

Sveučilište J.J. Strossmayera u
Osijeku
Fakultet primijenjene matematike i
informatike

SEMINARSKI RAD - MSBP

BAZA PODATAKA ZA AUTOŠKOLU

TIN ARAMBAŠIĆ

Osijek, lipanj 2025.

SADRŽAJ

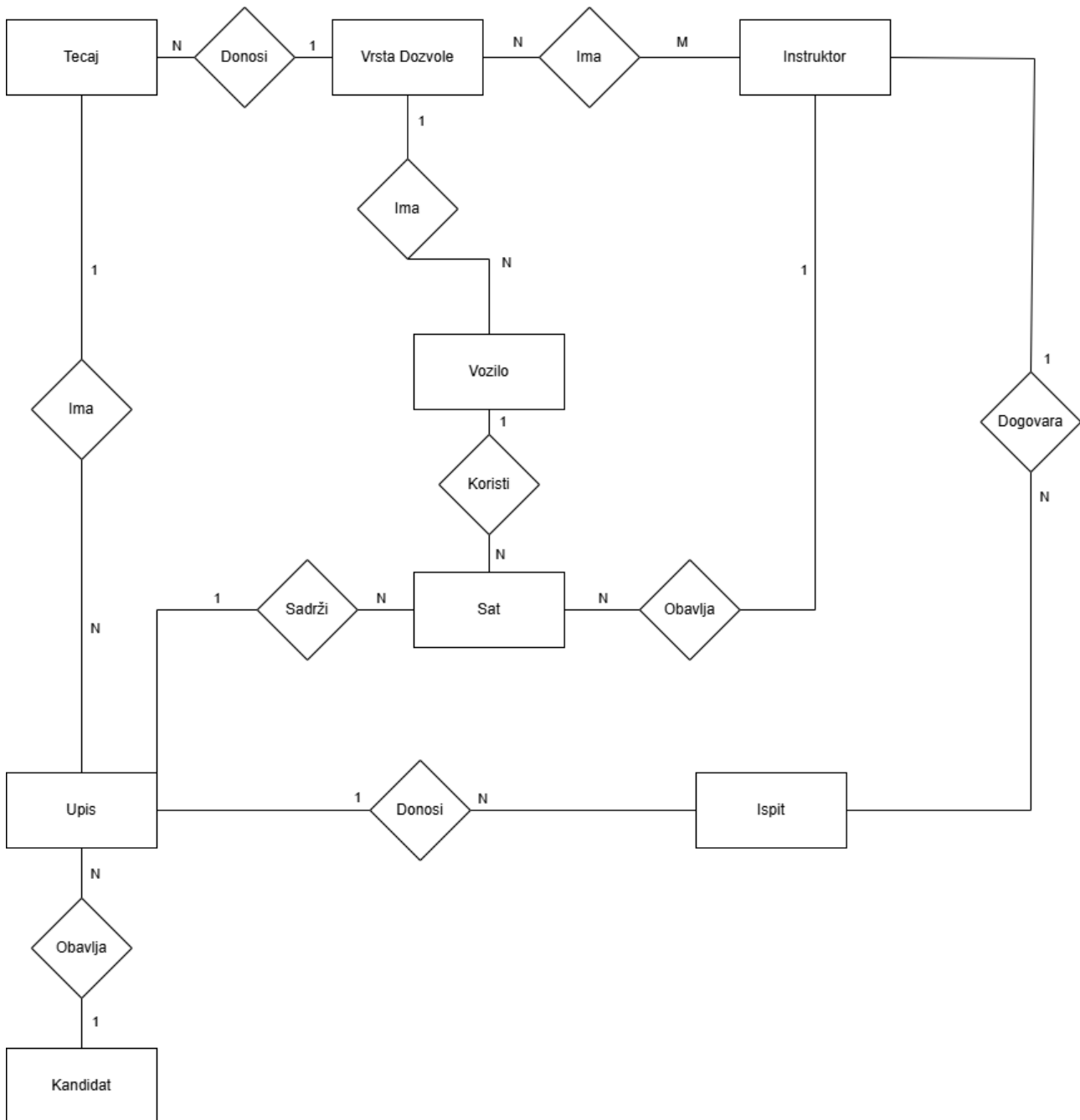
1. UVOD	3
2. Model baze podataka	4
2.1. MEV Dijagram.....	4
2.2. Relacijski model.....	5
2.3. Kreiranje tablica.....	5
2.4. Unos podataka u tablice	6
3. Upiti	7
3.1. Jednostavni upiti	7
3.1.1. Upit 1	7
3.1.2. Upit 2.....	8
3.1.3. Upit 3.....	8
3.2. Složeni upiti	9
3.2.1. Upit 1	9
3.2.2. Upit 2.....	10
3.2.3. Upit 3.....	11
4. Okidači	12
5. Procedre	13
6. Funkcije.....	14
7. Zaključak.....	15

