**PROJEKAT iz predmeta “BAZE PODATAKA 3”**

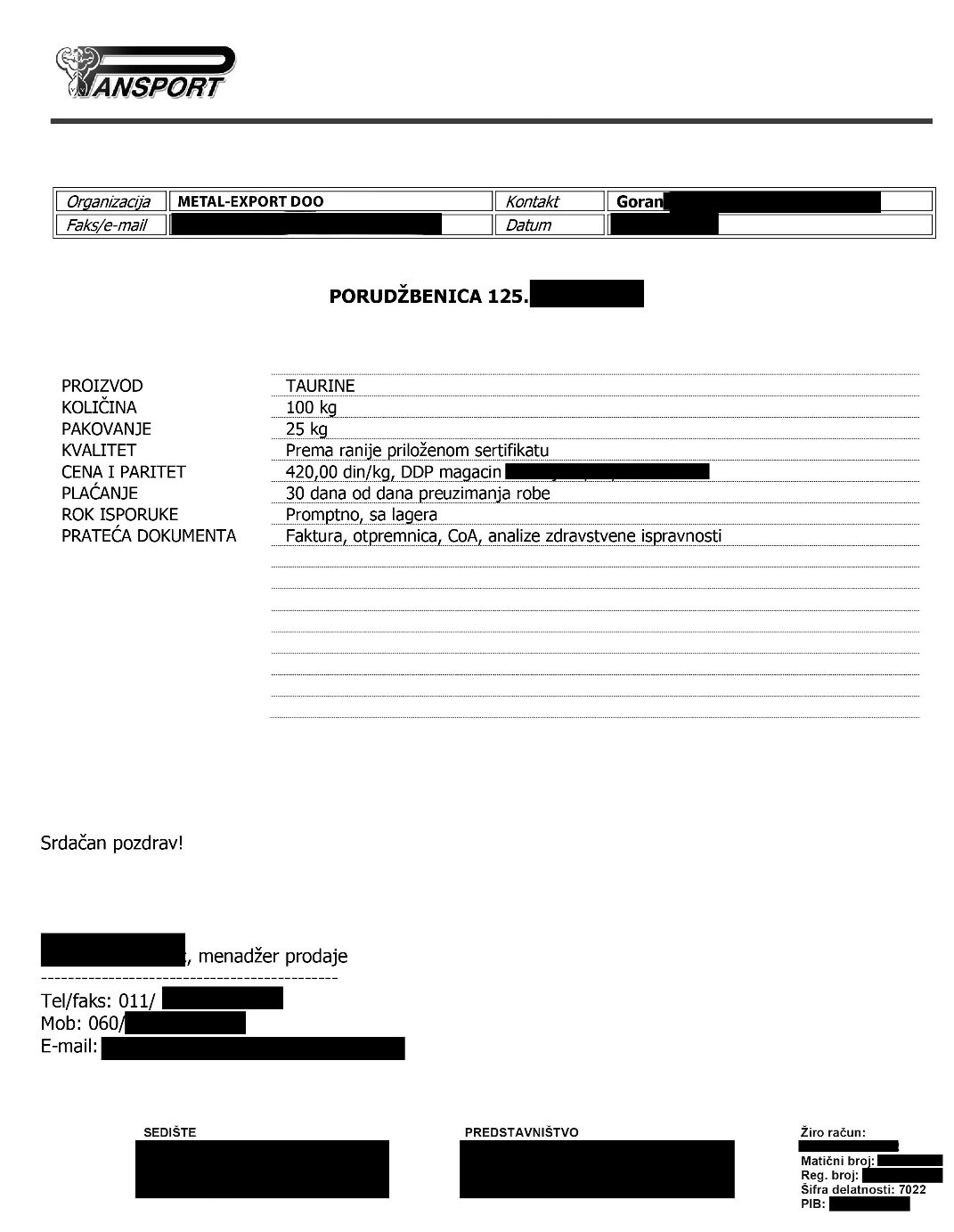
|  |  |  |  |
| --- | --- | --- | --- |
| **IME I PREZIME** | | **Anja Đorđević** | |
| Broj indeksa | | **2023/3027** | |
| Studijski modul | | Informacione tehnologije i sajber bezbednost | |
| Stečena diploma osnovnih akademskih studija | | Diplomirani inženjer organizacionih nauka | |
| **NAZIV PROJEKTA** | | Informacioni sistem za proces dobavljanja suplementacije u kompaniji Pansport | |
| Napomena | |  | |
| *Verzija* | *Datum promene* | | *Opis promene* | |
| 1. | 1. 11. 2023. | | **Opis projekta i dokumenti** | |
| 2. | 7. 11. 2023. | | **PMOV i relacioni model 1. verzija** | |
| 3. | 13. 11. 2023. | | **PMOV i relacioni model 2. verzija, denormalizacija 1. verzija** | |
| 4. | 20. 11. 2023. | | **Denormalizacija 2. verzija** | |
| 5. | 27. 11. 2023. | | **Korisnički definisani tipovi, implementacija trigera** | |
| 6. | 20. 12. 2023. | | **Optimizacija baze podataka** | |
| 7. | 9. 1. 2024. | | **Tehnologije za implementaciju projekta** | |

