

UNIVERZITET U BEOGRADU

FAKULTET ORGANIZACIONIH NAUKA

BAZE PODATAKA 2

- Projektni rad -

*„ Poslovni proces ~Prodaja~ u preduzeću VWG
Inženjering“*

Mentor: dr Slađan Babarogić

Student: **Vojislav Ristivojević 2016/3079**, email: batika@gmail.com

Beograd, januar 2017.

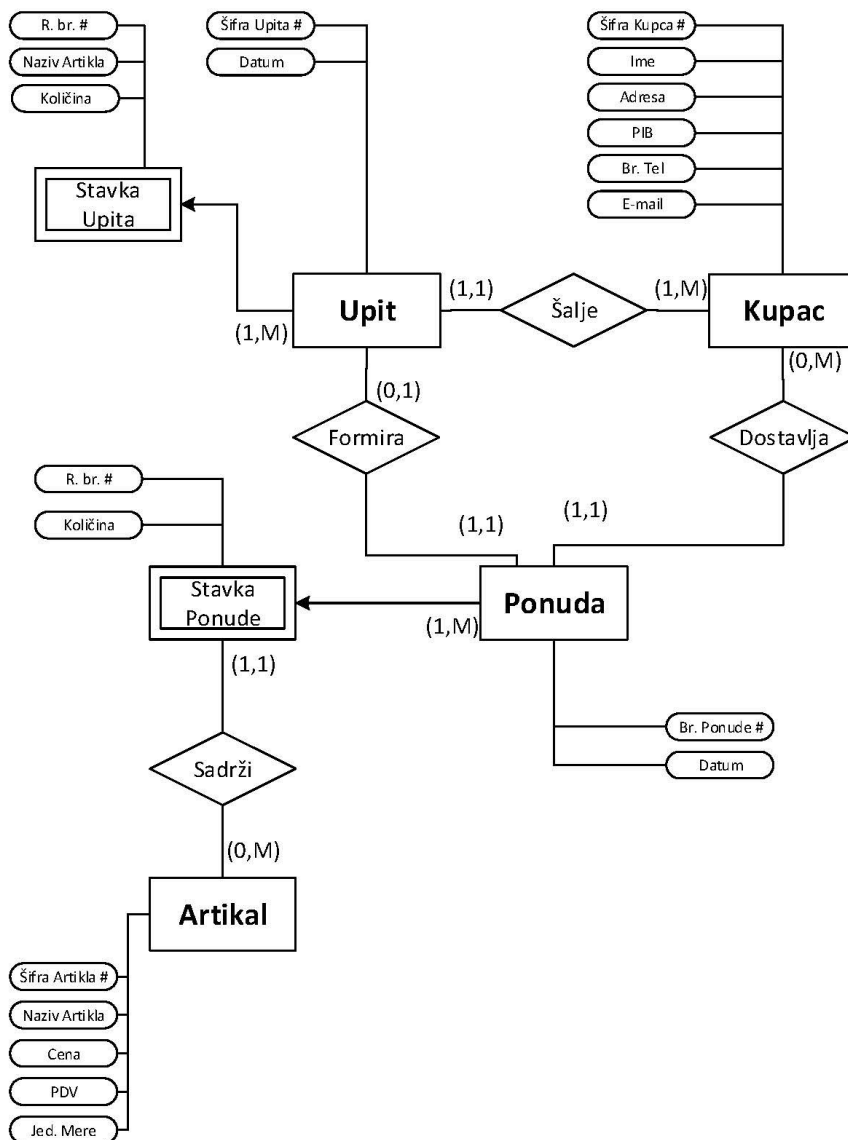
Verbalni opis

Preduzeće VWG Inženjering d.o.o. iz Beograda se bavi prodajom, instalacijom i održavanjem sistema tehničke zaštite, kao i pružanjem konsultantskih usluga iz oblasti fizičko-tehničke bezbednosti.

Prodaja proizvoda

Proces prodaje proizvoda počinje prijemom upita kupca koji pristiže elektronskom poštom ili telefonom. Na osnovu upita, unose se osnovni podaci kupca u bazu i kreira se ponuda koja se dostavljaju kupcu. Ponuda sadrži spisak traženih artikala sa aktuelnim cenama. Artikli su prethodno uneti u bazu i postavljene su im aktuelne cene.