1. UVOD

Tema ovog seminarskog rada obuhvaća sve potrebne atribute i informacije za vođenje jedne autoškole u nekome gradu te omogućuje lak pristup svim potrebnim podacima kojima bi netko želio pristupiti. Pored pristupa, podaci se također mogu i ukloniti ili promijeniti lakše nego vodeći podatke preko papira.

2. Model baze podataka

2.1. MEV Dijagram



Sve veze su dohvatljive preko upisa, tj. sve tablice se mogu pristupiti iz drugih tablica. Instruktor i kandidat su glavne tablice pošto sve ostalo mora biti popunjeno barem jednim podatkom iz te dvije tablice.

2.2. Relacijski model

Kandidat		
int	KandidatID	PK
string	Ime	
string	Prezime	
string	Adresa	
string	Telefon	
string	Email	
date	DatumUpisa	

INSTRUKTOR		
int	InstruktorID	PK
string	Ime	
string	Prezime	
string	Telefon	
string	Email	
date	DatumZaposlenja	

VRSTADOZVOLE		
int	VrstaDozvoleID	PK
string	Kategorija	
string	Opis	

INSTRUKTORDOZVOLE		
int	InstruktorID	FK
int	VrstaDozvoleID	FK

VOZILO		
int	VoziloID	PK
string	Registracija	
string	Marka	
string	Model	
int	GodinaProizvodnje	
string	VrstaMjenjaca	
int	VrstaDozvoleID	FK

TECAJ		
int	TecajID	PK
string	Naziv	
string	Opis	
int	Trajanje	
numeric	Cijena	
int	VrstaDozvoleID	FK

UPIS		
int	UpisID	PK
int	KandidatID	FK
int	TecajID	FK
date	DatumUpisa	
string	Status	

SAT		
int	SatID	PK
int	UpisID	FK
int	InstruktorID	FK
int	VoziloID	FK
datetime	DatumVrijeme	
int	Trajanje	
string	VrstaSata	

ISPIT		
int	IspitID	PK
int	UpisID	FK
date	DatumIspita	
string	Rezultat	
int	InstruktorID	FK

Svi odnosi jedan-prema-više prikazani su stranim ključevima u tablicama. Veza više-prema-više prikazana je svojom zasebnom tablicom (tablica InstruktorDozvole). Sve vrijednosti u tablicama se smatraju obvezne.

2.3. Kreiranje tablica

```
CREATE TABLE Kandidat (
    KandidatID SERIAL PRIMARY KEY,
    Ime VARCHAR(50) NOT NULL,
    Prezime VARCHAR(50) NOT NULL,
    Adresa TEXT NOT NULL,
    Telefon VARCHAR(20) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    DatumUpisa DATE NOT NULL
);
```

```
CREATE TABLE Instruktor (
    InstruktorID SERIAL PRIMARY KEY,
    Ime VARCHAR(50) NOT NULL,
    Prezime VARCHAR(50) NOT NULL,
    Telefon VARCHAR(20) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    DatumZaposlenja DATE NOT NULL
);
```

Tablice se kreiraju naredbom CREATE TABLE te unutar zagrada definiramo tipove podatka i nazive varijabli za tablicu koju kreiramo.

Primarni ključ označava se sa naredbom PRIMARY KEY te se nijedan primarni ključ ne smije podudarati s drugim u istoj tablici.

```
CREATE TABLE Upis (  
    UpisID SERIAL PRIMARY KEY,  
    KandidatID INTEGER NOT NULL,  
    TecajID INTEGER NOT NULL,  
    DatumUpisa DATE NOT NULL,  
    Status VARCHAR(20) NOT NULL CHECK(Status IN ('Aktivan', 'Završen', 'Otkazan')),  
    FOREIGN KEY (KandidatID) REFERENCES Kandidat(KandidatID),  
    FOREIGN KEY (TecajID) REFERENCES Tecaj(TecajID)  
);
```

Strani ključ se definira sa FOREIGN KEY i referencira se na primarni ključ neke druge tablice.