# OPIS PROJEKTA

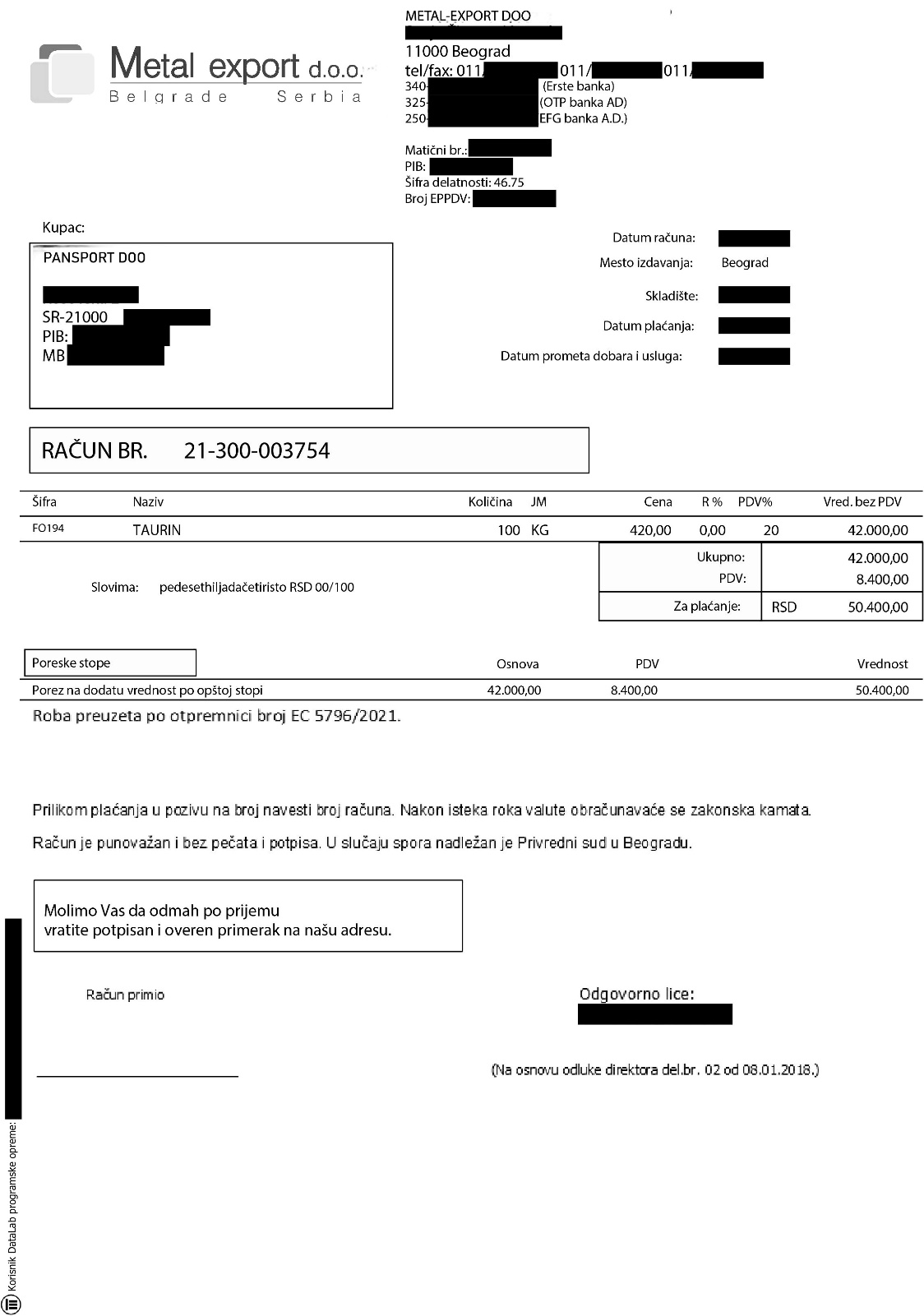
## Opis podsistema za koji se projektuje baza podataka

|  |
| --- |
| Deo sistema koji će biti projektovan odnosi se na proces dobavljanja suplementa Taurine u kompaniji Pansport. Kompanija sastavlja **porudžbenicu** za dobavljača Metal-Export, koji zatim isporučuju **račun, otpremnicu** i **izveštaj** o ispitivanju materijala od strane kompanije SGS. Na samom kraju procesa, Pansport popunjava **prijemnicu**. |

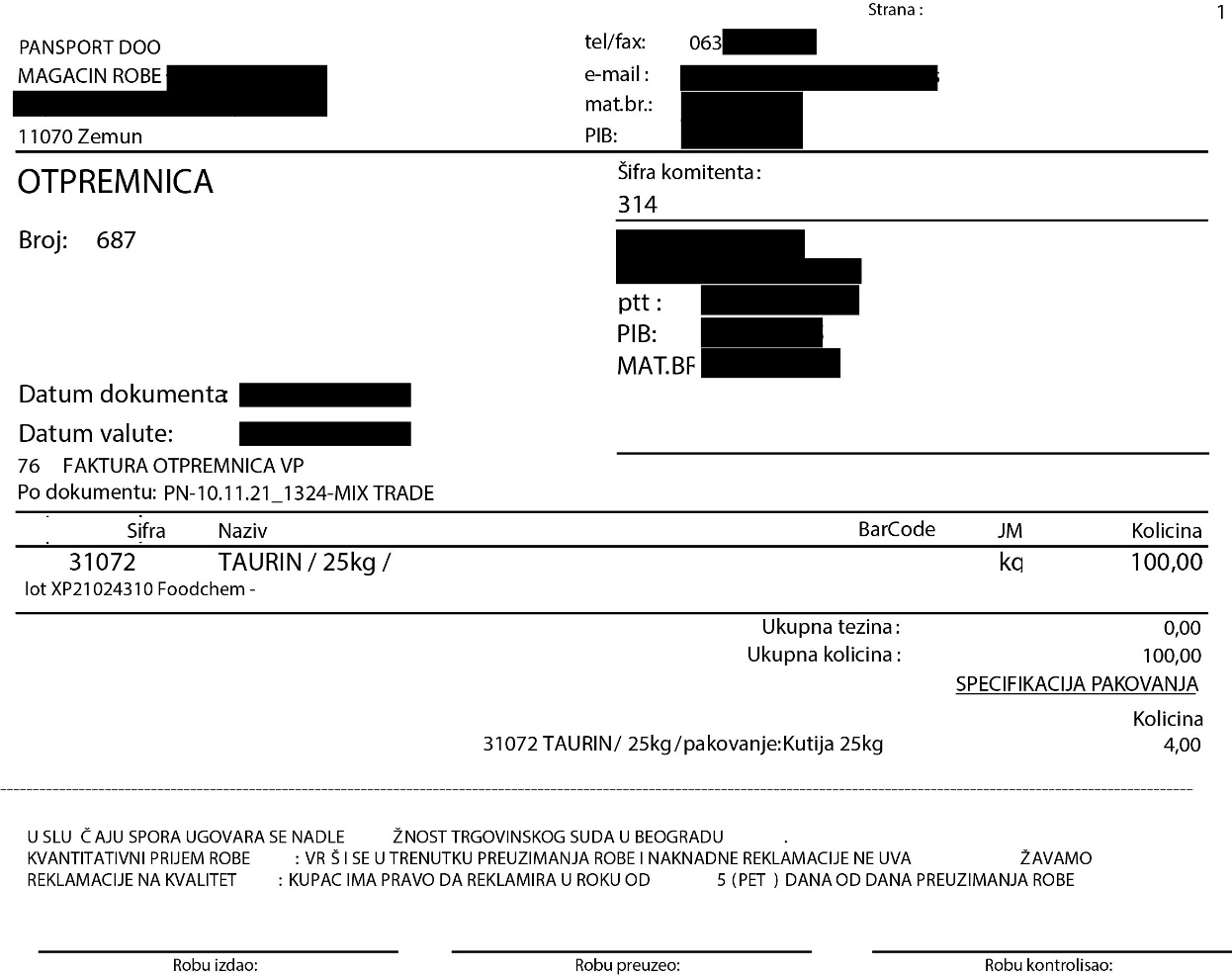
## Poslovna dokumenta koja se koriste u procesu



