER JOIN KasnoVracanje kv ON p.PosudbaID = kv.PosudbaID

INNER JOIN Recenzije r ON k.KnjigaID = r.KnjigaID

GROUP BY

k.KnjigaID, k.Naslov

HAVING

COUNT(DISTINCT kv.KasnoVracanjeID) >= 1

AND AVG(r.Ocjena) < 3;

**Detalji:**

* pomoću naredbe SELECT odabiremo ID i naslov knjige iz tablice Knjige (alias k). Također, pomoću agregatne funkcije AVG izračunavamo prosječnu ocjenu koju je knjiga dobila (iz tablice Recenzije), dok se COUNT(DISTINCT kv.KasnoVracanjeID) koristi za brojanje koliko je puta neka knjiga vraćena sa zakašnjenjem
* Tri INNER JOIN naredbe spajaju 4 tablice:
* tablicu Knjige s tablicom Posudbe preko zajedničkog stupca KnjigaID kako bi se dobile sve posudbe povezane s knjigama
* tablicu Posudbe s tablicom KasnoVracanje preko zajedničkog stupca PosudbaID kako bi se pronašla kašnjenja u vraćanju
* tablicu Knjige s tablicom Recenzije putem stupca KnjigaID kako bi se prikupile sve ocjene vezane za pojedinu knjigu
* GROUP BY osigurava da rezultat sadrži agregirane vrijednosti (broj kašnjenja, prosjek ocjena) **za svaki jedinstveni naslov knjige**
* **HAVING filtrira rezultate tako da prikazuje samo one knjige koje:**
* imaju barem jedno zakašnjelo vraćanje
* imaju prosječnu ocjenu recenzije manju od 3
* iako i u ovom slučaju možemo pretpostaviti da će spajanje s tablicom KasnoVracanje automatski ukloniti knjige bez kašnjenja, uvjet je ipak naveden zbog bolje čitljivosti i eventualne otpornosti na buduće izmjene

Primjer korištenja u praksi:

Ovaj upit pomaže knjižnici u identifikaciji "problematičnih" knjiga.

Knjižnica želi analizirati koje knjige nisu dobro prihvaćene među korisnicima, a uz to ih korisnici i često vraćaju sa zakašnjenjem. Te informacije mogu poslužiti kao temelj za donošenje odluka o uklanjaju pojedinih naslova iz fonda, njihovoj zamjeni boljim izdanjima, ili eventualno marketinškim kampanjama za promociju "nepopularnih" knjiga.

## Naslovi knjiga sortirani po pripadajućim žanrovima i prosječnim ocjenama

CREATE VIEW Knjige\_po\_ocjenama\_i\_zanrovima AS

SELECT

k.KnjigaID AS Knjiga\_ID,

k.Naslov AS Naslov\_knjige,

kat.Naziv AS Zanr,

AVG(r.Ocjena) AS Prosjecna\_ocjena

FROM

Knjige k

INNER JOIN Kategorije kat ON k.KategorijaID = kat.KategorijaID

LEFT JOIN Recenzije r ON k.KnjigaID = r.KnjigaID

GROUP BY

k.KnjigaID, k.Naslov, kat.Naziv

ORDER BY

kat.Naziv ASC,

Prosjecna\_ocjena DESC;

**Detalji:**

* pomoću naredbe SELECT odabiremo naslov i ID knjige iz tablice Knjige (alias k), naziv žanra iz tablice Kategorije (alias kat), te prosječnu ocjenu koju je knjiga dobila, koristeći agregatnu funkciju AVG() nad ocjenama iz tablice Recenzije
* INNER JOIN se koristi za povezivanje tablica Knjige i Kategorije preko stupca KategorijaID. Time dohvaćamo podatak kojem žanru pripada pojedina knjiga (budući da svaka knjiga mora imati žanr, koristimo unutarnje spajanje)
* LEFT JOIN spaja tablicu Recenzije s tablicom Knjige prema stupcu KnjigaID. Ovaj tip spajanja omogućuje da u rezultat budu uključene i one knjige koje još nemaju nijednu recenziju (njihova prosječna ocjena će biti NULL)
* GROUP BY koristimo nakon korištenja agregatne funkcije AVG() za grupiranje rezultata po ID-u knjige, naslovu i žanru
* ORDER BY sortira rezultate prvo abecedno po nazivu žanra, a zatim po prosječnoj ocjeni unutar svakog žanra. Na taj način se unutar svakog žanra prvo prikazuju najbolje ocijenjene knjige