2.4. Unos podataka u tablice

```
INSERT INTO Instruktor (Ime, Prezime, Telefon, Email, DatumZaposlenja) VALUES  
( 'Ivan', 'Horvat', '0911122333', 'ivan.horvat@autoskola.hr', '2018-03-12'),  
( 'Sanja', 'Kovačić', '0922233444', 'sanja.kovacic@autoskola.hr', '2019-05-20'),  
( 'Mario', 'Babić', '0953344555', 'mario.babic@autoskola.hr', '2020-01-15'),  
( 'Tanja', 'Jurić', '0984455666', 'tanja.juric@autoskola.hr', '2020-08-30');
```

Svi podaci koji su obvezni moraju se unijeti u tablice(ako piše NOT NULL u deklaraciji)

3. Upiti

3.1. Jednostavni upiti

3.1.1. Upit 1

```
Select Marka, Model from Vozilo where VrstaMjenjaca='Ručni';
```

Upit vraća Marku i Model za sva vozila koja imaju ručni mjenjač.

Rezultat upita:

	marka character varying (50) 🔒	model character varying (50) 🔒
1	Yamaha	YZF-R125
2	Kawasaki	Ninja 400
3	BMW	R 1250 GS
4	MAN	TGE 3.140
5	Honda	CBR125R
6	Yamaha	MT-07
7	Ducati	Multistrada V4
8	KTM	125 Duke
9	Honda	CBR500R
10	KTM	1290 Super Duke R
11	Renault	Clio
12	Ford	Transit
13	Suzuki	GSX-R125
14	KTM	390 Duke
15	Harley-Davidson	Sportster S

3.1.2. Upit 2

Upit koji vraća ID kandidata koji su i dalje aktivni u tečaju koji ima ID=2

```
Select KandidatID, TecajID, Status from Upis where TecajID=2 and Status='Aktivan';
```

	kandidatid integer	tecajid integer	status character varying (20)
1	4	2	Aktivan
2	20	2	Aktivan

3.1.3. Upit 3

Jednostavan upit koji vraća Ime i Prezime kandidata te datum upisa kandidata za sve kandidate koji su upisali tečaj između 1.5.2024. i 1.8.2024.

```
Select Ime || ' ' || Prezime as "Kandidat", DatumUpisa from Kandidat where DatumUpisa between '2024-05-01' and '2024-08-01';
```

	Kandidat text	datumupisa date
1	Ante Lovrić	2024-05-03
2	Marija Vuković	2024-05-10
3	Davor Radić	2024-05-20
4	Lara Petrović	2024-06-01
5	Karlo Matić	2024-06-08
6	Tea Kralj	2024-06-15
7	Zoran Perić	2024-06-25
8	Sandra Jakovljević	2024-07-03
9	Tomislav Šimić	2024-07-10
10	Nika Kovačić	2024-07-18

U upitu se može uočiti konkatencija 2 redka tablice u jedan, to se postiže operatorom

|| ' ' ||

3.2. Složeni upiti

3.2.1. Upit 1

```
select Rezultat, Ime || ' ' || Prezime as "Kandidat" from Ispit  
left join Upis using(UpisID) left join Kandidat using (KandidatID);
```

Upit povezuje tablicu Ispit sa tablicom Kandidat preko tablice Upis te ispisuje imena i prezimena kandidata i rezultate njihovih ispita.

	rezultat character varying (20)	Kandidat text
1	Položen	Marko Horvat
2	Položen	Ana Kovačević
3	Nije položen	Ivan Babić
4	Nije izašao	Petra Jurić
5	Položen	Luka Novak
6	Položen	Maja Knežević
7	Položen	Josip Marković
8	Nije položen	Iva Tomić
9	Nije izašao	Filip Pavlović
10	Položen	Elena Božić
11	Nije položen	Ante Lovrić
12	Položen	Marija Vuković
13	Položen	Davor Radić
14	Nije položen	Lara Petrović
15	Položen	Karlo Matić
16	Nije položen	Tea Kralj
17	Položen	Zoran Perić
18	Položen	Sandra Jakovljević
19	Nije položen	Tomislav Šimić
20	Položen	Nika Kovačić