Slika 1 - Porudžbenica



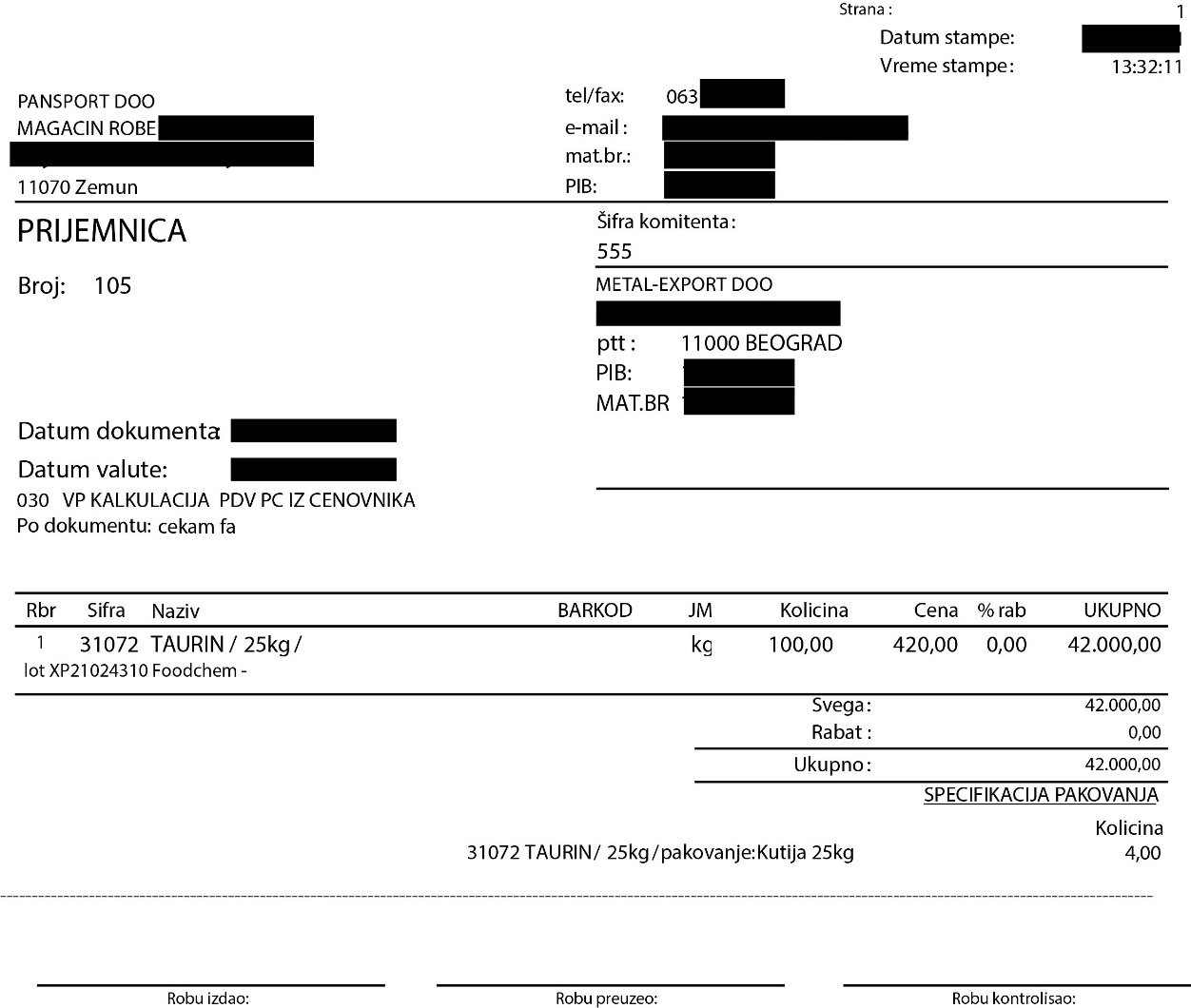
Slika 2 – Račun



Slika 3 – Otpremnica



Slika 4 - Izveštaj o ispitivanju

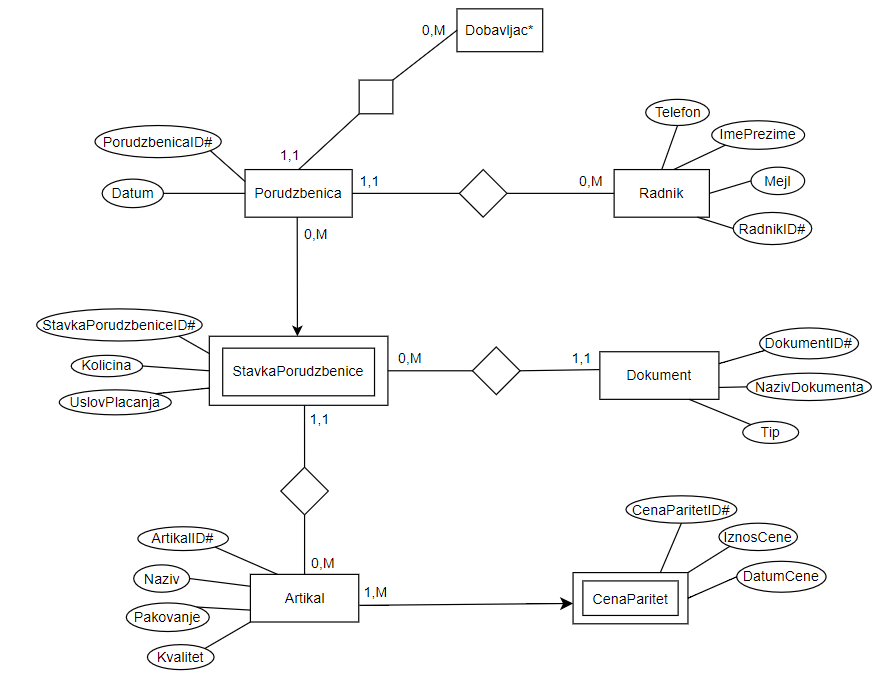


Slika 5 - Prijemnica

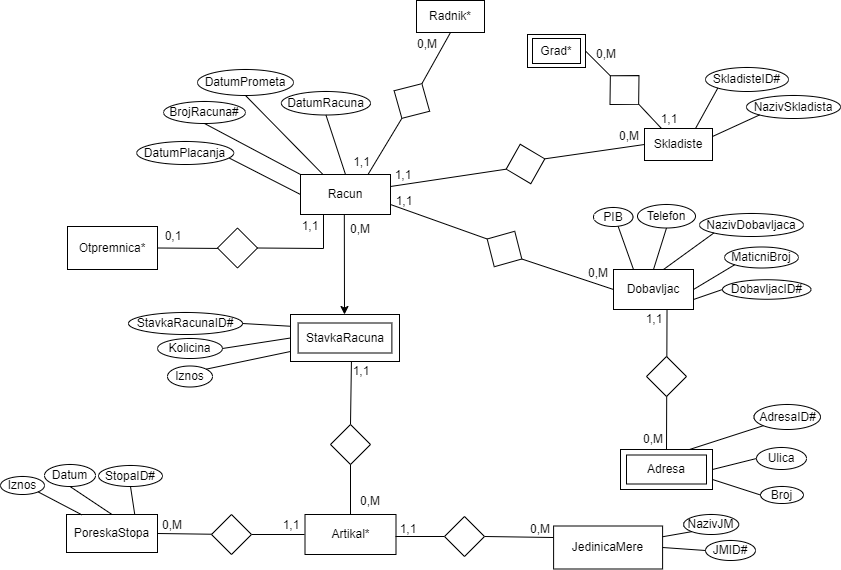
# PROJEKTOVANJE BAZE PODATAKA

## Model podatka

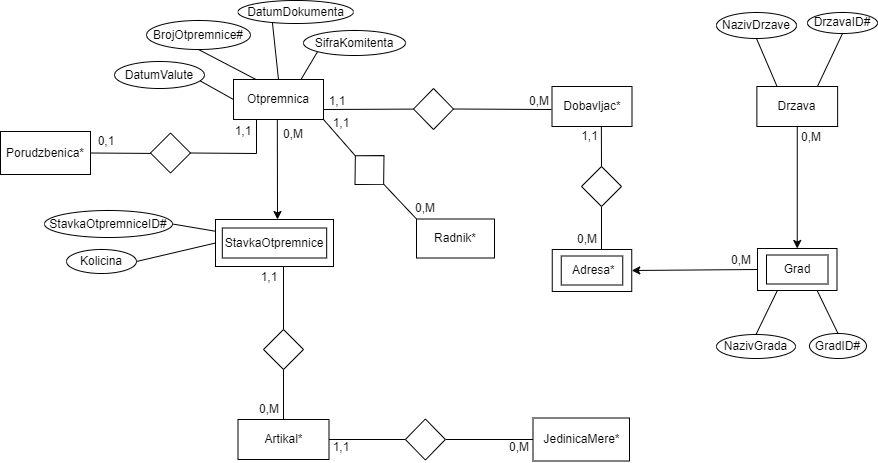
Porudžbenica



Račun



Otpremnica



Izveštaj o ispitivanju

A diagram of a company

Description automatically generated

Prijemnica

A diagram of a diagram

Description automatically generated

## Relacioni model

**Porudzbenica** (porudzbenicaID, datum, *radnikID, dobavljacID*)

**StavkaPorudzbenice** (porudzbenicaID, stavkaPorudzbeniceID, kolicina, uslovPlacanja, *artikalID*)

**Radnik** (radnikID, imePrezime, telefon, mejl)

**Dokument** (dokumentID, nazivDokumenta, tip, *porudzbenicaID, stavkaPorudzbeniceID*)

**Artikal** (artikalID, naziv, pakovanje, kvalitet, *JMID, stopaID*)

**CenaParitet** (artikalID, cenaParitetID, iznosCene, datumCene)

**Racun** (brojRacuna, datumRacuna, datumPlacanja, datumPrometa, *dobavljacID, skladisteID, brojOtpremnice, radnikID*)

**StavkaRacuna** (brojRacuna, stavkaRacunaID, iznos, kolicina, *artikalID*)