Primjer korištenja u praksi:

Knjižnica želi pratiti koje knjige su najpopularnije unutar svakog žanra, kako bi korisnicima mogla preporučiti naslove prema njihovim interesima.

## Knjige sa statusom "dostupno" poredane po imenu autora

CREATE VIEW dostupne\_knjige AS

SELECT

k.KnjigaID,

k.Naslov,

k.Status,

a.ImePrezime AS Autor

FROM

Knjige k,

KnjigaAutori ka,

Autori a

WHERE

k.KnjigaID = ka.KnjigaID

AND ka.AutorID = a.AutorID

AND k.Status = 'dostupno'

ORDER BY

a.ImePrezime ASC;

**Detalji:**

* pomoću SELECT naredbe dohvaćamo ID knjige, njen naslov i status iz tablice Knjige (alias k) te dohvaćamo ime i prezime autora iz tablice Autori (alias a)
* u ovom upitu umjesto INNER JOIN naredbe koristimo drugačiju sintaksu gdje se sve tablice navode u FROM dijelu, a uvjeti spajanja pišu se u WHERE dijelu
* pomoću WHERE uvjeta povezujemo tablicu Knjige s tablicom KnjigaAutori preko KnjigaID, te povezujemo tablicu Knjiga Autori s tablicom Autori preko AutorID kako bismo dohvatili podatke o autorima povezanim s knjigama.
* uvjet k.Status = 'dostupno' osigurava da u rezultat budu uključene samo knjige dostupne za posudbu
* ORDER BY sortira konačni rezultat prema imenu i prezimenu autora u abecednom redoslijedu

Primjer korištenja u praksi:

Knjižnica želi brzo generirati popis svih trenutno dostupnih knjiga organiziran po autorima.

Na taj način pomaže u usluzi korisnicima na info pultu ili web katalogu te boljoj organizaciji fonda knjižnice.

## Knjige sa oštećenim primjercima poredanih po broju oštećenih primjeraka i imenu autora

CREATE VIEW popis\_ostecenih\_knjiga AS

SELECT

k.KnjigaID,

k.Naslov,

a.ImePrezime AS Autor,

COUNT(p.PrimjerID) AS broj\_ostecenih

FROM

Knjige k

INNER JOIN Primjeri p ON k.KnjigaID = p.KnjigaID

INNER JOIN

KnjigaAutori ka ON k.KnjigaID = ka.KnjigaID

INNER JOIN

Autori a ON ka.AutorID = a.AutorID

WHERE

p.Status = 'Oštećeno'

GROUP BY

k.KnjigaID, k.Naslov, a.ImePrezime

ORDER BY

broj\_ostecenih DESC, a.ImePrezime ASC;

**Detalji:**

* SELECT naredbom odabiremo ID i naslov knjige iz tablice Knjige (alias k), ime i prezime autora iz tablice Autori (alias a), te broj oštećenih primjeraka koji se računa pomoću agregatne funkcije COUNT() iz tablice Primjeri
* INNER JOIN između tablica Knjige i Primjeri preko stupca KnjigaID omogućuje pristup svim primjercima određene knjige
* INNER JOIN **s tablicom** KnjigaAutori povezuje knjige s njihovim autorima putem many-to-many veze. Ova tablica omogućuje da knjiga ima više autora i obratno
* INNER JOIN **s tablicom** Autori omogućuje dohvat imena i prezimena autora
* WHERE uvjet filtrira samo one primjerke knjiga koji imaju status 'oštećeno'
* GROUP BY se koristi nakon COUNT() funkcije. Svi oštećeni primjerci iste knjige i autora se zbrajaju
* na kraju ORDER BY prvo sortira rezultat po broju oštećenih primjeraka u silaznom redoslijedu, a zatim sortira abecedno po imenu autora (u slučaju da više knjiga ima isti broj oštećenih primjeraka)

Primjer korištenja u praksi:

Knjižnica koristi ovaj pogled kako bi identificirala knjige koje su fizički najviše oštećene. Na temelju rezultata može planirati obnavljanje knjižnog fonda.