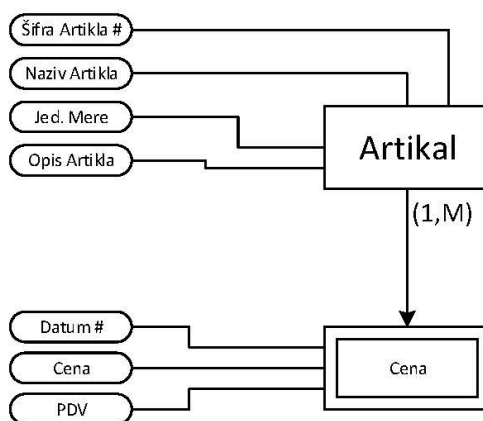
Vojislav Ristivojević
2016/3079



PMOV – VWG Inženjering

Podmodel: PRODAJA

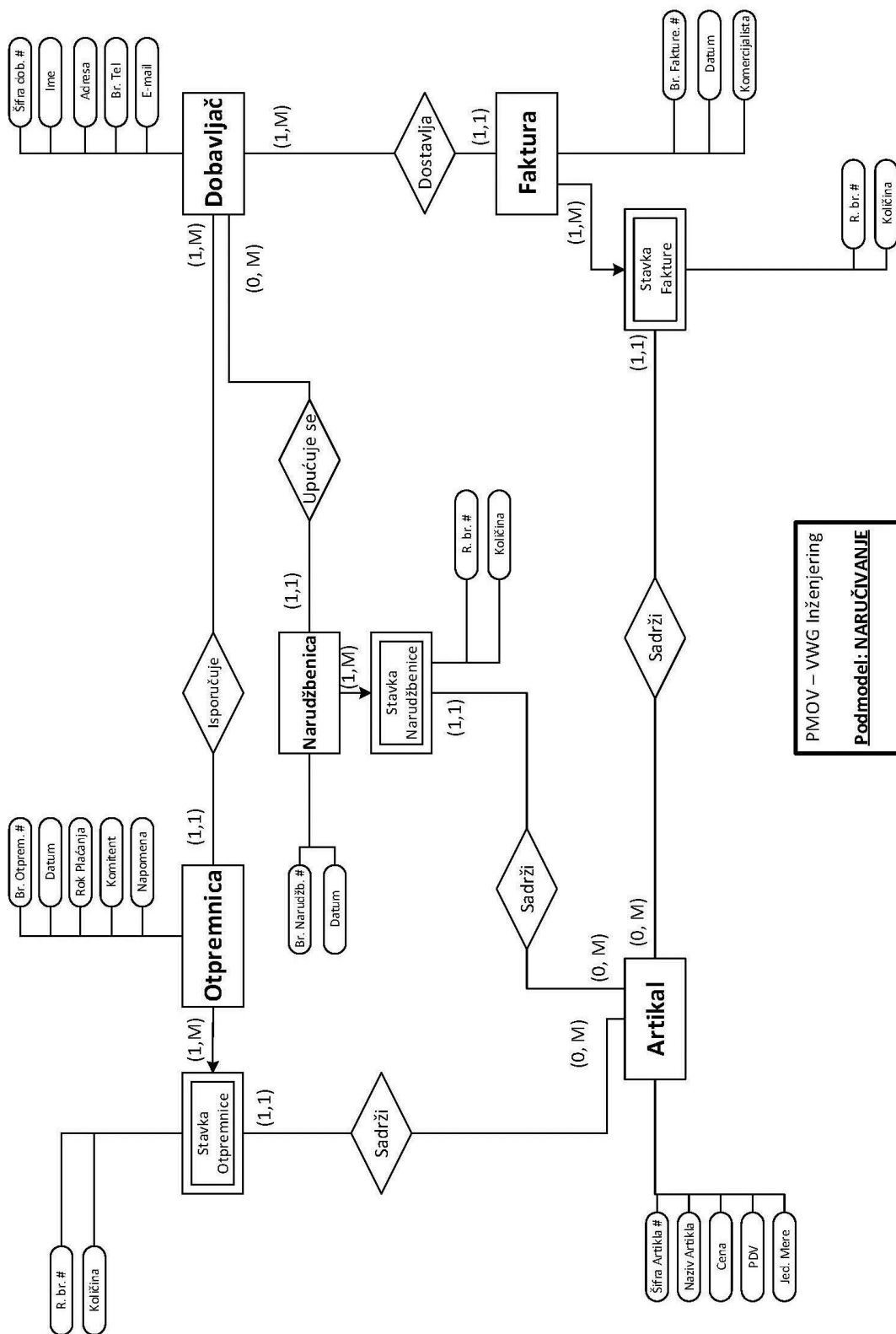
Vojislav Ristivojević
2016/3079



PMOV – VWG Inženjering

Podmodel: CENOVNIK

Vojislav Ristivojević
2016/3079



PMOV – VVWG Inženjering
Podmodel: NARUČIVANJE
 Vojislav Ristivojević
 2016/3079

Relacioni model - VWG Inženjering

Vojislav Ristivojević, 2016/3079

Podmodel : PRODAJA

Kupac (SifraKupca, Ime, Adresa, PIB, BrTel, email)