**PoreskaStopa** (stopaID, iznos, datum)

**Dobavljac** (dobavljacID, nazivDobavljaca, telefon, maticniBroj, PIB, *adresaID, drzavaID, gradID*)

**JedinicaMere** (JMID, nazivJM)

**Skladiste** (skladisteID, nazivSkladista, *drzavaID, gradID*)

**Adresa** (adresaID, drzavaID, gradID, ulica, broj)

**Otpremnica** (brojOtpremnice, datumDokumenta, datumValute, sifraKomitenta, *dobavljacID, porudzbenicaID, radnikID*)

**StavkaOtpremnice** (brojOtpremnice, stavkaOtpremniceID, kolicina, *artikalID*)

**Grad** (drzavaID, gradID, nazivGrada)

**Drzava** (drzavaID, nazivDrzave)

**Izvestaj** (brojIzvestaja, laboratorijskiBroj, *uzorakID, labID*)

**StavkaIzvestaja** (brojIzvestaja, stavkaIzvestajaID, *metodaID, parametarID, ispitivanjeID*)

**ReferentnaVrednost** (metodaID, parametarID, dozvoljenaVrednost, *JMID*)

**MetodaIspitivanja** (metodaID, nazivMetode)

**Uzorak** (uzorakID, nazivUzorka, datumUzorka)

**Laboratorija** (labID, PIB, telefon, *adresaID, drzavaID, gradID*)

**MikrobioloskiParametar** (parametarID, nazivParametra)

**Ispitivanje** (metodaID, parametarID, ispitivanjeID, datumPocetkaIspitivanja, datumZavrsetkaIspitivanja, utvrdjenaVrednost)