Također se mogu raditi neke analize kao što su praćenje oštećenja po nakladnicima ili popularnosti autora.

## Knjige s oštećenim primjercima i prosječnom ocjenom nižom od 3

CREATE VIEW ostecene\_knjige\_s\_losim\_recenzijama AS

SELECT DISTINCT

k.KnjigaID,

k.Naslov,

p.Status,

AVG(r.Ocjena) AS prosjecna\_ocjena

FROM

Knjige k,

Primjeri p,

Recenzije r

WHERE

k.KnjigaID = p.KnjigaID

AND k.KnjigaID = r.KnjigaID

AND p.Status = 'Oštećeno'

GROUP BY

k.KnjigaID, k.Naslov, p.Status

HAVING

AVG(r.Ocjena) < 3

ORDER BY

prosjecna\_ocjena ASC;

**Detalji:**

* **pomoću naredbe** SELECT dohvaćamo ID i naslov knjige iz tablice Knjige (alias k), status primjerka iz tablice Primjeri (alias p) te prosječnu ocjenu koju je knjiga dobila, koristeći agregatnu funkciju AVG() nad ocjenama iz tablice Recenzije (alias r)
* DISTINCT unutar SELECT naredbe služi za uklanjanje potencijalnih duplikata
* FROM dio upita povezuje tablice Knjige, Primjeri i Recenzije, koristeći uvjete u WHERE klauzuli. Knjiga se povezuje i s primjercima i s recenzijama preko KnjigaID
* uvjet p.Status = 'Oštećeno' unutar WHERE filtrira primjerke knjiga koji su oštećeni
* GROUP BY se koristi nakon AVG() te grupira rezultat po naslovu i statusu primjerka
* HAVING dodatno filtrira oštećene rezultate tako da u konačni rezultat ulaze samo one knjige koje imaju prosječnu ocjenu manju od 3
* na kraju ORDER BY sortira oštećene knjige tako da se prvo prikažu knjige s najnižom prosječnom ocjenom

Primjer korištenja u praksi:

Knjižnica koristi ovaj pogled za redovitu reviziju svog knjižnog fonda kako bi se osiguralo da su naslovi u ponudi fizički ispravni i zanimljivi korisnicima.

Ovo je jedan od načina kako da se objektivno prepoznaju knjige koje potencijalno nebi trebale biti dio fonda.

# Upiti – Maja Kovačević (0009074625)

# Cilj mog modula

U sklopu timskog projekta "Sustav za upravljanje knjižnicom", moj zadatak je bio osmisliti i implementirati modul koji pokriva evidenciju događaja u knjižnici, povezanost događaja s određenim knjigama, te vođenje inventara knjiga. Kroz ove tri povezane tablice prikazan je dodatni sadržaj i funkcionalnost koju jedna suvremena knjižnica može nuditi korisnicima.

Modul omogućuje:

* Centralizirano čuvanje podataka o događajima knjižnice.
* Praćenje izlaganja knjiga na događajima.
* Evidentiranje fizičkih primjeraka knjiga (inventar) i statistički uvidi u stanje inventara.

Knjige dodane u posljednjih godinu dana

Prikazuje sve stavke iz knjižničnog inventara koje su nabavljene u posljednjih 12 mjeseci. Koristan je za praćenje novih unosa u fond knjižnice, godišnje izvještaje o nabavi, identifikaciju novih knjiga i opreme za promotivne aktivnosti.