Upit (SifraUpita, *SifraKupca*, Datum)

StavkaUpita (Rbr, SifraUpita, NazivArtikla, Kolicina)

Ponuda (BrPonude, Datum, *SifraKupca*, *SifraUpita*)

StavkaPonude (Rbr, BrPonude, Kolicina, *SifraArtikla*)

Predracun (BrPredracuna, Datum, TipPlacanja, Uslovi, TekuciRn, Isporuka, *SifraKupca*, *BrPonude*)

StavkaPredracuna (Rbr, BrPredracuna, Kolicina, *SifraArtikla*)

IzvodStanja (Brlzvoda, Datum, Konto, ImeSubjekta, Duguje, Potražuje)

Artikal (SifraArtikla, NazivArtikla, Cena, PDV, JedMere)

Podmodel : NARUČIVANJE

Dobavljac (SifraDobavljacka, Ime, Adresa, BrTel, email)

Faktura (BrFakture, Datum, Komercijalista, *SifraDobavljacka*)

StavkaFakture (Rbr, BrFakture, Kolicina, *SifraArtikla*)

Otpremnica (BrOtpremnice, Datum, RokPlacanja, Komitent, Napomena, *SifraDobavljacka*)

StavkaOtpremnice (Rbr, BrOtpremnice, Kolicina, *SifraArtikla*)

Narudzbenica (BrNarudzbenice, Datum, *SifraDobavljacka*)

StavkaNarudzbenice (Rbr, BrNarudzbenice, Kolicina, *SifraArtikla*)

Artikal (SifraArtikla, NazivArtikla, Cena, PDV, JedMere)

Podmodel : CENOVNIK

Artikal (SifraArtikla, NazivArtikla, JedMere, OpisArtikla)

Cena (Datum, SifraArtikla, Cena, PDV)

DENORMALIZOVAN Relacioni model - VWG Inženjering

Vojislav Ristivojević, 2016/3079

Podmodel : PRODAJA

Kupac (SifraKupca, ImeKupca, Adresa, PIB, BrTel, email)

Upit (SifraUpita, *SifraKupca*, Datum)

StavkaUpita (Rbr, SifraUpita, NazivArtikla, Kolicina)

Ponuda (BrPonude, Datum, *SifraKupca*, *SifraUpita*, **ImeKupca**)

StavkaPonude (Rbr, BrPonude, Kolicina, *SifraArtikla*)

Predracun (BrPredracuna, Datum, TipPlacanja, Uslovi, TekuciRn, Isporuca, *SifraKupca*, *BrPonude*)

StavkaPredracuna (Rbr, BrPredracuna, Kolicina, *SifraArtikla*)

IzvodStanja (Brlzvoda, Datum, Konto, ImeSubjekta, Duguje, Potražuje)

Artikal (SifraArtikla, NazivArtikla, Cena, PDV, JedMere)

Podmodel : NARUČIVANJE

Dobavljac (SifraDobavljacka, ImeDobavljacka, Adresa, BrTel, email)

Faktura (BrFakture, Datum, Komercijalista, *SifraDobavljacka*)

StavkaFakture (Rbr, BrFakture, Kolicina, *SifraArtikla*)

Otpremnica (BrOtpremnice, Datum, RokPlacanja, Komitent, Napomena, *SifraDobavljacka*)

StavkaOtpremnice (Rbr, BrOtpremnice, Kolicina, *SifraArtikla*)

Narudzbenica (BrNarudzbenice, Datum, *SifraDobavljacka*)

StavkaNarudzbenice (Rbr, BrNarudzbenice, Kolicina, *SifraArtikla*)

Artikal (SifraArtikla, NazivArtikla, Cena, StopaPDV, JedMere)