**Prijemnica** (brojPrijemnice, datumDokumenta, datumValute, datumVremeStampe, *dobavljacID, otpremnicaID*)

**StavkaPrijemnice** (brojPrijemnice, stavkaPrijemniceID, iznos, kolicina)

# DENORMALIZACIJA RELACIJA

## Denormalizacija relacija uz narušavanje 2NF – pre-joining tehnika

**Pre denormalizacije**



**Adresa** (adresaID, drzavaID, gradID, ulica, broj)

**Grad** (drzavaID, gradID, nazivGrada)

Funkcionalna zavisnost: adresaID, drzavaID, gradID -> ulica, broj

**Posle denormalizacije**

## 

**Adresa** (adresaID, drzavaID, gradID, ulica, broj, nazivGrada)

**Grad** (drzavaID, gradID, nazivGrada)

Funkcionalne zavisnosti: adresaID, drzavaID, gradID -> ulica, broj, nazivGrada

drzavaID, gradID -> nazivGrada

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| Grad | Insert |  | Ne |  |
| Update | nazivGrada | Da | Nakon izmene vrednosti kolone **nazivGrada** u tabeli **Grad**, pokreće se triger koji izmenjenu vrednost ažurira u tabeli **Adresa**. |
| Delete |  | Ne |  |
| Adresa | Insert |  | Da | U kolonu **nazivGrada** tabele **Adresa** upisuje unetu vrednost kolone **naziv** tabele **Grad** na osnovu vrednosti **gradID**. |
| Update | nazivGrada | Da | Zabraniti direktno ažuriranje ove kolone. |
| gradID | Nakon izmene vrednosti kolone **gradID**, pokreće se triger koji menja vrednost u koloni **nazivGrada**. |
| Delete |  | Ne |  |

## Denormalizacija relacija uz narušavanje 3NF – pre-joining tehnika

**Pre denormalizacije**

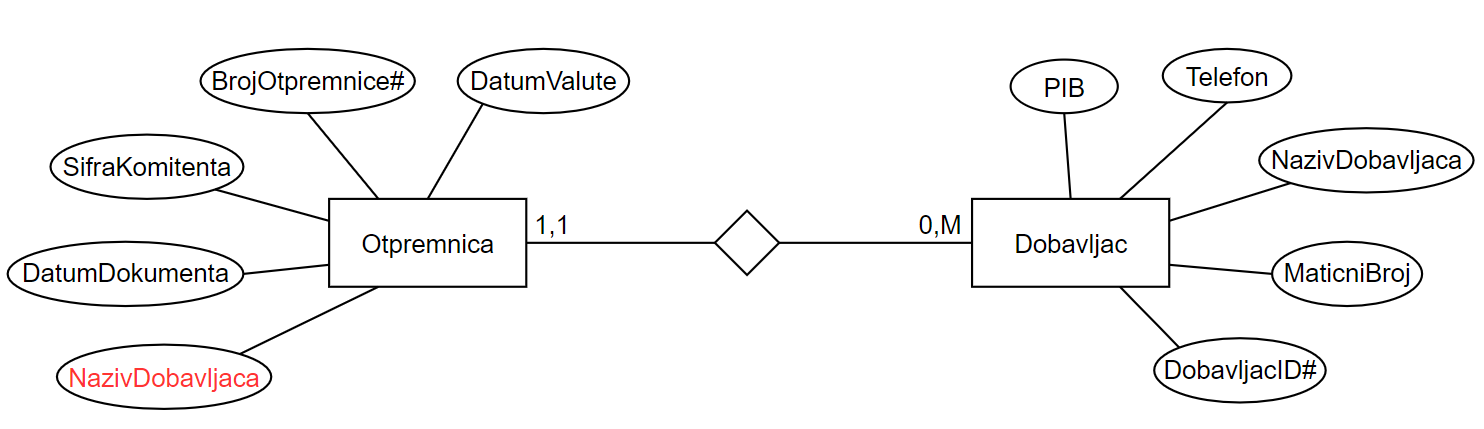
A white background with black and white text

Description automatically generated with medium confidence

**Otpremnica** (brojOtpremnice, datumDokumenta, datumValute, sifraKomitenta, *dobavljacID, porudzbenicaID, radnikID*)

**Dobavljac** (dobavljacID, nazivDobavljaca, telefon, maticniBroj, PIB, *adresaID, drzavaID, gradID*)

Funkcionalna zavisnost: brojOtpremnice -> datumDokumenta, datumValute, sifraKomitenta, dobavljacID, porudzbenicaID, radnikID

**Posle denormalizacije**

**Otpremnica** (brojOtpremnice, datumDokumenta, datumValute, sifraKomitenta, nazivDobavljaca, *dobavljacID, porudzbenicaID, radnikID*)

**Dobavljac** (dobavljacID, nazivDobavljaca, telefon, maticniBroj, PIB, *adresaID, drzavaID, gradID*)

Funkcionalne zavisnosti: brojOtpremnice -> datumDokumenta, datumValute, sifraKomitenta, nazivDobavljaca, dobavljacID, porudzbenicaID, radnikID

dobavljacID -> nazivDobavljaca

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| Dobavljac | Insert |  | Ne |  |
| Update | nazivDobavljaca | Da | Prilikom promene vrednosti kolone **nazivDobavljaca** u tabeli **Dobavljac**,pokreće se triger koji promenjenu vrednost ažurira i u tabeli **Otpremnica**. |
| Delete |  | Ne |  |
| Otpremnica | Insert |  | Da | U kolonu **nazivDobavljaca** tabele **Otpremnica** upisuje unetu vrednost kolone **nazivDobavljaca** tabele **Dobavljac** na osnovu vrednosti **dobavljacID**. |
| Update | nazivDobavljaca | Da | Zabraniti direktno ažuriranje ove kolone. |
| dobavljacID | Nakon izmene vrednosti kolone **dobavljacID**, pokreće se triger koji menja vrednost u koloni **nazivDobavljaca**. |
| Delete |  | Ne |  |

## Denormalizacija relacija uz narušavanje 3NF – short-circuit keys tehnika

**Pre denormalizacije**

**A black line with a diamond

Description automatically generated**

**Racun** (brojRacuna, datumRacuna, datumPlacanja, datumPrometa, *dobavljacID, skladisteID, brojOtpremnice, radnikID*)

**Skladiste** (skladisteID, nazivSkladista, *drzavaID, gradID*)

**Grad** (drzavaID, gradID, nazivGrada)

Funkcionalna zavisnost: brojRacuna -> datumRacuna, datumPlacanja, datumPrometa, dobavljacID, skladisteID, brojOtpremnice, radnikID