CREATE VIEW dodano\_u\_zadnjih\_god\_dana AS

SELECT \*

FROM inventar

WHERE DatumNabavke >= CURDATE() - INTERVAL 1 YEAR;

SELECT \*

FROM dodano\_u\_zadnjih\_god\_dana

WHERE Stanje = 'Oštećeno';

**Detalji:**

* CREATE VIEW dodano\_u\_zadnjih\_god\_dana AS – kreira pogled naziva dodano\_u\_zadnjih\_god\_dana
* SELECT \* FROM Inventar – odabire sve atribute iz tablice Inventar, uključujući InventarID, KnjigaID, DatumNabavke i Stanje
* WHERE DatumNabavke >= CURDATE() - INTERVAL 1 YEAR – filtrira stavke koje su nabavljene unutar zadnjih godinu dana od današnjeg datuma (CURDATE())
* WHERE Stanje = 'Oštećeno' – filtrira samo oštećene stavke među novim primjercima

Prikaz stanja knjiga, neovisno o naslovima

Pogled knjige\_stanje prikazuje koliko se fizičkih primjeraka knjiga nalazi u svakom stanju, poput *Novo, Oštećeno, Izgubljeno,* itd. Ovaj pogled je koristan za brzu provjeru stanja fonda knjižnice, prepoznavanje postoji li potreba za obnovom inventara te izvještavanje o stanju imovine knjižnice.

CREATE VIEW knjige\_stanje AS

SELECT

i.Stanje,

COUNT(\*) AS BrojKnjiga

FROM Inventar i

GROUP BY i.Stanje;

**Detalji:**

* CREATE VIEW knjige\_stanje AS - kreira pogled s nazivom knjige\_stanje
* SELECT i.Stanje, COUNT(\*) AS BrojKnjiga:
  + - * Stanje – opis stanja primjerka (Novo, Oštećeno);
* BrojKnjiga – broj primjeraka u tom stanju
* FROM Inventar i - izvlači podatak iz tablice Inventar
* GROUP BY i.Stanje - grupira rezultate po stanju kako bi dobili broj za svaku kategoriju

## Naslovi knjiga grupirani prema stanju

Detaljan pregled naslova svih knjiga prema njihovom stanju i datumu nabavke. Daje mogućnost identifikacije stanja po naslovima, korisno za obnovu fonda.

CREATE VIEW naslovi\_po\_stanju AS

SELECT

k.Naslov,

i.Stanje,

i.DatumNabavke

FROM Inventar i

JOIN Knjige k ON i.KnjigaID = k.KnjigaID

ORDER BY i.Stanje, i.DatumNabavke;

SELECT \* FROM naslovi\_po\_stanju

WHERE Stanje = 'Oštećeno'

ORDER BY DatumNabavke DESC;

**Detalji:**

* CREATE VIEW naslovi\_po\_stanju AS – kreira novi pogled naslovi\_po\_stanju
* SELECT k.Naslov, i.Stanje, i.DatumNabavke:
* Naslov - naziv knjige; Stanje - fizičko stanje knjige (Novo, Oštećeno, Izgubljeno);
* DatumNabavke - kada je primjerak kupljen/nabavljen
* FROM Inventar JOIN Knjige – spaja tablice Inventar (fizičke jedinice) i Knjige (naslovi)
* ORDER BY i.Stanje, i.DatumNabavke – sortiranje po stanju i datumu

Prikaz naslova knjiga i pripadajućih događaja

Prikazuje koji su naslovi uključeni u koje događaje. Korisno za pregled književnih večeri, radionica itd.