Podmodel : CENOVNIK

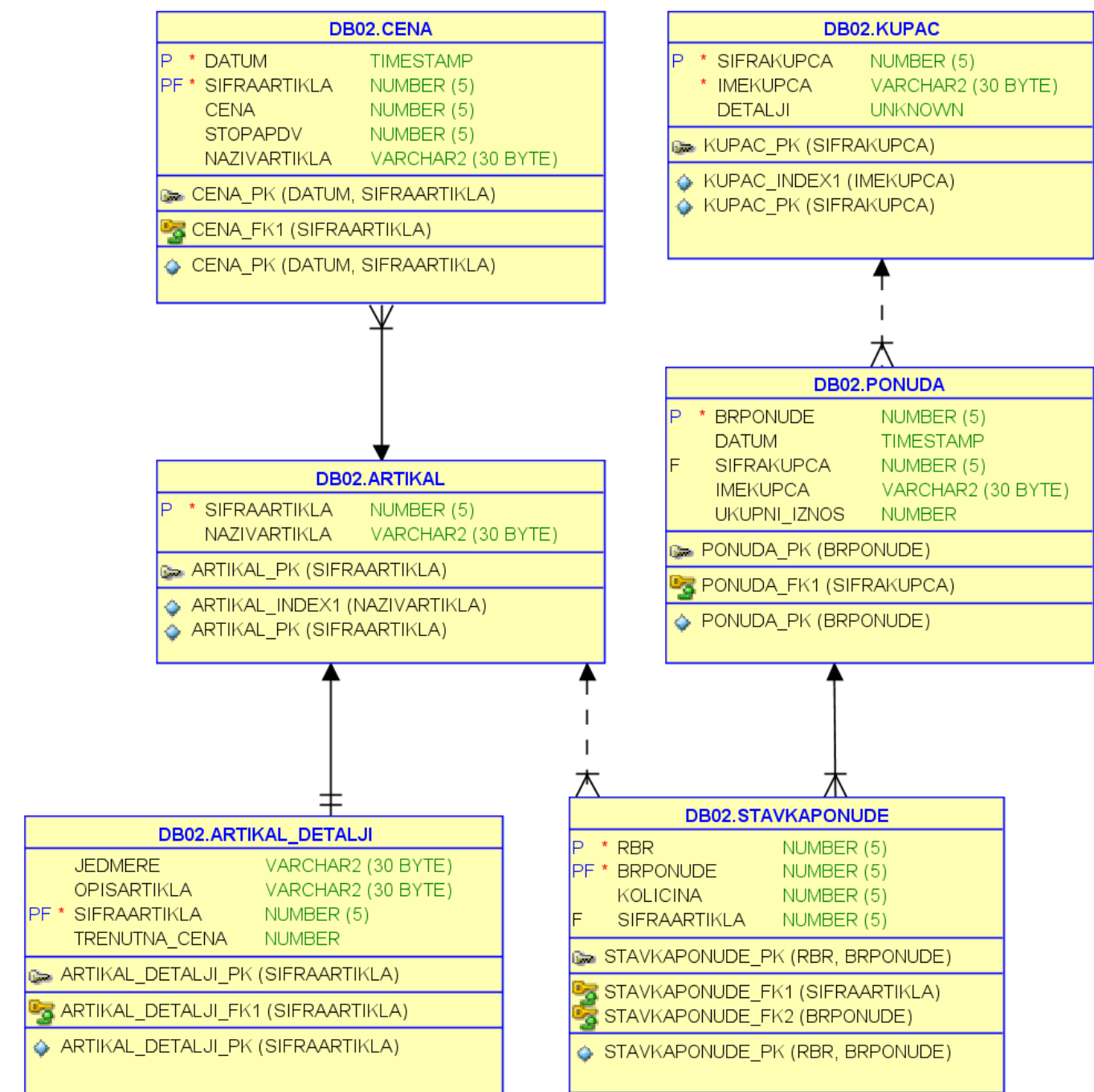
Artikal (SifraArtikla, NazivArtikla, JedMere, OpisArtikla)

Cena (Datum, SifraArtikla, Cena, StopaPDV, **NazivArtikla**)

Tabela trigeri - VWG Inženjering - Vojislav Ristivojević 2016/3079

Tabela	Tip trigeri	Kolona	Potreban	Šta treba da uradi?
Ponuda	Insert	ImeKupca	DA	Unosi ImeKupca iz tabele Kupac na osnovu SifraKupca
	Update	ImeKupca	DA	Sprečava ažuriranje
	Update	SifraKupca	DA	Ažurira ImeKupca na osnovu SifraKupca
Kupac	Insert		NE	
	Update	ImeKupca	DA	Ažurira ImeKupca u tabeli Ponuda
	Delete		NE	Rešeno referencijalnim integritetom