3.2.2. Upit 2

Upit koji vraća kategoriju vozila, broj vozila iz svake kategorije, broj sati iz svake obuke, i postotak sati polaganja za tu kategoriju

```
SELECT
    vd.Kategorija,
    COUNT(DISTINCT v.VoziloID) AS BrojVozila,
    COUNT(s.SatID) AS BrojSati,
    ROUND(COUNT(s.SatID) * 100.0 / SUM(COUNT(s.SatID)) OVER (), 1) AS Postotak
FROM Vozilo v
JOIN VrstaDozvole vd ON v.VrstaDozvoleID = vd.VrstaDozvoleID
LEFT JOIN Sat s ON v.VoziloID = s.VoziloID
GROUP BY vd.Kategorija
ORDER BY BrojSati DESC;
```

	kategorija character varying (10)	brojvozila bigint	brojsati bigint	postotak numeric
1	B	4	5	25.0
2	A1	4	4	20.0
3	A2	4	4	20.0
4	C	4	4	20.0
5	A	4	3	15.0

3.2.3. Upit 3

Upit vraća ukupne obuke instruktora po kategoriji sa detaljima vozila i postotkom posla

```
WITH InstruktorSati AS (
    SELECT
        i.InstruktorID,
        i.Ime || ' ' || i.Prezime AS Instruktor,
        vd.Kategorija,
        COUNT(s.SatID) AS BrojSati,
        SUM(s.Trajanje) AS UkupnoTrajanje,
        v.Marka || ' ' || v.Model AS Vozila
    FROM Instruktor i
    JOIN Sat s ON i.InstruktorID = s.InstruktorID
    JOIN Vozilo v ON s.VoziloID = v.VoziloID
    JOIN VrstaDozvole vd ON v.VrstaDozvoleID = vd.VrstaDozvoleID
    GROUP BY i.InstruktorID, vd.Kategorija, v.marka, v.model
),
UkupniSati AS (
    SELECT SUM(BrojSati) AS TotalSati
    FROM InstruktorSati
)
SELECT
    isa.Instruktor,
    isa.Kategorija,
    isa.BrojSati,
    isa.UkupnoTrajanje,
    isa.Vozila,
    ROUND((isa.BrojSati * 100.0) / us.TotalSati, 2) AS PostotakUkupno
FROM InstruktorSati isa
CROSS JOIN UkupniSati us
ORDER BY BrojSati DESC;
```

	instruktor text	kategorija character varying (10)	brojsati bigint	ukupnotrajanje bigint	vozila text	postotakukupno numeric
1	Ivan Horvat	B	2	90	Volkswagen Golf 8	10.00
2	Sanja Kovačić	B	1	60	Škoda Octavia	5.00
3	Mario Babić	C	1	150	Ford Transit	5.00
4	Mario Babić	A2	1	60	Yamaha MT-07	5.00
5	Sanja Kovačić	C	1	180	Mercedes Sprinter	5.00
6	Ivan Horvat	A1	1	90	Yamaha YZF-R125	5.00
7	Tanja Jurić	A	1	90	Ducati Multistrada V4	5.00
8	Tanja Jurić	B	1	60	Peugeot 208	5.00
9	Tanja Jurić	A2	1	90	Honda CBR500R	5.00
10	Mario Babić	A1	1	45	KTM 125 Duke	5.00
11	Sanja Kovačić	A	1	90	Ducati Multistrada V4	5.00
12	Tanja Jurić	A1	1	60	Suzuki GSX-R125	5.00
13	Mario Babić	B	1	90	Renault Clio	5.00
14	Ivan Horvat	C	1	120	MAN TGE 3.140	5.00
15	Tanja Jurić	C	1	120	Iveco Daily	5.00
16	Ivan Horvat	A2	1	60	Kawasaki Ninja 400	5.00
17	Sanja Kovačić	A2	1	90	Yamaha MT-07	5.00
18	Sanja Kovačić	A1	1	45	Honda CBR125R	5.00
19	Ivan Horvat	A	1	120	BMW R 1250 GS	5.00

4. Okidači

Okidač koji vraća grešku ako se pokuša unijeti datum završetka kandidata koji je raniji nego datum upisa u autoškolu.

```
ALTER TABLE Upis ADD COLUMN DatumZavrsetka DATE;
commit;

CREATE OR REPLACE FUNCTION provjeri_datume()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.DatumZavrsetka IS NOT NULL AND NEW.DatumZavrsetka < NEW.DatumUpisa THEN
        RAISE EXCEPTION 'Datum završetka % ne može biti prije datuma upisa %',
            NEW.DatumZavrsetka, NEW.DatumUpisa;
    END IF;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;
commit;

CREATE TRIGGER sprijeci_datume
BEFORE INSERT OR UPDATE ON Upis
FOR EACH ROW
EXECUTE FUNCTION provjeri_datume();
commit;
```