**Posle denormalizacije**

**A line graph with a red line

Description automatically generated**

**Racun** (brojRacuna, datumRacuna, datumPlacanja, datumPrometa, *dobavljacID, skladisteID, brojOtpremnice, radnikID, drzavaID, gradID*)

**Skladiste** (skladisteID, nazivSkladista, *drzavaID, gradID*)

**Grad** (drzavaID, gradID, nazivGrada)

Funkcionalne zavisnosti: brojRacuna -> datumRacuna, datumPlacanja, datumPrometa, dobavljacID, skladisteID, brojOtpremnice, radnikID, drzavaID, gradID

skladisteID -> drzavaID, gradID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| Skladiste | Insert |  | Ne |  |
| Update | gradID | Da | Ukoliko postoji **Racun** za dato **Skladiste**, ažurira *gradID* u tabeli **Racun** na zadatu vrednost. |
| Delete |  | Ne |  |
| Racun | Insert |  | Da | U kolonu *gradID* tabele **Racun** upisuje se vrednost kolone *gradID* iz tabele **Skladiste** na osnovu vrednosti kolone *skladisteID*. |
| Update | skladisteID | Da | U kolonu *gradID* tabele **Racun** upisuje se odgovarajuća vrednost iz kolone *gradID* tabele **Skladiste** na osnovu nove vrednosti za *skladisteID*. |
| gradID | Sprečava direktnu izmenu. |
| drzavaID | Sprečava direktnu izmenu. |
| Delete |  | Ne |  |

# KORISNIČKI DEFINISANI TIPOVI I TRIGERI

## Definicija korisničkom tipa i njegovo korišćenje

**STRUKTUIRANI TIP**

CREATE OR REPLACE TYPE CenaParitetTip AS OBJECT (

iznosCene DECIMAL (10,2),

datumCene DATE,

MEMBER FUNCTION getIznosCene RETURN DECIMAL,

MEMBER FUNCTION getDatumCene RETURN DATE,

CONSTRUCTOR FUNCTION CenaParitetTip(iznosCene DECIMAL, datumCene DATE)

RETURN SELF AS RESULT)

INSTANTIABLE NOT FINAL;

CREATE OR REPLACE TYPE BODY CenaParitetTip AS

MEMBER FUNCTION getIznosCene RETURN DECIMAL IS

BEGIN

RETURN SELF.iznosCene;

END;

MEMBER FUNCTION getDatumCene RETURN DATE IS

BEGIN

RETURN SELF.datumCene;

END;

CONSTRUCTOR FUNCTION CenaParitetTip(iznosCene DECIMAL, datumCene DATE)

RETURN SELF AS RESULT IS

BEGIN

IF iznosCene <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Iznos cene mora biti veci od 0.');

END IF;

SELF.iznosCene := iznosCene;

SELF.datumCene := datumCene;

RETURN;

END;

END;

CREATE TABLE CenaParitet (

artikalID NUMBER,

cenaParitetID NUMBER,

naziv VARCHAR2(50),

cena CenaParitetTip,

CONSTRAINT PK\_CENAPARITET2 PRIMARY KEY (artikalID, cenaParitetID),

CONSTRAINT FK\_CENAPARITET2 FOREIGN KEY (artikalID) REFERENCES ARTIKAL (artikalID)

);

**DISTINCT TIP**

CREATE OR REPLACE TYPE PIBTip AS OBJECT (

oznakaPIB VARCHAR2(9),

MEMBER FUNCTION getPIBTip RETURN VARCHAR2)

INSTANTIABLE FINAL;

CREATE OR REPLACE TYPE BODY PIBTip AS

MEMBER FUNCTION getPIBTip return VARCHAR2 IS

BEGIN

RETURN SELF.oznakaPIB;

END;

END;

CREATE TABLE Dobavljac (

dobavljacID NUMBER,

nazivDobavljaca VARCHAR2 (50),

telefon VARCHAR2 (15),

maticniBroj VARCHAR2 (30),

PIB PIBTip,

adresaID NUMBER,

drzavaID NUMBER,

gradID NUMBER,

constraint PK\_Dobavljac2 PRIMARY KEY (dobavljacID),

constraint FK\_Dobavljac2 FOREIGN KEY (adresaID, drzavaID, gradID) REFERENCES Adresa (adresaID, drzavaID, gradID),

check (REGEXP\_LIKE (PIB.oznakaPIB,'[0-9]{9}'))

);

## Realizacija proceduralne logike uz pomoć trigera

Implementacija trigera definisanih u specifikaciji trigera prilikom narušavanja 2NF:

1. Triger koji postavlja vrednost kolone *nazivGrada* u tabeli Adresa na osnovu vrednosti *gradID*

CREATE OR REPLACE TRIGGER TR\_ADRESA\_INSERT\_NAZIV\_GRADA

BEFORE INSERT

ON Adresa

FOR EACH ROW

DECLARE

v\_nazivGrada varchar2 (50);

BEGIN

SELECT g.nazivGrada

INTO v\_nazivGrada

FROM Grad g

WHERE g.gradID = :NEW.gradID;

:NEW.nazivGrada := v\_nazivGrada;

END;

1. Triger koji sprečava direktnu izmenu kolone *nazivGrada* tabele Adresa

CREATE OR REPLACE TRIGGER TR\_ADRESA\_FORBID\_UPDATE\_NAZIV

BEFORE UPDATE OF nazivGrada ON Adresa

FOR EACH ROW

BEGIN

IF :NEW.nazivGrada != :OLD.nazivGrada THEN

RAISE\_APPLICATION\_ERROR (-20005, 'Zabranjeno azuriranje kolone nazivGrada');

END IF;

END;

1. Triger koji sprečava direktnu izmenu kolone *gradID* tabele Adresa

CREATE OR REPLACE TRIGGER TR\_ADRESA\_FORBIDUPDATE\_IDGRADA

BEFORE UPDATE OF gradID ON Adresa

FOR EACH ROW

BEGIN

IF :NEW.gradID != :OLD.gradID THEN

RAISE\_APPLICATION\_ERROR (-20006, 'Zabranjeno azuriranje kolone gradID');

END IF;

END;

1. Triger koji prilikom promene vrednosti kolone *nazivGrada* tabele Grad, postavlja vrednost kolone *nazivGrada* odgovarajućim rekordima tabele Adresa na osnovu spoljnog ključa *gradID*