Tabela	Tip trigeri	Kolona	Potreban	Šta treba da uradi?
Artikal	Insert		NE	
	Update	NazivArtikla	DA	Ažurira NazivArtikla u tabeli Cena
	Delete		NE	Rešeno referencijalnim integritetom
Cena	Insert	NazivArtikla	DA	Unosi NazivArtikla iz tabele Artikal na osnovu SifraArtikla
	Update	NazivArtikla	DA	Sprečava ažuriranje
	Delete		NE	Rešeno referencijalnim integritetom



VERTIKALNO PARTICIONISANJE

-- DDL for Table ARTIKAL

```
CREATE TABLE "DB02"."ARTIKAL"
(
  "SIFRAARTIKLA" NUMBER(5,0),
  "NAZIVARTIKLA" VARCHAR2(30 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

-- DDL for Table ARTIKAL_DETALJI

```
CREATE TABLE "DB02"."ARTIKAL_DETALJI"
(
  "JEDMERE" VARCHAR2(30 BYTE),
  "OPISARTIKLA" VARCHAR2(30 BYTE),
  "SIFRAARTIKLA" NUMBER,
  "TRENUTNA_CENA" NUMBER
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

-- DDL for View ARTIKAL_VIEW

```
CREATE OR REPLACE FORCE VIEW "DB02"."ARTIKAL_VIEW" ("SIFRAARTIKLA", "NAZIVARTIKLA", "JEDMERE",
"OPISARTIKLA", "TRENUTNA_CENA") AS
select
a.sifraartikla, a.nazivartikla, ad.jedmere, ad.opisartikla, ad.TRENUTNA_CENA
from ARTIKAL a, ARTIKAL_DETALJI ad
where ad.SIFRAARTIKLA = a.SIFRAARTIKLA;
```

TRIGERI ZA VERTIKALNO PARTICIONISANJE

```
-----
-- DDL for Trigger ARTIKAL_INSERT
-----
```

```
CREATE OR REPLACE TRIGGER "DB02"."ARTIKAL_INSERT"
instead of insert ON artikal_view
for each row
declare x number;
begin

insert into artikal
(NAZIVARTIKLA) values (:NEW.NAZIVARTIKLA);

select sifraartikla
into x
from ARTIKAL
where NAZIVARTIKLA = :NEW.NAZIVARTIKLA;

insert into artikal_detalji
(jedmere, opisartikla, sifraartikla) values (:NEW.jedmere, :NEW.opisartikla, x);

end artikal_insert;
/
ALTER TRIGGER "DB02"."ARTIKAL_INSERT" ENABLE;
```

```
-----
-- DDL for Trigger ARTIKAL_UPDATE
-----
```

```
CREATE OR REPLACE TRIGGER "DB02"."ARTIKAL_UPDATE"
instead of update ON artikal_view
for each row
begin

update artikal set nazivartikla=:new.nazivartikla where sifraartikla = :new.sifraartikla;

end ARTIKAL_UPDATE;
/
ALTER TRIGGER "DB02"."ARTIKAL_UPDATE" ENABLE;
```

```
-----
-- DDL for Trigger ARTIKAL_DELETE
-----
```

```
CREATE OR REPLACE TRIGGER "DB02"."ARTIKAL_DELETE"
instead of delete ON artikal_view
for each row
begin

delete from artikal_detalji where sifraartikla = :old.sifraartikla;

delete from artikal where sifraartikla = :old.sifraartikla;

end ARTIKAL_DELETE;
/
ALTER TRIGGER "DB02"."ARTIKAL_DELETE" ENABLE;
```

INDEKSI

```
-----
-- DDL for Index ARTIKAL_INDEX1
-----
```

```
CREATE UNIQUE INDEX "DB02"."ARTIKAL_INDEX1" ON "DB02"."ARTIKAL" ("NAZIVARTIKLA")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

```
-----
-- DDL for Index KUPAC_INDEX1
-----
```

```
CREATE INDEX "DB02"."KUPAC_INDEX1" ON "DB02"."KUPAC" ("IMEKUPCA")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

USKLADIŠTENE PROCEDURE

-- DDL for Package SIFRA_PACK

```
CREATE OR REPLACE PACKAGE "DB02"."SIFRA_PACK" AS
```

```
sifra number;
```

```
brponude number;
```

```
END SIFRA_PACK;
```