CREATE VIEW dogadaji\_s\_naslovima AS

SELECT

d.NazivDogadaja,

d.Datum,

k.Naslov AS NaslovKnjige

FROM KnjigaDogadaji kd

JOIN Dogadaji d ON kd.DogadajID = d.DogadajID

JOIN Knjige k ON kd.KnjigaID = k.KnjigaID

ORDER BY d.Datum;

SELECT \* FROM dogadaji\_s\_naslovima

ORDER BY Datum DESC;

SELECT \* FROM dogadaji\_s\_naslovima

WHERE NazivDogadaja = 'Noć knjige';

**Detalji:**

* CREATE VIEW dogadaji\_s\_naslovima AS – kreira pogled naziva dogadaji\_s\_naslovima
* SELECT d.NazivDogadaja, d.Datum, k.Naslov AS NaslovKnjige:
* NazivDogadaja – npr. “Noć knjige”
* Datum – kada se događaj odvija
* NaslovKnjige: knjiga koja se koristi
* FROM KnjigaDogadaji JOIN Dogadaji JOIN Knjige - povezuje veznu tablicu s događajima i naslovima.
* ORDER BY d.Datum – rezultati sortirani kronološki
* ORDER BY Datum DESC – sortira rezultate tako da su najnoviji događaji prikazani prvi
* WHERE NazivDogadaja = 'Noć knjige' – filtrira rezultate tako da prikazuje samo naslove knjiga koji su bili dio događaja "Noć knjige"

## Prikaz broja knjiga uključenih u događaj

Prikazuje koliko je knjiga povezano s pojedinim događajem u knjižnici. Korisno pri procjeni veličine i opsega događaja.

CREATE VIEW broj\_knjiga\_po\_dogadaju AS

SELECT

d.NazivDogadaja,

d.Datum,

COUNT(kd.KnjigaID) AS BrojKnjiga

FROM Dogadaji d

LEFT JOIN KnjigaDogadaji kd ON d.DogadajID = kd.DogadajID

GROUP BY d.DogadajID, d.NazivDogadaja, d.Datum;

SELECT \* FROM broj\_knjiga\_po\_dogadaju

ORDER BY Datum DESC, BrojKnjiga DESC

LIMIT 10;

SELECT \* FROM broj\_knjiga\_po\_dogadaju

WHERE BrojKnjiga > 1;

**Detalji:**

* CREATE VIEW broj\_knjiga\_po\_dogadaju AS – kreira pogled
* SELECT d.NazivDogadaja, d.Datum, COUNT(kd.KnjigaID) AS BrojKnjiga:
* NazivDogadaja i Datum – osnovne info o događaju
* BrojKnjiga – ukupan broj povezanih knjiga
* LEFT JOIN – za prikaz događaja koji još nemaju dodijeljene knjige. Koristi se kako bi se prikazali i događaji koji nemaju još dodijeljene knjige.
* GROUP BY – za grupiranje po jedinstvenim događajima
* SELECT \* FROM broj\_knjiga\_po\_dogadaju – dohvaća sve retke iz pogleda broj\_knjiga\_po\_dogadaju
* ORDER BY Datum DESC, BrojKnjiga DESC – rezultati su sortirani po datumu silazno/descending, tako da su najnoviji događaji prvi. Ako više događaja ima isti datum, onda se dodatno sortiraju po broju knjiga (više knjiga → više prioriteta)
* LIMIT 10 – ograničava prikaz na samo 10 najnovijih i najvećih događaja
* WHERE BrojKnjiga > 1 – prikazuje samo one događaje u kojima je uključeno više od jedne knjige

# Zaključak

U sklopu ovog projekta implementiran je kompletan modul baze podataka za upravljanje knjižnicom.

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. Pogledi omogućuju i pregled najnovih izdanja knjiga po autoru, pregled knjiga koje su bile dio raznih događaja, praćenje rezencija oštećenih knjiga, praćenje članova – njihovih posudbi, kazni te recenzija, praćenje trajanja posudbi te popularnosti kategorija, praćenje trenutno dostupnih knjiga. Nabrojeni su samo mali dio pogleda koji naš sustav omogućuje, ostatak je prikazan u dokumentaciji i kodu.

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 zadovoljili smo sve funkcionalne i tehničke zahtjeve, a baza je spremna za daljnji razvoj i integraciju s korisničkim sučeljem ili aplikacijom.