CREATE OR REPLACE TRIGGER TR\_GRAD\_UPDATE\_NAZIV

AFTER UPDATE OF nazivGrada ON Grad

FOR EACH ROW

DECLARE

v\_noviNaziv VARCHAR2(50);

v\_gradID NUMBER;

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

v\_noviNaziv := :NEW.nazivGrada;

v\_gradID := :NEW.gradID;

EXECUTE IMMEDIATE 'ALTER TRIGGER TR\_ADRESA\_FORBID\_UPDATE\_NAZIV DISABLE';

UPDATE Adresa

SET nazivGrada = v\_noviNaziv

WHERE gradID = v\_gradID;

EXECUTE IMMEDIATE 'ALTER TRIGGER TR\_ADRESA\_FORBID\_UPDATE\_NAZIV ENABLE';

COMMIT;

END;

Implementacija trigera definisanih u specifikaciji trigera prilikom narušavanja 3NF (short-circuit keys):

1. Triger koji prilikom izmene kolone *gradID* tabele Skladiste, postavlja odgovarajućim računima vrednost kolone *gradID* na novu na osnovu vrednosti spoljnog ključa *skladisteID*

CREATE OR REPLACE TRIGGER TR\_SKLADISTE\_UPDATE\_GRAD

AFTER UPDATE OF gradID ON Skladiste

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

IF :NEW.gradID != :OLD.gradID THEN

EXECUTE IMMEDIATE 'ALTER TRIGGER TR\_RACUN\_FORBID\_UPDATE\_GRAD DISABLE';

UPDATE Racun

SET Racun.gradID = :NEW.gradID

WHERE Racun.skladisteID = :NEW.skladisteID;

EXECUTE IMMEDIATE 'ALTER TRIGGER TR\_RACUN\_FORBID\_UPDATE\_GRAD ENABLE';

END IF;

COMMIT;

END;

1. Triger koji prilikom unosa novog računa postavlja vrednost kolone *gradID* tabele Racun na odgovarajuću iz tabele Skladiste na osnovu spoljnog ključa *skladisteID*

CREATE OR REPLACE TRIGGER TR\_RACUN\_INSERT

BEFORE INSERT ON Racun

FOR EACH ROW

DECLARE

v\_gradID NUMBER;

v\_drzavaID NUMBER;

BEGIN

SELECT gradID INTO v\_gradID

FROM Skladiste

WHERE skladisteID = :NEW.skladisteID;

:NEW.gradID := v\_gradID;

SELECT drzavaID INTO v\_drzavaID

FROM Skladiste

WHERE skladisteID = :NEW.skladisteID;

:NEW.drzavaID := v\_drzavaID;

END;

1. Triger koji prilikom promene vrednosti kolone *skladisteID* tabele Racun ažurira vrednost kolone *gradID* na odgovarajuću iz tabele Skladiste na osnovu nove vrednosti spoljnog ključa *skladisteID*

CREATE OR REPLACE TRIGGER TR\_RACUN\_UPDATE\_SKLADISTE

BEFORE UPDATE OF skladisteID ON Racun

FOR EACH ROW

DECLARE

v\_gradID NUMBER;

BEGIN

IF :NEW.skladisteID != :OLD.skladisteID THEN

SELECT gradID INTO v\_gradID

FROM Skladiste

WHERE skladisteID = :NEW.skladisteID;

:NEW.gradID := v\_gradID;

END IF;

END;

1. Triger koji zabranjuje direktnu izmenu vrednosti kolone *gradID* tabele Racun

CREATE OR REPLACE TRIGGER TR\_RACUN\_FORBID\_UPDATE\_GRAD

BEFORE UPDATE OF gradID ON Racun

FOR EACH ROW

BEGIN

IF :NEW.gradID != :OLD.gradID THEN

RAISE\_APPLICATION\_ERROR (-20009, 'Zabranjeno azuriranje kolone gradID');

ROLLBACK;

END IF;

END;

Implementacija trigera definisanih u specifikaciji trigera prilikom narušavanja 3NF (pre-joining):

1. Triger koji prilikom promene vrednosti kolone nazivDobavljaca tabele Dobavljac, postavlja vrednost kolone nazivDobavljaca odgovarajućim rekordima tabele Otpremnica na osnovu spoljnog ključa dobavljacID

CREATE OR REPLACE TRIGGER TR\_DOBAVLJAC\_UPDATE\_NAZIV

AFTER UPDATE OF NazivDobavljaca ON Dobavljac

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

EXECUTE IMMEDIATE 'ALTER TRIGGER TR\_OTPR\_FORBID\_UPDATE\_NAZIV DISABLE';

UPDATE Otpremnica

SET nazivDobavljaca = :NEW.nazivDobavljaca

WHERE dobavljacID = :OLD.dobavljacID;

EXECUTE IMMEDIATE 'ALTER TRIGGER TR\_OTPR\_FORBID\_UPDATE\_NAZIV ENABLE';

COMMIT;

END;

1. Triger koji postavlja vrednost kolone nazivDobavljacau tabeli Otpremnica na osnovu vrednosti dobavljacID

CREATE OR REPLACE TRIGGER TR\_OTPREMNICA\_INSERT

BEFORE INSERT ON Otpremnica

FOR EACH ROW

DECLARE

v\_nazivDobavljaca VARCHAR2(50);

BEGIN

SELECT nazivDobavljaca INTO v\_nazivDobavljaca

FROM Dobavljac

WHERE dobavljacID = :NEW.dobavljacID;

:NEW.nazivDobavljaca:=v\_nazivDobavljaca;

END;

1. Triger koji sprečava direktnu izmenu kolone nazivDobavljaca tabele Otpremnica

CREATE OR REPLACE TRIGGER TR\_OTPR\_FORBID\_UPDATE\_NAZIV

BEFORE UPDATE OF nazivDobavljaca ON Otpremnica

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR (-20000, 'Nije dozvoljena direktna izmena naziva dobavljaca u okviru tabele Otpremnica.');

ROLLBACK;

END;

1. Triger koji nakon izmene vrednosti kolone dobavljacID, menja vrednost u koloni nazivDobavljaca

CREATE OR REPLACE TRIGGER TR\_OTPR\_UPDATE\_ID

BEFORE UPDATE OF dobavljacID ON Otpremnica

FOR EACH ROW

DECLARE

v\_nazivDobavljaca VARCHAR2(40);

BEGIN

SELECT d.nazivDobavljaca INTO v\_nazivDobavljaca

FROM Dobavljac D

WHERE d.dobavljacID = :NEW.dobavljacID;

:NEW.nazivDobavljaca := v\_nazivDobavljaca;

END;

# OPTIMIZACIJA BAZE PODATAKA

## Definisanje i analiza indeksa

## Indeks nad tekstualnim poljem

Usled učestale potrebe za pretragom naziva dobavljača u okviru otpremnice, uočena je mogućnost optimizacije.