-- DDL for Procedure PONUDA_UKUPNO_PROC

```
set define off;
```

```
CREATE OR REPLACE PROCEDURE "DB02"."PONUDA_UKUPNO_PROC" (br_pon IN NUMBER) AS
```

```
suma NUMBER:=0;
```

```
BEGIN
```

```
SELECT SUM(a.trenutna_cena*sp.kolicina) INTO suma
```

```
FROM ARTIKAL_DETALJI a join stavkaponude sp on
```

```
(a.sifraartikla=sp.sifraartikla)
```

```
WHERE brponude=br_pon;
```

```
UPDATE ponuda
```

```
SET UKUPNI_IZNOS=suma
```

```
WHERE brponude=br_pon;
```

```
END ponuda_ukupno_proc;
```

-- DDL for Trigger PONUDA_UKUPNO

```
CREATE OR REPLACE TRIGGER "DB02"."PONUDA_UKUPNO"
```

```
BEFORE INSERT OR UPDATE OR DELETE
```

```
ON stavkaponude
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF (INSERTING OR UPDATING) then
```

```
BEGIN
```

```
SIFRA_PACK.brponude := :NEW.brponude;
```

```

END;
ELSE
BEGIN
SIFRA_PACK.brponude:= :OLD.brponude;
END;
END IF;
END;
/
ALTER TRIGGER "DB02"."PONUDA_UKUPNO" ENABLE;

```

```

-----
-- DDL for Trigger PONUDA_UKUPNO_2
-----

```

```

CREATE OR REPLACE TRIGGER "DB02"."PONUDA_UKUPNO_2"
AFTER INSERT OR UPDATE OR DELETE ON stavkaponude
DECLARE
br_pon NUMBER:=SIFRA_PACK.brponude;
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azur_ukupni_iznos DISABLE';

PONUDA_UKUPNO_PROC(br_pon);

COMMIT;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azur_ukupni_iznos ENABLE';
END;
END;
/
ALTER TRIGGER "DB02"."PONUDA_UKUPNO_2" ENABLE;

```

```

-----
-- DDL for Procedure CENA_ARTIKLA
-----

```

```

set define off;

CREATE OR REPLACE PROCEDURE "DB02"."CENA_ARTIKLA" (sifra_art IN NUMBER)
as
cena_art ARTIKAL_DETALJI.TRENUTNA_CENA%type;

begin

cena_art:=0;
select (c.cena*(1 + c.STOPAPDV/100)) into cena_art from cena c

```

```
where c.sifraartikla = sifra_art and datum =
(select max(datum) from cena where sifraartikla = sifra_art);
```

```
update ARTIKAL_DETALJI
set TRENUTNA_CENA = cena_art
where sifraartikla = sifra_art;
```

```
end cena_artikla;
```

```
-----
-- DDL for Trigger UNESI_AKTUELNU_CENU1
-----
```

```
CREATE OR REPLACE TRIGGER "DB02"."UNESI_AKTUELNU_CENU1"
BEFORE INSERT or UPDATE or DELETE of cena
on cena
FOR EACH row
BEGIN
IF (INSERTING OR UPDATING) then
BEGIN

SIFRA_PACK.sifra := :NEW.sifraartikla;

END;
ELSE
BEGIN
SIFRA_PACK.sifra := :OLD.sifraartikla;
END;
END IF;
END;
/
ALTER TRIGGER "DB02"."UNESI_AKTUELNU_CENU1" ENABLE;
```

```
-----
-- DDL for Trigger UNESI_AKTUELNU_CENU2
-----
```

```
CREATE OR REPLACE TRIGGER "DB02"."UNESI_AKTUELNU_CENU2"
AFTER INSERT or UPDATE of cena on cena
DECLARE
SIFRA_ART NUMBER:= SIFRA_PACK.sifra;
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azur_trenutna_cena DISABLE';
```

```

CENA_ARTIKLA(SIFRA_ART);

COMMIT;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azur_trenutna_cena ENABLE';
END;
END;
/
ALTER TRIGGER "DB02"."UNESI_AKTUELNU_CENU2" ENABLE;