342 ▼

343

```
INSERT INTO Upis (KandidatID, TecajID, DatumUpisa, DatumZavrsetka, Status)
VALUES (2, 5, '2024-03-01', '2024-02-01', 'Završen');
```

Data Output

Messages

Notifications

```
ERROR: Datum završetka 2024-02-01 ne može biti prije datuma upisa 2024-03-01
CONTEXT: PL/pgSQL function provjeri_datume() line 4 at RAISE
```

5. Procudre

Procedura koja na poziv mijenja cijenu kategorije C

```
CREATE OR REPLACE PROCEDURE promijeni_cijenu_kategorije_C(  
    IN nova_cijena NUMERIC(10,2)  
)  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    UPDATE Tecaj  
    SET Cijena = nova_cijena  
    WHERE VrstaDozvoleID = (  
        SELECT VrstaDozvoleID  
        FROM VrstaDozvole  
        WHERE Kategorija = 'C'  
    );  
  
    RAISE NOTICE 'Cijena za kategoriju C promijenjena na % kn', nova_cijena;  
  
    RAISE NOTICE 'Promjena primijenjena na % tečaj/a', (SELECT COUNT(*) FROM Tecaj WHERE VrstaDozvoleID = (  
        SELECT VrstaDozvoleID  
        FROM VrstaDozvole  
        WHERE Kategorija = 'C'  
    ));  
END;  
$$;  
commit;
```

```
371 call promijeni_cijenu_kategorije_C(5000.00);
```

Data Output Messages Notifications

NOTICE: Cijena za kategoriju C promijenjena na 5000.00 kn
NOTICE: Promjena primijenjena na 1 tečajeva
CALL

```
373 SELECT t.Naziv, t.Cijena, v.Kategorija  
374 FROM Tecaj t  
375 JOIN VrstaDozvole v ON t.VrstaDozvoleID = v.VrstaDozvoleID  
376 WHERE v.Kategorija = 'C';
```

Data Output Messages Notifications

	naziv character varying (100)	cijena numeric (10,2)	kategorija character varying (10)
1	Tečaj za C	5000.00	C

6. Funkcije

Funkcija koja računa godine iskustva instruktora

```
CREATE OR REPLACE FUNCTION izracunaj_godine_iskustva(
    p_instruktor_id INTEGER
)
RETURNS NUMERIC(5,2)
LANGUAGE plpgsql
AS $$
DECLARE
    v_datum_zaposlenja DATE;
    v_godine_iskustva NUMERIC(5,2);
BEGIN

    SELECT DatumZaposlenja INTO v_datum_zaposlenja
    FROM Instruktor
    WHERE InstruktorID = p_instruktor_id;

    IF v_datum_zaposlenja IS NULL THEN
        RETURN 0;
    ELSE
        v_godine_iskustva := EXTRACT(YEAR FROM age(current_date, v_datum_zaposlenja)) +
            (EXTRACT(MONTH FROM age(current_date, v_datum_zaposlenja)) / 12.0);
        RETURN ROUND(v_godine_iskustva, 2);
    END IF;
END;
$$;
commit;
```

```
406 SELECT
407     Ime || ' ' || Prezime AS Instruktor,
408     DatumZaposlenja,
409     izracunaj_godine_iskustva(InstruktorID) AS GodineIskustva
410 FROM Instruktor
411 ORDER BY GodineIskustva DESC;
```

Data Output Messages Notifications

	instruktor text	datumzaposlenja date	godineiskustva numeric
1	Ivan Horvat	2018-03-12	7.25
2	Sanja Kovačić	2019-05-20	6.08
3	Mario Babić	2020-01-15	5.42
4	Tanja Jurić	2020-08-30	4.83

7. Zaključak

Baze podataka su izuzetno potrebne u svakoj djelatnosti, od autoškoli sve do zdravstvenih ustanovi. Upiti i funkcije su jedne od mandatornih stvari u bazi podataka pošto nam omogućuju najbolji pregled podataka i računanje nad podacima. Okidači su također korisni jer nas sprječavaju od grešaka koje mogu narušiti naše podatke i stabilnost baze.