Upit:



Kreiranje indeksa:



Plan izvršenja pre kreiranja indeksa:

A screenshot of a computer

Description automatically generated

Plan izvršenja nakon kreiranja indeksa:

A screenshot of a computer

Description automatically generated

## Indeks nad spoljnim ključem

Upit:



Kreiranje indeksa:



Plan izvršenja pre kreiranja indeksa:

A screenshot of a computer

Description automatically generated

Plan izvršenja nakon kreiranja indeksa:

A screenshot of a computer

Description automatically generated

## Horizontalno particionisanje

CREATE TABLE ARTIKAL2 (

artikalID NUMBER,

naziv varchar2(50),

pakovanje varchar2(20),

kvalitet varchar2(100),

jmID NUMBER,

stopaID NUMBER,

CONSTRAINT PK\_ARTIKAL PRIMARY KEY (artikalID),

CONSTRAINT FK\_ARTIKAL FOREIGN KEY (jmID) REFERENCES JEDINICAMERE (jmID),

FOREIGN KEY (stopaID) REFERENCES PORESKASTOPA (stopaID)

)

PARTITION BY RANGE(KVALITET)

(

PARTITION LosKvalitet VALUES LESS THAN (50),

PARTITION DobarKvalitet VALUES LESS THAN (100)

);

## Vertikalno particionisanje

Za potrebe ovog primera biće izvršeno vertikalno particionisanje tabele **Porudzbenica** na podatke koji se češće koriste (osnovni podaci) i one koji se smatraju detaljnijim informacijama kojima se ređe pristupa. Kreiran je i **pogled** kako bi apstrahovao rad sa particijama uz odgovarajući triger.

Kreiranje tabela:

A close up of text

Description automatically generated A close-up of a computer code

Description automatically generated

Kreiranje pogleda:

A close-up of a text

Description automatically generated

Kreiranje **INSTEAD OF** trigera nad pogledom za odgovarajuće **DML** naredbe:

A screenshot of a computer program

Description automatically generated

## 

## Primena drugih optimizacionih tehnika

**Hard-Coded Values**

Model pre optimizacije:

A diagram of a diagram

Description automatically generated

**Artikal** (artikalID, naziv, pakovanje, kvalitet, *JMID, stopaID*)

**JedinicaMere** (JMID, nazivJM)

Model posle optimizacije:

A diagram of a flowchart

Description automatically generated

**Artikal** (artikalID, naziv, pakovanje, kvalitet, nazivJM*, stopaID*)

Constraint koji se dodaje zbog hard-coded values tehnike optimizacije i naredbe koje se izvršavaju:

A close-up of a computer screen

Description automatically generated

**Storing Derivable Values**

Konceptualni model pre optimizacije:

A diagram of a company

Description automatically generated

**Relacioni model pre optimizacije:**

Racun (brojRacuna, datumRacuna, datumPlacanja, datumPrometa, *dobavljacID, mestoID, skladisteID*)

StavkaRacuna (brojRacuna, stavkaRacunaID, uslovPlacanja, iznos, kolicina, poreskaStopa, *artikalID*)

Konceptualni model posle optimizacije:

A diagram of a company

Description automatically generated

**Relacioni model posle optimizacije:**

Racun (brojRacuna, datumRacuna, datumPlacanja, datumPrometa, ukupno, *dobavljacID, mestoID, skladisteID*)

StavkaRacuna (brojRacuna, stavkaRacunaID, uslovPlacanja, iznos, kolicina, poreskaStopa, *artikalID*)

Tabela specifikacije trigera:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| **Racun** | Insert |  | NE |  |
| Update | Ukupno | DA | Sprečava direktnu izmenu polja ukupno. |
| Delete |  | NE |  |
| **StavkaRacuna** | Insert |  | DA | Prilikom unosa nove StavkeRacuna, okida se triger koji poziva proceduru za ažuriranje polja ukupno na Racunu. |
| Update | Kolicina | DA | Prilikom ažuriranja nove StavkeRacuna, okida se triger koji poziva proceduru za ažuriranje polja ukupno na Racunu. |
| Delete |  | DA | Prilikom brisanja StavkeRacuna, okida se triger koji poziva proceduru za ažuriranje polja ukupno na Racunu. |

A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated

# TEHNOLOGIJE ZA IMPLEMENTACIJU PROJEKTA

## SUBP korišćen za implementaciju baze podatka (kratak opis)

## U procesu implementacije baze podataka, Oracle je odabran kao ključni sistem za upravljanje podacima. Oracle se ističe zbog svoje izuzetne pouzdanosti, skalabilnosti i visokih performansi, čineći ga optimalnim izborom za zadovoljenje zahteva složenih poslovnih potreba. Korišćenje Oracle-a omogućava efikasno upravljanje podacima uz podršku za napredne funkcionalnosti, visoke sigurnosne standarde i sposobnost rukovanja kompleksnim transakcijama. Uz to, za razvoj i održavanje baze podataka koristi se SQL Developer, alat koji dodatno olakšava rad s Oracle bazom podataka putem intuitivnog korisničkog interfejsa i optimizacije SQL upita.

## Programsko okruženje za razvoj korisničkog interfejsa (kratak opis)

## Pri razvoju korisničkog interfejsa, korišćeno je programsko okruženje zasnovano na programskom jeziku Java. Aplikacija je implementirana u NetBeans15 razvojnom okruženju, uz upotrebu Swing biblioteke za kreiranje korisničkog interfejsa. Java je odabrana zbog svoje svestranosti i moćnih mogućnosti programiranja, pružajući stabilnost i performanse. NetBeans, kao integrirano razvojno okruženje, olakšava procese razvoja aplikacija. Kroz primenu Swing biblioteke postignuta je jednostavna izrada korisničkog interfejsa, prilagodljivog potrebama aplikacije, čime se korisnicima pruža intuitivno i efikasno iskustvo korišćenja.