```

KORISNIČKI DEFINISANI TIP

```

-----
-- DDL for Type DETALJI_KUPCA
-----

```

```

CREATE OR REPLACE TYPE "DB02"."DETALJI_KUPCA"
AS
OBJECT
(
  adresa VARCHAR2(30),
  PIB NUMBER(8),
  brtel NUMBER(8),
  email VARCHAR2(30),

  MEMBER FUNCTION get_adresa
  RETURN VARCHAR2,
  MEMBER FUNCTION get_PIB
  RETURN NUMBER,
  MEMBER FUNCTION get_brtel
  RETURN NUMBER,
  MEMBER FUNCTION get_email
  RETURN VARCHAR2
)

INSTANTIABLE NOT FINAL;
/
CREATE OR REPLACE TYPE BODY "DB02"."DETALJI_KUPCA"
AS
  MEMBER FUNCTION get_adresa
  RETURN VARCHAR2
IS

```

```

BEGIN
  RETURN SELF.adresa;
END;
MEMBER FUNCTION get_PIB
  RETURN NUMBER
IS
BEGIN
  RETURN SELF.PIB;
END;
MEMBER FUNCTION get_brtel
  RETURN NUMBER
IS
BEGIN
  RETURN SELF.brtel;
END;
MEMBER FUNCTION get_email
  RETURN VARCHAR2
IS
BEGIN
  RETURN SELF.email;
END;
END;

```

/

-- DDL for Table KUPAC

```

CREATE TABLE "DB02"."KUPAC"
(
  "SIFRAKUPCA" NUMBER(5,0),
  "IMEKUPCA" VARCHAR2(30 BYTE),
  "DETALJI" "DB02"."DETALJI_KUPCA"
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

```

-- DDL for View KUPAC_VIEW

```

CREATE OR REPLACE FORCE VIEW "DB02"."KUPAC_VIEW" ("SIFRAKUPCA", "IMEKUPCA", "ADRESA", "PIB",
"BRTEL", "EMAIL") AS
SELECT
k.SIFRAKUPCA "SIFRAKUPCA",
k.IMEKUPCA "IMEKUPCA",
k.DETALJ.get_adresa() "ADRESA",
k.DETALJ.get_pib() "PIB",
k.DETALJ.get_brtel() "BRTEL",
k.DETALJ.get_email() "EMAIL"
FROM kupac k;

```

TRIGERI

```

-----
-- DDL for Trigger AZURIRAJ_IMEKUPCA
-----

```

```

CREATE OR REPLACE TRIGGER "DB02"."AZURIRAJ_IMEKUPCA"
AFTER UPDATE of imekupca ON kupac
FOR EACH ROW
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azuriranje_imekupca DISABLE';
UPDATE PONUDA
set PONUDA.IMEKUPCA = :NEW.IMEKUPCA
where ponuda.SIFRAKUPCA = :NEW.SIFRAKUPCA;
COMMIT;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azuriranje_imekupca ENABLE';
END;
END;
/
ALTER TRIGGER "DB02"."AZURIRAJ_IMEKUPCA" ENABLE;

```

```

-----
-- DDL for Trigger AZURIRAJ_IMEKUPCA_PO_SIFRI
-----

```

```

CREATE OR REPLACE TRIGGER "DB02"."AZURIRAJ_IMEKUPCA_PO_SIFRI"
BEFORE UPDATE of sifrakupca ON ponuda

```

```

FOR EACH ROW
DECLARE
n_imekupca VARCHAR2(30);
BEGIN
  SELECT IMEKUPCA
  INTO n_imekupca
  FROM KUPAC
  where SIFRAKUPCA =:NEW.SIFRAKUPCA;
  :new.imekupca := n_imekupca;
END;
/
ALTER TRIGGER "DB02"."AZURIRAJ_IMEKUPCA_PO_SIFRI" ENABLE;

```

```

-----
-- DDL for Trigger AZURIRAJ_NAZIVARTIKLA
-----

```

```

  CREATE OR REPLACE TRIGGER "DB02"."AZURIRAJ_NAZIVARTIKLA"
after UPDATE of nazivartikla ON artikal
FOR EACH ROW
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azur_nazivartikla DISABLE';
UPDATE CENA
set cena.nazivartikla = :NEW.nazivartikla
where SIFRAARTIKLA = :NEW.SIFRAARTIKLA;
COMMIT;
BEGIN
EXECUTE IMMEDIATE 'ALTER TRIGGER zabrani_azur_nazivartikla ENABLE';
END;
END;
/
ALTER TRIGGER "DB02"."AZURIRAJ_NAZIVARTIKLA" ENABLE;

```

```

-----
-- DDL for Trigger STAVKAPONUDE_TRG3
-----

```

```

  CREATE OR REPLACE TRIGGER "DB02"."STAVKAPONUDE_TRG3"
BEFORE INSERT ON STAVKAPONUDE
FOR EACH ROW
declare tmp_rbr number;

BEGIN

```

```

IF INSERTING AND :NEW.RBR IS NULL THEN
  select COUNT(rbr) INTO tmp_rbr FROM stavkaponude where brponude = :new.brponude;
  tmp_rbr := tmp_rbr +1;
  :NEW.RBR := tmp_rbr;
END IF;
END;
/
ALTER TRIGGER "DB02"."STAVKAPONUDE_TRG3" ENABLE;

```

```

-----
-- DDL for Trigger UNESI_IMEKUPCA
-----

```

```

CREATE OR REPLACE TRIGGER "DB02"."UNESI_IMEKUPCA"
BEFORE INSERT ON ponuda
FOR EACH ROW

```

```

DECLARE n_imekupca VARCHAR2(30);

```

```

BEGIN
  SELECT IMEKUPCA
  INTO n_imekupca
  FROM KUPAC
  where SIFRAKUPCA =:NEW.SIFRAKUPCA;
  :new.imekupca := n_imekupca;
END;
/

```

```

ALTER TRIGGER "DB02"."UNESI_IMEKUPCA" ENABLE;

```

```

-----
-- DDL for Trigger UNESI_NAZIVARTIKLA
-----

```

```

CREATE OR REPLACE TRIGGER "DB02"."UNESI_NAZIVARTIKLA"
before INSERT ON CENA
FOR EACH ROW

```

```

DECLARE n_nazivartikla VARCHAR2(30);

```

```

BEGIN
  SELECT nazivartikla
  INTO n_nazivartikla
  FROM ARTIKAL
  where SIFRAARTIKLA =:NEW.SIFRAARTIKLA;
  :new.nazivartikla := n_nazivartikla;
END;

```

```

/
ALTER TRIGGER "DB02"."UNESI_NAZIVARTIKLA" ENABLE;
-----
-- DDL for Trigger ZABRANI_AZURIRANJE_IMEKUPCA
-----

CREATE OR REPLACE TRIGGER "DB02"."ZABRANI_AZURIRANJE_IMEKUPCA"
AFTER UPDATE of imekupca ON ponuda
FOR EACH ROW

BEGIN
RAISE_APPLICATION_ERROR (
NUM => -20000,
MSG => 'NE SMETE DA AZURIRATE IME KUPCA ODAVDE');
END;
/
ALTER TRIGGER "DB02"."ZABRANI_AZURIRANJE_IMEKUPCA" ENABLE;
-----
-- DDL for Trigger ZABRANI_AZUR_NAZIVARTIKLA
-----

CREATE OR REPLACE TRIGGER "DB02"."ZABRANI_AZUR_NAZIVARTIKLA"
AFTER UPDATE of nazivartikla
ON cena
FOR EACH ROW

BEGIN
RAISE_APPLICATION_ERROR (
NUM => -20000,
MSG => 'NE SMETE DA AZURIRATE NAZIV ARTIKLA ODAVDE');
END;
/
ALTER TRIGGER "DB02"."ZABRANI_AZUR_NAZIVARTIKLA" ENABLE;
-----
-- DDL for Trigger ZABRANI_AZUR_TRENUTNA_CENA
-----

CREATE OR REPLACE TRIGGER "DB02"."ZABRANI_AZUR_TRENUTNA_CENA"
AFTER UPDATE of trenutna_cena
ON artikal_detalji
FOR EACH ROW

BEGIN
RAISE_APPLICATION_ERROR (
NUM => -20000,

```

```
MSG => 'NE SMETE DA DIREKTNO AZURIRATE TRENUTNU CENU');
END;
/
ALTER TRIGGER "DB02"."ZABRANI_AZUR_TRENUTNA_CENA" DISABLE;
-----
-- DDL for Trigger ZABRANI_AZUR_UKUPNI_IZNOS
-----

CREATE OR REPLACE TRIGGER "DB02"."ZABRANI_AZUR_UKUPNI_IZNOS"
BEFORE UPDATE of ukupni_iznos
ON ponuda
FOR EACH ROW

BEGIN
RAISE_APPLICATION_ERROR (
NUM => -20000,
MSG => 'NE SMETE DA DIREKTNO AZURIRATE UKUPNI IZNOS');
END;
/
ALTER TRIGGER "DB02"."ZABRANI_AZUR_UKUPNI_IZNOS" DISABLE;
```